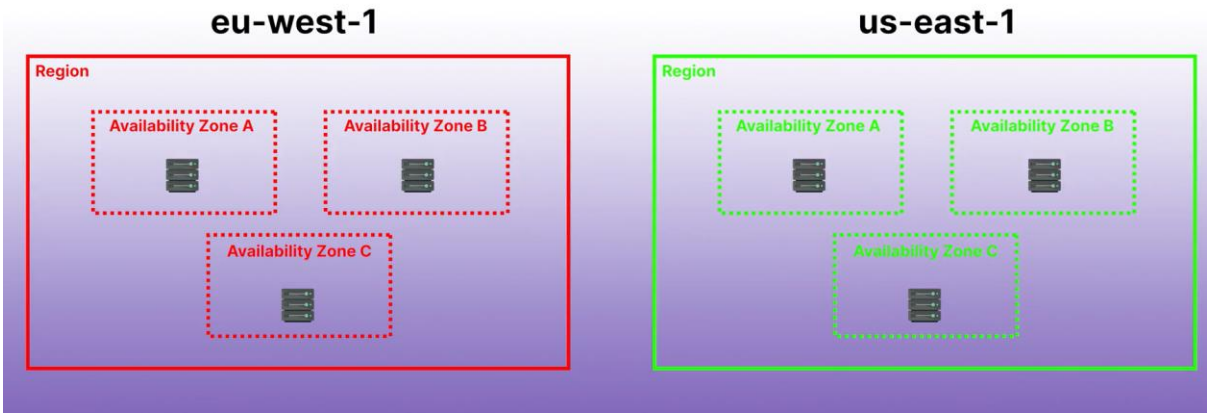
Week 5. AWS Fundamentals

제 1강. AWS Global Infrastructure

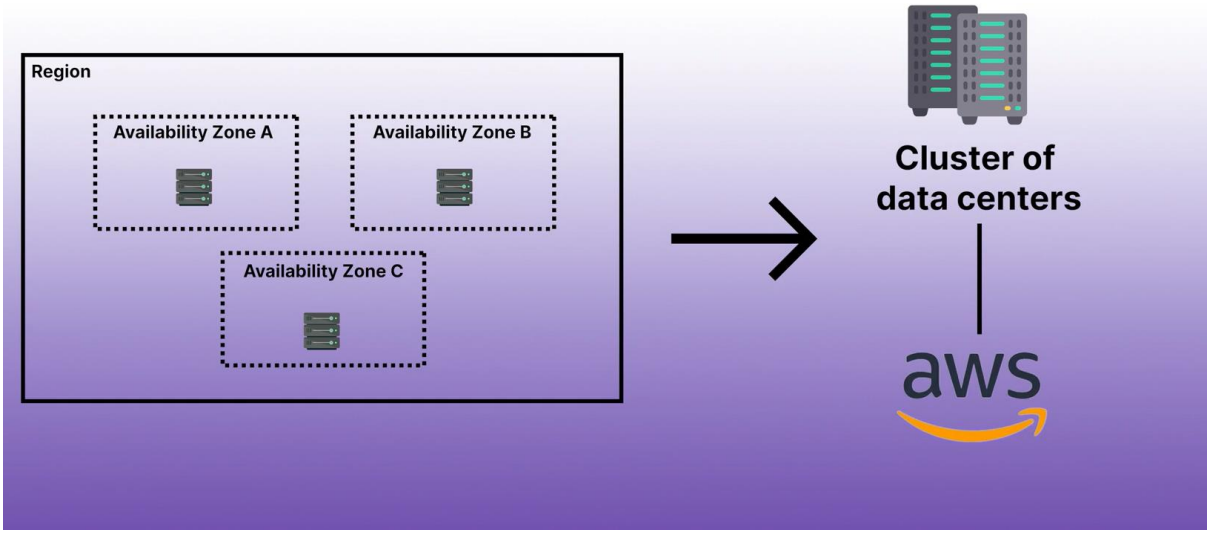
- AWS Global Infrastructure enables highly-available, scalable and reliable serivces to millions of users in the worldwide.
- Digital services and applications running on AWS must run in 24 hours no matter where you are.

1. AWS regions

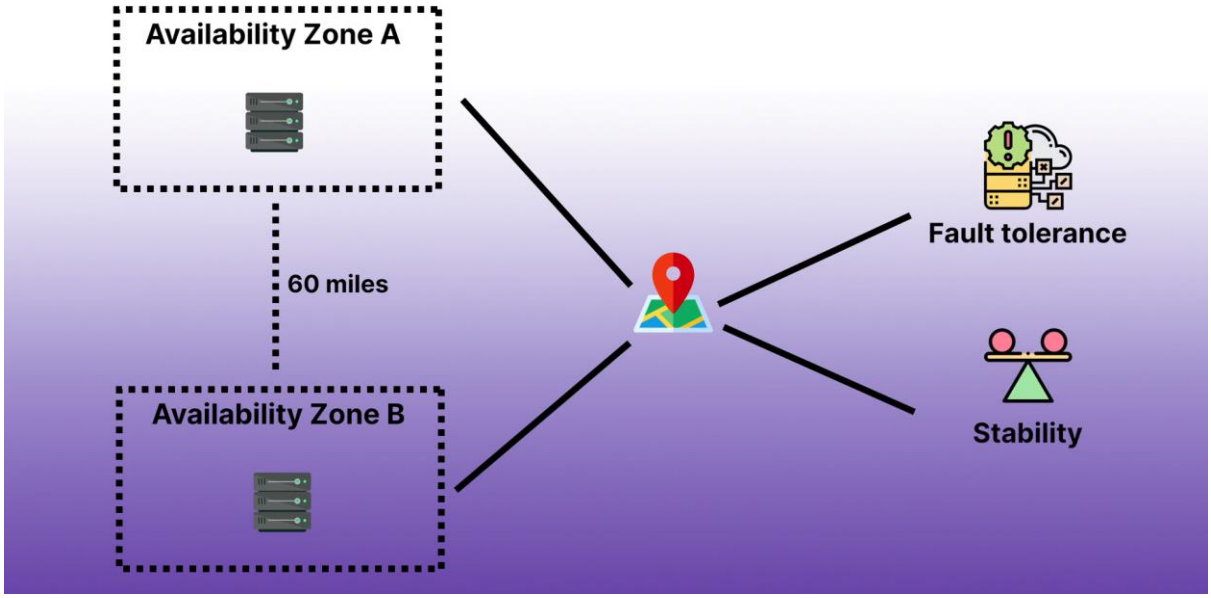
- AWS regions are global network of power stations, serving a specific geographical area which are independent one another. For example, eu-west-1 region does not affect us-east-1.



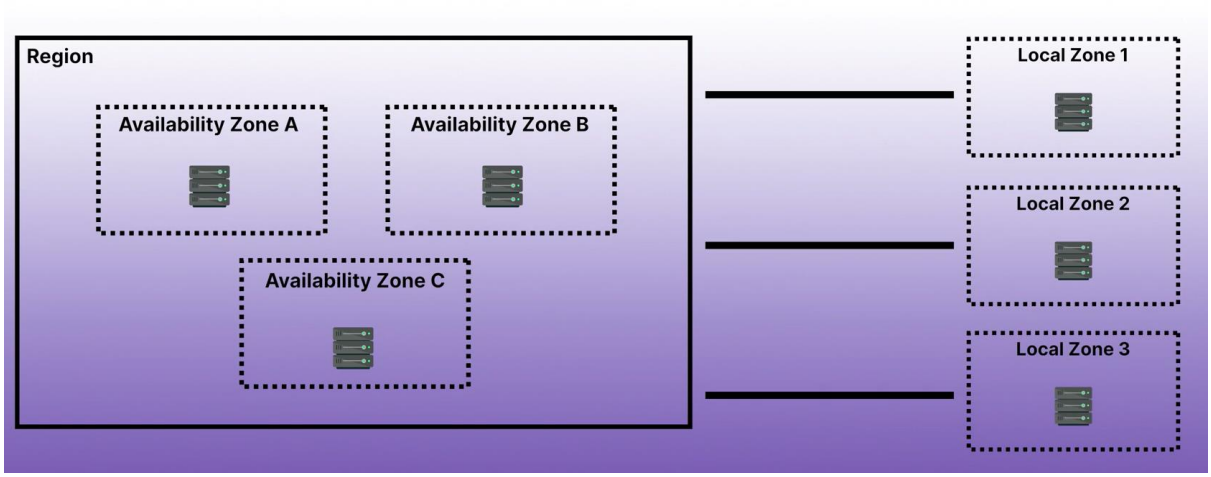
- This independence enables AWS to provide resilient services for Netflix.
- Each region is composed of the cluster of data centers.



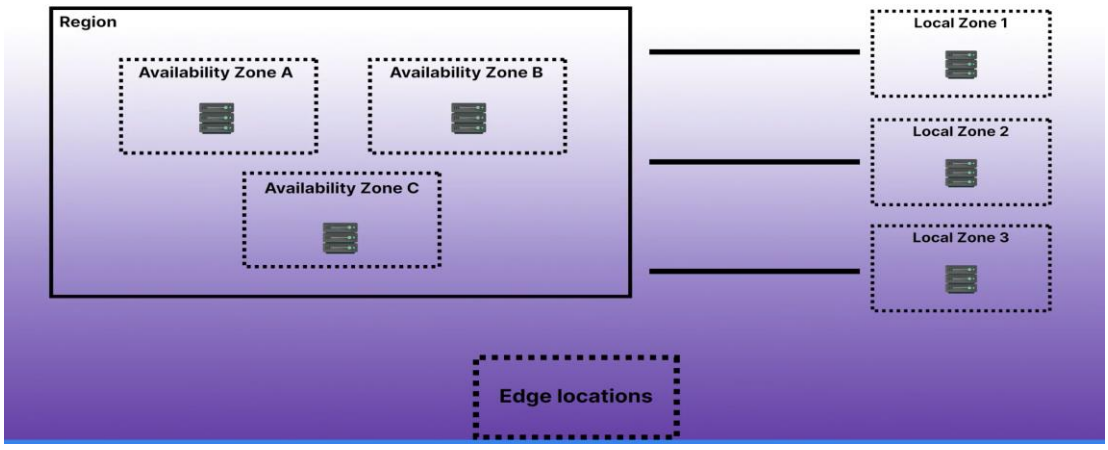
- There are multiple availability zones which are power generators within a power station. Each AZ has its own data centers with redundant power, networking and connectivity, allowing for seamless and reliable services, even if there is any issue.
- Availability zones are robustic engines operating a power station, even if one engine encounters a problem.
- Each AZ is 60 miles apart from each other, providing geographical dispersions for fault tolerance and stability.



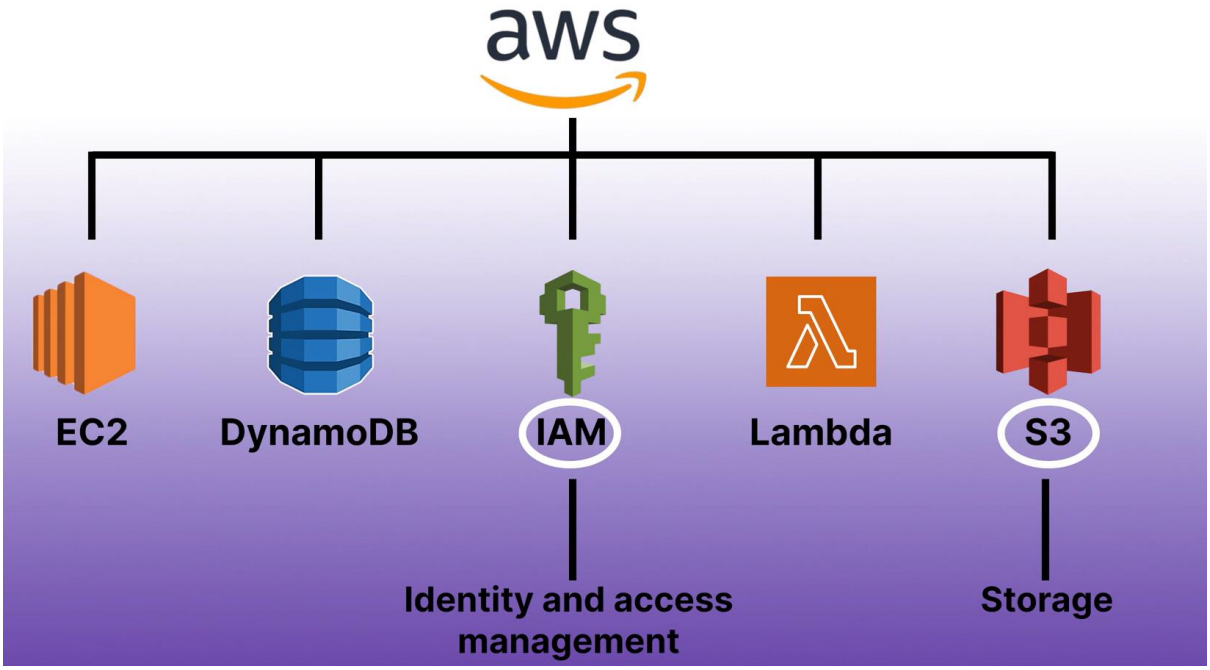
- We also have smaller and localized local zones, which are kinds of power stations extended from the main power station, region to bring services closer to users.



- Local zones are desgined for latency-sensitivie applications, ensuring users to access core AWS services fast and reliably.
- AWS also has edge locations, which is a kinds of final transformer and distribution line of electricity for homes or businesses.



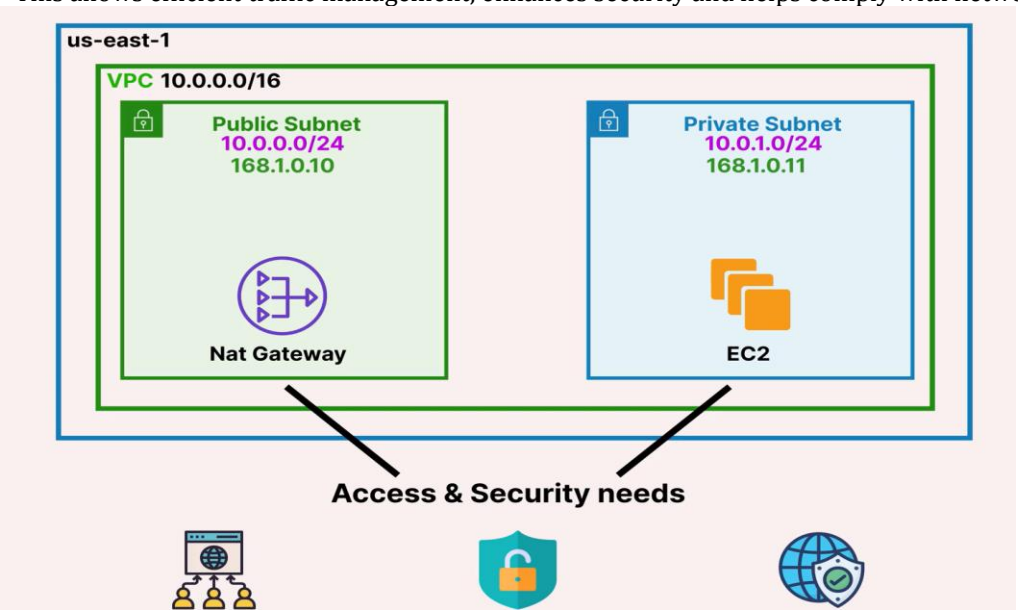
- Edge locations cache data closer to users for even lower latency, related to Amazon CloudFront, which is the AWS's last-mile delivery for users, allowing for the quickest access to data as needed.



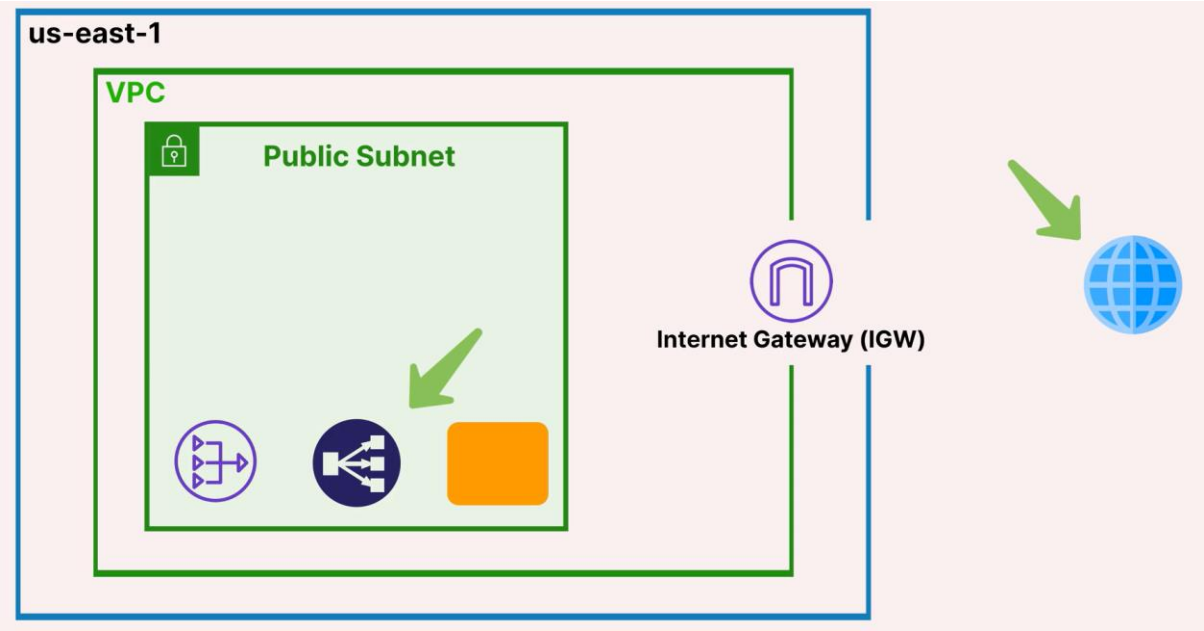
- Some AWS services are confined to specific regions but for example, IAM and S3 are global services not constrained by geographic locations.
- AWS regions have a specific naming convention such as eu-west-2, identifying a specifc geographic locations and corresponding available services.
- AWS clients are able to create an application, highly available, fault-tolerance and scaling globally by utilizing AWS global infrastructure such as regions, multi-AZs deployment, local zones and edge locations.

제 2강. Public vs Private Subnets

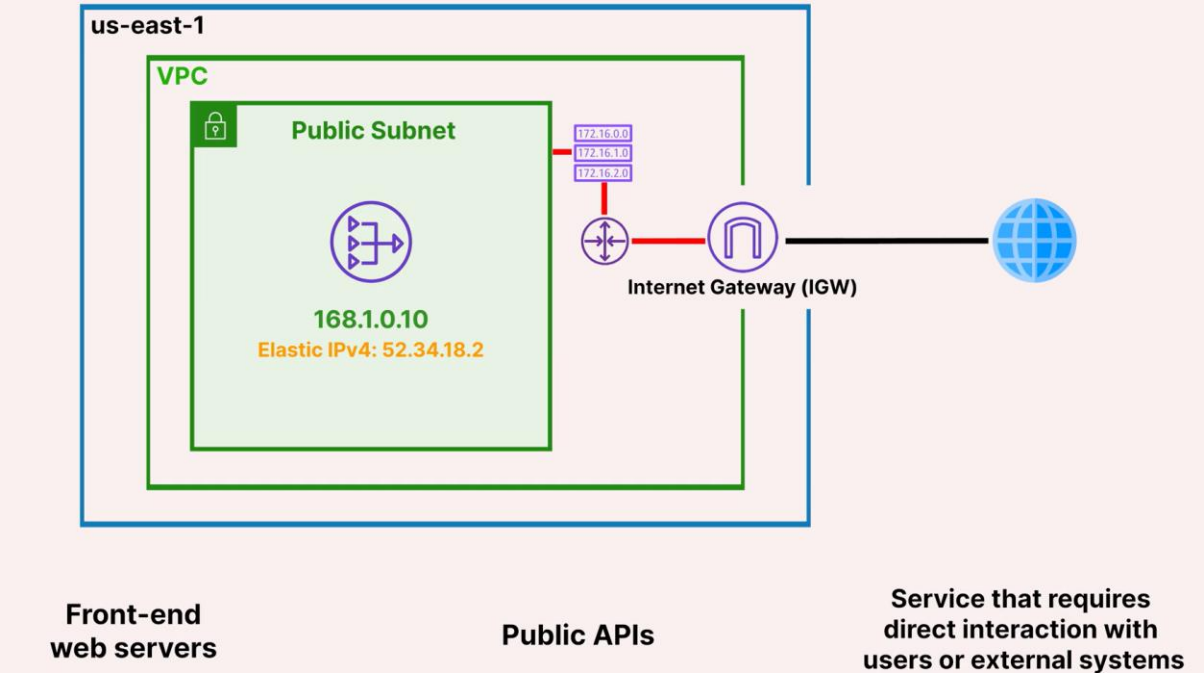
- There are public subnets and private subnets in VPC, which are designated areas serving specific roles of functions. They sub-divisions of your VPC IP address ranges, that allows segments and organize cloud resources based on varying levels of access and security needs.
- This allows efficient traffic management, enhances security and helps comply with network policy.



- The primary distinctions between public subnet and private subnet are accessibility from the Internet and role in network architecture.
(1) Public subnet
- Resources in a public subnet can directly access through the internet. This accessibility is enabled by Internet Gateway. VPC provides a route for communication between resources in the public subnet and the internet.

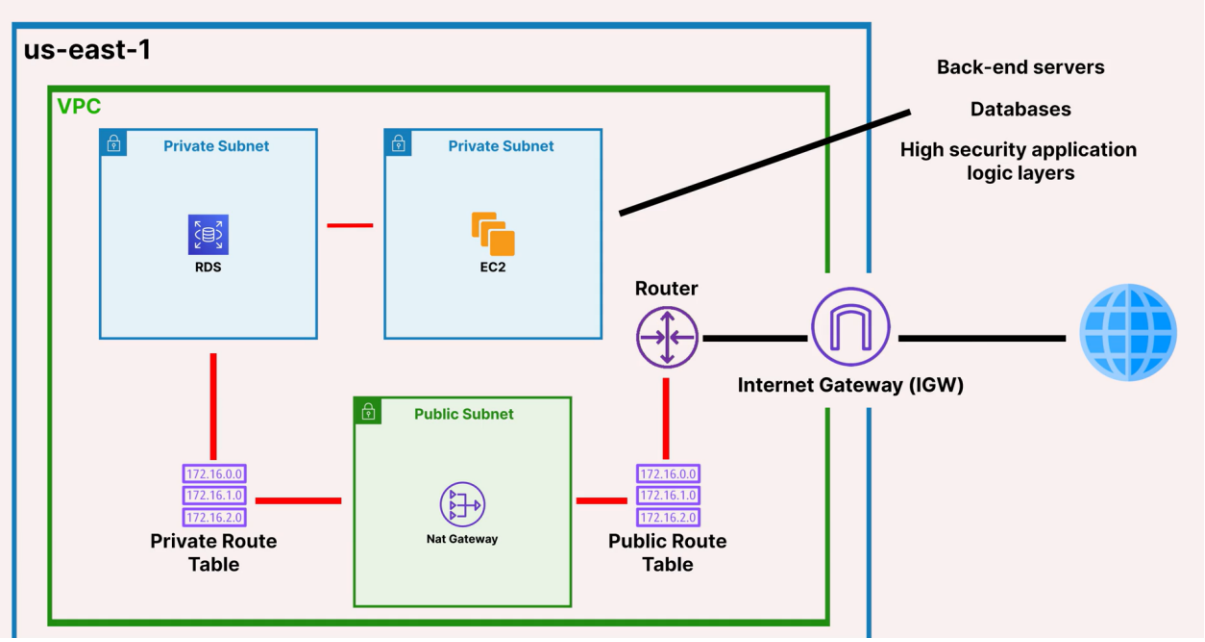


- Public IP address or Elastic IPv4 must be assigned to a public subnet or resource in the subnet in order from them to be accessed by the internet.
- This public subnet must have a route table containing a route connected to the Internet Gateway. A public subnet is ideal for front-end web servers, public APIs and service that requires direct interaction with users or external systems.



(2) Private subnet
- A private subnet is ideal for resources that should not be accessed directly from the internet.
- Resources in each private subnet and in a public subnet can communicate with each other but this private route table cannot access the Internet Gateway directly. Instead, there is a router allowing public resources

to access the Internet through a public route table.

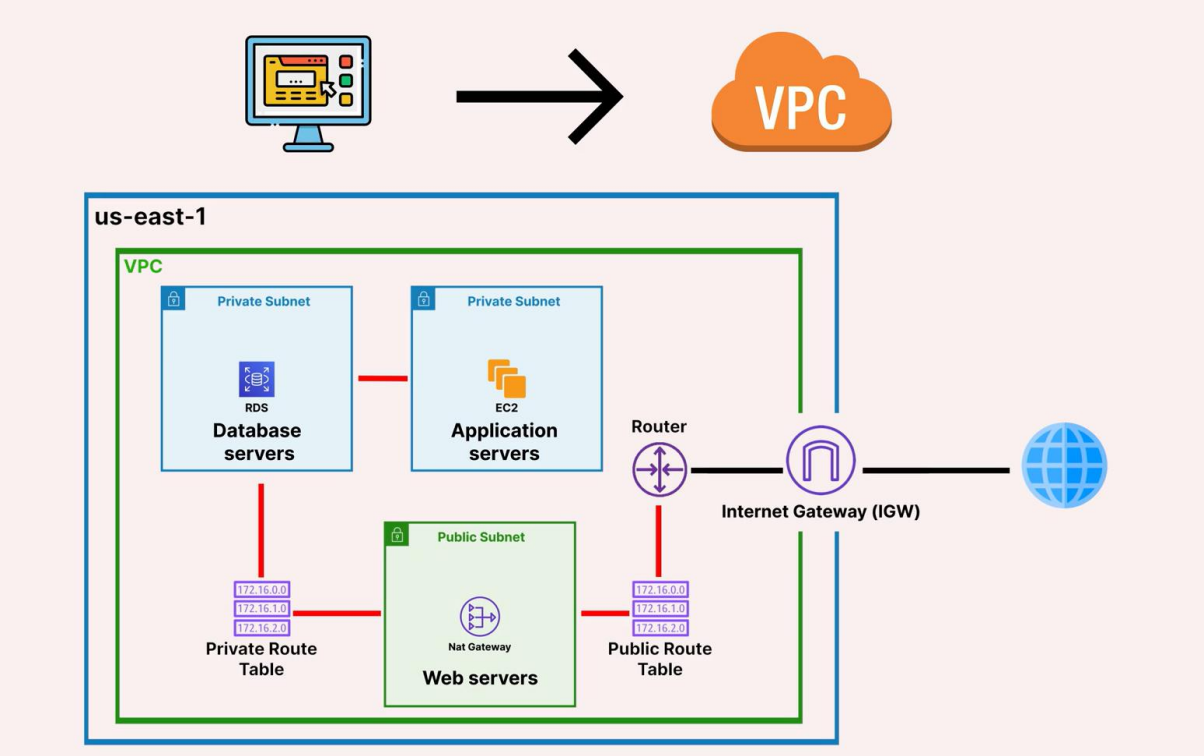


- This router only allows outbound traffic from resources but not allow any inbound traffic from the internet.
- A private subnet is used for back-end servers, databases and high-security application logic layers that can be accessed via the public subnet indirectly.

