Pivotal

PRODUCT DOCUMENTATION

JMX Bridge for PCF®

Documentation

Version 1.9

Published: 29 Jan 2019

Table of Contents

Pivotal Cloud Foundry JMX Bridge

⚠ **IMPORTANT:** The Pivotal Cloud Foundry (PCF) JMX Bridge tile is deprecated, and no further development will be made against this product.

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 VMs:

- The JMX provider
- A Nozzle for the Loggregator Firehose. For more information about how a Firehose nozzle works, see Nozzles.

Product Snapshot

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

Element	Details
Version	v1.9.9
Release date	February 9, 2018
Compatible Ops Manager version(s)	v1.11.3 or later, v1.12.x, v2.0.x, v2.1.x, v2.2.x
Compatible Elastic Runtime version(s)	v1.11.x, 1.12.x
Compatible Pivotal Application Service version(s)	v2.0.x, v2.1.x, v2.2.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

View the source for this page in GitHub

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.

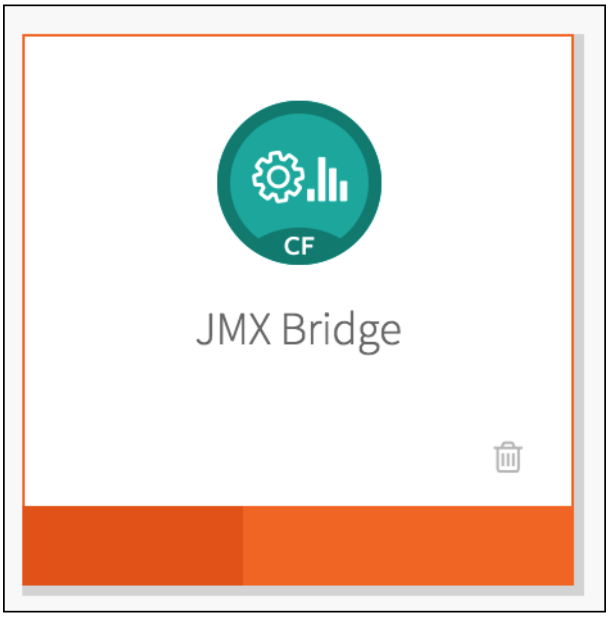
Step 1: Install the JMX Bridge Tile

Note: To use the Firehose Nozzle, you **must** install Elastic Runtime before JMX Bridge. Starting in v1.8.7, the JMX Bridge product enforces this install order dependency.

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.



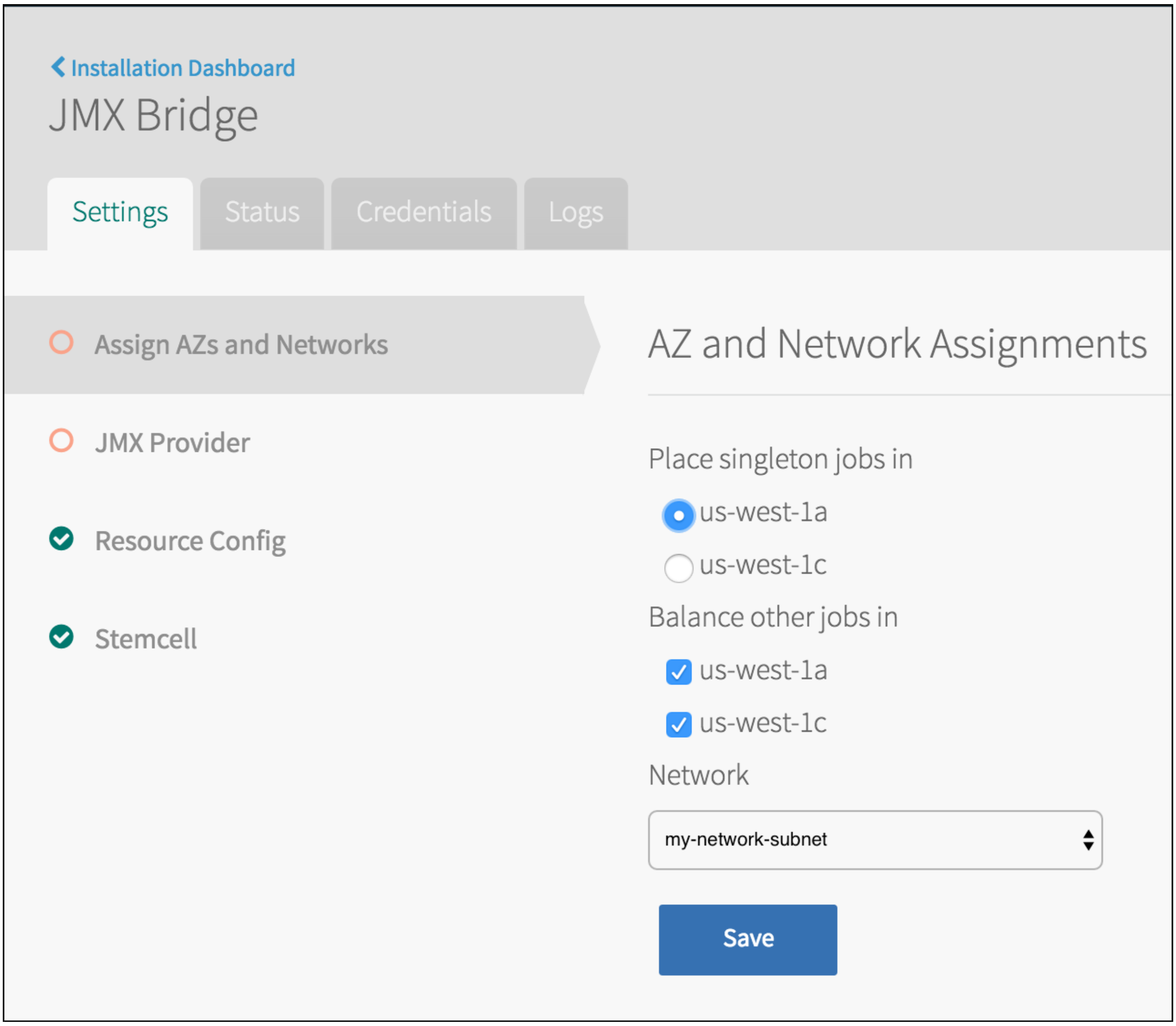
The orange bar on the **JMX Bridge** tile indicates that the product requires configuration.

Step 2: Assign Availability Zones and Networks

1. Select **Assign AZs and Networks**. This section shows the availability zones (AZs) that you create when configuring Ops Manager Director.

2. (**vSphere and Amazon Web Services Only**) Select an AZ under **Place singleton jobs in**. Ops Manager runs Metrics jobs with a single instance in this AZ.

3. (**vSphere and Amazon Web Services Only**) Select one or more AZ under **Balance other jobs in**. Ops Manager balances instances of Metrics jobs with more than one instance across the AZs that you specify.



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

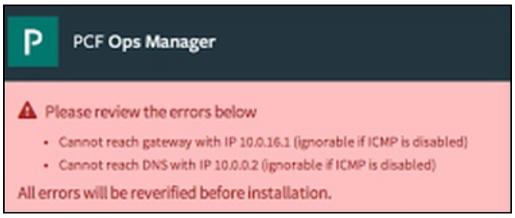
my-network-subnet

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, the following verification error displays because the PCF security group blocks ICMP. You can ignore this error.



Please review the errors below

- Cannot reach gateway with IP 10.0.16.1 (ignorable if ICMP is disabled)
- Cannot reach DNS with IP 10.0.0.2 (ignorable if ICMP is disabled)

All errors will be reverified before installation.

Step 3: Configure JMX Provider

1. Select **JMX Provider**.

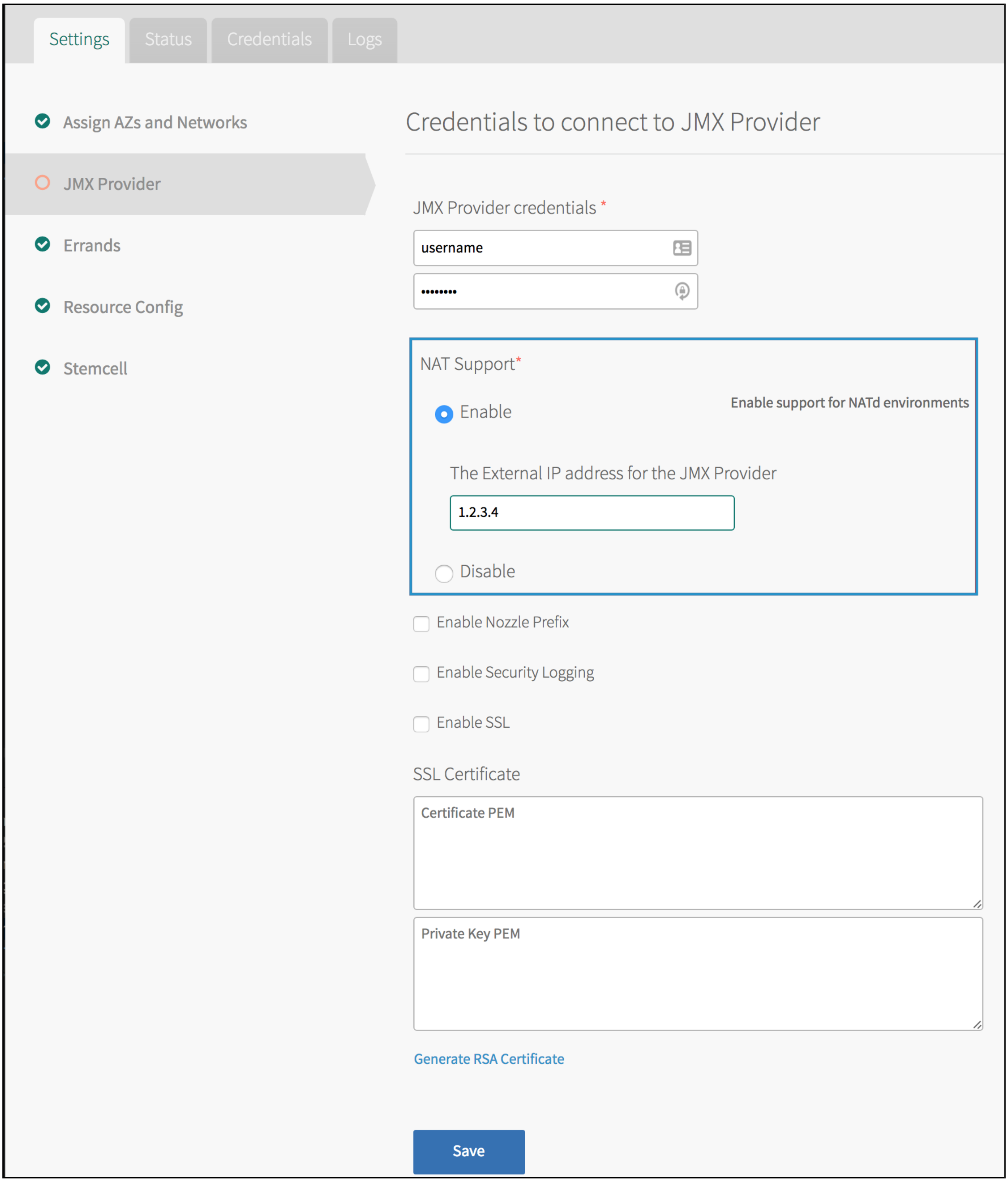
2. Enter a new username and password into the **JMX Provider credentials** username and password fields.

