AWS Cloud Computing

What is cloud computing?

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

What are the benefits of cloud computing?

1. Cost
    a. You do not have to put up-front investment, and in the cloud you pay-as-you-go. This drastically reduces cost on IT infrastructure.
2. Agile
    a. Anything you get from the cloud takes a few minutes and clicks, for example We want an oracle database, we get it in a few clicks.
3. Maintenance
    a. You don’t have to do lots of maintenance, cloud providers take care of maintenance.
4. Security
    a. Initially customers felt cloud is not secure, but time has proved, cloud is more secure than their own data centers. Cloud offers lots of security controls to safeguard our applications and data.
    b. Security is shared responsibility model
5. Global Scale
    a. You may have business in one region today and you can scale your business globally.

What do we do here?

We are going to learn AWS cloud concepts like, compute, storage, databases, load balancing, DNS, etc.. and set up those services for our customers to deploy applications.

Cloud has different Streams

1. Architect
2. Administrator
3. Developer
4. Bigdata Developer

5. DevOps Engineer
6. ML/AI
7. IOT
8. Etc..

Cloud Deployment Models

1. Public Cloud (Very popular and widely used)
    a. If we are using AWS for cloud, and all infra is from AWS public cloud, then we say it is public cloud.
    b. We share certain resources with other tenants. Public cloud is cheaper than private cloud.
2. Private Cloud (Very rarely Used)
    a. Many cloud resources are dedicated to specific tenants. And they are costly.
    b. They are like their own data centers, but managed by cloud providers, and they get cloud benefits.
3. Hybrid Cloud.
    a. It is a combination of public, private and on-premise stuff.

Cloud Service Offerings

Cloud offers somany services, those services are offered in of of the following pattern

1. IaaS (Infrastructure as a Service)
    a. AWS offers virtual machines for example, we have to patch and do other maintenance.
2. PaaS (Platform as a Service)
    a. AWS offers the platform, for example “Elastic Beanstalk”, this service directly offers tomcat, docker, IIS, and other platforms, so we do not have to setup VM, do patching, etc..
3. SaaS (Software as a Service)
    a. The application itself is provided by the vendor, we consume it using a web browser or mobile app, for example gmail, facebook, linkedin, etc..

AWS Global Infrastructure

AWS maintains data centers across countries. AWS has 20 plus regions.

Region: Region means a geographical location, for example “mumbai” is aws region.
Region is usually a combination of 2 or more Availability Zones.
Availability Zone: Each region is compound of two or more Availability Zones, AZs are meant for HA(High Availability)
Edge Locations: I will explain this in topic

Sign Up for AWS free accounts

https://aws.amazon.com/free

1. PAN card is optional (you could say no PAN)
2. For Support (leave default option (no charge))
3. This account is free for 12 months

AWS EC2 (Elastic Cloud Compute)

1. EC2 is nothing but a virtual server in aws cloud
2. EC2 offers a wide variety of operating systems like Linux, Ubuntu, Solaris, Windows etc.

Launch EC2 Instance

Launch Linux virtual server with 1 CPU, 1GB Memory and 10 GB HDD.

1. Choose AMI (Amazon Machine Image) a template which contains OS, and other pre-installed softwares.
2. Choose instance type, this offers, CPU, Memory(RAM), and network performance
3. Networking details (Go with default values and will discuss more later)
4. Storage, The hard disk for EC2.
5. Security group, this is a firewall for ec2 instances.
6. Tags, add one or more tags
7. Launch instance. By choosing a keypair.

Connecting to Linux EC2 Instance

1. For connecting with remote linux servers we use SSH(Secure SHell).

Connecting From Windows

1. Install ssh client
    a. Putty
    b. MobaXterm
2. Install MobaXterm
    a. And connect

Connecting From Mac OS

1. Open terminal
2. chmod 400 ~/Downloads/hari-naveen.pem(one time)
3. ssh -i ~/Downloads/hari-naveen.pem ec2-user@13.232.26.125

Exercises

1. Launch RedHat Linux 8
2. With 1 CPU, 1GB Memory and 15 GD HDD

3. Connect with MobaXterm
4. Create empty file with your name
5. List the file
6. Terminate EC2 instance.

Launch Windows EC2 Instance

Connecting to Remote Windows VMs

Connecting from Windows

1. For connecting to remote windows servers we use RDP (Remote Desktop Protocol)
2. RDP is built into windows (you do not have to install explicitly)

Connecting to Windows EC2 from Mac OS

1. Install RDP client on mac os (Microsoft Remote Desktop)
2. Gather Public IP, Username and Password

Connecting to Windows EC2 from Windows OS

Launch Linux EC2 and Install a small website on it

1. Launch Amazon Linux EC2 instance
2. Install Apache web server
    a. Apache is lightweight web server for running web applications
    b. sudo yum install httpd -y
3. Host a dummy website on apache web server
    a. Inorder to host a website on apache, you have to place your code under /var/www/html/
    b. cd /var/www/html/
    c. sudo vi index.html
4. Start apache server
    a. sudo service httpd start
    b. sudo chkconfig http on

Access Above application from the Browser

http://public-ip

Exercises

1. Launch Windows EC2 instance, connect using RDP and later terminate it
2. Launch Amazon Linux Instance, Install apache and host a dummy website, and access it from the browser.

3. Terminate EC2 instance.

Resizing EC2 instances

- To the same EC2 instance we want to add/remove CPU and Memory.
- Resize requires a stopping instance.
- CPU, Memory and network performances are associated with instance type.

AMI (Amazon Machine Images)

1. AMI is a template to launch an EC2 instance, The template contains OS.
2. There are AWS managed AMIs.
3. We can create custom AMIs of our choice.
4. One of the way to deploy applications in AWS is through custom AMI
5. For example we could have
    a. Jenkins AMI
    b. SonarQube AMI
    c. Our Application AMI
    d. Nexus AMI

Creating Custom AMI

My AMI should have apache installed with a custom application.

1. Launch EC2 instance
2. Install packages you want on your custom AMI
3. Create EC2 image
4. You can terminate the instance used for creating this template

Launch EC2 from Custom AMI

Launch Instances, click my AMIs, pick the AMI and create it.

- AMIs are region specific, that is AMIs present in Mumbai won’t be available in other regions.
- We have options to copy to cross regions.

AMI Permissions

1. By default AMIs are private, meaning it's available to authorised users in the same account.
2. We can share AMIs with a cross account.
3. We can make AMIs public, so that every aws account will have visibility.

Where AMIs are stored

AMIs are stored in AWS S3 (Simple Storage Service)

Exercises

1. Create a custom AMI with apache, when acced from the browser it should display “Hari’s Custom AMI”.
2. Copy AMI to singapore
3. Launch EC2 in singapore, access EC2 and test your application
4. Clean up, delete AMIs and EC2 instances.

EC2 Backup

Taking EC2 backup is nothing but creating AMI

EC2 Instance Types

AWS offers different instance types for meeting demands of different types of work loads

1. General Purpose
    a. General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories, etc..
    b. This is most commonly used for setting up dev/test environments.
    c. It is not intended for mission critical production workloads.
2. Compute Optimised
    a. Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.
3. Memory Optimised
    a. Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.
4. Storage Optimised
    a. 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.
5. GPU Optimised
    a. GPU means Graphics Processing Unit
    b. A GPU instance is recommended for most deep learning purposes. Training new models will be faster on a GPU instance than a CPU instance. You can scale

sub-linearly when you have multi-GPU instances or if you use distributed training across many instances with GPUs.

6. Etc…

AWS AMD Processors to reduce cost

It provides better performance with low cost.

Instance Family, Generation and Size

t2.micro

- t belongs to general purpose
- 2 is generation
- Micro is size

t3a.medium

- T belongs to general purpose
- 3 is generation
- Medium is size
- a is AMD processor

Burstable Performance Instances

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-performance-instances.html

EC2 Instance Purchase Options

1. On Demand Instances
    a. Use this when you do not have long term commitments, you may need instances temporarily for a few hours, days or weeks.
    b. On demand instances are billed per hour or per seconds,
    c. Cost is high for on-demand when compared with other purchase options.
    d. For example We are working on POC (Proof Of Concept), We need 3 EC2 instances for 5 days.
    e. Billing stops when EC2 stops
2. Reserved Instances
    a. We go for reservations if we have long term commitments, we can choose either 1 year or 3 years.
    b. In return we get significant discounts, you get upto 60% discounts
    c. After purchasing reserved instances we can't return it back to AWS
    d. You can sell your reserved instances in AWS marketplace
    e. Billing remains the same, no matter if you are running it or not.
    f. Reserved instances are not physical instances, if reservations match on-demand instances it falls under reserved billing

    g. Payment Options
        i. All upfront
        ii. Partial Upfront
        iii. No Upfront
3. Dedicated Host
    a. Here you will purchase a physical host, this host is totally dedicated to you, and all VMS will be yours.
    b. The benefit of Dedicated Host is you can Bring Your Own Licence (BYOL).
4. Scheduled Reserved Instances (Certification)
    a. For example, Your application may have scheduled batch jobs running daily for 4 four hours, this is the use case for Scheduled Reserved Instances.
5. Spot Requests
    a. Request unused EC2 instances, which can reduce your Amazon EC2 costs significantly.
    b. You get upto 90% discounts
    c. Spot Instances have spot interruption, meaning AWS can take your capacity at any time by giving 2 minutes notice.

