



Amazon ECS Essentials Tips #1-2

신정섭

Solutions Architect

Agenda

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

대규모 트래픽 준비하기

사전에 준비하기

1. Quota & API Throttling 체크하기

- Amazon ECS도 AWS의 다른 리소스와 함께 운영되기 때문에 대규모 서비스에서는 Quota 제한에 걸리는 경우가 있음 (상향 요청 필요)
 - 예를 들어 ALB의 경우, ALB 당 / Region 당 Target 개수가 1000개로 제한됨
- ECS 운영 중에 과도한 ECS API가 요청되면 AWS 에서 Throttling 함
 - AWS CloudTrail 로그를 통해 throttling 이벤트를 감지할 수 있음
 - Throttling 이벤트는 Amazon EventBridge 에도 emit 됨
- 만약 직접 ECS API를 다루는 경우 AWS SDK 등을 활용하지 않는다면 재시도 로직을 직접 구현해야 함
 - Amazon EventBridge + AWS Lambda 조합으로 backoff retry

사전에 준비하기

2. Amazon ECS Service Auto Scale 사용하기 (예약된 스케줄)

- Amazon ECS 는 [Application Auto Scaling](#)의 예약된 조정을 지원함
 - scalable targte 생성
 - scheduled-action 정의

사전에 준비하기

3. Amazon EventBridge 사용하기

- Amazon ECS 클러스터의 현재 상태에 대한 거의 실시간에 가까운 알림을 받을 수 있음
- [Amazon ECS 이벤트](#)는 Amazon EventBridge 으로 전송함
 - 컨테이너 인스턴스 상태 및 변경
 - Task 상태 변경 (특히, Spot 인스턴스의 종료 알림)
 - Service 액션(INFO, WARN, ERROR) 및 배포 상태 변경
- 각 이벤트는 AWS Lambda 등을 이용하여 목적에 맞는 처리가 가능함(알림 처리)
- Amazon EventBridge Scheduler를 사용하면 Cron 작업 가능

사전에 준비하기

4. Throttling or Virtual Waiting Room 준비하기

- Throttling
 - 전체 시스템의 장애보다 Throttling을 통해서 일부 사용자에게만 Fallback 정책 적용
 - Amazon API Gateway 등을 활용하면 Throttling 설정 가능
 - 사용자의 UX 경험을 위한 API 등은 별도 분리 or Lambda 처리
- Virtual Waiting Room 솔루션 활용
 - 콘서트 / 스포츠 티켓 판매, 블랙 프라이데이 등의 특별 할인 행사 / 온라인 수강 신청 등
 - <https://aws.amazon.com/ko/solutions/implementations/virtual-waiting-room-on-aws/>

사전에 준비하기

5. Spot Instance 활용하기

- 평소에도 Spot Instance를 활용하면 Spike 트래픽에 대응 가능
- 대규모 트래픽의 경우 Spot Instance를 적극 활용해서 사전에 프로비저닝하기
- Spot 인스턴스 회수 가능성을 최소로 만드려면?
 - 여러 리전 및 가용 영역을 사용
 - 여러 인스턴스 유형을 사용
 - EC2 Spot 인스턴스를 사용하는 경우, “용량 최적화 전략”을 사용

빠르게 배포하기

1. 컨테이너 이미지 최적화하기

- 컨테이너 기동 시간의 50% 이상은 컨테이너 이미지 다운로드 시간
- 컨테이너 이미지는 작을수록 배포와 스케일링에 유리함
- 최적화된 베이스 이미지(Slim 이미지 & Multi Stage 빌드)를 만들자
- 애플리케이션은 꼭 필요한 Dependency만 포함하도록
- 애플리케이션 크기 자체가 크다면 분리 고려

