Amazon ECS Essentials Tips #1-1

신 정 섭

Solutions Architect

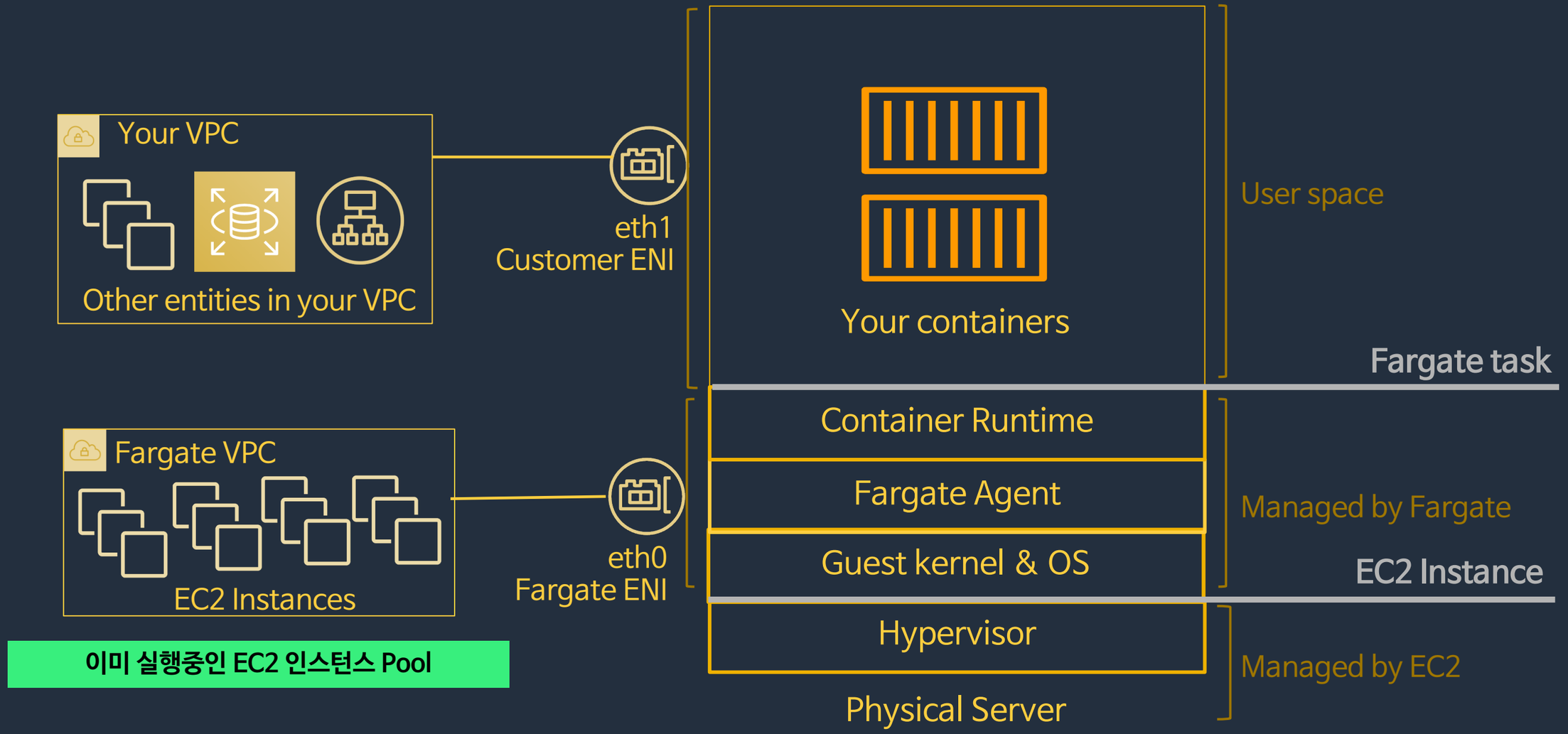
Agenda

- Fargate Dive Deep
- Service Connect
- 카오스 엔지니어링를 위한 AWS FIS
- Amazon ECS 비용 최적화
- 대규모 트래픽 준비하기

