

# 2022 A Basic CS skill: ABC Winter School

## Special Topic 2

Team 8

2022 / 02 / 23



ULSAN NATIONAL INSTITUTE OF  
SCIENCE AND TECHNOLOGY

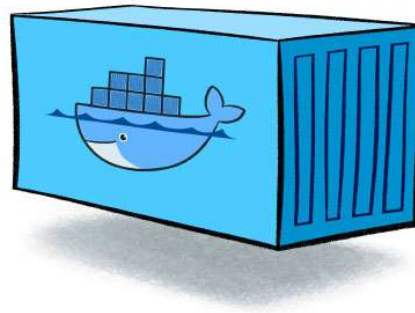
# Docker

---

- Container를 이용해 각종 응용 프로그램을 빠르게 구축, 테스트, 배포할 수 있도록 도와주는 SW 플랫폼 입니다.
- Linux 환경에서 Container를 생성하고 사용할 수 있도록 도와줍니다.



응용 프로그램



도커 컨테이너



서버

# Docker 설치하기

- 우분투 환경에서 다음 명령어를 사용하면 docker가 설치 됩니다.
- `sudo snap install docker`
- `sudo apt-get install docker.io`
- 이 후 터미널에 docker를 입력했을 때, 다음과 같은 화면이 나타나면 docker 설치가 성공적으로 완료 된 것입니다.

```
ubuntu@ip-172-31-40-165:~$ docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/home/ubuntu/snap/docker/796/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var
                        and default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string   Trust certs signed only by this CA (default "/home/ubuntu/snap/docker/796/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/home/ubuntu/snap/docker/796/.docker/cert.pem")
  --tlskey string      Path to TLS key file (default "/home/ubuntu/snap/docker/796/.docker/key.pem")
  --tlsverify          Use TLS and verify the remote
  -v, --version        Print version information and quit

Management Commands:
  builder      Manage builds
  config       Manage Docker configs
  container    Manage containers
  context       Manage contexts
  engine        Manage the docker engine
  image        Manage images
  network      Manage networks
  node          Manage Swarm nodes
  plugin        Manage plugins
  secret        Manage Docker secrets
  service       Manage services
  stack         Manage Docker stacks
  swarm         Manage Swarm
  system        Manage Docker
  trust         Manage trust on Docker images
  volume        Manage volumes

Commands:
  attach        Attach local standard input, output, and error streams to a running container
  build          Build an image from a Dockerfile
  commit        Create a new image from a container's changes
  cp            Copy files/folders between a container and the local filesystem
```

# Docker 명령어

---

- Docker는 os 커널을 사용하기 때문에 반드시 sudo를 붙여서 사용해야 합니다.
  - docker ps
  - docker search [이미지 이름]
  - docker pull [이미지 이름]
  - docker images
  - docker run [이미지 이름]
  - docker build [Dockerfile 경로]
  - docker rm [컨테이너 이름]
  - docker rmi [이미지 이름]
  - docker attach [컨테이너 이름]

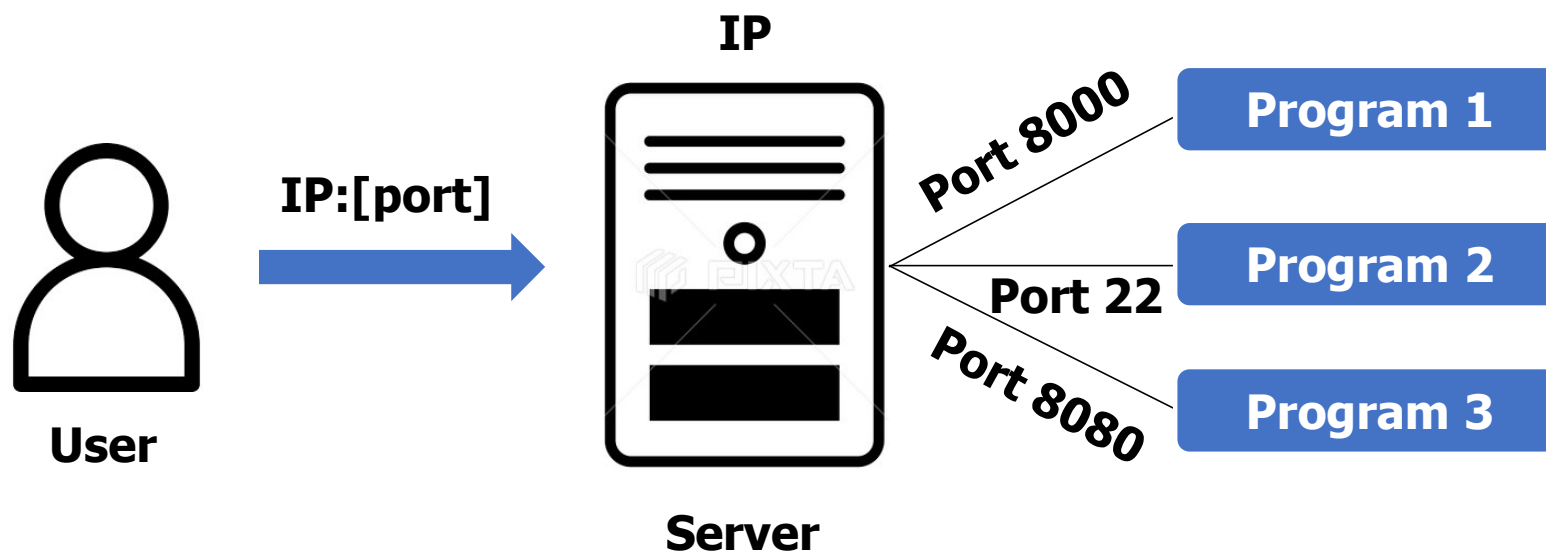
# Docker file 작성 방법

---

- Dockerfile은 아래와 같은 여러 명령어들을 이용해 작성합니다.
  - FROM : 컨테이너의 기반이 되는 이미지
  - MAINTAINER : 작성자 정보
  - RUN : shell script나 명령어를 실행
  - CMD : 컨테이너가 실행되었을 때 사용할 명령어
  - LABEL : 라벨 작성 (docker inspect 명령어로 확인 가능)
  - ENV : 컨테이너 내부에서 환경변수 설정
  - ADD : 파일 / 디렉토리 추가
  - COPY : 파일 복사
  - ENTRYPOINT : 컨테이너가 시작되었을 때 딱 한 번 실행할 스크립트 실행
  - VOLUME : docker volume 마운트
  - WORKDIR : RUN, CMD, ENTRYPOINT가 실행 될 작업 디렉토리

