AppDynamics Application Performance Monitoring Platform 21.4

AppDynamics Application Performance Monitoring Platform

This page provides information on installing, configuring, and administering an on-premises AppDynamics Application Performance Monitoring (APM) Platform deployment.

Installation Overview

Before you install the platform, review the requirements for the components you plan to install and prepare the host machines. The requirements vary based on the components you deploy and the size of your deployment.

For the Controller and Events Service, you first need to install the AppDynamics Enterprise Console. You then use the application to deploy the Controller and Events Service. Note that the Events Service can be deployed as a single node or a cluster. The Enterprise Console is not only the installer for the Controller and Events Service; it can manage the entire lifecycle of new or existing AppDynamics Platforms and components.

You cannot use the Enterprise Console to perform the End User Monitoring (EUM) Server installation. Instead, you must use a package installer that supports interactive GUI or console modes, or a silent response file installation.

Follow these tasks before you start the installation process for the AppDynamics APM Platform:

- Review the Platform Requirements.
- Verify the Enterprise Console host meets the requirements to host the application and the Controller since they share the same host by default for Express Install. You do have the option to install the Enterprise Console on a different host than the Controller's using Custom Install.
- On Linux, verify that you have assigned execute permissions to the installation script with the following command:

```
chmod 775 platform-setup-64bit-linux.sh
```

You can get the software for installing the platform components from the AppDynamics download site. See Download AppDynamics Software

Platform Components and Tools

An on-premises AppDynamics Platform installation consists of several, separately installed and configured components. These include the Controller, MySQL database, Events Service, and optionally the EUM Server.

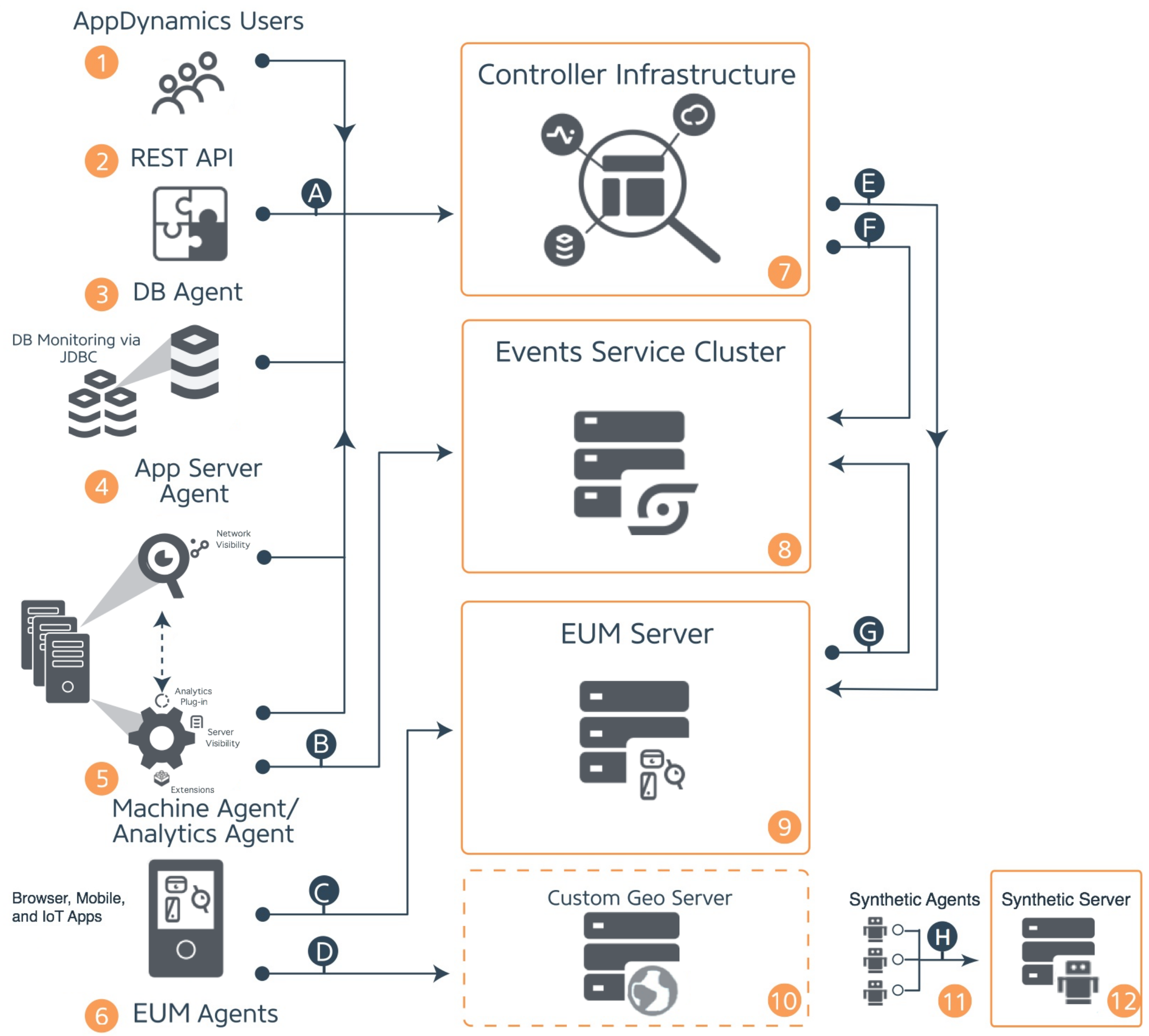
The AppDynamics Enterprise Console is a GUI and command-line based application that can manage the installation, configuration, and administration of the Controller and Events Service.

For the EUM Server, you must continue to use the package installer to deploy the EUM Cloud. See EUM Server Deployment

After you install the platform, you can configure and manage different components with component-specific scripts. Based on how you deploy the platform, you might use a combination of the Enterprise Console and package installers to install and manage the various components of the platform.

On-Premises Deployment Architecture

The following diagram depicts the components of a complete on-premises AppDynamics APM Platform deployment. It shows how the components interact to fulfill application, database, infrastructure, end-user monitoring, and more.



Depending on the scale of your deployment, your requirements, and the products you are using, your own deployment is likely to consist of a subset of the components shown in the diagram.

You can find a more detailed diagram, as well as a SaaS architecture diagram on PDFs. For a diagram of the Enterprise Console, see the Enterprise Console Platforms Architecture. For a diagram of the Synthetic Server Deployment, see the Synthetic Server Deployment Architecture.

Platform Components

The following table describes how the components work together in the AppDynamics platform.

Product Feature	Components Involved
Application Performance Management	4 App Server Agents attach to monitored applications and send data to the 7 Controller via connection A.
Server Visibility	5 Machine Agents reside on monitored servers and report data to the 7 Controller via connection A.
Application Analytics	The Analytics Dynamic Service (formerly called the Analytics plugin) on the 4 App Server agent communicates with a local 5 Analytics Agent instance. One or more Analytics Agents in a deployment send data to the 8 Events Service via connection B. The Analytics Agent is bundled with the Machine Agent but can be installed and run individually as well.

Database Visibility	The ③ Database Agent connects by JDBC to monitored databases. The agent sends data to the ⑦ Controller (via connection Ⓐ), which uses the ⑧ Events Service to store certain types of data.
End-User Monitoring	For an on-premises EUM installation, you configure a connection to the web and mobile real user monitoring agents to the on-premises ⑨ EUM Server via connection Ⓒ. The EUM Server sends data to the ⑧ Events Service cluster via connection Ⓖ. The optional ⑩ Custom EUM Geo Server stores EUM Geo Resolution data taken via connection Ⓓ. The optional ⑫ Synthetic Server receives synthetic job requests from the Controller, which are then fetched from the Synthetic Services via connection Ⓗ.

Platform Connections

The following table lists and describes the traffic flow between AppDynamics platform components.

Connection	Source	Destination	Traffic	Protocol	Default Port(s)
Ⓐ	① AppDynamics users through the web GUI, ② REST API, ③ Database Agent, ④ Application Server Agent, and ⑤ Machine and Analytics Agents	⑦ Controller	APM/Database Metrics	HTTP	8090
				HTTPS	8181
Ⓑ	⑤ Analytics Agent	⑧ Events Service Cluster	Log and Transaction Analytics Event Data	HTTP	9080
Ⓒ	⑥ Real User Monitoring (RUM) Agents	⑨ End-User Monitoring (EUM) Server	EUM Beacon Data	HTTP	7001
				HTTPS	7002
Ⓓ	⑥ Real User Monitoring (RUM) Agents	⑩ Custom EUM Geo Server	EUM Geo Resolution Mapping Data	HTTP	80
				HTTPS	443
Ⓔ	⑦ Controller	⑨ EUM Server	EUM Metric Data	HTTP	7001
				HTTPS	7002 (demo mode only)
Ⓕ	⑦ Controller	⑧ Events Service Cluster	Events Service API Store	HTTP(S)	9080
			Events Service API Store Admin	HTTP(S)	9081
Ⓖ	⑨ EUM Server	⑧ Events Service Cluster	Events Service API Store (EUM Event Data)	HTTP(S)	9080
			Events Service API Store Admin (EUM Event Data)	HTTP(S)	9081
Ⓗ	⑪ Synthetic Agents	⑫ Synthetic Server	Synthetic Measurement Data	HTTP	10101
				HTTPS	10102

ⓘ The default port 9081 is the Admin port (HTTP).

Data Storage Location

Data is stored in the following locations:

- APM configuration and metric data in the on-premises Controller MySQL database
- EUM event data in the Events Service
- Transaction and log analytics data in the Events Service
- EUM Geo Resolution data in the on-premises GeoServer
- EUM Synthetic data in the on-premises Synthetic Server

Installation and Upgrade Overview

The installation and upgrade process for the AppDynamics platform consists of pre-installation steps to prepare your network and host machines for installation, installation tasks, and post-install steps to complete the required configuration. See Planning Your Deployment

After this process, you can perform optional configurations and administrative tasks described in Secure the Platform.

To start the installation or upgrade process, see Platform Requirements for information about requirements and pre-installation tasks.

Planning Your Deployment

This section provides an overview of how to plan your on-premises deployment.

Related pages:

- Platform Requirements
- Controller System Requirements
- EUM Server Requirements
- Events Service Requirements

Before You Begin

Before you upgrade or install the platform, perform the following tasks:

- Choose your appropriate Enterprise Console install or upgrade path.
- Review the requirements for the components you plan to install and prepare the host machines. The requirements vary based on the components you deploy and the size of your deployment.
- Download the Enterprise Console and start your platform installation.
- Verify that a user account with write permissions for the installation directory you want to use exists. Install all components with the same user or a user with equivalent permissions.

You can refer to the child pages for the platform requirements and deployment guides.

> ⓘ Irrespective of the server's computing power, installing more than one controller on the same server is not supported.

Choose How You Want to Install

The deployment option you choose to follow should be based on the components of the AppDynamics platform that you want to deploy. Deployment options that include the Controller and Events Service installations require the Enterprise Console, while the EUM Server installation requires the use of package installers.

Based on your AppDynamics deployment, you may use the Enterprise Console for the following tasks:

- Install a Controller and an embedded Events Service.
- Install an HA pair and scaled-out Events Service.
- Install or upgrade the Events Service on Linux.
- Install or upgrade an Events Service on Windows that runs on a single host.
- Discover and manage a Controller and Events Service.
- Upgrade a Controller or HA pair and preserve customizations after the upgrade.
- Upgrade MySQL to the latest supported version.

The following tasks cannot be performed using the Enterprise Console, and therefore, must be performed manually:

- Install or upgrade an HA pair on Windows.
- Install or upgrade an Events Service on Windows that runs on multiple hosts as a cluster.
- Install the EUM Server.
- Install the Synthetic Server.

Installation and Upgrade Quick Starts

Based on how you deploy and what components of the AppDynamics you deploy, there are different installation and upgrade steps to follow.

New Installations Quick Start

This installation path describes the use of the Enterprise Console to install the Controller and Events Service. The EUM Server must be installed separately as it cannot be installed using the Enterprise Console.

1. Install the Enterprise Console.
2. Use the Enterprise Console to create the platform and add hosts.

3. Install the Controller and Events Service on the same host by using Express Install. Use Custom Install to install a scaled-out Events Service that runs on a host that is separate from the Controller. Custom installations give more flexibility on where and how to install Controller and Events Service.
4. Install the EUM Server.
5. Complete the post-install tasks for the Controller, Events Service, and EUM Server.

For a more detailed description of this path, see Platform Installation Quick Start.

Upgrade Quick Start

This is the upgrade path to follow using the Enterprise Console and EUM package installer:

1. Install the Enterprise Console.
2. Discover and upgrade an existing Events Service with the Enterprise Console.
3. Upgrade the Production EUM Server with the package installer.

For a more detailed description of this path, see Discovery and Upgrade Quick Start.

Platform Component Compatibility

For versions 20.x+, to ensure that the platform components work properly, be sure you install compatible component versions based on the following guideline:

Platform Version	Platform Component Version Compatibility
20.2+	EUM Server version >= Controller version (Events Service 20.2+ is backward compatible with the other platform components.)

> The Synthetic Server 20.x+ versions are compatible with any of the other component versions 20.x+ and the Synthetic Private Agent 20.x+.

To get the versions of the Events Service, Controller, and Events Service, see Getting Platform Versions. For the EUM Server and Synthetic Server, use the following endpoints:

- `http(s)://<on-prem-eum-server_domain-name>:7001/eumcollector/get-version`
- `http(s)://<on-prem-synthetic-server_domain-name>:10101/version`

Download the Software

To use the Enterprise Console to install the AppDynamics Platform, you need to download the Enterprise Console installer and, if needed, the EUM Server package installer.

For more information on downloading the software, see Download AppDynamics Software.

Platform Requirements

The requirements for different components in the AppDynamics platform are based on the performance profile you select. This page describes the different performance profiles and how to determine the profile size you need.

CPU and Memory Space Requirements

When the Enterprise Console host is shared with the Controller host, it should have enough space to match the Controller host requirements, since there is no need for additional memory for the Enterprise Console.

However, when the Enterprise Console host is not shared with the Controller host, then it requires additional memory and disk space.

See Enterprise Console Requirements and Prepare the Controller Host for additional space requirements.

Network Considerations

Your network or the host machine may have built-in firewall rules that you will need to adjust to accommodate the AppDynamics on-premises platform. Specifically, you may need to permit network traffic on the ports used in the system. For more information, see Port Settings.

For expected bandwidth consumption for the agents, see the requirements documentation for app agents, see Install App Server Agents.

System User Account

You need to install all platform components with a single user account or accounts that have equivalent permissions on the operating system. The user needs to have write permissions for the installation directory.

Operating System Support

These operating systems support the AppDynamics platform:

Linux (64 bit)	Microsoft Windows (64 bit)
• RHEL 7.x and 8.x • CentOS 7 and 8.x • Ubuntu 14, 16, 18.x, and 20.x • openSUSE Leap 12 and 15 • Amazon Linux 1 and 2	• Windows Server 2012 and 2012 R2 • Windows Server 2016 • Windows Server 2019

You can use the following file systems for machines that run Linux:

- ZFS
- EXT4
- XFS

On-premises controller deployments on Linux are only supported on x86-64.

Internationalization Support

The Controller and App Agents provide full internationalization support, with support for double- and triple-byte characters. This support provides the following abilities:

- Controller UI users can enter double- or triple-byte characters into text fields in the UI.
- The Controller can accept data that contains double- or triple-byte characters from instrumented applications.

Network Bandwidth Requirements

See Administer App Server Agents for information on bandwidth usage in an AppDynamics deployment.

More Information

For requirements that are specific to product components, see the following pages:

- Controller System Requirements
- EUM Server Requirements
- Events Service Requirements
- Synthetic Server Requirements

Port Settings

When deploying AppDynamics, you may need to open ports in a network firewall or configure a load balancer to enable communication between the Controller and the rest of the AppDynamics platform.

For SaaS, you only need to adjust your infrastructure to accommodate the HTTPS port provided to you by AppDynamics. For an on-premises deployment, however, you may need to make additional adjustments based on the information here.

Platform Component Ports

The following ports are open in a platform deployment. The "external" column indicates whether connections to the port occur entirely within the Controller host or from outside the host, and therefore may require firewall or load balancer configuration changes.

Port Name	Default	External?
Enterprise Console port	9191	Yes. The application uses port 9191 for all traffic.
SSH port	22	The port needs to be open between the Enterprise Console and the remote hosts it manages. This is for Unix only and is not configurable. If you have a requirement to configure the port, contact AppDynamics support.
Database server port	3388	No
Default database port	3377	No
Application server admin port	4848	No
Application server JMS port	7676	No
Application server IIOP port	3700	No
Application server primary port (HTTP)	8090	Yes
Application server SSL port (HTTPS)	8181	Yes
Events Service REST API port	9080	If the Events Service and Controller are on different hosts, you need to configure the port in the firewall or load balancer.
Events Service REST API admin port	9081	If the Events Service and Controller are on different hosts, you need to configure the port in the firewall or load balancer.
Reporting service HTTP port	8020	No
Reporting service HTTPs port	8021	No
EUM server port (HTTP)	7001	If EUM and the Controller are on different hosts, you need to configure the port in the firewall or load balancer.
EUM server SSL port (HTTPS)	7002	If EUM and the Controller are on different hosts, you need to configure the port in the firewall or load balancer.

At installation time, you can enter different ports manually. After installation, you can change the port settings by either reinstalling the Controller or by editing the port configuration as defined on the **Enterprise Console Configurations** page or in the underlying GlassFish application server, as described in the following sections.

Editing Controller Port Configurations

You can modify connection settings through the Enterprise Console UI. The Enterprise Console will automatically update all occurrences in the controller for you. You do not need to manually update all the files and manage the sequence to restart services. See Update Platform Configurations.

You can also edit the ports manually by editing configuration files used by the application server for the Controller domain. Updating the ports manually, however, will cause the Enterprise Console to have no visibility into the updates and cause health-check errors.

The following sections list the settings you need to modify to change a port.

Change the Primary Server Listening Port

1. In `domain.xml`, change the port number as it appears in these locations:
    - The value of the `network-listener` element with the attribute id="`http-listener-1`" for the primary listening port, or `http-listener-2` for the secure listening port to the new port setting.
    - The JVM argument values for the Controller HTTP port and Controller services port under the config element named `server-config`.
2. For each deployed agent, navigate to proxy/conf in the agent home directory and change the `controller-port` value in `controller-info.xml`.

Change the Database Port

1. In `domain.xml`, change the database listening port where it appears under the `jdbc-connection-pool` element named `controller_mysql_pool`. It appears as the value of the property named `portNumbe` .
2. Edit the file `appserver/glassfish/domains/domain1/imq/instances/imqbroker/props/config.properties` to change the "`imq.persist.jdbc.mysql.property.url`" variable so that it includes the new port number. This variable is the JDBC connection string.
3. In `db/db.cnf`, set the "`port=`" variable to your new port setting.
4. In `bin/controller.bat (.sh)`, change the "`DB_PORT`" variable to your new port setting.

Change the Glassfish Admin Listening Port

1. In `domain.xml`, change the port attribute value of the `http-listener` element to the new port. This is the element with an id attribute value of "`admin-listener`".
2. Also in the Controller home directory, change the `adminPort` value in `.install4j/response.varfile`. This ensures that the new port number is not overwritten in a future Controller upgrade.

Change the JMS Port

1. In `domain.xml`, change the port attribute value for the `jms-host` element with the name attribute of `default_JMS_host`.
2. Change the `jmsPort` value in `.install4j/response.varfile`. This ensures that the new port number is not overwritten in a future Controller upgrade.

Change the IIOP Listening Port

1. In `domain.xml`, edit the port attribute value of the `iiop-listener` element with an id attribute of `orb-listener-1`.
2. Change the `iiopPort` value in `<controller_home>/.install4j/response.varfile`. This ensures that the new port number is not overwritten in a future Controller upgrade.

Enable Appserver Health Checks for HTTPS

If you disable or lock the Controller's HTTP port, you will need to configure the Appserver health check to contact the Controller's HTTPS port instead. You can do so by completing the following steps:

1. In `domain.xml`, set the HTTP listener, `http-listener-1`, to `enabled=false`.
2. Restart the Controller.
3. Use the Enterprise Console to discover and upgrade the Controller.
   The Enterprise Console will default to the HTTPS port.

Physical Machine Controller Deployment Guide

The following pages describe considerations and instructions for deploying the Controller on physical machines.

- Prepare the Controller Host
- Controller Data and Backups
- Migrate the Controller

Prepare the Controller Host

Related Pages:

- Controller System Requirements

This page describes the common configuration, tuning, and environment requirements for the machine that hosts the Controller.

These considerations apply whether the machine runs Linux or Windows or is a virtual machine. For specific considerations for your operating system type, see the related pages links.

Time Synchronization Service

A time synchronization service, such as the Network Time Protocol daemon (ntpd), should be enabled on the Controller host machine.

MySQL Conflict

Certain Linux installation types include MySQL as a bundled package. No MySQL instances other than the one included in the Controller host should run on the Controller host. Verify that no such MySQL processes are running.

Virtual Memory Space

The virtual memory size (swap space on Linux or Pagefile space on Windows) should be at least 10 GB on the target system, and ideally 20 GB.

Verify the size of virtual memory on your system and modify it if it is less than 10 GB. Refer to the documentation for your operating system for instructions on modifying the swap space or Pagefile size.

Disk Space

In addition to the minimum disk space required to install the Controller for your profile size, the Enterprise Console writes temporary files to the system temporary directory, typically /tmp on Linux or c:\tmp on Windows. The Enterprise Console requires 1024 MB of free temp space on the controller host.

On Windows, in case of an error due to not meeting the above requirement, you can set the temporary directory environment variable to a directory with sufficient space for the duration of the installation process. You can restore the setting to the original temp directory when the installation is complete.

Network Ports

Review the ports that the Controller uses to communicate with agents and the rest of the AppDynamics platform. For more information, see Port Settings.

Note that on Linux systems, port numbers below 1024 may be considered privileged ports that require root access to open. The default Controller listen ports are not configured for numbers under 1024, but if you intend to set them to a number below 1024 (such as 80 for the primary HTTP port), you need to run the Enterprise Console as the root user.

Prepare Linux for the Controller

This page describes the configuration requirements and considerations for using a Linux system as a Controller host machine.

User Account Requirements

The user account you use to perform the installation must have the following permissions:

- read, write and execute permissions on the directory where you install the Controller
- write permission on the `/etc/.java/.systemprefs` directory

If you are installing other AppDynamics Platform server components, such as the EUM Server or Application Analytics Processor, on the same machine, it is recommended that you perform the installation as the same user or a user with the same permissions on the target machine.

Virus Scanners

Configure virus scanners on the target machine to ignore the AppDynamics Enterprise Console directory and database directory (or simply the entire Controller directory). Code is never executed from the data directory, so it is generally safe to exclude this directory from virus scanning. The default location of the data directory is `<controller_home>/db/data`.

Also configure virus scanners to trust the Controller launcher, database executable, reporting service launcher, and events service (analytics processor) launchers. The launcher names are:

- Controller launcher: `AppDynamicsDomain1Service.sh`
- MySQL executable: `mysqld.sh`
- Events service launcher: `analytics-processor.sh`
- Reporting service launcher: `appdynamicsreportingservice.sh`

Anti-virus Exclusions

If you are running an antivirus program on your Linux system, it must meet one of the following conditions:

- The anti-virus program is read-only; it only detects and reports issues but never modifies files
- The anti-virus program excludes the MySQL data directory (`datadir`), which is often set to the path `db/data`.

If the program does not meet either of those conditions, it can randomly corrupt the MySQL database and hence the controller.

netstat Network Utility

Verify that your distribution of Linux includes the `netstat` network utility. If it does not, install the utility. The Controller installation uses `netstat` to determine whether MySQL processes are running.

For example, you can install the package that includes `netstat` with the following command on CentOS:

```
yum install net-tools
```

libaio Requirement

The Controller requires the `libaio` library to be on the system. This library facilitates asynchronous I/O operations on the system. Note if you have a NUMA based architecture, then you are required to install the `numactl` package.

Install `libaio` on the host machine if it does not already have it installed. The following table provides instructions on how to install `libaio` for some common flavors of Linux operating system.

Linux Flavor	Command

• Red Hat
• CentOS
• Amazon
	Use `yum` to install the library, such as:
	• `yum install libaio`
	• `yum install numactl`
	• `yum install tzdata`
	• `yum install ncurses-libs-5.x`
	For RHEL8, CentOS8, and Amazon2, you install the `ncurses-libs-5.x` library using a `rpm` file downloaded from a trusted source:
	`sudo rpm -ivh --force ncurses-base-5.x.rpm`
	`sudo rpm -ivh --force ncurses-libs-5.x.rpm`
	Note: The `ncurses-libs` depends on the `ncurses-base` so you should install the `ncurses-base` first.
	Example of a trusted source for `rpm` download:
	http://mirror.centos.org/centos/7/os/x86_64/Packages/ncurses-base-5.9-14.20130511.el7_4.noarch.rpm
	http://mirror.centos.org/centos/7/os/x86_64/Packages/ncurses-libs-5.9-14.20130511.el7_4.x86_64.rpm
	• To install version 6, follow these steps:
	You must either create symlinks for `ncurses-libs-5` which points to `ncurses-libs-6`, or install the `ncurses-compat-libs` package, to provide ABI version 5 compatibility.
	RHEL8 symlink:
	`sudo ln /usr/lib64/libtinfo.so.6.1 /usr/lib64/libtinfo.so.5`
	`sudo ln /usr/lib64/libncurses.so.6.1 /usr/lib64/libncurses.so.5`
	CentOS8 symlink:
	`sudo ln /usr/lib64/libtinfo.so.6.1 /usr/lib64/libtinfo.so.5`
	`sudo ln /usr/lib64/libncurses.so.6.1 /usr/lib64/libncurses.so.5`
	Amazon2 symlink:
	`sudo ln -s /usr/lib64/libncurses.so.6.0 /usr/lib64/libncurses.so.5`
	`sudo ln -s /usr/lib64/libtinfo.so.6.0 /usr/lib64/libtinfo.so.5`
	RHEL8 compat-libs:
	`sudo yum install -y ncurses-compat-libs`
	CentOS8 compat-libs:
	`sudo yum install -y ncurses-compat-libs`
	Amazon2 compat-libs:
	`sudo yum install -y ncurses-compat-libs`

Fedora	Install the library RPM from the Fedora website: • `yum install libaio` • `yum install numactl` • `yum install tzdata`

Ubuntu	Use `apt-get`, such as:
	- `sudo apt-get install libaio1`
	- `sudo apt-get install numactl`
	- `sudo apt-get install tzdata`
	- `sudo apt-get install libncurses5`
	ⓘ For Ubuntu20 you can install `libncurses5` or `libncurses6`.
	- If you choose `libncurses5`:
	`sudo apt-get install libncurses5`
	- If you choose `libncurses6`:
	`sudo apt-get install libncurses6`
	Note: For `libncurses6` you need to create symlink for `libncurses5` pointing to `libncurses6`.
	`sudo ln -s /usr/lib/x86_64-linux-gnu/libncurses.so.6.2 /usr/lib/x86_64-linux-gnu/libncurses.so.5`
	`sudo ln -s /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2 /usr/lib/x86_64-linux-gnu/libtinfo.so.5`

Debian	Use a package manager such as APT to install the library (as described for the Ubuntu instructions above).

SLES12 and SLES15	Use `zypper` to install the library, such as:
	`sudo zypper install libxml2-2`
	`sudo zypper install libxml2-tools`
	`sudo zypper install libaio1`
	`sudo zypper install numactl`
	`sudo zypper install libcurses5`
	`sudo zypper install tzdata`

tzdata Requirement

Ubuntu version 16 and higher requires the `tzdata` package in order to install the Enterprise Console and Controller.

> ⓘ The `tzdata` package is also required by the MySQL connector.

To install `tzdata`, use `apt-get`, such as:

- `sudo apt-get install tzdata`

Configure User Limits in Linux

AppDynamics requires the following hard and soft per-user limits in Linux:

- Open file descriptor limit (`nofile`): 65535
- Process limit (`nproc`): 8192

The following log warnings may indicate insufficient limits:

- Warning in database log: "Could not increase number of max_open_files to more than xxxx".
- Warning in server log: "Cannot allocate more connections".

To check your existing settings, as the root user, enter the following commands:

```
ulimit -S -n
ulimit -S -u
```

The output indicates the soft limits for the open file descriptor and soft limits for processes, respectively. If the values are lower than recommended, you need to modify them.

Where you configure the settings depends upon your Linux distribution:

- If your system has a `/etc/security/limits.d` directory, add the settings as the content of a new, appropriately named file under the directory.
- If it does not have a `/etc/security/limits.d` directory, add the settings to `/etc/security/limits.conf`.
- If your system does not have a `/etc/security/limits.conf` file, it is possible to put the `ulimit` command in `/etc/profile`. However, check the documentation for your Linux distribution for the recommendations specific for your system.

To configure the limits:

1. Determine whether you have a `/etc/security/limits.d` directory on your system, and take one of the following steps depending on the result:
    - If you *do not* have a `/etc/security/limits.d` directory:
        a. As the root user, open the `limits.conf` file for editing:

        ```
        /etc/security/limits.conf
        ```

        b. Set the open file descriptor limit by adding the following lines, replacing `<login_user>` with the operating system username under which the Controller runs:

        ```
        <login_user> hard nofile 65535
        <login_user> soft nofile 65535
        <login_user> hard nproc 8192
        <login_user> soft nproc 8192
        ```

    - If you *do* have a `/etc/security/limits.d` directory:
        a. As the root user, create a new file in the `limits.d` directory. Give the file a descriptive name, such as the following:

        ```
        /etc/security/limits.d/appdynamics.conf
        ```

        b. In the file, add the configuration setting for the limits, replacing `<login_user>` with the operating system username under which the Controller runs:

        ```
        <login_user> hard nofile 65535
        <login_user> soft nofile 65535
        <login_user> hard nproc 8192
        <login_user> soft nproc 8192
        ```

2. Enable the file descriptor and process limits as follows:

    > ⓘ This step is not required for RHEL/CentOS version 5 and later. The below file has been combined into `/etc/pam.d/system-auth`, and already contains the required line.

    a. Open the following file for editing:

    ```
    /etc/pam.d/common-session
    ```

    b. Add the line:

    ```
    session required pam_limits.so
    ```

3. Save your changes to the file.

When you log in again as the user identified by `login_user`, the limits will take effect.

Fonts Needed for the Reporting Service

The Reporting Service depends upon certain system libraries and resources that are usually included in standard Linux distributions. However, certain lightweight flavors of Linux may be lacking the requirements, primarily font libraries. The Reporting Service requires Fontconfig and FreeType installed as well as at least one sans-serif font. Errors in the reporting server log will indicate missing components, such as a missing `libfontconfig.so` file.

The following table lists one operating systems and the commands to install the required libraries:

Operating System	Command

CentOS 6.1, 6.2; CentOS 6.3, 6.4, and 6.5, Fedora 14	`$ yum install fontconfig freetype urw-base35-fonts`
	`$ yum groupinstall hebrew-support`
	`$ yum langinstall he_IL`

CentOS 7.x, Redhat 7.x	`$ yum install fontconfig`
	`$ yum groupinstall Fonts # Only needed for Chinese/Japanese`

Ubuntu 8, 12, 14	`$ sudo apt-get update`
	`$ sudo apt-get install libfreetype6 libfreetype6-dev libfontconfig`
	`$ sudo apt-get install language-support-he language-pack-he`
	`$ sudo apt-get install culmus culmus-fancy xfonts-efont-unicode xfonts-efont-unicode-ib xfonts-intl-european msttcorefonts`

Ubuntu 13	`$ sudo apt-get install libfontconfig`
	`$ sudo apt-get install language-support-he language-pack-he`
	`$ sudo apt-get install culmus culmus-fancy xfonts-efont-unicode xfonts-efont-unicode-ib xfonts-intl-european msttcorefonts`

See Administer the Reporting Service for information on configuring the service.

GNU C Libraries

The Reporting Service requires GLIBCXX_3.4.9 or later and GLIBC_2.7 or later to run.

For more information and download instructions, see https://www.gnu.org/software/libc/.

Prepare Windows for the Controller

This page provides operational and setup guidelines for running the Controller on Windows.

User Account Requirements

The user account you use to install the Controller must have administrative privileges on the host Windows machine.

> ⓘ If the host Windows machine is the Enterprise Console host, the Windows user account is configured to run with administrative privileges by default.

Virus Scanners

Configure virus scanners on the target machine to ignore the AppDynamics Enterprise Console directory and database directory (or simply the entire Controller directory). Code is never executed from the data directory, so it is generally safe to exclude this directory from virus scanning. The default location of the data directory is `<controller_home>\db\data`.

Also configure virus scanners to trust the Controller launcher, database executable, reporting service launcher, and events service (analytics processor) launchers. The launcher names are:

- Controller launcher: `AppDynamicsDomain1Service.exe`
- MySQL executable: `mysqld.exe`
- Events service launcher: `analytics-processor.exe`
- Reporting service launcher: `appdynamicsreportingservice.exe`

Windows Defender Scanning

Exclude the Controller data directory (`<controller_home>\db\data`), or simply the entire Controller directory, from scanning by Windows Defender. If you are not sure whether Windows Defender is running on the system, check for it in your local Services list. You can either configure the Controller data directory to be excluded in the Windows Defender Control Panel, or disable the service altogether if it is not needed.

For details on how to view services and exclude directories in Windows Defender, refer to the documentation for your version of Windows.

Windows Indexing Service

Ensure that the Windows indexing service is configured to ignore the Controller data directory (`<controller_home>\db\data`), or simply the entire Controller directory.

The data directory does not contain any files that are meaningful to the indexer, so it can be excluded from indexing. To exclude the directory from indexing, you can add the directory to the excluded directories list in the **Indexer Control Panel**, disable indexing in directory preferences, or stop the indexing service entirely.

To add the directory to the excluded directory list, follow these steps:

1. From the **Control Panel Indexing Options** dialog, click the **Modify** button.
2. In the **Indexed Locations** dialog, navigate to and select the **Controller** data directory.
3. Clear the checkbox for the data directory and click **OK**.

Windows Update

Configure the Windows Update preferences so that the server is not automatically restarted after an update. To configure the restart policy:

1. Open the **Local Group Policy Editor** dialog (search for and run the `gpedit` executable).
2. Navigate to the Windows Update component. In the tree, you can find it under **Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components**.

3. Double-click on the **No auto-restart...** setting.
4. Select the **Enabled** option and click **Apply**.

.NET Framework

Components of the .NET Framework 3.5 are required to allow the Controller to be installed as a Windows service on the target machine. The installer checks your system and indicates if .NET 3.5 is not found. Follow the instructions on the Enterprise Console to get the required components.

> ⓘ Even if you have the latest version of .NET installed, you still have to install .NET 3.5. This is due to a Glassfish requirement where the Glassfish launcher explicitly requires .NET 3.5.

Windows 7

The Controller is automatically installed as a Windows service. Windows 7 operating system must have the hotfix described in http://support.microsoft.com/kb/2549760. This hotfix ensures that the Windows registry modifications made by the installer to extend the default service timeouts work as expected. The installer checks for the presence of the hotfix and warns you if it is not found.

Controller Data and Backups

This page provides an overview of how to configure and administer Controller data storage.

About Controller Data Storage

The Controller requires persistent data storage to store the following type of information:

- Design of your applications (all metadata about business transactions, tiers, policies, and so on)
- History of the performance of your applications (metric data)
- Transaction snapshot data and events
- History of incidents that occurred (both resolved and unresolved incidents)

Controller Data Directory Location

By default, the AppDynamics Controller uses MySQL as its storage mechanism. The Controller bundles a MySQL instance with the Controller. At installation time, the Enterprise Console creates the necessary tables and artifacts in the database.

By default, the database files and data are stored in: `<controller_home>/db`.

Manage the Database User Password

The Enterprise Console creates the user account that the Controller uses to log into the database to perform database-related operations. The username of the account is "root", and the password is the one you supply to the Enterprise Console during the Controller installation process.

When attempting to access the data, the Controller reads the database user password from these sources and in the priority shown:

- From the `MYSQL_ROOT_PASSWD` environment variable
- From user input to a command line prompt

If you do not keep the password in the environment variable, you will need to supply it in response to a command-line prompt whenever performing an operation that involves accessing the database, starting the database, stopping the database, or logging into the database.

Moving the Controller Data Directory

After installation, you can move the data directory to a new location. This may be necessary, for example, if there is not enough disk space available during Controller installation.

If you are using symlinks, you must create the symlink outside of the root Controller install directory and move the data directory to the new volume after you install the Controller.

Warning: Do not mount a file system on `<controller_home>/db/data`. During Controller upgrade, the Enterprise Console moves the data directory to `data_orig`. Upgrade will fail if the Enterprise Console cannot complete this move.

You can also update the `datadir` path on the Controller Database Configurations page of the Enterprise Console GUI.

To relocate the Controller data directory

1. Stop the Controller and its database. See Start or Stop the Controller.
2. Modify following properties in the `<controller_home>/db/db.cnf` file to point to the new location of the data directory.

```
datadir
tmpdir
log
slow_query_log_file
```

3. Copy (or move) the existing data directory `<controller_home>/db` to the new location.
   For example, to copy the data on Linux:

```
cd <controller_home>/db/
cp data <new-location>
```

4. Start the Controller. See Start or Stop the Controller.
5. Check the `database.log` and `server.log` for any errors related to the database connection.

Controller Data Backup and Restore

Related pages:

- Enterprise Console Back Up and Restore

AppDynamics strongly recommends that you perform routine data backups of the Controller.

One method of maintaining backups of the Controller is to implement high availability. With high availability, the database on the secondary Controller keeps a replicated copy of the data on the primary Controller. A secondary Controller also makes it practical to take cold copies of the Controller data, since you can shut down the secondary to copy its data without affecting Controller availability. For information on HA, see Controller High Availability (HA).

Other approaches include using a disk snapshot mechanism or using database backup tools. The BackupTools section describes tools that support each approach. In addition to regular backups, back up the Controller and Enterprise Console before upgrading or migrating them from one server to another.

This page provides an overview of the tasks and considerations related to backing up the Controller. Note that your Controller should be shut down before performing any import functions.

> It is to be noted that controller versions 4.3 and later will work only on restoring and backing up the `<Controller Home>/.appd.scskeystore` file.

Best Practices for Backups

To perform a complete backup of the Controller, the following three directories must be backed up:

1. Controller install directory
2. MySQL `datadir`
3. JRE directory used for Glassfish

Backing up the entire system each night may not be feasible when dealing with the large amount of data typically generated by a Controller deployment. To balance the risk of data loss against the costs of performing backups, a typical backup strategy calls for backing up the system at different scopes at different times. That is, you may choose to perform partial backups more frequently and full backups less frequently.

The scope of a Controller backup can be categorized into these levels:

- Level 1: A light backup of the installation environment only
- Level 2: A metadata backup involving all metadata associated with the installation except for big data tables.
- Level 3: Backs up all data, either by performing a cold backup of the /data directory or a hot backup using a third-party tool.

A possible backup strategy may be to perform a level 1 and level 2 backup very frequently, say nightly, and a level 1 and level 3 backup about once a week. In addition to performing a level 1 or 2 backup, you should also back up the data for the Enterprise Console with `mysqldump` on a regular basis. A level 3 backup also backs up the Enterprise Console data. See Enterprise Console Back Up and Restore for more information.

Light Backup (Level 1)

A light backup targets Controller configuration files like `db.cnf` and `domain.xml`. This type of backup lets you avoid having to reconfigure the Controller in case of machine failure.

To perform this type of backup, simply copy everything in the Controller installation directory EXCEPT the data directory.

While it is recommended that you copy the entire Controller home except for the data directory when performing a light backup, particularly before performing a Controller upgrade, there are scenarios in which you may wish to copy only site-specific configuration files. This may be the case if you are migrating an existing Controller configuration to a new Controller installation, for example. For a list of those files, see Migrate the Controller.

Metadata Backup (Level 2)

A metadata backup exports the data that encapsulates the environment monitored by the Controller. Metadata defines the applications monitored by the Controller, business transactions, policies, and so on. It does not include what can be thought of as "runtime data", the big data tables that contain the metrics, snapshots, events, and top summary stats (top SQL, top URLs, and so on) generated in the monitored environment. By backing up metadata, you can avoid having to reconfigure monitored applications in the Controller in the event of a failure.

To perform this type of backup:

1. Run the script described in Using mysqldump to back up the Controller.
2. Then augment it with a copy of:

   a. The Controller Java keystore (600 byte file): `<controller install>/.appd.scskeystore`
   b. And the Enterprise Console keystore: `platform-admin/.appd.scs`

Complete Backup (Level 3)

A complete backup saves all runtime data associated with the Controller installation. It captures the actual metrics data, snapshots, and so on.

Some third-party backup tools, such as Percona XtraBackup, do not rely on transactions so you can perform a hot backup of your system (that is, back up the Controller database while it is running).

You can perform a complete backup as either:

1. A cold backup of all three directories (Controller install directory, MySQL `datadir`, and JRE directory). To perform a cold backup, shut down the Controller app server and database. Then, create an extra copy of the three directories using the `cp -r` command, the `tar` utility, `rsync`, or others.
2. A hot backup, which means the Controller is running.
   a. If you have a high availability setup for the Controller, you can shut down the database on the secondary Controller. Then, you can perform a cold backup on the secondary Controller and restart the database.
   b. If you do not have a high availability setup for the Controller, use a third-party tool such as Percona XtraBackup to back up the MySQL `datadir`. Then, use the `cp -r` command, the `tar` utility, `rsync`, or others to back up the Controller install directory and the JRE directory.

   > ⚠ Percona XtraBackup can fail to hot backup Controllers that are too busy. To avoid this error,
   >
   > - Back up the secondary Controller instead, if it exists.
   > - Increase the MySQL log sizes.
   > - Perform the hot backup when the Controller is less busy.

Backup Tools

This section lists a few third-party tools that you can use to back up Controller data. The list is not exhaustive; you can use any tool capable of backing up MySQL data with the Controller. It is up to you to test your backup and restore process. However, the tool you decide on should back up the data as binary data.

For Linux systems:

- Percona XtraBackup

For Windows systems:

- Zmanda Recovery Manager for MySQL

An alternative to using a database backup tool is to use a disk snapshot tool to replicate the disk or partition on which the Controller data resides. Options include:

- ZFS volume manager. For more information, see Using ZFS methods for data backup.

Details for performing this type of backup are beyond the scope of this documentation. For more information, refer to administration documentation applicable to your specific operating system.

Back Up the Controller with mysqldump

The `mysqldump` utility is a MySQL backup tool that is included with the Controller instance of MySQL.

While `mysqldump` is not recommended for use on large data tables, such as the Controller metric data tables, it is useful for backing up Controller metadata. Metadata defines the monitored domain for the Controller, including applications, business transactions, alert configurations, and so on.

The following instructions assume that the binary path for the Controller's MySQL instance is in the PATH variable. The path to the Controller's instance of MySQL must precede any other MySQL path on your system. This prevents conflicts with other database management systems on your machine, such as a MySQL instance included by default with Linux.

The database binary files for the Controller database are in `<controller_home>/db/bin`.

> ⚠ Before using `mysqldump`, first ensure that the Controller app server is stopped. If you attempt to run `mysqldump` while the app server is running, it will severely degrade the performance and stability of the Controller.

To use `mysqldump`, run the `mysqldump` executable, passing the root username, password, and output file. The executable is located in the following directory:

`<controller_home>/db/bin`

The command should be in the form:

```
mysqldump -u root -p<password> <ignore-table_statements> > /tmp/metadata_dump.sql
```

For a full example that shows which tables to exclude for a metadata backup, see the contents of the metadata backup script described in the next section.

Sample mysqldump Script

The following script illustrates how to use `mysqldump` to export Controller metadata while excluding runtime data tables by script.

- Linux: ControllerMetadataBackup.sh.txt

> Backing up the Controller with a custom MySQL data directory location using mysqldump will result in an incomplete and unusable metadata dump. The default location for the Controller's MySQL data directory is `appd_install_dir/db/data`. If you do not see the data directory here then that means an alternate directory or mount point was selected and configured for your MySQL data directory.
>
> To fix this issue:
>
> 1. Examine `<Controller Home>/db/db.cnf`
>   a. Locate the `datadir` parameter. It contains the path to the MySQL data directory on your Controller host.
> 2. Edit the metadata backup script
>   a. Replace `$appd_install_dir/db/data` with the `datadir` path you located in `db.cnf`

To use the script:

1. Download the version appropriate for your operating system.
2. Rename the file to remove the .txt extension.
3. Modify the contents of the file as described in the script comments.

Import Controller Data with mysql

When you restore or migrate the Controller, you can import the data you exported with `mysqldump`.

Shut down your Controller before using the following command to import the data into a database:

```
$install/db/bin/mysql -u controller -p<ControllerDBpassword> < metadata_dump.sql
```

It will overwrite the tables. You can clone your installation to another host and test your restoration there.

> A Controller that is installed with a custom MySQL data directory location requires additional flags.

Sample Data Backup Script

The following script uses Percona XtraBackup to back up Controller data. To use it, you need the `percona-xtrabackup` or `xtrabackup` and qpress packages. For information on installing XtraBackup, see the Percona installation documentation.

To use the script, download the following file:

- appdynamics-backup.sh.txt

Rename the script (by removing the .txt extension). In the script:

- Verify or edit the values of the `CONTROLLER_HOME` and `DESTINATION` variables at the beginning of the script for your environment.
- Edit the if/then/else clause at the end of the script if you want to implement backup file rotation, call your enterprise backup system to pick up the compressed Controller database image, or send an alert if the backup fails for any reason.

The following commands demonstrate how to restore a compressed backup image:

```
mkdir /path/to/big/staging/folder

# unpack the compressed backup archive
cd /path/to/big/staging/folder && xbstream -xv < /path/to/backups/dir/controller-yyyymmdd.xbstream

# decompress the backup image and apply the log taken during backup
CONTROLLER_HOME=/path/to/AppDynamics/Controller && cd /path/to/big/staging/folder \
&& innobackupex --decompress --parallel=16 . && innobackupex \
--defaults-file=$CONTROLLER_HOME/db/db.cnf --use-memory=1GB --apply-log --parallel=16 .

# Move a prepared backup into an empty controller data directory
CONTROLLER_HOME=/path/to/AppDynamics/Controller && cd /path/to/big/staging/folder \
&& innobackupex --defaults-file=$CONTROLLER_HOME/db/db.cnf --move-back .
```

For more information on these options, see the Percona innobackupex option reference.

Using a Backup to Migrate to a New Physical Server

You can use either a hot or cold backup procedure to migrate Controller data to a new server. However, we recommend performing cold backups. While a hot backup does not bring down the Controller for an extended amount of time, it does introduce the possibility of data loss, since hot backups capture the state of the data only when the hot backup starts.

To perform a cold backup, simply shut down the Controller and back up the data directory located in `<controller_home>/db`.

Controller Disk Space and the Database

This page discusses best practices for managing disk space for the MySQL database used by the Controller.

Disk Space Considerations

To ensure database integrity, the Controller automatically shuts itself down when available disk space falls below 1 GB.

Before it reaches that point, the Controller displays a low disk space alert in the UI and writes an error level event to server.log. The point at which the Controller generates the alert depends on its profile, as follows:

- For large and extra large profiles: 10 GB or less
- For all other profiles: 2 GB or less

The Controller shuts itself down when there is less than 1 GB on the disk regardless of the Controller profile type.

> ⚠ It's important to note that the Controller monitors the disk or partition that it is installed on. If the Controller data resides on a different disk or partition from the Controller home directory, you will need to monitor available space on that disk or partition separately.

Managing Disk Space

If the disk space is low, you need to reduce the size of the Controller database.

To manage how much disk space the Controller database uses, you can change the amount of data retained in the Controller database. See Database Size and Data Retention.

Database Size and Data Retention

This page provides both the on-premises and SaaS default data retention periods for data stored by the Controller and instructions for modifying on-premises values. AppDynamics manages SaaS Controller deployments, which eliminates the need for manual modifications.

Migrate the Controller

Related pages:

- Controller High Availability

This page describes how to migrate a Controller from a physical or virtual machine (VM) to a new physical machine.

Before Starting

Migrating the Controller often results from the need to move the Controller to new hardware due to increased load. Before starting, make sure that the new hardware meets the AppDynamics requirements as described in Controller System Requirements. Specifically, you should review the Controller hardware performance profiles and the hardware requirements per profile information to verify that the target Controller hardware meets the RAM size and Disk I/O requirements.

⚠ You will need to update the MAC address associated with your license since licenses are tied to the machine MAC addresses. You can also acquire new license files for the new Controller hardware. Send the MAC addresses to salesops@appdynamics.com and request a new license file or two new licenses, if upgrading to an HA pair. See Apply or Update a License File for more information.

If you are performing a migration and upgrade for a 4.3 version Controller, you should first migrate the Controller. Then, you can upgrade the Controller to 4.5 or higher by installing the Enterprise Console and using the Discover & Upgrade feature. This also applies to migrations involving different OS environments.

ⓘ VMotioning, or migrating a VMware guest with a running Controller inside it from one host to another, is not supported. Doing so will lead to dropped metrics and UI performance problems.

Migrating a Linux Controller

You can use the high availability features provided by the Enterprise Console to migrate a Controller from one machine to another. It is assumed that the Controller you need to migrate is already managed by the Enterprise Console (it has been installed or discovered by the Enterprise Console). See Enterprise Console for more information.

You add the new host as an HA pair to the old host, set the new host as active, and then remove (decommission) the old host. When finished, the Controller will run on the new host.

Before starting, you should review the requirements and concepts related to Controller High Availability.

To migrate a Linux Controller:

1. Log in to the **Enterprise Console** UI interface.
2. Select the Platform containing the host you want to migrate.
3. In the **Hosts** page, add the new host (the one to serve as the new target host) and provide the credentials for connecting to that host.
4. In the **Controller** page, select **Add Secondary** and select the new target host. Provide the DB root password and Controller root password, and select **Submit**.
5. In the **Controller** page, select **HA failover**.
   The Primary Controller is now running on the new host.
6. Update the license MAC address or apply a new license for the new machine. See Before Starting for more information.
7. Decommission the old Controller from the **Controller** page:
   a. Select **Remove Controller**, or run the following command on the Enterprise Console host:

   ```
   platform-admin.sh submit-job --job remove --service controller --args removeBinaries=true
   ```

   b. Select the **remove binaries** option. (Do not select **Remove entire cluster**.)
   The Controller is now running on the newly provisioned host.

You can keep the same access key from the old Controller. To migrate or update your access key, see Controller Secure Credential Store.

ⓘ You must update the license rule access keys.

Migrating a Windows Controller

Since high availability features are not available on Windows, you must use an alternative procedure to migrate a Controller from one machine to another. You use the Enterprise Console to manually install and move the `.appd.scskeystore` and the `datadir` from the old host to the new host. Once completed, the Controller will run on the new host.

Before starting, you should review the requirements and concepts related to Controller High Availability.

To migrate a Windows Controller:

1. Install the Enterprise Console on the new Controller host from where you are running the existing Controller host.
2. Use the Enterprise Console to install the same version of the Controller on your new Controller host using the same passwords as on the existing Controller host.
3. Shut down the Controller Appserver and database on both Controller hosts.
4. Copy `<controller_home>/.appd.scskeystore` and the Controller's MySQL `datadir` from the old host to the correct locations on the new host.

   > ⓘ When migrating the data, ensure that the destination MySQL version is the same as the source version.

5. Start the Controller Appserver and database on the new host.
   The Controller is now running on the newly provisioned host.

You can keep the same access key from the old Controller. To migrate or update your access key, see Controller Secure Credential Store.

> ⓘ You must update the license rule access keys.

Prepare Virtual Machines for the Controller

The following are considerations and requirements for the machine hosting the Controller.

Controller Sizing Restrictions

Demo, small, and medium profile Controllers can run on virtual machines that meet the performance requirements otherwise specified. Large deployments are not supported except as indicated in Controller System Requirements.

> ⚠ The Controller is not supported on virtual machines with oversubscribed physical CPUs like T2 AWS instances. These Burstable Performance Instances are CPU-throttled and do not have dedicated storage bandwidth. We recommend you use a Fixed Performance Instance type instead.

Fully Reserved RAM

The memory allocation for the Controller's virtual machine must be fully reserved RAM. Reserve as much as possible of the total memory allocation.

On VMWare VMs

Reserved Memory Configuration

For information on how to configure reserved memory on VMWare, see Set Memory Reservation on a Virtual Machine.

Trend Micro Configuration

On vSphere platforms shipped with Trend Micro, the Trend Micro process needs to be disabled in order for replication jobs to work. You can also add the entire Enterprise Console and Controller directories to the exclusion list to avoid this conflict.

See Trend Micro and VMware Virtualization for more information.

On Hyper-V VMs

On Microsoft Hyper-V, "Dynamic Memory" needs to be disabled and "Static Memory" needs to be enabled.

To disable a Hyper-V VM from using Dynamic Memory:

1. Open the Hyper-V Manager.
2. Select the VM you want to configure in the **Virtual Machines** pane, making sure the VM is powered off.

   > ⓘ You cannot enable or disable Dynamic Memory if the VM is in either the Running or Saved state.

3. Right-click the VM to bring up the context menu.
   a. Select **Settings**.
   b. Click the **Memory** page.
4. Uncheck the **Enable Dynamic Memory** box.

See Virtualization: Optimizing Hyper-V Memory Usage for more information.

Licenses on VMs

The Controller license is bound to the MAC address of the host machine. To run the Controller on a virtual machine, you must ensure that the host virtual machine uses a fixed MAC address.

I/O Performance Requirements

A factor that often limits the performance of the Controller is the underlying disk I/O performance of the host machine.

Virtual machines, in particular, are less apt to provide sufficient I/O performance requirements for the Controller, compared to similarly specified physical machines. In any case, whether you are deploying the Controller to a physical or virtual machine, you need to ensure that the machine meets the I/O performance requirements set forth in Controller System Requirements.

Host Name Entry

The fully qualified hostname for the application server is the address at which Controller UI users and application agents will use to access the Controller. You specify this hostname when you install the Controller. The hostname needs to be in the /etc/hosts file on the machine.

The following example shows an entry in `/etc/hosts` with the IP 21.43.65.987, the fully qualified hostname `application1.mycompany.com` and the alias `app1`:

```
21.43.65.987 application1.mycompany.com app1
```

Elastic Network Interface (ENI)

For AWS, provision an ENI for each Controller host and link the license to the ENI. For more information about ENI, see the AWS documentation at the following link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html.

AWS Controller Deployment Guide

The AppDynamics Controller is certified to run on an AWS environment with the Aurora Database. This page provides information on installing, configuring, and administering a Controller deployment on AWS with Aurora Database.

Installation Overview

You can deploy a medium- or large-scale Controller in AWS using Aurora. Aurora provides higher performance than MySQL, which allows you to scale your Controller to handle more metrics.

Before you install the Controller, review the requirements for the components you plan to install and prepare the host machines. The requirements vary based on the components you deploy and the size of your deployment.

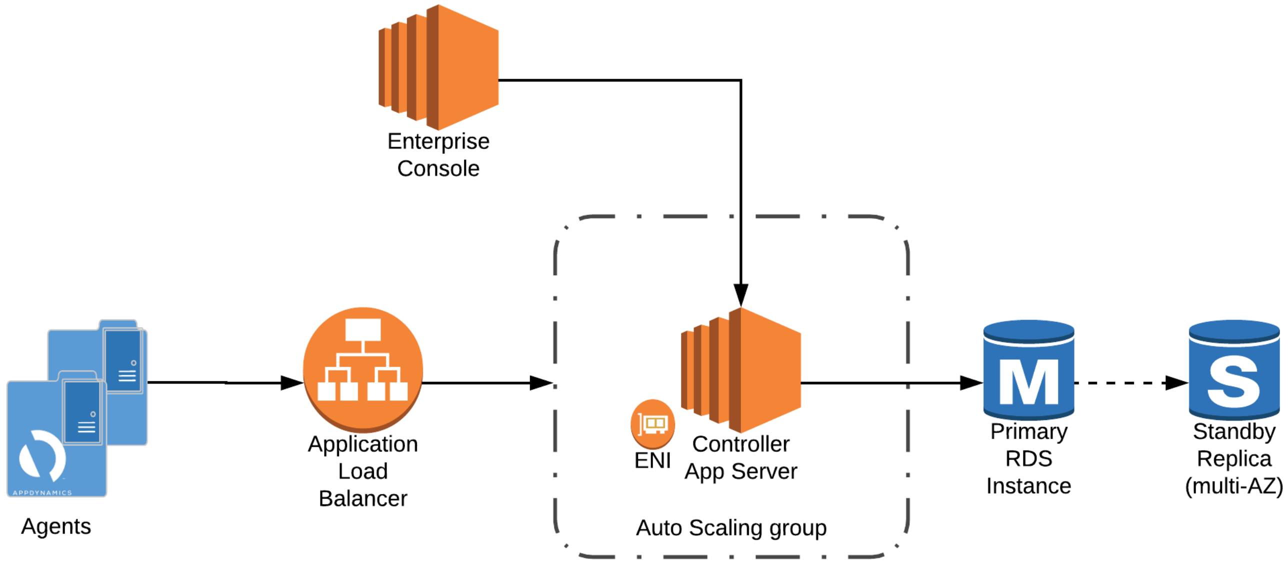
You can manually deploy your Controller on AWS. See Deploy the Controller on AWS
This deployment requires attention and time because you have to set up all of the configurations. However, this gives you freedom to customize your deployment.

You use the Enterprise Console to deploy the Controller by specifying Aurora as the database type. The Enterprise Console is the installer for the Controller, and you can use it to manage the entire lifecycle of new or existing AppDynamics Platforms and components.

You can get the software for installing the platform components from the AppDynamics download site. See Download AppDynamics Software

AppDynamics on AWS Architecture

The following diagram depicts the components of an AppDynamics Controller deployment on AWS.



Configure an Application Load Balancer in front of the Controller and ensure SSL terminates at the Elastic Load Balancer.

With Amazon Relational Database Service (RDS), you no longer need to worry about database backups, as it takes care of this for you. Also, you no longer need to implement high availability (HA) on your own, since you can instead leverage the Standby Replica that Aurora/RDS offers and the Aurora database is horizontally scalable. With the multi-AZ deployment option, Aurora offers 99.95% availability.

Controller High Availability and AWS

- If your data is migrated to the Aurora DB, you can create a new Controller from the Amazon Machine Image (AMI) in case of failure. With Aurora as the database, HA scenarios are not required because the Aurora database is horizontally scalable.
- It is a good practice to cut the new root AMI every time you make a configuration change to the Controller.
- You can configure a read replica instance while creating Aurora DB, which should satisfy most database replication requirements.

- You can configure the backup policy while creating the Aurora DB and modify it later.

Prepare the AWS Machine for the Controller

Related pages:

- Prepare Linux for the Controller
- Prepare Windows for the Controller

This page provides considerations and requirements for AWS instances that host the Controller.

Instance Sizing

For AWS instance sizing by metric ingestion rate, see the Controller Sizing table.

The actual metrics generated can vary greatly depending on the nature of the application and the AppDynamics configuration. Be sure to validate your sizing against the metric ingestion rate before deploying to production.

Elastic Network Interface (ENI)

For AWS, provision an ENI for each Controller host and link the license to the MAC address of the ENI. For more information about ENI, see the AWS documentation at the following link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html.

Deploy the Controller on AWS

This page describes the procedure used to deploy the AppDynamics Controller to an AWS environment.

You can set custom configurations when you manually deploy the Controller to AWS. From these steps, you manually set up security groups, database parameter groups, Amazon Relational Database Service (RDS), Aurora DB instance, EC2 instance, Elastic Network Interface (ENI) for the Controller, DNS CNAMEs, and listeners for your load balancer.

Afterward, you install the Enterprise Console, and then use the Enterprise Console to install the Controller and configure it for the AWS environment.

Based on the Amazon Machine Image (AMI), you can provision a replacement EC2 instance for the Controller as needed.

For help with your deployment, contact your AppDynamics account, sales, or professional services representative.

Before You Begin

- Review if Amazon Aurora DB is available in your region – Check the AWS Region Table in the AWS documentation to see if the Amazon Aurora - MySQL-compatible service is available in your particular region.
- Amazon RDS Password Requirements – There are some naming constraints in the Amazon Relational Database Service (RDS). The master password for Aurora DB can be any printable ASCII character except "/", """, or "@", and must contain 8 to 41 characters. Master password constraints differ for each database engine.
  For details on naming constraints in Amazon RDS, see the AWS documentation.

Deploy the Controller on AWS

To manually deploy the Controller on AWS:

1. Create Security Groups
2. Create Custom DB Parameter Groups
3. Launch an Amazon RDS Aurora DB Instance
4. Create Database User for Controller
5. Launch an EC2 Instance for the Controller
6. Create the ENI for the Controller
7. Create DNS CNAMEs
8. Install the Enterprise Console in an AWS Environment
9. Install the Controller in AWS Using Aurora
10. Apply Controller Optimizations
11. Configure a Load Balancer
12. Configure Listener Rules

Troubleshooting the Installation

Issue: Controller EC2 Instance Stops

If the Controller EC2 instance stops almost immediately after you install the Controller, there may be an issue with your EBS devices. AWS may report that it is not able to boot from the volumes. If the EC2 machine stops, check and update your EC2 volumes so that they are mounted correctly.

Create Security Groups

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic.

At a minimum, we recommend creating the following security groups when deploying AppDynamics in AWS using Aurora DB.

> You can create additional security groups to align with your organization's standards.

Required Security Groups

Use the instructions provided in the AWS security groups documentation to create these required security groups:

Security Group for the AppDynamics Enterprise Console

Security group name: `appd-ec-security-group`

Inbound rule: Allow all inbound TCP traffic on ports 22 and 9191

Outbound rules:

- Allow outbound TCP traffic to `appd-appserver-security-group` on port 22
- Allow outbound TCP traffic to `appd-db-security-group` on port 3388

Security Group for the AppDynamics Controller Appserver

Security group name: `appd-appserver-security-group`

Inbound rules:

- Allow all inbound TCP traffic on port 22
- Allow inbound TCP traffic on ports 8090-8097 from `appd-elb-security-group`

Outbound rule: Allow outbound TCP traffic to `appd-db-security-group` on port 3388

Security Group for AppDynamics Database Instances

Security group name: `appd-db-security-group`

Inbound rule: Allow inbound traffic on port 3388 from `appd-appserver-security-group` and `appd-ec-security-group`

Outbound rule: No outbound access allowed

Security Group for Load Balancer in Front of the AppDynamics Controller

Security group name: `appd-elb-security-group`

Inbound rule: Allow all inbound HTTPS traffic on port 443

Outbound rule: Allow outbound TCP traffic to `appd-appserver-security-group` on ports 8090-8097

Create Custom DB Parameter Groups

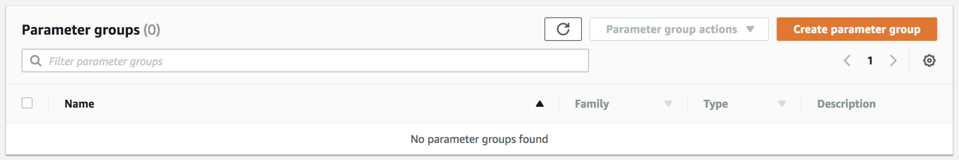
You must modify some of the parameters for your database instance.

> Any parameters not modified retain their default values.

Parameter	Recommended Value
`innodb_file_format`	Barracuda
`innodb_large_prefix`	1 (0 for Aurora 5.6)
`innodb_lock_wait_timeout`	180
`innodb_max_dirty_pages_pct`	20
`lock_wait_timeout`	180
`log_bin_trust_function_creators`	1
`max_allowed_packet`	104857600
`max_heap_table_size`	1610612736
`query_cache_type`	0
`sql_mode`	0
`tmp_table_size`	67108864
`wait_timeout`	31536000

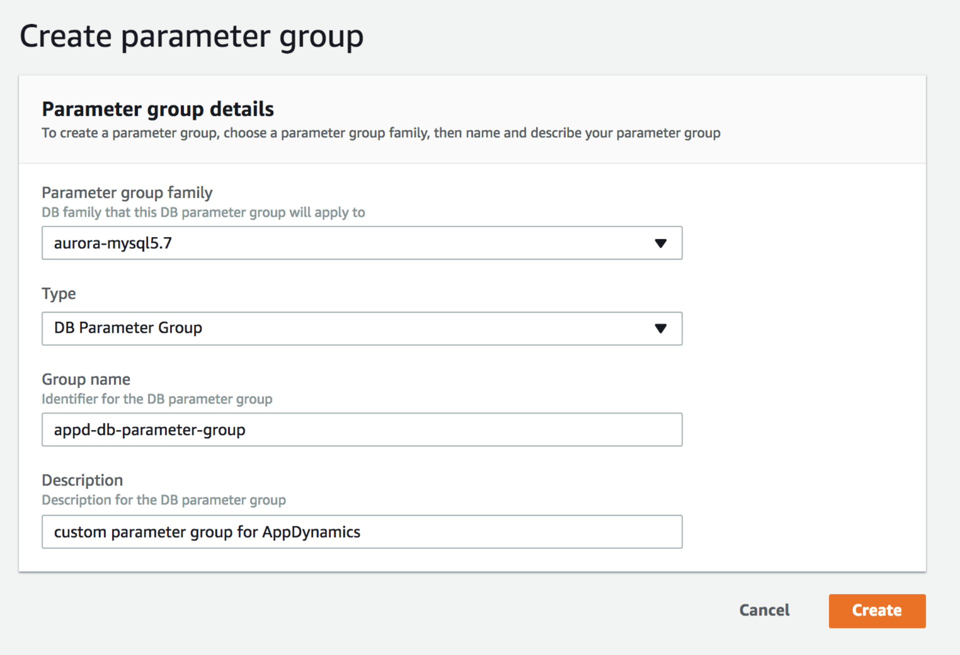
To create custom DB parameter groups:

1. Navigate to the Parameter groups page on the Amazon RDS in the AWS console.
2. Click **Create parameter group** on the top right of the page.



No parameter groups found

3. Enter the group details:



Create parameter group

Parameter group details

To create a parameter group, choose a parameter group family, then name and describe your parameter group

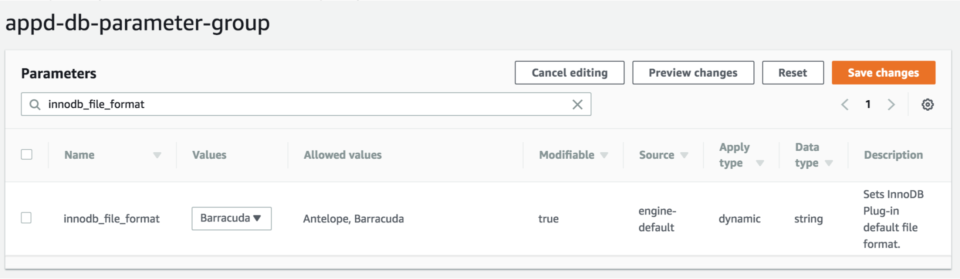
Parameter group family
DB family that this DB parameter group will apply to
aurora-mysql5.7 ▼

Type
DB Parameter Group ▼

Group name
Identifier for the DB parameter group
appd-db-parameter-group

Description
Description for the DB parameter group
custom parameter group for AppDynamics

4. Modify the parameter values in this group:



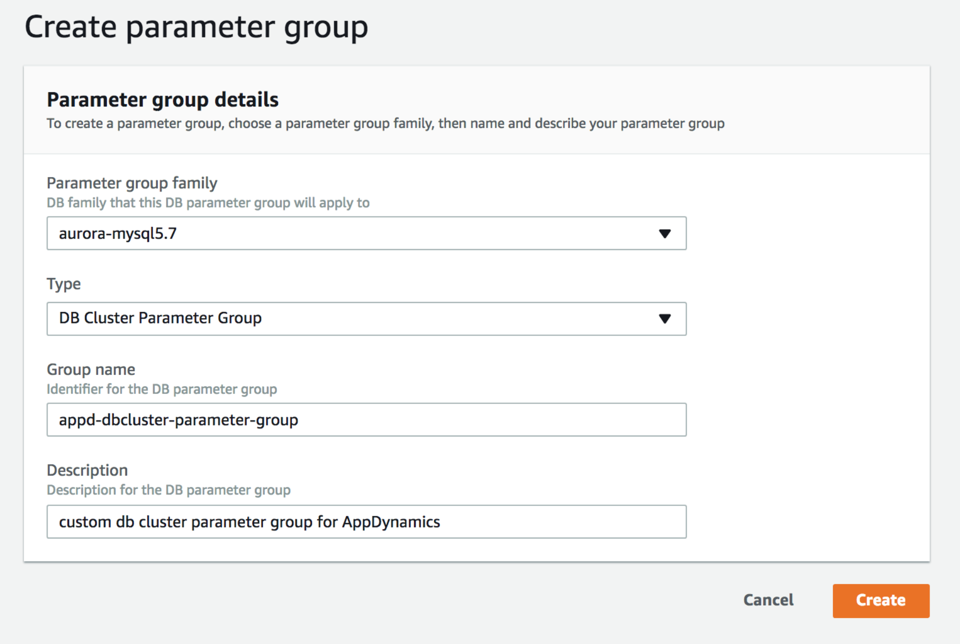
appd-db-parameter-group

Parameters

innodb_file_format

□	Name	Values	Allowed values	Modifiable	Source	Apply type	Data type	Description
□	innodb_file_format	Barracuda ▼	Antelope, Barracuda	true	engine-default	dynamic	string	Sets InnoDB Plug-in default file format.

5. Create a custom DB cluster parameter group:



Create parameter group

Parameter group details

To create a parameter group, choose a parameter group family, then name and describe your parameter group

Parameter group family
DB family that this DB parameter group will apply to
aurora-mysql5.7 ▼

Type
DB Cluster Parameter Group ▼

Group name
Identifier for the DB parameter group
appd-dbcluster-parameter-group

Description
Description for the DB parameter group
custom db cluster parameter group for AppDynamics

6. Modify the parameter values in this group as follows:

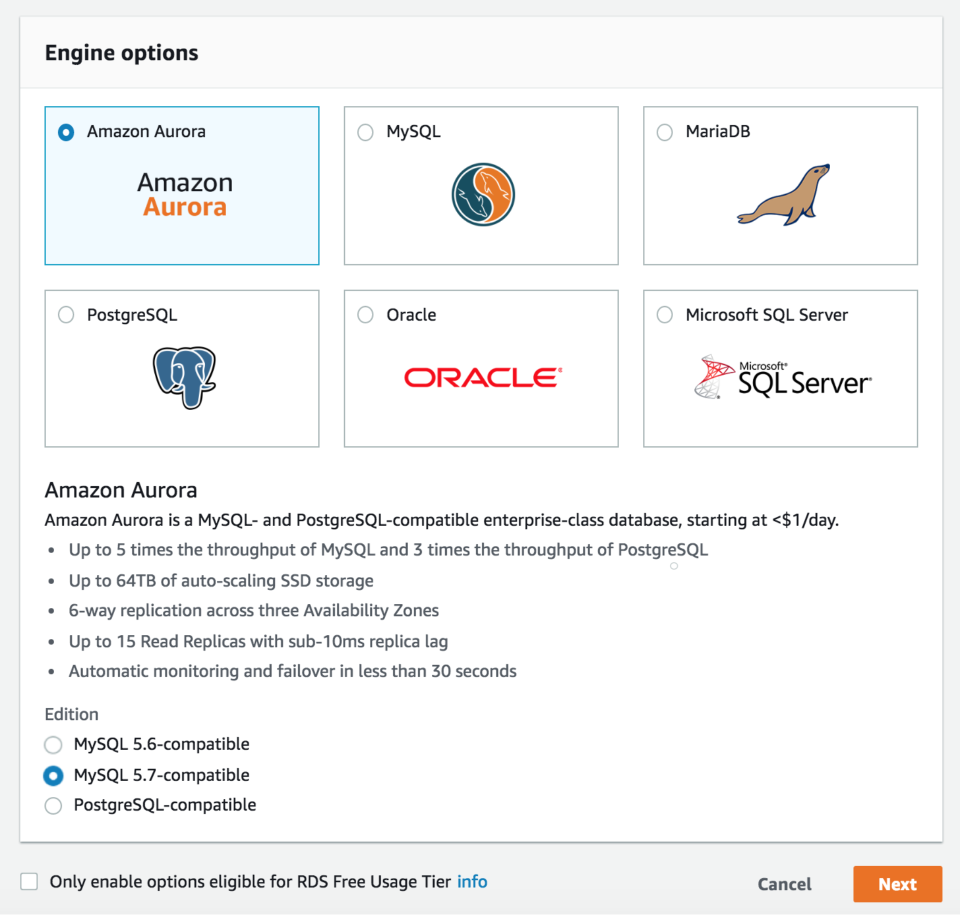
Parameter	Recommended Value
`character_set_client`	`utf8`
`character_set_connection`	`utf8`

character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
collation_connection	utf8_general_ci
collation_server	utf8_unicode_ci
innodb_default_row_format	DYNAMIC
innodb_file_per_table	1
lower_case_table_names	1

Launch an Amazon RDS Aurora DB Instance

You launch an Amazon RDS Aurora DB Instance from the AWS console.

1. Select **Aurora** with the **MySQL 5.7-compatible** option.



Engine options

● Amazon Aurora ○ MySQL ○ MariaDB
○ PostgreSQL ○ Oracle ○ Microsoft SQL Server

Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <$1/day.

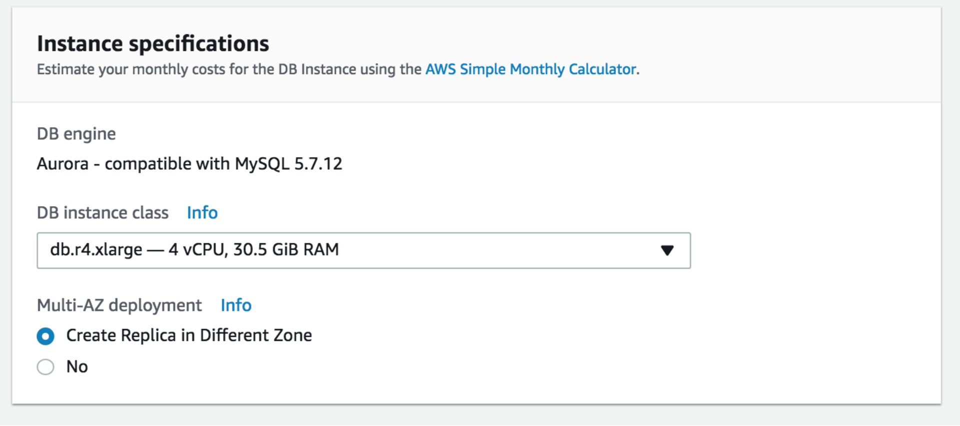
- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL
- Up to 64TB of auto-scaling SSD storage
- 6-way replication across three Availability Zones
- Up to 15 Read Replicas with sub-10ms replica lag
- Automatic monitoring and failover in less than 30 seconds

Edition

○ MySQL 5.6-compatible
● MySQL 5.7-compatible
○ PostgreSQL-compatible

□ Only enable options eligible for RDS Free Usage Tier info

2. Select the desired DB Instance class. See Prepare the AWS Machine for the Controller for instance sizing information
3. Select **Create Replica in Different Zone** to have Amazon RDS maintain a synchronous standby replica in a different Availability Zone than the DB instance. If a planned or unplanned outage of the primary occurs, Amazon RDS will automatically fail over to the standby replica.



Instance specifications

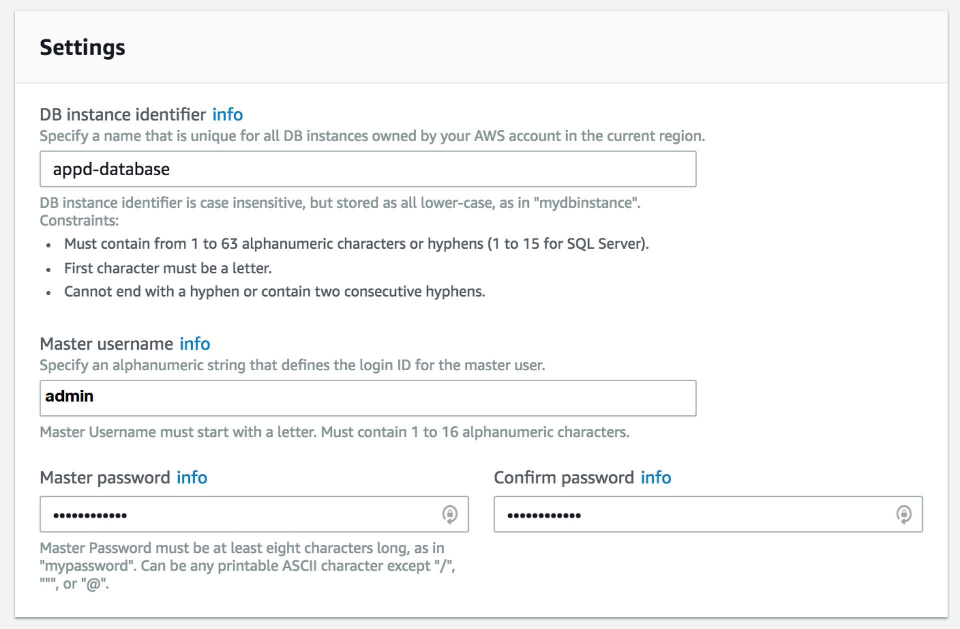
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.

DB engine
Aurora - compatible with MySQL 5.7.12

DB instance class Info
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM ▼

Multi-AZ deployment Info
● Create Replica in Different Zone
○ No

4. Enter an identifier for the DB instance. The master username must be `admin`.



Settings

DB instance identifier info
Specify a name that is unique for all DB instances owned by your AWS account in the current region.
appd-database

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".
Constraints:
- Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

Master username info
Specify an alphanumeric string that defines the login ID for the master user.
admin

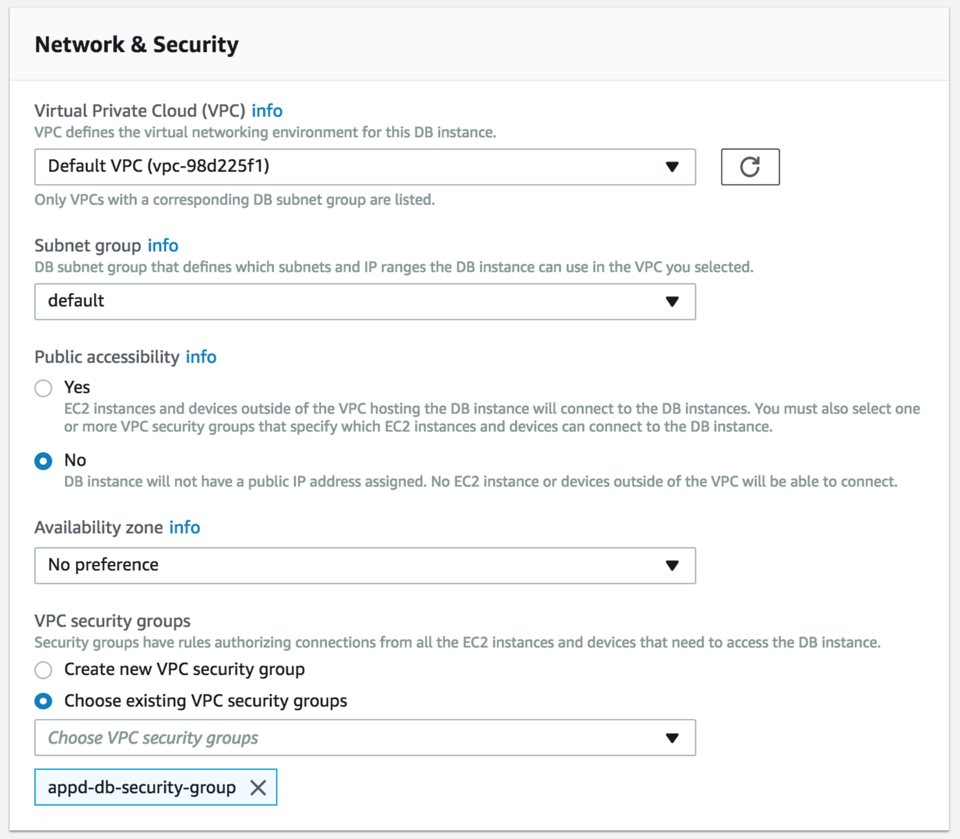
Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

Master password info
••••••••••••

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", """, or "@".

Confirm password info
••••••••••••

5. Select the Default VPC and subnet group. The DB should not be accessible publicly, so select **No** for this option.
6. For the security group, select the **appd-db-security-group** that you created previously.



Network & Security

Virtual Private Cloud (VPC) info
VPC defines the virtual networking environment for this DB instance.
Default VPC (vpc-98d225f1)
Only VPCs with a corresponding DB subnet group are listed.

Subnet group info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.
default

Public accessibility info

○ Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

● No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

Availability zone info
No preference

VPC security groups
Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.

○ Create new VPC security group

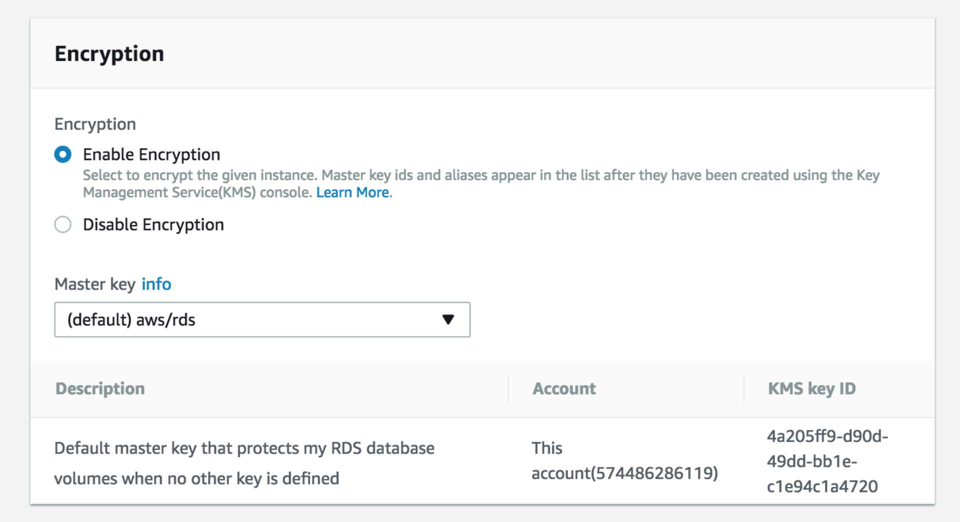
● Choose existing VPC security groups

Choose VPC security groups

appd-db-security-group ✕

7. For the database options, you do not need to specify the DB cluster identifier, nor enter a database name because the installer creates the necessary databases for you.
8. Use the default database port of `3388`.
9. Specify the custom parameter groups that you created previously.

10. We recommend that you select **Enable Encryption** for data at rest.



Encryption

Encryption

● Enable Encryption
Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. Learn More.

○ Disable Encryption

Master key info

(default) aws/rds ▼

Description	Account	KMS key ID
Default master key that protects my RDS database volumes when no other key is defined	This account(574486286119)	4a205ff9-d90d-49dd-bb1e-c1e94c1a4720

11. Use the default options for the remaining settings.
12. Click **Launch DB Instance**, and your new database will be available in a few minutes.

Create Database User for Controller

During installation, AppDynamics must create additional databases and users in the Aurora database for the AppDynamics Controller application to interact with the Aurora database server.

To create the Aurora database:

1. Create the Aurora database using `admin` as the primary username.
2. After the Aurora database instance is created successfully, log in to the ec2 instance as `admin`:

   ```
   mysql -u admin -h <rds-aurora-endpoint> -P 3388 -p
   ```

3. To create a new `'root'` user, enter:

   ```
   CREATE USER 'root'@'%' IDENTIFIED BY 'controller';
   ```

4. To check for the grants of the primary username (`admin`), enter:

   ```
   mysql> SHOW GRANTS FOR admin;
   ```

   Resulting output:

   ```
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------
   Grants for admin@%
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER, SHOW
   DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE
   VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, LOAD FROM S3, SELECT INTO
   S3, INVOKE LAMBDA ON *.* TO 'admin'@'%' WITH GRANT OPTION
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------------------------------------------------------------
   ---------------------------------------------------
    1 row in set (0.00 sec)
   ```

5. Apply the grants (listed in the output) for the new `root` user that you created in Step 1. The `root` user will have the same grants as the `admin` user.

   ```
   mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX, ALTER,
   SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT,
   CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, LOAD FROM S3, SELECT
   INTO S3, INVOKE LAMBDA ON *.* TO 'root'@'%' WITH GRANT OPTION
   ```

   Resulting output:

   ```
   Query OK, 0 rows affected (0.01 sec)
   ```

6. Once the `root` user has the same privileges as the primary username `admin`, verify that you can log in to the database as `root`, and then continue with the installation.

   - If you do not have users "root@*x.x.x.x*" and "root@ip-*x-x-x-x*.ec2.internal", ignore these users and continue to work with the `root@%`.

- If you have users "root@*x.x.x.x*" and "root@ip-*x-x-x-x*.ec2.internal", then instead of using the previous `GRANT` command, use this `GRANT` command:

  ```
  mysql> GRANT ALL ON `%`.* TO 'root'@'ip-x-x-x-x.ec2.internal' identified by 'controller' WITH GRANT OPTION;
  mysql> GRANT ALL ON `%`.* TO 'root'@'x.x.x.x' identified by 'controller' WITH GRANT OPTION;
  mysql> GRANT RELOAD ON *.* TO 'root'@'ip-x-x-x-x.ec2.internal' identified by 'controller' WITH GRANT OPTION;
  mysql> GRANT RELOAD ON *.* TO 'root'@'x.x.x.x' identified by 'controller' WITH GRANT OPTION;
  ```

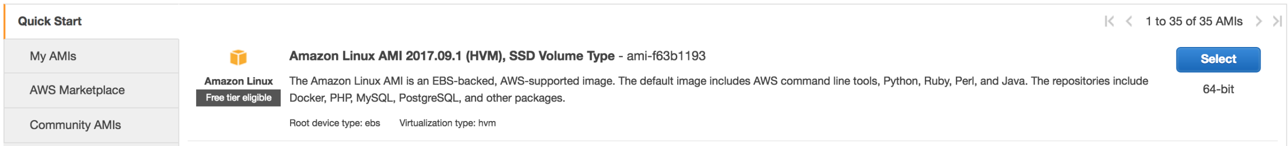
After installation, you can revoke the primary-level privileges from the Aurora *root* user without interfering with the Controller. However, primary-level privileges for Aurora *root* user are required prior to upgrading the Controller.

Launch an EC2 Instance for the Controller

This example uses an Amazon Linux AMI (which is provided by AWS: `Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-f63b1193`). The Amazon Linux AMI is an EBS-backed, AWS-supported image.

The default image includes AWS command-line tools: Python, Ruby, Perl, and Java.

The repositories include: Docker, PHP, MySQL, PostgreSQL, and other packages.



Quick Start

My AMIs

AWS Marketplace

Community AMIs

1 to 35 of 35 AMIs

Amazon Linux

Free tier eligible

Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-f63b1193

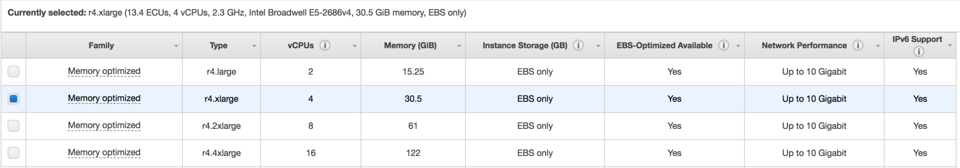
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

Select

64-bit

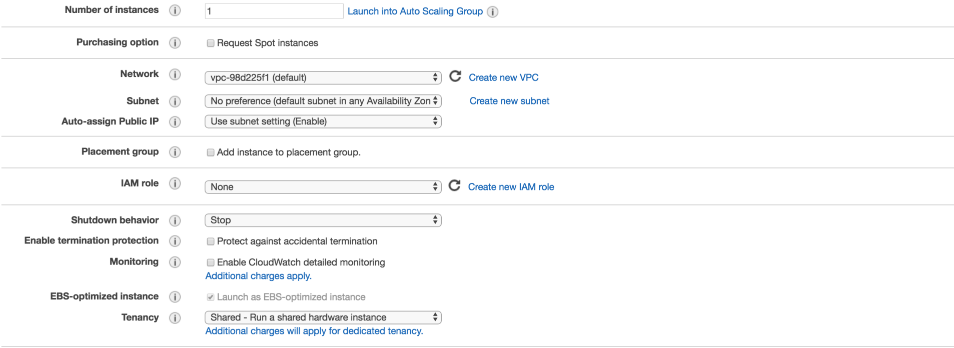
1. Select the instance type.



Currently selected: r4.xlarge (13.4 ECUs, 4 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 30.5 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	Memory optimized	r4.large	2	15.25	EBS only	Yes	Up to 10 Gigabit	Yes
☑	Memory optimized	r4.xlarge	4	30.5	EBS only	Yes	Up to 10 Gigabit	Yes
☐	Memory optimized	r4.2xlarge	8	61	EBS only	Yes	Up to 10 Gigabit	Yes
☐	Memory optimized	r4.4xlarge	16	122	EBS only	Yes	Up to 10 Gigabit	Yes

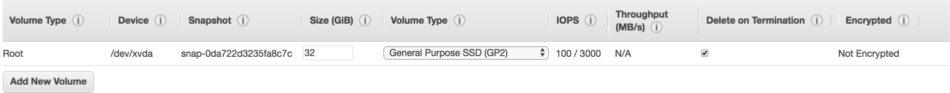
2. Use the default instance settings.



Number of instances	1 Launch into Auto Scaling Group
Purchasing option	☐ Request Spot instances
Network	vpc-98d225f1 (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zon Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	☐ Add instance to placement group.
IAM role	None Create new IAM role
Shutdown behavior	Stop
Enable termination protection	☐ Protect against accidental termination
Monitoring	☐ Enable CloudWatch detailed monitoring Additional charges apply.
EBS-optimized instance	☑ Launch as EBS-optimized instance
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

3. Increase the storage amount from the default 8 GB to 32 GB.

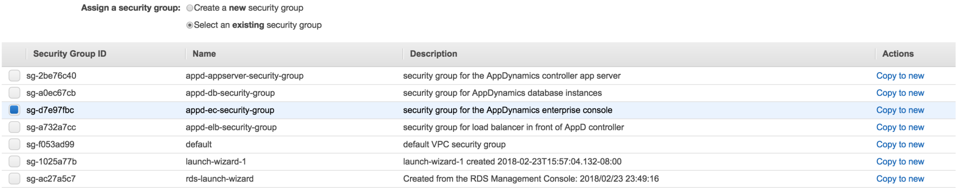
> ⚠ You will need considerably more space on a larger server.



Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0da722d3235fa8c7c	32	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

4. Assign the instance to the `appd-appserver-security-group` that you previously created.



Assign a security group: ○ Create a **new** security group

● Select an **existing** security group

	Security Group ID	Name	Description	Actions
☐	sg-2be76c40	appd-appserver-security-group	security group for the AppDynamics controller app server	Copy to new
☐	sg-a0ec67cb	appd-db-security-group	security group for AppDynamics database instances	Copy to new
☑	sg-d7e97fbc	appd-ec-security-group	security group for the AppDynamics enterprise console	Copy to new
☐	sg-a732a7cc	appd-elb-security-group	security group for load balancer in front of AppD controller	Copy to new
☐	sg-f053ad99	default	default VPC security group	Copy to new
☐	sg-1025a77b	launch-wizard-1	launch-wizard-1 created 2018-02-23T15:57:04.132-08:00	Copy to new
☐	sg-ac27a5c7	rds-launch-wizard	Created from the RDS Management Console: 2018/02/23 23:49:16	Copy to new

5. To launch the instance, you must specify a key pair. If you are using an existing key pair, ensure that you have access to the private key file; otherwise, generate a new key pair and download it from the AWS console. The private key file is required to connect to the instance using SSH. The new instance should be available after a few minutes. Once the instance is available, you can verify the status using the AWS console. Connect to it through SSH, enter:

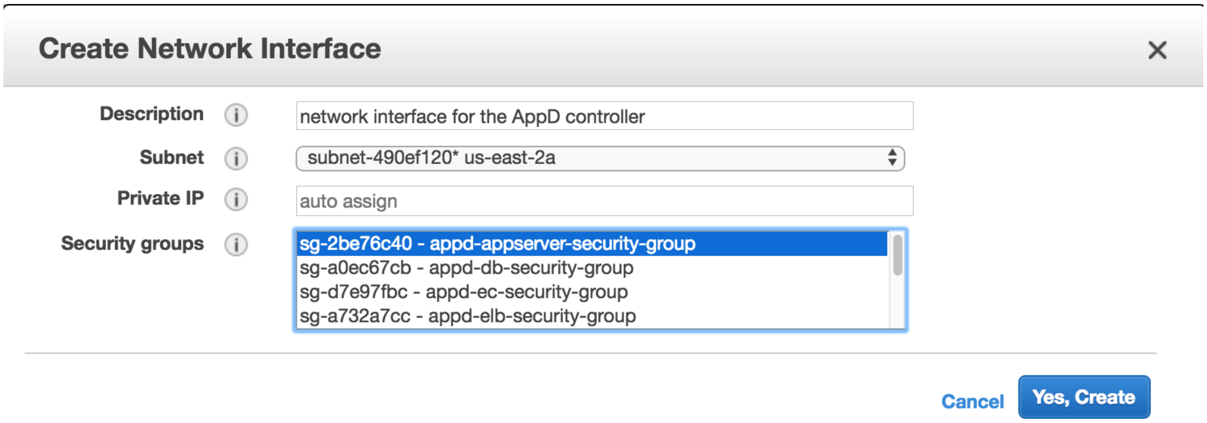
```
ssh -i "<private key file>.pem" ec2-user@ec2-18-222-75-189.us-east-2.compute.amazonaws.com
```

6. Substitute the appropriate path and filename for your private key file.

Create the ENI for the Controller

You need to introduce an ENI which acts as a secondary network interface that you can attach and detach from the Controller EC2 instance.

You then have to associate the AppDynamics license file to the MAC address of this network interface instead of the MAC address of the underlying EC2 instance.



Create Network Interface

Description: network interface for the AppD controller

Subnet: subnet-490ef120* us-east-2a

Private IP: auto assign

Security groups:
- sg-2be76c40 - appd-appserver-security-group
- sg-a0ec67cb - appd-db-security-group
- sg-d7e97fbc - appd-ec-security-group
- sg-a732a7cc - appd-elb-security-group

Attach the new network interface to the Controller EC2 instance.

Create DNS CNAMEs

AppDynamics recommends that you create aliases used to connect the Aurora database instance to the Controller EC2 instance.

For example:

```
appdcontroller.mydomain.com  appdcontrollerdb.cxylwiexaqo2.us-east-2.rds.amazonaws.com (DNS name for the Aurora DB instance)

appd-database.mydomain.com  ip-172-31-22-161.us-east-2.compute.internal (private DNS name for the ENI attached to the Controller)
```

You should use these aliases when you install the Controller through the Enterprise Console. Using aliases prevents you from tightly coupling the Enterprise Console with the specific Aurora DB instance or EC2 instance hosting the Controller.

For example, if the database were to fail completely, and you needed to restore the database from a snapshot, then you would have a new DNS name for the Aurora DB instance. Pointing the Enterprise Console at an alias, instead of the DNS name for the Aurora DB instance itself, allows you to only update the DNS alias, and leave the Enterprise Console configuration unchanged.

However, if you need to move the Controller to a different EC2 instance, which resides in another Availability Zone (AZ), you can just update the DNS alias to point to the ENI in the new AZ.

For purposes of testing an AWS configuration, it may not be possible to increase DNS aliases. As a result, you can just add entries to the `/etc/hosts` file on both the Enterprise Console and Controller EC2 instances. For example:

```
172.31.17.84 appdcontroller
172.31.25.80 appd-database
```

Install the Enterprise Console in an AWS Environment

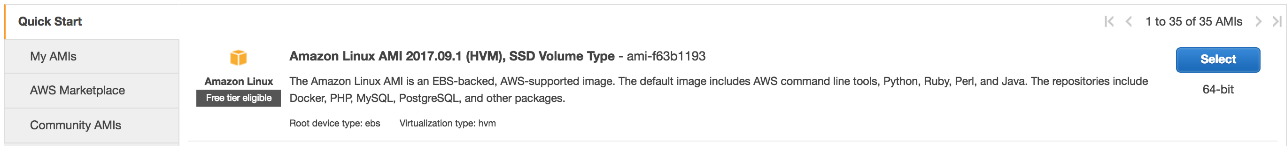
This page describes how to install the the Enterprise Console in an AWS environment using an EC2 instance. You then use the Enterprise Console to install the Controller on a separate EC2 instance (using Aurora DB as the backend).

Launch the EC2 Instance

This example uses this AMI (which is provided by AWS): `Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-f63b1193`. The Amazon Linux AMI is an EBS-backed, AWS-supported image.

The default image includes AWS command-line tools: Python, Ruby, Perl, and Java.

The repositories include: Docker, PHP, MySQL, PostgreSQL, and other packages.

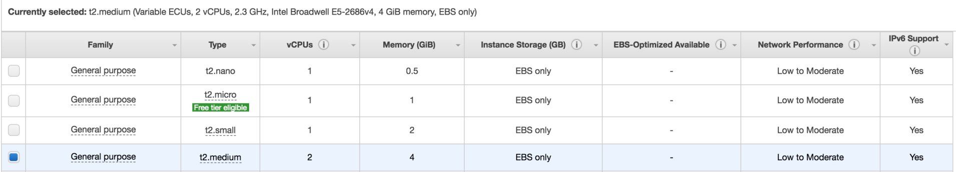


Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-f63b1193

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs   Virtualization type: hvm

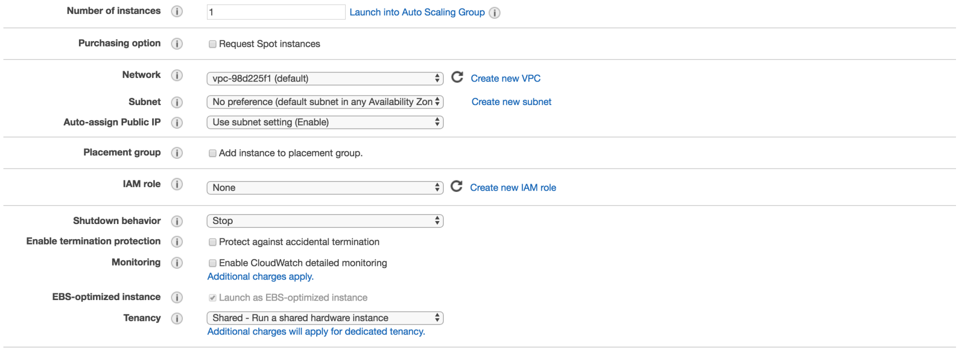
1. Select the Instance type. The Enterprise Console has only modest requirements, therefore you can select the **t2.medium** instance type.



Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4, 4 GiB memory, EBS only)

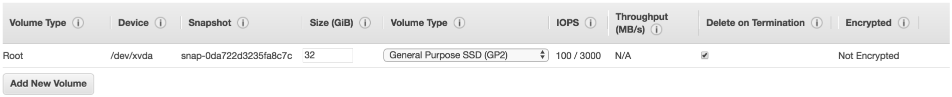
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
■	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

2. Use the default instance settings.



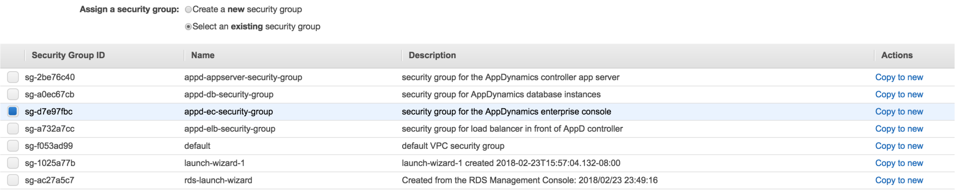
Number of instances	1 Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances
Network	vpc-98d225f1 (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zon Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	□ Add instance to placement group.
IAM role	None Create new IAM role
Shutdown behavior	Stop
Enable termination protection	□ Protect against accidental termination
Monitoring	□ Enable CloudWatch detailed monitoring Additional charges apply.
EBS-optimized instance	☑ Launch as EBS-optimized instance
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

3. Increase the storage amount from the default 8 GB to 32 GB.



Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0da722d3235fa8c7c	32	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

4. Assign the instance to the `appd-ec-security-group` that you created previously.



Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security Group ID	Name	Description	Actions
□	sg-2be76c40	appd-appserver-security-group	security group for the AppDynamics controller app server	Copy to new
□	sg-a0ec67cb	appd-db-security-group	security group for AppDynamics database instances	Copy to new
■	sg-d7e97fbc	appd-ec-security-group	security group for the AppDynamics enterprise console	Copy to new
□	sg-a732a7cc	appd-elb-security-group	security group for load balancer in front of AppD controller	Copy to new
□	sg-f053ad99	default	default VPC security group	Copy to new
□	sg-1025a77b	launch-wizard-1	launch-wizard-1 created 2018-02-23T15:57:04.132-08:00	Copy to new
□	sg-ac27a5c7	rds-launch-wizard	Created from the RDS Management Console: 2018/02/23 23:49:16	Copy to new

Prepare the Instance

To launch the instance, you must specify a key pair. If you are using an existing key pair, ensure you have access to the private key file; otherwise, generate a new key pair and download it from the AWS console. The private key file is required to connect to the instance using SSH.

The new instance should be available after a few minutes. Once the instance is available, you can verify the status using the AWS console. Connect to it through SSH, enter:

```
ssh -i "<private key file>.pem" ec2-user@ec2-18-222-75-189.us-east-2.compute.amazonaws.com
```

Then, substitute the appropriate path and filename for your private key file.

Install the Enterprise Console

Use `scp` to transfer the Enterprise Console installer binary to your EC2 instance, enter:

```
scp -i "<private key file>.pem" platform_setup.sh ec2-user@ec2-18-222-75-189.us-east-2.compute.amazonaws.com:
/data
```

Then, SSH to your EC2 instance and run the `installer` to install Enterprise Console:

```
cd /data
chmod 700 platform_setup.sh
./platform_setup.sh -c
```

While installing the Enterprise Console, you are prompted to either select a database port, or accept the default port of 3377.

> Do not use port 3388 because it conflicts with the Controller database port which is used later in the installation process.

You must have write access to the Enterprise Console installation directory you select.

When installing one Controller in the AWS environment, it is easier to install both the Controller and Enterprise Console on the same host.

However if you plan to install multiple Controllers and want to manage them through a single Enterprise Console instance, then you should install the Enterprise Console and the Controller on separate hosts.

Complete the installation of Enterprise Console, and make a note of any passwords you specify during the installation process.

Install the Controller in AWS Using Aurora

This page describes how to install the Controller in an AWS environment using an EC2 instance for the Controller Appserver, and an AWS RDS Aurora instance for the database. This page uses the Enterprise Console that you previously installed.

> Before you install the Controller using Aurora as the database, you must adjust the time zone of the Aurora database to match the time zone of the Controller server. By default, AWS sets the time zone equal to the UTC time zone. See updating the Aurora RDS time zone

To install the Controller in an AWS environment using an EC2 instance for the Controller Appserver, and an AWS RDS Aurora instance for the database:

1. From the Enterprise Console instance, create a new platform:

   ```
   cd ./appdynamics/platform/platform-admin/bin
   ./platform-admin.sh create-platform --name <platform_name> --installation-dir /data/appdynamics/platform
   /product
   ```

2. Add a new host to the platform and install the Controller on the same host as the Enterprise Console:

   ```
   ./platform-admin.sh add-hosts --platform-name testplatform --hosts localhost
   ```

3. Install the Controller using Aurora as the database. Substitute the appropriate values for the `admin user name`, `passwords`, and `Controller host` and `port`. Ensure that `databaseType` is set to `Aurora`, and use the `private DNS name` of the network interface attached to the Controller EC2 instance instead of the `DNS name` for the EC2 instance itself.

   ```
   ./platform-admin.sh submit-job --platform-name testplatform --service controller --job install --args
   controllerProfile=<profile_size> controllerPrimaryHost=<network_interface_private_DNS_name>
   controllerTenancyMode=single controllerRootUserPassword="<password>" mysqlRootPassword="<password>"
   controllerAdminUsername="admin" controllerAdminPassword="<password>" databaseType=Aurora
   controllerDBPort=3388 controllerDBHost="<auroraHost>"
   ```

   The installer connects to the Aurora DB instance, and creates the necessary databases, tables, and other objects. After a few minutes, the Controller should be installed and ready to use.

Apply Controller Optimizations

This page describes how to apply optimizations to the Controller.

Glassfish Configuration

You can configure domain protocols, network listeners, transports, and thread pools from the Enterprise Console UI. You can edit them from the AppServer Configurations page by selecting the platform, and navigating to **Configurations > Controller Settings > Appserver Configurations**.

These files contain sample content for each of the profiles:

- `domain protocols.txt`
- `domain network listeners`
- `domain transports.txt`
- `domain thread pools.txt`

> ⓘ The Enterprise Console restarts the Controller after you submit your configurations.

Configure a Load Balancer

You can configure a load balancer after you have installed the Controller, and it is running in EC2. The load balancer distributes traffic across multiple ports on the Controller.

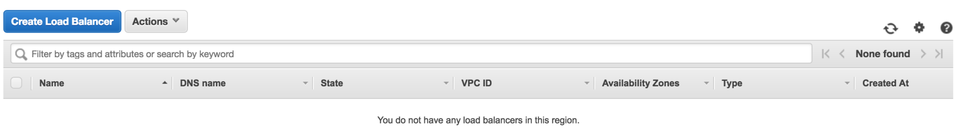
If using HTTPS, an SSL certificate should be available. For testing, you can generate a self-signed certificate using Open SSL (described here). You can then import that certificate using AWS Certificate Manager (ACM).

> ⚠ You can create a load balancer to align with your organization's standards.

Create a Load Balancer

To create a load balancer:

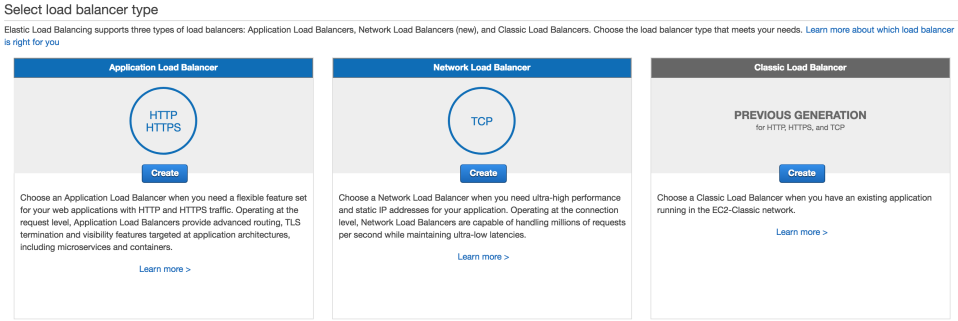
1. Navigate to the Load Balancers page on the EC2 Dashboard in the AWS console.
2. Select **Create Load Balancer** at the top left of the page.



Name	DNS name	State	VPC ID	Availability Zones	Type	Created At

You do not have any load balancers in this region.

3. Select **Application Load Balancer** as the load balancer type to terminate SSL at the ELB.



Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more about which load balancer is right for you

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing, TLS termination and visibility features targeted at application architectures, including microservices and containers.

Learn more >

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second while maintaining ultra-low latencies.

Learn more >

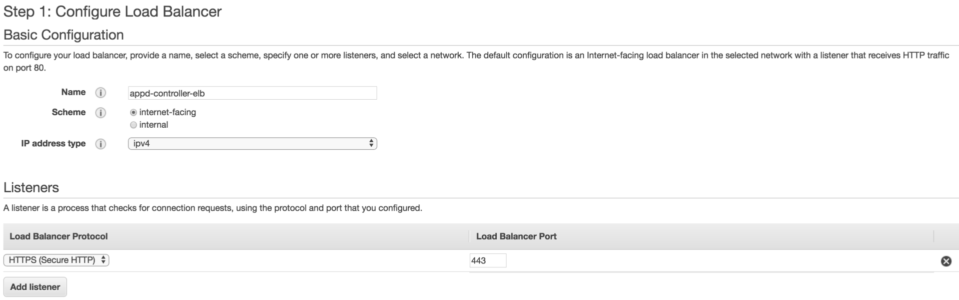
Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

Learn more >

4. Enter a name for the new load balancer, and select **HTTPS (Secure HTTP)** as the load balancer protocol to accept HTTPS traffic only.



Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

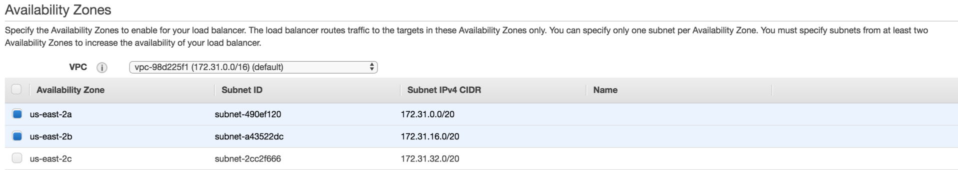
Name: appd-controller-elb
Scheme: ● internet-facing ○ internal
IP address type: ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTPS (Secure HTTP)	443

5. Select the availability zones to enable for the new load balancer. It should include the availability zone in which the Controller Appserver EC2 instance resides.



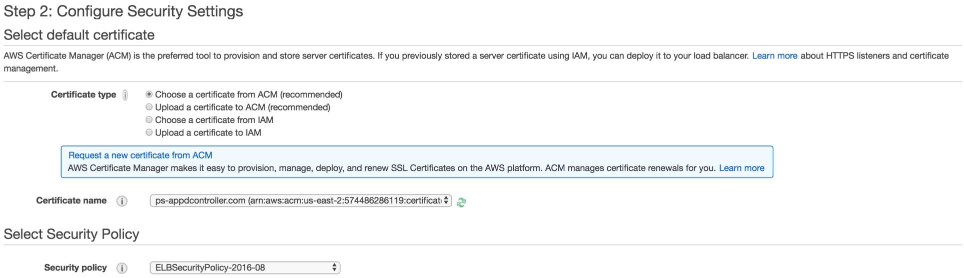
Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC: vpc-98d225f1 (172.31.0.0/16) (default)

	Availability Zone	Subnet ID	Subnet IPv4 CIDR	Name
☑	us-east-2a	subnet-490ef120	172.31.0.0/20	
☑	us-east-2b	subnet-a43522dc	172.31.16.0/20	
□	us-east-2c	subnet-2cc2f666	172.31.32.0/20	

6. Select the certificate that was imported to the ACM.



Step 2: Configure Security Settings

Select default certificate

AWS Certificate Manager (ACM) is the preferred tool to provision and store server certificates. If you previously stored a server certificate using IAM, you can deploy it to your load balancer. Learn more about HTTPS listeners and certificate management.

Certificate type
- ● Choose a certificate from ACM (recommended)
- ○ Upload a certificate to ACM (recommended)
- ○ Choose a certificate from IAM
- ○ Upload a certificate to IAM

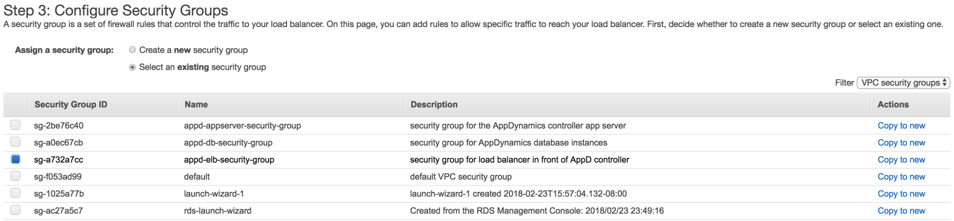
Request a new certificate from ACM
AWS Certificate Manager makes it easy to provision, manage, deploy, and renew SSL Certificates on the AWS platform. ACM manages certificate renewals for you. Learn more

Certificate name ps-appdcontroller.com (arn:aws:acm:us-east-2:574486286119:certificat

Select Security Policy

Security policy ELBSecurityPolicy-2016-08

7. Specify the security group.



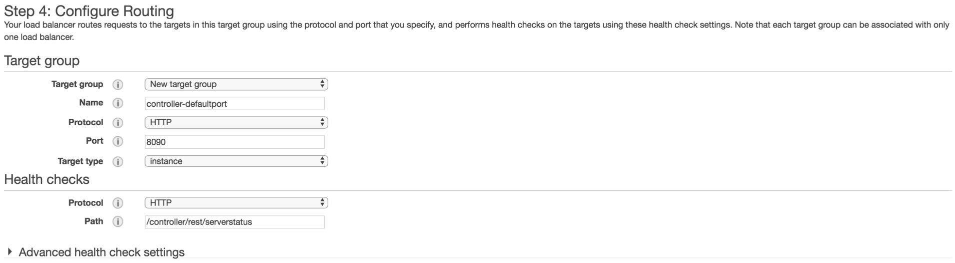
Step 3: Configure Security Groups
A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:
- ○ Create a **new** security group
- ● Select an **existing** security group

Filter VPC security groups

	Security Group ID	Name	Description	Actions
□	sg-2be76c40	appd-appserver-security-group	security group for the AppDynamics controller app server	Copy to new
□	sg-a0ec67cb	appd-db-security-group	security group for AppDynamics database instances	Copy to new
☑	sg-a732a7cc	appd-elb-security-group	security group for load balancer in front of AppD controller	Copy to new
□	sg-f053ad99	default	default VPC security group	Copy to new
□	sg-1025a77b	launch-wizard-1	launch-wizard-1 created 2018-02-23T15:57:04.132-08:00	Copy to new
□	sg-ac27a5c7	rds-launch-wizard	Created from the RDS Management Console: 2018/02/23 23:49:16	Copy to new

8. For the initial configuration, set the load balancer to route all traffic to port 8090 using HTTP, and define the standard health check for the Controller.



Step 4: Configure Routing
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

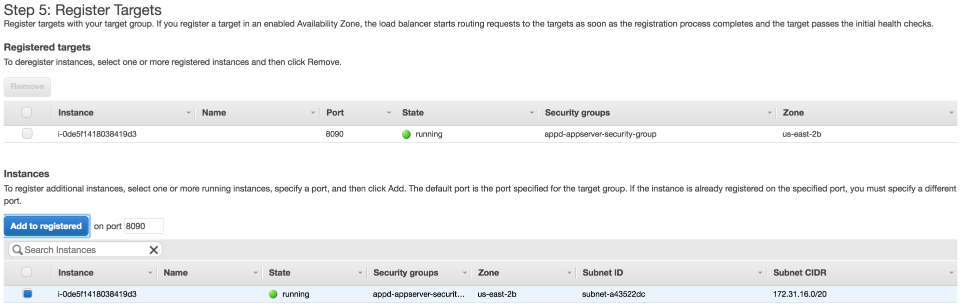
Target group New target group
Name controller-defaultport
Protocol HTTP
Port 8090
Target type instance

Health checks

Protocol HTTP
Path /controller/rest/serverstatus

▸ Advanced health check settings

9. Add the Controller EC2 instance as a registered target.



Step 5: Register Targets
Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

□	Instance	Name	Port	State	Security groups	Zone
□	i-0de5f1418038419d3		8090	● running	appd-appserver-security-group	us-east-2b

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

on port 8090

□	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
☑	i-0de5f1418038419d3		● running	appd-appserver-securit...	us-east-2b	subnet-a43522dc	172.31.16.0/20

10. Launch the new load balancer. It may take a few minutes before the load balance occurs.
11. Verify that you can access the Controller UI through the load balancer.

> If your SSL cert is self-signed, a browser warning displays. You can ignore this warning if you are testing the UI. However for Agent traffic, you need a valid certificate that is trusted by the Agent.

Specify the External Load Balancer URL

Use the Enterprise Console UI to update the Controller configuration and specify the external load balancer URL.

1. Open a browser and navigate to the UI:

```
http://<hostname>:<port>
```

   9191 is the default port.
2. Navigate to AppServer Configurations by selecting the platform: **Configurations > Controller Settings > Appserver Configurations**.
3. Enter the external load balancer URL in the appropriate field and select **Save**.

⚠ After applying the optimizations, you should create an AMI.

Configure Listener Rules

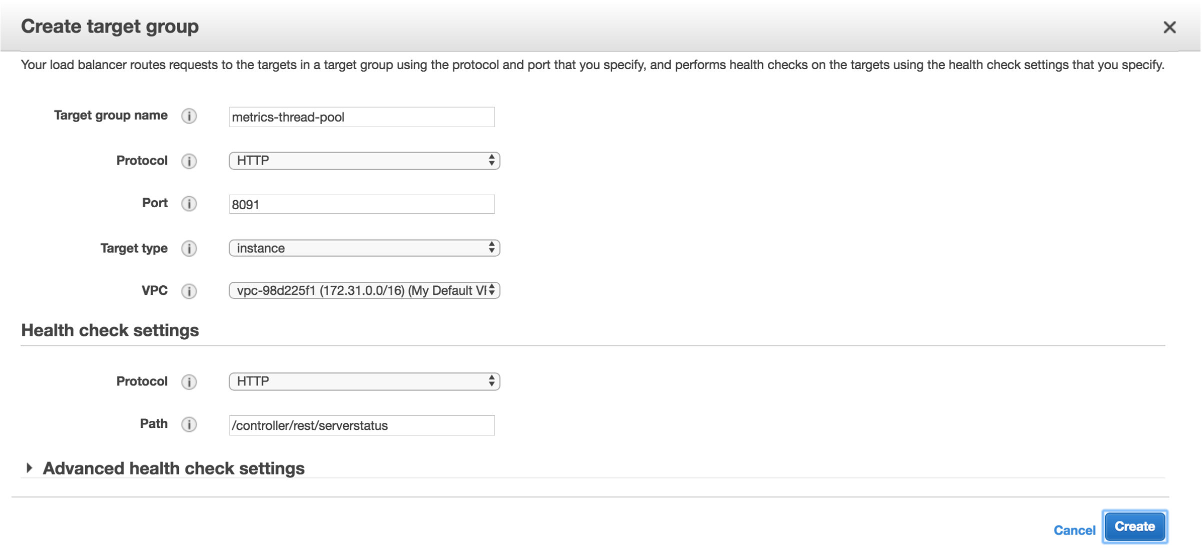
Depending on the type of request, you can define additional rules to route traffic to various ports on the Controller. This table defines rules that users typically include for their load balancer:

Pool Name	URL Pattern	Port Number	Description
`metrics-thread-pool`	`/controller/instance/*/metrics` `/controller/instance/*/metrics*`	8091	Agent metric data upload
`config-thread-pool`	`/controller/instance/*/applicationConfiguration*`	8092	Agent configuration requests
`agent-thread-pool`	`/controller/instance/*`	8093	Other Agent requests
`status-thread-pool`	`/controller/rest/serverstatus`	8094	Server status ping by load balancer
`http-thread-pool`	Default / User Traffic	8090	Default thread pool for all other traffic
`restui-default-thread-pool`	`/controller/restui/*`	8095	Default thread pool for all `restui` traffic
`restui-analytics-thread-pool`	`/controller/restui/analytics/*`	8096	Thread pool traffic for analytics traffic

Create Target Groups

You must create a target group for each of the rules listed in the table, with the exception of the default `http-thread-pool` rule (for which you can use the default target group).

For example, you create a target group for the `metrics-thread-pool` with these settings:



Create target group

Your load balancer routes requests to the targets in a target group using the protocol and port that you specify, and performs health checks on the targets using the health check settings that you specify.

Target group name	metrics-thread-pool
Protocol	HTTP
Port	8091
Target type	instance
VPC	vpc-98d225f1 (172.31.0.0/16) (My Default VI

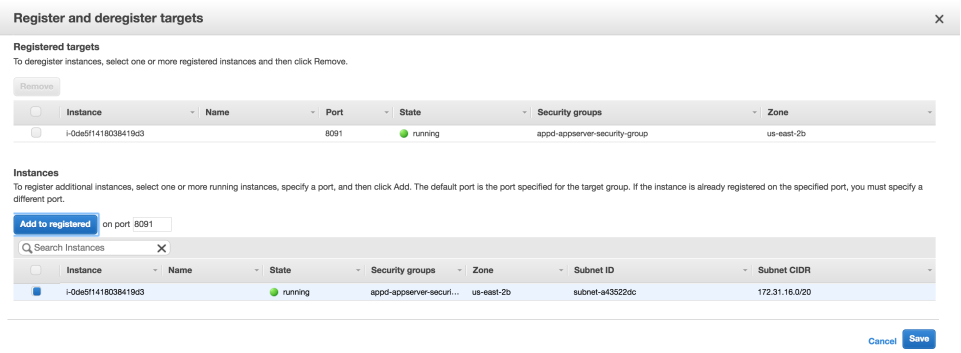
Health check settings

Protocol	HTTP
Path	/controller/rest/serverstatus

You can use the health check path for all of the target groups, however you may want to decrease the frequency because performing the same check on all ports every 30 seconds is not required.

Register Targets

1. To associate each target group with the EC2 instance, enter these settings:



Register and deregister targets ×

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

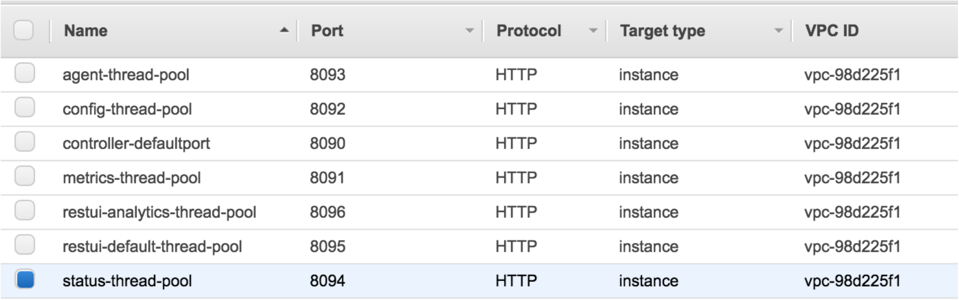
□	Instance	Name	Port	State	Security groups	Zone
□	i-0de5f1418038419d3		8091	● running	appd-appserver-security-group	us-east-2b

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

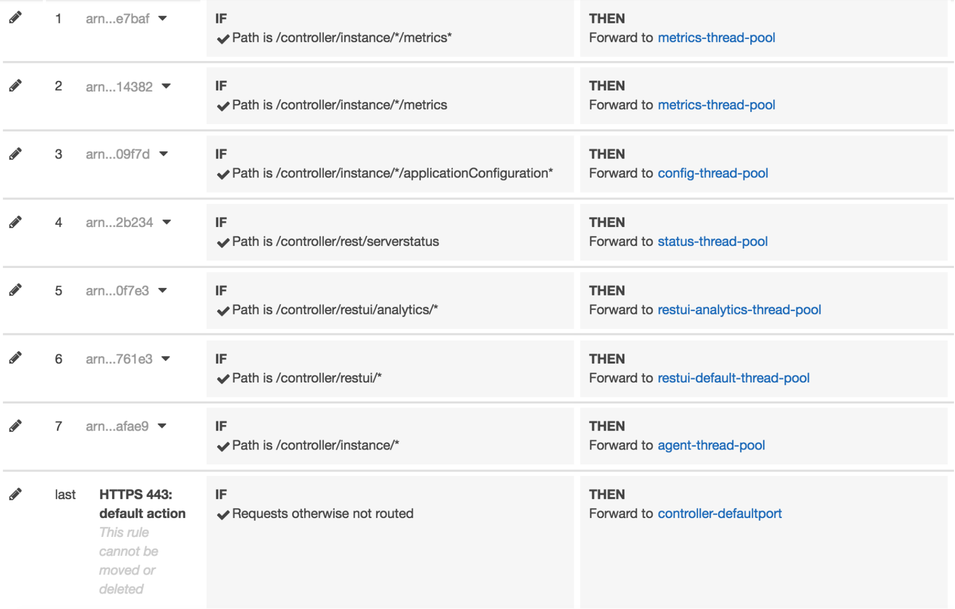
□	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
■	i-0de5f1418038419d3		● running	appd-appserver-securi...	us-east-2b	subnet-a43522dc	172.31.16.0/20

The full list of target groups should display:



□	Name	Port	Protocol	Target type	VPC ID
□	agent-thread-pool	8093	HTTP	instance	vpc-98d225f1
□	config-thread-pool	8092	HTTP	instance	vpc-98d225f1
□	controller-defaultport	8090	HTTP	instance	vpc-98d225f1
□	metrics-thread-pool	8091	HTTP	instance	vpc-98d225f1
□	restui-analytics-thread-pool	8096	HTTP	instance	vpc-98d225f1
□	restui-default-thread-pool	8095	HTTP	instance	vpc-98d225f1
■	status-thread-pool	8094	HTTP	instance	vpc-98d225f1

2. After the target groups have been defined, you can add new listener rules to map the traffic to the appropriate target group (based on the path requested):



		IF	THEN
1	arn...e7baf	✓ Path is /controller/instance/*/metrics*	Forward to metrics-thread-pool
2	arn...14382	✓ Path is /controller/instance/*/metrics	Forward to metrics-thread-pool
3	arn...09f7d	✓ Path is /controller/instance/*/applicationConfiguration*	Forward to config-thread-pool
4	arn...2b234	✓ Path is /controller/rest/serverstatus	Forward to status-thread-pool
5	arn...0f7e3	✓ Path is /controller/restui/analytics/*	Forward to restui-analytics-thread-pool
6	arn...761e3	✓ Path is /controller/restui/*	Forward to restui-default-thread-pool
7	arn...afae9	✓ Path is /controller/instance/*	Forward to agent-thread-pool
last	**HTTPS 443: default action** *This rule cannot be moved or deleted*	✓ Requests otherwise not routed	Forward to controller-defaultport

ⓘ The order of the rules is important because some paths may match multiple rules.

Create and Manage the AMI

You can create an Amazon Machine Image (AMI) to recover from an unexpected event or create a new Controller from the same image.

The AMI image should be created for the Controller, including required optimizations, and an auto-scaling group should be created based on the AMI. You can configure the Elastic Load Balancer (ELB) to send traffic to instances in the auto-scaling group, which will consist of one EC2 instance at a time.

Create an AMI and Auto Scaling Group

To ensure that you can recover from any interruptions, you need to create an AMI that includes the Controller. Then, you need to create an auto scaling group based on the AMI.

Creating an AMI and auto scaling group involves the following steps:

- Step 1: Shut Down the Enterprise Console
- Step 2: Create an AMI
- Step 3: Create the Launch Configuration
- Step 4: Create an Auto Scaling Group

Step 1: Shut Down the Enterprise Console

You must stop the Enterprise Console before creating an AMI:

```
bin/platform-admin.sh stop-platform-admin
```

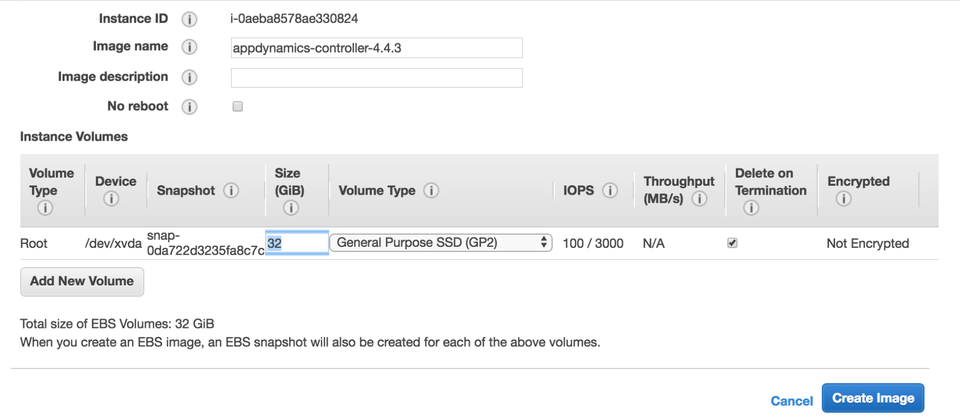
This ensures that all jobs and services are stopped in order to create a clean image.

Step 2: Create an AMI

You can create an AMI to use to deploy a new Controller without having to go through the installation steps again.

> A new AMI should be created whenever the Controller version is upgraded, or configuration changes are made. This ensures that the changes are propagated to the new EC2 instance, in the event that you need to move to a different availability zone, for example. At the same time, previous AMI instances should be retained as well, in case a rollback is required.

1. Stop the Controller Appserver once optimizations have been applied to it.
2. Click **Create Image** to create an AMI for the Controller.



Instance ID: i-0aeba8578ae330824

Image name: appdynamics-controller-4.4.3

Image description:

No reboot: □

Instance Volumes

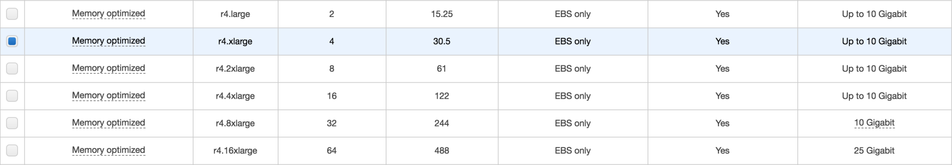
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0da722d3235fa8c7c	32	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Total size of EBS Volumes: 32 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Step 3: Create the Launch Configuration

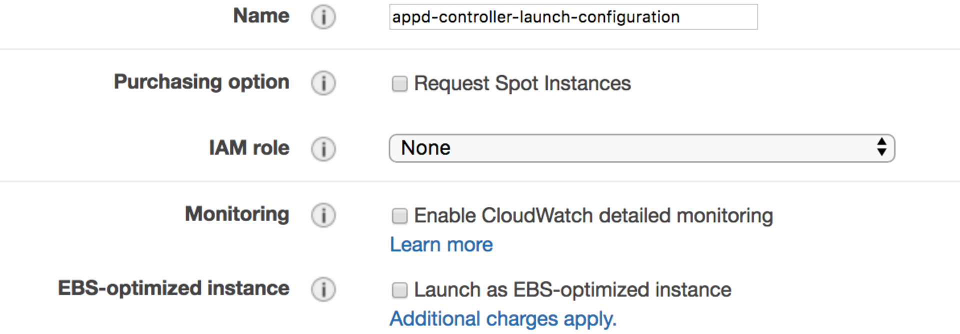
Once the AMI is created, you can create an auto scaling group that uses it. But first, you need to create the launch configuration:

1. Select the appropriate instance type for the app server of your AMI.



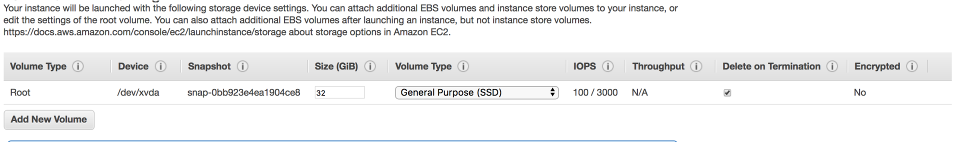
□	Memory optimized	r4.large	2	15.25	EBS only	Yes	Up to 10 Gigabit
■	Memory optimized	r4.xlarge	4	30.5	EBS only	Yes	Up to 10 Gigabit
□	Memory optimized	r4.2xlarge	8	61	EBS only	Yes	Up to 10 Gigabit
□	Memory optimized	r4.4xlarge	16	122	EBS only	Yes	Up to 10 Gigabit
□	Memory optimized	r4.8xlarge	32	244	EBS only	Yes	10 Gigabit
□	Memory optimized	r4.16xlarge	64	488	EBS only	Yes	25 Gigabit

2. Specify a name for the launch configuration.



Name ⓘ	appd-controller-launch-configuration
Purchasing option ⓘ	□ Request Spot Instances
IAM role ⓘ	None
Monitoring ⓘ	□ Enable CloudWatch detailed monitoring Learn more
EBS-optimized instance ⓘ	□ Launch as EBS-optimized instance Additional charges apply.

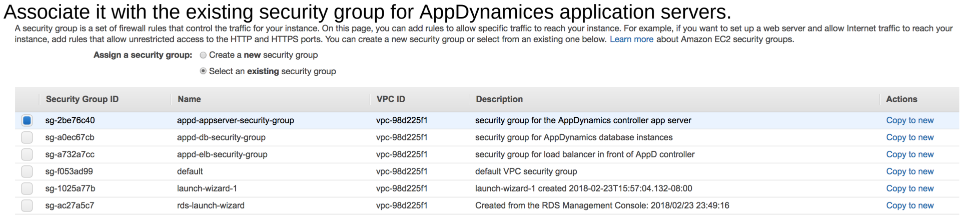
3. Specify storage configuration.



Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. https://docs.aws.amazon.com/console/ec2/launchinstance/storage about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-0bb923e4ea1904ce8	32	General Purpose (SSD)	100 / 3000	N/A	☑	No

4. Associate it with the existing security group for AppDynamices application servers.



A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group
● Select an **existing** security group

	Security Group ID	Name	VPC ID	Description	Actions
■	sg-2be76c40	appd-appserver-security-group	vpc-98d225f1	security group for the AppDynamics controller app server	Copy to new
□	sg-a0ec67cb	appd-db-security-group	vpc-98d225f1	security group for AppDynamics database instances	Copy to new
□	sg-a732a7cc	appd-elb-security-group	vpc-98d225f1	security group for load balancer in front of AppD controller	Copy to new
□	sg-f053ad99	default	vpc-98d225f1	default VPC security group	Copy to new
□	sg-1025a77b	launch-wizard-1	vpc-98d225f1	launch-wizard-1 created 2018-02-23T15:57:04.132-08:00	Copy to new
□	sg-ac27a5c7	rds-launch-wizard	vpc-98d225f1	Created from the RDS Management Console: 2018/02/23 23:49:16	Copy to new

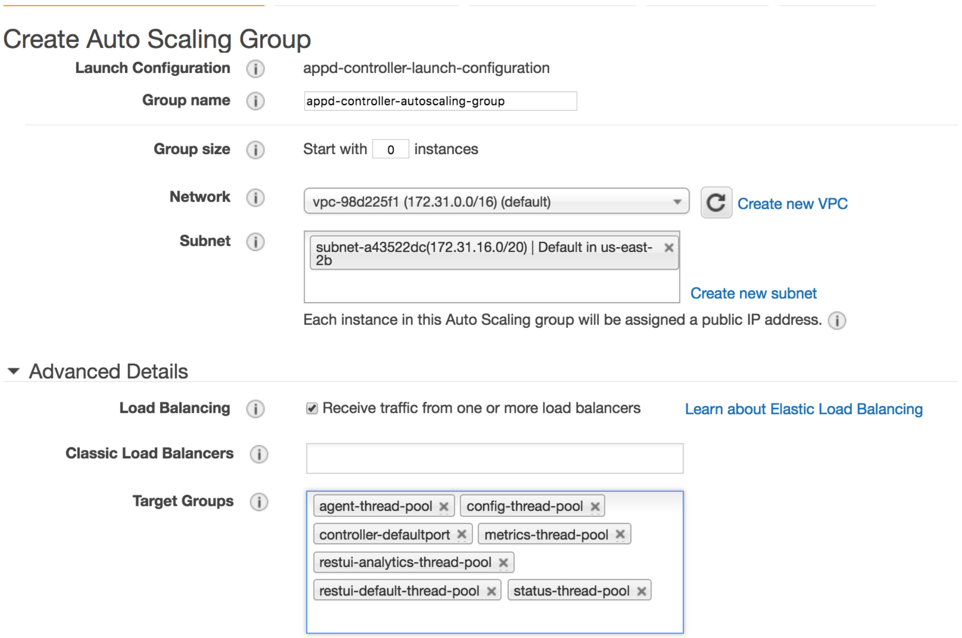
Step 4: Create an Auto Scaling Group

You should create an auto scaling group based on this AMI to ensure that one node is running at any given time.

To create an auto scaling group:

1. Specify a name for the auto scaling group, enable traffic from the load balancer, and map the target groups created earlier.

> ⚠ Start with zero instances, as you will need to add your existing instance to the group later.
> Keep the group at its initial size of zero, as you do not want to add more than one Controller at any point in time.



Create Auto Scaling Group

Launch Configuration: appd-controller-launch-configuration

Group name: appd-controller-autoscaling-group

Group size: Start with 0 instances

Network: vpc-98d225f1 (172.31.0.0/16) (default)

Subnet: subnet-a43522dc(172.31.16.0/20) | Default in us-east-2b

Each instance in this Auto Scaling group will be assigned a public IP address.

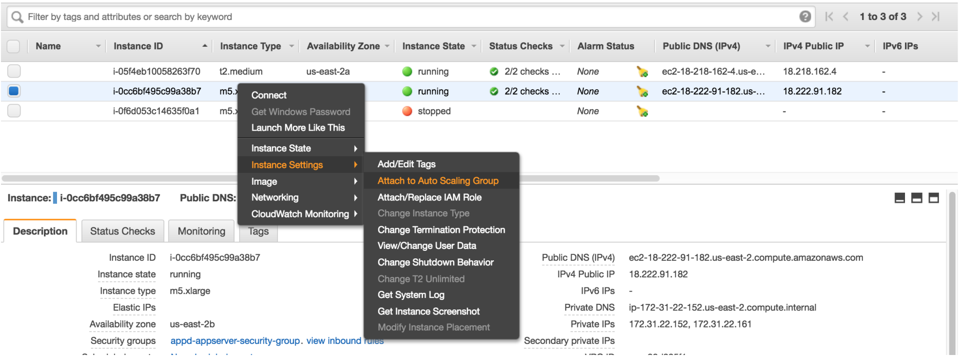
Advanced Details

Load Balancing: ☑ Receive traffic from one or more load balancers

Classic Load Balancers:

Target Groups: agent-thread-pool, config-thread-pool, controller-defaultport, metrics-thread-pool, restui-analytics-thread-pool, restui-default-thread-pool, status-thread-pool

2. Accept the default values for the rest of the settings, and create the auto scaling group.
3. Once the auto scaling group is successfully created, edit the properties to specify the max allowable instances as 1 (up from 0).
4. Go back to the list of EC2 instances, and add the EC2 instance for the Controller Appserver to the new auto scaling group.



Filter by tags and attributes or search by keyword

1 to 3 of 3

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
	i-05f4eb10058263f70	t2.medium	us-east-2a	running	2/2 checks ...	None	ec2-18-218-162-4.us-e...	18.218.162.4	-
	i-0cc6bf495c99a38b7	m5.x		running	2/2 checks ...	None	ec2-18-222-91-182.us-...	18.222.91.182	-
	i-0f6d053c14635f0a1	m5.x		stopped		None		-	-

Instance: i-0cc6bf495c99a38b7 Public DNS:

Instance ID: i-0cc6bf495c99a38b7
Instance state: running
Instance type: m5.xlarge
Elastic IPs:
Availability zone: us-east-2b
Security groups: appd-appserver-security-group. view inbound rules

Public DNS (IPv4): ec2-18-222-91-182.us-east-2.compute.amazonaws.com
IPv4 Public IP: 18.222.91.182
IPv6 IPs: -
Private DNS: ip-172-31-22-152.us-east-2.compute.internal
Private IPs: 172.31.22.152, 172.31.22.161
Secondary private IPs:

Updating the AMI

The AMI should be updated whenever changes are made to the controller EC2 instance. For example:

- When a new license file is dropped on the Controller.
- When O/S configuration changes are made.
- When Controller configuration changes are made, either via the Enterprise Console or directly to domain.xml or other configuration files.

Migrate the Controller Database to Amazon Aurora

You have the option to migrate an existing 4.4.3 or latest on-premises Controller database to Aurora DB. The Controller might already reside in AWS, or in your data center.

Although the Controller already contains a MySQL database, you are recommended to migrate the MySQL database to Aurora because it offers replication, high availability, and elasticity. The Amazon Relational Database Service (RDS) tool handles provisioning, patching, backup, recovery, failure detection, and repair of the database. Also, Aurora DB offers encryption at rest, with encryption of all automated backups, snapshots, and replicas in the same cluster.

If your Controller is not already in AWS, then follow Migrate the Controller to migrate it. Once this is complete, you should have a Controller running on one or two EC2 instances in AWS, depending on whether or not your existing Controller deployment is high availability (HA), with the MySQL database hosted on those instances.

> ⚠ Commands to start and stop the database do not work with Aurora DB.

Migrate MySQL to an Aurora Database for 4.4.3 or Latest

You can create and configure Amazon Aurora to serve as the Controller database for 4.4.3 or the latest version. This process, which uses mysqldump to migrate the database, is required for Controllers running MySQL version 5.7. See Back Up the Controller with mysqldump for more information.

> ⓘ Since Controller upgrades to 4.4.3 from 4.3.x or earlier would use MySQL 5.5, it is important that you know what your Controller MySQL version is. Please refer to Bundled MySQL Database Version to learn how to check your MySQL version and upgrade it if necessary.

Note that running a Controller on AWS requires that some of the `cluster` parameter group and db parameter group settings be adjusted. See Deploy the Controller on AWS for more information.

Migrating MySQL to an Aurora Database involves the following steps:

> ⓘ If you attempt to upgrade or move a Controller migrated without its `liquibase-stored` procedures, the upgrade will fail. You must recreate these stored procedures manually in AWS.

1. Step 1: Provision an Empty Aurora Database
2. Step 2: Use mysqldump to Export from MySQL
3. Step 3: Use mysqldump to export stored procedures from the AppDynamics database
4. Step 4: Use mysql to Import to Aurora
5. Step 5: Configure the Controller to Use the Aurora Database

Step 1: Provision an Empty Aurora Database

You first need to start up a new instance of Aurora, using the desired instance type and other custom settings as explained in Deploy the Controller on AWS. Ensure that the database instance is created using port 3388.

Step 2: Use mysqldump to Export from MySQL

> ⚠ Before using `mysqldump`, first, ensure that the Controller app server is stopped. If you attempt to run `mysqldump` while the app server is running, it will severely degrade the performance and stability of the Controller.

To use `mysqldump`, run the `mysqldump` executable, passing the root username, password, and output file:

1. Run the following command to navigate to the executable directory:

   ```
   cd <controller_home>/db/bin
   ```

2. Use the following command to export the database from MySQL:

```
./mysqldump -u root --databases controller licensing mds_auth mds_configuration mds_entitysearch mds_infra_core mds_infra_server mds_license mds_metadata mds_metering mds_rbac mds_topology --single-transaction --compress --order-by-primary -p "<password>" > backup.sql
```

3. In order to import the resulting file into Aurora, you need to replace the following line:

```
/*!50013 DEFINER=`controller`@`localhost` SQL SECURITY DEFINER */
```

With:

```
/*!50013 DEFINER=`controller`@`%` SQL SECURITY DEFINER */
```

Step 3: Use mysqldump to export stored procedures from the AppDynamics database

Run the following command to export the stored procedures from the AppDynamics database.

```
./mysqldump -u root -p --protocol=TCP -h 127.0.0.1 -P <controller_mysql_port> --no-create-db --skip-add-drop-table --no-create-info --skip-disable-keys mysql proc --result-file=/staging/path/for/mysql.proc.sql
```

This command, through the `--result-file` option, dumps the stored procedures to `/staging/path/for/mysql.proc.sql`.

Step 4: Use mysql to Import to Aurora

1. Run the following command to navigate to the executable directory:

```
cd <controller_home>/db/bin
```

2. Connect to the new Aurora instance:

```
./mysql -u root -p"<password>" -h <hostname>.<aws-region>.rds.amazonaws.com -P 3388 --protocol=TCP
```

3. Then create the Controller user, and grant it permissions:

```
CREATE USER 'controller'@'%' IDENTIFIED BY 'controller';
GRANT USAGE ON *.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `controller`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `licensing`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_auth`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_configuration`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_entitysearch`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_infra_core`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_infra_server`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_license`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_metadata`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_metering`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_rbac`.* TO 'controller'@'%';
GRANT ALL PRIVILEGES ON `mds_topology`.* TO 'controller'@'%';
FLUSH PRIVILEGES;
```

ⓘ **Note**

The Aurora database is protected by security groups to prevent access from unauthorized sources.

4. Import the database backup:

```
./mysql -u controller -P 3388 -H <hostname>.<aws-region>.rds.amazonaws.com -p "controller" --
protocol=TCP < backup.sql
```

5. Import the stored procedures:

> ⚠ The import must be made by an admin or root user.

```
./mysql -u admin -P 3388 -H <hostname>.<aws-region>.rds.amazonaws.com -p "controller" --protocol=TCP <
/staging/path/for/mysql.proc.sql
```

Step 5: Configure the Controller to Use the Aurora Database

1. Change the configuration in the Controller to use the Aurora DB:
   a. In the file `<controller_home>/appserver/mq/lib/props/broker/default.properties`, set the property `imq.persist.jdbc.mysql.property.url` to your Aurora database:

```
imq.persist.jdbc.mysql.property.url=jdbc\:mysql\://<aurora-db>.<aws-region>.rds.amazonaws.com\:3388
/controller
```

   b. In the file `<controller_home>/appserver/glassfish/domains/domain1/config/domain.xml`, set the property `serverName` to the value of your Aurora database:

```
<property name="serverName" value="<aurora-db>.<aws-region>.rds.amazonaws.com"></property>
<property name="portNumber" value="3388"></property>
<property name="url" value="jdbc\:mysql\://<aurora-db>.<aws-region>.rds.amazonaws.com\:3388
/controller?nullNamePatternMatchesAll=true&amp;allowLoadLocalInfile=true&amp;
cachePrepStmts=true&amp;prepStmtCacheSize=25&amp;dumpQueriesOnException=true&amp;
rewriteBatchedStatements=true&amp;useSSL=false&amp;maxAllowedPacket=104857600"/>
```

   Add the following line to the same file, right before the `javaagent` option:

```
<jvm-options>-Dappdynamics.controller.use.global.datadir.query.for.disk.space.check=false</jvm-
options>
```

   c. In the file `<controller_home>/bin/controller_maintenance.xml`, set the property `db-host` to the value of your Aurora database:

```
<property name="db-host" value="<aurora-db>.<aws-region>.rds.amazonaws.com"/>
<property name="db-port" value="3388"/>
```

   d. In the file `<controller_home>/bin/setup.xml`, set the property `db-host` to the value of your Aurora database:

```
<property name="db-host"value="<aurora-db>.<aws-region>.rds.amazonaws.com"/>
<property name="db-port" value="3388"/>
```

2. Remove the following cache folders from `<controller_home>/appserver/glassfish/domains/domain1` as follows:
   a. `rm -rf osgi-cache/`
   b. `rm -rf generated/`
3. With the Controller service installed, start the Controller with root:

```
service controller start
```

4. Verify that the Controller is running successfully. The local MySQL database should be shut down, and you should see the migrated data in Aurora, which can be verified via the Controller UI.

Reset the Controller Database Root User Password

To reset the Controller Database root user password:

1. Log in to the RDS instance as admin:
   a. `./mysql -u admin -h <rds-aurora-endpoint> -P 3388 -p`
2. Use MySQL to run the commands:
   a. To specify the Controller database, enter: `use mysql;`
   b. To reload the MySQL grant tables, enter: `FLUSH PRIVILEGES;`
   c. To determine your MySQL version, enter: `select version();`
   d. If you are using MySQL version 5.5, to configure the new password for the root user, enter: `update mysql.user set password=password('<new-password-here>') where user like 'root%';`
   e. If you are using MySQL version 5.7. to configure the new password for the root user, enter: `update mysql.user set authentication_string=password('<new-password-here>') where user like 'root%';`
   f. To reload the MySQL grant tables, enter: `FLUSH PRIVILEGES;`
   g. To exit MySQL, enter: `quit`

Change the Controller Database Root User Password

By default, the password for the controller user of the Aurora database used in your AppDynamics deployment is `controller`.

To change the Controller Database root user password:

1. Log in to the RDS instance as root:
   a. `./mysql -u root -h <rds-aurora-endpoint> -P 3388 -p`
2. Use MySQL to run the commands:
   a. To specify the Controller Database, enter: `use mysql;`
   b. To reload the MySQL grant tables, enter: `FLUSH PRIVILEGES;`
   c. To determine your MySQL version, enter: `select version();`
   d. If you are using MySQL version 5.5, to configure the new password for the root user, enter: `update mysql.user set password=password('<new-password-here>') where user like 'controller%';`
   e. If you are using MySQL version 5.7, to configure the new password for the root user, enter: `update mysql.user set authentication_string=password('<new-password-here>') where user like 'controller%';`
   f. To reload the MySQL grant tables, enter: `FLUSH PRIVILEGES;`
   g. To exit MySQL, enter: `quit`
3. Update the `controller-db-password` alias with the new Controller DB password in Glassfish:

```
cd <controller_home>/appserver/glassfish/bin
./asadmin update-password-alias controller-db-password
```

4. Restart the Controller AppServer for the changes to take effect:

```
cd <controller_home>/bin
./controller.sh stop-appserver
./controller.sh start-appserver
```

5. Verify the change in asadmin:

```
cd <controller_home>/appserver/glassfish/bin
./asadmin ping-connection-pool controller_mysql_pool
```

Upgrade or Move the Controller on AWS

If your Controller instance is deployed on AWS and uses Aurora for its database, you can use the Enterprise Console CLI commands to either *discover and update* or to just *upgrade* your Controller to the latest version. You *cannot* use the Enterprise Console UI, however, to perform the upgrade.

After backing up the Aurora database, use the upgrade method below that best meets your needs:

- Discover and Upgrade - Use this method if you are not sure if you need to upgrade.
- Upgrade - Use this method if you know you are using an older version and want to upgrade. For example, if you want to upgrade from 4.4.3 to the latest.

You can also move the Controller to a new EC2 instance to meet updated performance requirements. For platform-agnostic upgrade instructions, see Upgrade the Controller Using the Enterprise Console.

Back Up the Aurora Database

Before upgrading the Controller, you should back up the Aurora database instance. You should also ensure that you have an Amazon Machine Image (AMI) that accurately reflects the current Controller instance.

> The Enterprise Console upgrades the schema to the latest version. However, upgrading the Controller does not upgrade the Aurora DB server.

Discover and Upgrade the Controller on AWS

The Enterprise Console provides the `discover_upgrade` command that will discover your Controller, determine its version, and then upgrade it if needed.

1. Log in to the Enterprise Console:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

2. Run the following command:

   ```
   bin/platform-admin.sh submit-job --service controller --job discover-upgrade --args
   controllerPrimaryHost="<hostname>" controllerRootUserPassword="<password>" mysqlRootPassword="
   <password>"  databaseType=aurora destinationDirectory="<controllerInstallationDirector>"
   ```

   If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`
3. Update the AMI after the job finishes.

Upgrade the Controller on AWS

The Enterprise Console provides the `upgrade` command to upgrade the Controller to the latest version. For example, if you are using Controller 4.4.3 and want to upgrade to the latest, you can simply use the `upgrade` command.

1. Log in to the Enterprise Console:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

2. Run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh submit-job --service controller --job upgrade --args controllerRootUserPassword="
   <password>" mysqlRootPassword="<password>"
   ```

   If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`

3. Update the AMI after the job finishes.

Rollback to a Previous AMI

If a rollback is required, complete the following steps:

1. Create a new Aurora instance, using the database snapshot you took earlier as the source.
2. Stop the upgraded Controller if it is still running:

```
bin/platform-admin.sh stop-controller-appserver
```

3. Repoint the database DNS alias to the new Aurora instance.
4. Terminate the EC2 instance hosting the current Controller. This should cause a new EC2 instance to be provisioned using the existing AMI, with the older Controller version.
5. Attach the ENI to the new EC2 instance.
6. Verify that the Controller is working as it was before the upgrade.

Move the Controller on AWS

You can move the Controller to a new EC2 instance by completing the following steps:

1. Terminate the EC2 instance hosting the current Controller. This should cause a new EC2 instance to be automatically provisioned using the AMI.

> The auto-scaling group and launch configuration are defined with the AMI. Therefore, if the existing EC2 instance in the auto-scaling group dies, it is automatically replaced with a new EC2 instance based on the same AMI.

2. Attach the ENI to the new EC2 instance.
3. Verify that the Controller is working as expected.

Platform Installation Quick Start

This page provides a high-level view of using the Enterprise Console to install the AppDynamics platform, including the Controller and Events Service.

See Discovery and Upgrade Quick Start for the upgrade quick start guide.

About these Steps

The process described in this page uses the Enterprise Console to perform the following tasks:

- Install a Demo Controller and embedded Events Service.
- Install an Events Service that runs on hosts that are separate from the Controller.
- Switch the Controller to a high availability pair.

This quick start is intended to introduce you to the Enterprise Console and AppDynamics platform. Before you start, review the requirements pages for the platform components.

Install the Enterprise Console

1. Install the Enterprise Console with the following command:

   Linux

   ```
   ./platform-setup-64bit-linux.sh
   ```

   Windows

   ```
   platform-setup-64bit-windows.exe
   ```

   When the installation wizard launches, complete it to install the Enterprise Console. For more information about how to install the Enterprise Console, see Install the Enterprise Console.
2. Verify that the Enterprise Console successfully installed by opening the GUI using the host and port you specified during installation:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
3. Complete the installation process with the GUI or command line.

GUI Installation Quick Start

After you install the Enterprise Console, you can complete the platform installation process with the GUI. The following steps describe how to install a Controller with an embedded Events Service. It then describes how to scale up the Events Service to run on a separate host. To install a Controller with a scaled-up (unembedded) Events Service, see Custom Install.

You can install the Events Service on a separate host directly by selecting a Custom Installation at step 2.

1. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
2. Select **Express Install** to install a Controller and Events Service on a shared host.
3. In the **Add a Host** section, choose **Use Enterprise Console Host**. This option installs a Controller with an embedded Events Service on the same host as the Enterprise Console.
4. Choose **Demo** as the Controller profile if you are just getting to know the system.

5. Fill out the Controller configuration fields as indicated. See Express Install for details.
   You now have a running, testable system and can deploy agents. Next, we'll walk you through the steps for scaling up the Events Service. A scaled-up deployment is strongly recommended for installations meant for something other than for demonstration or exploratory purposes.

Scale Up the Events Service

In this step, you expand your platform to use a scaled-up Events Service deployment. It is important to note that data is not migrated from an embedded instance to a scaled-up instance.

1. In the Enterprise Console UI home page, click on the Platform you created previously.
2. Click **Hosts** in the left navigation menu.
3. Add a least three hosts for a scaled-up Events Service, providing a host name and an SSH credential that enables the Enterprise Console to access the host.
4. When finished, go to the **Events Service** page.
5. From the more menu (...), choose **Scale Up Events Service**.
6. In the **Scale Up Events Service** dialog, choose the **Prod** profile and select the hosts you created previously from the **Host** drop-down menu.
7. Click **Submit**.
8. Now set up a load balancer for the Events Service hosts, as described in Load Balance Events Service Traffic.
9. When done configuring the load balancer, you need to indicate to the Controller the addressable URL for the Events Service. In the Controller Administration Console, set these Controller properties:
   - Set `appdynamics.non.eum.events.use.on.premise.events.service` to true.
   - Set `appdynamics.analytics.server.store.url` to the URL of the Events Service as exposed at the load balancer. For example, `appdynamics.analytics.server.store.url=http://es-master-host:9080/`.

Set Up Controller High Availability

Now that you have a running Controller and Events Service, try setting up high availability for the Controller.

Before proceeding, see Set Up a High Availability Deployment. Specifically, follow the steps indicated in the section on configuring Controller High Availability pair environment in for prerequisites.

Once you have fulfilled the prerequisites, follow these steps:

1. In the Enterprise Console UI home page, click on the Platform you created previously.
2. Click **Hosts** in the left navigation menu.
3. Add a single new host. Provide a hostname and an SSH credential that enables the Enterprise Console to access the host.
4. When finished, go to the **Controller** page.
5. Click **Add Secondary Controller**.
6. Provide the same passwords as the primary database and Controller root password, and click **Submit**.

Now you have two Controllers running in passive-active high availability mode. To complete the configuration, you need to set up a load balancer, so that you can easily switch traffic between the Controllers. See Use a Reverse Proxy. After configuring a load balancer, you would need to specify the URL address for reaching the Controller pair in the Enterprise Console UI by clicking **Configurations** > **Controller Settings** > **Appserver Configurations**, and specifying the URL in the **External URL** field.

You now have a functioning platform, consisting of Controller pair and a scaled up Events Service instance.

Install the EUM Server

If using EUM, you can continue by installing an EUM Server. The EUM Server is installed separately, not with the Enterprise Console.

Follow the steps below to get started:

1. Read an overview of the EUM Server in EUM Server Deployment.
2. Learn to install and configure the EUM Server by following the instructions in Install a Demo EUM Server.
3. Install the EUM Server for production.

Discovery and Upgrade Quick Start

Related pages:

- Upgrade Platform Components
- Discover Existing Components

This quick start guide describes how to use the Enterprise Console to discover existing AppDynamics components, such as a Controller, and use it to upgrade them.

Install the Enterprise Console

You can install the Enterprise Console on the same host as the Controller, or provide a separate host for installation.

1. Install the Enterprise Console with the following command:

   Linux

   ```
   ./platform-setup-64bit-linux.sh
   ```

   Windows

   ```
   platform-setup-64bit-windows.exe
   ```

   The installation wizard launches. Complete the wizard to install the Enterprise Console. For more information about how to install the application, see Enterprise Console.
2. Verify that the Enterprise Console successfully installed by opening the GUI using the host and port you specified during installation:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.

Discover and Upgrade with the Enterprise Console GUI

The options presented are based on the state of your platform. If you have not created a platform, choose Discovery from the options presented. Complete the wizard to create a platform, add hosts, and discover the Controller and Events Service.

If you created a platform already, such as with the CLI, complete the following steps:

1. Identify the host(s) where the Controller and Events Service are installed.
2. On the **Credentials** page, add credentials to the platform if you are using remote hosts.

   ⓘ Remember to provide the private key file for the Enterprise Console machine when adding a credential.

3. Navigate to the Hosts page and add hosts.
   a. If the Controller and Events Service are installed on the same host as the Enterprise Console, select **Add Enterprise Console Host**.
   b. If the Controller and Events Service are installed on different hosts than the Enterprise Console, input the host(s). Select a credential for the host(s) and add credentials to the platform.

Discover and Upgrade with the CLI

1. Navigate to the directory where you installed the Enterprise Console.
2. Create a platform with the following command:

   ```
   bin/platform-admin.sh create-platform --name <platform_name> --installation-dir
   <platform_installation_directory>
   ```

3. Add credentials that the platform can use to access hosts with the following command:

   ```
   bin/platform-admin.sh add-credential --credential-name <name> --type ssh --user-name <username> --ssh-
   key-file <file_path_to_the_key_file>
   ```

   `<file path to the key file>` is the private key for the Enterprise Console machine. The installation process deploys the keys to the hosts.

4. Add hosts to the platform:

   ```
   bin/platform-admin.sh add-hosts --hosts localhost host_2 host_3 --credential <credential_name>
   ```

   At a minimum, make sure to bootstrap all of the hosts where the AppDynamics server-side components, namely the Controller and Events Service, are installed.

   > You may also use the loopback address '127.0.0.1' or the machine's actual hostname in place of 'localhost'.

Enterprise Console

Related pages:

- Install the Enterprise Console
- Administer the Enterprise Console

The Enterprise Console is the installer for the Controller and Events Service. You can use it to install and manage the entire lifecycle of new or existing on-premises AppDynamics Platforms and components. The application provides a GUI and command-line interface.

There is no customer-facing application for SaaS Controllers since they are managed by the AppDynamics Operations team.

If your Enterprise Console host goes down, it does not impact Controllers, Events Service, or High Availability (HA) pairs. Those services will continue to run independently of the application. You can then discover all platforms on a new Enterprise Console host without any impact on the components.

> ⚠ If the Enterprise Console is not available, the auto-failover option for HA pairs will also not be available when there is an issue with the primary Controller. Therefore, it is important to keep the Enterprise Console host in a healthy state.

Enterprise Console Details

The Enterprise Console encompasses management features, installation modes, and lifecycle monitoring.

Enterprise Console Features

The Enterprise Console allows you to perform the following tasks:

Multi-Platform Management

- Manage multiple platforms at the same time using the application

  > ⓘ The Enterprise Console does not require all services within a given platform to have the same major version number.

- Discover, install, and upgrade Controllers, Event Services, and MySQL nodes

  > ⚠ Note that all services on Windows machines must be installed on the Enterprise Console host when using the Enterprise Console since the application does not support remote operations on Windows.

HA-Lifecycle Management (Available on Linux only)

- Manage HA pair lifecycle without the use of the CLI based HA-toolkit or sudo privileges
- Perform failover management

Other Features

- Manage Controller and Events Service lifecycle
- Utilize GUI & CLI support
- Manage AppServer and database configurations

Platform Install Modes

There are two install types that you can use to deploy your platform:

Express Install

- The quickest way to get started for fresh Controller installations
- Install the Controller and an embedded Events Service on a single host

Custom Install

- Configure and customize user inputs and installation/data directories for the Controller and Events Service
- Install or upgrade single or HA Controllers and scaled-up Events Service in a distributed setup

> Controller HA pairs and Events Service clusters are not available on Windows machines through the Enterprise Console since the application does not support remote operations on Windows.
>
> You can manually install or upgrade multi-node Events Service clusters. See Install the Events Service on Windows or Upgrade the Events Service Manually.

- Add one or more Events Service nodes
- Discover & Upgrade older platform services

Lifecycle Monitoring

On the Platforms page, you can see all of your platforms, their statuses, and the statuses of their services. Once you have selected a platform to view, the screen is separated into different tabs:

Hosts

Hosts are the actual hardware devices that are connected to the platform. You can add, remove, or change the credentials of your hosts in this tab.

> You cannot add a host in a Windows Enterprise Console machine.

Controller

The Controllers page shows the primary and secondary roles of the Controllers and their MySQL nodes. The entire lifecycle operations of Controllers and MySQL nodes can be performed here. You can also see the External URL, which is the IP of the primary machine. Health statuses for the Controllers are also available. You can Add a Secondary Controller if you would like to create an HA pair, then initiate an HA failover if you want to trigger a failover. You can also start or stop a Controller, Upgrade a Controller and MySQL, and more.

Events Service

The Events Service page displays your Events Service cluster, which can be made using one to three machines. Again there is an entire lifecycle of operations you can do.

Credentials

Credentials are your host's usernames and private keys. They are required to SSH or connect to the hosts via system user name and private keys.

Jobs

All of the jobs that you perform on your platform can be seen on the Jobs page. It is a nice way to keep track of your jobs and also see which jobs have failed.

Configurations

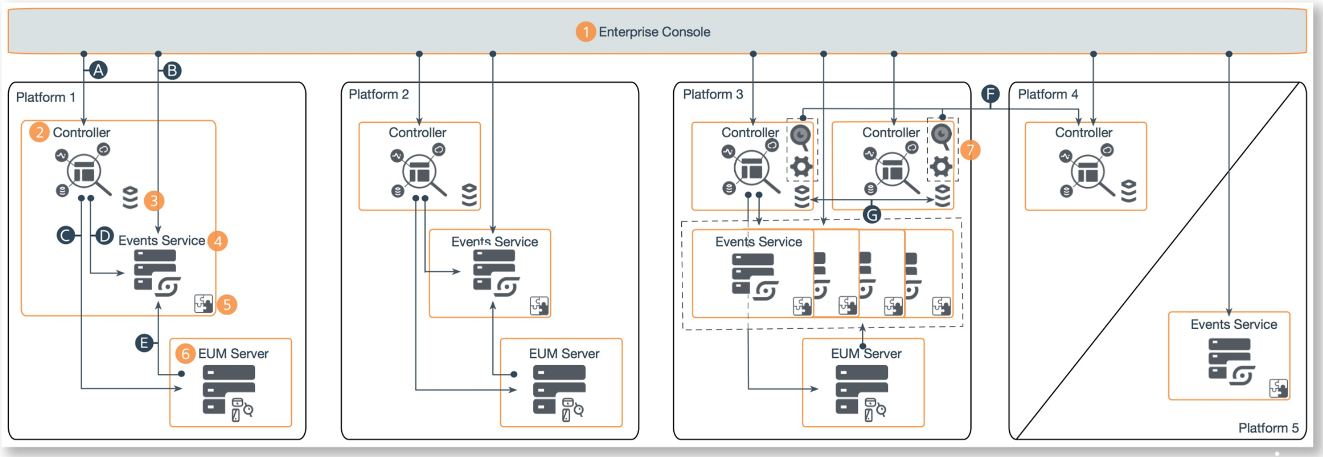
Configurations are important since they let you customize your installations. Configuration settings on the Enterprise Console are separated into three categories: Platform, Controller, and Events Service Settings.

The Controller Settings contains the most configurable settings. The AppServer Configurations under Controller Settings allows you to see all of the Domain configurations which you can initiate from this point or configure your ports. The Database Configurations lets you edit your MySQL settings. So you do not have to tweak the machine, you can do everything from the Console itself.

Enterprise Console Platforms Architecture

The following diagram depicts five platform examples that can be deployed and managed by the Enterprise Console.

> You cannot use the Enterprise Console to install the End User Monitoring (EUM) Server. Instead, you must use a package installer that supports interactive GUI or console modes, or a silent response file installation.



Depending on the scale of your deployment, your requirements, and the products you are using, your own application environment is likely to consist of a subset of the components shown in the diagram.

You can find the full On-Premises Deployment Architecture diagram on Application Performance Monitoring Platform, as well as a more detailed On-Premises and SaaS architecture diagram on PDFs.

Enterprise Console Platforms

The following table describes how the components work together in the above platforms.

Platform Number	Components Involved
Platform 1	Platform 1 depicts a single ② Controller with a local ④ Events Service and ⑥ EUM Server. The local Events Service contains an ⑤ API Store.
Platform 2	Platform 2 depicts a single ② Controller with a remote, single host ④ Events Service and ⑥ EUM Server. The remote Events Service contains an ⑤ API Store and can be expanded to a cluster by adding two or more machines.
Platform 3	Platform 3 depicts an HA ② Controller pair with a remote ④ Events Service cluster and ⑥ EUM Server. The Events Service cluster contains an ⑤ API Store on all nodes. The cluster must have three or more nodes.
Platform 4	Platform 4 depicts a single monitoring ② Controller. This Controller monitors the HA pair in platform 3 by receiving metrics via connection Ⓕ from the ⑦ App and Machine Agents. See Manage a High Availability Deployment for more information.
Platform 5	Platform 5 depicts a single shared ④ Events Service. A shared Events Service can connect to multiple ② Controllers from other platforms, minimizing required maintenance and cost. See Events Service Deployment for more information.

Enterprise Console Platform Connections

The following table lists and describes the traffic flow between the above components in the platforms.

Connection	Source	Destination	Traffic	Protocol	Default Port(s)
Ⓐ	① Enterprise Console	② Controller	Controller Health Checks / Controller Management	HTTP	8090
				HTTPS	8181

B	1 Enterprise Console	4 Events Service	Events Service API Store	Events Service Health Checks / Events Service Management	HTTP(S)	9080
			Events Service API Store Admin		HTTP(S)	9081
C	2 Controller	6 End User Monitoring (EUM) Server	EUM Metric Data		HTTP	7001
					HTTPS	7002 (demo mode only)
D	2 Controller	4 Events Service	Events Service API Store	Analytics Event Data	HTTP(S)	9080
			Events Service API Store Admin		HTTP(S)	9081
E	6 EUM Server	4 Events Service	Events Service API Store	EUM Event Data	HTTP(S)	9080
			Events Service API Store Admin		HTTP(S)	9081
F	7 App and Machine Agents	2 Controller	Monitoring Metric Data		HTTP	8090
					HTTPS	8181
G	3 MySQL Database	3 MySQL Database	MySQL Database Replication		TCP	3388

ⓘ There is no communication from the Controller to the Enterprise Console.

Enterprise Console Requirements

Related pages:

- Controller System Requirements
- Events Service Requirements

The Enterprise Console can run on the same host as the Controller and the embedded Events Service. If this is the case, the machine you choose to run the Enterprise Console must meet the requirements for all the components that run on that machine.

However, we recommend that you place the Enterprise Console on its own separate dedicated host, particularly if you deploy Controllers as High Availability pairs.

Supported Web Browsers

The AppDynamics Enterprise Console UI is an HTML 5-based browser application that works best with the latest version of any browser.

The Enterprise Console UI has been tested with and supports the last two versions of these browsers:

- Safari
- Chrome
- Firefox
- Microsoft Edge
- Internet Explorer

Certain types of ad blockers can interfere with features in the Enterprise Console UI. We recommend disabling adblockers while using the Enterprise Console UI.

CPU Requirements

The Enterprise Console is not CPU intensive and therefore can manage multiple platforms with two Cores.

Memory Space Requirements

The Enterprise Console requires an additional memory of one GB of free RAM for Java and MySQL processes.

Disk Space Requirements

The Enterprise Console requires ten GB of free space to install. After the Enterprise Console installation, there must be at least one GB of additional space on the Enterprise Console host in order to perform any operations, such as installing a remote Controller.

Network Protocol Requirements

The Enterprise Console requires SSH or Secure File Transfer Protocol (SFTP) to be properly configured and enabled for it to use remote hosts.

> ⚠ To access remote hosts, the Enterprise Console uses Java Secure Channel (JSch) API with the provided key file. The Enterprise Console does not support SSH Jump Server. If you use an SSH Jump Server, or have jump host configuration, please contact your AppDynamics representative for deployment options.

cURL

You must install cURL on systems that run Linux.

Required Libraries

Linux systems must include these libraries for Enterprise Console operation:

- `libaio`
- `numactl` package, which includes `libnuma.so.1` for RHEL, CentOS, and Fedora, and `libnuma1` for Ubuntu and Debian
- `glibc2.12`

  > This `glibc` version is included into a given operating system release, and therefore cannot be updated.

- `tzdata` for RHEL, CentOS, Fedora, , openSUSE Leap 12 and Leap 15, and Ubuntu version 16 and higher

  > The `tzdata` package is also required by the MySQL connector.

- `libncurses5` (and above) for Ubuntu, CentOS, Debian, openSUSE Leap 12 and Leap 15, and Amazon Linux 2

  > ⚠ As of MySQL 5.5.57 and 5.7.19, `libtinfo.so.5` is a required prerequisite library.

- `ncurses-libs-5.x` for RHEL and CentOS

  > ⚠ As of MySQL 5.5.57 and 5.7.19, `libtinfo.so.5` is a required prerequisite library.

- SLES12 and SLES15 using `libxml2-2` and `libxml2-tools`

This table provides instructions on how to install the libraries on some common flavors of the Linux operating system.

> ✓ If you cannot install the library, check that you have a supported version of your Linux flavor.

Linux Flavor	Command

- Red Hat
- CentOS
- CentOS Stream
- Amazon
	Use `yum` to install the library, such as:
	- `yum install libaio`
	- `yum install numactl`
	- `yum install tzdata`
	- `yum install ncurses-libs-5.x`
	
	Ensure that only one package `mgr` for `rpm` and is installed before running the Enterprise Console installer.
	
	For RHEL8, CentOS8, and Amazon2 you can either manually install version 5 of `ncurses` or use version 6.
	- To install version 5, follow these steps:
	1. `sudo rpm -ivh --force ncurses-base-5.x.rpm`
	2. `sudo rpm -ivh --force ncurses-libs-5.x.rpm`
	
	The `ncurses-libs` depends on the `ncurses-base` so you must install the `ncurses-base` first. These are examples of a trusted source for `rpm` download:
	- http://mirror.centos.org/centos/7/os/x86_64/Packages/ncurses-base-5.9-14.20130511.el7_4.noarch.rpm
	- http://mirror.centos.org/centos/7/os/x86_64/Packages/ncurses-libs-5.9-14.20130511.el7_4.x86_64.rpm
	
	- To install version 6, follow these steps:
	You must either create symlinks for `ncurses-libs-5` which points to `ncurses-libs-6`, or install the `ncurses-compat-libs` package, to provide ABI version 5 compatibility.
	
	RHEL8 symlink:
	`sudo ln /usr/lib64/libtinfo.so.6.1 /usr/lib64/libtinfo.so.5`
	`sudo ln /usr/lib64/libncurses.so.6.1 /usr/lib64/libncurses.so.5`
	
	CentOS8 symlink:
	`sudo ln /usr/lib64/libtinfo.so.6.1 /usr/lib64/libtinfo.so.5`
	`sudo ln /usr/lib64/libncurses.so.6.1 /usr/lib64/libncurses.so.5`
	
	Amazon2 symlink:
	`sudo ln -s /usr/lib64/libncurses.so.6.0 /usr/lib64/libncurses.so.5`
	`sudo ln -s /usr/lib64/libtinfo.so.6.0 /usr/lib64/libtinfo.so.5`
	
	RHEL8 compat-libs:
	`sudo yum install -y ncurses-compat-libs`
	CentOS8 compat-libs:
	
	`sudo yum install -y ncurses-compat-libs`
	
	Amazon2 compat-libs:
	`sudo yum install -y ncurses-compat-libs`
	
	Use the following prerequisites to install on CentOS Stream:
	- `sudo yum install -y libaio.x86_64`
	- `sudo yum install -y numactl.x86_64`
	- `sudo yum install -y tzdata`
	- `sudo yum install -y ncurses-compat-libs.x86_64`

Fedora	Install the library RPM from the Fedora website: - `yum install libaio` - `yum install numactl` - `yum install tzdata`

Ubuntu	Use `apt-get`, such as:
	• `sudo apt-get install libaio1`
	• `sudo apt-get install numactl`
	• `sudo apt-get install tzdata`
	• `sudo apt-get install libncurses5`
	⚠ Ensure that only one package `mgr` between `dpkg` and `rpm` is installed before running the Enterprise Console installer. This `pkg` manager utility will be used to verify mandatory `pkgs` before the Enterprise Console installation.
	ⓘ For Ubuntu20 you can install `libncurses5` or `libncurses6`.
	• If you choose `libncurses5`:
	`sudo apt-get install libncurses5`
	• If you choose `libncurses6`:
	`sudo apt-get install libncurses6`
	Note: For `libncurses6` you need to create symlink for `libncurses5` pointing to `libncurses6`.
	`sudo ln -s /usr/lib/x86_64-linux-gnu/libncurses.so.6.2 /usr/lib/x86_64-linux-gnu/libncurses.so.5`
	`sudo ln -s /usr/lib/x86_64-linux-gnu/libtinfo.so.6.2 /usr/lib/x86_64-linux-gnu/libtinfo.so.5`

Debian	Use a package manager (such as APT) to install the library (as previously described in the Ubuntu instructions).

openSUSE Leap 12 and Leap 15	Use `zypper` to install the library, such as:
	- `sudo zypper install libaio`
	- `sudo zypper install libnuma1`
	- `sudo zypper install tzdata`
	- `sudo zypper install libncurses5`
	For openSUE Leap 15, you can install `libncurses5` or `libncurses6`.
	- If you choose `libncurses5`:
	`sudo zypper install libncurses5`
	- If you choose `libncurses6`:
	`sudo zypper install libncurses6`
	Note: For `libncurses6` you need to create symlink for `libncurses5` pointing to `libncurses6`.
	`sudo ln /lib64/libncurses.so.6.1 /lib64/libncurses.so.5`
	`sudo ln -s /lib64/libtinfo.so.6.1 /lib64/libtinfo.so.5ncurses-compat-libs`
	Ensure that only one package `mgr` for `rpm` and is installed before running the Enterprise Console installer. Also, you need to add the openSUSE machine repository before installing the `tzdata` package.
	`For openSUSE Tumbleweed run the following as root:`
	`zypper addrepo https://download.opensuse.org/repositories/home:amshinde/openSUSE_Tumbleweed/home:amshinde.repo`
	`zypper refresh`
	`zypper install tzdata`
	
	`For openSUSE Leap 42.1 run the following as root:`
	`zypper addrepo https://download.opensuse.org/repositories/home:amshinde/openSUSE_Leap_42.1/home:amshinde.repo`
	`zypper refresh`
	`zypper install tzdata`
	
	`For openSUSE 13.2 run the following as root:`
	`zypper addrepo https://download.opensuse.org/repositories/home:amshinde/openSUSE_13.2/home:amshinde.repo`
	`zypper refresh`
	`zypper install tzdata`
	
	`For openSUSE 13.1 run the following as root:`
	`zypper addrepo https://download.opensuse.org/repositories/home:amshinde/openSUSE_13.1/home:amshinde.repo`
	`zypper refresh`
	`zypper install tzdata`
	
	`You may run into file conflicts when two packages attempt to install files with the same name but different contents. If you choose to continue, the old files and their contents will be replaced.`
	
	`See the openSUSE website (https://software.opensuse.org/download.html?project=home%3Aamshinde&package=tzdata) to manually download and install the tzdata package.`

SLES12 and SLES15	Use `zypper` to install the library, such as: - `sudo zypper install libxml2-2` - `sudo zypper install libxml2-tools` - `sudo zypper install libaio1` - `sudo zypper install numactl` - `sudo zypper install libcurses5` - `sudo zypper install tzdata`

See Platform Requirements for operating system support information.

High Availability Requirements

You must install `rsync` if you plan on deploying a Controller (HA) pair. In addition, when using SSH or an SSH client, note that OpenSSH 5.3p1 is the minimum version supported by the Enterprise Console for HA.

Supported SSH Key Exchanges, Cipher Algorithms, MAC and Host Key Type

You can use these `ssh` key exchanges, cipher algorithms, MAC types, and host key types to customize the `ssh` configuration on your host(s):

Supported ssh	Details
Key Exchanges	• `diffie-hellman-group-exchange-sha1` • `diffie-hellman-group1-sha1` • `diffie-hellman-group14-sha1` • `diffie-hellman-group-exchange-sha256` • `ecdh-sha2-nistp256` • `ecdh-sha2-nistp384` • `ecdh-sha2-nistp521`
Cipher Algorithms	• `blowfish-cbc` • `3des-cbc` • `aes128-cbc` • `aes192-cbc` • `aes256-cbc` • `aes128-ctr` • `aes192-ctr` • `aes256-ctr` • `3des-ctr` • `arcfour` • `arcfour128` • `arcfour256`
MAC Type	• `hmac-md5` • `hmac-sha1` • `hmac-md5-96` • `hmac-sha1-96`
Host Key Type	• `ssh-dss` • `ssh-rsa` • `ecdsa-sha2-nistp256` • `ecdsa-sha2-nistp384` • `ecdsa-sha2-nistp521`

Install the Enterprise Console

This page provides information and instructions for installing the AppDynamics Enterprise Console to automate the task of installing and administering the Controller and Events Service. You must install the Enterprise Console to install these components.

About the Enterprise Console Installation

Though the Enterprise Console can run on the same host as the Controller and, if installed, the embedded Events Service, it is recommended that you install it on a separate host. Regardless, the machine that runs the Enterprise Console must meet the requirements for all the components that run on that machine, as outlined below.

The Enterprise Console and Controller must run on separate MySQL instances allowing the Enterprise Console to manage the Controller's instance independent of the Controller host, creating a lightweight setup that consumes less memory.

> ⚠ If you install the Enterprise Console on the Controller machine, it must be in a different directory in order to keep data separate. For instance, if the Controller is installed in `/opt/appdynamics/controller`, the Enterprise Console might be in `/opt/appdynamics/enterpriseconsole`.

You must also avoid port conflicts with the Controller database, which is 3388 by default, whereas the Enterprise Console database is 3377 by default.

The Enterprise Console installation path you choose must be writeable, i.e. the user who installed the Enterprise Console should have write permissions to that directory.

The Enterprise Console prevents multiple users from running commands at the same time. If a second user attempts to run a command while another command is in progress, the second command is not completed and an error message appears indicating that another command is in progress. To avoid such conflicts, the Enterprise Console should generally be used by a single user at a time.

You can enable HTTPS for the Enterprise Console during installation. See HTTPS Support for the Enterprise Console.

> ⚠ Cross-platform (OS) installation, e.g., installing the Enterprise Console on Linux and the Controller on Mac or Windows is not supported.

Installing on AWS Host

When installing the Enterprise Console on an AWS host, you must add the values for the following hostnames and IP addresses to the SAN:

- Public DNS (IPv4)
- IPv4 Public IP
- Private DNS
- Private IPs

Disk and Memory Space Requirements

The host must have enough disk space for the Enterprise Console and a platform, which includes a Controller. There is no need for additional memory for the Enterprise Console when it shares the same host as the Controller. However, when the Enterprise Console host is not shared with the Controller host, then it requires additional disk and memory space. See Enterprise Console Requirements and Prepare the Controller Host to make sure you meet the minimum space requirements.

Software Requirements

On systems that run Linux, you must have cURL and `netstat` installed. Linux systems must also have the `libaio` library installed. This library provides for asynchronous I/O operations on the system.

See Required Libraries for how to install `libaio` and other libraries on some common flavors of the Linux operating system.

Password Requirements

Due to browser incompatibilities, AppDynamics recommends using only ASCII characters for usernames, account names, and passwords

Configure your installation using these password settings:

GUI Mode Screen or Option Label	**Response File variable**	**Description**
Database Root User's Password	`mysqlRootUserPassword`	The password of the user account that the Controller uses to access its MySQL database. Do not use the single quotation mark ('), double quotation mark ("), or at sign (@) characters in this password.
Controller root User's Password	`rootUserPassword`	The Controller root user password. The root user is a Controller user account with privileges for accessing the system Administration Console. This password is used for the admin user of the built-in Glassfish application server as well. The Glassfish admin user lets you access the Glassfish console and the `asadmin` utility. See Access the Administration Console. Allowed characters in the password are: a-z, A-Z, 0-9, ., +, =, @, _, -, $, :, #, ,, (,), !, {, }
User Name (Admin User Setup)	`userName`	The username of the administrator account in the Controller UI. This is the administrator for the built-in account if single-tenant systems, or for the initial account for multi-tenant. See Update the Root User and Glassfish Admin Passwords. Usernames and passwords cannot include the @ or ! character. Also note that if this account will be used to access the REST API, additional limitations on the use of special characters in usernames apply. See Create and Manage Tenant Users.

In the password field, you may include the space character at the beginning as well as in the middle of the password string. However, you cannot start passwords with the space character when using the `response.varfile`.

GUI Installation

Before starting, get the Enterprise Console installer version appropriate for your target system. You can get the installer from the AppDynamics Downloads. When ready, follow these steps to install the Enterprise Console:

1. Navigate to the directory where you downloaded the install file.
2. Run the following commands:

   Linux

   ```
   ./platform-setup-64bit-linux.sh
   ```

   You can run the installer as non-root or root.

   Windows

   ```
   platform-setup-64bit-windows.exe
   ```

   It is recommended that you right-click the .exe file and select **Run as Administrator**.

3. After the GUI launches, use it to complete the installation. In Linux, you may also follow the steps in the installation wizard to complete the console installation.

   If you install the Enterprise Console on AWS, use the public DNS for the Enterprise Console hostname when prompted.

Silent Installation

To use the silent installation method, add the -q option, the response file, and the destination directory to the command to run the installer. For example, in Linux, run the following command:

```
./platform-setup-64bit-linux.sh -q -varfile ~/response.varfile
```

It is recommended that, if possible, you provide an absolute path as the installation path specified as the `dir` argument value and not a relative path as shown in the example.

For a Windows system:

```
platform-setup-64bit-windows.exe -q -varfile c:/response.varfile
```

Sample response files for silent installation

Linux

```
serverHostName=HOST_NAME
sys.languageId=en
disableEULA=true

platformAdmin.port=9191
platformAdmin.databasePort=3377
platformAdmin.dataDir=/opt/appdynamics/platform/mysql/data
platformAdmin.databasePassword=ENTER_PASSWORD
platformAdmin.databaseRootPassword=ENTER_PASSWORD
platformAdmin.adminPassword=ENTER_PASSWORD
platformAdmin.useHttps$Boolean=false
sys.installationDir=/opt/appdynamics/platform
```

The `sys.languageID` and `platformAdmin.dataDir` properties are optional. If not specified, the data directory will be in the `/mysql` directory under the platform directory.

Windows

```
serverHostName=HOST_NAME
sys.languageId=en
disableEULA=true
sys.adminRights$Boolean=true

platformAdmin.port=9191
platformAdmin.databasePort=3377
platformAdmin.dataDir=C\:\\AppDynamics\\Platform\\platform-admin\\mysql\\data
platformAdmin.databasePassword=ENTER_PASSWORD
platformAdmin.databaseRootPassword=ENTER_PASSWORD
platformAdmin.adminPassword=ENTER_PASSWORD
platformAdmin.useHttps$Boolean=false
sys.installationDir=C\:\\AppDynamics\\Platform
```

The `sys.languageID` and `platformAdmin.dataDir` properties are optional. If not specified, the data directory will be in the `\mysql` directory under the platform directory.

> ⓘ If you install the Enterprise Console on AWS, use the public DNS for the `serverHostName` value.

After the Installation

After you install the Enterprise Console, you can use the following methods to install the AppDynamics Platform:

- GUI: A graphical interface within a web browser to install the Controller and Events Service. You can select from Express Install or Custom Install of the platform, which includes the option to install a Controller and Events Service.
- Command-line: A CLI to install the Controller and Events Service.

After installing the Enterprise Console, you can select from the Express Install or Custom Install of the platform, which includes the option to install a Controller. For more information about those options, see Enterprise Console.

For information on installing the Controller or Events Service in unattended mode or via the command line, see Enterprise Console Command Line.

Accessing the Enterprise Console

Access the GUI for the Enterprise Console with the following URL:

```
http(s)://<hostname>:<port>
```

Specify the port and hostname you used when you installed the Enterprise Console. The default port is 9191. This port needs to be exposed from your firewall rules so you can access the port from any place. See Port Settings.

For example:

```
http(s)://aHost.aDomain:9191
```

With the GUI, you can install and manage the components of the AppDynamics platform, including tasks such as adding hosts or credentials, installing a Controller, and monitoring jobs.

If you cannot access the GUI, verify that the hostname and port number are correct. Additionally, ensure that the Enterprise Console is running.

The first time you access the GUI, the Enterprise Console shows the following options for installing the AppDynamics Platform:

- **Express:** Select this option for new installations of the Controller and Events Service. The services are installed on the same host.
- **Custom:** Select this option to customize your installation, including installing or upgrading the Controller and Events Service on separate hosts. By installing the Events Service on a separate host, you can create a 1 or 3+ node Events Service based on your needs. Installing an Events Service on a separate host with the Enterprise Console is only supported on Linux. If you want to install the Events Service on a separate host on Windows, see Install the Events Service on Windows.

> ⓘ The Events Service is installed by default with a Custom Installation unless you choose to unselect the Install Events Service option.

  - **Discover and Upgrade:** When performing a custom installation, you have the option to discover and upgrade an existing AppDynamics deployment, such as a Controller or Events Service. For example, if you use the package installer to install the Controller in a previous version of AppDynamics, you can use the discover and upgrade option to add the Controller to the AppDynamics platform that the Enterprise Console manages. The application will then upgrade the Controller to the same version of the Enterprise Console. Verify that the Controller and MySQL are running before you attempt to discover and upgrade them.

Troubleshooting the Installation

This section provides troubleshooting information for issues that may arise during the Enterprise Console installation.

Installation Stuck at License Agreement

If your installation becomes stuck at displaying the license agreement on the console, then the EULA may be having issues with special characters. To fix this issue, add the `-VdisableEULA=true` flag to your installation command or `response.var` file. For example:

```
./platform-setup-64bit-linux.sh -c -VdisableEULA=true
```

Enterprise Console Application Is in Use Error

If you get an error that states that the "Enterprise Console Application (9191/3377) is in use," you should check that you have specified the correct hostname during the installation.

Default Font Change on Linux Machines

If your Enterprise Console installation fails on a Red Hat system, it may be due to an install4j issue. If the default font has been changed, the JRE cannot interpret it, leading to a "could not display the GUI" error. You can fix this error by running the installation with `-VdisableEULA=true` and creating the file `/etc/fonts/local.conf` with the following contents:

```
<?xml version='1.0'?>
<!DOCTYPE fontconfig SYSTEM
'fonts.dtd'>
<fontconfig>
  <alias>
    <family>serif</family>

<prefer><family>Utopia</family></prefer>
  </alias>
  <alias>

<family>sans-serif</family>
    <prefer><family>Utopia</family></prefer>

</alias>
  <alias>
    <family>monospace</family>

<prefer><family>Utopia</family></prefer>
  </alias>
  <alias>

<family>dialog</family>
    <prefer><family>Utopia</family></prefer>

</alias>
  <alias>
    <family>dialoginput</family>

<prefer><family>Utopia</family></prefer>
  </alias>
</fontconfig>
```

Rename the Directory Error on Windows Machines

If the Enterprise Console installation fails with an error on "rename of the directory," it may be due to an antivirus scan. Stopping the antivirus scan on the machine fixes the issue. You should also exclude the Enterprise Console directory from the scan if it sits outside of the Controller directory. See Prepare Windows for the Controller.

Express Install

Related pages:

- Controller System Requirements
- Events Service Requirements

You can choose Express Install on the Install page when you first use the Enterprise Console or when you want to create a new platform. This install option provides you with a quick and simple way to install a fresh single node Controller and embedded Events Service.

This page guides you through the steps to perform an Express Install. Before you install the Controller with the Enterprise Console, verify that the Enterprise Console is running and the host machine meets the requirements for the Controller.

If you use the GUI to install the Controller, you can create the platform at the same time. If you use the command line, you must create the platform prior to installing the Controller. For more information, see Administer the Enterprise Console. Express Install does not give you the option to choose the version you would like to install, and instead automatically installs the latest version of each service. If you would like more control over your installation, see Custom Install.

Install the Platform Using GUI

Express Install is the quickest way to get started with setting up your own AppDynamics Platform. You can use it to install the Controller and an Events Service on a single host.

After you install the Enterprise Console, you can complete the platform installation process with the GUI:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
3. Select **Express Install** to install a Controller and Events Service on a shared host.
4. Enter a Name and the Installation Path for your platform.

   > The Installation Path is an absolute path under which all of the platform components are installed. The same path is used for all hosts added to the platform. Use a path which does not have any existing AppDynamics components (Controller, Events Service, etc.) installed under it.
   > The path you choose must be writeable, i.e. the user who installed the Enterprise Console should have write permissions to that folder. Also, the same path should be writeable on all of the hosts that the Enterprise Console manages.
   >
   > Example path: `/home/appduser/appdynamics/product`

5. Add a host by entering host machine-related information: Host Name, Username, and Private Key. To add the Enterprise Console host, click **Add Enterprise Console Host,** which will automatically populate the text field with the hostname of the Enterprise Console machine. You do not need to provide a credential. The Controller and Events Service will be installed on the same host as the Enterprise Console.

   > If the Controller is to be installed on a Windows machine, the Enterprise Console should be on the same machine. This is because Windows hosts are not supported on the Enterprise Console.

   > If you use the Enterprise Console host to install the Controller, then no Username or Private Key is required. You can also install the Controller and Events Service on a remote host. In that case, the Username and Private Key of the remote host would be required.
   >
   > As of OpenSSH version 7.8, the `ssh` keys generated are in OpenSSH format using Ed25519. However, the Enterprise Console expects the `ssh` keys to be formatted using the older `pem` format.

6. Install the Controller:
   a. Select a Profile size for your Controller. See Controller System Requirements for more information on the sizing requirements.
   b. Enter the required Username and Passwords. The default Controller Admin Username is admin.
7. Click **Install**.

After clicking Install you can monitor the status of your platform creation jobs on the Jobs page, which include Add Hosts, Controller Install, and Events Service Install Jobs. Note that the Controller Install Job takes a considerably longer time to complete than the other two jobs. When the jobs successfully complete, you can check the status of your platform, obtain the URL of the Controller, update platform configurations, and manage the lifecycle of your services.

See Install the Events Service on Linux for additional setting requirements and to learn how to scale up an embedded Events Service.

Custom Install

Related pages:

- Controller System Requirements
- Events Service Requirements

The Controller is the central component of the AppDynamics platform. Other components such as the Events Service connect to the Controller and stream metrics to be displayed. The Enterprise Console Custom Install option provides you with a configurable way to install a fresh Controller and Events Service. You can also Discover & Upgrade older platform services using Custom Install.

This page describes how to use Custom Install to install the Controller. Before you install the Controller with the Enterprise Console, verify that the Enterprise Console is running and the host machine meets the requirements for the Controller.

If you use the GUI to install the Controller, you can create the platform at the same time. If you use the command line, you must create the platform prior to installing the Controller. For more information, see Administer the Enterprise Console. If instead, you would like to get started with your platform as soon as possible, see Express Install.

Install the Controller Using GUI

Installing a fresh Controller, HA-pair, or Events Service cluster is simple with Custom Install. You can use it to configure your platform deployment freely.

After you install the Enterprise Console, you can complete the platform installation process with the GUI:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
3. Navigate to the **Install** homepage and click **Custom Install**.
4. Enter a name and installation path for your platform.

   > ⓘ The Installation Path is an absolute path under which all of the platform components are installed. The same path is used for all hosts added to the platform. Use a path which does not have any existing AppDynamics components installed under it.
   > The path you choose must be writeable, i.e. the user who installed the Enterprise Console should have write permissions to that folder. Also, the same path should be writable on all of the hosts that the Enterprise Console manages.
   >
   > Example path: `/home/appduser/appdynamics/product`

5. Add a host by entering host machine-related information: Host Name, Username, and Private Key. This is where the Controller and Events Service will be installed onto.

   > ⓘ As of OpenSSH version 7.8, the `ssh` keys generated are in OpenSSH format using Ed25519. However, the Enterprise Console expects the `ssh` keys to be formatted using the older `pem` format.

   For more information about how to add credentials and hosts, see Administer the Enterprise Console.

   > ⚠ If the Controller is to be installed on a Windows machine, the Enterprise Console should be on the same machine. This is because Windows hosts are not supported on the Enterprise Console.

6. Install Controller:
   a. Select **Install**.
   b. Select an available Target Version from the dropdown list.

      > ⓘ The list is populated by versions that the Enterprise Console is aware of. This means that you can install the Controller to any intermediate version or to the latest version as long as the Enterprise Console installer has been run for those versions.

   c. Select a Profile size for your Controller. See Controller System Requirements for more information on the sizing requirements.
   d. Enter the Controller Primary Host.
   e. Enter the Controller Secondary Host if you would like to install an HA pair.

> ⓘ It is highly recommended that you have the Enterprise Console on its own separate dedicated host, especially in the case of HA installations. This means you would need three hosts for the recommended HA setup. If the Enterprise Console and Controller are on the same host and that host becomes unavailable, the Enterprise Console will not be able to failover to the other Controller.
> You can set up an HA pair at a later time. See Set Up a High Availability Deployment for additional setting requirements and to learn how to add a secondary controller after your initial installation.

f. Optional: Enter the Tenancy Mode.
g. Under Advanced, enter additional information for your HA pair.
h. Enter the required Username and Passwords. The default Controller Admin Username is admin.

> ⓘ If you do not install a Controller at this time, you can always do so later by navigating to the **Controller** page in the GUI and clicking Install Controller.

7. Install Events Service:
   a. Select **Install**.
   b. Select the Profile size for your Events Service. See Controller System Requirements for more information on the sizing requirements.
   c. Optional: Enter the Installation and Data Directory.

> ⚠ You do not need to specify the installation or data directory for the Events Service installation. If you do, use a different one from the platform or mysql data directory.

   d. Optional: Enter the Elastic Search, REST API Admin, REST API, and Unicast Ports.
   e. Enter the Events Service Host. You can deploy a single Events Service node or an Events Service cluster made up of three or more nodes. The minimum size of an Events Service cluster is three. You can always add more at a later time on the Events Service page.

> ⓘ You can set up a scaled up Events Service at a later time. See Install the Events Service on Linux for additional setting requirements and to learn how to scale up an embedded Events Service.

> ⓘ If you do not install an Events Service at that time, you can always do so later by navigating to the **Events Service** page in the GUI and clicking Install Events Service.

8. Click **Install**.

After clicking Install you can monitor the status of your platform creation jobs on the Jobs page, which include Add Hosts, Controller Install, and Events Service Install Jobs. Note that the Controller Install Job takes a considerably longer time to complete than the other two jobs. When the jobs successfully complete, you can check the status of your platform, obtain the URL of the Controller, update platform configurations, and manage the lifecycle of your services.

Once any high availability controllers are installed, they can be found under the Controller page. All high availability lifecycle operations such as start, stop, upgrade, and failover can be performed from this page.

Administer the Enterprise Console

You can use the GUI or command line to perform the following platform administration tasks with the Enterprise Console:

- Create new platforms
- Remove existing platforms
- Manage credentials
- Manage hosts

> Some tasks may not be available through both the GUI and command line. Most of the commands described here are based on the Linux command line and cover common tasks for managing the platform. You can run similar commands with the Windows command prompt by replacing the `platform-admin.sh` script with `platform-admin.exe cli`.

Run the commands from the `<Enterprise Console installation directory>/platform-admin` directory. This page contains the minimum options and parameters required to run a command.

Some commands may have more options and parameters. To see these additional options, run the command with `-h` specified. For example, run the following command to see all the options and parameters for the create a platform command:

Start or Stop the Enterprise Console

The Enterprise Console must be running before you can perform other tasks or use the GUI.

> The same user who installed the Enterprise Console should be the same user who starts or stops the Enterprise Console.

Use the following commands to start or stop the Enterprise Console:

To stop the Enterprise Console, replace `start` with `stop`.

Manage Platforms

The platform is the collection of AppDynamics components and their hosts. The Enterprise Console supports up to 20 platforms at a time by default.

Create a Platform

To use the Enterprise Console for end-to-end installation and management, you must first create a platform. The Enterprise Console creates the platform when you complete the Express or Custom installations or discover existing components in the GUI.

To use the command line to create a platform, run the following command:

The platform installation directory is the absolute directory where the Enterprise Console installs all AppDynamics components on all of its hosts. Once you add a host to the platform, you can no longer change the directory. Additionally, the directory cannot contain a space.

Delete a Platform

You can use the Enterprise Console to delete a platform that is no longer in use. You may also want to consider editing the Platform's configuration instead. You can perform either action on the Platform view page of the GUI.

To use the command line to delete a platform, run the following command:

> ⚠ If you just deleted your current platform, you must clear the value of the `APPD_CURRENT_PLATFORM` variable to prevent unexpected errors when running future commands.

Manage Credentials

Manage the credentials that the Enterprise Console uses to access and perform tasks on hosts, such as adding a node to an Events Service. You can use the Credentials page in the GUI or the command line to manage credentials. You should add credentials to the platform before you add hosts.

Generate an RSA Key Pair

An RSA private key file is required to add credentials to a platform. The following steps will generate an RSA key pair that consists of the public key file `~/.ssh/id_rsa.pub` and the private key file `~/.ssh/id_rsa`.

1. Log in to the Enterprise Console host machine via SSH.
2. Switch to the user that is the owner of the Enterprise Console:

   ```
   sudo -i -u <user-owner of the EC>
   ```

3. Create the RSA key pair:

   ```
   ssh-keygen -t rsa -b 2048 -N '' -m pem
   ```

4. Accept the default location for the key pair at `~/.ssh`.
5. Confirm that the RSA public and private key files have been created:

   ```
   ls ~/.ssh/
   id_rsa
   id_rsa.pub
   ```

Add Credential

When you add a credential, you need the following information:

- Credential name
- Username
- Private key file

The credential name is the unique identifier for a credential and is used to specify the credential when you perform tasks such as adding a host. AppDynamics recommends that you follow the naming convention for all of your credential names. The `id_rsa`, RSA private key, should be created using the OpenSSL PEM encoding format over the Open SSH standard encoding.

> ⓘ The Unix system user specified in the username field must have writeable access to the platform directory.

You can add a credential in the GUI by clicking **Add**.

