

AWS

02-11-21

→ aws overview

sign up for aws

Email -

pass -

con-pass -

aws account name -

what is 2/2 checks passed

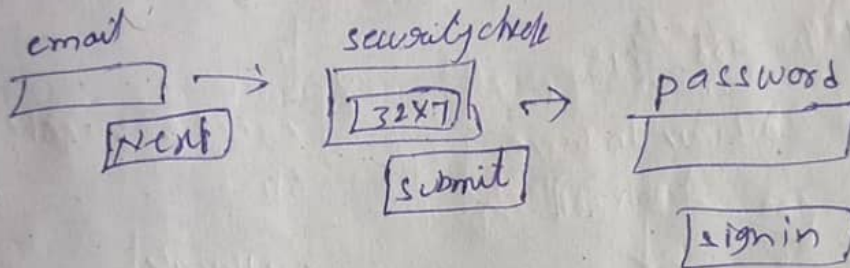
public IP - (will change

automatically when you

stop instance.

sign in

• Root user



EC2 - mean

"Amazon Elastic Compute Cloud"

used to develop and deploy applications faster.

Instance - virtual computing environments, known as instances.

- secure login i/m to your instances using key pairs

(aws stores public key, you store private key in secure place)

- storage volumes for temporary data that's deleted when you stop, or terminate your instance known as instance store volumes.

step 1: AMI - Amazon Machine Image

but contains OS, application servers, & applications

⇒ types of instances

products → Amazon EC2 → Instance TYPE details

↳ general purpose

→ compute optimized

→ memory " "

→ storage " "

static public ip addr

Elastic IP - (3)

N/w and Security - In left side.

↳ Elastic IPs

↳ click

allocate elastic ip address

create allocation

Allocate

cost \$0.00

it generates public ip

we need to add this public to Linux Instance.

actions

Associate elastic ip address

choose instance

Linux

Associate

i-0a110a901129086

check the linux - it shows elastic IP (65.154.90.100)

⇒ when we add elastic IP the public IP will not change any time.

step 1: AMI - Amazon Machine Image

but contain OS, application server, & application

⇒ types of Instance

product's → Amazon EC2 → Instance TYPE details

↳ general purpose

→ compute optimized

→ memory " "

→ storage " "

static public ip addr

Elastic IP - ③

N/w and Security - In left side.

↳ Elastic IPs

↳ click

allocate elastic ip address

create allocation

Allocate

cidr + 8

it generate public ip

we need to add this public to Linux Instance.

action

Associate elastic ip address

choose instance

Linux

Associate

i-0a110a9001129006

check the linux - it show elastic IP (65.154.92.100)

⇒ when we add elastic IP the public IP will not change any time.

Launch template: use Launch template to automate instance launches, simplify permission policy, and enforce best practices across your organization.

Easily update your launch parameters by creating a new launch template version.

Instances → Launch template

Snapshots:

A snapshot takes a copy of the EBS volume and places it in Amazon S3, where it is stored redundantly in multiple availability zones.

The initial snapshot is a full copy of the volume; snapshots store incremental block-level changes only.

This is a fast and reliable way to restore full volume data.

Elastic block store → volumes.

Volume → actions → create snapshot

① — Launch Instance

I. click on Launch Instance

1. select instance

① ex: Amazon Linux2 Am1 64-bit (x86)

2. choose an instance type

②

■ t2 t2.micro

free tier eligible ✓

Next

3. configure instance details // number of instances required

③

Next

4. add storage → Root value at default // minimum

④

Next

5. add tags Next // name to the instance

6. configure security groups // chose security group

Review

Review and Launch

7. Review Instance Launch // check all the details

Launch

Note: key for secure login

choose // create a new key pair

key pair type

① RSA

key pair name :

download key pair

Launch Instance

view Instance

① Connect through AWS CLI

→ select Instance

↳ click on **connect**

ssh client

→ click on **ec2-Instance connect** | **SSH**

Instance ID



public IP address



username

ec2-user

connect

sudo -i

② In git bash

→ ssh client

Instance ID



click on this it will copy

ssh -i "linux.pem" ec2-user @ ec2-65-2-73-202

→ open git bash

↳ paste here. → press enter,

we connect ec2

[ec2-user@ip-172-31-15-170]# sudo -i

→ yum update +Y

→ yum install httpd —①

→ service httpd status —②

→ service httpd start —③

↳ service httpd status

active (running)

Go into Instance.

