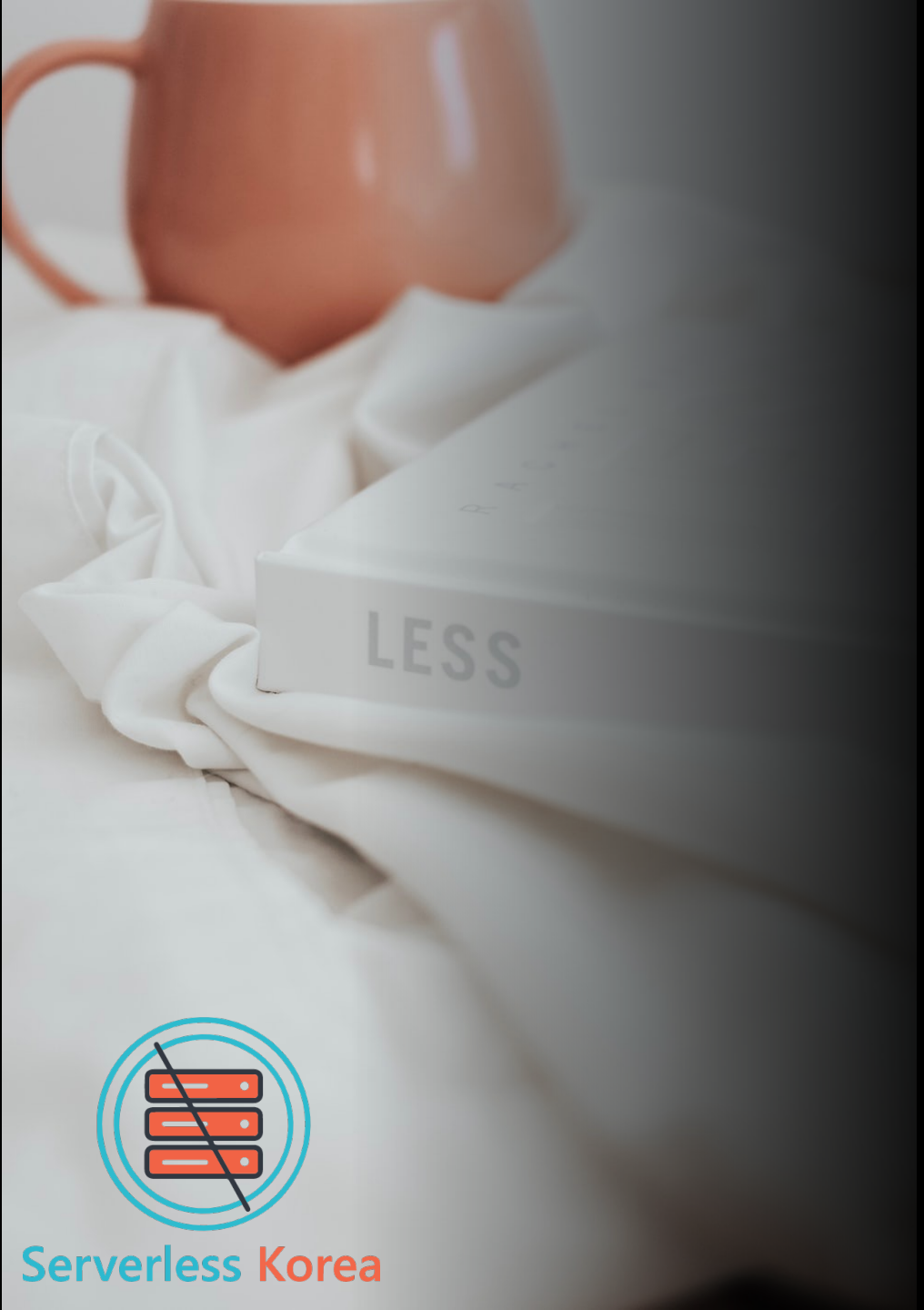


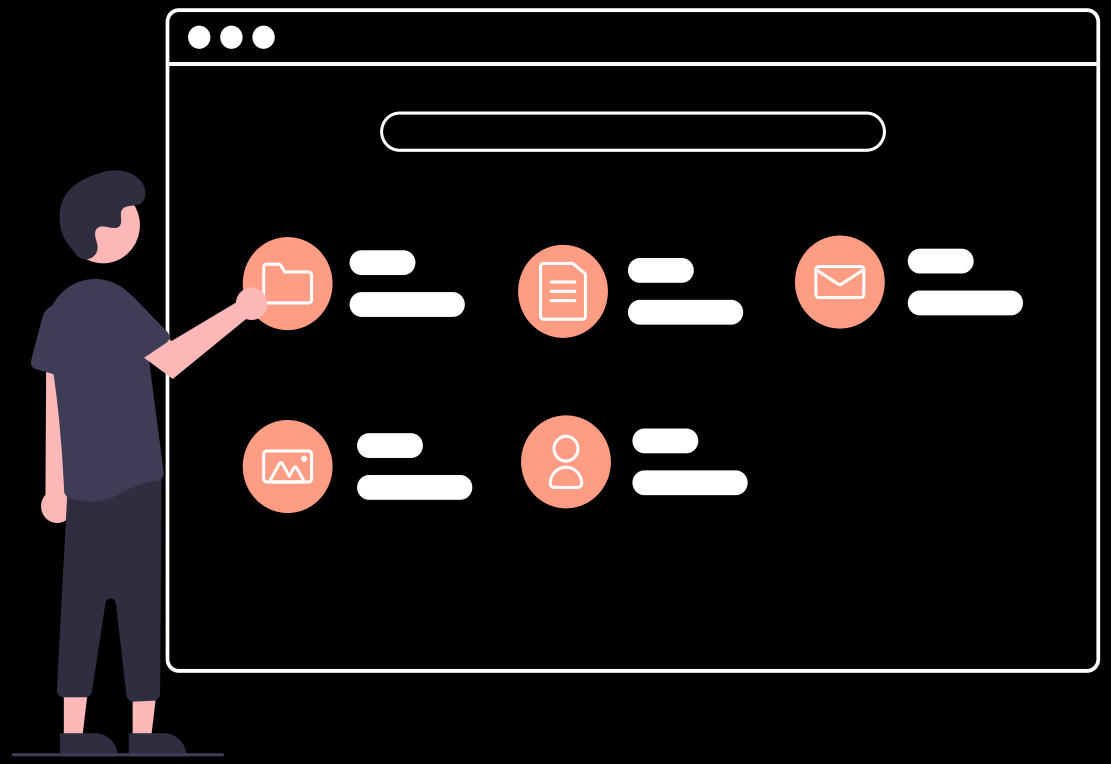
AWS Fargate로
쉽고 빠르게 웹서비스 만들기

Serverless Korea

CLOUDMATE
Managed Service Expert
Solution Architect Junior 정휘영

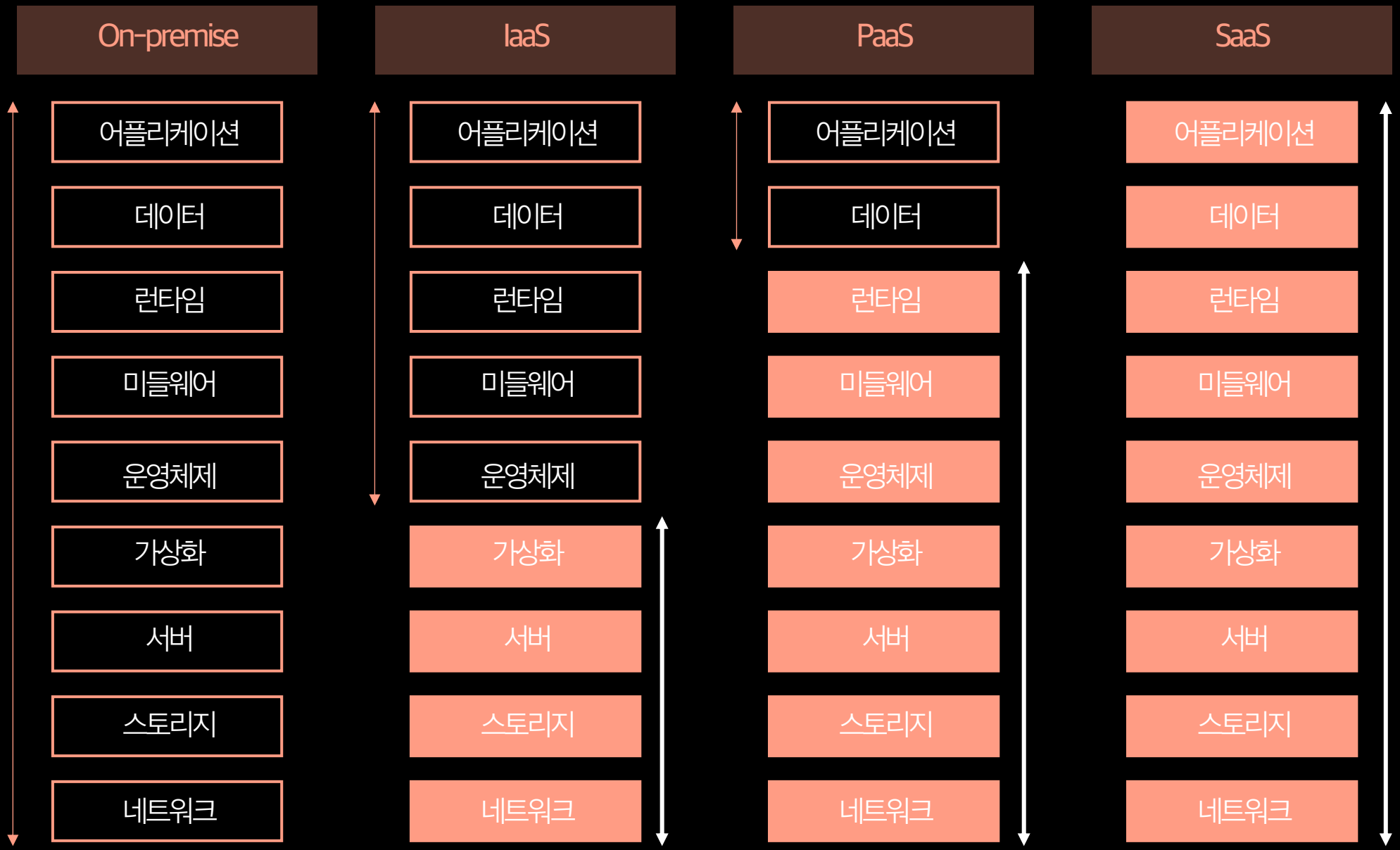

OPENUP

Index



시작에 앞서..

Shared Responsibility Model



사용자 입장에서 생각해보기

On-premises	IaaS	PaaS	SaaS
			
차를 만들어서 가기	직접 운전해서 가기	차를 빌려서 타고가기	택시 타고 가기

