****

AWS (Amazon Web Services)

****

What is EC2?

EC2

- EC2 stands for Elastic Compute Cloud.
- EC2 is a service in AWS that allows us to create Windows or Linux servers/instances.
- With the help of the EC2 service, we can scale up or scale down instances.
- EC2 is a region-specific service, meaning if we create instances in the Mumbai region, we cannot see those instances in another region, and vice versa.

Steps to create a new instance in EC2

1. Name and Tags

- Tags are essentially labels for the instance.
- We can give any tag/name to the instance.

2. Select AMI

- AMI stands for Amazon Machine Image, which is essentially a predefined template. In this step, you can select the operating system (OS) like Windows, Linux, or macOS.
- - If we want to launch a Linux server, select a Linux AMI.
- - If we want to launch a Windows server, select a Windows AMI.

Examples of AMIs include:

- 1. Red Hat Enterprise Linux
- 2. SUSE Linux Enterprise Server
- 3. Microsoft Windows Server 2019

3. Select Instance Type

- Here, we can select the instance size, including the number of CPUs and the amount of RAM. By default, it is set to 1 CPU and 1 GiB of RAM.

4. Select Key Pair

- A key pair is a file with a .pem extension. It consists of a combination of a public key and a private key. This key pair is used for secure connection to the EC2 instance.
- - To connect to a Windows machine remotely, we need a .pem file.
- - To connect to a Linux machine remotely, we need a .ppk file.

5. Network Settings

- In this step, we can add a security group. A security group acts as a set of firewall rules that control traffic to your instance.
- - By default, port 3389 is open for Windows instances and port 22 is open for Linux instances.

6. Configure Storage

- Here, we can specify the storage required for your instance.
- - By default, Windows instances are allocated 30 GiB and Linux instances are allocated 8 GiB.
- Storage is also referred to as a volume (either HDD or SSD).

7. Configure Instance

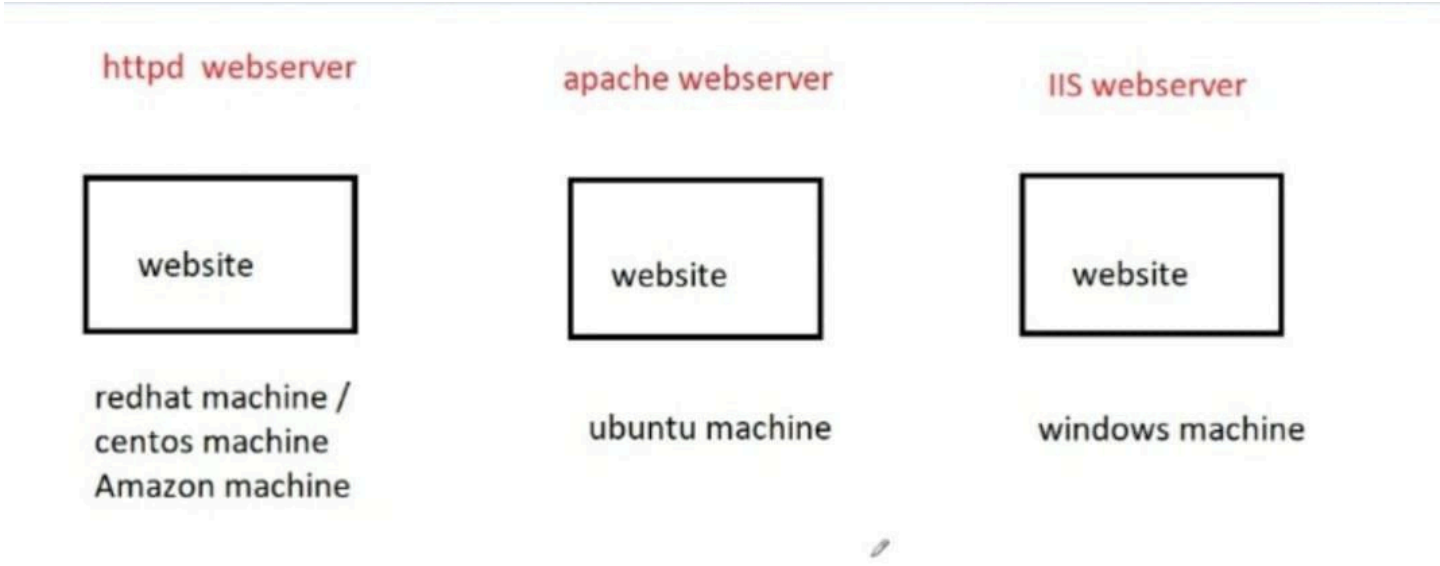
- In this step, we can specify the number of instances you want to create.

Putty

- A tool used to connect to a remote Linux machine.

Remote Desktop Connection

- A tool used to connect to a remote Windows machine.



****

Web Server

****

web server

- A web server is a server where the website is hosted.

How to create an httpd web server
(OR)
Steps to create a website or webpage in Linux

Install the httpd package using the command:

- yum install httpd -y

Start the httpd service using the command:

- systemctl start httpd

Create an index.html file and add content:

- echo "Hello all" > /var/www/html/index.html

Add port 80 in the security group.
To access the webpage or web server, enter the public IP of the machine in a browser:
Public_IP
(OR)
Public_IP:80
(OR)
http://Public_IP

****

check the status of the service

****

To check the status of the service

- systemctl status service_name

To start the service

- systemctl start service_name

To stop the service

- systemctl stop service_name

How to run a web server on a custom port number

- Go to the file /etc/httpd/conf/httpd.conf
- Replace 80 with any custom port number (e.g., 90).
- Restart the httpd service using the following command:
- systemctl restart httpd
- Add port 90 to the security group.
- Access the web page using:
- Public_ip:90

Can we change the instance type of an existing instance?

- Yes, we can change the instance type of an existing instance, but the instance must be in a stopped state.

Can we change the instance type of a running instance?

- No, we cannot change the instance type of a running instance.

How to change the instance type of an instance:

- Select the instance.
- Go to "Actions."
- Click on "Instance Settings."
- Click on "Change Instance Type."
- Select the desired instance
- type and click "Apply."

****

Apache webserver

****

Select an Ubuntu AMI and open ports 22 and 80 in the security group.
Log in as the ubuntu user.

Switch to the root user using the following command:

- sudo su -

Update the repository with:

- apt-get update -y

Install the Apache2 package using:

- apt-get install apache2 -y

Start the Apache2 service with:

- systemctl start apache2

Create an index.html file and add content using the following command:

- cat > /var/www/html/index.html

Enter the public IP address of the instance in a browser.

****

Steps to Create an IIS Web Server (or Host a Website on a Windows Instance)

****

1. Go to the Start menu and type "Server Manager."
2. Click on "Add Roles and Features."
3. In the "Server Roles" section, select "Web Server (IIS)."
4. Click on "Add Features" when prompted.
5. Click "Next," and then click "Install."
6. Wait for the installation to complete and then click "Close."
7. Navigate to the C drive: C:\inetpub\wwwroot
8. Delete all the existing files in the "wwwroot" folder.
9. Right-click and select "New" > "Text Document."
10. Double-click on the file and write some content.
11. Go to "File" > "Save As."
12. Save the file as "index.html" (Make sure to select "All Files" in the "Save as type" dropdown).
13. Save the file.
14. Finally, ensure that port 80 is open in your Security Group (SG) settings.

****

Bootstrap Script

****

Bootstrap Script

- If we need to define commands while creating a server, we can specify them in the bootstrap script.
- All the commands defined in the bootstrap script will run automatically once the server is in a running state.
- In the instance details section, there is an option for "User Data," where we can either write the bootstrap script directly or attach files containing the commands we want to execute when creating the server.

We need to start the bootstrap script by defining the shebang, which is:

- #! /bin/bash
- If we want to perform the same setup on multiple servers, we can also use the bootstrap script for that purpose.
- We simply need to write the bootstrap script and specify the number of instances to create during the "Configure Instance" step.

Example:-

```
#! /bin/bash
yum install httpd -y
systemctl start httpd
chkconfig httpd on
echo "Hello All" > /var/www/html/index.html
useradd nitin
mkdir /tmp/dir1
```

****

chkconfig command

****

chkconfig command

When we stop the server, the httpd service will also stop. However if we start the server again, the httpd service will remain in the stopped state, and we will need to manually start the httpd service.

To ensure that the httpd service starts automatically when the server boots, we can use the chkconfig command to enable it

chkconfig httpd on

This command will configure the httpd service to start automatically during system startup.

****

Elastic IP

****

Elastic IP:-

When we perform a stop and start operation on an EC2 instance, the public IP associated with the instance will change.

To prevent this, and ensure the public IP remains constant even after stopping and starting the instance, we can use an Elastic IP (EIP).

An Elastic IP is a static public IP address that can be associated with an EC2 instance, and it remains the same even after the instance is stopped and started.

Allocate

To request and get an Elastic IP from AWS.

Associate

To attach the Elastic IP to an EC2 instance.

Disassociate

To detach the Elastic IP from an EC2 instance, without releasing it.

Release

To return the Elastic IP to AWS and make it available for others to use.

Is Elastic IP free or paid?

If we allocate an Elastic IP from AWS and do not attach it to a server, we need to pay for it. In this case, we can say that the Elastic IP is paid.

If we allocate an Elastic IP from AWS and attach it to a server, we do not need to pay for it. In this case, we can say that the Elastic IP is free.

Reassociation

Suppose we have attached the Elastic IP (EIP) to instance1 and want to attach the same EIP to instance2.

Using the reassociation feature, we can attach the EIP to instance2 without manually disassociating it from instance1.

The EIP will be automatically disassociated from instance1 during the process.

Remote Desktop Connection

A tool used to connect to a remote Windows machine.

PuTTY

A tool used to connect to a remote Linux machine via SSH.

PuTTYgen

A tool used to convert .pem files to .ppk files and vice versa (i.e., .ppk files to .pem files).

****

Troubleshooting steps if unable to connect to the server:

****

1) Check if port 22 is open in the security group (SG) settings.
2) Verify that the correct username, public IP address, and key pair are being used.
3) Ensure there are no internet connectivity issues.

****

Troubleshooting steps if unable to access a webpage:

****

1) Check if the httpd package is installed by running the command:
   yum list httpd

2) Verify if the httpd service is running by using the command
   systemctl status httpd

3) Check if port 80 is open by running the command
   netstat -tulnp | grep 80

4) Ensure that the security group allows traffic on port 80 from any source.
5) Check if the web server is running on a custom port by inspecting the /etc/httpd/conf/httpd.conf file.
   If a custom port is used, make sure it's allowed in the security group.
6) Verify that the correct public IP address is being used.

