비밀번호는 AWS만 믿으라구

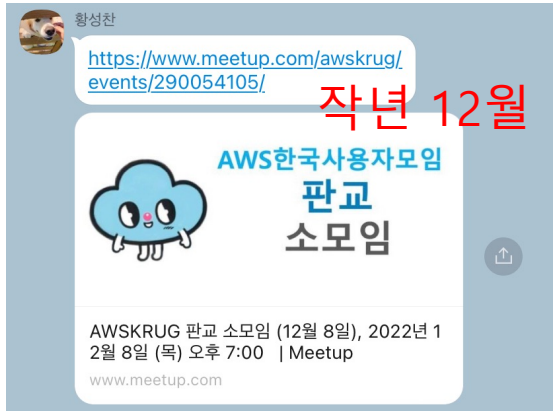
(aws secretsmanager 를 활용한 보안강화)

발표 전 여담

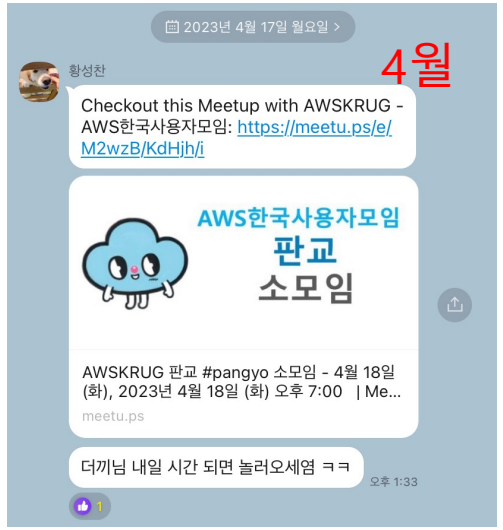
저에게 이쁜 알람 요정이 있습니다.

그의 이름은 황성찬

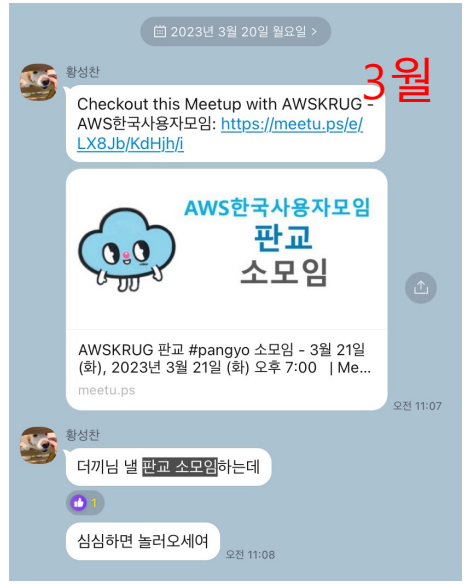
AWS 판교 소모임 알람봇(황성찬)



