

AWS Lambda 를 활용한 웹 기반 Serverless 개발환경 구축 사례

클라우드퀘스트 주식회사
이종훈 대표 / alexlee@cloudquest.net

클라우드퀘스트 주식회사

2013 설립

2022 법인 전환

클라우드 컨설팅 및 아키텍처 수립

projectoom.io 서비스 개발 운영

대표자 소개

이종훈 대표

1999 LG CNS / ERP / 아키텍처 그룹

2008 삼성 SDS / OSP

2011 Amadeus Singapore

2013 클라우드퀘스트 설립

2022 **projectroom.io** 런칭

LG계열사, 삼성전자, 대한항공, 건강보험공단
의료급여 프로젝트, 삼성 Account, AWS and GCP

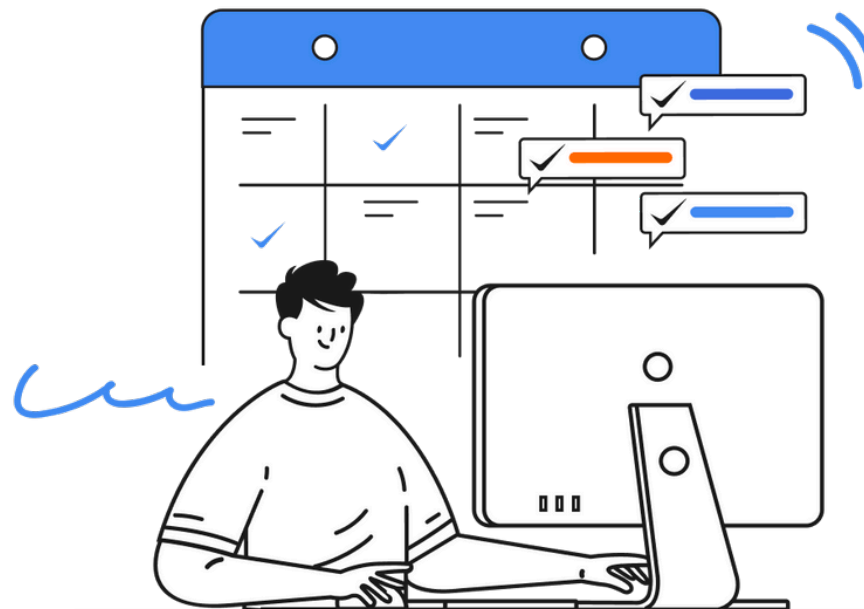
“쉽고 빠르게 웹 시스템을 구현할 수 있는 Serverless 클라우드 기반의 통합 웹 개발환경”

다양한 언어를 지원하는 개발 프레임워크

Lambda, S3 기반의 개발 및 테스트 환경

Git / Git Flow 를 통한 협업 지원

개발 지원 및 컨설팅



웹 개발에 필요한 기술 요소를 Serverless Resource 의 대응

Server 기반

ESB on EC2
S3, EFS

ECS, EKS,
EC2

RDS, Redis,
DynamoDB

ALB

기술 요소

Frontend
(html, js, css, image)

Backend
코드 실행 엔진

데이터 저장소
(DB, Cache, 3rd party Service)

통신 프로토콜
(JSON over http, WebSocket)

Serverless

S3, EFS

ECS, EKS,
Lambda

RDS, Redis,
DynamoDB

API Gateway
ALB

Lambda 중심의 플랫폼 구성

Pros.

