Red Hat Mobile Application Platform 4.7

Installing RHMAP

For Red Hat Mobile Application Platform 4.7

Last Updated: 2018-10-25

Red Hat Mobile Application Platform 4.7 Installing RHMAP

For Red Hat Mobile Application Platform 4.7

Legal Notice

Abstract

This document provides an overview of the installation procedure for a complete RHMAP instance.

Table of Contents

CHAPTER 1. OVERVIEW

In order to install RHMAP you need to use:

- Red Hat Enterprise Linux
- Red Hat subscription-manager
- Ansible
- OpenShift

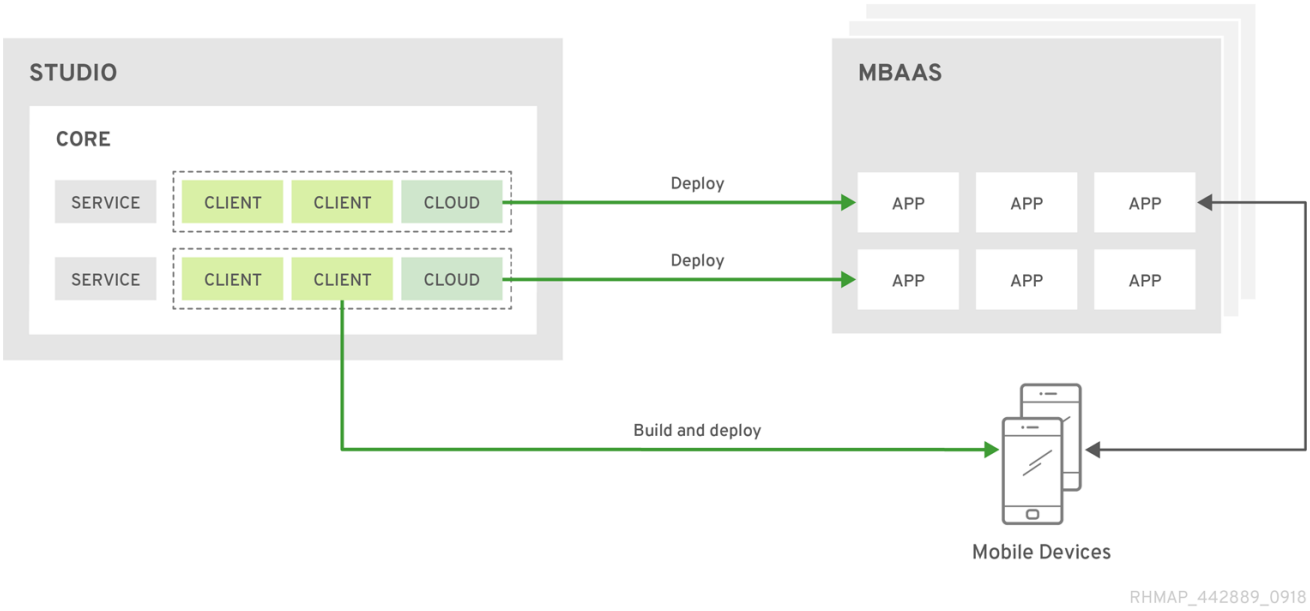
This documentation assumes a level of proficiency in these technologies. Red Hat recommends you consult the appropriate documentation or complete courses, for example, Red Hat Certifications before attempting to install RHMAP.

NOTE

Installing RHMAP self-managed on an OpenShift active-active configuration as described in the Architecture Guide is not supported.

You install RHMAP using Ansible, that is, you use an Ansible playbook (sometimes called an installer) to provision the following components:

- RHMAP Core
- RHMAP MBaaS



Familarity with Ansible will improve your experience and becomes more necessary if you system is complex, for example, if you require a HTTP Proxy.

Red Hat recommends you have OpenShift experience and a good understanding of your network configuration before attempting to install RHMAP in any configuration other than a single node system.

Follow the steps below, in the following order, to install RHMAP:

- Chapter 2, *Preparing Infrastructure for Installation* - Install Red Hat Enterprise Linux, register with Red Hat Subscription Manager, install OpenShift Container Platform on each node of the cluster. Install the RHMAP RPMs. See Appendix A, *RHMAP Ansible Playbook Pre-Requisite Checks* for more details of the requirements.

- Chapter 3, *Using a Container Image to Install RHMAP (Optional)* - Optional installation method, using a container image

- Chapter 4, *Provisioning an RHMAP 4.x Core*

- Chapter 5, *Provisioning an RHMAP 4.x MBaaS on OpenShift Container Platform* - you can install one or more MBaaSes.

- Chapter 6, *Post-installation Tasks* - set up email configuration or logging.

CHAPTER 2. PREPARING INFRASTRUCTURE FOR INSTALLATION

1. Determine which type of installation you require:

Installation Type	Replicaset	Core	MBaaS	Purpose
Single core and 1-node mbaas	no	1	1-node	Proof of Concept (POC) - all data and code stored on your infrastructure.
Single core with a 3-node MbaaS	yes	1	3-node	Production ready - All data and code stored on your infrastructure.

2. Make sure your infrastructure satisfies hardware requirements.
   Use the Red Hat Mobile Application Platform 4.x sizing tool to determine how many nodes with how many processors, and how much RAM and storage are required to run RHMAP. Alternatively, see the Infrastructure Sizing Considerations for Installation of RHMAP MBaaS for a full reference of all configurations.

3. Install Red Hat Enterprise Linux (RHEL).

   NOTE

   See the OpenShift Container Platform Tested Integrations site to determine which version of RHEL to install.

   Install RHEL on each machine that will serve as a node in the OpenShift cluster for the Core or MBaaS. For more information, see the RHEL Installation Guide.

