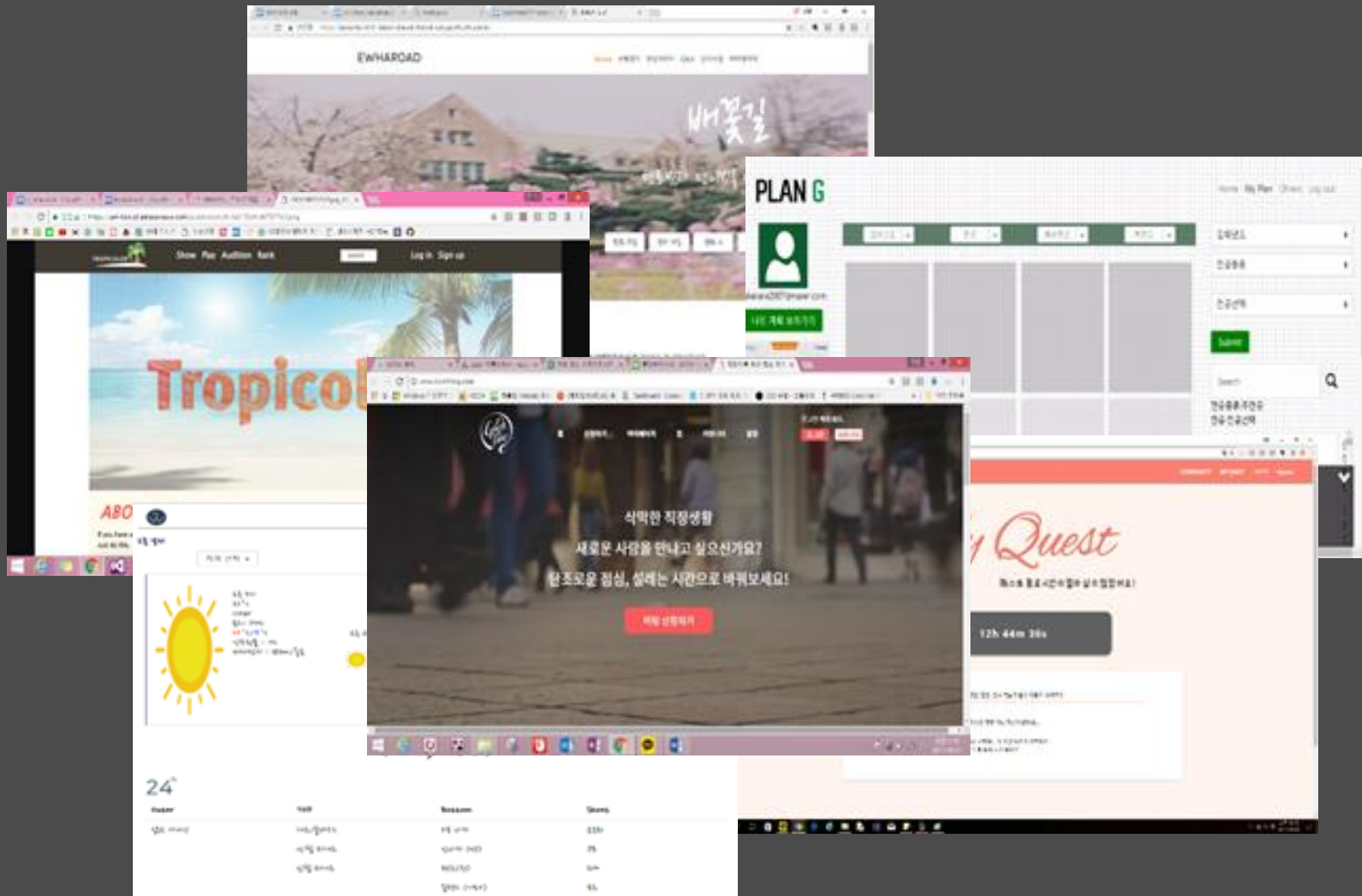


Heroku & AWS



우리가 만든 서비스를
실제로 운영하려면?

Deploy

애플리케이션이나 소프트웨어를 배포!

Heroku

배포를 도와주는 도구!

1. Heroku 가입

<http://www.heroku.com>

가입해주세요!

2. 배포환경 만들기

[gemfile]

1. `gem 'sqlite3' => group :development, :test` 로 옮기기
2. `group :production` 생성
3. `Group :production` 에 `gem 'pg'` , `gem 'rails_12factor'` 추가

2. 배포환경 만들기

```
gem 'rails', '4.2.5'
gem 'sqlite3'
gem 'sass-rails', '~> 5.0'
gem 'uglifier', '>= 1.3.0'
gem 'coffee-rails', '~> 4.1.0'
gem 'jquery-rails'
gem 'turbolinks'
gem 'jbuilder', '~> 2.0'
gem 'sdoc', '~> 0.4.0', group: :doc
```

```
group :development, :test do
  gem 'byebug'
end
```

```
group :development do
  gem 'web-console', '~> 2.0'
end
```

```
gem 'rails', '4.2.5'
gem 'sass-rails', '~> 5.0'
gem 'uglifier', '>= 1.3.0'
gem 'coffee-rails', '~> 4.1.0'
gem 'jquery-rails'
gem 'turbolinks'
gem 'jbuilder', '~> 2.0'
gem 'sdoc', '~> 0.4.0', group: :doc
```

```
group :development, :test do
  gem 'byebug'
  gem 'sqlite3'
end
```

```
group :development do
  gem 'web-console', '~> 2.0'
end
```

```
group :production do
  gem 'pg'
  gem 'rails_12factor'
end
```


3.Heroku 에 올리기

```
$ git init
```

```
$ git add .
```

```
$ git commit -m "쓰고 싶은 메시지"
```

```
$ sudo apt-get install heroku
```

```
$ heroku login (가입한 e-mail과 password 입력!)
```

```
$ heroku create [app_name]
```

(배포 후에 root_url은 [app_name].herokuapp.com 이 된다!)

```
$ git push heroku master
```

```
$ heroku run rake db:migrate
```

```
zzunvely:~/workspace (master) $ heroku create zzun
Creating ● zzun... done
https://zzun.herokuapp.com/ | https://git.heroku.com/zzun.git
```

3. Heroku 수정사항 반영하기

```
$ git add
```

```
$ git commit -m "수정관련 메시지"
```

```
$ git push heroku master
```

```
(DB를 수정했다면) $ heroku run rake db:migrate
```

사실 Heroku는 에러가 많이나요ㅠㅠ
프로젝트상의 문제일 수도 있고!
버전의 문제일 수 도 있어요!

AWS

(Amazon Web Service)



1. AWS

Cloud Computing?