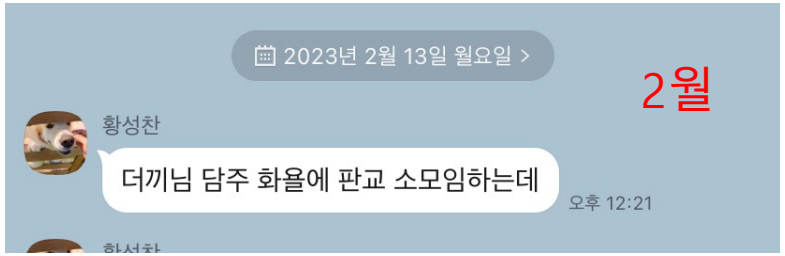
작년 12월



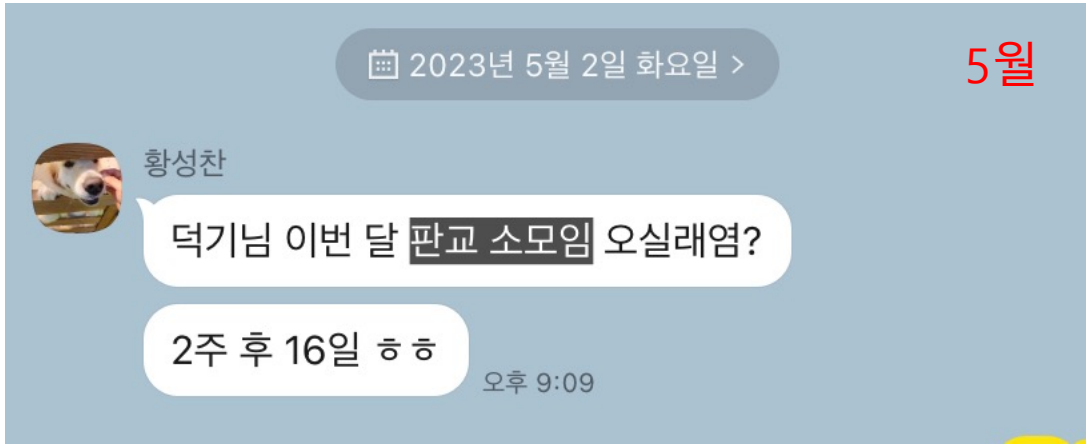
4월



3월



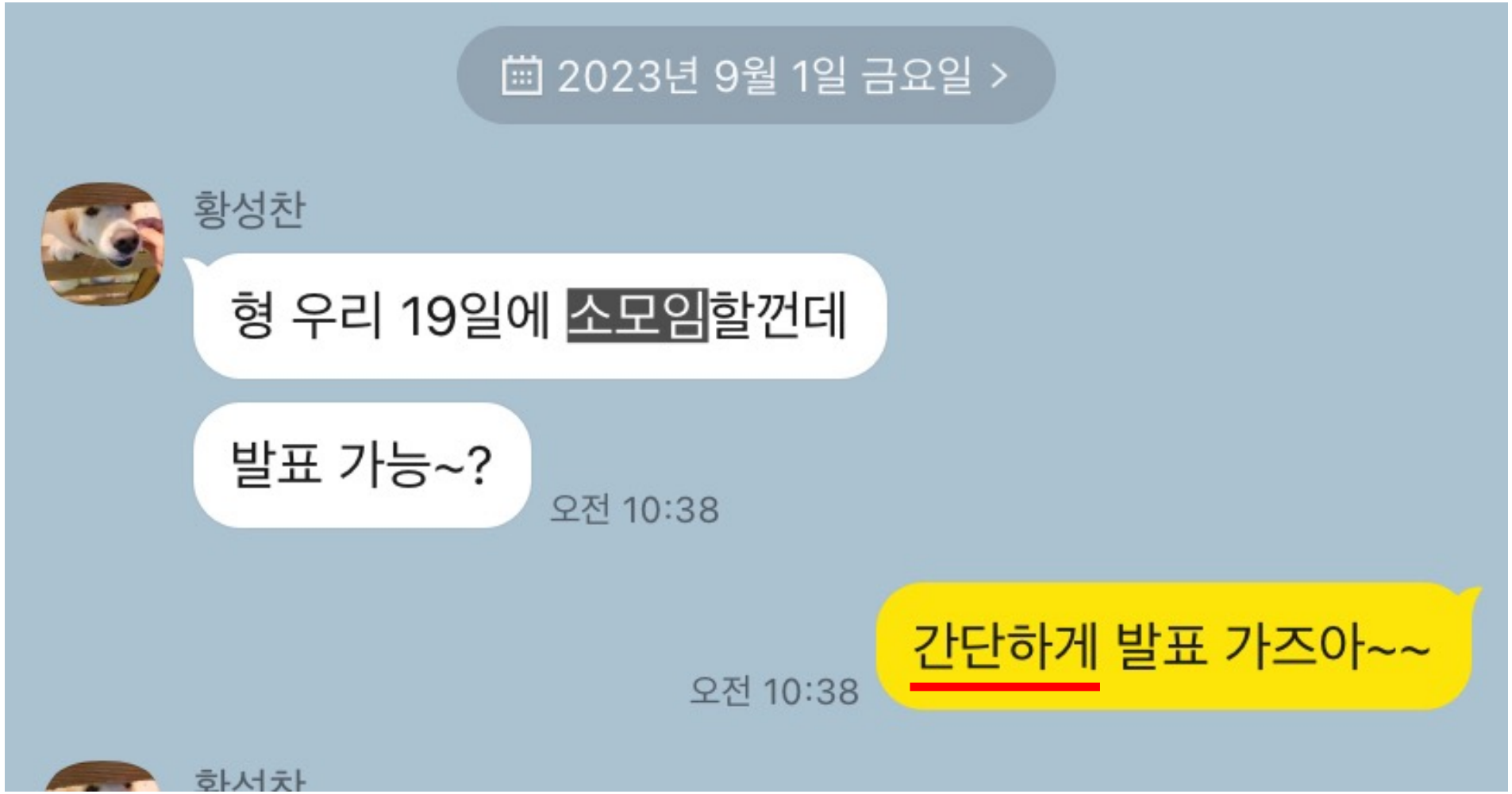
2월



5월

쿠팡이츠 쿠폰(신사임당)도 받음

은혜를 갚자.


2023년 9월 1일 금요일 >
황성찬
형 우리 19일에 소모임할껀데
발표 가능~?
오전 10:38
간단하게 발표 가즈아~~
오전 10:38

은혜를 갚자.

💁‍♂️ 메인 발표

- 주제: 비밀번호는 AWS만 믿으라구 feat. AWS Secretsmanager ????
- 발표자 : 안덕기

🥳 행사 마무리 : 메인 발표 종료 후 가볍게 잡담 나누며 행사를 마무리 합니다.

황성찬 ~~네 이놈~~

결론 : 속아서 메인 발표가 되었음

가벼운 마음으로 듣고 가세요!!

