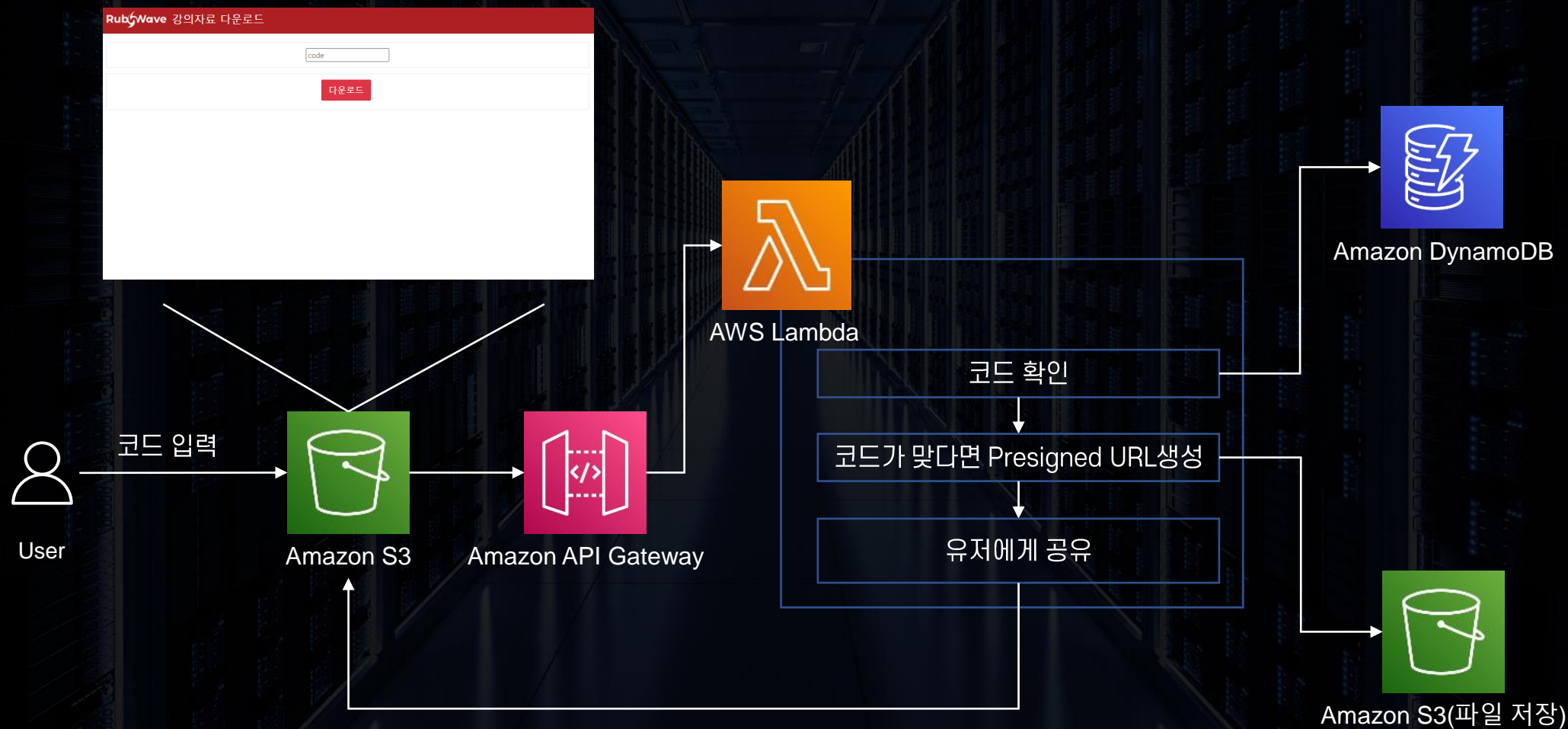
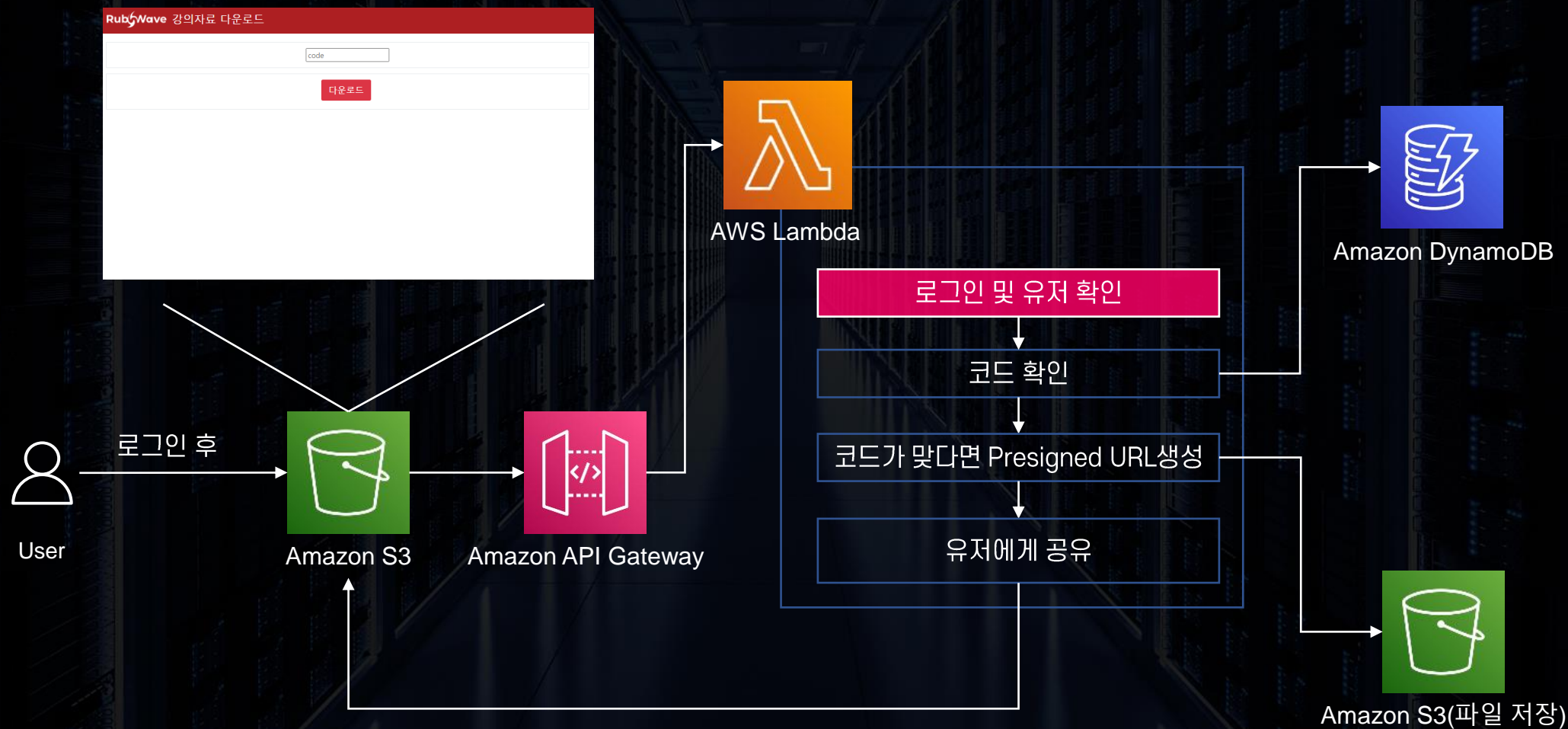
 AWS Serverless 기반 아키텍처링 및 개발

