Cloud Computing Demo

Who can learn this course?

Anyone with minimum graduation

1. B.com
2. BA/MA
3. B.Sc
4. BCA

I'm working for BPO. Can I learn and work in the cloud?

Yes, you can learn and work on the cloud.

I'm 2015/2016/2017 passedouts, how can I get into this field?

Yeah

What Are the opportunities and job calls for this?

All technologies and applications use cloud computing as a common platform, the demand is high for cloud computing.

AWS Cloud Computing

1. After completing this course, you can apply for AWS jobs
2. ***Covers AWS Solutions Architect Associate Certification***
3. This course covers advanced concepts like lambda, python and terraform
4. Duration of the course (50-60 days)

What is cloud computing?

Cloud computing is delivery of compute resources like, virtual servers, databases, networking, storage, load balancers, security, etc.. over the internet. In cloud computing we pay only for what we use.

Before the cloud, how customers were using the Infrastructure?

Customers have their own on-premise, data centers.

Benefits of cloud computing

1. Cost
    a. No upfront investment needed for purchasing infrastructure.
    b. You get topclass infrastructure for low cost.
2. Maintenance
    a. Maintaining data centers, servers and all is not our headache, it is taken care of by cloud providers.
    b. It gives a way for us to focus on our customer values.
3. Flexibility/Elasticity
    a. We can instantly add and remove servers. This flexibility will not be there while using on-premise.
4. Global Scale
    a. You can scale you applications globally in minutes
5. Agility
    a. Anything you want you get in a few minutes, but on premise you have to wait for weeks to months.
6. Security
    a. Cloud offers high standard security features, security is always shared responsibility that is it is responsibility of cloud provider and tenants.

Cloud Service Offerings (FAQ)

Cloud service offering are are

1. IaaS (Infrastructure as as Service)
    a. We will launch the server(vm)
    b. We will maintain the server for example patching etc.
2. PaaS (Platform as a Service)
    a. Here infrastructure (vm) is created by cloud provider
    b. The maintenance work like patching is taken care by cloud providers
    c. The application like tomcat/DB is also installed by the cloud
    d. We just use it.
3. SaaS (Software as a Service)
    a. We consume servers over the internet.
    b. Examples are gmail. Google, facebook, linkedin, etc…

Cloud Deployment Models

1. Public Cloud
    a. All IT resources are managed by a third party like AWS in their location.

    b. IT resources are not specific to customers, storage, networking devices, and hardware is shared by customers
2. Private Cloud
    a. The resources like hardware, storage, networking devices are specific to the customer, private cloud can be hosted with third party or private cloud be hosted at customers site.
    b. Private cloud is costlier than public cloud
3. Hybrid Cloud
    a. It is combination of public and private clouds

Why AWS? Why not Azure?

1. AWS has more services than any other cloud platform.
2. They work with customer feedback, they release updates and new serverviices aggressively.
3. AWS having more customers and more market share.

AWS Cloud Computing concepts

I want to host my website in the cloud, for that I want

1. VM to run my web application
2. Database
3. Domain and DNS

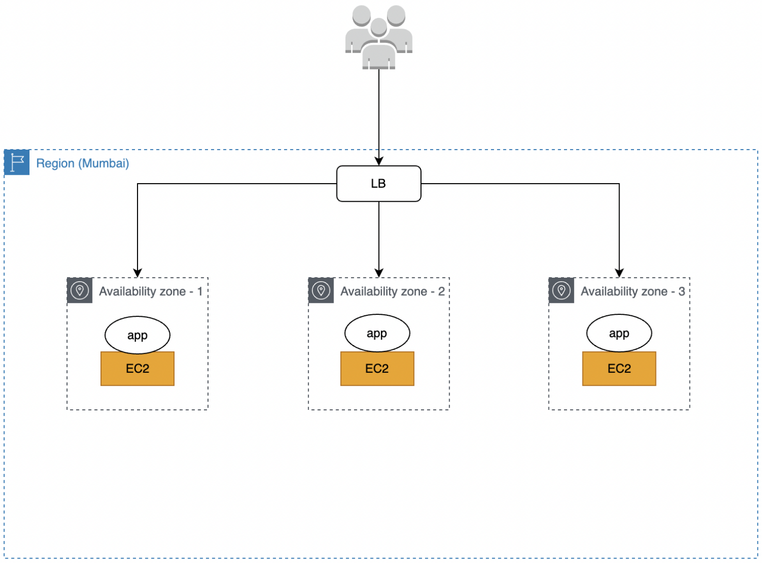
His web may be accessed by 500 people in a day
1 CPU, 1GB Memory, 20GB
For DB
1 CPU 1 GB Memory 50 GB
Route53 is DNS in AWS

Signup for free AWS account

AWS offers 1 year free account
https://aws.amazon.com/free
It will ask for email id
It will ask for credit card /debit card (don’t worry your money is safe)
It will ask for account verification using SMS or call
Follow along and get the account created

AWS Global Infrastructure

AWS maintains data centers across the globe, like, singapore, india, usa, sydney, south africa, etc.



Region

Region is a geographical location where AWS hosts its services, example regions are

- Mumbai
- Singapore
- Sydney
- Ireland
- Cape-town
- etc…

Availability Zone

1. Each region is divided into availability zones.
2. Each AZ, is isolated from other, failure in an AZ will not affect other AZs
3. To achieve HA for applications use multiple AZs.

AWS EC2 (Elastic Compute Cloud)

1. EC2 is virtual server in the cloud
2. We run application on EC2

What is a virtual machine?

https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine

Demo Launch Linux EC2 and Connect to EC2

1. AMI (Amazon Machine Image),
    a. Is a virtual machine template for launching ec2 instances
    b. There are AMIs for windows, ubuntu, macos, linux, etc…
2. Instance Type
    a. Here we pick CPU and memory
3. Configure Instance Details
    a. VPC, Subnet, etc.. will talk more about this later
4. Storage
    a. This is hard disk for your virtual machine
5. Tags
    a. Tags enables to add meta information
6. Security Groups
    a. Security group is a firewall for ec2 instance.
    b. More about this later
    c. For now allow traffic from everywhere

Connect to Linux EC2 Instance

1. Connect from AWS management console
2. Connect using SSH clients
    a. SSH is a protocol used for connecting to remote linux machine
    b. If we are connecting from windows laptop
        i. In earlier versions of windows SSH was not pre installed, and you have to install a tool for this(putty) or mobaxterm
        ii. In latest versions of windows it comes with powershell
    c. To work with putty, pem should be converted to ppk
3. Connect from macbook/ubuntu
    a. They got built in SSH
    b. Open terminal

Exercise

1. Launch Linux EC2 instance and connect with putty
2. Terminate EC2 instance

Install MobaXterm for Windows

Exercise → Install Mobaxterm and connect to Linux server
Exercise → Launch Windows EC2 instance and connect

AWS EC2 free tier limits

1. Only t2.micro is offered in free trier
2. You get 750 ec2 running hours per month.
3. AMIs also should be free tier eligible
4. Billing stops if ec2 stops
5. Storage has separate bill
    a. 30 GB is free under free tier
    b. Beyond 30Gb you are billed.

Demo - I want to run a website on AWS

1. Launch Linux EC2 instance
2. Ssh into ec2 and install and start web server
3. Put a website on web server
    a. To run website you need web server
    b. There are so many different web servers in the market
    c. Lets use apache web server
        i. sudo yum install httpd -y
        ii. sudo service httpd start
        iii. sudo service httpd status
    d. Place html file on apache
        i. sudo vi /var/www/html/index.html
4. Access it from a web browser.

Different Types of EC2 Instances

Customers will have different types of workloads for example

- Web applications
- Databases
- Batch Jobs
- Containers
- Analytics
- Graphics based applications

General Purpose

1. It is used for dev and test environments, these instances offer balanced CPU, Memory, Networking performance.

2. It is used for running internal applications, like jenkins, gitlab, sonar, etc..
3. Use this for running applications like
    a. Web applications
    b. Database
    c. Virtual desktops
    d. Anything.
    e. We can use this in prod as well for less critical applications.

We are running jfrog on m4

Compute Optimized

These instances are designed for CPU, it is designed for CPU intensive workloads
Note: Choose this type for applications which demand high CPU for permonces.

GPU optimized

Use this for GPU intensive workloads, like video rendering, audio rendering, image processing, etc..

