GIT

git checkout -b uat for creation of new branch & moving on to it

git branch uat

git checkout uat

git push origin uat(origin=remote by default remote server url)

git head ----refers to current looking commit id of that related branch

git reset --hard(revert the changes)

tagging is nothig but lebeling (do to commit commitid)

git cat-file -t commitid(-t is type)(-s sige) --(-p displys content)

sha1 algorithm---maintains same commit id if diff files consists same data

all of the changes stored is a tree

fast forward merge----(if no changes or no commits in other branch merge to master or anybranch)

two diff chnges done in same location--merge conflicts

git init

git add .

git commit -m "First commit"

git remote add origin remote repository URL

git push origin master

git log --oneline

git log --oneline --graph(command line visuals)

git clients---tortoise git, source tree

gti pull(git fetch+git merge)--git fetch will pull the latest details from the server--git merge will merge the latest details to the branch ---git pull it connect to the remote it fetch the changes

git tag rel\_1.0

git stash(saved working directry and come back to clear the dir)

git sash apply

git stash --list name

git stash pop(remove stash)

git cherry-pic cmtid(apply spesific to the any branch)

git rebase brnachname(dis adntnge is can not maintain the histoy which version this branch is created)

hooks --local hooks and server hooks

hooking is git actions to something external

email hook on post hook(hooks concept taken from svn)

cp war file to tomcat webapps---using shellscript or cmtools(ansible or chef

quality gates---sonarqube----static code analysis--code coverage

whether the build is successful or failure

if you want to break the build go for

build breakar plugin and sonar cube runner... critical issue(rules..need to pix) ..build fails...(jenkins)

only code analysis we can do from maven only

from maven also setup sonarqube---m2 folder--settings.xml(give sonar qube url) and command mvn sonar:sonar

Jfrog artifactory

delivery pipeline and artifactioy plugin should be install from jenkins

setting.xml into m2 home directory in maven deploy---it wolud deliver failure builds to repository(dis advntage)--

slave---adding no of exicuters--increases exicution power ---slave is jenkins agent--increase capability of jenkins server no need to increase hardware increase exicuters.increse exi not only same machine othe server aswell

exicuters---its a task or activity --a command to exicute

java web start

SSH Configuraion these two ways we can configure master slave

sharing the work accross the machines.(master-slave)

restict this project where we can run give here label name (In slave wht u have given mostly here gives project names)

slaves for building diff environments.GIT

different types of maven plugins

wt is the sequences in which maven serches the dependence of library

wt are the different types of buid profiels

clin lifecycls of the

wt is the snapshot in maven

snapshot b/w version

project fully qulifie nameified artyfact

creat backup files in jenkins

Maven Plugins are generally used to

create jar file

create war file

compile code files

unit testing of code

create project documentation

create project reports

mvn [plugin-name]:[goal-name]

mvn compiler:compile

Build plugins

They execute during the build process and should be configured in the <build/> element of pom.xml.

Reporting plugins

They execute during the site generation process and they should be configured in the <reporting/> element of the pom.xml.

clean

Types of Build Profile

Build profiles are majorly of three types.

Type Where it is defined

Per Project Defined in the project POM file, pom.xml

Per User Defined in Maven settings xml file (%USER\_HOME%/.m2/settings.xml)

Global Defined in Maven global settings xml file (%M2\_HOME%/conf/settings.xml)

Cleans up target after the build. Deletes the target directory.

compiler

Compiles Java source files.

surefire

Runs the JUnit unit tests. Creates test reports.

jar

Builds a JAR file from the current project.

war

Builds a WAR file from the current project.

javadoc

Generates Javadoc for the project.

antrun

Runs a set of ant tasks from any phase mentioned of the build.

