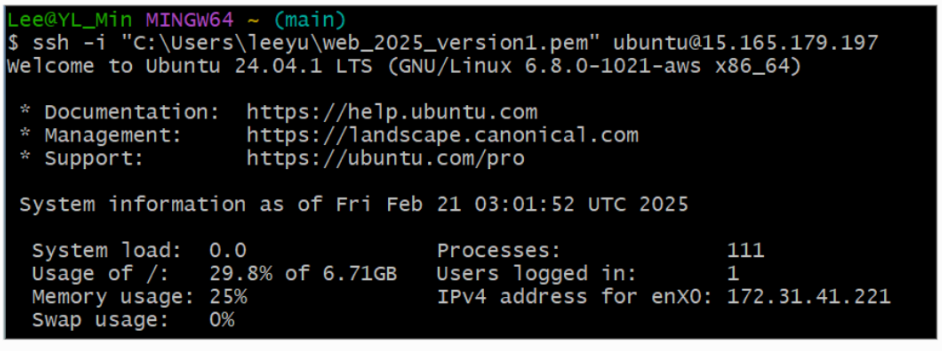
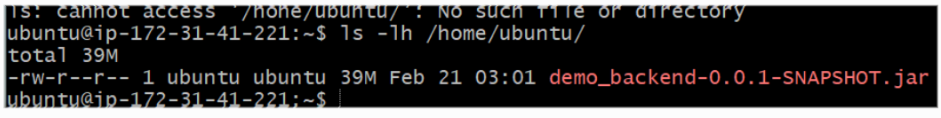
1.
EC2에 JAR 파일 업로드(EC2 클라우드 컴퓨터)에서 Java 프로젝트를 실행하려면 JAR파일이 필요하기 때문에 JAR파일을 내 로컬컴퓨터에서 만들어서 EC@에 업로드
- jar 파일 : java 프로젝트를 실행할수 있도록 압축한 파일
- scp : secyre Copy Protocol , 파일을 보낼때 사용하는 명령어
```sh
scp -i "/c/Users/leeyu/web_2025_version1.pem" "/c/Users/leeyu/portfolio/demo_spring_Vue/Ai-Diary-server/target/demo_backend-0.0.1-SNAPSHOT.jar" ubuntu@15.165.179.197:/home/ubuntu/
```

2.
EC2에 SSH 접속 - EC2는 클라우드 컴퓨터이므로 직접 키보드와 마우스로 조작 할 수 없음 , 대신 SSH(Secure Shell)로 이용해 컴퓨터에서 명령어를 입력
```sh
ssh -i "/c/Users/leeyu/web_2025_version1.pem" ubuntu@3.34.90.124
```



```

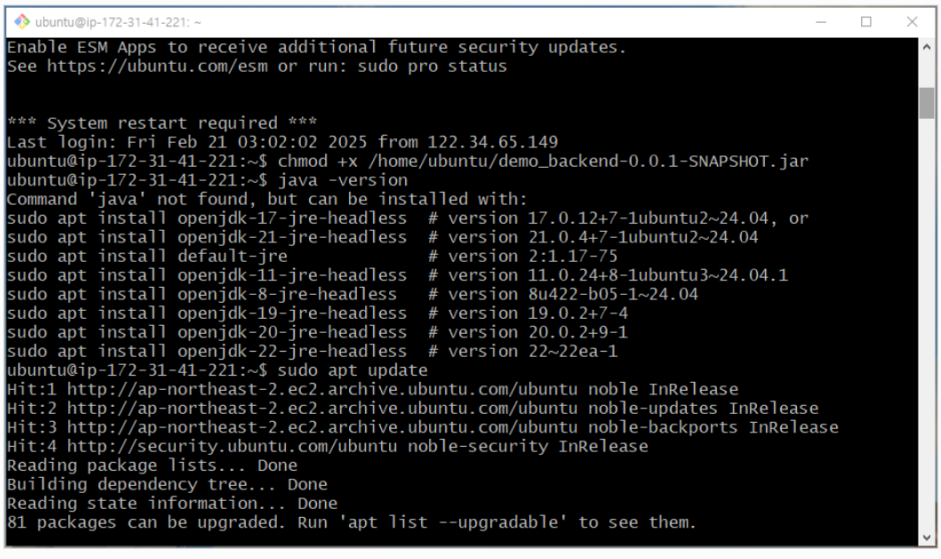
3.
```sh
ls -lh /home/ubuntu/
```



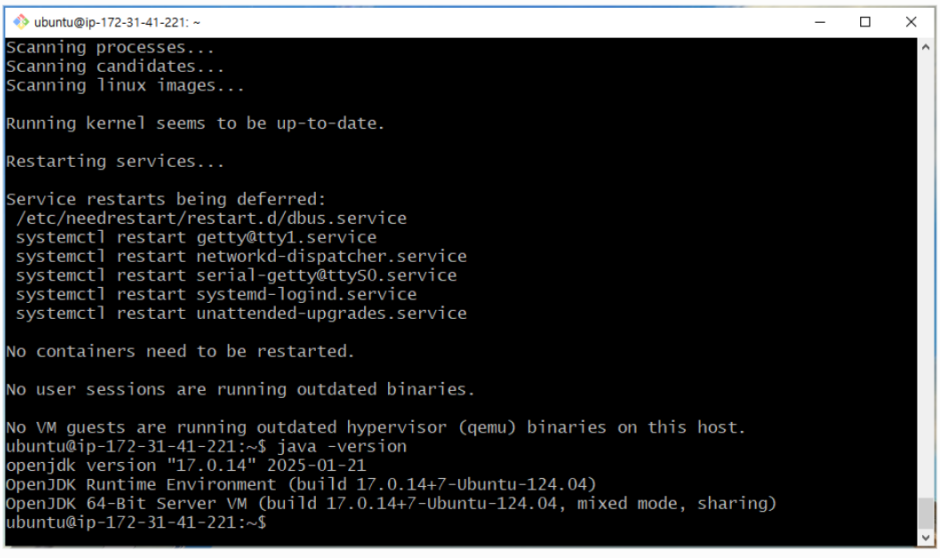
JAR 파일 실행 권한 부여와 JAVA설치 - chmod +x 명령어로 실행 권한 추가
- chmod : 파일 실행 권한을 추가하는 명령어

