==========

class-5

==========

What is Cloud ?

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet.

What is AWS cloud ?

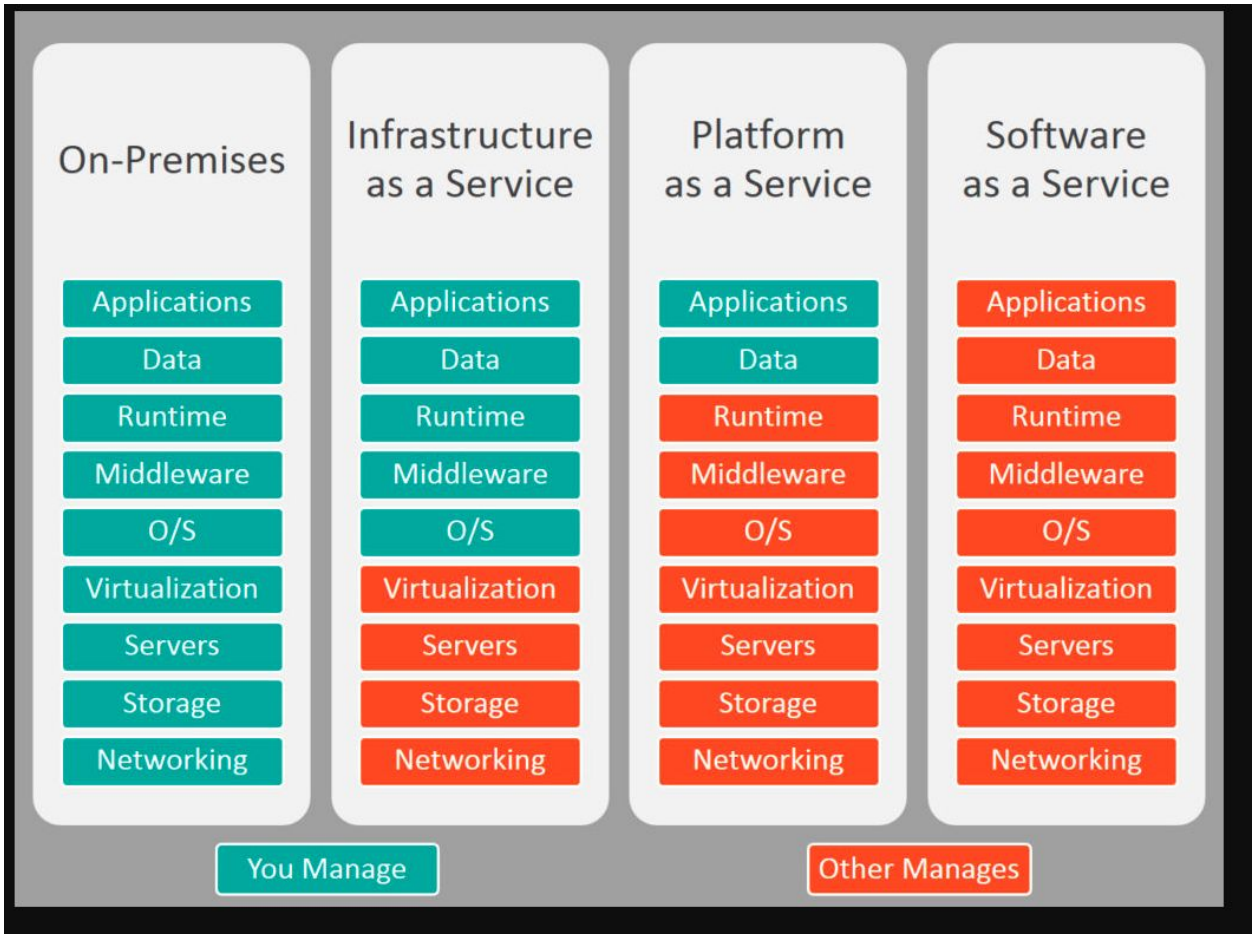
Amazon Web Services is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

AWS Plans

https://aws.amazon.com/premiumsupport/plans/

	BASIC	DEVELOPER	BUSINESS	ENTERPRISE
Cost	Free	$29/mo	$100/mo	$15,000/mo
Use Case		Experimenting	Production use	Mission-critical use
Tech Support	NO	Business hour via e-mail	24x7 via email, chat & phone	24x7 via email, chat & phone
SLA		12-24 hrs at local business hours	1 hr response to urgent support cases	15 min to critical support cases w/ priority
TAM & Support Concierge	NO	NO	NO	YES
Support Cases	None	1 Person, Unlimited Cases	Unlimited contacts/cases	Unlimited contacts/cases

Cloud Models



What is cloud/Cloud Computing ?
Providers
Iaas/SaaS/PaaS
AWS account
Computing ----->
Ec2/Load balancer

DataCenter/DisasterRecovery
Active/Active
Active/Passive

Cloud(AWS-Amazon Web Services)
==========

i want to run a application ?
Place
Server--RACK

SAN
NEtwork
Power

i want to watch a movie ?

Place
infra
movie
----------1cr

power ?

Place
infra
maintenance
power --------10cr

why we use cloud ?
pay Per Service
ondemand
Go global in mins

What is cloud ?
Cloud is service provider ,provides
infra
computing
network
storage
security
applications

IaaS -Infra As A Service
-->more control on the resources.
cpu
ram
storage
OS
stop/start/upgrade

PaaS - Platform As A Service
SaaS - Software As A service

agile--speed/fast

physical infra --Cloud providers
iaas--more fine grain on resources
paas--build and deploy
saas--use the apps

On-Premises(DC/DR)--migrate---cloud

interacting with AWS
Portal/Console -Web GUI--browser
cli - Command Line Interface
api - programs(SDK)

how to login to the aws console ?
Registration
https://portal.aws.amazon.com/billing/signup
https://portal.aws.amazon.com/

-->Creditcard/DebitCard
-->phone number

maheswargoud@gmail.com

AISPL

AWS support plans

Basic
Developer
Business

Enterprise (TAM-Technical Account Manager)

we use shared aws accounts.

For the registion credentials which i used those are called root.

dont use root user account for freequent logins.

Now login with root user-
lab:
create the IAM users
IAM= Identity Access Management

after login to the aws console.
see the region

what is IAM in AWS ?
security
To manage users/group/permissions(policy)/roles.

MFA - Multi Factor Authentication

AWS has services
Global -IAM,Route53
Region
Zone

AWS Service : IAM

AWS Identity and Access Management (**IAM**) enables you to manage access to **AWS** services and resources securely. Using **IAM**, you can create and manage **AWS** users and groups, and use permissions to allow and deny their access to **AWS** resources. **IAM** is a feature of your **AWS** account offered at no additional charge.

lab: IAM

create a alias name for your aws account

Login to the aws with root
goto the IAM service/dashboard.
see the account number

https://learnops.signin.aws.amazon.com/console

AWS account name: learnops

username: cloudadmin
password: 1234qaz

lab: Login with AWS IAM user

console---user/pawd-----IAM

tasks:
==========

- create a aws account -basic/free(12 months)
- create the alias name (IAM)
- create the user
  cloudadmin
  1234qaz

Class-6

==========
class-6
==========
Please review yesterday topics
interaction with aws
aws console/portal --username/password
cli/api -- access key/secret key

we can rotate keys /password for specific days.

we can reset the passwords in IAM.

auditing--CloudTrail
each console/api activiities recorded.
no monitoring can be done in cloudtrail

Trust Advisor

==========
it wil advice on the cost,security

IAM
CloudTrail
TrustedAdvisor
Route53

cost calculators
==========
TCO - Total cost ownership
(onprem-cloud-High level)

Simple Monthly Calculator

Cost Explorer -AWS service

charges/price

infra
---
compute
storage
network - out

note:
---
no charge for network(data)--IN

pricing model
---
ondemand
RI(Reserved pricing)-Bulk discounts
spot pricing--Bulk data center components

labs on the IAM
group
password reset --password rotation policy for
policy-permissions
role

what is diff between user and role ?

create the 3 groups
admin----administator
dev---readonly
devops--s3 permissions

authentication--username/password/keys
authorization --level of access(admin/readonly/permissions)

create the 3 users
admin1
dev1
devops1

Login with dev1 user to console.

create the error
User: arn:aws:iam::487461637069:user/dev1 is not authorized to perform: iam:CreateUser on resource: arn:aws:iam::487461637069:user/gangi1

solution :
dev1 dont have permissionos to create user.

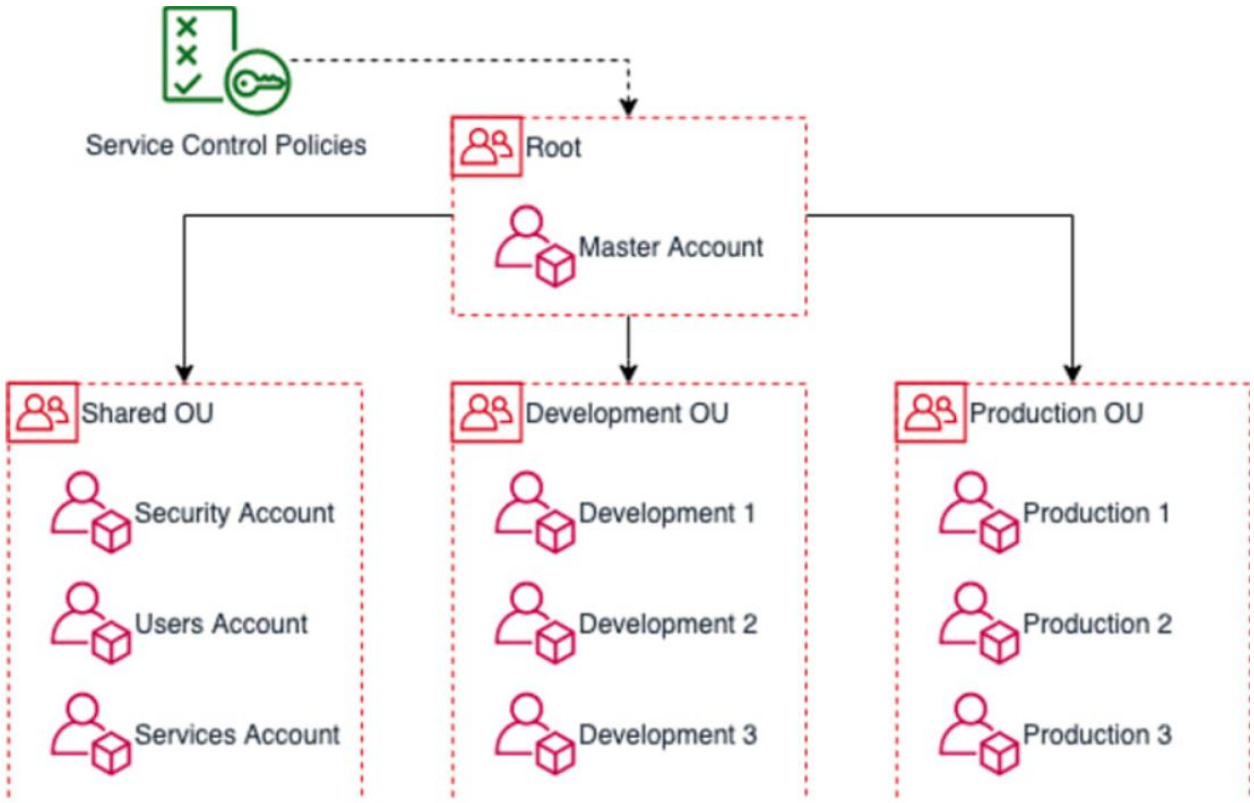
get the permissions for the dev1
or
create the user with admin

enable mfa for the root
login enter the mfa
remove the mfa
Login

IAM

AWS accounts - Manage multiple accounts
Organizations
AWS Organizations helps you configure AWS services and share resources across accounts in your organization. For example, Organizations integrates with AWS Single Sign-on to enable

you to easily provision access for all of your developers to accounts in your organization from a single place.



Single bill mgmt /Policies

Root account
Organization units accounts

Class-7

ldap and ad ?
sso
Admin flow and End User flow ?
training /labs ?
role ?

Admin flow vs End user flow

Browser

http://IP

end user connecting on port 80
in server port 80 will be running/listening.
then client will connect on 80 port number.

if 80 is down/not running our connectin will
refuse/reject.

to check remote port(running/listening)
telnet ip/hostname port
nc ip/hostname port

admin flow
==========
admin also use clients to connect to the OS(server)
server -------client
==========
Unix(linux)--putty/gitbash/mobxterm
Windows server--rdp client

admin has total control on the system/server/node/machine/host/website/processes

in OS also we have IAM concepts.
like: root user ,admin like user(sudo)

if you want to login to the server as a admin
we need to have credentials.
1)username/password
2)pem file --private key
3)without password

task: install the gitbash in windows

Total servers protected by firewall/security groups(SG)

what is firewall ?
it will have rules on what port number what ip address allowed/blocked(whitelist/blocklist)

0.0.0.0--internet

80------0.0.0.0/0-----what this rule says ?
80 port opened to internet

22-----26.45.87.92/32 --- what this rule says ?
22 port is allowed to particular IP.

telnet ip 80 ?
connection fail ?

port not listening --listen
server down --server running
please allow the firewall/sg
if 80 opens too many files

i am unable to connect to the remote port ?
what went wrong ?
port not listening --listen
server down --server running
please allow the firewall/sg
if 80 opens too many files

note: never open 22 port outside(Internet)
we allows to specific ips/within VPN

within machine thers is another security layer
iptables/firewallD(rules)

what is iptables ?
firewall inside the os .

what is security group ?
firewall outside the os .
A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. ... For each security group, you add rules that control the inbound traffic to instances, and a separate set of rules that control the outbound traffic.

vpn - virtual private network
A virtual private network extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network.

vpn clients(tool) installed in our laptops.

if i want to connect to the server

first i have to login to the vpn
then i can connect to the server
because 22 is allowed from what ip ?

keywords
========
vpn
firewall/security group
rules
0.0.0.0/0---internet
telnet
listening
refused
timeout --firewall
ssh client(putty/gitbash)
windows server---rdp client
ssh-22--SSHD
iptables

i am unable to connect to the server as admin ?
port listen --telnet ip 22
server up --ping
firewall ---allow

no too many files issue
no dns issue
vpn connecting
correct username/correct password
allowed admin ip whitelisted
remote port running
check iptables

Class-8

Compute-Launch the EC2
==========
Ec2 - Elastic Compute Cloud --Server(Linux/Windows/Ubuntu/CentOS)

firewall/sg
iptables
vpn
ssh client/rdp client
telnet /nc

https://www.google.com
telnet www.google.com 443

task:
Enable the telnet in your local machine

www.eenadu.net
telnet www.eenadu.net 80

jenkins.com:8080
telnet jenkins.com 8080

Windows OS
Power On --i wil be redirected to the Desktop
C:

D:
E:

Hard disc partioning = Data classification=file systems
C:
D:
E:

OS format/change
C: drive will be lost

users path

C:\Users\subba----this is my home directory

devops

C|\Users\devops---devops user home directory

C:\Users\subba\Desktop
C:\Users\devops\Desktop

Windows ---- unix(linux)
C:\Users---- /home ---all users home directories

linux

devops home directory ?
/home/devops

ec2-user
/home/ec2-user

root user ?

/home/root--------No

/root ---root home directory

/home/devops/*---user profile files/foders.

C:\Windows\system32\
windows commands location

C:\Users\subbu

cd ../../Windows

remote server(Linux)---compute----ec2

laptop(Windows)----------remote server
what are the info required to connect ?
public ip
port-22-SSHD
allow the port to 22---0.0.0.0/0 or only to my ip
install the ssh client in your windows machine(gitbash)
login : username/password or pem file or without password

launch the ec2(ubuntu)
==========
login to the aws console
select the region
Goto the compute---select ec2

1)Choose image(ami)--amazon machine image
static component --not running component

note: in org we create our own image(golden ami/security hardening ami)
every freequent days new image

we can share ami from one account to other aws account

2) select type of family(general/compute/memory/gpu)

3)Configure params

4)storage
5) tags
6)sg --only allow rules(no deny rules)---inbound ruels

in sg outbound by default allowed all.

what is sg ?
will have inbound and outbound rules.
it will have only allow rules
we are attaching to ec2

ec2

Region --mubai--ap-south-1
Zone --ap-south-1b

os: ubuntu
ssh client --------22---ubuntu(sshd)
private key --download--ubuntu
public ip : 13.233.51.234
port : 22

connect to the ec2/instance

what is ami and instance ?
ami=amazon machine image(OS)--sttic
instance = running component

step-1 launch the gitbash in your local
find the pem file location

ssh -i pemfile username@IP/Hostname

ssh -i "ubuntu.pem" ubuntu@ec2-13-233-51-234.ap-south-1.compute.amazonaws.com

switch to th root from ubuntu

install the apache/httpd in ubuntu(80--port)

13.233.51.234

protocal: http
ip :
port : 80

task:

create the ubuntu ec2 any region
connect to the ubuntu by using gitbash
ssh -i pemfile ubuntu@publicip (security group 22)
switch to the root user
install the apache2
apt-get update
apt-get install apache2 -y

access the page in browser using public ip (dont google)

you should have already allowed 80 in the sg

Prerequisite: Install the gitbash in your laptop/any ssh client

Login to the aws account- Sign into the Console
https://aws.amazon.com/console/

Class-9

task;
ec2-ubuntu-apache

public cloud-----AWS,Azure,GcP,DigitalOcean,Alibaba
private cloud ----IBM softlayer, OpenStack
HyBrid cloud --Mix (on-prem/public cloud)

laptop -----One OS
laptop-------virtulization ------multiple OS
laptop-------hypervisors--------multiple OS
physical cpu --virtual cpu ---vcpu
physical nic--eth----veth --virtual ether

first design the firewall rules/security group rules

redhat(ssh server(sshd))----22----myIP/VPN----------laptop
process/application/service(apache/httpd)--http server----80

52.66.203.118

C:\User\subbu\Desktop

Home directory : C:\users\subbu
C:\users ---/home
ec2-user

task: Launch the redhat with httpd(apache)
/home/ec2-user

pwd
hostname
who --who all currently logged in login
whoami -- current user name
last --history of loggged in user

ssh -i pemfile ec2-user@publicip

pwd
whoami
who
last
uptime

switch root ---sudo su

yum update -y
apache/nginx
yum install httpd
systemctl start httpd
systemctl stop httpd
systemctl restart httpd
systemctl status httpd

task: Launch the ubuntu with nginx
create a nginx security group
22 --your ip
80---anywhere

create a nginx pem file

launch ubuntu and install the nginx
apt-get install nginx -y

total account :
ubuntu-apache
redhat - apache
ubuntu -nginx

sg,ssh protocal

telnet machin1 22
telnet machin1 80

telnet machin2 22
telnet machin2 80

telnet machin3 22
telnet machin3 80

traceroute machine1
traceroute machine2
traceroute machine3

nslookup machine1
nslookup machine2
nslookkup machine3

check ping works or not

==========

class-10

==========

ec2
we will assign elastic ip(static public ip) to the ec2
we will assign additional ebs volume(storage/hard-disc)
backup the ec2
create our own ami

ap-south-1--region

check all region health status
https://status.aws.amazon.com/

ec2-status checks
1)system status check--if issue with aws infra--restart/replace
2)instance statuc check ---issue made by you---reboot

ec2--metadata--about information of ec2

task: enable detail monitoring(Cloudwatch)
select ec2 instance --goto action---CloudWatch monitoring---Select Enabe
Now onwards monitoring for ec2 for every 1 min.
by default 5 mins.

check your ec2 public ip --note down
stop ec2 - chec public ip (it will be empty)
start ec2- check public ip(it will be new ip compare to old one)

check system log --select ec2--action--instace setting---system log

if you need more eip raise a support technical ticket

create eip--associate--select eip and associate--select instance in drop down.

task: assign stroage volume(EBS volume)--elastic block (hard disc) stroage

ebs volume backup--snapshot

create our own ami (creat ec2 backup)

if you take ec2 backup it will be converted as ami.

we can create ami from snashot also.

select ec2---actions---image---create image

what is image(ami) and what is snashopt ?

image = ec2 backup
snapshot = ebs volume backup is called snapshot

keywords:
==========
eip=5 limits
status checks
normal public ip and never changing public ip ?
attach volume(ebs)--same zone
snapshot--backup of ebs volume
image backup(ami)

tasks:
validate the monitoring detailed(enable)
validate the status checks
check metadata of ec2
eip allocate to the ec2--release 4
stop/start/backup(ami)--check the system log
attach ebs volume

tomorrow task:
enable the ping
ebs volume types

in cloud ping(icmp) is disabled by default.
task: enable the ping for ec2

all region resources we can see in billing

Class-11

what is ec2
Ebs
What is EBS volume ?

