

Installing and deploying IBM MQ Advanced queue manager on IBM Cloud Pak for Integration – ICP4I v2019.2.1

[Waleed Arshad](#)

Published on 26/09/2018 / Updated on 04/10/2018

This document provides detailed steps to install and deploy an IBM MQ Advanced queue manager on IBM Cloud Pak for Integration ([ICP4I v2019.2.1](#)) using the IBM MQ Advanced Helm chart version v1.2.2. It gives steps to add a new queue manager instance on IBM Cloud Pak for integration. It also provides steps to create and use queue manager dashboard on IBM Cloud Pak for integration.

It is divided into three parts.

1. Pre-requisite instructions to add a new IBM MQ queue manager instance.
2. Configure and deploy IBM MQ Advanced helm chart to create a queue manager.
3. Create queue manager dashboard widget and its objects.

1. Pre-requisite instructions to add a new IBM MQ queue manager instance:

This section provides the commands to create pre-requisite resources.

- Create a new namespace with any name – in this example, I am using “mqadv” – for a new MQ queue manager instance, using the following steps:

Assuming that the IP Address of the master node is XXX.YY.ZZZ.NN, and that the IBM Cloud Pak for Integration is successfully installed.

Open (installed as part of IBM Cloud Pak for Integration installation)/p>



Click IBM Cloud Private → Manage → Namespace



Create a new Namespace for a new IBM MQ queue manager:

- Specify Name of the Namespace as **mqadv**
- Specify Pod Security Policy* as **ibm-anyuid-psp** from the drop-down list.
- Click **Create**.

- `kubectl create namespace <namespace name>`

```
$ kubectl create namespace mqadv
```

namespace/mqadv created

- Create role binding **ibm-anyuid-psp** for the mqadv namespace

```
kubectl -n mqadv create rolebinding ibm-anyuid-clusterrole-rolebinding --clusterrole=ibm-anyuid-clusterrole --group=system:serviceaccounts:mqadv
```

rolebinding.rbac.authorization.k8s.io/ibm-anyuid-clusterrole-rolebinding created

Create image pull secret to access the docker registry using target namespace “mqadv”. In the following command, you must replace <username>, <password> and <[xxxxxxx@domain.com](#)> with their actual values.

```
kubectl create secret -n mqadv docker-registry my-mq-secret --docker-server=mycluster.icp:8500 --docker-username=<username> --docker-password=<password> --docker-email=<xxxxxxx@domain.com>
```

secret/my-mq-secret created

- Create image pull secret to access the private docker registry under target namespace “kube-system”. This is required for Configuration settings for single sign-on under target during IBM MQ Helm chart deployment. In the following command, you must replace <username>, <password> and <[xxxxxxx@domain.com](#)> with their actual values:

```
kubectl create secret -n kube-system docker-registry my-private-mq-secret --docker-server=mycluster.icp:8500 --docker-username=<username> --docker-password=<password> --docker-email=<xxxxxxx@domain.com>
```

secret/my-private-mq-secret created

- If persistence is enabled, you either need to create a PersistentVolume, or specify a Storage Class if classes are defined in your cluster. Here, an IBM MQ compatible Persistence volume is created ahead of creating a Queue Manager in an IBM cloud private cluster to store queue manager’s data and logs files into a persistence storage.
- Example: Create persistence volume “mq-volume-1” for /mq mount storage by using

IBM Cloud Private Menu → Platform → Storage → PersistentVolume → Create

For more details see: [Working with storage\(NFS and hostpath\)](#)

2. *Configure and deploy IBM MQ Advanced helm chart to create a queue manager:*

- Open [IBM Cloud Pak for Integration Console](#)
- Click **Messaging → Create Instance**

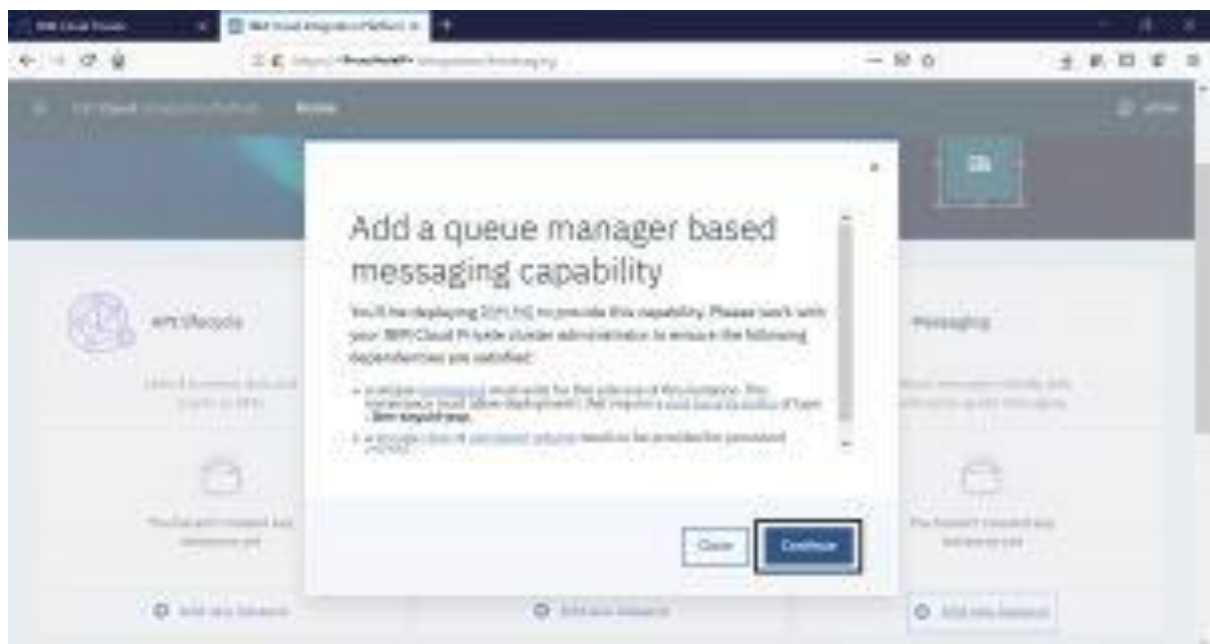


- Click **Add new instance** → **New Message Queue** instance in the Messaging section.



