**Kubernetes Cluster Setup: Use Minikube to create a local Kubernetes cluster. Document the initiation process. Detail any specific configurations applied to the cluster**

**Server Specification**

* **Centos 7 (2cpu 4GB ram 20GB storage )**

**Configure Instance**

* **sudo yum -y update**
* **Install a virtualization software**

Since we are going to run the single node cluster inside a virtual machine, we need to set up a virtualization software.

sudo yum -y install epel-release

sudo yum -y install libvirt qemu-kvm virt-install virt-top libguestfs-tools bridge-utils

* **Then, start and enable the libvirtd service**:

sudo systemctl start libvirtd

sudo systemctl enable libvirtd

* **Confirm the virtualization service is running with the command**:

systemctl status libvirtd

* **Next, add your user to the libvirt group**:

sudo usermod -a -G libvirt $(whoami)

* **Then, open the configuration file of the virtualization service:**

sudo vi /etc/libvirt/libvirtd.conf

sudo vi /etc/libvirt/libvirtd.conf

un-comment the below lines

unix\_sock\_group = "libvirt"

unix\_sock\_rw\_perms = "0770"

sudo systemctl restart libvirtd.service

**Installing minikube**

Download the Minikube binary package using the **wget** command:

wget <https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64>

* Then, use the [chmod command](https://phoenixnap.com/kb/linux-file-permissions" \t "_blank) to give the file executive permission:

chmod +x minikube-linux-amd64

* Finally, move the file to the **/usr/local/bin** directory:

sudo mv minikube-linux-amd64 /usr/local/bin/minikube

* Verify the installation by checking the version of the software:

minikube version

**Installing container run time**

* **Setup the repository**

sudo yum install -y yum-utils

sudo yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

* **Install docker engine**

sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo systemctl start docker

**Intalling Kubectl**

We need to set up kubectl, the command line tool for working with Kubernetes.

* Run the following command to download kubectl:

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s [https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl](https://storage.googleapis.com/kubernetes-release/release/stable.txt%60/bin/linux/amd64/kubectl)

* **Give it executive permission**

chmod +x kubectl

* **Move it to the same directory where you previously stored Minikube**:

sudo mv kubectl /usr/local/bin/

* **Verify the installation by running**:

kubectl version --client -o json

**Starting Minikube**

* **We need to install conntrack in root path**

sudo yum install conntrack-tools

* **start minikube using below command**

minikube start --force

**Install an Nginx ingress controller using Helm to expedite the process**

**Installing helm**

* **Download helm tr package from the below URL**

curl -o helm-v3.10.3-linux-amd64.tar.gz <https://get.helm.sh/helm-v3.10.3-linux-amd64.tar.gz>

* **Change the permission and unpack the tar file**

chmod 777 helm-v3.10.3-linux-amd64.tar.gz  
tar -zxvf helm-v3.10.3-linux-amd64.tar.gz

* **move the helm file to /us/local/bin**

mv helm /usr/local/bin

* **Check the helm version**

helm version

**Installing nginx ingress controller**

* **Add the helm repo**

helm repo add ingress-nginx <https://kubernetes.github.io/ingress-nginx>

* **Update the helm repo**

helm repo update

* **Install nginx ingress controller**

helm install my-ingress-nginx ingress-nginx/ingress-nginx --set controller.admissionWebhooks.enabled=false

**Sample Application Deployment:**

**Deploy a "Hello World" application using a pre-built Docker image.**

**Create a Kubernetes manifest for deployment, ensuring the application is accessible via the ingress.**

* **Start Docker**

sudo systemctl start docker

* **Run hello world docker image using below image**

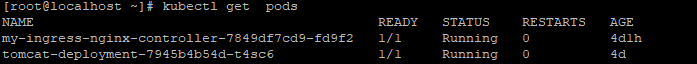
sudo docker run hello-world

**Screen Shots**

**Deploying Mini Kube**



**Deploying sample application**



**Deploying ingress using helm**