“JSON Web Token”

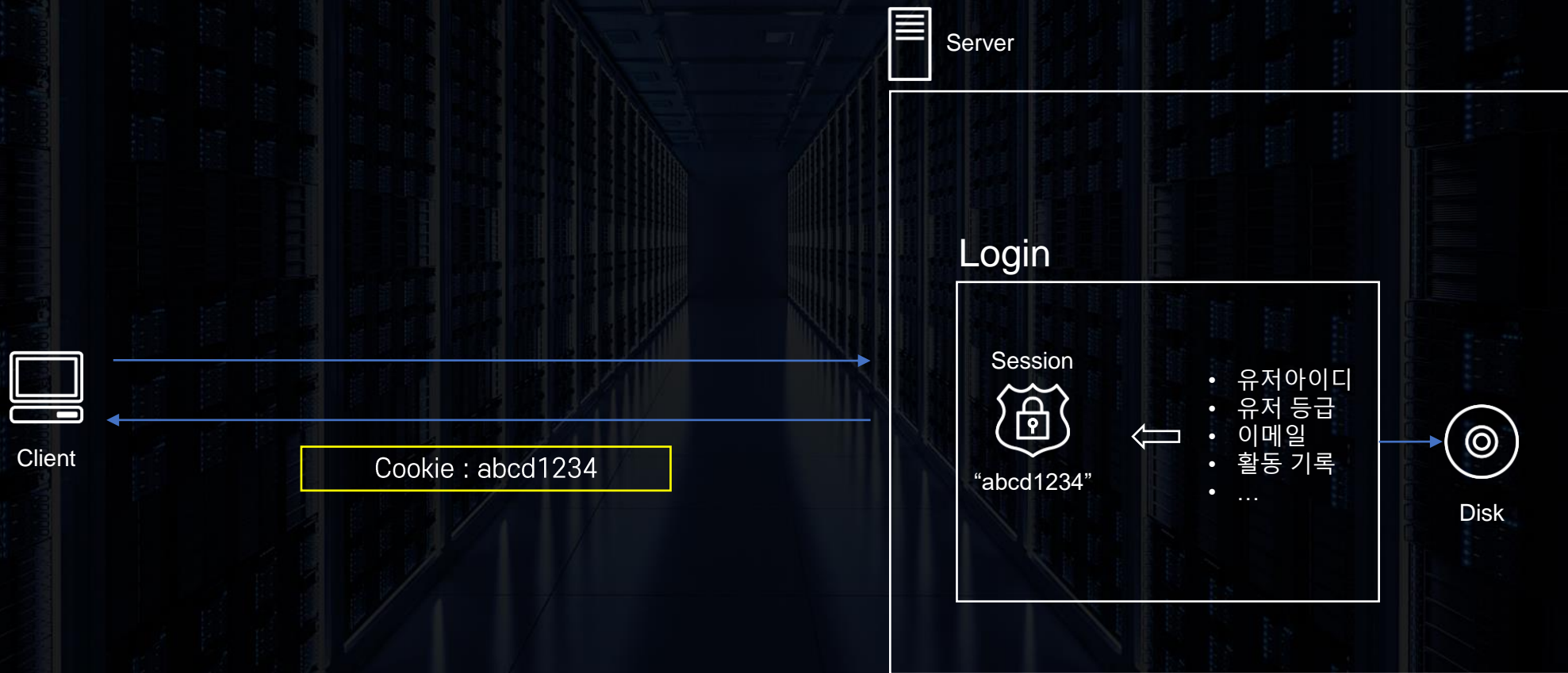
> 아키텍처



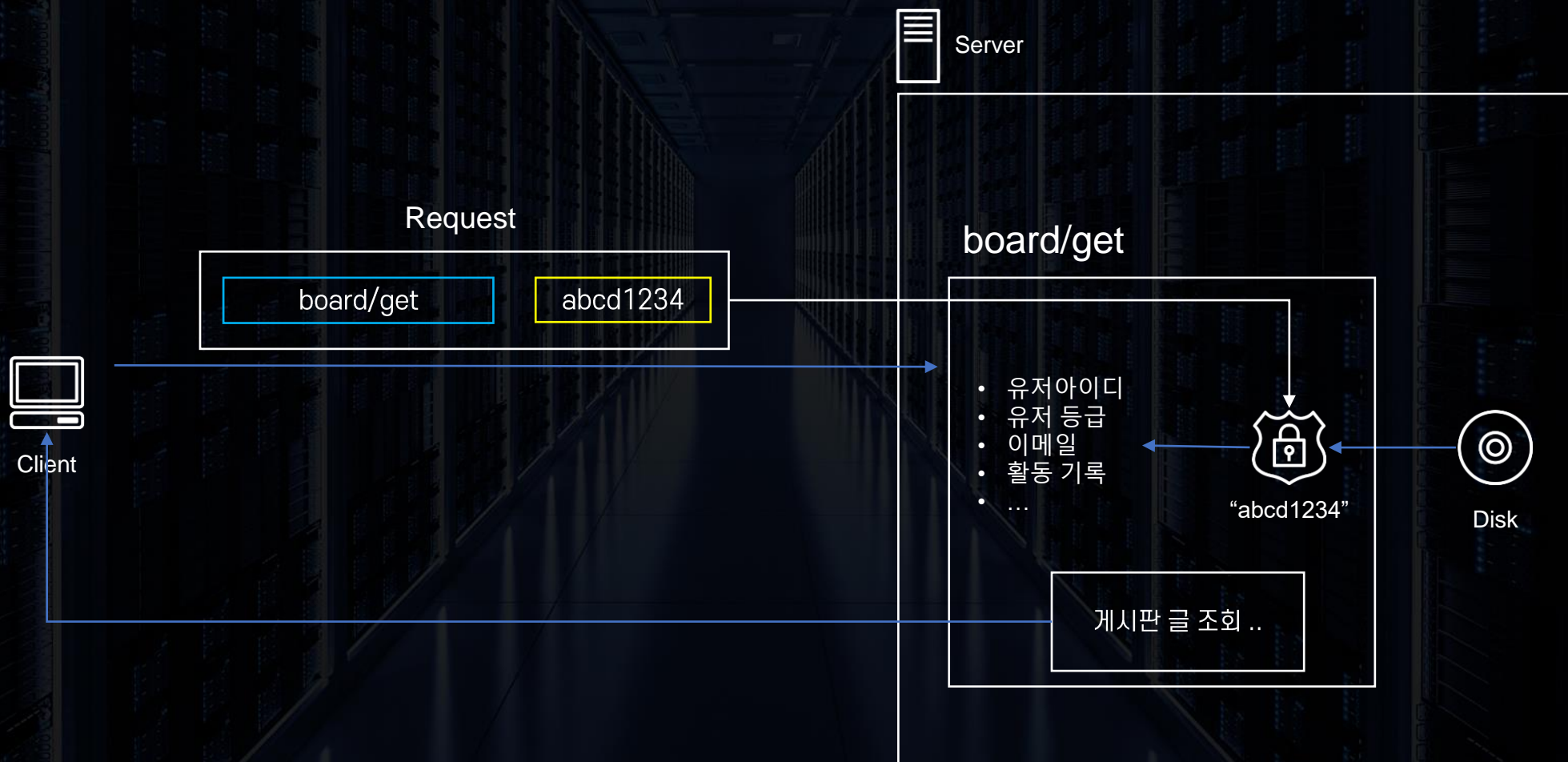
> 로그인 필요한 경우?



