

## 1장. AWS 탄생

### ▷ 제프 베이조스

- 아마존 창립자
- 2021년 기준 세계 부자 순위 2위
- 프린스턴 대학 물리학(양자역학 포기)  
전공이지만 포기하고 컴퓨터 공학과로 전과한 다음 졸업
- 뱅커스 트러스트 최연소 부사장(26세)
- 1994년 인터넷 서점 아마존 닷컴 창립(주변지인에게 200만 달러 투자받음)
- 종합쇼핑몰로 변신

평소 서비스 운영할 때 필요한 서버의 수에서 블랙프라이데이 등 서비스 사용자 폭증을 감당하기 위해 서버를 추가 구축하면서 잉여 자원이 발생함. 잉여 자원을 외부에 서비스 하면서 클라우드 서비스를 시작하게 됨.

## 2장. AWS 회원가입

- ▷ 준비물 : 해외결제 가능한 신용카드 및 직불카드
- ▷ 회원가입 - 우 상단 Region을 서울로 변경(AZ - Available Zone)

### ● EC2 서버 임대하기(Amazon Elastic Compute Cloud)

#### ① 인스턴스 시작

- 이름 : aws-v1
- 우분투(Ubuntu) 운영체제 선택
- 버전 24.04 프리티어 가능한 거

#### ② 키 페어 생성 ex> zzzmini-aws-key

#### ③ 키 페어 유형 - RSA

#### ④ 프라이빗 키 파일 형식 - pem → mobaX-term 접속 시 필요

#### ⑤ 로컬에 aws-key 폴더 생성 후 → 키 저장

#### ⑥ 보안그룹 그냥 체크 → 나머지도 디폴트로 그냥 체크

▶ ssh(secured Shell) 포트 22번 -- 터미널로 접속해서 명령어를 전송할 수 있는 포트

#### ⑦ 스토리지 구성 - 밑에 최대 허용치까지 입력

#### ⑧ 인스턴스 시작

#### ⑨ 모든 인스턴스 보기

#### ⑩ 실행 중으로 뜨면 EC2 서버 생성 완료

### 3장. EC2 서버 윈도우에서 접속하기

#### ● MobaXterm 다운로드 받기([www.mobaxterm.mobatek.net](http://www.mobaxterm.mobatek.net))



- ① 무료버전 : Download - Home Edition
- ② 다운로드 후 압축 풀고 설치 → 주욱 넥스트 → 많이 사용할 툴

#### ● AWS EC2 에 원격 접속하기

- ① EC접속 : MobaXtem 프로그램 실행 → Session 클릭 → SSH(Secured Shell) - 보안이 적용된 셸 → 리모트 호스트(AWS 로그인 후 생성한 EC2 인스턴스 - 인스턴스 정보 누르면 아래에 정보가 뜨는데 거기서 퍼블릭 IP v4 주소 복사 붙여넣기) → special username → ubuntu (애는 고정) → 포트 22번
- ② 밑에 Advanced SSH settings  
Use private key → 이곳에 먼저 생성한 aws-key를 찾아서 넣어줌.  
→ OK → 접속  
> pwd ; print working directory

## 4장. EC2 서버와 방화벽

로컬 컴퓨터에서 AWS EC2로 접속하기 위해서는 보안이 설정되어 있는 SSH(Secure Shell)을 이용하여 접속하며, 전체 가능 포트(0~65536) 중 특정 포트를 통해서만 접속하도록 설정해야 함. AWS EC2에서는 인바운드 규칙을 통해 접속 가능한 포트를 설정함.

### ① shell 이란?

하드웨어를 직접 제어하기 위해서 운영체제가 필요한데 사람이 OS에게 명령을 내려야 하는데 이때 사용하는 명령어가 shell 이다. 즉, 사람이 OS에게 명령을 내리는 명령어 집합이 Shell이다.

▶ SSH(Secure Shell) - 명령어가 encoding 되서 날아감...

▶ 추후 다른 포트들도 용도에 맞게 개방해야 함.

- 8080, 80, 3306(mysql) 등...

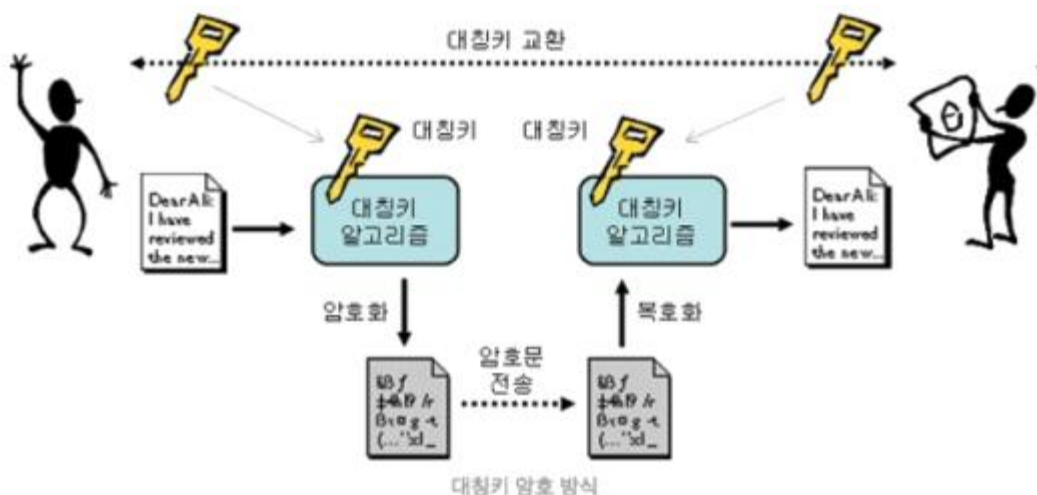
우리는 EC2 라는 빈 컴퓨터 인프라를 받아서 사용하는 것이다. 그래서 22번 포트를 개방해주는 것이며 우리가 원격으로 접속해서 Tomcat, JDK 등을 설치해서 사용하는 것이다.

## ● RSA란?

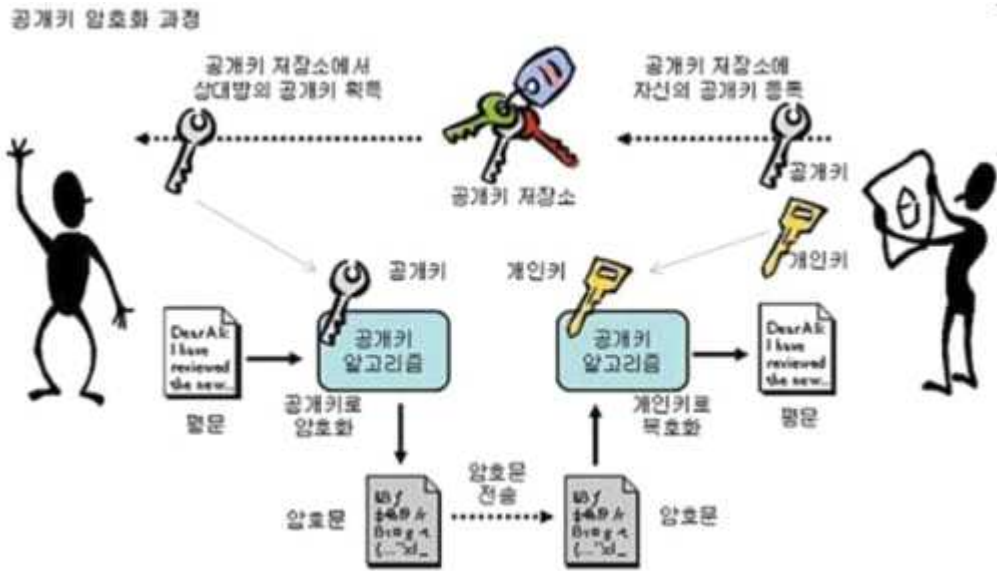
Use private key → 여기에 생성한 aws-key를 찾아서 넣어줌. 애가 뭘까?

### ① 대칭키

일반적으로 보안 및 인증을 위해 대칭키 교환방식을 사용하는데, 열쇠를 교환하기 힘들고, 탈취의 위험이 있어서 사용하지 않는다.



## ② RSA : 공개키 암호화방식(비대칭키 암호화방식)



그래서 키를 두 개 만들어 사용하게 됨.

A 열쇠 -> 잠금 -> B열쇠로 풀 수 있다.

