How to setup TLS security with .NET Core Windows client.

Specifications

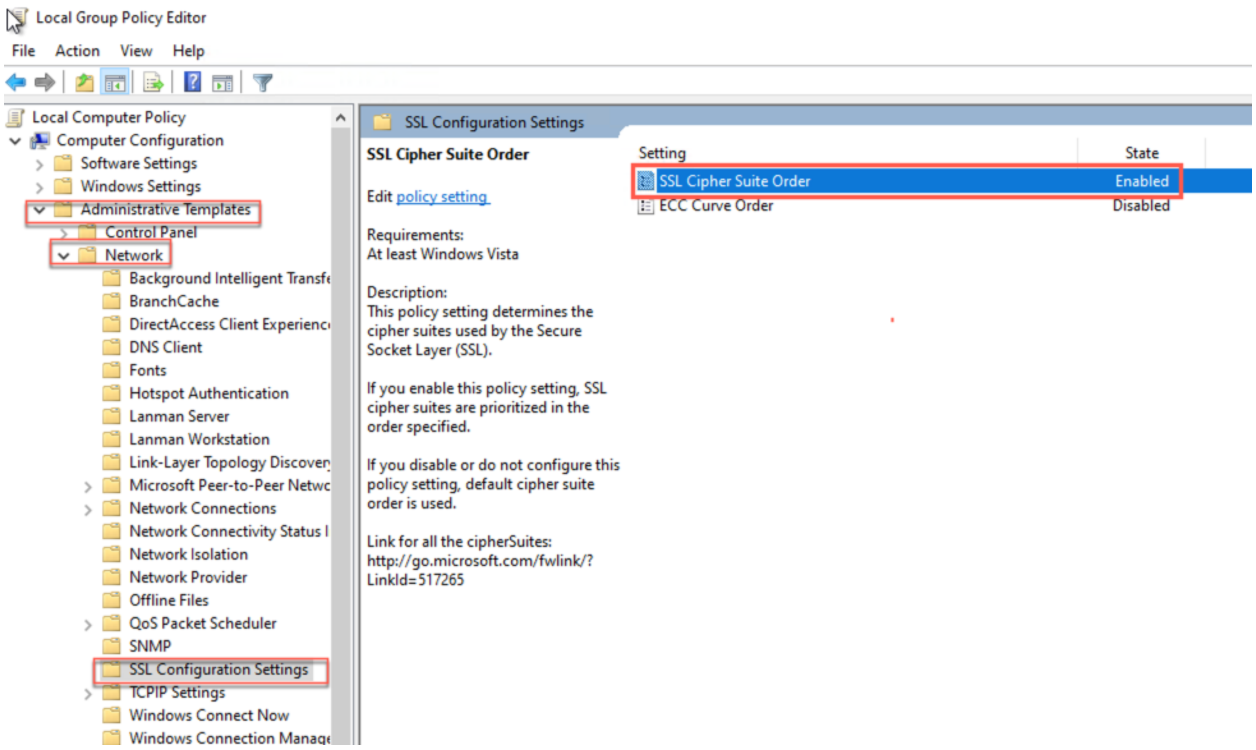
- IBM MQ Service on IBM Cloud V 9.3.3.2
- IBM MQ dotNet Client V 9.3.3.1
- Windows Client Machine (I used Windows 2022 Server for my testing)

Prerequisites

- Deploy IBM MQ Service in IBM Cloud. (The lite service will work)
- Deploy a Queue Manager within your Cloud MQ service
- Windows Client Machine
    - Install openssl
    - Install Visual Studio Developer Edition
    - Install IBM MQ Client V9.3.3.1

Enable SSL on your Windows Client.

You MUST have SSL policy enabled on your client machine. To ensure that you have this setup please run gpedit.exe and verify.



SSL Cipher Suite Order

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting determines the cipher suites used by the Secure Socket Layer (SSL).

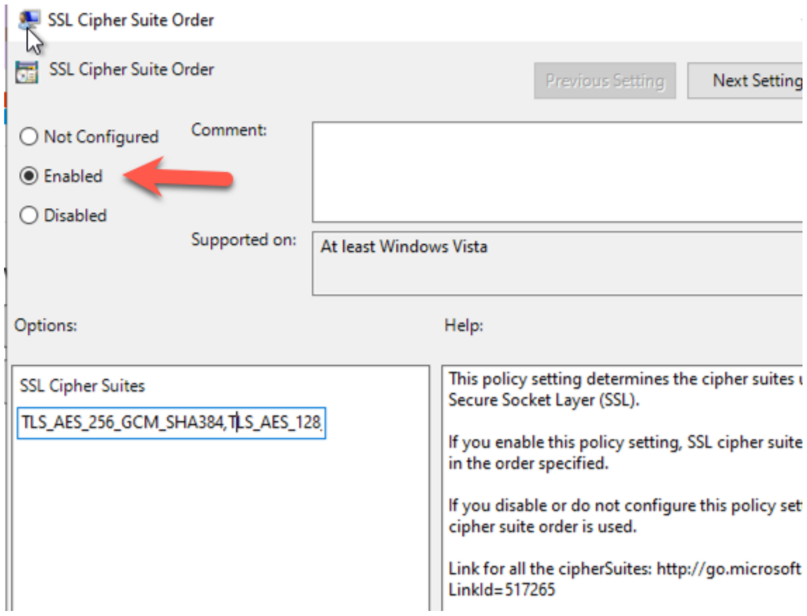
If you enable this policy setting, SSL cipher suites are prioritized in the order specified.

If you disable or do not configure this policy setting, default cipher suite order is used.

Link for all the cipherSuites:
http://go.microsoft.com/fwlink/?LinkId=517265

Setting	State
SSL Cipher Suite Order	Enabled
ECC Curve Order	Disabled

If the above setting is not enabled, you will need to enable this.



Not Configured [✓ Enabled] Disabled

Supported on: At least Windows Vista

SSL Cipher Suites

TLS_AES_256_GCM_SHA384,TLS_AES_128

This policy setting determines the cipher suites u
Secure Socket Layer (SSL).

If you enable this policy setting, SSL cipher suite
in the order specified.

If you disable or do not configure this policy set
cipher suite order is used.

Link for all the cipherSuites: http://go.microsoft
LinkId=517265

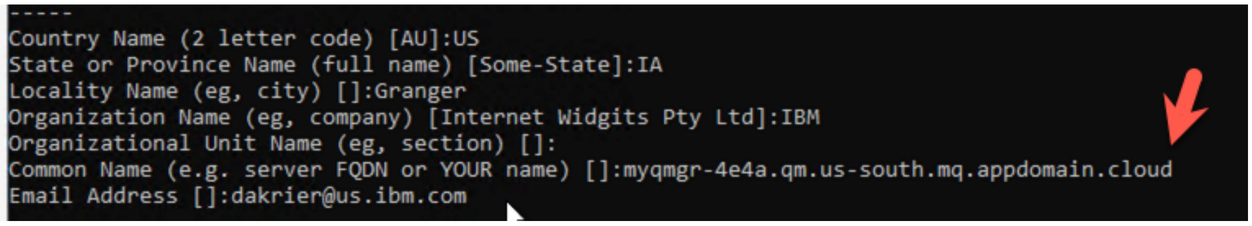
NOTE: *You can leave the default cipher Suites order field to the default setting. We just need to make sure that we are using one of the values in this filed when we make a connection with the client application.*

Create SSL Cert / Key

You will need to create a ssl cert and key using the below command. To do this you need to have openssl installed in your machine.

*** Example Command ***
openssl req -newkey rsa:2048 -nodes -keyout krierclientKey.pem -x509 -days 365 -out krierclientCert.pem

Below is an example of the settings I used. The only thing that is important to note on this is the FQDN name. You need to set this to the fully qualified domain name (host name) of your MQ Server. You can find this information in the connection information for your queue manager in the cloud.



```
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:IA
Locality Name (eg, city) []:Granger
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IBM
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:myqmgr-4e4a.qm.us-south.mq.appdomain.cloud
Email Address []:dakrier@us.ibm.com
```

Combine the Cert / Key into a single PEM file

# Combine the private key and public certificate into a single file
cat krierclientKey.pem > Krier.pem
cat krierclientCert.pem >> Krier.pem

Note: *if you are doing this on a windows machine, you can use a text editor to create a new file and paste the contents of the two PEM files into a NEW PEM file.*

Create the pfx (pk12) file for the Windows Cert Store

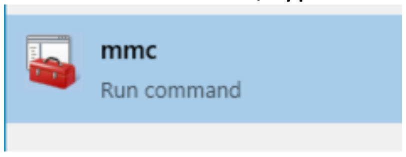
openssl pkcs12 -export -out krier.pfx -in krier.pem

Note: *We will use this new singe PEM file with both the key and the cert on the IBM Public Cloud Queue Manager.*

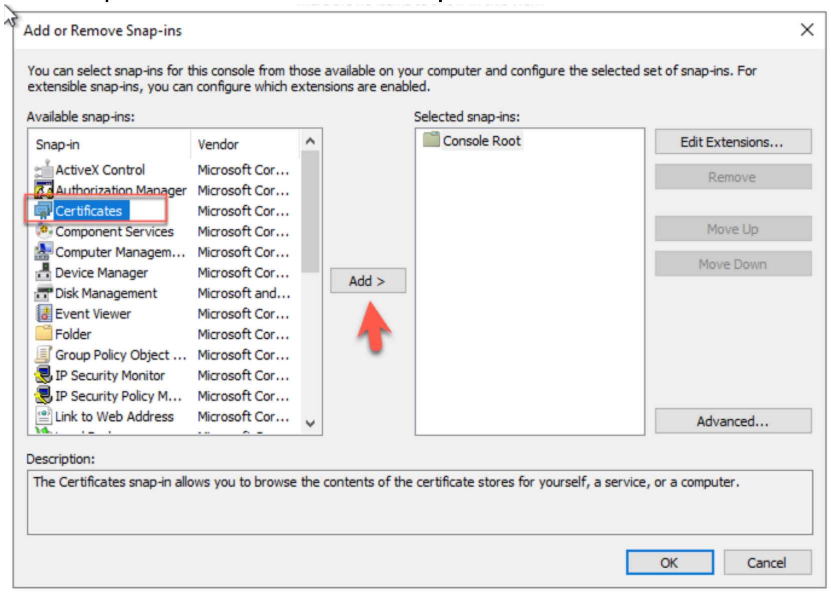
Import the pfx file into BOTH personal key store and Windows Trust Store

On your Windows client, you will need to run the MMC to modify the certificates for windows.

From the run menu, type mmc and hit enter. Open this app below.



From the file menu, select Add Remove Snap In and select the Certificates from the left hand menu option. Add that to the snap in and select OK.



