

# PCF JMX Bridge®

Version 1.8


## 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	14
Using SSL with a Self-Signed Certificate in JMX Bridge	17
JMX Bridge Resources	20
Troubleshooting and Uninstalling JMX Bridge	21
Application Security Groups	25
Release Notes and Known Issues	26

## Pivotal Cloud Foundry JMX Bridge

 **WARNING:** As of Pivotal Cloud Foundry (PCF) v2.1, JMX Bridge v1.8 is deprecated.

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 two virtual machines:

- The JMX provider
- A Nozzle for the [Loggregator Firehose](#) 

## Product Snapshot

The following table provides version and version-support information about PCF JMX Bridge:

Element	Details
Version	v1.8.23
Release date	August 10, 2017
Compatible Ops Manager version(s)	v1.8.x, v1.9.x, v1.10.x
Compatible Elastic Runtime version(s)	v1.8.1 or later, v1.9.x, v1.10.x
IaaS support	AWS, Azure, GCP, OpenStack, and vSphere
IPsec support	Yes

## JMX Bridge User Guide

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

## Deploying JMX Bridge

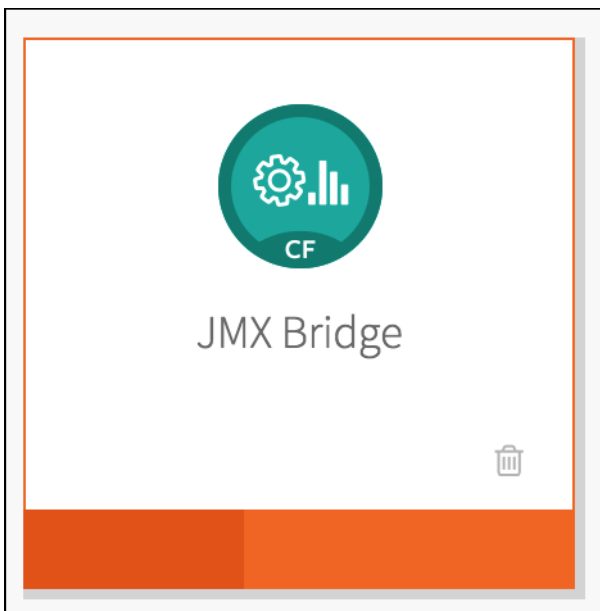
Page last updated:

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. JMX Bridge requires the components to be installed in this order.

1. [Download JMX Bridge](#).
2. Import JMX Bridge into Ops Manager by following the instructions for [Adding and Importing Products](#).
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 the BOSH Director.
2. (**vSphere and Amazon Web Services Only**) Select an Availability Zone under **Place singleton jobs in**. This Availability Zone is where Ops Manager runs Metrics jobs that only have a single instance.
3. (**vSphere and Amazon Web Services Only**) Select one or more Availability Zones under **Balance other jobs in**. Ops Manager balances 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

JMX Provider

Resource Config

Stemcell

### AZ and Network Assignments

Place singleton jobs in

☒ us-west-1a

☐ us-west-1c

Balance other jobs in

☒ us-west-1a

☒ us-west-1c

Network

vpc-volt

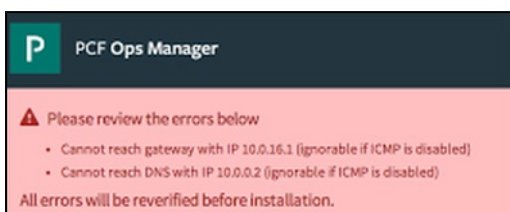
Save

4. Select a Network from the drop down menu.

**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. You can ignore this error. It only appears because the PCF security group blocks ICMP.



## 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.

- Record these credentials. You use these to connect JMX clients to the JMX Provider.

## (Optional) Step 4: Enable NAT Support

- Select the **Enable** radio button. NAT support is disabled by default. Enabling NAT support allows you to set the NAT IP as the host IP. By default, the internal IP address of the JMX Provider VM is set as the host IP.
- Enter the NAT IP as the External IP address in the form `0.0.0.0`

The screenshot shows the 'Credentials' tab of the JMX Provider configuration. The left sidebar lists several sections: 'Assign AZs and Networks', 'JMX Provider' (which is highlighted), 'Errands', 'Resource Config', and 'Stemcell'. The main content area is titled 'Credentials to connect to JMX Provider'. It contains two main sections: 'JMX Provider credentials' and 'NAT Support'. In the 'JMX Provider credentials' section, there are two input fields: the first contains 'admin' and the second is masked with asterisks. A 'Change' link is positioned below the password field. In the 'NAT Support' section, the 'Enable' radio button is selected. Below this, there is a text field labeled 'The External IP address for the JMX Provider' which contains the value '1.2.3.4'. Further down, there are two unchecked checkboxes: 'Enable Security Logging' and 'Enable SSL'. Below these is an 'SSL Certificate' section with two large text areas for 'Certificate PEM' and 'Private Key PEM'. A 'Generate RSA Certificate' link is located below the 'Private Key PEM' field. At the bottom of the configuration area is a blue 'Save' button.

- If you have enabled or disabled `NAT Support`, click **Save**.

**Note:** To connect to the JMX Provider after install, you **must** use the specified IP address. The IP address displayed in the `Status` tab always reflects the internal IP address of the JMX Provider VM, not the external IP address.

## (Optional) Step 5: Enable Security Logging

1. Select the **Enable Security Logging** checkbox. Access to the JMX endpoint is logged to STDOUT by default. You can enable security logging in the JMX Bridge tile configuration by selecting this checkbox, or disable it by deselecting this checkbox. Security logging is enabled by default.

Settings Status Credentials Logs

Assign AZs and Networks

JMX Provider

Errands

Resource Config

Stemcell

### Credentials to connect to JMX Provider

JMX Provider credentials \*

admin

\*\*\*\*\*

[Change](#)

NAT Support\*

☐ Enable

☒ Disable

☒ Enable Security Logging

☐ Enable SSL

Require clients to connect via SSL

SSL Certificate

Certificate PEM

Private Key PEM

[Generate RSA Certificate](#)

[Save](#)

2. If you made changes to Security Logging, click **Save**.

**Note:** Related log output is made available by initiating a JMX Provider logs download from the JMX Bridge tile configuration status tab, then fetching the download from the logs tab.

