IBM ECM System Monitor
Field Guides

Deploying ESM on OpenShift

August 29th, 2023

CENIT AG
Michael Wohland

Content

Introduction

This guide will use IBM RedHat OpenShift Kubernetes Service (ROKS) to showcase the deployment procedure on an OpenShift 4.x cluster. If you are working on an OpenShift cluster in a private data center, the following topics are not covered by this guide:

- Authenticating to the API of your private cluster with the **oc** command line tool.
- Providing container images through a private registry.
- Configuring DNS and load balancers for accessing the ESM UI from outside the cluster.

The deployment is performed using a Helm Chart. An ESM instance installed with Helm (a “Helm Release”) can consist of:

- 0 or 1 ESM Servers
- 0 or 1 ESM Agents

For most scenarios, a single release consisting of 1 ESM Server and 1 ESM Agent is sufficient for typical environments to be monitored within a cluster.

Prerequisites

ECM System Monitor

This guide is for customers running IBM ECM System Monitor (ESM). The container images and the Helm Chart are continously improved on each release of System Monitor. As far as possible, this guide will be continously updated to reflect these changes.

OpenShift

ECM System Monitor is supported on OpenShift (OCP) 4.x only. OpenShift 3.x versions are not supported.

Client Tools

- **`helm`** needs to be available (v3+). Helm 2.x is not supported.
- A version of the **`oc`** command line tool compatible with your OCP cluster's version is recommended.[1]
- To push the required container images to a Container Registry (such as the "IBM Container Registry"), you may use your preferred container runtime (such as **`docker`** or **`podman`**) or other tools (**`buildah`** or **`skopeo`**). In this document we will provide instructions for **`docker`**.
- The IBM Cloud CLI[2] (**`ibmcloud`**) will be required to interact with the IBM Container Registry. If you are using another registry (i.e. a private registry within the OCP cluster or another private registry on another cloud platform), the IBM Cloud CLI is not required.

[1] The latest version of **`oc`** can be found here: https://mirror.openshift.com/pub/openshift-v4/clients/oc/latest/

[2] How to setup the IBM Cloud CLI: https://cloud.ibm.com/docs/cli?topic=cli-getting-started

Downloads

Add Helm Chart Repository

The Helm Chart needs to be available on the system you are invoking client commands from (i.e. **`oc`** or **`helm`**). You can get the chart by adding the Helm repository:

`cenit-ag.github.io/helm-charts`

Run the following commands to add the repository to your Helm environment:

```
# Add repository
helm repo add cenit https://cenit-ag.github.io/helm-charts

# Update the repository index
helm repo update
```

Available versions of the chart can be listed with a search:

```
helm search repo cenit
```

Output:

```
NAME            CHART VERSION   […]
cenit/sm         1.1.7          […]
```

If the client you will be sending your commands from does not have an internet connection, you can pull the chart:

```
helm pull cenit/sm --version 1.1.7
```

This will download the chart as a .tgz file which can be used for an offline installation.

Notice: The `--version` flag is optional. If no version is stated, always the latest version will be downloaded.

The Helm chart repository also included extensive chart documentation. See following as an example for a specific chart version (`v1.1.7`):

https://github.com/cenit-ag/helm-charts/blob/main/sm/docs/sm-1.1.7.md

Download ESM Container Images

As for now, the required container images are only available as download from the IBM Software Repository using your entitled IBM account to download the container archives. They can be downloaded from

- IBM Password Advantage (major release versions)
- IBM Fix Central (fixpack versions)

Although IBM Fix Central only provides ESM fixpack versions, these containers are fully functional and do not require a container of a major release version.

For the present guide we will use the ESM Version 5.5.11.0-000

The Docker Containers for 5.5.11.0-000 have the following part numbers:

M0DRYML for the Server

M0DRZML for the Agent

For better readability and since the IF packages also use this naming convention, we rename the tar,gz files after download to esmserver.tar.gz and esmagent.tar.gz.

Preparations

Provide Container Images in IBM Container Registry

When using RedHat OpenShift on IBM Cloud, the ESM container images can be provided to the cluster through the IBM Container Registry. This is especially helpful if you want to use custom containers, i.e. equipped with 3rd party libraries or your custom monitoring scripts, instead of the default container images. Use the following steps to push the image to the registry:

1) Setup access to the registry follow the related quickstart documentation. This requires authentication with an IBM Cloud account.
   Notice: Your IBM Cloud account needs to be member of a Resource Group which has permissions to use the "IBM Container Registry Service".
2) Extract the container images for ESM Server and ESM Agent from the software archive you previously downloaded from IBM.
   - ESM Server: **`esmserver.tgz or esmserver.tar.gz`**
   - ESM Agent: **`esmagent.tgz or esmagent.tar.gz`**
3) Import both images seperately to your local container runtime:

```
docker load --input esmserver.tar
docker load --input esmagent.tar
```

4) Create a namespace for the images in the Container Registry by following the instructions in the IBM Cloud documentation. In this example we will use a private namespace called **`cenit`** in the **`de.icr.io`** registry (this namespace will not be accessible to you).

