03Nov2022
Day**20**
Kubernetes RBAC

Using RBAC Authorization

https://kubernetes.io/docs/reference/access-authn-authz/rbac/
Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within your organization

How enable RBAC in K8s cluster

To enable RBAC, start the API server with the --authorization-mode flag set to a comma-separated list that includes RBAC.

API objects

The RBAC API declares four kinds of Kubernetes object: *Role*, *ClusterRole*, *RoleBinding* and *ClusterRoleBinding*

Role and ClusterRole

-An RBAC Role or RBAC ClusterRole contains rules that represent a set of permissions.
-Permissions are purely additive, complete positive.

Role
A Role always sets permissions

RBAC Role
A Role always sets permissions within a particular *namespace*; when you create a Role, you have to specify the namespace it belongs in.
RoleBinding
A RoleBinding may reference any Role in the same namespace

RBAC ClusterRole
ClusterRole, by contrast, is a *non-namespaced* resource.
Cluster RoleBinding
To grant permissions across a whole cluster, you can use a ClusterRoleBinding.

Role vs ClusterRole

if want to define a Role within specific namespace, use Role; if want to define a Role cluster-based, use ClusterRole

RoleBinding vs ClusterRoleBinding

A RoleBinding grants permissions within specific namespace, ClusterRoleBinding grants permissions cluster-Based.

How specific user connects/communicates to K8s Cluster

```
# kubectl get nodes
# kubectl config view
# cat /etc/kubernetes/admin.conf
# whoami
root
# pwd
/root
# ll -a
drwxr-xr-x  3 root root     4096 Jun 12 06:41 .kube/
# ls .kube/
cache/  config
# cat .kube/config
```

need *ca*, *pub* and *private* key files to continuing this procedure.
ca, *pub* and *private* key default location in K8s Cluster:
ls /etc/kubernetes/pki/

Procedure, how user get access to K8s Cluster

Step1-user from integrated account manager platform
Step2-create private-key ->*.key
Step3-create certification Signing Requests ->*.csr
Step4-create public-key ->*.pub

Step1-user from integrated account manager platform

user from integrated platform likes IPA, LDAP, AD should be available.

```
# id devops
uid=1000(devops) gid=1000(devops) groups=1000(devops),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
# tail -1 /etc/passwd
devops:x:1000:1000:manent:/home/devops:/bin/bash
# kubectl create namespace testsp
# kubectl get ns
# mkdir certificates
# cd certificates/
# pwd
/root/certificates
```

Step2-create private-key
Step3-create certification Signing Requests
Step4-create public-key

1-create private-key

```
# pwd
/root/certificates
# openssl genrsa -out devops.key 2048
# ls
devops.key
```

2-create certification Signing Requests

```
# openssl req -new -key devops.key -out devops.csr -subj "/CN=devops/O=testsp"
# ls
devops.csr  devops.key
```

what is CA

A certificate authority server (CA server) offers an easy-to-use, effective solution to create and store asymmetric key pairs for encrypting or decrypting as well as signing or validating anything that depends on a public key infrastructure (PKI)

```
# cp /etc/kubernetes/pki/ca.* .
# ls
ca.crt  ca.key  devops.csr  devops.key
```

3-create public-key

```
# openssl x509 -req -in devops.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out devops.crt -days 365
# ls
ca.crt  ca.key  ca.srl  devops.crt  devops.csr  devops.key
```

verify crt file

```
# openssl x509 -enddate -noout -in devops.crt
notAfter=Nov  3 15:03:22 2023 GMT
```

reference

OpenSSL
https://www.digicert.com/kb/ssl-support/openssl-quick-reference-guide.htm