****

Region&Zone

****

Region

Region is a physical location around the world where we have cluster the data centers.
There are a total 32 regions & 102 zones are available in aws cloud
Each region consist of multiple zones.

Zone

zone is also called as availability zone(AZ)
each group of logical data centers is called as zone or availability zone(AZ).
zone is one or more discrete data centers with seperate power, networking inside region

****

netstat command

****

netstat command

- To see all open ports in our local machine
  netstat -tulnp

- To see whether a particular port is open or not on the local machine
  netstat -tulnp | grep port_no

  t means TCP
  u means UDP
  l means listening
  n means numeric
  p means program

****

nmap command

****

nmap command is used to see open ports of local & remote machines

To see all the open ports of the local machine

- nmap -Pn private_ip_of_local_machine

To check whether port number 22 is open or not on the local machine

- nmap -Pn -p 22 private_ip_of_local_machine

To check whether port number 80 is open or not on the local machine

- nmap -Pn -p 80 private_ip_of_local_machine

To see all the open ports of the remote machine

- nmap -Pn private_ip_of_remote_machine

To check whether port number 22 is open or not on the remote machine

- nmap -Pn -p 22 private_ip_of_remote_machine

To check whether port number 80 is open or not on the remote machine

- nmap -Pn -p 80 private_ip_of_remote_machine

****

telnet command

****

telnet command is used to see whether a particular port number is open or not on the local & remote machines

To check whether port number 22 is open or not on the local machine

- telnet private_ip_of_local_machine 22

To check whether port number 80 is open or not on the local machine

- telnet private_ip_of_local_machine 80

To check whether port number 22 is open or not on the remote machine

- telnet private_ip_of_remote_machine 22

To check whether port number 80 is open or not on the remote machine

- telnet private_ip_of_remote_machine 80

****

We need to install the nmap and telnet packages.

****

We need to install the nmap and telnet packages.

- yum install -y nmap
- yum install -y telnet

****

Types of instances in EC2

****

Types of instances in EC2

1. General Purpose Instances
In general purpose instances, we get a balance of CPU and memory.
Examples: M and T series.

2. Compute Optimized Instances
In compute optimized instances, we get optimized CPU performance for workloads that require high processing power.
Examples: C series.

3. Memory Optimized Instances
In memory optimized instances, we get optimized memory for workloads that require large amounts of RAM.
Examples: R series.

4. Storage Optimized Instances
In storage optimized instances, we get optimized storage performance for workloads that require high, fast storage throughput.
Examples: I and D series.

5. GPU/Accelerated Computing Instances
In GPU instances, we get enhanced graphics processing for tasks such as machine learning, AI, and 3D rendering.
Examples: G and P series

****

Types of purchasing options in ec2

****

Types of purchasing options in ec2

1. On-Demand Instances

- With on-demand instances, there is no commitment from the user to AWS regarding how long the instance will be used.
- Users pay for compute capacity by the hour or second, depending on the instance type, without long-term commitments.

2. Spot Instances

- In spot instances, there is no guarantee from AWS regarding how long the resources will be available. Spot instances are based on an auction/bidding process, where users bid for unused EC2 capacity.
- The instance is allocated to the user willing to pay the highest price, but AWS can terminate the instance at any time if the spot price exceeds the bid or capacity is no longer available.

3. Reserved Instances

- With reserved instances, there is a commitment between the user and AWS for a specific term (1 or 3 years) to use EC2 instances. In exchange for this commitment, users receive a discounted rate compared to on-demand pricing. AWS guarantees the instance capacity for the reserved term.

****

Types of AMIs (Amazon Machine Images)

****

Types of AMIs (Amazon Machine Images)

There are several types of AMIs (Amazon Machine Images) available for use in AWS:

1. Quickstart AMIs

- Quickstart AMIs are free-tier eligible, meaning they are available at no cost for users within the Free Tier usage limits.
- These AMIs are intended for practice purposes and experimentation.
- They are not verified or trusted by AWS, so users should exercise caution when using them in production environments.

2. AWS Marketplace AMIs

- AWS Marketplace AMIs are trusted and verified by AWS.
- These AMIs are listed in the AWS Marketplace, and AWS assigns ratings to them based on user feedback, which can help users determine their reliability.
- These AMIs are generally intended for production environments and come with support and maintenance options.

3. My AMIs

- In "My AMIs," there are two types:
- Owned by me: These are AMIs that the user has created.
- Shared with me: These are AMIs that other AWS users have shared with you.

4. Community AMIs

- When a user creates a private AMI and then chooses to make it public, it becomes a Community AMI.
- Community AMIs are shared with the broader AWS community, and they are not verified or trusted by AWS.
- Because they are public, they may contain configurations and settings that might not be secure, so caution should be taken when using them.

To create an image from an instance:

1. Select the instance.
2. Click on Actions > Create Image.
3. Provide the necessary details and click Create Image.

To launch an instance from the image (in the same region):

1. Select the image.
2. Click on Launch Instance.

To share an image from one region to another region:

1. Select the image.
2. Click on Actions > Copy AMI
3. Select the destination region and click Copy

To share an image from one account to another account:

1. Select the image.
2. Click on Actions> Modify Image Permissions
3. Enter the target account ID and click Save

To delete an image:

1. Select the image.
2. Click on Actions > Deregister

****

Load Balancer (LB)

****

Load Balancer (LB)

- A Load Balancer is a device or service in AWS used to distribute incoming traffic (load) evenly across multiple instances to prevent overloading any single instance.
- The load balancer helps prevent instances from being overwhelmed by high traffic and ensures high availability and fault tolerance.

Practical Steps

1. Create a Security Group (SG) with ports 22 & 80

- Open the AWS console and navigate to the EC2 service.
- Create a new Security Group that allows inbound traffic on port 22 (for SSH) and port 80 (for HTTP).

2. Create 2 Web Servers with Different Data in the index.html File

- Launch two EC2 instances with a web server
- Modify the index.html file on each server to display different content
- (e.g., "Web Server 1" in the first instance, and "Web Server 2" on the second instance).
- Attach the previously created Security Group (SG) to both instances.

3. Create a Target Group and Add Targets

- In the AWS console, navigate to the "Target Groups" section under EC2.
- Create a new target group and add both web server instances as targets.

4. Create a Load Balancer

- Create a new Application Load Balancer (ALB) or Network Load Balancer (NLB) as needed.
- Configure the load balancer to use the target group you created earlier.

5. Enter the DNS Name of the Load Balancer in a Browser

- After the load balancer is created, find its DNS name in the AWS console.
- Enter the DNS name in our browser to verify that the traffic is being distributed between the two web servers.

****

Load balancer ?

****

1. What is a load balancer?

2. Why do we use a load balancer?

3. What are the types of load balancers?

4. What is the difference between an Application Load Balancer and a Network Load Balancer?

5. What is the difference between TCP and UDP?

6. What are the strategies used in load balancing?

7. What are the components of a load balancer?

****

Example LB

****

```
#! /bin/bash

yum install -y httpd
systemctl start httpd
chkconfig httpd on

echo "welcome to ITpreneur institute 1" > /var/www/html/index.html

mkdir /var/www/html/linux
echo "welcome to linux batch 1" > /var/www/html/linux/index.html

mkdir /var/www/html/aws
echo "welcome to aws batch 1" > /var/www/html/aws/index.html
```

#! /bin/bash

```
yum install -y httpd
systemctl start httpd
chkconfig httpd on

echo "welcome to ITpreneur institute 2" > /var/www/html/index.html

mkdir /var/www/html/linux
echo "welcome to linux batch 2" > /var/www/html/linux/index.html

mkdir /var/www/html/aws
echo "welcome to aws batch 2" > /var/www/html/aws/index.html
```

****

Types of Load Balancers (LB)

****

Types of Load Balancers (LB)

1. Application Load Balancer (ALB)
2. Network Load Balancer (NLB)
3. Classic Load Balancer (CLB)
4. Gateway Load Balancer (GWLB)

Difference between Application LB & Network LB

Application Load Balancer (ALB)

- Operates at Layer 7 (Application Layer) of the OSI model.
- Supports HTTP and HTTPS traffic (ports 80 and 443).
- Introduces more latency compared to NLB
- Handles fewer requests compared to NLB.

Network Load Balancer (NLB)

- Operates at Layer 4 (Transport Layer) of the OSI model.
- Supports TCP and UDP traffic.
- Introduces less latency as compared to ALB
- NLB can handle millions of requests per second

Classic Load Balancer (CLB)

- Classic load balancer is a combination of both Application Load Balancer (Layer 7) and Network Load Balancer (Layer 4).
- Supports both HTTP/HTTPS (Layer 7) and TCP/UDP (Layer 4) traffic.
- Classic Load Balancer was more common before ALB and NLB gained popularity.

Difference between TCP & UDP

TCP (Transmission Control Protocol)

- Reliable communication protocol with no data loss.
- Slower compared to UDP
- Example: Email

UDP (User Datagram Protocol)

- Unreliable communication protocol with possible data loss.
- Faster than TCP
- Example: Voice calls or video streaming

****

OSI Model

****

OSI Model

- The Open Systems Interconnection (OSI) model defines a framework for network communication. It consists of 7 layers:

Application Layer (Layer 7)

- The topmost layer responsible for network services like HTTP, FTP, and email.

Presentation Layer (Layer 6)

- Manages data encoding, encryption, and compression.

Session Layer (Layer 5)

- Handles sessions or connections between applications.

Transport Layer (Layer 4)

- Ensures end-to-end communication with protocols like TCP and UDP.

Network Layer (Layer 3)

- Route packets using IP addresses; Routers operate at this layer.

Data Link Layer (Layer 2)

- Responsible for data frames, MAC addresses, and physical addressing; Switches operate at this layer.

Physical Layer (Layer 1)

- Deals with the physical connection, including cables, switches, and electrical signals.

