

Bridge®

# Pivotal Cloud Foundry JMX

Version 1.7

## User's Guide

© 2018 Pivotal Software, Inc.


## Table of Contents

Table of Contents	2
Pivotal Cloud Foundry JMX Bridge	3
Deploying JMX Bridge	4
Using JMX Bridge	10
Using SSL with a Self-Signed Certificate in JMX Bridge	12
Troubleshooting and Uninstalling JMX Bridge	15
Application Security Groups	17
JMX Bridge Release Notes and Known Issues	18

## Pivotal Cloud Foundry JMX Bridge

The Pivotal Cloud Foundry (PCF) JMX Bridge collects and exposes system data from Cloud Foundry components via a JMX endpoint. You can use this system data to monitor your installation and assist in troubleshooting.

The JMX Bridge tool is composed of the following three virtual machines:

- The JMX provider
- A VM that governs compilation
- A nozzle for the [Loggregator Firehose](#) 

## Product Snapshot

Current PCF JMX Bridge Details

**Version:** 1.7.10

**Release Date:** 03/09/2017

**Compatible Ops Manager Version(s) for Install:** >= 1.7.0

**Compatible Ops Manager Version(s) for Upgrade:** >= 1.7.0

**Compatible Elastic Runtime Version(s):** >= 1.7.0

**AWS support?** Yes

**vSphere support?** Yes

**OpenStack support?** Yes

## JMX Bridge User Guide

- [Deploying JMX Bridge](#)
- [Using JMX Bridge](#)
- [Using SSL with a Self-Signed Certificate in JMX Bridge](#)
- [Troubleshooting and Uninstalling JMX Bridge](#)
- [Application Security Groups](#)
- [Release Notes and Known Issues](#)

## Deploying JMX Bridge

Page last updated:

The JMX Bridge tool is a JMX extension for Elastic Runtime. Follow the instructions below to deploy JMX Bridge using the [Pivotal Cloud Foundry](#) (PCF) Operations Manager.

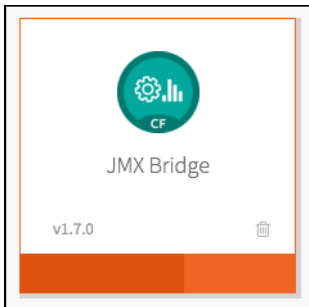
### Step 1: Install the JMX Bridge Tile

**Note:** To use the Firehose nozzle, you must install [Elastic Runtime](#) before installing JMX Bridge. If you install in reverse order, JMX will display numerous errors, and you may experience a bad connection between the Firehose nozzle and the JMX provider.

1. [Download JMX Bridge](#).
2. Import JMX Bridge into Ops Manager by following the instructions for [Adding and Importing Products](#).

**Note:** To upgrade from JMX Bridge from 1.6.x to 1.7.x, you must have Ops Manager version 1.7.8 or later.

3. On the Installation Dashboard, click the **JMX Bridge** tile.



The orange bar on the **JMX Bridge** tile indicates that the product requires configuration.

### Step 2: Assign Availability Zones and Networks

1. Select **Assign AZs and Networks**. This section shows the Availability Zones that you [create](#) when configuring Ops Manager Director.
2. (**vSphere and Amazon Web Services Only**) Select an Availability Zone under **Place singleton jobs in**. Ops Manager runs Metrics jobs with a single instance in this Availability Zone.
3. (**vSphere and Amazon Web Services Only**) Select one or more Availability Zones under **Balance other jobs in**. Ops Manager balances instances of Metrics jobs with more than one instance across the Availability Zones that you specify.

Installation Dashboard

## JMX Bridge

Settings Status Credentials Logs

Assign AZs and Networks

### AZ and Network Assignments

**JMX Provider**

**OpenTSDB Firehose Nozzle**

**Resource Config**

**Stemcell**

Place singleton jobs in

☒ us-west-1c

☐ us-west-1d

Balance other jobs in

☒ us-west-1c

☒ us-west-1d

Network

PCFNetwork

Save

4. Use the drop-down menu to select a Network.

**Note:** JMX Bridge uses the default Assigned Network if you do not select a different network.

5. Click **Save**.

**Note:** When you save this form, a verification error displays because the PCF security group blocks ICMP. You can ignore this error.



## Step 3: Configure JMX Provider

1. Select **JMX Provider**.
2. Enter a new username and password into the **JMX Provider credentials** username and password fields.
3. Record these credentials. You use these to connect JMX clients to the JMX Provider.

## (Optional) Step 4: Configure SSL

1. Select the **Enable SSL** checkbox. Enabling SSL requires JMX clients to use SSL to connect to the JMX Provider. If SSL is not enabled, JMX clients can connect to the JMX Provider without SSL credentials.

If you select the **Enable SSL** checkbox, you must also provide an SSL certificate and private key. There are two ways to provide an SSL certificate and private key:

- If you are using a signed certificate, paste an X.509 certificate in the **Certificate PEM** field and a PKCS#1 private key in the **Private Key** field.
- If you want to use SSL but do not want to use a signed certificate, you must perform the following actions:

1. Generate a self-signed certificate on the server.
2. Import the self-signed certificate to a trust store on the client.
3. Start jConsole, or another monitoring tool, with the trust store.

For more information, see [Using SSL with a Self-Signed Certificate](#).