(Certification) When spot interruption occurs by default EC2 is terminated, but we want to stop rather than terminating how?

Ans) Submit persistent spot requests and choos stop as a behavior.

(FAQ) When spot interruption occurs we want to copy log files on EC2 on to S3

We have to configure a cloud watch event, for that event we can choose a target to run a command on EC2 which copies logs to S3.

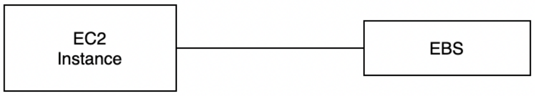
Exercises

1. Check what are savings plans in EC2
2. Check capacity reservations EC2

EC2 in free tier

1. 750 running hours in a month.
2. Only t2.micro instances are free under the free tier.
3. While choosing AMIs, make sure they are free tier eligible.

EBS(Elastic Block Store) Volumes



1. EBS is like a hard disk for ec2 instance.
2. EBS volumes are block level store
3. EBS offers different types of volumes designed for different use cases.
4. EBS volumes are persistent, that is data stored is permanent.
5. EBS volumes are highly durable, scalable
6. AWS manages redundant copies(2 copies) of the same data in the same AZ.
7. We can detach a volume from an instance and attach to different instances.
8. At any point of time EBS volumes can be resized(can increase but can't decrease)
9. Can we migrate EBS volume to another AZ or Region?
    a. Yeah, take a snapshot(EBS backup) and move it.
10. Backups of EBS volumes are called snapshots, backups are stored in S3(Simple Storage Service)
11. Root Volume & EBS Volume
    a. Root volume is the main volume where operating system is hosted
    b. EBS volume is the additional volume, where we run applications.
12. Can we encrypt root and data volumes?
    a. Yes
13. How do you attach new EBS volumes?(FAQ)
    https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html

    a. Create EBS Volume
    b. Attach volume to EC2
    c. Log on to EC2 and list volumes
        i. lsblk
    d. Create file system(don't create file system if you are restoring from snapshot)
        i. sudo mkfs.ext4 /dev/xvdf
    e. Create mount point
        i. sudo mkdir /app
    f. Mount the disk on above path
        i. sudo mount /dev/xvdf /app/
    g. Automount EBS volume when ec2 reboots.
        i. Make an entry in fstab file

EBS Volume Exercises

1. Attach additional volume, Configure auto mount for newly attached volumes
2. Check how to resize newly added volumes

EBS Volume Types

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

AWS has different EBS volume types designed for different types of workloads
We majorly see two types of volumes

1. Transactional workloads (SSD)
   a. You have web application and it has database
   b. This involves frequent read write with small IO size
2. Streaming Workloads (HDD)
   a. Optimized for large streaming workloads where the dominant performance attribute is throughput.
   b. Big Data analytics, data warehouse etc.
3. Previous Generation Disks
   a. Do not use this unless your applications perform well on this type of discks.

EBS Volume Types

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

SSD Type

1. General Purpose SSD
   a. It is used for dev/test environments.
   b. It is not designed for mission critical applications
   c. We can use it for small and medium databases in production
   d. We are charred for the size of disk
2. Provisioned IOPS SSD (Certification Exam)
   a. It is used for production workloads
   b. It is designed for mission critical applications
   c. It offers consistently high performance.
   d. We are charged for two things
      i. Size of the disk
      ii. For IOPS

HDD Type

1. Throughput Optimised HDD

    a. It is used for data warehousing, big data analytics etc.
    b. Use this for frequently accessed data
2. Cold HDD
    a. It is used for data warehousing, big data analytics etc.
    b. Use it for infrequently accessed data.

Certification Question)

We are choosing EBS volume for mission critical interactive web application, we need 16000 IOPS with single volume. Which volume type you recommend

1. General Purpose
2. Provisioned IOPS (Answer)
3. Throughput Optimised HDD
4. Cold HDD

We are choosing EBS volume for setting up databases for development and testing, choose wight volume type

1. General Purpose (Answer)
2. Provisioned IOPS
3. Throughput Optimised HDD
4. Cold HDD

(FAQ) Can we use HDD volumes as root volumes?

No, only SSD is used.

EBS Snapshots

1. Snapshots are EBS volume backups
2. Snapshots are stored in S3
3. Snapshots are useful to restore data if our disks fail.
4. We can copy snapshots to cross region
5. Snapshots are incremental, incremental means, the next backup will have changes made after previous backup.
6. Using snapshots we can migrate volumes from an AZ to another AZ

Exercises

1. Create snapshot
2. Copy the snapshot to different regions and create volume.
3. Create an ec2 instance and attach the above volume to this instance.
4. Delete snapshot, ec2.

AWS Instance Store (FAQ)

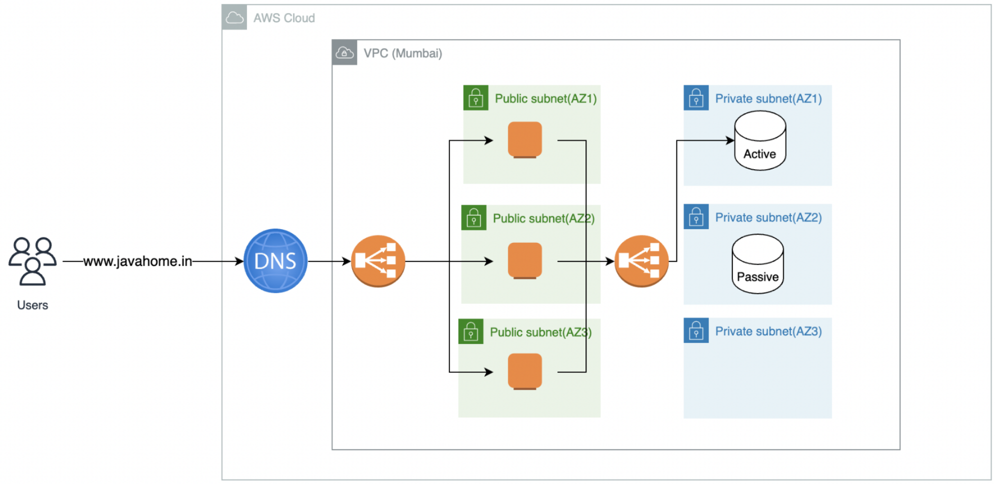
1. EC2 instances can be backed by EBS or Instance stores.
2. Instance store is available on the physical host of EC2 instance
3. Instance store is a temporary block level store, it is very cheap when compared to EBS volumes.
4. We lose instance store in following scenarios
   a. Underlying physical drive failed
   b. EC2 stopped
   c. EC2 terminated
   d. EC2 hibernated
5. We can't use it for stateful applications like databases
6. We can use it for cache, buffer and temporary data.

Launching EC2 with instance store

Exercises

1. Check what is lifecycle manager under EBS

AWS Networking (VPC)



- VPC stands for Virtual Private Cloud
- VPC is virtual data center in the cloud
- VPC isolates your aws resources in the cloud
- Using VPC we can set better security

- VPC and subnets are free

Create VPC

1. Choose arbitrary name
2. Choose following CIDR 10.0.0.0/24
3. Create VPC

CIDR Blocks

For creating VPCs and Subnets We should know about CIDR blocks.
CIDR (Classless Inter Domain Routing)

CIDR Explaination

10.0.0.0/24

- In the above CIDR notation, there are 4 octets, x.x.x.x/y
- Each octet is 8 bits
- With one octet we can represent up to 256 numbers (0-255)
- Value in each octet will be within (0-255)
- /24 is called netmask
- In the above CIDR total bits are 32(8+8+8+8)
- Each CIDR is always divided into 2, Network portion and Host portion
- /24 is netmask, means 24 bits are allocated to network and remaining 8 bits are used by hosts(IPs)
- With the above CIDR we get 2 to the power of 8 that is 256 ips.