```
bin/platform-admin.sh add-credential --credential-name <name> --type <ssh> --user-name <username> --ssh-key-file <file path to the key file>
```

Where `<file path to the key file>` is the private key for the Enterprise Console machine. The installation process deploys the keys to the hosts.

Remove Credential

Remove a credential that is no longer used. You cannot remove a credential that is still used by a host. You can remove a credential in the GUI by selecting the credential and clicking **Delete**.

```
bin/platform-admin.sh remove-credential --credential-name <name>
```

List Current Credentials

```
bin/platform-admin.sh list-credentials
```

Manage Hosts

The hosts are the machines used to run AppDynamics components such as the Controller and Events Service. For example, the Events Service can run on the same host as the Controller, a single host, or a cluster of three or more hosts.

You must properly configure the credential you use to add a new host on a remote host. This means that for a private key that you have specified, you must add the corresponding public key to the remote host `~/.ssh/authorized_hosts` file.

The Controller and Events Service must reside on the same local network and communicate by internal network. Do not deploy the cluster to nodes on different networks, whether relative to each other or to the Controller where the Enterprise Console runs. When identifying cluster hosts in the configuration, you will need to use the internal DNS name or IP address of the host, not the externally routable DNS name.

For example, in terms of an AWS deployment, use the private IP address such as 172.31.2.19 rather than public DNS hostname such as `ec2-34-201-129-89.us-west-2.compute.amazonaws.com`. You must then go to the Appserver Configurations under Controller Settings in the Enterprise Console GUI, and edit the external URL so you can access the page.

The host that runs the Enterprise Console is automatically created and added to the platform as the hostname of the Enterprise Console machine if you use the GUI to install or discover components. If you do not use the GUI, you must manually add this host.

Hosts can be managed through the Enterprise Console GUI on the Hosts page or the command line.

> The `id_rsa`, RSA private key, should be created using the OpenSSL PEM encoding format over the Open SSH standard encoding.

Set Up Remote Hosts

To set up seamless Enterprise Console communications with remote hosts, perform the following steps:

From the command line:

1. Set up the following passwordless SSH:
    - From the Enterprise Console to the Controller.
    - If HA then:
        - from the Enterprise Console to the primary Controller.
        - from the Enterprise Console to the secondary Controller.
        - From the primary Controller to the secondary Controller.
        - from the secondary Controller to the primary Controller.

From the Enterprise Console:

2. Identify the Enterprise Console Linux AppDynamics user's public/private key pair (usually in `~/.ssh`).

3. Add the Enterprise Console Linux AppDynamics user's public key (`~/.ssh/id_rsa.pub`) into the remote Controller server Linux user's `~/.ssh/authorized_keys`.

4. Then `chmod 600 /.ssh/authorized_keys`.

5. Test the SSH connection from the Enterprise Console server with the following:

```
ssh <remote user>@<remote-server> hostname
```

6. Verify that the remote hostname is printed. You may have to first answer yes to trust the server fingerprint.

7. Access the Enterprise Console UI Credential page, and add or edit the existing credentials.

8. Create a single credential, which will likely be the same for all remote hosts, with a name like: `EC-<ec linux user name>-<remote appd user name>` For example, `EC-ecappduser-appduser` , or `EC-appdyn` if the username is the same on the Enterprise Console and remote server.

9. Enter the remote server Linux username. It might be the same as the local Enterprise Console AppDynamics user.

10. Supply the Enterprise Console Linux user's `~/.ssh/id_rsa` contents as the private key as chosen in step 2.

Add Hosts

Before you add hosts to the platform, ensure that the required credentials are added to the platform.

> You should also check that the user ID created matches those in the credentials. Additionally, the path you specify for the platform base directory must exist.

You can add a host in the GUI by clicking **Add**.

```
bin/platform-admin.sh add-hosts --hosts host_1 host_2 host_3 --credential <credential name>
```

Instead of listing the hosts with --hosts, you can specify a text file with a line-separated list using the following command:

```
bin/platform-admin.sh add-hosts --host-file <file path to host file> --credential <credential name>
```

If you do not use the GUI, you must add the host for the Enterprise Console. This is also the host used by the Controller and embedded Events Service. The host is named "localhost" and does not require credentials. For example, run the following command:

> You may also use the loopback address '127.0.0.1' or the machine's actual hostname.

Remove Hosts

Before you remove a host, ensure that you remove all AppDynamics components from the host. You can remove a host in the GUI by selecting the host and clicking **Remove**.

```
bin/platform-admin.sh remove-hosts --hosts host_1 host_2 host_3
```

Instead of listing the hosts with `--hosts`, you can specify a text file with a line-separated list using the following command:

```
bin/platform-admin.sh remove-hosts --host-file <file path to host file>
```

If a host becomes unreachable, you can use the following command to remove it:

```
bin/platform-admin.sh remove-dead-hosts --hosts <host name>
```

This removes the host and all of its associated metadata from the Enterprise Console database.

> Running `remove-dead-hosts` could leave various services in an inconsistent state.

List Current Hosts

Update Host Credentials

Change the credential that the Enterprise Console uses to access hosts. You can change the host credential in the GUI by selecting the host and clicking **Change Credentials**.

```
bin/platform-admin.sh update-host-credential --hosts host_1 host_2 host_3 --credential <credential name>
```

Similar to the add and remove host commands, you can specify a text file instead of providing a list of hosts within the command.

Manage the Enterprise Console Admin User Password

Change the Admin User Password

You can change the password for the admin user with the following command:

Reset the Admin User Password

You can reset the password for the Enterprise Console's root user with the following command:

Resetting the password sets it to "admin".

Manage the Enterprise Console Database Root User Password

Change the Database Root User Password

To change the `platformAdmin.databaseRootPassword`, follow these steps:

1. Stop the Enterprise Console by running the following command in `<EC_home>/Platform/platform-admin/bin`:

   ```
   bin/platform-admin.sh stop-platform-admin
   ```

2. Run the following command in `<EC_home>/Platform/mysql/bin`:

   ```
   bin/mysqld --defaults-file="/<EC_home>/Platform/mysql/db.cnf" --skip-grant-tables
   ```

   Replace `<EC_home>` before running the command.
3. Open a new command prompt window.
4. Connect to the database without using a password by running the following command in `<EC_home>/Platform/mysql/bin`:

   ```
   bin/mysql -u root -h 127.0.0.1 -P 3377 --protocol=TCP
   ```

5. Execute the following queries:

   ```
   update mysql.user set authentication_string=password('<new_password_here>') where user like 'root%';
   flush privileges;
   quit;
   ```

   Replace `<new_password_here>` before executing the queries.
6. Quit the command prompt.
7. Stop the Enterprise Console Database by running the following command in `<EC_home>/Platform/platform-admin/bin`:

   ```
   bin/platform-admim.sh stop-platform-admin
   ```

   > ⚠ This is to stop the MySQL DB and start it without using the `--skip-grant-tables` option that is in the next step.

8. Start the Enterprise Console by running the following command in `<EC_home>/Platform/platform-admin/bin`:

   ```
   bin/platform-admin.sh start-platform-admin
   ```

9. Verify the login by running the following command in `<EC_home>/Platform/mysql/bin`:

```
bin/mysql -u root -p -P 3377 -h 127.0.0.1
```

Manage the Enterprise Console Database User Password

Change the platformAdmin Database User Password

To change the `platformAdmin.databasePassword`, follow these steps:

> ⓘ Use the application ID that owns the AppDynamics platform installation when running the commands.

1. Start the Enterprise Console, if it is not already running.
   Run the following command, replacing `<EC_home>` before running the command:

```
<EC_home>/platform/platform-admin/bin: ./platform-admin.sh start-platform-admin
```

2. Navigate to the `<EC_home>/platform/mysql/bin` directory.
3. Log in to the database as the root user by running the following command:

```
./mysql -u root -p -h 127.0.0.1 -P 3377 --protocol=TCP
```

4. Execute the following queries, replacing `<new_password_here>` before executing the query.

```
update mysql.user set authentication_string=password('<new_password_here>') where user like
'platformadmin%'; flush privileges; quit;
```

5. Verify the login by running the following command in the `<EC_home>/Platform/mysql/bin` directory:

```
./mysql -u platformadmin -p -P 3377 -h 127.0.0.1
```

6. Navigate to the `<EC_home>/platform/platform-admin/bin` directory.
7. Run the command below to encrypt the new password that was set for `platformadmin` in step 4.

```
./platform-admin.sh encrypt --text '<new_password_here>'
```

   Take note of the encrypted password.
8. Navigate to the `<EC_home>/Platform/platform-admin/conf directory.`
9. Backup the `PlatformAdminApplication.yml` file.
10. Using a text editor, such as `vi`, open the `PlatformAdminApplication.yml` file and find the encrypted `password:` line in the `database:` section.
11. Update the encrypted password, replacing the text after the `password:` entry with the password noted after step 7 and save the changes.
12. Stop the Enterprise Console by running the following command in the `<EC_home>/Platform/platform-admin/bin directory`:

```
./platform-admin.sh stop-platform-admin
```

13. Start the Enterprise Console by running the following command in the `<EC_home>/Platform/platform-admin/bin directory`:

```
./platform-admin.sh start-platform-admin
```

14. Validate that the Enterprise Console started successfully.

Change the Installation Directory

The installation directory you specify when creating the platform is used to install all AppDynamics components. This directory can be changed but requires that you uninstall all AppDynamics components.

1. Uninstall all AppDynamics components that you installed.
2. Remove all hosts from the platform, including the localhost. For example, run the following command to remove the localhost:

```
bin/platform-admin.sh remove-hosts --hosts localhost
```

3. Change the installation directory:

```
bin/platform-admin.sh update-platform --installation-dir <directory>
```

> The installation directory cannot contain a space

After you change the installation directory, you must add hosts and reinstall AppDynamics components.

Troubleshooting Administration Tasks

Check Availability of the Enterprise Console

You can use the following API to check the availability of the Enterprise Console:

`http://econsole-host:9191/service/version`

Error While Adding Hosts

While adding hosts to your platform, you may run into an "Enterprise Console host expansion failed" error. If you do, you should ensure that SFTP is enabled in your `sshd_config`, and restart the SSH service.

Remove Unused Artifacts

The purge clean up script (purge.sh) is available from Enterprise Console version 4.5.17 or later.

To remove unused artifacts from a prior Enterprise Console installation, you can run a purge clean up script (purge.sh). The purge.sh script preserves any artifacts required to run and upgrade the components in your existing platform.

- The purge clean up script must be run with the same user who installed the Enterprise Console.
- Verify that there are no jobs currently running from the Enterprise Console.

From the CLI:

1. Use the plan command to show the purge script execution plan.

   ```
   <platform-admin>/pa-purger/purge.sh plan
   ```

   The execution plan shows you the exact commands to run when you use the apply command.
2. Use the apply command to run and apply the commands of the purge script execution plan.

   ```
   <platform-admin>/pa-purger/purge.sh apply
   ```

3. Restart the Enterprise Console.

Old JREs that are not being used by any product services (Events Service or Controller) are cleared during the implementation of job upgrade-orcha which is ran when the Enterprise Console is upgraded. You can use the upgrade-orcha job on the adhoc basis to clean up old JREs.

```
#linux
./platform-admin.sh upgrade-orcha
#windows
platform-admin cli upgrade-orcha
```

Enterprise Console Command Line

The Enterprise Console command line utility allows you to perform orchestration tasks in an automated way. It is designed with the following limitations in mind:

- There is not a complete match of all the functionalities provided by the web UI.
- The command line utility is only supported to run on the host where the Enterprise Console is installed.

Command Line Directory

The platform-admin.sh|bat script in the `<Enterprise Console home>/platform-admin/bin` directory on the host machine provides a set of commands to install and manage the AppDynamics platform.

To see the operations available for the Enterprise Console, from the command line, navigate to directory and run the script with -h specified:

And you can view the format for each command in the command line by specifying the -h argument for a specific command:

You can also use `bin/platform-admin.sh list-jobs --service <controller or events-service>` to see a list of jobs available for the provided service. You can then see what parameters are required for the provided job using `bin/platform-admin.sh list-job-parameters --job <job_name> --service <controller or events-service>.`

Not all commands available on Linux are available on Windows. Refer to the list of the Enterprise Console commands displayed with the `-h` parameter.

The Enterprise Console prevents multiple users from running commands at the same time. If a second user attempts to run a command while another command is in progress, the second command is not completed and an error message appears indicating that another command is in progress. To avoid such conflicts, the Enterprise Console should generally be used by a single user at a time.

Logging into and out of the Enterprise Console

Commands for logging in and out of the Enterprise Console are:

- `login --user-name <admin_username> --password <admin_password>`
  If it has been more than one day since your last session, you will have to log in before you are able to use the command line utility. You will also have to log in again if you see the following error message:

  ```
  error: Command failed due to an error: Unauthorized
  API code 401
  Session expired. Please login and run the command again.
  ```

- `logout`

Managing the Password of the Enterprise Console

Commands for resetting or changing your Enterprise Console password are:

- `reset-password`
  If you forget your admin password or run into a 401 error, run this command to reset your password to its default value, `admin`. You will need to log out then log back in for this change to take effect.
- `change-password --user-name <username> --password <old_password> --new-password <new_password>`

Starting and Stopping the Enterprise Console

Just as you can start and stop the Controller and Events Service, you can start and stop the Enterprise Console process. Commands for starting and stopping the Enterprise Console are:

- `start-platform-admin`

- `stop-platform-admin`

The Enterprise Console must be running to install or administer Events Service nodes.

Getting Platform Versions

You can get the version of the Enterprise Console and any other components that you installed with the Enterprise Console.

To get the version of the Enterprise Console:

To get the version of one of the platform components:

Setting the Current Platform

The optional parameter, `--platform-name`, can be passed in each of your commands to set the current working platform. However, you can set the environment variable, `APPD_CURRENT_PLATFORM`, so that you do not have to pass the current working parameter with each of your commands. You can set this variable using `setenv` or `export`. The Enterprise Console will pick up the value if it is present.

If you happen to provide the `--platform-name` parameter while `APPD_CURRENT_PLATFORM` is set, the value passed through the flag will override the environment variable.

All Platforms Flag

You can use the `--all` flag to denote that you want to modify all platforms with your command. For example, you can upgrade all platform binaries at once by running the following command:

Additional Enterprise Console Commands

The following is a list of frequently used Enterprise Console commands:

- `create-platform --name <platform_name> --description <description> --installation-dir <install_dir>`
- `delete-platform --name <platform_name>`
- `upgrade-orcha --platform-name <platform_name>`
  This command triggers an upgrade of the Orcha module binaries on all remote Orcha machines in the provided platform. The platform should not be running any jobs on its services when you run this command.
- `list-supported-services`
- `show-platform-admin-version`

Update Platform Configurations

Configurations are important since they let you customize your installations. The Enterprise Console enables you to configure these settings via GUI and CLI. However, note that there is limited support for updating service configurations through the CLI. Therefore, it is recommended that you use the GUI for updating configurations, especially for multi-line values.

Configuration settings on the Enterprise Console are separated into three categories: Platform, Controller, and Events Service Settings.

Platform Properties

The Platform configuration allows you to update platform description in the UI. Updating the Platform path is only allowed in the CLI.

Controller Settings

The Controller Settings pages allow you to tune your controller. You can configure settings such as database configurations, JVM options, listeners, and thread pools, for both single or high availability controllers.

Appserver Configurations

The AppServer Configurations page under Controller Settings allows you to edit most of the domain.xml configurations. You can also change the ports and update the controller from a smaller to a higher profile. The configurations are categorized under Basic, JVM Options, and SSL Certificate Management:

Basic

- Profile: Demo, small, medium, or large

  You can change the Controller profile from a smaller profile to a larger one. Before doing so, ensure that the host machine meets the requirements for the profile size you want to use.

  > The Enterprise Console checks the disk size for the transaction log dir and db data dir for medium and large profiles only. If the transaction log is in a separate mount, then it will check for half of the minimum recommended disk size.

  This process is not reversible, and you cannot move from a larger to a smaller profile size. If you tune the Controller heap settings or database configuration settings, even to be greater than the recommended settings for the new profile, those settings will be preserved. Otherwise, the AppDynamics recommended settings are applied.
  To increase the Controller profile size, navigate to AppServer Configurations by choosing the platform, **Configurations**, **Controller Settings**, and **Appserver Configurations**. At the top of the page, select a new profile, then click **Save**.
  Alternatively, you can also use the CLI to increase the Controller profile size to meet increased demand:

  ```
  bin/platform-admin.sh update-controller-profile --profile <profile size>
  ```

  For more information, see Controller System Requirements.
- Tenancy Mode
- External Load Balancer URL (HA only): This is the deep link URL.
- Internal Virtual IP Address (HA only)
- Ports: Server Port: 8090; Admin Port: 4848; SSL Port: 8181; IIOP Port: 3700; JMS Port: 7676
  To change the Controller ports, navigate to AppServer Configurations by choosing the platform, **Configurations**, **Controller Settings**, and **Appserver Configurations**. Near the top of the page, specify new ports and scroll down to click **Save**. This will restart the Controller. Note that the new ports should be available.
- Glassfish Admin Password
- Database User Password
  If you do not specify this password then it will use the default. Click How to Change the Controller Database Root User Password for detailed steps.

  > After a fresh installation or upgrade, the database user password will be hidden in `domain.xml` in the Appserver directory as an alias.

  Alternatively, you can also use the CLI to change your database user password:

For MySQL DB:

```
bin/platform-admin.sh submit-job --platform-name <platform_name> --service controller --job update-
passwords --args newDatabaseUserPassword=<password>
```

- Advanced configurations
    - NUMA Controller Configuration: This setting is preserved upon upgrades.
    - NUMA Database Configuration: This setting is preserved upon upgrades.

JVM Options

- JVM Options: You can update the JVM options via this page without having to use the `modifyJvmOptions` utility or any other external scripts.
- Domain Http Services
- Domain Protocols
- Domain Network Listeners

> ⚠ Disabling the HTTPS listener is not allowed.

- Domain Transports
- Domain Thread Pools

You can also update the domain config settings using the CLI by following the steps below:

1. Download all four configurations to individual files. See Deploy the Controller on AWS for more information on the config file.
2. Create and load four variables:
    - `` new_network=`cat domain-network-listeners.txt` ``
    - `` new_protocol=`cat domain-protocols.txt` ``
    - `` new_thread=`cat domain-thread-pools.txt` ``
    - `` new_transports=`cat domain-transports.txt` ``
3. Go to `platform-admin/bin` and log in.

```
cd platform-admin/bin
```

```
./platform-admin.sh login --user-name=admin --password=password
```

4. Run the following command on the Enterprise Console host:

```
./platform-admin.sh update-service-configurations --service controller --job update-configs --args
domainProtocols="$new_protocol" domainTransports="$new_transports" domainNetworkListeners="$new_network"
domainThreadpools="$new_thread"
```

SSL Certificate Management

- You can view and edit the SSL Certificate here.

Controller Database Configurations

You can make database configuration changes using:

- Database Configuration UI Page
- Enterprise Console CLI

Database Configuration UI Page

Use the Database Configuration UI page to edit your MySQL settings. This is helpful since you do not have to tweak the configuration file on the database host.

If your RAM memory is greater than 200 GB and you are using a NUMA based architecture, you can specify the Linux nodes (typically CPU socket numbers) from which both processes and memory will be allocated for each AppDynamics component. For example on a two-socket motherboard, AppDynamics recommends the following node configuration settings:

- Glassfish should should allocate its threads/processes and memory on the first node: 0
- MySQL should allocate its threads/processes and memory on the second node: 1

> For the node configuration settings, you can enter an integer or comma separated list of integers. For example, for Glassfish, you can enter **0** or **0,1**; for MySQL, you can enter **1** or **2,3**.

The configurations include:

- DB Configuration Settings
  Data Directory: You can change the `datadir` path and database port via this page.

  > You cannot change certain configurations, such as the MySQL root directory, through the Enterprise Console.

- DB Root Password
  The Enterprise Console does not allow you to change the MySQL root password. However, if you change the MySQL root password for the Controller, you should update the database root password in the Database Configurations page so that the Enterprise Console is aware of the new password.

Enterprise Console CLI

You can update the Controller database configuration programmatically using the Enterprise Console CLI. This enables you to preserve configuration settings during an upgrade.

> - If you are using High Availability Toolkit (HATK), you must manually apply these settings on the secondary Controller and restart the secondary server.
> - These instructions are specific to the UNIX operating system.

To update the database configuration using CLI:

1. Copy the `db.cnf` file from your primary Controller host onto the Enterprise Console host, for example `db.cnf.new` file.
2. Edit the `db.cnf.new` file to add new settings or update existing values.
3. Load the `db.cnf.new` file into an environment variable:

   ```
   new_db_cnf=`cat db.cnf.new`
   ```

4. Go to the `platform-admin/bin` directory and log in:

   ```
   ./platform-admin.sh login --user-name=admin --password=password
   ```

5. Run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh submit-job --service controller --job db-update-config --args
   mysqlCnfContent="$new_db_cnf"
   ```

Reports Service Configurations

The Reports Service Configurations page allows you to update the reports service ports:

- Reporting Service HTTP Port: 8020
- Reporting Service HTTPS Port: 8021

You can also view and edit the SSL Certificate here.

Events Service Settings

The Events Service configurations are read-only. See Administer the Events Service to learn how you can manage your Events Service. You can see the following configurations:

- Profile: This value is either Dev or Prod.

- Installation Directory
- Data Directory
- Ports
    - Elastic Search Port: 9200
    - REST API Port: 9080
    - REST API Admin Port: 9081
    - Unicast Port: 9300

Update Controller Configurations

You can update Controller configurations such as the deep link URL, JVM options, and network listeners.

Update the Deep Link URL

To update the deep link URL:

1. Go to `platform-admin/bin` and log in.

    ```
    cd platform-admin/bin
    ```

    ```
    ./platform-admin.sh login --user-name=admin --password=password
    ```

2. Run the following command on the Enterprise Console host:

    ```
    ./platform-admin.sh update-service-configurations --service controller --job update-configs --args
    controllerExternalUrl=<server-protocol>://<controller-host>:<controller-port>
    ```

    where `server-protocol` is http or https.

Update JVM Options

To update JVM options and network listeners:

1. Go to `platform-admin/bin` and log in.

    ```
    cd platform-admin/bin
    ```

    ```
    ./platform-admin.sh login --user-name=admin --password=password
    ```

2. List all of the Controller configurations from the Enterprise Console:

    ```
    ./platform-admin.sh list-service-configurations --service controller > controller-configs.conf
    ```

3. Open `controller-configs.conf`, and copy all JVM options. Then paste them into a separate file, and edit the desired parameters.
4. Run the following command on the Enterprise Console host:

    ```
    ./platform-admin.sh update-service-configurations --service controller --job update-configs --args
    controllerNonHostJvmOptions='All JVM options from the previous step including the updated fields'
    ```

    For example:

    ```
    ./platform-admin.sh update-service-configurations --service controller --job update_configs --args
    controllerNonHostJvmOptions='-Djava.awt.headless=true, -Djdk.corba.allowOutputStreamSubclass=true, ...'
    ```

Retaining Configuration Changes

The Enterprise Console recognizes and retains many common customizations to the `domain.xml`, `db.cnf`, and other configuration files, but is not guaranteed to retain them all. If you have made manual configuration changes to the files, verify the configuration after updating.

You can also remove the Controller from the Enterprise Console and rediscover it to preserve the configuration changes:

1. Go to `platform-admin/bin` and log in.

   ```
   cd platform-admin/bin
   ```

   ```
   ./platform-admin.sh login --user-name=admin --password=password
   ```

2. On the Controller page, click on **Remove Controller**, or run the following commands on the Enterprise Console host:

   > ⓘ If `removeBinaries=false` then the Enterprise Console forgets the Controller without impacting or uninstalling the Controller.

3. Discover the Controller by using the **Discover & Upgrade** feature as if you were upgrading the Controller using the Enterprise Console, or run the following command on the Enterprise Console host:

   > ⓘ You must specify and provide the full path to the existing Controller directory.

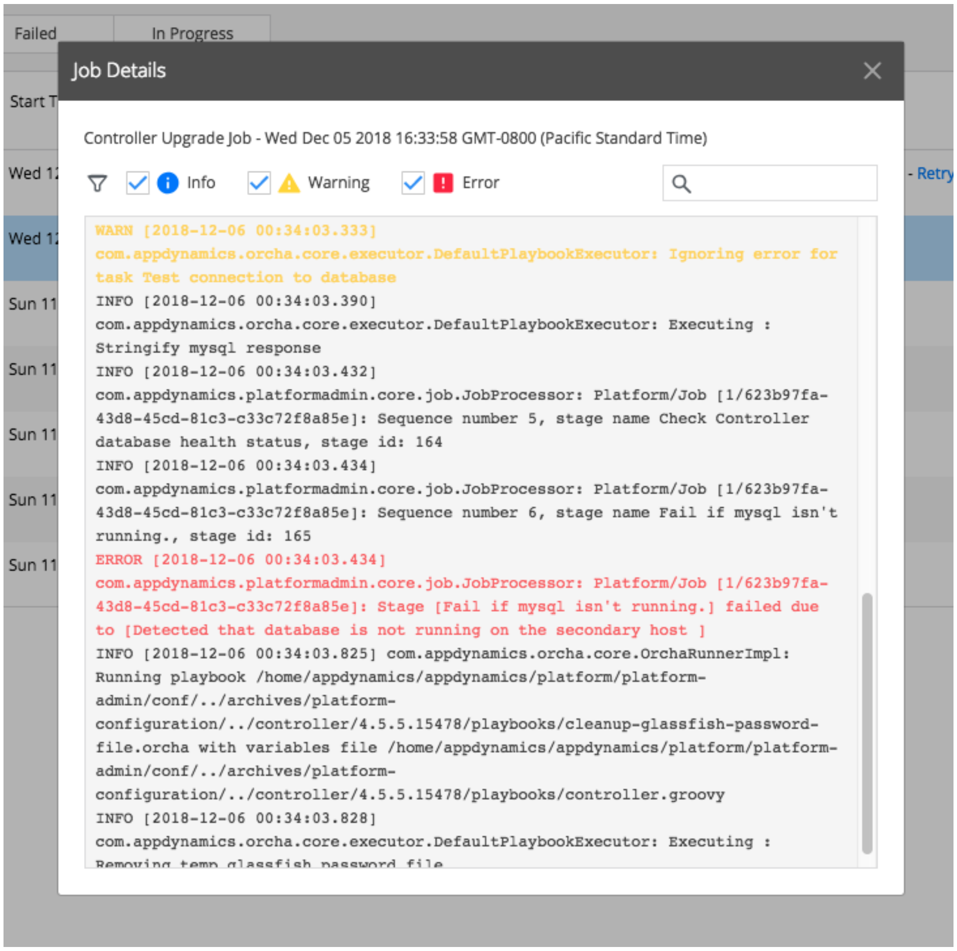
Enterprise Console Jobs

The Enterprise Console saves all jobs that you perform on the application on the Jobs page.

View Job Details

The Jobs page displays all of your jobs by their names, start times, last updated times, and statuses. You can classify the jobs based on their status: Successful, Failed, and In Progress. You can also search through all of your jobs.

You can view additional job details by clicking **View Details** in the Status column of the job.



Controller Upgrade Job - Wed Dec 05 2018 16:33:58 GMT-0800 (Pacific Standard Time)

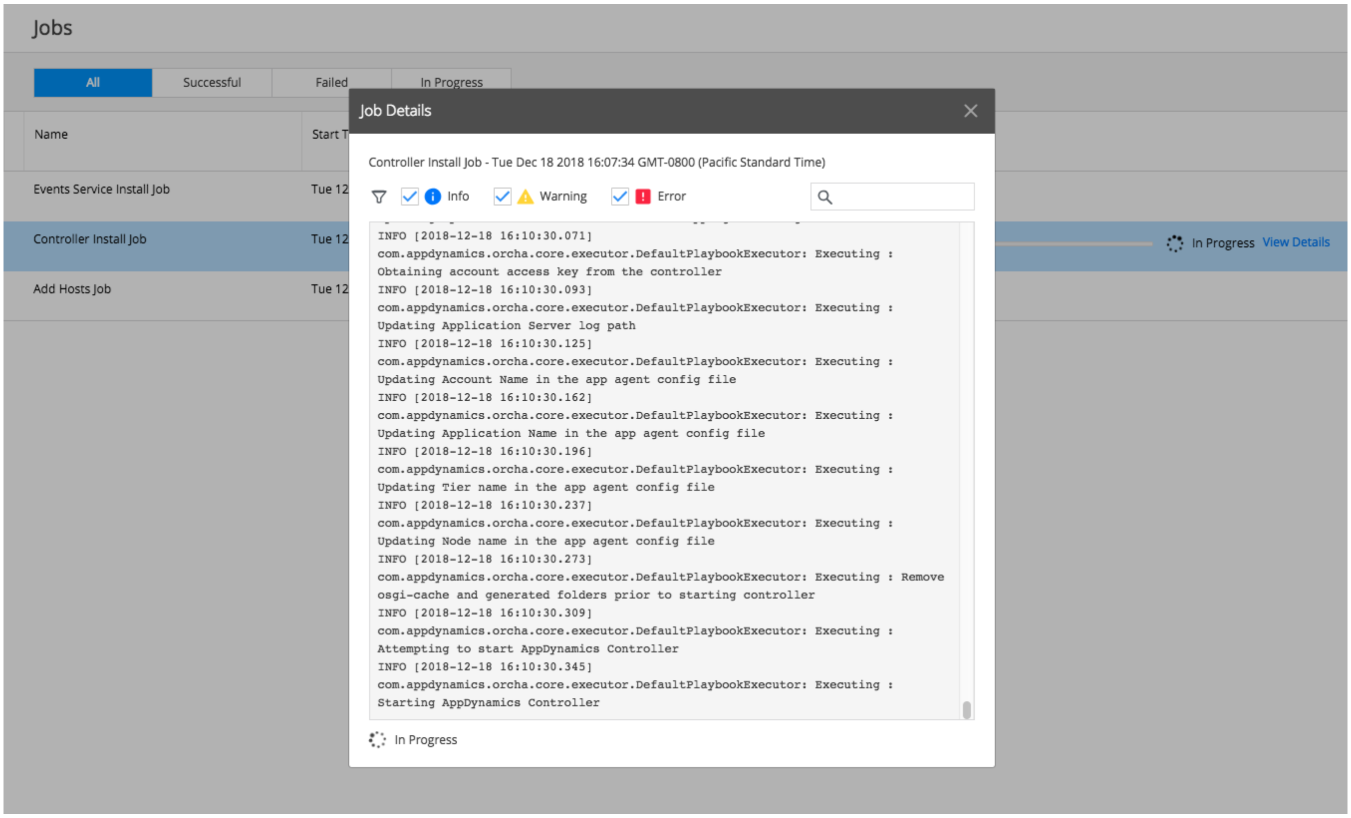
```
WARN [2018-12-06 00:34:03.333]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Ignoring error for
task Test connection to database
INFO [2018-12-06 00:34:03.390]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Stringify mysql response
INFO [2018-12-06 00:34:03.432]
com.appdynamics.platformadmin.core.job.JobProcessor: Platform/Job [1/623b97fa-
43d8-45cd-81c3-c33c72f8a85e]: Sequence number 5, stage name Check Controller
database health status, stage id: 164
INFO [2018-12-06 00:34:03.434]
com.appdynamics.platformadmin.core.job.JobProcessor: Platform/Job [1/623b97fa-
43d8-45cd-81c3-c33c72f8a85e]: Sequence number 6, stage name Fail if mysql isn't
running., stage id: 165
ERROR [2018-12-06 00:34:03.434]
com.appdynamics.platformadmin.core.job.JobProcessor: Platform/Job [1/623b97fa-
43d8-45cd-81c3-c33c72f8a85e]: Stage [Fail if mysql isn't running.] failed due
to [Detected that database is not running on the secondary host ]
INFO [2018-12-06 00:34:03.825] com.appdynamics.orcha.core.OrchaRunnerImpl:
Running playbook /home/appdynamics/appdynamics/platform/platform-
admin/conf/../archives/platform-
configuration/../controller/4.5.5.15478/playbooks/cleanup-glassfish-password-
file.orcha with variables file /home/appdynamics/appdynamics/platform/platform-
admin/conf/../archives/platform-
configuration/../controller/4.5.5.15478/playbooks/controller.groovy
INFO [2018-12-06 00:34:03.828]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Removing temp glassfish password file
```

If the job you are viewing failed, then you can see what the error was that caused it to fail, and retry it at its last checkpoint.

> Warnings and errors are color-coded in yellow and red, respectively.

View Job Progress

You can view job details on the UI while the job is in progress by selecting a job and clicking **View Details**. This should help with troubleshooting.



Jobs

Name	Start T
Events Service Install Job	Tue 12
Controller Install Job	Tue 12
Add Hosts Job	Tue 12

Job Details

Controller Install Job - Tue Dec 18 2018 16:07:34 GMT-0800 (Pacific Standard Time)

☑ Info ☑ Warning ☑ Error

```
INFO [2018-12-18 16:10:30.071]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Obtaining account access key from the controller
INFO [2018-12-18 16:10:30.093]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Updating Application Server log path
INFO [2018-12-18 16:10:30.125]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Updating Account Name in the app agent config file
INFO [2018-12-18 16:10:30.162]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Updating Application Name in the app agent config file
INFO [2018-12-18 16:10:30.196]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Updating Tier name in the app agent config file
INFO [2018-12-18 16:10:30.237]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Updating Node name in the app agent config file
INFO [2018-12-18 16:10:30.273]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing : Remove
osgi-cache and generated folders prior to starting controller
INFO [2018-12-18 16:10:30.309]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Attempting to start AppDynamics Controller
INFO [2018-12-18 16:10:30.345]
com.appdynamics.orcha.core.executor.DefaultPlaybookExecutor: Executing :
Starting AppDynamics Controller
```

In Progress

Platform Log Files

The platform log files include the following:

- **platform-admin-server.log:** Information about events of the install process such as extraction, preparation, and other post-processing tasks. It is located at `<platform_admin_home>/logs`.
- **server.log:** Information for the embedded Glassfish application server used by the Controller. It is located at `<controller_home>/logs`.
- **audit.log:** Information about Account/User/Group/Role CRUD/User login/logout/SSH connections, and all other operations. This can be used to forward auditable events from the AppDynamics controller into a central log management system or SIEM. It is located at `<controller_home>/logs`, and replicates the Controller Audit Report.
- **database.log:** Information for the MySQL database that is used by the Controller. It is located at `<controller_home>/db/logs`.
- **startAS.log**: Output generated by the underlying Glassfish domain for the Controller.
- **orcha-modules.log**: Information about all the tasks and commands that are run during installation. platform-admin-server.log has high-level information about the task executed. However, `orcha-modules.log` has more information on the specific command and output. It is located at `<platform-dir>/orcha/<version>/orcha-modules/logs/orcha-modules.log`.

Retrieve Log Files

You can use the Enterprise Console to retrieve log files for the Controller and Events Service. On the Controller or Events Service page, expand **More** options and click **Retrieve Log** to start a Retrieve Log job. When the job has successfully completed, it will retrieve and save the Controller or Events Service log files, which include AppServer, Reports Service, and DB logs, as a zip file to `/home/appdynamics/appdynamics/platform/platform-admin` on the Enterprise Console host.

You can also run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --service <controller or events-service> --job retrieve-log --platform-name
<name_of_the_platform>
```

Manage Log Files

You can manage your log files by setting up log rotation and adjusting the retention periods.

Log Rotation

The application server is preconfigured to rotate the server.log file regularly, based on settings in the domain configuration file. On Windows, there is an additional log file for Glassfish service launcher that needs to be rotated.

For the other log files, such as database.log or audit.log, you may need to set up log rotation to prevent them from consuming excessive disk space. The audit.log file, in particular, may grow quickly because it contains SSH connections to Controller and Event Services hosts that occur every minute. You also need to set up log rotation for an additional log file, `<controller_home>/db/data/slow.log`. This log contains information about slow MySQL queries.

The tool you use to perform the rotation depends on your operating system. On Linux, you can use the `mysql-log-rotate` script. The script is included with the Controller database installation at `<controller_home>/db/support-files`. You need to modify the script for your environment since it is not set up to rotate the database.log file by default. On other systems, you need to create or install a script that performs log rotation and make sure that it get run regularly, for example, by cron or an equivalent task scheduler.

Retention Period

Enterprise Console logs in `platfom-admin/logs` are automatically archived in a .gz format with a date-time stamp after they grow too large. The size of each archived log is around 300 KB. Only the latest seven files are archived in the logs directory.

If you want to retain the archived logs for longer periods of time, you can configure the settings in the Dropwizard configuration file, `PlatformAdminApplication.yml`. Under the logging/appenders/type: file section, you need to specify a larger `archivedFileCount`, as well as `maxFileSize`. See the Dropwizard Configuration Reference for more information.

Change the Default Location or Names of the Controller Logs

1. In a web browser, log in to the Controller's Glassfish administration console, as described in Access the Administration Console.
2. From the left-side navigation tree, expand **Configurations -> server-config** and click **Logger Settings**.
3. Set the new location for server.log by modifying the **Log File** value.
   The default value points to the logs directory located at the root of the Controller home directory.

   ```
   ${com.sun.aas.instanceRoot}/../../../../logs/server.log
   ```

   If you specify a directory that does not exist, it is created when you restart the application server.
4. Change the `database.log` location by opening the `<controller_home>/db/db.cnf` file. You can also change the `db.cnf` file from the Enterprise Console GUI Database Configurations page. Doing so also restarts the database server.

   > The Enterprise Console takes a backup when you modify these configurations.

5. Set the value of the `log-error` property to the new location of the `database.log` file. This directory location must exist before you restart the Controller or you will get start-up errors.
6. From either Linux or Windows, change the log location to the new location:

   Linux File

   ```
   <controller_home>/bin/controller.sh
   nohup ./asadmin start-domain domain1 > $INSTALL_DIR/logs/startAS.log
   ```

   Windows File

   ```
   <controller_home>/bin/controller.bat
   call asadmin.bat start-domain domain1 > %INSTALL_DIR%\logs\startAS.log
   ```

7. Open the `<controller_home>/appserver/glassfish/domains/domain1/config/domain.xml` file, and change the log location specified in the following:
   - log-root attribute of the domain element.
   - file attribute of the log-service element.
   - tx-log-dir attribute of the transaction-service element.
8. Copy if desired any existing logs from the default directory (`<controller_home>/logs`) to the new location.
9. Restart the Controller. See Start or Stop the Controller.
10. Verify that the `database.log` and `server.log` files are being written to the new location and remove the old log files. The `database.log` location is `controller/db/logs/database.log`.

Log Level Granularity

The Controller logs provide information about possible errors in Controller operations. By default, the Controller writes to the log at the `INFO` level. When debugging your Controller deployment, you may need to increase the logging level to generate additional information.

You can set the logging level by Controller component, which include:

- Agents
- Business Transactions (BTS)
- Events
- Incidents
- Information Points (IPS)
- Metrics
- Orchestration
- Rules
- Snapshots

Change Default Logging Level by Component

By default, the Controller generates logs at the INFO level. You can change the level for one or all of the components. This may be needed, for example, when you are debugging your system, and want the Controller to generate more information in the form of logs. On the other hand, you may wish to reduce logging verbosity to minimize the reduce the rate of growth of log files. The following steps describe how to change the default log levels.

1. In a web browser, log in to the Controller's Glassfish administration console, as described in Access the Administration Console.
2. From the left navigation tree, expand **Configurations > server-config** and click **Logger Settings**.
3. Click the **Log Levels** tab.
4. Modify those components that start with "com.appdynamics". By sorting the list by name, you can quickly access the com.appdynamics components. For each component, modify the log level by choosing a new level from the **Log Level** menu. For example, to debug the system, we suggest setting the log level to `FINE`.
5. Click **Save**.

Enterprise Console Back Up and Restore

The Enterprise Console keeps all data pertaining to its managed AppDynamics platform deployment in a MySQL database. To back up an Enterprise Console installation, you use MySQL commands to export and restore data. You will also need to back up the Enterprise Console's secure credential store file.

> ⓘ Backing up Enterprise Console data is a separate consideration from backing up a Controller. For more information on Controller backups, see Controller Data and Backups.

In the event that your Enterprise Console host fails, follow these steps to ensure that you can recover.

To back up the Enterprise Console:

1. Export the Enterprise Console data:

   ```
   mysqldump -u root -p <password> -h <mysql-hostname> -P <mysql-port> platform_admin > /tmp
   /platform_admin_dump.sql
   ```

   This puts the export file into a named file in the `/tmp` directory. Choose another location, if appropriate.
2. Change to the `mysql/bin` directory of Enterprise Console:

   ```
   cd <platform>/mysql/bin
   ```

3. Import the data into the backup Enterprise Console instance:

   ```
   mysql -u root -p <db_root_user_password> -h <mysql-hostname> -P <mysql-port> platform_admin < /tmp
   /platform_admin_dump.sql
   ```

4. Copy the Enterprise Console's secure credential store (SCS) file `<platform>/platform_admin/.appd.scs` to the same directory on your backup Enterprise Console instance. Make sure you retain the same file permissions for the SCS file (644) on the backup instance of the Enterprise Console.

Upgrade the Enterprise Console

Related pages:

- Upgrade Platform Components
- Install the Enterprise Console

To upgrade the Enterprise Console, you run the installer for the version of the application to which you want to upgrade on the Enterprise Console machine. The installer detects the Enterprise Console installation and upgrades that instance.

About the Upgrade

- You can upgrade across multiple versions at a time; that is, you do not need to run the installer individually for each intermediate version.
- You can enable HTTPS for the Enterprise Console during upgrades. See HTTPS Support for the Enterprise Console for more information.

Before Upgrading

- Before you start upgrading the Enterprise Console and your platform, make sure that you are using the correct upgrade order.
- The user upgrading the Enterprise Console should be the same user who installed the Enterprise Console.
- Make sure your Enterprise Console is running before you run the upgrade. This is to validate the Database Root User Password and Platform Admin Database Password. You will need to input the passwords when upgrading through the GUI.
- For first-time upgrades, you need to use the same response file you used for your silent install. See Silent Installation.
- Irrespective of first time or subsequent upgrades, you need to provide the relevant passwords in the response file.

Upgrade the Enterprise Console

The following steps describe how to upgrade the Enterprise Console on Linux and Windows. You use the Enterprise Console installer in GUI mode, console mode, or silent mode to perform the upgrade.

GUI Installation

If there is a newer version of Enterprise Console available, you can begin the upgrade process by downloading and installing the latest version from the AppDynamics download site on top of the existing application.

Before starting, get the Enterprise Console installer version appropriate for your target system. You can get the installer from the AppDynamics download site. When ready, follow these steps to install the Enterprise Console:

1. Navigate to the directory where you downloaded the install file.
2. Run the following commands:

   Linux

   ```
   ./platform-setup-64bit-linux.sh
   ```

   > ⓘ You can run the installer as non-root or root.

   Windows

   ```
   platform-setup-64bit-windows.exe
   ```

   > ✓ It is recommended that you right-click the exe and select Run as Administrator.

3. After the GUI launches, use it to complete the installation. In Linux, you may also follow the steps in the installation wizard to complete the console installation.

> If you install the Enterprise Console on AWS, use the public DNS for the Enterprise Console host name when prompted.

Silent Installation

To use silent mode, pass the response file that the installer generated at first installation to the installer. This response file is at the following location

- `<Enterprise_Console_home_directory>/.install4j/response.varfile`

If you have made any changes to the settings as originally configured by the installer—such as to the connection port numbers, tenancy mode, or data directory—make the same change in the response file before starting the upgrade.

You must also add the existing passwords to the file:

```
platformAdmin.databasePassword= ENTER_PASSWORD
platformAdmin.databaseRootPassword= ENTER_PASSWORD
platformAdmin.adminPassword= ENTER_PASSWORD
```

If you do not, the installer prompts you to add the password.

After Upgrading

After upgrading the Enterprise Console, you must first clear your browser cache before you can successfully log in to the Enterprise Console through the browser.

Uninstall the Enterprise Console

This page describes how to remove the Enterprise Console software and associated files using the uninstaller utility located in the Enterprise Console directory.

Before Starting

Uninstalling the Enterprise Console will not uninstall any of the components it has deployed or managed since the Enterprise Console installer is agnostic of the Controller and other services. Therefore, you do not need to first uninstall any components before uninstalling the Enterprise Console. However, if you accidentally uninstall the Enterprise Console without first uninstalling any components, then you will have to manually manage the remaining platforms and components. Leftover environments can be also be rediscovered using another Enterprise Console application.

Uninstall the Enterprise Console Manually

1. Open a console on the machine where the Enterprise Console is installed:
   - On Linux, open a terminal window and switch to the user who installed the Enterprise Console or to a user with equivalent directory permissions.
   - On Windows, open an elevated command prompt by right-clicking on the Command Prompt icon in the Windows Start Menu and choosing **Run as Administrator**.
2. From the command line, navigate to the Enterprise Console home directory.
3. Execute the uninstaller script to uninstall the Enterprise Console, as follows:
   - On Linux:

     ```
     ./installer/uninstallPlatform
     ```

   - On Windows:

     ```
     run installer/uninstallPlatfom.exe
     ```

   - To uninstall in quiet mode, add the `-q` option. For example:

     ```
     ./installer/uninstallPlatform -q
     ```

     With this option, you do not need to interact with the installer to complete the removal.

     > On the Enterprise console host and any of the hosts that the Enterprise Console manages, `<platform home dir>/jre` and `<platform home dir>/orcha` are not removed.

FAQs for the Enterprise Console

What is the difference between the Enterprise Console installer and the Controller installer?

The Enterprise Console installer only installs the Enterprise Console application. You need to use the Enterprise Console later on to install the Controller as well as other AppDynamics platform components.

The Controller Installer is used to install only the Controller application. From 4.4, AppDynamics does not support the Controller installer anymore. You should use the Enterprise Console instead to install, monitor, upgrade, and configure the Controller.

Where should I install the Enterprise Console application?

The Enterprise Console can be installed on the same host as the Controller. However, you should not install the Enterprise Console on the same host if the Controller is part of an HA pair. If the Enterprise Console and Controller are on the same host and that host becomes unavailable, the Enterprise Console will not be able to failover to the other Controller. For large deployments, it is also recommended that you install the Enterprise Console on a separate host.

Which hosts does the Enterprise Console need SSH access to?

The Controller and Events Service hosts.

How many SSH connections does the Enterprise Console make per minute?

For each single Controller node (HA Controller deployments will have two Controllers), the Enterprise Console will open approximately 10 SSH connections per minute. For each Events Services node (an Events Service cluster may have 3–5 nodes), the Enterprise Console will open 1 SSH connection per minute.

What protocols does the Enterprise Console use to connect to remote hosts?

The Enterprise Console uses Java Secure Channel (JSch) API with the provided key file to access remote hosts. In scenarios where you have an SSH jump server or jump host configuration, you will have to invest in additional provisions for your application to work. Consult your AppDynamics representative in such cases.

Can platforms share the same installation path?

No, platforms cannot share the same installation path or hosts. You must also choose an installation path that does not overlap with where the Enterprise Console is installed.

Can I use Express installation to discover and upgrade existing controllers and events services?

Express installation is a convenient way to create a complete platform on a single host in one step. It does not support discovering and upgrading an existing controller managed outside the Enterprise Console.

Can I make configuration changes outside of Enterprise Console?

It is not recommended that you make changes directly to AppDynamics configuration files, such as in domain.xml or db.conf. If you do make changes directly in configuration files, it is recommended that you make the equivalent change in the Enterprise Console. Note that such changes result in a restart the controller.

Do I need to back up the Enterprise Console database?

Yes, the Enterprise Console database is not automatically backed up. See Controller Data Backup and Restore for information on backing up the Controller and Enterprise Console databases.

Is Enterprise Console supported on Windows?

Yes, it is, but there are several differences between the Linux and Windows deployment. First, the platform components (Controller, MySQL database, Events Service, and EUM Server) all have to be installed on the same machine for Windows. The second difference is that the Enterprise Console on Windows does not support Controller High Availability. Finally, on Windows, you do not have the replication of the MySQL database and cannot install a second node through the Enterprise Console. On Linux, it is recommended to have at least three node clusters for the Events Service.

If you need to scale up the Events Service on Windows, see Install the Events Service on Windows for instructions.

How can I install the Controller without the Enterprise Console?

Starting from AppDynamics version 4.4, you can only install the Controller through the Enterprise Console. For earlier versions, you can still use the Controller installer.

Controller Deployment

Related pages:

- Install the Controller Using the CLI

This page introduces you to the tasks involved with deploying AppDynamics to its operating environment, including host preparation and Controller installation.

The system resources of the machine that hosts the Controller in a live environment must be able to support the expected workload.

Deployment Overview

Installing AppDynamics to a test or evaluation setting typically involves verifying system requirements, preparing the host, and then performing the Controller installation. These topics are described in Prepare the Controller Host and Install the Controller Using the Enterprise Console.

Deploying the Controller to its production operating environment normally introduces additional requirements and considerations. Security, availability, scalability, and performance all play an important role in production deployment planning. The following section lists the tasks related to deploying the Controller.

Deployment Tasks

Depending on your specific requirements and environment, deployment tasks may include:

- Ensure that target systems meet the Controller System Requirements for the Controller's expected workload.
- Implement Controller High Availability to ensure service continuity in the event of a failure of the Controller server.
- Configure the network environment. If deploying the Controller with a reverse proxy, configure passthrough of Controller traffic. Also, note other Network Requirements for the deployment environment.
- Implement security requirements for your environment. If clients will connect to the Controller by HTTPS, install your custom SSL server certificate on the Controller.
- Generate a password management strategy for the built-in system accounts in the Controller and platform.
- Make sure the mail server is properly configured for the Controller in the target environment and define your alerting strategy.
- Devise your backup strategy. A typical backup strategy consists of frequent partial backups with intermittent full backups.
- Plan your configuration maintenance and enhancement strategy. Changes to the configuration should be staged in a non-critical environment, and rolled into the live environment only after thorough testing. The AppDynamics UI and REST API offer the ability to export and import configuration settings from various contexts.
- Deploying App Agents is likely to be an ongoing task, especially in dynamic environments where monitored systems are regularly taken down and new ones brought up. There are two basic strategies for deploying large numbers of App Agents across a managed environment:
    1. Deploy the agents independently of the application inside the application server. This method ensures that re-deployments of the application do not overwrite the agent deployment.
    2. Integrate deployment of AppDynamics agents into the deployment of applications. This more sophisticated approach requires modifying the existing application deployment automation scripts.

  For details, see:
    - Automate Agent Deployment
    - Unattended Installation for .NET

Network Requirements

Deploying the Controller often calls for configuration changes to existing network components, such as firewalls or load balancers in the network. If the Controller will reside behind a load balancer or reverse proxy, you need to set up traffic forwarding for the Controller. You may also need to open ports used by AppDynamics on firewalls or any other device through which traffic must traverse.

The following are general considerations for the environment in which you deploy AppDynamics. See AppDynamics Quick Start for other network configuration requirements.

Correlation HTTP Header

AppDynamics adds a custom header to traffic in the monitored environment named `singularityheader`. This header enables AppDynamics to correlate traffic across tiers. It's important to ensure that any load balancer, proxy, or firewall in the network between monitored tiers or between the tiers and the Controller preserves the header added by AppDynamics.

Clock Management

To ensure consistent event time reporting across the AppDynamics deployment, App Agents attempt to synchronize their time with the Controller time.

They do so by retrieving the time from the Controller every five minutes. App Agents then compare the Controller's time to its own local machine's clock time. If the times are different, whether ahead or behind, it applies a time skew based on the difference to the timestamps for the metrics it reports to the Controller.

If, despite the agent's attempt to report metrics based on the Controller time, the Controller receives metrics that are time-stamped ahead of its own time, the Controller rejects the metrics. To avoid this possibility, AppDynamics recommends maintaining clock-time consistency throughout your monitored environment.

Admin Accounts Created at Installation

During the installation process, you need to configure several accounts for the Controller. These include the embedded MySQL database account, a root user account in the Controller, and an administrator in the Controller.

Usernames and passwords should not include the & or ! characters. If a user account needs to access the Controller REST API, additional limitations on the use of special characters in usernames apply. See Create and Manage Tenant Users.

About Controller Tenancy Mode

In most installations, the Controller operates in single-tenant mode. In multi-tenant mode, the Controller UI context is divided into separate accounts. Each account has its own set of users, agents reporting to it, and application monitoring configuration.

You choose the tenancy mode at installation time. You can switch the tenancy mode from single-tenant to multi-tenant mode later. It is not possible to switch from multi-tenant to single-tenant mode.

Having a single tenancy Controller is suitable for most installations. Only very large installations or installations that have very distinct sets of users may require multi-tenancy.

A summary of the differences between the modes follows:

- In multi-tenant mode:
    - You can create multiple accounts (tenants) in the Controller.
    - Each account will have its own set of users and applications.
    - The Controller login page includes an additional field where users need to choose an account to log in to.
    - Essentially, multi-tenant mode allows you to partition users and access application data in a logical, secure way.
- In single-tenant mode:
    - There is only one account (tenant) in the Controller system.
    - All users and applications are part of this single built-in account, so all users have access to all monitored Applications in this mode.
    - The account is not exposed to users in the Controller UI. The account field in the login page is omitted for single-tenant mode.
    - AppDynamics recommends a single-tenant mode for most installations.

For more information, see Multi-Tenant Controller Accounts.

Self-Monitoring the Controller

You can use the system account to self-monitor the Controller. When your system experiences noticeable performance issues with the Controller, you can log in to the system account to access and review the memory utilization trend for the last few hours.

> ⓘ Only the internal Java Agent bundled in the path: `<Controller_home>//appserver/glassfish/domains/domain1/appagent(-javaagent:${com.sun.aas.instanceRoot}/appagent/javaagent.jar)` with the controller, the application is supported for self-monitoring. Using a custom Java Agent from a different path is not supported for self-monitoring.

To self-monitor the Controller using the system account:

1. If you are already logged in to the Controller from your Browser, you must first log out as a non-administrative user of the Controller.
2. Log back into the Controller using the following credentials:
    - Account: `system`
    - User: `root`
    - password: <*`root password`*>
3. Select **Appdynamics Controller** application.
4. Select **Tiers and Nodes** > **Node1** > **Memory** to review the memory trend for the past 24 hours and locate the time when the performance issues occurred.

Controller System Requirements

Related pages:

- Prepare the Controller Host

This page describes hardware and software requirements for the Controller hosted on private or public cloud to help you prepare for your AppDynamics deployment.

> ⚠ **Note**
>
> The Controller requirements do not include Enterprise Console and Event Service. You need to prepare memory for each of those components.

About Controller Sizing Information

Every deployment is unique. Factors such as the nature of the application, workload, and the AppDynamics configuration can all affect the resources required for your specific scenario. Be sure to test the performance of your system in a staging environment, so that you can fully understand your requirements before deploying AppDynamics to its live operating environment.

Before installation, it's usually easiest to estimate your deployment size based on the number of nodes. For Java, for example, a node corresponds to a JVM. However, the best indicator of the actual workload on your Controller is provided by the metric ingestion rate.

After initial installation, you should verify your Controller sizing using the metric upload rate. You then need to continue to monitor the Controller for changing workload brought about by changes in the monitored application, its usage patterns, or in the AppDynamics configuration.

General Hardware Requirements

The following general requirements that apply to the machine on which you install the Controller:

- The Controller should run on a dedicated machine. A production Controller *must* run on a dedicated machine. The requirements here assume that no other major processes are running on the machine where the Controller is installed, including no other Controllers.
- The Controller is not supported on machines that use Power Architecture processors, including PowerPC processors. The Controller is supported on amd64 / x86-64 architectures.
- Ensure that the Controller host has approximately 200 MB of free space available in the system temporary directory.
- Disk I/O is a key element to Controller performance, particularly low latency. See Disk I/O Requirements for more information.

Controller Sizing

The following table shows Controller installation profiles by metric ingestion rate and node count. As previously noted, the actual metrics generated by a node can vary greatly depending on the nature of the application on the node and the AppDynamics configuration. Be sure to validate your sizing against the metric ingestion rate before deploying to production.

General Sizing for On-Premises

Profile	Max Metrics /Minute	Max Agents (Approx)	OS	Compute	Storage
Demo	10,000	5	Linux or Windows	2 Cores, 8 GB RAM	50 GB Note: This profile is not supported when installing with Aurora DB.
Small	50,000	50	Linux or Windows	4 Cores, 16 GB RAM	400 GB Note: This profile is not supported when installing with Aurora DB.
Medium	1,000,000	1,500	Linux or Windows	Bare-metal: 8 Cores, 128GB RAM	5 TB SAS SSDs. Hardware-based RAID 5 configuration
			Linux or Windows	VM: 16 vCPUs, 128GB RAM	5 TB SSD-based SAN with 10 Gb/s FCoE

Large	5,000,000	10,000	Linux	Bare-metal: 28 cores, 512 GB RAM	• 2 x 800 GB write-intensive NVMe cards for MySQL redo logs. Software-based (mdadm) RAID 1 configuration. • 20 TB SAS SSDs for main data volume. Hardware-based RAID 5 configuration
Extra Large	**For deployments that exceed the Large profile configuration defined here, contact AppDynamics Professional Services for a thorough viability evaluation of your Controller.**				

Amazon Web Services (AWS) Sizing for On-Premises

AWS Profile with Aurora	Max Metrics /Minute	Max Agents (Approx)	OS	Compute	Instance Size for AWS Aurora Storage	Block Storage (for Controller application files only)*
Medium	1,000,000	1500	Linux	EC2: r4. 2xlarge	db.r4.4xlarge	10 GB GP2 EBS Volume. We recommend using a different volume than the instance's root volume.
Large	5,000,000	10000	Linux	EC2: r4. 8xlarge	db.r4.16xlarge	10 GB GP2 EBS Volume. We recommend using a different volume than the instance's root volume.

* The specified disk space must be available for use by the Controller. Specifications do not include overhead from the operating system, file system, and so on.

Elastic Network Interface (ENI)

The ENI numbers were last updated on Feb 28, 2018.

For AWS, provision an ENI for each Controller host and link the license to the MAC address of the ENI. For more information about ENI, see the AWS documentation at the following link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-eni.html.

Additional Sizing Considerations

Note the following additional requirements:

- Large installations are not supported on virtual machines or systems that use network-attached storage.
- The RAM recommendations leave room for operating system processes. However, the recommendations assume that no other memory intensive applications are running on the same machine. While the Enterprise Console can run on the same host as the Controller in small or demo profile Controllers, it is not recommended for medium and larger profiles or for high availability deployments. Refer to the Enterprise Console Requirements if the Enterprise Console is on the same host as the Controller.
- Disk sizing shown in the sizing table represents the approximate space consumption for metrics, about 7 MB for each metric per minute.
- The motherboard should not have more than 2 sockets.
- See Calculating Node Count in .NET Environments for information related to sizing a .NET environment.
- The agent counts do not reflect additional requirements for EUM or Database Visibility. See the following sections for more information.

Disk I/O Requirements

A critical factor in a machine's ability to support the performance requirements of a Controller in a production environment is the machine's disk I/O performance.

There are two requirements related to I/O latency:

- This disk I/O must perform such that the maximum write latency for the Controller’s primary storage must not exceed 3 milliseconds while the Controller is under sustained load. AppDynamics cannot provide support for Controller problems resulting from excessive disk latency.
- Self-monitoring must be set up for the Controller. Self-monitoring consists of a SIM agent that measures the latency of data partitions on the Controller host, and the configuration needs to include dashboard and health rule alerts that trigger when the maximum latency exceeds 3 ms. For details on Controller self-monitoring, contact your AppDynamics account representative.

Disk I/O Operations

The AppDynamics Controller performs two types of I/O operations important to Controller performance:

- The MySQL intent log is very sensitive to latency, and MySQL performs writes using varying block sizes.

- MySQL’s InnoDB storage engine uses random, asynchronous, 16Kb reads and writes to move database pages between storage and cache. In a properly sized Controller, most reads are satisfied from one of the software caches.

It’s important for best performance that the stripe size of the RAID configuration matches the write size. The two write sizes are 16Kb (for the database) and 128Kb (for the logs). You should use the smallest stripe size supported, but no smaller than 16Kb. If using a hardware-based RAID controller, be sure that it supports these stripe sizes. The stripe size can be determined by the number of data disks multiplied by the strip/segment/chunk (the portion of data stored on a single disk).

SAN-based Storage Limitations

While onboard disks typically satisfy I/O requirements, SAN-based storage could be hampered by poor I/O latency performance. In addition, AppDynamics discourages the use of an NFS-mounted filesystem. NFS adds latency and throughput constraints that can negatively affect Controller performance and even lead to data corruption. Similarly, you should avoid iSCSI or other SAN technologies that are subject to quality of service issues from the underlying network.

If you choose to deploy one of these latency-challenged storage technologies on a system that is expected to process 1M metrics/min or greater, a mirrored NVMe configured as a write-back cache for all storage accesses is recommended. Configuring such a device will hide some of the longer latencies that have been seen in these environments.

In all cases, be sure to thoroughly test the deployment with real-world traffic load before putting an AppDynamics Controller into a live environment.

End User Monitoring (EUM) Considerations

End User Monitoring (EUM) typically increases the number of metrics collected. Accordingly, the Small Controller profile is not supported for installations that use EUM. A Medium profile running 20+ high-traffic BRUM/MRUM agents should be sized at a specification closer to a Large profile for EUM.

Specifically, EUM impact metrics as follows:

- Web RUM can increase the number of individual metric data points per minute by up to 22000
- Mobile RUM can increase the number of individual metric data points per minute by as much as 15 to 25K per instrumented application if your applications are heavily accessed. The actual number depends on how many network requests your applications receive.
- Monitoring EUM is memory intensive and may require more space allocated to the metrics cache.

ⓘ The number of separate EUM metric *names* saved in the Controller database can be larger than the kinds of individual data points saved. For example, a metric name for a metric for iOS 5 might still be in the database even if all your users have migrated away from iOS 5. So the metric name would no longer have an impact on resource utilization, but it would count against the default limit in the Controller for metric names per application. The default limit for *names* is 200,000 for Browser RUM and 100,000 for Mobile RUM.

Database Monitoring Considerations

The following guidelines can help you determine additional disk and RAM required for the machine hosting the Controller that is monitoring the Database Agent. For very large installations, you should work with your AppDynamics representative for additional guidelines.

For on-premises installations, the machine running the Controller and Event Service will require the following additional considerations, for a data retention period of 10 days:

- 1 – 10 collectors: 2 GB RAM, Single CPU
- 10 – 20 collectors: 4 GB RAM, 2 CPUs
- More than 20 collectors: 8 GB RAM, 4 CPUs

Sizing the Controller for the Events Service

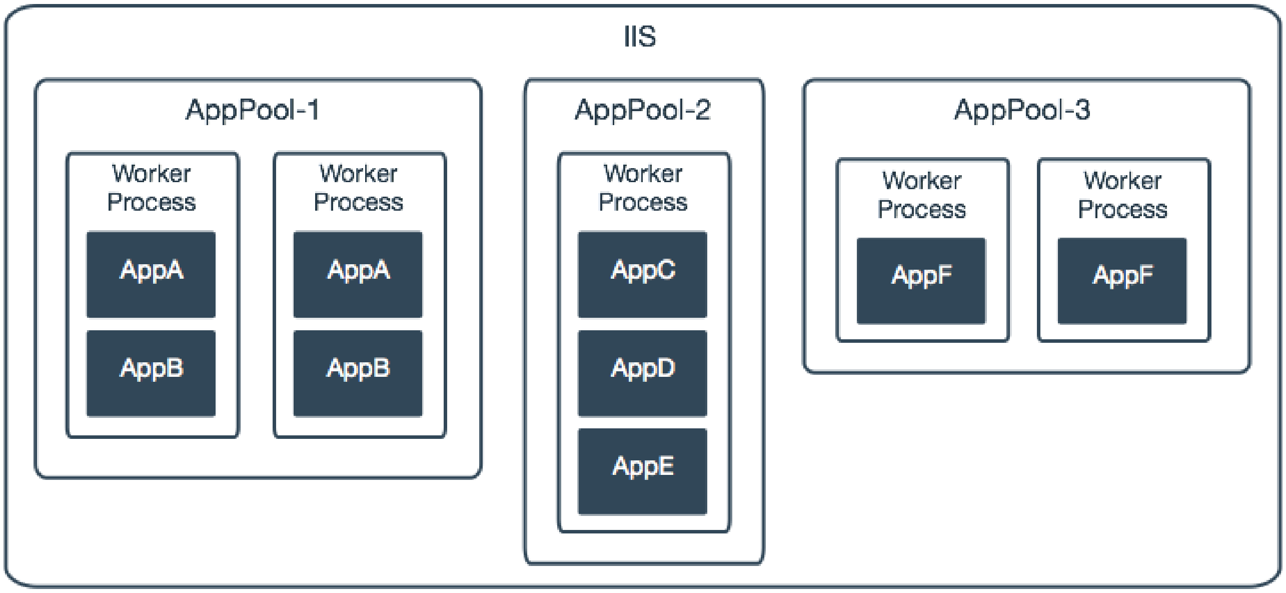
The Events Service is a file-based storage facility used by EUM, Database Monitoring, and Analytics. Database Monitoring uses the Events Service instance embedded in the Controller by default. The disk space required will vary depending upon how active the databases are and how many are being monitored.

For redundancy and optimum performance, the Events Service should run on a separate machine. For details on sizing considerations, see Events Service Requirements.

Calculating Node Count in .NET Environments

The .NET Agent dynamically creates nodes depending on the monitored application's configuration in the IIS server. An IIS server can create multiple instances of each monitored IIS application. For every instance, the .NET Agent creates a node. For example, if an IIS application has five instances, the .NET Agent will create five nodes, one for each instance.

The maximum number of instances of a particular IIS application is determined by the number of worker processes configured for its application pool, as illustrated in the following diagram:



The diagram shows three application pools — AppPool-1, AppPool-2, and AppPool-3 — with the following characteristics:

- AppPool-1 and AppPool-3 can have a maximum of two worker processes (known as a web garden), containing two applications (AppA, AppB) and one application (AppF), respectively.
- AppPool-2 can have one worker process. It has three applications.

To determine the number of nodes, for each AppPool, multiply the number of applications by the maximum number of worker processes. Add those together, as well as a node for the Windows service or standalone application processes.

The example would result in nine AppPool nodes. Adding one for a Windows service would result in a total of ten nodes, calculated as follows:

```
AppPool-1: 2 (applications) * 2 (max number of worker processes)  = 4
AppPool-2: 3 (applications) * 1 (max number of worker processes)  = 3
AppPool-3: 1 (application) * 2 (max number of worker processes)   = 2
Windows Service or standalone application process                 = 1
------
Total:                                                            = 10
```

To find the number of CLRs that will be launched for a particular .NET Application/App Pool:

1. Open the IIS manager and see the number of applications assigned to that AppPool.
2. Check if any AppPools are configured to run as a Web Garden. This would be a multiplier for the number of .NET nodes coming from this AppPool as described above.

Also see: http://technet.microsoft.com/en-us/library/cc725601(v=ws.10).aspx.

Asynchronous Call Monitoring Considerations

The Small profile is not supported for installations with extensive async monitoring. A Medium profile running 40+ agents may need to upgrade to a configuration closer to a Large profile if extensive async monitoring is added.

Specifically, monitoring asynchronous calls increases the number of metrics per minute to a maximum number of 23000 per minute.

Install the Controller Using the CLI

Related pages:

- Controller System Requirements

This page describes how to install the AppDynamics Controller using the CLI. You can also find information on settings you should have after installation. Alternatively, you can use the Enterprise Console GUI to install the Controller. For information about how to use the Enterprise Console to install the Controller, see Enterprise Console.

The Controller can be installed on the same host as the one on which the Enterprise Console is running or a remote host. Installing on the same host is not recommended, however, particularly for medium and large scale profiles or for high availability deployments.

Install the Controller Using the Command Line

Determine which host you plan to install your Controller on before starting. The host that runs the Enterprise Console is "localhost".

> ⓘ You may also use the loopback address '127.0.0.1' or the machine's actual hostname.

Complete the following steps carefully if you choose to install the Controller on this shared host rather than on a remote host. Note that all services on Windows machines must be installed on the Enterprise Console host since the Enterprise Console does not support remote operations on Windows.

In the `<Installation directory>/platform-admin` directory, run the following commands to install the Controller:

1. Create a platform:
2. Add the credential.

   For a remote host on Linux machines only:

   ```
   platform-admin.sh add-credential --credential-name <name> --type <ssh> --user-name <username> --ssh-key-
   file <file path to the key file>
   ```

   `<file path to the key file>` is the private key file. The installation process deploys the key to the Controller host.
   For the localhost:
   The localhost does not require credentials. You can, therefore, skip this step, especially for Windows deployments. For more information, see Manage Hosts.
3. Add the host.
   For a remote host on Linux machines only:

   ```
   platform-admin.sh add-hosts --hosts remotehost --credential <credential name>
   ```

   For the localhost:

4. Install the Controller on the host.
   On a remote host for Linux machines only:

   ```
   platform-admin.sh submit-job --service controller --job install --args
   controllerPrimaryHost=<remotehost> controllerAdminUsername=<user1> controllerAdminPassword=<password>
   controllerRootUserPassword=<rootpassword> mysqlRootPassword=<dbrootpassword>
   ```

   On the localhost:

Note that these are the required parameters for installing a Controller with a demo profile size. For information about optional configuration options, run the following command:

Installation Settings

The installation does some basic system checking of your environment as it performs the installation. It notifies you if it encounters conditions that need to be addressed.

Listening ports are configured for the Controller during installation. In GUI and CLI mode, the installation checks to make sure that each port it suggests is available on the system before suggesting it. You only need to edit a default port number if you know it will cause a future conflict or if you have some other specific reason for choosing another port.

Due to browser incompatibilities, AppDynamics recommends using only ASCII characters for usernames, passwords, and account names. The characters "%" and "|" are allowed in the Controller root password.

Verifying Controller Installation

Verify by navigating in a browser to the URL of the Controller UI:

```
http://<application_server_host_name>:<http-listener-port>/controller/rest/serverstatus
```

Log in using the credentials of the initial Controller administrator.

Licensing the Controller

Upon the first login, the Controller UI may prompt you that you need a valid license. You may have acquired the license file from AppDynamics.

To apply a license file manually, copy the license file to the Controller home directory. After moving the license file, allow up to 5 minutes for the license change to take effect.

Troubleshooting the Installation

A log for the installation process is automatically created in the platform-admin/logs/platform-admin-server.log. This file contains information for troubleshooting installation issues.

While installation is in progress, you can find the log file in the `platform-admin/logs/platform-admin-server.log`.

During installation and setup, the Enterprise Console tries to start the Controller. This procedure can take some time. If Controller installation fails, you can troubleshoot and identify the fix and retry from a checkpoint.

To diagnose the Controller, run the following command:

Refer to the Controller diagnostic data in `platform-admin-server.log`.

Controller High Availability

A High Availability (HA) Controller deployment helps you minimize the disruption caused by a server or network failure, administrative downtime, or other interruptions. An HA deployment is made up of two Controllers, one in the role of the primary and the other as the secondary.

The Enterprise Console automates the configuration and administration tasks associated with a highly available deployment on Linux systems. Controller HA pairs are not available on Windows Enterprise Console machines.

Essentially, to set up high availability for Controllers, you are configuring master-master replication between the MySQL instances on the primary and secondary Controllers.

An important operational point to note is that while the databases for both Controllers should be running, both Controller application servers should never be active (i.e., running and accessible by the network) at the same time. Similarly, the traffic distribution policy you configure at the load balancer for the Controller pair should only send traffic to one of the Controllers at a time (i.e., do not use round-robin or similar routing distribution policy at the load balancer).

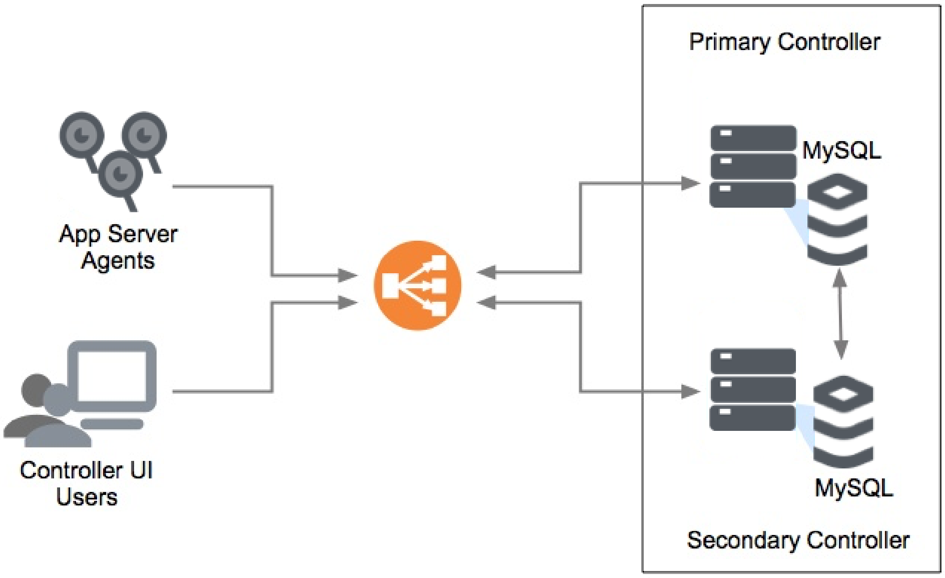
> ⓘ The Controller supports encrypted database replication.

Overview of High Availability

Deploying Controllers in an HA arrangement provides significant benefits. It allows you to minimize the downtime in the event of a server failure and take the primary Controller down for maintenance with minimal disruption. It fulfills requirements for backing up the Controller data since the secondary maintains an updated copy of the Controller data. The secondary can also be used to perform certain resource-intensive operations that are not advised to be performed on a live Controller, such as performing a cold backup of the data or accessing the database to perform long-running queries, say for troubleshooting or custom reporting purposes.

In HA mode, each Controller has its own MySQL database with a full set of the data generated by the Controller. The primary Controller has the master MySQL database, which replicates data to the secondary Controller's replica MySQL database. HA mode uses a MySQL Master-Master replication type of configuration. The individual machines in the Controller HA pair need to have an equivalent amount of disk space.

The following figure shows the deployment of an HA pair at a high level. In this scenario, the agents connect to the primary Controller through a proxy load balancer. The Controllers in an HA pair must be equivalent versions, and be in the same data center.



In the diagram, the MySQL instances are connected via a dedicated link for purposes of data replication. This is an optional but recommended measure for high volume environments. It should be a high capacity link and ideally a direct connection, without an intervening reverse proxy or firewall. See Load Balancer Requirements and Considerations on Set Up a High Availability Deployment for more information on the deployment environment.

Operating Considerations

In a high availability deployment, it is important that only one Controller is the active Controller at one time. Only the database processes should be running on the secondary so that it can maintain a replicated copy of the primary database.

The Controller app server process on the HA secondary can remain off until needed. Having two active primary Controllers is likely to lead to data inconsistency between the HA pair.

When a failover occurs, the secondary app server must be started or restarted (if it is already running, which clears the cache).

ⓘ To benefit from increased replication setup speeds, your server will need access to network resources capable of some hundreds of MB per second. By specifying replication setup parallelism, you can radically reduce setup times.

For example, if a single `rsync` is using only one-fifth of the available network capacity, you can achieve maximum throughput for setup by appending `-P r5` to end of the `replicate.sh` command. If this level of network traffic interferes with the ongoing Controller operation, you should monitor and adjust this setting.

- If you are using HA Toolkit version 3.54 and later, append `-P r5` to end of the `replicate.sh` command
- If you are using the HA module with Enterprise Console (version 4.5.17 and later), you must add the `--args numberThreadForRsync=5` to the CLI
- From the Enterprise Console UI, select **Number of parallel rsync threads** for incremental or finalize (depending on what stage you are performing)

Connecting Agents to Controllers in an HA Scenario

Under normal conditions, the App Agents and Machine Agents communicate with the primary Controller. If the primary Controller becomes unavailable, the agents need to communicate with the secondary Controller instead.

AppDynamics recommends that traffic routing be handled by a reverse proxy between the agents and Controllers, as shown in the figure above. This removes the necessity of changing agent configurations in the event of a failover or the delay imposed by using DNS mechanisms to switch the traffic at the agent.

If using a proxy, set the value of the Controller host connection in the agent configuration to the virtual IP or virtual hostname for the Controller at the proxy, as in the following example of the setting for the Java Agent in the `controller-info.xml` file:

```
<controller-host>controller.company.com</controller-host>
```

✓ For the .NET Agent, set the Controller high availability attribute to true in `config.xml`. See .NET Agent Configuration Properties.

If you set up automation for the routing rules at the proxy, the proxy can monitor the Controller at the following address:

```
http://<controller>:<port>/controller/rest/serverstatus
```

An active node returns an HTTP 200 response to GET requests to this URL, with `<available>true</available>` in the response body. A passive node returns 503, Service Unavailable, with a body of `<available>false</available>`.

For more information, see Use a Reverse Proxy.

Prerequisites for High Availability

Before You Begin

Ensure that these requirements are met:

- Controller installation pre-requisites for both servers are met. See Platform Requirements
- Two dedicated machines running Linux. The Linux operating systems can be Fedora-based Linux distributions (such as Red Hat or CentOS) or Debian-based Linux distributions (such as Ubuntu).
- In a Controller HA pair, a load balancer should route traffic from the Controller clients (Controller UI users and App Agents) to the active Controller. Before starting, make sure that a load balancer is available in your environment and that the virtual IP address for the Controller pair is known as presented by the load balancer.
- Open port number 3388 between the machines in an HA pair.
- The login shell must be bash (`/bin/bash`).
- A network link connecting the HA hosts can support a high volume of data. The primary and replica must be in the same data center, and there must be a dedicated network link between the hosts.

> ⚠ IO Latency must be under 3 ms.

- Passwordless ssh has been set up between two Controller hosts. See Set Up the SSH Key.
- SSH keys on each host allow `ssh` and `rsync` operations by the AppDynamics user.
- The `hosts` file (`/etc/hosts`) on both Controller machines should contain entries to support reverse lookups for the other node in the HA pair.
- Because Controller licenses are bound to the network MAC address of the host machine, the HA replica Controller requires an additional HA license. You should request a secondary license for HA purposes in advance.
- While adding high availability hosts as part of the add host operation, you determine and provide the remote user, of which the Controller needs to be installed as. The platform path you specify (while creating the platform) must be writable on the two HA hosts for the remote user specified during add host operation.
- The following packages are installed on both Controller hosts, and the relevant installation commands are provided:

Command	Yum based installer (RH, Centos, Amazon Linux)	Apt based installer (Ubuntu)
`lsof`	`yum install lsof`	`apt-get install lsof`
`ssh`	`yum install openssh-server`	`apt-get install openssh-server`
`awk`	`yum install gawk`	`apt-get install gawk`
`scp`	`yum install openssh-clients`	`apt-get install openssh-client`
`rsync`	`yum install rsync`	`apt-get install rsync`
`curl`	`yum install curl`	`apt-get install curl`
`sed (GNU)`	`yum install sed`	`apt-get install sed`
`openssl`	`yum install openssl`	`apt-get install openssl`
`ps`	`yum install procps`	`apt-get install procps`
`xmllint`	`yum install libxml2-utils`	`apt-get install libxml2-utils`
`timeout/base64/tr`	`yum install coreutils`	`apt-get install coreutils`

Set Up a High Availability Deployment

This page describes how to set up and deploy Controllers as a high availability pair. For installation and upgrade details, see Custom Install and Upgrade an HA Pair.

> ⚠ The Enterprise Console HA deployment works on Linux systems only. Controller HA pairs are not available on Windows machines using Enterprise Console.

About the HA Deployment Using the Enterprise Console

The Enterprise Console automates HA-related setup and administration tasks for the Linux operating system. It does not require `sudo` privileges and can be deployed as a non-root user on Unix operating systems. It works with most flavors of Linux, including Ubuntu and Red Hat/CentOS.

> ⚠ The servers of Controllers in an HA pair must be identical in terms of OS, CPU, RAM, and Disk. See Controller System Requirements.

You can:

- Configure Controllers in a high availability pair arrangement.
- Use the Enterprise Console to monitor the health of the primary Controller, App Server, and database, and failover to the secondary when needed.
- Use scripts that allows you to install the Controllers as a Linux service, and gracefully stop and start service in the event of a machine reboot.
- Failover to a secondary Controller manually (for example, when you need to perform maintenance on the primary).
- Revive a Controller (restore a Controller as an HA secondary after its database is more than seven days behind the primary as a replica).
- Set up a Controller HA pair.

Deploying Controllers as an HA pair ensures that service downtime in the event of a Controller machine failure is minimized. It also facilitates other administrative tasks, such as backing up data. For more background information, including the benefits of HA, see Controller High Availability (HA).

Before Starting

For general guidelines and requirements on how to deploy HA in your environment, see Prerequisites for High Availability. Your environment must meet the prerequisites.

User Privilege Escalation Requirements

After installing a Controller high availability via the Enterprise Console, it is recommended to install MySQL as a Linux service. This is to prevent against MySQL data integrity issues. Installing the Controller and MySQL as a Unix service will ensure that whenever the machine reboots, the service will be shut down and started gracefully.

- `/etc/sudoers.d/appdynamics` contains entries to allow the AppDynamics user to access the `/sbin/service` utility using `sudo` without a password. This mechanism is not available if the AppDynamics user is authenticated by LDAP.
- `/sbin/appdservice` is a `setuid` root program distributed in source form in `<controller_home>/controller-ha/init/appdservice.c`. It is written explicitly to support auditing by security audit systems. The `install-init.sh` script compiles and installs the program. It is executable only by the AppDynamics user and the root user. The script requires a C compiler to be available on the system. You can install a C compiler using the package manager for your operating system. For example, on Yum-based Linux distributions, you can use the following command to install the GNU Compiler, which includes a C compiler:

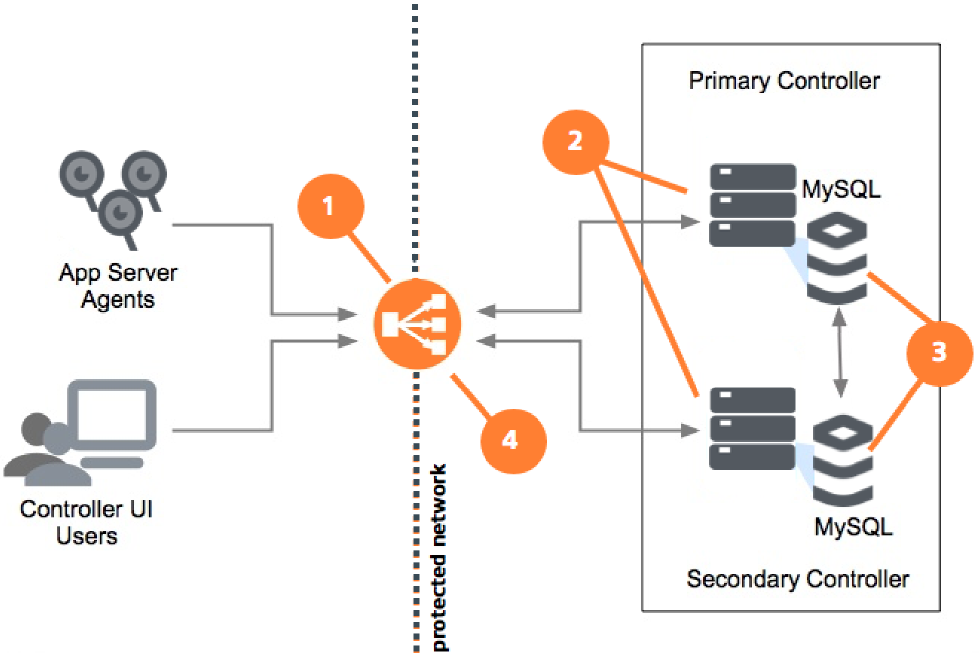
  ```
  sudo yum install gcc
  ```

Load Balancer Requirements and Considerations

Before setting up HA, a reverse proxy or load balancer needs to be available and configured to route traffic to the active Controller in the HA pair. Using a load balancer to route traffic between Controllers (rather than other approaches, such as DNS manipulation) ensures that a failover can occur quickly, without, for example, delays due to DNS caching on the agent machines.

An HA deployment requires the following IP addresses:

- One for the virtual IP of the Controller pair as presented by the load balancer used by clients, such as App Agents, to access the Controller. (Callout 1 in the following diagram.)
- Two IP addresses for each Controller machine, one for the HTTP primary port interface (2) and one for the dedicated link interface between the Controllers (3) on each machine. The dedicated link is recommended but not mandatory.
- If the Controllers will reside within a protected, internal network behind the load balancer, you also need an additional internal virtual IP for the Controller within the internal network (4).



When configuring replication, you specify the external address at which Controller clients, such as app agents and UI users, will address the Controller at the load balancer. The Controllers themselves need to be able to reach this address as well. If the Controller will reside within a protected network relative to the load balancer, preventing them from reaching this address, there needs to be an internal VIP on the protected side that proxies the active Controller from within the network. This is specified using the `-i` parameter.

The load balancer can check the availability of the Controller at the following address:

`http://<controller_host>:<port>/controller/rest/serverstatus`

If the Controller is active, it responds to a GET request at this URL with an HTTP 200 response. The body of the response indicates the status of the Controller in the following manner:

```
<serverstatus vendorid="" version="1">
   ...
   <available>true</available>
   ...
```

Ensure that the load balancer policy you configure for the Controller pair can send traffic to only a single Controller in the pair at a time (i.e., do not use round-robin or similar routing distribution policy at the load balancer). For more information about setting up a load balancer for the Controller, see Use a Reverse Proxy.

Set Up the Controller High Availability Pair

To set up high availability:

Step 1: Configure the Controller High Availability Pair Environment
Step 2: Install a Controller High Availability Pair
Step 3: Activate the Controller High Availability Pair
Step 4: Install as a Service

Step 1: Configure the Controller High Availability Pair Environment

The following sections provide more information on how to configure a few of the system requirements. They describe how to configure the settings on Red Hat Linux as a sample deployment. Note that the specific steps for configuring these requirements may differ on different systems. Consult your system documentation for specific details.

Host Reverse Lookups

You need to set up a reliable symmetrical reverse host lookup on each machine. To do this, enter the hostnames of the pair into the hosts files (`/etc/hosts`) on each machine. This is preferable over other approaches, such as using reverse DNS, which adds a point of failure.

To enable reverse host lookups, on each host:

1. In `/etc/nsswitch.conf`, enter `files` before `dns` to have the `hosts` file entries take precedence over DNS. For example:
   `hosts: files dns`
2. In `/etc/hosts` file, add an entry for each host in the HA pair. For example:
   ```
   192.168.144.128 host1.domain.com host1
   192.168.144.137 host2.domain.com host2
   ```

> To reduce errors, use the correct format of `/etc/hosts` files. If you have both dotted hostnames and short versions, you need to list the dotted hostnames with the most dots first and the other versions subsequently. This should be done consistently for both HA server entries in each of the two `/etc/hosts` files. Note that in the examples provided, the aliases are listed last.

Set Up the SSH Key

SSH must be installed on both hosts in a way that gives the user who runs the Controller passwordless SSH access to the other Controller system in the HA pair. You can accomplish this by generating a key pair on each node, and placing the public key of the other Controller into the authorized keys (authorized_keys) file on each Controller.

The following steps describe how to perform this configuration. The instructions assume an AppDynamics user named `appduser,` and the Controller hostnames are node1, the active primary, and node2, the secondary. Adjust the instructions for your particular environment. Also note that you may not need to perform every step (for example, you may already have the `.ssh` directory and don't need to create a new one).

Although not shown here, some of the steps may prompt you for a password.

On the primary (node1) host:

1. Change to the AppDynamics user, `appduser` in our example:

   ```
   su - appduser
   ```

2. Create a directory for SSH artifacts (if it doesn't already exist) and set permissions on the directory, as follows:

   ```
   mkdir -p .ssh
   chmod 700 .ssh
   ```

3. Generate the RSA-formatted key:

   ```
   ssh-keygen -t rsa -N "" -f .ssh/id_rsa -m pem
   ```

4. Secure copy the key to the other Controller:

   ```
   scp .ssh/id_rsa.pub node2:/tmp
   ```

On the secondary (node2) host:

1. As you did for node1, run these commands:

   ```
   su - appduser
   mkdir -p .ssh
   chmod 700 .ssh
   ssh-keygen -t rsa -N "" -f .ssh/id_rsa -m pem
   scp .ssh/id_rsa.pub node1:/tmp
   ```

2. Add the public key of node1 that you previously copied to the secondary Controller host's authorized keys and set permissions on the authorized keys file:

   ```
   cat /tmp/id_rsa.pub >> .ssh/authorized_keys
   chmod 700 ~/.ssh/authorized_keys
   ```

On the primary (node1) again:

1. Move the secondary's public key to the authorized keys

   ```
   cat /tmp/id_rsa.pub >> ~/.ssh/authorized_keys
   chmod 700 ~/.ssh/authorized_keys
   ```

To test the configuration, enter:

```
ssh -oNumberOfPasswordPrompts=0 <other_node> "echo success"
```

Verify that the `echo` command is successful.

Step 2: Install a Controller High Availability Pair

Once you have set up the environment, you can install a Controller HA pair on the primary machine. To add an HA secondary to an existing standalone Controller deployment, you only need to verify that the user who runs the Controller has write access to the Controller home.

To install a Controller HA pair:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
3. Verify that the credentials and hosts you want to use are added to the AppDynamics platform. For more information, see Administer the Enterprise Console.

   a. On the **Credential** page, add the SSH credentials for the host you want to install the primary Controller on. You can also run the following command on the Enterprise Console host:

      ```
      bin/platform-admin.sh add-credential --credential-name <name> --type <ssh> --user-name <username>
      --ssh-key-file <file path to the key file>
      ```

      > Remember to provide the private key file for the Enterprise Console machine when adding a credential.

   b. On the **Hosts** page, add the host using the credentials from above. You can also run the following command on the Enterprise Console host:

      ```
      bin/platform-admin.sh add-hosts --hosts secondaryhost --credential <credential name>
      ```

4. Navigate to the **Install** homepage and click **Custom Install**.
5. Name the platform:

   a. Enter a Name and the Installation Path for your platform.

      > The Installation Path is an absolute path under which all of the platform components are installed. The same path is used for all hosts added to the platform. Use a path which does not have any existing AppDynamics components installed under it. The path you choose must be writeable, i.e. the user who installed the Enterprise Console should have write permissions to that folder. Also, the same path should be writable on all of the hosts that the Enterprise Console manages.
      >
      > Example path: `<path_to_AppD>/appdynamics/platform`

      Or run the following command on the Enterprise Console host:

```
bin/platform-admin.sh create-platform --name <platform_name> --installation-dir
<platform_installation_directory>
```

6. Add two hosts:
   a. For each of the two hosts, enter their host machine information: Host Name, Username, and Private Key.
      This is the location onto which the Controllers will be installed. For more information about how to add credentials and hosts, see Administer the Enterprise Console.
7. Install Controller:
   a. Select **Install**.
   b. Select a Profile size for your Controller. See Controller System Requirements for more information on the sizing requirements.
   c. Enter the Controller Primary and Secondary hosts.
   d. Enter the required Username and Passwords. The default Controller Admin Username is `admin`.

   > ⓘ If you do not install a Controller at this time, you can always do so later by navigating to the **Controller** page in the GUI and clicking Install Controller.

   Or run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh submit-job --service controller --job install --args
   controllerPrimaryHost=<primaryhost> controllerSecondaryHost=<secondaryhost>
   controllerAdminUsername=<user1> controllerAdminPassword=<password>
   controllerRootUserPassword=<rootpassword> mysqlRootPassword=<dbrootpassword>
   ```

8. Click **Install**.

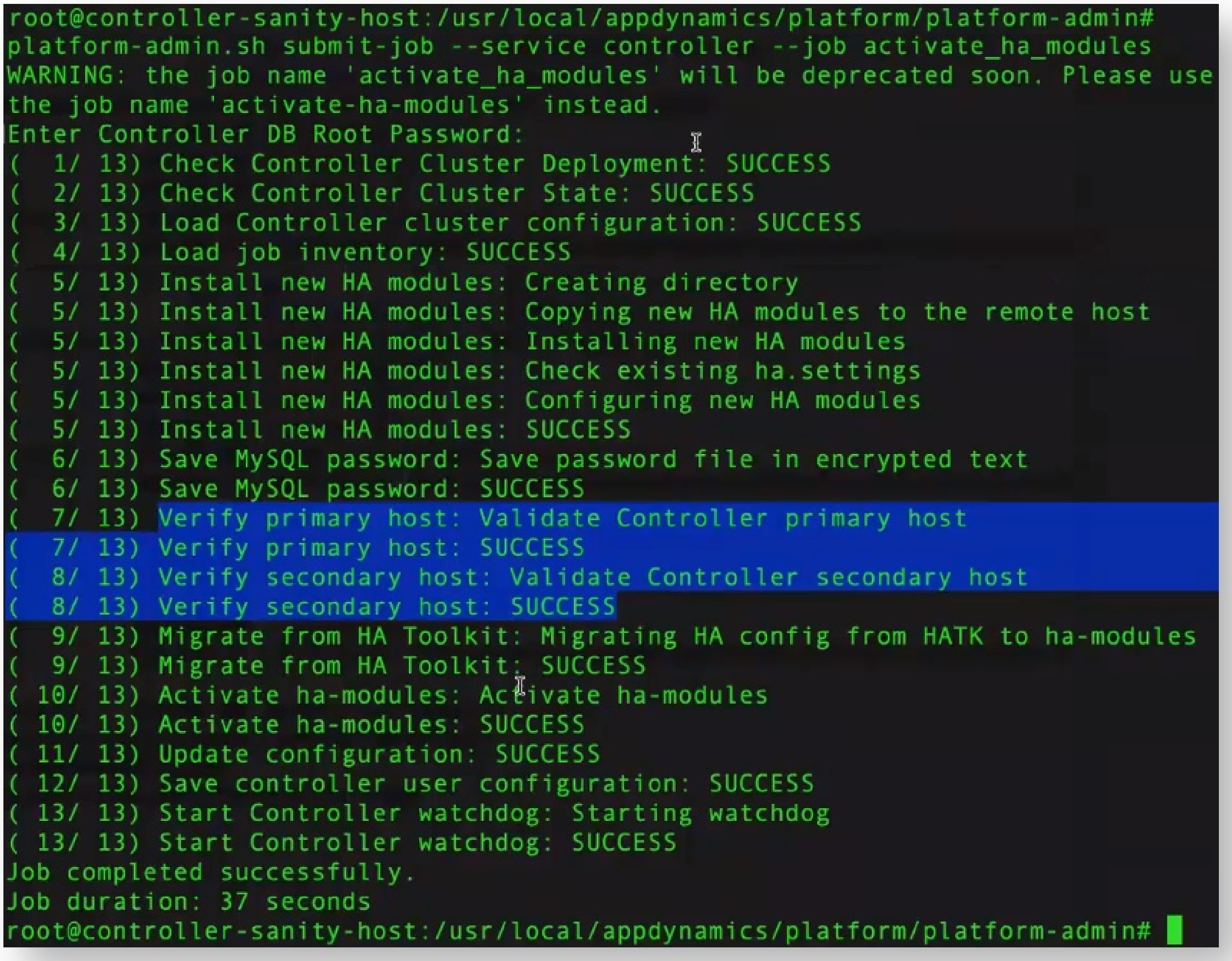
Step 3: Activate the Controller High Availability Pair

This step ensures that the Enterprise Console is no longer in the critical path of the HA Controller failover process.

Open a command shell on the Enterprise Console host and enter:

```
platform-admin.sh submit-job --service controller  --job activate-ha-modules
```

Your output should be similar to the following:



```
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
platform-admin.sh submit-job --service controller --job activate_ha_modules
WARNING: the job name 'activate_ha_modules' will be deprecated soon. Please use
the job name 'activate-ha-modules' instead.
Enter Controller DB Root Password:
(  1/ 13) Check Controller Cluster Deployment: SUCCESS
(  2/ 13) Check Controller Cluster State: SUCCESS
(  3/ 13) Load Controller cluster configuration: SUCCESS
(  4/ 13) Load job inventory: SUCCESS
(  5/ 13) Install new HA modules: Creating directory
(  5/ 13) Install new HA modules: Copying new HA modules to the remote host
(  5/ 13) Install new HA modules: Installing new HA modules
(  5/ 13) Install new HA modules: Check existing ha.settings
(  5/ 13) Install new HA modules: Configuring new HA modules
(  5/ 13) Install new HA modules: SUCCESS
(  6/ 13) Save MySQL password: Save password file in encrypted text
(  6/ 13) Save MySQL password: SUCCESS
(  7/ 13) Verify primary host: Validate Controller primary host
(  7/ 13) Verify primary host: SUCCESS
(  8/ 13) Verify secondary host: Validate Controller secondary host
(  8/ 13) Verify secondary host: SUCCESS
(  9/ 13) Migrate from HA Toolkit: Migrating HA config from HATK to ha-modules
(  9/ 13) Migrate from HA Toolkit: SUCCESS
( 10/ 13) Activate ha-modules: Activate ha-modules
( 10/ 13) Activate ha-modules: SUCCESS
( 11/ 13) Update configuration: SUCCESS
( 12/ 13) Save controller user configuration: SUCCESS
( 13/ 13) Start Controller watchdog: Starting watchdog
( 13/ 13) Start Controller watchdog: SUCCESS
Job completed successfully.
Job duration: 37 seconds
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
```

Step 4: Install as a Service

The Enterprise Console does not install the Controller as a service on Linux because it requires root user or `sudo` privileges. However, the Enterprise Console copies the `init` scripts on the Controller hosts in `<controller_home>/controller-ha`. To complete the installation, manually run the following:

1. Change directories to `<controller_home>/controller-ha`.
2. As the user who owns the Controller folder, run:

   ```
   set_mysql_password_file.sh -p <db root password> -s <secondary host>
   ```

   `-s` copies the file to the secondary host. Enter the MySQL database root user password in the command.
3. Change directories to `<controller_home>/controller-ha/init`.
4. Run `install-init.sh` as root user with one of the following options to select how to elevate the user privilege:
   - `-c` #use setuid c wrapper
   - `-s` #use sudo
   - `-p` #use prune wrapper
   - `-x` #use user privilege wrapper

You must run this script on both Controller HA pair servers. If you need to uninstall the service later, run the `uninstall-init.sh` script.

The status and progress of the deployment's various components are written to the logs.

Convert a Standalone Controller to a Controller High Availability Pair

You can convert a standalone Controller to a Controller HA pair through the Enterprise Console GUI. Ensure that you have completed the prerequisites in the above Configure the Controller High Availability Pair Environment section that requires both the primary and secondary hosts to talk to each other via passwordless SSH.

Additionally, you can use incremental replication to add a secondary Controller. See Initiate Controller Database Incremental Replication for more information.

If you are starting from a fresh installation, you will need to first create a platform, then add two credentials and hosts for your HA pair.

To convert a standalone Controller to a Controller HA pair:

1. Open the Enterprise Console GUI.
2. Verify that the credentials and hosts you want to use are added to the AppDynamics platform. For more information, see Administer the Enterprise Console.

   a. On the **Credential** page, add the SSH credentials for the host you want to install the secondary Controller on. You can also run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh add-credential --credential-name <name> --type <ssh> --user-name <username>
   --ssh-key-file <file path to the key file>
   ```

   ⓘ Remember to provide the private key file for the Enterprise Console machine when adding a credential.

   b. On the **Hosts** page, add the host. You can also run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh add-hosts --hosts secondaryhost --credential <credential name>
   ```

   The Enterprise Console uses this host for the HA pair.
3. On the **Controller** page, click **Add Secondary Controller**, and complete the wizard:

   a. Select the Controller Secondary Host that you added for the secondary Controller.
   b. Optional: Enter the External URL. This is the external load balancer URL, which should reflect this format: `http(s)://<external.vip>:<port>`
   c. Enter the DB Root Password, and re-enter it for confirmation.

