K8s
Session - 16
IAM

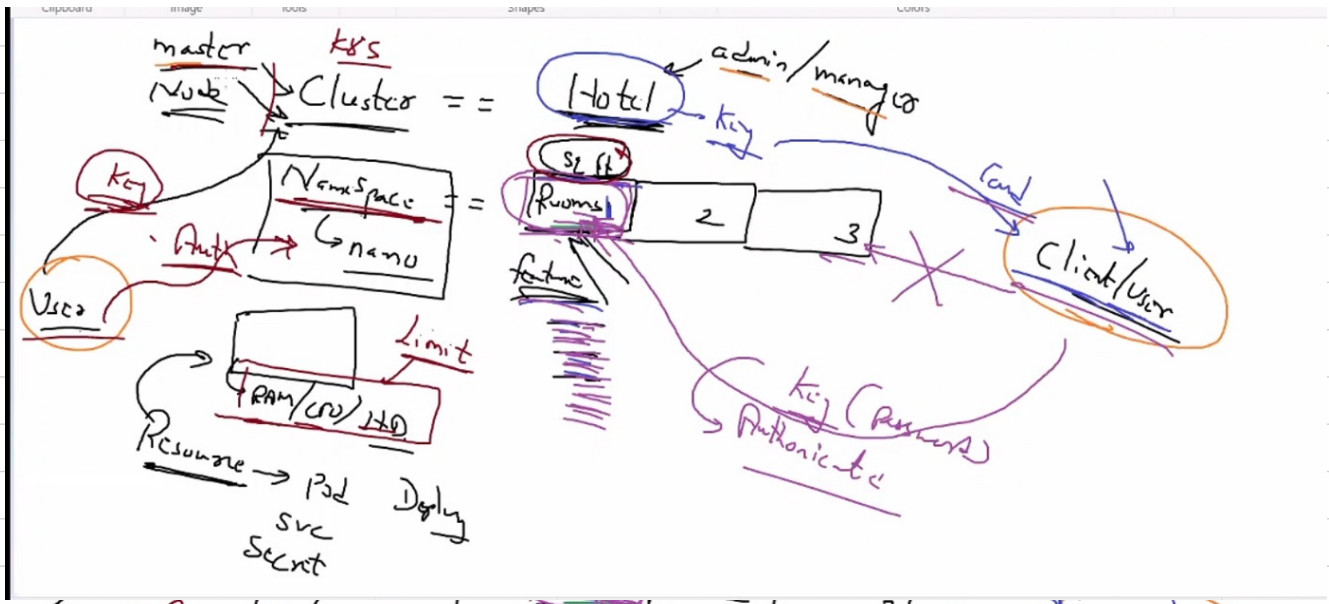
IAM (Identity and Access Management)

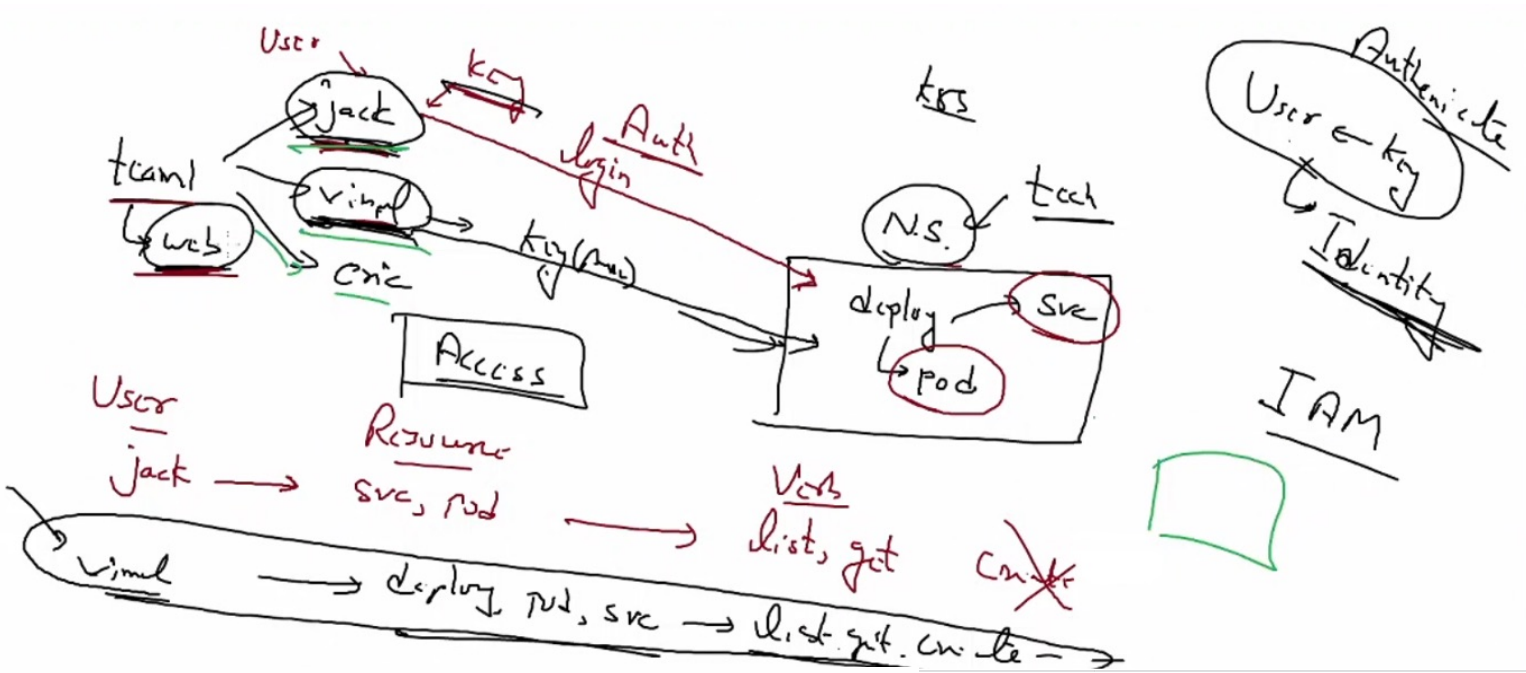
↳ RBAC (Role based Access Control)
Roles/user/Role-binding)

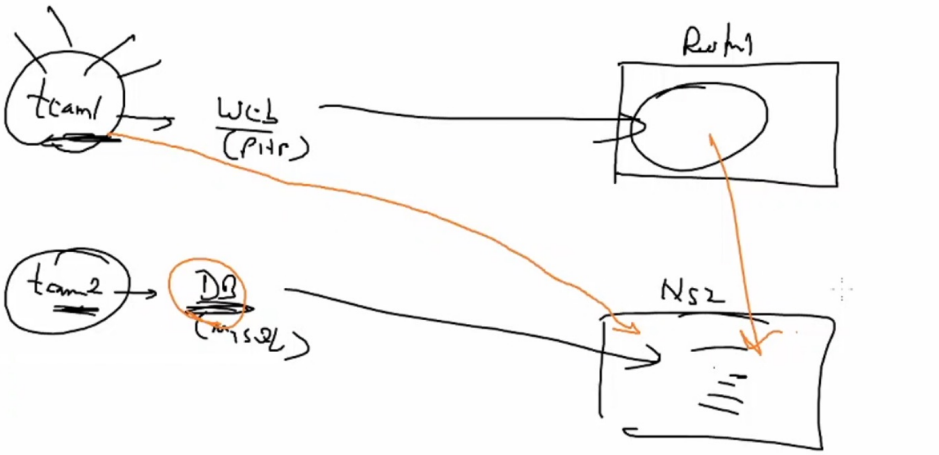
→ Since this topic is Related to Security so it will be applicable to both the cluster Single/Multi-node cluster