Shell

# deploy.sh

# Script to shutdown tomcat and redeploy webapp

#Check the number of arguments

if [ "$#" -ne 2 ]

then

echo "Missing arguments: webapp name and WAR file location"

exit 1

fi

if [ -z "$Jenkins" ]

then

echo "first argument"

exit 1

fi

if [ -z "$" ]

then

echo "Second argument cannot be empty"

exit 1

fi

tomcatHome="/usr/share/tomcat7"

echo "Tomcat home: $tomcatHome"

# Get the process ID of tomcat

pid=$(ps h -C java -o "%p:%a" | grep catalina | cut -d: -f1)

if [ "$pid" > 0 ]

then

echo "Shutting down tomcat PID $pid"

# Shutdown tomcat

#$tomcatHome/bin/shutdown.sh

kill -9 $pid

# Wait until tomcat is shutdown

while kill -0 $pid > /dev/null; do sleep 1; done

fi

# remove the old webapp

echo "Removing webapp $1"

mv $/var/lib/tomcat7/webapps/Jenkins.war $tomcatHome

rm -rf $tomcatHome/webapps/$1

# Copy the new WAR file to the webapps folder

cp $2 $tomcatHome/webapps/$1.war

# Change the permissions

chown tomcat:tomcat $tomcatHome/webapps/$1.war

# Start up tomcat

/etc/init.d/tomcat start

# Finished

echo "redeployed successfully"

Backup Script

# Delete all files in the workspace

rm -rf \*

# Create a directory for the job definitions

mkdir -p $BUILD\_ID/jobs

# Copy global configuration files into the workspace

cp $JENKINS\_HOME/\*.xml $BUILD\_ID/

# Copy keys and secrets into the workspace

if [ -f $JENKINS\_HOME/identity.key ];

then

cp $JENKINS\_HOME/identity.key $BUILD\_ID/

fi

if [ -f $JENKINS\_HOME/secret.key ];

then

cp $JENKINS\_HOME/secret.key $BUILD\_ID/

fi

if [ -f $JENKINS\_HOME/secret.key.not-so-secret ];

then

cp $JENKINS\_HOME/secret.key.not-so-secret $BUILD\_ID/

fi

cp -r $JENKINS\_HOME/secrets $BUILD\_ID/

# Copy job definitions into the workspace

rsync -am --include='config.xml' --include='\*/' --prune-empty-dirs --exclude='\*' $JENKINS\_HOME/jobs/ $BUILD\_ID/jobs/

# Create an archive from all copied files (since the S3 plugin cannot copy folders recursively)

tar czf $BUILD\_ID.tar.gz $BUILD\_ID/

# Remove the directory so only the archive gets copied to S#

rm -rf $BUILD\_ID

**Ansible**

--> shell/batch script

redabuilty maintainbuilty loss

misses consistent state and not easy becuse appplictions might scale as much as possible

how to do

need to write conditions in script-and install software as many times as

CMTools respsbile to giving suitable environments for an applications to work

--> IAC--Ansible--gurantty give cosistenty state

can be version control system will put it in git

what to do or what i want

Idempotency --when you exicute the script many times will give you the same(consistent) result

It will check the software is installed or not before install the software

-->push model --Ansible and salt --doesnt required to install agents to commuincate with master to node --inbulit popular mechanism SSH

need to configure user and inbuilt monitoring is not done in this to know the node is alive or not

light wegith --thin componet laptop also can act like control server in ansible(not require quardcore proccer)

node can be any os but master can be linux same in chef also

--> pull model --chef and puppet---required agent to be install

heavy weight becz every 30 mins communications happens

-->change management

-->provisioning

-->Automation

-->orchestration

installation

control server(redhat)

--> sudo yum update

--> sudo yum install epel-release

-->download it with wet --.rpm file

-->rpm -i .--rpm

--> sudo yum install ansible -y

--> ansible --version

--> sudo -i

--> adduser ansible

--> passwd ansible

--> visudo(if u woring on amazon only this stpes required)

--> after root

ansible ALL=(ALL) NOPASSWD: ALL

enabling password

--> vi /etc/ssh/sshd\_config

--> PasswordAuthentication yes

--> service sshd restart

--> exit

--> su ansible

--> cd ~

-->pwd --/home/ansible/

by defaulut inventory file located in

> cd /etc/ansible/

--> less hosts

go to root directory maintain orginal file(hosts)

-->root@ip>cd /etc/ansible/ mv hosts hosts.original

--> vi hosts

--> localhost

--> ansible -m ping all

--> yes(asking autheticity)

--> su ansible

--> ansible -m ping all

--> Authentication required generate a key

--> ssh-keygen(dont enter anything afer this click on enter)

--> ssh-copy-id localhost(machine name)