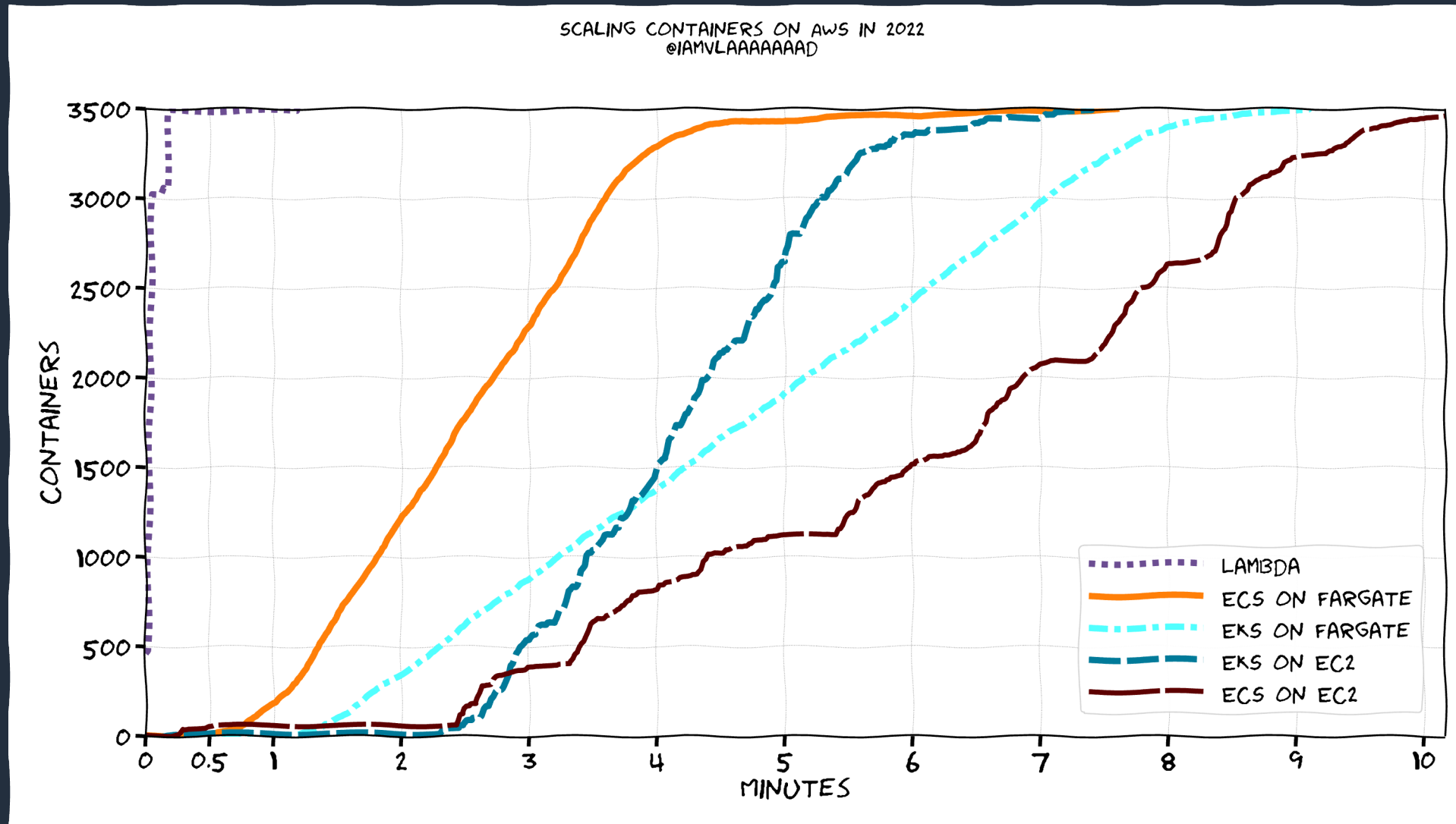
빠르게 배포하기

2. Task 부트스트랩 시간 올리기

- Task를 구성하는데 여러 가지 CPU를 필요하는 프로세스가 진행됨
 - start-up 스크립트 실행 / 압축 해제 / 라이브러리 다운로드 등
 - 애플리케이션 부트스트랩(Spring Boot의 경우 Bean 생성 등)
- [ECS Agent Metadata API](#) : Task 메타데이터 및 Docker 통계 정보 확인 가능
 - Metric을 CloudWatch로 전송해서 부트스트랩 시간 분석 가능
- vCPU를 조금 더 할당해주면 전체적인 부트스트랩 시간이 올라갈 수 있음
- ECS on EC2의 경우 Task의 수평 확장이 빠름 (binpack)

