1. **K8s install**

Ways to install k8s

Kubeadm

Minikube

Kops

K8s in gcp

Step1:

sudo su

step1 - On Master

sudo apt-get update

sudo apt-get install docker.io -y

service docker restart

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

Step1 - on workers

apt-get update

apt-get install docker.io -y

service docker restart

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" >/etc/apt/sources.list.d/kubernetes.list

apt-get update

apt install kubeadm=1.20.0-00 kubectl=1.20.0-00 kubelet=1.20.0-00 -y

systemctl enable kubelet.service

Step2:

On Master:

kubeadm init --pod-network-cidr=192.168.0.0/16

>Copy the token and paste it into the worker node.

Step3:

On Master:

exit

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

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

step4:

On Master:

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

\*\*\*\*new\*\*\*\*\*

kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml

kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml

watch kubectl get pods -n calico-system

\*\*\*\*\*\*\*\*\*\*\*\*

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml

when you r integerating sometting use this.

sudo usermod -aG docker jenkins

sudo service jenkins restart

Our Kubernetes installation and configuration are complete

--------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**kubectl get pod**

kubectl get pods --all-namespaces

\*\*\*\*\*if pods are not running in single cluster use below command \*\*\*\*\*

kubectl taint nodes --all node-role.kubernetes.io/master-

kubectl apply -f mongo-config.yaml

kubectl apply -f secret.yaml

kubectl apply -f mongo-app.yaml

kubectl apply -f web-app.yaml

kubectl get pod

kubectl get configmap

kubectl get secret

kubectl get svc

minikube ip

kubectl get pod -o wide

kubectl get node -o wide

kubectl get replicaset

kubectl get ns

minikube service webapp-service

++++++++++++++++++++++++++++++++++++++

kubectl delete deployment --all

kubectl delete secret –all

minikube delete

git init

git clone https://github.com/Baga114/kubernetes-ingress.git

cd kubernetes-ingress/deployments/

kubectl apply -f common/ns-and-sa.yaml

kubectl apply -f common/default-server-secret.yaml

kubectl apply -f common/nginx-config.yaml

kubectl apply -f rbac/rbac.yaml

kubectl apply -f deployment/nginx-ingress.yaml

kubectl get all -n nginx-ingress

kubectl expose deploy nginx-deploy-main --port 80

\*\*\*\*\*\*helm chart\*\*\*\*\*\*

curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null

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

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] 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

9 helm --version

13 helm create first

s

17 sudo apt install tree

18 tree first

24 cd first

25 ls

26 sudo nano values.yaml

27 ls

28 cd

31 helm install myhelloworld first

32 cd

34 kubectl get pods

35 kubectl get svc

\*\*\*\*\*After changing you can use this command

58 helm upgrade myhelloworld first

1. Installing Ansible

sudo apt-add-repository -y ppa:ansible/ansible

sudo apt-get update

sudo apt-get install -y ansible

ansible push type configruation management tool.

\*\*\*\*\*master\*\*\*\*\*\*\*

sudo apt-get update

sudo apt-get install python3 -y

sudo apt install software-properties-common -y

sudo apt-add-repository ppa:ansible/ansible

sudo apt update

sudo apt install ansible -y

\*\*\*\*\*\*\*\*keyless ssh\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*do it on master\*\*\*\*\*\*\*\*\*\*

ssh-keygen

cd .ssh

cat id\_rsa.pub

copy the file and go to slave

sudo nano .ssh/authorized\_key

\*\*\*\*\*\*\*\*on hosts/slave/agent\*\*\*\*\*\*\*\*\*\*

sudo apt-get update

sudo apt-get install python3

sudo nano .ssh/authorized\_keys

\*\*\*\*\*master\*\*\*\*\*

do ssh ubuntu@<ip address of slave>

now you are in salve to exit use exit command

\*\*to connect to slave use below method - \*\*

make group and add ip address of that group there for example

[baga]

