AIM: connect user to K8 cluster

1. Go to master. connect to putty.

2.

a. user – create private key

b. Generate CSR from private key

c. Send this CSR to master of cluster

d. master will sign this CSR(Certificate Signing Request) and CRT(Chinese reminder Theorem) is generated, the one who signed CSR is called CA(certificate Authority)

Q : where is CA present in K8 master ? cd /etc/Kubernetes/pki/ , ls, ca.crt

Q: how to create CSR? In linux kernel based OS, openssl is used.

e. send this CRT to user.

f. installing kubectl

g. Client should know where kube-master API running - give API server IP: port , user , pass: certificate/pvt key

h. Client - > https -> kubemaster – client must have CA crt to connect to https server

i. Set-credentials

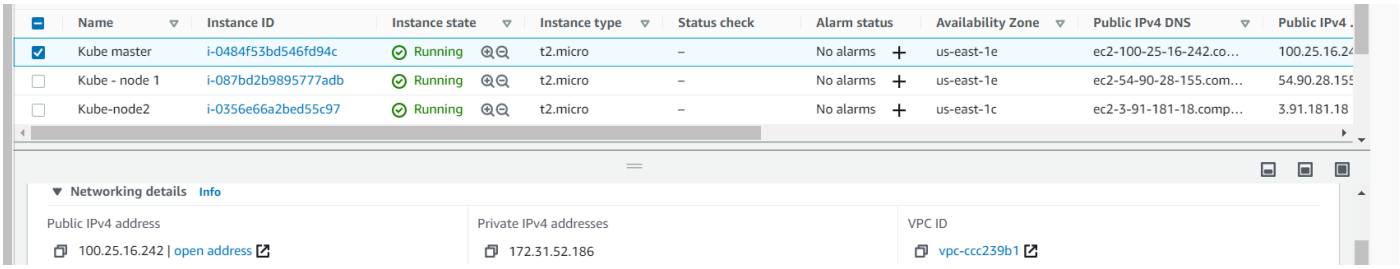
j. creating user not in k8 master – but in VM local system and have to provide the key based authentication.

k. need to set context

connect my local system to Multi node cluster on AWS.

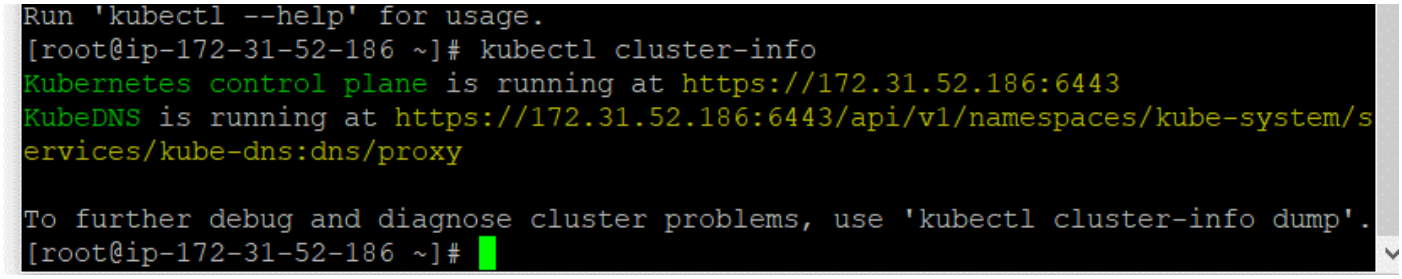
Cluster info

Go to master.



connect to putty.

kubectl cluster-info



What is Kubernetes Control Plane ?

Ans. the master has some programs running for the configuration of master. API server, CM, Schedular…..

These programs together known as Control Plane..

So, master Kubernetes control plane is running at IP : 172.31.52.186 and on port 6443.

AIM: user wants to connect to K8 cluster..