## (Optional) Step 6: Configure SSL

1. Select the **Enable SSL** checkbox. If you enable SSL, JMX clients are forced to use SSL to connect to the JMX Provider.

✓ Assign AZs and Networks

○ JMX Provider

✓ Errands

✓ Resource Config

✓ Stemcell

## Credentials to connect to JMX Provider

JMX Provider credentials \*

NAT Support\*

☐ Enable  
☒ Disable

☐ Enable Security Logging  
☒ Enable SSL

Require clients to connect via SSL

SSL Certificate

Certificate PEM

Private Key PEM

[Generate RSA Certificate](#)

**Save**

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](#).



Assign AZs and Networks

JMX Provider

Errands

Resource Config

Stemcell

## Credentials to connect to JMX Provider

JMX Provider credentials \*

[Change](#)

NAT Support\*

☐ Enable
 ☒ Disable

☐ Enable Security Logging

☒ Enable SSL
 Require clients to connect via SSL

SSL Certificate

```
-----BEGIN CERTIFICATE-----
MIIDRjCCAi6gAwIBAgIUUmIn4Z0hgstuT0EFZ75q7GiZingwDQYJKoZIhvcNAQEF
BQAwHzELMAkGA1UEBhMCVVMxEDAOBgNVBAoMB1Bpdm90YWwwHhcNMTYxMjExMTc
y
MTMxWhcNMTgxMjExMTcyMTMxWjAzMQswCQYDVQQGEwJVUzEQMA4GA1UECgwHUGI2
b3RkLmNldC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51ZC51
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAstNqGj0atb2KPPDwEwBlOMje43fZ3NpQl1oHH2ACn8T7Fyt
HPYWpvNpZJKTKj3FOyhDCl5yGprOQ86Ax+ptvas1nsY+4rPYfcHQBjpdgKMNgOKI
4lpK+K/Dds+7Nxywjc9oe3kpcLgwTbt9JgYDFNhyarBc7QyNU9SnBGMJjHa7j+gr
B2wnYnHEbVffU18S5e3IX8ZM39O6X6MNNPAI5gM3NMy1rdFtONkqguDjnu+te2AG
iXAFSLeflshkLpLjMgou90KpCBpFzJ6CYD70pD5e7e/Lp5C13M17C10HhVocF3e
-----
```

[Generate RSA Certificate](#)

Save

2. Click **Save**.

## (Optional) Step 7: Configure Errands

Errands are scripts that Ops Manager runs to automate tasks. By default, Ops Manager runs the post-install errands listed below when you deploy PCF JMX Bridge. However, you can prevent a specific post-install errand from running by deselecting its checkbox on the Errands page.

## Errands

Errands are scripts that run at designated points during an installation.

### Post-Deploy Errands

Smoke tests for JMX Bridge

Errand that run the smoke tests for JMX Bridge.

Default (When Changed)



There are no pre-delete errands for this product.

Save

- Select **Smoke tests for JMX Bridge** to cause the JMX Bridge to verify the following:
  - If the Firehose Nozzle is enabled, JMX Bridge verifies that the Nozzle is receiving metrics and that the product is not a slow consumer
  - If [BOSH Metrics](#) are enabled, JMX Bridge verifies that the product is receiving appropriate health metrics

**Note:** If errors occur during the install due to smoke tests, refer to the [troubleshooting documentation](#) for more information.

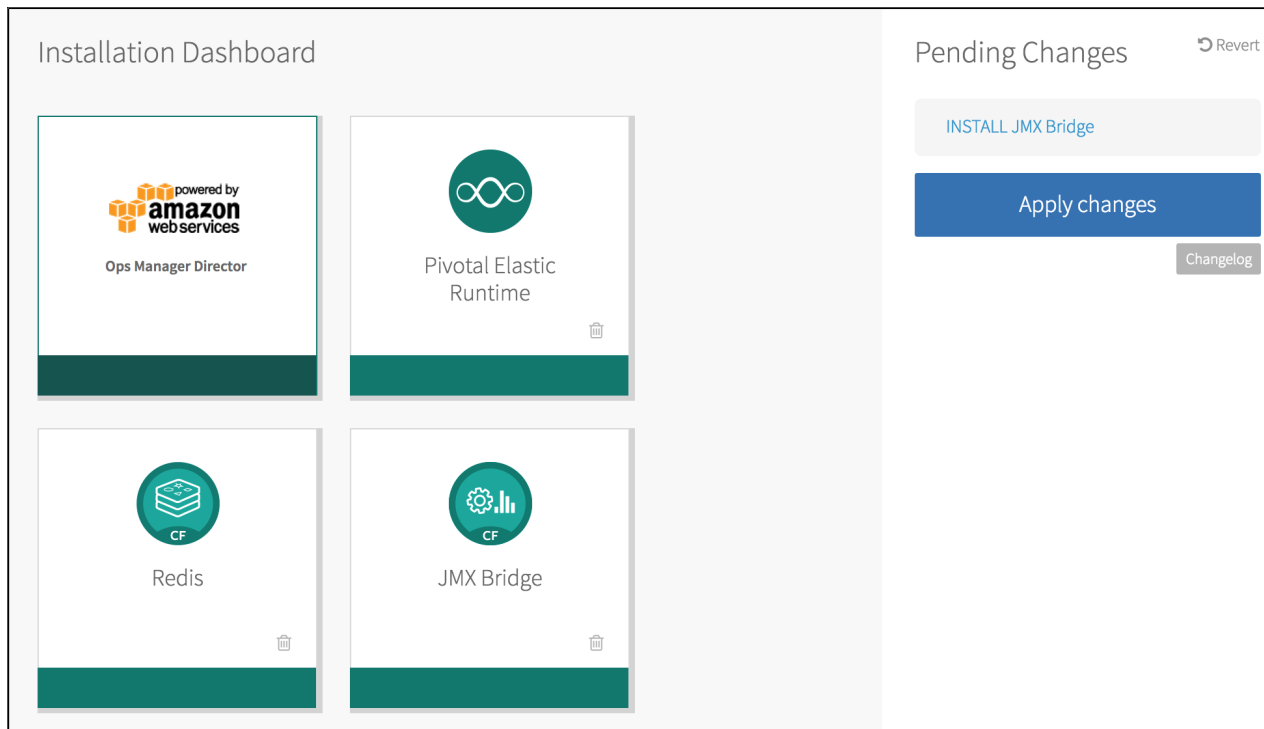
## (Optional) Step 8: Resource Configuration

**Note:** The Firehose Nozzle is enabled by default and requires [Elastic Runtime](#) .

To disable the Firehose Nozzle or stop receiving [Elastic Runtime](#) (including Diego) metrics, modify the instance count of the **OpenTSDB Firehose Nozzle** from **1** to **0**.

