## Configure codeready

## 

## Start and log into codeready

crc start -n 1.1.1.1 –disk-size 40

crc oc-env

@FOR /f "tokens=\*" %i IN ('crc oc-env') DO @call %i

crc console

oc login -u kubeadmin -p HqC3I-wgtiB-q7qCf-KEsuK <https://api.crc.testing:6443>

oc new project da-mq OR oc project da-mq

## build and deploy

### Input Image stream

I have a copy of the MQ for Developers image on my docker repos but the actual image can be used.

kind: ImageStream

apiVersion: image.openshift.io/v1

metadata:

  name: ibm-mqadvanced-server-integration

  namespace: da-mq

spec:

  lookupPolicy:

    local: false

  tags:

    - name: 9.1.3.0-r4-amd64

      annotations: null

      from:

        kind: DockerImage

        name: 'davexacom/ibm-mqadvanced-server-integration:9.1.3.0-r4-amd64'

      generation: 2

      importPolicy: {}

      referencePolicy:

        type: Source

status:

  dockerImageRepository: >-

    image-registry.openshift-image-registry.svc:5000/da-mq/ibm-mqadvanced-server-integration

  publicDockerImageRepository: >-

    default-route-openshift-image-registry.apps-crc.testing/da-mq/ibm-mqadvanced-server-integration

  tags:

    - tag: 9.1.3.0-r4-amd64

      items:

        - created: '2021-01-08T06:00:40Z'

          dockerImageReference: >-

            davexacom/ibm-mqadvanced-server-integration@sha256:0cd8142e68ae76a84b06540fd93caf9f964045dbd35c6d0c7efb309e29f2bbb0

          image: >-

            sha256:0cd8142e68ae76a84b06540fd93caf9f964045dbd35c6d0c7efb309e29f2bbb0

          generation: 2

create the input file and use oc create to create it.

oc create -f c:\openshift\data\build-mq-custom.yaml

oc adm policy add-scc-to-user anyuid -z default

scc "anyuid" added to: ["system:serviceaccount:da-mq:default"]

### Build

oc new-build https://github.com/DAVEXACOM/ibm-mqadvanced-server-tls-build.git

### deploy

oc new-app ibm-mqadvanced-server-tls-build --env LICENSE=accept --env MQ\_QMGR\_NAME=TLSPRQM1

oc status

### Create/Expose the service as a route

This is the “base” route, we’ll need to create the TLS route as well that maps the SNI to the TLS channel.

oc expose service/ibm-mqadvanced-server-tls-build

### Create the TLS route

This is the route that maps the SNI to the channel name

This host - tlsprqm12e-svrconn.chl.mq.ibm.com is the mapping for SVRCONN channel name TLSPRQM1.SVRCONN

kind: Route

apiVersion: route.openshift.io/v1

metadata:

name: tls-tlsprqm1p

namespace: da-mq

labels:

app: ibm-mqadvanced-server-tls-build

spec:

host: tlsprqm12e-svrconn.chl.mq.ibm.com

to:

kind: Service

name: ibm-mqadvanced-server-tls-build

weight: 100

port:

targetPort: 1414-tcp

tls:

termination: passthrough

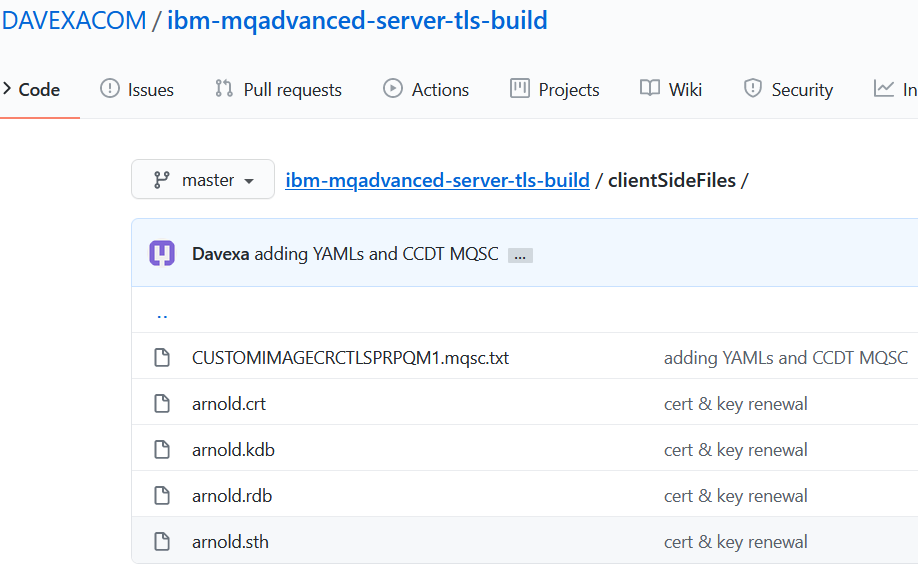
wildcardPolicy: None

oc create -f route-tls-tlsprqm1p.yaml

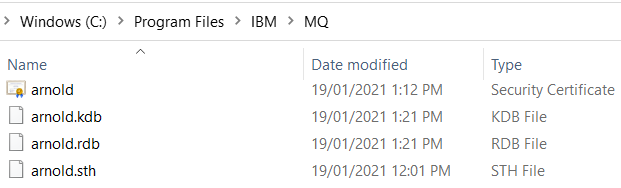
## Connect and test

### Client side keys and cert files

Download the following files from <https://github.com/DAVEXACOM/ibm-mqadvanced-server-tls-build/tree/master/ClientSideFiles>