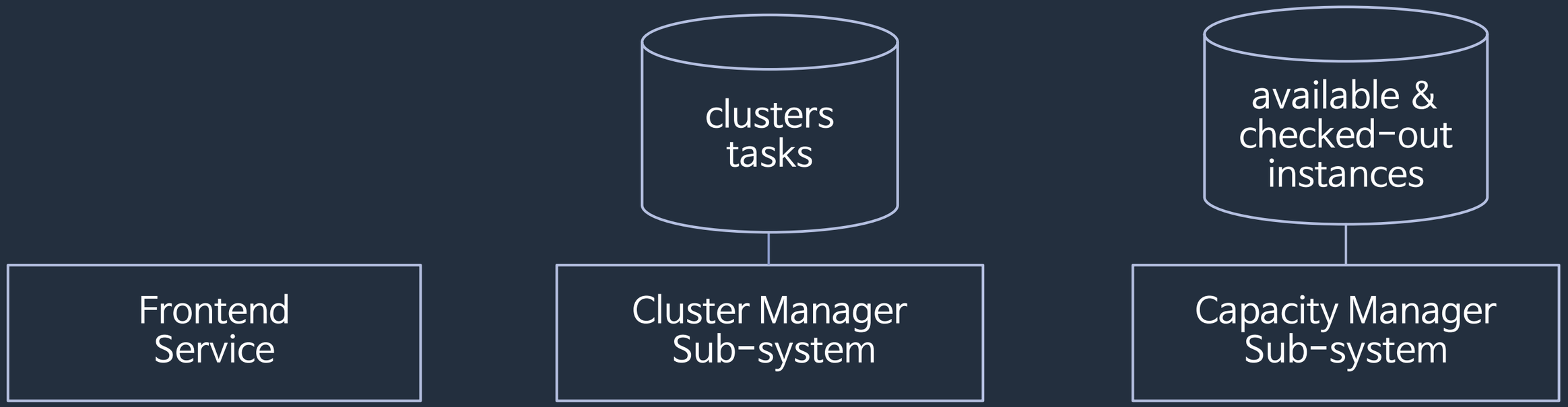
Fargate Dive Deep

Fargate Architecture

Fargate Data Plane



Fargate Control Plane

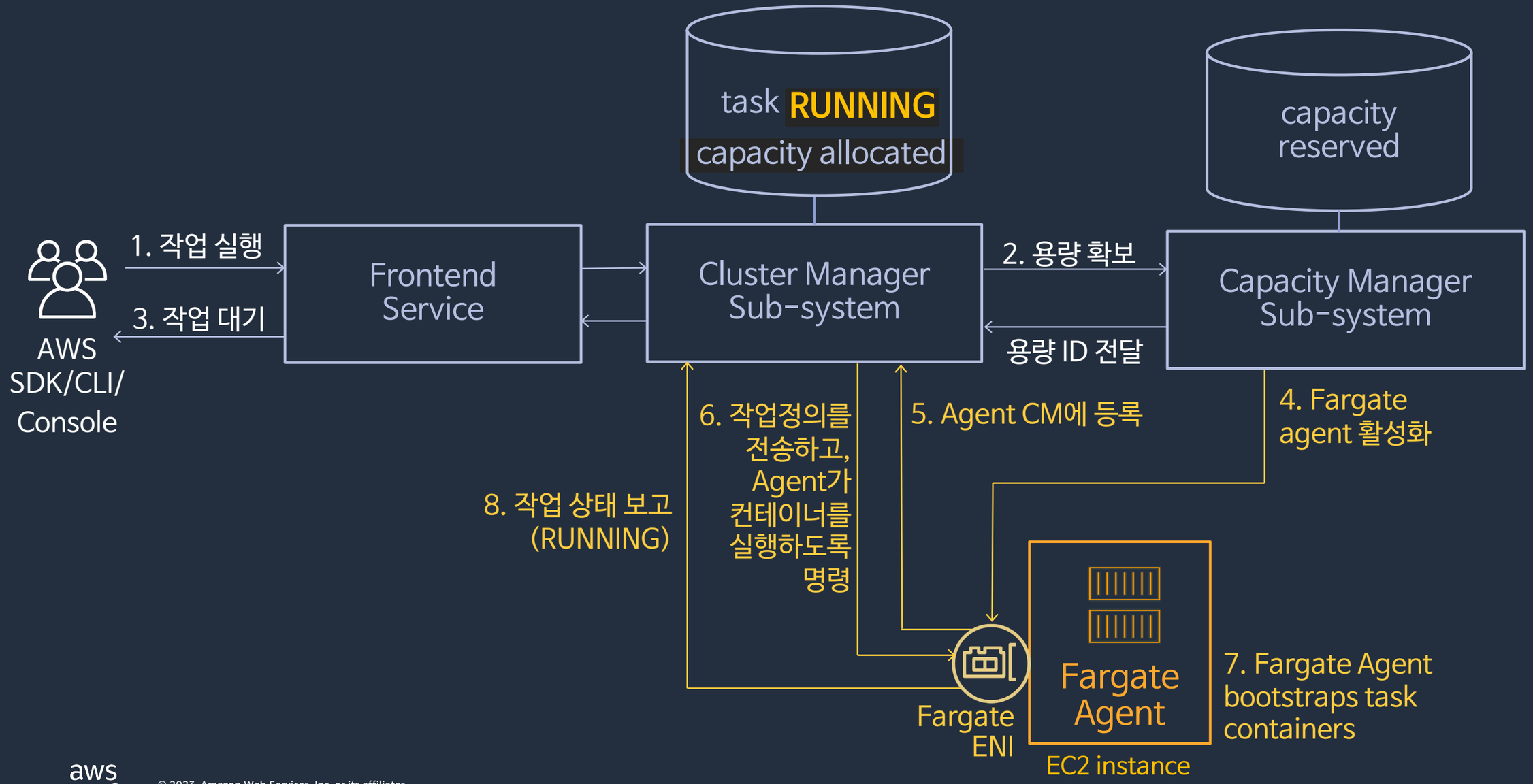


- Entry-point(API 도달점)
- Authentication
- Authorization
- Limit enforcement

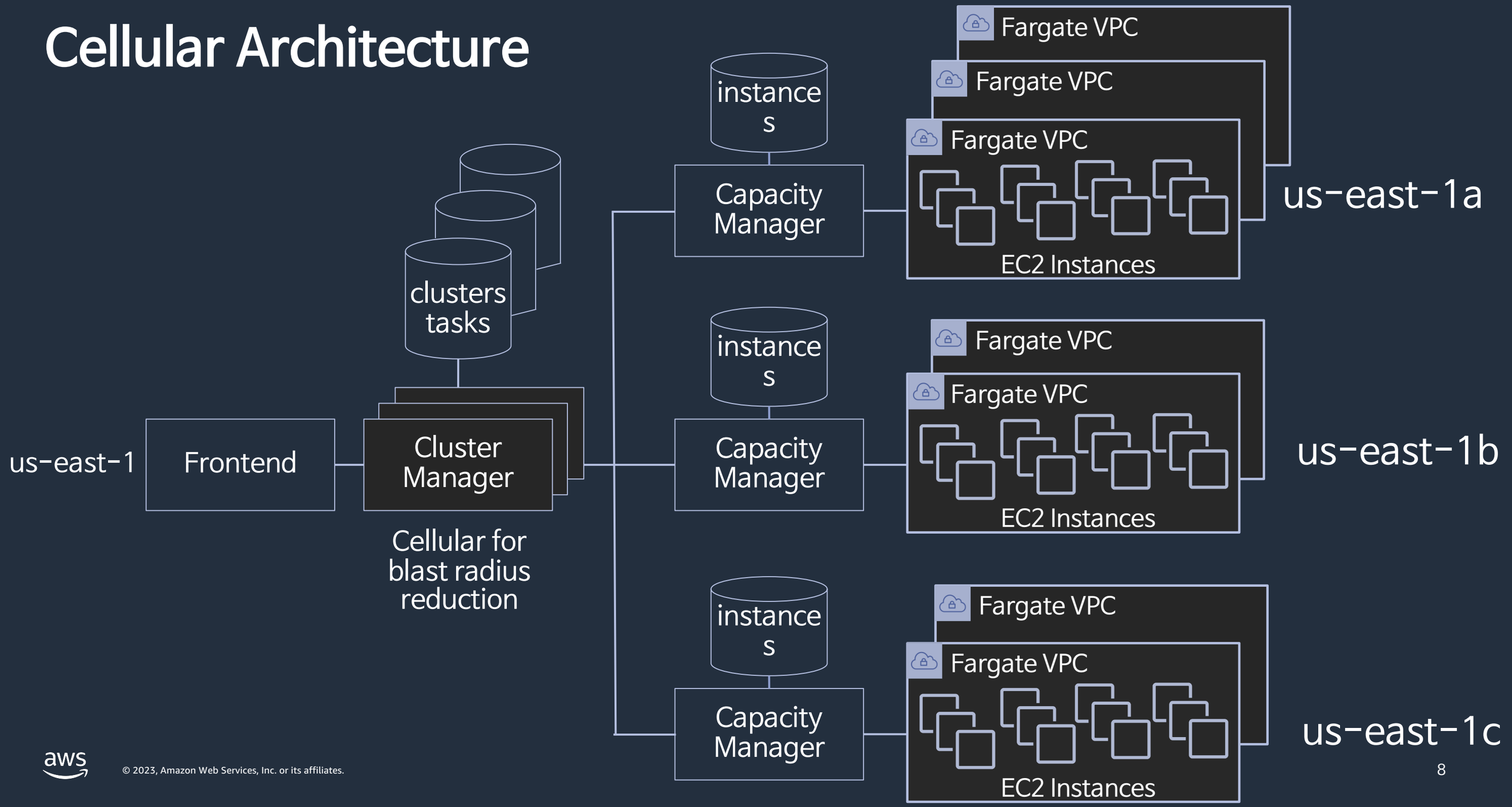
- 클러스터와 작업 상태 관리
- 데이터 플레인과 통신하고 작업 상태 변경을 컨트롤

- 인스턴스에 대한 상태 유지
- 인스턴스에 작업 배치(선택)
- 용량 보충

Task 실행 흐름



Cellular Architecture



Firecracker

Firecracker 는 안전한 멀티테넌트 컨테이너 및 기능 기반 서비스를 생성하고 관리하기 위해 특별히 설계된 오픈 소스 가상화 기술입니다. Lambda, Fargate, Athena 등의 서비스에서 워크로드를 지원합니다.