4. Register all cluster nodes using Red Hat Subscription Manager (RHSM) and attach the nodes to the RHMAP subscription.
   For each node in the cluster:

   a. Register the node with RHSM.
      Replace **<username>** and **<password>** with the user name and password for your Red Hat account.

      ```
      sudo subscription-manager register --username=<username> --
      password=<password>
      ```

      The output is similar to the following:

      ```
      Registering to: subscription.rhn.redhat.com:443/subscription
      The system has been registered with ID: abcdef12-3456-7890-1234-
      56789012abcd
      ```

   b. List the available subscriptions.

      ```
      sudo subscription-manager list --available
      ```

c. Find the pool ID for an RHMAP subscription and attach that pool. The pool ID is listed with the product subscription information.

```
sudo subscription-manager attach --pool=<pool_id>
```

The output is similar to the following:

```
Successfully attached a subscription for: Red Hat Mobile
Application Platform
```

5. Install OpenShift version 3.7, 3.9 or 3.10.
See the Installation and Configuration guide for detailed installation procedures.

IMPORTANT

Consider the following points when performing the installation:

- One of the options documented for OpenShift installation is to use the remaining free space from the volume group where your root file system is located. Red Hat recommends that you do not choose this option, that is, you use an existing, specified volume group, or an additional block device.
- In the OpenShift *Installation and Configuration* guide, skip steps that you have already performed as part of this procedure, for example, steps 1, 2, 3 and 4 in section *2.3.3. Host Registration*, which describe the registration process.

Use the Red Hat Mobile Application Platform 4.x sizing tool or the Infrastructure Sizing Considerations for Installation of RHMAP MBaaS document to determine how many nodes to configure in your OpenShift cluster.

2.1. INSTALL THE RHMAP OPENSHIFT TEMPLATES

Get the templates by installing the RPM package **`rhmap-fh-openshift-templates`**:

1. Update subscription-manager information

```
subscription-manager refresh
```

2. Enable the *RHMAP 4.7 RPMs* repository in RHEL.

```
subscription-manager repos --enable=rhel-7-server-rhmap-4.7-rpms
```

3. Install **`rhmap-fh-openshift-templates`**

```
yum install rhmap-fh-openshift-templates
```

The Ansible scripts to install RHMAP are installed into **`/opt/rhmap/4.7/rhmap-installer`** directory.

The following templates are installed into the **`/opt/rhmap/4.7/templates/core`** directory:

- **`fh-core-backend.json`**

- **fh-core-frontend.json**
- **fh-core-infra.json**
- **fh-core-mongo-config.json**
- **fh-core-mongo.json**
- **fh-core-mongo-replica.json**
- **fh-core-monitoring.json**
- **fh-core-mysql-config.json**
- **fh-core-mysql.json**
- **fh-core-mysql-master-slave.json**
- **fh-nginx-proxy-template.json**

2.2. CONFIGURE ANSIBLE FOR INSTALLING RHMAP COMPONENTS.

To install Ansible version 2.4, see the Ansible Installation Guide.

Playbooks are Ansible's configuration, deployment, and orchestration language. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. The playbooks required to install RHMAP are included in the RPM file described in Section 2.1, "Install the RHMAP OpenShift Templates". See http://docs.ansible.com/ansible/playbooks.html for more information.

Ansible requires ssh access to the nodes in your OpenShift cluster to perform the installation. To enable this, enter the nodes into an Ansible inventory file.

2.2.1. Accessing the Cluster

Validate that you can access the cluster using one of the following methods:

- Using the following command to log into your cluster: **ssh <ansible_ssh_user>@<master_node> -t "oc login --username=<oc_user> --password=<os_password>"**. If you use this method, set the **oc_user** and **oc_password** variables as described in Section 2.2.2, "Setting Up an Inventory File".
- Using OAuth authentication (RHMAP 4.5.1 or later):
  1. Determine the value required for the **oc_oauth_token** variable by visiting **https://<domain_name>/oauth/token/request**. This generates a short-lived token for the current user.
  2. Use the token to log into your cluster: **ssh <ansible_ssh_user>@<master_node> -t -e oc_oauth_token**.
  3. Set the **oc_oauth_token** variable as described in Section 2.2.2, "Setting Up an Inventory File".

2.2.2. Setting Up an Inventory File

The RPM file include several example inventory files that can be referenced as guidelines for your installation The inventory file acts as a configuration for your installation. See http://docs.ansible.com/ansible/intro_inventory.html for more information.

For a multi-node installation, consider the following template:

```
/opt/rhmap/4.7/rhmap-installer/inventory-templates/multi-node-example
```

There are a number of parameters that can be overridden. These include settings such as that required for an outbound HTTP Proxy, the OpenShift username and password to use, and the list of nodes you intend to use for RHMAP Core and MBaaS installation. These parameters represent the global configuration options when installing RHMAP. There are more specific parameters in the Core playbooks and MBaaS playbooks that are covered under the installation guides. Below is a list of the parameters you can set in this inventory file when installing RHMAP, their uses and their defaults.

