**Step 1**: **Backing up the Etcd cluster data**

• Use the following command to install **etcd-client**:

**sudo apt install etcd-client**

A screenshot of a computer

Description automatically generated

• Run the following command to list all the Pods with the **kube-system** namespace:

**kubectl get pods -n kube-system**

A screenshot of a computer

Description automatically generated

• Use the following command to get detailed information on the **etcd**:

**kubectl describe pods etcd-master.example.com -n kube-system**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

• Export the **advertise-client-url** to **advertise\_url** using the following command:

**export advertise\_url=https://172.31.1.43:2379**

**echo $advertise\_url**

**A screenshot of a computer

Description automatically generated**

**• Use the following command to save the etcd backup:**

**sudo ETCDCTL\_API=3 etcdctl \**

**--endpoints $advertise\_url \**

**--cacert /etc/kubernetes/pki/etcd/ca.crt \**

**--key /etc/kubernetes/pki/etcd/server.key \**

**--cert /etc/kubernetes/pki/etcd/server.crt snapshot save /tmp/mybackup**

A screenshot of a computer

Description automatically generated

**ls /tmp/**

A screenshot of a computer

Description automatically generated

**Step 2: Creating and verifying namespaces**

• Create a namespace called **cep-project2** using the following command:

**kubectl create namespace cep-project2**

**kubectl get namespace**

A screenshot of a computer

Description automatically generated

• Start a web service in the **cep-project2** namespace by using the following command:

**kubectl run web --namespace=cep-project2 --image=nginx --labels="app=web" --expose --port=80**

A black screen with white text

Description automatically generated

• Create **deny-from-other-namespaces.yaml** file:

A screenshot of a computer

Description automatically generated

• To create a network policy with namespace, **deny-from-other-namespaces**, run the following command:

**kubectl apply -f deny-from-other-namespaces.yaml**

A screenshot of a computer

Description automatically generated

• To verify the applied network policy, create a new namespace called **cep-project3** using the following commands:

**kubectl create namespace cep-project3**

**kubectl get ns**

A screenshot of a computer program

Description automatically generated

• Query this web service with the namespace **cep-project3** using the below command:

**kubectl run test-$RANDOM --namespace=cep-project3 --rm -i -t --image=alpine -- sh wget -qO- --timeout=2** [**http://web.default**](http://web.default)

A screenshot of a computer

Description automatically generated

**Step 3: Generating a certificate and private key in the worker node**

• Create a directory **role** and navigate to it.

**mkdir role**

**cd role**

A screenshot of a computer

Description automatically generated

• Run the following command which generates an RSA private key:

**sudo openssl genrsa -out user4.key 2048**

• Generate certificate requests using the following command:

**sudo openssl req -new -key user4.key -out user4.csr**

**Note**: Organization name: **namespace**

and common name: **user4**

A computer screen shot of a computer screen

Description automatically generated

• Run the following command to bind an identity to a private key using a digital signature:

**sudo openssl x509 -req -in user4.csr -CA /etc/kubernetes/pki/ca.crt -CAkey /etc/kubernetes/pki/ca.key -CAcreateserial -out user4.crt -days 500**