You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy M...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

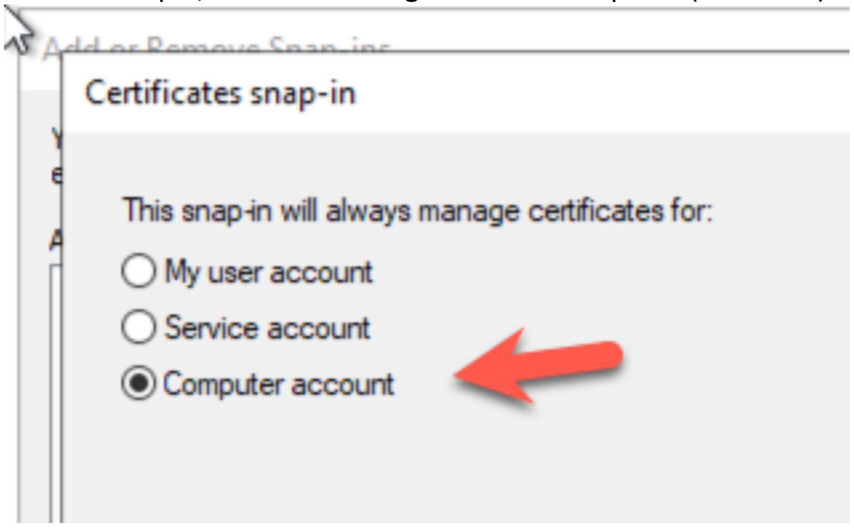
Selected snap-ins:

Console Root

Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

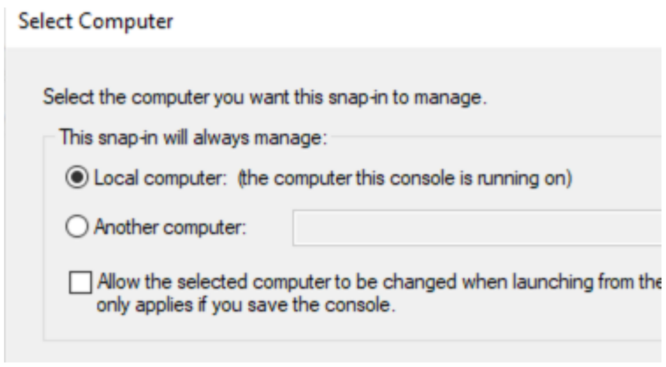
In our example, we will be using the Local Computer (*SYSTEM) keystore repository.



Certificates snap-in

This snap-in will always manage certificates for:

- [ ] My user account
- [ ] Service account
- [✓] Computer account


Select Computer

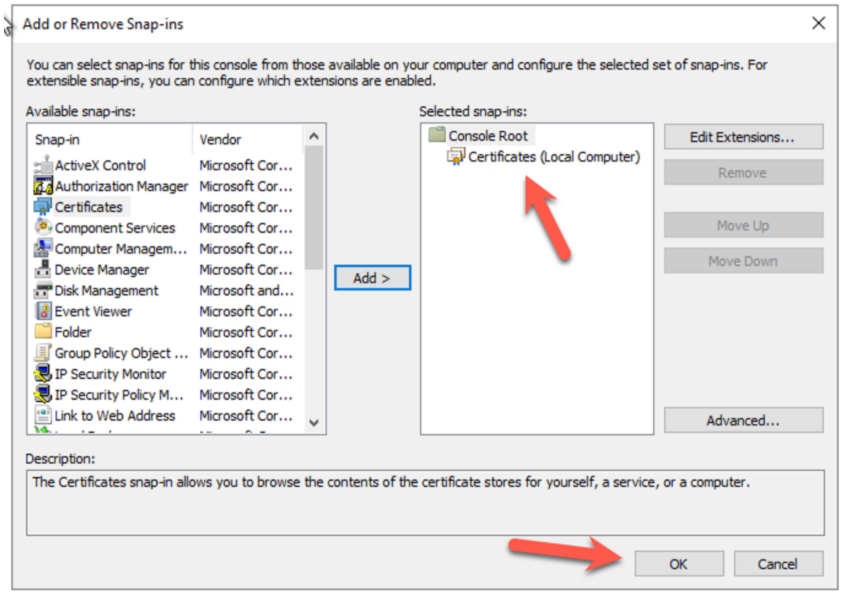
Select the computer you want this snap-in to manage.

This snap-in will always manage:

[✓ Local computer: (the computer this console is running on)]

Another computer:

Allow the selected computer to be changed when launching from the
only applies if you save the console.


Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy M...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

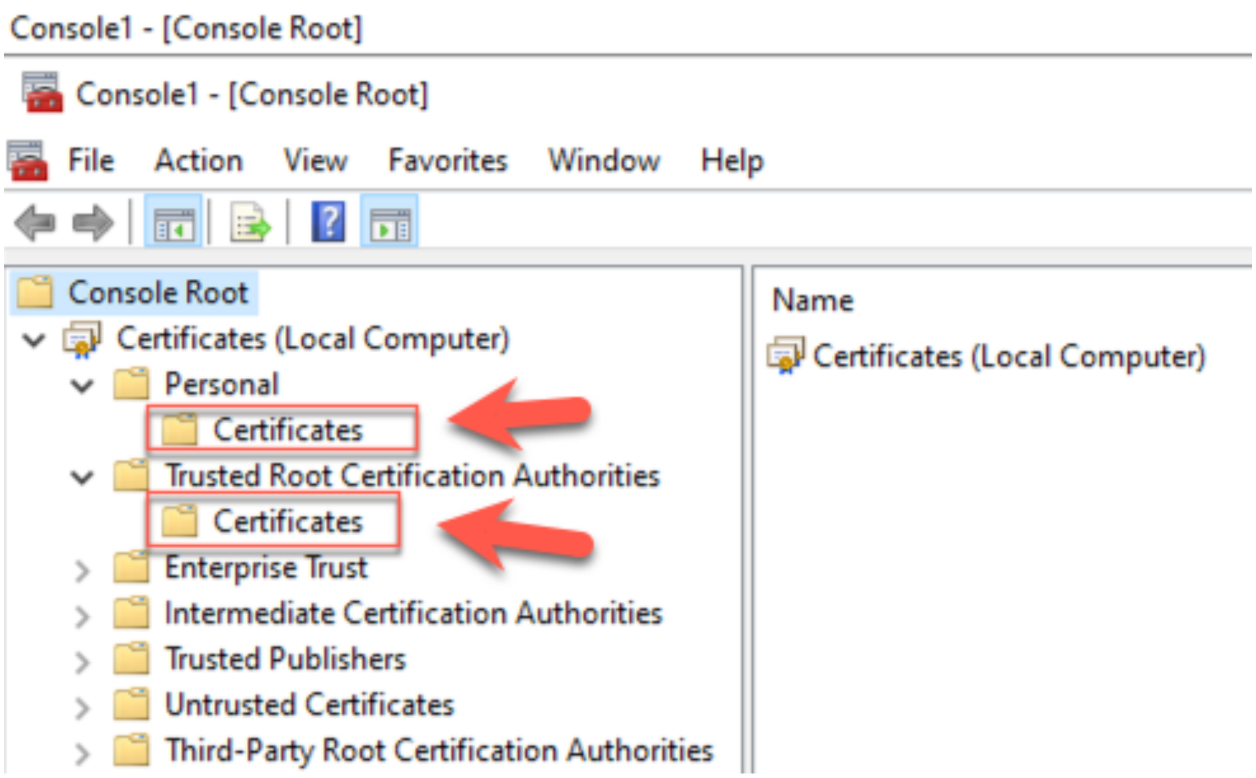
Selected snap-ins:

Console Root
Certificates (Local Computer)

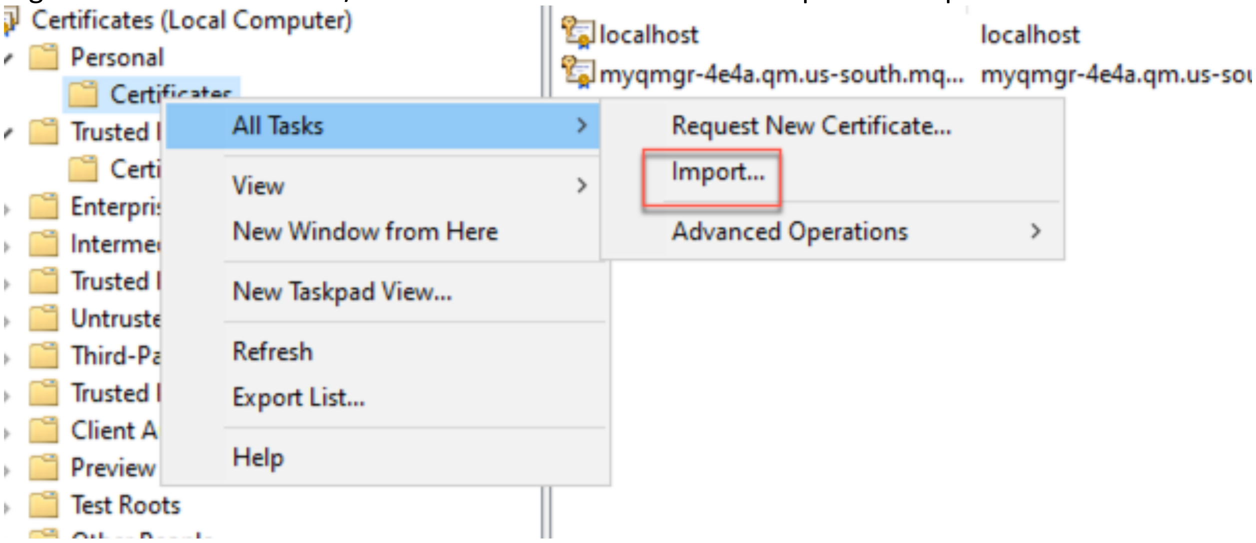
Description:

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

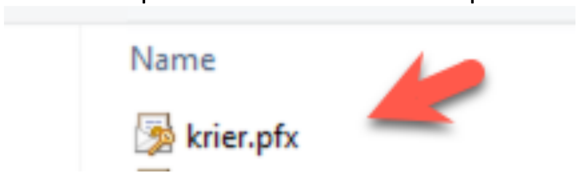
We need to add our pfx file to BOTH the personal repository and the trusted repository.


Console1 - [Console Root]
Console1 - [Console Root]
File
Action
View
Favorites
Window
Help
Console Root
Certificates (Local Computer)
Personal
Certificates
Trusted Root Certification Authorities
Certificates
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authorities
Name
Certificates (Local Computer)

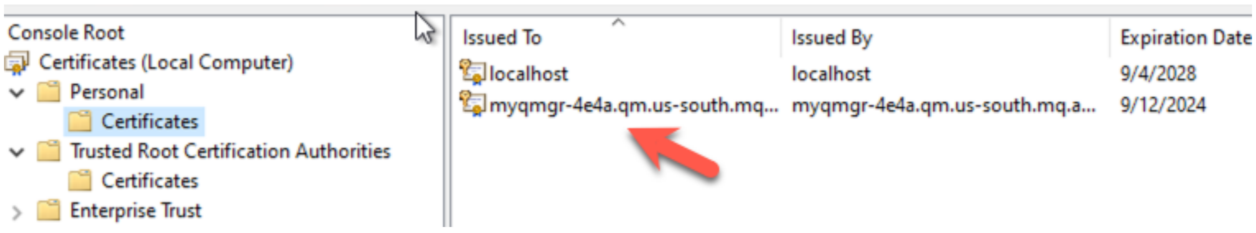
Right click on the Personal/Certificates folder and select the option to import



Select the pfx file we created in the previous step..



You should now see that in your personal key store.

