RBAC (Role based access control)

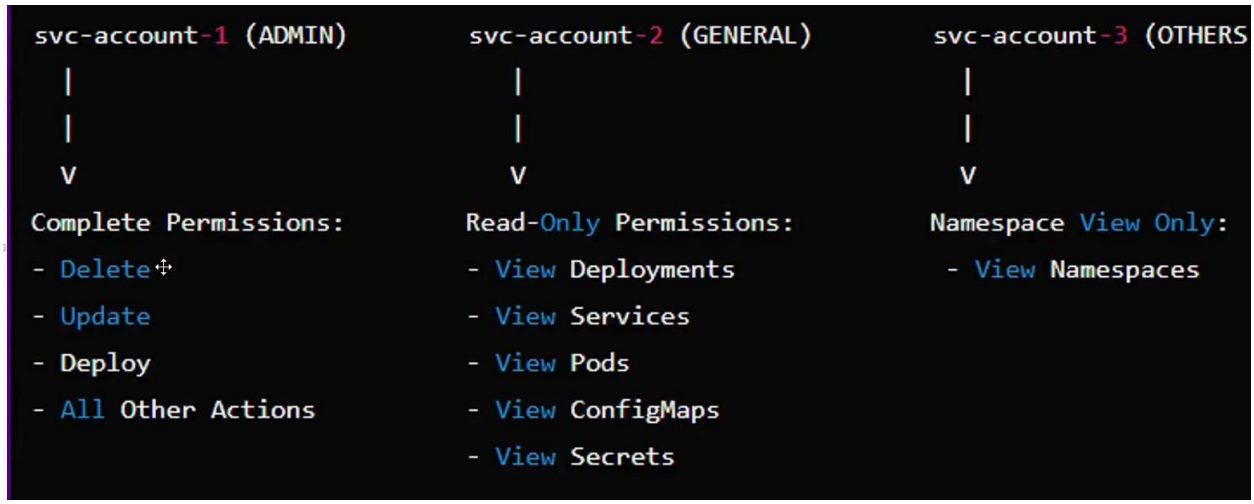
→A **Role** gives permission **within a specific namespace**, while a **ClusterRole** gives permissions **across the entire cluster** or **all namespaces**.

→ We create a **ServiceAccount** in Kubernetes to grant **automated processes or applications** specific permissions to interact with the cluster securely.

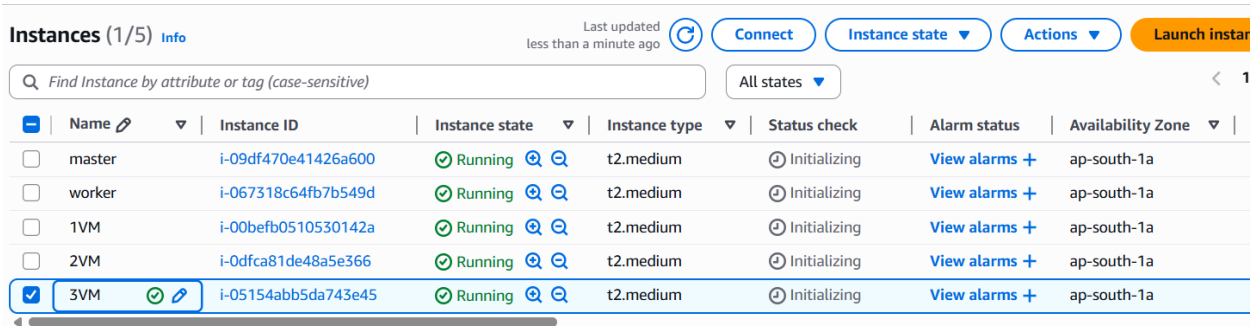
KubeConfig Use-

- **Stores cluster access info** like server address, user credentials, and certificates.
- **Let you switch between multiple clusters** easily using contexts.
- **Authenticates users** so they can securely access the cluster.
- **Used by kubectl** to know which cluster to talk to and how.