## Step 9: Apply Changes

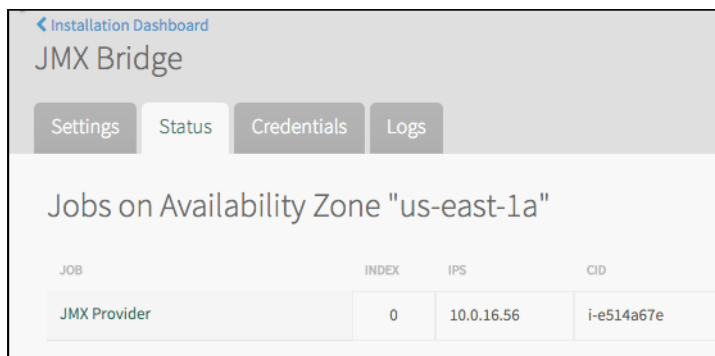
1. Navigate to the PCF Ops Manager Installation Dashboard.
2. In the Pending Changes view, click **Apply Changes** to install JMX Bridge.



After installation completes, a message appears stating that the changes have been applied.

## Step 10: Find the IP Address of the JMX Provider

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

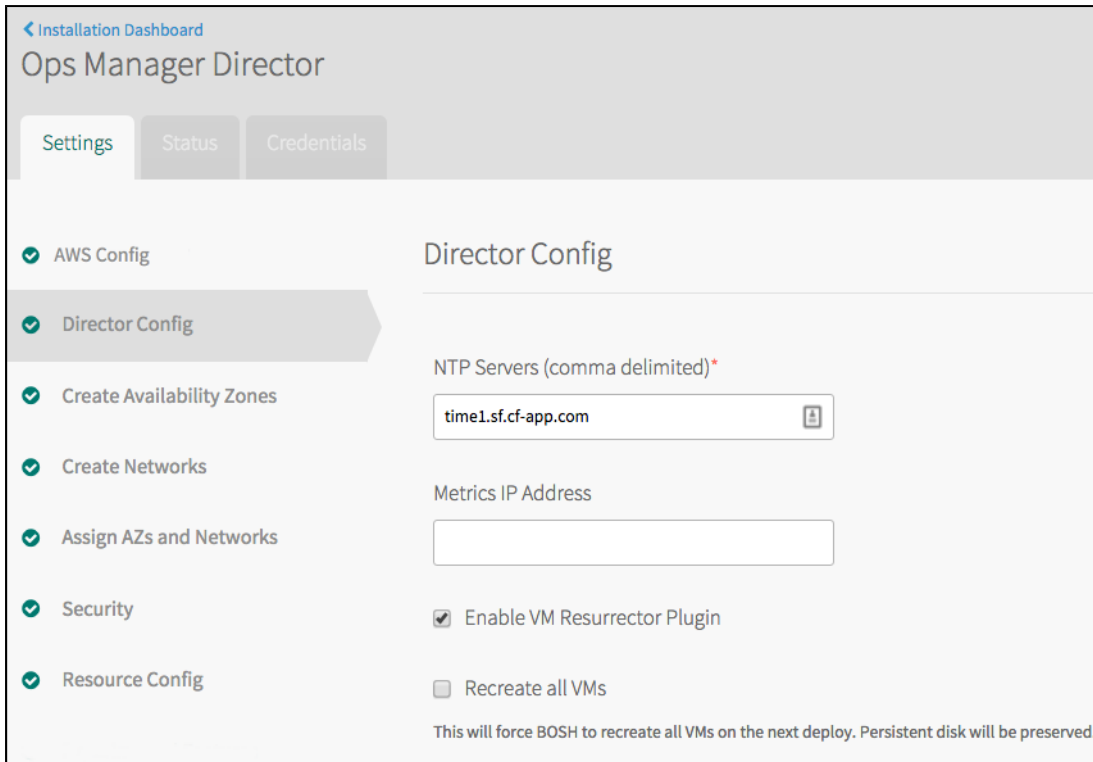


3. Record the IP address of the **JMX Provider**.

**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 11: Configure the Metrics IP Address

1. Return to the **Installation Dashboard**. Click the **BOSH 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

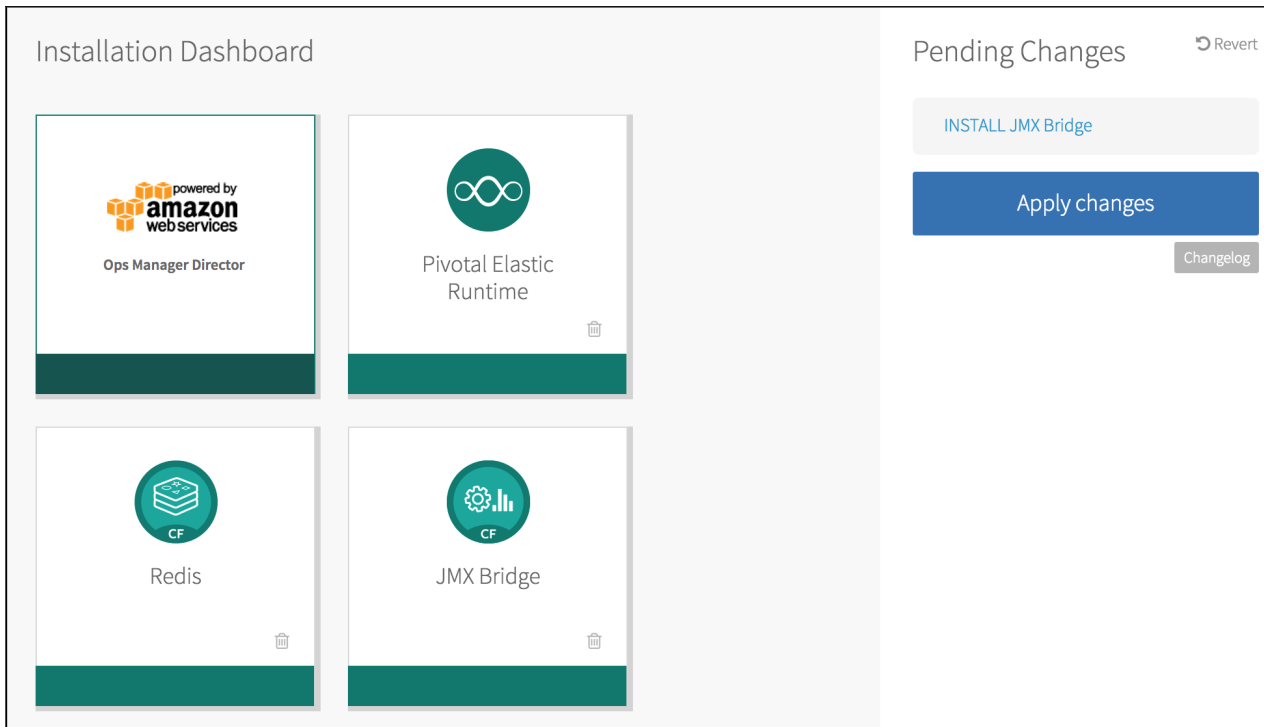
☐ Recreate all VMs

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 12: Complete Installation

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



Installation Dashboard

powered by amazon web services

Ops Manager Director

Pivotal Elastic Runtime

Redis

JMX Bridge

Pending Changes

Revert

INSTALL JMX Bridge

Apply changes

Changelog

When complete, a message appears stating that the changes have been applied.

2. Click **Return to Product Dashboard**. JMX Bridge is now installed and configured.



After installation and configuration, 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.

 **Note:** If using JMX Bridge v1.8 with PCF v1.10, please see the following recommended [Key Performance Indicators](#) .

## 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)
gorouter.requests [component=c]	Number of requests the router has received for component <b>c</b> since this instance of the router has started <b>c</b> can be CloudController or route-emitter	Counter (Integer)
