2024년 가을학기

Distributed Systems

HW 1. Load Balancer

남 재 현 (namjh@dankook.ac.kr)

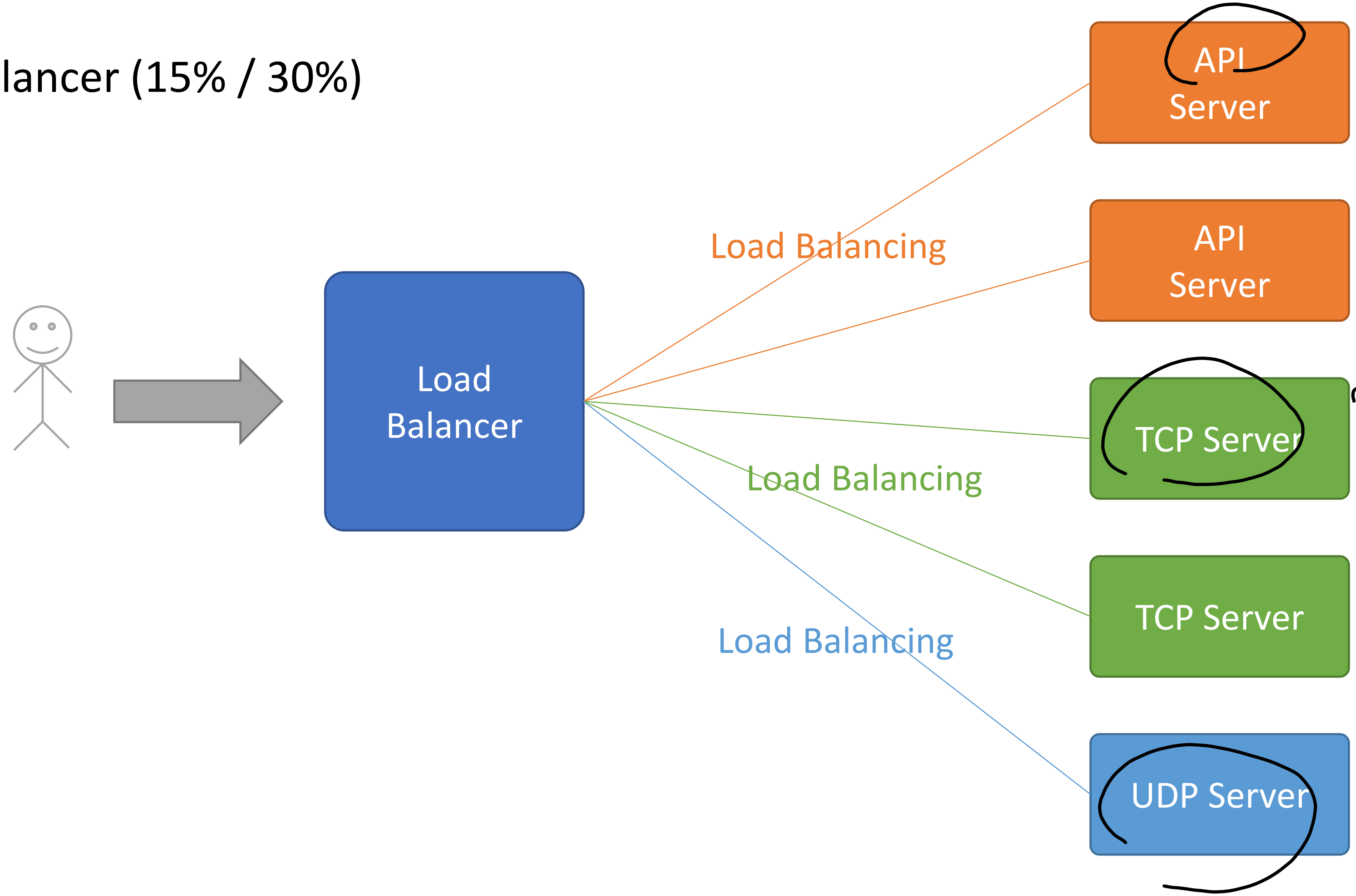
SW융합대학 컴퓨터공학과

단국대학교 DANKOOK UNIVERSITY BoanLab Networked Systems and Security

Overview

→ 로드밸런서 만들기.

