



Manual de Implementación IBM MQ on Containers



Artefacto 2022

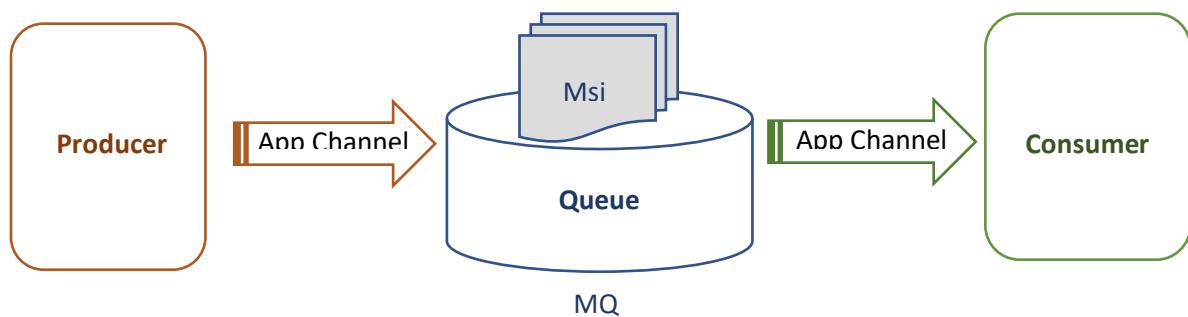
Elaborado por: Santiago Medina Morales

Índice

Introducción.....	3
Requerimientos Infraestructura	3
Implementación.....	4
Openshift Conectado - Cargue Catalogo IBM.....	4
Openshift Desconectado - Catalogo IBM Bastion Host	5
Prerrequisitos instalación	5
Download CASE files	5
Mirror de imágenes	6
Creación catalog Source	8
Cargue catalogo Redhat.....	9
Despliegue Operador IBM MQ	10
Creación de secret para pull image MQ	12
Prueba MQ.....	26
Configuración certificados TLS para Queue Manager deployment.....	29
Creación config map con comandos MQSC.....	29
Creación route Queue Manager	29
Creación Queue Manager desde yaml	30
Creación archivo CCDT.json.....	32
Prueba Storage	35
Adición de colas o modificación en configuraciones existentes (Manual).....	37
Despliegue Continuo – Tekton/Openshift pipelines	39
Cluster Privado Openshift	40
Tekton / Openshift Pipelines	40
Construcción de Imagen con cliente OC.....	40
Definición task	41
Definición Pipeline	42
Errores.....	44
Despliegue de Operador en namespace diferente a Openshift-operators.....	44
Ratificar el tipo de storage dependiendo del availability que se elija.....	45
Ejecución de pipeline con usuario default.....	45
Adicionales.....	45
Expand disk vm azure	45

Introducción

IBM MQ puede transportar cualquier tipo de datos como mensajes, lo que permite a las empresas crear arquitecturas flexibles y reutilizables, como entornos de arquitectura orientada a servicios (SOA) o en nuestro caso arquitecturas orientadas a microservicios. Funciona con una amplia gama de plataformas informáticas, aplicaciones, servicios web y protocolos de comunicación para la entrega de mensajes de alta seguridad. IBM MQ proporciona una capa de comunicaciones para la visibilidad y el control del flujo de mensajes y datos dentro y fuera de su organización.

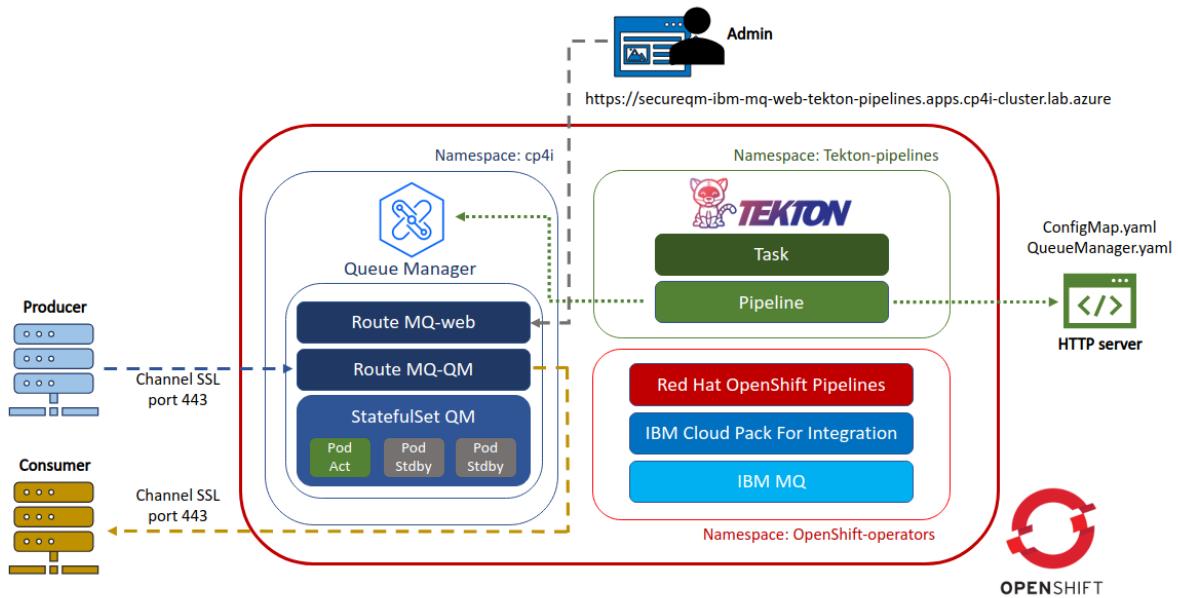


Requerimientos Infraestructura

Cantidad	Tipo	Referencia Azure	Detalle
3	Worker Nodes	Standard D16s v3 + Disco 512Gb (Storage Cluster OCS)**	VCPUs 16 RAM 64Gb
3	Master Nodes	Standard D8s v3	VCPUs 8 RAM 32Gb
1	Bastion - VM RHEL 8.2	Standard D2s v3 + Disco 64Gb (sistema) + 2048Gb (Repositorio Desconectado)	VCPUs 2 RAM 8Gb
1	Jumpbox - Windows	Standard D2s v3 + Disco 127Gb (sistema)	VCPUs 2 RAM 8Gb

** El tamaño del storage deberá estimarse de acuerdo al volumen transaccional de mensajería para MQ, en la sección de pruebas podrá visualizar un estimado.

Implementación



Openshift Conectado - Cargue Catalogo IBM

1. Nos dirigimos al boton +

The screenshot shows the Red Hat OpenShift Container Platform web interface. The sidebar is set to Operators > OperatorHub. The main view displays the OperatorHub page, which lists various operators available for installation. The operators listed include:

- 3Scale API Management (Community)
- Advanced Cluster Management for Kubernetes
- Advanced Cluster Security for Kubernetes
- Aerospike Kubernetes Operator (Marketplace)

2. Adicionamos catalogo IBM con el siguiente yaml

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: docker.io/ibmcom/ibm-operator-catalog
  updateStrategy:
```

registryPoll:
interval: 45m

3. Click en el botón Create
4. una vez desplegado dirigirse a Operators > Operator hub

Project: openshift-operators ▾

<input type="checkbox"/> Certified (191)	Decision Optimization Addon	CSV file for Hadoop	Operator to manage the lifecycle of IBM FCI and its components
<input type="checkbox"/> Community (199)			
<input type="checkbox"/> Marketplace (118)			
<input checked="" type="checkbox"/> ibm-operator-catalog (151)			
Provider			
<input type="checkbox"/> Red Hat (112)			
<input type="checkbox"/> APIMatic.io (1)			
<input type="checkbox"/> Aarna Networks (2)			
<input type="checkbox"/> Aerospike (2)			
<input type="checkbox"/> Alcide (1)			
Show 231 more			
Capability level			
<input type="checkbox"/> Basic Install (308)			
<input type="checkbox"/> Seamless Upgrades (274)			
<input type="checkbox"/> Full Lifecycle (72)			
<input type="checkbox"/> Deep Insights (51)			
<input type="checkbox"/> Auto Pilot (26)			
Infrastructure features			
<input type="checkbox"/> Disconnected (88)			
<input type="checkbox"/> Proxy-aware (16)			
<input type="checkbox"/> FIPS Mode (10)			

Financial Crimes Insight Alert Triage - Transaction List Screening provided by IBM FCI Operator to manage the lifecycle of IBM fcaii and its components

Financial Crimes Insight Claims Fraud provided by IBM FCI Operator to manage the lifecycle of IBM fcii and its components

FoundationDB Kubernetes Operator provided by IBM The Apple FDB kubernetes operator.

IBM Analytics Engine Powered by Apache Spark Service provided by cpd IBM Analytics Engine Powered by Apache Spark Service

IBM API Connect provided by IBM IBM® API Connect is a complete, modern, intuitive and scalable API platform that lets you create,...

IBM API Connect V2018 Upgrade provided by IBM Upgrade from V2018 description

Openshift Desconectado - Catalogo IBM Bastion Host

<https://www.ibm.com/docs/en/cloud-paks/cp-integration/2021.4?topic=cluster-adding-catalog-sources-bastion-host>

Prerrequisitos instalación [OBJ]

Software	Purpose
OpenSSL	Validating certificates when you run the air-gapped install scripts
Docker	Container management
Podman	Container management
httpd-tools	Creating an account when you run the air-gapped install scripts
cloudctl	Running CASE commands
oc	Red Hat OpenShift Container Platform administration
skopeo	Working with container images and registries in an air-gapped environment

Download CASE files

Para nuestro ejemplo vamos a utilizar el directorio \$HOME=/data

```
export CASE_NAME=ibm-integration-platform-navigator
export CASE_VERSION=1.6.1
export CASE_ARCHIVE=${CASE_NAME}-${CASE_VERSION}.tgz
export CASE_INVENTORY_SETUP=platformNavigatorOperator
export OFFLINEDIR=$HOME/offline
export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
```

```
cloudctl case save \
--repo $CASE_REPO_PATH \
--case $CASE_NAME \
--version $CASE_VERSION \
--outputdir $OFFLINEDIR
```

Mirror de imágenes

La imagen necesaria para la instalación es la de IBM MQ por lo cual vamos a definir las siguientes variables de entorno basados en eso.

```
export CASE_NAME=ibm-mq
export CASE_VERSION=1.8.1
export CASE_ARCHIVE=${CASE_NAME}-${CASE_VERSION}.tgz
export CASE_INVENTORY_SETUP=ibmMQOperator
```

```
cloudctl case launch \
--case $HOME/offline/$CASE_ARCHIVE \
--inventory $CASE_INVENTORY_SETUP \
--action configure-creds-airgap \
--args "--registry $LOCAL_DOCKER_REGISTRY --user admin --pass passw0rd"
```

```
cloudctl case launch \
--case $HOME/offline/$CASE_ARCHIVE \
--inventory $CASE_INVENTORY_SETUP \
--action configure-creds-airgap \
--args "--registry cp.icr.io --user cp --pass
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE2NDQ3NzQ
3MTEsImp0aSI6ljZhNzBiMTRjNTRjMTQ2ZTVhMTRkZGY5M2U0MmMwODRmln0.1PPrf5FWBT9c7hBITL
I506Z70RVB-o0L4m3zL8Q1Qbl" [REDACTED]
```

Configuración final de Mirror y configuración de cluster

```
export NAMESPACE=cp4i
oc create namespace $NAMESPACE
export CASE_NAME=ibm-cp-integration
export CASE_VERSION=2.4.0
```

```

export CASE_ARCHIVE=${CASE_NAME}-${CASE_VERSION}.tgz
export CASE_INVENTORY_SETUP=operator
export OFFLINEDIR=$HOME/offline
export OFFLINEDIR_ARCHIVE=offline.tgz
export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
export CASE_LOCAL_PATH=$OFFLINEDIR/$CASE_ARCHIVE
export LOCAL_DOCKER_REGISTRY_HOST=BastionOp.lab.azure
export LOCAL_DOCKER_REGISTRY_PORT=5000
export
LOCAL_DOCKER_REGISTRY=$LOCAL_DOCKER_REGISTRY_HOST:$LOCAL_DOCKER_REGISTRY_PORT
export LOCAL_DOCKER_USER=admin
export LOCAL_DOCKER_PASSWORD=passw0rd

```