패스워드 관리

비밀번호를 yml 파일에 정의

```yaml
spring:
  datasource:
    hikari:
      read-write:
        driver-class-name: org.mariadb.jdbc.Driver
        jdbc-url: jdbc:mariadb://localhost:3306/review?DB_CLOSE_ON_EXIT=FALSE&write
        username: root
        password: 1234
        maximum-pool-size: 1
        validation-timeout: 2000
        idle-timeout: 60000
        connection-timeout: 3000
        max-lifetime: 900000
      read-only:
        driver-class-name: org.mariadb.jdbc.Driver
        jdbc-url: jdbc:mariadb://localhost:3306/review?DB_CLOSE_ON_EXIT=FALSE&read
        username: root
        password: 1234
        maximum-pool-size: 1
        validation-timeout: 2000
        idle-timeout: 60000
        connection-timeout: 3000
        max-lifetime: 900000
```

Spring CLI 활용을 통한 암호화

```
hugh 🐵  ~/Development/config/spring-2.7.9/bin
▶ ./spring encrypt --key aws '1234'
6795d8fa8458455b663453ceaf59cb621116921530e134eb05e362444490eb51
```

Spring Cloud Encrypt

```yaml
encrypt:
  key: ${CONFIG_CLIENT_ENCRYPT_KEY}

spring:
  datasource:
    hikari:
      read-write:
        driver-class-name: org.mariadb.jdbc.Driver
        jdbc-url: jdbc:mariadb://localhost:3306/review?DB_CLOSE_ON_EXIT=FALSE&write
        username: root
        password: "{cipher}6795d8fa8458455b663453ceaf59cb621116921530e134eb05e362444490eb51"
        maximum-pool-size: 1
        validation-timeout: 2000
        idle-timeout: 60000
        connection-timeout: 3000
        max-lifetime: 900000
```

단점

- 감사 때 혼날 수도 있음
- 패스워드가 변경되는 시점과 워크로드가 재가동 되는 시점이 맞지 않으면 장애로 번질 수 있음
- Spring Cli는 양방향 암/복호화이기 때문에 패스워드를 알아낼 수 있음

아래 제품들을 통해 자동화하자.