다양한 프로그래밍 언어 런타임

저렴한 운영 유지 비용

쉬운 시작

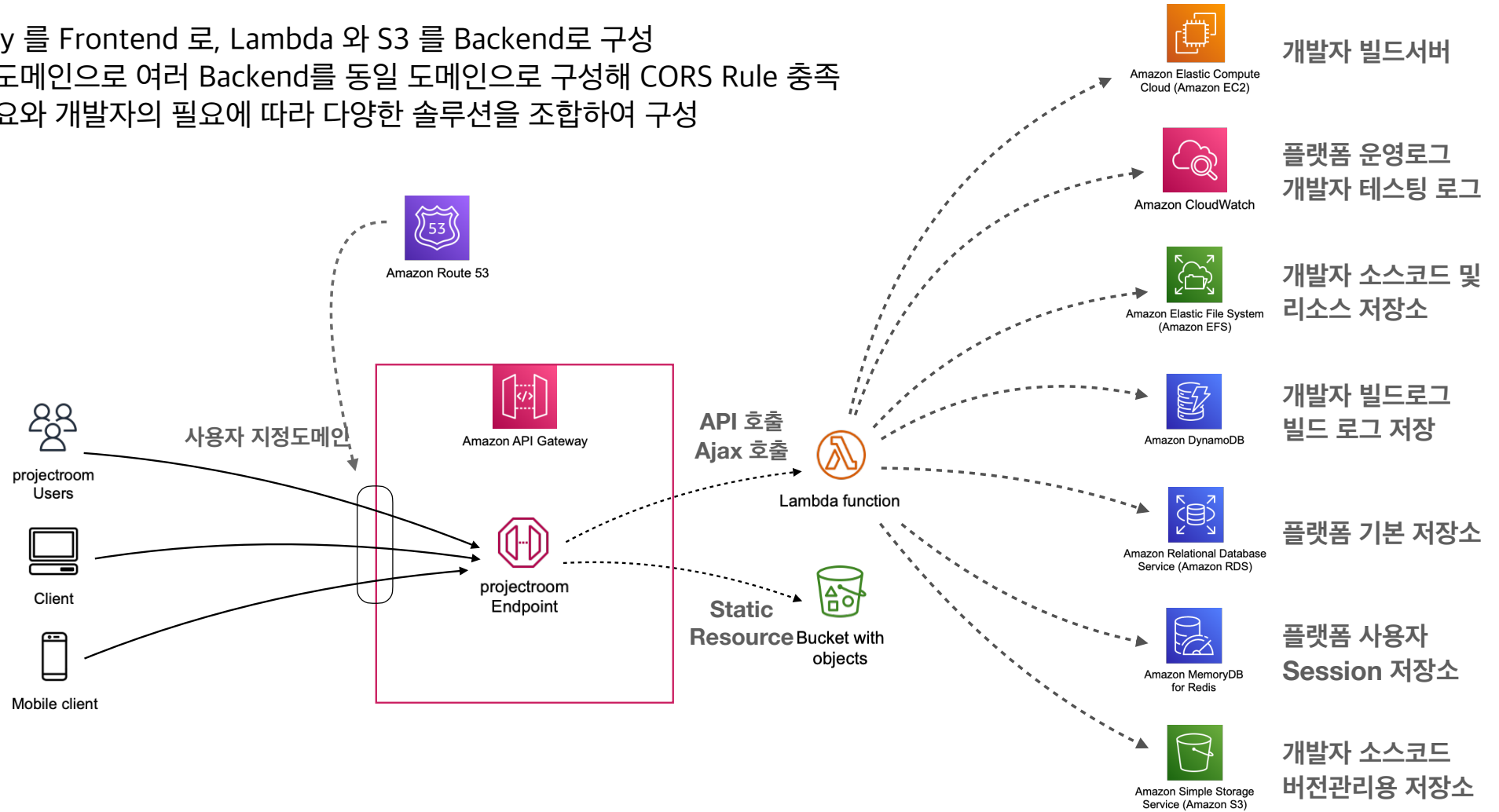
Cons.