   ⚠ Ensure to provide the same passwords during the secondary server installation as those that you provided for the primary server.

4. Select **Submit**.

Your HA pair will automatically set up, each with their own MySQL node.

Manage a High Availability Deployment

This page describes how to manage and troubleshoot Controllers as a high availability pair.

Set Up Monitoring for the HA Pair

You can set up monitoring for your HA pair by installing another Controller to act as the monitoring Controller.

1. If you do not already have an HA pair, set one up.
2. Install the monitoring Controller on the Enterprise Console host in a new platform by selecting Custom Install:
   a. Create a platform (e.g.: Controller Monitor Platform).

      > ⚠ This platform should not be used for installing any other services.

   b. Install a Controller.
   c. Make sure to unselect the Install Events Service option before clicking **Install**.
3. Complete the monitoring setup by installing and configuring the App Agents and Machine Agents on your HA pair:
   - Set Up App Agents for Monitoring
   - Install and Set Up Machine Agents for Monitoring

Set Up App Agents for Monitoring

You can set up App Agents, which are automatically installed on the Controller hosts by the Enterprise Console, on both Controllers of an HA pair to report to the monitoring Controller. This can be done by updating the JVM options of your HA pair platform. To set up your App Agents using the Enterprise Console, perform the following steps:

1. SSH into the primary Controller box and update the primary Controller App Agent's `controller-info.xml` by running the following commands:

```
cd <controller-install-dir>/appserver/glassfish/domains/domain1/appagent
cp conf/controller-info.xml ver<version#>/conf/
```

2. Repeat step 1 for the secondary Controller.
3. In the Enterprise Console UI, select your HA pair platform, and navigate to the JVM Options section by clicking **Configurations > Controller Settings** > **Appserver Configurations**.
4. Make the following updates to JVM Options:

   a. Update the `appdynamics.controller.hostName` to the monitoring Controller's IP.
   b. Add the following required `jvm-options` for monitoring:

```
-Dappdynamics.agent.applicationName=<app_name>, -Dappdynamics.agent.tierName=<tier_name>,
-Dappdynamics.agent.nodeName=<node_name>, -Dappdynamics.agent.accountName=<account_name>,
-Dappdynamics.agent.accountAccessKey=<access_key>
```

> ⓘ You can get your access key from the Controller UI: navigate to **Settings**> **License** > **Account**. Then click to show your access key. Note, when you log in to the Controller, use the account specified in `appdynamics.agent.accountName`.

5. Scroll down the page and click **Save**. The job will apply these properties and restart both the primary and secondary Controllers.
6. In the Enterprise Console UI, select your Controller Monitor Platform, and navigate to the Controller page.
7. Click on **External URL** on the widget to open the monitoring Controller's UI.
8. Log in to the Controller. You should be able to see the monitoring application for both the primary and secondary Controllers.

Install and Set Up Machine Agents for Monitoring

You must install Machine Agents on both Controllers of an HA pair to report to the monitoring Controller. These agents are Java programs that collect hardware metrics. To install and set up your machine agents, perform the following steps:

1. Install the Machine Agent on the primary Controller box. Do not start the agent.
2. Repeat step 1 for the secondary Controller.

3. Configure the Machine Agent properties for both Machine Agents by editing the `controller-info-xml` file located in the `<machine_agent_home>/conf` directory.
    a. Update the `<controller-host>` to the monitoring Controller's IP.
    b. Model the rest of your `controller-info-xml` file after the Example Configuration.
4. Start both Machine Agents.
5. In the Enterprise Console UI, select your Controller Monitor Platform, and navigate to the Controller page.
6. Click on **External URL** on the widget to open the monitoring Controller's UI.
7. Log in to the Controller. You should be able to see the monitoring application for both the primary and secondary Controllers.

Bouncing the Primary Controller Without Triggering Failover

The Enterprise Console does not allow you to stop and start the primary Controller without initiating failover. Therefore, to work around this, you will need to perform the following steps:

1. Log in to the Enterprise Console and navigate to the Appserver Configurations page by clicking through Configurations, followed by Controller Settings.
2. Deselect **Enable Auto Failover** and click **Save**.
3. SSH to the Controller machine where the Controller is installed.
4. Run the following commands on the Enterprise Console host:

    ```
    bin/platform-admin.sh stop-controller-appserver
    bin/platform-admin.sh start-controller-appserver
    ```

    This will bounce the primary Controller in HA mode.
5. Re-enable auto failover on the Enterprise Console Appserver Configurations page.

Starting and Stopping the Controller

The Enterprise Console does not allow you to shut down the primary Controller. However, you can restart the secondary Controller via the start and stop Controller commands.

To start or stop the Controller manually, use the following commands:

- To start:

    ```
    bin/platform-admin.sh start-controller-appserver --with-db
    ```

- To stop:

    ```
    bin/platform-admin.sh stop-controller-appserver --with-db
    ```

Automatic Failover

The Enterprise Console monitors the health of the primary Appserver and database. If the Appserver or database is unresponsive, the Enterprise Console will by default wait for five minutes before initiating a failover. This interval can be configured by updating the default value in the Domain Protocol text field on the Appserver Configurations page under Controller settings.

You can also disable or enable automatic failover through the CLI.

ⓘ Version 4.5.14 and above of the Enterprise Console comes with the High Availability (HA) modulewhich utilizes the Controller Watchdog for auto-failover. If you want to enable or disable the auto-failover, then the `watchdog` script needs to be running or stopped.

To disable and enable the Controller Watchdog with CLI using the following commands:

- To stop the Controller Watchdog:

```
./platform-admin.sh submit-job --job stop-controller-watchdog --service controller
```

- To start the Controller Watchdog:

```
./platform-admin.sh submit-job --job start-controller-watchdog --service controller
```

Performing a Manual Failover and Failback

To failover from the primary to the secondary manually, click the **HA Failover** option on the **Controller** page of the Enterprise Console or run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --service controller --job ha-failover --platform-name <name_of_the_platform>
```

This changes the Appserver on the secondary as primary and database on the secondary as the replication master. It also changes the old primary to secondary.

The process for performing a failback to the old primary is the same as failing over to the secondary. Simply run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --service controller --job ha-failover --platform-name <name_of_the_platform>
```

Note that if it has been down for more than seven days, you need to revive the database, as described in the following section.

Initiate Controller Database Incremental Replication

Re-enable Broken Replication

Incremental replication, replication via rsync when the primary database is up, is required in cases where the database replication on the secondary Controller is lagging behind the primary Controller by more than three days. This type of replication allows the primary Controller to keep operating while the disk contents are copied to the secondary node.

To initiate incremental replication:

1. Run the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh submit-job --service controller --job incremental-replication
   ```

   This launches a continuously running background job.
2. Make sure replication occurs four or more times, by checking `mysqlDir/incremental_sync.status` on the primary database host.

   Sample rsync status file output:

   ```
   rsync started at Mon Mar  5 11:49:56 PST 2018
   rsync completed at Mon Mar  5 11:50:56 PST 2018
   rsync started at Mon Mar  5 11:51:01 PST 2018
   rsync completed at Mon Mar  5 11:51:11 PST 2018
   ```

   ⓘ If replication fails, go to the secondary host and stop all rsync and ha-replicate.sh processes. Then try running the incremental-replication job again.

3. Finalize the job by running the following command on the Enterprise Console host:

   ```
   bin/platform-admin.sh submit-job --service controller --job finalize-replication
   ```

   This stops the incremental replication loop. The command will restart the primary Controller, resulting in downtime.
4. Make sure replication is working by checking that there is no significant gap between the primary and secondary Controllers. You can run the following command on the Enterprise Console host to check the replication status:

```
bin/platform-admin.sh show-service-status --platform-name <platform_name> --service controller
```

It may take a few minutes for the secondary status to catch up.

Add a Secondary Controller Using Incremental Replication

You can convert a single Controller with a large amount of data to an HA pair by using incremental replication. This way, you can rsync most of the Controller's data while the Controller is still running, limiting the downtime of adding a secondary Controller.

To add a secondary Controller using incremental replication:

1. Start the incremental replication, giving host and rsync parameters:

   ```
   bin/platform-admin.sh submit-job --service controller --job incremental-replication --args
   controllerSecondaryHost=1.1.1.1 rsyncThrottle=40000 rsyncCompress=true
   ```

   This launches a continuously running background job.
2. Make sure replication occurs four or more times, by checking `<controller_home>/controller-ha/tmp/replication.status` on the primary database host.
   Sample rsync status file output:

   ```
   rsync started at Mon Mar  5 11:49:56 PST 2018
   rsync completed at Mon Mar  5 11:50:56 PST 2018
   rsync started at Mon Mar  5 11:51:01 PST 2018
   rsync completed at Mon Mar  5 11:51:11 PST 2018
   ```

   > If replication fails, go to the secondary host and stop all rsync and ha-replicate.sh processes. Then try running the incremental-replication job again.

3. Run the add secondary job. The Enterprise Console will perform a final rsync and add the secondary.

   ```
   bin/platform-admin.sh submit-job --service controller --job add-secondary --args
   controllerSecondaryHost=secondary mysqlRootPassword=‘password'
   ```

   The command will restart the primary Controller, resulting in downtime.

   > Until you trigger the add-secondary command, the secondary Controller is not added to the Enterprise Console platform. Therefore, the Enterprise Console will not be able to perform any other operations on the secondary Controller.

If you need to stop replication, you can run the following command:

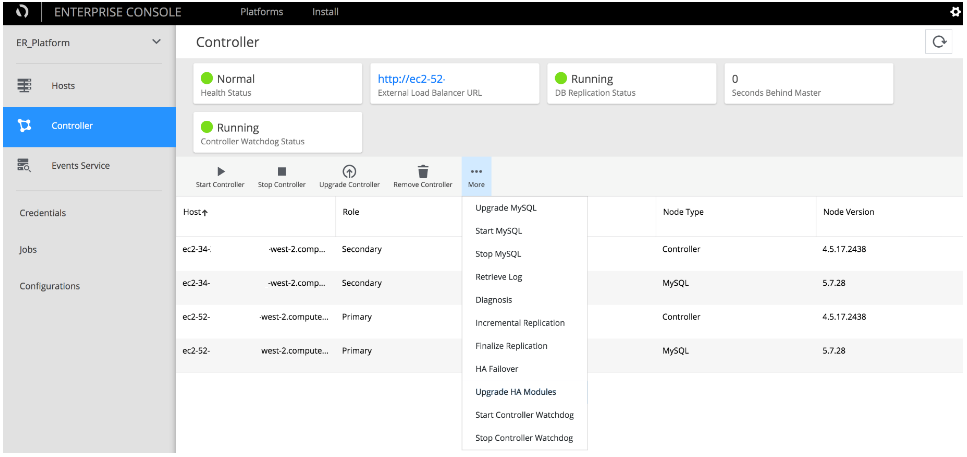
```
bin/platform-admin.sh submit-job --service controller --job stop-incremental-replication
```

Set Replication Factors for Rsync Threads

Using the Enterprise Console UI or the CLI, you can set the number of parallel rsync threads as a job parameter when you perform incremental or finalize replication.

- From the Enterprise Console UI:
  1. Log in to the Enterprise Console and access the Controller page.

2. From the **More** menu, based on which replication you are performing, select either **Incremental Replication** or **Finalize Replication**.



Normal
Health Status

http://ec2-52-
External Load Balancer URL

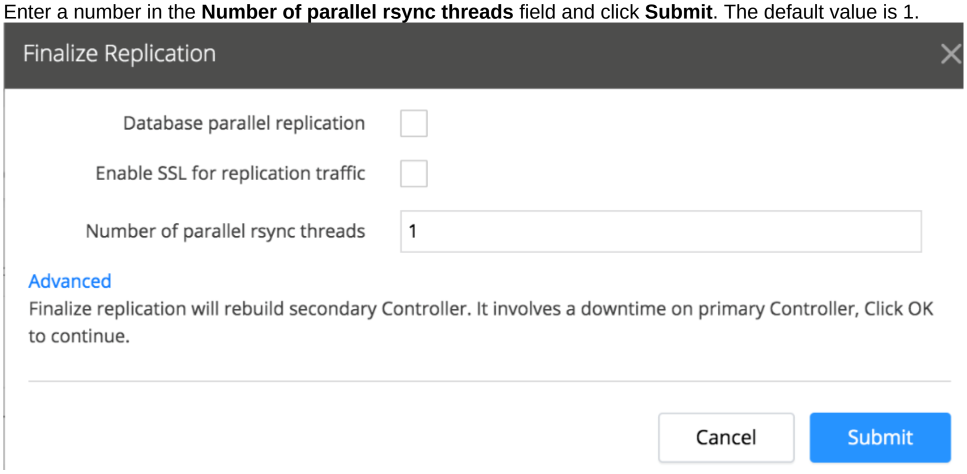
Running
DB Replication Status

0
Seconds Behind Master

Running
Controller Watchdog Status

Host ↑	Role	Node Type	Node Version
ec2-34- -west-2.comp...	Secondary	Controller	4.5.17.2438
ec2-34- -west-2.comp...	Secondary	MySQL	5.7.28
ec2-52- -west-2.compute...	Primary	Controller	4.5.17.2438
ec2-52- west-2.compute...	Primary	MySQL	5.7.28

3. Enter a number in the **Number of parallel rsync threads** field and click **Submit**. The default value is 1.



Database parallel replication □

Enable SSL for replication traffic □

Number of parallel rsync threads 1

Finalize replication will rebuild secondary Controller. It involves a downtime on primary Controller, Click OK to continue.

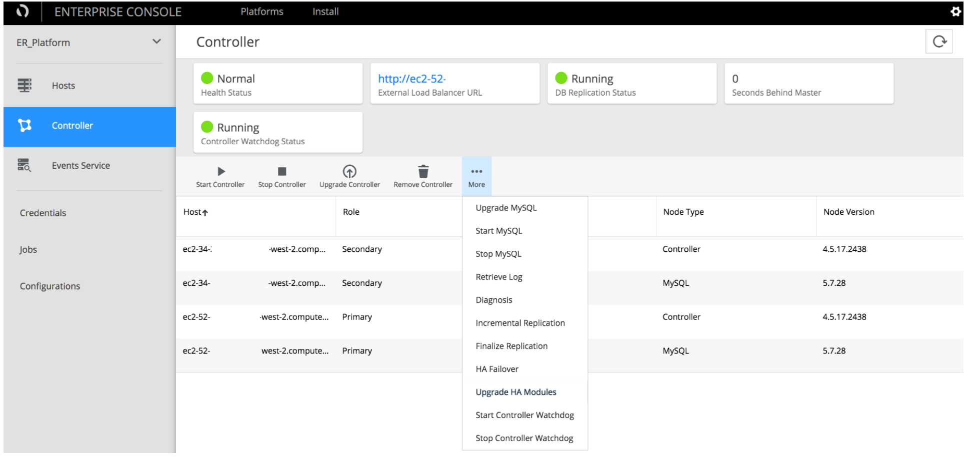
- From the CLI, based on which replication you are performing, run either of the following commands from the Enterprise Console host and set the `numberThreadForRsync` argument.

```
bin/platform-admin.sh submit-job --job incremental-replication --args numberThreadForRsync=<number> bin
/platform-admin.sh submit-job --job finalize-replication --args numberThreadForRsync=<number>
```

Enable MySQL5.7 Parallel Replication

Using the Enterprise Console UI or the CLI, you can enable MySQL5.7 parallel replication when you perform finalize replication.

- From the Enterprise Console UI:
    1. Log in to the Enterprise Console and access the Controller page.
    2. From the **More** menu, select **Finalize Replication**.



Normal
Health Status

http://ec2-52-
External Load Balancer URL

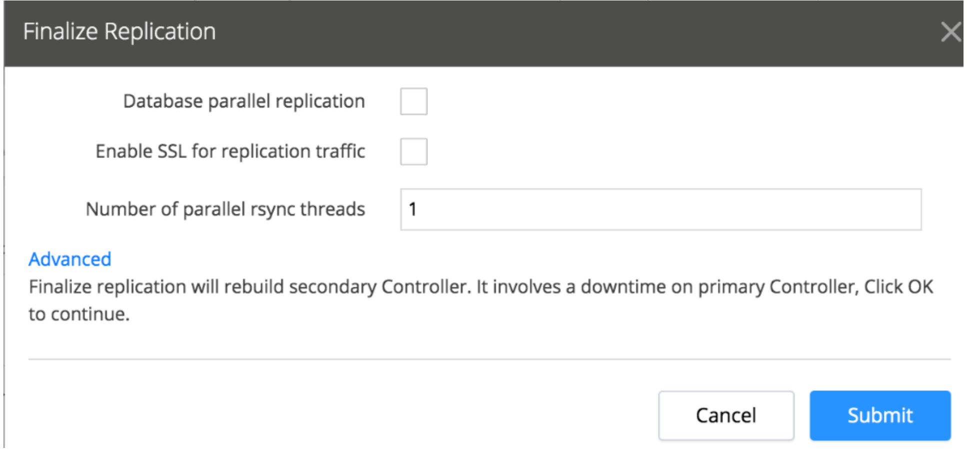
Running
DB Replication Status

0
Seconds Behind Master

Running
Controller Watchdog Status

Host ↑	Role	Node Type	Node Version
ec2-34- -west-2.comp...	Secondary	Controller	4.5.17.2438
ec2-34- -west-2.comp...	Secondary	MySQL	5.7.28
ec2-52- -west-2.compute...	Primary	Controller	4.5.17.2438
ec2-52- west-2.compute...	Primary	MySQL	5.7.28

3. Select the **Database parallel replication** check box to enable parallel replication with the MySQL5.7 database.



Database parallel replication ☐

Enable SSL for replication traffic ☐

Number of parallel rsync threads 1

Finalize replication will rebuild secondary Controller. It involves a downtime on primary Controller, Click OK to continue.

4. Click **Submit**.

- From the CLI, run the following command from the Enterprise Console host to enable MySQL5.7 parallel replication. The default value is true.

```
bin/platform-admin.sh submit-job --job finalize-replication --args dbParallelReplication=true
```

Troubleshooting the Incremental Replication Status

If your first incremental replication run is taking longer than usual, you can refer to the status file, `<controller_home>/incremental_sync.status`, to review a detailed list of files that are being rsynced. You can find the file in the primary Controller host under the Platform folder: `mysqlDir /<controller_home>/incremental_sync.status`.

Re-enable Controller Database Replication

The Controller databases can be synchronized using the replicate script if they have been out of sync for more than seven days. Synchronizing a database that is more than seven days behind a master is considered reviving a Controller database. Reviving a database involves essentially the same procedure as adding a new secondary Controller to an existing production Controller, as described in Set Up the Secondary Controller and Initiate Replication. You can also follow these steps in the case of an HA failover that failed at replication.

To re-enable replication or revive a Controller database:

1. On the Controller page, click **Remove Controller**, or run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --job remove --service controller
```

2. Enter the database root credentials.
3. Check **Remove Binaries**, or run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --job remove --service controller --args removeBinaries=true
```

4. Uncheck **Remove Controller Cluster**. If it is already unchecked, remove the secondary server.
5. Click **Submit**.
6. Add a secondary controller from the Controller page, or run the following command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --service controller --job add-secondary --args
controllerSecondaryHost=secondary mysqlRootPassword=‘password'
```

The command will restart the primary Controller, resulting in downtime.

The Enterprise Console will onboard the secondary Controller and re-enable replication.

Backing Up and Restoring Controller Data in an HA Pair

An HA deployment makes backing up Controller data relatively straightforward since the secondary Controller offers a complete set of production data on which you can perform a cold backup without disrupting the primary Controller service.

After setting up HA, perform a back up by stopping the Controller on the Enterprise Console and performing a file-level copy of the AppDynamics home directory (i.e., a cold backup). When finished, simply restart the Controller from the Enterprise Console. The secondary will then catch up its data to the primary.

When restoring the database from a back up in an HA or standalone environment, you should check that the primary and secondary servers ha.type and ha.mode are set properly to active and passive, respectively.

Updating the Configuration in an HA Pair

The Enterprise Console will copy any file-level configuration customizations made on the primary controller to the secondary controller, such as changes in domain.xml and `db.cnf`.

Over time, if you need to make modifications to the Controller configuration, always do those changes in the Enterprise Console on the Controller Settings page under Configurations. These changes will be preserved during upgrades. Any changes made outside the Enterprise Console will not be preserved after upgrade.

Troubleshooting HA

Controller Diagnostic Data

The Enterprise Console writes log messages pertaining to HA to the `platform-admin-server.log` on the Enterprise Console host.

To diagnose the Controller, run the following command:

```
bin/platform-admin.sh submit-job --platform-name <name_of_the_platform> --job diagnosis --service controller
```

Refer to the Controller diagnostic data in the `platform-admin-server.log`.

Sample Controller diagnostic data

Linux

```
Controller diagnostic data:
123.45.0.1:
controller_database: running
controller_appserver: running
reports_service: running
operating_system: Linux
controller_version: 004-004-001-000
controller_performance_profile: small
controller_ha_type: primary
controller_appserver_mode: active
controller_metric_data_per_min: N/A
slave_io_state: Waiting for master to send event
seconds_behind_master: 0
master_server_id: 567.
master_host: controller-secondary
master_ssl_allowed: No

123.45.0.2:
controller_database: running
controller_appserver: not running
reports_service: running
operating_system: Linux
controller_version: 004-004-001-000
controller_performance_profile: small
controller_ha_type: secondary
controller_appserver_mode: passive
```

Invalid HA Controller Roles

If your HA Controller roles in the Controller databases are incorrect, the Enterprise Console will prevent discover and upgrade jobs. An invalid HA Controller state is when both of your Controller role types are identical, such as in a primary/primary or secondary/secondary case.

To fix this issue:

1. Identify which server is the primary.
    a. Log in to one of the Controller databases by running the following command in the Controller installation directory:

    ```
    bin/controller.sh login-db
    ```

    b. Run the following command:

    ```
    select * from global_configuration_local where name=‘ha.controller.type’;
    ```

2. Ensure that `ha.controller.type` is set correctly in the database.
    a. Log in to the Controller database you would like to change by running the following command in the Controller installation directory:

    ```
    bin/controller.sh login-db
    ```

    b. Run the following commands to set the database to the primary or secondary:

3. Restart the database for the change to take effect on the Appserver:

    ```
    bin/platform-admin.sh stop-controller-appserver --with-db
    bin/platform-admin.sh start-controller-appserver --with-db
    ```

    If the secondary Appserver is already in a shutdown state, then there is no need to restart the database.
4. Verify the replication is healthy:

    ```
    show slave status\G
    ```

    `Slave_IO_Running` and `Slave_SQL_Running` should show `Yes`.

You may now retry the discover and upgrade job.

Failover Prevention

If failover is prevented on your Controller HA configuration, it may be due to one of two scenarios:

- The secondary database is down. Failover cannot occur when the secondary database is not running.
  To fix this issue:
    1. Restart the secondary database by running the following command on the secondary host:

    ```
    bin/controller.sh start-db
    ```

  If this does not enable failover, then it may be due to the second scenario.

- Database replication is not healthy. Failover is not allowed when the database replication is not healthy.
  There are various reasons why this may be the case. Please work closely with your AppDynamics account representative to correct the issue.

Migrate to the New HA Module Using Enterprise Console

As part of the new Controller HA Module in the Enterprise Console, Enterprise Console no longer manages the Controller failover. Instead, the new HA Module installs the Controller watchdog on the Controller hosts and the Controller hosts are now responsible for performing a failover. The new HA Module is packaged in Enterprise Console and allows you to migrate seamlessly from older HA implementations, such as the HA Toolkit (HATK), to this new HA Module. The new HA Module is included with the latest version of Enterprise Console, and is installed when you install or upgrade your Controller. However, it is not activated until you migrate an HA pair.

Who Should Use the HA Module?

- If you are a new user just starting to implement AppDynamics HA, then you should use the HA Module (this is the default option).
- If you are an existing user and you prefer to use the Enterprise Console UI integration instead of the command line (CLI) and HATK, then you should use the HA Module.
- If you are an existing user and you do not want to change existing DevOps, conduct additional training, or would like to continue using HATK features, then use HATK.

> ⓘ If your Controller version is earlier than version 4.5.13, then you must use the HA Toolkit (HATK) instead of the HA Module to install and configure High Availability.

Before You Begin

In addition the Prerequisites for High Availability, ensure the following requirements are met:

- Both servers have the same Linux username and groupname.
- Both servers have the same AppDynamics directory structure, same AppDynamics pathname on both servers without using symbolic links.

Migrate a Controller High Availability Pair

This section describes the Enterprise Console and Controller HA Pair upgrade processes required for two different scenarios:

- Scenario One: Deployment currently using both Enterprise Console and HA Toolkit
- Scenario Two: Deployment where Enterprise Console alone manages the HA Pair

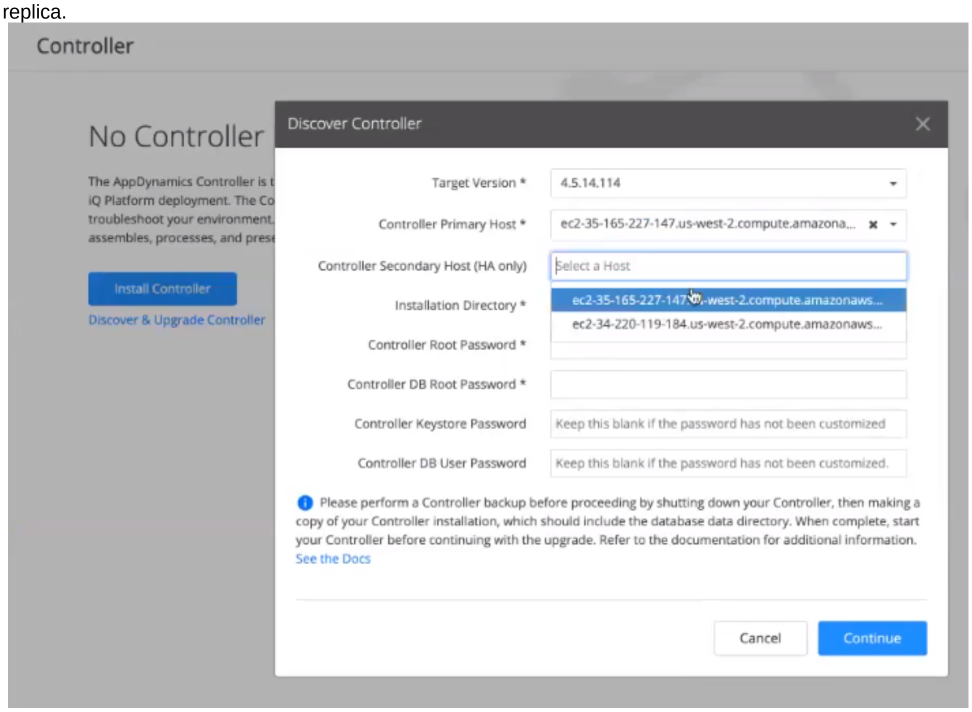
Scenario One: Upgrade Deployment that uses Enterprise Console and HA Tookit

1. Download and install the latest version of Enterprise Console from the Downloads page.
2. Open Enterprise Console and select **Controller**. In the Controller list, it displays only the primary Controller of the HA Pair. If you have an existing pair of HA Controllers which are not managed by Enterprise Console, you need to forget the Controller from within Enterprise Console. It will only show the primary Controller.
3. Select the Controller and select **Remove**.

> ⚠ In the Remove Controller dialog, deselect **Remove Binaries**. At this point, you are just removing the Controller from Enterprise Console but not the AppDynamics software that is running on the HA Controllers.

4. Before we can activate the new HA Module, we need to discover both Controllers from within Enterprise Console. Once the remove Controller job completes, select **Controller > Discover & Upgrade Controller**.

5. Note that in the Discover Controller dialog, you specify the hostnames of both the Controller Primary Host, and the Controller Secondary Host, or replica.



The AppDynamics Controller is t… iQ Platform deployment. The Co… troubleshoot your environment. … assembles, processes, and prese…

Target Version * 4.5.14.114

Controller Primary Host * ec2-35-165-227-147.us-west-2.compute.amazona...

Controller Secondary Host (HA only) Select a Host

- ec2-35-165-227-147.us-west-2.compute.amazonaws...
- ec2-34-220-119-184.us-west-2.compute.amazonaws...

Installation Directory *

Controller Root Password *

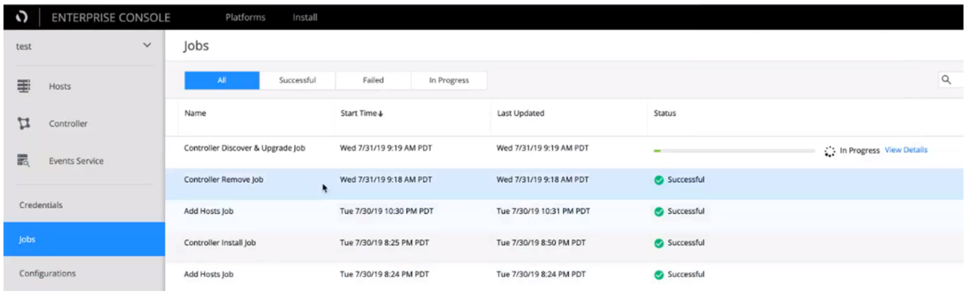
Controller DB Root Password *

Controller Keystore Password Keep this blank if the password has not been customized

Controller DB User Password Keep this blank if the password has not been customized.

Please perform a Controller backup before proceeding by shutting down your Controller, then making a copy of your Controller installation, which should include the database data directory. When complete, start your Controller before continuing with the upgrade. Refer to the documentation for additional information. See the Docs

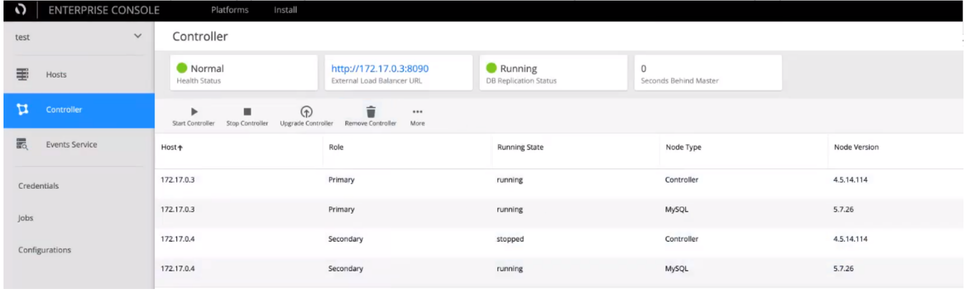
6. Complete the fields of the Discover Controller dialog and select **Continue**. This adds the HA Pair to Enterprise Console where you can then manage and upgrade.
   Access the Jobs page to see several jobs that have completed successfully. The Controller Discover & Upgrade Job will take a while to complete. Select **View Details** to track the progress of the tasks involved to discover and upgrade the Controller.



Jobs

Name	Start Time ↓	Last Updated	Status
Controller Discover & Upgrade Job	Wed 7/31/19 9:19 AM PDT	Wed 7/31/19 9:19 AM PDT	In Progress View Details
Controller Remove Job	Wed 7/31/19 9:18 AM PDT	Wed 7/31/19 9:18 AM PDT	✓ Successful
Add Hosts Job	Tue 7/30/19 10:30 PM PDT	Tue 7/30/19 10:31 PM PDT	✓ Successful
Controller Install Job	Tue 7/30/19 8:25 PM PDT	Tue 7/30/19 8:50 PM PDT	✓ Successful
Add Hosts Job	Tue 7/30/19 8:24 PM PDT	Tue 7/30/19 8:24 PM PDT	✓ Successful

When the discovery and upgrade process for the HA Pair is complete, the Controller page should be similar to the following:



Controller

● Normal — Health Status
http://172.17.0.3:8090 — External Load Balancer URL
● Running — DB Replication Status
0 — Seconds Behind Master

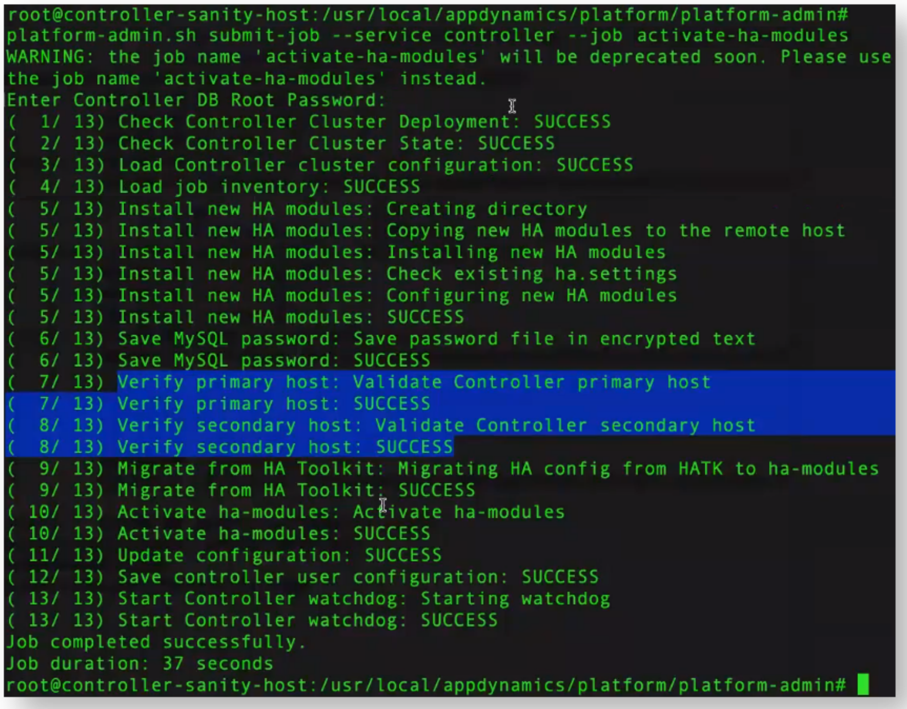
Host ↑	Role	Running State	Node Type	Node Version
172.17.0.3	Primary	running	Controller	4.5.14.114
172.17.0.3	Primary	running	MySQL	5.7.26
172.17.0.4	Secondary	stopped	Controller	4.5.14.114
172.17.0.4	Secondary	running	MySQL	5.7.26

Now, Enterprise Console knows about the HA Pair and has copied the new HA Module to the primary and secondary Controllers in the HA Pair.

7. To activate the HA Pair, open a command shell on the Enterprise Console host and enter the following:

```
platform-admin.sh submit-job --service controller  --job activate-ha-modules
```

You should see output similar to the following:



```
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
platform-admin.sh submit-job --service controller --job activate-ha-modules
WARNING: the job name 'activate-ha-modules' will be deprecated soon. Please use
the job name 'activate-ha-modules' instead.
Enter Controller DB Root Password:
(  1/ 13) Check Controller Cluster Deployment: SUCCESS
(  2/ 13) Check Controller Cluster State: SUCCESS
(  3/ 13) Load Controller cluster configuration: SUCCESS
(  4/ 13) Load job inventory: SUCCESS
(  5/ 13) Install new HA modules: Creating directory
(  5/ 13) Install new HA modules: Copying new HA modules to the remote host
(  5/ 13) Install new HA modules: Installing new HA modules
(  5/ 13) Install new HA modules: Check existing ha.settings
(  5/ 13) Install new HA modules: Configuring new HA modules
(  5/ 13) Install new HA modules: SUCCESS
(  6/ 13) Save MySQL password: Save password file in encrypted text
(  6/ 13) Save MySQL password: SUCCESS
(  7/ 13) Verify primary host: Validate Controller primary host
(  7/ 13) Verify primary host: SUCCESS
(  8/ 13) Verify secondary host: Validate Controller secondary host
(  8/ 13) Verify secondary host: SUCCESS
(  9/ 13) Migrate from HA Toolkit: Migrating HA config from HATK to ha-modules
(  9/ 13) Migrate from HA Toolkit: SUCCESS
( 10/ 13) Activate ha-modules: Activate ha-modules
( 10/ 13) Activate ha-modules: SUCCESS
( 11/ 13) Update configuration: SUCCESS
( 12/ 13) Save controller user configuration: SUCCESS
( 13/ 13) Start Controller watchdog: Starting watchdog
( 13/ 13) Start Controller watchdog: SUCCESS
Job completed successfully.
Job duration: 37 seconds
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
```

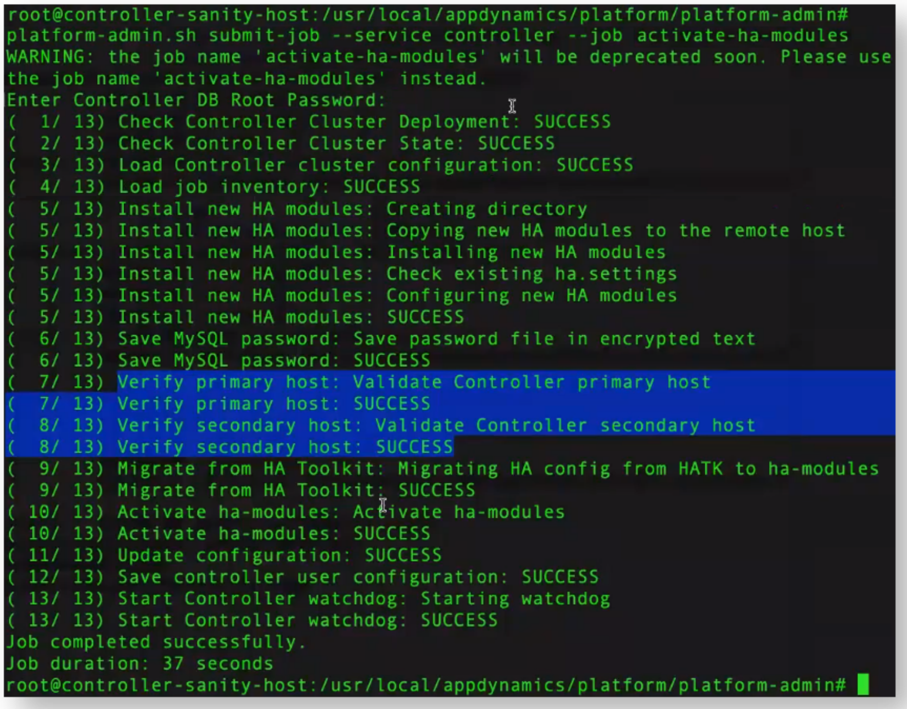
8. Verify that the Controller HA failover is working by terminating the Primary Controller and allow Controller Watchdog to trigger a failover. Then, wait a few minutes to ensure AppDynamics agents are reporting metrics to the Controller.

Scenario Two: Upgrade Deployment Managed by Enterprise Console Alone

1. Download and install the latest version of Enterprise Console from the Downloads page.
2. Upgrade the installed version of Enterprise Console by running the installer on the Enterprise Console host.
3. Upgrade both Controllers by selecting **Upgrade Controller**.
4. To activate the HA Pair, open a command shell on the Enterprise Console host and enter:

```
platform-admin.sh submit-job --service controller  --job activate-ha-modules
```

You should see output similar to the following:



```
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
platform-admin.sh submit-job --service controller --job activate-ha-modules
WARNING: the job name 'activate-ha-modules' will be deprecated soon. Please use
the job name 'activate-ha-modules' instead.
Enter Controller DB Root Password:
(  1/ 13) Check Controller Cluster Deployment: SUCCESS
(  2/ 13) Check Controller Cluster State: SUCCESS
(  3/ 13) Load Controller cluster configuration: SUCCESS
(  4/ 13) Load job inventory: SUCCESS
(  5/ 13) Install new HA modules: Creating directory
(  5/ 13) Install new HA modules: Copying new HA modules to the remote host
(  5/ 13) Install new HA modules: Installing new HA modules
(  5/ 13) Install new HA modules: Check existing ha.settings
(  5/ 13) Install new HA modules: Configuring new HA modules
(  5/ 13) Install new HA modules: SUCCESS
(  6/ 13) Save MySQL password: Save password file in encrypted text
(  6/ 13) Save MySQL password: SUCCESS
(  7/ 13) Verify primary host: Validate Controller primary host
(  7/ 13) Verify primary host: SUCCESS
(  8/ 13) Verify secondary host: Validate Controller secondary host
(  8/ 13) Verify secondary host: SUCCESS
(  9/ 13) Migrate from HA Toolkit: Migrating HA config from HATK to ha-modules
(  9/ 13) Migrate from HA Toolkit: SUCCESS
( 10/ 13) Activate ha-modules: Activate ha-modules
( 10/ 13) Activate ha-modules: SUCCESS
( 11/ 13) Update configuration: SUCCESS
( 12/ 13) Save controller user configuration: SUCCESS
( 13/ 13) Start Controller watchdog: Starting watchdog
( 13/ 13) Start Controller watchdog: SUCCESS
Job completed successfully.
Job duration: 37 seconds
root@controller-sanity-host:/usr/local/appdynamics/platform/platform-admin#
```

5. Verify that the Controller HA failover is working by terminating the Primary Controller and allows Controller Watchdog to trigger a failover. Then wait a few minutes to ensure AppDynamics agents are reporting metrics to the Controller.
6. When you enable the HA module, the HATK folder is renamed which results in broken services.
   To re-install the services correctly on a pair of HA servers, you must enter the following on the primary and secondary servers:
   a. Log in with `root` privileges.
   b. Change directories to: `<controller_home>/controller-ha/init`
   c. Using the HA module, you must first remove the services installed by HATK by running this script:

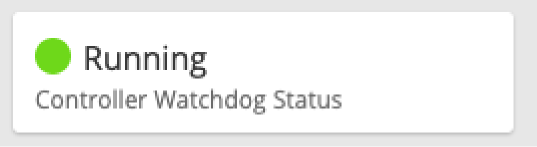
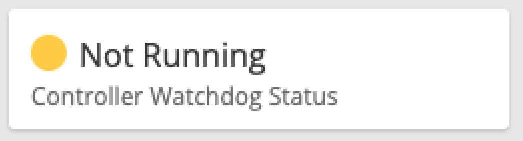
      ```
      ./uninstall-init.sh
      ```

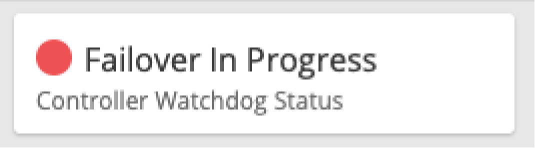
   d. Run `install-init.sh` and install the same services with one of the following options to elevate the user privilege:

      - `-c` #use setuid c wrapper
      - `-s` #use sudo
      - `-p` #use prune wrapper
      - `-x` #use user privilege wrapper

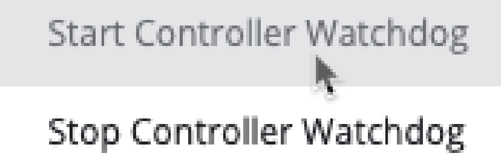
Watchdog Widget

After activating the HA Module, a new widget displays in the Enterprise Console UI indicating the Controller Watchdog status. It has three states:

Controller Watchdog Status	Definition
	The Controller watchdog process is constantly checking the health of your primary Controller. No user action is required.
	In the event that your primary Controller goes down, the failover is not automatic. AppDynamics recommends that you start the Controller watchdog using the *Start Controller Watchdog* option in Enterprise Console.
	Failover is currently in progress where your primary Controller and secondary Controller switch roles

 Failover In Progress Controller Watchdog Status	

Prior to performing any maintenance operations on your Primary Controller host (which could result in stopping the Controller), AppDynamics recommends that you select **Stop Controller Watchdog** to prevent a failover from initiating. Use the Enterprise Console UI to start and stop watchdog by clicking the watchdog widget, and then selecting **Stop** or **Start Controller Watchdog**.



Start and Stop the Controller Watchdog with CLI

You can start and stop the Controller Watchdog using the following commands:

Stop the Controller Watchdog

```
./platform-admin.sh submit-job --job stop-controller-watchdog --service controller
```

Start the Controller Watchdog

```
./platform-admin.sh submit-job --job start-controller-watchdog --service controller
```

Administer the Controller

This section contains pages on administering the Controller. Depending on how you installed and deployed the platform, you may have more than one method to administer the platform.

If you use the Enterprise Console, many tasks can be done through the GUI or command line. For information about how to use the Enterprise Console, see the pages in the Enterprise Console section.

Additionally, AppDynamics provides command-line tools for common operations, such as starting and stopping the platform components and their services. Some tasks involve using the administration interface for the underlying Glassfish application server, either the web interface or the command-line tool.

- Administer Controller configurations via the Enterprise Console.
- Other configurations:
  - `admin.jsp`: Access basic operational settings for the AppDynamics installation, including data retention settings, tenancy mode, and more.
  - Glassfish administration tool: Use the Glassfish admin tool, as admin, to configure settings in the underlying application server.

The pages in this section describe how to use the tools along with other administration tasks.

Start or Stop the Controller

You can start or stop the Controller, and also check the Controller health from the Enterprise Console GUI or CLI.

Scripts Used to Start or Stop the Controller

The scripts to start and stop the Controller and other AppDynamics platform processes are located in the `platform-admin/bin` directory for standalone or secondary HA Controllers. Using the `platform-admin` script, you can start the platform processes individually or all at once.

> To avoid the possibility of data corruption errors, be sure to stop the application server and database gracefully by using the stop scripts before shutting off or rebooting the machine on which the Controller is running.

You can start and stop the Controller and Controller services on the Controller page in the GUI or from the command line. When you start and stop the Controller, services and processes related to the Controller also start and stop, including the Reporting Service. If you use the GUI to start and stop the Controller, specify that you want to stop MySQL if you want to also stop the Controller Database.

> If your Enterprise Console manages multiple platforms, to distinguish the Controller platform, you must use the command line for standalone or secondary HA Controllers and specify the `--platform-name <name_of_the_platform>`.
>
> For example:
>
> ```
> platform-admin.sh start-controller-db --platform-name <name_of_the_platform>
> platform-admin.sh start-controller-appserver --platform-name <name_of_the_platform>
> ```
>
> To see all options, run `platform-admin.sh list-jobs --service controller --platform-name <platform-name>` from the command line. For more information, see Enterprise Console Command Line or Administer the Enterprise Console.

The Enterprise Console has a max wait time of 45 minutes when starting or stopping the Controller. You can set a timeout which exits the command and returns a failure by appending `--args controllerProcessTimeoutInMin=<minutes>` to the end of your start or stop command.

How to Start or Stop a Standalone or Secondary High Availability Controller

The commands below only apply to standalone and secondary HA Controllers. See Starting or Stopping a Primary Controller for information on how to start or stop primary Controllers.

Start and Stop the Controller App Server

> - When using the Enterprise Console, starting or stopping the Controller will also start or stop the Reporting Service.
> - You can only start or stop the Secondary Controller.

Start the Controller App Server

The `start-controller-appserver` command starts the database automatically.

Stop the Controller App Server

The `stop-controller-appserver` command does not stop the database.

Start and Stop the Controller Database

Start the Controller Database

Stop the Controller Database

How to Start or Stop a Primary Controller in High Availability Pairs

Use the following commands to start or stop primary Controllers in HA pairs. You must log in to the primary Controller host before running the scripts to start the `controller.sh` processes.

> ⚠
>
> - You should disable auto-failover before restarting a primary Controller. Do not forget to reenable auto-failover afterward.
> - If you enabled auto-failover in the Enterprise Console, and you stopped the app server to update certifications, the Enterprise Console will trigger a failover if it takes longer than five minutes to update.
> - If you are using a combination of the Enterprise Console with the High Availability Toolkit (HATK), then you can start or stop the Controller using services.

Start and Stop the Controller

To start the Controller, run this script from the Controller home:

To stop the Controller:

Start and Stop the Controller App Server

To start the app server, run this script from the Controller home:

To stop the app server, run this command:

Start and Stop the Controller Database

On Linux

To start the database, run this script from the Controller home:

```
bin/controller.sh start-db
```

To stop the database:

```
bin/controller.sh stop-db
```

How to Check Controller Health

On Linux

To check on the health of the Controller, run:

```
bin/platform-admin.sh check-controller-health
```

The following output shows the status of the Controller and its uptime:

```
Controller status : HEALTHY; Started 6 seconds ago.
```

Use a Reverse Proxy

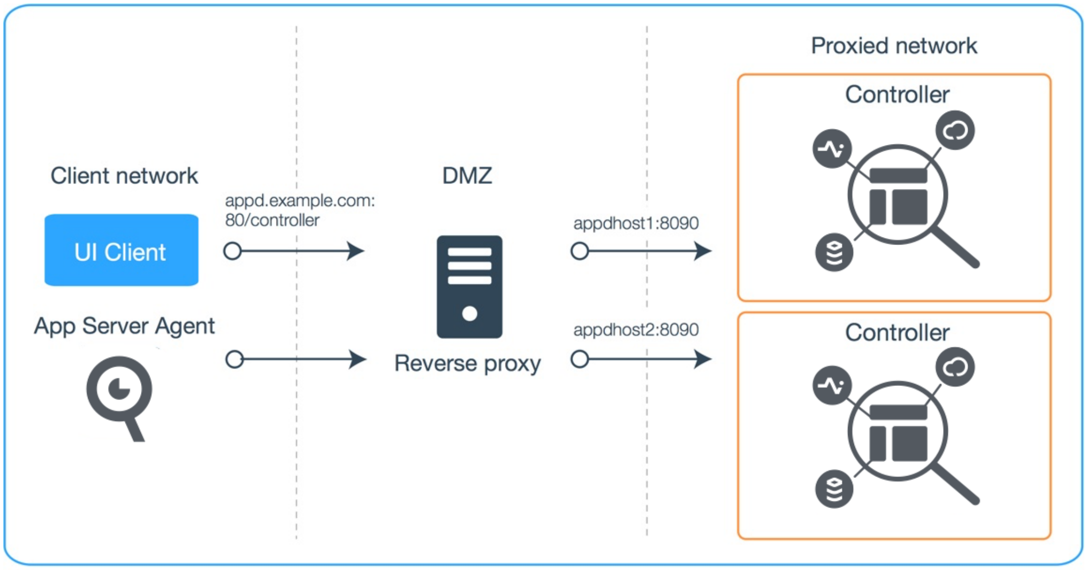
This page describes how to set up the Controller behind a reverse proxy.

Reverse Proxy Architectures

The AppDynamics Controller is often deployed behind a reverse proxy. The proxy presents a virtual IP address to external connections, such as agents and browser clients. The proxy often resides in the DMZ for the network, where it terminates SSL connections from external clients.

The proxy provides a security layer for the Controller, but it also enables you to move a Controller to another machine or switch between high availability pairs without having to reconfigure agents.

The following diagram illustrates the scenario:



As shown, the reverse proxy listens for incoming requests on a given path, /controller in this case, on port 80. It forwards matching requests to the HTTP listening port of the primary Controller at `appdhost1:8090`.

In terms of network impact in this scenario, switching active Controllers from the primary to the secondary in this scenario only requires the administrator to update the routing policy at the proxy so that traffic directed to the secondary instead of the primary.

These instructions describe how to set up the Controller with a reverse proxy. They also provide sample configurations for a few specific types of proxies, including NGINX and Apache Web Server.

This information is intended for illustration purposes only. The configuration requirements for your own deployment are likely to vary depending on your existing environment, the applications being monitored, and the policies of your organization.

While AppDynamics supports Controllers that are deployed with a reverse proxy, AppDynamics Support cannot guarantee help with specific set up questions and issues particular for your environment or the type of proxy you are using. For this type of information, please consult the documentation provided with your proxy technology. Alternatively, ask the AppDynamics community.

General Guidelines

The following describes general requirements, considerations, and features for deploying the AppDynamics Controller and App Agents with a reverse proxy.

- Set the deep link JVM option, `-Dappdynamics.controller.ui.deeplink.url`, to the value of the Controller URL. Use either the hostname or IP address of the Controller host (if directly accessible to clients) or to the VIP for the Controller as exposed at the proxy in the following format:

```
modifyJvmOptions add Dappdynamics.controller.ui.deeplink.url=http[s]://controller.corp.example.com[:port]
/controller
```

Use the URI scheme (http or https), hostname and port number appropriate for your Controller. The Controller uses the deep link value to compose URLs it exposes in the UI.

- If terminating SSL at the proxy, also set the following JVM options:

```
-Dappdynamics.controller.services.hostName=<external_DNS_hostname>
-Dappdynamics.controller.services.port=<external_port_usually_443>
```

- You should use the Enterprise Console's Controller Settings page to edit the JVM options to retain your settings. See Update Platform Configurations for more information.
- If the proxy sits between monitored tiers in the application, make sure that the proxy passes through the custom header that AppDynamics adds for traffic correlation, singularity header. Most proxies pass through custom headers by default.
- For App Agents, the Controller Host and Controller Port connection settings should point to the VIP or hostname and port exposed for the Controller at the reverse proxy. For details see Agent-to-Controller Connections.
- If using SSL from the agent to the proxy, ensure that the security protocols used between the App Agent and proxy are compatible. See the compatibility table for the SSL protocol used by each version of the agent.
- If the proxy (or another network device) needs to check for the availability of the Controller, it can use Controller REST resource at: `http://<host>:<port>/controller/rest/serverstatus`. If the Controller is active and if in high availability mode, is the primary, it returns an XML response similar to this one:

```
<serverstatus vendorid="" version="1">
   <available>true</available>
   <serverid/>
   <serverinfo>
       <vendorname>AppDynamics</vendorname>
       <productname>AppDynamics Application Performance Management</productname>
       <serverversion>003-008-000-000</serverversion>
   </serverinfo>
</serverstatus>
```

If the Controller is in standby mode, this resource returns a 503 response.

The following sections provide notes and sample configurations for a few specific types of proxies, including Nginx and Apache Web Server.

Using Nginx as a Simple HTTP Reverse Proxy

Nginx is a commonly used web server and reverse proxy available at http://nginx.org/.

To use Nginx as a reverse proxy for the Controller, simply include the Controller as the upstream server in the Nginx configuration. If deploying two Controllers in a high availability pair arrangement, include the addresses of both the primary and secondary Controllers in the upstream server definition.

The following steps walk you through the set up at a high level. It assumes you have already installed the Controller and have an Nginx instance, and you only need to modify the existing configuration to have Nginx route traffic to the Controller.

To route Controller traffic through an Nginx reverse proxy

1. Add a JVM option named `-Dappdynamics.controller.ui.deeplink.url`. Set its value to the URL for the Controller, as described in the guidelines above.
2. Shut down the Controller.
3. If terminating SSL at the proxy, also set the `-Dappdynamics.controller.services.hostName` and `-Dappdynamics.controller.services.port` JVM options to the external DNS hostname for the Controller and the external port number, typically 443.
4. In the Nginx home directory on the reverse proxy machine, open the `conf/nginx.conf` file for editing.
5. In the configuration file, add a cluster definition the specifies each Controller as an upstream server. For example:

```
 upstream appdcontroller {
  server 127.0.15.11:8090 fail_timeout=0;
}

server {
    listen 80;
    server_name appdcontroller.example.com;

    expires 0;
    add_header Cache-Control private;

    location / {
        proxy_set_header    Host $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-Proto https;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;

        proxy_pass          http://appdcontroller;
    }
}
```

In the sample, the Controller resides on 127.0.15.11 and has the fully qualified domain name `appdcontroller.example.com`.

6. Restart the Nginx server to have the change take effect.
7. Restart the Controller.

After the Controller starts, it should be able to receive traffic through Nginx. As an initial test of the connection, try opening the Controller UI via the proxy, that is, in a browser, go to `http://<virtualip>:80/controller`. For the App Agents, you'll need to configure their proxy host and port settings as described in the general guidelines above.

Using Apache as a Reverse Proxy

To use Apache as a reverse proxy, you need to make sure the appropriate Apache module is installed and enabled in your Apache instance. For HTTP proxying, this is typically `mod_proxy_http`. The `mod_proxy_http` module supports proxied connections that use HTTP or HTTPS.

To configure Apache with mod_proxy_http

1. Add a JVM option named `-Dappdynamics.controller.ui.deeplink.url`. Set its value to the URL for the Controller, as described in the guidelines above.
2. Shut down the Controller.
3. If terminating SSL at the proxy, also set the `-Dappdynamics.controller.services.hostName` and `-Dappdynamics.controller.services.port` JVM options to the external DNS hostname for the Controller and the external port number, typically 443.
4. On the machine that runs Apache, check whether the required modules are already loaded by your Apache instance by running this command:

   ```
   apache2ctl -M
   ```

   In the output, look for proxy modules as follows:

   ```
   proxy_module (shared)
   proxy_http_module (shared)
   ```

   The `proxy_module` is a dependency for `proxy_module_http`.
5. If they are not loaded, enable the Apache module as appropriate for your distribution of Apache. For example, on Debian/Ubuntu:
   a. Enter the following:

      ```
      sudo a2enmod proxy_http
      ```

   b. Restart Apache:

      ```
      sudo service apache2 restart
      ```

6. Add the proxy configuration to Apache. For example, a configuration that directs clients requests to the standard web port 80 at the proxy host to the Controller could look similar to this:

```
<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>

ProxyRequests       Off
ProxyPreserveHost   On

ProxyPass /controller http://controller.example.com:8090/controller
ProxyPassReverse /controller  http://controller.example.com:8090/controller
```

7. Apply your configuration changes by reloading Apache modules. For example, enter:

```
sudo service apache2 reload
```

8. Start the Controller.

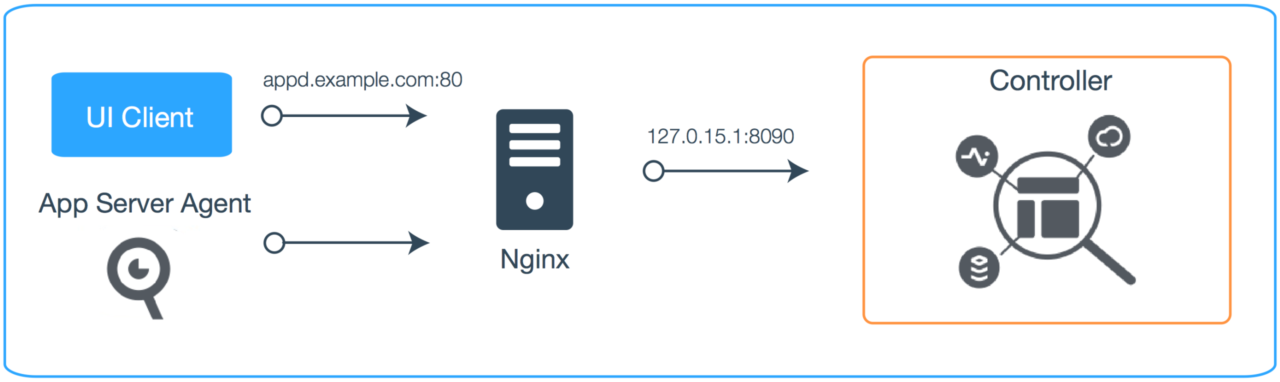
After the Controller starts, test the connection by opening a browser to the Controller UI as exposed by the proxy. To enable AppDynamics App Agents to connect through the proxy, be sure to set the proxy host and port settings in the proxy, as described in the general guidelines above. Also, be sure to apply any of the other general guidelines described in the general guidelines above.

Configure SSL Termination at the Reverse Proxy

This section describes how to set up security when the client-side connection to the proxy uses SSL that's terminated at the proxy. This assumes that the proxy and Controller are in a secured data center and the App Agents or UI browser client connections are from a potentially insecure network.

Terminating SSL at a proxy offloads the burden of SSL processing from the Controller to the proxy. This configuration is strongly recommended when deploying the Controller to large scale, high workload environments. Terminating SSL at a proxy also provides the benefit of having a central point in the data center for the security certificate and for key management.

This section provides a sample configuration for Nginx, but the concepts translate to other types of reverse proxies as well.



Configure the Proxy for SSL Termination

To perform SSL termination at the reverse proxy, you need to:

- Ensure that the App Agents can establish a secure connection with the proxy. See Agent and Controller Compatibility for SSL settings for various versions of the agent. Ensure that the proxy includes a server certificate signed by an authority that is trusted by the agent. Otherwise, you will need to install the proxy machine's server key.
- If using .NET App Agents in your environment, verify that the reverse proxy server uses a server certificate signed by a certificate authority (CA). The .NET App Agent does not permit SSL connections based on a self-signed server certificate.
- Configure the proxy to forward traffic between it and the Controller to a secure port between it and the client.

- The client App Agents and browser clients under this configuration *must use* the secure port to communicate with the Controller (i.e., the proxy). Configuring a mixed channel on the Controller as described here causes the agents to perform as if they were using a secure port. Therefore, you need to ensure that they use a secure port only.

A complete example configuration with Nginx performing SSL termination for the Controller would look something like this:

```
 upstream appdcontroller {
  server 127.0.15.11:8191 fail_timeout=0;
}

server {
    listen 80;
    server_name appdcontroller.example.com;
    return 301 https://$host$request_uri;
}
server {
    listen 443;
    server_name appdcontroller.example.com;

    ssl on;
    ssl_certificate /etc/nginx/server.crt;
    ssl_certificate_key /etc/nginx/server.key);

    ssl_session_timeout  5m;
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers  ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+EXP;
    ssl_prefer_server_ciphers   on;

    expires 0;
    add_header Cache-Control private;

    location / {
        proxy_set_header    Host $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-Proto https;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_redirect      http:// https://;
        proxy_pass          http://appdcontroller;
    }
}
```

⚠ TLSv1 should only be enabled if absolutely necessary.

This example builds on the configuration shown in the simple passthrough example. In this one, any request received on the non-SSL port 80 is routed to port 443. The server for port 443 contains the settings for SSL termination. The `ssl_certificate_key` and `ssl_certificate` directives should identify the location of the server security certificate and key for the proxy.

The configuration also indicates the SSL protocols and ciphers accepted for connections. The security settings need to be compatible with the AppDynamics App Agent security capabilities, as described on the Agent and Controller Compatibility page.

Cookie Security

The Controller sets the secure flag on the `X-CSRF-TOKEN` cookie sent over HTTPS. The secure flag ensures that clients only transmit the cookies on secure connections.

If you terminate HTTPS at a reverse proxy in front of the Controller, the Controller does not set cookie security by default because connections to the Controller would occur over HTTP.

To ensure cookie security, configure the reverse proxy to include the `secure` statement in the `set-cookie` statement. How to add the secure flag varies on the type of reverse proxy you use.

The following example shows how to set cookie security using HAProxy:

- If using HAProxy version 2.0 or earlier, enter:

```
rspirep ^(set-cookie:.*) \1;\ Secure
```

- If using HAProxy version 2.1 or later, enter:

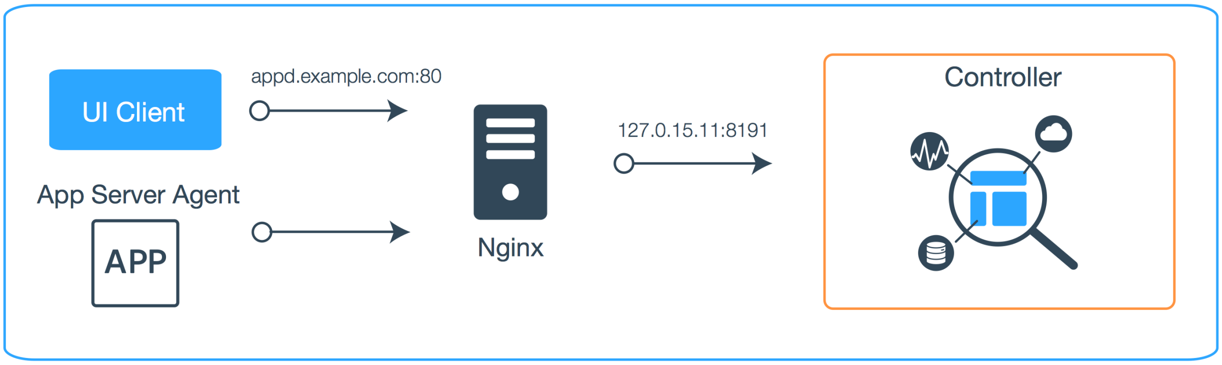
```
acl secured_cookie res.hdr(Set-Cookie),lower -m sub secure
http-response replace-header Set-Cookie (.*) \1;\ Secure if !secured_cookie
```

Using SSL from the Reverse Proxy to the Controller

Have the proxy connect to the Controller with SSL requires a minor modification to the proxy configuration. Simply specify the use of HTTPS as the protocol to connect to the backend or upstream server. In other words, for the Nginx configuration, this simply requires you to modify the `proxy_pass` value as follows:

```
proxy_pass          https://appdcontroller;
```

To complete the configuration, make sure you have configured SSL on the Controller as described in Controller SSL and Certificates.



Enable an Email Server

The SMTP email server must be configured to enable email and SMS notifications and digests to be sent by the Controller.

Permissions

For this activity, users need the predefined Account Owner role or another role with the **Configure Email / SMS** permission.

Configure an SMTP server

1. In the Controller UI, from the top navigation menu bar, click **Alert & Respond > Email/ SMS Configuration**.
2. Provide the connection information for the SMTP host and port.

    - A SaaS Controller should be preconfigured with the appropriate settings, but verify the settings as the following:

        a. **SMTP Host:** localhost
        b. **SMTP Port:** 25

      No authentication is needed.
    - For an on-premises controller, use a host and port settings for an SMTP server available in the controller deployment environment.
3. Customize sender address in notifications emails in the **From Address** field. By default, emails are sent by the root Controller user.
4. If the SMTP host requires authentication, configure the credentials in the **Authentication** settings.
5. If you want to add any text to the beginning of the notification, enter it in the **Notification Header Text** field.
6. If you are using SMS do one of the following:

    - Select **Default** and choose one of the available carriers from the pulldown menu.
    - Select **Custom** and enter the phone number receiving the message as <phone number>@<sms gateway>.

      For example, a mobile phone in the United States serviced by AT&T might be:

      ```
      4151234567@txt.att.net
      ```

      A mobile phone in the United Kingdom serviced by Textlocal might be:

      ```
      7412345678@txtlocal.co.uk
      ```

      See SMS gateway by country for information on most common SMS gateways.
7. Test the configuration by sending an email.
8. Save the settings.

Troubleshoot Notifications

If you do not receive notifications for health rule violations, it could be because the default SMTP server timeout period is too short. To troubleshoot, increase the value of the mail.smtp.socketiotimeout Controller Setting in the Administration Console. The default value is 30 seconds.

Update the Root User and Glassfish Admin Passwords

This page describes how to change the passwords for the root user and Glassfish admin for the Controller.

Root User and Account Owners

The root user is a built-in Controller user with global administrator privileges in the Controller environment. Only the root user can access the System Administration Console, where you can create and manage accounts in multi-tenant Controllers and configure global Controller settings in both single- or multi-tenant Controllers.

The root user is a superuser for the Controller. Unlike other types of users, you cannot remove the root user account or create other superuser accounts in the Controller. The password for the root user is set during installation, but you can change the root password in the Administration Console.

While the root user has global administrative privileges, account administrators act as administrators only within individual accounts in a multi-tenant Controller. It's typically the role of the root user to create accounts and an initial administrator for the account, and the role of each account administrator to create additional users within the account. See Roles and Permissions and Manage Users and Groups.

Change the Controller Root User Password

You can change the root user password from the AppDynamics Administration Console page.

To change the root user password:

1. Log in to the administration console as described in Access the Administration Console.
2. Click **Settings** ⚙ and choose **My Settings**.
3. Click **Edit > Change Password.**
4. Type the new password for the root user in the **New Password** and **Repeat New Password** fields.
5. Click **Save**.

> ⚠ Logging in to the Administration Console requires you to have the root user password. If you do not have the root user password, you need to reset it.

Reset Root User Password

If you have lost the AppDynamics root user password for your installation and need to reset it, follow these steps:

1. From the command line, change to the Controller's `bin` directory. For example, on Linux:

   ```
   cd <controller_home>/bin
   ```

2. Use the following script to log in to the Controller database of the Controller;
   - For Windows: `controller.bat login-db`
   - For Linux: `sh controller.sh login-db`

   You will see a MySQL prompt.
3. After running the script, you will be prompted to enter a password. Enter the root password for the Controller database.
4. From the MySQL prompt, enter the following SQL command to get root user details:

   ```
   select * from user where name='root' \G;
   ```

5. Use the following SQL command to change the password:

   ```
   update user set encrypted_password = sha1('<NewPassword>') where name = 'root';
   ```

   The hash for the password will be upgraded to PBKDF2 when you log in.
6. Restart the Appserver.

For information on setting the database root user password, see Controller Data and Backups.

Change the GlassFish Admin User Password

The Controller uses the built-in administrator account in the underlying GlassFish application server. The Enterprise Console connects to your underlying GlassFish administration server via the GlassFish admin password during Controller upgrades.

To update the GlassFish admin user password in the Enterprise Console, see the following steps:

1. Log in to the Enterprise Console and select the desired platform.
2. Select **Configurations** > **Controller Settings** > **Appserver Configurations**.
3. In the **Basic** tab, find **Glassfish Admin Password**. Enter the new password
4. Re-enter the new password in **Confirm Glassfish Admin Password**.
5. Click **Save**.

The password change takes immediate effect.

> ⚠ If you have updated the password in the **Configurations** tab, then you don't need to input the password again for the upgrade job.

Change the Controller Database Root User Password

> ⓘ If the Enterprise Console has NOT discovered the Controller:
>
> - Downtime is required to change the Controller database root user password. If you have installed a Controller HA pair, you must disable auto-failover to avoid an accidental failover while changing the password. For more details, see Automatic Failover.
> - You need to change the password on both of the Controller HA pairs.

To change the Controller database root user password:

1. Log in to the Controller host. From a command line, enter:

   ```
   cd <controller_home_dir>
   ```

2. To stop the App Server and the database, enter:

   ```
   bin/controller.sh stop
   ```

   > ⓘ If you are using MS Windows, you must use the Windows services to stop the Controller.

3. To start the database in insecure mode, enter:

   ```
   bin/controller.{sh|bat} start-db insecure
   ```

   The insecure option starts the database without password requirements. Use this option only to reset the password for the database. The option is similar to starting MySQL with the `--skip-grant-tables` option.
4. To log in to the database, enter:

   ```
   bin/controller.{sh|bat} login-db insecure
   ```

5. Use MySQL to run the following commands:
   a. To specify the Controller database, enter:

      ```
      use mysql;
      ```

b. To reload the MySQL grant tables, enter:

```
FLUSH PRIVILEGES;
```

c. To determine your MySQL version, enter:

```
select version();
```

d. Based on which MySQL version you are using:
MySQL 5.5 version: Configure the new password for the root user by entering:

```
update mysql.user set password=password('<new-password-here>') where user like 'root%';
```

MySQL 5.7 version: Configure the new password for the root user by entering:

```
update mysql.user set authentication_string=password('<new-password-here>') where user like
'root%';
```

e. To reload the MySQL grant tables, enter:

```
FLUSH PRIVILEGES;
```

f. To exit MySQL, enter:

```
quit
```

6. To stop the database, enter:

```
bin/controller.{sh|bat} stop-db
```

7. To start the App Server, enter:

```
bin/controller.sh start
```

> If you are using MS Windows, you must use the Windows services to start the Controller.

> In the case of a Controller HA pair, generate an obfuscated password file for the Controller Database root user using the below command on the primary Controller server. This command will generate the password file on both primary and secondary Controller servers.
>
> ```
> controller-ha/set_mysql_password_file.sh -p <new-password-here> -s
> <secondary_controller_hostname>
> ```

If the Enterprise Console has discovered the Controller:
To update the Controller database root password with the following steps:

1. Log in to the Enterprise Console and select the desired platform.
2. Select **Configurations** > **Controller Settings** > **Database Configurations**.
3. Enter the new password in **New password for Controller DB root user**.
4. Reenter the password in **Confirm New password for Controller DB root user**.
5. Select **Save**.

The password change takes immediate effect.

If you previously disabled auto-failover, you should now enable it.

Change the Controller DB User Password

Downtime is required to change the Controller database root user password. If you have installed a Controller HA pair, you must disable auto-failover to avoid an accidental failover while changing the password. For more details about disabling auto-failover, click Automatic Failover.

To change the Controller DB user password:

1. Log in to the Enterprise Console and select the desired platform.
2. Select **Configurations > Controller Settings > AppServer Configurations**.
3. In the **Basic** tab, enter the new password in **New password for Controller DB user**.
4. Re-enter the password in **Confirm New password for Controller DB user**.
5. Click **Save**.

The password change takes immediate effect.

Change the Controller Owner

This page provides instructions for changing the Controller owner.

You may need to change the user who is running the Controller services during a system migration or other event.

The procedure varies based on whether you are using the Enterprise Console.

In order to change the owner of the Controller, complete the following steps.

1. Stop all running Controller services using the following command in the machine terminal.
   - `./controller.sh stop`
2. Change the username and user group of the Controller directory.
   - `chown -R <New User>:<User Group> <Controller Folder>`
3. Update the new user in the `db.cnf` file located at `Controller/db/db.cnf`.
   - `  user=<New User>`
4. Start the Controller.
   - `./controller.sh start`

If you are not using the Enterprise Console

1. As the current user running the Controller services, shut down the Controller process:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh stop
   ```

2. Change the ownership (recursively) of the entire Controller directory to the new user. In this example, `appdynamics:admin` is the `user:group`, respectively:

   ```
   chown -R appdynamics:admin CONTROLLER_HOME_DIR/
   ```

3. If the Controller's data directory is outside of the `root` Controller's folder, then you must also change the owner of the database data files:

   ```
   chown -R appdynamics:admin .../data/
   ```

4. Change the user to the new username:

   ```
   CONTROLLER_HOME_DIR/db/db.cnf
   ```

5. Log in as the new user and start the Controller services:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh start
   ```

If you are using the Enterprise Console

1. Remove the Controller from the Enterprise Console by de-selecting the **Remove Binaries** option; otherwise, the binaries will be removed from the disk. To remove the Controller without uninstalling the Controller:

```
PLATFORM_HOME_DIR/bin/platform-admin.sh submit-job --service controller --job remove --args
removeBinaries=false --skip-confirm
```

2. As the current user running the Controller services, shut down the Controller process.

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh stop
   ```

3. Change the ownership (recursively) of the entire Controller directory to the new user. In this example, `appdynamics:admin` is the `user:group`, respectively:

   ```
   chown -R appdynamics:admin CONTROLLER_HOME_DIR/
   ```

4. If the Controller's data directory is outside of the `root` Controller's folder, then you must also change the owner of the database data files:

   ```
   chown -R appdynamics:admin .../data/
   ```

5. Change the user to the new username:

   ```
   CONTROLLER_HOME_DIR/db/db.cnf
   ```

6. Log in as the new user and start the Controller services:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh start
   ```

7. From the Enterprise Console, remove the hosts that were added:

   ```
   PLATFORM_HOME_DIR/bin/platform-admin.sh remove-dead-hosts --hosts $CONTROLLER_HOST --skip-confirm
   ```

8. Remove the credentials because the credentials are connected to the previous user.
9. Add the credentials using the new user, and then add the host.
10. Perform a Discover and Upgrade for the Controller.
11. (Optional) If you have installed the Linux services, then:
    a. Logged in as root, uninstall the services:

       ```
       HA/uninstall-init.sh
       ```

    b. Logged in as root, install the services using either the `-c` or `-s` options:

       ```
       HA/install-init.sh
       ```

Change the User Running the Controller Services

This page provides instructions for changing the user running the Controller services.

You may need to change the user who is running the Controller services during a system migration or other event.

The procedure varies based on whether you are using the Enterprise Console.

If you are not using the Enterprise Console

1. As the current user running the Controller services, shut down the Controller process:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh stop
   ```

2. Change the ownership (recursively) of the entire Controller directory to the new user. In this example, `appdynamics:admin` is the `user:group`, respectively:

   ```
   chown -R appdynamics:admin CONTROLLER_HOME_DIR/
   ```

3. If the Controller's data directory is outside of the `root` Controller's folder, then you must also change the owner of the database data files:

   ```
   chown -R appdynamics:admin .../data/
   ```

4. Change the user to the new username:

   ```
   CONTROLLER_HOME_DIR/db/db.cnf
   ```

5. Log in as the new user and start the Controller services:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh start
   ```

If you are using the Enterprise Console

1. Remove the Controller from the Enterprise Console by de-selecting the **Remove Binaries** option; otherwise, the binaries will be removed from the disk. To remove the Controller without uninstalling the Controller:

   ```
   PLATFORM_HOME_DIR/bin/platform-admin.sh submit-job --service controller --job remove --args
   removeBinaries=false --skip-confirm
   ```

2. As the current user running the Controller services, shut down the Controller process.

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh stop
   ```

3. Change the ownership (recursively) of the entire Controller directory to the new user. In this example, `appdynamics:admin` is the `user:group`, respectively:

   ```
   chown -R appdynamics:admin CONTROLLER_HOME_DIR/
   ```

4. If the Controller's data directory is outside of the `root` Controller's folder, then you must also change the owner of the database data files:

```
chown -R appdynamics:admin .../data/
```

5. Change the user to the new username:

   ```
   CONTROLLER_HOME_DIR/db/db.cnf
   ```

6. Log in as the new user and start the Controller services:

   ```
   CONTROLLER_HOME_DIR/bin/controller.sh start
   ```

7. From the Enterprise Console, remove the hosts that were added:

   ```
   PLATFORM_HOME_DIR/bin/platform-admin.sh remove-dead-hosts --hosts $CONTROLLER_HOST --skip-confirm
   ```

8. Remove the credentials because the credentials are connected to the previous user.
9. Add the credentials using the new user, and then add the host.
10. Perform a Discover and Upgrade for the Controller.
11. (Optional) If you have installed the Linux services, then:
    a. Logged in as root, uninstall the services:

       ```
       HA/uninstall-init.sh
       ```

    b. Logged in as root, install the services using either the `-c` or `-s` options:

       ```
       HA/install-init.sh
       ```

Access the Administration Console

This page provides information and access instructions for the AppDynamics Administration Console.

Deployment Support

-Premises

Related pages:

- Controller Settings for Machine Agents
- Controller Settings for Server Visibility
- Configure Controller Settings for Monitoring Database

> Do not confuse the AppDynamics Administration Console with the GlassFish application server administration console or the general application administration page in the Controller UI.

The AppDynamics Administration Console lets you configure certain global settings such as metric retention periods, UI notification triggers, tenancy mode, and accounts in multi-tenancy mode.

> AppDynamics recommends that you do not change Controller settings in the console unless under the guidance of an AppDynamics representative or as specifically directed by documentation.

Access the AppDynamics Administration Console

1. If you are logged into a Controller UI session with an account other than the root user, log out or open a new browser window in private (incognito) mode. If you do not, you will get an "Access Denied" error when you attempt to open the console page.
2. In the browser enter the URL of the Administration Console:

   ```
   http://<hostname>:<port>/controller/admin.jsp
   ```

   The console is served on the same port as the Controller UI, port 8090 by default.
3. Log into the system account with the root user password. The root user is a built-in global administrator for the Controller. Use the password you set for this user when installing. See Update the Root User and Glassfish Admin Passwords

   > The root user password is different from a normal AppDynamics account password. It is not the same as the account owner or account administrator password. If you are logged into the Controller using your current account, you need to log out of that account and then back in as the root user to access the Administration Console. You can change the Controller root user password if you wish. See Update the Root User and Glassfish Admin Passwords

Access the Glassfish Administration Console

In rare cases, you may need to log in to the AppDynamics Administration Console for the application server underlying the GlassFish server. The GlassFish administration console provides a browser interface for performing many of the same tasks you can perform using the admin command-line utility. You can access the console for the GlassFish server using the built-in user account named admin.

For security reasons, access to the GlassFish browser interface is limited to local machine access by default, so the following steps should be performed from a browser on the Controller machine. Attempts to access the console remotely trigger the error message "Secure Admin must be enabled to access the DAS remotely."

To access the GlassFish administration console:

1. From a web browser on the Controller machine, open the following URL:

   ```
   http://localhost:4848
   ```

   Note that port 4848 is the default port number for the GlassFish administration console, but it may have been set to another value at installation time. If the default port doesn't work and you are unsure of what port number to use, you can check the port configured for the network-listener element named admin-listener in the domain.xml file.
2. Log in as user admin.
   By default, the GlassFish user admin password is the same as the root user password for the Administration Console. You can change the GlassFish user admin password if you wish. See Update the Root User and Glassfish Admin Passwords

Modify the User Session Timeout

The Controller logs users out of Controller UI sessions after 60 minutes of inactivity by default. For an on-premises Controller, it's possible to modify the default timeout value, as follows:

1. Log in to the Administration Console as the AppDynamics root user.
2. Find and set the values for these properties:

    - `http.session.inactive.timeout`: The amount of time without a client request to the Controller after which the user session times out and the user will need to log in again to continue. The default is 3600 seconds (60 minutes).
    - `ui.inactivity.timeout`: The amount of time without user activity in the Controller UI after which the user session times out and the user will need to log in again to continue. The default is -1 (disabled).

Customize System Notifications

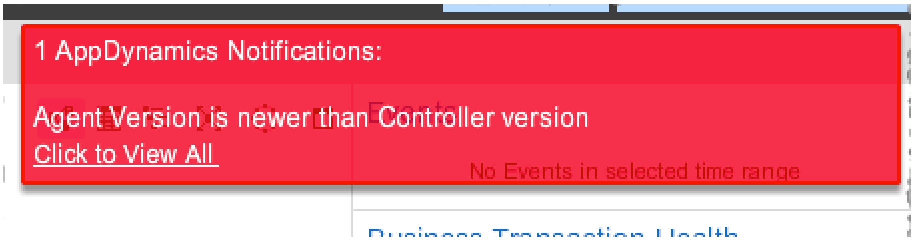
Related pages:

- Access the Administration Console

You can customize system events and system use notification messages from the Administration Console.

System Events Notification

Certain system events trigger event notification popups in the Controller UI.



1 AppDynamics Notifications:

Agent Version is newer than Controller version

Click to View All

No Events in selected time range

You can configure which type of events appear as notifications in the UI, as described here.

Configure Events that Trigger UI Notifications

1. Log in to the Administration Console.
2. Go to the **Controller Settings**.
3. Search for the `system.notification.event.types` property. The value of this property determines which type of events result in UI notifications.
4. Set the types of events you want to see by adding them to the comma-separated string in the dialog box. To disable notifications of a particular type of event, remove it from the list. Do not use spaces between commas.

Notification Event Types

Event Value	What This Event Notification Means
`LICENSE`	There is an issue with the status of your license.
`DISK_SPACE`	There is an issue with the amount of disk space left on your system.
`CONTROLLER_AGENT_VERSION_INCOMPATIBILITY`	A mismatch between the version of the agent and the version of the controller has been detected.
`CONTROLLER_EVENT_UPLOAD_LIMIT_REACHED`	The limit on the number of events per minute that can be uploaded to the controller from this account has been reached. Once the limit is reached no more events — other than certain key ones — are uploaded for that minute.
`CONTROLLER_RSD_UPLOAD_LIMIT_REACHED`	The limit on the number of request segment data (RSDs) per minute that can be uploaded to the controller from this account has been reached. RSDs are related to snapshots. Once the limit is reached no more RSDs — other than certain key ones — are uploaded for that minute.
`CONTROLLER_METRIC_REG_LIMIT_REACHED`	The limit for registering metrics for this account has been reached. No further metric registrations are accepted.
`CONTROLLER_ERROR_ADD_REG_LIMIT_REACHED`	The limit for registering error Application Diagnostic Data (ADDs) for this account has been reached. No further error ADD registration is accepted.
`CONTROLLER_ASYNC_ADD_REG_LIMIT_REACHED`	The limit for registering async ADDs for this account has been reached. No further async ADD registration is accepted.

`CONTROLLER_STACKTRACE_ADD_REG_LIMIT_REACHED`	The limit for registering StackTrace ADDs for this account has been reached. No further StackTrace ADD registration is accepted.
`AGENT_ADD_BLACKLIST_REG_LIMIT_REACHED`	If the Agent attempts to register an ADD above the limit, the Controller rejects the attempt and adds the ADD to a blacklist. There is a limit to the size of the blacklist. This event indicates that that limit has been reached.
`AGENT_METRIC_BLACKLIST_REG_LIMIT_REACHED`	If the Agent attempts to register a metric above the limit, the Controller rejects the attempt and adds the metric to a blacklist. There is a limit to the size of the blacklist. This event indicates that that limit has been reached.

System Use Notification

The system use notification is an optional and configurable message that includes information on privacy and security notices. If enabled, the displayed message must be acknowledged before granting the user further access.

Configure System Use Notification

1. Log in to the Administration Console.
2. Go to the **Controller Settings**.
3. Search for the `system.use.notification.message` property. The value of this property is the message of the system use.
4. Enter your post-login message, which will be displayed every time a user logs in, informing the user of the system usage requirements. There is a 1000 character limit.
   Here is an example message:

```
This is a U.S. Government computer system, which may be accessed and used only for authorized Government
business by authorized personnel. Unauthorized access or use of this computer system may subject
violators to criminal, civil, and/or administrative action. All information on this computer system may
be intercepted, recorded, read, copied, and disclosed by and to authorized personnel for official
purposes, including criminal investigations. Such information includes sensitive data encrypted to
comply with confidentiality and privacy requirements. Access or use of this computer system by any
person, whether authorized or unauthorized, constitutes consent to these terms. There is no right of
privacy in this system.
```

Multi-Tenant Controller Accounts

This page describes how to create and manage accounts in a multi-tenant Controller. The tenant mode determines whether the Controller UI offers single or multiple environments. See Controller Deployment.

Switch from Single-Tenant to Multi-Tenant Mode

> ⚠ Switching from single-tenancy to multi-tenancy mode is supported. However, switching from multi-tenancy to single-tenancy is not. Take precautions to ensure multi-tenancy is the correct mode for your environment.

If multi-tenancy is enabled for an on-premises Controller, users must enter the account name in the **Account** field when logging in to the Controller UI.

1. Navigate to the Administration Console.
2. Locate the `multitenant.controller` setting.
3. Set the value to `true`.

Create Accounts in Multi-Tenant Mode

In multi-tenant mode, you can add accounts as follows:

1. Log in to the AppDynamics Administration Console as the AppDynamics root user.
2. Click **Account Settings** and then **Add**.
3. Define the licensing entitlements that apply to the account.
   Account-level license unit limits let you prevent a particular account from using more licensing units than it should. You can view the total license units available through **Settings** ⚙ > **Admin** > **License**. See License Management in the Controller.

   > ⓘ The overall license limits applicable at the Controller level are independent of any specific limits you apply at the account level.

   Agent-based Licensing: For example, if an account is set up with a Java Agent limit of 100, you can ensure that the new account never interferes with the license availability of another account by setting the **Java Units Provisioned** value for the account to a much smaller limit. However, if you set it to 100 and other accounts are also set to that amount, the first 100 agents that connect to the Controller would occupy those units, regardless of the accounts they report in to. Similarly, you can limit the lifespan of the account by setting an expiration date for the license.

   Infrastructure-based Licensing: For example, if an account is set up with an Infrastructure Monitoring limit of 100, you can ensure that the new account never interferes with the license availability of another account by setting the **Infrastructure Monitoring** value for the account to a much smaller limit. However, if you set it to 100 and other accounts are also set to that amount, the first servers with CPU cores totalling up to 100 would occupy those units, regardless of the accounts they report in to. Similarly, you can limit the lifespan of the account by setting an expiration date for the license.
4. When finished defining entitlements, click **Save** 💾.

After enabling multi-tenant mode, users must specify the account they want to log into in the **Account** field in the Controller UI login screen. See:

- Java Agent Configuration Properties
- .NET Agent Configuration Properties
- Database Agent Configuration Properties
- Machine Agent Configuration Properties

Administer the Reporting Service

Related pages:

- Reports
- Port Settings

The Reporting Service is a standalone Controller process responsible for generating and transmitting reports. The Controller uses the Reporting Service to send both one-time reports and scheduled reports. For more information about the Reporting Service, see Fonts Needed for the Reporting Service and Installation Settings.

Configure the Service

You can configure the Reporting service with the files in the following directory: `<Controller home>/reporting_service/reports/config`. Configure Reporting Service behavior in the `user-config.json` file. Any configuration changes made in user-config.json override default behavior specified in default-config.json. You can configure properties such as the load timeout.

You can configure the Reporting Service with files in the following directory:

`<Controller home>/reporting_service/reports/config`

You can configure Reporting Service behavior in the *user-config.json* file. Any configuration changes made in `user-config.json` override default behavior specified in `default-config.json`.

Disabling HTTP or HTTPS Port

Some on-premise installations can disable the http or https connection. This is done using the same `reportServer:port` config value used to set the listening port. The `default-config.json` installation has the following values for port(s):

```
"reportServer": {
        "port": "8020",
        "portSecure": "8021",
     }
```

To disable https for a localhost system change the "portSecure" to "0":

```
"reportServer": {
        "port": "8020",
        "portSecure": "0",
     }
```

Alternatively, for security reasons if you want to force https, and http you can change the "port" to "0":

```
 "reportServer": {
        "port": "0",
        "portSecure": "8021",
     }
```

After making the change, stop and start the Reports Server as follows:

Windows:

```
cd <installroot>\controller\reporting_service\reports\bin
  reports-service.sh stop
  reports-service.sh start
```

Linux/Mac:

```
cd <installroot>/controller/reporting_service/reports/bin
  ./reports-service.sh stop
  ./reports-service.sh start
  ./reports-service.sh list
```

The reporting-server.log contains info on the port settings during startup.

Limit Reports Service Port Listening to Localhost

Many on-premise installations may want to limit port listening for the Reports Service node server and just listen for localhost connects. These requests do not go onto the network. This is done using the `reportServer:portHostname` and `reportServer:portSecureHostname` value used to set the `listen` hostname parameter. The `default-config.json` installation has these values for the port hostname values, shown below with their port(s):

Default values are:

```
"reportServer": {
      "port": "8020",
      "portHostname" : "",
      "portSecure": "8021",
      "portSecureHostname" : ""
    },
```

The `reporting-server.log` shows info on the hostname settings during startup.
Adding the port*Hostname fields to "localhost" as shown below in `user-config.json` limits the Report Service from connecting to a controller installed on the same host.

```
"reportServer": {
        "portHostname" : "localhost",
        "portSecureHostname" : "localhost"
    },
```

After making the change, stop and start the Reports Server as follows:

Linux/Mac:

```
cd <installroot>/controller/reporting_service/reports/bin
  ./reports-service.sh stop
  ./reports-service.sh start
  ./reports-service.sh list
```

Windows:

```
cd <installroot>\controller\reporting_service\reports\bin
  reports-service.bat stop
  reports-service.bat start
```

See `default-config.json` for more information about configurable properties.

Start and Stop the Service

You can start or stop the Reporting Service independent of the Controller. Run the following commands from the Controller home directory.

> ⓘ When using the Enterprise Console, starting or stopping the Controller will also start or stop the Reporting Service.

Check to see if the Reporting Service is running with the following command:

```
./reporting_service/reports/bin/reports-service.sh|bat list
```

Start the Reporting Service with the following command:

```
./reporting_service/reports/bin/reports-service.sh|bat start
```

Stop the Reporting Service with the following command:

```
./reporting_service/reports/bin/reports-service.sh|bat stop
```

View Logs

The Reporting Service uses the following logs in the Controller home directory:

- `/logs/reporting-server.log`. Prints if the report email was sent and details of the report object that was requested by the user.
- `/reporting_service/reports/logs/reporting-process.log`. Confirms the reporting service process started and whether or not exceptions occurred. Note that this log file is only used on Linux Controllers.

Troubleshooting the Reporting Service

To begin troubleshooting why a report failed to send, open the `server.log` file and find the runUUID for the report you tried to send. Then search for the log entry for the report.

Resolutions for common Reporting Service issues include the following:

- Verify that the Reporting Service is running.
- Verify the default ports for the service: 8020 for HTTP and 8021 for HTTPS
- Verify that the user account used to start the Reporting Service is the same as the account used to start the Controller.

Controller Audit Log

This audit capability creates an `audit.log` file and is used to monitor user activities and configuration changes in the Controller. Be aware that SaaS customers do not have access to the `audit.log` file as it is held on the AppD Controller server. The information is retrieved through the following actions.

Schedule a Controller Audit Report

You must have account-level permissions to view and configure scheduled reports. Use this report to view changes made to the user information, controller configuration, and application properties.

1. Click **Dashboards & Reports** > **Reports** > **Add Report**.
2. Enter **Report Title** and **Report Subtitle**.
    a. You can label a report CONFIDENTIAL using **Report Subtitle**.
    b. Optionally, select **Show Title Page** to include a title page at the beginning of your report file.
3. Select **Report Type** > **Controller Audit** to define the fields in the **Reports Data** tab.
4. Set the time ranges. You can create and manage custom time range if required.
    a. Note: Custom time range options are available for all the **Report Types**.
5. Select your report file format as PDF, JSON, or CSV.
    a. Optionally, uncheck the **Show Diff** box to remove the **Object Changes** column from your report file.
6. Choose the data to include or exclude from the drop-down list.
    a. Repeat as necessary with the following options:
7. Enter the attribute value.
8. Click **+ Add**.

You can create new, duplicate existing, or modify current reports as well as set an email delivery schedule to a defined list of recipients. You can also choose the **Send Report Now** right-click option for an immediate look at the audit details. Review the Reports documentation for more details on other types of reporting.

The Controller Audit reports on the following attributes:

`Date` and `time` range	`ObjectType`
`UserName`	`ObjectName`
`AccountName`	`ApiKeyId` (if applicable)
`Action`	`ApiKeyName` (if applicable)
`ApplicationName`	

Retrieve Controller Audit Log Report

The Controller Audit Log Report is sent by email according to the addresses added to the configurations page. This report captures the following information:

- User logins and information changes
- Controller configuration changes
- Application properties and object changes such as policies, health rules, and entities listed in the above table.
- Environment properties changes

AppDynamics supports PDF, JSON, and CSV output formats.

Retrieve Controller Audit History via API

You can retrieve Controller audit history through the ControllerAuditHistory API method, which returns the configuration and user activities record in a JSON or CSV file for the time range specified. This information is the same as that found in the file.

Format

```
GET /controller/ ControllerAuditHistory?startTime=<start-time>&endTime=<end-time>&include=<field>:
<value>&exclude=<field>:<value>
```

For example:

```
http://localhost:8080/controller/ControllerAuditHistory?startTime=yyyy-MM-dd&&endTime=yyyy-MM-
dd&include=filterName1:filterValue1&include=filterName1:filterValue1&exclude=filterName1:
filterValue1&exclude=filterName1:filterValue1
```

```
curl --user user1@customer1:welcome "http://demo.appdynamics.com:8090/controller/ControllerAuditHistory?
startTime=2015-12-19T10:50:03.607-0700&endTime=2015-12-19T17:50:03.607-
0700&timeZoneId=America&Francisco&include=userName:user1&include=action:LOGIN&exclude=accountName:
system&exclude=action:OBJECT_UPDATE"

[{"timeStamp":1450569821811,"auditDateTime":"2015-12-20T00:03:41.811+0000","accountName":"customer1","
securityProviderType":"INTERNAL","userName":"user1","action":"LOGIN"},{"timeStamp":1450570234518,"
auditDateTime":"2015-12-20T00:10:34.518+0000","accountName":"customer1","securityProviderType":"INTERNAL","
userName":"user1","action":"LOGIN"},{"timeStamp":1450570273841,"auditDateTime":"2015-12-20T00:11:13.841+0000","
accountName":"customer1","securityProviderType":"INTERNAL","userName":"user1","action":"OBJECT_CREATED","
objectType":"AGENT_CONFIGURATION"},
...
{"timeStamp":1450570675345,"auditDateTime":"2015-12-20T00:17:55.345+0000","accountName":"customer1","
securityProviderType":"INTERNAL","userName":"user1","action":"OBJECT_DELETED","objectType":"
BUSINESS_TRANSACTION"},{"timeStamp":1450570719240,"auditDateTime":"2015-12-20T00:18:39.240+0000","accountName":"
customer1","securityProviderType":"INTERNAL","userName":"user1","action":"APP_CONFIGURATION","objectType":"
APPLICATION","objectName":"ACME Book Store Application"},{"timeStamp":1450571834835,"auditDateTime":"2015-12-
20T00:37:14.835+0000","accountName":"customer1","securityProviderType":"INTERNAL","userName":"user1","action

curl --user user1@customer1:welcome "http://127.0.0.1:8080/controller/ControllerAuditHistory?startTime=2019-05-
28T08:00:03.607-0700&endTime=2019-05-28T11:32:03.607-0700&timeZoneId=America%2FSan%
20Francisco&include=applicationName:ACME"
[{"timeStamp":1559066415823,"auditDateTime":"2019-05-28T18:00:15.823+0000","accountName":"customer1","
securityProviderType":"INTERNAL","userName":"user1","action":"LOGIN","objectId":0,"applicationName":"ACME"}]
```

Input parameters

Parameter Name	Parameter Type	Value	Mandatory
start-time	Query	Start time in the format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Yes
end-time	Query	End time in the format: "yyyy-MM-dd'T'HH:mm:ss.SSSZ"	Yes
time-zone-id	Query	Time zone	No
include	Query	Restricted information in the Controller audit history	No
exclude	Query	Restricted information in the Controller audit history	No

> ⚠ To control the size of the output, the range between the start-time and end-time cannot exceed twenty-four hours. For periods longer than 24 hours, use multiple queries with consecutive time parameters.

- Multiple filters of the same type are allowed.
- The backend API treats include filters with the same <field> and relationship as "OR", and filters with different <field> and relationship as "AND".
- There is no direct interaction between include and exclude filters.

- Each filter needs to be a parameter, e.g., `include=filterName1:filterValue1&include=filterName2:filterValue2`. See the below examples.

Log File Information by Platform

What is Audited

The following entries are audited:

```
ACCOUNT
ACCOUNT_ROLE
ACTION_SUPPRESSION_WINDOW
AGENT_CONFIGURATION
ANALYTICS_DYNAMIC_SERVICE_HIERARCHICAL_CONFIGURATION
APPLICATION
APPLICATION_COMPONENT
APPLICATION_COMPONENT_NODE
APPLICATION_CONFIGURATION
APPLICATION_DIAGNOSTIC_DATA
ASYNC_TRANSACTION_CONFIG
BACKEND_DISCOVERY_CONFIG
BUSINESS_TRANSACTION
BUSINESS_TRANSACTION_CONFIG
BUSINESS_TRANSACTION_GROUP
CALL_GRAPH_CONFIGURATION
CUSTOM_ACTION
CUSTOM_CACHE_CONFIGURATION
CUSTOM_EMAIL_ACTION_PLAN_CONFIG
CUSTOM_EXIT_POINT_DEFINITION
CUSTOM_MATCH_POINT_DEFINITION
DASHBOARD
DIAGNOSTIC_SESSION_ACTION
DOT_NET_ERROR_CONFIGURATION
EMAIL_ACTION
ERROR_CONFIGURATION
EUM_CONFIGURATION
EVENT_REACTOR
GLOBAL_CONFIGURATION
GROUP
HTTP_REQUEST_ACTION
HTTP_REQUEST_ACTION_MEDIA_TYPE_CONFIG
HTTP_REQUEST_ACTION_PLAN_CONFIG
HTTP_REQUEST_DATA_GATHERER_CONFIG
INFO_POINT
JIRA_ACTION
JMX_CONFIG
MEMORY_CONFIGURATION
METRIC_BASELINE
MOBILE_APPLICATION
NODEJS_ERROR_CONFIGURATION
NOTIFICATION_CONFIG
OBJECT_INSTANCE_TRACKING
PHP_ERROR_CONFIGURATION
POJO_DATA_GATHERER_CONFIG
POLICY
PYTHON_ERROR_CONFIGURATION
RULE
RUN_LOCAL_SCRIPT_ACTION
SCHEDULED_REPORT
SERVICE_ENDPOINT_DEFINITION
SERVICE_ENDPOINT_MATCH_CONFIG
SMS_ACTION
SQL_DATA_GATHERER_CONFIG
THREAD_DUMP_ACTION
TRANSACTION_MATCH_POINT_CONFIG
USER
WORKFLOW
WORKFLOW_ACTION
```

The Audit report now supports Application Name for the above entities when applicable.

Supported Audit Actions

Below is the list of actions supported in auditing.

Note that not all of these actions are supported for all of the Audit Entries in the table above.

- `ACCOUNT_REENABLED`
- `ACCOUNT_ROLE_ADD_PERMISSION`
- `ACCOUNT_ROLE_REMOVE_PERMISSION`
- `ACKNOWLEDGE_GDPR_DATA_PRIVACY`
- `ANOMALY_DETECTION_CONFIG_CHANGED`
- `FLOW_ICON_MOVED`
- `GROUP_ADD_ACCOUNT_ROLE`
- `GROUP_REMOVE_ACCOUNT_ROLE`
- `LDAP_CONFIG_CREATED`
- `LDAP_CONFIG_DELETED`
- `LDAP_CONFIG_UPDATED`
- `LOG_LEVEL_CHANGED`
- `LOGIN`
- `LOGIN_FAILED`
- `LOGOUT`
- `LOGOUT_FAILED`
- `OBJECT_CREATED`
- `OBJECT_DELETED`
- `OBJECT_UPDATED`
- `SAML_AUTHENTICATION_CONFIG_CREATED`
- `SAML_AUTHENTICATION_CONFIG_DELETED`
- `SAML_AUTHENTICATION_CONFIG_UPDATED`
- `USER_ADD_ACCOUNT_ROLE`
- `USER_ADD_TO_GROUP`
- `USER_EMAIL_CHANGED`
- `USER_PASSWORD_CHANGED`
- `USER_PASSWORD_RESET`
- `USER_PASSWORD_RESET_COMPLETED`
- `USER_REMOVE_ACCOUNT_ROLE`
- `USER_REMOVE_FROM_GROUP`

Troubleshoot Controller Issues

This page provides troubleshooting information for issues that may arise during Controller installation and operation.

Controller Server Log

The primary log file for the Controller at the following location:

```
<controller_home>/logs/server.log
```

The first step in troubleshooting Controller issues typically involves checking the log file. Search the log for errors that may correspond to the issue you are encountering. If found, an error log may help you identify and resolve the issue.

Also, see installation troubleshooting information in Custom Install.

Identify Controller Performance Issues

The following are indications of Controller performance issues:

1. The Controller UI performs slowly. For short time ranges, such as 15 or 30 minutes, responses that take longer than 10 to 20 seconds can indicate that your Controller is under stress.
2. When the Controller's metric reporting lags 7 to 10 minutes behind the current time, it can be an indication that your Controller is under stress. A lag of about 3 to 5 minutes is normal.
3. When monitoring the Controller environment, you see that CPU, memory, and disk metrics are at about 75% capacity.

If you observe degradation in Controller performance, it may be due to one of the following:

- The hardware resources for the Controller might not match the correct Controller profile.
- The Controller performance profile may be incorrectly configured.

To troubleshoot Controller performance issues:

1. Confirm that the hardware matches the Controller profile you use. For details see Controller System Requirements.
2. Confirm that your disk performance matches the recommended thresholds for minimum disk performance. For details see Controller System Requirements.
3. Confirm that the Java SDK version is exactly the same as the Java version on the Controller. To display the version of Java used by the Controller:
   - Open the command-line utility.
   - Go to `<Controller_Installation_Directory>/jre/bin`
   - Run `java -version`.

Monitor heap usage

- On Windows, use the Task Manager to measure the memory usage for the Controller.
- On Linux, use the **top** command to get statistics for the memory data.