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

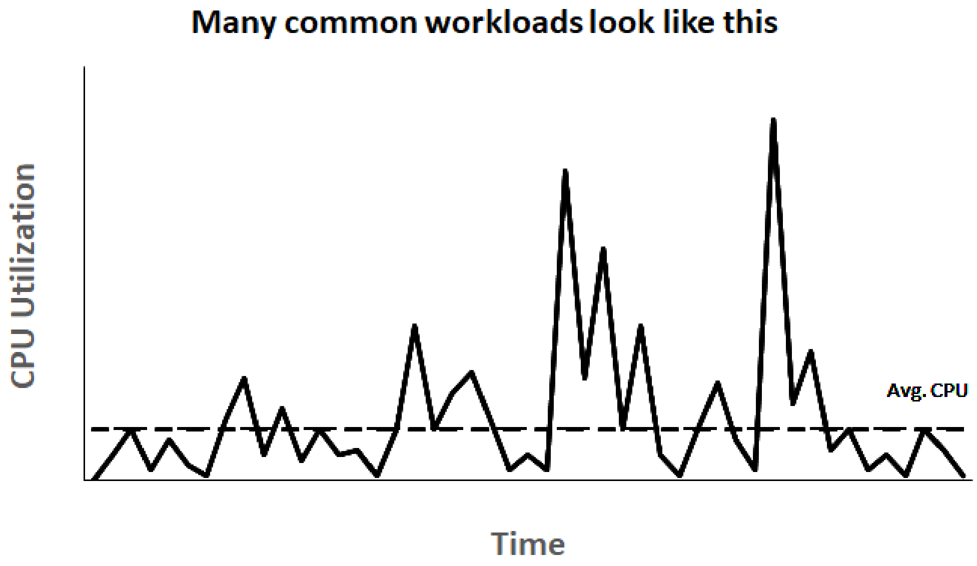
Accelerated Computing

Check by yourself

General Purpose Burstable Instance

These low-to-moderate CPU utilization workloads lead to wastage of CPU cycles and, as a result, you pay for more than you use. To overcome this, you can leverage the low-cost burstable general purpose instances, which are the T instances.
Example applications with such behaviour:

1. My website (www.javahome.in)
2. Jenkins
3. development and test server
4. github



How burstable instance work

1. If we take 1 CPU burstable instance, its baseline performance, I can offer more than 1 CPU based on CPU credits.
2. Burstable instances accumulate credits when idle. Credit points are spent when EC2 wants to perform above baseline
3. The number of points it accumulates

Naming convention of EC2 instances
t2.2xlarge
t - stands for general purpose burstable
2 - second generation
2xlarge → represents the size
Note: Always use latest generation

AWS pricing calculator

https://calculator.aws/#/

AMD processors

It offers more performance with less cost

t3.medium → 0.0448 (2CPU, 4GB Memory)
t3a.medium → 0.0246 (2CPU, 4GB Memory) → a indicated AMD processor

EC2 Instance Purchasing Options

AWS offers different purchase options to optimize the cost, as a solutions architect we should be aware of purchase options to optimize the cost in the project.

1. On-demand Instances
2. Reserved Instances
3. Spot Instances
4. Dedicated Hosts
5. Scheduled Instances
6. Savings Plans
7. Etc.

On-Demand Instances

1. Per hour billing is higher than other purchase options
2. You can launch it any time and terminate it anytime.
3. Billing stops when you stop/terminate an instance.
4. We use this when we do not have long term commitments
5. We use it for doing POC (Proof Of Concepts)
6. For example flipkart announces a big billion day for 7 days, we want additional capacity(ec2) for 7 days, on-demand is suitable.
7. By default what we use is on demand.

Reserved Instances

1. If we know we want instances for the long term, this is a good option.
2. We give long term commitment. Like 1 year or 3 years in return we get significant discounts
3. You can save upto 50% as a discounts
4. After you make a reservation you can't cancel it. But you can sell it in the AWS marketplace.
5. Billing will not stop even if you stop all instances.
6. There are different payment options
    a. All upfront
    b. Partial upfront
    c. No upfront

Spot Instances

1. Spot instances are allocated from unused capacity from AWS datacenter
2. It offers upto 90% discounts
3. It has spot interruption, when spot interruption occurs, amazon takes the instance by giving 2 minutes notice.

4. 2 minutes notice helps to grab a log file on the instance before it is terminated.
5. We can't deploy applications totally depending on spot, we have to mix and match, spot, reserved/ondemand.
6. When spot interruption can occur
    a. When spot price goes high
    b. If underlying hardware fails
    c. If AWS is running out of capacity

Persistent Spot Request Instance (Certification Question)

If we choose persistent request, for example spot interruption occurred due to max price
It will resubmit a spot request again.

Dedicated Hosts

1. Dedicated Hosts are used to bring your own license (BYOL).
2. If a customer is migrating to the cloud and he has licenses with him and those licenses should be used in the cloud. Then go with Dedicated Hosts.
3. AWS allocates a complete physical host, and all instances will be placed on this physical host.

Scheduled Instances

- This is similar to reserved, but this is used for batch jobs that run in specific schedules
- With Scheduled Reserved Instances, you can reserve capacity that is scheduled to recur daily, weekly, or monthly, with a specified start time and duration, for a one-year term

Savings Plans

Capacity Reservations

(faq)Which instance types are you using?

1. m5a.large (2CPU, 8GB RAM) using it for gitlab, sonar, jfrog
2. c5a.xlarge (4CPU, 8GB RAM) in production

(FAQ)Can we change instance type?

Yes, we can change instance type, to change this, the instance must be in a stopped state.

AMI (Amazon Machine Images)

1. It is a template for launching virtual machines

2. It can contain specific OS, applications and tools pre-configured.
3. We can use AWS managed AMIs or we can create custom AMIs as per our project needs.
4. There are different use cases for using custom AMIs
    a. We want to use hardened(secured) AMI for running applications
    b. For instance, We want all ec2 instances to be integrated with AD
    c. For example, We want the web application pre-installed and configured so that it can be used in auto-scaling.
5. AMI is region specific, images in mumbai will not be present in other regions unless you copy it.

AMI Permissions

1. AMI is by default private, that is users in same AWS account will have access
2. AMIs can be shared with other AWS accounts
3. AMIs can be made public, making public will be visible for all public accounts.

Creating Custom AMIs

1. Choose any existing AMI
2. Launch EC2 instance
3. SSH into it and configure the way you want your AMIs to be.
4. Select EC2 → Actions → Image and Templates → Create Image
5. Delete the instance used for creating custom AMI

Realtime Tip:

- Above steps can be automated using ***Hashicorp Packer***
- Packer AWS | Packer Create AWS AMI | Create golden images using hashicorp pa…

Demo Create Custom AMI

1. Launch EC2 from existing AMI
2. Install softwares and packages you want on this instance
    a. ssh into the instance
    b. sudo yum install httpd -y
    c. sudo vi /var/www/html/index.html
        i. Presss i to get into INSERT mode
        ii. Type any text into it
        iii. Press esc button :wq
    d. sudo service httpd start
    e. sudo chkconfig httpd on
        i. The above command enables apache on system reboot
3. Select EC2 got to actions and create image
4. Find image under AMI

Launch from above Image

Using AMI as a EC2 backup

For example we are running mysql database on EC2, and we want to take backup every 3 hours, this can be done using AMI

To automate EC2 backups

1. Write python and schedule it (lambda, this is discussed in later part of the course)
2. Use Data Lifecycle Manager (no code required)

Python Video

Python fundamentals in 60 minutes | Easy python tutorial for beginners | Learn python pro…

Amazon Data Lifecycle Manager

Automate creation, deletion, retention, copy of AMIS and snapshots.

Create lifecycle manager with following

1. Create Image every 3 hours
2. Set retention as 30 days
3. Copy images to cross region, ireland, so it will be useful for disaster recovery.

Python → lambda → every 3 hours it triggers

EBS(Elastic Block Store) Volumes

1. It is highly durable, scalable, *block level* persistent storage.
2. EBS is a hard disk for EC2 instance
3. We can attach one or more EBS volumes to EC2
4. On EBS volumes you can run operating systems, applications like web, databases.
5. Volume is AZ specific, instance and its volumes must be in same AZ
6. We can modify volume attributes at any time, that is we can change size, type, IOPS etc.
7. We can detach volume from an instance and attach to another instance
8. EBS supports encryption for securing data on the disk

(FAQ)What is root volume in EC2?

It is a primary disk on which the operating system is running.

(FAQ) Can we encrypt root volume?

Yes, we can encrypt root and any volume

(FAQ)Can we attach multiple single volumes to multiple instances?

Yes, depending on instance types and volume types

Types of Volumes

To understand types of volumes we should understand following attributes

1. IOPS(input Output Operation Per Second)
    a. It is number of reads and writes from and to the ebs volume
    b. The data size of read and write is small, it will be in kbs/mbs
    c. Databases need better IOPS for scaling.
2. Throughput
    a. The amount of data it can read and write per second
    b. Data Warehouse, big data analytics depend on throughput
    c. The size will be in mbs/gbs

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications. The volumes types fall into these categories:

- Solid state drives (SSD) — Optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.
- Hard disk drives (HDD) — Optimized for large streaming workloads where the dominant performance attribute is throughput.
- Previous generation — Hard disk drives that can be used for workloads with small datasets where data is accessed infrequently and performance is not of primary importance. We recommend that you consider a current generation volume type instead.

