



AWS EC2 Linux 환경에 Flask 웹서버 구축하기

Jin Won,

School of Computer Engineering,

Hansung University

Contents

- PuTTY를 사용하여 Windows PC에서 AWS EC2 Linux 서버로 접속한다.
- 터미널에 다음 페이지부터 등장하는 명령어들을 입력한다.

```
ubuntu@ip-172-31-43-88: ~  
$  
Using username "ubuntu".  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sun Jun 30 19:09:12 UTC 2019  
  
System load:  0.0                Processes:            87  
Usage of /:   24.8% of 7.69GB    Users logged in:     0  
Memory usage: 19%                IP address for eth0: 172.31.43.88  
Swap usage:   0%  
  
* MicroK8s 1.15 is out! It has already been installed on more  
  than 14 different distros. Guess which ones?  
  
  https://snapcraft.io/microk8s  
  
Get cloud support with Ubuntu Advantage Cloud Guest:  
  http://www.ubuntu.com/business/services/cloud  
  
7 packages can be updated.  
7 updates are security updates.  
  
Last login: Sun Jun 30 18:50:12 2019 from 221.146.31.238  
ubuntu@ip-172-31-43-88:~$
```

Contents

- 기설치된 서비스 업데이트 및 업그레이드 실시

```
ubuntu@ip-172-31-43-88:~$ sudo apt-get update
```

```
ubuntu@ip-172-31-43-88:~$ sudo apt-get upgrade
```

- python 설치

```
ubuntu@ip-172-31-43-88:~$ sudo apt-get install python3.7
```

- pip(각종 python 라이브러리 설치를 도와주는 도구) 설치

```
ubuntu@ip-172-31-43-88:~$ sudo apt-get install python3-pip
```

Contents

- python3 가상 환경 만들기

```
ubuntu@ip-172-31-43-88:~$ sudo apt-get install python3-venv
```

```
ubuntu@ip-172-31-43-88:~$ mkdir OnlineDataAnalyzer
```

```
ubuntu@ip-172-31-43-88:~$ cd OnlineDataAnalyzer/
```

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ python3 -m venv godjin
```

Contents

- Flask 설치

```
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ pip install Flask
```

- boto3 설치

```
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ pip install boto3
```

- nltk 설치

```
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ pip install -U nltk
```

Contents

- FileZilla를 통해 Windows PC에서 작업한 코드들을 Linux 서버로 가져온다.

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ ls  
__main__.py  __pycache__  application  godjin  venv
```

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer/application$ ls  
__init__.py  __pycache__  nlp  static  templates
```

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer/application/nlp$ ls  
EntityDetector.py  PhraseExtractor.py  __init__.py  accessKeys.csv  
NGramsExtractor.py  WordPairExtractor.py  __pycache__
```

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer/application/static$ ls  
download.js  main.js
```

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer/application/templates$ ls  
index.html
```

Contents

- Flask 웹서버를 실행하는 방법은 두 가지가 있다.
- 첫번째 방법: user가 PuTTY로 접속하고 있는 동안에만 프로세스가 살아있다.

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ source godjin/bin/activate
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ python __main__.py
* Serving Flask app "application" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
```

Contents

- 두번째 방법: user 접속 유무와 상관없이 프로세스가 살아있다.
백그라운드에서 돌아간다.

```
ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ source godjin/bin/activate
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ python __main__.py & disown
[1] 2394
(godjin) ubuntu@ip-172-31-43-88:~/OnlineDataAnalyzer$ * Serving Flask app "application" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
```


Contents

- 두번째 방법을 채택한 경우, 프로세스를 종료하려면 아래를 참고한다.

```
ubuntu@ip-172-31-43-88:~$ ps -aux | grep python
root      840   0.0  1.7 170880 17396 ?        Ssl  18:14   0:00 /usr/bin/pytho
3 /usr/bin/networkd-dispatcher --run-startup-triggers
root      865   0.0  2.0 187712 20208 ?        Ssl  18:14   0:00 /usr/bin/pytho
3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
ubuntu    2251   0.8  4.8 197528 49292 ?        S    18:49   0:00 python __main__
.py
ubuntu    2389   0.0  0.0  14856  1004 pts/1    S+   18:50   0:00 grep --color=au
to python
ubuntu@ip-172-31-43-88:~$ kill -9 2251
```

Contents

- 웹 페이지 접속

‘퍼블릭 DNS(IPv4):포트번호’

or

‘IPv4 퍼블릭 IP:포트번호’

The screenshot shows the AWS Management Console interface for an EC2 instance. The instance details table is as follows:

Name	인스턴스 ID	인스턴스 유형	가용 영역	인스턴스 상태	상태 검사	경보 상태	퍼블릭 DNS(IPv4)	IPv4 퍼블릭 IP	IPv6 IP
	i-0393bee8fa32951cf	t2.micro	us-east-2c	running	2/2 검사 통과	없음	ec2-3-17-28-229.us-east-2.compute.amazonaws.com	3.17.28.229	-

Below the console, a web browser window is open to the URL `3.17.28.229:8000`. The page displays an "Online Data Analyzer" interface with the following elements:

- Buttons: 파일 선택, 선택된 파일 없음
- Radio buttons: ☒ Named Entity Detection, ☐ N-Grams Extraction, ☐ Word Pair Extraction, ☐ Phrase Detection
- Buttons: 전송, 다운로드
- Text: 내용이 이곳에 출력됩니다.