--> give password (ansible)

--> ssh localhost(here there is no need of username and password)

--> exit

--> ansible -m ping all (adhoc commond)

node machine(redhat or any liux machine)

-->add ansibel user with sudo previlages

--> adduser ansible

--> passwd ansible

--> visudo

--> ansible ALL=(ALL) NOPASSWD: ALL

enabling password

--> vi /etc/ssh/sshd\_config

--> PasswordAuthentication yes

--> service sshd restart

same network go for private ip because while resatart the machine it does not change the ip

--> hostname -f (machine name)

controlserver

--> adding the node in inventory file

--> root@ip>ansible> vi hosts

--> su ansible

copy the generated key to node machine

--> ssh-copy-id nodename(private dnsname)

--> ssh nodename

--> exit

--> ansible -m ping all

ip-172-31-24-41.us-west-2.compute.internal

ansible all -m setup -a | 'filter=ipv4\*'

ansible all -m setup -a | 'filter=ansible\_default\_ipv4\_adress'

ansible-playbook all -vv playbook.yml(verbosity[disply log info])

debug module displays the values of system variables

vars:

Role is moduler way of writing payblook --splitting plabook into multyful files.

ansible galaxy--> sharing roles

ansible-galaxy init apache-webserver(rolesname)

--defaults>main.yml--

---tasks>main.yml

A template in Ansible is a file which contains all your configuration parameters, but the dynamic values are given as variables. During the playbook execution, depending on the conditions like which cluster you are using, the variables will be replaced with the relevant values

Jenkins Playbook example

java > java-1.8.0-openjdk.x86\_64 (openjdk-7-jdk)

tomcat7 (yum install tomcat)

http://ftp-chi.osuosl.org/pub/jenkins/war-stable/2.121.2/jenkins.war

---

- hosts: appserver

become: yes

tasks:

- name: install openjdk 7

apt:

name: openjdk-7-jdk

state: present

- name: install tomcat 7

apt:

name: tomcat

state: present

- name: download jenkins to webapps/

apt-key:

url: http://ftp-chi.osuosl.org/pub/jenkins/war-stablekeyring/2.121.2/jenkins.war

keyring: /usr/share/tomcat/webapps/jenkins.war

- name: enable and start tomcat

service:

name: tomcat

state: started

enabled: yes

[appserver]

ip-172-31-24-41.us-west-2.compute.internal

[local]

localhost

/home/ansible/PlaybookTests

/home/ansible/Hosts/jenkinshosts

ansible-playbook -i jenkinshosts jenkins.yml

systemctl start tomcat8

status tomcat

/var/lib/tomcat7/webapps/jenkins.war

ansible-playbook -i jenkinshosts jenkins.yml

sudo chown -R tomcat7:tomcat7 /var/lib/tomcat7/webapps/

sudo chmod -R u+rw /var/lib/tomcat7/webapps/

vi /var/lib/tomcat8/conf/tomcat-users.xml

<role rolename="manager-gui"/>

<role rolename="admin-gui"/>

<role rolename="manager-script"/>

<user username="ansible" password="ansible" roles="manager-gui,admin-gui"/>

/etc/tomcat7/tomcat-users.xml.

**Apache Playbook example**

ansible\_distribution

ansible\_os\_family

---

- hosts: localhost

become: yes

tasks:

- name: install apache on Redhat server

yum:

name: httpd

state: present

when: ansible\_os\_family == "RedHat"

ansible ip-172-31-24-41.us-west-2.compute.internal -m setup | more

---

- hosts: localhost

become: yes

tasks:

- name: install apache on Redhat server

yum:

name: httpd

state: present

when: ansible\_os\_family == "RedHat"

- name: install apache on ubuntu server

apt:

name: apache2

state: present

when: ansible\_os\_family == "Debian"

- name: print free memory

debug:

msg: "free memory is {{ansible\_memory\_mb.real.free}}"

---

- host: localhost

become: yes

tasks:

- name: install {{item}}

yum:

name: {{item}}

with\_items:

- git

- tree

- wget

Register

cat\_contents

tags

cat\_contents\_stdmsg

cat\_contents\_stderr

templates(jinga2)--dynamic varailbes will be exiecuted(while yaml file exiecution)---configurarion related

simplyfy the yaml file into multyful files

roles(ansible-galaxy) sharing roles --gives us a folder stracture

ansible-galaxy init qt.apche

set\_fact(varible define in playbook itself not in from var groups)

role consits of logic and any playbook will access the role

role is simple playbook cated across multiple files they do it for modularity

paralal exicution is forking(by default 20

ansible all -f 2 -m setup

practice

opensource application(openmrs) deploy it

java

tomcat

webapps/.war

mysql

**Docker**

docker 1.13(latest 18.03/17.06)

docker run -d -p 8080:8080 --name detached\_jenkins jenkins

docker attach container\_name (get into the container while in container run in detached mode ctrl+Pq exit)

docker exec container\_name cat /pathname(to display the content without entering into container)

docker run -d -it myfirstimage:1.0 /bin/bash(-d takes precedence over -it)

docker attaches only be the exiecutable which hasbeen started(if u want to see wtis happening while execution u can use attach)

docker exec (docker started with -d will exec recommanded and u can call any executable command)

docker build -t myjenkins:1.0

docker run -p 8080:8080 myjenkins:1.0

add command---add files from internet and downlowd as well.

copy command-- only added files from host machin to container.

multiple run statems(application with different versions)

Image layer--run statemts create a layer and uses base layer id storagge

DOCKER DOES NOT REPLICATE DATA ANYCAUSE

union file syestem is storage driver of docker

thin read and write layer

difference b/w volume and thin read write layer..

image layer is some thing write on to your disk..when ever your write mount point ur not going via imagelayer it will directly going..

docker volume is persistent data of the container

consume storage --if run statemt content is same then it will not created new imagelayer

data from host machine to container(ex: some shared folder)

image to image

container to contaniner(data only contanier)

docker run -it -v /data-mount ubuntu:trusty /bin/bash(backup of container)

docker inspect cid

docker run-d -it -v /home/buntu/share:/data-mount ubuntu:trusty /bin/bash(mapping)

docker exec cid ls /data-mount

docker volume create testvol

docker run -v testvol:/d

data only containers is a container dies immediately after starting and it has some volume.

docker run --name data-only -v /data-mount ubunt:xenial /bin/true(/bin/true which returns true zero)

docker run -d -it --volumes-from data-only ubuntu:trusty /bin/bash

docker inspect cid

containers reaching to other container via the ip address of it.(dns name required for using machine name)

docker checkpoint--->will basically ensure that all of the application state is maintained

---( into application status as well with file system export).

docker export--> file system export only.

docker network-->

default network for docker is bridge-0 network which is used to communicate your host machine to container.

bridg network-which is created when first time install the docker itself.

bridge network is used for single host machines.

three pillars of docker native networking is CNM, Libnetwork, Drivers.

CNM--> is like requriment document specification... network req doc specificatiom

Libnework--> implimentation of, CNM application created out of it

Drivers--> specific about to a network ex: wt is the driver which we connect to

driver single host multihost can not be same..

CNM--> three components Sandbox, endpoint, Network

Sandbox:> is nothing but network namespaces..(contains full stack implimentaion of ur n/w and it has a isolated space)

Endpoint: is nothing but adapter like in linux eth0, eth1

docker network inspect bridge

multihost nework created by natively by using swarm

swarm is native clustering in docker

using swarm we can create overlay network.

single host networking is only for one bdige adpter.

docker network create -d bridge --subnet 10.10.0.0/24 qt-bridge

docker network ls

docker network inspect qt-bridge

docker run --network qt-bridge --name c1 -d -it ubuntu:xenial /bin/bash

docker run --network qt-bridge --name c2 -d -it ubuntu:xenial /bin/bash

doker network inspect qt-bridge

dns resoultion is not presented by default..

dns is the service discovery which is enabled for all the custom network then ping by name also.

multihost networking-->

VXLAN tunnel(overlay)--> endpoints (VTEP 4798/udp) created on your machine and connected to a sandbox

tunneling happens through underlay.

docker swarm init

exiecute displyd command into other machine

manager and worker

enable udp port of 2789, 7946(tcp 2377)

