Cloud Provider:AWS
Server:Ubuntu 18.04 x6
Server List: 2 Controller(Master), 2(Worker), 1 Nginx(Reverse Proxy), 1 Ubuntu 18.04(local)

Kubeconfigs can be generated using kubectl:

- Use kubectl config **set-cluster** to set up the configuration for the location of the cluster.
- Use kubectl config **set-credentials** to set the username and client certificate that will be used to authenticate
- Use kubectl config **set-context** default to set up the default context.
- Use kubectl config **use-context** default to set the current context to the configuration provided

Kubeconfig files for various components of the Kubernetes cluster

- Kubelet(one for each worker node)
- Kube-proxy
- Kube-controller-manager
- Kube-scheduler
- Admin

Generating Kubeconfigs for the Cluster

// Create an environment variable to store the address of the Kubernetes API, and set it to the private IP of your load balancer cloud server

```
$ KUBERNETES_ADDRESS=<load balancer private ip>
```

// Generate a kubelet kubeconfig for each worker node

```
for instance in <worker 1 hostname> <worker 2 hostname>; do
  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.pem \
    --embed-certs=true \
    --server=https://${KUBERNETES_ADDRESS}:6443 \
    --kubeconfig=${instance}.kubeconfig
  kubectl config set-credentials system:node:${instance} \
    --client-certificate=${instance}.pem \
    --client-key=${instance}-key.pem \
    --embed-certs=true \
    --kubeconfig=${instance}.kubeconfig
  kubectl config set-context default \
    --cluster=kubernetes-the-hard-way \
    --user=system:node:${instance} \
    --kubeconfig=${instance}.kubeconfig
  kubectl config use-context default --kubeconfig=${instance}.kubeconfig
done
```

// Generate a kube-proxy kubeconfig

```
{
  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.pem \
    --embed-certs=true \
    --server=https://${KUBERNETES_ADDRESS}:6443 \
    --kubeconfig=kube-proxy.kubeconfig
  kubectl config set-credentials system:kube-proxy \
    --client-certificate=kube-proxy.pem \
    --client-key=kube-proxy-key.pem \
    --embed-certs=true \
    --kubeconfig=kube-proxy.kubeconfig
  kubectl config set-context default \
    --cluster=kubernetes-the-hard-way \
    --user=system:kube-proxy \
    --kubeconfig=kube-proxy.kubeconfig

  kubectl config use-context default --kubeconfig=kube-proxy.kubeconfig
}
```

//Generate a kube-controller-manager kubeconfig

```
{
  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.pem \
    --embed-certs=true \
    --server=https://127.0.0.1:6443 \
    --kubeconfig=kube-controller-manager.kubeconfig

  kubectl config set-credentials system:kube-controller-manager \
    --client-certificate=kube-controller-manager.pem \
    --client-key=kube-controller-manager-key.pem \
    --embed-certs=true \
    --kubeconfig=kube-controller-manager.kubeconfig

  kubectl config set-context default \
    --cluster=kubernetes-the-hard-way \
    --user=system:kube-controller-manager \
    --kubeconfig=kube-controller-manager.kubeconfig

  kubectl config use-context default --kubeconfig=kube-controller-manager.kubeconfig
}
```

//Generate a kube-scheduler kubeconfig

```
{
  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.pem \
    --embed-certs=true \
    --server=https://127.0.0.1:6443 \
    --kubeconfig=kube-scheduler.kubeconfig

  kubectl config set-credentials system:kube-scheduler \
    --client-certificate=kube-scheduler.pem \
    --client-key=kube-scheduler-key.pem \
    --embed-certs=true \
    --kubeconfig=kube-scheduler.kubeconfig

  kubectl config set-context default \
    --cluster=kubernetes-the-hard-way \
    --user=system:kube-scheduler \
    --kubeconfig=kube-scheduler.kubeconfig

  kubectl config use-context default --kubeconfig=kube-scheduler.kubeconfig
}
```

//Generate an admin kubeconfig

```
{
  kubectl config set-cluster kubernetes-the-hard-way \
    --certificate-authority=ca.pem \
    --embed-certs=true \
    --server=https://127.0.0.1:6443 \
    --kubeconfig=admin.kubeconfig

  kubectl config set-credentials admin \
    --client-certificate=admin.pem \
    --client-key=admin-key.pem \
    --embed-certs=true \
    --kubeconfig=admin.kubeconfig

  kubectl config set-context default \
    --cluster=kubernetes-the-hard-way \
    --user=admin \
    --kubeconfig=admin.kubeconfig

  kubectl config use-context default --kubeconfig=admin.kubeconfig
}
```

Distributing the Kubeconfig Files

//Move kubeconfig files to the worker nodes

```
$ scp <worker 1 hostname>.kubeconfig kube-proxy.kubeconfig cloud_user@<worker 1 public IP>:~/
```

```
$ scp <worker 2 hostname>.kubeconfig kube-proxy.kubeconfig cloud_user@<worker 2 public IP>:~/
```

//Move kubeconfig files to the controller nodes

```
$ scp admin.kubeconfig kube-controller-manager.kubeconfig kube-scheduler.kubeconfig cloud_user@<controller 1 public IP>:~/
```

```
$ scp admin.kubeconfig kube-controller-manager.kubeconfig kube-scheduler.kubeconfig cloud_user@<controller 2 public IP>:~/
```