2. Once you have provided an SSL certificate and private key, click **Save**.

## Step 5: JMX Resource Configuration

**Note:** Do not change the **OpenTSDB Firehose Nozzle** instance count unless you have a running Elastic Runtime installation.

To receive metrics data from the PCF Elastic Runtime firehose, including Diego metrics, change the **OpenTSDB Firehose Nozzle** instance count from **0** to **1**.

## Step 6: Elastic Runtime Resource Configuration

**Note:** Metrics coming from the Cloud Controller, the Gorouter, and etcd flow through the Collector and not the Firehose. In order to receive these metrics, you must enable the **Collector** in the Elastic Runtime Tile.

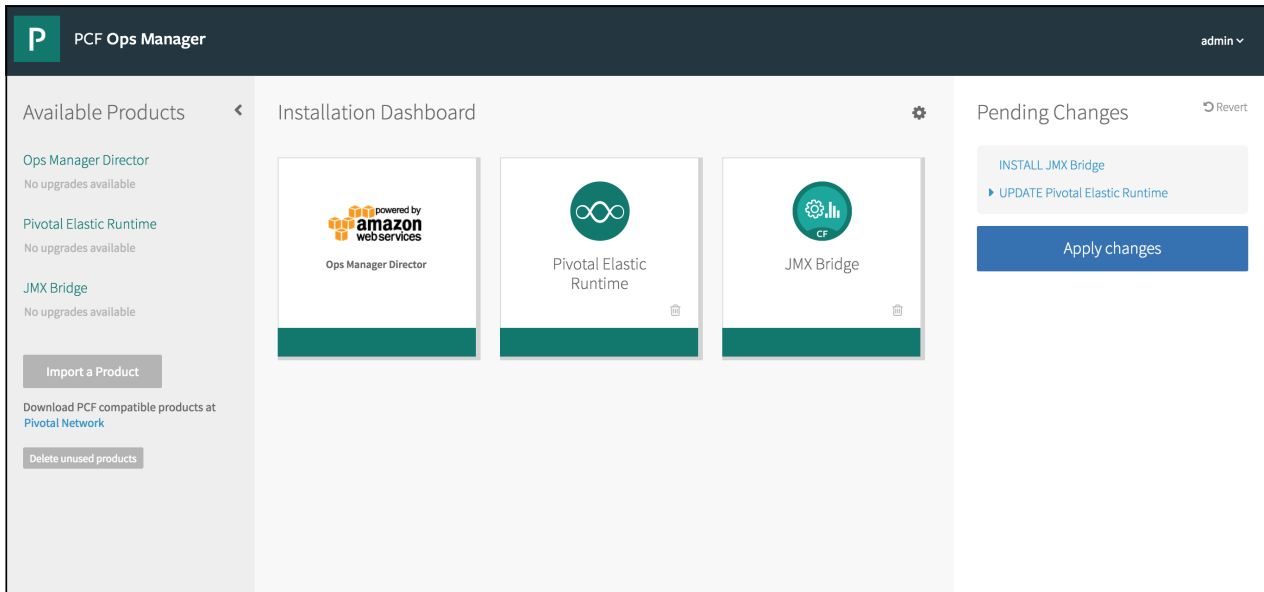
1. Navigate to the PCF Ops Manager Installation Dashboard.
2. Click on the **Elastic Runtime Tile** and select **Resource Config** under the **Settings** tab.
3. Change the **Collector** instance count from **0** to **1**.

Settings	Status	Credentials	Logs
Assign AZs and Networks	Resource Config		
Domains	JOB	INSTANCES	PERSISTENT DISK TYPE
Networking	Consul	Automatic: 1	Automatic: 1 GB
Application Containers	NATS	Automatic: 1	None
Application Developer Controls	etcd	Automatic: 1	Automatic: 1 GB
Application Security Groups	Diego BBS	Automatic: 1	Automatic: 1 GB
Authentication and Enterprise SSO	NFS Server	Automatic: 1	Automatic: 100 GB
Databases	Router	Automatic: 1	None
Internal MySQL	MySQL Proxy	Automatic: 1	None
File Storage	MySQL Server	Automatic: 1	Automatic: 100 GB
System Logging	Backup Prepare Node	Automatic: 0	None
Custom Branding	Cloud Controller Database (Postgres)	Automatic: 0	Automatic: 2 GB
Apps Manager	UAA Database (Postgres)	Automatic: 0	Automatic: 10 GB
Email Notifications	Apps Manager Database (Postgres)	Automatic: 0	Automatic: 1 GB
Restore CCDB Encryption Key	Cloud Controller	Automatic: 1	None
Smoke Tests	HAProxy	0	None
Experimental Features	Clock Global	Automatic: 1	None
Errands	Cloud Controller Worker	Automatic: 1	None
Resource Config	Collector	1	None
Stemcell	UAA	Automatic: 1	None
	Diego Brain	Automatic: 1	Automatic: 1 GB
	Diego Cell	Automatic: 3	None
	Doppler Server	Automatic: 1	None
	Loggregator Trafficcontroller	Automatic: 1	None
	Push Apps Manager	Automatic: 1	None
	Run Smoke Tests	Automatic: 1	None
	Notifications	Automatic: 1	None
	Run Notifications Tests	Automatic: 1	None
	Notifications-UI	Automatic: 1	None
	Run Notifications-UI tests	Automatic: 1	None
	Deploy CF Autoscaling App	Automatic: 1	None
	Register Autoscaling Service Broker	Automatic: 1	None
	Destroy autoscaling service broker	Automatic: 1	None

