

① Instance creation steps.

1) what is instance.

An instance is virtual server in the AWS cloud. With Amazon EC2 we can setup & configure the operating system & application that run on instance.

2) * Need of instance.

Instance in AWS are basically virtual environments. These virtual environments are isolated from the underlying base OS. i.e. user can rent the virtual environment (instance) on an hourly basis & deploy their applications on it.

3) Uses of instance.

An instance in a database is a combination of the program & memory used to access metadata and application data stored in physical files on a server.

* Step's To Create instance.

1) Login the AWS account.

2) Search EC2 instance on search Bar.

3) click on EC2 → Resources (Running) option with → Launch instance → search Bar will be open.

4) Search Ubuntu → select the ubuntu os with free tier eligible.

5) choose an instance Type.
t2 micro [free tier eligible]

- configure Instance Details

6) configure instance.

keep ~~set~~ all as By default.

7) add storage.

keep ~~set~~ all as By default.

8) add Tags.

click to add a Name tag

create an instance Name.

9) Configure & security group.

change the SSH
~~keep all as By default~~. type as - all traffic
& source as a - Any Any

10) Press Review & Launch.

11) create an existing key pair.

12) ~~enter~~ set a key pair name (only letters)

13) Download the key pair Paste the key on the
Desktop - Launch instance.

NOTE :- If System do not having Git software then
install it.

14) Go To EC2 Dashboard select the instance
& copy the Example (ssh)

- 15) open the Desktop window - Rclick -
Git Bush Here - paste the copied SSH -
enter
- 16) Type \rightarrow yes
- 17) use the command as $\$ \boxed{\text{sudo -i}}$
- 18) getting Rootuser with Ip address.

② LAMP installation.

what is LAMP.

→ LAMP is an open source software stack that provides a framework for creating PHP - Based high performance website & applications with ease.

it includes:- Linux, PHP, Apache, & mariadb or mysql.

Companies use the LAMP stack.

- 1) wordpress 2) Facebook 3) wikipedia 4) Tumblr

Installation of LAMP stack in AWS.

steps.

- 1) follow the step's & Same as a instance it ~~error~~ launching and connection. After getting the root user ip address.
- 2) write a command in same editor as a
 - i) apt-get update.
 - ii) apt-get install tasksel.
 - iii) type yes.
 - iv) tasksel.
- v) select the Jump Server with a space button. press enter & install it.

3) go to EC2 Dashboard - select instance name
- go to Networks - copy the public
IPv4 address - paste in Browser - enter
- it shows the Apache installation.

4) To Test the PHP installation follow the
Below steps.

1) go to New tab or go to search engine
& search steps to install lamp on ubuntu
in that select the 6th step as Test
PHP processing on web server it contains
2 commands.

5) 1st Command :-
This is a file name format.

sudo nano /var/www/html/info.php
↓
command file num with exten
type.

~~After~~ Type the command as it is on
the desktop window it will open new window
i.e. Blank file.

2nd Command :- For PHP code.

<?php
phpinfo();
?>

- copy these command on
- the new window
Press control+x

- press the γ key (for yes) - enter

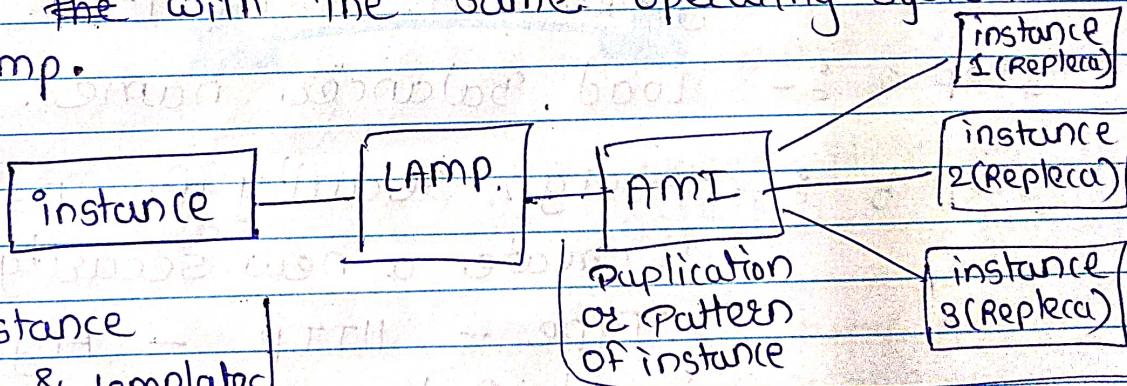
- 6) - Now go to the EC2 Dashboard select the instance & copy the private & ip address.
- Now go to google paste the ip address with `/info.php` - enter.

we will get PHP installation.