Copy to (or linux equivalent) default location c:\Program Files\IBM\MQ



### Create Client Connection channel in the CCDT

### Edit the MQSC file

Download the MQSC file from <https://github.com/DAVEXACOM/ibm-mqadvanced-server-tls-build/blob/master/ClientSideFiles/CUSTOMIMAGECRCTLSPRQM1.mqsc>

You get the CONNAME from the “base” route yaml created by the oc expose command, so Get the hostname from the route generated by the oc expose service and update the mqsc file

|  |
| --- |
| DEFINE CHANNEL(TLSPRQM1.SVRCONN) + |
|  |

|  |
| --- |
| CHLTYPE(CLNTCONN) + |
|  |

|  |
| --- |
| TRPTYPE(TCP) + |
|  |

|  |
| --- |
| CONNAME('ibm-mqadvanced-server-tls-build-da-mq.apps-crc.testing(443)') + |
|  |

|  |
| --- |
| CERTLABL('ibmmqarnold') + |
|  |

|  |
| --- |
| QMNAME('TLSPRQM1') + |
|  |

|  |
| --- |
| SSLCIPH(ANY\_TLS12) + |
|  |

REPLACE

### Set up the Environment in the Command Line window

Clear the MQSERVER variable

SET MQSERVER=

set MQSSLKEYR=C:\Program Files\IBM\MQ\arnold

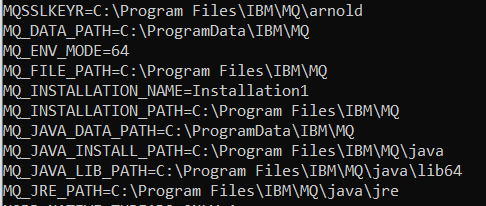
Clear the MQCHLLIB and MQCHLTAB variables

SET MQCHLLIB=

SET MQCHLTAB=

Check the MQClient.ini file as well to ensure there is no SSL stanza set.

Run SET on the command line and double check the MQ environment variables



### Runmqsc with -n flag to create CCDT

use runmqsc with -n flag and pipe in the MQSC file to create the MQ CCDT Table

runmqsc -n < CUSTOMIMAGECRCTLSPRPQM1.mqsc

5724-H72 (C) Copyright IBM Corp. 1994, 2019.

Starting local MQSC for 'AMQCLCHL.TAB'.

1 : DEFINE CHANNEL(TLSPRQM1.SVRCONN) +

: CHLTYPE(CLNTCONN) +

: TRPTYPE(TCP) +

: CONNAME('ibm-mqadvanced-server-tls-build-da-mq.apps-crc.testing(443)') +

: CERTLABL('ibmmqarnold') +

: QMNAME('TLSPRQM1') +

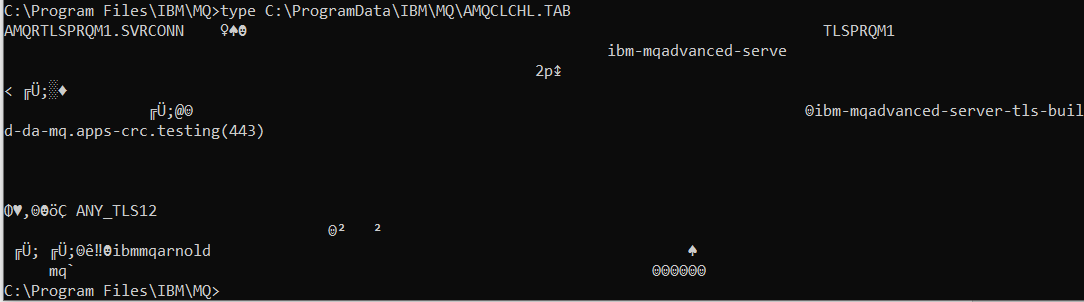
: SSLCIPH(ANY\_TLS12) +

: REPLACE

AMQ8014I: IBM MQ channel created.

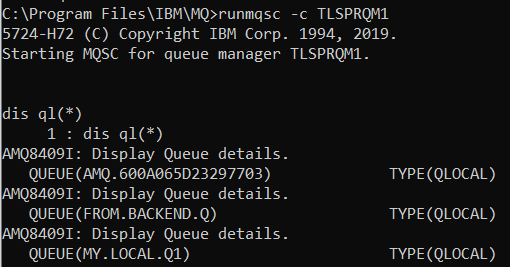
No commands have a syntax error.

### Check the CCDT Table and copy to default location

type C:\ProgramData\IBM\MQ\AMQCLCHL.TAB 

copy the AMQCLCHL.TAB file to the default location

copy C:\ProgramData\IBM\MQ\AMQCLCHL.TAB C:\Program Files\IBM\MQ



End