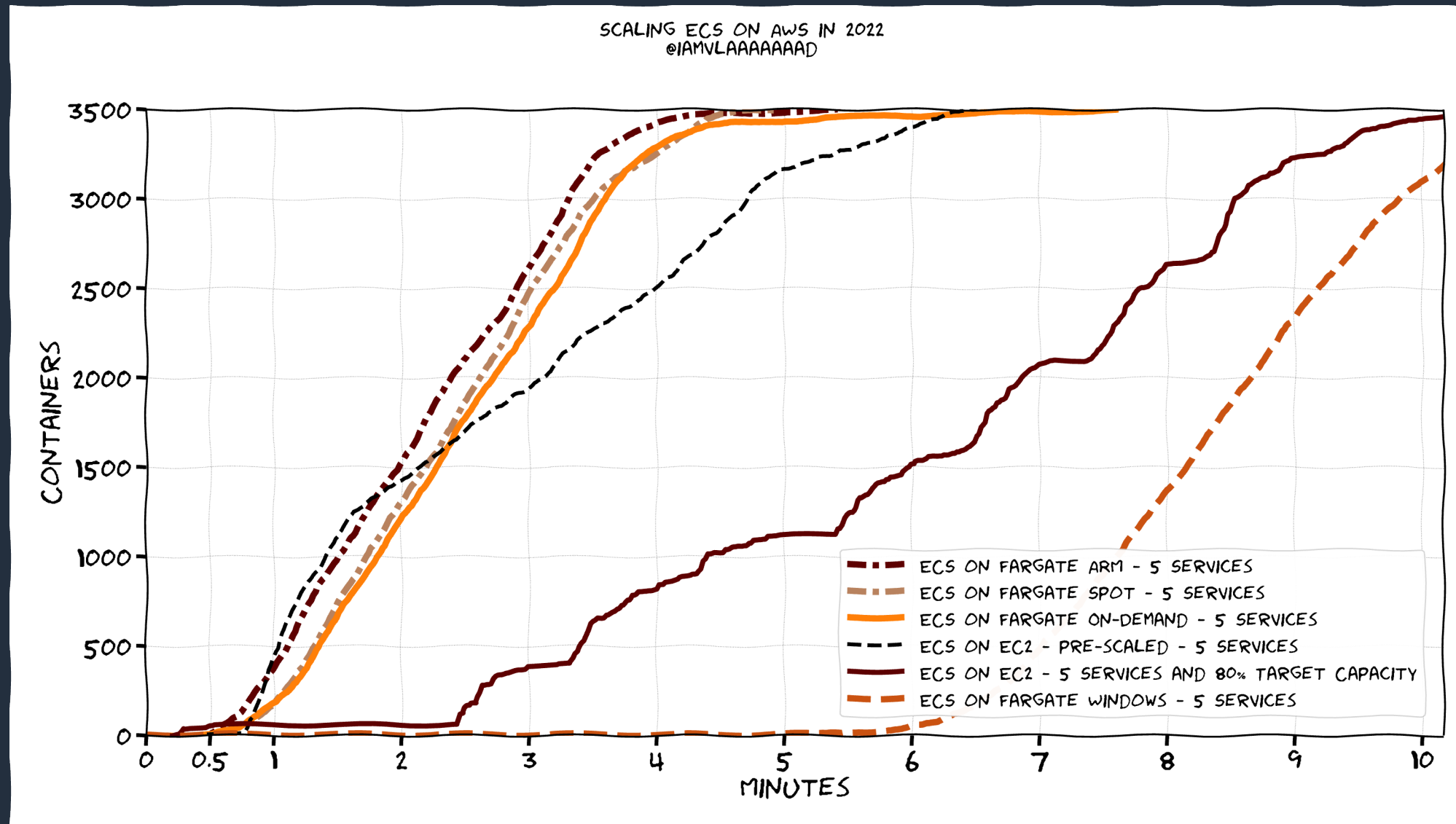
빠르게 배포하기

3. Fargate 사용하기



빠르게 배포하기

3. Fargate 사용하기



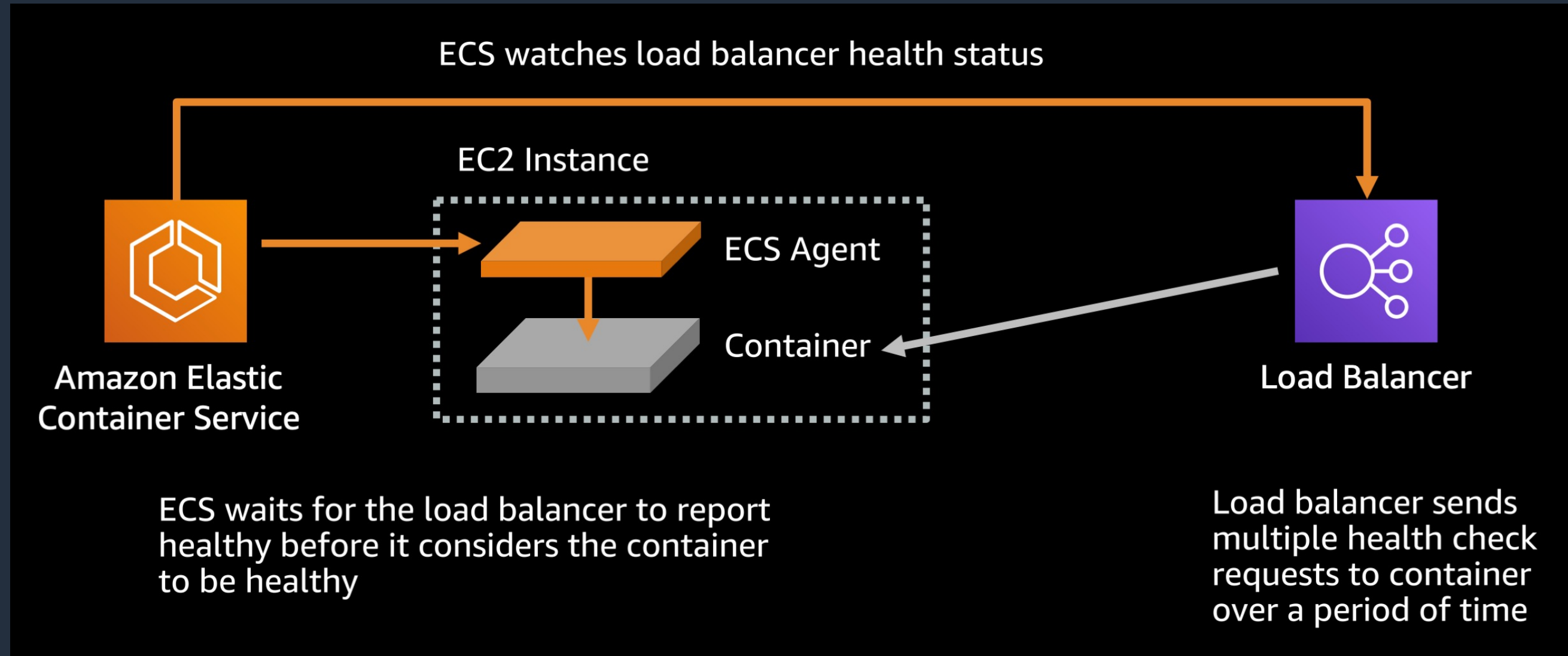
빠르게 배포하기

3. Fargate 사용하기

- ECS의 경우, Fargate는 EC2 보다 스케일링이 빠름
- ECS의 경우, Fargate는 ARM 프로세서도 지원함 (x86 대비 조금 빠름)
- Fargate는 매년 스케일링 속도 개선이 이뤄지고 있음 (2020 대비 2022년 12배 빠름)

빠르게 배포하기

4. 로드밸런서 healthcheck 설정 확인하기



빠르게 배포하기

4. 로드밸런서 healthcheck 설정 확인하기

- healthcheck 기본 설정값은 새로운 Task에 트래픽 유입까지 최대 2.5분이 소요
 - HealCheckInternalSeconds : 30s
 - HealthyThresholdCount : 5
- 빠른 스케일링(배포)을 위한 권장 값
 - HealCheckInternalSeconds : 5s
 - HealthyThresholdCount : 2

