

멋쟁이사자처럼 리눅스 세션

17년 11월 막주차

01

목표

1. AWS VPC, RDS, EC2 같이 사용해보기

01

목표

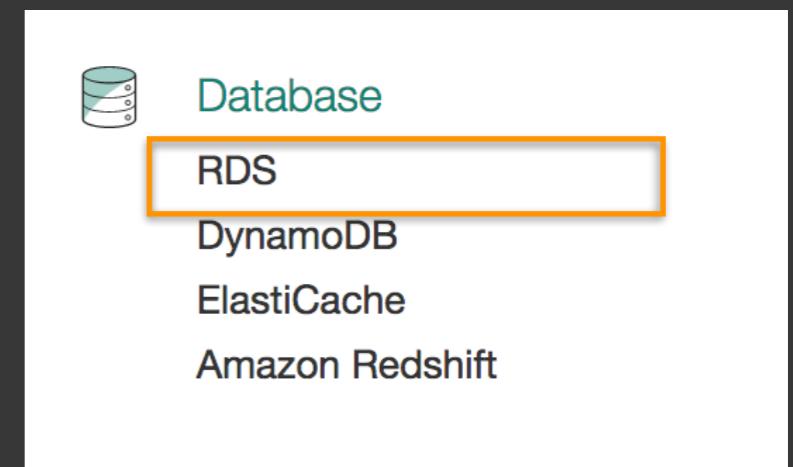
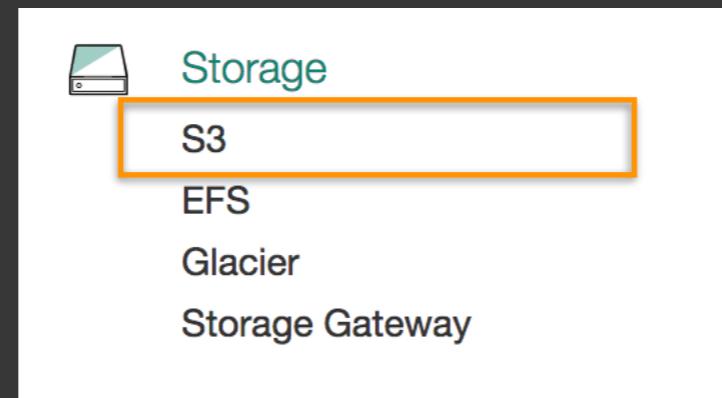
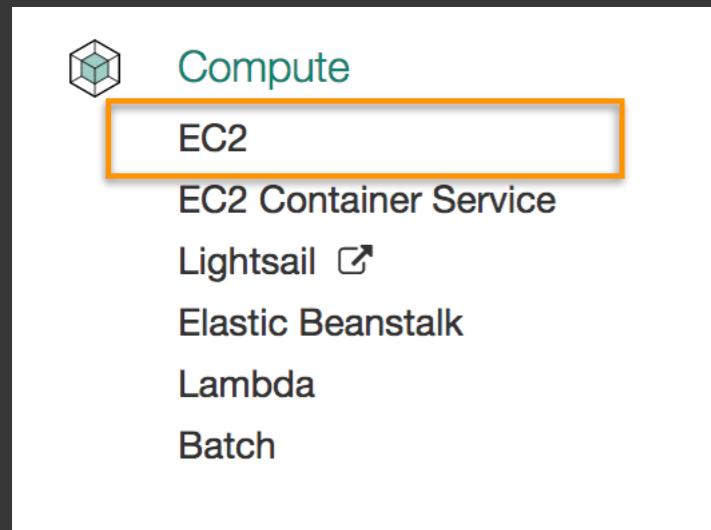


AWS VPC, RDS, EC2 같이 사용해보기

지난주에 배운 개념을 바탕으로 직접 실습을 해보자!

02

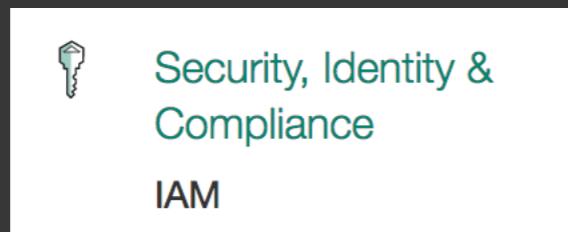
Amazon - Architecture



Elastic Compute Cloud

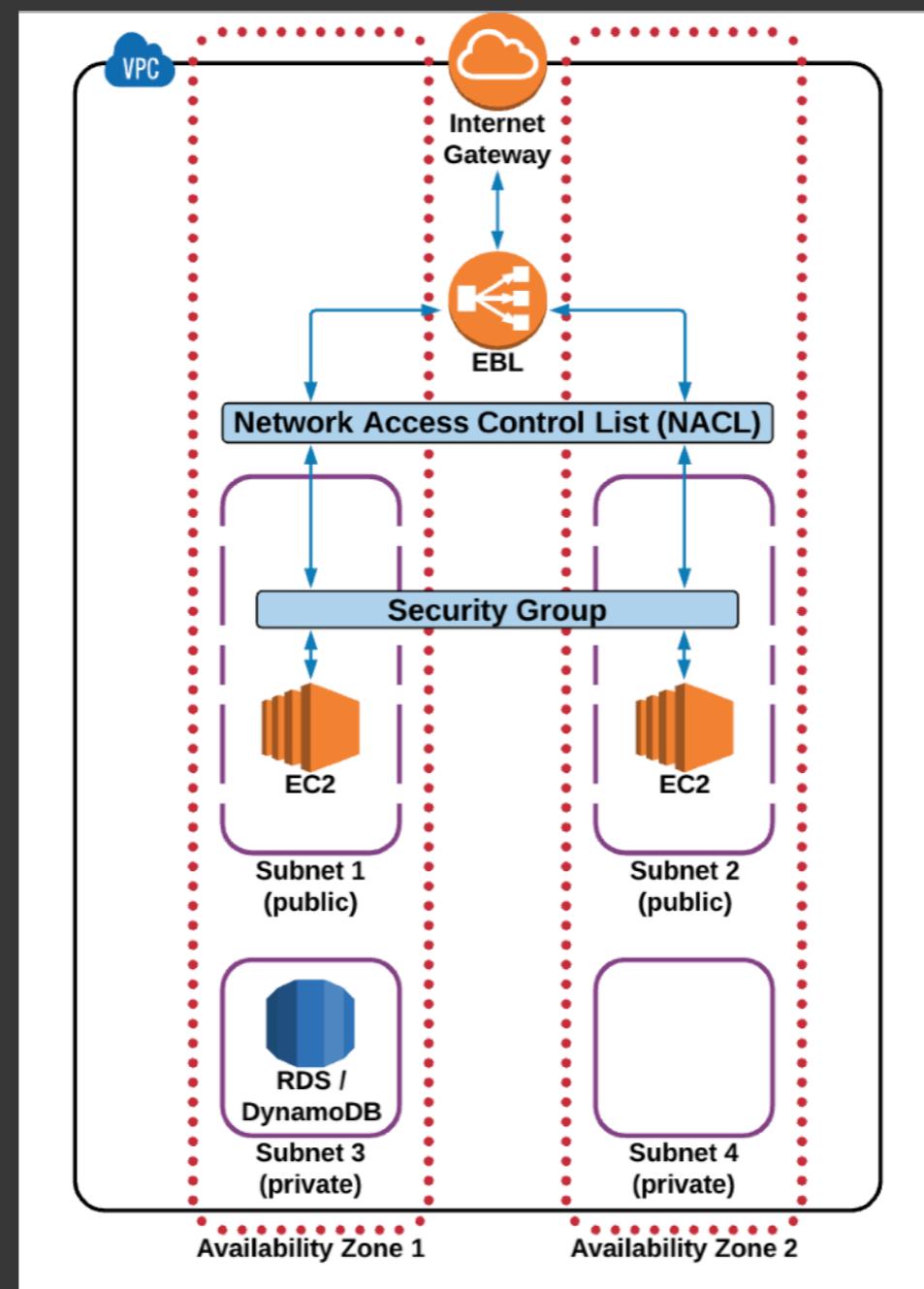
Simple Storage Service

Relational Database Service

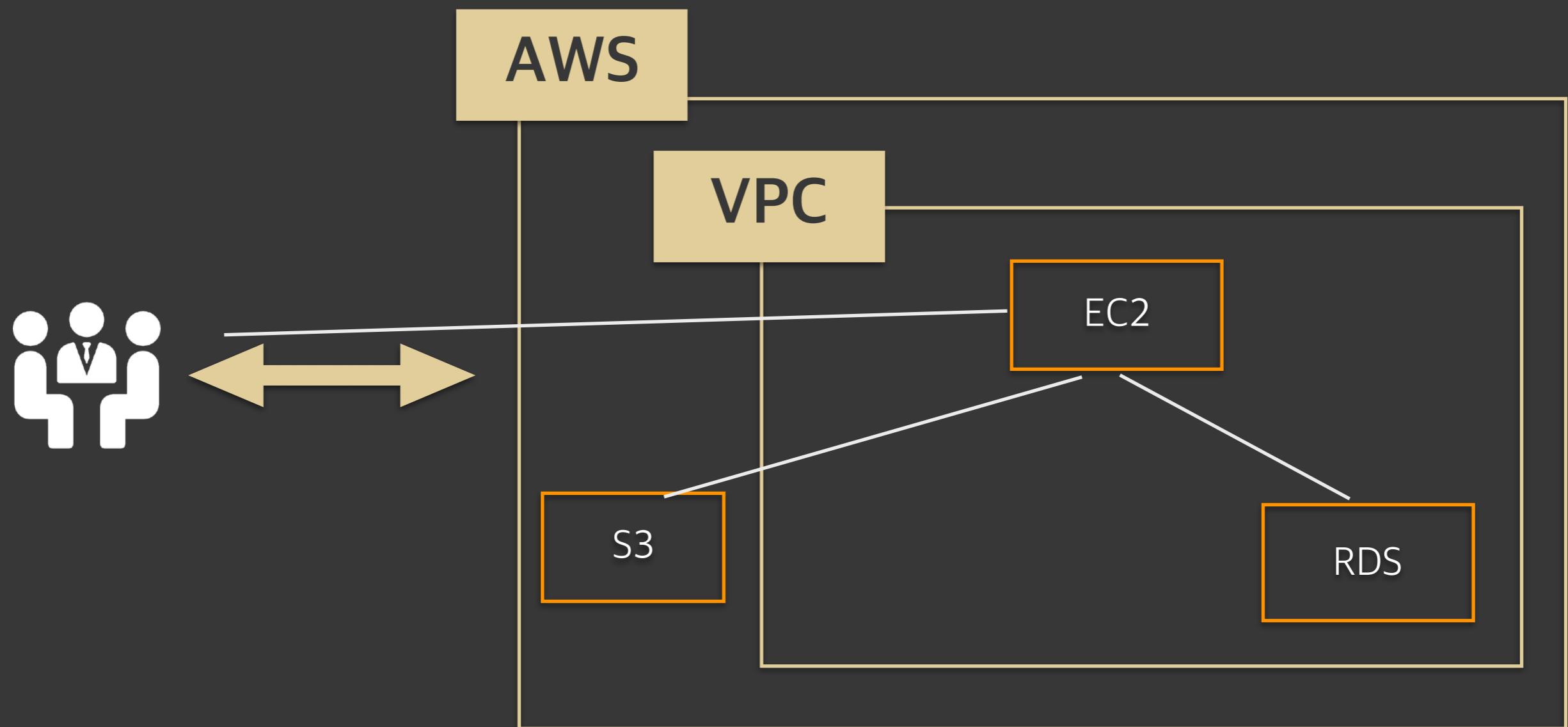


Identity & Access Mgt

Amazon - Architecture

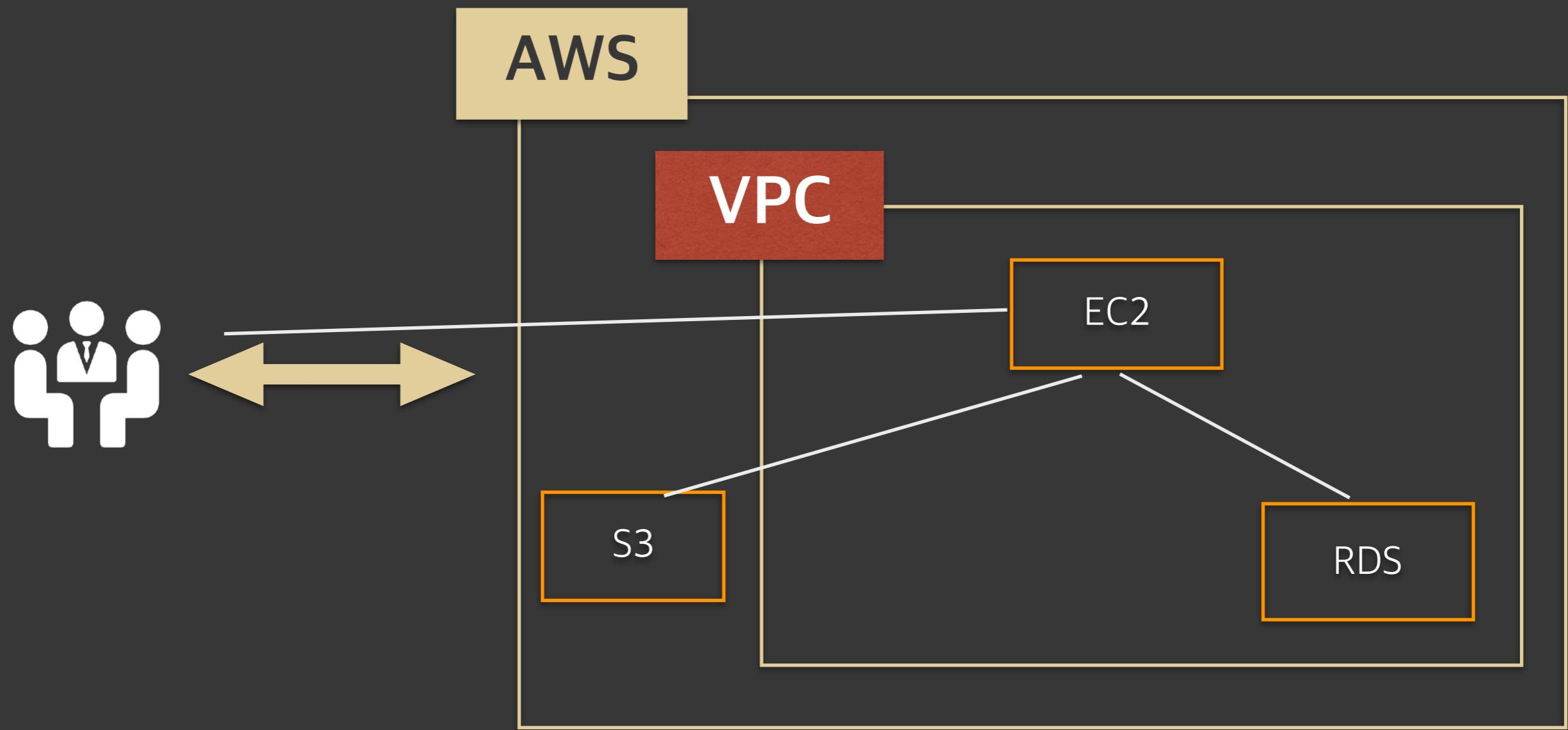


Amazon - Architecture



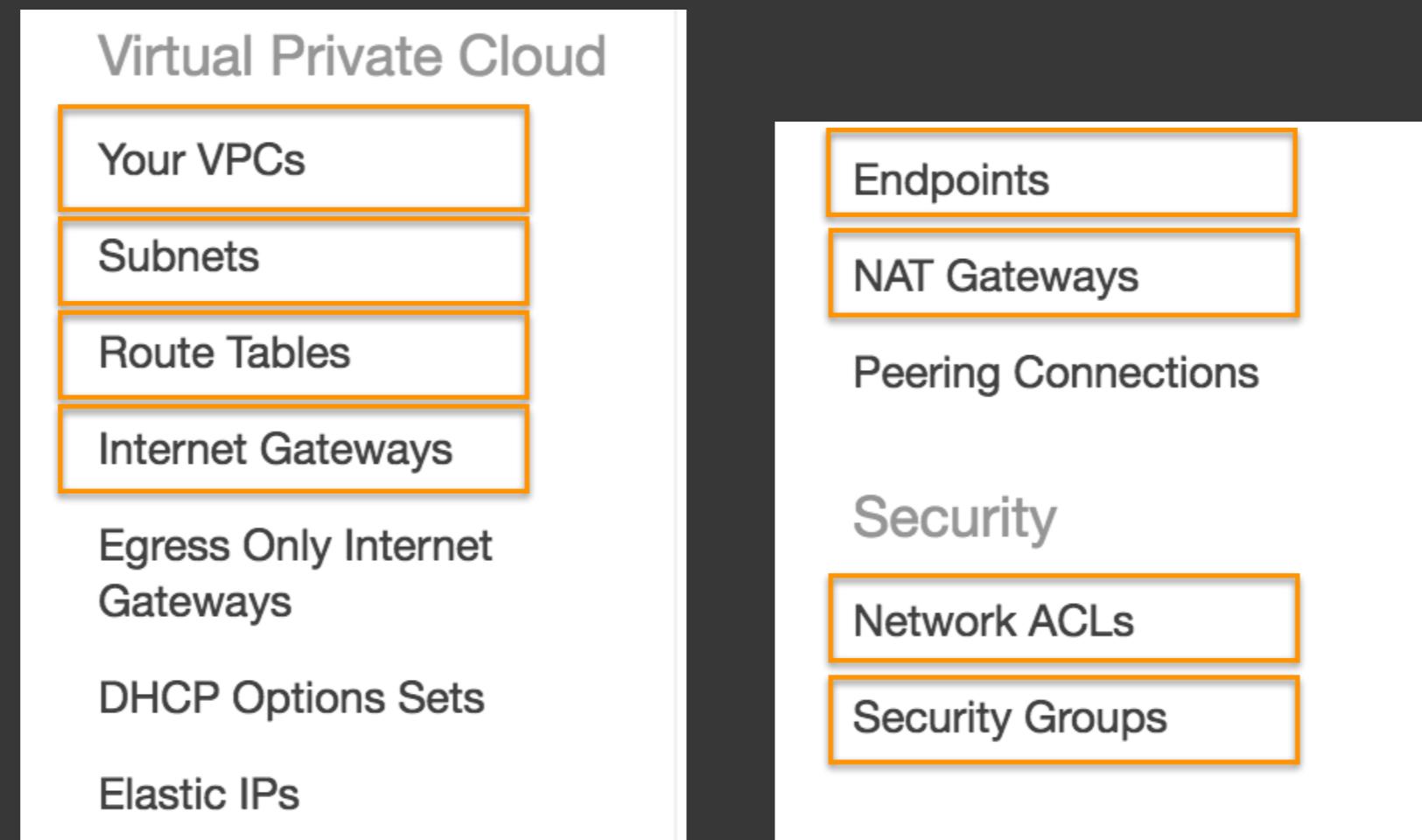
02

VPC 만들기



02

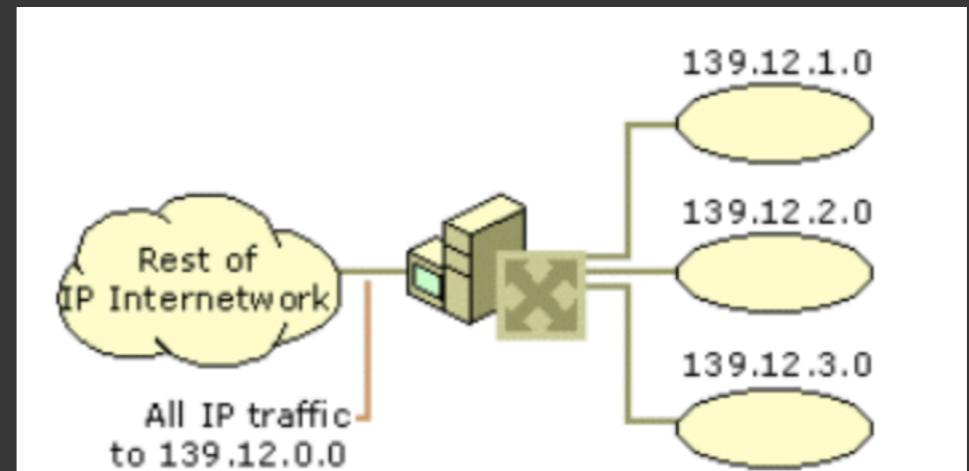
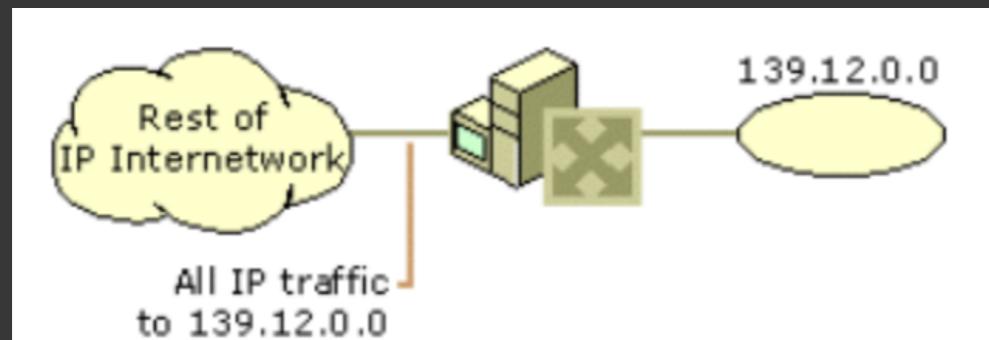
VPC 만들기



02

VPC 만들기 - subnet

Subnet

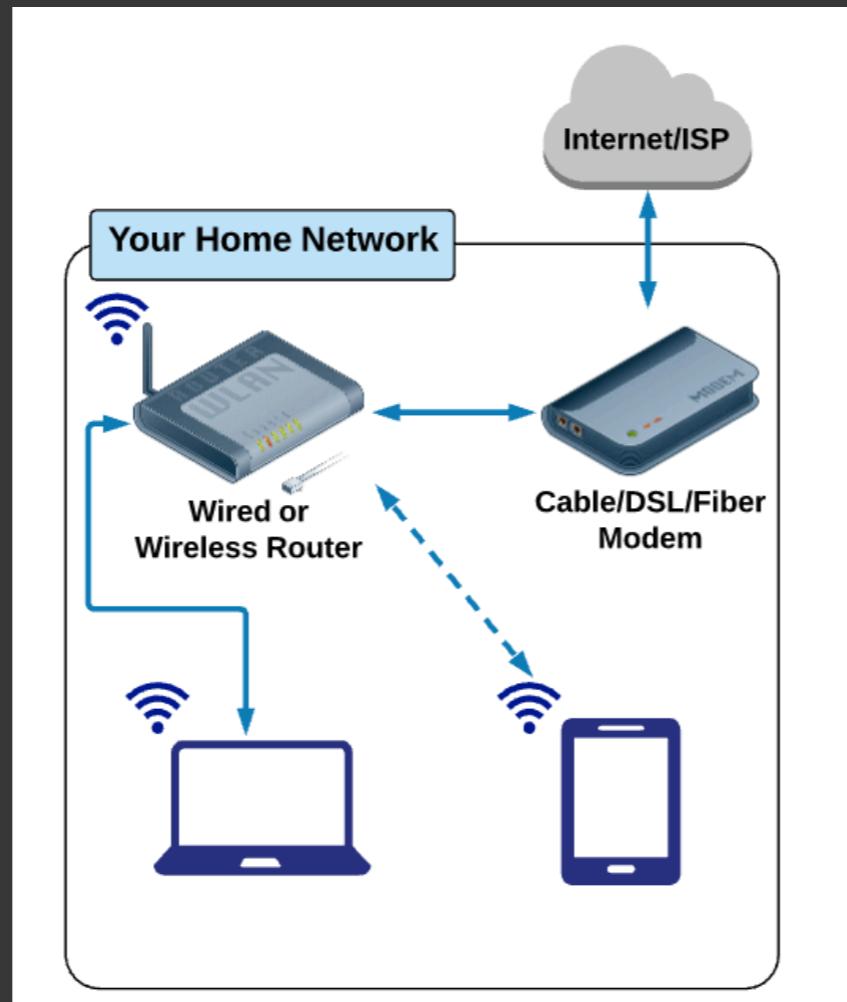


한정적인 IP 주소를 많은 기기에서 쓰기 위한 방법!

