



# **Red Hat Mobile Application Platform 4.7**

## **Operations Guide**

For Red Hat Mobile Application Platform 4.7



# Red Hat Mobile Application Platform 4.7 Operations Guide

---

For Red Hat Mobile Application Platform 4.7

## Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document describes how to perform monitoring, debugging, and logging for RHMAP.

## Table of Contents

<b>CHAPTER 1. MONITORING RHMAP WITH NAGIOS</b> .....	<b>3</b>
1.1. ACCESSING THE NAGIOS DASHBOARD	3
1.2. RETRIEVING NAGIOS LOGIN CREDENTIALS	4
<b>CHAPTER 2. MONITORING RHMAP WITH COCKPIT</b> .....	<b>6</b>
2.1. OVERVIEW	6
2.2. INSTALLATION	6
2.2.1. Installing Cockpit Manually	6
2.3. VIEWING THE CONTAINERS ON AN OPENSIFT NODE	7
2.4. VIEWING MULTIPLE HOSTS SIMULTANEOUSLY	7
<b>CHAPTER 3. CENTRALIZED LOGGING FOR CORE AND MBAAS COMPONENTS</b> .....	<b>8</b>
3.1. ENABLING CENTRALIZED LOGGING	8
3.2. ACCESSING LOGS THROUGH KIBANA WEB CONSOLE	8
3.2.1. Viewing Logs of a Single Pod	8
3.2.2. Accessing Kibana Directly	8
3.2.3. Configuring an Index Pattern	8
3.3. TRACKING INDIVIDUAL REQUESTS IN LOGS	9
3.4. IDENTIFYING ISSUES IN A RHMAP CORE	10
3.5. IDENTIFYING ISSUES IN AN MBAAS	11
3.6. VIEWING ALL DEBUG LOGS FOR A COMPONENT	11
3.7. ANALYZING THE SEARCH RESULTS	12
3.8. ENABLING FHC TO ACCESS CENTRALIZED LOGS	12
<b>CHAPTER 4. FH-SYSTEM-DUMP-TOOL</b> .....	<b>15</b>
4.1. OVERVIEW	15
4.2. INSTALLATION	15
4.3. REQUIREMENTS	15
4.4. USAGE	15
4.5. UNDERSTANDING THE OUTPUT	15
4.6. INFORMATION CONTAINED IN THE DUMP ARCHIVE	16
4.6.1. Platform Data	16
4.6.2. Project Data	16
4.7. DEBUGGING	16
<b>CHAPTER 5. PROPERTY MANAGEMENT</b> .....	<b>17</b>
5.1. REQUIREMENTS	17
5.2. MANAGING PROPERTIES	17
5.3. EXAMPLE PROPERTIES MODIFICATION	18



# CHAPTER 1. MONITORING RHMAP WITH NAGIOS

To monitor the status of the RHMAP Core, MBaaS, and their components, you can use the Nagios monitoring software that comes prepackaged with an RHMAP installation.

## Prerequisites

- An OpenShift user account with access to the RHMAP Core or MBaaS project.

## 1.1. ACCESSING THE NAGIOS DASHBOARD

1. To obtain the URL for the Nagios dashboard, enter the following command:

```
oc get route nagios --template "https://{{.spec.host}}"
```

```
https://nagios-rhmap-core2.local.feedhenry.io
```

2. Navigate to the URL to reach the Nagios dashboard login screen that prompts you for a user name and password:

### Authentication Required

**https://nagios-rhmap.local.feedhenry.io requires a username and password.**

User Name:

Password:

Cancel

Log In

3. Retrieve the Nagios Dashboard user name and password by following the [Section 1.2, “Retrieving Nagios Login Credentials”](#) section of this guide.
4. Enter the retrieved credentials into the login prompt.

Upon successful login, the Nagios Dashboard is displayed:

# Nagios®

## General

Home  
Documentation

## Current Status

Tactical Overview  
Map  
Hosts  
Services  
Host Groups  
Summary  
Grid  
Service Groups  
Summary  
Grid  
Problems  
Services (Unhandled)  
Hosts (Unhandled)  
Network Outages

Quick Search:

## Reports

Availability  
Trends  
Alerts  
History  
Summary  
Histogram  
Notifications  
Event Log

## System

Comments  
Downtime  
Process Info  
Performance Info  
Scheduling Info  
Configuration

## Nagios® Core™ Version 4.0.8

August 12, 2014  
Check for updates

**A new version of Nagios Core is available!**  
Visit [nagios.org](http://nagios.org) to download Nagios 4.2.0.

### Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

### Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

### Latest News

- Nagios Plugins 2.0.2 Released
- Nagios Projects Moved To GitHub
- Nagios Core 4.0.6 Released
- More news...