HTTP enable

② [0.0.0.0:80]

Details | security | Networking | storage

↓ security group's

click

Run Reachability analysis

edit inbound rules ✓

Inbound rules

SSH

custom

HTTP

custom

0.0.0.0/0 ✓

add rule

save rule ✓

→ find instance IP address in google and enter
① ↳ it open page

→ cd /var/www/

→ ls -ltr
 ... html, cgi-bin
 → cd html / → ls

→ vi index.html →

<!!> world </!!>

!wq

Refresh the page

world

① sudo -i
 yum update -y
 yum install httpd
 service httpd status
 service httpd start

(00)

sudo apt update
 sudo apt install apache2
 ip & so

var/www/html/index.html

↓ location of file

AWS Tiers

3-11-21

pricing → AWS Free Tiers.

Volume Types

Volumes

①

- 3 types.

1. SSD (solid state drives)

used for frequent Read/Write operations with small I/O size.

2. HDD (hard disk drives)

used for large streaming workloads → dominant
 → performance as throughput.

3. previous generation - HDD used for workloads with small data sets.

data access & performance is not primary importance

how to see volume

login → EC2 → (EC2) services

↳ EC2

Elastic block store ←

↳ volume.

create volume (EBS)

Elastic block store →

login → EC2 → volumes

↳ create volumes.

(EC2) snapshot's - with

elastic store

1st create snapshot
2nd add snapshot to volume

if volume required snapshot

create snapshot at instance

creation (EC2)

Actions

↳ create snapshot

VT → general

size → 1 GiB

AZ → ap-south-1a

create volume

Attach volumes

create volume

Actions

↳ Attach volume

first select volume and attach

volume -

Instance -

device -

Choose existing snapshot

Attach

Instance volume

EC2 Dashboard → Instance (running) → Instance ID
↳ storage

also we can attach that 2. values to another instance.

VID	DN	VS
+	sw	2.

Instances

based on hours.

Instance Types

(5)

1. on-demand instance — we use instances. but we don't have plan. on long term

2. Reserved instance. — we have a plan. on long term

Instance — Reserved instance → it is very cheap compared to ①.
↳ based on LT.

3. SPOT instance — get very cheaply compared to ① & ②.
↳ all are un-used instances.

Instances → Spot Request (or) spot instance.

above different purchasing options (Instances)

⑥ — Images (Image → AMI)

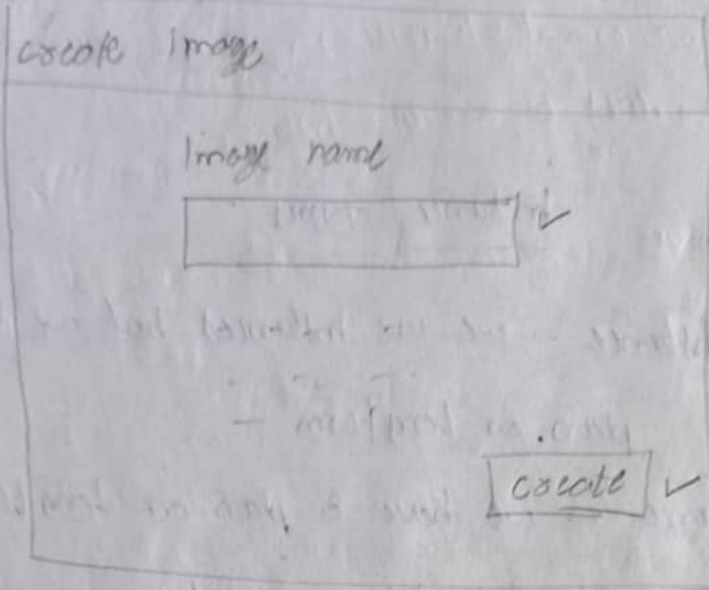
AMI — Amazon machine ~~instance~~ image

Actions Launch instances

↳ Image & templates

↳ create image.

