.

Load Balance Events Service Traffic

On this page:

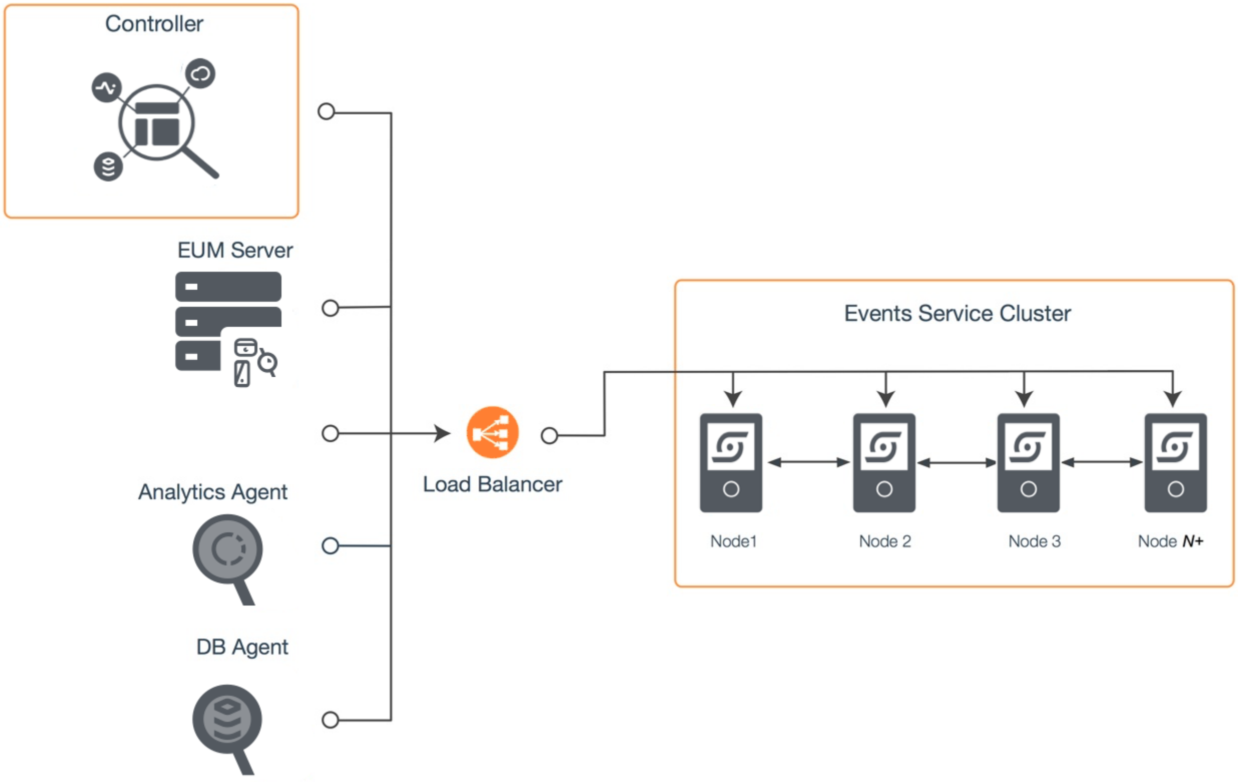
This topic takes you through the sample configuration for a load balancer for the Events Service. It introduces you to the concepts and requirements around load balancing Events Service traffic.

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, master 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 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.

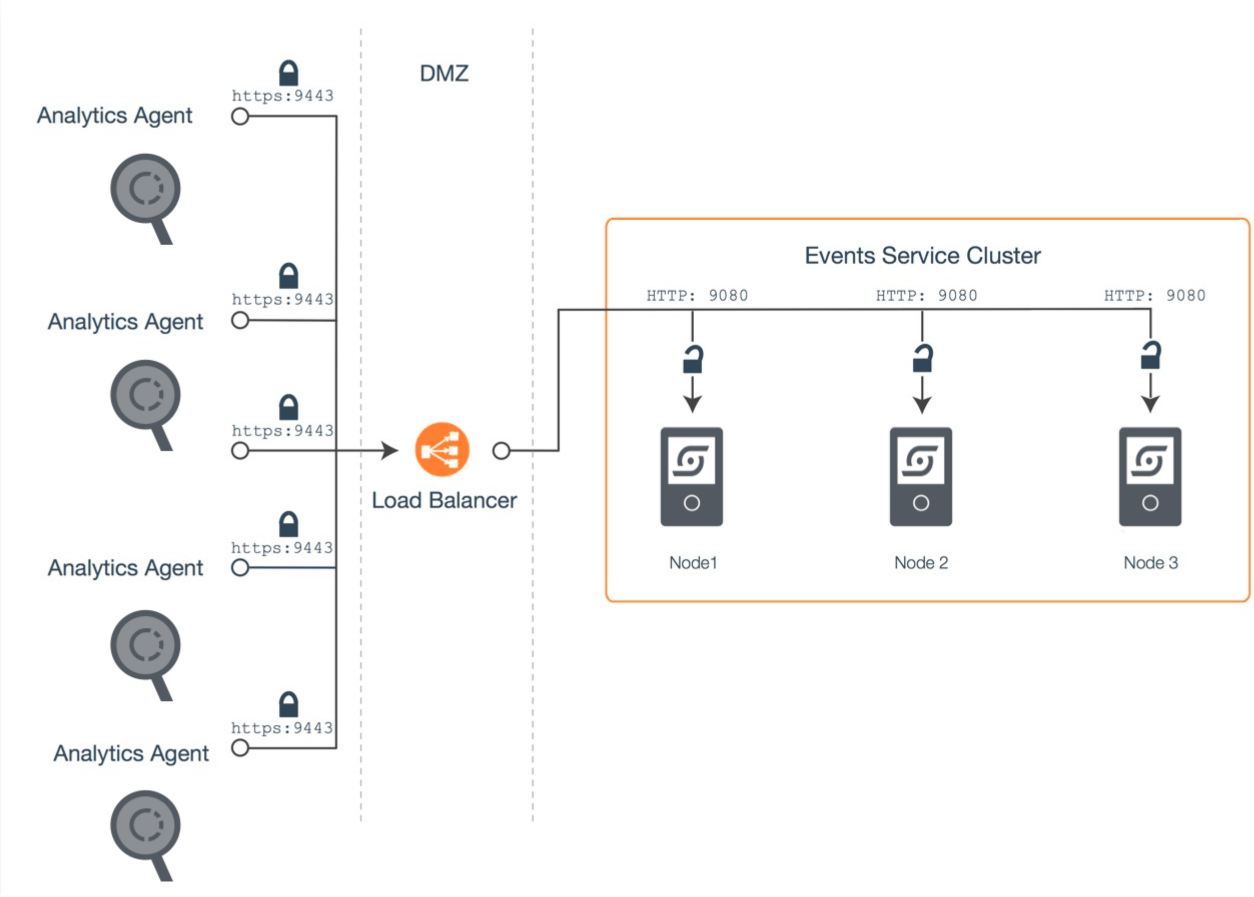
HA Proxy Sample Configuration: Terminating SSL at the Load Balancer

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. Also, as shown, include any intermediate CA certs, if present, when created 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>:9443/v1
```

7. Start the Analytics Agent (or restart it, if it is already running).
8. 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.analytics.server.store.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.