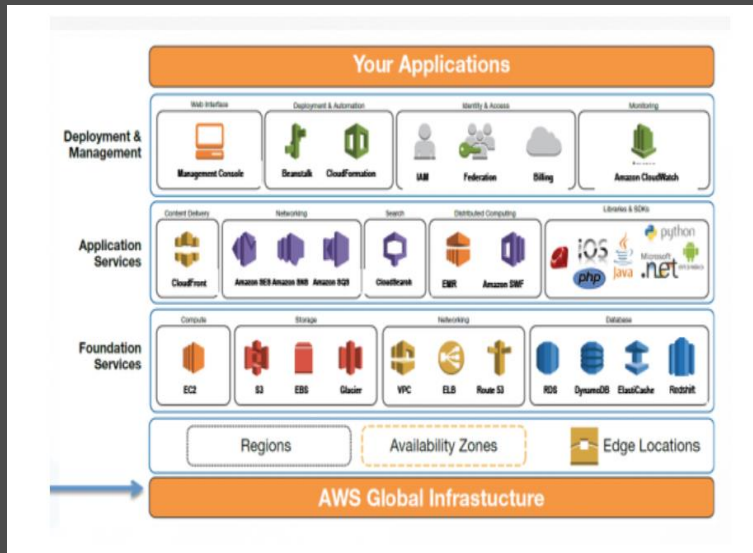


정보처리를 자신의 컴퓨터가 아닌
인터넷으로 연결된 다른 컴퓨터로

인프라 서비스 (IaaS)

서버, DB, 네트워크, Storage 등을
각자의 필요에 따라 선택하여
빌려쓰는 컴퓨터!

1. AWS

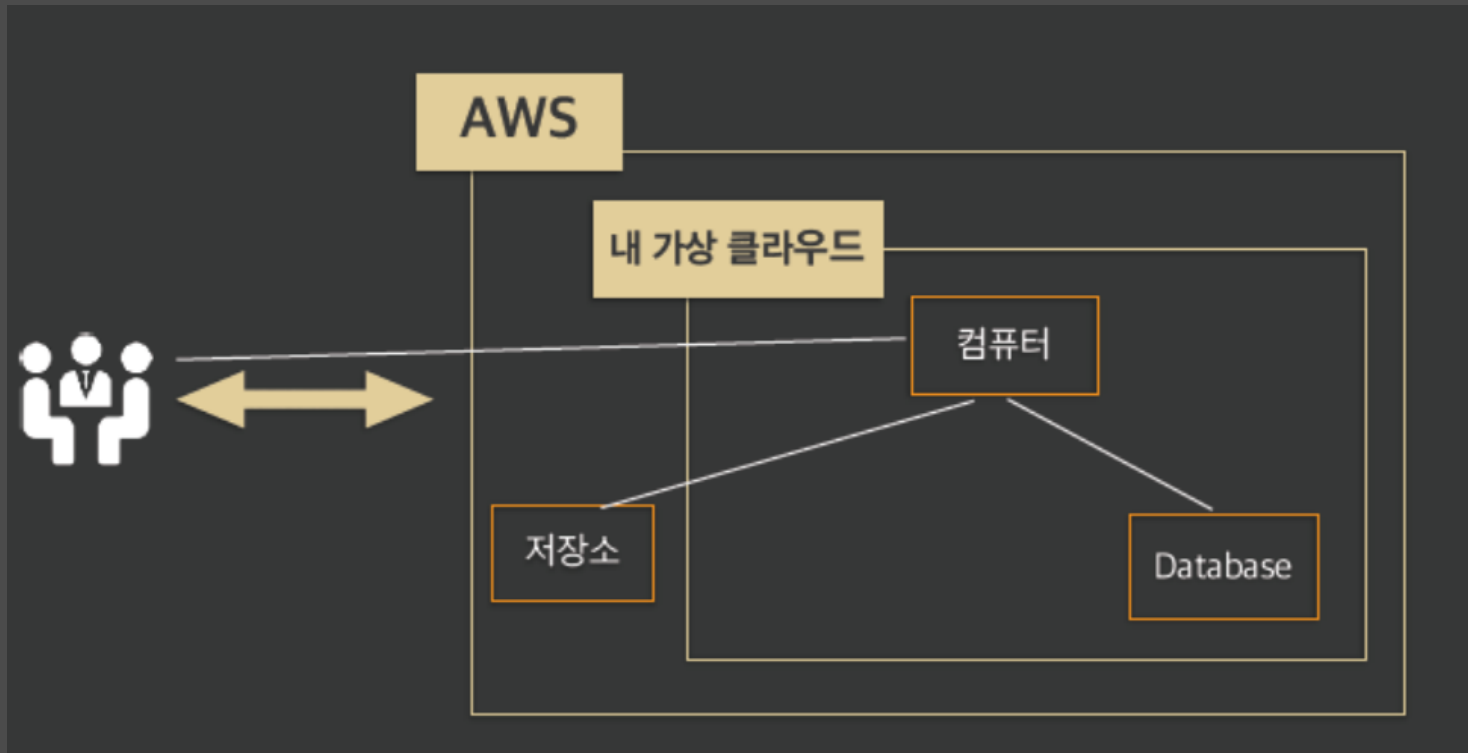


혼자서 모든 인프라 구축, 관리가능

서버 확장, 분산의 자동화 가능
- 급격한 사용량에 대응가능

1. AWS

Amazon architecture



1. AWS

– EC2(Elastic Computer Cloud)



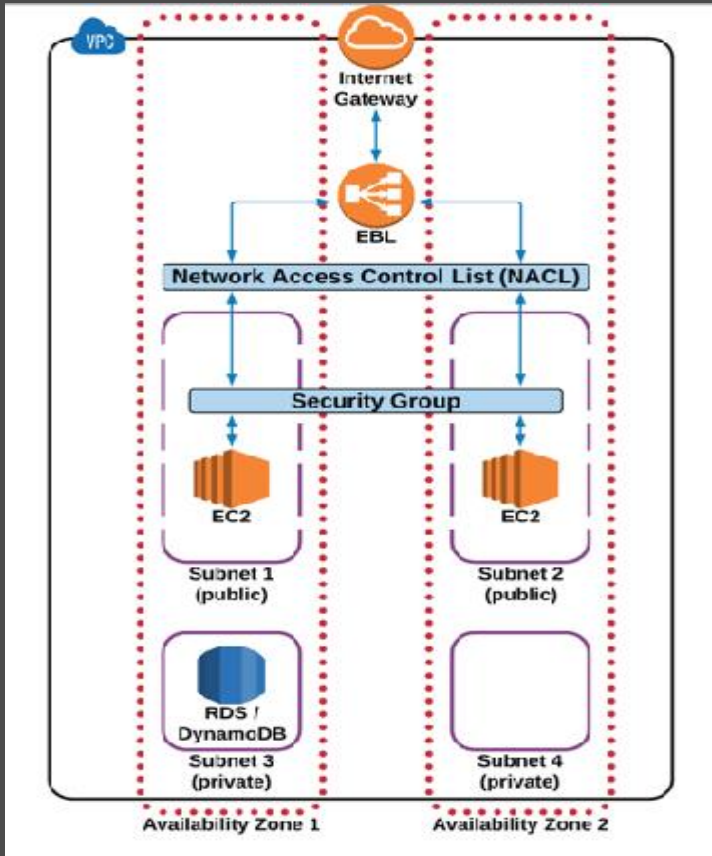
Amazon EC2

인스턴스

- 실행되는 가상의 서버
- OS, CPU, Memory 를 가짐

1. AWS

– EC2(Elastic Computer Cloud)



인스턴스 레벨의 보안

- 내 instance로의 접근을 통제하는 방화벽

+ 인스턴스에 SSH와 RDP로 접근하려고
public key, private key 생성

Key = 비밀번호

1. AWS

– S3(Simple Storage Service)



저장소 (=외장 하드)

= 정보를 저장하는 서비스

주로, 대용량 파일 저장

1. AWS

– S3(Simple Storage Service)

Bucket

unique all across AWS

Folder

Object

고유의 주소를 가짐

1. AWS

– RDS(Relational Database Service)



* DB(데이터베이스)란?