100.0.0.0/22

- Netmask here is 22
- 10 bits are available for hosts, which means 2 to the power of 10 which gives 1024 IPs.
- We choose netmask depending on how many IPs we need.
- For example we want 2000 plus ips then roughly we need 11 host bits and netmask should be 21.

We can't directly place compute resources into VPC, we must use subnets.

Creating Subnets

1. Create VPC using 2 subnets, VPC should have 256 IPs and each subnet should have 128 IPs
   a. We should get 3 CIDRs
      i. Create VPC (10.0.0.0/24)

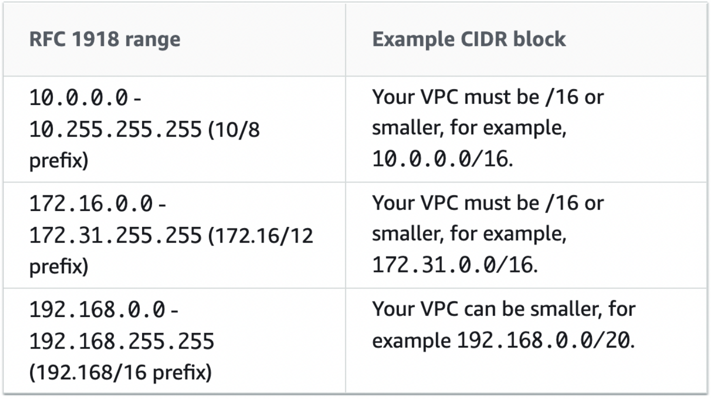
        ii. Create Subnet-1 (10.0.0.0/25)
        iii. Create Subnet-2 (10.0.0.128/25)

2. Create VPC using 2 subnets, VPC should have 512 IPs and each subnet should have 256 IPs

    1. CIDR for VPC (512)
        a. 192.168.0.0/23
    2. CIDR for subnet-1 (256)
        a. 192.168.0.0/24
    3. CIDR for subnet-2 (256)
        a. 192.168.0.0/24

CIDR and VPC best practices

1. Keep CIDR in private IP range



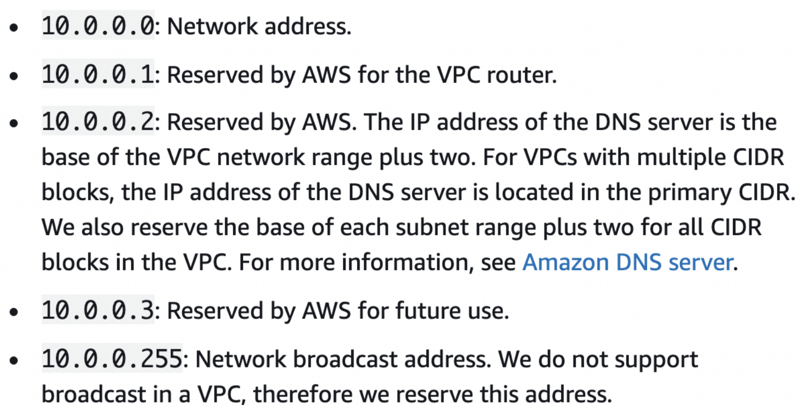
RFC 1918 range	Example CIDR block
`10.0.0.0` - `10.255.255.255` (10/8 prefix)	Your VPC must be /16 or smaller, for example, `10.0.0.0/16`.
`172.16.0.0` - `172.31.255.255` (172.16/12 prefix)	Your VPC must be /16 or smaller, for example, `172.31.0.0/16`.
`192.168.0.0` - `192.168.255.255` (192.168/16 prefix)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

2. Make sure your VPC CIDR is not overlapping with other VPCs and on premise CIDRs.
3. Always plan to have more space, because it will not be a problem to scale in future.

VPC CIDR Rule

It should be between /28 and /16 netmask

Every Subnet reserves 5 IPs



- `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.

Exercises

1. Create VPC with 4000 IPS and with 4 subnets
2. Create VPC with 4000 IPS and with 3 subnets
3. Create VPC with one public subnet
4. Launch EC2 instance into public subnet

(FAQ)What is a public subnet and private subnet?

Public Subnet:

- Public subnet is exposed to internet
- Means EC2 instances with public IP are accessible from the internet.
- We make a subnet public using internet gateway
- If EC2 in a public subnet is not having public IP, will it be able to connect to the internet and vice versa?

Private Subnet:

- Private subnet is one which is not exposed to the internet.
- We usually keep private applications like DBs, internal apps in a private subnet.

What is an internet gateway?

It is a virtual gateway which provides internet connection to VPC and subnets. It provides both inbound and outbound internet communications. One VPC can have only one internet gateway.

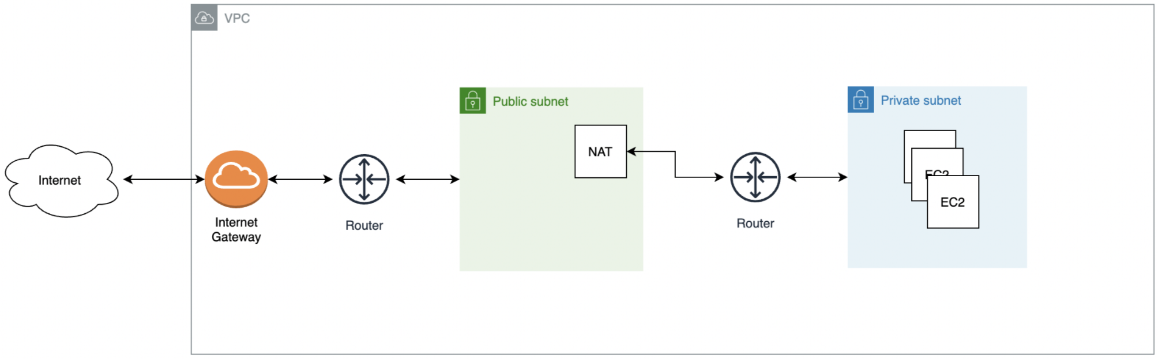
(FAQ) EC2 instances in private subnets want to download patches from the internet, how do you configure this?

Ans) Use NAT instances or NAT gateways.
Note: *In my project we are using NAT Gateways*

(FAQ) In which subnet you put your NAT

NAT should be put in a public subnet.

NAT (Network Address Translation)



1. NAT allows only outbound internet connections to ec2 in private subnets.
2. Ec2 instances in private subnets no need to have public IPs.

NAT Instance Demo

1. For this example make sure we have VPC with at least one public and one private
    a. VPC → 10.200.0.0/24
    b. Public Subnet → 10.200.0.0/25
    c. Private Subnet → 10.200.128.0/25
2. Create Internet Gateway
    a. Create internet gateway and attach to VPC
3. Add internet gateway to the default route table
    a. This configuration makes both subnets public
4. Create separate route table for private subnets
5. Attach private subnet to the above route table
6. Launch NAT Instance in public subnet with public IP using nat AMI
7. Open private route table and add route to NAT instance
8. Disable source destination check on NAT instance

Testing NAT Instance

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html#nat-test-configuration

1. Launch EC2 instance into private subnet
2. SSH into above instance and do ping google.com

NAT Gateway Demo

1. NAT gateway is not available under free tier, but still we can try it.
2. NAT Gateway pricing (In real time NAT gateway if preferred over NAT instance)
    a. Per hour charges
    b. Amount of data processed.

Differences Between NAT instance & NAT Gateway

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

Source Destination of NAT instances

1. AWS by default performs source destination checks on all EC2 instances by default

(FAQ) How are you connecting(SSH) to AWS from on-premises?

1. Jump Box or DMZ
    a. We put EC2 in public subnet
    b. Servers outside VPC will first ssh into above EC2 and then they can access other EC2 instances in VPC.
    c. It happens through internet
2. VPN
    a. ▶ How to Setup VPN in AWS? | Configure Open VPN in AWS | AWS VPN De…
    b. VPN uses internet, the speed also dependents on internet
    c. With VPN we can access ec2 instances over private IPs.
    d. VPC connections are secured.
3. Direct Connect
    a. This is expensive
    b. This is highly secured
    c. It's a very fast connection because it has a dedicated network between your office and aws.
    d. This is have use cases like migrate data securely

Route Tables

1. Route table will have bunch of routes
2. For example
    a. A route to other subnets
    b. A route to NAT
    c. A route to Internete
    d. A route to VPN
    e. Etc..

3. AWS creates a route table implicitly when you create VPC
4. Implicitly created one is called “Main” route table
5. Always by default all subnets are implicitly associated with main route table
6. We can create custom route tables
7. Each route table will have one default route, this route allows this subnet to communicate with all other subnets in same VPC
8. A subnet can have only one router, however same router can be associated with multiple subnets

Note: For IPV6 NAT gateways will not work, we must use Egress Only Internet Gateways

AWS Default VPC

- AWS accounts come with default VPC one per region
- In default VPC you will find only public subnets, one public subnet per AZ
- If no VPC is selected, AWS selects default VPC
- In default VPC subnet settings by default pics public IP

What is public IP?