(1) Enhance security
- The organized VPC structure design is crucial to enhance security for sensitive and critical workloads and make them separate from public-facing services to prevent unauthroized access.

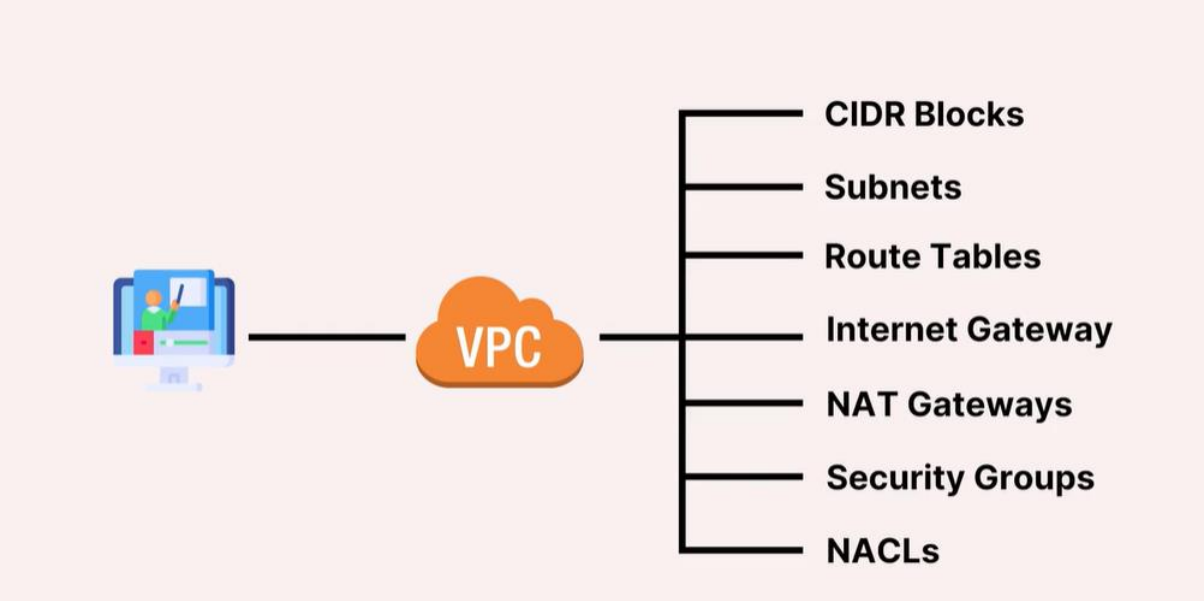
(2) Improve performance
- VPC can localize traffic to minimize latency and maximize throughput by making applications servers to communicate with closely-related services such as databases.

(3) Simplify Management
- We can adhere clear roles and policies to each subnet to facilitate the management of access controls, monitoring traffic and data governance requirements.

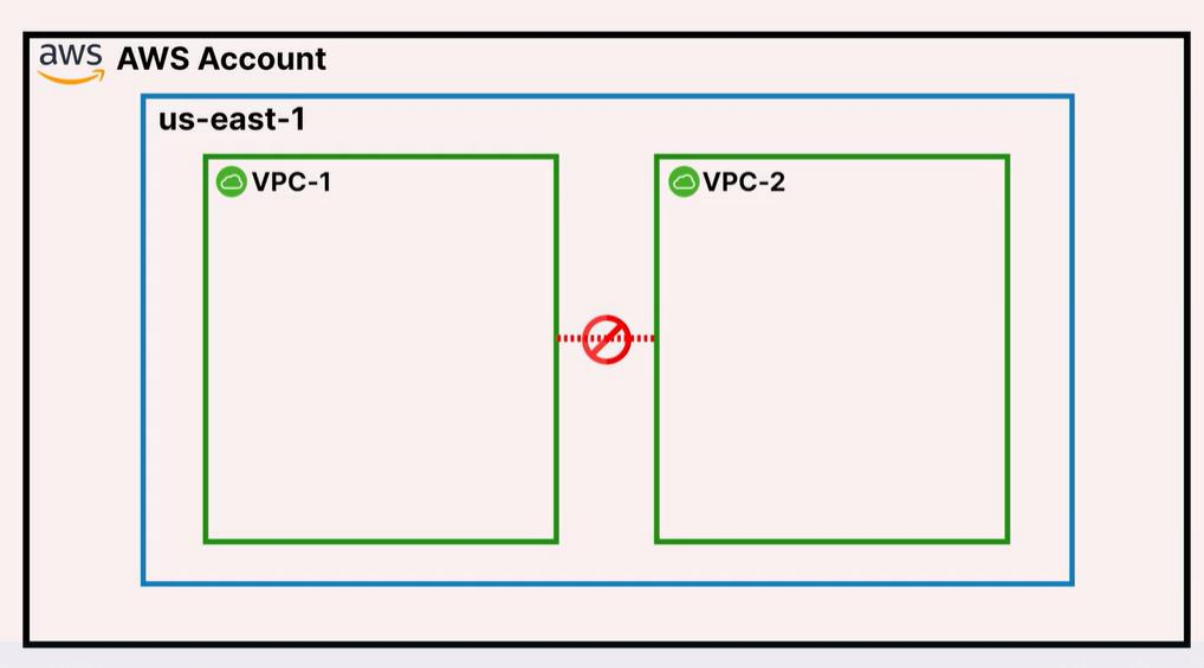


- For example, web servers are placed in a public subnet mangaing users' requests from the internet and place sensitive database servers and business logics (application servers) in private subnets to prevent unauthorized external access.

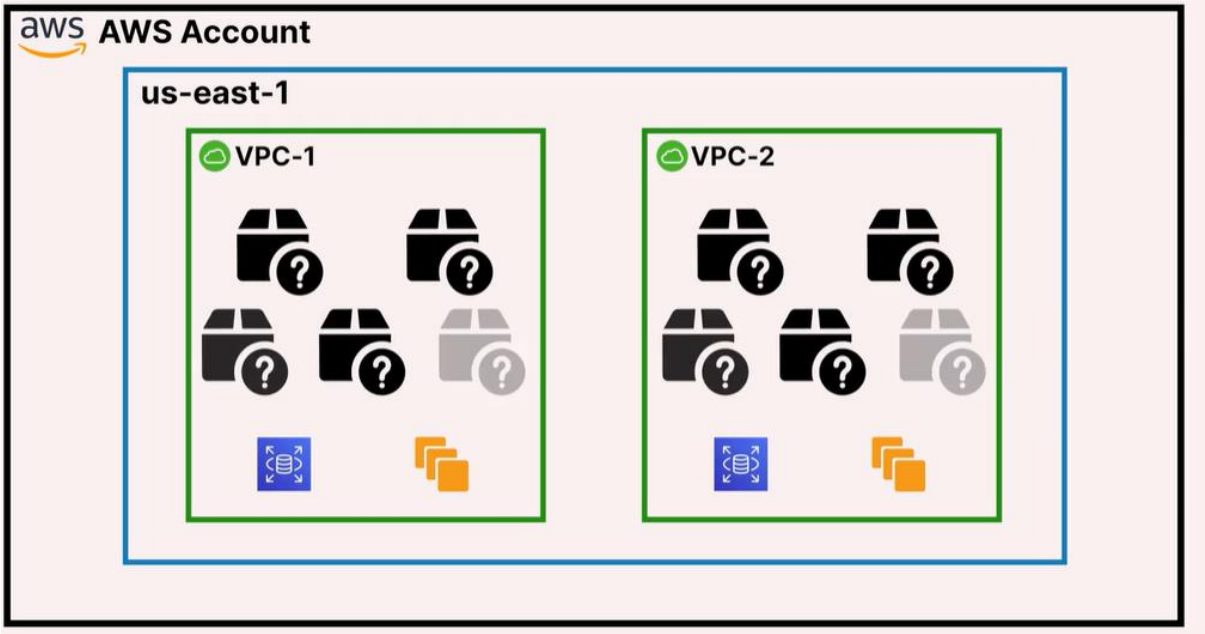
제 3강. Amazon VPC



- VPC is a core service that is provided by AWS and logically isolated private network for users and applications within AWS.
- VPC can create separate spaces within AWS cloud network where you can launch various resources in a virtual network (VPC).
- VPC is isolated from other virtual VPCs where you can fully contro AWS services such as Amazon RDS and Amazon EC2 and these services are connected via network privded by VPC.

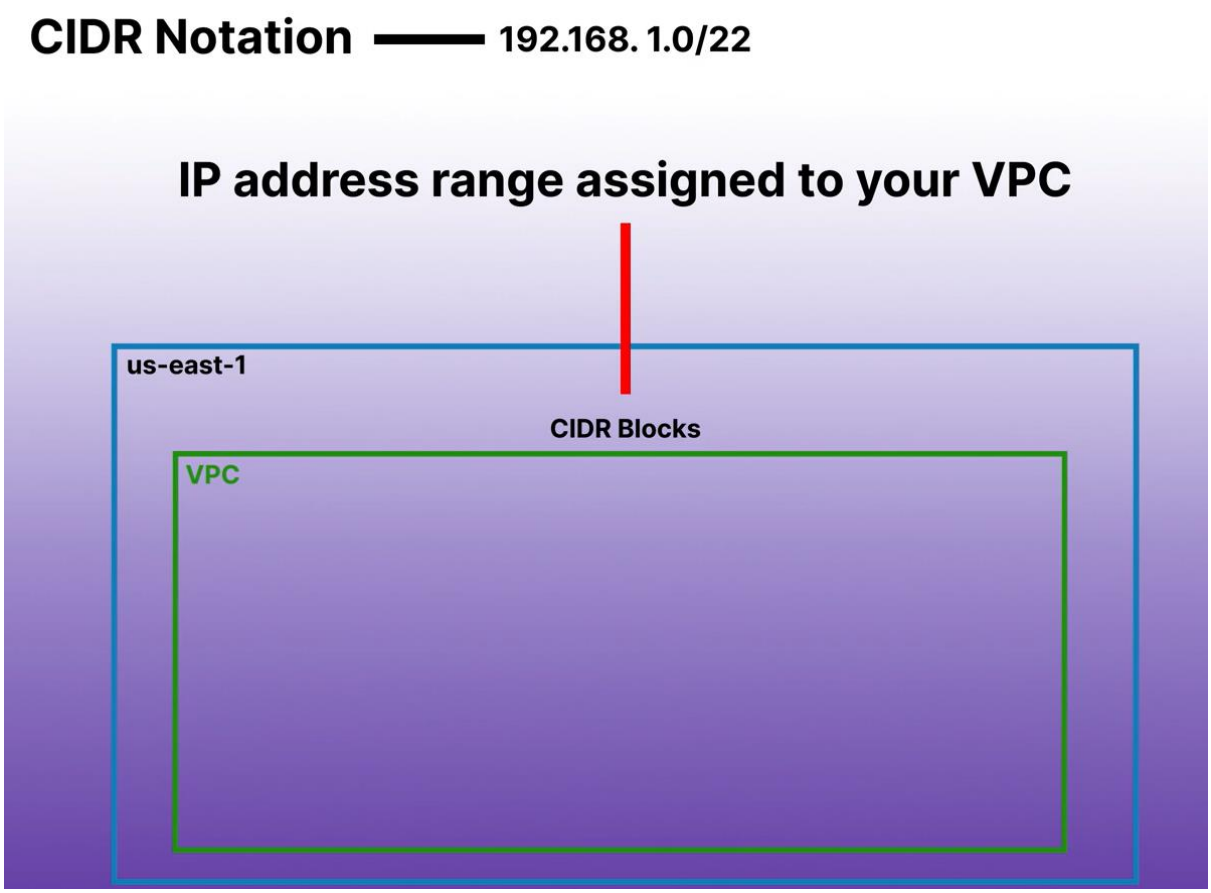


- There are a number of components within VPC.

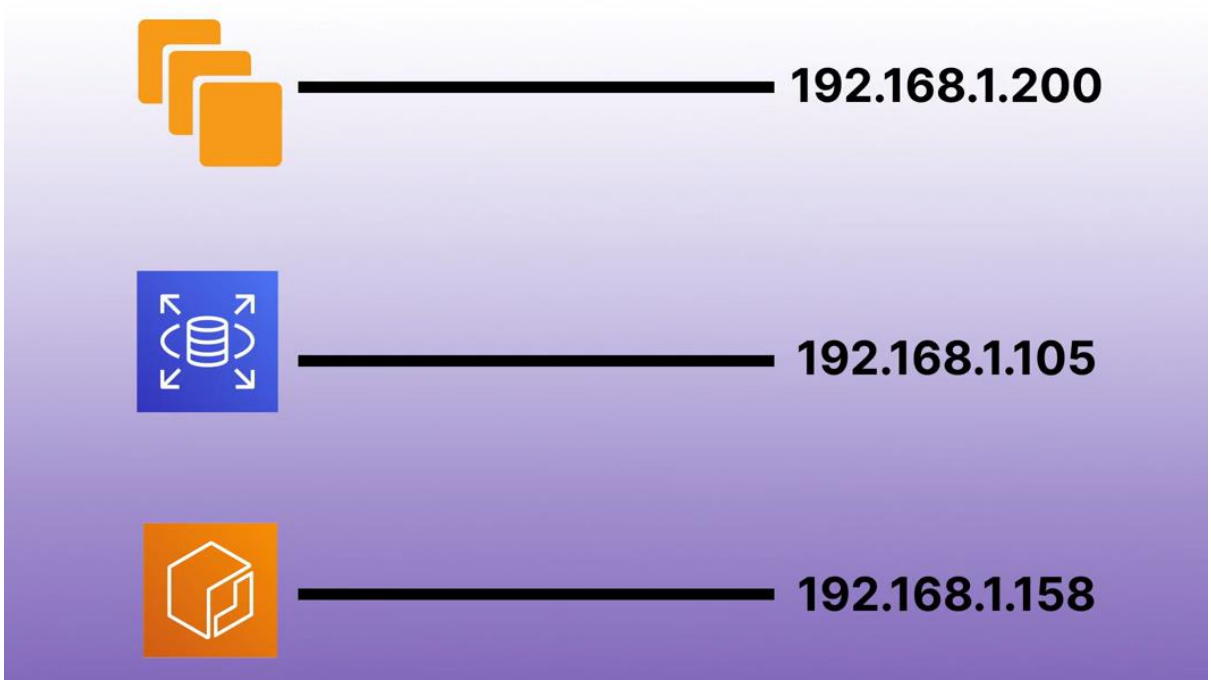


(1) CIDR Blocks

- CIDR Blocks is an IP address range assigned to your VPC. When you create a VPC, we must specify a specific IP address using CIDR notation.
- CIDR Blocks decide a IP address range and the number of resources luanched within VPC.



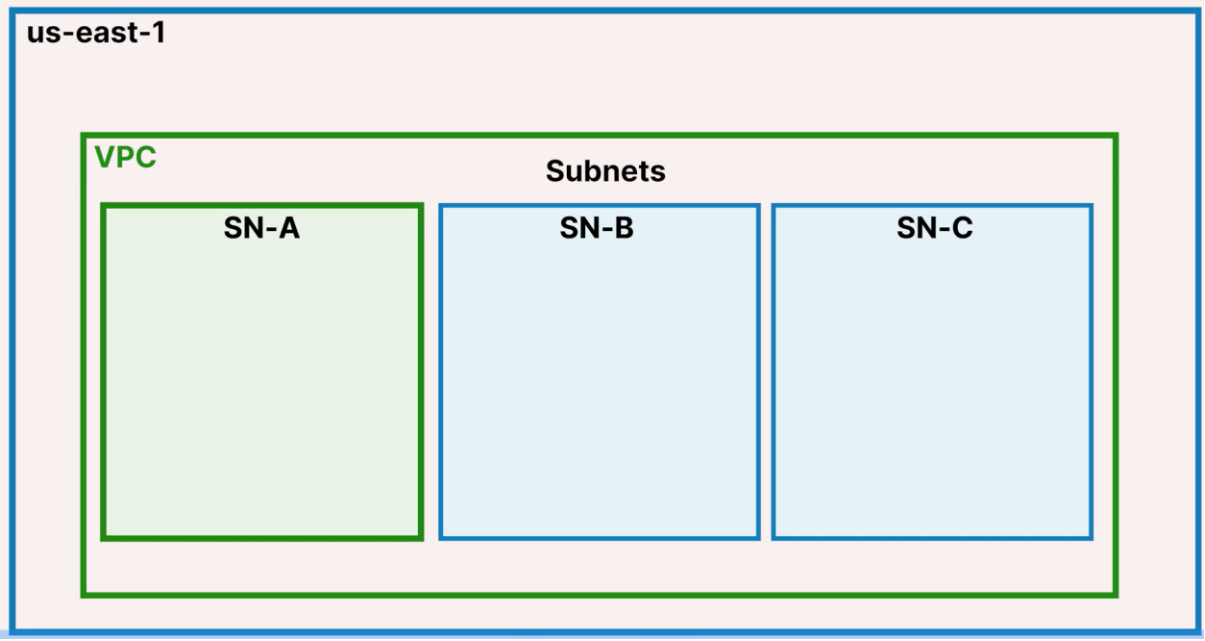
- CIDR Notation is a method of signing and managing IP addresses within a network. For instance, if you start with a VPC with a specific IP address, 192.168.1.0 and ends with 192.168.1.255. It has totally 255 IP addresses.
- Every device or resouce within a VPC such as servers, databases or containers must have each IP address within this range to ensure proper network communications and identifications.



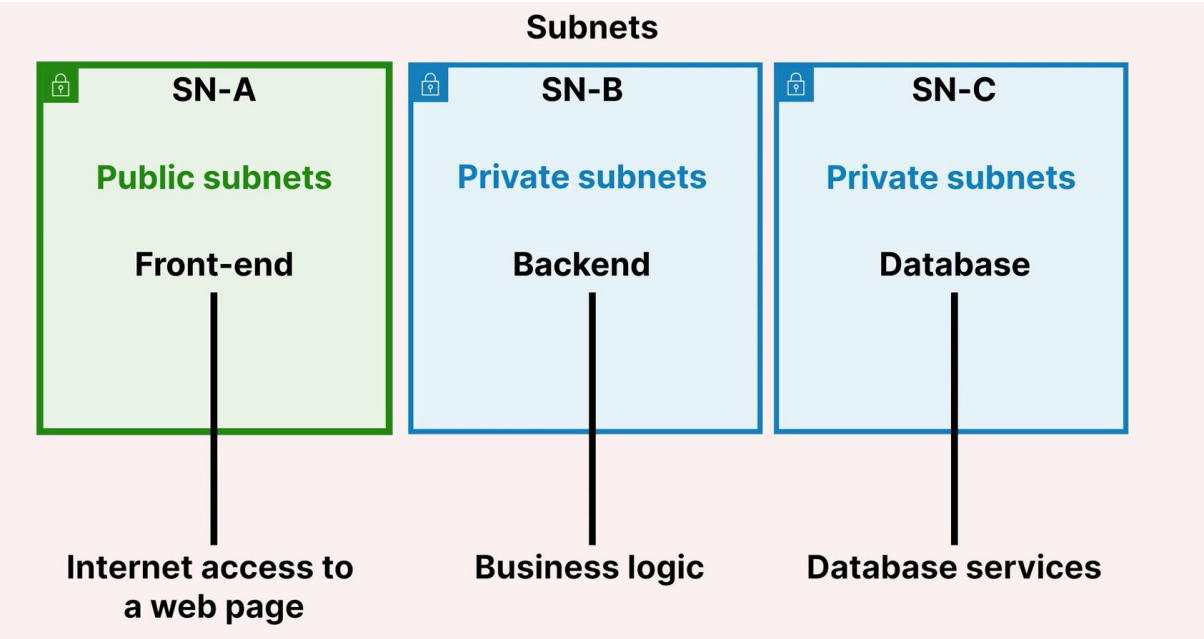
- Using this IP address range can organize a network infrastructure in a scalable and manageable way.

(2) Subnets

- Subnets divide a VPC into smaller sections. Each subnet can server a specific purpose such as public subnets and private subnets.



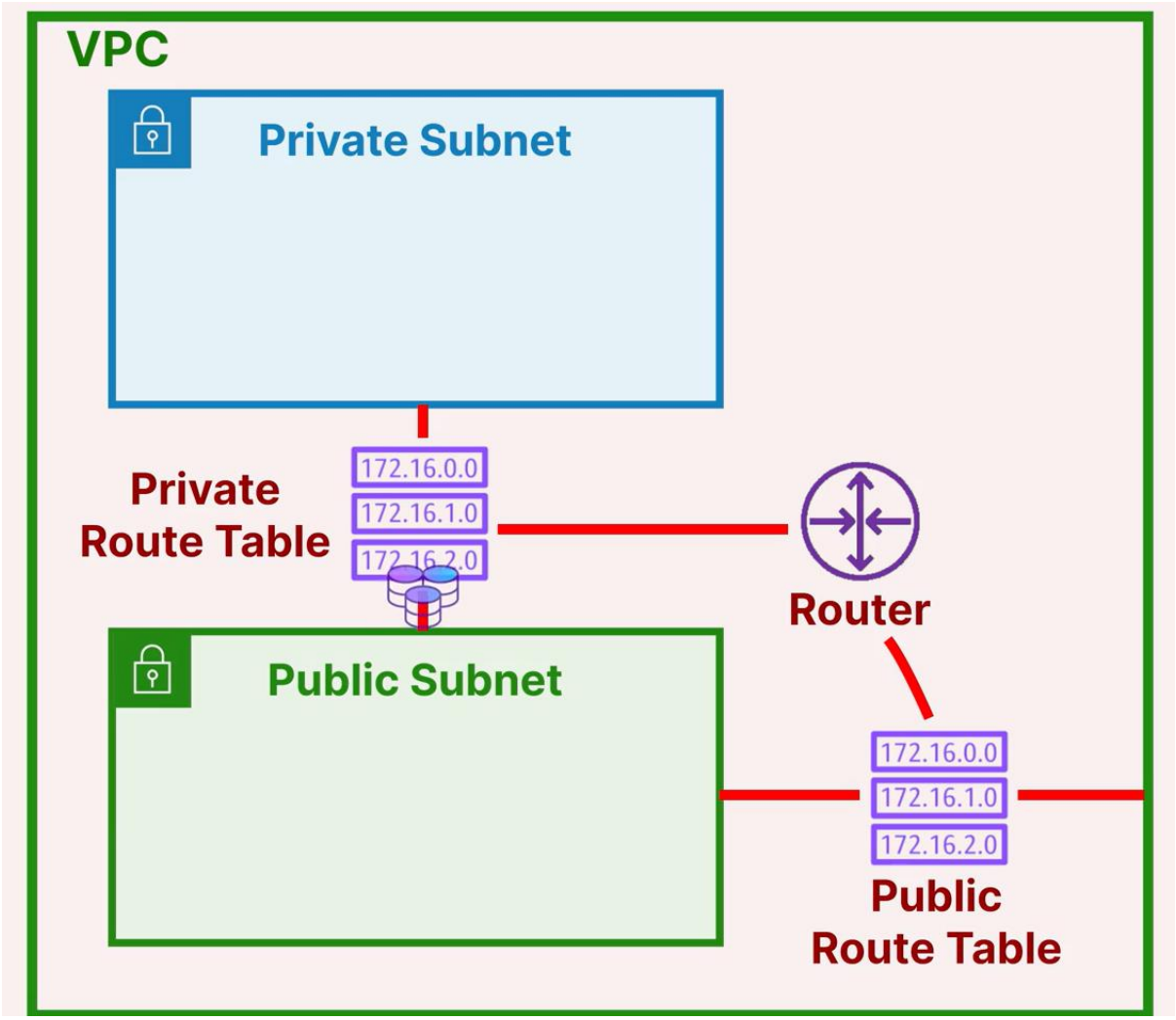
- Subnets serve different purposes. For example, public subnets server front-end for the internet access to a web page. Private subnets serve back-end servers such as business logics and database for database services.



- Subnets isolate public resources from private resources. Most AWS services can be served by private subnets. You can only use public subnets under controlled access and only when it is needed.

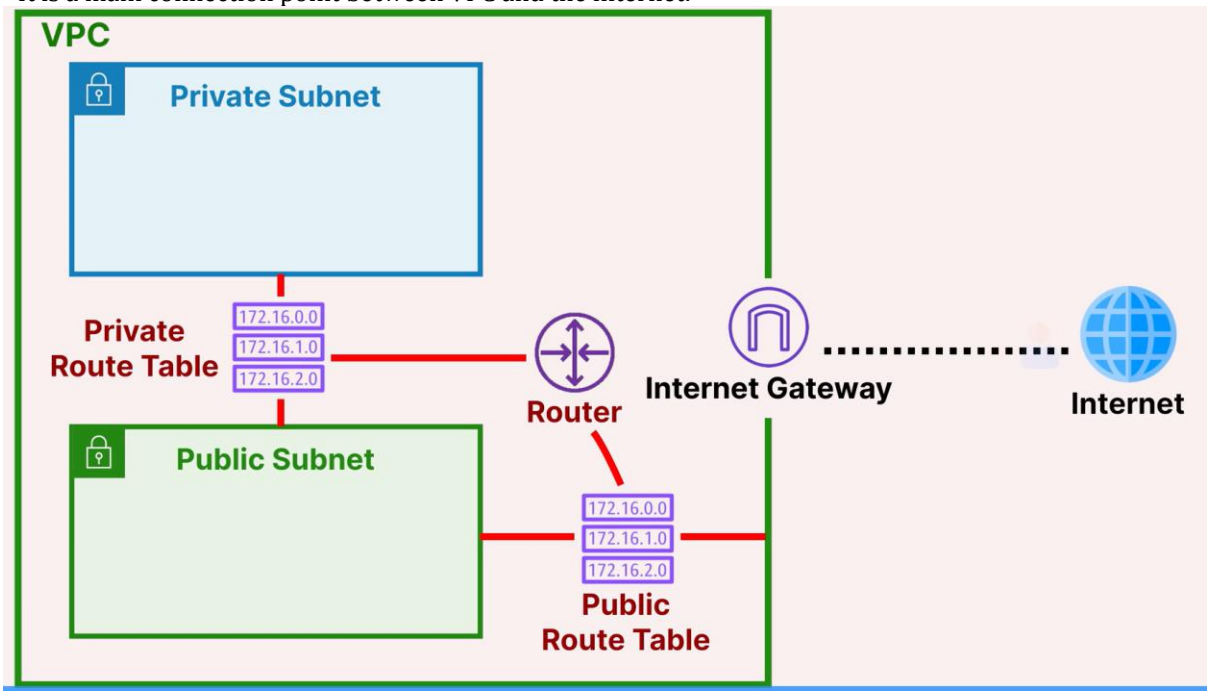
(3) Route table

- Think a route table as a street sign guiding traffic on a road. A Route table guides data packet within your VPC and also help VPC access the Internet and other destinations. The route table consists of a set of rules called route determining directions of traffic flow. Each subnet has its own routing tables but you can also make a customized route table and assign it to a subnet.
- Manually customized route table enhances security controlling each subnet for where they need to go and ensure they only go where they are supposed to.



