**Assignment 19 – Kubernetes Test Questions 8**

Date: 21 May 2024

Name: Mohamad Haffiz Bin Mohd Hissham

**Questions:**

Create

a **new user** “**sam**”.

Grant him access to the cluster.

User “sam”

should have **permission** to **create**, **list**, **get**, **update** and **delete** **pods**.

The private key exists at location:

**/root/sam.key** and csr at **/root/sam.csr**

**References:**

* [Certificates and Certificate Signing Requests | Kubernetes](https://kubernetes.io/docs/reference/access-authn-authz/certificate-signing-requests/)

**Steps:**

1. **Create private key** for user and store it under **/root/sam** folder as **sam.key**, **openssl genrsa -out /root/sam.key 2048**
2. Create a CertificateSigningRequest (CSR) and under **/root/** folder store it as **sam.csr**, **openssl req -new -key /root/sam.key -out /root/sam.csr**
   1. View **sam.csr** file to copy request code
3. Get the content of CSR created
   1. Decode the CSR using base64, **cat sam.csr | base64 | tr -d "\n"**
   2. Left everything empty for now
   3. Copy **CSR key** created above to paste it on **sam.yaml** file created after this step
   4. A screenshot of a computer

      Description automatically generated
4. Create YAML file for user creation, **nano sam.yaml**
   1. Copy the script on **Create a CertificateSigningRequest**’s section
   2. Copy **CSR request key** created on step 3 and paste it on **request attribute**
   3. Get the key until before ***controlplane $***
   4. **Edit the name** under metadata as well

apiVersion: certificates.k8s.io/v1

kind: CertificateSigningRequest

metadata:

  name: sam

spec:

  request: LS0tLS1CRUdJTi<....>DQVRFIFJFUVVFU1QtLS0tLQo=

  signerName: kubernetes.io/kube-apiserver-client

  expirationSeconds: 86400  # one day

  usages:

  - client auth

1. Apply the YAML, **kubectl apply -f sam.yaml**
2. Get the list of CSRs created, **kubectl get csr**
   1. A screenshot of a computer

      Description automatically generated
3. Approve the **CSR created** for user **sam**, **kubectl certificate approve sam**
   1. View list of CSRs again
   2. Additionally, can delete CSRs created as well if needed. **kubectl delete csr sam**
   3. A screenshot of a computer program

      Description automatically generated
4. **Certificate issued** is in **Base64-encoded** format, **export** the issued certificate from the CSR **as** **decoded** certificate. Save it as **sam.crt**
   1. **kubectl get csr sam -o jsonpath='{.status.certificate}'| base64 -d > sam.crt**
   2. A screen shot of a computer

      Description automatically generated
5. **Create** a developer **role**, **kubectl create role developer --verb=create --verb=get --verb=list --verb=update --verb=delete --resource=pods**
   1. **developer** = name of role we want to create. It could be **admin**, **operator**, **support**, etc
      1. **--verb** = What action the role could use
      2. **-- resource** = What resources the role could access. eg, **deployment**, **replicas**, **nodes**, etc
   2. Create a role with necessary permissions/privilege as per the question
6. **Bind** role created (**developer**) to user (**sam**) created earlier
   1. **kubectl create rolebinding developer-binding-sam --role=developer --user=sam**
   2. A screenshot of a computer screen

      Description automatically generated
7. Add user created into the **kubeconfig** file, **kubectl config set-credentials sam --client-key=/root/sam.key --client-certificate=/root/sam.crt --embed-certs=true**
   1. Use key and certificate created for set sam’s credential under K8s cluster
8. **Add** user created (**sam**) to user **context**, **kubectl config set-context sam --cluster=kubernetes --user=sam**
9. View user context, **kubectl config get-contexts**
   1. A screen shot of a computer

      Description automatically generated

\_\_\_

1. We can **switch user context** to **sam** if needed, **kubectl config use-context sam**
   1. **Asterisk (\*)** on the **current column** indicates user context the operator operate as
   2. A screenshot of a computer

      Description automatically generated