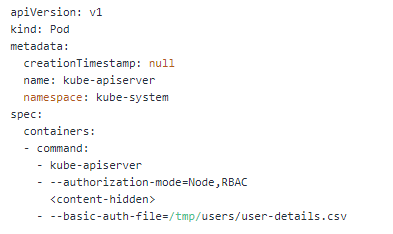
Security – 12 %

With Files

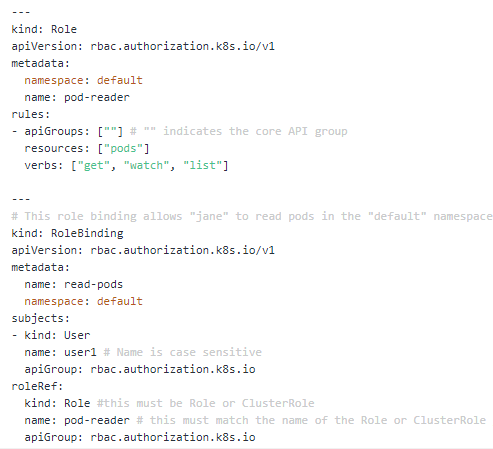
* Create a file with user details locally at /tmp/users/user-details.csv

1. password123,user1,u0001
2. password123,user2,u0002

* Modify the kube-apiserver startup options to include the basic-auth file /etc/kubernetes/manifests/kube-apiserver.yaml



* Create the necessary roles and role bindings for these users:



curl -v -k https://localhost:6443/api/v1/pods -u "user1:password123"