# Port

- 네트워크 통신 사이의 끝점을 의미합니다.
- SW적 의미
  - 네트워크 서비스나 특정 프로세스를 식별하는 논리적 통로
- Port number는 16 bit로 이루어져 있습니다. ( $0 \sim 2^{16}-1$  까지 존재)



# Port

---

- Well-known port (0~1023, 사용 금지)
  - 주로 시스템에서 사용 되는 포트

Well-Known Port	Application	Protocol
20	File Transfer Protocol (FTP) Data	TCP
21	File Transfer Protocol (FTP) Control	TCP
23	Telnet	TCP
25	Simple Mail Transfer Protocol (SMTP)	TCP
69	Trivial File Transfer Protocol (TFTP)	UDP
80	Hypertext Transfer Protocol (HTTP)	TCP
110	Post Office Protocol 3 (POP3)	TCP
194	Internet Relay Chat (IRC)	TCP
443	Secure HTTP (HTTPS)	TCP
520	Routing Information Protocol (RIP)	UDP

# Port

---

- Registered port (1024~49151, 조건에 따라 사용 가능)
  - 주로 user program level에서 사용

Registered Port	Application	Protocol
1812	RADIUS Authentication Protocol	UDP
1863	MSN Messenger	TCP
2000	Cisco Skinny Client Control Protocol (SCCP, used in VoIP applications)	UDP
5004	Real-Time Transport Protocol (RTP, a voice and video transport protocol)	UDP
5060	Session Initiation Protocol (SIP, used in VoIP applications)	UDP
8008	Alternate HTTP	TCP
8080	Alternate HTTP	TCP



# Django

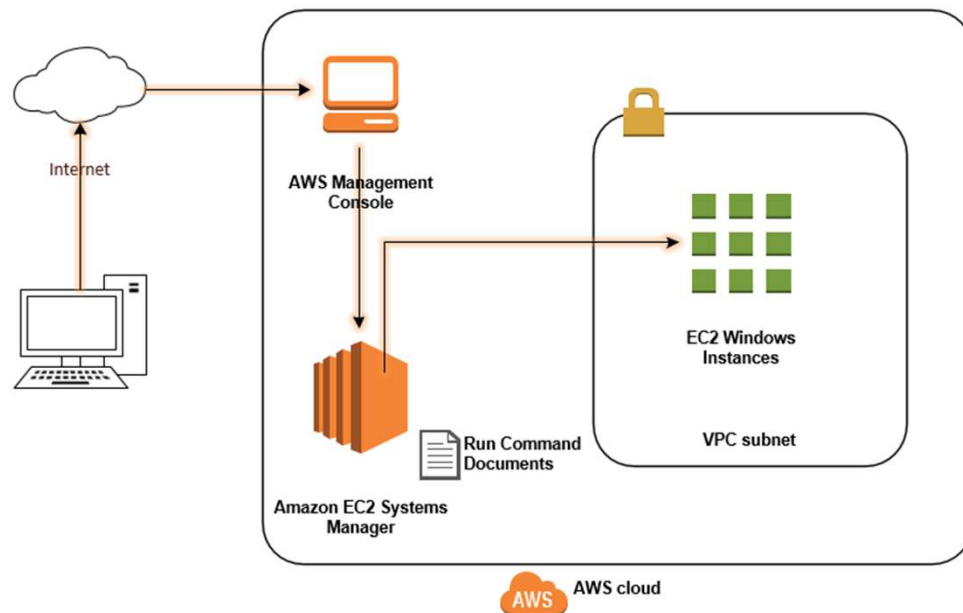
---

- Django는 웹 사이트 개발을 위한 Python 프레임 워크 입니다.
- 프론트엔드부터 백엔드까지 지원하며 프로그래머가 App 기능 개발에 좀 더 집중할 수 있도록 지원합니다.
- 인스타그램이 Django로 만들어졌다고 알려져 있습니다.



# AWS EC2

- IaaS의 대표적인 예시입니다.
- 사용자는 AWS로부터 “인스턴스”라고 불리는 가상의 서버를 할당 받아서 그 위에 사용자가 원하는 응용 프로그램을 실행할 수 있도록 합니다.
- 실행 중인 인스턴스의 사용량이나 실행 시간에 대해 비용을 지불해야 합니다.



## EC2

# AWS 가입

- <https://signin.aws.amazon.com>
- 위 링크에서 콘솔에 로그인 -> AWS 계정 새로 만들기 버튼을 통해 계정을 만들 수 있습니다.
- AWS를 이용하기 위해서는 해외 결제가 가능한 VISA나 Mastercard가 필요합니다.

12개월 프리 티어 액세스 포함  
AWS 계정

Amazon EC2, Amazon S3 및 Amazon DynamoDB 사용 포함  
제안 약관 전문은 [aws.amazon.com/free](https://aws.amazon.com/free) 참조

### AWS 계정 생성

이메일 주소

암호

암호 확인

AWS 계정 이름

전화번호

국가/리전

미국

주소

읍/면/동 우편번호, 회사 이름, 수신

아파트, 스위트룸, 가구, 빌딩, 층 등

구/군/시



시/도 또는 리전

우편 번호

# AWS 가입

- 그리고 반드시 마지막에 “기본 플랜”으로 설정을 해주셔야 불필요한 결제를 막을 수 있습니다.
- 이 후 등록한 신용카드 계좌가 유효한 계좌인지 확인을 하게 되면 가입이 완료 됩니다.