B 열쇠 -> 잠금 -> A열쇠로 풀 수 있다. 이런 방식을 공개키 기반 암호화 방식  
(A열쇠로 잠그면 B 열쇠로 풀 수 있고, B열쇠로 잠그면 A열쇠로 풀 수 있다.)

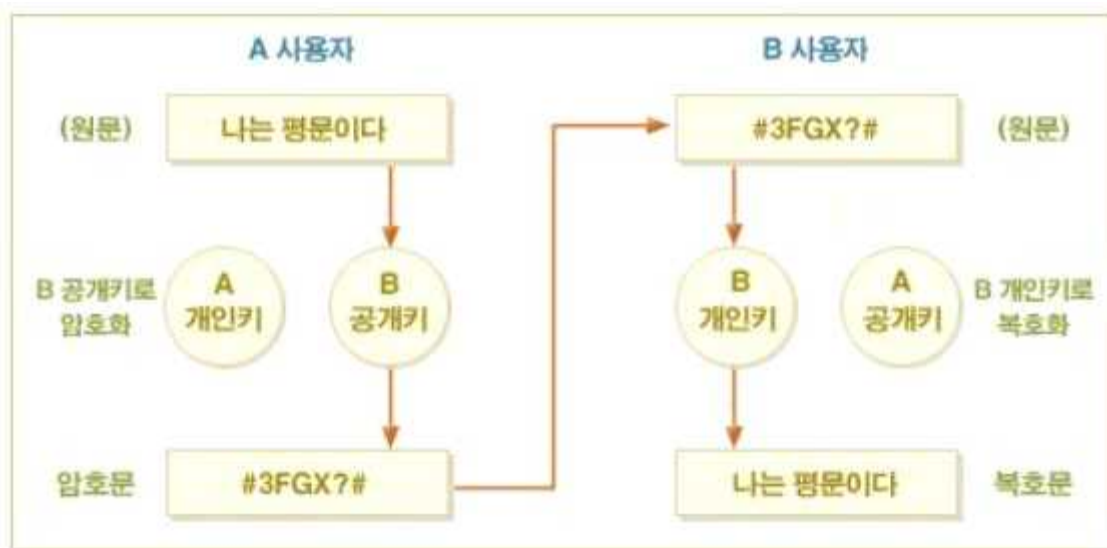
데이터를 암호화 할 때 A가 B에게 보낼 때 A의 비밀키로 잠그면 누구든 다 풀 수 있다. 모두 A의 공개키를 갖고 있기 때문, A의 공개키로 잠그면 누구도 열수 없다. A의 비밀키는 A만 갖고 있다. 그래서 B의 공개키로 잠가서 보내면 B만 열수 있다. C는 B의 비밀키가 없다.

A가 B의 공개키로 보낼 경우 C가 탈취해서 포함되어 있는 B의 공개키를 폐기하고 다른 키를 넣어서 B의 공개키로 암호화해서 보내면 B는 틀렸다고 하고 A에게 다시 요청하는 것이 반복되는 악의적 상황이 발생 함.

그래서 데이터를 누가 보냈는지도 필요함

※ 처리절차

- 1) A가 B에게 자료를 보낼 때 B의 공개키로 암호화 시킨 후 한번 더 A의 개인키로 감싼다.
- 2) B가 자료를 받으면 A의 공개키로 열어본다. → 서명 검증
- 3) A가 보낸 자료가 맞으면 B 자신의 개인키로 자료를 복호화 한 후 읽는다.



③ EC2 서버의 공개키 확인.

- /ubuntu/ 현재 폴더에서

> ls -al 하면 .ssh 라는 숨김 폴더가 있다.

> cd .ssh

> ls -al

여기에 authorized\_keys 가 있는데 애가 공개키다. 그리고 우린 아까 나의 비밀키를 내 컴에 저장해 뒀다.

내 비밀키로 잠가서 보내면 EC2 서버는 나의 공개키로 열어서 내가 접속한 것임을 확인한다. 그러면 세션이 만들어 진다.

## 5장. 리눅스 명령어

▷ MobaXterm 에서 EC2의 우분투 운영체제에 접속하여 테스트

- > clear : 화면 깨끗이 정리
- > pwd : 현재 경로 알려줌. (Print working directory)
- > cd (change directory)
- > cd .. : 상위 폴더로 이동
- > cd 경로 : 해당 절대 경로로 이동
- > cd / : root 로 이동
- > cd /home : 루트 아래의 home 디렉토리로 이동
- > cd /ubuntu : 로그인 사용자의 기본 폴더로 이동
- > cd ~ : 로그인 사용자 폴더로 이동

- ls 현재 위치의 파일과 폴더를 보여줌.

```
ubuntu@ip-172-31-32-134:/$ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
```

- 자세히 보려면 ls -l

```
ubuntu@ip-172-31-32-134:/$ ls -l
total 64
lrwxrwxrwx  1 root root    7 May 16 02:08 bin -> usr/bin
drwxr-xr-x  4 root root 4096 May 16 02:12 boot
drwxr-xr-x 15 root root 3180 Sep  2 02:23 dev
drwxr-xr-x 94 root root 4096 Sep  2 02:23 etc
drwxr-xr-x  3 root root 4096 Sep  2 02:23 home
lrwxrwxrwx  1 root root    7 May 16 02:08 lib -> usr/lib
```

### d rwx r-x r-x ?

- d 는 디렉토리, - 는 파일 ,l 은 symbolic link를 뜻함.

두 번째 글자부터 권한에 대한 정보를 나타내구요.

r 은 읽기(read), w 는 쓰기(write), x 는 실행(execute)을 해당 위치의 - 는 해당 권한 없음을 의미해요.

그래서 drwx의 경우 디렉토리의 읽기, 쓰기, 실행 권한 이 있음을 의미합니다.

그런데 맨 앞의 d 뒤에 rwx rwx rwx 가 이렇게나 붙습니다.

첫 번째 rwx는 소유자에 대한 퍼미션, 두 번째 rwx는 소유 그룹에 대한 퍼미션, 세 번째 rwx는 그 외 모든 사용자에게 대한 퍼미션입니다.

이 퍼미션을 숫자로 표현하는데 각 문자를 2진법으로 표현합니다.

각 퍼미션 기호를 숫자로 변환하면

$r = 4 / w = 2 / x = 1 / - = 0$

로 표현됩니다.

예시로 drwxr-xr-x를 보면, 디렉토리의 소유자의 경우 rwx 권한이 있으며 rwx는 4 2 1, 이 숫자를 합하면  $4+2+1 = 7$  이 됩니다. 그리고 소유 그룹은 r-x 권한이 있으며 r-x는 4 0 1, 이 숫자를 합하면  $4 + 0 + 1 = 5$  가 됩니다. 마지막으로 그 외 모든 사용자도 r-x로 소유 그룹과 동일한 5가 됩니다.

그래서 권한, 퍼미션을 이해하기 쉽게 10진법으로 777, 755라고 말합니다.

▷ --help 모든 명령어의 사용법을 알려줌.

> ls --all 또는 > ls -a

▷ 숨김파일 → OS의 중요한 파일로 형식은 .파일명 형태로 쓴다.

> ls -al : 모든 파일을 자세히 볼 것이다.

▷ 디렉토리 만들기

> mkdir user

※ CLI(Command Line Interface)는 응답이 없으면 잘 된 것이다.

▷ 빈 파일 생성하기

> touch a.txt

> ls -l

▷ 파일 삭제하기

> rm --help

-f : 파일 강제 삭제

-r : 디렉토리 내에 파일이 있어도 모두 삭제 후 폴더 삭제

▷ 실행 중지 ctrl + C → 취소

> pwd

> cd /home/ubuntu/user

> touch b.txt

```
> ls
> cd ..
> rm a.txt
> ls
> rm user 삭제 못함. rm -f user 삭제 안됨.
> rm -r user → 삭제하고자 하는 폴더 내 모든 파일을 삭제한 후 폴더를 지움.
```

#### ▷ 파일 옮기기

```
> cd ~ → ubuntu로 이동
> ls
> touch a.txt
> ls
```

#### ▷ 파일 복사하기

```
> cp a.txt b.txt
> ls
> mkdir user
> mv b.txt user
> ls
> mv a.txt /user/c.txt → 이동과 파일명 변경이 가능하다.
> cd user
> ls
> mv c.txt d.txt → 파일명 변경
> cd ..
> rm -r user
> clear
```

#### ▷ 연결파일(Soft Link) 만들기(윈도우의 바로가기 아이콘)

```
> touch a.txt
> ls
> ln -s a.txt a-link.txt
> ls
> a-link.txt -> a.txt
```



- cd /bin ---> bin 폴더는 실행파일이 모여 있는 곳이다. 여기에 보면 soft link 걸린 파일들을 볼 수 있다.

