

Pivotal™

PRODUCT DOCUMENTATION

PCF Metrics®

Version 1.2

User's Guide

© Copyright Pivotal Software Inc, 2013-2018

Table of Contents

Table of Contents	2
PCF Metrics	3
Installing PCF Metrics	4
Sizing PCF Metrics For Your System	7
Using PCF Metrics	11
Troubleshooting PCF Metrics	15
PCF Metrics Product Architecture	27
PCF Metrics Release Notes and Known Issues	32

PCF Metrics

Pivotal Cloud Foundry (PCF) Metrics stores logs, metrics data, and event data from applications running on PCF for the past two weeks. It graphically presents this data to help operators and developers better understand the health and performance of their apps. PCF Metrics includes the following data views:

- **Container metrics:** CPU, memory, and disk usage percentages
- **Network metrics:** requests per second, HTTP errors per second, and request latency
- **App events:** create, update, start, stop, and crash
- **Logs:** searchable application logs

For example, if you see a crash in the events view, you can zoom in on that time and view the corresponding container and network metrics. To help you troubleshoot, you can also see log output for that same time period.

Product Snapshot

Current PCF Metrics Details

Version: v1.2

Release Date: November 29, 2016

Compatible Ops Manager Version(s) for Install: v1.8.0 through v1.9.x

Compatible Ops Manager Version(s) for Upgrade: v1.7.8 through v1.9.x

Compatible Elastic Runtime Version(s): v1.8.9 through v1.9.x

AWS support? Yes

Azure support? Yes

GCP support? Yes

OpenStack support? Yes

vSphere support? Yes

 **Note:** The PCF Metrics tile operates in lockstep with Pivotal Elastic Runtime.

If you are upgrading from PCF v1.7 to PCF v1.8 and you are using PCF Metrics v1.0.x, you must update to a PCF Metrics v1.2 tile before proceeding with the upgrade.

PCF Metrics User Guide

See the following topics for details about PCF Metrics:

- [Installing PCF Metrics](#)
- [Sizing PCF Metrics For Your System](#)
- [Using PCF Metrics](#)
- [Troubleshooting PCF Metrics](#)
- [PCF Metrics Product Architecture](#)
- [Release Notes and Known Issues](#)

Installing PCF Metrics

This document describes how to install and configure Pivotal Cloud Foundry (PCF) Metrics.

For information about the components deployed as part of this install procedure, see the [PCF Metrics Product Architecture](#) topic.

Prerequisites

- Ensure that you have installed the [Elastic Runtime Tile](#).
- Ensure that you have installed v1.6 or later of the [Redis tile](#).
- If you are running PCF on AWS, then ensure that, in Elastic Runtime, you have changed the **Loggregator Port** to `4443` from its value of `443`.
- If you are running PCF on Google Cloud Platform (GCP), then do the following to configure the DNS entries to accommodate web sockets:
 1. Log in to the GCP console.
 2. In the menu, navigate to the **Networking** tab and click **Load Balancing**.
 3. Find the load balancer that corresponds to `ENVIRONMENT-cf-ws`.
 4. Record the IP address.
 5. Click **Cloud DNS**, then click **ENVIRONMENT-zone**.
 6. Click **Add Record Set**.
 7. Enter a DNS name for mysql-logqueue. The DNS name should be `mysql-logqueue.SYSTEM_DOMAIN`. Refer to your ERT Tile's configuration of the System Domain under the Domains configuration section.
 8. In the IPv4 address field, enter the IP address of the load balancer that you recorded in Step 4.
 9. Leave the other fields as default.
- 10. Repeat Steps 6–9 twice to create DNS records for elasticsearch-logqueue (`elasticsearch-logqueue.SYSTEM_DOMAIN`) and metrics (`metrics.SYSTEM_DOMAIN`).

Step 1: Add the PCF Metrics Tile to Ops Manager

 **Note:** PCF Metrics should be installed on the same network as the Elastic Runtime Tile.

1. Download the PCF Metrics file from [Pivotal Network](#).
2. Upload the PCF Metrics file to your Ops Manager installation.
3. Click **Add** next to the uploaded product description in the **Available Products** view to add PCF Metrics to your **Installation Dashboard**.

Step 2: Configure the PCF Metrics Tile

 **Note:** The following procedures offer a standard configuration. To customize PCF Metrics for high capacity, see the [Sizing PCF Metrics For Your System](#) topic.

From the **Installation Dashboard**, click the **PCF Metrics** tile.

Assign Availability Zones (AZs) and Networks.

1. Click **Assign AZs and Networks**.
2. Select an Availability Zone under **Place singleton jobs**.
Ops Manager runs Metrics jobs with a single instance in this Availability Zone.
3. Select one or more Availability Zones under **Balance other jobs**.
Ops Manager balances instances of Metrics jobs with more than one instance across the Availability Zones that you specify.
4. Use the drop-down menu to select a network.

5. Click **Save**.

Data Services Ports

For reference, the following table shows the port associated with each data service.

Service	Port
Elasticsearch	9200
MySQL	3306

MySQL Alerts

1. Click **MySQL Alerts**.
2. Set the **Email** value. Alerts for issues storing metrics into the MySQL cluster will be sent to this email address.

Data Store

1. Click **Data Store**.
2. Review the **Elastic Search Heap Size** value. Elastic Search memory allocation for Heap use. Set to 50% of the memory allocated to the smallest of the Elasticsearch instances in Resource Config or 31GB, whichever is smaller. Use a unit of M for megabytes or G for gigabytes.
3. Review the **MySQL InnoDB Buffer Size** value. Number of bytes for MySQL to use with memory allocation of InnoDB buffer. Set to 80% of the memory allocated to the MySQL instances in Resource Config. This does not take a unit, please provide the value in bytes.
4. Review the **MySQL Logqueue Count** value. You can increase this instance count at any time to accommodate higher levels of inbound metrics traffic.
5. Review the **Elasticsearch Logqueue Count** value. You can increase this instance count at any time to accommodate higher levels of inbound log traffic.
6. Review the **Ingestor Count** value. You can increase this instance count at any time to accommodate higher levels of Loggregator Firehose traffic.
7. Click **Save**.

Errands

1. Click **Errands**.

Note: The PCF Metrics tile selects all **Post-Deploy Errands** by default. Pivotal recommends that you do not deselect any errands as doing so can cause issues with the deployment of the tile. However, you can deselect the **Remove Legacy PCF Metrics CF Resources** errand after deploying v1.2 of the tile.
2. Review the **Post-Deploy Errands** and **Pre-Delete Errands**:
 - If this is the first deployment of the tile, all **Post-Deploy Errands** must be selected.
 - If you deselect the **Remove PCF Metrics 1.2 CF Resources** checkbox, artifacts may remain after the PCF Metrics tile uninstalls.
 - The following list describes what the **Smoke tests** errand does. See the [Smoke Test Errors](#) section of the *Troubleshooting PCF Metrics* topic for information on resolving errors discovered by this errand.
 - Confirms that MySQL ingests metrics
 - Confirms that Elasticsearch ingests logs
 - Confirms that the APIs return metrics and logs

Resource Config

1. Click **Resource Config**.
2. Review the resource configurations. By default, the settings match the instance types that are best suited for each job. For reference, the following table shows the default resource and IP requirements for installing the PCF Metrics tile:

Resource	Instances	Persistent	CPU	RAM	Ephemeral	Static IP	Dynamic IP
Elasticsearch Master	3	10 GB	4	16 GB	32 GB	3	0
Elasticsearch Coordinator	2	1 GB	2	16 GB	32 GB	1	0
Elasticsearch Data	4	100 GB	2	16 GB	32 GB	4	0
MySQL Server	2	100 GB	2	16 GB	32 GB	3	0
MySQL Proxy	2	n/a	2	16 GB	32 GB	2	0
MySQL Monitor	1 (not configurable)	n/a	2	16 GB	32 GB	0	1
Metron	1 (not configurable)	n/a	2	4 GB	32 GB	1	0

If you expect a high level of use, you may need to increase the disk resources available to your instances.

 **Note:** There have been issues with the Ops Manager Bosh Director correctly partitioning persistent disks larger than 2 TB.

3. Click **Save**.

Stemcell

1. Navigate to [Pivotal Network](#) and click **Stemcells**.
2. Download the appropriate stemcell version for your IaaS.

 **Note:** On AWS make sure to use a HVM stemcell if you are using the default instance sizes.

3. Click **Import Stemcell** and select the stemcell file you downloaded.

Step 3: Deploy PCF Metrics

Click **Apply Changes** to install the service. If the smoke tests fail, see the [Troubleshoot Smoke Test Errors](#) section of the *Troubleshooting PCF Metrics* topic.

Review the [Using PCF Metrics](#) topic for more information on how to log in, use, and interpret data from PCF Metrics.

Sizing PCF Metrics For Your System

This topic describes how to configure Pivotal Cloud Foundry (PCF) Metrics for high availability. Operators can use these procedures to optimize PCF Metrics for high capacity.

For more information about PCF Metrics components, see the [PCF Metrics Product Architecture](#) topic.

Configuring the Metrics Datastore

PCF Metrics stores metrics in a MySQL cluster.

To customize PCF Metrics for high capacity, you can add memory and persistent disk to the MySQL server nodes.

Considerations for Scaling

Because apps emit logs at different volumes and frequencies, you should not scale the MySQL server nodes in accordance to the number of app instances in your deployment. Because of the ease in scaling these components, we recommend starting with a minimal configuration then evaluating performance over a period of time and scaling. As long as persistent disk is being scaled up, there should not be any fear of losing data.

To calculate approximate starting memory and disk allocation for each MySQL server node, use the following formulas:

- **Memory:** As a general rule, each MySQL server node should have a minimum of 35 GB of memory. Increase from here if read performance starts to decrease.
- **Persistent Disk:** As a general rule, 10 million metrics stored in MySQL consumes approximately 6.5 GB of persistent disk on the VM.

Use these results as guidelines. Consider configuring your MySQL server nodes with additional memory and disk if your deployment adds additional app instances.

Procedures for Scaling

After determining the amount of memory and persistent disk required for each MySQL server node, perform the following steps:

1. Navigate to the Ops Manager Installation Dashboard and click the **Metrics** tile.
2. From the **Settings** tab of the **Metrics** tile, click **Resource Config**.
3. Modify the memory limit or persistent disk allocation as needed for your environment.