AWS는 귀하의 요구를 충족할 수 있는 선별된 지원 플랜을 제공합니다. 귀하의 AWS 사용량에 맞는 지원 플랜을 선택하십시오. [자세히 알아보기](#)

		
기본 플랜	개발자 플랜	비즈니스 플랜
무료	월 \$29부터	월 \$100부터

# AWS EC2 콘솔 대시보드

- AWS에서 제공하는 EC2를 웹 페이지를 통해 관리할 수 있는 서비스입니다.
- 루트 사용자 로그인 -> 전체 서비스에서 EC2 선택

## 로그인

☒ 루트 사용자  
무제한 액세스 권한이 필요한 작업을 수행하는 계  
정 소유자입니다. 자세히 알아보기

☐ IAM 사용자  
일일 작업을 수행하는 계정 내 사용자입니다. 자세  
히 알아보기

루트 사용자 이메일 주소

다음

## 전체 서비스

컴퓨팅  
**EC2**  
Lightsail  
Lambda  
Batch

# 키 페어 생성하기

- 본격적으로 EC2 인스턴스를 생성하기 전에, 키 페어가 필요합니다.
- 키 페어는 생성한 EC2 인스턴스 서버에 관리자로 접속할 수 있는 권한을 확인하는 용도로 사용됩니다.
- SSH의 private key – public key와 같습니다.
- AWS 대시보드 -> 리소스에서 키 페어 -> 키 페어 생성
- 접속하면 오른쪽 그림처럼 나옵니다.

키 페어 생성

**키 페어**  
프라이빗 키와 퍼블릭 키로 구성되는 키 페어는 인스턴스에 연결할 때 자격 증명을 증명하는 데 사용하는 보안 자격 증명 세트입니다.

이름  
  
이름에는 최대 255개의 ASCII 문자가 포함됩니다. 앞 또는 뒤에 공백을 포함할 수 없습니다.

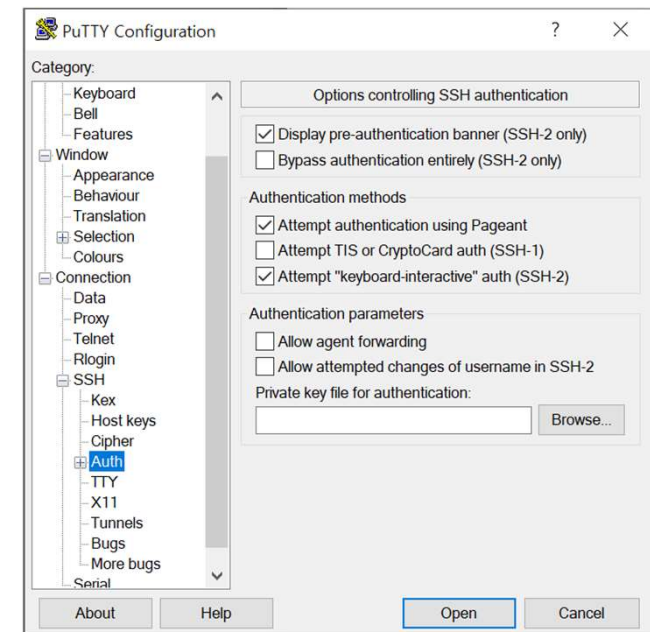
파일 형식  
☐ pem  
OpenSSH와 함께 사용  
☒ ppk  
PuTTY와 함께 사용

태그(선택 사항)  
리소스에 연결된 태그가 없습니다.

태그를 50개 더 추가할 수 있습니다.

# 키 페어 생성하기

- 키 페어 이름을 입력한 뒤, pem 형식과 ppk 형식 중에 선택해야 합니다.
- Pem 형식의 파일은 일반적인 SSH를 사용할 때, 다음과 같이 사용합니다.
  - `ssh -i [pem 파일 이름] [ID]@[인스턴스 ip 주소]`
- Ppk 형식의 파일은 putty를 통해 접속을 할 때 사용 됩니다.
  - Connection -> SSH -> Auth -> Browse
  - 이후 ppk 파일 선택
- 키 페어 생성을 누르면 자동으로 다운로드 됩니다.