AWS Secretes Manager

AWS Key Management Service

Service Account

AWS SDK

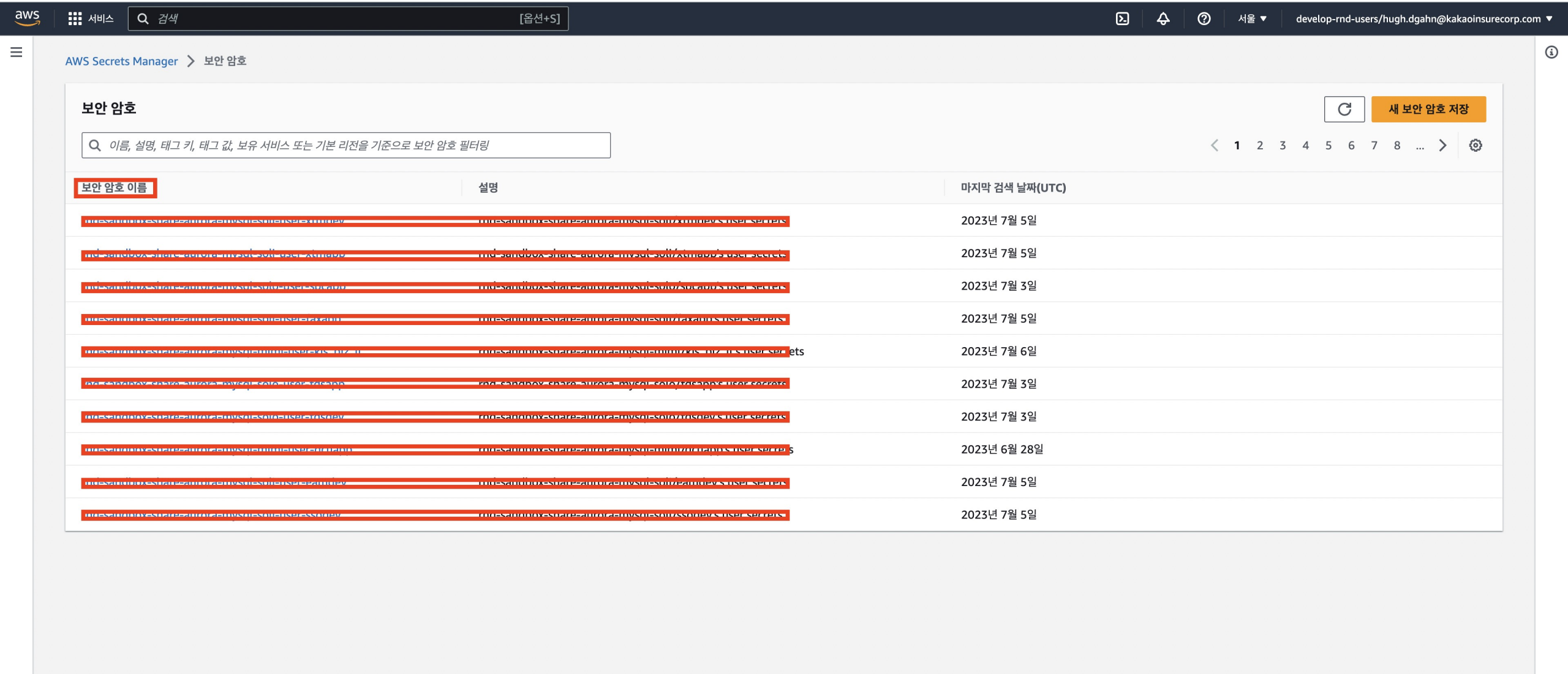
AWS Security Token Service

AWS Single Sign-On

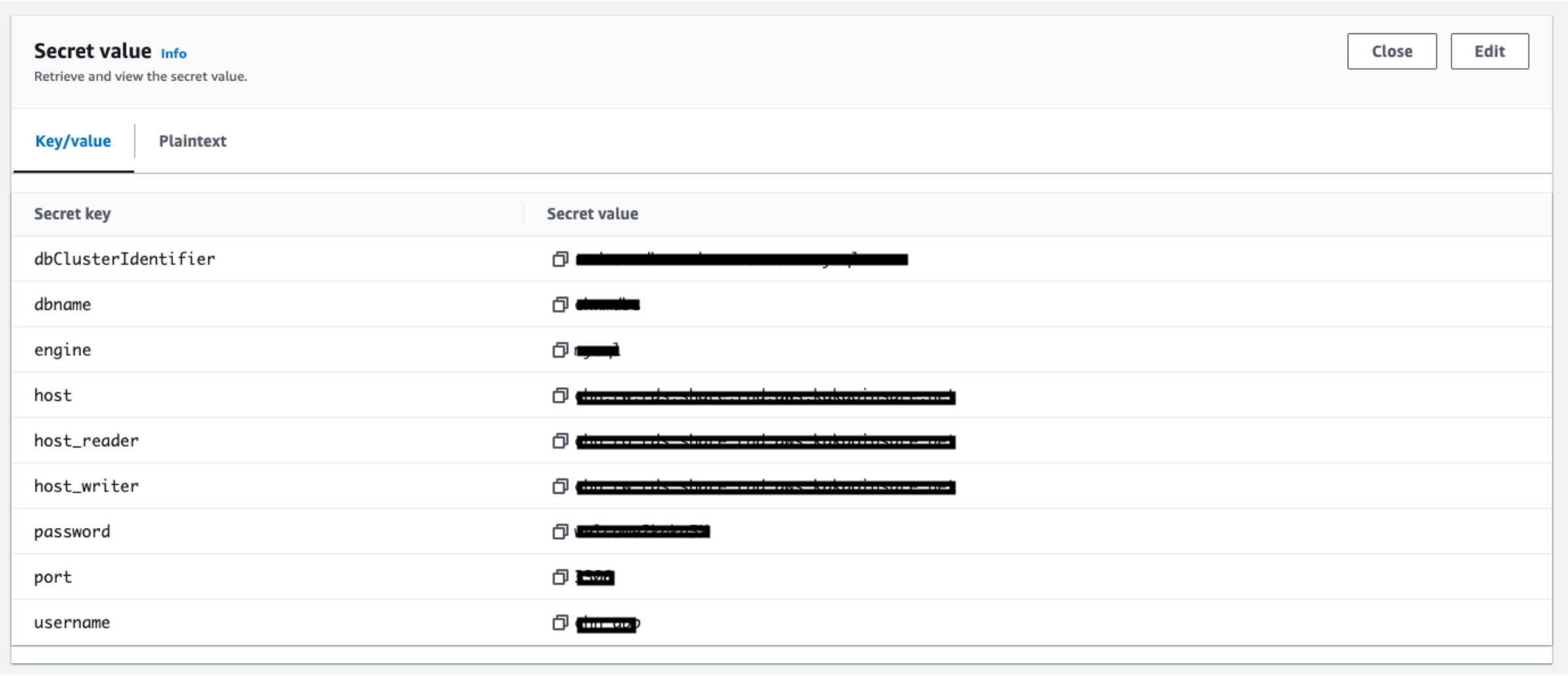
AWS assume role

작동 원리

AWS Secrets Manager



AWS Secrets Manager



접근하려면 인증과 권한이 필요함.

권한에 따라 할 수 있는 것이 다름

- manager
  - AWS Secrets Manager 접근 가능

- user
  - AWS Secrets Manager 접근 불가능
  - Assume role을 통해서 접근 가능 예정

- 즉, 권한에 따라 접근이 다르다.