데이터(정보)를 저장하는 서비스

* 관계형데이터란(SQL)?

table 에 column, row를 사용하여 데이터를 저장

구조화된 데이터를 나타내는 데 쓰인다

〈참고〉

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

1. AWS

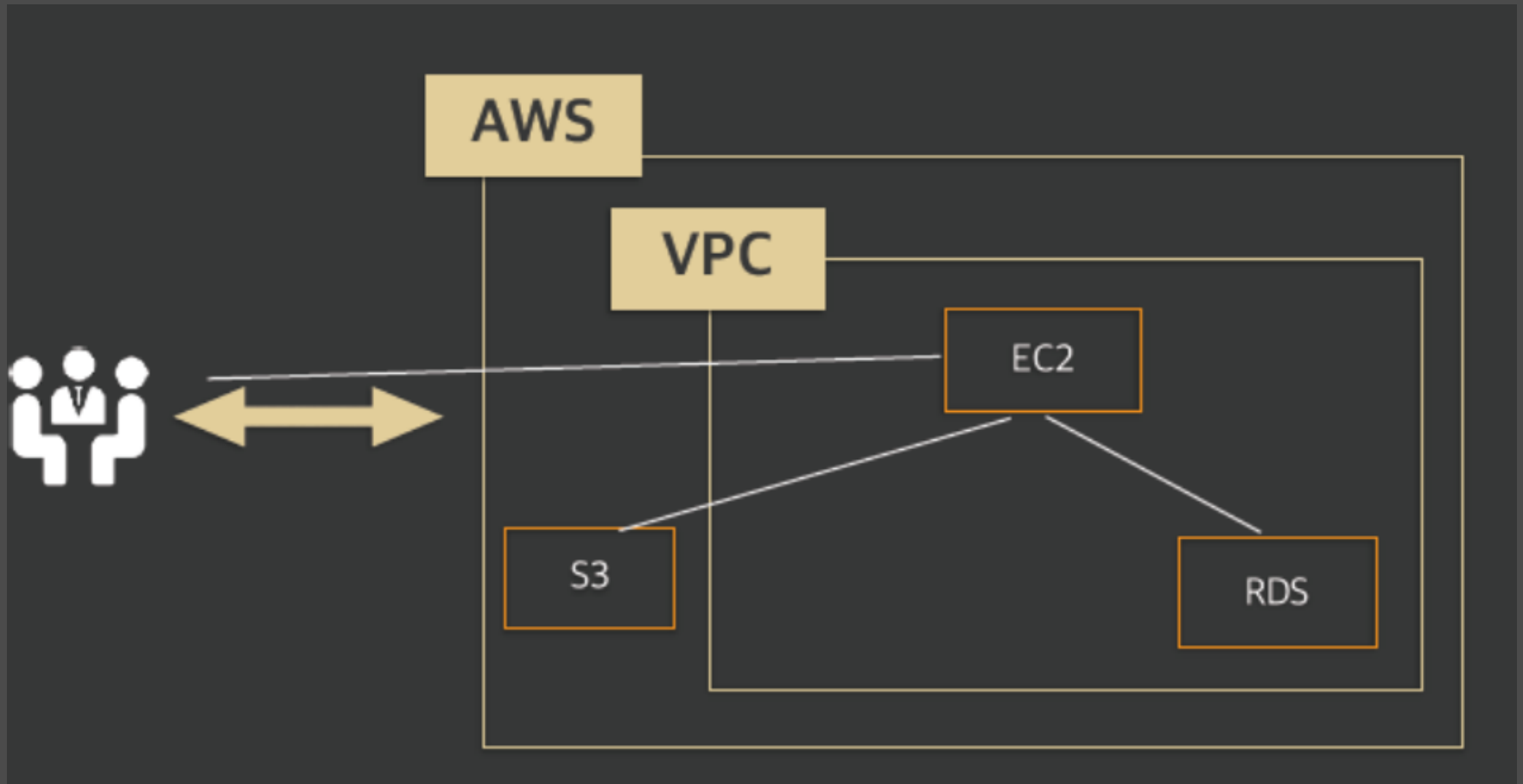
– RDS(Relational Database Service)



컴퓨터, 서버에 DB구축되어있음

하지만, 서버 규모커지면
-> DB관리 어려움
-> RDS 해결!

1. AWS



1. AWS

<참고>

* VPC란?

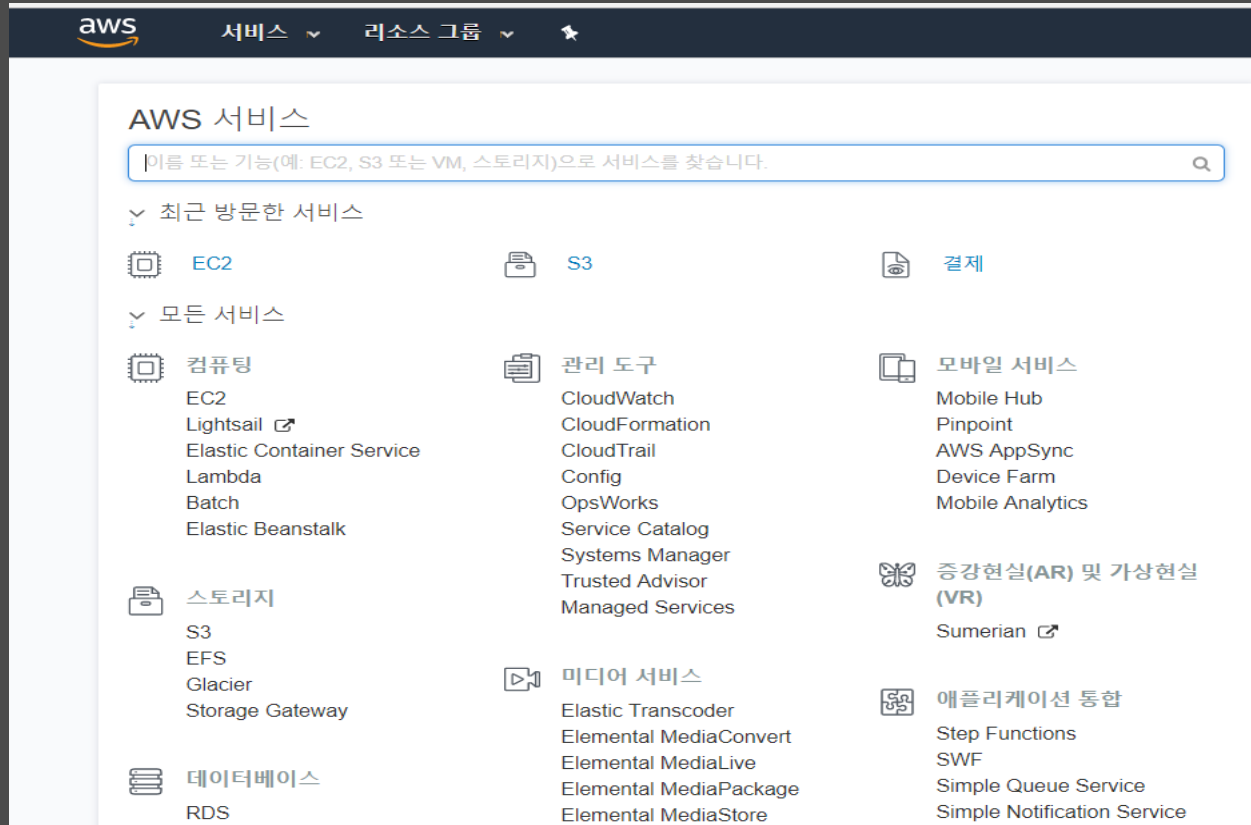
사용자의 AWS 계정 전용 가상 네트워크!