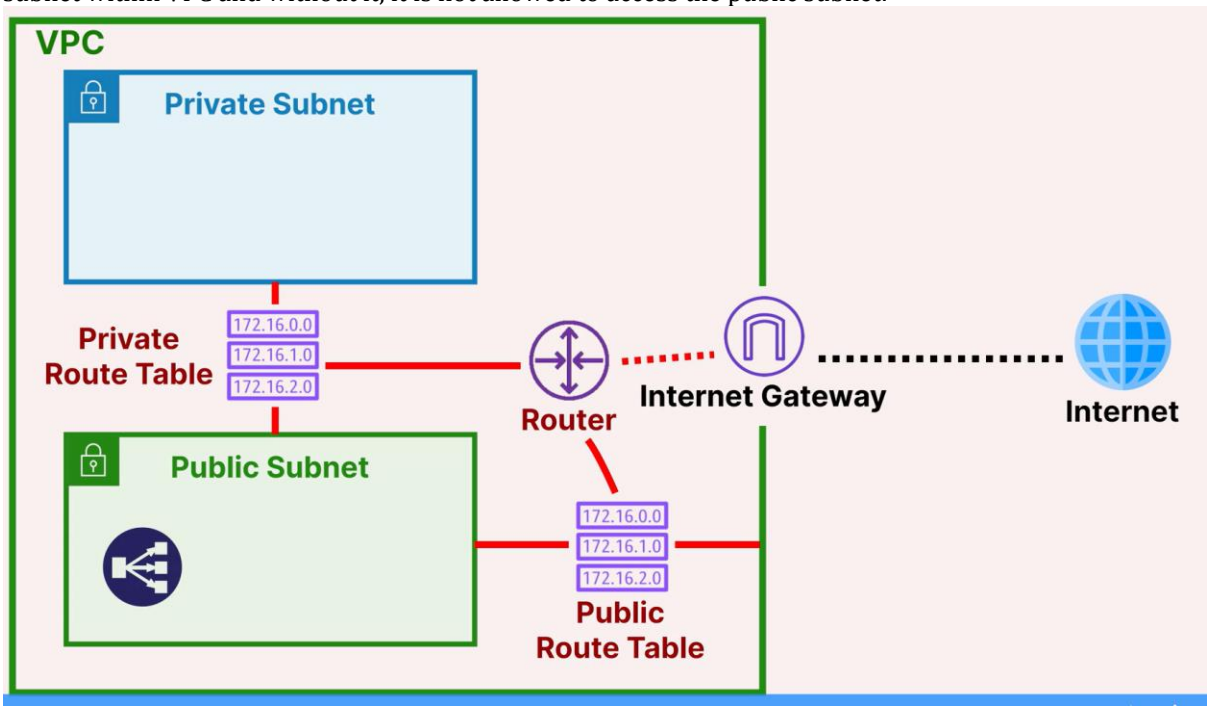
(4) Internet Gateway

- It is a main connection point between VPC and the internet.



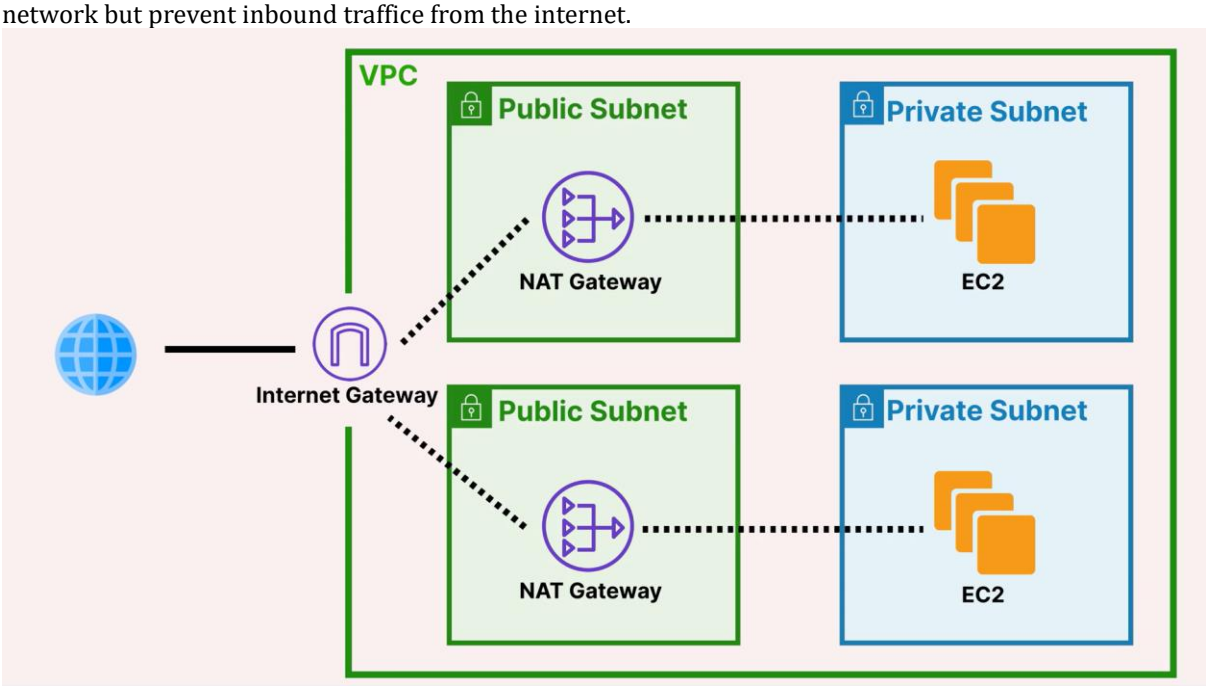
- Internet gateway allows traffic in and out of your VPC. Without the Internet gateway, resources in VPC is

totally isolated from public network. The internet gateway needs to be attached to a public subnet through a public route table defining the roles of the internet gateway to allow the public network to access a public subnet within VPC and without it, it is not allowed to access the public subnet.



(5) NAT (Network Address Translation) gateway

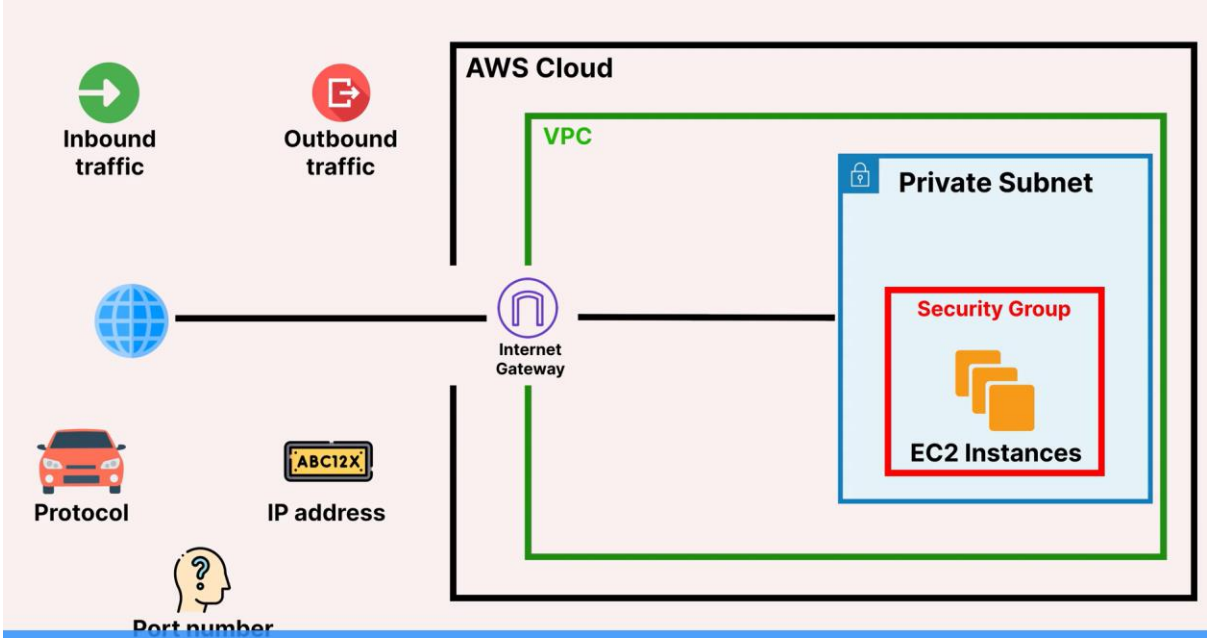
- NAT gateway allows outbound traffic from EC2 instances within private subnets toward the public network but prevent inbound traffice from the internet.



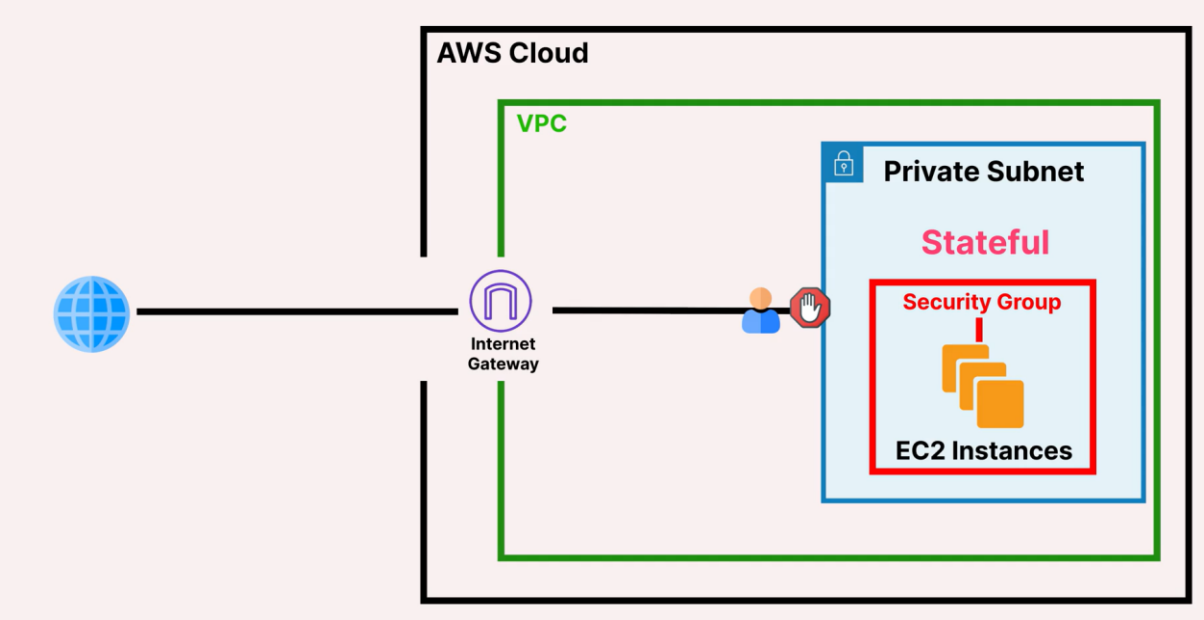
- Servers within private subnets need outgoing traffic to install software and security patch installation. Databases and applications should be installed within private subnets and have a route to NAT gateway routed to the Internet gateway directing toward the public internet.

(6) Security groups

- Security groups consist of rules dictating inboud traffic, who can enter and outboud traffic, who can leave, including information such as protocols, IP address and port number (the reason to visit), which ensure onl approved vistors can gain access.



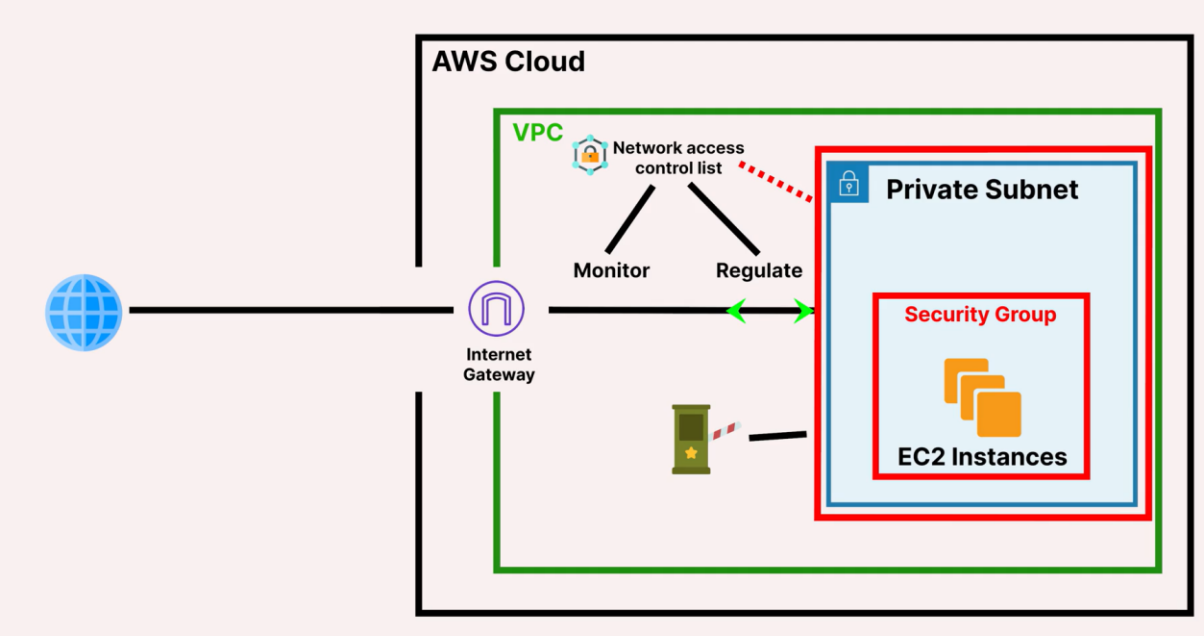
- Security groups are stateless, meaning that once a visitor is allowed in based on their rules. They remember this traffic and when the visitor goes back again, the security groups does not need to check again which means that the visitor is automaticall allowed to leave.
- This characteristic (stateless) manages traffic efficiently and keeps your cloud resource secured.



- In AWS, security groups are set up around EC2 instances and prevent unauthorized users based on rules. When configuring security rules, security groups can determine the types of traffic allowed.

(7) Network Access Control Lists

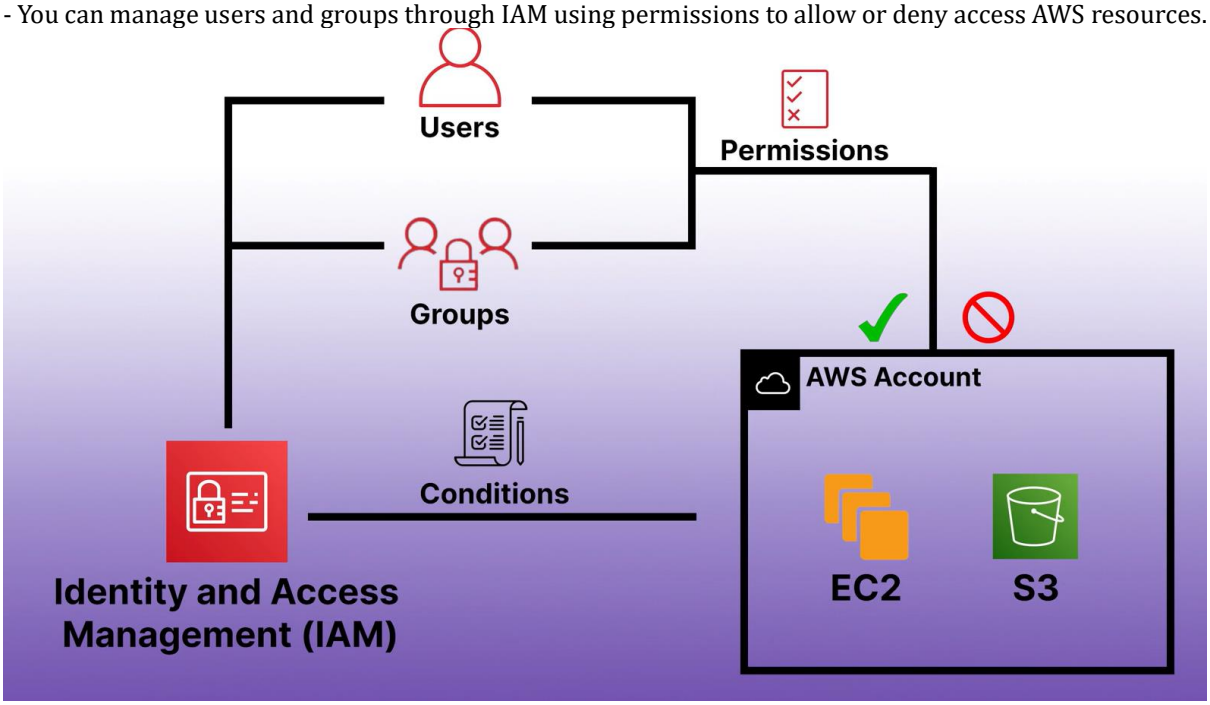
- NACLs act as a security check point attached to each subnet monitoring and regulating traffic in and out of subnets.



- It controls data packets tried to get through a private VPC based on rules but it is not stateful but stateless, meaning that every datapacket in and out the VPC must be checked whenever they go in and out of the subnet against rules.
- You need separte rules used for making data packets in and out of the private subnet respectively because NACLs do not remember the previous allowances.
- NACLs are ideal for enforcing broader traffic rules at the subnet level such as denying access to certain IP addresses or ranges across the entire subnet.
- Security groups cannot explicitly block certain IP addresses but NACLs fill this gap. Security groups manage access to individual instances.

제 4강. AWS IAM (Identity Access Management)

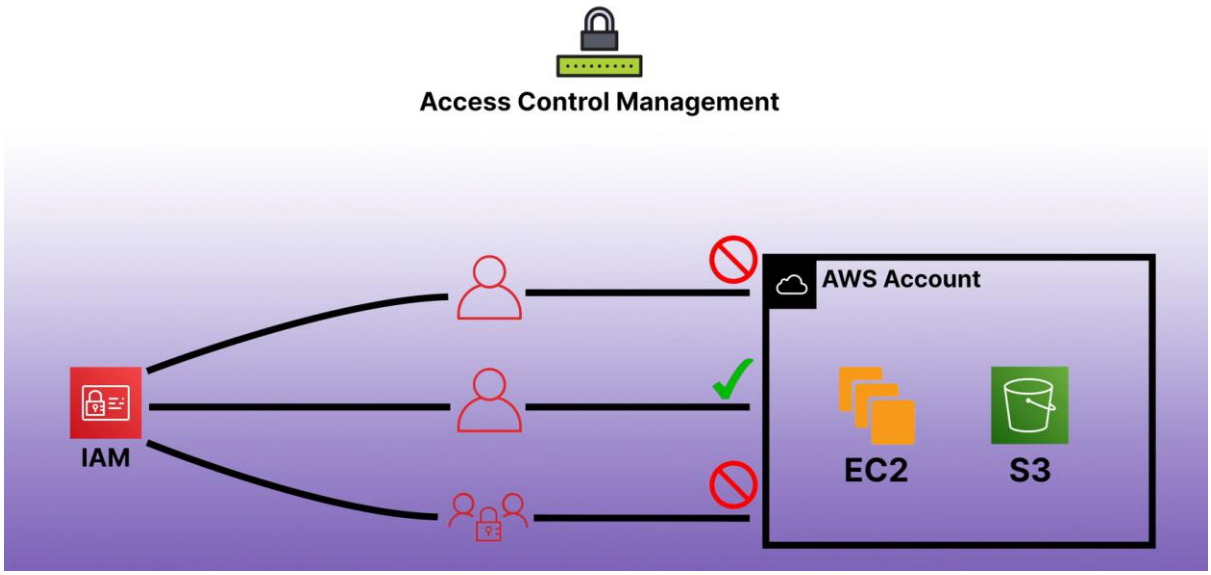
- AWS IAM ensures users to securely access AWS resources such as EC2 and S3 under specific conditions.
- You can manage users and groups through IAM using permissions to allow or deny access AWS resources.



- IAM is essential in managing cloud resources securely for several reasons.

(1) Access Control Management

- IAM can provide permissions to users and groups and specific what they can use and how they can use.



- Due to dynamic and scalable cloud nature, it becomes more complext to manage the access to cloud resources.

(2) Principle of least privilege

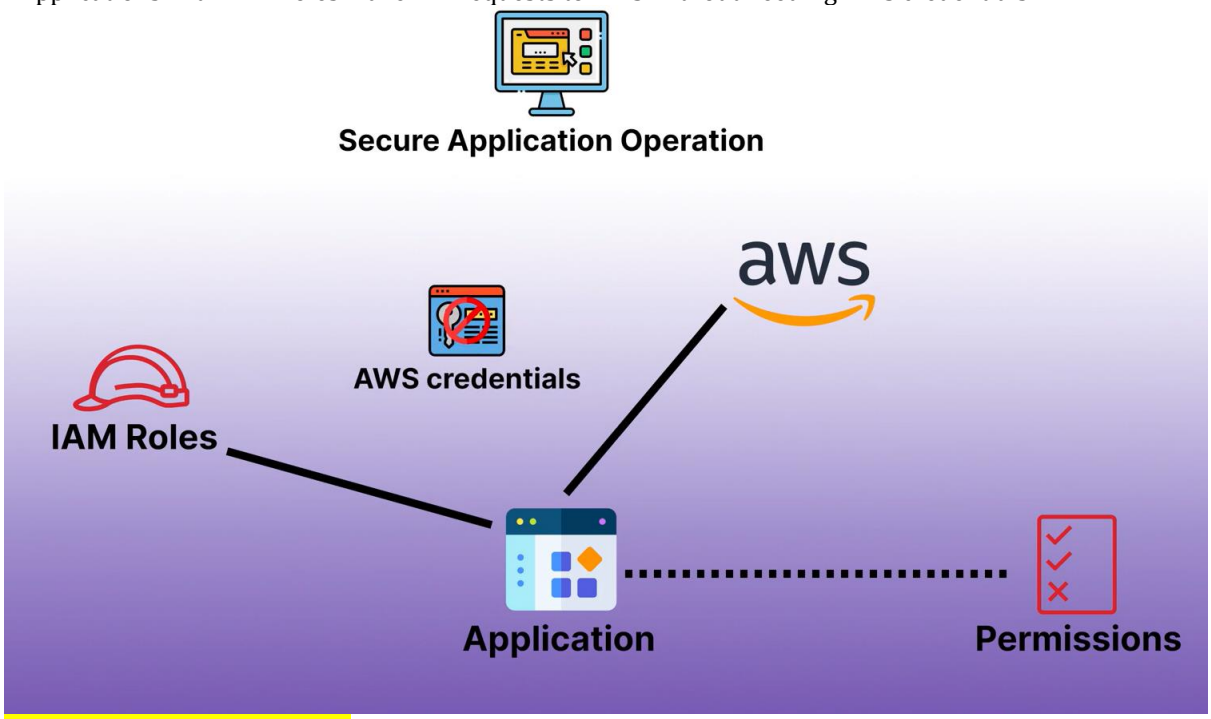
- IAM supports security best practices of granting the least privilege, providing minimum requirements for users and groups to complete tasks.
- This practice minimizesa errors and malicious access to cloud resources by limiting access rights of users and groups, providing the bare? minimum requirements to complete their tasks.

(3) Multi-Factor Authentication (MFA)

- IAM supports MFA, providing extra layers of security to request users to verify their identity using one or more authentication methods. This will significantly reduce unauthorized access AWS environment.

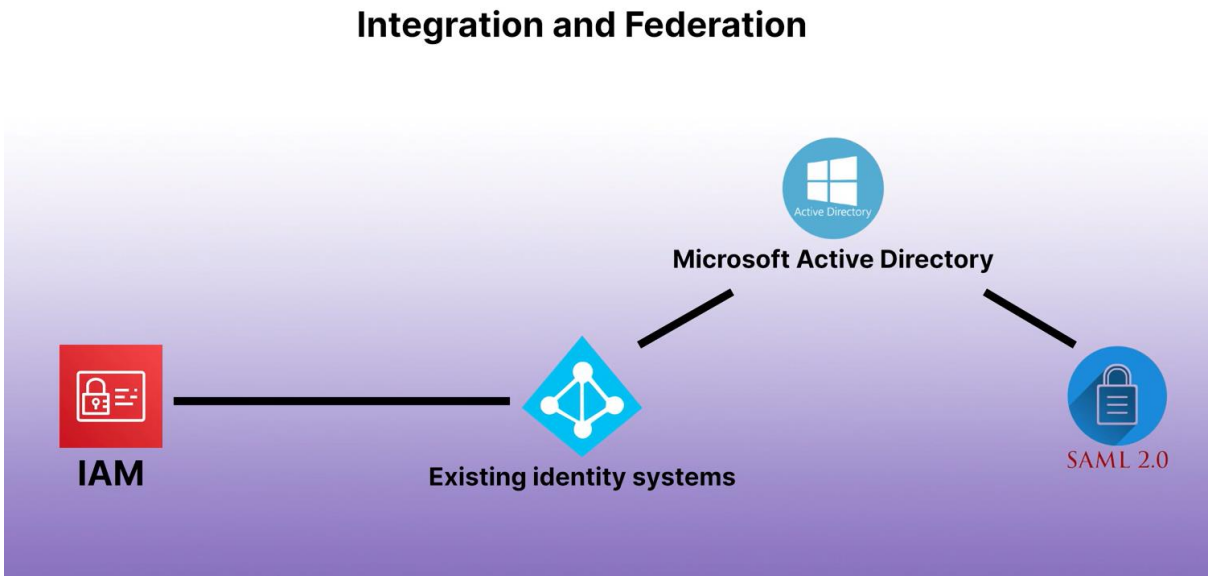
(4) IAM roles

- IAM also ensure applications running on AWS to have only the permissions they need to operate.
- Applications with IAM Roles make API requests to AWS without needing AWS credentials.

