Installation Procedure of RCW

1. Prerequisites for RCW Installation
    a. Ubuntu server 20.04 with following specs
        i. Disk partition should be done properly
        ii. Ip should be static
        iii. Openssh-server should be enable
        iv. Firewall be disable
    b. TIS machine should have window server 2016 and above(Only for TETRA integration)
        i. Gstreamer should be installed
        ii. Vc_redict 2015 and above should be installed
        iii. Firewall should be disable
        iv. Ip should be static
        v. Should be ping with Tetra BSC
        vi. Make sure TetraFlex dongle be inserted
    c. RCW Client workstation should have window 10 and above
        i. Google chrome should be installed with 103 version
        ii. Update of chrome should be disable
        iii. IP should be static
        iv. Firewall should be disable
        v. Desk microphone should be connected
        vi. Usb speakers should be connected
2. Install kubernetes as platform
    a. Take access of DAS server through putty
    b. Enter ip, username and password to login
    c. Download 'microk8s_source.tar.gz' file from given link : https://drive.google.com/file/d/1zBGcXPJH57F0yokQH0SXqQbM3bSAxHV-/view?usp=sharing
    d. Transfer 'microk8s_source.tar.gz' file to DAS server
    e. Run the following commands
        a. `sudo su`
        b. `tar xvzf microk8s_sources.tar.gz`

c. `cd microk8s_sources`
d. `dpkg -i *.deb`
e. `systemctl enable snapd --now`
f. `ln -s /var/lib/snapd/snap /snap`
g. `ln -s /var/lib/snapd/snap /snap`
h. `ls *.assert | xargs -n 1 snap ack`
i. `ls *.snap | xargs -n 1 snap install --classic`
j. `/snap/bin/microk8s start`
k. `ls *.tar| xargs -n 1 -I {} /snap/bin/microk8s ctr i import $(pwd)/{}`
l. `microk8s enable dns storage`
m. `sudo snap alias microk8s.kubectl kubectl`

f. Run 'microk8s status' and see the status

Output :

```
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dns            # CoreDNS
    ha-cluster     # Configure high availability on the current node
    registry       # Private image registry exposed on localhost:32000
    storage # Storage class; allocates storage from host directory
```

g. Now kubernetes installation has been done, After this need to install MCX applications
h. Make a TAR directory and transfer all tar files(Given by software team) to TAR directory
i. Transfer k8s and certificate files to ubuntu server with the same name.
j. Go to TAR directory by command : cd TAR and run below command

```
ls *.tar| xargs -n 1 -I {} /snap/bin/microk8s ctr i import $(pwd)/{}
```

k. Go to the K8s folder: cd k8s
    i. Configure the mcxserver.configmap.yaml according to your server and save.
    ii. Configure the ris.configmap.yaml according to your server and save
    iii. Configure the dispatcher.deployment.yaml according to your server ip and save
l. Apply the services from k8s directory
    i. kubectl apply -f services
m. Apply the deployment files with following command
    i. kubectl apply -f mysql.pv.yaml
    ii. kubectl apply -f mysql.deployment.yaml
    iii. kubectl apply -f kms.deployment.yam
    iv. kubectl apply -f drachtio.deployment.yaml
    v. kubectl apply -f idms.deployment.yaml
    vi. kubectl apply -f mcxserver.configmap.yaml
    vii. kubectl apply -f mcxserver.deployment.yaml
    viii. kubectl apply -f dispatcher.deployment.yaml
    ix. kubectl apply -f ris.configmap.yaml
    x. kubectl apply -f ris.deployment.yaml
n. Check status with command
    i. kubectl get pods

Output :