Variable	Description	Required
ansible_ssh_user	The SSH user Ansible uses to install RHMAP. This user must allow SSH-based authentication without requiring a password. Validate this setting as described in Section 2.2.1, “Accessing the Cluster”.	true
ansible_sudo	Allows Ansible to escalate privileges when required. For example when checking the required PVs exist.	true
target	The type of OpenShift are we targeting. The only value that is supported at this moment is **`enterprise`**	true
cluster_hostname	Cluster hostname. The base route for your OpenShift router.	true
domain_name	Customer subdomain name. For example **`rhmap`** which will become the base domain to access RHMAP: (rhmap.my.example.com)	true
oc_user	OpenShift User that runs commands	false
oc_password	OpenShift User password / token	false

Variable	Description	Required
login_url	The URL of the OpenShift server. For example, the URL used with you log in to the server using the **`oc`** client. Do not set this variable if using a http proxy, and the ansible tasks specified in the playbook run on an OpenShift node.	https://localhost:8443
skip_tls	If true, the OpenShift server's certificate will not be checked for validity. This will make your HTTPS connections insecure	false
kubeconfig	The path to the config file containing the client certificates for the **`system:admin`** user. Default value is **`/etc/origin/master/admin.kubeconfig`**	true
proxy_host	HTTP_PROXY_HOST value	false
proxy_port	HTTP_PROXY_PORT value	false
proxy_user	HTTP_PROXY_USER value (only needed if authentication is required, otherwise leave it blank)	false
proxy_pass	HTTP_PROXY_PASS value (only needed if authentication is required, otherwise leave it blank)	false
proxy_url	The syntax is: **`http://<proxy-host>:<proxy-port>`** or **`http://<proxy_user>:<proxy_pass>@<proxy-host>:<proxy-port>`**	false
url_to_check	URL to test prerequisite of outbound HTTP connection. Must be whitelisted if you use a HTTP Proxy server.	true
gluster_storage	This must be set to **`true`** of using Gluster FS for persistent storage. Default value is **`false`**.	true

Variable	Description	Required
gluster_metadata_name	The name which defines the Gluster cluster in the Persistent Volume definition.	true
gluster_endpoints	A comma separated list of IP values. Must be the actual IP addresses of a Gluster server, not FQDNs. For example, ["10.10.0.55","10.10.0.56"]	true
[master]	Note this is a host group. Provide a single OpenShift master node, for example: **`my-domain-master1.example.com`**.	true
[mbaas]	Note this is a host group. Each of the nodes you want to use to install the MBaaS, each node labeled as per the prerequisites, labelling is only required for the MBaaS. For example: **`my-domain-mbaas1.example.com`** **`my-domain-mbaas2.example.com`** **`my-domain-mbaas3.example.com`**	true
[core]	Note this is a host group. Each of the nodes you want to use to install the Core. **`my-domain-node1.example.com`** **`my-domain-node2.example.com`** **`my-domain-node3.example.com`**	true
oc_oauth_token	A token which enables access to the OpenShift instance without revealing the credentials of that instance	false

2.2.2.1. Configuring a Proxy Whitelist

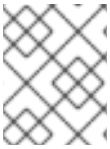
If you are using a proxy, add the following to the proxy whitelist:

- The wildcard DNS entry for your OpenShift Router, for example *example.example.net
- api-ssl.bitly.com
- github.com
- registry.npmjs.org
- fcm.googleapis.com
- db3.notify.windows.com
- gcm-http.googleapis.com
- subscription.rhn.redhat.com

CHAPTER 3. USING A CONTAINER IMAGE TO INSTALL RHMAP (OPTIONAL)

An alternative method for installation is to use the RHMAP installer container image, which includes Ansible and the templates, so you do not need to install Ansible. To use the RHMAP installer container image:

```
docker pull rhmap47/installer
```



NOTE

You still need to create an inventory file as described in Section 2.2.2, “Setting Up an Inventory File”.

The following example shows how you to run the image with an interactive terminal and mounting the directory that contains the inventory file:

```
docker run -it \
       -v ~/.ssh/id_rsa:/opt/app-root/src/.ssh/id_rsa:Z \
       -v ${HOME}/Desktop/rhmap-installer/inventories:/opt/app-
root/src/inventories \
       -e ANSIBLE_PRIVATE_KEY_FILE=/opt/app-root/src/.ssh/id_rsa \
    rhmap47/installer bash
```

CHAPTER 4. PROVISIONING AN RHMAP 4.X CORE

4.1. PREREQUISITES

This guide assumes several prerequisites are met before the installation:

- All nodes in the cluster must be registered with the Red Hat Subscription Manager. See Chapter 2, *Preparing Infrastructure for Installation* for detailed steps.
- RHMAP RPMs are installed as described in Section 2.1, “Install the RHMAP OpenShift Templates”.
- Many Core components require direct outbound internet access to operate, make sure that all nodes have outbound internet access before installation. If you use a proxy for outbound internet access, note the proxy IP address and port, you will require both for configuration during the installation.
- Ansible version 2.4 is installed on a management node which has SSH access to the OpenShift cluster. See Section 2.2, “Configure Ansible for installing RHMAP components.” for more information.
- An existing OpenShift Container Platform installation, version 3.7, 3.9 or 3.10.
- A wildcard DNS entry must be configured for the OpenShift router IP address.
- A trusted wildcard certificate must be configured for the OpenShift router. See Using Wildcard Certificates in OpenShift documentation.
- Administrative access to the OpenShift cluster via the **`oc`** cli tool. This user must be able to:
  - Create a **`Project`**, and any resource typically found in a **`Project`** (e.g. **`DeploymentConfig`**, **`Service`**, **`Route`**)
  - Edit a **`Namespace`** definition
  - Add a **`Role`** to a **`User`**
  - Manage Nodes, specifically **`labels`**
- The rhmap-installer will run a number of pre-req checks which must pass before proceeding with the installation. See RHMAP Installer Pre-Requisite Checks for details.

For information on installation and management of an OpenShift cluster and its users, see the official OpenShift documentation.

4.2. INSTALLATION

The installation of a Core in OpenShift Container Platform results in all Core components running in Replication Controller backed Pods, with Persistent Volumes for Core data.

The installation consists of several phases. Before the installation, you must prepare your OpenShift cluster:

- Set up persistent storage — you need to create Persistent Volumes to cover the Persistent Volume requirements of the Core.

- Label the nodes — nodes can be labeled if the Core components are to run on specific nodes.

After the OpenShift cluster is properly configured:

- Install the Core
- Verify the installation

4.2.1. Setting Up Persistent Storage

NOTE

Ensure that the persistent volumes are configured according to the OpenShift documentation for configuring PersistentVolumes. If you are using NFS, see the Troubleshooting NFS Issues section for more information.

