

# 빅데이터를 지탱하는 기술 Week 4 AWS & GCP

23.08.16 / 엄소은

# CONTENTS

01. Cloud Service

02. AWS

03. GCP

- Server
- Cloud Service

- AWS 기능
- AWS EC2

- VM Instance

# 1. Cloud Service

#### Server

- 서버(Server) = 어떤 역할을 수행해주는 컴퓨터 (GPU 연산을 대신 해주기 위해 고급 GPU가 장착된 컴퓨터, 웹페이지에 대한 정보를 보내주는 컴퓨터 etc)
- 서비스를 요청하는 사용자는 client, 요청 받은 작업을 해주는 것이 server



# 1. Cloud Service

#### Server 접속하는 방법 : SSH Protocol

- Protocol : 컴퓨터끼리 사용하는 통신규칙
- HTTP: 인터넷에서 웹 서버와 인터넷 브라우저 사이에 문서를 전송하는데 사용하는 프로토콜
- SSH(Secure SHell): 멀리 떨어진(원격) 컴퓨터(서버)를 안전하게 제어하기 위한 프로토콜(컴퓨터 사이 통신 규칙) 또는 이 프로토콜을 사용하는 프로그램
- SSH client 와 SSH server 의 관계로 상호작용하며 SSH server 가 설치된 운영체제를 제어한다.
- 중간에서 데이터를 가로채도 해석하지 못하게 강력하게 암호화된 방법을 통해 연결되어 있다.



# 1. Cloud Service

#### **Cloud Service?**

- 클라우드 서비스란, 인터넷을 통해 컴퓨팅 자원, 데이터 저장, 소프트웨어, 플랫폼 및 기타 IT 관련 서비스를 원격으로 제공하는 것
- 클라우드 서비스는 필요한 리소스(하드웨어, 소프트웨어, 데이터 저장소 등)을 필요한 만큼 요청하고 제공받는 <mark>온디맨드(</mark>ondemand) 방식으로 작동
- 사용자는 필요한 시점에 적절한 양의 자원을 신속하게 할당받거나 반환할 수 있음
- 사용자는 자원의 유연성과 확장성을 활용하여 비용을 절감



온프레미스(On-premise)



온디맨드(On-demand)

# 1. Cloud Service

#### Cloud Service 의 장점

- 서버실을 직접 물리적으로 세팅할 필요 X!
- 데이터 센터 어딘가에 이미 준비된 서버 사용 (Amazon, Google 과 같은 대기업들이 이미 구축해놓은 거대한 서버실)
- 서버 세팅 등을 신경쓰지 않고 서비스 운영에만 집중 가능
- 사용한 만큼 비용을 지불하여 서비스 운영에 효율성이 높아짐 (분단위 , 시간단위 과금)



# 1. Cloud Service

#### Cloud Service 의 유형

- laaS (Infrastructure as a Service)
  - 사용자가 관리할 수 있는 범위가 가장 넓은 클라우드 컴퓨팅 서비스 = 서버에서 제공하는 범위가 가장 작다
  - 사용자가 서버 OS, 미들웨어, 런타임, 데이터, 어플리케이션까지 직접 구성하고 관리 가능
  - AWS의 EC2 와 Google 의 Compute Engine
- PaaS (Platform as a Service)
  - laaS 형태의 가상화된 클라우드 위에 사용자가 원하는 서비스를 개발할 수 있도록 개발 환경 (Platform)을 미리 구축
  - laaS 보다는 관리상의 자유도가 낮다
  - Salseforce의 Heroku , Redhat의 Openshift
- SaaS (Software as a Service)
  - 바로 사용할 수 있는 소프트웨어 자체를 제공하는 것
  - 사용자는 구독료를 지불하고 소프트웨어를 이용
  - Slack, Microsoft 365, Dropbox