Public ip is used for exposing EC2 instances to the internet, In AWS public ips are dynamic, that is if ec2 stops public ip is released, and if ec2 starts again it gets a new public ip.

What is EIP (Elastic IP)?

- EIP is a static public IP, it will never change, until you release it.
- We can detach from an instance and attach to another instance.
- One EIP is free in AWS account, send EIP onwards we are charged.
- Even the first EIP is charred if it is associated with stopped EC2.

What is private IP in VPC?

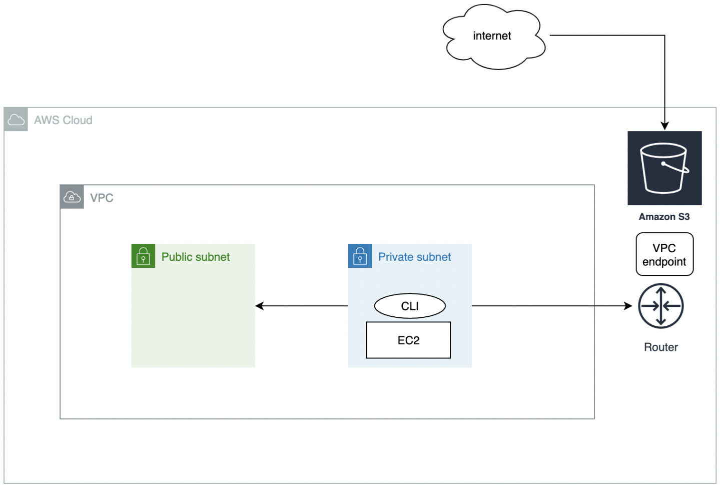
1. Private IPs are available within VPC
2. Private IPs are created using VPC CIDR

(FAQ) What is the VPC endpoint?

1. S3 is internet based storage, meaning we access S3 over the internet.
2. Sometimes your company policy dictates you should not access S3 over the internet.
3. By default uploading/downloading files to and from S3 is over the internet.
4. Using VPC Endpoint for S3 we can communicate to S3 within the AWS network.

VPC Endpoint Demo

https://youtu.be/FbJ1wPWoyxc

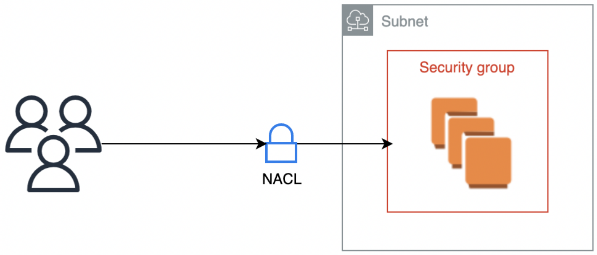


1. Create VPC with one Private and Public subnets
2. Launch EC2 in private subnet and see if you're able to access S3
3. Assign IAM role with S3 permissions to EC2 instance

Exercises

1. Check what is interface type endpoint

Securing VPC



1. We secure VPC using security groups and NACLs

Security Group

1. Security group is a firewall for EC2 instances
2. Security groups has inbound and outbound rules
3. Security groups are stateful (important)
   a. If there is inbound traffic to EC2, the security group checks only inbound rules even when traffic is returning back from ec2, this behavior is called stateful.

    b. If there is outbound traffic from EC2, the security group checks only outbound rules even when traffic is returning back from the target, this behavior is called stateful.
4. EC2 must at least have one security group
5. EC2, can maximum have 5 security groups per ENI
6. Security group does not have explicit allow or deny
7. Security Group sources
    a. My IP (It allows traffic only from your current system IP)
    b. Anywhere(It allows traffic from all ips)
    c. Custom

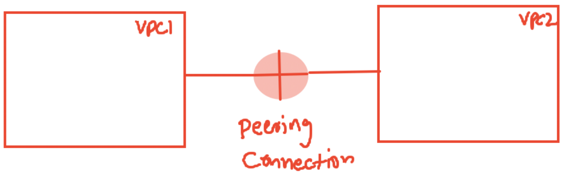
NACL (Network Access Control List)

1. It is a firewall at subnet
2. It has inbound and outbound rules
3. It has explicit allow or deny, that is we want to block IP, yes it could be done.
4. It is stateless, that is for inbound traffic it checks inbound and outbound rules both
5. One subnet can have one NACL and multiple subnets can have the same NACL.
6. Every VPC has default NACL
    a. Default NACL allows all inbound and outbound traffic
7. We can create custom NACL
    a. Default rules of custom NACL deny all inbound and outbound traffic.
8. NACL rule number
    a. Every rule has a number
    b. Rules are processed in ascending order (smallest to biggest)

(FAQ) Differences between security group and NACL

AWS WAF (Web Application Firewall)

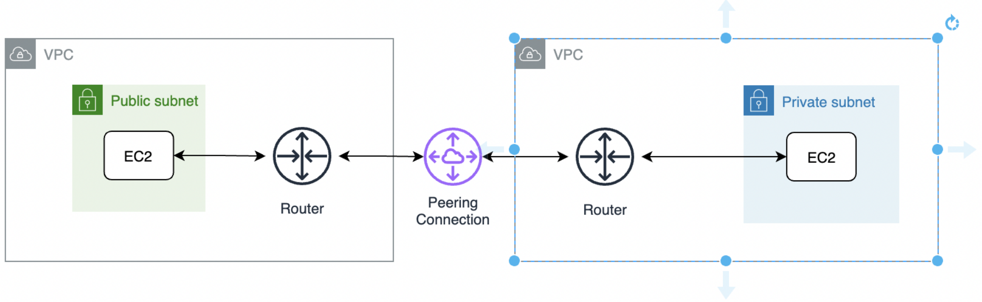
VPC Peering Connection (FAQ)



1. We use peering connections to connect VPCs
2. EC2 in two vpcs can connect with private IP
3. VPCs can be in same region or different region
4. VPCs can be in same aws account or different aws account

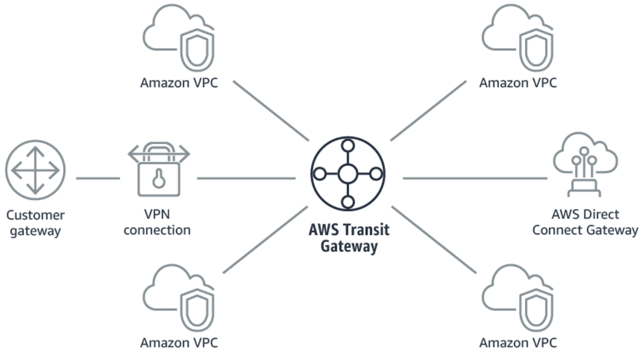
5. Does it support transitive peering, vpc-one peered with vpc-two, vpc-two peered with vpc-three, vpc-one and vpc-three cannot communicate

Peering Demo



1. Make Sure there are two VPCs
2. CIDRs should not overlap
3. Create peering connection
4. Get the connection accepted
5. Add routes in both the VPCs

(FAQ) AWS Transitive Gateway



1. We can think of this as an advanced version of vpc peering.
2. Using transitive gateways we can peer vpcs, vpns, direct connections etc.
3. This eliminates too many connections, with a single connection, we can join multiple VPCs, VPNs, Direct connects, etc.