Now, its time to create kubeconfig file

```
# pwd
/root/certificates
# kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://192.168.29.104:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
# kubectl --kubeconfig devops.kubeconfig config set-cluster kubernetes --server https://192.168.29.104:6443 --certificate-authority ca.crt
# ls
ca.crt  ca.key  ca.srl  devops.crt  devops.csr  devops.key  devops.kubeconfig
# cat devops.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://192.168.29.104:6443
  name: kubernetes
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
# kubectl --kubeconfig devops.kubeconfig config set-credentials devops --client-certificate devops.crt --client-key devops.key
```

```
# cat devops.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://192.168.29.104:6443
  name: kubernetes
contexts: null
current-context: ""
kind: Config
preferences: {}
users:
- name: devops
  user:
    client-certificate: devops.crt
    client-key: devops.key
# kubectl get ns
testsp          Active   35m
# kubectl --kubeconfig devops.kubeconfig config set-context devops-testsp --cluster kubernetes --namespace testsp --user devops
# cat devops.kubeconfig
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://192.168.29.104:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    namespace: testsp
    user: devops
  name: devops-testsp
current-context: ""                                  ->current-context: "" == --current in 'kubectl config set-context --namespace testsp' cmd and should be defines
kind: Config
preferences: {}
users:
- name: devops
  user:
    client-certificate: devops.crt
    client-key: devops.key
# vim devops.kubeconfig
current-context: "devops-testsp"                     ->append name in to double-quote
:wq!
```

verify

```
# kubectl --kubeconfig devops.kubeconfig get pods
Error from server (Forbidden): pods is forbidden: User "devops" cannot list resource "pods" in API group "" in the namespace "testsp"
```

Now, time to assign permission to user through Role and RoleBinding

-Role

```
# kubectl create role devops.role --namespace testsp --verb get,list --resource pod -o yaml --dry-run=client >devops.role.yaml
# ls
devops.role.yaml
# cat devops.role.yaml
# kubectl create role devops.role --namespace testsp --verb get,list --resource pod
# kubectl get role --namespace testsp
NAME          CREATED AT
devops.role   2022-11-03T15:31:04Z
# kubectl describe role --namespace testsp devops.role
Name:       devops.role
Labels:     <none>
Annotations:  <none>
PolicyRule:
Resources  Non-Resource URLs  Resource Names  Verbs
---------  -----------------  --------------  -----
 pods      []                 []              [get list]
```

Modify existing Role

```
# kubectl edit role --namespace testsp devops.role
```

-Rolebinding

```
# kubectl create rolebinding devops.rolebinding --namespace testsp --user devops --role devops.role -o yaml --dry-run=client >devops.rolebinding.yaml
# ls
devops.rolebinding.yaml  devops.role.yaml
# cat devops.rolebinding.yaml
# kubectl create rolebinding devops.rolebinding --namespace testsp --user devops --role devops.role
# kubectl get rolebindings.rbac.authorization.k8s.io --namespace  testsp
# kubectl describe  rolebindings.rbac.authorization.k8s.io --namespace  testsp
```

Modify existing Rolebinding

```
# kubectl edit  rolebindings.rbac.authorization.k8s.io --namespace  testsp
```

verify

```
# kubectl --kubeconfig devops.kubeconfig run nginx --image nginx --namespace testsp
Error from server (Forbidden): pods is forbidden: User "devops" cannot create resource "pods" in API group "" in the namespace "testsp"
# kubectl --kubeconfig devops.kubeconfig get pods --namespace testsp
No resources found in testsp namespace.
```

Now, make it accessible for target user, devops

```
# pwd
/root/certificates
# ls
ca.crt  ca.key  ca.srl  devops.crt  devops.csr  devops.key  devops.kubeconfig  devops.rolebinding.yaml  devops.role.yaml
# mkdir -p /home/devops/.kube
# cp devops.kubeconfig ca.crt devops.crt devops.key /home/devops/.kube
# ls -l /home/devops/.kube/
ca.crt  devops.crt  devops.key  devops.kubeconfig
# chown -R devops:devops /home/devops/.kube/
# ls -l /home/devops/.kube/
# mv /home/devops/.kube/devops.kubeconfig /home/devops/.kube/config
# ls /home/devops/.kube/
ca.crt  config  devops.crt  devops.key
```

