**TLS based authentication:**

[ generate key using 2049 algorithm]

openssl genrsa -out myuser.key 2048

create cert(CSR) request and assign to user alice and out to a file

openssl req -new -key myuser.key -subj "/CN=alice" -out myuser\_alice.csr

send the CSR for signing encode using base64

cat myuser\_alice.csr | base64

Create csr.yaml and apply it

| **apiVersion**: certificates.k8s.io/v1  **kind**: CertificateSigningRequest  **metadata**:  **name**: csr-for-alice  **spec**:  **groups**:  **- system**:authenticated  **usages**:  - digital signature  - key encipherment  - client auth  **signerName**: kubernetes.io/kube-apiserver-client  **request**: <encoded CSR> |
| --- |

**Authorization**:

**Roles and Role Bindings**

| **apiVersion**: rbac.authorization.k8s.io/v1  **kind**: Role  **metadata**:  **name**: developer  **rules**:  **- apiGroups**: [""] # "" indicates the core API group  **resources**: ["pods"]  **verbs**: ["get", "list", "update", "delete", "create"]  **- apiGroups**: [""]  **resources**: ["ConfigMap"]  **verbs**: ["create"]  ---  **apiVersion**: rbac.authorization.k8s.io/v1  **kind**: RoleBinding  **metadata**:  **name**: alice-developer-binding  **subjects**:  **- kind**: User  **name**: alice # "name" is case sensitive  **apiGroup**: rbac.authorization.k8s.io  **roleRef**:  **kind**: Role  **name**: developer  **apiGroup**: rbac.authorization.k8s.io |
| --- |

Validate the authorization:

kubectl auth can-i delete pods --as alice

**ClusterRole and ClusterRoleBinding**

| **apiVersion**: rbac.authorization.k8s.io/v1  **kind**: ClusterRole  **metadata**:  **name**: cluster-administrator  **rules**:  **- apiGroups**: [""] # "" indicates the core API group  **resources**: ["nodes"]  **verbs**: ["get", "list", "delete", "create"]  ---  **apiVersion**: rbac.authorization.k8s.io/v1  **kind**: ClusterRoleBinding  **metadata**:  **name**: alice-cluster-admin-rolebinding  **subjects**:  **- kind**: User  **name**: alice  **apiGroup**: rbac.authorization.k8s.io  **roleRef**:  **kind**: ClusterRole  **name**: cluster-administrator  **apiGroup**: rbac.authorization.k8s.io |
| --- |

Validate the authorization:

kubectl auth can-i delete nodes --as alice

**Work With Private Repository**

kubectl create secret docker-registry secret-for-dockerhub --docker-server docker.io --docker-username simplilearn --docker-password mypassword

| **root@ip-172-31-19-184**:~# cat pod-privateimage.yaml  **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: pod-with-private-image  **spec**:  **containers**:  **- name**: some-provate-image  **image**: private-repository/imagename:tag  **imagePullSecrets**:  **- name**: secret-for-dockerhub |
| --- |

**SecurityContext**

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: web-pod  **spec**:  **containers**:  **- name**: ubuntu  **image**: ubuntu  **command**: ["sleep", "3600"]  **securityContext**:  **capabilities**:  **add**: ["MAC\_ADMIN"]  **securityContext**:  **runAsUser**: 1001 |
| --- |