EC2 인스턴스 같은 AWS 인프라를 VPC에서 실행

AWS클라우드 안에서 다른 가상네트워크와 분리

IP 주소 범위를 선택, 서브넷 생성 등등 해서 보안설정가능

2. [실습] EC2 이용해서 서버구축하기!




2. [실습] EC2 이용해서 서버구축하기!



컴퓨팅

EC2

Lightsail 

Elastic Container Service

Lambda

Batch

Elastic Beanstalk



스토리지

S3

EFS

Glacier

Storage Gateway



데이터베이스

RDS

DynamoDB

ElastiCache

Amazon Redshift

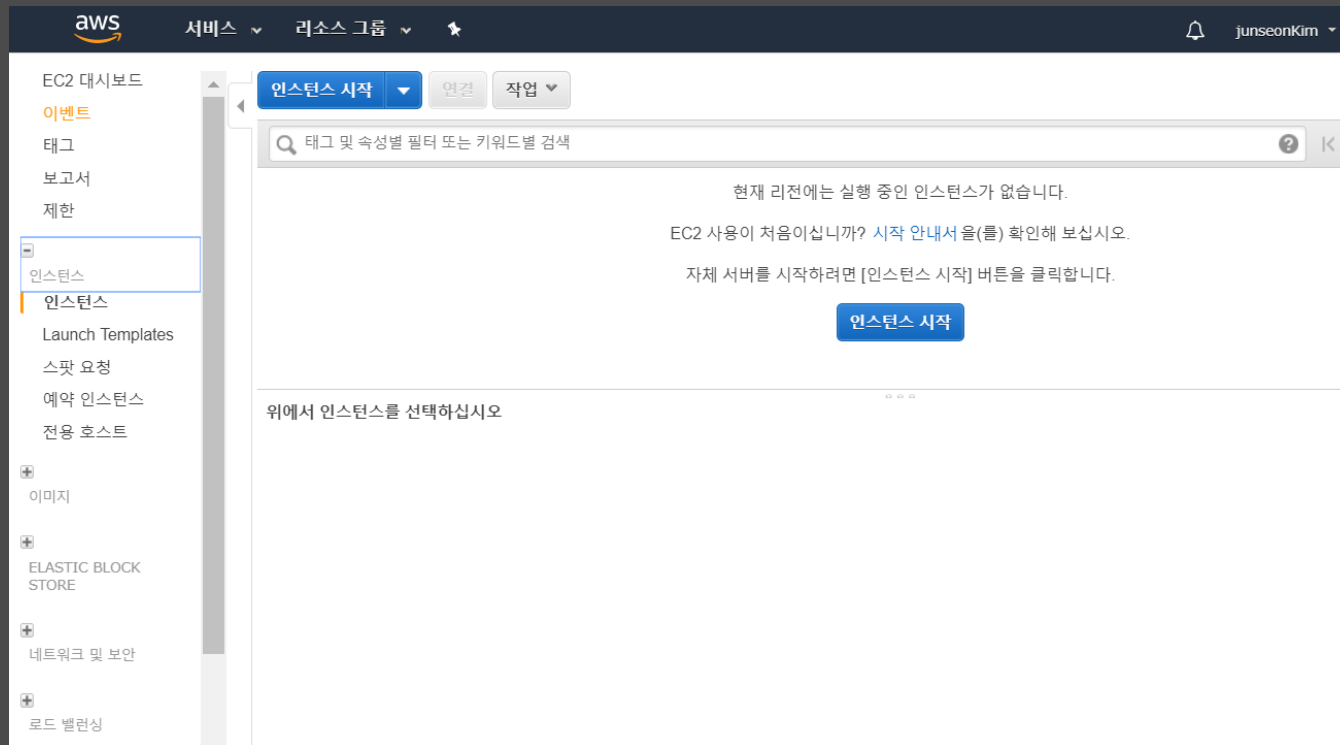
Elastic Compute Cloud

Simple Storage Service

Relational Database Service


2. [실습] EC2 이용해서 서버구축하기!

〈인스턴스 만들기〉



1. AMI 선택
2. 인스턴스 유형 선택
3. 인스턴스 구성
4. 스토리지 추가
5. 태그 추가
6. 보안그룹 구성
7. 검토

2. [실습] EC2 이용해서 서버구축하기!



서비스 ▾ 리소스 그룹 ▾

🔔 junseonKim ▾ 서울 ▾ 지원

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 1: Amazon Machine Image(AMI) 선택

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버, 애플리케이션)이 포함된 템플릿입니다. AWS, 사용자 커뮤니티 또는 AWS Marketplace에서 제공하는 AMI를 선택하거나, 자체 AMI 중 하나를 선택할 수도 있습니다.

