

AWS is a infrastructure as a service platform which allow much more power full than digital ocean.

There fore many companies like Netflix use AWS to host there application .

There are any AWS services that are grouped into categories like machine learning , block chain , IOT, game development .

We need general software development related services where we are developing , building and deploying our application on a server some of the services we will be focusing on :

1. Compute => EC2 (virtual servers that we can hire) we need virtual server to deploy our application eg: to install docker and run docker container in it.
2. Storage => if u have DB or some other service that need data persistence we need to save data some where , specially the S3 service which is the most popular service in AWS .
3. Database => AWS has its own database services we are not gonna use this we can use our own database like postgras to mysql which we can deploy and run on virtual servers which we can manage it our selfs .
4. Networking and content delivery= > VPC (isolated cloud resources)(we need to configure our virtual servers with firewall and network configure .
5. Security , identity and compliance = > IAM users manage access to AWS resources to mange groups , users , permissions in AWS .
6. Containers => repo for docker container or kubernetes services etc.

There are the core services to feel confident on AWS .

SCOPES OF SERVICES :

When u create your AWS account we automatically get a global top level resource which is your account with that account u can create infrastructure in any of the region available in AWS every region will have 2 or more availability zones which are actual physical data centres which will run virtual machine

U have 2 scopes in AWS => global , region and AZ scope in AWS different resources will be created in one of these scopes .

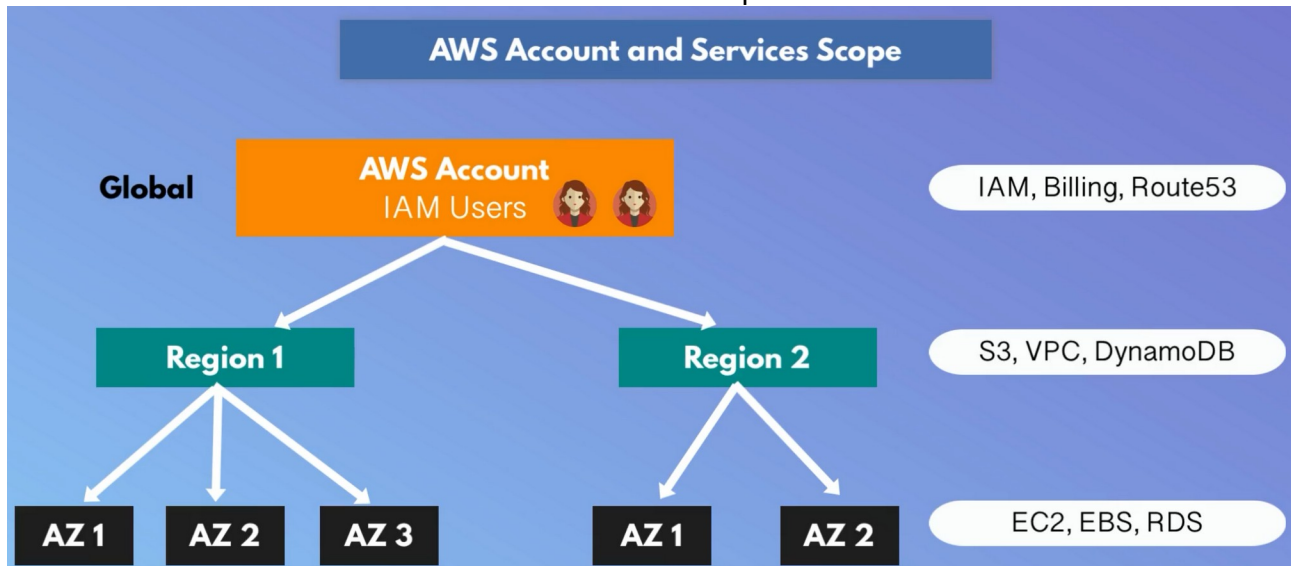
Eg: iam service to creating AWS users and there permission apply to the whole account , when u create a user with certain permission that user will have access to all the regions because they are created per account

U also get one billing for the whole account in AWS so these resources are scoped globally

Region scope : resources like S3 buckets for file storage and VPC for private network are created per region

AZ scope : services like EC2 instance which are virtual servers. , RDS database service are created inside the availability zone

All AWS services are created in one of those 3 scopes.



CREATE AN AWS ACCOUNT :

Use free tier account which u can use for 1 year for free after registering in AWS account /

Always delete the resource when u are done with learning as soon as possible so that u don't get charge of that .

IDENTITY AND ACCESS MANAGEMENT (IAM) :

Manage access to AWS services and resources .

Who is allowed to access your searches on serves and resources ?

Create and manage AWS user and groups and assign them permissions accordingly .

By default when u create a account on AWS u have a root user , root user has unlimited privileges .

Good practice is to create a admin user that has less privileges than root user .

Root user is the only when that can delete the entire account .

This is the first thing that AWS advices to do for security reasons

In addition to human users in AWS we also have **system users** : for eg: u have a application like Jenkins that automatically deploys docker container on AWS EC2 instance .

Or Jenkins automatically pushed docker image to AWS docker repo .

Access management =>

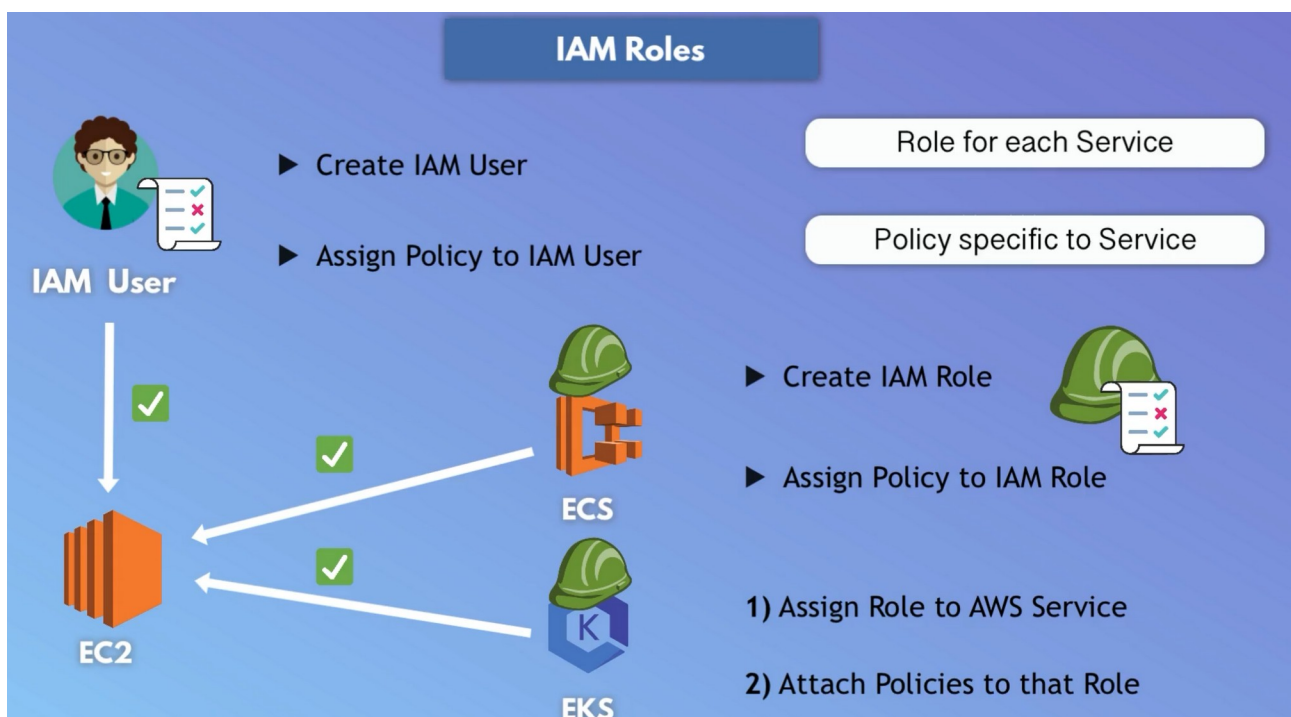
Users: human users or system users you give access to IAM user like Jenkins user giving Jenkins permission to connect or get access to your AWS account or to some services.

Groups : for granting access to multiple IAM user . To group those users that share similar permissions and manage that group at some time rather than managing them individually .

Roles : if u have multiple services in AWS that u are using in combination one series want to access some feature of other service

IAM USER VS IAM ROLES :

Iam user : human or system users users are assigned to users who are using your AWS account and creating services and configuring them etc. eg: u are a devops team member and u are configuring EC2 instance in AWS account and u can delegate your work to other AWS services so other AWS service will act as a user will have same permissions as u have to perform the same task in AWS we can not assign policies and permissions to AWS services directly like we to users directly to services so instead we assign service a role and then assign policy to that role



CREATE AN IAM USER :(ADMIN USER)

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name
admin

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

ⓘ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

AWS Management Console

AWS Command Line Interface

► Email and Password

► Access Key Pair

AWS User has separate credentials for each type

First option is to have a UI access to the user , Second option is to have a command line access to the user

U can select if user have one of those access types or both . Here we are going to have both type of access to the admin user .