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

(Optional) Step 4: Disable or Enable NAT Support

1. Select the **Enable** radio button. NAT support is disabled by default. This option 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.

2. Enter the NAT IP as the External IP address in the form `0.0.0.0`



Credentials to connect to JMX Provider

JMX Provider credentials *

username

••••••••

NAT Support*

● Enable

Enable support for NATd environments

The External IP address for the JMX Provider

1.2.3.4

○ Disable

□ Enable Nozzle Prefix

□ Enable Security Logging

□ Enable SSL

SSL Certificate

Certificate PEM

Private Key PEM

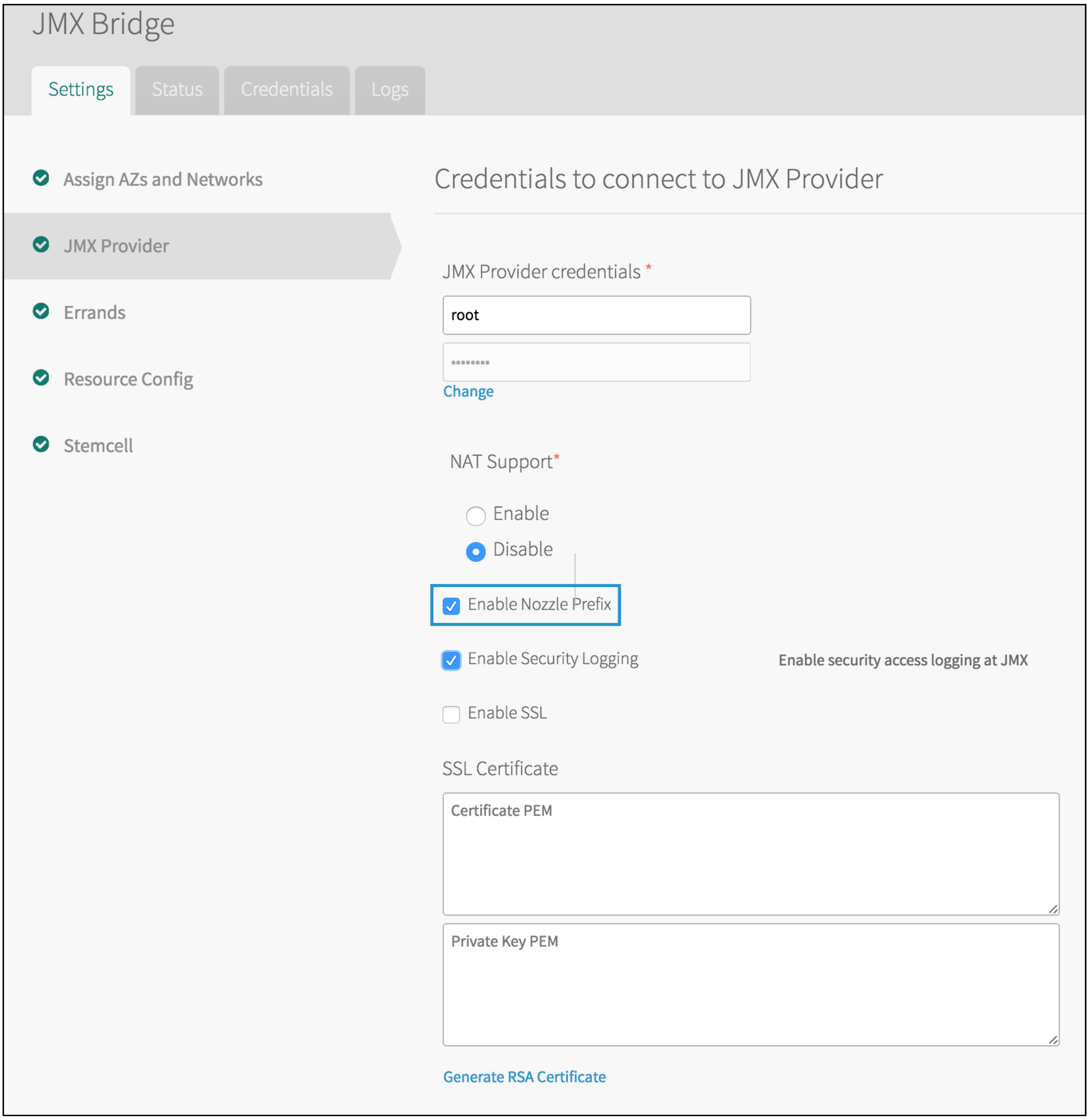
3. If you have enabled or disabled `NAT Support`, click **Save**.

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: Disable or Enable the Nozzle Prefix

By default, the Nozzle prefix `opentsdb.nozzle` prepends to Firehose-transmitted metrics, in order to maintain backward compatibility with prior versions of JMX Bridge.

1. If you do not require backward compatibility, you can make the metrics more readable by clearing the **Enable Nozzle Prefix** checkbox. This omits the `opentsdb.nozzle` prefix.



JMX Bridge

Credentials to connect to JMX Provider

JMX Provider credentials *

root

••••••••

Change

NAT Support*

○ Enable

● Disable

☑ Enable Nozzle Prefix

☑ Enable Security Logging — Enable security access logging at JMX

□ Enable SSL

SSL Certificate

Certificate PEM

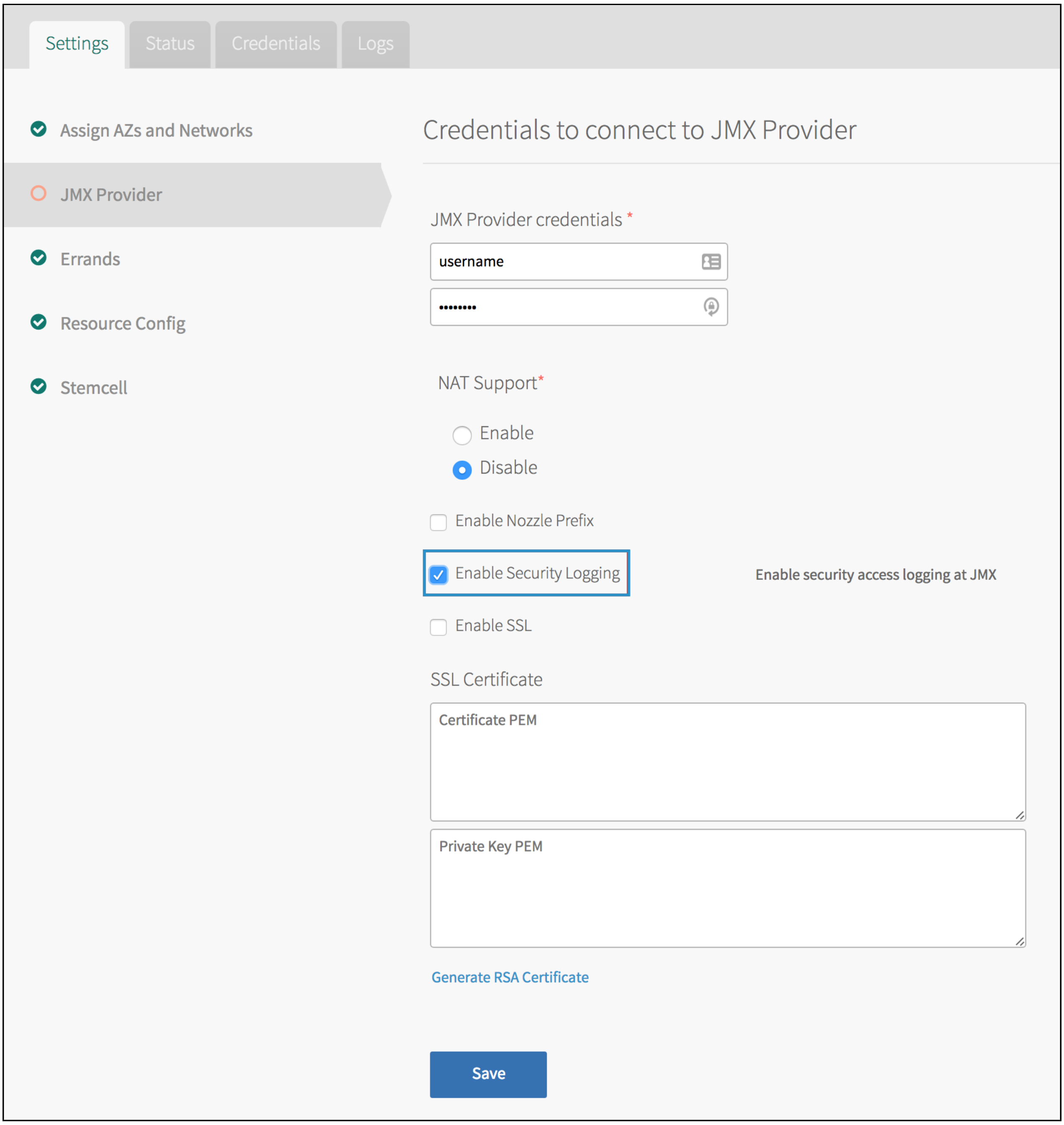
Private Key PEM

Generate RSA Certificate

2. If you have enabled or disabled `Enable Nozzle Prefix`, click **Save**. Enabling or disabling this feature causes temporary overlap of metrics coming through in both naming formats. Metrics with the former metric name format continue to appear in the MBean tree until the JMX Provider is restarted or the MBean store needs additional space to store new metrics.