# 1. Cloud Service

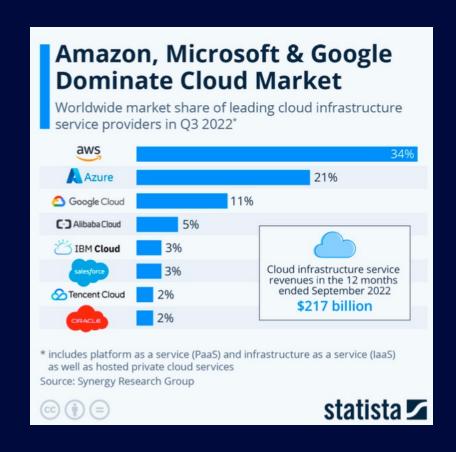
#### Cloud Service 의 유형



# 1. Cloud Service

#### laaS Service 는 어떤것을 써야 할까?

- Top 3: AWS, Azure, Google Cloud
- 각각의 장단점이 있다
- AWS
  - 가입 후 1년 Free tier 제공
  - 시간당 과금
  - 개별 서비스끼리 연동하여 프로젝트 가능
  - 대용량 데이터 처리에 유리
- GCP
  - 평생 Free tier 제공 (제한점 많음)
  - 분당 과금
  - 오픈소스와의 협업에 유리
  - TPU 라는 AI 칩 사용



# 1. Cloud Service

#### laaS Service 는 어떤것을 써야 할까?

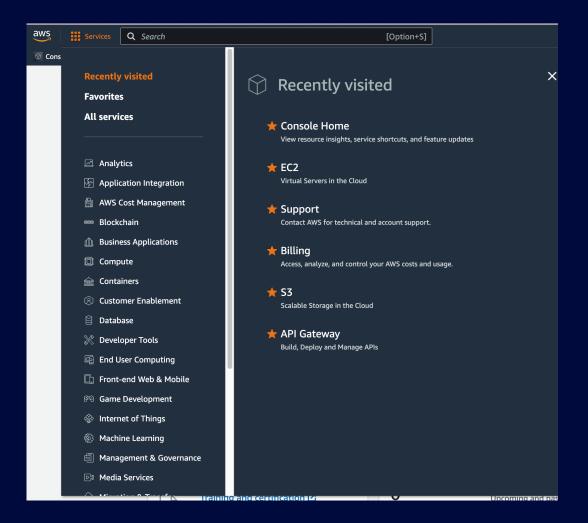
- 운영하고자 하는 서비스에 맞게 적절한 클라우드 사업자를 선택해야 함
- 실제로는 여러 클라우드 컴퓨팅 업체를 이용하는 경우도 많음



# YONSEI DATA SCIENCE LAB | DSL

#### **AWS Service**

- AWS에 들어가면 service 가 무지하게 많다
- 그 중에 우리가 가장 많이 사용할 것은
  - -> Console Home, EC2, S3, API Gateway, Support, Billing