Authentication is required.

types

1. user/pass
2. user/key
3. certificate based – easy to manage(today's practical)
   - 5 steps are involved here.
     a. user – create private key
     b. generate CSR from private key
     c. send this CSR to master of cluster
     d. master will sign this CSR(Certificate Signing Request) and CRT(Chinese reminder Theorem) is generated, the one who signed CSR is called CA(certificate Authority)

        Q : where is CA present in K8 master ? cd /etc/Kubernetes/pki/ , ls, ca.crt

        Q: how to create CSR? In linux kernel based OS, openssl is used.

     e. send this CRT to user.

Brief:

IAM – As it name suggests Identity Access Management, it has two process to do.

Identity => Authentication part - (here) Cert-based – openssl req.

Access => Roles created and attached to user via Role binding.

here, I am making a user with my local VM

a.

```
[root@localhost ~]# cd /kube_ws/
[root@localhost kube_ws]# ls
```

openssl genrsa -out akshay-key 1024

```
[root@localhost kube_ws]# openssl genrsa -out tango.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
......+++++
..+++++
e is 65537 (0x010001)
[root@localhost kube_ws]# ls
tango.key
[root@localhost kube_ws]#
```

we know openssl is used to create the key,

Q: what does genrsa , -out and 1024 in this command indicates. ?

Ans. genrsa –

if the requirement is that need private key with public key – These kind of keys known as Asymmetric keys(Ak) , and the algo used for Ak is “rsa”.

-out => to sav the key

1024 => hers it’s the size of key.

b.

from this pvt-key generate csr certificate..

openssl req -new -key tango.key -out tango.csr

```
[root@localhost kube_ws]# openssl req -new -key tango.key -out tango.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]:IN
State or Province Name (full name) []:Bihar
Locality Name (eg, city) [Default City]:patna
Organization Name (eg, company) [Default Company Ltd]:sacred_devil
Organizational Unit Name (eg, section) []:geeky
Common Name (eg, your name or your server's hostname) []:phir_hera_pheri
Email Address []:iskiTopiUskeSir@xyz.com

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

```
[root@localhost kube_ws]# ls
tango.csr  tango.key
[root@localhost kube_ws]#
```

c.

send this csr to master in dir /etc/Kubernetes/pki

Manually copying user tango.csr to master vim /etc/Kubernetes/pki/tango.csr

```
[root@ip-172-31-52-186 ~]# cd /etc/kubernetes/pki
[root@ip-172-31-52-186 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-172-31-52-186 pki]# vim tango.csr
[root@ip-172-31-52-186 pki]#
```

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB8TCCAVoCAQAwgZYxCzAJBgNVBAYTAklOMQ4wDAYDVQQIDAVCaWhhcjEOMAwG
A1UEBwwFcGF0bmExFTATBgNVBAoMDHNhY3JlZF9kZXZpbDEOMAwGA1UECwwFZ2Vl
a3kxGDAWBgNVBAMMD3BoaXJfaGVyYV9waGVyaTEmMCQGCSqGSIb3DQEJARYXaXNr
aVRvcGlVc2tlU2lyQHh5ei5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGB
AML5H0bdsluu1PPbqyJ8FCj2Fv5YtLdC9zUjczZP2t7GwHqSigluuoequYFJZ76y
RF/cThw8qZ4xAY02L/EjfchH9Nt1CevykONG4C1pdUL3vAHgJ+/MJdIBcc2fEmVR
mlE9CY94yoPzNsQajBwsrOpNYf9vvN9drbvq2X9BvGcZAgMBAAGgGjAYBgkqhkiG
9w0BCQcxCwwJY2hhbGxlbmdlMA0GCSqGSIb3DQEBCwUAA4GBAAGTvvpaWHoGXgdi
5OQ+BsWRYS2KB76jnnro3r99PBHTvKb4TCOfjc08aAzf2ElXYomx5HCJI/kjZoUi
Ld8dBds4HJC55EUaa3kEZkq/2HUFps5nJJef98njcL7CLEaPM6VgsVvndQ+m0tHO
kWZg3VmntUoAP7b+tjEK5rv6Antv
-----END CERTIFICATE REQUEST-----
~
~
~
~
~
~
~
~
~
~
~
"tango.csr"                                                    All
```