# AWS EC2 인스턴스 생성하기

- EC2 대시보드 -> 인스턴스 시작 버튼 클릭
- 또는 왼쪽 메뉴 -> 인스턴스 -> 인스턴스 시작 버튼 클릭

<div></div> <div><b>Amazon Linux</b></div> <div>프리 티어 사용 가능</div>	<div><b>Amazon Linux 2 AMI (HVM), SSD Volume Type</b> - ami-09282971cf2faa4c9 (64비트 x86) / ami-094d838e7f8147d58 (64비트 Arm)</div> <div>Amazon Linux 2는 5년간 지원을 제공합니다. Amazon EC2에 성능 최적화된 Linux kernel 4.14와 systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, 최신 소프트웨어 패키지를 추가적으로 제공합니다.</div> <div>루트 디바이스 유형: ebs   가상화 유형: hvm   ENA 활성화: 예</div>	<div>선택</div> <div><input checked="" type="radio"/> 64비트(x86)</div> <div><input type="radio"/> 64비트(Arm)</div>
<div></div> <div><b>Ubuntu Server 20.04 LTS (HVM), SSD Volume Type</b> - ami-067abcae434ee508b (64비트 x86) / ami-0e59aecf3dbfcf4e4 (64비트 Arm)</div> <div>프리 티어 사용 가능</div>	<div>Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a>).</div> <div>루트 디바이스 유형: ebs   가상화 유형: hvm   ENA 활성화: 예</div>	<div>선택</div> <div><input checked="" type="radio"/> 64비트(x86)</div> <div><input type="radio"/> 64비트(Arm)</div>
<div></div> <div><b>Ubuntu Server 18.04 LTS (HVM), SSD Volume Type</b> - ami-0b50511490117e709 (64비트 x86) / ami-0ee0bf6766e2dbdca (64비트 Arm)</div> <div>프리 티어 사용 가능</div>	<div>Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a>).</div> <div>루트 디바이스 유형: ebs   가상화 유형: hvm   ENA 활성화: 예</div>	<div>선택</div> <div><input checked="" type="radio"/> 64비트(x86)</div> <div><input type="radio"/> 64비트(Arm)</div>
<div></div> <div><b>SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type</b> - ami-097fc5cd098dd20d5 (64비트 x86) / ami-06fb0a38fed3dcdec (64비트 Arm)</div> <div>프리 티어 사용 가능</div>	<div>SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.</div> <div>루트 디바이스 유형: ebs   가상화 유형: hvm   ENA 활성화: 예</div>	<div>선택</div> <div><input checked="" type="radio"/> 64비트(x86)</div> <div><input type="radio"/> 64비트(Arm)</div>
<div></div> <div><b>Red Hat Enterprise Linux 8 (HVM), SSD Volume Type</b> - ami-07270d166cdf39adc (64비트 x86) / ami-05de3e74a170005d0 (64비트 Arm)</div> <div>프리 티어 사용 가능</div>	<div>Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type</div> <div>루트 디바이스 유형: ebs   가상화 유형: hvm   ENA 활성화: 예</div>	<div>선택</div> <div><input checked="" type="radio"/> 64비트(x86)</div> <div><input type="radio"/> 64비트(Arm)</div>

**반드시 "프리 티어 사용 가능"이 적힌 OS를 선택해야 합니다.**



# AWS EC2 인스턴스 생성하기

- 이 후 2단계 인스턴스 유형 선택에서 “프리 티어 사용 가능”이 적힌 유형을 선택한 후, “보안 그룹 구성” 탭까지 다음 버튼을 눌러 줍니다.

## 단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 수 있는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다.

보안 그룹 할당: ☒ 새 보안 그룹 생성  
☐ 기존 보안 그룹 선택

보안 그룹 이름:   
설명:

- 보안 그룹 구성 탭에서 보안 그룹 이름과 설명을 바꾼 뒤, 검토 및 시작하기 버튼을 누르면 마지막 검토 페이지로 이동합니다.
- 검토 페이지에서 오른쪽 하단에 시작하기 버튼을 눌러 줍니다.

# AWS EC2 인스턴스 생성하기

- 키 페어 선택 창이 나오면 이전 슬라이드에서 생성한 키 페어를 선택합니다.
- 이후 체크박스를 체크 해준 뒤, 인스턴스 시작 버튼을 눌러 줍니다.

## 기존 키 페어 선택 또는 새 키 페어 생성



키 페어는 AWS에 저장하는 **퍼블릭 키**와 사용자가 저장하는 **프라이빗 키 파일**로 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 추가됩니다. 퍼블릭 AMI에서 기존 키 페어 제거에 대해 자세히 알아보십시오.

기존 키 페어 선택

키 페어를 선택하십시오

abc\_mentoring

☒ 선택한 프라이빗 키 파일(abc\_mentoring.pem)에 액세스할 수 있음을 확인합니다. 이 파일이 없으면 내 인스턴스에 로그인할 수 없습니다.

취소

인스턴스 시작

# AWS EC2 인스턴스 생성하기

- 인스턴스가 생성이 되면 EC2 대시보드에 인스턴스 메뉴에서 생성된 인스턴스를 볼 수 있습니다.

The screenshot displays the AWS Management Console interface for EC2 instances. At the top, there's a header with '인스턴스 (1/1) 정보' and buttons for '연결', '인스턴스 상태', '작업', and '인스턴스 시작'. Below this is a search bar and a table listing instances. The table has columns for Name, ID, Status, Type, State checks, Alarms, Region, Public IPv4 DNS, and Public IPv4 address. One instance is listed with ID 'i-0ebec45d88bf71df', status '실행 중' (Running), type 't2.micro', and region 'ap-northeast-2'. Below the table, the details for the selected instance are shown, including tabs for '세부 정보', '보안', '네트워킹', '스토리지', '상태 검사', '모니터링', and '태그'. The '세부 정보' tab is active, showing a summary of the instance's configuration, such as its ID, status, type, and various network and security settings.

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...
-	i-0ebec45d88bf71df	실행 중	t2.micro	초기화	경보 없음	ap-northeast-2c	ec2-52-79-234-227.ap...	52.79.234.227

**인스턴스: i-0ebec45d88bf71df**

세부 정보 | 보안 | 네트워킹 | 스토리지 | 상태 검사 | 모니터링 | 태그

▼ 인스턴스 요약 정보

인스턴스 ID i-0ebec45d88bf71df	퍼블릭 IPv4 주소 52.79.234.227   <a href="#">개방 주소법</a>	프라이빗 IPv4 주소 172.31.40.165
인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-52-79-234-227.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a>	프라이빗 IPv4 DNS ip-172-31-40-165.ap-northeast-2.compute.internal
인스턴스 유형 t2.micro	탄력적 IP 주소 -	VPC ID vpc-0c1aac67
AWS Compute Optimizer 찾기 권장 사항을 위해 AWS Compute Optimizer에 옵트인합니다.   <a href="#">자세히 알아보기</a>	IAM 역할 -	서브넷 ID subnet-ec190ea0

# AWS EC2 인스턴스 접속하기

---

- 인스턴스에 접속하기 위해서는 아래 3가지를 알아야 합니다.
  - 인스턴스에 접속할 IP
  - 인스턴스 계정 이름
  - pem 또는 ppk 파일
- 여기서 pem 파일이나 ppk 파일은 키 페어를 생성하면서 다운로드 받은 파일을 의미합니다.
- ubuntu 인스턴스를 만든 경우, 처음 인스턴스 접속을 위한 계정 이름은 ubuntu 라는 이름으로 초기화 되어 있습니다.