## ● apt 명령어의 이해

신규 프로그램의 설치를 할 때 다운로드 이외의 다양한 내용을 처리해야 하는데 이를 간단히 처리할 수 있도록 제공하는 것이 우분투 apt 명령어이다.

### ▷ 프로그램의 다운로드 방법

1. 우분투 레포지토리 접근 → 설치가능한 모든 프로그램의 목록이 들어 있으며, 메뉴판 정도로 생각하면 된다.

/etc/apt/sources.list → 레포지토리에 경로 등록 (ex http://hello.com)

2. apt update

→ apt 목록을 갱신해 줌. 즉 메뉴판에 프로그램 목록을 갱신(권한 없음)

→ sudo apt update : root 권한으로 실행

3. apt install 프로그램명 → 해당 프로그램을 설치

- 다운로드 → 자동설치 - 윈도우 더블클릭해서 설치하는 것과 같음.

4. 새로운 프로그램이 나왔는데 우분투 측에서 아직 동기화를 해 놓지 않았을 경우

직접 해당 사이트에 접속해서 다운로드 받고 설치, 그러나, 좋은 방법은

5. PPA(개인저장소) 저장소에서 검색해 보면 웬만한 것은 다 등록되어 있기 때문에 쉽게 찾을 수 있다. -- 개인저장소는 말 그대로 우분투를 사용하는 개인사용자들의 저장소

6. 애네들은 /etc/apt/sources.list.d/ 디렉토리에 저장해 놓을 수 있다.

신규 프로그램의 사이트를 위의 디렉토리에 등록만 해 놓으면 끝이다.

7. 요기(/etc/apt/sources.list.d/)에 넣어 놓으면(주소등록) apt update 할 때 애네들꺼 까지 자동 갱신되어 등록된다.

## ● apt 명령어의 실습

> clear

> apt-cache search tomcat → 메뉴판 목록 조회 → Apache Tomcat 10 있음.

- 파이프 라인 『|』 → 결과가 파이프 라인을 거쳐서 새로운 명령의 입력이 된다.

- grep 은 뭔가를 찾는 명령

> apt list 이렇게 보면 너무 많다.

> apt-cache search tomcat | grep tomcat

```
ubuntu@ip-172-31-32-134:/etc/apt$ apt-cache search tomcat | grep tomcat
libtomcat9-embed-java - Apache Tomcat 9 - Servlet and JSP engine -- embed libraries
libtomcat9-java - Apache Tomcat 9 - Servlet and JSP engine -- core libraries
libtomcatjss-java - JSSE implementation using JSS for Tomcat
tomcat-jakartaee-migration - Apache Tomcat migration tool for Jakarta EE
tomcat9 - Apache Tomcat 9 - Servlet and JSP engine
tomcat9-admin - Apache Tomcat 9 - Servlet and JSP engine -- admin web applications
tomcat9-common - Apache Tomcat 9 - Servlet and JSP engine -- common files
tomcat9-docs - Apache Tomcat 9 - Servlet and JSP engine -- documentation
tomcat9-examples - Apache Tomcat 9 - Servlet and JSP engine -- example web applications
tomcat9-user - Apache Tomcat 9 - Servlet and JSP engine -- tools to create user instances
ubuntu@ip-172-31-32-134:/etc/apt$
```

> clear

> apt list | grep tomcat ---> tomcat 만 빨간색으로 뽑아서 보여줌.

> apt update

```
ubuntu@ip-172-31-32-134:/etc/apt$ apt list | grep tomcat
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

libtomcat9-embed-java/jammy-updates 9.0.58-1ubuntu0.1 all
libtomcat9-java/jammy-updates 9.0.58-1ubuntu0.1 all
libtomcatjss-java/jammy 8.0.0-2 all
tomcat-jakartaee-migration/jammy 0.2.0-1 all
tomcat9-admin/jammy-updates 9.0.58-1ubuntu0.1 all
tomcat9-common/jammy-updates 9.0.58-1ubuntu0.1 all
tomcat9-docs/jammy-updates 9.0.58-1ubuntu0.1 all
tomcat9-examples/jammy-updates 9.0.58-1ubuntu0.1 all
tomcat9-user/jammy-updates 9.0.58-1ubuntu0.1 all
tomcat9/jammy-updates 9.0.58-1ubuntu0.1 all
ubuntu@ip-172-31-32-134:/etc/apt$
```

권한 없음.

리눅스의 최상위 관리자는 root 이다. 나는 지금 누구?

> whoami

> ubuntu → 권한이 없으므로 sudo 로 실행(super user)

> sudo apt update

> apt-cache search tomcat

> apt-cache search tomcat | grep tomcat → 톱켓만 자세히

▷ 설치할 때도 super user로

> sudo apt install tomcat9

설치할 때 이런저런 용량 쓸거냐 물어본다. 여기서도 일단 n 때려주고, 입력 시 옵션으로 모두 설치를 넣을 수도 있다.

> sudo apt install -y tomcat9

▷ 톱켓이 잘 설치되어 있는지 확인하려면 포트를 잡았는지 보면 된다.

▷ netstat 를 통해 확인해 본다. 그런데 설치가 안되어 있음.

> sudo apt install net-tools

> clear  
> netstat -nlpt

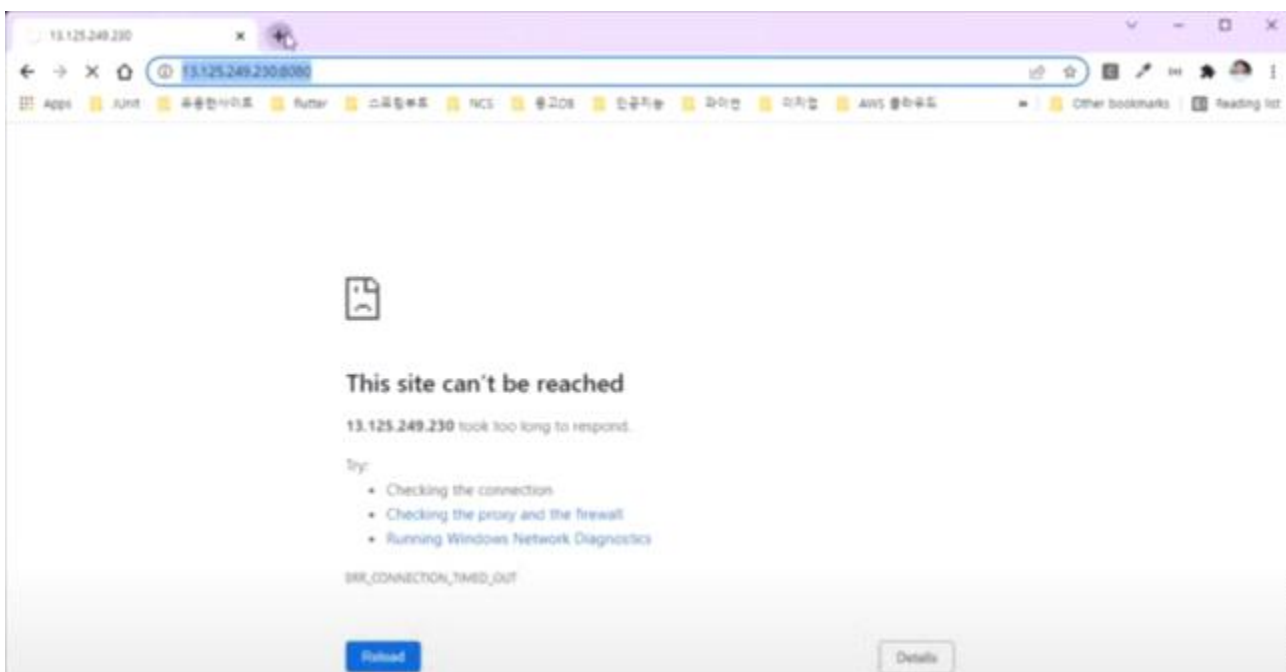
```
ubuntu@ip-172-31-42-251:~$ netstat -nlpt
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::8080                :::*                    LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
tcp6       0      0 :::1:6010              :::*                    LISTEN      -
ubuntu@ip-172-31-42-251:~$
```

## ▶ 테스트

① 크롬 열고...