제가 테스트해야 했던 것

- EKS의 Pod가 AWS Secrets Manager 접근 가능하게 해야 함
  - 서버 운영을 위하여

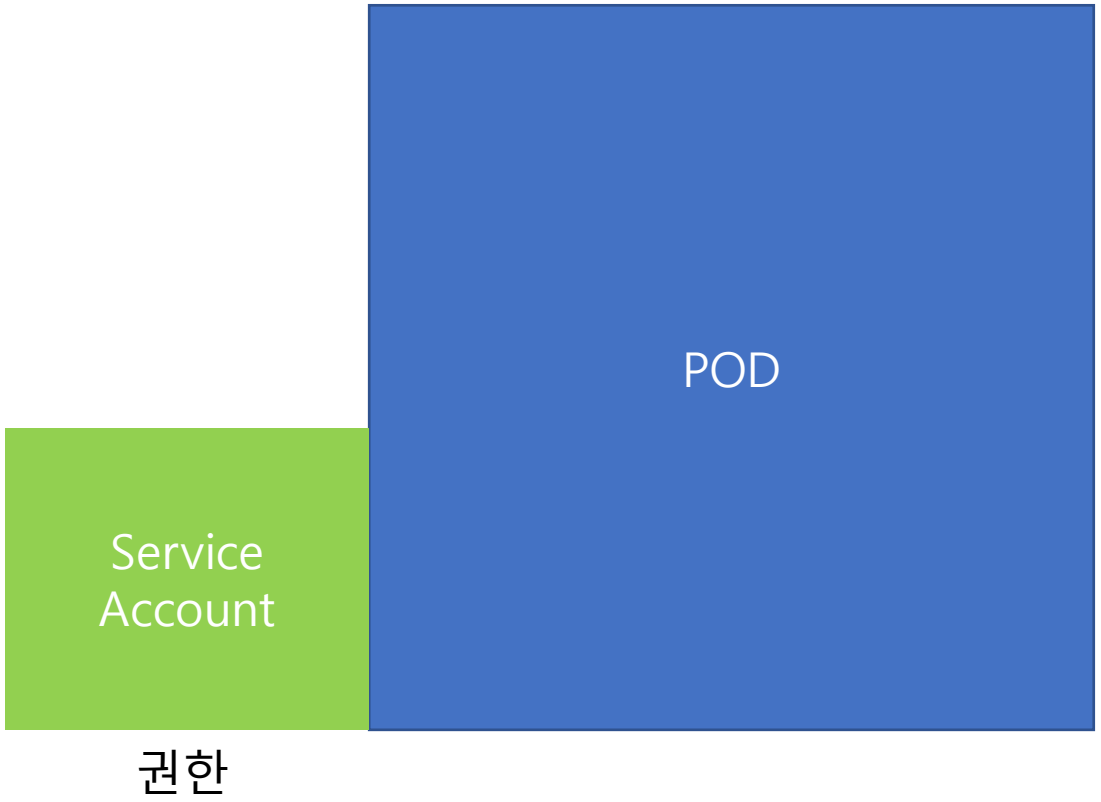
- 맥북(로컬)에서 AWS Secrets Manager 접근 가능하게 해야 함
  - 개발을 위하여

EKS와 로컬(맥북)은 다르다

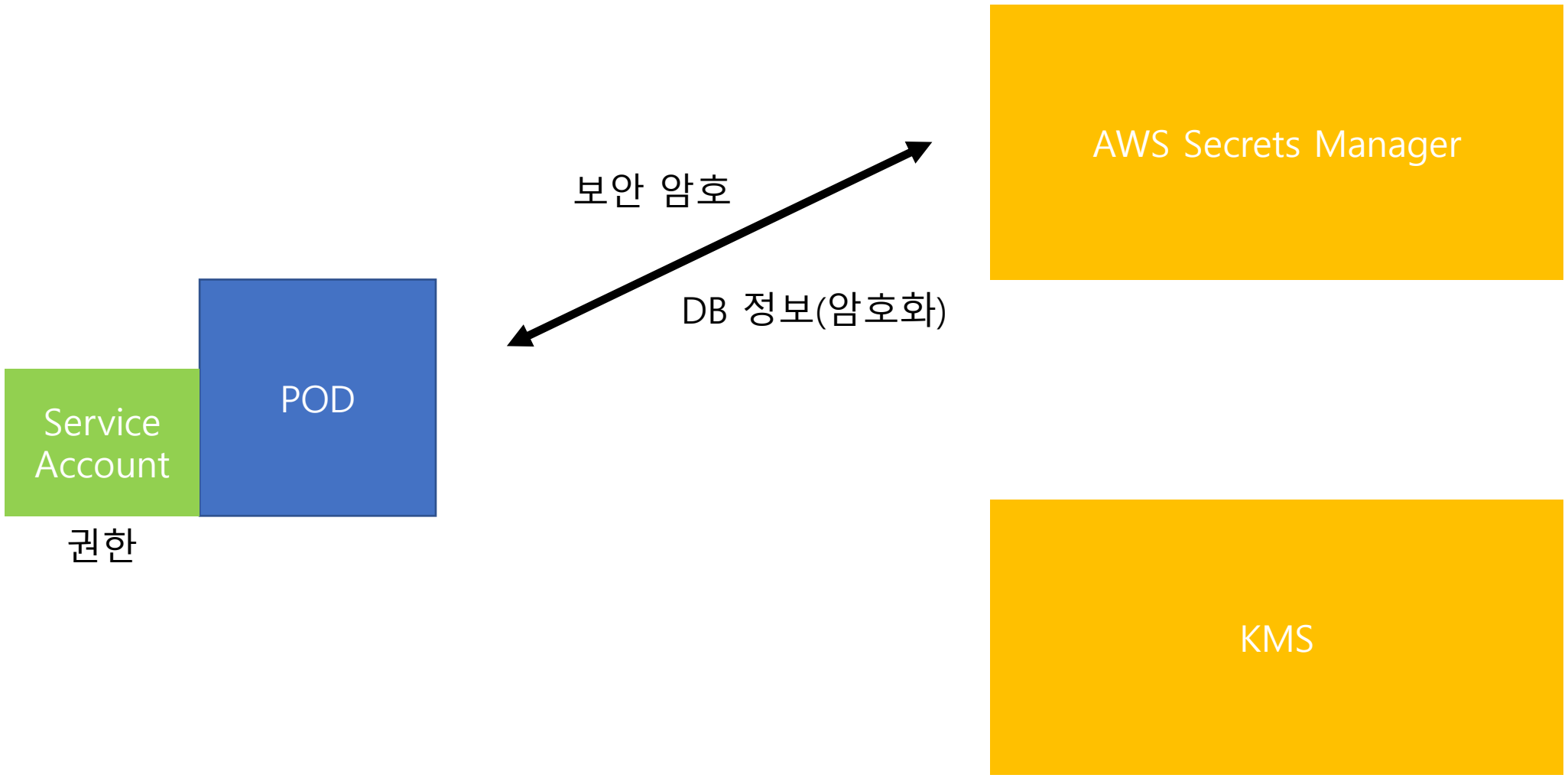
- AWS Secrets Manager에 접근하려면 권한이 있거나 -> EKS
- Access Key, Secret Access Key가 있거나
- Token 값이 필요함. -> 로컬 환경