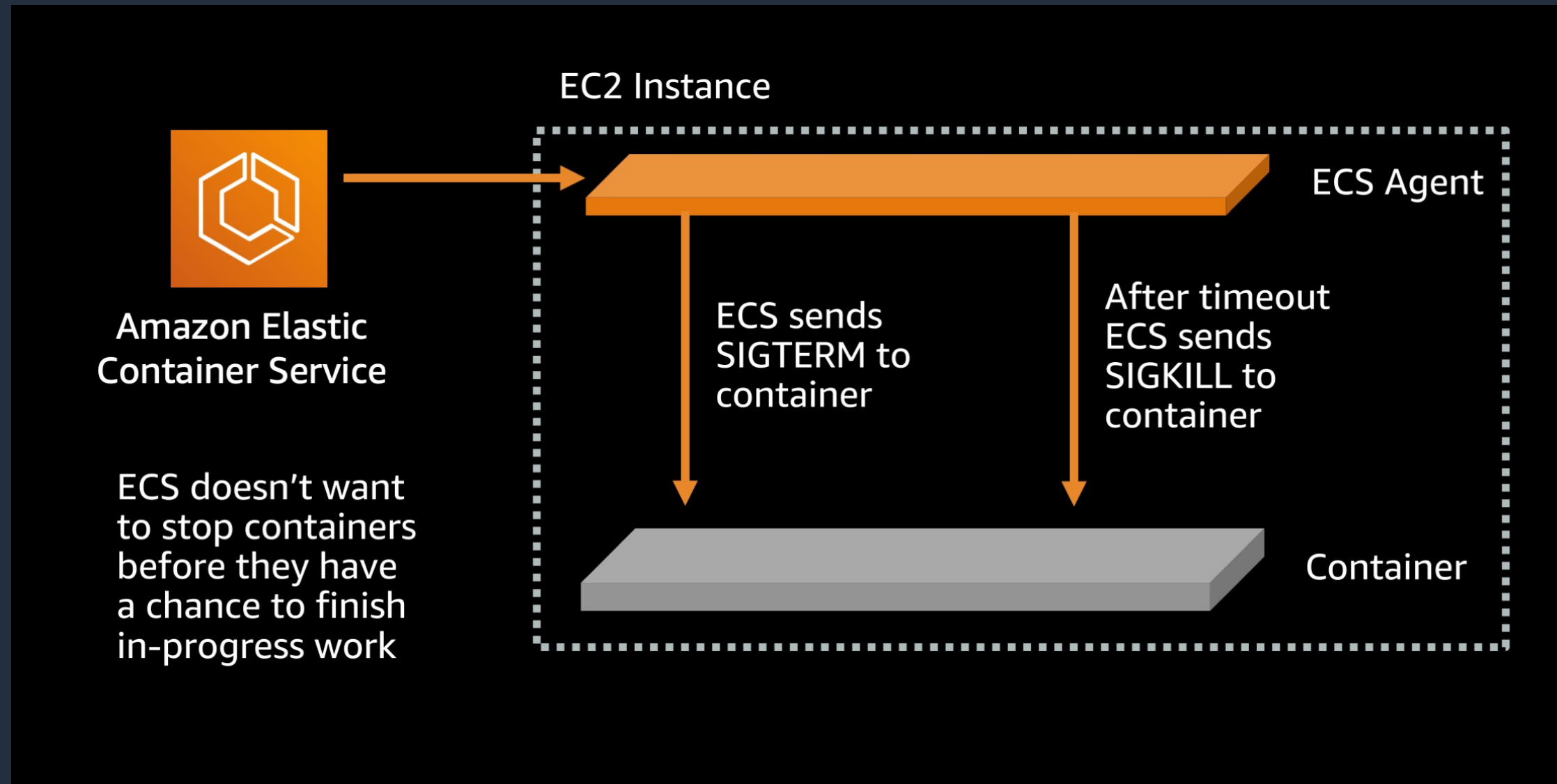
빠르게 배포하기

5. 로드밸런서 connection draining 설정 확인하기

- Task를 종료하면 로드 밸런서는 다운스트림 컨테이너에 대한 새 연결 전송을 중단하지만 기존 연결이 저절로 종료될 때까지 대기함
 - `deregistration_delay.timeout_seconds` : 300s
- 애플리케이션 평균 응답시간이 1초 미만인 경우에는 불필요한 대기 시간
- 빠른 Drain을 위한 권장 값
 - `deregistration_delay.timeout_seconds` : 5s