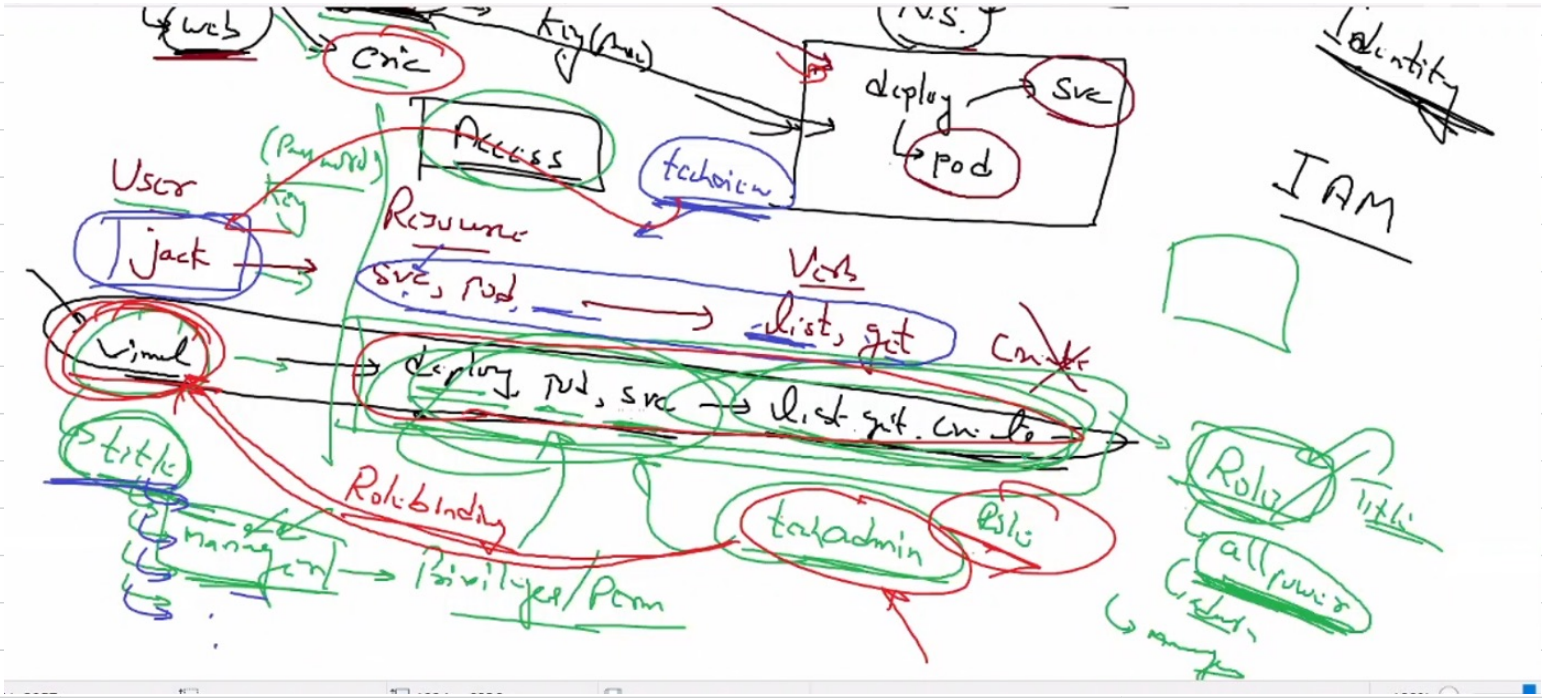
→ So If you want to implement Multi-Tenancy (tenant) means your client/user

Now watch Video till 33:28 min, not much Imp Concept Related to IAM

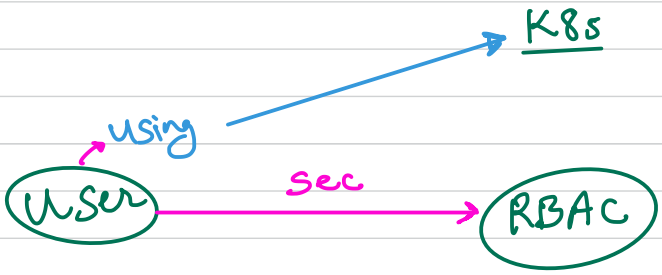



User
jack
vimal
team1
web
cmc
key
Auth
login
k8s
N.S.
tech
deploy
svc
pod
Access
User
jack
Resource
svc, pod
Verb
list, get
vimal
deploy, pod, svc
list get. create
Authenticate
User ← key
Identity
IAM


team1
Web
(Pilr)
Role1
team2
DB
(MySQL)
NS2


web
cmc
key
N.S.
deploy
svc
pod
Access
Identity
IAM
User
jack
vimal
Resource
svc, pod
Verb
list, get
deploy, pod, svc
list get. create
title
RoleBinding
manage
Privileged/Perm
tekadmin
Role
allpower

After test



So In K8s user security is govern by concept of RBAC

In K8s we will create multiple user, multiple Role but what this user can do we have to associate that particular Role is called Role-binding

That is a Reason you can see the Keyword here

```
C:\Program Files\Kubernetes>kubectl create --help
Create a resource from a file or from stdin.

 JSON and YAML formats are accepted.

Examples:
  # Create a pod using the data in pod.json
  kubectl create -f ./pod.json

  # Create a pod based on the JSON passed into stdin
  cat pod.json | kubectl create -f -

  # Edit the data in registry.yaml in JSON then create the resource using the edited data
  kubectl create -f registry.yaml --edit -o json

Available Commands:
  clusterrole             Create a cluster role
  clusterrolebinding      Create a cluster role binding for a particular cluster role
  configmap               Create a config map from a local file, directory or literal value
  cronjob                 Create a cron job with the specified name
  deployment              Create a deployment with the specified name
  ingress                 Create an ingress with the specified name
  job                     Create a job with the specified name
  namespace               Create a namespace with the specified name
  poddisruptionbudget     Create a pod disruption budget with the specified name
  priorityclass           Create a priority class with the specified name
  quota                   Create a quota with the specified name
  role                    Create a role with single rule
  rolebinding             Create a role binding for a particular role or cluster role
  secret                  Create a secret using specified subcommand
  service                 Create a service using a specified subcommand
  serviceaccount          Create a service account with the specified name
```

we can see this keywords here

But if you notice here, you cant find keyword available here to create a user here, or to create a key here, no option from the Kubernetes perspective. Why?