Now, customer should have theses files to make a connection.

```
# cat /home/devops/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority: ca.crt
    server: https://192.168.29.104:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    namespace: testsp
    user: devops
  name: devops-testsp
current-context: "devops-testsp"
kind: Config
preferences: {}
users:
- name: devops
  user:
    client-certificate: devops.crt
    client-key: devops.key
```

verify from outside through ssh connection

```
make ssh from MS-Win10
C:\Windows\system32>ssh devops@192.168.29.104
devops@192.168.29.104's password: ubuntu
$ kubectl get pods
No resources found in testsp namespace.
devops@master1:~$ kubectl get deployment
No resources found in testsp namespace.
devops@master1:~$ kubectl create deployment dpl1 --image nginx --replicas 2
error: failed to create deployment: deployments.apps is forbidden: User "devops" cannot create resource "deployments" in API group "apps" in the namespace "testsp"
devops@master1:~$ exit
```

Now, to increase performance put ca.crt, devops.crt and devops.key content in to **config** file.

```
# cd /home/devops/.kube/
# pwd
/home/devops/.kube
# ls
cache  ca.crt  config  devops.crt  devops.key
# cat ca.crt  | base64 -w0
```

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURVUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1EWXhNakEyTXpNeE1Wb1hEVE15TURZd09U
QTJNek14TVZvd0ZURVRNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBUE5vCnppZUxlM0dDWI3YUZRS2pERkZtZbzVSSkJBWHgrVWRkTHhJcFRwdkdRZGsv
di9oRk5zUUZsQW5YOG5kbTMKWnpVUlk4THR2Zlh4dWtRSXlYYUlrT0cvQVJiVkV4aHFMYUI0NmlQaWpmTlA5bVZSSjNmOGZOZFc3Mm1sRDNYbwozYyt1bitGY05hSHI5bSs5dkpzMkV4R3JTL0cxTDliUzd1V21IbzF3dm5NNkk2d2wzUzgvRHU3L
2dFdFdtWXVDClovVm5nbVlmUHRidXQ1RHI0dVA1TnZxa3l0dVJ6bUVNN2I4L052TklpaWdZdnBlV2F2MWdqcE8yV08wN0hZRlYKenhXcldnTTVwVCs5TjQ0UCt5SXora0laQUZGYzRjNVNxNCs2NXh3UStuNkZjSkgvNXFNOE55ODhiZmZaRy92QQ
pDM2kyaENQNkNkQk9qTnliUk1NQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURVUjBPQkJZRUZNaGkrVFN1UXFXT0ZXWG11aDdZUU1NYVRpazNNQlVHQTFVZEVRUU8KTU
F5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBRFBZUC9VN3BlbnkyRXlHc3p5LwpnTnE2WEtsYmdHay9GUUxvTGhGeHAzSmpVMDdKeXZFVy9QL1p0RVY4UjJGeTVWMmdGSTAvcDJYSDlTdlliMlZ6CjBIVEJvTS9J
UkV3VkZWWnU1ZitENWVqQ2swSjQxQXJ3VjJYQ21TOENDNWx5c2xMamVISVM4ZXdpeWlVMnc5Z3UKQ1IxN3Yxc2dzWTluNnNpci9McE5qalZTSTloT0J0ZnZhV2xPbmV2RG4wclhvU0pKWklpSzkyNlBwM0FqaDdoSgpWSGJCQ3pXSk9YOUVYe
m5nQWg0R3RQSDQ5NFNGOTNUbkdGNnZYbUFBWExRa2o4ZFNaOWF2OHBqanpRVjFzbEhYClhmZ2duS2tmR3VhbWZIN0ovRk1FZ1RtdmkwTFlCMFJ4TSs2VFpER2tiQVFmN3VNVnU2c3A3ZUNjK3E4R2VnOVEKdXRrPQotLS0tLUVORCBDRVJU
SUZJQ0FURS0tLS0tCg==

```
# cat devops.crt  | base64 -w0
```

LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN2akNDQWFZQ0ZBdTBOS0xkSERiV1l2YUI2L2ViMnBEQlNsZVdNQTBHQ1NxR1NJYjNEUUVCQ3dVQU1CVXgKRXpBUkJnTlZCQU1UQ210MVltVnlibVYwWlhNd0hoY05Nakl4TVRBek1UVXdNe
kl5V2hjTk1qTXhNVEF6TVRVdwpNekl5V2pBaU1ROHdEUVlEVlFRRERBWmtaWFp2Y0hNeER6QU5CZ05WQkFvTUJuUmxjM1J6Y0RDQ0FTSXdEUVlKCktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FRb0NnZ0VCQUs3aTdBNjk0MmpYSDFYYUR
ORmMvNjlLUnBZK1hZaHUKTGFNS2VHYTRNYk1KTldSdnlrajdadjRlQVY3cGxXV2RvalVBSENuVER1SUh2OUxDNFcwYzZ0S0p4V3hVd3FZRAp2V0hWWTYweXZha1M4MFZrK05rWDVnQW5SVFpbm5DM2VyZHJoRTZlaXpyZEtnYzZkTndsb0FWY
XN3cGF3VEVvCnNvclQxZUw2cDJuWklRMnh0Z213SzN5OC8rMHJ5VWtlZGJXQy9FbDRCSHJuSThleHBUMmd4Rkpzd2I1MUducmIKRmdHTHVnNEg2aFRkMkVCTWc4SXFUMFdzTHBNekRnaHlKWVWRqalo1WktOZS9MSk9iREh0OHFPOFJiOEl5T
lN1aQo2aDZWOVJmMW9wRHdzdFVnMDdGVmhod1pKK2QwMU5ZSUNHelZjclJiYm1leGFUSll0RWZjSFU4Q0F3RUFBVEFOCkJna3Foa2lHOXcwQkFRc0ZBQU9DQVFFQUdzOFI1QO9UZkNaYzZmNjlnMUJqdFlwaElYT1hKaENmb0JuUVM3bnUK
TTM2Z2ExVXJ0SGVERSsyVWg3UDZySmVzSVdQbHZVZjlDQ0UwNUFRdnE1T3RHQjVyRUM3T2trZ3pRRHo2ZU1tcwpibWlzK01COWVnVit6TkFRUDJZUzU3eEZQV0Q3WmxtSjZCblEwVG1oTFBxYTRNQ3lHNmRQQytuTWFyRWtqTFhHClJDYnUya
DdDaGpSSzZCTU4wM0hSemo5R3BHZjBTRDU4MWJjTm9uTVBuc01aUWVQeU5yaUEyVVdZRG5ONFJiTmcKUE4vdDd4dm1hb2VndHhYbmRLZ0dtRlM2Nys1WlJSNW90THNoZ1JidlQzNEFWbm9lOFh6SlZ6RnR6bGlqWFMveAp2TzFrVTJOTjdE
VUNyRGI1M1RhVm0wNkxrM09SdVM0TEJoZ0xVTEo5RGY5cm9BPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=

```
# cat devops.key  | base64 -w0
```

LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBUUVBcnVMc0RyM2phTmNmVmRvTTBWei9yMHBHbGo1ZGlHNHRvd3A0WnJneHN3azFaRy9LCINQdG0vaDRCWHVtVlpaMVNOUUFjS2RNTzRnZS8wc0xoYlJ6cTBvbkZ
iRlRDcGdPOVlkVmpyVEs5cVJMeUJXVDQKMlJmbUFDZEZObXllY0xkNnQydUVUcDZMT3QwcUJ6cDAzQ1dnQlZxekNsckJNU2l5aXRQVjR2cW5hZGtoRGJHMgpDYkFyZkx6LzdTdkpTUjUxdFlMOFNYZ0VldWNqd2ZHbFBhREVVbXpCdm5VYWV0c1
dBWXU2RGdmcUZOM1lRRXlIECndpcFBSYXd1a3pNT0NISWxoMk9Obmxrbz E3OHNrNXNNZTN5bzd4RnZ3akkxSzZMcUhwWDFGL1dpa1BDeTFTRFQKc2A241M1RVMWdnSWJOVnl0RnR1WWZGcE1saTBSOXdkVHdJREFRQUJBb0lCQU
hlTkxEcXppZ1d3NXEwaApCUEpBQ0FseTFkZnlvVDZrSTRKU3orSlh2Yzk1RUl0WXkxTU9kUWlyVWMwSFJyWW0xMVd4eXFIVXUzeU4rbDhWCmZVZFpaOXkwMWdrV1pzdE01N0dBQUNxNkk9BQ0xaeR2MUZiQVhHU0JLV0t1NzdMOVVoZUZH
VVZubWNZbUlMRXYKY3RmR2pBaXY2bzIwWEw2MmN1Z3Z6S0MwNXJmVEJUaVhBTEUxcSt3blZES3pMZFdWRHFNY3oyT2JSeUZrQ1dOeAprblc3TjlPT3h4T28zQmh1RDNMU1cycm9MNmkzVW1ON0g1OHBTbVNmNE1hSVdYTHBaYldSQWZ
WSi9ubTROWmlXClFvTVFuaTRabEgrSWRVOGk0Yit1Tk1rYktGbUFXOS9iMmJFZ01ML0FkcnhOMjg1VUtlNzQ3cStMa2tGS1l0dGcKSkFZU0xWRUNnWUVBM0ljLJQaGcvajZNK05BanJvWmdyNmhkclhLcU5PYndSdVpseWhOUndJRlZWdW1scW
9RTwpLRm9DNzhrbnVwT0cvWkxHMXF3cWEzQ1Z3bUZvZ3BrdlQ2Wm5iaVh0aDhZU2xQcjQvb1ArNm84YkhWd0N6eEFjCnpJNXZvQVVZQzgrbTlPc2FteGdJeXNwbEtTVVJSa0xNcC9FbDlDYmN6TnRRT3JiQ2FYN0I1Z3NDZ1lFQXl3UkQKMm5Xe
klWN0dvR3BkWnhPOU8yZzh0MThIcVpackltWDkyaGJEM3NNYzhyd0gycW44Z2hPNFM3STNEbVpZSk9xKwpKUlVVNXg5YWOzVjhPQVhNQXB0U284SzhtUW56U1ZsN3 ViTzJLZHZOdWdQQmQvTlo3Y082Rys0ZHVpc1JLWVZlCmc3bXhMMIJIN
UZGaTk4NFNsM0JlNy9nQ09sZStPUVo1OHpPdFJFMENnWUVBcXhFbmJmQUduWEs4UEU2SlAxcnkKS1pEN2lnN0d4ZDdYU01oMDhYcG5FbVBCWlhnOWtMWjduUVdsL1lnWEQzNWRXdHErNzh3QjQvSWoyZk5oOU4wTApmRWZqTitpb0I0aT
Q1V1NDUGp6NlE4M2JCTDFaV0hKeFJLVEJKbXFOS3F4Y0lPVTNjRlFHamZqTzNFbnF5b3dLCjRjUUk0d282Z1pjbTkwVEtMcmVyaGQwQ2dZQTRYRHpDSHo3Vm1zd2xxS0c1aGY5WXlMMWlZdDFydFRpVTJoKzUKTmV5Mm9SQWVTeGtiR1duaW
Fyc1ZPS3FVTXRGeG5rR2ZaZFRodlJDWWErdEtXa0dpejZBSG1uNWxOaC8yVXdOSQpNSjMzek8rMkhtak5kaCtYSkdrOGE4U1E5SGZWUVJOV0dtMFlMUXpRVTgvWDM3dkZrd3FvTC8vZVZwR1duemVuCnpzYmhXUUtCZ0IvY1k0WjUzbXZtRHpP
NHg2ZWFRckt2MlVUb3hDUHlPWmJCTXF5MzVRbFl0S0kvTk1qcnRkbFYKM05wYWJDbTBjTWQyTXRkOTI1d2JHbVRJeXBsOVVZTFo0WE5pYWxBa1M1cnljWkhBK2RsdlRRQkZVV01reW5rUApYTTVLUys5NWNyc0VDTG1nOG9EU0dmVDliZ2tkTFlvb0ZBUjY2K2JvQzJiUE8vaXEvU1k1Ci0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==