⚠ warning: There have been issues with Ops Manager BOSH Director using persistent disks larger than 2TB.

4. If you modify the memory allocation for the MySQL server nodes, you must also update the MySQL InnoDB Buffer Size setting. Pivotal recommends that you set the buffer size to 80% of the memory allocated to that VM. To change the MySQL InnoDB Buffer Size:
 - a. Navigate to the Ops Manager Installation Dashboard and click the **Metrics** tile.
 - b. From the **Settings** tab of the **Metrics** tile, click **Data Store**.
 - c. Update the **MySQL InnoDB Buffer Size** input field.

Configuring the Log Datastore

PCF Metrics uses Elasticsearch to store logs. Each Elasticsearch node contains multiple shards of log data, divided by time slice. To customize PCF Metrics for high capacity, you can scale the number of Elasticsearch data nodes.

Considerations for Scaling

To determine the number of Elasticsearch data nodes required for PCF Metrics, consider how many logs the apps in your deployment emit and the average size of each log.

If your average log size is 1 kilobyte, and each node has 1 terabyte of available disk space, then each node has a maximum storage capacity of 1 billion log messages. If your apps emit 3 billion logs over a 24-hour period, you need at least 3 nodes to hold the data and 3 additional nodes for high-availability replication.

This example assumes that your apps emit logs at a continuous rate over 24 hours. However, apps typically do not emit logs continuously. If your apps emit 2 billion of the 3 billion logs between 8 AM and 4 PM, you must determine the minimum node-to-shard ratio to accommodate that rate over the 8-hour period. Because your apps emit 1 billion logs over a 4 hour span, you need at least 6 nodes (24 hours/6 nodes = 4 hours worth of shards per node) to hold the data and an additional 6 nodes for high-availability replication.

You can also use the throughput of logs per minute to help determine how many Elasticsearch data nodes to provision. As a general rule, provision one data node for every 5000 logs received in one minute.

Procedures for Scaling

A warning: If you modify the number of Elasticsearch instances, the Elasticsearch cluster temporarily enters an unhealthy period during which it does not ingest any new logs data, due to shard allocation.

After determining the number of Elasticsearch nodes needed for your deployment, perform the following steps to scale your nodes:

1. Navigate to the Ops Manager Installation Dashboard and click the **Metrics** tile.
2. From the **Settings** tab of the **Metrics** tile, click **Resource Config**.
3. Locate the **ElasticSearchData** job and select the dropdown menu under **Instances** to change the number of instances.



4. Click **Save**.

Configuring the Ingestor

PCF Metrics deploys the Ingestor as an app within PCF. The Ingestor consumes logs and metrics from the Loggregator Firehose, sending metrics and logs to their respective Logqueue apps. To customize PCF Metrics for high capacity, you can scale the number of Ingestor app instances and increase the amount of memory per instance.

Considerations for Scaling

Because apps emit logs at different volumes and frequencies, you should not scale the Ingestor by matching the number of Ingestor instances to the number of app instances in your deployment.

Because Ingestor performance is affected by Loggregator performance, it can be difficult to determine in advance the proper configuration. Because of the ease in scaling these components, we recommend starting with a minimal configuration then evaluating performance over a period of time and scaling.

The Ingestor app can handle relatively large loads. For high availability, you must have at least two instances of the Ingestor app running. If your deployment runs fewer than 2000 app instances, two instances of the Ingestor app are sufficient.

Procedures for Scaling

A warning: If you decrease the number of Ingestor instances, you may lose data currently being processed on the instances you eliminate.

After determining the number of Ingestor app instances needed for your deployment, perform the following steps to scale the Ingestor:

1. Target your Cloud Controller with the Cloud Foundry Command Line Interface (cf CLI). If you have not installed the cf CLI, see the [Installing the cf CLI](#) topic.

```
$ cf api api.YOUR-SYSTEM-DOMAIN
Setting api endpoint to api.YOUR-SYSTEM-DOMAIN...
OK
API endpoint: https://api.YOUR-SYSTEM-DOMAIN (API version: 2.54.0)
Not logged in. Use 'cf login' to log in.
```

2. Log in with your UAA administrator credentials. To retrieve these credentials, navigate to the **Pivotal Elastic Runtime** tile in the Ops Manager Installation Dashboard and click **Credentials**. Under **UAA**, click **Link to Credential** next to **Admin Credentials** and record the password.

```
$ cf login
API endpoint: https://api.YOUR-SYSTEM-DOMAIN

Email> admin
Password>
Authenticating...
OK
```

3. When prompted, target the `metrics` space.

```
Targeted org system

Select a space (or press enter to skip):
1. system
2. notifications-with-ui
3. autoscaling
4. metrics

Space> 4
Targeted space metrics

API endpoint: https://api.YOUR-SYSTEM-DOMAIN (API version: 2.54.0)
User: admin
Org: system
Space: metrics
```

4. Scale your Ingestor app to the desired number of instances:

```
$ cf scale metrics-ingestor -i INSTANCE-NUMBER
```

5. Evaluate the CPU and memory load on your Ingestor instances:

```
$ cf app metrics-ingestor
Showing health and status for app metrics-ingestor in org system / space metrics as admin...
OK

requested state: started
instances: 1/1
usage: 1G x 1 instances
urls:
last uploaded: Sat Apr 23 16:11:29 UTC 2016
stack: cflinuxfs2
buildpack: binary_buildpack

state      since          cpu    memory    disk      details
#0  running   2016-07-21 03:49:58 PM  2.9%  13.5M of 1G  12.9M of 1G
```

If your average memory usage exceeds 50% or your CPU consistently averages over 85%, add more instances with `cf scale metrics-ingestor -i INSTANCE-NUMBER`.

In general, you should scale the Ingestor app by adding additional instances. However, you can also scale the Ingestor app by increasing the amount of memory per instance:

```
$ cf scale metrics-ingestor -m NEW-MEMORY-LIMIT
```

For more information about scaling app instances, see the [Scaling an Application Using cf scale](#) topic.

Configuring the Logqueues

PCF Metrics deploys a MySQL Logqueue and an Elasticsearch Logqueue as apps within PCF. The MySQL logqueue consumes metrics from the Ingestor and forwards them to MySQL. The Elasticsearch logqueue consumes logs from the Ingestor and forwards them to Elasticsearch. To customize PCF Metrics for high capacity, you can scale the number of Logqueue app instances and increase the amount of memory per instance.

Considerations for Scaling

The number of MySQL and Elasticsearch logqueues needed is dependent on the frequency of logs and metrics being forwarded by the Ingestor. As a general rule, for every 45,000 logs per minute, add 2 Elasticsearch logqueues. For every 17,000 metrics per minute, add 1 MySQL Logqueue. This is a general estimate and you may need fewer instances depending on your deployment. To optimize resource allocation, provision fewer instances initially and increase instances until you achieve desired performance.

Procedures for Scaling

To modify your Elasticsearch Logqueue app instances, you must first target your Cloud Controller, log in with your UAA administrator credentials, and target the `metrics` space by following steps 1-3 in the [previous section](#).

To scale your Logqueue app instances, perform the following command:

```
$ cf scale elasticsearch-logqueue -i INSTANCE-NUMBER
```

To scale the memory limit per Logqueue app instance, perform the following command:

```
$ cf scale elasticsearch-logqueue -m NEW-MEMORY-LIMIT
```

To modify your MySQL Logqueue app instances, you must first target your Cloud Controller, log in with your UAA administrator credentials, and target the `metrics` space by following steps 1-3 in the [previous section](#).

To scale your Logqueue app instances, perform the following command:

```
$ cf scale mysql-logqueue -i INSTANCE-NUMBER
```

To scale the memory limit per Logqueue app instance, perform the following command:

```
$ cf scale mysql-logqueue -m NEW-MEMORY-LIMIT
```

⚠ warning: If you decrease the number of Logqueue instances, you may lose data currently being processed on the instances you eliminate.

Using PCF Metrics

This topic describes how to log in to, use, and interpret data from Pivotal Cloud Foundry (PCF) Metrics.