The Core requires a number of persistent volumes to exist before installation. As a minimum, make sure your OpenShift cluster has the following persistent volumes in an **`Available`** state, with at least the amount of free space listed below:

Component	Minimum recommended size (Default)
MongoDB	25Gi
Metrics Data Backup	5Gi
FH SCM	25Gi
GitLab Shell	5Gi
MySQL	5Gi
Nagios	1Gi

To change the default storage requirements of a component:

1. Update the persistent volume claims in the Core OpenShift templates as described in the Persistent Volume documentation.
   The following example JSON object definition shows how to create a 25GB persistent volume with **`ReadWriteOnce`** access mode:

```
{
  "kind": "PersistentVolume",
  "apiVersion": "v1",
  "metadata": {
    "name": "examplePV"
  },
  "spec": {
    "capacity": {
      "storage": "25Gi"
    },
```

```
    "accessModes": [
      "ReadWriteOnce"
    ],
    "persistentVolumeReclaimPolicy": "Retain",
    "nfs": {
      "path": "/path/to/examplePV",
      "server": "172.17.0.2"
    }
  }
}
```

NOTE

For more information on the types of Access Modes read the Persistent Volume Access Modes documentation.

4.2.1.1. Root Squash Recommendations

Both GlusterFS and NFS have a root squash option available, however they do not function in the same manner, here are some further details regarding this setting:

4.2.1.1.1. GlusterFS

Enabling root-squash prevents remote super-users from having super-user privileges on the storage system.

Recommended setting: Off

4.2.1.1.2. NFS

root_squash prevents remote super-users from changing other user's files on the storage system.

Recommended setting: On

4.2.1.2. Persistent Storage Recommendations

It is recommended by OpenShift to only use **`HostPath`** storage for single node testing. Instead of **`HostPath`** storage, use another driver such as GlusterFS or NFS when installing the Core. For more information on types of persistent volume read the Types of Persistent Volumes documentation.

For detailed information on persistent volumes and how to create them, see Persistent Storage in the OpenShift Container Platform documentation.

4.2.2. Applying Node Labels

You can skip this entire labeling section if your OpenShift cluster only has a single schedulable node. In such case, all Core components will run on that single node.

Red Hat recommends you deploy the Core components to dedicated nodes, separated from other applications (such as the RHMAP MBaaS and Cloud Apps). However this deployment structure is not required.

To use an example, if you have two nodes where you would like the Core components to be deployed to, these two nodes should have a specific label e.g. **`type=core`**. You can check what **`type`** labels are applied to all nodes with the following command:

```
oc get nodes -L type
```

```
NAME          STATUS                     AGE       TYPE
ose-master    Ready,SchedulingDisabled   27d       master
infra-1       Ready                      27d       infra
infra-2       Ready                      27d       infra
app-1         Ready                      27d       compute
app-2         Ready                      27d       compute
core-1        Ready                      27d
core-2        Ready                      27d
mbaas-1       Ready                      27d       mbaas
mbaas-2       Ready                      27d       mbaas
mbaas-3       Ready                      27d       mbaas
```

To add a **`type`** label to the **`core-1`** and **`core-2`** nodes, use the following command:

```
oc label node core-1 type=core
oc label node core-2 type=core
```

4.2.3. Installing the Core Using Ansible

4.2.3.1. Setting Variables

The variables required for installation of RHMAP Core are set in **`/opt/rhmap/4.7/rhmap-installer/roles/deploy-core/defaults/main.yml`**. This file will allow you to configure the RHMAP Core project for your own environment.

4.2.3.2. Configure Monitoring Components

Setup the monitoring parameters with SMTP server details. This is required to enable email alerting via Nagios when a monitoring check fails. If you don't require email alerting or want to set it up at a later time, the sample values can be used.

```
monitoring:
  smtp_server: "localhost"
  smtp_username: "username"
  smtp_password: "password"
  smtp_from_address: "nagios@example.com"
  rhmap_admin_email: "root@localhost"
```

4.2.3.3. Configure Front End Components

- SMTP server parameters

The platform sends emails for user account activation, password recovery, form submissions, and other events. Set the following variables as appropriate for your environment:

```
frontend:
```

```
  smtp_server: "localhost"
  smtp_username: "username"
  smtp_password: "password"
  smtp_port: "25"
  smtp_auth: "false"
  smtp_tls: "false"
  email_replyto: "noreply@localhost"
```

- Git External Protocol

The default protocol is **`https`**. Change this to **`http`** if you use a self-signed certificate.

```
frontend:
  git_external_protocol: "https"
```

4.2.4. Running the Playbook to deploy RHMAP Core

To deploy the Core run the following command from **`/opt/rhmap/4.7/rhmap-installer`**, referencing your own inventory file:

```
ansible-playbook -i my-inventory-file playbooks/core.yml
```

The installer will run through all the tasks required to create the RHMAP Core project. It may take some time for all the Pods to start. Once all Pods are running correctly, you should see output similar to the following:

```
NAME                    READY     STATUS      RESTARTS   AGE
fh-aaa-1-ey0kd          1/1       Running     0          3h
fh-appstore-1-ok76a     1/1       Running     0          6m
fh-messaging-1-isn9f    1/1       Running     0          3h
fh-metrics-1-cnfxm      1/1       Running     0          3h
fh-ngui-1-mosqj         1/1       Running     0          6m
fh-scm-1-c9lhd          1/1       Running     0          3h
fh-supercore-1-mqgph    1/1       Running     0          3h
gitlab-shell-1-wppga    2/2       Running     0          3h
memcached-1-vvt7c       1/1       Running     0          4h
millicore-1-pkpwv       3/3       Running     0          6m
mongodb-1-1-fnf7z       1/1       Running     0          4h
mysql-1-iskrf           1/1       Running     0          4h
nagios-1-mtg31          1/1       Running     0          5h
redis-1-wwxzw           1/1       Running     0          4h
ups-1-mdnjt             1/1       Running     0          4m
```

Once all Pods are running the Ansible installer will run a Nagios checks against the RHMAP Core project to ensure it is healthy.

