

#Serverless

이상현 / Kurt Lee



- CTO, Catchfashion Inc
- Former Technical Leader, Vingle Inc
- AWS Serverless Hero

iOS / Frontend / Backend / Serverless /
Data Pipeline / API / Microservices

breath103@gmail.com

<https://github.com/breath103>

<https://medium.com/@kurtlee>

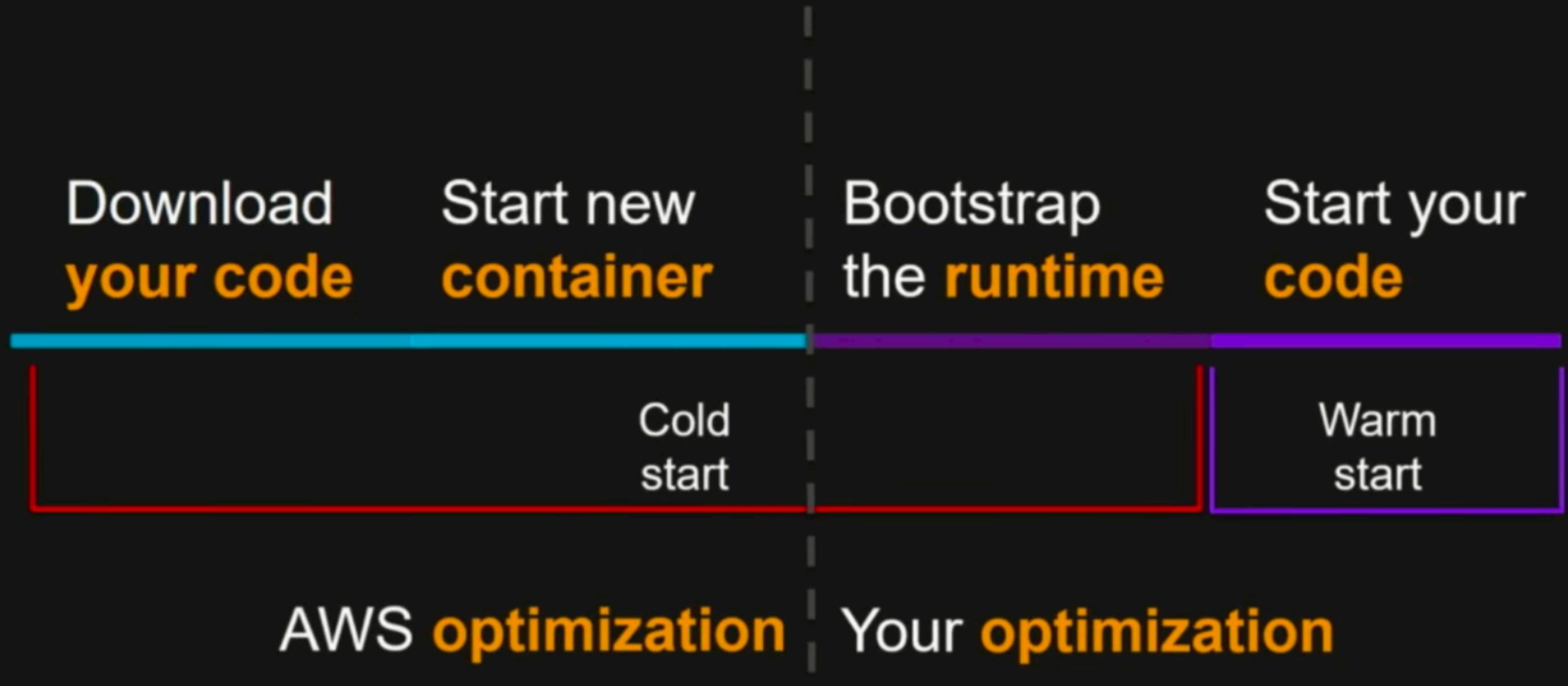
Agenda

1. AWS Lambda 의 컨테이너 재활용과 coldstart
제대로 이해하기
2. AWS Lambda function 하나에 기능을
얼마나 넣는게 좋은지?
3. 각종 AWS Developer Influencer Summit 썰

1. AWS Lambda 의 Coldstart와, 컨테이너 재활용과 제대로 이해하기

우선, Lambda의 작동 방식을
이해해야 합니다.