****

Different Strategies Used in Load Balancing

****

Different Strategies Used in Load Balancin

1. Round Robin Strategy

- Round Robin Strategy distributes incoming traffic evenly across all available servers in a sequential manner.
- Suppose there are two servers: server1 & Server2
- The load balancer sends the 1st request to Server1, the 2nd request to Server2
- then the 3rd request goes back to Server1 & 4th request to server2

2. Least Outstanding Request Strategy

- In the Least Outstanding Request Strategy, the load balancer checks which instance has the least load (fewest outstanding requests) and sends more requests to that instance.
- Suppose there are 2 servers: Server1 and Server2.
- Server 1 is already handling a higher load, and we receive 100 new requests.
- The load balancer will send more requests (e.g., 70 requests) to Server2, which has a lighter load, and the remaining 30 requests to Server1.

****

Components of Load Balancing

****

1. Target

- Each instance registered to the load balancer is called a target.

2. Target Group

- A target group is a collection of targets (instances) that the load balancer routes traffic to.

3. Listener

- A listener is a process that checks for incoming connection requests on a specific port (or protocol) and forwards them to the appropriate target group.

4. Health Check

- The load balancer performs health checks to determine whether the targets (instances) are healthy and able to handle traffic.

- lb sends ping request continuously to monitor the health of instances
- and after that if instances are healthy (in running state) then its duty of instances to send 200 ok respose to lb

****

Parameters in health check of instances

****

1. Timeout

- The time the load balancer waits for a reply from an instance after sending a ping request.
- If the instance does not respond with a "200 OK" within this time, it is considered a failure.

2. Time Interval

- The time between consecutive ping requests sent by the load balancer to the instances is called the time interval.

3. Healthy Threshold

- If the load balancer receives a "200 OK" response within the timeout, the instance is added to the healthy list.
- The number of continuous successful "200 OK" responses required for the load balancer to declare an instance healthy is called the Healthy Threshold.

4. Unhealthy Threshold

- If the load balancer does not receive a "200 OK" response within the timeout, it adds the instance to the unhealthy list.
- The number of continuous failed "200 OK" responses after which the load balancer declares an instance unhealthy is called the Unhealthy Threshold.

****

Auto scaling

****

Scaling

- Scaling is the process of increasing or decreasing the number of instances manually as per the requirements.

Auto scaling

- Auto scaling is the process of increasing or decreasing the number of instances automatically based on the requirements.

AutoScaling Practical steps

Step 1

- Create an SG (Security Group) with port 22 and 80.
- Create an empty TG (Target Group).
- Create an LB (Load Balancer) and attach the empty TG.

Step 2

- Create a launch template where we can define the OS, CPU, RAM, storage, key pair, SG, and bootstrap script for instances, which will be created automatically due to high load on the load balancer.

Step 3

- Create an Auto Scaling Group (ASG) where we can define the minimum, maximum, and desired capacity of instances.

- The minimum value defines how many instances should be running during low load on the load balancer.
- The maximum value defines how many instances should be running during high load on the load balancer.
- The desired value is a value between the minimum and maximum values.

step 4

- create topic in SNS to send notification to users

step 5

- Go to the cloudwatch and create 2 alarms
- 1st alarm for cpu > 80%
- 2nd alarm for cpu < 20%

step 6

- **Auto Scaling Groups ->**
- create 2 policies
- 1st policy is instance creation policy & attach alarm cpu > 80%
- 2nd policy instance removal policy & attach alarm cpu < 20%

****

SNS

****

SNS

- sns stands for simple notification service
- sns is region specific service
- with the help of sns service we get the notification of any activity which is going in Aws

- In sns we create a topic
- In topic we create subscriptions
- And in subscription we add an endpoint that is the mail id of the user to whom we want to send notification of activity.

****

To increase load on instance

****

To increase load on instance use below commands

```
amazon-linux-extras install epel -y
yum install stress -y
stress -c 90
```

deletetion steps

1. delete autoscaling
2. delete Launch Template
3. delete LB,TG,instances
4. delete topic & subscriptions
5. delete alarms

****

Components of auto scaling

****

Components of auto scaling

1. target
2. target group
3. load balancer
4. Launch Template
5. ASG
6. SNS
7. alarms
8. Policies

1. Target

- Each instance registered to the load balancer is called a target.

2. Target Group

- A target group is a collection of targets (instances) that the load balancer routes traffic to.

3. Listener

- A listener is a process that checks for incoming connection requests on a specific port (or protocol) and forwards them to the appropriate target group.

4. Health Check

- The load balancer performs health checks to determine whether the targets (instances) are healthy and able to handle traffic.

- lb sends ping request continuously to monitor the health of instances
- and after that if instances are healthy (in running state) then its duty of instances to send 200 ok respose to lb

5.step

create topic in SNS to send notification to users

6.step

Go to the cloudwatch and create 2 alarms
1st alarm for cpu > 80%
2nd alarm for cpu < 20%

7.step

create 2 policies
1st policy is instance creation policy & attach alarm cpu > 80%
2nd policy instance removal policy & attach alarm cpu < 20%

****

Scaling

****

Scaling

- Scaling is the process of increasing or decreasing the number of servers based on requirements.

There are two types of scaling:

1. Vertical Scaling

- Vertical scaling involves increasing or decreasing resources such as CPU, memory, or storage within an existing machine. It has two subtypes:

Scale Up

- This refers to increasing resources like CPU, memory, or storage in an existing machine.

Scale Down

- This refers to decreasing resources like CPU, memory, or storage in an existing machine.

2. Horizontal Scaling

- Horizontal scaling involves increasing or decreasing resources by adding or removing machines. It also has two subtypes:

Scale In

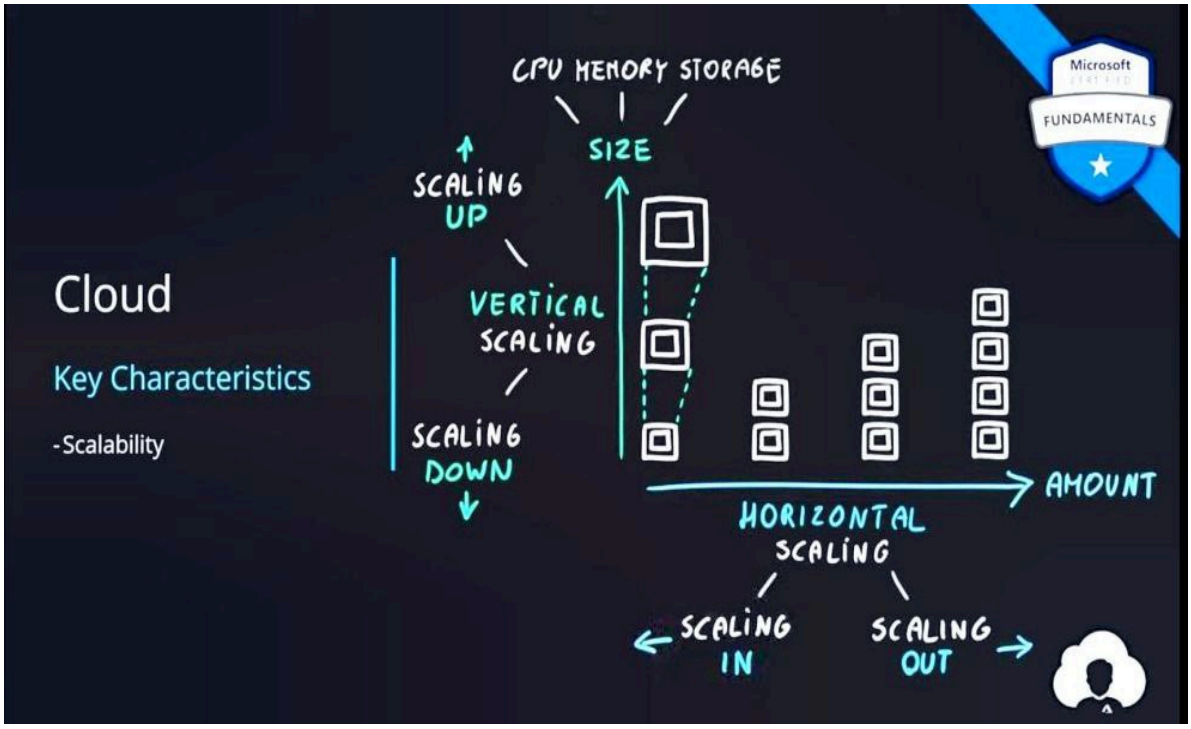
- This refers to decreasing resources by reducing the number of machines.

Scale Out

- This refers to increasing resources by adding more machines.

Auto Scaling

- Auto scaling is the process of automatically adjusting the number of servers based on the current requirements.



EBS service(EBS stands for Elastic Block Store)

- EBS stands for Elastic Block Store.

- The purpose of the EBS service is to increase the storage of both Linux and Windows instances by creating additional volumes.

- EBS supports both Linux and Windows OS, meaning we can increase the storage of both Linux and Windows instances.

- EBS is a zone-specific service, meaning we create an additional volume in a particular zone.
- We can only attach an extra volume to an instance that is in the same zone.
- If the instance and volume are in different zones, we cannot attach the volume to the instance.

- EBS stores persistent data, meaning the data is retained even if the EC2 instance to which the volume is attached is stopped, restarted, or terminated.

```
delete on termination --> yes

instance --> terminate

default volume --> delete --> yes
extra volume --> delete --> no

=====================================

delete on termination --> no

instance --> terminate

default volume --> delete --> no
extra volume --> delete -->    no
```

1st instance
To see the number of partitions or total storage

- df -h
- (or)
- lsblk
- (or)
- fdisk -l

Give the file system to the partition using the command

- mkfs -t ext4 /dev/xvdb
- Mount the partition to a directory
- mkdir /lekharaj
- mount /dev/xvdb /lekharaj
- cd /lekharaj
- Create some files:
- touch file1 file2 file3
- ls
- cd /
- Unmount the partition
- umount /lekharaj

Detach this volume from the first instance and attach it to the second instance (both the volume and the instance should be in the same zone as EBS volumes are zone-specific).