The IBM MQ Advanced Helm Chart Configuration page for installing the IBM MQ Advanced queue manager Helm release opens.

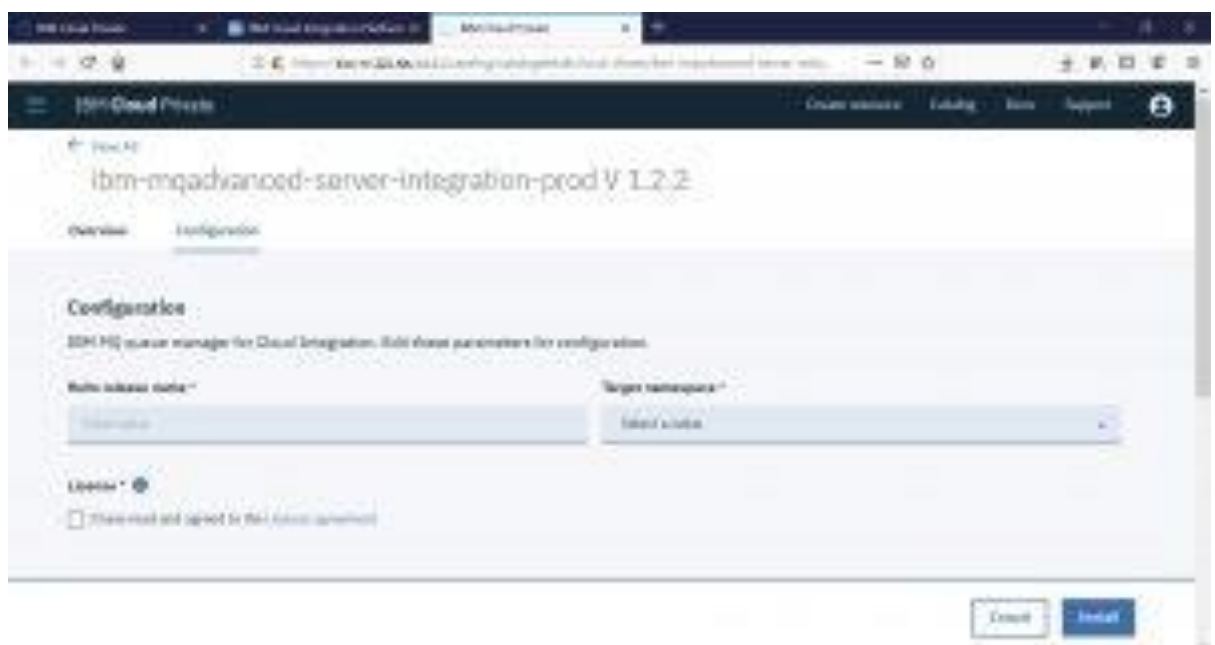
The following pre-requisites are already created in above steps.



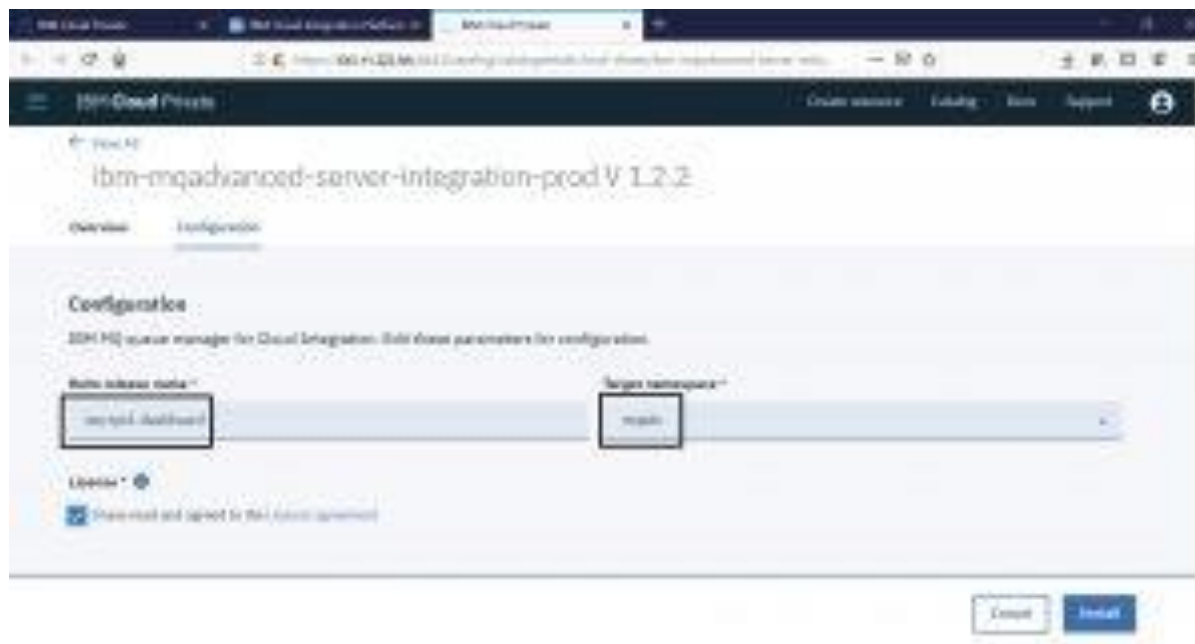
- Click **Continue**. The IBM MQ Advanced helm chart definition webpage opens.