→ Because the way K8s create user or do authentication is a very different way.

```
[root@master ~]# kubectl get nodes
NAME      STATUS     ROLES                  AGE    VERSION
master    Ready      control-plane,master   6d2h   v1.20.5
worker1   NotReady   <none>                 6d2h   v1.20.5
worker2   NotReady   <none>                 6d2h   v1.20.5
[root@master ~]#
[root@master ~]#
[root@master ~]# kubectl cluster-info
Kubernetes control plane is running at https://172.31.10.173:6443
KubeDNS is running at https://172.31.10.173:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
[root@master ~]#
```

The K8s main master who is controlling the Entire Kubernetes So there are multiple program there are multiple program who are working together And it makes a Entire K8s master

Api Server
Kube Controller
Scheduler
&c

↳ control-plane

→ So that System/node and group of this program is also known as Control Plane

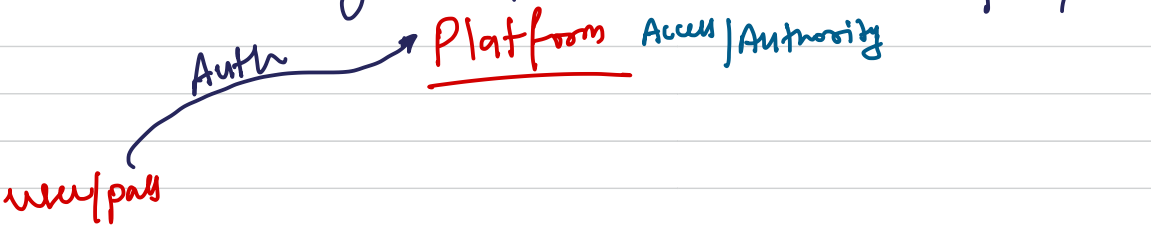
→ So that is a Reason may people used the term Control-plane instead of master.

So How to authenticate user

→ So technically If you talk about any cluster/platform platform may be any Cluster, cloud, Web-Application

→ If anybody want to use the platform, and If your platform is secure in the terms of, nobody can access the platform without identify this guys

→ In this Kind of Security they should have User name password, after they will Check and authenticate after that only, they will allow you to Enter into the platform

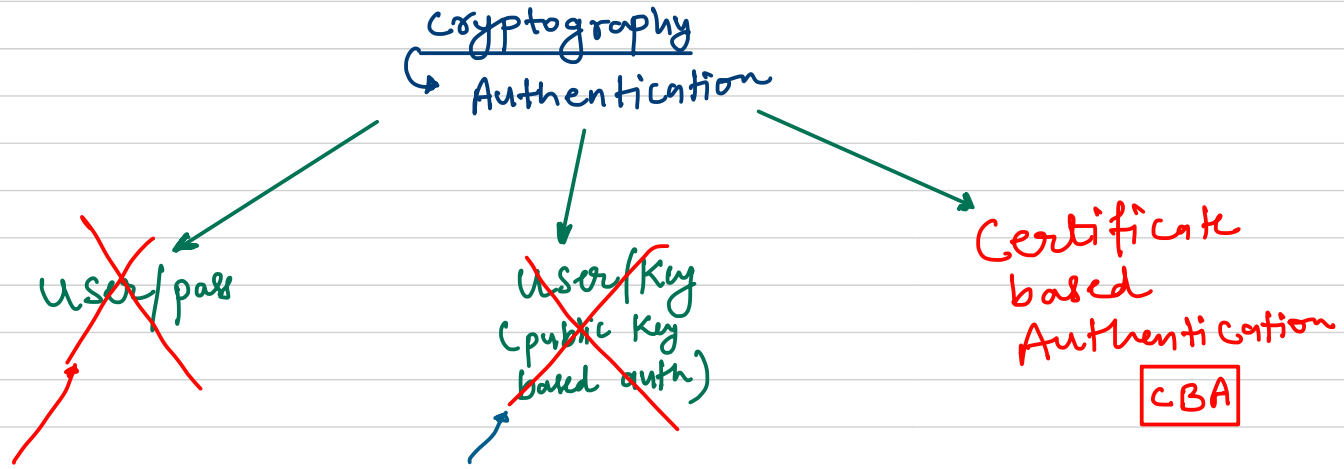


After Authentication what accss/authority they have is all decided obiviously this program

- So Here in K8s we can maintane the access by Role and Rolebiy but in K8s platfoom we dont have option to create user/password

So we dont have any option to manage the authentication
Why? So How it work

So If you Know about Cryptography



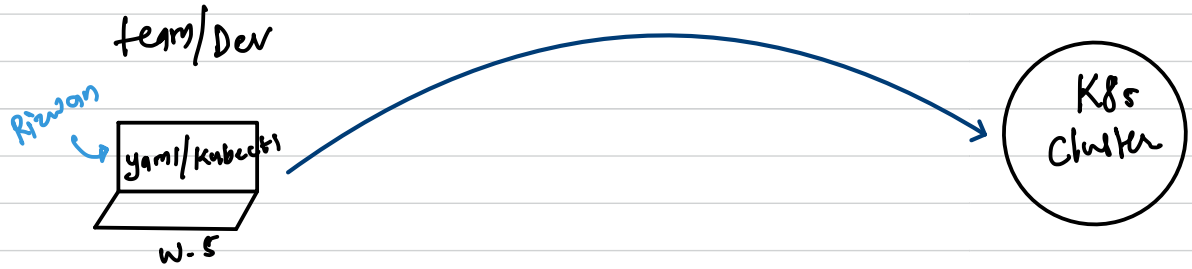
→ It is not good, It is good but it is very harder to manage this is like a ssh-keygen concept

→ So this CBA is a very great way/secure way in the terms of mangment of your password user identity and Keys

→ By the way If you talk about K8s, they Support authentication without authentication I wont allow Anybody Can use my Resources, Accss is a seperate part

→ typically K8s Support/Suggest CBA

So technically what kubernetes says that I am a cluster or one kind of app or maybe platform , running in the cloud or virtual machine, and You Are some different team or developer sitting on your laptop and you want to use my cluster or resources



From your laptop you will plan to write some yaml code or use kubectl command to deploy the things and use my resources

But point, here is whenever you use kubectl create command to the pod to create the resources , to create deployment or to run this YAML code so what you want as a developer let's say you are sitting on your system, you are workstation, and you write some YAML file and what I want. Finally when I run this YAML code by KUBECTL apply or create Behind the scene, my KUBECTL connect to k8s cluster and whatever we have written in this YAML code they should execute that means launch the resources/kind which we have requested for.

→ Before we do anything my K8s I am Secure System, I want you have to first authenticate me and technically K8s is not saying to us they are saying this to the program that we are using in Laptop (windows) means to Kubectl

And O.S never put the login name and password, what does it means?

and here as this Application known it is being used by human being, so they can type
So password based authentication is always create for human being

So that is a Reason this password based authentication is not at all useful for K8s.

Because here I (Rizwan) is not going on my behalf Kubectl command is going to K8s cluster.