If we login to that user we need a password as well so ui access will be through user name and password and from command line we gonna need secret access keys to login to AWS and execute command in terminal

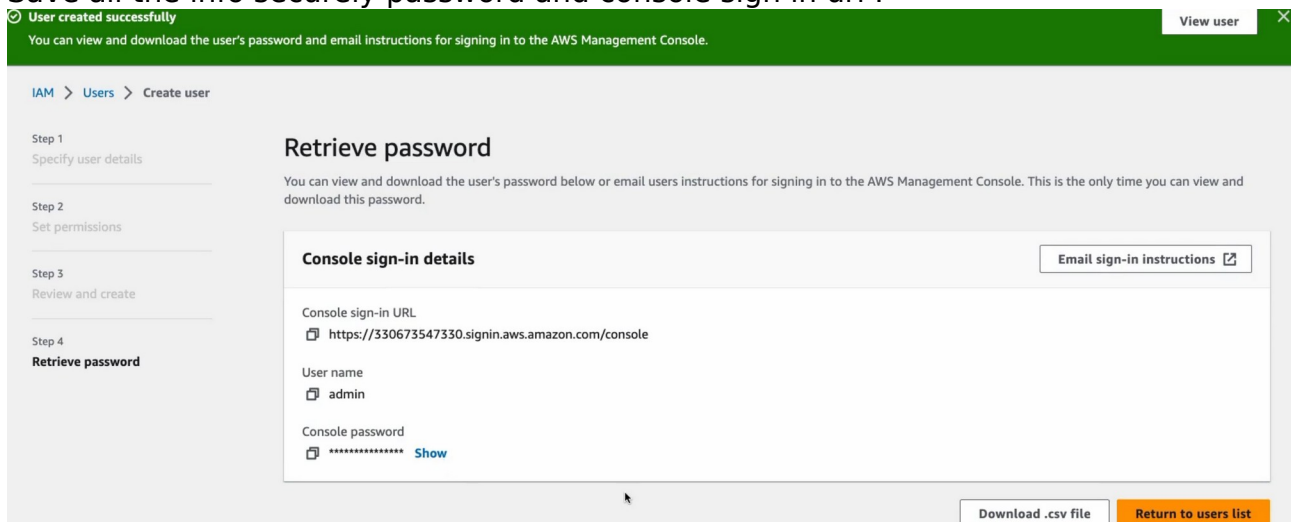
Now we can have a group that a user can be a part of it is the best practice to have all the users In a group

And then assign permissions to the group rather than directly to the user here admin user is going to be only one in the group so we can skip creation of group so attach policy directly to admin user .

Best practice here is : least privilege principle

Provide administrator access to admin user .

Save all the info securely password and console sign in url .



Change the password later access key for this user and configure access key locally on your laptop as well as install AWS command line interface to execute some commands and task .

From your local machine

Now login as admin user ==>

Programatic access ==> complete our account setup by giving our new admin access programatic access

Go to security credential and create a access key will allow us to connect to our AWS account using CLI . U can set a tag to differentiate btw multiple access keys .

REGIONS AND AVAILABILITY ZONES :

Cloud providers have physical data centers when u rent a virtual server from a different cloud provider these server are running physically some where in some data centre .

If some thing happen to that data centre u may use your data , configuration data of your set up etc.

So there for it have 2 or 3 data centres in that same region called availability zones . That let u replicate your data .

Eg: if a company is operating in USA but there user are in Asia they need to deploy there

Region



Virtual Private Cloud (VPC)

- ▶ VPC for each Region
- ▶ VPC is your own isolated network in the cloud
- ▶ VPC spans all the AZs (Subnets) in that Region

Region

Your VPC

Region

Your VPC

Availability Zone

Subnet

EC2 Instance

RDS Instance

...

Availability Zone

Subnet

EC2 Instance

RDS Instance

...

- ▶ Private network isolated from others

Your VPC

Tom's VPC

Lara's VPC



Regions & Availability Zones

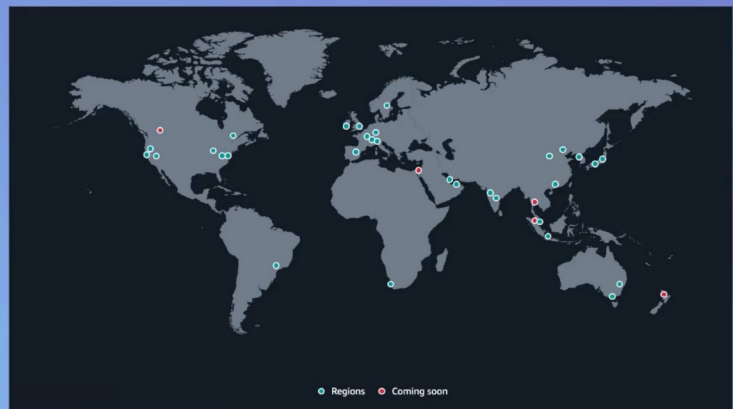
- ▶ Cloud providers have physical data centers
- ▶ Data centers all over the world
- ▶ Region = physical location, where data centers are clustered

30+ Regions

- ▶ Availability Zone = 1 or more discrete data centers

90+ Availability Zones

AWS Regions



application in Asia region server so that Asias users can access that application faster. Or your users are distributed so u can replicate your application in different regions .

VIRTUAL PRIVATE CLOUD :

When every u create your account in each region u will have a VPC .
On services list -> network and content delivery service -> VPC service for each regions VPC are created my default .

VPC are your own private isolated network in your cloud for each specific region
VPC spans all availability zones in that region .

VPC are virtual representation of network infrastructure if u have physical servers in your company which was traditional way of hosting your application before cloud u have sys admin which manage setup configure this whole network they set up the router , configure firewall so on .

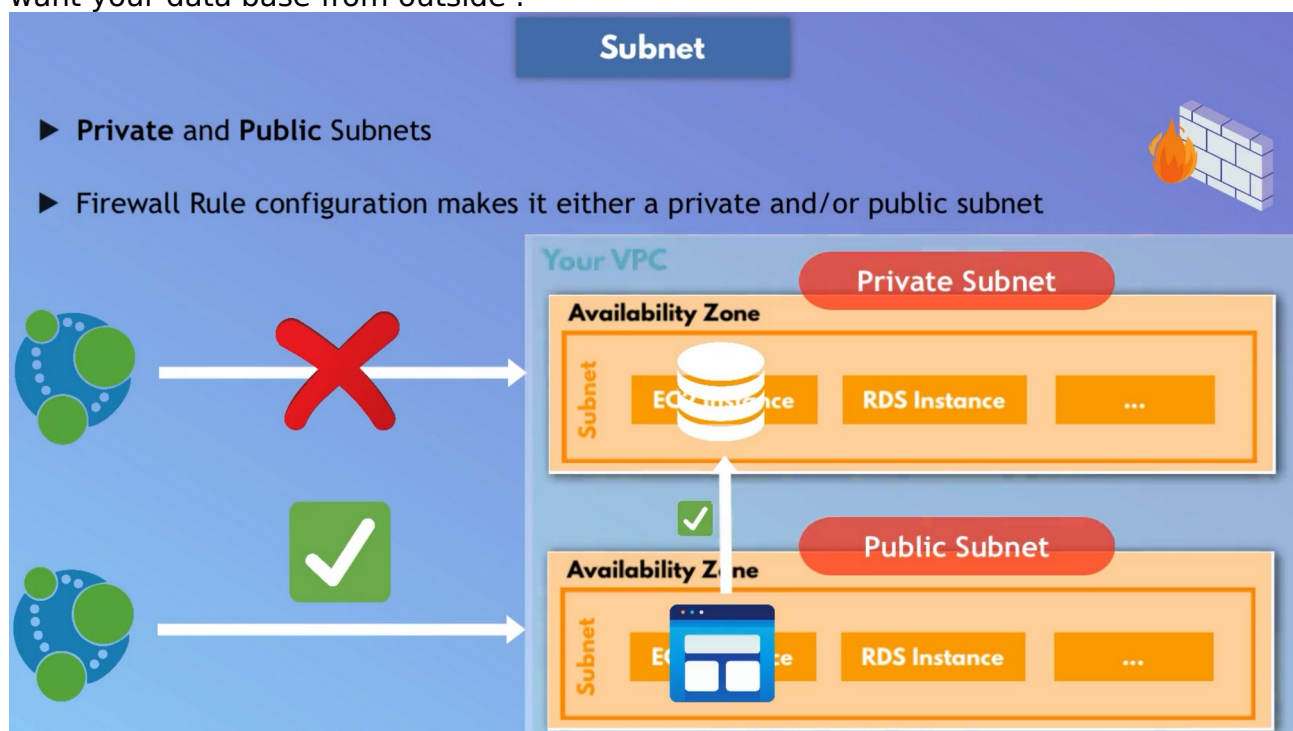
So this physical network infrastructure is basically moved in cloud and VPC is the representation of this.

There is a reason u have a default VPC on a region is when every u create a EC2 instance it has to run in VPC or what ever component u are launching it has to start inside the VPC .

Subnets : are the subnets of VPC its like a private network inside a network .

Private and public subnets : firewall rule make it as a private of public subnet when u block internet communication from out side u are making it as private subnet because external traffic can not make enter that subnet
But other services inside the vPC still have access .

Use case : u have a private subnet where your data base in running because u don't want your data base from outside .



Public subnet that allow external traffic into that subnet eg: web application . And your web application is allowed to talk to that private subnet to access database because they are in the same private network the VPC .

