# **Lab x: Provisioning Certificate Authority and generating Client Certs**

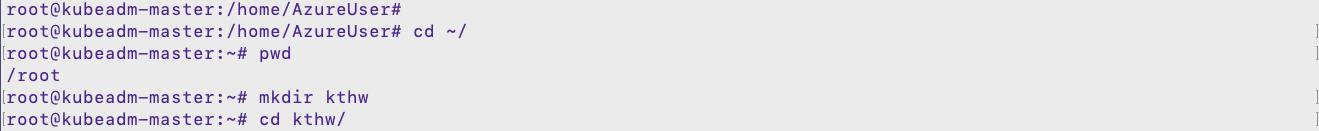
## **Provisioning certificate Authority**

1. In order to generate the certificates needed by Kubernetes, you must first provision a certificate authority
2. Create directory to store the certificates

$ cd ~/

mkdir kthw

cd kthw/



1. cfssljson and cfssl will need to be installed

$ sudo curl -s -L -o /bin/cfssl https://pkg.cfssl.org/R1.2/cfssl\_linux-amd64

$ sudo curl -s -L -o /bin/cfssljson https://pkg.cfssl.org/R1.2/cfssljson\_linux-amd64

$ sudo curl -s -L -o /bin/cfssl-certinfo https://pkg.cfssl.org/R1.2/cfssl-certinfo\_linux-amd64

$ sudo chmod +x /bin/cfssl\*



1. Use this command to generate the certificate authority. Include the opening and closing curly braces to run this entire block as a single command.

A picture containing bird

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## **Generating Client Certificate**

1. Admin Client certificate:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

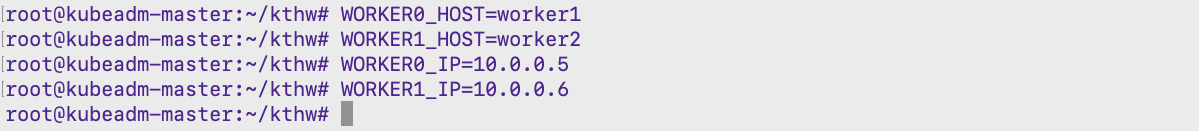
1. Kubelet Client certificates. Be sure to enter your actual cloud server values for all four of the variables at the top:

$ WORKER0\_HOST=<Public hostname of your first worker node cloud server>

$ WORKER0\_IP=<Private IP of your first worker node cloud server>

$ WORKER1\_HOST=<Public hostname of your second worker node cloud server>

$ WORKER1\_IP=<Private IP of your second worker node cloud server>



A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Controller Manager Client certificate:

A screenshot of a social media post

Description automatically generated

1. Kube Proxy Client certificate:

A screenshot of a cell phone

Description automatically generated

1. Kube Scheduler Client Certificate:

A screenshot of a social media post

Description automatically generated

$ ls -lrt

A close up of text on a white background

Description automatically generated

## **Generate server certificate for the Kubernetes API Server**

1. We will generate one server certificate, signed with all of the hostnames and IPs that may be used later in order to access the Kubernetes API

$ CERT\_HOSTNAME=10.38.0.1,<controller node 1 Private IP>,<controller node 1 hostname>127.0.0.1,localhost,kubernetes.default



A screenshot of a cell phone

Description automatically generated

A close up of text on a screen

Description automatically generated

## **Generate a certificate that will be used as that key-pair**

1. Kubernetes provides the ability for service accounts to authenticate using tokens. It uses a key-pair to provide signatures for those tokens. A certificate is ready to be used as a service account key-pair: service-account-key.pem and service-account.pem.

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## **Move the files onto the appropriate servers**

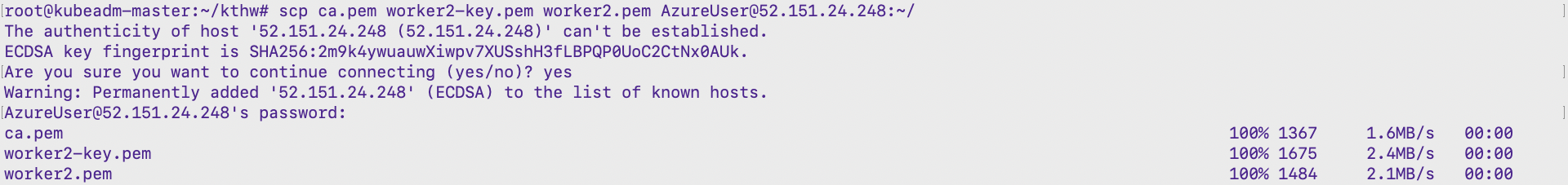
1. Move certificate files to the worker nodes:

$ scp ca.pem worker1-key.pem worker1.pem [AzureUser@20.36.7.180:~/](mailto:AzureUser@20.36.7.180:~/)

A screenshot of a cell phone

Description automatically generated

$ scp ca.pem worker2-key.pem worker2.pem [AzureUser@52.151.24.248:~/](mailto:AzureUser@52.151.24.248:~/)



1. Move certificate files to the controller nodes:

$ scp ca.pem ca-key.pem kubernetes-key.pem kubernetes.pem \

> service-account-key.pem service-account.pem [AzureUser@20.191.95.249:~/](mailto:AzureUser@20.191.95.249:~/)

A screenshot of a cell phone

Description automatically generated