vim config

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:                                    ->certificate-authority: ca.crt to certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURVUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJeU1EWXhNakEyTXpNeE1Wb1hEVE15TURZd09UQTJNek14TVZvd0ZURV
RNQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVBKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBUE5vCnppZUxIM0dDYWI3YUZRS2pERE1YbzFBcFRwdkdRZGsvdi9oRk5zUUZsQW5YOG5kbTMKWnpVUlk4THR
2Zlh4dWtRSXlYYUlrT0cvQVJiVkV4aHFMYUI0NmIQaWpmTlA5bVZSSjNmOGZOZFc3Mm1sRDNYbwozYyt1bitGY05hSHI5bSs5dkpzMkV4R3JTL0cxTDliUzd1V21IbzF3dm5NNkk2d2wzUzgvRHU3L2dFdFdtWXVDClovVm5nbVlmUHRidXQ1RHI0dVA1TnZxa3l0dVJ6bUVN
N2l4L052TklpaWdZdnBlV2F2MWdqcE8yV08wN0hZRlYKenhXcldnTTVwVCs5TjQ0UCt5SXora0laQUZGYzRjNVNxNCs2NXh3UStuNkZjSkgvNXFNOE55ODhiZmZaRy92QQpDM2kyaENQNkNkQk9qTnliUk1NQ0F3RUFBYU5aTUZjd0RnWURWUjBQQVFIL0JBUURBZ0trTU
E4R0ExVWRFd0VCCi93OUUZNQU1CQWY4d0hRWURWUjBPQkJZRUZNaGkrVFN1UXFXT0ZXWG11aDdZUU1NYVRpazNNQlVHQTFVZEVRUU8KTUF5Q0NtdDFZbVZ5Ym1WMFpYTXdEUVlKS29aSWh2Y05BUUVMQlFBRGdnRUJBRFBZUC9VN3BlbnkyRXlHc3p5LwpnTnE
2WEtsYmdHay9GUUxvTGhGeHAzSmpVMDdKeXZFVy9QL1p0RVY4UjJGeTVWMmdGSTAvcDJYSDlTdlliMlZ6CjBlVEJvTS9JUkV3VkZWWnU1ZitENWVqQ2swSjQxQXJ3VjJYQ21TOENDNWx5c2xMamVlSVM4ZXdpeWlVMnc5Z3UKQ1IxN3Yxc2dzWTluNnNpci9McE5qaIZ
TSTloT0J0ZnZhV2xPbmV2RG4wclhvU0pKWklpSzkyNlBwM0FqaDdoSgpWSGJCQ3pXSk9YOUVYem5nQWg0R3RQSDQ5NFNGOTNUbkdGNnZYbUFBWExxRa2o4ZFNaOWF2OHBqanpRVjFzbEhYClhmZ2duS2tmR3VhbWZlN0ovRk1FZ1RtdmkwTFlCMFJ4Tss2VFpER2tiQ
VFmN3VNVnU2c3A3ZUNjK3E4R2VnOVEKdXRrPQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://192.168.29.104:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    namespace: testsp
    user: devops
  name: devops-testsp
current-context: "devops-testsp"
kind: Config
preferences: {}
users:
- name: devops
  user:
    client-certificate-data:                                       ->client-certificate: devops.crt to client-certificate-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN2akNDQWFZQ0ZBdTBOS0xkSERiV1I2YUI2L2ViMnBEQlNsZVdNQTBHQ1NxR1NJYjNEUUVCQ3dVQU1CVXgKRXpBUkJnTlZCQU1UQ210MVltVnlibVYwWlhNd0hoY05Nakl4TVRBek1UVXdNekl5V2hjTk1qTXhNVEF6T