* Generating CA key & Certificate

# Create private key for CA

**openssl genrsa -out ca.key 2048**

# Create CSR using the private key

**openssl req -new -key ca.key -subj "/CN=KUBERNETES-CA" -out ca.csr**

# Self sign the csr using its own private key

**openssl x509 -req -in ca.csr -signkey ca.key -CAcreateserial -out ca.crt -days 1000**

* Generating Admin key & Certificate

# Geenrate private key for admin user

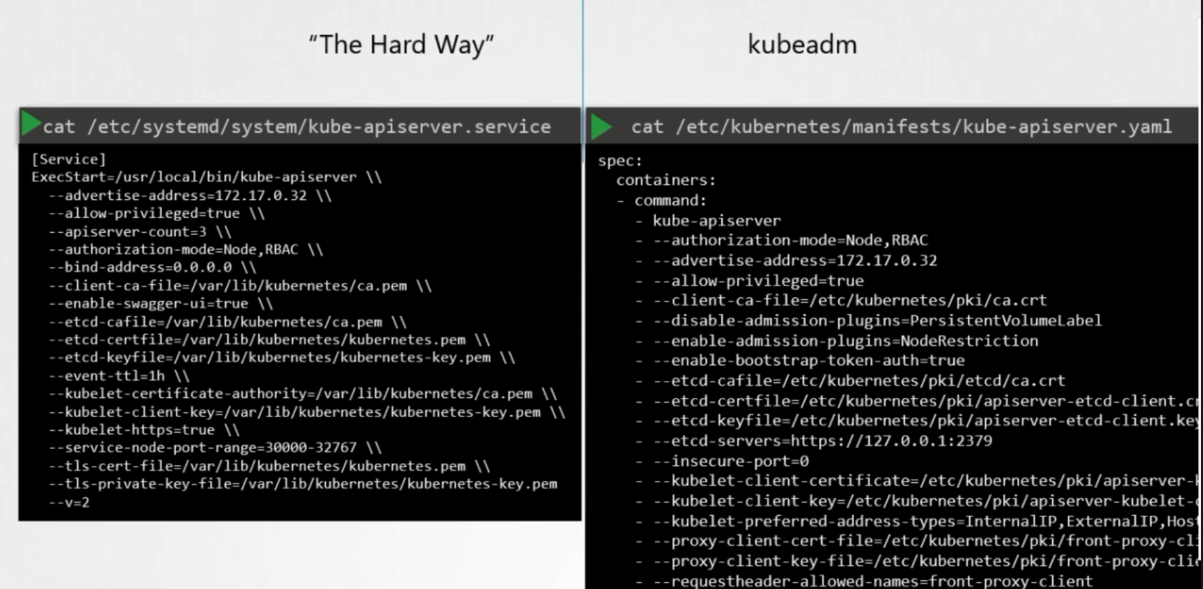
**openssl genrsa -out admin.key 2048**

# Generate CSR for admin user. Note the OU.

**openssl req -new -key admin.key -subj "/CN=admin/O=system:masters" -out admin.csr**

# Sign certificate for admin user using CA servers private key

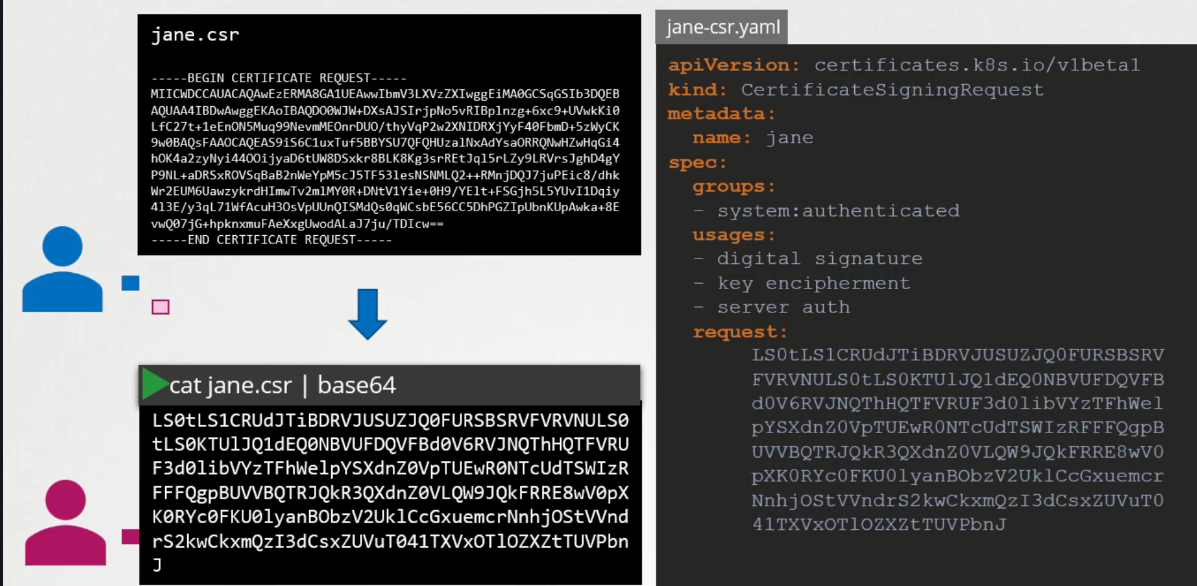
**openssl x509 -req -in admin.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out admin.crt -days 1000**