- ① create ami (own ami) → you create ami, first create image.
- ② create ami (own ami) → you create ami, first create image.
- ③ create ami (own ami) → you create ami, first create image.
- ④ create ami (own ami) → you create ami, first create image.
- ⑤ create ami (own ami) → you create ami, first create image.
- ⑥ create ami (own ami) → you create ami, first create image.
- ⑦ create ami (own ami) → you create ami, first create image.
- ⑧ create ami (own ami) → you create ami, first create image.
- ⑨ create ami (own ami) → you create ami, first create image.
- ⑩ create ami (own ami) → you create ami, first create image.



Images → AMIs

↳ it will take some to available

↳ not necessary
next terminate your
Instance in running state

→ we need specific server

↳ it is in our AMI, so launch our AMI

2 ways to launch AMI ① Launch Instance → HX AMIS

② Image

↳ AMI → Launch

Select key pair

linux / RSA

Launch ✓

→ our AMI is running like instance.

↳ connect in ssh

↳ in cli

↳ sudo -i

service httpd status
 " " start } we are not install apache
 " " status } here because it installed
 on terminated instances.

⑦ Load balancing

- type —
1. Application LB
 2. Network LB
 3. Gateway LB
- classic LB.

→ load balancing → Load balancers

→ how much traffic will direct
 " " " " convert

we need 2 instances
 one instance - install
 nginx
 one instance - install
 httpd.

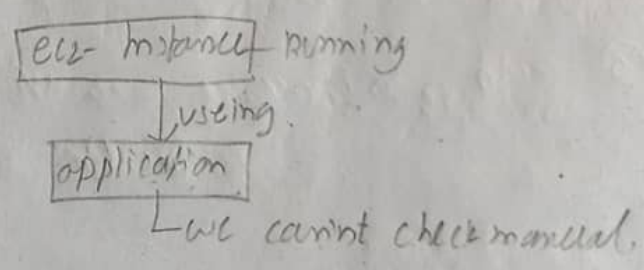
→ it distributes incoming application traffic across

multiple targets such as EC2 instances.

→ this increase availability of your application.

→ it will check health status also (application).

→ show application is
 running or not



Sample Load Balancing

→ Load balancers → at last → classic Load balancer
 ↳ create Load balancer

Define Load balancer

① Load balancer name: ✓
 LB:

enable advanced VPC configuration ☒
 (here select your instance available zone)