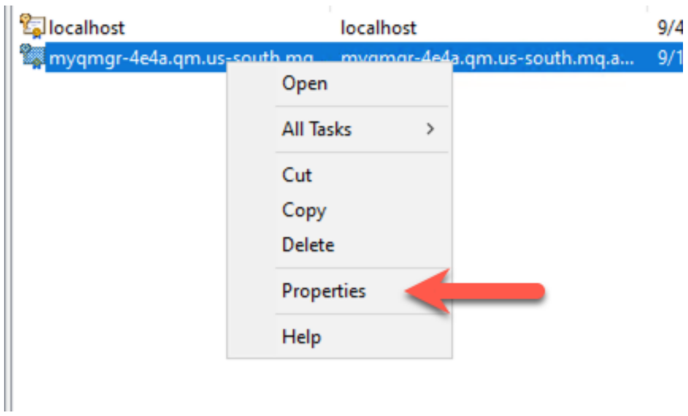


Issued To	Issued By	Expiration Date
localhost	localhost	9/4/2028
myqmgr-4e4a.qm.us-south.mq...	myqmgr-4e4a.qm.us-south.mq.a...	9/12/2024

Console Root
Certificates (Local Computer)
Personal
Certificates
Trusted Root Certification Authorities
Certificates
Enterprise Trust

Modify the label (friendly name of the cert you just imported)

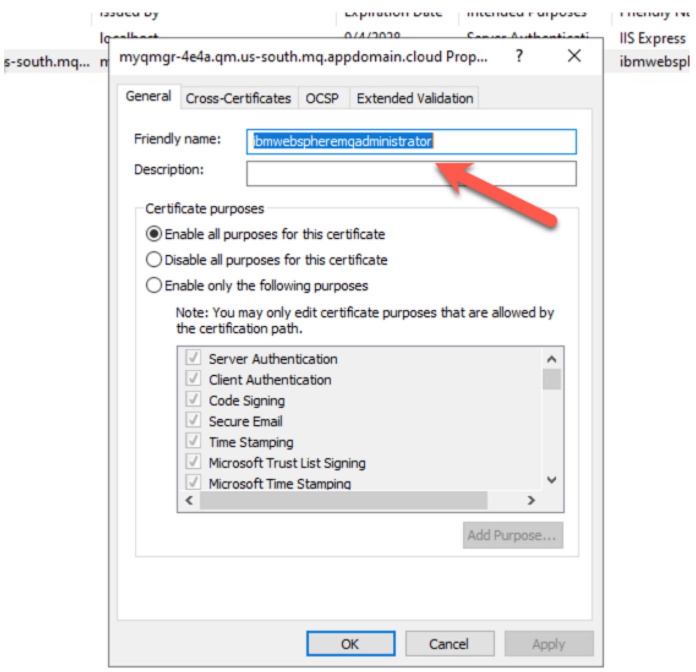
Right click on the cert in your keystore. Select properties.



This is VERY IMPORTANT.. Give your cert a friendly name in this format.

ibmwebspheremq + <The USERID of your Windows User who will be running the Application>

Note: I am logged in with the administrator userid, so that’s what I used in my example below.


IIS Express
ibmwebspl

myqmgr-4e4a.qm.us-south.mq.appdomain.cloud Prop...

General | Cross-Certificates | OCSP | Extended Validation

Friendly name: ibmwebspheremqadministrator

Description:

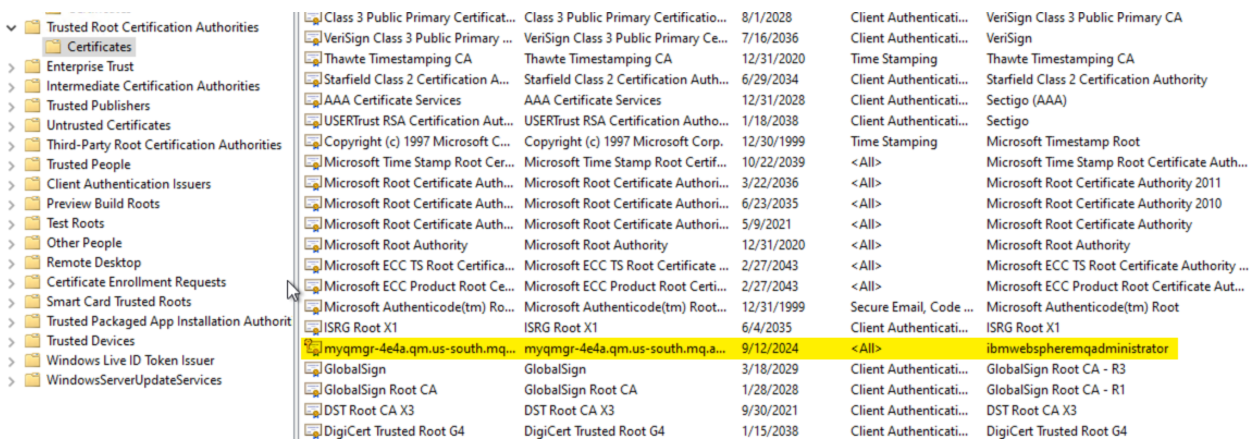
Certificate purposes

- [✓ Enable all purposes for this certificate]
- Disable all purposes for this certificate
- Enable only the following purposes

Note: You may only edit certificate purposes that are allowed by the certification path.

- ✓ Server Authentication
- ✓ Client Authentication
- ✓ Code Signing
- ✓ Secure Email
- ✓ Time Stamping
- ✓ Microsoft Trust List Signing
- ✓ Microsoft Time Stamping

IMPORTANT.. You need to perform the same steps to import the pfx file into the Trusted Root Certification Authorities keystore


- Trusted Root Certification Authorities
  - Certificates
- Enterprise Trust
- Intermediate Certification Authorities
- Trusted Publishers
- Untrusted Certificates
- Third-Party Root Certification Authorities
- Trusted People
- Client Authentication Issuers
- Preview Build Roots
- Test Roots
- Other People
- Remote Desktop
- Certificate Enrollment Requests
- Smart Card Trusted Roots
- Trusted Packaged App Installation Authorit
- Trusted Devices
- Windows Live ID Token Issuer
- WindowsServerUpdateServices

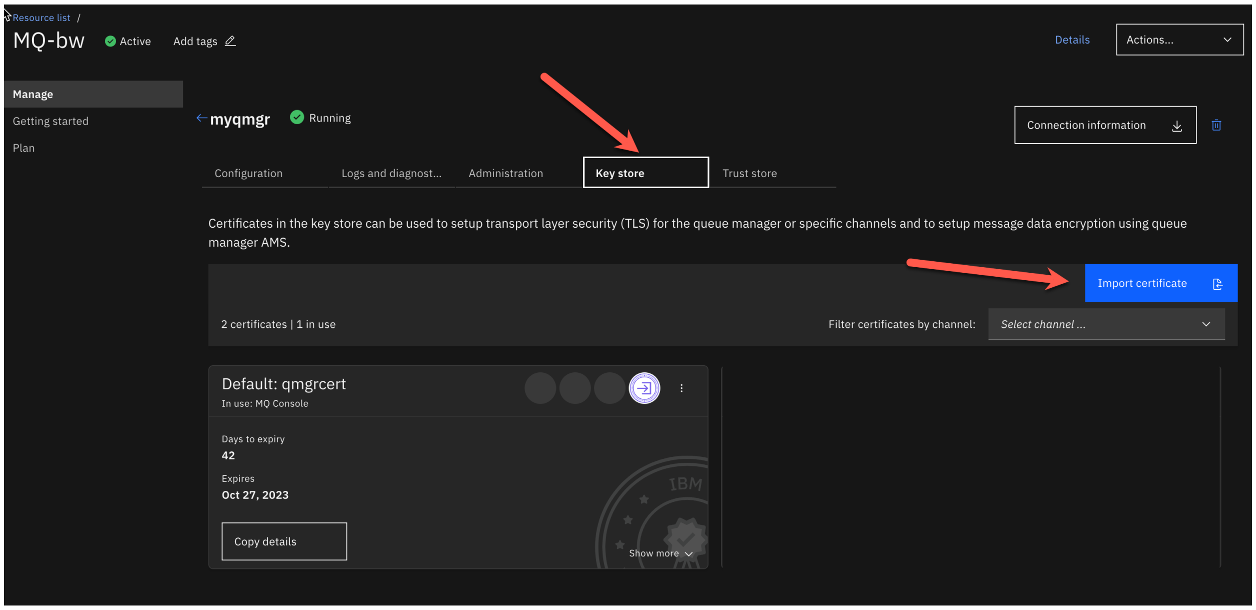
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/1/2028	Client Authenticati...	VeriSign Class 3 Public Primary CA
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	7/16/2036	Client Authenticati...	VeriSign
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping	Thawte Timestamping CA
Starfield Class 2 Certification A...	Starfield Class 2 Certification Auth...	6/29/2034	Client Authenticati...	Starfield Class 2 Certification Authority
AAA Certificate Services	AAA Certificate Services	12/31/2028	Client Authenticati...	Sectigo (AAA)
USERTrust RSA Certification Aut...	USERTrust RSA Certification Autho...	1/18/2038	Client Authenticati...	Sectigo
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping	Microsoft Timestamp Root
Microsoft Time Stamp Root Cer...	Microsoft Time Stamp Root Certif...	10/22/2039	<All>	Microsoft Time Stamp Root Certificate Auth...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	3/22/2036	<All>	Microsoft Root Certificate Authority 2011
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	6/23/2035	<All>	Microsoft Root Certificate Authority 2010
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...	5/9/2021	<All>	Microsoft Root Certificate Authority
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	<All>	Microsoft Root Authority
Microsoft ECC TS Root Certifica...	Microsoft ECC TS Root Certificate ...	2/27/2043	<All>	Microsoft ECC TS Root Certificate Authority ...
Microsoft ECC Product Root Ce...	Microsoft ECC Product Root Certi...	2/27/2043	<All>	Microsoft ECC Product Root Certificate Aut...
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...	12/31/1999	Secure Email, Code ...	Microsoft Authenticode(tm) Root
ISRG Root X1	ISRG Root X1	6/4/2035	Client Authenticati...	ISRG Root X1
myqmgr-4e4a.qm.us-south.mq...	myqmgr-4e4a.qm.us-south.mq.a...	9/12/2024	<All>	ibmwebspheremqadministrator
GlobalSign	GlobalSign	3/18/2029	Client Authenticati...	GlobalSign Root CA - R3
GlobalSign Root CA	GlobalSign Root CA	1/28/2028	Client Authenticati...	GlobalSign Root CA - R1
DST Root CA X3	DST Root CA X3	9/30/2021	Client Authenticati...	DST Root CA X3
DigiCert Trusted Root G4	DigiCert Trusted Root G4	1/15/2038	Client Authenticati...	DigiCert Trusted Root G4

Import the PEM into the keystroke for the QM on the cloud.

Log into your queue manager on the IBM Cloud.

Select the Key Store tab from the management console

Select IMPORT


Resource list /

MQ-bw Active Add tags

Details Actions...

Manage
Getting started
Plan

myqmgr Running

Connection information

Configuration | Logs and diagnost... | Administration | Key store | Trust store

Certificates in the key store can be used to setup transport layer security (TLS) for the queue manager or specific channels and to setup message data encryption using queue manager AMS.

Import certificate

2 certificates | 1 in use

Filter certificates by channel: Select channel ...