사용자 입장에서 생각해보기

On-premises	IaaS	PaaS	SaaS
			
차를 만들어서 가기	직접 운전해서 가기	차를 빌려서 타고가기	택시 타고 가기

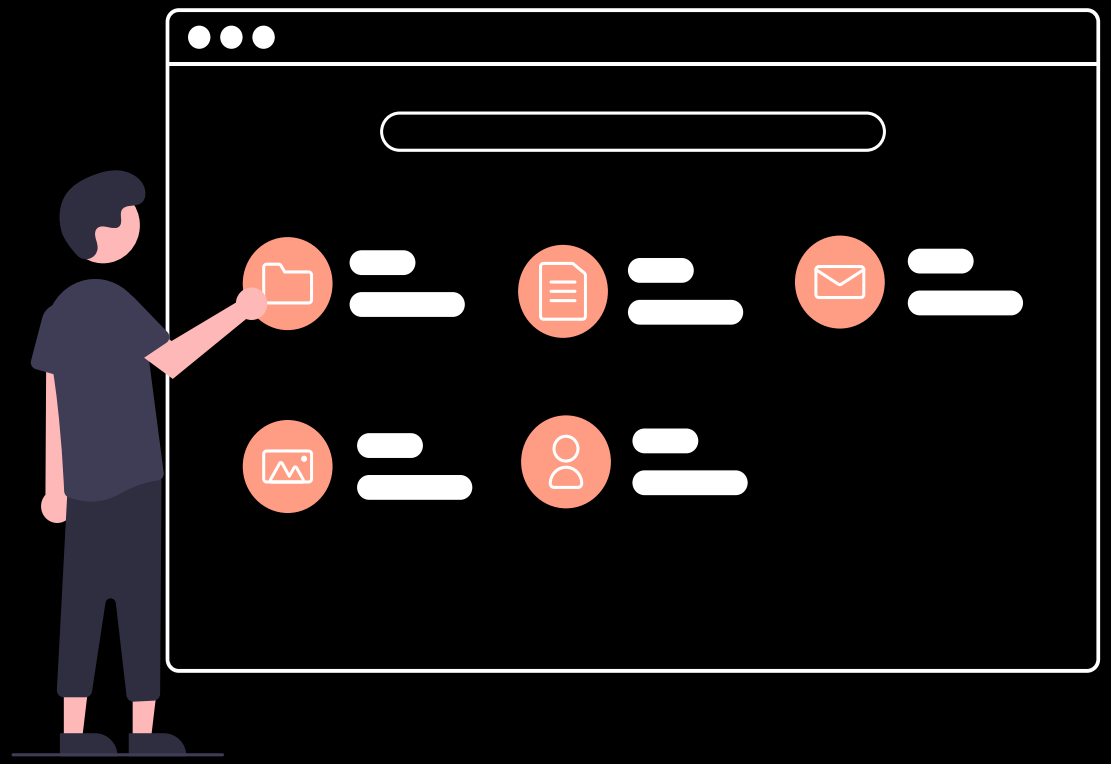
사용자 입장에서 생각해보기

On-premises	IaaS	PaaS	SaaS
			
차를 만들어서 가기	직접 운전해서 가기	차를 빌려서 타고가기	택시 타고 가기

사용자 입장에서 생각해보기

On-premises	IaaS	PaaS	SaaS
			
차를 만들어서 가기	직접 운전해서 가기	차를 빌려서 타고가기	택시 타고 가기

관리의 범위를 CSP와 사용자가 어디까지 책임 지는지가 핵심!