단순한 프로그래밍 구조

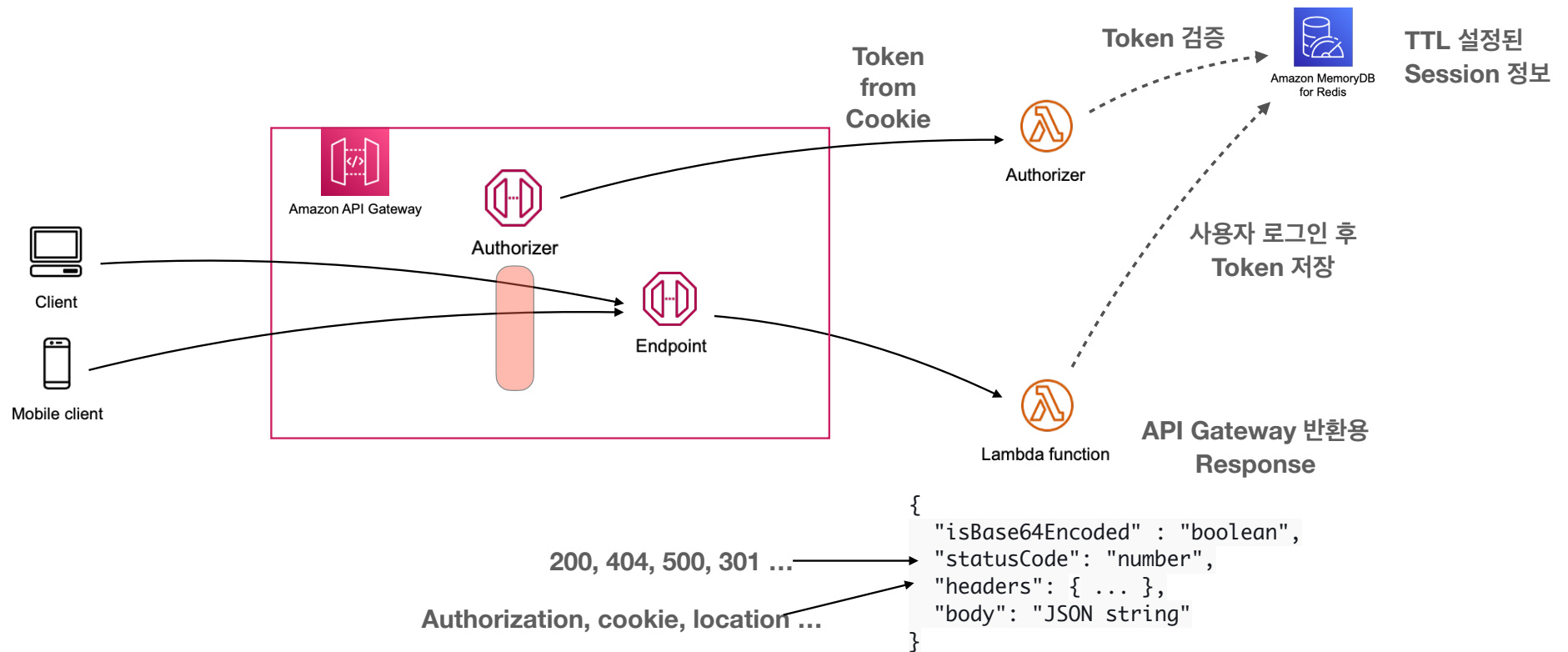
지속성있는 리소스 관리의 어려움

다양한 Serverless 기술이 필요

API Gateway 를 Frontend 로, Lambda 와 S3 를 Backend로 구성
 사용자 지정도메인으로 여러 Backend를 동일 도메인으로 구성해 CORS Rule 충족
 플랫폼의 필요와 개발자의 필요에 따라 다양한 솔루션을 조합하여 구성



REST API : Lambda 로 연결된 JSON in/out 의 API 구현
Authorizer : 사용자 접근제어 구현



API Gateway : REST API 프록시 리소스 구성

URL 이 고정되지 않고 특정 리소스 아래 주소가 계속 변하는 모든 경우를 수용하는 패턴

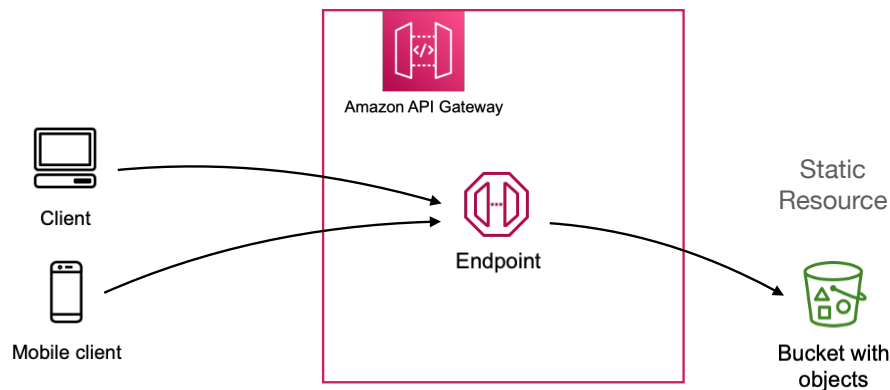
S3에 있는 static 파일의 위치 : [S3버킷이름]/static/global/css/all.css

[S3버킷이름]/static/main/js/custom.js

[S3버킷이름]/static/main/image/logo.png

모든 static 리소스의 최상위 루트

루트 아래에서 파일이 업로드 되는대로 만들어지는 다양한 하위 Path



새 하위 리소스

이 페이지를 사용하여 리소스에 대한 새 하위 리소스를 생성합니다.