```

cloudctl case launch \
--case $HOME/offline/$CASE_ARCHIVE \
--inventory $CASE_INVENTORY_SETUP \
--action configure-creds-airgap \
--args "--registry $LOCAL_DOCKER_REGISTRY --user $LOCAL_DOCKER_USER --pass
$LOCAL_DOCKER_PASSWORD"

```

Este proceso puede tomar Varias Horas

```

cloudctl case launch \
--case $CASE_LOCAL_PATH \
--inventory $CASE_INVENTORY_SETUP \
--action mirror-images \
--args "--registry $LOCAL_DOCKER_REGISTRY --inputDir $OFFLINEDIR --resource-group
ibmcpc4iDefault"
BastionOp.lab.azure:5000/
cpopen/ibm-integration-platform-navigator-catalog
manifests:
sha256:2b3fc924d130efd70012011130fdb5b7e2d47bd065b56c1dd08322ef42ceef39
sha256:5a3d8167ed2bf3b008346c7e1ff1116370374b066400bf250b331367944b935
sha256:760de49869bc27ebdea8f40a7e31b59ea2a7db71a61e4815a2c7e7d8160914f
sha256:760de49869bc27ebdea8f40a7e31b59ea2a7db71a61e4815a2c7e7d8160914f -> latest
☒ stats: shared=0 unique=0 size=0B
phase 0:
BastionOp.lab.azure:5000 cpopen/ibm-integration-platform-navigator-catalog blobs=0 mounts=0 manifests=4 shared=0
info: Planning completed in 30ms
sha256:5a3d8167ed2bf3b008346c7e1ff1116370374b066400bf250b331367944b935 BastionOp.lab.azure:5000/cpopen/ibm-integration-platform-navigator-catalog
sha256:2b3fc924d130efd70012011130fdb5b7e2d47bd065b56c1dd08322ef42ceef39 BastionOp.lab.azure:5000/cpopen/ibm-integration-platform-navigator-catalog
info: Mirroring completed in 30ms (0B/s)
sha256:760de49869bc27ebdea8f40a7e31b59ea2a7db71a61e4815a2c7e7d8160914f BastionOp.lab.azure:5000/cpopen/ibm-integration-platform-navigator-catalog:latest
[✓] CASE launch script completed successfully
OK

```

```

cloudctl case launch \
--case $HOME/offline/$CASE_ARCHIVE \
--inventory $CASE_INVENTORY_SETUP \
--action configure-cluster-airgap \
--namespace $NAMESPACE \
--args "--registry $LOCAL_DOCKER_REGISTRY --user $LOCAL_DOCKER_USER --pass
$LOCAL_DOCKER_PASSWORD --inputDir $OFFLINEDIR"

```

```

apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  name: ibm-cp-integration
spec:
  repositoryDigestMirrors:
    - mirrors:
        - BastionOp.lab.azure:5000/cp
      source: cp.icr.io/cp
    - mirrors:
        - BastionOp.lab.azure:5000/ibmcom
      source: docker.io/ibmcom
    - mirrors:
        - BastionOp.lab.azure:5000/appc-dev
      source: icr.io/appc-dev
    - mirrors:
        - BastionOp.lab.azure:5000/cpopen
      source: icr.io/cpopen
    - mirrors:
        - BastionOp.lab.azure:5000/integration
      source: icr.io/integration
    - mirrors:
        - BastionOp.lab.azure:5000/opencloudio
      source: quay.io/opencloudio
  ...
[INFO] Applying image content source policy
oc apply -f "/tmp/airgap_image_policy_Sp3jFdL6w"
imagecontentsourcepolicy.operator.openshift.io/ibm-cp-integration created
[✓] CASE launch script completed successfully
OK

```

```

oc get imageContentSourcePolicy
[azureuser@BastionOp offline]$ oc get imageContentSourcePolicy
NAME          AGE
ibm-cp-integration   5s
image-policy-0     12d
image-policy-1     12d

```

```

oc patch image.config.openshift.io/cluster --type=merge \
-p '{"spec":{"registrySources":[{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}]}}'
[azureuser@BastionOp offline]$ oc patch image.config.openshift.io/cluster --type=merge \
> -p '{"spec":{"registrySources":[{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}]}}'
image.config.openshift.io/cluster patched

```

```

Oc get nodes
[azureuser@BastionOp offline]$ oc get nodes
NAME          STATUS    ROLES   AGE   VERSION
cp4i-cluster-zwg95-master-0  Ready     master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-master-1  Ready,SchedulingDisabled  master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-master-2  Ready     master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus22-q4gh8 Ready     worker  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus22-rl4hk Ready,SchedulingDisabled  worker  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus24-d5s48 Ready     worker  12d  v1.21.8+ed4d8fd
[azureuser@BastionOp offline]$ oc get nodes
NAME          STATUS    ROLES   AGE   VERSION
cp4i-cluster-zwg95-master-0  Ready     master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-master-1  Ready     master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-master-2  Ready     master  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus22-q4gh8 Ready     worker  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus22-rl4hk Ready     worker  12d  v1.21.8+ed4d8fd
cp4i-cluster-zwg95-worker-eastus24-d5s48 Ready     worker  12d  v1.21.8+ed4d8fd

```

Creación catalog Source

```

export NAMESPACE=cp4i

```

```

cloudctl case launch \
--case ${HOME}/offline/${CASE_ARCHIVE} \

```

```
--inventory ${CASE_INVENTORY_SETUP} \
--action install-catalog \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --inputDir $HOME/offline --recursive"
```

```
36e1f662364f758058a2542ce41cf5e40d
catalogsource.operators.coreos.com/ibm-integration-operations-dashboard-catalog created
done
[✓] CASE launch script completed successfully
✖
operatorgroup.operators.coreos.com "common-service" deleted
-----Installing catalog source-----
# Will need to register the image name at cp.icr.io/cp/<image>, as the content clusters
# won't add the cp4i mapping for us.
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cp-integration-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Cloud Pak for Integration
  publisher: IBM
  sourceType: grpc
  image: BastionOp.lab.azure:5000/cpopen/ibm-cp-integration-catalog:1.4.0-2021-09-08-1314-df01d58b@sha256:70d6943b1b286301de18a302d4916f75dea4acdca01ffa
9471a877cb650322c
catalogsource.operators.coreos.com/ibm-cp-integration-catalog created
done
[✓] CASE launch script completed successfully
OK
```

```
oc get pods -n openshift-marketplace
oc get catalogsource -n openshift-marketplace
```

Cargue catalogo Redhat

```
podman login registry.redhat.io
podman login -u admin -p passw0rd https://BastionOp.lab.azure:5000
```

```
[azureuser@BastionOp ~]$ podman login -u admin -p passw0rd https://BastionOp.lab.azure:5000
WARNING! Using --password via the cli is insecure. Please consider using --password-stdin
Login Succeeded!
[azureuser@BastionOp ~]$
[azureuser@BastionOp ~]$ cat ${XDG_RUNTIME_DIR}/containers/auth.json
{
    "auths": {
        "BastionOp.lab.azure:5000": {
            "auth": "YWRTaW46cGFzc3cwcmQ="
        },
        "registry.redhat.io": {
            "auth": "c21lZGluYW1AdW5hbC5lZHUUY286WmV1czIxMDM="
        }
    }
}
```

```
REG_CREDS=${XDG_RUNTIME_DIR}/containers/auth.json
```

```
oc adm catalog mirror registry.redhat.io/redhat/redhat-operator-index:v4.6
BastionOp.lab.azure:5000/olm-mirror -a ${REG_CREDS} --insecure
```

Una vez finalizado encontraremos una carpeta con la definición del catalogo

```
[azureuser@BastionOp manifests-redhat-operator-index-1651700129]$ pwd
/home/azureuser/manifests-redhat-operator-index-1651700129
[azureuser@BastionOp manifests-redhat-operator-index-1651700129]$ ls -l
total 1648
-rwxrwxr-x. 1 azureuser azureuser 241 May  5 03:22 catalogSource.yaml
-rwxrwxr-x. 1 azureuser azureuser 116410 May  5 03:22 imageContentSourcePolicy.yaml
-rw-rw-r--. 1 azureuser azureuser 1564290 May  5 03:22 mapping.txt
```

```
oc create -f <path/to/manifests/dir>/imageContentSourcePolicy.yaml
oc apply -f catalogSource.yaml
```