```sh
chmod +x /home/ubuntu/demo_backend-0.0.1-SNAPSHOT.jar
```

- java -jar 실행시 command java not found 오류 발생한다면 java 설치 필요



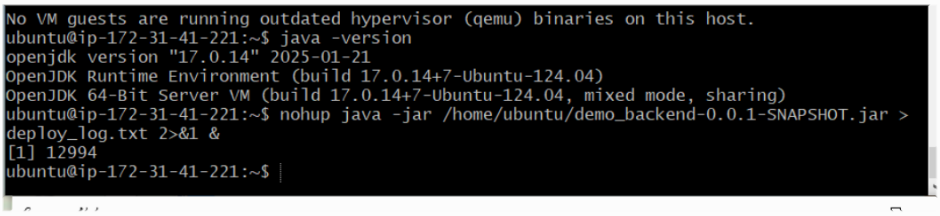

```sh
```
```

sudo apt update
sudo apt install openjdk-17-jre-headless -y
java -version

```




5. JAR파일 실행해서 EC2서버를 띄운다.
```sh
nohup java -jar /home/ubuntu/demo_backend-0.0.1-SNAPSHOT.jar > log.txt 2>&1 &
```
- nohup : SSH를 종료해도 프로그램 계속 실행된 실행 ,
SSH 연결이 끊겨도 계속 실행되도록 하기 위해 백그라운드 실행이 필요

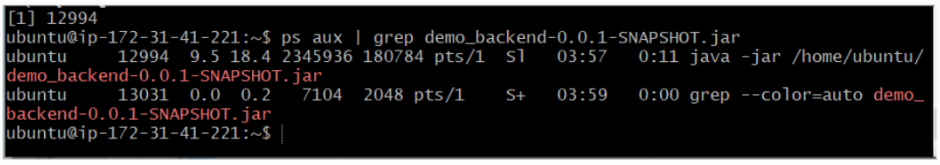
```

```
> deploy_log.txt 2>&1 &
```
실행 로그를 deploy_log.txt 파일에 저장



```sh
ps aux | grep demo_backend-0.0.1-SNAPSHOT.jar
```