Exercises

1. Create two VPCs and peer them
2. Create three VPCs and connect them with a transitive gateway.

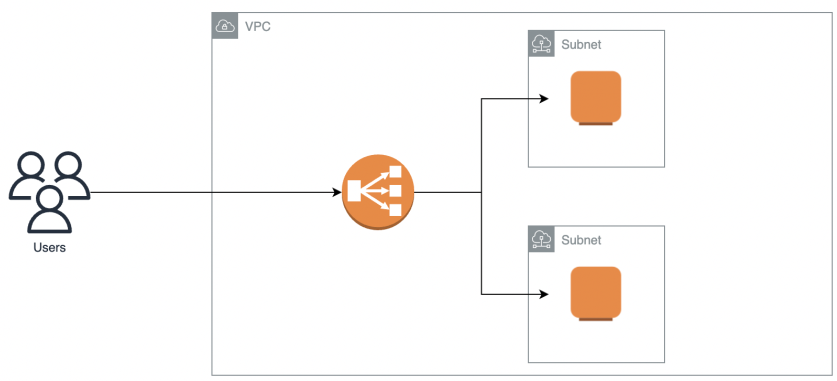
AWS Load Balancing

1. Load balancer acts like a single point of contact for backend ec2 instances.
2. Using load balancer we achieve HA for our applications
3. Using a load balancer we scale our applications.
4. AWS has its own load balancing solutions that are ELB, NLB, ALB and GLB.
5. There are popular third party load balancing solutions like nginx, Apache, Big IP, HA Proxy, etc…

ELB (Classic Load Balancer)

1. ELB acts as a single point of contact
2. Classic load balancers are called previous generation load balancers and prefer other options.
3. This supports two schemes
    a. Internet facing
    b. Internal
4. It has a health check feature, so that it routes traffic to only healthy instances.
5. If instances found unhealthy keep the instance out of service, *<u>it will not terminate.</u>*
6. ELB is vpc specific, that is one ELB can load balance only one VPC
7. ELB uses a round robin algorithm.
8. Classic load balancer works at OSI model layer 4 and 7

Classic Load Balancer Demo



1. We are going to use default VPC
2. We wanna launch 2 ec2 instances
3. On those instances install a web server and put a tiny website.
4. We will create ELB Classic

Connection Draining

For example we set connection draining value as 60 seconds, if we want to deregister or if ec2 is unhealthy it can’t be suddenly taken out of service, we should give some time for the instance to finish processing in-flight requests.

Stickiness

By default requests from a user are distributed to all ec2 instances in the loadbalancer. But with stickiness we can make a user request to stick to one instance.

Idle Timeout

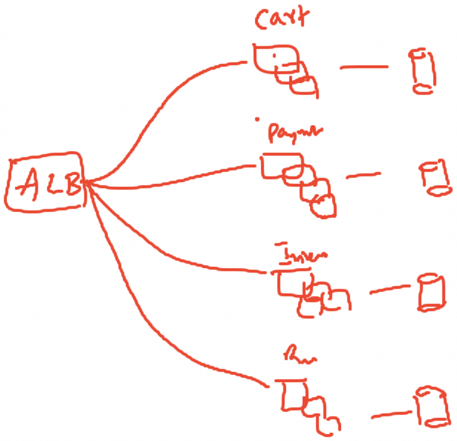
Idle Timeout is the number of seconds a connection can be idle before the load balancer closes the connection

Exercise

1) Setup classic load balancer.

Application Load Balancer (ALB) - (Certification)

1. Classic load balancer is previous generation load balancer
2. ALB is a layer 7 load balancer
3. It supports load balancing, EC2, external servers and lambda functions
4. It has support for path, port, query string based routing. This feature is very useful for microservices architecture
5. It has support for internal and external schemes
6. It also deals with health checks
7. ALB supports WAF (Web Application Firewall)



ALB Demo

1. We have to create a target group, the target group is nothing but a group of ec2 instances running specific applications.
2. Each target group represents one micro service

Exercises

1. What is NLB and setup NLB
2. Set up ALB with red and green target groups

FAQ) What are different types of load balancers?

FAQ) Differences between Classic and Application Load Balancer?

What is AWS WAF (Important for Certification)

1. It is a web application firewall
2. Protect RESTful APIs Using AWS Web Application Firewall (WAF) | AWS WAF Tut…

AWS Auto Scaling Group

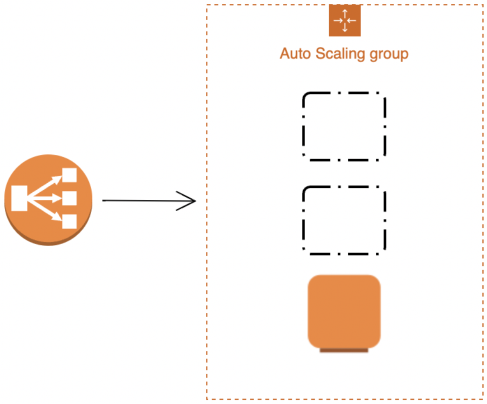
1. The idea of autoscaling is to add/remove instances on a need basis.
2. The capacity is dynamically changed by autoscaling, and it is cost effective.
3. Auto scaling itself is free, but autoscaling launches EC2 for that we have to pay.
4. It provides better fault tolerance, that is if an instance is unhealthy it terminates and launches a new one.
5. It provides better availability, that is it always maintains the exact capacity that is needed.
6. Autoscaling nicely integrates with load balancers

Autoscaling Components

1. Autoscaling Group
    a. The logical group of ec2 instances that is part of autoscaling
    b. Autoscaling contains a minimum size, number of instances will never go below this.
    c. It has a maximum size, that is the number of instances will never go above this.
    d. Desired size, we define this at the time of creating auto scaling or it can be dynamically updated by autoscaling based on load.

2. Launch Configuration
    a. Auto scaling uses launch configuration for launching ec2 instances
    b. This configuration contains following details
        i. AMI to use (custom AMI with our application pre configured)
        ii. Instance Type
        iii. EBS volumes
        iv. Security Groups
        v. IAM role
        vi. Key Name (pem file to use)

Auto Scaling Demo



1. Make Sure you have ELB
2. Create AMI with your application init.
3. Create Launch Configuration
4. Create auto scaling group

Auto Scaling cooldown period

1. In step wise scaling, if two consecutive scaling activities are triggered within a cooldown period, the second one is suppressed.

Auto Scaling instance Warmup

1. Instances launched in auto scaling can take some time for initialization and only then its metrics need to be counted in autoscaling.

(FAQ) When EC2 is launched/terminated in autoscaling group I want to perform custom operations

1. Install packages when ec2 is launched
2. Copy logs to S3 before ec2 is terminated.

Ans) Use auto scaling lifecycle hooks
https://docs.aws.amazon.com/autoscaling/ec2/userguide/tutorial-lifecycle-hook-lambda.html?icmpid=docs_ec2as_help_panel

(FAQ) How to protect an instance in auto calling from scale-in operation(remove)

Ans) Use scale in protection

Exercise

1. Check what is scheduled scaling
    a. https://docs.aws.amazon.com/autoscaling/ec2/userguide/schedule_time.html
2. Check what is predictive scaling
    a. https://aws.amazon.com/blogs/aws/new-predictive-scaling-for-ec2-powered-by-machine-learning/

Autoscaling update AMI(deploy new version of application)

https://youtu.be/c5Yst27jw4M

AWS Simple Services

1. Simple Notification Service (SNS)
    a. SNS is used for notifications, like CPU is high send alerts, Status Checks Failed, send alerts, etc..
    b. SNS Demo
        i. Create SNS topic
        ii. Subscribe EMAIL
        iii. Subscribe Phone
        iv. Test by clicking publish message
2. Simple Email Service (SES)
    a. This service is used for sending and receiving emails
    b. We can choose custom email as a sender email
    c. To send emails, we have to verify email or domain
3. Simple Queue Service (SQS)
    a. SQS is used for integrating applications
        i. For example application one wants to share data with application two

ii. Queues are used for asynchronous integration of applications

iii. Two types of queues

1. Standard
    a. It can process unlimited amount of messages
2. FIFO
    a. It can handle 300 messages per second or 3000 messages per second with its batch.
3. ***Long polling***
    a. It can wait upto 20 seconds and reduces the number of empty messages and reduces cost.

Cloud Watch

1. Cloud Watch is monitoring service in AWS
2. This service is more important if you are preparing for AWS DevOps Certification.
3. We need to briefly understand CloudWatch for our course.
4. We can monitor all AWS services like, EBS, EC2, Lambda, RDS, ELB, S3, etc..
5. We can use cloudwatch for on-premise applications, install cloudwatch agents and monitor them.
6. Using cloudwatch we can manage application logs
7. CloudWatch Events
    a. It supports two types of sources
        i. Schedule
            1. This takes cron expression
        ii. Event Source
            1. Any operation in AWS can be event source
            2. For example EC2 terminated can be an event
8. We can configure alarms and billing alerts

Monitoring EC2 Instance

From cloudwatch dashboard → metrics → all metrics → EC2

(FAQ) How do you monitor EC2 memory and disk using cloudwatch?

By default cloudwatch does not monitor them, we have to install cloudwatch agent and send custom metrics to cloudwatch.
https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/installing-cloudwatch-agent-commandline.html

Create Alarm In CloudWatch

Create Billing Alarm

AWS CloudWatch Logs

CloudWatch basic monitoring and detailed monitoring

1. Basic monitoring is free
    a. Here metric values are sent to cloudwatch at 5 minute frequency
2. Detailed monitoring comes with additional cost
    a. Here metric values are sent to cloudwatch at 1 minute frequency

Exercise
(FAQ) What are system checks and instance checks in EC2?