- Click **Configure** to configure the Helm Chart

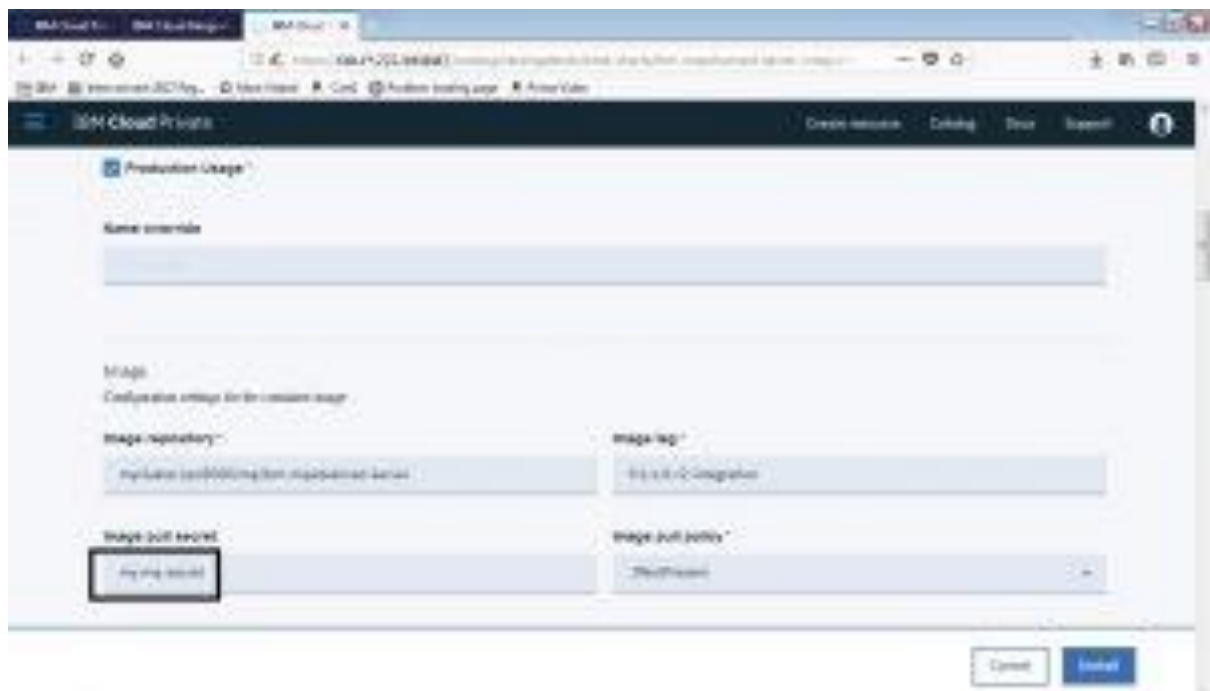


- Provide the helm release name and namespace, and accept the license as shown below.
- Configure the helm chart with the following parameter values:
 - Helm release name * : mq-qm1-dashboard
 - Target namespace : mqadv
 - Check Accept License agreement

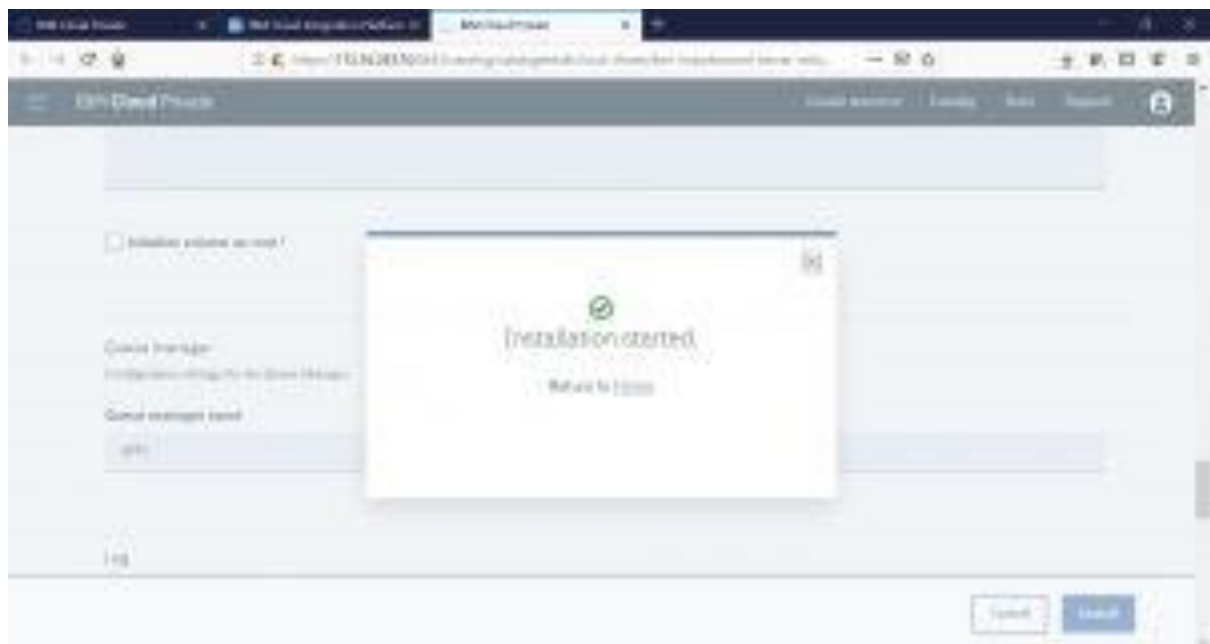


- Specify “image pull secret”: my-mq-secret

(This was created as a prerequisite.)



