

## How to connect IBM MQ Explorer to Remote IBM Cloud Queue Manager

[https://cloud.ibm.com/docs/mqcloud?topic=mqcloud-mqoc\\_configure\\_chl\\_ssl#keystore\\_jks\\_mac](https://cloud.ibm.com/docs/mqcloud?topic=mqcloud-mqoc_configure_chl_ssl#keystore_jks_mac)

### Prerequisites

- You have MQ Explorer installed on your remote machine.
- You have a MQ IBM Cloud Queue Manager deployed

### Download the Queue Manager Certificate.

You must download the cert from the keystore on the IBM Cloud Queue Manager. The download will be in the format of a PEM file. You need to use the cert that is associated with the queue manager.

**Note:** *The default cert that is associated with the queue manager is labeled qmgrcert*

### Create a java keystore

Below is an example of the keytool command to convert the CERT (your downloaded pem file) into a jks keystore.

```
keytool -importcert -file qmgrcert.pem -alias qmgrcert -keystore key.jks -storepass <your password>
```

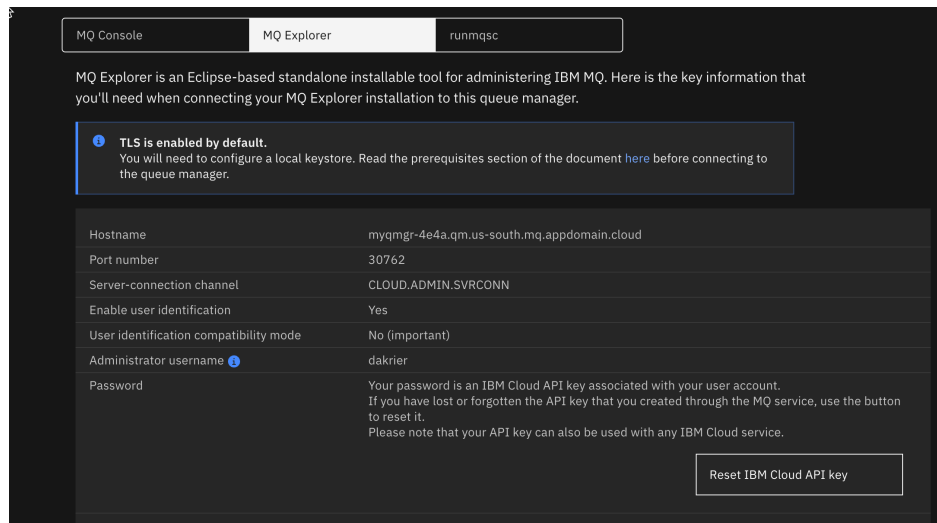
To view the keystore you can run this command below

```
Keytool -list -keystore key.jks -storepass <your password>
```

### Reset the IBM Cloud API Key if you do not have the key already.

You can find a button in the bottom right hand corner when you look at the following admin screen on IBM Cloud.

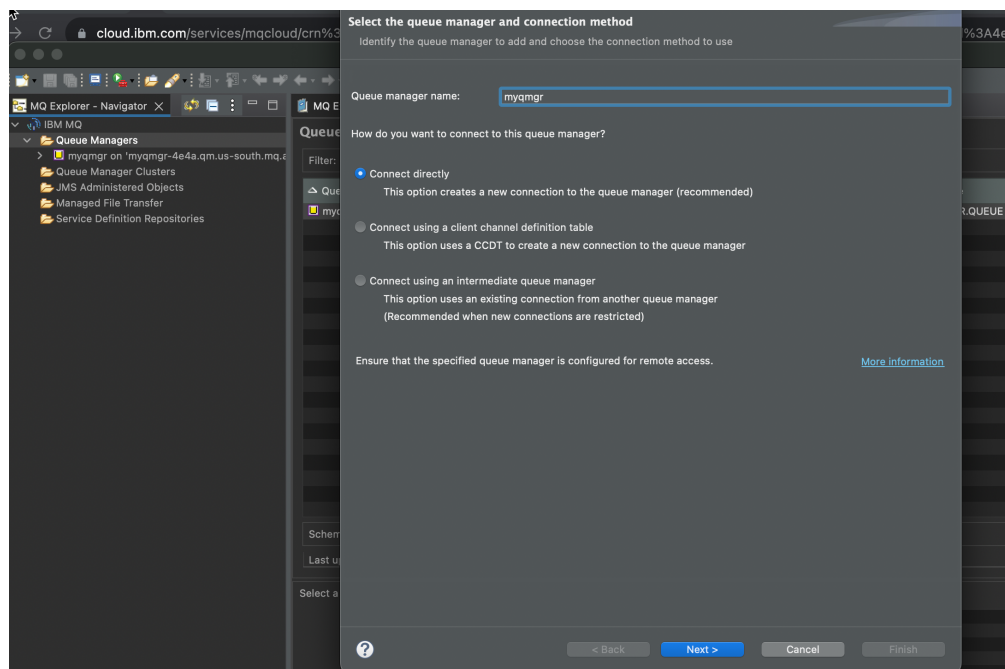
**Note:** *Download the apikey, you will only get one chance to view it.*



## Configure a remote queue manager in MQ Explorer

Open up MQ Explorer on your remote workstation. Right click and add a new remote queue manager.