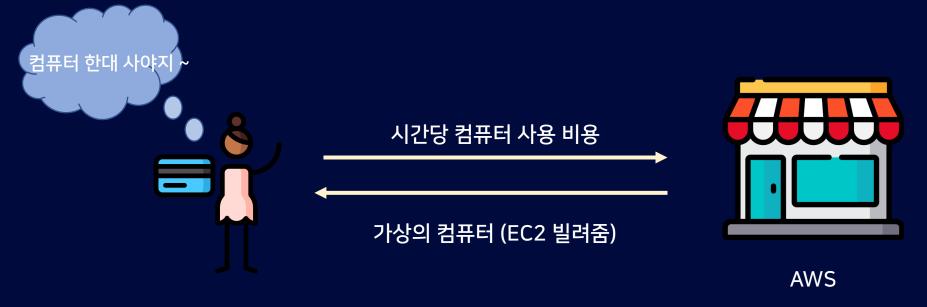


# **2. AWS**

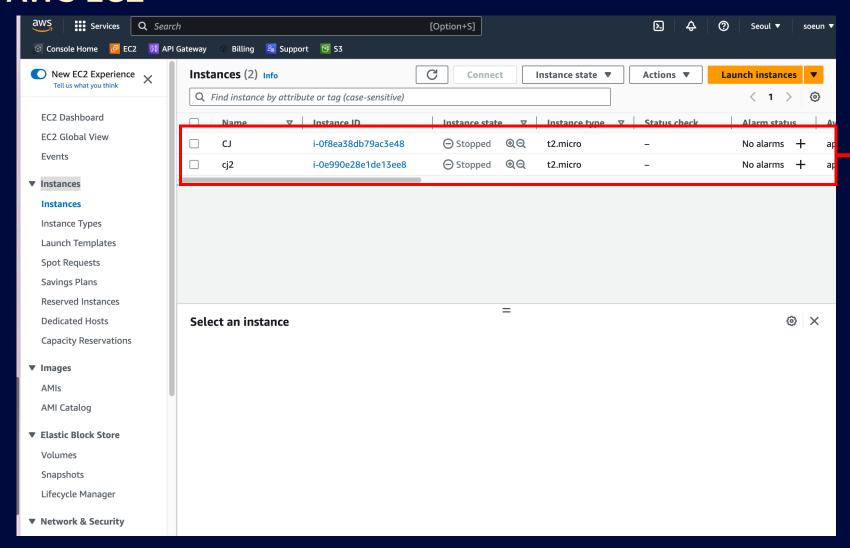
#### **AWS EC2**

- EC2 : Elastic Compute Cloud = 독립된 가상의 컴퓨터를 임대해주는 웹 서비스
- 물리적인 컴퓨터가 아닌, 리눅스 , 윈도우 등 OS 가 설치된 가상 머신
- 사용자는 원격으로 해당 컴퓨터를 제어하여 사용할 수 있다
- 사용목적에 맞게 OS, 메모리, GPU, 스토리지 등 인스턴스 스펙을 선택할 수 있다



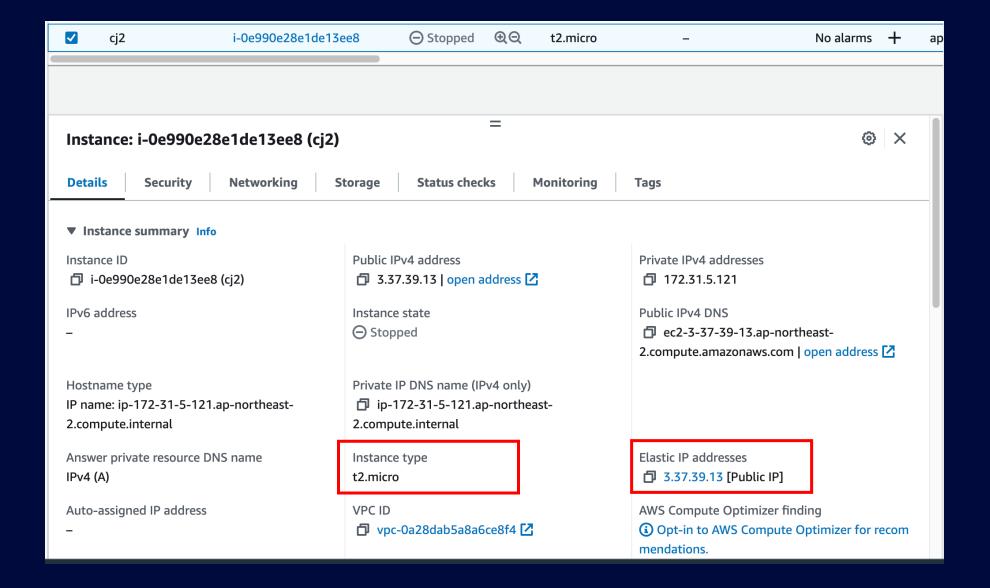


#### **AWS EC2**



내가 만든 Instance (=가상의 컴퓨터)