* Viewing certificate

**Openssl x509 –in /file-path.crt –text –noout**

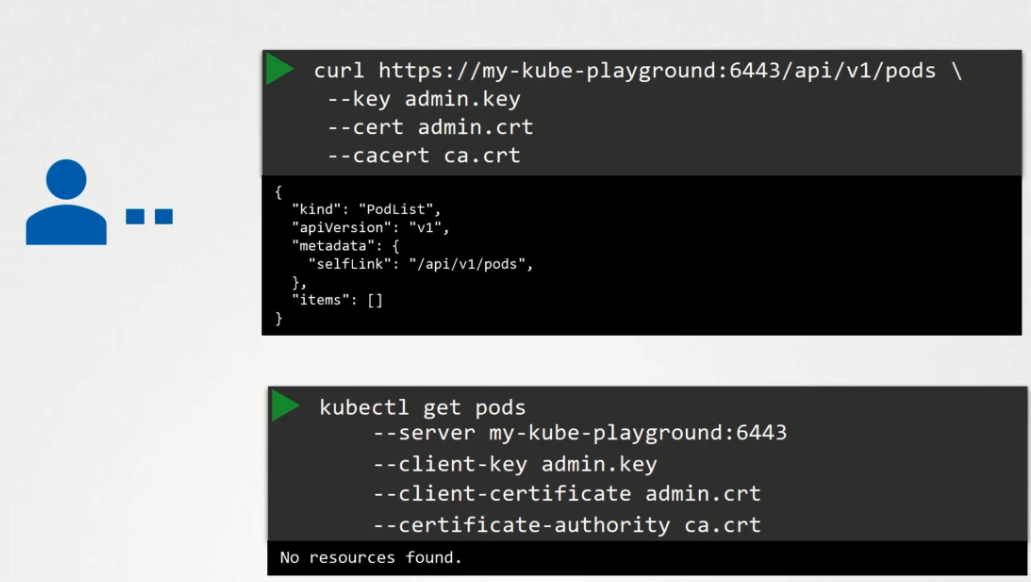
New user – Flow

1. Create the key : **openssl genrsa –out karthik.key 2049**
2. Create CsigningReq to Admin: **openssl req –new –key karthik.key –subj “/CN=karthik” –out karthik.csr**
3. Admin takes the key and create the certificate Signing Object **cat karthik.csr | base64**



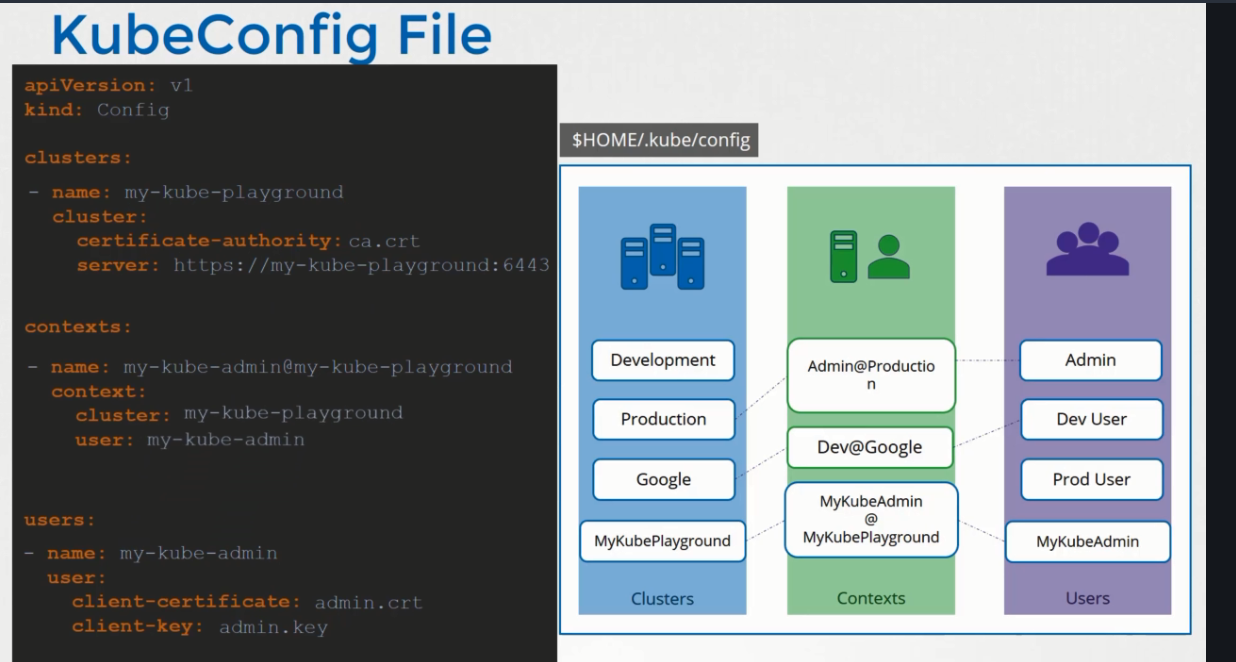
1. **Kubectl get csr**
2. **Kubectl certificate approve karthik**