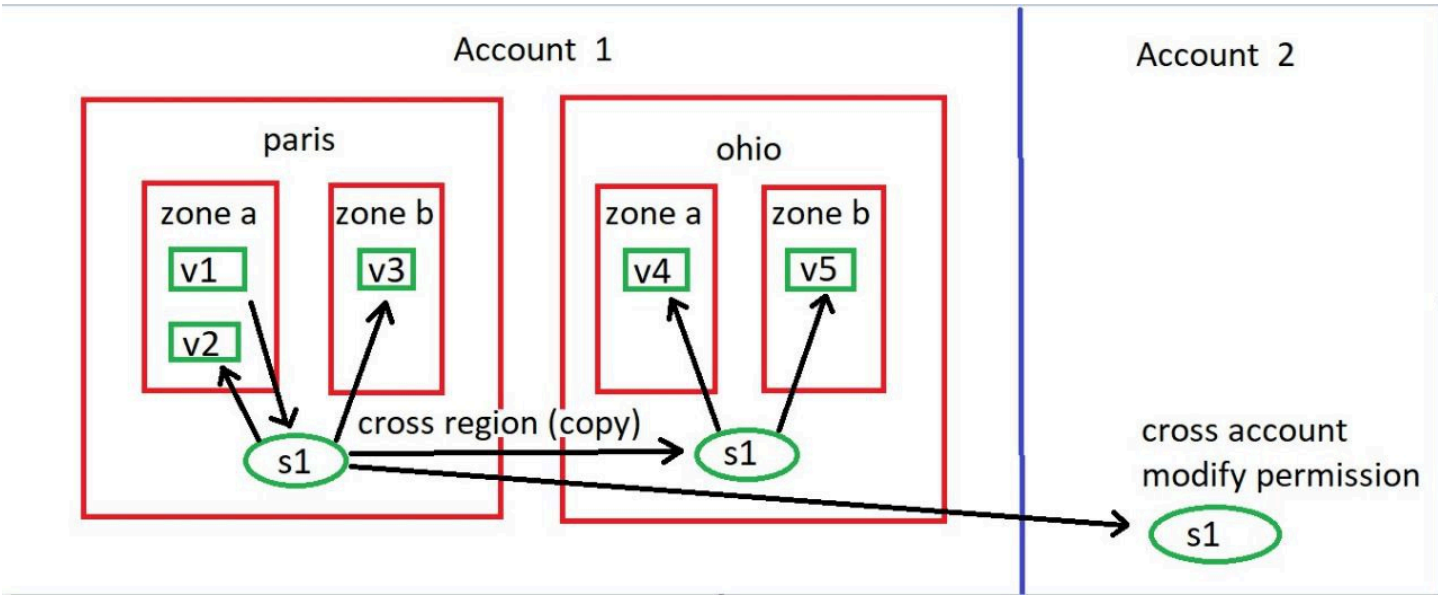
- Connect to the second instance:
- mkdir /sumit
- mount /dev/xvdb /sumit
- cd /sumit
- ls
- Unmount the partition:
- umount /sumit
- Detach the volume.
- Delete the volume.
- Delete the instances.

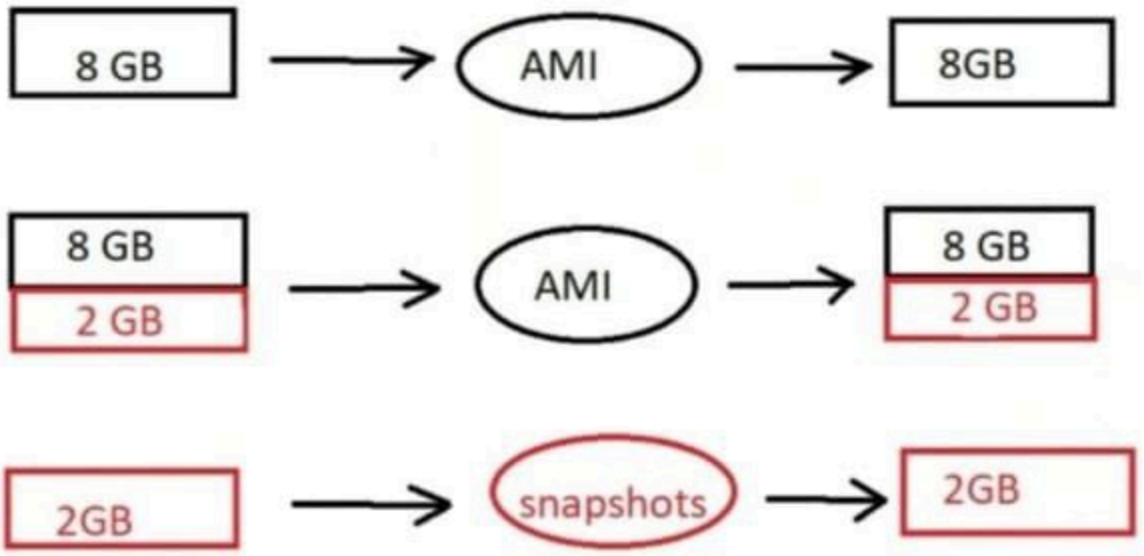
```
instance --> backup ---> AMI

extra volume --> backup --> snapshots

AMI --> share --> cross region --> copy AMI
snapshot --> share --> cross region --> copy snapshot

AMI --> share --> cross account -->  edit AMI permissions
snapshot --> share --> cross account --> modify permissions
```





****

Autoscaling???

****

- 1. What is autoscaling, and what is its purpose?
- 2. What are the components of autoscaling?
- 3. What are the types of scaling?
- 4. What is horizontal scaling, and what are the types of horizontal scaling?
- 5. What is vertical scaling, and what are the types of vertical scaling?
- 6. In autoscaling, do we use vertical autoscaling or horizontal autoscaling?
- 7. In day-to-day life, do we use vertical scaling or horizontal scaling?

****

EBS(Elastic Block Store.)

****

EBS Service

- EBS stands for Elastic Block Store.
- EBS is a zone-specific service.
- EBS stores data as blocks.
- EBS is used to add extra storage volume to existing instances.
- EBS stores persistent data.
- In EBS, we cannot attach the same volume to multiple instances at the same time.
- In EBS, large data is divided into small blocks, also called chunks.
- Each block is assigned a unique identifier number and then stored.
- This is done using a technology called SAN (Storage Area Network).

Types of Volumes:

1. General Purpose SSD
2. Provisioned IOPS
3. Throughput Optimized HDD
4. Cold HDD
5. Magnetic

IOPS

- IOPS stands for Input/Output Operations Per Second.
- In Provisioned IOPS, the input/output transaction speed is high.

Throughput

- Throughput refers to the number of bytes written per second.

Advantages of EBS

1) With EBS, we can increase the storage capacity of instances.
2) EBS supports both Linux and Windows operating systems.
3) EBS stores persistent data.
4) We can create additional volumes up to 16TB.
5) Since we cannot decrease the volume size, there is no risk of data loss.
6) In EBS, we can create snapshots of volumes, which can be shared across different regions and accounts. This allows us to take backups by creating snapshots.
7) EBS provides bootable drives, which contain the boot files necessary for the system startup.

Disadvantages of EBS

1) We cannot attach the same volume to multiple instances at the same time.
2) We cannot attach a volume to an instance that is in a different availability zone.
3) We cannot decrease the volume size.
4) We cannot create volumes smaller than 1 GB.

****

EFS (Elastic File System)

****

EFS

- EFS stands for Elastic File System.
- EFS is a region-specific service.
- EFS stores data as files.
- EFS is used to sync data between two or more instances.
- EFS supports only Linux operating systems.
- EFS does not provide bootable drives.
- EFS works on the NFS (Network File System) protocol, with port number 2049.
- The purpose of EFS is to sync directories that are on the same or different instances.

Difference Between EBS & EFS

- 1. EBS stands for Elastic Block Storage, and EFS stands for Elastic File System.
- 2. EBS is a zone-specific service, while EFS is a region-specific service.
- 3. EBS stores data as blocks, while EFS stores data as files.
- 4. EBS supports both Linux and Windows operating systems, whereas EFS supports only Linux operating systems.
- 5. EBS provides bootable drives, while EFS does not provide bootable drives.
- 6. In the case of EBS, no specific port is required, but in the case of EFS, port number 2049 (NFS) should be open.
- 7. The purpose of EBS is to create additional volumes, while the purpose of EFS is to sync directories that are on the same or different instances.

****

S3 (Simple Storage Service.)

****

S3 Service

- S3 stands for Simple Storage Service.
- S3 is a global service.
- S3 stores data as objects.
- S3 supports both Linux and Windows operating systems.
- S3 does not provide bootable drives.

In S3, there is a concept of a bucket, inside which we upload objects from either a Windows or Linux server.

- The bucket is region-specific.
- The name of a bucket is globally unique.
- The capacity of each bucket is 5 TB.
- We can create up to 100 buckets per account.
- Buckets can be private or public, and the objects inside buckets can also be private or public.
- When we create a bucket it is private by default.

(To make it publicj we can use ACL option Similarly, when we upload an object, it is private by default, and we can choose to make it public.)

****

Difference between EBS, EFS, and S3

****

1) EBS stands for Elastic Block Store,
EFS stands for Elastic File System, and
S3 stands for Simple Storage Service.

2) EBS is a zone-specific service,
EFS is a region-specific service, and
S3 is a global service.

3) EBS stores data as blocks, EFS stores data as files, and S3 stores data as objects.

4) EBS supports both Linux and Windows operating systems,
EFS supports only Linux operating systems, and
S3 supports both Linux and Windows operating systems.

5) EBS provides bootable drives,
EFS does not provide bootable drives, and
S3 does not provide bootable drives.

6) For EBS and S3, no port number is required, but for EFS, the NFS port (port 2049) should be open.

****

Versioning in S3

****

Versioning in S3

- Versioning is used to keep different versions of objects.
- When versioning is enabled, we can see both the current and non-current versions of objects.
- If we delete the current version of an object, a delete marker will be created.
- After deleting the delete marker, we can restore the current version of the object.
- If we want to permanently delete the current version, we need to enable the "Show Versions" option.
- Once the current version is deleted, the previous non-current version will become the current version.

****

Cross region replication of buckets

****

Cross region replication of buckets

- We use cross-region replication of buckets to back up objects and reduce latency (delay time).
- In cross-region replication, versioning must be enabled on the source bucket.
- There are two concepts in cross-region replication: the source bucket and the destination bucket.
- Replication can be configured within the same region, across regions, or across accounts.
- When an object is added to the source bucket, the same object appears in the destination bucket.
- If an object is deleted from the source bucket, it will still remain in the destination bucket.
- Any additions or deletions made in the destination bucket will have no effect on the source bucket.
- Replication can be set up from both directions.

****

Static Website Hosting in S3

****

Static Website Hosting in S3

- We can host a static website on Amazon S3 by enabling the Static Website Hosting option in the properties tab of the S3 bucket.
- Once enabled, we will receive a URL through which you can access the website hosted in S3.
- In S3, only static websites can be hosted. Dynamic websites cannot be hosted in S3.