# AWS EC2 인스턴스 접속하기

- 인스턴스 접속에 필요한 인스턴스의 IP는 EC2 대시보드에 인스턴스 탭에서 확인할 수 있습니다.

인스턴스: i-0ebec45d88bff71df

세부 정보 | 보안 | 네트워킹 | 스토리지 | 상태 검사 | 모니터링 | 태그

▼ 인스턴스 요약 정보

인스턴스 ID	퍼블릭 IPv4 주소
i-0ebec45d88bff71df	52.79.234.227   <a href="#">개방 주소법</a>
인스턴스 상태	퍼블릭 IPv4 DNS
실행 중	ec2-52-79-234-227.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a>
인스턴스 유형	탄력적 IP 주소
t2.micro	-

- 퍼블릭 IPv4 주소에 IP 주소나 DNS 주소를 통해 접속할 수 있습니다.

# AWS EC2 인스턴스 접속하기

- Pem 파일을 받은 경우,
  - `ssh -i [pem 파일 경로] ubuntu@[EC2 퍼블릭 IP]`

```
with1@DESKTOP-AH4J9IJ MINGW64 ~/OneDrive/바탕 화면
$ ls
abc_metoring.pem  desktop.ini
ATgor1chm.1nk*   UNIST.1nk*

with1@DESKTOP-AH4J9IJ MINGW64 ~/OneDrive/바탕 화면
$ ssh -i abc_metoring.pem ubuntu@54.208.40.58
Welcome to ubuntu 20.04.4 LTS (GNU/Linux 5.11.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Feb 25 12:28:55 UTC 2022

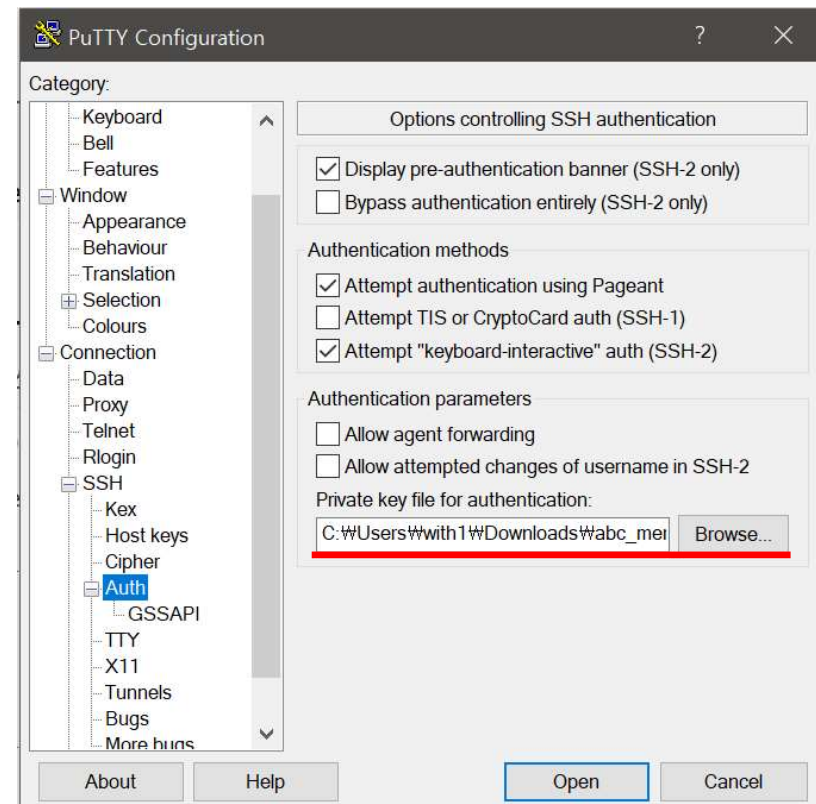
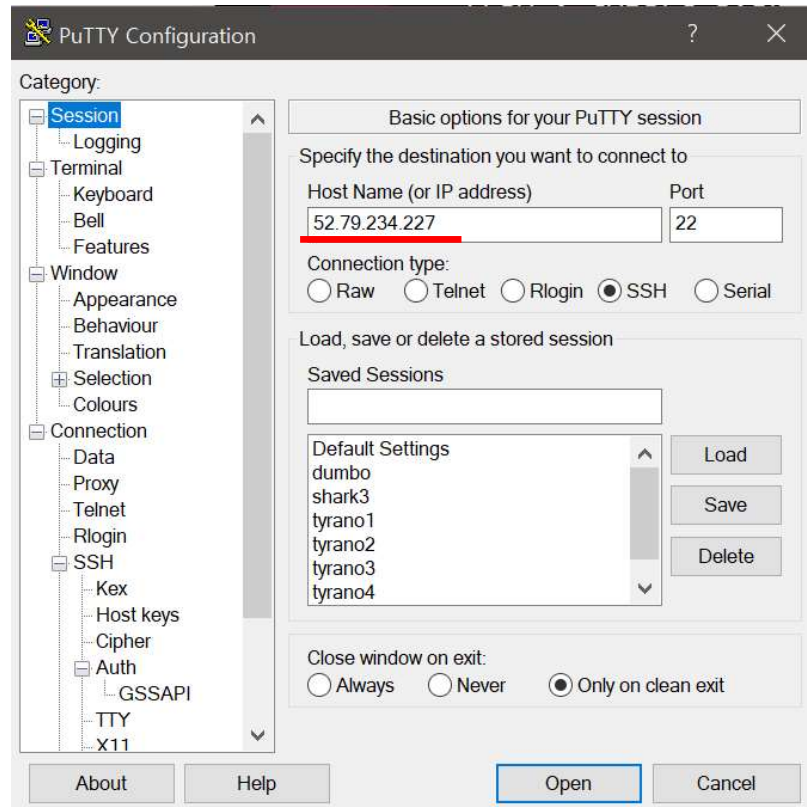
System load:  0.03                       Processes:            109
Usage of /:   41.0% of 7.69GB             Users logged in:      0
Memory usage: 41%                       IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                         IPv4 address for eth0:  172.31.95.208

7 updates can be applied immediately.
7 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Fri Feb 25 12:28:42 2022 from 58.126.54.18
ubuntu@ip-172-31-95-208:~$
```

# AWS EC2 인스턴스 접속하기

- Putty의 경우, Host name에 퍼블릭 DNS나 IP 주소를 입력하고 Connection -> SSH -> Auth 탭에서 ppk 파일을 Browse 합니다.