In VPC u have range of internal ip addresses that range is refined by default and u can change it even . So when every u create a new EC2 instance a ip address will be

assigned to it from this default range . We have multiple virtual machine inside the VPC they need to communicate with each other Its not for outside web traffic for communication inside the VPC . So private ip address is what allows them to talk to each other .

When we deploy EC2 instance we want it to be accessible in the Internet from outside therefore a private ip address we need to assign it to public ip address . And this is also configure in VPC service .

So when we create a EC2 instance it will get private ip address from the range of subnet inside which that instance will start and it will also get a public ip address . Private ip address is to do communication btw different services inside the VPC and public ip address to access that instance through the browser .

U also have internet gateway that will allow the internet access of your instance . Internet gate way connects the VPC to outside internet .so that u can get traffic inside your VPC .

U can request and download some thing

Controlling access :

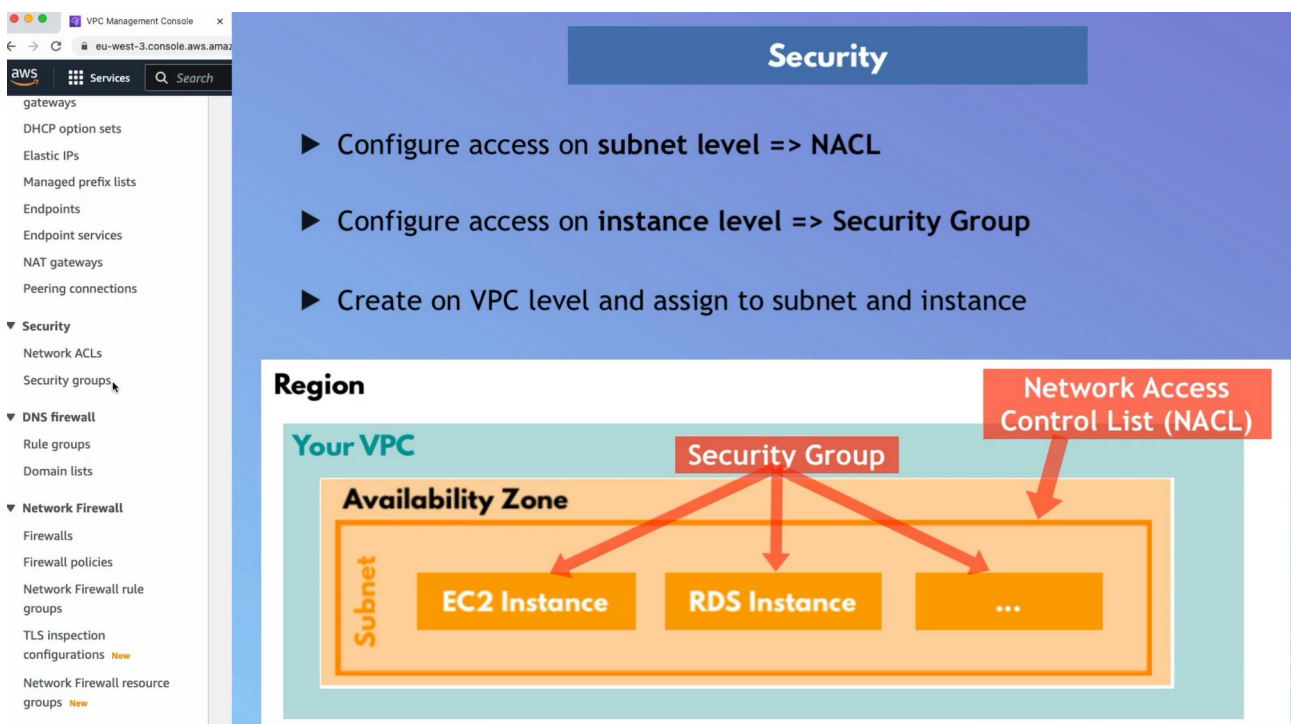
Secure your components

Control access to VPC

Control access to your individual server instance .

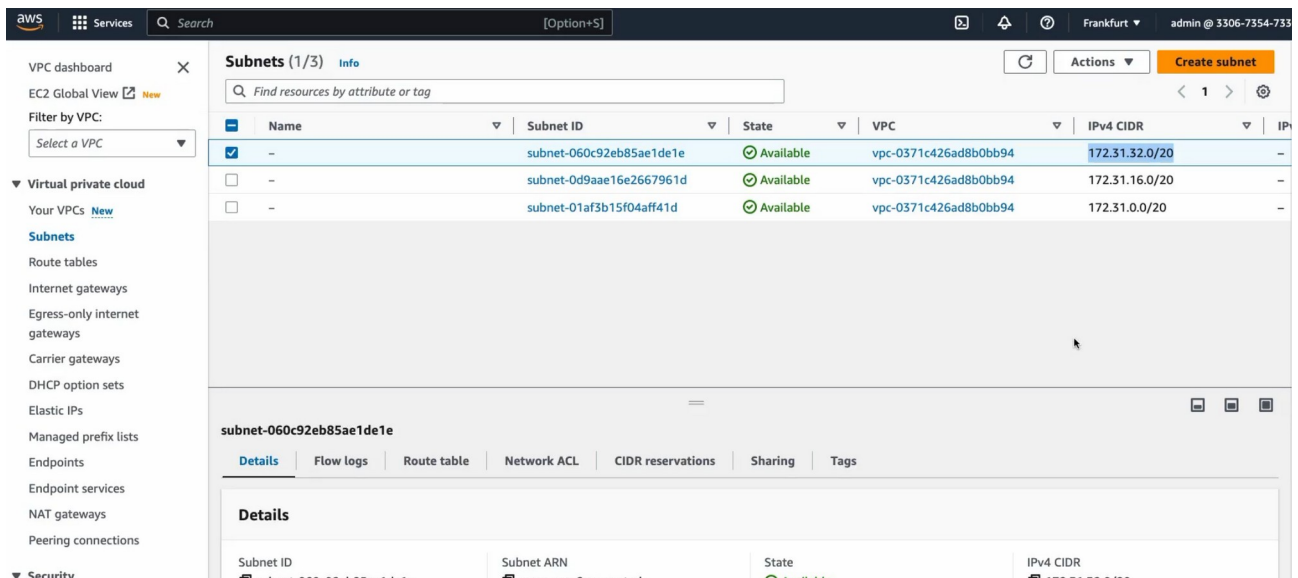
Security we. Can configure in the subnet level and the instance level .

Which port is accessible from the internet for each subnet . The way to control access control in the subnet it throw components called network access control list .(NACLs) .



CLASSLESS INTERDOMAIN ROUTING :

When we define a VPC a sider block is defined which represent the subnets which are part of that VPC



What is CIDR block

How to choose a CIDR block ?

AMAZON ELASTIC COMPUTE CLOUD(EC2) : virtual server at AWS cloud

Provides computing capabilities .

What we will do ?

We are gonna deploy a web application at EC2 instance .

1. Create an EC2 instance on AWS .
2. Connect the EC2 instance with ssh .
3. Install docker on EC2 instance
4. Run a docker container(docker login , pull , run) from private repo.
5. Configure EC2 firewall to access app externally from browser .

Lean AWS zero to hero :

Day 1 :

What is cloud ?

10 year back company buy server ibm or hp they bring those servers back and do all the configuration put all the wires u need to create network and maintain there datacenter .

First there was no virtualisation eg: if server is of 100 gm ram and app is of 1 gb they other 99 gb get wasted .

Virtualisation is a process of saving the waster of resources on the server eg: instead of deploying one application on the server using virtualisation u will create virtual server on those physical servers. And each servers u can deploy multiple application so now instead of buying multiple servers u can buy one server with required no of configurations and create multiple virtual machine on it.

Now by this virtualisation concept u can create a virtual machine from any part and the world . I don't know where they servers are they share the ip address with up of those virtual machine we can use them this is cloud it is interconnected u don't know where those resources are but u are using it. U are don't this with in the organisation this concept is know as private cloud

From other organisation no one can request u that instance and u will not give them that instance .

So Amazon , microsoft , google see a opportunity here eg: for a start up it is difficult to maintain all this like temperature control , configuration server maintains management etc.

Even for mid scale of high end companies it was hard to them to maintain all this .

So we will buy these servers we will create multiple infrastructure in multiple places in the world and one can buy a EC2 instance(virtual machine) from us . This concept is called public cloud .

Pay as u go services public cloud .

Why public cloud is so popular?

People want to get rid all this this maintenance of servers , network configuration every time u have to see if there is any security issue etc. os for a start up if they need to maintain these 15 - 20 server they need 15 to 20 people to do that which is over head .

So AWS has over 200 services when ever they see a popular demand of resource in the market they create a service out of it.

How AWS is popular than other ?

AWS has first comer advantage . They initially start is public cloud concept . Larger market share more job openings .

Are people moving away from the cloud ?

Moving from public cloud to private cloud is called cloud Cloud repatriation . Cloud to on premises but this no. is very less.

Its hard to maintain and set up in on premises . Money and maintenance are the two major this from them to move to private to public cloud .

Day 2 .

AWS IAM user .

Eg: in bank to enter the back u need to be a authenticated person and once to enter the back what authorisation u have to access a particular service .

So IAM in AWS is a AWS service which is doing the authentication and authorisation for u .