```
oc get catalogsource -n openshift-marketplace
```

NAME	DISPLAY	TYPE	PUBLISHER	AGE
appconnect-operator-catalogsource	IBM App Connect operator	grpc	IBM	41h
aspera-operators	Aspera Operators	grpc	IBM	41h
couchdb-operator-catalog	Couchdb Operator Catalog	grpc	IBM	41h
ibm-ai-wmltraining-operator-catalog	WML Core Training	grpc	IBM	41h
ibm-apiconnect-catalog	IBM APIConnect catalog	grpc	IBM	41h
ibm-automation-foundation-core-catalog	IBM Automation Foundation Core Operators	grpc	IBM	41h
ibm-cloud-databases-redis-operator-catalog	ibm-cloud-databases-redis-operator-catalog	grpc	IBM	41h
ibm-cp-integration-catalog	IBM Cloud Pak for Integration	grpc	IBM	41h
ibm-datapower-operator-catalog	DataPower Operator	grpc	IBM Content	41h
ibm-eventstreams	Event Streams Operators	grpc	IBM	41h
ibm-integration-asset-repository-catalog	IBM CP4I Asset Repository	grpc	IBM	41h
ibm-integration-operations-dashboard-catalog	IBM CP4I Operations Dashboard	grpc	IBM	41h
ibm-integration-platform-navigator-catalog	IBM CP4I Platform Navigator	grpc	IBM	41h
ibmmq-operator-catalogsource	IBM MQ	grpc	IBM	46h
opencloud-operators	IBMCLOUD Operators	grpc	IBM	41h
redhat-operator-index		grpc		58m

Despliegue Operador IBM MQ

<https://www.ibm.com/docs/en/ibm-mq/9.1?topic=openshift-installing-mq-operator-using-web-console>

1. En el panel de Navegación dirigimos a home > projects
2. Creamos un nuevo proyecto con el botón Create Project y le damos un nombre y descripción por ejemplo nombre: tekton-pipelines descripción: tekton-pipelines

A screenshot of the OpenShift web console showing a new project named "tekton-pipelines". The project details are as follows:

PR	tekton-pipelines	No display name	✓ Active	kube:admin
----	------------------	-----------------	----------	------------

3. En el panel de navegación nos dirigimos a Operators > OperatorHub
4. En el cuadro de filtro buscamos por cloudpak for Integration y damos click en Install

A screenshot of the OperatorHub showing search results for "Cloud Pak for Integration". A red box highlights the entry for "IBM Cloud Pak for Integration".

Cloud Pak for Integrati
IBM API Connect provided by IBM IBM® API Connect is a complete, modern, intuitive and scalable API platform that lets you create,...
IBM API Connect V2018 Upgrade provided by IBM Upgrade from V2018 description
IBM App Connect provided by IBM The tool for easily connecting apps, integrating data, and building APIs.
IBM Automation Foundation assets (previously IBM Cloud Pak for Integration Asset Repository) provided by IBM Provides a repository to quickly store, manage, retrieve and search integration assets, across...
IBM Cloud Pak for Integration provided by IBM Install and manage the Cloud Pa... Installed
IBM Cloud Pak for Integration Operations Dashboard provided by IBM Cross-component transaction...

5. En el cuadro de filtro buscamos por IBM MQ y damos click en Install

The screenshot shows a search interface for a catalog. On the left, there is a vertical list of categories: All Items, AI/Machine Learning, Application Runtime, Automation, Big Data, Cloud Provider, Database, Developer Tools, Development Tools, Drivers And Plugins, Integration & Delivery, Logging & Tracing, and Modernization & Migration. On the right, the search results are displayed. A search bar at the top contains the text 'ibm MQ'. Below the search bar, the results are shown in a card format. The card has a title 'ibm MQ provided by IBM', a small icon of a key and a lock, and a brief description: 'IBM MQ is an operator to manage the life cycle of IBM MQ queue managers. This operator is...'. The entire interface is set against a light gray background.

6. Seleccionamos la última versión disponible y en Namespace dejamos el por defecto para desplegarlo en todos los Namespaces.
7. Una vez finalizada la instalación en el menú Operators > Installed Operators nos deberá aparecer el Operador de IBM MQ, CloudPak for Integration (al instalar este operador también despliega de forma automática el IBM Automation Foundation Core y IBM Cloud Pak foundational Services).

Nota: El operador IBM Cloud Pak for Integration Operations Dashboard se utilizará en la sección de despliegue y el operador de RedHat Openshift se utilizará en la sección de Pruebas MQ.

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [ClusterServiceVersion documentation](#).

Name	Managed Namespaces
 IBM Automation Foundation Core	All Namespaces
1.3.6 provided by IBM	
 IBM Cloud Pak foundational services	All Namespaces
3.17.0 provided by IBM	
 IBM Cloud Pak for Integration Operations Dashboard	All Namespaces
2.5.5 provided by IBM	
 IBM Cloud Pak for Integration	All Namespaces
5.3.1 provided by IBM	
 IBM MQ	All Namespaces
1.8.1 provided by IBM	
 Red Hat OpenShift Pipelines	All Namespaces
1.5.2 provided by Red Hat	

Nota: el despliegue del Operador únicamente se pudo hacer en el namespace Openshift Operators esto debido a que por defecto al momento de la creación hace un llamado a un servicio ***ibm-mq-webhook.openshift-operators.svc*** ([ver error](#)). Este endpoint no es modificable al momento de realizar la instalación por lo cual se deja como una limitante del producto.

Creación de secret para pull image MQ

<https://www.ibm.com/docs/en/ibm-mq/9.2?topic=pyrhomp-preparing-your-red-hat-openshift-project-mq-using-red-hat-openshift-web-console>

Procedure

1. Get the entitlement key that is assigned to your ID.
 - a. Log in to [MyIBM Container Software Library](#) with the IBM ID and password that are associated with the entitled software.
 - b. In the **Entitlement keys** section, select **Copy key** to copy the entitlement key to the clipboard.
2. Create a secret containing your entitlement key, in the project where you want to deploy your queue manager.
 - a. From the navigation pane, click on **Workloads > Secret**.
The Secrets page is displayed.
 - b. In the **Project** drop-down, select the project into which you want to install IBM MQ
 - c. Click the **Create** button, and select **Image Pull Secret**
 - d. In the **Name** field, enter `ibm-entitlement-key`
 - e. In the **Registry Server Address** field, enter `cp.icr.io`
 - f. In the **Username** field, enter `cp`
 - g. In the **Password** field, enter the entitlement key you copied in the previous step
 - h. In the **Email** field, enter the IBM ID associated with the entitled software

Para obtener el password ingresar a la siguiente ruta y copiar el key

<https://myibm.ibm.com/products-services/containerlibrary>

Access container software

Below is an entitlement key you can use to verify and gain access to container software. Once you purchase container software, it'll appear in your library.

Entitlement key

Issued Date: April 22, 2022

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE2NTA2NDE0NDAsImp0aSI6IjIyNDc4YzU2NGI5YjQzMjFhZGQ1OTFiZThkYTE0NzUyIn0.DjP--bsdDUoJLOuUkVZngs4GIfV-7iotGSjeyql4Cs8
```

[Copy key](#)

[Get new key](#)

De esta forma debería quedar creado nuestro secret en el namespace Openshift-operators

Project: openshift-operators ▾

Edit key/value secret

Secret name *

ibm-entitlement-key

Unique name of the new secret.

Registry server address *

cp.icrio

For example quay.io or docker.io

Username *

cp

Password *

.....

Email

ferhernande@gbm.net

8. Ingresamos en el IBM MQ y procedemos con la creación del Queue Manager

Installed Operators > Operator details



IBM MQ

18.0 provided by IBM

[Details](#) [YAML](#) [Subscription](#) [Events](#) [Queue Manager](#)

Provided APIs

Queue Manager

A QueueManager is an IBM MQ server which provides queuing and publish/subscribe services to applications

[Create instance](#)

9. En este apartado vamos a diligenciar los campos nombre, etiquetas, versión

Project: openshift-operators ▾

IBM MQ > Create QueueManager

Create QueueManager

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 *
quickstart-cp4

Labels
app=frontend

License *
Settings that control your acceptance of the license, and which license metrics to use

Version *
9.2.5.0-r1

Setting that controls the version of MQ that will be used (required). For example: "9.15.0-r2" would specify MQ version 9.15.0, using the second revision of the container image. Container-specific fixes are often applied in revisions, such as fixes to the base image.

Queue Manager
Settings for the Queue Manager container and underlying Queue Manager

Tracing
Settings for tracing integration with the Cloud Pak for Integration Operations Dashboard

Web
Settings for the MQ web server

» Advanced configuration

10. Licencia

Dependerá de aquella que el cliente haya adquirido para mayor detalle en el siguiente link encontrará el listado de las licencias disponibles

<https://www.ibm.com/docs/en/ibm-mq/9.2?topic=mqibmcomv1beta1-licensing-reference>

Habilitamos la opción Accept, seleccionamos la licencia y dependiendo de la misma en el apartado License use podremos seleccionar entre (Production, NonProduction, development)

License *

Settings that control your acceptance of the license, and which license metrics to use

License acceptance *

accept
Whether or not you accept the license associated with this software (required)

License *

L-RJON-C7QG3S

The identifier of the license you are accepting. This must be the correct license identifier for the version of MQ you are using. See <http://ibm.biz/BdqvCF> for valid values.

License use

NonProduction

Setting that controls how the software will be used, where the license supports multiple uses. See <http://ibm.biz/BdqvCF> for valid values.

▼ Advanced configuration

License metric

Select License metric ▾

ManagedVirtualServer license metric to use. For example, "ProcessorValueUnit", "VirtualProcessorCore" or "ManagedVirtualServer". Defaults to "ProcessorValueUnit" when using an MQ "core" when using a Cloud Pak for Integration license

ProcessorValueUnit

VirtualProcessorCore

11. Queue Manager

Queue Manager

Settings for the Queue Manager container and underlying Queue Manager

Availability

Availability settings for the Queue Manager, such as whether or not to use an active-standby pair or native high availability

Type of availability

NativeHA

The type of availability to use. Use "SingleInstance" for a single Pod, which will be restarted automatically (in some cases) by Kubernetes. Use "MultiInstance" for a pair of Pods, one of which is the "active" Queue Manager, and the other of which is a standby. Use "NativeHA" for native high availability replication (requires MQ Operator 1.5.0 or higher). Defaults to "SingleInstance". See <http://ibm.biz/BdqAQa> for more details.

▼ Advanced configuration

Update strategy

Select Update strategy ▾

The update strategy to use for MultiInstance and NativeHA Queue Managers. Use "RollingUpdate" to enable automatic rolling updates whenever the Queue Manager configuration changes. Use "OnDelete" to disable automatic rolling updates. Queue Manager changes will only be applied when Pods are deleted (including Pod deletions triggered by external factors). Defaults to "RollingUpdate". Requires MQ Operator 1.6.0 or higher.

TLS

Optional TLS settings for configuring secure communication between NativeHA replicas. Requires MQ Operator 1.5.0 or higher.

Storage

Storage settings to control the Queue Manager's use of persistent volumes and storage classes

► Advanced configuration

El tipo de disponibilidad a utilizar. Utilice "Instancia única" para un único pod, que Kubernetes reiniciará automáticamente (en algunos casos). Utilice "MultiInstance" para un par de Pods, uno de los cuales es el Administrador de colas "activo" y el otro es un modo de espera. Utilice "NativeHA" para la replicación nativa de alta disponibilidad (requiere MQ Operator 1.5.0 o superior) donde desplegará por medio de statefull set 3 pods dos de los cuales quedarán en estado pasivo y uno activo. El valor predeterminado es "Instancia única"

SingleInstance

Multi-Instance

secureqm-ibm-mq-0	Running	1/1	0	secureqm-ibm-mq
secureqm-ibm-mq-1	Running	0/1	0	secureqm-ibm-mq

Native-HA

StatefulSets > StatefulSet details

nativeha-ibm-mq
Managed by nativeha

Details Metrics YAML **Pods** Environment Events

Filter Name Search by name...

Name	Status	Ready	Restarts	Node
nativeha-ibm-mq-0	Running	1/1	0	cp4i-cluster-gl4xx-worker-eastus2l-t87lq
nativeha-ibm-mq-1	Running	0/1	0	cp4i-cluster-gl4xx-worker-eastus2l-hs4dm
nativeha-ibm-mq-2	Running	0/1	0	cp4i-cluster-gl4xx-worker-eastus2l-j8hlj

Update strategy: se recomienda usar RollingUpdate con el fin de actualizar los pods sin generar fuera de servicio y garantizando la estabilidad de la nueva versión. O existe la opción OnDelete que deshabilita las actualizaciones continuas automáticas, los cambios del QM solo se aplicarán cuando se eliminan los pods (incluidas las eliminaciones de pods provocadas por factores externos).

Habilitar la opción Default delete Claim con el fin de garantizar la depuración el cluster

Persistent data: si se requiere que las colas y mensajes sean persistentes seleccionar un persistem claim (necesario para configuración multi-Instance)

MQSC: nos da la posibilidad de cargar mediante un Configmap/secret instrucciones MQSC por ejemplo para la creacion de Colas, restriccción de usuarios, creacion de Canales, entre otros.

Name: Nombre de nuestro QueueManager, este campo es completamente diferente a la inicial. El primer campo es para el nombre del objeto en Openshift, este segundo campo de nombre será el que quede registrado a nivel interno y el que necesitaremos al momento de establecer una conexión entre Consumidor/productor y Cola.

Resources: Este apartado es de suma importancia ya que nos permitirá modificar los recursos con los que se desplegará cada uno de los pods. En caso de observar problemas de rendimiento se deberá analizar y verificar estos valores tanto de límites como request.

The screenshot shows a user interface for configuring a Queue Manager. It includes sections for 'INI' (with a note about requiring MQ Operator 1.1.0 or higher) and 'MQSC' (with a note about requiring MQ Operator 1.1.0 or higher). A button labeled 'Add MQSC' is present. Below these, there is a 'Name' field containing 'QUICKSTART' and a note about underlying MQ Queue Manager names. Under 'Resources', there are two tables: 'Limits' and 'Requests', both defining resource requirements for CPU Cores, Memory, and Storage.

	CPU CORES	MEMORY	STORAGE
Limits	500m	50Mi	50Mi
Requests	500m	50Mi	50Mi

Storage: dependiendo del tipo de disponibilidad debemos elegir un storageclass con característica RWX (read Write Many) o RWO (Read Write Once) si es single instance o NativeHA. Si es Multilnstance únicamente se admite RWX. Para este ejemplo se utilizo el storage class de OCS (ocs-storagecluster-cephfs)

Si requerimos que nuestros mensajes sean o no persistentes debemos definir los campos en la siguiente sección.

No persistente : En la sección Type of Volume y teniendo en cuenta lo anterior respecto al tipo de disponibilidad deberá ser o Single instance o NativeHA para poder seleccionar Ephimeral

Persistente: Debemos seleccionar el default Class que sea de tipo RMX, seguido de la especificación del type of volume: Persistent-claim

Storage

Storage settings to control the Queue Manager's use of persistent volumes and storage classes

Default class

ocs-storagecluster-cephfs

Storage class to apply to all persistent volumes of this Queue Manager by default. Specific persistent volumes can define their own storage class which will override this default storage class setting. If "type of availability" is "SingleInstance" or "NativeHA", storage class can be of type ReadWriteOnce or ReadWriteMany. If "type of availability" is "MultiInstance", storage class must be of type ReadWriteMany.