빠른 시작


나의 AMI

AWS Marketplace

커뮤니티 AMI

☐ 프리 티어만 ⓘ

AMI 1~34/34




Amazon Linux
프리 티어 사용 가능

Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type - ami-1196317f

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

루트 디바이스 유형: ebs 가상화 유형: hvm

선택 64비트




Red Hat
프리 티어 사용 가능

Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type - ami-0f5a8361

Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type

루트 디바이스 유형: ebs 가상화 유형: hvm

선택 64비트




SUSE Linux
프리 티어 사용 가능

SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-f067c09e

SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

루트 디바이스 유형: ebs 가상화 유형: hvm

선택 64비트




Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-3066c15e

Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

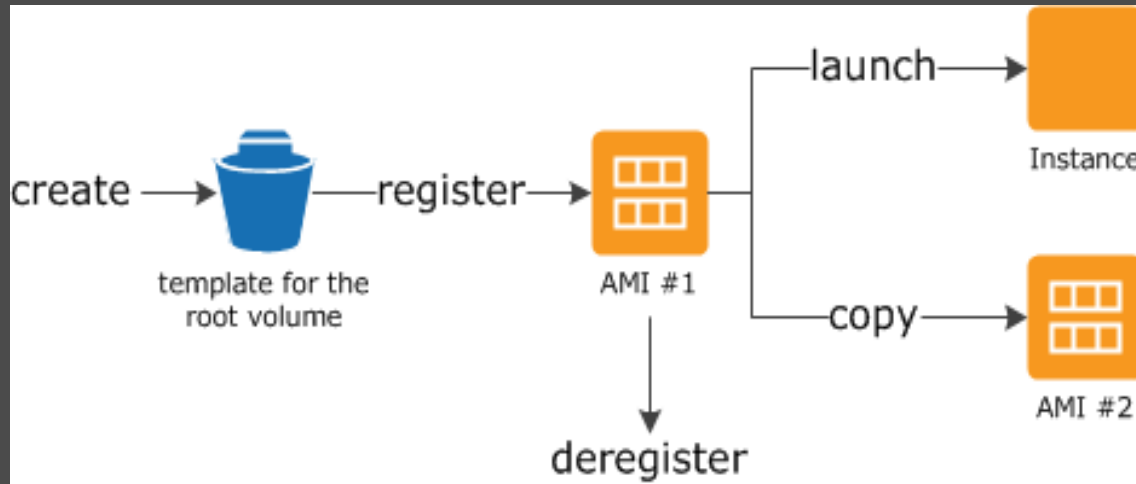
프리 티어 사용 가능

선택 64비트

 LIKE LION

2. [실습] EC2 이용해서 서버구축하기!


AMI(Amazon Machine Image)



Instance를 만드는 데 필요한 정보들을 저장해 놓은 template

-운영체제 / 어플리케이션서버 / 어플리케이션 등

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ ★

🔔 junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 2: 인스턴스 유형 선택

Amazon EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스
모리, 스토리지 및 네트워킹 용량이 다양하게 조합되어 있으며, 애플리케이션에 사용할 조
유형이 컴퓨팅 요건을 충족하는 방식에 대해 [자세히 알아보기](#)

필터링 기준: 모든 인스턴스 유형 ▾ 현재 세대 ▾ 열 표시/숨기기


프리티어 = 무료!

현재 선택된 항목: t2.micro (Variable ECU, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB 메모리, EBS 전용)

	패밀리 ▾	유형 ▾	vCPUs ⓘ ▾	메모리 (GiB) ▾	인스턴스 스토리지 (GB) ⓘ ▾	EBS 최적화 사용 가 능 ⓘ ▾	네트워크 성능 ⓘ ▾	IPv6 지 원 ⓘ ▾
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS 전용	-	낮음에서 중간	예
<input checked="" type="checkbox"/>	General purpose	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS 전용	-	낮음에서 중간	예
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS 전용	-	중간	예
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS 전용	-	중간	예
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS 전용	예	중간	예

[취소](#) [이전](#) [검토 및 시작](#) 다음: 인스턴스 세부 정보 구성

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ ★

🔔 junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 3: 인스턴스 세부 정보 구성

요구 사항에 적합하게 인스턴스를 구성합니다. 동일한 AMI의 여러 인스턴스를 시작하고 스팟 인스턴스를 요청하여 보다 저렴한 요금을 활용하며 인스턴스에 액세스 관리 역할을 할당하는 등 다양한 기능을 사용할 수 있습니다.

인스턴스 개수 ⓘ

1

Auto Scaling 그룹 시작 ⓘ

구매 옵션 ⓘ

☐ 스팟 인스턴스 요청

네트워크 ⓘ

vpc-4f348327 (기본값) ▾

🔄 새 VPC 생성

서브넷 ⓘ

기본 설정 없음(가용 영역의 기본 서브넷) ▾

🔄 새 서브넷 생성

퍼블릭 IP 자동 할당 ⓘ

서브넷 사용 설정(활성화) ▾

IAM 역할 ⓘ

없음 ▾

🔄 새 IAM 역할 생성

종료 방식 ⓘ

중지 ▾

종료 방지 기능 활성화 ⓘ

☐ 의도하지 않은 종료로부터 보호

모니터링 ⓘ

☐ CloudWatch 세부 모니터링 활성화
추가 요금이 적용됩니다.

테넌시 ⓘ

공유됨 - 공유된 하드웨어 인스턴스 실행 ▾
전용 테넌시에는 추가 요금이 적용됩니다.

CPU 사용률 ⓘ

☐ T2 무제한
추가 요금이 적용될 수 있습니다

취소 이전 검토 및 시작 다음: 스토리지 추가

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ 

🔔 junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 4: 스토리지 추가

인스턴스가 다음 스토리지 디바이스 설정으로 시작됩니다. 추가 EBS 볼륨 및 인스턴스 스토어 볼륨을 인스턴스에 연결하거나 루트 볼륨의 설정을 편집할 수 있습니다. 인스턴스를 시작한 후 추가 EBS 볼륨을 연결할 수도 있지만, 인스턴스 스토어 볼륨은 연결할 수 없습니다. Amazon EC2의 스토리지 옵션에 대해 [자세히 알아보십시오](#).

볼륨 유형 ⓘ	디바이스 ⓘ	스냅샷 ⓘ	크기(GiB) ⓘ	볼륨 유형 ⓘ	IOPS ⓘ	처리량(MB/초) ⓘ	종료 시 삭제 ⓘ	암호화 ⓘ
루트	/dev/sda1	snap-034bb11cedbafbc41	10	범용 SSD(GP2) ▾	100/3000	해당 사항 없음	<input checked="" type="checkbox"/>	암호화되지 않음

새 볼륨 추가

프리 티어 사용 가능 고객은 최대 30GB의 EBS 범용(SSD) 또는 마그네틱 스토리지를 사용할 수 있습니다. 프리 티어 자격 및 사용량 제한에 대해 [자세히 알아보기](#)

취소 이전 검토 및 시작 다음: 태그 추가

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾   junseonKim ▾ 서울 ▾ 

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 5: 태그 추가

태그는 대소문자를 구별하는 키-값 페어로 이루어져 있습니다. 예를 들어 키가 Name이고 값이 Webserver인 태그를 정의할 수 있습니다. 태그 복사본은 볼륨, 인스턴스 또는 둘 다에 적용될 수 있습니다. 태그는 모든 인스턴스 및 볼륨에 적용됩니다. Amazon EC2 리소스 태그 지정에 대해 [자세히 알아보기](#)

키 (최대 127자)	값 (최대 255자)	인스턴스 	볼륨 	
name	ZZUN Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
관리자	김준선	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
type	real server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

다른 태그 추가

(최대 50개 태그)

취소

이전

검토 및 시작

다음: 보안 그룹 구성

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ ★

🔔 junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 [자세히 알아보기](#)

보안 그룹 할당: ☒ 새 보안 그룹 생성
☐ 기존 보안 그룹 선택

보안 그룹 이름:

설명:


유형 ⓘ	프로토콜 ⓘ	포트 범위 ⓘ	소스 ⓘ	설명 ⓘ	
SSH ▾	TCP	22	내 IP ▾	61.73.113.208/32	예: 관리자 데스크톱용 SSH ✕
HTTP ▾	TCP	80	위치 무관 ▾	0.0.0.0/0, ::/0	예: 관리자 데스크톱용 SSH ✕

규칙 추가

인스턴스에 접근하는 방식
원격 제어 (ssh) / 웹서버로 사용 (http) / 파일업로드

취소 이전 검토 및 시작

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾  junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토






단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 [자세히 알아보기](#)

보안 그룹 할당: ☒ 새 보안 그룹 생성 ☐ 기존 보안 그룹 선택

보안 그룹 이름:

설명:

유형 	프로토콜 	포트 범위 	소스 	설명 
SSH ▾	TCP	22	<div>내 IP ▾ 61.73.113.208/32</div>	예: 관리자 데스크톱용 SSH ✕
HTTP ▾	TCP	80	<div>위치 무관 ▾ 0.0.0.0/0, ::/0</div>	예: 관리자 데스크톱용 SSH ✕

규칙 추가

Source

Anywhere(위치 무관) : 모든 접근 허용 (http)

My IP : 현재 웹 페이지의 나의 컴퓨터만


Custom IP (특정 컴퓨터) : 특정한 컴퓨터만

취소

이전

검토 및 시작

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ ★

junseonKim ▾ 서울 ▾ 지원 ▾

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 [자세히 알아보기](#)

보안 그룹 할당: ☐ 새 보안 그룹 생성 ☒ 기존 보안 그룹 선택

보안 그룹 ID	이름	설명	작업
<input type="checkbox"/> sg-5757fe3c	default	default VPC security group	새로 복사
<input type="checkbox"/> sg-d6fd56bd	rds-launch-wizard	Created from the RDS Management Console	새로 복사

위에서 인바운드 규칙을 확인할 보안 그룹을 선택합니다.

취소 이전 검토 및 시작

2. [실습] EC2 이용해서 서버구축하기!

기존 키 페어 선택 또는 새 키 페어 생성

키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 포함되어 있지 않다면, 키 페어를 제거해야 합니다. [키 페어 제거](#)에 대해 자세히 알아보십시오.

기존 키 페어 선택

키 페어를 선택하십시오

키 페어 없음



키 페어 없음

키 페어가 없습니다. 계속하려면 위에서 [새 키 페어 생성] 버튼을 클릭하여 새 키 페어를 작성하십시오.

기존 키 페어 선택 또는 새 키 페어 생성



키 페어는 AWS에 저장하는 퍼블릭 키와 사용자가 저장하는 프라이빗 키 파일로 구성됩니다. 이 둘을 모두 사용하여 SSH를 통해 인스턴스에 안전하게 접속할 수 있습니다. Windows AMI의 경우 인스턴스에 로그인하는 데 사용되는 암호를 얻으려면 프라이빗 키 파일이 필요합니다. Linux AMI의 경우, 프라이빗 키 파일을 사용하면 인스턴스에 안전하게 SSH로 연결할 수 있습니다.

참고: 선택한 키 페어가 이 인스턴스에 대해 승인된 키 세트에 포함되어 있지 않다면, 키 페어를 제거해야 합니다. [키 페어 제거](#)에 대해 자세히 알아보십시오.

새 키 페어 생성

키 페어 이름

zzunvely

키 페어 다운로드



계속하려면 먼저 프라이빗 키 파일(*.pem 파일)을 다운로드해야 합니다. 액세스할 수 있는 안전한 위치에 저장합니다. 파일은 생성되고 나면 다시 다운로드할 수 없습니다.



취소


인스턴스 시작

취소

인스턴스 시작

2. [실습] EC2 이용해서 서버구축하기!

 서비스 ▾ 리소스 그룹 ▾ 

 junseonKim ▾ 서울 ▾ 지원 ▾

 **지금 인스턴스를 시작 중입니다.**
다음 인스턴스 시작 개시: [i-044d3752af6e6cba4](#) [시작 로그 보기](#)

 **예상 요금 알림 받기**
[결제 알림 생성](#) AWS 결제 예상 요금이 사용자가 정의한 금액을 초과하는 경우(예를 들면 프리 티어를 초과하는 경우) 이메일 알림을 받습니다.

- [Linux 인스턴스에 연결하는 방법](#)
- [AWS 프리 티어에 대해 알아보기](#)

- [Amazon EC2: 사용 설명서](#)
- [Amazon EC2: 토론 포럼](#)

2. [실습] EC2 이용해서 서버구축하기!

인스턴스에 연결

다음에 연결: ☒ 독립 실행형 SSH 클라이언트
☐ 현재 웹 브라우저에서 Java SSH 클라이언트에 직접(Java 필요)

인스턴스 액세스 방법:

1. SSH 클라이언트를 개방하십시오. (PuTTY를 사용하여 연결 방법 알아보기)
2. 프라이빗 키 파일(zzunvely.pem)을 찾습니다. 마법사가 인스턴스를 시작하는 데 사용되는 키를 자동으로 검색합니다.
3. SSH가 작동하려면 키가 공개적으로 표시되지 않아야 합니다. 필요할 경우 이 명령을 사용합니다.

```
chmod 400 zzunvely.pem
```
4. 퍼블릭 DNS을(를) 사용하여 인스턴스에 연결:

```
ec2-13-124-246-163.ap-northeast-2.compute.amazonaws.com
```

예:

```
ssh -i "zzunvely.pem" ubuntu@ec2-13-124-246-163.ap-northeast-2.compute.amazonaws.com
```

대부분의 경우 위의 사용자 이름이 맞지만, AMI 사용 지침을 숙지하여 AMI 소유자가 기본 AMI 사용자 이름을 변경하지 않도록 하십시오.

인스턴스에 연결하는 데 도움이 필요한 경우 [연결 설명서](#)을(를) 참조하십시오.

닫기

SSH(Secure Shell)

- Unix계열의 원격제어 형식
- 공개 key 형태의 암호화방식

Mac의 경우
터미널을 여세요!

2. [실습] EC2 이용해서 서버구축하기!