☒ 프록시 리소스로 구성

☒ ⓘ

리소스 이름*

proxy

리소스 경로*

/static/ {proxy+}

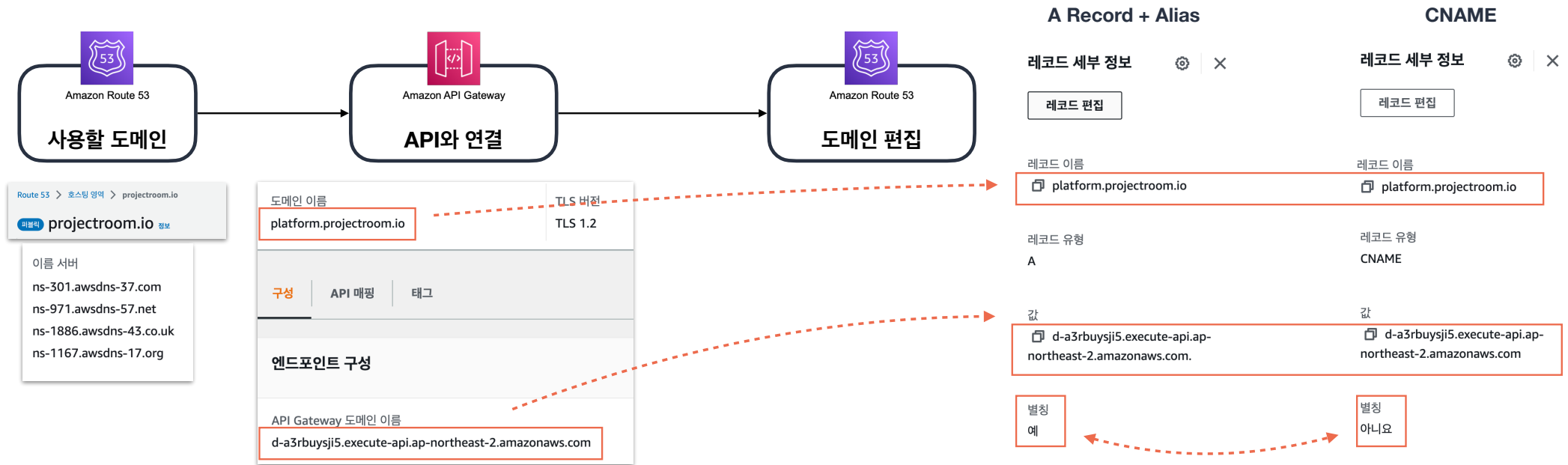
괄호를 사용하여 경로 파라미터를 추가
소스로 구성하면 그 하위 리소스에 대
스에 새 ANY 메서드를 추가합니다.

API Gateway : 사용자 지정 도메인 (1)

배포된 API 를 대상으로 사용자 지정도메인을 지정해 사용
와일드카드 도메인 적용 가능

AWS Certificate Manager 의 SSL 인증서 발급 및 적용 가능 (무료)

Route53 등록 도메인 (또는 호스팅 영역 등록 도메인) + 동일계정의 API : A Record 별칭 또는 CNAME 연결가능
타사 등록 도메인 또는 타계정 API : CNAME 연결가능



API Gateway : 사용자 지정 도메인 (2)