### EC2 만들기

- 1. Launch an instance
- 2. Instance 이름 짓기
- 3. OS, Amazon Machine Image 고르기
- 4. Instance type (몇개 vCPU 사용할 것인가, Memory 는 얼마 쓸 것인가) t2.micro 가 free tier 사용 가능
- 5. Key pair (Create new key pair key pair 이름 입력 후 key pair type RSA, key file format .pem) 후 Create key pair 클릭
- 6. Network setting : 내 가상 컴퓨터에 들어오는 접속 허용 정도 결정 (방화벽)
- 7. Storage : 스토리지 결정
- 8. Launch instance

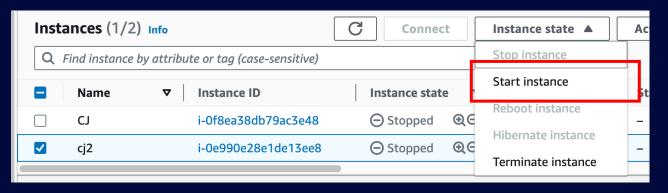


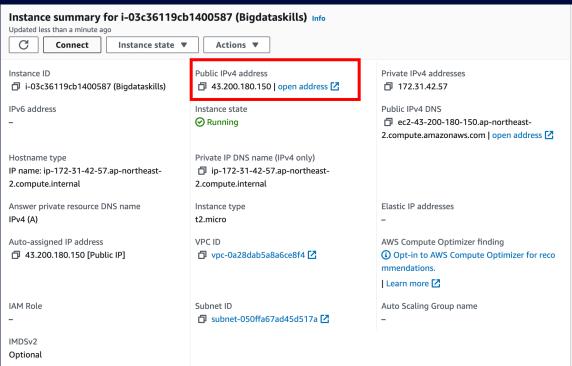
#### EC2 Instance 스펙

- EC2 Instance 타입
  - 범용(M), 마이크로(T), 컴퓨팅 최적화(C), 스토리지 최적화(I,D), 메모리 최적화(X,R), 가속화된 컴퓨팅(G,P,F)
- 성능
  - nano < micro < small < medium < large < xlarge < 2xlarge nano < 4xlarge ~</li>

# YONSEI DATA SCIENCE LAB | DSL

### EC2 만들기





Public IPv4 address 는 Instance 를 실행할 때마다 바뀐다!

#### EC2 Instance 시작하기

1. 내가 pem key 를 다운받은 디렉토리로 이동

```
~/yonsei/DSL/DSL_23-1/BigDataSkills/week4 main*
~/y/D/DSL_/BigDataSkills/week4 main !4 ?5 ls
bds.pem
```

2. chmod 600 bds.pem

```
~/y/8/DSL_/BigDataSkills/week4 | main !4 ?5 | chmod 600 bds.pem
```

3. ssh -i [pem key] [username]@[public IPv4 DNS 주소]

```
~/v/0/050 /BigDataSkills/week4 > main !4 ?5 > ssh −i bds.pem ubuntu@ec2−15−165−156−212.ap−northeast−2.c
ompute.amazonaws.com
```

- 'ec2-user' (for Amazon Linux)
- 'ubuntu' (for Ubuntu)
- admin (for Debian)
- centos (for CentOS)

### EC2 Instance 내 설정하기

ubuntu@ip-172-31-42-57:~\$

- 1. sudo apt-get update
- 2. sudo apt install python3
- 3. sudo apt install python3-pip → pip 를 통해 내가 원하는 라이브러리들을 Instance 내에 설치할 수 있다 !(transformers, torch..)