Security from the ground up
KVM-based virtualization

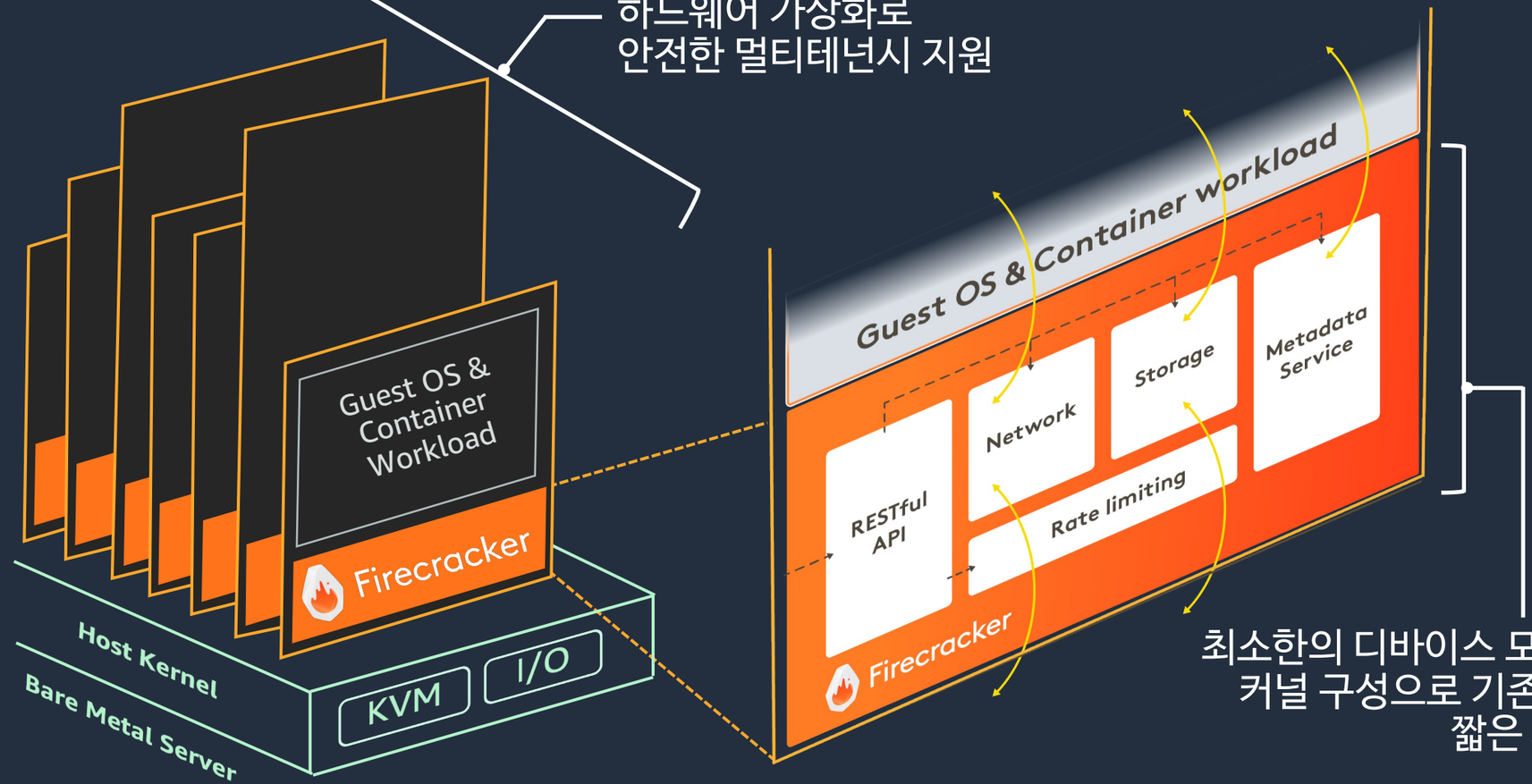
Speed by design
〈 125 ms to launch
150 microVMs per second/host