Static Website

- A static website is a website that consists of fixed content, meaning the data does not change based on user interaction or other factors.

Dynamic Website

- A dynamic website is a website where the content is generated or updated in real time, often based on user interaction or other changing data sources (e.g., databases, APIs).

****

Ways to Access/Use AWS

****

Ways to Access/Use AWS

1. AWS Management Console

- The web-based user interface for managing AWS services.

2. AWS CLI (Command Line Interface)

- A command-line tool that allows you to interact with AWS services through scripts or commands.

3. AWS SDK (Software Development Kit)

- A set of programming tools and libraries to integrate AWS services into applications in various programming languages.

4.AWS API (Application Programming Interface)

- A set of programmatic interfaces that allows direct communication with AWS services via HTTP requests.

When we use the aws configure command, we are prompted to provide the following details:

1. Access Key ID

- The AWS access key ID associated with your AWS account or IAM user.

2. Secret Access Key

- The secret access key corresponding to the provided access key ID.

3. Region

- The AWS region where you want to run your AWS services (e.g., us-west-2, us-east-1).

4. Output Format

- The format in which we want AWS CLI commands to return output. It can be one of the following:
- table
- text
- json

****

aws configure

****

aws configure

- aws ec2 describe-instances
- aws ec2 describe-instances --region ap-south-1
- aws ec2 stop-instances --instance-ids i-0680458055fe5104e
- aws ec2 start-instances --instance-ids i-0680458055fe5104e
- aws ec2 stop-instances --instance-ids i-027486740916dd1c1 --region ap-south-1
- aws ec2 start-instances --instance-ids i-027486740916dd1c1 --region ap-south-1

****

S3 buckets all commands

****

To see all buckets

- aws s3 ls

To create a bucket

- aws s3 mb s3://bucketname

To create a bucket in specific region

- aws s3 mb s3://bucketname --region region-name

To delete a bucket from a specific region

- aws s3 rb s3://bucketname --region eu-west-3

To delete a bucket

- aws s3 rb s3://bucketname

To delete a non-empty bucket

- aws s3 rb s3://bucketname --force

To see all objects in a particular bucket

- aws s3 ls s3://bucketname

To copy or upload an object to a bucket

- aws s3 cp /file1 s3://bucketname

To move an object to a bucket

- aws s3 mv /file1 s3://bucketname

To remove an object from a bucket

- aws s3 rm s3://bucketname/file1

To upload or copy the contents of dir1

- aws s3 cp /dir1 s3://bucketname --recursive

To upload or copy dir1

- aws s3 cp /dir1 s3://bucketname/dir1 --recursive

To upload or copy file1 from bucket1 to bucket2

- aws s3 cp s3://bucket1/file1 s3://bucket2

****

Types of users in AWS

****

Types of users in AWS

1. Root/Admin/Super User
2. IAM User (Normal User)

IAM

IAM stands for Identity and Access Management.

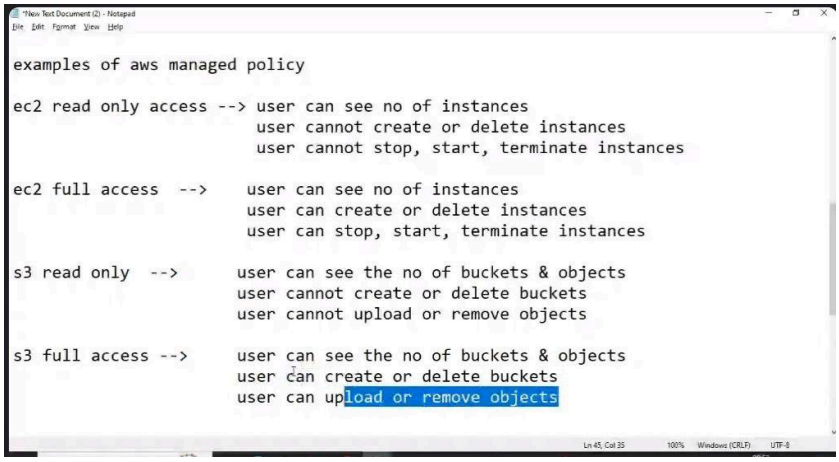
The IAM service is used to create users and groups in AWS.
With the help of IAM, we can apply permissions (policies) to IAM users and groups.

IAM is a global service.

Policy means user-to-service access

Types of policies in IAM

1. AWS Managed Policy
2. Customer Managed Policy
3. Inline Policy



****

How to create a user:

****

- **How to create a use**
- **How to create a group:**
- **Add a user to a group.**
- **Copy permissions.**
- **Attach policies.**
- **Administrator access.**
- **Change the user's password.**

****

Difference between the root user and an IAM user with administrator access

****

- What is the difference between the root user and an IAM user with administrator access?
-
- The root user can delete an AWS account, but an IAM user with administrator access cannot delete an AWS account.

****

MFA (Multi-Factor Authentication)

****

- MFA stands for Multi-Factor Authentication.
- It is used to protect both the root user account and IAM user accounts.
- MFA increases security by requiring an additional verification step to confirm the identity of the user.
- Without MFA, if someone gains access to your account password, they can easily log in and potentially make harmful changes, such as deleting resources or creating unwanted resources.
- To prevent this, MFA adds an extra layer of security.

How MFA works:

- MFA combines two factors for authentication:
- 1. The password for your account.
- 2. A security code (OTP) generated by a device you own, such as a mobile phone.

- When using MFA, after entering your email and password to log in to our AWS account, we will receive a one-time security code (OTP) on your device (e.g., mobile phone).
- We then enter this code to complete the login process.

Thus, MFA helps protect both root user accounts and IAM user accounts.

Examples of MFA applications:

1. Google Authenticator
2. Authy

****

Inline Policy

****

3.Inline Policy

- In an inline policy, we can define granular level access.

For example,

- if the requirement is for a user to view the number of instances and create instances, but not delete them, an inline policy would be the appropriate choice.
- This is because AWS managed policies are typically broader and do not offer such fine-grained control for specific actions on resources.
- Inline policies allow us to define custom permissions tailored to specific use cases, which is especially useful when you need more precise control over what actions a user can perform on resources, like allowing instance creation and viewing without deletion.

- A policy is a set of permissions that define what actions a user, group, or service can perform on specific AWS resources.
- Policies are attached to users, groups, or roles to grant them the necessary access.
- A role, on the other hand, is an AWS identity that can be assumed by users, groups, or services to gain specific permissions temporarily.

- Roles are typically used for service-to-service access.

For example,

EC2 to s3fullaccess
EC2 to s3readonlyaccess

****

AWS Lambda Service

****

AWS Lambda Service

- AWS Lambda is a serverless compute service in AWS with the help of that we can run our code without provisioning or managing servers.
- In Lambda, we create a Lambda function where we write our code.
- Lambda supports multiple programming languages, such as Python, Ruby, Java, .NET, Node.js, and Go.

Practical Steps:

1) Create an IAM Role:

- Create an IAM role lambda to
- DynamoDBFullAccess

2) Create a Bucket in S3

- Set up an S3 bucket where objects can be uploaded.

3) Create Lambda Function

- Create a Lambda function in AWS Lambda, and attach the IAM role.

4) Add a Trigger:

- Set up a trigger to invoke our Lambda function.
- For example, we can trigger it when an object is uploaded to the S3 bucket.

5) Deploy Code

- Deploy the code we wrote for the Lambda function.

6) Create Table in DynamoDB

- Create a DynamoDB table to store information.

7) Upload Objects in S3 and Check DynamoDB Table

- Upload objects to the S3 bucket and verify that entries are added to the DynamoDB table.

Deletion Steps:

1) Delete IAM Role

- Delete the IAM role that was created for the Lambda function.

2) Delete S3 Bucket

- Delete the S3 bucket where objects were uploaded.

3) Delete Lambda Function

- Delete the Lambda function from the Lambda console.

4) Delete DynamoDB Table

- Delete the DynamoDB table that was created.

For example,

```
import boto3
from uuid import uuid4
def lambda_handler(event, context):
    s3 = boto3.client("s3")
    dynamodb = boto3.resource('dynamodb')
    for record in event['Records']:
        bucket_name = record['s3']['bucket']['name']
        object_key = record['s3']['object']['key']
        size = record['s3']['object'].get('size', -1)
        event_name = record ['eventName']
        event_time = record['eventTime']
        dynamoTable = dynamodb.Table('newtable')
        dynamoTable.put_item(
            Item={'unique': str(uuid4()), 'Bucket': bucket_name, 'Object': object_key,'Size': size, 'Event':
event_name, 'EventTime': event_time})
```

Runtime
Python 3.10

****

SSM Service

****

SSM Service

- SSM means Systems Manager or Session Manager.

Purpose:

1. To connect to Linux or Windows servers without using Putty or Remote Desktop tools.
2. No need to open port 22 (SSH) or 3389 (RDP) in the Security Group.
3. No need for any key pair.
4. We can execute commands on multiple instances at the same time.

- With the help of the SSM service, we can connect to EC2 instances without using Putty or key pairs (no ports need to be allowed in the Security Group).

Steps to use SSM to connect to EC2 instances:

1. Create an IAM role for EC2 with the policy AmazonSSMManagedInstanceCore.
2. Create the instance and attach the IAM role to it.

Some AMIs already have the SSM agent installed.
For others, we can use the following commands to install and start it:

- yum list amazon-ssm-agent
- yum install amazon-ssm-agent -y
- systemctl status amazon-ssm-agent
- systemctl start amazon-ssm-agent

****

CloudWatch

****

CloudWatch

- CloudWatch is a service in AWS that allows us to monitor resources like EC2, EBS, S3, ELB, RDS, etc.
- We can monitor these resources using various metrics such as CPU, memory, network, disk utilization, etc.
- CloudWatch is a region-specific service.

Metric

- A metric is a unit of measurement.
- We use CPU utilization as the default metric to monitor the load on an instance.

Types of Alarm States:

1. Insufficient data
2. OK
3. Alarm

To monitor instances

1. Go to CloudWatch.
2. In the "Events" section,
3. select Rules
4. Click Create Rule
5. Select the EC2 service.
6. In the Event Type, choose EC2 Instance State-change Notification.
7. Add a notification topic

****

To monitor S3 bucket

****

To monitor S3 bucket

1. Go to the S3 service.
2. Select the particular bucket we want to monitor.
3. Navigate to the Properties tab of the bucket.
4. In the Events section, select Create event notification.
5. Choose the event types you want to monitor, such as ObjectCreated and
6. ObjectRemoved
7. Add an SNS topic for the notifications.
8. Ensure that we add a policy to the SNS topic to allow the S3 bucket to publish notifications to the topic.
9. This policy is necessary to allow S3 to send events to SNS.