- Specify “Registration image pull secret”: my-private-mq-secret

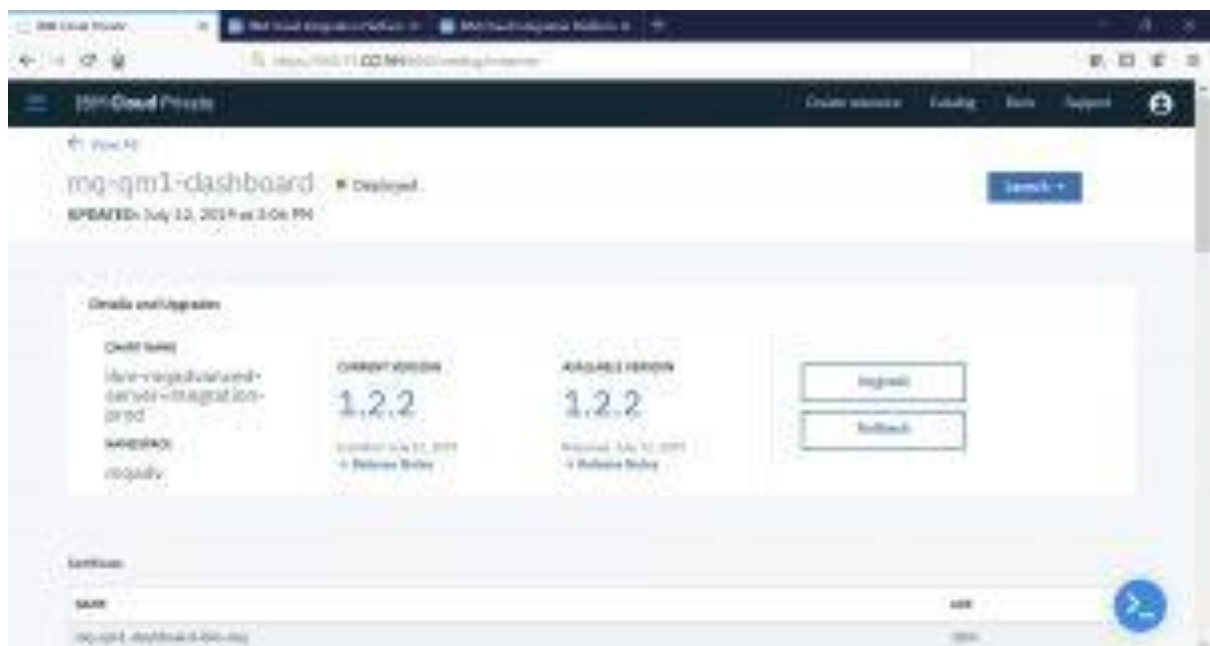
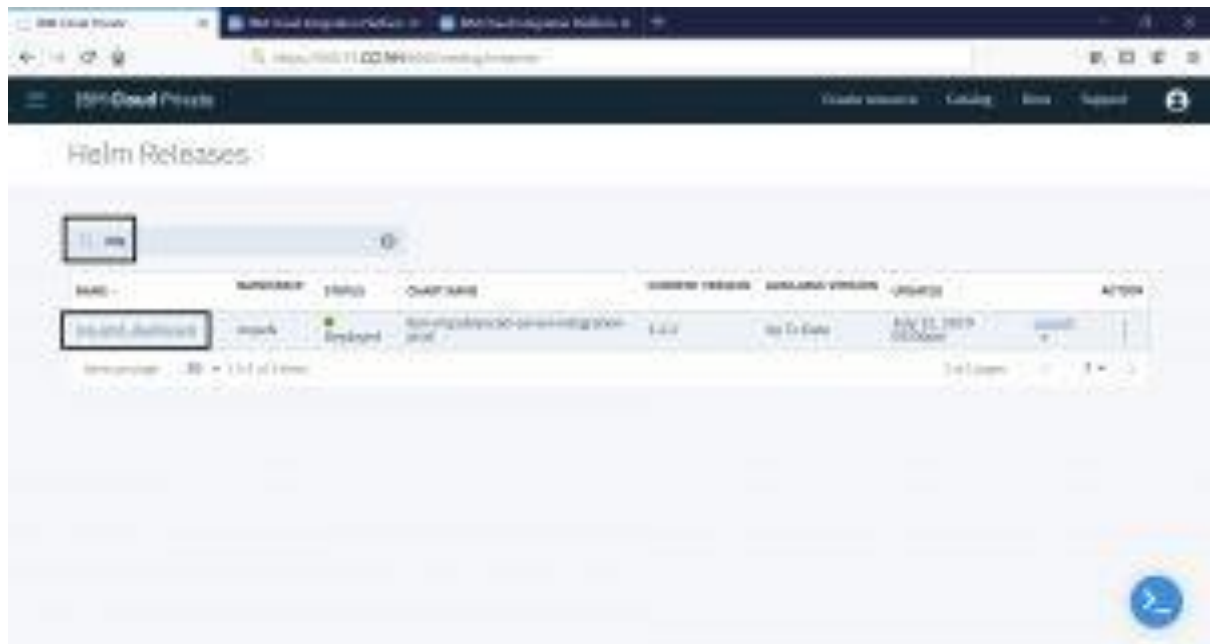


- Click **View Helm Release**
- [Open IBM Cloud Private Console](#) Here XXX.YY.ZZZ.NN is IP address of the master node.

