Kubernetes LAB

How to create an admin user in Kubernetes and assign permissions to create, get, list an delete pods throughout the cluster.

Steps to Add a New User to Your Kubernetes Cluster (Belonging to the admin Group):

1. Generate a Private Key for the User

First, generate a private key for the new user.

openssl genrsa -out maiky.key 2048

2. Create a CSR (Certificate Signing Request)

Next, create a Certificate Signing Request (CSR) that includes the admin group.

openssl req -new -key maiky.key -out maiky.csr -subj "/CN=maiky/O=admin" -out maiky.csr

- **/CN=maiky:** This is the name of the user being created.
- **/O=admin:** The group the user belongs to, which in this case is admin.

3. Sign the Certificate with the Cluster's CA

Use the cluster's **CA** (Certificate Authority) to sign the CSR and generate the user's certificate.

openssl x509 -req -in maiky.csr -CA /etc/kubernetes/pki/ca.crt -CAkey /etc/kubernetes/pki/ca.key

-out maiky.crt

4. Configure the New User in kubectl

Now, configure the user in kubectl. You can add this configuration to the ~/.kube/config file.

a) Set the credentials for the new user

kubectl config set-credentials maiky --client-certificate=/etc/kubernetes/pki/maiky.crt --client-key=/etc/kubernetes/pki/maiky.key

b) Create a context for the new user

kubectl config set-context maiky@kubernetes --cluster=kubernetes --user=maiky

5. Create a cluster role:

kubectl create clusterrole maiky-role –verb=list,get,create,delete –resource=pods

6. Create a cluster role binding:

kubectl create clusterrolebinding cluster-role-binding-maiky --clusterrole=maiky-role –user=maiky

7. Verify the User's Access:

Finally, verify that the new user has the correct permissions.

- Switch to the new context:

  # kubectl config use-context maiky@kubernetes

- Try a command to test access, such as listing the nodes:

  # kubectl get pods