Default: qmgrcert
In use: MQ Console

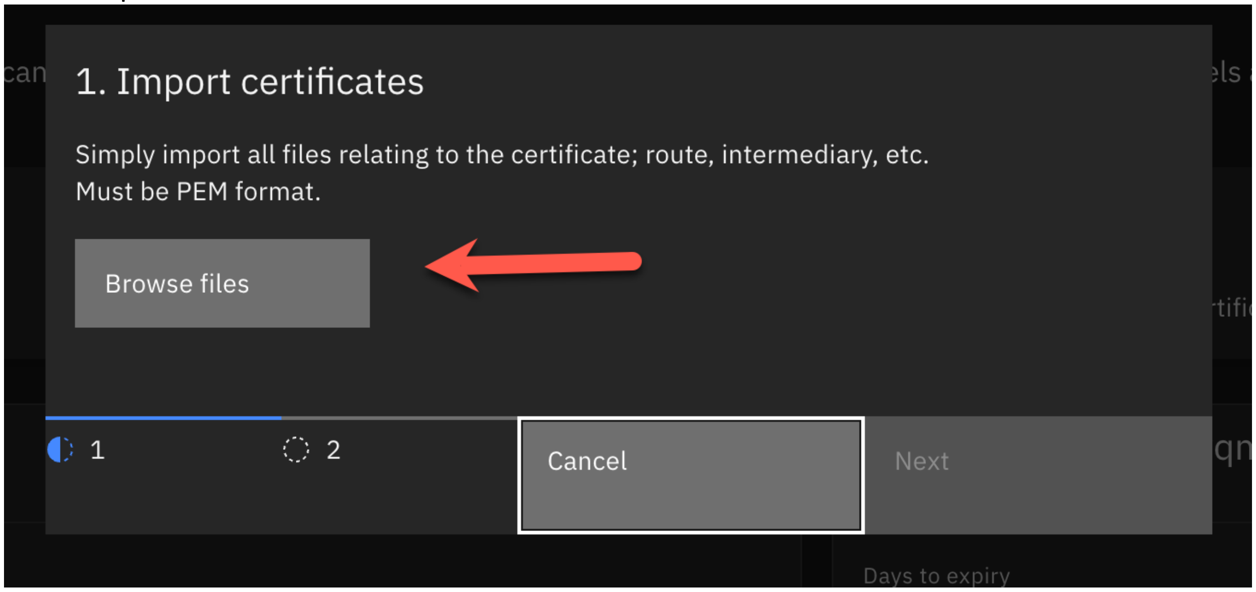
Days to expiry
42

Expires
Oct 27, 2023

Copy details

Show more

Browse to locate your PEM file. This will be the PEM file with your cert & key.. In my example it's the krier.pem file.


1. Import certificates

Simply import all files relating to the certificate; route, intermediary, etc.
Must be PEM format.

Browse files

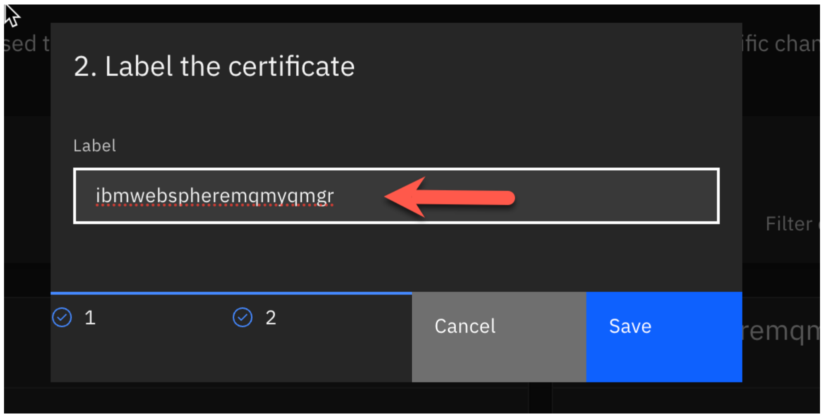
1 2

Cancel Next

Give your cert a name (Lable). This is important to label it correctly. .

You need to label it as follows.

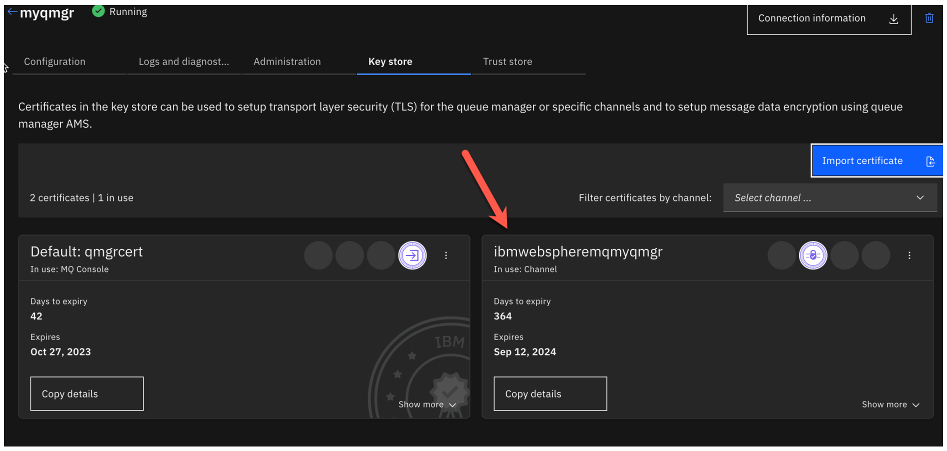
ibmwebspheremq + <Your queue manager name>



2. Label the certificate

Label

ibmwebspheremqmyqmgr



myqmgr Running

Certificates in the key store can be used to setup transport layer security (TLS) for the queue manager or specific channels and to setup message data encryption using queue manager AMS.

2 certificates | 1 in use

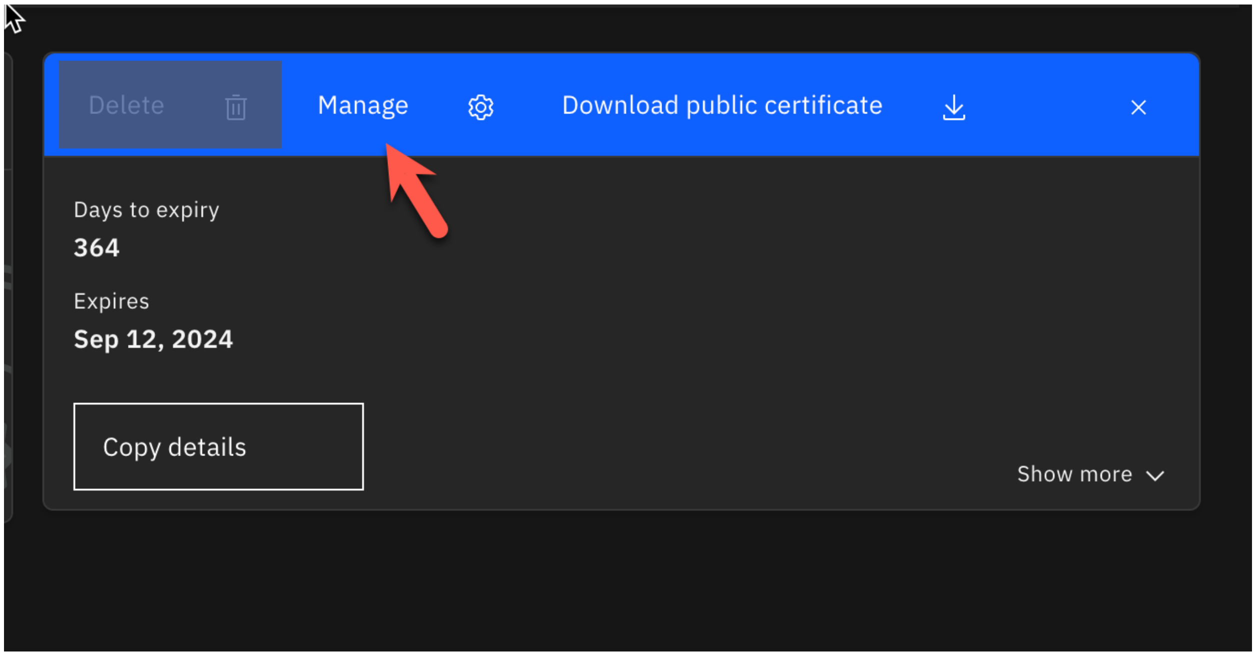
Filter certificates by channel: Select channel ...

Default: qmgrcert
In use: MQ Console
Days to expiry
42
Expires
Oct 27, 2023

ibmwebspheremqmyqmgr
In use: Channel
Days to expiry
364
Expires
Sep 12, 2024

Now we need to ensure that the cert is used for our channels.

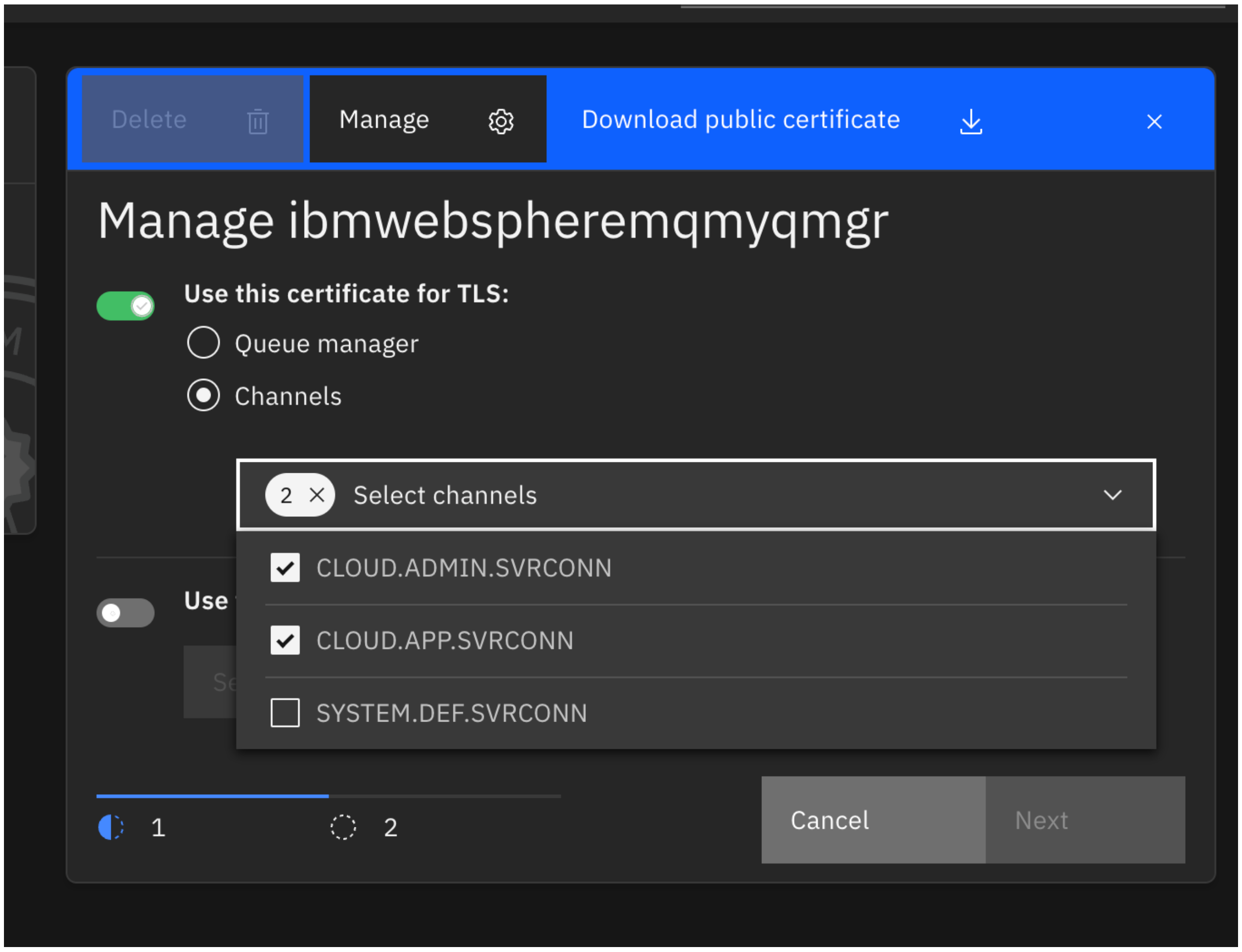
Select the 3 dots for the qert to see the management button.