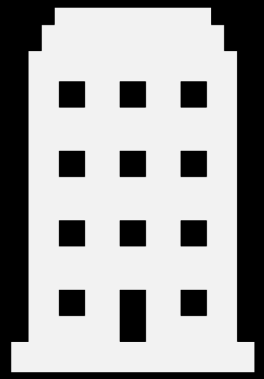
Serverless

Serverless

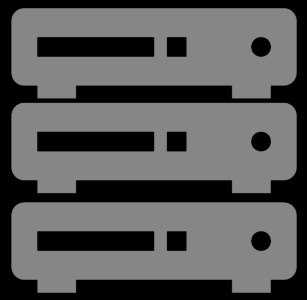
- 서버리스 = 서버가 없는게 아니다
- CSP가 정해놓은 틀 안에서 Application을 띄우는 것
- 사용자는 Application의 레벨을 컨트롤

Application 배포 시간의 변화

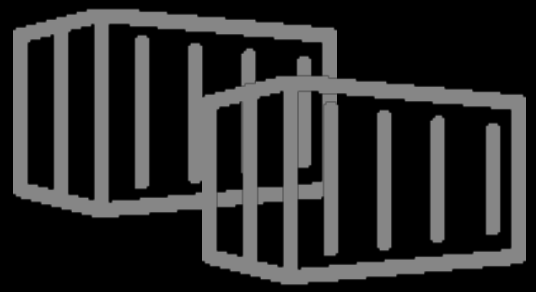
Weeks



Minutes

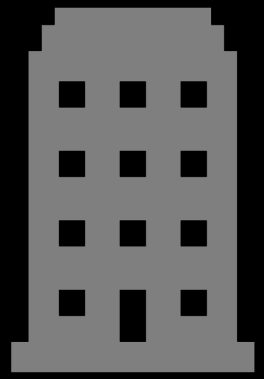


Seconds

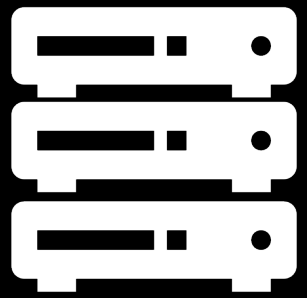


Application 배포 시간의 변화

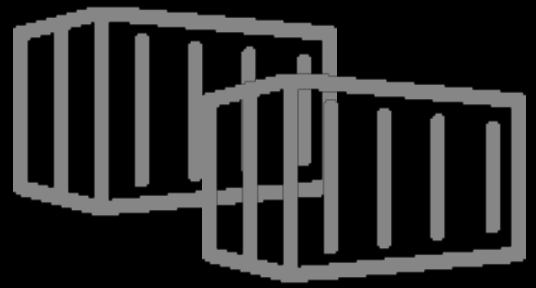
Weeks



Minutes

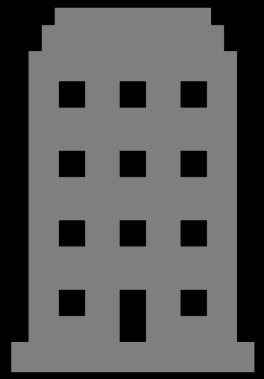


Seconds

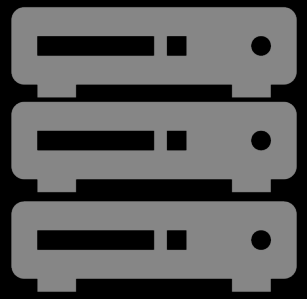


Application 배포 시간의 변화

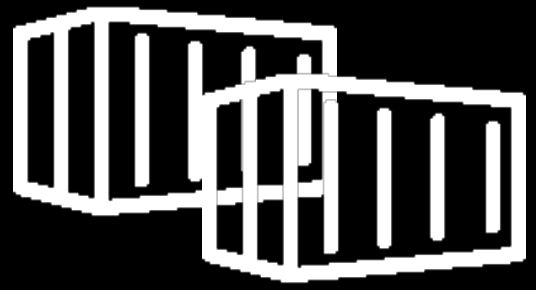
Weeks



Minutes



Seconds



관리대상을 줄이는 것도 시간절약

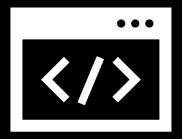
쉽게 이해해보는 Serverless

띵 동

- 대기자가 없다
- 부르면 온다 (실행하면 온다)
- 없는 것이 아니라 보이지 않는 것
- 사용자는 필요한 것만 사용한다.