4. Click **Save**.

## Step 7: Apply Changes

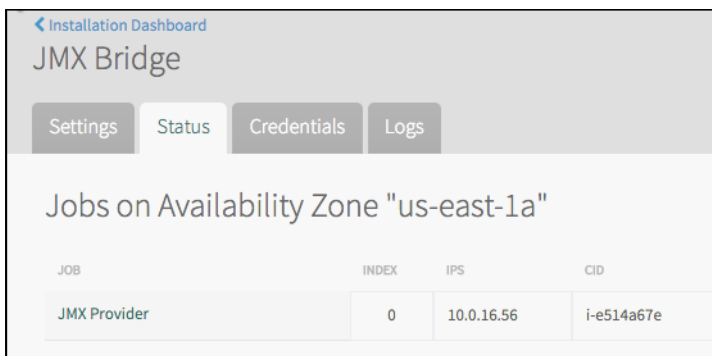
1. Navigate to the PCF Ops Manager Installation Dashboard.
2. In the Pending Changes view, click **Apply Changes** to install JMX Bridge.



- When complete, a “Changes Applied” message appears.

## Step 8: Find the IP of the JMX Provider

- Click **Return to Product Dashboard**.
- Click the **JMX Bridge** tile and select the **Status** tab.



- Record the IP address of the **JMX Provider**.

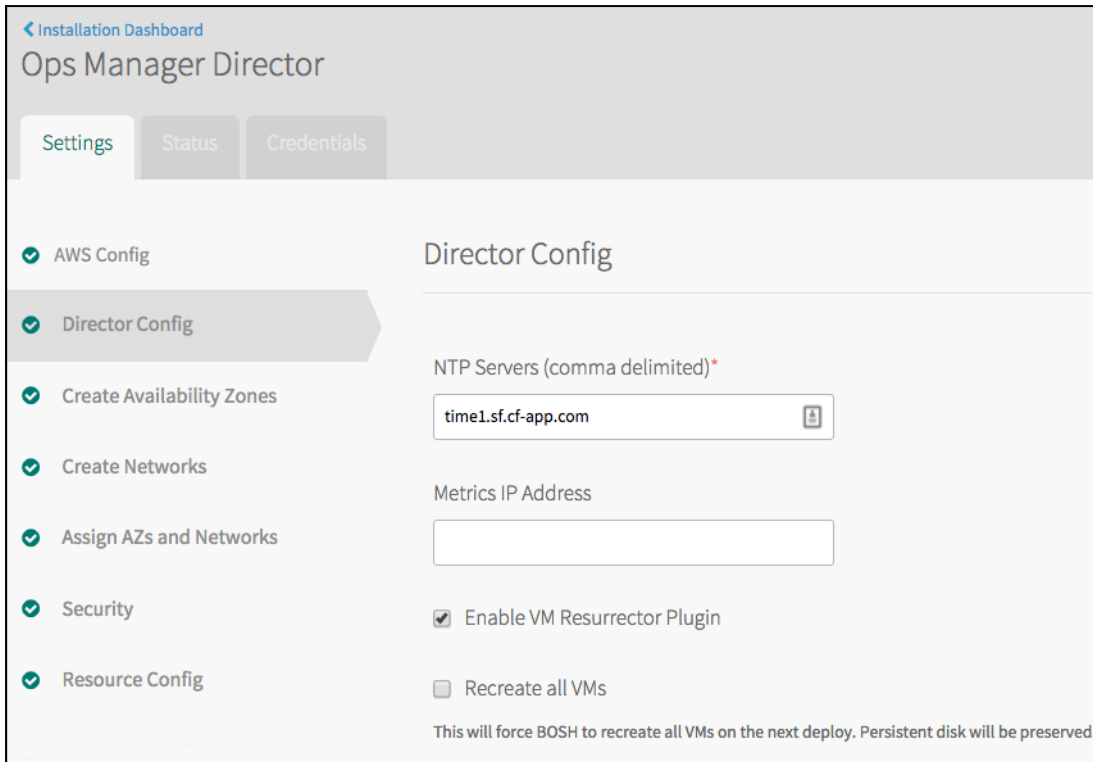


**Note:** After installation, your JMX client connects to this IP address at port 44444 using the credentials that you supplied. Also ensure that TCP port 44445 is open.

## Step 9: Configure the Metrics IP Address

- Return to the **Installation Dashboard**. Click the **Ops Manager Director** tile and select **Director Config**.