gorouter.responses [status=s,component=c]	Number of requests completed by component <b>c</b> since this instance of the router has started <b>c</b> can be CloudController or route-emitter <b>s</b> 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.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)
system.disk.ephemeral.percent	Percentage of ephemeral disk used on the VM	Gauge (Float, 0-100)
system.disk.ephemeral.inode.percent	Percentage of inodes consumed by the ephemeral disk	Gauge (Float, 0-100)
system.disk.persistent.percent	Percentage of persistent disk used on the VM	Gauge (Float, 0-100)
system.disk.persistent.inode.percent	The percentage of inodes consumed by the persistent disk	Gauge (Float, 0-100)
system.disk.system.percent	Percentage of system disk used on the VM	Gauge (Float, 0-100)

system.healthy	Indicates whether a VM system is healthy. `1` means the system is healthy, and `0` means the system is not healthy	Gauge (Float, 0-1)
system.load.1m	Amount of load the system is under, averaged over one minute	Gauge (Float)
system.mem.percent	Percentage of memory used on the VM	Gauge (Float)
system.swap.kb	Amount of swap used on the VM in KB	Gauge (Float)
system.swap.percent	Percentage of swap used on the VM	Gauge (Float, 0-100)



## 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.

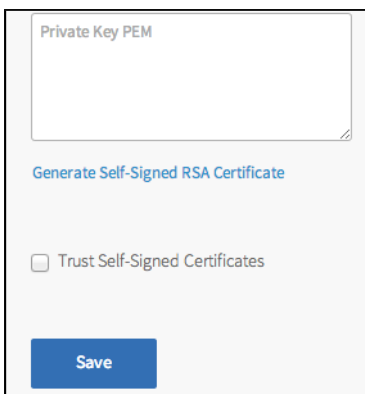
The following procedure configures a JMX user client to trust a self-signed certificate by importing the certificate to its truststore, an internal keystore. To use a trusted certificate signed by a CA, you only need to paste the Certificate and Key into the fields in the Ops Manager JMX Bridge tile,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

### Step 1: Supply SSL Certificate

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

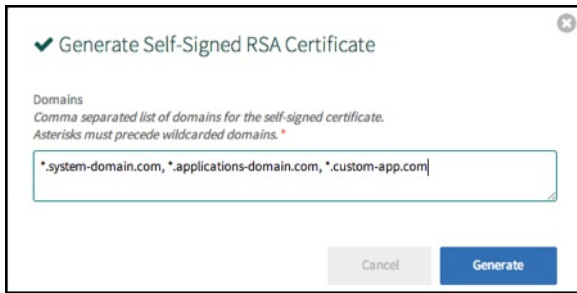
Follow the steps below to generate a self-signed certificate on your server:

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.