② Assign security groups

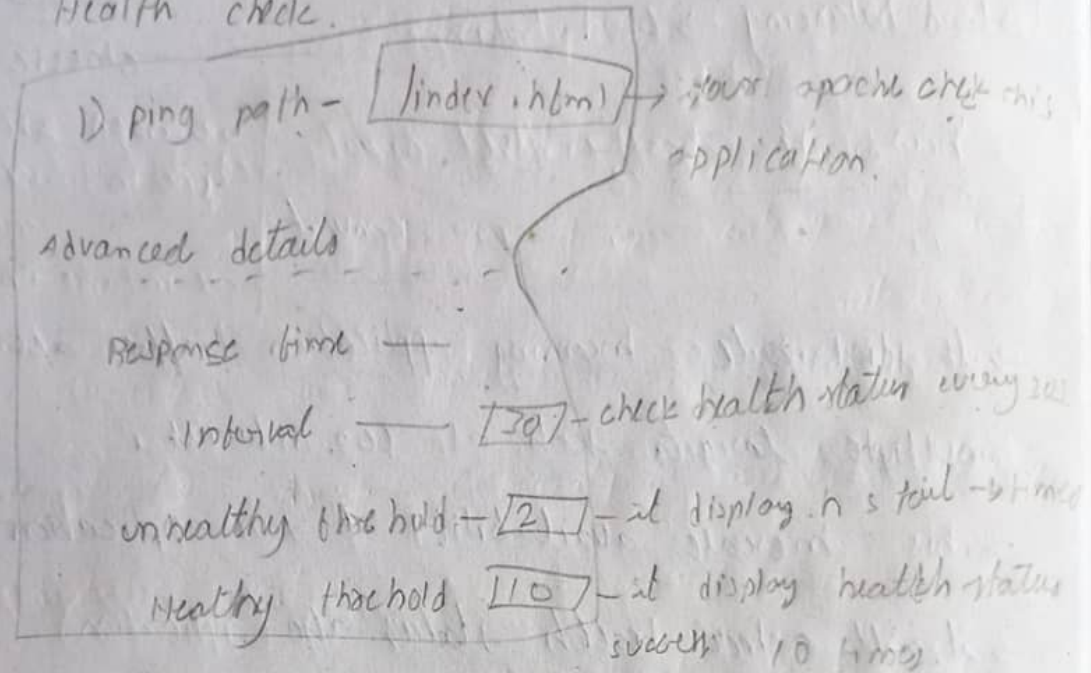
default

Launch wizard-2.

Next ✓

③ Next ✓

④ Health check.



⑤ add instance.

Next ✓

create ✓

Next ✓

Success

checking

Load balancing → load balancer.

--- after 2 mins it will get "in service" ---

When application is in running state.

<u>Inst id</u>	<u>availability zone</u>	<u>status</u>	<u>actions</u>
op-sech-1a		in service	

4-11-21

Add instance to Load balancer

In load balancer → instances (select ☒) → actions
↳ edit instances
↳ select instances

✓

✗

save

check health status

Description | Instances

*DNS name: myname-90869887-11-21-21
op-south-1-elb.amazonaws.com → paste in google.

status: 2 of 2 instances in service.

Instances

added

↓
here are instances
available

status

in service
in service

on instance you do some action

→ sudo

→ service httpd status

→ service httpd start/stop

→ service httpd status

In Running

↳ cd /var/www/html/

↳ vi index.html

→ <h1> server, </h1>

→ paste public ip in google

↳ it open apache (oo) server,

Access application through "load balancer"

Load balancer → description

↳ DNS name - alb-70302614...

↓
copied into google & display

(apache) server and requests

at display 2nd server

⑧ Auto scaling - it do scaling up & down automatically

→ AS monitors your applications and

automatically adjusts capacity to maintain steady;

predictable performance at lowest ^{possible} cost.

→ using AS easy to setup application scaling for

multiple resources across multiple services in minutes.

→ it can increase & decrease the instances

⇒ Auto Scaling

↳ Launch configurations

EX: IRATE application

↳ AS group

create AS group

1. create image (os)
2. Launch configuration
3. create auto scaling

Name: - we can give any name

Launch template:

Switch to Launch config

click on this it will change the below link

① create a Launch template Link

Click on create link. (same as Ami)

2) create
launch
configuration

→ L C Name

→ Ami

→ Instance type

⇒ security groups ☐ select an existing sg.
Launch wizard-2.

→ key pair (login)

key pair options

existing key pair

☒ I acknowledge

3) It was created in → Auto scaling
→ launch configuration.

use Ami code - it in instance launch

3

4) Auto scaling

→ L C

→ AS groups

→ create AS GT

→ set up hoot

Name:

L T

☒ set up

Next ✓

→ C 1 ↓ options

n/w — VPC — (V)

availability zone: 1a1b1c
Next ✓

→ configure advanced options

• • • Next ✓

→ Group size — optional

desired capacity — 2 // launch 2 instances

min capacity — 2

max capacity — 2

→ scaling policy — optional

• None

Next ✓

→ Add notification

Next ✓

→ Add tag

Next ✓

→ Review

Create ASG ✓

so, ASG is created

Note: when you add template in ASG group automatically
EC2 dashboard standard Instance is created.

visible on instance. you delete one instance it automatically create one more instance

add Load balancer to ASG ✓ check in ASG. it

in ASG

→ instance → **edit**

updating for creating

more instance

→ don't want again delete ASG.