\*\*you can also give name to slave\*\*

slave1 ansible\_ssh\_host=<id address of slave>

\*\*to check the connection between master and slave use below command\*\*

ansible -m ping all <or in the place all you can use the group name>

\*\*if it shows pong means its is connected\*\*

\*\*now make play book in master\*\*

\*\*first make directory\*\*

mkdir ansible

\*\*create play book\*\*

sudo nano playbook.yaml

\*\*start wrting\*\*

---

- hosts: slave1

become: yes

name: play1

tasks:

- name: install apache2

apt: name=apache2 state=latest

\*\*to check futher it is working or not use blow command\*\*

ansible -m ping slave1

\*\*to run this play book use \*\*

\*\*ansible-playbook <name of playbook>\*\*

ansible-playbook playbook.yaml

and use also add script file

\*\*\*\*now time for anisble role\*\*\*\*\*\*\*\*

\*\*go inside\*\*

cd /etc/ansible/roles

\*\*install tree\*

sudo apt install tree

\*\*now create the role\*\*

sudo ansible-galaxy init <role name> --offline

sudo apt install tree

\*\*type tree <role name> ro see the structure in tree format\*\*

tree <role name>

tree will appare

\*\*now go inside the role using role name for example\*\*

cd webpage

\*\*now go inside task from tree\*

cd tasks

\*\*there will be a yml file open that file\*\*

sudo nano main.yml

\*\*spcefied the task that you want to perform in that file\*\*

\*\*for exapmle\*\*

-include:install.yml

-include:configure.yml

-include:service.yml

\*\*now create all above file\*\*

sudo nano install.yml

\*\*inside file\*\*

---

- name:install Apache2

apt: name=apache2 update\_cache=yes state=latest

\*\*now\*\*

sudo nano configure.yml

\*\*inside file\*\*

---

- name: configure website

copy: src=index.html test=/var/www/html

\*\*now service\*\*

sudo nano service.yml

\*\*inside\*\*

---

- name: starting apache2 service

service: name=apache2 state=started

\*\*now specife website go to files from tree and type below cc\*

sudo nano index.html

\*\*now go back to /etc/ansible and create file there\*\*

sudo nano site.yaml

\*\*inside this\*\*

---

- hosts: servers

roles:

- webpage

\*\*now run the file\*\*

sudo ansible-playbook site.yaml --syntax-check

\*\*now run the play book\*\*

sudo ansible-playbook site.yaml

1. Installing Jenkins

sudo apt-get update

sudo apt-get install docker.io -y

sudo apt install ca-certificates

sudo apt-get install openjdk-11-jdk -y

wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins –y

1. Installing database

install database on machine.

systeml enable <name of datatype>

systeml start <name of datatype>

sudo su

mysql -h <hostname> -p 3306 -u admin -p

show databases;

CREATE DATABAES <name of database>

use <databases>;

CREATE TABLE employees (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

);

INSERT INTO employees values(1,'sam');

SHOW TABLES;

SECELT\*FROM employees;

1. Docker installation

docker all commands:-

figlet

uname -a

interactive and deattached mode

1. install docker in linux -

sudo apt-get update

sudo apt-get install docker.io

1. (a) for installing docker composer for microservice of file.

sudo apt install -y python3 python3-pip

sudo pip3 install docker-compose

docker-compose -v

2. check version- docker --version

3. download or use the image for example-

sudo docker pull (system name) i am using ubuntu here so it will be

sudo docker pull ubuntu

4. to see the download or all images use -

sudo docker images

5. This command helps to run the container, by using image name

docker run -it <name of image> (intractive mode)

sudo docker container run -it -d <name of image>

docker run -dt <name of image> (detached mode )

\*\*\*\*to change the name\*\*\*\*\*

docker run --name baga -dt -p 8080:8080 ubuntu

6. this command helps to listing all the ccontainer that are running (like ls)

sudo docker ps

7. this command is use to stop the conainer

sudo docker stop (container no or name)

use can also use kill to kill the contaier is stop is not working.

finally to delete the the container use sudo docker rm <conatiner id/no>

and you cant see that delete container by using rm command.

use can also use sudo docker rm -f <container id> if you are in hurry

8. this command is used to check the conatiner that are running currently

sudo docker ps

\*\*\*to build \*\*\*

sudo docker build . -t <file name>

9. this command is used to check the all container futher it is running or stop.

sudo docker ps -a

10. to get inside the container for using or doing whatever you want use below coomand

sudo docker exec -it <container id/no> bash

now you are in the container system is totaily alter in container

you can use exit command to come out from the system

11. want to remove the image from the system use -

sudo docker rmi <image id>

use ll for check all the directory.