근데 AWS Secrets Manager 값은

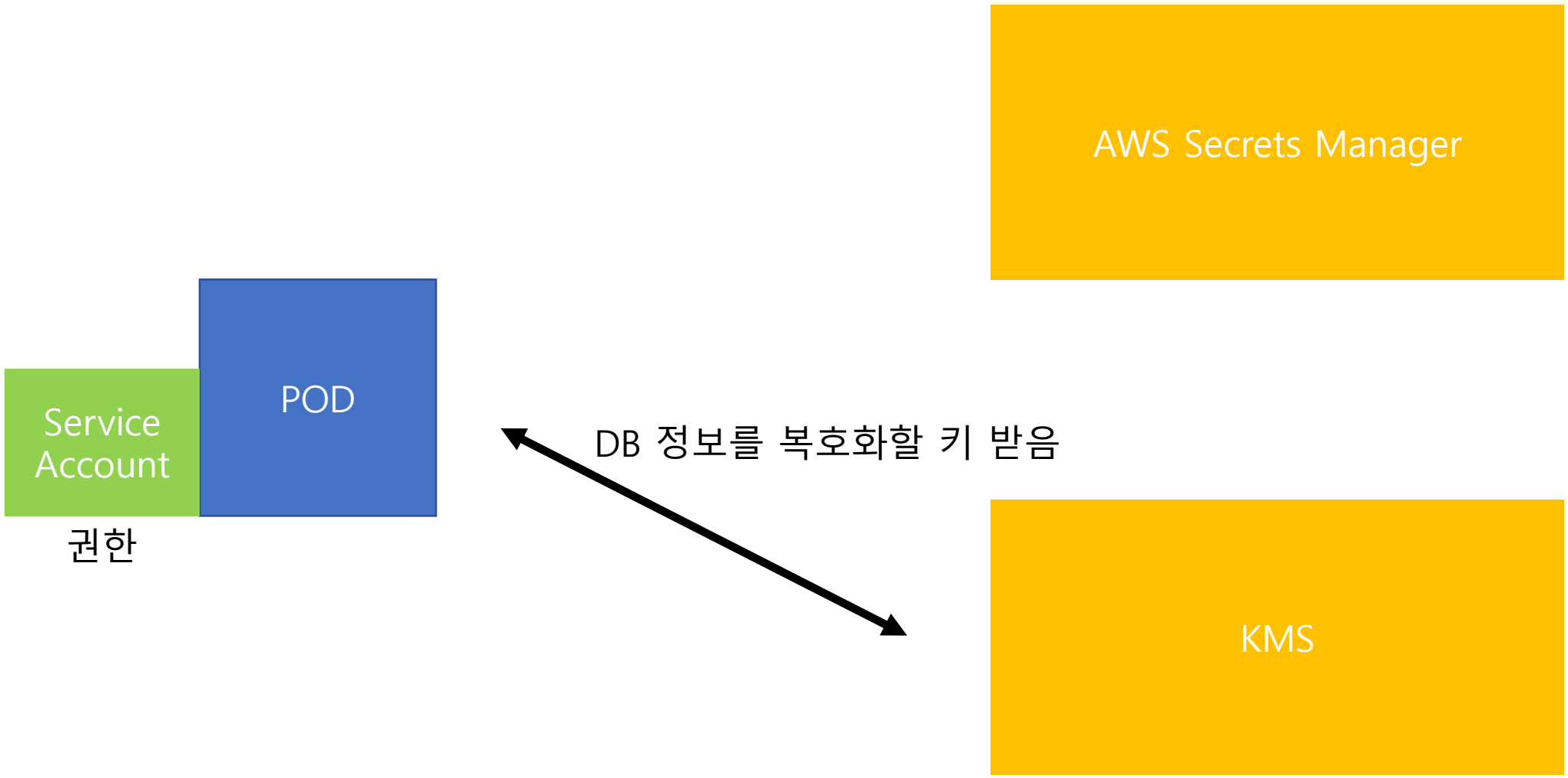
EKS는??