Note: HDD can’t be used as root volume

Solid state drives (SSD)

The SSD-backed volumes provided by Amazon EBS fall into these categories:

- General Purpose SSD — Provides a balance of price and performance. We recommend these volumes for most workloads.
- Provisioned IOPS SSD — Provides high performance for mission-critical, low-latency, or high-throughput workloads.

For more details about volume types check this link

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

How pricing works for EBS

1. General Purpose we are charged for size of the volume
2. Provisioned IOPS we are charged for size and IOPS both

Few points about HDD

1. Throughput Optimized HDD
   a. For frequently accessed data
2. Cold HDD
   a. For infrequently accessed data
   b. Using Cold HDD reduces cost

EBS Snapshots (volume backups)

We use backups so that in case of failures we restore from backups.
We automate ebs backups using Data Lifecycle Manager.

Exercise:

1. Automate EBS snapshot creation using DLM
2. Retention 10 days
3. Frequency 12 hours
4. Have copy in cross region with 3 days retention

Note: In AWS by default all backups go to S3

Snapshots are incremental, that is then the next backup contains only the changes happened after the previous backup.

Incremental backups incurs less cost.

Adding additional disks to EC2 instance(faq)

1. Create Volume
2. Attach volume to EC2
3. SSH in to EC2 instance
    a. lsblk
    b. The above command displays list of disks
    c. We have to mount the attached volume so that we can use it.
        i. Create filesystem on the volume
        ii. sudo mkfs.ext4 /dev/xvdf
        iii. Create mount point
        iv. sudo mkdir /mongodb
        v. sudo mount /dev/xvdf /mongodb/

Resizing a disk

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html#extend-file-system

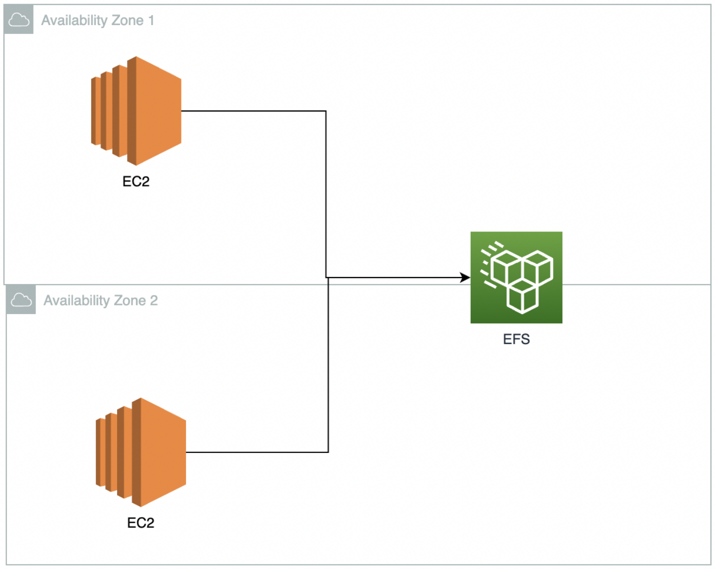
Instance Store (FAQ)

EC2 instances can be launched using either EBS or instance store.

1. Instance store is ephemeral, that is temporary, it is tightly coupled with ec2 lifecycle.
2. Data on instance store is lost on following scenarios
    a. Stopping EC2
    b. If ec2 is relaunched on different hardware because of hardware failures.
3. Its cost is very low, but it does not suit all the requirements
4. There are few use cases where you can use an instance store and reduce the cost.

AWS EFS (Elastic File System)

1. EFS is highly durable, scalable network file system in AWS
2. EFS can be attached to multiple instances in multiple availability zones
3. It is used by linux based instances
4. *For windows based instances use FSx*.
5. EFS supports encryption at rest and encryption in transit
6. You can backup EFS



EFS Demo

1. Create EFS
2. I wanna Create 2 EC2 instance in different AZs
3. Mount EFS on both the instances
4. Test EFS

AWS VPC (Virtual Private Cloud)

1. VPC is a virtual data center that isolates your resources from other tenants.
2. VPC offer better security to our resources
3. In VPC we put, ec2, lambda, databases and other compute resources.
4. The good practice is to set up applications inside VPC.
5. VPC spans a region for example mumbai, it spans all AZs in the region.
6. VPC is free
7. We can have one or more VPCs
8. For example we can have vpcs for different environments
   a. Dev environment
   b. Prod environment
   c. Test environment
9. Setting up vpc is a one time job, it is not a day to day job.

IP versions

1. There are two version ipv4 and ipv6
2. Ipv4 uses 32 bits

3. Ipv6 uses 128 bits
4. If bits are more, ip addresses space is more.

Creating VPC

- While creating vpc we should think about the number of ips required.
- To design ip address space and networking we have to use CIDR notations

CIDR (Classless Inter Domain Routing)

CIDR Example-1

- I want to design VPC with 100 ip space
- For example this is CIDR 200.10.0.0/25
- In the above CIDR /25 is called netmask
  - 24 represents network bits
  - 32-25 = 7 bits represents host bits(the bits used for ip addresses)

CIDR Example-2

I want VPC with 500 plus ips and I want two subnets

- For 500 ips host bits must be 9 bits
- Remaining is 32-9 = 23 goes to network

Let's choose following CIDR for VPC

10.200.0.0/23

For subnet-1 the CIDR should allocate 256 ips

10.200.0.0/24

For subnet-2 the CIDR should allocate 256 ips

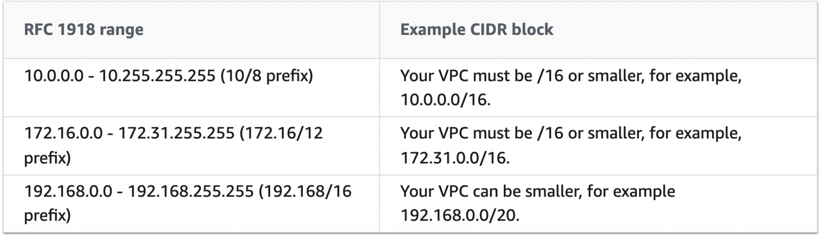
10.200.1.0/24

VPC validations

- Netmask must be between /16 and /28

VPC CIDR best practices (faq)

1. Don't overlap CIDR with other VPCs
2. Don’t overlap CIDR with on-premises
3. Keep CIDR in private IP space

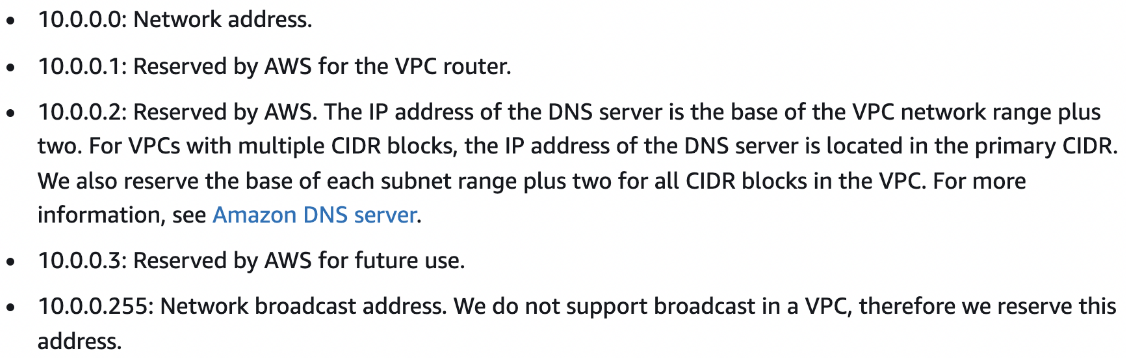


RFC 1918 range	Example CIDR block
10.0.0.0 - 10.255.255.255 (10/8 prefix)	Your VPC must be /16 or smaller, for example, 10.0.0.0/16.
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)	Your VPC must be /16 or smaller, for example, 172.31.0.0/16.
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)	Your VPC can be smaller, for example 192.168.0.0/20.

For practicing VPC use following CIDRS

172.16.0.0/16 → For VPC
172.16.0.0/24 → For Subnet-1
172.16.1.0/24 → For Subnet-2
172.16.2.0/24 → For Subnet-3
172.16.3.0/24 → For Subnet-4
172.16.4.0/24 → For Subnet-5
172.16.5.0/24 → For Subnet-6

AWS reserves 5 IPs for each subnet



- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC router.
- 10.0.0.2: Reserved by AWS. The IP address of the DNS server is the base of the VPC network range plus two. For VPCs with multiple CIDR blocks, the IP address of the DNS server is located in the primary CIDR. We also reserve the base of each subnet range plus two for all CIDR blocks in the VPC. For more information, see Amazon DNS server.
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

(FAQ)How are you connecting on-premises servers with VPC?