We might need to adjust the SNS topic's access policy by editing it and allowing permissions for the S3 service to publish messages.

```
{
 "Version": "2012-10-17",
 "Id": "example-ID",
 "Statement": [
  {
   "Sid": "example-statement-ID",
   "Effect": "Allow",
   "Principal": {
    "Service": "s3.amazonaws.com"
   },
   "Action": [
    "SNS:Publish"
   ],
   "Resource": "ARN",
   "Condition": {
      "ArnLike": { "aws:SourceArn": "arn:aws:s3:::BUCKET-NAME" },
      "StringEquals": { "aws:SourceAccount": "ACCOUNT-ID" }
   }
  }
 ]
}
```

ARN
Bucket name
Account id

****

CloudTrail

****

CloudTrail

- CloudTrail is a service in AWS that is used to monitor and log user activity in our AWS account.
- With CloudTrail, we can track which user performed specific actions and identify the source of those actions.
- CloudTrail is a region-specific service, meaning it records events in the region where it is enabled.

Difference between CloudWatch and CloudTrail

- CloudWatch is used to monitor AWS resources (e.g., EC2 instances, S3 buckets, RDS databases) by collecting and tracking metrics, logs, and setting up alarms for those resources.

CloudTrail is used to monitor user activity.

- It helps track which user performed specific actions on your AWS resources, providing detailed event logs that allow us to audit and secure your environment.

Types of monitoring in cloudwatch

1. basic monitoring
2. detailed monitoring

****

CloudFront

****

CloudFront

- CloudFront is a content delivery network (CDN) that helps reduce latency and speed up the delivery of static and dynamic content to users across the globe.
- CloudFront is a global service with over 310 edge locations and 13 regional edge locations worldwide.
- Edge Locations & Regional Edge Locations

1.Edge Locations

- These are data centers located around the world where content is cached. There are more than 310 edge locations globally.

2. Regional Edge Locations

- These are intermediate caching layers between the origin and edge locations. There are 13 regional edge locations.

TTL (Time to Live)

- TTL (Time to Live) refers to the amount of time content is cached at edge locations and regional edge locations before it expires.

- By default, the TTL for edge locations is 24 hours.
- By default, the TTL for regional edge locations is 1 year.
- We can customize the TTL to a specific value (in seconds) as per your needs.

After the default TTL expires (24 hours for edge locations), the cached content is moved to the regional edge location.

- If the content is not available at an edge location, the request will be sent to the regional edge location.
- If it's also not cached there, the request is sent to the origin.

Invalidation Request

- An invalidation request is used to manually remove content from CloudFront caches (both at the edge and regional edge locations) before it expires based on TTL.

Why Send an Invalidation Request?

- When changes are made to the origin (like updating a file or content), users might still receive the old data from the cache stored at the edge location.
- To ensure users get the updated content, an invalidation request is necessary to remove the outdated cache data.

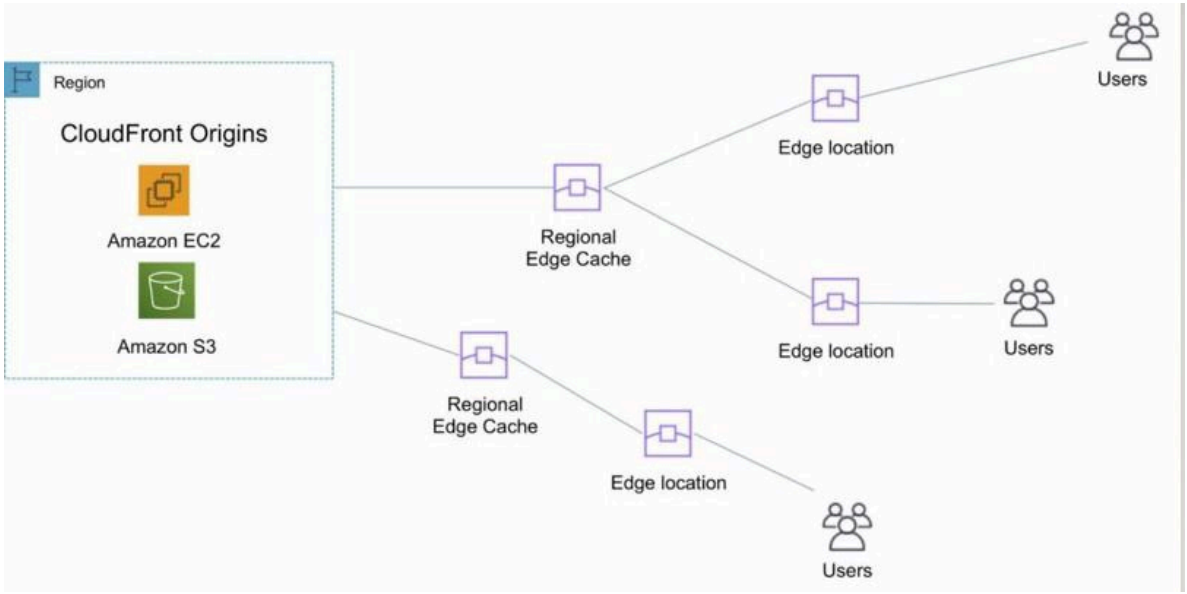
How CloudFront Works

1. When a user makes a request, it first goes to the nearest edge location.
2. The edge location checks if the requested content is in its cache:

- If the content is available in the cache, it is immediately delivered to the user, reducing latency.
- If the content is not cached at the edge location, the request is sent to the regional edge location

1. If the regional edge location doesn't have the cached content either, the request is sent to the origin (such as an EC2 instance or S3 bucket).
2. The content is fetched from the origin, sent back to the regional edge location, and then to the edge location.
3. The edge location caches the content for future requests and delivers it to the user.
4. On subsequent requests, the edge location serves the content directly from its cache, improving performance and reducing the need to contact the origin.

5. This process minimizes latency by caching content at locations closer to users.



****

Route 53

****

Route 53

- Route 53 is a service in aws which is used for domain registration, DNS routing, traffic management, and monitoring the health of resources like web servers and load balancers.
- Route 53 is a global service and can be used to manage domains and DNS records across AWS regions.

Functions of Route 53

1) Domain Registration

- Route 53 allows us to register new domain names or transfer existing ones.

2) DNS Management

- We can manage DNS records (such as A, CNAME etc.) to route traffic to resources within our AWS environment.

3) Traffic Management

- Route 53 helps manage how incoming traffic is distributed across multiple resources.
- It supports routing policies like weighted, latency-based, and geolocation-based routing.

4) Health Monitoring

- Route 53 can be configured to monitor the health of resources (such as web servers and load balancers) to ensure they are available and responsive.

Types of Top-Level Domains (TLDs) in AWS:

1) Generic Top-Level Domains (gTLDs):

- These are common domain extensions that include:
- .com, .org, .net

2) Country-Code Top-Level Domains (ccTLDs):

- These represent a specific geographic location or country. Examples include:
- .in (India), .us (United States), .uk (United Kingdom), .cn (China), .pk (Pakistan), etc.

Domain Provider/Registrar:

- A domain provider or registrar is an entity from which we purchase and manage our domain.
- AWS itself serves as a domain registrar, so we can purchase and manage domains directly through Route 53.
- However, other third-party domain registrars like GoDaddy are also popular choices.

****

Hosted zone

****

Hosted zone

- A hosted zone is a container in Route 53 that holds records for a specific domain.
- It is essentially a collection of DNS records (such as A, CNAME etc.) for a given domain.
- When we purchase a domain through AWS, Amazon automatically creates a hosted zone with the same name as the domain.
- In this hosted zone, we will typically find:

4 name server (NS) records:

- These point to the AWS Route 53 name servers for your domain.

1 Start of Authority (SOA) record

- This record contains essential information about the domain, such as the primary DNS server for the domain and the contact email address for domain administration.
- It also contains information related to domain record expiration and refresh intervals.

****

Types of routing policies in Route 53

****

Types of routing policies in Route 53

1) Simple Routing Policy

- In Simple Routing policy requests are distributed in a round-robin fashion.
- For example, if we have two servers, the first request goes to the first server, the second request goes to the second server, the third request goes to the first server again, and so on. This is a basic distribution mechanism.

2) Failover Routing Policy

- The Failover Routing policy is used to route traffic to a primary (active) server, and only when that server is unavailable (e.g., down or unreachable), traffic is routed to a secondary (passive) server.

For example,

- if we have two servers, the primary server will handle all the traffic unless it's not working.
- Once the primary server goes down, requests will be directed to the secondary server.
- When the primary server comes back online, traffic will revert to the primary server.

3. Geolocation Routing Policy

- The Geolocation Routing policy directs traffic based on the geographic location of the user.
- This can be useful when we want to direct users to the nearest or most appropriate server.

For example,

- if we have two servers, one in Mumbai and the other in Paris:
- - Requests from users in Mumbai will go to the Mumbai server.
- - Requests from users in Paris will go to the Paris server.
- This routing policy helps improve performance by serving users from geographically closer servers.

4) Latency-Based Routing Policy

- The Latency-Based Routing policy routes traffic to the server that provides the lowest latency to the user.
- This is useful for ensuring faster response times for users.

For example

- if a user is in the
- Mumbai region but the Mumbai server is under heavy load and has high latency, the request might be routed to the Paris server if it provides better latency.
- This routing policy helps improve the overall user experience by minimizing delays.

5. Weighted Routing Policy

- The Weighted Routing policy allows you to control the distribution of traffic across multiple resources by assigning different weights. We can define how much traffic each server should handle by assigning a weight value.

For example

- if we have two servers and want to send
- 70% of the traffic to the first server and 30% to the second, we would configure the weights accordingly.
- If we receive 10 requests, then:
- - 7 requests would go to the first server (70% of the traffic).
- - 3 requests would go to the second server (30% of the traffic).
- This is useful for load balancing or testing new servers.