So If you want you program will go and provide the password means Authenticate, we cant give login and password But because your program dont have the capability to tick here and type the password So,

Instead of password we will provide a Key, Key is also like a password but is big Key

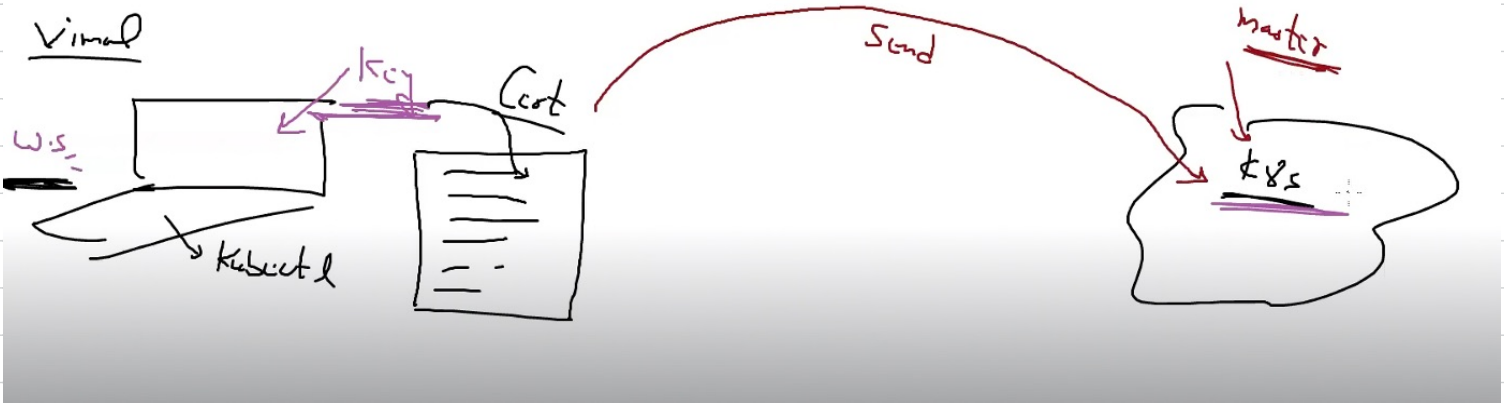
So we used Key based authentication,

So In Keybased authentication also we have two types

1) public Key based
2) certificate based Auth

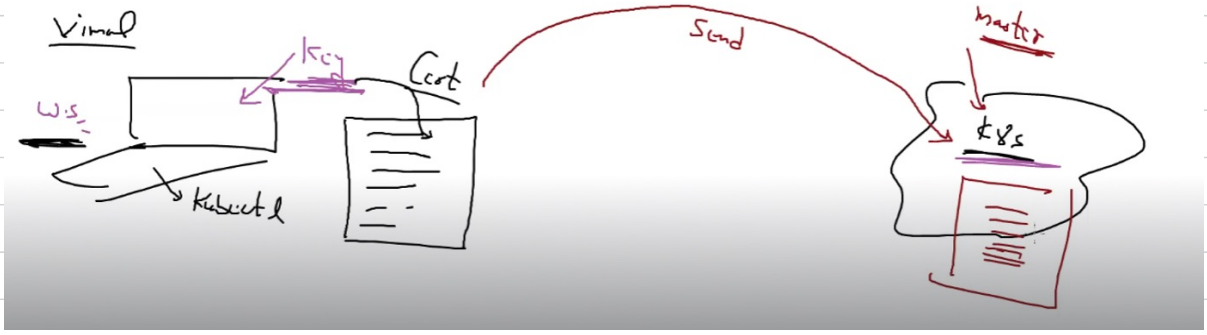
Certificate based Authentication

①



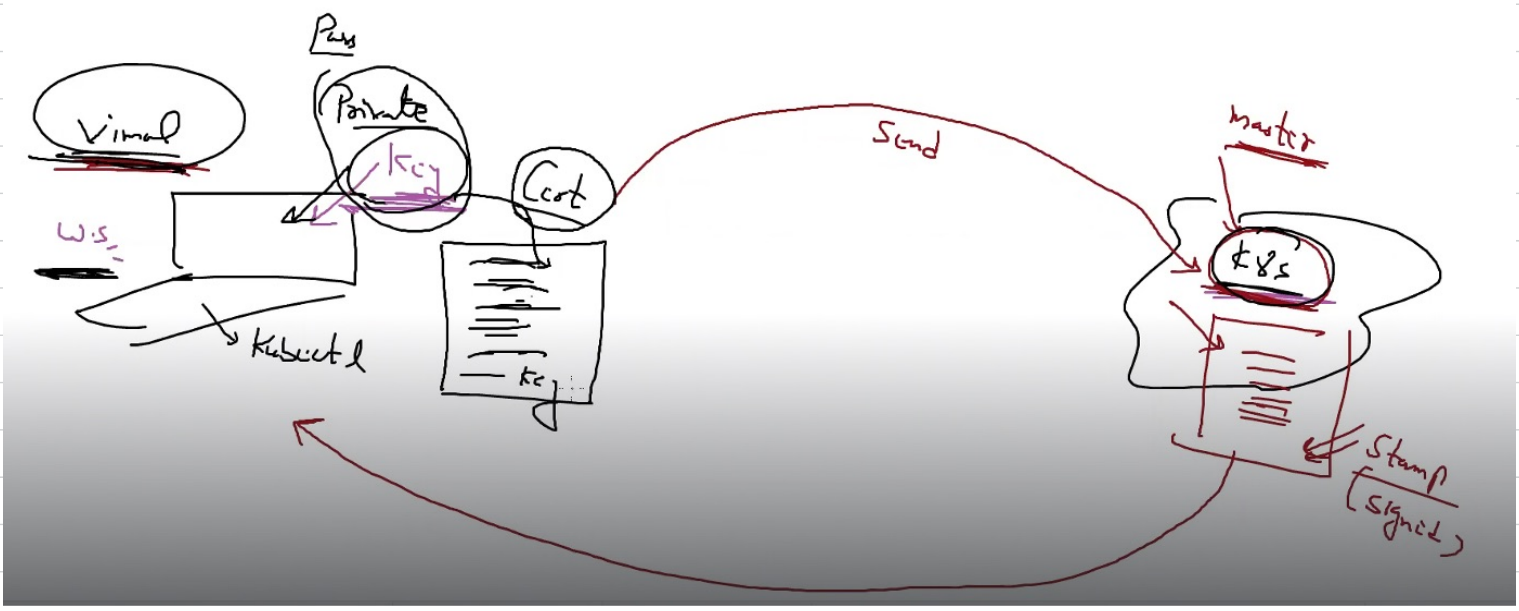
* So we have our own personal laptop in which kubectl program is install and we have K8s
* Now in my laptop, we will create our own personal Key, because there may be lots of developers and it will be very harder to ask again and again to the K8s master to create the password for every developer.