Queue Manager

Default PersistentVolume for any data normally under '/var/mqm'. Will contain all persisted data and recovery logs, if no other volumes are specified.

▼ Advanced configuration

Default delete claim

true

Whether or not all the volumes should be deleted when the queue manager is deleted. Specific persistent volumes can define their own value for deleteClaim which will override this defaultDeleteClaim setting. Defaults to false.

Persisted data

PersistentVolume details for MQ persisted data, including configuration, queues and messages. Required when using multi-instance Queue Manager.

Type of volume

Select Type of volume ▾

- ephemeral
- persistent-claim

ephemeral

persistent-claim

Enabled

true

Whether or not this volume should be enabled as a separate volume, or be placed on the default "queueManager" volume. Defaults to false.

Class

ocs-storagecluster-cephfs

Storage class to use for this volume. Only valid if "type" is "persistent-claim". If "type of availability" is "SingleInstance" or "NativeHA", storage class can be of type ReadWriteOnce or ReadWriteMany. If "type of availability" is "MultiInstance", storage class must be of type ReadWriteMany.

Delete Claim

false

Whether or not this volume should be deleted when the queue manager is deleted.

Se podrá definir un claim independiente si se requiere para colas y mensajes o simplemente aprovisionar uno para QM, Queue y Messages.

Persisted data

PersistentVolume details for MQ persisted data, including configuration, queues and messages. Required when using multi-instance Queue Manager.

▼ Advanced configuration

Enabled

false

Whether or not this volume should be enabled as a separate volume, or be placed on the default "queueManager" volume. Defaults to false.

Recovery logs

Persistent volume details for MQ recovery logs. Required when using multi-instance Queue Manager.

► Advanced configuration

12. Tracing - Si no desea habilitar el Operator Dashboard el valor por defecto para este parámetro sera False.

<https://www.ibm.com/docs/en/ibm-mq/9.2?topic=operator-integrating-cloud-pak-integration-operations-dashboard>

Prerrequisitos

- Contar con el operador de CloudPakfor integration
- Implementar el Platform UI del CP4I

The screenshot shows the 'OperatorHub' section of the IBM Cloud Pak for Integration interface. Under 'Installed Operators', the 'IBM Cloud Pak for Integration' operator is listed. The 'Platform UI' tab is selected. Below it, a table titled 'PlatformNavigators' lists one entry: 'integration-quickstart' (Kind: PlatformNavigator, Status: Ready). A 'Create PlatformNavigator' button is visible at the top right of the table area.

Una vez implementado el Platform UI vamos a poder tener la administracion a nivel consola centralizada de todos nuestros Queue Managers en los diferentes Namespaces, es importante tener en cuenta que deshabilitara y eliminara los Routes de acceso Web asociados a cada queue manager.

The screenshot shows the 'Integration tracing' section of the IBM Cloud Pak for Integration Operations Dashboard. It displays a table with two rows: 'holo2' (Error) and 'cp4i-od-dev'. In the 'Messaging' section, a list of namespaces is shown: 'ibm-common-services', 'ibm-common-services' (selected), 'openshift-operators', and 'tekton-pipelines'. The 'ibm-common-services' item is highlighted with a red box.

- Instalar el Operador de Cloud Pak for integration Dashboard

Nota: Tener en cuenta que este módulo también debe ser licenciado de forma independiente.

IBM Cloud Pak foundational services 3.17.0 provided by IBM	All Namespaces	Succeeded Up to date	May 9, 2022, 10:37 PM	CommonService	⋮
IBM Cloud Pak for Integration Operations Dashboard 2.5.5 provided by IBM	All Namespaces	Succeeded Up to date	May 16, 2022, 7:36 PM	Operations Dashboard OperationsDashboardServiceBinding	⋮
IBM Cloud Pak for Integration	All Namespaces	Succeeded	May 9, 2022, 10:59 PM	Platform UI	⋮

En el yaml de Queue Manager en la sección spec debe adicionar las siguientes líneas

```
spec:
```

```

tracing:
  enabled: true
  namespace: <Operations_Dashboard_Namespace>

```

Una vez inicie la ejecución del pod, quedara en este estado

Name	Status
secureqm-ibm-mq-0	CreateContainerConfigError

Namespace	Restart policy
tekton-pipelines	Always Restart

Labels	Edit	Active deadline seconds
app.kubernetes.io/component=integration		Not configured
app.kubernetes.io/managed-by=operator		
app.kubernetes.io/version=9.2.5.0		
controller-revision-hash=secureqm-ibm-mq-f46ff8896		
statefulSetName=secureqm-ibm-mq		
statefulset.kubernetes.io/pod-name=secureqm-ibm-mq-0		

Node selector	Node
No selector	cp4i-cluster-gl4xx-worker-eastus21-t87lq

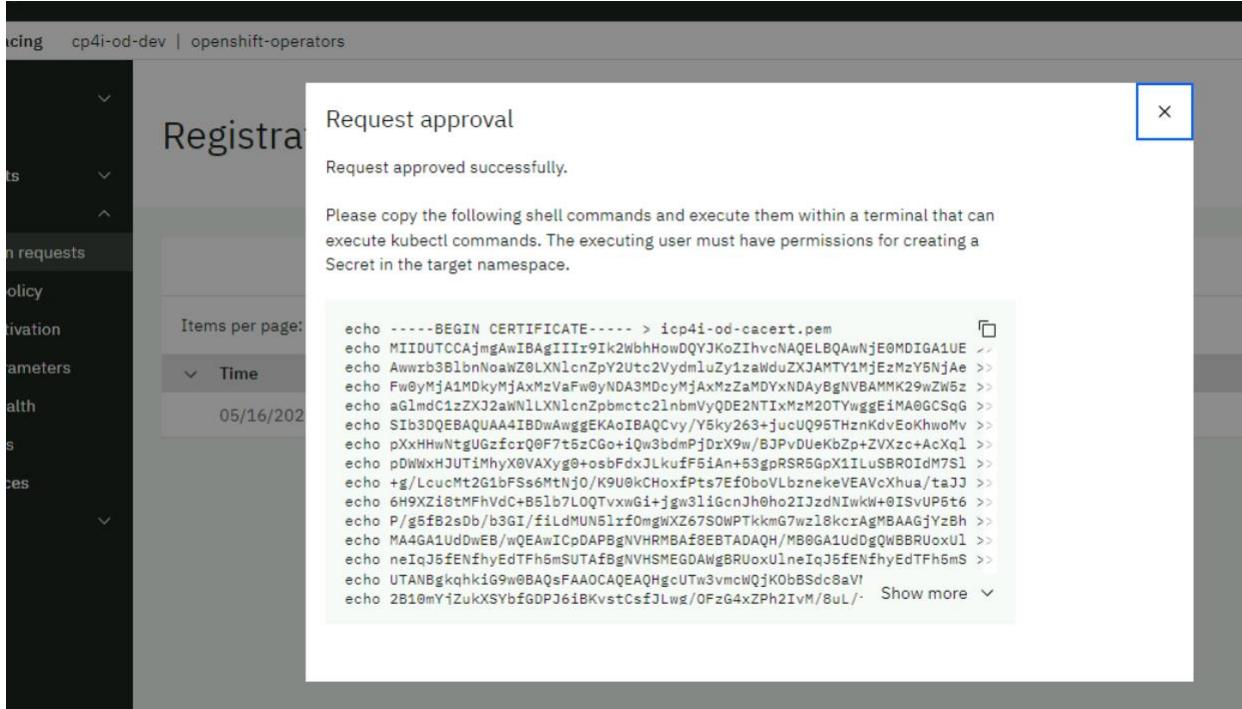
Hacer click en el boton aprobar

Time	Namespace	Pod name	Status
05/16/2022 20:32:36	tekton-pipelines	secureqm-ibm-mq-tracing-reg-bsmb9	New

Copiamos la salida del request approval y lo pegamos en nuestra maquina Bastion previamente logeado al cluster de openshift

Este procedimiento asegura la integración del Operation Dashboard por Namespace. Esto quiere decir que si realizamos esta configuración inicialmente y posterior desplegamos Queue Managers adicionales sobre el mismo namespace no deberá repetirse este procedimiento. Pero si por el contrario creamos un

nuevo namespace y realizamos la creación de un nuevo Queue Manager si deberá repetir este procedimiento.



```
echo -----BEGIN CERTIFICATE----- > icp4i-od-cacert.pem
echo MIIDUTCCAJmgAwIBAgIIR9Ik2WbhHowDQYJKoZIhvcNAQELBQAwnjE0MDIGA1UE >> icp4i-od-cacert.pem
echo Awwrb3BlbnNoaWZ0LXNlcnPzY2Utc2VydmluZy1zaWduZXJAMTY1MjEzMzY5NjAe >> icp4i-od-cacert.pem
echo Fw0yMjA1MDkyMjAxMzVaFw0yNDA3MDcyMjAxMzZaMDYxNDAYBgNVBAMMK29wZW5z >> icp4i-od-cacert.pem
echo aGImdC1zZXJ2aWNILXNlcnPbmctc2lnbmVyQDE2NTIxMzM2OTYwggEiMA0GCSqG >> icp4i-od-cacert.pem
echo S1b3DQEBAQUAA4IBDwAwggEKAoIBAQCVy/Y5ky263+jucUQ95THznKdvEoKhwoMv >> icp4i-od-cacert.pem
echo pXxHHwNtgUGzfcrQ0F7t5zCGo+iQw3bdmPjDrX9w/BJPvDUeKbZp+ZVXzc+AcXql >> icp4i-od-cacert.pem
echo pDWWxHJUTiMhyX0VAXyg0+osbFdxJLkufF5iAn+53gpRSR5GpX1LuSBROldM7SI >> icp4i-od-cacert.pem
echo +g/LcucMt2G1bFSs6MtNjO/K9U0kCHoxfPts7EfOboVLbznekeVEAvCxhua/taJJ >> icp4i-od-cacert.pem
echo 6H9XZi8tMFhVdC+B5lb7LOQTvxwGi+jgw3liGcnJh0ho2IJzdNIwkW+0ISvUP5t6 >> icp4i-od-cacert.pem
echo P/g5fB2sDb/b3GI/fiLdMUN5lrfOmgWXZ67SOWPTkkmG7wzl8kcrAgMBAAGjYzBh >> icp4i-od-cacert.pem
echo MA4GA1UdDwEB/wQEAWICpDAPBgnVHRMBAf8EBTADAQH/MB0GA1UdDgQWBKRUoxUI >> icp4i-od-cacert.pem
echo nIqJ5fENfhEdTFh6MSUTAEbgNVHSMEGDAwBgBRUoxUlneIqJ5fENfhEdTFh6MS >> icp4i-od-cacert.pem
echo UTANBgkqhkiGw0BAQsFAAACQEAQhgcuTw3vmcWqjKobBSdc8aVl >> icp4i-od-cacert.pem
echo 2B10mYiZukXSYbfGDPJ6iBKvstcsfJLwg/OFZg4xPh2IvM/8uL/- Show more
```

```

echo nelqJ5fENfhEdTFh5mSUTAfBgNVHSMEGDAWgBRUoxUlneIqJ5fENfhEdTFh5mS >> icp4i-od-
cacert.pem
echo UTANBgkqhkiG9w0BAQsFAAOCAQEAQHgcUTw3vmcWQjKObBSdc8aVMrgsVja17Rx7 >> icp4i-od-
cacert.pem
echo 2B10mYjZukXSYbfGDPJ6iBKvstCsfJLwg/OFzG4xZPh2lvM/8uL/jCBda8D4kVZc >> icp4i-od-
cacert.pem
echo +3fz4fL+ghJ9QjKCXNevPqH2rV/IRJWAhdK9drqShfut3Ryjt/Nu0kkfjvwVv4V2 >> icp4i-od-
cacert.pem
echo M9GbUAb6UZFgu9CRYueO+LvJqoRdheEV7iQdp8K96c7zHHcGy/ZszXNEfDiz0Opz >> icp4i-od-
cacert.pem
echo 6f7rF7q0bGMiDAhQc7iqdG3UKedA/iOvXTxdU0x0OIshnsvMO4YThZDjvMjIBdf >> icp4i-od-
cacert.pem
echo J3gJOq4BcqT9VhSb67iBK7T2WBZYEKswD0Um7UKcTs5k9D0Chw== >> icp4i-od-cacert.pem
echo -----END CERTIFICATE----- >> icp4i-od-cacert.pem
kubectl delete secret icp4i-od-store-cred -n tekton-pipelines
kubectl create secret generic icp4i-od-store-cred --from-literal=password=e8Dz+Dj#isUdras0 --from-
literal=username=OD_tekton-pip_TCSz30 --from-literal=tracingUrl=https://cp4i-od-dev-ibm-
integration-operations-dashboard-str-api-svc.openshift-operators.svc:9200 --from-file=icp4i-od-
cacert.pem=icp4i-od-cacert.pem -n tekton-pipelines

