

## **SEC CP4I v 2020.4.1**

### **Production ~ Installation**

Hakam Abdelqader ([hakam.abdelqader@ibm.com](mailto:hakam.abdelqader@ibm.com))

# Table of Contents

---

- 1. Openshift Production Configuration**
    - 1.1 Labeling nodes
    - 1.2 Taint Nodes
    - 1.3 Troubleshoot **Pending** pods
    - 1.4 Machine Config Pool and Machine Config Configuration
    - 1.5 Status of MCP
    - 1.6 Troubleshoot MachineConfigPool (MCP stuck in updating status)
      - 1.6.1 Resolution
    - 1.7 Moving Workloads to infra
      - 1.7.1 Ingress Controller
      - 1.7.2 Router
      - 1.7.3 Default Registry
      - 1.7.4 Monitoring
  - 2. Prerequisites (Mirroring CP4I operators and images)**
    - 2.1 Introduction
    - 2.2 Prepare a Docker registry
    - 2.3 Prepare a bastion host
    - 2.4 Create environment variables for the installer and image inventory
    - 2.5 Mirroring individual components
    - 2.6 Download the installer and image inventory
      - 2.6.1 Log in to OpenShift cluster as a cluster administrator
      - 2.6.2 Create a Kubernetes namespace
      - 2.6.3 Mirror the images and configure the cluster
      - 2.6.4 Create the CatalogSource
  - 3. Cloud Pak for Integration Platform Navigator**
  - 4. EventStreams**
  - 5. API Connect**
    - 5.1 Install API Connect
    - 5.2 Configure API Connect
      - 5.2.1 Configure a dummy email client
  - 6. App Connect**
  - 7. MQ**
- 

## Openshift Production Configuration

In this section we want to segregate the infra and storage nodes so workloads won't schedule on those nodes. We will be labeling, adding taints, schedule relevant pods to schedule on those node and removing any workloads that should be used up by worker nodes.

### Labeling nodes

1. Label **storage** nodes with `node-role.kubernetes.io/storage=''`

```

1 $ oc label node r.h-smoc-cloud-ocs3.ocpcluster.secsmoc.local node-
  role.kubernetes.io/storage=''
2 # node/rh-smoc-cloud-ocs3.ocpcluster.secsmoc.local labeled
3 $ oc label node rh-smoc-cloud-ocs2.ocpcluster.secsmoc.local node-
  role.kubernetes.io/storage=''
4 # node/rh-smoc-cloud-ocs2.ocpcluster.secsmoc.local labeled
5 $ oc label node rh-smoc-cloud-ocs1.ocpcluster.secsmoc.local node-
  role.kubernetes.io/storage=''
6 # node/rh-smoc-cloud-ocs1.ocpcluster.secsmoc.local labeled

```

## 2. Label **infra** nodes with `node-role.kubernetes.io/infra=''`

```

1 $ oc label node rh-smoc-cloud-infra1.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra=''
2 # node/rh-smoc-cloud-infra1.ocpcluster.secsmoc.local labeled
3 $ oc label node rh-smoc-cloud-infra2.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra=''
4 # node/rh-smoc-cloud-infra2.ocpcluster.secsmoc.local labeled
5 $ oc label node rh-smoc-cloud-infra3.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra=''
6 # node/rh-smoc-cloud-infra3.ocpcluster.secsmoc.local labeled

```

## Taint Nodes

Adding a **NoSchedule** OpenShift Container Storage taint is also required so that the **infra** node will only schedule OpenShift Container Storage resources and repel any other non-OpenShift Container Storage workloads.

### Storage

```

1 $ oc adm taint node <node> node.ocs.openshift.io/storage="true":NoSchedule
1 $ oc adm taint node rh-smoc-cloud-ocs1.ocpcluster.secsmoc.local
  node.ocs.openshift.io/storage="true":NoSchedule
2 # node/rh-smoc-cloud-ocs1.ocpcluster.secsmoc.local tainted
3 $ oc adm taint node rh-smoc-cloud-ocs2.ocpcluster.secsmoc.local
  node.ocs.openshift.io/storage="true":NoSchedule
4 # node/rh-smoc-cloud-ocs2.ocpcluster.secsmoc.local tainted
5 $ oc adm taint node rh-smoc-cloud-ocs3.ocpcluster.secsmoc.local
  node.ocs.openshift.io/storage="true":NoSchedule
6 # node/rh-smoc-cloud-ocs3.ocpcluster.secsmoc.local tainted
7

```

### Infra

```

1 $ oc adm taint node <node> node-role.kubernetes.io/infra=''
1 $ oc adm taint node rh-smoc-cloud-infra1.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra='':NoSchedule
2 # node/rh-smoc-cloud-infra1.ocpcluster.secsmoc.local tainted
3 $ oc adm taint node rh-smoc-cloud-infra2.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra='':NoSchedule
4 # node/rh-smoc-cloud-infra2.ocpcluster.secsmoc.local tainted
5 $ oc adm taint node rh-smoc-cloud-infra3.ocpcluster.secsmoc.local node-
  role.kubernetes.io/infra='':NoSchedule
6 # node/rh-smoc-cloud-infra3.ocpcluster.secsmoc.local tainted
7

```

## Troubleshoot **Pending** pods

### Problem

When you have a pod stuck in **Pending** state check the events for the following message:

```
1 | 0/23 nodes are available: 1 node(s) had taint {node-role.kubernetes.io/infra: }, that the pod didn't tolerate, 22 node(s) didn't match node selector.
```

### Resolution

```
1 | $ oc edit pod <pod-name> -n <namespace>
```

Add the following lines to the toleration section depending on what taint was described in the events:

```
1 spec:
2   tolerations:
3     ... #omitted (other tolerations configured)
4   # add the lines below (toleration section should already exists)
5   - key: node-role.kubernetes.io/infra # add key matching error in events
6     operator: Exists
7     effect: NoSchedule
```

## Machine Config Pool and Machine Config Configuration

Create two yaml files to create **MachineConfigPools**

### Infra

```
1 apiVersion: machineconfiguration.openshift.io/v1
2 kind: MachineConfigPool
3 metadata:
4   name: infra
5 spec:
6   machineConfigSelector:
7     matchExpressions:
8       - {key: machineconfiguration.openshift.io/role, operator: In, values:
[worker,infra]}
9   maxUnavailable: null
10  nodeSelector:
11    matchLabels:
12      node-role.kubernetes.io/infra: ""
13  paused: false
```

### Storage

```
1 apiVersion: machineconfiguration.openshift.io/v1
2 kind: MachineConfigPool
3 metadata:
4   name: storage
5 spec:
6   machineConfigSelector:
7     matchExpressions:
8       - {key: machineconfiguration.openshift.io/role, operator: In, values:
[storage]}
```

```

9  |     maxUnavailable: 0
10 |     nodeSelector:
11 |       matchLabels:
12 |         node-role.kubernetes.io/storage: ""
13 |     paused: false
14 |

1 | $ oc create -f mcp-infra.yaml -f mcp-storage.yaml

```

To check the status of machineconfigpool. **MachineCount** should equal **ReadyMachineCount**

```

1 | $ oc get mcp # machineconfigpool
2 | NAME      CONFIG                                     UPDATED
3 | UPDATING   DEGRADED    MACHINECOUNT   READY MACHINECOUNT   UPDATED MACHINECOUNT
4 | DEGRADED MACHINECOUNT AGE
5 |
6 | infra     rendered-infra-75dafd4d626a5109120a08a059e8158e   True   False
7 |           False   3          3          3          0
8 |           14d
9 | master     rendered-master-0cd9beec62ccc05897c12aa052328e83   True   False
10 |           False  3          3          3          0
11 |           47d
12 | storage    rendered-storage-75dafd4d626a5109120a08a059e8158e  True   False
13 |           False  3          3          3          0
14 |           14d
15 | worker     rendered-worker-75dafd4d626a5109120a08a059e8158e  True   False
16 |           False  14         14         14         0
17 |           47d
18 |
19 | $ oc get mc # machineconfig