### Don't Miss...

- Interested in speaking at Nagios World Conference 2014? Learn more and apply today at [go.nagios.com/conference](http://go.nagios.com/conference).
  - Improve your Nagios skillset with self-paced and instructor led training services.
- Don't miss the Nagios World Conference October 13th-16th, 2014. 3 days of presentations, industry experts, networking opportunities, and more. Register today before the conference fills up!



Copyright © 2010-2014 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.



All Nagios service checks are located under the *Services* section of the dashboard which is located in the navigation bar on the left-hand side of the Nagios homepage, see example *Services* page below:

# Nagios®

## General

Home  
Documentation

## Current Status

Tactical Overview  
Map  
Hosts  
Services  
Host Groups  
Summary  
Grid  
Service Groups  
Summary  
Grid  
Problems  
Services (Unhandled)  
Hosts (Unhandled)  
Network Outages

Quick Search:

## Reports

Availability  
Trends  
Alerts  
History  
Summary  
Histogram  
Notifications  
Event Log

## System

Comments  
Downtime  
Process Info  
Performance Info

### Current Network Status

Last Updated: Wed Aug 24 16:01:03 UTC 2016  
Updated every 90 seconds  
Nagios® Core™ 4.0.8 - [www.nagios.org](http://www.nagios.org)  
Logged in as nagiosadmin

View History For all hosts  
View Notifications For All Hosts  
View Host Status Detail For All Hosts

### Host Status Totals

Up Down Unreachable Pending  
1 0 0 0  
All Problems All Types  
0 1

### Service Status Totals

Ok Warning Unknown Critical Pending  
9 0 0 0 0  
All Problems All Types  
0 9

### Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Database::MongoDB::Health	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	OK: There are 1 primary and 2 secondary members in the replica set
	Storage::Pod::Disk Usage	OK	08-24-2016 16:00:52	0d 0h 0m 37s	1/3	Checked 101 volumes (0 critical, 0 warning)
	fr-mbaas::Health	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	No issues to report. All tests passed without error.
	fr-mbaas::Ping	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	HTTP OK: HTTP/1.1 200 OK - 208 bytes in 0.029 second response time
	fr-messaging::Health	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	No issues to report. All tests passed without error.
	fr-messaging::Ping	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	HTTP OK: HTTP/1.1 200 OK - 121 bytes in 0.053 second response time
	fr-metrics::Health	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	No issues to report. All tests passed without error.
	fr-metrics::Ping	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	HTTP OK: HTTP/1.1 200 OK - 121 bytes in 0.039 second response time
	fr-statsd::Ping	OK	08-24-2016 16:00:52	0d 0h 0m 11s	1/3	HTTP OK: HTTP/1.1 200 OK - 121 bytes in 0.043 second response time

Results 1 - 9 of 9 Matching Services

For further details on using Nagios, refer to the [Nagios documentation](#).

## 1.2. RETRIEVING NAGIOS LOGIN CREDENTIALS

The Dashboard login credentials are stored as environment variables within the Nagios container. To access them, enter the following command:

```
oc env dc/nagios --list | grep NAGIOS
```

Sample output:

```
NAGIOS_USER=nagiosadmin
NAGIOS_PASSWORD=tser56m1d6
```



The returned values are the login credentials to the Nagios dashboard.

## CHAPTER 2. MONITORING RHMAP WITH COCKPIT

### 2.1. OVERVIEW

System resources of nodes and containers in the Core and MBaaS on OpenShift 3 can be monitored and managed using *Cockpit*.

Cockpit is a system administration tool, that provides insights into how nodes and containers are performing. It lets you monitor current values and adjust limits on system resources, control lifecycle of container instances, and manipulate container images. For more information about Cockpit, refer to the official web site of the [Cockpit Project](#) and its [Documentation](#).

### 2.2. INSTALLATION

For most OpenShift 3 instances, Cockpit is most likely already installed on all nodes. This is not the case if your nodes use the RHEL Atomic Host, where Cockpit needs to be installed manually.