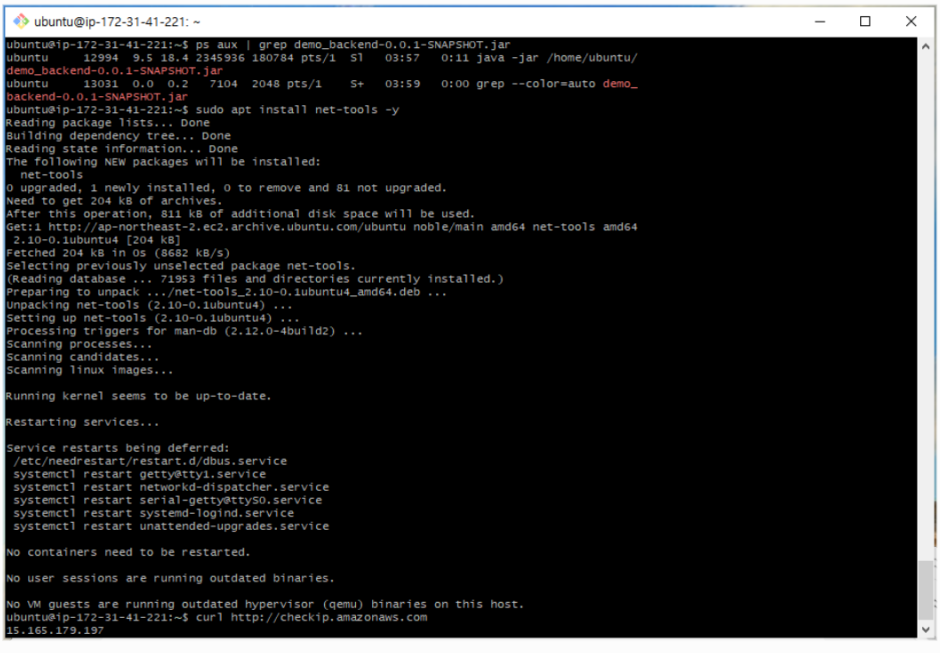
제대로 실행되고 있는지 확인가능



ps aux : 실행중인 프로세스 목록 확인
grep demo_backend : 실행중인 필터링

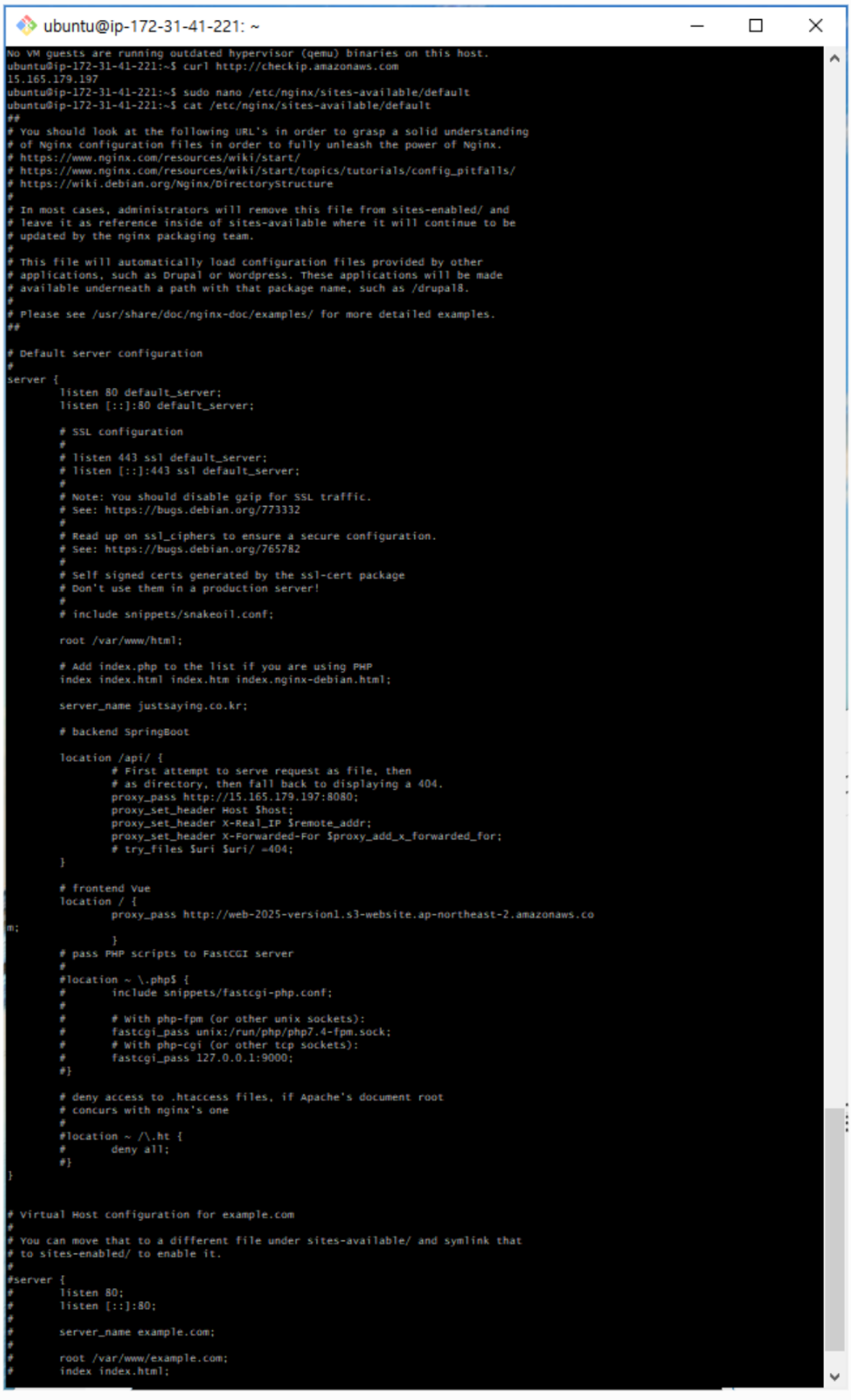
도메인(Domain) 설정 - IP주소를 사이트로 기억하기 쉽도록 이름을 사용
DNS(도메인 네임 시스템)는 도메인과 IP 주소를 연결해주는 시스템이다. 도메인은 단순한 이름이고 실제로 웹사이트가 동작하려면 IP주소가 필요하다 .