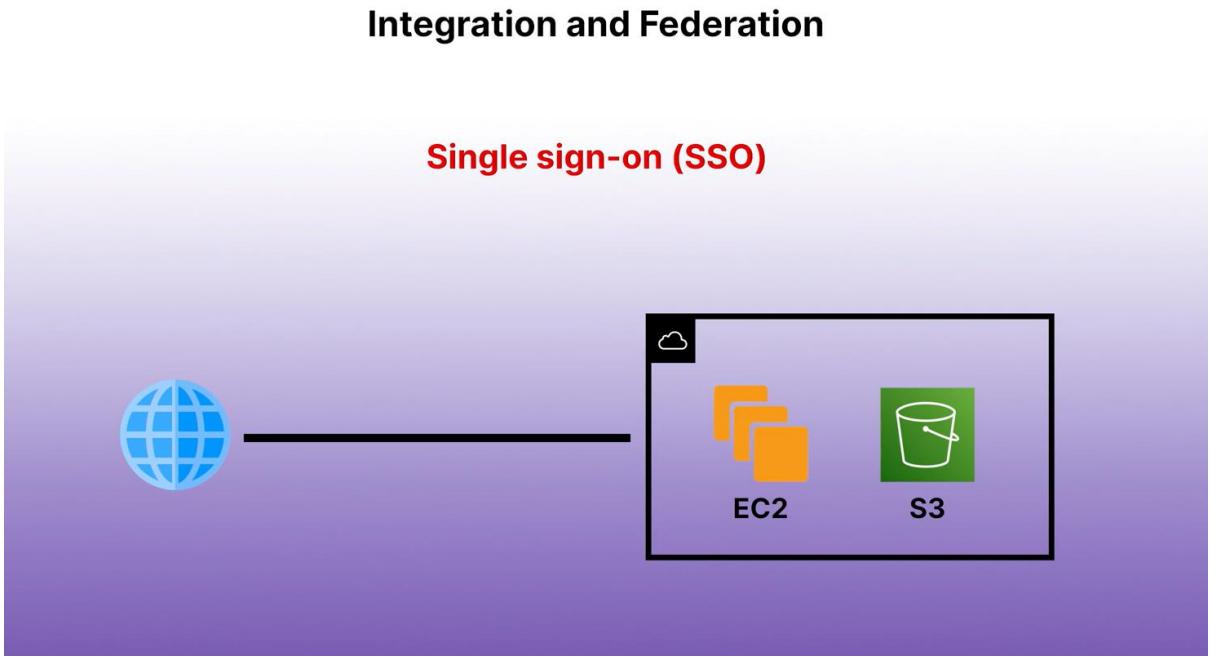


(5) Integration and Federation

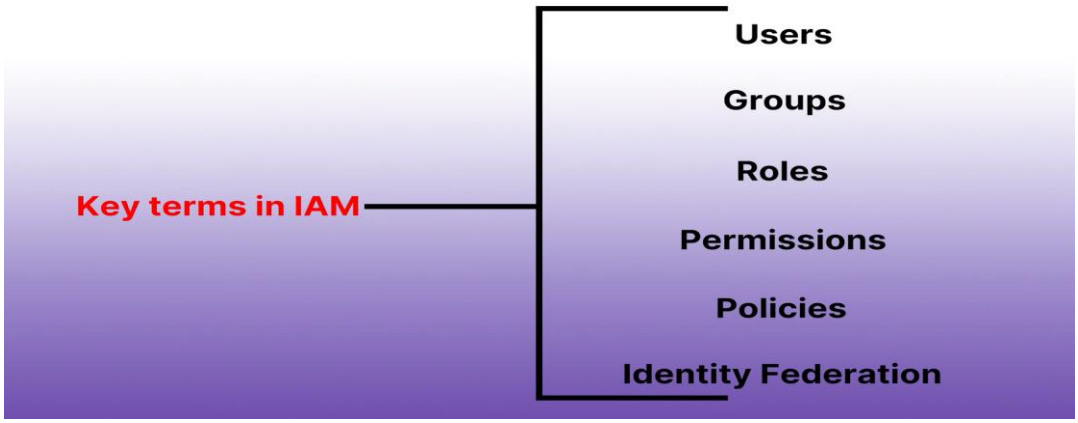
- IAM can integrate with exisiting identity systems such as Microsoft Active Directory or any other identity system supporting SAML 2.0



- You can use your own SSO (Single Sign-on) when accessing AWS resources, allowing users to utilize their own credentials that simplify the access management across organizations.



Key terms in IAM



(1) Users
- IAM users are individuals or services given permissions to access and interact with AWS resources.

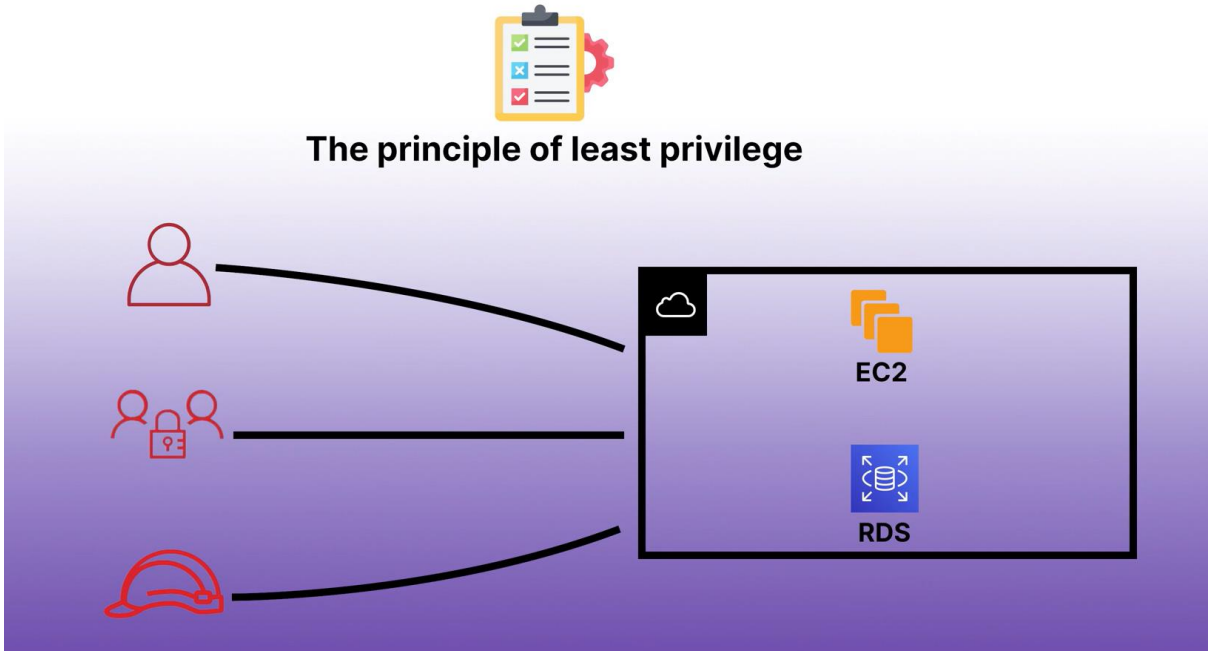
(2) Groups
- IAM groups organize and group users by similar roles and access needs for easier management of permissions.

(3) Roles
- IAM roles is a temporary access badges for limited access for specific AWS resoruces. IAM roles provide permissions for AWS resources access not using permanent user credentials.

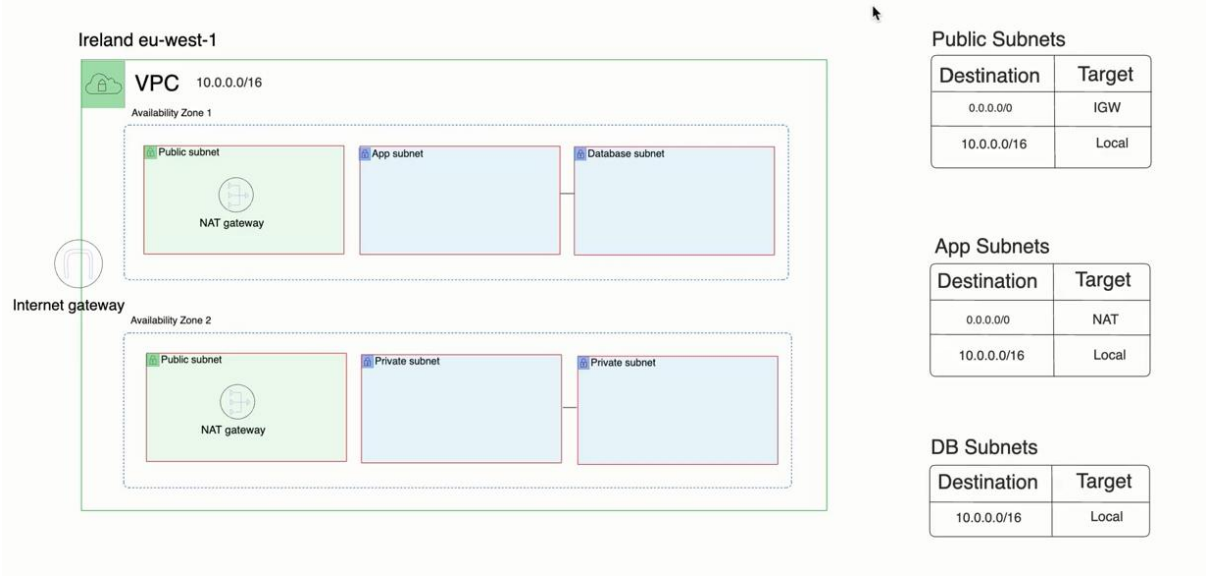
(5) Policies
- IAM policies define permissions or access rules about which AWS resources that users or roles can access and how they can interact with them.

(6) Identity Federation
- Identitiy federation allows users to authenticate using their existing identity from external systems. For examples, corporate directories can access AWS resources without creating additional IAM user account.

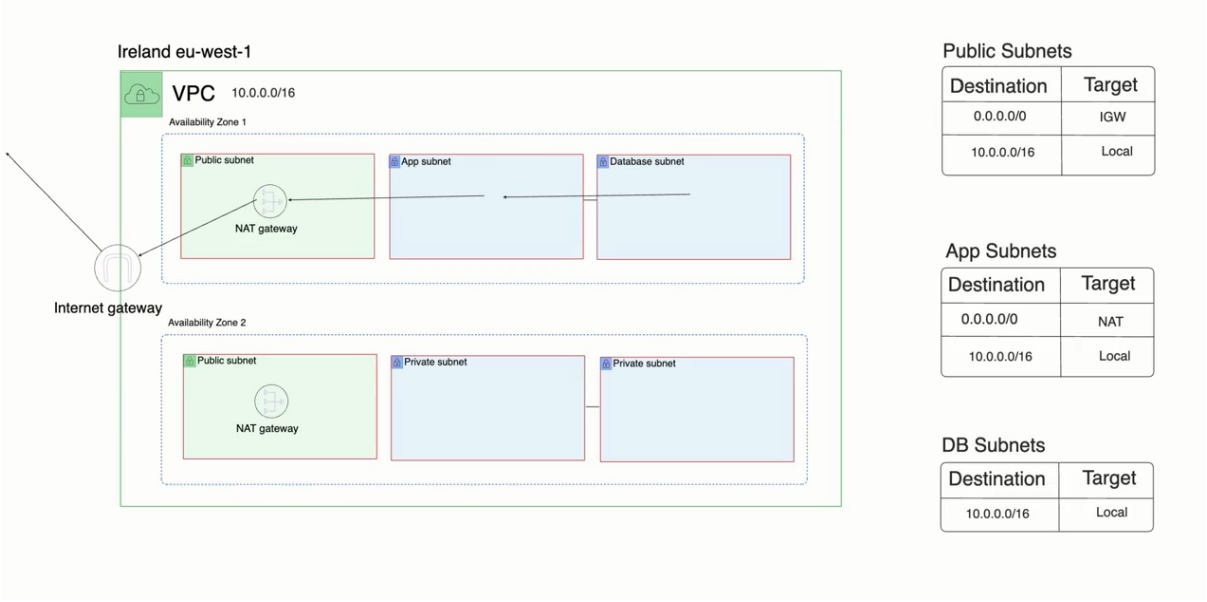
(7) The principle of least privilege
- Users, groups and roles can only have the minimum levels of access needed to perform their tasks, preventing unnecessary access to sensitive information.



제 6강. VPC Diagram



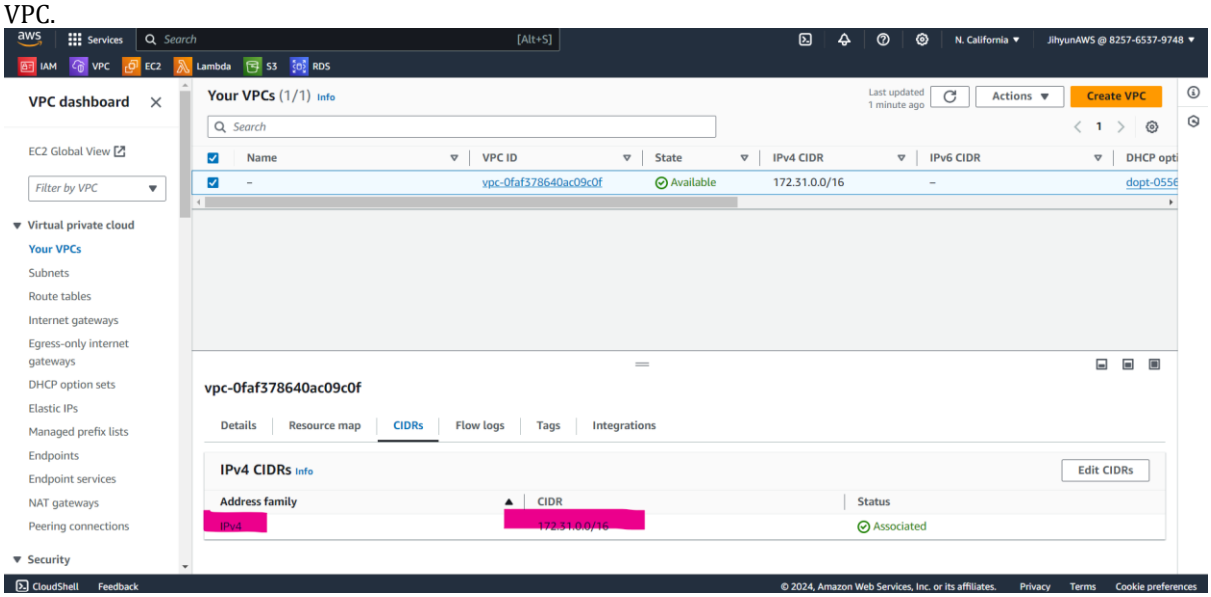
- On the right side, it is route tables including destination and target. In the public subnets, all possible IP addresses (0.0.0.0/0) would be routed to IGW (Internet Gateway).
- It means that all resources within the public subnets will be directed to IGW.
- In the app subnets, all routes or all possible IP addresses are routed to NAT gateway. All resources or possible IP addreses in the app subnets will be directed to NAT gateway, which will be redirected to IGW.
- 10.0.0.0/16 is CIDR blocks, allowing for local routing within our VPC which means that resources wtihin VPC can communicate with each other. For example, if database subnets do not have any CIDR block, other resources cannot be directed to the database subnet. If we add CIDR blocks into DB subnets, they will accept any traffic within VPC.
- There isn't any default route to the NAT gateway in the public subnets inside DB subnets. However, it can access the app subnet using CIDR blocks that can be accessed the NAT gateway through the app subnets.

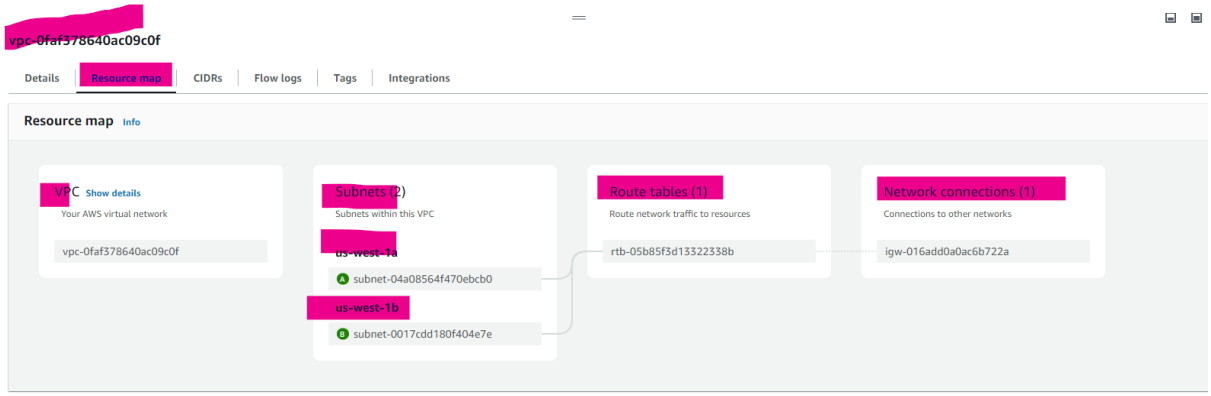


- NAT gateway allows routes from private subnets to the Internet but prevent unauthorized access from the Internet to the private subnets. It allows only outbound traffic but not inboud traffic, enhancing security in AWS environment because it prevents resources within private subnets from being exposed to external attacks.
- There are 2 different AZs and each NAT gateway is placed within a respective AZ and resources within the app subnets will be routed through the NAT gatway in the same AZ because there will be extra charges for cross-AZ.

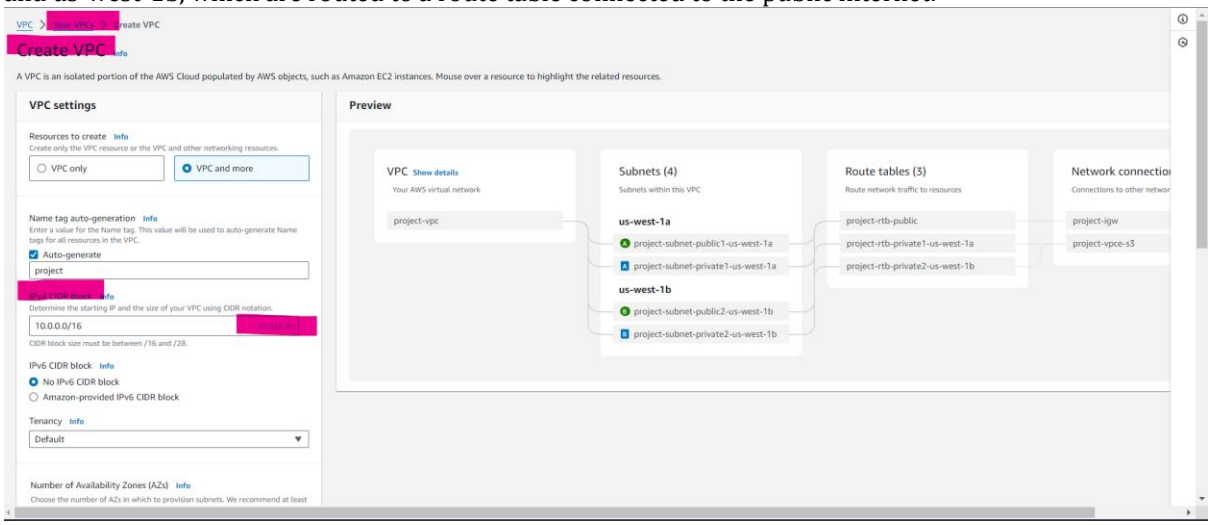
제 7강. Create your VPC

- When you click your VPCs, there is a default AWS VPC. There is a CIDR block which is the IP address of VPC.

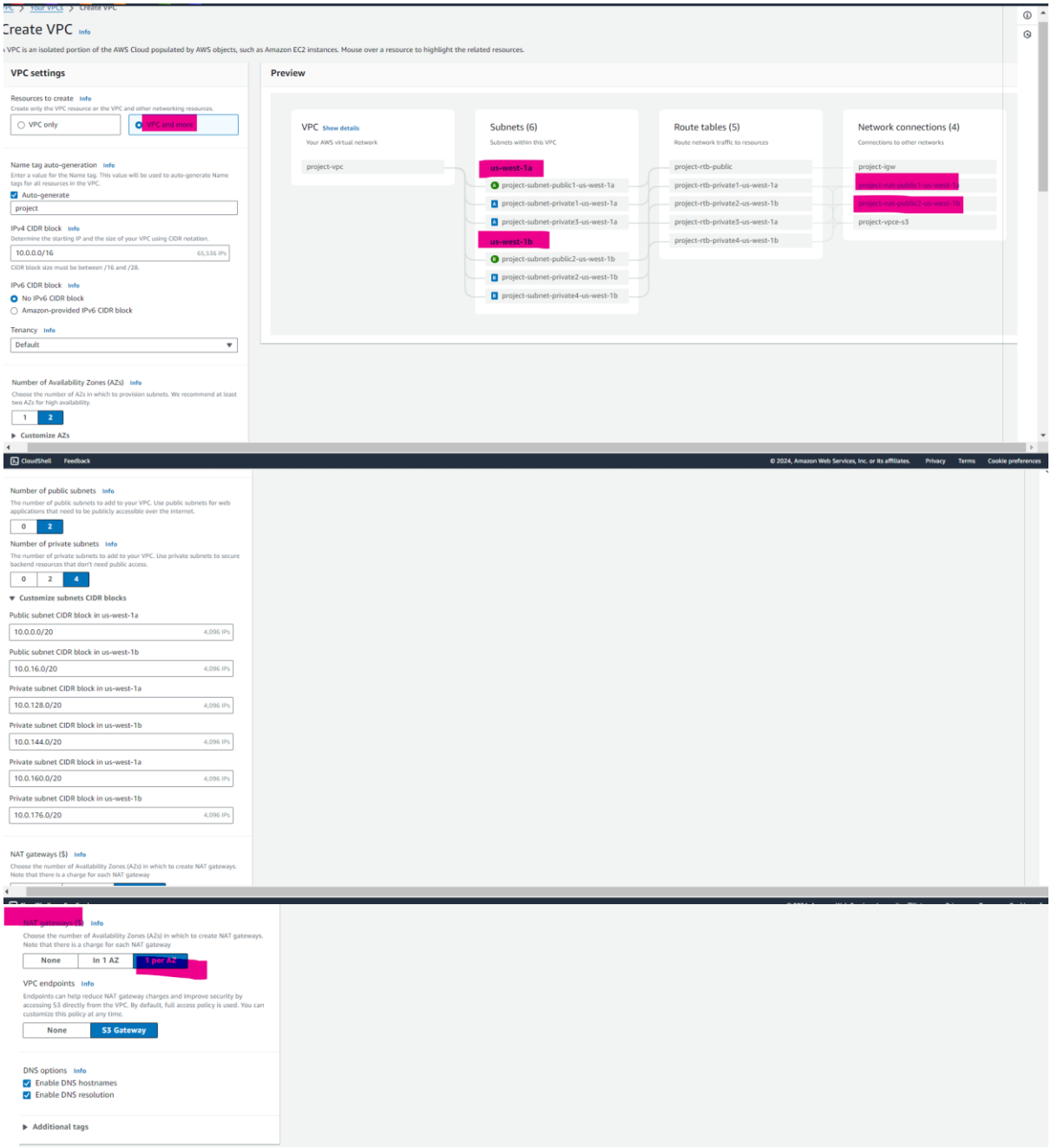




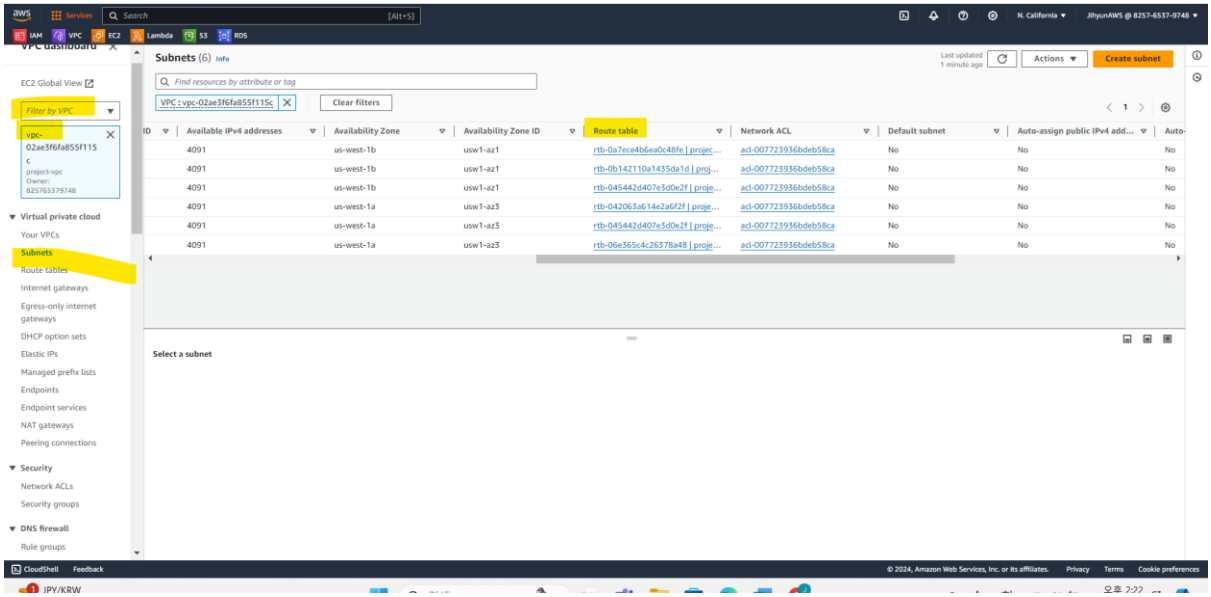
- In the resource map, there is a VPC with 2 different kinds of subnets in different AZs such as us-west-1a and us-west-1b, which are routed to a route table connected to the public internet.