12. we can create new image by using existing conatiner is command is

sudo docker commit <c - id> <dockerhub userid + name> that way we can

create the image of the container or custom container

sudo docker rm -f <c-id>

13. to remove all container use below command

sudo docker rm -f $(sudo docker ps -a -q)

\*\*\*\*\*\*for porting running\*\*\*\*\*\*\*\*\*\*\*

sudo docker run -it -p 82:80 -d usernameofdocerhub/name that you gave to image

14. sudo usermood -aG docker &user to only use docker not sudo with it

also used tag - docker tag <name of image><docker repo/name of image>

15. sudo docker push <name of file or folder>

16. to create a storage volue for docker use below command

docker volume create <name of value >

dicker run -dt -v <name of volume>:<folder> <name of image>

17. sudo docker run -it --mount source=vol1,destination=/app -d ubuntu

above command used for mount the volume to container , meaning of source is file name

,destination where and which name you want to store that file.

18. we use echo to create file inside the container for out side we use nano

19. multiple container can have same storage value

20. bind mount is also a storage, however, in this storage we have to specify

where this will host totally alter from deckor value in which we dont need to

warry about host or where this file will locate

below is command to mount bind mounting

sudo docker run -it -v /home/ubuntu/<folder or directory which you create use that name here>:/demo -d ubuntu

21. for easy of work we can link multi conatiner among

each other command is below

sudo docker run -it --<name of container> --link <name-of-conatiner> -d <image-name>

for linking use - sudo docker run -it --name container2 --link container1 -d ubuntu

for checking the link or quanity of coantiner use cat /etc/hosts

for interacting each other used ping <conatiner name whom you want to link>

22. for installing docker composer for microservice of file.

sudo apt install -y python3 python3-pip

sudo pip3 install docker-compose

docker-compose -v

sudo chmod +x /usr/local/bin/docker-compose

docker-compose up -d

docker-compose down

23. ymal files -

version: '3'

services:

sample1:

image: 'httpd'

ports:

- "80:80"

sample2:

image: 'nginx'

make directory in directory give file name docker-compose.yml and then paste above code.

then use command / sudo docker-compose up -d /

your container will start

if want to delete file use / docker-compose down /

i am geeting the issue when i am changing the network.

24. swarn has advantage over compose, we use compose for just demo purpose but for real application we use swarn.

sudo docker swarm init --advertise-addr=<ip address of master server>

25. $ docker service create –name <name-of-service> --replicas <number-of-replicas> --publish <portmapping>

<image-name>swarn

docker swarm join --token SWMTKN-1-3v5vl3k4ci1dsyse4l6xrye9lkwykdzjf7kszg3twsjub3jj31-ats2wg0xun88vof83qgzsyz3b 107.20.52.47:2377

......mysql details

......to add use

mysql -h <endpoint of data> -u <username> -p

show databases;

Create database docker;

Use docker;

Create table emp(name varchar(20), phone varchar(20));

.....suppose you have created emp data to check that use below command

select \* from emp;

/home/ubuntu/ci-cd-pipeline/git141

1. Linux command

For Practice on Onworks ( https://www.onworks.net/ )

Root password :- 123456

Link :- https://www.onworks.net/runos/create-os.html

:- https://www.onworks.net/os-distributions/ubuntu-based/free-ubuntu-online-version-20