Days to expiry
364

Expires
Sep 12, 2024

Select to use this cert for the channels you are connecting to. In our case it's both of these channels below.



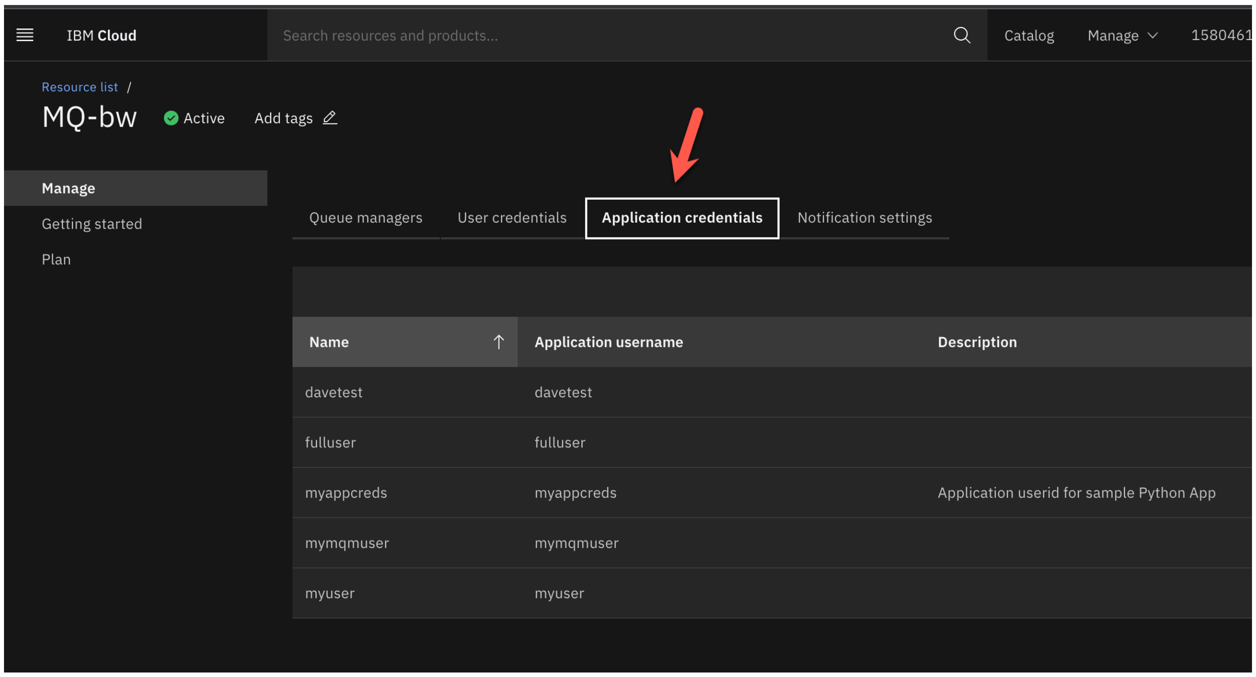
Manage ibmwebspheremqmyqmgr

Use this certificate for TLS:
Queue manager [✓ Channels]

2 Select channels
- [x] CLOUD.ADMIN.SVRCONN
- [x] CLOUD.APP.SVRCONN
- [ ] SYSTEM.DEF.SVRCONN

Create an USERID that will be used for channel authentication.

On the initial page for your IBM Cloud MQ Service, you will have the option to create a new application credential.



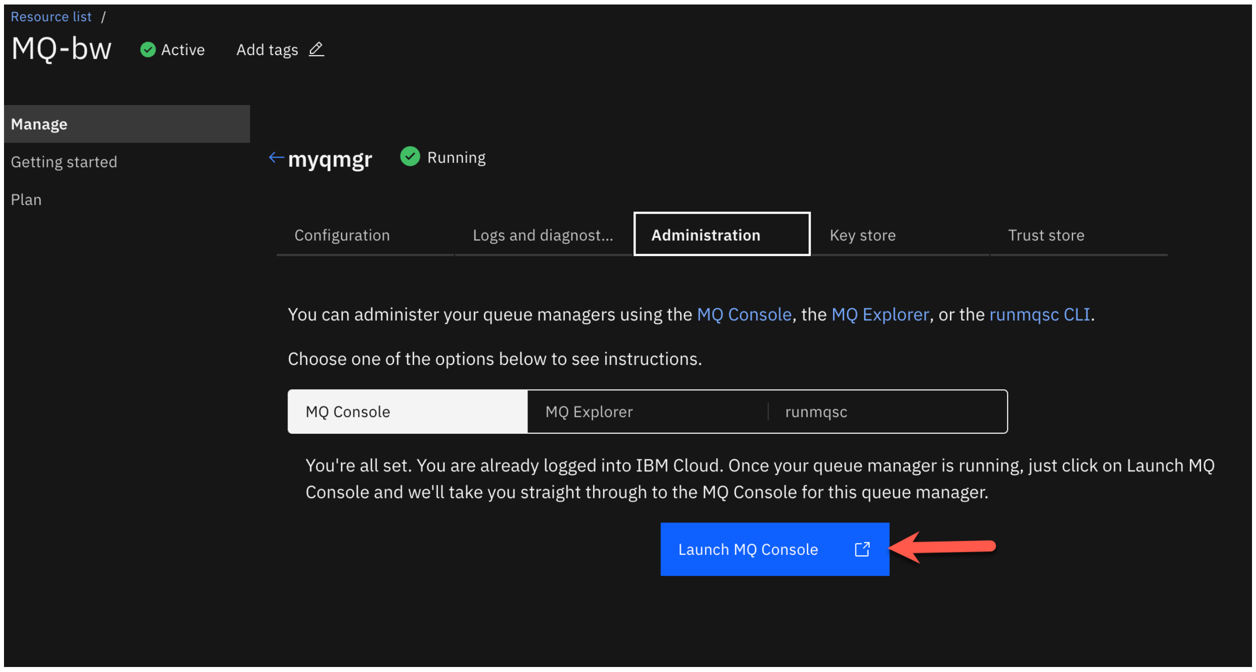
Name	Application username	Description
davetest	davetest	
fulluser	fulluser	
myappcreds	myappcreds	Application userid for sample Python App
mymqmuser	mymqmuser	
myuser	myuser	

Create an application credential.

Take note of your APIKEY, that will be your password when connecting.

Modify the SSL of the QM qm.ini Queue Manager configuration.

Next we need to make a few updates to our queue manager settings. Log into the MQ Console


Resource list /

MQ-bw Active Add tags

Manage
Getting started
Plan

myqmgr Running

Configuration | Logs and diagnost... | Administration | Key store | Trust store

You can administer your queue managers using the MQ Console, the MQ Explorer, or the runmqsc CLI.

Choose one of the options below to see instructions.

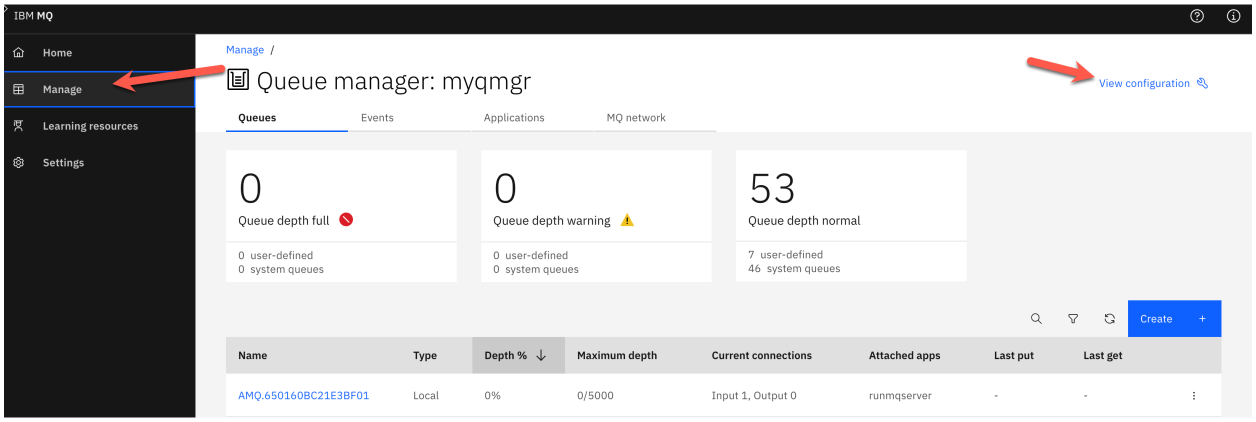
MQ Console | MQ Explorer | runmqsc

You're all set. You are already logged into IBM Cloud. Once your queue manager is running, just click on Launch MQ Console and we'll take you straight through to the MQ Console for this queue manager.

Launch MQ Console

Select manage on the left hand side to bring up the management console.

Select the “view configuration” link in the upper right hand corner.


IBM MQ

Home
Manage
Learning resources
Settings

Manage /

Queue manager: myqmgr

View configuration

Queues | Events | Applications | MQ network

0 Queue depth full — 0 user-defined, 0 system queues

0 Queue depth warning — 0 user-defined, 0 system queues

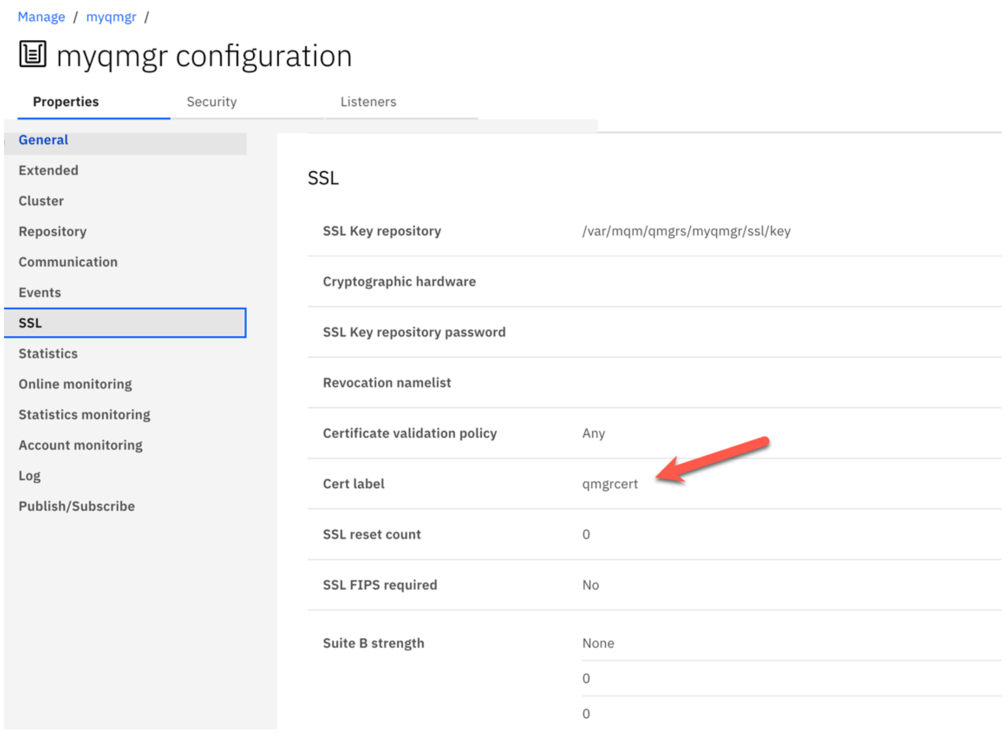
53 Queue depth normal — 7 user-defined, 46 system queues

Create +

Name	Type	Depth %	Maximum depth	Current connections	Attached apps	Last put	Last get
AMQ.650160BC21E3BF01	Local	0%	0/5000	Input 1, Output 0	runmqserver	-	-

Edit the SSL settings.