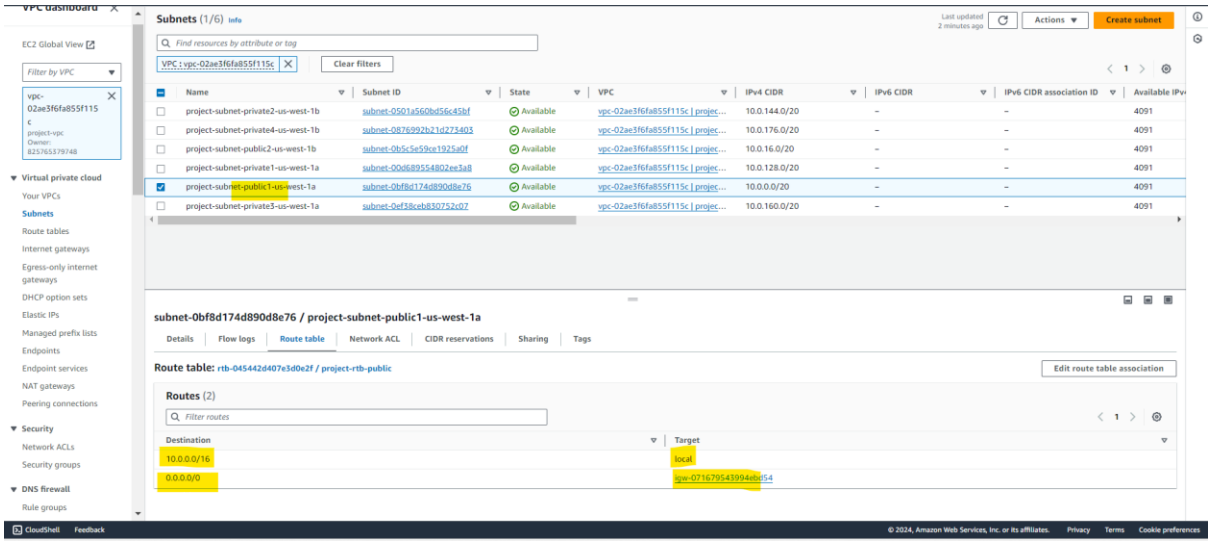
- When you create a VPC and more, there is a IPv4 CIDR block with 65,536 IP addresses.
- VPC endpoints connect internal connections to your S3 bucket or dynamoDB.



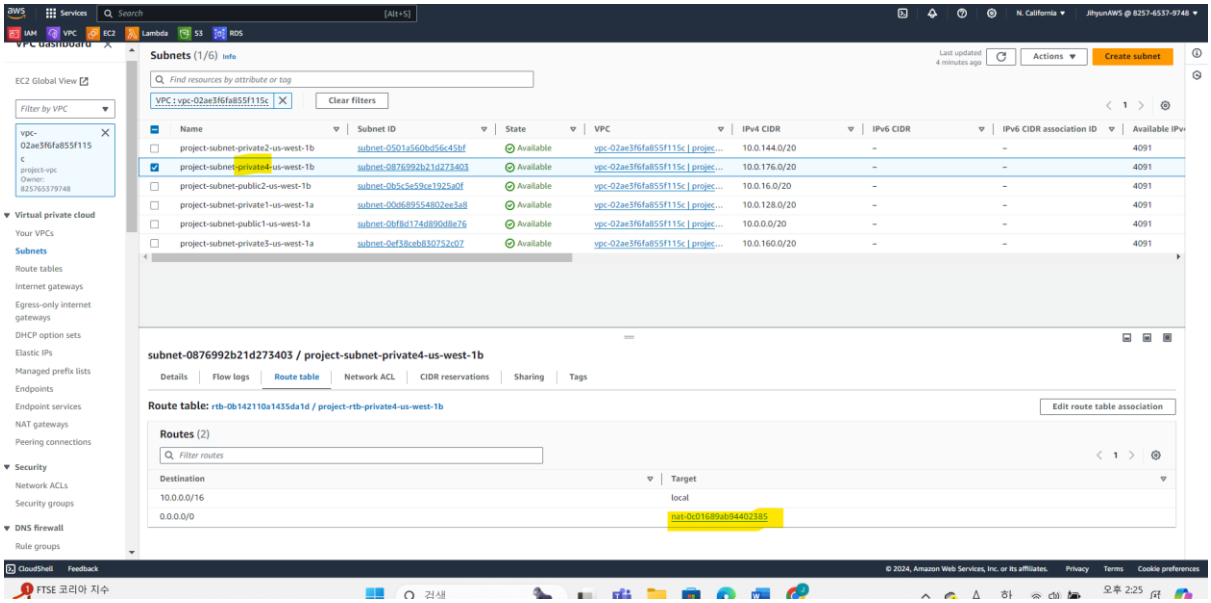
- There are 2 AZs closest to users by default such as us-west-1 and us-west-2. There are 2 different public subnets separted in each AZ and 4 different private subnets, among which 2 different private subnets are placed within each AZ.
- We can place each NAT gateway for each public subnet to make resources within private subnets connected to the public internet.
- 2 different public subnets are routed to the Internet gateway for the public internet connectivity.



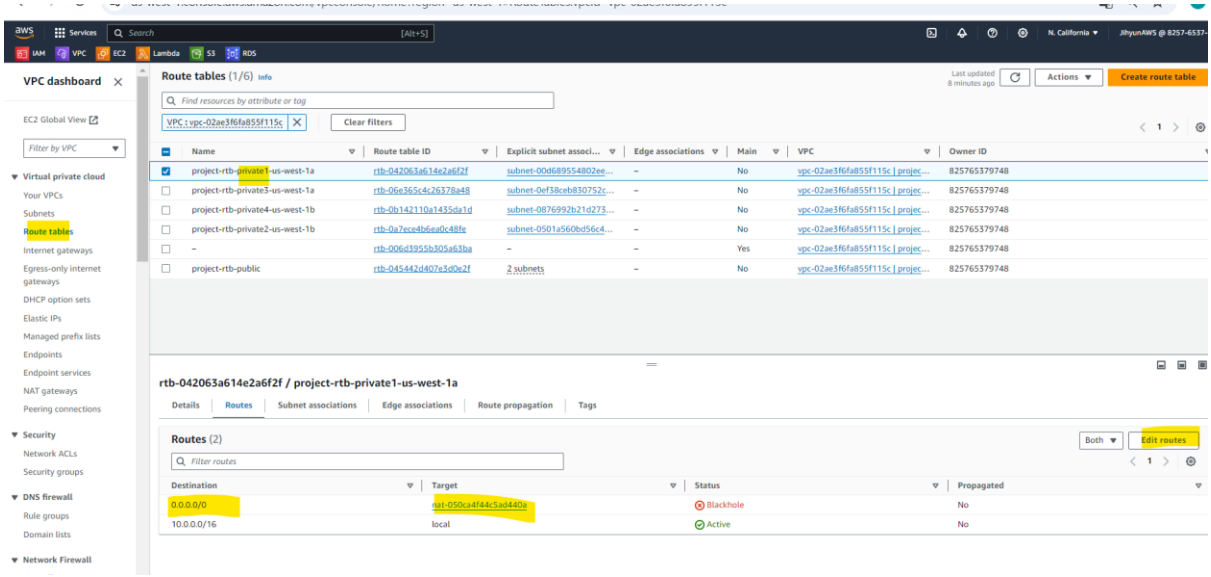
- Each subnet has its own route table. You can go to the subnets and filter the subnets.



- In a public subnet, you can see a IP address for local communication within VPC and all possible resources wtihin the public subnet can be routed to the internet gateway for external connectivity.



- In a private subnet, all possible resources can be routed to the NAT gateway in each public subnet in the same AZ for the external connectivity. And also, it is possible for internal communication with resources within VPC.



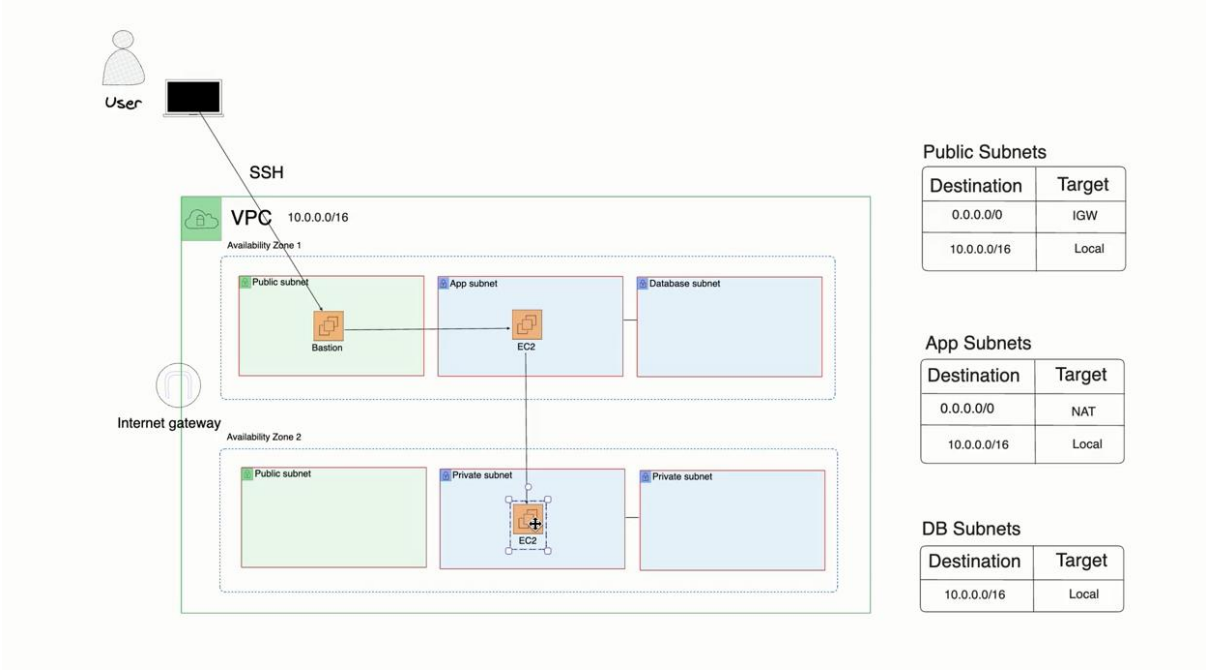
- In route tables and a private subnet, if there is any NAT gateway route table, we can edit routes and add a route table for routing to a NAT gateway in a public subnet.

제 11강. Updating Architecture

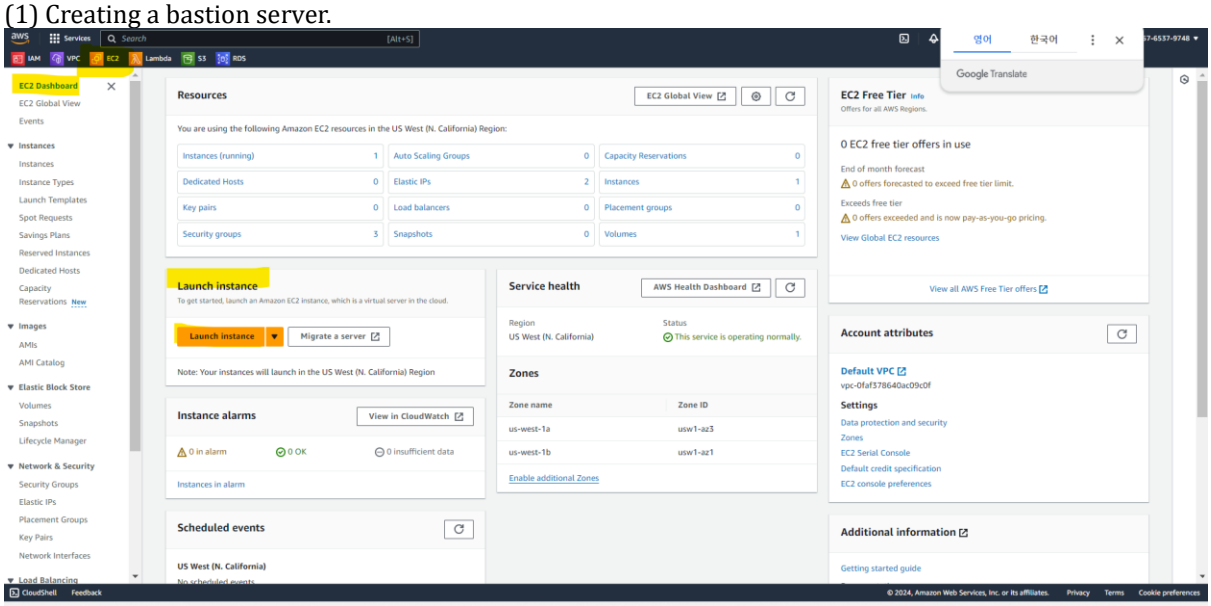
- Due to the fact that NAT gateway costs a lot of money, we replace it with Bastion host in a public subnet. This bastion host is connected to EC2 instances running in both app subnet and private subnet.
- We want to ping EC2 instances in the private subnet from the app subnet in AZ 1 while configuring secuirty groups for EC2 instances.

제 12강. Setting up Bastion host

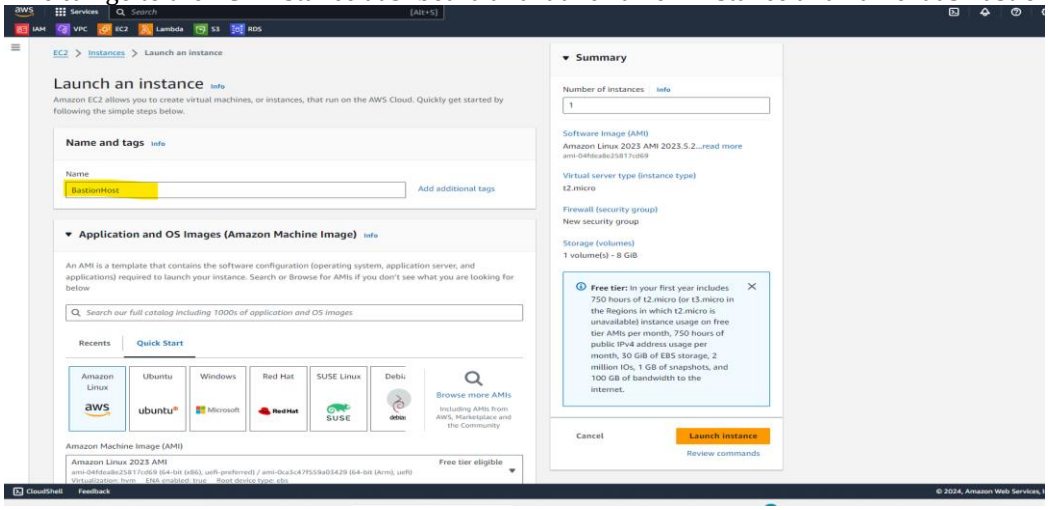
- A bastion host is attached using SSH by our local machines that can connect to EC instances in a app subnet and ping EC2 instances in the 2nd AZ from EC2 instance from the 1st AZ.



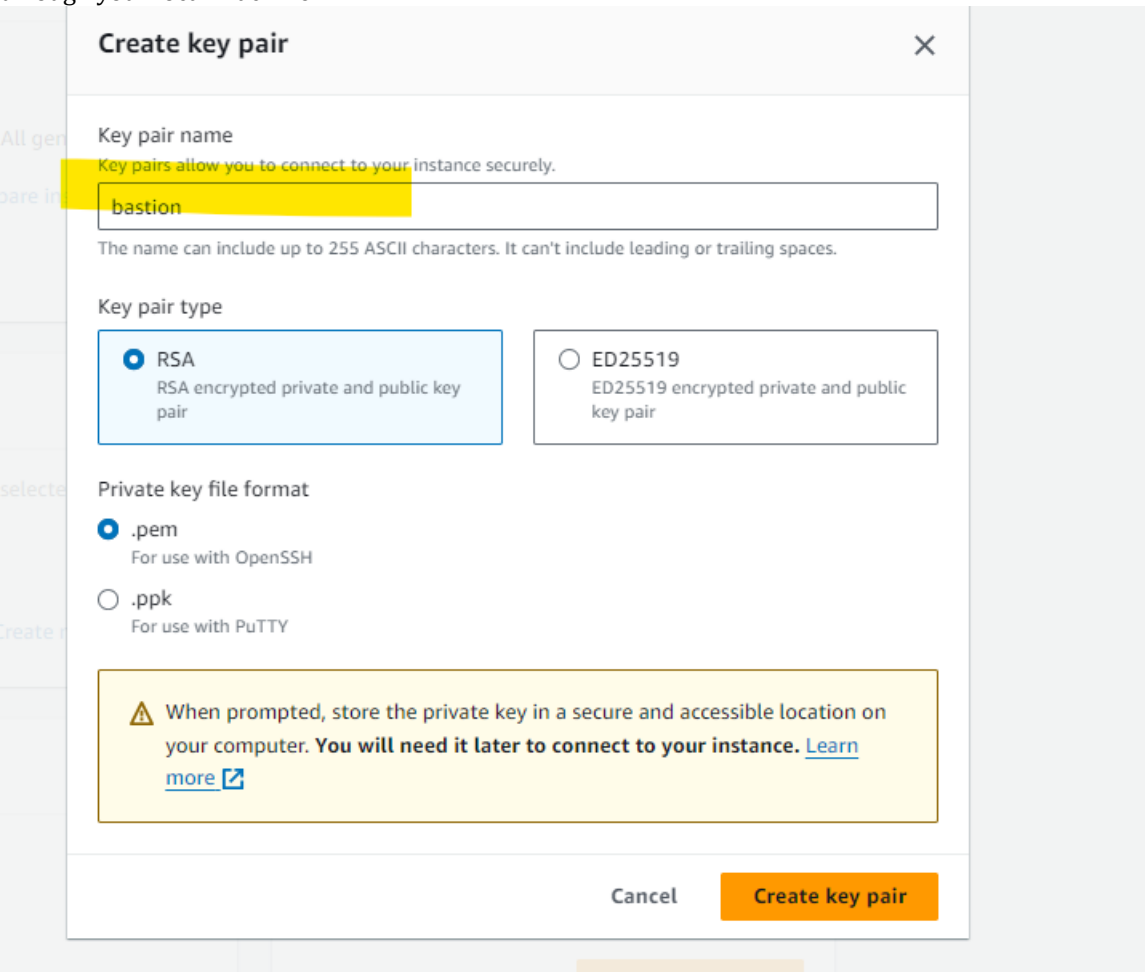
(1) Creating a bastion server.

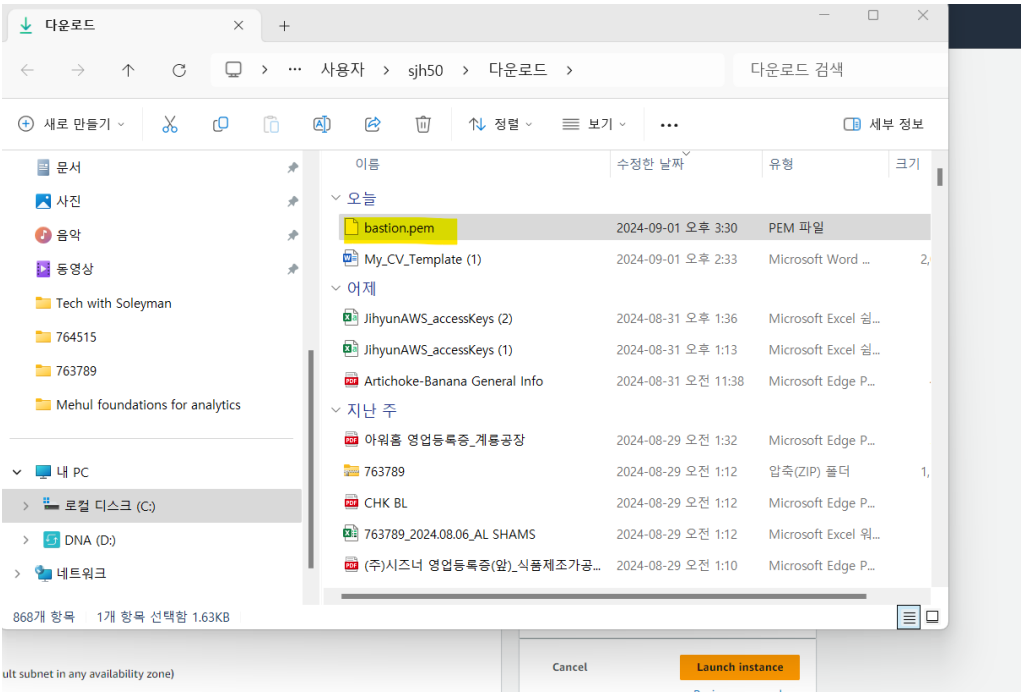


- We can go to the EC2 instance dashboard and launch a new instance and name it as BastionHost.

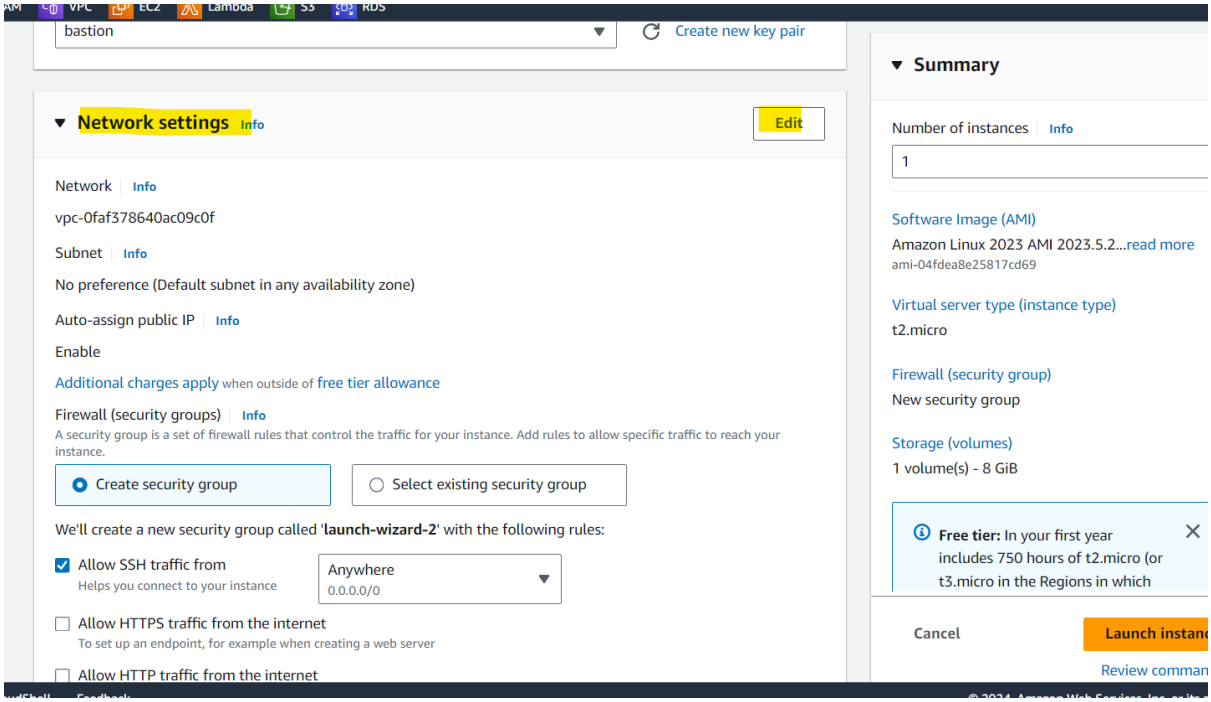


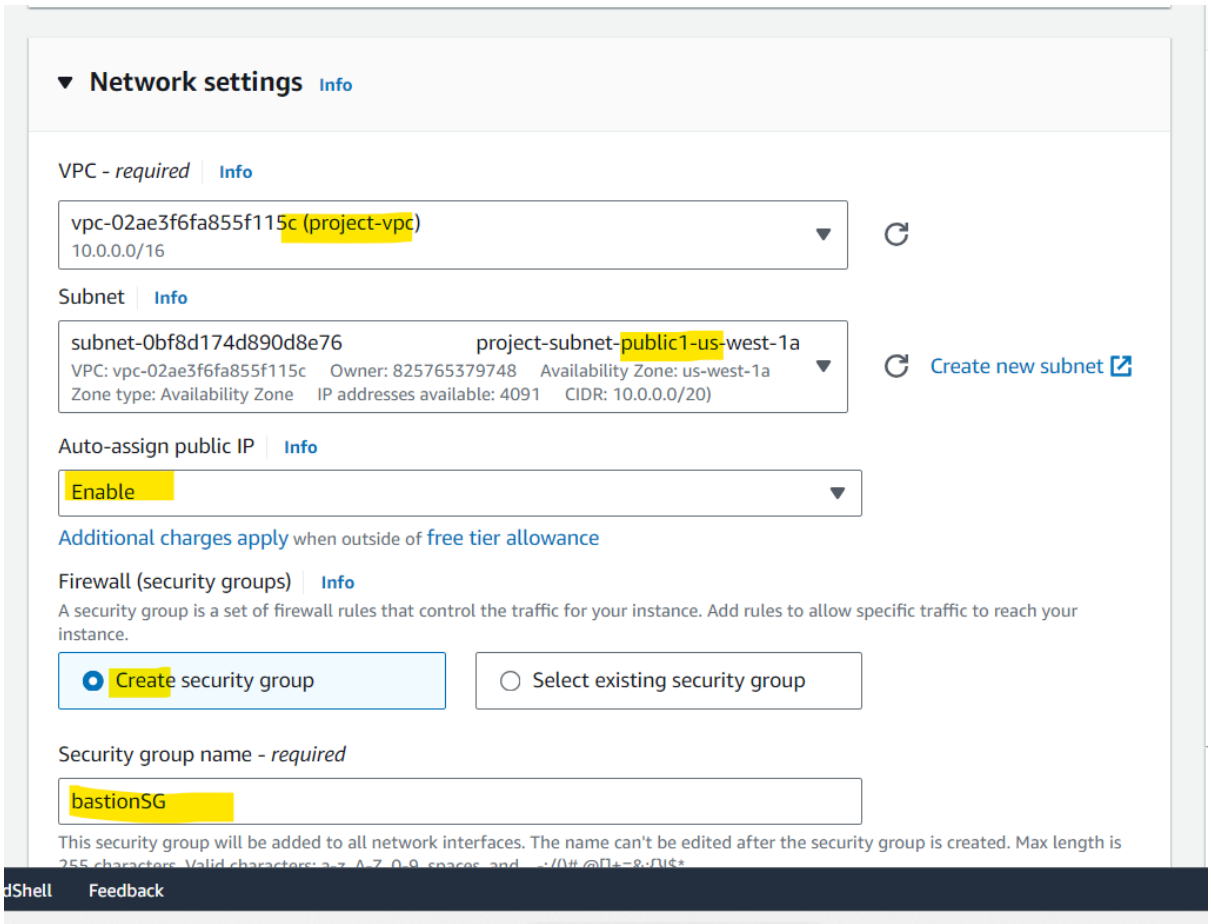
- Scroll down to key pairs and create a key pair that will be used for security connection to the BastionHost through your local machine.