②

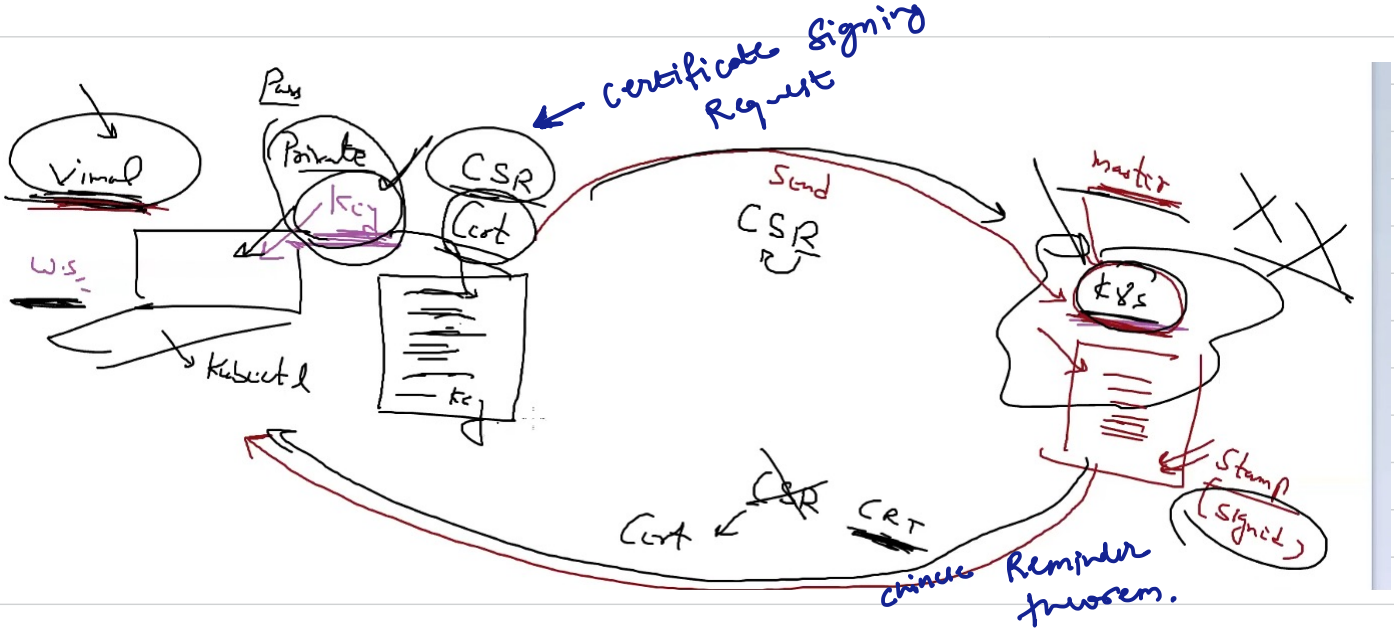


——> but after we create our key, we will do one more thing we will create one document and in this document, we will put some information like name, organisation, state, country, pin code, domain like some extra information and along with that, we will also put our key in this document. Technically this document is known as certificate

Now we will transfer this certificate into a master-node like we will mail them or we will transfer the certificate by SCP whatever authority we have.



So after receiving the certificate, the master node will open this certificate they will read the detail in the certificate and if they feel the details are correct and if they feel they can authenticate to the cluster finally, they put one stamp on the certificate and they sent back the same certificate to you like we can say signed certificate by the master



In the security world, the guy or we can say the system that have the authority to sign the certificate are known as certificate authority (CA)

So when you create kubernetes cluster your master node automatically becomes certificate authority.

```
[root@ip-172-31-91-29 ~]# cd /etc/kubernetes/pki/
[root@ip-172-31-91-29 pki]# pwd
/etc/kubernetes/pki
[root@ip-172-31-91-29 pki]# ls
apiserver.crt                etcd
apiserver-etcd-client.crt    front-proxy-ca.crt
apiserver-etcd-client.key    front-proxy-ca.key
apiserver.key                front-proxy-client.crt
apiserver-kubelet-client.crt front-proxy-client.key
apiserver-kubelet-client.key sa.key
ca.crt                       sa.pub
ca.key
[root@ip-172-31-91-29 pki]#
```

public key Infrastructure

Very Imp

And as we know, we never share our private key to the person we always share our public key. So whenever we create CSR, there are generally two types of key public and private key

This Concept is known as certificate based authentication

NOW WE WILL PROCEED WITH THE PRACTICAL PART

Step 1 To Create Key

```
[root@centos kubeaws]#
[root@centos kubeaws]# openssl genrsa 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
...........+++++
.......+++++
e is 65537 (0x010001)
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDE4C2mw16IiGIk0XIWvLIp4U3j+mxjM690Dsg3uVSSSHcw8+RQ
d18IJDNXmlex3U+UX4lL9uu+UxJeaccu1Js5b7uxSc1Y36HgDhx9EKJU7SoJW/R9
L3d40u/6fUL3j/rMSNh9D+NjrKocEQSJwwmKEq0npj9hwDh3Hvx/Inq0/wIDAQAB
AoGBAKJp0h9d0vHNmTGIe5cYSd5U2P33DE12SQr0WuXlhUg8cd+3hCWBpKtBuf1N
/YRk746WldxOMkLLoz6yAri2eeod0Dtj9tOFMZC1ubYoCsstjTx1m0YzgRPrrIzC
BQ1PMGR7V27zg9G1LrDePYumaFVQvCAQsyuEvjSObSSkMCl5AkEA/qupGnHk2PFK
JZbGGLKzT6lx0+UCGELh5QXB7YOIbbQ7xdQ08hRlaNoe5dj0Q+Ilr3fmX5Ap69B4
hb5pa4PMmwJBAMXnR/+KsUOq0GOwLxG6Yn2jZPixtElIKr1Ux++UyLcJsAjCJCnc
Ptw9sxlFE/SmNklu4Gvytw5SYxoCNvLJtW0CQQCKK86Kt4KHJZqp/DysR8A7l15X
radOTHNNE6uc1sFda+0GL1Uuhlxi3/zljakxb3As3GuUK55i/+9z+NQvbgnDAkBO
ymHTz3f7pMoKpl5XBImp05+jk7xfYKgDcPmY3Pd8SBMZqZGUSOeN79KaQPJM+M+s
NMSMtblDrnFeLXuK9duRAkBwec3lgt8Irr41Q+MRanss2WUwliemeegBUQKHYYyN
+OWZ7qZCyVOrrCn0+SK5iJ3SUc0gaJ+YBaqF7tc0nauX
-----END RSA PRIVATE KEY-----
[root@centos kubeaws]#
```

generate and rsa → is algo
1024 size.

Key Created

To save this key we have to give some Extra options along with this

```
[root@centos kubeaws]# openssl genrsa -out rizwan.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.....................+++++
.....................+++++
e is 65537 (0x010001)
[root@centos kubeaws]# cat rizwan.key
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDBQA3kNjoj0q8+GBtfWu2qrX9xXZSmP6Lh8knC1kQjOg/4uJS2
aKS/QUb9grtFDF86JsQzZzJ0Le/S0u99X2EfNPwAz4TKEDDRMTu+kjaPIeoXcObc
OiKdhOlo/qB3PkhuQRCuSQ66+G5+1faT716lliog7Ip1n7w+oi1KmRjd2QIDAQAB
AoGAO0i0OkNseJrkic8JbqpNaOPKyfNka8/v3xLeD/h9dLUWOz4c2QynJckXxyO4
4YjXGdbUr9RDxSJmhS2SrMp613eV5nYzldVhbE88ynRaR7GAXmfsA4tEnMq6CahZ
4Bbb1blmlz/qucUcY4bAJ++TTP6oVmTNpkIrd1tL5PUW5IECQQD6OzIPUDvmJ5DD
QLEYeNYd3Edw469w3HszERGlqtUso6sgCzlkVWotoxTYcqYF4+9XEdpvXqZPc5yD
dpPcRdrJAkEAxbSSCbANwDUo1uNIpQmFx8mDV8BOpx8k9sEZKqRqWh8lVjdmhiRx
joBHwwX22qeqV2x0HCtgSGfI23Q7Fg9ikQJBAIx4apO6IMYE8/kH+eKZAunkXJ58
0mHzA54ZOOTdT6otfsUQR/mREXfpEqynP5gS6qrgNNVMNuRnLDeIe1yuQjECQQCN
15Yq4yeKbdlGCry0w1c61kWoMG19k2emrTVMFbfDvrK/33Fr7juWfEyADAhmKYKZ
xFKtXyVt7R/66rH1isYxAkEAviL9w2ROVeOaJVDPCFz568fkmO/9ON99KMVcmm1H
I3wfrCHD65RO6yJK1l6C3ChAUlGAcrFJhidQLYinT4fL1g==
-----END RSA PRIVATE KEY-----
[root@centos kubeaws]#
```

saving

extension should be always .key

Key.