#### EC2 Instance 내 설정하기

• 내가 실행하고자 하는 파이썬 파일들을 EC2 내에서 실행하는 방법 → git clone OR Docker container run

```
ubuntu@ip-172-31-42-57:~$ git clone https://github.com/ddoddii/translate_juso.git
Cloning into 'translate_juso'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 66 (delta 33), reused 50 (delta 17), pack-reused 0
Receiving objects: 100% (66/66), 1.78 MiB | 16.46 MiB/s, done.
Resolving deltas: 100% (33/33), done.
```

```
ubuntu@ip-172-31-42-57:~$ ls
translate_juso
```

```
ubuntu@ip-172-31-43-247:~/translate_juso$ ls
api.py juso_api_eng.py juso_api_kor.py main.py model re_check.py requirements.txt
ubuntu@ip-172-31-43-247: /translate.jusof
```

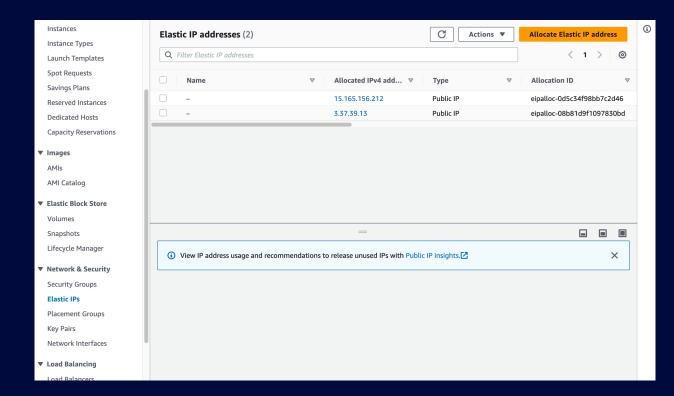


### EC2 Instance 내 설정하기

- conda 설치로 가상환경을 만들 수도 있다
  - 1. sudo wget <a href="https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Linux-x86\_64.sh">https://repo.anaconda.com/archive/Anaconda3-2023.07-2-Linux-x86\_64.sh</a>
  - 2. bash Anaconda3-2023.07-2-Linux-x86\_64.sh
  - 3. Install 동의 (yes) 입력

#### **Elastic IP**

- EC2 Instance 를 중지하고 새롭게 시작하면, 기존에 할당받은 IP 주소가 변경된다..
- 실제 서비스 중인 서버라면 ?? 대참사 (마치 자고일어나면 <u>www.naver.com</u> → <u>www.naaavver.com</u> 이 되는 상황)
- 따라서 내 Instance 에 고정 IP 주소를 생성하는 것 = Elastic IP

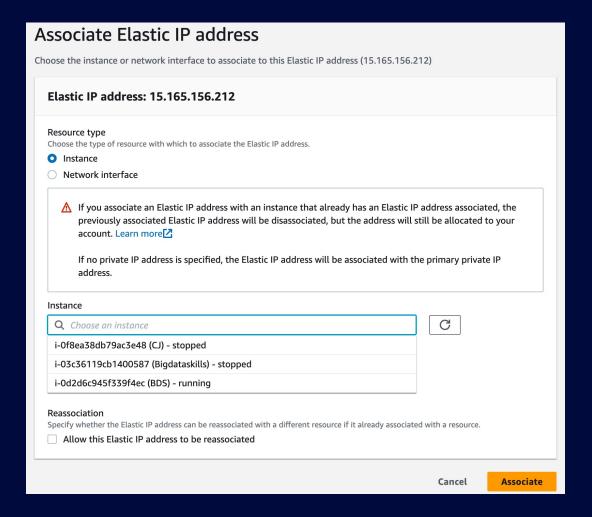


Network & Security > Elastic IPs

### Elastic IP

- 1. Allocate IP Address
- 2. Associate Elastic IP address
  - → 내가 만든 Instance 에 할당할 수 있다