Installation Dashboard

## Ops Manager Director

Settings Status Credentials

✓ AWS Config

✓ **Director Config**

✓ Create Availability Zones

✓ Create Networks

✓ Assign AZs and Networks

✓ Security

✓ Resource Config

### Director Config

NTP Servers (comma delimited)\*

time1.sf.cf-app.com

Metrics IP Address

☒ Enable VM Resurrector Plugin

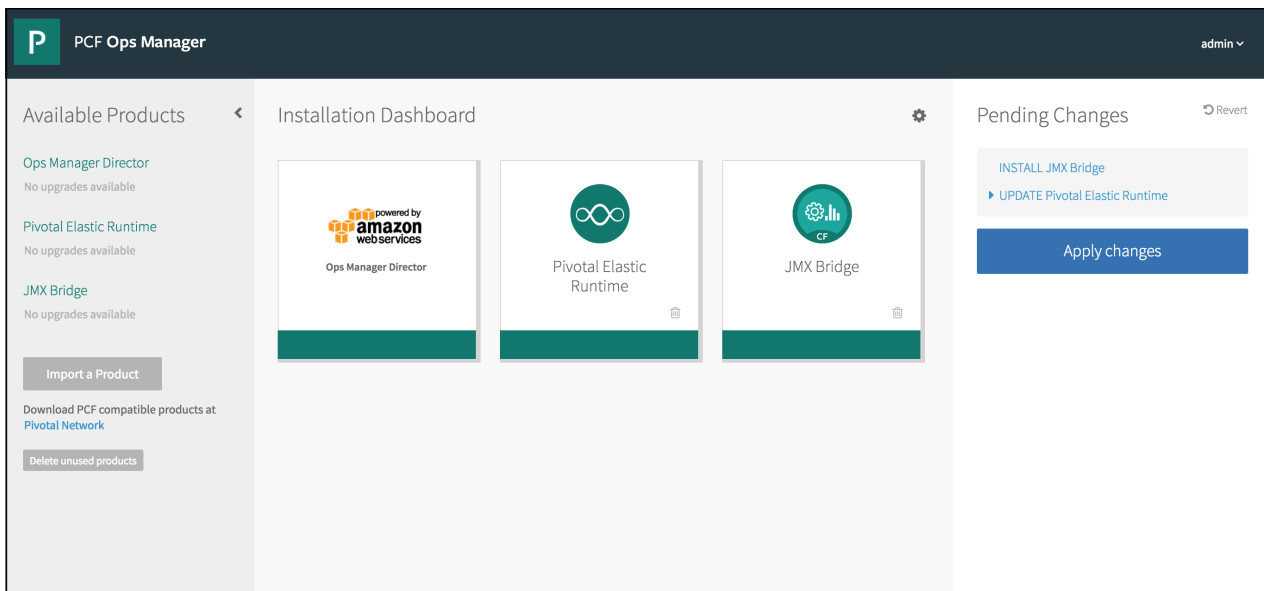
☐ Recreate all VMs

This will force BOSH to recreate all VMs on the next deploy. Persistent disk will be preserved.

2. In the **Metrics IP Address** field, enter the IP address of the JMX Provider. Click **Save**.

## Step 10: Complete Installation

1. In the Pending Changes view, click **Apply Changes**.



PCF Ops Manager admin

Available Products

Ops Manager Director  
No upgrades available

Pivotal Elastic Runtime  
No upgrades available

JMX Bridge  
No upgrades available

Import a Product

Download PCF compatible products at  
[Pivotal Network](#)

Delete unused products

### Installation Dashboard

Ops Manager Director

Pivotal Elastic Runtime

JMX Bridge

### Pending Changes

Revert

INSTALL JMX Bridge

UPDATE Pivotal Elastic Runtime

Apply changes

2. When complete, a “Changes Applied” message appears. Click **Return to Product Dashboard**. JMX Bridge is now installed and configured.

Once installed and configured, metrics for Cloud Foundry components automatically report to the JMX endpoint.

## Using JMX Bridge

Page last updated:

JMX Bridge is a Java Management Extensions (JMX) tool for Elastic Runtime. To help you monitor your installation and assist in troubleshooting, JMX Bridge collects and exposes system data from Cloud Foundry components via a JMX endpoint.

## Cloud Controller Metrics

JMX Bridge reports the number of Cloud Controller API requests completed and the requests sent but not completed.

The number of requests sent but not completed represents the pending activity in your system, and can be higher under load. This number will vary over time, and the range it can vary over depends on specifics of your environment such as hardware, OS, processor speeds, load, etc. In any given environment, though, you can establish a typical range of values and maximum for this number.

Use the Cloud Controller metrics to ensure that the Cloud Controller is processing API requests in a timely manner. If the pending activity in your system increases significantly past the typical maximum and stays at an elevated level, Cloud Controller requests may be failing and additional troubleshooting may be necessary.

The following table shows the name of the Cloud Controller metric, what the metric represents, and the metric type (data type).

METRIC NAME	DEFINITION	METRIC TYPE (DATA TYPE)
cc.requests.completed	Number of Cloud Controller API requests completed since this instance of Cloud Controller started	Counter (Integer)
cc.requests.outstanding	Number of Cloud Controller API requests made but not completed since this instance of Cloud Controller started	Counter (Integer)

See the [Cloud Controller](#) topic for more information about the Cloud Controller.

## Router Metrics

JMX Bridge reports the number of sent requests and the number of completed requests for each Cloud Foundry component.

The difference between these two metrics is the number of requests made to a component but not completed, and represents the pending activity for that component. The number for each component can vary over time, and is typically higher under load. In any given environment, though, you can establish a typical range of values and maximum for this number for each component.