VRVdwpNekl5V2pBaU1ROHdEUVlEVlFRRERBWmtaWFp2Y0hNeER6QU5CZ05WQkFvTUJuUmxjM1J6Y0RDQ0FTSXdEUVlKKktvWklodmNOQVFFQkJRQURnRnZOVQQURDQ0FRb0NnZ0VCQUs3aTdBNjk0MmpYSDFYUXORmMvNjILUnBZK1hZaHUKTGFNS2VHYTRNYk1
KTldSdnIrajdadjRlQVY3cGxXV2RValVBSENuVER1SUh2OUxDNFcwYzZ0S0p4V3hVd3FZRAp2V0hWWTYweXZha1M4MFZrK05rWDVnQW5SVFpzbm5DM2VyZHJoRTZlaXpyZEtnYzZkTndsb0FWYXN3cGF3VEVvCnNvclQxZUw2cDJuWklRMnh0Z213SzN5OC8rMHJ5VWtl
ZGJXQy9FbDRCSHJuSThleHBUMmd4Rkpzd2I1MUducmIKRmdHTHVnNEg2aFRkMkVCTWc4SXFUMFdzTHBNekRnaHlKWWRqalo1WktOZS9MSk9iREh0OHFPOFJiOEI5TlN1aQo2aDZWOVJmMW9wRHdzdFVnMDdGVmhod1pKK2QwMU5ZSUNHelZjclJiYm1leGFUSll0
RWZjSFU4Q0F3RUFBVEFOCkJna3Foa2lHOXcwQkFRc0ZBQU9DQVFFQUdzOFI1Q09UZkNaYzZmNjlnMUJqdFlwaElYT1hKaENmb0JuUVM3bnUKTTM2Z2ExVXl0SGVERSsyVWg3UDZySmVzSVdQbHZVZjlDQ0UwNUFRdnE1T3RHQjVyRUM3T2trZ3pRRHo2ZU1tcwpibWl
zK01COWVnVit6TkFRUDJZUzU3eEZQV0Q3WmxtSjZCblEwVG1oTFBxYTRNQ3lHNmRQQytuTWFyRWtqTFhHClJDYnUyaDdDaGpSSzZCTU4wM0hSemo5R3BHZjBTRDU4MWJjTm9uTVBuc01aUWVQeU5yaUEyVVdZRG5ONFJiTmcKUE4vdDd4dm1hb2VndHhYbmRLZ0
dtRlM2Nys1WlJSNW90THNoZ1JidlQzNEFWbm9lOFh6SlZ6RnR6bGlqWFMveAp2TzFrVTJOTjdEVUNyRGI1M1RhVm0wNkxrM09SdVM0TEJoZ0xVTEo5RGY5cm9BPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
    client-key-data:                                               ->client-key: devops.key to client-key-data:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dIQkFBS0NBUUVBcnVMc0RyM2phTmNmVmRvTTBWei9yMHBHbGo1ZGlHNHRvd3A0WnJneHN3azFaRy9LClNQdG0vaDRCWHVtVlpaMVNOUUFjS2RNTzRnZS8wc0xoYlJ6cTBvbkZiRlRDcGdPOVlkVmpyVE
s5cVJMelJXVDQKMlJmbUFDZEZObXlIY0xkNnQydUVUcDZMT3QwcUJ6cDAzQ1dnQlZxekNsckJNU2l5aXRQVjR2cW5hZGtoRGJHMgpDYkFyZkx6LzdTdkpTUjUxdFlMOFNYZ0VldWNqd2ZHbFBhREVVbXpCdm5VYWV0c1dBWXU2RGdmcUZOM1lRRXlECndpcFBSYXd1a3
pNT0NlSWxoMk9ObmxrbzE3OHNrNXNNZTN5bzd4RnZ3akkxSzZMcUhwWDFGL1dpa1BDeTFTRFQKc2ZXR0hCa241M1RVMWdnSWJOVnl0RnR1WWZGcE1saTBSOXdkVHdJREFRQUJBb0lCQUhlTkxEcXppZ1d3NXEwaApCUEpBQ0FseTFkZnlvVDZrSTRKU3orSlh2Yzk1R
UI0WXkxTU9kUWlyVWMwSFJyWW0xMVd4eXFIVXUzeU4rbDhWCmZVZFpaOXkwMWdrV1pzdE01N0dBQUNxMk9BQ0xXazR2MUZiQVhHU0JLV0t1NzdMOVVoZUZHVVZubWNZbUlMRXYKY3RmR2pBaXY2bzlwWEw2MmN1Z3Z6S0MwNXJmVEJUaVhBTEUxcSt3blZ