# AWS EC2 인스턴스 접속하기

- 이 후, open 버튼을 누른 후, ubuntu 이름으로 로그인하면 EC2 인스턴스 서버를 이용할 수 있습니다.

```
ubuntu@ip-172-31-40-165: ~  
login as: ubuntu  
Authenticating with public key "abc_mentoring"  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1037-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Wed Feb 17 09:44:10 UTC 2021  
  
System load:  0.0           Processes:            92  
Usage of /:   14.8% of 7.69GB Users logged in:       0  
Memory usage: 18%          IP address for eth0: 172.31.40.165  
Swap usage:   0%  
  
0 packages can be updated.  
0 of these updates are security updates.  
  
New release '20.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Wed Feb 17 09:32:42 2021 from 114.70.9.219  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-40-165:~$
```



# AWS EC2 설정

---

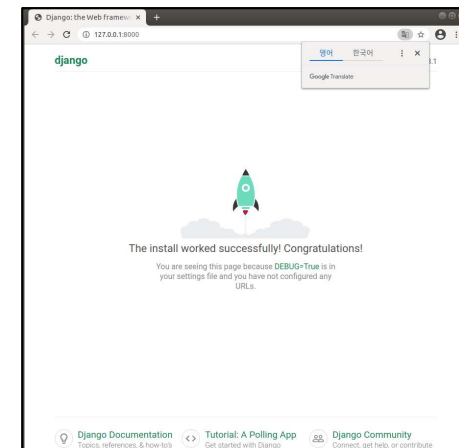
- EC2 접속이 완료되면 다음 명령어를 통해 EC2 환경을 업데이트 해줍니다.
- `sudo apt-get update`
- `sudo apt-get upgrade`
- `sudo apt-get install build-essential`
- 이 후 필요에 따라 vim이나 bash 또는 기타 환경 설정을 세팅해주면 됩니다.

# Docker를 통한 실제 서비스 배포

- 아래 링크는 Django라는 웹 프레임워크를 이용한 간단한 웹 페이지 서비스 프로그램입니다.
- [https://github.com/with1015/2022\\_Winter\\_ABC\\_Mentoring/tree/service](https://github.com/with1015/2022_Winter_ABC_Mentoring/tree/service)
- Python3와 pip가 있는 경우 다음 명령어를 통해 실행할 수 있습니다.
  - pip3 install Django
  - python3 manage.py runserver 8000

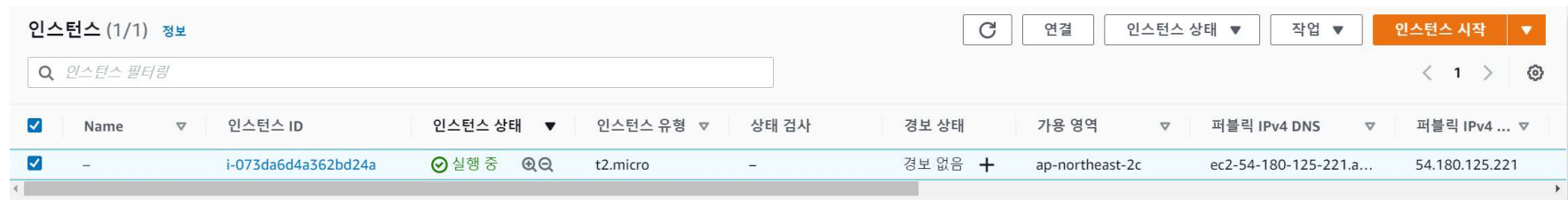
```
with1015@ubuntu:~/Docker_study/web_service/web_benchmark$ ls
db.sqlite3 manage.py web_benchmark
with1015@ubuntu:~/Docker_study/web_service/web_benchmark$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 18, 2021 - 06:24:00
Django version 3.1.6, using settings 'web_benchmark.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```



# Docker를 통한 실제 서비스 배포

- 하지만 이는 프로그램을 실행한 로컬 환경에서만 접속이 가능합니다.
- 실습을 통해 다른 사람들도 웹 페이지를 볼 수 있도록 docker와 AWS를 이용해 웹 서비스 배포를 해봅시다.
- 우선 배포하기에 앞서 AWS EC2에 인스턴스 하나를 만들어 준비하고, 인스턴스 내부에 docker를 설치 합니다.



The screenshot shows the AWS Management Console interface for EC2 instances. At the top, there's a header with '인스턴스 (1/1) 정보' and several action buttons like '연결', '인스턴스 상태', '작업', and '인스턴스 시작'. Below the header is a search bar labeled '인스턴스 필터링'. The main content is a table with columns for selection, Name, 인스턴스 ID, 인스턴스 상태, 인스턴스 유형, 상태 검사, 경보 상태, 가용 영역, 퍼블릭 IPv4 DNS, and 퍼블릭 IPv4 ... . There is one instance listed with ID 'i-073da6d4a362bd24a', status '실행 중' (Running), type 't2.micro', and public IP '54.180.125.221'.

<input checked="" type="checkbox"/>	Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS	퍼블릭 IPv4 ...
<input checked="" type="checkbox"/>	-	i-073da6d4a362bd24a	실행 중	t2.micro	-	경보 없음	ap-northeast-2c	ec2-54-180-125-221.a...	54.180.125.221

# Docker를 통한 실제 서비스 배포

---

- 그 다음 github에서 web benchmark 파일을 git clone을 통해 EC2 인스턴스에 복사합니다.