Amazon Elastic Block Store (EBS) is an easy to use, high performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale.

What is Snashot ?
An EBS snapshot is a point-in-time copy of your Amazon EBS volume, which is lazily copied to Amazon Simple Storage Service (Amazon S3).

snapshot
image
storage types(ebs)--elastic block storage

zone service example ?
ebs ,ec2
deployment diagrams---architct diagrams
draw.io
microsoft viso

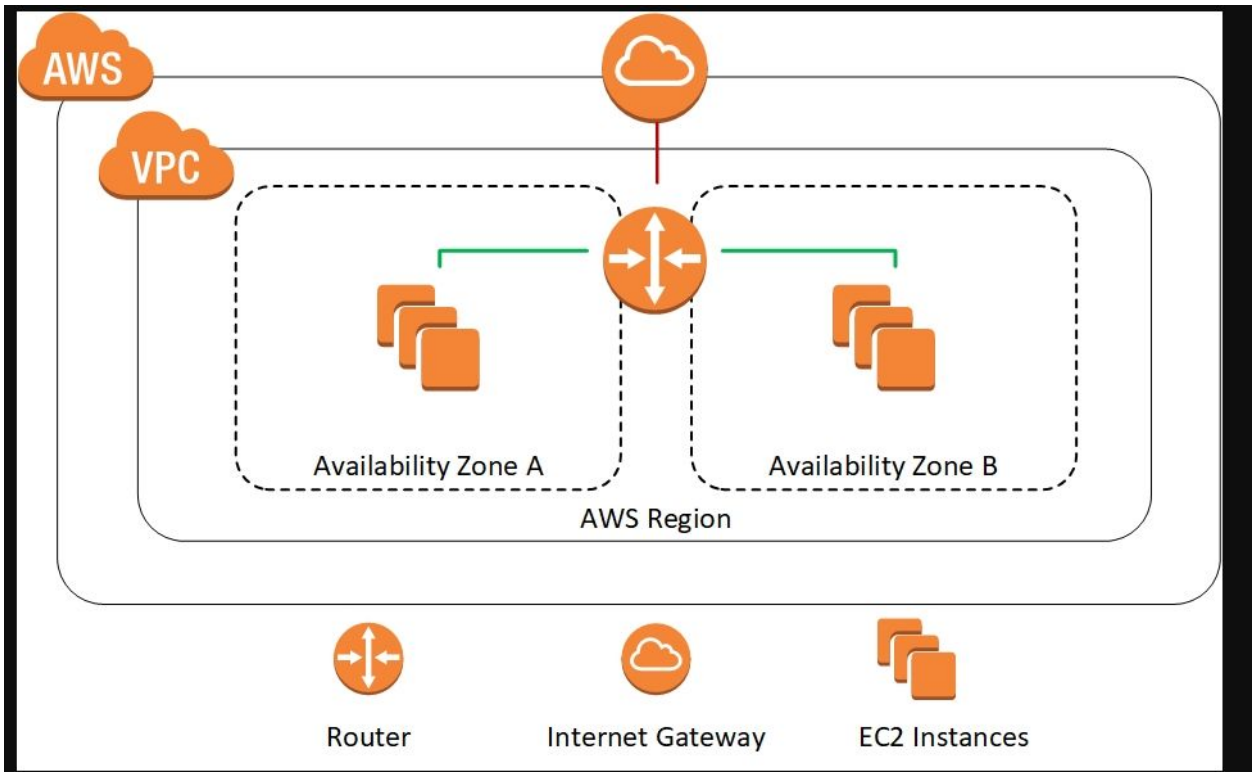
vpc=virtual private cloud-part of region
What is vpc ?
Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. ... You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

5 vpc per region
by default in your account you get default vpc
172.x.x.x---

we never use default vpc

when you launching a ec2 it wil be in a region
in a region in a vpc
in a vpc it wil go to subnet(az)

ebs is a zone service
ebs volume can be modified(scale in and scale out)(increase/decrease)



security group is vpc based service

default-vpc-------apache-sg----ec2
dev-vpc--------dev-apache-sg--not possibl to attach in default vpc ec2

C:\Users\subbu\Desktop
C:\Users\subbu\Download

ssh -i pemfile username@hostname/ip-------SSHD--shell/terminal

commands will be understand by shell

[username@hostname ~]

~ -- current user home directory
current user = ec2-user ----/home/ec2-user ----C:\Users\subba

To know drives(C drive/D drive..etc)--- df --disc fragment
To know file system in linux---df

to check memory usage(RAM)/Swap usage --free

in clouds swap will be disabled by default.

To know /check process/service/application
ps -ef | grep processname

| --filter(pipe)
grep - search for the string(word)

13.233.32.17

protocal:httpd
ip: 13.
port :80

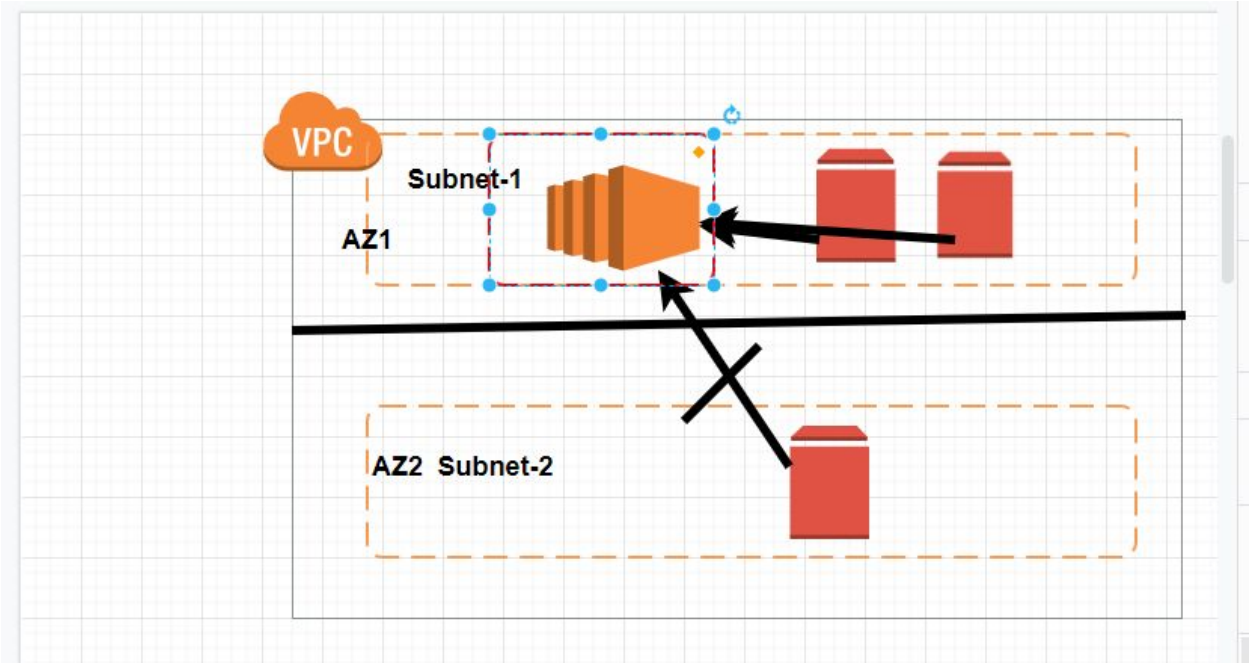
echo "<h1>this is custom webpage</h1>" >/var/www/html/index.html

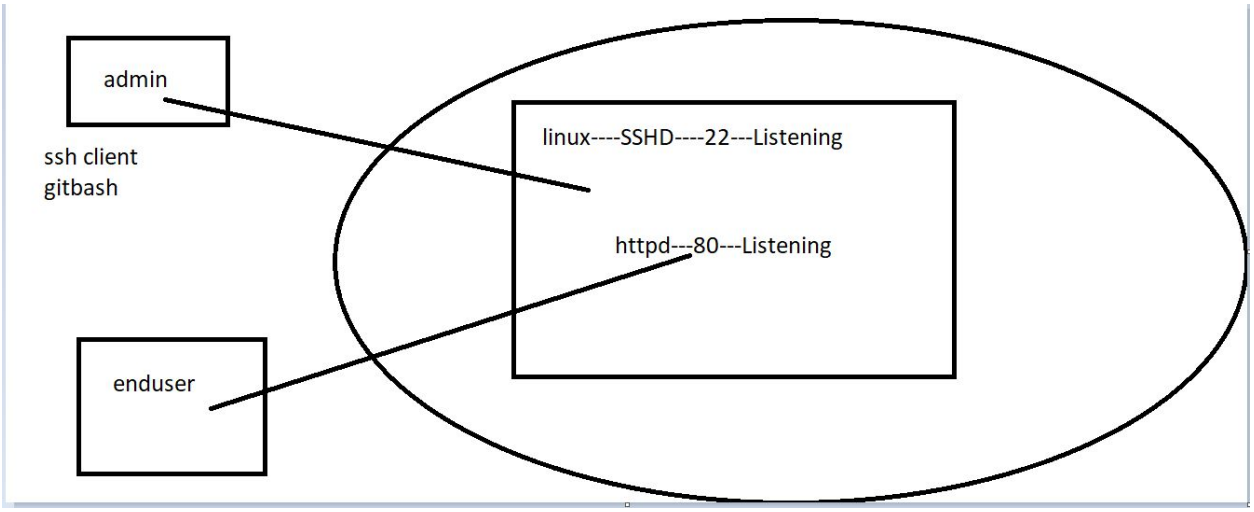
1 whoami
2 yum update -y
3 yum install httpd -y
4 ps
5 ps -ef

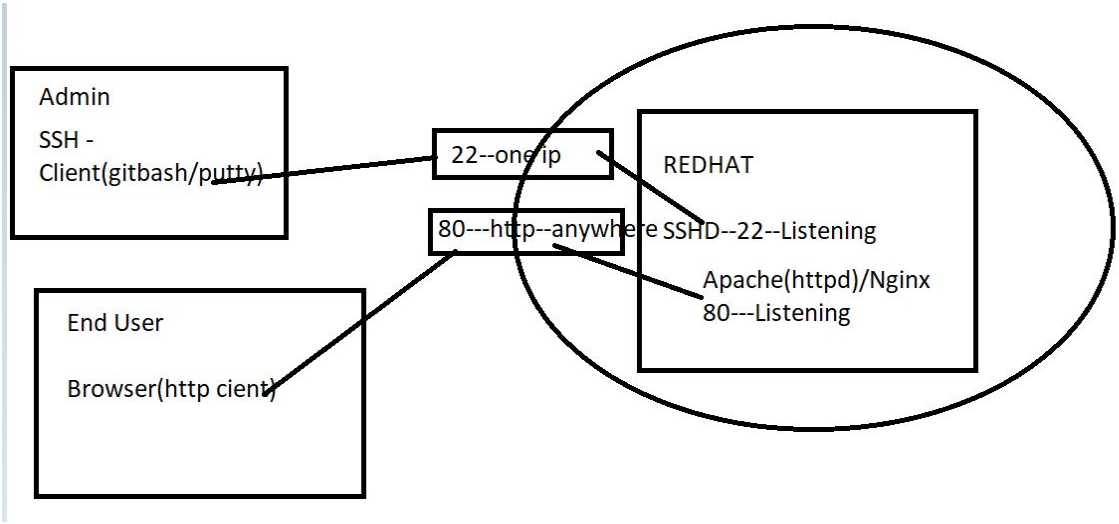
```
 6  ps -ef | grep httpd
 7  ps -ef | grep sshd
 8  systemctl start httpd
 9  ps -ef | grep httpd
10  systemctl stop httpd
11  ps -ef | grep httpd
12  systemctl start httpd
13  ps -ef | grep httpd
14  echo hai
15  a=10
16  echo a
17  echo $a
18  echo $b
19  echo "<h1>this is custom webpage</h1>"
20  watch ls
21  echo "<h1>this is custom webpage</h1>" >/var/www/html/index.html
22  history
```

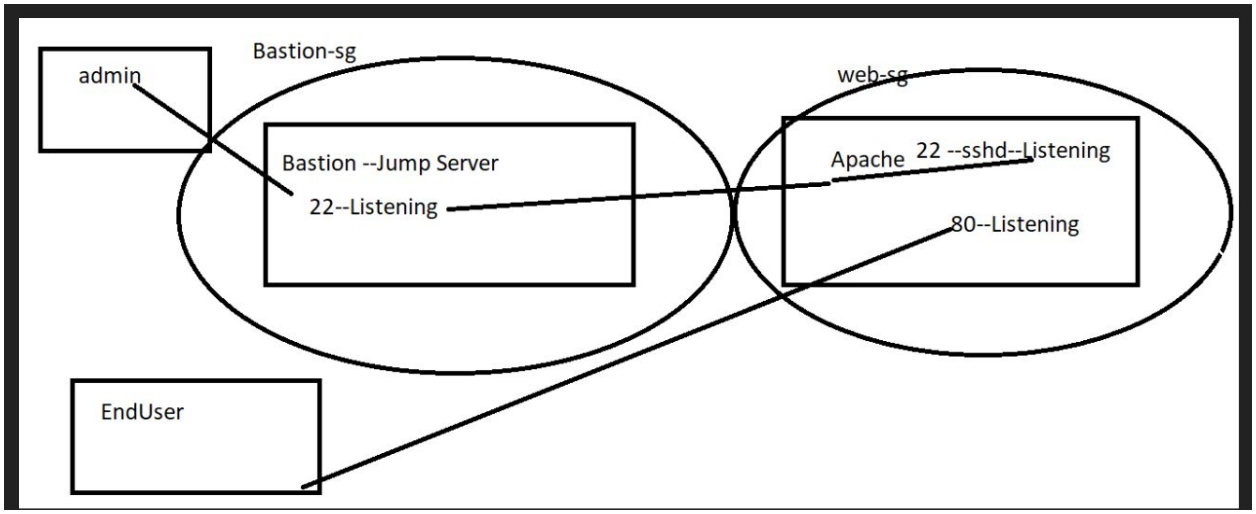
protect ec2 from deletion
ebs stroage volume types
launched c5 machine
vpc---subnet
timeout issue----use telnet command to see port--allow port in sg
connection refused issue --solve--check service

Task: launch c5.large




admin
ssh client
gitbash
linux----SSHD----22---Listening
httpd---80---Listening
enduser


Admin
SSH -
Client(gitbash/putty)
22--one ip
80---http--anywhere
End User
Browser(http cient)
REDHAT
SSHD--22--Listening
Apache(httpd)/Nginx
80---Listening

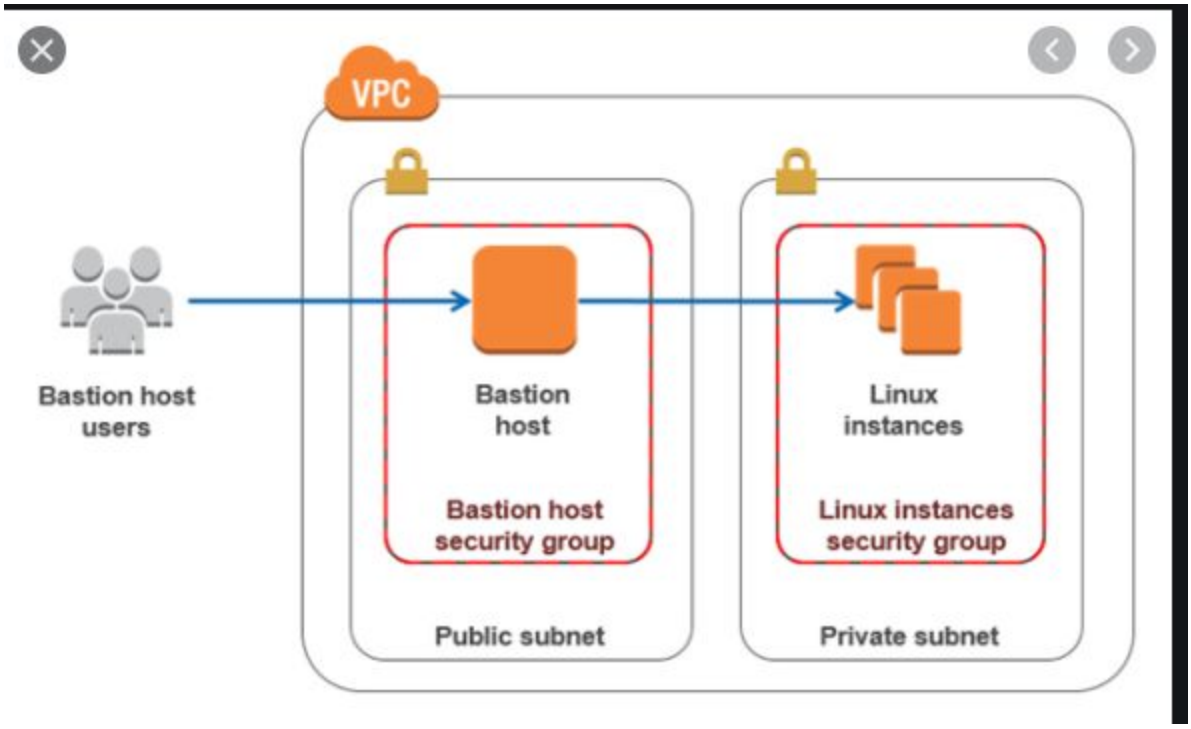


Class-12

Market Place

Shared Model

Bastion



What is Bastion ?