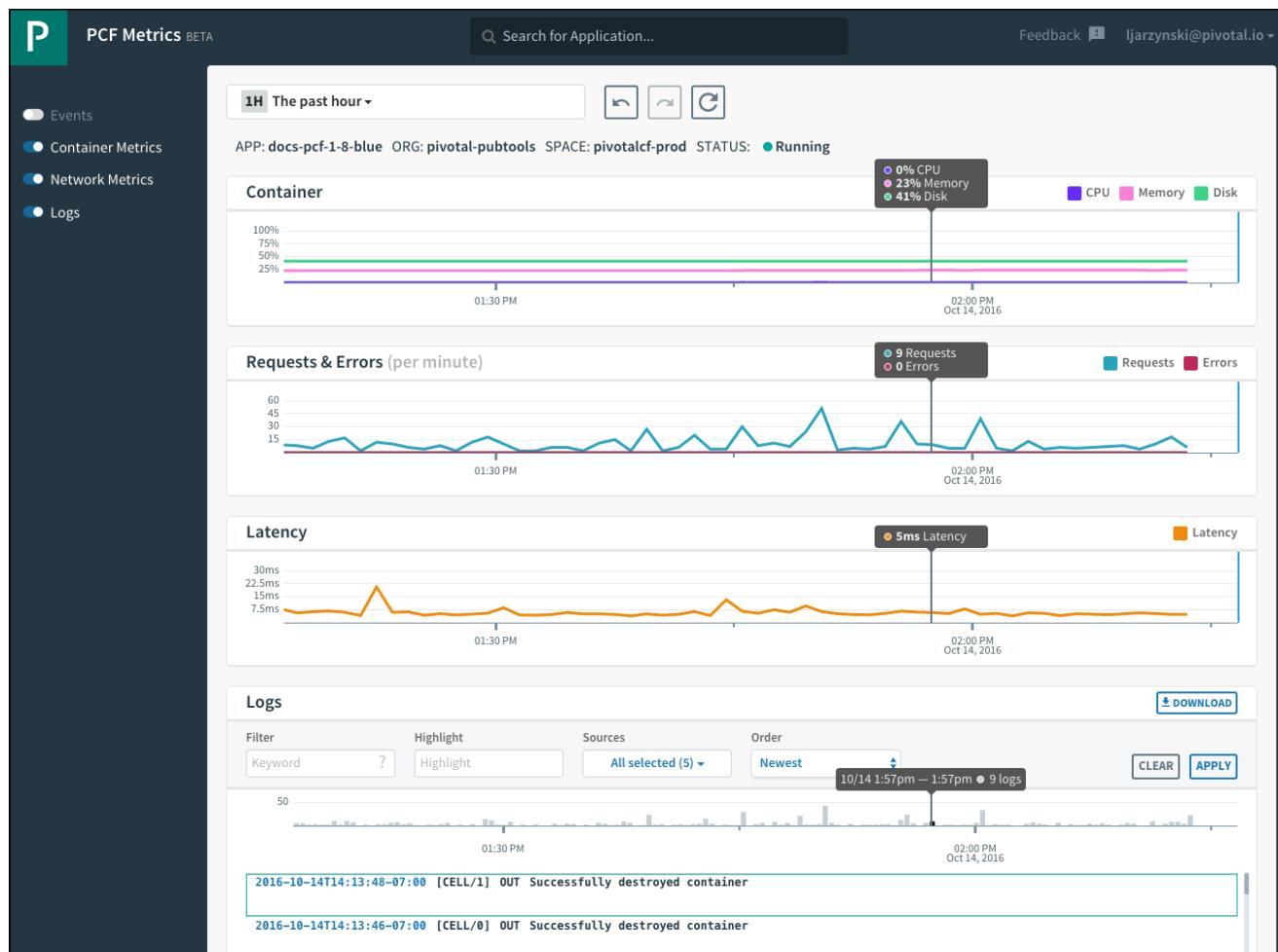
View an App in PCF Metrics

In a browser, navigate to `metrics.YOUR-SYSTEM-DOMAIN` and log in with your User Account and Authentication (UAA) credentials. Choose an app for which you want to view metrics. You can view any app for which you are a Space Developer.

The screenshot shows a search bar at the top with the placeholder "Search for Application...". Below it is a section titled "APPLICATIONS" containing three entries:

- docs-oss-pre-release (Status: cfcommunity > docs-prod)
- docs-oss-pre-release-blue (Status: cfcommunity > docs-staging)
- docs-oss-pre-release-green (Status: cfcommunity > docs-staging)

PCF Metrics provides you with a graphical representation of application data for a given time frame, along with the corresponding application logs. See the sections below to [Toggle the Data Views](#) and [Change the Time Frame](#). See the [Interpret Metrics](#) section for more information on each data view.



Toggle the Data Views

There are four data views: **Container Metrics**, **Network Metrics**, **Logs**, and **Events**. Select which views you want to see using the toggles in the

upper left.

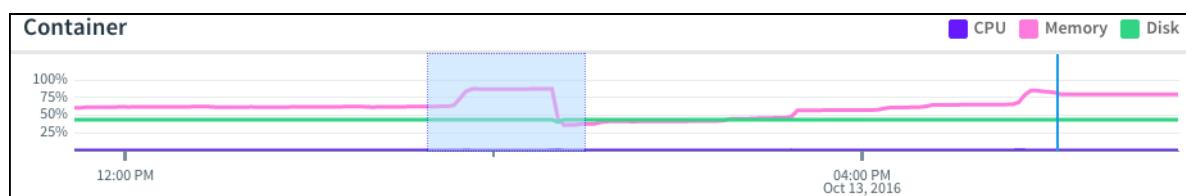
The screenshot shows the PCF Metrics BETA interface. On the left, there's a sidebar with a 'P' icon and four tabs: 'Events', 'Container Metrics', 'Network Metrics', and 'Logs'. The main area has a dropdown menu set to '1H The past hour'. Below it, it says 'APP: docs-pcfservices-green' and 'Events'. A single event entry is shown: 'crash'.

Change the Time Frame

The data views show time along the horizontal axis. You can change the time frame for all the data views by using the dropdown at the top of the page. Use the refresh button to update all views with the most current data.

This screenshot shows the top navigation bar of the PCF Metrics BETA interface. It includes a dropdown for selecting the time frame ('1H The past hour'), three refresh buttons (refresh, refresh with history, and a circular arrow), and status information ('tools SPACE: pivotalcf-staging STATUS: • Running'). Below the dropdown is a list of time frame options: '1H The past hour', '6H The past six hours', '1D The past day', '2D The past two days', '1W The past week', and '2W The past two weeks'. At the bottom right is a timestamp '04:30 PM'.

Zoom: From within any data view, you can also zoom in on areas of interest by clicking and dragging within the graph. This adjusts all of the views, including **Logs**, to show data from that time period.



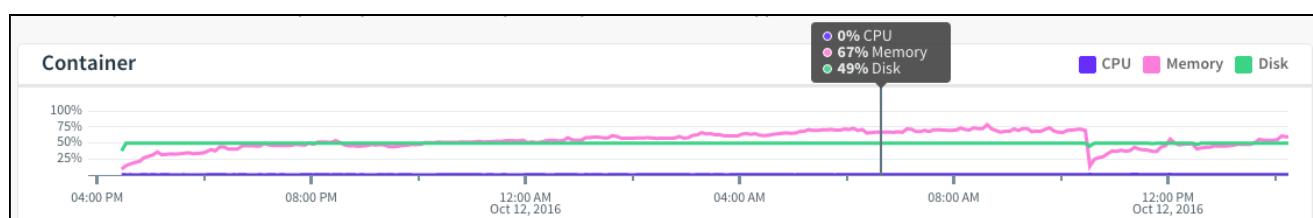
Drag: From underneath the x-axis of any data view, you can drag left or right to view data for an earlier or later time.

Interpret Metrics

PCF Metrics displays the following data. You can hover over each view to see data for a specific time.

Container Metrics

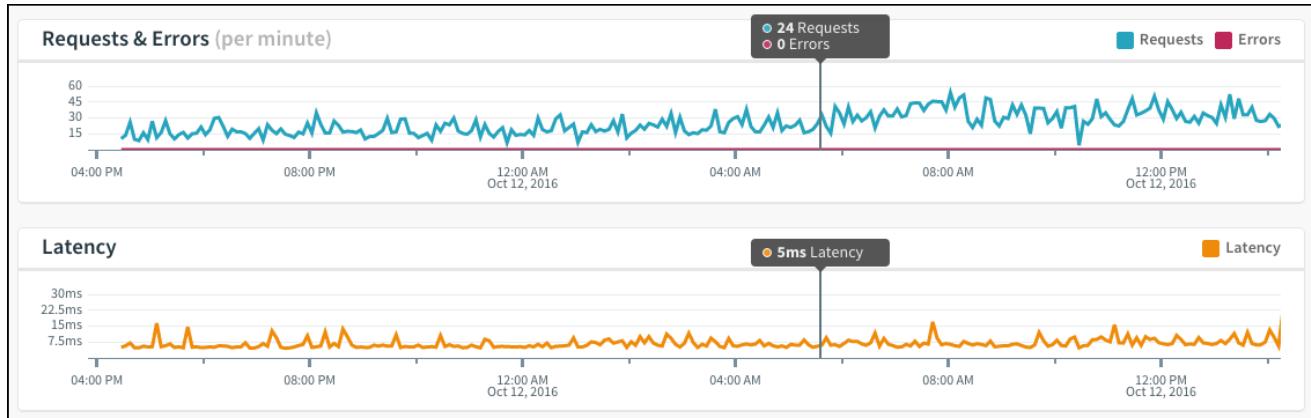
The **Container Metrics** view displays **CPU**, **Memory**, and **Disk** usage:



- A spike in **CPU** might point to a process that is computationally heavy. Scaling app instances can relieve the immediate pressure, but investigate the app to better understand and fix the root cause.
- A spike in **Memory** might mean a resource leak in the code. Scaling app memory can also relieve the immediate pressure, but look for and resolve the underlying issue so that it does not occur again.
- A spike in **Disk** might mean the app is writing logs to files instead of STDOUT, caching data to local disk, or serializing huge sessions to disk.

Network Metrics

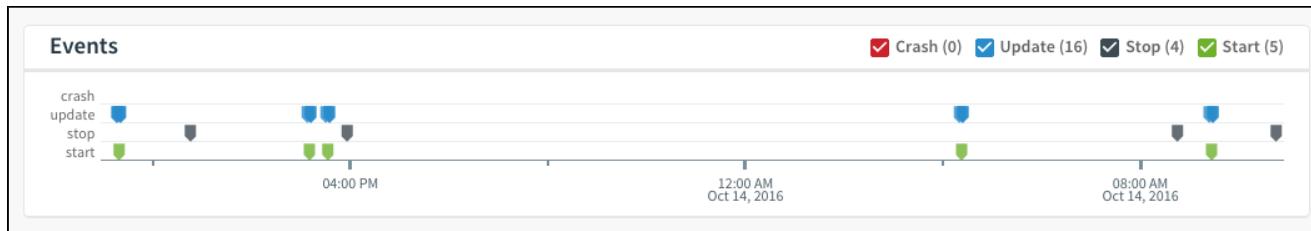
The **Network Metric** view displays HTTP **Requests & Errors** and **Latency**:



- A spike in HTTP **Requests** means more users are using your app. Scaling app instances can reduce the higher latency that may result.
- A spike in HTTP **Errors** means one or more 5xx errors have occurred. Check your app logs for more information.
- A spike in **Latency** means your users are waiting longer to use your app. Scaling app instances can spread that workload over more resources and result in faster response times.

Events

The **Events** view shows the following app events: **Create**, **Update**, **Start**, **Stop**, and **Crash**. You can change which events you see using the checkboxes in the upper right.



See the [About Starting Applications](#) and [Troubleshooting Application Deployment and Health](#) topics for more information about app events.

Logs

Logs displays application log data ingested from the [firehose](#). See the [Application Logging in Cloud Foundry](#) topic for more information.



PCF Metrics lists all logs by default, and the histogram displays log frequency for the current time frame. You can refine the list and histogram with the following:

- **Filter:** Perform a keyword search. The histogram updates with blue bars based on the filter. Hover over a histogram bar to view the amount of logs for a specific time based on your filter.
- **Highlight:** Enter a term to highlight within your search. The histogram updates with yellow bars based on the results. Hover over a histogram bar to view the amount of logs for a specific time that contain the highlighted term.
- **Sources:** Choose which sources to display logs from.
- **Order:** Modify the order in which logs appear.