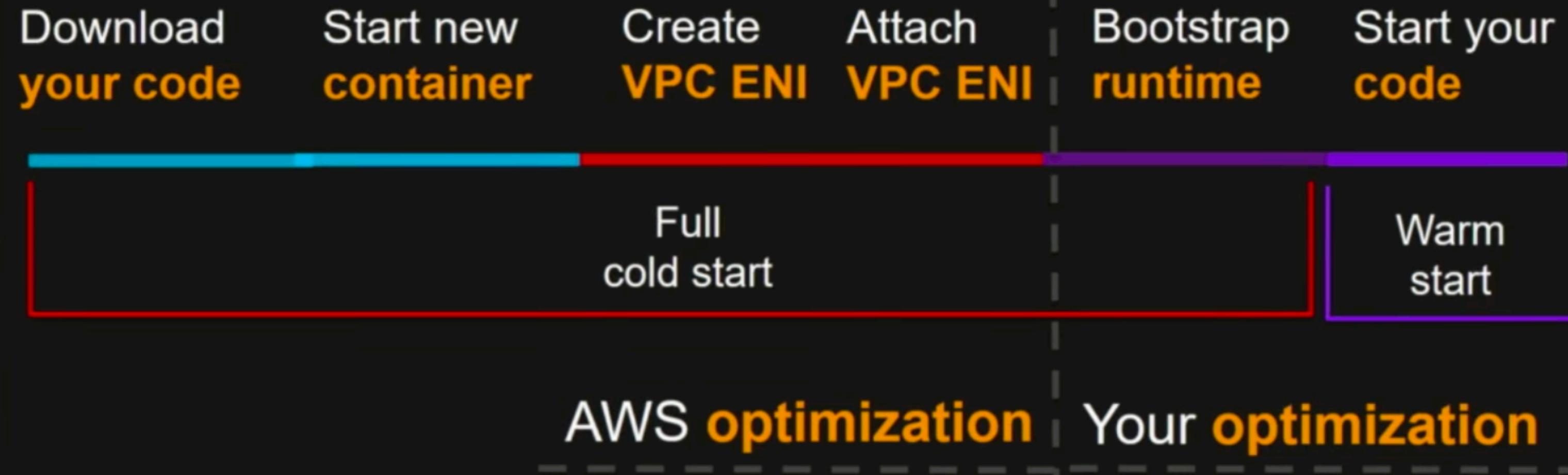
THE REQUEST LIFECYCLE



1. S3에서 Lambda 코드 (Zip) 을 EC2 instance로 다운로드
 - 같은 region에서 S3 다운로드 속도는 사실 엄청 빠르니까…
2. microVM (firecracker)을 시작
 - <https://firecracker-microvm.github.io/>
 - In addition to a minimal device model, Firecracker also accelerates kernel loading and provides a minimal guest kernel configuration. This enables fast startup times. Firecracker initiates user space or application code in less than 125ms and supports microVM creation rates of 150 microVMs per second per host.
3. Runtime bootstrap
 - Layer / runtime 초기화 시간 + 코드 초기화 시간
4. 실제 code 실행
 - 그냥 우리가 코드를 잘 써야해여^^

VPC를 쓴다면
좀더 복잡해집니다.

SECURED: VPC vs LATENCY



SECURED: VPC vs LATENCY

Download
your code Start new
container Create
VPC ENI Attach
VPC ENI Bootstrap
runtime Start your
code

VPC ENI 가 도대체 뭔가요?;;