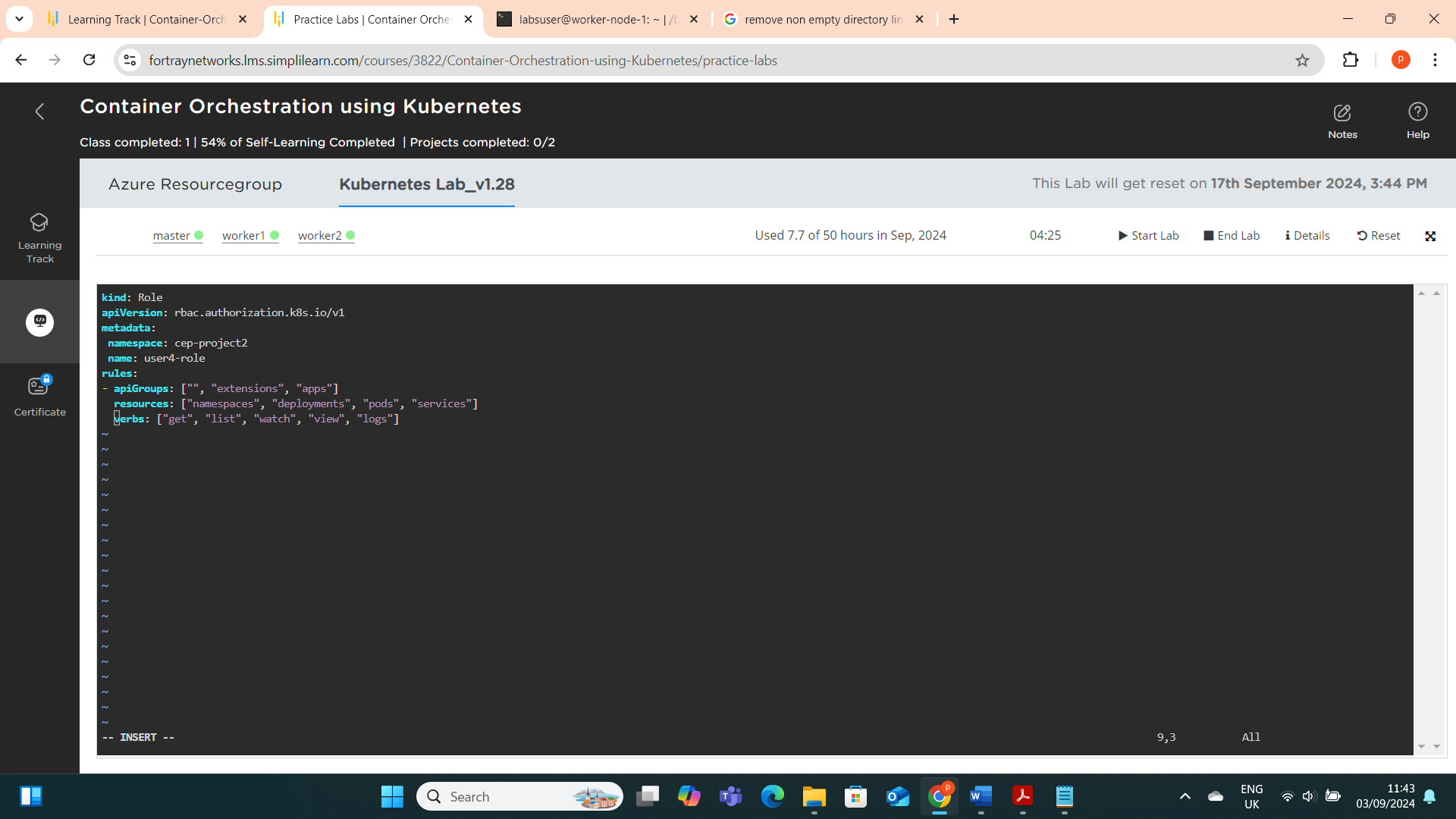
A screenshot of a computer

Description automatically generated

• Write the following code in the **role.yaml** file to create a role:

Vi role.yaml

This configuration will limit user4's access permissions.



• Create a role by using the following command:

**kubectl create -f role.yaml**

**kubectl get roles -n cep-project2**

A screen shot of a computer

Description automatically generated

• Write the following code in the **rolebinding.yaml** file to create a role binding:

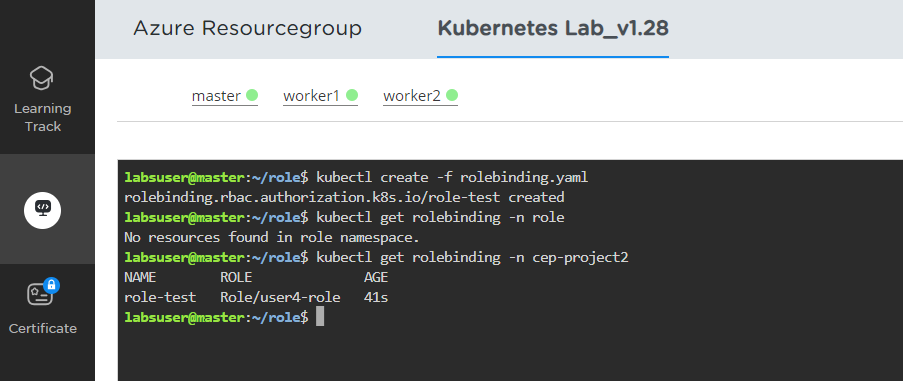
A screenshot of a computer

Description automatically generated

• Create a **rolebinding** using the following commands:

**kubectl create -f rolebinding.yaml**

**kubectl get rolebinding -n cep-project2**

****

**• Set credentials to user4 using the below command:**

**kubectl config set-credentials user4 --client-certificate=/home/labsuser/role/user4.crt --client-key=/home/labsuser/role/user4.key**

**A screen shot of a computer

Description automatically generated**

**• Set context to the user using the below command:**

**kubectl config set-context user4-context --cluster=kubernetes --namespace=cep-project2 --user=user4**

**A screenshot of a computer

Description automatically generated**

**• Show the current context using the below command:**

**kubectl config get-contexts**

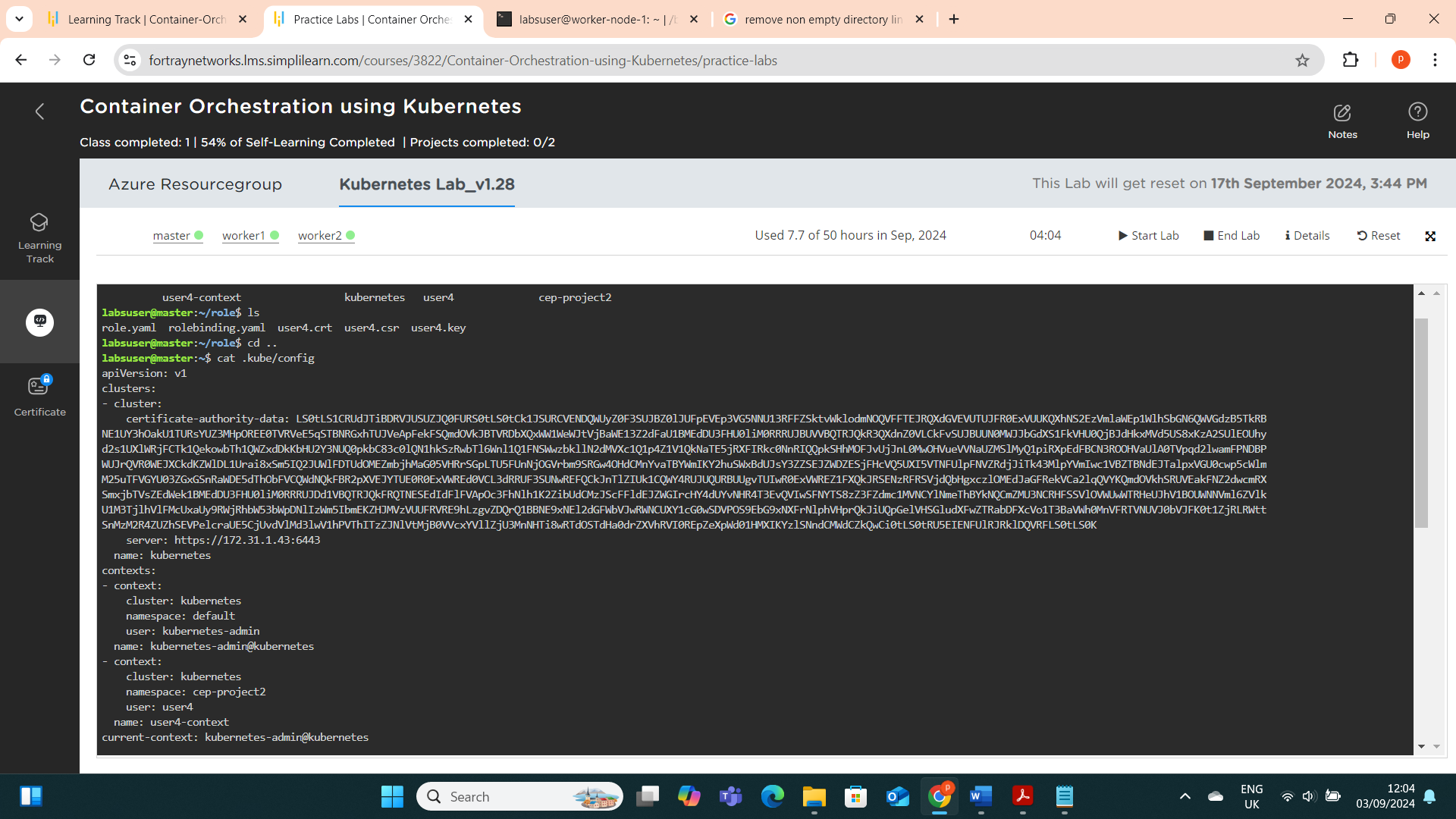
**A screenshot of a computer

Description automatically generated**

**Navigate to the home directory and view the config file**

**cd ..**

**cat .kube/config**

****

**Copy the $HOME/.kube/config file from the master node to the worker node.**

**A screenshot of a computer screen

Description automatically generated**