To check whether Cockpit is installed in your OpenShift cluster, try visiting the URL of the Cockpit web interface:

```
http://<master node host>:9090
```

If there's no response to the request, Cockpit is most likely not installed.

#### 2.2.1. Installing Cockpit Manually

1. Install Cockpit on nodes.

The following three steps must be repeated for each node you wish to monitor in your OpenShift cluster.

2. Log in to the node.

```
ssh <node host>
```

3. Install Cockpit packages.

```
yum install cockpit cockpit-docker
```

4. Enable and start the Cockpit service.

```
systemctl enable cockpit.socket  
systemctl start cockpit.socket
```

5. Create a Cockpit system user on master.

To log in to the Cockpit web interface, you will have to provide the username and password of an operating system user existing on the OpenShift master node. This guide refers to this user as the *Cockpit system user*. To allow Cockpit to access system resources, perform operations on Container and Kubernetes resources, the Cockpit system user must:

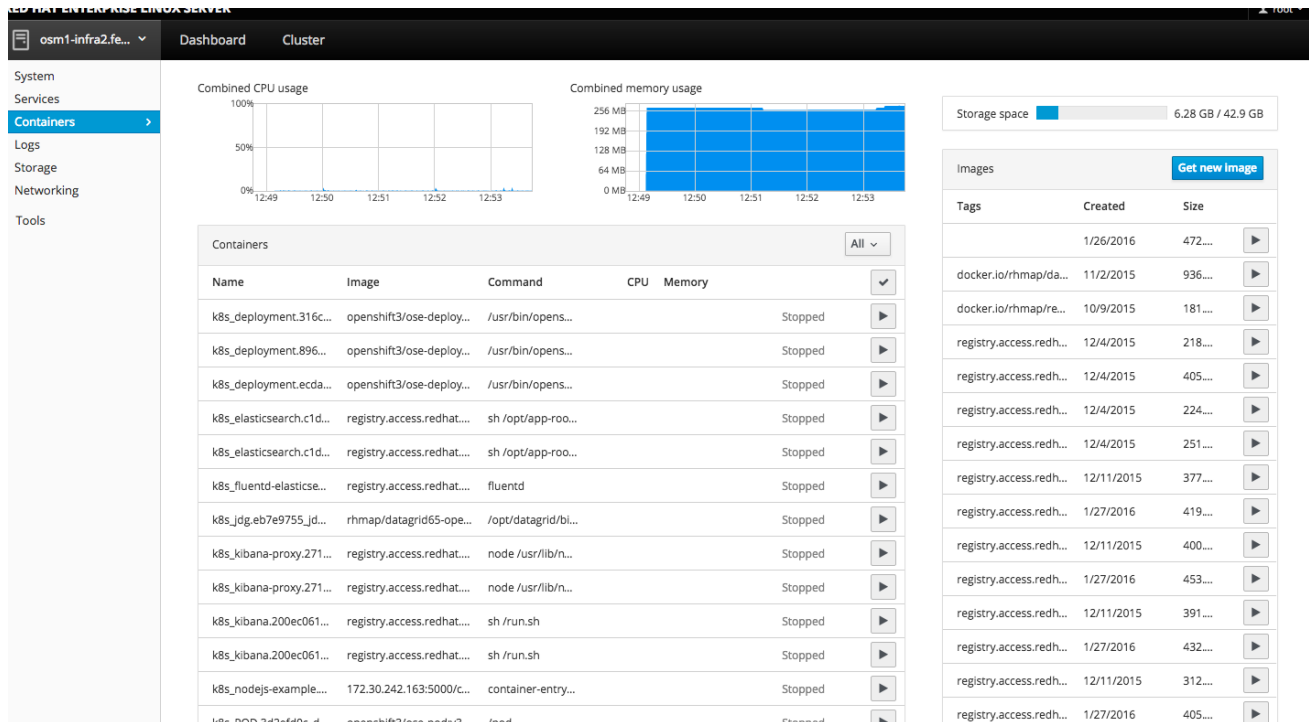
- be in the **docker** group;
- be able to log in to other nodes using **ssh**;

- be able to perform Kubernetes operations.

Create the Cockpit system user on the master node, or modify an existing user to have the necessary privileges.

## 2.3. VIEWING THE CONTAINERS ON AN OPENSIFT NODE

Navigate to the Cockpit dashboard for a node in a web browser (port 9090 by default) and log in as the Cockpit system user. To see all containers deployed on that node, click *Containers* in the left-hand side menu.