You can also click the **Download** button to download a file containing logs for the current search.

Troubleshooting PCF Metrics

This topic describes how to resolve common issues experienced while operating or using Pivotal Cloud Foundry (PCF) Metrics.

Errors during Deployment

The following sections describe errors that cause failure during a PCF Metrics tile and how to troubleshoot them.

Smoke Test Errors

PCF Metrics runs a set of smoke tests during installation to confirm system health. If the smoke tests discover any errors, you can find a summary of those errors at the end of the errand log output, including detailed logs about where the failure occurred.

The following tables describe common failures and how to resolve them.

Insufficient Resources

Error	Insufficient Resources																														
Cause	<p>Your PCF deployment has insufficient Diego resources to handle the apps pushed as part of a PCF Metrics installation.</p> <p>The PCF Metrics tile deploys the following apps:</p> <table border="1"> <thead> <tr> <th>App</th><th>Memory</th><th>Disk</th></tr> </thead> <tbody> <tr> <td>metrics-ingestor*</td><td>512MB</td><td>1GB</td></tr> <tr> <td>mysql-logqueue*</td><td>1GB</td><td>1GB</td></tr> <tr> <td>elasticsearch-logqueue*</td><td>512MB</td><td>1GB</td></tr> <tr> <td>metrics-aggregator</td><td>256MB</td><td>1GB</td></tr> <tr> <td>metrics</td><td>1GB</td><td>1GB</td></tr> <tr> <td>worker-app-dev</td><td>1GB</td><td>1GB</td></tr> <tr> <td>worker-app-logs</td><td>1GB</td><td>1GB</td></tr> <tr> <td>worker-health-check</td><td>1GB</td><td>1GB</td></tr> <tr> <td>worker-reaper</td><td>1GB</td><td>1GB</td></tr> </tbody> </table> <p>*You may have more than one instance of each of the Ingestor and Loqueue apps depending your sizing needs. You configure these instance counts as part of the Data Store pane of the tile.</p>	App	Memory	Disk	metrics-ingestor*	512MB	1GB	mysql-logqueue*	1GB	1GB	elasticsearch-logqueue*	512MB	1GB	metrics-aggregator	256MB	1GB	metrics	1GB	1GB	worker-app-dev	1GB	1GB	worker-app-logs	1GB	1GB	worker-health-check	1GB	1GB	worker-reaper	1GB	1GB
App	Memory	Disk																													
metrics-ingestor*	512MB	1GB																													
mysql-logqueue*	1GB	1GB																													
elasticsearch-logqueue*	512MB	1GB																													
metrics-aggregator	256MB	1GB																													
metrics	1GB	1GB																													
worker-app-dev	1GB	1GB																													
worker-app-logs	1GB	1GB																													
worker-health-check	1GB	1GB																													
worker-reaper	1GB	1GB																													
Solution	<p>Increase the number of Diego cells so that your PCF deployment can support the apps pushed as part of the PCF Metrics installation:</p> <ol style="list-style-type: none"> 1. Navigate to the Resource Config section of the Elastic Runtime tile. 2. In the Diego Cell row, add another Instance. 																														

Nginx Load Balancer

Error	The Smoke tests for Metrics UI errand failed. Or, the Smoke tests for Metrics UI checkbox is not selected and installation was successful, but the UI keeps loading and the graphs do not populate with data.
Cause	<p>The Nginx <code>proxy_buffering</code> property is on and causes Nginx to block SSE traffic.</p> <ol style="list-style-type: none"> 1. From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment: <pre>\$ cf target -o system -s metrics-v1-2</pre>

Solution	<p>2. Confirm that Smoke tests for Metrics UI errand was not run during installation by listing recent logs from the <code>worker-app-logs</code> and <code>worker-app-dev</code> apps:</p> <pre>\$ cf logs --recent worker-app-logs \$ cf logs --recent worker-app-dev</pre> <p>If neither log contains the text <code>jobStarted</code>, then the jobs are not queued because Nginx is blocking SSEs.</p> <p>3. Turn off the Nginx <code>proxy_buffering</code> property.</p>
-----------------	--

Failed Querying MySQL

Error	<code>Failed querying mysql</code>
Cause	The tile deployed without the necessary errands selected to keep the internal database schema in sync with apps.
Solution	<p>Re-deploy the tile with the following errands selected:</p> <ul style="list-style-type: none"> • Database migrations for PCF Metrics • Push PCF Metrics Data components • Push PCF Metrics UI component

Received No Results Back from MySQL - Failing

Error	<code>Received no results back from mysql - failing</code>
Cause	The Ingestor is not functioning properly.
Solution	<p>1. From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment:</p> <pre>\$ cf target -o system -s metrics-v1-2</pre> <p>2. Run <code>cf apps</code> to see if these apps are running:</p> <ul style="list-style-type: none"> ◦ <code>metrics-ingestor</code> ◦ <code>mysql-logqueue</code> <p>3. If the apps are not running, run the following commands to start them:</p> <pre>\$ cf start metrics-ingestor \$ cf start mysql-logqueue</pre> <p>4. Run the following commands and search the app logs for <code>ERROR</code> messages containing additional information:</p> <pre>\$ cf logs metrics-ingestor --recent \$ cf logs mysql-logqueue --recent</pre> <p>Note: In some cases, the apps cannot communicate due to TLS certificate verification failure. If your deployment uses self-signed certs, ensure the Disable SSL certificate verification for this environment box is checked in the Elastic Runtime Networking pane.</p>

Failed to Connect to MySQL

Error	<code>Failed to connect to mysql</code>
Cause	MySQL is not running properly.

1. Check the logs of the MySQL Server and MySQL Proxy jobs for errors.
 - You can download the logs from the PCF Metrics tile under the **Status** tab.

Solution

- From the cf CLI, target the `system` org and `metrics-v1-2` space of your PCF deployment:

```
$ cf target -o system -s metrics-v1-2
```

- Run the following command and ensure the security group can access the MySQL jobs:

 **Note:** PCF Metrics creates a default security group to allow all traffic to its apps.

```
$ cf security-group metrics-api
```

Failed to start elasticsearch client

Error	<code>Failed to start elasticsearch client</code>
Cause	Elasticsearch is not running correctly.
Solution	<ol style="list-style-type: none"> Check the logs of the Elasticsearch Master, Elasticsearch Coordinator, and Elasticsearch Data jobs for errors. <ul style="list-style-type: none"> You can download the logs from the PCF Metrics tile under the Status tab. From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment:

```
$ cf target -o system -s metrics-v1-2
```

- Run the following command and ensure the security group can access the Elasticsearch jobs:

 **Note:** PCF Metrics creates a default security group to allow all traffic to its apps.

```
$ cf security-group metrics-api
```

Never Received App Logs

Error	<code>Never received app logs - something in the firehose -> elasticsearch flow is broken</code>
Cause	Ingestor is not inserting logs correctly.
Solution	<ol style="list-style-type: none"> From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment:

```
$ cf target -o system -s metrics-v1-2
```

- Run `cf apps` to see if these apps are running:

- `metrics-ingestor`
- `elasticsearch-logqueue`

- If the apps are not running, run the following commands to start them:

```
$ cf start metrics-ingestor  
$ cf start elasticsearch-logqueue
```

- Run the following commands and search the app logs for `ERROR` messages containing additional information:

```
$ cf logs metrics-ingestor --recent  
$ cf logs elasticsearch-logqueue --recent
```

 **Note:** In some cases, you may discover a failure to communicate with Loggregator in the form of a bad handshake error.

Ensure the **Loggregator Port** setting in the Elastic Runtime tile **Networking** pane is set to the correct value. For AWS, it is `4443`. For all other IaaSes, it is `443`.

Metrics and Events Not Available

Error	<p>Network metrics are not available.</p> <p>Container metrics are not available.</p> <p>App events are not available.</p>
Cause	PCF Metrics is misconfigured and the frontend API does not receive logs from MySQL.
Solution	<ol style="list-style-type: none"> From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment: <pre>\$ cf target -o system -s metrics-v1-2</pre> Run the following command to check the app logs and investigate the error: <pre>\$ cf logs metrics --recent</pre>

Logs and Histograms Not Available

Error	<p>Logs are not available.</p> <p>Histograms are not available.</p>
Cause	PCF Metrics is misconfigured and the frontend API does not receive logs from Elasticsearch.
Solution	<ol style="list-style-type: none"> From the cf CLI, target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment: <pre>\$ cf target -o system -s metrics-v1-2</pre> Run the following command to check the app logs and investigate the error: <pre>\$ cf logs metrics --recent</pre>

Elasticsearch Instance does not Start

Error	The Deployment fails because an Elasticsearch instance does not start.
Cause	The instance might not start because its configured heap size is greater than that of the VM that hosts it.
Solution	<ol style="list-style-type: none"> From the PCF Metrics tile in Ops Manager, select the Data Store settings pane. Record the value in the Elastic Search Heap Size field. Select the Resource Config pane and ensure the following jobs have RAM greater than or equal to the Elastic Search Heap Size <ul style="list-style-type: none"> Elasticsearch Master Elasticsearch Coordinator Elasticsearch Data If any of the jobs do not have enough memory, do one of the following: <ul style="list-style-type: none"> Give the job more RAM Lower the Elastic Search Heap Size

No Logs or Metrics in the UI

In some cases, the PCF Metrics UI might not display metrics and logs after successfully deploying.

Follow the steps in this section to help locate the app or component causing the problem.

Step 1: Check your Load Balancer Configuration

If you use a load balancer, the event-stream mechanism used by the Metrics UI might be blocked. Refer to the table below to resolve this error.

If you do not use a load balancer, or this issue does not apply to your deployment, proceed to [Step 2: Check the PCF Metrics Apps](#).

Error	In the case of a customer using an F5 load balancer, metrics and logs were not visible in the UI despite successful ingestion and no UI errors reported.
Cause	The root of the issue was the traffic of type text/event-stream was blocked by the F5 load balancer.
Solution	When F5 was configured to allow event-stream traffic, the issue was resolved.

Step 2: Check the PCF Metrics Apps

1. From Ops Manager, click the Elastic Runtime Tile.
 - a. Click the **Credentials** tab.
 - b. Under the **UAA** job, next to **Admin Credentials**, click **Link to Credential**.
 - c. Record the username and password for use in the next step.

