**PROJECT DOCUMENTATION OF WORKFLOW, STEPS AND PROCEDURES**

**SETTING UP OUR KUBERNETES CLUSTER**

PC requirements for setup of our kubernetes cluster:

* Docker Installation
* Kubectl
* Rancher Kubernetes Engine
* Helm Installation

**Docker Installation Steps:**

Install docker using the repository in your Ubuntu machine.

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Step.1

Update the apt package index and install packages to allow apt to use a repository over HTTPS:

$ sudo apt-get update

$ sudo apt-get install \

ca-certificates \

curl \

gnupg \

lsb-release

Step 2.

Add Docker’s official GPG key

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

Step 3.

Add the stable repository using the following command

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Step 4

Install the Docker enginer

Update the apt package index, and install the latest version of Docker Engine and containerd

$ sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

***Then to be able to use docker without sudo, create a new user and add to docker group***

***Using these commands***

Add new User

sudo adduser devops\_engineer01

**# Fill in your password and other personal info**

sudo usermod -aG sudo devops\_engineer01

su - devops\_engineer01

**# Enter your password to login for the user created and then Add User to the Docker Group**

sudo groupadd docker

sudo usermod -aG docker $USER

- Re-Login or Restart the Server

Login to the newly created User

su - devops\_engineer01

**# enter password**

**Test without using sudo**

docker images

**Kubectl Installation Steps**

*Installing kubectl binary with curl on Linux*

Step 1.

Download the latest release with the command:

curl -LO "https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl"

Step 2.

Validate the binary (optional)

Download the kubectl checksum file:

curl -LO "https://dl.k8s.io/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl.sha256"

Validate the kubectl binary against the checksum file and make sure it outputs “kubectl: ok”

echo "**$(**<kubectl.sha256**)** kubectl" | sha256sum --check

Then install  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

To confirm installation:

kubectl

**Rancher Kubernetes Engine Installation Steps:**

wget -O rke\_linux-amd64 <https://github.com/rancher/rke/releases/download/v1.3.6/rke_linux-amd64>

Then rename the downloaded file to rke

mv rke\_linux-amd64 rke

chmod +x rke

sudo mv rke /*usr/*local/bin

Test the installation

rke --version

**# Turn swap off**

ssh-keygen -t rsa

cat ~/.ssh/id\_rsa

sudo vi /etc/ssh/sshd\_config

sudo service ssh restart

my\_addresses=$(hostname -I); read -a my\_address\_list <<< $my\_addresses; my\_address=${my\_address\_list[0]}; echo $my\_address

my\_addresses=$(hostname -I); read -a my\_address\_list <<< $my\_addresses; echo $my\_addresses

Sample addresses are:

192.168.44.247

172.17.0.1

Add the first one to your cluster.yml file as ur nodes server(Master, Worker and ETCD)

sudo apt-get install openssh-client

sudo apt-get purge openssh-server

sudo apt-get install openssh-server

<!-- ssh-copy-id -i ~/.ssh/id\_rsa.pub cybernetor066@192.168.49.1 -->

ssh-copy-id -i ~/.ssh/id\_rsa.pub devops\_engineer01@192.168.44.247

You can display the octal permissions for a file using the stat command:

stat -c %a [filename]

**# create configuration for our cluster**

sudo swapoff -a

rke config --name cluster.yml

rke up

**Then connect kubectl with cluster**

You can copy this file to

$HOME/.kube/config or, if you are working with multiple Kubernetes clusters, set the

KUBECONFIG environmental variable to the path of the config file

mkdir -p $HOME/.kube

sudo chown $(id -u):$(id -g) $HOME/.kube/config

cat kube\_config\_cluster.yml > $HOME/.kube/config

**Test kubectl connection**

kubectl get nodes

kubectl get deployments –all-namespaces

**Then set up the kubernetes dashboard for monitoring purposes**

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.4.0/aio/deploy/recommended.yaml

kubectl apply -f https://raw.githubusercontent.com/cybernetor066/nextflow\_api\_project/master/manifests/rke\_setup\_files/dashboard-adminuser.yaml

kubectl -n kubernetes-dashboard get secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{.secrets[0].name}") -o go-template="{{.data.token | base64decode}}"

#################################################

sudo apt install firewalld

sudo firewall-cmd --add-port=8001/tcp

nohup kubectl port-forward --address 0.0.0.0 services/kubernetes-dashboard -n kubernetes-dashboard 8001:443 &

Then access the dashboard remotely

http://<your-public-ip>:8001

curl https://64.227.106.166:8001/#/login

pgrep -a kubectl

kill <id>

#################################################

**Helm Installation Steps**

*Installing helm through package managers:*

curl https://*baltocdn.com*/helm/signing.asc | sudo apt-key add -

sudo apt-get install apt-transport-https --yes

echo "deb https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

sudo apt-get update

sudo apt-get install helm

Then test helm installation:

helm version