Using VPN connections

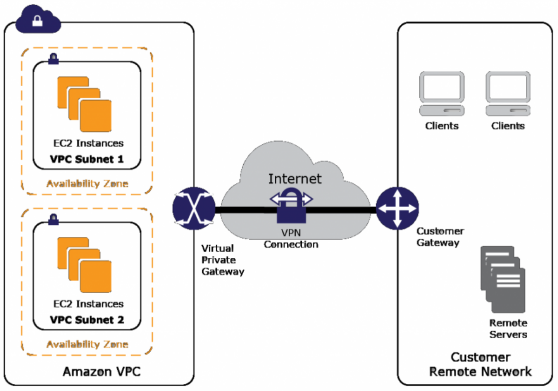


Figure 1: AWS managed VPN

The ways to connect on premises with VPC

1. VPN
    a. The communication happens over internet
    b. The performance depends on internet speed
    c. It is secured, because data is encrypted in transit
    d. If you are migrating huge data, then consider using direct connect.
2. Direct Connect
    a. This is a dedicated tunnel between on premises and AWS
    b. It is fast network
    c. It is highly secure, it doesn't use the internet.
    d. It is very costly.
3. Jump Box
    a. Jump box is ec2 instance with public IP exposed to internet
    b. Customers from on premises will ssh into jump box, from jump box they access ec2 instance in the vpc.

How to configure VPN

https://youtu.be/9Lk-ceYpSfU

(faq)What is a public subnet?

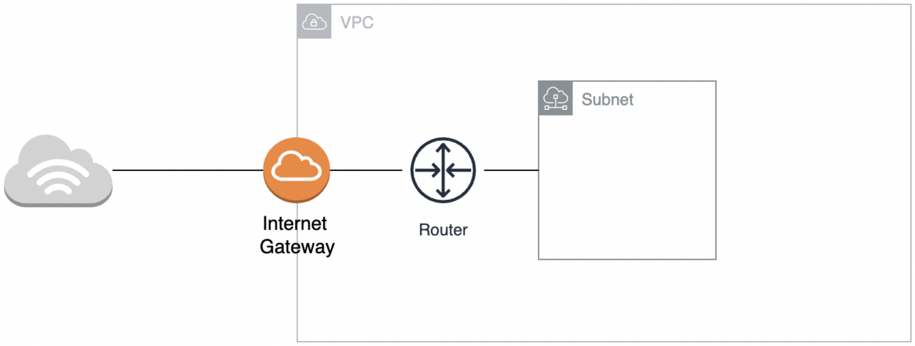
A subnet that is configured with the internet is called a public subnet, In aws we use internet gateway for public subnets.
Note: *Internet gateway offers both inbound and outbound internet connections*.

Internet Gateway

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet.
Internet gateway offer both inbound and outbound internet traffic

Demo - Configure Internet Gateway for a Subnet

1. Create VPC (172.16.0.0/16)
2. Create Subnet (172.16.0.0/24)
3. Create internet gateway
    a. There can be only one internet gateway to vpc at a time.
    b. It is fully managed by AWS (PAAS)
4. Attach internet gateway to VPC
5. Configure internet gateway in the route table



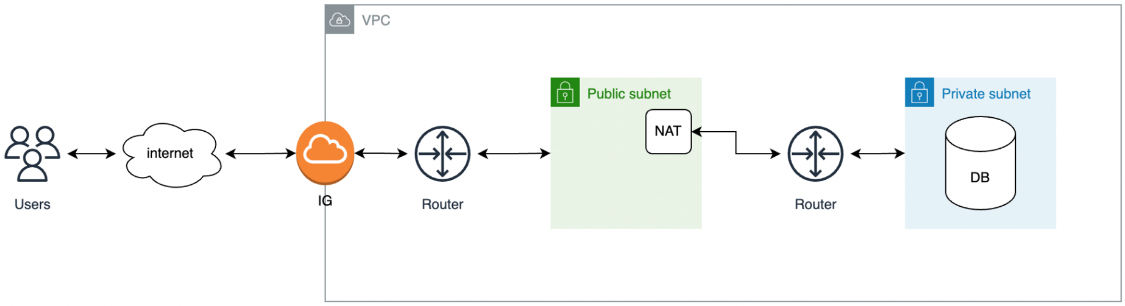
    a. Add a route in the route table

(faq)What is a private subnet?

A subnet which is not exposed to internet is private subnet

(faq) EC2 instances in a private subnet want to download software patches from the internet, how do you configure the internet in this case?

We should configure NAT, NAT allows only outbound internet connections.



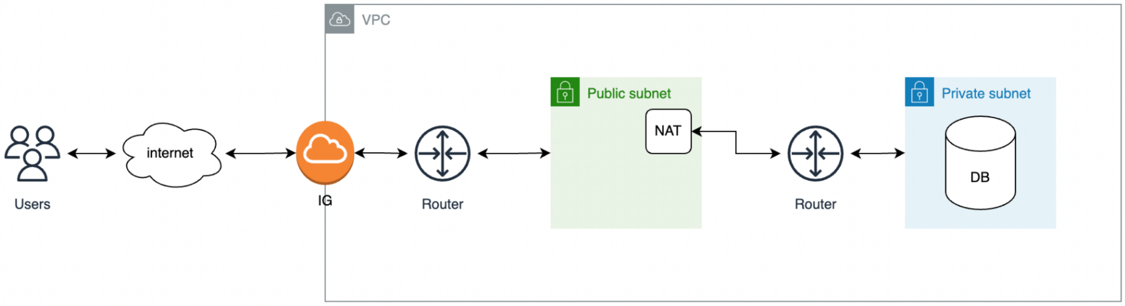
(FAQ)In which subnet NAT should be kept?

NAT should be placed in the public subnet, and traffic from private subnet should be routed to NAT instance.

AWS offers two types of NAT

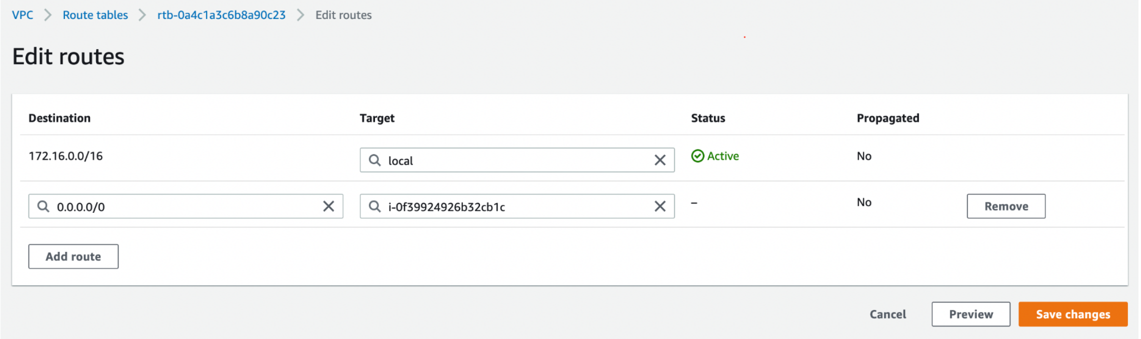
1. NAT instance
2. NAT Gateway (more commonly used)

NAT Instance Demo



1. Create VPC (172.16.0.0/16)
2. Create Public Subnet (172.16.0.0/24)
    a. Create internet gateway
        i. There can be only one internet gateway to vpc at a time.
        ii. It is fully managed by AWS (PAAS)
    b. Attach internet gateway to VPC
    c. Configure internet gateway in the route table
3. Create Private Subnet (172.16.1.0/24)

   a. Create separate route table for private subnet
   b. Associate private subnet with private route table
4. Launch NAT Instance in public subnet with public ip.
   a. You do this the way you launch EC2
   b. Select private route table
   c. Add route to NAT



VPC > Route tables > rtb-0a4c1a3c6b8a90c23 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
172.16.0.0/16	local	Active	No	
0.0.0.0/0	i-0f39924926b32cb1c	-	No	Remove

   d. Disable source destination check on NAT instance
      i. Select NAT instance → Actions → Networking → Change source/Destination check → stop.

Testing NAT instance configuration

- Launch ec2 instance in private subnet
- Ssh into above ec2 instance
- And ping google.com

Difference between NAT Instance and NAT gateway?