```

```

azureuser@Bastion0p bin]$ echo ----BEGIN CERTIFICATE---- > icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo MIIDUTCCAjmgAwIBAgIIR9Ik2WbhHowDQYJKoZIhvcaNQELBQAwNjE0MDIGA1UE >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo Awrb3BLnNoaWLXlcnzp2UtczVydmUzY1zaWduZKJAMTY1MjEzMzY5NjAe >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo Fw0yMjA1MDkyMjAxMzv0yNDA3MDcyMjAxhzzMDYxNDAYbgNVBAMMK29wZW5z >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo a6lmdC1zZXJ2aNLXlcnZpbmc2lnbmVyeDE2NTIxMz20TywgEiMA0GCSqG >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo S1b3DQEBAQUAA4IBDwAwggEKAoIBAQAvyy/5ky263+jucU095THznKdvEokhwoMv >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo pXhHwNttGzfcPzG0+1Qw3bdmPjDrX9w/BJPvDUekbzP+zVXzc+AcXql >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo pBWWxHJUTiMyhX0VAXygp0+sbFDxJLkuFF5iAn+53grpSR5GpxIlLuSBROIdM75l >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo +9/LcuMt2G1bf5s6mtNj0/K9U0kChoxPts7Ef0boVLbznekeVEAVcxhua/taJ >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo 6h9XZ18tMFhvd+851b7L00Tvxxg1+jgw3lignjhoho21zdNIwkW+ISvUP5t6 >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo P/g5fb2sDb/b3GI/fiLdMUN51rf0mgWXZ6750WP7Km67wzL8KcrAgMBAAGjYzBh >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo ne4GA1udwEB/wQEAWICpDApBgvNHRMBAfEBTAQH/MB0GA1udgQwBRUoxul >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo neIqJ5fENfhEdTFh5mSUTAfBgNVHSMEGDAWgBRUoxUlneIqJ5fENfhEdTFh5mS >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo UTANBgkqhkiG9w0BAQsFAAOCAQEAOHgCUtw3vmcW0jKObBSdc8aVMrgsVja17Rx7 >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo 2B10mYjZukXSYbfGDPJ6iBKvstCsfJLwg/OFzG4xZPh2lvM/8uL/jCBda8D4kVZc >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo +3fz4fL+ghJ9QjKCXNevPqH2rV/IRJWAhdK9drqShfut3Ryjt/Nu0kkfjvwVv4V2 >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo M9GbUAb6UZFgu9CRYueO+LvJqoRdheEV7iQdp8K96c7zHHcGy/ZszXNEfDiz0Opz >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo J3gJOq4BcqT9VhSb67iBK7T2WBZYEKswD0Um7UKcTs5k9D0Chw== >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo 6f7rF7q0bGMiDAhQc7iqdG3UKedA/iOvXTxdU0x0OIshnsvMO4YThZDjvMjIBdf >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo neIqJ5fENfhEdTFh5mSUTAfBgNVHSMEGDAWgBRUoxUlneIqJ5fENfhEdTFh5mS >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ echo -----END CERTIFICATE----- >> icp4i-od-cacert.pem
azureuser@Bastion0p bin]$ kubectl delete secret icp4i-od-store-cred -n tekton-pipelines
Error from server (NotFound): secrets "icp4i-od-store-cred" not found
[azureuser@Bastion0p bin]$ kubectl create secret generic icp4i-od-store-cred --from-literal=password=e8Dz+Dj#isUdras0 --from-literal=username=OD_tekton-
pip_TCSz30 --from-literal=tracingUrl=https://cp4i-od-dev-ibm-integration-operations-dashboard-str-api-svc.openshift-operators.svc:9200 --from-file=icp4i-
-od-cacert.pem=icp4i-od-cacert.pem -n tekton-pipelines
secret/icp4i-od-store-cred created

```

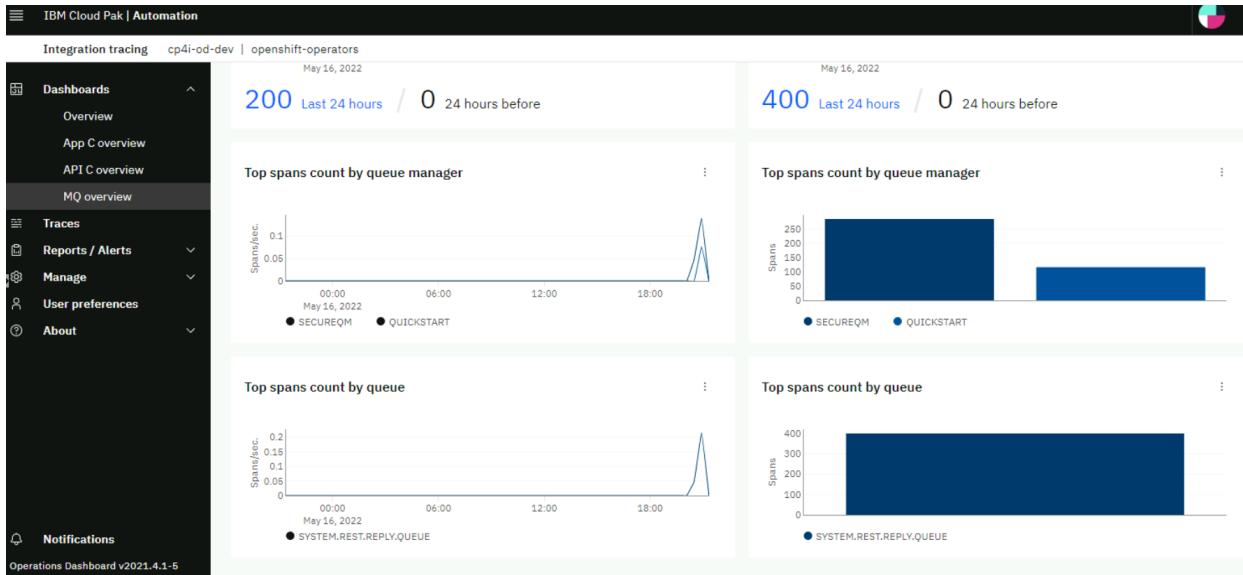
Una vez ejecutado se observará la creación del pod sin inconvenientes

The screenshot shows the Red Hat OpenShift Container Platform UI. The left sidebar is titled 'Administrator' and includes 'Home', 'Operators', 'Workloads' (selected), 'Pods' (selected), 'Deployments', 'DeploymentConfigs', 'StatefulSets', 'Secrets', 'ConfigMaps', 'CronJobs', 'Jobs', and 'DaemonSets'. The main content area shows a pod named 'secureqm-ibm-mq-0' in the 'tekton-pipelines' project. The pod status is 'Running'. The 'Details' tab is selected, displaying information such as Name: 'secureqm-ibm-mq-0', Namespace: 'tekton-pipelines', Status: 'Running', Restart policy: 'Always Restart', Active deadline seconds: 'Not configured', Pod IP: '10.129.3.128', and Node: 'cp4i-cluster-g4xx-worker-eastus21-t87lq'. Other tabs include 'Metrics', 'YAML', 'Environment', 'Logs', 'Events', and 'Terminal'.

Una vez desplegado el QueueManager podremos tener acceso a él desde el Platform UI de CP4I

The screenshot shows the IBM Cloud Pak for Automation Platform UI. The left sidebar includes 'Quick navigation', 'Integration tracing' (selected), 'Messaging' (highlighted with a red box), 'Support', and 'Documentation'. The main content area has three sections: 'Integration instance status' (status: 'Looking good!', message: 'All instances are currently functioning as normal.'), 'Integration tracing' (instance: 'cp4i-od-dev'), and 'Messaging' (queue manager instances: 'demo-qm', 'quickstart', and 'quickstart-cp4i').

Finalmente, desde el Operator Dashboard podremos verificar algunas métricas y envío de informes



13. Web

Este módulo es el que nos da la posibilidad de entrar a la administracion por medio de la web. Por defecto viene activado (Recomendado)

Installed Operators > ibm-mq.v1.8.1 > QueueManager details

QM secureqm Running

Resources				
Details	YAML	Resources	Events	
<input type="button" value="Filter"/> <input type="button" value="Name"/> <input type="text" value="Search by name..."/> <input type="button" value=""/>				
Name	Kind	Status	Created	
secureqm-ibm-mq	Service	Created	May 17, 2022, 7:24 PM	
secureqm-ibm-mq	ServiceAccount	Created	May 17, 2022, 7:24 PM	
secureqm-ibm-mq	StatefulSet	Created	May 17, 2022, 7:26 PM	
secureqm-ibm-mq-metrics	Service	Created	May 17, 2022, 7:24 PM	
secureqm-ibm-mq-qm	Route	Created	May 17, 2022, 7:24 PM	
secureqm-ibm-mq-web	Route	Created	May 17, 2022, 7:26 PM	

Si por algún motivo deseáramos quitar este módulo, al momento de desplegar nuestro Queue Manager no realizara la instalación del route web ni tampoco el servicio por el puerto 9443

Installed Operators > ibm-mq.v1.8.1 > QueueManager details

QM quickstart-cp4i  Running

Details YAML Resources Events

Filter		Name	Search by name...	/
Name	↑	Kind	↓	Status
 quickstart-cp4i-ibm-mq		Service		 Created 
 quickstart-cp4i-ibm-mq		ServiceAccount		 Created 
 quickstart-cp4i-ibm-mq		StatefulSet		 Created 
 quickstart-cp4i-ibm-mq-metrics		Service		 Created 
 quickstart-cp4i-ibm-mq-qm		Route		 Created 

Services > Service details

 quickstart-cp4i-ibm-mq

Managed by  quickstart-cp4i

Actions ▾

Details YAML Pods

Service details

Name
quickstart-cp4i-ibm-mq

Namespace
 tekton-pipelines

Labels

app.kubernetes.io/component=integration app.kubernetes.io/instance=quickstart-cp4i
app.kubernetes.io/managed-by=operator app.kubernetes.io/name=ibm-mq
app.kubernetes.io/version=9.2.50

Service routing

Service address

Type	Location
Cluster IP	172.30.90.11
Accessible within the cluster only	

Service port mapping

Name	Port	Protocol	Pod port or name
qmgr	 1414	TCP	 1414

Pod selector

Prueba MQ

<https://www.ibm.com/docs/en/ibm-mq/9.2?topic=manager-example-configuring-tls>

1. Ingresar a fix central para descargar el cliente MQ

https://www.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~WebSphere&product=ibm/WebSphere/WebSphere+MQ&release=9.2.0.0&platform=All&function=fixid&fixids=*IBM-MQC-Redist-*

	   Description	 Release date
<input type="checkbox"/>	<p>1 release level: → 9.2.5.0-IBM-MQC-Redist-Win64 Continuous Delivery: 9.2.5 IBM MQ C and .NET redistributable client for Windows x64</p> <p> Click here for product readme  Click here for installation instructions</p> <p> Click here for information on the Continuous Delivery Support Model</p> <hr/> <p>+ Show superseded fixes</p>	2022/02/24
<input type="checkbox"/>	<p>2 release level: → 9.2.5.0-IBM-MQC-Redist-LinuxX64 Continuous Delivery: 9.2.5 IBM MQ C redistributable client for Linux x86-64</p> <p> Click here for product readme  Click here for installation instructions</p> <p> Click here for information on the Continuous Delivery Support Model</p> <hr/> <p>+ Show superseded fixes</p>	2022/02/24

2. Para nuestro caso puntual utilizaremos el cliente para Linux

Download files using your web browser

Click the download link next to each file to download it.