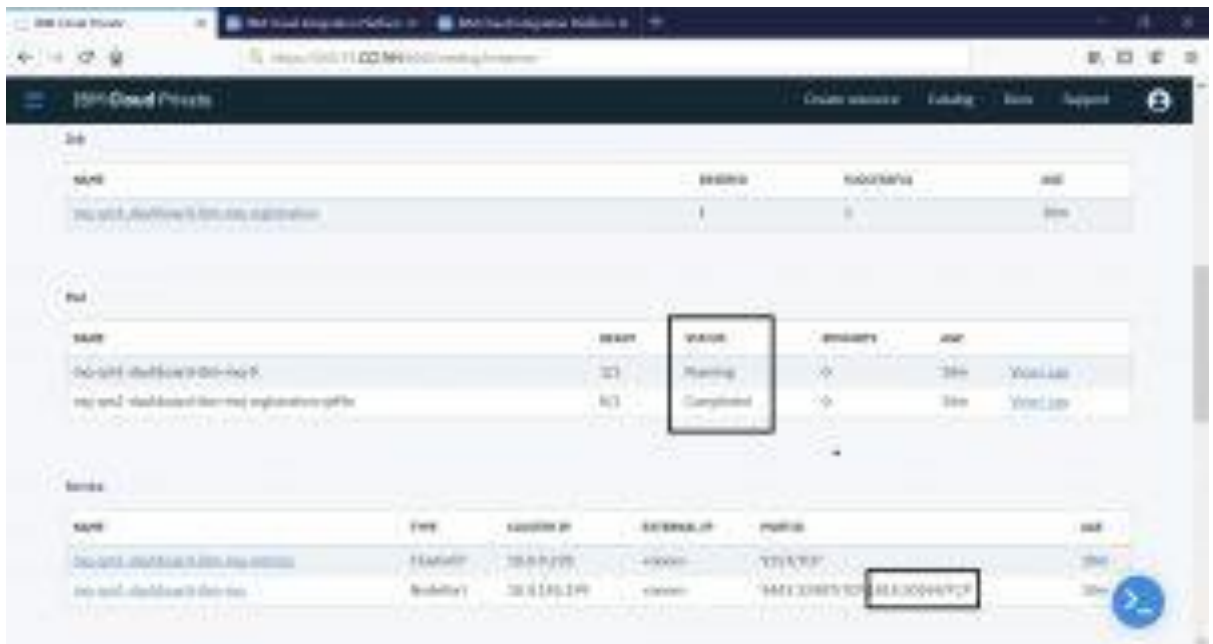


- Click **Menu → Workload → Helm Releases**

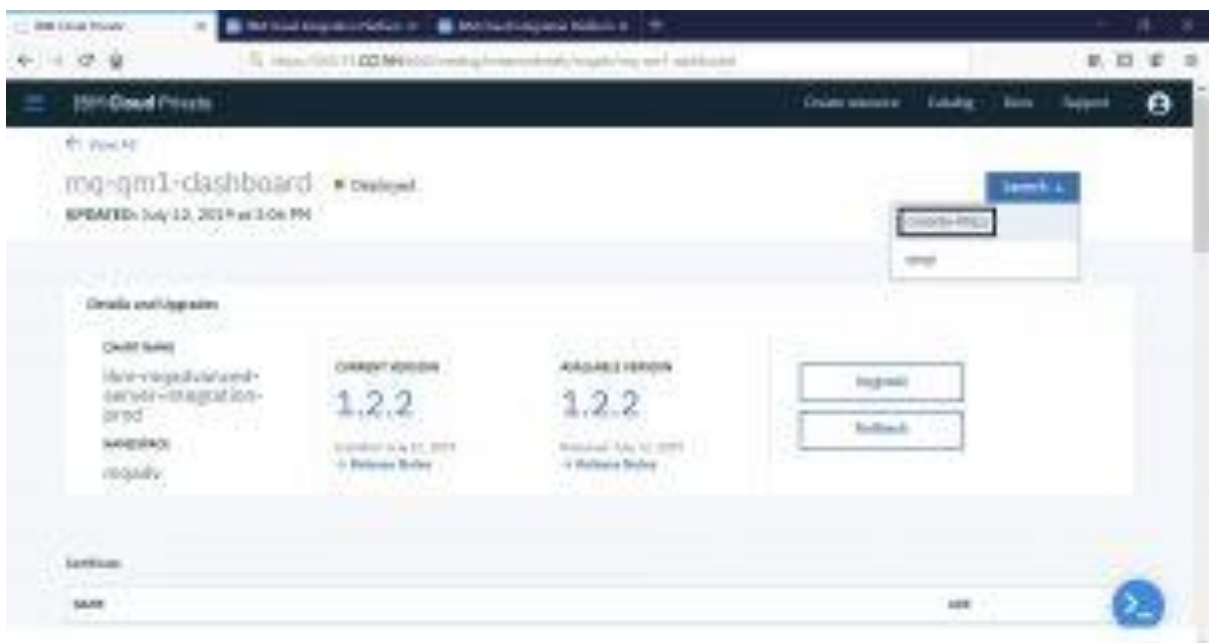
Search for keyword “mq” and click the helm release **mq-qm1-dashboard** (which was created in the previous step).



- Check the pod status. They should be in Running and Completed state, as shown below:
- The Service Port number is 32069. Mapping of queue manager listener port 1414 is used for external communication with queue manager.



- Click **Launch** → **console-https** to trust the MQ Dashboard URL.



To connect to the IBM MQ Dashboard, you need to trust the self-signed auto-generated certificate which is created during IBM MQ queue manager helm chart installation and deployment. Click **Advanced** and click **Accept the Risk and Continue**.

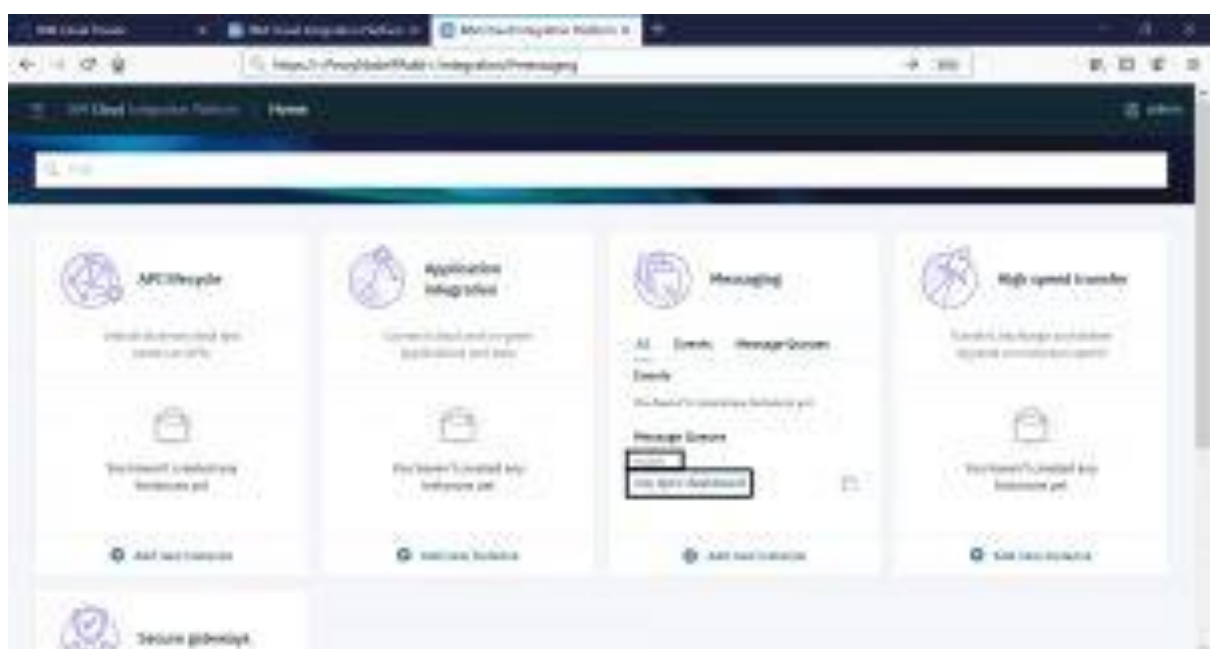


3. Create queue manager dashboard widget and its objects:

- Open the [IBM Cloud Integration Platform Console](#)

Under the Messaging Section, below Message Queues it shows link “mq-qm1-dashboard” created under mqadv namespace.