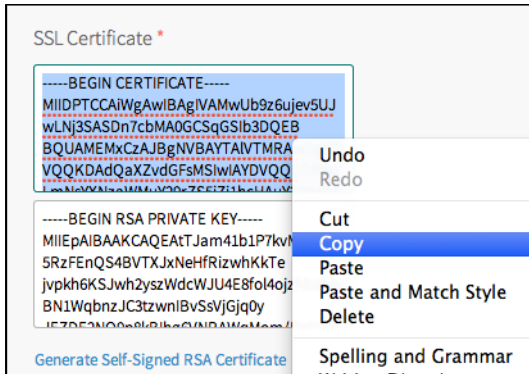


The screenshot shows a configuration form for the JMX Bridge tile. At the top, there is a text area labeled 'Private Key PEM'. Below it is a blue button labeled 'Generate Self-Signed RSA Certificate'. Underneath the button is a checkbox labeled 'Trust Self-Signed Certificates', which is currently unchecked.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



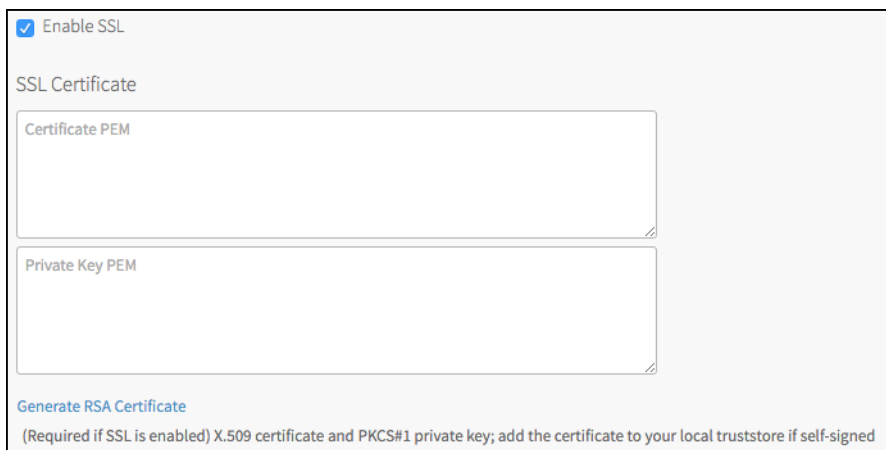
5. Select and copy the certificate.



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.

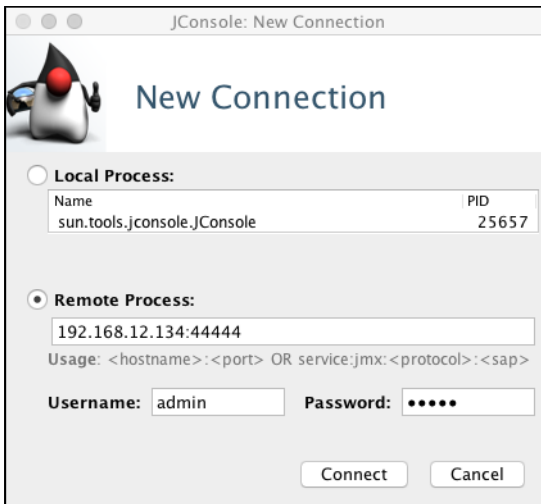
## Step 3: Start a Monitoring Tool with the Truststore

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.

## JMX Bridge Resources

### Resource Requirements

The following table shows the default resource and IP requirements for installing the tile:

Product	Resource	Instances	CPU	RAM	Ephemeral	Persistent	Static IP	Dynamic IP
JMX Bridge	JMX Provider	1	2	4GB	8GB	1GB	1	0
JMX Bridge	OpenTSDB Firehose Nozzle	1	2	4GB	8GB	1GB	1	0
JMX Bridge	Smoke Tests	1	4	4GB	8GB	0	1	0

### Guidelines

- If you anticipate a large volume of metrics coming from the Firehose, then scale up the number of OpenTSDB Firehose Nozzle instances accordingly.

## Troubleshooting and Uninstalling JMX Bridge

Page last updated:

This topic describes how to resolve common issues with the JMX Bridge for Pivotal Cloud Foundry (PCF) tile and how to uninstall the tile if necessary.


### Troubleshoot

The following sections provide help with troubleshooting JMX Bridge for PCF.

#### Dependency Error During Install

You might see the following error if your installed version of Elastic Runtime is not compatible with the version of JMX Bridge that you are attempting to install:


JMX Bridge requires 'cf' version '~> 1.8.0.0' as a dependency.

 Please review the errors below

- JMX Bridge requires 'cf' version '~> 1.8.0.0' as a dependency.

Follow the instructions below that correspond to your installed Elastic Runtime version:

- Elastic Runtime v1.8.0 to v1.8.16:**
  - Download version v1.8.17 or later of the [Elastic Runtime tile](#) and import it into your Ops Manager **Installation Dashboard**.
  - Click the newly added tile to review any configurable options. For more information, see [Installing Pivotal Cloud Foundry](#) and choose your IaaS.
  - Click **Apply Changes** to install both Elastic Runtime and JMX Bridge.
- Elastic Runtime v1.8.17 or later:** Upgrade the JMX Bridge tile to v1.8.9 or later.

 **Note:** A known issue exists with older versions of JMX Bridge and the newer versions of Elastic Runtime. JMX Bridge v1.8.2 through v1.8.8 only work with Elastic Runtime v1.8.0 through v1.8.16.

#### Cyclic Dependency Error During Install

You might see the following error if you have Elastic Runtime and JMX Bridge installed.

 Please review the errors below

- Found 1 set(s) of cyclic product dependencies: [Pivotal Elastic Runtime and JMX Bridge]

- JMX Bridge v1.8.7 or later; Elastic Runtime v1.8**
  - The Collector component is no longer compatible with JMX Bridge. All metrics that were previously flowing through the Collector are now flowing through the Firehose in Elastic Runtime v1.8.
  - Scale down the Collector in `Elastic Runtime > Resource Config` to `0`.
  - Click **Apply Changes** and continue your JMX Bridge installation.

 **Note:** This is only an issue for **Elastic Runtime v1.8**. The Collector does not exist in Elastic Runtime v1.9.

#### Missing Metrics from PCF Installation or Firehose

If you do not see expected metrics from Elastic Runtime in the JMX provider, verify that you installed Elastic Runtime before JMX Bridge. If you installed JMX Bridge first, perform the following steps:

- 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](#) .

- Grant **sudo** access to the machine:


```
$ sudo -i
```

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

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

## Missing BOSH Metrics

If you do not see expected metrics from BOSH, try the following steps:

- Make sure the IP address in **JMX Bridge > Status > JMX Provider** matches the value entered in **BOSH Director > Director Config > Metrics IP Address**.
- If the JMX Provider IP address matches the Metrics IP address and you see no BOSH metrics in the system, contact [Pivotal Support](#)  for help.