****

AWS(Regions,Zones,Instance Limits,VPC,S3)

****

AWS

- **Regions: 36**
- **Zones: 113**

CloudFront

- **Regional edge locations: 13**
- **Edge locations: 400+**

EC2

- **Instance Limits:**
- **Default is 20 per Region. You can request a limit increase. No specific maximum limit.**

VPC

- **By default, we can create 5 VPCs per region**
- **And 200 subnets per VPC**

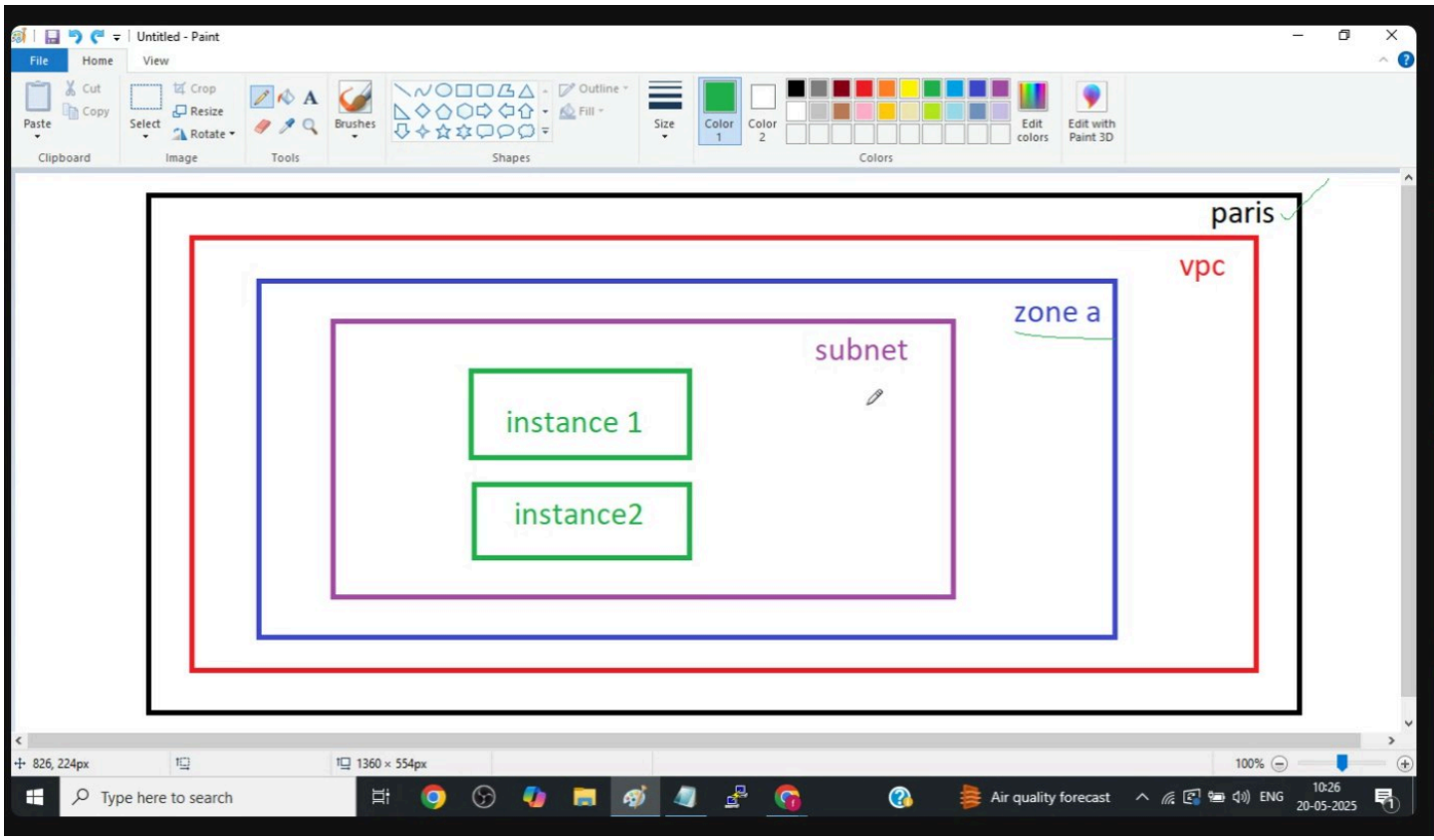
S3

- **By default, we can create up to 100 buckets in each of our AWS accounts. If we need more buckets, we can increase our account bucket limit to a maximum of 1,000 buckets by submitting a service limit increase.**
- **Maximum size of 1 bucket: 5 TB**

What is the limit of an S3 bucket?

- **The total volume of data and number of objects you can store are unlimited. Individual Amazon S3 objects can range in size from a minimum of 0 bytes to a maximum of 5 TB. The largest object that can be uploaded in a single PUT is 5 GB.**

	AWS	AZURE	GCP
LAUNCHING YEAR	2004	2010	2008
AVAILABILITY	84 availability zone & 24 Geographical locations	60+ region across all over the country	24 region & 74 total Zones
SERVICES	212+ SERVICES	200+ SERVICES	100+ Services
CLOUD SHARE	33% of the Market	21% of the Market	8% of the Market
COMPUTE ENGINE	EC2 (Elastic Compute System)	Virtual Machine	Compute Engine
NETWORKING	VIRTUAL PRIVATE CLOUD	VIRTUAL NETWORK (VNET)	CLOUD VIRTUAL NETWORK
CLIENTS	Netflix, BMW, Samsung, Unilever, Expedia	HP, Apple, Polycom, Honeywell, Johnson	LG, Toyota, Vodaphone, Spotify, New York Times
PRICING	Based on Charge per hour	Based on Charge per minute	Based on Charge per minute



****

VPC

****

1. What is VPC?
2. Why do we use VPC?
3. What is a subnet?
4. What are the types of subnets?
5. What is an Internet Gateway?
6. What is a NAT Gateway?
7. What is a NAT Instance?
8. What is a Route Table?
9. What is VPC Peering?
10. Why do we use VPC Peering?
11. What is Non-Transitive Peering?
12. What is CIDR?
13. Can we assign the same CIDR value to two VPCs?
14. Can we assign the same CIDR value to a VPC and a subnet?

****

VPC(Virtual Private Cloud.)

****

- VPC stands for Virtual Private Cloud.
- VPC is a region-specific service.
- With the help of VPC, we can create our own Virtual Private Cloud.
- VPC refers to a complete network.
- A VPC is divided into several smaller networks called sub-networks or subnets.
- Inside a VPC, we create subnets, which can be private or public.
- Within subnets, we have instances (public & private instances) where we host web servers and database servers.
- Typically, web servers are hosted in a public subnet,
- and databases are hosted in a private subnet.

****

Types of Subnets

****

Types of Subnets

1. Public Subnet
2. Private Subnet

Public Subnet

- A public subnet is a subnet associated with a public route table that has a route to an Internet Gateway.

Private Subnet

- A private subnet is a subnet associated with a private route table that has a route to a NAT Gateway.

****

Internet Gateway & NAT Gateway

****

Internet Gateway & NAT Gateway

Internet Gateway

- An Internet Gateway is a device associated with a public subnet.
- An Internet Gateway supports two-way communication,
- meaning instances can access the internet, and incoming requests from the internet can reach the instances.

NAT Gateway

- A NAT Gateway is a device associated with a private subnet.
- A NAT Gateway supports one-way communication, meaning instances can access the internet, but incoming requests from the internet cannot reach the instances.

****

Route Table (RT)

****

Route Table (RT)

- A Route Table contains a set of rules, called routes,
- that determine where network traffic from your subnet or gateway is directed.

There are two types of route tables:

Public Route Table

- A Public Route Table is associated with public instances and an Internet Gateway.

Private Route Table

- A Private Route Table is associated with private instances and a NAT Gateway.

CIDR(Classless Inter-Domain Routing)

- CIDR stands for Classless Inter-Domain Routing.
- The CIDR value determines the maximum number of subnets we can create inside a VPC
- and the number of instances/IP addresses we can create inside a subnet.

VPC Setup

1. Create a VPC
2. Create two subnets (one public subnet & one private subnet)
3. Create an Internet Gateway and attach it to the VPC
4. Create a NAT Gateway in the public subnet
5. Create two route tables (Public RT & Private RT)

Steps:

1. Create a VPC (Name = myvpc)

- Services → VPC → Your VPC → Create VPC
- CIDR Block: 10.10.0.0/16

2. Create Subnets

- Go to VPC → Subnets → Create Subnet (Public Subnet in Zone A)
- Select VPC: myvpc
- CIDR Block: 10.10.1.0/24
- Add a new subnet
- Go to VPC → Subnets → Create Subnet (Private Subnet in Zone B)
- Select VPC: myvpc
- CIDR Block: 10.10.2.0/24
- Click "Create Subnet"

3. Create an Internet Gateway

- Go to VPC → Internet Gateway → Create IGW (Name: pub-sub-igw)
- Attach this gateway to VPC (myvpc)

4. Create a NAT Gateway in the public subnet

5. Create Route Tables

- Go to VPC → Route Table → Create Route Table (Name: Public RT)
- Select VPC: myvpc
- Edit Route
- Add Route
- Destination: 0.0.0.0/0
- Target: Internet Gateway (igw)
- Subnet Association → Select Public Subnet

- Go to VPC → Route Table → Create Route Table (Name: Private RT)
- Select VPC: myvpc
- Edit Route
- Add Route
- Destination: 0.0.0.0/0
- Target: NAT Gateway
- Subnet Association → Select Private Subnet

6. Launch EC2 Instances

- Go to EC2 → Launch Instance
- Select Availability Zone A
- In "Configure Instance," select the new VPC
- Enable Public IP
- In Security Group (SG), create a new SG and allow SSH, HTTP, and ICMP protocols
- Security Groups are VPC-specific
- Repeat the above steps to create a second instance in the private subnet

Deletion Steps

1. Delete EC2 instances
2. Delete the NAT Gateway
3. Release the Elastic IP
4. Delete the VPC (subnets, IGW, and route tables will also be deleted)

Accessing Instances

- Try accessing the first instance (which is in the public subnet) → It should be accessible.
- Try accessing the second instance (which is in the private subnet) → It will not be accessible because the NAT Gateway supports one-way communication.

Now, connect to the first instance (which is in the public subnet).