IAM (Identity and Access management)

1. This topic is less important for interviews
2. But it is super important for aws job.
3. IAM offers following features
    a. User management
    b. Managing permissions to services
    c. Multi Factor Authentication (MFA)
    d. Federation with third party identity providers
    e. Integrating with Facebook, Gmail, Amazon etc..

Demo - Add new user in to AWS with admin permissions

AWS CLI & Programmatic Access

1. AWS CLI (Command Line Interface)
    a. Through commands we can access AWS
    b. For example start EC2, launch EC2, Launch RDS, etc..
    c. CLI has more operations than AWS Console

Demo - Using AWS CLI Start EC2 instance

1. Install AWS CLI
    a. https://aws.amazon.com/cli/
2. Grant access to AWS CLI
    a. Create IAM User with programmatic access

    b. Grab access keys and secret keys
3. Configure CLI with above keys
    a. aws configure
4. Now CLI and any SDKs running on this machine uses access keys and secret keys

AWS CLI with multiple profiles
aws configure --profile client-a

IAM Role

It is like a user, but it is used by aws service

IAM Role Demo

There is python code running on EC2 which wants to start and stop instances

What is the difference between an IAM role and User?

IAM Role is used by AWS services like EC2, Lambda, EKS, etc..
IAM User is used by users like, hari, ramesh, rahul, etc…

AWS Root User

The user that is used for creating AWS account
We should not use root users for day to day work, we should create identical IAM users and use it.

IAM User

A user created in IAM dashboard

Configure MFA for root used

1. You need to have physical tokens or virtual tokens
2. If we want physical tokens we have to purchase it from AWS
3. Virtual tokens are softwares you can install on laptops or mobile
    a. Install Google Authenticator on Android or IOS
    b. Install Authenticator for Windows phone

(FAQ) AWS Cross Account Access

Users and services in one aws account wants to access resources in another aws account

1. Java code running on EC2 in account “A” wants to access message from the queue that is in account “B”

2. IAM user in account A wants to access resources in account “B”

Demo Cross Account Access

IAM users in Hari’s account want to access EC2 in Veer’s account.
Note: *Cross Account Access will not work for root users.*

IAM Assume Role

Python

```python
import boto3

sts = boto3.client('sts')

veeras_keys =
sts.assume_role(RoleArn='arn:aws:iam::164307201436:role/admin-access-hari',
   RoleSessionName='demo-name')

print(veeras_keys)
```

CLI Assume Role

```
aws sts assume-role --role-arn arn:aws:iam::164307201436:role/admin-access-hari
--role-session-name s3-access-example
```

IAM Policy

1. Create IAM Policy to Send Message on all SQS

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "sqs:SendMessage",
            "Resource": "*"
        }
    ]
}
```

2. Create IAM Policy to Send Message on specific SQS

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "sqs:SendMessage",
         "Resource": "arn:aws:sqs:ap-south-1:762567530299:javahome-queue"
      }
   ]
}
```

3. Create IAM Policy with following permissions
   a. Start All Instances
   b. Stop All Instances

IAM Other Resources

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_iam-tags.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_identifiers.html

AWS Trusted Advisor

Use the Trusted Advisor dashboard to get an overview of the check results in your AWS account. Choose a check name or category to view the recommended actions or potential issues that Trusted Advisor has identified. Each check provides more information about how to address any issues. You can also download a summary of all check results.
Trusted Advisor provides reports on following aspects

1. Cost Optimisation
2. Service Limits
3. Security Checks
4. Fault Tolerance
5. Performance

AWS Cloud Trail

It tracks API activity and User usage
In our AWS account there are 3 users, I want to know what each and every user is doing in the AWS account. Lets say someone launched an EC2 instance and I wanna know who did that, someone stopped the database in production, I wanna know who did that.

- Cloudtrail by default shows last 90 days activities.
- If you want to retain events older than 90 days you have to create a trail and store those events in cloudwatch.

Send Email every time EC2 instance is launched

AWS Databases

AWS offers wide varieties of databases, for different work loads.

1. RDS (Relational Database Service)
2. In memory Databases
3. Datawarehouse
4. Non Relational Databases

RDS (Relational Database Service)

https://aws.amazon.com/rds/

RDS is AWS managed, It is easy to set up, operate and administrate relational databases in the cloud. Because it is a managed service, we are not going to do maintenance of infrastructure.

RDS Supports following engines

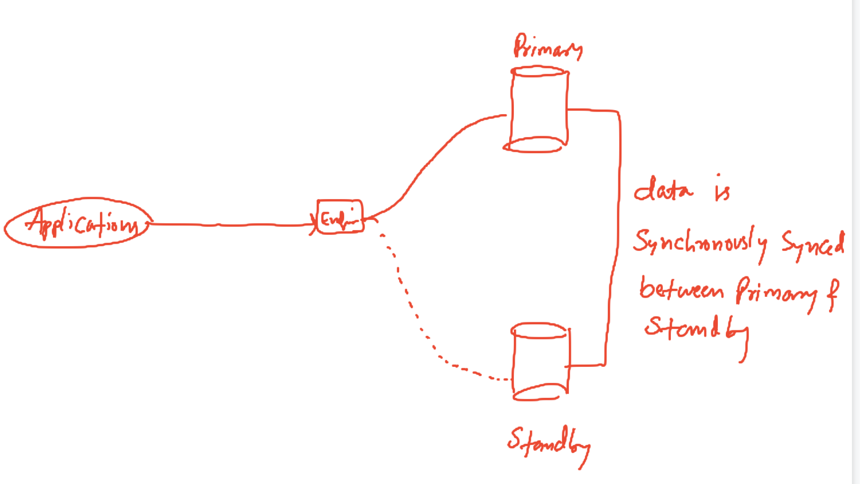
1. Oracle
2. My SQL
3. Microsoft SQL
4. PostgreSQL
5. Amazon Aurora
6. Maria DB

If we need an engine not supported by RDS, then how to use that?

Launch EC2 and Install and configure.

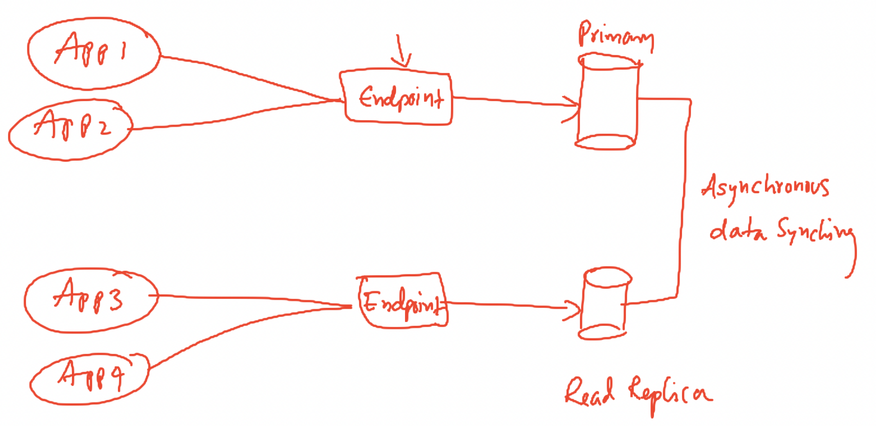
Launch RDS with mysql

RDS Multi AZ Deployment



- Multi AZ is used for designing High Availability for database
- Multi AZ is not for scaling database
- Data added to primary is automatically sync to standby.
- If primary fails, RDS automatically fails to standby.
- They charge extra for multi-az deployment.

RDS Read Replicas



- Read Replicas (RR) are ment for scaling databases
- Read Replica supports only select queries
- Data synching between master and replica is performed asynchronously
- RR can be in the same region or different region, we can leverage this feature for setting up disaster recovery, we can promote read replicas as a master.

RDS Backups

1. RDS supports automated backups, default 7, minimum 0 and maximum 35 days of retention.
2. We can perform manual backups too
3. Backups are stored in S3
4. RDS supports point-in time recovery.
5. Backups are taken ones in a day

RDS Event Subscriptions

Using this we can get certain notifications for certain events like

1. Low storage
2. Database failures
3. Failovers
4. Maintenance
5. Snapshots
6. Etc..

RDS Amazon Aurora

1. It is a mysql and postgresql compatible db.
2. Aurora DB is 5 times the throughput of mysql and 3 times the throughput of postgresql.
3. Maximum storage size of mysql is 64 TB
4. Aurora supports upto 128 TB
5. It supports upto 15 read replicas
6. Aurora does 6 way replication, it maintains 6 copies of data across 3 AZs.
7. Aurora supports global databases, we can set up multiple instances of DB across regions,
8. It has migration support from postgresql and mysql to aurora.
9. Aurora serverless feature
    a. You specify the minimum and maximum amount of resources needed, and Aurora scales the capacity based on database load. This is a good option for intermittent or unpredictable workloads.
    b. It will be cost effective

DynamoDB

1. DynamoDB is a non relational DB.
2. It can handle an unlimited amount of data.
3. Single digit millisecond performance at any scale.
4. It is non relational, it does not support joins, this improves the performance.
5. It can be used for web, gaming, e-commerce, etc..
6. Setting up dynamodb is pretty simple, we directly create tables and use it.
7. *Dynamodb is used by amazon for implementing shopping carts*.
8. *Dynamodb is also used for user session management.*

9. DynamoDB supports global database
10. DynamoDB has a feature DAX (Data Acceleration)

Redshift

1. Redshift is a data warehouse solution.
2. It has features like a columnar database, its architecture is finetuned for analytics like reporting etc.
3. Redshift is a relational database.
4. Redshift does not support read replicas.
5. Redshift has support for automated backups to cross regions.