```
ps -elf (expect to see a "java" process and a "mysql" process)

top (expect to see java and mysql with cpu greater then 0)
```

Timeout errors during Controller installation

While installing the Controller, the Enterprise Console attempts to start up the Controller application server and database. At first database startup, the application attempts to create the database schema, tables, and other artifacts needed by the Controller.

By default, the Enterprise Console waits 45 minutes for the Controller app server or database to start. When installing a medium or large profile Controller or into certain types of environments such as virtual machines, the time it takes to start up the system can exceed the default startup timeout period.

Controller does not start properly on Windows

Your Controller may not be starting due to file extensions of transaction logs created by Glassfish. Excluding the Controller data directory from being scanned by virus scanners as specified on Prepare Windows for the Controller does not account for these extent files found in the `<AppDynamicsInstall>\Controller\appserver\glassfish\domains\domain1\logs\server\tx` directory. When your antivirus detects these extensions, such as WRY, it may mistakenly stop the process of using these files so the Controller ultimately does not start.

These transaction logs are used to recover any failed Glassfish transactions, so deleting these logs on startup is not advised. Instead, configure your virus scanners to ignore the entire Controller directory.

No data in the Metrics Browser

This may indicate that the agents are not correctly configured. Begin troubleshooting by looking at the `server.log` file.

All log files for Controller are located in the `<Controller_Installation_Directory>/logs` folder.

Error Message	Solution
Error receiving metrics (node not properly modeled yet: Could not find component for node.	This error means the app agent tried to upload metric data for a specific node, but the node does not belong to any tier. Nodes must belong to tiers and these tiers must belong to a business application in order to receive metric data for that node. See Overview of Application Monitoring.
Received Metric Registration request for a machine that is NOT registered to any nodes. Sending back null!	This error indicates that the Controller received a registration request for metrics for a Machine Agent that listed a machine ID not yet associated with any node. Configure the Machine Agent to associate with the correct application, tier, and node. See Install the Machine Agent.
Agent upload blocked, as its reporting a time well into the future.	The App Agents attempt to report metric data using Controller time. The agents retrieve the time from the Controller every five minutes and report times using a skew of the local machine time, if different. If for some reason the App Agent reports metrics that are time-stamped ahead of the Controller time, the Controller rejects the metrics. To avoid this event, ensure that the system times for the machine on which the Controller is running and the machines for the app agents are in synchronization.

Controller shutdown does not increase free memory on Linux

You do not generally need to be concerned about the "free memory" value, as it will always trend towards zero. The Linux kernel tries to keep its cache as large as possible. As a result, the Linux kernel does not release the memory even after process termination. The memory is freed only if it is required by another process.

Controller process unexpectedly shut down

On Linux, memory allocation failures may cause the Controller process to be shut down unexpectedly by the Linux Out-of-Memory (OOM) Killer. The Controller log, `server.log`, does not provide information about the shutdown. Instead, to diagnose this event, check the system log (usually `/var/log/messages`) for "out of memory" entries written by the OOM killer, for example, as follows:

```
grep -i "Out of memory" /var/log/messages
```

If you encounter this log entry, make sure that you have allocated sufficient swap space on the Controller machine. AppDynamics recommends allocating a minimum of 10 GB of swap space.

Controller server swapping too often

If you encounter unexpected swapping on the Controller machine, you can configure how aggressive the operating system swaps by configuring the `swappiness` parameter. The `swappiness` parameter controls how often the Linux kernel moves processes out of physical memory and onto the swap disk.   The default value for the parameter is usually 60. When you decrease the value, you lower the tendency of the operating system to swap. This results in less default file caching.

See the documentation for your Linux distribution for recommendations on the value for the `swappiness` parameter. For example, RedHat recommends setting swappiness to 10 for CentOS and RedHat kernels version 2.6.32-303 or later if you encounter OOM issues even though swap space is still available.

Before you configure the `swappiness` parameter though, ensure that the machine has sufficient RAM and that the buffer pool size for MySQL is properly configured.

To configure swappiness

1. Check the current value for `swappiness`.

   ```
   /sbin/sysctl -a | grep swappiness
   ```

2. Set the `swappiness` parameter.

   For example, add the following line to set the `swappiness` parameter to 10.

   ```
   echo 10 > /proc/sys/vm/swappiness
   ```

3. Set the `swappiness` parameter in the `/etc/sysctl.conf` file to the same value you used in step 2.

   For example, add the following line to the `/etc/sysct1.conf` file:

   ```
   vm.swappiness = 10
   ```

Could not determine the IP address of this host error during installation

During the installation process, the Enterprise Console attempts to ping the Controller by the hostname or IP address you enter. If the ping is unsuccessful during the user input validation, the following error message appears: "Could not determine the IP address of this host. Please ensure that the IP address of the Controller host resolves to its hostname or to localhost. You may need to add an entry in the hosts file on the Controller host and retry the operation."

To make the hostname resolvable, add an entry for it to the hosts file on the machine on which you are installing the Controller. On Linux, the hosts file is typically at `/etc/hosts`. On Windows, look for the file at the following location, `C:\Windows\System32\Drivers\etc\hosts`, or the location appropriate for your version of Windows.

Add the entry in the form of the following example:

```
127.0.0.1 localhost myhostname
```

Use the IP address and hostnames appropriate for your system.

For example, the following shows the entry added as the third line of the default RedHat hosts file:

```
127.0.0.1    localhost.localdomain localhost
::1          localhost6.localdomain6 localhost6
198.51.100.2 myhost myhost.example.org
```

Controller Cannot Connect to the MySQL Database

The following exception message in server.log file indicates that the Controller cannot connect to its embedded database.

```
*Server log exception:* "Caused by: java.net.ConnectException: Connection refused"
```

If you encounter this error, verify that the Controller database is running properly. On Linux, you can do so using one of the following commands:

Linux	Windows	Description
`lsof -i:3388`	SysInternals Process Explorer, will provide a list of files opened by process with pid 3388.	List open files opened by process with pid 3388.
`netstat -anp | grep 3388`	`netstat -ano | find "3388"`	List all networking ports opened by process with pid 3388.

`ps -aef | grep mysql`	`tasklist /v | find "mysql"`	Lists all processes and then checks if the process with name "mysql" is active and alive.

If no processes are found, it indicates that the Controller database was incorrectly terminated. Start the Controller database again and verify the Controller `server.log` file for any error messages.

Stack overflow exception when installing the Controller installation on Windows

This exception is usually caused when you set the `-Xss` option to a lower value. We recommend changing this value to 96000.

Triggering automatic collection of Controller logs

Use the following console commands to trigger automatic capture of Controller log files:

- On Linux, run:

  ```
  bin/platform-admin.sh submit-job --platform-name test --service controller --job retrieve-log
  ```

- On Windows, open an elevated command prompt (in the Windows start menu, right-click the **Command Prompt** icon and choose **Run as Administrator**) and run:

  ```
  bin/platform-admin.exe cli submit-job --platform-name test --service controller --job retrieve-log
  ```

  The logs will be copied in the Enterprise Console host under `platform-admin/logs-controller-<platform-name>-<date-time-stamp>.zip`.

See Platform Log Files to learn how to manage your Controller logs.

Collecting Troubleshooting Information for the Controller

If opening a support case for Controller troubleshooting, you can facilitate the diagnosis of the problem by providing the following information:

- Submit all `platform-admin/logs/*` and `platform-admin/logs-controller-*.zip`, in particular the `server.log` files. You can also use the log file utility described in Triggering automatic collection of Controller logs to collect logs.
- If the Controller runs out of memory, it generates a heap dump. Submit all files in `<controller_home>/appserver/glassfish/domains/domain1/config/hprof`.
- Submit all `<controller_home>/appserver/glassfish/domains/domain1/config/gc.log` files.
- Submit information about the hardware and operating system configuration of the machine that is currently hosting the Controller, including operating system, bit version, CPU cores, clock speed, disk configuration, and RAM.
- Indicate the Performance profile of Controller. Run the controller diagnosis command which captures the information in `platform-admin-server.log`:

  Refer to the Controller diagnostic data in the `platform-admin-server.log`. See a sample Controller diagnostic data on Manage a High Availability Deployment page.

Issues Generating Audit Reports Immediately after Upgrading the Controller to 4.5

When the Controller upgrade is complete, audit reports may not work immediately. The audit database table is getting migrated only after the upgrade process and the migration takes at least an hour to complete. If audit reports are run before completing the migration process, audit table migration messages are logged in the `server.log` file.

No actions are required, try running the audit reports again after an hour.

Controller Dump Files

The following steps describe how to collect troubleshooting information for your Controller. You may be requested for the information when troubleshooting with the AppDynamics support team.

Get Heap and Histogram Dump Files

It is recommended that you install JDK on your system before using the following commands.

- Get the **process id of the Controller** to use in subsequent commands.

```
ps -ef | grep java
```

- Get the **heap dump before garbage collection** using the following command:

```
<java-jdk-install-dir>/bin/jmap -dump:format=b,file=heap_before_live.bin <Controller_pid>
```

- Get the **histogram before garbage collection** using the following command:

```
<java-jdk-install-dir>/bin/jmap -histo <Controller_pid> | head -200 > histo_before_live.txt
```

- Get the **histogram after garbage collection** using the following command:

```
<java-jdk-install-dir>/bin/jmap -histo:live <Controller_pid> | head -200 > histo_after_live.txt
```

Take Four Thread Dumps at Three Second Intervals

- Using the Controller process ID, execute the following command:

```
kill -3 <Controller_pid>
```

- Save the `<Controller_Installation_Directory>/appserver/glassfish/domains/domain1/logs/jvm.log` file.

Send the Files to the AppDynamics Support Team

If asked to provide the information to the AppDynamics support team, send the following files generated by these steps:

- `heap_before_live.bin`
- `histo_before_live.txt`
- `histo_after_live.txt`
- `jvm.log`

Controller Component Versions

This page describes how to check version information and the version of bundled components. This information is useful when troubleshooting the system or performing other administrative tasks.

> ⚠ AppDynamics maintains and updates the bundled components as part of the AppDynamics platform. Do not attempt to upgrade a bundled component independently of the platform upgrade procedure.

Controller Version

You can retrieve the Controller version in two ways:

1. From the AppDynamics UI:
   1. Click the **Settings** ⚙.
   2. Select **About AppDynamics.**
   3. Note the **Controller build** number.
2. From the command line of the Controller machine:
   1. Access the `README.txt` file located in the Controller home directory.

Bundled Glassfish Server Version

The Glassfish server is installed in `<controller_home>/appserver`.

The Glassfish server version is Glassfish 4.1.1.

Bundled MySQL Database Version

The AppDynamics Controller uses MySQL as its default database, where it stores configuration data, metrics data, transaction snapshot data and events, and the history of incidents that occurred (both resolved and unresolved incidents are stored). The MySQL database files are installed in `<controller_home>/db` by default.

The latest AppDynamics release bundles MySQL version 5.7.34.

Check MySQL Version

To check what your MySQL version is in your Controller, you can run the following command:

```
<controller_home>/bin/controller.sh login-db
select version();
```

Upgrade MySQL Version

Optionally, after you install or upgrade the Controller, you can upgrade the MySQL version with the Enterprise Console. Note that you cannot reverse this process.

> ⓘ A newly installed 4.5 Controller packages and uses MySQL 5.7. However, a Controller that is upgraded to 4.5 from a previous version where MySQL 5.5 is used, will also use version 5.5.

You can upgrade the MySQL version on the Controller page in the GUI or with the following command:

Bundled Java Version

The Controller bundles and uses Java Runtime Environment 8.54.0.22.

Enterprise Console Version

Run the command in the `<Enterprise Console installation directory>/platform-admin` to check the version:

Other Component Versions

See Legal Notices for the latest components included in the AppDynamics products and modules.

Upgrade the Controller Using the Enterprise Console

Related pages:

- Controller Data Backup and Restore
- Upgrade a Single Controller
- Upgrade an HA Pair

You can upgrade a Controller instance using the Enterprise Console. The Enterprise Console simplifies the upgrade process by allowing you to discover and upgrade single Controllers and HA-pairs.

About the Upgrade

- The Enterprise Console supports standalone and HA-pair Controller upgrades. Use the following table to determine your course of action based on your circumstances:

If your current Controller version is...	Controller version you want to upgrade to...	Actions to take...
>= 21.x	Controller version 21.x or the latest version	1. Access the downloads portal to download the Controller version. 2. Use the Controller installer and upgrade the Controller.

- You can choose which version you would like to upgrade the Controller to as long as the Enterprise Console is aware of that version. This means that you can upgrade the Controller to any intermediate version or to the latest version as long as the Enterprise Console installer has been run for those versions. However, you cannot upgrade the Controller to an older version.
- If you have a license for the older version, the license should work when upgrading the Controller to a new version. However, if you have a temporary license for the old version and now have a new license, the new license will not work on the old Controller. In this case, you should upgrade the Controller to the latest version before applying the license.
- An upgrade results in Controller downtime, but it is not necessary to stop agents during the Controller upgrade.
- The Enterprise Console expects a `.passwordfile` file to be present in the Controller home directory. The Enterprise Console reads this password and validates it against the Controller. Once the upgrade is complete, the Enterprise Console removes the file, and stores the password in its encrypted database.

> ⚠ If you change the Glassfish Admin Password manually, you also need to update it in the Enterprise Console Controller Settings.

Before Upgrading

- Before you start upgrading the Controller, make sure that you are using the correct update order.
- Review the latest Release Notes and the release notes for any intermediate versions between the current version of your instance and the version you are targeting to learn about issues and enhancements in those releases.
- Check the most recent Controller System Requirements and Troubleshoot Controller Issues to review your Controller's current workload and determine whether you need to change your performance profile and increase your hardware resources, if necessary.

> ⓘ You may change your Controller profile on the Platform Configurations pages of the Enterprise Console GUI, either before or after you upgrade your Controller. This process is not reversible, and you cannot move from a larger to a smaller profile size.

- Check the Controller's database.log for any errors. You can find the log at `<controller_home>/db/logs/database.log`. There should not be any `InnoDB: Error` lines in the log. If any errors are found, please reach out to AppDynamics Support before attempting the upgrade. Upgrading the Controller with a corrupt database may put the Controller in a bad state, with high recovery time.
- If you changed any Glassfish settings that are not JVM options, note your changes. You may need to configure them after an upgrade. The Enterprise Console recognizes and retains many common customizations to the domain.xml, db.cnf, and other configuration files, but is not guaranteed to retain them all. If you have made manual configuration changes to the files, verify the configuration after updating. See Retaining Configuration Changes to learn how to preserve changes.
- If you uninstall the Enterprise Console that was previously managing the Controller and use a new Enterprise Console instance to discover and upgrade the Controller, you need to first manually create the `.passwordfile` file in order for the Enterprise Console to continue with the discover and upgrade process. You can create the file in the Controller home directory, and add the `AS_ADMIN_PASSWORD=<controllerRootUserPassword>` value in it.

- When performing an upgrade while enabling HTTPS, check that the hostname does not start with a digit. The upgrade will fail if the Enterprise Console hostname starts with a digit due to a JDK limitation with the DNS name in CN/SAN.
- Back up the existing Controller following the steps in the next section.

Back up the Existing Controller

The Enterprise Console retains agent data, reports, configuration, and other types of data through an upgrade, including a copy of commonly modified configuration files under `<controller_home>/backup`. Nevertheless, to mitigate the risk of data loss in the event of an unexpected failure, be sure to back up the existing installation directory before applying an upgrade to the Controller.

> ⓘ The Enterprise Console does not back up MySQL data. You need to back up the data before upgrading standalone installations.

If the upgrade does not finish successfully for any reason, see Troubleshooting the Upgrade for more information.

To back up the Controller instance

1. Stop the Controller application server and database.
    - For Linux, run:

      ```
      platform-admin.sh stop-controller-appserver
      ```

    - For Windows, run:

      ```
      platform-admin.exe cli stop-controller-appserver --with-db
      ```

2. Back up the Controller home by copying the entire Controller home directory to a backup location. Note the following points:
    - If the data home for the Controller is not under the Controller directory, be sure to back up the database directory as well.
    - If it's not possible to back up the entire data set, you can selectively back up the most important tables. Use the Metadata Backup SQL script described and attached to the Controller Data Backup and Restore page.
3. Restart your Controller after completing the backup and before upgrading.

After Upgrading

- If you have configured settings in the `domain.xml` file, `db.cnf` or other configuration files manually or by using the Glassfish `asadmin` utility, verify those changes in the configuration files or Controller Configurations page of the GUI after the upgrade and re-apply any customizations that were not preserved. The Enterprise Console preserves several recommended customizations. After the upgrade, you can find backup copies of common configuration files in the `<controller_home>/backup` directory.
- As an optional step, your MySQL version can be upgraded after you upgrade the Controller, through the Enterprise Console GUI or by using the `mysql-upgrade` job. See Bundled MySQL Database Version for more information.

Troubleshooting the Upgrade

If the upgrade does not succeed for any reason, the Enterprise Console does not roll back changes on disk. This gives you an opportunity to diagnose and troubleshoot the issue before reattempting the installation or upgrade.

To troubleshoot the issue, check the installation log at `platform-admin/logs/platform-admin-server.log`. The Controller `server.log` file may contain additional information.

Resuming from Checkpoint

The Enterprise Console provides a feature to resume the upgrade from the last point of failure (checkpoint). The application creates a checkpoint at several stages during installation. If the installation fails, resuming from checkpoint would skip all the prior successfully completed stages and restart the installation from the beginning of the specific stage where the last point of failure occurred.

You can also resume a failed Controller job from the CLI by passing the flag `useCheckpoint=true` as an argument after `--args` in your command.

> ⓘ If for some reason, the upgrade from the last checkpoint is not successful, you may retry the upgrade from the beginning. Simply click **Retry** instead of Resume from the checkpoint. However, please note that retrying from the beginning after a failed upgrade may overwrite your customizations to `db.cnf` and `domain.xml`.

Timing Out

A common upgrade issue involves the upgrade process timing out. The Enterprise Console attempts to restart the Controller and database after applying an update or installation. For large databases and depending on the system resources, this can take a considerable amount of time. If the Enterprise Console cannot finish starting up the Controller within a set time-out period (30 minutes by default), the operation will fail.

You can increase the default time out period for system startup. The timeouts are defined in `platform-admin/archives/controller/<version>/playbooks/controller.groovy`. You can update the `controllerStartRetryTimeout = 10 * 60 seconds = 10 minutes`, and then retry the upgrade from the checkpoint.

> ⓘ When the Controller upgrade is complete, audit reports may not work immediately. The audit database table is getting migrated only after the upgrade process and the migration takes at least an hour to complete. If audit reports are run before completing the migration process, audit table migration messages are logged in the server.log file. No actions are required, try running the audit reports again after an hour.

Upgrade a Single Controller

Related pages:

- Controller Data Backup and Restore
- Upgrade the Controller Using the Enterprise Console

You can use the Enterprise Console to onboard and upgrade a single node Controller instance. The **Custom Install Discover & Upgrade** option in the GUI allows you to create a platform and discover a Controller.

Alternatively, if you have already created a platform, you must add credentials and hosts to the platform before you can perform discovery. Then, discover the Controller on the page. Discovering a Controller means that the Enterprise Console learns about your existing Controller deployment, such as profile, tenancy mode, existing domain configuration, and database configuration. This information is used to perform an upgrade.

About the Upgrade

The Enterprise Console supports Controller upgrades (standalone and HA-pair) starting from 20.2 and higher, to the latest version. Use the following table to determine your course of action based on your circumstances:

If your current Controller version is...	Controller version you want to upgrade to...	Actions to take...
Equal to version 20.2 or later	Controller version 20.2 or the latest version	1. Access the downloads portal to download the Controller version. 2. Use the Enterprise Console to upgrade from your existing Controller version to the Controller version you want.

Upgrade the Controller Using GUI

If there is a Controller upgrade available, you can begin the upgrade process either on the Custom Install or Controller page in the GUI.

ⓘ Ensure that the Controller and database are running prior to the upgrade. The Enterprise Console validates the database root password and Controller root passwords provided during the upgrade.

Upgrade the Controller from 20.x to Latest

To upgrade the Controller from 20.x to the latest version, you can use the Upgrade Controller feature:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Upgrade the Enterprise Console to the latest version.
3. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
4. Navigate to the **Controller** page of the platform.
5. Select the Controller host you would like to upgrade.
6. Select **Upgrade Controller**.
7. Select an available Target Version from the dropdown.

   ⓘ The list is populated by versions that the Enterprise Console is aware of. Of those versions, the list will only show versions that are the same or greater than the current Controller version.

8. Enter the required passwords and select **Submit**.

Upgrade the Controller Using CLI

If there is a Controller upgrade available, you can begin the upgrade process using the application CLI.

> Ensure that the Controller and database are running prior to the upgrade. The Enterprise Console validates the database root password and Controller root passwords provided during the upgrade.

Upgrade the Controller from 20.x to Latest

Upgrades from 20.x to the latest version can be performed on the Controller page of the Enterprise Console or with the following commands:

1. Upgrade the Enterprise Console to the latest version.
2. Navigate to the `<Enterprise Console home directory>/platform-admin directory`.
3. If it has been more than one day since your last session, you will have to log in with the following command:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

4. Apply the upgrade to the Controller with the following command:

   If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`

How to Reset the Database User Password

You can customize the database user password. However, if you are upgrading your system, you may have forgotten the password. How you reset the database user password depends on whether the Enterprise Console has discovered the Controller.

If the Enterprise Console has discovered the Controller

Use the Enterprise Console CLI commands to:

1. Log in to the Enterprise Console.

   ```
   platform-admin/bin/platform-admin.sh login --user-name admin --password EC_GUI_PASSWORD
   ```

   Replace `EC_GUI_PASSWORD` with your actual value.
2. Reset the database user password.

   ```
   platform-admin/bin/platform-admin.sh submit-job --platform-name <platform_name> --service controller --
   job update-passwords --args newDatabaseUserPassword=<password>
   ```

   Replace `<platform_name>` and `<password>` with your actual values.

As a result, an Enterprise Console job is generated where you can verify the success of the password reset.

If the Enterprise Console has NOT discovered the Controller

> Downtime is required to change the Controller Database user password. If you have installed a Controller HA pair, you must disable auto-failover to avoid an accidental failover while changing the password. For more details, refer to the Automatic Failover section.

1. Log in to the database as the root user by running the following command:

   ```
   ./mysql --user=root -p --host=127.0.0.1 --port=3388 --protocol=TCP
   ```

2. Execute the following queries, replacing `<new_password_here>` before executing the query.

   ```
   update mysql.user set authentication_string=password('<new_password_here>') where user like
   'controller%'; flush privileges; quit;
   ```

3. Verify the login by running the following command in the `<controller_home>/db/bin` directory:

```
./mysql -u controller -p -P 3388 -h 127.0.0.1
```

4. Update the password alias by using the below command in the `<controller_home>/appserver/glassfish/bin` directory:

   ```
   ./asadmin update-password-alias controller-db-password
   ```

   Enter the user name as admin, the admin password as Controller root user password, and the alias password as the Controller Database user password.

   > ⚠ In the case of a Controller HA pair, follow steps 1-3 on the secondary controller server. Then, copy the file `domain-passwords` found at `<controller_home>/appserver/glassfish/domains/domain1/config` from the primary to the secondary controller server.

5. Restart the Appserver.

Upgrade an HA Pair

Related pages:

- Upgrade the Controller Using the Enterprise Console
- Controller High Availability
- Controller Data Backup and Restore

You can use the Enterprise Console to onboard and upgrade an HA Controller pair. For HA pairs that are not managed by the Enterprise Console, use the discover and upgrade option; for HA pairs that are managed by the Enterprise Console, you must use the upgrade option.

This topic describes:

- Upgrade method benefits
- Available upgrade methods
- How to shut down the HA Toolkit (HATK) implementation

Upgrade Method Benefits

The upgrade method enables you to:

- Perform a set of pre-upgrade validations. A summary of validation errors is provided to you before you modify the system's state.
- Quickly restore the older Controller version from the preserved secondary version in case of any upgrade issues. As you upgrade the primary server, the secondary server is isolated, thereby providing you a backup from which to quickly restore the service.

Available Upgrade Methods

Use the CLI, or follow a step-by-step Upgrade wizard that guides you through the UI, to perform the upgrade.

Use the CLI Method to Upgrade

You can use the CLI to upgrade the HA Controller pair. For more details, select upgrade using the CLI.

Use the Upgrade Wizard Method to Upgrade

You can use the Upgrade Wizard to follow a step-by-step procedure that guides you through the UI to perform the upgrade. For more details, select upgrade using the Upgrade Wizard.

Shut Down the HA Toolkit (HATK) Implementation

Before you start the Controller HA upgrade from a pre-Enterprise Console deployment, you must first shut down the watchdog and assassin processes. This is only required if you have installed the HA Toolkit previously.

To shut down the HA Toolkit implementation:

- If the Controller services are installed with privilege escalation using setups option (-c option in install-init.sh) then running the following command on the secondary will stop the secondary appserver, watchdog, and assassin.

  ```
  /sbin/appdservice appdcontroller stop
  ```

  or
- If the Controller services are installed with privilege escalation using sudoers option (`-s` option in `install-init.sh`) then running the following command on the secondary will stop the secondary appserver, watchdog, and assassin.

  ```
  sudo /sbin/service appdcontroller stop
  ```

To check if the appserver, watchdog, and assassin stopped on the secondary, you can use the following commands based on the privilege escalation method used to install the Controller services:

```
/sbin/appdservice appdcontroller status
```

or

```
sudo /sbin/service appdcontroller status
```

For both stopping and checking the statuses, if you do not remember the privilege escalation method used to install services, then you can use both variants, one after the other, in any order.

Upgrade the HA Controller Pair Using the CLI

If are using the HA ToolKit (HATK) and made any customizations to it, AppDynamics recommends that you review your particular situation and determine if you should proceed with the migration. For more details, see HA module in the Enterprise Console.

Steps 1 through 3 consist of different procedures depending on your HA Controller pair deployment:

- Follow **Option 1 - Discover and Upgrade** for deployments that are not managed by the Enterprise Console.
  Use the discover and upgrade job to onboard your HA Controller pair to the Enterprise Console before upgrading the primary Controller.
- Follow **Option 2 - Upgrade** for HA pair deployments that are managed by the Enterprise Console.
  Use the upgrade option to upgrade your primary Controller managed by the current Enterprise Console instance.

To upgrade using the CLI:

- Step 1: Prepare the Upgrade
- Step 2: Perform a Failover and Check the HA Pair State
- Step 3: Upgrade the Primary Controller
- Step 4: Verify the Primary Upgrade
- Step 5: Upgrade the Secondary Controller
- Step 6: Verify the Secondary Upgrade

Step 1: Prepare the Upgrade

Step 2: Perform a Failover and Check the HA Pair State

To begin the upgrade, perform a failover to the secondary Controller. Since the primary Controller is known to be working, this provides a stable configuration to fail back to in case of upgrade issues.

To perform a failover and check the HA pair state:

Step 3: Upgrade the Primary Controller

You can upgrade the primary Controller using one of the following commands:

See Troubleshooting the Upgrade if the primary upgrade fails.

Step 4: Verify the Primary Upgrade

A primary upgrade success message displays if the upgrade is successful. However, AppDynamics recommends that you perform your own in-house verification on the Controller before proceeding to the secondary upgrade. For example, you can check that your agents are continuing to report to your Controller.

The following describes the expected state in the Enterprise Console.

Run the following commands on the Enterprise Console host:

```
bin/platform-admin.sh list-platform-service --platform-name <name_of_the_platform>
```

The desired output is `Controller: primary_upgraded`.

```
bin/platform-admin.sh list-node --service controller --platform-name <name_of_the_platform>
```

The output will display the host versions.

Sample Output

```
Available nodes in the controller service:
        Controller: role Primary, host 172.12.0.1, version 4.5.5.0, state running
         MySQL: role Primary, host 172.12.0.1, version 5.7.24 state running
        Controller: role Secondary, host 172.12.0.2, version 4.5.0.2, state stopped
         MySQL: role Secondary, host 172.12.0.2, version 5.7.21, state running
```

Check that all hosts show the new versions, that the Controller is running on the primary and stopped on the secondary, and that MySQL is running on both hosts. If you are not satisfied with the upgrade, see Verify the Primary Upgrade is Unsatisfactory.

Step 5: Upgrade the Secondary Controller

To upgrade the secondary Controller, run the the `upgrade-secondary` command on the Enterprise Console host:

```
bin/platform-admin.sh submit-job --service controller --job upgrade-secondary --platform-name
<name_of_the_platform> --args controllerRootUserPassword=<controller_root_password>
mysqlRootPassword=<mysql_root_password>
```

If the secondary Controller upgrade fails, see Upgrade the Secondary Controller Fails for possible recovery options.

Step 6: Verify the Secondary Upgrade

A secondary upgrade success message displays if the upgrade is successful. However, AppDynamics recommends that you perform your own in-house verification on the Controller before completing the upgrade.

The following describes the expected state in the Enterprise Console.

Run the following commands on the Enterprise Console host:

```
bin/platform-admin.sh list-platform-service --platform-name <name_of_the_platform>
```

The desired output is "Controller: provisioned".

```
bin/platform-admin.sh list-node --service controller --platform-name <name_of_the_platform>
```

The output will display the host versions.

Sample Output

```
Available nodes in the controller service:
        Controller: role Primary, host 172.12.0.1, version 4.5.5.0, state running
         MySQL: role Primary, host 172.12.0.1, version 5.7.24 state running
        Controller: role Secondary, host 172.12.0.2, version 4.5.5.0, state stopped
         MySQL: role Secondary, host 172.12.0.2, version 5.7.24, state running
```

Ensure that the secondary host version has been upgraded. Also, the primary should be in a running state, while the secondary Controller appserver should remain stopped. However, its MySQL process should be running.

Troubleshooting the Upgrade

Failover Issues

If you experience failover issues before upgrading, determine the condition of the secondary Controller. You may need to fix a broken secondary Controller before you attempt an upgrade.

The Primary Controller Upgrade Fails

Verify the Primary Upgrade is Unsatisfactory

If the primary upgrade succeeded, but you discovered problems during manual verification, you can roll back the upgrade by performing a failover and rebuilding the secondary.

If the downtime maintenance window is closing, you need to restore the older deployment version and service. Due to the recently performed failover, the current secondary host is a known-good host because it had been functioning as the primary host. You can quickly restore service by failing over to it, and then repairing the host (which experienced the failed upgrade) by rebuilding it as a secondary host.

From the Enterprise Console host:

1. Enter the `ha-failover` command to revert the primary host to the older version:

   ```
   bin/platform-admin.sh submit-job --service controller --job ha-failover --platform-name
   <name_of_the_platform> --args forceFailover=true
   ```

2. Enter the `incremental-replication` command to reduce downtime. When dealing with large data, this is recommended because replication time and downtime depend on data size.

   ```
   bin/platform-admin.sh submit-job --service controller --job incremental-replication --platform-name
   <name_of_the_platform>
   ```

3. Enter the `finalize-replication` command to rebuild the secondary host:

   ```
   bin/platform-admin.sh submit-job --service controller --job finalize-replication --platform-name
   <name_of_the_platform>
   ```

Upgrade the Secondary Controller Fails

If you receive the following error confirming a failed secondary upgrade:

```
Controller: secondary_upgrade_error
```

You can try the following recovery options:

- Option 1: If the failure is recoverable, you can retry the upgrade by entering the following command on the Enterprise Console host:

  ```
  bin/platform-admin.sh submit-job --service controller --job upgrade-secondary --platform-name
  <name_of_the_platform> --args controllerRootUserPassword=<controller_root_password>
  mysqlRootPassword=<mysql_root_password> useCheckpoint=true
  ```

- Option 2: If retrying the upgrade does not work, you can run the incremental-replication and finalize-replication jobs. This involves downtime on the primary. Enter the following commands on the Enterprise Console host:

  ```
  bin/platform-admin.sh submit-job --service controller --job incremental-replication --platform-name
  <name_of_the_platform>
  bin/platform-admin.sh submit-job --service controller --job finalize-replication --platform-name
  <name_of_the_platform>
  ```

  For more details, see Initiate Controller Database Incremental Replication.
- Option 3: If the machine dies or cannot be fixed by running the incremental-replication and finalize-replication jobs, you can remove the secondary Controller. This is a unique fix for an uncommon situation. Enter the following command on the Enterprise Console host:

  ```
  bin/platform-admin.sh submit-job --service controller --job remove --platform-name
  <name_of_the_platform> --args entireCluster=false removeBinaries=true
  ```

The flag, removeBinaries=true, is optional. The use of this flag depends on the situation of your deployment.
If `removeBinaries=false` then the Enterprise Console forgets the Controller without impacting or uninstalling the Controller.

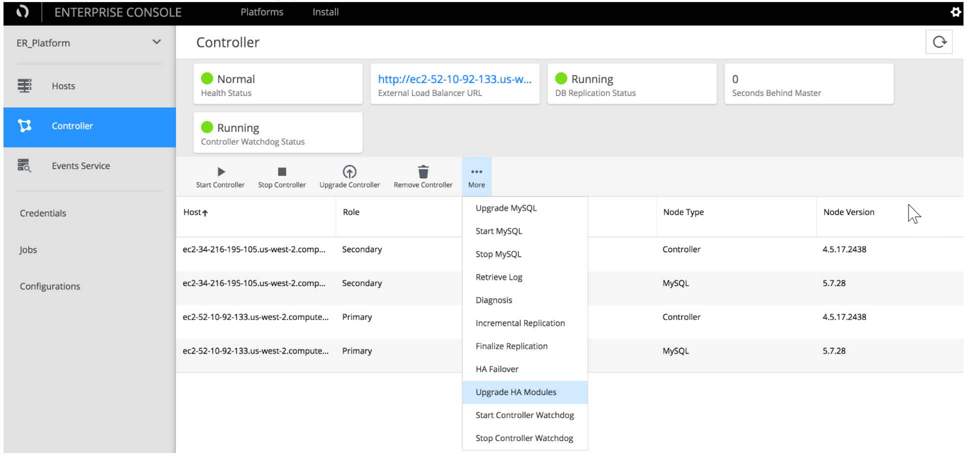
How to Upgrade the HA Modules Without Upgrading the Controller

Using the Enterprise Console UI or the CLI, you can upgrade the HA modules without having to upgrade the Controller. When both Controllers are managed by the Enterprise Console, the HA module is automatically upgraded when you upgrade your HA Controllers.

The following procedure describes an example scenario:

1. You are currently running version 4.5.13 of both Enterprise Console and Controller.
2. You activate the HA modules.
3. You then perform an upgrade only for the Enterprise Console to the latest version (4.5.15 or later).
4. You can upgrade the HA modules from either the Enterprise Console UI or the CLI:

- From the Enterprise Console UI:
  a. Log in to the Enterprise Console and access the Controller page.
  b. From the **More** menu, select **Upgrade HA Modules**.



Normal
Health Status

http://ec2-52-10-92-133.us-w...
External Load Balancer URL

Running
DB Replication Status

0
Seconds Behind Master

Running
Controller Watchdog Status

Host↑	Role	Node Type	Node Version
ec2-34-216-195-105.us-west-2.comp...	Secondary	Controller	4.5.17.2438
ec2-34-216-195-105.us-west-2.comp...	Secondary	MySQL	5.7.28
ec2-52-10-92-133.us-west-2.compute...	Primary	Controller	4.5.17.2438
ec2-52-10-92-133.us-west-2.compute...	Primary	MySQL	5.7.28

- From the CLI, run the following command on the Enterprise Console host to upgrade the HA modules:

```
bin/platform-admin.sh submit-job --job upgrade-ha-modules --service controller
```

The HA modules are upgraded to the latest version without upgrading the Controller. When you upgrade the HA modules, no downtime is required on the Controller, and all HA settings are preserved.

Upgrade the HA Controller Pair Using the Upgrade Wizard

You can use the HA Controller Upgrade Wizard once both of your Controllers (primary and secondary) have been onboarded into the Enterprise Console.

> If are using the HA ToolKit (HATK) and made any customizations to it, AppDynamics recommends that you review your particular situation and determine if you should proceed with the migration. For more details, see HA module in the Enterprise Console.

> Upgrading the HA Controller pair consists of different procedures based on your deployment:
>
> - Follow **Option 1 - Discover and Upgrade** for deployments that are not managed by the Enterprise Console.
>   Use the discover and upgrade job to onboard your HA Controller pair to the Enterprise Console before upgrading the primary Controller.
> - Follow **Option 2 - Upgrade** for HA pair deployments that are managed by the Enterprise Console.
>   Use the upgrade option to upgrade your primary and secondary Controllers managed by the current Enterprise Console instance.

To upgrade using the Upgrade Wizard:

- Step 1: Prepare the Upgrade
- Step 2: Enter Controller Credentials
- Step 3: Perform a Failover
- Step 4: Upgrade the Primary Controller
- Step 5: Verify the Primary Controller Upgrade
- Step 6: Upgrade the Secondary Controller

Step 1: Prepare the Upgrade

Step 2: Enter Controller Credentials

Step 3: Perform a Failover

Step 4: Upgrade the Primary Controller

See Troubleshooting the Upgrade if the primary upgrade fails.

Step 5: Verify the Primary Controller Upgrade

This step pauses the upgrade process and allows you to manually verify the new Controller version.

> Before you proceed with upgrading the secondary Controller, AppDynamics recommends that you verify that the Controller is working by logging in to the Controller and checking that metrics have been received within the last five minutes. If you are not satisfied with the upgrade, you can roll back to the older version from which you started.

Step 6: Upgrade the Secondary Controller

Troubleshooting the Upgrade

Fail Over and Rebuild Secondary

This completes rebuilding the current secondary server so both Controllers in the HA pair are now replicating data.

The Secondary Controller Upgrade Fails

When the secondary Controller upgrade fails:

How to Improve and Optimize Controller Database Performance

> The following procedure only applies to those Controllers that have been upgraded from version 4.2.8 or earlier, to a later version. In such cases, after you upgrade to later version, you may experience performance issues with the Controller database. However, for all new Controller installations using version 4.2.9 or later, the metric data tables are optimized.

To improve database performance when querying metrics, the primary key used by the metric data tables is read *optimized*. As a result, the primary key changes as follows:

From	To
ts_min, node/tier/app, metric_id	metric_id, node/tier/app, ts_min

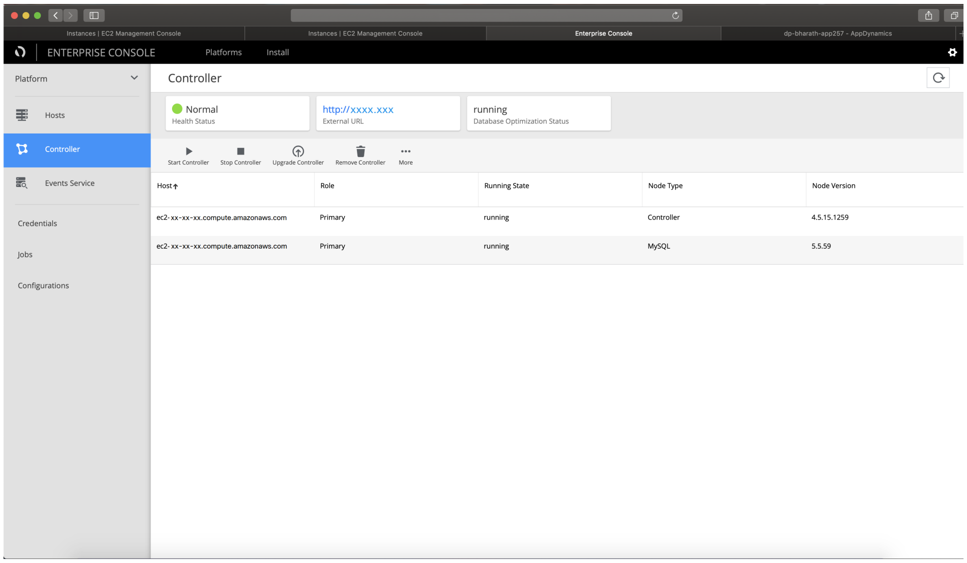
How to Run Database Optimization

You can use the Enterprise Console to run a database optimization job to optimize your database performance. To use this feature:

1. Upgrade your Enterprise Console to the latest version.
2. Upgrade your Controller to the latest version.
3. After the upgrade has completed, for any database table that can be optimized, you can run the database optimization job from the Enterprise Console.

   > No downtime is required and the database optimization job runs automatically.

   Select **Start Database Optimization** from the Controller page to start a process that runs in the background on your primary Controller host.

   Controller

   Normal
   Health Status

   http://xxxx.xxx
   External URL

   running
   Database Optimization Status

Host↑	Role	Running State	Node Type	Node Version
ec2- xx-xx-xx.compute.amazonaws.com	Primary	running	Controller	4.5.15.1259
ec2- xx-xx-xx.compute.amazonaws.com	Primary	running	MySQL	5.5.59

   The process performs several pre-checks to determine if there is enough disk space, and if any other database optimization process is running. The amount of disk space required is determined by the size of the tables to optimize. Based on the amount of Controller data, the database optimization job may take several hours to several days to complete.
4. Once all of the tables have been optimized successfully, the database optimization process completes and no longer displays on the page. To verify that all tables have been optimized, enter and run the following query:

```
cd <controller_home>/bin directory
./controller.sh login-db
        mysql>SELECT table_name
        FROM   information_schema.key_column_usage
        WHERE  table_name LIKE 'metricdata%'
     AND table_name != 'metricdata_min'
     AND table_name != 'metricdata_min_agg'
     AND column_name = 'ts_min'
     AND ordinal_position = 1;
```

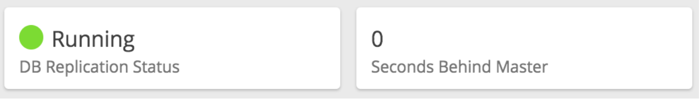
If the query returns any results, then those tables have not been optimized.
If the query returns zero records, then all of the tables were optimized successfully.

The database optimization job is supported on Linux OS only.

How to Run Database Optimization on a High Availability (HA) Controller Pair

Before you run database optimization on a HA Controller pair, you must ensure that the Controller database replication is in a healthy state.

- If both of the Controllers are onboarded into Enterprise Console, review the Controller page and note the following fields:



● Running
DB Replication Status

0
Seconds Behind Master

- If one of the Controllers is managed by HA toolkit (HATK):

a. Log in to the primary Controller host and enter:

```
cd <controller_home>/bin directory
```

b. Log in to the secondary Controller database and enter:

```
./controller.sh login-db
```

c. Enter:

```
SHOW SLAVE STATUS\G;
```

This results in the following output:

```
Seconds_Behind_Master: $Number_Of_Seconds_Behind_Master
```

If a non-zero number displays the output for this test, wait until the number changes to zero.

d. After you ensure that replication is working as expected, you can run the database optimization job from the Enterprise Console. Select **Start Database Optimization** from the Controller page to start a process that runs in the background on your primary Controller host. The process performs several pre-checks to determine if there is enough disk space, and if any other database optimization process is running. The amount of disk space required is determined by the size of the tables to optimize. Based on the amount of Controller data, the database optimization job may take several hours to several days to complete.

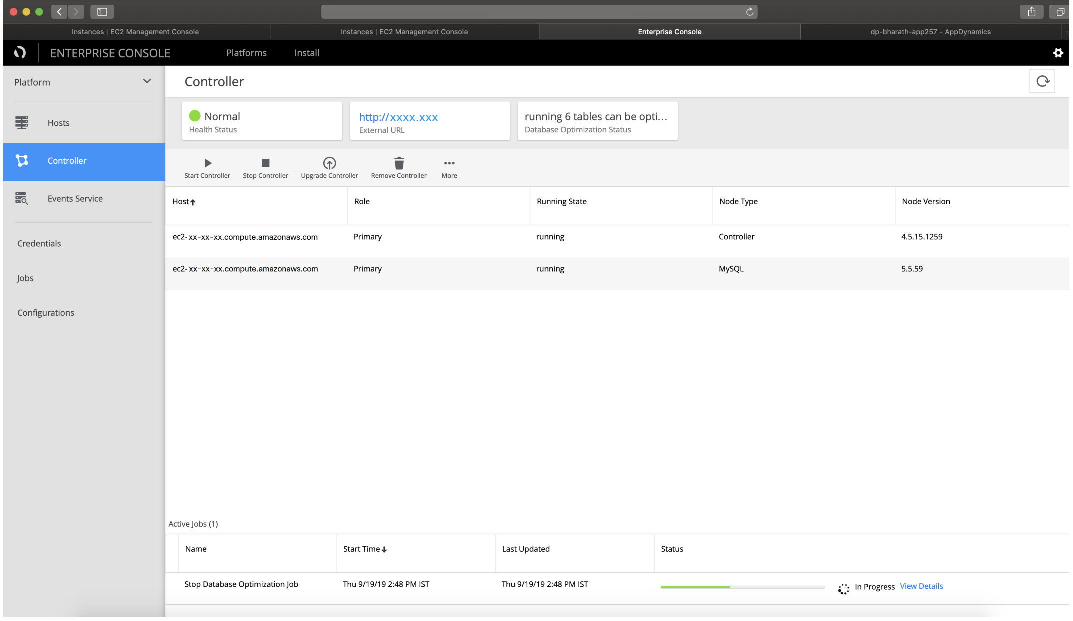
e. Once all of the tables have been optimized successfully, the database optimization process completes and no longer displays on the page. To verify that all tables have been optimized, enter and run the following query:

```
cd <controller_home>/bin directory
./controller.sh login-db
        mysql>SELECT table_name
        FROM   information_schema.key_column_usage
        WHERE  table_name LIKE 'metricdata%'
     AND table_name != 'metricdata_min'
     AND table_name != 'metricdata_min_agg'
     AND column_name = 'ts_min'
     AND ordinal_position = 1;
```

If the query returns any results, then those tables have not been optimized.
If the query returns zero records, then all of the tables were optimized successfully.

How to Stop Database Optimization

After the database optimization job has completed successfully, you can stop the process. From the Enterprise Console, select **Stop Database Optimization** from the Controller page:



Normal
Health Status

http://xxxx.xxx
External URL

running 6 tables can be opti...
Database Optimization Status

Host↑	Role	Running State	Node Type	Node Version
ec2-xx-xx-xx.compute.amazonaws.com	Primary	running	Controller	4.5.15.1259
ec2-xx-xx-xx.compute.amazonaws.com	Primary	running	MySQL	5.5.59

Active Jobs (1)

Name	Start Time↓	Last Updated	Status
Stop Database Optimization Job	Thu 9/19/19 2:48 PM IST	Thu 9/19/19 2:48 PM IST	In Progress View Details

ⓘ You may need to stop the database optimization process if it is using too many resources and you notice a performance impact on the Controller, or if you decide to reschedule the process to run at a later date.

Troubleshooting Database Optimization

The following table describes possible conditions that may cause errors to occur and actions to take to mitigate them:

Errors or Conditions	User Action
Job failed; `Database replication is broken` message displays.	Re-establish database replication incrementally, then finalize replication.
Ran out of disk space while the database optimization job was running, and job stops processing.	Free up disk space and restart database optimization job.

Uninstall the Controller

This page describes how to uninstall the Controller software and associated files from a platform using the Enterprise Console.

Before Starting

If you have installed the Events Service with the Enterprise Console, it is recommended that you uninstall the Events Service before you uninstall the Controller. See Uninstall the Events Service for more information.

In addition, if you have the EUM Server, Application Analytics, or other product modules installed, keep in mind that if you reinstall the Controller later, you will need to configure integration settings for the modules manually.

Optionally, stop the Controller before uninstalling as described in Start or Stop the Controller. If you do not stop the Controller, the uninstaller will do so for you. However, if your database or Controller generally take a long time to shut down, you can avoid the possibility of time-out errors during uninstallation by stopping the services manually.

Uninstall the Controller Using the Enterprise Console

You can uninstall the Controller on the Controller page in the GUI or by completing the following steps:

1. Open a console:
   - On Linux, open a terminal window and switch to the user who installed the Controller or to a user with equivalent directory permissions.
   - On Windows, open an elevated command prompt by right-clicking on the Command Prompt icon in the Windows Start Menu and choosing **Run as Administrator**.
2. From the command line, navigate to the Enterprise Console bin directory, `platform-admin/bin`.
3. Run the following command:

Note that you cannot use the other AppDynamics platform components without a Controller, so you must install a new Controller before you can resume using the platform.

EUM Server Deployment

The default End User Monitoring deployment assumes that EUM agents (Mobile and Browser) send their data to the EUM Cloud, a cloud-based processor. To deploy EUM completely on-premises, you need to install the EUM Server, the on-premises version of the EUM Cloud, as described here.

Installation Overview

The EUM Server receives data from EUM agents, processes and stores that data, and makes it available to the AppDynamics Controller. Certain EUM features—specifically, Browser Request Analytics and Mobile Request Analytics, features of Application Analytics that extend the functionality of Browser and Mobile Analyze—require access to the AppDynamics Events Service.

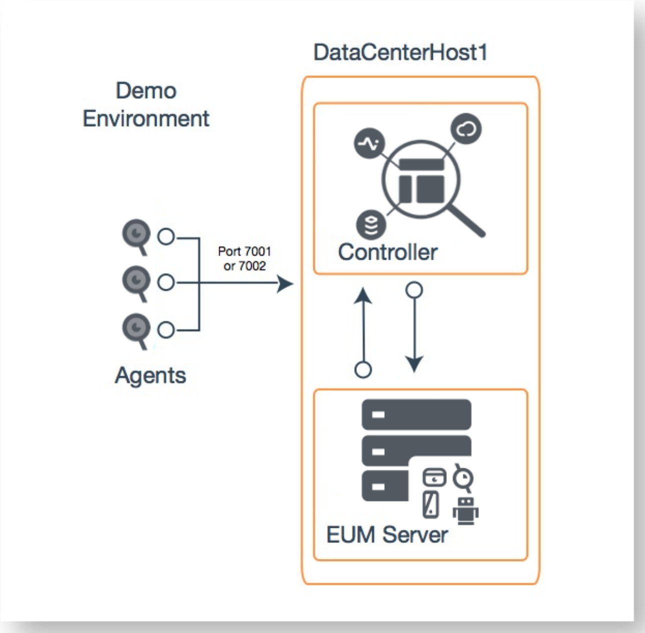
To set up a complete on-premises EUM Server deployment, therefore, you need to:

1. Determine which version of the EUM Server is compatible with your other platform components.
2. Install the on-premises Controller or prepare an in-service Controller to work with the EUM Server
3. Install the on-premises Events Service Deployment and configure it to work with your on-premises Controller
4. Install the on-premises EUM Server and configure it to work with your Events Service and Controller.

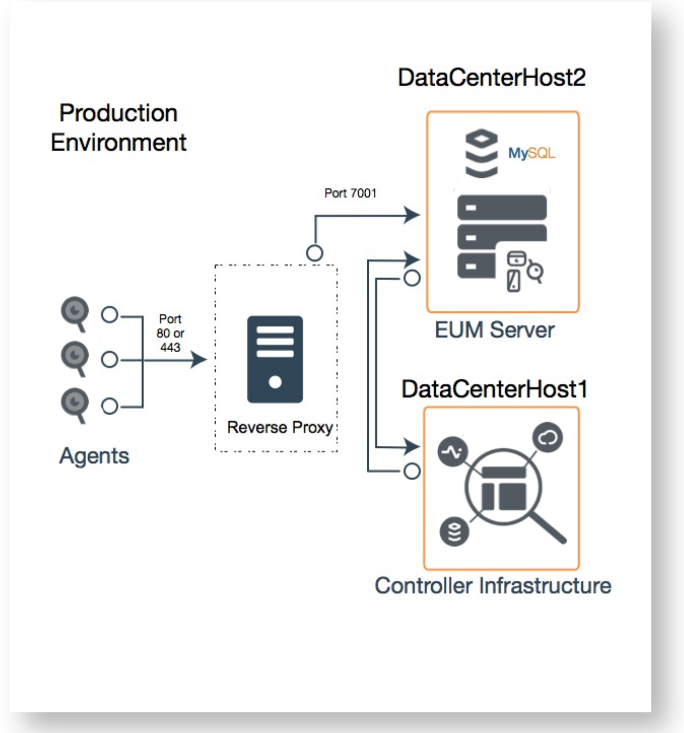
Deployment Modes for the EUM Server

For demonstration and light testing purposes, choose the Demo installation option, where the EUM Server and Controller are installed on the same host, and the EUM Server shares the Controller's MySQL instance. For production installation, choose the Product installation option, where the EUM Server and Controller sit on different hosts, and the EUM Server hosts its own MySQL instance.

In Demo mode, the EUM Server listens for connections on port 7001 or 7002. The secure port, 7002, uses a built-in, self-signed certificate, which is only used in demo mode.



In a production environment, the EUM Server is likely to operate behind a reverse proxy. A reverse proxy relieves the performance burden of SSL termination from the EUM Server. It also helps ease certificate management and security administration in general. Further, as the connection point for agent beacons, the Server needs to have the security layer of a proxy between itself and the external Internet.



ⓘ Using a reverse proxy is the recommended method of setting up HTTPS connections for an on-premises EUM Server. If this is not possible in your installation, however, it is possible to set HTTPS support manually. See information on setting up a custom keystore in Secure the EUM Server.

Embedded Geo Server

The EUM Server includes an embedded Geo Server that provides the geo information. The Geo Server either obtains geo information from your custom Geo Server or by resolving incoming IPv4 address (IPv6 addresses are *not* supported) with Neustar data or custom geo data.

Add New Geo Data

To add new geo data, you can either add a new Neustar data file or create the custom geo data file geo-ip-mappings.xml and place it in the directory `eum-processor/bin/`. The EUM Server automatically detects and loads new geo data files.

Update the Geo Server

The Geo Server is updated when you update the EUM Server.

Host Your Own Geo Server

Follow the instructions given in Install and Host a Custom Geo Server for Browser RUM.

Check Controller Version

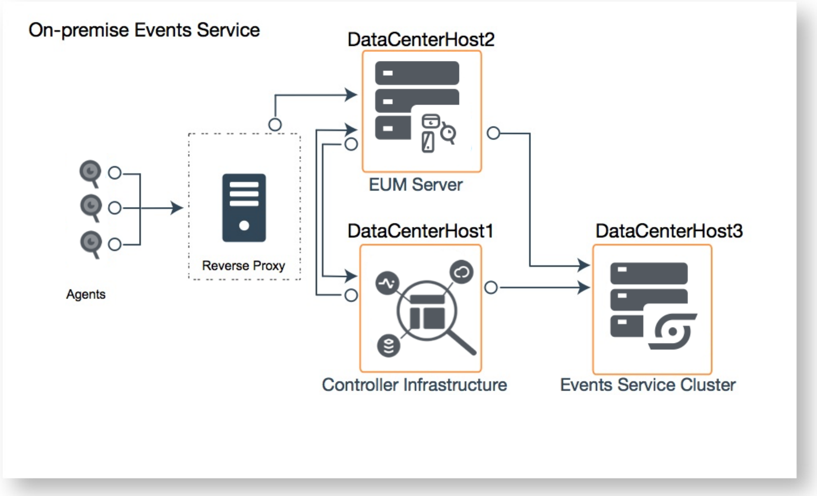
Before you run the EUM Server installer:

1. Check your Controller version. The 4.5 EUM Server works with the AppDynamics Controller version 4.5 or earlier. A Controller works with a EUM Server that is the same or a later version, which includes the major, minor, and patch version. Thus, a version 4.5.2 Controller works with a 4.5.2 or later version of the EUM Server, but that version 4.5.2 Controller does *not* work with a 4.5.1 or 4.5.0 version of the EUM Server. See Upgrade the Production EUM Server on information about upgrading the platform.
2. Back up the current version of your Controller.
3. Choose a time window that has minimum impact on service availability.

Install the On-Premises Events Service

The Analyze function in Browser RUM and the Crash Report and Analyze components in Mobile RUM rely on the AppDynamics Events Service, the Platform's unstructured document store. The Events Service that is configured by default for EUM is a cloud-based service.

If you are running on-premises and wish to keep all your processing on-premises, after installing and configuring the Controller, you must install an on-premises version of the Events Service as described in this section. Note that relying on the Events Service purely for use with the EUM UI does not require a separate Application Analytics license. Other uses *may* require a separate license.



There are multiple modes of deploying the Events Service. For detailed information on installing and configuring the Events Service, see Events Service Deployment.

Run the EUM Server Installer

Before starting, get the installer version appropriate for your target system. You can get the installer from the AppDynamics download site.

Run the EUM installer under the same user account on the target machine like the one used to install the Controller, or using an account that has read, write, and execute permissions to the Controller home directory. Installing with incompatible permission levels—for example, attempting to install the EUM Server as a regular user while the Controller was installed by root user—may result in installation or operation errors.

The EUM Server is automatically installed as a Windows service. All upgrades are automatically converted to a Window service.

The installer can be run in three modes:

- GUI
- Console
- Silent mode with `varfile`

See the following page for details on installing as appropriate for your deployment mode:

- For demo mode, see Install a Demo EUM Server.
- For production mode, see Install a Production EUM Server.

Update the Agents

You must update the address that agents use to send their beacons to the EUM Server based on your configuration. For Browser Real User Monitoring, the Controller updates the JavaScript agent. Simply re-download and deploy it, as described in Set Up and Configure Browser RUM. For Mobile RUM, the mobile applications themselves need to be updated, using the mobile SDKs. For more information, see Customize the Android Instrumentation and Customize the iOS Instrumentation.

Start and Stop the EUM Server and Database

The EUM Server is installed as a Windows service automatically. You can manage how you want this service to run using the Local Services dialog.

Start/Stop the EUM Server

On Linux, start the EUM server from the `eum-processor` directory in the EUM home as follows:

```
bin/eum.sh start
```

For a demonstration environment, run the command as sudo.

On Windows, if you ever need to start the EUM Server manually, you can do so by running:

```
bin\eum-processor.bat start
```

You can check if the server is running and accessible by going to `http://<hostname>:7001/eumaggregator/ping` with your browser. Your browser should display `ping.`

To stop the EUM Server, pass the stop command to the `eum` script. For example, on Linux, from the `eum-processor` directory, run:

```
bin/eum.sh stop
```

You can also start and stop the EUM database. On Windows, you can do so from the Windows Services.

Start/Stop the EUM MySQL Database

On Linux, you can start MySQL by navigating to the directory, `<EUM>/orcha/orcha-master/bin`, and running:

```
./orcha-master -d mysql.groovy -p ../../playbooks/mysql-orcha/start-mysql.orcha -o ../conf/orcha.properties -c
local
```

To stop MySQL on Linux, run:

```
./orcha-master -d mysql.groovy -p ../../playbooks/mysql-orcha/stop-mysql.orcha -o ../conf/orcha.properties -c
local
```

EUM Server Requirements

Related pages:

- On-premises EUM Server Sizing Guide
- Sizing XL profiles for on premises EUM Server and the Events Service: What's recommended?

This page lists the EUM Server requirements, offers sizing guidance, and shows you how to use configuration to modify the default settings. For additional EUM Processor sizing information, see Analytics' Recipe Book for on-prem configuration in the AppDynamics Community.

Hardware Requirements

The requirements and guidelines for the EUM Server machine (basic usage) are as follows:

- Minimum 50 GB extra disk space. See Disk Requirements Based on Resource Timing Snapshots to learn when more disk space is needed.
- 64-bit Windows or Linux operating system
- Processing: 4 cores
- 10 Mbps network bandwidth
- Minimum 8 GB memory total (4 GB is defined as max heap in JVM). See RAM Requirements Based on the Beacon Load to learn when more RAM is required.
- NTP enabled on both the EUM Server host and the Controller machine. The machine clocks need to be able to synchronize.

> ⓘ A machine with these specs can be expected to handle around 10K page requests a minute or 10K simultaneous mobile users. Adding on-premises Analytics capability requires increasing these requirements—particularly disk space—considerably, depending on the use case.

RAM Requirements Based on the Beacon Load

Beacons are sent to the EUM Server every 10 seconds, and each beacon can contain data for multiple events. You can configure the JavaScript Agent to limit the number of Ajax requests.

The table below specifies the required RAM based on your beacon load per minute and lists the content of a typical beacon.

Peak Beacons Per Minute	Typical Beacon Composition	RAM
~3K	600 sessions 1K base pages 2K virtual pages 7K Ajax requests	8 GB
~16K	1.8K sessions 5K base pages 10K virtual pages 40K Ajax requests	16 GB
~26K	3.6K sessions 8K base pages 17K virtual pages 62 Ajax requests	16 GB
~33K	3.9K sessions 10K base pages 20K virtual pages 74K Ajax requests	32 GB

>40K	12K base pages	32 GB

Disk Requirements Based on Resource Timing Snapshots

By default, the EUM Server accepts a maximum of 1K resource timing snapshots per minute and retains those snapshots for 15 days. On average, each snapshot takes 3 KB of disk space.

Because of the number of resource timing snapshots impact disk usage, you should follow the guidelines in the table below.

Number of Resource Timing Snapshots	Recommended Disk Space
~500	40 GB
~1000	64 GB
~1500	96 GB
~2000	128 GB

If needed, you can reduce the number of resource timing snapshots or reduce the disk space allotted for storing resource snapshots by doing one or more of the following:

- Configure the JavaScript Agent to modify and limit the number of resources to monitor.
- Use the EUM Server configuration `onprem.resourceSnapshotAllowance` to specify the maximum disk space allotted for storing resource snapshots. See EUM Server Configuration File for a complete list of configurations.
- Limit the number of snapshots retained by the EUM server by setting a global maximum, reducing the time that they are retained, or by filtering snapshots based on the network response time. See Limit the Number of EUM Snapshots for instructions.

Filesystem Requirements

The filesystem of the machine on which you install EUM should be tuned to handle a large number of small files. In practical terms, this means that either the filesystem should be allocated with a large number of inodes or the filesystem should support dynamic inode allocation.

Controller Version

The AppDynamics Platform you use with the EUM server must have a supported Controller version installed. Controllers only work with the same or later versions of the EUM Server. For example, the 4.5 EUM Server works with the AppDynamics Controller version 4.5 or earlier.

Open File Descriptor and User Process Limits

On Linux, also ensure that open file descriptor and user process limits on the EUM Server machine are set to a sufficient value. For the EUM Server, the hard and soft limits should be as follows:

- Open file descriptor limit (`nofile`): 65535
- Process limit (`nproc`): 8192

See "Configure User Limits in Linux" below for information on how to check and set user limits.

Configure User Limits in Linux

The following log warnings may indicate insufficient limits:

- Warning in database log: "Could not increase number of max_open_files to more than xxxx".
- Warning in server log: "Cannot allocate more connections".

To check your existing settings, as the root user, enter the following commands:

```
ulimit -S -n
ulimit -S -u
```

The output indicates the soft limits for the open file descriptor and soft limits for processes, respectively. If the values are lower than recommended, you need to modify them.

Where you configure the settings depends upon your Linux distribution:

- If your system has a `/etc/security/limits.d directory`, add the settings as the content of a new, appropriately named file under the directory.
- If it does not have a `/etc/security/limits.d` directory, add the settings to `/etc/security/limits.conf`.
- If your system does not have a `/etc/security/limits.conf` file, it is possible to put the `ulimit` command in `/etc/profile`. However, check the documentation for your Linux distribution for the recommendations specific for your system.

To configure the limits:

1. Determine whether you have a `/etc/security/limits.d` directory on your system, and take one of the following steps depending on the result:
    - If you *do not* have a `/etc/security/limits.d` directory:
        a. As the root user, open the `limits.conf` file for editing: `/etc/security/limits.conf`
        b. Set the open file descriptor limit by adding the following lines, replacing `<login_user>` with the operating system username under which the EUM Server runs:

        ```
        <login_user> hard nofile 65535
        <login_user> soft nofile 65535
        <login_user> hard nproc 8192
        <login_user> soft nproc 8192
        ```

    - If you *do* have a `/etc/security/limits.d` directory:
        a. As the root user, create a new file in the `limits.d` directory. Give the file a descriptive name, such as the following: `/etc/security/limits.d/appdynamics.conf`
        b. In the file, add the configuration setting for the limits, replacing `<login_user>` with the operating system username under which the EUM Server runs:

        ```
        <login_user> hard nofile 65535
        <login_user> soft nofile 65535
        <login_user> hard nproc 8192
        <login_user> soft nproc 8192
        ```

2. Enable the file descriptor and process limits as follows:
    a. Open the following file for editing: `/etc/pam.d/common-session`
    b. Add the line: `session required pam_limits.so`
3. Save your changes to the file.

When you log in again as the user identified by `login_user`, the limits will take effect.

Network Settings

The network settings on the operating system need to be tuned for high-performance data transfers. Incorrectly tuned network settings can manifest themselves as stability issues on the EUM Server.

The following command listing demonstrates tuning suggestions for Linux operating systems. As shown, AppDynamics recommends a TCP/FIN timeout setting of 10 seconds (the default is typically 60), the TCP connection keepalive time to 1800 seconds (reduced from 7200, typically), and disabling TCP window scale, TCP SACK, and TCP timestamps.

```
echo 5 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 1800 >/proc/sys/net/ipv4/tcp_keepalive_time
echo 0 >/proc/sys/net/ipv4/tcp_window_scaling
echo 0 >/proc/sys/net/ipv4/tcp_sack
echo 0 >/proc/sys/net/ipv4/tcp_timestamps
```

The commands demonstrate how to configure the network settings in the `/proc` system. To ensure the settings persist across system reboots, be sure to configure the equivalent settings in the `etc/sysctl.conf`, or the network stack configuration file appropriate for your operating system.

Required Libraries

- `libaio`
- `tar`

libaio Requirement

The EUM processor requires the `libaio` library to be on the system. This library facilitates asynchronous I/O operations on the system. Note if you have a NUMA based architecture, then you are required to install the `numactl` package.

Install `libaio` on the host machine if it does not already have it installed. The following table provides instructions on how to install `libaio` for some common flavors of the Linux operating system.

Linux Flavor	Command
Red Hat and CentOS	Use `yum` to install the library, such as: • `yum install libaio` • `yum install numactl`
Fedora	Install the library RPM from the Fedora website: • `yum install libaio` • `yum install numactl`
Ubuntu	Use apt-get, such as: • `sudo apt-get install libaio1` • `sudo apt-get install numactl`
Debian	Use a package manager such as APT to install the library (as described for the Ubuntu instructions above).

tar Requirement

You will need the `tar` utility to unpack the EUM Server installer.

Install `tar` on the host machine if it does not already have it installed. The following table provides instructions on how to install `tar` for some common flavors of the Linux operating system.

Linux Flavor	Command
Red Hat and CentOS	Use `yum` to install the library, such as: • `yum install tar`
Fedora	Install the library RPM from the Fedora website: • `yum install tar`
Ubuntu	Use `apt-get`, such as: • `sudo apt-get install tar`
Debian	Use a package manager such as APT to install the library (as described for the Ubuntu instructions above).

Install a Production EUM Server

You can run the installer using one of three methods:

- Interactive console mode
- GUI Installer
- Silent Installer

The GUI and silent installation methods are described below. To start the installer using interactive console mode, start the installer with the `-c` switch. The console mode prompts you for the equivalent information that appears in the GUI installer screens.

Additionally, you can run the installer using a Response file (for unattended installations). See Installing with the Silent Installer.

Requirements

- Before starting, download the installer distribution and extract it on the target machine. You obtain the EUM installer from the AppDynamics Download Center.
- To secure connections from agents to the EUM Server, AppDynamics strongly recommends that SSL traffic is terminated at a reverse proxy that sits in front of the EUM Server in the network path, and forwards connections to the EUM Server. However if this is not possible in your installation, it is possible to connect with HTTPS directly to the EUM Server. For information on setting up a custom keystore for production, see Secure the EUM Server.

  > ⚠ If you install and configure the Events Service with HTTPS support, you must perform a workaround for your EUM Server installation to complete properly. After the Events Service certificate configuration, install the EUM Server without Analytics enabled. Then, install the certificate into the EUM Server keystore following the steps described on the Secure the EUM Server page. Configure Analytics in the Events Services Properties, and restart the EUM Server.

- Before you install the EUM Server, Linux systems must have the `libaio` library installed. See the EUM Server Requirements.

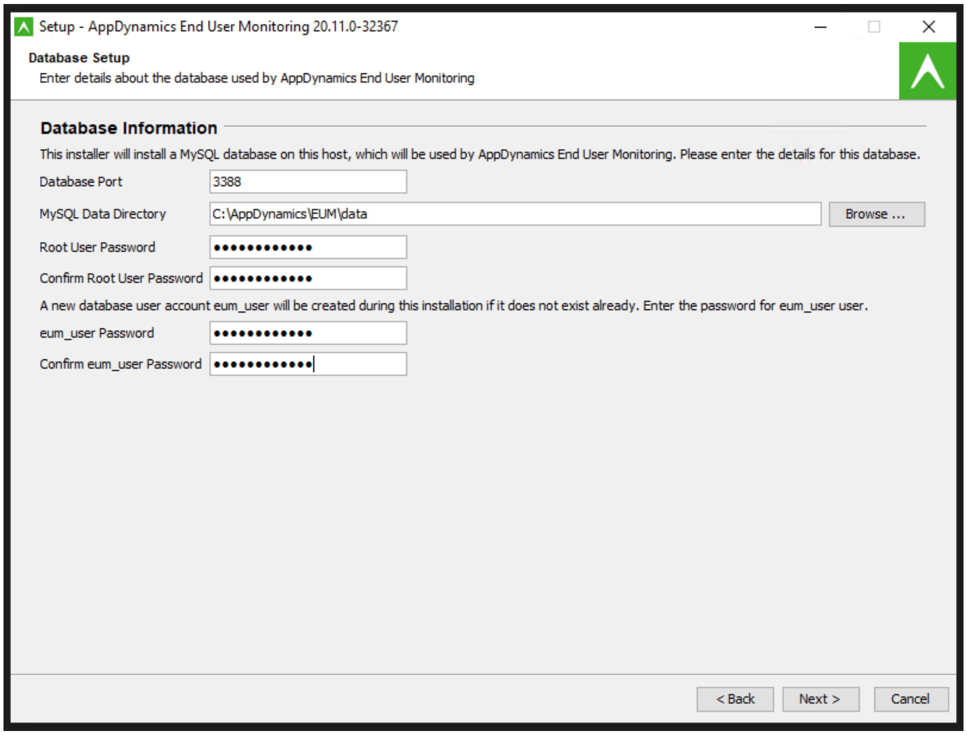
Install the EUM Server for a Production Deployment with the GUI Installer

Run the on-premises EUM installer on the machine on which you want to install the EUM Server.

1. Start the installer:

2. In the Welcome screen, click **Next** to continue.
3. Scroll to the end of the license agreement and accept the license agreement, then click **Next** to continue.
4. Select the destination directory, and click **Next** to continue.
5. Choose **Product** for the installation mode. This mode installs the EUM Server on this machine. Use this type if AppDynamics End User Monitoring and the AppDynamics Controller are installed on different hosts. Selecting this type will install a separate MySQL instance on this machine. Click **Next**.
6. In the Database Setup screen:
   a. Enter a new **Root User Password** and confirm it.
   b. Enter a new **eum_user Password** and confirm it.

c. Click **Next**.



Setup - AppDynamics End User Monitoring 20.11.0-32367

Database Setup

Enter details about the database used by AppDynamics End User Monitoring

Database Information

This installer will install a MySQL database on this host, which will be used by AppDynamics End User Monitoring. Please enter the details for this database.

Database Port: 3388

MySQL Data Directory: C:\AppDynamics\EUM\data

Root User Password: ●●●●●●●●●●●●

Confirm Root User Password: ●●●●●●●●●●●●

A new database user account eum_user will be created during this installation if it does not exist already. Enter the password for eum_user user.

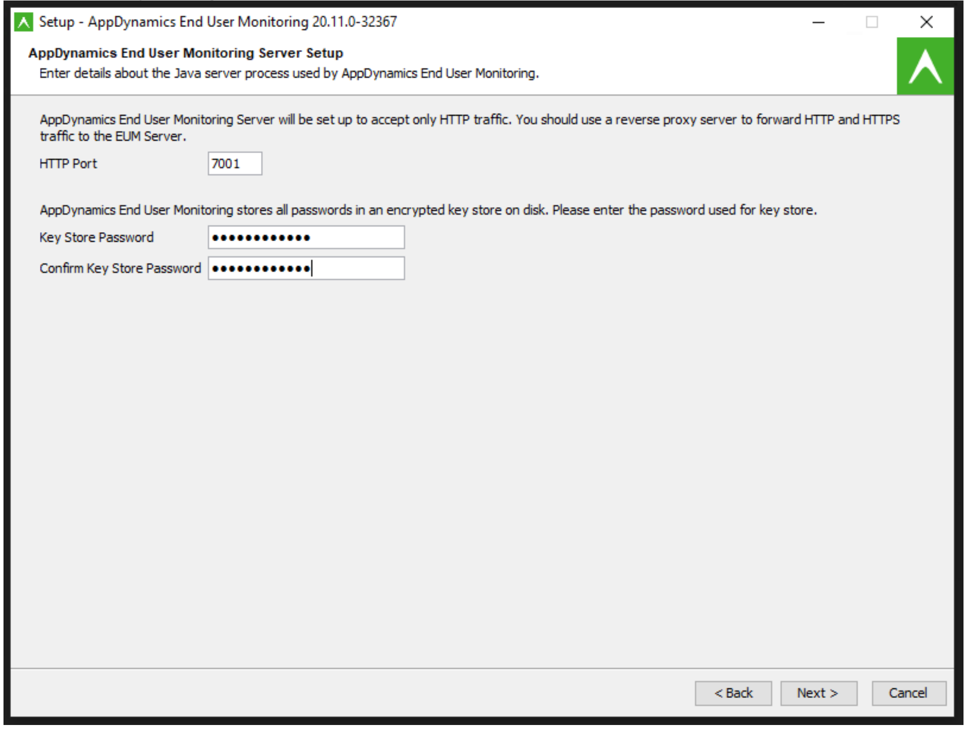
eum_user Password: ●●●●●●●●●●●●

Confirm eum_user Password: ●●●●●●●●●●●●

Usernames and passwords can only consist of ASCII characters. In addition, passwords cannot include the characters '^', '/', or '$'.

7. In the AppDynamics End User Monitoring Setup screen:

   a. Enter a new **key store password** and confirm it.



Setup - AppDynamics End User Monitoring 20.11.0-32367

AppDynamics End User Monitoring Server Setup

Enter details about the Java server process used by AppDynamics End User Monitoring.

AppDynamics End User Monitoring Server will be set up to accept only HTTP traffic. You should use a reverse proxy server to forward HTTP and HTTPS traffic to the EUM Server.

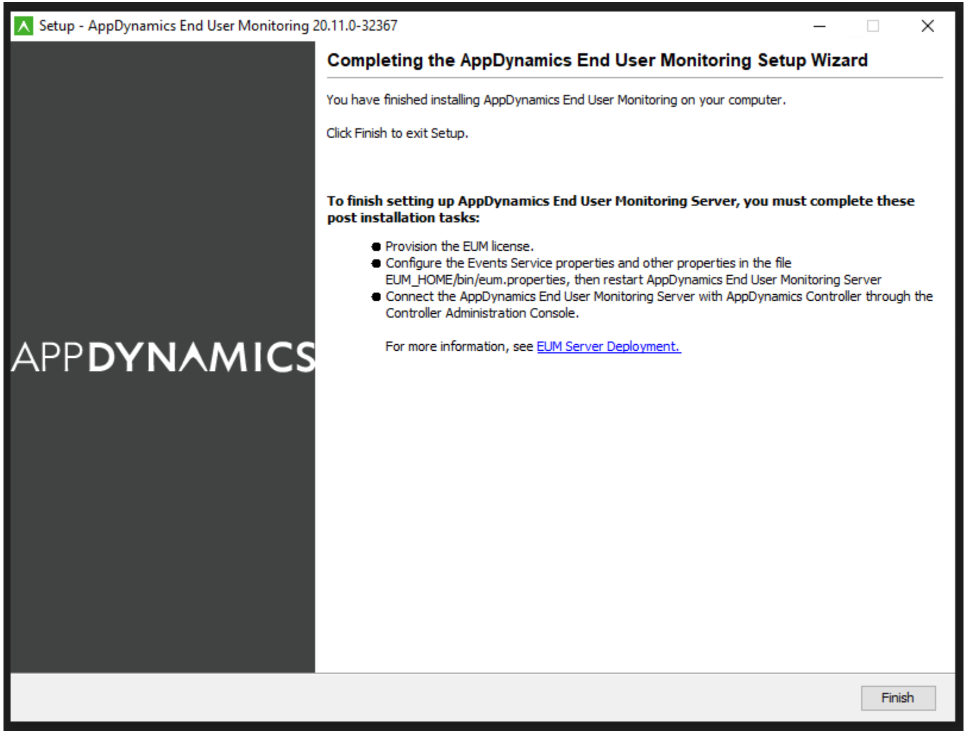
HTTP Port: 7001

AppDynamics End User Monitoring stores all passwords in an encrypted key store on disk. Please enter the password used for key store.

Key Store Password: ●●●●●●●●●●●●

Confirm Key Store Password: ●●●●●●●●●●●●

a. Click **Next**, then click **Finish**.



Completing the AppDynamics End User Monitoring Setup Wizard

You have finished installing AppDynamics End User Monitoring on your computer.

Click Finish to exit Setup.

To finish setting up AppDynamics End User Monitoring Server, you must complete these post installation tasks:

- Provision the EUM license.
- Configure the Events Service properties and other properties in the file EUM_HOME/bin/eum.properties, then restart AppDynamics End User Monitoring Server
- Connect the AppDynamics End User Monitoring Server with AppDynamics Controller through the Controller Administration Console.

For more information, see EUM Server Deployment.

This completes the initial configuration and setup of the EUM Server. When finished, the EUM Server is running.

Post-Installation Tasks

To complete the AppDynamics EUM Server installation, you must perform these additional post-installation tasks (as shown in the last AppDynamics End User Monitoring Setup Wizard screen):

- Configure JVM options
- Provision the EUM license
- Configure the Events Services properties in the `eum.properties` file
- Connect the EUM Server with the AppDynamics Controller
- Secure the EUM Server by setting up a custom keystore

> The EUM Server Installer only configures the HTTP port.

Configure JVM Options

Provision the EUM License

Follow the provision instructions based on your deployment type:

- Provision the EUM License for a Single-Tenant Controller
- Provision the EUM License for Multi-Tenant Controllers

Configure the Events Services Properties

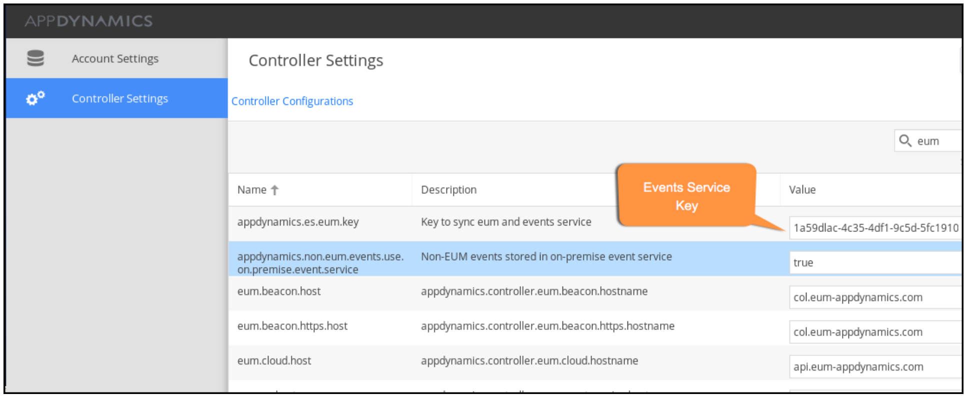
Configure the Events Services properties in the `eum.properties` file:

1. Ensure that Events Services is running.
2. Navigate to the `\EUM\eum-processor\bin` directory.
3. Open the `eum.properties` file to edit.
4. In the `eum.properties` file, enter these values:

Sample

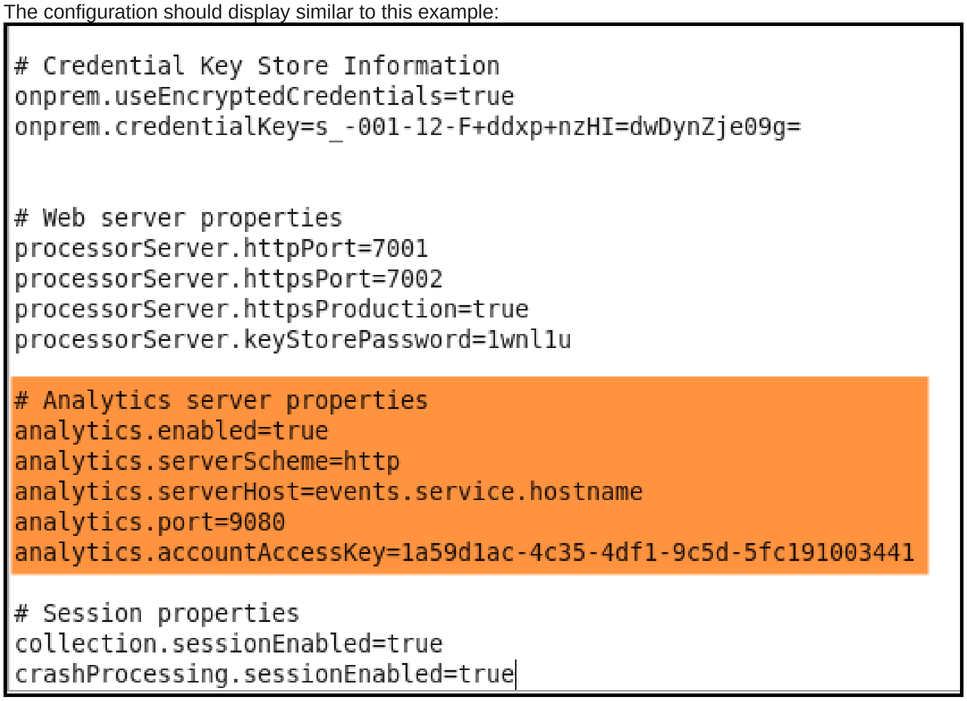
```
analytics.enabled=true
analytics.serverScheme=http
analytics.serverHost=hostname-events-service (needs to be the hostname of your Events Service)
analytics.port=9080
analytics.accountAccessKey=1a59d1ac-4c35-4df1-9c5d-5fc191003441
```

The <analytics.accountAccessKey> is the Events Service key that appears as the appdynamics.es.eum.key value in the Administration Console:



Name ↑	Description	Value
appdynamics.es.eum.key	Key to sync eum and events service	1a59dlac-4c35-4df1-9c5d-5fc1910
appdynamics.non.eum.events.use.on.premise.event.service	Non-EUM events stored in on-premise event service	true
eum.beacon.host	appdynamics.controller.eum.beacon.hostname	col.eum-appdynamics.com
eum.beacon.https.host	appdynamics.controller.eum.beacon.https.hostname	col.eum-appdynamics.com
eum.cloud.host	appdynamics.controller.eum.cloud.hostname	api.eum-appdynamics.com

The configuration should display similar to this example:



```
# Credential Key Store Information
onprem.useEncryptedCredentials=true
onprem.credentialKey=s_-001-12-F+ddxp+nzHI=dwDynZje09g=


# Web server properties
processorServer.httpPort=7001
processorServer.httpsPort=7002
processorServer.httpsProduction=true
processorServer.keyStorePassword=1wnl1u

# Analytics server properties
analytics.enabled=true
analytics.serverScheme=http
analytics.serverHost=events.service.hostname
analytics.port=9080
analytics.accountAccessKey=1a59d1ac-4c35-4df1-9c5d-5fc191003441

# Session properties
collection.sessionEnabled=true
crashProcessing.sessionEnabled=true
```

5. After updating the eum.properties file, restart the EUM Server.

Connect the EUM Server with the AppDynamics Controller

Connect the EUM Server with the AppDynamics Controller:

1. Log in to the Administration Console.
2. Set these Controller properties:
    - eum.cloud.host: http://eum-host-name:7001 – Location where the Controller will poll for EUM metrics.
    - eum.beacon.host: http://eum-host-name:7001 – Location where the JavaScript Agent will be configured to send out beacons over the HTTP protocol.
    - eum.beacon.https.host: https://eum-host-name:7002 – Location where JavaScript Agent will be configured to send out beacons over the HTTPS protocol.
    - eum.mobile.screenshot.host: http://host-name:7001 – Location where the Controller will search for mobile screenshots.

Installing with the Silent Installer

Instead of using the GUI installer, you can use the silent installer to perform an unattended installation. The silent installer uses a response file as a source for the initial configuration settings. It's useful for scripting installation or performing large scale deployments.

To use a response file for installation:

1. Create a file named `response.varfile` on the machine on which you will run EUM installer and include the following:

```
sys.adminRights$Boolean=false
sys.languageId=en
sys.installationDir=/AppDynamics/EUM
euem.InstallationMode=split
euem.Host=eumhostname
euem.initialHeapXms=1024
euem.maximumHeapXmx=4096
euem.httpPort=7001
euem.httpsPort=7002
mysql.databasePort=3388
mysql.databaseRootUser=root
mysql.dbHostName=localhost
mysql.dataDir=/usr/local/AppDynamics/EUM/data
mysql.rootUserPassword=singcontroller
mysql.rootUserPasswordReEnter=singcontroller
eumDatabasePassword=secret
eumDatabaseReEnterPassword=secret
keyStorePassword=secret
keyStorePasswordReEnter=secret
eventsService.isEnabled$Boolean=true
eventsService.serverScheme=http
eventsService.host=eventsservice_host
eventsService.port=9080
eventsService.APIKey=1a234567-1234-1234-4567-ab123456
```

2. Modify values of the installation parameters based on your own environment and requirements. Particularly ensure that the directory paths and passwords match your environment.
3. Run the installer with the following command:

Install a Demo EUM Server

You can run the installer in one of three modes. The GUI and silent installation methods are described below. To start the installer in interactive console mode, start the installer with the `-c` switch. The console mode prompts you for the equivalent information that appears in the GUI installer screens, as described below.

In addition, you can run the installer using a Response file (for unattended installations). See Installing with the Silent Installer.

About the Demo Installation

This mode is for demonstration and light testing only. If you are using the Events Service, it must be on a separate host.

If you do not already have an existing on-premises Controller, install it as described in Custom Install.

Installation Requirements

To install the Demo Installation, you are required to do the following:

- Install and run a Controller instance on the same host machine before starting the EUM Server installation.
- Install the EUM Server with the same user account used to install the Controller, or use an account that has read, write, and execute permissions to the Controller home directory.

Installing with the GUI Installer

1. Start the installer:
    - On Linux:

        a. From a command prompt, navigate to the directory to which you downloaded the EUM Server installer.
        b. Change permissions on the downloaded installer script to make it executable, as follows:

        ```
        chmod 775 euem-64bit-linux-4.5.x.x.sh
        ```

        c. Run the script as follows:

        ```
        ./euem-64bit-linux-4.5.x.x.sh
        ```

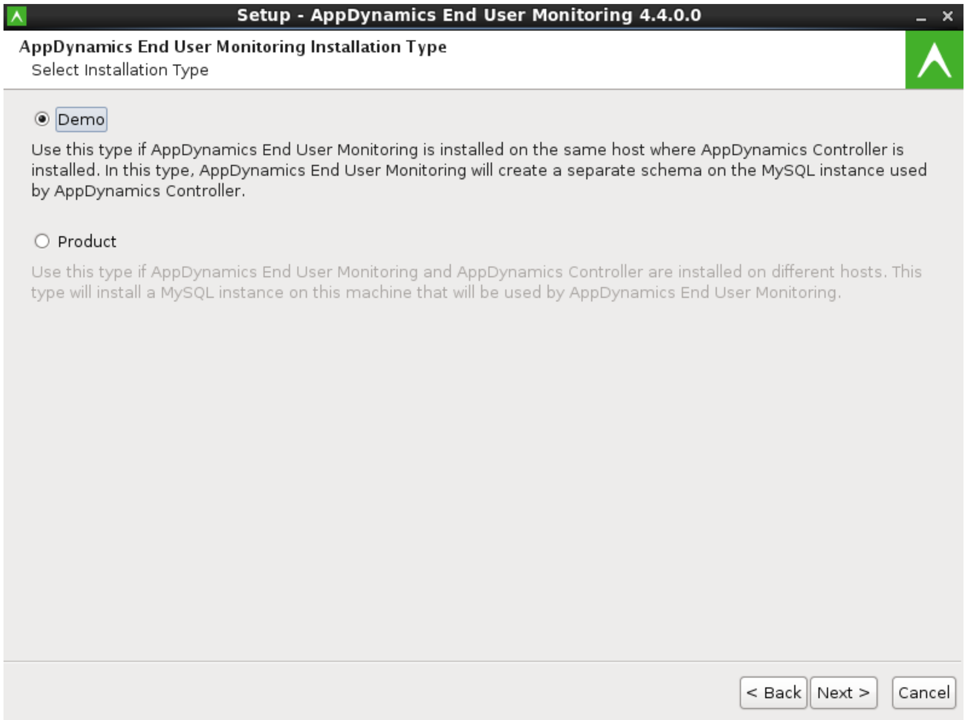
    - On Windows:

        a. Open an elevated command prompt (run as administrator) and navigate to the directory to which you downloaded the EUM Server installer.
        b. Run the installer:

        ```
        euem-64bit-windows-4.5.x.x.exe
        ```

2. In the Welcome screen, click **Next**.
   The **License Agreement** page appears.
3. Scroll to the end of the license agreement, accept the license agreement and click **Next** to continue.
4. Select the directory in which you want to install the server and click **Next**.

5. Choose **Demo** for the installation mode. In this mode, the installer looks for a Controller on the current host and an Events Service on a separate host. It then installs the EUM Server on the same host as the Controller. Click **Next**.



Setup - AppDynamics End User Monitoring 4.4.0.0

AppDynamics End User Monitoring Installation Type

Select Installation Type

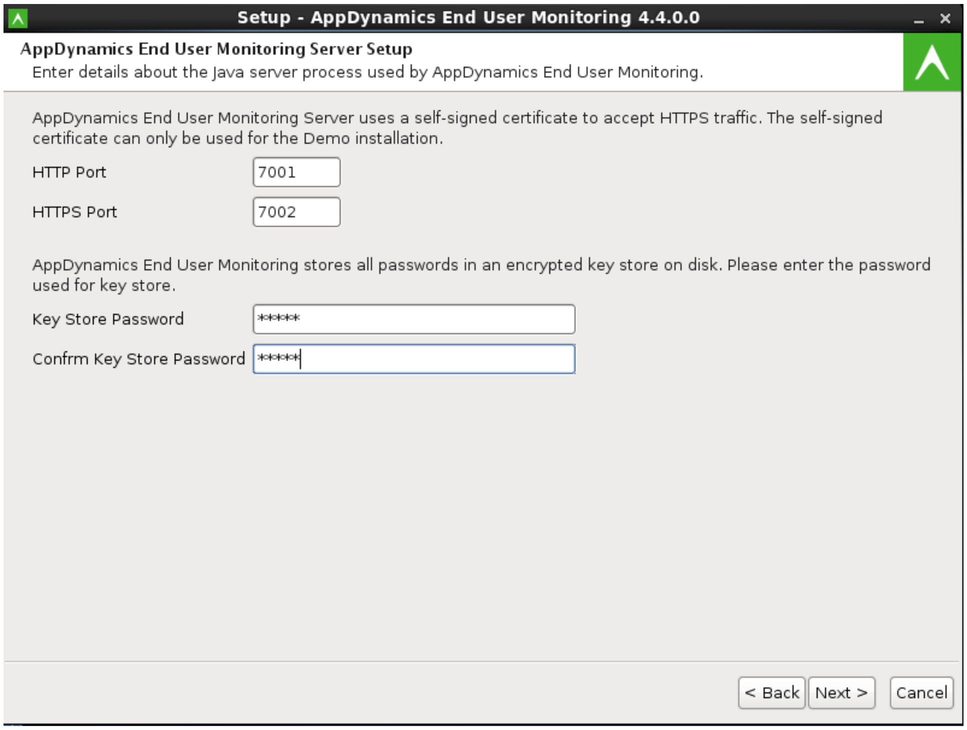
● Demo

Use this type if AppDynamics End User Monitoring is installed on the same host where AppDynamics Controller is installed. In this type, AppDynamics End User Monitoring will create a separate schema on the MySQL instance used by AppDynamics Controller.

○ Product

Use this type if AppDynamics End User Monitoring and AppDynamics Controller are installed on different hosts. This type will install a MySQL instance on this machine that will be used by AppDynamics End User Monitoring.

6. In the **Database Setup** dialog box:
   a. Enter a password in the **Root User Password** field.
   b. Enter and confirm a password for the **eum_user** database user account.
7. In the End User Monitoring Server Setup screen:
   a. Enter the HTTP or HTTPs listening ports on the EUM Server at which the Controller will connect to the EUM Server (the default HTTP port is 7001 and HTTPS is 7002).
   b. Enter a new password in the **Key Store Password** and confirm it.



Setup - AppDynamics End User Monitoring 4.4.0.0

AppDynamics End User Monitoring Server Setup

Enter details about the Java server process used by AppDynamics End User Monitoring.

AppDynamics End User Monitoring Server uses a self-signed certificate to accept HTTPS traffic. The self-signed certificate can only be used for the Demo installation.

HTTP Port: 7001

HTTPS Port: 7002

AppDynamics End User Monitoring stores all passwords in an encrypted key store on disk. Please enter the password used for key store.

Key Store Password: *****

Confrm Key Store Password: *****

Usernames and passwords can only consist of ASCII characters. In addition, passwords cannot include the characters '^', '/', or '$'.

Note that ports shown here are the location both to which the EUM Agents send their beacons and from which the Controller fetches the processed beacon data. Click **Next** and click **Finish.**

Post-installation Tasks

After installing the EUM server, you must perform three additional post-installation tasks:

- Provision the EUM license
- Configure the Events Services properties in the `eum.properties` file (optional)
- Connect the EUM server with the AppDynamics Controller

Provision the EUM License

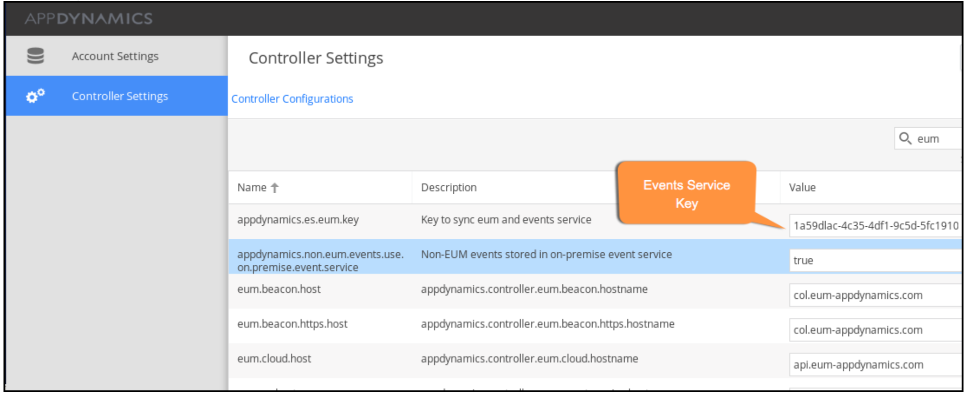
To provision the EUM license:

Configure the Events Service Properties

For the demo installation, Events Services configuration is not required for the EUM Server to start.

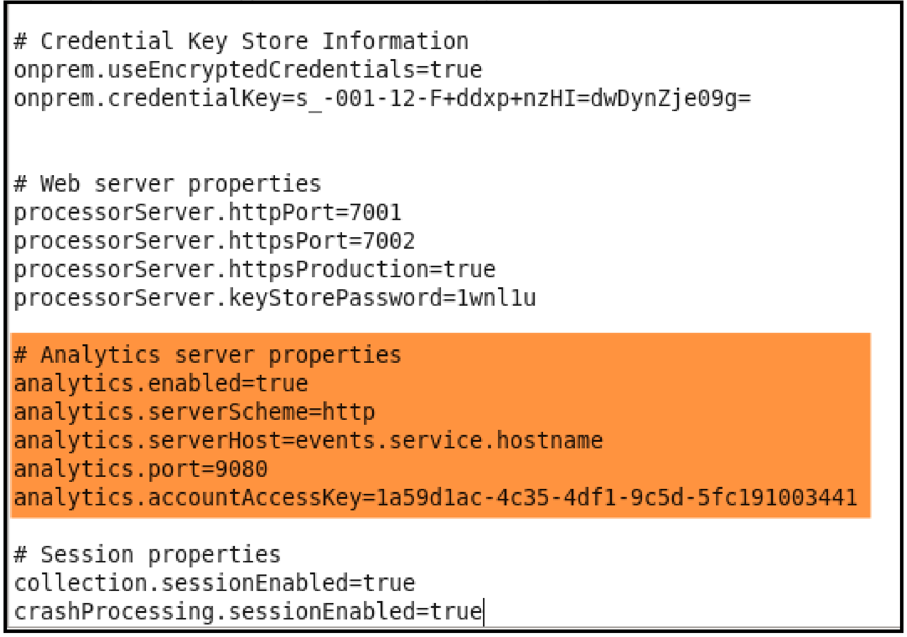
To configure the Events Service properties in the `eum.properties` file:

1. Navigate to and start your Controller.
2. Using the CLI, navigate to the `bin` directory.
3. Open the `eum.properties` file for editing.
4. In the `eum.properties` file, enter the following values:
   a. `analytics.enabled=true`
   b. `analytics.serverHost=<hostname>`, where `<hostname>` is the host where the event service is running
   c. `analytics.accountAccessKey=<eum_key>`, where `<eum_key>` is the Events Service key that appears as the `appdynamics.es.eum.key` value in the Administration Console:



Name	Description	Value
appdynamics.es.eum.key	Key to sync eum and events service	1a59dlac-4c35-4df1-9c5d-5fc1910
appdynamics.non.eum.events.use.on.premise.event.service	Non-EUM events stored in on-premise event service	true
eum.beacon.host	appdynamics.controller.eum.beacon.hostname	col.eum-appdynamics.com
eum.beacon.https.host	appdynamics.controller.eum.beacon.https.hostname	col.eum-appdynamics.com
eum.cloud.host	appdynamics.controller.eum.cloud.hostname	api.eum-appdynamics.com

The configuration should appear similar to the following example:



```
# Credential Key Store Information
onprem.useEncryptedCredentials=true
onprem.credentialKey=s_-001-12-F+ddxp+nzHI=dwDynZje09g=


# Web server properties
processorServer.httpPort=7001
processorServer.httpsPort=7002
processorServer.httpsProduction=true
processorServer.keyStorePassword=1wnl1u

# Analytics server properties
analytics.enabled=true
analytics.serverScheme=http
analytics.serverHost=events.service.hostname
analytics.port=9080
analytics.accountAccessKey=1a59d1ac-4c35-4df1-9c5d-5fc191003441

# Session properties
collection.sessionEnabled=true
crashProcessing.sessionEnabled=true
```

5. After updating the `eum.properties` file, restart the EUM Server.

Connect the EUM Server with the AppDynamics Controller

To connect the EUM server with the AppDynamics Controller:

1. Log in to the Administration Console.
2. Set the following Controller properties:
    a. `eum.cloud.host: http://localhost:7001` – This tells you where the Controller will poll for EUM metrics
    b. `eum.beacon.host: eum-host-name:7001` – This tells you where beacons are sent for page requests made with the HTTP protocol
    c. `eum.beacon.https.host: https://eum-host-name:7002` – This tells you where the beacons are sent for page requests made with the HTTPS protocol
    d. `eum.mobile.screenshot.host: eum-host-name:7001` – This tells you where the Controller will look for mobile screenshots

Installing with the Silent Installer

Instead of using the installer in GUI mode, you can use the silent installer to perform an unattended installation. The silent installer takes a response file as a source for the initial configuration settings. It's useful for scripting installation or performing large scale deployments.

To use a response file for a Demo installation:

1. Create a file named `response.varfile` on the machine on which you will run EUM installer with the following:

```
sys.adminRights$Boolean=false
sys.languageId=en
sys.installationDir=/AppDynamics/EUM
euem.InstallationMode=demo
euem.Host=controller
euem.initialHeapXms=1024
euem.maximumHeapXmx=4096
euem.httpPort=7001
euem.httpsPort=7002
mysql.databasePort=3388
mysql.databaseRootUser=root
mysql.dbHostName=localhost
mysql.dataDir=/usr/local/AppDynamics/EUM/data
mysql.rootUserPassword=singcontroller
mysql.rootUserPasswordReEnter=singcontroller
eumDatabasePassword=secret
eumDatabaseReEnterPassword=secret
keyStorePassword=secret
keyStorePasswordReEnter=secret
```

```
eventsService.isEnabled$Boolean=true
eventsService.serverScheme=http
eventsService.host=eventsservice_host
eventsService.port=9080
eventsService.APIKey=1a234567-1234-1234-4567-ab123456
```

2. Modify values of the installation parameters based on your own environment and requirements. Particularly ensure that the directory paths and passwords match your environment.
3. Run the installer with the following command:

   ```
   ./euem-64bit-linux-4.5.x.x.sh -q -varfile response.varfile
   ```

   On Windows, use:

   ```
   euem-64bit-windows-4.5.x.x.exe -q -varfile response.varfile
   ```

Provision EUM Licenses

Related pages:

- Install a Production EUM Server
- Troubleshoot EUM Server Installation
- Controller Deployment
- Multi-Tenant Controller Accounts

This page describes how to provision EUM licenses for single-tenant and multi-tenant Controllers.

Provision the EUM License for a Single-Tenant Controller

To provision the EUM license for a single-tenant Controller:

Provision the EUM License for Multi-Tenant Controllers

For each on-prem multi-tenant Controller account wanting EUM access, you are required to provision an EUM license. Provisioning EUM license units for different Controller accounts enable you to better track, manage, and limit license usage.

To enable the EUM Server to work with on-prem multi-tenant Controllers:

1. Complete the setup requirements.
2. Request EUM licenses for the Controller accounts requiring EUM access.
3. Provision EUM licenses for the Controller accounts.

Setup Requirements

- Deploy an on-premises AppDynamics Controller
- Set up multi-tenancy on the Controller

Request EUM Licenses

For each additional Controller account requiring EUM access, you will need to request EUM licenses from the AppDynamics Sales team. The AppDynamics Sales team can help you determine how many licenses and units per license will meet your needs.

Provision Licenses for Controller Accounts

To provision EUM licenses for each Controller account:

1. Log in to your EUM Server.
2. Change to the directory with the script for provisioning licenses:

3. Provision each license, one at a time, on the EUM Server by running the following command:

4. Log in as the administrator to your **Controller Admin Console** through `http://<hostname>:<port>/controller/admin.jsp`.
5. From **Account Settings**, select the Controller account that have EUM licenses and click **Edit**.
6. From the Controller account page:
   a. Enter the EUM license key and the EUM account name in the **EUM License Key** and the **EUM Account Name** fields.
   b. Click **Save**.
7. Repeat steps 5 and 6 for the other Controller accounts that have EUM licenses.
8. Verify that the EUM licenses are working and available in the Controller UI. There will be an error message if the new EUM Server is not connected properly.

Secure the EUM Server

If you use HTTPS connections in a production (split host) EUM Server installation, use a custom RSA security certificate for the EUM server. This page describes how to create an RSA security certificate, change the password for the credential keystore, and how to obfuscate a password for the security certificate keystore.

Set Up a Custom Keystore for Production

In demo mode, the EUM Server uses a default self-signed certificate named `ssugg.keystore`. This certificate is intended for demonstration and light testing only. Do not use self-signed certificates for production systems since they are less secure than Certificate Authority (CA) signed certificates. EUM requires that certificates use RSA as the key algorithm whether they are self-signed or CA-signed.

For Mobile Real User Monitoring, if you use the default or another self-signed certificate on your EUM Server for testing, you may receive the following error: "The certificate for this server is invalid". Ensure that your self-signed certificate is trusted by the simulator or device you use for testing. In real-world scenarios, a CA signed certificate should be used since a self-signed certificate needs to be explicitly trusted by every device that reports to your EUM processor.

To secure the EUM server with a custom certificate and keystore, generate a new JKS keystore and configure the EUM Server to use it.

The following instructions describe how to create a JKS keystore for the EUM Server with a new key-pair or an existing key-pair. Alternatively, you can also configure the EUM server to use an existing JKS keystore.

The instructions demonstrate the steps with the Linux command line, but the commands are similar to the commands used for Windows. Make sure to adjust the paths for your operating system.

Overview of the Steps

The procedure is made up of three parts:

1. Create a new certificate and keystore (1a) or import an existing certificate into a keystore (1b).
2. Configure the EUM Server to use the keystore.
3. Restart and test the new keystore.

Step 1a: Create a New Certificate and Keystore

1. At a command prompt, navigate to the `eum-processor` directory:

   ```
   cd <appdynamics_home>/EUM/eum-processor
   ```

2. Create a new keystore with a new unique key pair that uses RSA encryption:

   ```
   ../jre/bin/keytool -genkey -keyalg RSA -validity <validity_in_days> -alias 'eum-processor' -keystore bin/mycustom.keystore
   ```

   This creates a new public-private key pair with an alias of `'eum-processor'`. You can use any value you like for the alias.

   > The "first and last name" required during the installation process becomes the common name (CN) of the certificate. Use the name of the server.

3. Configure the keystore.
4. Specify a password for the keystore. You need to configure this password in the EUM configuration file later.
5. Generate a certificate signing request (CSR):

   ```
   ../jre/bin/keytool -certreq -keystore bin/mycustom.keystore -file /tmp/eum.csr -alias 'eum-processor'
   ```

   This generates a certificate signing request based on the contents of the alias, in the example `'eum-processor'`. You should send the output file (`/tmp/eum.csr`, in the example) to a Certificate Authority for signing. After you receive the signed certificate, proceed as follows.
6. Install the certificate for the Certificate Authority used to sign the `.csr` file:

```
../jre/bin/keytool -import -trustcacerts -alias myorg-rootca -keystore bin/mycustom.keystore -file /path/to/CA-cert.txt
```

This command imports your CA's root certificate into the keystore and stores it in an alias called `myorg-rootca`.

7. Install the signed server certificate as follows:

   ```
   ../jre/bin/keytool -import -keystore bin/mycustom.keystore -file /path/to/signed-cert.txt  -alias 'eum-processor'
   ```

   This command imports your signed certificate over the top of the self-signed certificate in the existing alias, in the example, `'eum-processor'`.
8. Import the root certificate from step 6 to the Controller truststore:

   ```
   keytool -import -trustcacerts -alias <alias_name> -file mycert.cer -keystore <complete_path_to_cacerts.jks>
   ```

Step 1b: Import an Existing Certificate into a JKS Keystore

If you have an existing public-private key pair that uses RSA, you must import them into a JKS keystore to use it for EUM.

1. At a command prompt, navigate to the `eum-processor` directory:

   ```
   cd <appdynamics_home>/EUM/eum-processor
   ```

2. Stop the EUM process.

   Run the following command:

   ```
   bin/eum.sh stop
   ```

3. If there is an existing custom JKS keystore, back it up:

   ```
   mv <keystore>.jks <keystore>.jks.old
   ```

4. Import the private and public key for your certificate into a PKCS12 keystore:

   ```
   openssl pkcs12 -inkey <private_key_file> -in <certificate_file> -export -out keystore.p12
   ```

5. Convert the PKCS12 keystore to JKS format:

   ```
   keytool -importkeystore -srckeystore keystore.p12 -srcstoretype pkcs12 -destkeystore <JKS_keystore> -deststoretype JKS
   ```

   This command creates a JKS keystore with the name specified in the `-destkeystore` property.
6. Specify a password for the keystore. Use this password when you configure EUM to use the new keystore.

Step 2: Configure the EUM Server to Use the New Keystore

1. Place the new keystore file in the following directory: `<appdynamics_home>/EUM/eum-processor/bin`.
2. Edit the `eum.properties` file in the bin directory.
3. If the property `processorServer.keyStorePassword` is set, remove or uncomment it.
4. Add the keystore filename as the following property:

   ```
   processorServer.keyStoreFileName=mycustom.keystore
   ```

5. Configure the password for the keystore. You can add the password to the file either in plain text or in the obfuscated form:
   - For a plain text password, add the password as the value for this property:

```
processorServer.keyStorePassword=mypassword
```

- For an obfuscated password:
    a. Get the obfuscated password by running the following command in the `eum-processor` directory in a new command terminal:

    ```
    bin/eum-credential-key.<bat|sh> obfuscate -plaintext <newpassword>
    ```

    b. Copy the output of the command to your clipboard.
    c. In `eum.properties`, paste the obfuscated password as the value of the `keyStorePassword` property:

    ```
    processorServer.keyStorePassword=<obfuscated_key>
    ```

    d. Add the `useObfuscatedKeyStorePassword` with the value set to true, as shown:

    ```
    processorServer.useObfuscatedKeyStorePassword=true
    ```

6. Save and close the file.

Step 3: Restart and Test

1. Restart the EUM Server. From the `eum-processor` directory, run the following commands:

```
bin/eum.sh stop
bin/eum.sh start
```

2. Verify the new security certificate works by opening the following page in a browser:

```
https://<hostname>:7002/eumcollector/get-version
```

If you get a successful response, the configuration succeeded.

Change the Certificate Keystore Password

The previous steps describe how to create a new keystore which is likely to have a new password. To change the keystore password without creating a new keystore, perform the following steps:

1. At a command prompt, navigate to the `eum-processor` directory:

```
cd <appdynamics_home>/EUM/eum-processor
```

2. Run the `keytool` command for creating a new password:

```
../jre/bin/keytool -storepasswd -keystore bin/ssugg.keystore
```

The sample command creates the password for the default demo keystore, `ssugg.keystore`. In your command, use the name of your own keystore as the value for `-keystore`.
3. Enter the existing password and new password when prompted.
4. Get the obfuscated key by running the following command in the `eum-proccessor` directory:

```
bin/eum-credential-key.<bat|sh> obfuscate -plaintext <newpassword>
```

5. Copy the output of the previous command to your clipboard.
6. In the `eum.properties` file in the `eum-processor/bin` directory, paste the obfuscated password as the value for the `keyStorePassword` property:

```
processorServer.keyStorePassword=<obfuscated_key>
```

7. If you did not previously use an obfuscated password, add the following property:

   ```
   processorServer.useObfuscatedKeyStorePassword=true
   ```

8. Save and close the file.
9. Restart the EUM Server.

Change the Credential Keystore Password for the EUM Database

When you install the EUM Server, you need to specify a password to use to secure the credential keystore for the EUM Server. After installation, you can change the password for the credential keystore. You may need to do this, for example, to comply with your organization's password rotation policy.

Note that completing these procedures requires a restart of the EUM Server.

To change the existing EUM server credential keystore password:

1. At a command prompt, navigate to the `eum-processor` directory:

   ```
   cd <appdynamics_home>/EUM/eum-processor
   ```

2. Generate a credential store with the new key using the following command:
   - On Linux:

     ```
     bin/eum-credential-key.sh generate_ks -storepass <new_password>
     ```

   - On Windows:

     ```
     bin\eum-credential-key.bat generate_ks -storepass <new_password>
     ```

   This creates and initializes a new credential file, `bin/credential.scs`.
3. Reencrypt the database password using the new credential store.
   - On Linux:

     ```
     bin/eum-credential-key.sh encrypt -storepass <new_password> -plaintext <DB_password>
     ```

   - On Windows:

     ```
     bin\eum-credential-key.bat encrypt -storepass <new_password> -plaintext <DB_password>
     ```

   The command prints out the encrypted form of the `DB_password` value you entered.
4. Copy the output from the previous command to your clipboard.
5. Open `bin/eum.properties` for editing, and replace the value of the `onprem.dbPassword` setting with the new encrypted password you copied to your clipboard.
6. Obfuscate the new credential key as follows:
   - On Linux:

     ```
     bin/eum-credential-key.sh obfuscate -plaintext <new_password>
     ```

   - On Window:

     ```
     bin\eum-credential-key.bat obfuscate -plaintext <new_password>
     ```

7. Copy the output of the previous command to your clipboard and in `eum.properties` replace the value of `onprem.credentialKey` with the value from your clipboard.

8. Save and close the properties file.
9. Restart the EUM server.

Change the EUM Database Password

At EUM Server installation time, you set a password for the EUM database. You can change it later as follows:

1. At a command prompt, navigate to the `eum-processor` directory:

   ```
   cd <appdynamics_home>/EUM/eum-processor
   ```

2. Encrypt the new database password using the credential key which you entered during installation:
   - On Linux:

     ```
     bin/eum-credential-key.sh encrypt -storepass <plain_credential_key> -plaintext <New_DB_password>
     ```

   - On Windows:

     ```
     bin\eum-credential-key.bat encrypt -storepass <plain_credential_key> -plaintext <New_DB_password>
     ```

     The command prints out the encrypted form of the `DB_password` value you entered.
3. Copy the output from the previous command to your clipboard.
4. Edit `bin/eum.properties` and replace the value of the `onprem.dbPassword` setting with the new encrypted password you copied to your clipboard.
5. Save and close the properties file.
6. Restart the EUM server.

Configure the EUM Server

This page describes administration and advanced configuration options for the EUM Server.

Configure Data Store Expiration

As part of the Analytics functionality used by EUM, the Server stores some data, like crash reports and resource snapshots, in a local blob store. The default setting of 30 days, but you can change the storage period to be longer or shorter by following these steps:

1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.sample.properties` with a text editor.
3. Copy the `onprem.crashReportExpirationDays` property and the `onprem.resourceSnapshotExpirationDays` property from the sample file into `eum.properties` and set it to whatever value you wish. The unit is *days*.
4. Restart the Server.

Set the Maximum Length of Page URLs Read From Beacons

By default, after the EUM Collector receives beacons, the EUM Processor will only read 512 characters of page URLs contained in the beacon. If the page URL exceeds 512 characters, the EUM Processor will truncate the page URL. You can configure the EUM Processor to read a longer page URL with the configuration `beaconReader.maxUrlLength`. The maximum length that can be set is 2048, which is imposed by the JavaScript Agent creating the beacon.

To change the maximum length of the page URL read by the EUM Processor:

1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Add the property `beaconReader.maxUrlLength` to the desired length (maximum is 2048):

   ```
   beaconReader.maxUrlLength=<max_length>
   ```

3. Restart the Server.

Update the EUM Server's Geo Server

The on-prem EUM Server ships with Neustar's IP GeoPoint database for managing the geolocation of IP addresses. You can get daily updates of the Neustar IP GeoPoint from the AppDynamics download site.

To keep your version of the database current, you need to update your copy of the database manually:

Configure the Port for the EUM Agent

The on-prem EUM Server by default uses the same port to collect data from the EUM agent and to send data through the API server to the Controller. You can configure the EUM Server to use a different port to collect data from the EUM agent by following the instructions below.

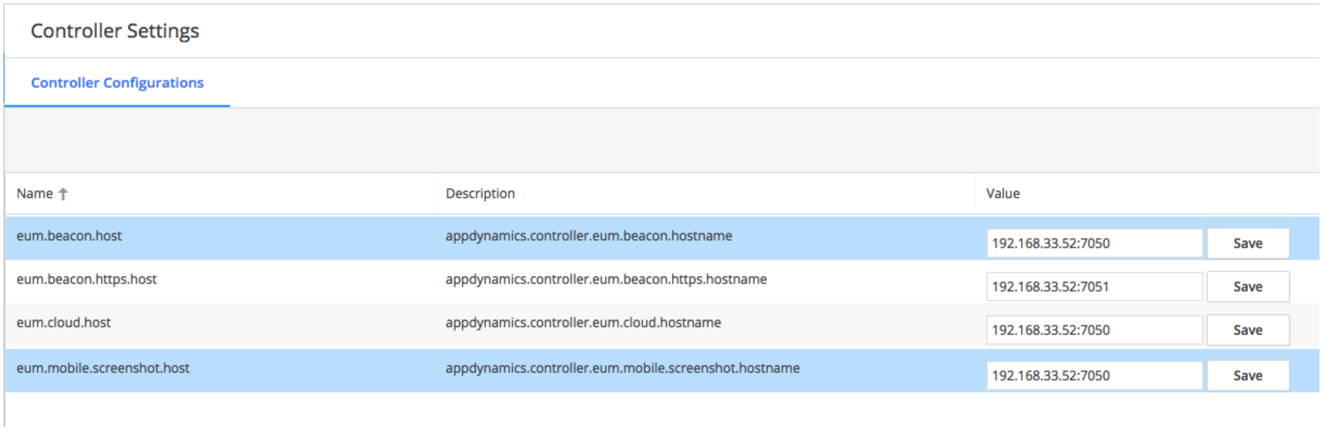
1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Add the following lines to `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties`, replacing `<PORT>` with the port you want the EUM server to listen to.

   ```
   processorServer.collectorHttpPort=<PORT>
   processorServer.collectorHttpsPort=<PORT>
   ```

3. Restart the EUM Server.

4. From the **Controller Admin UI**, change the ports for the properties `eum.beacon.host`, `eum.beacon.https.host`, `eum.cloud.host`, and `eum.mobile.screenshot.host` so that they are the same as those assigned to `processorServer.collectorHttpPort` and `processorServer.collectorHttpsPort`. This allows the beacon to communicate with the collector.

   For example, if you set `processorServer.collectorHttpPort=7050` and `processServer.collectorHttpsPort=7051`, you would then set the ports for the properties `eum.beacon.host` and `eum.mobile.screenshot.host` to `7050` for HTTP and `eum.beacon.https.host` to `7051` for HTTPS as shown below:



Controller Settings

Name ↑	Description	Value	
eum.beacon.host	appdynamics.controller.eum.beacon.hostname	192.168.33.52:7050	Save
eum.beacon.https.host	appdynamics.controller.eum.beacon.https.hostname	192.168.33.52:7051	Save
eum.cloud.host	appdynamics.controller.eum.cloud.hostname	192.168.33.52:7050	Save
eum.mobile.screenshot.host	appdynamics.controller.eum.mobile.screenshot.hostname	192.168.33.52:7050	Save

Limit the Number of EUM Snapshots

When an application has a high number of Ajax requests per page, the EUM Server retains a large number of snapshots that can include base, virtual, and Ajax pages as well as iFrames. You can limit the number of snapshots retained by the EUM server by setting a global maximum, reducing the time that they are retained, or by filtering snapshots based on the network response time.

Setting the Global Limit for Snapshots

You set the global limit on the number of snapshots to be retained per minute with the configuration `browserBeaconSampling.maxSamples`. The default value is `1000`. Once the limit is reached, all snapshots will be dropped indiscriminately. The limit can be globally configured through the `eum.properties` file.

1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Add the following line to `eum.properties`, replacing `<global_limit>` with the global maximum number of snapshots to retain.

```
browserBeaconSampling.maxSamples = <global_limit>
```

3. Restart the EUM Server.

Reduce the Lifespan of Event Snapshots

Another way to limit the number of EUM snapshots is to reduce the number of days that the event snapshots are retained. Event snapshots only apply to the crash reports, code issues, and IoT errors and are stored in the local blob store: `$APPDYNAMICS_HOME/EUM/eum-processor/store`

By default, the EUM Server retains the event snapshots for 90 days. If your Events Service retains events for fewer days (e.g., 14 days), you can safely change the EUM Server's retention period to be the same as the Events Service's retention period. If the EUM Server retains the event snapshots for fewer days than the Events Service, however, you may run into errors when viewing older events in the Controller UI.

> ⓘ When reducing the lifespan of event snapshots, you are not modifying the retention period of the Controller or the Events Service.

Setting the Lifespan for the Event Snapshots

1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Add the following line to `eum.properties,` replacing `<no_of_days>` with the number of days that you'd like to retain the event snapshots. The default is 90.

```
eventSnapshotStore.lifespanInDays = <no_of_days>
```

3. Restart the EUM Server.

Filtering Snapshots Based on the Network Response Time

You set a threshold that filters the snapshots based on the network response time. If the network response time is at or below the configured threshold, the snapshot is then retained. You set the threshold with the configuration `browserBeaconSampling.hierarchyAwareSamplerPageUXThreshold`.

Below are the supported threshold values and the snapshots that would be retained. The default value is `Slow`.

- `Normal` - Using this threshold value will retain all snapshots.
- `Slow` - Using this threshold value will retain snapshots having a network response time of slow, very slow, and stalled.

Setting the Threshold for the Network Response Time

1. Open `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties` with a text editor.
2. Add the following line to `eum.properties`, replacing `<threshold>` with one of the supported thresholds (`Normal` or `Slow`) for retaining snapshots.

```
browserBeaconSampling.hierarchyAwareSamplerPageUXThreshold = "<threshold>"
```

3. Restart the EUM Server.

Turn On Access Logs

By default, server access logging for the EUM Server's underlying application server is turned off. To turn it on, open `$APPDYNAMICS_HOME/EUM/eum-processor/conf/local-eum-processor.yml` with a text editor and find the following section under the `server` entry:

```
 requestLog:
        appenders: []
```

Add the following information:

```
requestLog:
        timeZone: UTC
        appenders:
          - type: file
            archive: true
            currentLogFilename: ../logs/access.log
            archivedLogFilenamePattern: ../log/accedd-%d.log.gz
```

Save the file and restart the EUM Server.

EUM Server Configuration File

You can configure the EUM Server by setting properties in the file `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties`. You are recommended to copy the reference sample file `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.sample.properties` to `$APPDYNAMICS_HOME/EUM/eum-processor/bin/eum.properties`, modify the settings to fit your needs, and then restart the EUM Server so that the new settings are applied.

The table below lists and describes the supported EUM properties, lists defaults, and specifies whether the property is required. The values for the database properties must conform with the MySQL syntax rules given in Schema Object Names.

EUM Property	Default	Required?	Description
`onprem.dbHost`	`dbHost`	Yes	The name of the database host.
`onprem.dbPort`	3388	Yes	The port to the database host.
`onprem.dbSchema`	`eum_db`	Yes	The name of the EUM database.
`onprem.dbUser`	`eum_user`	Yes	The user name for the EUM database.

`onprem.dbPassword`	N/A	Yes	The user password to the EUM database. The password can consist of any ASCII character except the characters '^', '/', or '$'.
`onprem.eventSnapshotDiskAllowance`	-1	No	The maximum disk space allotted for storing event snapshots. The default value of -1 allots unlimited disk space to store event snapshots. You can specify a positive integer representing the maximum number of bytes for storing event snapshots.
`onprem.fileStoreRoot`	`../store`	No	The path to the directory storing EUM data such as snapshots.
`onprem.crashReportExpirationDays`	365	No	The number of days that crash reports are retained.
`onprem.resourceSnapshotExpirationDays`	15	No	The number of days that resource snapshots are retained.
`onprem.resourceSnapshotDiskAllowance`	21474836480 (20 GB)	No	The maximum disk space allotted for storing resource snapshots. The default maximum disk space is 20 GB or 21474836480 bytes. You can specify a positive integer representing the maximum number of bytes for storing resource snapshots.
`processorServer.httpPort`	`7001`	No	The HTTP port to the EUM Processor. The EUM Processor runs in one process containing the collector, aggregator, crash-processor, and monitor services.
`processorServer.httpsPort`	`7002`	No	The HTTPS port to the EUM Processor.
`processorServer.httpsProduction`	`true`	No	The flag for turning enabling (true) or disabling HTTPS to the EUM Processor.
`processorServer.keyStorePassword`	N/A	No	The password to the Key Store for the EUM Processor.
`processorServer.keyStoreFileName`	`bin/ssugg.keystore`	No	The path to the file that stores the password to the Key Store for the EUM Processor.
`processorServer.collectorHttpPort`	`7001`	No	The HTTP port of the EUM Collector. By default, the EUM Collector shares the same port as the EUM Processor, but you can configure the port to be different. The EUM Collector receives the metrics sent from the JavaScript agent.
`processorServer.collectorHttpsPort`	`7002`	No	The HTTPS port of the EUM Collector.
`analytics.enabled`	`true`	Yes	The flag for enabling or disabling the Analytics Server.
`analytics.serverScheme`	`http`	No	The network protocol for connecting to the Analytics Server. It is only required with `analytics.enabled=true`.
`analytics.serverHost`	`events.service.hostname`	No	The hostname of the Analytics Server. It is only required with `analytics.enabled=true`.
`analytics.port`	`9080`	No	The port to the Analytics Server. It is only required when `analytics.enabled=true`.
`analytics.accountAccessKey`	`access-key`	No	The access key for connecting to the Analytics Server. It is only required with `analytics.enabled=true`.
`analytics.eventTypeLifeSpan.0.eventType`	`BrowserRecord`	No	The type of event to be saved. The following values are supported: • `BrowserRecord` • `MobileSnapshot` • `SessionRecord` • `MobileSessionRecord` If this property is set, you must also set `analytics.eventTypeLifeSpan.0.lifeSpan`.
`analytics.eventTypeLifeSpan.0.lifeSpan`	8	No	The number of days to retain the event records specified by `analytics.eventTypeLifeSpan.0.eventType`. If this property is set, you must also set `analytics.eventTypeLifeSpan.0.eventType`.
`analytics.eventTypeLifeSpan.1.eventType`	`MobileSnapshot`	No	The type of event to be saved. The following values are supported: • `BrowserRecord` • `MobileSnapshot` • `SessionRecord` • `MobileSessionRecord` If this property is set, you must also set `analytics.eventTypeLifeSpan.1.lifeSpan`.

`analytics.eventTypeLifeSpan.1.lifeSpan`	8	No	The number of days to retain the event records specified by `analytics.eventTypeLifeSpan.1.eventType`. If this property is set, you must also set `analytics.eventTypeLifeSpan.1.eventType`.
`analytics.eventTypeLifeSpan.2.eventType`	`SessionRecord`	No	The type of event to be saved. The following values are supported: - `BrowserRecord` - `MobileSnapshot` - `SessionRecord` - `MobileSessionRecord` If this property is set, you also must set `analytics.eventTypeLifeSpan.2.lifeSpan`.
`analytics.eventTypeLifeSpan.2.lifeSpan`	8	No	The number of days to retain the event records specified by `analytics.eventTypeLifeSpan.2.eventType`. If this property is set, you must also set `analytics.eventTypeLifeSpan.2.eventType`.
`analytics.eventTypeLifeSpan.3.eventType`	`MobileSessionRecord`	No	The type of event to be saved. The following values are supported: - `BrowserRecord` - `MobileSnapshot` - `SessionRecord` - `MobileSessionRecord` If this property is set, you also must set `analytics.eventTypeLifeSpan.3.lifeSpan`.
`analytics.eventTypeLifeSpan.3.lifeSpan`	8		The number of days to retain the event records specified by `analytics.eventTypeLifeSpan.3.eventType`. If this property is set, you must also set `analytics.eventTypeLifeSpan.3.eventType`.
`onprem.mobileAppBuildTimeSeriesRequestCountRollupDays`	7	No	The EUM Collector searches for the dSYM file in the beacon traffic for the configured number of days. If the dSYM file is not present during the configured time frame, a warning message is displayed in the Controller UI.
`onprem.maxNumberOfMobileBuildsWithoutDsym`	10	No	The maximum number of visible missing dSYM files in the Controller UI.
`collection.sessionEnabled`	`true`	No	The flag for enabling or disabling browser/mobile session collection. If you are upgrading the EUM Server from versions 4.2 and lower to 4.3 or higher, the default is `false`.
`collection.accessControlAllowOrigins.{n}`	*	No	By default, the EUM Collector responds with `Access-Control-Allow-Origin: *`. You can limit CORS to certain domains by using an integer property assigned to a URL as in the following: - `collection.accessControlAllowOrigins.0=http://example1.com` - `collection.accessControlAllowOrigins.1=http://example2.com` - `collection.accessControlAllowOrigins.2=http://example3.com`
`eventSnapshotStore.lifespanInDays`	90	No	The number of days that the event snapshots stored in the EUM Server's local blob store (`$APPDYNAMICS_HOME/EUM/eum-processor/store`) are retained. The event snapshots only apply to crash reports, code issues, and IoT errors.
`sessionization.webSessionRetentionMins`	5	No	The number of minutes that browser sessions are retained after they are closed. This allows browser sessions that begin and end at different times to be retained. The longer the configured retention time, the larger the number of closed sessions held in memory, resulting in higher memory usage.
`sessionization.mobileSessionRetentionMins`	5	No	The number of minutes that mobile sessions are retained after they are closed. This enables mobile sessions that begin and end at different times to be retained. The longer the configured retention time, the larger the number of closed sessions held in memory, resulting in higher memory usage.
`throttling.resourceSnapshot.maxTotalPerMinPerAccount`	1000	No	The maximum number of total resource snapshots retained each minute for an account.
`throttling.resourceSnapshot.maxNormalPerMinPerAccount`	800	No	The maximum number of resource snapshots of pages with a "Normal" user experience that are retained each minute for an account. In general, you want the number for this property to be smaller than that for `throttling.resourceSnapshot.maxTotalPerMinPerAccount`, so you can also retain resource snapshots of pages with a "Slow", "Very Slow", or "Stall" user experience.

`throttling.session.maxTrackedSessionsPerAccount`	50000	No	The maximum number of active sessions and unexpired closed sessions that are stored in memory for an account. When the maximum is reached, events that create new sessions will be dropped. This setting helps to control the memory used for sessions at the account level.

> From EUM Server version 4.5.1 and later, the property `crashProcessing.sessionEnabled` is no longer supported. Instead, the association of crashes with sessions is enabled by default. If you are using an earlier version (<4.5.1) of the EUM Server and want to upgrade to 4.5.1 or higher, you will need to remove the property `crashProcessing.sessionEnabled` from the `eum.properties` file to prevent the EUM Server from throwing errors.

EUM Server Endpoints

Related pages:

- Port Settings
- Synthetic Server Endpoints

The EUM Server has different endpoints serving distinct functions. This page provides a reference for testing the health and getting information about on-prem EUM Servers.

The endpoints include the following:

- **EUM API** - acts as the interface between the EUM Server and the Controller. The Controller retrieves EUM data from the EUM Server through the EUM API endpoint.
- **EUM Collector** - collects metrics from the EUM agents. The JavaScript Agent and Mobile Agents transmit data to the EUM Server through the EUM Collector endpoint.
- **EUM Aggregator** - collects and rolls up all the metrics per application and provide an interface for Controllers to download the metrics by application and timestamp.
- **Screenshot Service** - collects and serves image tiles that form mobile screenshots. The Mobile Agents transmit the tiles to the Screenshot Service, and the Controller retrieves the tiles to display the screenshots in mobile sessions.

EUM Server Endpoint URLs

The table below lists the endpoints, the default URL, and the supported paths.

EUM Server Endpoint	**Default URL**	**Paths / Description**	
EUM Collector	`http(s)://<domain-name>:7001/eumcollector`	`/adrum.gif`	Receives image beacons from the JavaScript Agent.
		`/beacons/browser`	Earliest endpoint for receiving CORS beacons from the JavaScript Agent.
		`/beacons/browser/v1/*`	The V1 endpoint for receiving CORS beacons from the JavaScript Agent.
		`/beacons/browser/v2/*`	The V2 and latest endpoint for receiving CORS beacons from the JavaScript Agent.
		`/get-version`	Returns the version, build, and commit, and timestamp of the EUM Processor.
		`/iot/v1/application/*`	The endpoint for IoT REST APIs. See the IoT REST API reference documentation for details.
		`/ping`	Returns whether the EUM Collector is accessible and running.
		`/mobileMetrics?version=2`	The endpoint used by the Mobile Agents to send mobile beacons via HTTP POST.
		`/whoami`	Returns the IP address, geo location, and information of the client making the request.
EUM Aggregator	`http(s)://<domain-name>:7001/eumaggregator`	`/currentTime`	Returns the current time of the EUM Aggregator as a Unix timestamp.
		`/get-version`	Returns the version, build, and commit, and timestamp of the EUM Processor.
		`/ping`	Returns whether the EUM Aggregator is accessible and running.
Screenshot Service	`http(s)://<domain-name>:7001/screenshots/v1`	`/version`	Returns the version, build, commit, and timestamp of the Screenshot Service.

Install and Host a Custom Geo Server for Browser RUM

Related pages:

- Host a Geo Server

By default, the locations of end-users are resolved using public geographic databases. You can host an alternate geo server for your countries, regions, and cities instead of using the default geo server hosted by AppDynamics.

You may prefer to host your own geo server because:

- You have intranet applications where the public IP address does not provide meaningful location information, but the user's private IP does.
- You have a hybrid application where some users access the application from a private location and some access it from a public one. If a user doesn't come from a specific private IP range mapped by the custom geo server, the system can be set to default to the public geo server.

To host a custom geo server:

1. Download the Geo Server File
2. Set the Location of the Geo Server
3. Create the IP Mapping File

Requirements for Geo Server Host

- 2 GB of memory
- Java 8.x

Download and Install the Geo Server File

Download the `GeoServer.zip` file from AppDynamics at https://download.appdynamics.com/download.

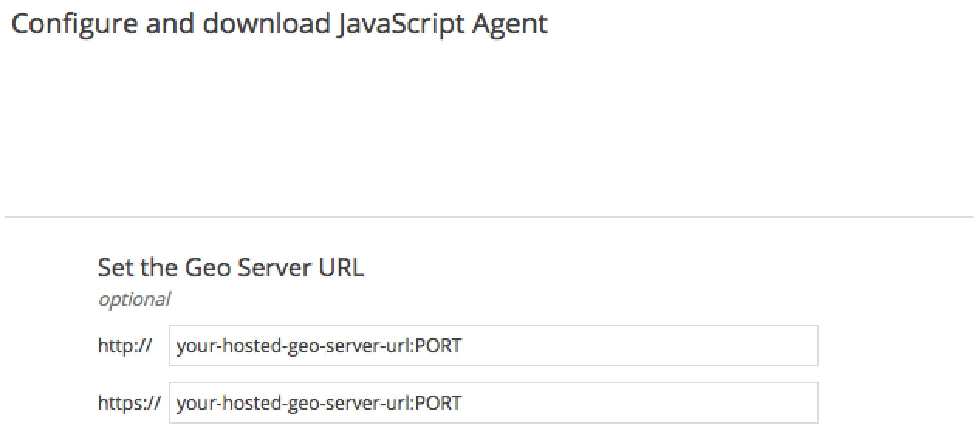
Uncompress the zip to a `GeoServer` folder with the following structure:

```
GeoServer
  schema.xsd                      <-- schema for geo-ip-mapping.xml configuration
  geo
     WEB-INF
        classes
           logback.xml       <-- configure logging in here
           ...
        web.xml               <-- other configurations here
        ...
 |    geo-ip-mappings.xml       <-- configure geo ip mapping here
  ...
```

To install the geo server, copy the `geo` folder to the `TOMCAT_HOME/webapps` of your Tomcat server. Do not deploy the server in the same container as the Controller.

Set the Location of the Geo Server

Enter the URL, including the context root, of your hosted geo server in the **Geo Server URL** field in the **Browser RUM** configuration screen in the Controller UI as shown below.



Configure and download JavaScript Agent

Set the Geo Server URL

optional

http:// your-hosted-geo-server-url:PORT

https:// your-hosted-geo-server-url:PORT

⚠ If you are using manual injection for your JavaScript agent, you must make sure that the copy of the script that you use is one that you have downloaded *after* this URL is set.

Create the IP Mapping File

The `geo-ip-mappings.xml` IP mapping file specifies the locations for which Browser RUM provides geographic data. It maps IP addresses to geographic locations.

Use the sample file in the `geo` subdirectory as a template. Any modifications at runtime are reloaded without a restart.

This file contains a `<mapping>` element for every location to be monitored. The file has the following format.

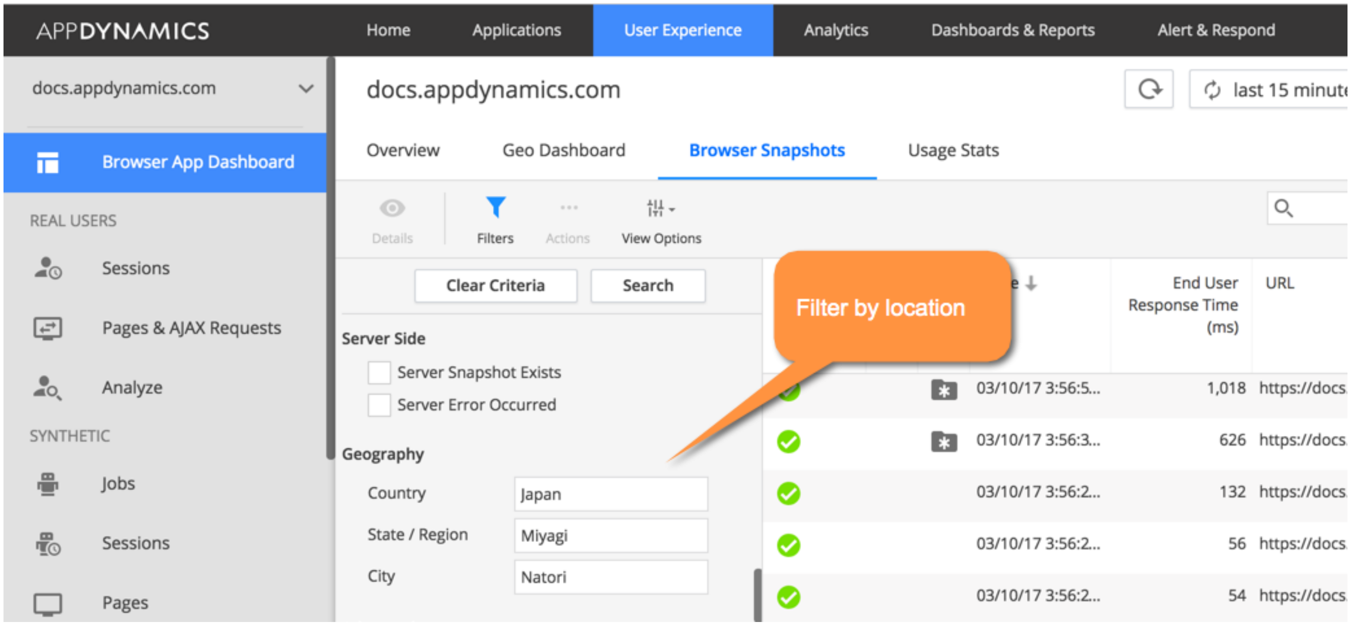
```
<mappings>
        <mapping>
                <subnet from="192.168.1.1" mask="255.255.255.0"/>
                <location country="United States" region="California" city="San Francisco"/>
        </mapping>

        <default country="United States" region="California" city="San Francisco"/>
</mappings>
```

You can also use IP-range-based mapping instead of subnet-based:

```
<mapping>
    <ip-range from="10.240.1.1" to="10.240.1.254" />
    <location country="France" region="Nord-Pas-de-Calais" city="ENGLOS" />
</mapping>
```

This data is visible in browser snapshots and can be used to filter browser snapshots for specific locations: The `<country>`, `<region>`, and `<city>` elements are required. If the values of <country> and <region> do not correspond to an actual geographic location already defined in the geographic database, map support is not available for the location in the map panel, but Browser RUM metrics are displayed for the location in the grid view of the geographic distribution, end user response time panel, trend graphs, browser distribution panel, and in the Metric Browser. The `<city>` element can be a string that represents the static location of the end-user. You will notice a `<default>` element. If there is an IP address that is not covered by your IP mapping file, this is the value that is used. To use a public geo server for non-covered IP addresses, see Using a Hybrid Custom-Public Geo Server Setup.



Filter by location

□ Server Snapshot Exists
□ Server Error Occurred

Country: Japan
State / Region: Miyagi
City: Natori

		e ↓	End User Response Time (ms)	URL
✓	✱	03/10/17 3:56:5...	1,018	https://docs
✓	✱	03/10/17 3:56:3...	626	https://docs
✓		03/10/17 3:56:2...	132	https://docs
✓		03/10/17 3:56:2...	56	https://docs
✓		03/10/17 3:56:2...	54	https://docs

The valid names for country and region are those used in the map in the geo dashboard. You can hover over a region in the dashboard to see the exact name (including spelling and case) of the region. See The Browser Geo Dashboard View.

Using a Hybrid Custom-Public Geo Server Setup

If you want Browser RUM to evaluate any non-mapped IP address using the public geo server, remove the `<default>` element. In this case, locating any non-mapped IP address is done in the EUM cloud, not locally.

Customize File Locations

You can customize where certain files are stored in the GeoServer directory.

Change Log Location

By default, logs are written to `TOMCAT_HOME/logs`, but you can configure this using `TOMCAT_HOME/webapps/geo/WEB-INF/classes/logback.xml`. Open the file with a text editor and edit the `LOG_HOME` property.

```
<property name="LOG_HOME" value="{path-to-file}/logs"/>
```

Change Mapping File Location

By default, the geo server looks for `geo-ip-mappings.xml` in `TOMCAT_HOME/webapps/geo/`. To change the location, open `TOMCAT_HOME/webapps/geo/WEB-INF/web.xml` with a text editor and change value for `AD_GEO_CONFIG_FILE`.

```
<web-app ...>
    <!-- ... -->
    <servlet>
        <servlet-name>FrontControllerServlet</servlet-name>
        <servlet-class>com.appdynamics.eum.geo.web.FrontControllerServlet</servlet-class>
        <context-param>
            <param-name>AD_GEO_CONFIG_FILE</param-name>
            <param-value>{path-to-file}/geo-ip-mappings.xml</param-value>
        </context-param>
        <!-- ... -->
    </servlet>
    <!-- ... -->
</web-app>
```

ℹ In previous versions of the geo server, the enclosing tag was a `<context-param>`. This has now been changed to an `<init-param>`.

For On-Premises EUM Servers Only: Use geo-ip-mappings.xml

If your installation uses an on-premises EUM Server *and* you have internal browsers from the same network as the Server that you want to identify, instead of setting up a separate custom geo-server, you can choose to simply modify the EUM Server's `geo-ip-mappings.xml` file as described above. The sample is in the `bin` directory of the EUM Server. The EUM Server automatically reads the file and uses it first to try and resolve the location, before using the Neustar IP database.

Precedence in Resolving Locations

The custom geo server resolves locations based on the following precedence, from highest to lowest:

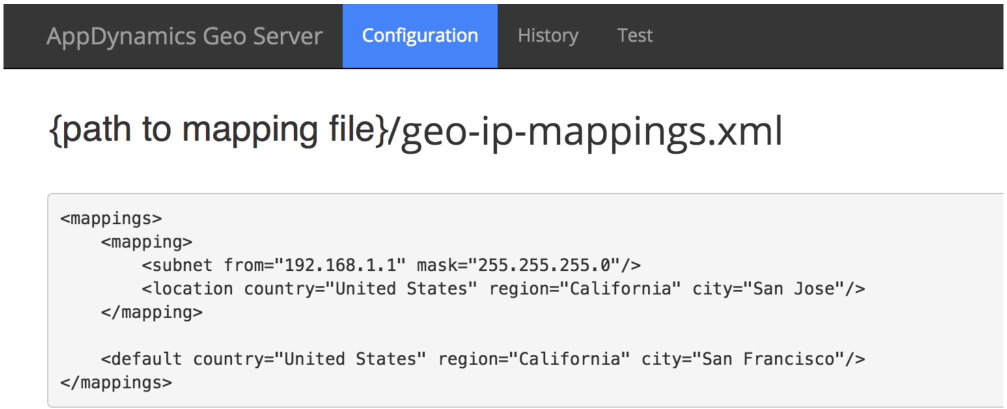
- An IP address set by customizing the JavaScript agent. For more information, see Set the Origin Location of the Request.
- An explicit query parameter: for example, `http://mycompany.com/geo/resolve.js?ip=196.166.2.1`.
- An IP provided using the `AD-X-Forwarded-For` header
- An IP provided using the `X-Real-IP` header
- An IP provided using the `X-Forwarded-For` header
- The remote address of the HTTP request

Debugging

ℹ Because this debugging feature has a small performance impact, it should be turned off before putting the geo server into production.

To aid in debugging, the geo server ships with a debugging web interface enabled. You can reach this interface by navigating to `http://<host>:<port>/geo/debug` with a web browser.

The first tab, **Configuration**, displays the contents of the mapping file currently in use.

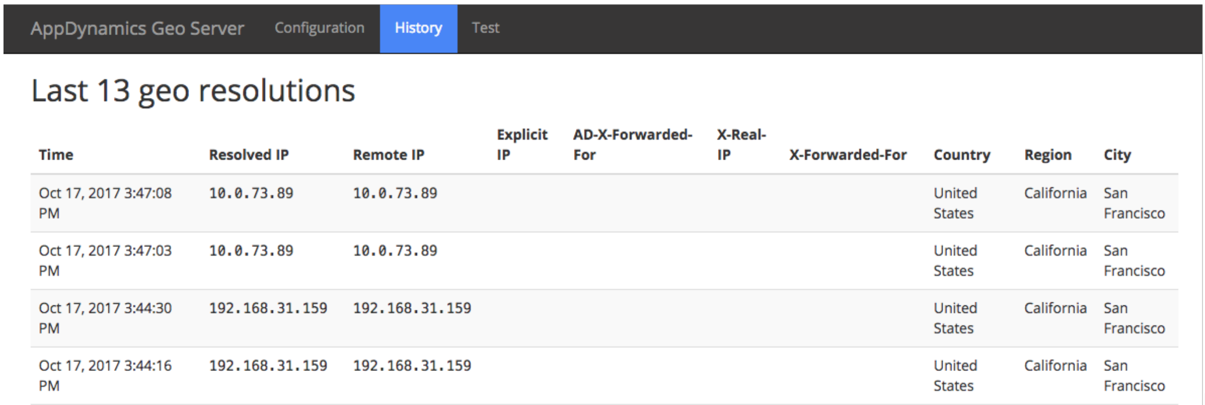


{path to mapping file}/geo-ip-mappings.xml

```
<mappings>
    <mapping>
        <subnet from="192.168.1.1" mask="255.255.255.0"/>
        <location country="United States" region="California" city="San Jose"/>
    </mapping>

    <default country="United States" region="California" city="San Francisco"/>
</mappings>
```

The second tab, **History**, shows the last few geo resolutions that have been performed.



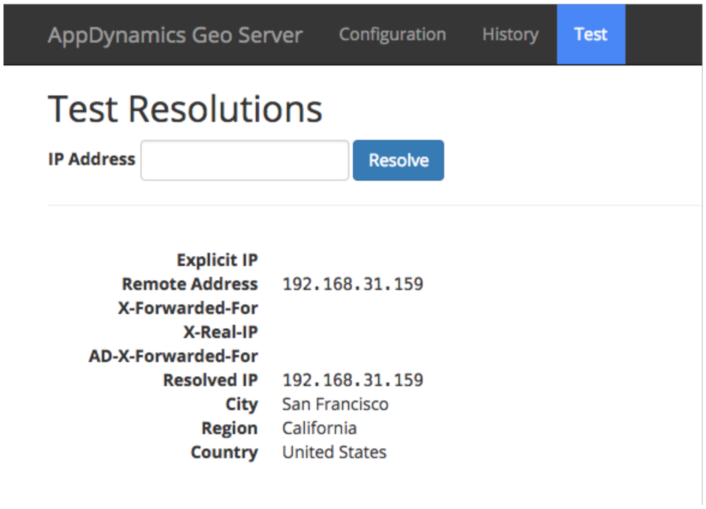
Last 13 geo resolutions

Time	Resolved IP	Remote IP	Explicit IP	AD-X-Forwarded-For	X-Real-IP	X-Forwarded-For	Country	Region	City
Oct 17, 2017 3:47:08 PM	10.0.73.89	10.0.73.89					United States	California	San Francisco
Oct 17, 2017 3:47:03 PM	10.0.73.89	10.0.73.89					United States	California	San Francisco
Oct 17, 2017 3:44:30 PM	192.168.31.159	192.168.31.159					United States	California	San Francisco
Oct 17, 2017 3:44:16 PM	192.168.31.159	192.168.31.159					United States	California	San Francisco

By default, the last 20 resolutions are shown, but this can be configured in `TOMCAT_HOME/webapps/geo/WEB-INF/web.xml.`

```
<web-app ..>
    <!-- ... -->
    <servlet>
        <servlet-name>FrontControllerServlet</servlet-name>
        <servlet-class>com.appdynamics.eum.geo.web.FrontControllerServlet</servlet-class>
        <!-- ... -->
        <init-param>
            <param-name>HISTORY_MAX_COUNT</param-name>
            <param-value>20</param-value>
        </init-param>
    </servlet>
    <!-- ... -->
</web-app>
```

The third tab, **Test**, can be used to test the mapping file by trying to resolve an arbitrary IP address.



Test Resolutions

IP Address

Explicit IP	
Remote Address	192.168.31.159
X-Forwarded-For	
X-Real-IP	
AD-X-Forwarded-For	
Resolved IP	192.168.31.159
City	San Francisco
Region	California
Country	United States

When you first navigate to this tab, it shows the geo resolution for your browser's IP address. The form in this tab can be used to try the resolution of another IP address.

Disabling debug

Open `TOMCAT_HOME/webapps/geo/WEB-INF/web.xml` and set `DEBUG_ENABLED` to `false.`

```
<web-app ..>
    <!-- ... -->
    <servlet>
        <servlet-name>FrontControllerServlet</servlet-name>
        <servlet-class>com.appdynamics.eum.geo.web.FrontControllerServlet</servlet-class>
        <!-- ... -->
        <init-param>
            <param-name>DEBUG_ENABLED</param-name>
            <param-value>false</param-value>
        </init-param>
    </servlet>
    <!-- ... -->
</web-app>
```

EUM Server Component Versions

This page describes how to check version information and the version of components bundled with the EUM Server. This information is useful when troubleshooting the system or performing other administrative tasks.

> ⚠ AppDynamics maintains and updates the bundled components as part of the AppDynamics Platform. Do not attempt to upgrade a bundled component independently of the platform upgrade procedure.

EUM Server Version

To view the version of your running EUM Server, run the following:

```
curl http(s)://<domain-name>:7001/v2/version
```

To get more information about the EUM Server, see EUM Server Endpoints.

Bundled MySQL Database Version

The AppDynamics EUM Server uses MySQL as its default database, where it stores license/account information, metadata, and applications names. The MySQL database files are installed in `<eum_home>/data` by default.

The latest AppDynamics release bundles MySQL version 5.7.33.

Bundled Java Version

The EUM Server bundles and uses Java 1.8.0_162.

Upgrade the Production EUM Server

This page describes how to upgrade an EUM Server to the latest production installation. This is usually done alongside an upgrade to the other platform components, such as the Controller and Events Service.

Who Should Use This Document

Anyone wanting to upgrade the EUM Server to the latest available version should use this document.

Before You Begin

Before you start upgrading the EUM Server, make sure that you are using the correct update order.

Upgrade Procedure

The instructions below show you how to upgrade your EUM Server to the latest version of the production EUM Server. Because in the EUM Server 4.4, the EUM MySQL database was moved from the Controller host machine to the EUM Server host machine, there are separate instructions below to help you migrate your data from versions earlier than 4.4 to the latest version. If you are upgrading from EUM Server 4.4 or higher to the latest version, you will not have to migrate your data, but you are advised to make a backup of your data.

Using SSL with the EUM Server

If you are upgrading to EUM Server 4.5.6 or higher, you are recommended to downgrade the version of the JRE bundled with the EUM Server to 1.8.0 _152 to avoid performance issues.

Troubleshoot EUM Server Installation

The following sections provide troubleshooting information for the EUM Server installation.

End User Data Does Not Appear in the Controller

If end user data does not appear in the Controller, follow these steps to troubleshoot the installation:

1. Check the Controller logs for errors in attempting to connect to the EUM Server. Also, see if the Controller UI allows you to enable EUM. If so, it's likely that the connection between the Controller and EUM Server is working.
2. Check the logs of the EUM Server, especially `<EUM_home>/logs/eum-processor.log`. In the log, verify that the server started successfully and is receiving beacons from agents.
3. Make sure that the EUM JavaScript Agent is actually injected into the monitored page and that the agent can load the remote JavaScript.
4. Use browser debugging tools to check for JavaScript errors in the monitored page.

License Not Installed

If the installer indicates that it was not able to install the license, or after installation, if the EUM Server fails to start with a license exception, try installing the license manually.

With the Controller running and accessible to the EUM Server machine, install the license manually. Before starting, make sure the `license.lic` file is at an accessible location on the EUM Server machine. Then install the license as follows:

1. Verify that the `JAVA_HOME/bin` is in the system PATH variable and points to a Java 1.7 instance.
2. In Windows, open an elevated command prompt (run as administrator).
3. From the command line, navigate to the `eum-processor` directory under your AppDynamics home.
4. From the `eum-processor` directory, run the following script:
   - On Linux:
     `./bin/provision-license <path_to_license_file>`
   - On Windows:
     `bin\provision-license.bat <path_to_license_file>`

EUM License Has Not Been Provisioned for an Account

Follow the instructions below to provision new EUM licenses for accounts on a multi-tenant Controller UI.

1. Navigate to the **Administration** page of your on-premises Controller: `http(s)://<hostname>:<port>/controller/admin.jsp`
2. Click **Accounts**.
3. Select the account name that you want to provision EUM for and click **Edit**.
4. Scroll down to the **End User Monitoring (EUM)** panel.
5. From the **Browser Real User Monitoring** section:
   a. Copy the EUM license key from your license file into the **EUM License Key** field.
   b. Select a license type (**EUM Lite** or **EUM Pro**) from the **License Type** dropdown.
   c. Enter your allotted Browser RUM units into the **Browser RUM Units Licensed** field.
   d. Set overages from the **Allow Overages** dropdown.
6. Complete the steps above for the **Mobile Real User Monitoring** section.
7. Click **Save**.

Exception When Updating Application Store

You may need to tune the database thread pool. The EUM Server ships with a blank c3p0 xml configuration file to help manage this. For information on using c3p0, see the docs here. You should make your changes to `<eum_server_home>/bin/c3p0.xml.`

"Too Many Open Files" Exception

Exception messages such as the following indicate insufficient open file descriptor limits on the EUM Server machine:

```
java.io.IOException: Too many open files
```

See EUM Server Deployment for operating system requirements, including recommended settings for `nofile` and `nproc` limits for the EUM Server operating system.

Controller Cannot Reach the Events Service

In the Administration Console, make sure that the Controller setting named `eum.es.host` is set to the correct connection settings for the Events Service instance, and that the Events Service is properly installed and running at that location. If the Events Service is a cluster with a load balancer in front of it, this should be the VIP of the Events Service as exposed at the load balancer.

For more information, see Connect to the Events Service.

Events Service Deployment

Related pages:

- Events Service Requirements
- Install the Events Service on Linux
- Install the Events Service on Windows

The AppDynamics Events Service is the on-premises data storage facility for unstructured data generated by Application Analytics, Database Visibility, and End User Monitoring deployments.

> ⓘ The following information and instructions are intended for on-premises deployments only. SaaS deployments are managed by AppDynamics.

About the Events Service

The MySQL database embedded in the Controller stores application metric and configuration data generated by the Controller. While an embedded database is sufficient for storing this type of data, the high-volume, performance-intensive nature of analytics data requires dedicated, horizontally scalable storage. In an AppDynamics deployment, this role is served by the AppDynamics Events Service.

If you are installing the server components for End User Monitoring, Application Analytics, or Database Visibility, you need to use a scaled-out on-premises Events Service. Scaled-out means that the service is not installed on the Enterprise Console host. This type of Events Service can be deployed as a single node or a cluster of three or more nodes based on your needs. Additionally, you can add nodes to a scaled-out Events Service after you install it. It is not recommended that you add the Controller host as part of the cluster. See Administer the Events Service.

However, for data redundancy and storage scalability or if you are using End User Monitoring, Application Analytics or Database Visibility in an on-premises Controller deployment, you need to deploy a dedicated Events Service installation.

Multiple AppDynamics components that depend on the Events Service should use the same Events Service instance or cluster.

Deployment Topology Overview

You can deploy the Events Service to a single node or to a cluster of three or more nodes. Clusters are horizontally scalable, so nodes can be added as your data storage requirements grow. A cluster also provides data replication and redundancy, helping to ensure data integrity in the event of a node failure.

The Controller includes an embedded Events Service instance used by the Database Visibility product by default. However, the embedded Events Service is not meant to be used with production Application Analytics or EUM installations, since it runs on the same machine as the Controller and does not offer data replication or scalability. It may be used for small-scale environments, especially those intended for demonstration or learning purposes. Note however that it's not possible to migrate data from the embedded Events Service to an external Events Service instance if upgrading later.

Your Events Service can be deployed to support multiple Controllers, becoming a shared Events Service.

Single Node Deployment

In a single node Events Service deployment, the Events Service runs on a dedicated machine. The Controller and other Events Service clients can connect directly to the Events Service node or through a load balancer. Deploying a single node Events Service behind a load balancer allows you to grow the deployment to a multi-node cluster easily, without having to modify the clients.

> ⚠ Single node deployment is recommended for test environments only. Production environments should deploy a multi-node cluster (see below for details).

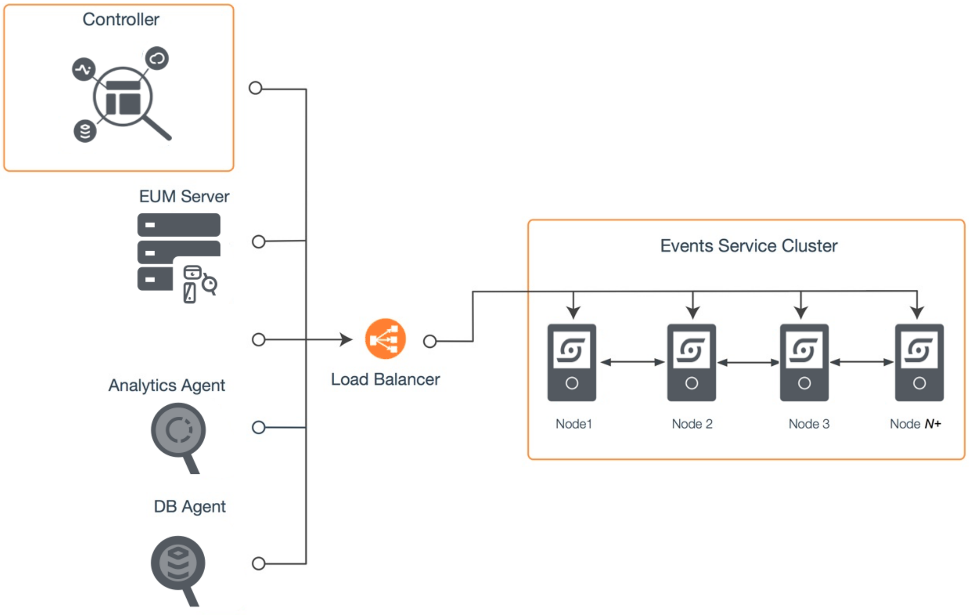
Multi-Node Cluster

A multi-node cluster is made up of three or more nodes. With a cluster, the Controller and other Events Service clients, the EUM Server and Analytics Agent connect to the Events Service through a load balancer, which distributes load to the Events Service cluster members.

AppDynamics recommends multi-node clusters for production environments. Multi-node clusters provide the following benefits:

- Safeguards against data loss: multi-node clusters replicate your data. If one node goes down in production, you would have at least two more nodes storing your data. Additionally, the Events Service continues to run as long as one node runs. Should a node go down in a single node deployment, the Events Service stops running and you would lose your data.
- Provides redundancy: in a multi-node cluster, you can swap nodes.

In a single-node deployment, connect through a load balancer or directly to the Events Service.



The nodes in a cluster swap a large amount of data. For this reason, when deploying a cluster, make sure to install all cluster nodes within the same local network, ideally, attached to the same network switch.

Supported Deployments

Use this table to determine the supported deployment type and environment for the Events Service.

Deployment Types	Development Environment	Production Environment
Multi-Node Clustered Events Service (3+ nodes) (version 20.2 or later)	Yes	Yes

Shared Events Service

You can connect multiple Controllers to a single on-premises Events Service deployment using the same procedure that is required to connect a single Controller. You would need to configure the Controller URLs for the Events Service to point to the shared Events Service, and make sure the Controller keys are correct. The Controllers can then handle syncing to the shared Events Service. There are no additional requirements.

When compared to a multiple Events Service cluster configuration, a shared Events Service configuration requires less maintenance and lowers cost. Since the Events Service is horizontally scalable, having a single large instance provides the same functionality as multiple instances.

There is no limit to the number of Controllers that can share an Events Service. However, it is recommended that you use separate Events Services for dev and prod. Work with your AppDynamics account representative if you plan to expand your shared Events Service cluster.

Default Ports

The default ports used by the Events Service are:

- Events Service API Store Port: 9080
- Events Service API Store Admin Port: 9081

The Events Service cluster members use additional ports for internal communication among the cluster members. All the ports used within the cluster are listed in the Events Service configuration file, `conf/events-service-api-store.properties`.

Secure the Events Service

By default, the `ad.es.node.http.enabled` property in the Events Service configuration file is set to `false`. To debug or troubleshoot issues with the Events Service, you or the AppDynamics Support team can enable this property and allow HTTP requests (such as node stats) in ElasticSearch by changing the setting to `true`. Following the debugging session, immediately disable this property to prevent any (potential) security vulnerabilities and restart the Events Service. The downtime is dependent on the hardware and the service will be unavailable until it restarts.

Events Service Requirements

Related pages:

- Prepare the Events Service Host

This page describes general hardware and software requirements for the machines that host Events Service nodes.

General Requirements

- Determine which version of the Events Service that is compatible with your other platform components.
- Use a supported Windows 64-bit or Linux 64-bit based operating system supported by the platform. See Platform Requirements.
- Solid-state drives (SSD) can significantly outperform hard disk drives (HDD), and are therefore recommended for production deployments. Ideally, the disk size should be 1 TB or larger.
- The Events Service must run on dedicated machines with identical directory structures, user account profiles, and hardware profiles.
- For heap space allocation, AppDynamics recommends allocating half of the available RAM to the Events Service process, with a minimum of 7 GB up to 31 GB.
- When testing the events ingestion rate in your environment, it is important to understand that events are batched. Ingestion rates observed at the scale of a minute or two may not reflect the overall ingestion rate. For best results, observe ingestion rate over an extended period of time, several days at least.
- The Events Service requires Java 8u172.
- Keep the clocks on your application, Controller, and Events Service nodes in sync to ensure consistent event time reporting across the AppDynamics deployment.
- Your firewall should not block the Events Service REST API port 9080, otherwise, the Enterprise Console will not be able to reach the Events Service remotely.

Hardware Capacity and Resource Planning

When estimating your hardware requirements, the first step is to determine the event ingestion rate (for Transaction Analytics) or the amount of data being indexed (for Log Analytics). This helps you to determine the number of analytics license units you will need.

Once you determine your license units requirements, it is important to consider other factors that affect the hardware capacity, such as the processing load of queries run against the Events Service and the actual type of hardware used. A physical server is likely to perform better than a virtual machine. You should also take into account seasonal or daily spikes in activity in your monitored environment in your considerations.

An event is the basic unit of data in the events service. In terms of application performance management, a Transaction Analytics event corresponds to a call received at a tier. A business transaction instance that crosses three tiers, therefore, would result in three events being generated. In application performance management metrics, the number of business transaction instances is reflected by the number of calls metric for the overall application. In End User Monitoring, each page view equates to an event, as does each Ajax request, network request, or crash report.

Events Service Node Sizing Based on License Units

You can plan your hardware requirements with the data in the section. It describes recommended hardware configurations (in the context of Amazon EC2 instance types) corresponding to the number of license entitlement units for Log and Transaction Analytics. See License Entitlements and Restrictions for details about license units for Log and Transaction Analytics.

For additional Events Service sizing information, see the following AppDynamics Community articles:

- Understanding EUM and Events Service concepts (describes the concepts necessary to build the profile)
- Build the Analytics traffic profile
- Size the Events Service and EUM using the profile (contains the Analytics Recipe Book for on-premises configuration)
- Limit EUM and Analytics usage (describes how to configure rules to enforce Analytics trade-offs when deploying Analytics on-premises)

The hardware shown for each license amount represents the hardware capacity of a theoretical combined load of both Transaction Analytics and Log Analytics events. The numbers used were derived from actual tests that were performed with an uncombined load, from which the following numbers were extrapolated. Note that the test conditions did not include query load and so may not be representative of a true production analytics environment.

> ⓘ The following table shows sizing recommendations and describes the size of the cluster used for testing. This does not mean you are limited to a seven-node event service. If you need to go beyond seven nodes, contact your AppDynamics account representative to ensure proper sizing for your specific environment.

It is to be noted that the retention can be 8, 30, 60 or 90 days which will directly affect storage requirements.

Event Type	AWS Machine Instance Type										
	i2.2xlarge (61 GB RAM, 8 vCPU, 1600 GB SSD)				i2.4xlarge (122 GB RAM, 16 vCPU, 3200 GB SSD)				i2.8xlarge (244 GB RAM, 32 vCPU, 6400 GB SSD)		
	1 node	3 nodes	5 nodes	7 nodes	1 node	3 nodes	5 nodes	7 nodes	1 node	3 nodes	5 nodes
Transaction Analytics license units	20	37	44	63	22	41	84	113	53	94	120
Log Analytics license units	7	10	17	19	16	19	32	44	39	116	270

The following points describe the test conditions under which the license units-to-hardware profile mappings in the table were generated:

- Average Log event size in bytes: 350
- Average size of business transaction event: 1 KB
- Tiers in business transaction: 3

The tests were conducted on virtual hardware and programmatically generated workload. Real-world workloads may vary. To best estimate your hardware sizing requirements, carefully consider the traffic patterns in your application and test the Events Service in a test environment that closely resembles your production application and user activity.

Minimum Events Service Node Sizing

To configure the Events Service 3 nodes minimum are required.

Database Visibility Events Service Sizing

Database Visibility features use the Events Service for storage. The ingestion capacity and sizing profile for Database Visibility Analytics events are equivalent to that of Log Analytics, with the size of the raw event being about 2 kilobytes on average.

End User Monitoring Events Service Sizing

End User Monitoring includes Analytics-related features that send data to the Events Service.

In End User Monitoring, each page view equates to an event, as does each Ajax request, network request, or crash report. There can be a few dozen Ajax requests for every page load. In general, the ingestion capacity and sizing profile for EUM or Database Visibility Analytics events are equivalent to that for Log Analytics, with the size of the raw events being about 2 kilobytes on average.

To calculate the sizing for EUM, multiply the peak number of browser records in a day by 12 KB. If peak capacity is reached, the Events Service simply starts dropping traffic.

The table below provides details about the memory and storage of different types of browser records. The default retention period is configurable.

Browser Record Type	Memory Requirements Per Event	Optional	Default Retention
BasePage, iFrame, Virtual Page	1 KB / 1.5 KB (with sessions enabled)	No	8 days
Ajax requests	1 KB	Yes	

> ⓘ By default, Ajax requests are not stored in the Events Service.

Prepare the Events Service Host

Related pages:

- Events Service Requirements

This page describes how to prepare the machine that will host Events Service nodes, along with general requirements for the environment.

Network and Port Settings

The Controller and Events Service must reside on the same local network and communicate by the internal network. Do not deploy the cluster to nodes on different networks, whether relative to each other or to the Controller and the Enterprise Console. When identifying cluster hosts in the configuration, you will need to use the internal DNS name or IP address of the host, not the externally routable DNS name.

For example, in terms of an AWS deployment, use the private IP address such as `172.31.2.19` rather than public DNS hostname such as `ec2-34-201-129-89.us-west-2.compute.amazonaws.com`.

On each machine, the following ports need to be accessible to external (outside the cluster) traffic:

- Events Service API Store Port: 9080
- Events Service API Store Admin Port: 9081

For a cluster, ensure that the following ports are open for communication between machines within the cluster. Typically, this requires configuring iptables or OS-level firewall software on each machine to open the ports listed

- 9300 – 9400

The following shows an example of iptables commands to configure the operating system firewall:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9081 -j ACCEPT
-A INPUT -m state --state NEW -m multiport -p tcp --dports 9300:9400 -j ACCEPT
```

If a port on the Events Service node is blocked, the Events Service installation command will fail for the node and the Enterprise Console command output and logs will include an error message similar to the following:

```
failed on host: <ip_address> with message: Uri [http://localhost:9080/_ping] is un-pingable.
```

If you see this error, make sure that the ports indicated in this section are available to other cluster nodes.

Configure Cluster Nodes that Run Linux

If deploying to Linux machines, on each node in the Events Service cluster, make these configuration changes:

1. Using a text editor, open `/etc/sysctl.conf` and add the following:

   `vm.max_map_count=262144`

2. Raise the open file descriptor limit in `/etc/security/limits.conf`, as follows:

   ```
   <username_running_eventsservice>     soft    nofile          96000
   <username_running_eventsservice>     hard    nofile          96000
   ```

   Replace `username_running_eventsservice` with the username under which the Events Service processes run. So if you are running Analytics as the user `appduser`, you would use that name as the first entry.

Configure SSH Passwordless Login

For Linux deployments, you will use the Enterprise Console to deploy and manage the Events Service cluster.

The Enterprise Console needs to be able to access each cluster machine using passwordless SSH for a non-embedded Events Service. Before starting, enable key-based SSH access as described here.

This setup involves generating a key pair on the Enterprise Console and adding the public key as an authorized key on the cluster nodes. The following steps take you through the configuration procedure for an example scenario. You will need to adjust the steps based on your environment.

If you are using EC2 instances on AWS, the following steps are taken care of for you when you provision the EC2 hosts. At that time, you are prompted for your PEM file, which causes the public key for the PEM file to be copied to the authorized_keys of the hosts. You can skip these steps in this case.

On the host machine, follow these steps:

1. Log in to the Enterprise Console host machine or switch to the user you will use to perform the deployment:

   ```
   su - $USER
   ```

2. Create a directory for SSH artifacts (if it doesn't already exist) and set permissions on the directory, as follows:

   ```
   mkdir -p ~/.ssh
   chmod 700 ~/.ssh
   ```

3. Change to the directory:

   ```
   cd .ssh
   ```

4. Generate PEM public and private keys in RSA format:

   ```
   ssh-keygen -t rsa -b 2048 -v -m pem
   ```

5. Enter a name for the file in which to save the key when prompted, such as `appd-analytics`.
6. Rename the key file by adding the `.pem` extension:

   ```
   mv appd-analytics appd-analytics.pem
   ```

   You will later configure the path to it as the `sshKeyFile` setting in the Enterprise Console configuration file, as described in Deploying an Events Service Cluster.
7. Transfer a copy of the public key to the cluster machines. For example, you can use scp to perform the transfer as follows:

   ```
   scp ~/.ssh/myserver.pub host1:/tmp
   scp ~/.ssh/myserver.pub host2:/tmp
   scp ~/.ssh/myserver.pub host3:/tmp
   ```

   Continuing with the example, `myserver` should be appd-analytics.
   The first time you connect you may need to confirm the connection to add the cluster machine to the list of known hosts and enter the user's password.
8. On each cluster node (host1, host2, and host3), create the .ssh directory in the user home directory, if not already there, and add the public key you just copied as an authorized key:

   ```
   cat /tmp/appd-analytics.pub >> .ssh/authorized_keys
   chmod 600 ~/.ssh/authorized_keys
   ```

9. Test the configuration from the host machine by trying to log in to a cluster node by `ssh`:

   ```
   ssh host1
   ```

   If unable to connect, make sure that the cluster machines have the `openssh-server` package installed and that you have modified the operating system firewall rules to accept SSH connections. If successful, you can use the Enterprise Console to deploy the platform.

If you encounter the following error, use the instructions in this section to double-check your passwordless SSH configuration:

```
Copying JRE to the remote host failed on host: 172.31.57.204 with message: Failed to upload file: java.net.
ConnectException: Connection timed out
```

Install the Events Service on Linux

Related Pages:

- Events Service Requirements

This page describes how to install and administer the Events Service on Linux systems through the CLI. Steps for scaling up an embedded Events Service using the Enterprise Console are also included.

The AppDynamics Enterprise Console automates the task of installing and administering an Events Service deployment through either the GUI or CLI. For information on installing Events Service using the Enterprise Console, see Custom Install.

> ⚠ You do not need to specify the installation or data directory for the Events Service installation. If you do, use a different one from the platform directory.

Events Service Host Requirements

Before starting, be sure to review the Release Notes for known issues and late-breaking information on using the Events Service and Enterprise Console. Also, observe the following requirements:

- The Events Service can be deployed as a single node or as a multi-node cluster of 3 or more nodes.
- The versions of Linux supported include the flavors and versions supported by the Controller, as indicated by Prepare Linux for the Controller.
- The Events Service must run on a dedicated machine. The machine should not run other applications or processes not related to the Events Service.
- Use appropriately sized hardware for the Events Service machines. The Enterprise Console checks the target system for minimum hardware requirements. For more information on these requirements, see the description of the profile argument to the Events Service install command in Install the Events Service Cluster.
- The Controller and Events Service must reside on the same local network and communicate by the internal network. Do not deploy the cluster to nodes on different networks, whether relative to each other or to the Controller where the Enterprise Console runs. When identifying cluster hosts in the configuration, you will need to use the internal DNS name or IP address of the host, not the externally routable DNS name.
  For example, in terms of an AWS deployment, use the private IP address such as `172.31.2.19` rather than public DNS hostname such as `ec2-34-201-129-89.us-west-2.compute.amazonaws.com`.
- Make sure that the appropriate ports on each Events Service host are open. See Port Settings for more information.
- The Enterprise Console uses an SSH key to access the Events Services hosts. See the section below for information on generating the key.
- Events Service nodes normally operate behind a load balancer. When installing an Events Service node, the Enterprise Console automatically configures a direct connection from the Controller to the node. If you deploy a cluster, the first primary node is automatically configured as the connection point in the Controller. You will need to reconfigure the Controller to connect through the load balancer VIP after installation, as described below. For sample configurations, see Load Balance Events Service Traffic.

Port Settings

Each machine must have the following ports accessible to external (outside the cluster) traffic:

- Events Service API Store Port: 9080
- Events Service API Store Admin Port: 9081

For a cluster, ensure that the following ports are open for communication between machines within the cluster. Typically, this requires configuring `iptables` or OS-level firewall software on each machine to open the ports listed

- 9300 – 9400

The following shows an example of `iptables` commands to configure the operating system firewall:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 9081 -j ACCEPT
-A INPUT -m state --state NEW -m multiport -p tcp --dports 9300:9400 -j ACCEPT
```

If a port on the Events Service node is blocked, the Events Service installation command will fail for the node and the Enterprise Console command output and logs will include an error message similar to the following:

```
failed on host: <ip_address> with message: Uri [http://localhost:9080/_ping] is un-pingable.
```

If you see this error, make sure that the ports indicated in this section are available to other cluster nodes.

Create the SSH Key

When installing Events Service, you will need to provide the SSH key that the Enterprise Console can use to access Events Service hosts remotely. Before starting, create the PEM public and private keys in RSA format. The key file must not use password protection.

For example, using `ssh-keygen`, you can create the key using the following command:

```
ssh-keygen -t rsa -b 2048 -v -m pem
```

Configure SSH Passwordless Login

The Enterprise Console needs to be able to access each cluster machine using passwordless SSH. Before starting, enable key-based SSH access.

This setup involves generating a key pair on the Enterprise Console host and adding the Enterprise Console's public key as an authorized key on the cluster nodes. The following steps take you through the configuration procedure for an example scenario. You will need to adjust the steps based on your environment.

If you are using EC2 instances on AWS, the following steps are taken care of for you when you provision the EC2 hosts. At that time, you are prompted for your PEM file, which causes the public key for the PEM file to be copied to the `authorized_keys` of the hosts. You can skip these steps in this case.

On the Enterprise Console machine, follow these steps:

1. Log in to the Enterprise Console machine or switch to the user you will use to perform the deployment:

   ```
   su - $USER
   ```

2. Create a directory for SSH artifacts (if it doesn't already exist) and set permissions on the directory, as follows:

   ```
   mkdir -p ~/.ssh
   chmod 700 ~/.ssh
   ```

3. Change to the directory:

   ```
   cd .ssh
   ```

4. Generate PEM public and private keys in RSA format:

   ```
   ssh-keygen -t rsa -b 2048 -v -m pem
   ```

   The key file must not use password protection.
5. Enter a name for the file in which to save the key when prompted, such as `appd-analytics`.

6. Rename the key file by adding the `.pem` extension:

   ```
   mv appd-analytics appd-analytics.pem
   ```

   You will later configure the path to it as the `sshKeyFile` setting in the Enterprise Console configuration file, as described in Deploying an Events Service Cluster.
7. Transfer a copy of the public key to the cluster machines. For example, you can use `scp` to perform the transfer as follows:

   ```
   scp ~/.ssh/myserver.pub host1:/tmp
   scp ~/.ssh/myserver.pub host2:/tmp
   scp ~/.ssh/myserver.pub host3:/tmp
   ```

Continuing with the example, `myserver` should be `appd-analytics`.
The first time you connect you may need to confirm the connection to add the cluster machine to the list of known hosts and enter the user's password.

8. On each cluster node (host1, host2, and host3), create the `.ssh` directory in the user home directory, if not already there, and add the public key you just copied as an authorized key:

   ```
   cat /tmp/appd-analytics.pub >> .ssh/authorized_keys
   chmod 600 ~/.ssh/authorized_keys
   ```

9. Test the configuration from the Controller machine by trying to log in to a cluster node by `ssh`:

   ```
   ssh host1
   ```

   > If unable to connect, make sure that the cluster machines have the `openssh-server` package installed and that you have modified the operating system firewall rules to accept SSH connections. If successful, you can use the Enterprise Console on the Controller host to deploy the Events Service cluster, as described next.

If the Enterprise Console attempts to install the Events Service on a node for which passwordless SSH is not properly configured, you will see the following error message:

```
./bin/platform-admin.sh add-hosts --hosts <es_host_1> <es_host_2> <es_host_3> --credential <credential name> --
platform-name <name of platform>
...
Failed to connect to the remote host. Please verify that the host name and credentials you provided are correct.
For more information, consult the documentation: https://docs.appdynamics.com/display/PRO45
/Administer+the+Enterprise+Console#AdministertheEnterpriseConsole-manage-hostsManageHosts
```

If you encounter this error, use the instructions in this section to double-check your passwordless SSH configuration. Also, you need to and `add-hosts` first then install `event-service`. If SSH configuration is not setup correctly, `add-hosts` commands will fail.

> ⚠ The `add-hosts` command is to add hostnames into platforms. During the `events-service` installation, the hosts come from the platform hosts, and then they are used on the `events-service`.

Tune the Operating System for Production Cluster Nodes

Before installing the Events Service cluster, you need to perform a few manual changes as described below. These are particularly relevant for production Events Service deployments. On each node in the cluster, make these configuration changes:

1. Using a text editor, open `/etc/sysctl.conf` and add the following:

   ```
   vm.max_map_count=262144
   ```

2. Raise the open file descriptor limit in `/etc/security/limits.conf`, as follows:

   ```
   <username_running_eventsservice>     soft    nofile          96000
   <username_running_eventsservice>     hard    nofile          96000
   <username_running_eventsservice>     soft    memlock         unlimited
   <username_running_eventsservice>     hard    memlock         unlimited
   ```

3. Disable swap memory by running the following command. Remove swap mount points by removing or commenting the lines in `/etc/fstab` that contain the word swap.

   ```
   swapoff -a
   ```

Installing the Events Service Using the GUI

In the GUI, the Express Install option automatically installs an Events Service on the same host as the Controller. The Custom Install option can install an embedded or scaled-up Events Service. If you install an embedded Events Service and want to switch to a scaled-up Events Service, complete the steps described in Scaling Up an Embedded Events Service.

Installing the Events Service Using the CLI

1. Set up load balancing. See Load Balance Events Service Traffic for information about configuring the load balancer.
2. At the command line, navigate to the `platform-admin` directory created at Enterprise Console installation. See Install the Enterprise Console.
3. If it has been more than one day since your last session, you will have to log in with the following command:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

4. Create a platform as follows:

   ```
   bin/platform-admin.sh create-platform --name <platform_name> --installation-dir
   <platform_installation_directory>
   ```

   The installation directory is the directory where the Enterprise Console installs all platform components.

   > The same installation directory should exist and is used on all remote nodes. This is done to maintain the homogeneity of the configuration across different nodes.

5. Add the SSH key that the Enterprise Console will use to access and manage the Events Service hosts remotely. (See Create the SSH Key for more information):

   ```
   bin/platform-admin.sh add-credential --credential-name <name> --type ssh --user-name <username> --ssh-
   key-file <file path to the key file> --platform-name <name of platform>
   ```

   `<file path to the key file>` is the private key for the Enterprise Console machine. The installation process uses the keys to connect to the Events Service hosts. The keys are not deployed, but instead, encrypted and stored in the Enterprise Console database.

   The `platform-name` parameter is optional.
6. Add hosts to the platform, passing the credential you added to the platform:

   ```
   bin/platform-admin.sh add-hosts --hosts es_host_1 es_host_2 es_host_3 --credential <credential name> --
   platform-name <name of platform>
   ```

   The `platform-name` parameter is optional.
7. On each Events Service destination node in the cluster, create an installation directory for the Events Service. This is the directory you specified as the `installation-dir` argument while creating the platform in step (2).
8. Again at the command line for the Enterprise Console machine, run the following command from the `platform-admin` directory. Pass the cluster configuration settings as arguments to the command. The format for the command is the following:

   ```
   bin/platform-admin.sh submit-job --platform-name <platform-name> --service events-service --job install
   --target-version <latest> --args profile=<dev> serviceActionHost=<es_host_1>
   serviceActionHost=<es_host_2> serviceActionHost=<es_host_3>
   ```

   The `platform-name` and `jvmTempDir` parameters are optional.

   Arguments are:

   - `jvmTempDir`: Use this argument to override default JVM temporary `/tmp` directory in Linux installations.
   - `hosts`: Use this argument or `host-file` to specify the internal DNS hostnames or IP addresses of the cluster hosts in your deployment. With this argument, pass the hostnames or addresses as parameters. For example:

     ```
     --hosts 192.168.32.105 192.168.32.106 192.168.32.107
     ```

   - `host-file`: As an alternative to specifying hosts as --`hosts` arguments, pass them as a list in a text file you specify with this argument. Specify the internal DNS hostname or IP address for each cluster host as a separate line in the plain text file:

```
192.168.32.105
192.168.32.106
192.168.32.107
```

- `profile`: By default (with profile not specified), the installation is considered a production installation. Specifying a developer profile (`profile dev`) directs the Enterprise Console to use a reduced hardware profile requirement, suitable for non-production environments only. The Enterprise Console checks for the following resources:
    - For a dev profile, 1 core CPU, 1 GB RAM, and 2 GB disk space.
    - Otherwise, 4 core CPU, 12 GB RAM, and 128 GB of disk space.

For example:

```
bin/platform-admin.sh add-hosts --hosts ip-172-31-20-21.us-west-2.compute.internal ip-172-31-20-22.us-
west-2.compute.internal ip-172-31-20-23.us-west-2.compute.internal
```

If using a hosts text file, use the following command:

```
bin/platform-admin.sh add-hosts --hosts --host-file=/home/appduser/hosts.txt
```

9. Log in to each Events Service node machine, and run the script for setting up the environment as follows:

   The `tune-system.sh` script is used for optimizing your environment. It is optional.

   a. Add execute permission to the `tune-system.sh` script:

   ```
   chmod +x tune-system.sh
   ./tune-system.sh
   ```

   b. Run the script:

   ```
   sudo <installation_dir>/events-service/processor/bin/tool/tune-system.sh
   ```

10. If you are using a load balancer, use the virtual IP for the Events Service as presented at the load balancer. Configure the Controller connection to the Events Service as follows:

    a. Open the Administration Console.
    b. In the Controller settings pane, find `appdynamics.on.premise.event.service.url` and change its value to the URL of the virtual IP for the Events Service at the load balancer.

    It may take a few minutes for the Controller and Events Service to synchronize account information after you modify connection settings in the console.

When finished, use the Enterprise Console for any Events Service administrative functions. You should not need to access the cluster node machines directly once they are deployed. In particular, do not attempt to use scripts included in the Events Service node home directories.
The Enterprise Console automatically updates the Controller configurations after installation.

Scaling Up an Embedded Events Service

The following steps describe how to scale up an Events Service that is on a shared host with the Controller. This allows the embedded Events Service to run on a separate host. You can also install the Events Service on a separate host directly by using the Custom Install.

1. Set up load balancing. See Load Balance Events Service Traffic for information about configuring the load balancer.
2. Open the Enterprise Console GUI.
3. Verify that the credentials and hosts you want to use are added to the AppDynamics platform. For more information, see Administer the Enterprise Console.

   a. On the **Credential** page, add the SSH credentials for the hosts on which you want to install the Events Service.
   b. On the **Hosts** page, add the hosts. The Enterprise Console uses these hosts for the scaled-up Events Service, which requires at least one host or three or more hosts.
4. On the **Events Service** page, navigate to More link and select the Events Service and click **Scale Up Events Service** under More and complete the wizard. When you enter hosts to use for a scaled-up Events Service, do not include the Controller host.

> You do not need to restart the Controller since that is automatically done for you by the scale-up job.

5. Log in to each node machine, and run the script for setting up the environment as follows:

```
sudo <installation_dir>/events-service/latest/bin/tool/tune-system.sh
```

6. Navigate to the **Controller** page.
7. By default, the Enterprise Console configures the Events Service connection in the Controller to refer to the first primary node defined in the cluster. If you are using a load balancer, as recommended, you need to change this Controller setting to point to the Events Service VIP as presented at the load balancer instead, as follows:

   a. Open the Administration Console.
   b. In the **Controller settings** pane, find `appdynamics.on.premise.event.service.url.`
   c. Change the value of the setting for the URL to the VIP for the Events Service at the load balancer.
8. By default, Database Monitoring stores events data in the Events Service embedded in the Controller. To have it use the Events Service you just deployed, ensure that the `appdynamics.on.premise.event.service.key` value matches with `ad.accountmanager.key.controller` value inside the `events-service-api-store.properties` file.

   > Note that only newly generated Database Monitoring data will be stored in the Events Service; previously collected data will remain in the embedded Events Service instance unless it is migrated to the new Events Service. See Connect to the Events Service.

9. It may take a few minutes for the Controller and Events Service to synchronize account information after you modify connection settings in the Enterprise Console.

Troubleshooting Installation

If the Enterprise Console crashes or shuts down while installing the Events Service, the GUI may display that the installation is in progress. To resolve this issue, uninstall the Events Service with the CLI. Then, install the Events Service with the CLI.

Install the Events Service on Windows

Related pages:

- Events Service Requirements

You can install and administer the Events Service on Microsoft Windows systems as a single node or a cluster. Common use cases include:

- **Single-node Events Service** — Good for demonstration purposes and other scenarios where data redundancy and high availability are not required.
- **Three-node cluster** — The minimum size for a production Events Service cluster.
- **Cluster of four nodes or more** — For deployments where increased load or expected sizing exceeds the capacity of a three-node cluster.

Contact AppDynamics customer support if you anticipate deploying a cluster of 10 or more nodes.

> ⓘ Creating more than one instance of the same service type is not supported on Windows.

> ⚠ You do not need to specify the installation or data directory for the Events Service installation. If you do, use a different one from the platform directory.

Decide Which Node(s) to Make Master Nodes

Every Events Service node is either a master node or a data node. In an Events Service cluster, the master node both acts as a storage node, and manages the state of the data across the cluster, including the state of the replica. In a single-node deployment, there's not much for the master to do.

The master node is the first node to start up. If the master node becomes unavailable, the worker nodes attempt to elect a new master.

As you install Events Service, you specify the configuration for each node, which consists of two pieces of information:

- Whether to enable the node to serve as a master, and
- How many nodes (minimum) must be available in the cluster for a new master to be elected

This table describes what values to specify as you install:

Order of Node in the Deployment	Master-enabled?	Minimum available nodes required to elect a new master
First (only) node in a single-node installation	true	1
First, second, and third nodes of a three-node cluster	true	2
Fourth (or higher) node of a cluster	false	2

> ⚠ Specifying the installation or data directory is optional for the Events Service. If you do this, the directory you specify must not be the platform directory.

Installation Quick Start

Before you can install the Events Service on Windows, you must use Enterprise Console to install the Controller. The EUM Server must be installed separately as it cannot be installed using the Enterprise Console.

1. Install the Enterprise Console.
2. Use the Enterprise Console to create the platform and add hosts.
3. To install the Controller and Events Service on the same host, use the Express Install option.
   Use the Custom Install to install a scaled-out Events Service that runs on a host that is separate from the Controller. Custom installations provide more flexibility on where and how to install Controller and Events Service.
4. (Optional) Install the EUM Server.
5. Complete the post-install tasks for the Controller, Events Service, and EUM Server.

Deploy a Single-Node Events Service

> Ensure that you have met all of the current Events Service Requirements, including:
>
> - Have Java Runtime Environment (JRE) version = 1.8
> - Defined Java in the Windows environment variable path, or a JRE folder that is accessible by `events-service.exe` in the relative path `..\..\jre`

To manually install the Events Service on a single node:

1. Unzip the Events Service distribution archive to a directory on the target host. This creates the `events-service` directory with the Events Service artifacts.
2. Begin configuring the connection to the Events Service in the Controller.

   a. With the Controller running, open the Administration Console as the root user.
   b. In the **Controller Settings** page, search for `appdynamics.on.premise.event.service.url`.
   c. Replace the default value to the internal hostname for the Events Service machine, and the default value for the Events Service listen port, 9080.
   For example: http://*hostname:9080*. Select **Save**.

   > If you are putting a load balancer in front of the Events Service (required for a cluster), this will be the VIP for the Events Service as exposed at the load balancer. In this case, it is likely you will need to return to this step after you finish deploying the node and configuring the load balancer.

3. Obtain the Controller key:
   a. While on the **Controller Settings** page, search for the `appdynamics.on.premise.event.service.key` setting.

   > If you do not install the default Events Service, then the `appdynamics.on.premise.event.service.key` setting is blank. To create the `appdynamics.on.premise.event.service.key`, use a UUID generator.

   b. Copy the setting's value. You will need this to complete Step 4.
   c. Close the Administration Console.
4. Configure the connection from the Events Service to the Controller:

   a. From your Windows Explorer, open the `events-service\conf\events-service-api-store.properties` file to edit.
   b. Locate the `controller-key` property and add it:

   ```
   ad.accountmanager.key.controller=controller-key
   ```

5. Verify the heap settings for the Events Service processes:

   a. Verify that the minimum and maximum heap settings for the two Events Service processes (the Events Service JVM, and the Elasticsearch processes, respectively) are correct and sufficient for your deployment.
   The settings:

      - Are located in the `events-service-api-store.properties` file.
      - Use `g` for gigabyte (GB) and `m` for megabyte (MB).

   b. For the Events Service process, verify:

   ```
   ad.jvm.options.name=events-service.vmoptions
   ad.jvm.heap.min=1g
   ad.jvm.heap.max=1g
   ```

   c. For the ElasticSearch process, verify:

   ```
   ad.es.jvm.options.name=events-service.vmoptions
   ad.es.jvm.heap.min=8g
   ad.es.jvm.heap.max=8g
   ```

   > A production Elasticsearch installation requires 8 GB. For a demonstration installation, you can retain the default of 1 GB.

6. Save and close the `events-service-api-store.properties` file.
7. Install the Events Service as a Windows service, and open the command prompt as an Administrator:

> ⓘ Do not use PowerShell to do this.

   a. Set the `JAVA_HOME` environment variable so it specifies your Java installation directory. For example: `JAVA_HOME=C:\Zulu\zulu-8-jre`.
   b. Change the directory to `events-service`, and then enter:

      ```
      bin\events-service.exe service-install -p conf\events-service-api-store.properties --auto-start
      ```

      i. This command also installs Elasticsearch (even though it contains no explicit reference to Elasticsearch).
      ii. The optional `auto-start` flag causes the Events Service to be installed as an automatically started service; without this flag, the Events Service is installed as a manually started service.
      iii. For verbose installation and operation logging (useful for troubleshooting), include the `log-verbose` flag.

8. Locate the service name for the Events Service:

   ```
   bin\events-service.exe service-list
   ```

9. Open the Windows services and select the **AppDynamics Events Service Api Store *xxxxx*** Select **Start**.
10. Check the health of the new node and verify service status:
    a. If "Healthy" appears as the service status, then it indicates that the process is operating normally:

       ```
       bin\events-service.exe check-health -hp localhost:9081
       ```

    b. For the port, pass the administration port for the Events Service, 9081 by default.

       ```
       [appduser@controller-one events-service]$ bin/events-service.exe check-health -hp 192.168.33.22:
       9081
       [2015-12-09T18:30:45,342-08:00] HV000001: Hibernate Validator 5.0.2.Final
       [2015-12-09T18:30:45,956-08:00] Individual statuses below:
       [2015-12-09T18:30:45,956-08:00] [192.168.33.22:9081] status is [200 OK]
       [2015-12-09T18:30:45,956-08:00] Overall status Healthy
       ...
       ```

    c. If the service status does not display as `Overall status Healthy`, then the service is unhealthy. To correct it, you need to determine the correct key mappings between the following Events Service configuration and the Controller settings:
       i. `appdynamics.es.eum.key` should map to `ad.accountmanager.key.eum`
       ii. `appdynamics.saas.event.service.key` should map to `ad.accountmanager.key.controller`
       iii. `appdynamics.saas.event.service.key` should map to `ad.accountmanager.key.mds`
       iv. If the values of the key mappings are blank in the Admin Console, then use a UUID generator to create them.
       v. Use a UUID generator to create a value for `ad.accountmanager.key.ops`.
       vi. Use a UUID generator to create a different value for the following keys (you can use the same UUID for all three keys):
          1. `ad.accountmanager.key.slm`
          2. `ad.accountmanager.key.jf`
          3. `ad.accountmanager.key.service`
11. Configure the connections from the Analytics Agent, EUM Server, or Database Monitoring agents to the Events Service, as described in Connect to the Events Service.

Deploy an Events Service Cluster (Three Nodes)

The following steps describe how to deploy an Events Service cluster made up of three nodes, the minimum size of the Events Service cluster. These steps apply whether you are performing a new installation of a cluster or expanding a single node deployment into a three-node cluster. For information on expanding beyond a three-node cluster, see Adding Nodes to a Cluster.

> ⚠ Note that all services on Windows machines must be installed on the Enterprise Console host since the Enterprise Console does not support remote operations on Windows. Therefore, you cannot use the Enterprise Console GUI to deploy an Events Service cluster.

Before starting, review the topology notes in Events Service Deployment and make sure that all machines in the cluster meet the system requirements.

When ready, perform these steps on each of the three nodes.

1. Follow the steps for configuring a single node cluster in the 1-node installation above. *Additionally*, configure the following settings in the `conf\events-service-api-store.properties` file:

   a. Change the value of the `ad.es.node.minimum_master_nodes` property to 2:

```
ad.es.node.minimum_master_nodes=2
```

The setting specifies the minimum number of master-eligible instances that must be available in order to elect a new master. Since an Events Service cluster has three master nodes, this value should be two for a cluster.

b. Set the value of `ad.es.event.index.shards` to the number of nodes, in this case, three:

```
ad.es.event.index.shards=3
```

You do not need to change this value if it is already higher than the number of nodes.

c. Set the replication factor to 1 by changing the `ad.es.event.index.replicas` and `ad.es.metadata.replicas` properties, as follows:

```
ad.es.event.index.replicas=1
ad.es.event.index.hotLifespanDays=10
ad.es.metadata.replicas=1
```

d. For the unicast `hosts` property, add the hostname or IP address, along with the port 9300, for each node in the cluster:

```
ad.es.node.unicast.hosts=node1.example.com:9300,node2.example.com:9300,node3.example.com:9300
```

e. Change the publish host to the IP address or hostname of this machine. For example:

```
ad.es.node.network.publish.host=node2.example.com
```

f. Configure heap space for the Events Service and ElasticSearch processes, as follows:

i. To set the Events Service process heap size to 1 GB, for example, use the following properties:

```
ad.jvm.options.name=events-service.vmoptions
ad.jvm.heap.min=1g
ad.jvm.heap.max=1g
```

For the setting value, g indicates gigabyte (GB), and m indicates megabyte (MB).

ii. For the ElasticSearch process, the heap size should be set to half the size of the available RAM on the system, up to a maximum of 31 GB. To set the ElasticSearch process heap size to 8 GB, for example, set the properties as follows:

```
ad.es.jvm.options.name=events-service.vmoptions
ad.es.jvm.heap.min=8g
ad.es.jvm.heap.max=8g
```

For the setting value, g indicates gigabyte (GB), and m indicates megabyte (MB).

g. Save and close the file.

2. Install the Events Service as a Windows service:

```
bin\events-service.exe service-install -p conf\events-service-api-store.properties  --auto-start
```

The optional auto-start flag causes the Events Service to be installed as an automatically started service. If you do not include the flag, the Events Service is installed as a manually started service. An additional option, `log-verbose`, increases the verbosity of installation and operation logging, which is useful for troubleshooting.

3. Enter the following command to find the service name for the Events Service:

```
bin\events-service.exe service-list
```

4. Pass the service name returned by the service-list command as the -s parameter argument in the following command:

```
bin\events-service.exe service-start -s "<Name from service-list>"
```

Be sure to enclose the name in double-quotes.

5. Check the health of the new node using the following command. At least two nodes must be running before you run the command.

   ```
   bin\events-service.exe check-health -hp localhost:9081
   ```

   For the port, pass the administration port for the Events Service, 9081 by default. Verify that "Healthy" appears as the service status, indicating that the process is operating normally:

   ```
   [appduser@controller-one events-service]$ bin/events-service.exe check-health -hp 192.168.33.22:9081
   [2015-12-09T18:30:45,342-08:00] HV000001: Hibernate Validator 5.0.2.Final
   [2015-12-09T18:30:45,956-08:00] Individual statuses below:
   [2015-12-09T18:30:45,956-08:00] [192.168.33.22:9081] status is [200 OK]
   [2015-12-09T18:30:45,956-08:00] Overall status Healthy
   ...
   ```

6. Configure a load balancer to distribute traffic to the Events Service cluster, as described in Load Balance Events Service Traffic.
7. Connect the Controller and other clients—Analytics Agent, EUM Server, or Database Monitoring agents—to the Events Service, as described in Connect to the Events Service.

Expand an Events Service Cluster

The Events Service cluster is horizontally scalable. You can add nodes to an existing cluster without affecting or having to restart the existing nodes.

> ⚠ Note that all services on Windows machines must be installed on the Enterprise Console host since the Enterprise Console does not support remote operations on Windows. Therefore, you cannot use the Enterprise Console GUI to expand an Events Service cluster.

Before starting, prepare the new cluster machine. Verify system requirements and prepare the environment as described above.

For each node beyond the original three master nodes, download and configure the nodes as previously described. The configuration steps for any nodes added to the cluster after the initial three master nodes are as follows:

1. For each cluster nodes beyond the initial three master nodes, open the `conf\events-service-api-store.properties` for editing and make these configuration changes:

   a. Set the `ad.es.node.master` value to false:

      ```
      ad.es.node.master=false
      ```

   b. Set the `ad.es.node.minimum_master_nodes` value to 2.

      ```
      ad.es.node.minimum_master_nodes=2
      ```

   c. Set the value of `ad.es.event.index.shards` to the number of nodes in the cluster. You do not need to change this value if it is already higher than the number of nodes.

      ```
      ad.es.event.index.shards=<number_of_nodes>
      ```

       You do not need to change this value if it is already higher than the number of nodes.

   d. For the unicast hosts property, add the hostnames or IP addresses of all nodes in the cluster, including the node you are adding. For each node specify the ports on which the nodes communicate, 9300-9400. For example:

      ```
      ad.es.node.unicast.hosts=node1.example.com[9300-9400],node2.example.com[9300-9400],node3.example.
      com[9300-9400],node4.example.com[9300-9400]
      ```

      You do not need to reconfigure the unicast hosts settings for existing cluster members, as the new node can join the cluster dynamically.

   e. Change the publish host to the IP address or hostname of this machine. For example:

      ```
      ad.es.node.network.publish.host=node4.example.com
      ```

   f. Configure heap space for the Events Service and ElasticSearch processes, as follows:

      i. To set the Events Service process heap size to 1 GB, for example, use the following properties:

```
ad.jvm.options.name=events-service.vmoptions
ad.jvm.heap.min=1g
ad.jvm.heap.max=1g
```

For the setting value, g indicates gigabyte (GB), and m indicates megabyte (MB).

ii. For the ElasticSearch process, the heap size should be set to half the size of the available RAM on the system, up to a maximum of 31 GB. To set the ElasticSearch process heap size to 8 GB, set the properties as follows:

```
ad.es.jvm.options.name=events-service.vmoptions
ad.es.jvm.heap.min=8g
ad.es.jvm.heap.max=8g
```

For the setting value, g indicates gigabyte (GB), and `m` indicates megabyte (MB).

g. Save and close the file.

2. Install the Events Service as a Windows service:

   ```
   bin\events-service.exe service-install -p conf\events-service-api-store.properties --auto-start
   ```

   The optional auto-start flag causes the Events Service to be installed as an automatically started service. If you do not include the flag, the Events Service is installed as a manually started service. An additional option, `log-verbose`, increases the verbosity of installation and operation logging, which is useful for troubleshooting.
3. Enter the following command to find the service name for the Events Service:

   ```
   bin\events-service.exe service-list
   ```

4. Pass the service name returned by the service-list command as the -s parameter argument in the following command:

   ```
   bin\events-service.exe service-start -s "<Name from service-list>"
   ```

5. Check the health of the new node:

   ```
   bin\events-service.exe check-health -hp localhost:9081
   ```

   Note: At least two nodes must be running before you run the command.

   For the port, pass the administration port for the Events Service, 9081 by default. Verify that "Healthy" appears as the service status, indicating that the process is operating normally:

   ```
   [appduser@controller-one events-service]$ bin/events-service.bin check-health -hp 192.168.33.22:9081
   [2015-12-09T18:30:45,342-08:00] HV000001: Hibernate Validator 5.0.2.Final
   [2015-12-09T18:30:45,956-08:00] Individual statuses below:
   [2015-12-09T18:30:45,956-08:00] [192.168.33.22:9081] status is [200 OK]
   [2015-12-09T18:30:45,956-08:00] Overall status Healthy
   ...
   ```

6. Modify your load balancer rules to include the new cluster node. For more information, see Load Balance Events Service Traffic.

Start and Stop the Events Service

At installation, the Events Service is installed as a service and is left running upon completion of the installation. You can stop and stop it as a service or as a foreground process, as described here or by using the GUI.

Start and Stop as a Foreground Process

To start the Events Service as a foreground process, however, use the following command:

```
bin\events-service.exe start -p conf\events-service-api-store.properties
```

To stop the Events Service as a foreground process, use this command:

```
bin\events-service.exe stop
```

Stop and Start as a Windows Service

You can stop and start the Events Service as a Windows service from the Services Manager. You can also stop and start it using the `events-service.exe` tool, as here:

1. Enter the following command to find the service name for the Events Service:

   ```
   bin\events-service.exe service-list
   ```

2. Pass the service name returned by the `service-list` command as the `-s` parameter argument to the following command. Enclose the service name in double-quotes.

   ```
   bin\events-service.exe service-start -s "<Name from service-list>"
   ```

To stop the service, run this command:

```
bin\events-service.exe service-stop -s "<Name from service-list>"
```

Remove a Node

To remove a node that is not enabled for operation as a master node from the cluster, simply stop the Events Services on the node or remove the machine it runs on from the network.

Note the following guidelines:

- You cannot remove nodes such that the resulting cluster size is two
- A cluster that consists of three or more nodes can't be reduced in size to a single node Events Service.

After you remove a node, be sure to adjust your load balancer rules to remove the old cluster member. See Load Balance Events Service Traffic for more information.

If you are not using a load balancer with a cluster deployment, keep in mind that the connection settings for the first master node that reports to the Controller at installation time are written to the Controller setting that identifies the Events Service to the Controller. If you remove a master node in that case, check whether the removed master node is node identified as the Events Service destination URL in the Controller connection settings; adjust the setting if so. See Connect to the Events Service for more information.

To reconfigure an existing node to enable operation as a master node, or add a new node with the master option enabled:

```
ad.es.node.master=true
```

If reconfiguring a node, restart the node after changing the configuration.

Administer the Events Service

This component is available for on-premises deployments only. SaaS deployments are managed by AppDynamics.

This page describes how to manage Events Service with the Enterprise Console. All of these tasks can be performed on the GUI or CLI.

Start and Stop the Events Service

The Events Service is an internal data storage engine used by the Database Visibility module. To use Database Monitoring, you need to start the Events Service.

The Events Service is automatically started after you install it.

On Linux

Start the Events Service on Linux by running this command:

```
bin/platform-admin.sh submit-job --platform-name <platform_name> --service events-service --job start
```

Stop the Events Service by running this command:

```
bin/platform-admin.sh submit-job --platform-name <platform_name> --service events-service --job stop
```

On Windows

Start the Events Service on Windows by running this command:

```
bin/platform-admin.exe cli submit-job --platform-name <platform_name> --service events-service --job start
```

Stop the Events Service by running this command:

```
bin/platform-admin.exe cli submit-job --platform-name <platform_name> --service events-service --job stop
```

Monitor Cluster Node Health

It is important to carefully monitor the health of the Events Service cluster, for a new deployment, especially to monitor disk consumption.

You can check the status of the cluster from the Controller page in the Enterprise Console GUI or the Controller machine using this command:

```
bin/platform-admin.sh show-events-service-health
```

The output shows possible issues and the steps you need to take to resolve them. For example, if the available disk is low, the resolution is to add nodes to the cluster.

The following are the potential errors and remediation steps:

Error	Explanation	Remediation
Cluster out of capacity	If the heap size of any Events Service Java process exceeds 80% utilization	Add Events Service nodes
Disk size remaining drops below 30%	The disk size of the identified node dropped below 30%	Add Events Service nodes
Events Service is not reachable but the host is reachable	The Events Service process on the identified node is not functioning properly	Restart the node

Machine is not reachable	The machine may be down, disconnected, or suffering failure	Try restarting the machine; if it continues, the node may need to be removed from the cluster
Cluster needs restart	A condition has been identified that requires a cluster restart	Restart the cluster
Cluster size is 2	Events Service cluster requires more than two nodes	Add a node

Expand the Cluster

You can use the Enterprise Console GUI or CLI to horizontally scale the Events Service cluster on Linux. To grow your existing deployment to contend with an increased workload, simply add nodes.

Before starting, prepare the new cluster machine. Verify system requirements and prepare the environment as described in Events Service Requirements.

It is important for any new machine in the cluster to have the same SSH-enabled user account as existing cluster members.

Once you have prepared the system, run the command for adding nodes:

```
bin/platform-admin.sh add-events-service-nodes --hosts host1  host2  host3
```

Alternatively, pass the hostnames of the new node as a file.