API Gateway 의 사용자 지정 도메인은 여러 API 매핑이 가능

도메인 이름
platform.projectroom.io

TLS 버전
TLS 1.2

상태
🟢 가용

구성

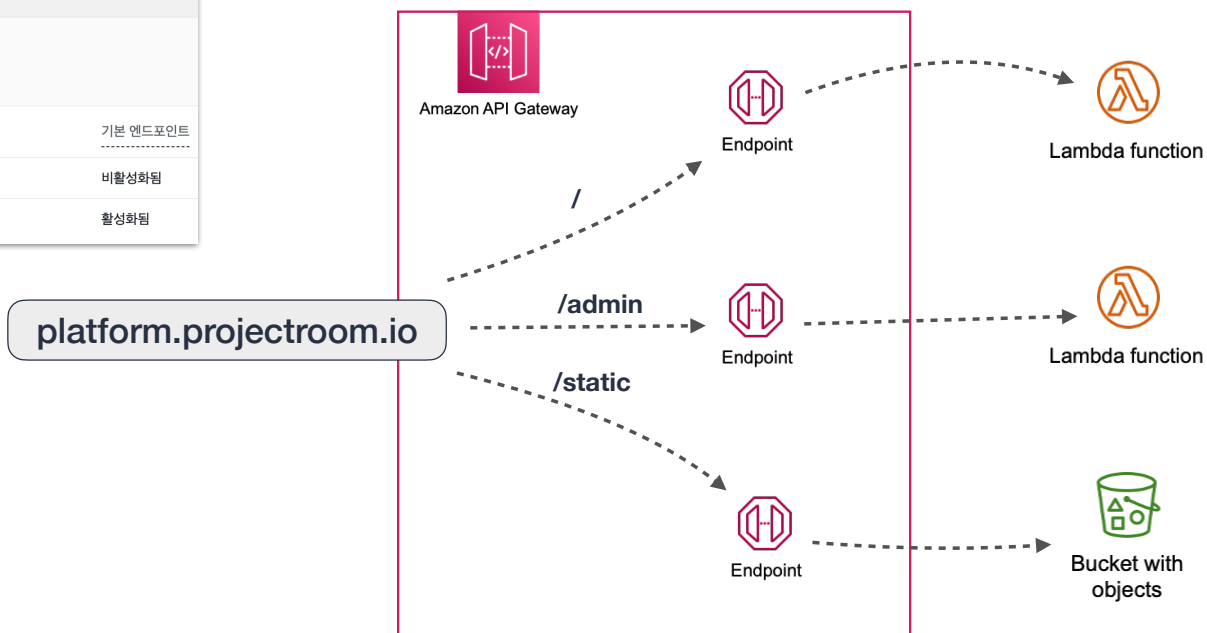
API 매핑

태그

API 매핑

도메인 이름에서 API 스테이지로 경로 매핑

API	스테이지	경로	기본 엔드포인트
projectroom-serverless-platform	prd	(none)	비활성화됨
projectroom-serverless-admin	dev	admin	활성화됨



API Gateway : WebSocket

WebSocket API 를 사용하여 Lambda 의 비동기 통신의 결과 회신에 사용

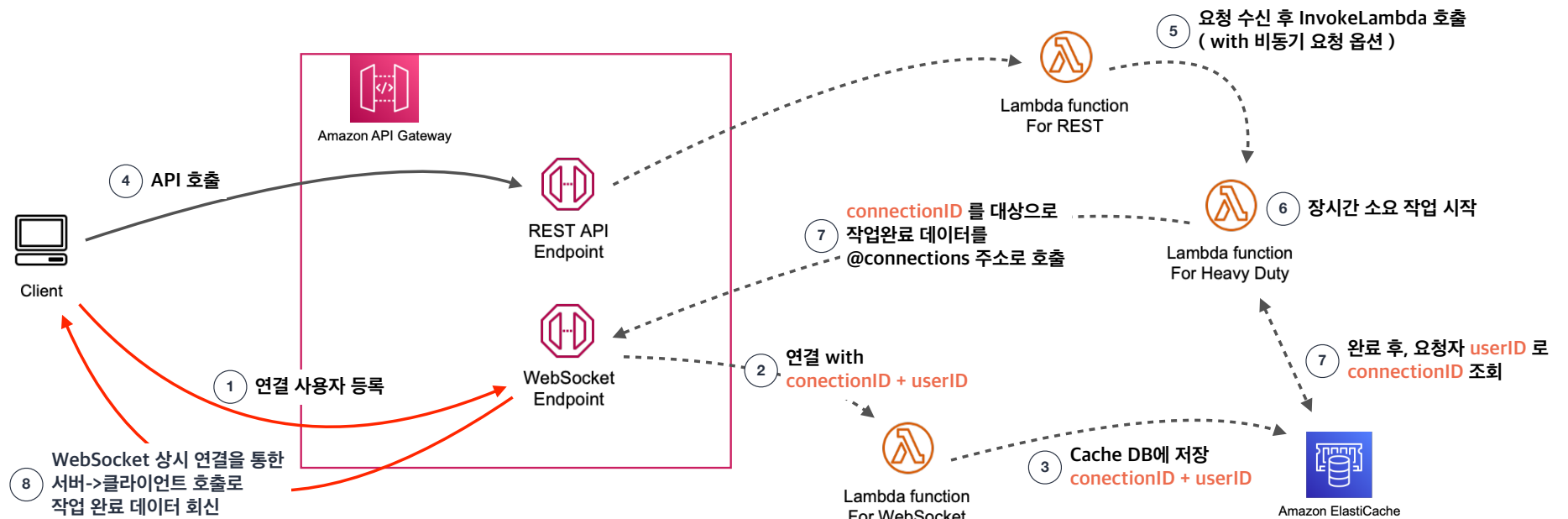
- API Gateway 의 최대 응답 대기 시간 : 30초 (연장 불가)
- 긴 작업시간이 필요한 모든 요청에서, “작업 수행 완료” 를 화면에 리포트 하는 일이 필요