Full
cold start

Warm
start

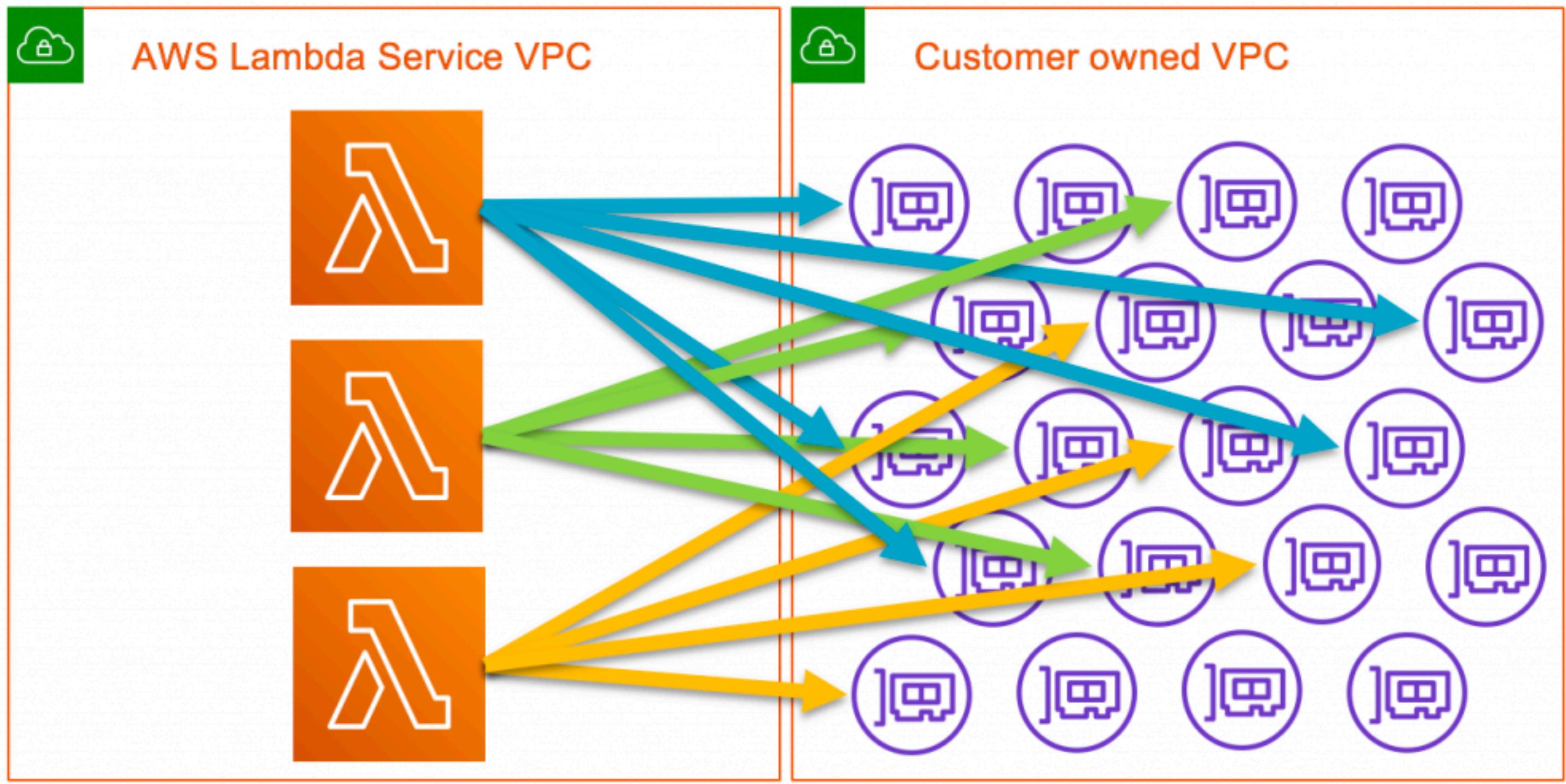
AWS **optimization** | Your **optimization**

간단 요약)

1. Lambda에서 특정 서비스들을 사용하기 위해서 / 보안상 이유로, Private VPC를 접근해야 하는 경우가 있음.
2. Lambda 자체는 AWS Lambda팀 소유의 VPC안에서 뜨도록 되어있어서, 해당 VPC안에서 ENI를 만들어서 Lambda와 연결하면, 해당 ENI는 고객소유의 VPC로 접속이 불가능
3. 그래서 현재 구현 = Lambda는 AWS Lambda팀 소유의 VPC 안에서 뜨고, 거기에 맞춰서 고객소유 VPC안에 ENI를 생성하고, Lambda container들은 필요에 따라 현재 놀고 있는 ENI를 찾아 연결하여 network call을 보냄
4. 근데 ENI를 부팅하는게 시간이 걸리는 데다가, (5분~15분 정도) 모든 Lambda들이 ENI를 잘 재활용 하는게 아니라서, 필요 이상으로 ENI를 만들다 보니 coldstart가 운나쁘면 ENI를 부팅할때 까지 시간이 걸리게 됨