```
bin/platform-admin.sh add-events-service-nodes --host-file=/home/appduser/hosts.txt
```

The file you pass to the command (`hosts.txt` in the example) should contain the internal DNS hostnames or IP addresses of the nodes to add. It does not need to list existing nodes in the cluster. These hosts should be part of the platform. For more information about how to add a host to the platform, see Administer the Enterprise Console.

Be sure to modify your load balancer rules to include the new cluster member in its routing rules. See Load Balance Events Service Traffic for more information.

Restart the Node

You can use the Enterprise Console GUI or CLI to restart your node.

Once you have prepared the system, run the command for restarting your node:

```
bin/platform-admin.sh submit-job --platform-name <platform_name> --service events-service --job restart-node --
args nodeActionHost=<node_name>
```

where `<node_name>` is the hostname of the node from the command:

```
bin/platform-admin.sh list-nodes --platform-name <platform_name> --service events-service
```

Restart the Cluster

You can use the Enterprise Console GUI or CLI to restart your cluster.

Once you have prepared the system, run the command for restarting your cluster nodes:

```
bin/platform-admin.sh submit-job --platform-name <platform_name> --service events-service -job restart-cluster
```

Remove or Replace a Node

The `uninstall-events-service` command removes the Events Service software and data from all cluster nodes. The Controller and Events Service share a database, so if you are uninstalling the Controller instance under which you ran the Enterprise Console to install the Events Service, you need to uninstall the Events Service with this command *before* you uninstall the Controller.

The `remove-events-service-node` command removes the Events Service software and data from a single node that you specify by hostname. You should only use this command if you have at least four nodes in your cluster. Removing an Events Service node from a three-node cluster is not supported. Identify the node to remove using the `--node` command line parameter.

After you remove a node, be sure to adjust your load balancer rules to remove the old cluster member. See Load Balance Events Service Traffic for more information.

If you are not using a load balancer with a cluster deployment, keep in mind that the connection settings for the first primary node that reports to the Controller at installation time are written to the Controller setting that identifies the Events Service to the Controller. If you remove a primary node in that case, check whether the removed primary node is node identified as the Events Service destination URL in the Controller connection settings (e.g., `appdynamics.on.premise.event.service.url`) and adjust the setting if so. See Connect to the Events Service for more information.

Note the following guidelines:

- You cannot remove nodes such that the resulting cluster size is two
- A cluster that consists of three or more nodes can't be reduced in size to a single node Events Service.

The `remove-events-service-node` command removes the Events Service software and data from a single node that you specify by hostname. You should only use this command if you have at least four nodes in your cluster. Removing an Events Service node from a three-node cluster is not supported. Identify the node to remove using the `--node` command line parameter.

This command removes the node specified in the argument:

```
bin/platform-admin.sh remove-events-service-node  --node 10.0.100.51
```

If you attempt to remove a primary node using the command shown above, the Enterprise Console notifies you that you are attempting to remove a primary node and cancels the operation. As indicated in the output, you can proceed to remove the primary node by rerunning the command with the `-f` force flag. When you remove a primary node, the cluster elects a new primary node from the existing data nodes. The election process may take a few seconds, during which new events cannot be processed. Be sure to perform this operation at a time when the impact of a brief interruption of service will be minimal.

If you have an unreachable node you would like to remove, but cannot due to the above restrictions, you can choose to replace it instead.

This command replaces the old node specified in the argument with the new node:

```
bin/platform-admin.sh submit-job --service events-service --job replace-node --args
originalNode=<old_host_address> newNode=<new_host_address>
```

⚠

- After removing the Events Service nodes from the cluster, you may observe that the value `appdynamics.es.eum.key` changed in the Controller `admin.jsp` and in the Events Service properties file, but not in the EUM properties file, `analytics.accountAccessKey`.
- Check if the key value changed in the Controller and the Events Service, then replace the key value with the EUM properties file: `analytics.accountAccessKey`.

Enable the Events Service to Use SSL

You can use the Enterprise Console CLI to enable the Events Service to use SSL. You will need a KeyStore file in JKS format, the password and the alias for the KeyStore. See Create a Certificate and Generate a CSR for detailed instructions for creating the Keystore.

1. Log in to the Enterprise Console:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

2. After successfully logging in, submit an `enable-ssl` job for the Events Service, providing the path to the KeyStore file, the KeyStore password, and KeyStore alias.

   ```
   bin/platform-admin.sh submit-job --platform-name <platform_name> --service events-service --job enable-
   ssl --args keystorePath=<path-to-keystore-jks-file> keystorePassword=<keystore_password>
   keystoreAlias=<keystore_alias>
   ```

3. Confirm that SSL has been enabled:

```
curl -k -v -X GET https://<events-service-domain>:9080/_ping
```

4. The output of the cURL command should show the TLS handshakes and the HTTP status 200:

```
...
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
...
< HTTP/1.1 200 OK
< Date: Fri, 10 May 2019 00:13:49 GMT
< X-Content-Security-Policy: default-src 'self'
...
```

Collect Events Service Logs

The Enterprise Console can collect logs from the nodes in the cluster. The following command retrieves node logs and bundles them, along with the Enterprise Console's own logs:

```
bin/platform-admin.sh retrieve-events-service-logs
```

When the command is finished, a ZIP file named `events-service.log.zip` is created in the location from which you ran the script. You can then extract the archive to troubleshoot or submit the archive for troubleshooting assistance to your AppDynamics representative. If the Enterprise Console failed to connect to one of the cluster nodes to retrieve logs for any reason, the connection error is written to a log file included in the archive.

Changing the SSH Key File

After initial installation, you may need up update the PEM file that gives the Enterprise Console access to node machines.

You can do so by creating the PEM file, as described in the discussion of configuring SSH passwordless login on Prepare the Events Service Host, and using the following command to install the new PEM file.

```
bin/platform-admin.sh set-user-credentials --ssh-key-file newkeyfile.pem
```

The change takes effect immediately.

Upgrade the Events Service

You can use the Enterprise Console to perform a rolling upgrade of the Events Service software on deployed nodes. For more information, see Upgrade the Events Service Using the Enterprise Console.

The general steps for upgrading an Events Service deployment are as follows:

1. Upgrade the Controller. (See Upgrade the Controller Using the Enterprise Console.)
2. Apply the upgrade to the Events Service nodes using the following command:

```
bin/platform-admin.sh upgrade-events-service
```

The Enterprise Console checks whether the Events Service is up to date relative to the current Controller version and, if not, performs the update.

Monitor Events Service Nodes with AppDynamics

You can use AppDynamics agents to monitor an Events Service node or cluster and generate diagnostic information for troubleshooting. The following steps outline the workflow.

1. Download the following agents from the AppDynamics Download Center:
   - Java Agent

- Machine Agent

2. Install both agents on each node in the cluster: first the Java Agent, then the Machine Agent.
3. On each node in the cluster, update the VM options for the Java Agent:
    a. Open the following file in a text editor:
    `<controller_home>/platform_admin/events-service/conf/events-service.vmoptions`
    b. Add the following lines to the end of the file:

    ```
    -javaagent:/opt/appdynamics/events-service/java_agent/ver4.5.0.0/javaagent.jar
    -Dappdynamics.agent.accountName=<account_name>
    -Dappdynamics.agent.applicationName=<events_service_app_name>
    -Dappdynamics.controller.hostName=<controller_host>
    -Dappdynamics.controller.port=443
    -Dappdynamics.controller.ssl.enabled=true
    -Dappdynamics.agent.nodeName=<events_service_node_name>
    -Dappdynamics.agent.tierName=<events_service_tier_name>
    ```

    Adjust the path to the Java Agent JAR, account name, and other values as appropriate.

    > ⓘ The business application name (`events_service_app_name`) and tier name (`events_service_tier_name`) should normally be the same for all nodes in an Events Service cluster, while each node must have a unique name (`events_service_node_name`).

    For more information on modeling applications in AppDynamics, see Application Modeling.
4. On each node in the cluster, define the Node Name and Tier Name used by the Machine Agent, as described in Independent Machine Agent Installation.

    > ⓘ The Node Name and Tier Name for each Machine Agent should be the same as the `events_service_node_name` and `events_service_tier_name` that you specify on each node.

5. Restart the Events Service on all nodes in the cluster: on the Controller host, navigate to `<controller_home>/platform_admin/events-service/` and enter the following command: `/bin/platform-admin.sh restart-events-service`
6. In the Controller UI, go to the Applications table and open the dashboard for the events_service_app_name application. (You might need to wait a few minutes for this application to appear as the Events Services on the nodes restart and begin sending data to the Controller.)
7. In the Application Dashboard, choose **Configure** > **Instrumentation**.
8. Select the `events_service_tier_name` tier and choose **Use Custom Configuration for this Tier**.
9. Under Custom Match Rules, create a new rule with the following attributes:

    - Entry Point = **Servlet**
    - Split Transactions Using Request Data = Use the first **4** segments in Transaction names

Update the Java Temporary Directory for Events Service Nodes

You can set up an optional override of the temporary Java directory for the Events Service by using the Platform Admin in the CLI. Complete the steps below to update the Java Temporary Directory for the Event Service.

> ⚠ After the Events Service settings update, the Events Service will require a restart initiated from the Platform Admin CLI or Enterprise Console GUI.

1. Run the command after installation:

```
/root/install/pa/platform-admin/bin/platform-admin.sh submit-job --service events-service --job
update_jvm_temp_dir --args jvmTempDir=/var/tmp
```

2. Make sure that the configuration is effective by running the following command:

```
/root/install/pa/platform-admin/bin/platform-admin.sh list-service-configurations --service events-service
```

3. Add the configurations for job `update_jvm_temp_dir`:

```
jvmTempDir (STRING): [/var/tmp]
```

4. To undo, use the following command:

```
/root/install/pa/platform-admin/bin/platform-admin.sh submit-job --service events-service --job
update_jvm_temp_dir
```

5. To add configurations for job `update_jvm_temp_dir` use the following command:

```
jvmTempDir (STRING): [null]
```

Load Balance Events Service Traffic

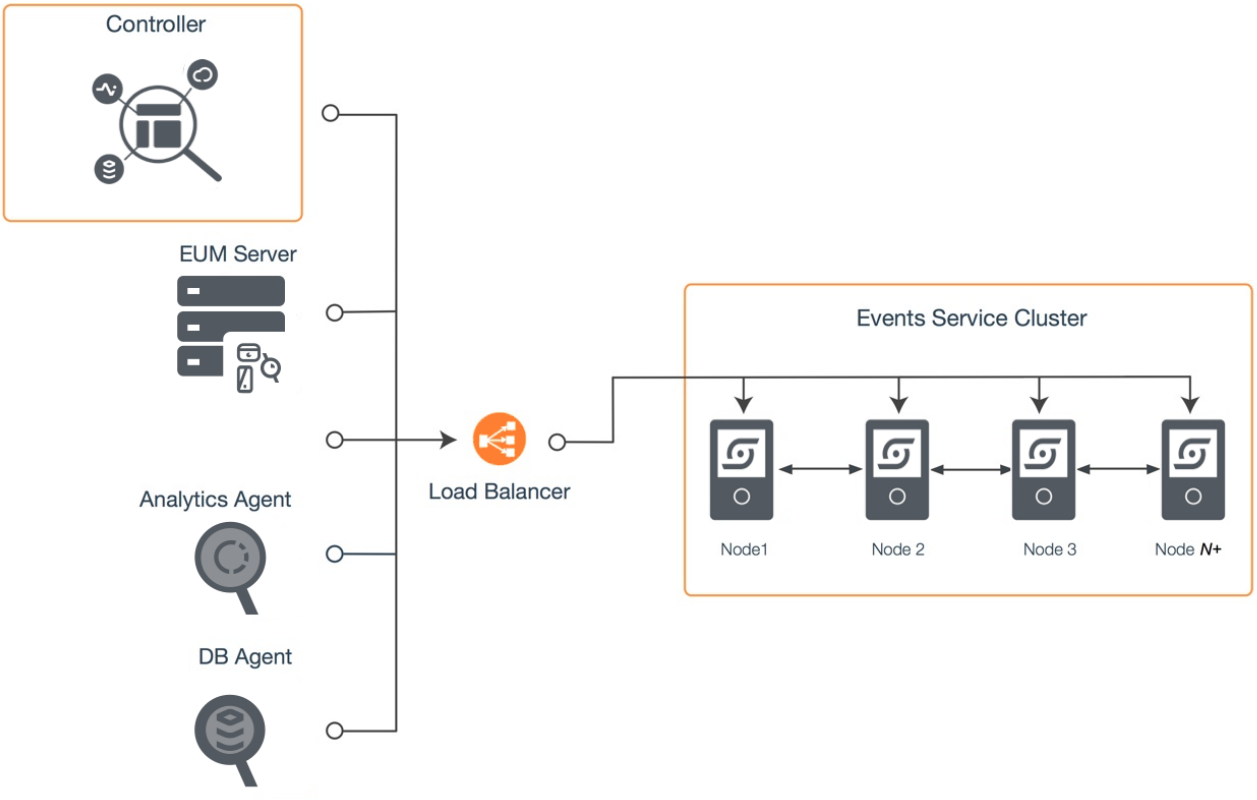
This page takes you through the sample configuration for a load balancer for the Events Service. It introduces you to the concepts and requirements around load balancing Events Service traffic.

Load Balancing Events Service Traffic Overview

To distribute load among the members of an Events Service cluster, you need to set up a load balancer. For a single node Events Service deployment, using a load balancer is optional but recommended, since it minimizes the work of scaling up to an Events Service cluster later.

To configure the load balancer, add the Events Service cluster members to a server pool to which the load balancer distributes traffic on a round-robin basis. Configure a routing rule for the primary port (9080 by default) of each Events Service node. Every member of the Events Service cluster, primary node or not, needs to be included in the routing rule. Keep in mind that increasing the size of the cluster will involve changes to the load balancer rules described here.

The following figure shows a sample deployment scenario. The load balancer forwards traffic for the Controller and any Events Service clients, Analytics Agents in this example.



About these Instructions

The following instructions describe how to install and configure a load balancer for the Events Service. The steps below provide two examples: load balancing with an Nginx and load balancing with HAProxy with SSL termination at the load balancer. The steps demonstrate commands in a CentOS 6.6 Linux operating system environment.

No two environments are exactly alike, so be sure to adapt the steps for your load balancer type, operating systems, and other site-specific requirements.

Nginx Sample Configuration

1. Install the Nginx software. You can install Nginx on most Linux distributions using the built-in package manager for your type of distribution, such as `apt-get` or `yum`. On a CentOS system, you can use `yum` as follows:

   ```
   sudo yum install epel-release
   sudo yum install nginx
   ```

2. Add the following configuration to a new file under the Nginx configuration directory, for example, to `/etc/nginx/conf.d/eventservice.conf`.

   ```
   upstream events-service-api {
      server 192.3.12.12:9080;
      server 192.3.12.13:9080;
      server 192.3.12.14:9080;
      server 192.3.12.15:9080;
      keepalive 15;
   }
   server {
      listen 9080;
      location / {
         proxy_pass http://events-service-api;
         proxy_http_version 1.1;
         proxy_set_header Connection "Keep-Alive";
         proxy_set_header Proxy-Connection "Keep-Alive";
      }
   }
   ```

   In the example, there's a single upstream context for the API-Store ports on the cluster members. By default, Nginx distributes traffic to the hosts on a round-robin basis.
3. Check the following operating system settings on the machine:
   - Permit incoming connections in the firewall built into the operating system, or disable the firewall if it is safe to do so. On CentOS 6.6, use the following command to insert the required configuration in `iptables`:

     ```
     sudo iptables -I INPUT -p tcp --dport 9080 -j ACCEPT
     ```

     To turn off the firewall, you can run these commands

     ```
     sudo service iptables save
     sudo service iptables stop
     sudo chkconfig iptables off
     ```

   - Disable if necessary selinux security enforcement by editing `/etc/selinux/config` and setting `SELINUX=disabled`. Restart the computer for this setting to take effect.
4. Start Nginx:

   ```
   sudo nginx
   ```

   Nginx starts and now direct traffic to the upstream servers. If you get errors regarding unknown directives, make sure you have the latest version of Nginx.

HA Proxy Sample Configuration: Terminating SSL at the Load Balancer

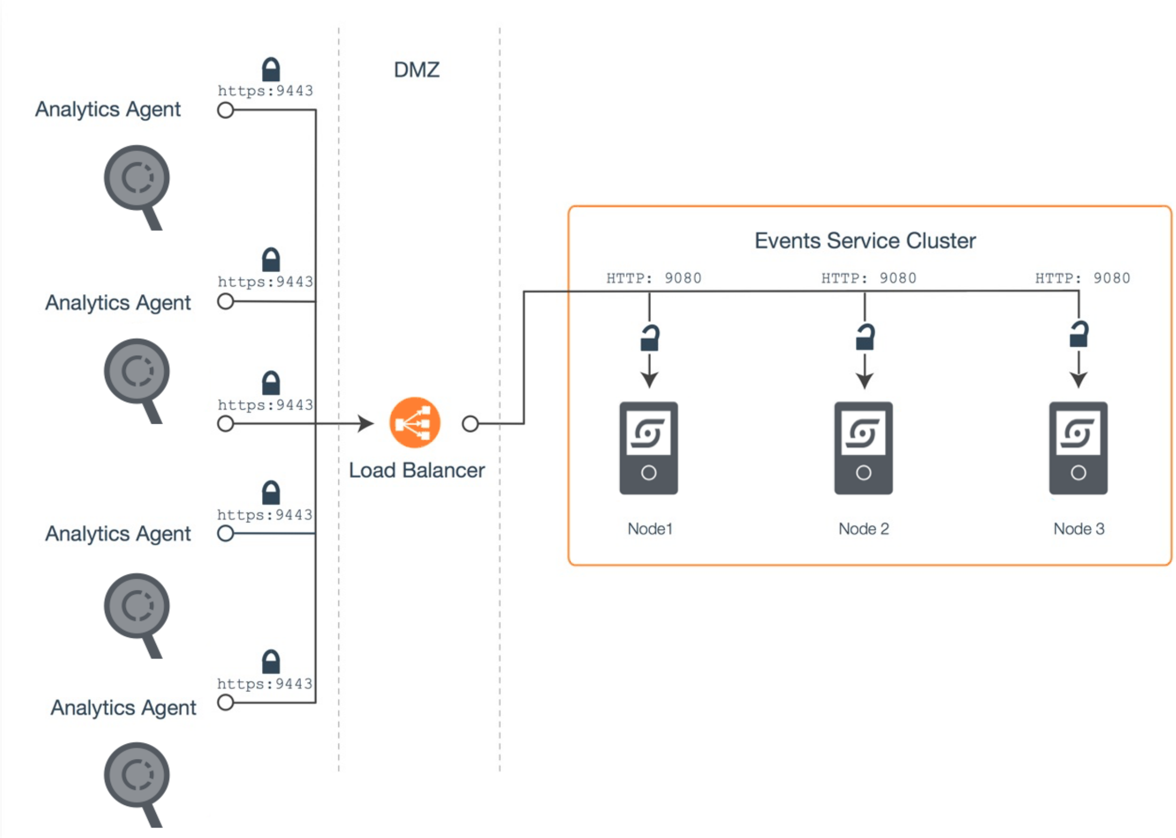
By terminating SSL at the load balancer in front of the Events Service cluster, you can relieve the Events Service machines from the processing burden of SSL. Since the connections between the load balancer and Events Service machines are not secured in this scenario, it is only suitable for deployments in which the load balancer and Events Service machines reside within an internal, secure network.

The following instructions describe how to set up SSL termination at the load balancer. These steps use HAProxy as the example load balancer. An overview of the steps are:

- Step 1: Install the HAProxy Software
- Step 2. Create the Security Certificate

- Step 3. Configure the Load Balancer
- Step 4: Configure the Agent
- Step 5: Configure the Controller

The following diagram shows a sample deployment reflected in the configuration steps:



Before Starting

To perform these steps, you need:

- Root access on the load balancer machine
- OpenSSL installed on the load balancer machine
- HAProxy software (minimum version HAProxy 1.5) on the load balancer machine

Step 1: Install the HAProxy Software

If not already installed, install HAProxy on the load balancer machine. The manner in which you install it depends on your operating system and the package manager it uses. If using `yum` package manager on Linux, for example, enter the following command:

```
sudo yum install haproxy
```

Step 2. Create the Security Certificate

The security certificate secures the connection between the load balancer and Events Service clients, including the Application Analytics Agent. You can use a self-signed certificate or a certificate signed by a certificate authority (CA) to secure the connection between the load balancer and clients. The following steps walk you through each scenario:

- Create a Self-Signed Certificate on the Load Balancer Machine
- Create a CA-Signed Certificate

For production use, AppDynamics strongly recommends the use of a certificate signed by a third-party CA or your own internal CA rather than a self-signed certificate.

Create a Self-Signed Certificate on the Load Balancer Machine

1. From the command line prompt on the Load Balancer machine, create a directory for the certificate resources and change to that directory:

```
sudo mkdir -p /etc/ssl/private
cd /etc/ssl/private/
```

2. Create the certificate by running the following command, replacing `<number_of_days>` with the number of days for which you want the certificate to be valid, such as 365 for a full year:

```
sudo openssl req -x509 -nodes -days <number_of_days> -newkey rsa:2048 -keyout ./events_service.key -out .
/events_service.crt
```

3. Respond to the prompts to create the certificate. For the Common Name, enter the hostname for the load balancer machine as identified by external DNS (that is, the hostname that agents will use to connect to the Events Service). This is the domain that will be associated with the certificate.
4. Put the certificate artifacts in a PEM file, as follows:

```
chmod 600 events_service.crt events_service.key
cat events_service.crt events_service.key > events_service.pem
chmod 600 events_service.pem
```

Create and Install a Certificate Signed by a Certificate Authority

1. From the command line prompt of the Load Balancer machine, create a directory for the certificate resources and change to that directory:

```
sudo mkdir -p /etc/ssl/private
cd /etc/ssl/private/
```

2. Generate a Certificate signing request (CSR) based on the private key. For example:

```
openssl req -new -sha256 -key /etc/ssl/private/events_service.key -out /etc/ssl/private/events_service.
csr
```

3. Submit the `events_service.csr` file to a third-party CA or your own internal CA for signing. When you receive the signed certificate, install it and the CA authority root certificate.
4. Depending on the format of the certificates returned to you by the Certificate Authority, you may need to put the certificate and key in PEM format, for example:

```
chmod 600 <ca_crt> events_service.key
cat <ca_crt> <intermediate_ca_crt_if_any> events_service.key > events_service.pem
chmod 600 events_service.pem
```

   In the command, replace `<ca_crt>` with the certificate returned to you by the Certificate Authority. Include any intermediate CA certs, if present, when creating the PEM file.

Step 3. Configure the Load Balancer

1. Open the HAProxy configuration file for editing, `/etc/haproxy/haproxy.cfg`.
2. Insert the following configuration at the end of the file. Replace the placeholder addresses with the host names or IP addresses of the cluster machines. The port should be the primary listening ports of the Events Service nodes.

```
     frontend events_service_frontend
      bind *:9443 ssl crt /etc/ssl/private/events_service.pem
      mode tcp
      reqadd X-Forwarded-Proto:\ https
      default_backend events_service_backend

      backend events_service_backend
      mode tcp
      balance roundrobin
        server node1 192.3.12.12:9080 check
        server node2 192.3.12.13:9080 check
        server node3 192.3.12.14:9080 check
```

3. Start the HAProxy load balancer:

```
sudo service haproxy restart
```

Step 4: Configure the Agent

Perform these steps on each machine on which the Analytics Agent runs.

1. Transfer a copy of the signed certificate, `events_service.crt`, to the home directory (denoted as `$HOME` in the instructions below) of the machine running the agent using Secure Copy (`scp`) or the file transfer method you prefer.
2. Copy the certificate file to the directory location of the trust store used by the agent:

```
cp $HOME/events_service.crt $JAVA_HOME/jre/lib/security/
```

3. Navigate to the directory and make a backup of the existing `cacerts.jks` file:

```
cd $JAVA_HOME/jre/lib/security/
cp cacerts.jks cacerts.jks.old
```

4. Import the certificate into the Java keystore:
    - If using a signed certificate, import the certificate as follows:

    ```
    keytool -import -trustcacerts -v -alias events_service -file /path/to/CA-cert.txt -keystore
    cacerts.jks
    ```

    - If using a self-signed cert, import the certificate as follows:

    ```
    keytool -import -v -alias events_service -file events_service.crt -keystore cacerts.jks
    ```

    When prompted, enter the password for the truststore (default is changeit) and enter yes when asked whether to trust this certificate.
5. Verify that the certificate is in the truststore:

```
keytool -list -keystore cacerts.jks -alias events_service
```

6. Navigate to the installation folder of the Analytics Agent and edit `conf/analytics-agent.properties` to change the value of the HTTP endpoint property:

```
http.event.endpoint=https://<External_DNS_hostname_Load_Balancer>:9080
```

7. Configure the following property names and pathways:

```
https.event.trustStorePath=<absolute path to trust store>
```

```
https.event.trustStorePassword=<base64 encoded password>
```

8. Start the Analytics Agent (or restart it, if it is already running).
9. Check the health of the agent. In a web browser, you can do so by going to the health check URL at `http://<analytics_agent_host>:9091/healthcheck?pretty=true`.

   If the agent is operating normally, the healthy field is set to true, as in the following example:

   ```
   "analytics-agent / Connection to https://<External_DNS_hostname_Load_Balancer>:9443/v1" :
   { "healthy" : true }
   ```

Step 5: Configure the Controller

If not already done, configure the connection from the Controller to the Events Service through the load balancer using a secure connection as well:

1. Transfer a copy of the signed certificate, `events_service.crt`, to the home directory (denoted as `$HOME` in the instructions below) of the machine running the Controller using Secure Copy (`scp`) or the file transfer method you prefer.
2. Navigate to the directory containing the Controller trust-store (as determined by the Controller startup parameter -Djavax.net.ssl.trustStore).
3. Make a backup of the existing `cacerts.jks` file:

   ```
   cp cacerts.jks cacerts.jks.old
   ```

4. Import the certificate into the Java keystore:

   - If using a signed certificate, import the certificate as follows:

     ```
     keytool -import -trustcacerts -v -alias <ca_cert_name> -file /path/to/CA-cert.txt -keystore
     cacerts.jks
     ```

   - If using a self-signed cert, import the certificate as follows:

     ```
     keytool -import -v -alias events_service -file events_service.crt -keystore cacerts.jks
     ```

   When prompted, enter the password for the truststore (default is changeit) and enter yes when asked whether to trust this certificate.
5. Verify that the certificate is in the truststore:

   ```
   keytool -list -keystore cacerts.jks -alias events_service
   ```

6. Restart the Controller.
7. From the Administration Console, search for the following Controller Setting: `appdynamics.on.premise.event.service.url`.
8. Set its value to the Load Balancer URL value: `https://<External_DNS_hostname_Load_Balancer>:9443/v1`

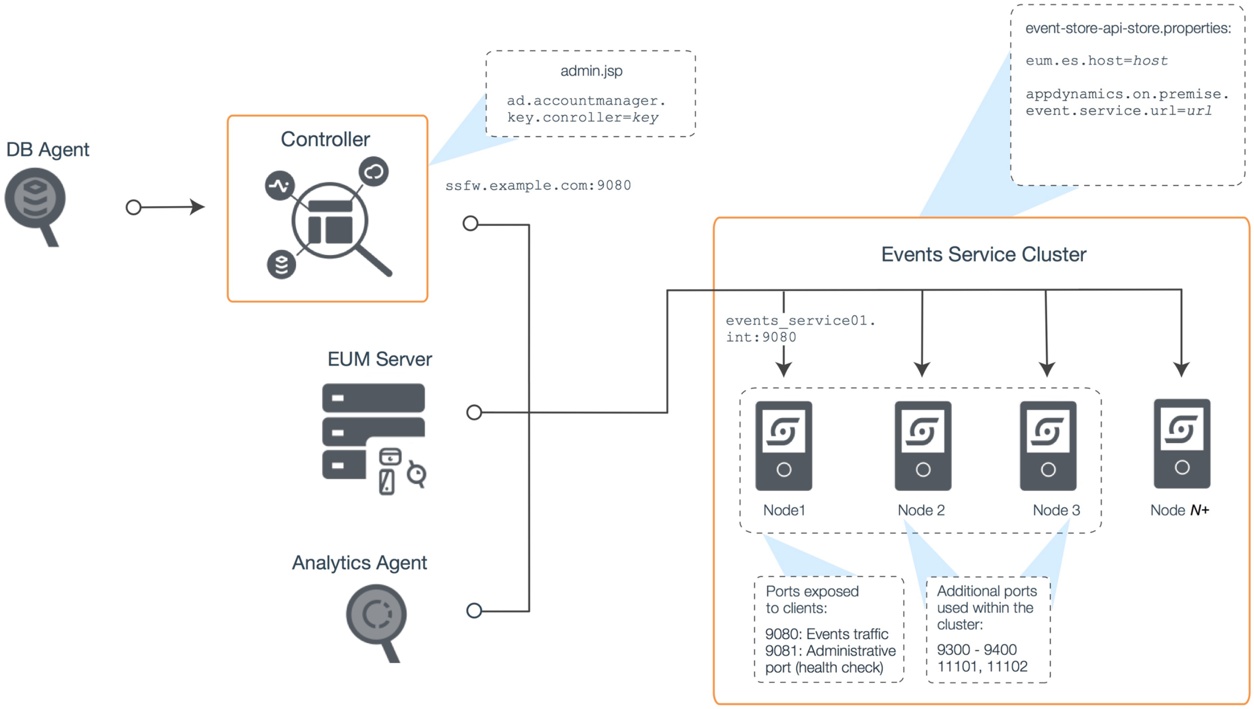
You can now verify that the Analytics UI is accessible and showing data.

Connect to the Events Service

This component is available for on-premises Controllers only. SaaS Controllers are managed by AppDynamics.

You must be connected to the AppDynamics Events Service to send data to it. AppDynamics uses API keys and connection URLs to establish connections between components.

You can configure these connection settings on the **Controller Settings** page of the **Admin Console** (see Access the Administration Console). The following sections describe the required connection settings for Database Visibility, Analytics, and End User Monitoring.



Database Visibility and Analytics Connection Settings

The Database Visibility module stores some of its data (such as wait state information and query information) in the Events Service. The Database Visibility module is the Database Monitoring product, which is already bundled with the controller. If you have both Analytics and Database Visibility installed, we recommend that they share an Events Service instance or cluster instance.

To connect these modules to the Events Service, open the **Admin Console** > **Controller Settings** and set values for the following properties:

- Set `appdynamics.on.premise.event.service.key` to the corresponding key in the properties file for Database Visibility.
- Set `appdynamics.on.premise.event.service.url` to the Events Service endpoint URL.
- Set `appdynamics.non.eum.events.use.on.premise.event.service` to `true`.

To set up the properties files required for Database Visibility:

1. Make note of the `appdynamics.on.premise.event.service.key` value from the **Admin Console**.
2. Open the file called `events-service-api.properties`, which you can find under the Events Service conf directory.
3. In the Events Service file, `events-service-api-store.properties`, ensure that the On-Premise Events key matches with the values of `ad.accountmanager.key.controller` and `ad.accountmanager.key.mds`
    a. See Configure the Events for details.
4. Stop and start the Analytics Service through the Enterprise Console or command line to pick up the new values.

End-User Monitoring Connection Settings

1. Open **Admin Console > Controller Settings**
2. Locate `eum.es.host` and verify it is set to the proper Events Service endpoint URL.
3. Navigate to the `<EUM_HOME>/eum-processor/bin/eum.properties` file
4. Set the following property values to connect the EUM to the Events Service:

   Sample

   ```
   analytics.enabled=true
   analytics.serverScheme=http
   analytics.serverHost=events.service.hostname
   analytics.port=9080
   analytics.accountAccessKey=1a59d1ac-4c35-4df1-9c5d-5fc191003441
   ```

5. Open the **Admin Console** > **Controller Settings**
6. Set the `eum.es.host` to the Events Service endpoint URL.

> The value of `appdynamics.es.eum.key` will automatically be set to the property `analytics.accountAccessKey` of the file `<EUM_HOME>/eum-processor/bin/eum.properties`.

Proxy Connection Settings

If you are connecting Database Visibility or Analytics through a proxy, you must set values for the following properties:

- `appdynamics.on.premise.event.service.proxy.host`
- `appdynamics.on.premise.event.service.proxy.port`
- `appdynamics.on.premise.event.service.proxy.user`
- `appdynamics.on.premise.event.service.proxy.password.file`
- `appdynamics.on.premise.event.service.proxy.use.ssl`

If you are connecting EUM through a proxy, you must set values for the following properties:

- `appdynamics.controller.http.proxyHost`
- `appdynamics.controller.http.proxyPort`
- `appdynamics.controller.http.proxyUser`
- `appdynamics.controller.http.proxyPasswordFile`

Back Up Events Service Data

Backing up Events Service data helps you to recover from hardware or another type of failure of an Events Service machine. A snapshot represents the backed up data for the entire Events Service cluster. In addition to using it for failure recovery, you can use a snapshot to migrate Events Service to a new instance.

The Events Service tool—`events-service.sh` for Linux and `events-service.exe` for Windows—includes commands for preparing the system for backing up with snapshots, generating a snapshot, and restoring from a snapshot, as described below.

The following instructions show sample commands for Linux. If using Windows, be sure to use the `events-service.exe` form of the executable rather than the `.sh` form, and adjust the sample directory paths as needed.

Prepare the File System

When planning your backup strategy, it is important to consider the storage location and frequency of backups. The system that will serve at the repository for snapshots must be able to handle high I/O demands in a performant manner. SSD-based storage is recommended. Also, ensure you have enough disk space on the repository system.

Only the first snapshot results in a full copy of the data. Each subsequent snapshot is incremental, applying only the changes since the last snapshot. Backing up frequently, therefore, does not result in substantially more storage overhead than backing up infrequently.

The Events Service includes tools for setting up the snapshot repository. It supports the following snapshot repository location types:

- A file system location that is shared among the Events Service nodes.
- An Amazon S3 bucket

After choosing and preparing the system that will host the snapshot, set up each Events Service node as follows:

1. If using FS, mount the shared filesystem at the default location for the backup repository, `<appd_home>/events-service/appdynamics-events-service-backup`. To change this backup location, set the the `ad.es.backupmanager.path.repo` setting in `conf/events-service-api-store.properties`. Keep in mind, however, that changing the properties file requires a restart of the Events Service node to have the change take effect.
2. Set up the repository on each node. Use the appropriate command for your repository type:
   - For shared file system:

     ```
     bin/events-service.sh snapshot-configure-fs -p conf/events-service-api-store.properties
     ```

   - For Amazon S3:

     ```
     bin/events-service.sh snapshot-configure-s3 -p conf/events-service-api-store.properties -bucket
     "s3-bucket-name"
     ```

     The `snapshot-configure-s3` command accepts additional optional arguments, including arguments for passing the access key and secret key for S3. Run `"bin/events-service.sh -h"` to view all options.

   Look for a message similar to the following to verify that the configuration succeeded:

   ```
   [2015-12-17T15:43:04,092-08:00] Successfully configured snapshot repository!
   ```

Create a Snapshot

After setting up the repository, you can generate a snapshot of the Events Service data. If you are backing up a cluster, you only need to run the command from one of the primary nodes. If you have more than one cluster, generate a snapshot for each one. Snapshots are cluster-specific.

Generate a snapshot:

```
bin/events-service.sh snapshot-run -p conf/events-service-api-store.properties
```

You can use this command to script regular backups based on your backup policy. The following output indicates that backup was successful:

```
Take snapshot request executed successfully. Snapshot itself may still be in progress.
```

To check the progress of the snapshot, use `snapshot-status`.

```
bin/events-service.sh snapshot-status -p conf/events-service-api-store.properties
```

If you don't specify a snapshot ID, the command gets the status for the most recent snapshot. You can use the `snapshot-list` command to see a list of available snapshots.

Restore from a Snapshot

By restoring a snapshot, you replace the data store configured for the Events Service with one that was previously saved as a snapshot. When you restore from a snapshot, you restore from a snapshot for a specific cluster. Run the command for each cluster

To restore a snapshot, use the `snapshot-restore` command, passing the properties file for the Events Service instance you are backing up. The following shows an example with sample output:

```
bin/events-service.sh snapshot-restore -p conf/events-service-api-store.properties
[2015-12-17T17:02:52,264-08:00] HV000001: Hibernate Validator 5.0.2.Final
[2015-12-17T17:02:52,811-08:00] Restore snapshot request executed successfully. Restore is now in progress. Use
the snapshot-status command to view the current restore status.
```

Check the status of the snapshot restore using the `snapshot-restore-status` command. For example:

```
bin/events-service.sh snapshot-restore-status -p conf/analytics-api-store.properties
[2017-01-03T14:37:46,647-08:00] HV000001: Hibernate Validator 5.2.2.Final
[2017-01-03T14:37:47,027-08:00] Restore is complete, your cluster should be fully functional!
```

You can restore a specific snapshot by passing the snapshot ID with the command. Otherwise, the most recent snapshot is restored.

```
bin/events-service.sh snapshot-restore -p conf/events-service-api-store.properties -id <snapshot_id>
```

You can use the `snapshot-list` command to get a list of snapshot IDs.

Migrating Events Service Data

In addition to data backup and recovery, you can use the snapshot utility to migrate data from one Events Service instance to another.

The target Events Service needs to be a fresh installation; that is, data in two different Events Service instances cannot be merged. Be sure to avoid configuring the Events Service URL with the new instance location in the Controller configuration until you have completed these steps.

To migrate Events Service data, follow these general steps:

1. Prepare the new Events Service nodes, as described above.
2. On each new Events Service node, mount the shared directory where the repository is located.
3. From a primary node in the new cluster, restore the snapshot by ID, as described above, passing the property file that defines the new cluster as the `-p` argument.
4. When finished, change the connection from the Controller to the Events Service and any Events Service clients, as described in Connect to the Events Service, to use the new instance.

Upgrade the Events Service

On this page:

- Before the Upgrade
- Upgrade the Events Service
- Verify the Upgrade

Related pages:

- Data Field Naming for Events Service 4.5.3 and Above

You can upgrade the Events Service either manually or by using the Enterprise Console.

You must upgrade manually if you:

- Do not use the Enterprise Console to deploy the Events Service
- Upgrade Events Service nodes hosted on remote Windows machines. The Enterprise Console does not support remote operations on Windows.

For on-premises deployments, 4.5.2 is the latest version of the Events Service. If you upgrade to a version of the Events Service other than the latest, run the Enterprise Console installer for the desired Events Service version.

Before the Upgrade

⚠ AppDynamics removed Search Guard from the on-premises Events Service version 4.5.2.20561. If your deployment requires Search Guard or a comparable feature, do not upgrade to this version of the Events Service.

AppDynamics will provide an alternative security feature with the next on-premises Events Service release.

See the Support Advisory for more details.

1. Download and install the new Enterprise Console.
2. Plan the order in which to upgrade platform components (not just Events Service).
3. Modify the `events-service-api-store.properties` file.
   - Replace the absolute path `APPLICATION_HOME` property in the file with an actual path.
   - This is required because while performing a discover and upgrade job, the Enterprise Console is unable to migrate the data directory for custom environment variables in the file. Failure to modify the file causes the upgraded Events Service to start with a new, blank data set.
4. Back up your `events-service.vmoptions` file.
   - This is required because the `events-service.vmoptions` file is not maintained when the Events Service is upgraded. After the upgrade completes, merge your backup copy of `events-service.vmoptions` into the new file.

ⓘ **Upgrading to a pre-4.1 Events Service**

To upgrade the Events Service software to a version earlier than 4.1, you must first manually upgrade the service to 4.1, and then use the Enterprise Console to discover the Events Service nodes.

Upgrade the Events Service

1. Run the upgrade Events Service command.
2. Discover the Events Service nodes using the Enterprise Console.

Verify the Upgrade

Once the upgrade completes:

1. The Events Service process should have restarted—verify its health status in the Enterprise Console GUI.
2. Merge your backup copy of the `events-service.vmoptions` file into the new copy of the file created by the upgrade.

Startup Script Paths

After an upgrade, you will find the Events Service startup script paths below:

```
<installDir>/appdynamics/events-service/processor/bin/events-service.sh
```

This may be different from their paths before the upgrade.

Upgrade the Events Service Using the Enterprise Console

This page describes how to upgrade a scaled-out Events Service on primarily Linux machines using the Enterprise Console.

Upgrade the Events Service Using GUI

If there is a Events Service upgrade available, you can begin the upgrade process either on the Custom Install or Events Service page in the GUI.

> You do not need to stop the Events Service before upgrading because the Enterprise Console does this for you.

After you upgrade the Events Service, upgrade the EUM server if it is part of your deployment. Then, upgrade the Controller.

Upgrade the Events Service from 4.1.x, 4.2.x, and 4.3.x to 4.4.x or Latest

> The Enterprise Console supports the installation of the Events Service on a Windows environment for a single node install. If you have several remote nodes, you will need to do a manual upgrade of the Events Service cluster and set the keys manually. See Connect to the Events Service

> The Enterprise Console manages the Controller and Events Service together, so their keys found in `events-service-api-store.properties` are set and synced to each other upon upgrade. However, we recommend confirming that the keys were synced correctly: `appdynamics.on.premise.event.service.key == ad.accountmanager.key.controller`

To upgrade the Events Service from 4.1.x, 4.2.x, and 4.3.x to 4.4.x or the latest version, you can use the Discover and Upgrade feature:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
3. Navigate to the **Install** homepage and click **Custom Install**.
4. Name the Platform:

   a. Enter a Name and the Installation Path for your platform.

   > The Installation Path is an absolute path under which all of the platform components are installed. The same path is used for all hosts added to the platform. Use a path which does not have any existing AppDynamics components installed under it. The path you choose must be writeable, i.e. the user who installed the Enterprise Console should have write permissions to that folder. Also, the same path should be writable on all of the hosts that the Enterprise Console manages.
   >
   > Example path: `/home/appduser/appdynamics/product`

5. Add a Host:

   > Note that all services on Windows machines must be installed on the Enterprise Console host since the Enterprise Console does not support remote operations on Windows. Therefore, you cannot add a host in a Windows Enterprise Console machine.

   a. Enter the host machine-related information: Host Name, Username, and Private Key. This is where the Events Service will be upgraded. Therefore, this needs to point to the host machine where the Events Service is currently up and running. For more information about how to add credentials and hosts, see Administer the Enterprise Console.
6. Click **Platforms**. Select the newly created platform and navigate to the **Events Service** page.
7. Discover Events Service:
   a. Select **Discover & Upgrade Events Service**.
   b. Select an available Target Version from the dropdown.

> The list is populated by versions that the Enterprise Console is aware of. This means that you can upgrade the Events Service to any intermediate version or to the latest version as long as the Enterprise Console installer has been run for those versions.

   c. Enter the Installation Directory.
   d. Enter the Events Service Host.

8. Click **Submit**.

The Enterprise Console will onboard the Events Service on the selected host machine to the application build. When the Enterprise Console discovers a component, it also checks to see if an upgrade is available and performs the upgrade. Plan for a downtime of the Events Service availability during this time. You can view the status of the upgrade job on the Jobs page.

Once the upgrade is complete, the Events Service Health status and related information can be accessed from the Events Service page.

> ⚠ After upgrading to 4.4.x or the latest, the commands to start and stop the Events Service change. See Administer the Events Service for more information.

Upgrade the Events Service from 4.4.x to Latest

> This procedure should be used to upgrade the Events Service when an existing platform in the Enterprise Console is already managing Events Service cluster nodes.

To upgrade the Events Service from 4.4.x to the latest version, you can use the Upgrade Events Service feature:

1. Check that you have fulfilled the Enterprise Console prerequisites before starting.
2. Upgrade the Enterprise Console to the latest version.
3. Open a browser and navigate to the GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   The default port is 9191.
4. Navigate to the **Events Service** page of the platform.
5. Select the Events Service host you would like to upgrade.
6. Click **Upgrade Events Service**.
7. Select an available Target Version from the dropdown list.

   > The list is populated by versions that the Enterprise Console is aware of. This means that you can upgrade the Events Service to any intermediate version or to the latest version as long as the Enterprise Console installer has been run for those versions.

8. Confirm the Upgrade.

Upgrade the Events Service Using CLI

If there is an Events Service upgrade available, you can begin the upgrade process using the application CLI.

> You do not need to stop the Events Service before upgrading because the Enterprise Console does this for you.

After you upgrade the Events Service, upgrade the EUM server if it is part of your deployment. Then, upgrade the Controller.

Upgrade the Events Service from 4.1.x, 4.2.x, and 4.3.x to 4.4.x or Latest

To upgrade the Events Service software from 4.1.x, 4.2.x, and 4.3.x to 4.4.x or the latest version, you will need to first download and install the Enterprise Console installer before performing the following steps:

1. Create a platform as follows:

```
bin/platform-admin.sh create-platform --name <platform_name> --installation-dir
<platform_installation_directory>
```

The installation directory is the directory where the Enterprise Console installs all platform components.

> ⓘ To avoid any failures, do not use the 4.3 or earlier Platform Admin installation directory. Instead, provide a new/empty directory.

2. Add the SSH key that the Enterprise Console will use to access and manage the Events Service hosts remotely. (See Create the SSH Key for more information):

   ```
   bin/platform-admin.sh add-credential --credential-name <name> --type ssh --user-name <username> --ssh-
   key-file <file path to the key file>
   ```

   `<file path to the key file>` is the private key for the Enterprise Console machine. The installation process deploys the keys to the Events Service hosts.

3. Add hosts to the platform, passing the credential you added to the platform:

   ```
   bin/platform-admin.sh add-hosts --hosts es_host_1 es_host_2 es_host_3 --credential <credential name>
   ```

4. Discover the Events Service nodes that are not yet integrated:

   ```
   bin/platform-admin.sh submit-job --service events-service --job discover-upgrade --platform-name
   <name_of_the_platform> --args destinationDirectory=<path_to_events_service> serviceActionHost=<es_host_1
   es_host_2 es_host_3>
   ```

   This command integrates the nodes into the Enterprise Console and also upgrades them. If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`

> ⚠ After upgrading to 4.4.x or the latest, the commands to start and stop the Events Service change. See Administer the Events Service for more information.

Upgrade the Events Service from 4.4.x to Latest

Upgrades from 4.4.x to the latest version can be performed on the Events Service page of the Enterprise Console or with the following commands:

1. Upgrade the Enterprise Console to the latest version.
2. Navigate to the `<Enterprise Console home directory>/platform-admin directory`.
3. If it has been more than one day since your last session, you will have to log in with the following command:

   ```
   bin/platform-admin.sh login --user-name <admin_username> --password <admin_password>
   ```

4. Apply the upgrade to the Events Service nodes with the following command:

   ```
   bin/platform-admin.sh submit-job --service events-service --job upgrade --platform-name
   <name_of_the_platform>
   ```

   If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`

Upgrade the Events Service Manually

This page describes how to manually upgrade an Events Service. This is useful for when you did not use the Platform Administration Application or the Enterprise Console to deploy the Events Service. The primary case for this would be for when you need to upgrade the Events Service nodes hosted on remote Windows machines.

Perform the Manual Upgrade

> ⚠ **Prerequisite**
>
> Java 1.8 is required for `events-service.exe` to work.

To upgrade the Events Service manually:

1. Download the Events Service distribution, `events-service.zip`, from the AppDynamics download site to the Events Service machine.
2. Stop the Events Service processes:

   ```
   bin/events-service.sh stop
   ```

3. Rename the existing Events Service directory, for example, to `events-service-backup`.
4. Unzip the Events Service distribution archive you downloaded to the location where you want the Events Service to run.
5. Migrate configuration changes from the properties files in the backup Events Service directory to the `conf/events-service-api-store.properties` file in the new Events Service directory. Depending on which type of deployment you are using, this involves inspecting and migrating settings from:
   - `events-service-all.properties`, or
   - `events-service-api-store.properties`

   > ⓘ If you are upgrading from 4.2 to 4.3.x or a later version you must edit the `events-service-api-store.properties` file by replacing the port ranges [9300-9400] with :9300.
   >
   > For example, in 4.2, the `events-service-api-store.properties` file looks like this:
   >
   > ```
   > ad.es.node.unicast.hosts=node1.example.com[9300-9400],node2.example.com[9300-9400],node3.example.com[9300-9400]
   > ```
   >
   > While in 4.3 or later, the file should look like this:
   >
   > ```
   > ad.es.node.unicast.hosts=node1.example.com:9300,node2.example.com:9300,node3.example.com:9300
   > ```

6. Configure the connection to the Events Service in the Controller, and get the Controller key for the Events Service configuration as follows:

   a. With the Controller running, open the Administration Console as the root user.
   b. In the Controller Settings page, search for `appdynamics.on.premise.event.service.url`.
   c. Replace the default value to the internal hostname for the Events Service machine and default Events Service listen port, 9080.
   d. Search for an additional setting, the one you will need to enable the connection from the Events Service to the Controller, `appdynamics.on.premise.event.service.key`.
   e. Copy the value of the property to your clipboard. You will need to configure this in the Events Service properties file next.
7. Configure the connection from the Events Service to the Controller:

   a. In a terminal, navigate to the `events-service` directory:

      ```
      cd events-service
      ```

   b. Open the `conf/events-service-api-store.properties` file for editing.
   c. Find the following property and replace `controller-key` with the copied key:

```
ad.accountmanager.key.controller=controller-key
```

8. Ensure that the following critical properties are configured appropriately in the `events-service-api-store.properties` file:
    - `ad.accountmanager.keyNamesCSV=EUM,CONTROLLER,MDS,OPS,SLM,JF`
    - `ad.accountmanager.key.eum=`
    - `ad.accountmanager.key.controller=`
    - `ad.accountmanager.key.mds=`
    - `ad.accountmanager.key.ops=`
    - `ad.accountmanager.key.slm=`
    - `ad.accountmanager.key.jf=`
    - `ad.accountmanager.key.service=`
    - `ad.jvm.heap.min=1g`
    - `ad.jvm.heap.max=1g`
    - `ad.es.jvm.heap.min=1g`
    - `ad.es.jvm.heap.max=1g`
9. Move `ad.es.node.unicast.hosts` property in `events-services-api-store.properties` while upgrading from 4.2 to 4.3.x or later.
10. Save and close the `events-service-api-store.properties` file.
11. Verify that the new Events Service home directory exists. The Event Service home directory is determined by the `ad.es.path.home` property in the property file used to start up the Events Service.
    If the directory does not exist, create it. For example, create the following directory: `/opt/appdynamics/events-service/appdynamics-events-service`
12. Move (do not copy) the old Events Service data directory to the new Events Service home directory. For example:

    ```
    mv /opt/appdynamics/events-service-backup/appdynamics-events-service/data /opt/appdynamics/events-service
    /appdynamics-events-service/
    ```

13. Restart the Events Service processes from the new directory:

    ```
    nohup bin/events-service.sh start -p conf/events-service-api-store.properties &
    ```

14. Check the health of the node.
    Windows
    `bin\events-service.exe check-health -hp localhost:9081`
    Linux
    `curl -XGET localhost:9081/healthcheck?pretty=true`

    Verify that "Healthy" appears as the service status, indicating that the process is operating normally:

    ```
    [appduser@controller-one events-service]$ bin/events-service.exe check-health -hp 192.168.33.22:9081
    [2015-12-09T18:30:45,342-08:00] HV000001: Hibernate Validator 5.0.2.Final
    [2015-12-09T18:30:45,956-08:00] Individual statuses below:
    [2015-12-09T18:30:45,956-08:00] [192.168.33.22:9081] status is [200 OK]
    [2015-12-09T18:30:45,956-08:00] Overall status Healthy
    ...
    ```

15. Configure the connections from the Analytics Agent, EUM Server, or Database Monitoring agents to the Events Service, as described in Connect to the Events Service.
    For information on performing these steps, see Install the Events Service on Windows.

> ⚠ If you upgrade to 4.4.x from 4.3.x or an earlier version, the commands to start and stop the Events Service change. See Administer the Events Service for more information.

Data Field Naming for Events Service 4.5.3 and Above

This page explains how to update data field names.

To use this procedure appropriately for your deployment, follow the guidelines below.

Deployment	Version	Required Action
SaaS	4.5.3 and later	Update data field names as described below.
On-premises	4.5.2 and older	No action is required, but AppDynamics strongly recommends that you update data field names to prepare for future Events Service releases.

Update Overview

Version 4.5.3 of the Events Service runs Elasticsearch 5.6, whereas previous Events Services run earlier versions of Elasticsearch. To upgrade to Events Service 4.5.3, you may need to rename some fields used in collecting transaction analytics, log analytics, or other types of data.

Review the requirements below and follow instructions where applicable. To understand the rationale for the changes, see More About Field Names.

Requirements

Do Not Use Empty Field Names

In pre-5.6 versions of Elasticsearch it was possible to create fields with empty names. This is no longer allowed. Empty field names now cause indexing errors.

Required action: Give alphanumeric names to any fields whose names are empty.

Do Not Use Dots in Field Names

In the past, some customers have used dots to separate name components in a semantically meaningful way. This is no longer recommended and may cause the upgrade to Events Service 4.5.3 to fail.

Strongly recommended action: Replace dots in field names with hyphens or underscores.

More About Field Names

This section discusses dotted field names, meaning field names with embedded periods ('`.`'), such as `a.b.c` or `transit.signals.yellow`.

Elasticsearch stores data in JSON documents whose structure is hierarchical. Dotted field names can be used to query into those JSON documents: the field name is treated as a path whose components are separated by dots. See https://www.elastic.co/guide/en/elasticsearch/reference/2.4/dots-in-names.html.

It can be impossible to know whether a dotted field name is intended as a path to a JSON element, or just a plain field name. To treat this problem in a consistent way, Elasticsearch, beginning with version 5.6, always automatically expands dotted field names into hierarchical JSON structures. Each dot creates another level nested lower in the hierarchy.

Events Service 4.5.3 attempts to gracefully handle dots in field names and allow the default behavior of Elasticsearch. However, in a few corner cases, Elasticsearch still fails to index events to which field names correspond. In these situations, the only recourse is to change the field names.

These corner cases can be avoided by following three rules:

1. Field names cannot contain multiple consecutive dots
2. Field names cannot start or end with dots
3. Field names cannot share prefixes

The rest of this section explains these rules.

Field Names Cannot Contain Multiple Consecutive Dots

Field names that contain multiple consecutive dots expand into structures where the name of some elements is the empty string. These are invalid as JSON objects. Note the empty names of the most deeply nested nodes in the following example:

```
"a.very.long.field.name.truncated.with.dots..." ->
"a": {
  "very": {
    "long": {
      "field": {
        "name": {
          "truncated": {
            "with": {
              "dots": {
                "": {
                 "": {
                  "": {
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Field Names Cannot Start or End with Dots

Field names that start or end with dots expand into structures where the name of some elements is the empty string. These are invalid as JSON objects. For example:

```
".a.b.c" ->
"": {
  "a": {
    "b": {
      "c": "value of field"
    }
  }
}


"a.b.c." ->
"a": {
  "b" : {
    "c": {
      "" : {
      }
    }
  }
}
```

Field Names Cannot Share Prefixes

When two or more field names have the same prefix, it becomes impossible to create valid JSON objects for them all.

Consider the following field names and values:

```
a.b = "alphabaker"

a.b.c = "alphabakercharlie"
```

Trying to share a prefix runs into trouble because:

- all the nodes that need to be created are text nodes, and
- some nodes need to be nested, but
- in JSON, text nodes are not allowed to contain other objects.

We'll demonstrate the problem by examining what happens when Elasticsearch tries to create the JSON objects for our example.

- The first field name results in "b" being mapped to a text node with the value "`alphabaker`."

```
"a": {
  "b": = "alphabaker"
}
```

- To expand the second name, Elasticsearch tries to map "c" to a text node with the value "`alphabakercharlie.`" This fails because "c" needs to be nested within "`b,`" which is a text node and cannot contain nested objects.

Use the Data Migration Tool

The Data Migration tool uses a collection of `Python3` files.

Install the Data Migration Tool

To install the Data Migration tool, download `migration_tool.zip`.

This example shows the unzipped structure:

```
tool/ main.py readme.txt requirements.txt src/ tool.json
```

Set Up the `python3` Environment

1. To verify if `python3` is installed on your system, enter:

   ```
   which python3
   ```

   > ⚠ If `python3` is not found, then install `python3` by entering: `sudo yum install python36 -y`

2. To verify if the `pip3` package manager is installed, enter:

   ```
   which pip3
   ```

   > ⚠ If the `pip3` package manager is not found, then see Installing Packages and enter: `sudopython3 -m pip install--upgrade pip setuptools wheel`

3. To install the libraries which run the migration python script, enter this command within the Data Migration tool directory:

   ```
   pip3 install -r requirements.txt --user
   ```

Configure the Data Migration Tool

Before you can run the migration script, you must configure it properly. All configuration is stored in the `tool/tool.json` file in JSON format. These sections provide guidance on how to configure each property.

Clusters

Clusters are defined as a collection of Events Service clusters. Each cluster has the following properties:

Properties	Description
`api_url`	URL pointing to one of the Events Service's API nodes or its load balancer.
`certificate_file`	Path to the PEM file.
`check_hostname`	(Optional) Indicates whether to check the hostname when verifying the certificate. Default is `true`.
`es_url`	URL pointing to one of the Elasticsearch's master nodes.
`es_url_internal`	Internal URL pointing to one of the Elasticsearch's master nodes. Used by other clusters for remote re-indexing.

`es_version`	Relates to the Elasticsearch version.
`keys`	Controller and OPS keys for Events Service. These keys are located in the `conf/events-service-api-store.properties` file.

This example defines Events Services: `es2`, `es6`, and `xpack_es6`.

> `es6` has SSL enabled, while `xpack_es6` has Elasticsearch X-Pack enabled.

```
{
  "clusters": {
    "es2": {
      "keys": {
        "CONTROLLER": "27410b11-296a-49e1-b2d2-d2371ab94d64",
        "OPS": "45c25bad-636c-432f-b0bd-f8ec428c8db4"
      },
      "api_url": "35.162.126.253:9080",
      "es_url": "35.162.126.253:9200",
      "es_url_internal": "172.31.12.185:9200",
      "es_version": 2
    },
    "es6": {
      "keys": {
        "CONTROLLER": "7db43bff-97d3-4d5e-828a-a2eacb693e07",
        "OPS": "ac7424d1-ae96-4e10-ad82-a2eca50db133"
      },
      "api_url": "https://34.209.245.68:9080",
      "certificate_file": "/Users/jun.zhai/es6.pem",
      "check_hostname": false,
      "es_url": "34.209.245.68:9200",
      "es_version": 6
    },
    "xpack_es6": {
      "keys": {
        "CONTROLLER": "07b055f8-a97b-4ccb-a239-2267d452c4ea",
        "OPS": "c582cc8a-f3bc-419f-a42a-7bb8adad05b8"
      },
      "api_url": "52.89.86.93:9080",
      "es_url": "http://elastic:1234@52.89.86.93:9200",
      "es_version": 6
    }
  }
}
```

Migration

This section describes the migration properties:

Properties	Description
`accounts`	(Optional) Specify which accounts to migrate. Defaults to everything in source Events Service.
`search_hits`	Maximum documents fetched in the Elasticsearch query. Default is 5000.
`remote_reindex_concurrency`	Maximum number of remote re-index tasks launched concurrently. Default is 4.
`remote_reindex_scroll_batch_size`	Batch size for remote reindex. Default is 8000. See Re-index API.
`reindex_task_polling_interval`	Frequency in seconds of how often to check the status of ongoing remote reindex tasks. Default is 60 seconds.
`starting_max_fields_per_index`	Maximum fields allowed when creating a new index. Default is set to 1000. The value should be same as the `ad.es.event.index.startingMaxFieldsPerIndex` in `conf/events-service-api-store.properties` file.

Migration Properties Examples:

Example 1: Migrates everything from source Events Service:

```
{
  "migration": {
    "search_hits": 5000,
    "remote_reindex_concurrency": 6,
    "remote_reindex_scroll_batch_size": 8000,
    "reindex_task_polling_interval": 60,
    "starting_max_fields_per_index": 1000
  }
}
```

Example 2: Migrate all event types in accounts, `customer15_9611293a-c56f-4c9a-aa11-9f6bffcb42ce`, `log_v1`, and `custom_event` event types in account `customer1_229f6fbf-b42f-4d66-a56b-a2324d8b169d`. This example does not migrate any other accounts.

```
{
  "migration": {
    "accounts": {
      "customer15_9611293a-c56f-4c9a-aa11-9f6bffcb42ce": [],
      "customer1_229f6fbf-b42f-4d66-a56b-a2324d8b169d": [
        "log_v1",
        "custom_event"
      ],
    },
    "search_hits": 5000,
    "remote_reindex_concurrency": 4,
    "remote_reindex_scroll_batch_size": 8000,
    "reindex_task_polling_interval": 60,
    "starting_max_fields_per_index": 1000
  }
}
```

Upgrade the Events Service to >= 20.9.0

Upgrade Procedure

To upgrade the On-premises Events Service to >= 20.9.0, AppDynamics recommends that you follow this procedure.

> The Events Service version 20.9.0 is packaged with the Enterprise Console version 21.2.4 or newer.

> If the incoming data load is heavy, you may expect delays in Step 3d during the data migration process.
>
> Data Migration
>
> - If you choose not to migrate data, then continue with Steps 1 through 3c, and do not complete Step 3d.
> - If you choose to migrate your data then data conflicts in may occur in Step 3d.
>
> Upgrade Completion
>
> - If the upgrade fails, then you should revert the Controller to the old cluster; data loss may occur.
> - If the upgrade succeeds, then you can delete the old cluster.

1. Prepare the machine instances:
   a. Identify a utility machine running Linux, to run the Enterprise Console and data migration scripts. There are no strict hardware requirements

      > Make sure that the older (source) Events Service cluster, new (target) Events Service cluster, and utility machine all reside on the same network.

   b. Ensure the target Events Service cluster has similar or better hardware than that of the source Events Service cluster. Ensure that the operating system (OS) on the new machine has these settings:
      i. Increase the file descriptors limit in the `/etc/security/limits.conf` file, and then reboot the machine. Use `ulimit -n` to verify the value.

      ```
      *                 soft    nofile          66000
      *                 hard    nofile          66000
      # End of file
      ```

      ii. Set `vm.max_map_count` in the `/etc/sysctl.conf` file, and then reboot the machine. Use `cat /proc/sys/vm/max_map_count` to verify the value.

      ```
      vm.max_map_count=262144
      ```

2. Set up the new staging cluster:
   a. Download and install the latest Enterprise Console, which includes Events Service 21.2.4. See AppDynamics Downloads and Platform Installation Quick Start.
   b. In the Enterprise Console, install the On-premises Events Service. See Install Events Service on Linux.
   c. Enable the load balancer. See Load Balance Events Service Traffic.
   d. To enable the Secure Socket Layer (SSL) in the Events Service, see Enable the Events Service to Use SSL.
   e. To configure the Events Service for data migration, complete this procedure:
      i. Enable HTTP port in `events-service-api-store.properties`:

      ```
      ad.es.node.http.enabled=true
      ```

      ii. Add the remote reindex `whitelist` in the `events-service-api-store.yml` file:

```
    - className: com.appdynamics.analytics.processor.elasticsearch.configuration.
ElasticsearchConfigManagerModule
      properties:
          nodeSettings:
              cluster.name: ${ad.es.cluster.name}
              ...
              indices.fielddata.cache.size: ${ad.es.fielddata.cache.size}
              reindex.remote.whitelist: "<IP address to one of the nodes in older cluster>:
9200"
```

iii. Restart the new Events Service cluster from the Enterprise Console.

3. Migrate the data:

   a. See Use the Data Migration Tool to install and configure the data migration script on the utility instance.
   b. Enter this command to migrate the metadata:

   ```
   python main.py migrate metadata es2 es6
   ```

   c. Navigate to the Controller **Admin** page and set your Controller to the new Events Service. See Connect to the Events Service.

   > ⚠ If you choose to migrate your data, then proceed to Step 3d.

   d. Enter this command to migrate data from the old cluster to the staging cluster:

   ```
   python main.py migrate data es2 es6
   ```

Uninstall the Events Service

You can remove an on-premises Events Service from individual nodes or all at once. An embedded Events Service is uninstalled along with the Controller.

Uninstall the Events Service with the Enterprise Console

You can uninstall the Events Service through the GUI on the Events Service page.

To do so through the CLI, you can use the `uninstall-events-service command`, which removes the Events Service software and data from all cluster nodes:

After uninstalling Events Service, the only trace of the Events Service remaining on the host may be a file named `orcha-modules.log`. It appears in the logs directory at the former installation root directory. To remove all traces of the Events Service, manually remove the log file after removing the Events Service with the Enterprise Console.

To uninstall the Events Service from a single node with the Enterprise Console, see the Removing a Node section on Administer the Events Service.

Uninstall the Events Service as a Windows Service

You can remove the Events Service as a Windows service after installation through the GUI. You can also use the following command to retain the Events Service on the machine.

To remove the Events Service as a Windows service:

1. Use the list service command to find the service name for the Events Service: `bin\events-service.exe service-list`

   Starting the ZooKeeper alone only brings up the process that manages index rollover. The Events Service node is not fully started until you start the API-Store process as well, as described next.
2. Use the name returned for the service as the -s parameter argument to the following command: `bin\events-service.exe service-uninstall -s "<Name from service-list>"`

   Be sure to enclose the name in double-quotes.

Synthetic Server Deployment

The Synthetic Server dispatches and processes requests and depends on Synthetic Agents for executing and reporting measurements.

The Synthetic Server receives synthetic job requests from the Controller and then the jobs are fetched from the Synthetic Services by the Synthetic Agents. Once the measurement results are received from the Synthetic Agents, the Synthetic Server stores, processes, and transmits the results to the EUM Server.

Installation Overview

To set up a complete on-premises Synthetic Server deployment, therefore, you need to:

1. Install the on-premises Controller or prepare an in-service Controller to work with the EUM Server.
2. Install the on-premises Events Service and configure it to work with your on-premises Controller.
3. Install the on-premises EUM Server and configure it to work with your Events Service and Controller.
4. Install the on-premises Synthetic Server and configure it to work with the EUM Server and the Controller.
5. Install and configure one or both types of Synthetic Agents.
6. Secure the Synthetic Server (recommended).
7. Monitor the Synthetic Server (recommended).

Synthetic Server Components

The on-premises Synthetic Server consists of the following three services:

- Synthetic Scheduler
- Synthetic Shepherd
- Synthetic Feeder Client

ⓘ This document also discusses the Synthetic Server Feeder, which communicates with the Synthetic Client Feeder, but is only a service of the SaaS Synthetic Server.

Synthetic Scheduler

The first service is the Synthetic Scheduler, which is a cron-like service that sends job requests at configured intervals. The Synthetic Scheduler handles the CRUD operations for jobs and manages the events generated for synthetic warnings and errors that occur in the measurement results. The Synthetic Scheduler also validates the beacons, triggers warning and error events if needed, and forwards the beacons to the EUM Server.

Synthetic Shepherd

The second service is Synthetic Shepherd. This service manages and dispatches jobs to the Synthetic Agents. In addition, the Synthetic Shepherd saves the measurement results to the filesystem and sends beacons containing the data to the Synthetic Scheduler.

Synthetic Feeder Client

The third service is the Synthetic Feeder Client that communicates with the SaaS Synthetic Feeder Server to access the Synthetic Hosted Agents. (If you are only deploying Synthetic Private Agents, you do not need to use the Synthetic Feeder Client.) These services use the WebSocket protocol to coordinate data transfer to your system without having to open any ports in the firewall.

Synthetic Agents

When deploying the on-premises Synthetic Server, you can deploy one or both Synthetic Agent types:

- Synthetic Hosted Agents - Synthetic Agents that are hosted and maintained by AppDynamics
- Synthetic Private Agents - Synthetic Agents that you install, configure, run, and maintain in your infrastructure

Comparison of the Synthetic Agent Types

The following table compares the two types of agents and provides the benefits and main use cases for both.

Synthetic Agent Type	Key Benefits / Use Cases
Synthetic Hosted Agent	• Access to a fleet of geographically distributed agents • Reduced ownership/resource costs: no hardware or cloud computing costs • Ease-of-use: no need to deploy/configure/manage agents • Scalability: Synthetic Hosted Agents are only deployed when needed, and more agents are readily available if the workload increases
Synthetic Private Agent	• Monitoring of internal sites and services that are not publicly accessible • Complete control over the agent configurations and environment

Overview of Installation and Configuration Steps

The following table provides an overview of the installation and configuration steps for each type of Synthetic Agent.

Synthetic Agent Type	Required Steps
Synthetic Hosted Agent	1. Acquire the license "Browser Synthetic User Monitoring - Hosted Agent - On-Premise". 2. Verify that the license has an HMAC key. 3. Configure SSL for the Synthetic Server (recommended). 4. Connect the on-premises Synthetic Server to the SaaS EUM API Server and SaaS Synthetic Server.
Synthetic Private Agent	1. Acquire one of the following licenses: • Browser Synthetic Monitoring - Private Agent - Per Location (on-premises) • Browser Synthetic Monitoring - Private Agent - Unlimited Locations (on-premises) 2. Install Synthetic Private Agents. 3. Connect the Synthetic Private Agent to the on-premises Synthetic Server . 4. Configure SSL for the Synthetic Server (recommended). 5. Start and maintain the Synthetic Private Agent.

Summary of Synthetic Server and Agents

The table below summarizes the function and ports used by each service and the Synthetic Agents:

Service/Agent	Functions	Protocol	Default Ports
Synthetic Scheduler	• Sends requests to execute jobs based on a configured frequency. • Validates the beacons containing the measurement results. • Handles the CRUD operations for jobs.	HTTP	12101
		HTTPS	12102
Synthetic Shepherd	• Registers the Synthetic Agents. • Creates and maintains the queue of all measurement requests. • Manages how the results arriving from the agents are processed, stored and forwarded. • Saves accompanying screenshots and generates thumbnails for each screenshot that go with the measurement.	HTTP	10101
		HTTPS	10102
Synthetic Feeder Server (Synthetic Hosted Agents)	• Deployed in SaaS. • Pushes screenshots and measurement results to the Synthetic Feeder-Client.	WebSocket (encrypted)	16001

Synthetic Feeder Client (Synthetic Hosted Agents)	- Deployed in on-premises. - Establishes a WebSocket connection with the SaaS Synthetic Server. - Sends the screenshots and measurement results it received from the SaaS Synthetic Server to the Synthetic Shepherd.	WebSocket (encrypted)	16101
Synthetic Agents	- Fetches jobs from the Synthetic Shepherd. - Uses WebDriver and Selenium to execute these jobs on browsers. - Registers with Synthetic Shepherd. - Uploads screenshots to Synthetic Shepherd. - Handles communication with Synthetic Shepherd.	N/A	The Synthetic Agents do not listen on any port. They only temporarily open internal random ports to fetch job requests from the Synthetic Shepherd and to send the measurement results of executed jobs to the Synthetic Shepherd.

Synthetic Service Data

The Synthetic Server stores data on the filesystem of the host machine and data in the EUM Server's MySQL database. The table below lists the types of data and the storage location.

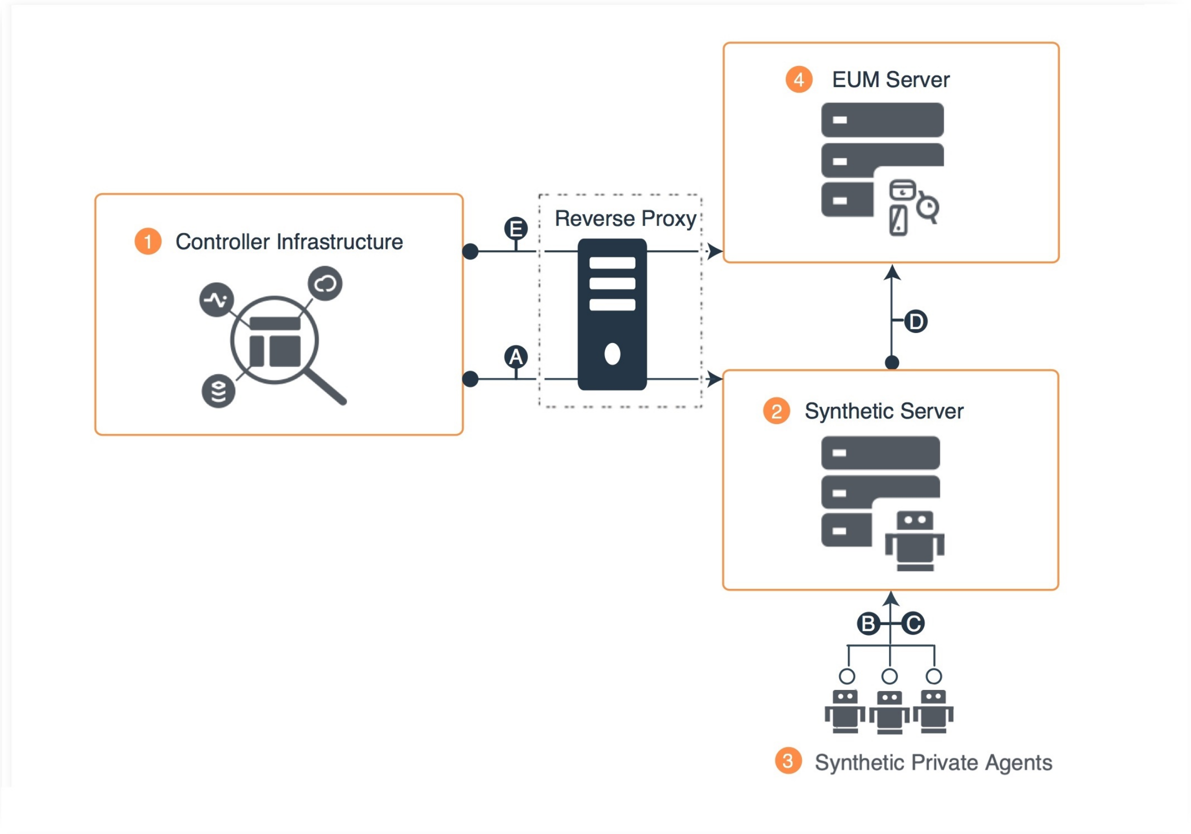
Data Storage Format	Data	Location of Data Storage
MySQL	- Information about the entire agent fleet. - Measurement request queues. - In-flight and archived measurements. - Schedules: a schedule is a configuration according to which the Synthetic Scheduler sends measurement requests to the Synthetic Shepherd. Those requests are queued until enough Synthetic Agents are available to process them, at which point they are dequeued and become measurements.	EUM Server's MySQL database
File System	- Resource snapshots - Script output - Measurement results - Screenshots	Host machine of the Synthetic Server

Synthetic Server Deployment Architecture

The following sections describe and provide diagrams of the different on-premises Synthetic Server deployments. The diagrams show the connections and data flow between the components of the deployments. For information about the other AppDynamics platform components, see Platform Components and Platform Connections.

Synthetic Private Agents Deployment

The following diagram shows the connections and data flow between the on-premises Synthetic Server and the EUM Server and Synthetic Private Agents.



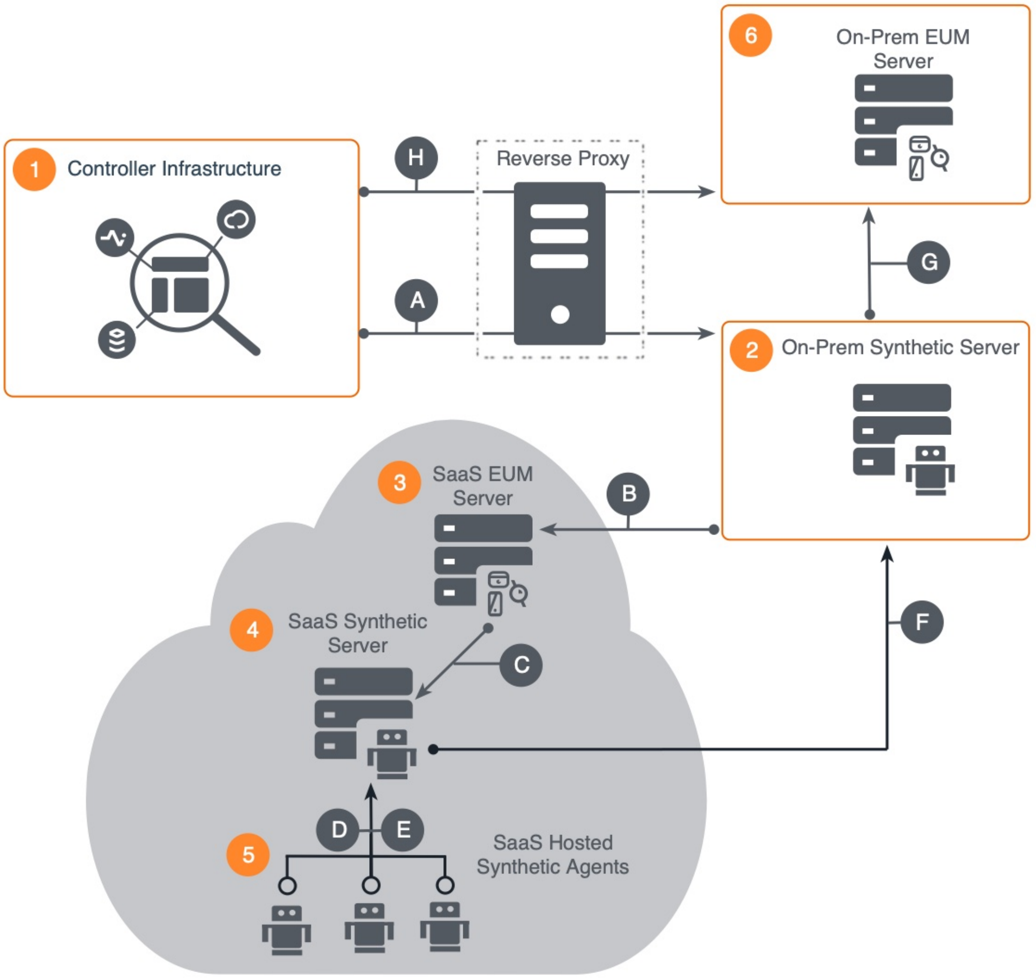
Synthetic Server Connections

The following table lists and describes the traffic flow between the Synthetic Server and the other components.

Connection	Source	Destination	Protocol	Default Port(s)
A	When a user creates a synthetic job, the 1 Controller sends a request for the job with its configured frequency to the 2 on-premises Synthetic Server. The synthetic jobs are then placed in a queue.	2 on-premises Synthetic Server	HTTP(S)	• 12101/12102 • 10101/10102
B	The 3 Synthetic Private Agent fetches the job requests from the Synthetic Server and then executes them on a browser using Selenium.	2 on-premises Synthetic Server	HTTP(S)	10101/10102
C	The 3 Synthetic Private Agent then sends the measurement results to the Synthetic Server.	2 on-premises Synthetic Server	HTTP(S)	10101/10102
D	The 2 on-premises Synthetic Server stores some data on file, and then processes and converts the data into a beacon, which is then transmitted to the 4 EUM Server through the EUM API. The Synthetic Server also writes data to the EUM Server's MySQL database.	4 on-premises EUM Server	HTTP(S) JDBC	7001/7002 3388
E	The 1 Controller polls the 4 EUM Server for the measurement results and displays them in the Synthetic Sessions.	4 on-premises EUM Server	HTTP(S)	7001/7002

Synthetic Hosted Agents Deployment

The following diagram shows the connections and data flow between the on-premises Synthetic Server, the SaaS EUM Server, the SaaS Synthetic Server, the Synthetic Hosted Agents, and the on-premises EUM Server.



Synthetic Server Connections

The following table lists and describes the traffic flow between the Synthetic Server and the other components.

Connection	Source	Destination	Protocol	Default Port(s)
A	When a user creates a synthetic job, the 1 Controller sends a request for the job with its configured frequency to the 2 on-premises Synthetic Server. The job requests are then placed in a queue.	2 on-premises Synthetic Server	HTTP(S)	12101 /12102
B	The 2 on-premises Synthetic Server sends the job requests to the 3 SaaS EUM Server.	3 SaaS EUM Server	HTTP(S)	7001/7002

C	The 3 SaaS EUM Server forwards the requests to the 4 SaaS Synthetic Server.	4 SaaS Synthetic Server	HTTP(S)	10001/10002
D	The 5 Synthetic Hosted Agents fetch the job requests from the 4 SaaS Synthetic Server and then executes them on a browser using Selenium.	4 SaaS Synthetic Server	WebSocket (encrypted)	16001
E	The 5 Synthetic Hosted Agents send the measurement results to the 4 SaaS Synthetic Server.	4 SaaS Synthetic Server	HTTP(S)	10001/10002
F	The 4 SaaS Synthetic Server Feeder sends the measurement results to the 2 on-premises Synthetic Client Feeder.	2 on-prem Synthetic Server	WebSocket (encrypted)	16101
G	The 2 on-premises Synthetic Server stores some data on file, and then processes and converts the data into a beacon, which is then transmitted to the 6 on-premises EUM Server through the EUM API. The on-prem Synthetic Server also writes data to the EUM Server's MySQL database.	6 on-premises EUM Server	HTTP(S)	7001/7002
			JDBC	3388
H	The Controller polls the 6 on-premises EUM Server for the measurement results and displays them in the Synthetic Sessions.	6 on-premises EUM Server	HTTP(S)	7001/7002

Synthetic Server Requirements

This page lists the Synthetic Server requirements, offers sizing guidance, and shows you how to modify the default settings.

AppDynamics Platform Requirements

To deploy the Synthetic Server, you need to install the following AppDynamics platforms:

Component	Minimum Version
Controller	20.3 and higher
Events Service	4.5.2 and higher
Synthetic Agent	• Synthetic Private Agent 20.3 or higher • Synthetic Hosted Agent 20.3 or higher

Certain Synthetic Server features—specifically, Synthetic Sessions Analytics, features of Application Analytics that extend the functionality of Synthetic Sessions—require access to the AppDynamics Events Service.

Synthetic Agent Requirements

The following table lists the requirements for deploying Synthetic Private Agents and Synthetic Hosted Agents.

Synthetic Agent	Requirements
Synthetic Private Agents	See Requirements for the Synthetic Private Agent.
Synthetic Hosted Agents	• Synthetic Hosted Agent license • AppDynamics Access (HMAC) Key (part of the license file for Synthetic Hosted Agent)

Hardware Requirements

These requirements assume that the Synthetic Server is installed on a separate machine. If other AppDynamics platforms are installed on the same machine, the requirements (particularly for memory) could vary greatly and require many more resources.

- Storage: 50 GB free disk space
- Memory: 8 GB memory
- CPU: 64-bit CPU with at least 2 cores
- Network bandwidth: 50 Mbps

NTP should be enabled on both the EUM Server host and the Controller machine. The machine clocks need to be able to synchronize.

Scaling Requirements

You are required to have one EUM account for each on-premises deployment of the Synthetic Server. The machine hosting the Synthetic Server should be able to support 100 concurrent Synthetic Agents or 10 locations with 10 Synthetic Agents per location.

If you need the Synthetic Server to support more than 100 concurrent Synthetic Agents, see Increase the Synthetic Agent Support.

Operating System Support

The Synthetic Server is supported on the following operating systems:

Linux (64 bit)
• RHEL 6.x and 7.x • CentOS 6 and 7 • Ubuntu 14 and 16 • SUSE 12

You can use the following file systems for machines that run Linux:

- ZFS
- EXT4
- XFS

> On-premises deployments on Linux are only supported on Intel architecture. Windows is not supported at this time.

Network Requirements

The network settings on the operating system need to be tuned for high-performance data transfers. Incorrectly tuned network settings can manifest themselves as stability issues.

The following command listing demonstrates tuning suggestions for Linux operating systems. As shown, AppDynamics recommends a TCP/FIN timeout setting of 10 seconds (the default is typically 60), the TCP connection `keepalive` time to 1800 seconds (reduced from 7200, typically), and disabling TCP window scale, TCP SACK, and TCP timestamps.

```
echo 5 > /proc/sys/net/ipv4/tcp_fin_timeout
echo 1800 >/proc/sys/net/ipv4/tcp_keepalive_time
echo 0 >/proc/sys/net/ipv4/tcp_window_scaling
echo 0 >/proc/sys/net/ipv4/tcp_sack
echo 0 >/proc/sys/net/ipv4/tcp_timestamps
```

The commands demonstrate how to configure the network settings in the `/proc` system. To ensure the settings persist across system reboots, be sure to configure the equivalent settings in the `etc/sysctl.conf` or the network stack configuration file appropriate for your operating system.

Software Requirements

The Synthetic Server requires the following software to run and function correctly. You are *required* to have outbound internet access to install Python, `pip`, and flake8.

Software	Required Version	Function
Java	8	The Synthetic Server requires JDK 8 to run services such as Synthetic Scheduler and Synthetic Shepherd. You need to set the environmental variable `JAVA_HOME` to the home directory of the JDK.
Python	2.7	The Synthetic Server relies on Python to validate scripts.

`pip`	9+	Python uses `pip` to install software. For example, `pip` could be used on some Linux distributions to install `flake8`, a Python utility used to lint scripts.
		If the machine where you're installing the Synthetic Server does *not* have Internet access, run the following steps to fetch and install `flake8`:
		1. From a machine with internet access and `pip` installed:
		a. Create a directory for the `flake8` library:
		`mkdir ~/flake8`
		b. Download the `flake8` package:
		c. Zip and tar the `flake8` package:
		`tar cvfz flake8.tgz ~/flake8`
		d. Copy `flake8.tgz` to the `$HOME` directory of the host machine of the Synthetic Server.
		2. From the host of the Synthetic Server that has no internet access, but does have `pip` installed:
		a. Unzip and extract the `flake8.tgz` file:
		`tar xvfz flake8.tgz ~/flake8`
		b. Change to the `flake8` directory.
		c. Install the `flake8` library with `pip` with the following command, replacing `<version>` with the correct version.

libaio	N/A	The Synthetic Server requires the `libaio` library to be on the system. This library facilitates asynchronous I/O operations on the system. See How to Install libaio for instructions.

How to Install libaio

Install `libaio` on the host machine if it does not already have it installed. You may require outbound internet access if you don't have a locally hosted repository.

The following table provides instructions on how to install `libaio` for some common flavors of the Linux operating system. Note, if you have a NUMA based architecture, then you are required to install the `numactl` package.

Linux Flavor	Command
Red Hat and CentOS	Use `yum` to install the library, such as: • `yum install libaio` • `yum install numactl`
Fedora	Install the library RPM from the Fedora website: • `yum install libaio` • `yum install numactl`
Ubuntu	Use apt-get, such as: • `sudo apt-get install libaio1` • `sudo apt-get install numactl`
Debian	Use a package manager such as APT to install the library (as described for the Ubuntu instructions above).

Install the Synthetic Server

You run the Synthetic Server installer from the command line. The installer relies on the`inputs.groovy` file to configure the network connections to the on-premises EUM Server, and if you are using Synthetic Hosted Agents, to the SaaS EUM Server and SaaS Synthetic Server as well.

To install the Synthetic Server, follow the steps below:

Prepare for the Installation

Before starting the installation, verify that you have:

- Successfully deployed the Controller, EUM Server, and the Events Service.
- Downloaded the Synthetic Server installer package from the AppDynamics Download Center. The installer package will be listed on the Downloads site as "Synthetic Server (zip)".

Grant Access to the EUM Server MySQL Database

The Synthetic Server installer modifies the EUM MySQL database schema and the Synthetic Server stores data in the EUM MySQL database. Thus, you will need to grant MySQL users from the machine hosting the Synthetic Server privilege to the EUM Server's MySQL database.

1. Log on to the machine where the EUM MySQL database is located.
2. Connect to the MySQL Server with the EUM Server database. For example, if you are using the default EUM MySQL database, do the following:
    a. Change to `<installDir>/AppDynamics/EUM`.
    b. Connect to the EUM MySQL database:

    ```
    mysql/bin/mysql -u root -h <eum_server_hostname> -S <eum_server_mysql_sock> -P
    <eum_server_mysql_port> -p
    ```

3. From the MySQL monitor, grant privileges to the MySQL user `root` of the Synthetic Server machine. The installer will use the MySQL user `root` t o update the EUM database schema. Be sure to replace `<on-prem-synthetic_server_hostname>` with the URL to your Synthetic Server.

    ```
    mysql> GRANT ALL PRIVILEGES ON eum_db.* TO 'root'@'<on-prem_synthetic_server_hostname>' IDENTIFIED BY
    <db-root-password>;;
    ```

    > The MySQL `root` user from the Synthetic Server is not related to the Linux user account that is installing the Synthetic Server. For example, the Linux user account `ubuntu` can run the installer, but the installer will use the MySQL user `root` when connecting to the EUM Server MySQL database to update the database schema.

4. You will also need to grant access to the MySQL user `eum_user` to write data to the EUM database (`eum_db`). Be sure to replace `<on-prem-synthetic_server_hostname>` with the URL to your Synthetic Server.

    ```
    mysql> GRANT ALL PRIVILEGES ON eum_db.* TO 'eum_user'@'<on-prem_synthetic_server_hostname>' IDENTIFIED
    BY <db_eum_user_password>;
    ```

5. Set the password for the MySQL user `root.` The password should be the same as the one specified by the `db_root_pwd` in the `inputs.groovy` file.

    ```
    mysql> SET PASSWORD FOR 'root'@'<on-prem-synthetic_server_hostname>' = PASSWORD('<root_password>');
    ```

6. Confirm that you have granted permission for `eum_user` and `root`:

```
show grants for eum_user@<on-prem_synthetic_server_hostname>;
show grants for root@<on-prem_synthetic_server_hostname>;
```

Unzip the Synthetic Server Installer Package

1. Copy the Synthetic Server installer package (`appdynamics-synthetic-server-<version>.zip`) to the machine that will be hosting the Synthetic Server.
2. Create a directory for storing the Synthetic Server installer, such as `synthetic-server`.
3. Move the Synthetic Server installer package to the directory you created.
4. Change to the directory you created for storing the Synthetic Server installer.
5. Unzip the Synthetic Server installer package.