You can filter the list to only display running containers, using the dropdown menu above the list of containers. This view lets you see the RAM and CPU usage of all running containers.

If you select an MBaaS node, you will see the containers for all MBaaS components. Clicking on a container will show the current logs, CPU shares, and RAM usage. In the *Tools* menu on the left hand side, you can get terminal access into the node for further investigation.

## 2.4. VIEWING MULTIPLE HOSTS SIMULTANEOUSLY

Cockpit can connect to multiple hosts from a single Cockpit session. This can be useful to compare resource usage of two or more machines in the same dashboard. See [Multiple Machines](#) in the Cockpit documentation for more information.

## CHAPTER 3. CENTRALIZED LOGGING FOR CORE AND MBAAS COMPONENTS

Logging output from RHMAP Core and MBaaS components can be aggregated and accessed through a web console when using a RHMAP Core or MBaaS backed by OpenShift Enterprise 3 (OSEv3).

### 3.1. ENABLING CENTRALIZED LOGGING

Aggregated logging is enabled by deploying an *EFK logging stack* to your OSEv3 instance, which consists of the following components:

- [Elasticsearch](#) indexes log output collected by Fluentd and makes it searchable.
- [Fluentd](#) collects standard output of all containers.
- [Kibana](#) is a web console for querying and visualizing data from Elasticsearch.

To enable this functionality, follow the official OpenShift guide [Aggregating Container Logs](#).

### 3.2. ACCESSING LOGS THROUGH KIBANA WEB CONSOLE

The Kibana web console is where logs gathered by Fluentd and indexed by Elasticsearch can be viewed and queried. You can access the Kibana web console via the OpenShift web console, or directly by its URL configured through the **KIBANA\_HOSTNAME** in the deployment procedure.

#### 3.2.1. Viewing Logs of a Single Pod

If you have configured **loggingPublicURL** in [section 28.5.4](#) of the deployment procedure, the OpenShift web console allows you to view the log archive of a particular pod.

1. In the OpenShift web console, select a project you are interested in.
2. Click on the *Pods* circle of the specific service.
3. Choose one of the pods to inspect.
4. Click on the *Logs* tab.
5. Click on the *View Archive* button at the top right corner to access the logs of the chosen pod in the Kibana web console.



#### NOTE

By default, Kibana's time filter shows the last 15 minutes of data. If you don't see any values, adjust the *Time filter* setting to a broader time interval.

#### 3.2.2. Accessing Kibana Directly

You can access the Kibana web console directly at [https://KIBANA\\_HOSTNAME](https://KIBANA_HOSTNAME), where **KIBANA\_HOSTNAME** is the host name you set in step 4 of the deployment procedure.

#### 3.2.3. Configuring an Index Pattern

When accessing the Kibana web console directly for the first time, you are presented with the option to configure an index pattern. You can also access this configuration screen in the *Settings* tab.

For MBaaS deployments, there is an index pattern in the format **<MBaaS ID>-mbaas.\***, matching the ID of the deployed MBaaS target.

For RHMAP Core deployment, there is an index pattern **core.\***.

To make queries more efficient, you can restrict the index pattern by date and time.

1. Select the *Use event times to create index names*
2. Enter the following pattern in the *Index name or pattern* input text field. For example:

```
[onprem-mbaas.]YYYY.MM.DD
```

3. You will see output similar to the following below the input field

```
Pattern matches 100% of existing indices and aliases
onprem-mbaas.2016.02.04
onprem-mbaas.2016.02.05
```

4. Click *Create* to create the index based on this pattern.
5. You can now select this newly created index in the *Discover* tab when doing searches, as well as in other parts, such as the *Visualizations* tab.

### 3.3. TRACKING INDIVIDUAL REQUESTS IN LOGS

Every request to the RHMAP platform has a unique internal identifier assigned, which helps in identifying the sequence of events in Core and MBaaS components triggered by the request.

For example, if a user is deploying a form to an environment, the ID in the logging statements resulting from the request will be identical.