5) Tag the images for the registry (replace namespace **`cenit`** with your namespace – also use the correct version number of ESM for the tag) :

```
docker tag \
    esmserver:5.5.11.0-000 \
    de.icr.io/cenit/esmserver:5.5.11.0-000

docker tag \
    esmagent:5.5.11.0-000 \
    de.icr.io/cenit/esmagent:5.5.11.0-000
```

6) Login to IBM Cloud Container Registry (as described in the quickstart docs).

7) Push the tagged images:

```
docker push de.icr.io/cenit/esmserver:5.5.11.0-000
docker push de.icr.io/cenit/esmagent:5.5.11.0-000
```

8) List the pushed images:

```
ibmcloud cr image-list
```

Output (truncated):

```
Repository                      Tag           Digest         Size
de.icr.io/cenit/esmagent        5.5.11.0-000   a7941079a4c7   367 MB
de.icr.io/cenit/esmserver       5.5.11.0-000   6d1b52f06af3   281 MB
```

The images are now available in your registry namespace.

Login to the Cluster

For IBM Cloud ROKS request an OAuth token in the Management panel of your ROKS cluster. Example login (token and hostname are obfuscated):

```
oc login \
    --token=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX \
    --server=https://cXXX-e.eu-de.containers.cloud.ibm.com:30416
```

Notice: On Windows, instead of the classic CMD, it is recommended to use the Powershell as later commands in this document will make use of PowerShell-specific capabilities.

Create an OpenShift Project

This guide will use the **oc** CLI for most of the interactions with OCP. At first we need to create an OpenShift project. Alternatively you can use any other project (or “namespace”). In this guide, we will name the namespace **esm**.

```
oc new-project esm
```

Several remarks on namespaces:

- After creating a new namespace, **oc** automatically switches into this namespace. To switch between namespaces use **oc project <namespace>**. To determine which context is currently configured, use **oc project**.
- The chart can be installed multiple times in the same namespace.

Configure Access to IBM Container Registry

By default, on ROKS only the default namespace automatically has access to the Container Registry in the same IBM Cloud account. To enable access to the container images from our just created namespace, a Secret has to be provided containing registry credentials.

The following commands are used to read the existing default secret provided by IBM Cloud in the OCP cluster's default namespace and copy it to your namespace.

Copy Secret With `oc` on Linux

```
NAMESPACE=esm
# --> Replace this with the namespace used for your installation

oc get secret all-icr-io -n default -o yaml \
    | sed "s/default/$NAMESPACE/g" \
    | oc apply -f -
```

Copy Secret With `oc` on Windows

The following commands can be executed in a PowerShell:

```
$NAMESPACE="esm"
# --> Replace this with the namespace used for your installation

oc get secret all-icr-io -n default -o yaml `
    | %{$_ -replace "default", $NAMESPACE} `
    | oc apply -f
```

Notice: Above commands require your OCP login to have access to the default namespace. If you do not have a access to this namespace and cannot get the secret, you have the following alternative options:

a) Have another user with elevated permissions run the command for you.
b) Follow instructions to create image pull secrets on ROKS in the official IBM Cloud documentation.[3]

[3] https://cloud.ibm.com/docs/openshift?topic=openshift-registry#other_registry_accounts

Create RBAC Resources (only =< 5.5.11.0-000)

Notice: This steps is only required in version =< 5.5.11.0-000 as these container images still require privileged access during execution. Privileged execution of containers in disabled by default on OpenShift clusters, thus we need to enable this using a Service Account. If you plan to install a newer version, the Service Account and Admin Policy Changes do not have to be executed.

Create a Service Account:

```
oc create serviceaccount sm-svc-acc
```

Add the created account to the **`privileged`** Security Context Constraint:

```
oc adm policy add-scc-to-user privileged -z sm-svc-acc
```

Notice: Committing policy changes is a cluster-wide change and required elevated permissions.