```

### Final output

```

1 | $ oc get nodes -l node-role.kubernetes.io/infra
2 | NAME          STATUS  ROLES   AGE  VERSION
3 |
4 | rh-smoc-cloud-infra1.ocp...  Ready   infra   25d  v1.19.0+f173eb4
5 | rh-smoc-cloud-infra2.ocp...  Ready   infra   25d  v1.19.0+f173eb4
6 | rh-smoc-cloud-infra3.ocp...  Ready   infra   25d  v1.19.0+f173eb4
7 |
8 | $ oc get nodes -l node-role.kubernetes.io/storage
9 | rh-smoc-cloud-ocs1.ocp...    Ready   storage  25d  v1.19.0+f173eb4
10 | rh-smoc-cloud-ocs2.ocp...    Ready   storage  25d  v1.19.0+f173eb4
11 | rh-smoc-cloud-ocs3.ocp...    Ready   storage  25d  v1.19.0+f173eb4
12

```

## Status of MCP

Troubleshoot any problem with Nodes in **Not Ready** state

```

1 | $ oc get node -o custom-
2 | columns=STATE:metadata.annotations.machineconfiguration\\.openshift\\.io/state
3 | ,DESIRED:metadata.annotations.machineconfiguration\\.openshift\\.io/desiredCon
4 | fig,CURRENT:metadata.annotations.machineconfiguration\\.openshift\\.io/current
5 | Config,REASON:metadata.annotations.machineconfiguration\\.openshift\\.io/reaso
6 | n

```

```
1 | $ oc get nodes -o jsonpath='{range .items[*]}{@.metadata.name}{\'\t\t\'}\n|   {@.metadata.annotations.machineconfiguration\\.openshift\\.io/state}{\'\n\'}{end}\n|   grep -v Done
```

#### Reference:

- 1. <https://access.redhat.com/solutions/5055931>

Drain nodes of all the workloads

```
1 | $ oc cordon <node-name> # ScheduleDisabled\n2\n1 | $ oc adm drain <node-name> --force --delete-local-data --ignore-daemonsets\n2\n1 evicting pod openshift-storage/rook-ceph-mon-a-6f7c6dbdb4-m5fnc\n2 evicting pod openshift-storage/rook-ceph-osd-0-7f78c9dbbf-29vk9\n3 evicting pod openshift-storage/rook-ceph-osd-4-695b975576-7srpb\n4 error when evicting pod "rook-ceph-mon-a-6f7c6dbdb4-m5fnc" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n5 error when evicting pod "rook-ceph-osd-0-7f78c9dbbf-29vk9" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n6 error when evicting pod "rook-ceph-osd-4-695b975576-7srpb" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n7 evicting pod openshift-storage/rook-ceph-osd-4-695b975576-7srpb\n8 evicting pod openshift-storage/rook-ceph-mon-a-6f7c6dbdb4-m5fnc\n9 evicting pod openshift-storage/rook-ceph-osd-0-7f78c9dbbf-29vk9\n10 error when evicting pod "rook-ceph-mon-a-6f7c6dbdb4-m5fnc" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n11 error when evicting pod "rook-ceph-osd-0-7f78c9dbbf-29vk9" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n12 error when evicting pod "rook-ceph-osd-4-695b975576-7srpb" (will retry after\n5s): Cannot evict pod as it would violate the pod's disruption budget.\n13\n1 | $ oc uncordon <node-name> # Scheduleable\n\n1 | $ oc scale deploy <pod-name> --replicas=0 -n openshift-storage
```

#### References

- 1. <https://access.redhat.com/solutions/5034771>
- 2. <https://github.com/openshift/machine-config-operator/blob/master/docs/custom-pools.md>  
[https://docs.openshift.com/container-platform/4.6/machine\\_management/creating-infrastructure-machinesets.html#moving-resources-to-infrastructure-machinesets](https://docs.openshift.com/container-platform/4.6/machine_management/creating-infrastructure-machinesets.html#moving-resources-to-infrastructure-machinesets)
- 3. [.html#moving-resources-to-infrastructure-machinesets](#)
- 4. <https://www.redhat.com/en/blog/openshift-container-platform-4-how-does-machine-config-pool-work>

# Troubleshoot MachineConfigPool (MCP stuck in updating status)

MCP stuck in updating status when checked from ocp CLI:

```
1 [ocpuser@ocphost ~]$ oc get mcp
2 NAME      CONFIG          UPDATED    UPDATING   DEGRADED   MACHINECOUNT
3 READYMACHINECOUNT  UPDATEDMACHINECOUNT  DEGRADEDMACHINECOUNT  AGE
4 master    rendered-master-xxxxx  True      False      False      3
5           3                  0          300d
6 worker    rendered-worker-xxxxx  False     True      False      21
7           1                  0          300d
```

Trying to `rsh` into pod return below error:

```
1 [ocpuser@ocpnode ~]$ oc rsh <pod_name> -n <project_name>
2 error: unable to upgrade connection: Unauthorized
```

## Resolution

**Step 1** Find out the node where the MCP is stuck with below command:

```
1 $ oc get nodes -o jsonpath='{range .items[*]}{@.metadata.name}{\'\t\'}
2   {@.metadata.annotations.machineconfiguration.openshift\\.io/state}{\'\n\'}{end}'
3 | grep -v Done
```

**Step 2** Login into the node using SSH and see the logs from `machine-config-daemon` on the node:

```
1 $ crictl ps | grep -i machine
2
3 $ crictl logs <container_id>
   =>Container ID from above command output
```

Below logs observed from `machine-config-daemon` container:

```
1 I0218 14:36:51.075295 2225251 update.go:99] error when evicting pod "mypdb-1"
  (will retry after 5s): Cannot evict pod as it would violate the pod's
  disruption budget.
2 I0218 14:36:56.081506 2225251 update.go:99] error when evicting pod "mypdb-1"
  (will retry after 5s): Cannot evict pod as it would violate the pod's
  disruption budget.
```

**Step 3** Get the `PodDisruptionBudget` YAML definition and observe why it's not able to evict the pod.

Here, PDB `maxUnavailable` is set to 1 and none of the pod is healthy, so it is not able to evict the pod.

```
1 ...
2 ...
3 spec:
4   maxUnavailable: 1
5   selector:
6     matchLabels:
7       app.kubernetes.io/instance: mypdb
8       app.kubernetes.io/name: mypdb
9       component: server
10  status:
11    currentHealthy: 0
12    desiredHealthy: 2
13    disruptionsAllowed: 0
14    expectedPods: 3
15    observedGeneration: 1
```

**Step 4** Patch `maxUnavailable` parameter to an appropriate value:

```
1 |   $ oc patch pdb <pod_name> -n <project_name> --type=merge -p '{"spec": {"maxUnavailable":3}}'
```

**Step 5** Observe if the pod has been rescheduled on another node. If not, delete the pod forcefully. Now the pod should be running on another node.

**Step 6** Wait for 15-20 minute, if `MachineConfigPool` isn't updating delete `machine-config-daemon` pod for the node (where MCP is stuck). Check [Step 1](#) for node name.

**Step 7** After MCP has been applied successfully on all the desired nodes, revert the changes performed in [Step 4](#).

## Moving Workloads to infra

### Ingress Controller

We need to edit it and modify the `spec` section to add a `NodeSelector` stanza in the following format

```
1 | $ oc edit ingresscontroller default -n openshift-ingress-operator -o yaml
```

Replace `spec: {}` with the following - that references the `infra` label to the `spec` section, as shown:

```
1 | spec:
2 |   nodePlacement:
3 |     nodeSelector:
4 |       matchLabels:
5 |         node-role.kubernetes.io/infra: ""
```

Confirm the correct nodes are in the right place by running

```
1 | $ oc get pod -n openshift-ingress -o wide
```

Add tolerations for the pod configurations you want to schedule on the infra node, like router, registry, and monitoring workloads. Add the following code to the `deployment` object specification:

### Router

```
1 | $ oc get deployments -n openshift-ingress
2 |
3 | $ oc edit deployment router-default -n openshift-ingress
```

Place code under `spec.template.spec..tolerations`

```
1 | spec:
2 |   template:
3 |     spec:
4 |       tolerations:
5 |         - effect: NoSchedule
6 |           key: node-role.kubernetes.io/infra
7 |           operator: Exists
```

## Default Registry

View the `config/instance` object:

```
1 | $ oc get configs.imageregistry.operator.openshift.io cluster -o yaml
```

To move the default registry, it is a similar pattern - first by editing the `config/instance` object

```
1 | oc edit config/cluster
```

Add the following in the `spec`

```
1 | nodeSelector:
2 |   node-role.kubernetes.io/infra: ""
```

Verify the registry pod has been moved to the infrastructure node.