(Optional) Step 6: Disable or Enable Security Logging

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



Credentials to connect to JMX Provider

JMX Provider credentials *

username

••••••••

NAT Support*

○ Enable

● Disable

□ Enable Nozzle Prefix

☑ Enable Security Logging — Enable security access logging at JMX

□ Enable SSL

SSL Certificate

Certificate PEM

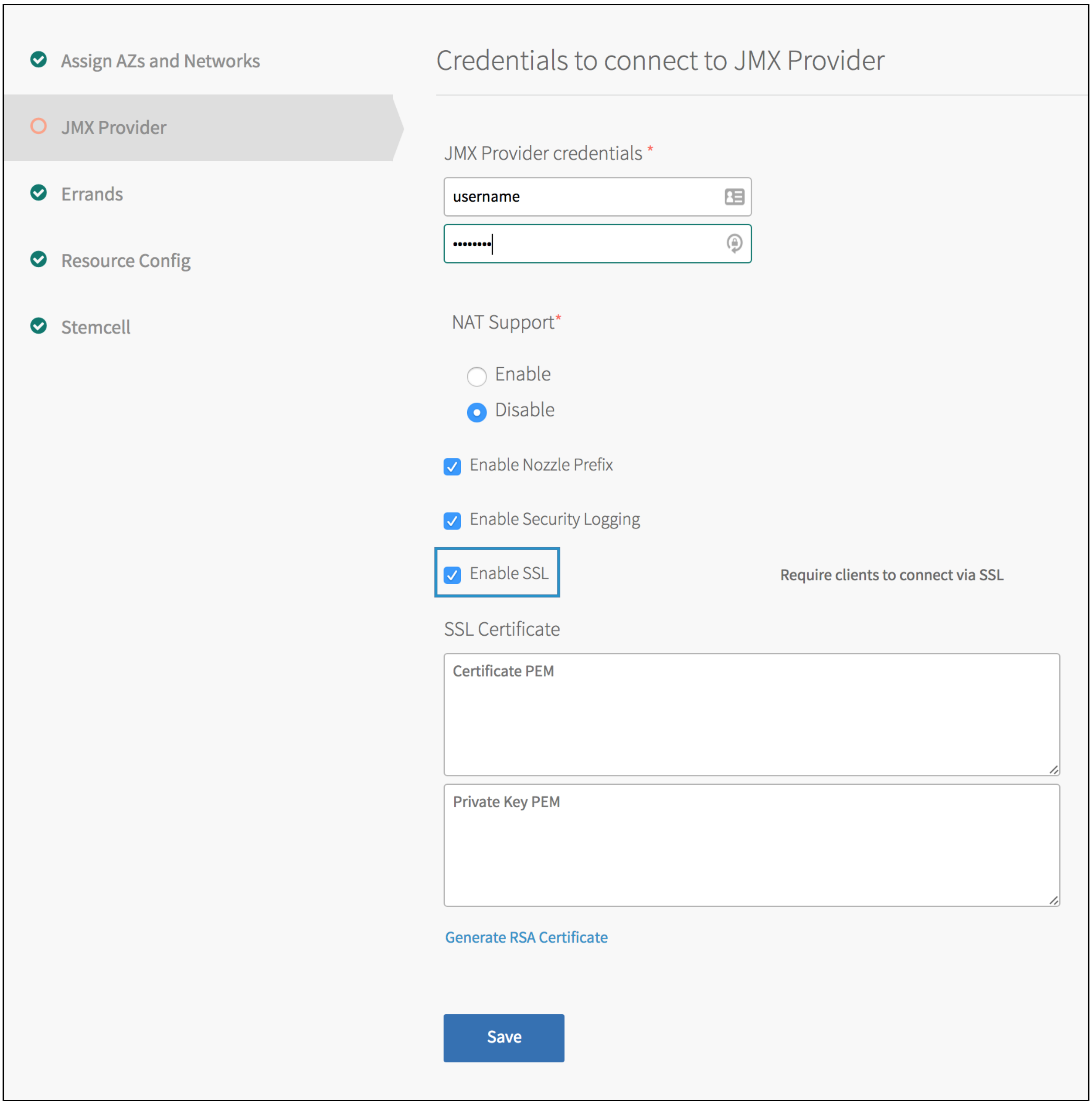
Private Key PEM

2. If you have enabled or disabled `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 7: 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.



Credentials to connect to JMX Provider

JMX Provider credentials *

username

••••••••

NAT Support*

○ Enable
● Disable

☑ Enable Nozzle Prefix

☑ Enable Security Logging

☑ Enable SSL — Require clients to connect via SSL

SSL Certificate

Certificate PEM

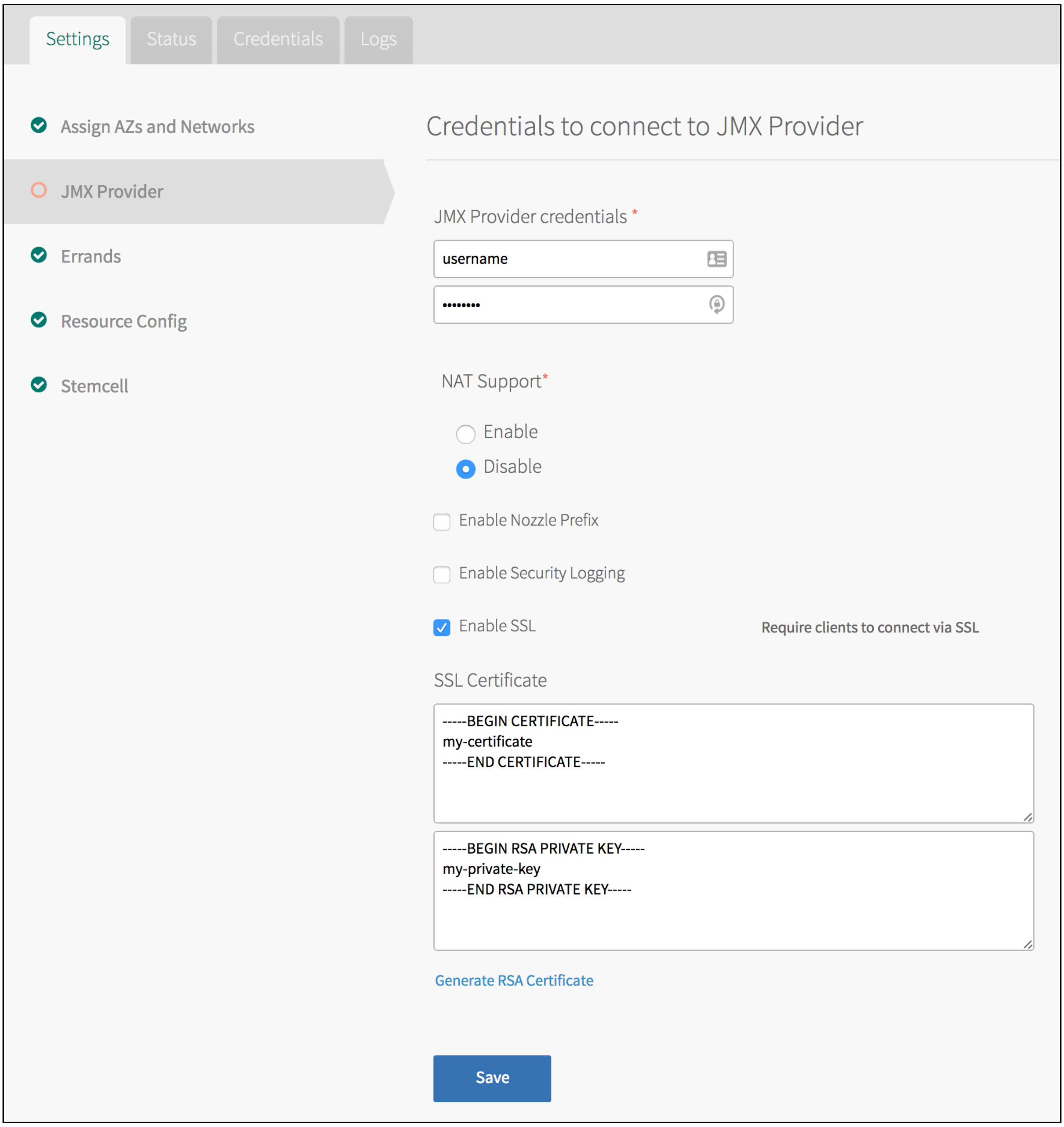
Private Key PEM

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.



Credentials to connect to JMX Provider

JMX Provider credentials *

username

••••••••

NAT Support*

○ Enable

● Disable

□ Enable Nozzle Prefix

□ Enable Security Logging

☑ Enable SSL — Require clients to connect via SSL

SSL Certificate

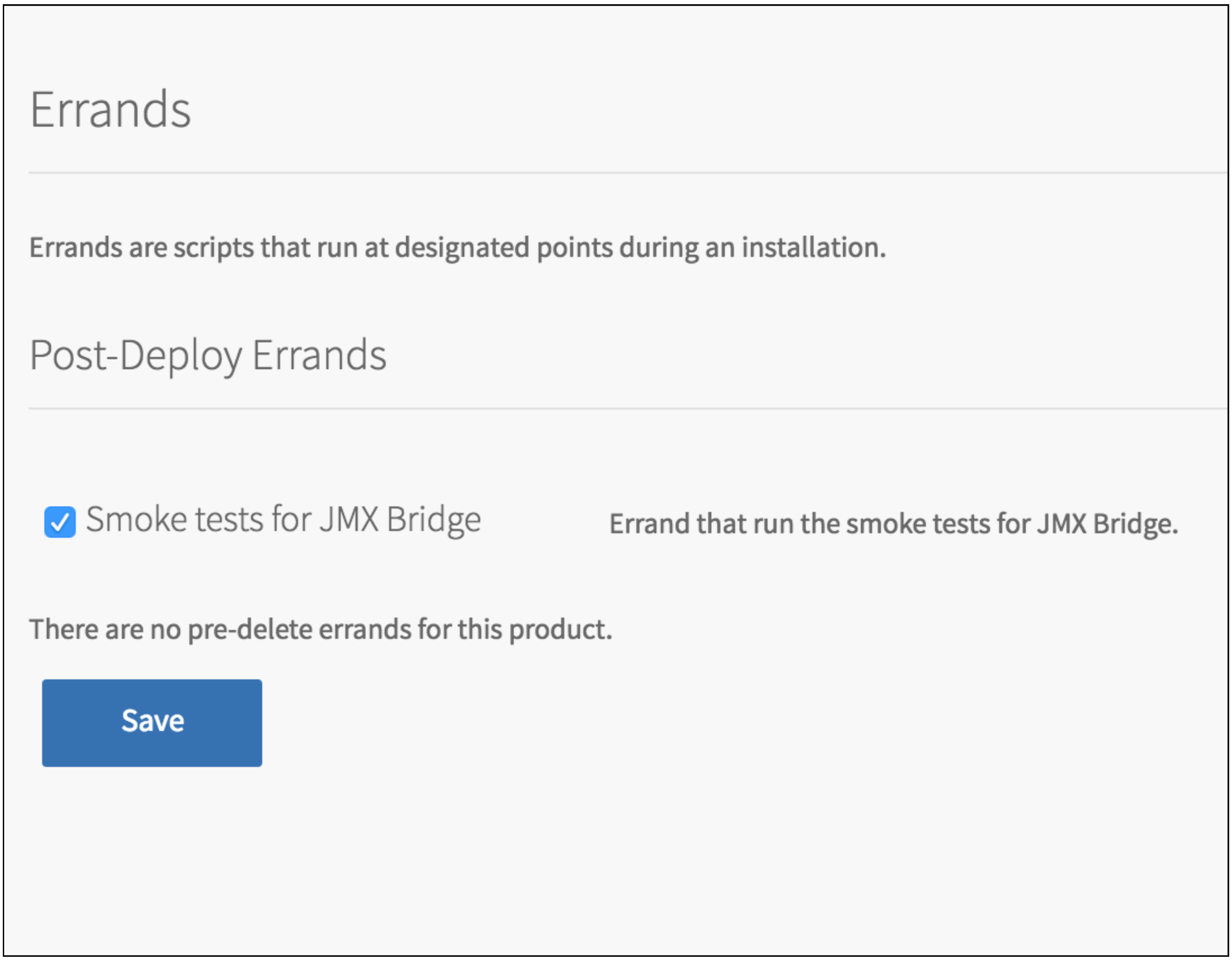
```
-----BEGIN CERTIFICATE-----
my-certificate
-----END CERTIFICATE-----
```

```
-----BEGIN RSA PRIVATE KEY-----
my-private-key
-----END RSA PRIVATE KEY-----
```

2. After providing an SSL certificate and private key, click **Save**.

(Optional) Step 8: 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.