Verify the Certlabel property is set to your correct cert for the queue manager. In our case the cert for the queue manager is "qmgrcert".



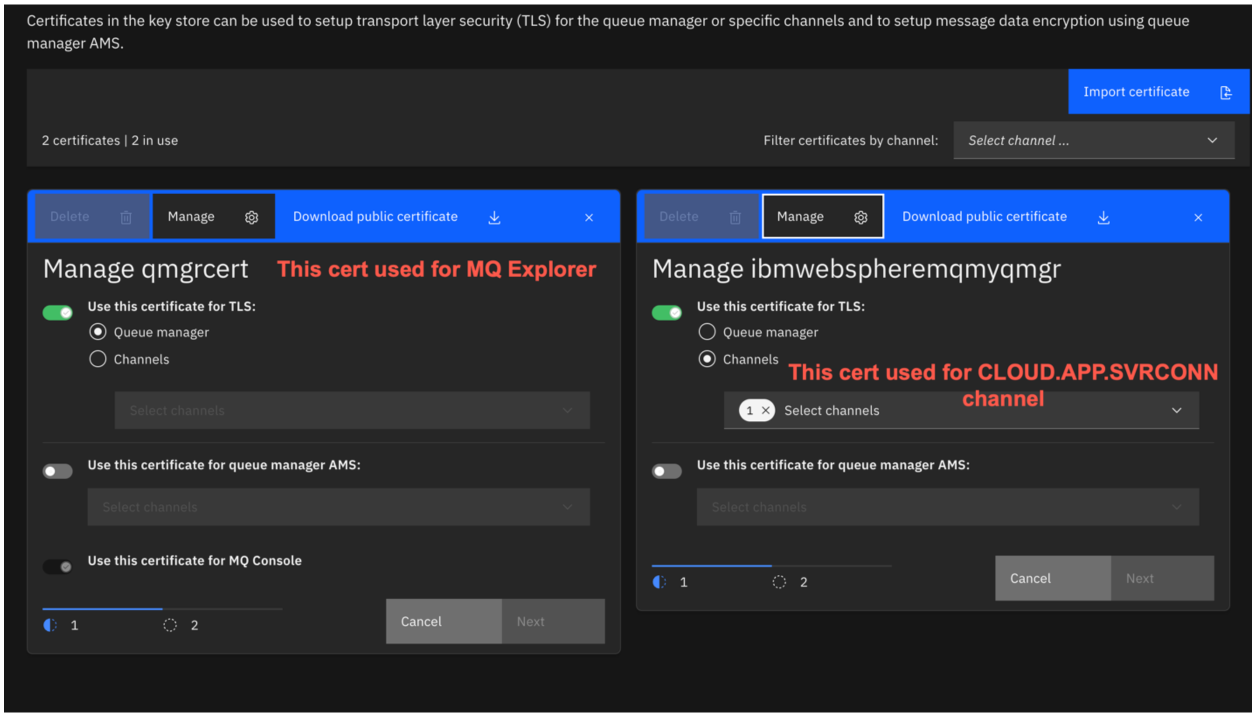
Manage / myqmgr /

myqmgr configuration

SSL

SSL Key repository	/var/mqm/qmgrs/myqmgr/ssl/key
Cryptographic hardware	
SSL Key repository password	
Revocation namelist	
Certificate validation policy	Any
Cert label	qmgrcert
SSL reset count	0
SSL FIPS required	No
Suite B strength	None
	0
	0

The queue manager certs, which we have 2 defined.



Certificates in the key store can be used to setup transport layer security (TLS) for the queue manager or specific channels and to setup message data encryption using queue manager AMS.

Import certificate
2 certificates | 2 in use
Filter certificates by channel:
Select channel ...
Delete
Manage
Download public certificate

Manage qmgrcert **This cert used for MQ Explorer**

Use this certificate for TLS: on
- [✓ Queue manager]
- Channels

Use this certificate for queue manager AMS: off

Use this certificate for MQ Console: off

Manage ibmwebspheremqmyqmgr

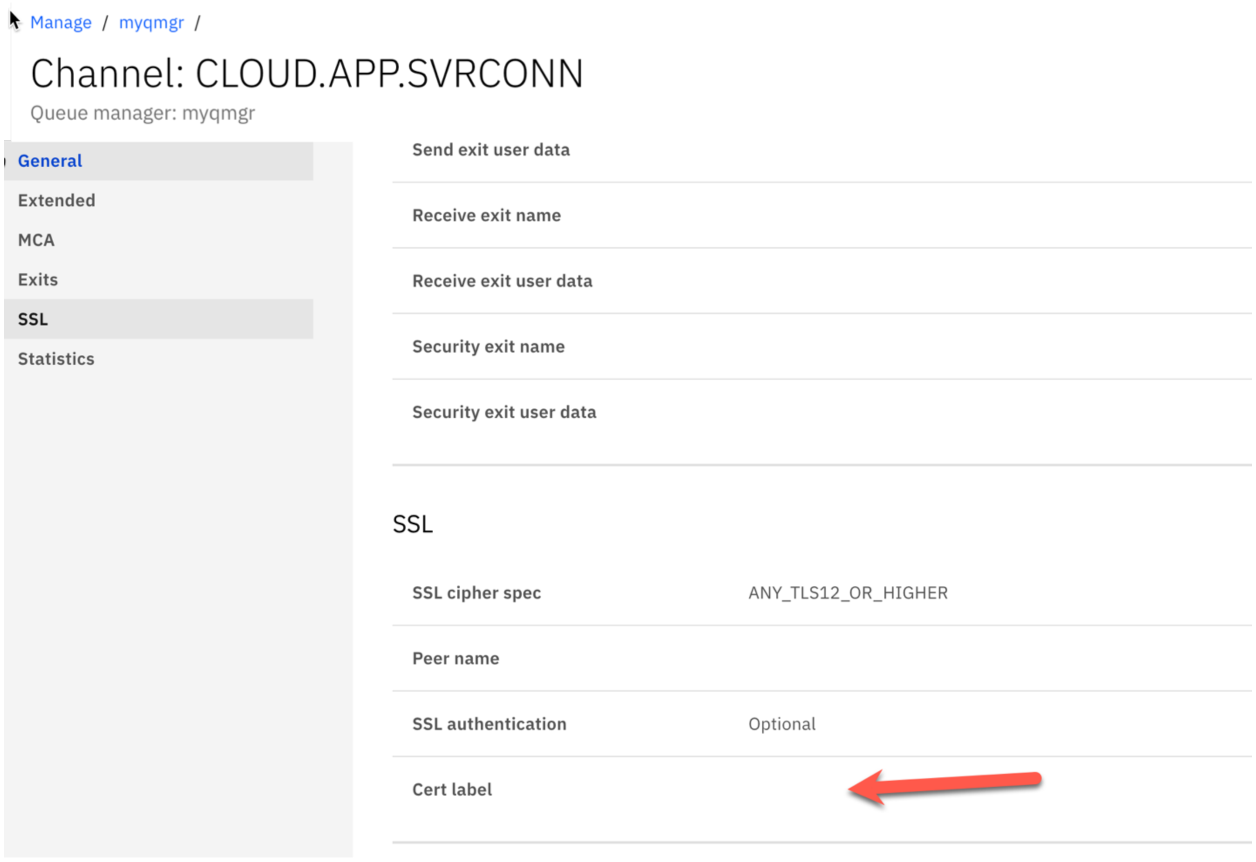
Use this certificate for TLS: on
- Queue manager
- [✓ Channels]

This cert used for CLOUD.APP.SVRCONN channel

1 × Select channels

Use this certificate for queue manager AMS: off

The channel SSL cert should label should be BLANK as displayed below. This setting can be found on the application channel settings.


Manage / myqmgr /

Channel: CLOUD.APP.SVRCONN

Queue manager: myqmgr

Send exit user data	
Receive exit name	
Receive exit user data	
Security exit name	
Security exit user data	

SSL

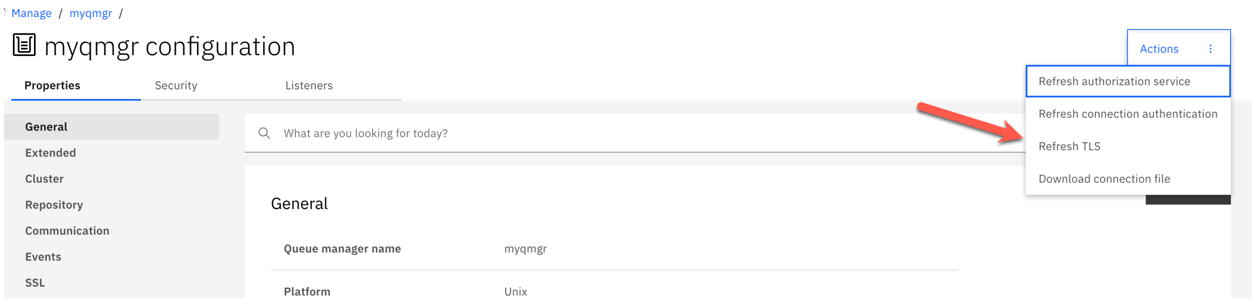
SSL cipher spec	ANY_TLS12_OR_HIGHER
Peer name	
SSL authentication	Optional
Cert label	

Any time you change the TLS certs, we need to make sure these settings are picked up by the queue manager. You need to refresh the TLS configuration.

Refresh TLS Security

At the top of the configuration page you will see a ACTION button.