1. Run the following command to identify the node where the registry pod is located:

```
1 | $ oc get pods -o wide -n openshift-image-registry
```

2. Confirm the node has the label you specified:

```
1 | $ oc describe node <node_name>
```

Review the command output and confirm that `node-role.kubernetes.io/infra` is in the `LABELS` list.

## Monitoring

To move monitoring, we need to create a ConfigMap

```
1 | apiVersion: v1
2 | kind: ConfigMap
3 | metadata:
4 |   name: cluster-monitoring-config
5 |   namespace: openshift-monitoring
6 | data:
7 |   config.yaml: |+
8 |     alertmanagerMain:
9 |       nodeSelector:
10 |         node-role.kubernetes.io/infra: ""
11 |     prometheusK8s:
12 |       nodeSelector:
13 |         node-role.kubernetes.io/infra: ""
14 |     prometheusOperator:
15 |       nodeSelector:
16 |         node-role.kubernetes.io/infra: ""
17 |     grafana:
18 |       nodeSelector:
19 |         node-role.kubernetes.io/infra: ""
20 |     k8sPrometheusAdapter:
21 |       nodeSelector:
22 |         node-role.kubernetes.io/infra: ""
23 |     kubeStateMetrics:
24 |       nodeSelector:
25 |         node-role.kubernetes.io/infra: ""
26 |     telemeterClient:
27 |       nodeSelector:
28 |         node-role.kubernetes.io/infra: ""
29 |     openshiftStateMetrics:
```

```
30      nodeSelector:  
31          node-role.kubernetes.io/infra: ""
```

Applying the above can be done using `oc create`

```
1 | $ oc create -f cluster-monitoring-configmap.yaml
```

After a few minutes, you can check that the pods are in the correct place with the following command

```
1 | $ oc get pod -n openshift-monitoring -o wide
```

Apply the new config map:

```
1 | $ oc create -f cluster-monitoring-configmap.yaml
```

Watch the monitoring pods move to the new machines:

```
1 | $ watch 'oc get pod -n openshift-monitoring -o wide'
```

If a component has not moved to the `infra` node, delete the pod with this component:

```
1 | $ oc delete pod -n openshift-monitoring <pod>
```

The component from the deleted pod is re-created on the `infra` node.

## References

- 1. [https://docs.openshift.com/container-platform/4.6/machine\\_management/creating-infrastructure-machinesets.html#machineset-creating\\_creating-infrastructure-machinesets](https://docs.openshift.com/container-platform/4.6/machine_management/creating-infrastructure-machinesets.html#machineset-creating_creating-infrastructure-machinesets)
- 2. [https://docs.openshift.com/container-platform/4.6/storage/persistent\\_storage/persistent-storage-ocs.html](https://docs.openshift.com/container-platform/4.6/storage/persistent_storage/persistent-storage-ocs.html)

# Prerequisites (Mirroring CP4I operators and images)

## Introduction

You can make the operators for IBM Cloud Pak for Integration available to a cluster that has no internet connectivity by using a *bastion host*, which is a machine that has connectivity both to the internet and to the cluster in the restricted environment. This process allows the necessary images to be mirrored directly to an internal image registry for use by the cluster.

- Install an OpenShift 4.6 cluster (see [operating-environment]).
- Prepare a Docker registry and make it available. For more information, see [Prepare a Docker registry](#).
- Configure a bastion host. For more information, see [Prepare a bastion host](#).
- `cloudctl` version 3.6.1-2002 or later is required.

## Prepare a Docker registry

A local Docker registry is used to store all images in your restricted environment. You must create such a registry and ensure that it meets the following requirements:

- Supports [Docker Manifest V2, Schema 2](#).
- Is accessible from both the bastion host and your OpenShift cluster nodes.
- Has the username and password of a user who can write to the target registry from the bastion host.
- Has the username and password of a user who can read from the target registry that is on the OpenShift cluster nodes.
- Allows path separators in the image name.

An example of a simple registry is included in [Creating a mirror registry for installation in a restricted network](#) in the OpenShift documentation.

The internal Red Hat OpenShift registry is not compliant with Docker Manifest V2, Schema 2, so it is not suitable for use as a private registry for restricted environments.

Verify that you:

- Have the credentials of a user who can write and create repositories. The bastion host uses these credentials.
- Have the credentials of a user who can read all repositories. The OpenShift cluster uses these credentials.

## Prepare a bastion host

Prepare a bastion host that can access the OpenShift cluster, the local Docker registry, and the internet.

- The bastion host must be on a Linux x86\_64 platform, or any operating system that the IBM Cloud Pak CLI and the OpenShift CLI supports.
- The bastion host `locale` must be set to English.

Complete these steps on your bastion node:

1. Install OpenSSL version 1.11.1 or higher.
2. Install Docker or Podman on the bastion node.

- For example, to install Docker on a Red Hat machine, run these commands:

```
1 | yum check-update  
2 | yum install docker
```

- To install Podman, see [Podman Installation Instructions](#).

3. Install the latest version of the IBM Cloud Pak CLI, using the binary file that is applicable for your platform. The minimum supported version of the binary file is 3.4.4. For a listing of available binaries, see [cloud-pak-cli](#).

1. Download the binary file.

```
1 | wget https://github.com/IBM/cloud-pak-  
cli/releases/latest/download/cloudctl-linux-amd64.tar.gz
```

2. Run the following commands to modify and move the file.

```
1 | chmod 755 cloudctl-linux-amd64.tar.gz  
2 | mv cloudctl-linux-amd64.tar.gz /usr/local/bin/cloudctl
```

3. Confirm that `cloudctl` is installed:

```
1 | cloudctl --help
```

The `cloudctl` usage is displayed.

4. Install the `oc` OpenShift CLI tool. For more information, see [Getting started with the CLI](#) in the Red Hat OpenShift documentation.

5. Install the `skopeo` CLI version `1.0.0` or higher. For more information, see [Installing skopeo from packages](#).

6. Create a directory that serves as the offline store.

Following is an example directory. This example is used in the subsequent steps.

```
1 | mkdir $HOME/offline
```

This offline store must be persistent to avoid transferring data more than once. The persistence also helps to run the mirroring process multiple times or on a schedule.

## Create environment variables for the installer and image inventory

Create the following environment variables with the installer image name and the image inventory.

```
1 | export CASE_ARCHIVE=ibm-cp-integration-2.1.0.tgz
2 | export CASE_INVENTORY_SETUP=operator
```

Using this CASE archive will mirror the whole Cloud Pak for Integration. **If does not work mirror individual components**

Event Streams is deprecated and is not included in the CASE archive for Cloud Pak for Integration. Event Streams is still available as an individual component as described in the following section.

## Mirroring individual components

To mirror part of the Cloud Pak, use the CASE archive and inventory item for an individual component, and repeat the process for each component that you need to be available in your restricted environment.

CASE files for IBM components can be found in the [IBM CASE repository](#).

The CP4I component CASE files available for individual mirroring are:

IBM Cloud Pak for Integration Platform Navigator

```
1 | export CASE_ARCHIVE=ibm-integration-platform-navigator-1.1.0.tgz
2 | export CASE_INVENTORY_SETUP=platformNavigatorOperator
```

IBM API Connect

```
1 | export CASE_ARCHIVE=ibm-apiconnect-2.0.1.tgz
2 | export CASE_INVENTORY_SETUP=apiconnectOperatorSetup
```

IBM App Connect

```
1 | export CASE_ARCHIVE=ibm-appconnect-1.1.0.tgz
2 | export CASE_INVENTORY_SETUP=ibmAppconnect
```

IBM MQ

```
1 | export CASE_ARCHIVE=ibm-mq-1.3.0.tgz
2 | export CASE_INVENTORY_SETUP=ibmMQOperator
```

## Download the installer and image inventory

Download the installer and image inventory to the bastion host.