(도메인은 가비아에서 구입, 네임서버 변경 필요)




```sh
sudo apt install net-tools -y
```

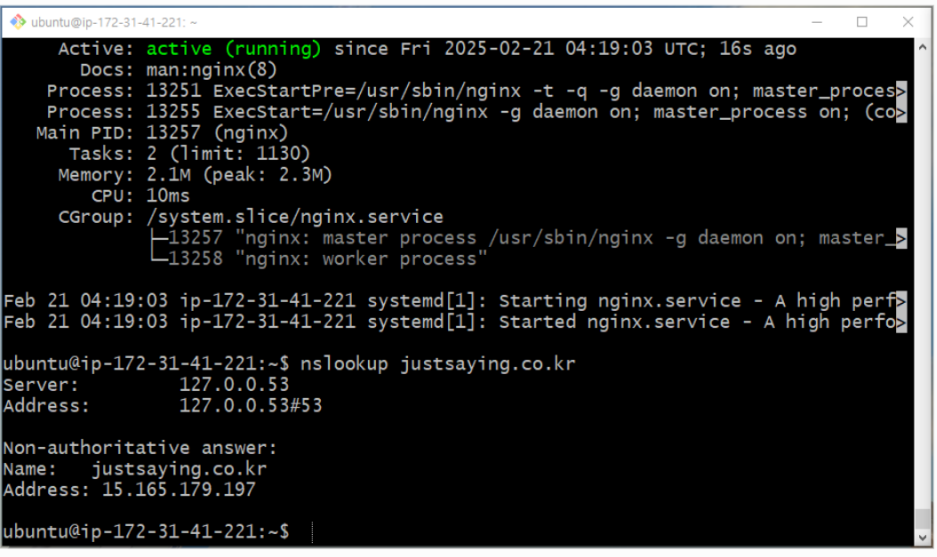
curl http://checkip.amazonaws.com




```

net-tools : 네트워크 도구 ,
curl http://checkip.amazonaws.com : EC2의 공인 IP확인
(탄력적 IP와 같은지 확인, 다르다면 A레코드 변경 필요)
EC2의 공인 IP를 확인 할 수 있다.
```

```nginx
server {
    listen 80;
    server_name justsaying.co.kr;

    # 백엔드(Spring Boot) 요청 처리
    location /api/ {
        proxy_pass http://15.165.179.197:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
    }

    # 프론트엔드(Vue) 요청 처리 - AWS S3에서 Vue 가져오기
    location / {
        proxy_pass http://web-2025-version1.s3-website.ap-
northeast-2.amazonaws.com;
    }
}