There are no pre-delete errands for this product.

- Select **Smoke tests for JMX Bridge**to cause the JMX Bridge to verify the following:
  - If the Firehose Nozzle is enabled, that the Nozzle is receiving metrics
  - If the Firehose Nozzle is enabled, that the product is not a slow consumer
  - If BOSH Metrics are enabled, 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.

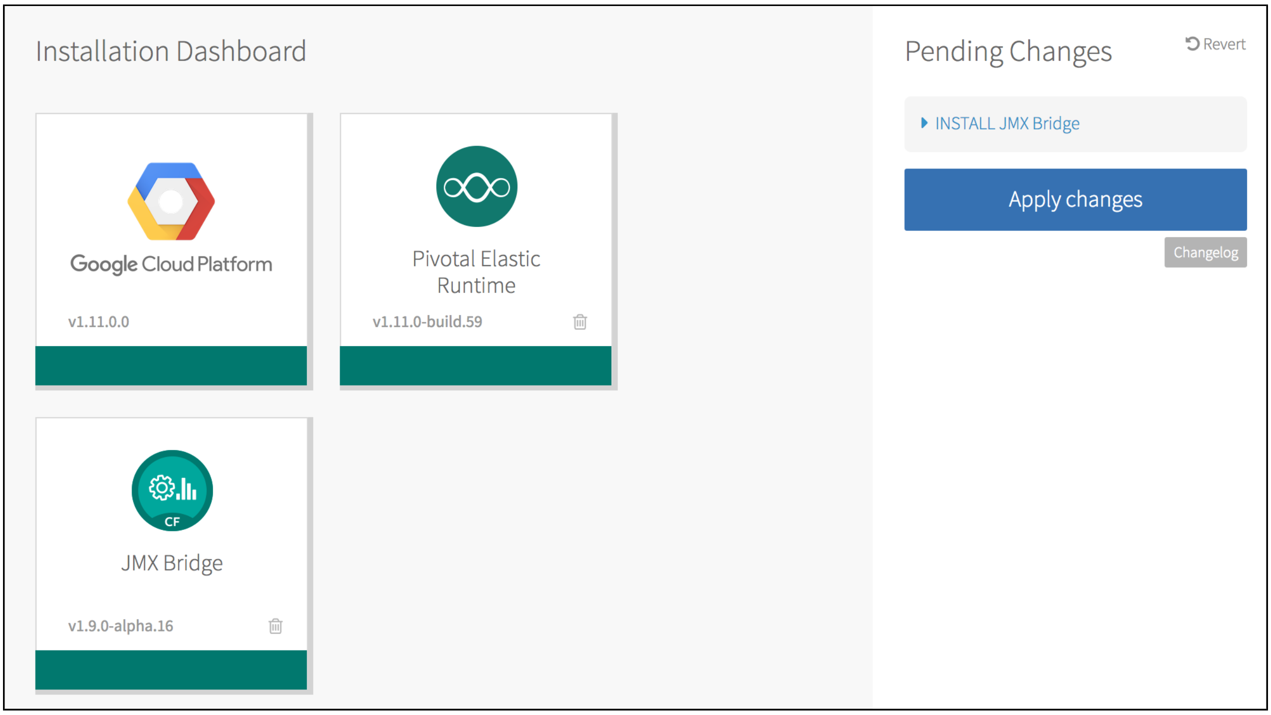
(Optional) Step 9: Resource Configuration

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

Step 10: 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.

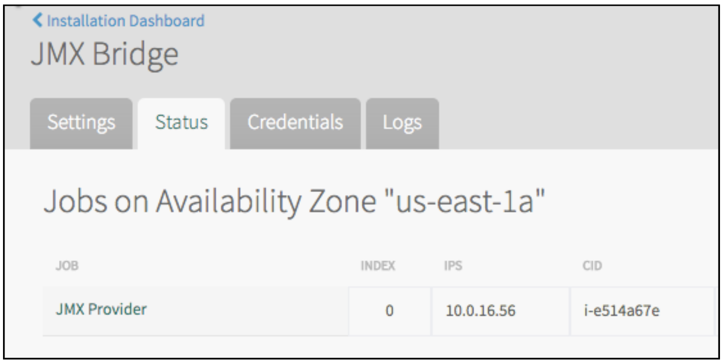
Step 11: Provide a JMX Provider IP Address

If you want to consume BOSH system metrics with JMX Bridge, follow the steps below.

> **Note**: In PCF v2.0 and later, BOSH system metrics are available through the Loggregator Firehose. This means JMX Bridge consumes them by default through its Firehose nozzle and you do not need to complete this section. If you do, you may receive duplicate data. For more information, see the v1.9.5 section of *Release Notes and Known Issues*.

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.



Jobs on Availability Zone "us-east-1a"

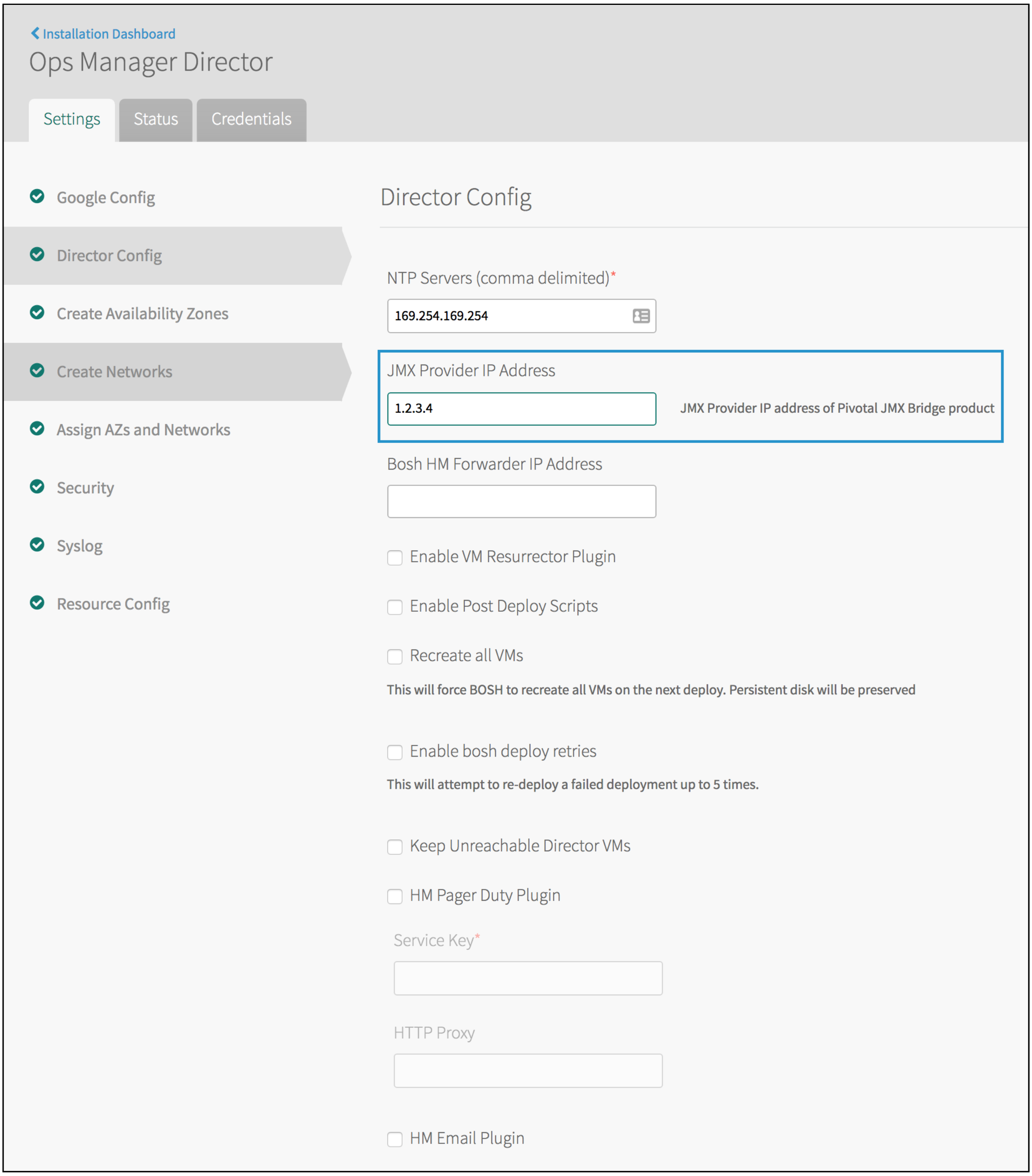
JOB	INDEX	IPS	CID
JMX Provider	0	10.0.16.56	i-e514a67e

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.

Enter the JMX Provider IP Address

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



Director Config

NTP Servers (comma delimited)*

169.254.169.254

JMX Provider IP Address

1.2.3.4

JMX Provider IP address of Pivotal JMX Bridge product

Bosh HM Forwarder IP Address

□ Enable VM Resurrector Plugin

□ Enable Post Deploy Scripts

□ Recreate all VMs

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

□ Enable bosh deploy retries

This will attempt to re-deploy a failed deployment up to 5 times.