- L3/4 Load Balancer (15% / 30%)



API ″

TCP load balan

UDP ″

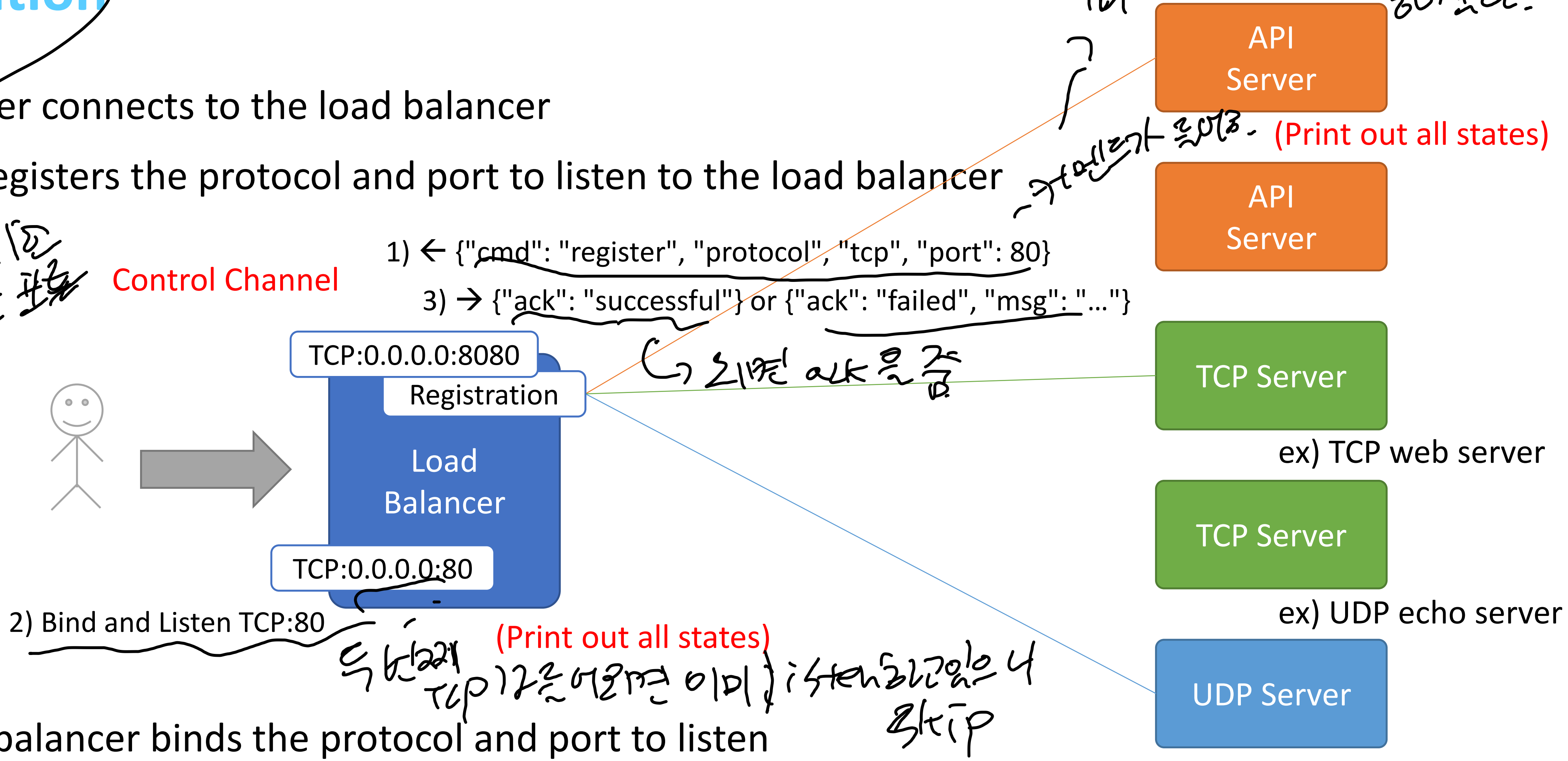
Registration

TCP의 80번 포트는 load balanc 하고있다.

- Each server connects to the load balancer
- Then, it registers the protocol and port to listen to the load balancer

클라이언트가 들어옴.