Eg: if an new employ comes in then devOps engineer with root access will create a user of this employ with access to the services it want in AWS .

IAM -> users , policy , groups , policies, roles .

User => u have provide authentication for user to enter the AWS creation of user .

Policies => what that user has to do

Groups -> in a company there are people who keep on joining and leaving the organisation if every time u create a user and attach a policies then this will waste your time . Developers , QA , DB-admins and others groups as a devOps engineer u will create these groups.

Eg; now if an employee joins the company, the employee will send the JIRA request to devOps engineer and saying I belong to developers group and then devOps engineer will put that user to that dev group . Where u do all of these things in a service called IAM .

Roles :-> when two AWS want to talk to each other .

In interview always tell that we don't use root user and use admin user . If there is a new developer we create an IAM user to that developer and then add that IAM user to his particular group eg dev , test etc. groups.

EC2 deep dive :

Elastic cloud compute : combo of CPU , Ram , Disk . Asking AWS for virtual server .with hypervisor and the concept of virtual machine u can users server into multiple virtual machines .

AWS is a public cloud

Elastic means these services are elastic in nature which we can increase in size or decrease them . Or use those resources in any way .

So there are lots of services in AWS with elastic as prefix : like elastic load balance , etc.

ECC : u are asking AWS to give me a virtual machine on which is elastic in nature .

Why ? Use EC2 instance : ? : instead to creating your own hypervisor on a server . I will my own start creating these virtual machine and start giving to the devops engineers .
Drawback : every time u need to create these virtual machine and time consuming and security issues .

To get rid to these maintenance , to manage these management AWS come into picture .

Pay as you : u don't want these servers to be up during the night or any particular time so we .

On these cases u can shut down the servers and AWS will not ask u for nay money .

TYPES OF EC2 INSTANCE. :

1. General purpose :
2. Compute optimised
3. Memory
4. Storage optimised
5. Accelerated compute

It depend son the type of application : is it machine leaning , data analytic , gaming , memory big data or general .

REGIONS AND AVAILABILITY ZONES :

U can ask AWS to give u a ec2 instance in us in a specific availability zones .

Eg: u are working for a us client and they ant there data as close as possible as a devops engineer u will create EC2 instance in that region .

To decrease the latency we need to make datacenter closest to the end user .

Even if u use free tier instance : there is a time limit of 750 hours per month for a year only one instance should be running through out the year t2.micro . So u have to manage the time of multiple intense .

Key pair : helps u to login to that instance is a combination of public and private key .

Instance has a public key and u will have a private key

-> create new key pair -> set key pair name -> key pair type RSA then create key pair.

Now in terminal go to downloads and search for that pem file containing keys .

-> Ssh -I first_login.pem [ubuntu@13.201.3.164](http://13.201.3.164) (**public ip address of the created instance**)

-> **chmod to changet the permission of first_login.pem**

Chmod 600 first_login.pem

Now I will be able to access this instance server .

:: whoami , who will tell u which user are u using in this instance .

:: sudo su - , will switch u to the root user on instance .

: if u are in ubuntu user or any other user instead of the root user , u will use the sudo commands to get the privileges of root user .

Always update the apt repo : apt update .

Now installing Jenkins on our instance :

First we need to install java to run Jenkins

: sudo apt install openjdk -11-jdk

-> go to Jenkins official website search install Jenkins in ubuntu => weekly release run that command given here in terminal instance .

: systemctl status Jenkins , this command will let u know tif Jenkins services is running or not .

: by default Jenkins run on the port 8080

<http://13.201.3.164:8080/> : , here u can access your Jenkins via public ip address of your instance but it will run but u can don't access it why ?

Because if u create instance in AWS part from it there are lot of setting.

Your application is by default if not accessible in the out side world , we need to open that port 8080 ,

In was instance : in security section there is a inbound and outbound traffic rules :

Iadd new inbound rule -> allowing port 8080 to open .

VIRTUAL PRIVATE CLOUD :

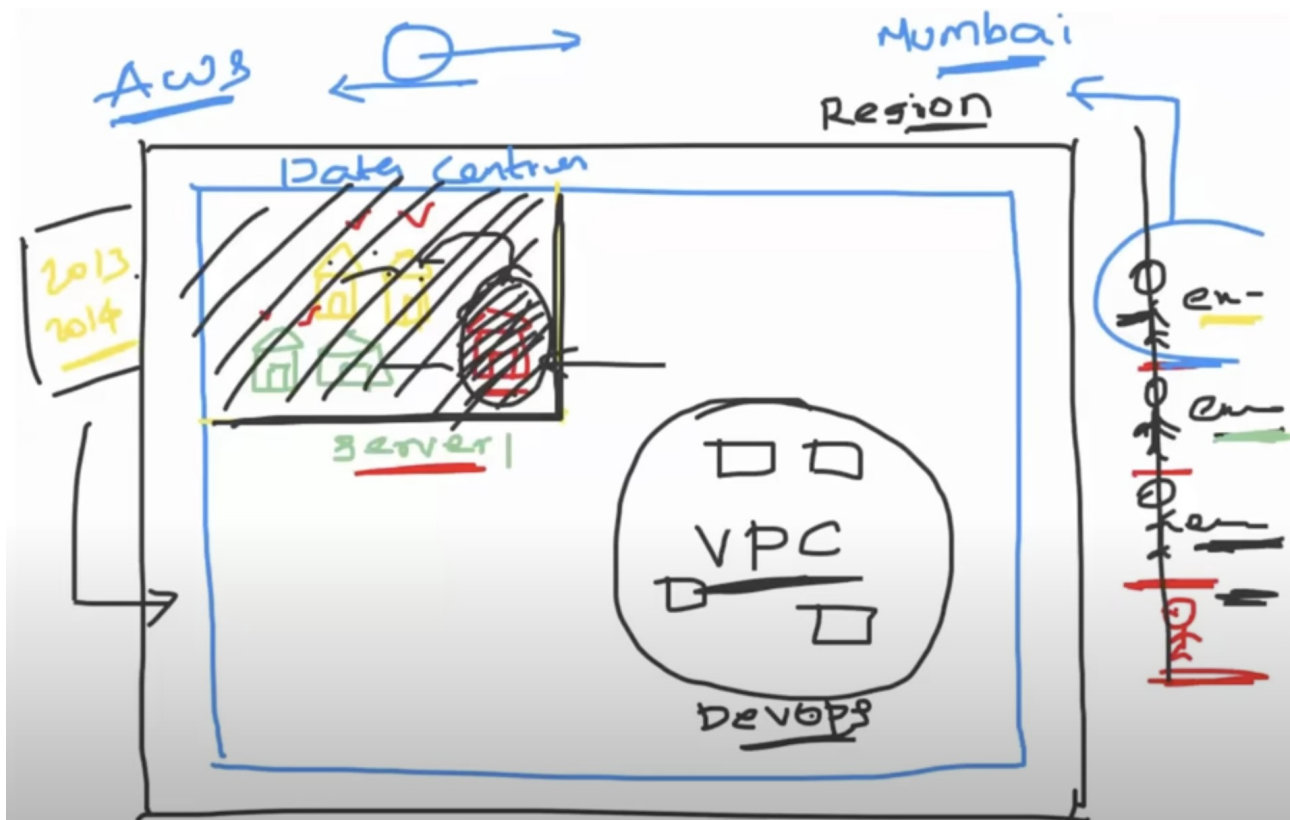
AWS have data centres with multiple servers with AWS rent the virtual machine to the companies .

With in the same verser was is providing the rent or virtual machine to different companies .

Which can lead to security breach .

If in one of the virtual machine(rented server of a company) if that company did not maintain security and hacker comes there it can effect other virtual machine also in that server .

So till 2013 and 2014 this was happening to solve this security breach AWS introduced VPC .



AWS devops engineers build these VPC , they will configure every thing inside the VPC /

U can define the size of VPC with the range of ip addresses .

For my project I need these no of ip addresses

VPC has 1 ip address (one network) but I will split that network into subnets this concept is know as subnetting .

Now devOps engineer will create a gate way to enter this VPC . Through the internet gateway there is a public subnet through which the user enters the vpc .

So u will create a route table through the route table u will define through where the request should go to which server or instance or project

Security group will guard that server or instance or project where the request is routed inside the VPC .

NOTE : From internet there is a person who wants to access a application from the internet application has ip address a.b.x.d/xx , There is a internet gate way of a VPC Inside VPC the whole VPC have a ip address range and inside VPC u have divided the VPC network into subnets for hosting different servers .

Now the request will pass through the gate way and there is a public subnet (public subnet is the one that can be access to the public out side the VPC but the request has to pass through the internet gateway in the public subnet there is a Elastic load balance , your request from the external world has reached to the load balancer . But how will the request will do to those private subnets so there for there should be a proper route from load balancer to the private subnet . Route table will define u the route .