You can also access and view the Nagios dashboard at this point. The status of these checks can be useful if something has gone wrong during installation and needs troubleshooting.

To access Nagios, follow the Accessing the Nagios Dashboard section in the Operations Guide.

See the Troubleshooting guide if any Pods are not in the correct state or the installation has failed prior to this.

4.2.5. Verifying The Installation

1. Log in to the Studio
   To retrieve the **URL** for the Core Studio, use the following command:

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

   The Admin username and password are set in the **millicore DeploymentConfig**. To view them use this command:

   ```
   oc env dc/millicore --list| grep FH_ADMIN
   ```

   ```
   FH_ADMIN_USER_PASSWORD=password
   FH_ADMIN_USER_NAME=rhmap-admin@example.com
   ```

   See the Troubleshooting guide if you are unable to login to the Studio.

4.3. POST-INSTALLATION STEPS

- Adjusting System Resource Usage of the Core - Adjust the system resource usage of Core components as appropriate for your production environment.
- Optional: Set up centralized logging - see the Enabling Centralized Logging section for imformation about how to deploy a centralized logging solution based on ElasticSearch, Fluentd, and Kibana.

CHAPTER 5. PROVISIONING AN RHMAP 4.X MBAAS ON OPENSHIFT CONTAINER PLATFORM

5.1. OVERVIEW

An OpenShift Container Platform cluster can serve as an MBaaS target and host your Cloud Apps and Cloud Services. This guide provides detailed steps to deploy the RHMAP 4.x MBaaS on an OpenShift Container Platform cluster.

5.2. PREREQUISITES

This guide assumes several prerequisites are met before the installation:

- Ansible version 2.4 is installed on a management node which has SSH access to the OpenShift cluster. See Section 2.2, “Configure Ansible for installing RHMAP components.” for more information.
- All nodes in the cluster must be registered with the Red Hat Subscription Manager. See Chapter 2, *Preparing Infrastructure for Installation* for detailed steps.
- The MBaaS requires outbound internet access to perform npm installations, make sure that all relevant nodes have outbound internet access before installation.
- An existing OpenShift Container Platform installation, version 3.7, 3.9 or 3.10.
- The OpenShift Container Platform master and router must be accessible from the RHMAP Core.
- A wildcard DNS entry must be configured for the OpenShift Container Platform router IP address.
- A trusted wildcard certificate must be configured for the OpenShift Container Platform router. See Using Wildcard Certificates in OpenShift Container Platform documentation.
- Image streams and images in the **`openshift`** namespace must be updated to the latest version. Refer to sections Updating the Default Image Streams and Templates and Importing the Latest Images in the OpenShift Container Platform Installation and Configuration guide.
- You must have administrative access to the OpenShift cluster using the **oc** CLI tool, enabling you to:
  - Create a *project*, and any resource typically found in a project (for example, *deployment configuration*, *service*, *route*).
  - Edit a *namespace* definition.
  - Create a *security context constraint*.
  - Manage nodes, specifically *labels*.
- The rhmap-installer will run a number of pre-req checks which must pass before proceeding with the installation. See RHMAP Installer Pre-Requisite Checks for details.

For information on installation and management of an OpenShift Container Platform cluster and its users, see the official OpenShift documentation.

5.3. INSTALLATION

The installation of an three-node MBaaS in OpenShift Container Platform results in a resilient three-node cluster:

- MBaaS components are spread across all three nodes.
- MongoDB replica set is spread over three nodes.
- MongoDB data is backed by persistent volumes.
- A Nagios service with health checks and alerts is set up for all MBaaS components.

The installation consists of several phases. Before the installation, you must prepare your OpenShift Container Platform cluster:

- Set up persistent storage - you need to create Persistent Volumes with specific parameters in OpenShift Container Platform.
- Label the nodes - nodes need to be labeled in a specific way, to match the node selectors expected by the OpenShift Container Platform template of the MBaaS.
- Network Configuration - configuring the SDN network plugin used in OpenShift Container Platform so that Cloud Apps can communicate with MongoDB in the MBaaS.

After the OpenShift Container Platform cluster is properly configured:

- Install the MBaaS from a template
- Verify the installation

5.3.1. Before The Installation

The installation procedure poses certain requirements on your OpenShift Container Platform cluster in order to guarantee fault tolerance and stability.

5.3.1.1. Network Configuration

Cloud Apps in an MBaaS communicate directly with a MongoDB replica set. In order for this to work, the OpenShift Container Platform SDN must be configured to use the **`ovs-subnet`** SDN plugin. For more detailed information on configuring this, see Migrating Between SDN Plug-ins in the OpenShift documentation.

5.3.1.1.1. Making Project Networks Global

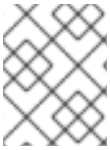
If you cannot use the **`ovs-subnet`** SDN plugin, you must make the network of the MBaaS project global after installation. For example, if you use the **`ovs-multitenant`** SDN plugin, projects must be configured as global. The following command is an example of how to make a project global:

```
oadm pod-network make-projects-global live-mbaas
```

To determine if projects are global, use the following command:

```
oc get netnamespaces
```

In the output, any projects that are configured global have namespaces with a value of "0"



NOTE

If a project network is configured as global, you cannot reconfigure it to reduce network accessibility.

For further information on how to make projects global, see Making Project Networks Global in the OpenShift Container Platform documentation.