Below user's will be created with different permission



Created 5 virtual machines (VMs): 2 for the master and node, and 3 for testing access levels to the cluster. On the 3 test VMs, kubectl will be installed and a separate kubeconfig file will be created for each user using a specific token. The kubeconfig file will include the certificate authority data and the cluster endpoint URL, allowing users to access, view, and manage the cluster according to their assigned roles.



➔ YAML to create a service account and role and role binding for admin. Where admin will be having full level of access

```
➔ apiVersion: v1
➔ kind: ServiceAccount
➔ metadata:
➔   name: admin
➔   namespace: default
➔ ---
➔ apiVersion: rbac.authorization.k8s.io/v1
➔ kind: ClusterRole
➔ metadata:
➔   name: admin-role
➔ rules:
➔   - apiGroups: ["*"]
➔     resources: ["*"]
➔     verbs: ["*"]
➔ ---
➔ apiVersion: rbac.authorization.k8s.io/v1
➔ kind: ClusterRoleBinding
➔ metadata:
➔   name: admin-rolebinding
➔ roleRef:
➔   apiGroup: rbac.authorization.k8s.io
➔   kind: ClusterRole
➔   name: admin-role
➔ subjects:
➔   - kind: ServiceAccount
➔     name: admin
➔     namespace: default
```

```
ubuntu@ip-172-31-42-65:~$ kubectl apply -f admin.yml
serviceaccount/admin created
clusterrole.rbac.authorization.k8s.io/admin-role created
clusterrolebinding.rbac.authorization.k8s.io/admin-rolebinding created
ubuntu@ip-172-31-42-65:~$
```

→ YAML to create a service account and role and role binding for general user, where the user will be having view resource permissions only

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: general
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: general-role
rules:
  - apiGroups: [""]
    resources: ["pods", "services", "endpoints", "namespaces"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps", "extensions"]
    resources: ["deployments", "replicasets", "daemonsets", "statefulsets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs", "cronjobs"]
    verbs: ["get", "list", "watch"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
```

```
  name: general-rolebinding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: general-role
subjects:
  - kind: ServiceAccount
    name: general
    namespace: default
```

```
ubuntu@ip-172-31-42-65:~$ kubectl apply -f gen.yml
serviceaccount/general created
clusterrole.rbac.authorization.k8s.io/general-role created
clusterrolebinding.rbac.authorization.k8s.io/general-rolebinding created
ubuntu@ip-172-31-42-65:~$
```

→ YAML to create a service account and role and role binding for other user, where the user will be having view permission for namespaces only

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: others
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: others-role
rules:
  - apiGroups: [""]
```

```
  resources: ["namespaces"]
  verbs: ["get", "list", "watch"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
 name: others-rolebinding
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: others-role
subjects:
 - kind: ServiceAccount
   name: others
   namespace: default
```

```
ubuntu@ip-172-31-42-65:~$ kubectl apply -f other.yml
serviceaccount/others created
clusterrole.rbac.authorization.k8s.io/others-role created
clusterrolebinding.rbac.authorization.k8s.io/others-rolebinding created
ubuntu@ip-172-31-42-65:~$
```

