

# AI기법과 활용

## Week-02. Computer Vision Basic

2022-Summer 서종원

시작하기 전에 복습

# 작업 과정의 큰 그림

## 로컬환경과 서버 환경



로컬환경  
=  
우리의 컴퓨터

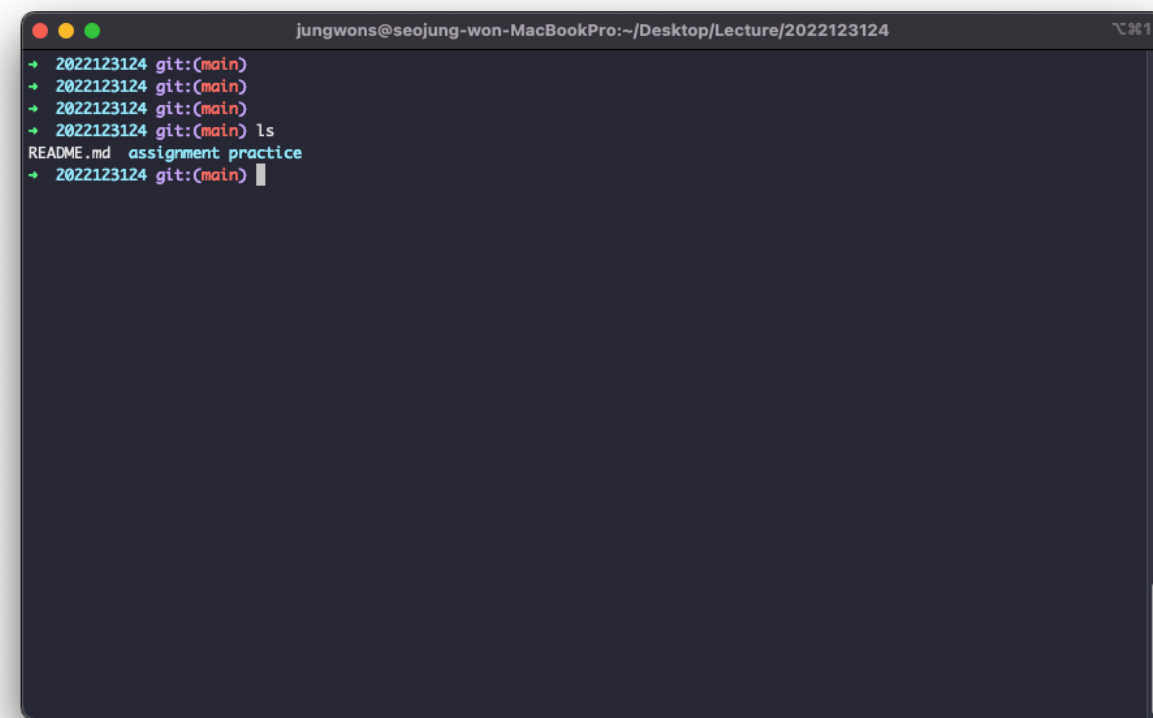
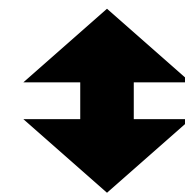


서버환경  
=  
Google Cloud

# 작업 과정의 큰 그림

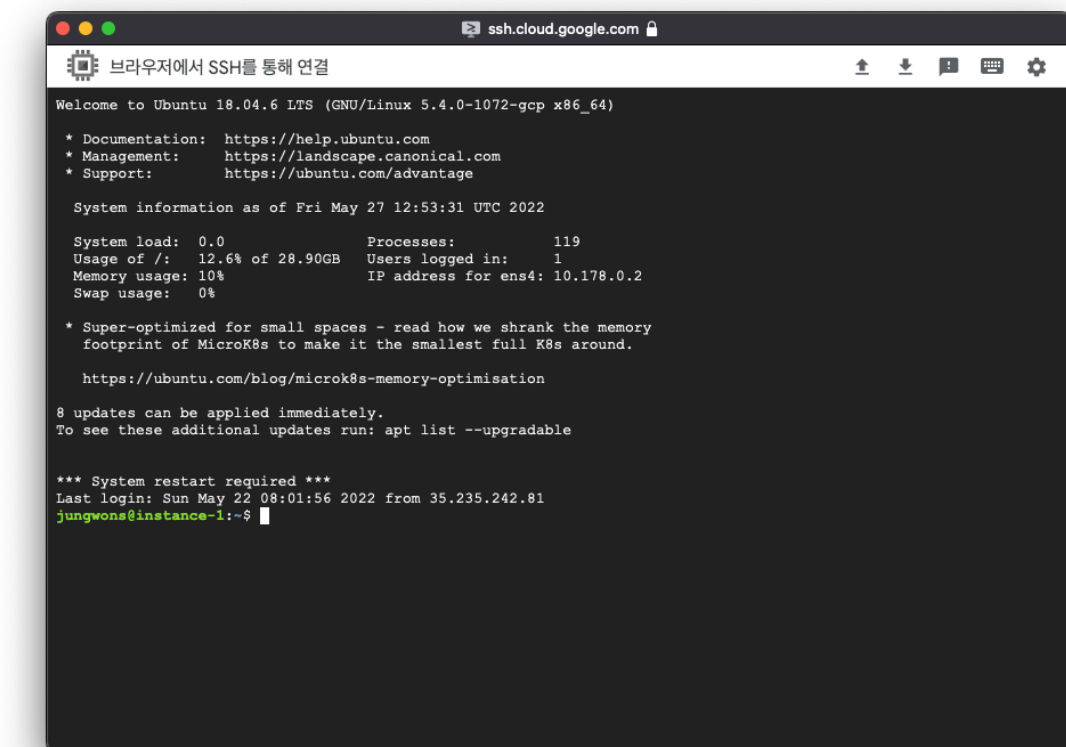
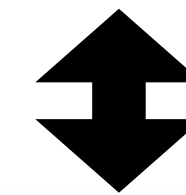
## 로컬환경과 서버 환경

로컬환경



Gitbash 또는 Conda Prompt

서버환경

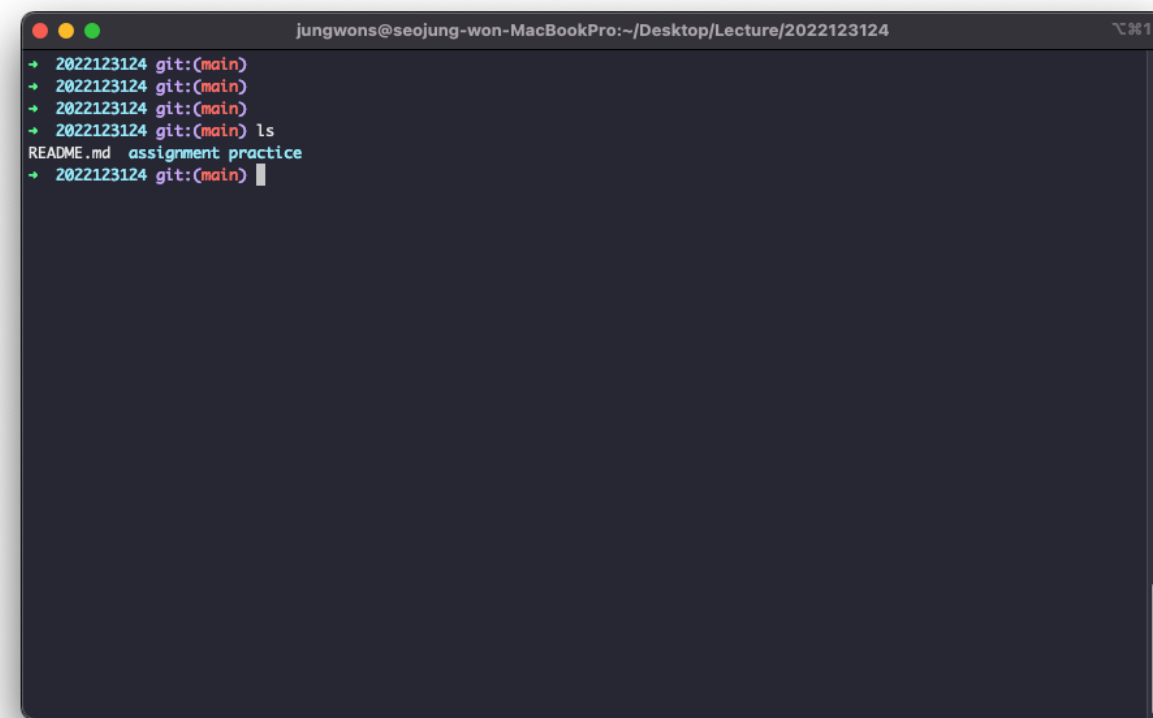
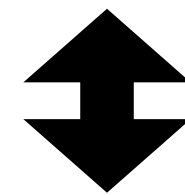