Use these metrics to ensure that the Router is passing requests to other components in a timely manner. If the pending activity for a particular component increase significantly past the typical maximum and stays at an elevated level, additional troubleshooting of that component may be necessary. If the pending activity for most or all components increases significantly and stays at elevated values, troubleshooting of the router may be necessary.

The following table shows the name of the Router metric, what the metric represents, and the metric type (data type).

METRIC NAME	DEFINITION	METRIC TYPE (DATA TYPE)
router.requests [component=c]	Number of requests the router has received for component <code>c</code> since this instance of the router has started <code>c</code> can be CloudController or route-emitter	Counter (Integer)
router.responses [status=s,component=c]	Number of requests completed by component <code>c</code> since this instance of the router has started <code>c</code> can be CloudController or route-emitter <code>s</code> is http status family: 2xx, 3xx, 4xx, 5xx, and other	Counter (Integer)

See the [Router](#) topic for more information about the Router.

## Diego Metrics

Pivotal JMX Bridge reports metrics for the Diego cells and from the Diego Bulletin Board System (BBS). The following tables show the name of the Diego metric, what the metric represents, and the metric type (data type).

For general information about Diego, see the [Diego Architecture](#) topic.

## Diego Cell Metrics

Pivotal JMX Bridge reports the following metrics for each Diego cell. If you have multiple cells, JMX Bridge reports metrics for each cell individually. The metrics are not summed across cells.

Use these metrics to determine the size of your deployment or when to scale up a deployment, and to track the status of Long Running Processes (LRP) in the Diego life cycle.

METRIC NAME	DEFINITION	METRIC TYPE (DATA TYPE)
rep.CapacityTotalMemory	Total amount of memory available for this cell to allocate to containers.	Gauge (Float)
rep.CapacityRemainingMemory	Remaining amount of memory available for this cell to allocate to containers.	Gauge (Float)
rep.CapacityTotalDisk	Total amount of disk available for this cell to allocate to containers.	Gauge (Float)
rep.CapacityRemainingDisk	Remaining amount of disk available for this cell to allocate to containers.	Gauge (Float)
rep.ContainerCount	Number of containers hosted on the cell.	Gauge (Integer)

## Diego BBS Metrics

Pivotal JMX Bridge reports these metrics from the Diego BBS, and are deployment-wide metrics. Use these metrics to inspect the state of the apps running on the deployment as a whole.

METRIC NAME	DEFINITION	METRIC TYPE (DATA TYPE)
bbs.CrashedActualLRPs	Total number of LRP instances that have crashed.	Gauge (Integer)
bbs.LRPsRunning	Total number of LRP instances that are running on cells.	Gauge (Integer)
bbs.LRPsUnclaimed	Total number of LRP instances that have not yet been claimed by a cell.	Gauge (Integer)
bbs.LRPsClaimed	Total number of LRP instances that have been claimed by some cell.	Gauge (Integer)
bbs.LRPsDesired	Total number of LRP instances desired across all LRPs.	Gauge (Integer)
bbs.LRPsExtra	Total number of LRP instances that are no longer desired but still have a BBS record.	Gauge (Integer)
bbs.LRPsMissing	Total number of LRP instances that are desired but have no record in the BBS.	Gauge (Integer)

## Virtual Machine Metrics

JMX Bridge reports data for each virtual machine (VM) in a deployment. Use these metrics to monitor the health of your Virtual Machines.

The following table shows the name of the Virtual Machine metric, what the metric represents, and the metric type (data type).

METRIC NAME	DEFINITION	METRIC TYPE (DATA TYPE)
system.mem.percent	Percentage of memory used on the VM	Gauge (Float, 0-100)
system.swap.percent	Percentage of swap used on the VM	Gauge (Float, 0-100)
system.disk.ephemeral.percent	Percentage of ephemeral disk used on the VM	Gauge (Float, 0-100)
system.disk.system.percent	Percentage of system disk used on the VM	Gauge (Float, 0-100)
system.cpu.sys	Amount of CPU spent in system processes	Gauge (Float)
system.cpu.user	Amount of CPU spent in user processes	Gauge (Float)
system.cpu.wait	Amount of CPU spent in waiting processes	Gauge (Float)

## Using SSL with a Self-Signed Certificate in JMX Bridge

Page last updated:

Secure Socket Layer (SSL) is a standard protocol for establishing an encrypted link between a server and a client. To communicate over SSL, a client needs to trust the SSL certificate of the server.

This topic explains how to use SSL with a self-signed certificate in JMX Bridge (formerly Ops Metrics). This SSL layer secures traffic between JMX Bridge and the user, and is separate from the SSL layer [configured between Elastic Runtime](#) and the rest of the Ops Manager environment.

There are two kinds of SSL certificates: signed and self-signed.

- **Signed:** A Certificate Authority (CA) signs the certificate. A CA is a trusted third party that verifies your identity and certificate request, then sends you a digitally signed certificate for your secure server. Client computers automatically trust signed certificates. Signed certificates are also called *trusted certificates*.
- **Self-signed:** Your own server generates and signs the certificate. Clients do not automatically trust self-signed certificates. To communicate over SSL with a server providing a self-signed certificate, a client must be explicitly configured to trust the certificate.

**Note:** Certificates generated in Elastic Runtime are signed by the Operations Manager Certificate Authority. They are not technically self-signed, but they are referred to as 'Self-Signed Certificates' in the Ops Manager GUI and throughout this documentation.