02

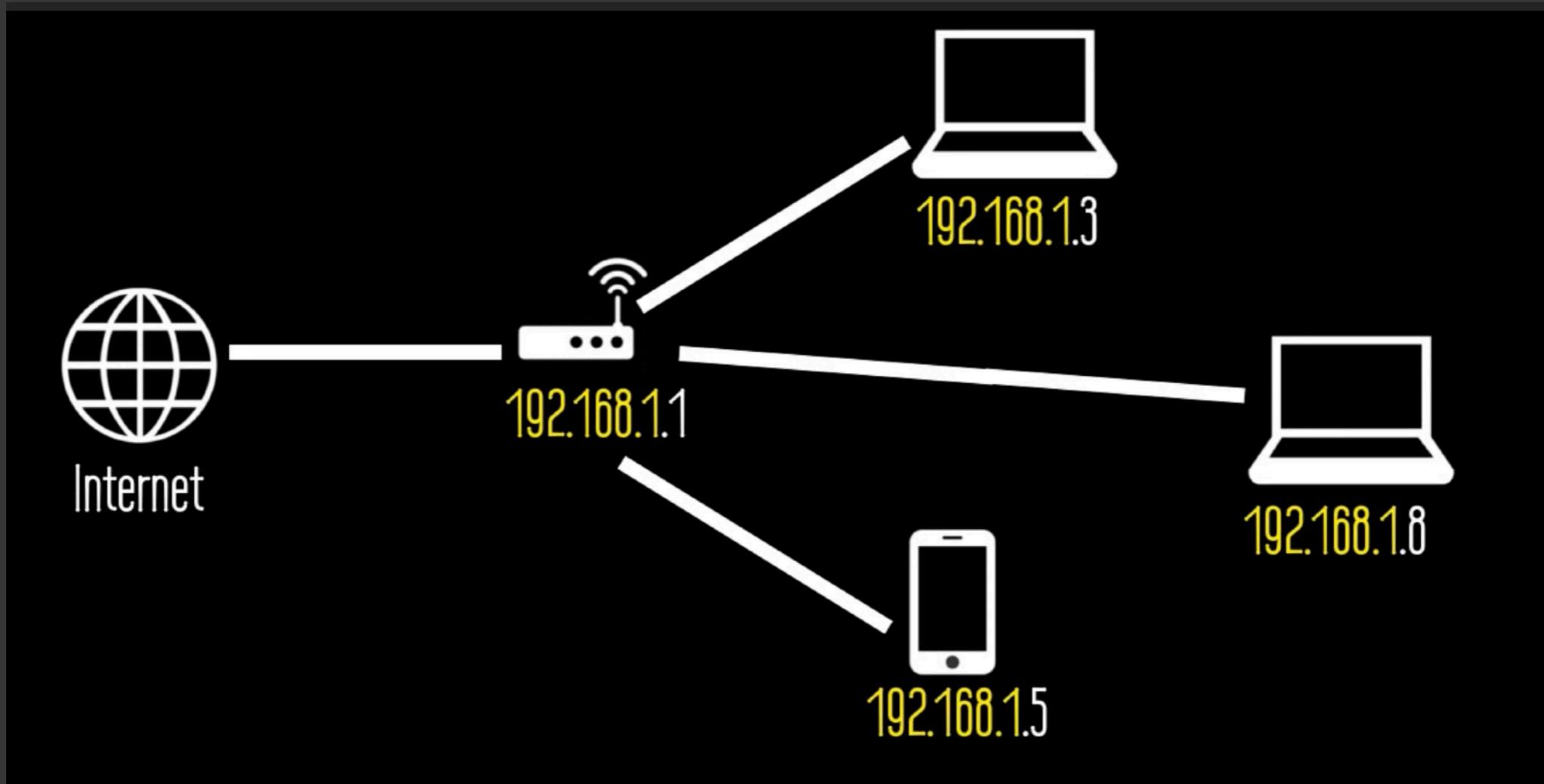
VPC 만들기 - subnet

Home Network



02

VPC 만들기 - subnet



Network portion / host portion (unique)

02

VPC 만들기 - subnet

Subnet Mask

어떤 host인지를 알 수 있게 해주는 애

32bit로 이루어진 IP address = 8bit * 4개

00000000 . 00000000 . 00000000 . 00000000

10. 10. 2. 1

000000101 . 00000101 . 00000010 . 00000001

&&

255. 255. 255. 0

subnet mask: 11111111 . 11111111 . 11111111 . 00000000

02

VPC 만들기 - subnet

10. 10. 2. 1

000000**101** . 00000**101** . 000000**10** . 0000000**1**

&&

255. 255. 255. 0

subnet mask: 11111111 . 11111111 . 11111111 . 00000000

10. 10. 2. 0

10. 10. 2 > network portion

0 > host portion

즉, 8bit만큼 = 2^8 만큼의 host를 가질 수 있는 것

02

VPC 만들기

Subnet

VPC의 IP 주소 범위
선택한 subnet안에서 AWS 리소스 시작

Public Subnet

인터넷에 연결되어야 하는 리소스

Private Subnet

인터넷에 연결될 필요없는 리소스

02

VPC 만들기

우리가 할 일

서버에 대한 공개적인 access 차단

+

퍼블릭 웹 애플리케이션을 실행

Public Subnet

Private Subnet

인터넷에 연결되어야 하는
Web Server

인터넷에 연결되면 안되는
Database Server

02

VPC 만들기

Public Subnet

인터넷에 연결되어야 하는
Web Server

Outbound Traffic을 이용
인터넷으로 직접전송

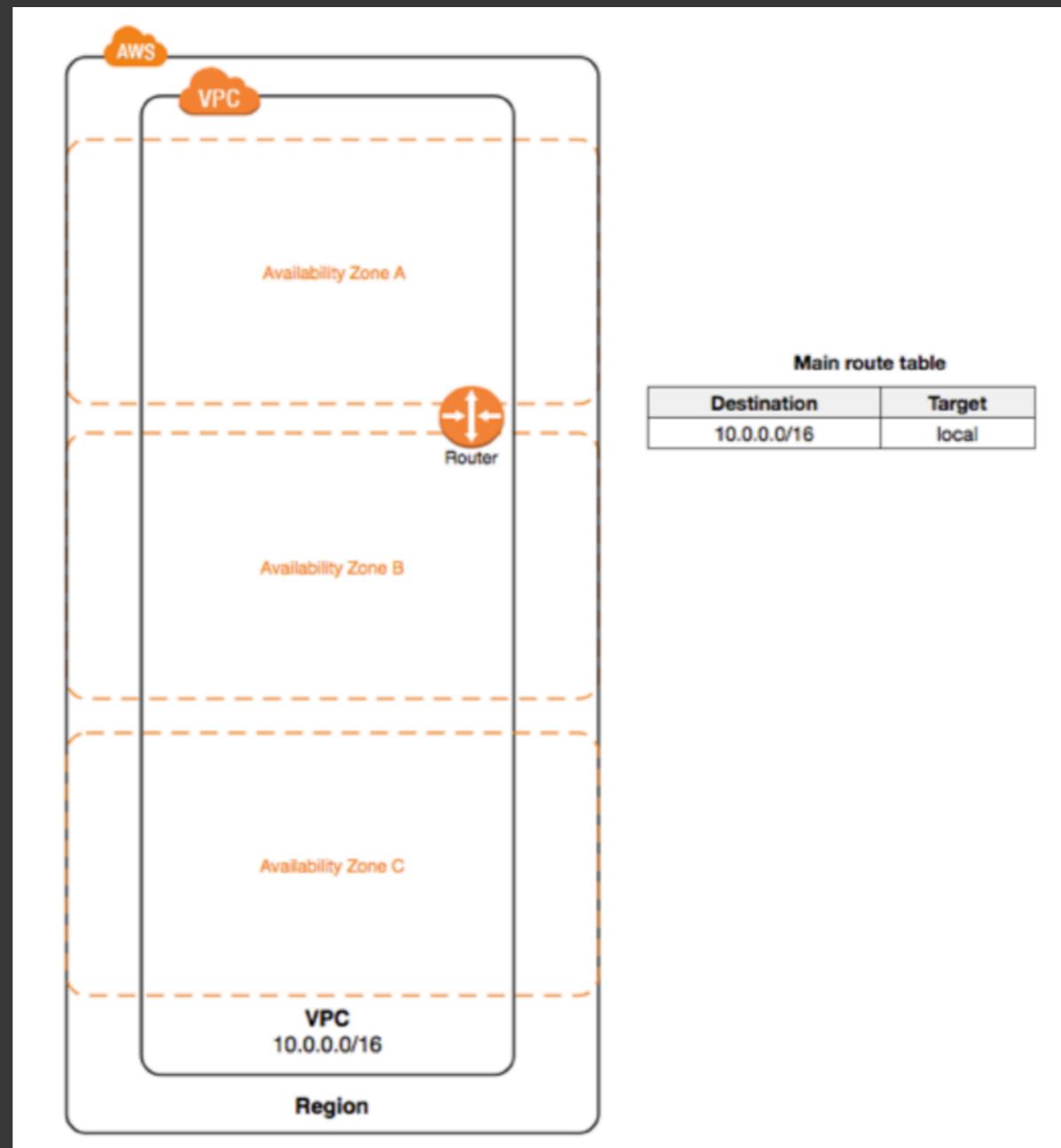
Private Subnet

인터넷에 연결되면 안되는
Database Server

인터넷 직접전송 불가능
NAT(네트워크 주소변환)
게이트웨이를 사용하여
인터넷에 엑세스 가능

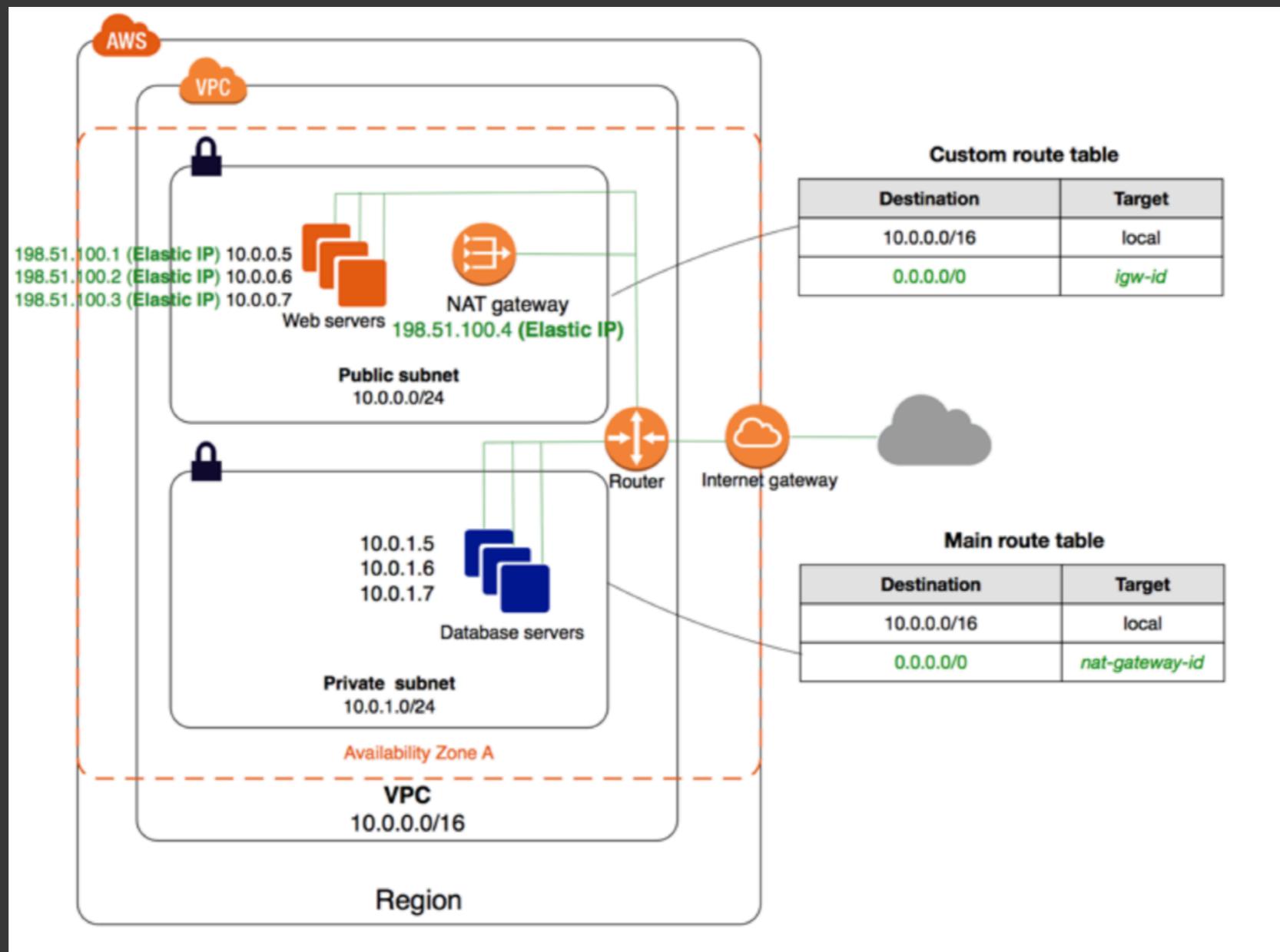
02

VPC 만들기



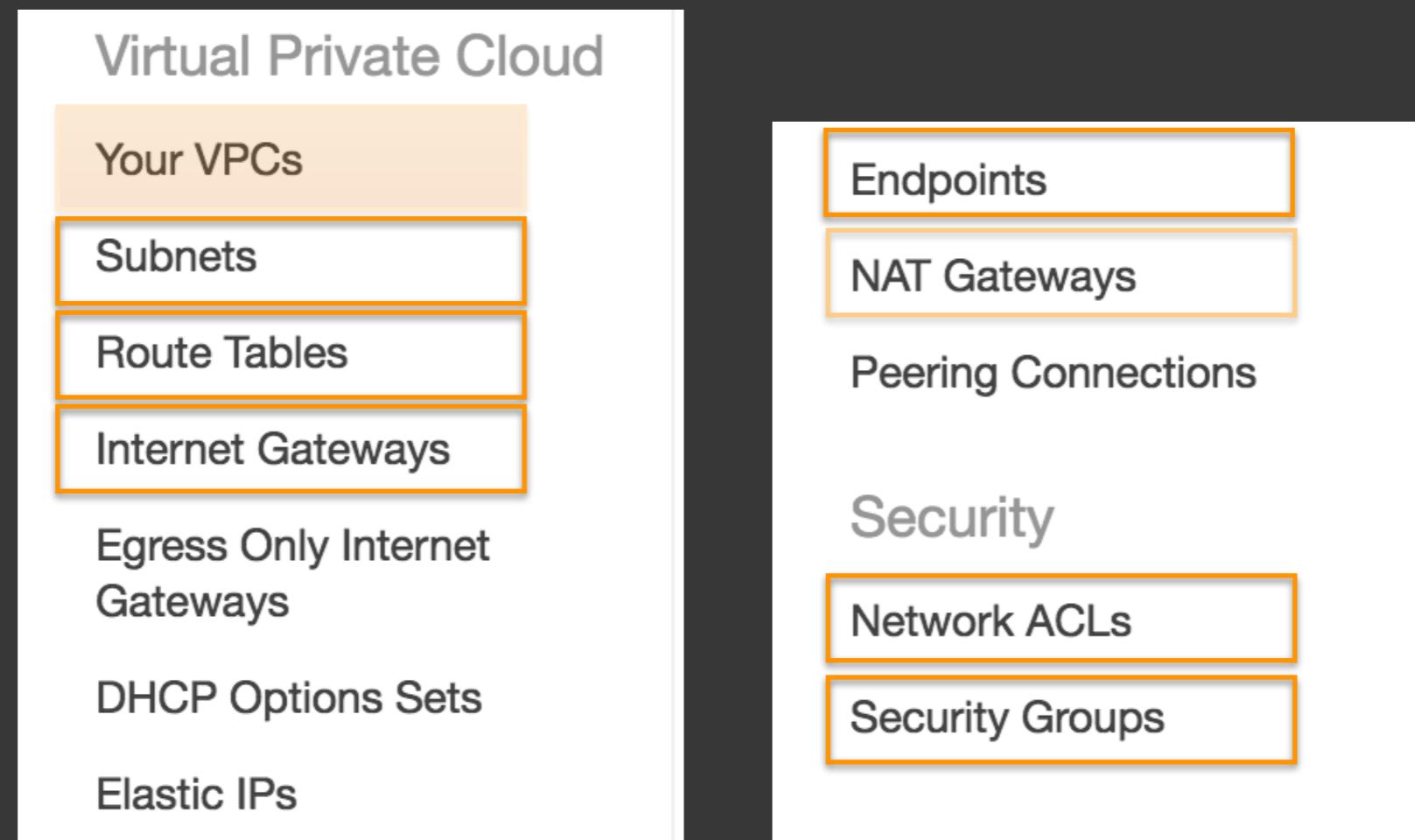
02

VPC 만들기



02

VPC 만들기



02

VPC 만들기

Create VPC ×

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an Amazon-provided IPv6 CIDR block with the VPC.

Name tag i

IPv4 CIDR block* i

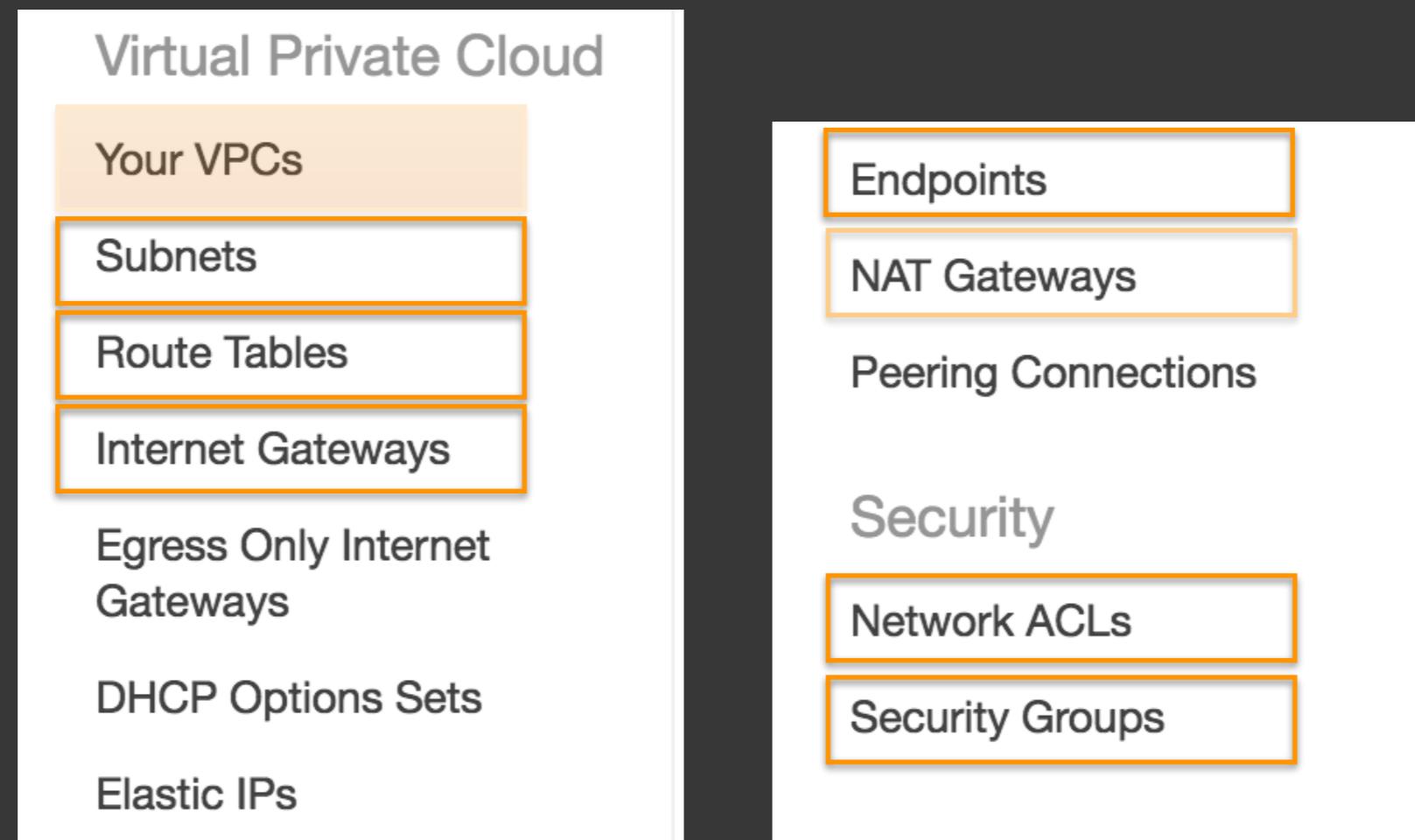
IPv6 CIDR block* No IPv6 CIDR Block i
 Amazon provided IPv6 CIDR block

