Summary

- A service account in Kubernetes is used for identity & access management
- In the service account, the name of an account is like a username & instead of a password we have a token
- Listing all the service accounts in all the namespaces

```
C:\Users\Vimal Daga>kubectl get sa  --all-namespaces
NAMESPACE         NAME                                 SECRETS   AGE
default           default                              1         119d
ingress-nginx     default                              1         32d
ingress-nginx     ingress-nginx                        1         32d
ingress-nginx     ingress-nginx-admission              1         32d
kube-node-lease   default                              1         119d
kube-public       default                              1         119d
kube-system       attachdetach-controller              1         119d
kube-system       bootstrap-signer                     1         119d
kube-system       certificate-controller               1         119d
kube-system       clusterrole-aggregation-controller   1         119d
kube-system       coredns                              1         119d
kube-system       cronjob-controller                   1         119d
kube-system       daemon-set-controller                1         119d
kube-system       default                              1         119d
kube-system       deployment-controller                1         119d
kube-system       disruption-controller                1         119d
kube-system       endpoint-controller                  1         119d
kube-system       endpointslice-controller             1         119d
```

- Every service account has a token for authentication

```
C:\Users\Vimal Daga>kubectl  get  sa    default  -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: "2021-11-30T16:13:37Z"
  name: default
  namespace: default
  resourceVersion: "428"
  uid: 0a372bfe-6d30-46ba-a143-46a052207ab2
secrets:
- name: default-token-qmvqq
```

- Directory in which the token is stored inside a pod

```
}[root@mypod /]# cd /var/run/
console/  httpd/    log/      sepermit/ systemd/  utmp
faillock/ lock/     secrets/  setrans/  user/
[root@mypod /]# cd /var/run/secrets/
[root@mypod secrets]# ls
kubernetes.io
[root@mypod secrets]# cd kubernetes.io/
[root@mypod kubernetes.io]# ls
serviceaccount
[root@mypod kubernetes.io]# cd serviceaccount/
```

```
[root@mypod serviceaccount]# ls
ca.crt  namespace  token
[root@mypod serviceaccount]# cat token
eyJhbGciOiJSUzI1NiIsImtpZCI6Ii00c19rdlR1SmF4SnFjU2pjcllacW5xbk
6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwI
mh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiw
0IiwicG9kIjp7Im5hbWUiOiJteXBvZCIsInVpZCI6ImU1YTcxMzlkLTA4ZTQtN
3VudCI6eyJuYW1lIjoiZGVmYXVsdCIsInVpZCI6IjBhMzcyYmZlLTZkMzAtNDZ
2NDg1NzMwODV9LCJuYmYiOjE2NDg1Njk0NzgsInN1YiI6InN5c3RlbTpzZXJ2a
olYV8doZr42_cZ8qWGtCywLn5jyRXFHwmf9_pVh7gEqBdLOmKN50eF9jvyOIBO
-QlyG0zAcWPAzm_J82-4jGmQvbxpkXnEJ9z3A5yFCyv8umHgIjKoooXdmp2ob5
laGXp-lTIaloMFWQktHhxL1jrJMK2txkGUbsEscio_OOy02M38atm926_QZWeQ
hg[root@mypod serviceaccount]#
```

- How to use a token in the curl command
  - First, we have to store the token in a variable

```
hg[root@mypod serviceaccount]# t=$( cat token )
[root@mypod serviceaccount]# echo $t
eyJhbGciOiJSUzI1NiIsImtpZCI6Ii00c19rdlR1SmF4SnFjU2pjcllacW5xbkM0WWtNWVJOdE
6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNjgwMTA1M
mh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXF
0IiwicG9kIjp7Im5hbWUiOiJteXBvZCIsInVpZCI6ImU1YTcxMzlkLTA4ZTQtNDhhZC1hNTU3L
3VudCI6eyJuYW1lIjoiZGVmYXVsdCIsInVpZCI6IjBhMzcyYmZlLTZkMzAtNDZiYS1hMTQzLTQ
2NDg1NzMwODV9LCJuYmYiOjE2NDg1Njk0NzgsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3Vud
olYV8doZr42_cZ8qWGtCywLn5jyRXFHwmf9_pVh7gEqBdLOmKN50eF9jvyOIBO4h86bdH41gbz
-QlyG0zAcWPAzm_J82-4jGmQvbxpkXnEJ9z3A5yFCyv8umHgIjKoooXdmp2ob5u9mNyuBXeXP9
laGXp-lTIaloMFWQktHhxL1jrJMK2txkGUbsEscio_OOy02M38atm926_QZWeQzjmmNEijZvFJ
```

  - Passing token in the curl command

```
[root@mypod serviceaccount]# curl  -H   "Authorization: Bearer $t"  https://192.168.59.104:8443/api/v1
secure
```

- For using any resources of Kubernetes we have to go through the API program
- Any program that wants to use the API service of Kubernetes needs a token for authorization
- How to create a service account
  - Command:- kubectl create serviceaccount (name)

```
C:\Users\Vimal Daga>kubectl create serviceaccount mysa
serviceaccount/mysa created
```

- Tokens for service account is stored in the secret service of Kubernetes

```
C:\Users\Vimal Daga>kubectl get sa
NAME      SECRETS   AGE
default   1         119d
mysa      1         5s

C:\Users\Vimal Daga>kubectl describe  sa mysa
Name:                mysa
Namespace:           default
Labels:              <none>
Annotations:         <none>
Image pull secrets:  <none>
Mountable secrets:   mysa-token-ndsz9
Tokens:              mysa-token-ndsz9
Events:              <none>

C:\Users\Vimal Daga>kubectl get secrets
NAME                  TYPE                                  DATA   AGE
default-token-qmvqq   kubernetes.io/service-account-token   3      119d
mysa-token-ndsz9      kubernetes.io/service-account-token   3      21s
mysecret              Opaque                                2      77d
mysql-pass-8d668bfdmt Opaque                                1      76d
```

- The manifest file of role binding for a service account

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eric-user-montoring-binding
  namespace: testing
subjects:
  - kind: ServiceAccount
    name: mysa
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: mymonitor-role
  apiGroup: rbac.authorization.k8s.io
```