Scale and efficiency
〈 5 MB memory footprint per microVM

Firecracker

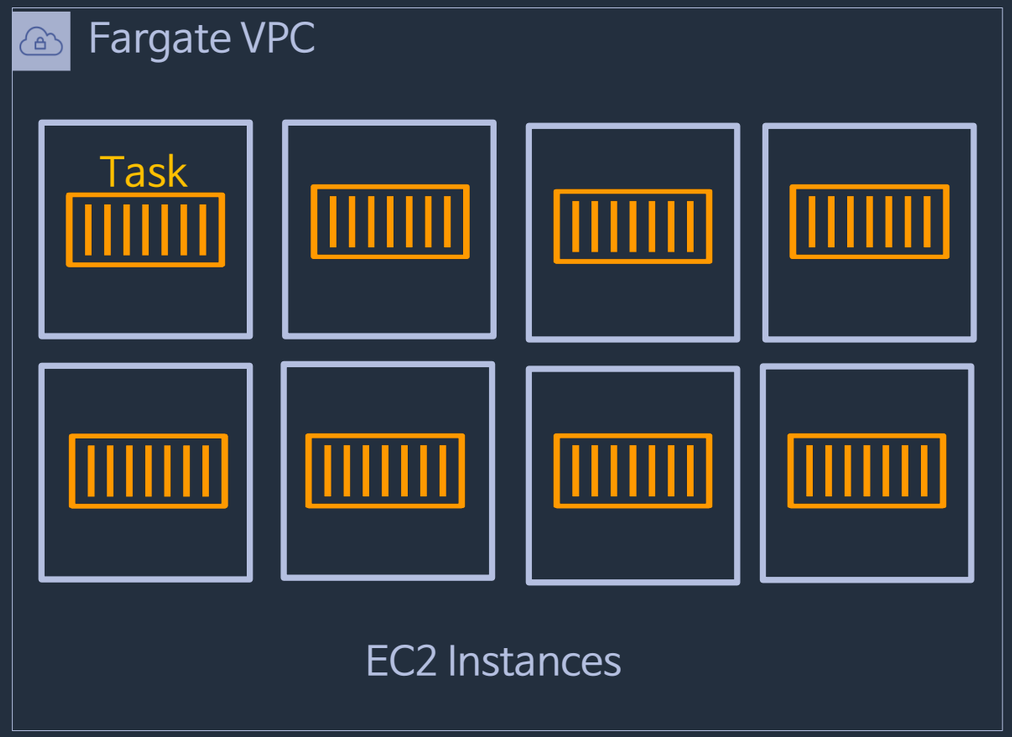
하드웨어 가상화로
안전한 멀티테넌시 지원

사용자 공간
프로세스로
실행되는
CPU/메모리
전반에서
구성 가능한
마이크로
가상 머신

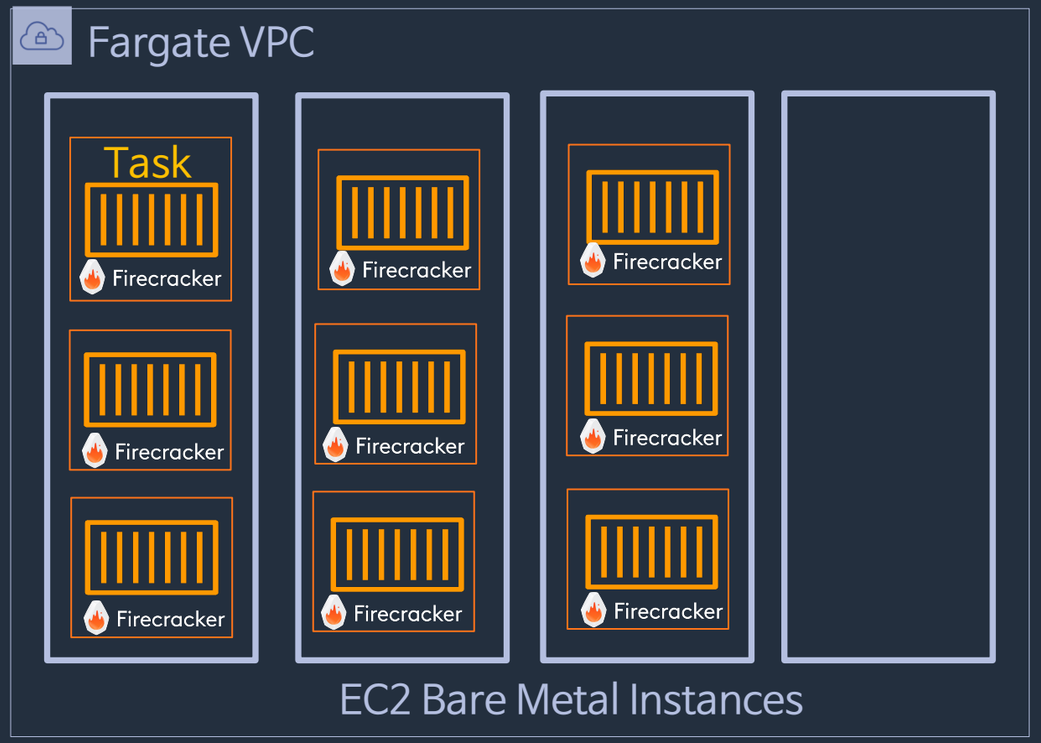


최소한의 디바이스 모델과 최적화된
커널 구성으로 기존 VM보다 훨씬
짧은 시동 시간 제공

How Firecracker helps scalability



VS



Single Tenant EC2 Instances

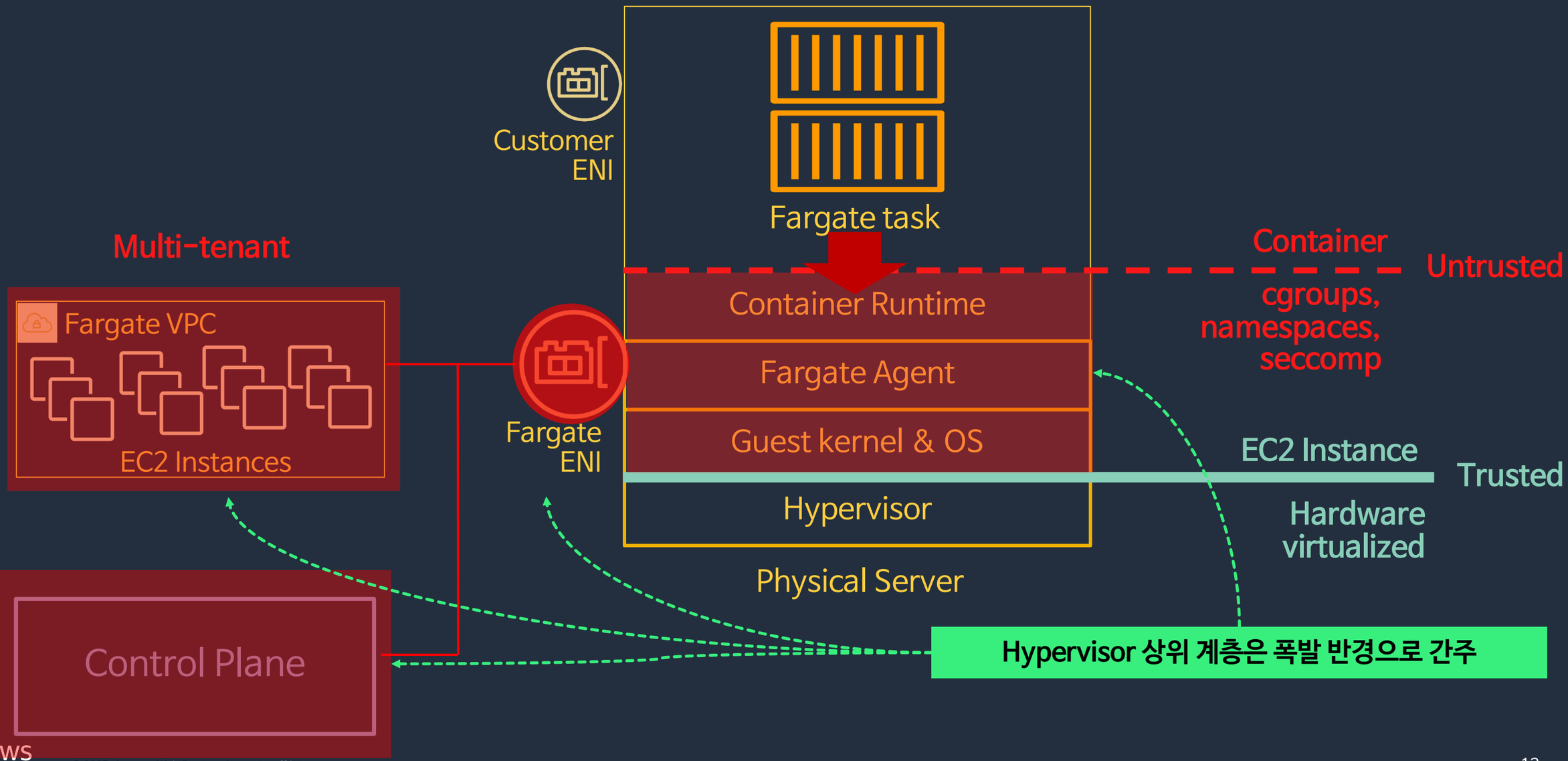
Churn Rate

Tens of millions of Fargate task launches a week → Tens of millions of EC2 instance launches a week (heavy-duty)

Multi-Tenant Bare Metal Instances with Firecracker

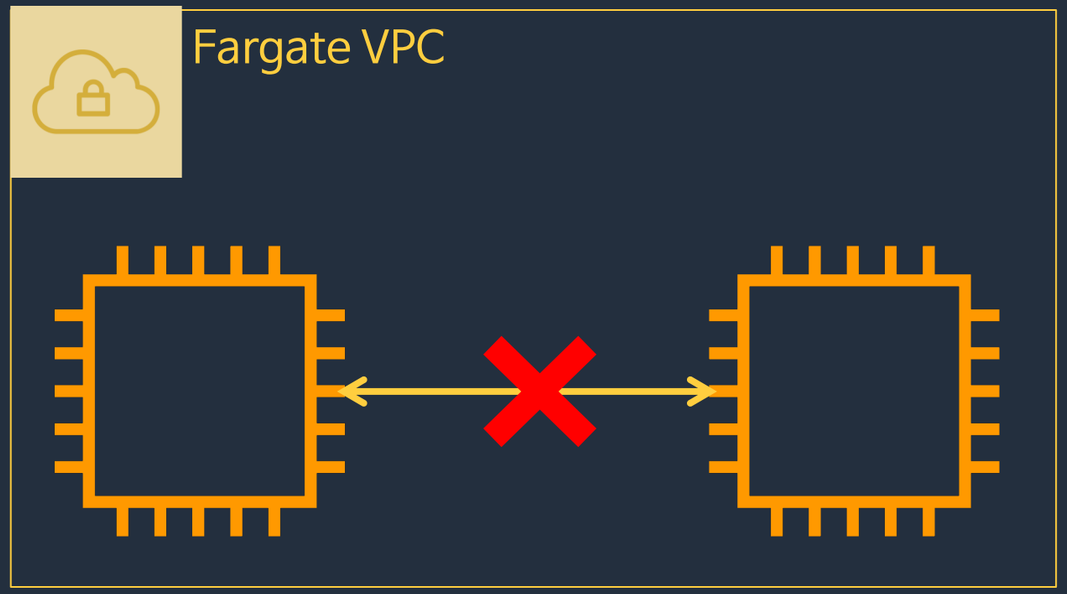
Tens of millions of Fargate task launches a week → Tens of millions of MicroVM launches a week (lightweight) → Much lower rate of EC2 instance launches (heavy-duty)