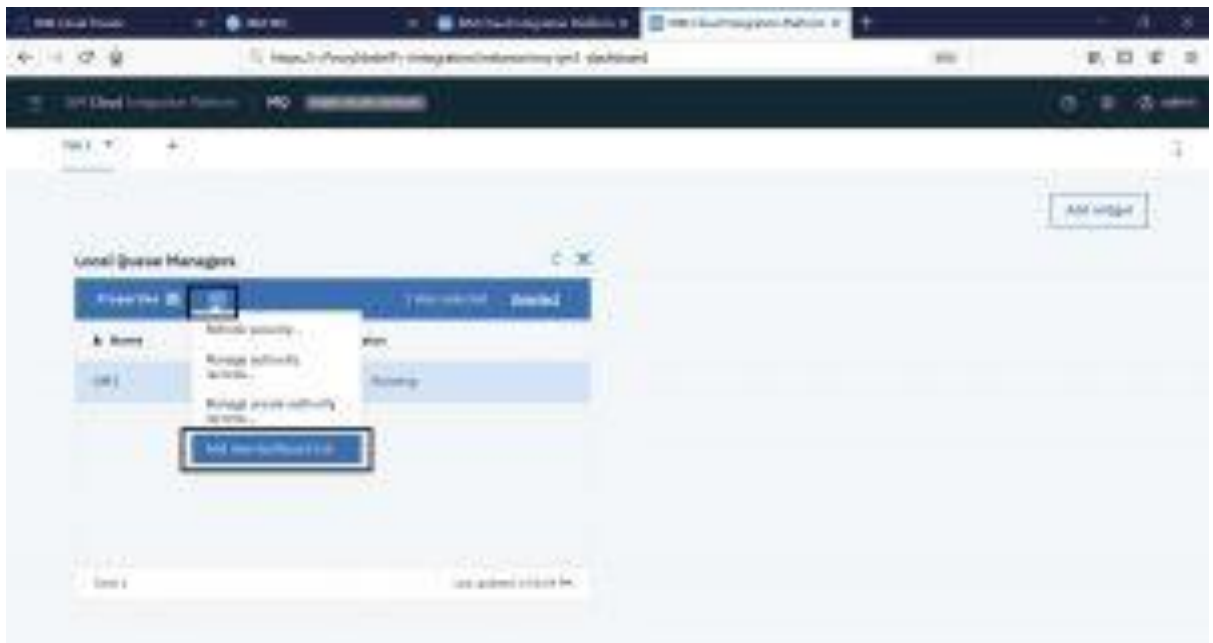
“mq-qm1-dashboard” link opens the QM1 queue manager dashboard.



- Click **mq-qm1-dashboard** under the Messaging section. The QM1 queue manager dashboard opens (shown below).