Step 27 Now we will Create CSR

```
[root@centos kubeaws]#
[root@centos kubeaws]# openssl req -new -key rizwan.key -out rizwan.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:india
string is too long, it needs to be no more than 2 bytes long
Country Name (2 letter code) [XX]:In
State or Province Name (full name) []:MH
Locality Name (eg, city) [Default City]:NGP
Organization Name (eg, company) [Default Company Ltd]:LW
Organizational Unit Name (eg, section) []:linux
Common Name (eg, your name or your server's hostname) []:rizwan
Email Address []:rizwan@pathan.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[root@centos kubeaws]#
```

Now we will copy this CSR so that master can sign so we can login.

```
[root@master ~]# cd /etc/kubernetes/pki/
[root@master pki]# pwd
/etc/kubernetes/pki
[root@master pki]#
[root@master pki]# vim rizwan.csr
[root@master pki]# ll
total 60
-rw-r--r-- 1 root root 1261 Mar  3 14:45 apiserver.crt
-rw-r--r-- 1 root root 1135 Mar  3 14:45 apiserver-etcd-client.crt
-rw------- 1 root root 1679 Mar  3 14:45 apiserver-etcd-client.key
-rw------- 1 root root 1679 Mar  3 14:45 apiserver.key
-rw-r--r-- 1 root root 1143 Mar  3 14:45 apiserver-kubelet-client.crt
-rw------- 1 root root 1675 Mar  3 14:45 apiserver-kubelet-client.key
-rw-r--r-- 1 root root 1066 Mar  3 14:45 ca.crt
-rw------- 1 root root 1675 Mar  3 14:45 ca.key
drwxr-xr-x 2 root root  162 Mar  3 14:45 etcd
-rw-r--r-- 1 root root 1078 Mar  3 14:45 front-proxy-ca.crt
-rw------- 1 root root 1675 Mar  3 14:45 front-proxy-ca.key
-rw-r--r-- 1 root root 1103 Mar  3 14:45 front-proxy-client.crt
-rw------- 1 root root 1679 Mar  3 14:45 front-proxy-client.key
-rw-r--r-- 1 root root  673 Mar 11 18:05 rizwan.csr
-rw------- 1 root root 1679 Mar  3 14:45 sa.key
-rw------- 1 root root  451 Mar  3 14:45 sa.pub
[root@master pki]#
```

go to this location in master node

and Save the file here

Step 8> Now we are Signing the certificate from master Node

So for signing the certificate we have to follow some standard parametres

```
[root@master pki]# openssl x509 -req -in rizwan.csr -CA ca.crt  -CAkey ca.key -out rizwan.crt
Signature ok
subject=/C=In/ST=MH/L=NGP/O=LW/OU=linux/CN=rizwan/emailAddress=rizwan@pathan.com
Getting CA Private Key
ca.srl: No such file or directory
140076023904160:error:06067099:digital envelope routines:EVP_PKEY_copy_parameters:different parameters:p_lib.c:137:
140076023904160:error:02001002:system library:fopen:No such file or directory:bss_file.c:402:fopen('ca.srl','r')
140076023904160:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:404:
[root@master pki]#
```

what does this command means.

But here we are getting Error so what does it mean?

So, as we have some standard in the market this "x509" is the commonly used standard. And what we say there is some certificate that someone has requested you can get from this file go in and read this file

After read and because you are master you have your own CA so use your certificate and you are CA you are going to sign someone's certificate so attach your private key CA has their own private Key which is located in the same folder. This files have been created when you have configured the master note.

And after signing the certificate save the certificate with .crt extension

So what does above Error means

→ **So whenever they sign a certificate they give one serial number to that certificate , so for the serial number they don't have any database created so that is why we are getting this error. So for the first time we will use some extra parameter to resolve this error.**

```
[root@master pki]#
[root@master pki]# openssl x509 -req -in rizwan.csr -CA ca.crt  -CAkey ca.key -CAcreateserial -out rizwan.crt
Signature ok
subject=/C=In/ST=MH/L=NGP/O=LW/OU=linux/CN=rizwan/emailAddress=rizwan@pathan.com
Getting CA Private Key
[root@master pki]#
```

Successful.

```
sa.key  sa.pub
[root@master pki]# cat ca.srl
A226A11188395AAB
[root@master pki]#
```

Serial no. file.

```
[root@master pki]# ls
apiserver.crt              apiserver.key                 ca.crt  etcd                front-proxy-client.crt  rizwan.csr
apiserver-etcd-client.crt  apiserver-kubelet-client.crt  ca.key  front-proxy-ca.crt  front-proxy-client.key  sa.key
apiserver-etcd-client.key  apiserver-kubelet-client.key  ca.srl  front-proxy-ca.key  rizwan.crt              sa.pub
[root@master pki]# cat sa.
sa.key  sa.pub
[root@master pki]# cat ca.srl
A226A11188395AAB
[root@master pki]# cat rizwan.crt
-----BEGIN CERTIFICATE-----
MIIChTCCAW0CCQCiJqERiDlaqzANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQQDEwpr
dWJlcm5ldGVzMB4XDTIzMDMxMTE4MzAwMVoXDTIzMDQxMDE4MzAwMVoweDELMAkG
A1UEBhMCSW4xCzAJBgNVBAgMAk1IMQwwCgYDVQQHDANOR1AxCzAJBgNVBAoMAkxX
MQ4wDAYDVQQLDAVsaW51eDEPMA0GA1UEAwwGcml6d2FuMSAwHgYJKoZIhvcNAQkB
FhFyaXp3YW5AcGF0aGFuLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
wUAN5DY6I9KvPhgbX1rtqq1/cV2Upj+i4fJJwtZEIzoP+LiUtmikv0FG/YK7RQxf
0ibEM2cydC3v0tLvfV9hHzT8AM+EyhAw0TE7vpI2jyHqF3Dm3DoinYTpaP6gdz5I
bkEQrkkOuvhuftX2k+9epZYqIOyKdZ+8PqItSpkY3dkCAwEAATANBgkqhkiG9w0B
AQsFAAOCAQEAtQMOiZo9wfcuAaIfi9SYDs7Zg280iw8ZCfjvqZURoT32AKsQCDtw
mykl3bVfq8MZN9IkGYTW0BB8s5jDHXOuTmW0PoOweRH87h3bn9Rof/RJUhhmM72T
N6vXvbWXJX+p0av/qrb0YX49mhaESmz+wznMYaR11zAqmTkKWwKJhbYRJdEojDCb
RiSpdHdbgsNCvyzKRlLhOrf8SPkaqfJ86m+hyAhtxK4q79AlwLEbyxSTTmgbIhtN
UGouPqqb2EAl/cylbFZXuGc6tA2d3mY8QlDaIOf/SGJ/dJCxxNdDGqUOaOBPgRkc
nRYKszzszSCiL4pWufeqo4FQY7JOP/t7gQ==
-----END CERTIFICATE-----
[root@master pki]#
```

Now we will Copy this rizwan.crt from master to new machine

```
[root@centos kubeaws]# ll
total 12
-rw-r--r--. 1 root root 937 Mar 12 00:00 rizwan.crt
-rw-r--r--. 1 root root 672 Mar 11 13:56 rizwan.csr
-rw-------. 1 root root 891 Mar 11 13:49 rizwan.key
[root@centos kubeaws]#
```

Now we have to do some Extra things in new system

```
[root@centos kubeaws]#
[root@centos kubeaws]# kubectl
bash: kubectl: command not found...
[root@centos kubeaws]# cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
[root@centos kubeaws]# sudo yum install -y kubectl
Kubernetes                                                    78 kB/s | 167 kB     00:02
Last metadata expiration check: 0:00:01 ago on Sun 12 Mar 2023 12:06:42 AM IST.
Dependencies resolved.
================================================================================
 Package              Architecture          Version                 Repository                Size
================================================================================
Installing:
 kubectl              x86_64                1.26.2-0                kubernetes                11 M
```

We have Install Kubectl Command here.