root@ip-172-31-52-186:/etc/kubernetes/pki

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB8TCCAVoCAQAwgZYxCzAJBgNVBAYTAklOMQ4wDAYDVQQIDAVCaWhhcjEOMAwG
A1UEBwwFcGF0bmExFTATBgNVBAoMDHNhY3JlZF9kZXZpbDEOMAwGA1UECwwFZ2Vl
a3kxGDAWBgNVBAMMD3BoaXJfaGVyYV9waGVyaTEmMCQGCSqGSIb3DQEJARYXaXNr
aVRvcGlVc2tlU2lyQHh5ei5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGB
AML5H0bdsluu1PPbqyJ8FCj2Fv5YtLdC9zUjczZP2t7GwHqSigluuoequYFJZ76y
RF/cThw8qZ4xAY02L/EjfchH9Nt1CevykONG4C1pdUL3vAHgJ+/MJdIBcc2fEmVR
mlE9CY94yoPzNsQajBwsrOpNYf9vvN9drbvq2X9BvGcZAgMBAAGgGjAYBgkqhkiG
9w0BCQcxCwwJY2hhbGxlbmdlMA0GCSqGSIb3DQEBCwUAA4GBAAGTvvpaWHoGXgdi
5OQ+BsWRYS2KB76jnnro3r99PBHTvKb4TCOfjc08aAzf2ElXYomx5HCJI/kjZoUi
Ld8dBds4HJC55EUaa3kEZkq/2HUFps5nJJef98njcL7CLEaPM6VgsVvndQ+m0tHO
kWZg3VmntUoAP7b+tjEK5rv6Antv
-----END CERTIFICATE REQUEST-----
```

d. make the master sing the csr

how to sign it ?

very commonly used std : x509

```
[root@ip-172-31-52-186 pki]# openssl x509 -req -in tango.csr -CA ca.crt -CAkey c
a.key -out tango.crt
Signature ok
subject=/C=IN/ST=Bihar/L=patna/O=sacred_devil/OU=geeky/CN=phir_hera_pheri/emailA
ddress=iskiTopiUskeSir@xyz.com
Getting CA Private Key
ca.srl: No such file or directory
140707606677408:error:06067099:digital envelope routines:EVP_PKEY_copy_parameter
s:different parameters:p_lib.c:137:
140707606677408:error:02001002:system library:fopen:No such file or directory:bs
s_file.c:402:fopen('ca.srl','r')
140707606677408:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:404:
[root@ip-172-31-52-186 pki]#
```

openssl x509 -req -in tango.csr -CA ca.crt -CAkey ca.key -out tango.crt

openssl - create key

x509 - standard

-req : requirement

-in tango.csr : providing csr file

-CA ca.crt : providing the CA(present in master) to sing tango.csr

CAkey ca.key – providing CA key(present in master)

-out tango.crt - after signing, save as tango.crt

error ? Why ?

Ans. for every crt , it gives a number ie, need a DB.
since it's the first time , I have to create this DB . HOW ? - CAcreateserial

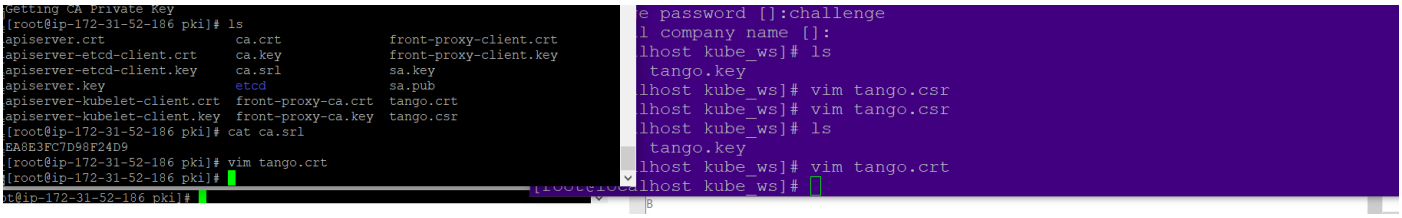
openssl x509 -req -in tango.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tango.crt