Tenancy i

Cancel **Yes, Create**

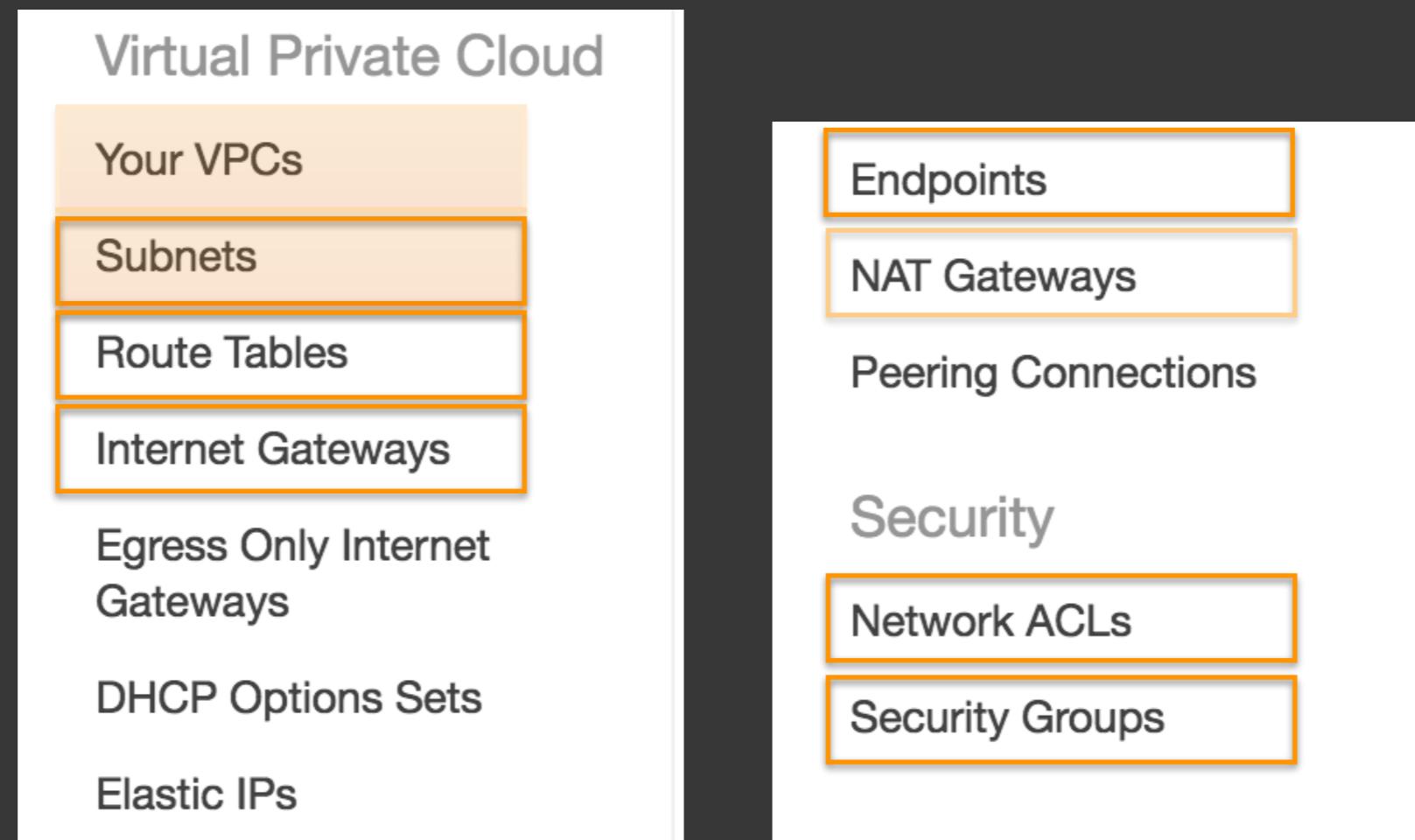
02

VPC 만들기



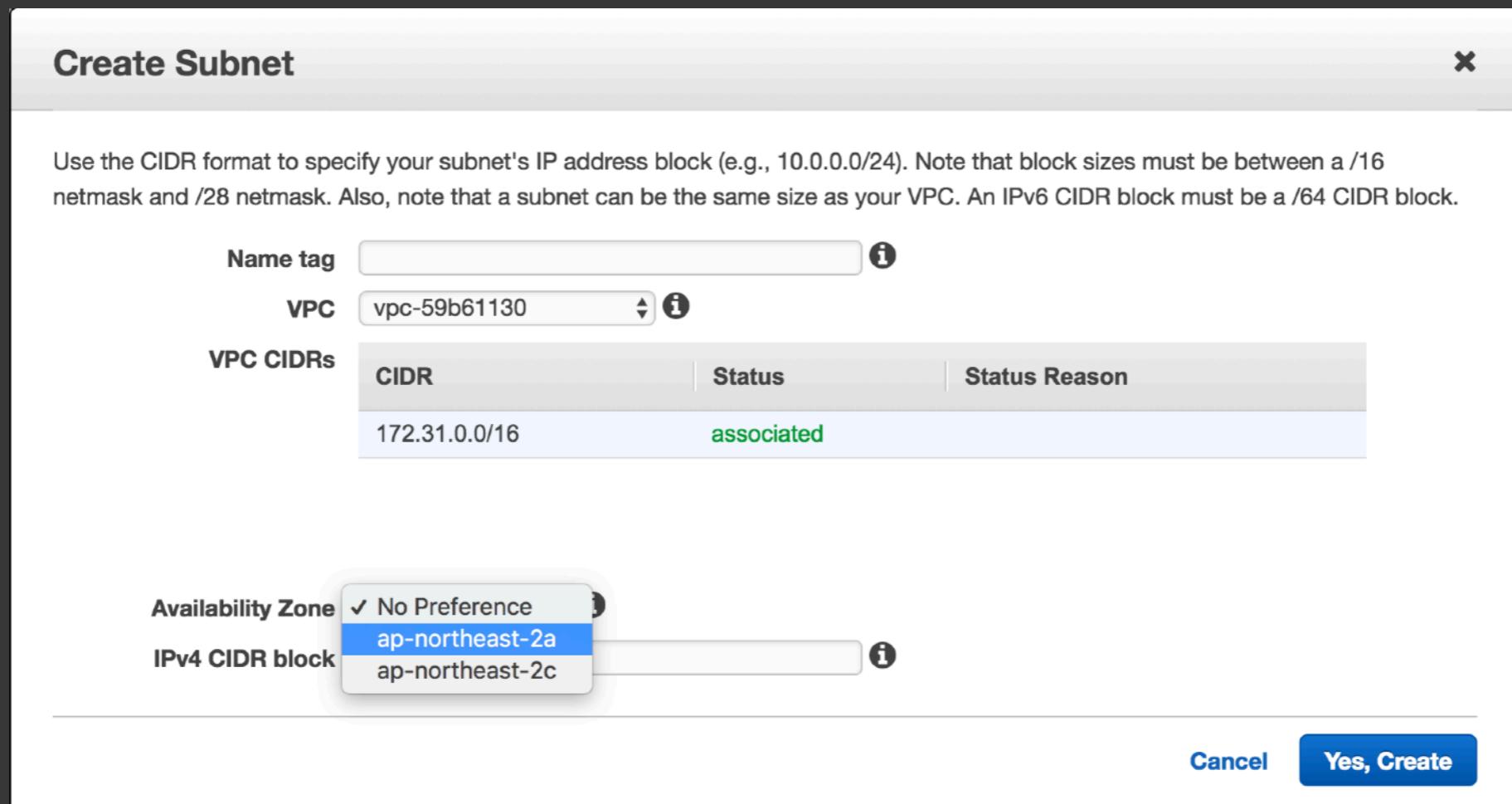
02

VPC 만들기



02

VPC 만들기



A와 B는 Availability Zone 다르게 설정해야!!

02

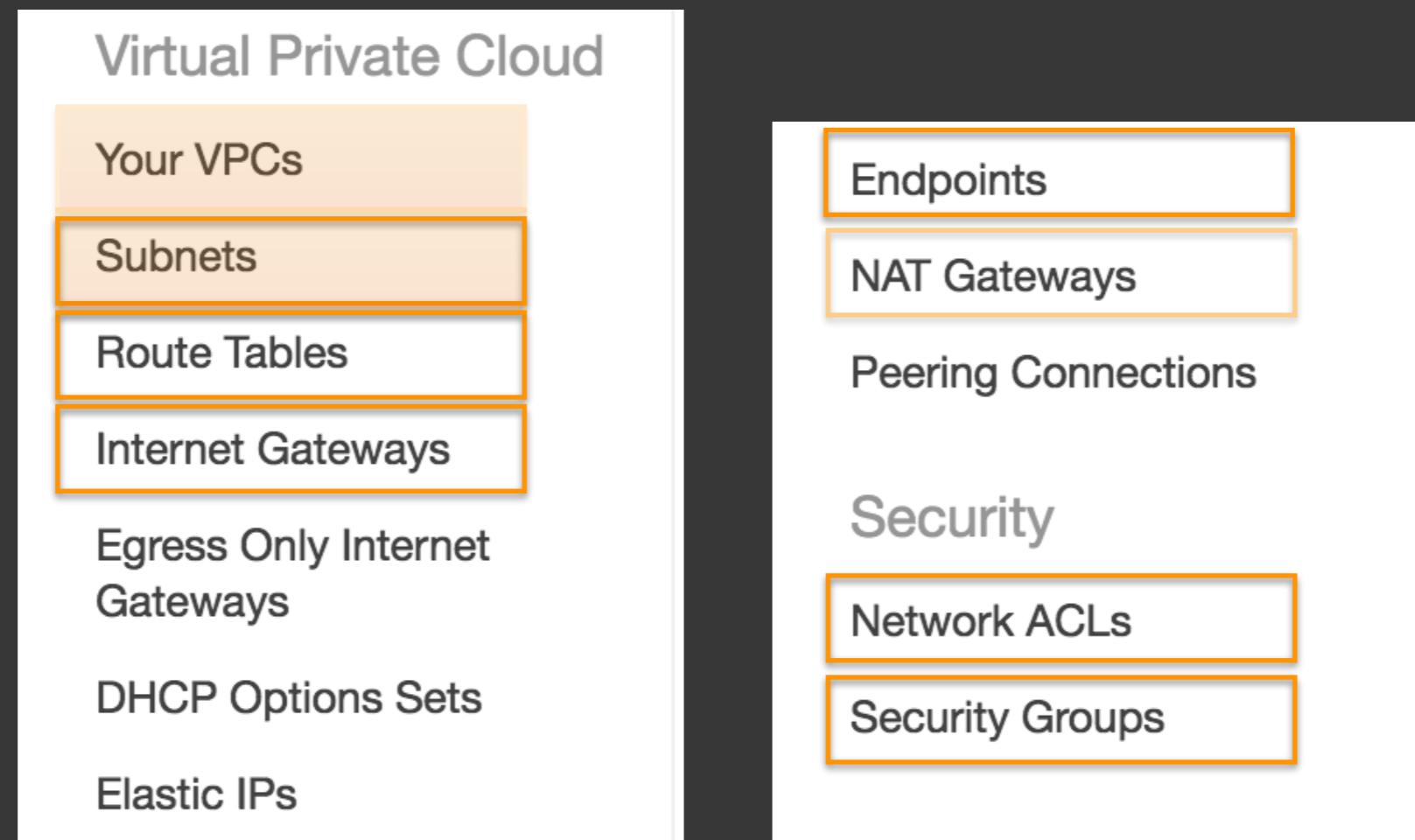
VPC 만들기

The screenshot shows a list of subnets in an AWS management console. The subnets are grouped by their associated VPC, which is highlighted with an orange box. The VPC column shows 'vpc-4adf7422 | lion_app' for all four subnets. The IPv4 CIDR columns also show different ranges: 10.0.1.0/24, 10.0.0.0/24, 10.0.101.0/24, and 10.0.100.0/24. The last two rows are for subnets without a VPC association.

	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
<input type="checkbox"/>	Public subnet B	subnet-fe8475b2	available	vpc-4adf7422 lion_app	10.0.1.0/24	251	
<input type="checkbox"/>	Public subnet A	subnet-ab27a1c3	available	vpc-4adf7422 lion_app	10.0.0.0/24	251	
<input type="checkbox"/>	Private Subnet B	subnet-fb8475b7	available	vpc-4adf7422 lion_app	10.0.101.0/24	251	
<input type="checkbox"/>	Private Subnet A	subnet-6727a10f	available	vpc-4adf7422 lion_app	10.0.100.0/24	251	
<input type="checkbox"/>		subnet-f34faebe	available	vpc-59b61130	172.31.16.0/20	4091	
<input type="checkbox"/>		subnet-9c60c1f5	available	vpc-59b61130	172.31.0.0/20	4089	

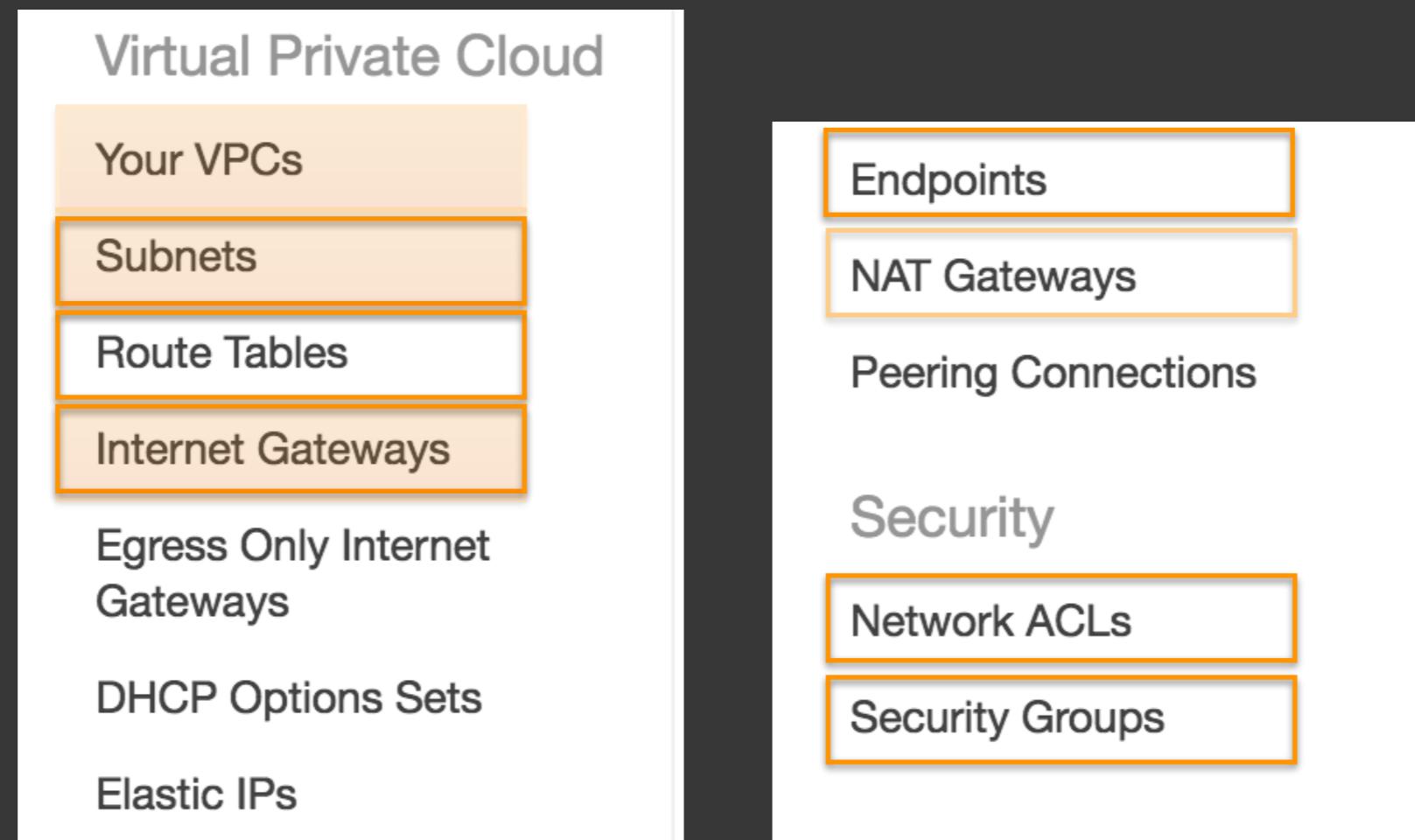
02

VPC 만들기



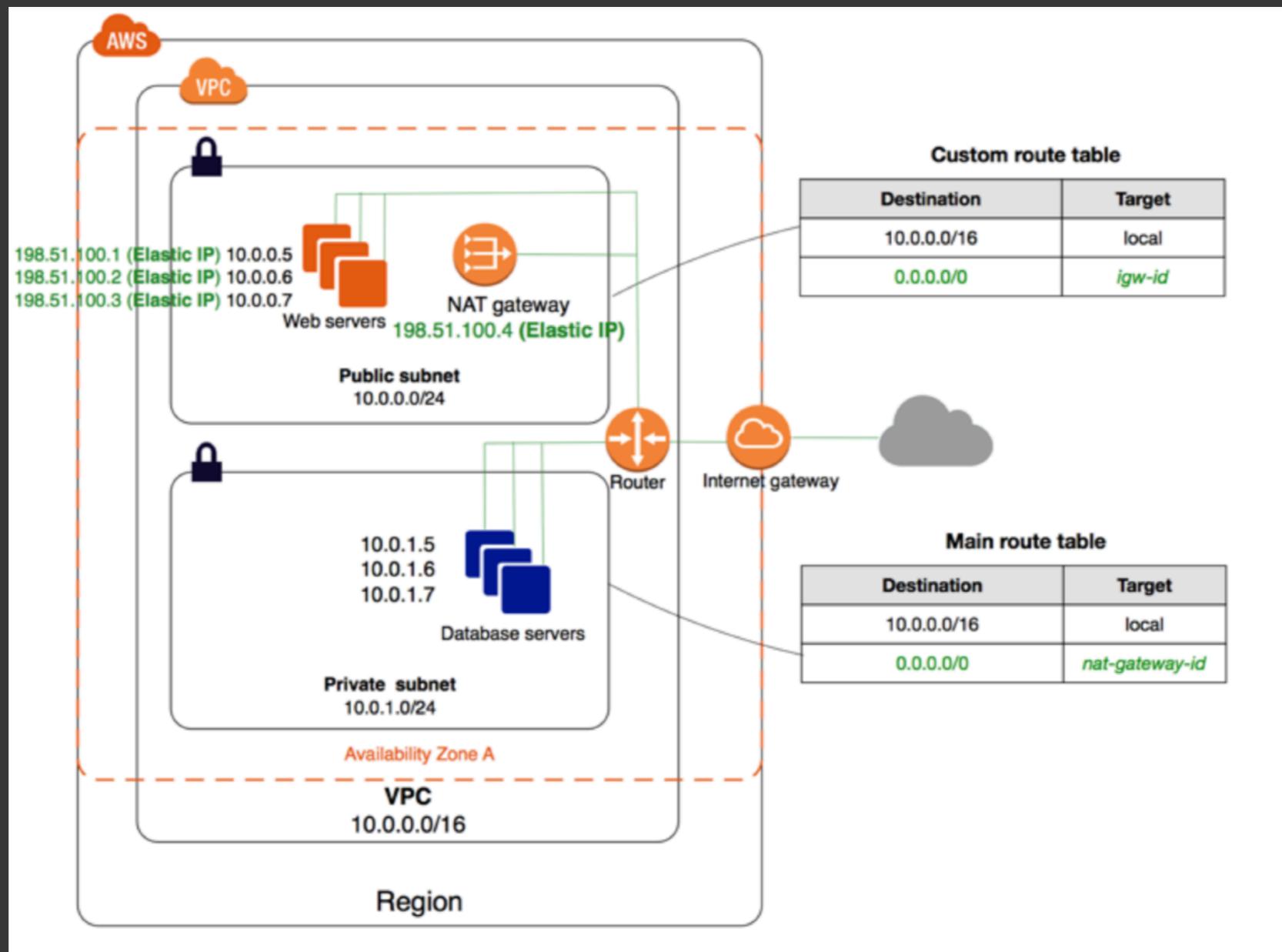
02

VPC 만들기



02

VPC 만들기



02

VPC 만들기

Internet Gateway

VPC의 인스턴스와 인터넷 간에 통신을 가능하게 함

- VPC에 internet gateway를 붙인다. (attach)
- subnet의 route table이 gateway를 가리키게 한다.
- VPC의 instances들이 unique IP address 를 갖도록 한다.
- NACL, Security group 이 traffic 보내게 설정한다.

02

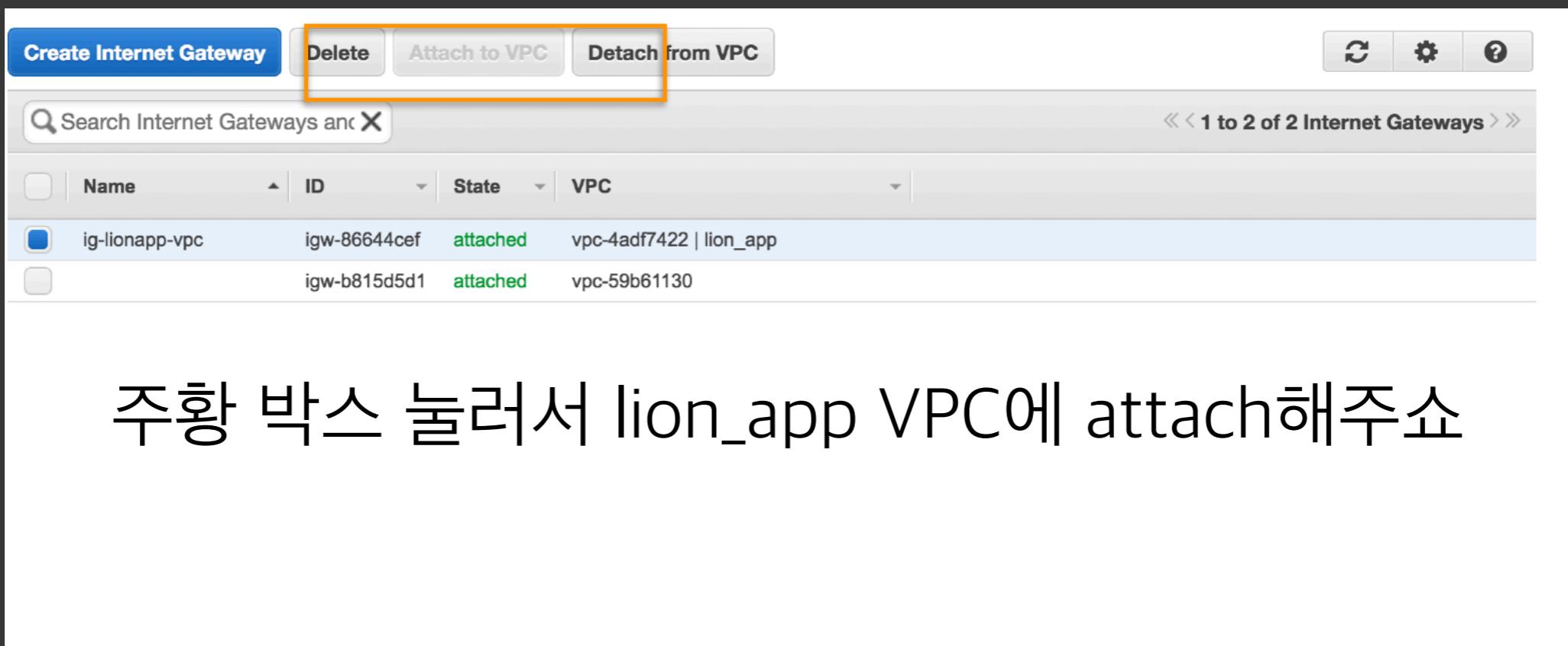
VPC 만들기

Internet Gateway

Public 애들이 얘를 통해서 인터넷과 통신한다고 배웠다.