(3) 1) AMI Creation step's :-

what is AMI :-

AMI stands for Amazon machine image. which Basically used to increase the instance which is same as the Base instance and contain the with the same operating system with lamp.



steps

select the Base instance

go to the Image & templates

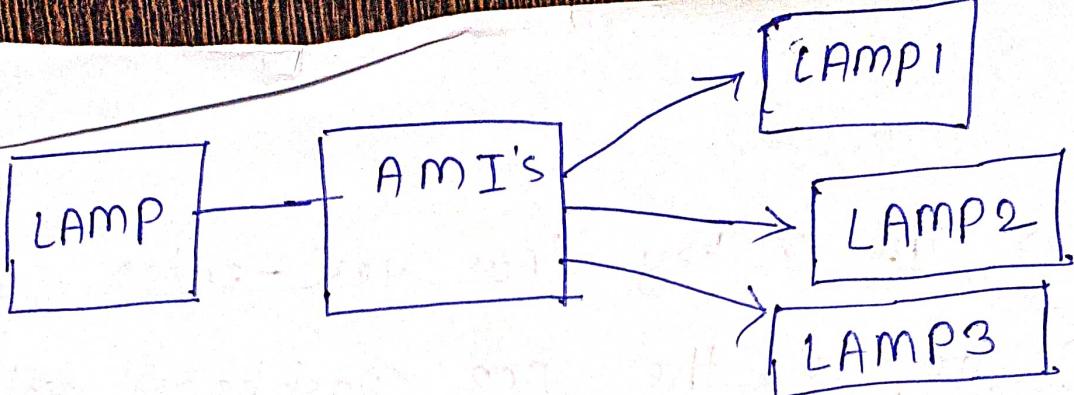
create image

image name

create image

Follow the step's according to launch instance

6) the only difference is that no of launching instance should be according to user.



while creating multiple instance through AMI's there is no connection between the LAMPS & AMI's. To give the set the connection with main LAMP to New creating LAMP we use Load Balancers.

(Q) steps To use Load Balancers

Step 1 :- After installing LAMP & AMI'S go to the Load Balancer's

Step 2 :- Create Load Balancer.

Step 3 :- classic load balancer - Previous generation. - Create

Step 4 :- Load Balancer name.

Step 5 :- Assign Security groups.

i) Create a new security group.

ii) Type - HTTP. - Anywhere.

iii) Add Rule

iv) All traffic - Anywhere.

Step 6 :- Configure Health check.

i) Response Time → set - 2 sec

ii) Interval as a → set - 5 sec

iii) Unhealthy threshold → set - 2 sec

iv) Healthy threshold → set - 5 sec.

Step 7 :- Add EC2 instances.

i) select the instances.

Step 8 :- Add Tags.

i) give key name

ii) value.

Step 9 :- Review

Step 10 :- Create it.

Step 11 :- go to instances

Step 12 :- check the instances are in
Server Inservice.

Step 13 :- go to the Description

Step 14 :- copy the DNS name.

Step 15 :- go to google paste it & search
Apache will be installed.

Step 16 :- on the search bar type info.php
enter we will get the php
installation.

Step 18 :- Refresh it we will get
multiple IP address with a
single link on the same
server.

- 16) Set the alarm name & Alarm description
Next.
- 17) Create alarm.
- 18) Now go to Git Bash.
- 19) Write command `apt-get install stress`
- 20) `stress -c 4`
- 21) Then go to matrix view.
- 22) If the stress line reaches above 50 the instance will be stopped automatically.

(5)

* what is SNS?

1) SNS stands for Simple Notification Services.