WebSocket URL: `wss://kmwrb6odp4.execute-api.ap-northeast-2.amazonaws.com/prd`

연결 URL: `https://kmwrb6odp4.execute-api.ap-northeast-2.amazonaws.com/prd/@connections`

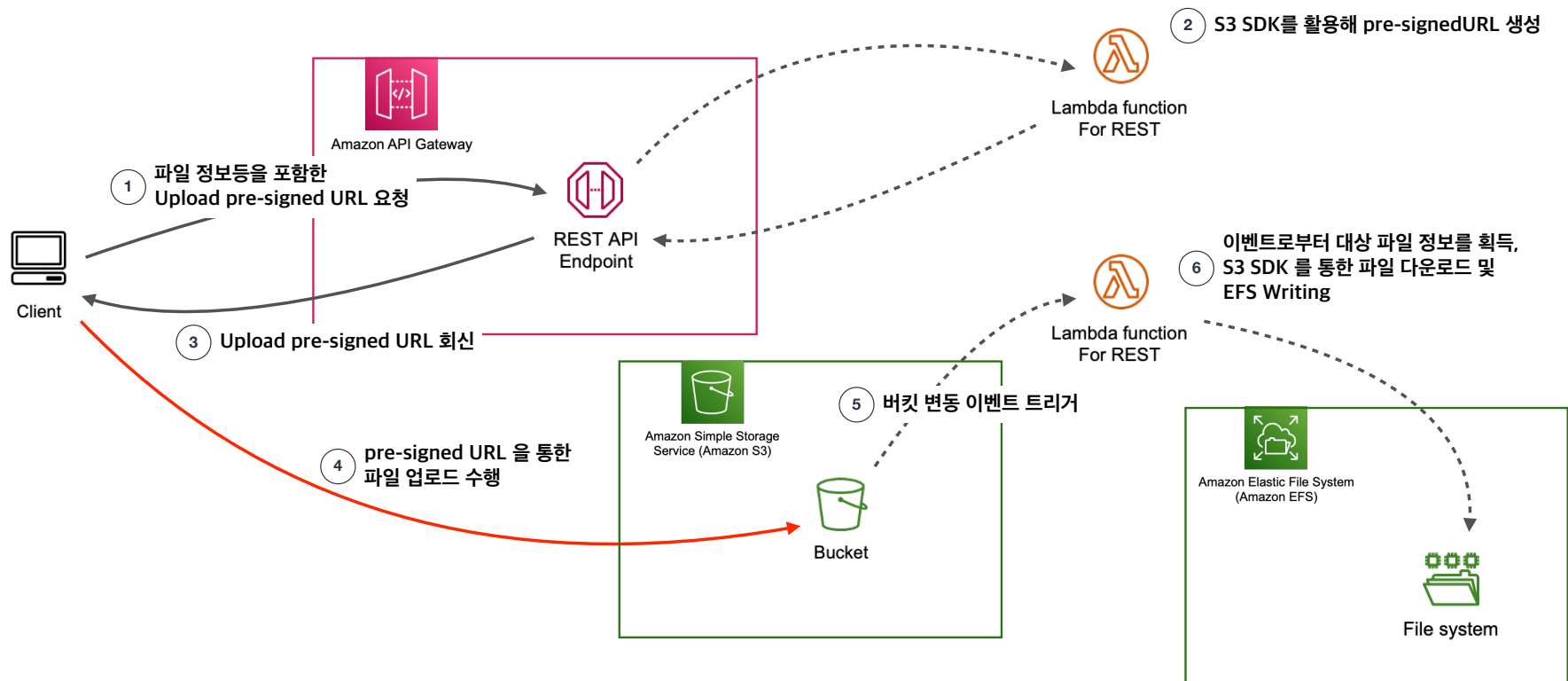
WebSocket 연결을 위해 화면에서 API Gateway 를 호출하는 주소 (화면 -> 서버 메시징은 이 연결을 통해 가능)