Mac 사용자

1. 터미널파일 실행 후
2. `chmod 400 /pem파일경로/[파일명].pem`
3. `ssh -i /pem파일경로/[파일명].pem ec2-user@인스턴스dns주소`

(예시)

터미널파일 실행 후

```
chmod 400 zzunvely.pem
```

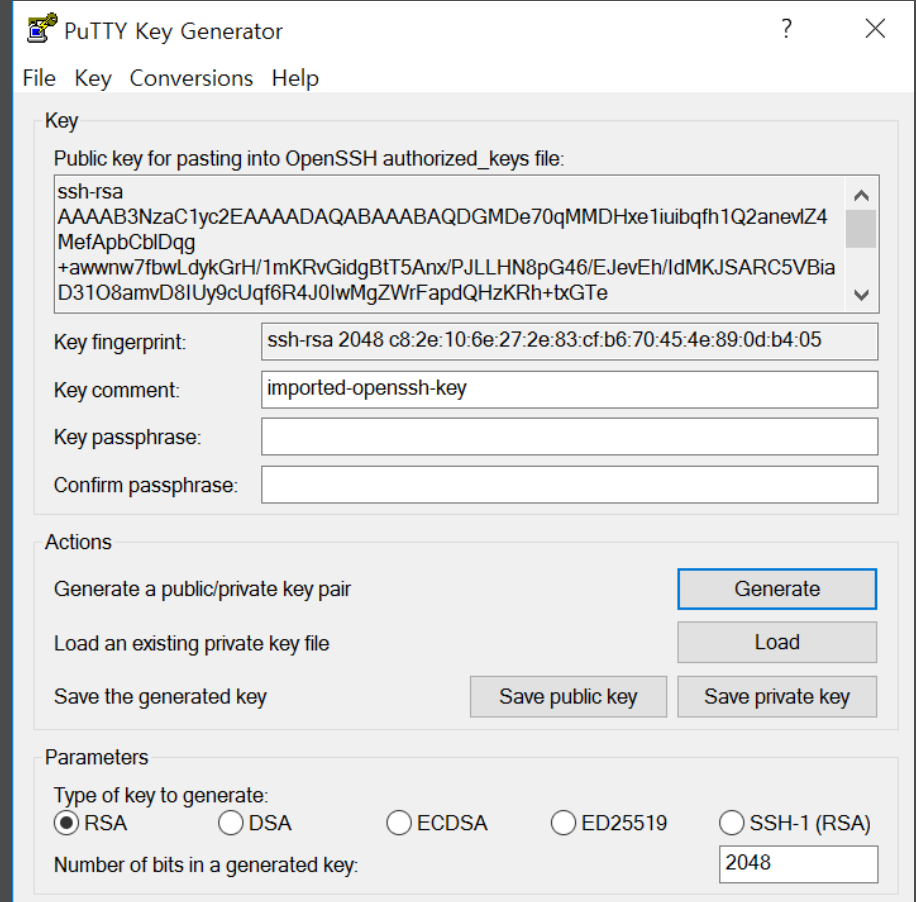
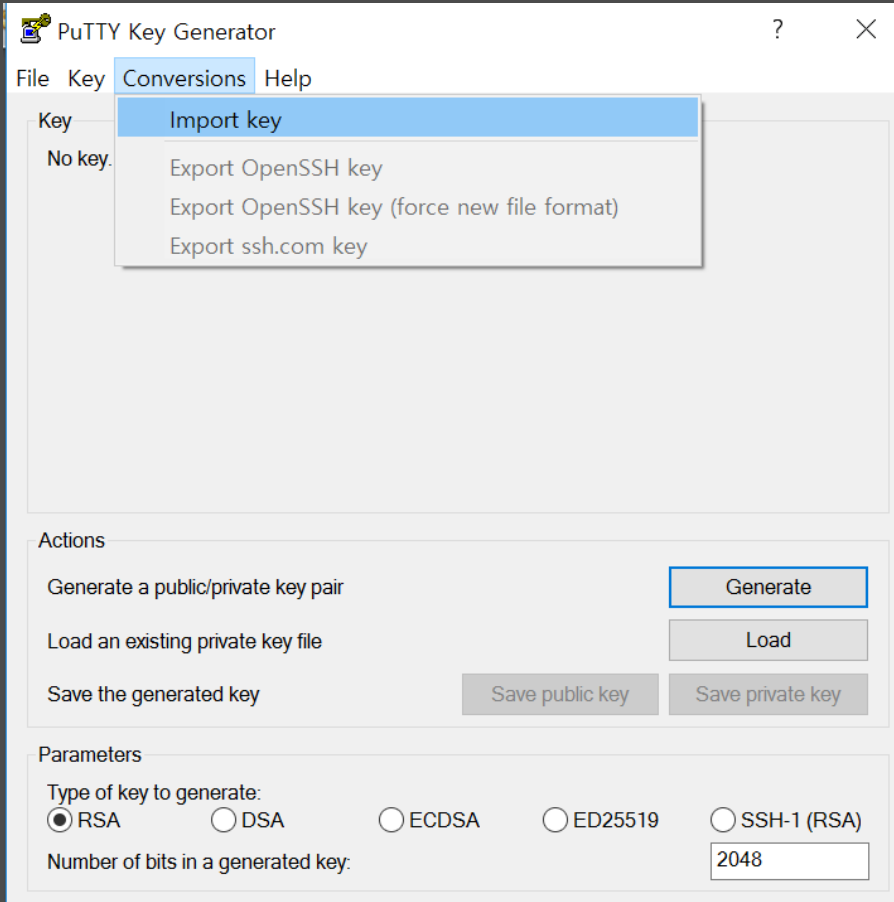
```
ssh -i "zzunvely.pem" ubuntu@ec2-13-124-246-163.ap-northeast2.compute.amazonaws.com
```


2. [실습] EC2 이용해서 서버구축하기!

Window 사용자

1. puttygem 을 실행
2. 메뉴의 conversions 에서 import key선택
3. 파일형식을 '.Allfiles' 로 변환하여 방금 전에 다운로드한 키페어파일(pem)을 선택
4. Private key를 눌러서 ppk파일 생성

2. [실습] EC2 이용해서 서버구축하기!



2. [실습] EC2 이용해서 서버구축하기!

Putty를 실행시킵니다

AWS페이지에서 본인의 인스턴스를 선택하고,
설명 부분에있는 인스턴스의 Public DNS주소를 확인

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, a table lists instances with columns for Name, ID, Type, Region, State, Health, and Monitoring. The instance 'i-044d3752af6e6cba4' is shown in the 'running' state. Below the table, the '설명' (Description) tab is selected, showing details for the instance. The '퍼블릭 DNS(IPv4)' (Public DNS (IPv4)) is highlighted with an orange box, indicating the address to use for connecting to the instance via Putty.