EKS에서는


AWS Secrets Manager
보안 암호
DB 정보(암호화)
POD
Service Account
권한
KMS

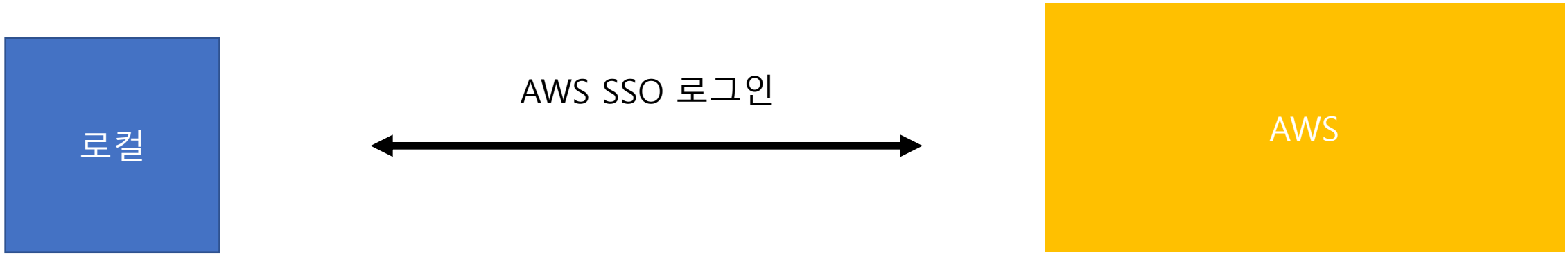
EKS에서는


AWS Secrets Manager
POD
Service Account
권한
DB 정보를 복호화할 키 받음
KMS

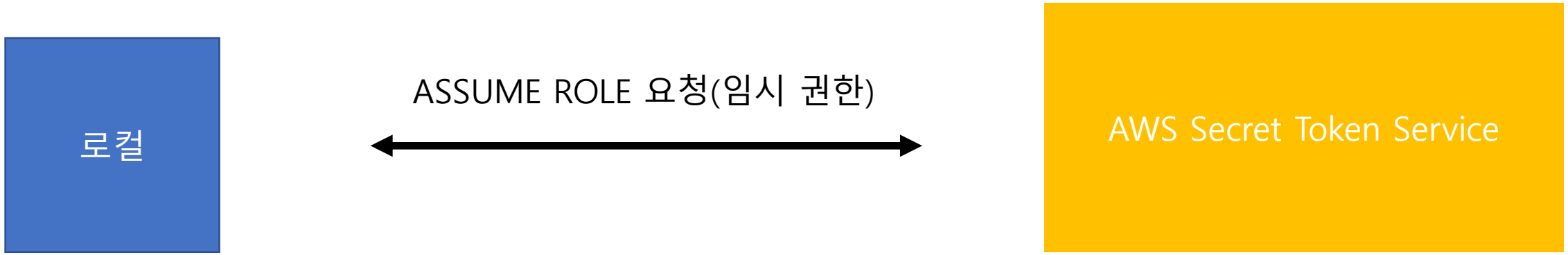
EKS에서는

맥북(로컬)에서는?