This is a messaging service for both application-to-application and application-to-person (A2P) communication.

To do steps to enroll SNS services.

- 1) go Search the SNS in AWS
- 2) click on the simple notification service's (SNS)
- 3) go to the Topics
- 4) create topic
- 5) give the topic name
- 6) rest ↳ keep all as By default.
- 7) Now go to subscription
- 8) create the subscription
- 9) Topic ARN — select
- 10) Select the protocol as per required.
- 11) Create the Subscription — you will get the confirmation to your email or selected protocol.

13) accept it or conform it you will get the subscription.

~~Next~~

Steps to enroll for Alarm or cloud watch

- 1) Select the instance.
- 2) go to Search Box in AWS.
- 2) search cloudwatch click on that
- 3) go to in alarm
- 4) create alarm
- 5) select metric
- 6) Paste the instance id.
- 7) enter
- 8) EC2 > Peer - instance metrics in [click on that] as a cloud watch provides the in parameter to watch as on that. in that we select the CPU utilization
- 9) select metric.
- 10) keep all as By default.
- 11) ~~Peer~~ change the time period as 1 min.
- 12) & Define the threshold value as [50]
- 13) Add notification as — my sns.
- 14) edit the EC2 Action as a stop this instance.
- 15) then go to Next.

Auto Scaling.

(6)

- 1) what is Auto Scaling.

Aws Auto scaling continually monitors your applications to make sure that they are operating at your desired performance levels.

It allows to automatically add or remove EC2 instance according to conditions which user define.

According to user choice we can set all the parameters with automatically same as instance & increases #. we should increase & decrease the instance by with SNS. as incre-

required :- 1) instance with LAMP.

~~step's~~ 2) AMI

3) Load Balance

4) SNS

5) Launch Configurations

6) Auto Scaling.

Step 1 :- Follow the steps from + to instance with LAMP to SNS.

Step 2 :- go to launch configurations create launch configuration given the number.

Step 3 :- select the AMI

Step 4 :- choose the instance type

Step 5 :- create the new security group.

Step 6 :- go to Rules type select All traffic & source type as a Anywhere.

Step 7 :- key pair choose existing key pair

Step 8 :- acknowledge.

Step 9 :- create launch configuration.

Auto Scaling group.

Step 1 :- create Auto scaling group

Step 2 :- give name of Auto scaling.

Step 3 :- switch to launch configuration

Step 4 :- Select the launch configuration name.

Step 5 :- choose instance launch options

i) select VPC which is By default selected.

ii) select subnet as per our choice - Next.

Step 6 :- Configure advanced options.

i) Attach to an existing load balancer

ii) select load balancer name

iii) Health check as 100.

iv)

Step 7 :- Configure group size & scaling. Set all 2 - 2 - 2 - 2

Step 8 :- Add notification's [add sns name]

step 7 :- Add tags which is [optional].
Set as By - default - nxt.