브라우저 상에서의 터미널

# 작업 과정의 큰 그림

## 로컬환경과 서버 환경

로컬환경



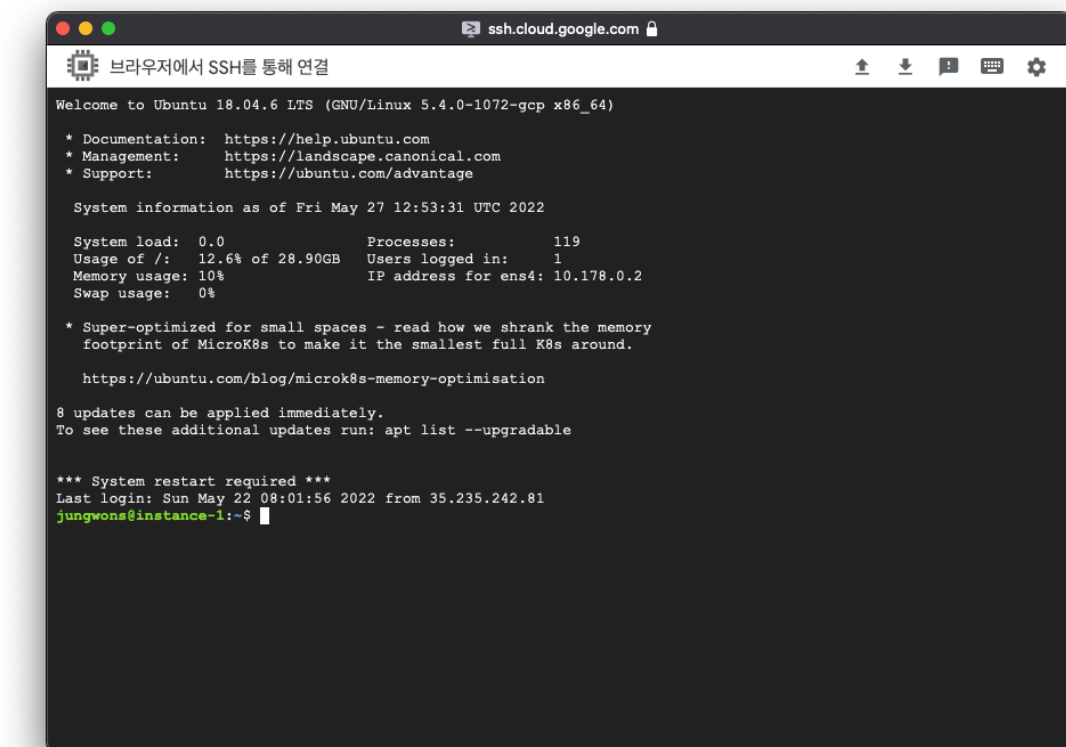
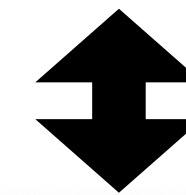
Gitbash 또는 Conda Prompt



Git push

Git pull

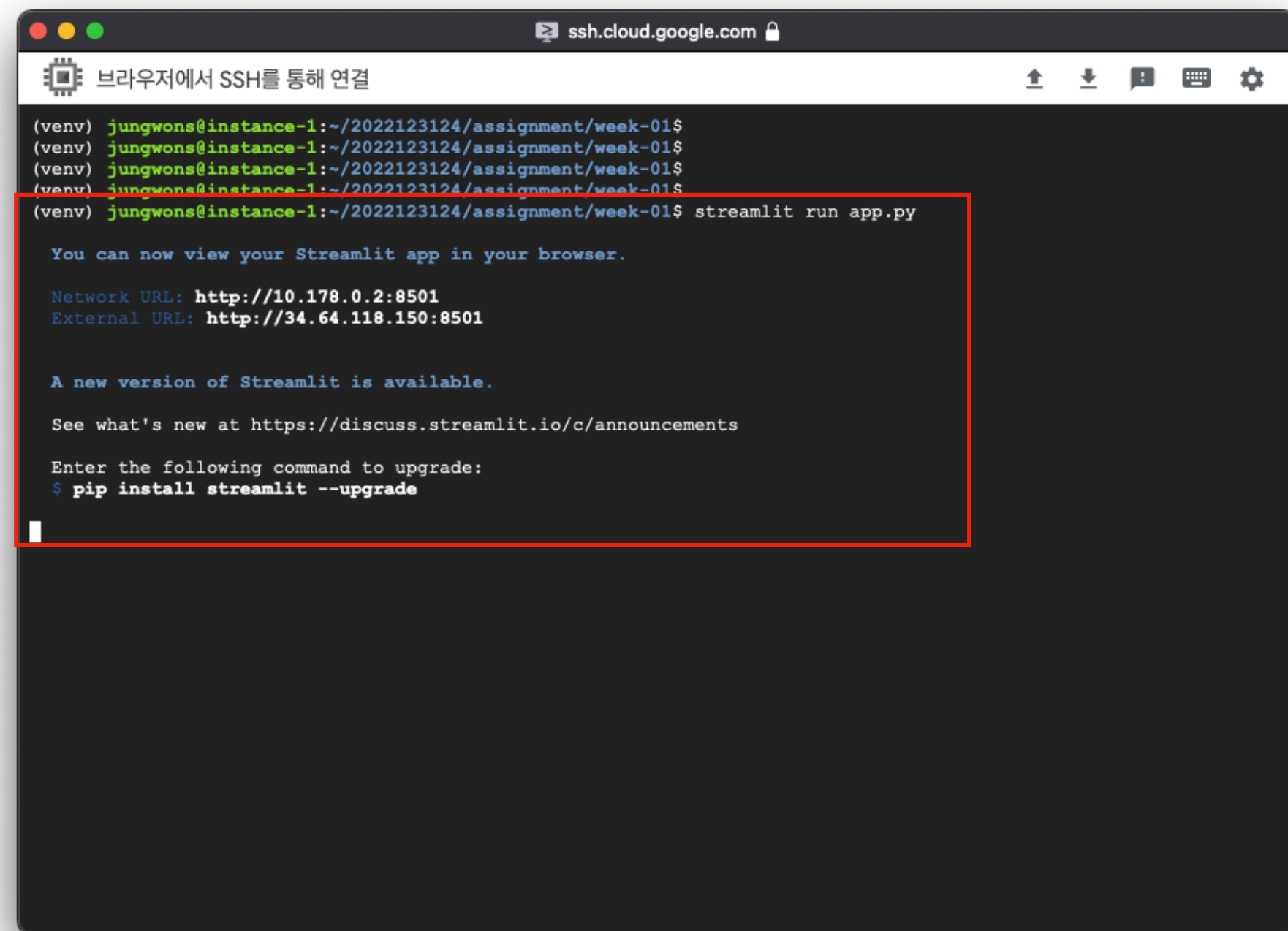
서버환경



브라우저 상에서의 터미널

# 백그라운드 작업

## 로컬환경과 서버 환경



The screenshot shows a terminal window connected via SSH to a Google Cloud instance. The user is in a virtual environment (venv) and has executed the command `streamlit run app.py`. The output indicates the app is running and provides network and external URLs. A red box highlights the output text.

```
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$ streamlit run app.py  
  
You can now view your Streamlit app in your browser.  
  
Network URL: http://10.178.0.2:8501  
External URL: http://34.64.118.150:8501  
  
A new version of Streamlit is available.  
  
See what's new at https://discuss.streamlit.io/c/announcements  
  
Enter the following command to upgrade:  
$ pip install streamlit --upgrade
```

이 상태에서 터미널을 끄게 된다면?

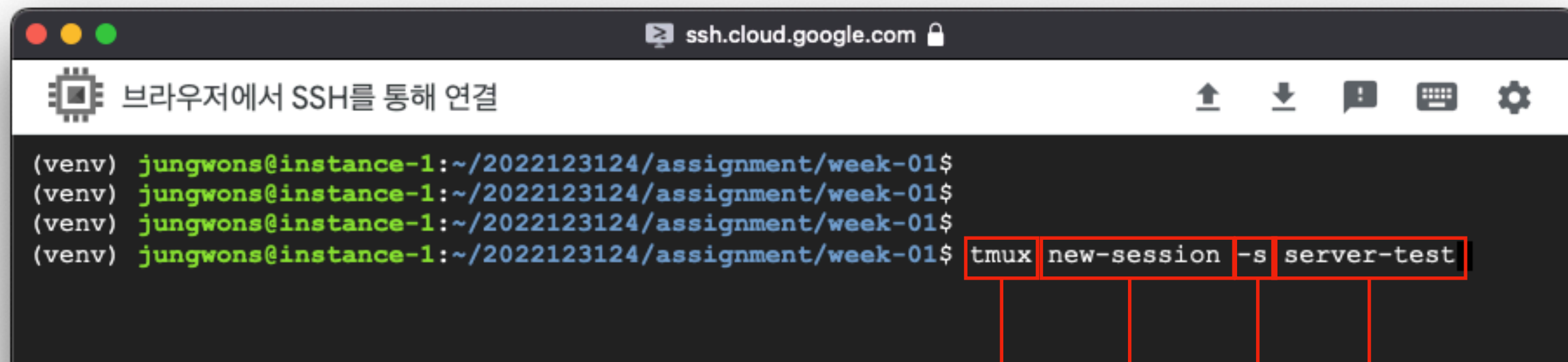
돌고있는 streamlit 앱 또한 종료

현재 상태를 foreground에서 돌리고 있다고 표현

# 백그라운드 작업

## 로컬환경과 서버 환경

tmux를 활용하여 백그라운드 세션 실행



```
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$  
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$ tmux new-session -s server-test
```