## Validating JMX Bridge MBeans

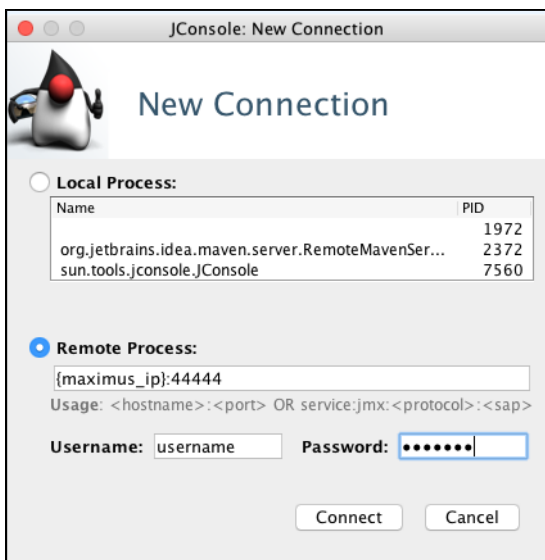
If you do not see metrics from JMX Bridge in your third-party tooling integration as expected, first try the following steps to quickly debug whether there is an issue with the JMX Bridge product or if the issue is with the tooling integration:

- Verify [Java 6+](#)  is installed.

- Run `jconsole`:

```
$ jconsole
```

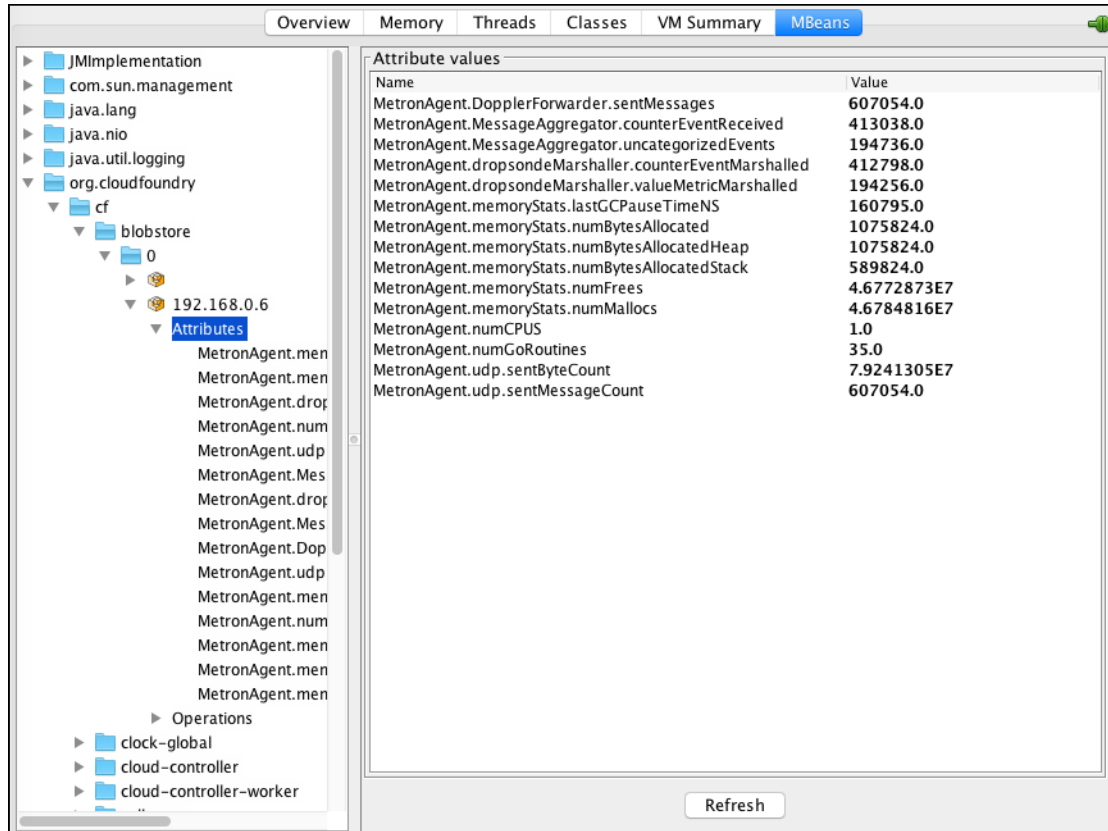
- Select **Remote Process** and enter the IP of the **JMX Provider** VM with port `44444`.
- Fill in the username and password for the **JMX Provider** that was entered during installation of JMX Bridge.
- Click **Connect**.



- Allow **Insecure connection** if SSL was not enabled.



You can now view all MBeans emitted by JMX Bridge.



**Note:** If you have enabled SSL, see [Using SSL with a Self-Signed Certificate in JMX Bridge](#).

## Set Up Port Forwarding for JMX

If you are connecting to jconsole from a location different from the install location (for example, deployed on AWS or GCP), you have to set up port forwarding to access the MBeans.

1. Set up port forwarding on one tab of your console and keep it open:

```
ssh -D 7777 username@pcf.domain.com -T
```

2. Start `jconsole` in a new tab and set up the `socksProxyPort` to the forwarded port:

```
jconsole -J-DsocksProxyHost=localhost -J-DsocksProxyPort=7777
```

3. Navigate `jconsole` as normal.

## Smoke Tests

If errors occur when the smoke tests run, you can find the errors in the **ChangeLog** for the installation. Some common failures are listed below.

<b>Error</b>	<code>internalMetricsAreSent() Fails</code>
<b>Cause</b>	The JMX Provider did not receive internal health metrics from the OpenTSDB Firehose Nozzle.
<b>Solution</b>	Restart the OpenTSDB Firehose Nozzle VM and check the logs to verify it is running correctly.

<b>Error</b>	<code>receivingFirehoseMetrics() Fails</code>
<b>Cause</b>	The OpenTSDB Firehose Nozzle is not receiving metrics from the Firehose.
<b>Solution</b>	Restart the OpenTSDB Firehose Nozzle VM and check the logs to verify it is connected to the Firehose. If you see a lot of reconnect attempts in the logs then you likely need to scale up the number of OpenTSDB Firehose Nozzle instances in the <b>Resource Config</b> tab.