Step 8 :- Review

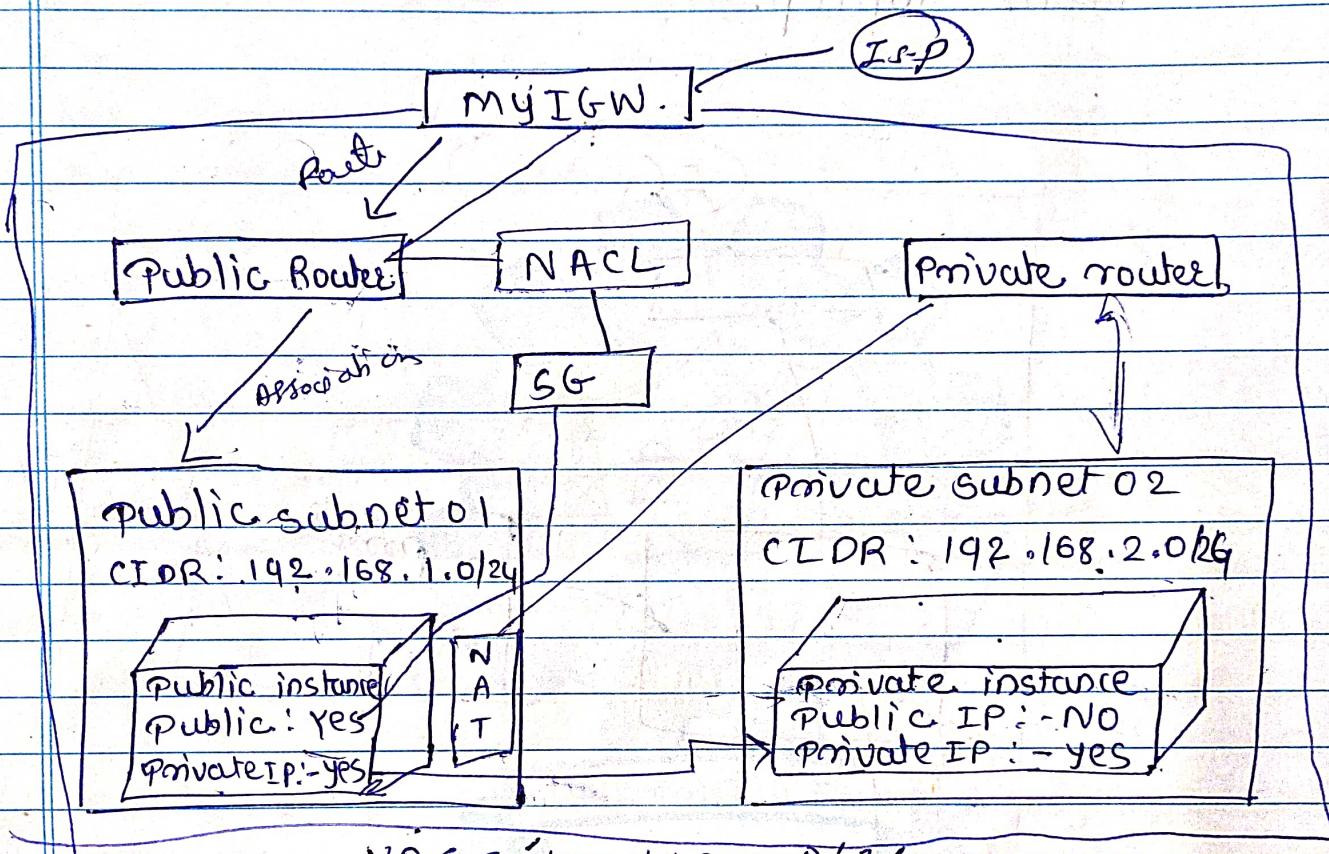
Step 9 :- Create it.

After that number of instance's will be increased & we get notification from mail.

(7)

what is meant By VPC?

→ VPC stands for virtual private cloud (vpc). Basically it has a private cloud hosted within a public cloud.