빠르게 배포하기

6. SIGTERM 반응성 높이기



빠르게 배포하기

6. SIGTERM 반응성 높이기

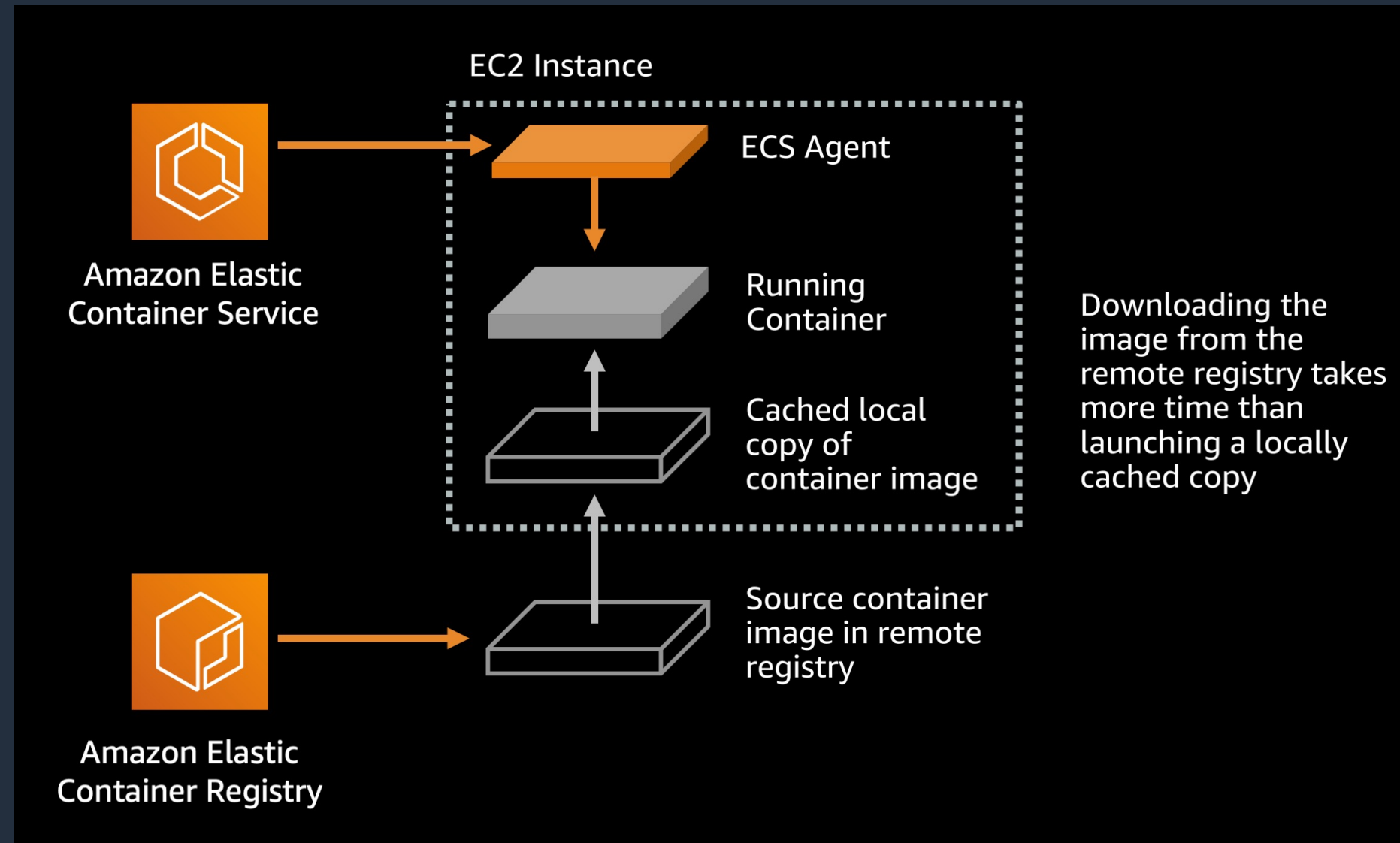
- 로드밸런서 드레인이 완료되면 ECS Agent는 컨테이너에 SIGTERM 신호를 전송
 - 애플리케이션이 수행 중인 작업을 완료하도록 경고하는 것
 - 해당 신호를 받으면 애플리케이션의 Graceful 섯다운 동작을 해야하지만... 대부분 무시함

```
process.on('SIGTERM', function() {  
  server.close();  
})
```

- SIGTERM 신호 이후 기본 대기 시간 이후에 SIGKILL 명령이 내려짐 (ECS_CONTAINER_STOP_TIMEOUT : 30s)
- 애플리케이션의 응답 속도가 평균 1초라면 이 시간을 2s로 설정하는 것이 권장됨
 - 가장 좋은 방법은... SIGTERM 처리를 애플리케이션에서 처리하자