tmux로

새로운 Session을 만들겠다

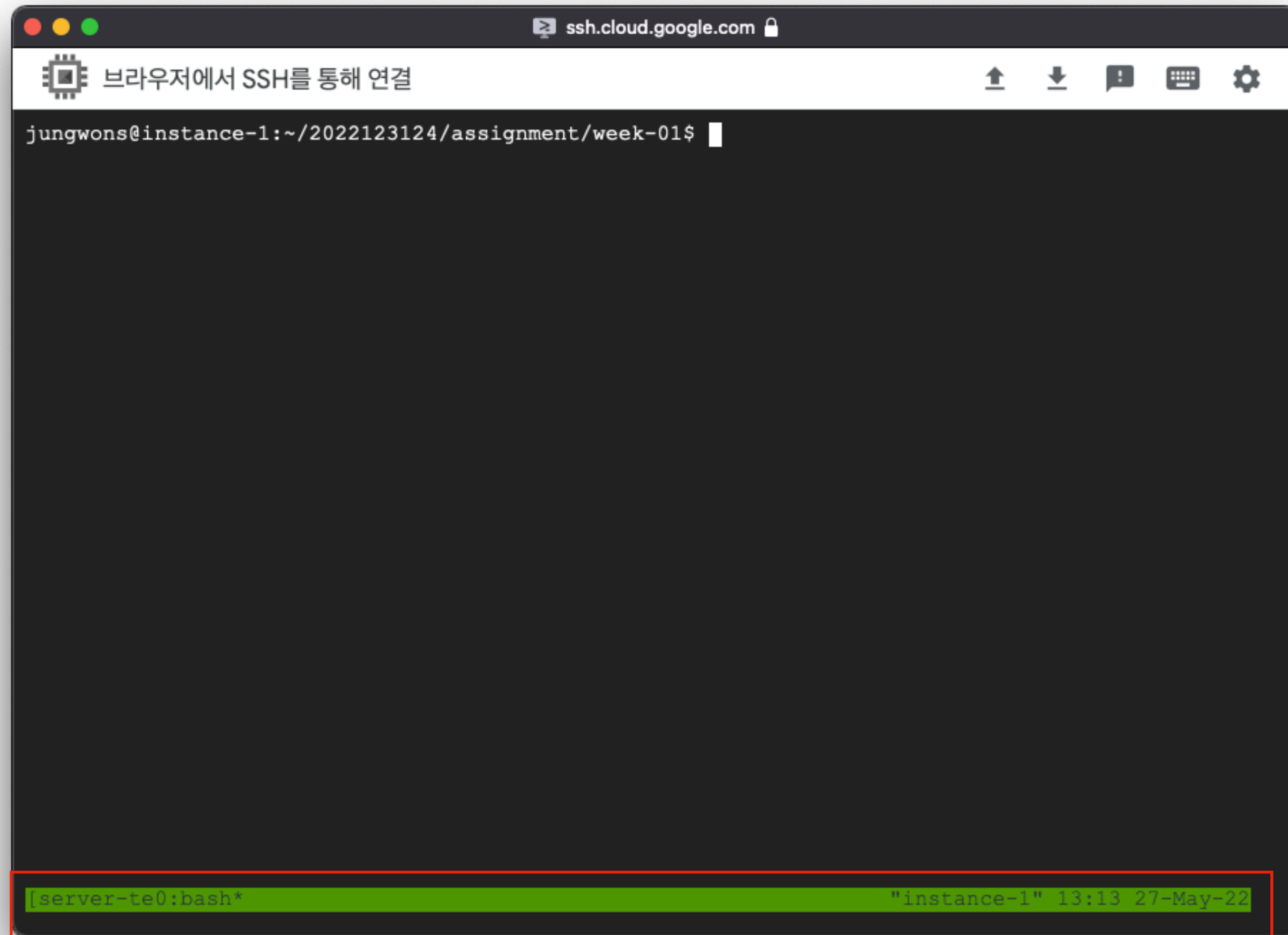
그 세션의 이름은..

server-test다



# 백그라운드 작업

## 로컬환경과 서버 환경



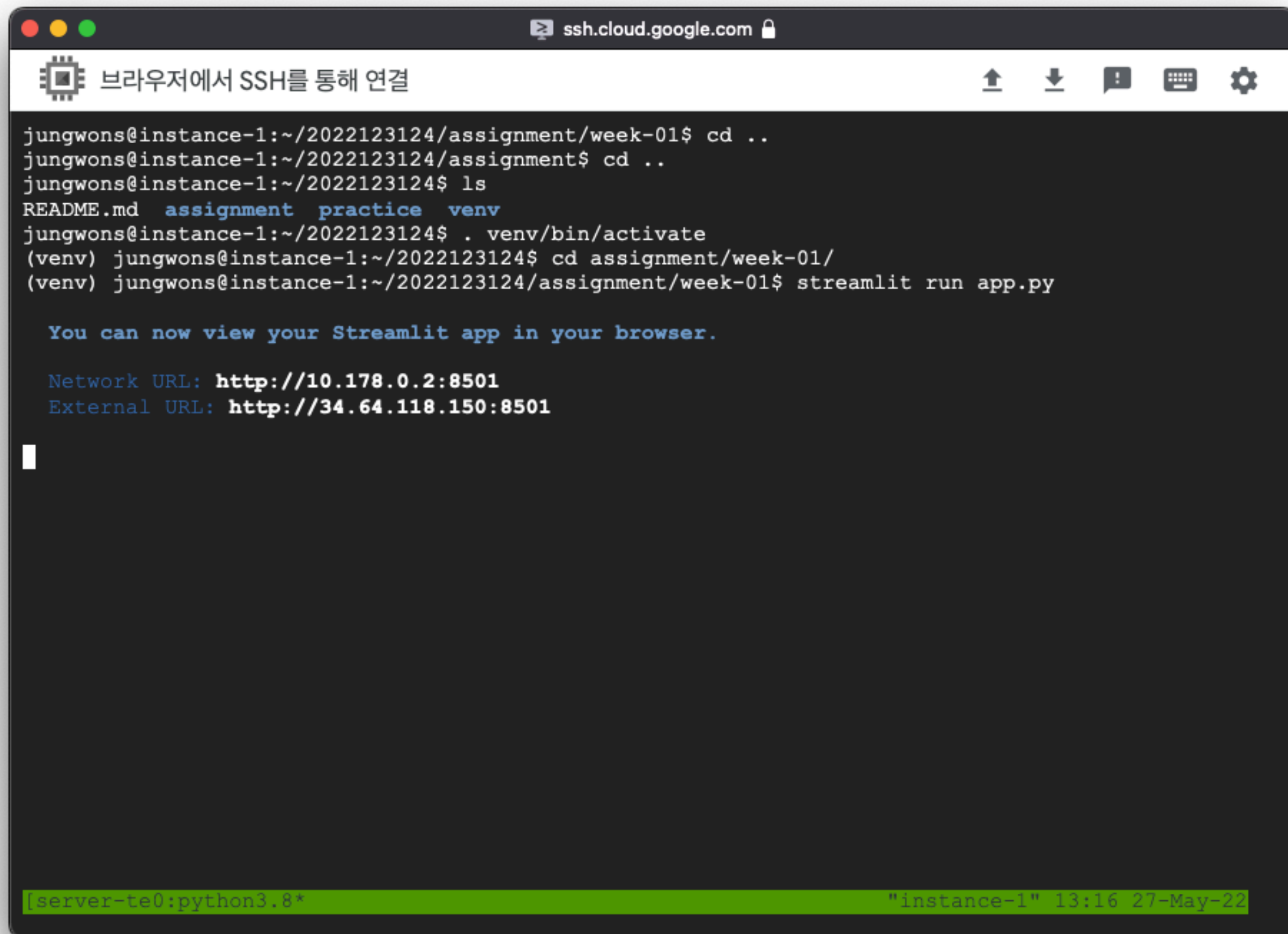
이 상태에서  
다시 가상환경을 활성화 시키고 앱을 실행

Tmux 백그라운드 세션은 하단의 다음과 같은 표시가 있습니다.



# 백그라운드 작업

## 로컬환경과 서버 환경



```
ssh.cloud.google.com
브라우저에서 SSH를 통해 연결
jungwons@instance-1:~/2022123124/assignment/week-01$ cd ..
jungwons@instance-1:~/2022123124/assignment$ cd ..
jungwons@instance-1:~/2022123124$ ls
README.md  assignment  practice  venv
jungwons@instance-1:~/2022123124$ . venv/bin/activate
(venv) jungwons@instance-1:~/2022123124$ cd assignment/week-01/
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$ streamlit run app.py

You can now view your Streamlit app in your browser.

Network URL: http://10.178.0.2:8501
External URL: http://34.64.118.150:8501

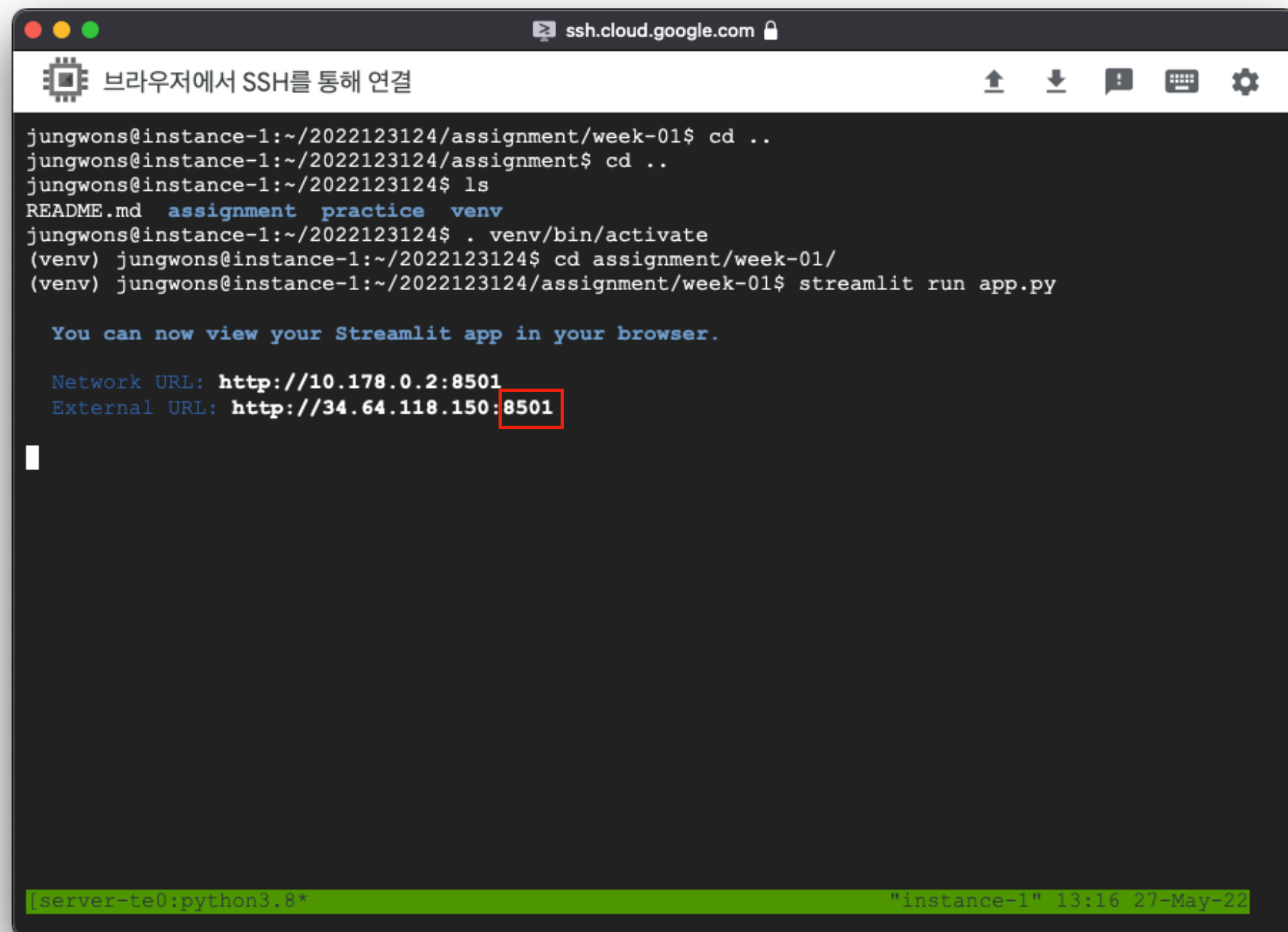
[server-te0:python3.8* "instance-1" 13:16 27-May-22]
```

이 상태에서 백그라운드 세션을 빠져나오실때는  
박자에 맞춰  
ctrl+b d

아니면 그냥 끄셔도 됩니다!

