Controller Deployment

On this page:

Related pages:

- Install the Controller Using the CLI

This topic 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.
- For high volume environments, tune the OS and Controller settings for high 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 o ffer 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 Java Agent Deployment
- Unattended Installation for .NET

Network Requirements

Deploying the Controller often calls for configuration changes to existing network components, such as to 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 Manage Users and Groups for more information.

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 to 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 single-tenant mode for most installations.

For more information, see Controller Accounts (Multi-Tenancy).