```
ubuntu@ip-172-31-36-224:~$ git clone https://github.com/with1015/web_benchmark.git
Cloning into 'web_benchmark'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 24 (delta 9), reused 20 (delta 5), pack-reused 0
Unpacking objects: 100% (24/24), done.
ubuntu@ip-172-31-36-224:~$ ls
web_benchmark
ubuntu@ip-172-31-36-224:~$
```

# Docker를 통한 실제 서비스 배포

- 다음과 같이 EC2 인스턴스의 퍼블릭 IPv4 주소를 web\_benchmark 폴더 내부에서 settings.py에 ALLOWED\_HOSTS 리스트에 추가합니다.

인스턴스: i-073da6d4a362bd24a

세부 정보 | 보안 | 네트워킹 | 스토리지 | 상태 검사 | 모니터링 | 태그

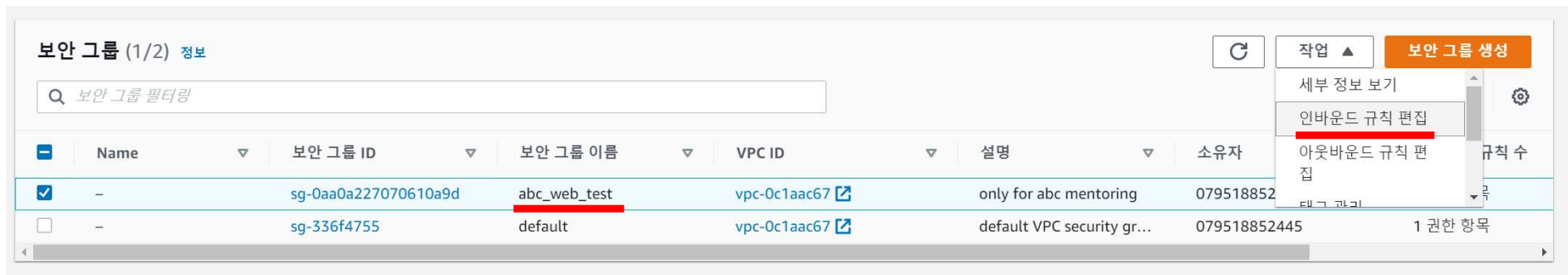
▼ 인스턴스 요약 정보

인스턴스 ID i-073da6d4a362bd24a	퍼블릭 IPv4 주소 54.180.125.221   <a href="#">개방 주소법</a>
인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-54-180-125-221.ap-northeast-2.compute.amazonaws.com   <a href="#">개방 주소법</a>

```
27
28 ALLOWED_HOSTS = ['127.0.0.1', '54.180.125.221']
29
30
```

# Docker를 통한 실제 서비스 배포

- EC2 대시보드 왼쪽 메뉴 -> 네트워크 및 보안 -> 보안그룹
- 생성한 EC2 인스턴스 이름을 가진 그룹을 선택하고 작업 -> 인바운드 규칙 편집을 클릭해줍니다.



보안 그룹 (1/2) 정보

Q 보안 그룹 필터링

	Name	보안 그룹 ID	보안 그룹 이름	VPC ID	설명	소유자	
<input checked="" type="checkbox"/>	-	sg-0aa0a227070610a9d	abc_web_test	vpc-0c1aac67	only for abc mentoring	079518852	
<input type="checkbox"/>	-	sg-336f4755	default	vpc-0c1aac67	default VPC security gr...	079518852445	1 권한 항목

작업 ▲

- 세부 정보 보기
- 인바운드 규칙 편집
- 아웃바운드 규칙 편집
- 태그 관리

보안 그룹 생성

# Docker를 통한 실제 서비스 배포

- 인바운드 규칙 편집에서 **규칙 추가** 버튼을 누른 뒤, **사용자 지정 TCP -> 포트 범위 8000 -> 소스는 위치 무관** 으로 설정해주고 규칙 저장 버튼을 누릅니다.

## 인바운드 규칙 편집 정보

인바운드 규칙은 인스턴스에 도달하도록 허용된 수신 트래픽을 제어합니다.

### 인바운드 규칙 정보

유형 <small>정보</small>	프로토콜 <small>정보</small>	포트 범위 <small>정보</small>	소스 <small>정보</small>
SSH ▼	TCP	22	사용자 지정 ▼
사용자 지정 TCP ▼	TCP	8000	위치 무관 ▼

# Docker를 통한 실제 서비스 배포

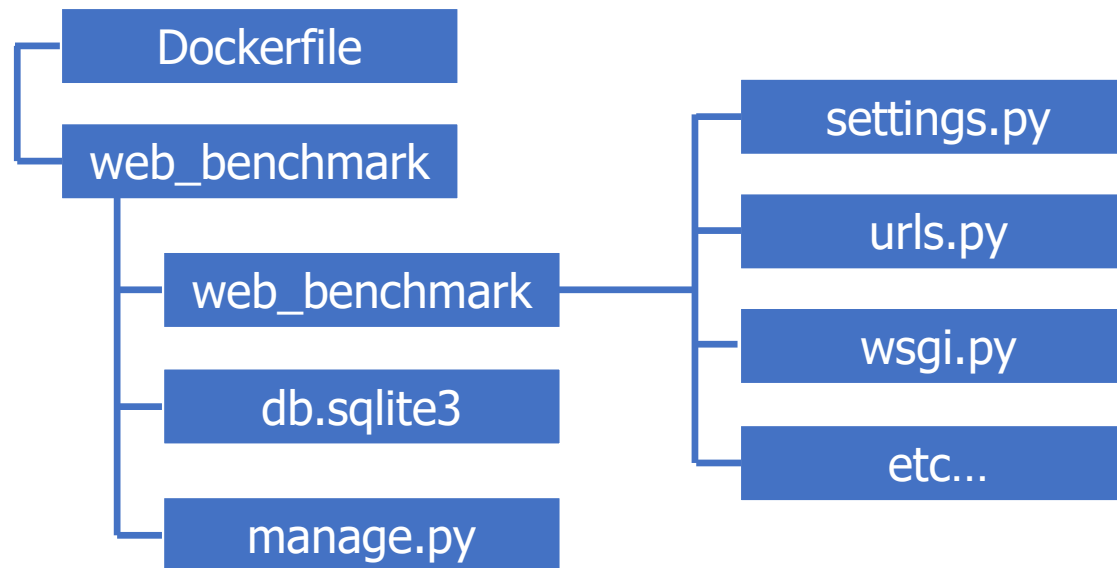
- EC2 인스턴스에 Dockerfile을 만든 후, 다음과 같이 작성해줍니다.