Order number:	430997830
Total size:	74.13 MB

[Show normalized list](#) | [Hide normalized list](#)

release level: [9.2.5.0-IBM-MQC-Redist-LinuxX64](#)

-  [Click here for product readme](#)
-  [Click here for installation instructions](#)
-  [Click here for information on the Continuous Delivery Support Model](#)

Continuous Delivery: 9.2.5 IBM MQ C redistributable client for Linux x86-64

The following files implement this fix.

-  [9.2.5.0-IBM-MQC-Redist-LinuxX64.tar.gz](#) (74.13 MB)
-  [9.2.5.0-IBM-MQC-Redist-LinuxX64.tar.gz.sha256sum](#) (65 bytes)
-  [9.2.5.0-IBM-MQC-Redist-LinuxX64.tar.gz.sig](#) (512 bytes)

[Back](#)

3. También podemos descargar directamente desde la línea de comandos con wget

```
wget https://ak-delivery04-
mul.dhe.ibm.com/sdfdl/v2/sar/CM/WS/0abaq/0/Xa.2/Xb.jusyLTSp44S0eQvgboPXUhLGv6TVACNTiPa
H0JTA8GZyYmqLx-ad1AV62FI/Xc.CM/W
S/0abaq/0/9.2.5.0-IBM-MQC-Redist-
LinuxX64.tar.gz/Xd./Xf.LPR.D1VK/Xg.11742706/Xi.habanero/XY.habanero/XZ.wqLco-
Fk88XwMu0u0kXoqem52nhH6oiL/9.2.5.0-IBM-MQC
-Redist-LinuxX64.tar.gz
```

4. Descomprimir en ruta deseada que para nuestro ejemplo llamaremos /data

```
tar xvf 9.2.5.0-IBM-MQC-Redist-LinuxX64.tar.gz
```

```
4096 Feb  7 19:07 swidtag
4096 Feb  7 19:07 samp
149  Feb  7 19:07 README.Redist
4096 Feb  7 19:07 msg
174037 Feb  7 19:07 MANIFEST.Redist
    15 Feb  7 19:07 MANIFEST -> MANIFEST.Redist
4096 Feb  7 19:07 licenses
4096 Feb  7 19:07 lib64
4096 Feb  7 19:07 lib
4096 Feb  7 19:07 lap
4096 Feb  7 19:07 java
4096 Feb  7 19:07 inc
4096 Feb  7 19:07 gskit8
77729647 Feb 16 18:57 9.2.5.0-IBM-MQC-Redist-LinuxX64.tar.gz
4096 Apr 26 16:59 bin
```

5. Para poder ejecutar los comandos de los pasos siguientes es recomendable cargar en nuestro PATH los siguientes binarios /data/bin y /data/samp/bin. En el home del usuario encontraras el archivo .bashrc
6. Editar el archivo

```
vi ~/.bashrc
```

7. Adicionar la siguiente linea antes del export de la variable path

```
PATH="$PATH:/data/bin:/data/samp/bin"
source ~/.bashrc
#
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific environment
PATH="$HOME/.local/bin:$HOME/bin:$PATH"
PATH="$PATH:/data/bin:/data/samp/bin"
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

8. Generar una llave privada autofirmada con su respectiva llave publica (se requiere tener instalado openssl o cualquier otro software que nos permita generar certificados)

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

9. La Key de la base de datos es usada como trustore para el cliente de aplicación. Crear el client key database

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

10. Adicionar la llave publica previamente creada a la llave de cliente de base de datos.

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

```
1704 Apr 26 16:04 tls.key  
1115 Apr 26 16:04 tls.crt  
 88 Apr 26 16:04 clientkey.rdb  
 88 Apr 26 16:04 clientkey.crl  
193 Apr 26 16:04 clientkey.sth  
5088 Apr 26 16:04 clientkey.kdb  
 487 Apr 26 16:59 CCDT.json
```

Configuración certificados TLS para Queue Manager deployment

11. Crear TLS secret con la key previamente generada

```
oc create secret tls example-tls-secret --key="tls.key" --cert="tls.crt" -n openshift-operators
```

Creación config map con comandos MQSC

([ver comandos](#))^[obj]

12. Crearemos el configmap con el detalle de la nueva cola, canal

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: example-tls-configmap  
data:  
  tls.mqsc: |  
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE  
    DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)  
    SSLCIPH('ANY_TLS12_OR_HIGHER')  
    SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Creación route Queue Manager

13. Creamos el route correspondiente al service de nuestro queue Manager (No es indispensable, únicamente se hará en dado caso quiera cambiar el host con el que se invoca al servicio)

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-tls-route
spec:
  host: secureqmchl.chl.mq.ibm.com
  to:
    kind: Service
    name: secureqm-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Creación Queue Manager desde yaml

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: secureqm
spec:
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: SECUREQM
  mqsc:
    - configMap:
        name: example-tls-configmap
      items:
        - tls.mqsc
  availability:
    type: NativeHA
  updateStrategy: RollingUpdate
  storage:
    defaultClass: ocs-storagecluster-cephfs
  persistedData:
    enabled: false
  queueManager:
    class: ocs-storagecluster-cephfs
    deleteClaim: true
    size: 2Gi
    type: persistent-claim
  recoveryLogs:
    enabled: false
  template:
    pod:
```

```

containers:
- env:
  - name: MQSNOAUT
    value: 'yes'
  name: qmgr
version: 9.2.5.0-r1
web:
  enabled: true
pki:
  keys:
  - name: example
  secret:
    secretName: example-tls-secret
  items:
  - tls.key
  - tls.crt

```

14. Para verificar que el queue manager fue desplegado verifíquelo con el siguiente comando

```
oc get qmgr secureqm
[azureuser@Bastion0p bin]$ oc get qmgr secureqm
NAME      PHASE
secureqm  Running
```

Project: openshift-operators ▾

[Installed Operators](#) > [Operator details](#)



IBM MQ

1.8.0 provided by IBM

[Details](#) [YAML](#) [Subscription](#) [Events](#) [Queue Manager](#)

QueueManagers

Name ▾

Search by name...



Name ↑

Kind ⇕

Status ⇕

quickstart-cp4i

QueueManager

Phase ✅ Running

secureqm

QueueManager

Phase ✅ Running

15. Validar url del route creado

```
oc get routes secureqm-ibm-mq-qm
```

```
[azureuser@BastionOp bin]$ oc get routes secureqm-ibm-mq-qm
NAME          HOST/PORT
secureqm-ibm-mq-qm  secureqm-ibm-mq-qm-openshift-operators.apps.cp4i-cluster.lab.azure      PATH  SERVICES      PORT  TERMINATION  WILDCARD
                                                               secureqm-ibm-mq-qm
                                                               1414  passthrough  None
```

Creación archivo CCDT.json

16. Este archivo contiene todos los datos del queue manager, la estructura es la siguiente

```
{
  "channel": [
    {
      "name": "SECUREQMCHL",
      "clientConnection": {
        "connection": [
          {
            "host": "secureqm-ibm-mq-qm-tekton-pipelines.apps.cp4i-cluster.lab.azure",
            "port": 443
          }
        ],
        "queueManager": "SECUREQM"
      },
      "transmissionSecurity": {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Cargue variables de entorno

17. Estas variables serán leídas por los binarios amqsputc(poner mensaje en la cola) y amqsgetc(obtener mensaje). MQCCDTURL tendrá la información de nuestra cadena de conexión y MQSSLKEYR tendrá la llave que verificará nuestro QM

```
export MQCCDTURL='<full path to file>/CCDT.JSON'
export MQSSLKEYR='<full path to file>/clientkey'

export MQCCDTURL='/data/mq/bin/CCDT.json'
export MQSSLKEYR='/data/mq/bin/clientkey'
```

18. Para poner mensaje haremos uso del comando amqsputc

```
amqsputc EXAMPLE.QUEUE SECUREQM
```

Para la prueba dejaremos dos mensajes “Hola mundo” y “mensaje2 probando”

```
[azureuser@Bastion0p bin]$ ./amqsputc EXAMPLE.QUEUE SECUREQM
Sample AMQSPUT0 start
target queue is EXAMPLE.QUEUE
Hola mundo
mensaje2 probando

Sample AMQSPUT0 end
[azureuser@Bastion0p bin]$
```

19. Para ingresar al Modulo web nos dirigimos a la consola de Openshift > networking > Routes y hacemos click en el location del queue manager previamente creado

Project: openshift-operators

Routes

Name	Status	Location
example-tls-route	Accepted	https://secureqmchl.chl.mq.ibm.com
quickstart-cp4i-ibm-mq-qm	Accepted	https://quickstart-cp4i-ibm-mq-qm-openshift-operators.apps.cp4i-cluster.lab.azure
quickstart-cp4i-ibm-mq-web	Accepted	https://quickstart-cp4i-ibm-mq-web-openshift-operators.apps.cp4i-cluster.lab.azure
secureqm-ibm-mq-qm	Accepted	https://secureqm-ibm-mq-qm-openshift-operators.apps.cp4i-cluster.lab.azure
secureqm-ibm-mq-web	Accepted	https://secureqm-ibm-mq-web-openshift-operators.apps.cp4i-cluster.lab.azure

20. Modulo web

Welcome to IBM MQ

Let's get you going!

IBM MQ

- Home
- Manage
- Learning resources
- Settings

Manage SECUREQM
Your local queues (47)

Create a queue

Application quick start

Download connection file

The screenshot shows the IBM MQ Queue manager interface. On the left is a dark sidebar with navigation links: Home, Manage, Learning resources, and Settings. The main area has a title 'Queue manager: SECUREQM' and tabs for Queues, Topics, Subscriptions, and Communication. Below the tabs are three summary boxes: 'Queue depth full' (0), 'Queue depth warning' (0), and 'Queue depth normal' (47). A note says 'Looking for system queues on the table? Click here to see how'. Below this is a table of queues:

Name	Type	Depth %	Maximum depth
AMQ.62681FAB25AB1901	Local	0%	13/5000
EXAMPLE.QUEUE	Local	0%	4/5000

21. En manage > Nombre de cola creada

The screenshot shows the IBM MQ Queue manager interface. The sidebar has the 'Manage' link selected. The main area shows 'Local Queue: EXAMPLE.QUEUE' with 'messages'. It displays '4 messages (0.08%)' and 'Maximum queue depth: 5000'. Below is a table of messages:

Timestamp	Application ID	User ID	Application data
2022-4-26 21:31:45	amqsputc	azureuser	mensaje2 probando
2022-4-26 21:31:37	amqsputc	azureuser	Hola mundo

22. También podemos verificar los mensajes en la cola con el comando