Commands Dump

| history  kubectl create user  kubectl api-resources | grep -i user  vim basic\_users.csv  curl https://localhost:6443  curl https://localhost:6443 -k  ll  kubectl get pods -A  kubectl get pod -n kube-system kube-apiserver-ip-172-31-19-184 -o yaml  kubectl get pod -n kube-system kube-scheduler-ip-172-31-19-184  kubectl get pod -n kube-system kube-scheduler-ip-172-31-19-184 -o yaml  vim /etc/kubernetes/scheduler.conf  cd /etc/kubernetes/pki  ll  openssl x509 -in ca.crt -text -noout  ll  openssl x509 -in apiserver-etcd-client.crt -text -noout  ll  kubectl get pods  cd  cd .kube/  ll  vim config  openssl genrsa -out myuser.key 2048  cp myuser.key ~  cd  ll  openssl req -new -key myuser.key -subj "/CN=alice" -out myuser\_alice.csr  ll  cat myuser\_alice.csr | base64  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl explain CertificateSigningRequest  kubectl explain CertificateSigningRequest.spec  vim csr.yaml  kubectl apply -f csr.yaml  vim csr.yaml  kubectl apply -f csr.yaml  kubectl describe certificatesigningrequests.certificates.k8s.io csr-for-alice  kubectl certificate approve csr-for-alice  kubectl describe certificatesigningrequests.certificates.k8s.io csr-for-alice  kubectl get certificatesigningrequests.certificates.k8s.io csr-for-alice -o yaml  kubectl get certificatesigningrequests.certificates.k8s.io csr-for-alice -o yaml | grep certificate  kubectl delete csr csr-for-alice  vim csr.yaml  kubectl apply -f csr.yaml  kubectl certificate approve csr-for-alice  kubectl get certificatesigningrequests.certificates.k8s.io csr-for-alice -o yaml | grep certificate  kubectl get certificatesigningrequests.certificates.k8s.io csr-for-alice -o yaml  cat  | base64 --decode  echo  | base64 --decode  vim alice.crt  ll  curl https://localhost:6443 --key myuser.key --cert alice.crt  curl https://localhost:6443 --key myuser.key --cert alice.crt -k  curl https://localhost:6443 --key myuser.key --cert alice.crt -k --cacert /etc/kubernetes/pki/ca.crt  curl https://localhost:6443/api/v1/pods --key myuser.key --cert alice.crt -k --cacert /etc/kubernetes/pki/ca.crt  openssl x509 -in alice.crt -text -noout  vim alice.crt  vim csr.yaml  echo  | base64 --decode  cat csr.yaml  vim .kube/config  kubectl config  kubectl config current-context  kubectl config set-cluster -h  kubectl config set-cluster client1-cluster --server=https://client1.example.com:6443  kubectl config set-credentials -h  kubectl config set-credentials alice --client-certificate alice.crt --client-key myuser.key  vim .kube/config  kubectl config set-context -h  kubectl config set-context client1-context --cluster client1-cluster --user alice  vim .kube/config  kubectl config get-contexts  kubectl config use-context client1-context  kubectl config get-contexts  kubectl get pods  kubectl config set-context alice@kubernetes --cluster kubernetes --user alice  kubectl config use-context alice@kubernetes  kubectl config get-users  kubectl get pods  kubectl config get-contexts  kubectl config use-context kubernetes-admin@kubernetes  kubectl get pods  kubectl proxy  mkdir RBAC  cd RBAC  vim role.yaml  kubectl apply -f role.yaml  kubectl get role  kubectl describe role developer  vim rolebinding.yaml  kubectl get rolebindings.rbac.authorization.k8s.io  kubectl apply -f rolebinding.yaml  kubectl get rolebindings.rbac.authorization.k8s.io  kubectl describe rolebindings.rbac.authorization.k8s.io  kubectl auth  kubectl auth can-i create pods  kubectl auth can-i delete nodes  kubectl auth can-i create pods --as alice  kubectl auth can-i delete pods --as alice  kubectl auth can-i delete nodes --as alice  car role.yaml rolebinding.yaml  cat role.yaml rolebinding.yaml  kubectl config use-context alice@kubernetes  kubectl get pods  kubectl config use-context kubernetes-admin@kubernetes  cat ../csr.yaml  kubectl config use-context alice@kubernetes  kubectl get pods  kubectl get nodes  kubectl get cm  curl https://localhost:6443/api/v1/pods --key myuser.key --cert alice.crt -k  curl https://localhost:6443/api/v1/pods --key myuser.key --cert alice.crt -k --cacert /etc/kubernetes/pki/ca.crt  cd ..  curl https://localhost:6443/api/v1/pods --key myuser.key --cert alice.crt -k --cacert /etc/kubernetes/pki/ca.crt  curl https://localhost:6443/api/v1/pods --key myuser.key --cert alice.crt -k --cacert /etc/kubernetes/pki/ca.crt --user alice  cd RBAC  kubectl api-resources  vim clusterRole.yaml  kubectl apply -f clusterRole.yaml  kubectl get-con  kubectl config get-contexts  kubectl config use-context kubernetes-admin@kubernetes  kubectl apply -f clusterRole.yaml  kubectl auth can-i delete nodes --as alice  kubectl config use-context alice@kubernetes  kubect get nodes  kubectl get nodes  kubectl config use-context kubernetes-admin@kubernetes  cat clusterRole.yaml  cd ..  vim pod-privateimage.yaml  kubectl create secret docker-registry secret-for-dockerhub --docker-server docker.io --docker-username simplilear --docker-password mypassword  cat pod-privateimage.yaml  kubectl proxy  vim pod-securityContext.yaml  kubectl apply -f pod-securityContext.yaml  kubectl describe pod web-pod  kubectl exec web-pod -- id  kubectl logs web-pod  kubectl delete pod web-pod  vim pod-securityContext.yaml  kubectl apply -f pod-securityContext.yaml  kubectl exec web-pod -- id  vim pod-securityContext.yaml  kubectl delete pod web-pod  cat pod-securityContext.yaml  kubectl config get-contexts  vim network-policy.yaml  kubectl apply -f network-policy.yaml  kubectl exec -it backend-pod -- bash  kubectl get nodes |
| --- |