→ Load balancing

□ APP, NET

✓ classic load balancing

alb ✓

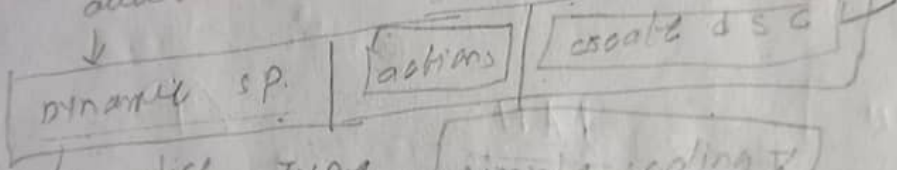
36-20

Dynamic scaling policy (contain cpu time)

update ✓

ASG

→ automatic scaling



→ policy type

simple scaling ✓

S P n

cpu

cpu-instance

cloud watch alarm

cpu-utilization

⇒ InstanceId = i-2oz...

create a cloudwatch alarm

→ we need to create metric

metric

graph: **cpu** metric

next

→ c2

→ proc - instance metrics

→ # server - cpu utilization

select metric

then

50 ✓

Next

Configure actions → notification

→ email

→ display console

Next

→ add name and description next

Alarm name: cpu utilization

→ preview & create

create alarm

→ Take the action

add

1

percentage of group

capacity units

create

+ cpu utilization reach 50%. it take action means increase instances.

Q IAM

IAM polices ⑨

Identify and Access Management (IAM)

→ you use IAM to control who is authenticated (signin) and authorized (has permissions) to use resources.

→ it provide team level permissions.

→ To get access from one service to another service

→ create user's and provide access permissions to that user.

① create user

access management

↳ user groups

↳ users.

↳ add users

add users ✓

① → User name * } multiple users you can add at a time
add another user

↳ select how access type - 2 types

only login access {
- ☒ Access Key - programmatic access
 EX: CLI, SDK, API.
- ☒ password - ius management console access.
 ↳ is a user interface (UI)

we have 3 options → add user to group

② → Add user → set permissions - next ✓ Next permission

③ → Add user → next ✓

④ → Add user Review - create user ✓

⑤ → Success.

Download CSV

	user	Access ID	password	expiry
✓	seri			

→ download csv file and send to that user so he can access aws (it is excel file)

we want to access services from aws console (CLI) we Access key ID, SKI

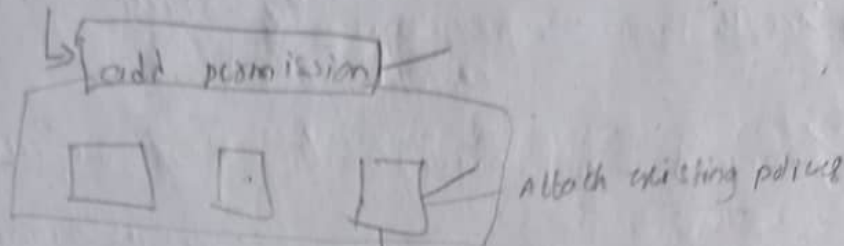
- user name - used for accessing console.
- password -
- Access ID
- Secret Key ID
- Login Link: 06dc267ad20

a) Login user have only access permissions

① Use above link to login. password contain UCIW, etc.
In user: add permissions to user (sri)

→ sri → you need ↓ add/removable MFA (multi-factor authentication)
click → dashboard → add MFA
→ MFA

→ summary



→ ec2 full access

ec2 / 4

Review

add permission

→ check in sri (user) login

→ he can launch instance why because we given admin permission to user!

② Policies and to Remove

→ create policy

→ json

→ Remove load balancer code! in shell.

policy - code
1 - ec2 - full

ec2 full

policy → policy usage

permissions → json → display ec2 permissions code

Name

load balancer
remove

ment ✓

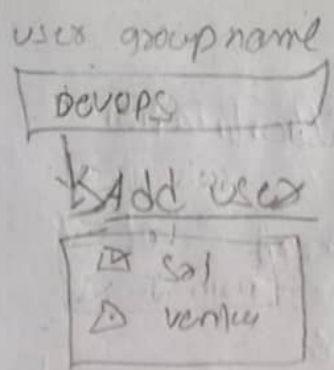
ment ✓

create policy ✓

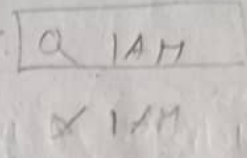
use in create policy

③ users groups

→ create user:



→ Attach Permissions policy



Success

create group

④ Role

→ suppose we give access from ec2 - s3

create role

→ select type of trusted entity

EC2 ✓

next permission

→ Attach PP

Q s3 full access

X s3 fa

next

→ add tags

next

→ Review

Role name

ec2 to s3

check in new instance launch process on

create role

can figure history

IAM role: ☐

automatically access

s3 also.