You will need to supply the correct connection information as showed in the below example.



**Add Queue Manager**

**Specify new connection details**  
Provide details of the connection you want to set up

Queue manager name: myqmgr

Connection details

Host name or IP address: myqmgr-4e4a.qm.us-south.mq.appdomain.cloud

Port number: 30762

Server-connection channel: CLOUDADMIN.SVRCONN

☐ Is this a multi-instance queue manager?

Connection details to second instance

Host name or IP address:

Port number: 1414

Server-connection channel: SYSTEM.ADMIN.SVRCONN

☐ Automatically connect to this queue manager at startup or if the connection is lost

☒ Automatically refresh information shown for this queue manager

Refresh interval (seconds): 300

? < Back Next > Cancel Finish

**Specify security exit details**  
Provide the name and location of a security exit and optionally some exit data

Queue manager name: myqmgr

☐ Enable security exit

Exit name:

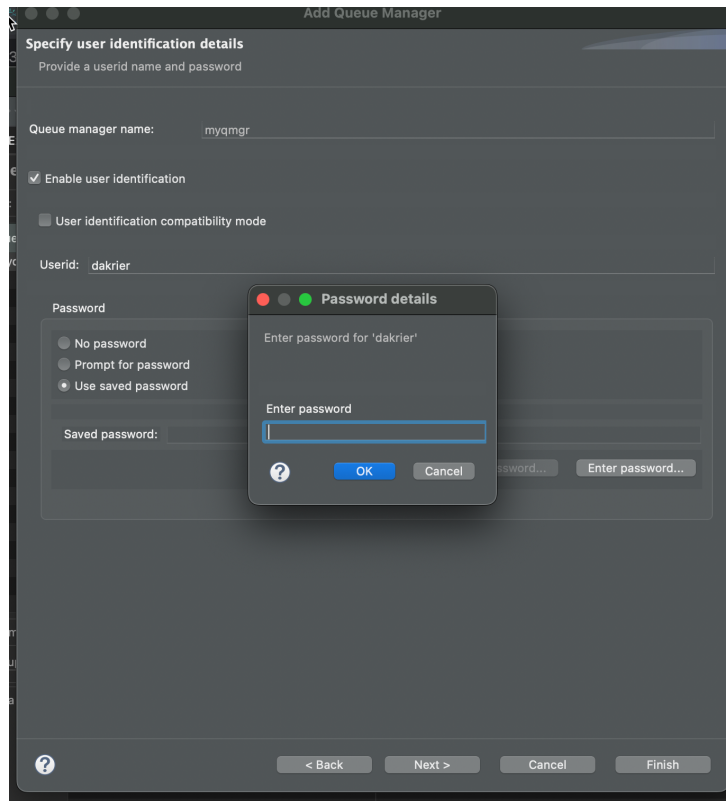
☒ in directory Browse...

☐ in jar Browse...

Exit data:

? < Back Next > Cancel Finish

The screen shot below is where you need to enter the apikey for your admin user. This should be in the file you downloaded when you rest the API Key.



You need to enable TLS security and point both the personal keystore and the trusted keystore to your repository, which is the jks file.

The keystore has a password. You actually set that password in the first step. You will need to enter the password for your keystore for both the private and trusted stores.

Add Queue Manager

Specify SSL certificate key repository details

Provide the location and password of a trusted store and optionally a personal certificate store

Queue manager name: myqmgr

☒ Enable SSL key repositories

Trusted Certificate Store

Store name: /Users/dakier/code/WebstormProjects/mqtest933/key.jks Browse...

Password: ..... Clear password... Enter password...

Personal Certificate Store

Store name: /Users/dakier/code/WebstormProjects/mqtest933/key.jks Browse...

Password: ..... Clear password... Enter password...

? < Back Next > Cancel Finish

The cipher spec should match what you have on the MQ Server.

**Note:** *The default is ANY\_TLS12\_OR\_HIGHER*

**Add Queue Manager**

**Specify SSL option details**  
Select which SSL options to use - these can only be enabled after a trusted store has been defined on the previous page

Queue manager name:

SSL FIPS required:

☒ Enable SSL options

CipherSpec


Set security for this end of the connection

SSL CipherSpec:

Any common CipherSpec using TLS 1.2 or above supported by both ends of the channel

SSL reset count:

Peer name:



You should now be able to connect.