Create Secret for Default Administrator Password

The ESM Admin UI provides a built-in **`admin`** user for login. The Helm Chart will automatically set a secure password on installation. Set this password by creating a secret:

```
oc create secret generic my-sm-admin-pw --from-literal=password='s3cr3t'
```

Installation

Specify Installation Parameters

The Helm Chart contains an example values file for OpenShift installations. Create a copy of this file and make sure that this file is properly managed in a version control system or a backup is created of this modified file after the installation procedure.

```
cp examples/values/roks_persistent.yaml custom.yaml
```

In the **custom.yaml** values file the installation parameters have to be adjusted to match the present environment:

Parameter	Value
server.image.repository	de.icr.io/cenit/esmserver
server.image.tag	5.5.11.0-000
agent.image.repository	de.icr.io/cenit/esmagent
agent.image.tag	5.5.11.0-000

Please consider the following remarks on other settings in the file:

Parameter	Remark
platform	Must be **ocp** for installations on OpenShift.
server.route.enabled	If **true**, the chart will create a route that will make the ESM Server UI accessible from external. Consider this for security reasons.
server.persistence.enabled	If **true**, the ESM Server's configuration database and collected monitoring data will be persisted to a Persistent Volume.
server.persistence.storageClass	By default, this uses the ReadWriteOnce Storage Class **ibmc-block-bronze**. For other classes on ROKS refer to the official documentation. For non-ROKS OCP clusters (i.e. running on-premises), contact the OCP administrators to find out an appropriate Storage Class.
server.persistence.size	Defines the size of the requested Persistent Volume. Defaults to **20Gi**. This is also the minimum for the default Storage Class **ibmc-block-bronze**.
adminPasswordSecret	A reference to the secret created earlier in this guide (**my-sm-admin-pw**). If you chose another name for the secret, set the name in this attribute (see example in **roks_persistent.yaml**).
imagePullSecrets	A list of secrets used to authenticate against image registries. In our case, this is the previously created pull secret **all-icr-io** (see example in **roks_persistent.yaml**). Modify if you plan to pull from another private registry.

Install Using Helm

Perform the installation using following command:

```
helm install mysm cenit/sm -f custom.yaml
```

- **mysm** is the release name.
- **cenit/sm** points to the chart in the repository we added earlier in this guide. This can also be replaced with the path to a .**tgz** archive containing the chart or a directory with the chart sources from within the .**tgz** archive.
- **custom.yaml** is the values file containing default values prepared in the previous chapters.
- Optionally the flag **--dry-run** can be added in order to simulate the Helm templating proecedure on the client side without committing any changes to the targeted cluster. This is helpful in case you want to validate if your values YAML file (**custom.yaml** in the example) is syntactically correct.

Alternatively, these settings can later on also be passed to the Helm installation using the flag **--set**. [4]

```
helm install mysm cenit/sm -f sm/examples/values/roks_persistent.yaml \
    --set server.image.repository=de.icr.io/cenit/esmserver \
    --set server.image.tag=5.5.11.0-000 \
    --set agent.image.repository=de.icr.io/cenit/esmagent \
    --set agent.image.tag=5.5.11.0-000
```

The command **helm list** will show you all deployed releases. Command **helm status mysm** will give you a current status of the chart including the post-installation notes, which include some helpful commands for further troubleshooting.

Notice: After a successful Helm installation attempt, the “NOTES” passage in the output features several **kubectl** commands to validate the installation. You can replace **kubectl** with **oc** to run these commands without having **kubectl** available on your command line.

[4] Link to Helm documentation: https://helm.sh/docs/helm/helm_install/

Validation

Check Resource States

Check ESM Server for readiness (takes 120 seconds or more):

```
kubectl rollout status statefulset mysm-smserver
```

Check ESM Agent for readiness (takes 120 seconds or more):

```
kubectl rollout status deployment mysm-smagent
```

Check Logs

Check ESM Server logs:

```
kubectl logs mysm-smserver-0 -f
```

Check ESM Agent logs:

```
kubectl logs \
    $(kubectl get pod -n esm -l smagent-instance=mysm \
    -o=jsonpath='{.items[0].metadata.name}') -f
```

Show Pods

After a successful deployment the command **`oc get pod -n esm`** should show you the ESM Server and the ESM Agent Pods running and ready:

```
NAME                              READY   STATUS    RESTARTS   AGE
mysm-smagent-67b754ff8d-jnh42     1/1     Running   0          5m22s
mysm-smserver-0                   1/1     Running   0          5m22s
```

The running Pods can also be shown in the OpenShift Admin Console:

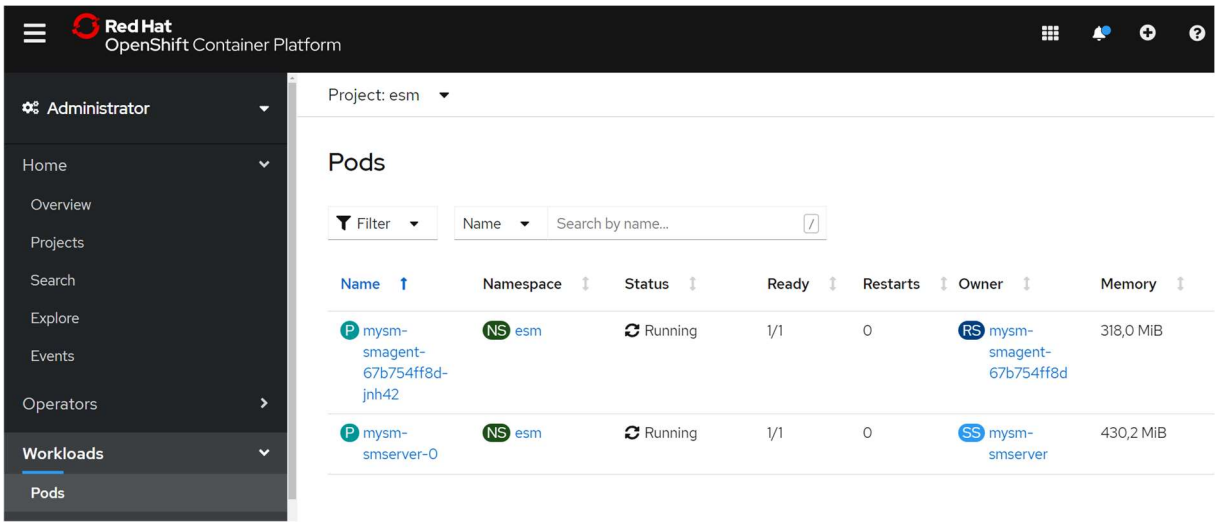

Figure 1: ESM Pods running in OpenShift Cluster

Access ESM Server UI Using Route

The example configuration **`roks_persistent.yaml`** creates a Route that exposes the ESM Server UI through an external HTTPS endpoints:

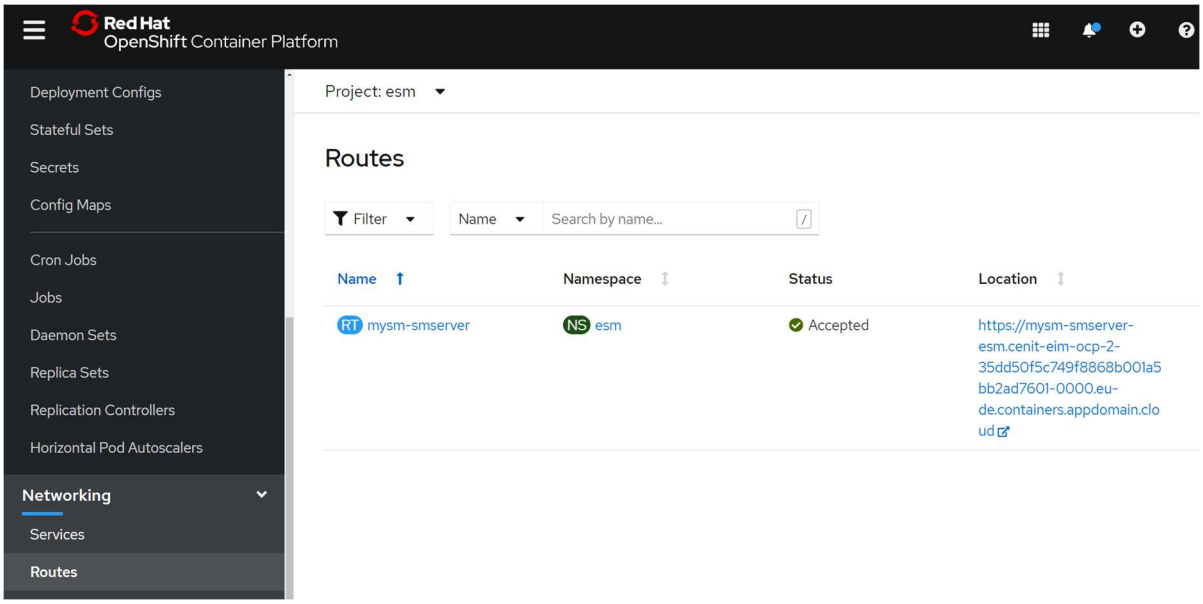

Figure 2: Routes in OpenShift Console

The URL pointing to the service can also be obtained through the **`oc`** command:

```
RELEASE=mysm
echo "https://$(oc get route $RELEASE-smserver -o=jsonpath='{.spec.host}')"
```