MFA - Multi-factor authentication

IAM dashboard - not perfect

used for more secure

Add MFA ✓

Activate MFA ✓

↳ virtual MFA device

continue ✓

1st you need to install your mobile authentication

app. Then QR code is available to

can enable it through scan → it enable

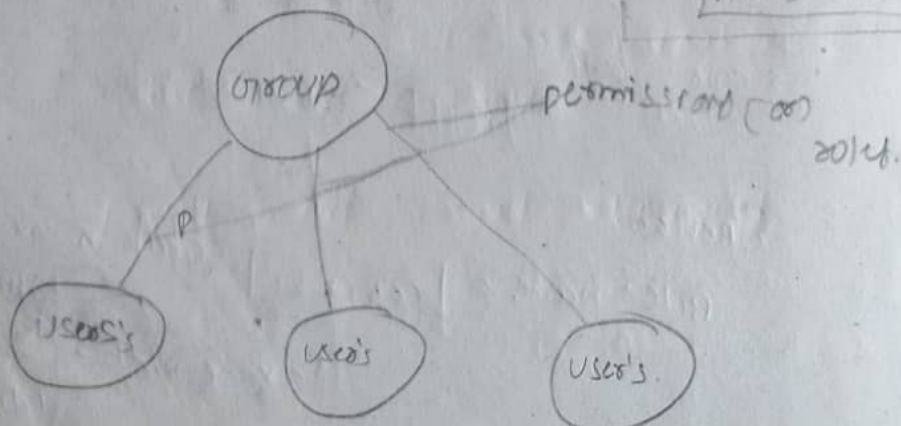
in your mobile → immediately password generate

you enter in QR below MFA code:

MFA code 1:

it generate 2 passwords in 30s of interval time

Assign MFA



5-11-21

10

VPC

(Virtual private cloud)

- using 1 vpc to create multiple ~~vpc~~ EC2 machines.
- used to create own private n/w range

create vpc

1. VPC

Launch vpc wizard

2. CIDR max range in google

→ CIDR conversion table / HPE

we use 1/22 IP

addresses, but get one

IP address per host

1. select a vpc configuration

VPC PBP

2. VPC with a single public subnet.

Select

⇒ IPv4 CIDR block

100.0.0/16

(65531 IP addresses available)

→ VPC IP address based on your requirement.

Ex.

11.11.11.11

1/32

⇒ VPC name: VPC-custom

In the route table

public subnet → have internet gate way

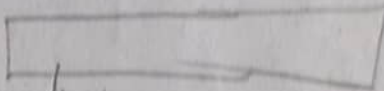
private subnet → have no internet gate way.

VPC → elastic IP creation

→ elastic IP

Allocate elastic IP address

Allocable ✓
successful

=> Elastic IP allocation ID: 
↓
select EIP

create VPC
↓
it takes some time

check in

VPC - Dashboard

-
-

Subnets — contain public private } subnets

Route tables — is related to public subnet. create gateway

| Routes | → Ex! ^{public} igw - o351fb
↓
is a Internet gateway.

| subnet association | → ^{private} ^{not - o3552x1c7b5L (not gateway)} public subnet is added.

private subnet is not associated.

add vpc to instance

25:00

in Launch Instance step ⑤. configure instance.

⇒ ② Subnet:

① Network:

here show your public/private subnets

select vpc

is use — public first

above: public subnet

⇒ ③ Auto-assign public ip:

step ③. configure instance

subnet:

select 2. we — private

above: private subnet

→ connect with in the vpc only.

Diff → use not gateway to connect internet.