**AWS**

s3 simple storage service(bilk storage content)

you cant select avzones trys to replicated into all the AVzones

in ec2 will select avzones

bucket properties-

versioning- enable versioning/suspend versioning (you will not deleted automatically once version is enable suspend version over written the latest version)

Logging-

Tags-

Set Permissions- in aws by default (aws account holders) annonymous is public(access control list gives the permiison for only aws account holder not to others)

Manage Public permissions

Manage system Permissions

Management--

upload(files and folder) --> Storage class--->

Standard (most costliest option)- object durabuilty(probabulity file getting corrupt is 99.99(11 times)(0.001% chance of file lost in a year)

- object avilabuilty(99.99%)

- backedup in every avzones

Standard-IA --> obj durability is 99.99(11times) same as standard

obj avaibulity is 99.90%(lees compare with standard)

Reduced redundancy storage-- object durabuilty(99.99%(2 nines))

oject availbulity(99.99%( same as Standard)

not as many backup what standars

Glacier(cheafest option)--obj durabulity(99.99(11times)

-- meant for long term

-- back of kind of thing uses it

change storage classes-->

Encryption-> none

Amazon s3 master-key(amazone provided encryption mechanism)

AWS KMS master-key

you dont trust amazon encrypt the data and transfer to amazon(openSSL)

D/F actual file system and s3

file system directly will have web interface?(webserver)

S3 is not a file system it is an object storage.(any file its text or video or anyfile we can upload)--blob kind of storage..

In file system protocal is diffetent(ext/ntfs)

In s3 ur not only charge for storing you will also charge for transfer(charg in GB min unit is GB)

Content Delivery N/w

cloud front--

object storage details-->

put

get

size(GB)

object life cycle--> change storage classes automatically

glacier we cant use directly will it use obj life cycle

select file or folder > Management > add life cycle rule(rule naeme -file name or foldername)> transition(current/previous versions--Add transition) > expiration

standard and reducedreduncy

Backup stratagies-->speedy recovery--->incremental/full backup

onsite backup-->putting in extarnal harddisk

NAS(Network attached storage)

--> snowball edge

offsite backup-->google drive(takes time to move)

archival-->takes days(contains entire data)

ec2 instant types (purchasing options):

charge is based on hours(min unit 1hr)

on demand:creating the machine and delete on time

reserved instances: long term commit

spot:

AMI: Pre configured package(os in owr machine)

AWS MarketPlace(might charge for software)

Community AMI not charge for software

AMI's are private(myami) and public

Instance types:

micro instances

General purpose

compute optimized

GPU Graphics

GPU Compute

memory optimized

storageoptimized

Elastic block storage: it is a storage volume( hard disk)

EBS has more than 1 harddisk(1 is root vlolume[it contains the os] other additional volume)

IOPS (256kb 1IOPS)--qulality measured in iops

SSD(salid state disk)

Snapshot (copy of the data)

Security Groups: specific to a machine

open the ports to specific ip address(ex: 30.30.0.0/16)

NACL: Specific to a network(subnet level)

Route table and internet gateway: stop some traffic getting it

ec2 classic (dec 2013 before)--while restrat public and private ips will change

Insantcestore-- > when machine is delted(storage volume will delete)

EBS baked volumes--> after deleting the machine also will have storage volume(externalharddisk)(storage option while creating ec2 delete on termination bydefalut selected)

from snapshot only will create volumes from one region to onther region

from snapshot machine image cration also possible

we can attach volumes from other availabuiltyzone in same regions

increase machine volumes without stopping the machine

ebs volume types

Generalpurpose SSD: root volumes (3iops/gib)(1024kbytes in ssd in non ssd 256kb per second read/write operation can do)

provisioned IPOS: provisiond upto 20iops and will uses in large kind of database

magnetic: chepest among all cannot be measured in iops

df-h

fdisk -l(look into files)(dev/xvda location)

mkfs -t ext4 dev/xvdf(format creating the file system)

mount dev/xvdf newdir

elastic volume: increase the volume from volumetype(hdisk) to another volume type(gp2[general purpose]increase the size but not decress)

ipos--increased aswell decreases also

Route53--> map domainname with aws resources(load balancer)

loadbalance-->keeps tracks of machines

Autoscaling-->increasing and decreasing machines

Egress only internetgateways

DHCP optionset(org uses internal domain naming server which is the option set for dns is host machine name)x

elastic ips

peering connection

ECS--

Repository--> images registry

Cluster--> container insances will declare here

Task Defination-> Information about the containers

Services-->balace the tasks using autoscaling instance will be create after the service creation.