A bastion is a special purpose server instance that is designed to be the primary access point from the Internet and acts as a proxy to your other EC2 instance

protect ec2 from deletion
ebs stroage volume types
launched c5 machine
vpc---subnet
timeout issue----use telnet command to see port--allow port in sg
connection refused issue --solve--check service

---
ec2----network---vpc----subnet --az

OS basics
/home/----all normal users home directories - C:\Users
/root --- root user home directory
/opt and /usr --- C:\Program Files
/etc/ --- C:\Windows or C: Drive

ssh -i pemfile username@ip

ssh -i pemfile docker@ip address --------

docker ---Desktop : C:\Users\docker\Desktop

one nomral user cant access other normal user files
present user : docker
target user : jenkins
one normal user can switch to other normal user
by entering other user password.

whoami -----docker
su - jenkins (enter jenkins user password)
su =switch user

kernel-4
0-999 ---users id are os level users(system user ids)
ec2-user =1000 (created by us /aws)
root =0 (created by system)

0-1024 --ports will be allocated to system users(admin can only access)

create the users in linux
==========
useradd docker----/home/docker
useradd jenkins----/var/lib/jenkins--we can custom home directory
useradd k8----/home/k8
useradd ansible----/etc/ansible--custom home directory

all above users ids will have 1000+
we can customize users properties

we can disable the shell to users(by default shell assigned)

task:bastion and web server(apache/nginx) lab
what is bastion ?
jump server to access other private ec2 enviornments.

two ec2
two sg
one pem file

step-1
create the security groups
bastion-sg(you will get a security group id)

inbound
ssh ----22---myip/vpn ip/office ips----

outbound: egress
inbound: ingress

step-2
create the web security group
two inbound(ingress) rules
80----0.0.0.0/0--anywhere
22---bastion ip or bastion-sg

step-3:
launch bastion and assign bastion-sg
launch web server assign web-sg

step-4:
connect to the bastion server and from there jump(connect)
to web server.

bastion-sg
sg-0f62d75cce2fa384b

task: image scanning tools(vunerabilites)-csv

ssh -i verginia.pem ec2-user@52.23.185.33

ec2 connect
pem file
username
ip

copy the file from your local(Downloads) to the bastion

scp -i pemfile copyingfile username@ipaddress:path

scp -i verginia.pem verginia.pem ec2-user@ip:/home/ec2-user

tools: winscp/filezilla(copy files local to unix servers)

yum update -y && yum install httpd -y

start: systemctl start httpd
172.31.81.194

task:
bastion
webservers-2(httpd)--ami=amazon linux2(t2.micro)--default vpc--1a
first webserver---webpage1
second webserver --webpage2

Class-13

Userdata
Adding users
Login with users
Public and Private keys
Import

ssh -i pemfile username@ip

ssh -i pemfile kubernetes@master
hostname ---what is the output ?

master
whoami ?
kuberenetes
pwd ?
/home/kubernetes

LoadBalancers
Before we were running on top of datacenter/on-premises

www.vm.com

DNS---www.vm.com--
Name--IP---A record
NAme---NAME --- CNAME record
Name----name(aws cloud)---alias record/cname

what is cname and alias records in dns ?
cname=canonical name
cname= name can be reigistered with other name record.
for ex: awsloadbalncerroute53.com -----www.awsload.com

alias = the resource mapping has to be in the same cloud
what are the DNS records you know ?
A record
CNAME record
alias record
mx --mail servers

VM1--public ip--httpd
VM2--public ip--httpd
VM3--public ip--httpd

One DNS name can have multiple IPs

sbiclerk.com ---only specific people --private DNS(private ec2 names)
www.sbionline.com---any where you can access --public DNS

attach ec2 machines to the load balancer
load balancer also will have ip address/cname/alias name
lb ips are managed by aws cloud(they will be changing)

3 lbs(software)
classic lb(tcp/http/https/tls)---old--aws going to remove

application lb--l7(http/https)
network lb ---l4(tcp/tls)

single point of contact for the backend ec2 machines

bastion-sg
inbound(ingress)---
ssh---22---myip (sg id)

loadbalancer-sg
http--80---anywhere(sg id)

webserver-sg
http---80----load balancer sg-id
ssh---22---bastion-sg

i dont want to install apache on webserver by login ?
solution-1:
take ec2---install---httpd-----create---ami
launch the ami with 3 ec2 counts

solution-2:
launch the ec2 with user data(boot data/init data)

solution-3:
launch empty 3 ec2 and use
ansible/chef/puppet/salt softwares to install
httpd(apache)

ec2------ami

yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd (next reboot/restart/launch)--register
as a os level service.

i have a ec2 machine, if i restart/reboot , my service has to come up
automatically ?

enable the service as a os level service(systemctl enable httpd)

keywords:
userdata/initdata/boot data
public dns/private dns
dns record types(a record/cname/alias/mx)
load balancers(clb/alb/nlb)
clb-tcp/http/https/tls
alb-http/https
nlb-tcp/tls

Class-14

imp: diff between nlb and alb ?
l4 and l7 lb ?

class-14:
==========
load balancers-cont-
session
sticky session
session affinity
on-prem load balancer

bash=Shell=who will understand unix commands.

#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd

userdata:
base64/md5/sha/rsa/rsa1024/custom algorithm

ssh -vvv = debug of ssh command login

task: change the server name

```
echo hai
a=10
echo a
echo $a

echo ${b}
```

```
echo hai >test
```

> = redirect symbol
test=file
above we are not printing on screen
we are redirecting output to a file(test)
we have created a file or overiden the file(test)

```
echo "bastion" >/etc/hostname
```

etc=folder
hostname=filename
>=redirect the output to a file

1) change the bastion name
2) change the webservers names web1/web2/web3

```
echo "name" >/etc/hostname
hostname -F /etc/hostname
```

telnet
nc

```
sudo yum update -y
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

Classic Load Balancer

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

- we attach lb to the ec2 instances
- lb is listning on what port number ?80
- forwards the request to instances on what port number ?80
- lb will have dns names(a record/alias record/cname)
- lb will check health check to backend ec2
- if outofservice/fail it wont route the traffic.

Request------lb(dns name)----503 error code
instances are out of service

Sticky Session
tab=session
in a session we sends request.

Sticky session refers to the feature of many commercial load balancing solutions for web-farms to route the requests for a particular session to the same physical machine that serviced the first request for that session.

session affninity/sticky session
forward one session requests to same server.

browser---tab(session)----flipkart----------websrve1
i will buy one pair of shoes

lbs will support session affinity

lab:

bastion --ubuntu
nginx --ubuntu -2
classic lb --attach nginx---access lb dns name

bastion
ssh --22--myIP

lb--

http--80---anywhere

nginx
ssh--22---bastion-sg-id
http-80---lb-sg-id

apache-lb
nginx-lb

Jenkins
https://medium.com/@itsmattburgess/installing-jenkins-on-amazon-linux-16aaa02c369c

Class-15

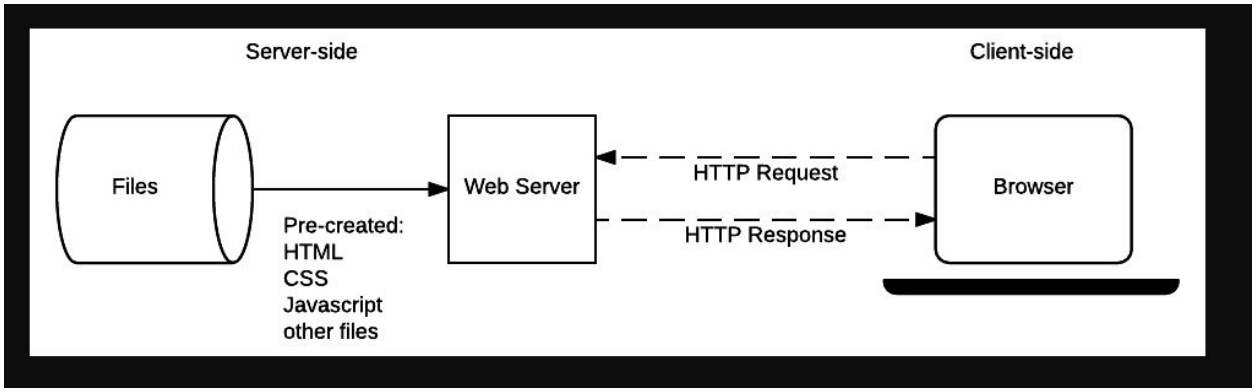
application architectures--layers
ec2(apache/nginx)--html
ec2(tomcat)--java
databases(mysql)
load balancers
basics of unix

windows---recycle bin,notepad--Desktop applications--one user
Dekstop Applications
By definition, a desktop application means any software that can be installed on a single computer (laptop or a desktop) and used to perform specific tasks. Some desktop applications can also be used by multiple users in a networked environment.

Web Applications

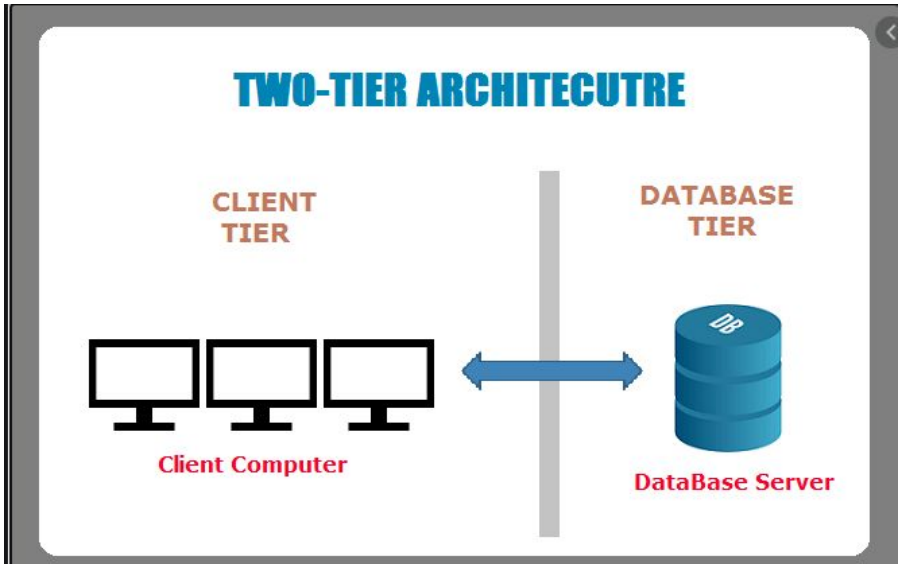
A web application is an application software that runs on a web server, unlike computer-based software programs that are stored locally on the Operating System of the device. Web applications are accessed by the user through a web browser with an active internet connection.

https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview



2-Tier Architecture

A two-tier architecture is a software architecture in which a presentation layer or interface runs on a client, and a data layer or data structure gets stored on a server. Separating these two components into different locations represents a two-tier architecture, as opposed to a single-tier architecture



3-Tier Architecture

A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

N-Tier Architecture
https://medium.com/redbus-in/hotel-website-flow-redesign-part-3-213343163632

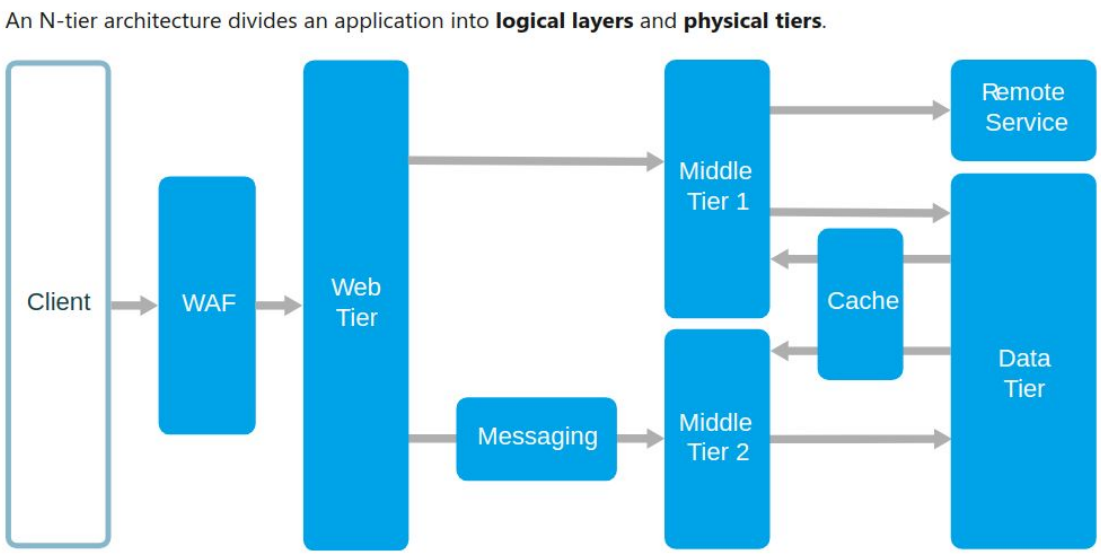
Browser---www.google.com ----web applications---n number of users

ec2----apache-----Browser(http)---public ip
client(browser)----------server(apache) -- web application
client-server model --2 tier

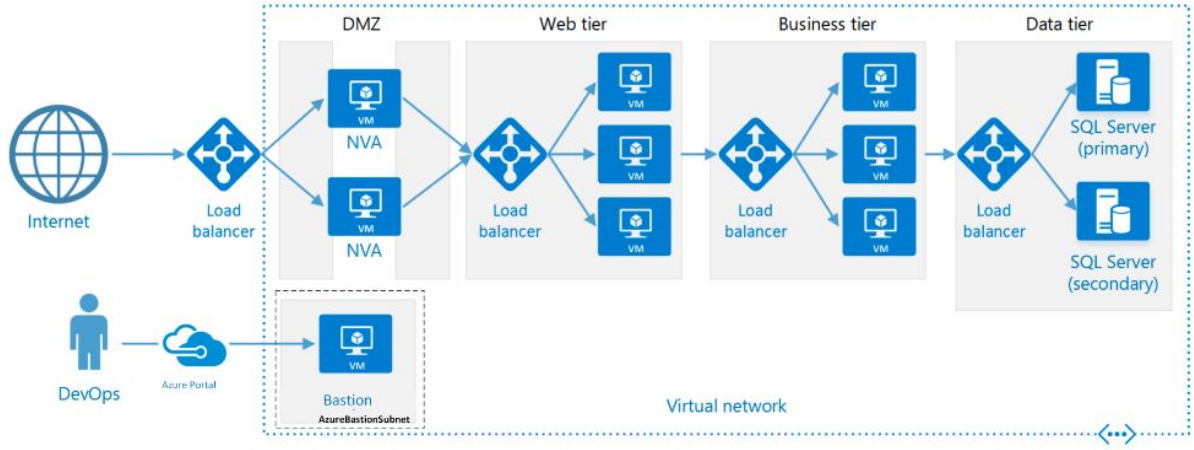
client-----apache------database(data)-----3 tier architecture

client----webserver(apache/nginx)-----application server-----database--3 tier architecure

client---lb---webservers-----lb----app servers---database--n tier



This section describes a recommended N-tier architecture running on VMs.



what is desktop app ?
what is webapp ?
what is client and server model ?
what is two tier and three tier architecture ?
what is n-tier architecture ?

api==application=url=service

Task:
load balancer
2 web servers
bastion---admins
bastion-sg
ssh---22--myIP
loadbalancer-sg
80--anywhere
web-sg
22--bastion security group
80--load balancer securty group

client-----load balancer----webservers------database(3306)

Total security group: 4 security groups
bastion -sg
==========
SSH--22--MyIP

lb-sg
==========
http---80----anywhere
web-sg
==========
http--80----lb-sg
ssh---22----bastion-sg

database-sg
==========
tcp----3306---web-sg
ssh----22----bastion-sg

1)launch the bastion
2)launch the webservers
userdata/init
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "web" >/var/www/html/index.html

name =
web_server

bin--binaries---commands
bash--shell--

webservers-1270057527.us-east-1.elb.amazonaws.com

1) create sg
2)create bastion
3)create web(user data)
4)create database ec2
5) create lb attach web ec2

check the web page working or not from web (allow sg to anywhere and again reset)

create the index.html in the webserver.

ssh -i verginia.pem ec2-user@3.80.124.27

pwd
/home/ec2-user

Connect to the database (ssh to the database)

Install mysql
https://linuxconcept.com/install-mysql-on-red-hat-7-operating-system/

Class-16

Scale in and Scale out of ec2
CLB and ALB
Basics of Linux

Webserver(apache/nginx) will serve static content(frontend)
ex: images,html,css,pdf,docs
index.html

app servers(tomcat,jboss..etc) will serve dynamic content(backend)
ex: java/python/nodeJs/.net

what is diff b/w web and app servers ?
can you tell me app flow ?

client ---frontend----backend-----cache(redis)------database

client---static-----business----cache(in-memory)---data(storage)

task:
client---lb---webserver------database(mysql)

ssh -i pem username@hostname

~ - Home directory
/home/ec2-user -- Home directory for the ec2-user

/ -- Root directory (not root user directory)
/home--Home directories for the normal users
/home/docker
/home/jenkins---/var/lib/jenkins---home directory
/root - root home direcotry
/opt - optional packages(softwares)
/usr - user softwares/packages/service/application/process
/etc - OS config file
/tmp -temporary
/bin - binaries(commands/scripts)
/dev - devices(disc)
/sbin -system binaries(root)
/mnt - mount (temporary)
/proc - run time info
/var - run time + logs
/boot - os boot files and kernal

```
[ec2-user@bastion /]$ ls
bin  dev  home  lib64  media  opt   root  sbin  sys  usr
boot  etc  lib   local  mnt    proc  run   srv   tmp  var
[ec2-user@bastion /]$
```

check commands from where they are running
which <command>

which ls
which free

ls
ls -l (long list)--detailed
ls -lr(reverse the output)
ls -lt (time based sorting)
ls -ltrh (show the size of the files in human readble)

ls - commands
command -options

To go to home

cd
cd ~
cd /home/ec2-user

cd - (alt+tab / goto the previous directory)

touch command to create the zero size file/empty file

compressed file:
==========
.zip
.gz
.tar
.bz

files.zip ----unzip files.zip

lab: setup the static website in ec2

https://www.free-css.com/template-categories/jquery

apache----------/var/www/html/

repository/artifactory =software storing location/pkg saving location

develop(build)-----upload------repository---download--install(deploy)

build------artifactory-----deploy

https://www.free-css.com/assets/files/free-css-templates/download/page258/template-1.zip

cd /tmp
wget /curl
unzip
template-1

Class-17

users in unix:
one normal user want to be like root user : sudo
in unix os default user will be created during os installation : root
root user home directory : /root

d - directory/folder
- file

when you give a request for the apache, it accept the
connection on 80 port and serve the files from /var/www/html/

copy the files from /tmp/temxxx to /var/www/html/

cp -r * /var/www/html/

mv (cut and paste)

task: host a static website in ec2 , attach the ec2 to load balancer

bastion--change the name---copy the pem file to the bastion
webserver--connect from bastion --change the name
install the apache
Download the static website -/tmp/
unzip
copy from /tmp/ to DocuementRoot/WebRoot(/var/www/html)
create the lb and attach the ec2 (webserver)
check the dns name:you will get a response page

launch the ami

create the userdata

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl enable httpd
systemctl start httpd
cd /tmp/
wget https://www.free-css.com/assets/files/free-css-templates/download/page258/template-1.zip
unzip template-1.zip
cd template-1/
cp -r * /var/www/html/
```

task: increase the size of the machine(stop)
or take ami and launch the new machine with new size

CustomAMI/GoldenAMI/SecurityHardeningAMI
==========
create the ami
share ami from one ac to other account
copy the ami from one reg to other reg
increase the size of instance(two ways)-stop and incrase
or take ami and launch the new machine with new size

clb :http/https/tcp

cons: for each website webservers we have to create separate clb.

clb-web1--------website1
clb-web2---------website2
if website count increase clb count increase.

if one lb with many websites choose application load balancer
alb: http/https

alb----------website1
alb----------website2

client----------clb---------ec2
client----------alb----targetGroup-----ec2

how to create secure hardening(golden image) ami in aws ?

Class-18

alb
targetGroup
database

ALB:-Application load balancer
L7 layer
only http/https
Slow , why ?--http headers and body(application headers)
it converts tcp packets to http and https
it converts network packets to application layer packets
Request parameters/headers
==========

http://www.google.com/
protocal: http
name: www.google.com (host)
port: 80
contextRoot: /(path)

clientIP :

Browser(agent):
time:

Response Headers:
==========
response body: html/text/video/audio...etc(MIME)

task: debug the request headers and response headers

open a browser --inspect---network ---clear---enter the url
see the status codes
200=success
404=file not found
429=too many requests
503= backened issue/servers issue
30x=cache

how do you check website response ?
how do you measure performance one api ?(postman tool api)
Goto the browser and make a inspect in the network secion ,
we can see all the apis/requests reponse.

whcih lb has visibility for the headers ?
alb
ALB checks with the rules
host based
path based

www.google.com

protocal: http
hostname: www.google.com
port: 80

https://www.google.com/

ALB converts your request from http to https(ssl)

data in transit---secure--https-ssl
data in rest --alogorithms

https protects the data over network(data in motion/transit)
ALB will have sg
ALB will listens/accept the connections 80/443 port numbers
ALB will accept the connections and forwards/redirect to target group
and targetGroup will check health check to ec2 and sends request.

we can create TG during alb creation or before or later.

in ALB we can enable logging.
lab:
create the alb
create the sg
create the tg----------1
create the webserver

alb---listener----forward----targetGroup----targets(ec2)

secure the ami(golden image)
==========
choose the right base image
limited ports/deny unused ports
remove unwanted users
install the secure packeges
patch regularly
rotate the passwords
dont allow root login
enable the ssl if required

what is pentya virus attack ?
what is ssl heartbleed attack ?
what is sql injection ?
what is DDos attack ?
what is ransomeware attack ?