→ w/o not gateway it will not connect to internet

connection of private subnet

terminal not connect to public subnet.

→ copy linux .pem file data

→ In terminal vi linux .pem

→ paste code here

→ chmod 400 linux .pem

Now we connect public subnet.

we create private ec2-machine. we cannot connect

Directly to private Machine

we connect through public only.

1. connect through public
after private

Deletion.

① → we can't delete ~~directly~~ "VPC" directly.

1. first delete instances.

② → delete "NAT gateway"

③ → elastic IP's. → dis associate E IP's (or) Re/attach elastic IP address

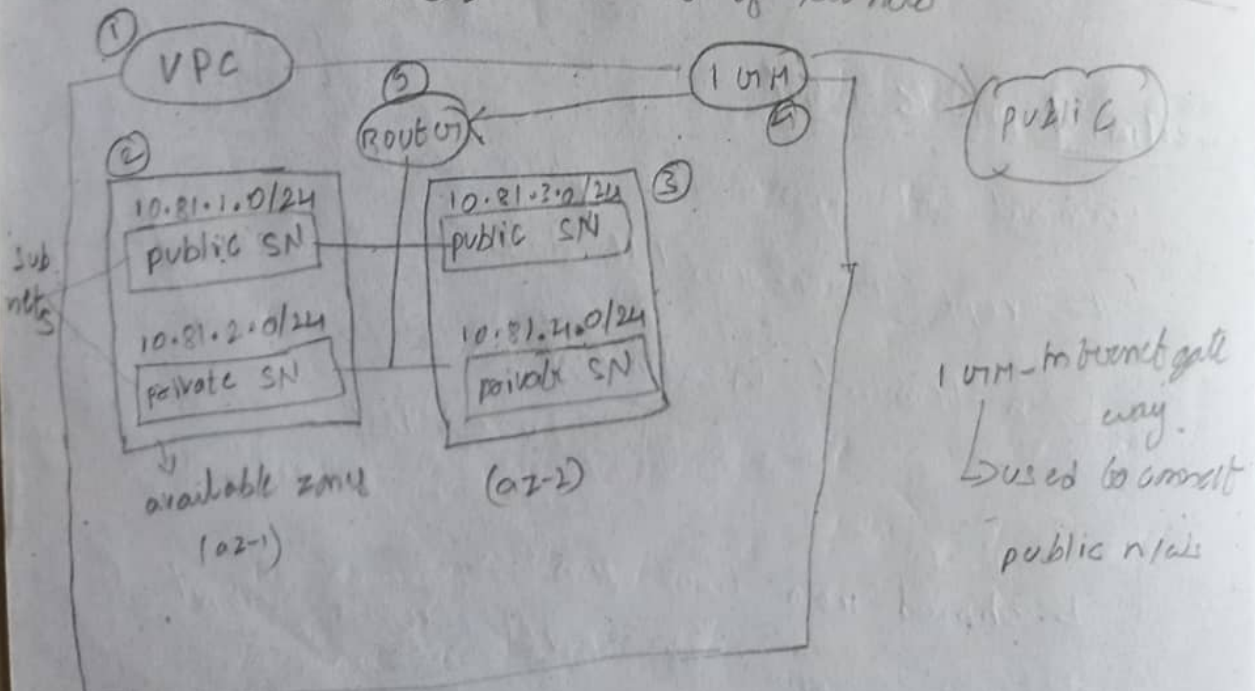
④ - your vpc's → → actions

→ delete VPC

delete. - unknown

delete

10.81.0.0/16 → CIDR in az - create no of subnets



VPC

a VPC

~~your vpc~~
 ↓
 your vpc |
 ↳ create vpc

name tag: vpc-1

IPv4 CIDR block: 10.81.0.0/16

yes create

see on your vpc's

2
2.1 subnets |

↳ create subnets

name tag: public-subnet

vpc: select your vpc-1

availability zone: mumbai-1a

IPv4 CIDR block: 10.81.1.0/24

create

2.2 create subnets

name tag: private-subnet

vpc: your vpc-1

availability zone: mumbai-1a

IPv4 CIDR block: 10.81.2.0/24

create

3.1 create subnets

name tag: public-subnet-2

vpc: your vpc-1

availability zone: mumbai-1b

IPv4 CIDR block: 10.81.3.0/24

create

3.2

create subnet :