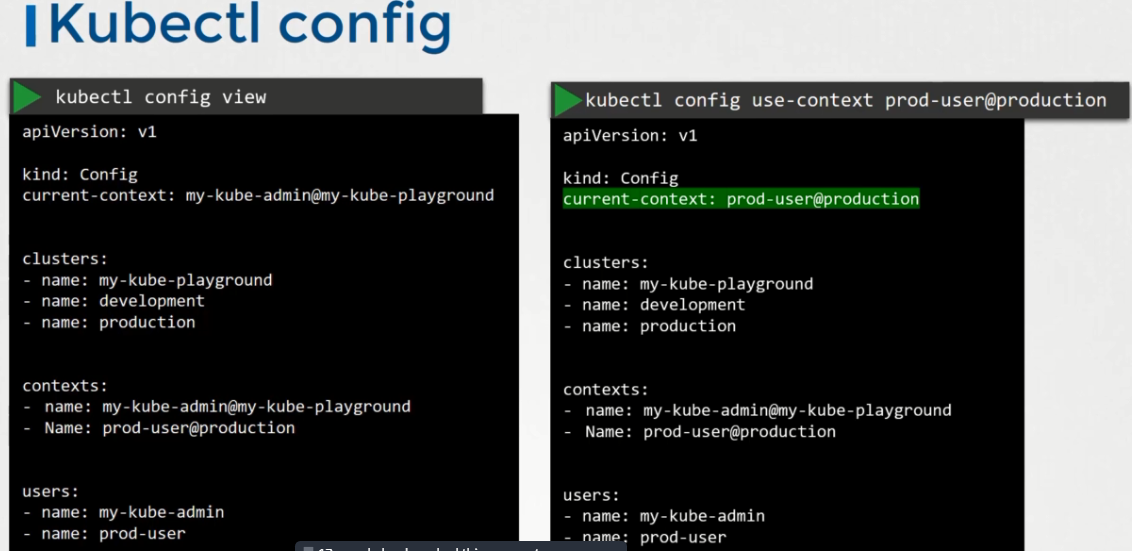
* **KubeConfig**

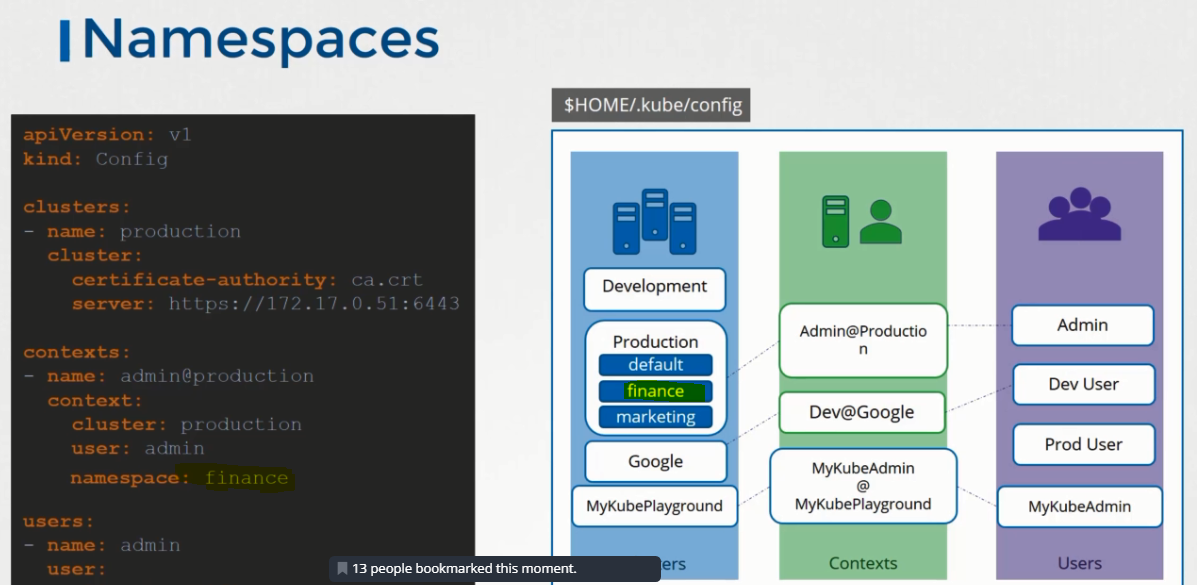
 Every time we cannot type the cert details in the command