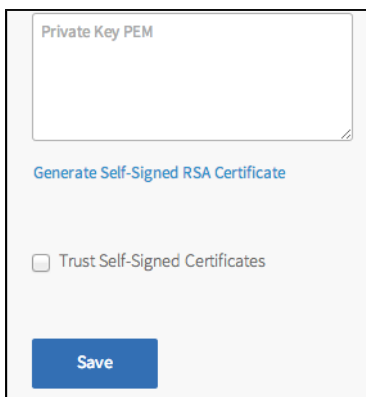
The following procedure configures a JMX user client to trust a self-signed certificate by importing the certificate to its truststore, an internal keystore. To use a trusted certificate signed by a CA, you only need to paste the Certificate and Key into the fields in the Ops Manager JMX Bridge tile, as shown in [Step 1, Option 2](#), below.

### Step 1: Supply SSL Certificate

#### Option 1: Generate Self-Signed Certificate

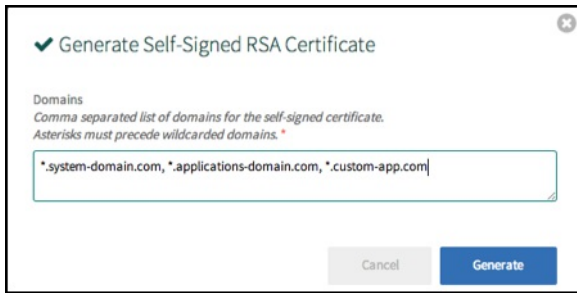
Follow the steps below to generate a self-signed certificate on your server:

1. In Pivotal Ops Manager, click the **JMX Bridge** tile.
2. Check **Enable SSL**.
3. Click **Generate Self-Signed RSA Certificate** and check the **Trust Self-Signed Certificates** box.

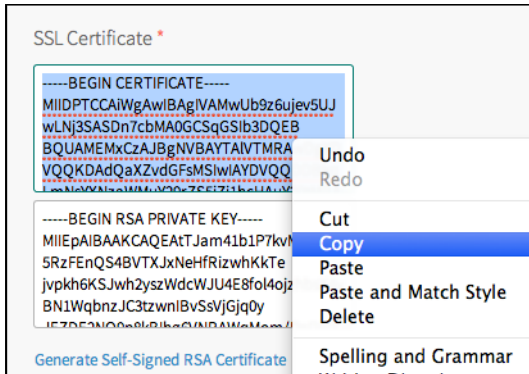


The screenshot shows a configuration form for the JMX Bridge tile. At the top, there is a text area labeled 'Private Key PEM'. Below it is a blue button labeled 'Generate Self-Signed RSA Certificate'. Underneath the button is a checkbox labeled 'Trust Self-Signed Certificates', which is currently unchecked. At the bottom of the form is a blue button labeled 'Save'.

4. Enter your system and application domains in wildcard format. Optionally, also add any custom domains in wildcard format. Click **Generate**.



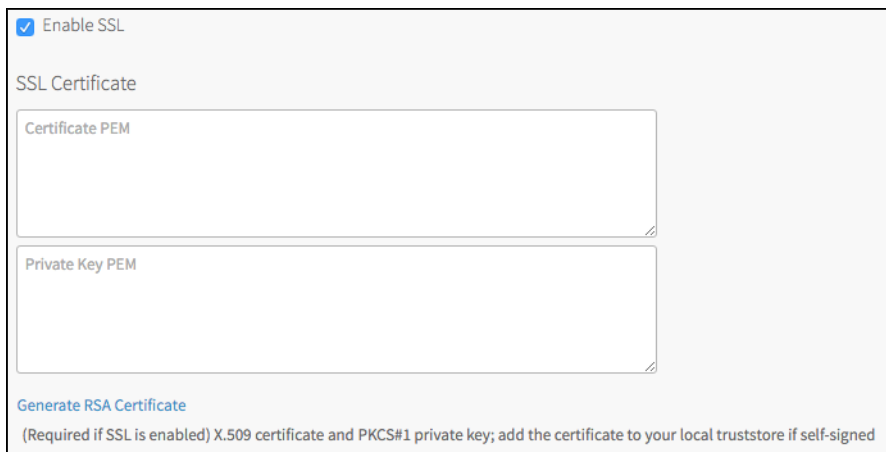
5. Select and copy the certificate.



6. Paste the certificate into a text file and save as a `.cer` file, such as `MY-JMX-BRIDGE.cer`.

## Option 2: Use an Existing Self-Signed Certificate

1. In Pivotal Ops Manager, click the **JMX Bridge** tile.
2. Check **Enable SSL**.
3. Paste your certificate and private key into the appropriate boxes. This is your X.509 certificate and PKCS#1 private key.



## Step 2: Import the Self-signed Certificate to a Truststore

Follow the steps below to import the self-signed certificate to your client:

1. Copy your certificate file `MY-JMX-BRIDGE.cer` from your server to your client.
2. Navigate to the client directory where you copied the saved certificate.
3. Use `keytool -import` to import the certificate with an alias of `ops-metrics-ssl` to the truststore `localhost.truststore`:

```
$ keytool -import -alias ops-metrics-ssl -file MY-JMX-BRIDGE.cer -keystore localhost.truststore
```

- If `localhost.truststore` already exists, a password prompt appears. Enter the keystore password that you recorded in a previous step.
- If `localhost.truststore` does not exist, you must create a password.