Elastic Cache

1. It is an in-memory database, instead of storing data on disk, it stores data in RAM, It offers fast access to data.
2. It is used to solve performance issues at application level.
3. It is a good choice for session management.
4. It supports two engines
    a. Memcached
    b. Redis

S3 (Simple Storage Service)

- It is object based store
- It is right choice for storing
    - Photos
    - Videos
    - Excel sheets
    - PDF
    - Word Doc etc.
    - It is used as a data lake, to store big data.
    - DevOps teams use it to store artifacts war, jar, zip, exe, etc..)
    - S3 is used for storing logs centrally

S3 Characteristics

1. S3 is object based store
2. We can store unlimited amount of data
3. Each Object(file) on S3 can’t exceed 5TB.
4. To store data on S3 we have to create bucket
    a. And we should store objects in buckets.
5. Objects can be uploaded and downloaded directly
6. S3 is an internet based service.

S3 Bucket Naming Rules

1. Bucket name must be unique across aws accounts
2. It must be in lowercase
3. It can be alphanumeric
4. It does not allow special characters, however bucket name can contain, "." and "-"

S3 Bucket Versioning (Certification Exam)

1. We should use versioning to recover from accidental deletion of objects.
2. Ones enabled we can't disable, we can only suspend.

S3 Encryption

- We can encrypt objects at S3, by default encryptions are disabled.
- We can encrypt objects at rest using KMS

S3 Event Notifications

1. Let's say we want email notification if an object is deleted from S3.
2. We receive json files into S3, we have to automatically process them and persist data into RDS(mysqL)

S3 Static Website Hosting

1. If there is a static website(html,CSS,Javascript, Node, Angular) in your project, the best place to deploy that is S3.
2. Steps to host static website
    a. Grant public read access for all objects in S3.
    b. Upload your website code
    c. Enable static website hosting under bucket properties.

Server Access Logging

This keeps track of all operations happening on s3 bucket.

S3 Storage Classes (Certification)

Choosing the right storage class optimises the cost.

1. Standard Storage
    a. It maintains 3 copies across multiple AZs
    b. It is used for frequently accessed data
    c. It gives 99.9999999 durability
    d. 99.99% of availability

2. Standard IA (Infrequently Accessed)
    a. It maintains 3 copies across multiple AZs
    b. It is used for infrequently accessed data
    c. It gives 99.9999999 durability
    d. 99.99% of availability
    e. Storage cost is cheaper than Standard Class.
3. Reduced Redundancy
    a. It maintains 2 copies
    b. It is meant for frequently accessed data.
    c. Use this if you have a similar copy somewhere else.
4. Intelligent Tiering
    a. Sometimes we don’t know the access patterns of objects, then put them in this class.
    b. It monitors access patterns and accordingly it is placed in the right storage class.
5. One Zone IA
    a. Long-lived, infrequently accessed, non-critical data.
6. Glacier
    a. Long-term data archiving with retrieval times ranging from minutes to hours
7. Glacier Deep Archive
    a. Long-term data archiving with retrieval times within 12 hours

S3 Lifecycle Rules

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time.

1. For example we store application logs in S3, we may need to access them frequently when logs are new, and when they become old, we do not access them frequently, so to optimize the cost, We want to set following rules on objects
    a. After 30 days move to Standard IA
    b. After 60 days move to Glacier
    c. After 90 days move to Glacier Deep Archive
    d. After 1 year they expire.
2. We enabled versioning in S3 to prevent accidental deletion of objects, due to this we see lots of previous version objects sitting in the S3, we want to automatically delete them after 45 days.

S3 Replications

1. Cross Region Replication
    a. Source and destination buckets must be in two different regions
    b. Versioning must be enabled on both the buckets.

    c. After replication is enabled only future objects are replicated and old objects are not replicated.
2. Same Region Replication
    a. You are going to replicate objects within the same region.

Amazon S3 & Glacier Object Lock (You check yourself)

AWS Snowball

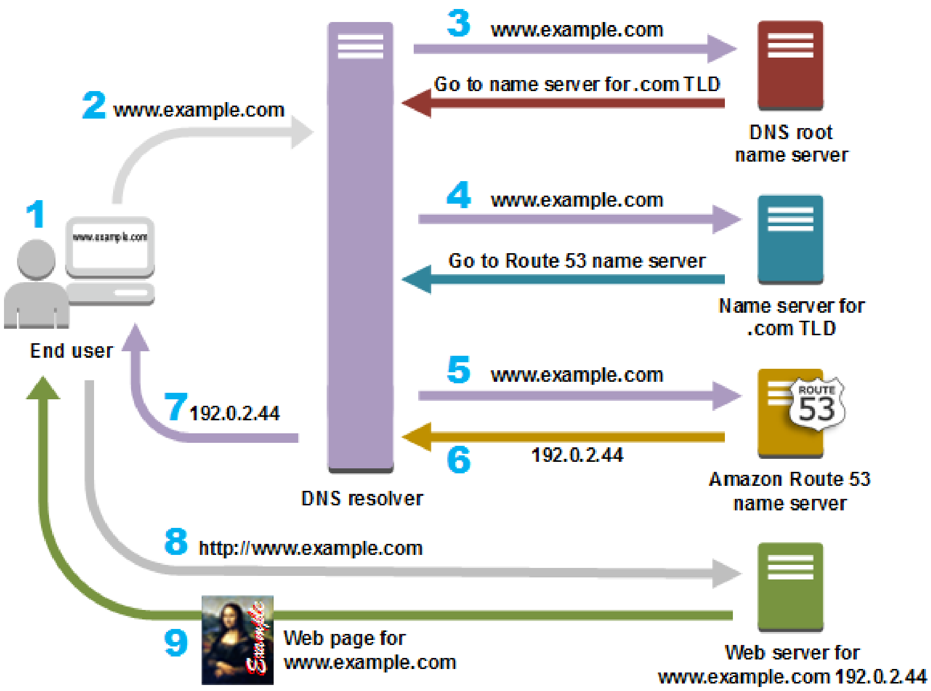
AWS Snowball is a data transport solution that accelerates moving terabytes to petabytes of data into and out of AWS using storage appliances designed to be secure for physical transport.

AWS Route53

1. It is DNS in AWS
2. Why name Route 53?
    a. The default port of DNS server is 53 so they named it Route 53
3. We can register for new domains using Route53
4. If you have domain registered with third party and you can import it into Route53
5. To see demonstrations with route 53 we need to register a domain.

DNS (Domain Name System)

DNS is internet phone book



Route 53 Demo, Setup

1. We do not want to pay for domain in route 53, so we wanna get one domain for free in dot.tk website
2. Signing into dot.tk with gmail account
3. Create one free domain
4. Goto Route 53, create public hosted zone with the domain name created in dot.tk portal
5. Grab NS records in route 53 and update them in dot.tk

Demo Map Domain With EC2 Instance

Goto hosted zone, create record, choose type A and paste your IPV4 and create.

Demo Map Domain With ELB

Goto hosted zone, create record, choose type A and Alias Yes, choose load balancer URL and create.

To Know more about DNS record types, check this

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/ResourceRecordTypes.html

(FAQ) How can we load balance EC2 instances in two regions?