# FAQ

## 주소가 계속 변합니다



ssh.cloud.google.com

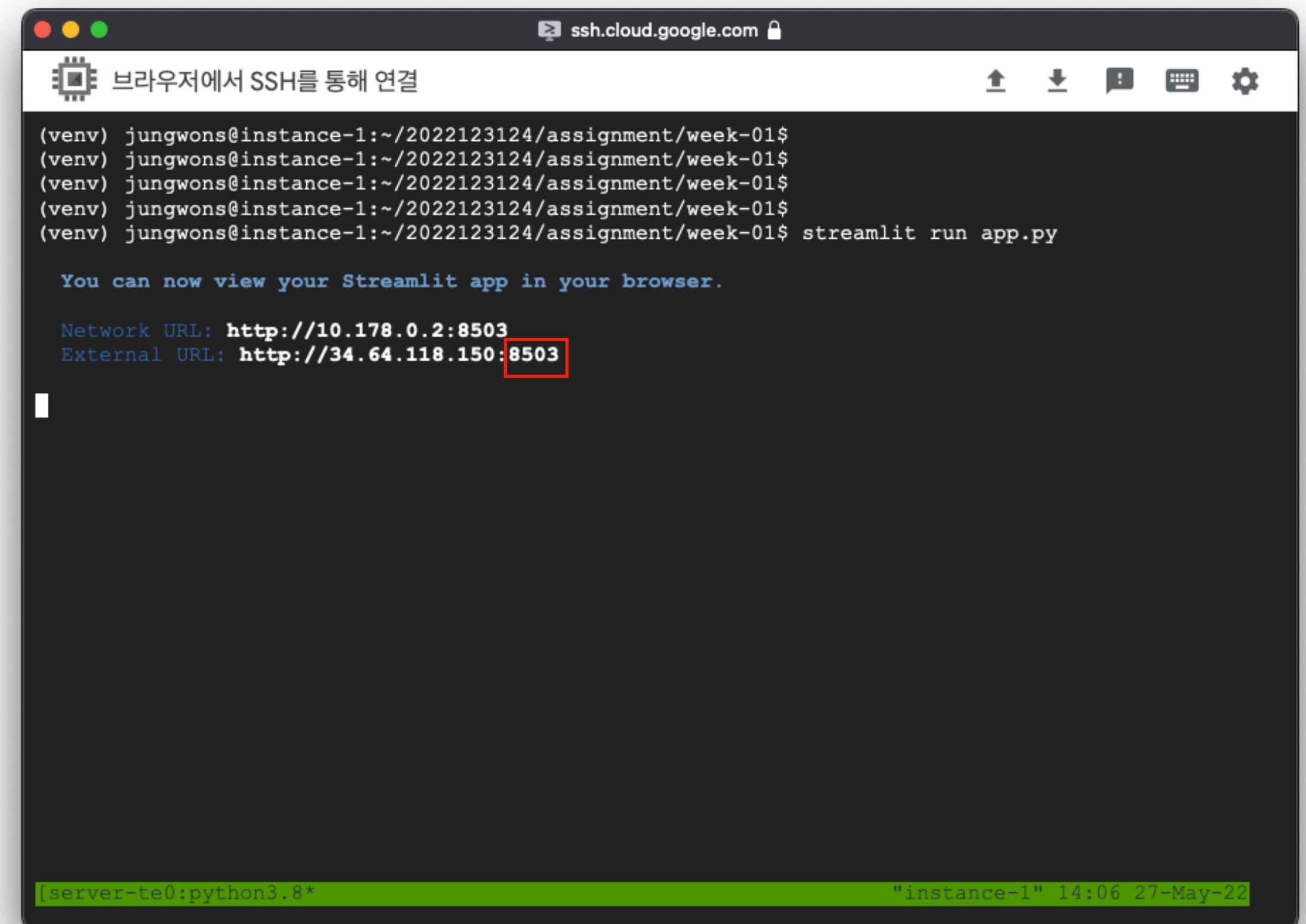
브라우저에서 SSH를 통해 연결

```
jungwons@instance-1:~/2022123124/assignment/week-01$ cd ..
jungwons@instance-1:~/2022123124/assignment$ cd ..
jungwons@instance-1:~/2022123124$ ls
README.md  assignment  practice  venv
jungwons@instance-1:~/2022123124$ . venv/bin/activate
(venv) jungwons@instance-1:~/2022123124$ cd assignment/week-01/
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$ streamlit run app.py

You can now view your Streamlit app in your browser.

Network URL: http://10.178.0.2:8501
External URL: http://34.64.118.150:8501
```

[server-te0:python3.8\* "instance-1" 13:16 27-May-22]



ssh.cloud.google.com

브라우저에서 SSH를 통해 연결

```
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$
(venv) jungwons@instance-1:~/2022123124/assignment/week-01$ streamlit run app.py

You can now view your Streamlit app in your browser.

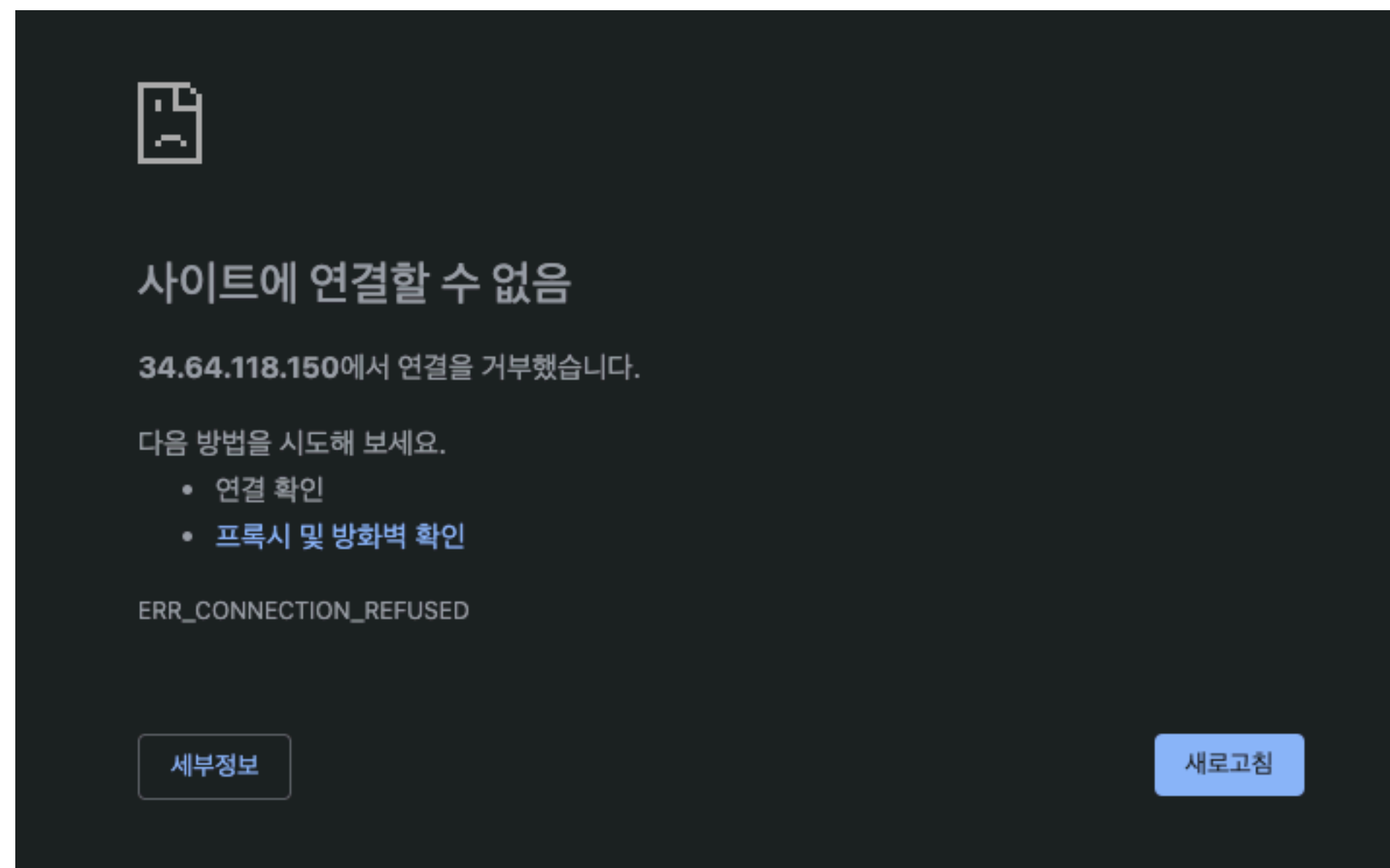
Network URL: http://10.178.0.2:8503
External URL: http://34.64.118.150:8503
```

[server-te0:python3.8\* "instance-1" 14:06 27-May-22]

이미 돌고 있는 streamlit app이 있는 경우 다른 포트로 새로 돌리게됩니다.