02

VPC 만들기



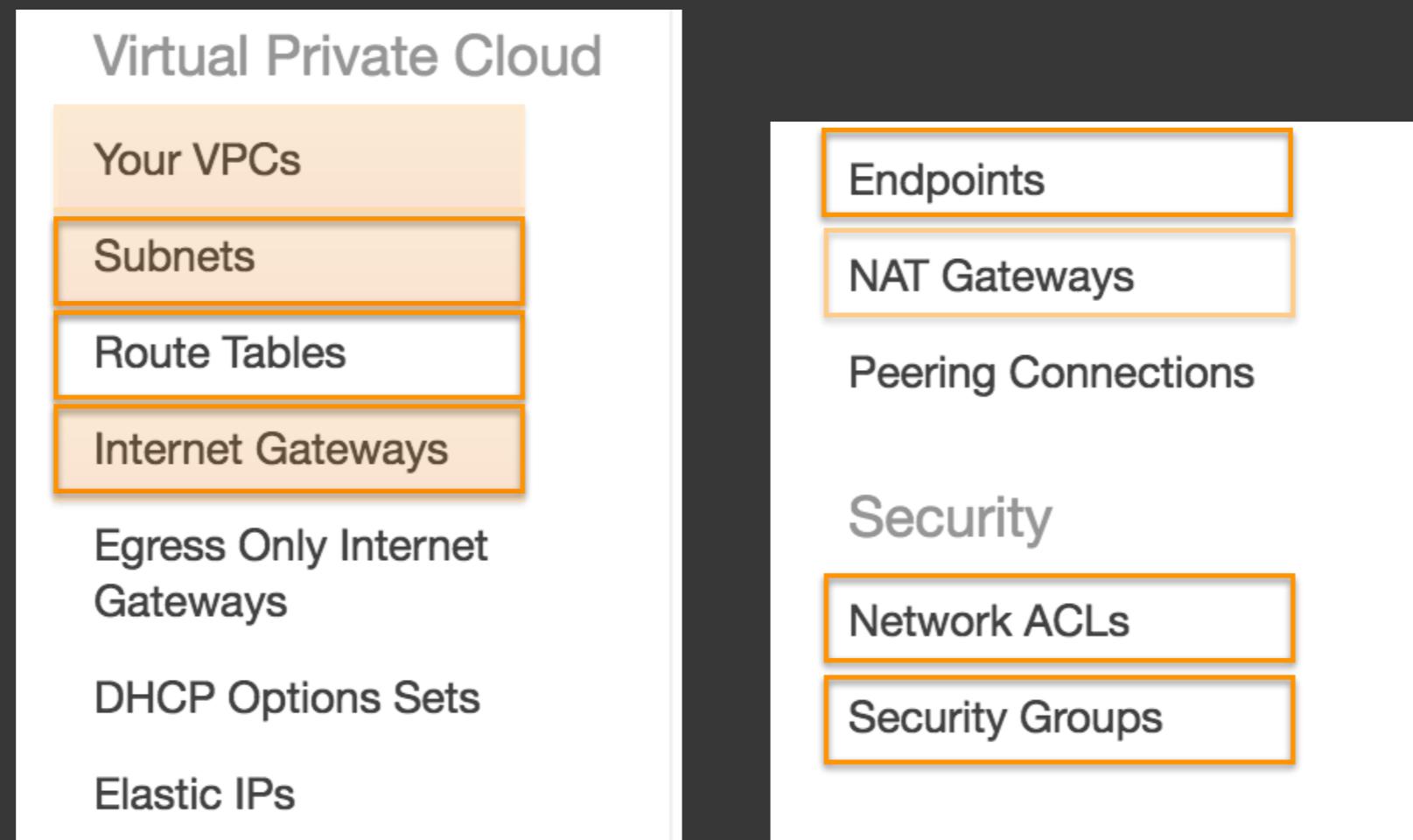
The screenshot shows the AWS Management Console interface for managing Internet Gateways. At the top, there are four buttons: 'Create Internet Gateway' (blue), 'Delete' (disabled), 'Attach to VPC' (disabled), and 'Detach from VPC' (highlighted with a yellow box). Below the buttons is a search bar with placeholder text 'Search Internet Gateways and X'. To the right of the search bar are navigation links '« « 1 to 2 of 2 Internet Gateways » »'. The main area displays a table with two rows of data:

	Name	ID	State	VPC
<input type="checkbox"/>	ig-lionapp-vpc	igw-86644cef	attached	vpc-4adf7422 lion_app
<input type="checkbox"/>		igw-b815d5d1	attached	vpc-59b61130

주황 박스 눌러서 lion_app VPC에 attach해주쇼

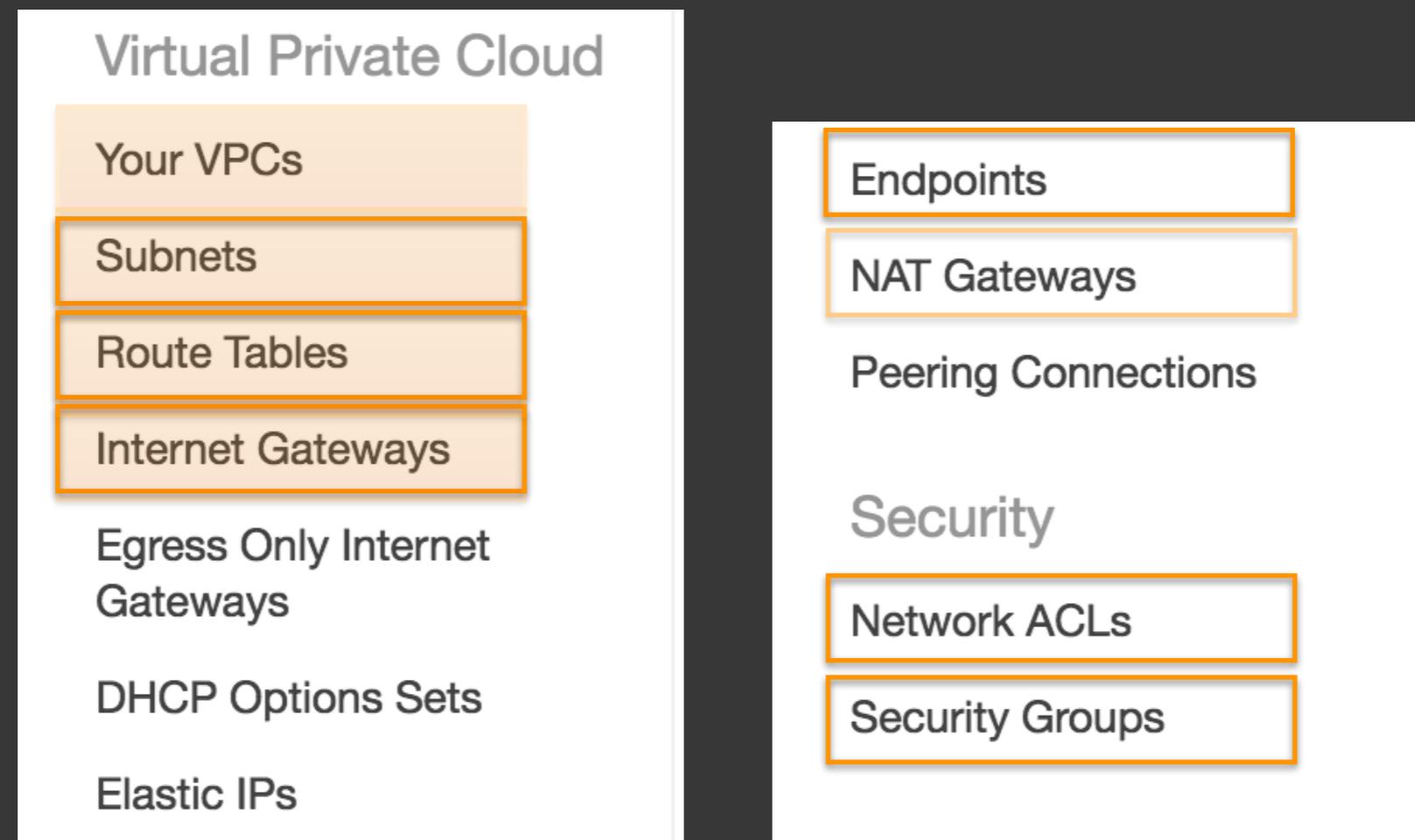
02

VPC 만들기



02

VPC 만들기



02

VPC 만들기

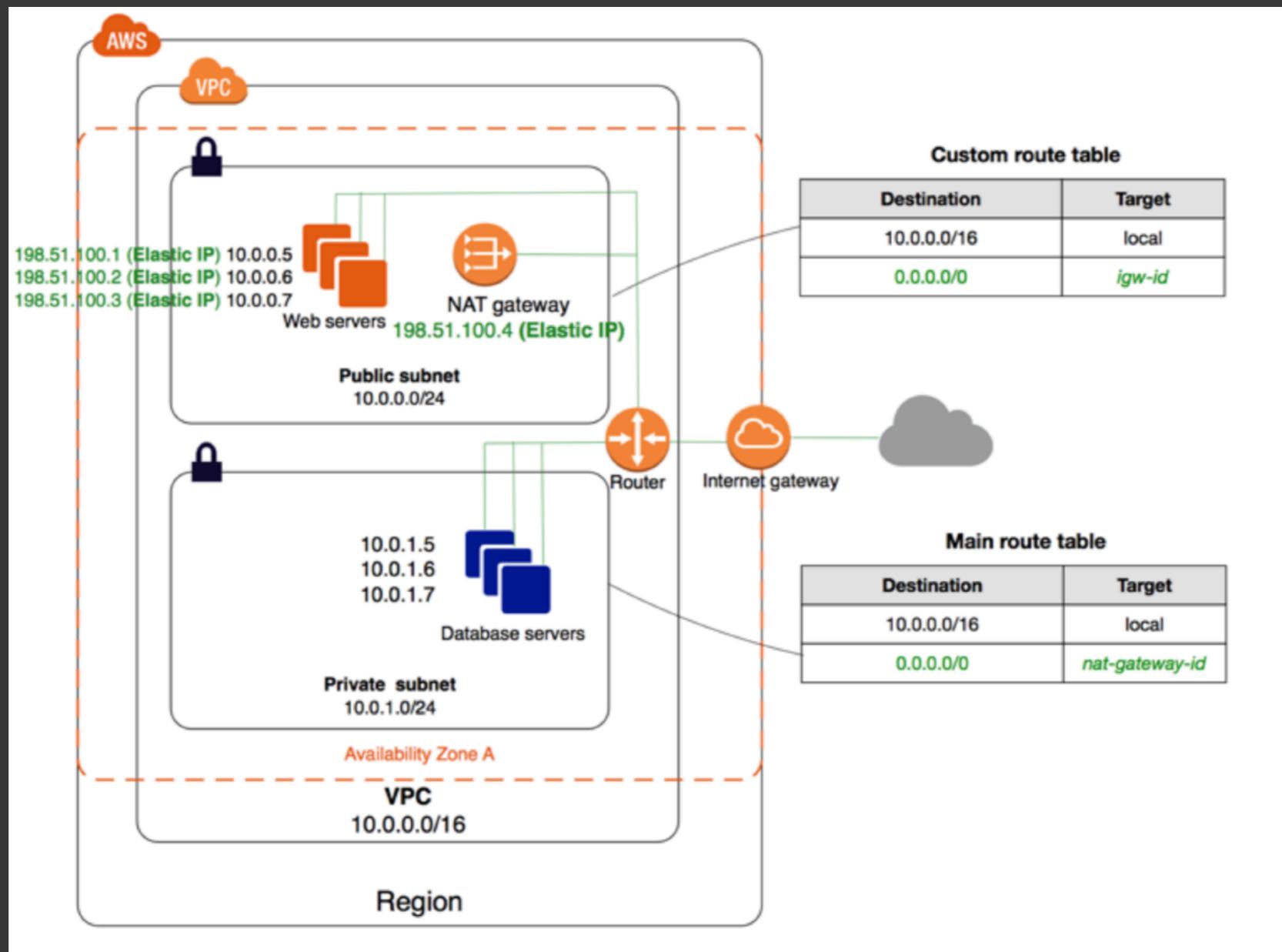
Route Table

네트워크 트래픽을 전달할 위치를 결정하는 데 사용되는 규칙 집합이 포함

- 각 subnet은 라우팅테이블에 연결되어있어야 함
- 테이블은 subnet에 대한 routing을 제어
- subnet과 라우팅테이블은 n:1 or 1:1

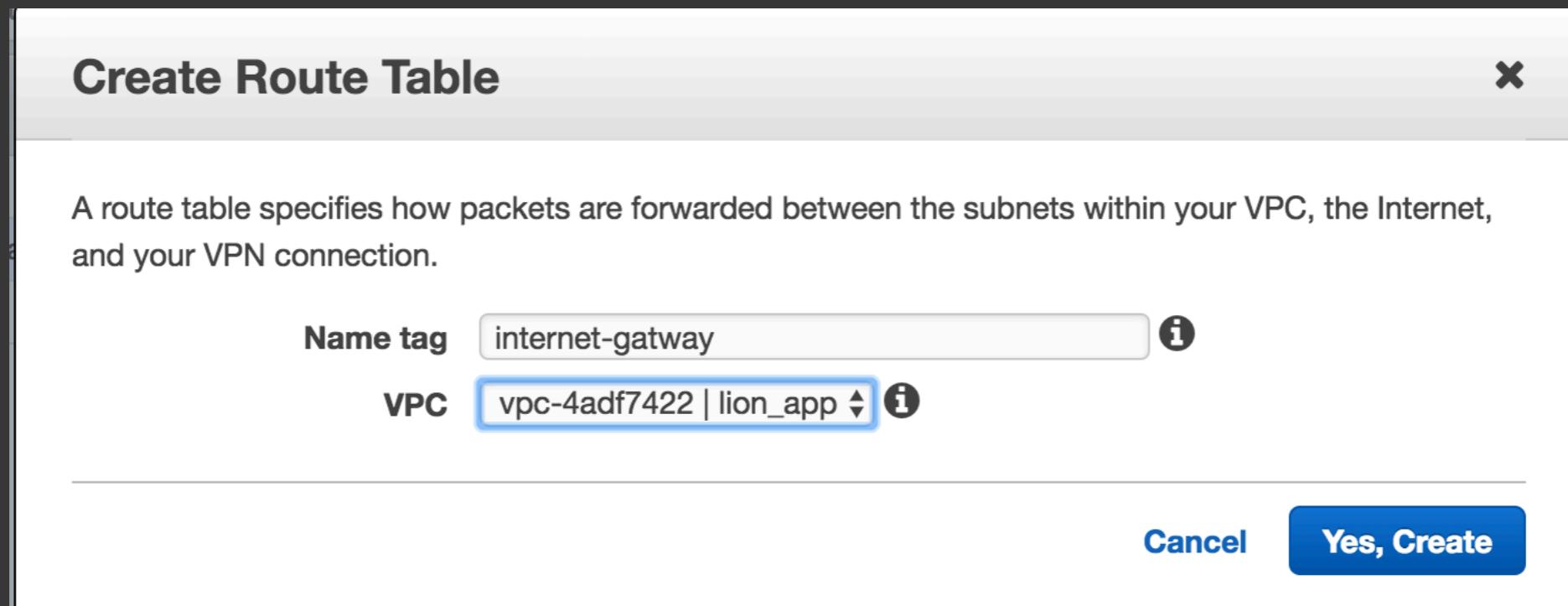
02

VPC 만들기



02

VPC 만들기



Subnet Associations				
Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Route Table
<input checked="" type="checkbox"/>	subnet-ab27a1c3 Public subnet A	10.0.0.0/24	-	Main
<input checked="" type="checkbox"/>	subnet-fe8475b2 Public subnet B	10.0.1.0/24	-	Main

02

VPC 만들기

The screenshot shows the AWS Route Tables management interface. At the top, there are buttons for 'Create Route Table', 'Delete Route Table', and 'Set As Main Table'. Below is a search bar and a navigation bar indicating '1 to 3 of 3 Route Tables'. The main table lists three route tables:

Name	Route Table ID	Explicitly Associated	Main	VPC
	rtb-28e67c40	0 Subnets	Yes	vpc-4adf7422 lion_app
internet-gateway	rtb-99e47ef1	2 Subnets	No	vpc-4adf7422 lion_app
	rtb-03ca676a	0 Subnets	Yes	vpc-59b61130

Below this, a modal window titled 'Route 설정' (Route Configuration) is displayed. It shows the configuration for the 'internet-gateway' route table. The table has two rules:

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-86644cef	No		X

Buttons for 'Cancel' and 'Save' are at the bottom left of the modal.

우리의 internet gateway를 통해서 인터넷 접근하게
Route 설정

02

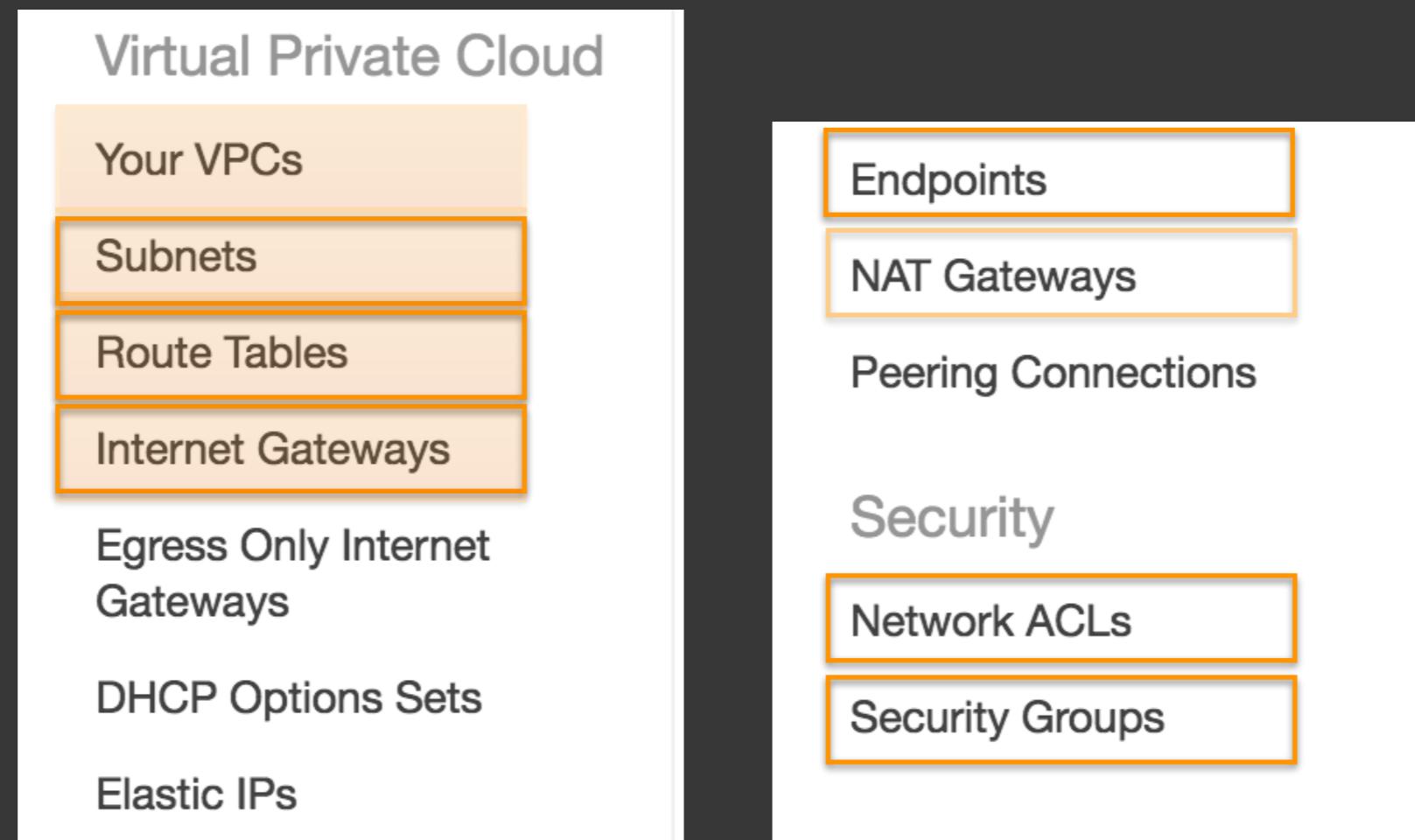
VPC 만들기

Route Table

Public subnet 애들이 internet gateway를 통해서
인터넷과 통신할 수 있도록
트래픽을 전달하는 규칙을 만든 것임

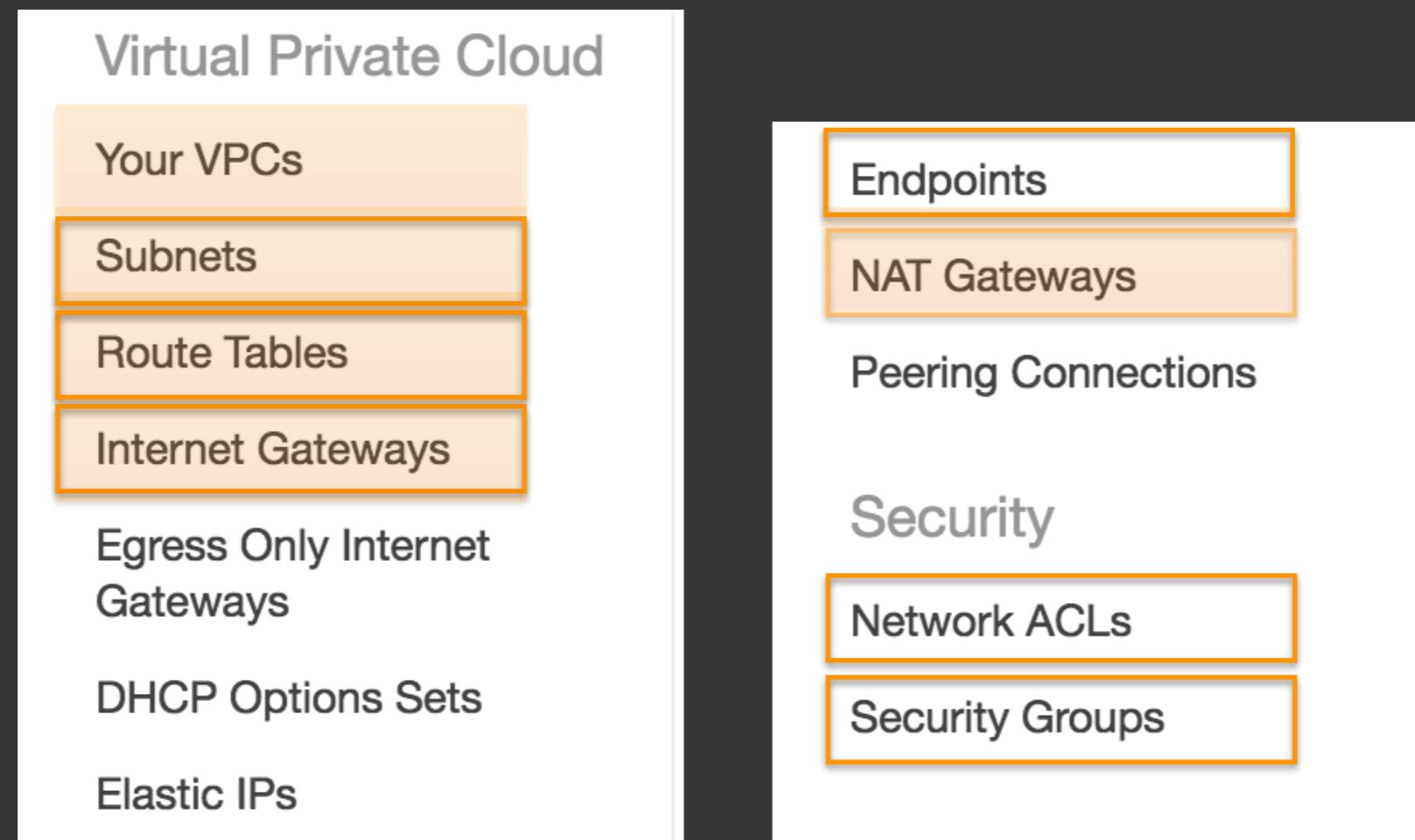
02

VPC 만들기



02

VPC 만들기



02

VPC 만들기

NAT gateway

Private Subnet이 인터넷에 접근할 때는
NAT gateway 통해서!