Configure the Installation

1. From a command prompt, navigate to the directory where you unzipped the Synthetic Server installer package.
2. Copy the sample configuration file: `cp inputs.groovy.sample inputs.groovy`
3. Edit the file `inputs.groovy` and make changes to the properties listed below:

Property	Change to Make	Description
`db_host`	Assign the URL to the machine hosting your on-premises EUM Server to the `db_host` property.	The public DNS to the machine hosting the EUM Server.
`db_port`	Change the value to "3388". This is the default port for the EUM Server's MySQL database.	The port that the EUM Server's MySQL database is listening on.
`db_username`	Change the value to `eum_user`. This is the default user for the EUM Server.	The MySQL user that accesses the EUM Server's MySQL database.
`db_password`	Assign the password for the MySQL user `eum_user` to remotely access the EUM Server's MySQL database.	The password that you set for the user that is specified by `db_username`. The value of `db_username` should be `eum_user`.
`collector_host`	Assign the public DNS to the machine hosting your on-premises EUM Server to the `collector_host` property.	The public DNS to the machine hosting the EUM Server.
`collector_port`	Confirm that the value is "7001". This is the default port of the EUM Server.	The port that the EUM API Server and EUM Collector are listening on. The default is '7001'.
`key_store_password`	Assign the key store password you set when installing your on-premises EUM Server to the `key_store_password` property.	The key store password you set during the installation of the EUM Server.
`localFileStoreRootPath`	Assign a file path where you want the Synthetic Server to store data to the `localFileStoreRootPath` property. The Synthetic Server must be able to read and write to the path and the files in the path.	The path where the Synthetic Server stores data such as the measurement results and the screenshots.

Run the Installer

From the root directory of the installer, run the following command as the `root` user.

```
unix/deploy.sh install
```

In the output from the `install` command, you should see the log of completed tasks similar to the following:

```
Task [facts for localhost] completed executing in [274] ms.
Task [Create the encryption directory] completed executing in [78] ms.
Task [Create keystore for encryption] completed executing in [796] ms.
Task [Create the encrypted password] completed executing in [566] ms.
Task [Obfuscate the key store password] completed executing in [397] ms.
Task [Read created password] completed executing in [46] ms.
Task [Read the obfuscated key store password] completed executing in [43] ms.
Task [Change configurations for the shepherd and scheduler conf] completed executing in [81] ms.
Task [Read created password] completed executing in [24] ms.
Task [Read the obfuscated key store password] completed executing in [26] ms.
Task [Change configurations for the shepherd and scheduler conf] completed executing in [76] ms.
Task [Read created password] completed executing in [29] ms.
Task [Read the obfuscated key store password] completed executing in [20] ms.
Task [Change configurations for the shepherd and scheduler conf] completed executing in [31] ms.
Task [Delete the encryption directory] completed executing in [47] ms.
Task [Change configurations for the liquibase properties file] completed executing in [26] ms.
Task [Update schema of SQL DB to include synthetic schema] completed executing in [2412] ms.
Task [Install flake8 for script linting] completed executing in [1671] ms.
Task [Start the synthetic services] completed executing in [67] ms.
```

Verify the Installation Was Successful

1. Confirm that the Synthetic Server is running:

   ```
   ps aux | grep synthetic-processor
   ```

   > ⓘ If you have `jps` installed, you can also just run it to verify the Synthetic Server are running.

2. Verify that the Synthetic Scheduler and Synthetic Shepherd are listening on the default ports:

   ```
    netstat -lan | grep "1[0,2,6]10[1,2]"
   ```

3. With `mysql` installed on the Synthetic Server machine, you can verify that the Synthetic Server machine can connect to the EUM Server MySQL `eum_db` database:

   ```
   mysql -h <eum_server_instance> -P 3338 -D eum_db -u eum_user -p
   ```

4. If you cannot connect to the EUM Server MySQL database, return to Grant Privileges to the EUM Server MySQL Database and complete the steps again.

Perform Post-installation Tasks

After installing the Synthetic Server, perform the following additional post-installation tasks:

1. Configure the Controller for the Synthetic Server
2. Install Synthetic Private Agents (Optional)
3. Make configurations to use one or both of the Synthetic Agents:
   a. Synthetic Private Agent
   b. Synthetic Hosted Agent
4. Secure the Synthetic Server (Recommended)
5. Monitor the Synthetic Server (Recommended)

Configure the Controller for the Synthetic Server

Related pages:

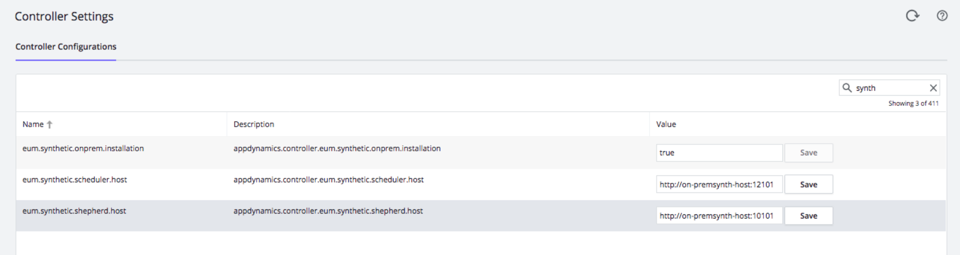
- Access the Administration Console
- Configure the EUM Server

For the Synthetic Server to function correctly, you need to set the URLs and ports of Synthetic Scheduler and Synthetic Shepherd in the Controller Administration Console.

1. Navigate to the Controller Administration Console: `http://<hostname>:<port>/controller/admin.jsp`
2. Click **Controller Settings**.
3. From the **Controller Configurations** pane, enter the correct values for the properties given in the table below. If no protocol is specified, the protocol defaults to `https://`.

Controller Configuration	Description	Example Value
`eum.synthetic.onprem.installation`	The flag for enabling on-premises Synthetic Server. This should be `true`.	`true`
`eum.synthetic.scheduler.host`	The URL and port to the Synthetic Scheduler on the machine hosting the Synthetic Server. The default port is 12101.	`http://<synthetic-server-domain>:12101`
`eum.synthetic.shepherd.host`	The URL and port to the Synthetic Shepherd on the machine hosting the Synthetic Server. The default port is 10101.	`http://<synthetic-server-domain>:10101`

4. Your configurations for the Synthetic Server should be similar to those in the screenshot of the **Controller Configurations** pane below.



Controller Settings

Controller Configurations

synth

Showing 3 of 411

Name ↑	Description	Value
eum.synthetic.onprem.installation	appdynamics.controller.eum.synthetic.onprem.installation	true
eum.synthetic.scheduler.host	appdynamics.controller.eum.synthetic.scheduler.host	http://on-premsynth-host:12101
eum.synthetic.shepherd.host	appdynamics.controller.eum.synthetic.shepherd.host	http://on-premsynth-host:10101

Connect the Synthetic Private Agents to the Synthetic Server

The information and instructions below are intended for On-Premise deployments only. SaaS deployments are managed by AppDynamics.

Related pages:

- Install the Synthetic Private Agent

The Synthetic Private Agent fetches jobs from and reports measurement results to the Synthetic Shepherd service of the Synthetic Server. Thus, you must correctly configure the Synthetic Agent so it can connect to Synthetic Shepherd.

To connect the Synthetic Agent to on-prem Synthetic Server, follow the steps below:

Prepare for the Agent Configuration

Before you configure the connections:

- Confirm that you have installed the Synthetic Private Agent.
- Confirm that the EUM Server is running.
- Get the EUM account and the license key from the Controller Admin Console or the EUM Server properties file.

Configure the Synthetic Private Agent

1. Stop the Synthetic Private Agent if it's running by double-clicking the desktop icon Stop Agent.
2. Change to the directory `C:\appdynamics\synthetic-agent\synthetic-driver\conf`.
3. Edit the file `synthetic-driver.yml`.
4. Assign your EUM account and the license key to the properties `eumAccount` and `licenseKey,` and the URL to Synthetic Shepherd to `shepherdUrl` as shown below:

```
## Use the URL to your Synthetic Server and the port to the Synthetic Shepherd (10101)
shepherdUrl: http://<on-prem-synthetic-server-host>:10101
## You can get the values for this from the Controller Admin Console > Controller Settings
## or the properties 'property_eum-account-name' and 'property_eum-license-key' from your license file.
privateClient:
    eumAccount: "<eum_account>"
    licenseKey: "<license_key>"
```

5. Save the file.
6. Restart the Synthetic Private Agent by double-clicking the desktop icon Start Agent.

Verify the Private Location in the Controller

Follow the instructions given in Create a Job and Choose Locations to create a synthetic job using the private location where your Synthetic Private Agent is running.

Connect to the SaaS Environment for the Synthetic Hosted Agent

Related pages:

- Synthetic Agent Locations

For your on-premises Synthetic Server to use Synthetic Hosted Agents, you need to configure the on-premises Synthetic Server to communicate with the SaaS EUM Server and SaaS Synthetic Server.

Follow these steps to use Synthetic Hosted Agents:

For a complete list of Synthetic Hosted Agent browser locations, containers, and providers, go to Synthetic Agent Locations.

Verify the License Has an HMAC Key

After you have obtained a license to use the Synthetic Hosted Agent, be sure to check that the `license.lic` file has the `property_eum-hmac-key` field that is assigned a keyed-hash message authentication code (HMAC) similar to the following:

```
property_eum-hmac-key=1a88392cb0004b45b555a854b80f23f5
```

The HMAC key is used to authenticate your on-prem deployment to the SaaS EUM Server and the SaaS Synthetic Server. Without the HMAC key, your on-premises Synthetic Server will not be able to use Synthetic Hosted Agents.

Apply the License

To apply the license:

1. Copy the license file to the Controller home directory. After moving the license file, allow up to 5 minutes for the license change to take effect.
2. Follow the instructions given in Provision EUM Licenses based on your deployment.

Configure the Connection to the SaaS EUM and Synthetic Servers

Configure the connections from the on-prem Synthetic Server to the SaaS EUM and Synthetic Servers based on the region where you have an EUM account.

SaaS EUM Account in the Americas

If your SaaS EUM account is in the Americas, do not change the following default settings given in `inputs.groovy`:

```
feeder_server_url = "wss://synthetic-feeder.api.appdynamics.com"
saas_cloud_api_url = "https://api.eum-appdynamics.com"
```

SaaS EUM Account in Other Regions

If your SaaS EUM account is not in the Americas, follow these instructions to configure your on-premises Synthetic Server to use the SaaS EUM Server and SaaS Synthetic Server in your region.

1. From the on-premises Synthetic Server, edit the `inputs.groovy` file.
2. For the `feeder_server_url` and `saas_cloud_api_url` fields, enter the SaaS URLs for your region.

Administer the Synthetic Server

You can use the command line to perform platform administration tasks with the Synthetic Server, such as starting and stopping the services. This page describes the available commands, the log files, and the endpoints for testing the reachability and health of the Synthetic Server. Run the commands from the root directory of the Synthetic Server home.

Start and Stop the Synthetic Server

Start the Synthetic Server from the root directory of the Synthetic Server home as follows:

```
unix/deploy.sh start
```

To check if the Synthetic Server services are running and accessible, run the following command and confirm that there is output:

```
netstat -lan | grep "1[0,2,6]10[1,2]"
```

To stop the Synthetic Server:

```
unix/deploy.sh stop
```

Increase the Synthetic Agent Support

By default, the machine hosting the Synthetic Server should be able to support 100 concurrent Synthetic Agents. To support more than 100 concurrent Synthetic Agents, modify the throttle configuring for the Synthetic Shepherd and Synthetic Scheduler. The default maximum number of requests per second is 60. By increasing the maximum number of requests per second, the Synthetic Server can support more Synthetic Agents.

To increase the maximum number of requests per second that the Synthetic Server can receive:

1. Log on to the machine hosting the Synthetic Server.
2. Change to the root directory of the Synthetic Server home.
3. Edit the file `synthetic-processor/conf/synthetic-shepherd.yml` and increase the value for the property `maxRequestsPerSecondOverall`. In this example configuration, the value is increased to 80.

   ```
   throttleConfiguration:
       maxRequestsPerSecondOverall: 80
   ```

4. Edit the file `synthetic-processor/conf/synthetic-scheduler.yml` and increase the value for the property `maxRequestsPerSecondOverall`. Again, in this example configuration, the value is increased to 80.

   ```
   throttleConfiguration:
       maxRequestsPerSecondOverall: 80
   ```

5. Restart the Synthetic Server:

   ```
   unix/deploy.sh stop
   unix/deploy.sh start
   ```

6. You can verify that the settings have been updated by checking the logs for the Synthetic Server you changed:

   ```
   cat logs/scheduler/synthetic-scheduler.log | grep -oP -- "maxRequestsPerSecondOverall=\d+"
   cat logs/shepherd/synthetic-shepherd.log | grep -oP -- "maxRequestsPerSecondOverall=\d+"
   ```

7. You should also check the health of the Synthetic Server.

Create Preset Health Rules and Dashboards

Once you have monitored the Synthetic Server, run the following command to create the preset health rules and dashboards:

```
unix/post_deploy.sh
```

See Create Preset Dashboards and Health Rules to learn more about the preset health rules and dashboards.

Upgrade the Synthetic Server

You can update the Synthetic Server and the database schema without uninstalling and reinstalling the Synthetic Server using the `update` command. See Upgrade the Synthetic Server for instructions.

Check the Health of the Synthetic Server

To check if the Synthetic Server is running, you can run the following. You should receive the response `pong`.

```
curl <on-prem-synthetic_server_url>:10102/ping
curl <on-prem-synthetic_server_url>:12102/ping
curl <on-prem-synthetic_server_url>:16102/ping
```

To check the health of the Synthetic Server:

```
curl <on-prem-synthetic_server_url>:10102/healthcheck?pretty=true
curl <on-prem-synthetic_server_url>:12102/healthcheck?pretty=true
curl <on-prem-synthetic_server_url>:16102/healthcheck?pretty=true
```

If the Synthetic Server is healthy, the response should be similar to the following:

curl <on-prem-synthetic_server_url>:10102/healthcheck?pretty=true

```
{
  "authentication" : {
    "healthy" : true
  },
  "deadlocks" : {
    "healthy" : true
  },
  "httpClient" : {
    "healthy" : true
  },
  "quartzScheduler" : {
    "healthy" : true
  }
```

curl <on-prem-synthetic_server_url>:16102/healthcheck?pretty=true

```
{
  "deadlocks" : {
    "healthy" : true
  }
```

curl <on-prem-synthetic_server_url>:12102/healthcheck?pretty=true

```
{
  "authentication" : {
    "healthy" : true
  },
  "cluster" : {
    "healthy" : true
  },
  "deadlocks" : {
    "healthy" : true
  },
  "linter" : {
    "healthy" : true
  },
  "quartzSynthBackgroundScheduler" : {
    "healthy" : true
  },
  "quartzSynthJobScheduler" : {
    "healthy" : true
  }
```

Log Files for the Synthetic Server

The Synthetic Server creates the following error log files for each service:

- `<installDir>/logs/synthetic-scheduler.err`
- `<installDir>/logs/synthetic-shepherd.err`
- `<installDir>/logs/synthetic-feeder-client.err`

For general (non-error) log files, see the following directories:

- `<installDir>/logs/scheduler`
- `<installDir>/logs/shepherd`
- `<installDir>/logs/feeder-client`

The naming convention for the general log files is `<log>-YYYY-MM-DD.log`. You will need to set up a policy to archive or delete the general log files to prevent running out of disk space.

Secure the Synthetic Server

You are recommended to configure the Synthetic Server to use SSL to secure network connections. This page describes how to create a custom keystore and then configure the Synthetic Server to use it to implement SSL

Set Up a Custom Keystore for the Synthetic Server

The following sections describe and show an example of how to create a custom RSA security certificate, generate a new JKS keystore, and sign the certificate.

Install Prerequisite Libraries

Make sure the following libraries are installed on the Synthetic Server:

- `keytool`
- `openssl`

Create a Certificate and Keystore

Use the `keytool` command to create a keystore that uses RSA encryption then generate a certificate signing request (CSR).

The following steps show you an example of how to do both.

1. Log in to the Synthetic Server machine.
2. From a command-line shell, navigate to the root directory of the Synthetic Server:

   ```
   cd <synthetic_server_root>
   ```

3. Create a new keystore with a new unique key pair that uses RSA encryption:

   ```
   <path_to_jre>/jre/bin/keytool -genkey -keyalg RSA -validity <validity_in_days> -alias 'synthetic-server'
   -keystore ./mycustom.keystore
   ```

   This creates a new public-private key pair with an alias of "synthetic-server". You can use any value you like for the alias. The "first and last name" required during the installation process becomes the common name (CN) of the certificate. Use the name of the server.
4. Configure the keystore by entering the information requested at the command prompt.
5. Specify a password for the key store. You need to configure this password in the Synthetic Server configuration file later.
6. Generate a certificate signing request (CSR):

   ```
   <path_to_jre>/jre/bin/keytool -certreq -keystore ./mycustom.keystore -file /tmp/synthetic-server.csr -
   alias 'synthetic-server'
   ```

   This generates a certificate signing request based on the contents of the alias; in the example, it is "synthetic-server".

Sign and Install the Signed Certificate

Once you have a CSR, you request a Certificate Authority to sign it and then install the signed certificate.

The following steps are a continuation of the process from Create a Certificate and Keystore:

1. Send the output file from the last step (`/tmp/synthetic-server.csr` in this example) to a Certificate Authority for signing.
2. Install the certificate for the Certificate Authority used to sign the `.csr` file:

```
<path_to_jre>/jre/bin/keytool -import -trustcacerts -alias myorg-rootca -keystore ./mycustom.keystore -
file /path/to/<CA-root-cert>
```

This command imports your CA's root certificate into the keystore and stores it in an alias called "myorg-rootca".

3. Install the signed server certificate as follows:

   ```
   <path_to_jre>/jre/bin/keytool -import -keystore ./mycustom.keystore -file /path/to/<signed-cert>  -alias
   'synthetic-server'
   ```

   This command imports your signed certificate over the top of the self-signed certificate in the existing alias; in the example, it is "synthetic-server".
4. Import the root certificate to the other platform components connecting to the Synthetic Server through HTTPS:

   ```
   keytool -import -trustcacerts -alias <alias_name> -file mycert.cer -keystore <complete_path_to_cacerts.
   jks>
   ```

Configure the Synthetic Server to Use the Keystore

Follow the steps below to configure the Synthetic Server to use the signed certificate and its password.

1. Edit the Synthetic Scheduler configuration file at `<installation directory>/conf/synthetic-scheduler.yml` and add the `applicationConnectors` object shown below under `server`:

   ```
   server:
       ...
       applicationConnectors:
           - type: https
             port: <port>
             keyStorePath: <path to JKS files>
             keyStorePassword: <jks file password>
             validateCerts: false
   ```

   If you don't already have a signed certificate, see Create and Sign an RSA Security Certificate.
2. Edit the Synthetic Shepherd configuration file at `<installation directory>/conf/synthetic-shepherd.yml` and add the `applicationConnectors` object shown below under `server`:

   ```
   server:
       ...
       applicationConnectors:
           - type: https
             port: <port>
             keyStorePath: <path to jks file>
             keyStorePassword: <jks file password>
             validateCerts: false
   ```

3. Restart the Synthetic Server.
4. Verify the connection to the HTTPS port.

Configure a Proxy for the Synthetic Server

Related pages:

- Use a Reverse Proxy

This page describes how to configure your on-prem Synthetic Server to use a proxy server to communicate with the SaaS EUM Server and SaaS Synthetic Server. You can set up a proxy to add a security layer for your on-prem Synthetic Server.

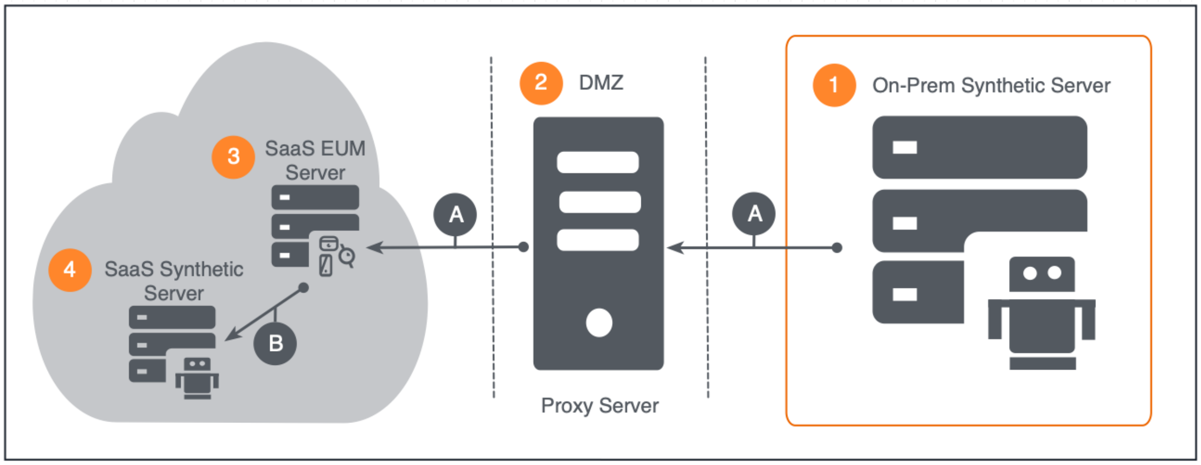
The configuration consists of the following steps:

Synthetic Server Proxy Architecture

You can also, optionally, set up a proxy to forward traffic from the SaaS Synthetic Server Feeder to the on-prem Synthetic Server Client Feeder, but your proxy server must support WebSockets.

Proxy for Downstream Traffic (Recommended)

The proxy often resides in the DMZ for the network and presents a virtual IP address to forward requests from the on-prem Synthetic Server to SaaS Synthetic Server through the SaaS EUM Server as shown below.

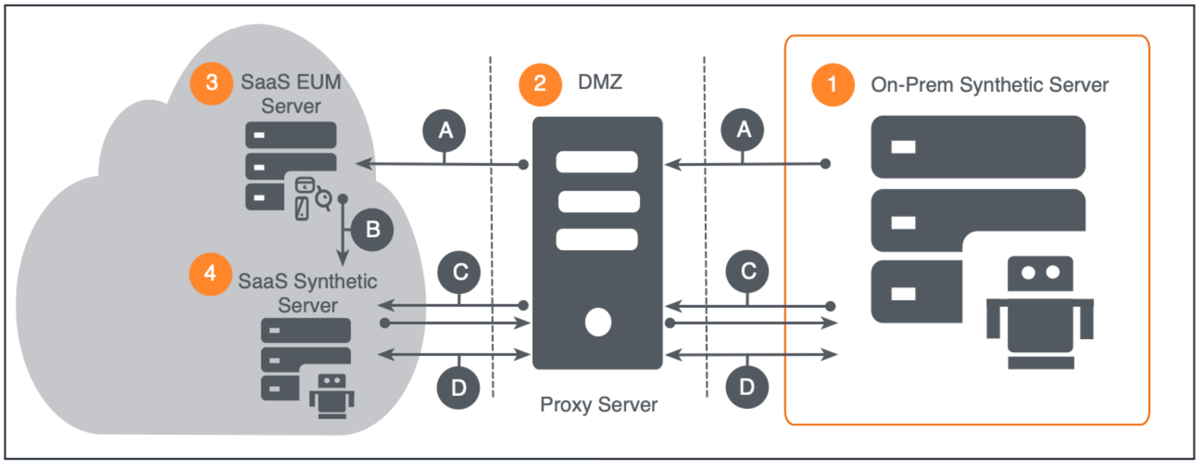


Connection	Source	Destination	Protocol	Default Port(s)
A	The 1 on-prem Synthetic Scheduler sends the job requests to the 2 proxy server, which then forwards the requests to the 3 SaaS EUM Server	3 SaaS EUM Server	HTTP(S)	7001/7002
B	The 3 SaaS EUM Server forwards the job requests to the 4 SaaS Synthetic Server.	4 SaaS Synthetic Server	HTTP(S)	N/A

The on-prem Synthetic Server Client Feeder communicates with the SaaS Synthetic Server Feeder through a bi-directional WebSocket connection that is not made through the proxy server.

Proxy for Downstream and Upstream (Optional)

When setting a proxy for downstream and upstream traffic, the proxy must support WebSocket. The on-prem Synthetic Server Client Feeder initializes the WebSocket connection with the SaaS Synthetic Server Feeder through the proxy server. Once the WebSocket connection is established, the on-prem Synthetic Server Client Feeder and the SaaS Synthetic Server Feeder can use the persistent connect to make bi-directional requests.



Connection	Source	Destination	Protocol	Default Port(s)
A	The 1 on-prem Synthetic Server (Scheduler) sends the job requests to the 2 proxy server, when then forwards the requests to the 3 SaaS EUM Server.	3 SaaS EUM Server	HTTP(S)	7001/7002
B	The 3 SaaS EUM Server forwards the requests to the 4 SaaS Synthetic Server.	4 SaaS Synthetic Server	HTTP(S)	N/A
C	The 1 on-prem Synthetic Server (Client Feeder) establishes a WebSocket connection with the 4 SaaS Synthetic Server (Synthetic Server Feeder) through the 2 proxy server.	4 SaaS Synthetic Server	HTTP(S)	80/443
		1 on-prem Synthetic Server	HTTP(S)	80/443
D	The 4 SaaS Synthetic Server (Synthetic Server Feeder) and 1 on-prem Synthetic Server (Client Feeder) communicate bi-directionally through the 2 proxy server. Most of the traffic is from the SaaS Synthetic Server Feeder to the on-prem Synthetic Client Feeder.	1 on-prem Synthetic Server	WebSocket	16101
		4 SaaS Synthetic Server	WebSocket	16001

Configure the Synthetic Server to Use Proxy

You need to configure the Synthetic Server to use the proxy to forward traffic to the SaaS EUM Server. In the examples configurations below, the proxy URL is `127.0.0.1` and the proxy port is `3128`. You will need to replace those values with the URL and port of your proxy server.

Synthetic Shepherd

From the host machine of the Synthetic Server, set the `proxyUrl` and `proxyPort` properties in the `<synthetic-server_installation_dir>/synthetic-processor/conf/synthetic-shepherd.yml` file to point to the URL and port of the proxy server, which the Synthetic Shepherd will send requests.

```
saasLink:
  proxyUrl: "127.0.0.1"
  proxyPort: 3128
```

Synthetic Scheduler

Set the `proxyUrl` and `proxyPort` properties in the `<synthetic-server_installation_dir>/synthetic-processor/conf/synthetic-scheduler.yml` file to point to the URL and port of the proxy server so that Synthetic Shepherd also sends requests to the proxy.

```
saasLink:
  proxyUrl: "127.0.0.1"
  proxyPort: 3128
```

Synthetic Feeder Client

Finally, set the `proxyUrl` and `proxyPort` properties in the `<synthetic-server_installation_dir>/synthetic-processor/conf/synthetic-feeder.yml` file to point to the URL and port of the proxy server so that Synthetic Server Client Feeder also sends requests to the proxy.

```
websocketConfiguration:
    proxyUrl: "127.0.0.1"
    proxyPort: 3128
```

Note

Feeder Client for Synthetic services does not support Proxy authentication.

Configure Your Proxy Server

You can use Squid, Apache, Nginx, or another proxy server, but these instructions only cover Squid. If you are using a proxy to forward traffic between the on-prem Synthetic Server Client Feeder and the Synthetic Server Feeder, your proxy will need to support the WebSocket protocol.

Squid

1. Add the following configurations to the Squid configuration file at `/etc/squid/squid.conf`.

   ```
   http_access allow localhost
   http_access allow all
   http_port 3128
   ```

2. Restart `squid`.

Monitor the Synthetic Server

AppDynamics recommends that you monitor the performance of the Synthetic Server with the Java Agent. Once you have instrumented the Synthetic Server, you can use preset capacity monitoring dashboards and health rules or create custom dashboards and health rules based on JMX and the Synthetic Server metrics.

Follow the steps below to monitor the Synthetic Server:

Install the Java Agent

You should install the Java Agent in the same directory as the Synthetic Server installation directory. See Install the Java Agent for instructions.

Configure the Java Agent

Configure the Java Agent to report metrics to a specific Controller:

1. Change to `<agent_home>/conf/`.
2. Edit the `controller-info.xml` file so that the values for the following elements match your Controller information, application name, tier name, and node name:
   - `<controller-host>`
   - `<controller-port>`
   - `<application-name>`
   - `<tier-name>`
   - `<node-name>`
3. For example, your `controller-info.xml` file might look similar to the following:

```
<controller-info>
    <controller-host>192.168.1.20</controller-host>
    <controller-port>8090</controller-port>
    <application-name>SyntheticServer</application-name>
    <tier-name>SchedulerTier</tier-name>
    <node-name>SchedulerNode</node-name>
</controller-info>
```

ⓘ You must ensure that the application name, tier name, and node name are the same as the `javaagent.jar` parameters when you attach the Java Agent to the Synthetic Server.

Connect the Synthetic Server with the Controller

Before you installed the Synthetic Server, you needed to configure the Synthetic Server to connect to the EUM Server's MySQL database and the EUM Collector. In this section, you will set configurations in `inputs.groovy` to connect the Synthetic Server to the Controller, so that the predefined health rules and dashboards can be created.

In the `inputs.groovy` file, make sure you have set the following properties. Replace placeholders in brackets with information about your Controller as well as the application and tier that are being monitored.

```
controller_host = "http(s)://<url_to_machine_running_controller>"     // The URL to your on-prem Controller
controller_port = "<port_number>"                                     // The default is 8090.
controller_account = "<controller_account>"                           // Account used for running post-deploy
tasks
controller_username = "<controller_username>"                         // Username for making API calls to
controller
prompt_for_password = "false"                                         // When false, the password below will be
used without prompting.
controller_password = "<controller_password>"                         // Password used for username. It is not
stored in any config files.
controller_synth_app = "<app_name_set_in_controller-info.xml>"        // This is the application shown in the
Controller and is based on the value given in the <application-name> element in controller-info.xml.
controller_shepherd_entity = "<tier_name_set_in_controller-info.xml>" // This is the tier shown in the
Controller and is based on the value given in the <tier-name> element in controller-info.xml
```

Attach the Java Agent to the Synthetic Server

To attach the Java Agent to the Synthetic Server, set Java options through the variables `SCHEDULER_OPTS` and `SYNTHETIC_SHEPHERD_OPTS`. The node names and tier names given in the examples below can be modified for your use case. One JVM process requires one Java Agent to be attached.

1. Set the options for the Synthetic Scheduler so that the Java Agent is attached to the JVM process:

   ```
   SCHEDULER_OPTS="-javaagent:./java_agent/javaagent.jar -Dappdynamics.agent.applicationName=synthonprem -
   Dappdynamics.agent.nodeName=synthetic-scheduler -Dappdynamics.agent.tierName=scheduler-tier"
   ```

2. Set the options for the Synthetic Shepherd so that the Java Agent is attached to the JVM process:

   ```
   SYNTHETIC_SHEPHERD_OPTS="-javaagent:./java_agent/javaagent.jar -Dappdynamics.agent.
   applicationName=synthonprem -Dappdynamics.agent.nodeName=synthetic-shepherd -Dappdynamics.agent.
   tierName=shepherd-tier"
   ```

3. Set the options for the Synthetic Feeder Client so that the Java Agent is attached to the JVM process:

   ```
   FEEDER_CLIENT_OPTS="-javaagent:./java_agent/javaagent.jar -Dappdynamics.agent.
   applicationName=synthonprem -Dappdynamics.agent.nodeName=synthetic-feeder-client -Dappdynamics.agent.
   tierName=feeder-client-tier"
   ```

4. Export the variables for the Synthetic Server options:

   ```
   export SYNTHETIC_SHEPHERD_OPTS
   export SCHEDULER_OPTS
   export FEEDER_CLIENT_OPTS
   ```

5. From the Synthetic Server installer directory, run the following to stop and start the Synthetic Server:

   ```
   unix/deploy.sh stop
   unix/deploy.sh start
   ```

Verify the Instrumentation of the Synthetic Server

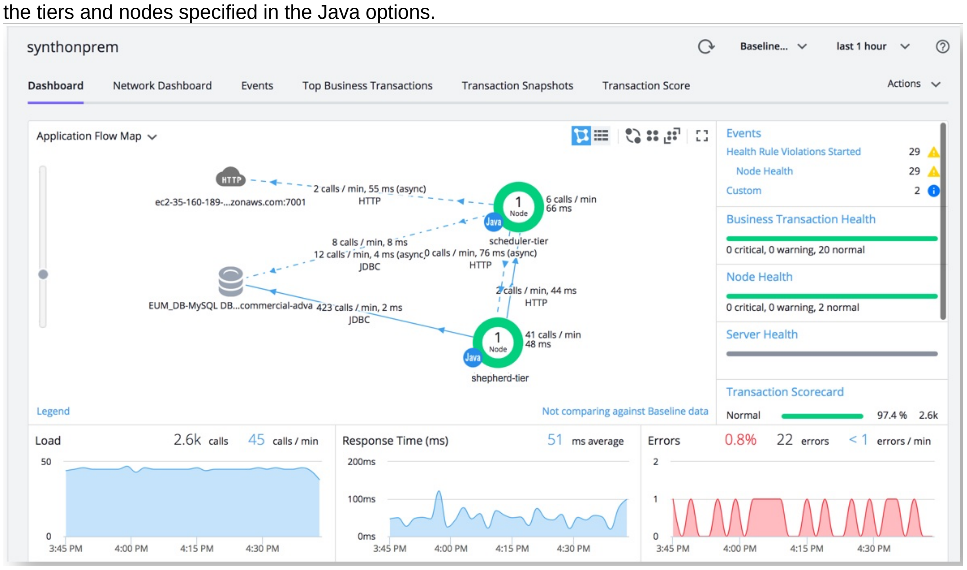
1. Confirm that the Java Agent is running:

   ```
   ps -ef | grep javaagent
   ```

2. Check the Java Agent logs:

```
tail <java-agent>/ver<agent-version>/logs/<node-name>/agent-XXXX.log
```

3. Navigate to your Controller.
4. You should see a business application with the name assigned in the `controller-info.xml` file similar to the following. You should also see the tiers and nodes specified in the Java options.


synthonprem
Baseline...
last 1 hour
Dashboard
Network Dashboard
Events
Top Business Transactions
Transaction Snapshots
Transaction Score
Actions
Application Flow Map
HTTP
ec2-35-160-189-...zonaws.com:7001
2 calls / min, 55 ms (async)
HTTP
6 calls / min
66 ms
1
Node
Java
scheduler-tier
8 calls / min, 8 ms
12 calls / min, 4 ms (async)
JDBC
0 calls / min, 76 ms (async)
HTTP
EUM_DB-MySQL DB...commercial-adva
423 calls / min, 2 ms
JDBC
2 calls / min, 44 ms
HTTP
41 calls / min
48 ms
shepherd-tier
Legend
Not comparing against Baseline data
Events
Health Rule Violations Started 29
Node Health 29
Custom 2
Business Transaction Health
0 critical, 0 warning, 20 normal
Node Health
0 critical, 0 warning, 2 normal
Server Health
Transaction Scorecard
Normal 97.4 % 2.6k
Load 2.6k calls 45 calls / min
50
0
3:45 PM 4:00 PM 4:15 PM 4:30 PM
Response Time (ms) 51 ms average
200ms
100ms
0ms
3:45 PM 4:00 PM 4:15 PM 4:30 PM
Errors 0.8% 22 errors < 1 errors / min
2
1
0
3:45 PM 4:00 PM 4:15 PM 4:30 PM

Create Preset Dashboards and Health Rules

The Synthetic Server comes with the `post_deploy.sh` command to help create preset dashboards and health rules to monitor the Synthetic Server. The health rules are based on metrics generated by Synthetic Shepherd and not the host machine of the Synthetic Agent or the Java Agent, so they serve to complement the JMX health rules such as CPU usage, memory consumption, etc.

About the Preset Dashboards and Health Rules

The preset health rules issue warnings for when the Synthetic Server has a *busy* percentage of 80% and issue critical alerts when that busy percentage reaches 90%. The busy percentage is evaluated over the last 30 minutes.

Understanding the Busy Percentage

Each Synthetic Agent can only run one job at a time. When running a job, the Synthetic Agent is busy, and when it's not running, the Synthetic Agent is *not* busy. The busy percentage indicates the percentage of time that a Synthetic Agent is running jobs. For example, if a Synthetic Agent ran a job that took five minutes and no other jobs, then over the last 10 minutes, it was 50% busy. The busy percentage is based on a metric reported every minute and calculated using pages per minute (PPM). The busy percentage can be used to monitor one Synthetic Agent, a group of Synthetic Agents, or a location.

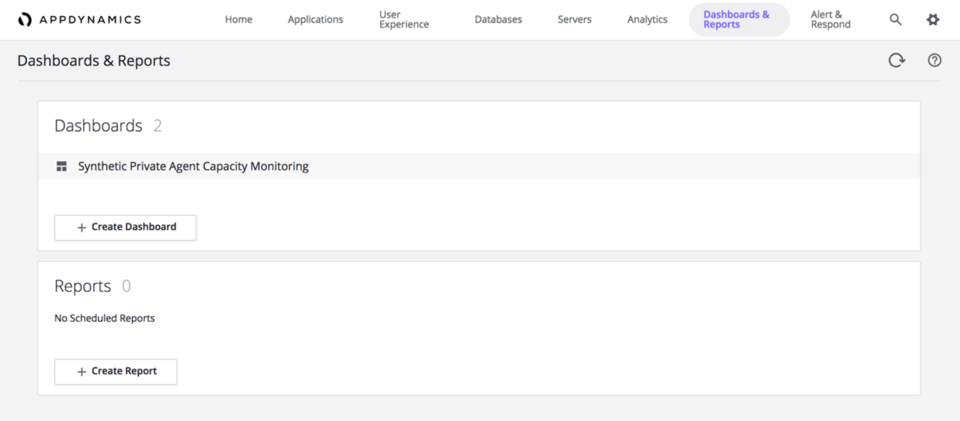
Create the Dashboards and Health Rules

Once you have verified that the instrumentation was successful and that business applications were created:

1. Log in to the machine hosting your Synthetic Server.
2. Change to the Synthetic Server home.
3. Run the following command to create the preset dashboards and health rules:

```
unix/post_deploy.sh
```

4. From the Controller UI, navigate to the **Dashboards & Reports** page.
5. You should see the dashboard **Synthetic Private Agent Capacity Monitoring** as shown here:



Dashboards & Reports

Dashboards 2

Synthetic Private Agent Capacity Monitoring

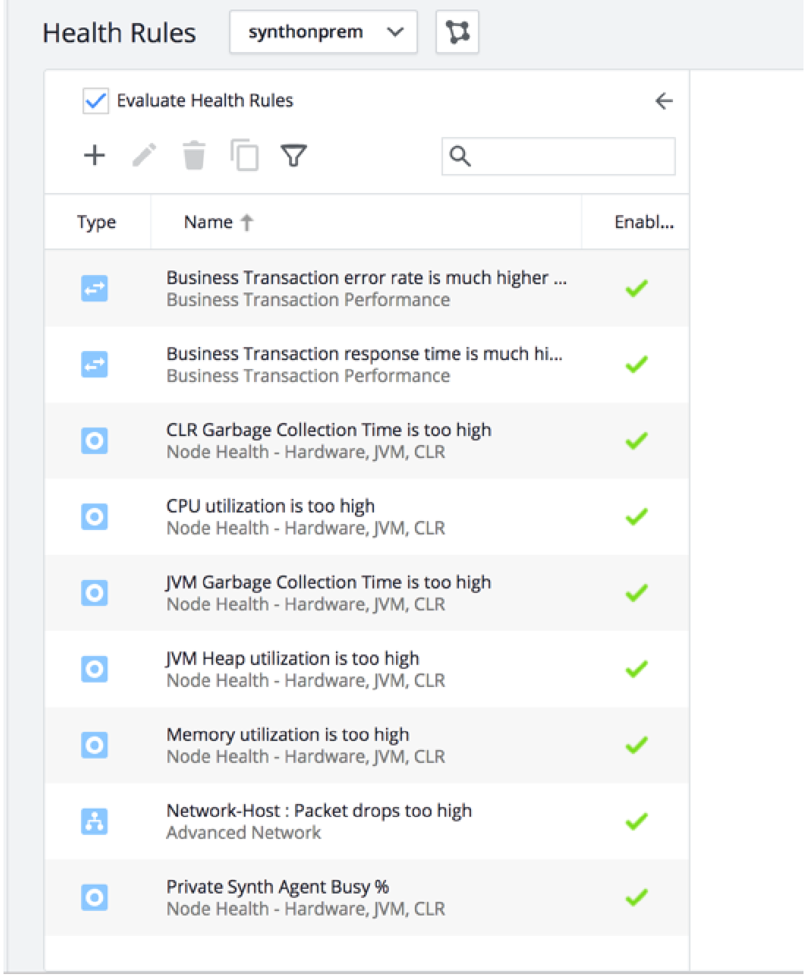
+ Create Dashboard

Reports 0

No Scheduled Reports

+ Create Report

6. Open the business application for the on-premises Synthetic Server.
7. Navigate to **Alert & Respond > Health Rules**.
8. From **Alert & Respond > Health Rules,** you will see the health rules that you created in Create Preset Dashboards and Health Rules.



Health Rules synthonprem

☑ Evaluate Health Rules

Type	Name	Enabl...
	Business Transaction error rate is much higher ... Business Transaction Performance	✓
	Business Transaction response time is much hi... Business Transaction Performance	✓
	CLR Garbage Collection Time is too high Node Health - Hardware, JVM, CLR	✓
	CPU utilization is too high Node Health - Hardware, JVM, CLR	✓
	JVM Garbage Collection Time is too high Node Health - Hardware, JVM, CLR	✓
	JVM Heap utilization is too high Node Health - Hardware, JVM, CLR	✓
	Memory utilization is too high Node Health - Hardware, JVM, CLR	✓
	Network-Host : Packet drops too high Advanced Network	✓
	Private Synth Agent Busy % Node Health - Hardware, JVM, CLR	✓

Create Custom Dashboards and Health Rules

The Java Agent will report metrics generated by Synthetic Server such as CPU usage, memory consumption, garbage collection, etc. You can use these metrics to create health rules and custom dashboards.

See the following pages for instructions:

- Custom Dashboards
- Configure Health Rules

Synthetic Server Endpoints

Related pages:

- Port Settings
- EUM Server Endpoints

The Synthetic Server has different endpoints serving distinct functions. This page provides a reference for testing the health and getting information about on-premises Synthetic Servers.

The endpoints include the following:

- **Synthetic Shepherd** - manages and dispatches jobs to the Synthetic Agents. In addition, the Synthetic Shepherd saves the measurement results to the filesystem and sends beacons containing the data to the Synthetic Scheduler.
- **Synthetic Scheduler** - handles the CRUD operations for jobs and manages the events generated for synthetic warnings and errors that occur in the measurement results.
- **Synthetic Feeder Client** - communicates with the SaaS Synthetic Feeder Server to access the Synthetic Hosted Agents.

Synthetic Server Endpoint URLs

The table below lists the endpoints, the default URL, and the supported paths.

<table>
<tr><th>Synthetic Server Endpoint</th><th>Default URL</th><th>Paths / Description</th></tr>
<tr><td>Synthetic Shepherd</td><td><code>http(s)://<domain-name>:10101</code></td><td><table><tr><td>/version</td><td>Returns the version, build, commit, and timestamp of the Synthetic Shepherd.</td></tr></table></td></tr>
<tr><td>Synthetic Scheduler</td><td><code>http(s)://<domain-name>:12101</code></td><td><table><tr><td>/version</td><td>Returns the version, build, commit, and timestamp of the Synthetic Scheduler.</td></tr></table></td></tr>
<tr><td>Synthetic Feeder Client</td><td><code>http(s)://<domain-name>:16101</code></td><td><table><tr><td>/version</td><td>Returns the version, build, commit, and timestamp of the Synthetic Feeder Client.</td></tr></table></td></tr>
</table>

Upgrade the Synthetic Server

This page describes how to upgrade the Synthetic Server to the latest version. This is often done alongside an upgrade to the other platform components, such as the EUM Server, the Controller, and the Events Service.

What the Upgrade Does

The upgrade procedure updates the schemas for the EUM Server database and the JAR files for the Synthetic Server if either is needed.

> The upgrade steps will not update 3rd-party software such as the Python library Flake8.

Who Should Use This Document

To upgrade the Synthetic Server to the latest available version, use this document.

Before You Begin

Before you upgrade the Synthetic Server:

1. Back up your current installation. There is no way to downgrade and changes are permanent.
2. Update platform components in the correct update order.

Upgrade Procedure

Follow the instructions below to upgrade your Synthetic Server.

Pre-requisite

The previous version of the Synthetic services database should be maintained.

Get the Latest Synthetic Server Installer

1. Go to the AppDynamics Downloads site and download the latest version of the Synthetic Server installer package.
2. Copy the Synthetic Server installer package (`synthetic-installer-linux-<version>.zip`) to the machine hosting your Synthetic Server.

Upgradation Steps

To upgrade Synthetic services first go to the installed Synthetic services path `cd <Synthetic_home>` and follow the steps given here:

1. Run the following command to stop Synthetic services:

   ```
   "/unix/deploy.sh stop"
   ```

2. Take a back up of `inputs.groovy` file from `<Synthetic_home>` before deleting contents of `<Synthetic_Home>`. This is done to preserve previous configuration details.
3. Delete all the files and folders under `<Synthetic_home>.`
4. Copy and unzip the new Synthetic services installer under `<Synthetic_home>.`
5. Replace `inputs.groovy.sample` file with the backed up file (mentioned in step 2).
6. Run the following command to install new services:

```
"./unix/deploy.sh install"
```

Verify the New Version Has Been Installed

1. Once the installation is complete, confirm the Synthetic Server is running by entering:

```
ps aux | grep synthetic-processor
netstat -lan | grep "1[0,2,6]10[1,2]"
```

2. To view the updated version, curl to the `version` endpoint:

```
curl <on-prem-synthetic_server>:10101/version
curl <on-prem-synthetic_server>:12101/version
curl <on-prem-synthetic_server>:16101/version
```

Secure the Platform

AppDynamics includes security features that help to ensure the safety and integrity of your deployment.

About Controller Security

The Controller is installed with an HTTPS port enabled by default. SSL secures client connections and allows clients to authenticate the Controller. The Controller UI supports HTTP Basic Authentication, along with SAML and LDAP authentication. Role-based access controls in the UI allow you to manage user privileges.

While the security features of the Controller are enabled out of the box, there are some steps you should take to ensure the security of your deployment. These steps include but are not limited to:

- The SSL port uses a self-signed certificate. If you intend to terminate SSL connection at the Controller, you should replace the default certificate with your own, CA-signed certificate. If you replace the default SSL certificate on the Controller, you will also need to establish trust for the Controller's public key on the App Agent machine.

  > As an alternative to terminating SSL at the Controller, you can put the Controller behind a reverse proxy that terminates SSL, relieving the Controller from having to process SSL.

- Along with a secure listening port, the Controller provides an unsecured, HTTP listening port as well. You should disable the port or block access to the point from any untrusted networks.
- Make sure that your App Agents connect to the Controller or to the reverse proxy if terminating SSL at a proxy, with SSL enabled.
- The Controller and underlying components, Glassfish and MySQL, include built-in user accounts. Be sure to change the passwords for the accounts regularly and in general, follow best practices for password management for the accounts. For information on changing the passwords for built-in users, see Update the Root User and Glassfish Admin Passwords.

Proxy Controller Connections

The AppDynamics Controller is often deployed to a protected network behind a proxy, which presents a virtual IP address to external connections, including to agents and browser clients. The proxy itself resides in the DMZ for the network and often terminates SSL connections from the client connections.

If clients use SSL, the reverse proxy can terminate SSL connections or maintain SSL through to the Controller. Terminating SSL at the proxy removes the processing burden from the Controller machine.

Using a secure proxy can simplify administration as a whole, by centralizing SSL key management to a single point. It allows you to use alternative PKI infrastructures, like OpenSSL.

See Use a Reverse Proxy for more information.

Additional Security Topics

- Secure the EUM Server
- Analytics and Data Security
- Agent-to-Controller Connections (for information on Securing Agent Connections)
- Roles and Permissions (for information regarding potentially sensitive user permissions)

HTTPS Support for the Enterprise Console

There is built-in HTTPS support for the Enterprise Console using a self-signed keystore file. The Enterprise Console supports either HTTP or HTTPS based on your choice during fresh installation or upgrade. You can reconfigure the Enterprise Console to revert from HTTPS back to HTTP, or vice versa, at any time.

About Enterprise Console SSL and Certificates

For the HTTPS client, the Enterprise Console packages the latest Mozilla truststore `cacerts.jks`, as it contains standard certificates. The Enterprise Console creates a `keystore.jks` file which contains a self-signed certificate. This certificate is imported into `cacerts.jks` during installation or upgrade.

For production use, AppDynamics strongly recommends that you replace the self-signed certificate with a certificate signed by a third-party Certificate Authority (CA) or your own internal CA.

This page describes how to:

- enable HTTPS for the Enterprise Console during installation or upgrade.
- update the certificate to a signed one.
- customize keystore credentials.

> Replacing the entire keystore is not recommended unless you first export the existing artifacts from the default keystore and import them into your own keystore.
>
> It is also not recommended that you create your own self-signed certificate.

The exact steps to implement security typically vary depending on the security policies for the organization. For example, if your organization already has a signed certificate to use, such as a wildcard certificate used for your organization's domain, you can import it into the keystore using the Enterprise Console's update-certificate command. Otherwise, you can obtain a new one along with a certificate signing request.

Before Starting

On Linux machines, the Enterprise Console uses curl to check the responsiveness of the application URL. Therefore, SSL needs to have the latest NSS package to work.

For example, you can update the NSS package to the latest with the following command on CentOS:

```
yum update nss
```

> Your update procedure or NSS package name may differ depending on your Linux operating system.

Enable Enterprise Console HTTPS for Fresh Installation

You can enable HTTPS during the Enterprise Console installation by selecting HTTPS as your preferred connection type:

1. Follow the steps to install the Enterprise Console:
    - For GUI Installation, click the HTTPS checkbox.

      > If the checkbox is left unselected, then the installation will default to HTTP.

      The Enterprise Console will create a self-signed certificate and use it for your HTTPS connection.
    - For Silent Installation, edit the platform admin response `varfile` with the following parameter:

```
platformAdmin.useHttps$Boolean=true
```

This enables HTTPS connection and is the same as checking **Enable Https Connection** in the wizard installation option. Setting the parameter to false uses HTTP.

2. Optional: The Enterprise Console will create a self-signed certificate and use it for HTTPS connection. You can replace it with a certificate signed by a third-party CA. See Update to a Signed Certificate.

The Enterprise Console then configures the HTTPS protocol and disables HTTP in the Dropwizard configuration file, PlatformAdminApplication.yml. See the Dropwizard Configuration Reference for more information.

Example Enterprise Console HTTPS connector configuration:

```
server:
  type: simple
  connector:
    type: encrypted-https # encrypted-https is a customized HTTPS connector type in Enterprise Console, with
keystore password encrypted.
    port: 9191 # DO NOT REMOVE alternatives are 8080
    keyStorePath: /appdynamics/platform-admin/conf/keys/keystore.jks
    keyStorePassword: s_-001-12-v/yKyIweuGQ=iLpBEDTqfP7vj++WP+MKEg==
    trustStorePath: /appdynamics/platform-admin/conf/keys/cacerts.jks
    trustStorePassword: s_-001-12-hdLwJEOZbns=kDmS/pLvq2A43iCWLJEcTg==
    certAlias: ec-server # DO NOT change cert alias name in keystore files.
    validateCerts: false
    supportedProtocols: [TLSv1.2]
    bindHost: 0.0.0.0
  applicationContextPath: /
```

ⓘ The Enterprise Console encrypts all plain text passwords in the configuration file.

Enable Enterprise Console HTTPS for Upgrades

You can enable HTTPS from HTTP during upgrades from 4.4.3 to 4.5 and post-4.5 upgrades. You can also enable HTTPS for an existing Enterprise Console instance by reinstalling the Enterprise Console application:

1. Follow the steps to upgrade the Enterprise Console.
   a. Change the platform admin response varfile with:

```
platformAdmin.useHttps$Boolean=true
```

2. Complete the upgrade.
3. Optional: A self-signed certificate is created by the Enterprise Console and SSL is configured, just like it is with a fresh installation. You can replace it with a certificate signed by a third-party CA. See Update to a Signed Certificate.

ⓘ When you upgrade your HTTPS supported Enterprise Console to future release versions, the Enterprise Console will follow the below protocols:

- **Upgrades with a self-signed certificate (the Enterprise Console installed certificate)**: The Enterprise Console will always recreate the new `keystore.jks`, with a new self-signed certificate in it, and update the `cacerts.jks` file with the new self-signed certificate under the `<EC_installationDir>/conf/keys` folder.
- **Upgrades with a signed certificate**: The Enterprise Console will not modify your signed certificate, leaving it as it was before the upgrade.
  Note: Do not change the `serverHostName` in the admin response varfile from a private IP/hostname (if you used a private IP /hostname as the `serverHostName` for a previous Enterprise Console fresh install or upgrade) to a public IP as the Enterprise Console will not support the signed certificate afterwards. This restriction only applies to upgrades with a signed certificate.
- **Upgrades with customized keystore/truststore path and passwords**: The Enterprise Console will back up the `.jks` files only if they are under the `<EC_installationDir>/conf/keys` folder. The Enterprise Console will restore your keystore/truststore paths and passwords in `PlatformAdminApplicationl.yml` before the upgrade, even if you move the `.jks` files or change the password.

⚠ Changing the .jks files location is not recommended as they will not be backed up by the Enterprise Console if they are in another location.

Configure HTTPS for Enterprise Console in SAN Deployments

To configure HTTPS for the Enterprise Console deployed for a Subject Alternative Name (SAN) on AWS, you will need to generate keys from the `san.cnf` file. The instructions below show you how to enter multiple hostnames and aliases for the Enterprise Console in the `san.cnf` file and then generate the keys with it.

1. Create your `san.cnf` file for the SAN. In the following example `san.cnf` file, multiple domain names and aliases are defined in `[ alt_names ]`.

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt             = no
[ req_distinguished_name ]
countryName           = IN
stateOrProvinceName   = Karnataka
localityName          = Bangalore
organizationName      = Appdynamics
commonName            = ECserver
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1   = ECserver.com
DNS.2   = ECserver.secondary.com
DNS.3   = ECserver.alias1.com
DNS.4   = ECserver.alias2.com
IP.1        = 10.10.10.10
IP.2        = 10.10.10.9
```

2. `Using the san.cnf file, generate the private key and CSR with the following openssl command:`

```
openssl req -new -newkey rsa:2048 -nodes -out sslcert.csr -keyout private.key -config san.cnf
```

3. `Check the CSR to confirm the SANs are correct:`

```
openssl req -noout -text -in sslcert.csr | grep DNS
openssl req -noout -text -in sslcert.csr | grep IP
```

4. `Sign the CSR by a certified authority (CA).`
5. `Update the certificate for the Enterprise Console:`

```
./platform-admin.sh update-certificate --private-key <privateKeyfile> --ssl-cert <sslCertFile> --ssl-
chain <sslChainfile1> <sslChainfile2> <...>
```

Customize Keystore Credentials

You can customize keystore credentials. The Enterprise Console preserves your customized keystore/truststore passwords.

> ⚠ To keep your customized files backed up, place them under the `<EC_installationDir>/conf/keys` directory. Placing your files anywhere besides `<EC_installationDir>/conf/keys` is not recommended because they may not be backed up.

If you change the keystore content for the Enterprise Console, you must re-run the `change-keystore-password` command and re-encrypt. Then, you need to restart the Enterprise Console. See Controller Secure Credential Store for more information.

Get Encrypted Password

You can update the password for the `.jks` files. If you do, you must also update the password in the `PlatformAdminApplication.yml` file.

> ⚠ Do not change the `supportedProtocols` in the `PlatformAdminApplication.yml` file.

To get an encrypted password:

> ⓘ - Make a note of the generated password to use in Step 4.
>
> - For the `keystore.jks` file to work for the Enterprise Console, the `storepass` and `keypass` must be the same.

1. Use the Enterprise Console CLI to encrypt the new password:

   ```
   ./platform-admin.sh encrypt -t '<plain_text_password>'
   ```

2. Change the storepass in keystore.jks by using the `<plain_text_password>` from Step 1:

   ```
   keytool -storepasswd -keystore keystore.jks
   ```

3. Change the `ec-server keypass` in `keystore.jks` by using the `<plain_text_password>` from Step 1:

   ```
   keytool -keypasswd -alias ec-server -keystore keystore.jks
   ```

4. Use the encrypted password to update the Enterprise Console Dropwizard confirmation `yml` file (`PlatformAdminApplication.yml`) for the key "keyStorePassword".
5. Restart the Enterprise Console.

Update to a Signed Certificate

You can update the built-in self-signed certificate created by the Enterprise Console with a CA signed certificate from an eligible CA authority.

> ⚠ If you want to reuse the existing public key from the Java keystore to generate a CSR request, you must import the signed certificates manually. See step 6 to 9 in Create a Certificate and Generate a CSR for more information.

Most Linux distributions include OpenSSL. If you are using Windows or your Linux distribution does not include OpenSSL, you may find more information on the OpenSSL website.

1. Obtain a signed certificate:
   a. Create a csr request.

      ```
      //Some CAs will create everything for you, including the private key. You may use the following
      keytool command to create a csr request from existing keystore.jks.
      keytool -certreq -alias ec-server -keystore keystore.jks -file AppDynamics.csr
      ```

      or

      ```
      //You can also use the following openssl command to create your own private key and csr request.
      openssl req -new -newkey rsa:2048 -nodes -out <name of csr request file>.csr -keyout <name of
      private key>.key -subj "/C=<custom>/ST=<custom>/L=<custom>/O=<custom>/OU=<custom>/CN=<hostname>"
      ```

   b. Submit the certificate signing request file generated by the command (AppDynamics.csr in our example command) to your Certificate Authority of choice. When it's ready, the CA will return the signed certificate and any root and intermediate certificates required for the trust chain. The response from the Certificate Authority should include any special instructions for importing the certificate, if needed. If the CA supplies the certificate in text format, just copy and paste the text into a text file.
2. Run the Enterprise Console update certificate CLI command:

```
./platform-admin.sh update-certificate --private-key <privateKeyfile> --ssl-cert <sslCertFile> --ssl-chain <sslChainfile1> <sslChainfile2> ...
```

> ⚠ Refer to the following help points when running this command:
>
> - The `privateKeyfile`, `sslCertFile`, and `ssl-chain` files do not have any file format restrictions. Any file format, such as `.pkey` and `.txt`, should work, as long as it is readable.
> - The `privateKeyfile` file content must follow the PKCS8 format.
> - `sslCertFile` is your SSL certificate file.
> - `ssl-chain` files are additional certificates, such as intermediate certificates. These are optional, and you may provide as many of them as you would like.

This command updates the certificate in the keystore and truststore in the configuration yml file.

3. Restart the Enterprise Console for the new SSL configurations to take effect.

Verify the Use of SSL

You can test that your HTTPS support works by logging into the Enterprise Console GUI.

To make sure the configuration works, use a browser to connect to the Enterprise Console over the default secure port, port 9191:

```
https://<hostname>:9191
```

Specify the hostname you used when you installed the Enterprise Console. The default port is 9191. This port needs to be exposed from your firewall rules so you can access the port from any place. See Port Settings for more information.

Make sure the Enterprise Console entry page loads in the browser correctly.

> ⚠ Depending on your browser, you may have to perform additional steps to verify your connection. For instance, for self-signed certificates on Chrome, you have to click to proceed. On Firefox, you have to create a security exception to proceed.

You can also verify that your configuration works by running commands on the Enterprise Console CLI.

Expired Certificate

In case of an expired certificate, the Enterprise Console CLI will still continue to work, but the CLI will also print out a warning, notifying you that the certificate has expired.

Upgrades to the Enterprise Console remain unaffected by expired certificates; when you try to upgrade the Enterprise Console without knowing that the certificate has expired, the upgrade should still succeed.

You can update the Enterprise Console self-signed certificate by reinstalling the application. For your own signed certificate, you can obtain a new one from CA and run the CLI command from Update to a Signed Certificate.

Revert the Enterprise Console to HTTP

You can fall back to HTTP from HTTPS support by reinstalling the Enterprise Console application.

1. Follow the steps to upgrade the Enterprise Console.
   a. Change the platform admin response varfile with:

   ```
   platformAdmin.useHttps$Boolean=false
   ```

2. Complete the upgrade.

The Enterprise Console backs up any self-signed or signed certificate that is under `<EC_installationDir>/conf/keys`. However, if you manually move the `keystore.jks` and `truststore.jks` to your own location, then you will need to back up your customized certificates and SSL related files on your own before the upgrade.

Troubleshooting Common SSL Related Issues

This section covers a few of the most common Enterprise Console SSL related issues. It may help to enable `-Djavax.net.debug=ssl` in the platform-admin.sh execute CLI command section before troubleshooting.

Enterprise Console CLI Issue

If the installation or upgrade succeeds but the Enterprise Console CLI does not work, the CLI will remind you to check if the serverHostName you entered in the installer varfile is valid. The error will state that the `Hostname %s not verified.`

Certificates Without an Extension Field Issue

Certificates without an extension field, especially free signed certificates, will not work. They will lead to a KeyUsage error. Only signed certificates from an eligible CA authority should be used.

Enterprise Console Restart Issue

NSS package needs to be updated to the latest version on Linux machines. The Enterprise Console may not be able to restart if the NSS package is not updated to the latest version on Linux machines. See Before Starting for more information.

Controller SSL and Certificates

The Controller comes with a preconfigured HTTPS port (port 8181 by default) that is secured by a self-signed certificate. This page describes how to replace the default certificate with your own custom certificate.

About Controller SSL and Certificates

For production use, AppDynamics strongly recommends that you replace the self-signed certificate with a certificate signed by a third-party Certificate Authority (CA) or your own internal CA. If you are deploying .NET Agents, you must replace the self-signed certificate with one signed by a CA, since the .NET agents do not work with self-signed certificates.

Controller SSL Certificates

You can manage your Controller SSL certificate on the Enterprise Console UI under Configurations. The Appserver Configurations and Reports Service Configurations pages both contain sections that display the SSL certificate information and provide an Edit Certificate option.

Controller Keystore and Artifacts

This page describes how to replace the existing key in the default keystore. Replacing the entire keystore is not recommended unless you first export the existing artifacts from the default keystore and import them into your own keystore.

The default Controller keystore includes the following artifacts:

- **glassfish-instance**: A self-signed private key provided the Glassfish application server.
- **s1as**: A self-signed private key provided with the Glassfish application server used by the Controller for secure communication on port 8181.
- **reporting-instance**: A private key used by the reporting service, the service that enables scheduled reports.

Update Keystore Passwords

You can modify the password for the `keystore.jks` and `cacert.jks` files that are used to generate the keystore artifacts. The password for both files must be the same.

You cannot modify, however, the password for the `reporting-service.pfx` file that is generated by the keystore artifact `reporting-instance` and used by the Reporting Service.

How to View the Keystore

You can view the contents of the keystore yourself using the keytool utility. To do so, from the <controller_home>/jre/bin directory, run the following command. Enter the default keystore password changeit when prompted.

```
keytool -list -v -keystore /home/ec2-user/appDplatform/product/controller/appserver/glassfish/domains/domain1
/config/keystore.jks
```

The exact steps to implement security typically vary depending on the security policies for the organization. For example, if your organization already has a certificate to use, such as a wildcard certificate used for your organization's domain, you can import the existing certificate into the Controller keystore. Otherwise, you'll need to generate a new one along with a certificate signing request. The following sections take you through these scenarios.

Before Starting

The following instructions describe how to configure SSL using the Java `keytool` utility bundled with the Controller installation. You can find the `keytool` utility in the following location:

- `<controller_home>/jre/bin`

The steps assume that the keytool is in the operating system's path variable. To run the commands as shown, you first need to put the `keytool` utility in your system's path. Use the method appropriate for your operating system to add the `keytool` to your path.

While the directory paths in this topic use forward slashes, the instructions apply to both Linux and Windows Operating System environments. The steps note where there are differences in the use of commands between operating systems.

Create a Certificate and Generate a CSR

If you don't have a certificate to use for the Controller, create it as follows.

> AppDynamics requires using a X.509 digital certificate, which works with any file type.

In these steps, you generate a new certificate within the Controller's active keystore, so it has immediate effect.

The steps are intended to be used in a staging environment, and require the Controller to be shut down and restarted. Alternatively, you can generate the key as described here but in a temporary keystore rather than the Controller's active keystore. After the certificate is signed, you can import the key from the temporary keystore to the Controller's keystore.

1. At a command prompt, change directories to the following location:

   ```
   <controller_home>/appserver/glassfish/domains/domain1/config
   ```

2. Create a backup of the keystore file. For example, on Linux, you can run:

   ```
   cp keystore.jks keystore.jks.backup
   ```

   On Windows, you can use the copy command in a similar manner.
3. If it's still running, stop the Controller.
4. Delete the existing certificate with the alias s1as from the keystore:

   ```
   keytool -delete -alias s1as -keystore keystore.jks
   ```

5. Create a new key pair in the keystore using the following command. This command creates a key pair with a validity of 1825 days (5 years). Replace 1825 with the validity period appropriate for your environment, if desired.

   ```
   keytool -genkeypair -alias s1as -keyalg RSA -keystore keystore.jks -keysize 2048 -validity 1825
   ```

   Follow the on-screen instructions to configure the certificate. Note that:

   - For the first and last name, enter the domain name where the Controller is running, for example, `controller.example.com`.
   - Enter the default password for the key, `changeit`.

   This generates a self-signed certificate in the keystore. We'll generate a signing request for the certificate next. You can now restart the Controller and continue to use it. Since it still has a temporary self-signed certificate, browsers attempting to connect to the Controller UI will get a warning to the effect that its certificate could not be verified.

   > See Change Keystore Password for information on changing the default password for the keystore and certificates.

6. Generate a certificate signing request for the certificate you created as follows:

   ```
   keytool -certreq -alias s1as -keystore keystore.jks -file AppDynamics.csr
   ```

7. Submit the certificate signing request file generated by the command (`AppDynamics.csr` in our example command) to your Certificate Authority of choice.
   When it's ready, the CA will return the signed certificate and any root and intermediary certificates required for the trust chain. The response from the Certificate Authority should include any special instructions for importing the certificate if needed. If the CA supplies the certificate in text format, just copy and paste the text into a text file.
8. Import the signed certificate:

   ```
   keytool -import -trustcacerts -alias s1as -file mycert.cer -keystore keystore.jks
   ```

   This command assumes the certificate is located in a file named `mycert.cer.`

9. If you get the error "Failed to establish chain from reply", install the issuing Certificate Authority's root and any intermediate certificates into the keystore. The root CA chain establishes the validity of the CA signature on your certificate. Although most common root CA chains are included in the bundled JVM's trust store, you may need to import additional root certificates, such as certificates belonging to a private CA. To do so:

   ```
   keytool -import -alias [Any_alias] -file <path_to_root_or_intermediate_cert> -keystore <controller_home>
   /appserver/glassfish/domains/domain1/config/cacerts.jks
   ```

When done importing the certificate chain, try importing the signed certificate again.

Import an Existing Keypair into the Keystore

These steps describe how to import an existing public and private key into the Controller keystore. We'll step through this scenario assuming that the existing public and private keys need to be converted to a format compatible with Java Keystore, say from DER format to PKCS#12. You'll need to use OpenSSL to combine the public and private keys, and then use `keytool` to import the combined keys into the Controller's keystore.

Most Linux distributions include OpenSSL. If you are using Windows or your Linux distribution does not include OpenSSL, you may find more information on the OpenSSL website.

This assumes that we have the following files:

- private key: `private.key`
- signed public key: `cert.crt`
- CA root chain: `ca.crt`

The private key you use for the following steps must be in plain text format. Also, when performing the following procedures, do not attempt to associate a password to the private key as you convert it to PKCS12 keystore form. If you do, the following steps can be completed as described, but you will encounter an exception when starting up the Controller, with the error message: `java.security.UnrecoverableKeyException: Cannot recover key`.

To import an existing keypair into the Controller keystore

1. Use OpenSSL to combine your existing private key and public key into a compatible Java keystore:

   ```
   openssl pkcs12 -inkey private.key -in cert.crt -export -out keystore.p12
   ```

2. If the Controller is still running, stop it.
3. Change to the `keystore` directory:

   ```
   cd <controller_home>/appserver/glassfish/domains/domain1/config/
   ```

4. Create a backup of the `keystore` file. For example, on Linux, you can run:

   ```
   cp keystore.jks keystore.jks.backup
   ```

   On Windows, you can use the copy command in a similar manner.
5. Delete the self-signed certificate with alias `s1as` from the default keystore:

   ```
   keytool -delete -alias s1as -keystore keystore.jks
   ```

6. Import the PKCS #12 key into the default keystore:

   ```
   keytool -importkeystore -srckeystore keystore.p12 -srcstoretype pkcs12 -destkeystore keystore.jks -
   deststoretype JKS
   ```

7. Update the alias name on the key pair you just imported:

   > ⚠ The alias name should be `s1as`. Do not change it from this name.

   ```
   keytool -changealias -alias "1" -destalias "s1as" -keystore keystore.jks
   ```

8. Change the password of the imported private key:

   ```
   keytool -keypasswd -keystore keystore.jks -alias s1as -keypass <.p12_file_password> -new <password>
   ```

   For the new private key password, use the default (`changeit`) or the master password set as described in Change Keystore Password, if changed.
9. If you get the error "Failed to establish chain from reply", install the issuing Certificate Authority's root and any intermediate certificates into the keystore. The root CA chain establishes the validity of the CA signature on your certificate. Although most common root CA chains are included in the `cacerts.jks` truststore, you may need to import additional root certificates. To do so:

   ```
   keytool -import -alias <Any_alias> -file <path_to_root_or_intermediate_cert> -keystore <controller_home>
   /appserver/glassfish/domains/domain1/config/cacerts.jks
   ```

   When done, try importing the signed certificate again.
10. Start the Controller.

Verify the Use of SSL

To make sure the configuration works, use a browser to connect to the Controller over the default secure port, port 8181:

```
https://<controller_host>:8181/controller
```

Make sure the Controller entry page loads in the browser correctly. Also, verify that the browser indicates a secure connection. Most browsers display a lock icon next to the URL to indicate a secure connection.

After changing the certificate on the Controller, you will need to import the public key of the certificate to the agent truststore. For information on how to do this, see the topic specific for the agent type:

- EUM aggregator: Troubleshoot Your EUM Setup
- Java Agent: Enable SSL (Java)
- .NET: Enable SSL (.NET)

If there is no proxy configured and the agent is reporting to the Controller itself, then the following changes are also mandatory:

1. Run the following command:

   ```
   platform-admin.sh stop-controller-appserver
   ```

   On Windows, run this command from an elevated command prompt (which you can open by right-clicking on the Command Prompt icon in the Windows Start menu and choosing **Run as administrator**):

   ```
   platform-admin.exe cli stop-controller-appserver
   ```

2. Search for the following properties in `<controller_home>/appserver/glassfish/domains/domain1/config/domain.xml`, and replace the port with the SSL port, as the non-secure port is disabled:

   > ⚠ You should also edit the domain.xml configurations on the Controller Settings page of the Enterprise Console to retain your settings. See Update Platform Configurations for more information.

   ```
   -Dappdynamics.controller.port=
   -Dappdynamics.controller.services.port=
   ```

3. In the following property, change the protocol from HTTP to HTTPS, and change the port to the secure port.

   ```
   -Dappdynamics.controller.ui.deeplink.url=
   ```

   You can also use REST API to update the deeplink URL:

```
curl -k --basic --user root@system --header "Content-Type: application/json" --data '
{ "controllerURL": "https://<controller>:<ssl_port>" }' https://<controller>:<ssl_port>/controller/rest
/accounts/<ACCOUNT-NAME>/update-controller-url
```

4. Add the following JVM argument anywhere above or below the above JVM arguments to ensure the internal agent connects using SSL.

```
-Dappdynamics.controller.ssl.enabled=true
```

5. Run the following command:

```
platform-admin.sh start-controller-appserver
```

On Windows, run the following in an elevated command prompt:

```
platform-admin.exe cli start-controller-appserver
```

You can also use the `modifyJVMOptions.sh` script to make the changes.

Change Keystore Password

The default password for the keystore used by the Controller is `changeit`. This is the default password for the Glassfish keystore, and is a well-known (and thus insecure) password. For a secure installation, you need to change it.

> Changing the password in this manner does not affect the administration password you use to access the Glassfish administration console. See Update the Root User and Glassfish Admin Passwords for information on changing this password.

To change the password you must use the Glassfish administration tool (rather than the `keytool` utility directly). Using the Glassfish administration tool allows the Glassfish instance to access the keys at runtime.

If you change the keystore password directly using the `keytool`, the Controller generates the following error message at start up:

```
Caused by: java.lang.IllegalStateException: Keystore was tampered with,
or password was incorrect
```

If you encounter this scenario, change the password using the `asadmin` utility.

To change Glassfish passwords

1. Stop the Controller.
2. Change the Glassfish master password:

```
<controller_home>/appserver/glassfish/bin/asadmin change-master-password --savemasterpassword=true
```

Changing the master password with `asadmin` changes the password for the domain-passwords, `cacerts.jks`, and `keystore.jks` stores (including the s1as, reporting-instance, and glassfish-instance private keys in `keystore.jks`).

However if you customized any additional keys or existing key passwords, and they do not match the master password, when you change the master password, the following error is generated:

```
./asadmin change-master-password --savemasterpassword=true
Enter the new master password>
Enter the new master password again>
Caught an Exception: {0}
Command change-master-password executed successfully.
```

This indicates that the store password for `keystore.jks` has been set to the master password, but one or more of the private keys still has a different key password and do not match the master password. This prevents the Controller application from starting and generates the following error:

```
java.lang.Error: java.security.UnrecoverableKeyException: Cannot recover key
```

1. To resolve this issue, update each of the private key passwords in `keystore.jks` (s1as, reporting-instance, and glassfish-instance) to ensure that they match the master password by entering the following `keytool` command:

   > Replace the `<JRE_HOME>`, `<alias_name>`, and `<controller_home>` variables with your information before executing the `keytool` command.

   ```
   <JRE_HOME>/bin/keytool -keypasswd -alias <alias_name> -keystore <controller_home>/appserver/glassfish
   /domains/domain1/config/keystore.jks -storepass <master_password>
   ```
2. To confirm that the default values for `<alias_name>` are `s1as,` reporting-instance, and glassfish-instance, execute the following command to list the contents of `keystore.jks`:

   > Replace the `<JRE_HOME>`, `<alias_name>`, and `<controller_home>` variables with your information before executing the `keytool` command.

   ```
   <JRE_HOME>/bin/keytool -list -keystore <controller_home>/appserver/glassfish/domains/domain1/config
   /keystore.jks -storepass <master_password>
   ```

   If the key password matches the master password, the message `"Passwords must differ*"*` displays when entering the new key password. This validates that the key password was set correctly.
3. Restart the Controller and ensure it starts without errors.

Updating an Expired Certificate

The steps to renew an expired or soon-to-expire certificate are similar to those for replacing the default certificate, as documented in Create a Certificate and Generate a CSR. To update the expired certificate:

1. Back up the existing keystore.
2. At a command prompt, change directories to the following location:

   ```
   <controller_home>/appserver/glassfish/domains/domain1/config
   ```

3. Create a backup of the keystore file.

   a. On Linux, you can run the following command:

      ```
      cp keystore.jks keystore.jks.backup
      ```

   b. On Windows, you can use the copy command in a similar manner.

      > If the controller is still running, stop the controller.

4. Since you already have a Java keystore, run the following command to issue a certificate signing request. You should use this keystore for the `csr` and **not create a new one**. You will be importing the new certificate into this keystore.

   ```
   keytool -certreq -alias s1as -keystore keystore.jks -file AppDynamics.csr
   ```

5. Submit the certificate signing request file `Appdynamics.csr` generated by the above example command to your Certificate Authority of choice.

   a. When it's ready, the Certificate Authority will return the signed certificate and any root and intermediary certificates required for the trust chain.
   b. The response from the Certificate Authority should include instructions for importing the certificate if needed.

   > If the Certificate Authority supplies the certificate in text format, copy and paste the text into a text file.

6. You can list out the obtained certificate as follows if it is not in text format.

```
keytool -printcert -v -file <your obtained certificate>
```

7. Import the signed certificate obtained into the keystore that you already have.

   ```
   keytool -import -alias s1as -file <your obtained certificate> -keystore keystore.jks
   ```

   a. The imported certificate will replace the old one, provided you use the same alias as the previous one.
   b. Sometimes the root and intermediate certificates of the certification authority are also expired. If that's the case, you will see the message `Failed to establish chain from reply`.
8. If the root and intermediate certificates of the certification authority are expired, they also have to be imported in your `cacerts.jks` so that the chain of trust can be established. You can follow your certification authority’s instructions to download the root and intermediate certificates.

   a. Keep the same alias as before for root and intermediate when you import these certificates into `cacerts.jks`.

   ```
   keytool -import -alias <previous alias used for the certificate> -file
   <path_to_root_or_intermediate_cert> -keystore <controller_home>/appserver/glassfish/domains/domain1
   /config/cacerts.jks
   ```

Trust Stores and Keystores

- Java trust store, cacerts, contain root certificates of well-known certification authorities. The validity of a certificate presented during the TLS/SSL (Transport Layer Security/Secure Sockets Layer) session is checked from `cacerts.jks`. There are no private keys or passwords in cacerts. They will contain the intermediate and root certificates of certification authorities.
- Java Keystore is used to store private key and the identify certificate for the server, which means that the keystore is used to store your server’s credentials.

Set the Security Protocol

Related Pages:

- Agent-to-Controller Connections

This page describes the security protocol used by an on-premises Controller, and how you can modify it.

Default Security Protocol

The Controller secures connections using TLSv1.2 by default. However, you can change the security protocols used by the Controller if needed. For instance, you need to change the protocol if you are using agents that don't support TLSv1.2. These agents include:

- Java Agent version 3.8.1 or earlier (see Agent and Controller Compatibility for complete SSL compatibility information)
- .NET Agent running on .NET Framework 4.5 or earlier

If upgrading the agents or .NET framework is not possible, you will need to enable TLSv1 and SSL3 on the Controller using the `asadmin` command-line utility. To use the utility, you will need to supply the password configured for the root user for the Controller.

These changes require a restart of the Controller application server, which results in a brief service downtime. You may wish to apply these change when the downtime will have the least impact.

To maintain a secure environment, APIs that are downstream of the Controller should also use TLS. If SSL3 is required, you can enable it. See the Oracle JDK 8 documentation.

Enable TLS for a Controller

1. Open a browser and navigate to the Enterprise Console GUI:

   ```
   http(s)://<hostname>:<port>
   ```

   9191 is the default port.
2. Navigate to AppServer Configurations by choosing the platform, **Configurations**, **Controller Settings**, and **Appserver Configurations**.
3. In the Domain Protocols box on the JVM Options tab, edit the `configs.config.server-config.network-config.protocols.protocol.http-listener-2.ssl.tls-enabled=false` parameter to `true`.
4. Click **Save**.

> ⓘ You do not need to restart the Controller application server since the configuration change job automatically does so for you.

Enabling Stronger Encryption Keys

By default, the Controller's embedded Java runtime only supports up to 128-bit encryption key lengths for secure connections. You can, however, enable up to 256-bit encryption keys so the Controller can establish connections using the stronger ciphers.

To enable stronger keys in encryption keys in the Controller, follow the instructions for the Controller version you are running.

After restarting the Controller app server, the following cipher suites become available:

- `TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`
- `TLS_RSA_WITH_AES_256_CBC_SHA256`
- `TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384`
- `TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384`
- `TLS_DHE_RSA_WITH_AES_256_CBC_SHA256`
- `TLS_DHE_DSS_WITH_AES_256_CBC_SHA256`
- `TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA`
- `TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA`
- `TLS_RSA_WITH_AES_256_CBC_SHA`
- `TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA`
- `TLS_ECDH_RSA_WITH_AES_256_CBC_SHA`

- `TLS_DHE_RSA_WITH_AES_256_CBC_SHA`
- `TLS_DHE_DSS_WITH_AES_256_CBC_SHA`

Configure a Controller SSH Key

You can set up SSH (Secure Shell) with public/private key pairs so that you do not have to type the password each time you access a Controller machine by SSH. Setting up keys allows scripts and automation processes to access the Controller easily. You can generate DSA or if you want stronger encryption, RSA keys.

Set up SSH Key Pairs Using DSA

1. Run the `ssh` command that sets up the key pair:

   ```
   % ssh-keygen -t dsa
   ```

2. At the following prompt, press **Enter** to accept the default key location, or type another:

   ```
   Generating public/private dsa key pair.

   Enter file in which to save the key (~/.ssh/id_dsa):
   ```

3. Press return at the password prompt:

   ```
   Enter passphrase (empty for no passphrase):
   ```

4. Press **Return** again to confirm the password:

   ```
   Enter same passphrase again:
   ```

   You should see the following information:

   ```
   Your identification has  been saved in ~/.ssh/id_dsa

   Your public key has been saved in  ~/.ssh/id_dsa.pub

   The key fingerprint is: <Some really long string>
   ```

   If SSH continues to prompt you for your password, verify your permissions in your remote `.ssh` directory. It should have only your own read/write /access permission (octal 700):

   ```
   % chmod 700 ~/.ssh
   ```

5. Open the local `~/.ssh/id_dsa.pub` file and paste its contents into the `~/.ssh/authorized_keys` file on the remote host.
6. Update the permissions on the `authorized_keys` file on the remote host as follows:

   ```
   % chmod 600 ~/.ssh/authorized_keys
   ```

Set up SSH Key Pairs Using RSA

Run the `ssh` command that sets up the key pair:

```
% ssh-keygen -t rsa
```

The generated files will be named `id_rsa` and `id_rsa.pub`, instead of `id_dsa` and `id_dsa.pub`.

Otherwise, the remaining steps are identical to those beginning with step 2 in the steps above.

Controller Secure Credential Store

Related pages:

- Controller Data and Backups

The Controller creates a secure credential keystore that holds a secret key used to encrypt credentials.

Stored Credentials

The secure credential store manages the following credentials:

- LDAP authentication user password. See LDAP Authentication.
- Database collector credentials, including database user password and the machine user password.
- SMTP server/Email passwords.
- AppDynamics account access keys.

Back up the credential store as part of your normal backup procedures for the Controller, as described in the following section.

Secure Credential Store Backup

Make sure your Controller backup plan includes the secure credential keystore file `.appd.scskeystore.` In the case that the secure credential keystore file should become corrupted, restore the `.appd.scskeystore` file from backup.

If you run the Controller in high availability mode, both the primary Controller and the secondary Controller must use the same secure credential keystore file. If you use an HA deployment strategy, verify that it propagates the secure credential keystore file from the primary to the secondary.

Replace a Compromised Secure Credential Store

The following steps describe how to replace a secure credential store. It assumes the following:

- You have a single-tenant Controller installation.
- You know the plain-text value of your Account Access Key. You can view the access key in the Controller under **Settings > License**.

As detailed in the sections that follow, the steps are broken into these parts:

1. Create a new secure credential store.
2. Update the Controller with the password of the new secure credential store.
3. Update the account access key.
4. Update the account access key for the system account.
5. Restart the Controller and update passwords.

Create a new Secure Credential Store

1. Rename the existing secure credential keystore file.
2. Initialize a new secure credential keystore using the secure credential store utility.

   By default the utility installs to: `<controller_home>/tools/lib/scs-tool.jar`

   For example:

   ```
   /controller/jre8/bin/java -jar ./scs-tool.jar generate_ks -filename '<controller_home>/.appd.
   scskeystore' -storepass 'MyCredentialStorePassword'
   ```

   The secure credential store utility confirms it created and initialized the keystore:

   ```
   Successfully created and initialized new KeyStore file: /opt/appdynamics/Controller/.appd.scskeystore
   Verification - New KeyStore file: /opt/appdynamics/Controller/.appd.scskeystore is properly initialized.
   ```

Update the Controller with the new Secure Credential Store Password

1. Shut down the Controller.
2. Obfuscate the password you used to initialize the secure credential keystore:

```
/controller/jre8/bin/java -jar <controller_home>/tools/lib/scs-tool.jar obfuscate -plaintext
'<Secure_Credential_Store_Password>'
```

For example:

```
/controller/jre8/bin/java -jar /opt/appdynamics/Platform/controller/tools/lib/scs-tool.jar obfuscate -
plaintext 'MyCredentialStorePassword'
```

The secure credential store utility writes out an obfuscated password for use in the Controller configuration. For example:

```
s_gsnwR6+LDch8JBf1RamiBoWfMvjjipkrtJMZXAYEkw8=
```

3. Log in as the root user:

```
<controller_home>/bin/controller.sh login-db
```

> On Windows, use controller.bat.

4. Update the secure credential keystore password to the newly obfuscated password:

```
UPDATE global_configuration_cluster
SET value = '<obfuscated_secure_credential_keystore_password>'
WHERE name = 'scs.keystore.password';
```

Update the Account Access Key

1. Log in as the root user:

```
<controller_home>/bin/controller.sh login-db
```

> On Windows, use controller.bat.

2. Update the account access key for the account to the plain text string. When the Controller starts, it will encrypt the account access key:

```
UPDATE account
SET access_key = '<plain_text_account_access_key>',
    encryption_scheme = NULL
WHERE id = <account_id>;
```

> You can get the account id by running the following query: `select id account_id,name account_name,access_key, encryption_scheme from account;`

3. Only if you changed the plain text value of the account access key. Update the account access key for the agent users:

```
UPDATE user
SET encrypted_password = SHA1('<plain_text_account_access_key>')
WHERE account_id = <account_id>
AND name = 'singularity-agent';
```

If you changed the plain text value of the account access key, you need to update the access key for all the agents.

> ⓘ The access key belongs to the "customer1" account in a single-tenant Controller and the "default" account in a multi-tenant Controller. In addition, `account_id` is the account id of the "customer1" account in a single-tenant Controller and the "default" account in a multi-tenant Controller.

4. If you have default license rules, update the account access key using `v1_license_rules` API.

> ⚠ For earlier Controller versions, you must use browser tools to migrate license rules.

Update the Account Access Key for the System Account

1. Generate the new access key for the system account:

```
../jre/1.8.0_152/bin/java -jar ./tools/lib/scs-tool.jar encrypt -filename ./.appd.scskeystore -storepass
'REPLACE_TO_NOT_OBFUSCATED_STOREPASS_VALUE' -plaintext 'NEW_SYSTEM_ACCOUNT_ACCESS_KEY'
```

2. Once you have generated the system account access key:
   a. Edit the `controller-info.xml` file to add your specific information:

```
<controller-dir>/appserver/glassfish/domains/domain1/appagent/ver4.X.X.X/conf/controller-info.xml
```

   b. Edit the `credential-store-password` value with the obfuscated `storepass` value.
   c. Edit the `account-access-key` with new encrypted access key value.
   d. Run SQL:

```
update account set access_key='ENCRYPTED_SYSTEM_ACCOUNT_ACCESS_KEY' where id=1; update mds_account.
account set access_key='ENCRYPTED_SYSTEM_ACCOUNT_ACCESS_KEY' where id='00000000-0000-0000-0000-
000000000001'; update mds_account.account set access_key='ENCRYPTED_SYSTEM_ACCOUNT_ACCESS_KEY'
where id='00000000-0000-0000-0000-000000000002';
```

   e. Stop `appserver`.
   f. Start `appserver`.

> ⓘ If you use LDAP, DBmon, or HTTP Request Actions and Templates, then you must also reconfigure those components with the same passwords to ensure that they are encrypted with new SCS key.

Restart the Controller and Update Passwords

1. Restart the Controller.
2. Log in to the Controller as a user with the following permissions:
   - Administer users, groups, roles, authentication, etc.
   - Configure Email / SMS.
3. As necessary, re-enter the following passwords:
   - LDAP authentication user password. See Configure Authentication Using LDAP.
   - Database collector credentials:
     - database user password. See "Add a Database Collector" on Configure Database Collectors.
     - machine user password. See "Configure the Database Agent to Monitor Server Hardware" on Configure Database Collectors.
   - SMTP server / Email password. See Enable an Email Server.

Mutual Authentication

In addition to implementing Server Authentication, you can also implement mutual (client and server) authentication. Client authentication enables the Controller to ensure that only authorized and verified agents can establish connections. These procedures outline the workflow to implement mutual authentication.

Before Starting

- Theses agents support client authentication:
    - Java Agent
    - Database Agent
    - Machine Agent
    - .NET Agent for Windows

      > Excludes Azure PaaS environments, such as Azure App Services.

- It is good practice to set up and verify client authentication on one agent first. After you confirm that client authentication is working for that agent, proceed with configuring additional agents.
- If you have a "hybrid" environment, with Server Authentication only for some agents and Server and Client Authentication for others, you might want to set up and configure multiple HTTP Listeners in Glassfish: one for Server Authentication only, and another for both Server and Client Authentication.
- The procedures described on this page use the default key and keystore password (`changeit`) for the keystore. Before you proceed with this workflow, it is good practice to:

    1. Change this default password, as described in "Change Keystore Password" under Controller SSL and Certificates.
    2. Use the new password when you perform these procedures.
- Instead of using plain text passwords in the procedures, you can specify encrypted or obscured passwords as described in Encrypt Agent Credentials.

Set Up Client Authentication

These steps describe how to set up Client-based Authentication:

1. Set up server authentication on the Controller.
2. Set up server authentication on agents.
3. Set up a client keystore on the Controller.
4. Configure agents to access the client keystore on the Controller.
5. Enable client authentication on the Controller.

Set Up Server Authentication on the Controller

1. Open a CLI window on the Controller host and cd to the `<controller-keystore-home>` directory: `<controller-home>/appserver/glassfish/domains/domain1/config`
2. Create a trusted root certificate store to preserve the Controller public key. The following command exports the public key (`s1as`) and stores it in the new certificate store file `server.cer`.

    ```
    <java-home>/bin/keytool -export -alias s1as -file server.cer \
                                 -keystore keystore.jks
                                 -storepass changeit
    ```

3. (*Optional*) To view information about the public and private keys in `keystore.jks`, enter the following command:

    ```
    <java-home>/bin/keytool -list -v -alias s1as  \
                            -keystore ./keystore.jks \
                            -storepass  changeit
    ```

4. Import the Controller public key from `server.cer` into an agent keystore. The following command creates a new keystore (`agent-truststore.jks`); the Controller sends the public key from this keystore to agents when they set up an SSL connection.

```
<java-home>/bin/keytool -import -v -alias controller_alias -file server.cer \
                        -keystore agent_truststore.jks \
                        -storepass changeit
```

5. This command displays the certificate and asks if you trust it. Answer **`yes`**.

Set Up Server Authentication on Agents

For each authorized agent:

1. Copy the `agent_truststore.jks` file from the Controller to the root directory of the agent (`<machine-agent-home>` or `<java-agent-home>` or `<database-agent-home>`).
2. For each authorized agent, specify the following properties in the `<agent-home>/conf/controller-info.xml` file as follows:
    - `<controller-ssl-enabled>true</controller-ssl-enabled>`
    - `<controller-port>443</controller-port>`
    - `<controller-keystore-filename>agent_truststore.jks</controller-keystore-filename>`
    - `<controller-keystore-password>changeit</controller-keystore-password>`

Set Up a Client Keystore on the Controller

In this procedure, you create a signed certificate and import it into the client keystore. These steps use the Controller to sign the certificate, but you can also use a third-party Certificate Authority (CA).

1. (*Optional*) To view information about the public and private key in the Controller keystore, enter:

```
<java-home>/bin/keytool -list -v -alias s1as  \
                        -keystore ./keystore.jks \
                        -storepass changeit
```

2. Create a keystore (`clientkeystore.jks`) that includes the Controller public/private keypair. In `<controller-keystore-home>`, enter:

```
<java-home>/bin/keytool -genkey -alias client-alias -keyalg RSA \
                        -keystore clientkeystore.jks \
                        -storepass changeit \
                        -keypass changeit
```

The keytool prompts you for your name, organization, and other information it needs to generate the key. AppDynamics App Agents use SunX509 as the default keystore factory algorithm. If keystores in your environment use something other than SunX509, you need to specify the algorithm to the App agent. You can do so using the system property `appdynamics.agent.ssl.keymanager.factory.algorithm`. For example, to set the algorithm to PKIX, add this to the startup command of the agent-monitored JVM:

```
-Dappdynamics.agent.ssl.keymanager.factory.algorithm=PKIX
```

3. Generate a certificate signing request (`client.csr`) that can be signed by a Certificate Authority (CA).

```
<java-home>/bin/keytool -certreq -v -alias client-alias -file client.csr \
                        -keystore clientkeystore.jks \
                        -storepass changeit \
                        -keypass changeit
```

4. Get the request (`client.csr`) signed by a trusted CA. This command uses the Controller as a CA, which creates a new file (`signedClient.cer`) with the Controller-signed certificate.

```
<java-home>/bin/keytool -gencert -infile ./client.csr -outfile signedClient.cer -alias s1as \
                        -keystore ./keystore.jks \
                        -storepass changeit \
                        -keypass changeit
```

5. (*Optional*) To view information about the signed certificate, enter:

```
<java-home>/bin/keytool -printcert -v -file ./signedClient.cer
```

6. Verify that the certificate is signed by the authentic Certificate Authority. You can:
   - Copy the public key of the signing authority into the trusted root set, or
   - Import the public key of the signing authority into the client keystore.

   This command does the latter by importing the Controller public key from `server.cer` into `clientkeystore.jks`.

   ```
   <java-home>/bin/keytool -import -v -alias controller_alias -file server.cer \
                           -keystore clientkeystore.jks \
                           -storepass changeit
   ```

   This command asks if you trust the certificate; when you enter `yes`, this message should display:

   ```
            Certificate was added to keystore
            [Storing clientkeystore.jks]
   ```
7. (*Optional*) To view the contents of `clientkeystore.jks`, enter:

   ```
   <java-home>/bin/keytool -list -v -keystore clientkeystore.jks -storepass changeit
   ```

   The keystore should show entries for `controller-alias` and `client-alias` (which is still unsigned).
8. Import the signed public key certificate into the client keystore. This command imports `signedClient.cer` into `clientkeystore.jks`.

   ```
   <java-home>/bin/keytool -importcert -v -alias client-alias -file ./signedClient.cer \
                           -keystore clientkeystore.jks \
                           -storepass changeit \
                           -keypass changeit
   ```

   You now have a password-protected `clientkeystore.jks` file on the Controller with a signed certificate that verifies the Controller's authenticity.
9. Verify that the trusted root certificate on the Controller includes the public key of the signing authority. This procedure used the Controller as the Certificate Authority, so the public key is already included. To verify, enter:

   ```
   <java-home>/bin/keytool -list -v -alias client-alias \
                           -keystore clientkeystore.jks -storepass changeit
   ```

   The public key of the signing authority should now be part of the agent's public key certificate.

Configure Agents to Access the Client Keystore on the Controller

For each authorized agent:

1. Copy the `clientkeystore.jks` file from the Controller to the following directory on the agent:
   - Database agent: `<database agent home>/conf`
   - Java agent: `<java-agent-home>/conf`
   - Machine Agent: `<machine-agent-home>`
2. Specify these properties in the `<agent-home>/conf/controller-info.xml` file as follows:

   - `<use-ssl-client-auth>`**true**`</use-ssl-client-auth>`
   - `<asymmetric-keystore-filename>`**clientkeystore.jks**`</asymmetric-keystore-filename>`
   - `<asymmetric-keystore-password>`**changeit**`</asymmetric-keystore-password>`
   - `<asymmetric-key-password>`**changeit**`</asymmetric-key-password>`
   - `<asymmetric-key-alias>`**client-alias**`</asymmetric-key-alias>`

Enable Client Authentication on the Controller

From the Controller:

1. If it is not currently running, start the Controller.
2. Open a CLI panel and `cd` to `<controller-home>/appserver/glassfish/bin`.

3. To start the AppServer Admin tool, enter: `./asadmin`
   You should now see the asadmin prompt: `asadmin>`
4. Enter: `list-http-listeners`
   This lists the available set of HTTP listeners. For example:
   ```
    http-listener-1
   http-listener-2
   admin-listener
   ```
5. Configure one of these listeners by running the following commands. In this example, we are configuring `http-listener-2`:

   ```
   set configs.config.server-config.network-config.protocols.protocol.http-listener-2.ssl.key-store=keystore.jks
   set configs.config.server-config.network-config.protocols.protocol.http-listener-2.ssl.client-auth-enabled=true
   set configs.config.server-config.network-config.protocols.protocol.http-listener-2.ssl.trust-store=cacerts.jks
   ```

6. To verify that the properties are set correctly, run the following command. Again, this example assumes `http-listener-2`.

   ```
    get configs.config.server-config.network-config.protocols.protocol.http-listener-2.*
   ```

7. Restart the Controller.

Upgrade Platform Components

The information and instructions below are intended for On-Premise deployments only. SaaS deployments are managed by AppDynamics.

Related pages:

- Upgrade the Controller Using the Enterprise Console
- Upgrade the Events Service
- Upgrade the Production EUM Server
- Upgrade the Controller Using the Enterprise Console
- Discovery and Upgrade Quick Start

Before Upgrading

Before you upgrade to a newer version of the platform, complete the following tasks:

- Review the Product Announcements and Alerts page.
- Review the latest Release Notes, as well as the release notes for intermediate versions between the current version and the version you are upgrading to.
- Review the compatibility support page.
- Verify that you meet the requirements described in the Platform Requirements for the components you use.

Upgrade Order

Follow the instructions for each service in this order:

1. Upgrade the Enterprise Console
2. Upgrade the Events Service
3. Upgrade the EUM Server
4. Upgrade the Controller

Discover Existing Components

Related pages:

- Discovery and Upgrade Quick Start
- Administer the Enterprise Console

You can use the Enterprise Console to discover and upgrade existing AppDynamics components, such as a Controller or Events Service. After you discover the components, they can be added to the platform and be managed by the application. This process can be performed with the GUI or command line. You can discover Controllers that are version 4.1 or later.

Before you discover existing components, you need the following information:

Controller

- Controller root user password
- Installation directory
- Database password
- Host name or IP address

Events Service

- Host names or IP addresses
- Installation directory. Note that if you have an Events Service cluster, this directory is the same on every node.
- Host credentials (SSH key file and username)

Using the GUI

You can use the Custom Install Discover & Upgrade option in the GUI to create a platform and discover a Controller and Events Service. Alternatively, if you have already created a platform, you must add credentials and hosts to the platform before you can perform discovery. Then, discover the Controller or Events Service on the page for that component. For more information about how to add credentials and hosts, see Administer the Enterprise Console.

When the Enterprise Console discovers a component, it also checks to see if an upgrade is available and performs the upgrade.

Using the Command Line

Controller

Use the discover-upgrade-controller command to discover and upgrade a Controller:

```
bin/platform-admin.sh discover-upgrade-controller --host <host> --controller-root-password <root user password
for Controller> --installation-dir <Controller installation directory> --db-root-password <password for
Controller database>
```

If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`

Events Service

The discover-events-service command can discover and upgrade an Events Service.

```
bin/platform-admin.sh discover-events-service --hosts <host 1> <host 2> <host 3> --installation-dir <Events
Service installation directory>
```

Instead of listing each host, you can specify a line-separated list in a text file:

```
bin/platform-admin.sh discover-events-service -n <file path for the host file> --installation-dir <Events
Service installation directory>
```

If your upgrade fails, you can resume by passing the flag `useCheckpoint=true` as an argument after `--args.`