# FAQ

## 접속이 안됩니다 1

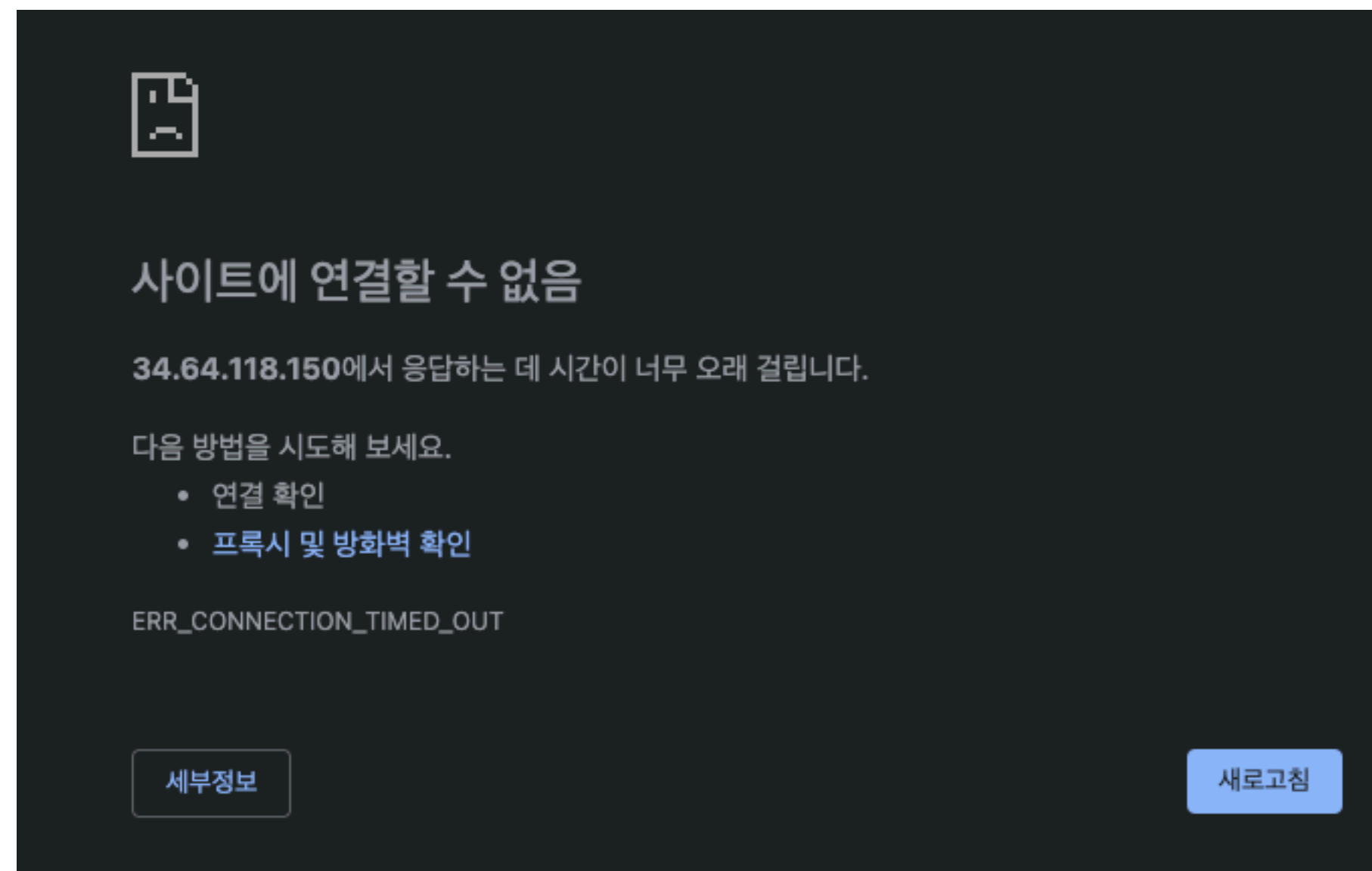


연결을 거부한 경우에는, 서버에 접근은 가능한데,  
서버에서 요청에 대응하는 앱이 돌고 있지 않다.

주소를 복사하는 과정에서 ctrl+c를 누르셔서 그럴 수 있습니다.  
ctrl+c는 터미널상에서 강제종료 단축키입니다.

# FAQ

## 접속이 안됩니다 2



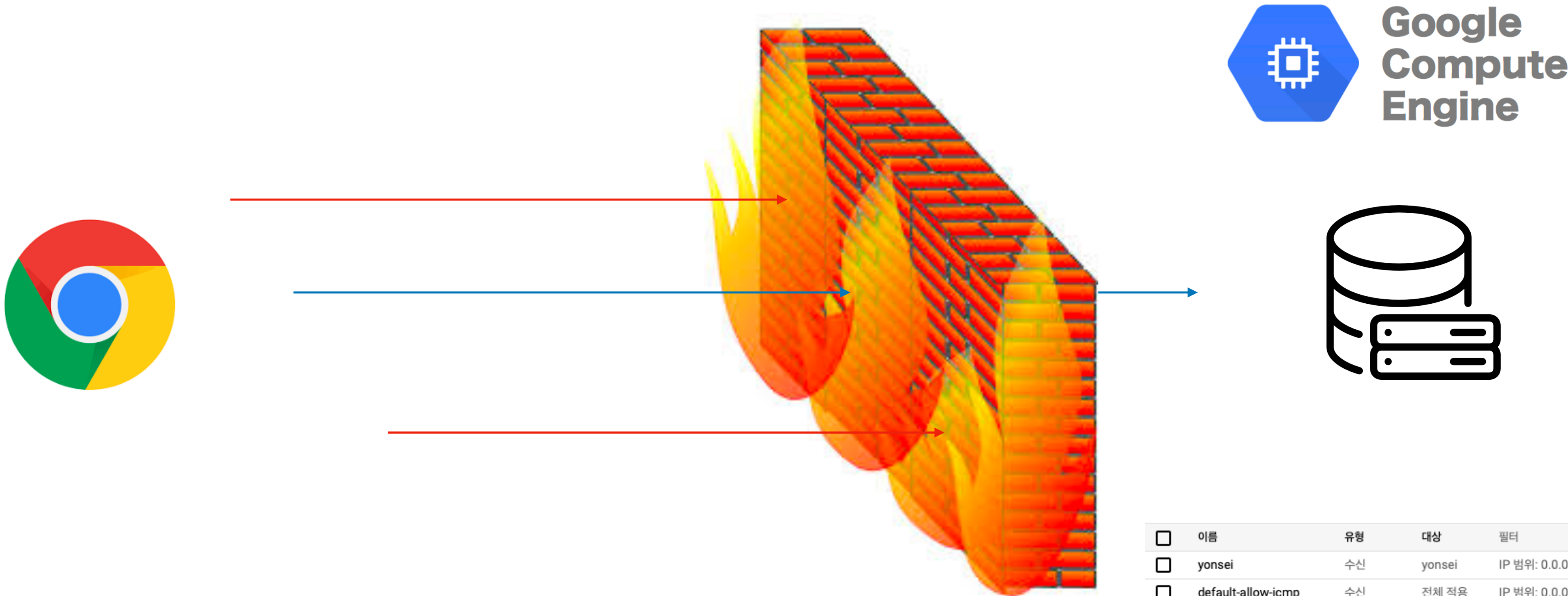
응답하는데 시간이 너무 오래 걸리는 경우

GCP 방화벽 세팅하는 과정에서 문제가 있었을 수 있습니다.



# 방화벽과 Compute Engine

방화벽 세팅은 무엇인가요?



외부 접속에 대해서 통제를 하는 역할

<input type="checkbox"/>	이름	유형	대상	필터	프로토콜/ 포트	작업
<input type="checkbox"/>	yonsei	수신	yonsei	IP 범위: 0.0.0.0/0	tcp:8000-9000	허용
<input type="checkbox"/>	default-allow-icmp	수신	전체 적용	IP 범위: 0.0.0.0/0	icmp	허용
<input type="checkbox"/>	default-allow-internal	수신	전체 적용	IP 범위: 10.128.0.0/9	tcp:0-65535 udp:0-65535 icmp	허용
<input type="checkbox"/>	default-allow-rdp	수신	전체 적용	IP 범위: 0.0.0.0/0	tcp:3389	허용
<input type="checkbox"/>	default-allow-ssh	수신	전체 적용	IP 범위: 0.0.0.0/0	tcp:22	허용