Task Security & Isolation



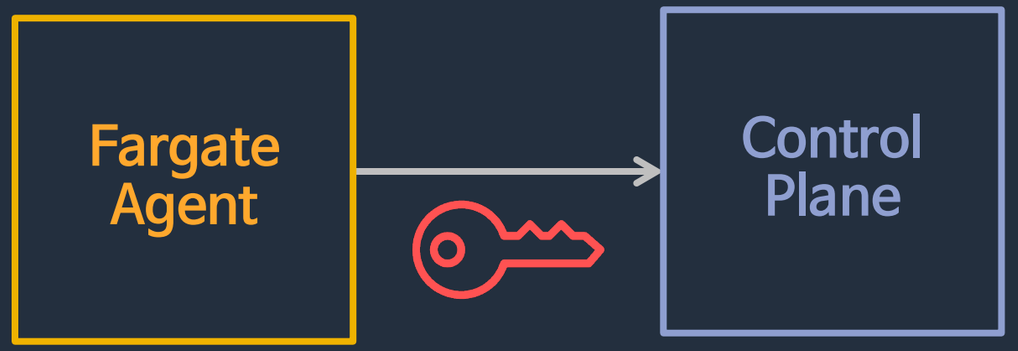
보안 완화 조치

Instance-to-Instance



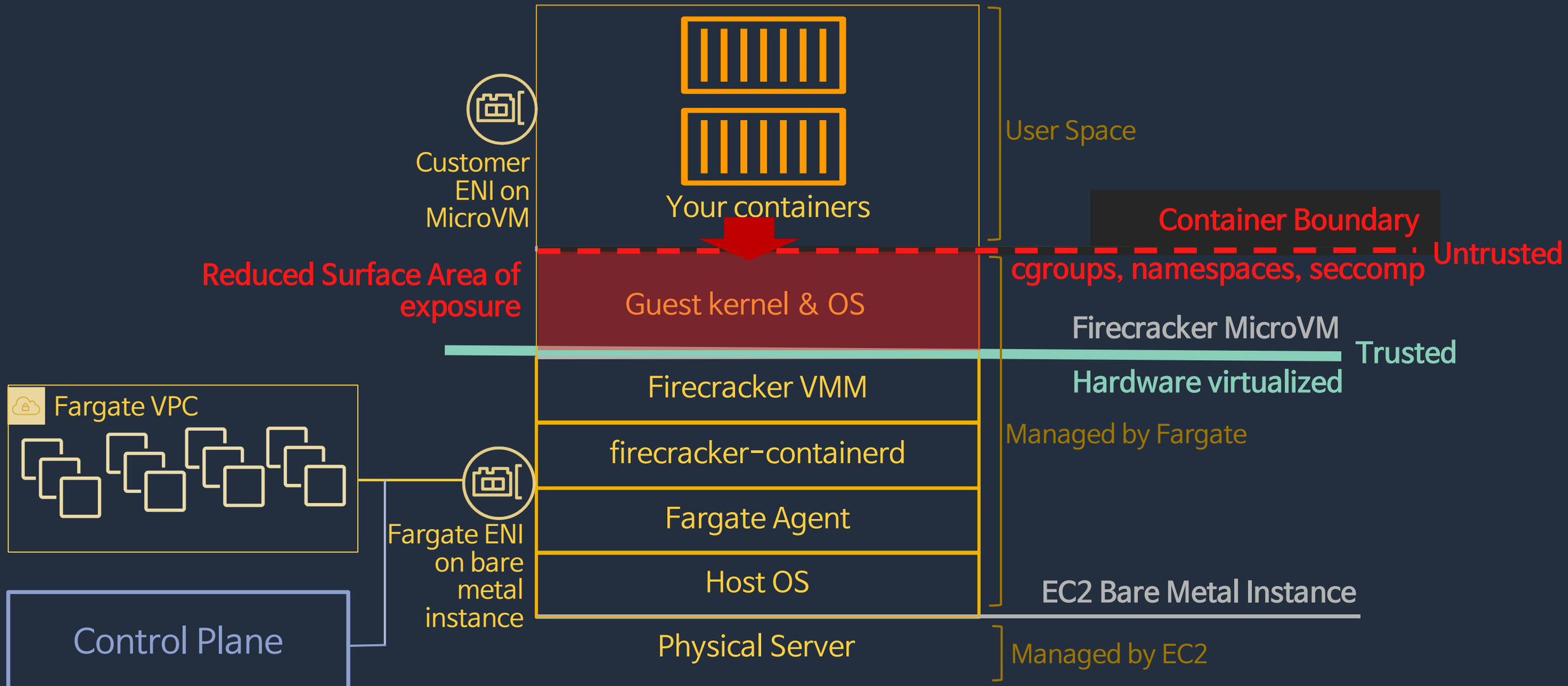
보안 그룹은 인스턴스 간의 모든
인스턴스 간 통신을 차단

Instance-to-Control Plane

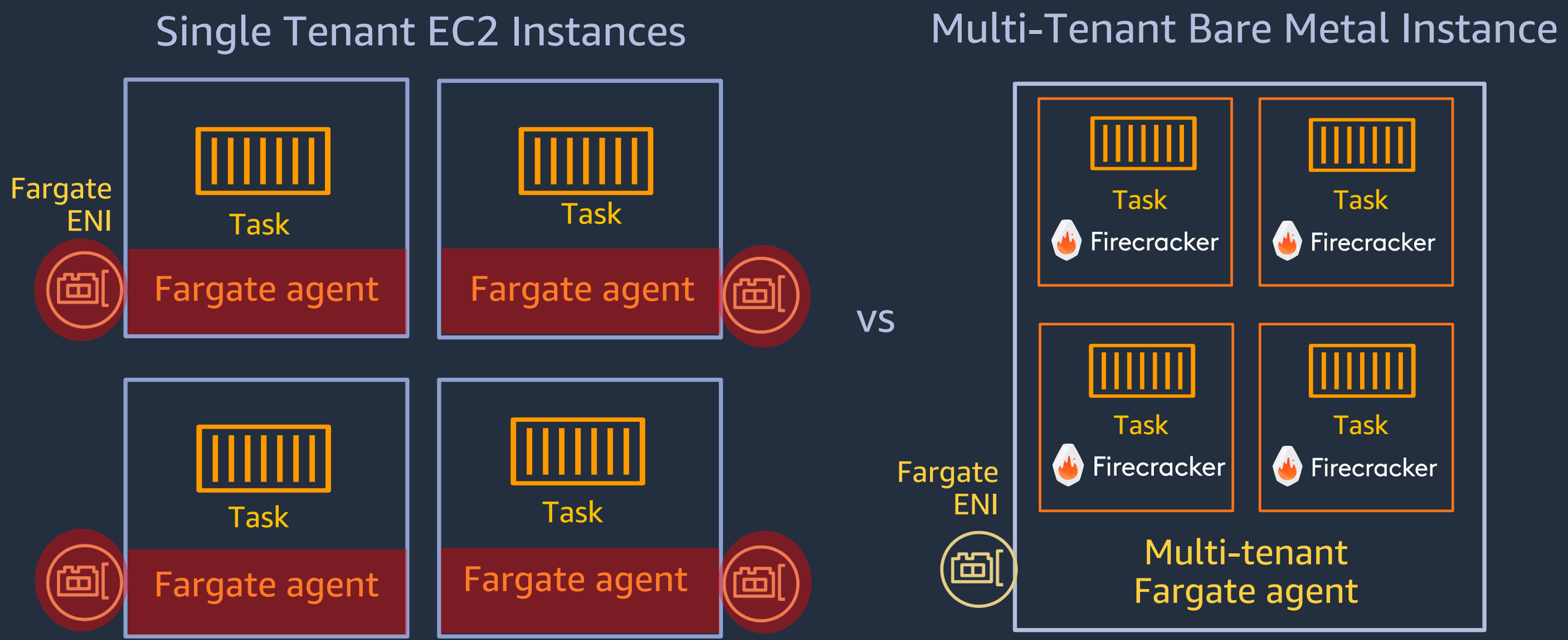


최소 권한으로 설정된 권한
에이전트는 로컬 작업에 대한 상태만 검색/변경