[illegible]



Control Channel

1) ← {"cmd": "register", "protocol", "tcp", "port": 80}

3) → {"ack": "successful"} or {"ack": "failed", "msg": "..."}

↳ 되면 ack를 줌

(Print out all states)

ex) TCP web server

ex) UDP echo server

2) Bind and Listen TCP:80

(Print out all states)

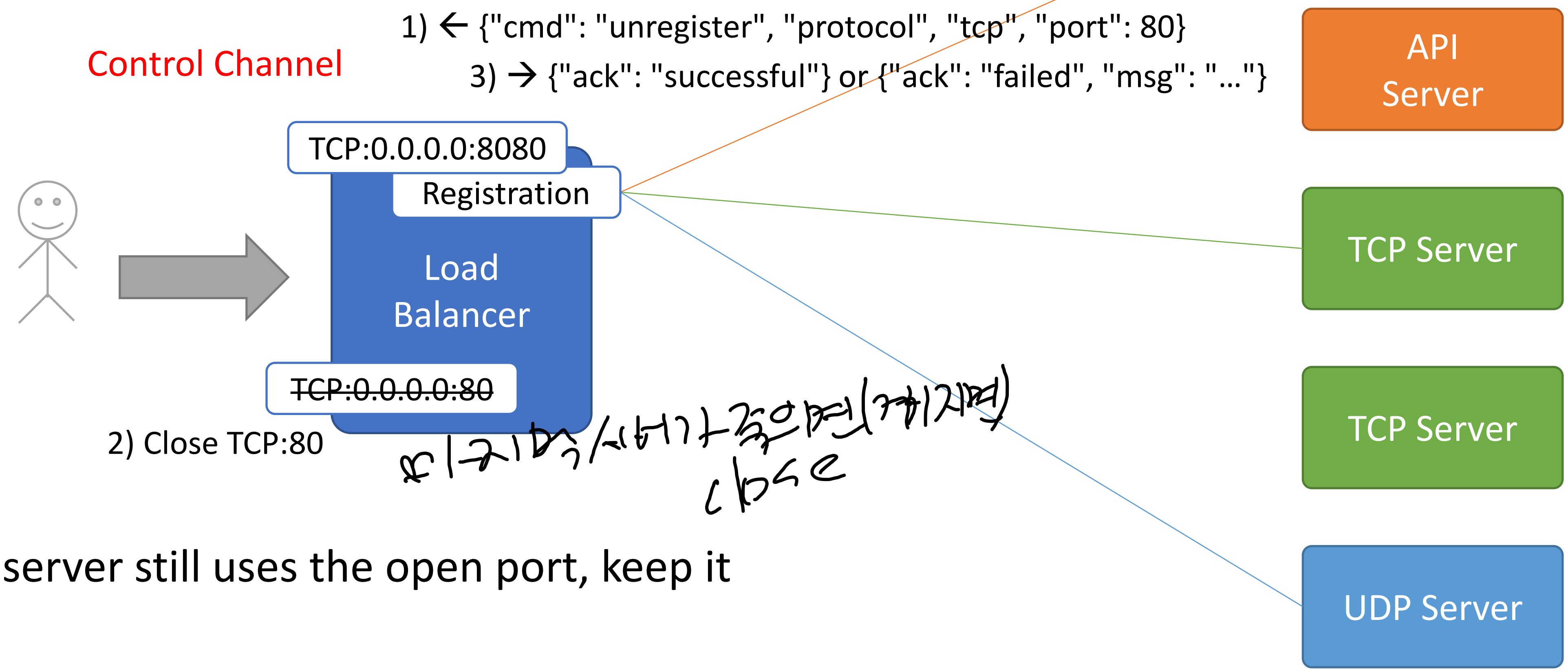
두번째 TCP가 들어오면 이미 listen되고있으니 skip

- The load balancer binds the protocol and port to listen
- Then, the load balancer relays the traffic to the corresponding server(s)

Unregistration

- 서버를 끄면 unregistere 해제한다.

- When a server is terminated (the control channel is closed), by default, the load balancer cleans up the open port as well