This step downloads the selected CASE file and its dependencies to the local machine.

It also produces CSV files listing the images and helm charts included in each CASE file. The CP4I components do not include any helm charts.

```
1 | $ cloudctl case save \
2 |   --case https://github.com/IBM/cloud-
3 |   pak/raw/master/repo/case/${CASE_ARCHIVE} \
4 |   --outputdir ${HOME}/offline/
```

## Log in to OpenShift cluster as a cluster administrator

Log in to the OpenShift cluster. For example:

```
1 | $ oc login <cluster host:port> --username=<cluster admin user> --password=<cluster admin password>
```

## Create a Kubernetes namespace

Create an environment variable with a namespace to install, then create the namespace. For example:

```
1 | export NAMESPACE=cp4i
2 | oc create namespace ${NAMESPACE}
3 |
```

## Mirror the images and configure the cluster

Complete these steps to mirror the images and configure your cluster:

1. Store authentication credentials for the IBM Entitled Registry.

See [entitlement-key] for how to obtain your entitlement key.

After obtaining your entitlement key, run the following command to configure credentials for the IBM Entitled Registry:

```
1 | cloudctl case launch \
2 |   --case ${HOME}/offline/${CASE_ARCHIVE} \
3 |   --inventory ${CASE_INVENTORY_SETUP} \
4 |   --action configure-creds-airgap \
5 |   --namespace ${NAMESPACE} \
6 |   --args "--registry cp.icr.io --user cp --pass <entitlement key> --inputDir ${HOME}/offline"
```

This command stores and caches the registry credentials in a file on your file system in the `${HOME}/airgap/secrets` location.

2. Create environment variables with the local Docker registry connection information.

```
1 |
2 | export LOCAL_DOCKER_REGISTRY_HOST=rh-smoc-cloud-inst.secsmoc.local
3 | export LOCAL_DOCKER_REGISTRY_PORT=5000
4 | export LOCAL_DOCKER_REGISTRY=${LOCAL_DOCKER_REGISTRY_HOST}:${LOCAL_DOCKER_REGISTRY_PORT}
5 | export LOCAL_DOCKER_USER=admin
6 | export LOCAL_DOCKER_PASSWORD=admin123
7 |
```

3. Configure a global image pull secret and [ImageContentSourcePolicy](#).

To enable your disconnected cluster to access images from your private registry, it must be configured to use your private registry as a mirror of the images hosted in the online registries, and to be able to access those images.

This step configures an [ImageContentSourcePolicy](#) for the images listed in the component CASEs. See [Configuring image registry repository mirroring](#) in the Red Hat OpenShift documentation for more details.

This step also configures the global cluster pull secret for the cluster to allow it to access the private registry. See [Adding the registry to your pull secret](#) in the Red Hat OpenShift documentation for more details.

In OpenShift version 4.6, this step performs a rolling restart of all cluster nodes. The cluster resources might be unavailable until the new [ImageContentSourcePolicy](#) and global cluster pull secret is applied.

```

1 | cloudctl case launch \
2 |   --case ${HOME}/offline/${CASE_ARCHIVE} \
3 |   --inventory ${CASE_INVENTORY_SETUP} \
4 |   --action configure-cluster-airgap \
5 |   --namespace ${NAMESPACE} \
6 |   --args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} -- \
7 |         pass ${LOCAL_DOCKER_PASSWORD} --inputDir ${HOME}/offline"

```

- Verify that the ImageContentSourcePolicy resource is created.

```
1 | oc get ImageContentSourcePolicy
```

- Verify your cluster node status.

```
1 | oc get nodes
```

After the `ImageContentsourcePolicy` and global image pull secret are applied, you might see the node status as `Ready`, `Scheduling`, or `Disabled`. Wait until all the nodes show a `Ready` status.

- Configure an authentication secret for the local Docker registry.

This step needs to be done only one time.

```

1 | cloudctl case launch \
2 |   --case ${HOME}/offline/${CASE_ARCHIVE} \
3 |   --inventory ${CASE_INVENTORY_SETUP} \
4 |   --action configure-creds-airgap \
5 |   --namespace ${NAMESPACE} \
6 |   --args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} -- \
7 |         pass ${LOCAL_DOCKER_PASSWORD}"

```

The command stores and caches the registry credentials in a file on your file system in the `${HOME}/.airgap/secrets` location.

- Mirror the images to the local registry.

This command calls the `oc image mirror` command to mirror images from the online registry to the private registry:

```

1 | cloudctl case launch \
2 |   --case ${HOME}/offline/${CASE_ARCHIVE} \
3 |   --inventory ${CASE_INVENTORY_SETUP} \
4 |   --action mirror-images \
5 |   --namespace ${NAMESPACE} \
6 |   --args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} -- \
7 |         pass ${LOCAL_DOCKER_PASSWORD} --inputDir ${HOME}/offline"

```

If you are using an insecure registry, you must also add the local registry to the `insecureRegistries` list for your cluster:

```

1 | oc patch image.config.openshift.io/cluster --type=merge \
2 |   -p '{"spec":{"registrySources":{"insecureRegistries": \
3 |     ["${LOCAL_DOCKER_REGISTRY}"]}}}'

```

## Create the CatalogSource

CP4I can be installed by adding the `CatalogSource` for the mirrored operators to your cluster and using OLM to install the operators.

- Create a catalog source.

This command adds the `CatalogSource` for the components to your cluster, so the cluster can access them from the private registry:

```

1 | clouddctl case launch \
2 |   --case ${HOME}/offline/${CASE_ARCHIVE} \
3 |   --inventory ${CASE_INVENTORY_SETUP} \
4 |   --action install-catalog \
5 |   --namespace ${NAMESPACE} \
6 |   --args "--registry ${LOCAL_DOCKER_REGISTRY} --inputDir ${HOME}/offline --
recursive"

```

- Verify that the **CatalogSource** for the Cloud Pak installer operator is created.

```

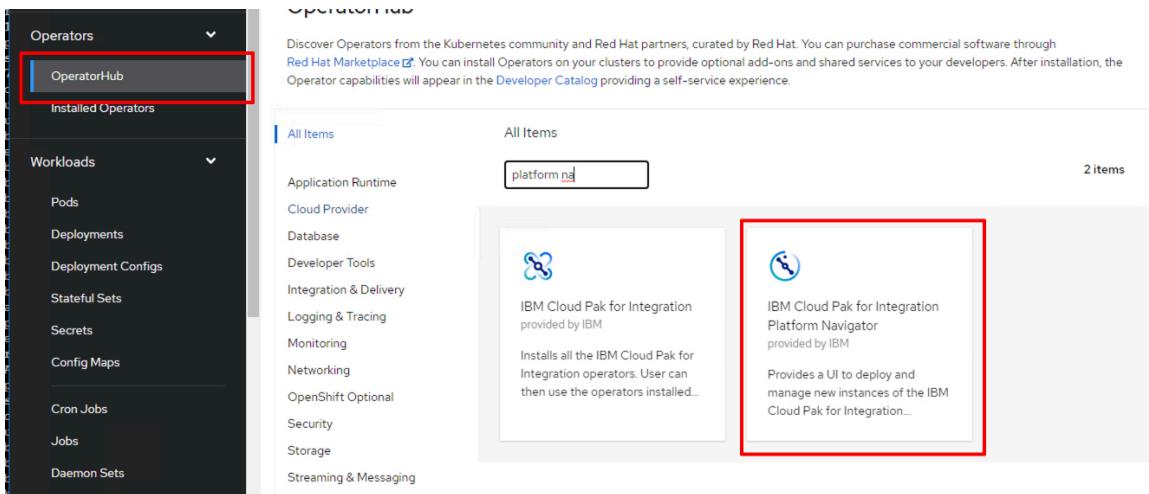
1 | oc get pods -n openshift-marketplace
2 | oc get catalogsource -n openshift-marketplace

```

## Cloud Pak for Integration Platform Navigator

Now that we have mirrored the images we can open **Web console** to install all the operators

- Install **Platform Navigator Operator** from **OperatorHub**





## IBM Cloud Pak for Integration Platform Navigator

4.1.1 provided by IBM

**Install**

---

**Latest Version**  
4.1.1

**Capability Level**

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

**Provider Type**  
IBM CP4I Platform Navigator

**Provider**  
IBM

**Infrastructure Features**  
Disconnected

**Repository**  
N/A

**Introduction**

IBM Cloud Pak for Integration Platform Navigator provides a UI to allow users to deploy instances of the Cloud Pak components, and allows navigation between them in a simple manner.