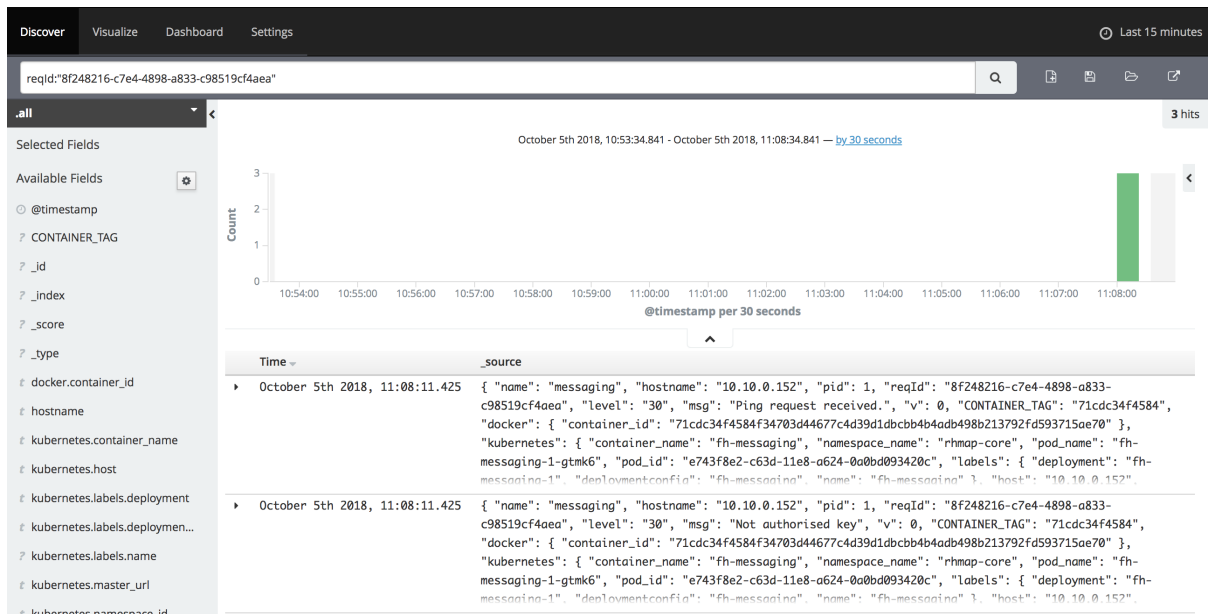


#### NOTE

Only requests from **fhc** and Studio get an identifier assigned, **not** requests from mobile applications to an MBaaS.

Search for log statements related to a specific request in one of the following ways:

- **Using Kibana**
  - Filter by the **reqId** field. For example **reqId:"8f248216-c7e4-4898-a833-c98519cf4aea"**.
  - Use the **.all** index to search in logs from components of both Core and MBaaS.



### • Using fhc

1. Enable **fhc** to access the logging data, as described in [Section 3.8, “Enabling fhc to Access Centralized Logs”](#).
2. Use the **admin logs syslogs** command of **fhc**:

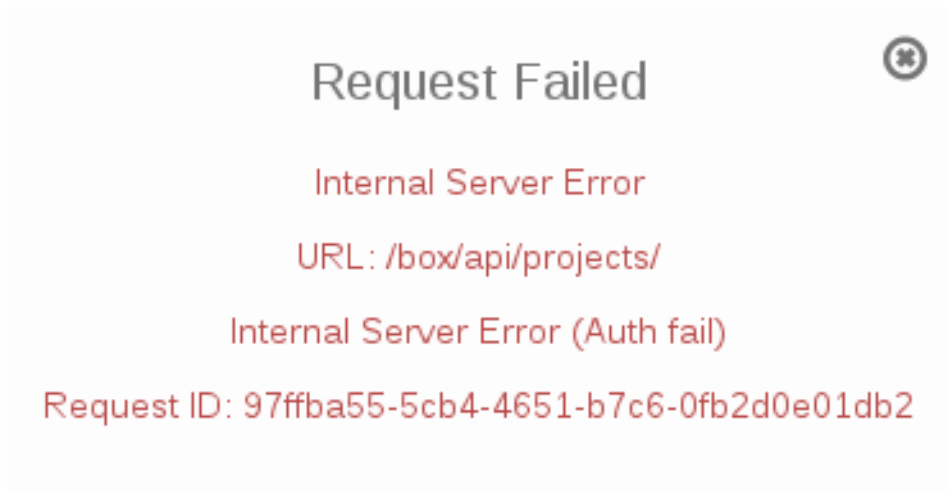
```
fhc admin logs syslogs --requestId 559d8f74-32d2-4c6e-b1a2-b46d2993e874 --projects="core,mbaas"
```

Set **--projects** to a comma-separated list of OpenShift project names to search in.