□ Keep Unreachable Director VMs

□ HM Pager Duty Plugin

Service Key*

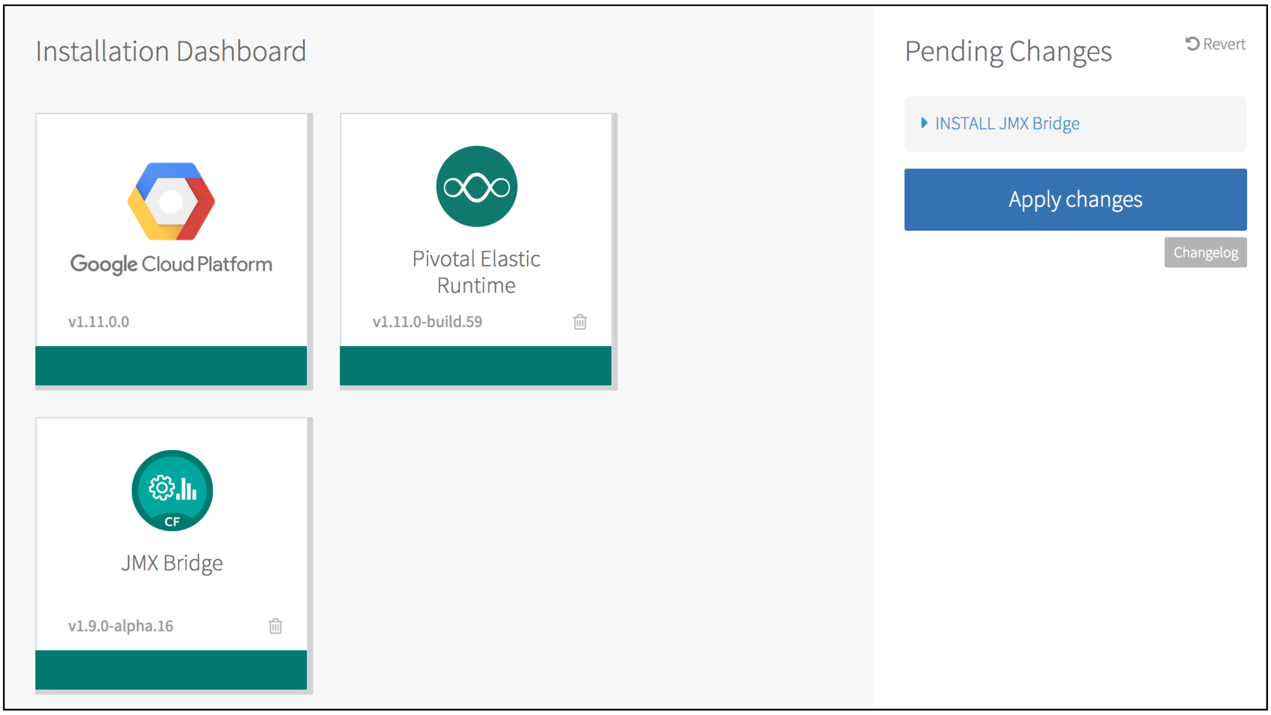
HTTP Proxy

□ HM Email Plugin

2. In the **JMX Provider 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**.



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.

View the source for this page in GitHub

Using JMX Bridge

Page last updated:

JMX Bridge is a Java Management Extensions (JMX) tool for Pivotal Application Service. To help you monitor your installation and assist in troubleshooting, JMX Bridge collects and exposes system data from Pivotal Cloud Foundry components via a JMX endpoint. JMX Bridge consumes platform metric data from the Firehose via the `firehose-jmx-nozzle` and system health metrics directly from BOSH.

Guidance on Key Metrics of Monitoring Interest

JMX Bridge reports all platform metric data being transmitted from PCF via BOSH and the Firehose. Not all platform metrics are of equal interest when monitoring PCF. Reference Monitoring Pivotal Cloud Foundry for recommendations of key indicators.

Example - Virtual Machine Metrics

JMX Bridge reports data for each virtual machine (VM) in a deployment. Use these types of 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)

View the source for this page in GitHub

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

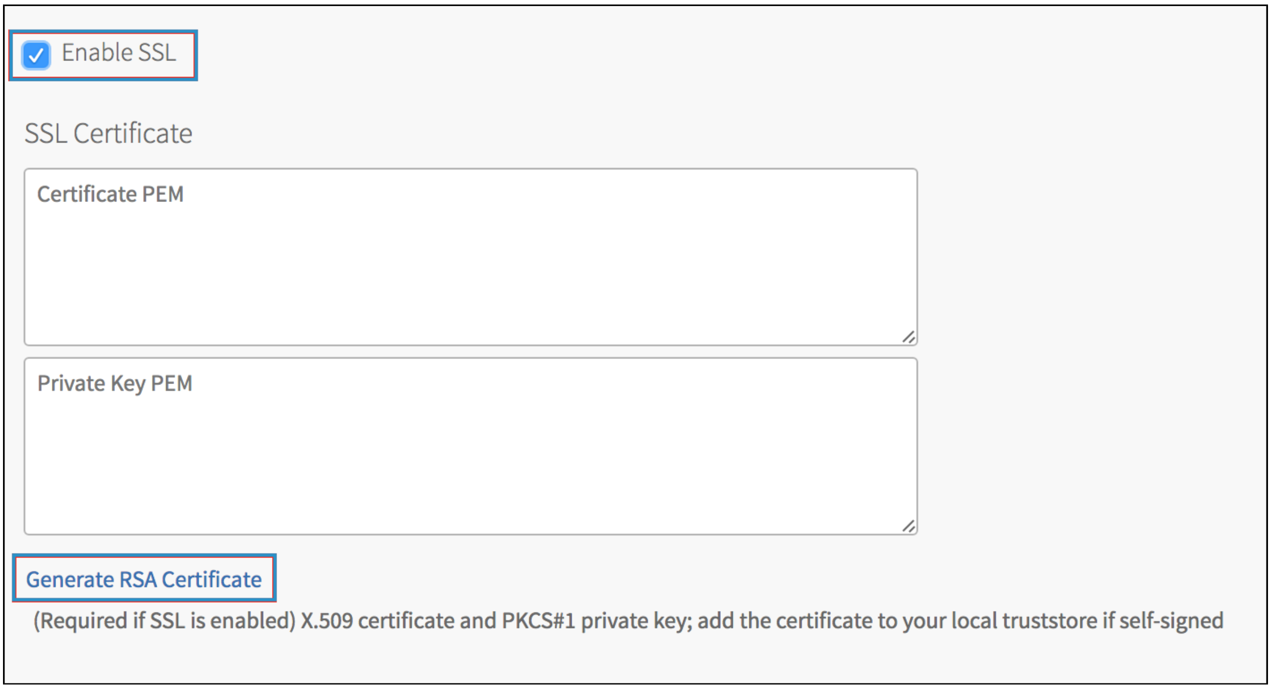
Option 1: Generate Self-Signed Certificate

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

1. In Pivotal Ops Manager, click the **JMX Bridge** tile.

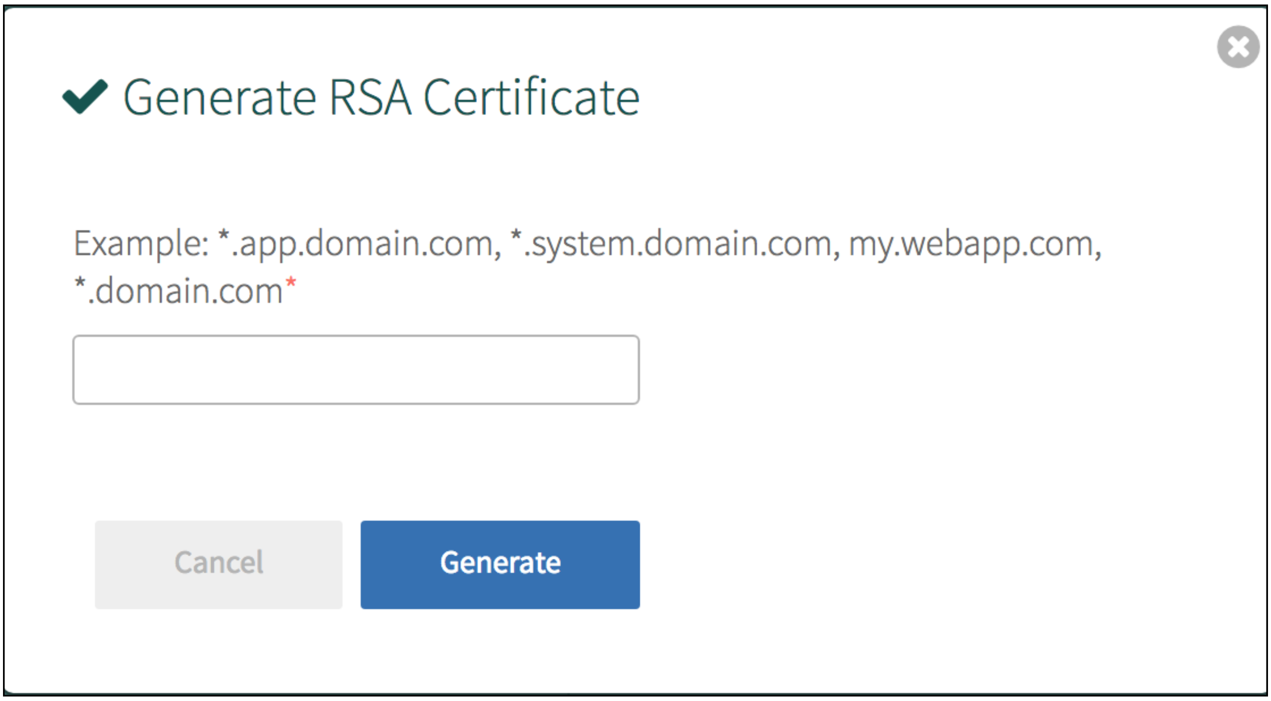
2. Check **Enable SSL**.