Class-19

alb
targetGroup
database

ALB:-Application load balancer
L7 layer
only http/https
Slow , why ?--http headers and body(application headers)
it converts tcp packets to http and https
it converts network packets to application layer packets
Request parameters/headers
==========

http://www.google.com/
protocal: http
name: www.google.com (host)
port: 80
contextRoot: /(path)

clientIP :
Browser(agent):
time:

Response Headers:
==========
response body: html/text/video/audio...etc(MIME)

task: debug the request headers and response headers

open a browser --inspect---network ---clear---enter the url
see the status codes
200=success
404=file not found
429=too many requests
503= backened issue/servers issue
30x=cache

how do you check website response ?
how do you measure performance one api ?(postman tool api)
Goto the browser and make a inspect in the network secion ,
we can see all the apis/requests reponse.

whcih lb has visibility for the headers ?
alb
ALB checks with the rules
host based

path based

www.google.com

protocal: http
hostname: www.google.com
port: 80

https://www.google.com/

ALB converts your request from http to https(ssl)

data in transit---secure--https-ssl
data in rest --alogorithms

https protects the data over network(data in motion/transit)
ALB will have sg
ALB will listens/accept the connections 80/443 port numbers
ALB will accept the connections and forwards/redirect to target group
and targetGroup will check health check to ec2 and sends request.

we can create TG during alb creation or before or later.

in ALB we can enable logging.
lab:
create the alb
create the sg
create the tg----------1
create the webserver

alb---listener----forward----targetGroup----targets(ec2)

secure the ami(golden image)
==========
choose the right base image
limited ports/deny unused ports
remove unwanted users
install the secure packeges
patch regularly
rotate the passwords

dont allow root login
enable the ssl if required

what is pentya virus attack ?
what is ssl heartbleed attack ?
what is sql injection ?
what is DDos attack ?
what is ransomeware attack ?

Class-19
==========
Database and Application(backend-python)
Setting up of flask(python) with database in ec2
app server=flask
lang=python
database=mysql
ami=Ubuntu

what is database ?
collection of data(tables)

wha is table ?
Rows and Columns

mysql(RDBMS/sql)-------nosql(monogoDB)

top
netstat
grep -w -word match
grep -c - count
grep -i -ignore case sensitive

Install the database
apt-get install -y mysql-server mysql-client

package : mysql-server
        mysql-client

service : start mysql-server (service mysql start)
systemctl : systemctl start mysql

debugging of netstat
protocal bindIP:port clientIP:port TCP-conn-Status pid/name

clientIP:port
clientPort =Epheramal port (35kto65k)

Two tcp status are problem

time_wait--waiting for the resource
closed_wait --unable to close connection

mysql>CREATE USER 'dbuser'@'%' IDENTIFIED BY 'Passw0rd';
mysql> CREATE DATABASE employee_db;
mysql>GRANT ALL PRIVILEGES ON employee_db.* TO 'dbuser'@'%';

mysql> GRANT ALL ON *.* to db_user@'%' IDENTIFIED BY 'Passw0rd';
mysql> USE employee_db;
mysql> CREATE TABLE employees (name VARCHAR(20));

INSERT INTO employees VALUES ('Devops');

http://54.158.219.33:5000 => Welcome
http://54.158.219.33:5000/how%20are%20you => I am good, how about you?
http://54.158.219.33:5000/read%20from%20database => Devops

Client---------Flask(app.py(db info))----------Mysql(database--table)

Class-20

Class-20:
==========
Please review load balancers
5000

http://publicIP:5000/

tasK: access the flask from lb

total how many sg
3 security group
lb-sg
80----anywhere

bastion-sg
ssh--22--myIp

flask-sg
5000--lb-sg
22---bastion-sg

2nd Task
==========
ssh -i pemfile ec2-user@loadblanacerENDPOINT

lb accepts connection on which port ?22
forwards on which port ?22

clb
Alb

What is Route53 ?
Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost

effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses

setting up of Route53(DNS)
==========
Networking: Route53
HostedZone
RecordSet
RoutingPolicies

1)select your DNS name/Hosted Zone (rctcloud.in)

2)whoiswho--check your DNS

3)buy the domain name from DNS providers

gangcloud.net

Domain:
Godaddy
we can transfer domains to aws
rctcloud.in

ns-270.awsdns-33.com
ns-1932.awsdns-49.co.uk
ns-1467.awsdns-55.org
ns-748.awsdns-29.net

update in the GoDaddy

www.abc.com--------abc.com
alb.rctcloud.in------
xxx.test.com

route53------DNS
Global
Buy the domain in provider
you get default NS records in the provider(goDaddy)
we need to forward from godaddy to aws to do that
in AWS route53, create the hz with dns name matching.

once you create you get the NS in aws with the hz.
copy those ns and update to the Godaddy by deleting default(takes time)
next in the hosted zone create the record set and map with load balancers.

Class-21

Domain:
Godaddy
we can transfer domains to aws
rctcloud.in

ns-270.awsdns-33.com
ns-1932.awsdns-49.co.uk
ns-1467.awsdns-55.org
ns-748.awsdns-29.net

update in the GoDaddy

www.abc.com--------abc.com
alb.rctcloud.in------
xxx.test.com

route53------DNS
Global
Buy the domain in provider
you get default NS records in the provider(goDaddy)
we need to forward from godaddy to aws to do that
in AWS route53, create the hz with dns name matching.

once you create you get the NS in aws with the hz.

copy those ns and update to the Godaddy by deleting default(takes time)
next in the hosted zone create the record set and map with load balancers.

Route53
HostedZone
RecordSet
NS
we can map resources from one aws account to other aws
account route53.

i wanted to register the load balancer in the route53

alb.rctcloud.in

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "<h1>This is DNS demo</h1>" >/var/www/html/index.html
```

18.207.222.63

http://webserver.rctcloud.in/

Browser----Name: webserver.rctcloud.in.
.in
rctcloud.in --Godaddy--NS(4)-----AWS--Route53---
HostedZone---RecordSet----webserver.rctcloud.in-IP

ip
protocal
port:

What is Name Server ?

A DNS name server is a server that stores the DNS records, such as address (A, AAAA) records, name server (NS) records, and mail exchanger (MX) records for a domain name (see also List of DNS record types) and responds with answers to queries against its database.

ns-1932.awsdns-49.co.uk.
ns-1467.awsdns-55.org.
ns-748.awsdns-29.net.

rut53
clb.rctcloud.in
ALIAS
Web-388407514.us-east-1.elb.amazonaws.com

Task:

Create the DNS in Godaddy and create the hosted zone in the Route53
Take the Name servers in the Route53 and update in the Godaddy
Create the ec2, install the apache and access with your own DNS name updating in the Route53
Create the ec2, install the apache and attach to CLB, access with your own DNS name updating in the Route53

Create the ec2, install the apache and attach to ALB, access with your own DNS name updating in the Route53

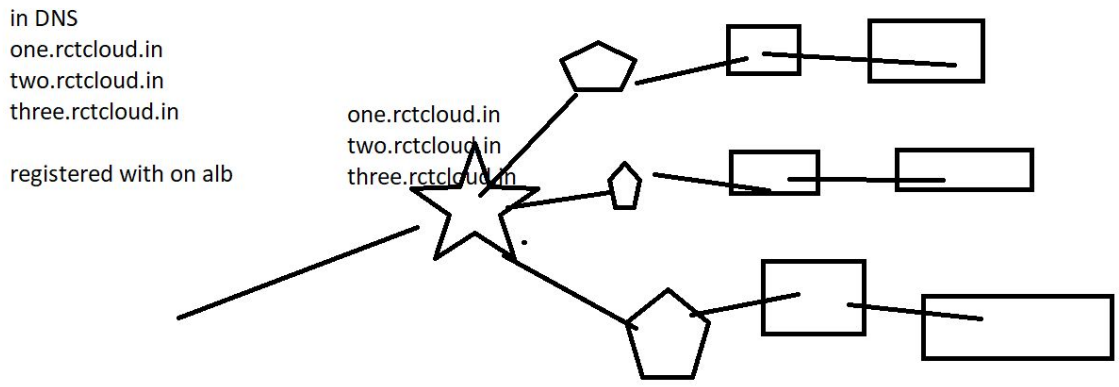
Class-22

What is Target Group ?

A target group tells a load balancer where to direct traffic to : EC2 instances, fixed IP addresses; or AWS Lambda functions, amongst others. When creating a load balancer, you create one or more listeners and configure listener rules to direct the traffic to one target group

Hosted based rules with alb
create the 3 ec2 machines(apache,nginx,jenkins)
create 3 targetgroups(apache,nginx,jenkins)
create the load balancer
create 2sg(1ec2,1lb)

update the rules
update the DNS with alb



what is the task : install the jenkins
software: jenkins
package: jenkins
binary: jenkins
artifact: jenkins
application: jenkins

windows laptop----vlc

browser-----google----vlc----website---vlc.exe

internet
repository (software/binary packages storing location)
artifactory

how yum works ?
yum searches in the repos location and downloads the packages.

jenkins installation
https://www.linuxtechi.com/install-configure-jenkins-on-centos-7-rhel-7/

What is Jenkins ?
Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.
Task:
Take 3 ec2 machines and each install with apache,nginx and jenkins
Create the alb
Create the 3 target groups named with apache,nginx,jenkins
Map the ALB in the DNS with 3 urls names matching with your zone name.
Update the rules in the ALB

Class-23
==========

bastion.rctcloud.in

ssh -i pem username@

healthcheck
/health.html

ec2
apache
/var/www/html/

curl -I http://localhost/health.html

STATUS CODE-200

port running or not
netstat -anlp | grep

EPEL
=======
Extra package Extension library
yum repolist
what is epel ?
enabling the package location to install
/etc/yum.repos.d/*.repo

yum install package-------baseURL
how yum works ?
searches in the repos base url(/etc/yum.repos.d/*.repo)

yum alternative is rpm(manual).

to generate thread and heap dumps reports for java process.

kill -3 pid
memory
top--cpu

task: one alb dns name with mutliple record set(dns)
    note: host based rules
create the alb (sg)-80---anywhere
create the targetgroups(apache,nginx,jenkins)
craeate the ec2 machines(apache,nginx,jenkins)-sg(80/8080/--alb,22--)
update the dns names in the route53(3 domain names with alb)
attach the ec2 to the tg(before health check inside with curl)
check the status codes
update the alb host based rules.
you should have all ec2 health in the targetgroups.

access the 3 websites
apache.hz
nginx.hz
jenkins.hz

key: security group,ec2,yum,package manager,status codes,
host based rules, alb,target group, route53,record set, hz,cname,alias
dependencies,rpm,repo/artifactory,jenkins

class:24
==========

Path based
nlb
ssl
vpc ---3tier architecure

website paths/contextRoot
---
https://www.espn.in/cricket-----/cricket
https://www.espn.in/football----/football
https://www.espn.in/tennis------/tennis
https://www.espn.in/nba------/nba

Path: check the path and forward to the respective tg.

We are sending from end user to alb (http) and alb to target group(http)
its pure plain http end to end.

ssl(secure socket layer-https) termination/ssl offloading
ssl offloading at alb level by adding ssl listner(443).

SSL Service:
ACM :Amazon Certificate Manager---https(tls/ssl)
free
auto-renew

alb
2 tg
2 ec2---install apache---
goto the
cd /var/www/html/
mkdir app1 /mkdir app2

create the index.html in both the folders.

path.rctcloud.in

/app1
/app1/*
http://path.rctcloud.in/app1/

http://path.rctcloud.in/

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html/
mkdir app1
cd app1
echo "<h1>this is app1</h1>" >index.html
```