사용자는 어플리케이션 레벨만 사용하자

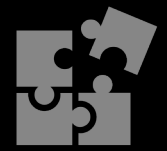
Serverless 의 장점



서버 준비
&
관리 불필요

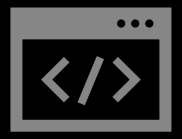
미실행시 과금 X

사용량 기반
자동 확장



가용성

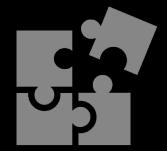
Serverless 의 장점



서버 준비
&
관리 불필요

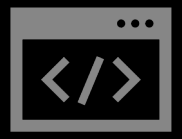
미실행시 과금 X

사용량 기반
자동 확장



가용성

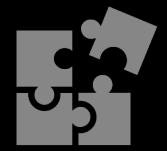
Serverless 의 장점



서버 준비
&
관리 불필요

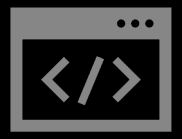
미실행시 과금 X

사용량 기반
자동 확장



가용성

Serverless 의 장점



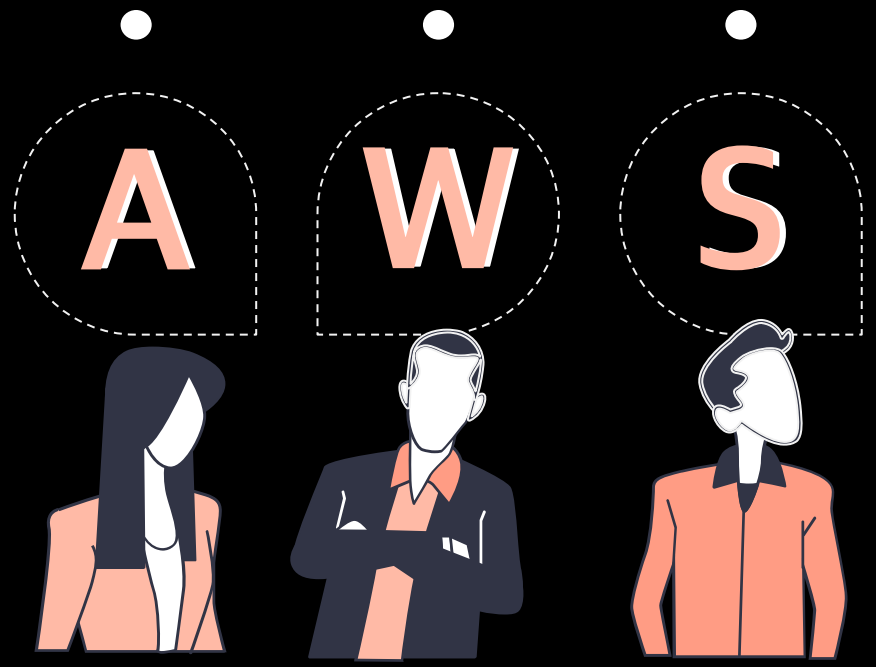
서버 준비
&
관리 불필요

미실행시 과금 X

사용량 기반
자동 확장

가용성

토이 프로젝트나 초기 서비스 런칭할 때 추천



AWS Serverless

AWS Lambda ?

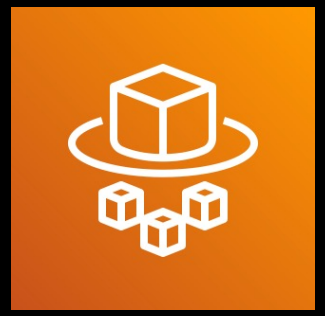
AWS Lambda

애플리케이션이나 백엔드 서비스에 대한 코드를 별도의 관리 없이 실행

- 서버 프로비저닝 / 관리 없이 코드 실행
- 사용량에 따른 지속적 규모 조정
- 높은 가용성 및 자동 복구
- 사용한 만큼 지불

여러분은 코드만 올려요 ☺

AWS Fargte ?



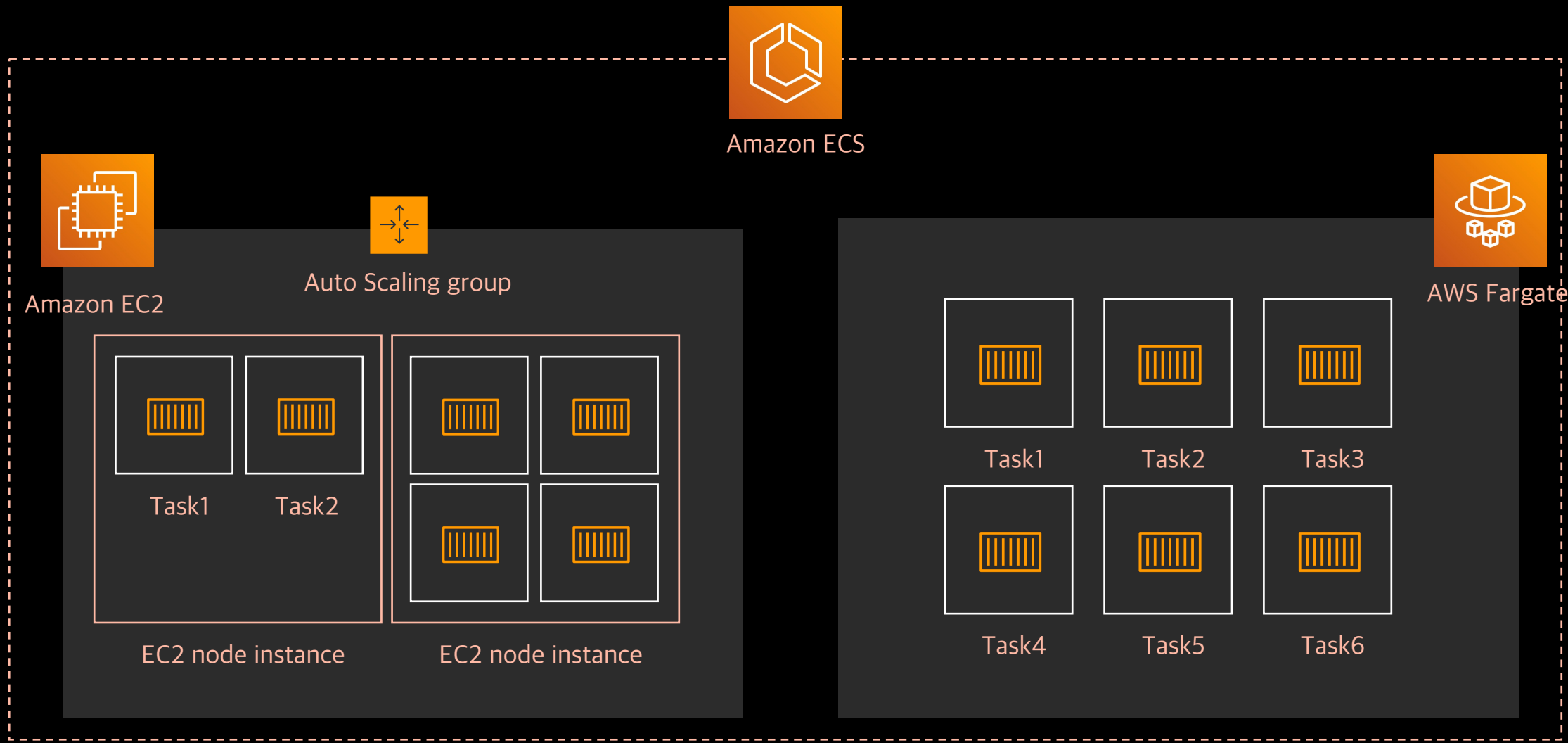
AWS Fargate

인프라를 관리할 필요 없이 컨테이너를 배포하고 관리하는 컴퓨팅 엔진

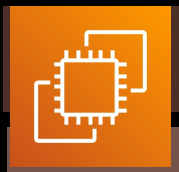
- 기존의 인스턴스 기반으로 컨테이너 동작이 아닌 서버리스 컴퓨팅 엔진으로 동작
- 설정된 리소스로 Task 실행
- OS 레벨의 컨트롤 하지 않아도 됨
- 어플리케이션 구축과 운영에만 집중 가능

여러분은 어플리케이션 구축과 운영에만 집중해요 ☺

어떤 상황에서 쓰는 것이 좋을까



비교해보기



ECS EC2

컨트롤에 용이하다.

