USER CREATION AND GIVING ACCESS TO PARTICULAR NAME SPACE

1.kubectl create ns rasa

2.kubectl get ns

3. Generating private key for akash (john.key)

$ openssl genrsa -out akash.key 2048

4.Generating certificate signing request (john.csr)

$ openssl req -new -key john.key -out john.csr -subj "/CN=john/O=finance"

openssl req -new -key john.key -out john.csr -subj "/CN=akash"

5.Copy kubernetes ca certificate and key with root user

$ cp /etc/kubernetes/pki/ca.{crt,key} /home/engteam

6.Sign the certificate using certificate authority

$ openssl x509 -req -in akash.csr -CA ca.crt –CAkey ca.key -CAcreateserial -out akash.crt -days 365

7.ca certificate , john certificate , john key we need

8. kubectl --kubeconfig akash.kubeconfig config set-cluster kubernetes --server https://10.100.102.31:8443 --certificate-authority=ca.crt

We can see details by kubectl config view command

9. kubectl --kubeconfig akash.kubeconfig set-credentials akash --client-certificate (directory) --client-key (directory)

Now we are setting context

10 kubectl --kubeconfig akash.kubeconfig set-context akash-kubernetes -- cluster kubernets --namespace rasa - - user akash

Edit vi john.kubeconfig

Set context john-kubernetes

Cp john.kubeconfig ~/.kube/config

(or else)

We can copy the config file edit as user and give the certificate and key

Decode the key as cat akash.crt | base64 -w0 cat akash.key | base64 -w0

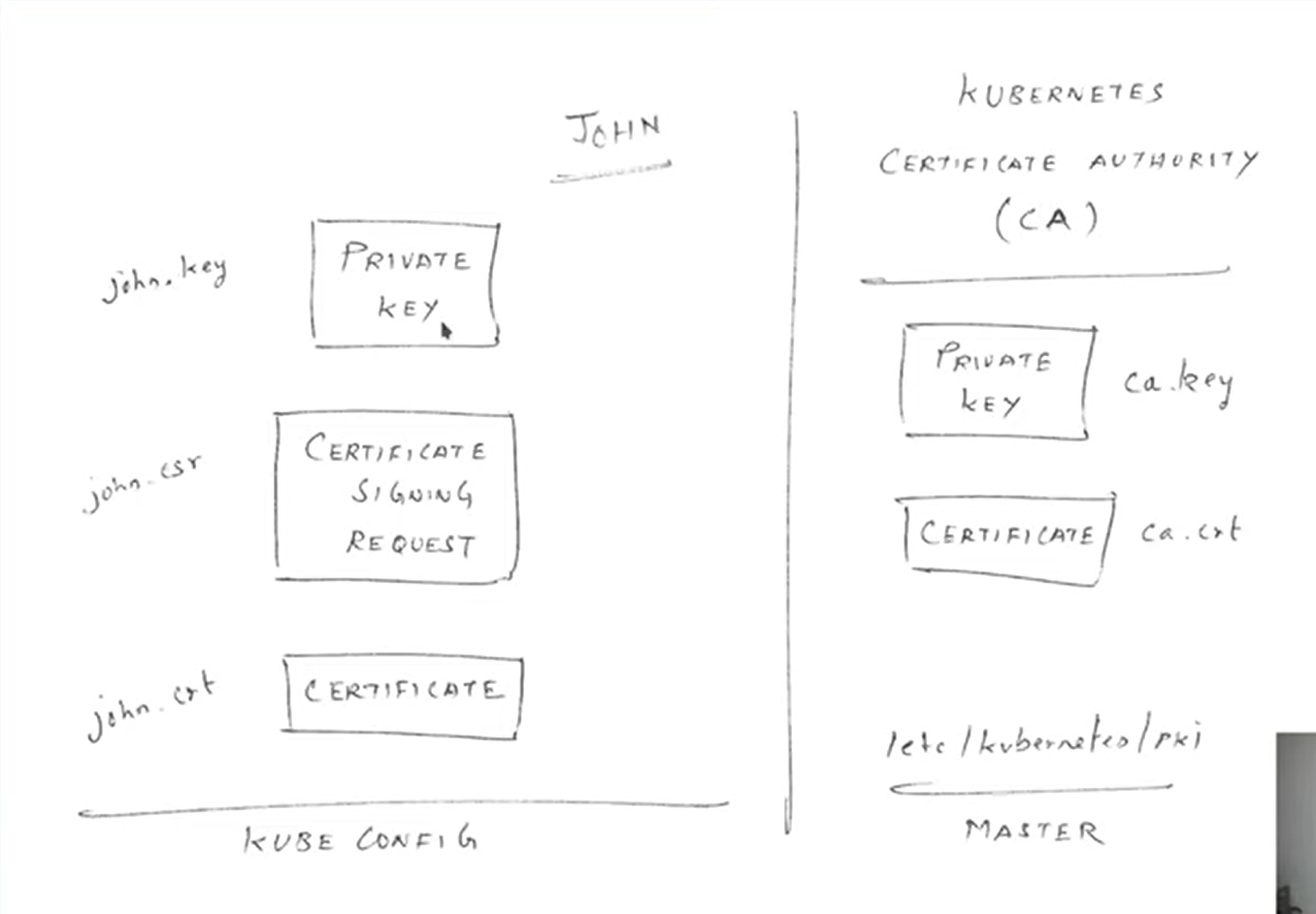


Now we are going to create a role

Kubectl create role akash-rasa --verb=get,list,delete --resource=pods --namespace rasa

Now we can do a role-binding with john

kubectl create rolebinding akash-rasa-rolebinding -- role= akash-rasa - - user=akash - - namespace rasa



mkdir akash & cd akash

add all certificate & key

kubectl config set-credentials akash --client-certificate (directory) --client-key (directory)

kubectl config set-context akash-kubernetes -- cluster kubernets --namespace rasa - - user akash