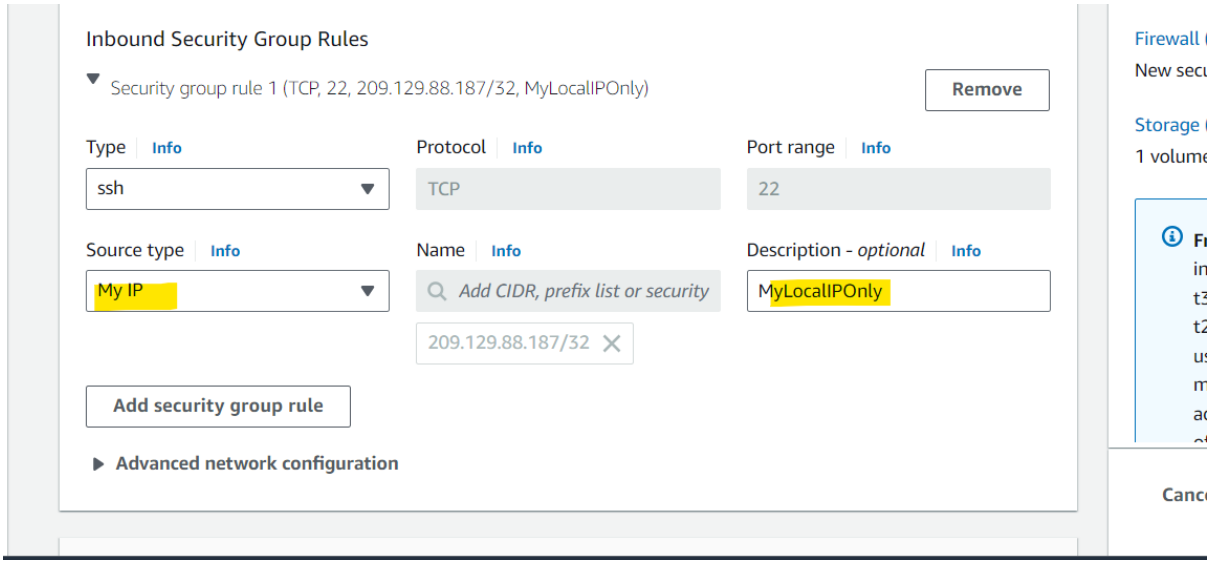


- And then moving down to the network settings and clicking edit.



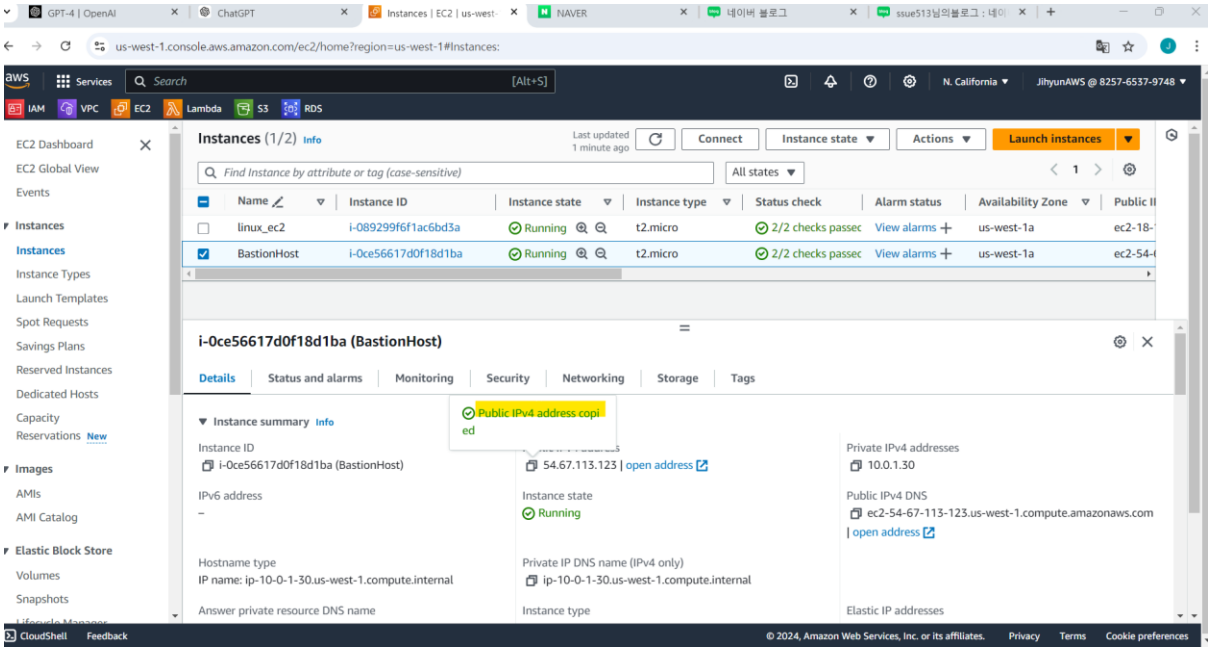


- In the network setting, you can select customized VPC and subnet must be a public subnet where this bastion EC2 instance will be placed. Auto-assign public IP is enabled and create a security group for this bastion EC2 instance.



- In inbound security group rules, Source type should be My IP only because we want to make only our local machine to access the Bastion Host.

제 13강. SSH into Bastion host

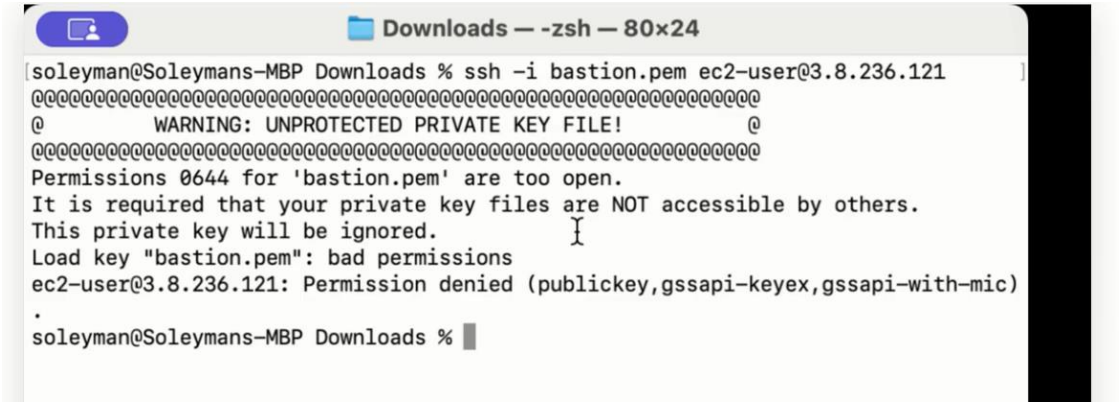


- Copy the public IPv4 for Bastion Host.

1. Running SSH commnads to connect to the Bastion host and also make sure where you download the key pair for this instance.

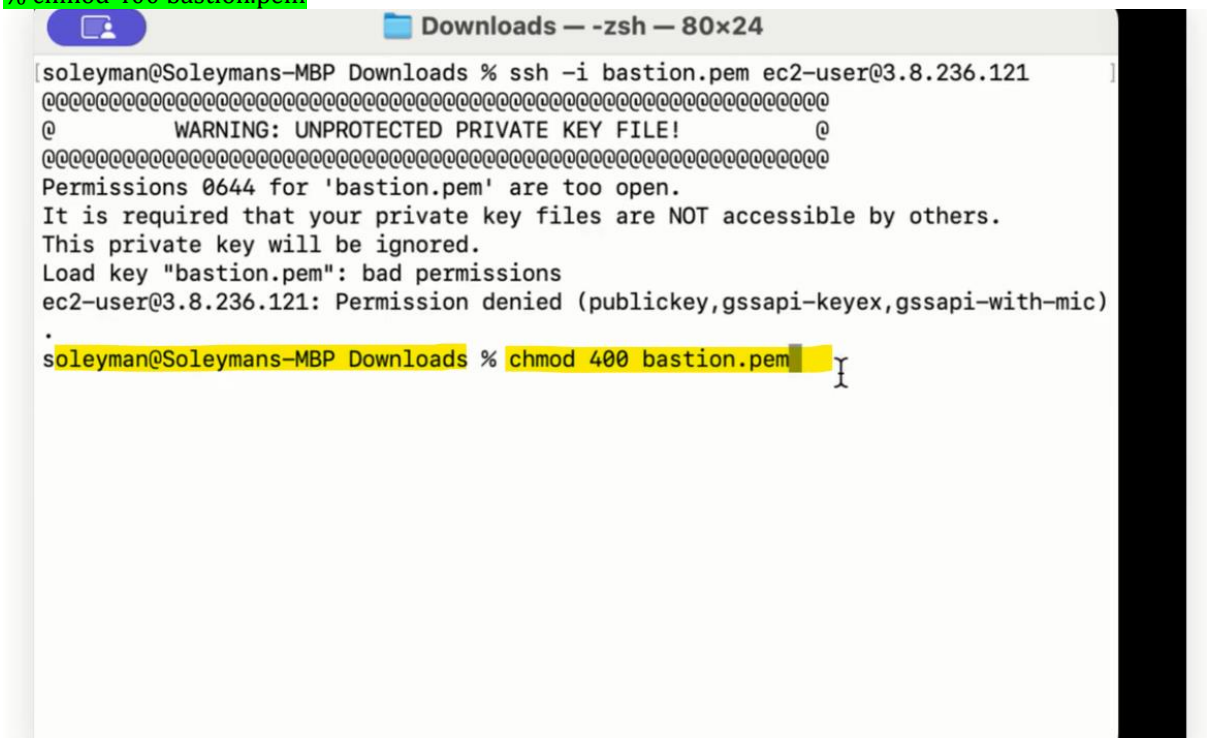
% -i bastion.pem ec2-users@54.67.113.123

- write down the public IPv4 for the instance after @ but the CLI is asking to lockdown the bastion.pem file because it is a secret key that allows for the secure access to Bastion EC2 instance.



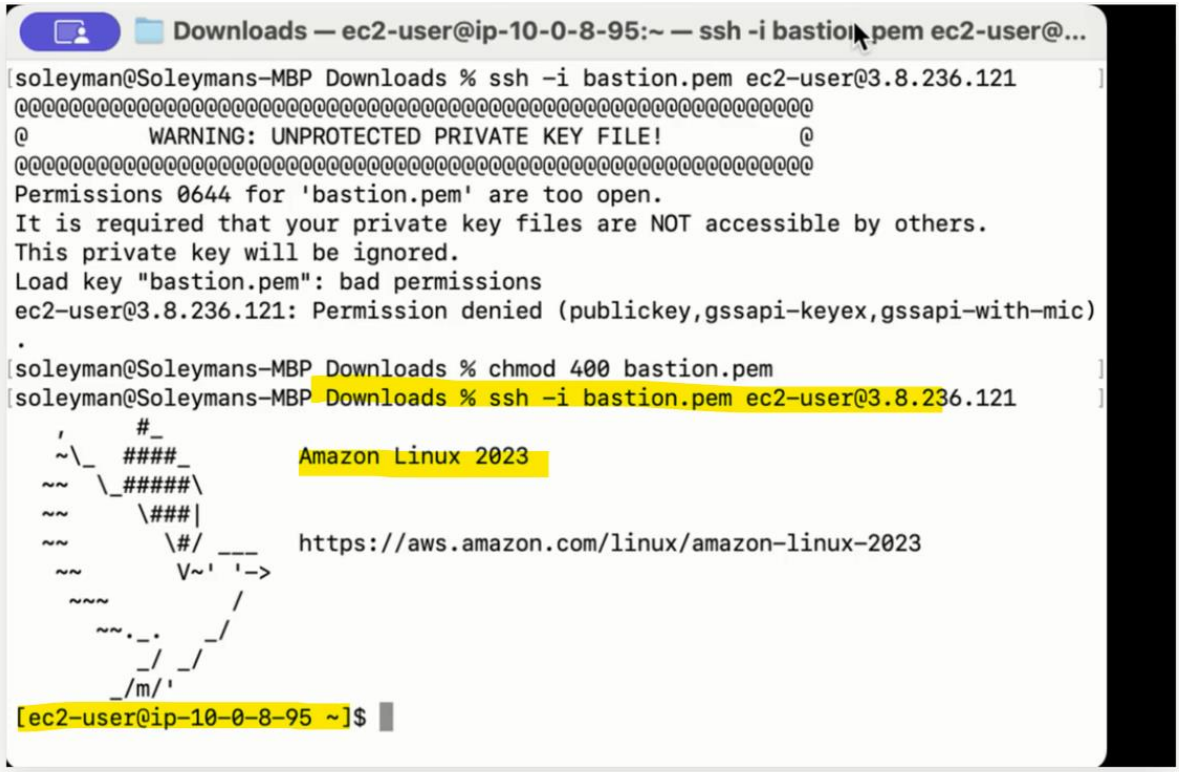
2. We need to change the permissions but please make sure that the bastion key file should be located in the same directory you designated.

% chmod 400 bastion.pem



3. After changing the permissions, we rerun the same command again.

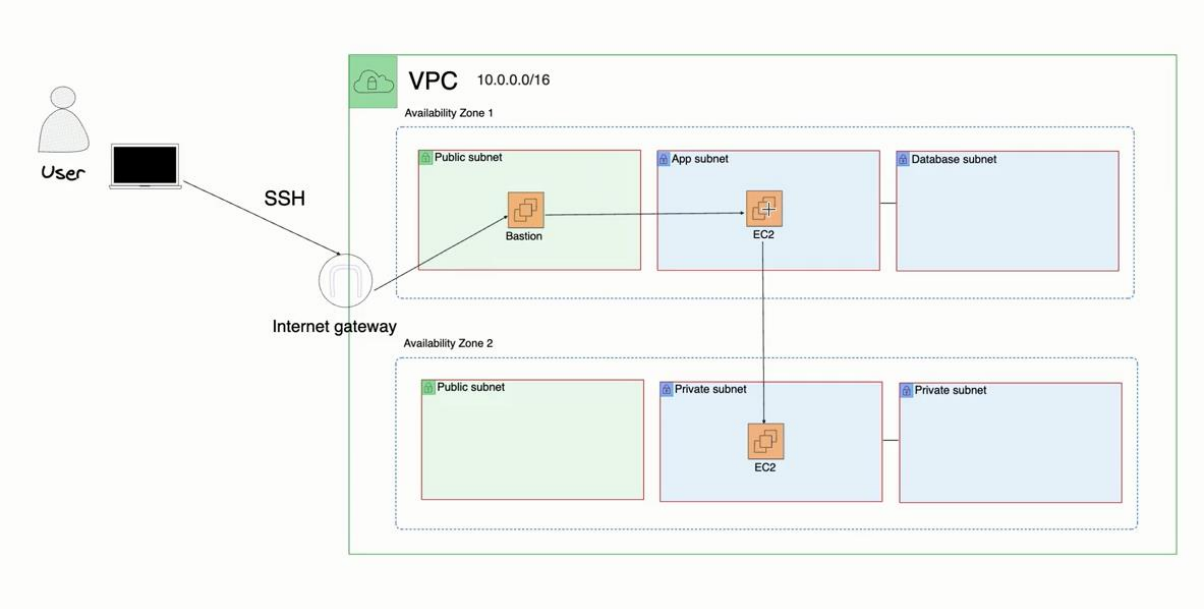
% -i bastion.pem ec2-users@54.67.113.123



- And now we can successfully access the bastion host and 10-0-8-95 is the private IP address for this instance and successfully get into the Bastion host via our local machine.

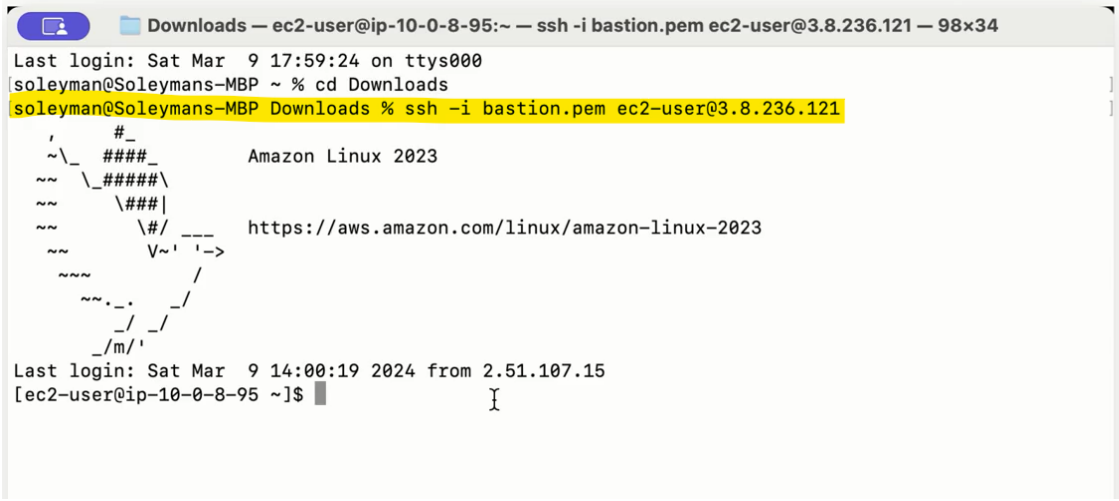
제 14강. Routing in Action

- We need the public internet connectivity if we want to SSH into EC2 instances in private subnets. We can use NAT gateway but we need to pay a lot of money. Instead, we are using the Bastion Host to help private EC2 instances connect to the public internet via SSH from your local machine.

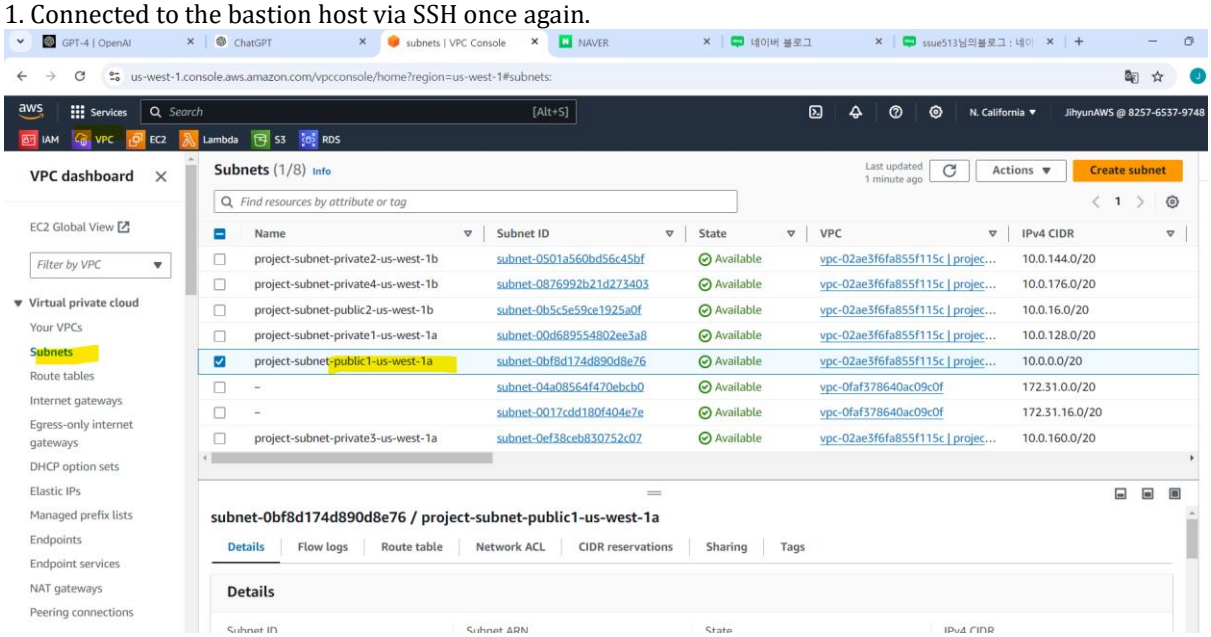


- However, the actual route is that SSH is connected to the Internet gateway to connect to the Bastion Host. If we remove the Internet gateway, we no longer access the Bastion Host via SSH with local IP. The bastion Host has its route to the Internet gateway.

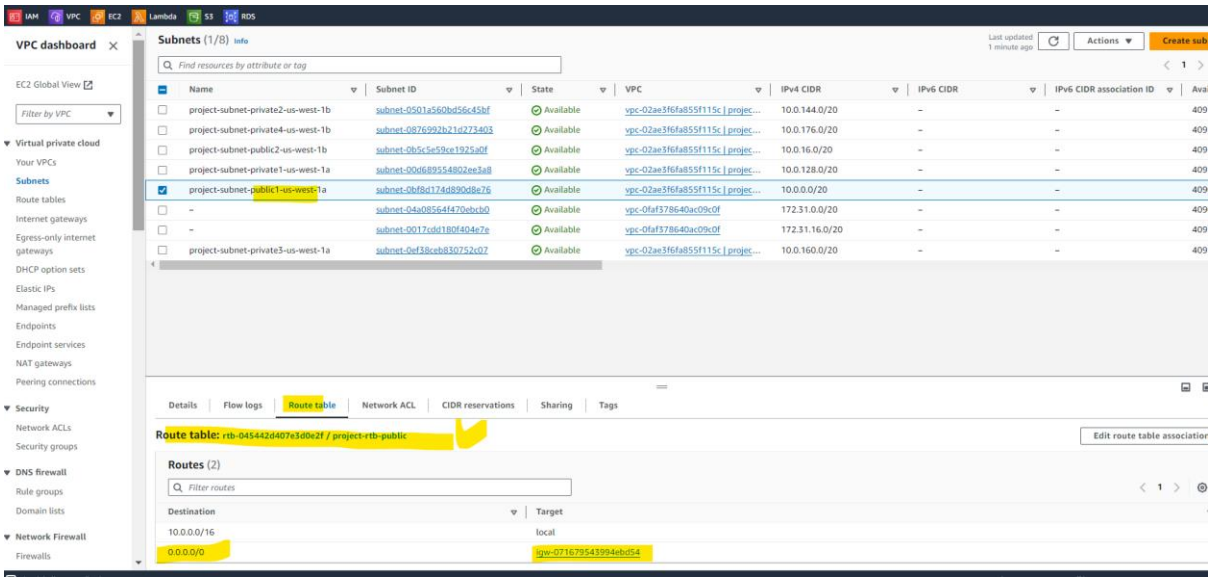
What about removing the route for the Internet gateway?



1. Connected to the bastion host via SSH once again.

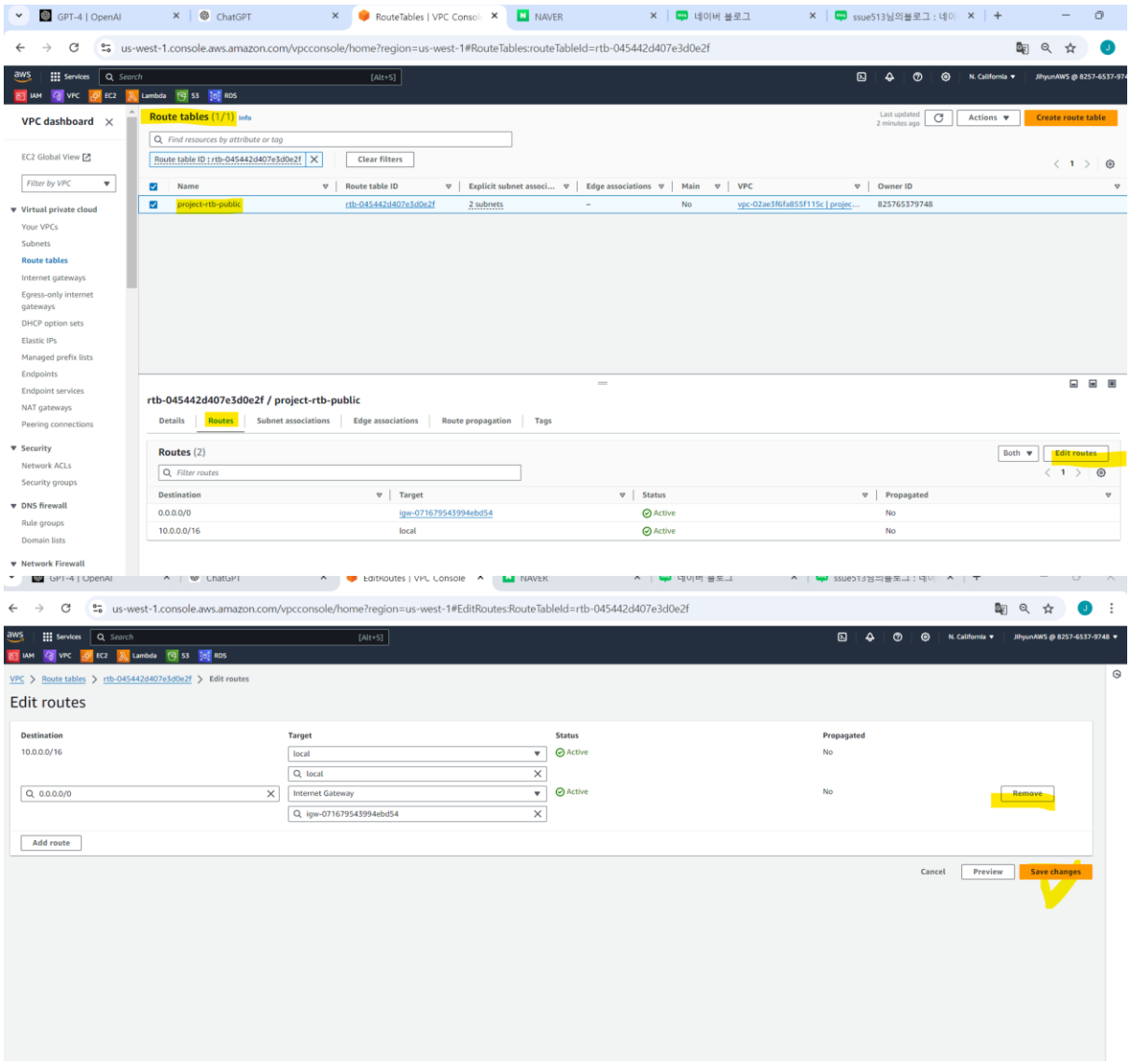


- Moving onto the subnets in VPC and click the public subnet where the Bastion host is placed.