이번주 내용 시작

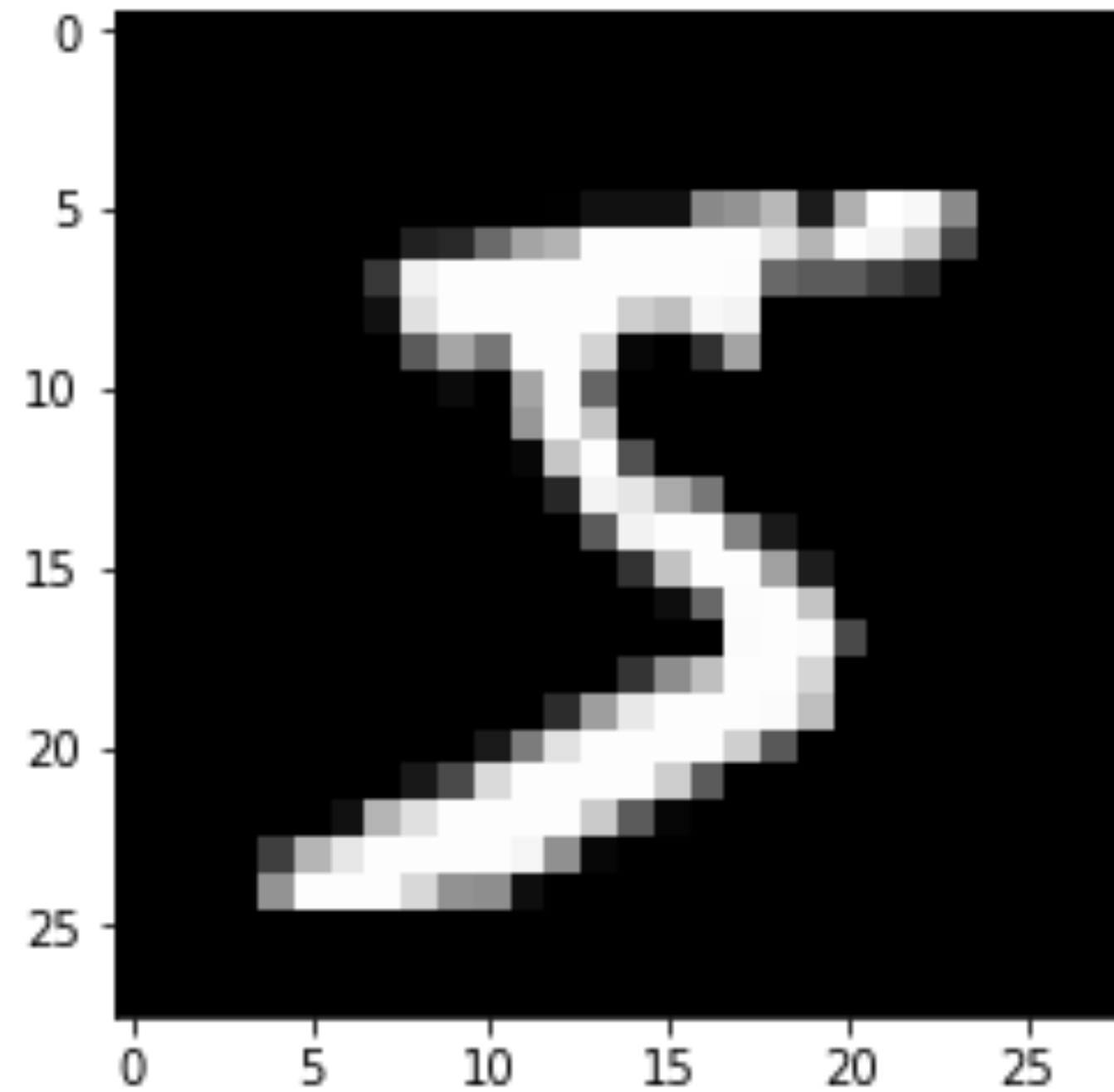
# 기계의 관점에서의 이미지와 사람의 관점에서의 이미지



Proyas, A. (2004). I, Robot. Twentieth Century Fox 중 장면



# 컴퓨터 상에서의 이미지는 결국 행렬

[illegible]

# 픽셀 (Pixel)

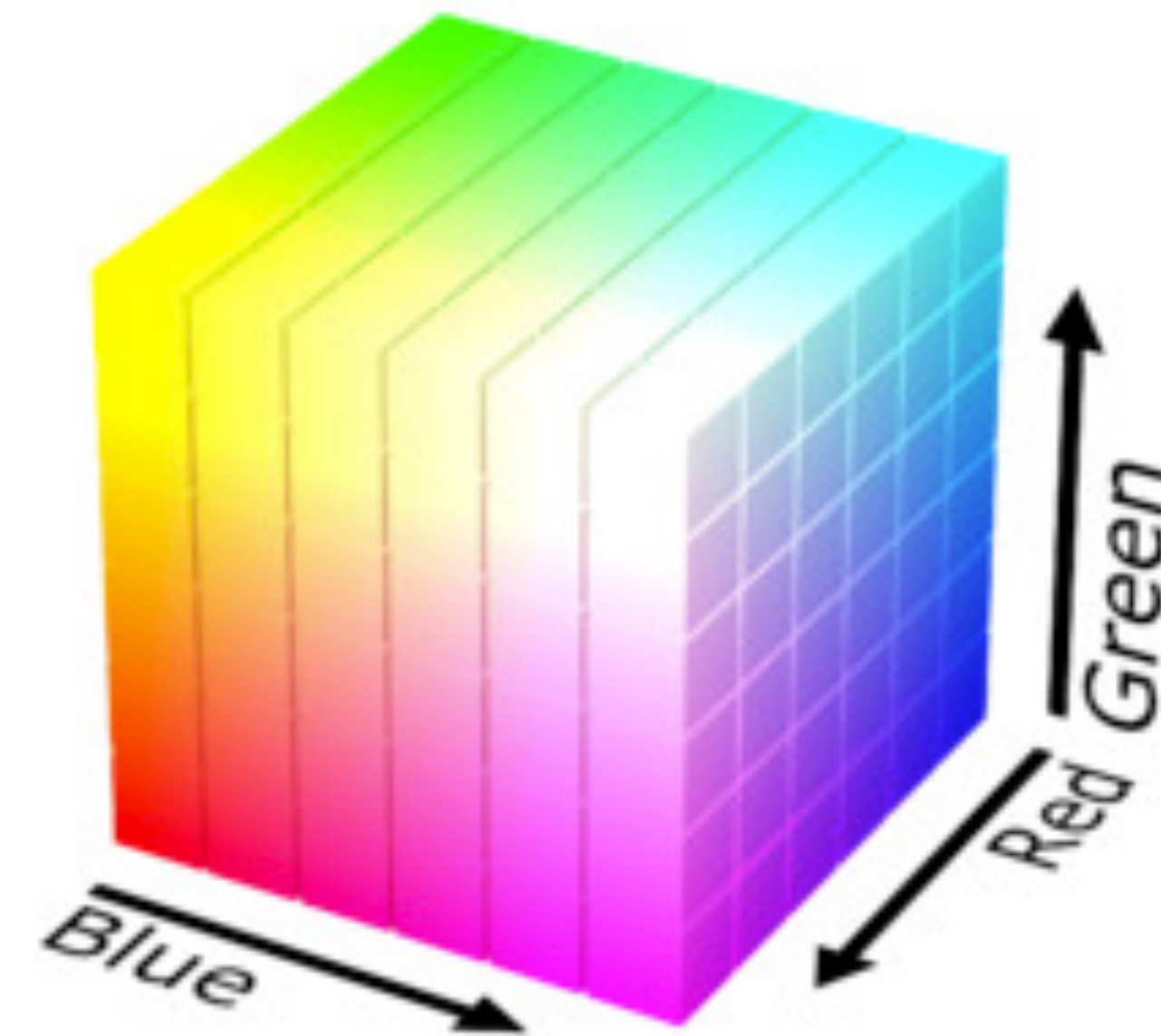
## 이미지의 정의

- 3차원 세상에 대한 2차원 화면
- 디지털 이미지는 한정된 값의 묶음으로 2차원 이미지를 숫자로 표현
- 이러한 값을 **픽셀**이라고 부르며, 집합으로서 이미지를 나타냄
- 다른말로 픽셀은 컴퓨터화면에 표시 될 수 있는 가장 작은 단위
- 디지털 이미지는 픽셀 매트릭스로 컴퓨터에 표시
- 이미지의 각 픽셀은 정수로 저장