→ **Generate Tokens for ServiceAccounts**

For Admin Service Account

#kubectl -n default create token admin

ubuntu@ip-172-31-42-65:~$ kubectl -n default create token admin

eyJhbGciOiJSUzI1NiIsImtpZCI6IlR4ZzdEYXJXb1V5b1hJLXJQZHVtZ0RfTDhxejRLWGpEMWRL
dklqVnlSMXMifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsd
XN0ZXIubG9jYWwiXSwiZXhwIjoxNzUzNzg4MDI2LCJpYXQiOjE3NTM3ODQ0MjYsImlzcyI6Im
h0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiMDYx
MTc4NWMtMzQxMi00Y2YxLWFlN2QtNTFmMmI4NTU4ZGJiIiwia3ViZXJuZXRlcy5pbyI6eyJuY
W1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZSI6ImFkbWluIiwidWlkI
joiMjUyNWJhMmMtNDdlYy00MGE4LTllMzMtMzc1MGQ2YTdmODEwIn19LCJuYmYiOjE3NT
M3ODQ0MjYsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmFkbWluIn0.R
XYGxtXlWqlufv_8JjghsdkCqgOFJ8VbSRKhLOF6uVOjMap499zya1j7lNHnLim8482HFx4aaF-
BT3HsDporAl6doI9R8Jh3LBSJ4_ZPADrlN-
m4TZ26vGF7sr0D2NK1XLpcxXDLQzzKTn9Xxr625DQATUb1DiPe2D6FTHt9ADhTAWzJC55U3
OFoh3ut6r_B5BTnBlqxuIl7uJQ4X9hXRRgObqnlQbMLJfqxlMuZceB1B5N-
kJW5WkGVVTIrA2tmgie1j-5uBDhwRkAWSBkxM-
Lt760tzlZiYuD_fZtuioPtEs6fQoRYVnntuDzpBWge7slnmyaZKA_NVTVuBsqYrA

For General Service Account

kubectl -n default create token general

ubuntu@ip-172-31-42-65:~$ kubectl -n default create token general

eyJhbGciOiJSUzI1NiIsImtpZCI6IlR4ZzdEYXJXb1V5b1hJLXJQZHVtZ0RfTDhxejRLWGpEMWRL
dklqVnlSMXMifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsd
XN0ZXIubG9jYWwiXSwiZXhwIjoxNzUzNzg4MDYzLCJpYXQiOjE3NTM3ODQ0NjMsImlzcyI6Im
h0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiNTc3
NTI4NGEtYTQwMC00ZTA5LWIwZTktNzdkZDg5OTQ2ZmJkIiwia3ViZXJuZXRlcy5pbyI6eyJuYW
1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZSI6ImdlbmVyYWwiLCJ1a
WQiOiJmNTgyMzViZi0zYWM0LTQzNmMtYWNlZS04ZjhjY2FmY2ViNTAifX0sIm5iZiI6MTc1Mz
c4NDQ2Mywic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6Z2VuZXJhbCJ9.l
HrC8fcrm-M2Dv5JM-hHKsM7cZ-
6mOYTM7uhwsoc1REUjpQSDCqTqngCqC1239qJ4kqvP_X5oNzuQGkQ-
Uw2xMAc1htb1Yv20WUk4DEq5Ul3wCsDopeSnUco7qSSdBGEb43ALuhny3ESDuIK8J_LkXu
0TPfw0ftyJyDQcuNI_E4-
fj_vhawWqidPhIE8wtgOg3A20sdhauiv33fGIrbS4hCadg9onCs4EGximPOyu5RxsQxe7V6IXVt
PqMV3bv6s33n0m_FEkr55uw9M9leppdZfQuUrStUMylBRZ8Kdwx_owXgZw84VXFNMG9Tffv
kRKRuzkRI26fnN-KOMy4gIJw

For Others Service Account

kubectl -n default create token others

ubuntu@ip-172-31-42-65:~$ kubectl -n default create token others

eyJhbGciOiJSUzI1NiIsImtpZCI6IlR4ZzdEYXJXb1V5b1hJLXJQZHVtZ0RfTDhxejRLWGpEMWRLdklqVnlSMXMifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzUzNzg4MDgzLCJpYXQiOjE3NTM3ODQ0ODMsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiZTNjYjQ0OTQtYmZhOC00Mzg4LWEyNWMtMzYxNjI1MjRjOTcyIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJkZWZhdWx0Iiwic2VydmljZWFjY291bnQiOnsibmFtZSI6Im90aGVycyIsInVpZCI6IjljY2VmNzhjLTEzNzAtNDUyOS1hN2M5LTViNjQ1MGQ2ZmY4NCJ9fSwibmJmIjoxNzUzNzg0NDgzLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6ZGVmYXVsdDpvdGhlcnMifQ.s52-lW5rmYYaYTh8B9xbGOQSarw16m0jtbK4y7UZXs25OZgsVwlsBxTEdJNv8UwcWG4ZOJbSUeqdIAkKF3Mas2KJIlxadg__TpK8xQq2EoOhVbmwfwQsFbsIjOGo8KefoKu2suyw6Ux2vLzN9NTpXSujJHsf4rfXmadEXDJHWRL0zkxtGiSx85CVVF79y4TH7aHRs1IxnnnkyOkyKk866ivcgTvEpWe6XnsWYJY0LDxw0BETj8tBF5bRlLGMJd5SbvMj8kFA3ytyYJ_fBwbg2nSQ16ShPfVoAa2sLfAWJhlYH5z13s9G7JhHEcO4ZtLqN-a65uUAWVE8MpTVM3FuFw