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

Note: In real time NAT gateway is more used over NAT instance

NAT Instance Source Destination Check

- In AWS by default source and destination check happens for all ec2 instances.
- We should stop source and destination check for NAT instance

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html#EIP_Disable_SrcDestCheck

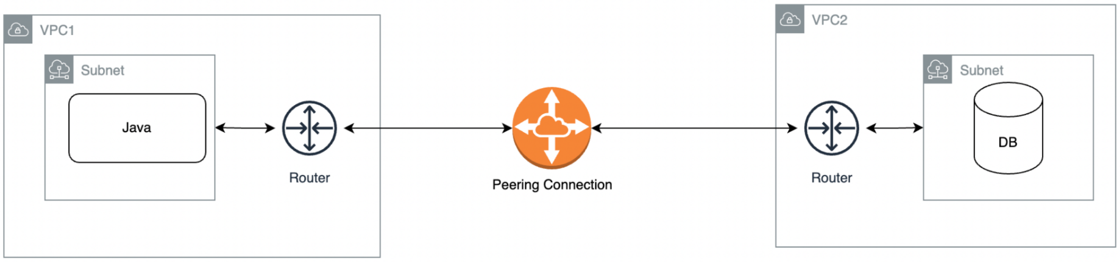
Bastion Host / Jump Box

https://aws.amazon.com/blogs/security/how-to-record-ssh-sessions-established-through-a-bastion-host/

Default VPC

1. It is VPC implicitly created by aws in every region when you create aws account
2. It contains only public subnets
3. By default ec2 instances are launched in default vpc
4. By default ec2 instances get public ip
5. We can delete default VPC
6. We can recreate if it is deleted

VPC Peering Connections

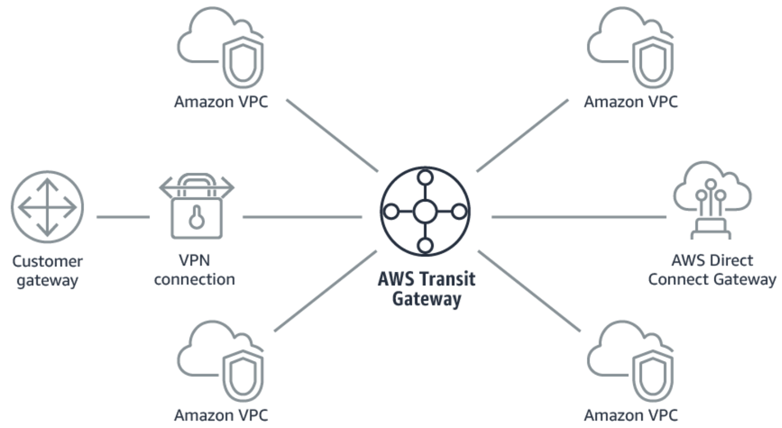


Peering connections facilitates joining 2 or more vpcs into a single network, and applications can communicate over its private IP.

Demo VPC Peering

1. VPCs can be in same account or different account
2. VPCs can be in same region or different region
3. CIDRs should not overlap
4. Create VPC One
   a. CIDR for VPC 172.16.0.0/16
   b. CIDR for subnet, 172.16.0.0/24
5. Create VPC Two
   a. CIDR for VPC 172.17.0.0/16
   b. CIDR for subnet, 172.17.0.0/24
6. Create VPC peering connection
   a. Create peering connection choose local vpc as vpc-one and target vpc as vpc-two
   b. Select the connection and approve it.
7. Make entries in route tables of vpc-one and vpc-two

Transit Gateway



AWS Transit Gateway connects VPCs and on-premises networks through a central hub. This simplifies your network and puts an end to complex peering relationships. It acts as a cloud router – each new connection is only made once.
Transit gateways has following additional capabilities over vpc peering

1. Connect with on-premise using vpn and direct connect
2. It simplifies the network with a single connection.

VPC Endpoints

The application running on ec2 is accessing S3 for uploading and downloading sensitive files to and from S3 over the internet.
By default communications happen through the internet.
But our banking customer has a policy that upload and download should not happen over the internet.

The solution for this is to make communications between ec2 and s3 over aws private network.
To solve this we should use vpc endpoint.

Securing VPCs

We can secure VPCs using security groups and Network Access Control List (NACL)

Security Groups

1. Security group is a virtual firewall that secures, EC2, RDS and lambda functions
2. Security group contains inbound and outbound rules
3. Inbound rules will control inbound traffic and outbound rules control outbound traffic

4. We open a port by adding a rule, for example we want to open 80 ports to the internet, we add an inbound rule for this.
5. If no rules are present in the security group then it means it allows nothing.
6. Instance can have upto 5 security groups
7. Security groups are stateful
8. Understanding security group sources
    a. My IP (Automatically pics your current system ip)
    b. Anywhere (Opens the traffic from all sources)
    c. Custom
        i. We can mention one or more CIDR
        ii. We can put custom ip addresses
        iii. We can put security group as a source
            1. We can put a security group as a source so that instance with this security group is allowed.

Network Access Control List (NACL)

1. It is also virtual firewall like security group
2. NACL acts at subnet level
3. There will be default NACL which is created while creating VPC.
4. Default NACL allows all inbound and outbound by default (Certification Exam)
5. All subnets are implicitly associated to default NACL
6. We can create custom NACL
7. Default rules of custom NACL is deny all inbound and outbound traffic
8. One subnet can have only NACL at a time.
9. NACL is stateless (faq)
    a. For inbound traffic it verifies both inbound & outbound rules
10. NACL supports explicit allow/deny
11. I want to block an ip, how?
    a. Do this in NACL
12. Understanding rule numbers in NACL
    a. Rules are executed from smallest rule numbers to biggest rule numbers (ascending order)
    b. If a matching rule is found it executes this rule without going to the next one.

(FAQ) What are the differences between security groups and NACL?

AWS Load Balancing

- Load balancer distributed incoming application traffic to backend servers(example, ec2 instances)

- Load balancer serves as a single point of contact for your applications.
- It is used for designing HA and scalability for our applications

(FAQ) What are the different types of load balancers supported by AWS?

As of today AWS supports 4 types of load balancers

1. Classic Load Balancer
2. Application Load Balancer
3. Network Load Balancer
4. Gateway Load Balancer

OSI layer reference

https://www.cloudflare.com/en-in/learning/ddos/glossary/open-systems-interconnection-model-osi/

Classic Load Balancer

1. Classic load balancers are considered previous generation load balancers, we should avoid using them.
2. This is layer 4(Transport) and layer 7(Application) type of load balancers.
3. Classic load balancer supports only ec2 backend
4. We can configure health checks, so that the traffic is routed only to healthy EC2 instances.
5. We can configure SSL certificates for https protocol, it also supports SSL termination.
6. The process of decrypting encrypted requests and closing https connection is called SSL termination.
7. It is good practice to leave SSL termination activity to the load balancer rather than doing it on an EC2 instance.
    a. For creating SSL certificates, AWS has native service called ACM (Amazon Certificate Manager)
8. It can load balance ec2 instances in a single VPC (single region)
9. We can secure load balancer using security groups
10. It supports two schemes
    a. Internal load balancer
    b. External load balancer (internet facing)

Classic Load Balancer Demo

1. For this demo let's take two ec2 instances in different AZs
2. Install apache servers on both, and put a simple html file
3. Create classic load balancer

Connection Draining

Connection Draining allows existing requests to complete before the load balancer shifts traffic away from a deregistered or unhealthy back-end instance
Default value is 300 seconds

Idle Timeout

The amount of time taken by the backend server(ec2) to respond to loadbalancer.
Default value is 60 seconds, sometimes we have to increase this value.

Stickeyness

By default it is disabled, if enabled a request from an user always sticks to the same server(ec2 instance).

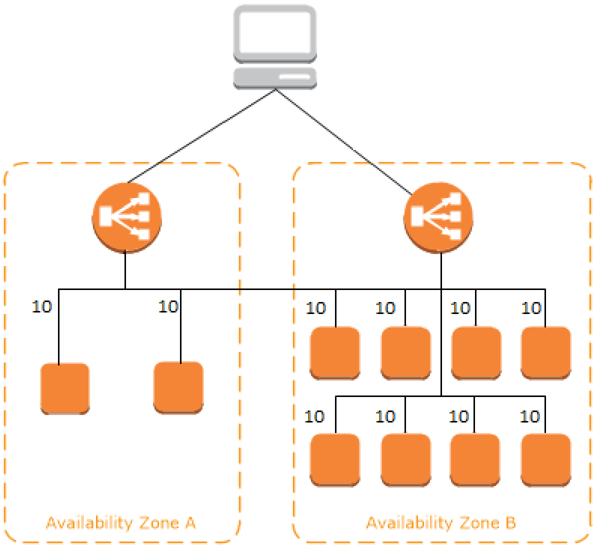
Load Balancer Access Logs

By enabling we can store access logs, the details about the client, ip address, timestamp location into S3 bucket.