02

VPC 만들기

NAT Gateways > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

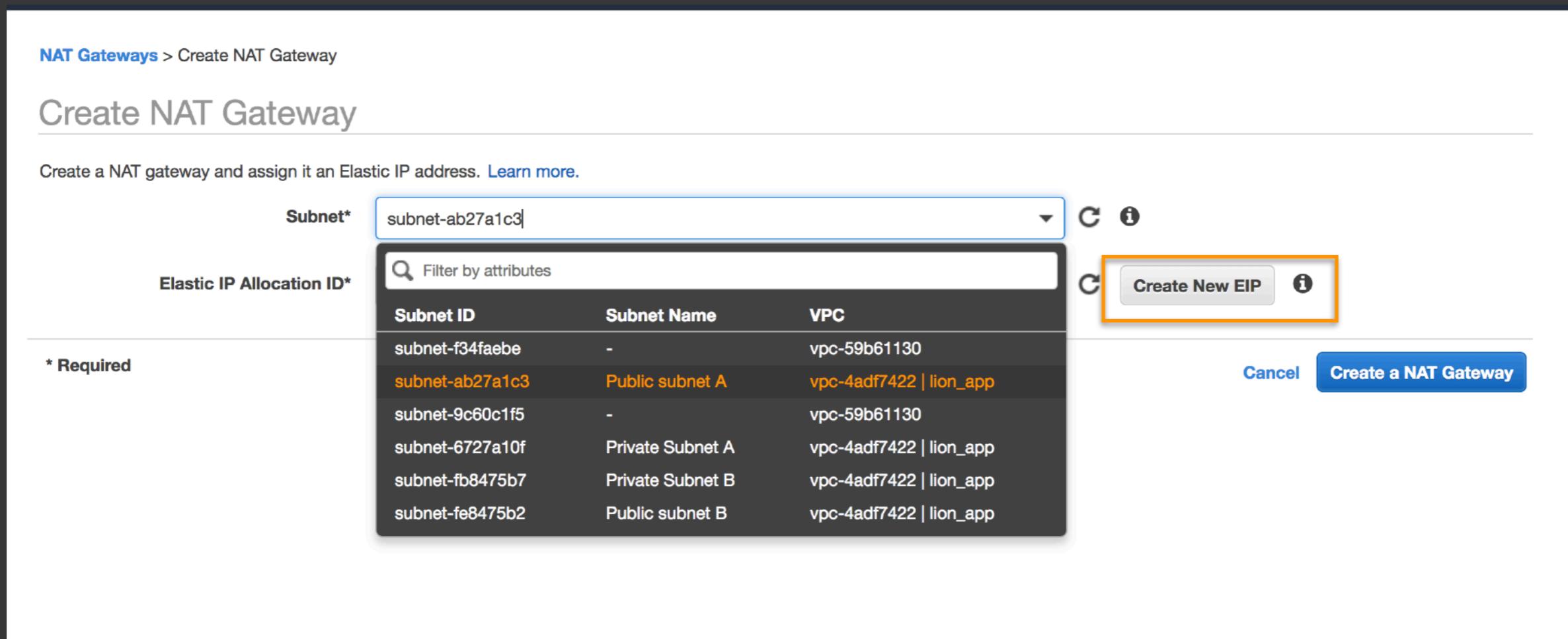
Subnet* C i

Elastic IP Allocation ID* C i Filter by attributes Create New EIP i

* Required

Subnet ID	Subnet Name	VPC
subnet-f34faebe	-	vpc-59b61130
subnet-ab27a1c3	Public subnet A	vpc-4adf7422 lion_app
subnet-9c60c1f5	-	vpc-59b61130
subnet-6727a10f	Private Subnet A	vpc-4adf7422 lion_app
subnet-fb8475b7	Private Subnet B	vpc-4adf7422 lion_app
subnet-fe8475b2	Public subnet B	vpc-4adf7422 lion_app

[Cancel](#) [Create a NAT Gateway](#)



02

VPC 만들기

The screenshot shows the AWS Management Console interface for managing NAT Gateways. At the top, there is a blue button labeled "Create NAT Gateway". Below it is a search bar with the placeholder text "Filter by tags and attributes or search by keyword". To the right of the search bar are navigation icons: back, forward, and a "1 to 1 of 1" indicator. The main area is a table with the following columns: Name, NAT Gateway ID, Status, Status Message, Elastic IP Address, Private IP Address, and Network Interface. A single row is displayed, representing a NAT gateway named "nat-049c7c81a1ee...", which is currently "available". Its elastic IP address is 52.79.55.230, private IP address is 10.0.0.194, and it is associated with the network interface eni-bd21f3e2.

Name	NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address	Network Interface
nat-049c7c81a1ee...	available	-		52.79.55.230	10.0.0.194	eni-bd21f3e2

02

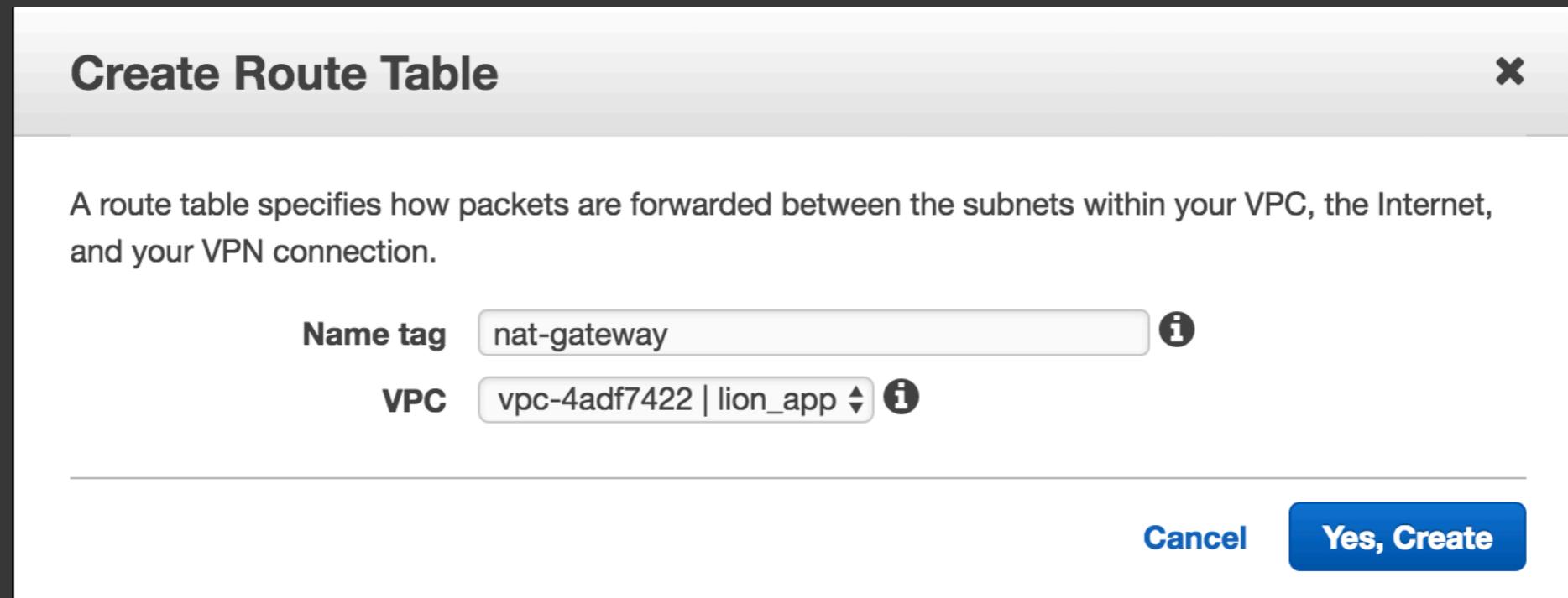
VPC 만들기

Route Table

Private subnet 애들이 nat gateway를 통해서
인터넷과 통신할 수 있도록
트래픽을 전달하는 규칙을 만들자

02

VPC 만들기



nat-gateway를 다시 만든다.

02

VPC 만들기

Summary Routes Subnet Associations Route Propagation Tags

Cancel **Save**

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	nat-05e265605394a914	No		X

Summary Routes **Subnet Associations** Route Propagation Tags

Edit

Subnet	IPv4 CIDR	IPv6 CIDR
subnet-6727a10f Private Subnet A	10.0.100.0/24	-
subnet-fb8475b7 Private Subnet B	10.0.101.0/24	-

The following subnets have not been explicitly associated with

02

VPC 만들기

[Create Route Table](#) [Delete Route Table](#) [Set As Main Table](#)

Search Route Tables and their X « « 1 to 4 of 4 Route Tables » »

Name	Route Table ID	Explicitly Associated	Main	VPC
nat-gateway	rtb-869a00ee	2 Subnets	No	vpc-4adf7422 lion_app
	rtb-28e67c40	0 Subnets	Yes	vpc-4adf7422 lion_app
internet-gateway	rtb-99e47ef1	2 Subnets	No	vpc-4adf7422 lion_app
	rtb-03ca676a	0 Subnets	Yes	vpc-59b61130

[Edit](#)

View: All rules

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	nat-05e265605394a9142	Active	No

02

VPC 만들기

Public Subnet

Internet gateway

EC2

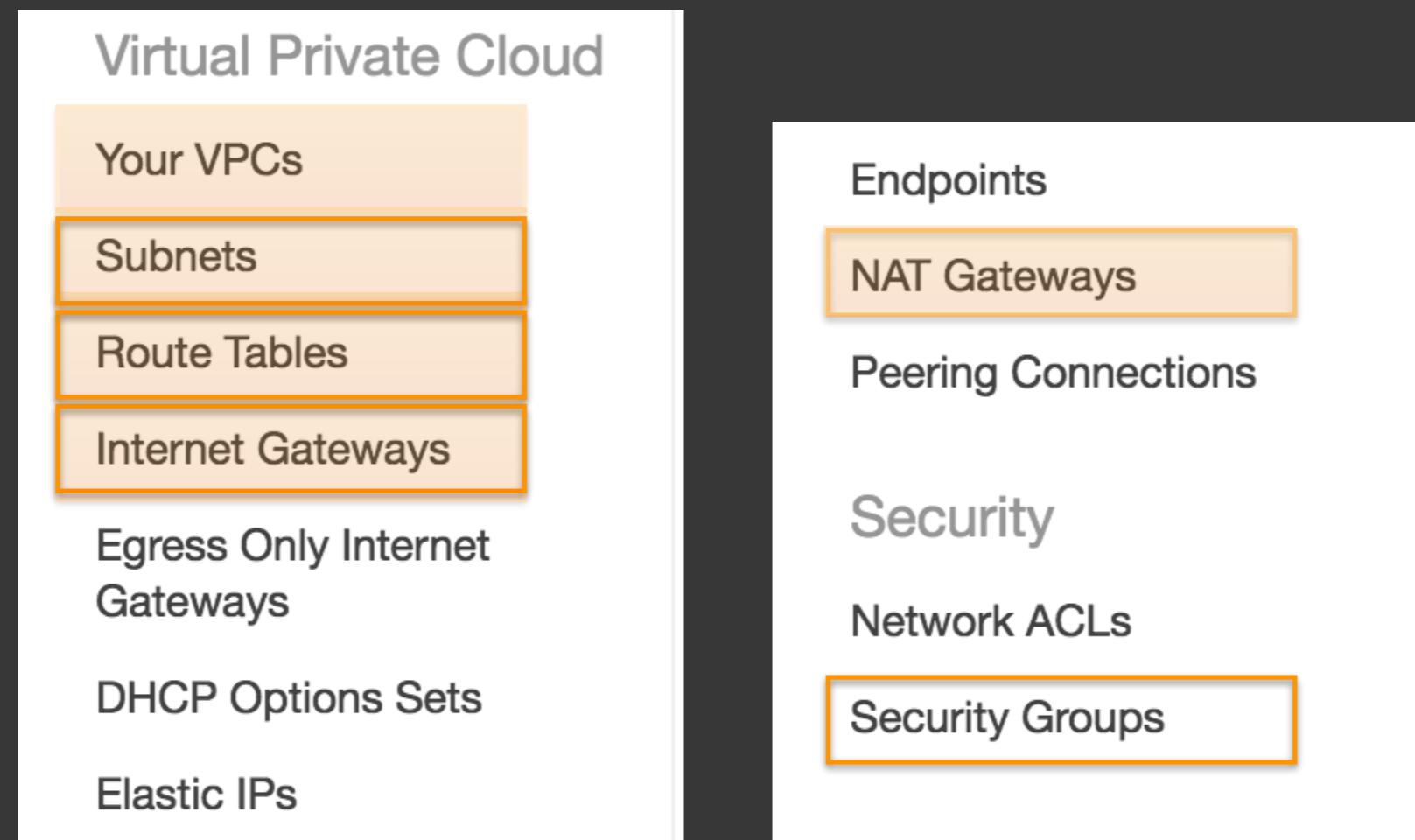
Private Subnet

NAT gateway

RDS

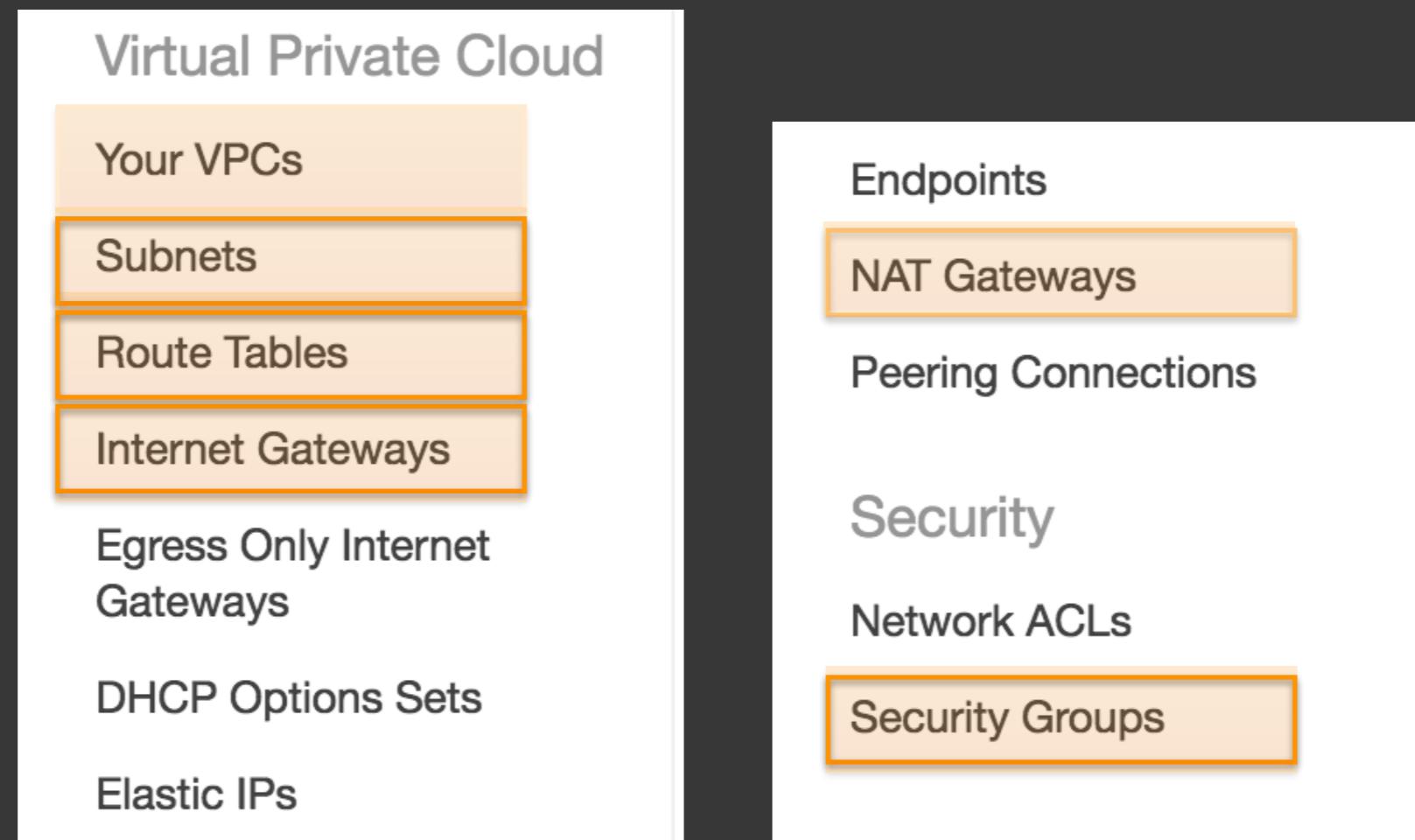
02

VPC 만들기



02

VPC 만들기



02

VPC 만들기

Security Group

서로 다른 instance 접근을 위한 security group 만들기

- 관리자
- Web Server 그룹
- DB Server 그룹

02

VPC 만들기

DEYim

가장 밑단의 관리자 / RDS에 접근할 수 있는 컴퓨터

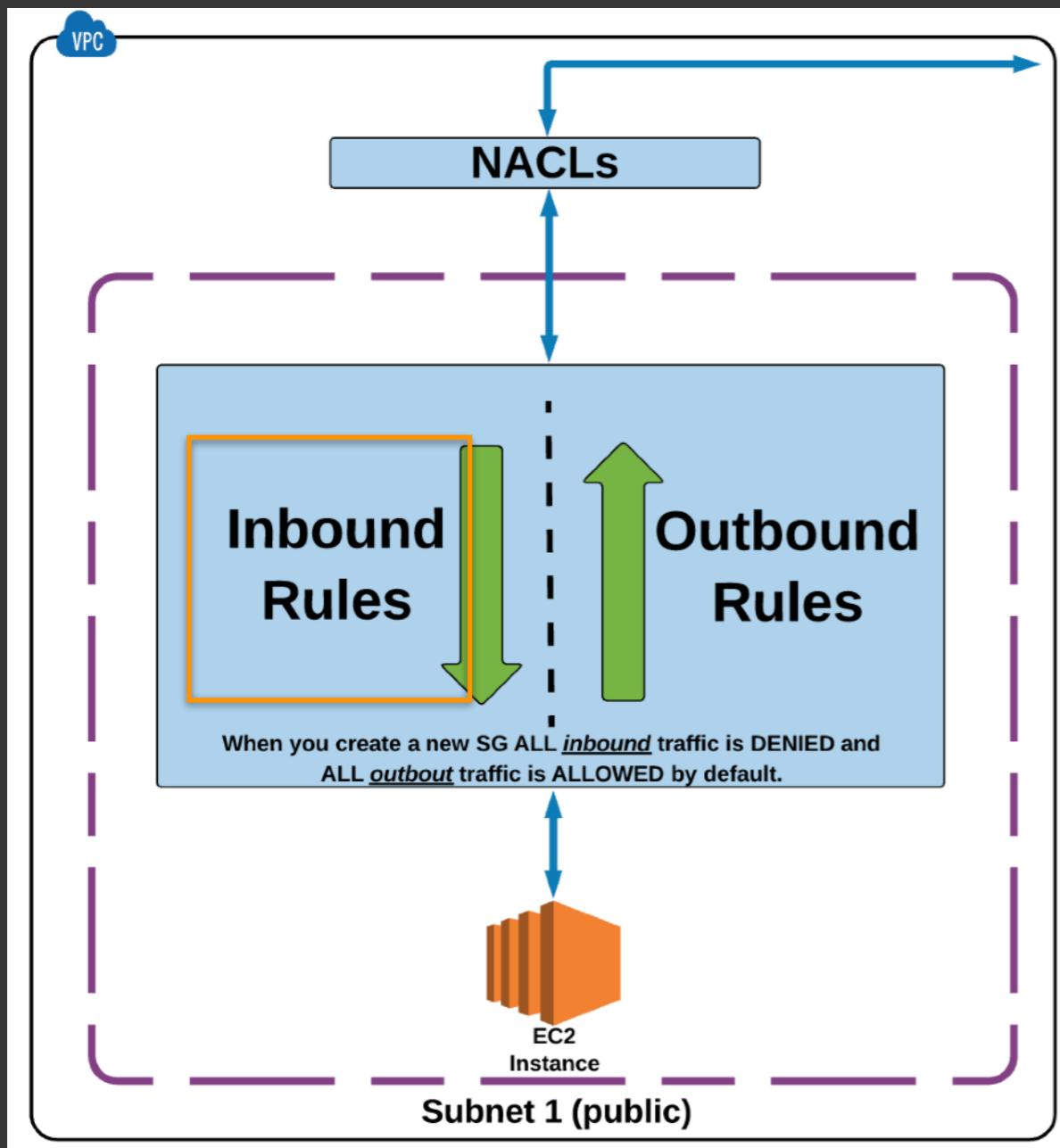
DEYim DB Server

RDS에 접근 가능한 그룹

DEYim Web Server

Web Server EC2에 접근 가능한 그룹

Amazon - EC2 security group



모든 트래픽은 명시적인 허락
이 있어야 내 컴퓨터로 접근할 수 있다.

거절은 없다. 허가만 있다.