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html/
mkdir app2
cd app2
echo "<h1>this is app2</h1>" >index.html
```

task:path based rules with alb

http://path.hz/app1/-------tg1
http://path.hz/app2/------tg2

NLB: Network Load balancer

=====

The load balancer distributes incoming traffic across multiple targets, such as Amazon EC2 instances. This increases the availability of your application. You add one or more listeners to your load balancer.

network load balancer

- l4
- tcp/tls
- this is faster than l7
- ssh --tcp
- kubernetes solutions it works
- we cant see request/body headers.
- client:ip/socke
- nlb: ip/socket
- ip to ip communication
- socket to socket communication.
- port to port
- nlb can forward tcp to http
- we can have ssl at nlb level also.
- nlb has target group

flow:
enduser----tcp-----nlb-----tcp----targetgroup------ec2

Class-25

Class-25:

==========

Ec2
EBS --storage types
ALB-types
Elastic IP
Vertical Scaling
Routing policies
AMI/Share AMI regions /Accounts

S3 and IAM role

==========
Compute : Ec2
Storage: EBS/S3
Network: Route53
Security : IAM
Monitoring/Ops: CloudTrail /CloudWatch /TrustedAdvisor /Billing

Simple Storage Service(S3)=Google Drive
buckets = folder
objects = files/folders
s3 is object storage
s3 cant be used for dynamic languages purpose (php/java/python/.net)
we can use for static content(videos/images/photos)
we designs applications(ec2) which will take data(static) and uploads
into s3(developers writes code for that)

client ------apps(ec2-dynamic)---------s3(static)

jobseeker------naukri(ec2)----------s3(resumes)

in migrations we uploads all the data to s3

s3=data lake(stores bulk data)
one bucket can store 5TB
we can upload 5gb at a time , more than 5gb multi-part
we can access buckets cross accounts
we can apply bucket level policies(acl)--access control list
 (only specific resource can access)
s3 supports versioning
by default s3 bucket is private
s3 supports encyption - AES256 /KMS
data in transit security(https) and data in rest (encryption)
s3 is region service , but we can acces global
s3 storage classes
moving one storage to other storage (lifecycle)

task:
goto the s3 and create a bucket and upload index.html
bucket name: take unique name

Next class:
Ec2------IAM role-------S3

how hotstar using s3 ?
s3 use cases

Class-26

task:
run a static website in aws

ec2---ebs----os----installed apache---download the website
extract and copy to /var/www/html/

another ec2 machine

2 elastic ip

load balancer

ec2 machine down ?

s3 is managed and serverless service(no os)
s3 we can use for static website hosting.

when you enable cloudfront to s3, your data will be copied to
all the edge locations(150+)

network : CloudFront (Caching service)--Edge location
cloudfront has ability to protect attacks
we can whitelist/black list geo-based clients

Amazon CloudFront is a content delivery network offered by
Amazon Web Services. Content delivery networks provide
a globally-distributed network of proxy servers which cache content,
such as web videos or other bulky media, more locally to consumers,
thus improving access speed for downloading the content

origin = from where cloudfront picks the data
s3
ec2
lb

task:
run a static website on s3 and enable cloudfront
download the website
create the bucket(name : dns name)
upload (drag and drop to bucket)
make objects public

enable static website hosting and copy the endpoing and access

http://static.rctcloud.in.s3-website-us-east-1.amazonaws.com

now add route53 with s3 endpoint--make sure same bucket name
access the website with the DNS name

client------route53-----s3(staticwebsite)----bucket----index.html

enable cloudfront
web
rtmp
origin: select the bucket -->crete dist

you will get the default endpoint for the cloudfront.
take cloudfront endpoint register in the r53

client-----r53----cloudfront-----s3---bucket--index.html
d1jvfxacakq663.cloudfront.net

https://www.myevian.com/fr/

Responsibilites of customer and cloud
AWS
Regions
AZ
Edge location

client
Applications
Data
Security of data/apps

security 'of'----AWS
security 'in'----client

Class-iam-role-s3-access-26
==========

IAM role
s3 access
lifecycle
security
bucket policies
hot-warm-cold

what is iam user and iam role ?
user will be used by admins
role will be used by services

only console need username and password

cli/api ----access key and secret key (session key)

task:
create the IAM role and give s3 access
role for the ec2
to access : s3

amazon ami:
aws <servicename> <commands>

aws s3 ls
aws s3 mb s3://create.rctcloud.in

aws route53 create-hosted-zone

configure a.key and s.key in ec2 with aws cli

pre-req: aws cli
Generate access key and secret key for iam user

user --IAM ---access key and secret key

task:
access s3 from ec2 from aws cli using IAM role(s3 full access)
access s3 from ec2 from aws cli using access key and secret key

10 admins
10 iam roles

Enable th bucket policies
use policy generator

ARN should follow the following format: arn:aws:s3:::<bucket_name>/<key_name>.
Use a comma to separate multiple values.

Effect: Allow/Deny
Prinicipal: who has to access(iam role)
Action : list/delete/create/update
ARN : target (s3)

enable lifecycle
==========
moving one stroage to other stroage type.

imp: cost

task:
enable lifecycle
enable bucket policy using policy generator

Class-s3-policies-security-27:

storage
ebs---hard disc---ec2

why do you choose s3 ?
unlimited data capacity
scalable storage
minimcal storage cost
data you can access global
high availability
static webshosting feature

security

bucket policies(sg)
encrption(data in rest/transition)
outside (access key and secret key)
lifecycle /expiration policies
enable replication(cross region replication)
async operation

sync and async ?
sync=without delay
async=with delay

task:
i have data in on-prem 500gb where do you migrate in cloud
i have static assets, in cloud where can i store ?
i wanted to backup the files
my application wanted to store static content , where do you store

storage types
security : policies and encryption types/access
iam role
maintanace:
replication :

i have s3 bucket

bucket name: resumes

application =ec2----naukri
application =ec2----linkdein

control resume s3 bucket only naukri can access ?
bucket policy: nauri

control naukri application can access only one bucket ?
IAM role --inline policy

what is iam policy and what is bucket policy ?

task: replication

create two buckets
primary.hz----enable replica for the primary
replica.hz

task: cross account s3 bucket access
two aws account
iam role
bucket policy

SSL(secure socket layer--https) Certificate service:
security: ACM
Amazon Certification Manager
pre-req: you need domain name

*.hz
wildcard ssl

Class-ssl-28:
==========

How ssl works ?
Steps for ssl setup

SSL(secure socket layer--https) Certificate service:
security: ACM
Amazon Certification Manager
pre-req: you need domain name

*.hz
wildcard ssl

Keywords:
https/ssl/tls--versions
public and private keys/certificates(symetric and assymetric)
session key
certificates
self signed certificates---we wont use , test purpose
CA Signed Certificates(Godaddy/GeoSign/Verisign/symantec/AWS)
expired and renewal

why we are using ?
security(data in transition)
trust

what is ssl ?
secure socket layer , provides data security over network/transit.

How it protects ?
uses certificates(symetric and assymetric/sha-rsa based algorithms)

symeteric key:
---
plain content----------symetric key------receiver
encypted content-------symetric key------plain text

Single key
Same key to encrypt and decrypt
Risk: if key get compromised

Asymetric key:
---
public key and private key

client
plain content ----encrpt----public key
encypted content----decrypt----private key

server
plain content ----encypt----private key
encrypted content---decrypt---public key

ssl will use symetric and asymetric
-->public and private- asymetric---userame /password
-->session key --symetric---mfa

ssl will use encyption algorithms: sha-rsa-2048(1024-old)/4096

https

rctcloud.in
rctcloud.org
rctcloud.com

www.sbi.in ---original website
www.sbi.com----fake website --username and password

Browser ----www.sbi.in
trusted certificates

CA=what is your website name ?business/location/valid =trusted

client-------------------------server(private cert/public cert)
server will send public cert
Browser will check public cert is valid

https://www.onlinesbi.com/

task: how ssl handshake works ?

class-ssl-acm-29:
==========

lab :
ec2 with load balancer and ssl -apache
ec2 with load balancer and ssl - jenkins

task:
Enable the ssl for our domains

AWS Service : ACM
Goto the ACM -

for import(outside certs) :
openssl/keytool commands
1)Generate the private key(it will ask pass phrase)--
later we have to remove from private key
2)Create CSR -Certificate Sign Request

3)Send CSR to CA
4)CA will give public(1 or 2) and intermediate(10) and Root(20 years-30 years)
5)import those to acm

Now use the acm certificates
Request public certificate ---provide the domain name
it internally creates a route53 record set with cname for validation.
see the status of issues in acm

Now map the acm certificate at load balancer level

create the load balancer
allow two listeners 80/443
map the acm certificate and allow security policy-TLS.1.2-2018
created the target group and attach ec2

https://ssllabs-1954084000.us-east-1.elb.amazonaws.com
protocal: http

create the error : cert name invalid with comman name
solve the error : create r53 record set

Now access with domain name: http://www.rctcloud.in/

http://www.rctcloud.in/

client(browser)----------server(lb)

https://www.rctcloud.in/

https://www.rctcloud.in/

www.google.com--https://www.google.com/
www.rctcloud.in --https://www.rctcloud.in/
task:
forward/rewrite/redirect http to https

in 80 listener write a rule to https

http://www.rctcloud.in/---https://www.rctcloud.in/

Flow of https with lb

client ---www.rctcloud.in-----Route53---ALB----80---Rules--Redirect--
443--https--301----443(listener)---Rules---forward to targetgroup

ssl offloading
ssl termination
redirect rule

labs:
client-----R53------ALB--80-443----ec2(apache)--20--sslapache.hz
client-----R53------ALB--80-443----ec2(jenkins)--20--ssljenkins.hz

Issues

1. Unable to connect application

DNS_PROBE_FINISHED_NXDOMAIN
ERR_CONNECTION_REFUSED-80

telnet ipaddress port
process running or not --- ps -ef | grep httpd
port listening or not--- netstat -anlp | grep "80"
service httpd status
systemctl status httpd
curl -I http://localhost
cpu
memory

Class-vpc-30

Pleaser review 3 tier architecture
N-Tier architecture apps

what is IAM
what ec2
what is s3
what is ebs
what is route53
what is cloudwatch
what is cloudtrail
what is trustedAdvisor
what is cloudFront

VPC
=========
Virtual Private Cloud

default vpc

IPv4 CIDR -172.31.0.0/16

CIDR calculations : says how many network ips can allocated in a n/w
cidr.xyz
/28
/24
/16

/28
32-28=4=16-2=14 ips get allocated
/24
32-24=8=256-2=254 ips get allocated
/16
32-16=16=65536-2=65534
/32
32-32=0=1

pre-req: CIDR (/16)

why do we need vpc ?
i wanted to control total network instead default of aws
security purpose
devide the network/layers(web/business/datbase)-note: n-tier arch
i wanted to allocate private ips/network
never use default vpc
never use public cidr range to vpc

vpc is combination/division of subnets
for:
public subnet---bastion,lb--cidr
private subnet(application subnet/app subnet)-cidr
private subnet(data subnet)-cidr

internet----------internet Gateway(IGW)----------vpc----public subnet
lab: goto your default vpc ,delete the igw
launch ec2 and try to connect

we create igw and attaches to vpc

nat=network address translator
internet-----igw----vpc----public-----private----data----nat gw

what is igw ?
to enter from public to vpc, we use igw
what is nat gw ?
private to public ,we go via nat/gw

we creates the nat-gw in public subnet and routes to private subnet.
route tables
for all pub subnets one route tables(associate)
for group of private subnets - separate route table

add igw with public subnet---route tables
add nat-gw with private --route tables

VPC
=======
cidr
subnet
igw
nag-gw
route tables

steps to create vpc
====================
decide on the subnets
decide on the cidr range

1) create the vpc give the cidr range
2)create the subnets
3)create the igw and attach to vpc
4) create the nat-gw in public subnet
5)create the route tables and assocate subnets
6)add the igw with public,nat-gw with private in route tables

Acc1----vpc-----public subnet-------app subnet-------data subnet

Acc2----vpc-----ingress subnet-----middleware subnet----database sub

vpc peering , we can connect from one vpc to other vpc
for ex: app subnet-acc1-vpc1 can connect to database sub-acc2-vpc2

for all environments can we use same vpc ?
10 applications

10-vpc
for each apps ----dev/stg/pre-prod/prod

for all prod we use separate vpcs-10 vpc
for other environments we can club in one vpc(dev/stg/pre-prod)-10vpc
total 20 vpcs

ec2(acceess key/s.key)--iam role
aws s3 ls

if you want to access s3 without seding packet to internet , use
s3 endpoints.

what is vpc ?
what is vpc peering ? why we use
why we use vpc ?
what is igw and nat-gw ?
i want to connect private to public , what are the steps ?nat-gw
what is endpoints ?

i want to connect private to s3, do you prefer nat-gw or endpoints ?

class-vpc-lab-31
==========

vpc lab

vpc cidr range
10.0.0.0/16

subnet : 3 - public , app/pvt and data/pvt

public-subnet-1 10.0.1.0/24
public-subnet-2 10.0.2.0/24
public-subnet-3 10.0.3.0/24

private-subnet-1 10.4.0.0/24
private-subnet-2 10.5.0.0/24
private-subnet-3 10.6.0.0/24

data-subnet-1 10.7.0.0/24
data-subnet-2 10.8.0.0/24
data-subnet-3 10.9.0.0/24

step-1: crate the vpc
vpc Name: dev-vpc
cidr range : 10.0.0.0/16

goto the vpc --select your vpc(dev-vpc)--actions-->
Edit DNS host name-->enable

steps-2: create the subnets
first create the pub subnets
public-subnet-1--cidr: 10.0.1.0/24---az-1a--dev-vpc
public-subnet-2--cidr: 10.0.2.0/24---az-1b--dev-vpc
public-subnet-3--cidr: 10.0.3.0/24---az-1c--dev-vpc

select pub subnet ---actions ---modify auto IP settings--enable

create the private subnets(app)

private-subnet-1 10.0.4.0/24---az-1a--dev-vpc
private-subnet-2 10.0.5.0/24---az-1b--dev-vpc
private-subnet-3 10.0.6.0/24---az-1c--dev-vpc

create the data subnets(app)
data -subnet-1 10.0.7.0/24---az-1a--dev-vpc
data -subnet-2 10.0.8.0/24---az-1b--dev-vpc
data -subnet-3 10.0.9.0/24---az-1c--dev-vpc

create the igw --actions--attach--dev-vpc

create the nat-gw , select pub-sub-1--elastic ip

create the route tables
3 public---public --route(for all pub subnets create one route)
3 app --app-route
3 data --data-route

note: ec2 machines are in subnets,subnets will be in az

NACl : protects subnets and its stateless,
it will have inbound and outbound rules(both we have add)
it follow order of rules with numbers
we can have deny rules

SG : protects ec2/load balancers/rds and its stateful
it will have inbound rules
it wont have any order
we have only allow rules
diff b/w nacl and sg ?

task: Epheramal port numbers /dynamic port numbers ?

Class-vpc-lab2- 32

vpc
Please review yesterday topics
steps

Decide the cidr range
Decide the subnets : public subnet/private(app)/database(private)

created the vpc
created the subnets
created the igw and attached to vpc
created the nat-gw in the public subnet
create the route tables
public route--3 subnets--3 public
app route -- 3 subnets -- 3 app subnets
data route -- 3 subnets -- 3 data subnets

vpc:

create the route table and associate subnets
public route--3 subnets--3 public

in route tables
route igw to the public subnets
0.0.0.0/0 ---Internet -----IGW----Public Route---PublicSubnet
route nat-gw to the private subnets
private route tables
we added 0.0.0.0/0---- nat-gw

internet -----igw---routetables(public)---public subnet---
private subnet----data subnet-----data route tables(private)---nat-gw--
public subnet----public route---igw----internet

can you tell me how the flow goes to/from private subnet (dont include data subnet) ?

internet----igw----public route---public subnets---private subnet---
private routable----nat-gw----public subnet---public route---igw---internet

Internet--IGW--Public Route tables--Public Subnet--Private Subnet--Private Route tables--NAT GW--Public Subnet--Public Route--IGW--Internet

task: create the components in our dev-vpc

create the bastion ---public subnet
create the web ---private subnet1, web2-private subnet2
crate the data base --- data subnet

connect to bastion using the public ip
connect to the web
steps:
copy the pem to the bastion ec2-user home directory
scp -i devvpc.pem devvpc.pem ec2-user@bastionIP:/home/ec2-user/
scp -i devvpc.pem devvpc.pem ec2-user@13.234.19.92:/home/ec2-user/

connect to the bastion
connect to the web using private ip (have the pem file in bastion)

10.0.1.0/24 ---public

Unable to connect to the ec2

security group within vpc

check the security group

internet------igw----vpc----public route tables---

create the lb and attach the web1/web2 to lb
bastion
ssh -i pem ec2-user@web1
privateHostedzone
web1---privateIP
web1
web2

Class-vpc-rds-33:

VPC
application architectures
Please review

vpc creation procedure

created vpc-cidr
subnets
igw --vpc
nat-gw- public
route tables

s3 with ebs

public subnet --- bastion, load balancers(clb,alb,nlb)
private subnet --- web servers(apache)
data subnet   --- database

database : how did db installe ?

service : ec2 --installed the mysql ---ebs
cons if you install db on ec2:
==========
we have to maintain ec2 , down(replica)
db version : 5.5 -8 --manual upgrade
backup of ec2

AWS as a service for the DB : RDS

database on ec2 vs RDS(no os)
RDS--mysql,postgresql,aurora(mysql/postgres),oracle,ms-sql
Managed service---AWS

RDS -- SQL / Database--Managed
Aurora(mysql/postgres)-AWS
Mysql
Postgresql
Oracle
MS-Sql

i have a db on vm , in migration which service do you choose for the db ?
rds

Task:
create the rds in data subnet
what type of database :mysql

pre-req: data-subnet1,data-subnet2,data-subnet3
rds---group
group subnets : data-subnet1,data-subnet2,data-subnet3

bigger username :

16 char

devvpclabsmysql
Login#8B1MysqlDevvpc

instace types : t, m, r, c
db types : db.tx
db.x

ec2--sg
lb-sg
rds -sg

Database as a service : RDS
aurora(mysql/psql)
mysql
postgresql
oracle
ms-sql
maria

why do we go for rds ?--serverless---db.m5.large
managed service
auto-upgrade
auto-backups
auto-failover
cross-region replicas

task: connect from flask to rds

flask on which subnet ? private subnet/app

own vpc
rds
flask
alb

lab:
flask.rctcloud.in----->R53--->ALB(80)------Flask(5000)----RDS(3306)
dev-vpc-alb-sg
dev-vpc-flask-sg
dev-vpc-rds-sg

Class-34-ssh-password-less

Please review vpc and RDS basics
linux
jenkins
github/gitlab
build and deploy

linux ----------------linux
username/password
pem file
without pem/without username/password---*

ssh passwordless setup

bastion to web/flask/app subnets

jenkins
vpc - 192.168.0.0/16
subnets
prod-vpc-pub1-192.168.1.0/24
prod-vpc-pub2-192.168.2.0/24
prod-vpc-pub3-192.168.3.0/24
prod-vpc-pri1-192.168.4.0/24
prod-vpc-pri2-192.168.5.0/24
prod-vpc-pri3-192.168.6.0/24
prod-vpc-data1-192.168.7.0/24
prod-vpc-data2-192.168.8.0/24
prod-vpc-data3-192.168.9.0/24

igw
nat-gw
route tables

security groups

prod-bastion-vpc-sg - sg-0b1724b22d9bb318e
22---myIP

prod-app-vpc-sg
22 -- bastion

prod-jenkins-vpc-sg
22 -- bastion

ssh password less setup

connect to the bastion ----other ec2
current user : root
ssh-keygen
algorithm: rsa
current user : root home directory : /root
/root/.ssh/id_rsa --- private
/root/.ssh/id_rsa.pub --- public

/root/.ssh
id_rsa --private key
id_rsa.pub --public key

bastion : pem /tmp

target server : jenkins/app1/app2/app3
target user :root
/root/.ssh/authorized_keys

copy the bastion server(root user) id_rsa.pub file
to target server target user(root) authorized_keys

1)generate the keys in the bastion server
current user : root
it will generate two keys (id_rsa and id_rsa.pub)
current user home directory : /root/.ssh/

2)copy the id_rsa.pub to the target server
target user : root
current user home directory : /root/.ssh
file name: authorized_keys

Goto the bastion --root ----ssh ------appserver---root-----su--ec2-user
Goto the bastion --root ----ssh ------appserver1---ec2-user
--/home/ec2-user/.ssh/authorized_keys
Goto the bastion --root ----ssh
------appserver2---ec2-user--/home/ec2-user/.ssh/authorized_keys
Goto the bastion --root ----ssh
------appserver3---ec2-user--/home/ec2-user/.ssh/authorized_keys

copy the public

bastion server
user: docker
connect to the bastion server , switch to the docker user
ssh-keygen
/home/docker/.ssh/id_rsa(private),id_rsa.pub(public)

copy the public key to the
target servers: app server
user: deploy

connect to the app server--deploy user

/home/deploy/.ssh/authorized_keys (copy the above docker user public key)
tell me steps to connect from docker to deploy user without password/pem file ?
include the home directories path and steps clerarly

ssh password less setup

current server
current user : generate ssh-keygen(id_rsa and id_rsa.pub) ---current user home directory .ssh

target server
target user : target user home directory : .ssh/authorized_keys

current : Ansible server
user : root

target server : tomcat server
user : tomcat

ssh password less setup
ansible to tomcat

connect to the ansible server
switch to the root user :
root user home directory : /root/

ssh-keygen

/root/.ssh/id_rsa and id_rsa.pub
only public file we have to copy to tomcat server (id_rsa.pub)

target server : tomcat
user : tomcat ----- /home/tomcat/.ssh/authorized_keys

jenkins(keygen---id_rsa.pub) --------3000 servers

Class-35-Tomcat-pvt -subnet

Task: setup of apps(java) in app server

languages:
static code(html/jquery/reactJS) -----Apache/Nginx/S3--webserver
python (app.py)----------Flask
java ----------Apache Tomcat---appServer

Connect to the bastion with ec2-user
ssh -i pemfile ec2-user@13.229.147.45

switch to the root user in the bastion from ec2-user

ec2-user to root
sudo su -
sudo su
sudo -i

check whoami ---root in the bastion

bastion-root user to app1-server----ec2-user
connect to the app server1
ssh ec2-user@192.168.4.72

in App Server
setup of apps(java) in app server
installation of tomcat
pre-req: java(jdk/jre)

linux: amazon/centos/ubuntu/redhat

java : oracle java(sun)/openJDK(redhat)/IBM java/...
yum install java
yum install java-11-openjdk

rpm(redhat)/deb(ubuntu) :

download the rpm : wget
  wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.rpm

Install after download of java rpm file

rpm -ivh jdk-8u131-linux-x64.rpm

JDK=JRE

check java version
java -version

https://tomcat.apache.org/

Install the tomcat
Download the tomcat(zip) --- /opt/

wget
http://apachemirror.wuchna.com/tomcat/tomcat-9/v9.0.38/bin/apache-tomcat-9.0.38-windows-x64.zip

Unzip the tomcat
unzip apache-tomcat-9.0.38-windows-x64.zip

/opt/tomcat9/bin ---tomcat stop/start scripts
/opt/tomcat9/webapps ----deploy the apps
/opt/tomcat9/conf -----configuratin files ---server.xml
/opt/tomcat9/logs--------check the logs ----catalina.out

rename the tomcat directory (/opt)
mv apache-tomcat-9.0.38 tomcat9

Goto the tomcat bin
cd tomcat9/bin

change the permissions to the startup/shudown sscripts

chmod 755 *.sh

next start tomcat

./startup.sh (/opt/tomcat9/bin)

check tomcat running
ps -ef | grep tomcat

check tomcat port

netstat -anlp | grep ":8080"

appServer-SG--ec2

8080-----sg-0d451d8ebd4b0df2d

load-balnacer---80---sg-0d451d8ebd4b0df2d

alb---targetgroup(8080)--/----------ec2-----tomcat(8080)

targetGroup

alb

ec2

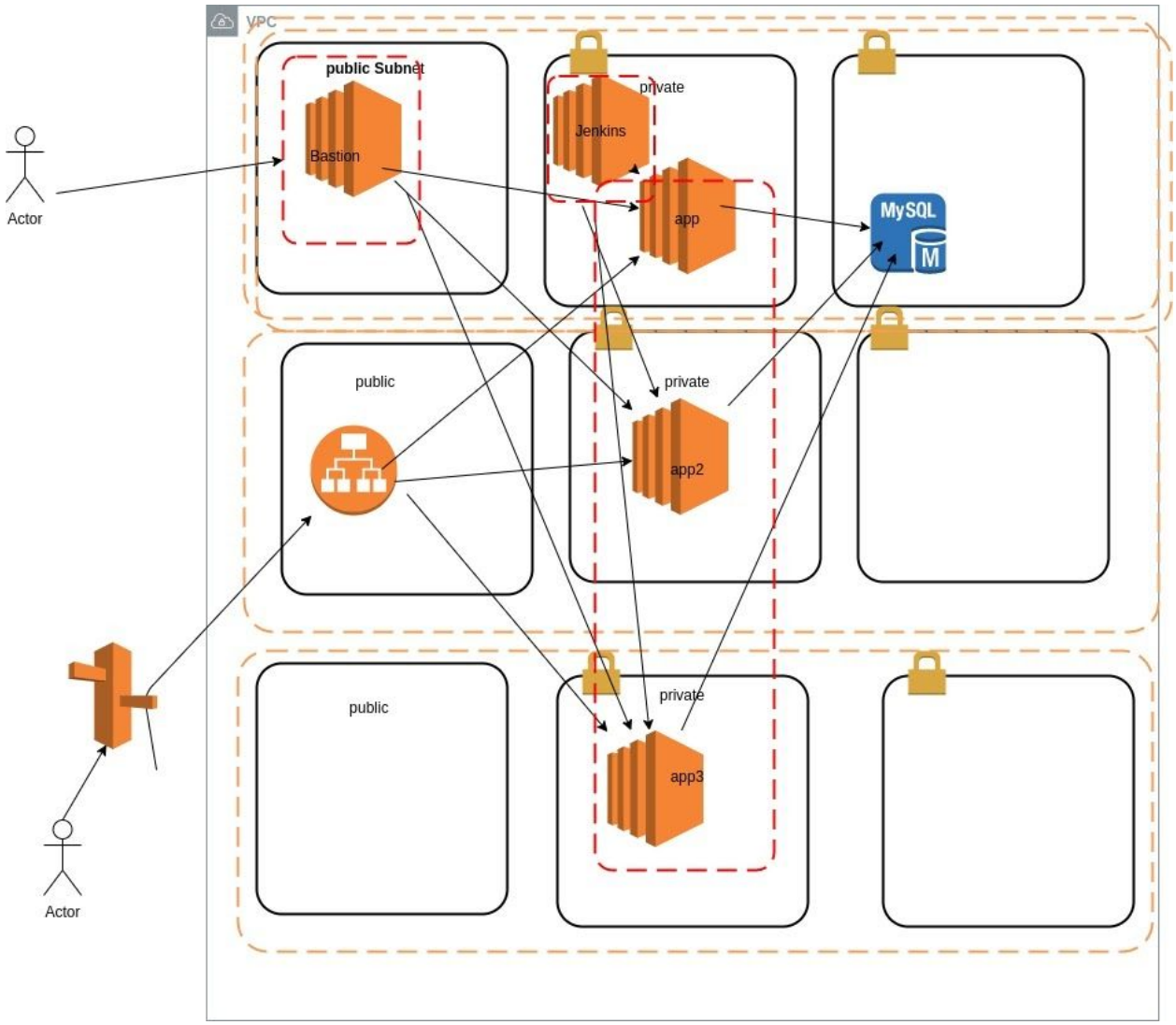
subnet

alb-----targetGroup------ec2

Tomcat installation script

Tomcat Installation script