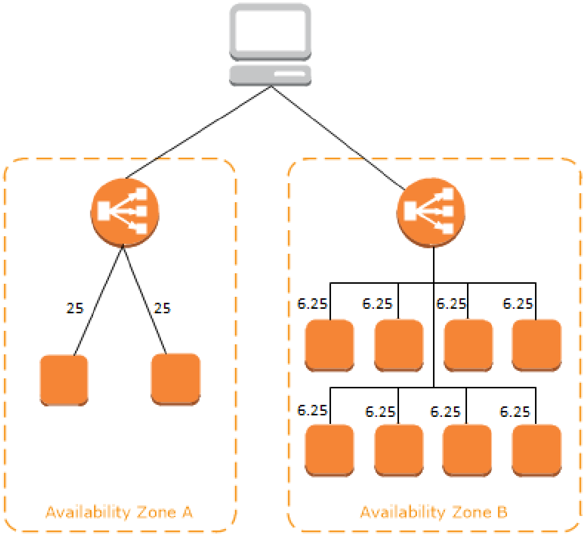
Cross Zone Load Balancing

By default every ec2 instance gets equal amount or load, but if cross zone load balancer is enabled. Each AZ gets an equal amount of load.

With Cross Zone not enabled



With cross zone enabled



AWS WAF(Web Application Firewall)

WAF protects web applications from common exploits like

1. DOS attacks (Denial of Service)
2. DDOS (Distributed Denial of Service)
3. Cross site scripting
4. SQL Injections
5. etc..

Application Load Balancer

1. It can be integrated with WAF
2. It can be integrated with global accelerator
3. It is designed for microservices architecture, where it enables routing based on path, port, query string, etc…
4. It supports load balancing ec2, on-premises, lambda functions

Demo - Application Load Balancer

Target Groups

Is a logical group of servers, for example we got 2 ec2 instances running microservice one, we put them into one target group.
For each micro service we create a separate target group.

Create target Group

Create ALB

Network Load Balancer (NLB)

1. It works at layer 4 (transport layer)
2. It supports only TCP/UDP
3. It can handle millions of requests per second.
4. Supports static ip address per subnet
5. It supports load balancing containers
6. NLB supports long lived connection which is ideal for websockets and gaming applications.

NLB Demo

https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-network-load-balancer.html

AWS Auto Scaling

Scalability is the ability of an application to handle a growing number of requests.
There are two types scaling

1. Vertical scaling
   a. Without adding additional servers, we're gonna increase CPU & Memory of existing servers.
   b. Vertical scaling has limitations, at some point vertical scaling stops scaling
2. Horizontal Scaling
   a. Adding additional servers to the existing servers is called horizontal scaling.
   b. It does not have limitations.
3. AWS auto scaling performs horizontal scaling

Benefits of autoscaling

1. The capacity is adjusted automatically without manual intervention.
2. It is cost effective, it launches instances when needed and terminates instances when not needed.
3. You don’t have to predict the load and setup capacity accordingly
4. It's recommended for all kinds of applications, because of the above benefits and there is no additional cost.

Autoscaling Components

1. Auto Scaling group
    a. This is a logical group of ec2 instances participating in autoscaling.
    b. The group will have minimum size, maximum size and desired size
    c. Desired size changes instantly at runtime depending on the load
2. Launch Template / Launch Configuration
    a. It is a template containing following details used by auto scaling to launch ec2 instances
        i. Instance type
        ii. Security group
        iii. IAM role
        iv. Key pair
        v. volume
3. Auto scaling policies
    a. The policies which determines when to add and when to remove instances
4. Auto Scaling can be integrated with load balancers

Auto Scaling Demo

1. Have a load balancer.
2. Create custom AMI with your application deployed on it.
3. Create launch configuration using above AMI
4. Select above launch configuration and create autoscaling group

How do you deploy new version of application into autoscaling

1. Create new AMI with latest code on it
2. Create new launch configuration
3. Update launch configuration in auto scaling group
4. Refresh auto scaling which replaces instances having older versions with newer versions.

(FAQ) In auto scaling is there a way to protect instances from termination?

Yes, We can do this by adding scale-in protection

(FAQ)What are different types of scaling policies supported by autoscaling?

1. Target tracking policy
2. Simple scaling policy
3. Step scaling policy

4. Scheduled scaling
    a. You may have applications with predictive load, let's say for example your application becomes busy on monday and it keeps busy till wednesday, and from thursday onwards the load decreases.
    b. For example as per above schedule we wanna set different min,max and desired values.
    c. You can have this one time or recurring schedule.
5. Predictive scaling policies
    a. This uses historical data to predict the load of applications and scale accordingly

Health Check Grace Period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

Auto Scaling cooldown period

Change the amount of time after a simple scaling activity completes before another scaling activity initiated by simple scaling policies can start. You can configure the length of time based on your instance startup time or other application needs. Cooldown periods help to prevent Amazon EC2 Auto Scaling from launching or terminating additional instances before the effects of previous activities are visible.

(FAQ) When instances are launched in auto scaling I want to install a software, how?

Use lifecycle hooks and perform the above operation

(FAQ) We are using spot instances in auto scaling, before spot interruption, we wanna copy log files in ec2 into S3.

Use lifecycle hooks

(FAQ) AWS EC2 Userdata

Using userdata we can run custom scripts on ec2 instances at launch time. This feature is useful to install packages on ec2 instances. However for advanced use cases consider using ansible for AWS SSM.
Reference to cloud-init directives
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html#user-data-cloud-init

1. You can run *bash scripts* or *cloud-init* directives on linux servers
2. You can run *powershell* scripts on windows instances

Debugging Userdata scripts

```
/var/log/cloud-init-output.log
```

(faq)What is ec2 instance metadata?

This provides details about ec2 instance itself, it provides information like ami-id, instance-type, instance-id, etc.

curl http://169.254.169.254/latest/meta-data/
curl http://169.254.169.254/latest/meta-data/ami-id
curl http://169.254.169.254/latest/meta-data/hostname
curl http://169.254.169.254/latest/meta-data/instance-id
curl http://169.254.169.254/latest/meta-data/instance-type
curl http://169.254.169.254/latest/meta-data/public-keys/

What are placement groups in AWS

IAM (Identity and Access Management)

- IAM is used for managing access to users and services
- For example java code running on EC2 wants to access S3 bucket, this is configured using the IAM role.
- Federating with third party identity providers like Microsoft AD.
- Integrating with web or mobile applications, line you application presents login with (facebook, gmail, amazon, etc)
- Managing cross account, Users in one account can access resources in another account, multi aws account is common pattern now a days, there will be accounts like development, staging, production accounts (FAQ)
- We can have an MFA.

AWS IAM Root user
- The email id we used to create aws account is root user
- Don’t use this for day to day activities, create identical admin user in IAM and use it
- Enabling MFA on root and other users is good practice and provides an additional layer of security.

AWS IAM User
- The user created from IAM dashboard

IAM Programmatic Access

This helps use to configure access to programs(ex: java,python,nodejs,aws cli, terraform)

Demo configure programmatic access

Create an IAM user with programmatic access and save keys in a safe place.

Install and Configure AWS CLI with above credentials

https://aws.amazon.com/cli/
Configure CLI with access keys and secret keys
aws configure
Provide access key id, secret access key, region and default output format.

Note: Now on local we can run any code like python, java, nodejs ects all of them are designed by default to use above configured credentials.

Location of access key and secret key

<USER_HOME>/.aws
For example → /Users/kammana/.aws

Can we maintain multiple credentials for multiple accounts

aws configure –profile dev

Problems with secret key and access key

We have to safely maintain them and periodically rotate them.

Running CLI and Python on EC2

We can use iam user with programmatic access but not recommended so we will use IAM role.

IAM Policy

IAM policy is a JSON object which represents set of operations an identity(role, user) can perform

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
    ]
}
```

Create IAM policy to allow describe, start and stop operations on EC2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

We can have condition in a policy

For example the above policy should be allowed only from specific IP range

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "172.16.0.0/22"
                }
            }
        }
    ]
}
```

Configure MFA for IAM user

IAM Cross Account Access

1. This allows users in one AWS account to manage resources in another AWS account.
2. In modern infrastructure patterns and architecture there are so many use cases for cross account access.
3. I want to stop ec2 instances in another aws account with credentials in the current account.
4. ***Note: cross account access will not work for root users***
5. For this Demo
    a. In account one create IAM user

AWS IAM account settings

Using this you can have your own password policy to manage aws account access.