ES3pMZFdWRHFNY3oyT2JSeUZrQ1dOeAprblc3TjlPT3h4T28zQmh1RDNMU1cycm9MNmkzVW1ON0g1OHBTbVNmNE1hSVdYTHBaYldSQWZWSi9ubTROWmlXClFvTVFuaTRabEgrSWRVOGk0Yit1Tk1rYktGbUFXOS9iMmJFZ01ML0FkcnhOMjg1VUtlNzQ3cStMa2tGS1
l0dGcKSkFZU0xWRUNnWUVBM0IjL3JQaGcvajZNK05BanJvWmdyNmhkclhLcU5PYndSdVpseWhOUndJRlZWdW1scW9RTwpLRm9DNzhrbnVwT0cvWkxHMXF3cWEzQ1Z3bUZvZ3BrdlQ2Wm5iaVh0aDhZU2xQcjQvb1ArNm84YkhWd0N6eEFjCnpJNXZvQVVZQzgrbTlP
c2FteGdJeXNwbEtTVVJSa0xNcC9FbDlDYmN6TnRRT3JiQ2FYN0I1Z3NDZ1IFQXl3UkQKMm5XekIWN0dvR3BkWnhPOU8yZzh0MThlcVpackltWDkyaGJEM3NNYzhyd0gycW44Z2hPNFM3STNEbVpZSk9xKwpKUlVVNXg5YW0zVjhPQVhNQXB0U284SzhtUW56U1ZsN3ViT
zlLZHZOdWdQQmQvTlo3Y082Rys0ZHVpc1JLWVZlCmc3bXhMMlJINUZGaTk4NFNsM0JINy9nQ09sZStPUVo1OHpPdFJFMENnWUVBcXhFbmJmQUduWEs4UEU2SlAxcnkKS1pEN2InN0d4ZDdYU01oMDhYcG5FbVBCWlhnOWtMWjduUVdsL1lnWEQzNWRXdHErNzh3
QjQvSWoyZk5oOU4wTApmRWZqTitpb0l0aTQ1V1NDUGp6NlE4M2JCTDFaV0hKeFJLVEJKbXFOS3F4Y0lPVTNjRlFHamZqTzNFbnF5b3dLCjRjUUk0d282Z1pjbTkwVEtMcmVyaGQwQ2dZQTRYRHpDSHo3Vm1zd2xxSOc1aGY5WXlMMWlZdDFydFRpVTJoKzUKTmV5Mm9
SQWVTeGtiR1duaWFyc1ZPS3FVTXRGeG5rR2ZaZFRodlJDWWErdEtXa0dpejZBSG1uNWxOaC8yVXdOSQpNSjMzek8rMkhtak5kaCtYSkdrOGE4U1E5SGZWUVJOV0dtMFlMUXpRVTgvWDM3dkZrd3FvTC8vZVZwR1duemVuCnpzYmhXUUtCZ0lvY1k0WjUzbXZtRHpPNH
g2ZWFRckt2MlVUb3hDUHlPWmJCTXF5MzVRbFlOS0kvTk1qcnRkbFYKM05wYWJDbTBjTWQyTXRkOTl1d2JHbVRJeXBsOVVZTFo0WE5pYWxBa1M1cnljWkhBK2RsdlRRQkZVV01reW5rUApYTTVLUys5NWNyc0VDTG1nOG9EU0dmVDliZ2tkTFlvb0ZBUjY2K2JvQzJiUE8v
aXEvU1k1Ci0tLS0tRU5EIFJTQSBQUklWQVRFIEtFWS0tLS0tCg==
```

:wq!

```
# rm -rf ca.crt devops.crt devops.key
# ls
cache  config
```

verify from outside through ssh connection

make ssh from MS-Win10

C:\Windows\system32>ssh devops@192.168.29.104

devops@192.168.29.104's password: **ubuntu**

```
$ kubectl get pods
No resources found in testsp namespace.
devops@master1:~$ kubectl get deployment
No resources found in testsp namespace.
devops@master1:~$ kubectl create deployment dpl1 --image nginx --replicas 2
error: failed to create deployment: deployments.apps is forbidden: User "devops" cannot create resource "deployments" in API group "apps" in the namespace "testsp"
devops@master1:~$ exit
```

Done!

Mohammad Ali Naghval
ali@prodevans.com