1. **Create the user key file and generate the CSR file:**

**openssl genrsa -out newcorpsite.key 2048**

**openssl req -new -key newcorpsite.key -subj "/CN=newcorpsite" -out newcorpsite.csr**

1. **Encode the CSR file:**

**cat newcorpsite.csr | base64 | tr -d "\n"**

1. **Put the output of the above command in the "request" field of the**

**vi certificateSigningRequest.yaml**

**apiVersion: certificates.k8s.io/v1**

**kind: CertificateSigningRequest**

**metadata:**

**name: newcorpsite**

**spec:**

**request: <encoded-csr-output>**

**signerName: kubernetes.io/kube-apiserver-client**

**expirationSeconds:** 31536000

**usages:**

**- client auth**

1. **Create the CertificateSigningRequest:**

**kubectl create -f certificateSigningRequest.yaml**

1. **Check the CertificateSigningRequest status:**

**kubectl get csr**

1. **Approve the CertificateSigningRequest:**

**kubectl certificate approve newcorpsite**

1. **Retrieve the .crt file from the approved CSR:**

**kubectl get csr newcorpsite -o json | jq -r '.status.certificate' | base64 -d > newcorpsite.crt**

**Now you have the user certificate file (rasa.crt) that can be used along with the private key (rasa.key) to authenticate as the "rasa" user.**

1. **Create a role with the required permissions:**

**kubectl create role newcorpsite-developer --verb=create,delete,list,watch,update,edit --resource=pods,deployments --namespace=corpsite**

1. **Create a role binding for the "rasa" user:**

**kubectl create rolebinding newcorpsite-developer-binding --user=newcorpsite --role=newcorpsite-developer --namespace=corpsite**

1. **Set the credentials for the "rasa" user:**

**kubectl config set-credentials newcorpsite --client-key=newcorpsite.key --client-certificate=newcorpsite.crt**

**Set the context for the "rasa" user: ( kubectl config get-contexts) to get contexts name**

**kubectl config set-context newcorpsite@kubernetes --cluster=kubernetes --user=newcorpsite --namespace=corpsite**

**Use the following command to switch to the "rasa" user's context:**

**kubectl config use-context kubernetes-admin@kubernetes**

**To verify if the "rasa" user has the necessary permissions, you can use the kubectl auth can-i command:**

**kubectl auth can-i delete deployments --as=newcorpsite**

**kubectl auth can-i list pods --as=newcorpsite**

1. **After ensuring that the permissions are set correctly, you can share the edited .kube/config file with the "rasa" user to allow them to connect to the cluster and perform the required operations.**

**To change the context to the "rasa" user:**

**kubectl config use-context newcorpsite@kubernetes**

**To check the current context:**

**kubectl config current-context**

**kubernetes-admin@kubernetes**

**Vi cluster-role.yaml**

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: node-namespace-viewer-newcorpsite

rules:

- apiGroups: [""]

resources: ["nodes", "namespaces"]

verbs: ["get", "list", "watch"]

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: node-namespace-viewer-binding

subjects:

- kind: User

name: newcorpsite # Replace with your username or ServiceAccount

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: ClusterRole

name: node-namespace-viewer-newcorpsite

apiGroup: rbac.authorization.k8s.io

2. point output be like