Config file has 3 sections. 1. Cluster 2. Contexts 3.Users

Context is bridge the cluster and Users. Providing access to which user has to use which cluster





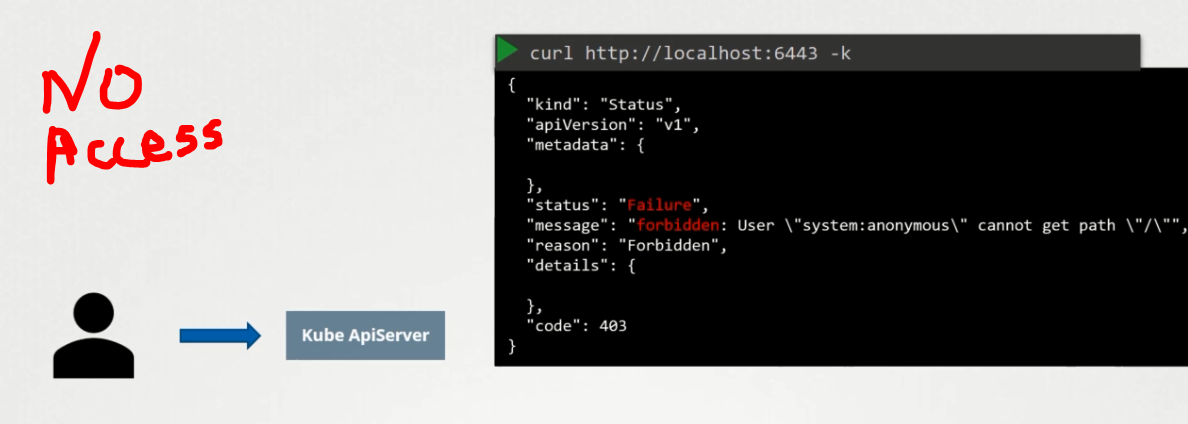


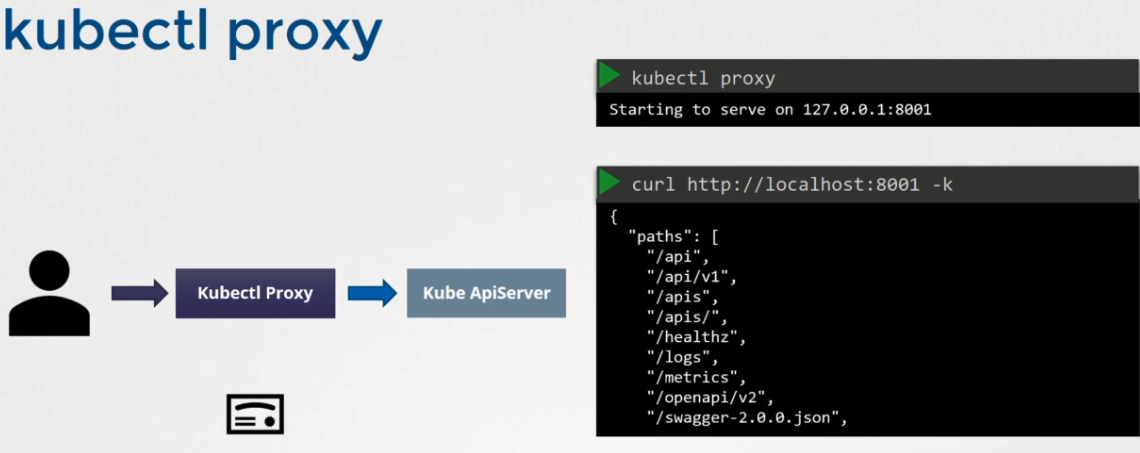
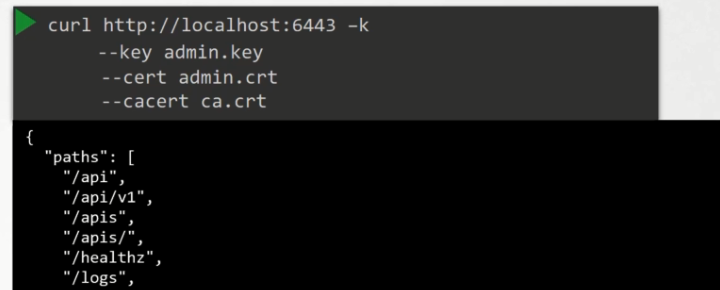