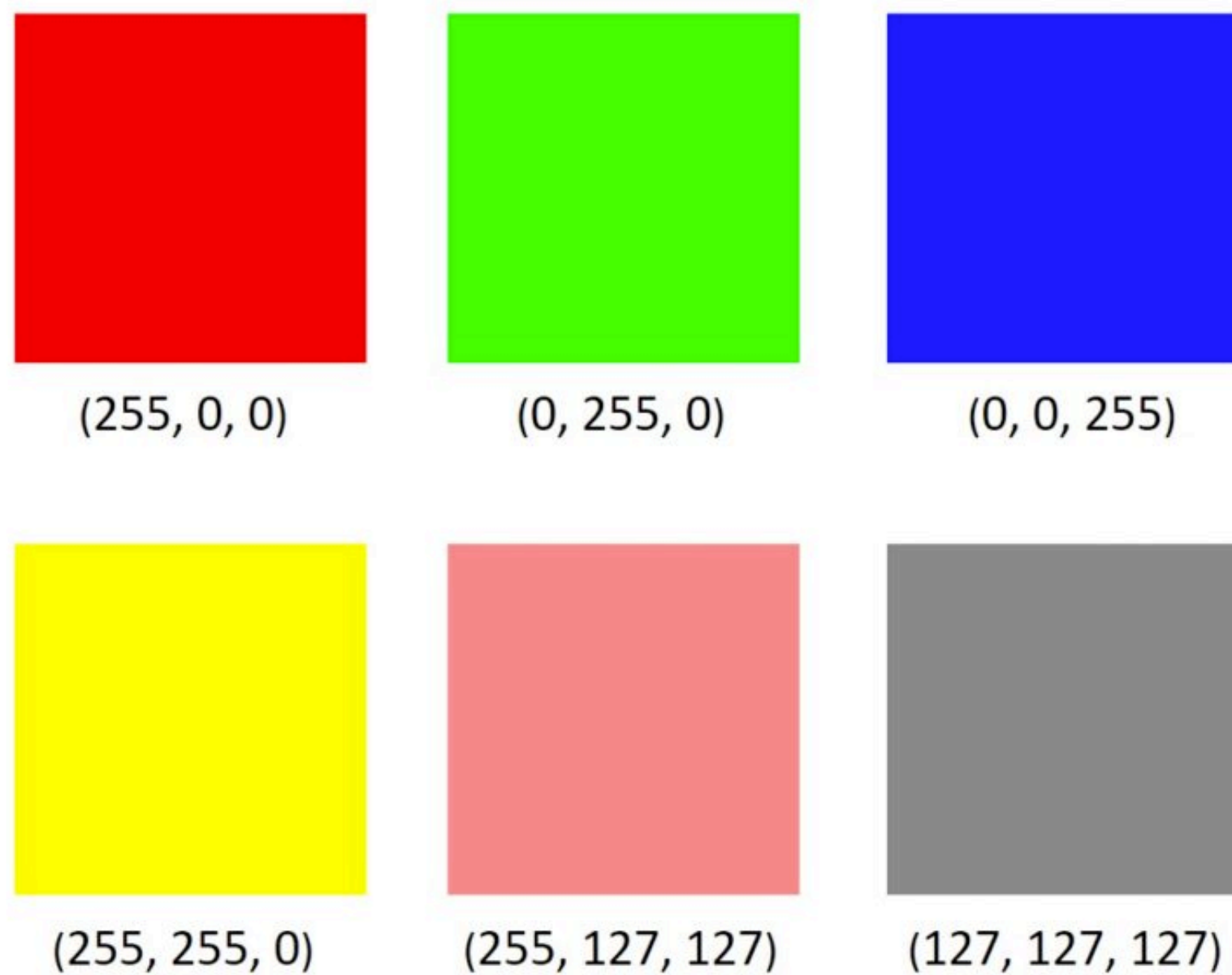
# RGB

## Red-Green-Blue

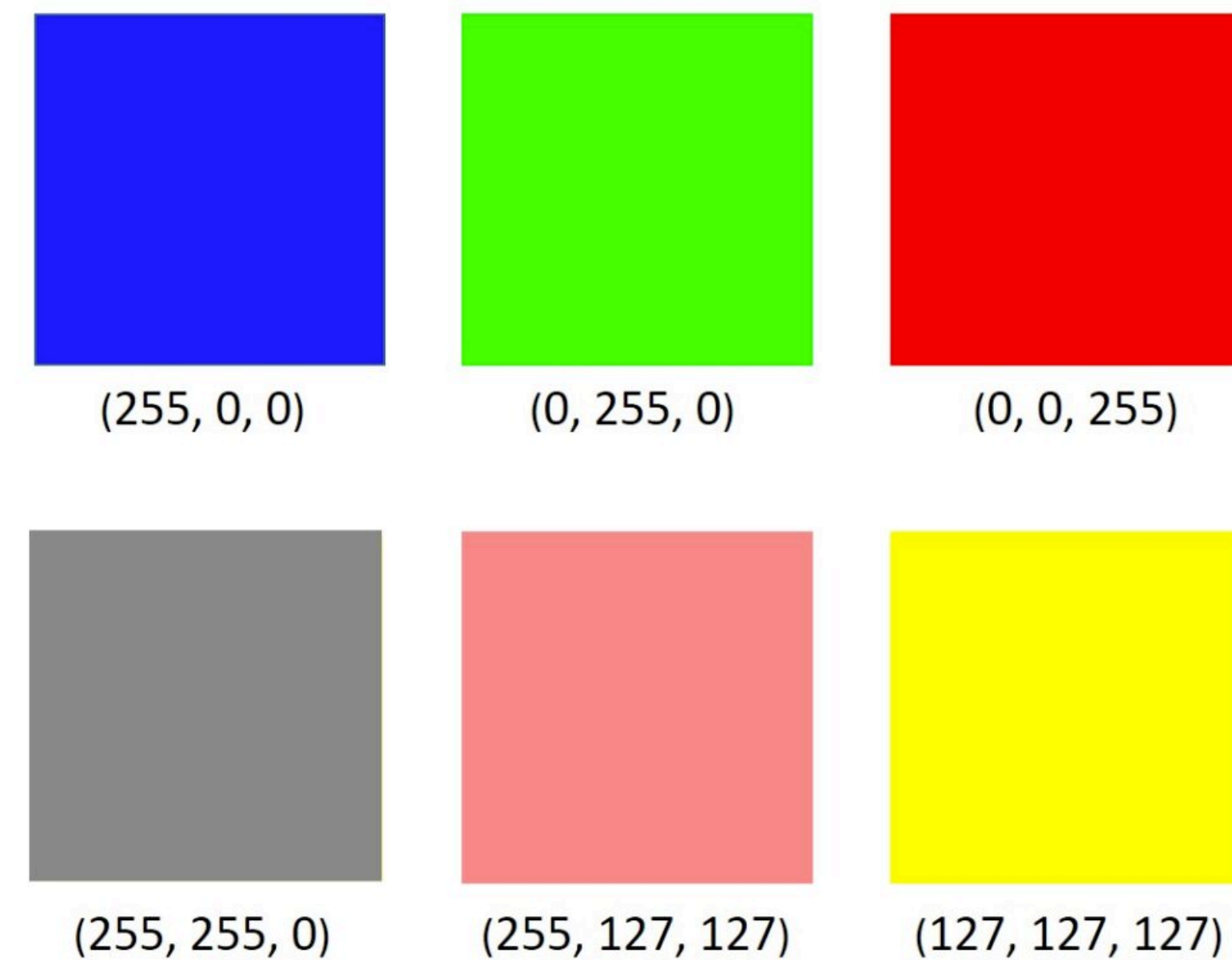
- 회색조 이미지를 처리하는 경우 0 (검은 색 픽셀)에서 최대 255 (흰색 픽셀)
  - 이 둘 사이의 숫자는 회색 음영
- 반면 컬러 이미지는 3 개의 행렬로 표현
  - 각 행렬은 채널이라고도하는 하나의 기본 색상을 나타냄
  - 가장 일반적인 색상 모델은 빨강, 녹색, 파랑 (RGB)
  - 이 세 가지 색상은 함께 혼합되어 광범위한 색상을 생성
  - OpenCV는 색상 이미지를 역순으로로드: BGR 채널



## RGB model



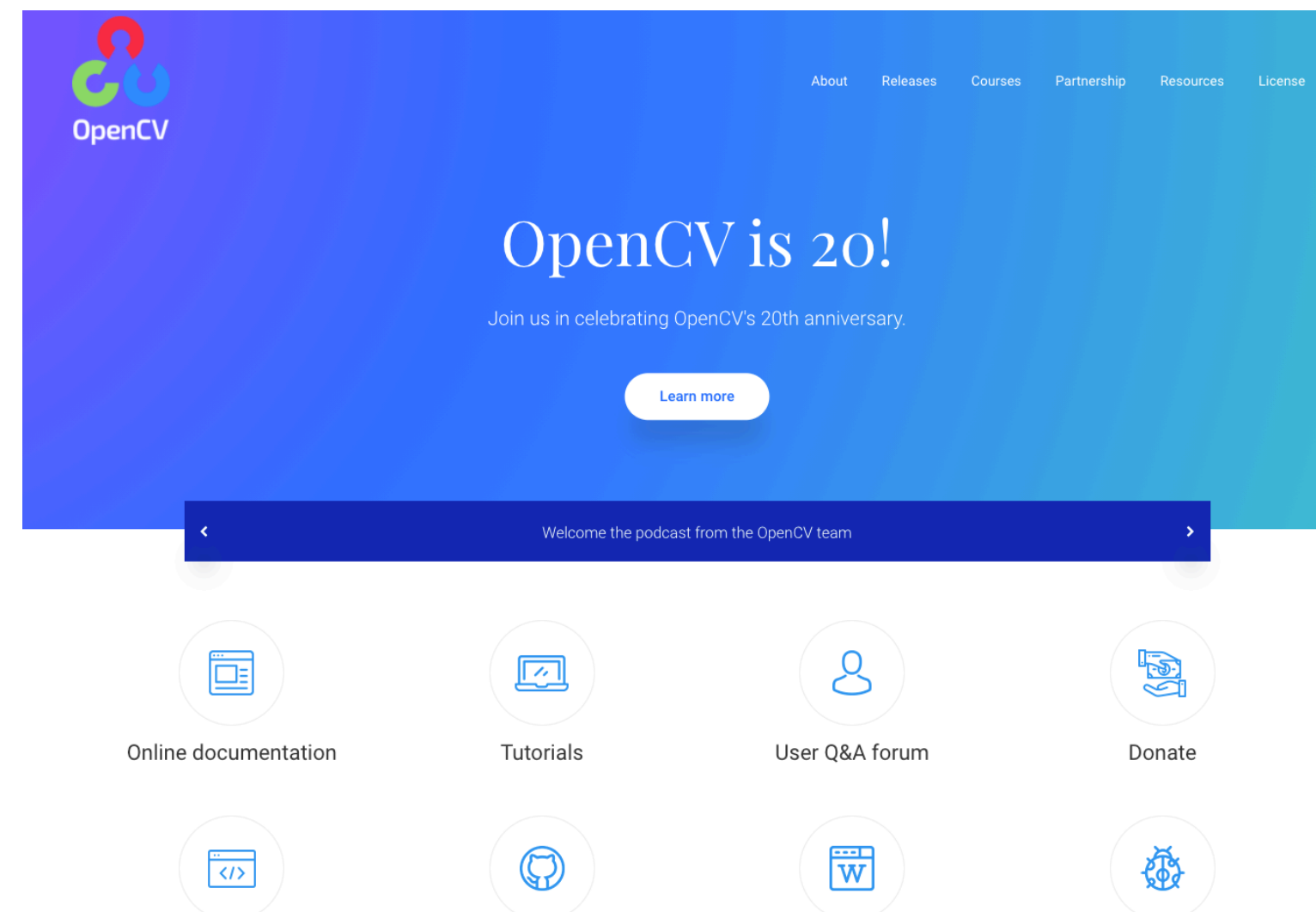
## BGR model in OpenCV



# OpenCV

## 이미지 데이터를 다루기 위한 라이브러리

- OpenCV(Open Source Computer Vision)은 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리
  - 이 라이브러리는 윈도우, 리눅스 등에서 사용 가능한 크로스 플랫폼이며 오픈소스 BSD 허가서 하에서 무료로 사용가능
  - OpenCV는 TensorFlow , Torch / PyTorch 및 Caffe의 딥러닝 프레임워크를 지원