② EC2 서버 아이피에 포트번호 붙여서 확인하면

ex> 131.125.249.250:8080



※ 방화벽에 8080 포트가 막혀있어서 fail됨

## ● 인바운드 규칙 수정

① aws로 접속해서 로그인 함.

② 좌측 인스턴스 클릭해서 우측에 보면 내 아이피 보임

③ 아래에 보안 탭 선택 → 인바운드 규칙(어떤 포트 열어줄까) → 인바운드 규칙 바로 위에 보안그룹 눌러주고 → 인바운드 규칙편집

④ 사용자지정 TCP / 8080 / Anywhere → 규칙저장

⑥ 다시 크롬에서 접근해 보면



※ 톰캣이 잘 돌아가고 있다...

## ● ppa 저장소 사용(ppa : personal package archive) -- 개인저장소

> netstat -nlpt

```
ubuntu@ip-172-31-42-251:~$ netstat -nlpt
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010         0.0.0.0:*               LISTEN      -
tcp6       0      0 :::8080                 :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::1:6010               :::*                    LISTEN      -
ubuntu@ip-172-31-42-251:~$
```

8080에서 톰캣 돌고 있음

> sudo apt-cache search tomcat 메뉴판에 있는 톰캣의 목록 버전 보여줌

※ 톰캣 10 삭제하고 톰캣 8 설치하고 싶음

삭제 두 가지 방법

1. sudo apt remove tomcat10 (설정파일 안지움)
2. sudo apt --purge remove tomcat10 (설정파일까지 지움)

> sudo apt --purge remove tomcat10

삭제할까요? 네 ... 잘 삭제 됨..

> netstat -nlpt → 8080 안보이면 잘 삭제된 거임.

### ① 톰캣 9 설치

> sudo apt install tomcat9

```
ubuntu@ip-172-31-32-134:/etc/apt$ sudo apt install tomcat8
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package tomcat8
ubuntu@ip-172-31-32-134:/etc/apt$
```

---- 설치 안됨.. 없으니까

---- 우리 메뉴판에 없음...

---- 그럼 우리가 저장소에 추가해서 설치해야함.

② 크롬 열고 → launchpad ubuntu ppa 검색 → Personal Package Archives : Ubuntu - Launchpad 있음...

③ 해당 사이트 가서 → tomcat9 버전 있음.

④ 요 사이트는 즐겨찾기(<https://launchpad.net/ubuntu/+ppas>) 해야 함. 없는 프로그램을 여기서 다운받아 설치 해야 함.

⑤ 요기에 저장소에 추가하는 명령어까지 나옴.

예> sudo add-apt-repository ppa:rivergod/tomcat8.5 (맨뒤 갖고 있는 사람 이름/프로그램명)

⑥ 터미널로 다시 와서

위에 꺼를 Ctrl+Shift+Insert(붙여넣기)하면 저장소에 추가된다.

> cd /etc/apt

> ls

sources.list 파일 있음

> cat sources.list → 안에 있는 내용 확인

확인해 보면 메인저장소에 있는 목록을 확인할 수 있음.

> clear

개인적으로 추가한 건 sources.lists.d 폴더에 있음.,

> cd sources.list.d

> ls

여기에 추가한 tomcat 9 보인다. -- 누구껀지도...

> cat riv --- 탭키 누르면 자동완성

> 그럼 톰캣이 실제 있는 곳의 주소가 sources.list.d 디렉토리에 등록된다. ---> 메뉴판 업데이트 해야지

> sudo apt install tomcat9 실행안됨... 없으니까

> sudo apt update 하고

> sudo apt install tomcat9

거부당함.. 업데이트가 안됨.... 우분투버전이 있는데 24.04 인데 코드명이 jammy인데 아까 그 페이지 가서 Overview of published packages 에 보면 jammy 없다.

글구 위에 작은 녹색글씨 눌러보면 거기도 없다. jammy를 지원해야만 설치가 가능하다.

즉, 둘의 버전이 맞아야 한다.

▷ 버전확인하는 법

> lsb\_release -a

```
ubuntu@ip-172-31-32-134:/etc/apt$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 22.04.2 LTS
Release:      22.04
Codename:     jammy
ubuntu@ip-172-31-32-134:/etc/apt$
```

그래서 내 우분투를 다운그레이드 할 수 없으니까. tomcat 9 버전중에 focal를 지원하는 놈을 찾는다.

좀 찾아보면... Apache Tomcat 9x 들어가서 확인해 보면 애는 focal지원함.

명령찾아서 실행하면 안되지... 먼저 저장된 놈이 있어서 ..

▷ 그럼 잘못된 버전의 ppa 저장소 개를 먼저 지우고.....

> sudo add-apt-repository --remove ppa:rivergod/tomcat8.5

> sudo apt update → 잘되면 잘 지워진거임.

다시 아까 그놈 복붙

> sudo add-apt-repository ppa:ttympuu/tomcat

→ 앤 제대로 설치 됨.

> ls

> sudo apt update

> clear

- > sudo apt-cache search tomcat → 설치된 버전 프로그램이름 확인하고....
- > sudo apt install tomcat9
- 잘 설치 됨.
- > netstat -nlpt
- > lsb\_release → 우분투 버전과 코드네임 확인 가능

```
ubuntu@ip-172-31-42-251:/etc/apt/sources.list.d$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 20.04.3 LTS
Release:      20.04
Codename:     focal
ubuntu@ip-172-31-42-251:/etc/apt/sources.list.d$
```

## ● 프로세스 실습

apt명령어로 프로그램을 설치 하면.. -- 서비스로 등록됨.

> service --status-all 돌아가는 프로세스가 보임.

```
ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$ service --status-all
[ + ] acpid
[ + ] apparmor
[ + ] appport
[ + ] atd
[ - ] console-setup.sh
[ + ] cron
```

> 지금은 service 명령을 잘 안쓰지만 일단 +는 도는 놔, -는 안도는 놔.

> sudo service tomcat8 stop

확인할 때 제일 좋은 거

> netstat -nlpt

> sudo service tomcat8 start

> netstat -nlpt

지금은 service 명령어보다 systemctl 제어명령을 주로 사용 함.  
service 명령어는 systemctl의 wrapper script 이다.

> clear

> sudo systemctl list-unit-files

--- 실행중인 시스템이 좌악 나온다. apt로 설치하면 웬만하면 여기에 다 등록이 된다.

> ctrl + c 또는 q 도 먹힘.

> sudo systemctl list-unit-files | grep tomcat8

-- 톰캣의 상태

> sudo systemctl status tomcat9

```
● tomcat8.service - LSB: Start Tomcat.
   Loaded: loaded (/etc/init.d/tomcat8; generated)
   Active: active (running) since Sat 2023-09-02 05:04:11 UTC; 5min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 10395 ExecStart=/etc/init.d/tomcat8 start (code=exited, status=0/SUCCESS)
    Tasks: 30 (limit: 1141)
   Memory: 69.1M
    CGroup: /system.slice/tomcat8.service
            └─10423 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/
```

종료 시키기

> sudo systemctl stop tomcat9



→ 크롬 띄워놓고 확인해 보자 ....

▷ ps 프로세스 명령어도 알아야 한다.

> ps -ef (PID - 프로세스 아이디, C - 프로세스 점유율)....

현재 실행중인 프로세스를 다 볼 수 있다.