2. Log in to the Cloud Foundry Command Line Interface (cf CLI) using the credentials from the previous step.

```
$ cf login -a https://api.YOUR-SYSTEM-DOMAIN -u admin -p PASSWORD
```

3. When prompted, select the `system` org and the `metrics-v1-2` space.

4. Ensure that the output displays the following apps, each in a `started` state:

- o `metrics-ingestor`
- o `mysql-logqueue`
- o `elasticsearch-logqueue`
- o `metrics-aggregator`
- o `metrics`
- o `worker-app-dev`
- o `worker-app-logs`
- o `worker-health-check`
- o `worker-reaper`

5. Check the logs of each app for errors using the following command:

```
$ cf logs APP-NAME --recent
```

If you do not see any output, or if you did not find any errors, proceed to [Step 3: Check the Elasticsearch Cluster](#).

Step 3: Check the Elasticsearch Cluster

1. From Ops Manager, select the PCF Metrics tile.
2. Under the **Status** tab, record the IP of an **Elasticsearch Master** node.
3. Use `bosh ssh` to access the VM from the previous step. See the [Advanced Troubleshooting with the BOSH CLI](#) topic for instructions.
4. Run the following command to list all the Elasticsearch indices:

```
$ curl localhost:9200/_cat/indices?v | sort

green open app_logs_1477512000 8 1 125459066      0  59.6gb   29.8gb
green open app_logs_1477526400 8 1 129356671      0  59.1gb   29.5gb
green open app_logs_1478174400 8 1 129747170      0  61.9gb   30.9gb
...
green open app_logs_1478707200 8 1 128392686      0  63.2gb   31.6gb
green open app_logs_1478721600 8 1 102005754      0  53.5gb   26.5gb
health status index    pri rep docs.count docs.deleted store.size pri.store.size
```

- a. If the curl does not return a `success` response, Elasticsearch might not even be running correctly. Inspect the following logs for any failures or errors:

- `/var/vcap/sys/log/elasticsearch/elasticsearch.stdout.log`
- `/var/vcap/sys/log/elasticsearch/elasticsearch.stderr.log`

5. Examine the `status` column of the output.

- a. If the status of any of the indices is not `green`, restart the Logqueue app:

```
$ cf restart elasticsearch-logqueue
```

- b. Run the curl command periodically to see if the indices recover to a `green` status.

6. Run the curl command several more times and examine the most recent index to see if the number of stored documents periodically increases.

 **Note:** The last row of the output corresponds to the most recent index. The sixth column displays the number of documents for the index.

- a. If all indices show a `green` status, but the number of documents does not increase, there is likely a problem further up in ingestion. Proceed to to [Step 4: Check the Elasticsearch Logqueue](#).

Step 4: Check the Elasticsearch Logqueue

1. Run `cf apps` to see if the `elasticsearch-logqueue` app instances are `started`.

2. If any instance of the app is `stopped`, run the following command to increase logging:

```
$ cf set-env elasticsearch-logqueue LOG_LEVEL DEBUG
```

- a. Run the following command to stream logs:

```
$ cf logs elasticsearch-logqueue
```

- b. In a different terminal window, run the following command:

```
$ cf restage elasticsearch-logqueue
```

- c. Watch the logs emitted by the `elasticsearch-logqueue` app for errors.

- A common error is that the app cannot connect to Elasticsearch because a user deleted the application security group (ASG) that PCF Metrics creates to allow the Logqueue app to connect to the Elasticsearch VMs. You can run `cf security-group metrics-api` to see if the ASG exists. If not, see the documentation on [Creating Application Security Groups](#).

3. If the app is started and you do not find any errors, proceed to [Step 5: Check the Metrics Ingestor](#).

Step 5: Check the Metrics Ingestor

1. Run `cf apps` to see if the `metrics-ingestor` app instances are `started`.

2. If any of the app instances are `stopped`, run the following command to increase logging:

```
$ cf set-env metrics-ingestor LOG_LEVEL DEBUG
```

- a. Run the following command to stream logs:

```
$ cf logs metrics-ingestor
```

- b. In a different terminal window, run the following command:

```
$ cf restart metrics-ingestor
```

- c. Watch the logs emitted by the `metrics-ingestor` app for errors. Refer to the list below for common errors:

- **Cannot connect to the firehose**: PCF Metrics creates a UAA user to authenticate the connection to the firehose. This user must have the `doppler.firehose` authority.
- **Cannot connect to the logqueues**: There might be a problem with the UAA, or it could be throttling traffic.
- **WebSocket Disconnects**: If you see WebSocket disconnects logs in the Ingestor app, consider adding additional Ingestor instances. The Firehose may be dropping the Ingestor connection to avoid back pressure.

3. If the app is started and you do not find any errors, proceed to [Step 6: Check MySQL](#).

Step 6: Check MySQL

1. From Ops Manager, select the PCF Metrics tile.
2. Under the **Status** tab, record the IP of a **MySQL Server** node.
3. Use `bosh ssh` to access the VM from the previous step. See the [Advanced Troubleshooting with the BOSH CLI](#) topic for instructions.
4. Log in to mysql by running `mysql -u USERNAME -p PASSWORD`

 **Note:** If you do not know the username and password, you can run `cf env mysql-logqueue` with the `system` org and the `metrics-v1-2` space targeted.

5. Verify that the database was bootstrapped correctly:

- a. Run `show databases` and check for a `metrics` database.
 - i. If there is no `metrics` database, the `migrate_db` errand of the BOSH release might not have run or succeeded.

6. Run `use metrics` to select the `metrics` database:

```
mysql> use metrics;
```

7. Run `show tables` and ensure you see the following tables:

Tables_in_metrics
app_event
app_metric_rollup
container_metric_1477353600
container_metric_1477440000
container_metric_1477526400
container_metric_1477612800
container_metric_1477699200
container_metric_1477785600
container_metric_1477872000
container_metric_1477958400
container_metric_1478044800
container_metric_1478131200
container_metric_1478217600
container_metric_1478304000
container_metric_1478390400
container_metric_1478476800
container_metric_1478563200
container_metric_1478649600
http_start_stop_1477353600
http_start_stop_1477440000
http_start_stop_1477526400
http_start_stop_1477612800
http_start_stop_1477699200
http_start_stop_1477785600
http_start_stop_1477872000
http_start_stop_1477958400
http_start_stop_1478044800
http_start_stop_1478131200
http_start_stop_1478217600
http_start_stop_1478304000
http_start_stop_1478390400
http_start_stop_1478476800
http_start_stop_1478563200
http_start_stop_1478649600
schema_version

8. Enter the following query several times to verify that the value returned does not decrease over time:

```
mysql> select count(*) from metrics.app_metric_rollup where timestamp > ((UNIX_TIMESTAMP() - 60) * POW(10, 3));
```

This command displays the rate at which metrics flow in over the last minute.

- a. If the command returns `0` or a consistently decreasing value, the problem is likely further up in ingestion. Proceed to [Step 7: Check the MySQL Logqueue](#).

Step 7: Check the MySQL Logqueue

1. Run `cf apps` to see if the `mysql-logqueue` app instances are `started`.

2. If any instance of the app is `stopped`, run the following command to increase logging:

```
$ cf set-env mysql-logqueue LOG_LEVEL DEBUG
```

- a. Run the following command to stream logs:

```
$ cf logs mysql-logqueue
```

- b. In a different terminal window, run the following command:

```
$ cf restart mysql-logqueue
```

- c. Watch the logs emitted by the `mysql-logqueue` app for errors.

- A common error is that the app cannot connect to MySQL because a user deleted the application security group (ASG) that PCF Metrics creates to allow the Logqueue app to connect to the MySQL VMs. You can run `cf security-group metrics-api` to see if the ASG exists. If not, see the documentation on [Creating Application Security Groups](#).

3. If the app is started and you do not find any errors, proceed to [Step 8: Check the Metrics Aggregator](#).

Step 8: Check the Metrics Aggregator

1. Run `cf apps` to see if the `metrics-aggregator` app instances are `started`.
2. If any instance of the app is `stopped`, run the following command to increase logging:

```
$ cf set-env metrics-aggregator LOG_LEVEL DEBUG
```

- a. Run the following command to stream logs:

```
$ cf logs metrics-aggregator
```

- b. In a different terminal window, run the following command:

```
$ cf restart metrics-aggregator
```

- c. Watch the logs emitted by the `metrics-aggregator` app for errors.

- A common error is that the app cannot connect to MySQL because a user deleted the application security group (ASG) that PCF Metrics creates to allow the aggregator app to connect to the MySQL VMs. You can run `cf security-group metrics-api` to see if the ASG exists. If not, see the documentation on [Creating Application Security Groups](#).

MySQL Node Failure

In some cases, a MySQL server node may fail to restart. The following two sections describe the known conditions that cause this failure as well as steps for diagnosing and resolving them. If neither of the causes listed apply, the final section provides instructions for re-deploying BOSH as a last resort to resolve the issue.

Cause 1: Monit Timed Out

Diagnose

Follow these steps to see if a `monit` time-out caused the MySQL node restart to fail:

1. Use `bosh ssh` to access the failing node, using the IP address in the Ops Manager Director tile **Status** tab. See the [Advanced Troubleshooting with the BOSH CLI](#) topic for instructions.
2. Run `monit summary` and check the status of the `mariadb_ctrl` job.
3. If the status of the `mariadb_ctrl` job is `Execution Failed`, open the following file: `/var/vcap/sys/log/mysql/mariadb_ctrl.combined.log`.
 - a. If the last line of the log indicates that MySQL started without issue, such as in the example below, `monit` likely timed out while waiting for the job to report healthy. Follow the steps below to resolve the issue.

```
{"timestamp":1481149250.288255692,"source":"/var/vcap/packages/mariadb_ctrl/bin/mariadb_ctrl","message":"/var/vcap/packages/mariadb_ctrl/bin/mariadb_ctrl.mariadb_ctrl started","log_level":1,"data":{}}
```

Resolve

Run the following commands to return the `mariadb_ctrl` job to a healthy state:

1. Run `monit unmonitor mariadb`.
2. Run `monit monitor mariadb`.
3. Run `monit summary` and confirm that the output lists `mariadb_ctrl` as `running`.