→ This is freshly Configured VM with K8s we dont have any Idea about where master is located.

So Client/User Should Know few things about, where is ① master API Server Running means Ip and port

② When I reach this master what will be my username

③ what will be my password and as we are using certificate based Auth, so where is my Certificate and private key

```
ght host or port?
[root@localhost kubews]# kubectl --server  -cert
```

If you Run any Command you have to use this part so Instead what we can do we can put this one file Kube config

```
[root@centos kubeaws]# kubectl version
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short
.  Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.2", GitCommit:"fc04e732bb3e7198d2fa44efa545
7c7c6f8c0f5b", GitTreeState:"clean", BuildDate:"2023-02-22T13:39:03Z", GoVersion:"go1.19.6", Compiler:"gc", Platf
orm:"linux/amd64"}
Kustomize Version: v4.5.7
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@centos kubeaws]#
[root@centos kubeaws]#
[root@centos kubeaws]#
```

and because we dont have Kubeconfig file we are getting this Error

So How to Set the Kubeconfig file for the User

→ So there are various ways to Create this file or we Can also copy this file from Master node but we are not going to do that.

root@ip-172-31-91-29:~/.kube

```
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURFekNDQWZ
1Z0F3SUJBZ0lJVk5MU0pMakJWeTB3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2E
zVmlaWEp1WlhSbGN6QWVGdzB5TVRBeU1ESXhOekF5TVRkYUZ3MHlNakF5TURJeE56QXlNakJhTURReAp
GekFWQmdOVkJBb1REbk41YzNSbGJUcHRZWE4wWlhKek1Sa3dGd1lEVlFRREV4QnJkV0psY201bGRHVnp
MV0ZrCmJXbHVNSUlCSWpBTkJna3Foa2lHOXcwQkFRUUZBQU9DQVE4QU1JSUJDZ0tDQVFFQXh0TFJ5MGJ
mNzFSbTdFZHAKSHNBY21mRDBiSWNNVG01THNDbzdmYzFLT2hzeTBkWi9ITG5oWWRLdzhLTHoySENjVEp
IcWYrRjNoSGF6MkY5dQo0MVh2ZVF2YlZVL3Q1ejVtamFEQjk4MTlxWTdsbWVvek9GODhBZ21sRlNqUW8
0WjlmeVZKclJVZnlWbmVJWVphCm9uQ05KaEEyYlc3NzZnV0UzbVFPT1BleW1tZzFPL2hIVE9zVzBWcTI
3dzVlNlNSeGUrTFlZL3hzRXBpQjVYVjQKdFRHV3dOSGpCbVhKZjRWVjkyeWJwUU9hVTF0QzNpb0JLZWo
0ckpSZmJKWHp2QXFHaFAxSy9CcW1mWkpwaXNKTQorSk5zakJLNWdXMUw0c3BzVVdmSVJUYzlNSUY5WHF
hcEJlOVJtQWVKWWNBRzNHVXhMenV0UWZUVVBkL0JWV3B2Ci9iN2VLUUlEQVFBQm8wZ3dSakFPQmdOVkh
ROEJBZjhFQkFNQ0JhQXdFd1lEVlIwbEJBd3dDZ1lJS3dZQkJRVUgKQXdJd0h3WURWUjBqQkJnd0ZvQVV
6OFE1QXZUbVUxR0h5bVdxOGVncFo4dGVuQTB3RFFZSktvWklodmNOQVFFTApCUUFEZ2dFQkFLTTNEZG1
RL1RVNVo4MGw0aHg4cXpMeTVWNDNSYmFGWWp4b2NBZWg0a0dtaWRXNDgwYktqWVpDC1lJYmFWNzBPNEF
kd2RGNUZFdDAweEFlMXQwVzlWdnhKbzZabE1nVDJhN1N1RjZ3ZDU5dWZtYkJpV2M1em1CWVAKZlp5S3d
4S1RGdVlDSmlFdlVkbHJzejFtQzRPYlFEVXc1aVV2RzNiRzl0SFFWSmprSHo3SHBwZStZL1FPcktETwp
NcG1JTXcxWFZQc3dDRjdwcmJ6Tjl3ekV0c1c3RW5wWU5xcDh3VWZld0w5QS91MnRraGxSVEVNSCtDNWh
kYzNLClRsSjVtZUNoUGN4Mng0cHFTOEtmWS83TnpUbStmanA3ZS9WRmVuN1FZdFNYSk1PcmdrakRVVGt
kbVVKN3pEOVgKdWhTRkNhbEhHSFZzZndCeVE4ZUk0TmhvWjRPMU82ND0KLS0tLS1FTkQgQ0VSVElGSUN
```

we dont want to use this password because this is Admin file so It has all the power

So what we wanted is one of my team member

and we have one namespace so he will login and we want to give limited power to this user so obiriously we will give diff username and password to this user

So we will create this config file in our user system

```
[root@centos ~]#
[root@centos ~]# kubectl config --kubeconfig rizwan.kubeconfig set-cluster --server http://13.233.60.160:6443
```

So we are telling kube config asking here to create new kube config file with the name rizwan.kubeconfig

And what we are keeping in this file we are setting up the cluster like we are providing details about the cluster like what is the IP address, what will be Port no