```

server_name justsaying.co.kr; 도메인 적용
proxy_pass:
/api/ :EC2(백엔드) 요청,
/ : S3(프론트엔드) 요청




```sh
sudo systemctl restart nginx
sudo systemctl status nginx
nslookup justsaying.co.kr

```

EC2(Elastic Compute Cloud) : 인터넷에 있는 컴퓨터(서버)
S3(Simple Storage Service) : 파일 저장소
* EC2 서버에서 모든 파일을 저장하면 서버가 느려지기 때문에 S3에 파일을 올려두고 필요할때만 가져오면 더 빠르고 안전하다.

* s3 (vue)
http://web-2025-version1.s3-website.ap-northeast-2.amazonaws.com/
* ec2(server)
 http://3.34.90.124:8080 으로 들어가면 성공

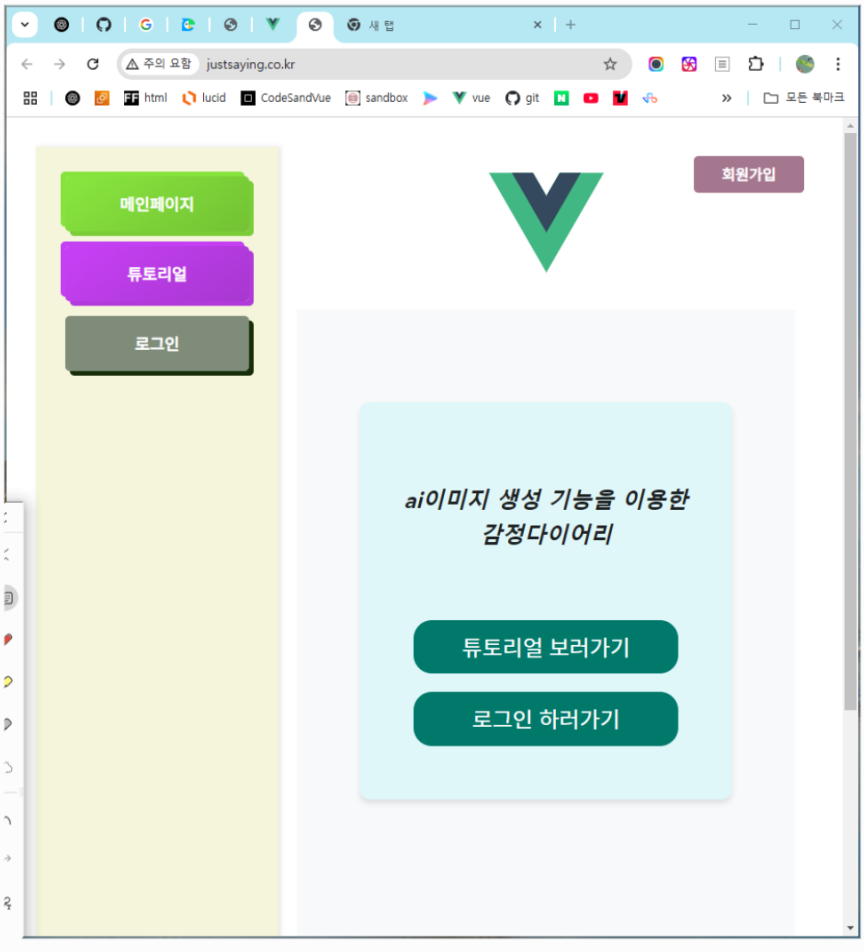
탄력적 ip 15.165.179.197

* Nginx가 80번 포트에서 사용자의 요청을 받고
*vue 프론트엔드는 s3에서 가져오고
spring boot 백엔드는 EC2 (8080포트)로 연결

Nginx가 모든 요청을 받아서 프론트엔드는 S3, 백엔드는 EC2로 보낸다.

* 퍼블릭 IP를 사용하다 다시 가동하면 ip가 바뀌는 문제때문에 탄력적 ip 부여
 출력 결과가 탄력적 IP(Elastic IP)와 동일해야 함!
만약 다르면 A 레코드를 탄력적 IP로 변경해야 함.

* http://justsaying.co.kr 들어가면 화면이 뜨면 성공



* 실행한 명령어 자체를 저장하는 방법 history > git_commands.txt*