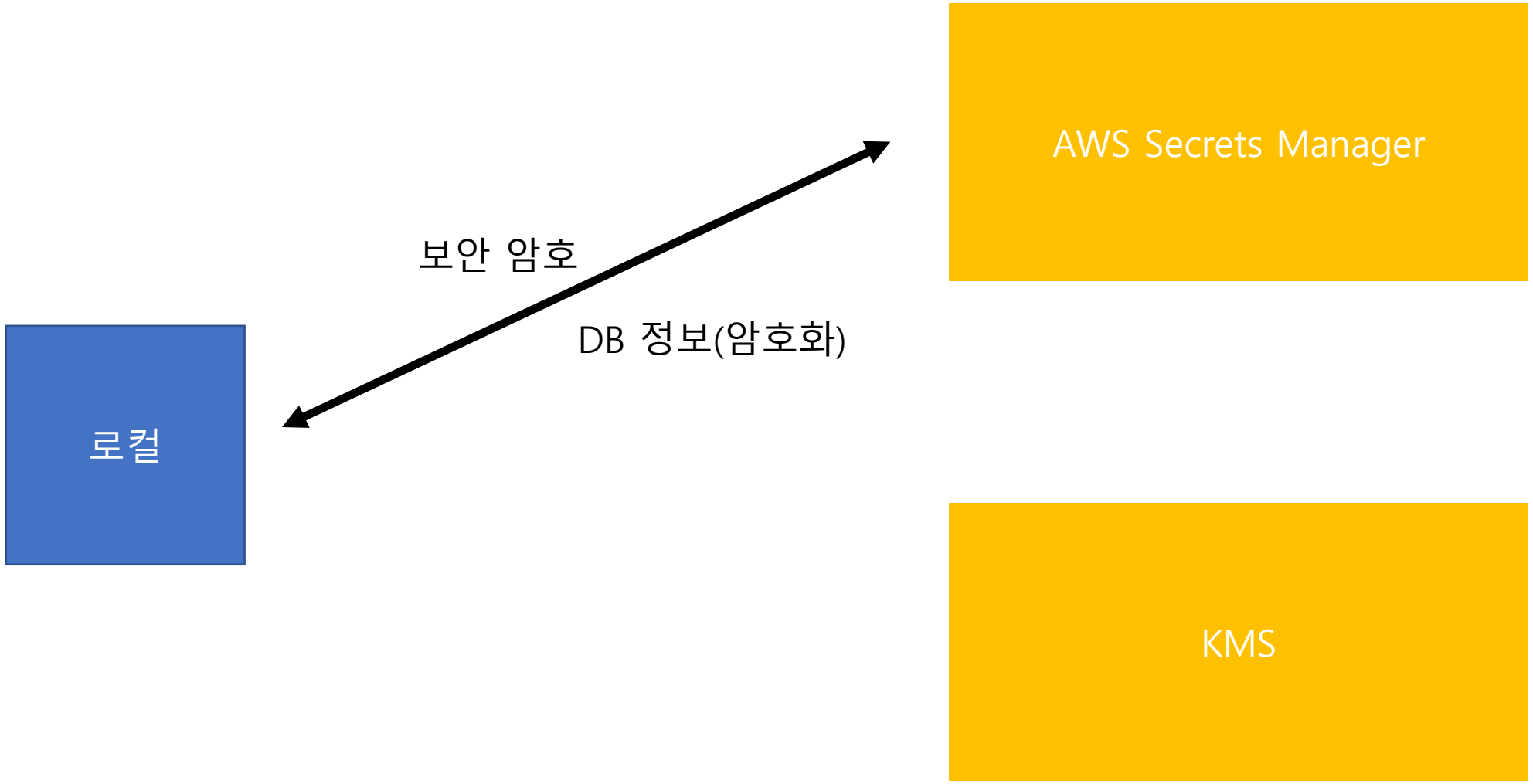
로컬에서는



로컬에서는

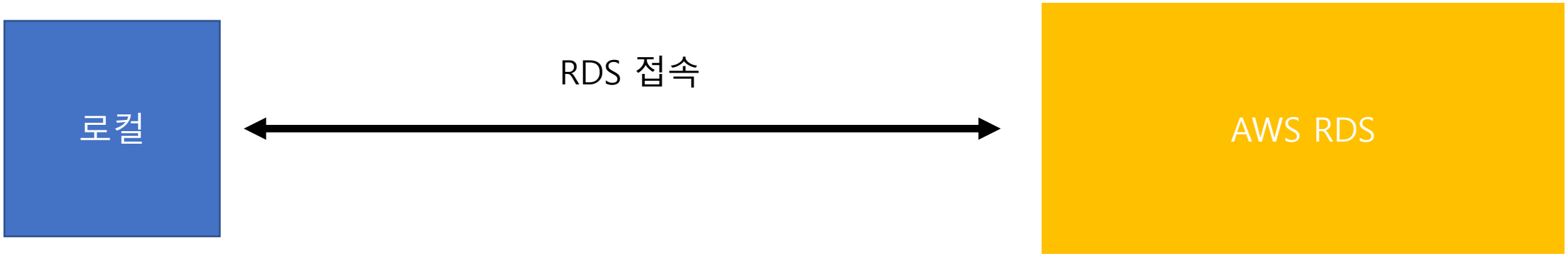


로컬에서는



로컬에서는

로컬에서는



장점

- 보안팀의 적극적인 지지
- 패스워드 관리는 DBA에게 완전히 양도

단점

- 권한이 없으면 확인할 수 있는게 없다.
- 로컬에서 서버를 실행하기 위해서는 주기적으로 로그인과 assume role를 획득 해야된다.

AWS Secret Manager 말고는?

Google Secret Manager
Configuration Provider
for ASP.NET Core

Azure Key Vault

그래도 AWS 제품 써주실거죠?

코드 구현

회사 코드라 공개를 못하네요..

하지만 AWS SDK와 함께라면 두려울게 없죠

https://github.com/aws/aws-sdk-java

감사합니다.