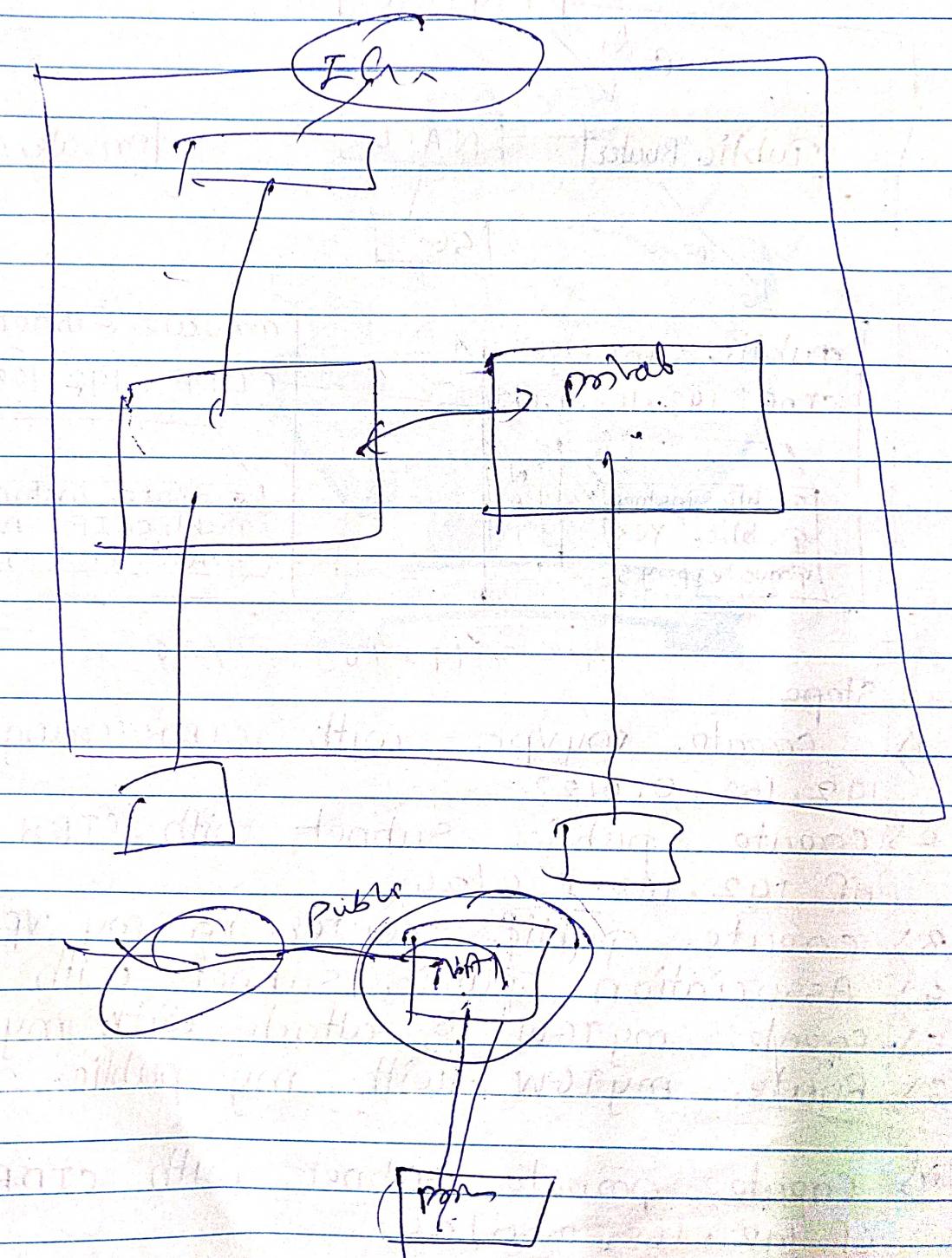


$$VPC = 192.168.0.0/26$$

Steps

- 1) create my vpc with CIDR range of 192.168.0.0/22
- 2) create public subnet with CIDR Range of 192.168.1.0/24
- 3) create public router in my vpc
- 4) Association public subnet with public router
- 5) create my IGW & attach with my vpc
- 6) Route my IGW with my public router
- 7) Create private subnet with CIDR range of 192.168.2.0/24

- 8) create private Router in my VPC.
- 9) Association private subnet with private router.
- 10) create public instance & try to login from laptop.



(7)

VPC steps.

Part (I) VPC creation.

- Search VPC → Create VPC → Set VPC only
- Name → [public VPC] → CIDR Range → 192.168.1.0 /24 → Create VPC.

Part (II) Subnet creation.

- go to subnet → Create subnet → Select the recent created VPC with name → Subnet name → publicsubnet → Set the IP as → 198.168.1.0 /24 → Create subnet. → Attach VPC

Part (III) Create Route Table.

- go to Route table → Create Route Table → give name → public route table. → Select the same VPC → Create the Route table.

Part (IV) Internet gateway.

- go to IGW → Create TGW → give name as → myIGW → Create internet gateway.

Part (V) Association.

- go to Route table → Select the Route table which is created for public
- go to Subnet association → Edit subnet association → Select the public subnet → Save associations.

Part (VI) Route table connection.

- go to Route table → Select the public Route table → Route → Edit routes
- Set Destination → 0.0.0.0/0 → Target as as → Tgw → Save changes.

