

EC2 접근 트러블 슈팅

▼ Version 1.0.0

```
#application.properties
spring.application.name=backend

server.port=8080

# jdbc:postgresql://localhost:5432/aid 에서 localhost가 아닌 db인 이유.
# - localhost:
#   - 로컬에서 작업할 때 사용하며 호스트 머신 자신의 5432 포트에 접근한다.
# - db:
#   - EC2 내의 compose.yaml에 작성된 postgresql의 services 이름이다.
#   - 즉, EC2 내의 db 컨테이너(서비스명)의 5432 포트에 연결을 하겠다는 요청이다.
#   - SSH 터널링이 된 상태이기 때문에 원활히 연결이 된다.
spring.datasource.url= jdbc:postgresql://db:5432/aid
spring.datasource.username=postgres
spring.datasource.password=i13b209tpass

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

spring.jwt.secret=Aid-Secret-Key-For-Ssafety-Gongtong-Project-By-Team-209 #JWT 토큰 서명을 위한 시크릿 키
spring.jwt.token.access-expiration-time=3600000 # 1시간 (ms 단위)
spring.jwt.token.refresh-expiration-time=1209600000 # 14일
```

▼ Version 1.0.1 (2025-08-05)

- 이전에는 연결이 됐었는데 왜인지 모르겠다.
- 이번에는 SSH 터널링을 통한 DB 연결을 위해 더 정확한 경로를 지정하였다.
 - spring container
 - 로컬 PC(host.docker.internal)의 포트 5432
 - EC2의 5432 포트

→ postgres-db 컨테이너 5432 포트

```
#application.properties
spring.application.name=backend

server.port=8080

# host.docker.internal: 스프링 컨테이너에서 바라보는 호스트 머신.
# spring container
# → 로컬 PC(host.docker.internal)의 포트 5432
# → EC2의 5432 포트
# → postgres-db 컨테이너 5432 포트
spring.datasource.url=jdbc:postgresql://host.docker.internal:5432/aid
spring.datasource.username=postgres
spring.datasource.password=i13b209tpass

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

spring.jwt.secret=Aid-Secret-Key-For-Ssafy-Gongtong-Project-By-Team-209 #JWT 토큰 서명을 위한 시크릿 키
spring.jwt.token.access-expiration-time=3600000
spring.jwt.token.refresh-expiration-time=1209600000
```