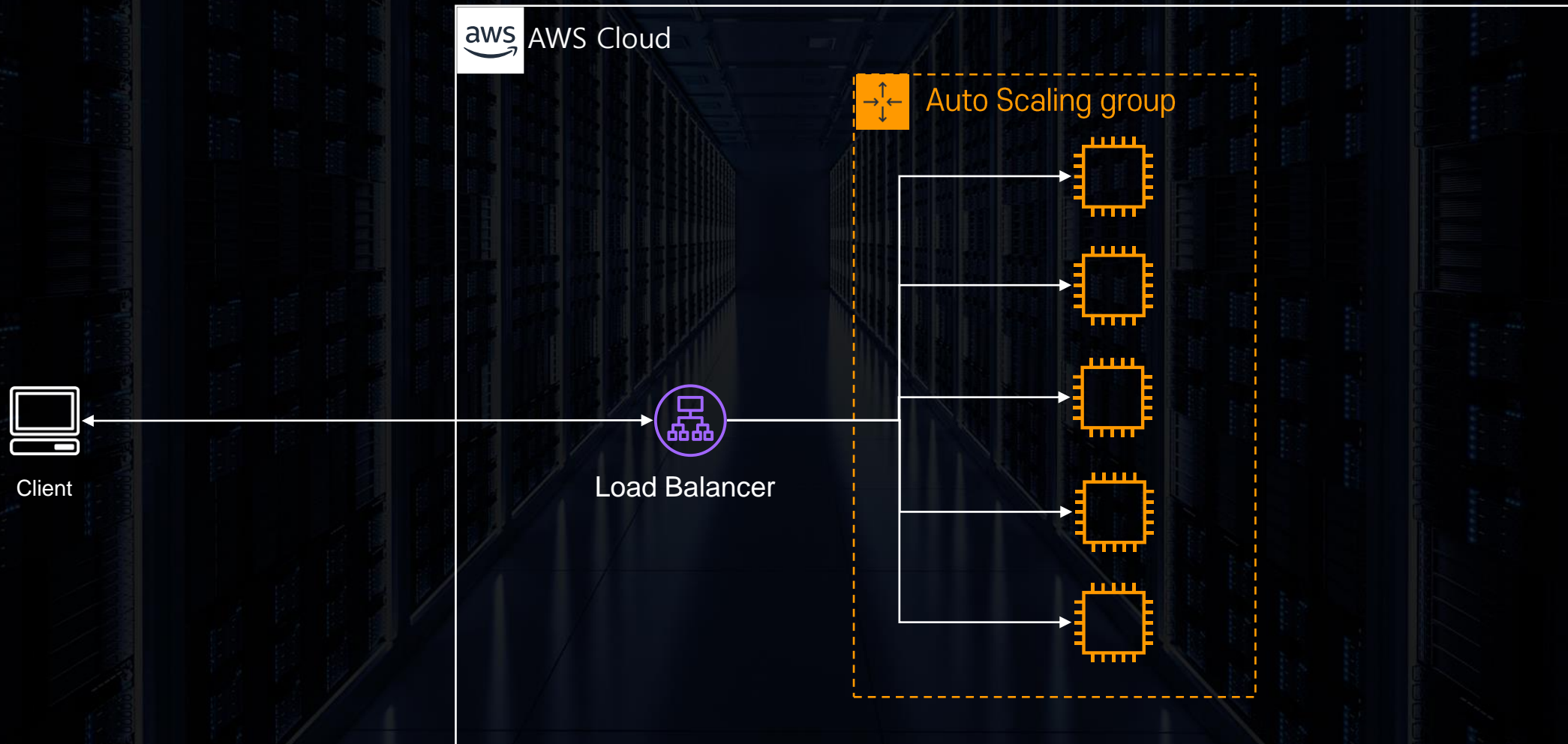
> 웹 서버의 세션



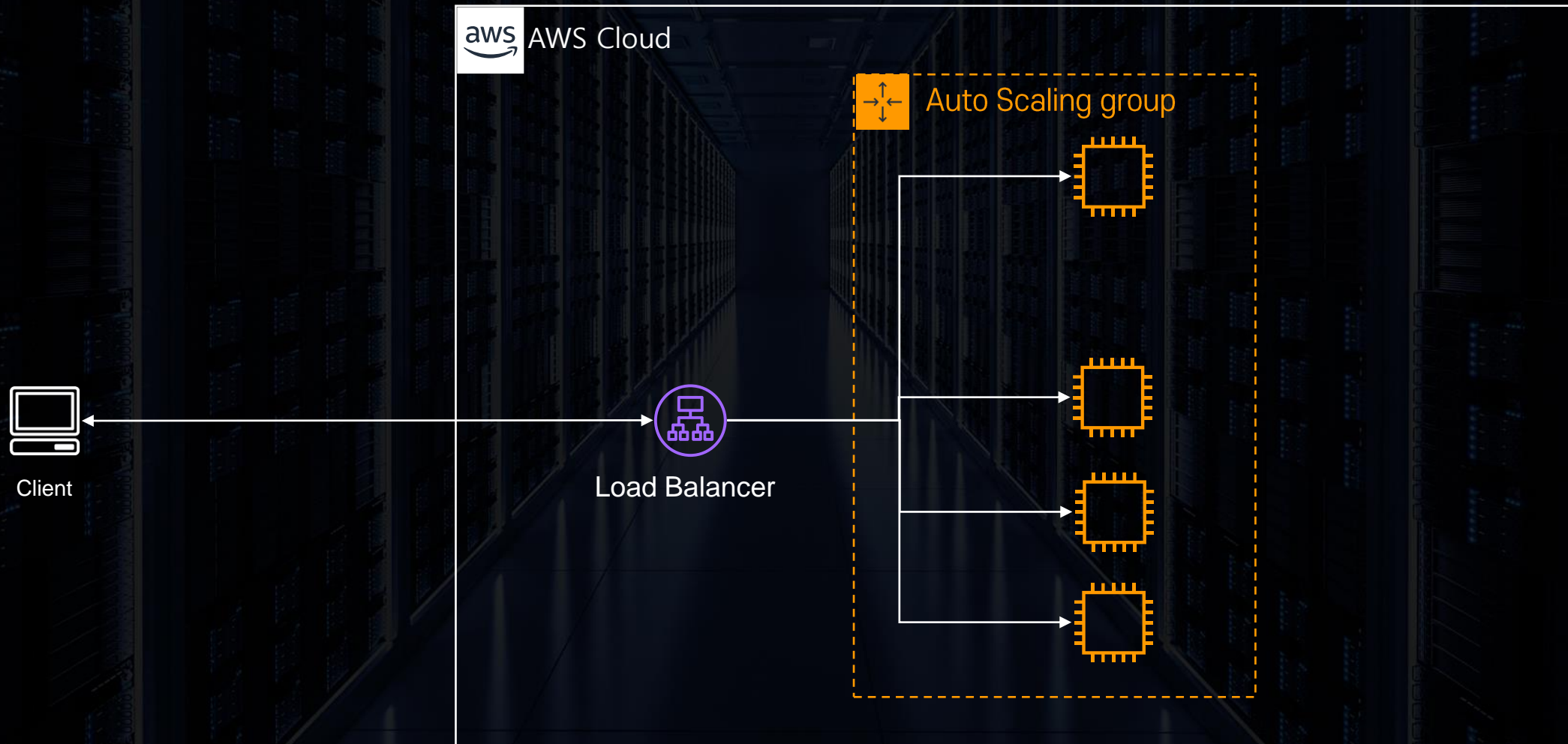
> 웹 서버의 세션