```
#!/bin/bash
#change to the /tmp
cd /tmp
#Download the java
wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie"
http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163
/jdk-8u131-linux-x64.rpm
#install the rpm
rpm -ivh jdk-8u131-linux-x64.rpm
#change to the opt
cd /opt
#Download the Tomcat
wget
http://apachemirror.wuchna.com/tomcat/tomcat-9/v9.0.38/bin/apache-tomcat-9.0.38-windows-
x64.zip
#unzip the tomcat
unzip apache-tomcat-9.0.38-windows-x64.zip
#delete the zip file
rm -f apache-tomcat-9.0.38-windows-x64.zip
#rename
mv apache-tomcat-9.0.38 tomcat9
#change the permissions for the scripts in the bin directory
cd tomcat9/bin
chmod 755 *.sh
#start the tomcat
./startup.sh
```



task: Connect from jenkins ec2 to tomcat ec2 servers

check the keys
in which server ?
 jenkins --whcich user ----

checkc tomcat ec2 security group , from jenkins allowed ---22

deploy a shoppingcart.war in the tomcat

https://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/wls/10g/r3/cluster/session_state/files/shoppingcart.zip

we cant deploy java code directly

java code ----package-----war

Download
wget
https://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/wls/10g/r3/cluster/session_state/files/shoppingcart.zip

unzip shoppingcart.zip

it will create a shoppingcart.war

copy to the webapps of tomcat
cp shoppingcart.war /opt/tomcat9/webapps/

prod-alb-231615854.ap-southeast-1.elb.amazonaws.com/shoppingcart

problem:
alb with session data is missing
sticky session alb

task: install the tomcat on appServer
pre-req: java /rpm
stop/start tomcat
ALB ---targetgroup---8080

deploy the shoppingcart.war in the tomcat/webapps

Class-36-architecture-diagram-36

On-prem architecture

application : java(war files--tomcat)
vm ---OS --10 vms
database : mysql
load balancer
webservers(apache/nginx)(mod_proxy/mod_jk)

flow :
enduser-----dns ---lb -----webserver(r.proxy)----tomcat----database

technologies:
tomcat (java)
mysql
webservers(apache)
load balancer

AWS Cloud
==========
existing architecture
existing OS
existing RAM/CPU for webservers/tomcat/mysql

VPC
public(ingress)
private(app)
data

elastic cache: memcache/redis
computing : ec2/elastic ip /ebs /load balancer
network: vpc /subnets/igw/nat-gw/security groups/route53
ssl : ACM
database : RDS
storage : s3
Cloudwatch

webserver has the capacity of behaving as a lb

webserver forwards the requests to backend app servers.
by using proxy modules.
---------webserver-------ec2(tomcat)
---------lb-------ec2(tomcat)
http://www.shoppingcart.com/shoppingcart

72 tomcats

12 webservers

task: stop all services in your project in production

1hour

ssh tomcat1 ---cd /opt/tomcat9/bin/shutdown.sh
ps -ef | grep tomcat
kill -9 pid

---->5 mins

start

deploy the war files
hot deployment----------wont prefer
cold deployment
stop
deploy
start
---10 mins---720 mins

jenkins

connect to the tomcat
stop
ps
kill
start

start
deploy
No server login /no ssh

Class-37-jenkins installation

why we do need ssh password less setup ?
what is jenkins
its the automation server(CI) , takes instructions
and runs on the remote server

why are we using here
tomcat stop
tomcat start
deploy (war)

we can install on top of tomcat(jenkins.war)

jenkins : pre-req java
version : 2.15x
free vs paid(cloudBees Jenkins)

plugins
jobs (task)/pipeline

jenkins=Hudson

we are going to configure ssh password less
setup between jenkins to tomcat
jenkins :
user : jenkins
home directory : /var/lib/jenkins

tomcat:
user: root

Goto the jenkins server(jenkins user)

ssh-keygen
(id_rsa,id_rsa.pub)---/var/lib/jenkins/.ssh

copy the id_rsa.pub from jenkins server jenkins user
home directory

tomcat : user: root : /root/.ssh/authorized_keys

Install the jenkins
connect to the bastion
jump to the jenkins from bastion(Pem file should be in bastion)

yum install jenkins

gitbash installation in the laptop

download internet--gitbash website
package :gitbash -----repo

EPEL : jenkins epel

yum : /etc/yum.repos.d/*.repo

wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo

rpm --import http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key

yum install jenkins --nogpgcheck

to see file content :
more
less

cat

user creation process in linux
user created
user id
group created
group id
home directory
home directory will have few profile files
login shell :bash
/etc/passwd will have a user entry

when i do login with jenkins user , not able to login why ?
default jenkins user wont have login bash shell
/etc/passwd

how can i login if i want ?
we can change in the /etc/passwd for the jenkins user
from /bin/false to /bin/bash

or when you login to jenkins use this command

su - jenkins -s /bin/bash (swtich with bash shell)
task: change the jenkins port number
change the jenkins startup user

os level sginals
SIG

kill -3 SIG
kill -9 SIGTERM

/var/lib/jenkins/secrets/initialAdminPassword

install the java
install the jenkins
created the alb
login to the jenkins

home directory or installtion directory

/var/lib/jenkins

https://jenkins.rctcloud.in

jenkins installation

https://medium.com/@itsmattburgess/installing-jenkins-on-amazon-linux-16aaa02c369c

Class-38-jenkins-jobs-tomcat

configure the ssh password less setup b/w
jenkins to tomcat servers

automatically configure the pub keys to the new machines : userdata/initData/bootData

user : ec2-user---nomral user

crate user: docker
useradd docker
check profile files by switching to the docker
ls -ltra
pwd
o/p: /home/docker
environment files: .bashrc/.bash_profile

set the password for the docker user(root/docker)
passwd username

do the grep in /etc/passwd file

cat /etc/passwd | grep -i jenkins

grep -ic "keyword" filename --count
grep -in "keyword" filename -- match line no
grep -iw "keyword" filename -- exact word match

switch to the jenkins from root user
su - jenkins -s /bin/bash

/var/lib/jenkins : config.xml

jobs
logs
plugins
workspace--we can clear/delete

sg-075c2671d26010ef3

task/job: stop tomcat from jenkins

connect to the tomcat ----
cd /opt/tomcat9/bin
./shutdown.sh

/opt/tomcat9/bin/shutdown.sh

/opt/tomcat9/bin/startup.sh

id docker
root : admin

job----cmd-----jenkins---

ssh tomcatIP"

lab:
start/stop tomcat from jenkins

Class-39-Jenkins-CICD

How job/task/pipeline works

jenkins process user : jenkins
home directory : /var/lib/jenkins

jenkins console ----job ---free style----
java------war
job name: build-war-package

Options in the job :
General----------name,security,logs,parameters
Source Code Managment ---where is my code
Build Triggers ---- how job will run auto/manual--night builds/schedule builds/cron build
Build Environment --- do i need use tools/envs
Build -----tasks execute section
Post Build ---next job/report / email/slack/

cron tab:

Min Hours DayOfMonth Month DayOfWeek script/cmd

Min - 0 to 59
Hours - 00 to 23
DayOf Month - 1 to 31
Month - 1 to 12
DayOfWeek - 0 to 6

run a job 3am every day

0 3 * * * script

every 5 mins

*/5 * * * * script

monday 11pm to 4am at 30 mins

30 23,00,1,2,3,4 * * 1 script

30 23-4 * * 1 script

Build a package
war file

java code

hello.java ---compile ------hello.class---pkg---war---tomcat

code ----build(compile)----package----war/jar/ear
hello.html ------apache

source code (java)--------jenkins----Build----WAR

developer uploads the code to Source code repository (SCM)

to interact with scm we use two protocals
ssh --password less
http ---username/password

SCM : Github / Gitlab /BitBucket /SVN

Repo URL

jenkins also interacts with scm two protocals
ssh
http

hello.java

java build tools : maven/ant (compile+package)---war

flow of job :

developer writes the code and pushes to scm (repo)

jenkins takes the repo code with url and copies to the
jenkins workspace. :
/var/lib/jenkins/workspace/build-war-package/

we use tools like (maven/ant) on the workspace folder on the code and do a compile and package with the given commands.

jenkins workspace

/var/lib/jenkins/workspace/<job-name>/*.java

all the build instructions in the jenkin job runs on
workspace.

ex:

job name: test ---execute shell : rm -f demo

path : /var/lib/jenkins/workspace/test/

job name: docker-image-build
shell : docker build -t
path: /var/lib/jenkins/workspace/docker-image-build/

job name: maven-build-war
shell : mvn clean package
path: /var/lib/jenkins/workspace/maven-build-war/

jobs configuration

/var/lib/jenkins/jobs/maven-build-war
/var/lib/jenkins/jobs/docker-image-build

Tools :

SCM : github
CI tool : jenkins
Build Tool : maven
Artifactory : S3

Middleware server : Tomcat

Class-40-CICD-Steps

CICD_jenkins_Tomcat

step-1: Developer builds the code and pushes to SCM
Repo url
protocal: http/ssh
webhooks --preferred
poll scm ---wont prefer

step-2: SCM sends event(webhook) to jenkins to trigger build

step-3: jenkins job will start and takes the code from
SCM and copies to workspace.

Build:
workspace : *.java-----compile---*.class----*.war
maven : mvn clean package <arg>

Junit Test cases--test their code them self
CodeQuality testing--Jacaco,Cobertura,CheckStyle

Reports --html/json---SonarQube

After all goes well in the build phase we get a
package/target/binary/artifact/software

example: *.war

Step-4: Save the package/delivery into artifactory

artifactory:
s3
nexus
jfrog
----------CI-1,2,3,4 steps----------

CI : github,jenkins,maven,testcases,sonarQube,artifactory,

==========
CD :
Continuios Delivery(manual)/Continuios Deployment(automation)

Step-5: Take the delivery/package/artifact from artifactory and copies to workspace

Step-6: Jenkins has package in workspace and copies to
target Servers.

Ansible : Deployment for many tomcats

Step-7: Test the functionality with test cases(selineium/postman)---QA

project-2:
html
html------SCM----webhooks----Jenkins---ws---
zip------artifactory
artifactory-----jenins---ws----deploy---ansible---apache/ngin/

TestDriven development

project-3:
php
php------SCM----webhooks----Jenkins---ws---
tar------artifactory
artifactory-----jenins---ws----deploy---ansible---apache/nginx/

Class-41-CI-WAR file lab

create the build -- war
upload the war to s3

lang: java
source code: code(*.java)

create your user id in the scm
register for the github

create the repository

fork: bring repository from one user a/c to other

https://github.com/kliakos/sparkjava-war-example

github : free/paid(enterprise)

take your own project which forked and give in jenkins

java---compile+package = maven

maven: /opt/maven36/bin/

mvn clean package

/var/lib/jenkins/workspace/build-maven-war/mvn

/opt/maven36/bin/mvn

task: change the permanent bash for the jenkins

editor: vi /vim/nano
filename: /etc/passwd

set the line numbers in vi : press esc, shift+;
set nu
goto the end of file--shift+g
goto the first line--gg
goto the specific line number--esc ;shift+; , line number
goto the end of line (shift+a)
goto the start of line(shift+i)

/opt/maven36/bin/mvn clean package

package: ws/target

now copy to the s3
ec2-------s3

aws s3 cp ws/target/*.war s3://bucketname/

we can interact with s3 by using
access key /secret key (outside)
IAM role(within aws)

create the bucket

jenkins--jenkins user
aws s3 ls
fail

reason: credentials missing/iam role

create the iam role
attach iam role to ec2

aws s3 cp file s3://bucketname

aws s3 cp s3uploaddemo s3://rctcloud-artifactory/
aws s3 cp s3://rctcloud-artifactory/s3uploaddemo .

task:
copy the file from local to s3
copy the file from s3 to local

where are you : ec2
authentication : access key /secret key
i am role

what is the component : s3

war file upload to s3

where is war file ?
/var/lib/jenkins/workspace/build-war-maven/target/sparkjava-hello-world-1.0.war

Repo url : https://github.com/sreemeka82/sparkjava-war-example/

Build

/opt/maven36/bin/mvn clean package
warpath=/var/lib/jenkins/workspace/build-war-maven/target/
warfile=sparkjava-hello-world-1.0.war
bucketname=rctcloud-artifactory
aws s3 cp $warpath/$warfile s3://$bucketname/

task:
build the war file and upload to s3

technologies:

github(scm)/repository url
jenkins

Class-41-CICD-Deployment

network
app layers--installation/stop/start/imp file /logs
CICD layer
logging---ELK/Splunk
monitoring--Cloudwatch/AppDynamics/DataDog

SCM-----jenkins---maven--build--rename---upload--s3

trigger: manual
1)SCM----webhooks---jenkins
2)Deploy

task:
Enable the webhooks for the jenkins

plugins
jobs

what are the plugins
folder
github plugin
maven
ant
pipeline
docker
ansible
slack
email

install a plugin (manaul--private/Forward proxy)
github integration

AWS -----jenkins(private)----NAT-GW
On-prem-----jenkins(private)----Forward Proxy---internet

export HTTP_PROXY=URL

how do you know server is on-prem and cloud ?

yum .repos ---/*.repo --cloud provider
traceroute
agent /opt/

task:
Enable the webhooks for the jenkins
1)
first goto the scm -repo and add jenkins url

jenkinsurl/github-webhook/

jenkins-675730040.ap-southeast-1.elb.amazonaws.com/github-webhook/

http://jenkins-675730040.ap-southeast-1.elb.amazonaws.com/github-webhook/

2)
jenkins-plugin-github integration

goto the job
build-trigger

webhook is important to trigger from scm to ci server

Deployment
==========
password less setup between jenkins(jenkins)-----tomcat(root)

jenkins-aws cli---scp

s3

Tomcat
rctcloud-artifactory
Download the package from s3 to the workspace

aws s3 cp s3://rctcloud-artifactory/sparkhello-10.war .

deploy
copy the package to the tomcat(scp/ansible/shell script)
scp packagename.war username@serverIP:/opt/tomcat9/webapps/
scp sparkhello-10.war root@192.168.4.148:/opt/tomcat9/webapps/sparkhello.war

shoppingcart.war

/shoppingcart

sparkhello-10.war

/sparkhello-10

update the code : 11
CI
CD

two jobs

build--no--10,11,12
parameters
deploy

pass parameter from one job to other job
pre-req: plugin

BUILD_NUMBER----CI

tomorrow:
javaa

maven
ant
junit test cases--console
reports ---console

staticcode

Apache---logs

task:
cicd with tomcat --webhooks,post build,plugins

linux
ssh ip

/var/log/secure(messages) ---login failures

login --ps -ef | grep tomcat

user ---root ---/root/user profile files--home directory
.bashrc(alias)
.bash_profile ---user login commands/path/environment variables
task:
setup the alias/setup the login message
print the login message
/etc/profile --for all users alias/path

Class-42-Container

Containers
diff b/w container and vm ?

container=vm
vm=OS
container=no OS

vm=ssh
contaienr=exec/login

vm=apps(tomcat/apache/nginx/jenkins/database)
container=apps(tomcat/apache/nginx/jenkins/database)

ec2 =instance =ami=image(linux/amazon/ubuntu/windows)
static =OS(os binaries/os software/os packages)
vm=ami-image--------run------instance
container=image--------run------container

https://images.contentstack.io/v3/assets/blt300387d93dabf50e/bltb6200bc085503718/5e1f209a63d1b6503160c6d5/containers-vs-virtual-machines.jpg

xen =hypervisor
vmware = hypervisor
hyper-v=hypervisor ---physical server

why we need container ?

moving vm infra from one cloud to other cloud difficult
google cloud--vm
aws cloud----vm

we can move containers across the clouds.
we can move container from local to any environment.

1)container is independent on any platform.

2)Agile
time:
On-prem --1year
Cloud ---- months--vm
Container -- hours/mins

3) resource utilization

No os is packaged

4)reduction of kernel calls in the containers

5)cost ---100 apache in one vm

how do you optimize the cost in the aws ?
we converts apps from vms to the containers

6)ports ---

image---run----container

q)upgrading /existing app
container

local---container---app

containers --security patches

role: build the image

OS = app + linux
OS = database + linux
OS = jenkins + linux

OS = 4gb ram + 4 cpu
app ---1gb --1cpu

resource optimization

vm = os =4cpu---4gb ram
applicaiton=1

container=
4 application
each app-1gb-1cpu

4gb ram
4cpu
OS=binary(os)+app(1cpu/1gb ram)

os = raw os/source code
kernel

unix---source code=1mb=scratch/binary

Redhat--source code+redhat kernel =4gb
ubuntu-- source code+ ubuntu kernel=4gb
IBM -- source code + ibm kernel =4gb

redhat os(binary+kernel) + app=8gb=vm

source code+app=1GB=container--->kernel
scratch+app=

we dont use os = scratch+app(10mb)=11mb

app(10mb)
vm=4gb= 4gb+10mb

Class-43-Dockerfiles

vm
ec2=instance=ami(image)=app+agents (logging/monitoring/security)

container
container=image=(app+agents)

scratch+app1----docker engine--->kernel(os)
scratch+app2----docker engine--->kernel(os)
scratch+app3----docker engine--->kernel(os)
scratch+app4----docker engine--->kernel(os)

Container
docker----all providers supports
rocket
podman

Docker-19.x

Community Edition
Enterprise

Installation of Docker--18.x
yum install docker -y
apt-get install docker -y
dnf install docker -y
pip install docker -y

start
systemctl start docker
service docker start

running or not
ps -ef | grep -i docker
systemctl status docker

installation : Docker engine(server)

client: docker

docker commands talks to the docker engine server.

docker(client)----------DockerEngine(server/daemon)

docker version
docker images

docker hub ---docker maintains registry for the images.

Tomcat installation
vm
ami=ec2=java+tomcat----ami---share--private/public
container

Dockerfile ---github

a=10
env a 10

Dockerfile instructions

FROM
ENV
ARG
RUN
ADD---downloads/unzip automatically
COPY---manul copy (noraml cp)
EXPOSE
CMD/Entrypoint

tomcat----from ---
openjdk:15-jdk-oraclelinux7---FROM oraclelinux:7-slim--oraclelinux:7-slim

name:tag

test : default keyword for the tag: latest

httpd---Dockerfile----FROM debian:buster-slim

WORKDIR -- pwd

date;\
ls;\
ps;\
free;\
df ;

tomcat
tomcat dockerfile----openjdk dockerfile----oracle-linux-dockerfile---scratch

httpd
httpd dockerfile---debian dockerfile----scratch

keywords:

docker client
docker engine
docker hub-----image registry (public/private(ecr))
Dockerfile
name:latest

Dockerfile
FROM sourceimage/baseimage
ENV
ARG
LABEL
MAINTAINER
WORKDIR
PWD
COPY
ADD
RUN
EXPOSE
CMD/ENTRYPOINT

install the shoppingcart.war in tomcat
vm

ec2----java---tomcat---shoppingcart.war(webapps)

docker

Dockerfile
FROM tomcat
COPY shoppingcart.war xxx/webapps/

apache--static website

static.zip

Dockerfile
FROM httpd

ADD static.zip xxxx/www/html

Class-44- Networks-Running container

Dockerfile
FROM tomcat
COPY shoppingcart.war xxx/webapps/

apache--static website

static.zip

Dockerfile
FROM httpd
ADD static.zip xxxx/www/html

flow:
Dockerfile
image
container
DockerEngine

Write the Dockerfile
Build the image
Run the image

Dockerfile----Build----image-----Run----Container

lab: take one ec2 install the docker engine
Dockerfile----Build----image-----Run----Container

docker images ---DockerEngine----local registry/repo

docker pull imagename:tagname
docker pull busybox

docker pull tomcat:jdk11-openjdk-slim

docker history -- to see layer

container run

docker run imageName:Tag

docker run httpd