// different kubeconfig file

**kubectl config --kubeconfig=/root/my-kube-config use-context research**

* **Kubectl Proxy**

 bec certificate required



Kubectl – http proxy to access the kube api server; kube proxy – responsible to connect all the pods/services





Commands:

Kubectl get roles

Kubectl get rolebindings

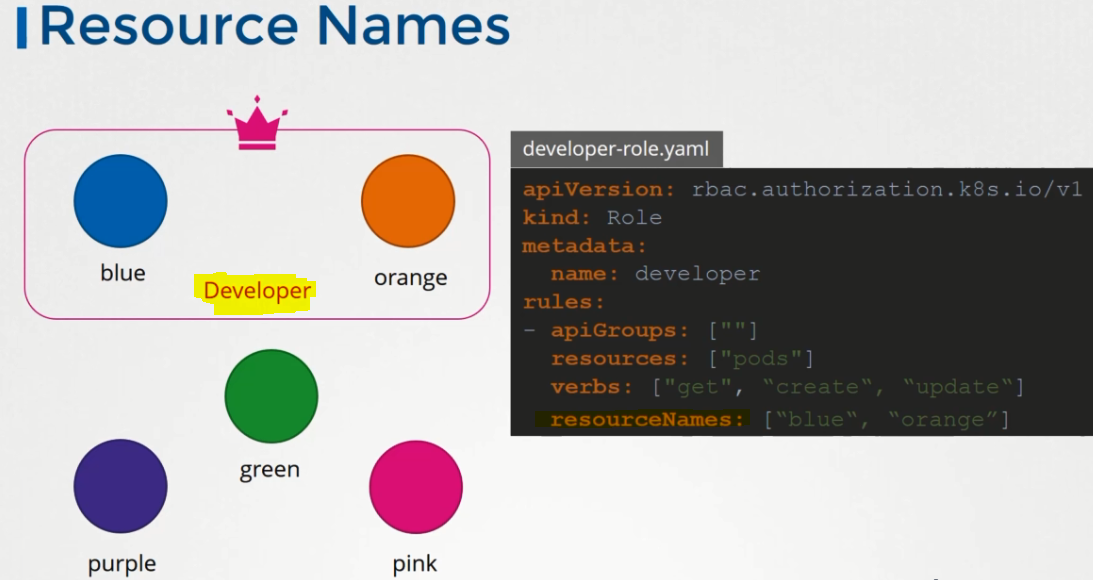