Created the 3 namespaces

```
ubuntu@ip-172-31-42-65:~$ kubectl create namespace ns1
namespace/ns1 created
ubuntu@ip-172-31-42-65:~$ kubectl create namespace ns2
namespace/ns2 created
ubuntu@ip-172-31-42-65:~$ kubectl create namespace ns3
namespace/ns3 created
ubuntu@ip-172-31-42-65:~$
```

➔ Deployed a pods in Namespace2 (ns2)

```
ubuntu@ip-172-31-42-65:~$ kubectl apply -f ds.yml -n ns2
deployment.apps/boardgame-deployment created
service/boardgame-ssvc created
ubuntu@ip-172-31-42-65:~$
```

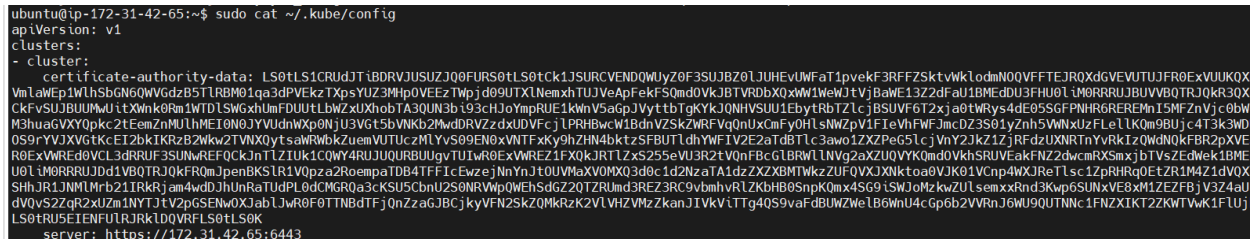
Create Kubeconfig Files in each VM's

Use the tokens generated in the previous step to create kubeconfig files for each ServiceAccount.

Save this content to ` admin-kubeconfig.yaml

Cmd to check the kubeconfig data

sudo cat ~/.kube/config



Kubeconfig yaml for admin user , and now we will apply this kubeconfig file in newly created server and test the access

apiVersion: v1

clusters:

- cluster:

certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJUHEvUW
FaT1pvekF3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp
1WlhSbGN6QWVGdzB5TlRBM01qa3dPVEkzTXpsYUZ3MHpOVEEzTWpjd09UTXlNemxhTUJ
VeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRR
UJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUUMwUitXWnk0Rm1WTDlSWGxhUmFDUUtLbWZ
xUXhobTA3QUN3bi93cHJoYmpRUE1kWnV5aGpJVyttbTgKYkJQNHVSUU1EbytRbTZlcjBSUVF
6T2xja0tWRys4dE05SGFPNHR6REREMnI5MFZnVjc0bWoyc1JMM3huaGVXYQpkc2tEemZn
MUlhMEI0N0JYVUdnWXp0NjU3VGt5bVNKb2MwdDRVZzdxUDVFcjlPRHBwcW1BdnVZSkZW
RFVqQnUxCmFyOHlsNWZpV1FIeVhFWFJmcDZ3S01yZnh5VWNxUzFLellKQm9BUjc4T3k3
WDU5czVIOS9rYVJXVGtKcEI2bkIKRzB2Wkw2TVNXQytsaWRWbkZuemVUTUczMlYvS09EN0
xVNTFxKy9hZHN4bktzSFBUTldhYWFIV2E2aTdBTlc3awo1ZXZPeG5lcjVnY2JkZ1ZjRFdzUXNR
TnYvRkIzQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQ
WY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJRTlZxS255eVU3R2tVQnFBcGlBRWllNVg2aXZ
UQVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd
1VBQTRJQkFRQmJpenBKSlR1VQpza2RoempaTDB4TFFIcEwzejNnYnJtOUVMaXVOMXQ3d0
c1d2NzaTA1dzZXZXBMTWkzZUFQVXJXNktoa0VJK01VCnp4WXJReTlsc1ZpRHRqOEtZR1M4Z