Cause 2: Bin Logs Filled up the Disk

Diagnose

1. Use `bosh ssh` to access the failing node. See the [Advanced Troubleshooting with the BOSH CLI](#) topic for instructions.
2. Open the following log file: `/var/vcap/sys/log/mysql/mysql.err.log`.
3. If you see log messages that indicate insufficient disk space, the [persistent disk](#) is likely storing too many bin logs. Confirm insufficient disk space by doing the following:
 - a. Run `df -h`.
 - i. Ensure that you see the `/var/vcap/store` folder is at or over `90%` usage.
 - b. Navigate to `/var/vcap/store/mysql` and run `ls -al`.
 - i. Ensure that you see many files named with the format `mysql-bin.#####`.

In MySQL for PCF, the server node does not make use of these logs and you can remove all except the most recent bin log. Follow the steps below to resolve the issue.

Resolve

1. Log in to mysql by running `mysql -u USERNAME -p PASSWORD`

 **Note:** If you do not know the username and password, you can run `cf env mysql-logqueue` with the `system` org and the `metrics-v1-2` space targeted.

2. Run `use metrics;`.

3. Run the following command:

```
mysql> PURGE BINARY LOGS BEFORE 'YYYY-MM-DD HH:MM:SS';
```

Re-deploy BOSH to Restart the Node

If troubleshooting based on the causes mentioned above did not resolve the issue with your failing MySQL node, you can follow the steps below to recover it. Pivotal recommends only using this procedure as a if there are no other potential solutions available.

 **warning:** This procedure is extremely costly in terms of time and network resources. The cluster takes a significant amount of time to put the data replicated to the rest of the cluster back into the rebuilt node. This procedure consumes considerable network bandwidth as potentially hundreds of gigabytes of data needs to transfer.

Stop the Ingestor App

1. From Ops Manager, click the Elastic Runtime Tile.
 - a. Click the **Credentials** tab.
 - b. Under the **UAA** job, next to **Admin Credentials**, click **Link to Credential**.
 - c. Record the username and password for use in the next step.
2. Log in to the cf CLI using the credentials from the previous step.

```
$ cf login -a https://api.YOUR-SYSTEM-DOMAIN -u admin -p PASSWORD
```

3. Target the `system` org and `metrics-v1-2` space of your PCF deployment:

```
$ cf target -o system -s metrics-v1-2
```

4. Stop data flow into the Galera cluster:

```
$ cf stop metrics-ingestor
```

Edit Your Deployment Manifest

1. Follow the steps in the [Log in to BOSH](#) section of the Advanced Troubleshooting with the BOSH CLI topic to target and log in to your BOSH Director. The steps vary slightly depending on whether your PCF deployment uses internal authentication or an external user store.
2. Download the manifest of your PCF deployment:

```
$ bosh download manifest YOUR-PCF-DEPLOYMENT YOUR-PCF-MANIFEST.yml
```

Note: You must know the name of your PCF deployment to download the manifest. To retrieve it, run `bosh deployments` to list your deployments and locate the name of your PCF deployment.

3. Open the manifest and set the number of instances of the failed server node to `0`.
4. Run `bosh deployment YOUR-PCF-MANIFEST.yml` to specify your edited manifest.
5. Run `bosh deploy` to deploy with your manifest.
6. Run `bosh disks --orphaned` to see the [persistent disk](#) or disks associated with the failed node.
 - a. Record the `CID` of each persistent disk.
 - b. Contact [Pivotal Support](#) to walk through re-attaching the orphaned disks to new VMs to preserve their data.
7. Open the manifest and set the number of instances of the failed server node to `1`.
8. Run `bosh deploy` to deploy with your edited manifest.
9. Wait for BOSH to rebuild the node.

MySQL SST Disabled Error

If you see the message below on a failing node in `/var/vcap/sys/log/mysql/mysql.err.log`, you can resolve the error by following the instructions in the [Interruptor Logs](#) section of the MySQL for PCF documentation.

```
WSREP_SST: [ERROR] ##### (20160610 04:33:21.338)
WSREP_SST: [ERROR] SST disabled due to danger of data loss. Verify data and bootstrap the cluster (20160610 04:33:21.340)
WSREP_SST: [ERROR] ##### (20160610 04:33:21.341)
```

Log Errors

Error	The PCF Metrics UI does not show any new logs from Elasticsearch.
Cause	The tile deployed with the <code>Push PCF Metrics Data Components</code> errand deselected
Solution	<p>Restart the Elasticsearch Logqueue using the cf CLI as follows:</p> <ol style="list-style-type: none"> 1. Target the <code>system</code> org and <code>metrics-v1-2</code> space of your PCF deployment: <pre>\$ cf target -o system -s metrics-v1-2</pre> <ol style="list-style-type: none"> 2. Run the following command to restart the Logqueue application: <pre>\$ cf restart elasticsearch-logqueue</pre> <p>Note: To avoid having to apply this fix in the future, select the checkbox to enable the <code>Push PCF Metrics Data Components</code> <code>errand</code> before your next tile update.</p>

503 Errors

Error	You encounter <code>503</code> errors when accessing the PCF Metrics UI in your browser.				
Cause	Your Elasticsearch nodes may have become unresponsive.				
Solution	<p>Check the Elasticsearch index health by following the procedure below, and consider adding additional Elasticsearch nodes.</p> <ol style="list-style-type: none"> 1. Retrieve the IP address of your Elasticsearch master node by navigating to the Metrics tile in the Ops Manager Installation Dashboard, clicking the Status tab, and recording the IP address next to ElasticSearchMaster. <table border="1" style="margin-left: 20px;"> <tr> <td>ElasticSearchMaster</td> <td>0</td> <td>10.0.16.53</td> <td>i-63548dfa</td> </tr> </table> 2. SSH into the Ops Manager VM by following the instructions in SSH into Ops Manager. 3. From the Ops Manager VM, use <code>curl</code> to target the IP address of your Elasticsearch master node. Follow the instructions in the Cluster Health topic of the Elasticsearch documentation. 	ElasticSearchMaster	0	10.0.16.53	i-63548dfa
ElasticSearchMaster	0	10.0.16.53	i-63548dfa		

Fail to fetch Apps

Error	The metrics app fails to fetch apps for a user with the proper credentials.
Cause	The client's browser's plugins/cookies inject extraneous content in requests to CAPI, causing CAPI to reject the request
Verifying the error	<p>You can verify having this error by using either of the two methods:</p> <ul style="list-style-type: none"> - Examining the network requests to cloud controller in a browser, and see if there are any out of the ordinary content being sent with the request (<code>/cc/apps</code>) **Or** - Use a browser's incognito mode to see if the metrics app is able to fetch apps
Solution	Clear browser cookies/plugins

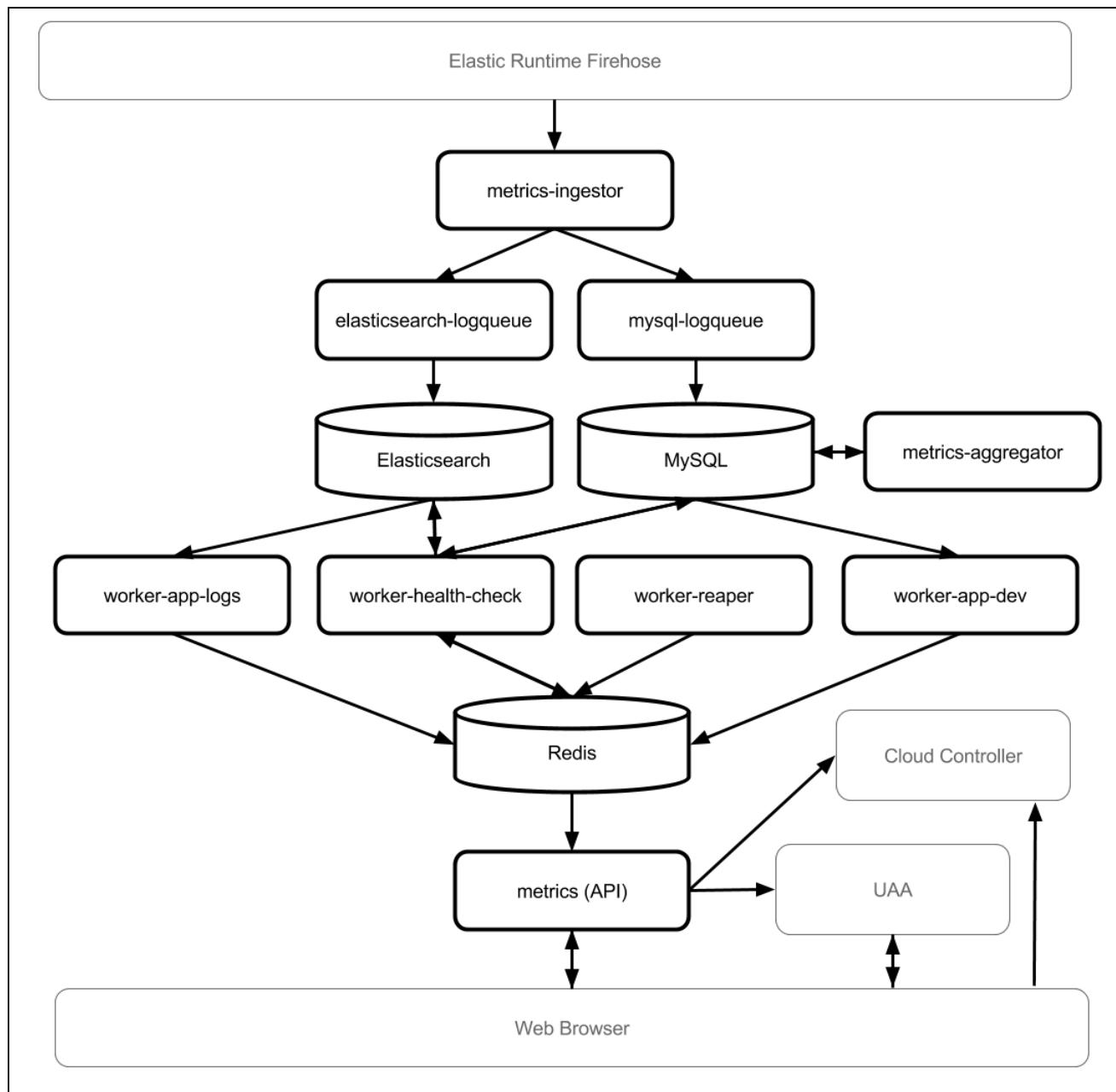
PCF Metrics Product Architecture

This topic describes the product architecture of Pivotal Cloud Foundry (PCF) Metrics.