```
amqsgetc EXAMPLE.QUEUE SECUREQM
[azureuser@Bastion0p bin]$ ./amqsgetc EXAMPLE.QUEUE SECUREQM
Sample AMQSGET0 start
message <mensaje 1>
message <mensaje 2>
message <mensaje 3>
```

Prueba Storage

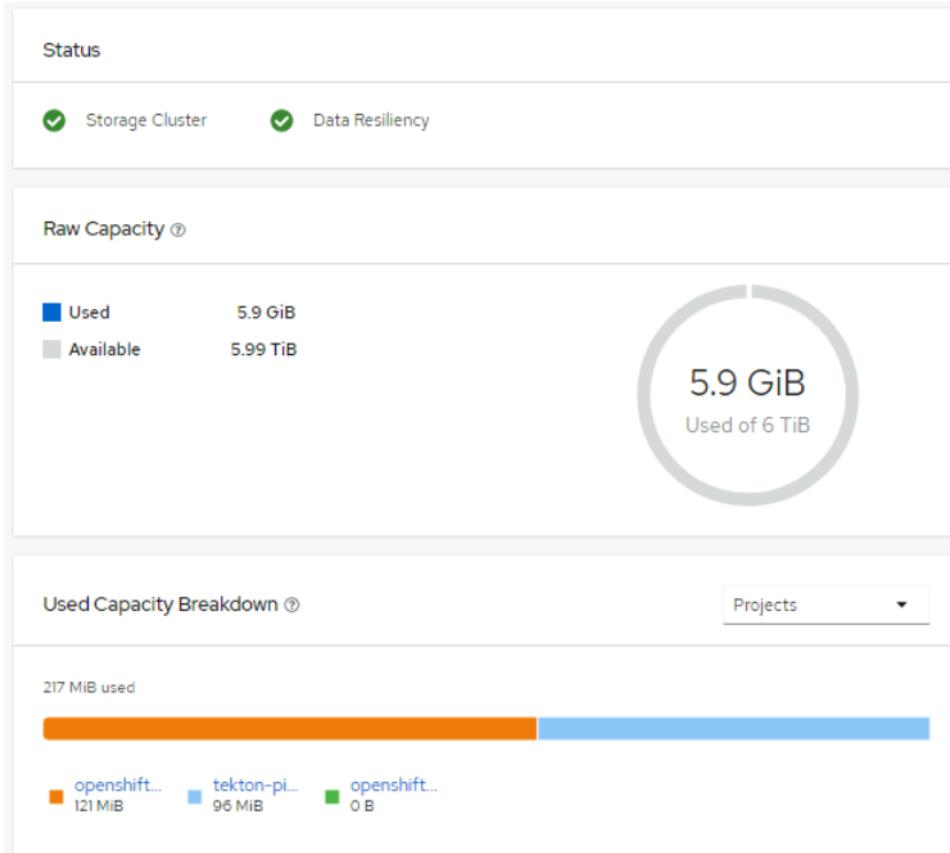


Imagen Capacidad Inicial

Para la ejecucion de la prueba se crea el siguiente script, el cual inyectara 10000 mensajes en la cola EXAMPLE.QUEUE, el tamaño del mensaje es de aprox 1024 bytes

```
#!/bin/bash
for (( c=0; c<=10000; c++ ))
do
echo "$c Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer ac commodo ipsum. Morbi condimentum at lorem vel pharetra. Duis ac fe
ugiat ante. Ut nec porttitor nisi, sit amet fringilla arcu. Vivamus quis mollis leo, vitae efficitur est. Duis dui purus, eleifend et libero nec, effici
tur malesuada nisl. Morbi posuere aliquam mollis. Duis ac mattis metus. Nulla congue cursus
malesuada. Donec vitae euismod diam, eget feugiat ante. Viva
mus venenatis enim eu tempus scelerisque. Phasellus vitae nunc luctus, rhoncus sapien eu, faucibus
arcu. Vivamus leo nisl, sodales at nisi sed, faucibus
sagittis nisi. Praesent vel enim ut tellus posuere placerat. Mauris tempus nunc eget nibh blandit
aliquam. Integer semper ante id leo eleifend, quis rh
oncus sem cursus. Vestibulum porttitor nisi non libero commodo malesuada. Aliquam luctus eu massa
ac consequat. Aliquam ullamcorper odio a erat ultrices
```

```
accumsan. Nullam ut elementum purus, ut vulputate diam. Aenean imperdiet lacus in ex tristique  
dictum. Nunc id odio a felis" | ./amqsputc EXAMPLE.QUEUE  
SECUREQM  
done
```

Manage / SECUREQM /

Local Queue: EXAMPLE.QUEUE

messages

10000 messages (16.67%)

Maximum queue depth: 60000

Una vez concluido el cargue de los mensajes podemos observar que ocupo 0.06 Gb o 6000Mb. Si examinamos a nivel del pod y sus filesystem presentados podemos observar que únicamente la data de los mensajes ocupan 15Mb

```
sh-4.4$ pwd  
/mnt/mqm-data/qmgrs/SECUREQM/queues  
sh-4.4$ du -hs *  
2.0K  !!GHOST!4269DF61!0!8FCFC885!2  
15M   EXAMPLE!QUEUE  
2.0K   EXAMPLE!QUEUE1  
2.0K   SYSTEM!ADMIN!ACCOUNTING!QUEUE
```

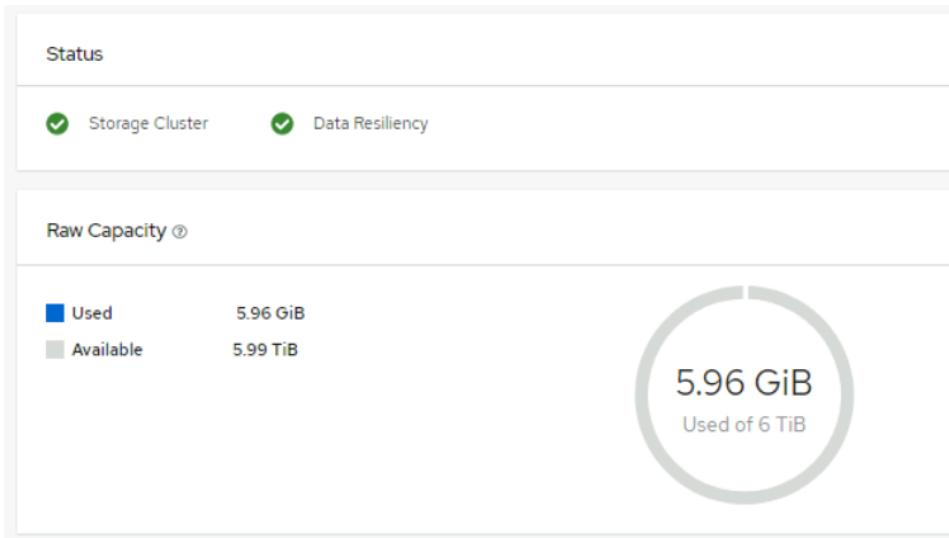


Imagen capacidad Final

El resto del espacio ocupado corresponde a Logs, archivos de Producto, metadata, entre otros.

Adición de colas o modificación en configuraciones existentes (Manual)

1. Para este ejemplo crearemos una cola Remota y modificaremos el atributo de profundidad de la cola previamente creada. Vamos a la consola de Openshift > Workloads > configmap
2. Seleccionamos el configmap previamente creado ([ver configmap](#))

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar is titled 'Administrator' and includes 'Home', 'Operators', 'Workloads' (with sub-options 'Pods', 'Deployments', 'DeploymentConfigs', 'StatefulSets', 'Secrets'), and 'ConfigMaps'. The main content area is titled 'ConfigMaps' and shows a list of configmaps. One configmap, 'example-tls-configmap', is highlighted with a red box. Other configmaps listed include 'config-logging', 'ibm-bedrock-version', and 'ibmcloud-cluster-info'.

3. Editamos el Yaml y damos en guardar

Previo

ConfigMaps > ConfigMap details

CM example-tls-configmap

Details YAML

```
1 kind: ConfigMap
2 apiVersion: v1
3 metadata:
4   name: example-tls-configmap
5   namespace: openshift-operators
6   uid: 51b7b96c-f637-437c-afa8-ef7510d432fa
7   resourceVersion: '5519506'
8   creationTimestamp: '2022-04-26T16:28:12Z'
9   managedFields: ...
19 data:
20   tls.mqsc: >
21     DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
22
23     DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
24     SSLCIPH('ANY_TLS12_OR_HIGHER')
25
26     SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
27
```

Posterior

Data

tls.mqsc

```
DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE MAXDEPTH(1000)
DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL) SSLCIPH('ANY_TLS12_OR_HIGHER')
SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
DEFINE QRREMOTE('EXAMPLE.REMOTE') REPLACE
```

4. Una vez realizados los cambios nos dirigimos a workloads > pods y eliminamos el que corresponda a nuestro queue manager

Nota: Toda creación o modificación realizada en el portal será borrada, si desea hacer de la misma una configuración permanente debe incluirlo en el configmap y reiniciar los pods del queue Manager

Name	Status
ibm-mq-5d98797c7f-gjj49	Running
openshift-pipelines-operator-75587c76cf-nfnqd	Running
quickstart-cp4i-ibm-mq-0	Running
secureqm-ibm-mq-0	Running

Verificar que la subida no presente errores. De ser así verificar el yaml del configmap donde detalla los comandos MQSC

Details Metrics YAML Environment Logs Events Terminal

Log streaming... qmgr Current log ▾

108 lines

```
2022-04-29T22:27:16.713Z AMQ8939I: Automatic MQSC configuration script has completed, and contained 7 command(s), of which 0 had errors. [ArithInsert1(7), CommentInsert1(0)]
2022-04-29T22:27:16.745Z AMQ5037I: The queue manager task 'DEFERRED_DELIVERY' has started. [ArithInsert2(1), CommentInsert1(DEFERRED_DELIVERY)]
2022-04-29T22:27:16.745Z AMQ5037I: The queue manager task 'STATISTICS' has started. [ArithInsert2(1), CommentInsert1(STATISTICS)]
2022-04-29T22:27:16.745Z AMQ5051I: The queue manager task 'ASYNCQ' has started. [ArithInsert2(1), CommentInsert1(ASYNCQ)]
2022-04-29T22:27:16.745Z AMQ5051I: The queue manager task 'RESOURCE_MONITOR' has started. [ArithInsert2(1), CommentInsert1(RESOURCE_MONITOR)]
2022-04-29T22:27:16.745Z AMQ5051I: The queue manager task 'MULTICAST' has started. [ArithInsert2(1), CommentInsert1(MULTICAST)]
2022-04-29T22:27:16.746Z AMQ5052I: The queue manager task 'QPUBSUB-CTRLR' has started. [ArithInsert2(1), CommentInsert1(QPUBSUB-CTRLR)]
2022-04-29T22:27:16.746Z AMQ5052I: The queue manager task 'QPUBSUB-SUBPT-NLCACHE' has started. [ArithInsert2(1), CommentInsert1(QPUBSUB-SUBPT-NLCACHE)]
2022-04-29T22:27:16.746Z AMQ5051I: The queue manager task 'EXPIRE' has started. [ArithInsert2(1), CommentInsert1(EXPIRE)]
2022-04-29T22:27:16.746Z AMQ5052I: The queue manager task 'QPUBSUB-QUEUE-NLCACHE' has started. [ArithInsert2(1), CommentInsert1(QPUBSUB-QUEUE-NLCACHE)]
2022-04-29T22:27:16.746Z AMQ5052I: The queue manager task 'PUBSUB-DAEMON' has started. [ArithInsert2(1), CommentInsert1(PUBSUB-DAEMON)]
2022-04-29T22:27:16.746Z AMQ5975I: 'IBM MQ Distributed Pub/Sub Controller' has started. [CommentInsert1(IBM MQ Distributed Pub/Sub Controller)]
2022-04-29T22:27:16.746Z AMQ5037I: The queue manager task 'DEFERRED-MSG' has started. [ArithInsert2(1), CommentInsert1(DEFERRED-MSG)]
2022-04-29T22:27:16.746Z AMQ5051I: The queue manager task 'PRESERVED-Q' has started. [ArithInsert2(1), CommentInsert1(PRESERVED-Q)]
2022-04-29T22:27:16.746Z AMQ5051I: The queue manager task 'Q-DELETION' has started. [ArithInsert2(1), CommentInsert1(Q-DELETION)]
2022-04-29T22:27:16.748Z AMQ9722W: Plain text communication is enabled.
2022-04-29T22:27:16.747Z AMQ5037I: The queue manager task 'MARKINTSCAN' has started. [ArithInsert2(1), CommentInsert1(MARKINTSCAN)]
2022-04-29T22:27:16.747Z AMQ5051I: The queue manager task 'ACTVTRC' has started. [ArithInsert2(1), CommentInsert1(ACTVTRC)]
2022-04-29T22:27:16.748Z AMQ5975I: 'IBM MQ Distributed Pub/Sub Fan Out Task' has started. [CommentInsert1(IBM MQ Distributed Pub/Sub Fan Out Task)]
2022-04-29T22:27:16.748Z AMQ5975I: 'IBM MQ Distributed Pub/Sub Command Task' has started. [CommentInsert1(IBM MQ Distributed Pub/Sub Command Task)]
2022-04-29T22:27:16.750Z AMQ5026I: The listener 'SYSTEM.LISTENER.TCP.1' has started. ProcessId(459). [ArithInsert1(459), CommentInsert1(SYSTEM.LISTENER.TCP.1)]
2022-04-29T22:27:16.750Z AMQ5975I: 'IBM MQ Distributed Pub/Sub Publish Task' has started. [CommentInsert1(IBM MQ Distributed Pub/Sub Publish Task)]
2022-04-29T22:27:16.952Z AMQ5886I: Queued Publish/Subscribe Daemon started for queue manager SECUREQM. [CommentInsert1(SECUREQM)]
2022-04-29T22:27:40.062Z Started web server
2022-04-29T22:27:46.745Z AMQ5041I: The queue manager task 'AUTOCONFIG' has ended. [CommentInsert1(AUTOCONFIG)]
```

5. Verificamos en la consola de Queue Manager

← → ⌂ | Not secure | <https://secureqm-ibm-mq-web-openshift-operators.apps.cp4i-cluster.lab.azure/ibmmq/console/#/manage/qmgr/SECUREQM/queues>

IBM MQ

Home Manage Learning resources Settings

Queues Topics Subscriptions Communication

0	0	47
Queue depth full	Queue depth warning	Queue depth normal
0 user-defined 0 system queues	0 user-defined 0 system queues	2 user-defined 45 system queues

Looking for system queues on the table? [Click here to see how](#)

Name	Type	Depth % ↓	Maximum depth
AMQ.626C657223ECC301	Local	0%	0/5000
EXAMPLE.QUEUE	Local	0%	0/500
REMOTE2.QUEUE	Remote	-	-

Items per page: 10 ▾ 1-3 of 3 items

Despliegue Continuo – Tekton/OpenShift pipelines

Vamos a desplegar la última versión hasta la fecha de tekton para ello ejecutamos el siguiente comando