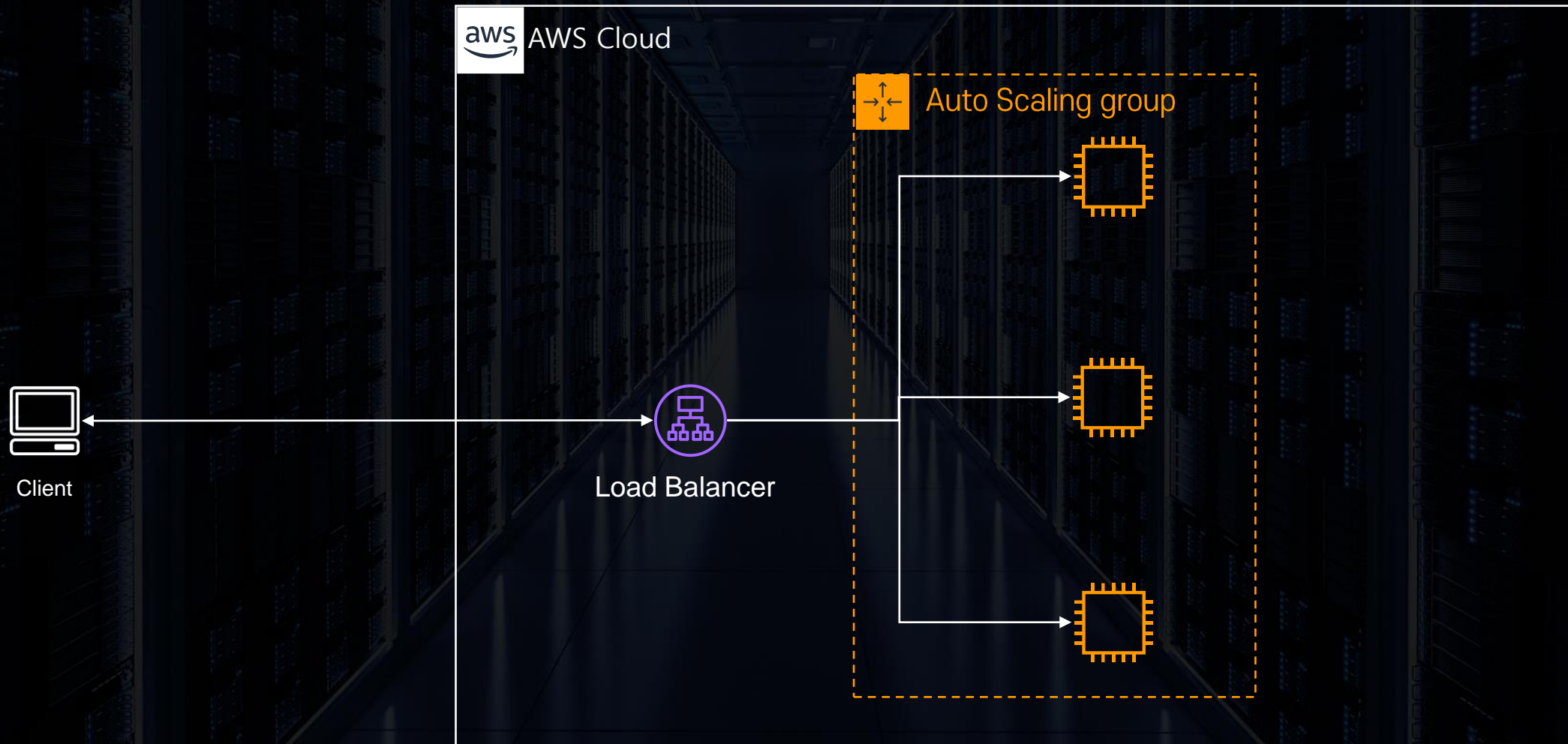
> 클라우드 환경에서 세션 관리



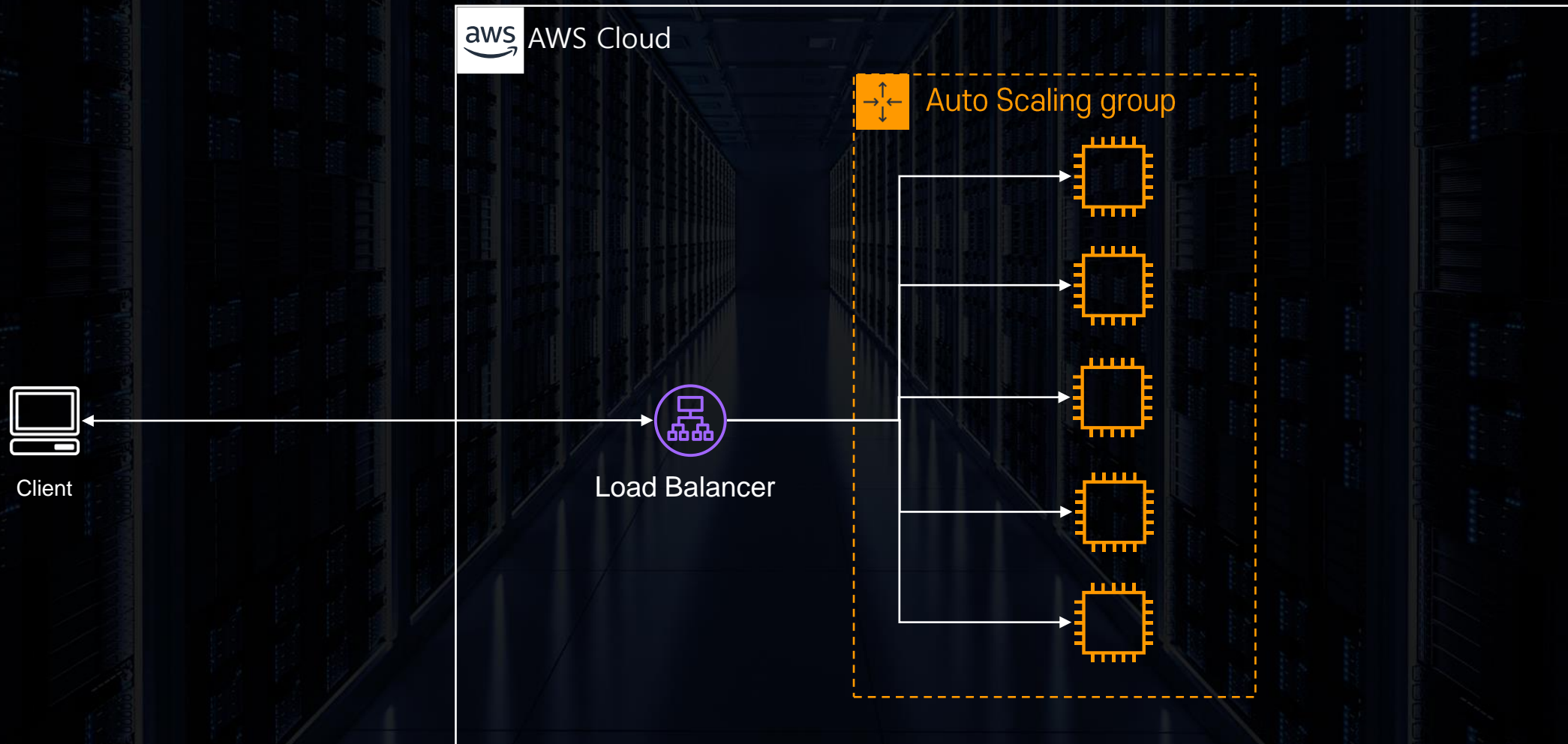