```
for i in {1..100}
do
 docker run -d httpd
done
```

stop docker container

docker stop <containerID> (dokcer ps)

remove container
docker rm <containerID>

image remove
docker rmi imageID

docker stop $(sudo docker ps -aq)

stop all running containers
docker stop $(docker ps -aq)

remove all containers

docker rm $(docker ps -aq)

install the docker engine
start the docker engine

docker images
docker pull imageName
docker pull tomcat
docker pull busybox

docker history imageID

docker run -d imagename
docker run -d httpd

docker ps
docker ps -a

docker stop containerID
docker rm contaienrID

docker rmi imageID

vpc----ec2-----docker engine----container---network---IP/Subnet/RouteTables/Gateway/NAT

how many types of network and what they for ?
Docker creates it own network(pvt)
bridge --default
host
overlay----------multi nodes/ec2/docker engines
macvlan

none

Class-45- Build image_run image

Why we do use Container technology ?

development and delivery goes agile mode

ec2--installed the docker engine

ran the containers

docker run -d imagename

outside: Docker hub --registry(image/docker registry/container registry)

image pull
history
container stopped
container remove

image remove

all containers----docker rm $(docker ps -aq)
all images ----- docker rmi $(docker ps -aq)

https://github.com/ykarthickeyan/DockerMavenHelloworld/blob/master/pom.xml

docker ps
docker ps -a

image: pack of application binaries

Dockerfile----build---image------run----container

static website
vm=ec2(ami)-----apache----/var/www/html/
download the static website ----unzip----copy---/var/www/html/

container
docker image(apache)-----run --see the website

containers will be in private network and also container port also .

if i want to acccess container from outiside, we cant.
until we do port mapping (hostPort:containerPort)
ex:

docker run -d -p 8080:80 httpd

18.138.22.254:8080

task: run the multiple apache containers

docker run -d -p 32565:8080 jenkins

task: run the images
httpd
tomcat
jenkins
nginx
nodeJs
busybox

Host the static website container

Dockerfile
FROM httpd
#ADD url /var/www/html/
COPY . /usr/local/apache2/htdocs/

sourceCode + Dockerfile
docker build -t static-web .

Build the docker image----SouceCode+Dockerfile
docker build -t imagename:tagName . (current context)

Run
docker run -d -p(port mapping) imagename:tagname

multi stage dockerfile
https://github.com/dstar55/docker-hello-world-spring-boot

Class-46- CICD-K8-Intro

Host the static website container

Dockerfile
FROM httpd
#ADD url /var/www/html/
COPY . /usr/local/apache2/htdocs/

sourceCode + Dockerfile
docker build -t static-web .

Build the docker image----SouceCode+Dockerfile
docker build -t imagename:tagName . (current context)

Run
docker run -d -p(port mapping) imagename:tagname

multi stage dockerfile
https://github.com/dstar55/docker-hello-world-spring-boot

copy and create the image
persist

VM=

Developer----Code----SCM----maven+pom.xml-------Jenkins---Build--pkg--upload---artiffactory

maven project+pom.xml+Dockefile-------SCM---Jenkins---Build
maven build ---war file
Docker image build---docker image

Dockerfile

```
FROM tomcat
COPY xxxx.war xxx/webapps/
```

static code+Dockerfile

```
FROM httpd
COPY . /usr/local/apache2/htdocs
```

php code+Dockerfile

```
FROM httpd
COPY phpcode /usr/local/apache2/htdocs
```

nodeJs +Dockerfile

```
FROM node:versionTag
RUN npm build /install
```

Based on the language Dockerfile base image

node---nodeImage
php --- apacheImage
static--Nginx/apacheImage
java----tomcat/javaimage

Build the Docker image
language: static code
Build : jenkins(Docker engine+git+zip)
Job: SCM--epo---execute build----docker build -t imagename:build-no
image upload to the container registry(push)

1)create the empty repo in the scm
scm
https--username and password
ssh ---public key

local repo(add+commit+push) ----------scm

local repo=workspace
add -------staging area
commit
push-----scm

files--localworkspace----add---commit---push

empty: git init
add to local repo with remote repo:

git remote add origin https://github.com/maheswargoud/static-microservice.git

app---
frontend---scm--repo----js----apache

backend----scm--repo---java---package----war---tomcat

Class-47- CICD-Docker-Jenkins

steps:

1)Create the repo in the SCM (central/remote repo)--ssh
instruction: copy
pre-req for ssh : update the id_rsa.pub to the remore repo

2)create the local repo in the local (local repo)
echo "# static-app-micro" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:maheswargoud/static-app-micro.git
git push -u origin main(ssh)

local repo----------remote repo
git(client) ----------github(server)

git init ---it creates the empty repo
git add remote origin url : add the remote repo to the local repo

files create(index.html and Dockerfile)
versioning : shanum/commitID

localrepo(workspace)-----staging----commit -----push---scm

ws---git add ---staging----git commit -m "msg"----git push

github----CodeCommit(aws)/Gitlab

validation: in the scm you will have few files(index.html/Dockefile)

3)CI sever(jenkins)
job ---scm ---build--steps

pre-req; update the jenkins ssh keys to the scm

jenkins is running with jenkin user on the vm
so we have to generate ssh key to the jenkins user

validation:
Build the job and see the files in the workspace(index.html/Dockerfile)

4)build the docker image on workspace
docker build -t imagename:tag

docker build -t static-app:$BUILD_NUMBER .

pre-req: install the Docker Engine on jenkins server

docker-engine will be running with root user
jenkins will be jenkins user
if jenkins tries to run docker commands, you will get error
to resolve add jenkins to the sudoers
/etc/sudoer

vi /etc/sudoer
visudo

usermod -a -G wheel jenkins

debug: /var/log/messages or secure

/etc/sudoer.d/90---xx
jenkins

1)update the code in the repo(ssh)
2) pull the code in the jenkins(ssh)
3)build the docker image in the jnekins(sudo)

git pull and git fetch ?

merge and rebase ?

Class-48- CICD-Build and Run Tomcat Image

5) Push the image from jenkins to the Container Registry
Container Registry: own registry (private), Docker hub,ecr,nexus,jfrog

jenkins(ec2)----build---login--push-----ECR

pre-req: create the repo in the ECR
to login to the ec2(jenkins) to the ecr we need have permissions
(iam role of jenkins(s3+ecr))

aws ecr get-login-password --region ap-southeast-1 | sudo docker login --username AWS
--password-stdin 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com
sudo docker build -t
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/static-app-micro:$BUILD_NUMBER .
sudo docker push
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/static-app-micro:$BUILD_NUMBER

docker tag sourceimage tagetimage

docker tag static-app:7
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/static-app-micro:7

login to the jenkins
docker images
docker login -----credentials(we will bring from ecr)
docker push

validate : check the images numbers in the ecr
total build has to be automated with webhooks

docker images storing location: /var/lib/docker/

task: run the ecr-image in one ec2
pre-req: instal the Docker-engine, assign iam role,login,pull,run

docker run -d -p 8080:80 ecr-url/reponame:tagimage

docker run -d -p 8080:80
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/static-app-micro:7

push the code
task: build and run the tomcat image

localworkspace: Dockerfile+shoppingcart.war(java)
SCM

create the job

aws ecr get-login-password --region ap-southeast-1 | sudo docker login --username AWS --password-stdin 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com
sudo docker build -t 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/shoppingcart:$BUILD_NUMBER .
sudo docker push 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/shoppingcart:$BUILD_NUMBER

Basics of docker commands
image build
image run
container check
container remove --docker rm
stop
image remove --docker rmi
login to the container
docker exec -it containerID bash/sh

docker volume/storage:
containers non-persistence

docker run -d -p hostPort:containerPot -v hostPath:containerPath

Class-49- CICD-ECS/EKS

Container runtime environment : Docke engine---ec2

container : ec2
drawback:
managing of nodes our responsibility
100 containers--manual

docker run
docker run
docker run
xxxxx100 times==port volumes....secrets/..
scaline in/scale up --increase
scale out/scale down---decrease

Ecs/Eks
=======
Elastic Container Service----AWS
Elastic Kubernetes Service--Opensource--using by AWS

nodes will take care by aws
replicas=100
containers/pods will have limits(cpu/ram/network...)
pod autoscaling automaically(cpu/ram/load)

ECS/EKS/Openshift/Kuberenetes = Run+manage=Orchestration

Orchestration=CD
ECS= free + nodes+containers(pods/task(fargate))
cost
node-ec2
task--fargate

Cluster
Task/Pod/Container=running

Apache-10 task--Service---load balancer(target group)--ALB--Route53
Tomcat-20 task--Service---load balancer(target group)--ALB--Route53
Python- 5 task--Service---load balancer(target group)--ALB--Route53

group of tasks=service

ECS
scm---code+dockefile----image build(jenkins)---ecr-----TaskDefinition(image)
----Service(tasks)-------lb---route53

for every image build we have to update the td and service
task definition= will have image info and limits and ports and env
service = will have running of task definition (tasks)

SCM
Jenkins
Build imge ---ECR
next
Create the ECS cluster
Task definition
create the service and attach to taget group(lb(alb/nlb/clb))
diff b/w fargate cluster and ec2 ?
fargate = pay cost for tasks=nodes manged by aws
nodes = pay for nodes=nodes managed by us

Cluster
Task Definition
name:
limits
Contaier =ecr image url

498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/static-app-micro:7

image build ---ECR

image run ---ECS -- Cluster(fargate/ec2)
ECS Cluster
Task Definition -image location/limits/ports/logs/env
Service (pre-req-lb)---Task Definition---attach --load balancer(tg)
register: tasks

docker basic commands

Class-50- k8 introduction

why do we go for container technology ?

resource usage
any platform
bootup fast
agiie
development vs infra

orchestration
To manage multiople nodes and to manage multiple containers
k8 architecture
---
why k8 ?
why not ecs ?
AWS only have ecs

local :kubernetes--single node cluster --minikube---testing/development
On-Prem ---vm--k8 installation
AWS --vm--k8 installation/EKS(only run apps)
AKS
Gke

k8--container orchestration
opensource
kubernetes.io
labs/learning: katakoda

k8 cluster size ?---20 nodes prod
70+ apps
1000+ pods

K8 architecture

installatin -local/prod
authentication
kubectl
practice area:
katakoda/kubernetes.io

Class-51- CICD into the kuberntes

Developer ---Code----------SCM

Operations --Dockerfile----------SCM

Jenkins ----Build the image(Dockefile+code)
ECR - push the docker image : build number

CD
kuberntes

kubectl commands for deployment of image --imperative
or
yaml/yml files declarative (ops) for deployment of image

Deployment of image(objects)
==========
Pod--unmanaged
Deployment-------most used
ReplicationSet/ReplicationController(old)
DaemonSet
StatefulSet

Networking(objects)
==========
Service(To group pods and to expose to outside)
Ingress

Storage(objects)
==========
StorageClass
PersitentVolume(like ebs volume/disc)
PersistentVolumeClaim----attaches/mounts to deployment objects

Secrets(object)/Envionment Variables
==========
ConfigMap(plain)
Secret (encrypt)

CICD
SCM----image----k8----deployment objects(pod/deployment/rs/rc/ss/ds)

DaemonSet ----image----on each node one pod
(monitoring/logging agents)

pod

Deployment unit
unmanaged

Deployment---pods(container)
---
managed
replica=10---ReplicationSet --to maintain the replicas(no of pods)

StatefulSet

To save the container data(stateful apps)

SCM
index.html ----helloworld
Dockerfile

Jenkins
Build the image:1
push the image:1 to the ECR

k8--deployment
==========
hello.yaml

```
apiVersion: v1
kind: Deployment
metadata:
  name: helloworld
spec:
  replicas: 1
  strategy: rollout
```

```
  containers:
    image: ecr-repo/image:1
    name:
    port:
    volumes:
    env :
    secrets:
```

kubectl apply -f hello.yaml
k8 worker which image will run ?

helloworld(Deployment)-----Pods-----Service---access---helloworld
deployment(image tag change)
---kubectl apply -f xx.yml
deployment
sevice (to access the pods/entrypoint to the pods)

Class-52-monolithic microservices

Questions
what is the diff between pod and deployment ?
pod is unmanaged
deployment is managed

what is Pod ?
Pod is container
Deploment of unit

what is kubernetes architecture ?
what is kubectl and kublet ?
why do we need kube config ?
what is kube-proxy ?
what is service ?
microservice(orchestration) vs monolithic(vm)

Monolithic
Google --application---modules(20)
google.war ---Tomcat(webapps)----Vm----load balancer----R53

if vm goes down all apps will get affect
repo=1
package=1
language=java

microservice
each module =application(20)--repo=20,package=20
frontend/home=reactJS
youtube=python
maps=java
gmail=php
sso (Single SignOn)

frontend(service)---SSO(service)---service discovery
maps ---sso
gmail --sso
youtube---sso

Google

2nd Example
Monolithic
Facebook(php) -----run----Apache(lamp/wamp)--vm
chat
images
likes
pages
comments
groups
share
search
sso

repo = 1
package =1

MicroService
repo=9--Dockerfile
package = 9 docker images

fronend(app/svc)---login(app/svc)----feeds (app/svc)-----share(svc)
internal service

what is microservices ?
running on top of k8 =application
we are grouping the pods into a service (application)
one application(service) will connect to the other application(serivce)
service discovery.

frontend(api)----pods(frontend image)--10 pods
share--service/app ---pods ---10 pods

frontend(api/svc/app)-----backend(api/svc/app)

apiGw= to protect the apis(applications)
if one app want to connect with other app with security we will have apigw

what is apigateway ?
what is microservice ?(running apps independently and giving service)
benefits : independent on each service /availability/maintainance easy
what is service discovery ?

api=svc=application
api=security (bearer token)=postman testing

5 mins video of postman

Cluster creation
vm=manual installation
eks=installation aws

Class-53-EKS installation

1)aws vms-ec2(min-2 nodes)
base image
Container Runtime : Docker Engine(install)
Master : etcd,apiserver,scheduler , controller manager
worker : kublet , kube-proxy

manually download the binaries and install
On os we have a services
applications: pods

kuberntes in Hardway cluster setup

2)kubeadm (min 2 nodes)
master:Docker-engine, pods(etcd,apiserver,scheduler,controller)
worker: Docker-engine,pods(kublet,kube-proxy)

install the kubeadm on both maste and worker

master: kubeadm init (it gives token)
worker : kubeadm join token...(take the token from master)

3)EKS (Dont manage master by ourself--AWS will manager master)
Take care only the workers(eks optimized ami)(already componets intalled)

eksctl (to create the cluster)

4)kops
5)Rancher

setup the eksctl cluster

take the ec2
attach iam role--administrator
install:
eksctl
curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname
-s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin

commands - run

/home/ec2-user/.local/bin:/home/ec2-user/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
eksctl /tmp/

commands ---PATH

,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"

iam-authenticator
curl -o aws-iam-authenticator https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.8/2020-09-18/bin/linux/amd64/aws-iam-authenticator

,aws cli

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install

create the cluster

eksctl create cluster ----

eksctl create cluster \
--name prod-cluster \
--version 1.18 \
--region ap-southeast-1 \
--nodegroup-name linux \
--nodes 4 \
--nodes-min 2 \
--nodes-max 4 \
--with-oidc \
--ssh-access \
--ssh-public-key eksadmin \
--managed

Cluster Connect
EKS
kubectl get nodes
kubectl get pods
kubectl get service

kubectl get deployment
kubectl get secret
kubectl get configmap
kubectl get pv
kubectl get pvc
kubectl get rc
kubectl get rs
kubectl get daemonset
kubectl get statefulset

Class-54-EKS-connect cluster

Connect to the cluster
laptop
kubectl
kubeconfig (user home directory/.kube/config)

export kUBECONFIG=path/dev_config
(user home directory/.kube/dev_config)
export kUBECONFIG=path/stg_config
kubectl xxx
(user home directory/.kube/stg_config)
export kUBECONFIG=path/preprod_config
(user home directory/.kube/preprod_config)
export kUBECONFIG=path/prod_config
(user home directory/.kube/prod_config)

if one config we switch context with kubectl command
kubectl config set-context clustername
ec2
kubectl
kubeconfig

note: if we wanted to connect to the external api from our private subnets
if external api not allowed to 0.0.0./0, they can whitelist only NAT

eksctl ---cloudformation (automatically creates the vpc and nodes and eks)

eksctl delete cluster --name=prod-cluster

aws eks --region ap-southeast-1 update-kubeconfig --name dev-cluster

it will generate the kubeconfig

kubeadm
---
kubeadm init
/etc/kuberntes/admin.conf

copy this file to the user hhome directory/.kube/config
config

k8-network---docker network---weavent/flannel/callico

cluster---master----not ready--weavent(network)-----ready

NameSpace
kubectl get pods -n kube-system
apiserver
etcd
scheduler
controller
kube-proxy

kubectl get pod---default namespace

To know the nodes
kubectl get nodes

what is namespace ?
Group of related objects deployed into the spefic name

kubectl create ns <nsname>
web-ns.yaml

```
apiVersion: v1
kind: NameSpace
metadata:
  name: web
spec:
```

```
apiVersion: v1
kind: Namespace
metadata:
```

```
  name: shoppingcart
```

kubectl get ns web -o yaml >test.yaml

k8(master)(apiserver)

kubectl(config)
export KUBECONFIG
kubectl xxxxx

kubectl get ns
kubectl get po -n kube-system
kubectl get nodes
kubectl get po --- default
kubectl create ns <name>
kubectl get ns name -o yaml

Class-55-k8-Deployments

In k8 we will run applications
application run ?as a pod --Container---application ---image--Dockerfile
if i want to deploy image we have diff objects in k8
we will run apps in the namespaces

Deployment of app

image
own image
docker hub

Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
    image: repourl/image1:tag
    name:  test
  ports:
    containerPort:
```

```
  limits:
    memory: 512mb
    cpu: 1
  request:
    memory: 256mb
    cpu: 0.5
    image: repourl/image2:tag
    name:  test
  ports:
    containerPort:
  limits:
    memory: 512mb
    cpu: 1
  request:
    memory: 256mb
    cpu: 0.5
```

```
apiVersion: v1
kind: Deployment
metadata:
  name: test
spec:
  replicas: 1
  label:
  container:
    image: imageurl
```

Pod healthcheck
livenessprobe--if health fails--it will restart
readinessprobe---if health fails -- it will stop traffic

Deployment----image----pod --ReplicaSet/Contoller --replica=1

```
apiVersion: v1
kind: ReplicaSet
metadata:
spec:
  containers:
    image:
```

```
apiVersion: v1
kind: DaemonSet----based on the nodes--on each node one pod--pod
metadata:
spec:
  containers:
    image:


apiVersion: v1
kind: Statefulset
metadata:
spec:
  containers:
    image:
  volumes(it will have locally in container)
```

how do you deploy application in k8 ?