Kubectl describe role developer

Kubectl describe rolebinding devuser-developer-binding

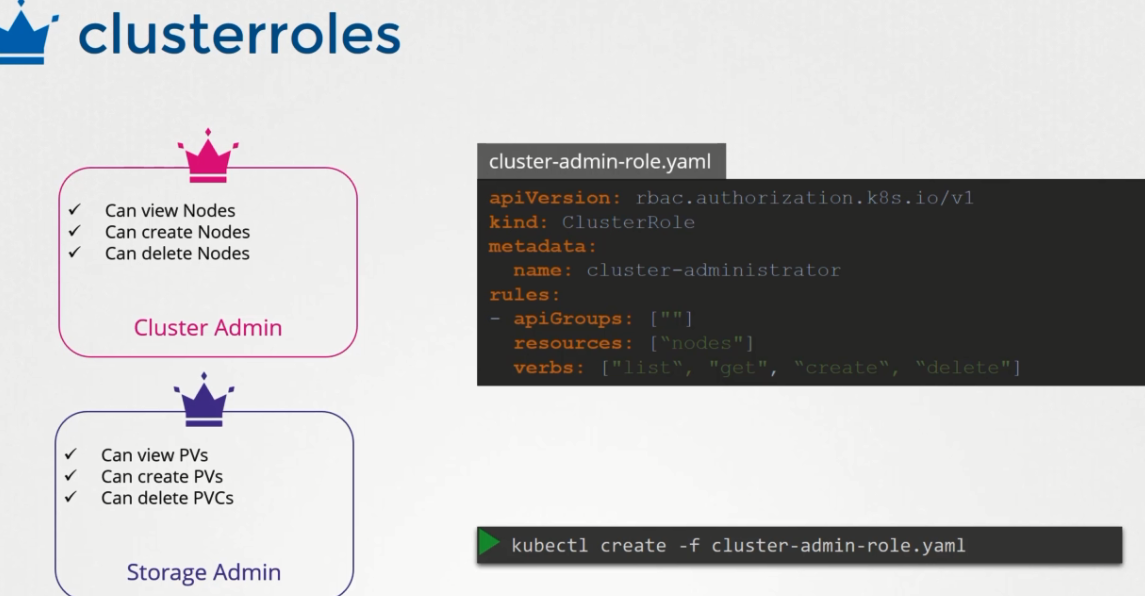
Check Access:

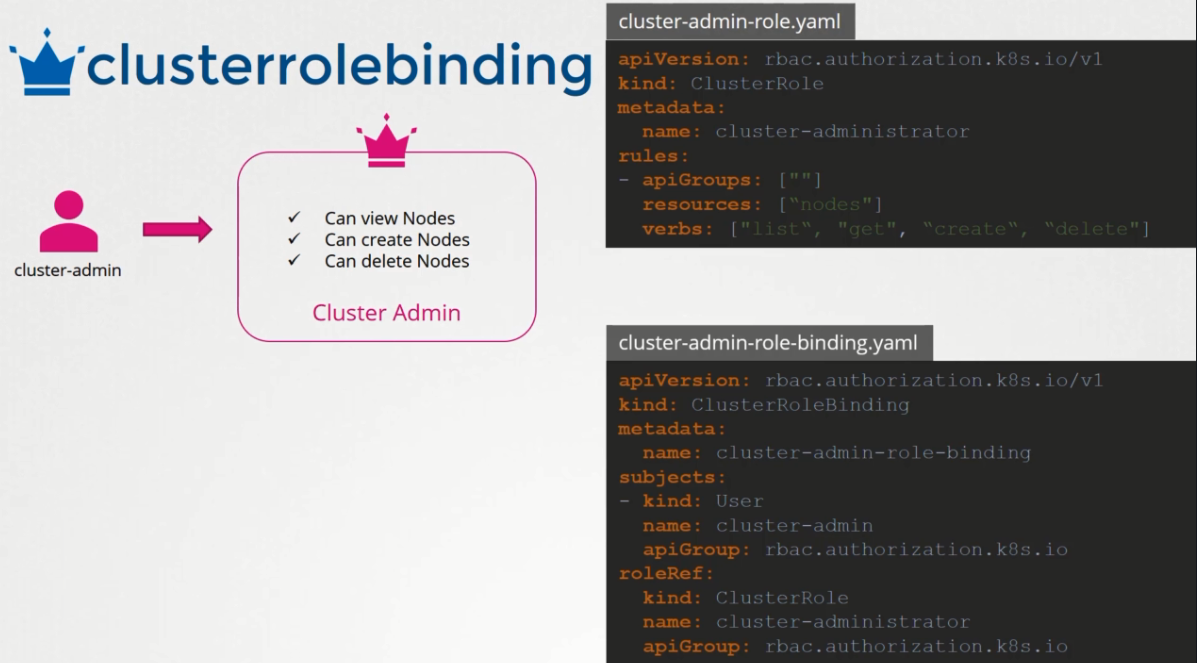
Kubectl auth can-i create pods

Kubectl auth can-I create deployments –as dev-user –namespace test

Pod-level restrictions 

Cluster-role and rolebinding





**Security context**