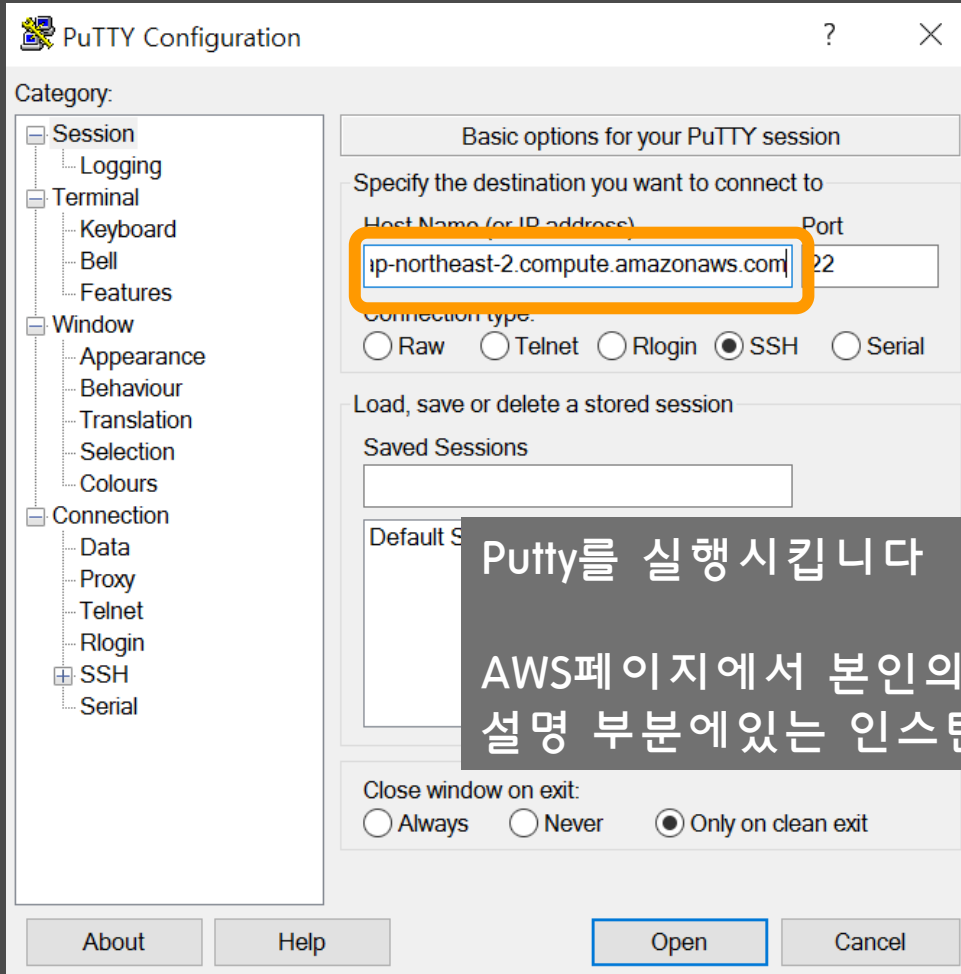
Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사	경보 상태
	i-044d3752af6e6cba4	t2.micro	ap-northeast-2a	running	2/2 검사 통과	없음

인스턴스: i-044d3752af6e6cba4 퍼블릭 DNS: ec2-13-124-246-163.ap-northeast-2.compute.amazonaws.com

설명 상태 검사 모니터링 태그

인스턴스 ID	i-044d3752af6e6cba4	퍼블릭 DNS(IPv4)	ec2-13-124-246-163.ap-northeast-2.compute.amazonaws.com
인스턴스 상태	running	IPv4 퍼블릭 IP	13.124.246.163
인스턴스 유형	t2.micro	IPv6 IP	-
탄력적 IP		프라이빗 DNS	ip-172-31-2-150.ap-northeast-2.compute.internal

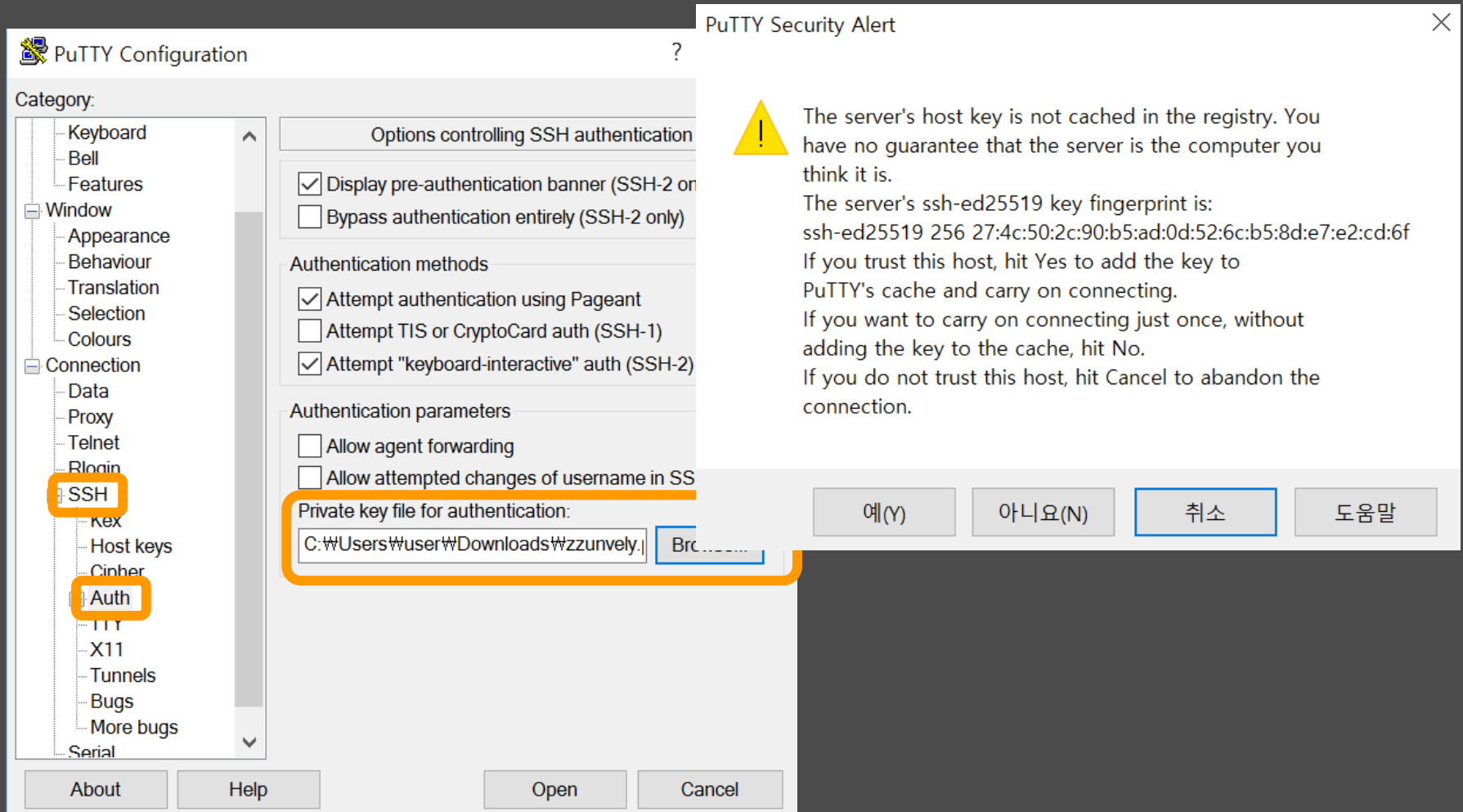
2. [실습] EC2 이용해서 서버구축하기!




PuTTY를 실행시킵니다

AWS페이지에서 본인의 인스턴스를 선택하고,
설명 부분에있는 인스턴스의 Public DNS주소를 붙여주기

2. [실습] EC2 이용해서 서버구축하기!



2. [실습] EC2 이용해서 서버구축하기!

 ec2-13-124-246-163.ap-northeast-2.compute.amazonaws.com - PuTTY

login as: ubuntu

```
ubuntu@ip-172-31-2-150: ~  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-1041-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-2-150:~$
```

내 서버에
들어오기 성공!