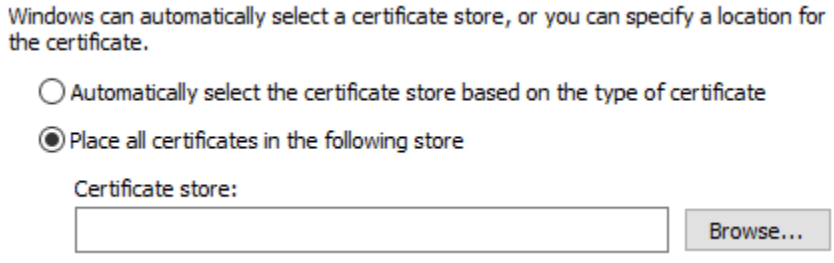
mysql-54b8579d84-dpl2q	1/1	Running	0 (25h ago)	37d
kms-deployment-bb6d56994-8b796	1/1	Running	0 (25h ago)	12d
Drachtio-deployment-78c87f7d67-m	1/1	Running	0 (25h ago)	25d
idms-847ff6445c-6jnq5	1/1	Running	0 (25h ago)	37d
mcxserver-7bcdbb685f-zgsjj	1/1	Running	0	19h
dispatcher-798868866f-jksv6	1/1	Running	0	3h53m
ris-664b77bf67-tgclb	1/1	Running	0	25h

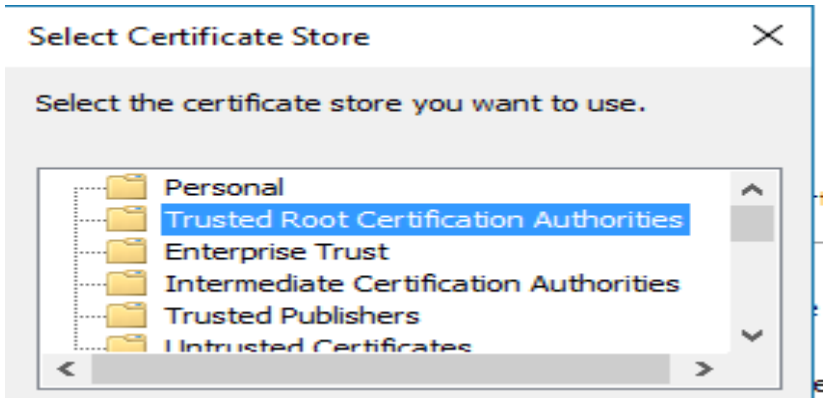
o. If output is like above that means installation has completed properly

p. Now time to login dispatcher client with following steps

   i. Install rootCA certificate on dispatcher client(only application for window)
   Download link:
   https://drive.google.com/file/d/1kZ876EcWmnpam-S9kcqifk-73o62sQHe/view?usp=sharing

      a. To install, double click on rootCA
      b. Select Local User
      c. After this two option will will shown, select 2nd one and browse





d. After that click FINISH and ok
e. Installation of certificate has been installed

ii. Open chrome, enter DAS ip and port
Ex: https://192.168.1.137:30300

iii. Enter with username - admin with default password - 12345678
iv. Create your own client id and password
v. Now enter created id and password and press login
vi. After logging select the input and output peripherals to make communication. Make sure I/O peripherals are selected otherwise it will created issue in communication

3. Steps to update deployments

1. Make sure which deployment you have update and for deployment you should have deployment TAR's and corresponding configuration
2. Upload the TAR's on the server
3. Now import/extract the TAR's by command:
   microk8s ctr image import **tar_name**
4. Delete the deployment file by command:
   kubectl delete -f folder/deployment.yaml
5. Apply the deployment
   kubectl delete -f folder/deployment.yaml
6. Wait to established the pod and the check status of pods
   kubectl get pods

Example 👍 :

Suppose you have a tar file : mcxserver.tar

To import tar : microk8s ctr image import mcxserver.tar

After successfully done delete the deployment, my deployment folder is k8s:

kubectl delete -f k8s/deployment.yaml

Apply deployment file: kubectl apply -f k8s/deployment.yaml

FAQ

1. ImagepullBackError:
   To resolve the issue verify the image name that you have imported and the image name of the deployment file. Correct the image name and restart the pod
2. RunContainerError:
   Verify the certificate path in the deployment files