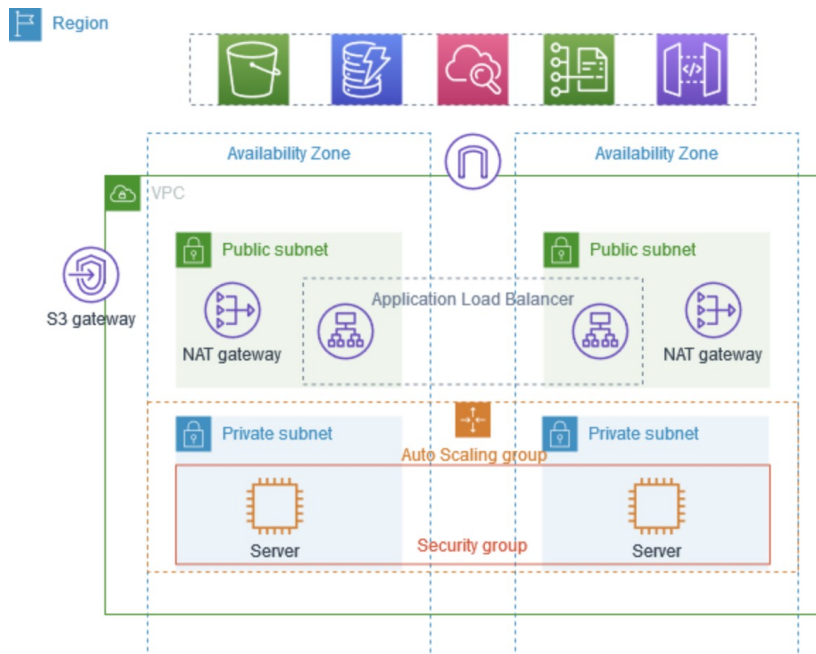
And last is the security group which is the gate to the private subnets .

NACL are the automation for security group

NAT gateway , the request has some how reached the private subnet , but if the private subnet wants to access something from the internet >

If that is a private subnet it is a bad practice to expose our server ip address to the internet .

To avoid this u need to do the masking of ip addresses of private subnet -> by converting it to the public ipaddress of load balancer or router .



VPC Flow Logs

A flow log captures information about the IP traffic going to and from network interfaces in your VPC.

AWS Security Group and NACL:

VPC is the one who introduced virtual private cloud in the world of public cloud .

Which added lots of security ,

When creating u should define the ip address range of your VPC , how many ip address u want to use inside your VPC (network) , private subnet can not be access from the internet . Load balance is directly access able the outside world , load balance has the access to the private subnets , for each subnet u can add more security in private subnet .

Network Access Control List (NACL)

A Network Access Control List is a stateless firewall that controls inbound and outbound traffic at the subnet level. It operates at the IP address level and can allow or deny traffic based on rules that you define. NACLs provide an additional layer of network security for your VPC.(subnet lvl security)

But if u add the security at the EC2 instance level , its called security group .

In AWS security is a shared responsibility , AWS says to the company that we will tell u to use VPC , u can add security groups , u can add NACL , API gate way along that we will need the help of devOps or AWS engineers ,

All these security things DevOps engineers plays a very critical role to configure all these .

aws Services Search [Option+S] Mumbai admin @ 3397-1273-

Select a VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

Security

- Network ACLs
- Security groups

VPC > Network ACLs > acl-0bb0d9c5e3c92b3e3

acl-0bb0d9c5e3c92b3e3

Actions

Details Info

Network ACL ID acl-0bb0d9c5e3c92b3e3	Associated with 4 Subnets	Default Yes	VPC ID vpc-06730c86f73e22426 / demo vpc-vpc
Owner 339712731626			

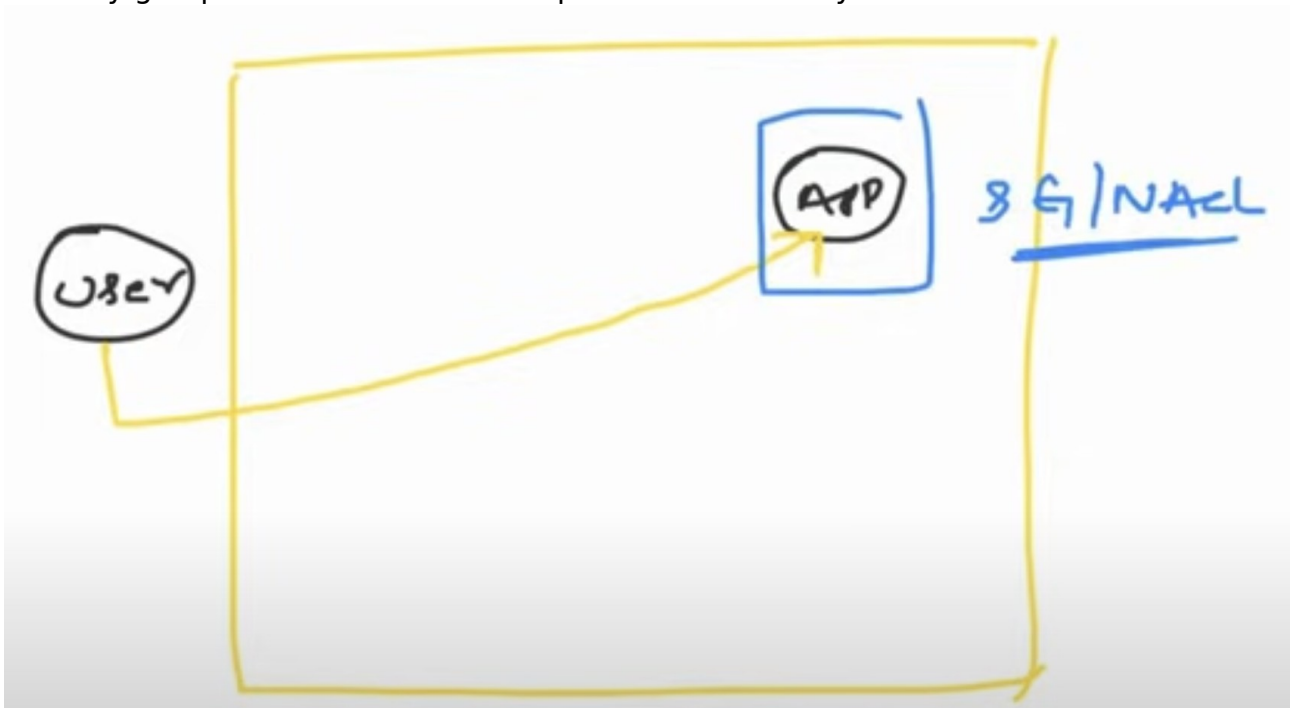
Inbound rules Outbound rules Subnet associations Tags