1dVQXlMYkFUSHhJR1JNMlMrb21IRkRjam4wdDJhUnRaTUdPL0dCMGRQa3cKSU5CbnU2S0
NRVWpQWEhSdGZ2QTZRUmd3REZ3RC9vbmhvRlZKbHB0SnpKQmx4SG9iSWJoMzkwZUls
emxxRnd3Kwp6SUNxVE8xM1ZEZFBjV3Z4aUFreGE0dVQvS2ZqR2xUZm1NYTJtV2pGSENwO
XJablJwR0F0TTNBdTFjQnZzaGJBCjkyVFN2SkZQMkRzK2VlVHZVMzZkanJIVkViTTg4QS9vaFd
BUWZWelB6WnU4cGp6b2VVRnJ6WU9QUTNNc1FNZXIKT2ZKWTVwK1FlUjhpCi0tLS0tRU5
EIENFUlRJRklDQVRFLS0tLS0K

```
    server:  https://172.31.42.65:6443  # Your K8s API server endpoint
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: admin
  name: admin-context
current-context: admin-context
kind: Config
preferences: {}
users:
- name: admin
  user:
    token: <admin-token>  # Replace with the generated token
```

After copying the kubeconfig file to the newly hosted VM, set the KUBECONFIG environment variable so the system can use it as this specific file is kubeconfig file and authenticate to cluster

```
kubectl 1.33.3 from Canonical✓ installed
ubuntu@ip-172-31-37-251:~$ sudo vi kubeconfig-admin.yml
ubuntu@ip-172-31-37-251:~$
```

Set the `KUBECONFIG` environment variable to point to the desired kubeconfig file.

export KUBECONFIG=/path/to/admin-kubeconfig.yaml

```
ubuntu@ip-172-31-37-251:~$ export KUBECONFIG=/home/ubuntu/kubeconfig-admin.yml
ubuntu@ip-172-31-37-251:~$ kubectl get nodes
NAME              STATUS   ROLES           AGE   VERSION
ip-172-31-37-17   Ready    <none>          84m   v1.30.14
ip-172-31-42-65   Ready    control-plane   85m   v1.30.14
ubuntu@ip-172-31-37-251:~$
```

➔ We can see admin user with admin privileges able to do all the task and successfully authenticated to nodes

```
ubuntu@ip-172-31-37-251:~$ kubectl get ns
NAME              STATUS   AGE
default           Active   86m
kube-node-lease   Active   86m
kube-public       Active   86m
kube-system       Active   86m
ns1               Active   57m
ns2               Active   57m
ns3               Active   57m
ubuntu@ip-172-31-37-251:~$ kubectl get all -n ns2
NAME                                        READY   STATUS    RESTARTS   AGE
pod/boardgame-deployment-7599597df4-5p6bc   1/1     Running   0          52m
pod/boardgame-deployment-7599597df4-tw6vh   1/1     Running   0          52m

NAME                     TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/boardgame-ssvc   LoadBalancer   10.99.127.108   <pending>     80:31367/TCP   52m

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/boardgame-deployment    2/2     2            2           52m

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/boardgame-deployment-7599597df4    2         2         2       52m
ubuntu@ip-172-31-37-251:~$
```

```
ubuntu@ip-172-31-37-251:~$ kubectl delete pod boardgame-deployment-7599597df4-5p6bc -n ns2
pod "boardgame-deployment-7599597df4-5p6bc" deleted
```