```
Dockerfile+
1 FROM python:3.6
2 MAINTAINER Hyunjoon_Jeong "with1015@unist.ac.kr"
3
4 RUN apt-get update
5 COPY . /app
6
7 WORKDIR /app/web_benchmark
8
9 RUN pip3 install Django
10
11 EXPOSE 8000
12
13 CMD ["python", "manage.py", "runserver", "0:8000"]
14
```



# Docker를 통한 실제 서비스 배포

- Dockerfile과 web\_benchmark 폴더는 반드시 다음과 같은 디렉토리 구조를 유지해야 합니다.



# Docker를 통한 실제 서비스 배포

- Dockerfile이 완성되었으면, 다음과 같은 명령어를 통해 Dockerfile을 빌드하여 배포 이미지를 생성합니다.
  - `sudo docker build -t abc_mentoring:abc .`
- 성공적으로 build 되었을 경우 다음과 같이 docker images 명령어를 통해 생성된 이미지를 확인할 수 있습니다.

```
ubuntu@ip-172-31-36-224:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
abc_mentoring	abc	8102017c9943	30 seconds ago	933MB
python	3.6	1ce843b4cf8c	35 hours ago	875MB

```
ubuntu@ip-172-31-36-224:~$
```

# Docker를 통한 실제 서비스 배포

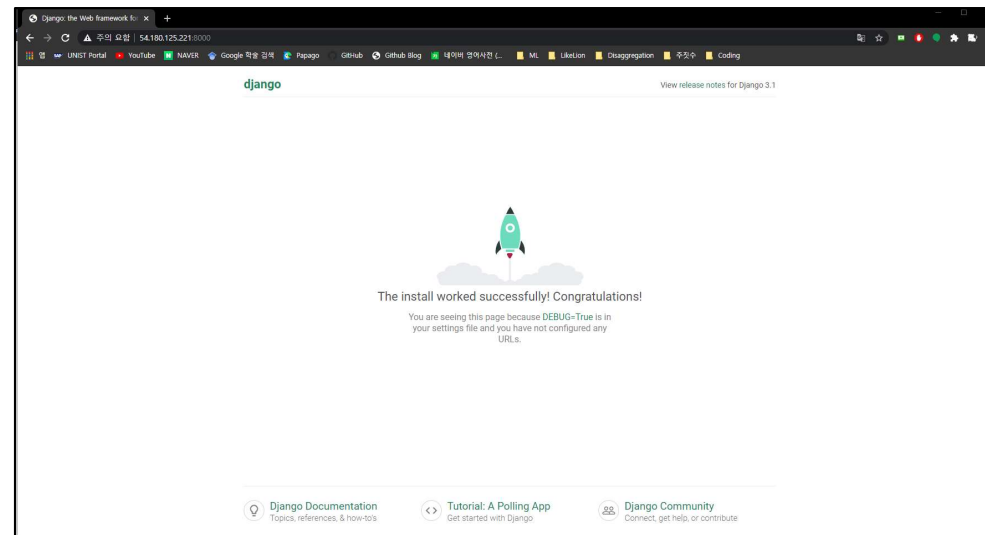
---

- 그 다음, 생성한 이미지를 컨테이너로 만들어야 합니다.
- 다음 명령어를 통해 이미지를 컨테이너로 만들고 실행할 수 있습니다.
  - `sudo docker run -p 8000:8000 --name abc_web abc_mentoring:abc`
  - 또는 `sudo docker run -d -p 8000:8000 --name abc_web abc_mentoring:abc`
- -d 옵션은 docker 컨테이너를 백그라운드로 실행하는 것을 의미합니다.
- -p 옵션은 컨테이너를 포트를 통해 접속할 수 있도록 열어줍니다.
- --name 옵션은 컨테이너에 이름을 부여합니다.

# Docker를 통한 실제 서비스 배포

- 컨테이너가 성공적으로 실행되면 크롬을 열고 [EC2 IPv4 주소]:8000를 주소창에 입력하면 다음과 같이 웹 화면이 나타나게 됩니다.

```
ubuntu@ip-172-31-36-224:~$ docker run -p 8000:8000 --name abc_test abc_mentoring:abc
Watching for file changes with StatReloader
[18/Feb/2021 07:34:09] "GET / HTTP/1.1" 200 16351
[18/Feb/2021 07:34:09] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
[18/Feb/2021 07:34:10] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[18/Feb/2021 07:34:10] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[18/Feb/2021 07:34:10] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
```



# Docker를 통한 실제 서비스 배포

---

- 한번 docker 컨테이너를 통해 서버에 올라가게 되면 컨테이너를 종료하지 않는 이상 터미널을 종료해도 웹 서버가 꺼지지 않습니다.
- 다음 명령어를 이용해 생성한 컨테이너와 빌드한 이미지를 제거할 수 있습니다.
- `sudo docker kill [웹 서비스 컨테이너 이름]`
- `sudo docker rm [웹 서비스 컨테이너 이름]`
- `sudo docker rmi [빌드했던 이미지 이름]`

# Activity

---

- AWS EC2를 이용해 간단한 Django를 올려봅시다.

# Thank you