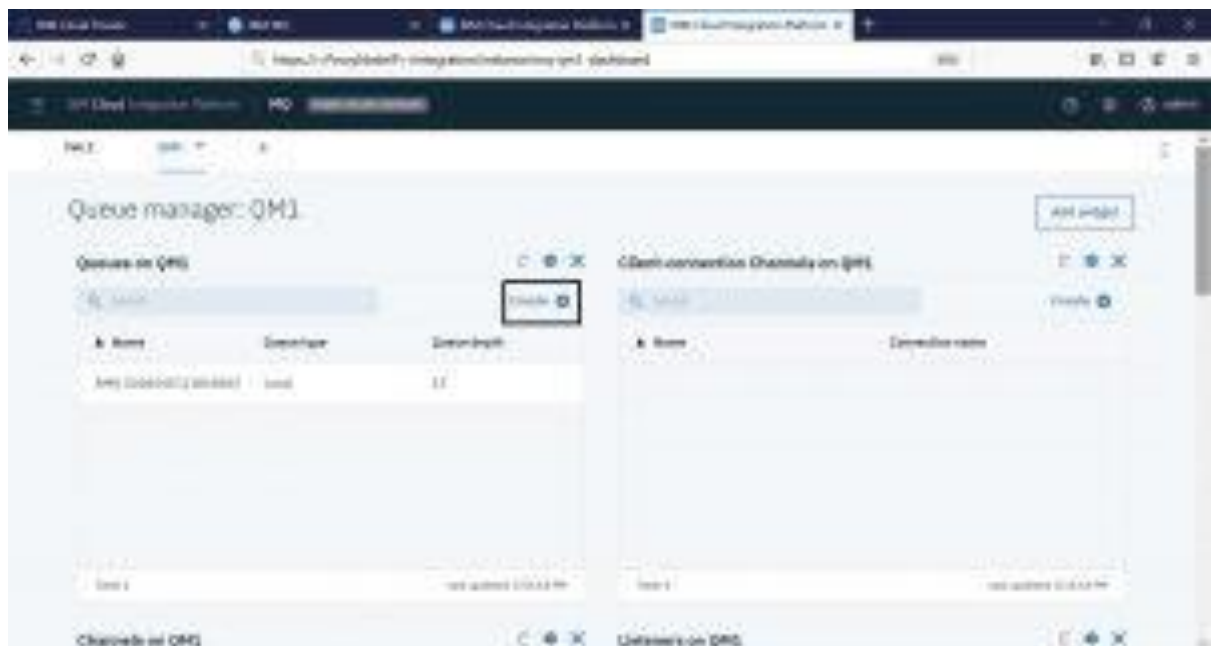


- Select the queue manager QM1 and Add new dashboard tab

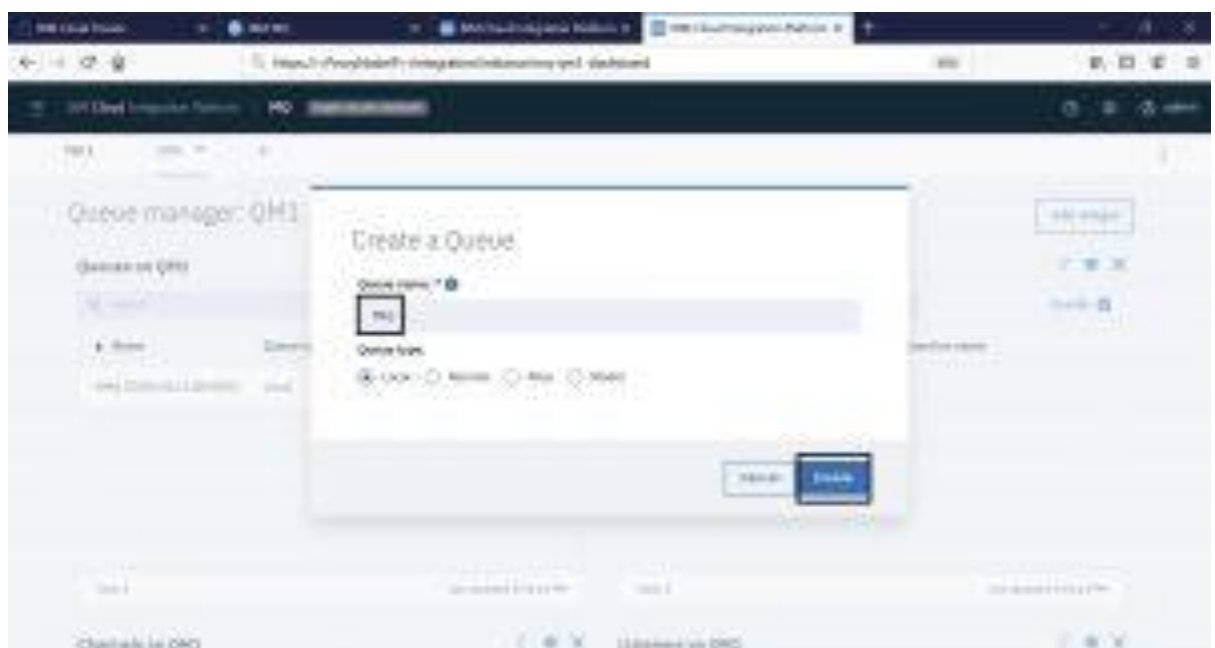


All queue manager object's types widget are added.

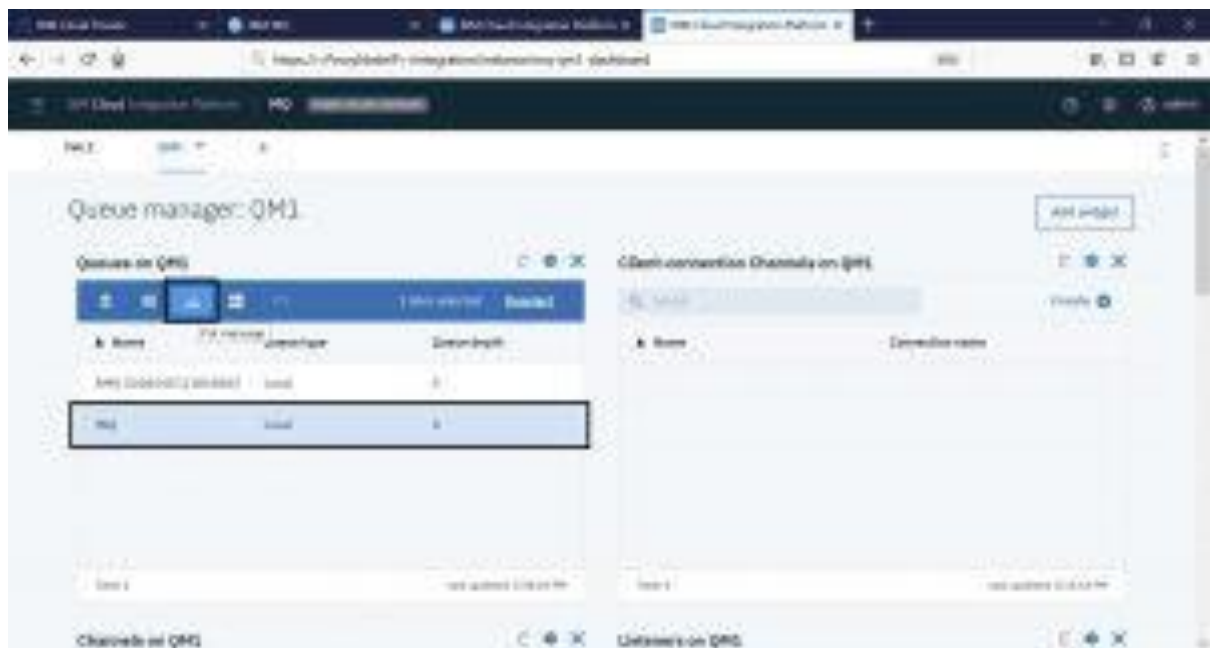
- To create a local queue click "Create" in the Queues Widget.



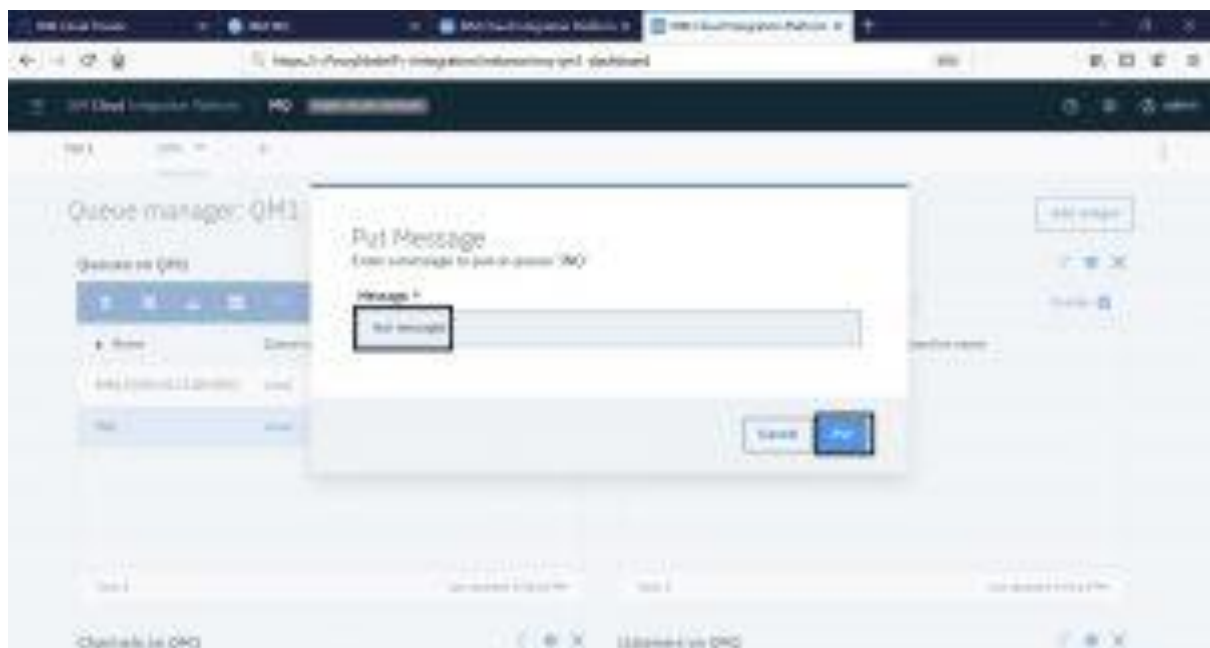
- Specify the queue name INQ and click **Create**.