Inbound rules (2) Edit inbound rules

Filter inbound rules

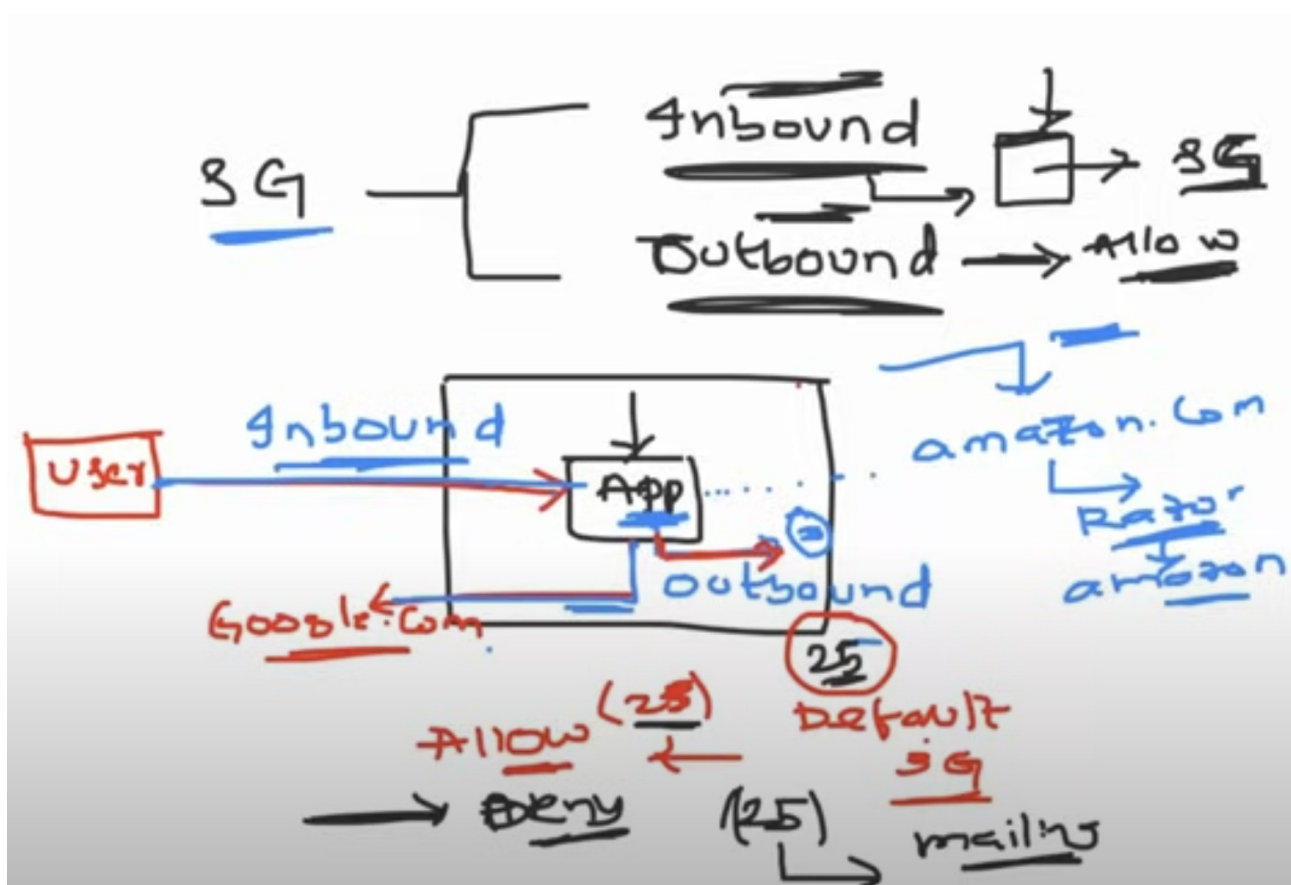
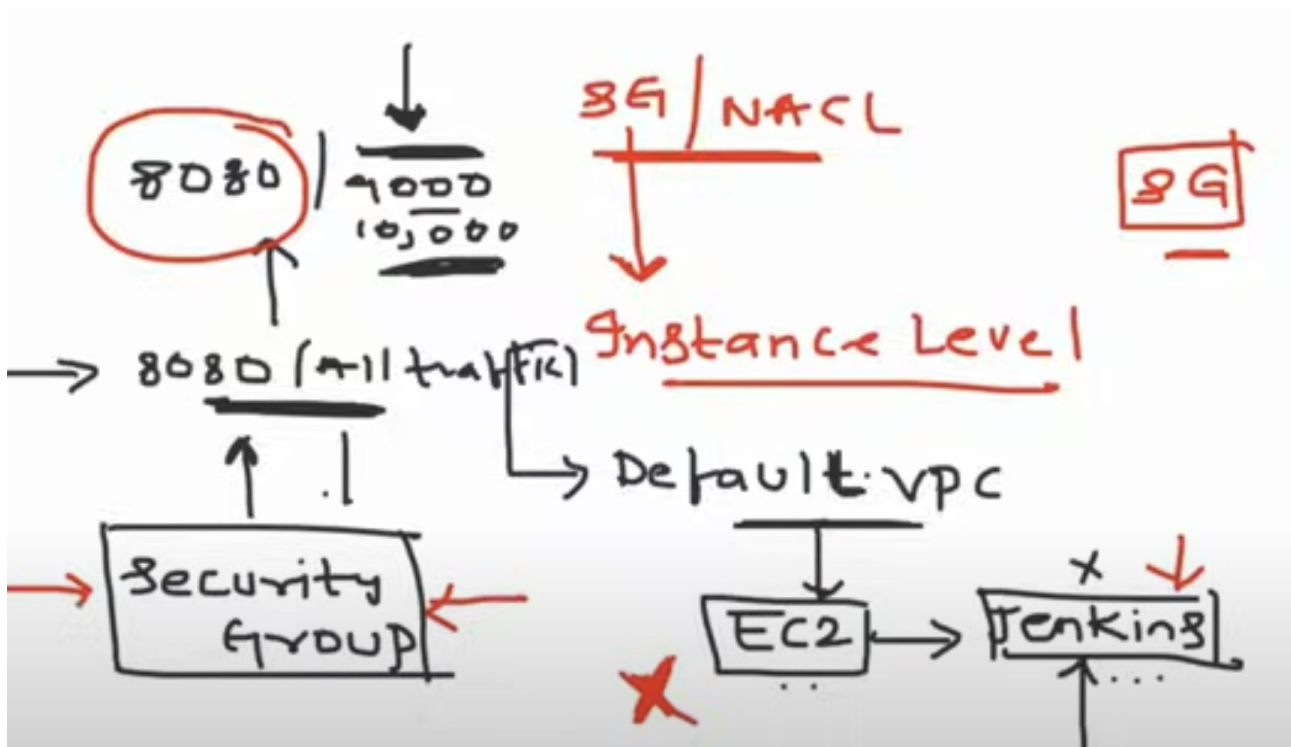
Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Security group and NACL are the last point of the security .



By default AWS will give u the default VPC ,

If any thing we need to configure in instance level we need a security group
 AWS is doing its part by default it is not accepting any traffic on Jenkins application .
 And as a devOps engineers we need to to our part by opening the inbound traffic for
 Jenkins port 8080 . And allowed traffic to enter your application . This is done by
 security group in AWS .

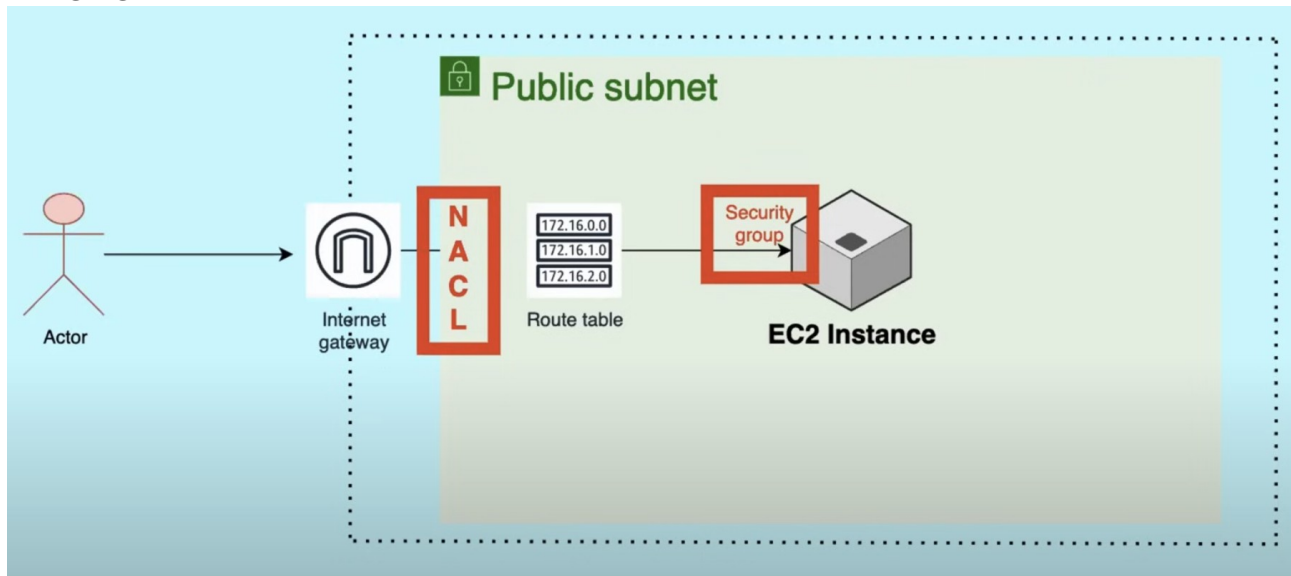


By default security group will allow all the outbound activity Except port(25) .
 Port 25 is for mailing services to prevent the spams .
 And inbound traffic is denied by default by security groups .

Security group is applied in the EC2 instance level .

NACL is applied in the subnet level what kind of traffic u want to deny + allow
Where as security group is only for what kind of traffic u want to allow

PRACTICLE :



This box is a VPC(range of ip addresses) AWS will provide by default Internet gateway , NACL , route table.

Create VPC on AWS and choose VPC and more this will give u by default resources for u like public private subnets , route table , internet gateway

Now after creating vPc create a EC2 instance -> in network configuration choose the demo-vpc .

-> choose public subnet region (don't forget to attach key pair).

Login to created EC2 instance and update the packages .

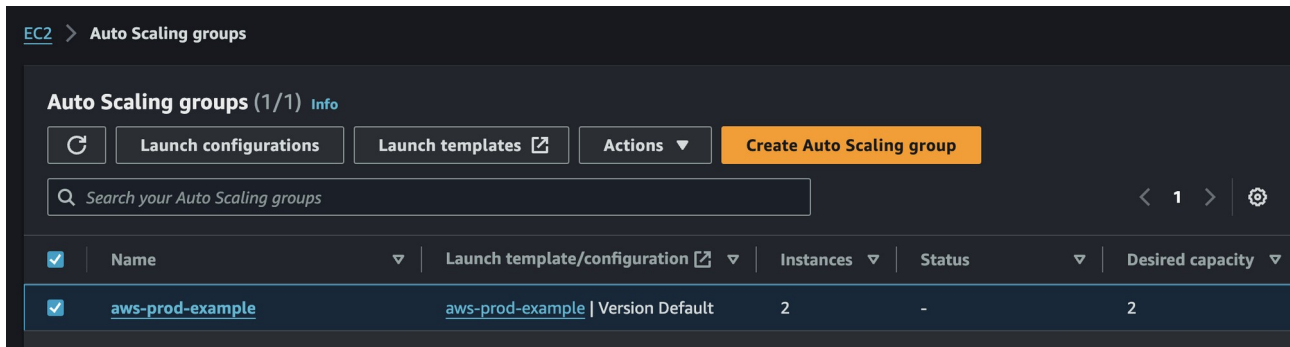
Check if the python is installed : `python3`

: run the http server : using -> `python3 -m http.server 8000`

But u will not able able to access this -> as u have only inbound for ssh key pair in security group of login to the instance on port 22 now u also need to configure same for 8000.

NACL which is the first layer of defence allow all the inbound traffic in our instance we can see inside our VPC .

Here least now will be verified first ..



What NACL did here from the Internet gateway u get entry to the subnet NACL is allowing all the traffic and forward to the route table but why request is not going to the EC2 instance because security group is blocking it .