종료가 안 될 때 강제종료를 해야 한다. 서비스로 등록이 안 되어 있는 프로세스도 종료해야 할 경우(내가 직접 압출파일 다운- 압축풀고 - 실행파일 실행 한 경우)

> kill -l (옵션 보기) → 강제 종료

```
ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD    18) SIGCONT    19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU    23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF    28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN   35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5 60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

9는 약간 찢찢하다고 함. (하던 일을 다 마무리 하지 못하고 종료됨.)

15번이 가장 안전하게 종료하는 거임.

삭제하려면 PID를 알아야 하니까

> ps -ef | grep tomcat9

```
ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$ ps -ef | grep tomcat8
tomcat8 10423 1 2 05:04 ? 00:00:03 /usr/lib/jvm/default-java/bin/java -Djava.util
.logging.config.file=/var/lib/tomcat8/conf/logging.properties -Djava.util.logging.manager=org.apac
he.juli.ClassLoaderLogManager -Djava.awt.headless=true -XX:+UseConcMarkSweepGC -Djdk.tls.ephemeral
DHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina
.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /usr/share/tomcat8/bin/bo
otstrap.jar:/usr/share/tomcat8/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat8 -Dcatalina.hom
e=/usr/share/tomcat8 -Djava.io.tmpdir=/tmp/tomcat8-tomcat8-tmp org.apache.catalina.startup.Bootstr
ap start
ubuntu 10469 1439 0 05:06 pts/0 00:00:00 grep --color=auto tomcat8
ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$
```

> whoami

ubuntu

> ps -ef 해 보면 tomcat을 실행한 주인아이디가 ubuntu 가 아님. 그래서 sudo로 죽여야 함.  
확인해 보면 tomcat8 임....

> sudo kill 177419 → 애가 PID → 찾고 → 원래 15(SIGTERM) 인데 생략가능 함.

> netstat -nlpt

그런데

> sudo systemctl status tomcat9을 확인해 보면

```

ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$ sudo systemctl status tomcat8
● tomcat8.service - LSB: Start Tomcat.
   Loaded: loaded (/etc/init.d/tomcat8; generated)
   Active: active (exited) since Sat 2023-09-02 04:46:02 UTC; 13min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 10186 ExecStart=/etc/init.d/tomcat8 start (code=exited, status=0/SUCCESS)

Sep 02 04:45:57 ip-172-31-38-253 systemd[1]: Starting LSB: Start Tomcat....
Sep 02 04:45:57 ip-172-31-38-253 tomcat8[10186]: * Starting Tomcat servlet engine tomcat8
Sep 02 04:46:02 ip-172-31-38-253 tomcat8[10186]: ...done.
Sep 02 04:46:02 ip-172-31-38-253 systemd[1]: Started LSB: Start Tomcat..
ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$

```

exited 로 보이므로 현재 실행중은 아닌데 잠깐 일시 정지된 것으로 인식하고 있음.

그래서 다음 명령이 안 먹힐 때가 있다.

```
> sudo systemctl start tomcat9
```

구동 후 다시 확인해도

```
> sudo systemctl status tomcat9 → exited로 뜸
```

왜 그냐? sudo systemctl stop tomcat9로 종료시켜야 하는데 그냥 프로세스를 강제 죽여버려서 인식을 못하는 거임.

이럴때는 톰캣을 리스타트 시켜야함.

```
> sudo systemctl restart tomcat9
```

```
> sudo systemctl status tomcat9
```

```

ubuntu@ip-172-31-38-253:/etc/apt/sources.list.d$ sudo systemctl status tomcat8
● tomcat8.service - LSB: Start Tomcat.
   Loaded: loaded (/etc/init.d/tomcat8; generated)
   Active: active (running) since Sat 2023-09-02 05:04:11 UTC; 21s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 10395 ExecStart=/etc/init.d/tomcat8 start (code=exited, status=0/SUCCESS)
    Tasks: 30 (limit: 1141)
   Memory: 71.9M
    CGroup: /system.slice/tomcat8.service
           └─10423 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/

```

실제 수행한 것은

Process : 177600 ExecStart=/etc/init.d/tomcat8 start 이렇게 실행 함..

만약 서비스에도 등록이 안되어 있다면 위의 놈을 직접 실행시켜야 함.

## ▶ 정리

systemctl 입장에서는 exit 된 것이고 안전한 종료 후에는 restart 시켜야 한다. 그러니까 강제 종료 후 systemctl stop을 시킨 후 start 하면 잘 된다.

```
> sudo kill 177419
```

```
> sudo systemctl stop tomcat9
```

```
> sudo systemctl start tomcat9
```

이래도 동작 함.... 어찌보면 이 순서를 기억했다가 처리하는 것이 낫다.

실행되고 있는 프로세스를 확인할 때 최종 입력한 명령어도 프로세스화 되어서 보여지고 애플 그랩해도 마지막 그랩한 애도 보인다. 애플 제외하려면 즉 그랩 안보이게 하려면

```
> ps -ef | grep tomcat9 | grep -v grep
```

```
> ps -ef | grep tomcat9 | grep -v grep | awk '{print $1}'
```

```
ubuntu@ip-172-31-38-253:~$ ps -ef | grep tomcat8 | grep -v grep | awk '{print $1}'
tomcat8
ubuntu@ip-172-31-38-253:~$
```

이렇게 출력할 수 있다.

파이프라인을 통해 가져온 문자열을 공백을 이용해 토큰화 시키고 그것 중에 1번째 것을 가져옴.(공백을 기반으로 토큰화 시켜주는 명령이 awk 임.)

```
tomcat8 1번째
177727 2번째
1
1
13:04
?
00:00:04
/usr/lib/jvm/default-java/bin/java -Djava.ut
```

이렇게 찾은 프로세스 아이디를 죽이는데 쓸건데

이때 "백틱은 치환명령어"으로 감싼 부분을 실행결과로 치환해 준다. 백틱은 키보드 좌 상단 틸드(~) 밑에 있는 놈이다.

```
> sudo kill `ps -ef | grep tomcat8 | grep -v grep | awk '{print $2}`
```

실행하면 죽음.

```
> ps -ef 으로 톰캣 죽었는지 확인
```

```
> sudo systemctl status tomcat9
```

## ● 리눅스 명령어 vi 에디터 사용법

> vim a.txt → 일반모드

> i → INSERT → 모드 맨아래 보임

hello everyone

bye

ESC 키 누름 → 명령행 모드

:wq (저장 후 종료)

> cat a.txt

→ 키패드로 이동은 되나 키보드가 안 먹히는 상태 임.

→ 쉼표(:) 명령행 모드

- wq 저장 후 종료, q! 강제 종료
- i 는 입력, dd 는 행 삭제,
- yy 현재 행 복사, p 붙여넣기, ctrl + b : 페이지 스크롤하기
- shift + v 블럭지정
- y 는 블럭복사, \$ 행 맨뒤로 가기

윈도우 메모장 같은 프로그램으로 기능은 별로이나 가장 호환성이 좋아서 널리 사용되고 있음.

## ● chmod (권한)

현재 접속유저는 ubuntu, 최상위 유저는 root

```
> su root  
password
```

현재 비밀번호를 모르는 상태임.

```
ubuntu@ip-172-31-42-251:~$ su root  
Password:  
su: Authentication failure  
ubuntu@ip-172-31-42-251:~$ sudo root passwd  
sudo: root: command not found  
ubuntu@ip-172-31-42-251:~$ sudo passwd root  
New password: █
```

루트의 비밀번호 만들어야 함.

```
> sudo passwd root  
New password : 1234 ---> 표시 안남  
> su root  
Password :  
root@...../home/ubuntu# ----> root 로 로그인함.
```

```
> vi test1.txt  
> i  
hello world  
:wq
```

```
> cat a.txt --- 잘보임  
> 유저변경  
> su ubuntu -- 루트에서 일반계정 변경시엔 비밀번호 안물어봄.. 어차피 루트가 최고니까  
> ls -l
```

```
ubuntu@ip-172-31-38-253:~$ ls -l  
total 8  
-rwxrwxrwx 1 root root 15 Sep  2 05:46 a.txt  
-rw-rw-r-- 1 root root 13 Sep  2 05:48 test.txt  
ubuntu@ip-172-31-38-253:~$ █
```

```
- rw- r-- r-- 1 root root
```

첫자 : - : 파일, d 디렉토리, l softlink

첫세번째 : 소유자(여기서는 루트 권한)

두번째 : 그룹

세번째 : anyone 누구나

위에꺼는 644의 권한을 줌

결국 ubuntu는 root가 만든 파일에 대해 anyone으로 읽기 권한밖에는 없음

루트가 만든 파일을 누구나 수정할 수 있도록 권한 수정

> vi a.txt

> 수정 후 저장하려면 저장 안됨.

esc :q! --- 저장하지 않고 강제 종료해야 함.

> 권한 변경해야 함.

> chmod 646 a.txt

권한 없음으로 파일 모드 안바뀜... 그러나 sudo 권한을 위임받아 변경함.

> sudo chmod 646 a.txt

바뀜

> vi a.txt

i -- 수정모드

수정 가능함.

esc :wq 저장도 됨.

> sudo chmod 777 a.txt ----> 모든 권한... 777

실행 파일은 속성 x 를 갖고 녹색으로 표시 된다.