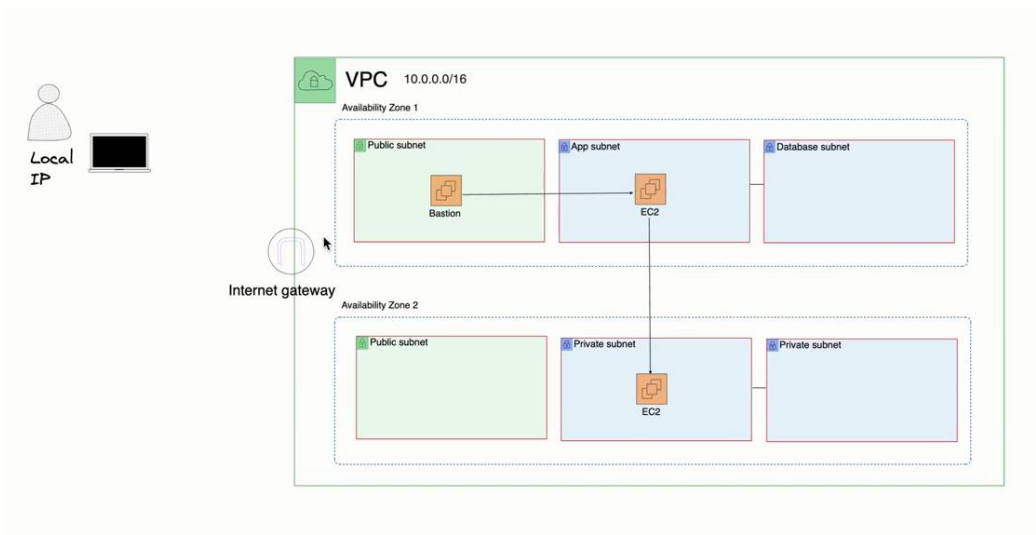


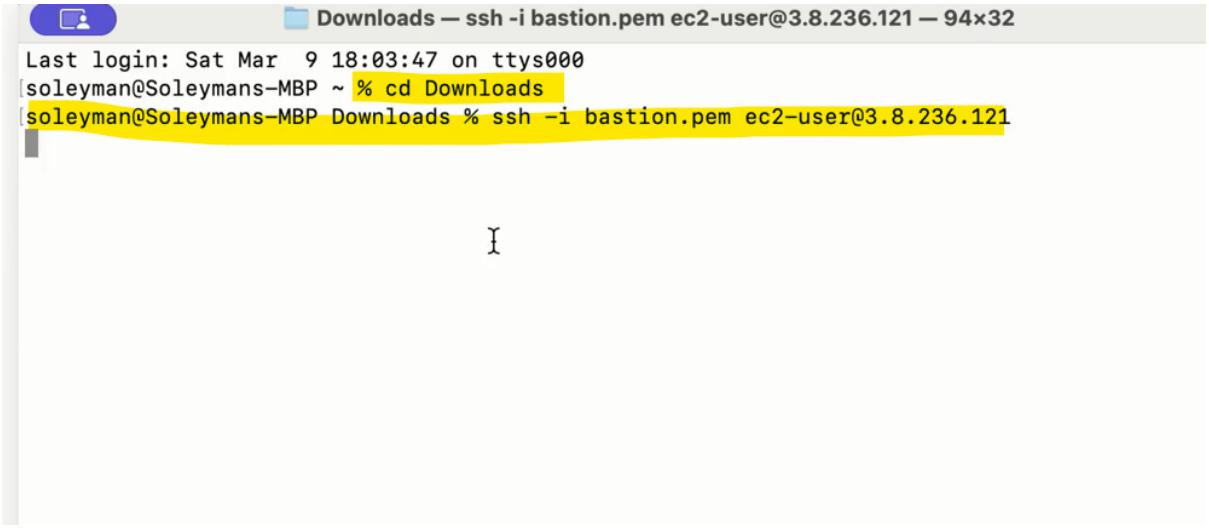
- All possible routes in the public subnet can be accessed the internet gateway and click the actual route

table.

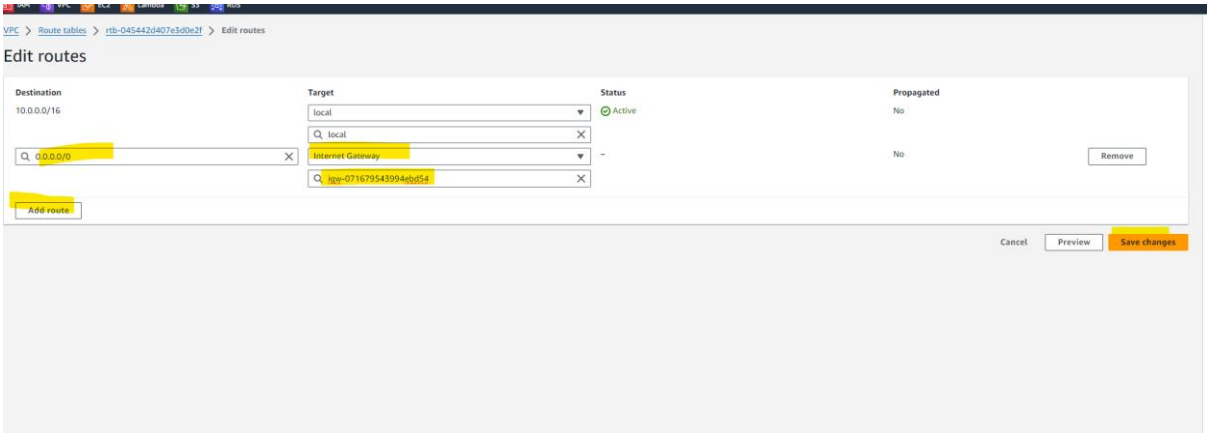


- After removing all possible routes for the Internet gateway, there isn't any route to the IGW from the public subnet. We did not detach the IGW from the VPC but there isn't any remaining route toward the IGW.

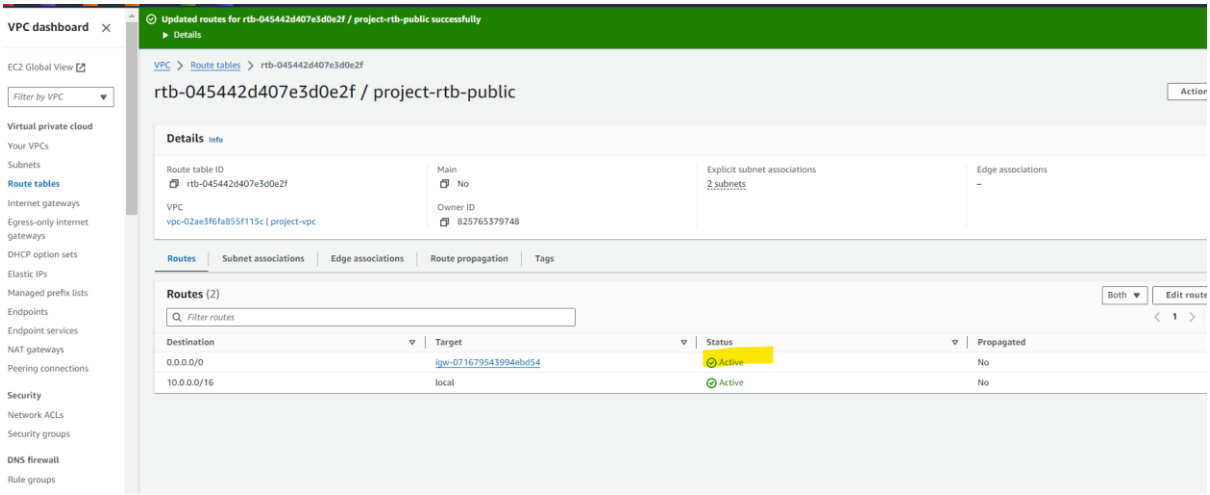




- After changing the directory to Downloads file where the key pair is located, we try again to SSH into the public subnet but it is not allowed due to the removal of all possible routes to the IGW from the public subnet.

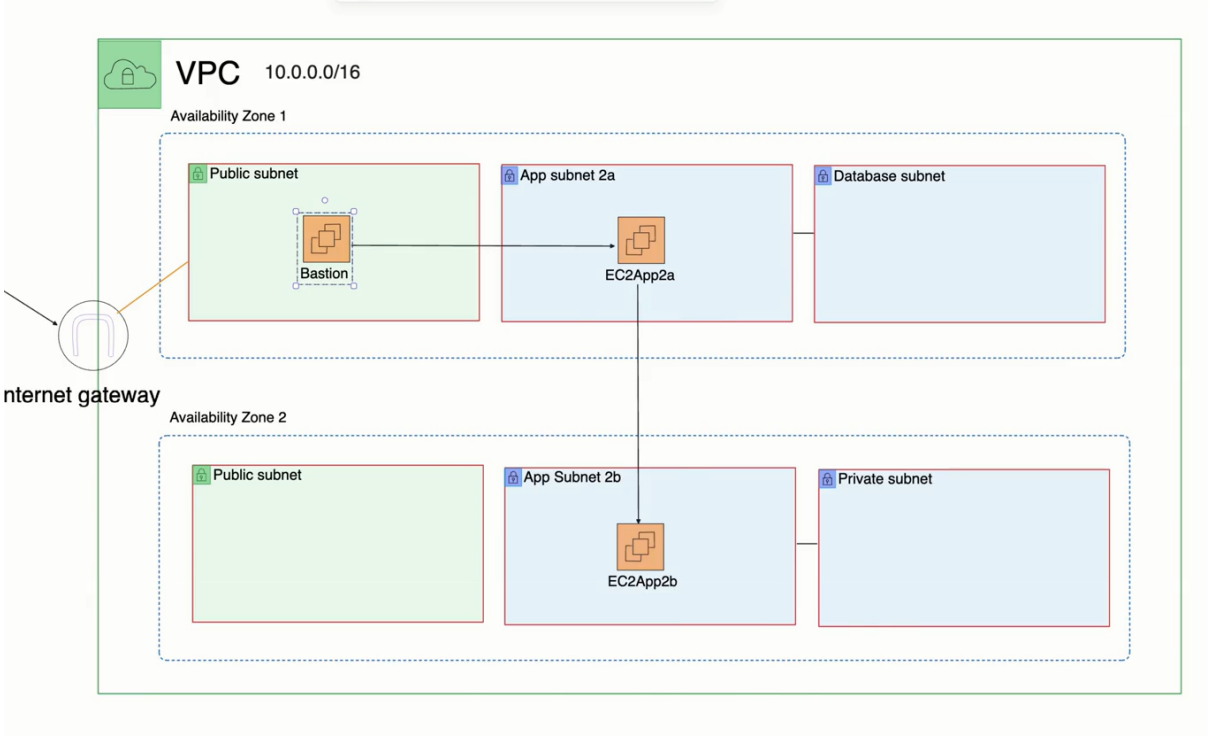


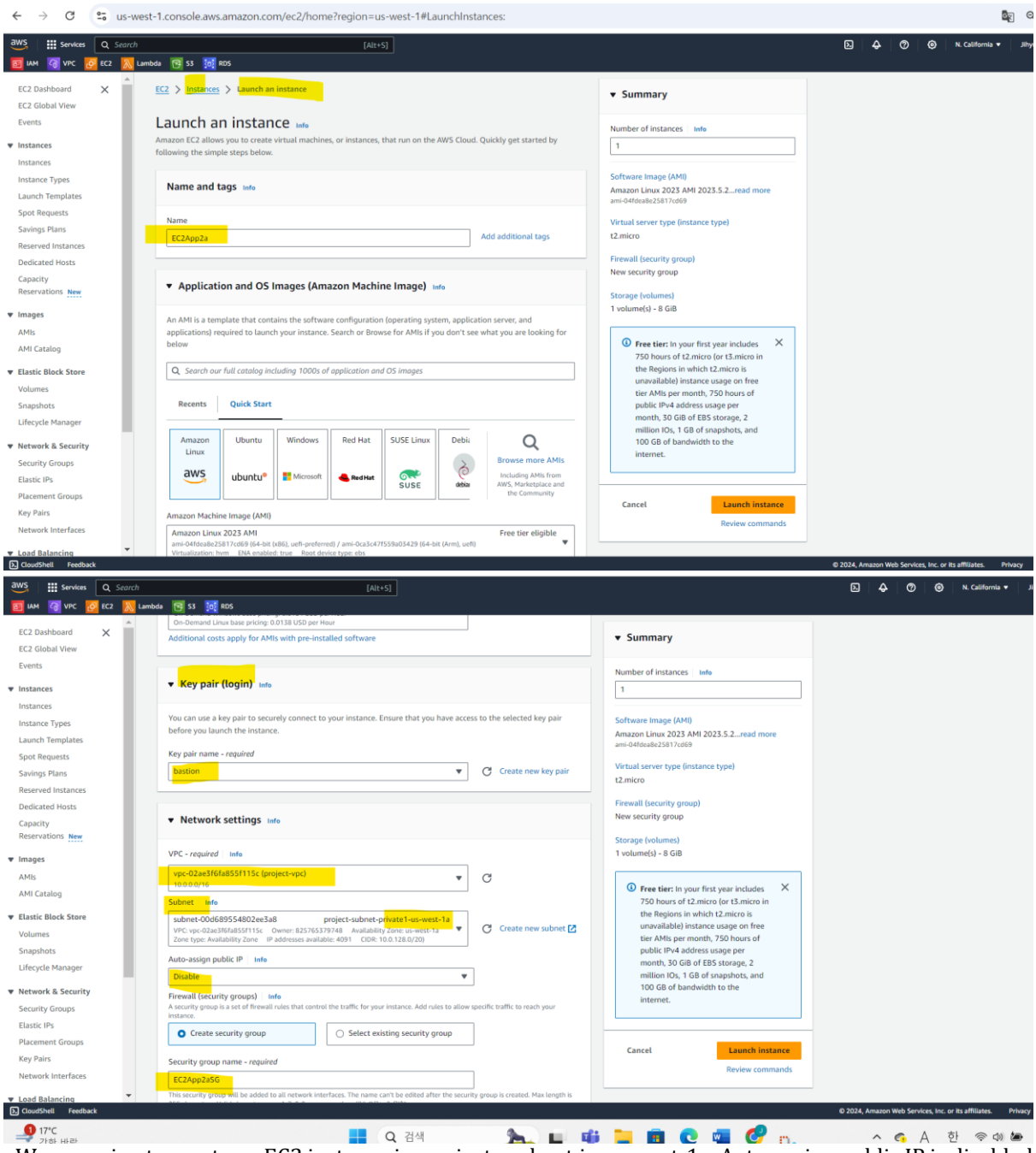
- We are now adding the route for IGW once again with the correct VPC ID.



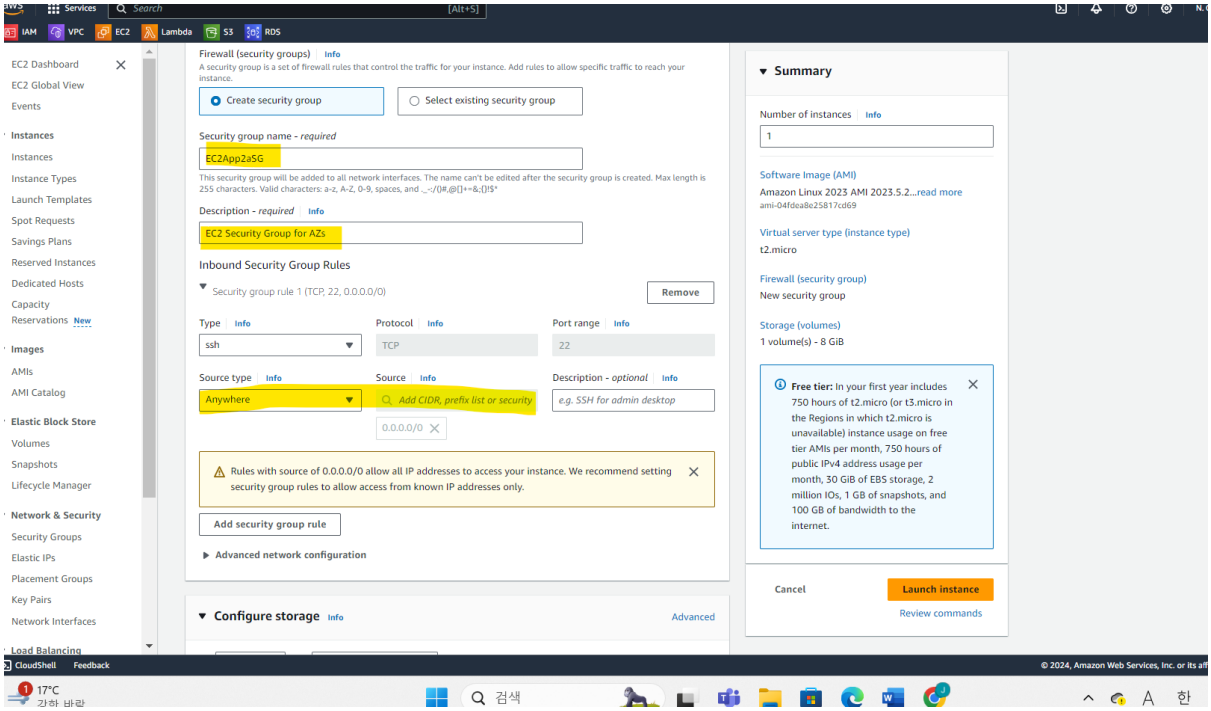
```
Downloads — ec2-user@ip-10-0-8-95:~ — ssh -i bastion.pem ec2-user@3.8.236.121 — 94×32
Last login: Sat Mar  9 18:03:47 on ttys000
soleyman@Soleymans-MBP ~ % cd Downloads
soleyman@Soleymans-MBP Downloads % ssh -i bastion.pem ec2-user@3.8.236.121
ssh: connect to host 3.8.236.121 port 22: Operation timed out
soleyman@Soleymans-MBP Downloads % ssh -i bastion.pem ec2-user@3.8.236.121
       #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
Last login: Sat Mar  9 14:04:29 2024 from 2.51.107.15
[ec2-user@ip-10-0-8-95 ~]$
```

v



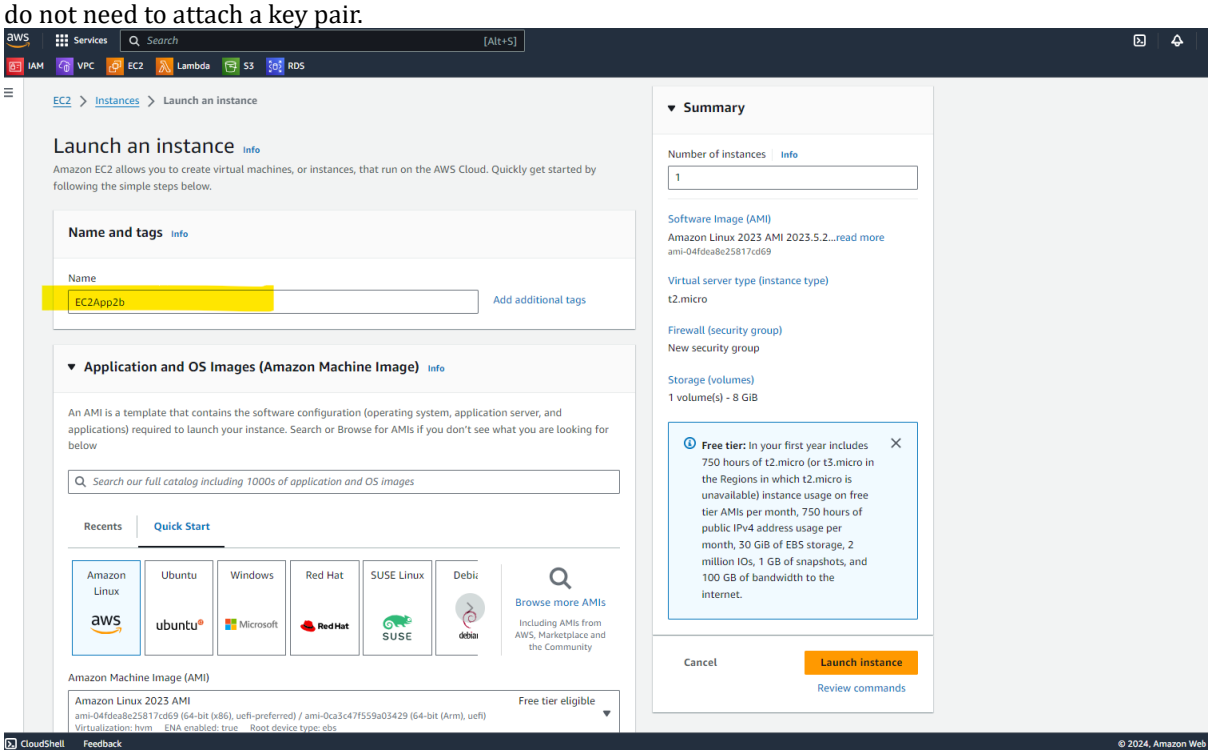


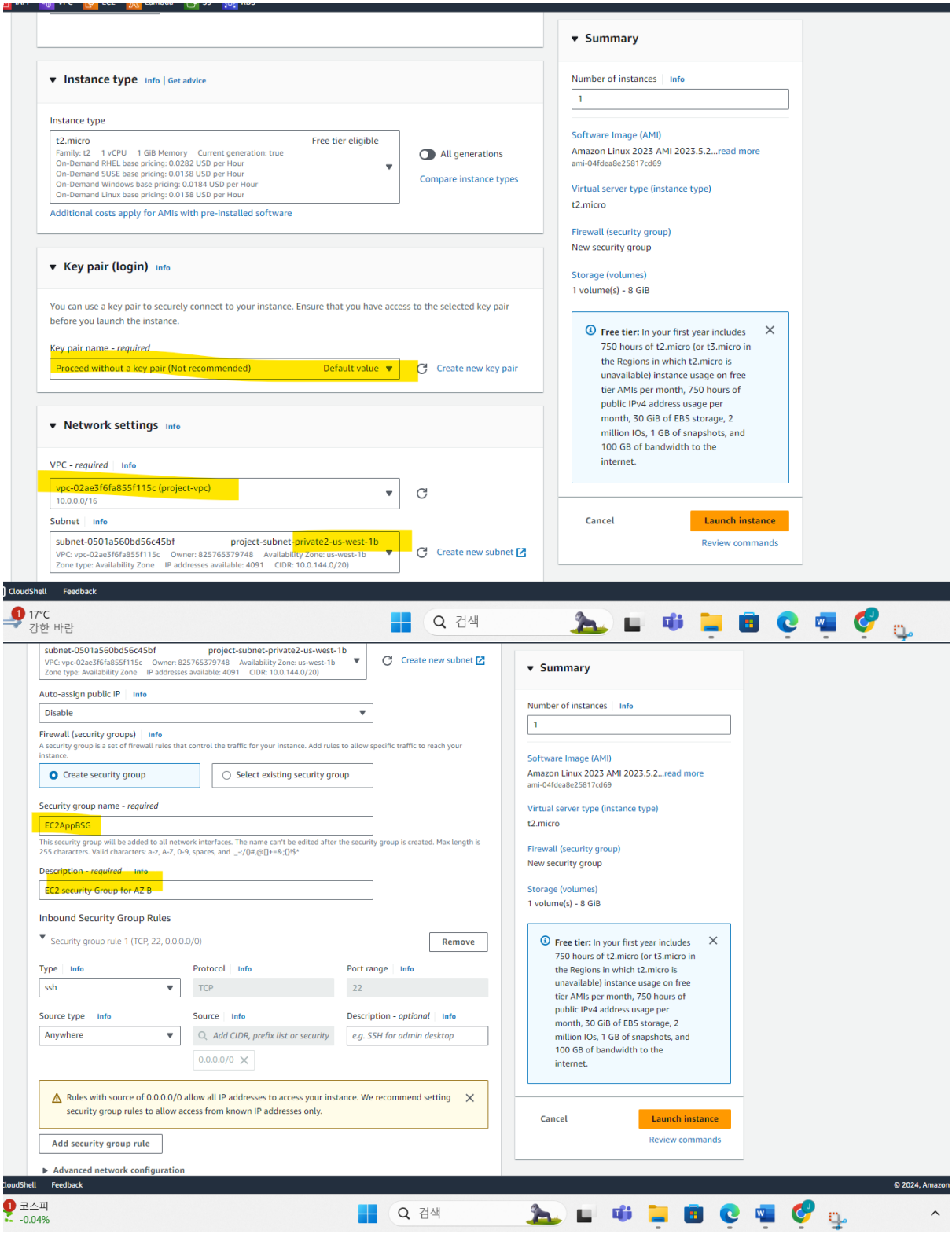
- We are going to create an EC2 instance in a private subnet in us-west-1a. Auto-assign public IP is disabled and in case of a key pair, a new key pair should be created for the new EC2 instance in the private subnet. However, in this case, we simply attach the same key pair that we used for the bastion host.



- We are going to revise inbound security group rules soon.

