Kubernetes

1. Configurar certificados para RBAC

- Creamos un certificado para almacenar todos los ficheros

```
mkdir certs
cd certs
```

- Generamos una clave privada para un usuario llamado “desa1”. Se crea un fichero denominado desa1.key

```
openssl genrsa -out desa1.key 4096
```

- Vamos a crear un archivo de configuración para gestionar la petición de solicitud de firma. Le llamamos “desa1.csr.cnf”

```
[ req ]
default_bits = 2048
prompt = no
default_md = sha256
distinguished_name = dn
[ dn ]
CN = desa1
O = desarrollo
[ v3_ext ]
authorityKeyIdentifier=keyid,issuer:always
basicConstraints=CA:FALSE
keyUsage=keyEncipherment,dataEncipherment
extendedKeyUsage=serverAuth,clientAuth
```

- Ahora hacemos la solicitud de firna. Con este commando se genera un fichero “desa1.csr”

```
openssl req -config desa1.csr.cnf -new -key desa1.key -nodes -out desa1.csr
```

- Generamos una llamada al cluster de Kubernetes para aprobar o rechazar la petición de firma. En este caso se crea un objeto de tipo **CertificateSigningRequest** y le pasamos desa1.csr codificado en base 64
- Entre los usages indicamos “server auth” y “client auth”

```
cat <<EOF | kubectl apply -f -
apiVersion: certificates.k8s.io/v1beta1
kind: CertificateSigningRequest
metadata:
  name: desa1-firma   # nombre de la peticion
spec:
  groups:
  - system:authenticated
  request: $(cat desa1.csr | base64 | tr -d '\n')
  usages:
  - digital signature
  - key encipherment
  - server auth
  - client auth
EOF
```

- Comprobamos que el certificado está:

```
kubectl get csr
```

- Comprobamos que la petición está pendiente

```
NAME            AGE   REQUESTOR          CONDITION
desa1-firma   37s   minikube    Pending
```

- Aprobamos el certificado

```
kubectl certificate approve desa1-firma
```

- Lo descargamos en el fichero desa1.crt en el directorio certs.

```
kubectl get csr sammy-authentication -o jsonpath='{.status.certificate}' | base64 --decode > desa1.crt
```

- Creamos un directorio llamado “.kube” dentro del directorio “certs”
- Copiamos el fichero config de /home/Kubernetes/.kube al directorio .kube de certs.
- Lo dejamos de forma que pueda acceder al cluster con el usuario “desa1”

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /home/kubernetes/.minikube/ca.crt
    server: https://192.168.99.101:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    namespace: ventas
    user: desa1
  name: desa1-context
current-context: desa1-context
kind: Config
preferences: {}
users:
- name: desa1
  user:
    client-certificate: /home/kubernetes/certs/desa1.crt
    client-key: /home/kubernetes/certs/desa1.key
```

- Para poder usar el usuario sin tener que poner el ficher de confioguración definimos la variable KUBECONFIG para que apunte al directorio /home/kubernetes/certs/.kube/config
- Con eso ya Podemos probar por ejemplo

```
kubectl get pods
```

- Debe generar un error, al no tener permisos de acceso. Algo que arreglaremos en los siguientes capítulos
-