```
kubectl apply -f https://storage.googleapis.com/tekton-releases/pipeline/previous/v0.35.0/release.yaml
```

Cluster Privado Openshift

```
oc image mirror gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/webhook:v0.35.0=BastionOp.lab.azure:5000/gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/webhook:v0.35.0 -a ${REG_CREDS} --insecure
```

```
oc image mirror gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/controller:v0.35.0=BastionOp.lab.azure:5000/gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/controller:v0.35.0 -a ${REG_CREDS} --insecure
```

Tekton / Openshift Pipelines

Construccion de Imagen con cliente OC

Para la ejecucion de las task requerimos una imagen que contenga el cliente oc para poder ejecutar los (oc apply -f [route, configmap, queuemanager].yaml)

Haremos uso de nuestra maquina Bastion

Creamos un directorio llamado docker en nuestro home, seguido creamos un archivo Dockerfile

```
mkdir docker  
cd docker  
vi Dockerfile  
[azureuser@BastionOp docker]$ pwd  
/home/azureuser/docker
```

```
##Descargamos una versión concreta de UBUNTU, a través del tag  
FROM ubuntu:20.04  
MAINTAINER gbm  
##Actualizamos el sistema  
RUN apt-get update  
##Instalamos nginx  
RUN apt-get install -y nginx  
##Arrancamos NGINX a través de ENTRYPOINT para que no pueda ser modificado en la creación del contenedor  
ENTRYPOINT ["/usr/sbin/nginx", "-g", "daemon off;"]  
##Exponemos el Puerto 80  
EXPOSE 80  
##Inicialización archivo kubeconfig  
RUN mkdir /.kube  
RUN > /.kube/config
```

```

RUN chmod -R 777 ./kube/config
##instalacion Curl, telnet, wget, oc
RUN apt-get install -y curl
RUN apt-get install -y telnet
RUN apt-get install -y wget
RUN wget https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/stable-4.8/openshift-client-linux.tar.gz
RUN tar -xvf openshift-client-linux.tar.gz -C /bin oc kubectl

```

Construimos nuestra imagen basandonos en este Dockerfile

```

podman build . -t clientes
podman images

```

Nota: asegúrese que se encuentra dentro de la carpeta docker

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/Clientes	latest	b2f6197703c6	38 minutes ago	360 MB
<none>	<none>	946c3e7da08e	46 minutes ago	360 MB
<none>	<none>	d32db8f18466	About an hour ago	360 MB
docker.io/library/ubuntu	20.04	53df61775e88	6 days ago	75.1 MB
docker.io/library/registry	2	2e200967d166	4 weeks ago	24.7 MB
docker.io/jc21/registry-ui	latest	bbd97a6e83f8	23 months ago	526 MB

```

podman push localhost/Clientes BastionOp.lab.azure:5000/Clientes/Clientes -a ${REG_CREDS} --
insecure

```

Definicion task

```

apiVersion: tekton.dev/v1beta1
kind: Task
metadata:
  name: deploy-yaml
  namespace: tekton-pipelines
spec:
  params:
    - default: ""
      description: yamlfile alocated in the bastion host
      name: yamlfile
      type: string
  steps:
    - args:
        - '-c'
        - "oc apply -f $(params.yamlfile)"
      command:
        - /bin/bash
  image: 'BastionOp.lab.azure:5000/Clientes/Clientes'
  name: date

```

Definición Pipeline

```
apiVersion: tekton.dev/v1beta1
kind: Pipeline
metadata:
  name: despliegue-mq
  namespace: tekton-pipelines
spec:
  tasks:
    - name: deploy-route
      params:
        - name: yamlfile
          value: 'http://BastionOp.lab.azure:80/IBMMQ/route.yaml'
      taskRef:
        kind: Task
        name: deploy-yaml
    - name: deploy-configmap
      params:
        - name: yamlfile
          value: 'http://BastionOp.lab.azure:80/IBMMQ/configmap.yaml'
      taskRef:
        kind: Task
        name: deploy-yaml
    - name: deploy-queuemanager
      params:
        - name: yamlfile
          value: 'http://BastionOp.lab.azure:80/IBMMQ/queuemanager.yaml'
      runAfter:
        - deploy-configmap
        - deploy-route
      taskRef:
        kind: Task
        name: deploy-yaml
```

Ejecutamos nuestro pipeline con la opción Run y verificamos el estado en PipelineRuns

Pipelines

Create ▾

Pipelines PipelineRuns PipelineResources Conditions

Filter Name Search by name... []

Name	Last run	Task status	Last run status	Last run time
PL despliegue-mq	PLR despliegue-mq-wm9mz4	[green bar]	Succeeded	May 17, 2022, 7:24 PM

Start
Start last run
Add Trigger
Edit labels
Edit annotations
Edit Pipeline
Delete Pipeline

Pipelines PipelineRuns PipelineResources Conditions

Filter Name Search by name... []

Name	Last run
PL despliegue-mq	PLR despliegue-mq-wm9mz4

En esta vista podremos verificar la ejecución de cada una de las tareas y el log respectivo de cada una de ellas.

PipelineRuns > PipelineRun details

PLR despliegue-mq-wm9mz4 Succeeded

Details YAML TaskRuns Logs Events

PipelineRun details



PipelineRuns > PipelineRun details

PLR despliegue-mq-q9k1ai Succeeded

Tech preview

Actions ▾

Details YAML TaskRuns Logs Events

Download | Download all task logs | Expand

deploy-queuemanager

STEP-DATE

```
2022/05/06 18:00:13 warning: unsuccessful cred copy: ".docker" from "/tekton/creds" to "/": unable to create destination directory: mk
queuemanager.mq.ibm.com/secureqm1 created
```

Si llega a presentar algún tipo de errores en el namespace donde se realizó la creación del Queue Manager diríjase a la sección de errores ([ver aquí](#))

Errores

Despliegue de Operador en namespace diferente a Openshift-operators

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

Name *

Labels

License *

Settings that control your acceptance of the license, and which license metrics to use

Version *

Setting that controls the version of MQ that will be used (required). For example: "9.15.0-r2" would specify MQ version 9.15.0, using the second revision of the container image. Container-specific fixes are often applied in revisions, such as fixes to the base image.

Queue Manager

Settings for the Queue Manager container and underlying Queue Manager

Tracing

Settings for tracing integration with the Cloud Pak for Integration Operations Dashboard

Web

Settings for the MQ web server

Advanced configuration

Error

Fix the following errors:

- Error "failed calling webhook "openshift-operators/v1beta1validator.queueManagers.mq.ibm.com": Post "https://ibm-mq-webhook.openshift-operators.svc:443/validate/v1beta1?timeout=10s": service "ibm-mq-webhook" not found" for field "undefined".

Ratificar el tipo de storage dependiendo del availability que se elija.

The screenshot shows the 'Advanced configuration' section of the Tracing and Web settings. Both sections have an 'Error' status with a red exclamation mark. The Tracing error message states: 'Fix the following errors: admission webhook "openshift-operators.v1beta1.validator.queueManagers.mq.ibm.com" denied the request: QueueManager.mq.ibm.com "quickstart-cp4i" is invalid: spec.queueManager.storage.queueManager.type: Invalid value: "ephemeral": Native-HA queue managers cannot be used with "ephemeral" storage'. The Web error message is partially visible. At the bottom, there are 'Create' and 'Cancel' buttons.

Ejecución de pipeline con usuario default

```
from server for: "http://BastionOp.lab.azure:80/IBMMQ/configmap.yaml": configmaps "example-tls-configmap" is forbidden: User "system:serviceaccount:tekton-pipelines:default" cannot get resource "configmaps" in API group "" in the namespace "openshift-operators"
```

The screenshot shows the 'PipelineRuns' details page for a pipeline named 'despliegue-mq-9jx3tt'. The status is 'Failed'. The 'Logs' tab is selected, showing the terminal output of the pipeline tasks. One task, 'deploy-route', failed with the error: 'tekton/creds" to "/": unable to create destination directory: mkdir ./.docker: permission denied'. Another task failed with the error: 'configmaps "example-tls-configmap" is forbidden: User "system:serviceaccount:tekton-pipelines:default" cannot get resource "configmaps" in API group "" in the namespace "openshift-operators"'. There are tabs for 'Details', 'YAML', 'TaskRuns', 'Logs', and 'Events'.

Solución:

Dependiendo del namespace en el cual estamos indicando la creacion del Queue Manager debemos dar el rol necesario a la service account definida

```
oc adm policy add-role-to-user edit system:serviceaccount:tekton-pipelines:default --rolebinding-name=role-tekton-op-operators -n tekton-pipelines
```

Adicionales

Expand disk vm azure

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/expand-disks>

Esta actividad fue necesaria debido a que al momento de desplegar los catálogos de IBM y RedHat (instalación desconectada) el directorio /data de la maquina bastion llego a su 100%, dentro del size

inicial se estimó únicamente un disco de 500Gb para almacenar dichos catálogos sin embargo se verifica que el catálogo de IBM tiene un peso aproximado de 300Gb y el de RedHat 900Gb.

Dentro del proceso se procede a apagar la maquina redimensionar los discos y posteriormente subir la maquina y ejecutar los siguientes comandos.

```
[azureuser@Bastion0p ~]$ sudo parted /dev/sda
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) resize
Error: The resize command has been removed in parted 3.0
(parted) print
Model: Msft Virtual Disk (scsi)
Disk /dev/sda: 2199GB
Sector size (logical/physical): 512B/4096B
Partition Table: loop
Disk Flags:

Number  Start   End     Size    File system  Flags
 1      0.00B  2199GB  2199GB  ext4

(parted) resizepart
Partition number? 1
Warning: Partition /dev/sda is being used. Are you sure you want to continue?
Yes/No? yes
End? [2199GB]? 2199GB
(parted) quit
Information: You may need to update /etc/fstab.
```

```
[azureuser@Bastion0p ~]$ sudo resize2fs /dev/sda
resize2fs 1.45.4 (23-Sep-2019)
Filesystem at /dev/sda is mounted on /data; on-line resizing required
old_desc_blocks = 128, new_desc_blocks = 256
The filesystem on /dev/sda is now 536870912 (4k) blocks long.
```