Control Channel

1) ← {"cmd": "unregister", "protocol", "tcp", "port": 80}

3) → {"ack": "successful"} or {"ack": "failed", "msg": "…"}

2) Close TCP:80

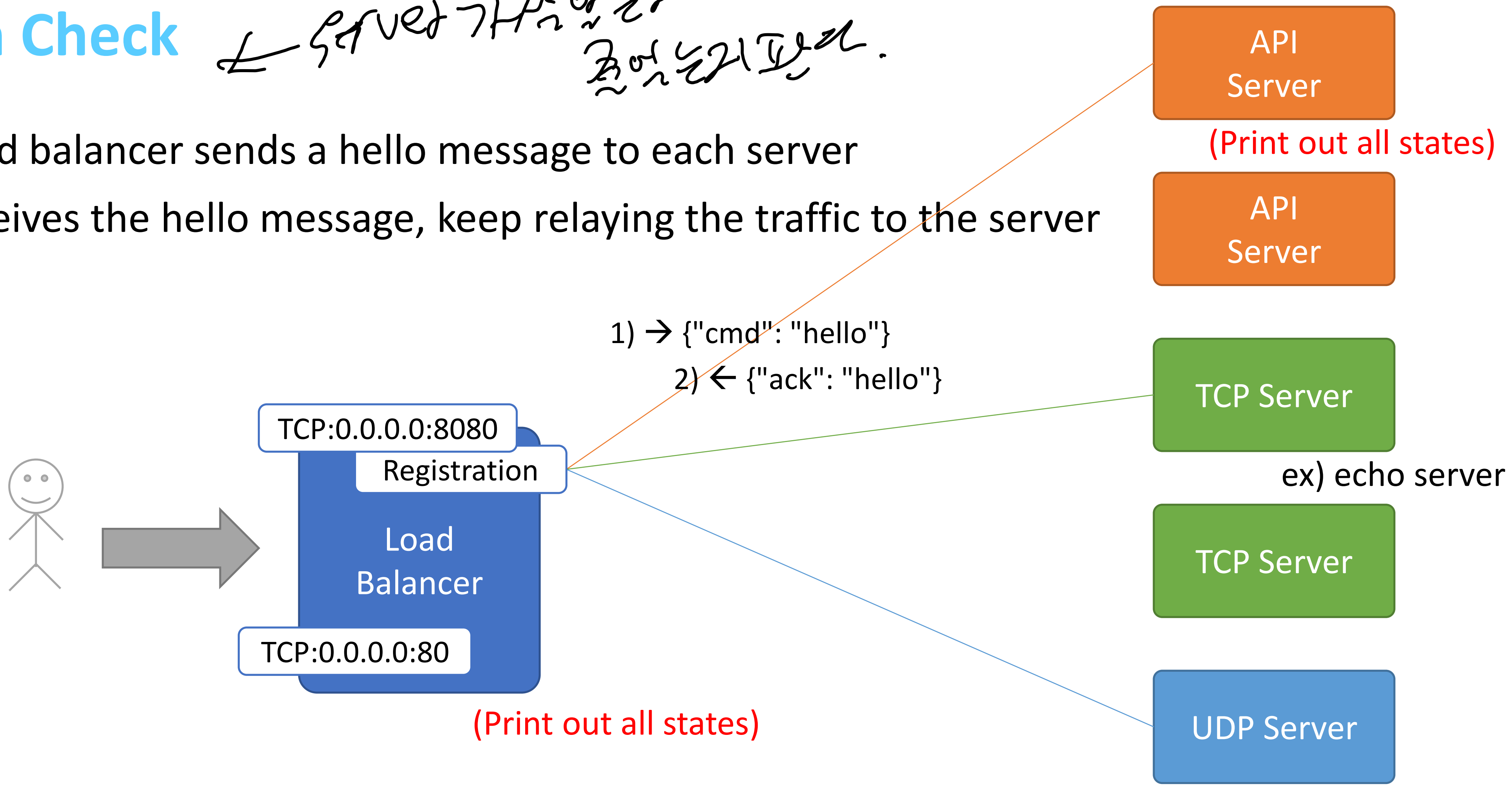
위 과정/서버가 죽으면(꺼지면)
close

- If another server still uses the open port, keep it

Health Check

← server가 살았는지 죽었는지 판단.