Classic load balancer(from day1)-- region wide clb . layer 4 and 7, SMTP(25) , http and https,

tcp and ssl

* does not support eip—PAAS layer.
* Does support db server select a port called tcp and3306

Disadvantage—multiple SSL certificates requires multiple elb

Healthcheck—it will check the response time machine with configured time based if it machine die trffic sends to other machine.

For Multiple regions we have multiple clbs(ex: asia.qt in Mumbai region and us.qt in oregan). Congire rules in Route 53

When we have multy region lb in alb and clb we have deploy in route 53

Create clb->elb name->elb inside (default vpc) ->http port🡪assign security groups🡪configure health checks->ping protocol-http->ping port-80->ping path->index.php

Advanced details->Response timeout – 3 seconds, interval 5, unhealthy threshold 2,healthy thresashold 3🡪add ec2 instances🡪

Launch confiugraions🡪autoscalling->create an autoscaling group->launch configuration->select ami->choose instace type->launch configuration name🡪 enable cloud watch detailed monitoring.-> user data(in advanced details)🡪select storage🡪add security group🡪create lauch config

Auto scaling Group🡪Group name🡪Group Size start with 1 Instance🡪Network->subnet

Advance Details-> Load balancing(received traffic from 1 or more lb)->select load balancer->health check type—elb or ec2 machine select->health check grace period 300 sec (default)

Configure scaling policies🡪use scaling policies to adgust the capacity of the group.

Scale start with 1 to 5 instances->scale group size name-a->metric type(avg cpu utilization)or use the autoscaling group using step or scaling policies.(to increase and decrease the scale policies.-->Increase group size🡪execute policy🡪create alarm

Send notification to ---🡪when ever avg of CPU utilz is >=60 per🡪per atleast 1 consecutive period of 1 min🡪name of alarm->create alarm

Take the action add 1 instance🡪instance needs 60 seconds to warm of after each step

Decrease size🡪create alarm-> cpu utiz less than 40 %-->remove 1 instance🡪add notification🡪review>create autoscaling group

Route 53-> Hosted zones🡪create hosted zone ->domain name🡪list of server name will disply(added to goddy) dns server mapped.-->two kind of records..a & c

C record gives your ip address of ur host name

A record gives the name of other domain.

Create record set ->name(devops4qt.com)🡪alias target (load balancer name)

Record set is consists of multiple domain alias name(ex: [www.devop4qt.com,\*](http://www.devop4qt.com,*) devops4qt.com)

ALb--> layer 7 only supports only https and http not support with tcp--

Health checks and cloudwatch improved

supports Path-based routing(ex: sales.qt ,hr.qt liks portions of domain it sends tarffic to specific machine in same one load balncer is an advantage(mail.google.com, drive.google.com—target group setup instaces for an each portion of web app) Container Support, WebSockets, HTTP/2

we cant do alb to db servers.(replicated instances in target group instaces)

listener—rules configure🡪target group

cloud watch ->

CloudWatch displays all billing data and alarms in [US East (N. Virginia)](https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1). Please switch your region to [US East (N. Virginia)](https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1) to view billing metrics and set alarms to monitor your estimated bill across all regions.

CloudWatch Events helps you to respond to state changes in your AWS resources. When your resources change state they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a pre-determined schedule. For example, you can configure rules to

About project

**Sprint**

**Modules**

**MySprint** -🡪 OverView 🡪Profile and Security

* Billing 🡪 Payment Center
* More 🡪 Change Plan

**Support**

* Self Service🡪 Support Center
* Device Help🡪 Protection Center
* More🡪feedback, Contuct us

**Shop**

--🡪Phones-🡪all phones details

🡪plans🡪 all plans

-🡪 More🡪 more offers

**Activate**

-🡪 Activate🡪Activate your new phone

Stores

Open Cart

**ClickTest** -- Middlesex Uk

ClicTest is a comprehensive software testing, one stop solution for functional testing, performance testing, API Testing, compatibility testing, test management, defect tracking and others.