EC2 node instance에 Task

인스턴스 유형의 제어 가능하다.

OS 수정 , 변경이 필요한 경우에 사용한다.

Spot 인스턴스, RI, Saving Plan 가능하다.

Fargate

확장성에 용이하다.

스케줄링의 자유롭다.

Batch성 작업에 좋다.

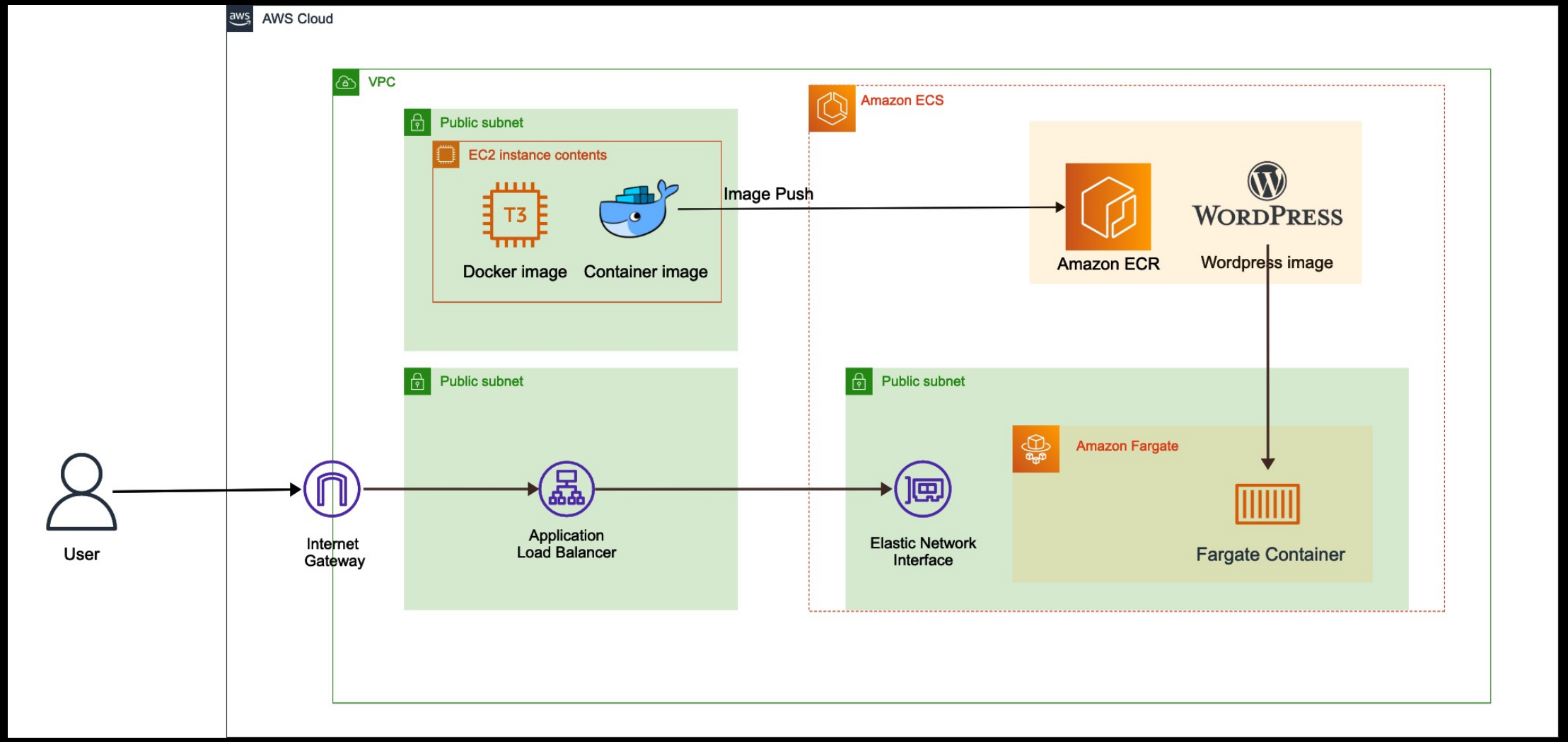
언제든지 Make ! 원하는 만큼 Use !

각각의 Container에 Task를 배치한다.

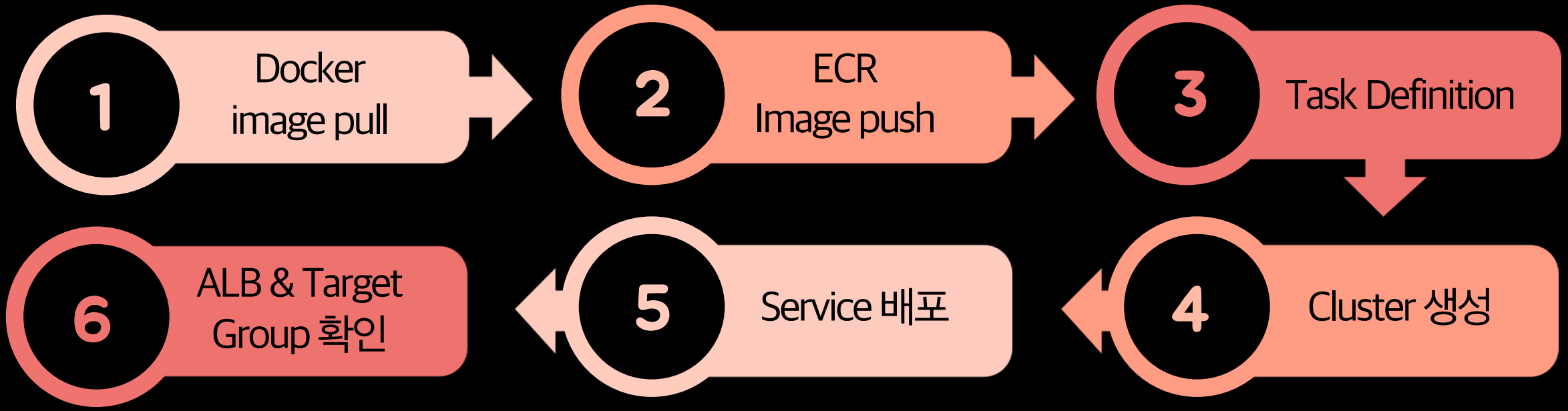
시간 절약이 필요하다면 Fargate ☺

[Demo]

Demo Architecture



Step by Step


1
Docker
image pull
2
ECR
Image push
3
Task Definition
4
Cluster 생성
5
Service 배포
6
ALB & Target
Group 확인

Why Wordpress?