How Firecracker helps isolation

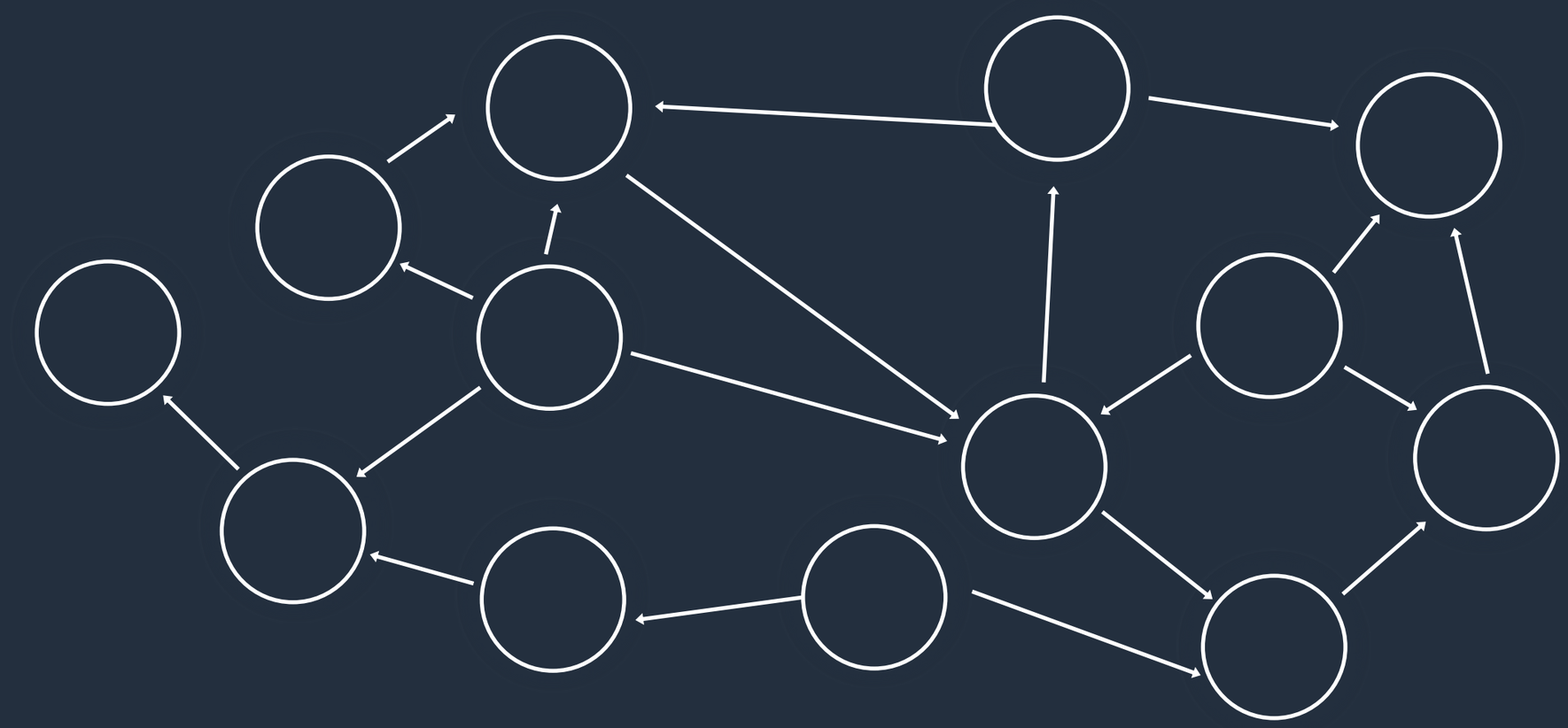


How Firecracker helps utilization: Data Plane 오버헤드



Service Connect

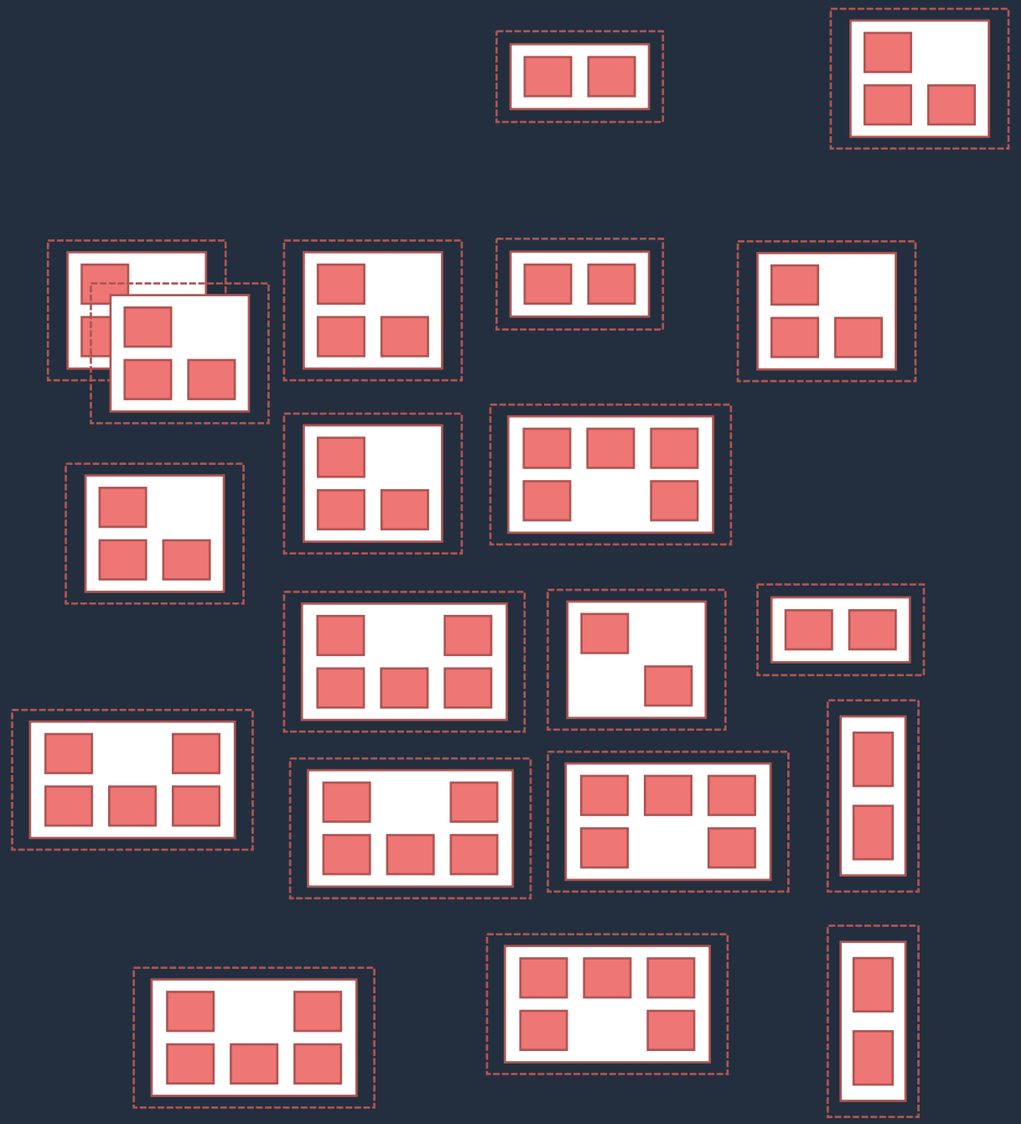
Complexity of modern architecture



고가용성을 위해서는 리소스의 현재 인스턴스 및 연결 가능한 인스턴스와의 안정적인 검색과 연결이 중요합니다.