Part VII → create instance.

go to EC2 → Launch instance → select abu
 → choose instance type → configure instance
 → set n/w (+2.micro) public vpc → subnet as
 as public subnet → public IP → Enable →
 Add storage → click add to add a Name tag
 → public instance → configure security group
 → Add Rule → All traffic → Anywhere →
 Anywhere → Review & Launch the instance

Part VIII → connect instance & check.

select the public instance → Connect →
 copy Example → open git Bash Here →
 paste the Example → yes → if we get
 $192.168.1.216$ 1 → (correct).

Part IX → create private subnet.

go to Subnet → create subnet →
 select the public vpc → give name as
 myprivatesubnet → CIDR Range →
 $192.168.2.0/24$ → create subnet.

Part X → create private Route Table
 go to Route table → select the set the
 name as private Route table → select
 same vpc.

Part XI → make association with pm
 Subnet.

go to private subnet

go to Routetable → select private subnet → Save association. → go to Routes → Edit Routes → Add route → set 0.0.0.0/0 →

Part (XII) → Create NAT

go to NAT → Create NAT → set name as mynat → select the public subnet → Allocate Elastic IP → Create NAT gateway

Part (XIII) → Routing a private subnet with NAT

go to Route table → select private RT → go to Route → Edit Route → Add route → set as 0.0.0.0/0 as Destination → NAT gateway as Target → save changes

Part (14) → create a private instance with (same VPC & private subnet).

To Test the VPC.

① go to the public instance connect with bash paste the example → yes → sudo -i get root ip →

② go to the private instance connect with bash paste the example → yes → we do not get any root ip.

- ③ → go to public git Bash window to run the command. : —

Ping google.com.

ctrl + c = To exit or stop.

Ping Amazon.com.

ctrl + c = To exit or stop.

- ④ Now To access the data from private instance (database) write the Command : —

Ping (private instance)
private ip

Ex:- Ping 192.168.2.244 we are getting data.

- ⑤ To Create file in the private instance (database) write the Command : —

vi (Private key name).pem

Ex:- vi privatekey.pem (New window will be open)

- ⑥ Now open the private key file Copy the key paste it on New window. To exit write command as

:q! []

- ⑦ To check the file mode write command as

ls -la

it shows only read mode.

- ⑧ To change the file mode write command as

chmod 06000 (private key name)

Ex: chmod 06000 privatekey.com.

- ⑨ check again the file mode write command as

ls -la

it shows Both Read & write

- ⑩ Now To access or ~~access~~ feaching the private instance, write the command as.

ssh ubuntu@192.168.2.244 -i privatekey.pem
 (private instance
 private key ip
 add)

- 1) VPC only
- 2) my VPC name.
- 3) CIDR Range - 192.168.0.0/22
- 4) create VPC.
- 5) Create subnet
- 6) my VPC.
- 7) my publicsubnet name
- 8) Range - 192.168.1.0/24
- 9) Route table.
- 10) my publicRT → my VPC create
- 11) Route table → Subnet → SA →
ESA →
S → A
- 12) IGW → my GW → Create.
IGW → select → Action → Attach to VPC.
→ my VPC → A.

Route Table → my -
 Route Table → Route → Select → Route →
 edit Routes → Add route → 0.0.0.0/0 →
 Target (IGW) → save image. (Destination)

EC2 → Launch instance.

config → my VPC → subnet → Enable
my public instance → Add → all traffic → Anywhere.

Create key → download → Launch instance.

⑧ cloud formation.

what is AWS cloudformation?

→ AWS cloudformation is a service that helps you model & set up your AWS resources so that you can spend less time managing those resources & more time focusing on your applications that run in AWS.

In Cloudformation we can create a VPC by single file (JSON) format. this file will be uploaded on stack. so the cloud formation will work automatically. it creates the VPC, public & private Subnet's, Route tables for both, Internet Gateway, NAT. So there is no need to be created all n/w separately.

Steps to do Cloud formation.