3. Click **Generate Self-Signed RSA Certificate**.



(Required if SSL is enabled) X.509 certificate and PKCS#1 private key; add the certificate to your local truststore if self-signed

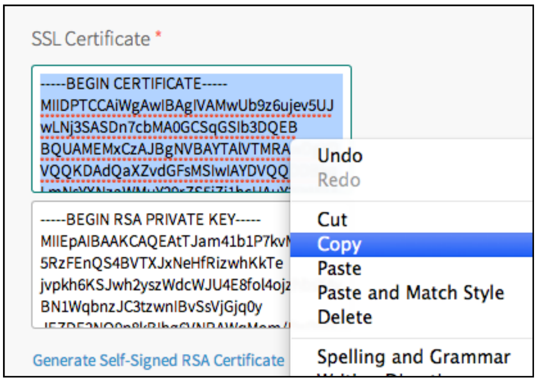
4. Enter your system and application domains in wildcard format. Optionally, also add any custom domains in wildcard format. Click **Generate.**



Generate RSA Certificate

Example: *.app.domain.com, *.system.domain.com, my.webapp.com, *.domain.com*

5. Select and copy the certificate.



SSL Certificate *

```
-----BEGIN CERTIFICATE-----
MIIDPTCCAiWgAwIBAgIVAMwUb9z6ujev5UJ
wLNj3SASDn7cbMA0GCSqGSIb3DQEB
BQUAMEMxCzAJBgNVBAYTAlVTMRA
VQQKDAdQaXZvdGFsMSIwIAYDVQQ
```

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAtTJam41b1P7kvN
5RzFEnQS4BVTXJxNeHfRizwhKkTe
jvpkh6KSJwh2yszWdcWJU4E8fol4oj
BN1WqbnzJC3tzwnlBvSsVjGjq0y
```

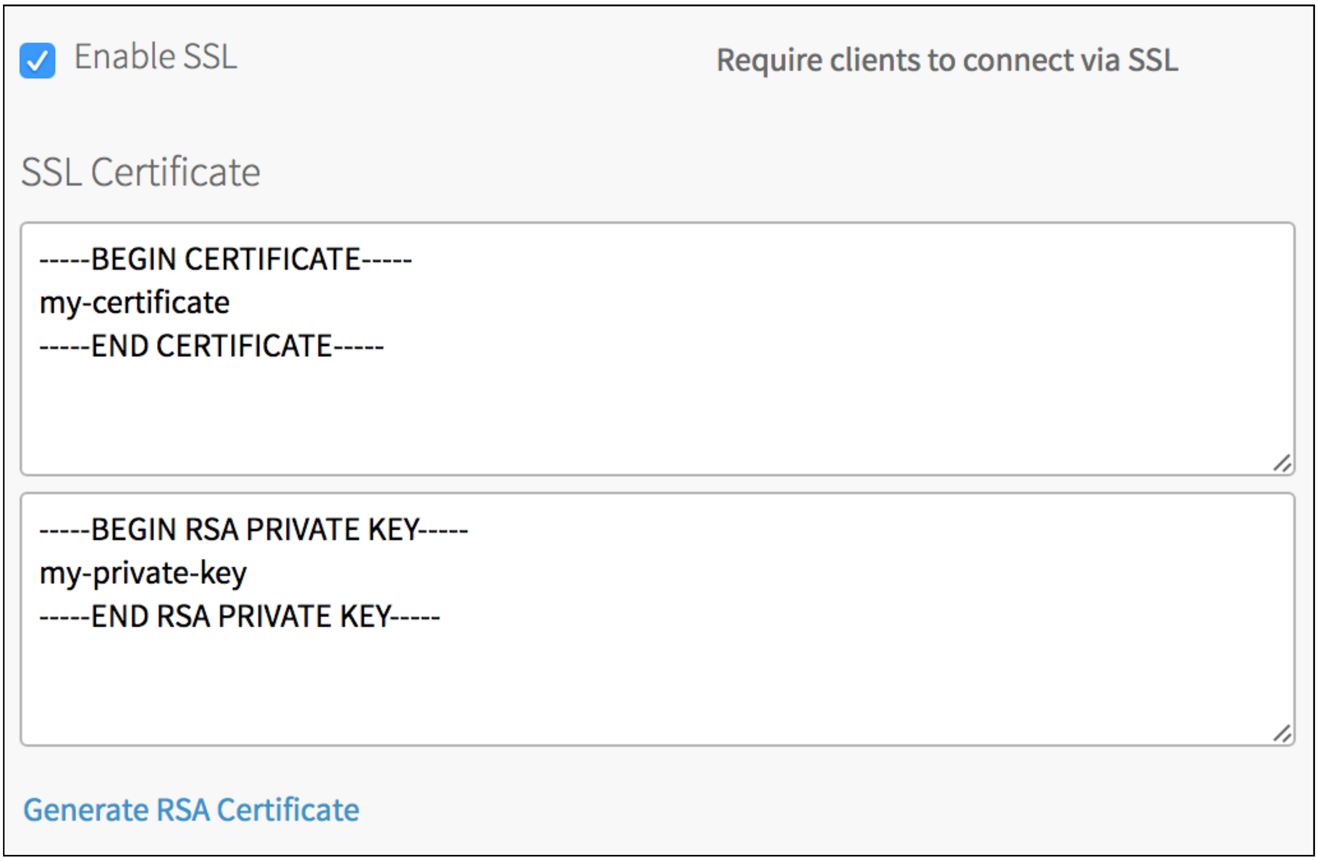
6. Paste the certificate into a text file and save as a `.cer` file, such as `MY-JMX-BRIDGE.cer`.

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.



☑ Enable SSL

Require clients to connect via SSL

SSL Certificate

```
-----BEGIN CERTIFICATE-----
my-certificate
-----END CERTIFICATE-----
```

```
-----BEGIN RSA PRIVATE KEY-----
my-private-key
-----END RSA 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.

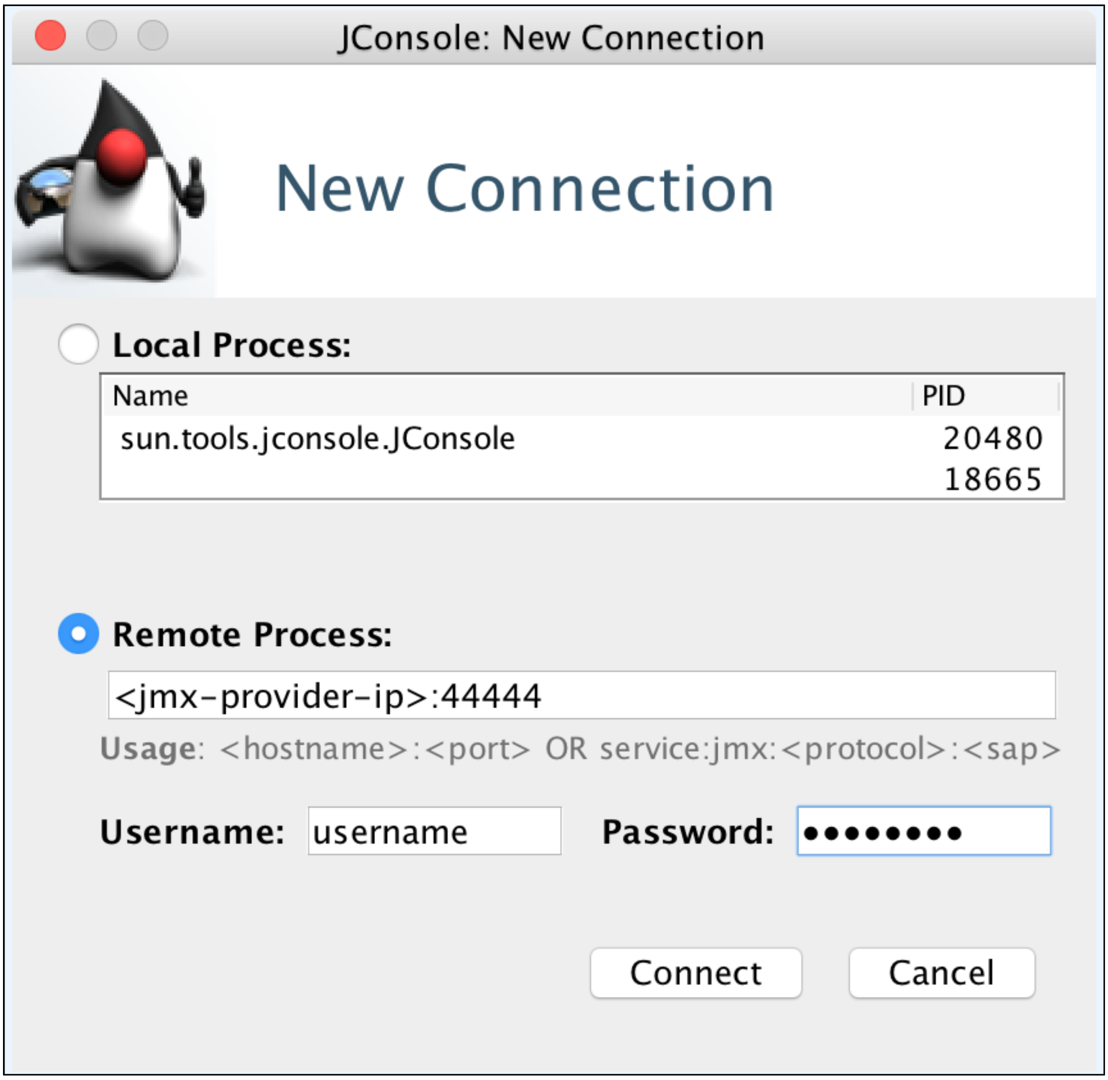
Step 3: Start a Monitoring Tool with the Truststore

After 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` .



JConsole: New Connection

New Connection

○ Local Process:

Name	PID
sun.tools.jconsole.JConsole	20480
	18665

● Remote Process:

<jmx-provider-ip>:44444

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

Username: username Password: ••••••••

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.

View the source for this page in GitHub

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	4 GB	8 GB	1 GB	1	0
JMX Bridge	JMX Firehose Nozzle	1	2	4 GB	8 GB	1 GB	1	0
JMX Bridge	Smoke Tests	1	4	4 GB	8 GB	0	1	0

Guidelines

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

View the source for this page in GitHub

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.

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:

1. SSH into the **jmx-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 `jmx-firehose-nozzle` job.

   ```
   $ monit restart jmx-firehose-nozzle
   ```

Missing BOSH Metrics

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

1. Make sure the IP address in **JMX Bridge > Status > JMX Provider**matches the value entered in **Ops Manager Director > Director Config > JMX Provider IP Address**.

2. If the addresses do not match 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:

1. Verify Java 6+ is installed.

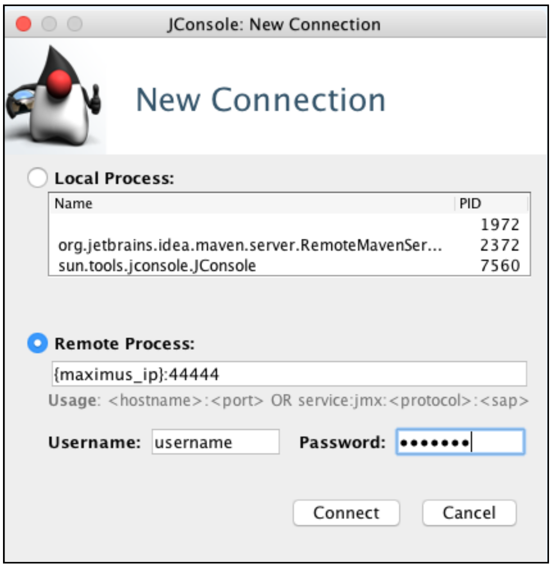
2. Run `jconsole`:

   ```
   $ jconsole
   ```

3. Select **Remote Process** and enter the IP of the **JMX Provider** VM with port `44444`.

4. Fill in the username and password for the **JMX Provider** that was entered during installation of JMX Bridge.

5. Click **Connect**.



○ Local Process:

Name	PID
	1972
org.jetbrains.idea.maven.server.RemoteMavenSer...	2372
sun.tools.jconsole.JConsole	7560