**Details**

The following components can be managed in the Platform Navigator:

- IBM Cloud Pak Asset Repository, to store, manage, retrieve and search integration assets
- IBM Cloud Pak Operations Dashboard, enabling IBM integration products to provide tracking information
- IBM API Connect, implementing managed APIs
- IBM App Connect Enterprise, providing integration workflows
- IBM App Connect Designer, to build integration flows with prebuilt connectors and templates
- IBM MQ, for secure and reliable assured delivery
- IBM Event Streams, for event handling based on Kafka
- IBM Aspera HSTS, for large file transfers
- IBM DataPower Gateway, for a purpose-built security and integration gateway

When installing operator select **All namespaces on the cluster**

[OperatorHub](#) > [Operator Installation](#)

### Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

**Update Channel\***

v1.0

v1.1-eus

**Installation Mode\***

All namespaces on the cluster (default)  
Operator will be available in all namespaces.

A specific namespace on the cluster  
Operator will be available in a single namespace only.

**Installed Namespace\***

PR openshift-operators

**Approval Strategy\***

Automatic

Manual



IBM Cloud Pak for Integration  
provided by IBM

Provided APIs

No Kubernetes APIs are provided by this Operator.

Activate  
Go to Setti

**Install**   **Cancel**

1. In the left navigation, click **Operators > Installed Operators**.
2. Click **IBM Cloud Pak for Integration Platform Navigator**.

The screenshot shows the OpenShift OperatorHub interface. On the left, there's a sidebar with 'Administrator' at the top, followed by 'Home', 'Operators' (which is expanded), 'OperatorHub' (highlighted with a red box), and 'Installed Operators' (also highlighted with a red box). Below 'OperatorHub' are 'Workloads', 'Pods', 'Deployments', 'Deployment Configs', and 'Stateful Sets'. On the right, the main area is titled 'Installed Operators' with a sub-instruction: 'Installed Operators are represented by Cluster Service Versions within this namespace. You can install a new operator by creating a new Cluster Service Version using the Operator SDK'. There are two search/filter boxes: one for 'Name' containing 'plat' and another for 'plat'. A table lists the installed operator: 'IBM Cloud Pak for Integration Platform Navigator' (version 4.1.1 provided by IBM), with its icon and details highlighted with a red box. The table has columns for 'Name' and 'Managed Namespaces'.

3. Click **Platform Navigator**, then click **Create Instance**

The screenshot shows the 'Operator Details' page for the 'IBM Cloud Pak for Integration Platform Navigator'. At the top, it says 'Project: cp4i' and 'Installed Operators > Operator Details'. Below that is the operator card for 'IBM Cloud Pak for Integration Platform Navigator' (version 4.1.1 provided by IBM). The card includes a 'Details' tab (which is active and highlighted with a red box) and other tabs for 'YAML', 'Subscription', 'Events', and 'Platform Navigator'. Under the 'Details' tab, there's a section titled 'Provided APIs' containing a box for 'Platform Navigator'. This box contains a 'Create Instance' button, which is also highlighted with a red box. Below the box, there's a 'Description' section.

4. Change the **name** of Instance

## Create PlatformNavigator

Create by completing the form. Default values may be provided by the Operator authors.

Configure via:  Form View  YAML View

**Note:** Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name \*

cp4i-navigator-prod

Labels

app=frontend

### 5. Accept the License

Name \*

cp4i-navigator-prod

http

Labels

app=frontend

License

License acceptance information.

Replicas

2

License

License acceptance information.

License Accept \*

true

Accept Terms and Conditions - Please refer to <https://ibm.biz/cp4i-licenses> to select the appropriate LI field

License LI

L-RJON-BUVMQX

Optional - Select the correct LI for a given version. Not all versions support this field. For versions before 2020.4.1-0, please delete this field from the CR in YAML view

### 6. Create (Ensure version is set to 2020.4.1-eus)

Replicas

2

The desired number of replica pods for the Platform Navigator.

MQ Dashboard

true

Enables the MQ Grafana dashboard for monitoring MQ instances.

Version

2020.41-eus

The desired version or channel of the Platform Navigator.

> Advanced Configuration

Create

Cancel

### 7. Ensure all pods in `ibm-common-services` are running (should be around **65-67 pods**) and the two pods in Operator hub was created

Project: openshift-operators					
Name	Status	Ready	Restarts	Node	Labels
cp4i-navigator-prod-ibm-inte-3c22-deployment-74bfb84458-drgf6	Running	2/2	0	rh-smoc-cloud-worker5.ocpcluster.secsmoc.local	app.kubernetes.io/label...=dep... app.kubernetes.io/label...=cp4i-n... app.kubernetes.io/label...=ibm-integ... app.kubernetes.io/label...=ibm-integ... inter...=cp4i-navig... pod-temp...=74bfb...
cp4i-navigator-prod-ibm-inte-3c22-deployment-74bfb84458-pnndl	Running	2/2	0	rh-smoc-cloud-worker3.ocpcluster.secsmoc.local	app.kubernetes.io/label...=dep... app.kubernetes.io/label...=cp4i-n... app.kubernetes.io/label...=ibm-integ... app.kubernetes.io/label...=ibm-integ... inter...=cp4i-navig... pod-temp...=74bfb...

8. After completed you can find the route here

The screenshot shows the OpenShift console interface. The left sidebar has a dark theme with the following navigation items:

- Explore
- Events
- Operators
- OperatorHub
- Installed Operators
- Workloads > (highlighted by a red box)
- Networking > (highlighted by a red box) **1**
- Services
- Routes **2** (highlighted by a red box)
- Ingresses
- Network Policies

The main content area has a light blue header bar with the message "You are logged in as a temporary administrative user. Update the cluster OAuth configuration". Below the header is a search bar with the placeholder "Project: cp4i" and a dropdown arrow, with a red box and a red circle containing the number "3" positioned above it.

## Routes

Filter:  Name:  Search by name...

Name	Status	Location
cp4i-navigator-prod-pn	Accepted	<a href="https://cp4i-navigator-prod-pn-cp4i.apps.ocpcluster.secsmoc.local">https://cp4i-navigator-prod-pn-cp4i.apps.ocpcluster.secsmoc.local</a> <b>4</b>

Navigate to Platform Navigator login using **Openshift Authentication** or **Admin**:

Login details can be found by running the following command:

```
1 | $ oc extract secret/ibm-iam-bindinfo-platform-auth-idp-credentials -n ibm-common-services --to=--
```

## Link

1. <https://cp4i-navigator-prod-pn-openshift-operators.apps.ocpcluster.secsmoc.local/>

# EventStreams

Type	Status	Updated	Message
⚠ Warning	True	3 hours ago	Communication between Event Streams components is plain text. If encrypted data is required, edit spec.security.internalTls to provide value 'TLSv1.2'.
⚠ Warning	True	3 hours ago	No authorization is enabled for Kafka. Any verified user will have permissions to access and perform actions against your Kafka brokers. If Kafka authorization is required, edit spec.strimziOverrides.kafka.authorization.type to provide value "runas".
⚠ Warning	True	3 hours ago	No authentication is enabled for Kafka listeners. Any client can connect to your Kafka brokers. However, you will have reduced access to Event Streams and Cloud Pak for Integration features, including the use of the UI, and monitoring and metrics capabilities. To access all features, enable authentication for Kafka by setting the authentication type in one or more of the following locations: spec.strimziOverrides.kafka.listeners.external.authentication, spec.strimziOverrides.kafka.listeners.tls.authentication, and spec.strimziOverrides.kafka.listeners.plain.authentication
🟢 Ready	True	3 hours ago	