Select the button and refresh the TLS settings

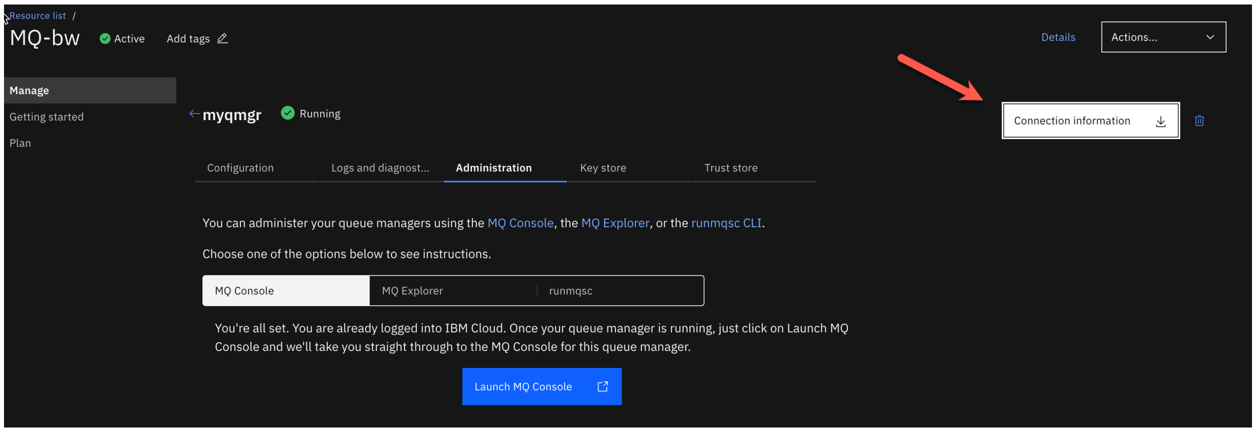

Manage / myqmgr /

myqmgr configuration

General

Queue manager name	myqmgr
Platform	Unix

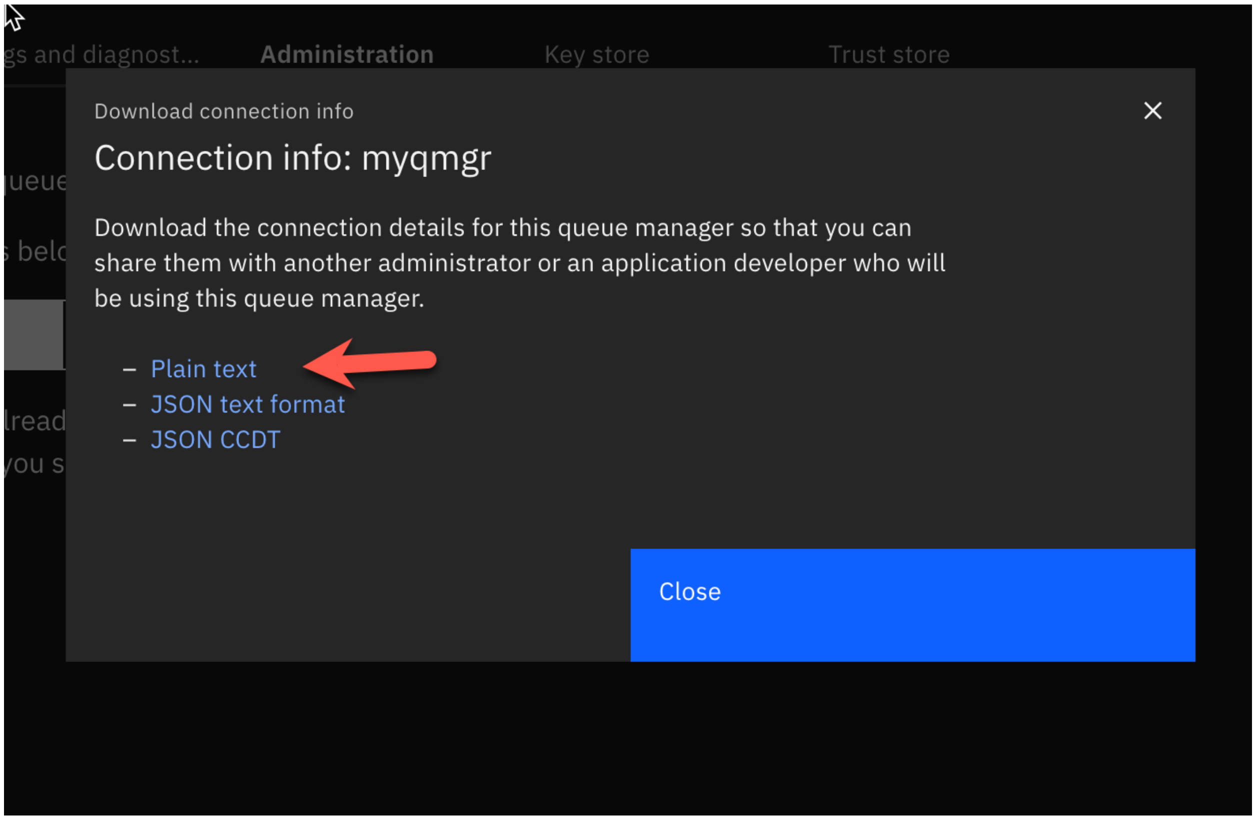
Get your connection information for your queue manager



You can administer your queue managers using the MQ Console, the MQ Explorer, or the runmqsc CLI.

Choose one of the options below to see instructions.

You're all set. You are already logged into IBM Cloud. Once your queue manager is running, just click on Launch MQ Console and we'll take you straight through to the MQ Console for this queue manager.



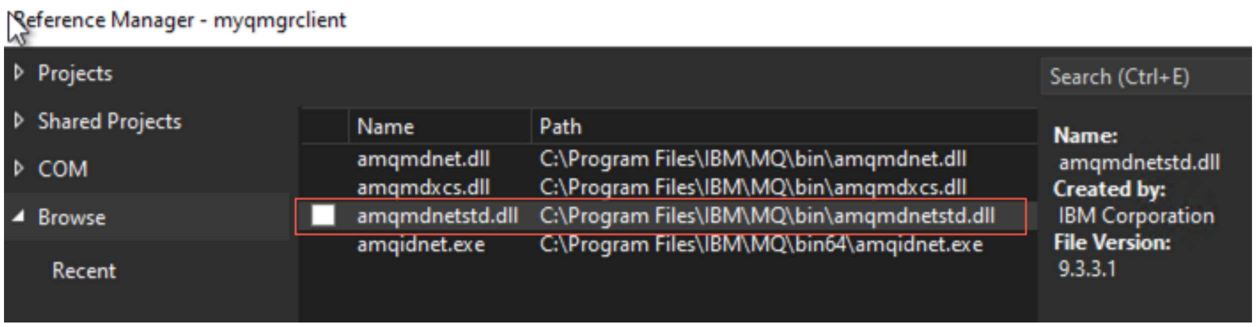
Download the connection details for this queue manager so that you can share them with another administrator or an application developer who will be using this queue manager.

We will need this information for our application.

Running a Sample Client .NET App

I am using Visual Studio code to test my applications. You need to use the IBM MQ .Net client library to connect. You have 2 options to setup your project to use the MQ Client libraries.

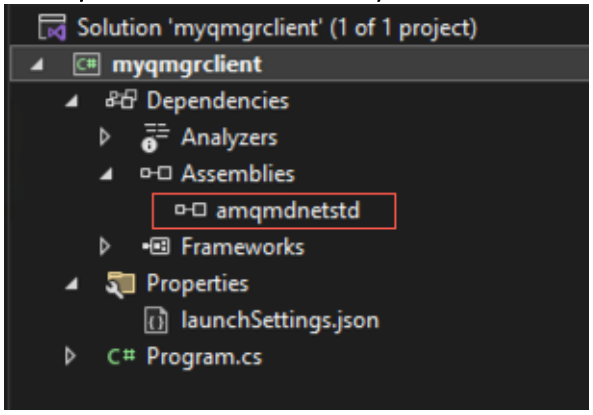
Option 1.) Add a reference to the amdmdnetstd.dll from the installed windows mq client.


Reference Manager - myqmgrclient

Name	Path
amqmdnet.dll	C:\Program Files\IBM\MQ\bin\amqmdnet.dll
amqmdxcs.dll	C:\Program Files\IBM\MQ\bin\amqmdxcs.dll
amqmdnetstd.dll	C:\Program Files\IBM\MQ\bin\amqmdnetstd.dll
amqidnet.exe	C:\Program Files\IBM\MQ\bin64\amqidnet.exe

Name: amqmdnetstd.dll
Created by: IBM Corporation
File Version: 9.3.3.1

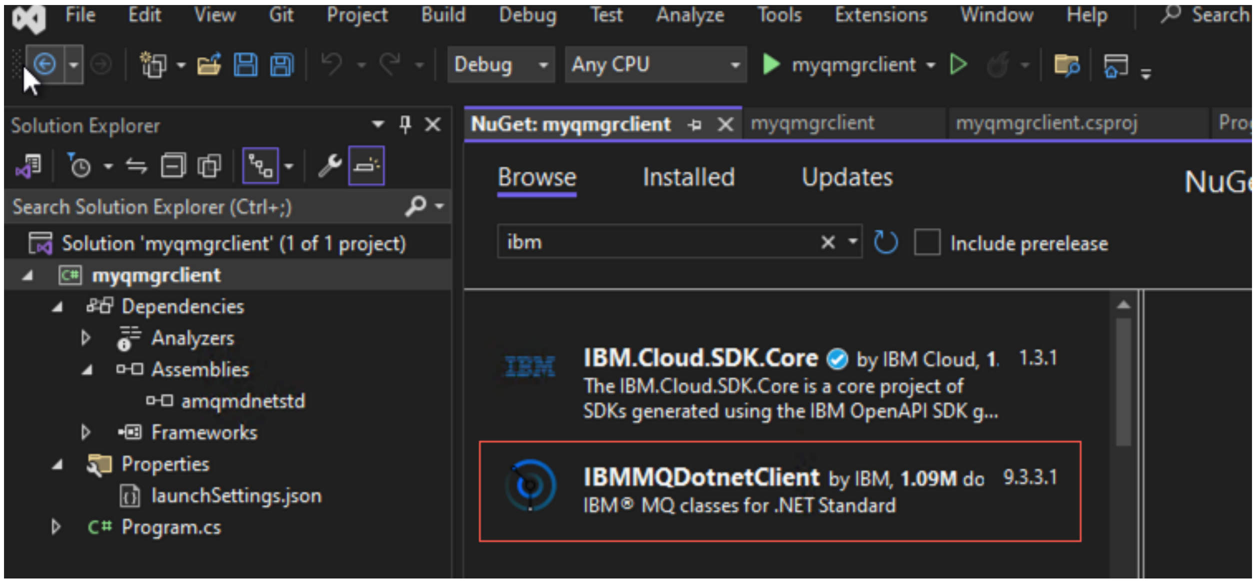
After you add this reference you should see it in your project as displayed below.


```
Solution 'myqmgrclient' (1 of 1 project)
  myqmgrclient
    Dependencies
      Analyzers
      Assemblies
        amqmdnetstd
      Frameworks
    Properties
      launchSettings.json
    Program.cs
```

The advantage of this option is that you can run MQ Client Tracing, to help you debug.

Option 2.) You can use nuGet from Visual Studio IDE to add the IBM MQ Client to your project.

Search for packages from IBM and you will find the IBMMQDotnetClient package.


NuGet: myqmgrclient — Browse | Installed | Updates

Search: ibm — Include prerelease

IBM.Cloud.SDK.Core by IBM Cloud, 1. 1.3.1
The IBM.Cloud.SDK.Core is a core project of SDKs generated using the IBM OpenAPI SDK g...

IBMMQDotnetClient by IBM, 1.09M do 9.3.3.1
IBM® MQ classes for .NET Standard

*** Below is the sample code ***

```
using IBM.WMQ;
using System;
using System.Collections;
using System.Text;

namespace myqmgrclient
{
   internal class Program
   {
      static void Main(string[] args)
      {
         string strQueueManagerName = "myqmgr";
         string strChannelName = "CLOUD.APP.SVRCONN";
         string strQueueName = "DEV.QUEUE.1";
         string strServerName = "myqmgr-4e4a.qm.us-south.mq.appdomain.cloud";
         int intPort = 30762;
         string strMsg = "Hello IBM, this is a message";


          Hashtable queueProperties = new Hashtable
         {
            { MQC.HOST_NAME_PROPERTY, strServerName },
            { MQC.CHANNEL_PROPERTY, strChannelName },
            { MQC.PORT_PROPERTY, intPort },
            { MQC.CONNECT_OPTIONS_PROPERTY, MQC.MQCNO_RECONNECT },
            { MQC.TRANSPORT_PROPERTY, MQC.TRANSPORT_MQSERIES_MANAGED },
            { MQC.SSL_CERT_STORE_PROPERTY, "*SYSTEM" },
            { MQC.SSL_CIPHER_SPEC_PROPERTY, "TLS_RSA_WITH_AES_128_CBC_SHA256" }
         };

         //Set Username
         MQEnvironment.UserId = "davetest";

         //Set Passowrd
         MQEnvironment.Password = "F-5QDP8lO_cl0j7521wEchXxzd2Yv7DeA_gPtttqa28ASBV";

         //Define a Queue Manager
         try
         {
            Console.WriteLine(Environment.UserName.ToLower());
            Console.WriteLine(MQEnvironment.CertificateLabel);
            MQQueueManager myQM =
                  new MQQueueManager(strQueueManagerName, queueProperties);

            // creating a message object
            MQMessage queueMessage = new MQMessage();
            queueMessage.WriteString("test message");

            //Define a Queue
```

```
                var queue = myQM.AccessQueue
                (strQueueName, MQC.MQOO_OUTPUT + MQC.MQOO_FAIL_IF_QUIESCING);
                MQPutMessageOptions queuePutMessageOptions = new MQPutMessageOptions();
                queue.Put(queueMessage, queuePutMessageOptions);
                queue.Close();
                Console.WriteLine("Success");
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex);
                Console.WriteLine(ex.Message);
            }
            Console.ReadLine();
        }
    }
}
```

Note: There are more samples provided with the MQ Client.