This will load the ESM UI's login interface:

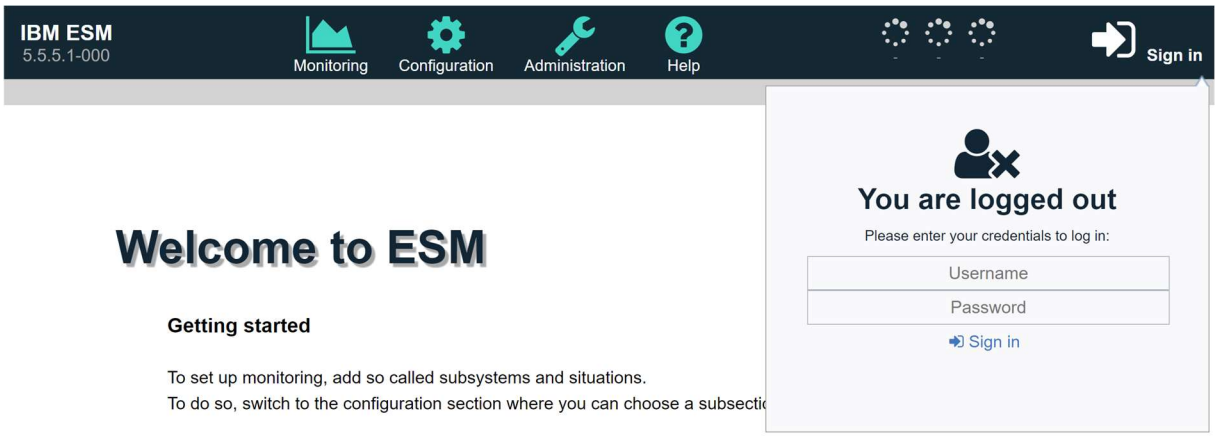

Figure 3: ESM UI Login Interface

To login to the UI, utilize user **`admin`** and the password you defined earlier when creating the Secret **`my-sm-admin-pw`**.

Access ESM Server UI Using Port Forwarding

If your cluster is configured in a way that external access is not permitted or technically impossible, you can alternatively tunnel the access to the ESM Server UI through **`kubectl`** or **`oc`** to your local client machine:

```
oc port-forward -n esm mysm-smserver-0 8443:8443
```

As long as the shell of this running command is kept open, you will be able to access the UI from the machine you executed the command using URL: https://localhost:8443

If no browser is available on the local system, use curl command for a connection test (should return **`200`**):

```
curl -skL -o /dev/null https://localhost:8443 -w "%{http_code}\n"
```

Updates

Updates to newer versions of container images or to change deployment configuration can be accomplished by replacing **`helm install`** with **`helm upgrade`**. With the additional flag named **`--install`**, it will be ensured that **`helm`** performs a new install instead of an update, in case a release with the given name cannot be found. Example:

```
helm upgrade --install mysm sm/ -f sm/examples/values/roks_persistent.yaml \
    --set server.image.repository=de.icr.io/cenit/esmserver \
    --set server.image.tag=5.5.11.0-000 \
    --set agent.image.repository=de.icr.io/cenit/esmagent \
    --set agent.image.tag=5.5.11.0-000
```

Uninstallation

Uninstall the Helm release using:

```
helm uninstall mysm
```

Before running this command, make sure to have the correct namespace context configured (using **`oc project`**). A safer approach is to excplicitly state the namespace in the uninstall command:

```
helm uninstall mysm -n esm
```

This will remove all resources managed by the Helm chart. This does not include:

- Manually created secrets using oc (like the access to the container registry or for the ESM admin password).
- The Persistent Volume of ESM Server.

Although the creation of the ESM Server Pods's Persistent Volume is initiated by the Helm Chart, the chart does not manage the volume, as this is accomplished using Dynamic Volume Provisioning. The positive side-effect of this behavior is that after an accidential uninstallation, all persisteted data (i.e. configuration done in the UI or collected monitoring data) is not automatically deleted, but still retained.

To completely erase all resources related to ESM from the cluster, run the following commands:

```
# Delete PVC (PV should be deleted automatically then)
RELEASE=mysm
oc delete pvc -n esm data-$RELEASE-smserver-0

# Delete admin password secret
oc delete secret -n esm my-sm-admin-pw

# Delete registry access secret
oc delete secret -n esm all-icr-io
```

Appendix

List of Figures

Contact Information

If you have any questions, please contact us at ECM.SystemMonitor@cenit.com.

CENIT AG
Phone: +49 711 7825 30
Email: ECM.SystemMonitor@cenit.com