If you find in the platform navigator event streams not ready and have a yellow warning sign do the following:

```

1  spec:
2    security:
3      internalTls: TLSv1.2 # change from NONE to TLSv1.2
4    license:
5      accept: true
6    use: CloudPakForIntegrationProduction
7    strimziOverrides:
8      kafka:
9        config:
10          inter.broker.protocol.version: '2.6'
11          interceptor.class.names:
12            com.ibm.eventstreams.interceptors.metrics.ProducerMetricsInterceptor
13              log.message.format.version: '2.6'
14              offsets.topic.replication.factor: 1
15              transaction.state.log.min_isr: 1
16              transaction.state.log.replication.factor: 1
17            listeners:
18              tls:
19                authentication:
20                  type: tls # change plain: {} to the following
21            metrics: {}
22            # add these lines
23            authorization:
24              type: runas
25            # end of addition
26            replicas: 3
27            storage:
28              class: ocs-storagecluster-ceph-rbd
29              size: 100Gi
30              type: persistent-claim
31            zookeeper:
32              metrics: {}
33              replicas: 1
34              storage:
35                class: ocs-storagecluster-ceph-rbd
36                size: 100Gi
37                type: persistent-claim

```

```

37   adminUI: {}
38   restProducer: {}
39   apicurioRegistry: {}
40   adminApi: {}
41   collector: {}
42   version: 10.2.0-eus

```

<https://ibm.github.io/event-streams/installing/configuring/>

## API Connect

### Install API Connect

1. Install Operator in **apic** project

The screenshot shows the OperatorHub interface within a Kubernetes project named 'apic'. A search bar at the top contains the text 'apic'. Below the search bar, there are two tabs: 'All Items' and 'All Items'. On the left, a sidebar lists categories: Application Runtime, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, and Networking. In the main area, a search result for 'apic' is shown. The result includes a thumbnail icon, the name 'API Designer provided by Red Hat', a brief description, and a link. To the right of this result, another item is listed: 'IBM API Connect provided by IBM'. This second result is also highlighted with a red box. Below the results, a note states '19 items'.

The screenshot shows the detailed view of the 'IBM API Connect' operator on the OperatorHub. The title is 'IBM API Connect 2.1.0 provided by IBM'. On the left, a sidebar lists categories: Application Runtime, Cloud Provider, Database, Developer Tools, Integration & Delivery, Logging & Tracing, Monitoring, Networking, OpenShift Optional, Security, and Storage. In the main area, there is a large blue 'Install' button highlighted with a red box. To the right of the button, there are sections for 'Latest Version' (2.1.0), 'Capability Level' (Basic Install, Seamless Upgrades selected), 'Provider Type' (IBM APIConnect catalog), 'Provider' (IBM), 'Repository' (N/A), 'Container Image' (ibmcom/apiconnect:2.1.0), 'Introduction', 'Summary', and 'Features'.

2. After the APIC operator installed Navigate to the **Platform Navigator > Create capability**

<https://cp4i-navigator-prod-pn Openshift-operators.apps.ocpcluster.secsmoc.local/>

Welcome to IBM Cloud Pak for Integration

**Capabilities** Runtimes

Capabilities provide tools for creating and managing some types of integration instances.

Q Find

Name Capability type Namespace Version Status

Create capability +

### 3. Click **API Connect** then click **Next**

Back to Pak Home

## Create capability

Capabilities provide tools for designing and managing some types of integration instances. Select which type you would like to create.

Some capabilities can't be created yet. This may be because you don't have permission to create capabilities, or the capability operators aren't installed in this OpenShift project. Contact your cluster administrator to gain access, or install the operators in OpenShift and then return to this page.

**API Connect**

Expose, manage, share and monetize APIs securely across clouds.

**App Connect Dashboard**

Create and manage instances of App Connect runtimes.

**App Connect Designer**

Develop integration flows with prebuilt connectors and no-code templates.

**Operations Dashboard**

Runtime diagnosis and troubleshooting. Analyze performance bottlenecks and latency issues with end to end data tracing.

Install operators in OpenShift

### 4. Select **Production** under types then click **Next**

Back to Pak Home

## Create an instance of API Connect

For more help with instance creation check the [documentation](#)

Select type Configuration

Select one of the following options, you will be able to customize it in the following step:

<b>One Node - Minimum</b> Best suited for light, non-critical workloads such as basic development and testing. Includes one node of each type (management, API gateway, analytics, and portal).	<b>Three Node - Production</b> Ideal for heavy, critical workloads. Includes three nodes of each type (management, API gateway, analytics, and portal) to improve capacity and availability.
<input checked="" type="radio"/> VPCs 12 <input checked="" type="radio"/> CPU 12 <input checked="" type="radio"/> Memory 48 GB <input checked="" type="radio"/> Storage 280 GB	<input checked="" type="radio"/> VPCs 48 <input checked="" type="radio"/> CPU 48 <input checked="" type="radio"/> Memory 144 GB <input checked="" type="radio"/> Storage 556 GB

Activate Windows  
Go to Settings to activate Windows.

### 5. Fill in the details as below changing the following

- o **Name**
- o **Accept license,**
- o Change License use to **Production**,
- o Select **Deployment Profile = n12xc4.m12**
- o **Storage Class** to match **ocs-storagecluster-cephfs**

Create an instance of API Connect | For more help with instance creation check the documentation [documentation](#)

Select type  Configuration

[Back](#) [Create](#)

[UI form](#) [YAML](#)

Advanced settings  Off

**Details**

**Metadata**

① Name: apic-prod Namespace: apic

② License

③ Licence acceptance (optional)  On

License agreement <https://ibm.biz/gictoolkitlic> must be accepted during install. If you do not accept the license, the product will not install.

④ License use (optional): production

Select the type of license that will be applied to this installation. Default is nonproduction. For more info, see: <https://ibm.biz/BdgUdf>

⑤ Deployment profile (optional): n12xc4.m12

Select the size of the instances that should be used for this deployment. n3xc4.m16 is recommended for development and testing n12xc4.m12 is recommended for high availability and failover resiliency. For more details, see: <https://ibm.biz/BdgUdz>

Activate Windows [Go to Settings to activate Windows](#)

#### 6. Update the **Storage type** and **Version** to match below

**Deployment profile (optional)**

n12xc4.m12

Select the size of the instances that should be used for this deployment. n3xc4.m16 is recommended for development and testing n12xc4.m12 is recommended for high availability and failover resiliency. For more details, see: <https://ibm.biz/BdgUdz>

**Product version (optional)**

10.0.1.1-eus

Select a specific version or channel of the product. For more details and compatibility info, see: <https://ibm.biz/BdgUdf>

**Storage class (optional)**

ocs-storagecluster-ceph-rbd

Specify the RWO block storage class to use for persistence storage. This can be further customized in advanced settings. If not specified the default cluster storage class is used. Specification of the storage class is highly recommended. For more info, see: <https://ibm.biz/BdgUd2>

Activate Windows [Go to](#)

#### 7. Click **Create**

[Back](#) [Create](#)

It be accepted during install. If you do not accept the license, the product will not install.

installation. Default is nonproduction. For more info, see: <https://ibm.biz/BdgUdf>

Make sure the deployment succeeded. (lots of pods with a subset below)

```

1 $ oc get pods -n apic | grep apic
2
3 apic-a7s-cj-rollover-1612818900-x2rnt          0/1
Completed    0           5m20s
4 apic-a7s-client-79b58ccd5f-kd24h            1/1
Running      0           45m