- 1) Download the visual studio install the app.
- 2) Create the separate folder on desktop.
- 3) Now go to visual studio create the new file & save it on desktop folder.
- 4) Now 1st take the IP & make in that write the command. as

"AWS::TemplateFormatVersion": "2010-09-09",

"parameters": {

Availability zone (Public, Private)

"Resources": {

VPC { (name)

cidr range: "198.168.0.0/22"

"publicsubnet": {

ref (AZ)

ref (VPC)

"privatesubnet": {

"myIGW": {

ref {

"AttachGateway": {

ref (my VPC)

"publicRT": {

Ref: "myVPC"

"publicRTACS": {

Ref ("publicsubnet")

Ref {"publicRT"}

"Publicroute": {

ref :- PublicRT, myIGW

"privatesubnet": {

ref :- "private AZ" }

ref :- VPCID }

"privateRT": {

ref :- my VPC

```
"privateRTACs": {
    "ref": "privatesubnet"
    "ref": "PrivateRT"
}
```

```
"myNAT": {
    "ref": "publicsubnet"
}
```

```
"myEIP": {
    "Default": {
    }
}
```

```
"RouteNATGateway": {
```

```
    "Ref": "PrivateRT"
    "Desti": "0.0.0.0/0"
    "NatRef": "NAT"
}
```

}

Save the file with Control-C

Now go to cloudFormation create the stack.

- 1) use a sample template
- 2) upload a template file
- 3) choose file (which is in devops folder)
- 4) upload it automatically create the stack it will created subnets, Route tables, assoc, NAT, IGW, etc.

5) check the NPC & \Rightarrow check the IP address.

⑨ RDS Public.

RDS - stands for Relational Database.

Steps

- 1) Create database.
- 2) Select my SQL
- 3) Select the version as our choice
- 4) Select template as free-tier
- 5) Write the name as publicdatabase.
- 6) Set username & password.
- 7) Set the public access as yes.
- 8) Set the Availability zone.
- 9) Create database
- 10) Now install the sqlyog Community app.
- 11) Go to Sqlyog → go to file → new connection
→ set the name → Host address copy
the host address from my stack → click
on the stack it will show's the Endpoint
address copy it & paste on the Host add.
→ Username as a host → password
→ port - 3306. → connect it.
- 12) Connect it → it will open new window
→ click on info.schema → Right click on table

10

RDS - Private.

RDS - Private database creation.

- 1) Create database.
- 2) Set my sql.
- 3) Select the version.
- 4) Select the free tier.
- 5) Write the database name.
- 6) Create username & password.
- 7) Set the public access as no.
- 8) Set the Availability zone.
- 9) Create database.
- 10) Now create the private database instance
(go to EC2 launch private instance).
- 11) After creation of database install MySQL workbench 8.0 CE. After click on that New window will be opened then click on \oplus symbol. it will open new window give the connection name.
- 12) choose the connection method as Standard TCP / IP over SSH.

- 3) SSH Host name :- copy & paste the Public IP address of Private instance. with extension :22
- :22 → is a port of ~~instance~~.SSH
- 4) Set the SSH username as a Ubuntu.
- 5) Select the SSH key file from storage.
- 6) Select MySQL Hostname as a Step ① goto database → click on it that → it will shows the end point address → copy that & paste it as SSH Host name
- 7) MySQL server port: - 3306
- 8) Username.
- 9) Write a same password of database
- 10) Test the connection
- 11) Then click on OK.
- 12) Click on database which will we create on the below ~~the~~ Table option is there Right click on that Table. Create the table.
- 13) Enter the table name — details. & all ~~the~~ Apply it. The database will be created.

RDS - Replica.

For normal RDS push & pop or Read & write operation will be working on single RDS. So the server load will be increased. To avoid that we will creating Replica of RDS ^{read}.

steps for creating Replica.

① step ① :-

After creating private dB go to action's.

- click on Read Replica.
- create a name for new RDS
- create RDS [Read the [it will take 10 to 15 mins]
- Now go to the sql workbench.
- gave th give the connection name
- Select the Connection method as TCP/IP method over SSH.
- SSH - Host name:
copy the private add of same instance which is connected to the main RDS.

- SSH User name -
ubuntu.

- SSH key file :- (selected key file)
- SQL Hostname: copy the new RDS end point
- Username
- Password :-
- Test connection :-
- OK.
- Now after checking the new RDS it will shows the main RDS details.
- when we are deleting ~~new~~^{main} RDS the new RDS will be work's as main RDS.