**Modules**

### **Products🡪** [**ClicAutomate**](https://www.clictest.com/products/clicautomate/)**,** [**ClicLoad**](https://www.clictest.com/products/clicload/)**,** [**ClicManage**](https://www.clictest.com/products/clicmanage/)**,** [**ClicAPI**](https://www.clictest.com/products/clicapi/)**,** **ClicBrowse,** **ClicBug**

### **Pricing  🡪**

**Blogs🡪**

**Resourses🡪 Case Studies, Press Releases, Webinars, AboutUs, Contact us**

[Foremost Insurance Group](https://www.foremost.com/)

Insurance Coverage Products

Home Insurance Coverage Products

## **Business Insurance Coverage**

## **Motor Home Insurance Coverage**

Online Payment

Claims

Learning Center

**Kubernetes**

**Intra pod communications: Two containers with in a pod i.e done by a localhost with diff port nos**

**We should never have two containers in a pod expect to be open same port.**

**Inter-pod communications: two pod communication via IP Address**

**Pod life cycle—Pending, Running, succeeded and failed states. Its depends upon docker container lifecycle.**

**Master Node(s): This machine is the controller from which you can deploy Kubernetes pods. Pods are a set of containers that can be deployed across multiple nodes.**

**Pods typically run containerized applications that are very dependent on each other.**

**Worker Nodes: These machines provide an environment where Kubernetes can run.**

**etcd: This is a key-value store used for configuration data and master state information.**

**Scheduler: This service selects appropriate nodes for unscheduled pods**

**API Server: This service provides RESTful Kubernetes API used to update the etcd database. It validates incoming requests before updating etcd.**

**Controller Manager: This service takes care of all other services – discovering and managing of nodes, monitoring pods, etc.**

**Kubelet: This service manages pods on the node based on directions from the API Server on the master.**

**Kubeadm is a tool built to provide kubeadm init and kubeadm join as best-practice “fast paths” for creating Kubernetes clusters.**

**kubeadm performs the actions necessary to get a minimum viable cluster up and running. By design, it cares only about bootstrapping, not about provisioning machines. Likewise, installing various nice-to-have addons, like the Kubernetes Dashboard, monitoring solutions, and cloud-specific addons, is not in scope.**

**Instead, we expect higher-level and more tailored tooling to be built on top of kubeadm, and ideally, using kubeadm as the basis of all deployments will make it easier to create conformant clusters**

Install AWS CLI

Install kops

Install kops

4. Create a new IAM user or use an existing IAM user and grant following permissions:

5. Configure the AWS CLI by providing the Access Key, Secret Access Key and the AWS region that you want the Kubernetes cluster to be installed:

6. Create an AWS S3 bucket for kops to persist its state:

bucket\_name=imesh-kops-state-store  
aws s3api create-bucket \  
--bucket ${bucket\_name} \  
--region us-east-1

Enable versioning for the above S3 bucket:

aws s3api put-bucket-versioning --bucket ${bucket\_name} --versioning-configuration Status=Enabled

8. Provide a name for the Kubernetes cluster and set the S3 bucket URL in the following environment variables:

export KOPS\_CLUSTER\_NAME=imesh.k8s.local  
export KOPS\_STATE\_STORE=s3://${bucket\_name}

 Create a Kubernetes cluster definition using kops by providing the required node count, node size, and AWS zones.

The node size or rather the [EC2 instance type](https://aws.amazon.com/ec2/instance-types/) would need to be decided according to the workload that you are planning to run on the Kubernetes cluster:

kops create cluster \  
--node-count=2 \  
--node-size=t2.medium \  
--zones=us-east-1a \  
--name=${KOPS\_CLUSTER\_NAME}

kops create cluster

kops edit cluster --name ${KOPS\_CLUSTER\_NAME}

kops validate cluster

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-dashboard.yaml>

kops get secrets kube --type secret -oplaintext

kubectl cluster-info

https://<kubernetes-master-hostname>/ui

kops get secrets admin --type secret -oplaintext

kops create cluster --cloud=aws --zones=us-east-1d --name=useast1.k8s.appychip.vpc --dns-zone=appychip.vpc --dns private

kops update cluster useast1.k8s.appychip.vpc –yes

**Replication Controller—Desired state will be mainted.**

**In rc.yml file—selector-will try to find out what is that ur looking out for.**

**Kubectl create –f rc.yml**

**Kubectl get rc**

**Kubectl describe rc hello-rc**

**Kubectl apply –f rc.yml**

**Service -- service is to select a selector label—service knows about pod using labels.**

**Based out of the label it would create endpoints. If endpoint is empty your service is not speaking with any pod. Endpoint is pod ip address.**

