Docker

Docker subcommands :->

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04

attach Attach to a running container

build Build an image from a Dockerfile

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes on a container's filesystem

events Get real time events from the server

exec Run a command in a running container

export Export a container's filesystem as a tar archive

history Show the history of an image

images List images

import Import the contents from a tarball to create a filesystem image

info Display system-wide information

inspect Return low-level information on a container or image

kill Kill a running container

load Load an image from a tar archive or STDIN

login Log in to a Docker registry

logout Log out from a Docker registry

logs Fetch the logs of a container

network Manage Docker networks

pause Pause all processes within a container

port List port mappings or a specific mapping for the CONTAINER

ps List containers

pull Pull an image or a repository from a registry

push Push an image or a repository to a registry

rename Rename a container

restart Restart a container

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

save Save one or more images to a tar archive

search Search the Docker Hub for images

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop a running container

tag Tag an image into a repository

top Display the running processes of a container

unpause Unpause all processes within a container

update Update configuration of one or more containers

version Show the Docker version information

volume Manage Docker volumes

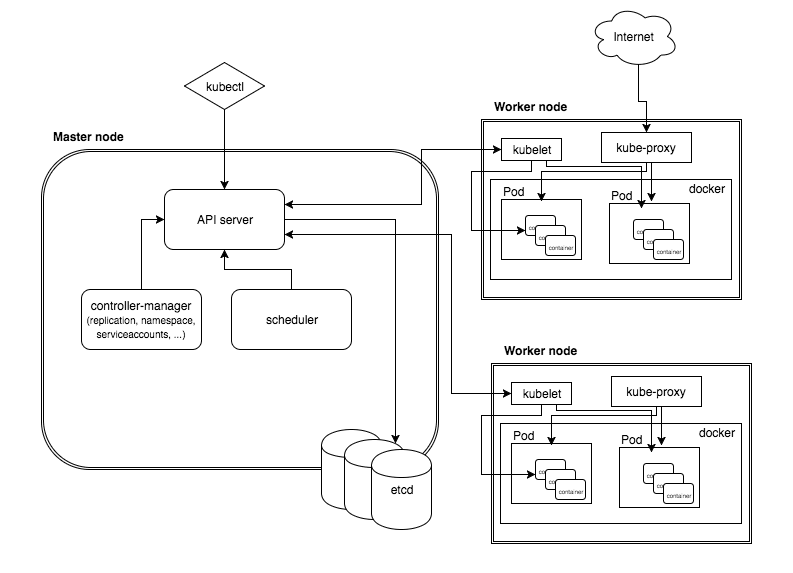
wait Block until a container stops, then print its exit code

Docker-compose

Docker-compose:-> compose from scratch

(https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-14-04)

Kubrnetes Notes



Architecture for kubrenetes

Kubectl:

Kubrenets is a orchestration ( arranges something in order carefully ) tool sponsored by google and redhat used for clustering docker environment in aws cloud It can support for azure but it is stable only for aws

We are doing this set up with namecheap.com

**prerequisites :**

must have account in aws && namecheap.com need to buy one domain

-- linux machine

-- create s3 bucket (keep in mind in which region u r creating )

--create iam user with full permissions ( policies )

--route53

Step1—create dns hostedzone with <subdomain><domainname>.<extension name>

It will automatically creates 4 domains for high availability, and also it will create one more domain SOA start of authority

Step2 – login to namecheap.com and enter all 4 dns names under your domain name in manage section select ns and host as <subdomain-name> & copy dns names from route 53 dns names run the following command

#nslookup –type=ns <subdomain name>.<domainname.club>

#nslookup api. <subdomain name>.<domainname.club>

Itwill shows one ip then it will conform that api got created in master server

Step3 – goto linux console install kubectl and kops form google from given links and also install awscli

Kubectl—

https://kubernetes.io/docs/tasks/tools/install-kubectl/

Kops--

https://github.com/kubernetes/kops/blob/master/docs/install.md

to install awscli u must need python-pip if its ubuntu install python-minimal and then pip

and then run

#pip install awscli

#aws configure

And enter the accesskeys and secretkeys from iam user so that it can access the any service

NOTE—

Create iam user with full permission or with full access with s3, loadbalancer, vpc, autoscaling policies

Run the command to assign the below values

# Kops create cluster –name=kubrnetes.kishore.club –state=s3://nametype59 –zones=us-east-2a –node-count=2 –node-size=t2.micro –master-size=t2.micro --dns-zone=kubrnete.kishore4u.club

When you update then your cluster will ready

# kops update cluster –name= <name of the cluster > - -yes - -state=s3://<bucket name >

#kops delete cluster kubrenetes.kishore4u.club - - yes - - state=s3://<bucketname>

You can go to cat /etc/resolve.conf and can gives dns names

NOTE:- when you use kops command must give - -state=s3://<bucket name> so that it will stores the logs whats going on

#kops validate cluster –state=s3://<bucket name >

#cat .kube/config

You can check some changes in this file

So that it will shows information about ur kubernetes cluster info like node status, instance group then it will display is ready message

Welcome to kubectl – by using kubectl we can pass commands to api > it will checks to the SCHEDULER it will checks with ETCD (it is the repository used to store the running pod key values in json format ) ---🡪then it will contact to kubelate it will decides in which service it need to create container

Then below process will done

To see the nodes that running by using below commands

#kubectl get nodes

--------To list the nodes and master in kubernetes producton-grade cluster

#kubectl get pods

--------To list the pods

#kubectl get svc or service

--------To list the services

#kubectl delete pod <pod name >

------- To delete pod

#kubectl delete svc <service name>

------ To delete service

#kubectl get rc

----- to check the replica controls

# kubectl get rs

----- to check the replication set

# kubectl get deployment

To check the deployment

# kubectl run hello-minikube --image=gcr.io/google\_containers/echoserver:1.4 --port=8080

**Ansible Notes**

Configuration management tools

-Ansible

-Puppet

-Chef

-Salt stack

|  |  |  |
| --- | --- | --- |
| Ansible Puppet |  | Chef |
| Works on contral master & server archicture | Puppetmaster, nodes | Chef DK, chef central repository, nodes |
| Language ued yml | Dsl | ruby |
| Playbooks |  | cookbooks |
| Executes code in sequental | random | sequental |
| Files: inventory, ansible.cfg,  <example>.yml | <example>.pp |  |
| Roles | Manifests |  |
| Ansible galaxy | Puppet forge |  |

**Ansible:** ansible is a powerfull configuration management tool works on master and node archicture

it had mulitiple files to do configuration changes ansible.cfg, inventory file, ansible.yml file

Ansible.cfg: it is a befault configuration file for ansible

Inventory file: is used to store remote server information like hostname, username, password.

SYNTAX for inventory file

Server name Ansible\_ssh\_user= , ansible\_ssh\_host=, ansible\_ssh\_password=

[groupname]🡪 used to give group name below this we can give server names it will assigned to particular group