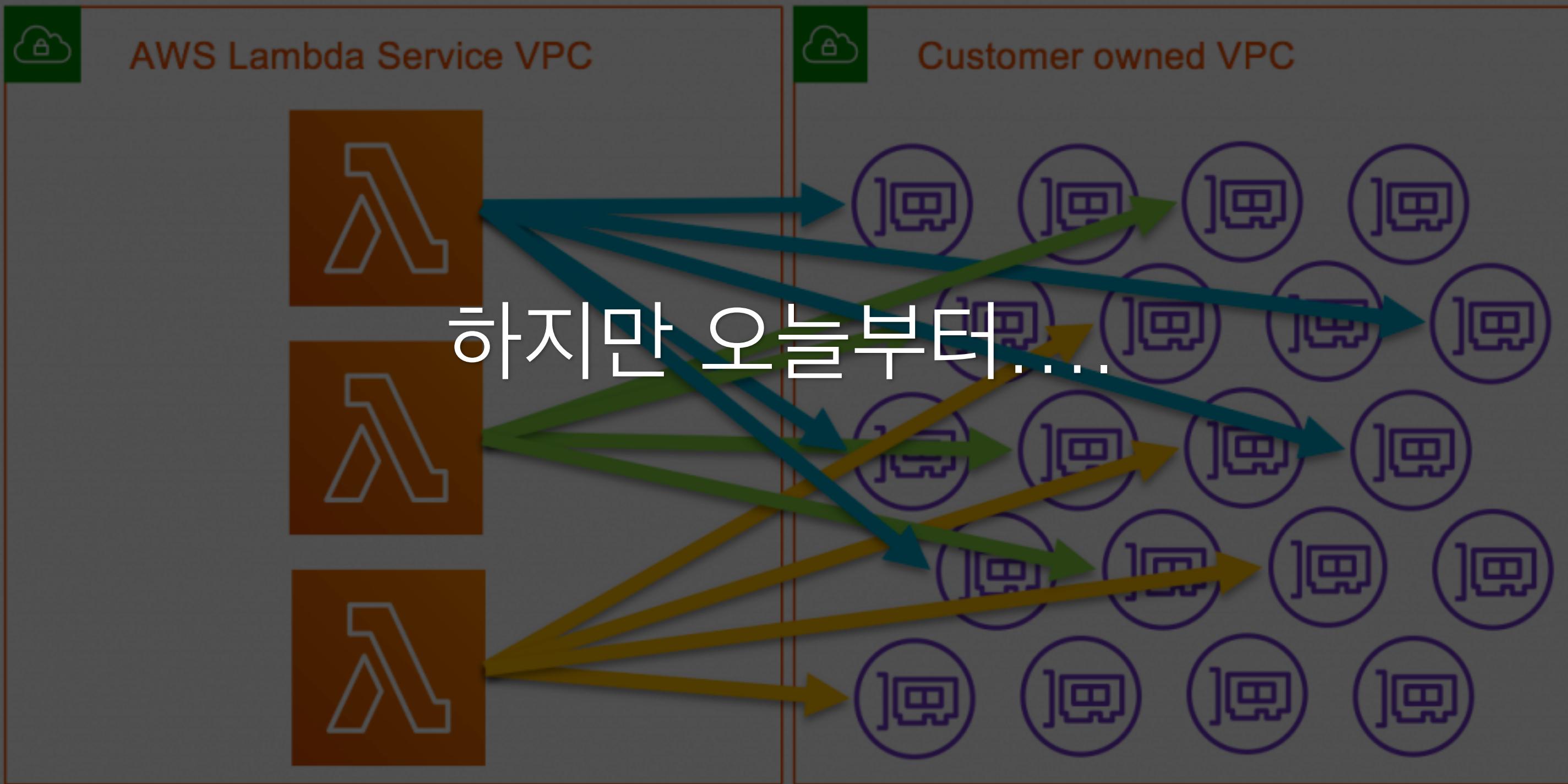


AWS Cloud



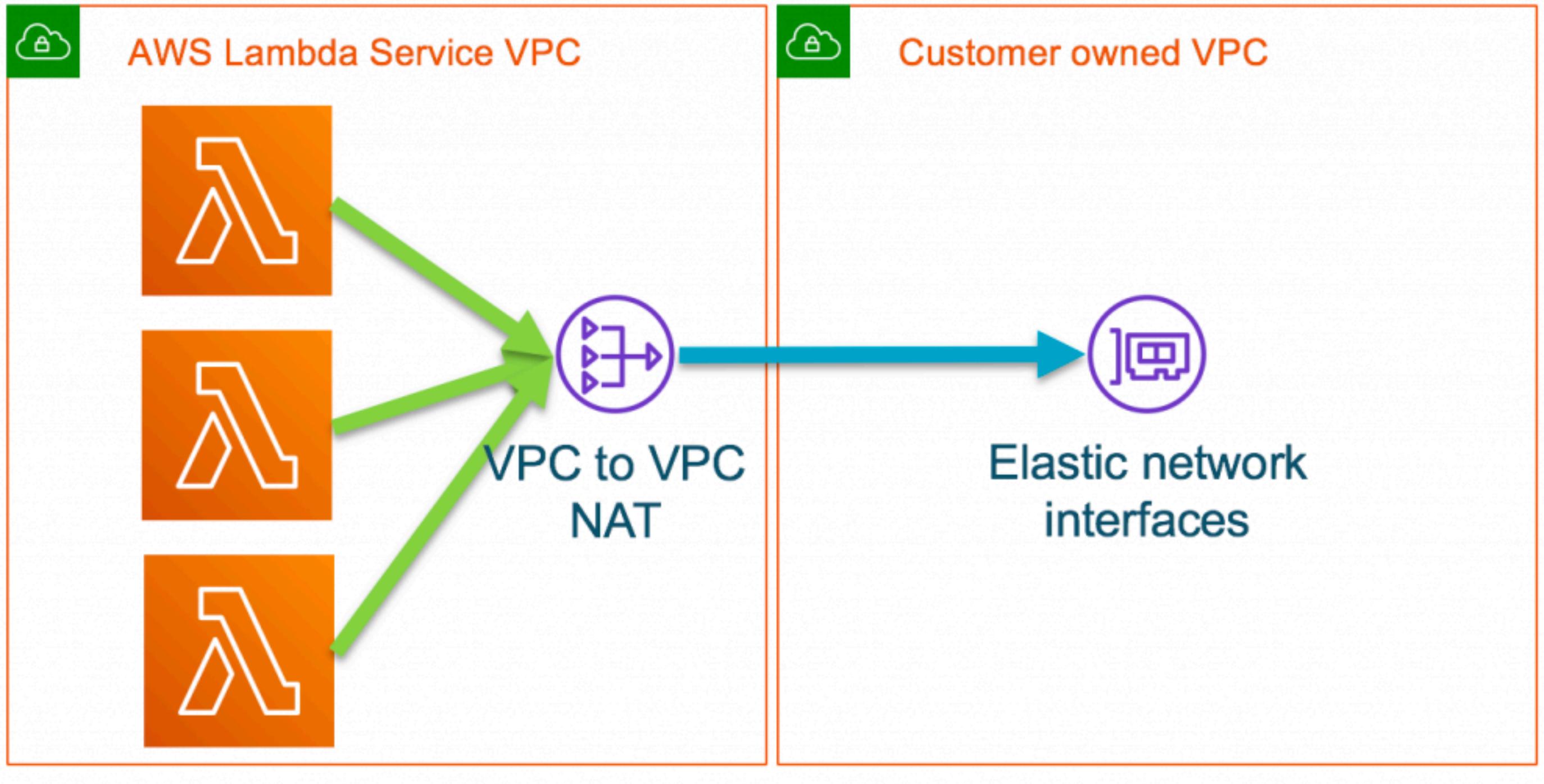


AWS Cloud





AWS Cloud



Q&A

Lambda에서 컨테이너 재활용으로 뽑아먹기

모든 분들이 이해하셔야 하는 사실 한가지,

Lambda != function

Lambda = Code + Memory

따라서, handler 밖에서 선언된 내용들
(handler가 종료되도 메모리 해제되지 않는
내용들) 은, "운이 좋으면" 재활용 됩니다

예를들어…

1. 00:00 에 100번 동시에 호출하고
2. 00:02 에 50번 동시에 호출하면?

1. 00:00 에 100번 동시에 호출하고
2. 00:03 에 300번 동시에 호출하면?