name tag : private-subnet-2

vpc : your vpc-1

availability zone : mumbai-1b

IPV4 CIDR block : 10.81.110/24

4

Internet gateway

↳ create Internet gateway

name tag : igw-1

create

it is in detached state

igw-1 → actions

↳ attach to vpc

↳ select your vpc

attach

now it is in attached state

5

Route tables

↳ create Route table

name tag : rt-1

vpc : your-vpc-1

yes, create

from
Link → Route table to VPC

rb-1

Summary | Routes | subnet associations
↓
it shows the subnets

Edit

Destination	target	Status	Permit
10.81.0.0/16	local	active	
0.0.0.0/0	igw-objekt		x

add

add another route

0.0.0.0/0 → used to connect out with world

igw - select (oo) paste your IGW here

click on **save**

default - all subnets are private

we give public access to public subnets

subnets |

public-subnet

→ do same as 2 public subnets

description | flow logs | route Table
↓

Edit route table association

↓
Route Table ID: select your route table
rtb.

↓
it add your public ip block

0.0.0.0/0	igm - obidtib.
-----------	----------------

In server launch.

- 1.
- 2.
3. configure instance.

Network : — select your vpc

subnets : it show 4 subnets. select 1st subnet

Instance:

after launch instance.

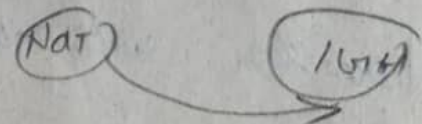
db

Description

private ips : 10.81.3.128

it your 1st subnet ip

NAT - use private subnet to connect outside world



11- S3 Bucket 46:00

simple storage service

→ you use Amazon S3 to store and retrieve any amount of data at any time, from any where

→ Amazon S3 stores ^{large amount of} data of objects within buckets
→ it is global level

create bucket

create bucket

Bucket name:

aws region:

☐ Block all public access.

☐ if enable any one can't access.

☒ disable no one can't access.

☒ I acknowledge that

Bucket versioning.

• disable

② enable → create version with time

Ex: edit and existing file it save with time & version
again edit (add 2 new lines code) it specify to v2.

create bucket

upload - click on bucket name → it open

open the bucket → upload

→ Add files, Add folders
→

permissions

↳ Access control list (ACL)

• current public-read access → any one in the world can access. owner will have read & write

• private → only you can access

properties → 1. order/name

↳ Storage class

upload ✓

→ open upload - file → you can access through URL.

✓ properties / permissions / versions
↳ URL → click → o/p.

Storage class	Designed for	availability zones	duration
• standard		≥ 3	—
• intelligent		≥ 3	—
• standard-IA		≥ 3	30 days
• one zone-IA		≥ 1	30 days

standard-IA — its cost is very high compared to one-zone-IA

default — standard.

o/p: when you edit existing file and upload it it can merely uploaded file, but it will show previous & new o/p.

delete bucket: first delete data in that bucket.


see version

In your bucket, enable (☒ show versions.)

↳ it shows version.

Exist
newly uploaded.

bucket

↳  Image → Actions → we have multiple options

Act → we give permissions to particular user.

Read/write perm on ~~file~~ data

canonical id


Link

public permission

click on Image (or) full name
properties (permissions) version

↳ edit

↳ every one


 I understand

save changes

file size maximum - 5TB

click on Image

↳ open // view your image

object URL  — — — // paste it in google . give pp

VPC

~~search VPC → your VPCs in left side~~
→ ~~create VPC~~

Elastic File System (EFS):

Amazon EFS is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily. It is a web serving & content management.

→ EFS provide a high throughput file system for
↳ is a simple, serverless, set-and-forget, elastic file system. There is no minimum fee or setup charge. You pay only for the storage you use. for read and write access to data stored in infrequent access storage classes.

IPV4 →

we can specify life cycle management
we can move files from one machine to another (or) data → machine → instance