**Copy the user4.crt and user4.key files from the master node to the worker node in the /role directory**

**A screenshot of a computer

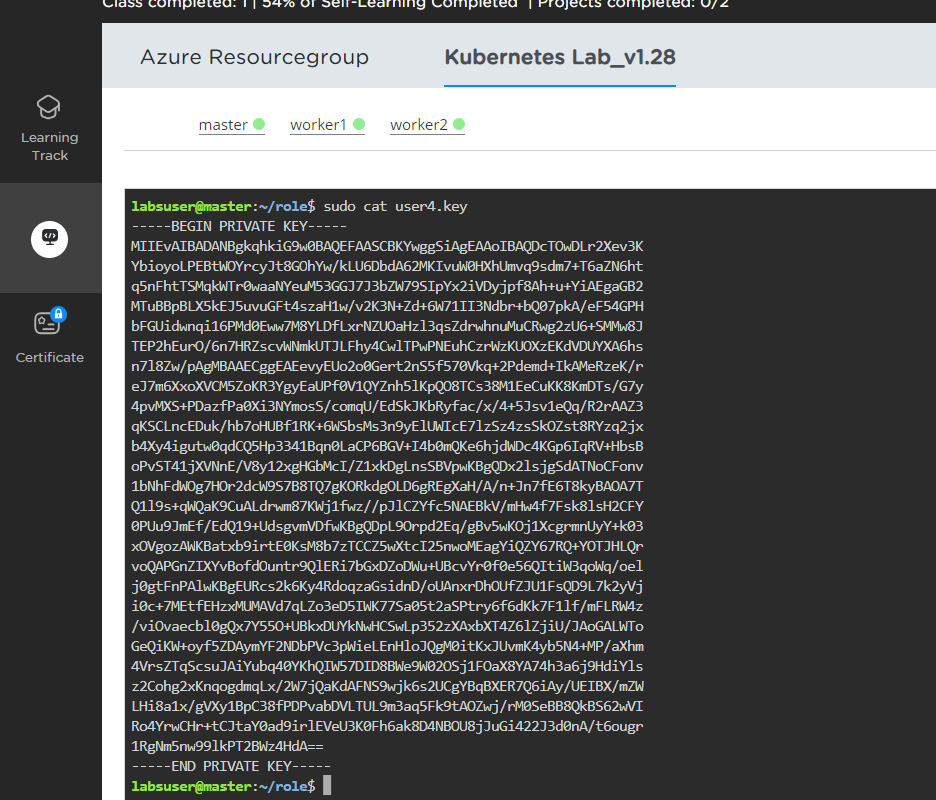
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

****

**A screenshot of a computer

Description automatically generated**

Also current context must be changed to **user4-context** in the myconfig as indicated in the below screenshot.

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

Run the following commands to see if the roles we created have just view access enabled:

**kubectl get pods --kubeconfig=myconfig**

To ensure that all authentication and RBAC protocols have been completed successfully, try listing nodes in the worker node:

**kubectl get nodes --kubeconfig=myconfig**

**A screenshot of a computer

Description automatically generated**

As you can see, user4 does not have access to list the cluster nodes.

**Step 4: Upgrading the Kubernetes cluster with the latest version**

This step is not required because the lab is already up to date.

A screenshot of a computer

Description automatically generated