➔ Now same things will test for general user

general-kubeconfig.yaml

apiVersion: v1

clusters:

- cluster:

```
    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJZCt1WD
R


NSjVsVUl3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1
WlhSbGN6QWVGdzB5TkRBM01UTXhOakkxTXpkYUZ3MHpOREEzTVRFeE5qTXdNemRhTUJ
VeApFekFSQmdOVkJBTVRDbXQxWW1WeWJtVjBaWE13Z2dFaU1BMEdDU3FHU0liM0RRR
UJBUVVBQTRJQkR3QXdnZ0VLCkFvSUJBUURVU3NOMGxEbWhFUlZFOE92MTBKeFFTeTByd
XVOOTJxNkhRMGN6VU5PZTFuckdPNGZKY0FGdmNYSG0KM2FCR0J1V0lQOGtvWUNwUEh
nVjd5NjYxR2ZqU0UrUnpYb2tHcTY3ZldFZng1bFJScUVHTjByL0kvcndQY2pPagpoazh5R0RO
bnRzU2hnbzhsZUVWYk5YNkNhcWltRDFGMGxiejk1YUg3VEhiZ3k4RFU1V1NzRk9PWVlWW
mdlSDg2Cm5kK3gveG5VMUdwbkhLbFV0VDlnQldrSTI4b1pFZG4yS21zbHlNMVpjeFF4cGd
0NldtV0VGc1lDZHJUeDZTa0oKZ1dseHE3alZUNHNISktKRFo5bXUxQkFLalZneVcxT0Z5SEFR
WWZhWURGS093NjFrNjJhNkdHdk96S05sYS90YQpheSt1M2Z2cUYxaFNldGp6NlJZQ0NFMj
l1aElWQWdNQkFBR2pXVEJYTUE0R0ExVWREd0VCL3dRRUF3SUNwREFQCkJnTlZIUk1CQ
WY4RUJUQURBUUgvTUIwR0ExVWREZ1FXQkJSYmdWMkdkS01lSnF3YVlQSk1JeVJ5d28raU
56QVYKQmdOVkhSRUVEakFNZ2dwcmRXSmxjbTVsZEdWek1BMEdDU3FHU0liM0RRRUJDd
1VBQTRJQkFRQ3ZtQXlnaXQ3SwpLdGZYRDBzcmxTVS84ODYzMEJXNStWQ0pMSTUwaFUz
NTNGM1lJY2FEd3V1NDBGY2RMdUgyM2pHV0ozWnZOYWo2Ck1Qa3ZGWHNxbUZpOXJIM
E82UTU0K0NCRTZuUmRBelplblo1Nmg1QlFyZmIyNmdUYVVrMWVMM3daV2dGTWRvOEYK
eGJFVFhxbXRRSDFpU1ZmakRuN3RXWXdpVVp0VmFwZGY4LzBwWEdnY01jdjZOcy9xRHJ5
bjY0d2wrTlk0VENscwpZYXJ1WmQ4blkrM010bGxwQ0VYajJGOEN5V3diaXBRN1p1ZG14cU
RVZ242aGQ3MTVWWmo1Zml2aXFlSnQzdUZxCldBa3B4Y0Q1eUEyNHF0RXpGcUVYUjhm
OFRYRzZKZFpvTmtKUGxEODBiQjhjRlVRcTgxWkZPaHhnV3NzNGxoZGQKYnFTelZLU0tYckFI
Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K

    server: https://172.31.45.104:6443  # Your K8s API server endpoint

  name: kubernetes

contexts:

- context:

    cluster: kubernetes

    user: general

  name: general-context

current-context: general-context
```

kind: Config

preferences: {}

users:

- name: general

  user:

    token: <general-token> # Replace with the generated token

➔ We can see general user is only having view permissions.