- Create a file and copy the private key:
- vim keypair.pem
- chmod 400 keypair.pem

- SSH into the second instance (which is in the private subnet) from the first instance:
- ssh -i keypair.pem ec2-user@private_ip_of_2nd_m/c

Now, we can download packages:

- yum install -y mysql
- ping google.com

Diff bet Nat Gateway & Nat instance

- Nat Gateway and Nat Instance both support one-way communication (meaning we can access the internet from the server, but incoming requests from the internet cannot reach the server).
- Nat Instance is cheaper compared to Nat Gateway.
- Nat Instance cannot handle a large number of requests, whereas Nat Gateway can handle a large number of requests.

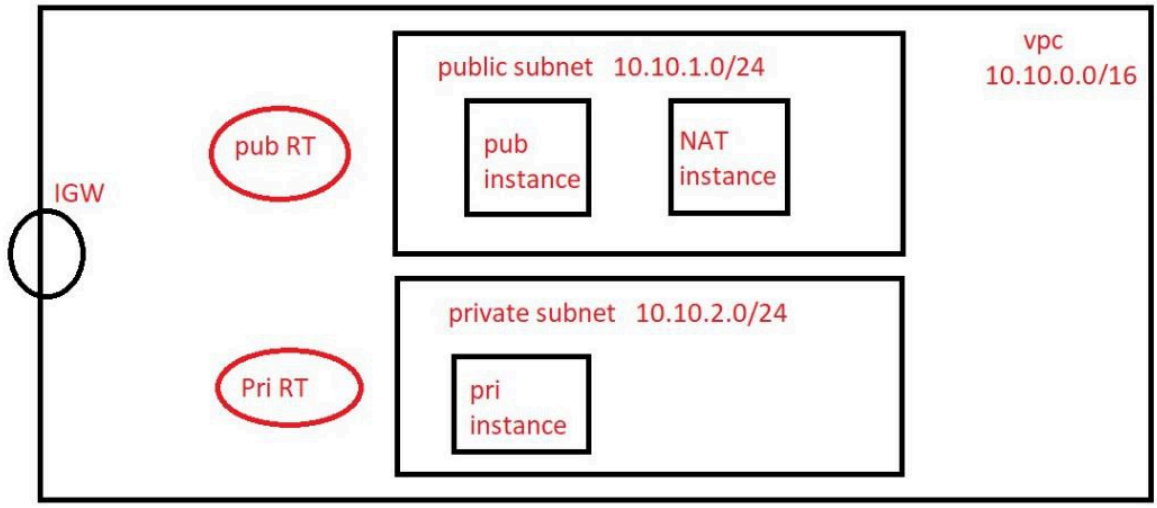
setup of nat instance

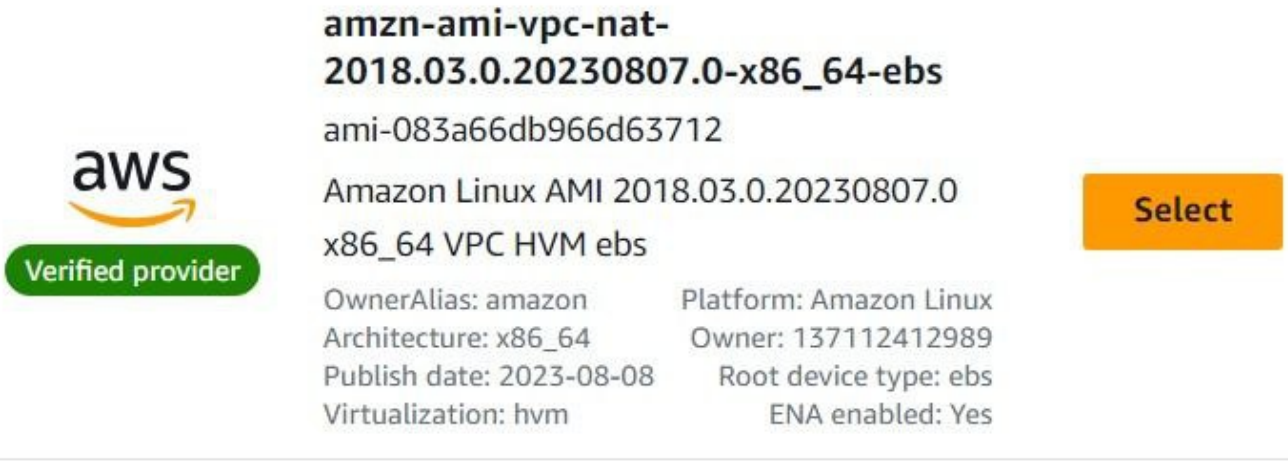
1. Go to Community AMI,
2. Search for "nat" and select the first AMI.
3. In the "Configure Instance" step, select your VPC, public subnet, and enable the public IP.
4. In the Security Group (SG), allow all traffic in inbound rules.
5. Select the Nat Instance, go to "Actions," then "Networking," and choose "Change Source/Destination Check."
6. Inside that option, there is a "Stop" setting; just tick the "Stop" option.
7. Enter the Nat Instance in the private route table.
8. Connect to the Nat Instance using PuTTY.
9. Create a file and paste the key pair data.

Use the command:

- ssh -i keypair username@private_ipaddress
- While creating a Nat Instance, add all traffic in the Security Group (SG).

- Go to Actions → Networking → Change Source/Destination Check→ Stop





****

VPC Peering

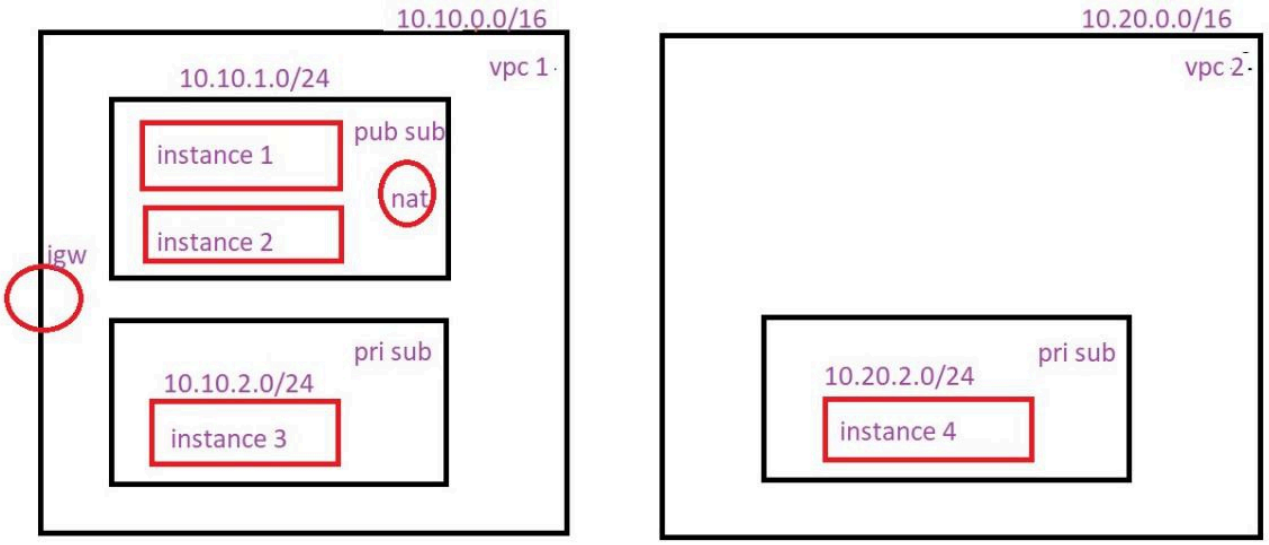
****

VPC Peering

- VPC peering is used to connect instances that are in different VPCs, whether they belong to the same account or different accounts.
- However, both instances must be in the same cloud provider, such as AWS, GCP, or Azure.
- With VPC peering, we cannot connect instances that are in different cloud providers.

For example,

- if one instance is in AWS and the other is in GCP or Azure, VPC peering cannot be used.
- Similarly, VPC peering cannot be used to connect instances between on-premises and the cloud.



****

VPN(Virtual Private Network)

****

VPN(Virtual Private Network)

- VPN stands for Virtual Private Network.
- VPN is used to connect instances that are in different clouds.

For example,

- if one instance is in AWS and the second instance is in GCP or Azure, we can use a VPN to connect these two instances.
- Additionally, with the help of a VPN, we can connect instances between on-premises and the cloud.

Transit Gateway

- AWS Transit Gateway connects our Virtual Private Clouds (VPCs) and on-premises networks through a central hub.
- Suppose we have five VPCs and need to create connections between all of them. In that case, we would have to set up 10 VPC peering connections. Instead of doing this, we can create a Transit Gateway and connect all five VPCs to it, allowing them to communicate with each other efficiently.

CIDR

- CIDR stands for Classless
- Inter-Domain Routing.
- The CIDR value determines the maximum number of ipaddresses we can assign within a VPC and within a subnet.

How to calculate the maximum number of ip addresses we can assign within a VPC using CIDR?

Suppose we have the CIDR 10.10.0.0/28

Here, /28 is called the prefix

- 32 - prefix
- 32 - 28 = 4

Now, assume this value 4 as n and use the formula:

- 2^n
- $2^4 = 16$

So, with the /28 range, we can assign 16 ipaddresses within one VPC.

CIDR Notation	Addresses	Addresses
/8	2^{24}	16,777,216
/9	2^{23}	8,388,608
/10	2^{22}	4,194,304
/11	2^{21}	2,097,152
/12	2^{20}	1,048,576
/13	2^{19}	524,288
/14	2^{18}	262,144
/15	2^{17}	131,072
/16	2^{16}	65,536
/17	2^{15}	32,768
/18	2^{14}	16,384
/19	2^{13}	8,192
/20	2^{12}	4,096
/21	2^{11}	2,048
/22	2^{10}	1,024
/23	2^{9}	512
/24	2^{8}	256
/25	2^{7}	128
/26	2^{6}	64
/27	2^{5}	32
/28	2^{4}	16
/29	2^{3}	8
/30	2^{2}	4

NACL

- NACL stands for Network Access Control List , which controls incoming (inbound) and outgoing (outbound) traffic from subnets.
- NACL deals with port numbers and is applied at the subnet level.

Difference between SG & NACL

- SG (Security Group) is applied at the instance level, whereas NACL is applied at the subnet level.
- In SG, we can only allow ports; we cannot deny them.
- In NACL, we can allow as well as deny ports.
- In NACL, rules are prioritized by rule numbers.
- - A lower number has higher priority.
- - For example, if Rule1 denies HTTP and Rule2 allows HTTP, NACL will deny HTTP.