4. Verify the details of the imported certificate.

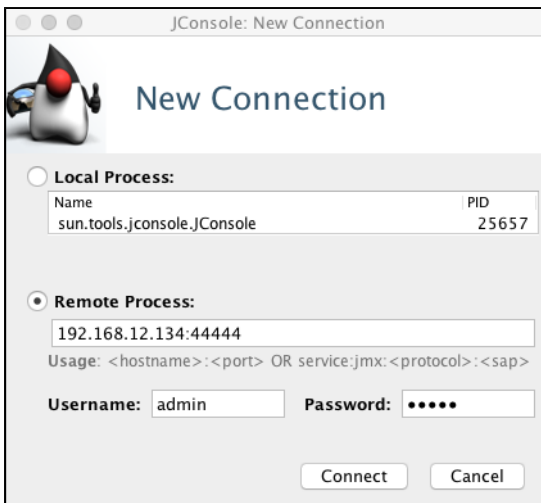
## Step 3: Start a Monitoring Tool with the Truststore

Once you import the self-signed certificate to `localhost.truststore` on the client, configure your monitoring tool, such as Jconsole, to use the truststore. You do this from a command line, by starting your monitoring tool with the location and password of the truststore.

1. Pass in the location of `localhost.truststore` to your monitoring tool with the `javax.net.ssl.trustStore` property, and its password with the `javax.net.ssl.trustStorePassword` property. For example, you would invoke jConsole with:

```
$ jconsole -J-Djavax.net.ssl.trustStore=/lib/home/jcert/localhost.truststore -J-Djavax.net.ssl.trustStorePassword=KEYSTORE_PASSWORD
```

2. In the **Remote Process** field, enter the fully qualified hostname of the Maximus server, port number `44444`.



3. To complete the **Username** and **Password** fields, refer to the **Credentials** tab of the JMX Bridge tile in Pivotal Ops Manager. By default, these credentials are `admin` and `admin`.


Your monitoring tool should now communicate with your server through the SSL connection.

## Troubleshooting and Uninstalling JMX Bridge

Page last updated:

The JMX Bridge tool (formerly Ops Metrics) is a JMX extension for Elastic Runtime. JMX Bridge collects and exposes system data from Cloud Foundry components via a JMX endpoint. Use this system data to monitor your installation and assist in troubleshooting.

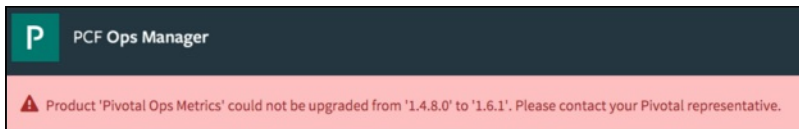
The JMX Bridge tool is composed of three virtual machines:

- The JMX Provider
- A VM that governs compilation
- A nozzle for the [Loggregator Firehose](#) 

To deploy JMX Bridge, see the [Deploying JMX Bridge](#) topic.

## Resolve Upgrade Error for 1.4 to 1.6

If you see the following error during an upgrade from Ops Metrics 1.4 to 1.6 (now JMX Bridge):




1. [Uninstall Ops Metrics 1.4](#)
2. [Install Ops Metrics 1.6](#) 

## Resolve Upgrade Error for 1.6 to 1.7

If you see the following error during an upgrade from Ops Metrics 1.6 to JMX Bridge 1.7.x:

**This product requires a stemcell that is older than the currently installed product. Download the newest version of this product from Pivotal Network and try again.**

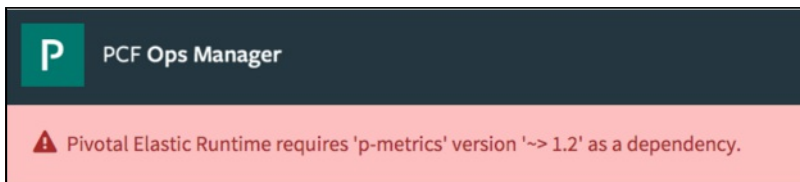
[Upgrade Ops Manager](#)  to 1.7.8 or higher. The error above results from a known issue and newer versions of Ops Manager resolve the issue.