## 3.4. IDENTIFYING ISSUES IN A RHMAP CORE

If you encounter unexpected errors in RHMAP Core UI, you can use Kibana's *Discover* tab to find the root of the problem. Every request that the RHMAP Core UI sends has a unique identifier that can be used to gather the relevant logs. The following steps describe the procedure:

1. Identify the request ID associated to the failed request you want to investigate  
Errors in the platform usually manifests in UI as a notification pop-up, containing information about the URL endpoint the failed request was targeting, the error message and the Request ID. Take the note of the Request ID.



2. Query for the relevant logs in Kibana

Log in to your Kibana instance and go to the *Discover* tab. Enter a query in form **reqId:** and you should see all of the logs relating to the failing request.

Useful fields to display include:

- **msg**
- **message**
- **kubernetes\_container\_name**
- **level**

### 3.5. IDENTIFYING ISSUES IN AN MBAAS

If you suspect that an error of an MBaaS component may be the cause of an issue, you can use Kibana's *Discover* tab to find the root of the problem. The following steps describe the general procedure you can follow to identify issues.

1. Select the index for the MBaaS target you are interested in  
Use the dropdown just below the input bar in the *Discover* view to list all available indices. An [index](#) is similar to a database in relational database systems. Select which index your searches will be performed against.
2. Select a time interval for your search  
Click the *Time Filter* (clock icon) and adjust the time interval. Initially, try a broader search.
3. Perform a simple search  
To search for all error events, perform a simple search for **error** in the *Discovery* field. This will return the number of hits within the chosen time interval.
4. Select the **msg** or **message** field to be displayed  
On the left hand side of the *Discover* view is a list of fields. From this list you can select fields to display in the document data section. Selecting a field replaces the **\_source** field in the document data view. This enables you to see any error messages and might help you refine your original search if needed. You can also select more fields to help you locate the issue.

### 3.6. VIEWING ALL DEBUG LOGS FOR A COMPONENT

If searching for error messages doesn't help, you can try looking into the debug logs of individual components.

1. Select the index for the target that you are interested in
2. Start a new search  
Click on the *New Search* button to the left of the search input bar, which looks like a document with a plus sign.
3. Search a component for all debug messages  
For example, to search for all debug messages of the **fh-messaging** component, enter the following query:

```
type: bunyan && level: 20 && kubernetes_container_name: "fh-
messaging"
```

-

If you know some part of the error message, you can specify that as part of the search:

```
type: bunyan && level: 20 && kubernetes_container_name: "fh-
messaging" && "Finished processing"
```

You can narrow down your search further by time, as described in step 5 above.

As a reference, the following are the Bunyan log levels:

```
TRACE = 10;
DEBUG = 20;
INFO = 30;
WARN = 40;
ERROR = 50;
FATAL = 60;
```

## 3.7. ANALYZING THE SEARCH RESULTS

### 1. Narrow down the time interval

The histogram shows search hits returned in the chosen time interval. To narrow down the search in time you have the following options:

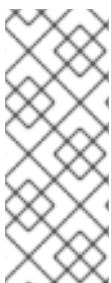
- Click on a bar in the histogram to narrow down the search to that bar's time interval.
- Select a time window in the date histogram by clicking and dragging between the start/end time you are interested in.

### 2. Inspect the document data

Once you narrow down the search, you can inspect the document data items. Apart from the **msg** and **message** fields, you might be interested in **kubernetes\_pod\_name** to see the pod a message originates from.

## 3.8. ENABLING FHC TO ACCESS CENTRALIZED LOGS

To enable the **fhc admin logs syslogs** feature for searching platform logs by request IDs, configure **fh-supercore** to have access to Elasticsearch by following the steps in this section.



### NOTE

If **fh-supercore** is not configured for access to Elasticsearch, running **fhc admin logs syslogs** yields an error message similar to the following:

```
FH-SUPERCORE-ERROR - Aggregated Logging is not enabled for this
cluster.
```

1. Enable centralized logging, as described in [Section 3.1, "Enabling Centralized Logging"](#).
2. Create a route to allow external access to Elasticsearch.
  - a. Log in to your OpenShift cluster

```
oc login <url-of-openshift-master>
```



- 
- b. Select the existing logging project.

```
oc project <logging-project-name>
```

- c. Create a route to allow external access to Elasticsearch. Replace the values in angle brackets as appropriate for your environment.

```
oc create route passthrough --service=<elasticsearch-route-name>
--hostname=<elasticsearch-hostname>.<openshift-master-hostname>
```

3. Create a secret for **fh-supercore**.

To read from Elasticsearch, fh-supercore will use the existing Kibana credentials. The existing Kibana certificate and key can be used. These can be read from the existing secret and decoded.