# 파일 확장자

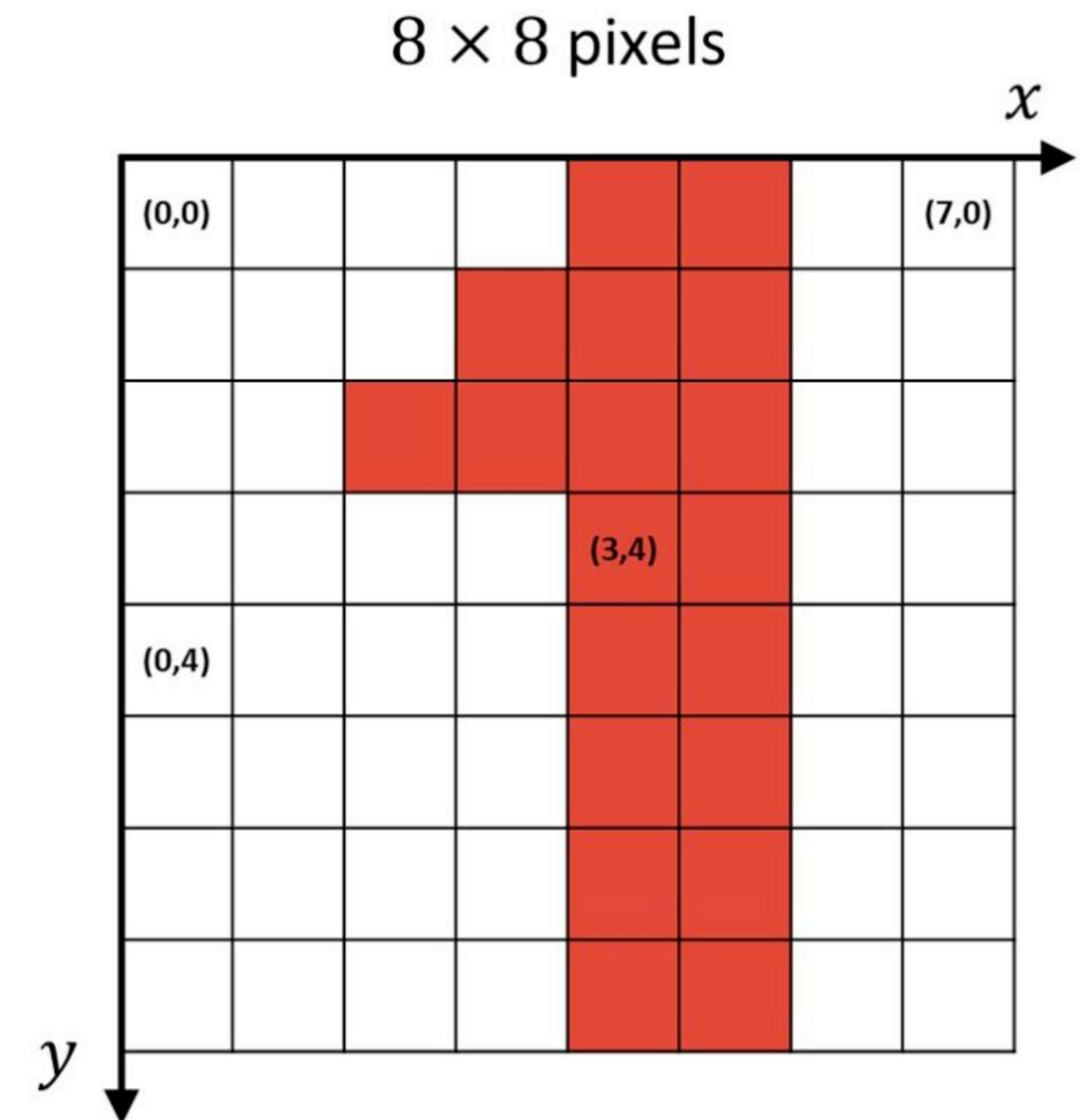
## OpenCV가 지원하는 파일 확장자의 종류

- OpenCV에서 동일한 파일 형식으로 저장되거나 전송되지 않을 수도 있음
  - 일부 파일 형식은 이미지를보다 효율적으로 표현하기 위해 다른 형식의 압축을 사용
- 종류
  - Windows bitmap (bmp, dib)
  - Netpbm – Portable image formats (pbm, pgm, ppm)
  - Sun Raster (sr, ras)
  - **JPEG (jpeg, jpg, jpe)**
  - JPEG 2000 (jp2)
  - TIFF files (tiff, tif)
  - **Portable network graphics (png)**

# 좌표계

## 컴퓨터상에서의 픽셀 좌표

- 픽셀은 두 개의  $(x, y)$  좌표로 접근
  - $x$  값은 열을 나타내고  $y$  값은 행을 나타냄.
- 이미지의 왼쪽 상단 모서리에는 원점 좌표  $(0,0)$ 
  - $x$  좌표 값은 오른쪽으로 갈수록 증가
  - $y$  좌표 값은 아래로 갈수록 증가
- 각각의 픽셀에 접근해서 조작 가능





# 비디오

## 이미지의 시퀀스

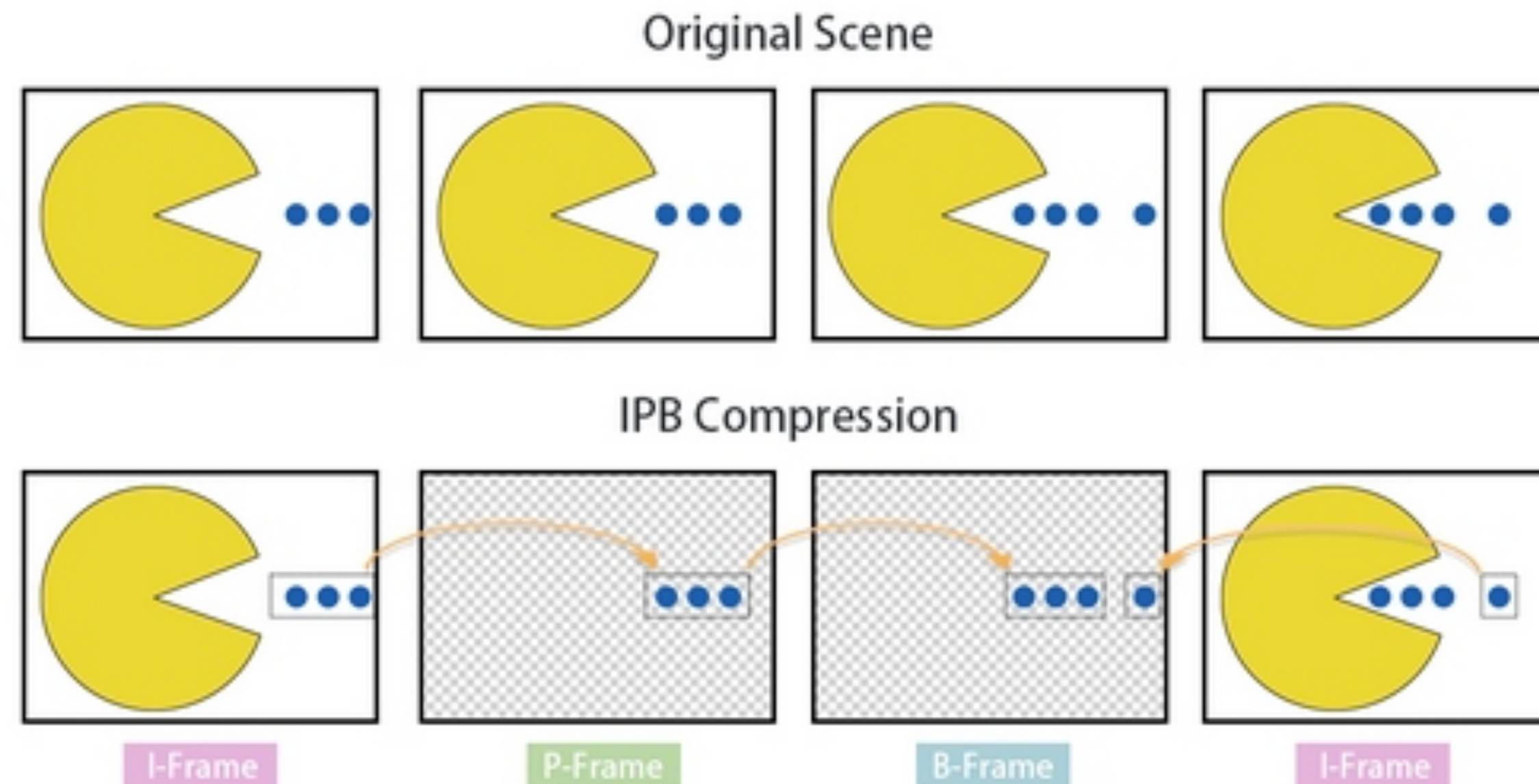
- 비디오는 연속된 이미지를 순차적으로 전환 시켜주는 효과
  - 음성은 보통 따로 저장
- 영상의 한 장면을 보통 프레임이라고 표현
  - 영화는 보통 1초에 24프레임 (24 FPS)



# 비디오 압축 방식

모든 프레임을 다 원본으로 저장해야 할까?

- 30fps의 1시간 동영상의 프레임 수 :  $3,600 \times 30 = 108,000$ 
  - frame 하나에 500KB 라고 가정하면 : 54GB
- I-Frame, P-Frame, B-Frame 등으로 구별하여, 변화되는 구간만 저장





# 얼굴 인식

## Haar feature-based cascade classifiers

- Haar feature-based cascade classifiers를 사용한 객체 감지
  - Paul Viola와 Michael Jones가 2001 년 논문 "Rapid Object Detection using a Boosted Cascade of Simple Features"에서 제안한 효과적인 객체 감지 방법




















Haar Cascade Visualization  
<https://youtu.be/hPCTwxF0qf4>

# 얼굴 인식

## Haar feature-based cascade classifiers

- 미리 트레이닝 된 XML파일을 사용

 haarcascade_eye.xml	some attempts to tune the performance	7 years ago
 haarcascade_eye_tree_eyeglasses.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalcatface.xml	fix files permissions	5 months ago
 haarcascade_frontalcatface_extended.xml	fix files permissions	5 months ago
 haarcascade_frontalface_alt.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_alt2.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_alt_tree.xml	some attempts to tune the performance	7 years ago
 haarcascade_frontalface_default.xml	some attempts to tune the performance	7 years ago
 haarcascade_fullbody.xml	Some mist. typo fixes	3 years ago
 haarcascade_lefteye_2splits.xml	some attempts to tune the performance	7 years ago
 haarcascade_licence_plate_rus_16stages.xml	Added Haar cascade for russian cars licence plate detection, 16 stage...	6 years ago
 haarcascade_lowerbody.xml	Some mist. typo fixes	3 years ago
 haarcascade_profileface.xml	some attempts to tune the performance	7 years ago
 haarcascade_righteye_2splits.xml	some attempts to tune the performance	7 years ago
 haarcascade_russian_plate_number.xml	Create haarcascade_russian_plate_number.xml	6 years ago
 haarcascade_smile.xml	fixing models to resolve XML violation issue	3 years ago
 haarcascade_upperbody.xml	Some mist. typo fixes	3 years ago

**E.O.D**