Connecting microservices reliably is hard

- 많은 수의 서비스
- 기하급수적으로 증가하는 복잡성
- 여러 버전과 단계가 공존
- 동적으로 확장되는 인프라
- 비정상적인 엔드포인트 교체



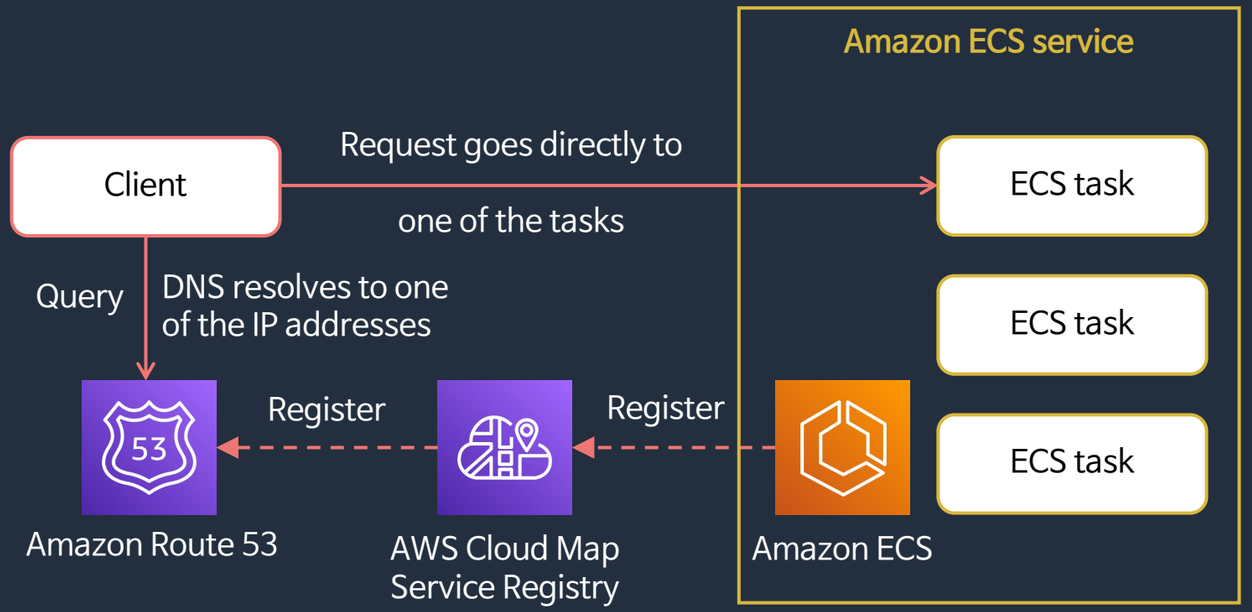
Amazon ECS 내부 통신 방법

- Amazon ECS Service Discovery
- Amazon Elastic Load Balancer
- AWS App Mesh
- Amazon ECS Service Connect

Using Amazon ECS Service Discovery

- 매우 간단한 DNS 검색 서비스
- Cloud Map은 자동으로 Route53에 필요한 항목을 등록하고 업데이트
- 클라이언트는 Route53에서 얻은 연결 정보를 활용해서 직접 통신

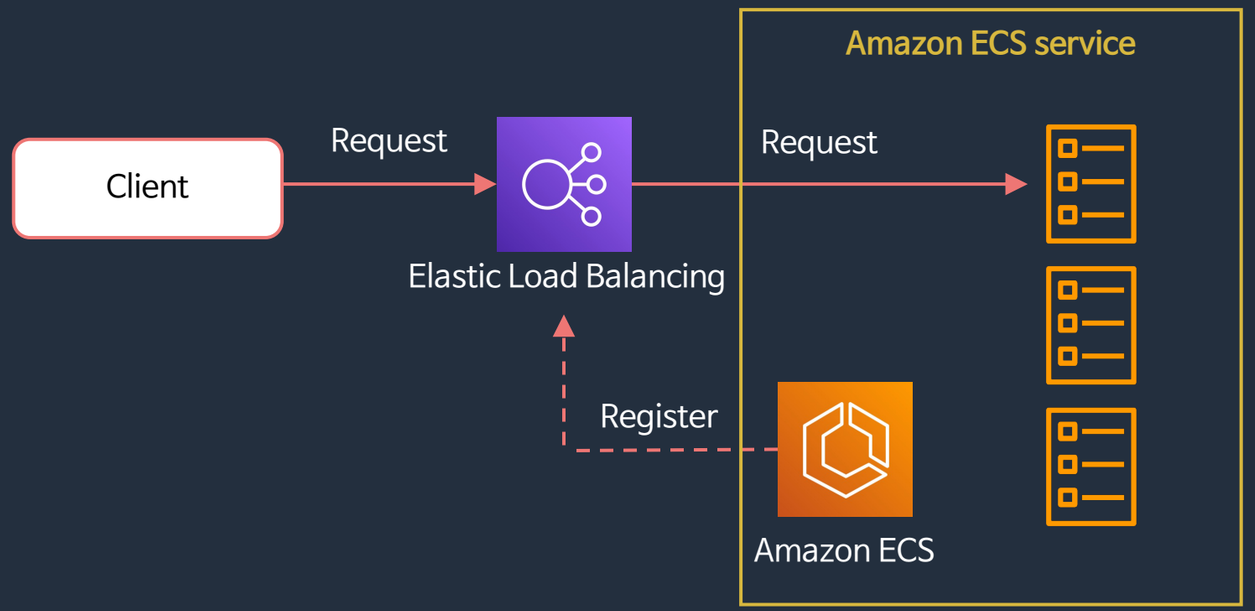
- 트래픽에 대한 Telemetry 정보 제공 안함
- DNS 검색은 라운드-로빈 방식만 지원



Using Elastic Load Balancing

- 가장 많이 활용되는 방법으로 클라이언트는 ELB를 통해 통신
- 작업(Task)은 ELB의 타겟 그룹으로 설정됨
- ELB에서 제공되는 Telemetry 정보

- ELB 자체는 비용이 발생하는 추가 리소스
- 레이턴시가 추가됨



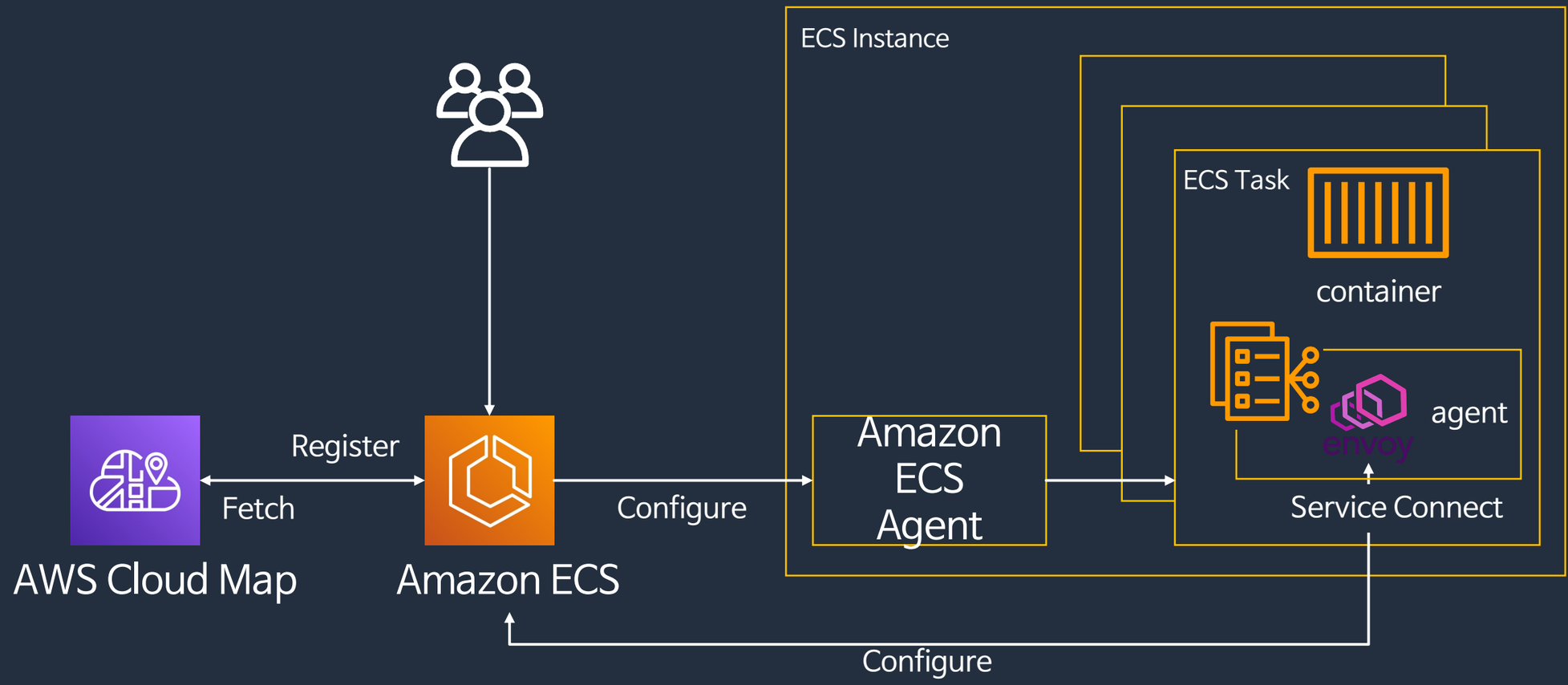
잠깐, 마이크로서비스 간 내부 통신이라고?