**Ex: label selector: zone=prod version=v1**

**Pod Zone :prod version=v1**

**Service Discovery: DNS based**

**envrnment varibles**

**Service Type: clusterip: Stable internal cluster ip(inside network)**

**Nodeport: exposes the app outside of the cluster by adding a cluster wide port(speak with outside network)**

**Load balancer: Node port will speak with elb.(specifically designed for cloud)**

**Svc.yml**

**Metadata: name of ur service**

**Spec: is specification for this service .**

**Selecter : what are the labels which will want to search for it.**

**Service has to be created for deployment and rc as well**

**Whenever we create a nodePort and that is port being opened on every node in cluster.**

**Deployment: purpose is to ensure a new version is updated easily.and the version is not working we can rollback easily.**

**Replicationcontroller will create replication set while creating deployments.**

**Replication set(Revisions) is historical evidence of what is happing during that version.**

**Deploy.yml**

**Strategy—update will be done one by one in pods(ex:2)**

**Maxserge-1(increase by 1 and decrease by 1)**

**Kubectl is an orchestrator🡪its gives us the uniqe interface and its gives us the status of almost everything.**

**Kubectl apply –f deploy.yml –record**

**Kubectl rollout status deployment Jenkins**

**Kubectl rollout undo deplolyment Jenkins –-to-revision=1**

**kubeadm sets up a minimally viable cluster for your use.**

**Infrastructure kubeadm assumes that you already have your servers provisioned somewhere in the cloud.**

**Machines--with machines setup with some type of UNIX, you can install kubeadm onto each machine and then set the control plane (the master) on one machine.**

**Bootstrapping--once the master is set up, you’re ready to do the bootstrapping where you connect the nodes to the control plane.**

**Kubernetes API -- The Kubernetes API is downloaded and initialized.**

**Addons API -- As a last step, kubeadm intends to support Add-ons in the future. Add-ons contain a large number of community contributed projects that cover things like logging, monitoring and load balancers.**

**Kops on the other hand is responsible for the entire lifecycle of the cluster, from infrastructure provisioning to upgrading to deleting, and it knows about everything: nodes, masters, load balancers, cloud providers, monitoring, networking, logging etc.**

[**Minikube**](https://kubernetes.io/docs/setup/minikube/)**is the recommended method for creating a local, single-node Kubernetes cluster for development and testing. Setup is completely automated and doesn’t require a cloud provider account.**

**Containaization application—Docker file(know the installation steps and do the differnet instractions run copy etc) commands.**

**Build the docker file and then building the clustering out of it. (Becz want to scale)**

**Build code(version)**

**Build Image(usrname/<appname>:version)**

**Docker push**

**Docker build –t test:1.0**

**Docker tag test:1.0 usrname/test:latest**

**Docker login**

**Usrname**

**Password**

**Docker push username/test:latest**

**Aws ecr for docker hub private registry is default.**

**Docker Compose-> is compose multiple containers in one yml file**

**Overlay is meant for multi services but compose is not for multi machine. It is single machine setup.**

**Docker compose up(create and start) and Docker compose down(stop and remove) commands are for download its necessary and started and destroyed. Its starts and destroys(images and networks) the orchestrator.**

**Docker compose start and Docker compose stop and just like docker start and stop commands.**

**Linking containers. Linking between two containers are use secure tunnel in between tow machies.**

**Don’t know about other machine conatiners ip address and continers which we going to link so we**

**Relay on environment vairables.**

**--link container name: alias name Ex(--link some-mysql:mysql)**

**Environment variable pattern**

**<Aiyas>\_Port\_<port>\_<Protocol>\_ADDR(its will tells you more about the machine)**

**We cannot link 3 machines using linking containers but overlay network can be done across machines.**