> 클라우드 환경에서 세션 관리



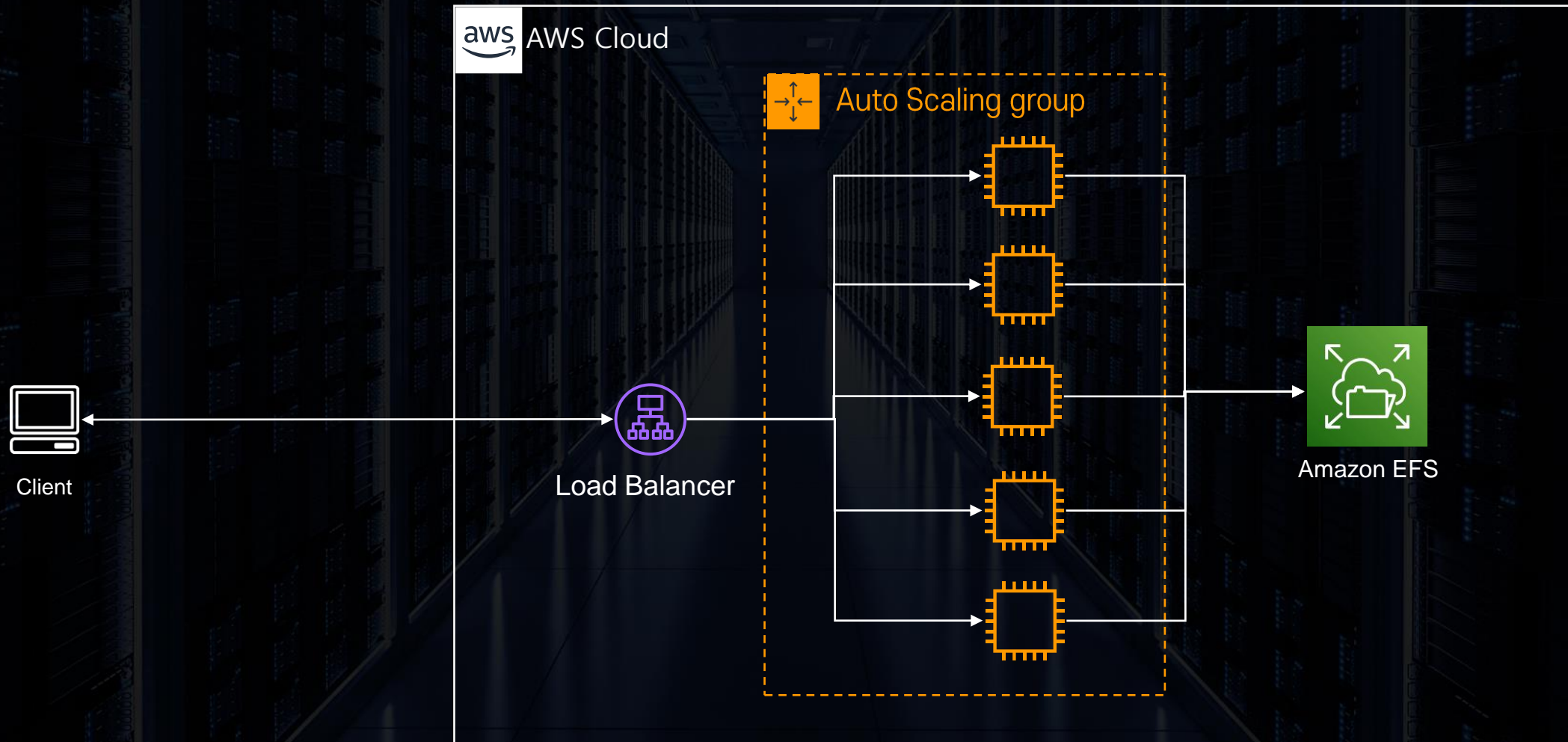
> 클라우드 환경에서 세션 관리



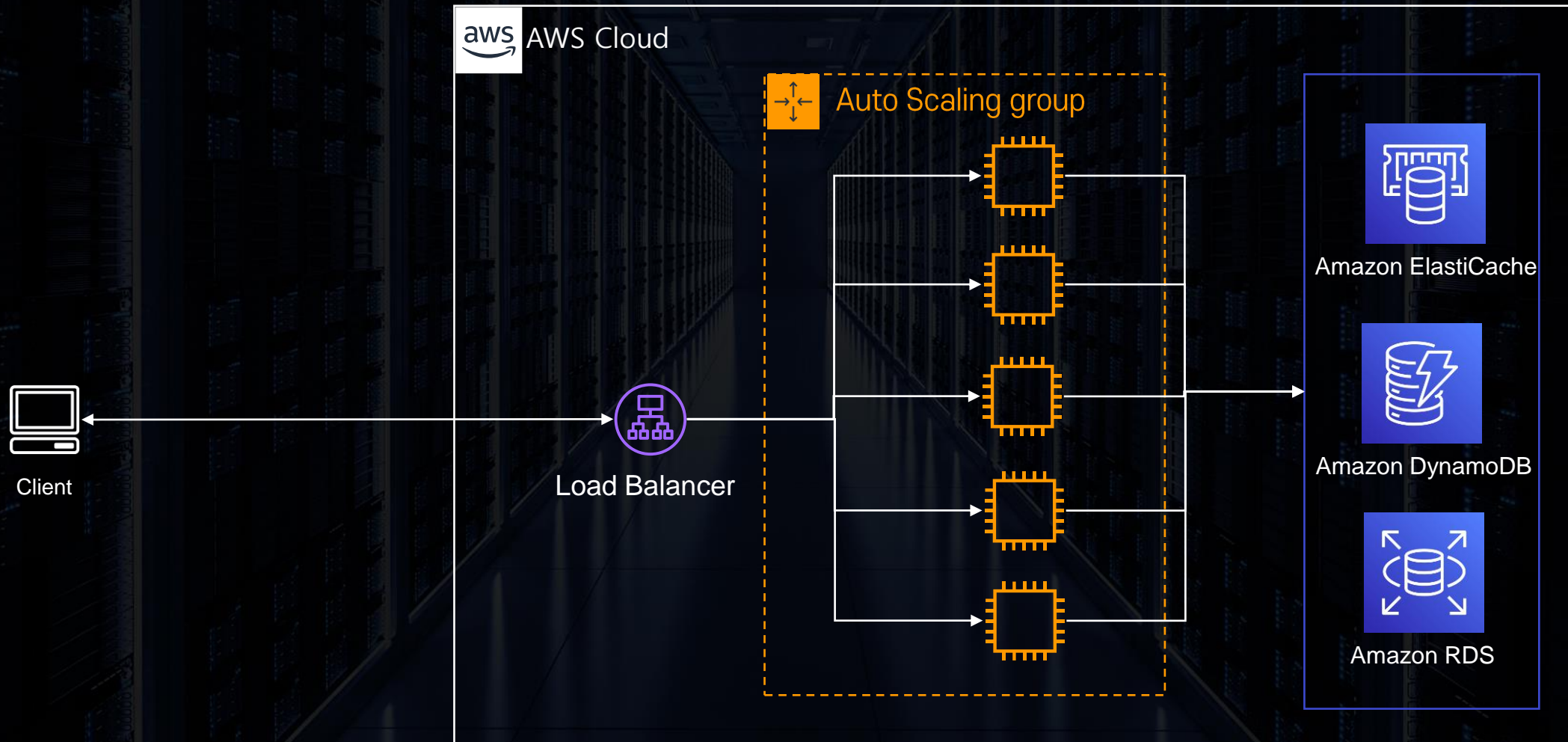