```

5	apic-a7s-ingestion-669f4d4bb4-nwfwx	1/1
	Running 0 45m	
6	apic-a7s-mtls-gw-c44d4df6f-2fptt	1/1
	Running 0 43m	
7	apic-a7s-storage-coord-56965cf6f7-kxsk5	1/1
	Running 0 45m	
8	apic-a7s-storage-data-0	1/1
	Running 0 45m	
9	apic-a7s-storage-master-0	1/1
	Running 0 45m	
10	apic-configuration-wrbdw	0/1
	Completed 0 37m	
11	apic-gw-0	3/3
	Running 0 41m	
12	apic-gw-od-registration-job-6nmh5	0/1
	Completed 0 45m	
13	apic-mgmt-08730b55-postgres-765d98c78f-msqb2	1/1
	Running 0 58m	
14	apic-mgmt-08730b55-postgres-backrest-shared-repo-5796c7887nzjfk	1/1
	Running 0 60m	
15	apic-mgmt-08730b55-postgres-pgbouncer-6f48bb576-tvldx	1/1
	Running 0 56m	
16	apic-mgmt-08730b55-postgres-stanza-create-hgmgq	0/1
	Completed 0 56m	
17	apic-mgmt-analytics-proxy-598b786df5-frzzz	1/1
	Running 0 50m	
18	apic-mgmt-apim-647f46f7b4-c9hmq	1/1
	Running 0 47m	
19	apic-mgmt-billing-798667ccdd-48rwp	1/1
	Running 0 50m	
20	apic-mgmt-client-downloads-server-7f7ff89858-rhsnw	1/1
	Running 0 55m	
21	apic-mgmt-hub-b98b4f5b6-fwqgg	1/1
	Running 0 55m	
22	apic-mgmt-juhu-6cc6556549-48bd5	1/1
	Running 0 48m	
23	apic-mgmt-ldap-5d99986776-1pbqv	1/1
	Running 0 55m	
24	apic-mgmt-lur-7995cc77b9-vd9pq	1/1
	Running 0 47m	
25	apic-mgmt-nats-operator-6747c9b5f4-h5bw9	1/1
	Running 0 55m	
26	apic-mgmt-nats-streaming-operator-6dc6794f46-b957q	1/1
	Running 0 55m	
27	apic-mgmt-natscluster-1	1/1
	Running 0 55m	
28	apic-mgmt-portal-proxy-64ccd9494c-prs1s	1/1
	Running 0 50m	
29	apic-mgmt-stancluster-1	1/1
	Running 0 54m	
30	apic-mgmt-taskmanager-5b85ccbd4f-dhv7j	1/1
	Running 0 50m	
31	apic-mgmt-turnstile-55f79cbf8d-kvrv2	1/1
	Running 0 50m	
32	apic-mgmt-ui-7bb65c78fd-kghpf	1/1
	Running 0 55m	

33	apic-mgmt-up-apim-data-populate-0-to-6-216243c7-dlk5d	0 / 1
Completed	0	53m
34	apic-mgmt-up-apim-schema-0-to-6-216243c7-v2cjw	0 / 1
Completed	0	55m
35	apic-mgmt-up-lur-data-populate-0-to-2-216243c7-4h65w	0 / 1
Completed	0	54m
36	apic-mgmt-up-lur-schema-0-to-2-216243c7-77zww	0 / 1
Completed	0	55m

To access your APIC instance you can navigate to the **Platform navigator**

Name	Capability type	Namespace	Version	Status
apic-prod	API Connect	apic	10.0.1.1-1423-eus	Ready
ace-prod-dashboard	App Connect Dashboard	ace	11.0.0.10-r3-eus	Ready

## Configure API Connect

### Configure a dummy email client

1. Navigate to Cloud Manager

Cloud Manager access route

Name	Status	Location	Service
apic-prod-mgmt-admin	Accepted	https://apic-prod-mgmt-admin-apic.apps.ocpcluster.secsmoc.local	apic-prod-mgmt-juhu
apic-prod-mgmt-api-manager	Accepted	https://apic-prod-mgmt-api-manager-apic.apps.ocpcluster.secsmoc.local	apic-prod-mgmt-juhu

Password

```
1 | oc get secrets -n apic apic-prod-mgmt-admin-pass -ojsonpath='{.data.password}' | base64 --decode && echo ""
```

- 2.

The screenshot shows the IBM Cloud Pak for Integration Runtimes page. At the top, there's a navigation bar with tabs for 'Capabilities' and 'Runtimes'. Below the navigation is a search bar labeled 'Find'. A table lists a single capability: 'apic-prod' (API Connect), which is in the 'Namespace' 'apic', 'Version' '10.0.1.1-1423-eus', and 'Status' 'Ready'. To the right of the table is a vertical sidebar with options: 'Cloud manager' (highlighted with a red box), 'Logs', 'Monitoring', 'Edit', 'Change version', and 'Delete'. A small 'i' icon is also present.

3. Login using credentials

```
$ oc extract secret/ibm-iam-bindinfo-platform-auth-idp-credentials -n ibm-common-services --to=
```

4. Navigate to **Settings > Notifications > Edit**

The screenshot shows the 'Settings' page in IBM Cloud Pak for Integration. On the left, a sidebar menu has 'Notifications' highlighted with a red box and a circled '1'. On the right, under the 'Sender & email server' section, there's a configuration form with fields for 'Name' (set to 'APIC Administrator') and 'Email' (set to 'admin@apiconnect.net'). A red box highlights the 'Edit' button at the top right of this form. A circled '3' is placed near the 'Edit' button.

5. Configure Email Server

Can't find the one you want? [Configure Email Server](#)

6. Enter the following and click **Save**

IBM Cloud Pak for Integration | API Connect apic | apic-prod

### Email server configuration

Title  
**Dummy Mail Server**

Name  
dummy-mail-server

Address  
**smtp.sendgrid.net**

Port  
**587**

Authenticate user (optional)

Authenticate password (optional)

TLS client profile (optional)  
**Default TLS client profile:1.0.0**

Secure connection

### Test connection

Send a test email to confirm that the email server is properly configured.

Cancel **Save**

Test email

## Reference

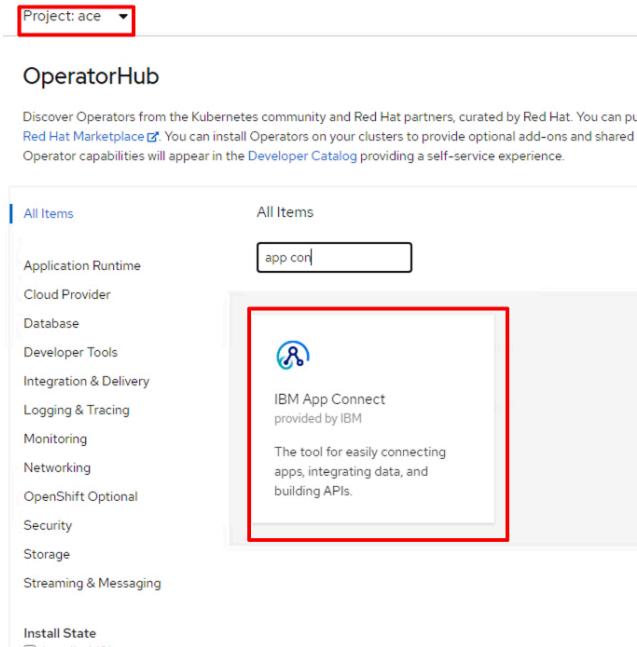
1. <https://www.ibm.com/docs/en/api-connect/10.0.1.x?topic=configuration-basic-settings>

## Link

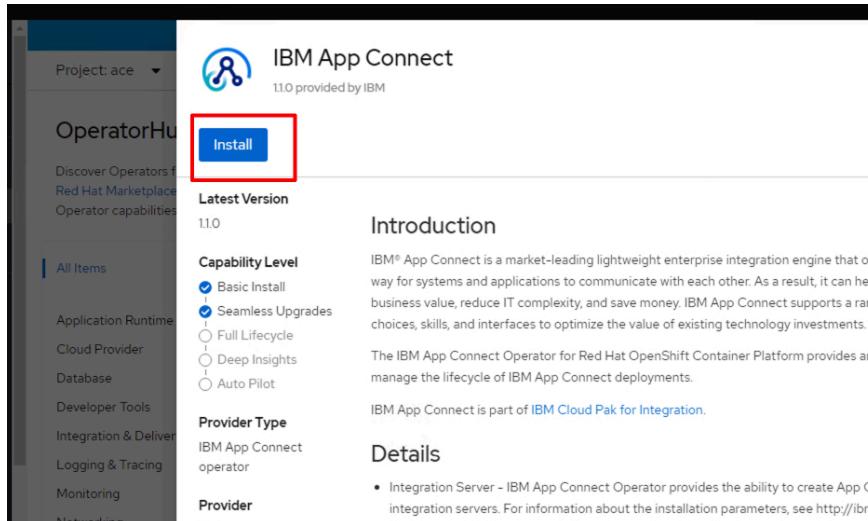
1. <https://apic-prod-mgmt-admin-apic.apps.ocpcluster.secsmoc.local/admin/>
2. <https://apic-prod-mgmt-api-manager-apic.apps.ocpcluster.secsmoc.local/manager>

# App Connect

## 1. Install Operator in ace project



The screenshot shows the OperatorHub interface for a project named 'ace'. A red box highlights the 'Project: ace' dropdown menu. In the search bar, the text 'app connect' is entered, and a red box highlights the search results. The search results show a single item: 'IBM App Connect provided by IBM'. This item has a red border around it. Below the search results, there is a sidebar with categories like Application Runtime, Cloud Provider, Database, etc., and an 'Install State' section indicating 'Installed (0)'.