다양한 플러그인

SEO 친화적

코딩 지식 없어도 됨

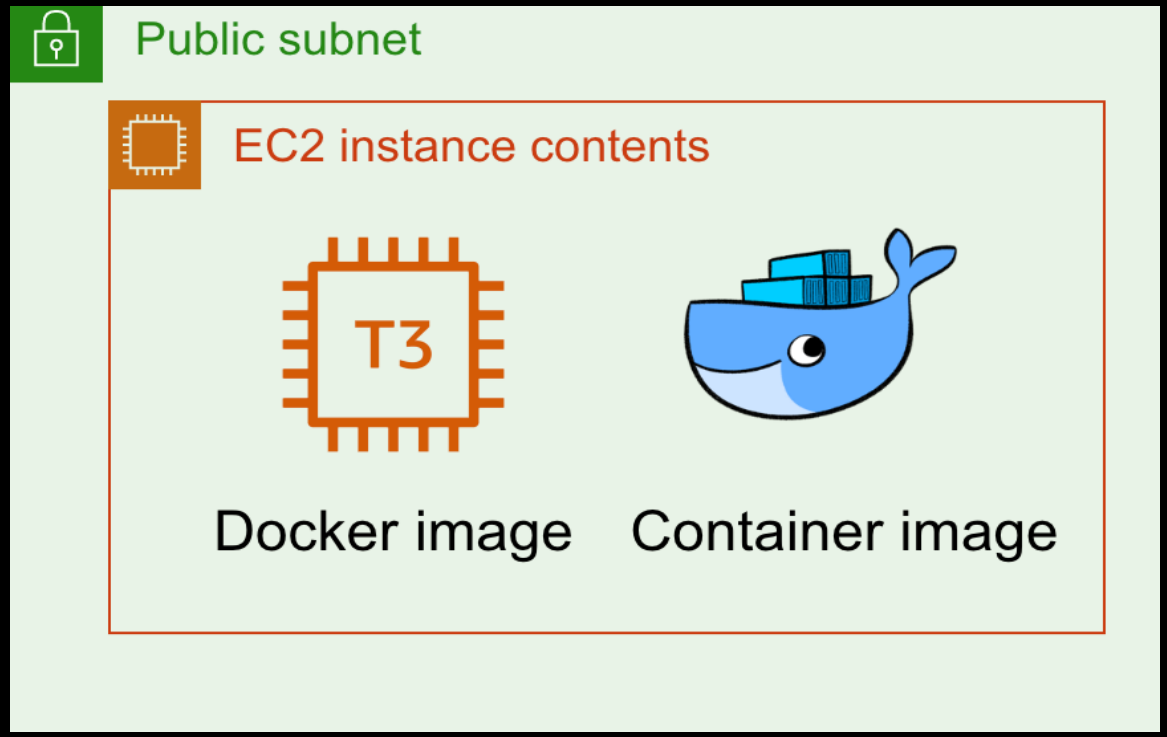
높은 자유도

쉬운 UI

[사전 작업]

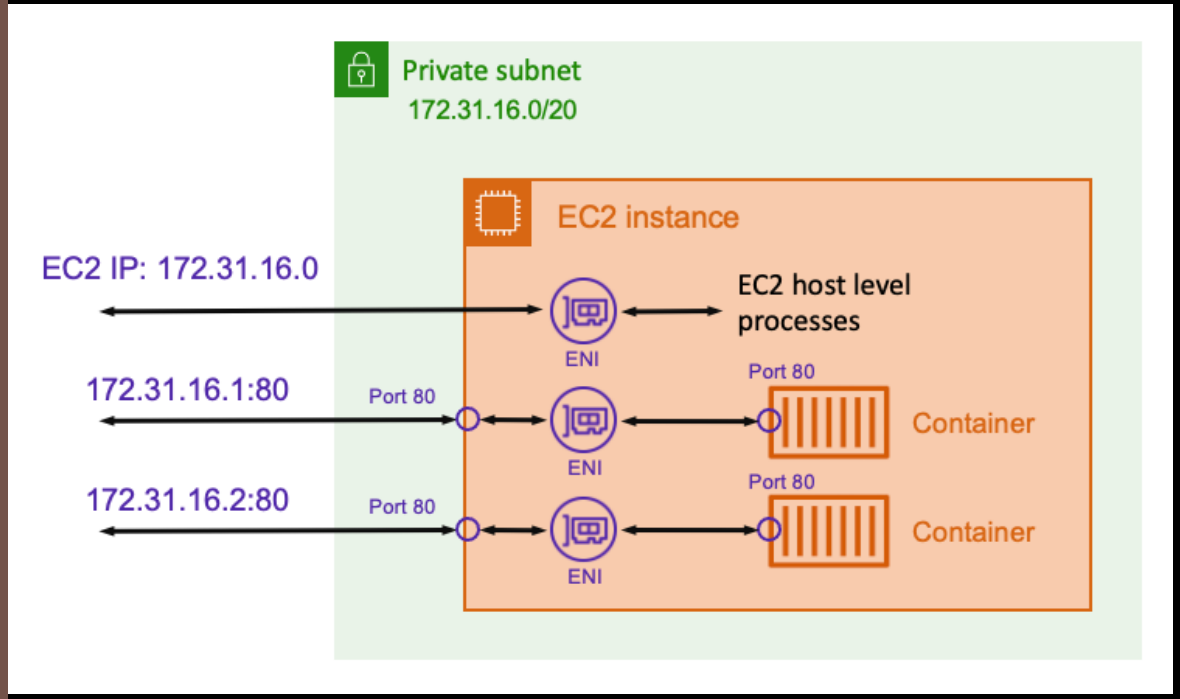
데모 시작 전에 먼저 준비

- ✓ VPC, Subnet 설정
- ✓ Docker image push 용 EC2 instance
  - ✓ AWS configure login
  - ✓ AWS CLI 2 version
  - ✓ Docker 설치



AWSVPC

- ✓ Task-ENI 연결 : 네트워크 구성이 간소화
- ✓ 포트에 제한 없음
- ✓ 리스닝 개별 관리
- ✓ 로드밸런서에 ip 대상으로 연결



1 단계
태스크 정의 및 컨테이너 구성

2 단계
환경, 스토리지, 모니터링 및 태그 구성

3 단계
검토 및 생성

태스크 정의 및 컨테이너 구성

태스크 정의 구성

태스크 정의 패밀리 정보
고유한 태스크 정의 패밀리 이름을 지정합니다.
태스크 정의 패밀리 이름 작업정의 이름
최대 255개의 문자(대문자 및 소문자), 숫자, 하이픈 및 밑줄이 허용됩니다.