> 클라우드 환경에서 세션 관리



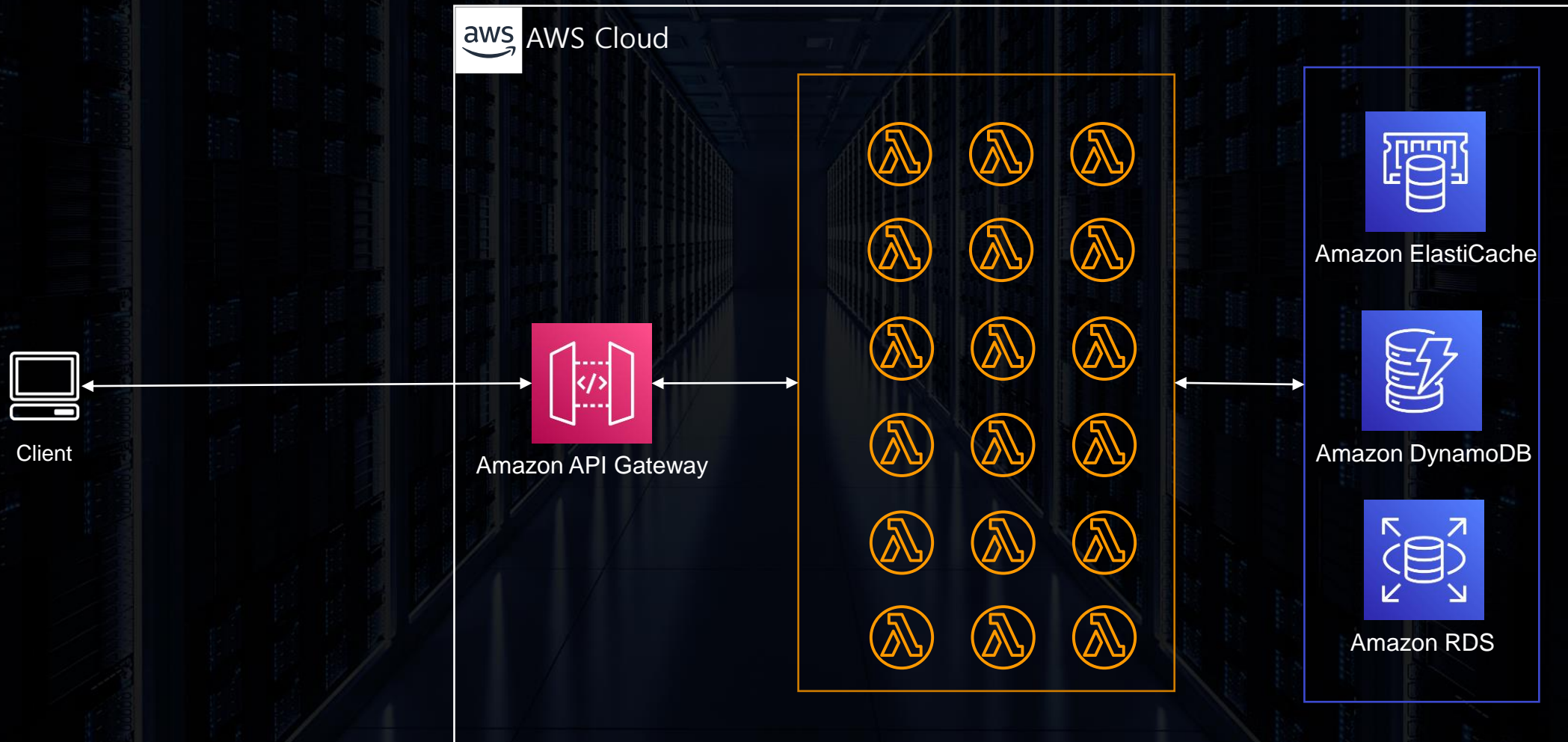
> 클라우드 환경에서 세션 관리



> 클라우드 환경에서 세션 관리



> Serverless 환경에서 세션 관리



› 세션 정보를 클라이언트가 보관하는 방법

- 유저아이디
- 유저 등급
- 이메일
- 활동 기록
- ...