02

VPC 만들기

DEYim

내 IP 주소만 SSH로 접근 (까만 화면)

DEYim DB Server

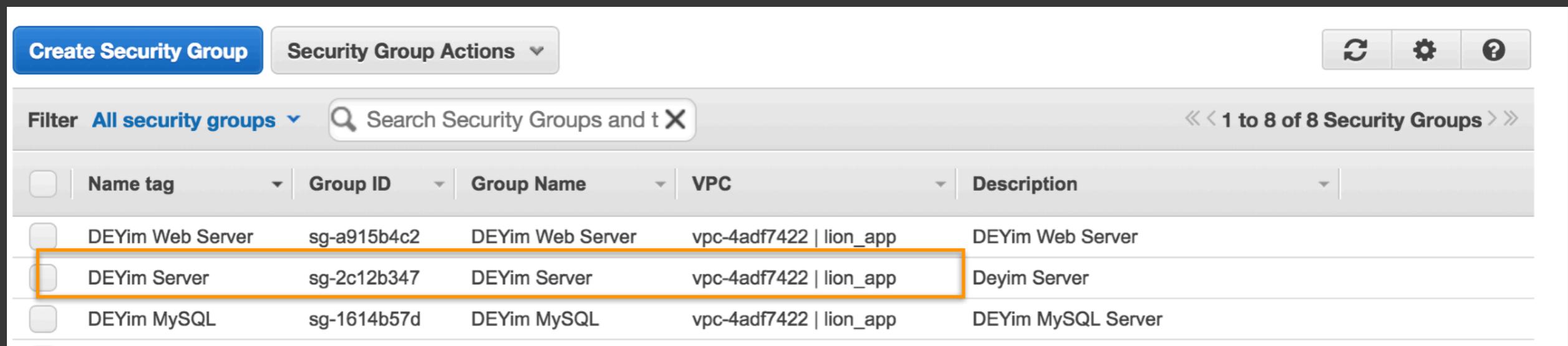
MySQL/Aurora 에만 접근할 수 있으며
Deyim, Deyim Web Server 만 접근 가능

DEYim Web Server

HTTP와 SSH 모두 누구나 접근할 수 있음
(SSH는 추후에 보안 높게 변경해도 상관 없음)

02

VPC 만들기

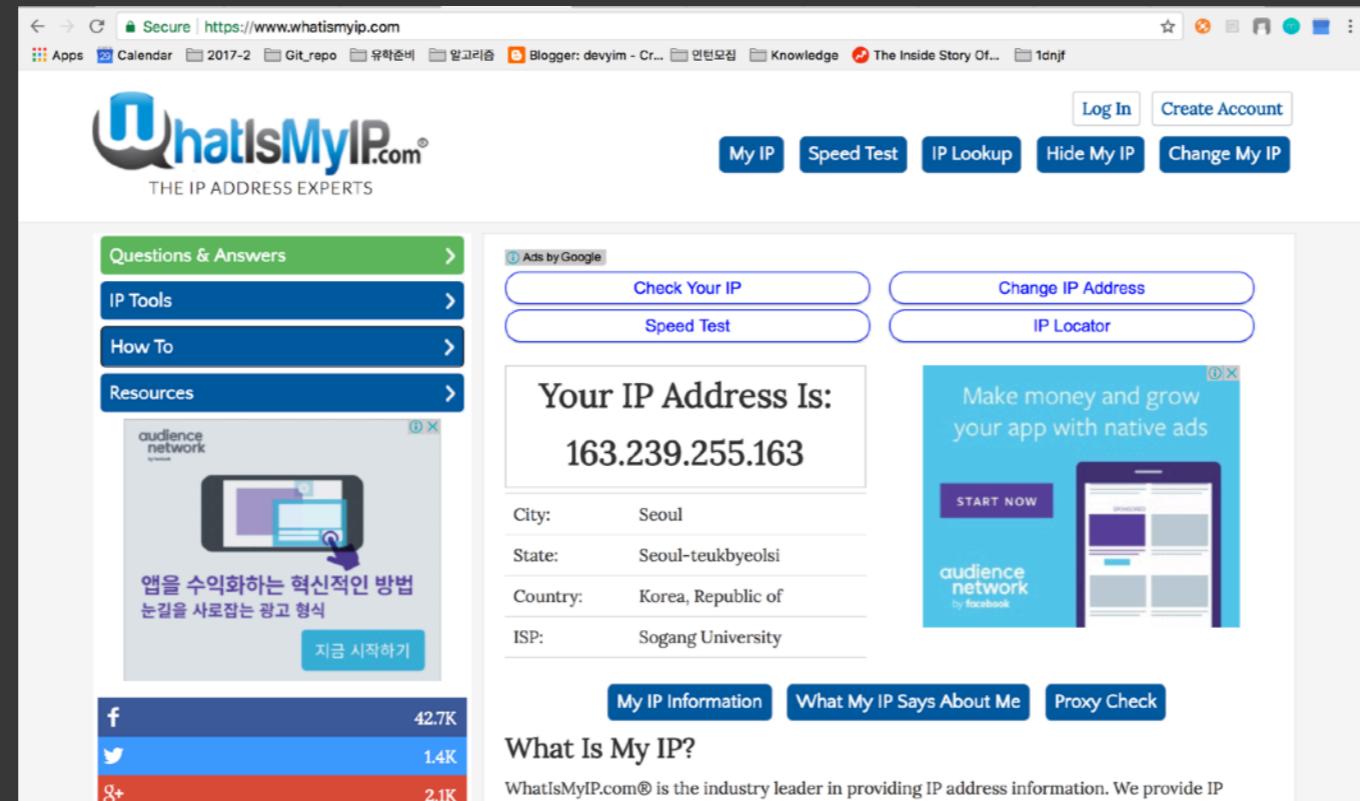


The screenshot shows the AWS Security Groups management interface. At the top, there are buttons for 'Create Security Group' and 'Security Group Actions'. Below that is a search bar with the placeholder 'Search Security Groups and t X'. To the right of the search bar are navigation icons for refresh, settings, and help. The main area displays a table of security groups with the following data:

Name tag	Group ID	Group Name	VPC	Description
DEYim Web Server	sg-a915b4c2	DEYim Web Server	vpc-4adf7422 lion_app	DEYim Web Server
DEYim Server	sg-2c12b347	DEYim Server	vpc-4adf7422 lion_app	Deyim Server
DEYim MySQL	sg-1614b57d	DEYim MySQL	vpc-4adf7422 lion_app	DEYim MySQL Server

02

VPC 만들기



what is my ip 치고 들어가서
내 IP확인하기

02

VPC 만들기

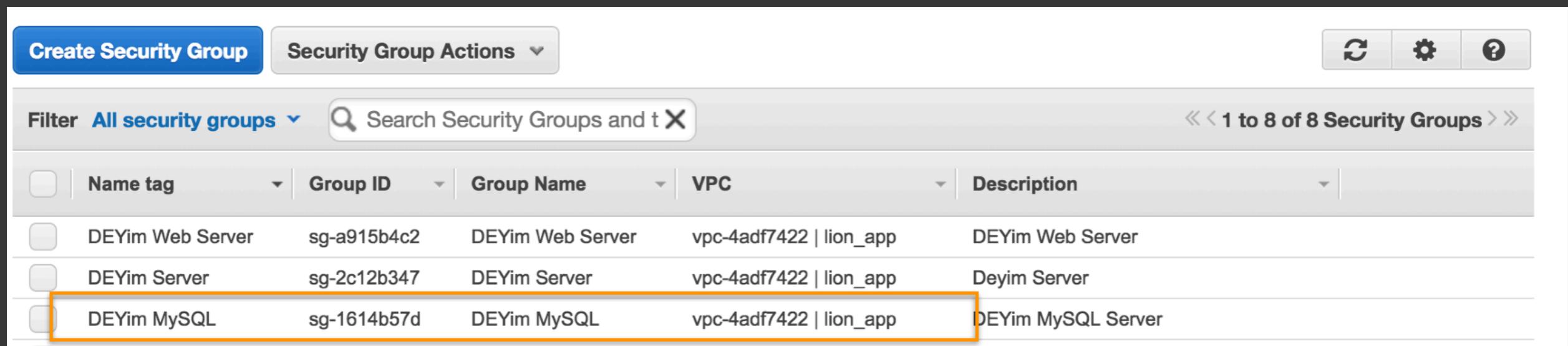
The screenshot shows the 'Inbound Rules' tab selected in a VPC configuration interface. The top navigation bar includes 'Summary', 'Inbound Rules' (selected), 'Outbound Rules', and 'Tags'. Below the navigation is a toolbar with 'Cancel' and 'Save' buttons. The main area displays a table for inbound rules:

Type	Protocol	Port Range	Source	Description	Remove
SSH (22)	TCP (6)	22	163.239.255.163/32	(info icon)	(remove icon)

At the bottom left is a button labeled 'Add another rule'.

02

VPC 만들기



The screenshot shows the AWS Security Groups management interface. At the top, there are buttons for 'Create Security Group' and 'Security Group Actions'. Below that is a search bar with the placeholder 'Search Security Groups and t X'. To the right of the search bar are navigation icons for refresh, settings, and help. The main area displays a table of security groups with the following data:

Name tag	Group ID	Group Name	VPC	Description
DEYim Web Server	sg-a915b4c2	DEYim Web Server	vpc-4adf7422 lion_app	DEYim Web Server
DEYim Server	sg-2c12b347	DEYim Server	vpc-4adf7422 lion_app	Deyim Server
DEYim MySQL	sg-1614b57d	DEYim MySQL	vpc-4adf7422 lion_app	DEYim MySQL Server

02

VPC 만들기

The screenshot shows the AWS VPC Inbound Rules configuration interface. It displays two nearly identical rule entries, each consisting of three fields: Type, Protocol, and Port Range.

Rule 1:

- Type: MySQL/Aurora (3306)
- Protocol: TCP (6)
- Port Range: 3306

Rule 2:

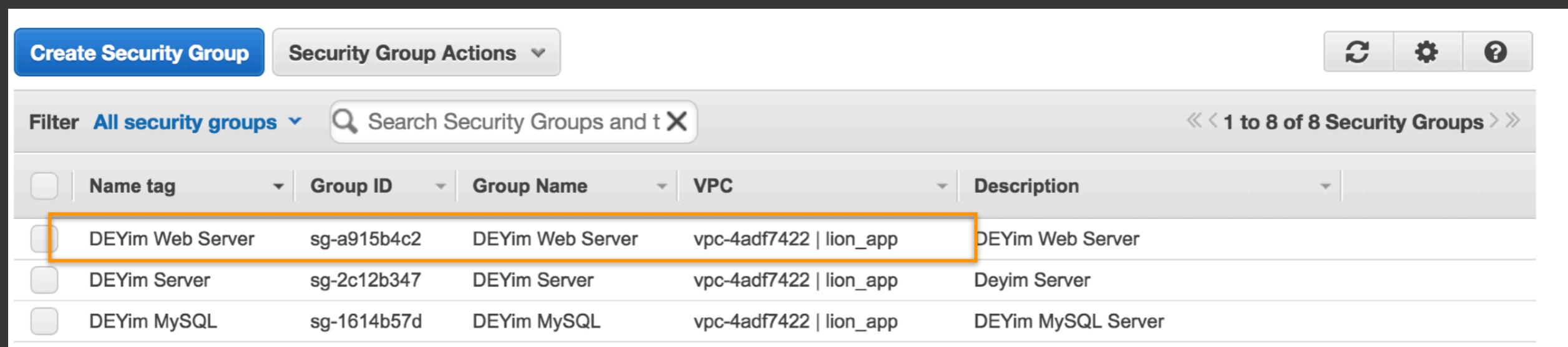
- Type: MySQL/Aurora (3306)
- Protocol: TCP (6)
- Port Range: 3306

A dropdown menu is open over the second rule, listing several security group IDs and their descriptions. The selected item is "sg-2c12b347 | DEYim Server". Other items in the list include:

- sg-1614b57d | DEYim MySQL
- sg-2c0dac47
- sg-2c12b347 | DEYim Server
- sg-a915b4c2 | DEYim Web Server
- pl-48a54021 | com.amazonaws.ap-northeast-2.dynamodb
- pl-78a54011 | com.amazonaws.ap-northeast-2.s3

02

VPC 만들기



<input type="checkbox"/>	Name tag	Group ID	Group Name	VPC	Description
<input checked="" type="checkbox"/>	DEYim Web Server	sg-a915b4c2	DEYim Web Server	vpc-4adf7422 lion_app	DEYim Web Server
<input type="checkbox"/>	DEYim Server	sg-2c12b347	DEYim Server	vpc-4adf7422 lion_app	Deyim Server
<input type="checkbox"/>	DEYim MySQL	sg-1614b57d	DEYim MySQL	vpc-4adf7422 lion_app	DEYim MySQL Server

02

VPC 만들기

The screenshot shows the 'Inbound Rules' tab selected in a VPC configuration interface. A blue 'Edit' button is visible. Below it is a table with columns: Type, Protocol, Port Range, Source, and Description. Two rules are listed:

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	0.0.0.0/0	
SSH (22)	TCP (6)	22	0.0.0.0/0	

02

VPC 만들기

The screenshot shows two side-by-side AWS VPC Inbound Rules configurations.

Left Window (Security Group sg-c644eaad):

- Summary:** DEYim
- Inbound Rules:** Selected tab.
- Edit:** Button.
- Table Headers:** Type, Protocol, Port Range, Source, Description.
- Rules:**
 - Type: SSH (22), Protocol: TCP (6), Port Range: 22, Source: 163.239.255.163/32, Description: MySQL/Aurora (3306).
 - Type: MySQL/Aurora (3306), Protocol: TCP (6), Port Range: 3306, Source: sg-c644eaad, Description: MySQL/Aurora (3306).

Right Window (Security Group sg-ce43eda5):

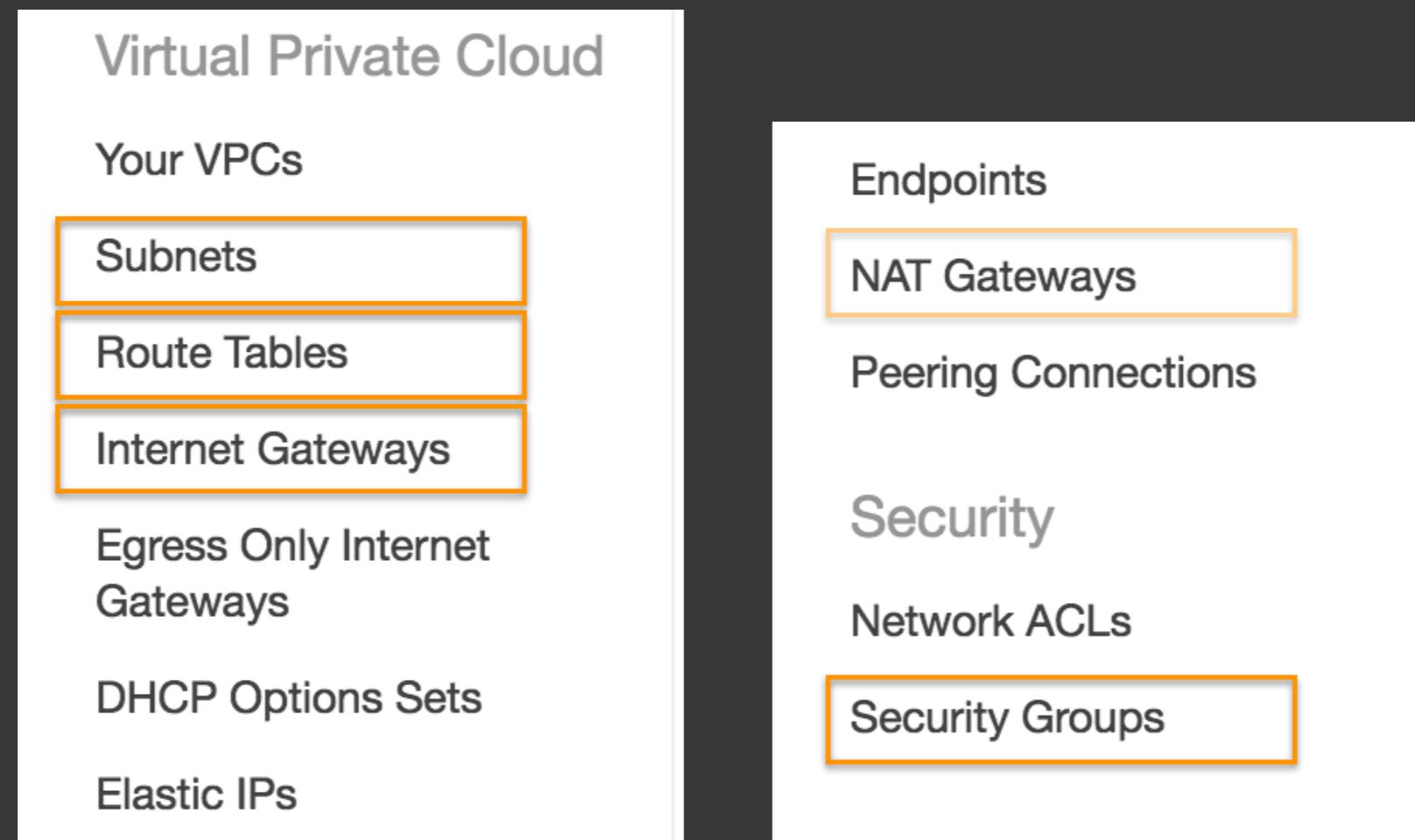
- Summary:** DEYim
- Inbound Rules:** Selected tab.
- Edit:** Button.
- Table Headers:** Type, Protocol, Port Range, Source, Description.
- Rules:**
 - Type: MySQL/Aurora (3306), Protocol: TCP (6), Port Range: 3306, Source: sg-ce43eda5, Description: MySQL/Aurora (3306).

The screenshot shows a single AWS VPC Inbound Rules configuration window.

- Summary:** DEYim
- Inbound Rules:** Selected tab.
- Edit:** Button.
- Table Headers:** Type, Protocol, Port Range, Source, Description.
- Rules:**
 - Type: HTTP (80), Protocol: TCP (6), Port Range: 80, Source: 0.0.0.0/0.
 - Type: SSH (22), Protocol: TCP (6), Port Range: 22, Source: 0.0.0.0/0.

02

VPC 만들기



02

VPC 만들기

서버에 대한 공개적인 액세스를 차단

+

퍼블릭 웹 애플리케이션을 실행

Public Subnet

Private Subnet

인터넷에 연결되어야 하는
Web Server

인터넷에 연결되면 안되는
Database Server

02

VPC 만들기

Public Subnet

Internet Gateway

Private Subnet

NAT Gateway

Route Table

Internet Gateway, NAT Gateway
위한 route table을 VPC에 붙여

Security Group

EC2, RDS를 각각 제어하기 위한
security group을 만들어

03

Elastic Beanstalk



03

Elastic Beanstalk



인프라에 관계없이 AWS 클라우드에서 애플리케이션을
신속하게 배포하고 관리

애플리케이션을 업로드
용량 프로비저닝, 로드 밸런싱, 조정, 애플리케이션 상태 모니터링에
대한 세부 정보를 자동으로 처리
배포가 매우 매우 쉬워짐!

03

Elastic Beanstalk



를 만드려했으나.....

강의 준비하느라 인스턴스 지웠다 만들다를 반복하다보니
인스턴스의 health가 뜯망하기 시작, Launch 가 절대안되는 불상사가 발생
이 강의는 접을 수 밖에 없었다고한다..
눈물을 머금고 일반 EC2로 대체 -> 이따가 할거예요

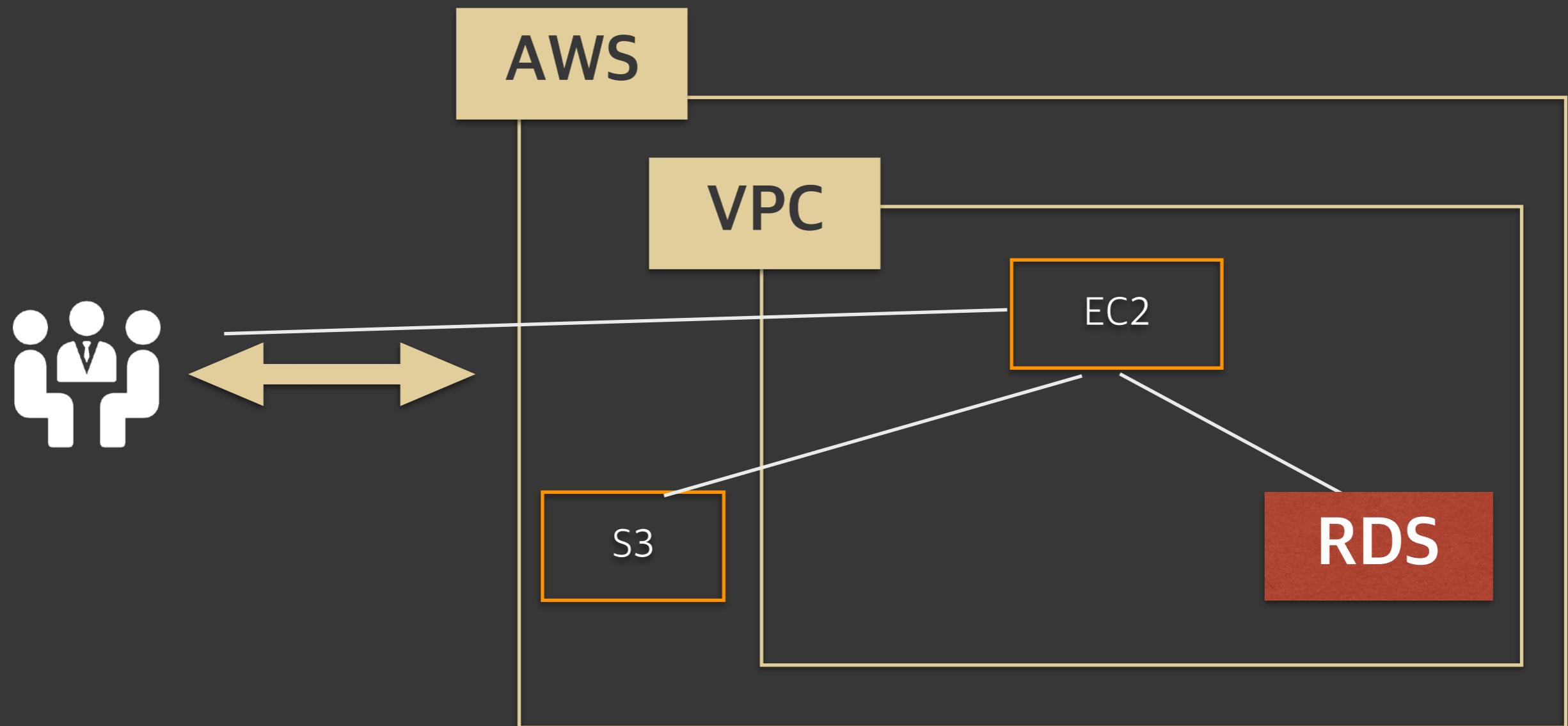
03

Elastic Beanstalk

일단 RDS부터 보고 넘어가겠습니다~

02

VPC 만들기



04

RDS



LIKE LION

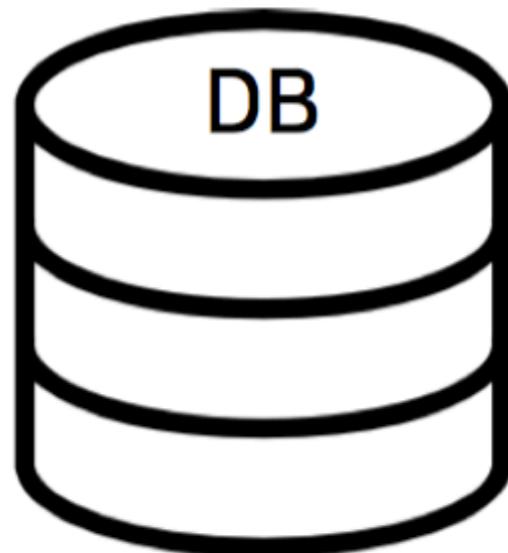
03

Elastic Beanstalk

모델과 데이터베이스에 대한 복습!

DB(데이터베이스)란?

데이터(정보)를 저장할 수 있는 **서비스**



따라서 '**저장**'이라는 기능을 하려면
따로 이용해야 하는 서비스!

레일즈와는 **별개인 서비스!!!**

RDS

데이터베이스를 다루려면?