WebSocket 응답을 위해 서버측에서 호출하는 주소 (Signed URL 호출 필요)



pre-signed URL을 사용한 파일 업로드 다운로드

- Multipart -> Lambda -> S3 방식은 Serverless 에 적합하지 않음
- 업로드 / 다운로드시 pre-signed URL 을 발급하여 Lambda 를 경유하지 않고 파일을 직접 업로드 다운로드
- S3 업로드시 버킷을 대상으로 이벤트 트리거 발생 => EFS파일 배치 연계 가능



No more “tail -f service.log”

Serverless 들어가서 로그를 확인할 Server 가 없음...

Use “aws logs tail /aws/lambda/your-lambda-function --follow --profile user”

- Lambda 실행 런타임이 자동으로 CloudWatch 로그 그룹을 형성
- 서버 콘솔을 대체하는 기능이 가능
- 다중화 서버로부터 로그 취합이 불필요
- AWS API 를 통해 관리 시스템 UI 로 통합 가능
- Lambda 의 과금 정보가 함께 표시되는 장점
- 반응 속도가 느린 것은 단점

CloudWatch 로그 그룹 이름

AWS user profile

1. Command line tool (feat. Git)

- Git (w/ ssh) binaries for AWS Lambda (<https://github.com/lambci/git-lambda-layer>)
- Git 커맨드를 Lambda 에서 사용 가능
- Lambda 구현 언어의 외부 Command 프로세스 구동 명령어를 이용
(Node.js executeSync(), Java 의 Process 클래스 등)

Layer selection
Select an existing AWS-vended layer or layer in your account, or provide a layer that has been shared with you.

☐ Select from list of runtime compatible layers
☒ Provide a layer version ARN

Provide a layer version ARN

Layer version ARN [Info](#)
Provide the ARN of a layer to add to your function.