빠르게 배포하기

7. 컨테이너 캐시 설정하기



빠르게 배포하기

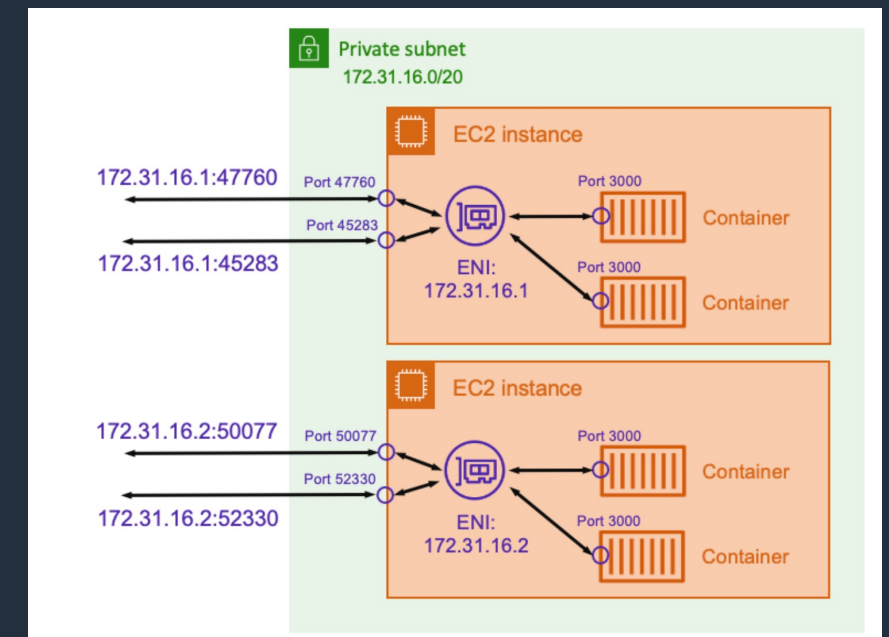
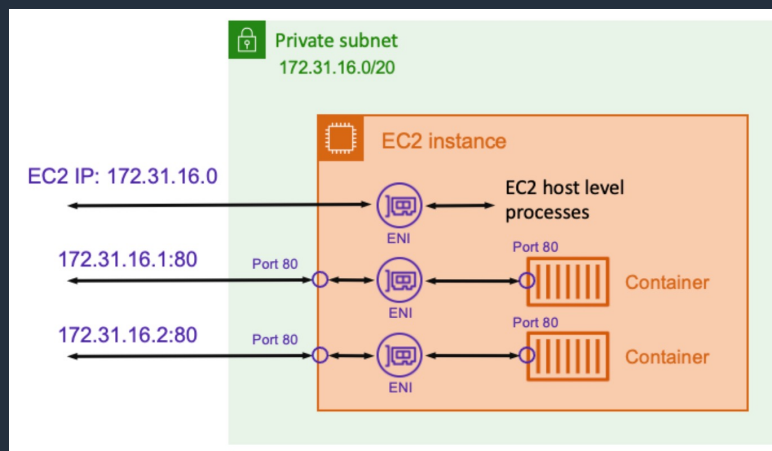
7. 컨테이너 캐시 설정하기

- ECS Agent는 기본적으로 컨테이너 이미지를 항상 원격 저장소에서 Pull
 - ECS_IMAGE_PULL_BEHAVIOR : default
- 권장 설정 값
 - ECS_IMAGE_PULL_BEHAVIOR : once or prefer-cached
 - ECS가 원격 레지스트리에서 다운로드하는 대신 호스트의 디스크 캐시에 있는 이미지를 사용
- Fargate의 경우에는 1 Task 1 VM 구조라 로컬 캐싱이 불가능
 - vCPU 추가 할당 / 작은 base 이미지 / ECR 이 같은 리전으로 사용
 - [Seekable CIO 기술을 활용하여 컨테이너 이미지 Lazy Loading 하기](#)

빠르게 배포하기

8. 네트워크 모드 수정하기

- awsvpc 모드를 사용하면 Amazon ECS는 각각의 Task 별로 ENI를 생성하여 VPC private IP를 할당함
 - Task 별로 각각의 보안 그룹 설정이 가능하고, 통신을 제어할 수 있는 유연성이 향상됨
 - Task를 배포하는 상황에서 ENI를 프로비저닝하고 설정하는 시간이 소요됨
- 빠른 배포를 위해서라면 brdige 모드를 고려





Thank you!

신정섭