```
ubuntu@ip-172-31-38-253:~$ sudo chmod 777 a.txt
ubuntu@ip-172-31-38-253:~$ ls -l
total 4
-rwxrwxrwx 1 root root 15 Sep  2 05:46 a.txt
ubuntu@ip-172-31-38-253:~$
```

## ● 리눅스 명령어 chown(그룹 변경)

> su root

> vi test.txt

Hello world!

> sudo chmod 664 test.txt

```
ubuntu@ip-172-31-38-253:~$ su root
Password:
root@ip-172-31-38-253:/home/ubuntu# vi test.txt
root@ip-172-31-38-253:/home/ubuntu# sudo chmod 664 test.txt
root@ip-172-31-38-253:/home/ubuntu# ls -l
total 8
-rwxrwxrwx 1 root root 15 Sep  2 05:46 a.txt
-rw-rw-r-- 1 root root 13 Sep  2 05:48 test.txt
root@ip-172-31-38-253:/home/ubuntu#
```

> 그룹은 안 바뀐다.

> sudo chown root:ubuntu test.txt

```
ubuntu@ip-172-31-38-253:~$ ls -l
total 8
-rwxrwxrwx 1 root root 15 Sep  2 05:46 a.txt
-rw-rw-r-- 1 root ubuntu 13 Sep  2 05:48 test.txt
ubuntu@ip-172-31-38-253:~$
```

원래 -rw-rw-r-- root root 여기서 첫째 root 는 소유자, 두번째 root는 그룹소유자 인데...  
위 명령으로 그룹소유자에 ubuntu가 추가 되었다.

> vi test2.txt → 수정 가능하다.

### ● find 명령어

> netstat -nlpt

> sudo find / -name tomcat

관리자 권한으로 루트부터 이름으로 검색할거야 찾을 이름은 tomcat 이란다.

시간이 지나면 .. 보인다. tomcat 뒤에 \* 붙이면 다 나온다.

톰캣8 파일만 찾아보자

> sudo find / -name tomcat8

윈도우는 프로그램 파일즈 폴더에 설치한 파일 전체가 놓이지만, 리눅스는 각각의 폴더 역할이  
정해져 있기 때문에 프로그램을 설치하면 분산되서 저장된다.

```
ubuntu@ip-172-31-38-253:~$ sudo find / -name tomcat8
/var/cache/tomcat8
/var/log/tomcat8
/var/lib/tomcat8
/usr/share/doc/tomcat8
/usr/share/tomcat8
/usr/libexec/tomcat8
/etc/init.d/tomcat8
/etc/cron.daily/tomcat8
/etc/tomcat8
/etc/default/tomcat8
/etc/logrotate.d/tomcat8
ubuntu@ip-172-31-38-253:~$
```

> cd /var/lib/tomcat8 → 탭키 누르면 자동 완성 됨.

> ls

> cd webapps

> cd ROOT

> ls → index.html 파일 보임....

```
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$ ls -l
total 8
drwxr-xr-x 2 root root 4096 Sep  2 04:34 META-INF
-rw-r--r-- 1 root root 1896 Sep  2 04:34 index.html
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$
```

위 파일 중 index.html 권한을 ubuntu에게 줄거예요, 그런데 숫자가 아닌 다른 방법으로 권한



줘 볼 겁니다.

> sudo chmod u+x,g+wx,o+x index.html → 띄어쓰기 하면 안됨... 멘탈 나감...

u : user, g : group, o : others 의 약자임...

> ls -l

ubuntu 계정은 others 그룹에 들어가니까 이리 해도 ubuntu는 수정 못함. 그래서 ubuntu에게 수정할 수 있도록 others 권한에 w 권한 추가

> sudo chmod o+w index.html or sudo chmod o=rw index.html

```
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$ sudo chmod u+x,g+wx,o+x index.html
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$ ls -l
total 8
drwxr-xr-x 2 root root 4096 Sep  2 04:34 META-INF
-rwxrwxr-x 1 root root 1896 Sep  2 04:34 index.html
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$ sudo chmod o+w index.html
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$ ls -l
total 8
drwxr-xr-x 2 root root 4096 Sep  2 04:34 META-INF
-rwxrwxrwx 1 root root 1896 Sep  2 04:34 index.html
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/webapps/ROOT$
```

> vi index.html

내용은 dd 로 지우고

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Apache Tomcat</title>
</head>
<body>
<h1>Tomcat 8000 port!</h1>
</body>
</html>
```

이렇게 남기고 저장 후 종료

esc :wq

> cat index.html

웹 브라우저 열고 54.180.102.254:8080





▷ 포트는 어찌 바꾸는가.

> cd ..

> cd ..

> ls

/var/lib/tomcat8 폴더 밑에 conf 라는 설정파일을 담는 폴더가 있음

> cd conf

> ls

여기 들어가면 server.xml 파일이 있다. 해당 파일이 수정가능한지 확인

> ls -l

권한 없으니 sudo 권한 대행받아 수정해 봅시다.

> sudo vi server.xml

페이지 단위로 스크롤하는 단축키가 뭐? ctrl+b 이지요

찾아가서 port를 8000 으로 수정

```
<Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
            port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
```

i 누르고 수정해야 함.... 잊지말기

esc :wq

→ 테스트 해 봐야지 → 될 것 같지만 안됨...

→ 설정 파일이 바뀌면 서비스를 다시 재시작 해야 함.

> sudo systemctl restart tomcat8

> netstat -nlpt

```

ubuntu@ip-172-31-38-253:/var/lib/tomcat8/conf$ sudo servicectl
sudo: servicectl: command not found
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/conf$ sudo systemctl restart tomcat8
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/conf$ netstat -nlpt
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                     LISTEN      -
tcp6       0      0 :::1:6010               :::*                     LISTEN      -
tcp6       0      0 :::8000                  :::*                     LISTEN      -
ubuntu@ip-172-31-38-253:/var/lib/tomcat8/conf$ █

```

※ 웹브라우저에서 확인 → 안된다.... 음 → AWS 방화벽 때문  
AWS 홈페이지에 다시 들어가서 로그인하고

▶ EC2 가서 보안규칙 → 인바운드 규칙에 8000 번 포트 열어 줘야 함.

위에 보안그룹 누르고 인바운드 규칙에 규칙추가 8000 anywhere 규칙 바꿔주고 추가 저장.

## ● 리눅스 명령어 표준입출력

표준입출력을 이해하려면 log를 확인하는 개념을 먼저 이해해야 한다.

> sudo find / -name catalina.out

```

ubuntu@ip-172-31-38-253:/var/log/tomcat8$ sudo find / -name catalina.out
/var/log/tomcat8/catalina.out
ubuntu@ip-172-31-38-253:/var/log/tomcat8$ █

```

얘가 톰캣의 로그를 보여주는 파일이야

> cd /var/log/tomcat9

> sudo tail -f catalina.out → 파일의 끝을 보여주는데 -f는 계속 변경하는 것도 보여준다.

얘는 이렇게 돌게 하고 위 창 탭에서 우 클릭 duplicate 해서 현재 콘솔 복제창 생성

여기서 톰캣을 재시작해 보고 tail 명령어에서 어떤 변화가 일어 나는지 확인

> sudo systemctl tomcat8 restart

▶ tail 은 모니터링 할 때 많이 쓴다.

현재는 catalina 파일을 모니터로 보고 있다.

● 표준 입출력

- 표준입력스트림 : stdin
- 표준출력스트림 : stdout
- 오류출력스트림 : stderr

● 표준스트림

파일 디스크립터	설 명
0	표준입력(Standard Input)
1	표준출력(Standard output)
2	표준에러(Standard Error)

1 > &2 : 표준 출력을 오류로 내보낸다.

2 > &1 : 표준 오류를 출력으로 내보낸다.

지금 현재는 표준 출력으로 모니터로 출력되고 있는 것을 파일로 표준 출력을 변경해서 파일로 저장을 해 보겠다.

원래 tail 실행 되는 창으로 갑니다. -> clear

먼저 파일하나 만들고,;..

> sudo touch mylog.out

> sudo tail -f catalina.out > mylog.out

권한이 없음.... 다시 줘야지

> sudo chmod 777 mylog.out

> sudo tail -f catalina.out > mylog.out

반응 없음... 지금 계속 생성 로그를 파일에 쓰고 있기 때문에 종료가 안됨..

옆 탭으로 이동해서

> cd /var/log/tomcat9

> vi mylog.out

새로운 탭 열고 (복제 알지?)

서버 재 시작

> sudo service tomcat8 restart

순환구조이므로 ..... 두번째 탭을 열고 catalina.out 파일을 다시 보면 수정되어 있지...