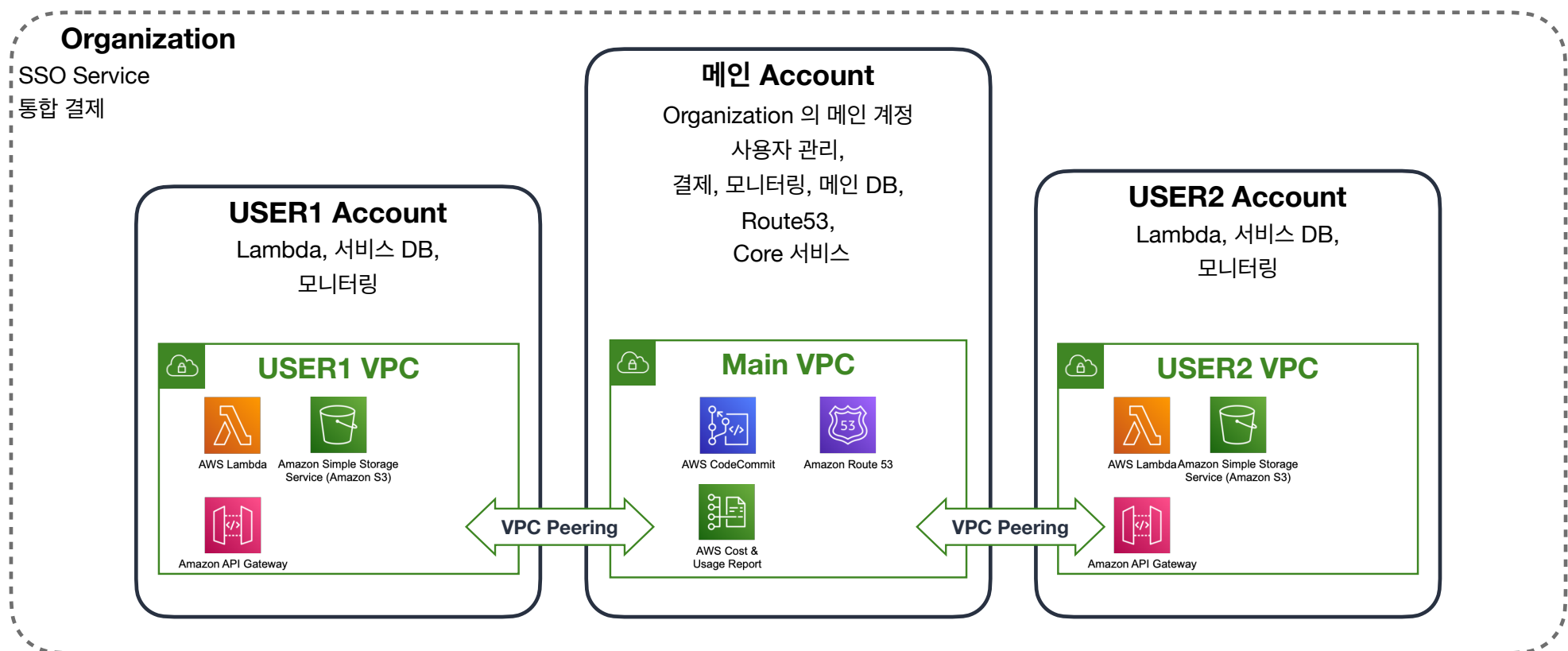
2. External Libraries

- Lambda 계층 만들기 및 공유(https://docs.aws.amazon.com/ko_kr/lambda/latest/dg/configuration-layers.html)
- 배포 시간과 본 함수 패키지 크기(함수, Layer 각각 250M 제한)를 줄일 수 있음
- 최대 5개 까지 추가 가능

각 Lambda 런타임에 대한 계층 경로	
런타임	경로
Node.js	nodejs/node_modules
	nodejs/node14/node_modules (NODE_PATH)
Python	python
	python/lib/python3.9/site-packages (사이트 디렉터리)
Java	java/lib (CLASSPATH)
Ruby	ruby/gems/2.7.0 (GEM_PATH)
	ruby/lib (RUBYLIB)
모든 런타임	bin (PATH)
	lib (LD_LIBRARY_PATH)

API Gateway, Lambda 를 비롯한 AWS 서비스 자원의 Quota 가 존재

- “User 별 Region 당 000” 방식으로 카운트
- Quota 를 늘리는 방법 ==> **하나의 Organization 아래 여러 User 를 운용하는 것**



서비스 요건에 유연하게 대응하는 아키텍처 : 빌드서버 구축의 사례

빌드 서버 요건	EC2 + Auto Scaling Group	ECS Fargate Lambda with Machine Image	Code Build
반응성 : 빠른 시작	GOOD Spring Boot Servlet	BAD Provisioning or Cold/Warm Start	BAD Provisioning
빌드 라이브러리 캐시 저장소 : 반복 수행시 빌드 속도 향상	GOOD EBS on EC2	BAD Temp Storage EFS	BAD Temp Storage
추가적인 응용구성 DB / Redis / WebSocket / CloudWatch	GOOD Based on Application F/W	GOOD Based on Application F/W	BAD CLI is not enough
가용성 / 확장성	GOOD EC2 Auto Scaling Group	GOOD Auto Scaling	GOOD Auto provisioning by Request
운영 비용	BAD 상시 유지 필요	GOOD 필요할 때 만들고, 쓴 만큼 과금	GOOD 필요할 때 만들고, 쓴 만큼 과금

QnA