Step 1 :- Click on run machine

Step 2 :- Click on the 9 dots bottom- left of the screen

Step 3 :- In the search bar type terminal and open it

Step 4 :- Now, you can start practicing

Shortcuts

To copy any text in terminal just select it

To paste any text just right click

To get the file name automatically write initial letter and press tab

Use above arrow key to use previous commands which you have used

Commands

clear :- To clear the terminal screen

mkdir <dirname> :- To create a directory

mkdir -p <dir>/<subdir>/<sub1dir> :- You can directory and sub directories inside them.

ls :- To get the list of files & directories

cd :- To change the directory or to directly go in home

Cd dir/dirA/dir1 :- TO directly go in dir1

cd .. :- To go back to the previous directory

cd ../../../ :- To directly go back to 3 directory

touch <filename> :-To create a empty file

cat <filename> :- To print the content of the file on terminal

cat > <filename> :- To modify the content of the file

cat >> <filename> :- TO add the content in the file

sudo nano <filename> :- To create and open the file

Ctrl + S :- TO save the file

Ctrl + x : TO exit the file

vi <filename> :- To create and open the file

Hit "i" :- To get in to the insert mode

esc :- TO exit the insert mode

:wq :- TO save & exit

:w :- To only save the file

:q :- To only exit the file

rm <filename> :- to remove the file

rm -r <filename> :- to remove the file / dir forcefully

rmdir <dirname> :- to remove the directory

whoami :- to know the current user

pwd :- to print the current working directory

history :- To check the history of previous commands which are used

1. Git command

Origin master

git commands

token - ghp\_aIolKBpO8qplCXtaIu5GzovDxtdHbD4MGumz

\*\*\*\*\*\*\*delete\*\*\*\*\*\*\*\*\*\*\*

vim file save :wq or if you dont want to save :q!

rm -rf \*

rm -fr .git

git branch | grep -v "master" | xargs git branch -D

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*fatal: detected dubious ownership in repository at '/var/www/html'

To add an exception for this directory, call:

git config --global --add safe.directory /var/www/html\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* if got above error use below command\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

git config --global --add safe.directory '\*'

\*\*some time if pull does'nt work use below command\*\*

git remote set-url origin <url>

\*\*\*\*\*\*use below method to add files\*\*\*\*\*\*

\*/var/www/html$\* \*\*this is main command\*\* sudo mv random\ web/\* .

1. git init -to entre file in git

2. git status - used to check the file status whether they staged or not

git log

3. git add - is used to stage the file you can write files name after git add command or just add . after command for all files tostage.

4. git commit -m "message what really yoy are commititng"

5. git remote add origin "url of your repositary"

git remote rm "url" - to remove repositary

git remote remove origin

6. git push origin master - to push your file to remote means to github. this will ask your github username and password.

7. git clone -to copy repository from remote to local for getting work done.

8. git pull origin master - to pull the file from hithub

to add branch for adding someting new to your develment files you create different branch to work so that legacy of master branch wont effect

note - you should be on derictory to make the new branch

special note you should have atleast 1 commit to add the branch or even check the status.

9. git branch <name of branch>- add new branch

10. git branch - to see how many branches are there

11. git branch -D <name of branch which you want to delete>

12. git checkout <name of branch> - is used to switch the brances

13. git log - - to see the history all command and repositary

git log --graph

git log --graph --pretty=oneline

14. git stash - used to save file for temorary to switch to other work

15. git stash pop - is used to get back that file that we haved temorary saved.

16. git revert - to go back to the the previous version in which you want

press q to get out from git revert command

17. git checkout <entre commit to check the status of your revert>

18. git diff <commit1> <commit2> - is used to see the differnce b/w two commit or files.

19. git merge <>- should only used for remote branch to merge two branch

git merge does not alter the history log

20. git rebase - to merge file in local repositary just for test purpose

21. git mergetool and then hit entre - used to slove the merge conflict

then there will be vi editer open to after (i is used to insert somthing in vi)doing change you can save the file by using :w quit by using :q

forking in github for copy someones code

then pull

sudo docker rm -f $(sudo docker ps -a -q)

sudo docker build /home/ubuntu/website-jenkins/workspace/develop test-job -t develop-test

sudo docker run -itd --name c1 develop-test