we have to creat the image (Dockerfile+code)
we will build the image
save/push the image to the repo

create the objects (Pod/Deployment/Rc/Rs/Ds/SS)
in Deployment(will have the imageurl/secret(extenal repo)/ecr(iam role))

kubectl apply web-deployment.yaml

staticpods -- copy the yaml at the kubelet path(/etc/kublet/**)
apiserver
etcd
scheduler
controller
kube-proxy

```
apiVersion: v1
kind: Pod
```

```
metadata:
  name: busybox-sleep
spec:
  containers:
  - name: busybox
    image: busybox
    args:
    - sleep
    - "1000000"


apiVersion: v1
kind: Pod
metadata:
  name: nginx1
  namespace: web
spec:
  containers:
  - name: nginx1
    image: nginx:12
```

problem:
pod is not running

1)describe
kubectl get po
kubectl describe po <podname>

kubectl logs podID/name
kubectl logs -f podId/name

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
```

```
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/

task: limits and request
https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource/

describe
logs
top
top --nodes
metrics sever
nodeaffinity/nodeselector
taint
statefultset/pv

task: we will deploy our own image
access outside
persitence voulme

Class-56-connecting apps in k8

EndUser Flow/Connecting to the apps in k8
app is running inside the pod(Container)

vm
==========
VPC ---public----private(app)-----data(db)
Route53---public(alb---rules(path/host))--tg----private(app)-----data(db)

K8
==========
VPC----public------private(k8-cluster)--------data(db)

--sevice-----pod(intranet)--------

apiVersion: v1
kind: Service
metadata:
  name: test-service
spec:
  select: deployment(label)---this label is deployment/pod/ss/ds/rs
type: nodePort

as per above def : we have created svc object
type : ClusterIP (privateIP)--default
type: nodePort
nodeIP:nodePort

under one service we can have many pods

service(Nodeport)------pods(100)

apiversin: v1
kind: Service
metadata:
  name: test-svc
spec:
  selector: label of pod(deployment)
type: LoadBalancer (only works in the cloud)
automatically load balancerr(public) will get created
loadbalancer-----nodes(pods)

app1(deployment)----serivce(loadbalancer)---lb
app2(deployment)----serivce(loadbalancer)---lb

service: none(headless)

what are the service types in the k8 ?
clusterIP
nodePort
Loadbalancer
None

Cons: more load balancer will get created here.

type: Load balancer (alb)
path rules
host rules

IngressController(deployment(pod--image---nginx))
kong
Istio

rueles(path/host) we will write inside the nginx

service(loadbalancer--nlb)---pod(rules(path/host))
tcp------pod(nginx--http/https/rules/path/host)

nginx=acts as alb

https://aws.amazon.com/blogs/opensource/network-load-balancer-nginx-ingress-controller-eks/
apiversion: v1
kind: Ingress
metadata:
 name: test-ing
spec:
 rules:
  path
  host : wwww.app1.com backendService: app1-svc
www.app1.com ----Route53---NLB---svc(ingress-controller)---
pod(ingress-controller-nginx---rules)--------svc(app1)----deployment(app1)

Deploy ingress-controller
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-0.32.0/deploy/static/provider/aws/deploy.yaml

Deployment--image--nginx-controller
Service- load balancer--nlb

NLB-----Svc----Deployment(ingress-controller-nginx)

appliction
Deployment
service
ingress

pods(apple/bana/svc)
$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/apple.yaml
$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/banana.yaml

kubectl create -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/example-ingress.yaml

Ingress---rules---backend--apple--svc---selector--apple-pod

www.test.com ---NLB---ingresscontroller--nginx(rules)
rules

namespace--test
ingress---backendservice:
svc---selector--label
deployment

$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/apple.yaml

$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/banana.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: example-ingress
```

```
  annotations:
#    ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
    nginx.ingress.kubernetes.io/force-ssl-redirect: "false"
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: anthonycornell.com
    http:
      paths:
        - path: /apple
          backend:
            serviceName: apple-service
            servicePort: 5678
        - path: /banana
          backend:
            serviceName: banana-service
            servicePort: 5678
        - path: /pet
          backend:
            serviceName: pet-service
            servicePort: 5678


Ingress(rules)----SVC---Pod


apiVersion: v1
kind: Pod
metadata:
  name: busybox-sleep
spec:
  containers:
  - name: busybox
    image: busybox
    args:
    - sleep
    - "1000000"


apiVersion: v1
kind: Pod
metadata:
```

```
  name: nginx1
  namespace: web
spec:
  containers:
  - name: nginx1
    image: nginx:12
```

problem:
pod is not running

1)describe
kubectl get po
kubectl describe po <podname>

kubectl logs podID/name
kubectl logs -f podId/name

```
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2 # tells deployment to run 2 pods matching the template
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/

task: limits and request
https://kubernetes.io/docs/tasks/configure-pod-container/assign-cpu-resource/

describe
logs
top
top --nodes
metrics sever
nodeaffinity/nodeselector
taint
statefultset/pv

task: we will deploy our own image
access outside
persitence voulme

Class-57-build and deploy our own image

Build and deploy our own image into the kubernetes
==========
CI
Code(staticcode) +Dockerfile
push the code to the SCM(branches--main)
jenkins--job(build the docker image): staticapp:buildnumber
push the ecr (staticapp:buildnumber)

CD
Create the static-deployment.yaml (image: ecrrepourl/staticapp:buildnumber)
Jenkins(kubectl+kubeconfig)
bring the deployment yaml inside the jenkins
kubectl apply -f static-deployment.yaml
we will be doing the regular deployment(so we will change the deployment yaml)

Access
Service
Ingress

Project
==========

ns
service---selector
ingress---backendservice

task: in jenkins with jenkins user kubectl
https://kubernetes.io/docs/tasks/tools/install-kubectl/
jenkins user : kubeconfig
aws cli
aws-iam-authenticator
https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html

space issue:
df -h (fine the file system)
go inside the file system
du -sm * | sort -rn

aws eks update-kubeconfig --name dev-cluster

aws eks --region ap-southeast-1 update-kubeconfig --name dev-cluster

error: You must be logged in to the server (Unauthorized)

arn:aws:sts::498449435961:assumed-role/jenkins/i-0373238f4cf395559

/sbin:/usr/sbin:/bin:/usr/bin
/sbin:/usr/sbin:/bin:/usr/bin

adding new user/new role to the eks
https://aws.amazon.com/premiumsupport/knowledge-center/amazon-eks-cluster-access/

aws s3 ls (ec2)

An error occurred (AccessDeniedException) when calling the DescribeCluster operation: User: arn:aws:sts::498449435961:assumed-role/jenkins/i-0373238f4cf395559 is not authorized to perform: eks:DescribeCluster on resource: arn:aws:eks:ap-southeast-1:498449435961:cluster/dev-cluster

Build the image---push the image to ecr

write the deployment yaml
kubectl
deploy

ns--staticapp-k8
svc
ingress

$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/apple.yaml
$ kubectl apply -f
https://raw.githubusercontent.com/cornellanthony/nlb-nginxIngress-eks/master/banana.yaml

```
kind: Service
apiVersion: v1
metadata:
  name: staticapp-svc
  namespace: staticapp-k8
spec:
  selector:
    app: staticapp-deployment
  ports:
    - port: 80
```

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
#    ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/ssl-redirect: "false"
    nginx.ingress.kubernetes.io/force-ssl-redirect: "false"
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: staticapp.rctcloud.in
    http:
      paths:
```

```
      - path: /apple
        backend:
          serviceName: staticapp-svc
```

Class-58-CICD-k8-Deploy

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app:  staticapp-svc
  name:  staticapp-svc
  namespace:  staticapp-k8
spec:
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  selector:
    app:  staticapp-deployment
```

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: staticapp-ing
 namespace: staticapp-k8
 annotations:
   kubernetes.io/ingress.class: nginx
spec:
  rules:
 - host: staticapp.rctcloud.in
   http:
     paths:
     - path:
       backend:
         serviceName: staticapp-svc
         servicePort: 80
```

capture the logs of ingress controller

103.110.170.82 - - [17/Nov/2020:04:04:02 +0000] "GET / HTTP/1.1" 503 600 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36" 475 0.000 [staticapp-k8-staticapp-svc-80] [] - - - - 914b9276690833ce303a6068b0dfddf3
103.110.170.82 - - [17/Nov/2020:04:04:02 +0000] "GET /favicon.ico HTTP/1.1" 503 600 "http://staticapp.rctcloud.in/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36" 428 0.000 [staticapp-k8-staticapp-svc-80] [] - - - - a111f99a7d97c17435568c7f7889b7f9

Deployment with the jenkins
image

code+Dockerfile

```
#image
#docker build -t tagname .
docker build -t
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/staticapp-k8:$BUILD_NUMBER
#ecr login
aws ecr get-login-password --region ap-southeast-1 | docker login --username AWS
--password-stdin 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com
#ecr push
docker push
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/staticapp-k8:$BUILD_NUMBER
```

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: staticapp-deployment
spec:
  selector:
    matchLabels:
```

```
    app: staticapp-deployment
  replicas: 2
  template:
    metadata:
      labels:
        app: staticapp-deployment
    spec:
      containers:
      - name: staticapp-deployment
        image: 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/staticapp-k8:7
        ports:
        - containerPort: 80
```

```
cp /var/lib/jenkins/staticapp/static-deployment.yaml .

sed - to replace the keywords in a file

sed -i "s/staticapp-k8:[0-9.]*/staticapp-k8:$BUILD_NUMBER/g" static-deployment.yaml

kubectl apply -f static-deployment.yaml

sed -i "s/staticapp-k8:[0-9.]*/staticapp-k8:10/g" static-deployment.yaml
```

Admin

code---scm---jenkins---ecr----deployment(sed the image)---deploy

Enduser
url----r53---nlb---ingress-ctrler--svc----ingress-controller--nginx---rules--
appsvc----appdeployment---app pods

monolithic
2 vms--Tomcat
using anisble
deployment

2 vms --apache
using ansible
deployment

2vms --nginx
using ansible
deployment

k8
tomcat
nginx
apache

CI Commands
==========
#CI
#image
#docker build -t tagname .
sudo docker build -t
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/staticapp-k8:$BUILD_NUMBER .
#ecr login
aws ecr get-login-password --region ap-southeast-1 | sudo docker login --username AWS
--password-stdin 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com
#ecr push
sudo docker push
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/staticapp-k8:$BUILD_NUMBER

CD Commands
==========
#CD
#Bring the yaml file to the workspace
cp /var/lib/jenkins/staticapp/static-deployment.yaml .
#Replace the image tag
sed -i "s/staticapp-k8:[0-9.]*/staticapp-k8:$BUILD_NUMBER/g" static-deployment.yaml
#deploy the application

kubectl apply -f static-deployment.yaml

Class-59-CICD-k8-Deployment

deployment of shoppingcart

namespace
svc
ingress

deployment

codebase(java---shoppingcart.war)+Dockerfile(Tomcat)
COPY
Docekrfile

FROM tomcat
COPY shoppingcart.war

code(java)(pom.xml)+Dockerfile
maven build ---war

where are we deploying ?
k8(image)
java --package---maven build/ant build/graddle build
war/jar/ear
Dockerfile

code---jenkins job--maven package--war --Dockerfile --copy war

Dockerfile
FROM A as a
COPY src /opt
COPY pom.xmo /opt
mvn package

FROM tomcat
COPY from a = war xx/webapps

docker build -t hello:1 .

Maven build container
https://raw.githubusercontent.com/dstar55/docker-hello-world-spring-boot/master/Dockerfile

```
FROM maven:3.5.2-jdk-8-alpine AS maven_build
COPY pom.xml /tmp/
COPY src /tmp/src/
WORKDIR /tmp/
RUN mvn package

#pull base image
FROM openjdk:8-jdk-alpine
#expose port 8080
EXPOSE 8080
#copy hello world to docker image from builder image
COPY --from=maven_build /tmp/target/hello-world-0.1.0.jar /data/hello-world-0.1.0.jar
#default command
CMD java -jar /data/hello-world-0.1.0.jar
```

deployment of java(springboot)--jar
java -jar xxx.jar
tomcat(8080)

```
#CI
aws ecr get-login-password --region ap-southeast-1 |sudo docker login --username AWS
--password-stdin 498449435961.dkr.ecr.ap-southeast-1.amazonaws.com
#image build
sudo docker build -t
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/springboot:$BUILD_NUMBER .
#image push
sudo docker push
498449435961.dkr.ecr.ap-southeast-1.amazonaws.com/springboot:$BUILD_NUMBER
```

#CD

#CD
#Bring the yaml file to the workspace
cp /var/lib/jenkins/springboot/springboot-deployment.yaml .
#Replace the image tag
sed -i "s/springboot:[0-9.]*/springboot:$BUILD_NUMBER/g" springboot-deployment.yaml
#deploy the application

kubectl apply -f springboot-deployment.yaml

Login to the pod
kubectl exec -it <podID> bash -n namespace

Two appliction
staticapp ---jquery
springboot----java

Docker-compose
==========
is not for prod
learning /local: docker-compose

important file: docker-compose.yml
image build /run
multiple images build/multiple container run/linking /volume--local

install the docker-compose

docker-compose up -d ---docker-compose.yaml ---instructions
docker-compose ps
docker-compose down
docker-compose status

how do you start one contaienr after another (name1 starts at last)
services:
  name1
   build:

    depends on name2,name3
  name2
   image:

name3

what is docker-compose
i want to build/run multiple images in the local
docker-compose.yml
services:
depends

code+docker-compose.yaml(local)+Dockerfile

docker--build image
docker-compose--local
k8--prod

monolithic to micro
vm ------k8
multiple vms------k8

install tomcat on vm
yaml

pre-req: vm (tasks/playbook/role-tomcat)
1)ssh login --pem file/ssh password less setup(ssh keygen)
2)download the java(wget/curl--commands---shell/get_url)-rpm
3)install the java (rpm -ivh *.rpm)--command/shell
4)download (wget/curl---command/shell/get_url)--zip--source/target
5)unzip(command--shell/unarchive)--src/target
6)chmod(command--chmod---shell)--755 /opt/tomcat(notify)--7th
7)start tomcat(startup.sh)--handler
8)download war (src/target)
9)stop tomcat(shutdown.sh)
10)copy war (webapps)--notify

what is handler/nofier
what is task
what is playbook(group of tasks)
what is role (unit of objective)--installing tomcat

ansible agentless

modules
download(get_url)
copy

apache (role/playbook/task)
pre-req: vm (password less setup)

package: yum install httpd----linux
package: apt-get install apache2 ---debian(ubuntu)

service: name: httpd status: start

```
---
- hosts: apache
  sudo: yes
  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enabled mod_rewrite
      apache2_module: name=rewrite state=present
      notify:
        - restart apache2

  handlers:
    - name: restart apache2
      service: name=apache2 state=restarted
```

ansible agent less
tasks(commands)
playbook--yaml
notify/task/handler

vm
application
moniotirng
logging

ansible

k8
application
monitoirng
Logging

Class-60-Ansible
Ansible

We are using here for vms mgmt
To install packages on vms
To stop /start /upgrade on vms
Agentless
using ssh (password less setup)

all linux activities/windows tasks we can do using ansible

OS layer---infra layer
Middleware layer--Apache
Application layer---html/php
Monitoring agent--ops layer
Logging agent --ops layer

playbook=group of task = yaml

ls ---task/play
useradd
download
extract
copy
remove

ansible-master(controllers)
nodes(managed nodes)

opensource
paid: ansible tower(redhat)--GUI
python

setup--master
=========
Enable EPEL
yum install ansible or pip install ansible

setup password less setup between master and nodes
generate the keys at the master and copy to the nodes

nodes
==========
tasks

ansible ---ad-hoc commands

ansible -m ping <servergroup>

ansible-playbook playbook.yml

ip address (inventory(cmdb)--group of servers)
static inventory
dynamic inventory

/etc/ansible/hosts --inventory

/etc/hosts(local dns)--C:\Windows\System32\drivers\etc\hosts

inventory
[web]
10.x.x.x
10.23.x.x
[db]
9.x.x.x

[tomcat]
xxxxx

ansible -m ping all
ansible -m ping tomcat

playbook.yml

```
---
- hosts: <servergroup>
  become: yes (sudo/root)
  become_user: jenkins

  tasks:
  - name: install the apache
    yum:
      name: {{package}}
      state: present

  - name: start the apache
    service:
      name: httpd
      state: started


  - name: Download the website
    get_url:
      url: http:/xxxx.zip
```

```
    dest: /tmp/xxx.zip


  - name: copy from tmp to opt
    copy:
    src:
    dest
```

vm

role-vm-apache-setup
middleware
application
monitoringagent
loggingagent

role: vm-nginx-setup
middle-nginx
application
monitoring
loggingagent

role---playbook---task--action
role
vm-apache-setup
---
main.yml
---
hosts: <servergroup>
become:
roles:
  middleware
  application
  logging
  monitoring

roles/
middleware(default/files/template/tasks/vars/handler)
application(default/files/template/tasks/vars/group_vars)
logging(default/files/template/tasks/vars/group_vars)
monitoring(default/files/template/tasks/vars/group_vars)

role
site.yml(common(all),webserver(web),db(db))
hosts(web,db)

roles
---
common
web---tasks,handlers,template
db

index.html
==========
a=10
$a

playbook
---

files(index.html)

main.yaml
file:
src: index.html
dest: /var/www/html/

templates
---
index.html.j2

{{a}}

main.yaml
template:
src: index.html.j2
dest: /var/www/html

/var/wwwh/html/index.html
10

what is diff betwenn files and templates
files are normal files used to copy to the destination
templates have jinja format and will replace values dynamically
during the playbook execution times

yum
group
user
get_url
command
file

template
service
when
ansible_os_family

roles---site.yaml(selinux,tomcat)

tomcat---tasks(main.yaml)

tasks
-name : install java

- name: creat the group

- name create the user

you can create the roles
ansible-galaxy

ansible
ansible-playbook
ansible-galaxy

validate :
https://github.com/ansible/ansible-examples
lamp-simple
tomcat-standalone

interview questions on ansible
what is inventory
what is static and dynamic

what is cmdb
what is task
what is playbook
what is role
diff between playbook and task
diff between playbook and role
i want to run specifici task/exclude task
i want to exclude speicifi role/exclude role
i want to run a task based on the os
diff between ansible and other tools
how ansible works (agentless/ssh)
can you write the playbooks
tomcat installatin
nginx
apache

Installing the website using the ansible

```
---
- hosts: localhost
  become: yes

  tasks:
  - name: Install apache packages
    yum:
      name: httpd
      state: present

  - name: Ensure the httpd service is running
    service:
      name: httpd
      state: started
    ignore_errors: yes

  - name: remove the deployment files
    file:
      path: /var/www/html/
      state: absent

  - name: Download the website
    get_url:
      url:
https://www.free-css.com/assets/files/free-css-templates/download/page261/reflux.zip
      dest: /tmp/reflux.zip

  - name: Extract zip into /tmp
    unarchive:
      src: /tmp/reflux.zip
      dest: /tmp

  - name: copy the html to the www root
    copy: src=/tmp/templatemo_531_reflux/ dest=/var/www/html/ remote_src=yes

  - name: remove the directory
    file:
      path: /tmp/templatemo_531_reflux/
      state: absent

  - name: remove the zip file
    file:
      path: /tmp/reflux.zip
      state: absent
```

Interview Questions

What is IAM ?
How many ways to interact with AWS ?
How do you manage multiple AWS concepts ?
How to restrict one IAM user to access one region only ?
What is EBS volume ?
What is Scaling /Vertical and HZ ?
How DNS works ? How www.xyz.com works ?
Diff between ami and snapshot ?
How to avoid reboot of instance while taking snapshot ?
What is security group ?
Can i crease machine vertical ?
What is bastion ?
What is user data /init data ?
What is Clb , ALB and NLB ?
Help me with the flow of ALB ?
Diff between ALB and NLB ?
What is Route53 ?
What are the routing policies in Route53 ?
What is CNAME and ALias ?
What is A Record ?
What is jenkins ?
What is sticky session ?
What is package ?
What is repo /artifactory ?
What is maven ?
Diff b/w local and remote repo in maven ?
What are lifecycle phases ?
What is CICD ?
Diff b/w su and su - ?

What are the bashrc and bashprofile ?