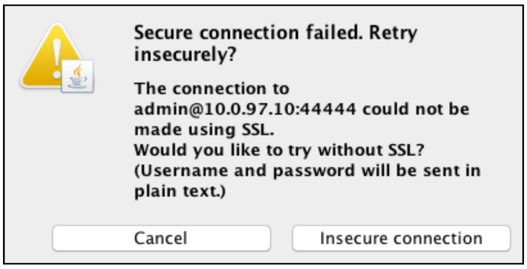
● Remote Process:

{maximus_ip}:44444

Usage: <hostname>:<port> OR service:jmx:<protocol>:<sap>

Username: username Password: •••••••

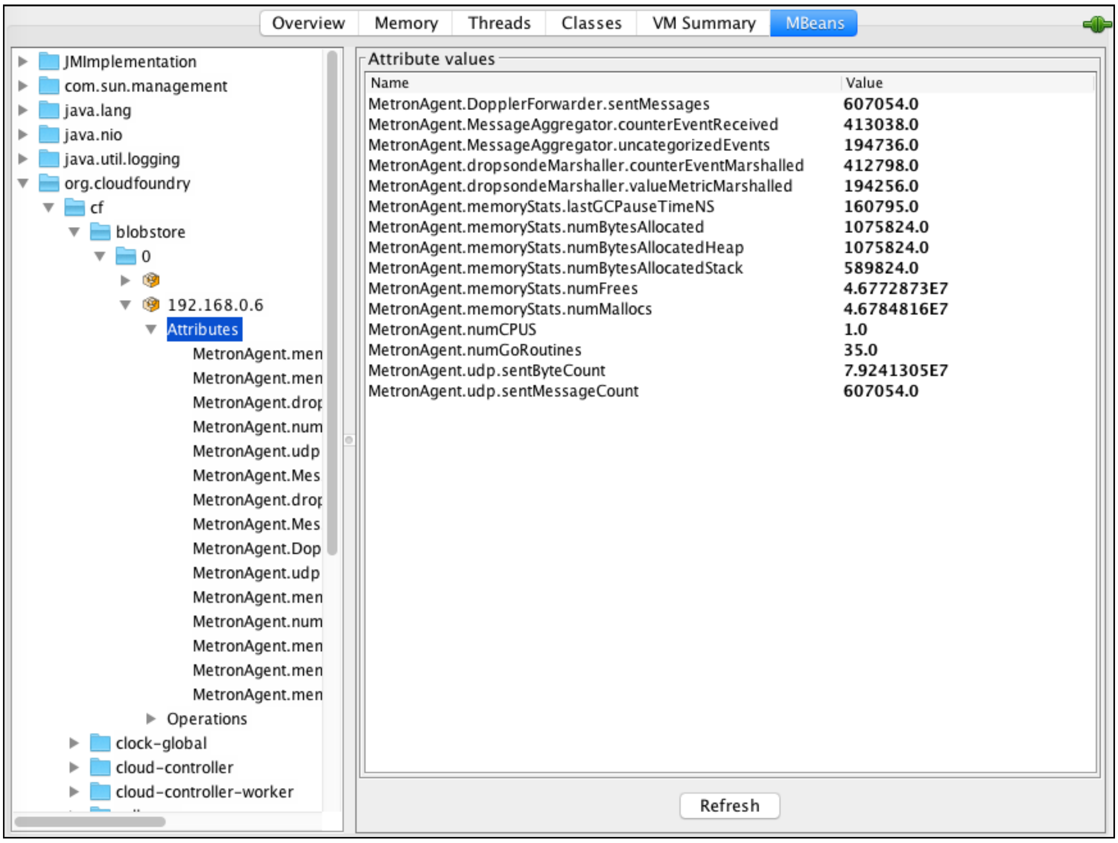
6. Allow **Insecure connection** if SSL was not enabled.



Secure connection failed. Retry insecurely?

**The connection to admin@10.0.97.10:44444 could not be made using SSL.
Would you like to try without SSL?
(Username and password will be sent in plain text.)**

You can now view all MBeans emitted by JMX Bridge.



Name	Value
MetronAgent.DopplerForwarder.sentMessages	607054.0
MetronAgent.MessageAggregator.counterEventReceived	413038.0
MetronAgent.MessageAggregator.uncategorizedEvents	194736.0
MetronAgent.dropsondeMarshaller.counterEventMarshalled	412798.0
MetronAgent.dropsondeMarshaller.valueMetricMarshalled	194256.0
MetronAgent.memoryStats.lastGCPauseTimeNS	160795.0
MetronAgent.memoryStats.numBytesAllocated	1075824.0
MetronAgent.memoryStats.numBytesAllocatedHeap	1075824.0
MetronAgent.memoryStats.numBytesAllocatedStack	589824.0
MetronAgent.memoryStats.numFrees	4.6772873E7
MetronAgent.memoryStats.numMallocs	4.6784816E7
MetronAgent.numCPUS	1.0
MetronAgent.numGoRoutines	35.0
MetronAgent.udp.sentByteCount	7.9241305E7
MetronAgent.udp.sentMessageCount	607054.0

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.

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

Error	`receivingFirehoseMetrics() Fails`
Cause	The JMX Firehose Nozzle is not receiving metrics from the Firehose.
Solution	Restart the JMX 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 JMX Firehose Nozzle instances in the **Resource Config** tab.

Uninstall

To uninstall the JMX Bridge for PCF tile, see Deleting a Product.

View the source for this page in GitHub

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.

View the source for this page in GitHub

Release Notes and Known Issues

v1.9.9

Release Date: February 12, 2018

Release Notes

- Stemcell for v1.9.9 is now v3468
- Network connections will time out and reconnect if no new metrics are received for 30 seconds.

Known issues

- The new flow of BOSH system metrics (see release notes for v1.9.5 for more details) cannot be disabled. Therefore, if you are currently using the PCF JMX Bridge tile to consume them, you may receive duplicate data. To prevent this, delete **JMX Provider IP Address** in **Director Config** of your Ops Manager Director tile.

  Deleting the IP address means that BOSH system metrics will no longer be sent to JMX Bridge using the direct connection from the BOSH Director to the JMX Provider. As these BOSH system metrics are now available in JMX Bridge by default through its Firehose nozzle, breaking the prior direct connection by deleting the JMX Provider IP address prevents the duplication of BOSH metrics for JMX Bridge consumers.

v1.9.5

Release Date: November 15, 2017

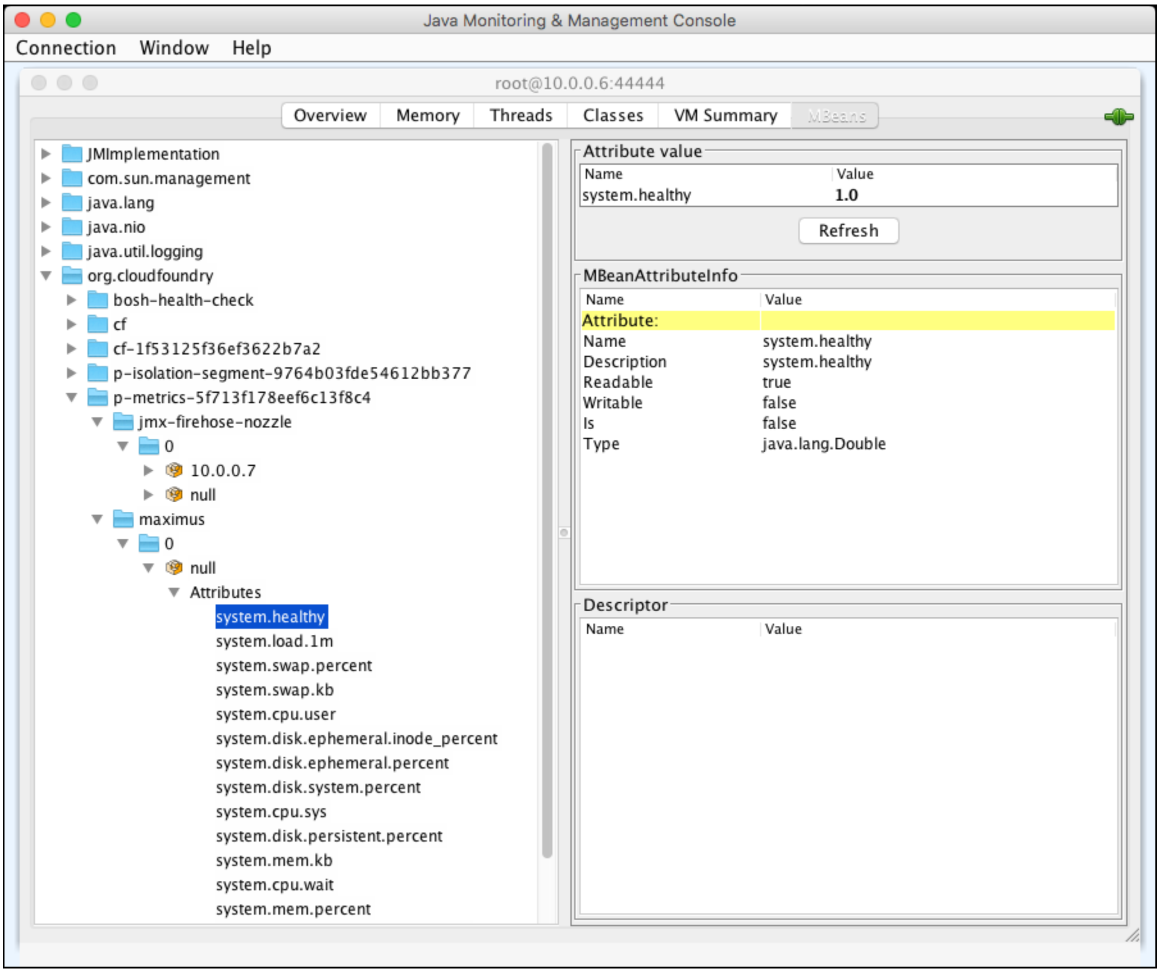
Release Notes

- Stemcell for v1.9.5 is now v3445
- PCF now forwards BOSH health metrics generated for all VMs in a deployment to the Loggregator Firehose by default. For more information about this feature and its implementation, see the *BOSH System Metrics Forwarder* section in the Overview of the Loggregator System.

  The new flow of BOSH system metrics cannot be disabled. Therefore, if you are currently using the PCF JMX Bridge tile or the BOSH HM Forwarder to consume them, you may receive duplicate data. To prevent this, you can do the following:

  - Stop using PCF JMX Bridge to consume BOSH system metrics outside of the Firehose. See Known Issues.
  - Uninstall the BOSH HM Forwarder.
- Because BOSH System metrics now come from the Firehose, their namespaces are different in PCF JMX Bridge. For an explanation of how metric names differ between PCF 2.0 and earlier versions, see the following table.