정리 표준입력 0, 표준출력 1, 표준에러 2.... 방향 전환은 >

나중에 스프링 프로젝트를 배포할 때 디버깅 할 때 필요하니 이해하고 있어야 함.

## 6장. 배포v1 - 흐름이해하고 EC2 생성하기

### ① 배포v1 - 프로젝트 만들기

→ 참고 프로젝트 : <https://github.com/zzzmini/aws-v1.git>

### ● EC2서버 생성 및 고정IP 설정

※ EC2 인스턴스 생성 - 하나는 무료 두 개부터는 비용 발생함.

▶ ec2 서버 - t2.micro 서버

- 프리티어 사용자들(가입한지 1년이 안된)
- 750시간 무료(한달)
- 1일 24시간
- 30일 720시간
- 32일 744시간

※ ec2 서버 - t2.micro 서버 2대 - 1488시간 → 그러니까 비용이 발생한다.

① 기존에 만든 EC2 서버를 종료 → 마우스 우클릭 인스턴스 종료 처리 함.

② 새로운 인스턴스 생성

③ 4단계 볼륨 유형 SSD

▷ 태그 -- 인스턴스가 여러개 일 때 구분할 때 사용

④ 보안그룹 - 8080 위치무관

⑤ 검토 및 시작

⑥ 키페어는 기존 있는거 사용

→ 기존 중지시킨 인스턴스는 며칠(몇시간) 지나면 자동으로 사라짐.

좀 기다리면 서버 만들어짐.

추가적으로 세팅해야는 것이 있음.

⑦ 고정아이피를 설정해야 함.

- 엘라스틱(탄력적) 아이피라고 함.

- 지속적인 서비스를 제공하려면 고정IP가 필요하다.....

▶ EC2 는 유동아이피를 준다.

퍼블릭 ipv4 는 유동아이피지.. 애를 고정아이피로 바꿔야 한다. 왼쪽 메뉴를 보면 탄력적 IP 가 있다. 애가 고정아이피 만드는 것임.

⑧ 탄력적 IP 주소 할당 --> 만들기... 자동으로 만들어 줌.....

주의 명심..... 비용생기니까.....

**※프리티어에게는 탄력적 IP 한개를 무료로 제공하나 주의 또 주의 해야 함!!!**

조심조심 명심 : EC2 - 탄력적 IP를 연결한 경우만 무료

- 생성하고 연결안한 경우,
- 2개 생성한 경우,
- EC2 서버 종료 시 탄력적 IP 반환 안시켰을 때, 이때 IP가 놓고 있으니까

위 상황일 때 비용이 발생한다. 서버 종료 시 엘라스틱 아이피도 반드시 삭제해야 한다.

**우 상단 콤보상자에서 탄력적 IP 주소 연결(릴리즈는 삭제)**

연결 창 설정 쉬움.



## 탄력적 IP 주소 연결 정보

이 탄력적 IP 주소에 연결할 인스턴스 또는 네트워크 인터페이스를 선택합니다. (43.200.135.202)

**탄력적 IP 주소: 43.200.135.202**

리소스 유형  
탄력적 IP 주소를 연결할 리소스의 유형을 선택합니다.

☒ 인스턴스  
☐ 네트워크 인터페이스

⚠ 탄력적 IP 주소가 이미 연결된 인스턴스와 탄력적 IP 주소를 연결하면 이전에 연결된 탄력적 IP 주소가 연결 해제 되지만 주소는 여전히 계정에 할당됩니다. [자세히 알아보기](#)

프라이빗 IP 주소를 지정하지 않으면 탄력적 IP 주소가 기본 프라이빗 IP 주소와 연결됩니다.

인스턴스

X

프라이빗 IP 주소  
탄력적 IP 주소를 연결할 프라이빗 IP 주소입니다.

X

▷ 다시 인스턴스에 와서 하단 탄력적 아이피 주소 확인하면 고정IP 할당 받은 거 보임..

▷ 화면 새로고침 하면...

인스턴스 (1) 정보										인스턴스 상태	작업	탄력적 IP 삭제
Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	정보 상태	가용 영역	탄력적 IPv4 DNS	탄력적 IPv4 ...	탄력적 IP			
inst-v1	i-05c278e2fd93a06db	보통	m5.xlarge	3/2가 정상 통과...	중요 없음	ap-northeast-2	ec2-43-200-135-202.ap...	43.200.135.202	43.200.135.202			

## ⑨ 연결 테스트 해보자

MobaXterm을 연다..

1. 기존의 탭에 연결된 EC2 서버는 중지했으니까. 왼쪽창에서 delete session

2. 새로 만들자.....

아이피 주소 넣고 username: ubuntu port 22

아래 Advance 에서 프라이빗 키 연결 ---> 저장된 키 알지?

연결 오케...

고정아이피는 메모장에 넣어서 저장해 놓자...

## 7장. 배포v1 - 배포완료

### ● jdk 설치합니다.

> git --version

① 설치되어 있음.

② 깃에 가서 코드 창 우측에 Clone 밑 HTTPS 주소 복사

③ 다시 와서 /home/ubuntu 폴더에 깃 소스를 받을 거임.

> pwd

> git clone shift+insert 하면 복사한 주소가 복붙됨...

다운을 받자...

```
ubuntu@ip-172-31-37-154:~$ git --version
git version 2.25.1
ubuntu@ip-172-31-37-154:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-37-154:~$ git clone https://github.com/codingspecialist/aws-v1.git
Cloning into 'aws-v1'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 31 (delta 0), reused 31 (delta 0), pack-reused 0
Unpacking objects: 100% (31/31), 61.01 KiB | 349.00 KiB/s, done.
ubuntu@ip-172-31-37-154:~$ ls
aws-v1
ubuntu@ip-172-31-37-154:~$ cd aws-v1
ubuntu@ip-172-31-37-154:~/aws-v1$ ls
README.md build.gradle gradle gradlew gradlew.bat settings.gradle src
ubuntu@ip-172-31-37-154:~/aws-v1$
```

→ 경로로 들어가보면 깃에 있던 내용이 보인다.

### ● gradle을 통해서 테스트와 빌드를 진행할 거야

목적파일 즉 실행파일로 바뀌어야 함. 자바는 jar 파일이 곧 실행파일이다.

```
ubuntu@ip-172-31-37-154:~/aws-v1$ ls -l
total 32
-rw-rw-r-- 1 ubuntu ubuntu 1016 Mar 27 07:10 README.md
-rw-rw-r-- 1 ubuntu ubuntu 1137 Mar 27 07:10 build.gradle
drwxrwxr-x 3 ubuntu ubuntu 4096 Mar 27 07:10 gradle
-rw-rw-r-- 1 ubuntu ubuntu 8070 Mar 27 07:10 gradlew
-rw-rw-r-- 1 ubuntu ubuntu 2674 Mar 27 07:10 gradlew.bat
-rw-rw-r-- 1 ubuntu ubuntu 24 Mar 27 07:10 settings.gradle
drwxrwxr-x 4 ubuntu ubuntu 4096 Mar 27 07:10 src
ubuntu@ip-172-31-37-154:~/aws-v1$
```

gradle 애는 폴더구... **gradlew** 가 실행파일임...

실행파일인데 실행권한이 없다. 소유자가 ubuntu 니까 실행권한을 직접 주자

순서

1. 자바 설치(JDK - 자바를 개발하는 툴이 모여있고, JRE -자바를 실행을 시켜주는 애)
  - JDK로 마인크래프트 게임 개발
  - 일반 사용자들이 마인크래프트 게임 하려면 JRE가 필요

> sudo apt update

> sudo apt install openjdk-17-jdk

2. github 에서 소스 다운받기

```
> git clone https://github.com/...../aws....git
```

3. gradlew 실행권한주기

```
> cd 다운받은 폴더 팁... cd ~ ----사용자 홈폴더로 이동
```

```
> cd ~/awsv1
```

```
> chmod u+x gradlew
```

4. gradlew로 프로젝트를 jar 파일로 변경하기

```
./gradlew build ---- ./ 현재 폴더를 안찾고 환경변수에서 찾아서 실행해 줘서 실행 안될 수 있다. 반드시 폴더 위치를 명시..
```

5. java로 jar를 실행

```
cd build/libs
```

```
java -jar *.jar ==== 요래 해도 되지...
```

만서도.... 결국은 nohup으로 돌린다. 아래서...

```
> chmod u+x gradlew
```

```
> ./gradlew build
```

빌드 안됨.... JDK가 없으니 안되지....