1. 00:00 에 100번 호출하고,
2. 00:01 에 100번 호출하고,
3. 00:05 에 200번 호출하면?

```
let i = 0;
export async function handler(event) {
  i += 100;

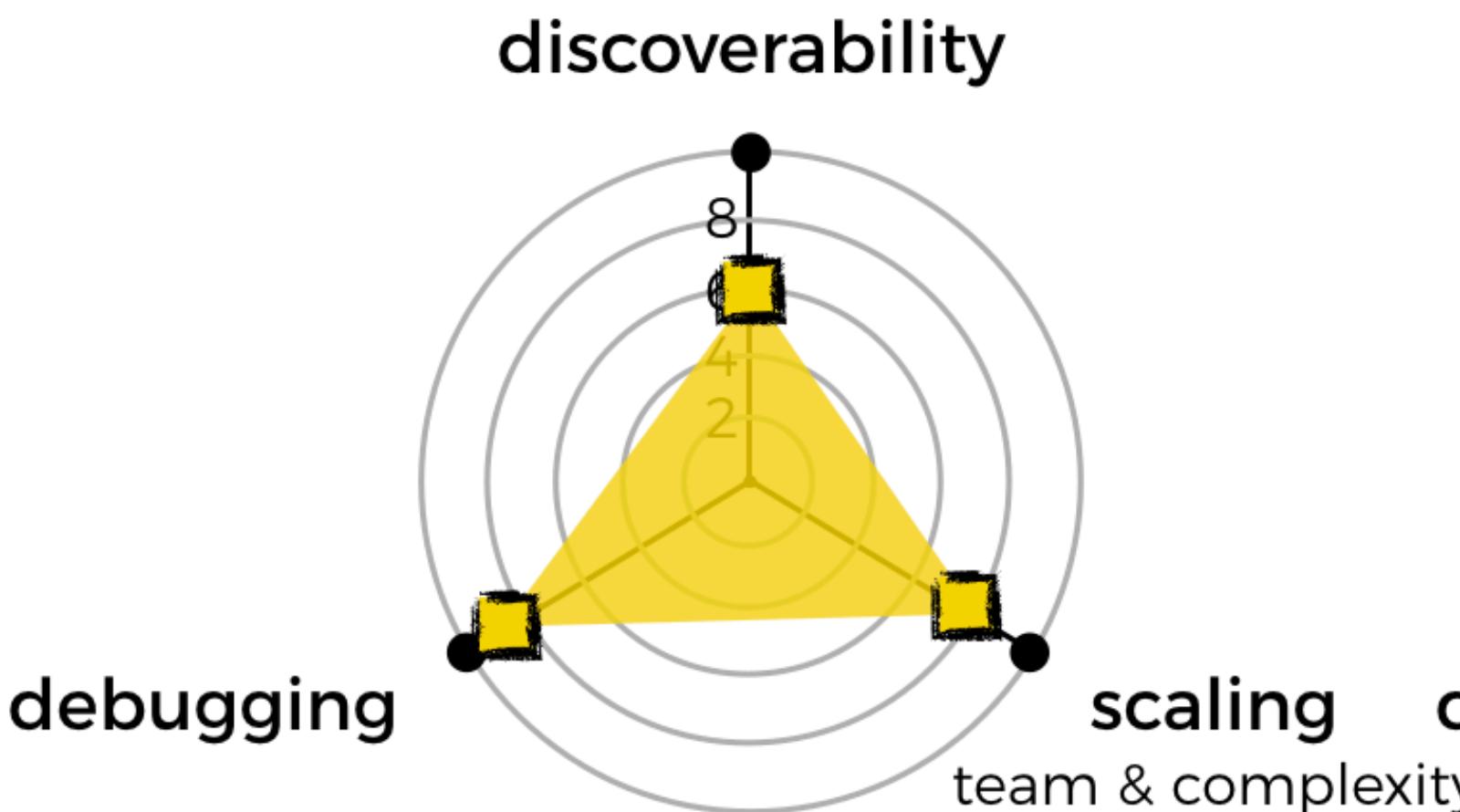
  await new Promise((resolve, reject) => {
    setTimeout(resolve, 1500);
  });

  return i;
}
```