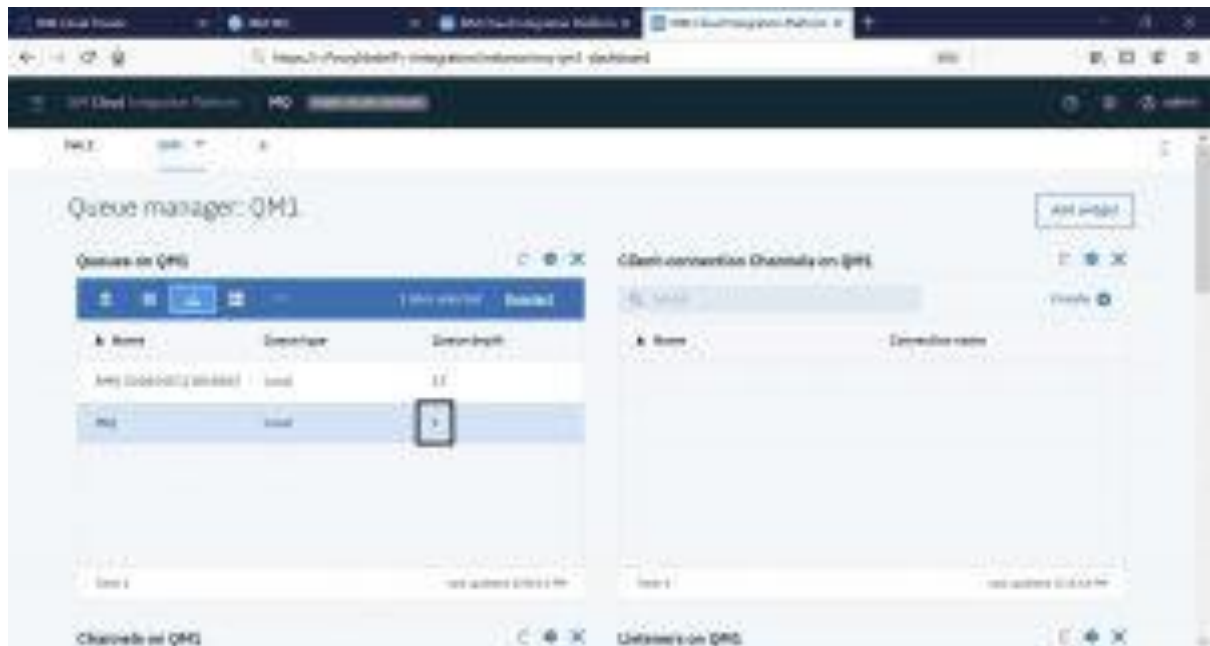


- To put a message into the queue use **Put** message shown below.

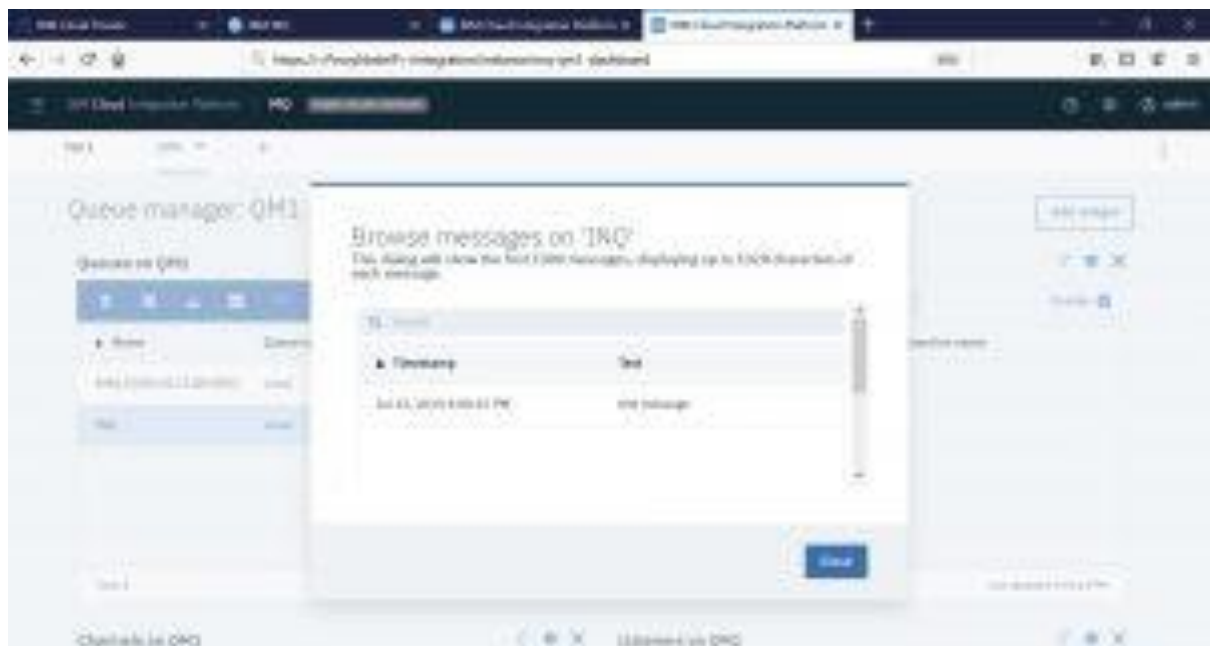


- Enter input message and click **Put**, as shown below.





To browse the messages on INQ, click **browse**.



Conclusion:

This article shows how to install, configure and deploy IBM MQ Advanced queue manager in IBM Cloud Pack for Integration. How to launch the queue manager dashboard and add queue manager widget. It also shows how to create queue manager objects local queue, put the sample test message and browse the message. Similarly you can create other queue manager objects like channels, Authentication Information etc. using this queue manager dashboard.

References:

- [IBM Cloud Pak for Integration Installation](#)
- [IBM Cloud Pak for Integration V2019.2.1 announcement](#)