So go to the security group -> add port 8000 inbound traffic , custom TCP , allow ipv4 . Now if u can access for 8000 via your instance .

For my organisation or this particular VPC I will block port 8000 in the NACL -> instead of allowing all the traffic remove that rule and add new rule

Endpoint services	Rule number	Type	Protocol	Port range	Source	Allow/Deny
NAT gateways	100	Custom TCP	TCP (6)	8000	0.0.0.0/0	Deny
Peering connections	*	All traffic	All	All	0.0.0.0/0	Deny

Now request will not reach that 8000 port application because u as a devOps engineer has blocked that port in NACL it self . It will be denied for the entire subnet wether u are EC2 instance or any thing

One more new thing : make new rule in NACL =>

Rule 1 = 100 allow all
Rule 2 = 200 , 8000 deny

So this will allow the access port 8000 as it follow the rule no - > ascending order .

Or u can block the ip address from a specific range from other country u can do that also here in NACL .

Note : NACL act as a first layer of defence .

ROUTE 53 :

Route 53 provide DNS as a service -> (domain name system) .

Like EC2 provide compute as a service

These application or router load balancer there get assigned with ipaddress . But we don't use ip address in real world .

DNS service is the one that maps your service to the ip address .

When u create a load balancer AWS will assign it to a ip address when ever a user want to access this application through this load balancer u can not give the ip address of this load balancer reason is 1. Ip address are hard to remember 2. Ip address can change.

It is easy to remember the domain names .

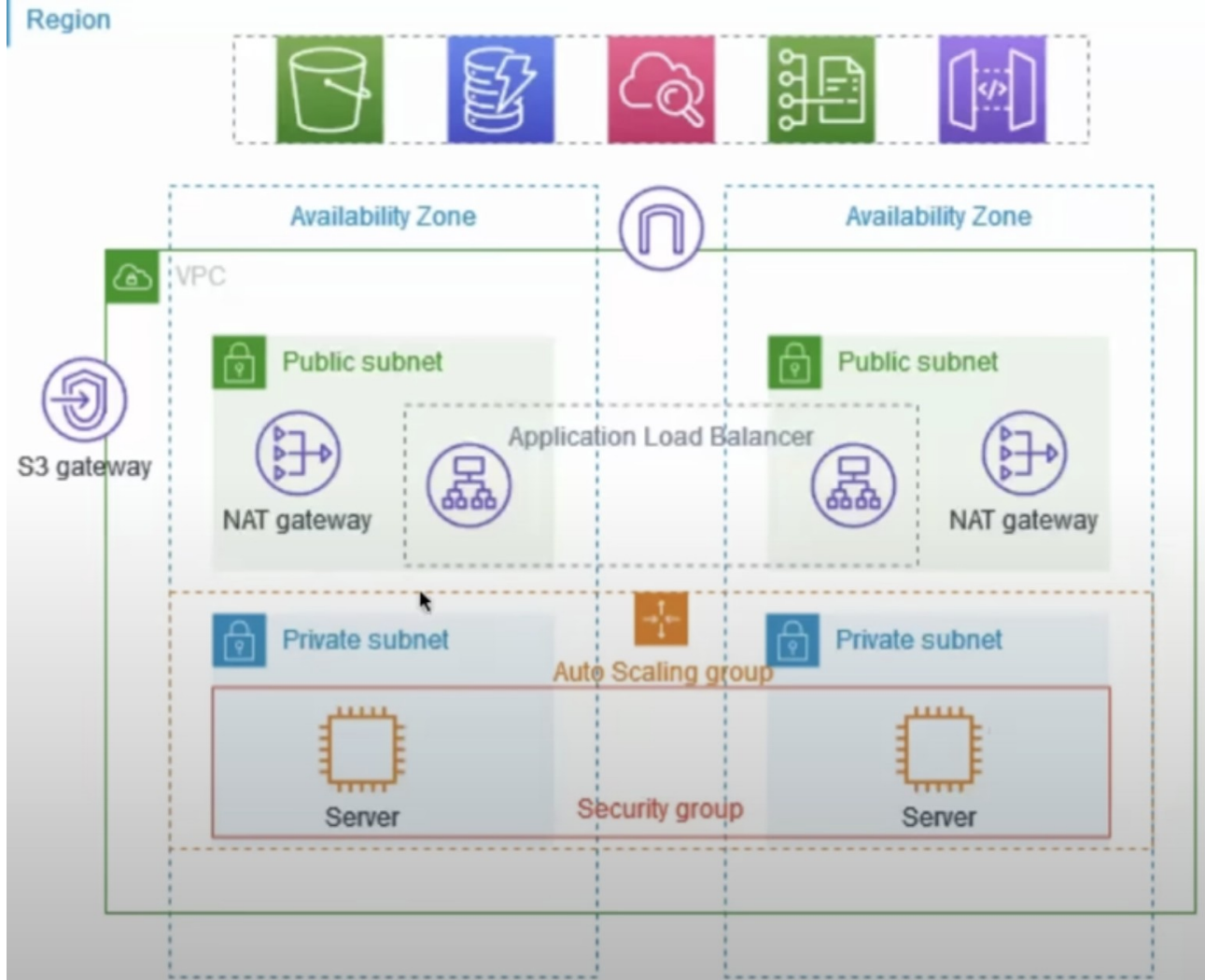
AWS says that I will provide u DNS as a service as DNS is a very complex topic . Because if there is a application u want to expose to the internet with the Domain name first thing u will do is buy a domain name.

So AWS says if u are hosting your application on AWS platform

In route 53 there is something called domain registration where u can by a domain in AWS and then in the hosted Zones u list all the DN with Ip address mapping .

AWS project :

VPC with public private subnet with production .



What is the purpose -> to improve resiliency , you deploy the servers In two availability zones , by using an auto scaling group and an application load balancer . For addition security you deploy the servers in the private subnets . The servers receive the request through the load balancer . The server can connect to the internet through the NAT gateway . To improve resiliency u deploy NAT gateway in the both availability zones .

Why u add two availability zones => so that data centre of the AZ of a specific region of AWS goes down other is still running and serving the traffic for u.

Why use NAT gate Way => so that these server can access internet to download some thing from internet and access other services from the internet this NAT gate way convert the private ip address of these server to the public one so that private ip address not get exposed to the internet . It mask the ip address of private sever

Few thing to know before :

Auto scaling group : If u want to deploy the application in two availability zone so instead of creating your EC2 instance two times what u can do u can tell

Auto scaling group to create min of two replica because if these servers are not able to handle that much of traffic it can scale your servers .

Load balancer : if two servers are there Load balancer if get 100 req it will send 50 to first server and 50 to second one .

Bastion host or jump server : instead of directly connection to the private server u can connect to a bastion so that there will be a proper login machinist u can do a proper auditing of who is access this private subnet

Start ->

1. Create VPC
2. 1 NAT gate way per AZ
3. From EC2 u can see - > Create the autoScaling group
4. Create a launch template -> so that u can use this across multiple autoscaling groups . -> add which os -> configurations -> key value pair -> create a new security group -> choose to launch in new created VPC . -> add inbound security group rule (ssh port 22 open it) source any where .

Add another security group of the application running in that instances . -> custom TCP open port 8000

Finally laugh template .

[5.it](#) is same as EC2 instance configuration because u are using security group for auto scaling EC2 instances .

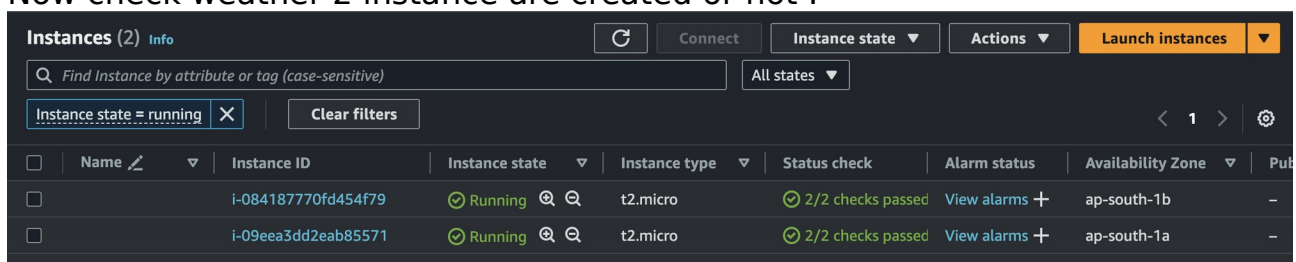
- 6 . Now go to auto scaling choose the created launch template -> next - > choose the create VPC

7. Now choose the Availability zones u want these EC2 instance to be available in the private subnet . Choose those two private subnets .

8. Not creating load balancer for his private subnets . We will create a application load balancer in the public subnet . In this autoscaling group configuration for this not creating any load balancer .

9. No scaling policies

Now check weather 2 instance are created or not .



The screenshot shows the AWS Management Console 'Instances' page. It displays two EC2 instances, both in a 'Running' state. The first instance has ID i-084187770fd454f79 and is in the ap-south-1b availability zone. The second instance has ID i-09eea3dd2eab85571 and is in the ap-south-1a availability zone. Both are t2.micro instances with 2/2 status checks passed.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>		i-084187770fd454f79	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	-
<input type="checkbox"/>		i-09eea3dd2eab85571	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-

10. Now before creating application load balancer u need to install application in these created server instances.

11. These intense don't have a public ip address of the security so who to I log in it this without public ip address thats where bastion or jump host comes into the picture.