5.3.1.2. Persistent Storage Setup

NOTE

Ensure that the persistent volumes are configured according to the OpenShift documentation for configuring PersistentVolumes. If you are using NFS, see the Troubleshooting NFS Issues section for more information.

Some components of the MBaaS require persistent storage. For example, MongoDB for storing databases, and Nagios for storing historical monitoring data.

As a minimum, make sure your OpenShift Container Platform cluster has the following persistent volumes in an **`Available`** state, with at least the amount of free space listed below:

- Three **50 GB** persistent volumes, one for each MongoDB replica
- One **1 GB** persistent volume for Nagios

For detailed information on PersistentVolumes and how to create them, see Persistent Storage in the OpenShift Container Platform documentation.

5.3.1.3. Apply Node Labels for MBaaS

By applying labels to OpenShift Container Platform nodes, you can control which nodes the MBaaS components, MongoDB replicas and Cloud Apps will be deployed to.

This section describes the considerations for:

- Section 5.3.1.3.1, “Labelling for MBaaS components”
- Section 5.3.1.3.2, “Labelling for MongoDB replicas”

Cloud apps get deployed to nodes labeled with the default **`nodeSelector`**, which is usually set to **`type=compute`** (defined in the OpenShift Container Platform master configuration).

5.3.1.3.1. Labelling for MBaaS components

Red Hat recommends that MBaaS components are deployed to dedicated nodes and that these nodes are separated from other applications, for example, RHMAP Cloud Apps.

Refer to Infrastructure Sizing Considerations for Installation of RHMAP MBaaS for the recommended number of MBaaS nodes and Cloud App nodes for your configuration.

For example, if you have 12 nodes, the recommendation is:

- Dedicate three nodes to MBaaS and MongoDB.
- Dedicate three nodes to Cloud Apps.

To achieve this, apply a label, such as **`type=mbaas`** to the three dedicated MBaaS nodes.

```
oc label node mbaas-1 type=mbaas
oc label node mbaas-2 type=mbaas
oc label node mbaas-3 type=mbaas
```

Then, when creating the MBaaS project, as described later in Section 5.3.2, “Installing the MBaaS”, set this label as the **`nodeSelector`**.

You can check what **`type`** labels are applied to all nodes with the following command:

```
oc get nodes -L type
```

```
NAME          STATUS                        AGE       TYPE
ose-master    Ready,SchedulingDisabled      27d       master
infra-1       Ready                         27d       infra
infra-2       Ready                         27d       infra
app-1         Ready                         27d       compute
app-2         Ready                         27d       compute
app-3         Ready                         27d       compute
mbaas-1       Ready                         27d       mbaas
mbaas-2       Ready                         27d       mbaas
mbaas-3       Ready                         27d       mbaas
```

In this example, the deployment would be as follows:

- Cloud apps get deployed to the three dedicated Cloud App nodes **`app-1`**, **`app-2`**, and **`app-3`**.
- The MBaaS components get deployed to the three dedicated MBaaS nodes **`mbaas-1`**, **`mbaas-2`**, and **`mbaas-3`** (if the **`nodeSelector`** is also set on the MBaaS Project).

5.3.1.3.2. Labelling for MongoDB replicas

In the production MBaaS template, the MongoDB replicas are spread over three MBaaS nodes. If you have more than three MBaaS nodes, any three of them can host the MongoDB replicas.

To apply the required labels (assuming the three nodes are named **`mbaas-1`**, **`mbaas-2`**, and **`mbaas-3`**):

```
oc label node mbaas-1 mbaas_id=mbaas1
oc label node mbaas-2 mbaas_id=mbaas2
oc label node mbaas-3 mbaas_id=mbaas3
```

You can verify the labels were applied correctly by running this command:

```
oc get nodes -L mbaas_id
```

```
NAME          STATUS                        AGE       MBAAS_ID
10.10.0.102   Ready                         27d       <none>
10.10.0.117   Ready                         27d       <none>
```

```
10.10.0.141   Ready                      27d       <none>
10.10.0.157   Ready                      27d       mbaas3
10.10.0.19    Ready,SchedulingDisabled   27d       <none>
10.10.0.28    Ready                      27d       mbaas1
10.10.0.33    Ready                      27d       <none>
10.10.0.4     Ready                      27d       <none>
10.10.0.99    Ready                      27d       mbaas2
```

See Updating Labels on Nodes in the OpenShift Container Platform documentation for more information on how to apply labels to nodes.

5.3.1.3.2.1. Why are MongoDB replicas spread over multiple nodes?

Each MongoDB replica is scheduled to a different node to support failover.

For example, if an OpenShift Container Platform node failed, data would be completely inaccessible if all three MongoDB replicas were scheduled on this failing node. Setting a different **`nodeSelector`** for each MongoDB **`DeploymentConfig`**, and having a corresponding OpenShift Container Platform node in the cluster matching this label will ensure the MongoDB Pods get scheduled to different nodes.

In the production MBaaS template, there is a different **`nodeSelector`** for each MongoDB **`DeploymentConfig`**:

- **`mbaas_id=mbaas1`** for **`mongodb-1`**
- **`mbaas_id=mbaas2`** for **`mongodb-2`**
- **`mbaas_id=mbaas3`** for **`mongodb-3`**

5.3.2. Installing the MBaaS

5.3.2.1. Setting Variables

The variables required for installation of RHMAP MBaaS are set in the following file:

```
/opt/rhmap/4.7/rhmap-installer/roles/deploy-mbaas/defaults/main.yml
```

Set up the monitoring parameters with SMTP server details, which are required to enable email alerting from Nagios. If you do not require email alerting or want to set it up at a later time, the sample values can be used.

```
monitoring:
  smtp_server: "localhost"
  smtp_username: "username"
  smtp_password: "password"
  smtp_from_address: "nagios@example.com"
  rhmap_admin_email: "root@localhost"
```