## Uninstall


To uninstall the JMX Bridge for PCF tile, see [Deleting a Product](#) .



## 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.

## Release Notes and Known Issues

 **Note:** Before v1.7.X, JMX Bridge was known as Ops Metrics. For Ops Metrics release notes and known issues, see [Version 1.6.X](#) .

### Version 1.8.24

**Release Date: October 13, 2017**

#### Release Notes

- Maintenance update of the following product dependencies:
  - OpenJDK now v1.8.0.144
  - Golang now v1.9.1
- The stemcell for v1.8.24 remains v3363

#### Known issues

- None

### Version 1.8.23

**Release Date: August 10, 2017**

#### Release Notes

- Maintenance update of the following product dependencies:
  - OpenJDK now v1.8.0.141
  - Golang now v1.8.3
  - Guava now v23.0
  - Bouncy Castle now v1.57
- The stemcell for v1.8.23 remains v3363

#### Known issues

- None

### Version 1.8.22

**Release Date: May 25, 2017**

#### Release Notes

- Improvements to OpenTSDB Firehose Nozzle behavior during connectivity issues
  - When the OpenTSDB Firehose nozzle had a connectivity issue to the maximus vm, the metrics queue could fill to capacity, leaving the nozzle unable to sufficiently recover once the endpoint connection was restored
  - Because the product emphasizes transmission of only the latest metric values, in the case of a backup due to an inability to write to maximus, the firehose nozzle will now drop backed-up messages and will write out an error with the number of messages being dropped. This will allow the

nozzle to better recover upon recovery of the endpoint connection

- Example log message output `2017/05/22 18:40:29 Could not write to maximus VM. Dropping 2942 messages.`
- The stemcell for v1.8.22 remains v3363

## Known issues

- None

## Version 1.8.21

**Release Date:** May 12, 2017

## Release Notes

- Maintenance update of the following product dependencies:
  - OpenJDK now v1.8.0.131
  - Golang now v1.8.1
  - SLF4J now v1.7.25
  - Logback now v1.1.11
  - GRPC-ALL now v1.3.0
  - Guava now v21.0
  - Bouncy Castle bcprov-jdk15on now v1.56
- The stemcell for v1.8.21 remains v3363

## Known issues

- None

## Version 1.8.20

**Release Date:** May 01, 2017

## Release Notes

- Updates the stemcell required to v3363

## Known issues

- None

## Version 1.8.18

**Release Date:** March 09, 2017

## Release Notes

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

- Updates the golang version used in the product to v1.8
- The stemcell for v1.8.18 remains v3263

## Known issues

- None

## Version 1.8.14

**Release Date:** January 18, 2017

## Release Notes

- JMX Bridge v1.8.14 — BUG FIX: Fixes the issue identified in earlier v1.8.x versions where the JMX Bridge smoke tests would show an SSL connection error when running with SSL enabled.
- The stemcell for v1.8.14 remains v3263

## Known issues

- None

## Version 1.8.12

**Release Date:** January 5, 2017

## Release Notes

- JMX Bridge v1.8.12 — This maintenance patch makes an improvement to how the firehose connection to the JMX provider is tested. The prior `SlowConsumer` metric was not a consistent enough indicator that the firehose nozzle may need to scale. This patch introduces a metric `opentsdb.nozzle.totalFirehoseDisconnects` that increments the number of JMX Bridge firehose disconnects and logs each disconnect to STDOUT.
  - This metric can be used to help indicate a need to scale the product firehose nozzle. When experiencing truncated or dropped messages, first restart the Firehose Nozzle VM and check logs to validate it is connected. If several disconnects are logged, you should increase the number of OpenTSDB Firehose Nozzle instances in the tile Resource Config. Firehose reconnect attempts will show a delay of 5 seconds between attempts in order to reduce the risk of excessive calls to UAA. For more info on troubleshooting the firehose connection, see [troubleshooting](#).
- The stemcell for v1.8.12 remains v3263

## Known issues

- BUG IDENTIFIED: The integration tests will given an SSL connection error when running with SSL enabled. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.14 which contains a fix to this bug.

## Version 1.8.11

**Release Date:** December 15, 2016

## Release Notes

- JMX Bridge v1.8.11 — **Enhancement:** A new optional feature to enable support of an External IP in a NAT'd customer environment has been added for operators within the tile config under JMX Provider

- Operators are able to set the NAT IP as the host IP, at which point the JMX Provider can be connected to at the specified IP
- This enhancement addresses usage of JMX Bridge in a NAT'd network, where the product provided endpoint would become unavailable. While a manual configuration was possible to overcome this, the manual change would not hold through subsequent environment updates
- This feature is disabled by default. Once enabled, it will continue to persist through future updates until operator disabled
- See [how to enable](#)
- Release also includes a 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
- The stemcell for v1.8.11 remains v3263

## Known issues

- BUG IDENTIFIED: The integration tests will given an SSL connection error when running with SSL enabled. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.14 which contains a fix to this bug.

## Version 1.8.9

**Release Date: November 29, 2016**

## Release Notes

- JMX Bridge v1.8.9 — Earlier versions of Elastic Runtime v1.8 had an issue with the `provides_product_version` property. Elastic Runtime v1.8.17 and later correct this issue, which necessitated an update to how JMX Bridge v1.8 handles the property.
  - We recommend you upgrade to Elastic Runtime 1.8.17 or later, however JMX Bridge v1.8.9 can support Elastic Runtime v1.8.0 or later
  - Earlier versions of JMX Bridge (v1.8.2 through v1.8.8) will now only work with Elastic Runtime v1.8.0 through v1.8.16.
- The stemcell for v1.8.9 remains v3263

## Known issues

- BUG IDENTIFIED: The integration tests will given an SSL connection error when running with SSL enabled. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.14 which contains a fix to this bug.