Location: C:\Program Files\IBM\MQ\tools\dotnet\samples\

Debugging – Connectivity Issues..

On the Client Application

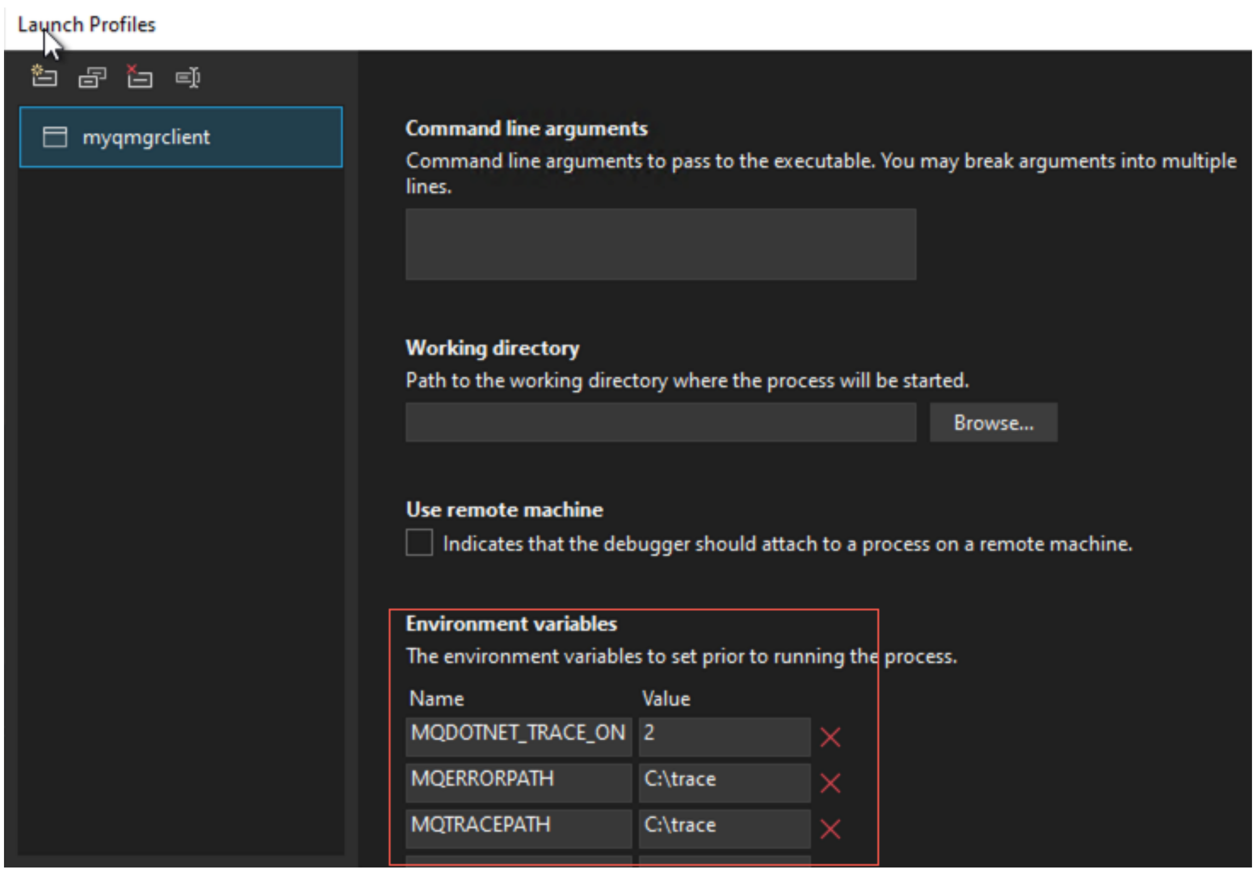
I would highly recommend you turn on MQ Tracing for your windows client.

IMPORTANT: This requires you to reference the IBM MQ Client DLL from the installed MQ Client on your local windows machine.

You will need to set 3 environment variables for your application.

Select the debug menu and then select debug properties to show this window below. Set the environment variables.

NOTE: *You need to close down Visual Studio and reopen it to get the variables to be picked up by the editor.*



Launch Profiles

myqmgrclient

Command line arguments

Command line arguments to pass to the executable. You may break arguments into multiple lines.

Working directory

Path to the working directory where the process will be started.

Browse...

Use remote machine

Indicates that the debugger should attach to a process on a remote machine.

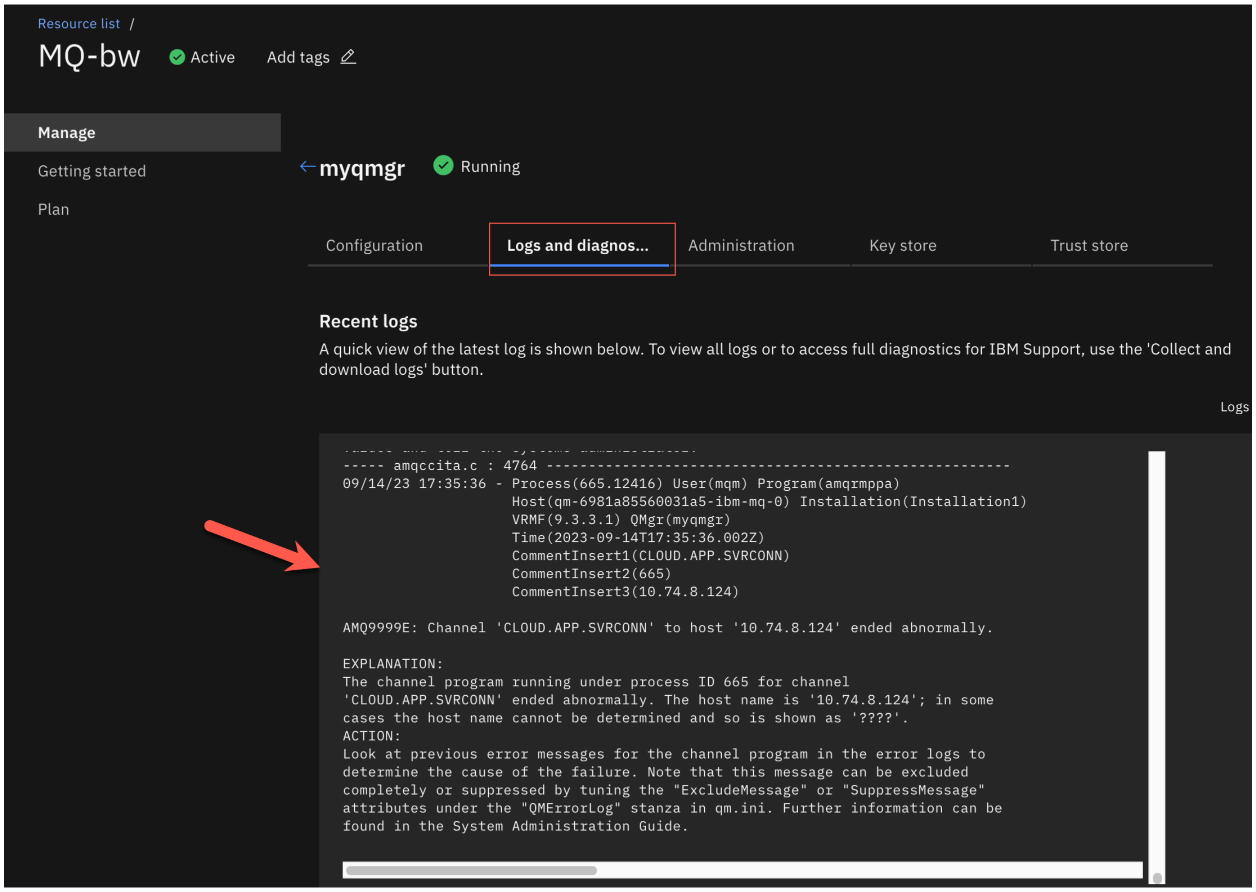
Environment variables

The environment variables to set prior to running the process.

Name	Value
MQDOTNET_TRACE_ON	2
MQERRORPATH	C:\trace
MQTRACEPATH	C:\trace

On the MQ Server.

You can see any connection errors in the logs on the server. Below is a screen shot of the logs.



Recent logs

A quick view of the latest log is shown below. To view all logs or to access full diagnostics for IBM Support, use the 'Collect and download logs' button.

```
----- amqccita.c : 4764 --------------------------------------------------------
09/14/23 17:35:36 - Process(665.12416) User(mqm) Program(amqrmppa)
                    Host(qm-6981a85560031a5-ibm-mq-0) Installation(Installation1)
                    VRMF(9.3.3.1) QMgr(myqmgr)
                    Time(2023-09-14T17:35:36.002Z)
                    CommentInsert1(CLOUD.APP.SVRCONN)
                    CommentInsert2(665)
                    CommentInsert3(10.74.8.124)

AMQ9999E: Channel 'CLOUD.APP.SVRCONN' to host '10.74.8.124' ended abnormally.

EXPLANATION:
The channel program running under process ID 665 for channel
'CLOUD.APP.SVRCONN' ended abnormally. The host name is '10.74.8.124'; in some
cases the host name cannot be determined and so is shown as '????'.
ACTION:
Look at previous error messages for the channel program in the error logs to
determine the cause of the failure. Note that this message can be excluded
completely or suppressed by tuning the "ExcludeMessage" or "SuppressMessage"
attributes under the "QMErrorLog" stanza in qm.ini. Further information can be
found in the System Administration Guide.
```

Links of interest

- Connecting to IBM MQ with SSL Article
- How to import intermediate and root certificates via MMC
- Troubleshooting MQ SSL
- Sample code repository
- (Download) – Sample instructions for SSL
- Python SSL Unmanaged .Net Framework Sample How To
- MQ Documentation – Configure TLS for Managed .NET App
- Cyper Spec for MQ Information
- Key Repository's for Managed .Net Applications
- Running .Net Core application on Linux
- How to perform common IBM Management Cert Tasks
- IBM MQ in Kubernetes – Enabling TLS with signed CA
- Generating PK12 Certs
- Creating self signed Certs for use with IBM MQ on Cloud
- Stack Overflow – 2393 Error - Resolution
- Stack Overflow – MQ TLS Error Resolution with Tracing

- How to create a PEM file
- Documentation – Understanding the Certificate Label Requirements
- How to create a pfx file from cert and private key
-