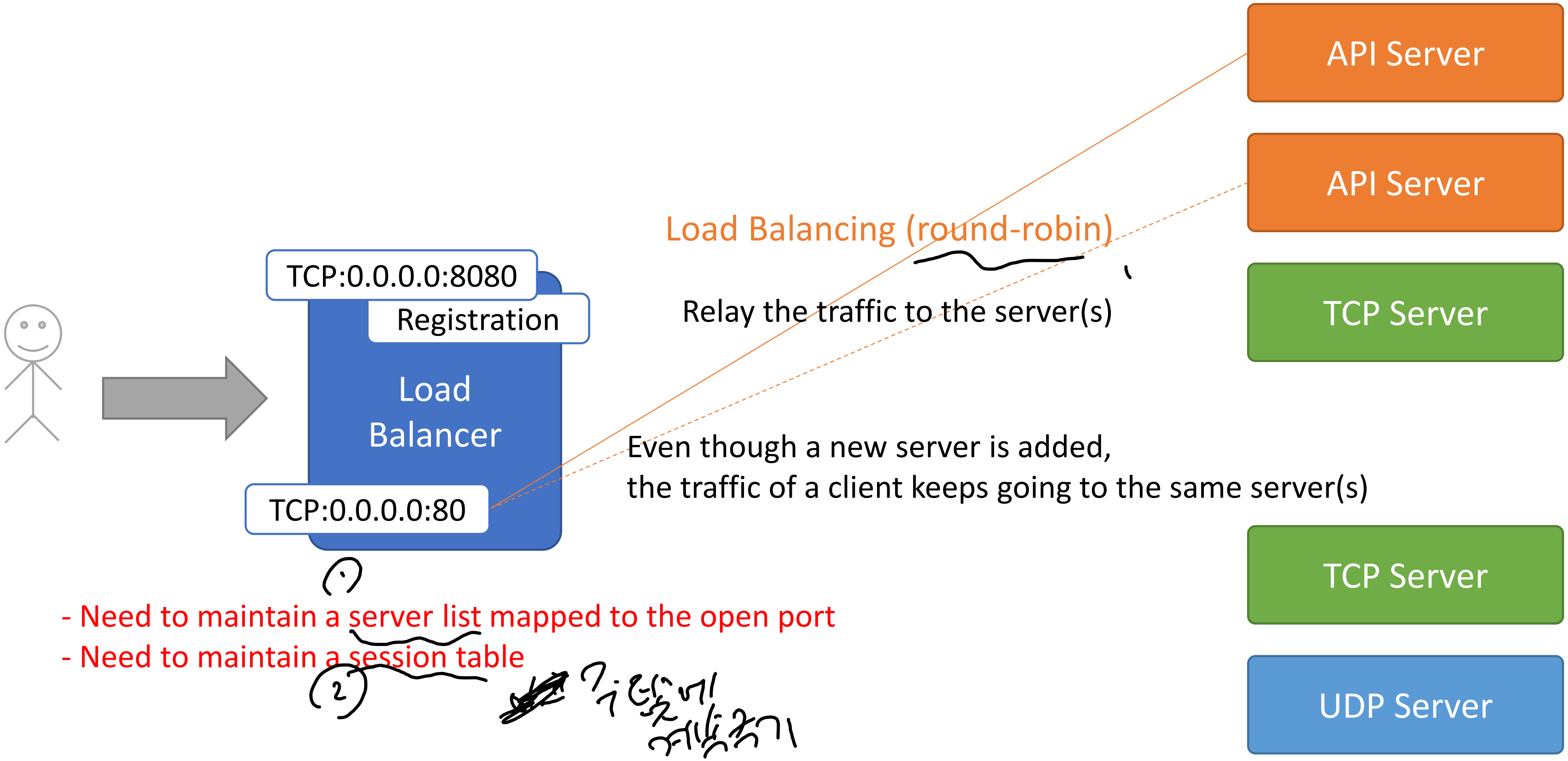
- The load balancer sends a hello message to each server
- If it receives the hello message, keep relaying the traffic to the server



- If not (timeout), clean up the connection to the server and stop relaying the traffic to the server
- If there is no server to relay the traffic, clean up the open port

Load Balancing

- A client connects to the open port, relay the traffic to the corresponding server(s)



Load Balancing (round-robin)

Relay the traffic to the server(s)

Even though a new server is added,
the traffic of a client keeps going to the same server(s)

① - Need to maintain a server list mapped to the open port

② - Need to maintain a session table

기억해
저장하기

Development Environment

- OS
  - Ubuntu basis

- Programming Language
  - C/C++
  - Java
  - Python
  - Go

Document

Document should look professional
- Be neat and clean
- Be easy to read
- Do not simply export a doc from Notion

- Introduction 서론 -
  - Brief description of the assignment
- Implementation
  - 구조적으로 표현.
  - How to implement the assignment from the **architectural view** (not code level) 왜 이렇게 만들었나.
  - **Explain why** you took such a way to implement the assignment
- Verification
  - How to set up the test environment ← 환경
    - Should be specific (e.g., how to install dependency packages)
  - How to compile and run the assignment
    - Should be specific (e.g., exact commands to compile the assignment, something I need to provide)
  - How you verified the functionality of the assignment
    - Screenshots showing that the assignment is working properly ← 캡쳐.
  - How you tested the performance of the assignment
    - Graphs and tables showing how much the assignment performs well (compared to what?)

성능 그래프.

Submission

- Upload the source code and the document to the e-learning system
  - [학번].zip
    - 예시) 32123456.zip
  - [학번].pdf
    - 예시) 32123456.pdf
- Due Date
  - 10:30AM on Nov. 31st, 2024
    - Right before the class

Q & A