컨테이너 - 1 정보

필수 컨테이너 | 제거

컨테이너 세부 정보

이름과 컨테이너 이미지를 지정하고 컨테이너를 필수로 표시할지 지정합니다. 각 태스크 정의에는 필수 컨테이너가 하나 이상 있어야 합니다.

이름: 컨테이너 이름
이미지 URI: ECR에 올린 이미지 URI
필수 컨테이너: 예

포트 매핑 정보
포트 매핑을 추가하여 컨테이너가 호스트의 포트에 액세스하여 트래픽을 송수신할 수 있도록 합니다.

컨테이너 포트	프로토콜	
80	TCP	제거

포트 매핑 더 추가 컨테이너 포트 매핑하기

▼ 환경 변수 - *선택 사항* 정보

개별적으로 추가
키 값 페어를 추가하여 환경 변수를 지정합니다.

하나의 작업정의에서 여러 개의 컨테이너 추가 가능!

2 단계

환경, 스토리지, 모니터링 및 태그 구성

3 단계

검토 및 생성

▼ 환경

태스크 정의에 대한 인프라 요구 사항을 지정합니다.

앱 환경 정보

태스크 정의에 대한 인프라를 지정합니다.

어플리케이션 환경 지정

AWS Fargate(서버리스) ✕

운영 체제/아키텍처 정보

Linux/X86_64

태스크 크기 정보

태스크를 위해 예약할 CPU 및 메모리의 양을 지정합니다.

CPU: 1 vCPU

메모리: 3 GB

할당할 메모리와 CPU 설정

▶ 컨테이너 크기 - *선택 사항* 정보

▼ 태스크 역할, 네트워크 모드 - *조건부*

태스크 역할 정보

태스크 IAM 역할은 태스크의 컨테이너가 AWS 서비스에 API 요청을 할 수 있도록 허용합니다. IAM 콘솔에서 태스크 IAM 역할을 생성할 수 있습니다.

없음

역할 설정하기(권한)

네트워크 모드 정보

태스크에 사용되는 네트워크 모드입니다. 기본적으로 AWS Fargate(서버리스)앱 환경이 선택되면 awsvpc 네트워크 모드가 사용됩니다. Amazon EC2 인스턴스 앱 환경을 선택한 경우 awsvpc 또는 브리지 네트워크 모드를 사용할 수 있습니다.

awsvpc

사이드카에 대한 CPU 및 메모리 할당
사이드카가 없는 경우 태스크 정의에 자동으로 추가하는 모니터링 및 로깅 옵션이 있습니다. AWS는 선택한 옵션에 따라 CPU 및 메모리 조정 권장 사항을 제공합니다.

AWS Fargate에서 실행 중인 태스크에는 로그 수집을 사용하는 것이 좋습니다. 로그 수집에 대해 자세히 알아보세요.

☑ 로그 수집 사용 정보
기본 구성을 사용하여 컨테이너 로그를 로깅 대상에 전송하도록 태스크를 구성합니다.

Amazon CloudWatch

Amazon CloudWatch의 요금 정보를 참조하세요.

로그 수집 사용하기

키	값 유형	값	
awslogs-group	값	/ecs/test	
awslogs-region	값	ap-northeast-1	
awslogs-stream-prefix	값	ecs	
awslogs-create-group	값	true	제거

추가

☐ 트레이스 수집 사용 정보
Amazon ECS는 애플리케이션의 트레이스를 AWS X-Ray로 라우팅하기 위해 OpenTelemetry 사이드카용 AWS Distro를 생성합니다. AWS X-Ray에 대한 요금 정보를 참조하세요.

☐ 지표 수집 사용 정보 미리 보기
Amazon ECS는 사용자 지정 컨테이너 및 애플리케이션 지표를 Amazon CloudWatch 또는 Amazon Managed Service for Prometheus로 라우팅하는 OpenTelemetry 사이드카용 AWS Distro를 생성합니다.

기존 클러스터

기존 클러스터를 선택합니다. 새 클러스터를 생성하려면 클러스터 페이지로 이동합니다.

test0414

▶ 컴퓨팅 구성 (고급)

배포 구성

작업정의 > 서비스배포

애플리케이션 유형 정보

실행할 애플리케이션 유형을 지정합니다.

- 서비스
  중지 및 다시 시작할 수 있는 장기 실행 컴퓨팅 작업(예: 웹 애플리케이션)을 처리하는 태스크 그룹을 시작합니다.
- 태스크
  실행하고 종료하는 독립 실행형 태스크(예: 배치 작업)를 시작합니다.

태스크 정의

기존 태스크 정의를 선택합니다. 새 태스크 정의를 생성하려면 태스크 정의 페이지로 이동합니다.

☐ 수동으로 개정 지정하기

선택한 작업 정의 패밀리의 가장 최근 개정 100개 중에서 선택하는 대신 수동으로 개정을 입력합니다.

패밀리: test-taskdefinition

개정: 1

서비스 이름

이 서비스에 대한 고유한 이름을 지정합니다.

서비스 이름 지정

배포 유형 정보

서비스에 대한 배포 컨트롤러 유형을 선택합니다.

롤링 업데이트

롤링업데이트

실행 중인 최소 태스크 정보

서비스 배포 중 허용되는 실행 중인 태스크의 최소 백분율을 지정합니다.

100

실행 중인 최대 태스크 정보

서비스 배포 중 허용되는 실행 중인 태스크의 최대 백분율을 지정합니다.

200

블루/그린은
신형 UI는 아직 업데이트 중?!

▼ 로드 밸런싱 - *선택 사항*

로드 밸런서 유형 정보

서비스에서 실행 중인 태스크에서 수신 트래픽을 분산하도록 로드 밸런서를 구성합니다.

Application Load Balancer

Application Load Balancer

새 로드 밸런서를 생성할지, 아니면 기존 로드 밸런서를 선택할지 지정합니다.

- (●) 새 로드 밸런서 생성
- ( ) 기존 로드 밸런서 사용

로드밸런서 유형 지정하기

로드 밸런서 이름

로드 밸런서에 대한 고유한 이름을 지정합니다.

리스너 정보

로드 밸런서가 연결 요청을 수신 대기할 포트 및 프로토콜을 지정합니다.

이젠 여기서 만들 수있어요

포트

80

프로토콜

HTTP

대상 그룹 정보

로드 밸런서가 서비스의 태스크로 요청을 라우팅하는 데 사용할 대상 그룹을 생성합니다.

대상 그룹 이름

프로토콜

대상그룹도 만들 수 있어요

상태 확인 경로 정보

/

Default로 상태확인 경로는 /
상태검사유예기간은 0

상태 검사 유예 기간 정보

0

초

▼ 네트워킹

VPC 정보

사용할 가상 사설 클라우드를 선택합니다.

vpc-01cdb064
default-vpc | 기본값

서브넷

태스크 스케줄러가 배치 시 고려해야 하는 VPC 내 서브넷을 선택합니다.