‘데이터베이스’언어를
알아야지!
(sql문)

출처: 땅강 박로켓의 1학기 강의자료

LIKE LION

04

RDS

루비문법

```
person = Customer.new
person.firstName = "정현"
person.lastName = "박"
person.phoneNumber = "01047105000"
person.emailAddress = "rocket@likelion.org"
person.priority = "coding & startup"
person.save
```

→

자동 변환!

```
SQLQuery9.sql - L...top1\Jeremyk (58) SQLQuery8.sql - L...
INSERT INTO [Test].[dbo].[Customer]
([FirstName]
,[LastName]
,[PhoneNumber]
,[EmailAddress]
,[Priority]
,[CreateDate])
VALUES
(FirstName, varchar(25),>
,LastName, varchar(25),>
,PhoneNumber, varchar(15),>
,EmailAddress, varchar(25),>
,Priority, int,>
,CreateDate, datetime,>)
```



출처: 띵강 박로켓의 1학기 강의자료

LIKE LION

Model이 다룰 수 있는 부분

Id	Name	Phone	Address
1	박정현	01047105555	르호봇
2	임다은	01045612354	르호봇다락방
3	이찬하	01054716523	남문쪽

출처: 땅강 박로켓의 1학기 강의자료

04

RDS

posts 테이블을
조작하려면?
(데이터를 추가/삭제/수정/읽기)

Post 모델!!!

comments 테이블을
새로 만들려면?

마이그레이션!

출처: 땅강 박로켓의 1학기 강의자료

LIKE LION

04

RDS

데이터베이스와 모델은 독립적이다



Rails가 default로 쓰는 sqlite3 말고
다른 데이터베이스도 사용이 가능하겠구나!



그럼 내 맘대로 데이터베이스를 골라볼까?

04

SQL vs NoSQL

Relational DB

SQL, 관계형 데이터베이스

table에 related data를 저장한다.
(column, row를 사용)

전화번호부처럼 매우 구조화된 데이터를 나
타내는 데 쓰임

Non - Relational DB

Not only SQL = NoSQL

Jason 비슷한 형식(name - value)으로
related data를 저장

구조화 되지 않은 / 조금 구조화된 정보들을
저장할 때 쓰임

04

SQL vs NoSQL

Relational DB

구조화: 정리되어있고, join시키기 쉽다
error checking: relationship이 꼭 필요함
reduced duplication
fast join: cross reference 등 할 때 빠름
stability

Non - Relational DB

large data set, speed, unstructured
유연성: no normalization (데이터들이 같은 field를 가질 필요 없음)
easy distribution, fast query

04

SQL vs NoSQL

Writer

Post

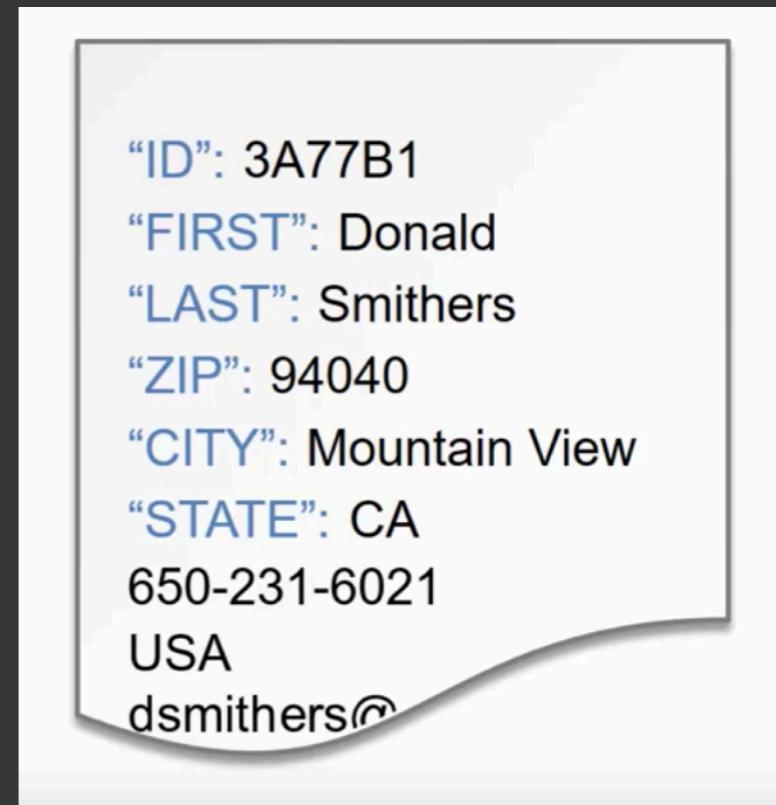
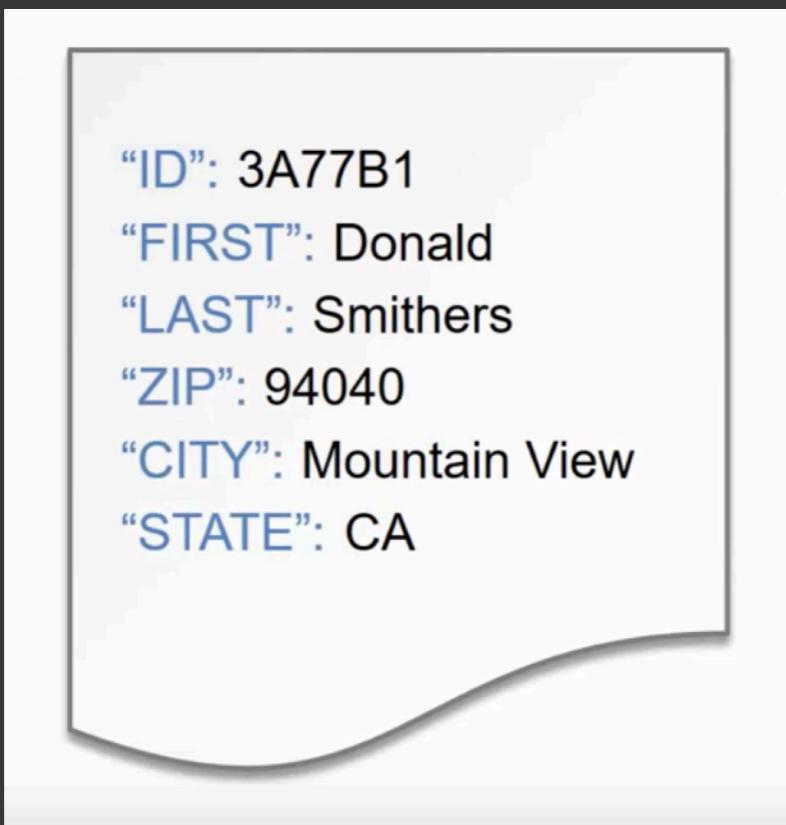
Page-tree

Menu

Comment

Ad

SQL vs NoSQL



04

SQL vs NoSQL

신제품 카탈로그를 DB로 관리한다고 가정해보자. 새로운 제품을 DB에 등록하려고 한다
기존 테이블과 맞지 않는 field가 있다.

Relational DB

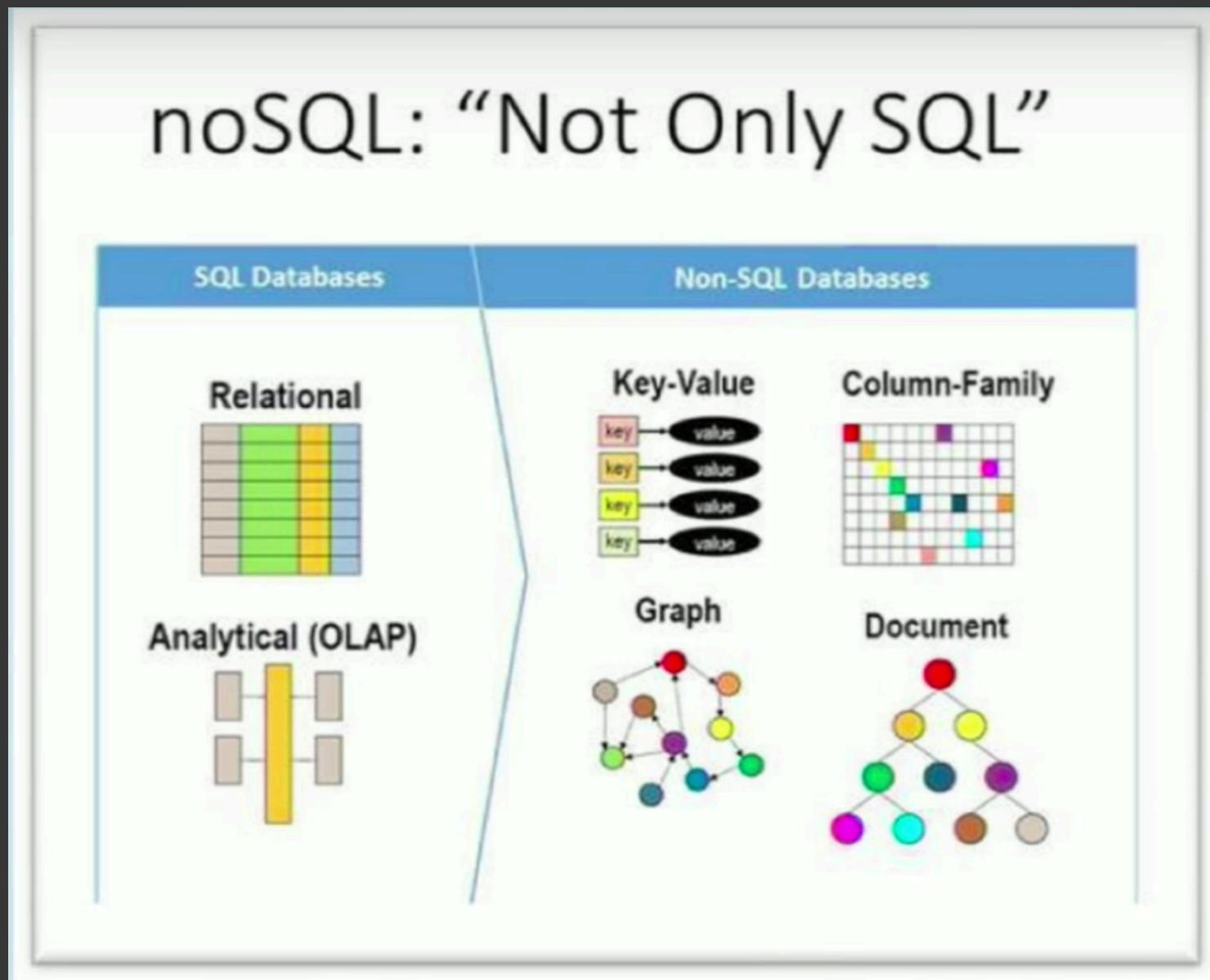
테이블에 새 field를 추가할지
product data를 바꿔줄 지에 대한
결정이 필요함
(힘들..)

Non - Relational DB

그냥 데이터를 추가해놓고,
다음에 common field가 되면 category로
만들어줘도 된다.

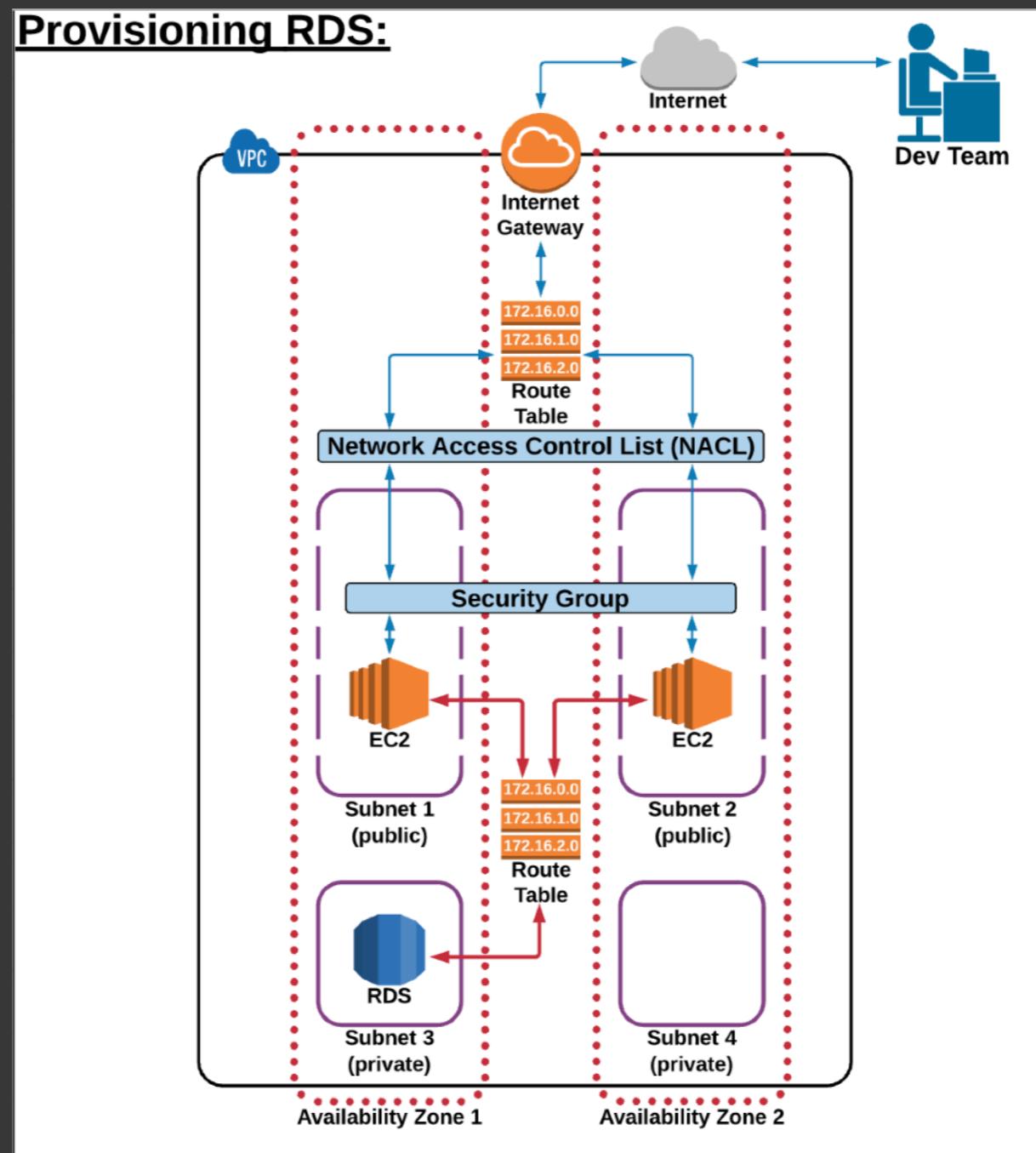
SQL vs NoSQL

noSQL: “Not Only SQL”

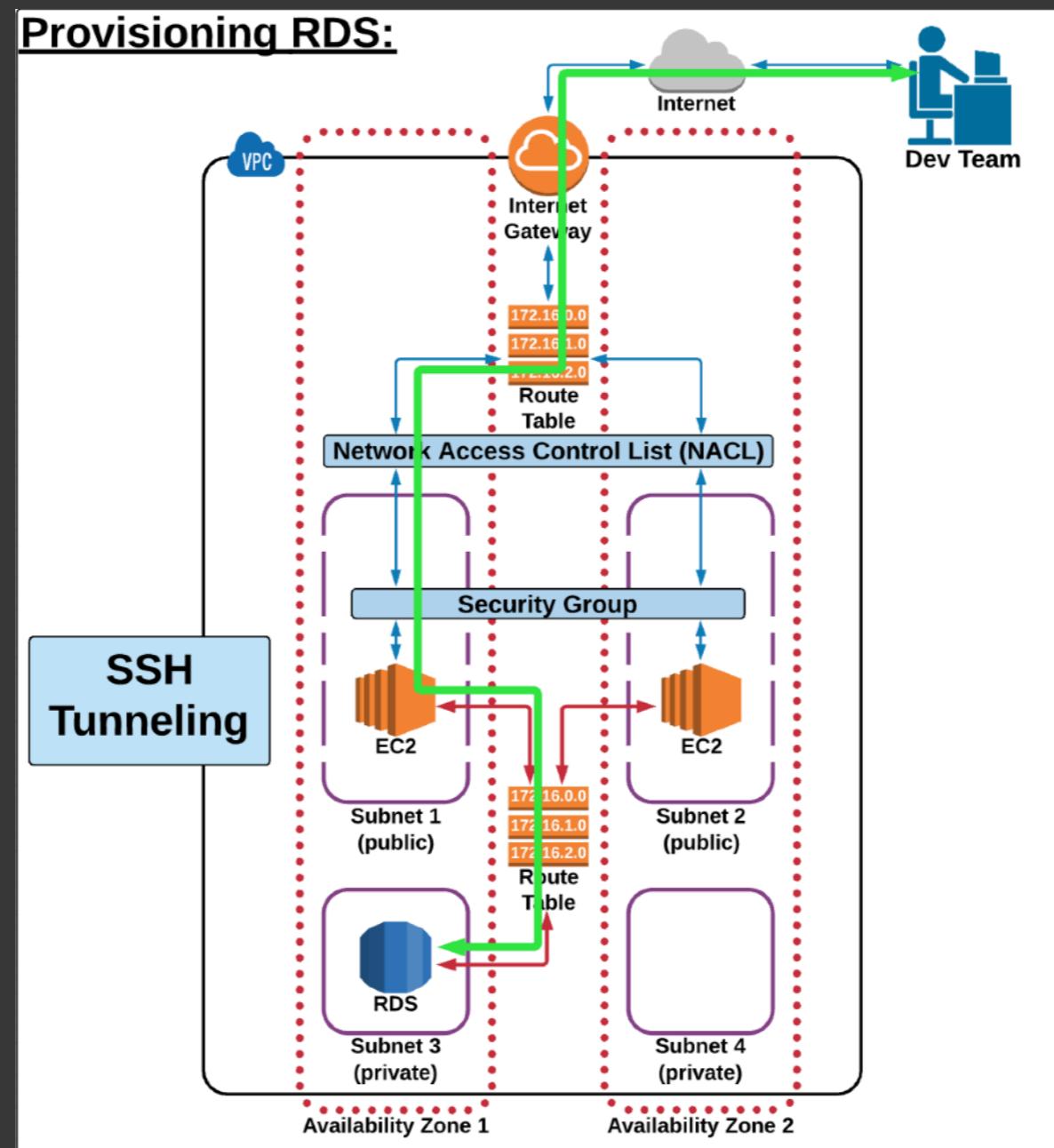


- Document Store
- Column Store
- Graph Database
- Hybrid Cache Store

RDS



RDS



04

RDS

Subnet Group

LIKE LION

Create DB subnet group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

Subnet group details

Name

Description

VPC

VPC Identifier corresponding to the subnets you want to use for the DB subnet group

04

RDS

Add subnets

Add subnet(s) to this subnet group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

[Add all the subnets related to this VPC](#)

Availability zone

ap-northeast-2c ▾

Subnet

subnet-fb8475b7 (10.0.101.0/24) ▾ [Add subnet](#)

Subnets in this subnet group (2)			
Availability zone	Subnet ID	CIDR block	Action
ap-northeast-2a	subnet-6727a10f	10.0.100.0/24	Remove
ap-northeast-2c	subnet-fb8475b7	10.0.101.0/24	Remove

04

RDS

RDS instance

04

RDS

다 default, free tier setting으로 설정하여
계속~

Settings

DB instance identifier [info](#)
Specify a name that is unique for all DB instances owned by your AWS account in the current region.

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".

Master username [info](#)
Specify an alphanumeric string that defines the login ID for the master user.

Master Username must start with a letter.

Master password [info](#)

Master Password must be at least eight characters long, as in "mypassword".

Confirm password [info](#)

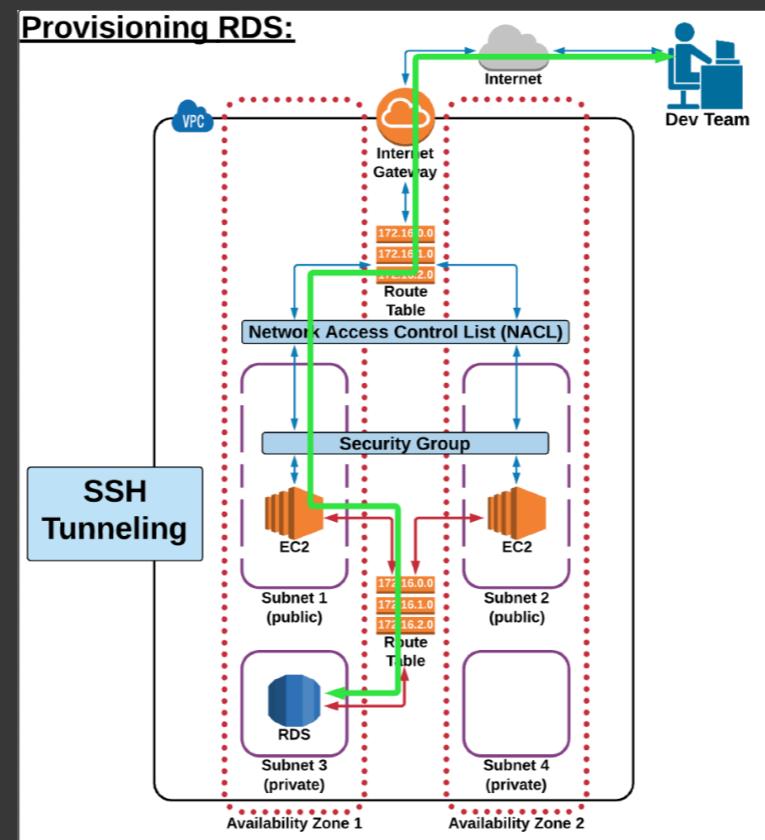
04

RDS

Configurations	Security and network	Instance and IOPS	Maintenance details
ARN arn:aws:rds:ap-northeast-2:331932344038:db:test	Availability zone ap-northeast-2a	Instance Class db.t2.micro	Auto minor version upgrade Yes
Engine MySQL 5.6.37	VPC LionApp (vpc-29dc6b41)	Storage Type General Purpose (SSD)	Maintenance window fri:13:06-fri:13:36
License Model General Public License	Subnet group private subnet	Storage 20 GB	Backup window 18:43-19:13
Created Time Sat Nov 25 20:12:39 GMT+900 2017	Subnets subnet-7117e53d subnet-c3bd3fab	Availability and durability	Pending Modifications None
DB Name test	Security groups default (sg-3f47e954) (active) DEYim DBgroup (sg-b242ecd9) (active)	DB instance status available	Pending maintenance none
Username test	Publicly accessible No	Multi AZ No	Encryption details
Option Group default:mysql-5-6	Endpoint test.cdorjyh00dth.ap-northeast-2.rds.amazonaws.com	Automated backups Enabled (7 Days)	Encryption enabled No
Parameter group default.mysql5.6 (in-sync)		Latest restore time November 27, 2017 at 3:30:00 PM UTC+9	
Copy tags to snapshots			

RDS

RDS instance를 위한 EC2



04

RDS

RDS를 조작할 EC2 생성합니다.