AWS S3 (Simple Storage Service)

1. Amazon Simple Storage Service (Amazon S3) is an *object storage* service that offers industry-leading scalability, data availability, security, and performance
2. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as
    a. Data lakes
    b. Websites
    c. Mobile applications
    d. Backup and restore
    e. Archive, enterprise applications
    f. IoT devices
    g. Big data analytics.
3. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

Few example for object based storage

1. Google Drive
2. Dropbox

3. Icloud
4. Youtube
5. Facebook uses object based storage to store images and videos

S3 Characteristics

1. It is region specific
2. S3 maintains upto 3 replicas across multiple availability zones.
3. It offers upto (eleven nines) 99.999999999% of durability.
4. It is internet based service
5. We can directly upload or download objects, it is not a sub service like EBS is a sub service of EC2.
6. It can store unlimited amount of data
7. Each object can’t be more than 5TB

S3 Bucket operations using CLI

1. Create Bucket
    a. aws s3 mb s3://javahome-mybucket-2022
    b. Bucket name must be unique across all aws accounts
2. Upload files
    a. aws s3 cp pods.yml s3://javahome-mybucket-2022
3. Downloading File
    a. aws s3 cp s3://javahome-mybucket-2022/pods.yml hari.yml

S3 Storage Classes (Certification & Interviews)

- By choosing right storage class we can reduce cost for S3
- S3 offers following storage classes
    - Standard Storage
        - It is used for frequently accessed objects
        - It maintains three replicas of each object across multiple AZs
        - Standard Storage is default storage class
        - It offers 99.999999999 durability
        - It offers 99.99 availability
    - Standard IA (Infrequently Accessed)
        - It is used for infrequently accessed objects, cost is lower than Standard
        - It maintains three replicas of each object across multiple AZs
        - It offers 99.999999999 durability
        - It offers 99.99 availability
    - Reduced Redundancy
        - It is for frequently accessed objects
        - It maintains only 2 replicas

        - It offers 99.99 durability
        - It offers 99.99 availability
    - Intelligent Tiering
        - Sometimes we can’t predict access patterns and in such cases intelligent tiering does it for you, based on access patterns it will move them into the right storage class.
    - One Zone IA (Infrequently Accessed)
        - It stores data in single Zone and cost is less
        - It is to maintain a secondary copy to meet compliance.
    - Glacier
        - The Amazon S3 Glacier storage classes are purpose-built for data archiving, providing you with the highest performance, most retrieval flexibility, and the lowest cost archive storage in the cloud.
        - All S3 Glacier storage classes provide virtually unlimited scalability and are designed for 99.999999999% (11 nines) of data durability.
        - The S3 Glacier storage classes deliver options for the fastest access to your archive data and the lowest-cost archive storage in the cloud.
        - Retrieving data from glacier take 5-12 hours
    - Glacier Deep Archive
        - It is cheaper than Glacier.
        - Data retrieval time is about 12 hours.

Glacier Note: moving multiple small files into the glacier will cost you more.

S3 Versioning

1. If versioning is enabled and we are uploading the same file again and again maintains multiple versions of the same file.
2. It is useful to keep track of changes happening to a file.
3. With versioning you can recover more easily from both unintended user actions and application failures. (certification)
4. After enabling versioning we can’t disable but we can suspend.

S3 Default Encryption (Interview)

1. For example we are working for a pharmaceutical company, they have lots of patient data stored on S3 and as per government policy the data at rest needs to be encrypted.
2. S3 supports encryption at rest. But by default this option is disabled.

S3 Event Notifications

1. We can perform custom operations for specific events
2. For example we want to automatically process JSON coming to S3 and insert that data into mysql DB.

3. If a file is deleted we want email alert

S3 Server Access Logs

Server access logging provides detailed records for the requests that are made to a bucket

S3 Static Website Hosting (Certification)

1. S3 supports hosting static websites
2. Other options to host static website is EC2, ECS, Elastic Beanstalk
3. S3 has following benefits for hosting static website
    a. No servers to manage
    b. No hourly charge for S3
    c. No need to architect for scalability
    d. No need to architect for high availability

Steps to host static website on S3

1. Enable static website hosting under properties
2. Under permissions disable block public access
3. Under bucker policy add following JSON

```
a. {
b.     "Version": "2012-10-17",
c.     "Statement": [
d.         {
e.             "Sid": "PublicReadGetObject",
f.             "Effect": "Allow",
g.             "Principal": "*",
h.             "Action": "s3:GetObject",
i.             "Resource": "arn:aws:s3:::javahome-mybucket-2022/*"
j.         }
k.     ]
l. }
```

S3 Object Lock (Certification)

1. With S3 Object Lock, you can store objects using a write-once-read-many (WORM) model.

S3 Replication

1. There are several use cases for S3 replication, for example there is a government policy which dictates organizations to maintain multiple copies of the same data at specific distances.

2. We may be running global applications, these applications use S3 for storing and retrieving data. Right now s3 bucket is in a specific location (ex: mumbai) application in the USA takes more time to put and retrieve data because of network distance, to optimize this situation, we can use S3 replication and maintain data local to the applications and improve performance.
3. It supports buckets in same and different region
4. It supports buckets in same and different account
5. Versioning needs to be enabled for bucket replication

(FAQ)S3 Lifecycle Rule (Certification)

Using lifecycle rules we can automate transitioning of objects to low storage class and eventually we can expire them.
For example we have date in S3 and we want to set following rule

1. Create/Upload objects to Standard Storage
2. Objects with age 30 days move to Standard IA
3. Objects with age 90 days move to glacier
4. Objects with age 365 days move to glacier deep archive
5. Objects with age 5 years expire.

S3 Pre Signed URL

With S3 presigned URL we can temporarily grant access on objects to anonymous users.

AWS Databases

AWS offers a wide variety of database options, in aws databases are managed services.

RDS (Relational DataBase)

Relational DataBases have specific characteristics.

1. Static schema
2. Supports joins
3. They stop scaling at certain point

AWS RDS supported engines

1. Oracle
2. My SQL
3. Microsoft SQL
4. Amazon Aurora
5. PostgreSQL
6. MariaDB

RDS Demo Launch MySql server

RDS Multi AZ deployments (FAQ)

1. RDS will set up 2 db instances in two AZs one as a primary and second one as a standby.
2. RDS will take care of synching data from primary to standby automatically and synchronously.
3. If primary fails it will automatically failover to standby, the changes are not needed in the application when failure occurs, because endpoint will not change.
4. Multi AZ is meant for high availability but not for scalability

RDS Read Replicas

1. Read replica is additional database instance which supports only select queries.
2. Data is replicated on to RR asynchronously.
3. RR instance can be is same region or different region
4. We can leverage RR for setting up disaster recovery.
5. RR is meant for performance improvement.
6. If required you can promote RR as a master, so it then supports both insert and select.
7. The use case of using RR is there might me single instance and multiple applications connecting to it and causing performance issues, to fix this issue, RR is one solution

RDS Backups

- Backups are performed once in a day, we can choose the backup window of our choice.
- While backup is performed there will be brief suspension of IO operations which causes performance issues, however if there are multi AZ configured, backups are performed on standby instances, it will not cause performance issues.
- RDS supports both manual backups and automated backups.
- Default retention is 7 days, maximum retention is 35 days and minimum retention is 0 days(no backup).
- We can copy snapshots from one region to another region.
- Backups are kept in S3.
- It supports point in time recovery.

RDS Event Notifications and subscriptions

We can create event subscriptions for

1. Low storage
2. Failure

3. Failover
4. Etc..

RDS Logs

Note: RDS logs can be streamed into cloudwatch logs

Amazon Aurora

1. It is a proprietary database from Amazon.
2. It offers 5 time throughput of mysql and 3 times throughput of postgresql
3. It is cheaper than other popular databases.
4. It offer six way replications, every data is replicated six time on 3 AZs, it is highly durable
5. It has self healing mechanism, any of the replica is corrupted it self heals
6. It supports upto 15 read replicas with auto scaling.
7. Its disk can automatically scale upto 128 TB
8. Because of its clustered architecture it performs well
9. Aurora offer global database, means it can be distributed across regions so that global applications will have data in their local regions
10. Aurora supports serverless.

AWS Redshift

- It is relational DB
- It does not support read replicas
- It is data warehouse
- It architecture is clustered (master and worker nodes)
- It uses massive parallel processing which improves query performance
- Redshift has Cross-Region Snapshots, to automatically copy snapshots to cross regions.

Elastic Cache

1. It is distributed in-memory database
2. In-memory stands for (RAM)
3. In-memory databases drastically improves database performances
4. Elasticache supports two engines
    a. Redis
    b. Memcached
5. One use case, it can be used for storing user session tokens.
6. DynamoDB also can be used for above use case

DynamoDB (Important for Certification Exam)

1. Fast flexible, NoSQL database for single digit millisecond performance at any scale.
2. It can handle any amount of data
3. Good for
    a. Gaming
    b. Mobile applications
    c. Web applications
    d. Etc.
4. Global tables are supported by dynamodb.