Client

AWS Lambda



- 유저아이디
- 유저 등급
- 이메일
- 활동 기록
- ...

> JSON Web Token

> JSON Web Token

- RFC 7519 표준으로 JSON 포맷 기반으로 사용자에게 대한 정보를 저장하는 Claim 기반의 토큰
- Claim 기반
 - 토큰에 대상의 정보를 담은 방식
 - 장점
 - 토큰 자체에 데이터를 저장하기 때문에 별도의 저장과정의 필요가 없음
 - 자체적인 만료시간 구현 가능
 - 단점
 - 토큰 자체의 제어 불가능(취소 불가능)
 - 토큰에 정보를 담고 있어 탈취당하였을 경우 정보가 공개됨
- AWS의 API Gateway등의 서비스에서 지원

› JSON Web Token

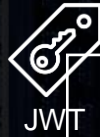
› JSON Web Token

- JWT의 구성
 - Header : 토큰의 타입과 알고리즘 종류
 - Payload : 실제 데이터가 저장되는 공간
 - iss: 토큰 발급자(Issuer)
 - sub: 토큰 제목(Subject)
 - exp: 토큰 만료시간(Expiration)
 - iat: 토큰의 발급 시간(Issued At)
 - Signature
 - Header와 Payload의 정보를 검증하기 위한 고유한 암호화 스트링

> JWT

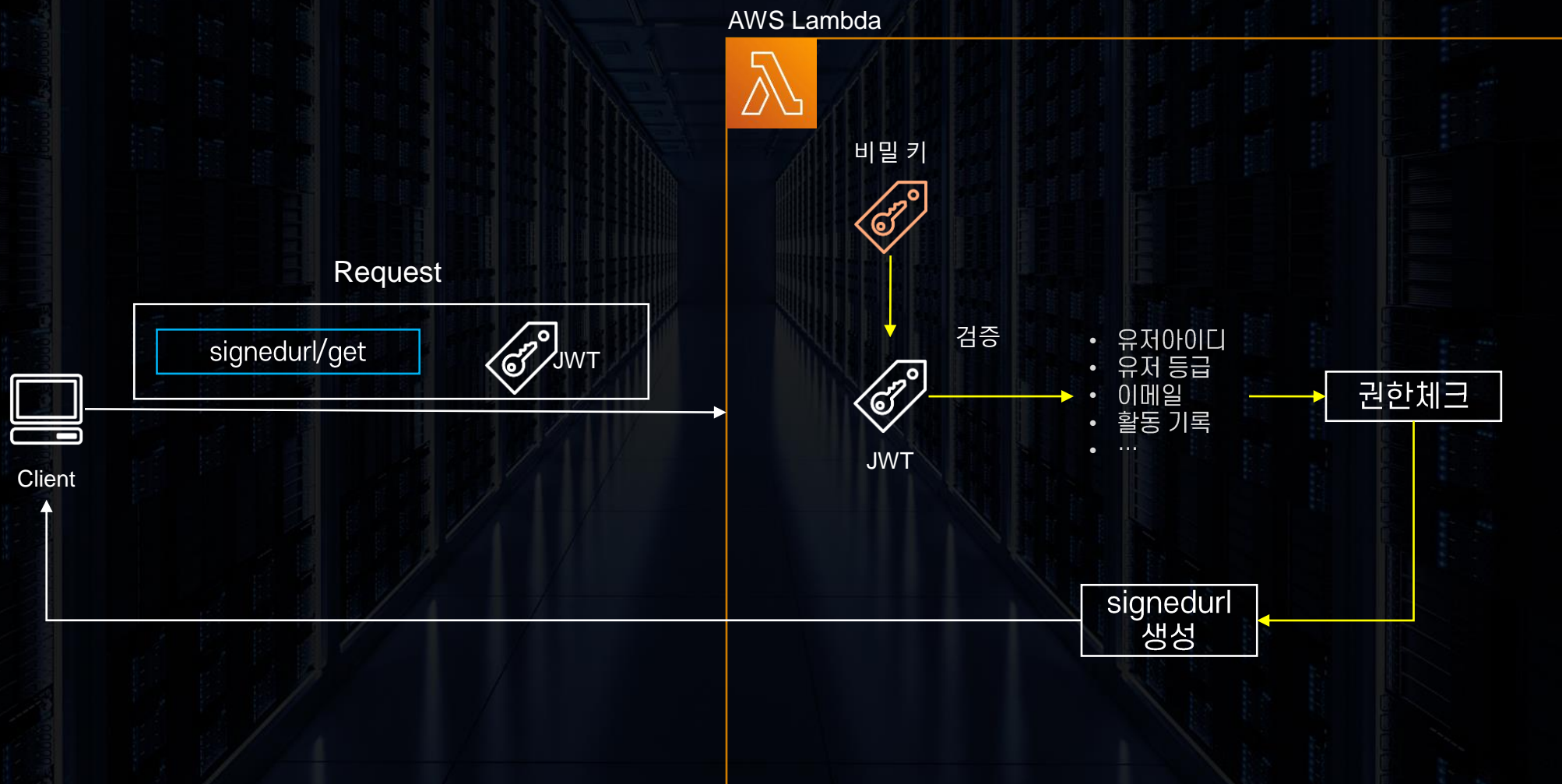


> JWT



```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}  
.  
{  
  "sub": "1234567890",  
  "email": "spark@rubywave.io",  
  "nickname": "spark",  
  "rank": "15",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```


> Elastic Load Balancer



› Demo – JWT Token

› Demo – JWT Token

- JWT Token을 생성하는 Lambda 함수 생성
- JWT Token 생성
- JWT Token 정보를 확인하는 Lambda 함수로 JWT 토큰 확인



rubywave.io

spark@rubywave.io