## Uninstall JMX Bridge (formerly Ops Metrics)

1. Scale the collector resource from Elastic Runtime to .

Settings	Status	Credentials	Logs
Assign AZs and Networks	Resource Config		
Domains			
Networking	Consul	Automatic: 1	Automatic: 1 GB
	NATS	Automatic: 1	None
Application Containers	etcd	Automatic: 1	Automatic: 1 GB
Application Developer Controls	Diego BBS	Automatic: 1	Automatic: 1 GB
Application Security Groups	NFS Server	Automatic: 1	Automatic: 100 GB
Authentication and Enterprise SSO	Router	Automatic: 1	None
	MySQL Proxy	Automatic: 1	None
	MySQL Server	Automatic: 1	Automatic: 100 GB
Databases	Backup Prepare Node	Automatic: 0	None
	Cloud Controller Database (Postgres)	Automatic: 0	Automatic: 2 GB
Internal MySQL	UAA Database (Postgres)	Automatic: 0	Automatic: 10 GB
	Apps Manager Database (Postgres)	Automatic: 0	Automatic: 1 GB
File Storage	Cloud Controller	Automatic: 1	None
System Logging	HAProxy	0	None
	Clock Global	Automatic: 1	None
Custom Branding	Cloud Controller Worker	Automatic: 1	None
Apps Manager	Collector	1	None
	UAA	Automatic: 1	None
Email Notifications	Diego Brain	Automatic: 1	Automatic: 1 GB
	Diego Cell	Automatic: 3	None
Restore CCDB Encryption Key	Doppler Server	Automatic: 1	None
Smoke Tests	Loggregator Trafficcontroller	Automatic: 1	None
Experimental Features	Push Apps Manager	Automatic: 1	None
	Run Smoke Tests	Automatic: 1	None
Errands	Notifications	Automatic: 1	None
	Run Notifications Tests	Automatic: 1	None
	Notifications-UI	Automatic: 1	None
Resource Config	Run Notifications-UI tests	Automatic: 1	None
	Deploy CF Autoscaling App	Automatic: 1	None
Stemcell	Register Autoscaling Service Broker	Automatic: 1	None
	Destroy autoscaling service broker	Automatic: 1	None

If you do not scale the collector resource, you get the following error:



2. Proceed with uninstallation. See the [Deleting a Product](#) section of the [Adding and Deleting Products](#) topic for details.

## Missing Metrics from PCF Installation or Firehose

If you are not seeing expected metrics from Elastic Runtime in the JMX provider, verify that you installed Elastic Runtime prior to JMX Bridge. If JMX Bridge was installed first, perform the following steps:

1. SSH into the `opentsdb-firehose-nozzle` VM. For information about how to use the BOSH CLI to SSH into a VM, see [Advanced Troubleshooting with the BOSH CLI](#).
2. Grant `sudo` access to the machine:

```
$ sudo -i
```

3. Restart the `opentsdb-firehose-nozzle` job.



```
$ monit restart opentsdb-firehose-nozzle
```



## Application Security Groups

PCF applications do not interact directly with the PCF JMX Bridge tile. Therefore, you do not need to create Application Security Groups (ASGs) to interact with the bridge from an external application.

## JMX Bridge Release Notes and Known Issues

 **Note:** Before version 1.7.X, JMX Bridge was known as Ops Metrics. For Ops Metrics release notes and known issues, see [Version 1.6.X](#) .

### Release Notes

#### Version 1.7.12

**Release Date:** April 28, 2017

- Updates the stemcell required to 3363

#### Version 1.7.10

**Release Date:** March 09, 2017

- Maintenance update of the following product dependencies:
  - Updates the OpenJDK version used in the product to v1.8.0.121
  - Updates the go-lang version used in the product to v1.8

#### Version 1.7.8

**Release Date:** December 15, 2016

- Fixes a potential issue in v1.7.7 where the tile may fail to perform as expected if the optional Enable SSL feature was active. Although no customer issues were reported, v1.7.8 is a corrective patch to prevent any possible performance risk when using the Enable SSL feature

#### Version 1.7.7

**Release Date:** December 12, 2016

- Maintenance update of the following product dependencies:
  - SLF4J now v1.7.21
  - Logback now v1.1.7
  - Guava now v20.0
  - Bouncy Castle bcprov-jdk15on now v1.55

#### Version 1.7.6

**Release Date:** November 10, 2016

- Updates the stemcell required to 3263

#### Version 1.7.5

**Release Date:** November 09, 2016

- Updates the OpenJDK version used in the product to v1.8.0.111

## Version 1.7.4

**Release Date:** October 05, 2016

- Updates the stemcell required to 3233

## Version 1.7.3

**Release Date:** September 14, 2016

- Updates the NOAA client and other service related dependencies.
  - An increase in disconnects between the OpenTSDB Firehose Nozzle and the Firehose, resulting in a need to restart the nozzle, was diagnosed as the NOAA client was becoming too far out-of-date. Version 1.7.3 updates the NOAA client and other service related dependencies.
- Stemcell remains the same at 3232

## Version 1.7.2

**Release Date:** May 20, 2016

- Fixes issue where you could not upgrade from Ops Metrics 1.6.10+ if you had already upgraded to OpsMan 1.7
- Stemcell remains the same at 3232

## Version 1.7.1

**Release Date:** May 06, 2016

- Removes the bosh job ID from metric name
- Updated stemcell for 1.7.1 to 3232

## Version 1.7.0

**Release Date:** May 06, 2016

- Product name has been changed from “Ops Metrics” to “JMX Bridge”
- Stemcell for 1.7.0 is 3149

## Known issues for 1.7.X

- There is nothing that prevents the user from turning on the Nozzle deployment when Elastic Runtime is not present.
  - Enabling the nozzle when Elastic Runtime is not deployed or enabled will produce the following error:
  - “RuntimeError - unknown product ‘cf’ in ((..cf.cloud\_controller.system\_domain.value))”
  - To fix this, reduce the number of nozzle counts to zero until Elastic Runtime is enabled.
- Because deploying the OpenTSDB firehose nozzle is currently optional and disabled by default, smoke tests for Elastic Runtime have been disabled until a later release.

## Past Minor Version 1.6.X

Release Notes for 1.6.X releases can be found [here](#) 