- a. Read the secret and output to JSON format.

```
oc get secret logging-kibana -o json
```

This will output a base64-encoded representation of the certificate in "data.cert" and key in "data.key". We can now decode this and create a plain-text key and cert in our temp directory. Replace the output from the above command into the commands below.

- b. Decode the key and output to the **/tmp** directory or otherwise.

```
echo "<contents-of-data.key>" | base64 --decode >
/tmp/supercoreKey.key
```

- c. Decode the certificate.

```
echo "<contents-of-data.cert>" | base64 --decode >
/tmp/supercoreCert.crt
```

- d. Switch to the Core project.

```
oc project <core-project-name>
```

- e. Create a secret for **fh-supercore** that will use the Kibana credentials to perform searches.

```
oc secrets new <core-secret-name> key=/tmp/supercoreKey.key
crt=/tmp/supercoreCert.crt
```

A new secret is created in the core project called **<core-secret-name>** as specified above.

4. Update the deployment configuration of **fh-supercore**.

- a. Open the editor for **fh-supercore** deployment configuration.

```
oc edit dc fh-supercore
```

- b. Set properties.

Name	Value
<b>FH_ES_LOGGING_ENABLED</b>	<b>true</b>
<b>FH_ES_LOGGING_HOST</b>	<a href="https://&lt;elasticsearch-hostname&gt;.&lt;openshift-master-hostname&gt;">https://&lt;elasticsearch-hostname&gt;.&lt;openshift-master-hostname&gt;</a>
<b>FH_ES_LOGGING_KEY_PATH</b>	<b>/etc/fh/es-keys/key</b>
<b>FH_ES_LOGGING_CERT_PATH</b>	<b>/etc/fh/es-keys/crt</b>
<b>FH_ES_LOGGING_API_VERSION</b>	<b>1.5</b> ( <i>the version of Elasticsearch used by Openshift 3.2</i> )

For example, if **<core-secret-name>** was **supercore-elasticsearch**

```
spec:
  template:
    spec:
      volumes:
      -
        name: supercore-elasticsearch-volume
        secret:
          secretName: supercore-elasticsearch
      containers:
      -
        name: fh-supercore
        volumeMounts:
        -
          name: supercore-elasticsearch-volume
          readOnly: true
          mountPath: /etc/fh/es-keys
```

## CHAPTER 4. FH-SYSTEM-DUMP-TOOL

### 4.1. OVERVIEW

The `fh-system-dump-tool` allows you to analyze all the projects running in an OpenShift cluster and reports any problems discovered. Although this tool reports errors found in any project on the OpenShift Platform, it is primarily used to debug issues with RHMAP Core and MBaaS installations.

Running `fh-system-dump-tool` may take some time, depending on the complexity of the environment. When the analysis is finished, the tool reports any commonly found issues that might reflect a problem on the cluster or a project.

The `fh-system-dump-tool` archives the dump directory and the analysis results in a `tar.gz` file, which can be emailed to Red Hat Support, or decompressed and read locally.

### 4.2. INSTALLATION

Install the `fh-system-dump-tool` using the following command:

```
subscription-manager repos --enable= rhel-7-server-rhmap-4.7-rpms  
yum install fh-system-dump-tool
```

### 4.3. REQUIREMENTS

The `fh-system-dump-tool` requires a local [installation of the oc binary](#).

The `fh-system-dump-tool` also requires that the `oc` binary has a [logged in user](#) on the platform you wish to analyze. For `fh-system-dump-tool` to analyze a project, the logged in user must have access to that project and the logged in user must have the `cluster-reader` role, or equivalent permissions.

A Core or MBaaS running on OpenShift also contains a Nagios pod which monitors the platform and detects issues. The `fh-system-dump-tool` uses the Nagios data to analyze the platform and find faults. If the `fh-system-dump-tool` cannot locate Nagios it cannot perform a complete analysis.