Overview

The diagram below displays the components of PCF Metrics in bold, as well as the Cloud Foundry components that the PCF Metrics system interacts with.

PCF Metrics deploys several Cloud Foundry apps as part of the install process. These components are the bold rectangles in the diagram. The cylinders represent the data storage components of PCF Metrics.



See the following sections to understand the several processes that happen within the PCF Metrics system.

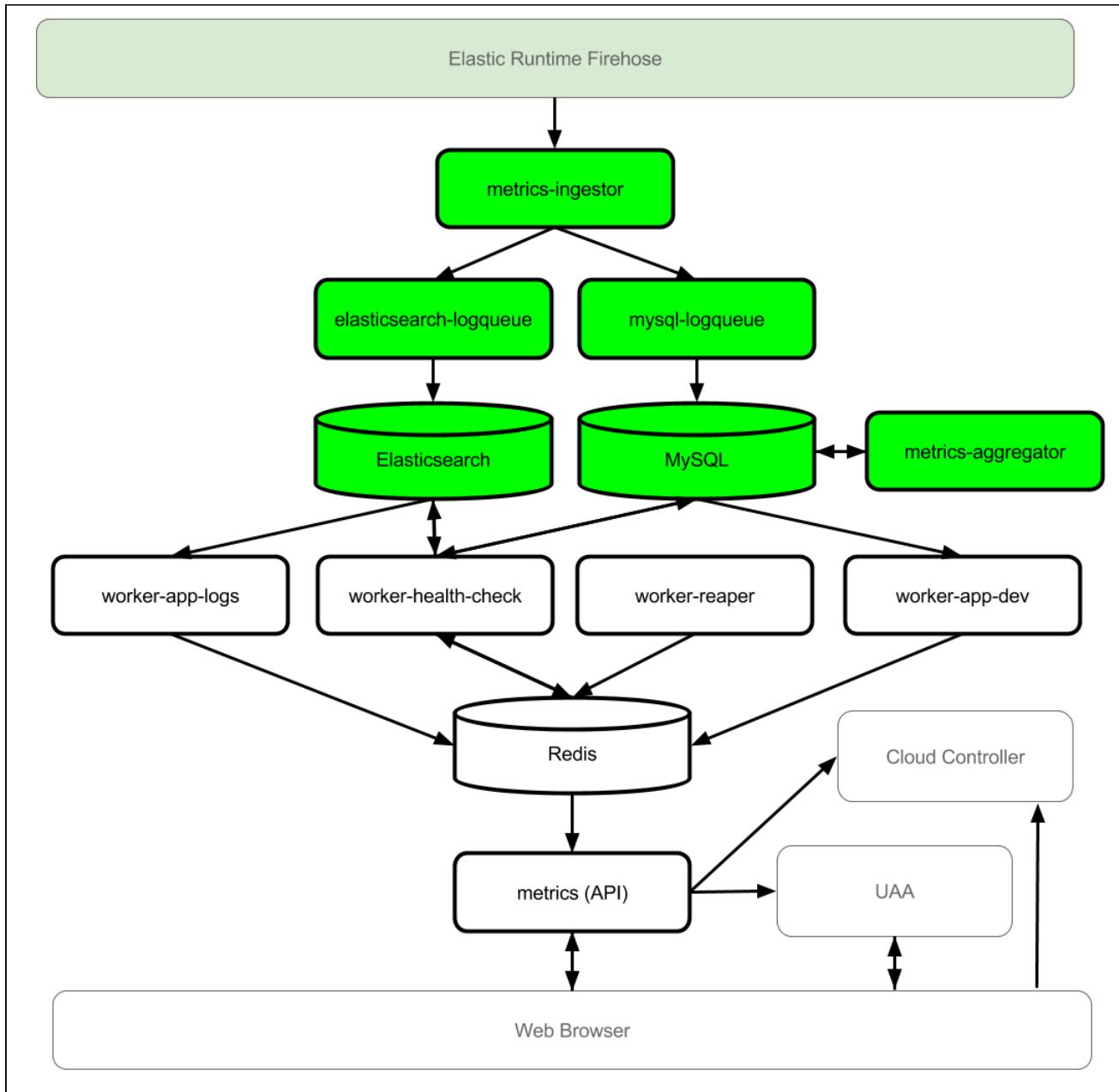
How Data Flows from the Firehose to the Datastores

This section describes how PCF Metrics fills its datastores. PCF Metrics uses two datastores:

- The MySQL component stores metric and event data from the apps running on your PCF deployment.
 - Examples of events are `start` and `stop`.
 - Examples of metrics are *container metrics* such as CPU and *network metrics* such as Requests.
 - The Elasticsearch component stores logs from the apps running on your PCF deployment.

Components

The diagram below highlights the components involved in the process of getting metric and log data into the Elasticsearch and MySQL datastores.



Process

The following table describes how the components act during each stage.

Stage	Description
1	The <code>metrics-ingestor</code> app does the following: <ul style="list-style-type: none">• Receives app logs from the Firehose and forwards them to both the <code>elasticsearch-logqueue</code> and <code>mysql-logqueue</code> apps

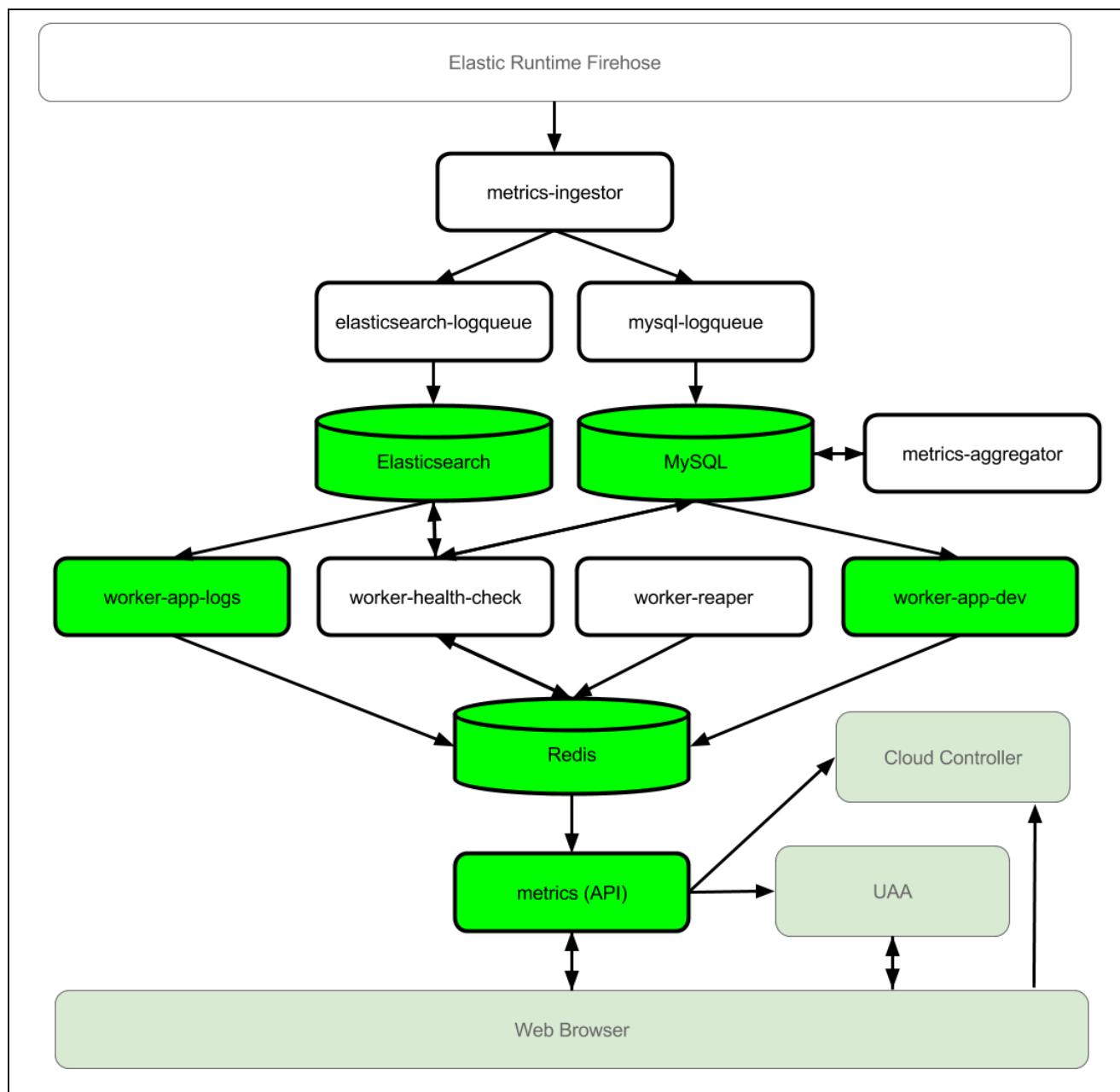
	<ul style="list-style-type: none"> • Receives container metrics and network metrics (HTTPStartStop events) from the Firehose and forwards them to the <code>mysql-logqueue</code> app
2	<p>Each of the logqueues act independently, writing information to the datastores:</p> <p>Elasticsearch logqueue</p> <p>The <code>elasticsearch-logqueue</code> app buffers logs and writes them to the Elasticsearch datastore.</p> <p>MySQL logqueue</p> <p>The <code>mysql-logqueue</code> app buffers logs and writes each data type to MySQL as follows:</p> <ul style="list-style-type: none"> • Container metrics: Inserts messages into the <code>container_metric</code> table of MySQL • Network: Inserts messages into the <code>http_start_stop</code> table of MySQL • App logs: Parses log messages for an app event name and inserts the message into the <code>app_event</code> table of MySQL
3	<p>The <code>metrics-aggregator</code> app, which runs according to an <code>AGGREGATE_FREQUENCY</code> property, does the following to aggregate the data stored in MySQL:</p> <ol style="list-style-type: none"> 1. Retrieves container and network metrics from MySQL 2. Aggregates the data for each app over the last four minutes, grouped by one minute intervals 3. Inserts the aggregated data into the <code>app_metric_rollup</code> table of the MySQL component

How the PCF Metrics UI Retrieves Data from the Datastores

This section describes the flow of data through the system when you interact with the PCF Metrics UI.