Which is a mediator between your private subnet and the external public subnet .so I will create a bastion host in the public subnet and access the private subnet from there.

12. Creating a bastion host -> launch a EC2 instance . -> make sure u add a security group which has access to ssh because u need to ssh into the EC2

instance the bastion host and from there u need to ssh to this private subnet and install the application

And in same network setting make sure this bastion host is created in the same VPC .

13. Auto assign public ip address enable this option

14. Now once launch this instance I will ssh this instance and from there I will ssh to this private subnet . So I need to copy the key value pair to the bastion host also to access the private subnet .

```
15. (base) vishaldwivedi@vishals-MacBook-Air-2 Downloads % scp -i /Users/vishaldwivedi/Downloads/first_login.pem ubuntu@43.204.234.77:/home/ubuntu
```

```
16(base) vishaldwivedi@vishals-MacBook-Air-2 Downloads % ssh -i first_login.pem ubuntu@43.204.234.77
```

```
17. ubuntu@ip-10-0-28-146:~$ ls  
first_login.pem
```

18. Now ssh key pair is available here .

19. Now I can log in to the private subnet instance

20 copy the private ip address of one of the instance .

21 and now I am able to log in to that private subnet through the public bastion instance .in VPC .

22 . Python3 -m http.server 8000

23 .create a load balancer and attach these two instance at private subnet as private group s

24. In EC2 there are load balancer .

25. Load balancer should be internet facing should have the access of internet gateway and should be in the public subnet .

26. Choose created VPC

27. choose both availability zones. Put it in the public subnet .

28 . Security group of the load balancer u are allowing all the traffic or not .

29.add target group which EC2 instance u want to access . Choose port 8000 these are the instance I need to access on port 8000

30.now add this target group in the port 8000

31: create application load balancer . => accessing the load balancer in port 80.

32.load balancer is not accessible now because the subnet that u have attached to the load balancer goes not expose port 80 .

33. So go to the security group add inbound traffic rule-> http ,port 80 any where from internet Ip v 4 , save rules.

34 . Now in the load balancer -> copy DNS name

35: load is going to the healthy EC2 instance via a health check .

AWS interview based question :

From the above topic till now >

Q: You have been assigned to design a VPC architecture for a 2-tier application. The application needs to be highly available and scalable.

How would you design the VPC architecture?

A: In this scenario, I would design a VPC architecture in the following way.

I would create 2 subnets: public and private. The public subnet would contain the load balancers and be accessible from the internet. The private subnet would host the application servers.

I would distribute the subnets across multiple Availability Zones for high availability. Additionally, I would configure auto scaling groups for the application servers.

(Using auto scaling groups if tomorrow the req go high auto scaling groups can immediately increase your instances , availability is solved using AZ).

2 tier arch-> create a public and a private subnet in public subnet place the load balancer and in the private subnet place the 2 tier based arch)

Q: Your organization has a VPC with multiple subnets. You want to restrict outbound internet access for resources in one subnet, but allow outbound internet access for resources in another subnet. How would you achieve this?

A: To restrict outbound internet access for resources in one subnet, we can modify the route table associated with that subnet. In the route table, we can remove the default route (0.0.0.0/0) that points to an internet gateway. (So there is no traffic that can flow in or out of the subnet)

This would prevent resources in that subnet from accessing the internet. For the subnet where outbound internet access is required, we can keep the default route pointing to the internet gateway.

Q: You have a VPC with a public subnet and a private subnet. Instances in the private subnet need to access the internet for software updates. How would you allow internet access for instances in the private subnet?

A: To allow internet access for instances in the private subnet, we can use a NAT Gateway or a NAT instance.

We would place the NAT Gateway/instance in the public subnet and configure the private subnet route table to send outbound traffic to the NAT Gateway/instance. This way, instances in the private subnet can access the internet through the NAT Gateway/instance.

(NAT gateway do the network address translation it will that the private subnet instance ip address and translate its ip address to the public ip address of the NAT gateway and it will send the request using masking).

Q: You have launched EC2 instances in your VPC, and you want them to communicate with each other using private IP addresses. What steps would you take to enable this communication?

A: By default, instances within the same VPC can communicate with each other using private IP addresses.

To ensure this communication, we need to make sure that the instances are launched in the same VPC and are placed in the same subnet or subnets that are connected through a peering connection or a VPC peering link.

Additionally, we should check the security groups associated with the instances to ensure that the necessary inbound and outbound rules are configured to allow communication between them.

(If they are in the same subnet then they will be in the same ip -address range and CIDR block)(vPc pairing one subnet is in one vPC and other subnet is in other VPC)

Q: You want to implement strict network access control for your VPC resources. How would you achieve this?

A: To implement granular network access control for VPC resources, we can use Network Access Control Lists (ACLs).

NACLs are stateless and operate at the subnet level. We can define inbound and outbound rules in the NACLs to allow or deny traffic based on source and destination IP addresses, ports, and protocols.

By carefully configuring NACL rules, we can enforce fine-grained access control for traffic entering and leaving the subnets.

(Security group if for the subnet level security and naCL in subnet level)

Q: Your organization requires an isolated environment within the VPC for running sensitive workloads. How would you set up this isolated environment?

A: To set up an isolated environment within the VPC, we can create a subnet with no internet gateway attached.

This subnet, known as an "isolated subnet," will not have direct internet connectivity. We can place the sensitive workloads in this subnet, ensuring that they are protected from inbound and outbound internet traffic.

However, if these workloads require outbound internet access, we can set up a NAT Gateway or NAT instance in a different subnet and configure the isolated subnet's route table to send outbound traffic through the NAT Gateway/instance.

Q: Your application needs to access AWS services, such as S3 securely within your VPC. How would you achieve this?

A: To securely access AWS services within the VPC, we can use VPC endpoints. VPC endpoints allow instances in the VPC to communicate with AWS services privately, without requiring internet gateways or NAT gateways.

We can create VPC endpoints for specific AWS services, such as S3 and DynamoDB, and associate them with the VPC.

This enables secure and efficient communication between the instances in the VPC and the AWS service

Q: What is the difference between NACL and Security groups ? Explain with a use case ?

A: For example, I want to design a security architecture, I would use a combination of NACLs and security groups. At the subnet level, I would configure NACLs to enforce inbound and outbound traffic restrictions based on source and destination IP addresses, ports, and protocols. NACLs are stateless and can provide an additional layer of defense by filtering traffic at the subnet boundary.

At the instance level, I would leverage security groups to control inbound and outbound traffic. Security groups are stateful and operate at the instance level. By carefully defining security group rules, I can allow or deny specific traffic to and from the instances based on the application's security requirements.

By combining NACLs and security groups, I can achieve granular security controls at both the network and instance level, providing defense-in-depth for the sensitive application.

Q: What is the difference between IAM users, groups, roles and policies ?

A: IAM User(authentication): An IAM user is an identity within AWS that represents an individual or application needing access to AWS resources. IAM users have permanent long-term credentials, such as a username and password, or access keys (Access Key ID and Secret Access Key). IAM users can be assigned directly to IAM policies or added to IAM groups for easier management of permissions.

IAM Role: An IAM role is similar to an IAM user but is not associated with a specific individual. Instead, it is assumed by entities such as IAM users, applications, or services to obtain temporary security credentials. IAM roles are useful when you want to grant permissions to entities that are external to your AWS account or when you want to delegate access to AWS resources across accounts. IAM roles have policies attached to them that define the permissions granted when the role is assumed.

IAM Group: An IAM group is a collection of IAM users. By organizing IAM users into groups, you can manage permissions collectively. IAM groups make it

easier to assign permissions to multiple users simultaneously. Users within an IAM group inherit the permissions assigned to that group. For example, you can create a "Developers" group and assign appropriate policies to grant permissions required for developers across your organization.

IAM Policy(authorisation): An IAM policy is a document that defines permissions and access controls in AWS. IAM policies can be attached to IAM users, IAM roles, and IAM groups to define what actions can be performed on which AWS resources. IAM policies use JSON (JavaScript Object Notation) syntax to specify the permissions and can be created and managed independently of the users, roles, or groups. IAM policies consist of statements that include the actions allowed or denied, the resources on which the actions can be performed, and any additional conditions.

Q: You have a private subnet in your VPC that contains a number of instances that should not have direct internet access. However, you still need to be able to securely access these instances for administrative purposes. How would you set up a bastion host to facilitate this access?

(It is a path from public subnet to the private subnet to access private subnet)

A: To securely access the instances in the private subnet, you can set up a bastion host (also known as a jump host or jump box). The bastion host acts as a secure entry point to your private subnet. Here's how you can set up a bastion host:

Create a new EC2 instance in a public subnet, which will serve as the bastion host. Ensure that this instance has a public IP address or is associated with an Elastic IP address for persistent access.

Configure the security group for the bastion host to allow inbound SSH (or RDP for Windows) traffic from your IP address or a restricted range of trusted IP addresses. This limits access to the bastion host to authorized administrators only.

Place the instances in the private subnet and configure their security groups to allow inbound SSH (or RDP) traffic from the bastion host security group.

SSH (or RDP) into the bastion host using your private key or password. From the bastion host, you can then SSH (or RDP) into the instances in the private subnet using their private IP addresses.