### 4.4. USAGE

The `fh-system-dump-tool` creates a directory called `rhmap-dumps` in the working directory and stores archive data in that directory.

To execute the tool use the following command:

```
fh-system-dump-tool
```

### 4.5. UNDERSTANDING THE OUTPUT

When the tool starts, it stores dump data and then performs an analysis. If the tool encounters any issues during the analysis phase, the errors are output to `stderr`. For more information on debugging errors, see [Debugging](#).

Once the dump and analysis process is complete, the tool alerts the user of possible errors found in the OpenShift cluster and projects.

Finally, the dump and the analysis results are all archived into a **tar.gz** file and the tool reports the location of this file, which is timestamped. If you need to send this file for additional support, make sure that the file name and contents are unaltered, unless you are instructed otherwise by Red Hat Support.

## 4.6. INFORMATION CONTAINED IN THE DUMP ARCHIVE

Review the list of platform-level and project-level data that is included in the dumped archive, in case you consider any of the information to be sensitive, before sending the dump archive by email.

### 4.6.1. Platform Data

At a platform level, the dump includes:

- Description of all persistent volumes
- The version of the **oc** client in use
- Details and permissions of the currently logged in OpenShift user
- The output of the **oc adm diagnostics** command
- The version of the **fh-system-dump-tool** used
- The name of all the projects the current user has access to
- The results of the analysis

### 4.6.2. Project Data

For each project discovered in the cluster, the following data is included in the dumped archive:

- The definition in OpenShift for:
  - configuration maps
  - deployment configurations
  - persistent volume claims
  - pods
  - services
  - events
- The most recent logs for all available pods

## 4.7. DEBUGGING

Start debugging by reviewing the output from the analysis phase.

To debug a system, you only need access to the archive file. In the root of the archive is a file named **analysis.json** which contains a summary of all the issues discovered while scanning the OpenShift cluster and projects. Use this file to start looking for potential issues with the analyzed OpenShift platform or the RHMAP Core and MBaaS projects installed on it.

## CHAPTER 5. PROPERTY MANAGEMENT

The following steps describe how to configure the RHMAP command line tool (fhc) to enable property management.

### 5.1. REQUIREMENTS

Follow the instructions to [install the RHMAP command line tool \(fhc\)](#), and make sure that it is working.



#### WARNING

Altering properties might adversely affect RHMAP. Do not modify properties other than the example in this guide unless you are instructed to do so by Red Hat support.

### 5.2. MANAGING PROPERTIES

1. To manage properties you must first set the configuration to target the cluster using the following command, where **<cluster-url>** is the domain name of the target:

```
fhc target <cluster-url>
```

For example, if RHMAP is running on the domain <https://rhmap.example.com>, the user would set the target by entering:

```
fhc target https://rhmap.example.com
```

2. To modify the cluster properties, log in as the admin user to ensure that you have the required permissions. Either use the **fhc login** command and respond to the prompts for a username and password or use the **fhc login <username> <password>** command.



#### NOTE

If you using a self-managed Core the credentials for the admin user are stored as environment variables, **\${FH\_ADMIN\_USER\_NAME}** and **\${FH\_ADMIN\_USER\_PASSWORD}**, in the Millicore pod.

3. Use the **fhc clusterprops** command to modify properties. The following CRUDL commands are available:
  - Create: **fhc clusterprops create <property> <value>;**
  - Read: **fhc clusterprops read <property>;**
  - Update **fhc clusterprops update <property> <value>;**
  - Delete **fhc clusterprops delete <property> <value>;**

- List `fhc clusterprops list;`

## 5.3. EXAMPLE PROPERTIES MODIFICATION

As an example, you might require that when a new user is created, an invitation email is sent to the user, and the user must set their password. You can use the `password.setExpiryTime` property to make sure that the email expires.



### WARNING

Altering properties might adversely affect RHMAP. Do not modify properties other than the example in this guide unless you are instructed to do so by Red Hat support.

1. Enter the following command to view the current setting for this property:

```
fhc clusterprops read password.setExpiryTime
```

The output displays in a similar format to the following:

guid	Name	Value
none	password.setExpiryTime	720

2. To change the value to 500, use the update command:

```
fhc clusterprops update password.setExpiryTime 500
```

3. Verify the setting by entering:

```
fhc clusterprops read password.setExpiryTime
```

Check the output:

guid	Name	Value
none	password.setExpiryTime	500