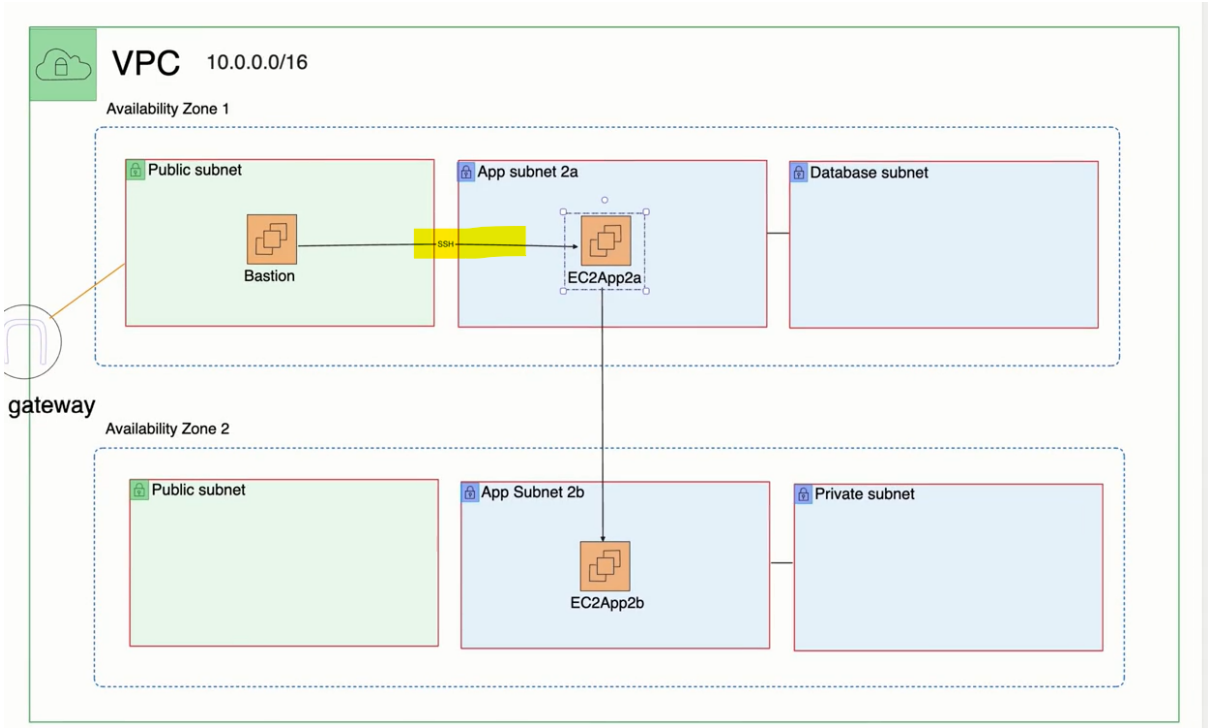
And also we should create a new EC2 instance in a different private subnet in a different AZ. In this case, we do not need to attach a key pair.

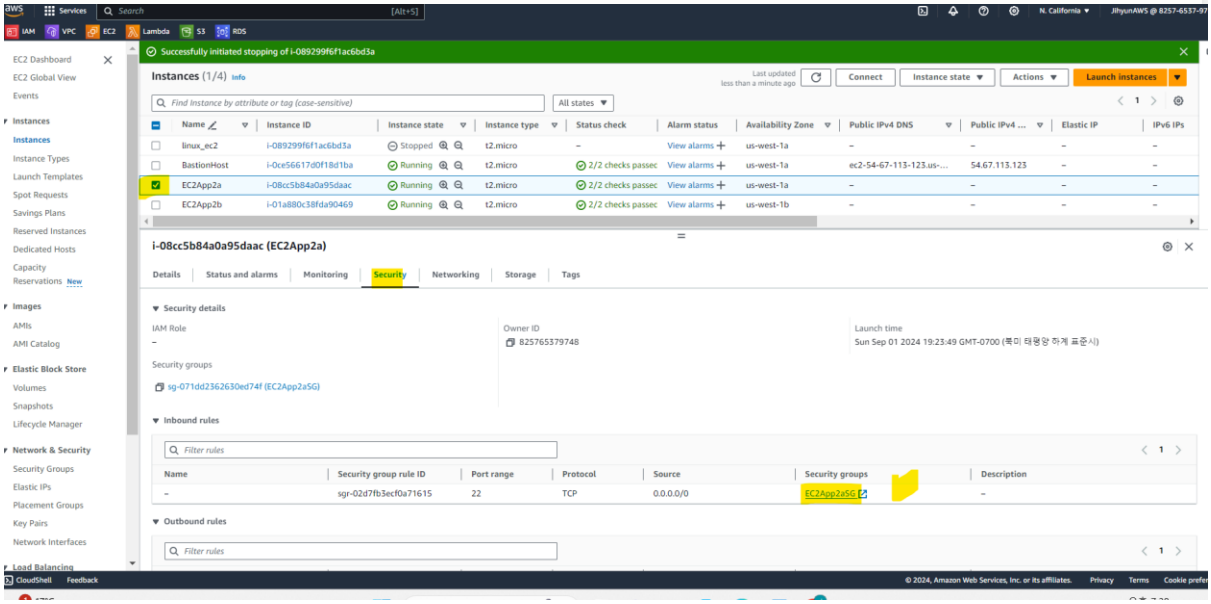


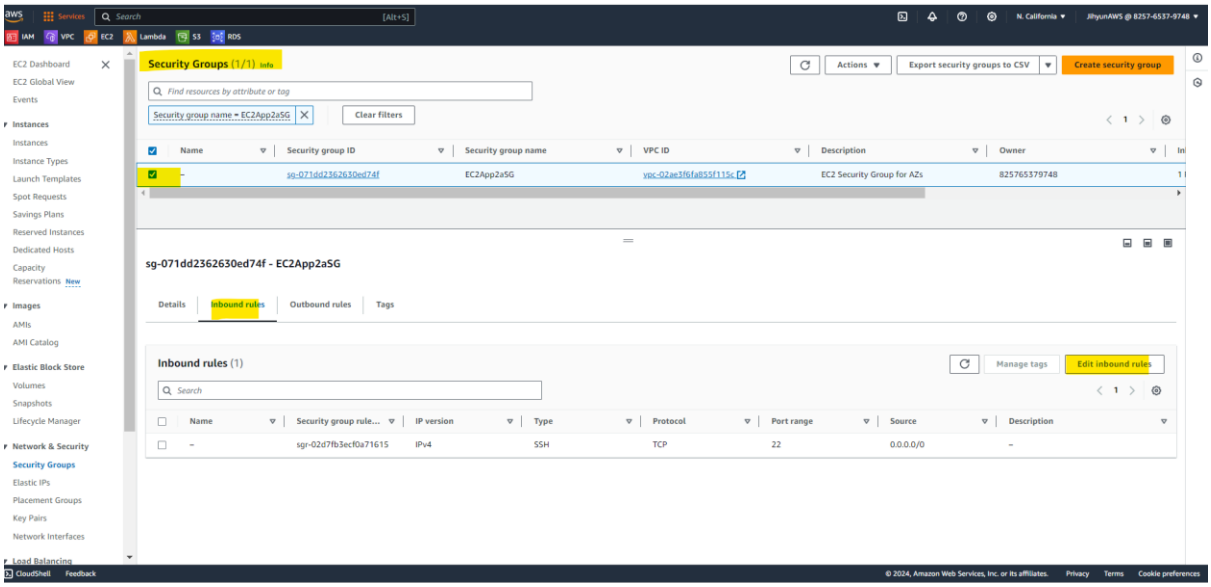


제 16강. Updating SGs

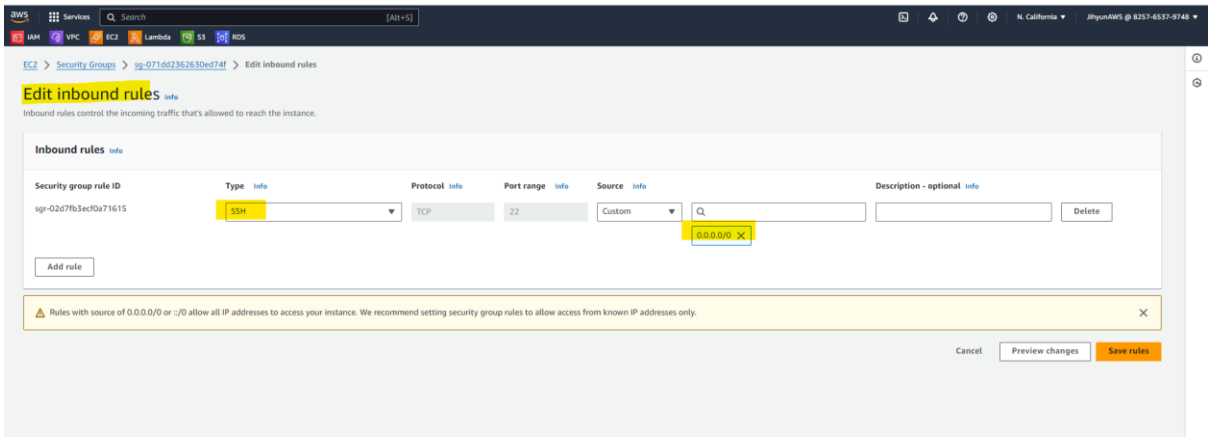
- We need to allow SSH connection from the Bastion Host to EC2App2a instance in a private subnet.


VPC
10.0.0.0/16
Availability Zone 1
Public subnet
Bastion
SSH
App subnet 2a
EC2App2a
Database subnet
gateway
Availability Zone 2
Public subnet
App Subnet 2b
EC2App2b
Private subnet

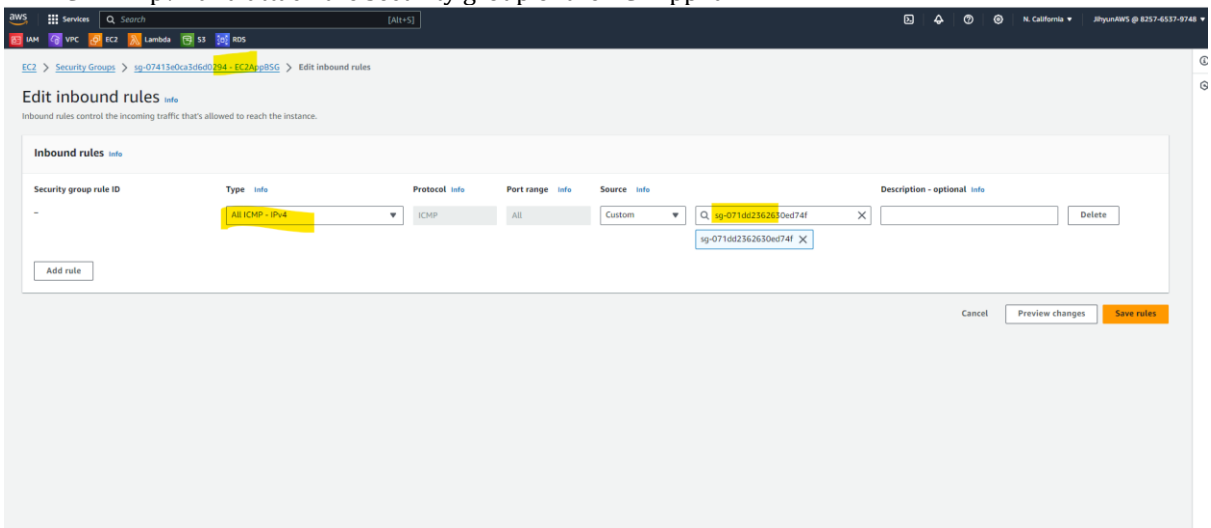

Successfully initiated stopping of i-089299f6f1ac6bd3a
Instances (1/4)
Connect
Instance state
Actions
Launch instances
linux_ec2 i-089299f6f1ac6bd3a Stopped t2.micro
BastionHost i-0ce56617d0f18d1ba Running t2.micro 2/2 checks passed us-west-1a 54.67.113.123
EC2App2a i-08cc5b84a0a95daac Running t2.micro 2/2 checks passed us-west-1a
EC2App2b i-01a880c38fda90469 Running t2.micro 2/2 checks passed us-west-1b
i-08cc5b84a0a95daac (EC2App2a)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Security details
IAM Role
Owner ID
825765379748
Launch time
Security groups
sg-071dd2362630ed74f (EC2App2aSG)
Inbound rules
Name
Security group rule ID
Port range
Protocol
Source
Security groups
Description
sgr-02d7fb3ecf0a71615
22
TCP
0.0.0.0/0
EC2App2aSG
Outbound rules



- When editing inbound rules, the default mode is that any traffic 0.0.0.0/0 can SSH into the EC2App2a instance. Just delete this rule and add a new rule, including a security group of the Bastion Host or a private IP address for the Bastion Host.

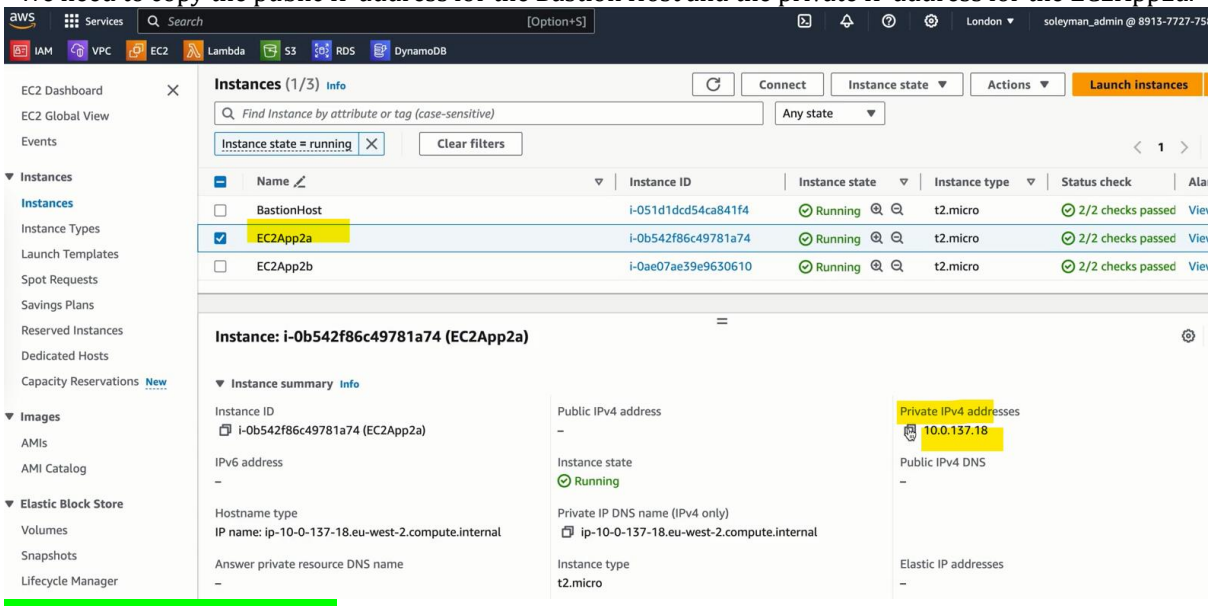


- In the 2nd EC2 instance in a different AZ, we do not need to SSH into the EC2App2b from the EC2APP2a. We just ping a requset (IP address) to the EC2App2b and get a response from it. The security group type is ALL ICMP – Ipv4 and attach the security group of the EC2App2a.



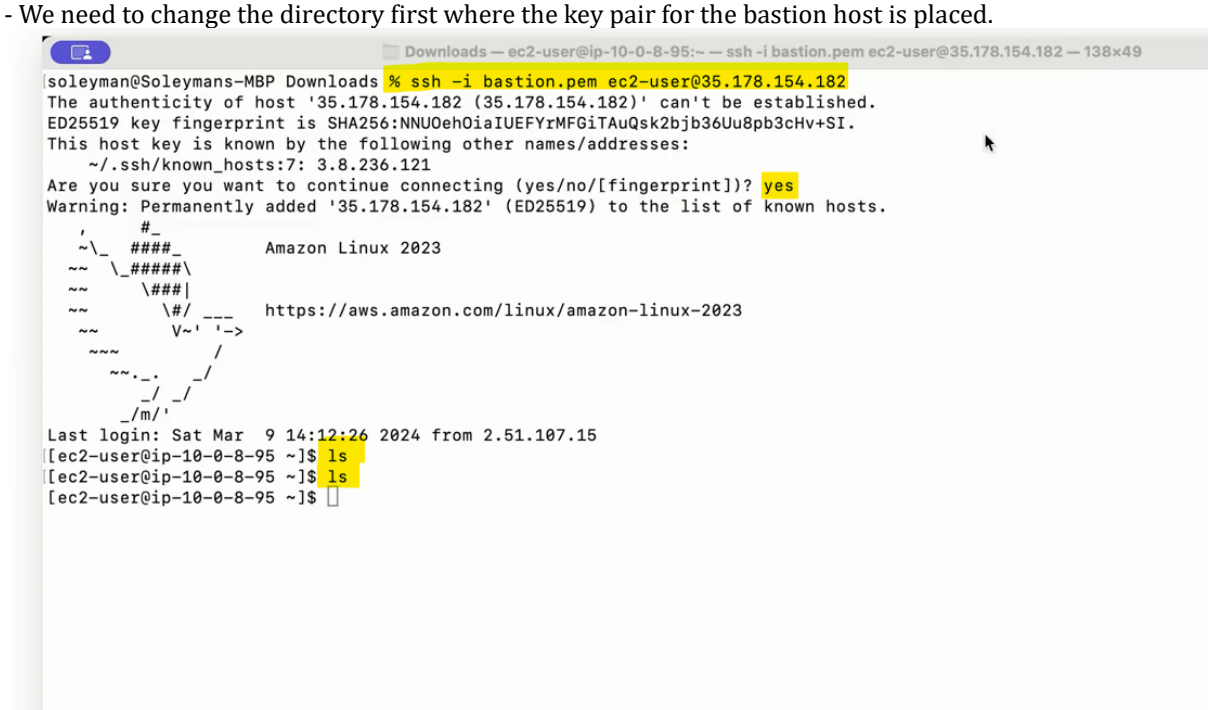
제 17강. Copying keypairs and pinging requests.

- We pass the key file to the bastion host placed in the public subnet to SSH into it via the Internet gateway.
- We copy the key file and pass it over to the EC2App2a in a private subnet in the AZ 1 to connect to this private instance via the bastion host.
- We need to copy the public IP address for the Bastion Host and the private IP address for the EC2App2a.



(1) SSH into the Bastion host.

- We need to change the directory first where the key pair for the bastion host is placed.



- We passed the key pair, bastion.pem to the EC2 user with the public IP address of the Bastion Host. And when we put ls, there is nothing in the Bastion Host. At this time, the personal local IP address must be added into the inbound rule of the Bastion Host.
- After loging into the Bastion Host, we need to copy the key pair placed in our local machine to the Bastion host, opening a new tab on CLI and type this command to copy the key pair in your local machine.
- scp -i<your_bastion_key_file> <give a new key name "private.pem"> ec2-user@<bastion-public-ip>:~/
-After this, get back to the original CLI and enter ls and there should be the copied key file on your Bastion host.

```
        _/m/'
Last login: Sat Mar  9 14:12:26 2024 from 2.51.107.15
[ec2-user@ip-10-0-8-95 ~]$ ls
[ec2-user@ip-10-0-8-95 ~]$ ls
[ec2-user@ip-10-0-8-95 ~]$ ls
bastion.pem
[ec2-user@ip-10-0-8-95 ~]$
```

- After this, we are going to log into the EC2App2a instance.

$ ssh -I bastion.pem ec2-user@<the private IP adress for the private instance>

```
[ec2-user@ip-10-0-8-95 ~]$ ls
[ec2-user@ip-10-0-8-95 ~]$ ls
[ec2-user@ip-10-0-8-95 ~]$ ls
bastion.pem
[ec2-user@ip-10-0-8-95 ~]$ ssh -i bastion.pem ec2-user@10.0.137.18
The authenticity of host '10.0.137.18 (10.0.137.18)' can't be established.
ED25519 key fingerprint is SHA256:+qhLVyn5TJoZi/lkHdobj2F6yuPUfi7BItl6G+elH2g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.137.18' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-137-18 ~]$
```

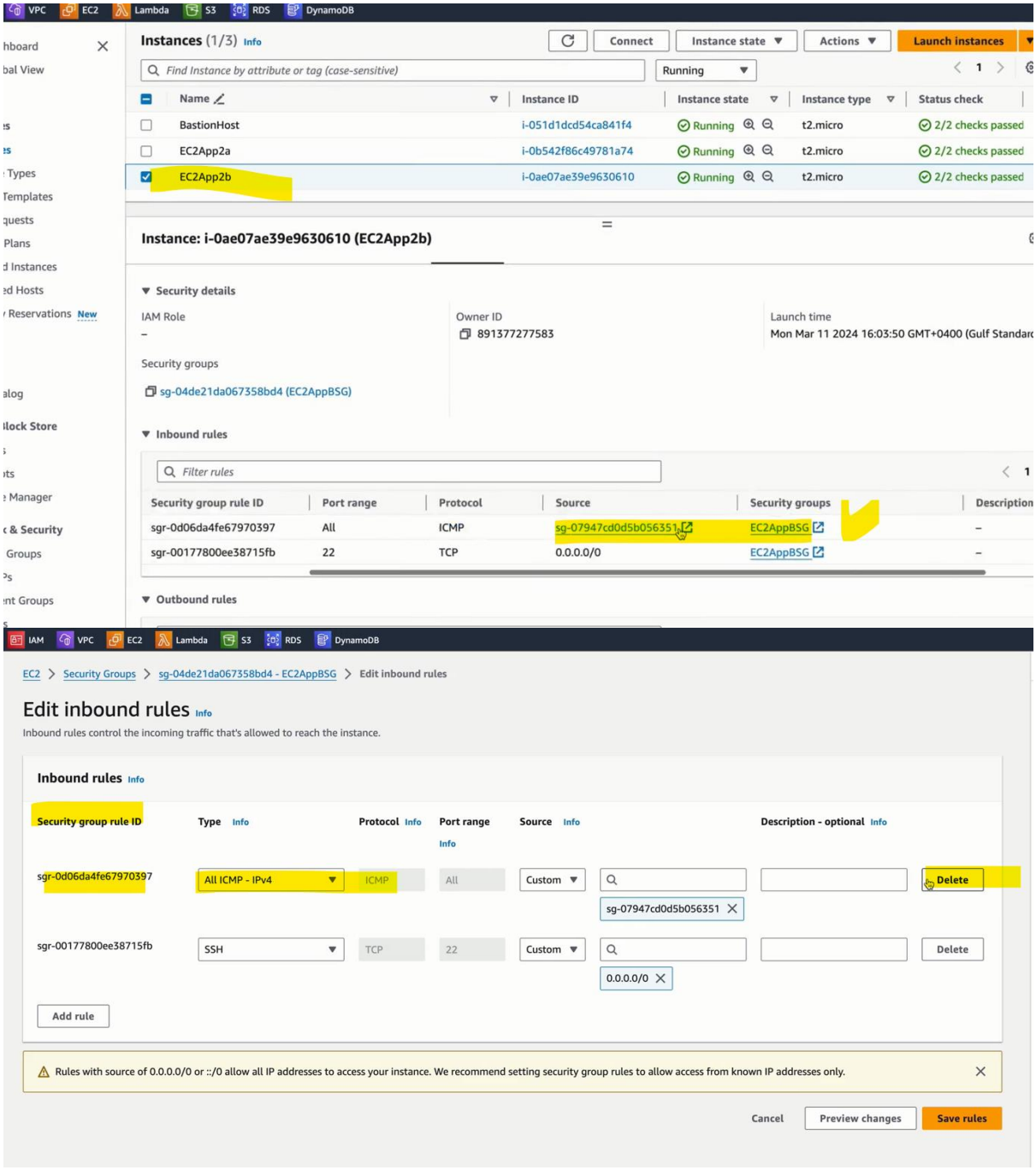
- Next, we should ping the 3rd private EC2 instance IP address

$ ping <the private IP adress for the 3rd EC2 instance in another private subnet> and then we can see the responses back from the 3rd EC2 instance.

```
bastion.pem
[ec2-user@ip-10-0-8-95 ~]$ ssh -i bastion.pem ec2-user@10.0.137.18
The authenticity of host '10.0.137.18 (10.0.137.18)' can't be established.
ED25519 key fingerprint is SHA256:+qhLVyn5TJoZi/lkHdobj2F6yuPUfi7BItl6G+elH2g.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.137.18' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-137-18 ~]$ ping 10.0.146.173
PING 10.0.146.173 (10.0.146.173) 56(84) bytes of data.
64 bytes from 10.0.146.173: icmp_seq=1 ttl=127 time=1.39 ms
64 bytes from 10.0.146.173: icmp_seq=2 ttl=127 time=1.04 ms
64 bytes from 10.0.146.173: icmp_seq=3 ttl=127 time=1.05 ms
64 bytes from 10.0.146.173: icmp_seq=4 ttl=127 time=1.14 ms
64 bytes from 10.0.146.173: icmp_seq=5 ttl=127 time=1.10 ms
64 bytes from 10.0.146.173: icmp_seq=6 ttl=127 time=1.24 ms
64 bytes from 10.0.146.173: icmp_seq=7 ttl=127 time=1.19 ms
64 bytes from 10.0.146.173: icmp_seq=8 ttl=127 time=1.02 ms
64 bytes from 10.0.146.173: icmp_seq=9 ttl=127 time=1.03 ms
64 bytes from 10.0.146.173: icmp_seq=10 ttl=127 time=1.05 ms
64 bytes from 10.0.146.173: icmp_seq=11 ttl=127 time=0.966 ms
```

If you want to end the ping process, press ctrl+c

What happens if we remove the inbound rule of the EC2App2b to prevent the ping?



- In the inbound rule of the security group for the EC2App2b, we removed the rule allowing the ping from the EC2App2a.

```
64 bytes from 10.0.146.173: icmp_seq=4 ttl=127 time=1.14 ms
64 bytes from 10.0.146.173: icmp_seq=5 ttl=127 time=1.10 ms
64 bytes from 10.0.146.173: icmp_seq=6 ttl=127 time=1.24 ms
64 bytes from 10.0.146.173: icmp_seq=7 ttl=127 time=1.19 ms
64 bytes from 10.0.146.173: icmp_seq=8 ttl=127 time=1.02 ms
64 bytes from 10.0.146.173: icmp_seq=9 ttl=127 time=1.03 ms
64 bytes from 10.0.146.173: icmp_seq=10 ttl=127 time=1.05 ms
64 bytes from 10.0.146.173: icmp_seq=11 ttl=127 time=0.966 ms
64 bytes from 10.0.146.173: icmp_seq=12 ttl=127 time=1.82 ms
64 bytes from 10.0.146.173: icmp_seq=13 ttl=127 time=0.988 ms
^C
--- 10.0.146.173 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12017ms
rtt min/avg/max/mdev = 0.966/1.156/1.815/0.221 ms
[ec2-user@ip-10-0-137-18 ~]$ ping 10.0.146.173
```

- And in the CLI, we ping againg to the 3rd EC2 instance due to the fact that the security group of the inbound rule for this process was removed.

```
13 packets transmitted, 13 received, 0% packet loss, time 12017ms
rtt min/avg/max/mdev = 0.966/1.156/1.815/0.221 ms
[ec2-user@ip-10-0-137-18 ~]$ ping 10.0.146.173
PING 10.0.146.173 (10.0.146.173) 56(84) bytes of data.
^C
--- 10.0.146.173 ping statistics ---
32 packets transmitted, 0 received, 100% packet loss, time 32215ms

[ec2-user@ip-10-0-137-18 ~]$
```

제 18강. Project Wrap UP and Terminate Instances