```
[root@ip-172-31-52-186 pki]# openssl    x509    -req -in tango.csr   -CA ca.crt
  -CAkey ca.key   -CAcreateserial   -out tango.crt
Signature ok
subject=/C=IN/ST=Bihar/L=patna/O=sacred_devil/OU=geeky/CN=phir_hera_pheri/emailA
ddress=iskiTopiUskeSir@xyz.com
Getting CA Private Key
[root@ip-172-31-52-186 pki]# ls
apiserver.crt                ca.crt              front-proxy-client.crt
apiserver-etcd-client.crt    ca.key              front-proxy-client.key
apiserver-etcd-client.key    ca.srl              sa.key
apiserver.key                etcd                sa.pub
apiserver-kubelet-client.crt front-proxy-ca.crt  tango.crt
apiserver-kubelet-client.key front-proxy-ca.key  tango.csr
[root@ip-172-31-52-186 pki]# cat ca.srl
EA8E3FC7D98F24D9
[root@ip-172-31-52-186 pki]#
```

Right now, ca.srl has only one entry as we created crt just once.

e.

copy the master vim /etc/Kubernetes/pki/tango.crt to local user vim /kube_ws/tango.crt

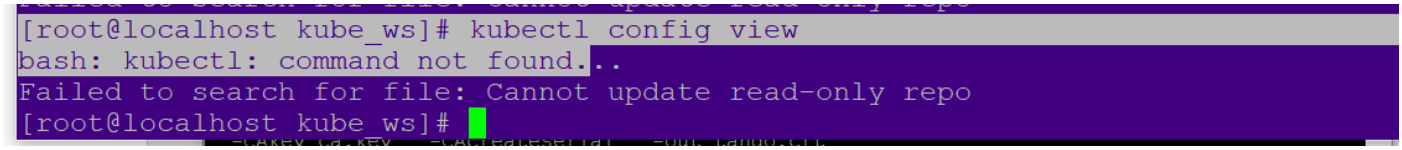


now, we may remove the tango.key, its of no use now..

f.

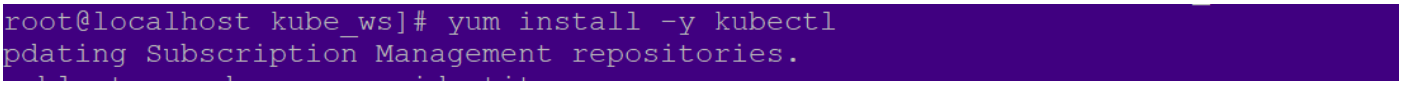
all set, certificate based authentication is achieved,

user can now see the status of k8 cluster by kubectl config view



installing kubectl

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1

gpgcheck=1

repo_gpgcheck=1

gpkgey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
httpsg//packages.cloud.google.com/yum/doc/rpm-package-key.gpg

EOF

yum install -y kubectl
```



```
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/
  Installing       : kubectl-1.20.4-0.x86_64                                1/
  Verifying        : kubectl-1.20.4-0.x86_64                                1/
Installed products updated.

Installed:
  kubectl-1.20.4-0.x86_64
```

g.

```
[root@localhost kube_ws]# kubectl version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6
e03ec3fea7933c4b5263d151aafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVe
rsion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or po
rt?
[root@localhost kube_ws]#
```

y this error?

client should know where kube-master API running : three thing client should know

1. API server IP : port
2. user
3. pass: certificate/pvt key

kubectl config view

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

So how to provide these 3 info to kubeconfig?