The screenshot shows the details page for the 'IBM App Connect' operator. A red box highlights the 'Install' button. The page includes sections for 'Latest Version' (1.1.0), 'Capability Level' (Basic Install, Seamless Upgrades selected), 'Provider Type' (IBM App Connect operator), and 'Provider' (IBM). The 'Introduction' and 'Details' sections provide general information about the operator.

You are logged in as a temporary administrative user. Update

OperatorHub > Operator Installation

## Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator updated.

**Update Channel \***

v1.0  
 v1.1-eus

**Installation Mode \***

All namespaces on the cluster (default)  
 Operator will be available in all namespaces.  
 A specific namespace on the cluster  
 Operator will be available in a single namespace only.

**Installed Namespace \***

ace

**Approval Strategy \***

Automatic  
 Manual

**Install** **Cancel**

- After the ACE operator installed Navigate to the **Platform Navigator > Create capability**  
<https://cp4i-navigator-prod-pn Openshift-operators.apps.ocpcluster.secsmoc.local/>

Welcome to IBM Cloud Pak for Integration

Capabilities

Runtimes

Capabilities provide tools for creating and managing some types of integration instances.

Name	Capability type	Namespace	Version	Status

Create capability

- Click Ace Connect Dashboard then click **Next**

Back to Pak Home

### Create capability

Capabilities provide tools for designing and managing some types of integration instances. Select which type you would like to create.

**Some capabilities can't be created yet.**  
 This may be because you don't have permission to create capabilities, or the capability operators aren't installed in this OpenShift project. Contact your cluster administrator to gain access, or install the operators in OpenShift and then return to this page.

**Next**

<b>API lifecycle and secure access</b> API Connect	<b>Application integration</b> App Connect Dashboard	<b>Runtime diagnosis and troubleshooting</b> Operations Dashboard
-------------------------------------------------------	---------------------------------------------------------	----------------------------------------------------------------------

Expose, manage, share and monetize APIs securely across clouds.

Create and manage instances of App Connect runtimes.

Develop integration flows with prebuilt connectors and no-code templates.

Analyze performance bottlenecks and latency issues with end-to-end data tracing.

- Select **Production** under types then click **Next**

Select one of the following options, you will be able to customize it in the following step:

<b>Quick start</b>	<b>Production</b>
Quickly deploy a development dashboard with a single replica	Deploy a production dashboard that uses multiple replicas to increase resilience and availability
VPCs 0	VPCs 0
CPU 0.5	CPU 6
Memory 0.75 GB	Memory 3.75 GB
Storage 50B	Storage 50B

5. Fill in the details as below changing the following

- o **Name**
- o **Accept license,**
- o Change License use to **Cloud Pak for Integration Production,**
- o **Replicas = 2**
- o **Storage Class to match ocs-storagecluster-cephfs**

UI form    YAML

Advanced settings

Off

**Details**

Name: appconnect-prod    Namespace: ace

License

Accept:  On

License LI: L-APEH-BTHFYQ    License use: CloudPakForIntegrationProduction

Replicas (optional): 2

Storage

Storage class (optional): ocs-storagecluster-cephfs

6. Update the **Storage type** and **Version** to match below

e help with instance creation check the [documentation](#)

[Back](#) [Create](#)

License LI	License use
L-APEH-BTHFYQ	CloudPakForIntegrationProduction

Replicas (optional)  
2  
The number of replica pods to run

**Storage**

Storage class (optional)  
**ocs-storagecluster-cephfs**

Size (optional)  
5Gi  
Dashboard requires a persistent volume with ReadWriteMany access mode. If using IBM Cloud, set the storage class to ibmc-file-gold-gid for best performance - Required for type persistent-claim

Storage type (optional)  
**persistent-claim**  
Persistent claim storage is required for sharing BAR files across replicas. When using ephemeral storage, BAR files will be lost when Dashboard pods restart.

**Version**

Channel or version (optional)  
**11.0.0-eus**  
Specify the version of the App Connect Dashboard you would like to create

7. Click **Create**

[Back](#) [Create](#)

It be accepted during install. If you do not accept the license, the product will not install.

Installation. Default is nonproduction. For more info, see: <https://ibm.biz/BdgJdf>

To access your ACE instance you can navigate to the **Platform navigator**

Not secure | cp4i-navigator-prod-pn-openshift-operators.apps.ocpcluster.secsmoc.local/

IBM Cloud Pak for Integration | Integration Home

Welcome to IBM Cloud Pak for Integration

Capabilities Runtimes

Show more

Capabilities provide tools for creating and managing some types of integration instances.

Name	Capability type	Namespace	Version	Status
apic-prod	API Connect	apic	10.0.1.1-1423-eus	Ready
<b>ace-prod-dashboard</b>	App Connect Dashboard	ace	11.0.0.10-r3-eus	Ready

## Link

1. <https://ace-prod-dashboard-ui-ace.apps.ocpcluster.secsmoc.local/home>

## 1. Create MQ Operator

Project: mq ▾

## OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software from Red Hat Marketplace. You can install Operators on your clusters to provide optional add-ons and shared services to your development environment. Operator capabilities will appear in the Developer Catalog providing a self-service experience.

All Items

Application Runtime

Cloud Provider

Database

Developer Tools

Integration & Delivery

Logging & Tracing

Monitoring

Networking

OpenShift Optional

Security

Storage

Streaming & Messaging

Install State

Installed (1)

Not Installed (9)

Provider Type

All Items

mq



AMQ Broker  
provided by Red Hat

AMQ Broker Operator provides the ability to deploy and manage stateful AMQ Broker broker...



IBM Cloud Pak for Integration  
provided by IBM

Installs all the IBM Cloud Pak for Integration operators. User can then use the operators installed...



IBM Cloud Pak for Integration  
Platform Navigator  
provided by IBM

Provides a UI to deploy and...



IBM MQ  
provided by IBM

IBM MQ is an operator to manage the life cycle of IBM MQ queue managers. This operator is...

Project: mq ▾

OperatorHub

Discover Operators from the Red Hat Marketplace. Operator capabilities will appear in the Developer Catalog providing a self-service experience.

All Items

Application Runtime



IBM MQ

1.3.0 provided by IBM

Install

Latest Version

1.3.0

Capability Level

- Basic Install
- Seamless Upgrades
- Full Lifecycle

IBM MQ is messaging middleware that enables organizations to move mission-critical and business data across multiple platforms and environments. It provides a single message bus for distributed systems and premises environments.

The IBM MQ Operator for Red Hat OpenShift provides a simple way to manage the life cycle of IBM MQ queue manager instances.

## Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up-to-date.

### Update Channel \*

- v1.0
- v1.1
- v1.2
- v1.3-eus

### Installation Mode \*

- All namespaces on the cluster (default)  
Operator will be available in all namespaces.
- A specific namespace on the cluster  
Operator will be available in a single namespace only.

### Installed Namespace \*

PR mq

### Approval Strategy \*

- Automatic
- Manual

Install

Cancel

## References

- <https://www.ibm.com/docs/en/ibm-mq/9.2?topic=dqmorcpc-deploying-queue-manager-using-cloud-pak-integration-platform-navigator>
- <https://www.ibm.com/docs/en/ibm-mq/9.2?topic=operator-example-configuring-multi-instance-queue-manager>