Components

The diagram below highlights the components involved in this process.



Process

The following table describes how the components act during each stage.

Stage	Description
1	A user launches <code>metrics.SYSTEM-DOMAIN</code> in a browser and enters her UAA credentials.
2	After the UAA authorizes the user, the browser does the following: <ol style="list-style-type: none"> Retrieves through the Cloud Controller API a list of apps that the user can access Displays a page in which the user can select any app returned by the Cloud Controller API
3	A user selects an app from the dropdown menu, which does the following: <ol style="list-style-type: none"> Opens a Server-Sent Events (SSE) connection to the <code>metrics</code> app (metrics API) Sends HTTP Put requests to the metrics API to retrieve metrics and logs for the specified time frame
4	The metrics API receives the requests from the browser and does the following:

	<ol style="list-style-type: none"> 1. Communicates with the UAA and Cloud Controller to confirm that the user can access data for the requested app 2. Creates jobs on Redis channels that describe the type of metric, log, or event requested, as well as the time period <p>Note: PCF Metrics uses Redis as a pub-sub ↗ mechanism between the metrics API and worker apps to marshal metrics and logs.</p>
5	<p>The <code>worker-app-dev</code> and <code>worker-app-logs</code> apps, which subscribe to the job channels on Redis, recognize the jobs created by the metrics API. The apps remove their corresponding jobs and do the following:</p> <ol style="list-style-type: none"> 1. Retrieve data from the datastores: <ol style="list-style-type: none"> a. <code>worker-app-dev</code> queries MySQL to retrieve any metrics and events requested for the time period. b. <code>worker-app-logs</code> queries Elasticsearch to retrieve the logs for the time period requested. 2. Publish the data to Redis
6	Redis forwards the data to the metrics API.
7	The metrics API streams the data to the browser over SSE, and the PCF Metrics UI displays the data requested by the user.

How Worker Apps Monitor the System

The following table describes the two worker components that PCF Metrics uses to monitor other components in the system.

Worker Component	Function
<code>worker-health-check</code>	<p>The health-check worker is an app that does the following every minute:</p> <ul style="list-style-type: none"> • Checks whether the apps deployed by PCF Metrics can reach the MySQL, Elasticsearch, and Redis datastores • Records the number of MySQL connections and Redis channels
<code>worker-reaper</code>	<p>The reaper worker is an app that removes orphaned connections from the <code>worker-app-dev</code> and <code>worker-app-logs</code> apps to Redis.</p> <p>PCF Metrics requires the reaper worker because Redis does not remove its connections to <code>worker-app-dev</code> and <code>worker-app-logs</code> if they restart.</p>

PCF Metrics Release Notes and Known Issues

v1.2.11

Release Date: July 12, 2017

Notes

The following list describes what's new in PCF Metrics v1.2.11:

- **MySQL Upgrade:** PCF Metrics v1.2.11 now uses v32.9 of the MySQL release.

Known Issues

See the Known Issues section for the previous release.

v1.2.10

Release Date: June 6, 2017

Notes

The following list describes what's new in PCF Metrics v1.2.10:

- **Stemcell Bump:** PCF Metrics v1.2.10 includes a major stemcell bump from 3263.x to 3363.x.

Known Issues

See the Known Issues section for the previous release.

v1.2.9

Release Date: April 14, 2017

Notes

The following list describes what's new in PCF Metrics v1.2.9:

- **Internetless Installations:** PCF Metrics v1.2.9 removes multiple unnecessary dependencies that prevented the tile from being installed in an internetless environment.
- **Reduced MySQL Disk Usage:** Raw data in MySQL is now pruned after 2 days, greatly reducing the amount of disk space required to store metrics in MySQL.

Known Issues

See the Known Issues section for the previous release.

v1.2.8

Release Date: April 7, 2017

 **Note:** You may run into smoke test failures when installing this tile. Consider upgrading to our newest patch, v1.2.9, which is more stable

Notes

The following list describes what's new in PCF Metrics v1.2.8:

- **Elasticsearch Resource Reduction:** PCF Metrics v1.2.8 reduces VMs used by removing the Elasticsearch Coordinator job and allowing configuration of Elasticsearch Master node count.
- **Simplified Tile Configuration:** Removed redundant Elasticsearch Heap Size property from OpsMan tile config.
- **Internetless Installations:** Removed multiple unnecessary dependencies that prevented the tile from being installed in an internetless environment.
- **Reduced MySQL Disk Usage:** Raw data in MySQL is now pruned after 2 days, greatly reducing the amount of disk space required to store metrics in MySQL.

Known Issues

See the Known Issues section for the previous release.

v1.2.7

Release Date: Feb 09 2017

Notes

- Metrics works in PCF environment with certificates signed by custom CA

Known Issues

See the Known Issues section for the previous release.

v1.2.6

Release Date: January 27 2017

Notes

- Includes a cli patch that resolves tile install failure due to resource matching timeouts
- Enabled the rejoin-unsafe errand for MySQL

Known Issues

See the Known Issues section for the previous release.

v1.2.5

Release Date: January 13 2017

Notes

- Addresses conflict between mysql and metrics which caused mysql monitoring to fail intermittently
- Upgraded to new mysql release which stabilizes metrics tile upgrades

Known Issues

See the Known Issues section for the previous release.

v1.2.4

Release Date: January 10 2017

Notes

- Addresses a redis alpha version dependency bug

Known Issues

See the Known Issues section for the previous release.

v1.2.3

Release Date: December 16 2016

Notes

- Enables installs for internetless environments
- Optimize MySQL disk usage by reducing bin log retention
- Added MySQL and ES configuration changes to improve the stability of cluster during restarts
- Fixed an integration issue with Apps manager that restores link to metrics

Known Issues

See the Known Issues section for the previous release.

v1.2.1

Release Date: November 29 2016

Notes

The following sections describe what's new in PCF Metrics v1.2.1.

New Requirements and Configurations

PCF Metrics v1.2.1 requires the Redis tile v1.6 or later.

Consider the following updates when configuring and deploying PCF Metrics v1.2.1:

- **MySQL Alerts:** You can now specify an email address that receives alerts from PCF Metrics about MySQL storage issues. See the [MySQL Alerts](#) section of the *Installing PCF Metrics* topic for more information.
- **Data Store:** You can now configure additional properties in the **Data Store** pane. See the [Data Store](#) section of the *Installing PCF Metrics* topic for more information.
- **Errands:** The tile includes new errands. See the [Errands](#) section of the *Installing PCF Metrics* topic for more information.
 - The new **Smoke Test** tile errand summarizes any errors that occur during the test, linking directly to new [troubleshooting documentation](#).
- **Resource Config:** See the [Increased Data Persistence](#) section below to learn about updates affecting this configuration pane.
- **Stemcell:** This release uses stemcell v3263.

Increased Data Persistence

PCF Metrics v1.2.1 increases the retention period for log, metric, and event data to two weeks. To support this increase, PCF Metrics v1.2.1 includes the following component updates:

- **Log datastore:** The default size of the Elasticsearch cluster is now nine VMs instead of three.
- **Metrics datastore:** The tile stores metrics data in MySQL instead of IMDS and increases the default number of metrics storage VMs to seven instead of five to support higher capacity and introduce high availability.
- **All VMs:** The default persistent disk usage is ten times greater than in v1.1.x.

See the [Sizing PCF Metrics for Your System](#) topic for updated guidance and the [PCF Metrics Product Architecture](#) topic for an updated explanation of the components.

Improved UI

PCF Metrics v1.2.1 improves the user experience by removing the **Dashboard** screen and replacing it with an enhanced **Explore** screen. The following list describes how PCF Metrics v1.2.1 further refines its UI. To view the UI and understand the new functionality, see the [Using PCF Metrics](#) topic.

- **Viewing historical data:** You can now view data for the past two weeks.
- **Monitoring app events:** The new **Events** view displays all start, stop, crash, and update events.
- **Searching Logs:** The **Logs** view now includes a histogram that displays log frequency for the current time frame. The histogram updates based on your search results. You can also download your logs.
- **Setting the time frame:** PCF Metrics v1.2.1 improves time frame selection.

Known Issues

The following sections describe the known issues in PCF Metrics v1.2.1.

Compatibility with Elastic Runtime

PCF Metrics v1.2.x requires Elastic Runtime v1.8.9 or later.

Data Loss when Upgrading from v1.1 to v1.2

The upgrade process from v1.1 to v1.2 acts in the following sequence:

1. Removes the data storage components of v1.1
2. Deploys v1.2 data storage and ingestion components

The upgrade process does not save any v1.1 data and the new components do not begin ingesting and storing log or metrics data until they successfully deploy.

Data Loss during Stemcell Upgrades

The Elasticsearch cluster updates when PCF Metrics applies a new stemcell, which causes decreased log ingestion and, consequently, the loss of logs. Some storage occurs during this time because canary deployed nodes can intake some of the load from firehose. Ingestion reduction may vary depending on deployment size, but you can expect some loss.

In testing, the Elasticsearch cluster took approximately five minutes to update and experienced between a 40% and 60% reduction in logs ingested and stored.

Smoke Test Failure

The PCF Metrics **Smoke Test** errand may fail if your deployment authenticates user sign-ons with an external SAML identity provider or an external LDAP server. In some cases, these external user stores have an additional login procedure that prevents the errand from authenticating with the deployment and validating against the Metrics API.

If you experience this issue, disable the **Smoke Test** errand in the PCF Metrics tile and click **Apply Changes** to run the install again.

See the [Configure Authentication and Enterprise SSO](#) section of the *Configuring Elastic Runtime* topic for more information on what configurations can lead to this failure.

For Operators who Deploy PCF Metrics using BOSH

If both of the following are true, you may experience issues while using PCF Metrics:

- You deploy PCF Metrics using BOSH instead of using the PCF Metrics tile in Ops Manager.
- You use self-signed certificates.

Pivotal recommends using certificates issued by a Certificate Authority for BOSH deployments of this product.

Past Minor v1.1.x

Release Notes for v1.1.x releases can be found [here](#)

Past Minor v1.0.x

Release Notes for v1.0.x releases can be found [here](#)