h

Asking to create new config file in VM

go to VM: kubectl config –kubeconfig tango.kubeconfig set-cluster –server https:// 100.25.16.242:6443

100.25.16.242 : public IP of master

Scenario

Client - > https -> kubemaster – client must have CA crt to connect to https server

go to master – cd /etc/kubernetes/pki/ copy ca.crt to VM cd/kube_ws/ca.crt

```
[root@localhost kube_ws]# vim ca.crt
[root@localhost kube_ws]# ls
ca.crt  tango.crt  tango.csr  tango.key
[root@localhost kube_ws]#
```

Run

kubectl config --kubeconfig tango.kubeconfig set-cluster myawkubecluster --server https://100.25.16.242:6443 --certificate-authority=ca.crt

```
[root@localhost kube_ws]# kubectl  config  --kubeconfig  tango.kubeconfig  set-cluster myawkub
ecluster --server https://100.25.16.242:6443 --certificate-authority=ca.crt
Cluster "myawkubecluster" set.
[root@localhost kube_ws]#
```

```
Cluster "myawkubecluster" set.
[root@localhost kube_ws]# ls
ca.crt  tango.crt  tango.csr  tango.key  tango.kubeconfig
[root@localhost kube_ws]#
```

```
[root@localhost kube_ws]# vim tango.kubeconfig
[root@localhost kube_ws]#
```

root@localhost:/kube_ws

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://100.25.16.242:6443
  name: myawkubecluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
~
~
```

verifying

```
[root@localhost kube_ws]# kubectl config view  --kubeconfig tango.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://100.25.16.242:6443
  name: myawkubecluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
[root@localhost kube_ws]#
```

i.

```
[root@localhost kube_ws]# kubectl get pods
The connection to the server localhost:8080 was refused - did you specify the right host or p
rt?
[root@localhost kube_ws]#
```

Still the error why ?

have to provide the config file

```
Failed to search for file: cannot update read-only repo
[root@localhost kube_ws]# kubectl get pods --kubeconfig tango.kubeconfig
The connection to the server localhost:8080 was refused - did you specify the right host or po
rt?
[root@localhost kube_ws]#
```

still error why ?

We know IP of serer but don't know the user and password.

need to create the file

help: kubectl config -h

```
  current-context Displays the current-context
  delete-cluster  Delete the specified cluster from the kubeconfig
  delete-context  Delete the specified context from the kubeconfig
  delete-user     Delete the specified user from the kubeconfig
  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
file
```

set-credentials

j. creating user not in k8 master – but in VM local system and have to provide the key based authentication.

i.e now is the time to use vimal.crt

kubectl config --kubeconfig tango.kubeconfig set-credentials tango --client-certificate tango.crt --client-key tango.key

```
[root@phir_hera_pheri ~]# kubectl config --kubeconfig tango.kubeconfig set-crede
ntials  tango --client-certificate tango.crt --client-key tango.key
User "tango" set.
[root@phir_hera_pheri ~]#
```

kubectl config view --kubeconfig tango.kubeconfig

```
[root@phir_hera_pheri kube_ws]# kubectl config view --kubeconfig tango.kubeconfi
g
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://100.25.16.242:6443
  name: myawkubecluster
- cluster:
    certificate-authority: ca.crt
    server: https://100.25.16.242
  name: myawsk8cluster
contexts: null
current-context: ""
kind: Config
preferences: {}
users:
- name: tango
  user:
    client-certificate: tango.crt
    client-key: tango.key
[root@phir_hera_pheri kube_ws]#
```

user is set.

```
    client-key: tango.key
[root@phir_hera_pheri kube_ws]# kubectl get pods --kubeconfig tango.kubeconfig
The connection to the server localhost:8080 was refused - did you specify the ri
ght host or port?
[root@phir_hera_pheri kube_ws]#
```

again same error why ?

we need to set the contexts too….