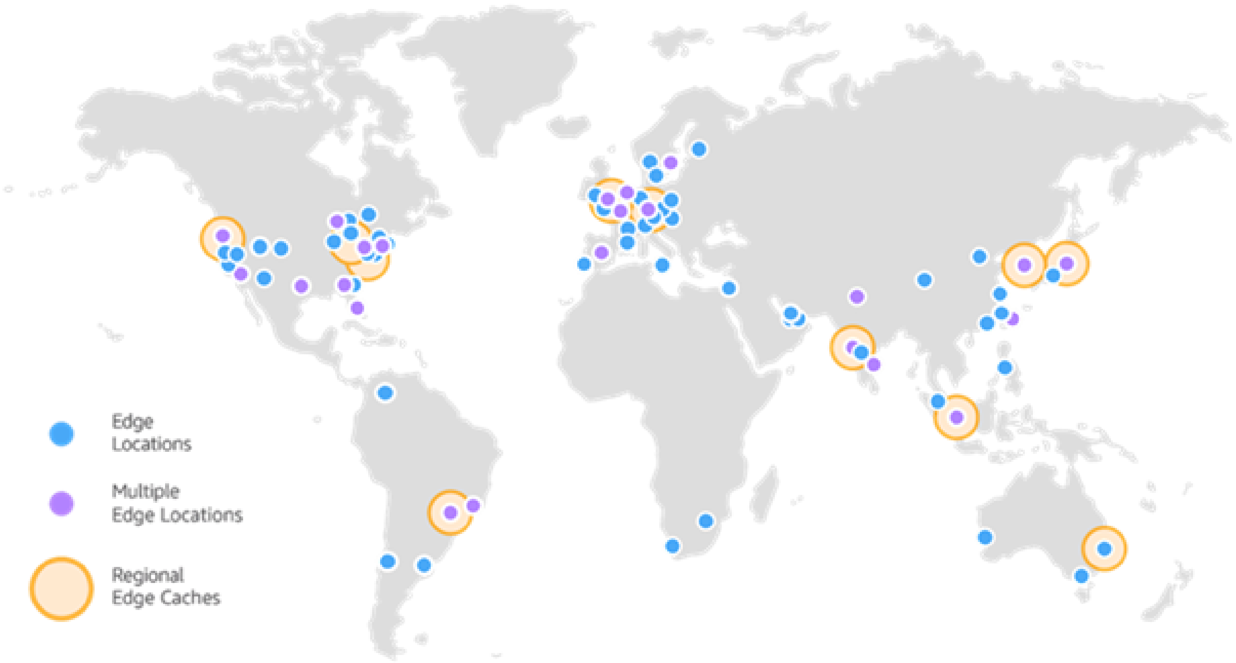
Put one load balancer in each regions and configure it in route 53

Route53 Routing Policies

Routing policies let you choose how Route 53 routes traffic to your resources. If you have multiple resources that perform the same operation, such as serving content for a website, choose a routing policy other than simple. Here's a brief comparison:

- **Simple**: Simple records use standard DNS functionality.
- **Weighted**: Weighted records let you specify what portion of traffic to send to each resource.
- **Geolocation**: Geolocation records let you route traffic to your resources based on the geographic location of your users.
- **Latency**: Latency records let you route traffic to resources in the AWS Region that provides the lowest latency. All resources must be in AWS Regions.
- **Failover**: Failover records let you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy.
- **Multivalue answer**: Multivalue answer records let you configure Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries.

AWS CloudFront / Content Delivery Network (CDN)



1. Cloudfront is a service which caches the content of applications and offers better performance.

2. Cloudfront can cache static and dynamic content.
3. Cloudfront uses edge locations for caching the content.
4. Cloudfront can be configured for
    a. S3
    b. Http endpoints
5. Origin server
    a. Origin server is the one where actual application is hosted
    b. If application is hosted on S3 and S3 is origin
    c. If the application is hosted on EC2 then EC2 is the origin.
6. Origin Access Identity (vv Important)
    a. This is a special user in cloudfront, using which we allow only CDN to communicate with S3.
7. Pricing Class
    a. All edge locations
    b. North America & Europe
    c. Use North America, Europe, Asia, Middle East, and Africa
8. WAF Integration
    a. CDN can be integrated with WAF and can secure our applications
9. Cloudfront pre-signed URL and signed cookies.
    a. Using above techniques we can grant access to objects for specific users.

CloudFront Demo S3

1. Create S3 with static website hosting.
2. Create CloudFront Distribution

Python Programming in One Hour

https://youtu.be/XokLzvcB4pY

Automate Boring Stuff with Python

https://automatetheboringstuff.com/

Python & AWS Lambda Functions

Setup local environment to work with python and AWS

1. Install python3
2. Install Boto3
    a. Boto3 is python SDK for AWS
    b. pip3 install boto3

3. Create IAM user programmatic access and configure keys on your laptop
4. Install Visual Studio Code

Youtube Boto3 & Lambda videos

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

AWS Lambda Functions

1. AWS lambda is a serverless compute engine, we can run code without thinking about servers.
2. For lambda we pay only for execution time.
3. It supports various languages like
    a. Python
    b. Java
    c. .Net
    d. Go lang
    e. Ruby
    f. Nodejs
    g. Custom runtimes
4. How to invoke lambda functions
    a. Lambda functions can be invoked based on event or based on schedule.

Lambda Function, automatically starts the EC2 instance if it stops.

Send email notifications when EC2 is terminated.

AWS Events structure

https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/EventTypes.html#ec2_event_type

Lambda memory & CPU settings

Default memory is 128MB, CPU is directly proportional to memory.

Lambda Timeout

Default timeout is 3 seconds, max can be 15 minutes.

Lambda and VPC configuration

For example, the lambda function has python code to talk with RDS, so lambda should be scheduled inside VPC.

Lambda environment variables

Invoke lambda when EC2 is launched

AWS Lambda use cases

1. Auto tagging ec2 instances

Infrastructure as Code & Terraform

- IaC is nothing but create infrastructure through code
- In aws infrastructure can be any resource like, VPC, Subnets, RDS, EC2, Lambda, S3, etc…

CloudFormation is an alternative option for Terraform.

Setup terraform locally

1. Install terraform on your local machine

(FAQ)What is a terraform plan?

Terraform plan is a command that presents the changes it's going to make when we run the apply command.

Destroying all resources

terraform destroy --auto-approve

Deleting a specific resource

Comment the resource you want to delete and run the apply command.

Terraform state file

1. Terraform maintains details of aws resources that are managed by terraform in a state file.
2. When we do terraform apply it uses a state file to figure out the changes it has to perform.
3. State file is stored in a local workspace, with a name terraform.tfstate.

(FAQ)What happens if we delete a state file and run the terraform apply command?

It creates a new set of resources by keeping old ones.

(FAQ) Where do you store your state file?

In our project terraform state files are placed in S3 bucket.

(FAQ) How do you prevent multiple terraform developers from making changes to state files simultaneously.

We have to configure a lock, such that only one person is allowed to make changes at a time.

Terraform Resource Targeting (FAQ)

```
terraform apply --target=aws_vpc.main
terraform destroy --target=aws_subnet.main
```

Terraform tainting a resource (FAQ)

Sometimes our resources are partially created and not fully functional and we want to recreate that resource.
Resources that are tainted are recreated when we terraform apply.

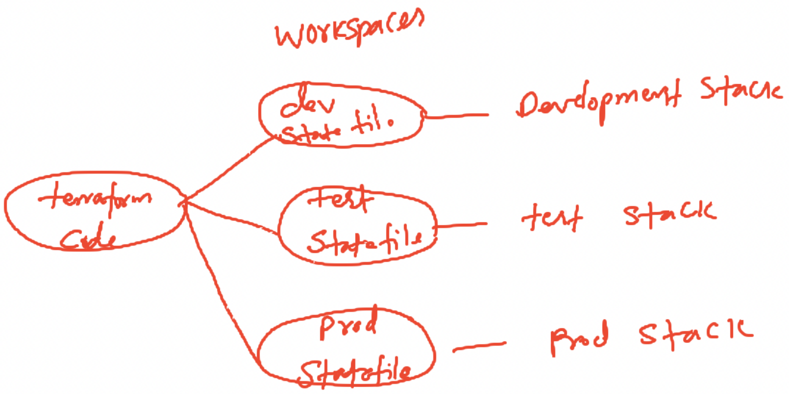
Terraform Data Sources (FAQ)

Using data sources we can fetch certain details from AWS.

1. We can get AWS account ID using datasource
2. We can get list of availability zones

Terraform Local variables

Terraform Workspaces (FAQ)



Using workspaces we can create multiple state files to manage multiple environments with a single code base.

terraform workspace list
terraform workspace new test
terraform workspace new prod
terraform workspace select default

Difference between local and global variables

1. Local variables can't be injected at command line
2. Local variables accept expressions
3. Global variables can be injected at command line
4. Global variables does not support expression, it take only static values

https://github.com/javahometech/iac-nov-2021.git

Terraform Import

Import helps migrating manually created resources into terraform.

Terraform modules

Using terraform modules we can develop reusable terraform resources, for example if we create a module for RDS, it can be reused.

Create networking module

- This module should create vpc, public subnets, private subnets, etc.

Multiple Providers

https://www.terraform.io/docs/language/providers/configuration.html