## Version 1.8.8

**Release Date: November 22, 2016**

## Release Notes

- JMX Bridge v1.8.8 — BUG FIX: Fixes the issue identified in v1.8.7 where the new integration tests would fail if the doppler instances were greater than 1. As of this version, the integration tests now correctly handle multiple doppler instances.
- Stemcell for v1.8.8 remains v3263

## Known issues

- BUG IDENTIFIED: The integration tests will given an SSL connection error when running with SSL enabled. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.14 which contains a fix to this bug.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Version 1.8.7

**Release Date: November 10, 2016**

### Release Notes

- JMX Bridge v1.8.7 - Now enforces the dependency stated in the product documentation to install [Elastic Runtime](#) before installing JMX Bridge
  - Enforcing the correct installation order ensures a good connection between JMX Bridge and the firehose
  - If installation of JMX Bridge is attempted before Elastic Runtime, the user will now receive an installation error stating the required dependency
- **Enhancement:** The firehose nozzle is now enabled by default on installation
  - This enhancement addresses common feedback that the majority of use cases require the firehose to be enabled, and therefore eliminating the nozzle enablement step speeds product installation
  - For product use cases that do not include firehose nozzle usage, the OpenTSDB nozzle may still be disabled
  - OpenTSDB nozzle enabled/disabled preference will continue to be retained through future upgrades
- **Enhancement:** The tile now includes Smoke Tests which are executed via errand upon installation/upgrade
  - Any errors resulting from the smoke tests will be summarized at the end of the log output, and can be referenced against the product [troubleshooting](#) documentation
- Updates the OpenJDK version to v1.8.0.111
- Stemcell for v1.8.7 remains v3263

### Known issues

- BUG IDENTIFIED: The integration tests will fail if the doppler instances are greater than 1. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.8 which contains a fix to this bug.
- BUG IDENTIFIED: The integration tests will given an SSL connection error when running with SSL enabled. The resolution to this issue is to disable the smoke tests, or to upgrade to v1.8.14 which contains a fix to this bug.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Version 1.8.6

**Release Date: October 11, 2016**

### Release Notes

- JMX Bridge v1.8.6 — BUG FIX: As of PCF v1.8, the firehose began to send the dropsonde envelope index as a GUID instead of an integer. The OpenTSDB nozzle, if enabled, continued to assume parsing by integer causing some null values to be output. As of v1.8.6, the OpenTSDB nozzle now correctly parses the index field as a GUID.
- Stemcell for v1.8.6 remains v3263

### Known issues

- 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 optional and disabled by default, smoke tests for Elastic Runtime have been disabled until a later release.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Version 1.8.4

**Release Date:** September 22, 2016

### Release Notes

- JMX Bridge v1.8.4 — now with stemcell v3263

### Known issues

- BUG IDENTIFIED: Prior versions of JMX v1.8.x before v1.8.6 did not properly account for a Firehose change (appearing as of PCF v1.8), where the Firehose (aka Loggregator) began to send the dropsonde envelope index as a GUID instead of an integer. When enabled, the OpenTSDB nozzle parsed the index field as an integer, causing parsing failure. As of JMX v1.8.6 and later, this OpenTSDB parsing now assumes GUID instead of integer.
- 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 optional and disabled by default, smoke tests for Elastic Runtime have been disabled until a later release.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Version 1.8.3

**Release Date:** September 20, 2016

### Major Features

- Access to the JMX endpoint is now logged to STDOUT by default. This security logging can be enabled/disabled in the JMX Bridge tile configuration by checking/unchecking “Enable Security Logging” under JMX Provider.
  - Related log output is made available by initiating a JMX Provider logs download from the JMX Bridge tile configuration status tab, then fetching the download from the logs tab.

### Release Notes

- JMX Bridge v1.8.3 — BUG FIX: Corrects issue in v1.8.2 where the new security logging feature was not available to the operator in the tile config.
- Stemcell for v1.8.3 remains v3262

### Known issues

- BUG IDENTIFIED: Prior versions of JMX v1.8.x before v1.8.6 did not properly account for a Firehose change (appearing as of PCF v1.8), where the Firehose (aka Loggregator) began to send the dropsonde envelope index as a GUID instead of an integer. When enabled, the OpenTSDB nozzle parsed the index field as an integer, causing parsing failure. As of JMX v1.8.6 and later, this OpenTSDB parsing now assumes GUID instead of integer.
- 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 optional and disabled by default, smoke tests for Elastic Runtime have been disabled until a later release.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Version 1.8.2

**Release Date:** September 16, 2016

### Release Notes

- JMX Bridge v1.8.2 is targeted for PCF v1.8.X
- Stemcell for v1.8.2 is v3262


### Known issues

- BUG IDENTIFIED: The new security logging feature expected in JMX Bridge v1.8.X was not available to the operator in the tile config in version v1.8.2. JMX Bridge v1.8.3 corrects this.
- BUG IDENTIFIED: Prior versions of JMX v1.8.x before v1.8.6 did not properly account for a Firehose change (appearing as of PCF v1.8), where the Firehose (aka Loggregator) began to send the dropsonde envelope index as a GUID instead of an integer. When enabled, the OpenTSDB nozzle parsed the index field as an integer, causing parsing failure. As of JMX v1.8.6 and later, this OpenTSDB parsing now assumes GUID instead of integer.
- 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 optional and disabled by default, smoke tests for Elastic Runtime have been disabled until a later release.
- JMX Bridge v1.8.2 through v1.8.8 will only work with Elastic Runtime v1.8.0 through v1.8.16

## Metric Name Changes in PCF v1.8

In JMX Bridge v1.8, [metrics](#)  from a PCF deployment travel through two paths to the JMX Bridge endpoint: the Firehose and the BOSH Director.

In PCF v1.7 and JMX Bridge v1.7, the collector was still a metrics flow path for Cloud Controller, Gorouter, and ETCD metrics. With PCF v1.8 and JMX Bridge v1.8, the collector was retired and these components now broadcast via the Firehose.

Some older metrics may have been retired or had their names updated, see [PCF v1.8 metrics](#)  for more information on the individual component metrics.

In JMX Bridge, all metrics that travel through the JMX Bridge Firehose nozzle are prepended with the naming convention `opentsdb.nozzle.<metric>`

## Past Minor Version 1.7.X

Release Notes for v1.7.X releases can be found [here](#) .