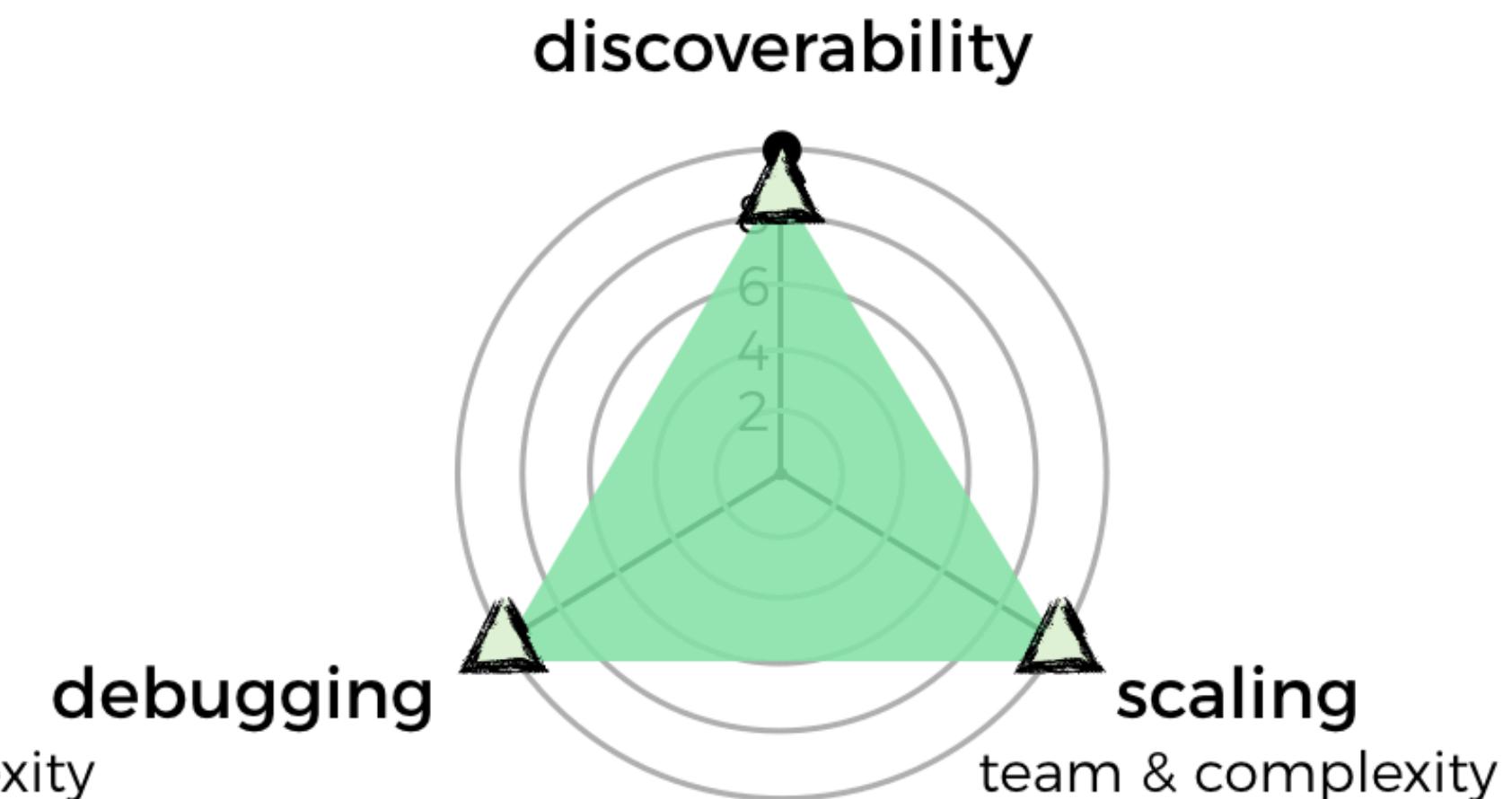
따라서, 접속량이 일정하다 / 적다,
in memory cache처럼 사용하실수 있습니다.

2. Lambda로 Microservice 구성할때,
Lambda 각각의 적정한 구현 범위는?

monolithic



single-purposed



3. 각종 AWS Developer Influencer Summit 썰 (외부로 나가면 안되는)

DAY 1

IN SUCH A BIG WAY.





















Q&A

{CATCH}

Next Generation Luxury Commerce