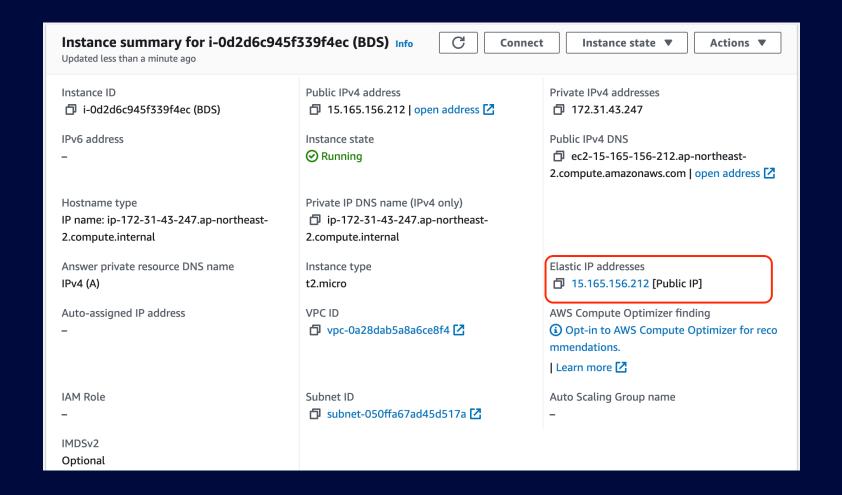




# **2. AWS**

#### **Elastic IP**

• 내 Instance 의 고정 IP 가 할당 되었다!



# 3. GCP

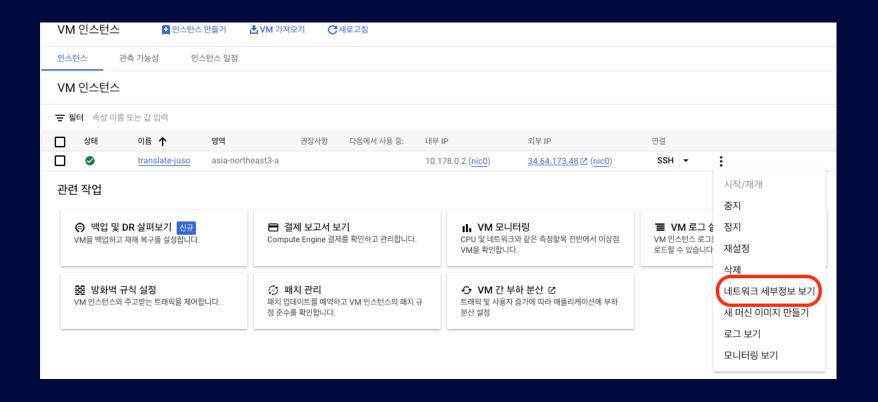
#### GCP 에서 VM Instance 만들기



- AWS 에서 EC2 Instance = GCP 에서 VM Instance
- GCP도 인스턴스를 중지하고 시작할 때마다 외부IP 가 바뀐다

# 3. GCP

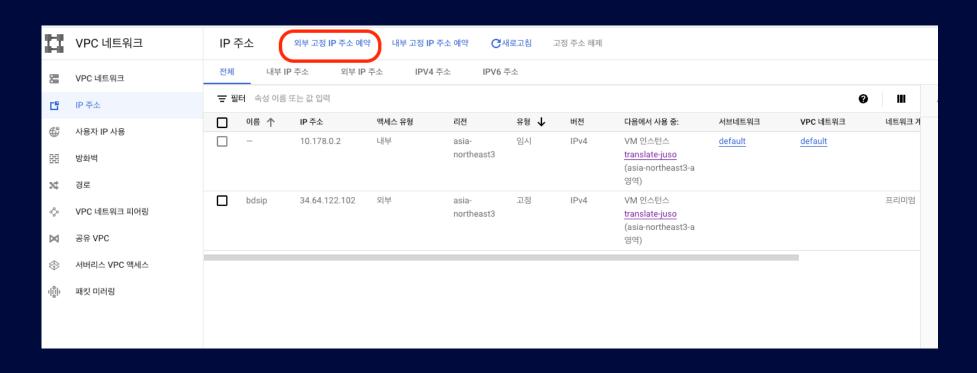
### GCP 에서 고정 IP 주소 만들기





# 3. GCP

# GCP 에서 고정 IP 주소 만들기



# 4. Reference

- https://cloud.google.com/docs/geography-and-regions?hl=ko
- https://docs.aws.amazon.com/ec2/index.html?nc2=h\_ql\_doc\_ec2