Before Running this command we also need to copy ca.crt from master user system.

```
-rw-r--r-- 1 root root 1143 Mar  3 14:45 apiserver-kubelet-client.crt
-rw------- 1 root root 1675 Mar  3 14:45 apiserver-kubelet-client.key
-rw-r--r-- 1 root root 1066 Mar  3 14:45 ca.crt
-rw------- 1 root root 1675 Mar  3 14:45 ca.key
-rw-r--r-- 1 root root   17 Mar 11 18:30 ca.srl
drwxr-xr-x 2 root root  162 Mar  3 14:45 etcd
-rw-r--r-- 1 root root 1078 Mar  3 14:45 front-proxy-ca.crt
-rw------- 1 root root 1675 Mar  3 14:45 front-proxy-ca.key
-rw-r--r-- 1 root root 1103 Mar  3 14:45 front-proxy-client.crt
-rw------- 1 root root 1679 Mar  3 14:45 front-proxy-client.key
-rw-r--r-- 1 root root  936 Mar 11 18:30 rizwan.crt
-rw-r--r-- 1 root root  673 Mar 11 18:05 rizwan.csr
-rw------- 1 root root 1679 Mar  3 14:45 sa.key
-rw------- 1 root root  451 Mar  3 14:45 sa.pub
[root@master pki]# cat ca.crt
-----BEGIN CERTIFICATE-----
MIIC5zCCAc+gAwIBAgIBADANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQQDEwprdWJl
cm5ldGVzMB4XDTIzMDMwMzE0NDUwNFoXDTMzMDIyODE0NDUwNFowFTETMBEGA1UE
AxMKa3ViZXJuZXRlczCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANEf
IAlFyDU+lcdVoxfxoottFjxfSCWaZgnPjjomfDkAg32vU07NpcAJ15wyj4kOuyra
FVRc6cDsKHVltOufycDnKtWqFtBjicOFDsEWrG8fGldKnHb+FoIu69raoiQ5grzv
qnswRSQqT3X2YwCvPLabX9+jocOYeghl6wZJdelNSUjRrjBDJqLj6tSZUGTNAX8d
ihx5cbcA1mdTObgsm5n0fIS7osRqKBLYTv708yWsJpsKnYJrN1cFvOQ8v0vJtR9C
HBl4pS1bfIKj4mgapSi3uWQSOG5m4jZkcIRvwSJf7Ro4fyynlMebiRE7l0gRsNUf
vUiGZD05pTKpCth5nPsCAwEAAaNCMEAwDgYDVR0PAQH/BAQDAgKkMA8GA1UdEwEB
/wQFMAMBAf8wHQYDVR0OBBYEFHd85USX0AkpNvqxrnLwJmm7pBcpMA0GCSqGSIb3
DQEBCwUAA4IBAQA8+l71DUzUVP0weIS0J4BsAZNY9QVWPE3k1nMgfS3A00CCmgqe
jaMGF5K4OEIb9VwJp2x3+wjjielWdeIdKqtBIobKgTxzrFNRS0jo/vSgaOlkcdXs
z9jq+nWnfBZiySegeb1bPPBpSXGCj2imehGRj75KGkBVLo7hlFzSDRofEags4qAG
P2t17ZTcNUl8ZqaeWmXeG+D/UZ9e23FRhlZMT8HHwkd8yO60TxqRipOcXYnEeMrw
NoTaIMhLPaZU0IRffWy2Bf+KeRODsMfBIekZT6ja6U+pFq3OHQ7Xoy1O5Mu6jlM1
8NdhQDsuczHtVEErK7Br6XYWsXvFoeB4QiQz
-----END CERTIFICATE-----
[root@master pki]#
```

Copy this from master to user system

```
[root@centos kubeaws]# ll
total 12
-rw-r--r--. 1 root root 937 Mar 12 00:00 rizwan.crt
-rw-r--r--. 1 root root 672 Mar 11 13:56 rizwan.csr
-rw-------. 1 root root 891 Mar 11 13:49 rizwan.key
[root@centos kubeaws]# vim ca.crt
[root@centos kubeaws]# ll
total 16
-rw-r--r--. 1 root root 1067 Mar 12 23:36 ca.crt
-rw-r--r--. 1 root root  937 Mar 12 00:00 rizwan.crt
-rw-r--r--. 1 root root  672 Mar 11 13:56 rizwan.csr
-rw-------. 1 root root  891 Mar 11 13:49 rizwan.key
[root@centos kubeaws]#
```

copied from master.

```
[root@centos kubeaws]#
[root@centos kubeaws]#
[root@centos kubeaws]# kubectl config --kubeconfig rizwan.kubeconfig set-cluster awskubecluster --server http://13.233.60.160:6443 --certificate-authority=ca.crt
Cluster "awskubecluster" set.
[root@centos kubeaws]#
```

HTTPS

Any name you can give

Sucessful

```
[root@centos kubeaws]# ll
total 20
-rw-r--r--. 1 root root 1067 Mar 12 23:36 ca.crt
-rw-r--r--. 1 root root  937 Mar 12 00:00 rizwan.crt
-rw-r--r--. 1 root root  672 Mar 11 13:56 rizwan.csr
-rw-------. 1 root root  891 Mar 11 13:49 rizwan.key
-rw-------. 1 root root  207 Mar 12 23:41 rizwan.kubeconfig
[root@centos kubeaws]# cat rizwan.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: http://13.233.60.160:6443
  name: awskubecluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
[root@centos kubeaws]#
```

As it is a https connection so we know which CA sign this to we also know about CA certificate.

IP Add we have Read

Again one Resource

```
[root@centos kubeaws]# kubectl config  view
apiVersion: v1
clusters: null
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
[root@centos kubeaws]#
```

But here we are not getting any Info because Because config command always Read the file from Root directory So we have to Pass Extra parameters

```
[root@centos kubeaws]# kubectl config  view --kubeconfig  rizwan.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: http://13.233.60.160:6443
  name: awskubecluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
[root@centos kubeaws]#
```

Now we are able to See

Now we will try to fetch Details of Cluster

```
[root@centos kubeaws]# kubectl get pods --kubeconfig rizwan.kubeconfig
E0312 23:54:53.603419    3018 memcache.go:238] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32
s": dial tcp [::1]:8080: connect: connection refused
E0312 23:54:53.603797    3018 memcache.go:238] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32
s": dial tcp [::1]:8080: connect: connection refused
E0312 23:54:53.604889    3018 memcache.go:238] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32
s": dial tcp [::1]:8080: connect: connection refused
E0312 23:54:53.606308    3018 memcache.go:238] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32
s": dial tcp [::1]:8080: connect: connection refused
E0312 23:54:53.607727    3018 memcache.go:238] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32
s": dial tcp [::1]:8080: connect: connection refused
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@centos kubeaws]#
```

Because in our file we dont have any Information about user

```
[root@centos kubeaws]# kubectl get pods --kubeconfig rizwan.kubeconfig  set-credentials rizwan --client-certifica
te rizwan.crt --client-key rizwan.key
```

we have set the details

```
[root@centos kubeaws]# kubectl get pods --kubeconfig rizwan.kubeconfig  set-credentials rizwan --client-certifica
te rizwan.crt --client-key rizwan.key
E0312 23:58:23.925860    3060 memcache.go:238] couldn't get current server API group list: Get "http://localhost:
8080/api?timeout=32s": dial tcp [::1]:8080: connect: connection refused
E0312 23:58:23.926208    3060 memcache.go:238] couldn't get current server API group list: Get "http://localhost:
8080/api?timeout=32s": dial tcp [::1]:8080: connect: connection refused
E0312 23:58:23.927425    3060 memcache.go:238] couldn't get current server API group list: Get "http://localhost:
8080/api?timeout=32s": dial tcp [::1]:8080: connect: connection refused
E0312 23:58:23.930543    3060 memcache.go:238] couldn't get current server API group list: Get "http://localhost:
8080/api?timeout=32s": dial tcp [::1]:8080: connect: connection refused
E0312 23:58:23.931947    3060 memcache.go:238] couldn't get current server API group list: Get "http://localhost:
8080/api?timeout=32s": dial tcp [::1]:8080: connect: connection refused
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@centos kubeaws]#
```

Still Error

→ This Lead us to the new concept of Context because in this file all details are there but Context is not present

```
[root@centos kubeaws]# kubectl config  view --kubeconfig  rizwan.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: http://13.233.60.160:6443
  name: awskubecluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
[root@centos kubeaws]#
```

without Context Kubectl doesnt work

So we have to set Context

we have command for this

```
get-clusters    Display clusters defined in the kubeconfig
get-contexts    Describe one or many contexts
get-users       Display users defined in the kubeconfig
rename-context  Renames a context from the kubeconfig file.
set             Sets an individual value in a kubeconfig file
set-cluster     Sets a cluster entry in kubeconfig
set-context     Sets a context entry in kubeconfig
set-credentials Sets a user entry in kubeconfig
unset           Unsets an individual value in a kubeconfig file
use-context     Sets the current-context in a kubeconfig file
view            Display merged kubeconfig settings or a specified kubeconfig
ile
```

Will Continue this topic in next session

Session End