## ● 자바설치 합시다.

```
> sudo apt update → 메뉴판 업데이트
```

메뉴판에 자바가 있는 지 찾음.

```
> sudo apt-cache search jdk | grep openjdk-17
```

빨강색으로 보이긴 하는데 자바 버전을 확인해 봐야 할 듯 함... 우린 17로 개발해 놔서... 음!!!

있음.....

JDK 설치하고 컴파일 그 다음에 실행파일 만들자

```
>sudo apt install openjdk-17-jdk
```

설치 후 설치 확인하고 빌드

```
> java --version
```

```
ubuntu@ip-172-31-38-253:~$ java --version
openjdk 17.0.8.1 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu120.04, mixed mode, sharing)
ubuntu@ip-172-31-38-253:~$
```



```
> ls
> ./gradlew build
```

```
ubuntu@ip-172-31-37-154:~/aws-v1$ ./gradlew build
Downloading https://services.gradle.org/distributions/gradle-7.3.3-bin.zip
.....10%.....20%.....30%.....40%.....50%.....60%.....70%.....80%.....90%.....100%

Welcome to Gradle 7.3.3!

Here are the highlights of this release:
- Easily declare new test suites in Java projects
- Support for Java 17
- Support for Scala 3

For more details see https://docs.gradle.org/7.3.3/release-notes.html

Starting a Gradle Daemon (subsequent builds will be faster)
```

프로젝트를 jar 파일로 굽는다.

다 되면 확인

```
> ls
-- 빌드 폴더가 생긴다.
> cd build
> ls
> cd libs
> ls
```

```
ubuntu@ip-172-31-37-154:~/aws-v1/build/libs$ ls
v1-0.0.1-SNAPSHOT.jar
ubuntu@ip-172-31-37-154:~/aws-v1/build/libs$
```

## ● 실행해 보자

> java -jar v1-0- 탭 누르고 자동완성

```
ubuntu@ip-172-31-37-154:~/aws-v1/build/libs$ java -jar *.jar

Spring
:: Spring Boot ::
(v2.6.3)

2022-03-27 07:31:01.818 INFO 16367 --- [main] com.metacoding.v1.ViApplication : Starting V1
7 (/home/ubuntu/aws-v1/build/libs/v1-0.0.1-SNAPSHOT.jar started by ubuntu in /home/ubuntu/aws-v1/build/libs)
2022-03-27 07:31:01.825 INFO 16367 --- [main] com.metacoding.v1.ViApplication : No active pr
2022-03-27 07:31:04.186 INFO 16367 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat init
2022-03-27 07:31:04.214 INFO 16367 --- [main] o.apache.catalina.core.StandardService : Starting se
2022-03-27 07:31:04.215 INFO 16367 --- [main] org.apache.catalina.core.StandardEngine : Starting Se
```

크롬 열고.....

고정아이피주소:8080/aws/v1 이 주소가 될거임.. 잘 나오면

고정아이피주소:8080/aws/v1?number=0 WARN

고정아이피주소:8080/aws/v1?number=-1 ERROR

고정아이피주소:8080/aws/v1?number=1 INFO

터미널에 로그 나오는지 확인....

근데 터미널 종료하면 서비스도 종료 됨.... 이런 저런 것들을 추가로 셋팅 해야함.

## 8장. 배포v1 - nohup으로 백그라운드 실행하기

※ 터미널이 꺼져도 종료가 안 되게 설정

먼저 자바스프링 프로젝트의 build.gradle 에

```
jar(  
    enabled = false  
)
```

를 추가해야 함.

▶ 애플리케이션을 안 넣어주면 jar 파일이 두 개가 생긴다. 필요 없는 plane 자바냐? 하튼 이런거 하나 더 생겨서 \*.jar 로 실행할 때 헷갈림.

> cd aws-v1

> **nohup java -jar \*.jar &** → &를 붙이면 백그라운드에서 돌아간다.

nohup 명령어는 리눅스에서 프로세스를 실행한 터미널의 세션 연결이 끊어지더라도 지속적으로 동작할 수 있게 해주는 명령어입니다.

> nohup --version

> clear

> nohup java -jar \*.jar &

재접속

> nohup java -jar \*.jar &

다시 테스트.... 애플리케이션은 백그라운드로 실행하고 있음.... .백그라운드 실행은 로그가 남지 않는다.....

터미널 종료하고 ... 다시 테스트 하면 ... 잘 뜬다...

> netstat -nlpt

실행 안 됨.. 애플리케이션도 다시 설치해야 함.....EC2 서버바깥으니까...

> sudo apt install net-tools

> netstat -nlpt

----- 8080 잘 도는지 확인....

로그가 안보이니.... 어찌 확인하는가?

▶ nohup은 자동으로 로그를 남긴다.

어디?

> cd libs

> ls

요기 보면 nohup.out 이라는 로그파일을 생성해 준다.

> cat nohup.out

> clear

모니터링하려면....

> tail -f nohup.out

```
ubuntu@ip-172-31-38-253:~/aws-v1/aws-v1/build/libs$ tail -f nohup.out
2023-09-02T12:20:42.350Z INFO 51487 --- [          main] o.apache.catalina.core.StandardService :
Starting service [Tomcat]
2023-09-02T12:20:42.351Z INFO 51487 --- [          main] o.apache.catalina.core.StandardEngine :
Starting Servlet engine: [Apache Tomcat/10.1.12]
2023-09-02T12:20:42.631Z INFO 51487 --- [          main] o.a.c.c.C.[Tomcat].[localhost].[/] :
Initializing Spring embedded WebApplicationContext
2023-09-02T12:20:42.636Z INFO 51487 --- [          main] w.s.c.ServletWebServerApplicationContext :
```

## ● 타임존 변경 및 종료스크립트 코딩하기

> timedatectl

```
ubuntu@ip-172-31-38-253:~/aws-v1/aws-v1/build/libs$ timedatectl
Local time: Sat 2023-09-02 12:26:18 UTC
Universal time: Sat 2023-09-02 12:26:18 UTC
RTC time: Sat 2023-09-02 12:26:18
Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

보면 UTC 미국시간이다.

애를 한국시간으로 바껴야 한다.

> timedatectl list-timezones

엄청나게 많은 시간존이 나와서 grep하자 서울 찾아야지

> timedatectl list-timezones | grep Seoul

> **sudo timedatectl set-timezone Asia/Seoul**

```
ubuntu@ip-172-31-38-253:~/aws-v1/aws-v1/build/libs$ sudo timedatectl set-timezone Asia/Seoul
ubuntu@ip-172-31-38-253:~/aws-v1/aws-v1/build/libs$ timedatectl
          Local time: Sat 2023-09-02 21:27:21 KST
          Universal time: Sat 2023-09-02 12:27:21 UTC
            RTC time: Sat 2023-09-02 12:27:21
            Time zone: Asia/Seoul (KST, +0900)
System clock synchronized: yes
          NTP service: active
            RTC in local TZ: no
ubuntu@ip-172-31-38-253:~/aws-v1/aws-v1/build/libs$
```

하면 Asia/Seoul 로 타임존이 바뀜.

## ● aws-v1의 문제점.

1. 자동화 스크립트가 없다
2. 서버가 멈춘다. ---> 재배포시 ---> 새로운 서버에 던지면 되는데...음..
3. 재배포할 때는 임시서버에서 배포해 보고 테스트하고 문제 없으면 본 서버에 던지면 되는데....  
그게 없다.. 그것을 해결하는 것이 깃헙액션 + 엘라스틱 빈스톡 → 무중단 배포  
그래서 aws-v1은 테스트를 못하고 바로 배포가 됨.
4. 사용자 폭증 시 못 버틴다.
5. github 코드 push만하면 자동으로 배포될 수는 없을까?
6. EC2 서버에 프로그램 설치 귀찮으니 자동으로 해 주면 좋지 않을까? --- 기본적으로 JDK가 설치 되어있는 엘라스틱 빈 스톱을 사용하면 된다.

**나중에 EC2 죽일때는 탄력 아이피 해제하고 릴리즈까지 꼭 하시오...!**

빌드해서 jar 파일 구울 때 gradle로 빌드 될 때는 늘 테스트가 성공해야만 build를 진행한다. 그래서 테스트 코드는 무조건 하나는 있어야 한다.

테스트 없이 빌드를 하고 싶을 때는

**> ./gradlew build -x test**

--- 요래 주면 테스트 없이 빌드 한다....