5.3.2.2. Run the Playbook

To provision a 1-node MBaaS, enter:

```
ansible-playbook -i my-inventory-file playbooks/1-node-mbaas.yml
```

To provision a 3-node MBaaS, enter:

```
ansible-playbook -i my-inventory-file playbooks/3-node-mbaas.yml
```

5.3.3. Verifying The Installation

1. Ping the health endpoint.
   If all services are created, all Pods are running, and the route is exposed, the MBaaS health endpoint can be queried as follows:

   ```
   curl `oc get route mbaas --template "
   {{.spec.host}}"`/sys/info/health
   ```

   The endpoint responds with health information about the various MBaaS components and their dependencies. If there are no errors reported, the MBaaS is ready to be configured for use in the Studio. Successful output will resemble the following:

   ```
   {
     "status": "ok",
     "summary": "No issues to report. All tests passed without error",
     "details": [
       {
         "description": "Check Mongodb connection",
         "test_status": "ok",
         "result": {
           "id": "mongodb",
           "status": "OK",
           "error": null
         },
         "runtime": 33
       },
       {
         "description": "Check fh-messaging running",
         "test_status": "ok",
         "result": {
           "id": "fh-messaging",
           "status": "OK",
           "error": null
         },
         "runtime": 64
       },
       {
         "description": "Check fh-metrics running",
         "test_status": "ok",
         "result": {
           "id": "fh-metrics",
           "status": "OK",
           "error": null
         },
         "runtime": 201
       },
       {
         "description": "Check fh-statsd running",
         "test_status": "ok",
         "result": {
   ```

```
        "id": "fh-statsd",
        "status": "OK",
        "error": null
      },
      "runtime": 7020
    }
  ]
}
```

2. Verify that all Nagios checks are passing.
   Log in to the Nagios dashboard of the MBaaS by following the steps in the Accessing the Nagios Dashboard section in the Operations Guide.

   After logging in to the Nagios Dashboard, all checks under the left-hand-side **Services** menu should be indicated as **OK**.

See the Troubleshooting guide if any of the checks are not in an **OK** state.

After verifying that the MBaaS is installed correctly, you must create an MBaaS target for the new MBaaS in the Studio.

5.4. CREATING AN MBAAS TARGET

1. In the Studio, navigate to the *Admin* > *MBaaS Targets* section. Click *Create MBaaS Target*.
2. Enter the following information

   - **MBaaS Id** - a unique ID for the MBaaS, for example: **`live-mbaas`**.
   - **OpenShift Master URL** - the URL of the OpenShift Container Platform master, for example, **`https://master.openshift.example.com:8443`**.
   - **OpenShift Router DNS** - a wildcard DNS entry of the OpenShift Container Platform router, for example, **`*.cloudapps.example.com`**.
   - **MBaaS Service Key**
     Equivalent to the value of the **`FHMBAAS_KEY`** environment variable, which is automatically generated during installation. To find out this value, run the following command:

     ```
     oc env dc/fh-mbaas --list | grep FHMBAAS_KEY
     ```

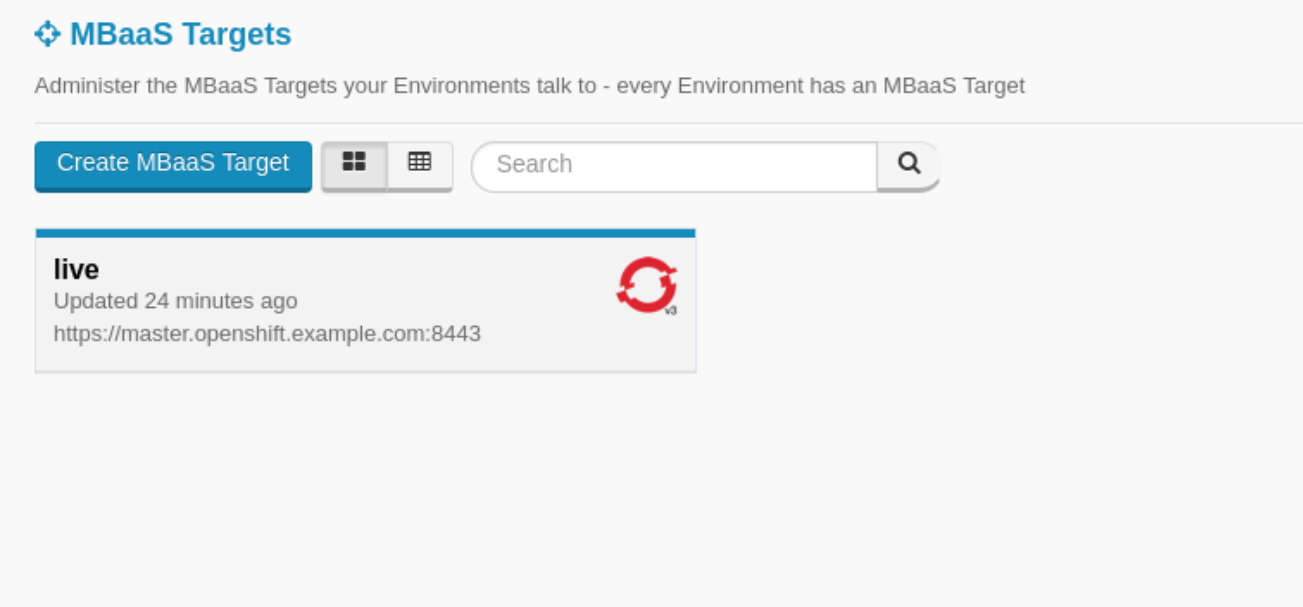
     Alternatively, you can find the value in the OpenShift Container Platform Console, in the *Deployment config* of the **`fh-mbaas`**, in the *Environment* section.
   - **MBaaS URL**
     A URL of the route exposed for the **`fh-mbaas-service`**, including the *https* protocol prefix. This can be retrieved from the OpenShift Container Platform web console, or by running the following command:

     ```
     echo "https://"$(oc get route/mbaas -o template --template
     {{.spec.host}})
     ```

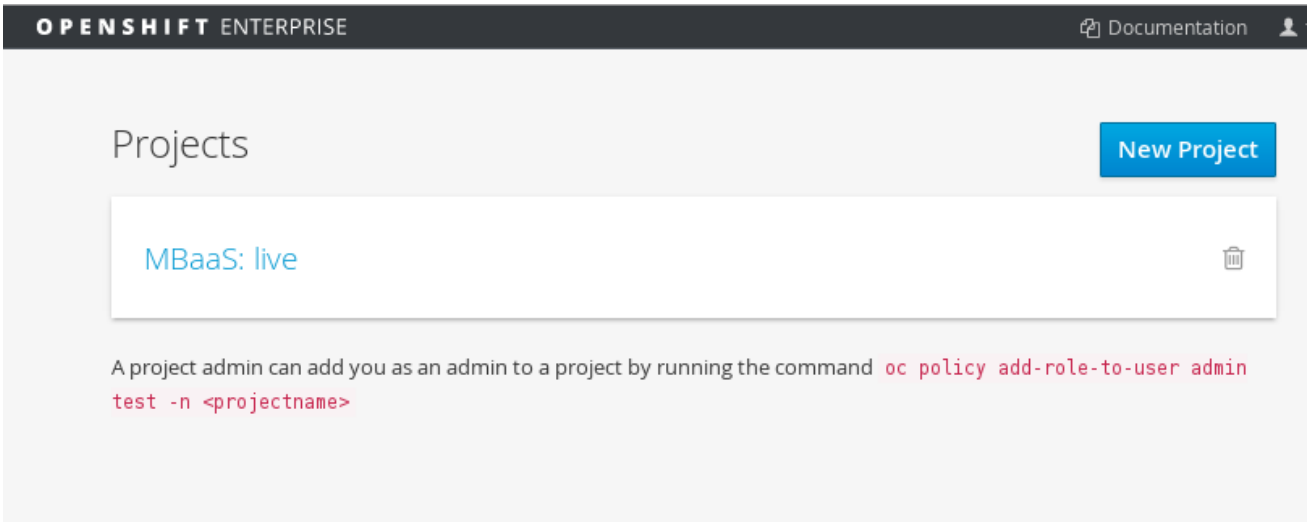
- **MBaaS Project URL** - (Optional) URL where the OpenShift Container Platform MBaaS project is available e.g. https://mbaas-mymbaas.openshift.example.com:8443/console/project/my-mbaas/overview.
- **Nagios URL** - (Optional) Exposed route where Nagios is running in OpenShift Container Platform e.g. https://nagios-my-mbaas.openshift.example.com.

3. Click *Save MBaaS* and you will be directed to the MBaaS Status screen. The status should be reported back in less than a minute.

Once the process of creating the MBaaS has successfully completed, you can see the new MBaaS in the list of MBaaS targets.



In your OpenShift Container Platform account, you can see the MBaaS represented by a project.



5.5. AFTER INSTALLATION

- Create an Environment - you must create at least one environment for the MBaaS to be usable by Cloud Apps and Cloud Services

- Adjusting System Resource Usage of the MBaaS and Cloud Apps - Red Hat recommends that you adjust the system resource usage of MBaaS components as appropriate for your production environment

CHAPTER 6. POST-INSTALLATION TASKS

After installing the Core and the MBaaS, you can enable several features to access all functionality of the RHMAP cluster:

- Set up centralized logging.
  - Enabling Centralized Logging
- Set up monitoring.
  - Monitoring RHMAP with Cockpit
  - Monitoring RHMAP with Nagios

Core

- Set up email configuration for the Core.
  - Modifying SMTP Server Setup in the Core

MBaaS

- Enable the MBaaS and Cloud Apps to make use of all available system resources.
  - Adjusting System Resource Usage of the MBaaS and Cloud Apps
- Set up email configuration for the MBaaS.
  - Setting Up SMTP for Cloud App Alerts

APPENDIX A. RHMAP ANSIBLE PLAYBOOK PRE-REQUISITE CHECKS

When you run an RHMAP Ansible Playbook, it performs the following checks.

A.1. PREREQUISITE CHECKS

For every component, the installer attempts to check for a wildcard certificate, the correct PV requirement and also checks:

Description	Parameter Name	Default Value	Fail/warning/recommended
Outbound Internet Connection	url_to_check	https://www.npmjs.com	fail only in strict_mode

A.2. CHECKS ON RHMAP CORE

Description	Parameter Name	Default Value	Fail/warning/recommended
Min number of CPUs	`min_required_vCPUS`	4	fail only in strict_mode
Min system memory per node (in MB)	`required_mem_mb_threshold`	7000	fail only in strict_mode
Min total free memory of all nodes (in KB)	`warning_kb_value`	4000000	warning

A.3. CHECKS ON 1 NODE MBAAS

Description	Parameter name	Default value	Fail/warning/recommended
Min number of CPUs	`min_required_vCPUS`	2	fail only in strict_mode
Min system memory per node (in MB)	`required_mem_mb_threshold`	7000	fail only in strict_mode
Min total free memory of all nodes (in KB)	`warning_kb_value`	4000000	warning

A.4. CHECKS ON 3 NODE MBAAS

Description	Parameter name	Default value	Fail/warning/recommended
Min number of CPUs	`min_required_vCPUS`	2	fail only in strict_mode
Min system memory per node (in MB)	`required_mem_mb_threshold`	7000	fail only in strict_mode
Min total free memory of all nodes (in KB)	`warning_kb_value`	4000000	warning