PCF Version	Explanation
1.12 and earlier	Example Metric: `system.healthy` Description: The BOSH Director delivers the metric name. The metric is nested in the tree structure by deployment name, VM name, VM instance number, and attributes for that VM instance. The sub-node of VM instance number is always named null. Reference Image:



Attribute value

Name	Value
system.healthy	1.0

MBeanAttributeInfo

Name	Value
Attribute:	
Name	system.healthy
Description	system.healthy
Readable	true
Writable	false
Is	false
Type	java.lang.Double

Descriptor

Name	Value

2.0

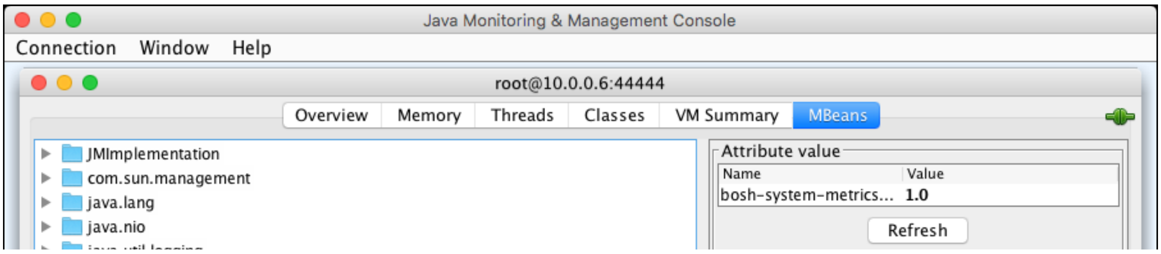
Example Metric:

`bosh-system-metrics-forwarder.system.healthy`

Description:

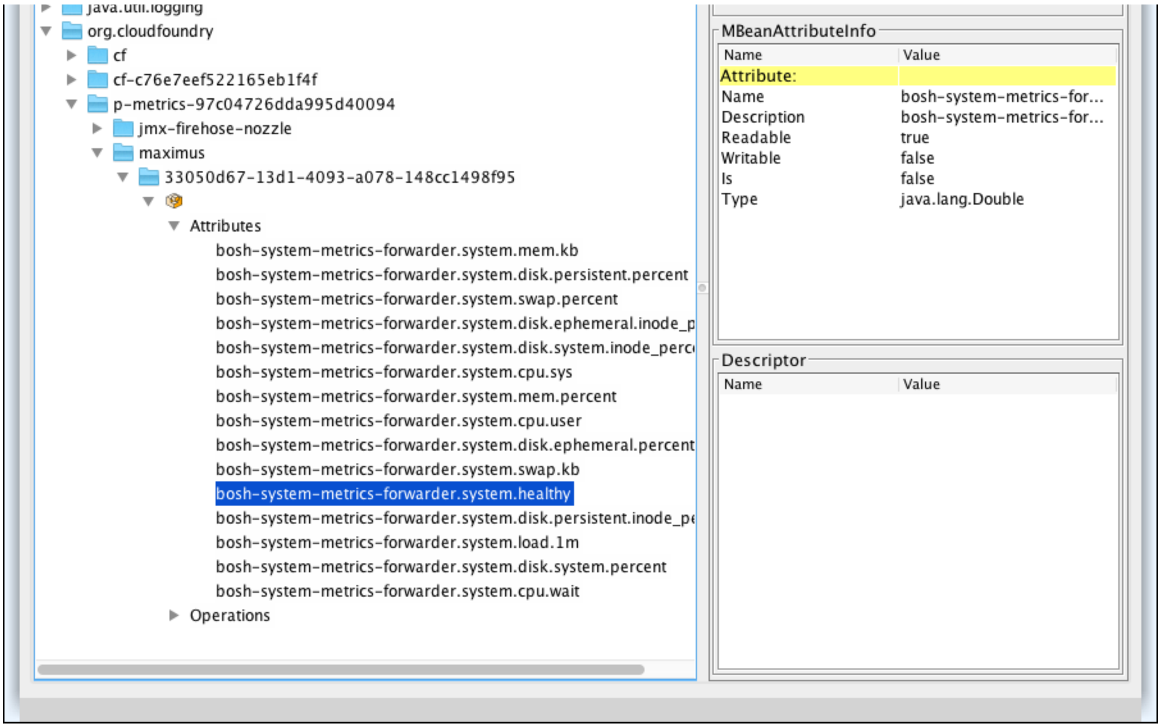
The Firehose delivers the metric name. The tree shows the VM GUID instead of the VM instance number and the sub-node is always empty. This namespacing affects all previous BOSH health metrics.

Reference Image:



Attribute value

Name	Value
bosh-system-metrics...	1.0



```
▶ java.util.logging
▼ org.cloudfoundry
  ▶ cf
  ▶ cf-c76e7eef522165eb1f4f
  ▼ p-metrics-97c04726dda995d40094
    ▶ jmx-firehose-nozzle
    ▼ maximus
      ▼ 33050d67-13d1-4093-a078-148cc1498f95
        ▼
          ▼ Attributes
              bosh-system-metrics-forwarder.system.mem.kb
              bosh-system-metrics-forwarder.system.disk.persistent.percent
              bosh-system-metrics-forwarder.system.swap.percent
              bosh-system-metrics-forwarder.system.disk.ephemeral.inode_p
              bosh-system-metrics-forwarder.system.disk.system.inode_perc
              bosh-system-metrics-forwarder.system.cpu.sys
              bosh-system-metrics-forwarder.system.mem.percent
              bosh-system-metrics-forwarder.system.cpu.user
              bosh-system-metrics-forwarder.system.disk.ephemeral.percent
              bosh-system-metrics-forwarder.system.swap.kb
              bosh-system-metrics-forwarder.system.healthy
              bosh-system-metrics-forwarder.system.disk.persistent.inode_pe
              bosh-system-metrics-forwarder.system.load.1m
              bosh-system-metrics-forwarder.system.disk.system.percent
              bosh-system-metrics-forwarder.system.cpu.wait
          ▶ Operations
```

MBeanAttributeInfo

Name	Value
Attribute:	
Name	bosh-system-metrics-for...
Description	bosh-system-metrics-for...
Readable	true
Writable	false
Is	false
Type	java.lang.Double

Descriptor

Name	Value

Known issues

- The new flow of BOSH system metrics cannot be disabled. Therefore, if you are currently using the PCF JMX Bridge tile to consume them, you may receive duplicate data. To prevent this, delete **JMX Provider IP Address** in **Director Config** of your Ops Manager Director tile.

  Deleting the IP address means that BOSH system metrics will no longer be sent to JMX Bridge using the direct connection from the BOSH Director to the JMX Provider. As these BOSH system metrics are now available in JMX Bridge by default through its Firehose nozzle, breaking the prior direct connection by deleting the JMX Provider IP address prevents the duplication of BOSH metrics for JMX Bridge consumers.

v1.9.3

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
- To ensure compatibility with upcoming versions of PCF, JMX Bridge will no longer be statically allocating IPs. However, once installed, the allocated IP will remain the same through future upgrades provided the tile has not been uninstalled. Any tooling that currently relies on obtaining the allocated IP through the Ops Manager's `GET /api/v0/deployed/products/:product_guid/static_ips` endpoint can now use the `GET /api/v0/deployed/products/:product_guid/status` endpoint to retrieve this information.
- Stemcell for v1.9.3 remains v3363

Known issues

- None

v1.9.2

Release Date: August 11, 2017

Release Notes

- Maintenance update of the following product dependencies:
  - OpenJDK now v1.8.0.141
  - Golang now v1.8.3
  - GRPC-ALL now v1.5.0
  - Guava now v23.0
  - Bouncy Castle-ALL now v1.57
  - netty-tcnative-boringssl-static now 2.0.5.Final
- JMX Bridge release now uses SHA-256 checksums for improved security
- Updates `requires_product_versions` in the metadata to be less strict in order to allow JMX Bridge v1.9 to also work with the future PCF v1.12
- Stemcell for v1.9.2 remains v3363

Known issues

- None

v1.9.1

Release Date: June 15, 2017

Major Features

- JMX Bridge v1.9 contains two major architecture changes in support of overall security improvements. No breaking changes in the metrics output or format have been introduced in this effort, so consumption of both BOSH and platform metrics should continue to be seamless for end-users upon upgrading from JMX Bridge v1.8 to JMX Bridge v1.9.
  - The former Firehose consumer, `opentsdb-firehose-nozzle`, has been updated to a new, gRPC nozzle `jmx-firehose-nozzle`
  - BOSH metric data now flows from Ops Manager to JMX Bridge via the new `jmx-bosh-plugin`. The related installation field in Ops Manager has been renamed from "Metrics IP" to "JMX Provider IP Address", and it automatically configures the new plugin. Ops Manager will migrate IP entries in

the “Metrics IP” field to the “JMX Provider IP Address” field upon upgrade from PCF v1.10 to PCF v1.11

- New optional configuration to “Disable or Enable the Nozzle Prefix”; enabled by default
    - The behavior of the JMX Bridge to prepend `opentsdb.nozzle` onto all non-BOSH Firehose-transmitted metrics outputted was an artifact of the prior Firehose nozzle. As this was replaced, this artifact was also eliminated. In order to avoid breaking changes to consumers of prior JMX Bridge versions, a new optional feature, enabled by default, was created to continue to prepend this `opentsdb.nozzle` value onto metrics previously outputted with it
    - If you do not require backward compatibility, you can make the metrics more readable by clearing the **Enable Nozzle Prefix** checkbox. This omits the `opentsdb.nozzle` prefix, transmitting the metric values in their pcf-emitted name format.
    - This backwards-compatibility feature may be deprecated in future versions of JMX Bridge. It is strongly recommended that consumer name mappings be updated when possible, and this feature to prepend `opentsdb.nozzle` then disabled.
- Support for Firehose transmission of tagged metrics
    - The Firehose is capable of transmitting metrics with tags. While few emitted platform metrics are currently using this capability, for those that are, JMX Bridge v1.9 will pass the tag data along in the same format as seen when using `cf nozzle`
    - An example of a tagged metric is `DopplerServer.listeners.receivedEnvelopes`. When viewing this metric via JMX Bridge or `cf nozzle` the transmitted format is `DopplerServer.listeners.receivedEnvelopes[event_type=ContainerMetric,protocol=grpc]`

Release Notes

- JMX Bridge v1.9.1 is targeted for PCF v1.11.x
    - JMX Bridge users upgrading to PCF v1.11, must also upgrade to JMX Bridge v1.9 due to a architecture-driven dependency between Ops Manager v1.11 and JMX Bridge v1.9
- Stemcell for v1.9.1 is v3363

Known issues

- Installing JMX Bridge v1.9 on Ops Manager v1.11.0 impacts BOSH metrics output as follows:
    - `ID` information does not come through JMX Bridge
    - `IP Property` comes through as a blank string instead of `null` in JMX Bridge

  If you want to consume BOSH metrics from PCF v1.11 using JMX Bridge v1.9, you must use Ops Manager v1.11.3 or later.

Past Minor v1.8.x

Release notes for v1.8.x releases can be found here.

View the source for this page in GitHub