- Service Mesh 개념의 등장과 이를 위한 Sidecar proxy 패턴
- Envoy 프로젝트
  - “The network should be transparent to applications. When network and application problems do occur it should be easy to determine the source of the problem.”
  - 2016년 Lyft에서 시작한 사내 프로젝트
  - OSS project
  - CNCF(Cloud Native Computing Foundation) project
  - Istio, AWS App Mesh, Amazon ECS Service Connect 에서 활용

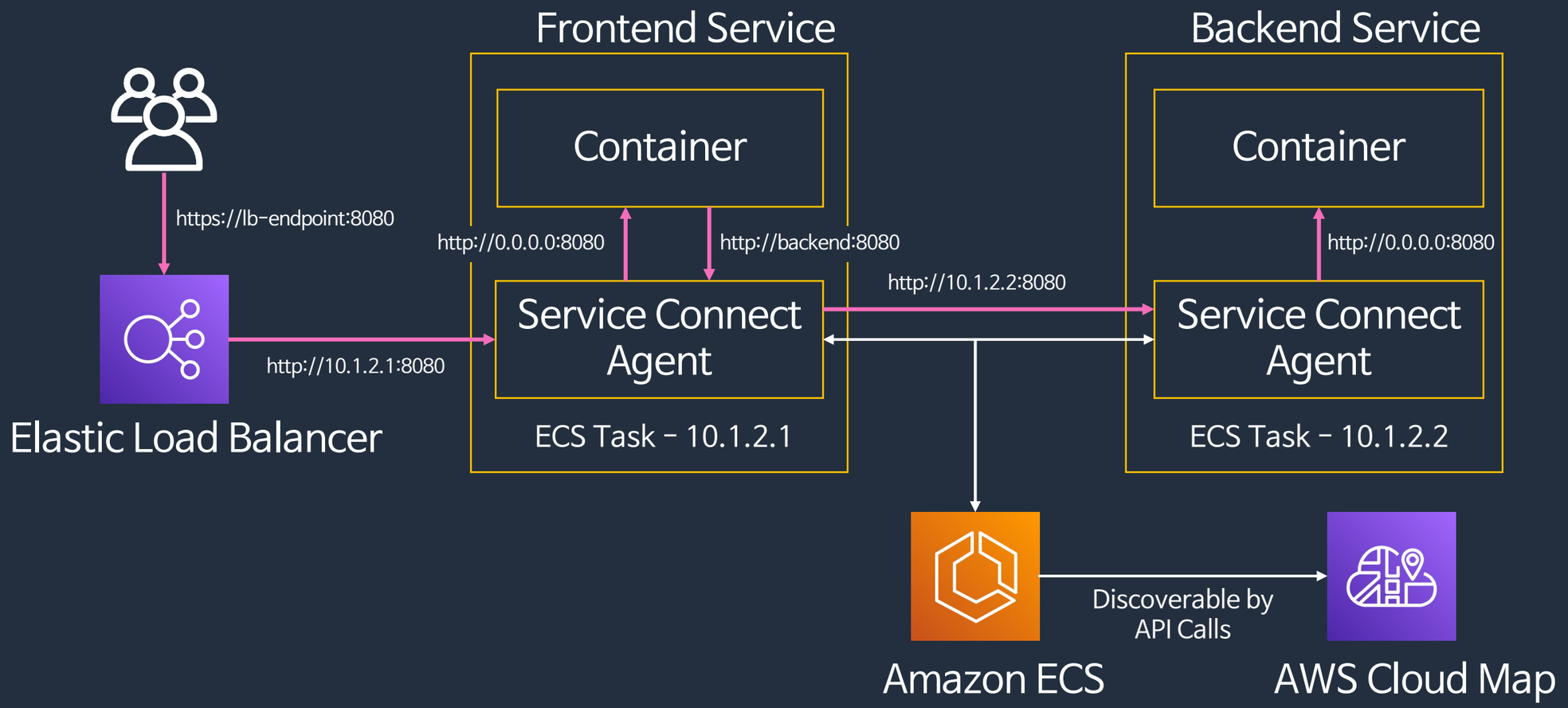
Using Amazon ECS Service Connect

- Amazon ECS Service Discovery의 진화된 버전
- 애플리케이션 개발자에게 동일한 간단한 경험과 풍부한 기능 제공
  - Envoy Proxy 기반의 Service Connect Agent가 Sidecar로 동작
  - Amazon ECS 콘솔 및 CloudWatch에서 제공되는 풍부한 트래픽 원격 분석 기능
  - 트래픽 헬스 체크
  - 복원력을 갖춘 통신을 위한 자동 재시도
  - 강력해진 롤링 배포

Service Connect 설정 과정



Service Connect 트래픽 흐름



Service Connect 타입 – 클라이언트, 클라이언트-서버

- **1/ 클라이언트 타입**
  - 프론트엔드, Reverse Proxy, ELB 등을 통해 외부 트래픽을 수신하는 경우 사용
  - 서비스는 통신을 위해 다른 서비스를 검색해야 하지만, 자신은 검색될 필요가 없음

- **2/ 클라이언트-서버 타입**
  - 백엔드, 미들웨어, 내부 통신이 필요한 마이크로 서비스 등에서 사용
  - 서비스는 통신을 위해 다른 서비스를 검색하고, 마찬가지로 다른 서비스가 검색할 수 있어야 함
  - 서비스 검색에 필요한 Port alias, Discovery, DNS, Port 등 설정이 필요함

Service Connect 배포하기

- 클라이언트를 배포하기 전에 백엔드 서비스를 배포하여 안전한 배포 및 롤백 보장 필요
- 1/ 백엔드 서비스에 대한 클라이언트-서버 서비스 구성
  - 네임스페이스에 Service Connect 엔드포인트를 생성
  - 프론트엔드에서 사용하는 것과 동일한 DNS 또는 클라이언트 별칭을 사용
- 2/ 프론트엔드 서비스에 클라이언트 or 클라이언트-서버 서비스 연결 구성 추가

Service Connect 고려사항

- Service Connect Agent는 Sidecar Proxy이기 때문에 자체적인 CPU, Memory 리소스가 필요
  - 유휴 상태일 때 Service Connect 컨테이너는 100개의 CPU 유닛과 40MiB의 메모리 사용
  - Service Connect Proxy 컨테이너의 작업 CPU 및 메모리에 256개의 CPU 유닛과 최소 64MiB의 메모리를 추가하는 것을 권장
- 롤링 업데이트만 지원
- ECS 서비스는 오직 하나의 네임스페이스에 설정 가능
- ECS 플랫폼 버전 확인 필요
  - Fargate Linux platform 1.4.0+ / ECS Agent 1.67.2+

Thank you!

신 정 섭