AWS Simple Services

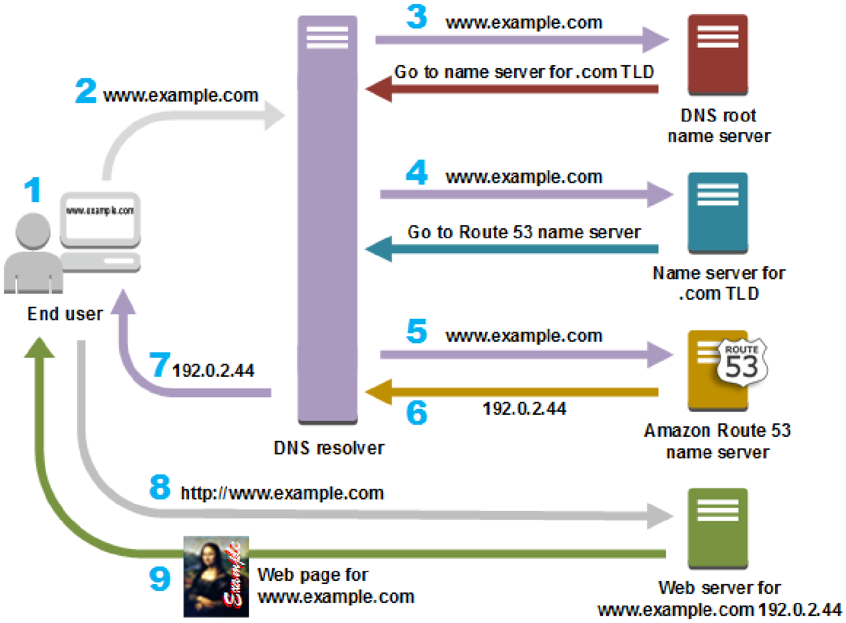
1. Simple Notification Service (SNS)
    a. SNS follows publish and subscribe pattern
    b. For using SNS, first we should create SNS topic
    c. And add subscribers to the topic
    d. And publish messages to the topic
2. Simple Email Service (SES)
    a. This service is used for sending and receiving emails, from custom sender and recipient.
    b. It is useful for dealing with transactional messages.
    c. Send emails about orders placed, send emails about OTP for transactions
    d. etc.
3. Simple Queue Service (SQS)
    a. Queues are used for integrating applications asynchronously
    b. Short polling and long polling
    c. https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-short-and-long-polling.html

Python 60 minutes Video

https://youtu.be/XokLzvcB4pY

Route 53

1. Route 53 is called DNS server
    a. The job of DNS server is to provide mapping between human readable hostnames with ip addresses.
    b. DNS is like an internet phonebook.
    c. How DNS resolutions (IQ)



2. We can register domains with route 53
3. Can configure health checks

Route 53 Hosted Zone

Hosted zone is collection of records specific to a domain

(FAQ) What are different types of records in Route 53

1. A record
    a. Used for IPV4
2. AAAA Record
    a. Used for IPV6
3. CNAME
    a. Used for mapping a domain to another domain

(FAQ) What is difference between CNAME and alias

Alias is built in feature of route 53, and it is used for mapping a domain with aws native endpoints like load balancers, S3, cloudfront, elasticbeanstalk, api gateways etc..
The biggest difference between CNAME and alias is

- Alias queries are free of cost
- CNAME incurs additional cost

Route53 routing policies

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html

- Simple routing policy
- Weighted routing policy
- Failover routing policy
- Geolocation routing policy

TTL in Route 53

Time To Live, is the amount of time records are cached in the browser or DNS resolver

CloudFront (CDN)

1. CloudFront is a network of cache servers, we call them edge locations.
2. Edge locations are available in all regions.
3. It helps in caching the responses close to customers and improves application performance.

Origin Access Identity (OAI)

It is a special identity which allows only cloudfront to access s3 bucket

AWS Cloud Watch

1. It is a monitoring tool in AWS which monitors resources(ec2,s3,ebs, etc) and applications (java,python, .net, etc).
2. It is also used for centralized logging
3. We can configure alarms
4. We can create dashboards
5. We can configure alerts on exceptions in log files

(FAQ) How do you monitor memory of EC2?

This metric is not directly available, we have to install cloudwatch agent on EC2 to collect memory metrics.
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html

(FAQ) What is the difference between cloudwatch and cloud trail?

CloudWatch monitors resources and applications, and cloud trail monitors user and api activity.

Cloud Trail

1. Cloud trail monitors user and api activity
2. By default cloud trails maintain 90 days of management events.
3. If we need events history more than 90 days and we want them in S3 bucket to perform user and security auditing we have to create a trail.
4. Creating a trail incurs charges.

KMS (Key Management Service)

KMS is used for encryption and decryption.
The idea is to protect data from bad guys, for example EBS volume is encrypted and this disk is held by a hacker, he may not be able to see actual data, because he can’t decrypt it.

AWS Lambda, Python & Boto3

1. Lambda is serverless compute services
2. Run the code without thinking about servers
3. We are charged for execution time and not for idle time.
4. Lambda functions reduces cost a lot
5. Lambda functions can be developed using any language but popular languages used are
   a. Python
   b. NodeJs
   c. It however supports other languages
      i. Java
      ii. .Net
      iii. Ruby
      iv. Go lang
      v. Etc..
   d. It also supports custom runtime

Boto3 & Python

Note: AWS has provided SDKs (Software Development Kits) using which we can interact with AWS
Boto3 is the module for python for programmatically interacting with AWS.

Setting UP Python and Boto3 on local laptop

1. *Make sure AWS CLI is installed and configured with access keys and secret keys*
2. Install python
   a. https://www.python.org/downloads/

3. Install Boto3

Write and schedule lambda function to delete unused volumes

1. Create lambda function
2. Schedule Lambda Function

```
import boto3

client = boto3.client('ec2')
sns_client = boto3.client('sns')
def lambda_handler(event, context):
    # TODO implement, deleting unused volumes should be here
    response = client.describe_volumes(
        Filters=[
            {
                'Name': 'status',
                'Values': [
                    'available',
                ]
            },
        ]
    )

    for volume in response['Volumes']:
        print(volume['VolumeId'])
        client.delete_volume(
            VolumeId=volume['VolumeId']
        )
    # Send email notification using SNS
    response = sns_client.publish(
        TopicArn='arn:aws:sns:ap-south-1:553410407942:javahome-app',
        Message='Unused volumes are deleted',
        Subject='Volumes Deleted'

    )
```

(Task)Find and send email about unencrypted EBS volumes

(Task) When a file is uploaded into S3 bucket, get its metadata and insert into DynamoDB

Meta Information Required

1. Message ID(NA)
2. Created By (Event)
3. Creation Timestamp (get_object_attributes)
4. Completed time stamp (NA)

(TASK) Fetch ECS cluster details and put it in CSV file and upload to S3 bucket

(TASK) Refresh specified tables in production database into test environment

1. Take RDS snapshot (backup)
2. Restore RDS instance
3. Perform backup of specified tables
4. Restore those tables from the DB in step 2, and refresh in dev/test database

Lambda Runtime Settings

1. Runtime
    a. Like python, java, golang
2. Handler
    a. <file-name>.<method-name>
    b. Lambda service looks for a python file with file-name

(FAQ) What is the lambda layer?

It is a zip of external dependencies, used by lambda functions.

Lambda Memory & Timeout

- Lambda can run maximum 15 minutes
- Default timeout is 3 seconds
- Default memory is 128MB, it can be increased upto 10GB
- Increasing memory proportionally increases CPU

AWS Lambda VPC configuration

1. Bydefault lambda functions run in its own dedicated VPC
2. However we can configure lambda to run within our VPC so that for example lambda wants to connect with RDS in our VPC.

(FAQ) What is a cold and warm start in Lambda?

When a lambda function is triggered

1. Download the code
2. Create runtime environment
3. Execute lambda code

Cold Start Is nothing but *downloading the code and creating runtime* for executing the code.
Warm start is nothing but *executing lambda code on pre created* runtime

AWS Lambda Provisioned Concurrency

Provisioned concurrency allows lambda functions to maintain precreated runtime environment to avoid latency issues due to cold start.

AWS Lambda can be integrated with EFS

This feature was not there before, maybe it was introduced 2-3 years ago.

(FAQ) What is the maximum size of a lambda file system?

512 MB

Lambda Function Environment Variables

Environment variables are used to pass arguments to applications
Note: We can use aws parameter store to store configuration data

Boto3 Lambda Youtube Videos

https://youtube.com/playlist?list=PLH1ul2iNXl7sxXBK6LkTf6McNGWrB_9wg

Terraform Deploy Lambda Function | Terraform IAM Role | AWS Terraform | Terraform La…

Build Test & Deploy Lambda functions
https://dev.to/hoangleitvn/how-to-build-test-and-deploy-lambda-function-to-aws-53cj