서브넷 선택

subnet-3c91cf4b
ap-northeast-1a

subnet-0c9fe224
ap-northeast-1d

subnet-1b493f42
ap-northeast-1c | JH-EMR-TEST-Subnet

보안 그룹 정보

기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다.

기존 보안 그룹 선택

보안 그룹 정보

기존 보안 그룹을 선택하거나 새 보안 그룹을 생성합니다.

기존 보안 그룹 선택

새 보안 그룹 생성

보안그룹은 로드밸런서의 보안그룹

보안 그룹 이름

기존 보안 그룹을 선택합니다.

sg-c5b049a1
default

EC2 > 대상 그룹 > testtg001

testtg001

작업

arn:aws:elasticloadbalancing:ap-northeast-1:782621889128:targetgroup/testtg001/b125d5f041b6dc35

세부 정보

대상 유형	프로토콜 : 포트	프로토콜 버전	VPC
IP	HTTP: 80	HTTP1	vpc-0271e78d6398e3017
IP 주소 유형	**로드 밸런서**		
IPv4	testalb001		

대상 합계	정상	비정상	사용되지 않음	초기	Draining
1	1	0	0	0	0

대상 | 모니터링 | 상태 검사 | 속성 | 태그

Wordpress 이슈?! 꿀팁

상태 검사 설정

편집

프로토콜	경로	포트	정상 임계 값
HTTP	/	트래픽 포트	5 연속 상태 검사 성공
비정상 임계값	**제한 시간**	**간격**	**성공 코드**
2 연속 상태 검사 실패	5 초	30 초	302,200

왜 그럴까요

```
curl -IL hwiyoungtestalb-475429996.ap-northeast-1.elb.amazonaws.com
HTTP/1.1 302 Found
Date: Thu, 14 Apr 2022 07:07:10 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Server: Apache/2.4.53 (Debian)
X-Powered-By: PHP/7.4.28
Location: http://hwiyoungtestalb-475429996.ap-northeast-1.elb.amazonaws.com/wp-admin/setup-config.php

HTTP/1.1 200 OK
Date: Thu, 14 Apr 2022 07:07:11 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Server: Apache/2.4.53 (Debian)
X-Powered-By: PHP/7.4.28
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
```

Redirect

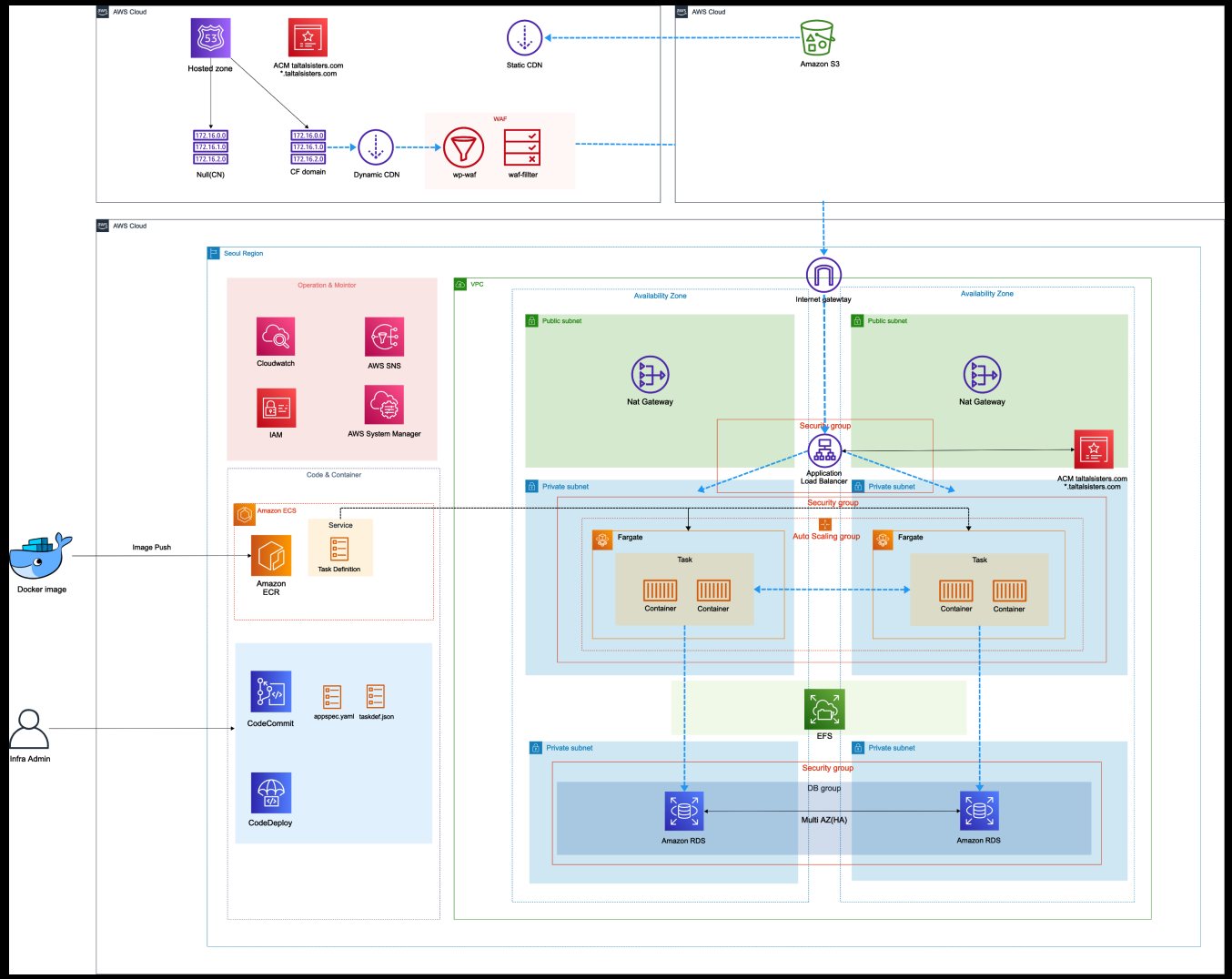
Responsecodemismatch >> Unhealthy

HTTP Request - Response 200

Architect Expansion

더 잘 써 보고 싶어!

- ✓ RDS - Database
- ✓ Fargate - 예약된 작업
- ✓ 자동화 - CI/CD
- ✓ Caching - Cloudfront(CDN)
- ✓ WAF, EFS, Route53 …



Summary

1. PaaS, Serverless
2. AWS Serverless
3. ECS EC2 vs Fargate
4. Wordpress 띄워보기
5. 더 디테일하게

Thank You

So much

Serverless Korea

CLOUDMATE
Managed Service Expert