- ‘UBUNTU AMI 선택’
- VPC 만들 때 같이 만든 ‘DEYim’ security group
 - 우리가 만든 VPC
 - Public Subnet A
 - Elastic IP address 사용

04

RDS

EC2에 들어가십시오
설치과정에서 mysql 비밀번호를 설정하라면, 아까 RDS 설정한 애랑 똑같이 설정 (편의상)

```
sudo yum update
sudo yum install mysql-server mysql-client libmysqlclient-dev
mysql -h RDS endpoint 주소 -P 3306 -u RDS유저이름 -p
아까 설정한 비번 치기
```

04

RDS

```
bye
ubuntu@ip-10-0-0-168:~$ mysql -h test.cdorjyh00dth.ap-northeast-2.rds.amazonaws.com -P 3306 -u test -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 322
Server version: 5.6.37-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE `데이터베이스명` CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

mysql> █
```

04

RDS

이제 RDS에 들어올 수 있게되었습니다.

짝짝짝

04

RDS

간단한 MYSQL 명령어 -> 아직 하셔도 암것도 없어염

show databases;

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| development    |
| innodb        |
| mysql          |
| performance_schema |
| sys            |
| test           |
+-----+
7 rows in set (0.00 sec)
```

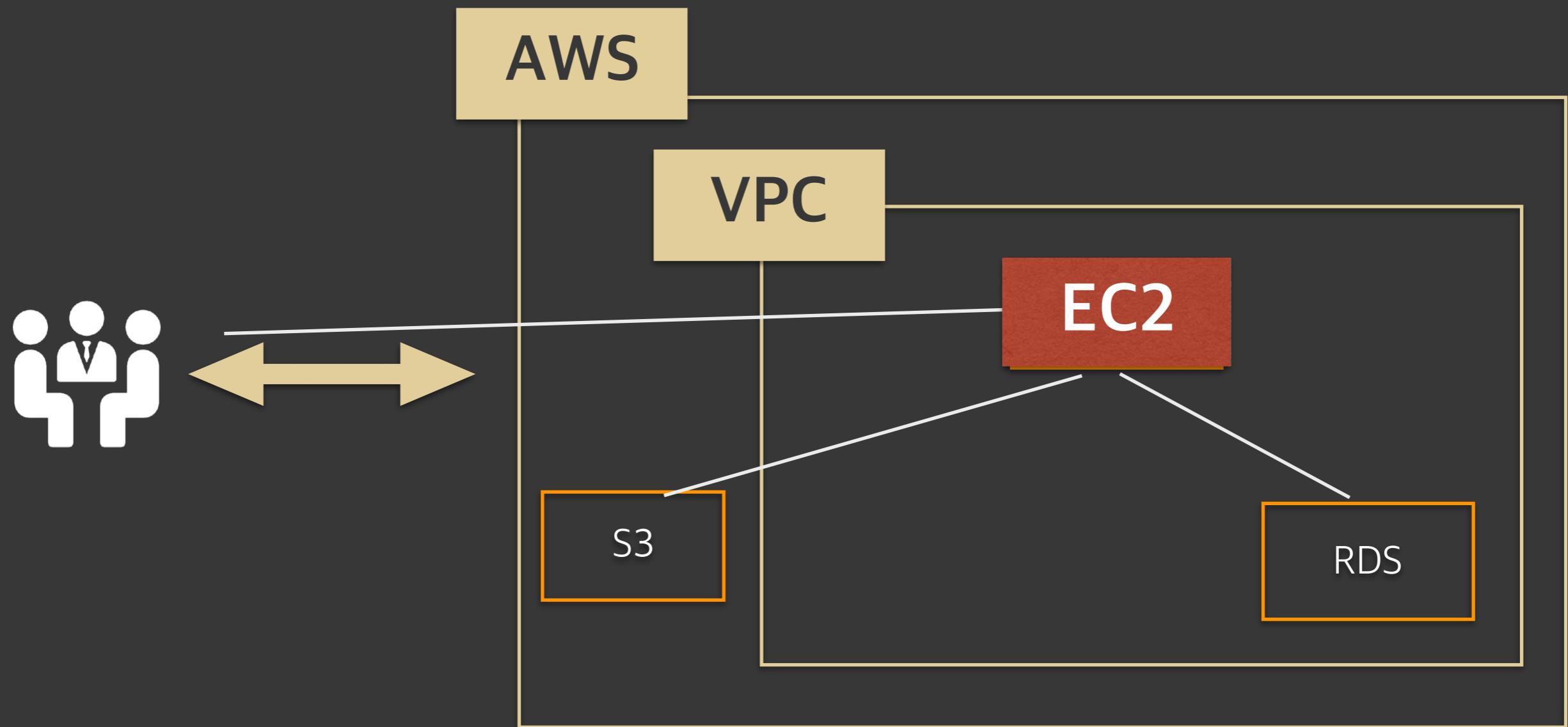
use 데이터베이스이름;
show tables;

```
Database changed
mysql> show tables;
+-----+
| Tables_in_test      |
+-----+
| ar_internal_metadata |
| posts                |
| schema_migrations   |
| univs                |
| users                |
+-----+
5 rows in set (0.00 sec)
```

desc 테이블이름;

```
mysql> desc univs;
+-----+-----+
| Field      | Type       |
+-----+-----+
| id         | bigint(20) |
| name       | varchar(255)|
| created_at | datetime  |
| updated_at | datetime  |
+-----+-----+
```

EC2



05

EC2

이제 Web Server 용 EC2를 만들 차례입니다.

05

EC2

Web Server 용 EC2를 생성합니다.

- ‘UBUNTU AMI 선택’
- VPC 만들 때 같이 만든 ‘DEYim Web Server’ security group
 - 우리가 만든 VPC
 - Public Subnet A
 - Elastic IP address 사용

05

EC2

이제 Ruby와 Rails를 깔 것입니다.
리눅스 명령어들을 잘 따라오세요

EC2

#우분투를 기본적으로 update해주기

```
sudo apt-get update
```

Dependency들 설치해주기. 즉 레일즈 돌리는 데 필요한 것들임. 자세히 노노

```
sudo apt-get install -y git-core curl zlib1g-dev build-essential libssl-dev  
libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev  
libcurl4-openssl-dev python-software-properties libffi-dev nodejs mysql-server  
mysql-client libmysqlclient-dev imagemagick
```

Install rbenv and ruby-build plugins

```
git clone https://github.com/rbenv/rbenv.git ~/.rbenv  
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc  
echo 'eval "$(rbenv init -)"' >> ~/.bashrc  
git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build  
echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
```

컴퓨터 껌다 키기랑 비슷하게

```
exec $SHELL
```

sudo apt-get / git clone / echo / >> / '~/. ' 뭔지 곰곰히 생각해보기

EC2

```
# ruby 깔기
rbenv install -v "2.3.4"
rbenv global "2.3.4"
ruby -v
```

```
# install bundler
echo "gem: --no-document" > ~/.gemrc
gem install bundler
```

```
#rails 설치
gem install rails
gem install rails -v 4.2.7
rbenv rehash
rails -v
```

05

EC2

새 레일즈 APP을 만듭니다. 단! MYSQL을 쓰는 앱!
rails new APP이름 -d mysql

app으로 들어가서 bundle install하고 database.yml
을 수정합니다.

```
cd APP이름  
bundle install  
cd config  
ls
```

어떤 파일들이 있나 보시고,
cat database.yml
#평소 만들었을 때의 database.yml과 조금 다릅니다.

05

EC2

```
# SQLite version 3.x
#   gem install sqlite3
#
#   Ensure the SQLite 3 gem is defined in your Gemfile
#   gem 'sqlite3'
#
default: &default
  adapter: sqlite3
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  timeout: 5000

development:
<<: *default
  database: db/development.sqlite3

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
<<: *default
  database: db/test.sqlite3

production:
<<: *default
  database: db/production.sqlite3
```

```
# MySQL. Versions 5.1.10 and up are supported.
#
#   Install the MySQL driver
#   gem install mysql2
#
#   Ensure the MySQL gem is defined in your Gemfile
#   gem 'mysql2'
#
# And be sure to use new-style password hashing:
#   http://dev.mysql.com/doc/refman/5.7/en/old-client.html
#
default: &default
  adapter: mysql2
  encoding: utf8
  pool: 5
  username: test
  password: test1234
  #   socket: /var/run/mysqld/mysqld.sock

development:
  adapter: mysql2
```

일반 Rails App

MYSQL 용 Rails App

05

EC2

AWS document에 있는 내용

```
production:  
  adapter: mysql2  
  encoding: utf8  
  database: <%= ENV['RDS_DB_NAME'] %>  
  username: <%= ENV['RDS_USERNAME'] %>  
  password: <%= ENV['RDS_PASSWORD'] %>  
  host: <%= ENV['RDS_HOSTNAME'] %>  
  port: <%= ENV['RDS_PORT'] %>
```

여기에 맞춰 database.yml development 부분만 수정해보자!

05

EC2

```
development:  
  adapter: mysql2  
  encoding: utf8  
  database: test  
  pool: 5  
  username: test  
  password: test1234  
  host: test.cdorjyh00dth.ap-northeast-2.rds.amazonaws.com
```

친절하게 제거까지 보여드립니다
host는 end point 찾아서 넣으세요

05

EC2

comments 테이블을
새로 만들려면?

마이그레이션!

`rails g migration create_comments`
`rake db:migrate`

05

EC2

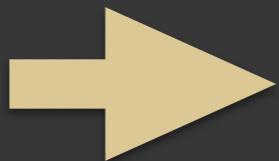
```
ubuntu@ip-10-0-0-144:~/DemoApp$ rails g migration create_comments
Running via Spring preloader in process 9943
  invoke  active_record
    create    db/migrate/20171127074244_create_comments.rb
ubuntu@ip-10-0-0-144:~/DemoApp$ rake db:migrate
== 20171127074244 CreateComments: migrating =====
=====
-- create_table(:comments)
 -> 0.0177s
== 20171127074244 CreateComments: migrated (0.0179s) =====
=====
```

**rails g migration create_comments
rake db:migrate**

05

EC2

```
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| ar_internal_metadata |
| posts |
| schema_migrations |
| univs |
| users |
+-----+
5 rows in set (0.00 sec)
```



```
mysql> show tables;
+-----+
| Tables_in_test      |
+-----+
| ar_internal_metadata |
| comments            |
| posts               |
| schema_migrations |
| univs               |
| users               |
+-----+
6 rows in set (0.00 sec)
```

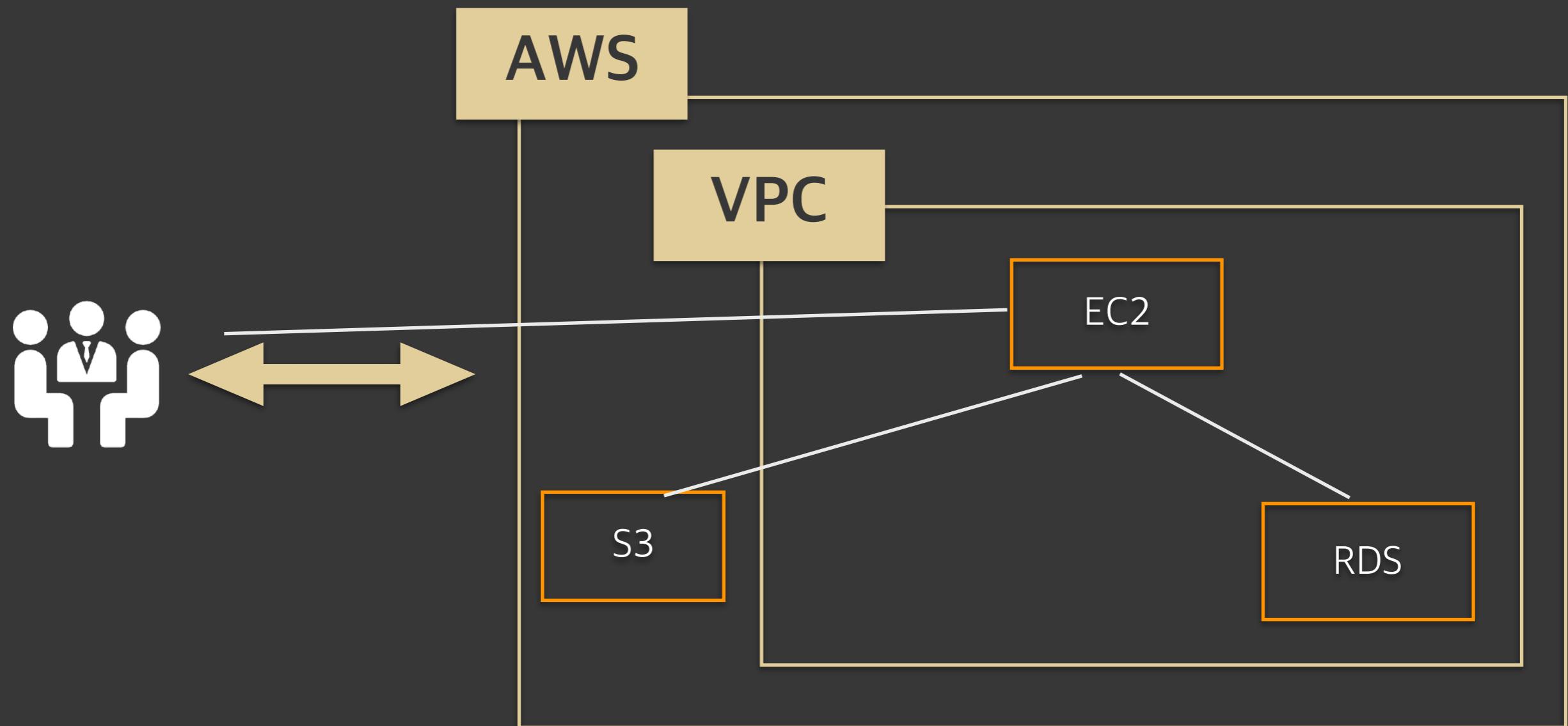
use test;
show tables;

05

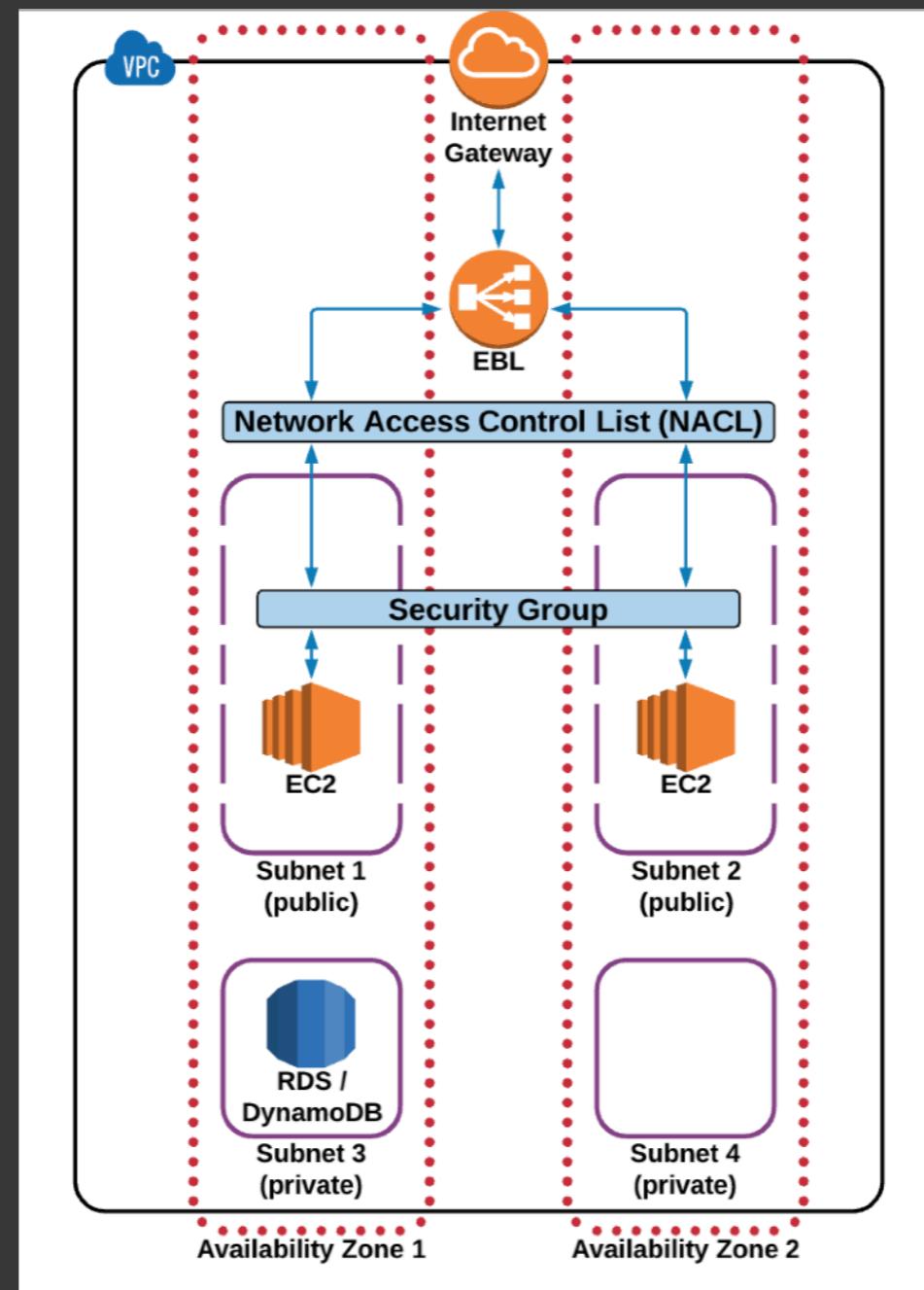
EC2

```
| information_schema |  
| development       |  
| innodb            |  
| mysql             |  
| performance_schema |  
| sys               |  
| test              |  
+-----+  
7 rows in set (0.00 sec)  
  
mysql> use test  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_test |  
+-----+  
| ar_internal_metadata |  
| posts           |  
| schema_migrations |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> desc posts  
    ->  
  
^C  
mysql> show tables;  
+-----+  
| Tables_in_test |  
+-----+  
| ar_internal_metadata |  
| posts           |  
| schema_migrations |  
| users            |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql> []  
  
(in /home/ubuntu/DemoApp)  
== 20171127053133 CreatePosts: migrating ======  
=====  
-- create_table(:posts)  
  -> 0.0157s  
== 20171127053133 CreatePosts: migrated (0.0158s) ======  
=====  
  
ubuntu@ip-10-0-0-144:~/DemoApp/config$ rails s  
=> Booting Puma  
=> Rails 5.1.4 application starting in development  
=> Run `rails server -h` for more startup options  
Puma starting in single mode...  
* Version 3.11.0 (ruby 2.3.4-p301), codename: Love Song  
* Min threads: 5, max threads: 5  
* Environment: development  
* Listening on tcp://0.0.0.0:3000  
Use Ctrl-C to stop  
^C- Gracefully stopping, waiting for requests to finish  
== puma shutdown: 2017-11-27 05:36:26 +0000 ===  
- Goodbye!  
Exiting  
ubuntu@ip-10-0-0-144:~/DemoApp/config$ cd ..  
ubuntu@ip-10-0-0-144:~/DemoApp$ rails g model User name:string score:integer  
Running via Spring preloader in process 3399  
  invoke  active_record  
    create   db/migrate/20171127054127_create_users.rb  
    create   app/models/user.rb  
  invoke  test_unit  
    create   test/models/user_test.rb  
    create   test/fixtures/users.yml  
ubuntu@ip-10-0-0-144:~/DemoApp$ rake db:migrate  
== 20171127054127 CreateUsers: migrating ======  
=====  
-- create_table(:users)  
  -> 0.0189s  
== 20171127054127 CreateUsers: migrated (0.0192s) ======  
=====  
  
ubuntu@ip-10-0-0-144:~/DemoApp$ ]
```

Wrap Up

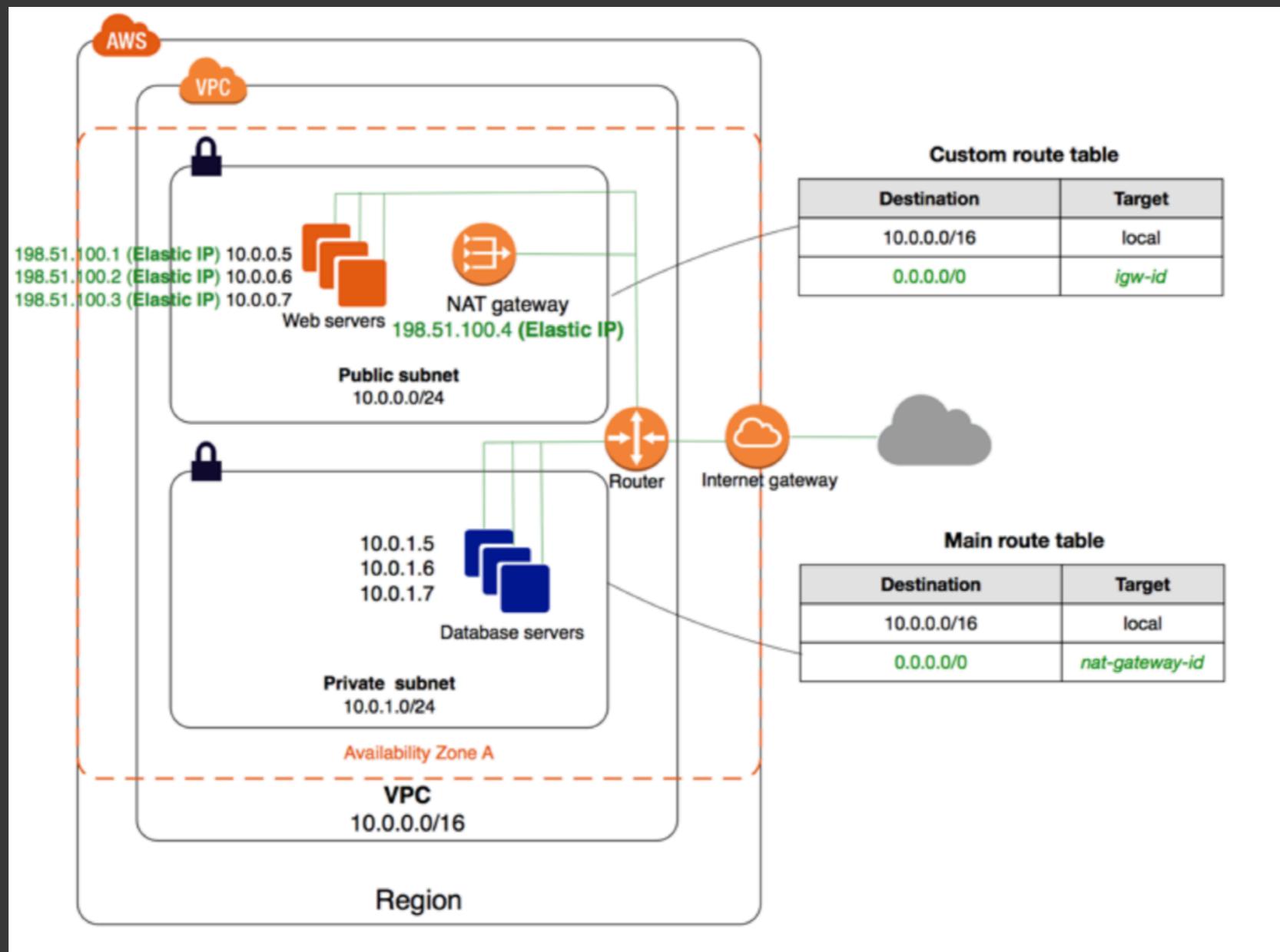


Wrap Up



06

Wrap Up



Wrap Up

