


클라우드기초

AWS ECR, ECS, EC2...

최종석(jschoi@ssu.ac.kr)






 **CentOS**

[Download](#) [About](#) [Community](#) [Documentation](#) [Help](#)

Download




Upcoming EOL Dates

CentOS Stream 8 end of builds is **May 31, 2024**. CentOS Linux 7 end of life is **June 30, 2024**. Read the **information on upgrade and migration options**.

CentOS Linux

CentOS Stream

7-2009

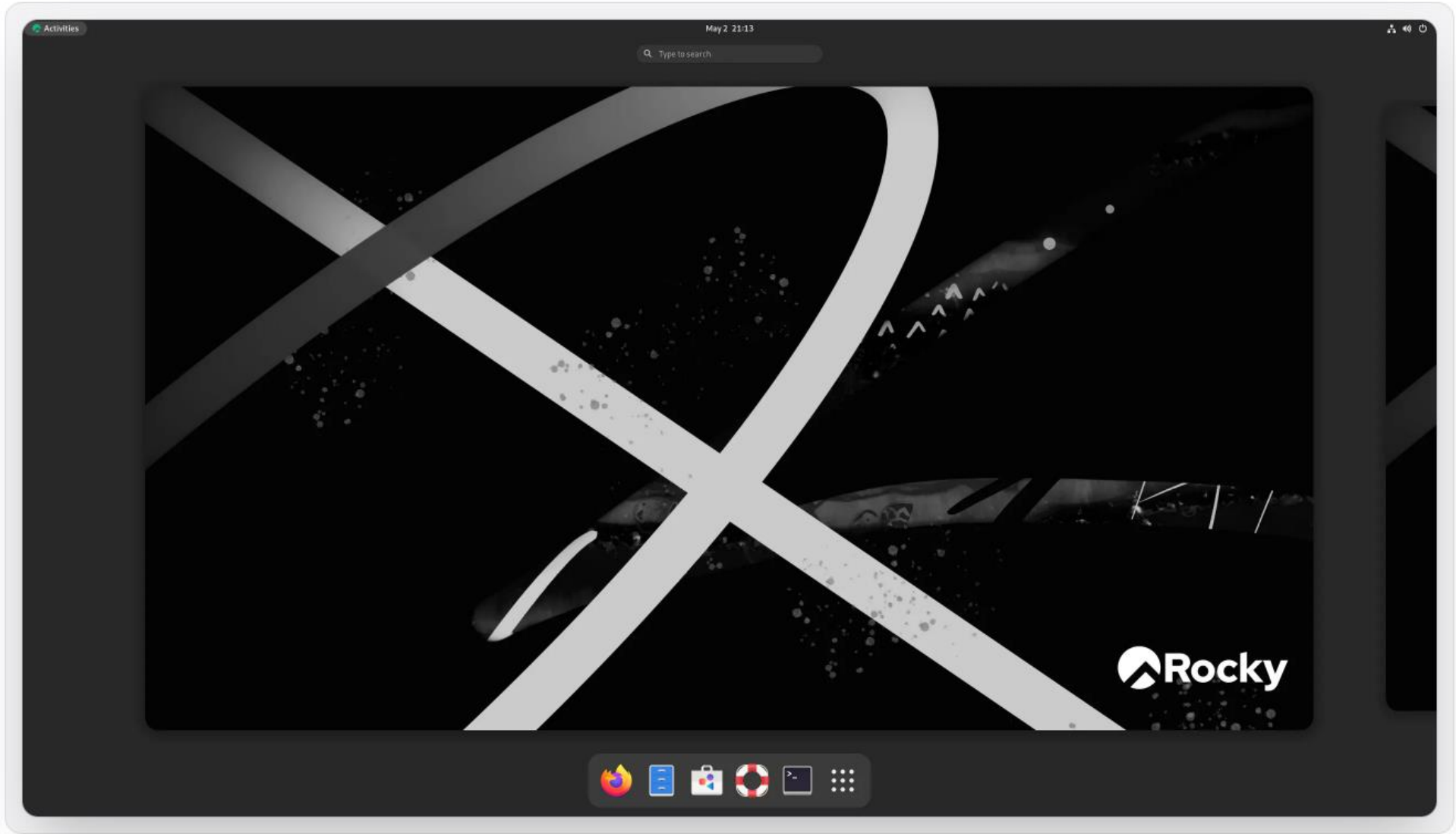
 Architectures	Packages	Others
x86_64	RPMs	Cloud Containers Vagrant

[Home](#) / [Download](#)

On this page:

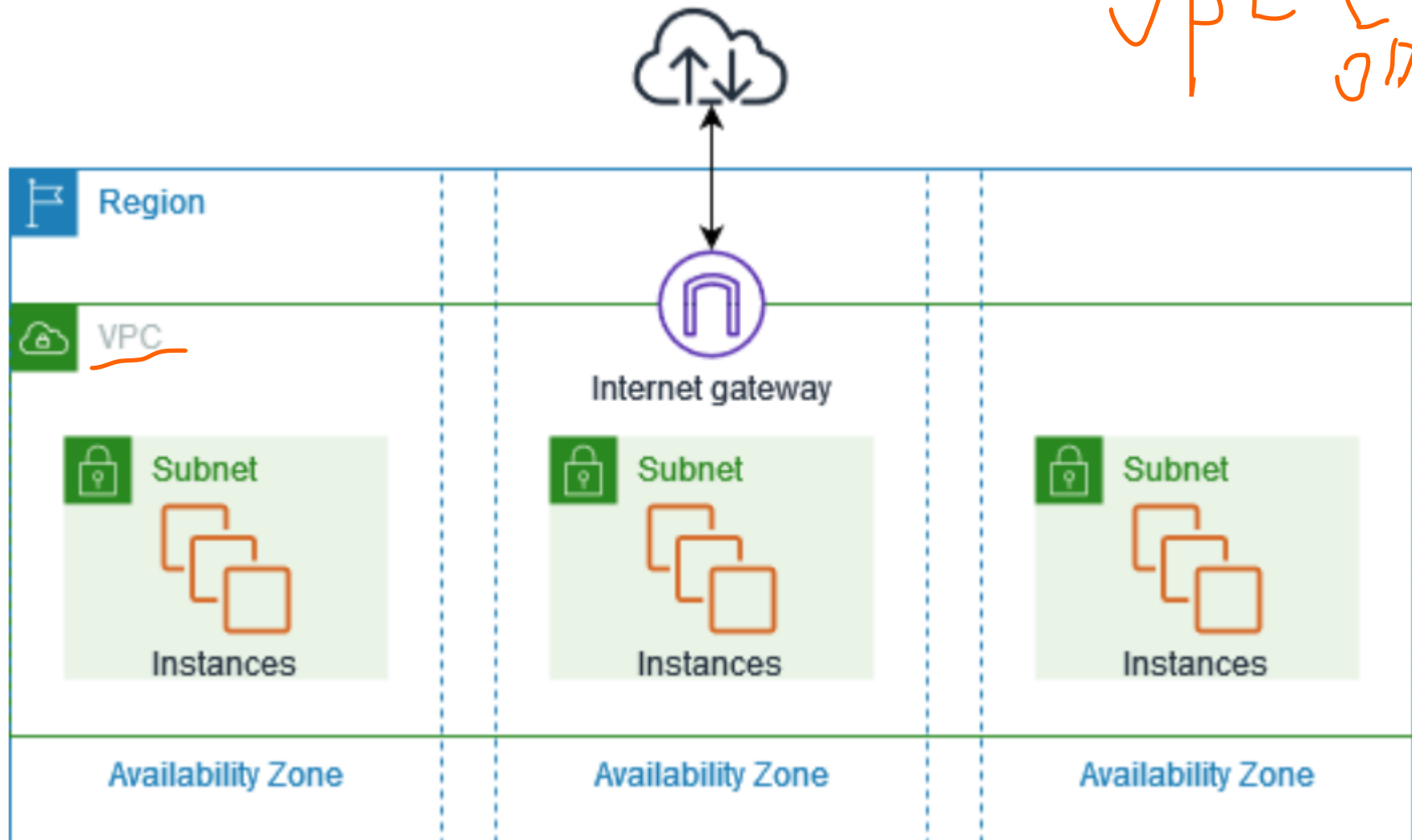
- Upcoming EOL Dates

» <https://rockylinux.org/ko>



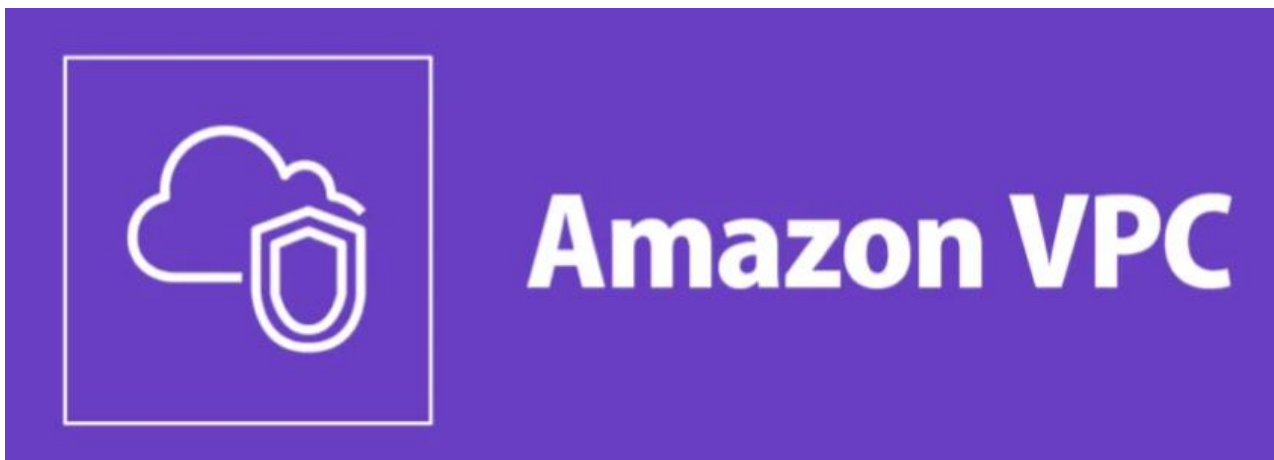
» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

VPC at
top



» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

VPC(Amazon Virtual Private Cloud)



- ✓ VPC는 자체 데이터 센터에서 운영하는 기존 네트워크와 아주 유사한 가상 네트워크
- ✓ 서브넷은 VPC의 IP 주소 범위로 서브넷을 추가한 후에는 VPC에 AWS 리소스 배포할 수 있음
- ✓ VPC에 액세스할 때 사용할 수 있는 웹 인터페이스를 제공하며, Amazon CLI를 통해서도 컨트롤 가능

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

S3(Simple Storage Service)



- ✓ 파일 서버의 역할을 하는 서비스
- ✓ 많은 사용자가 접속을 해도 이를 감당하기 위해서 시스템적인 작업을 하지 않아도 되며, 저장할 수 있는 파일 수의 제한이 없음
- ✓ 파일에 인증을 붙여서 무단으로 액세스 하지 못하도록 할 수 있으며, REST, SOAP 환경을 제공
- ✓ 편리한 복원 기능이 존재

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

S3(Simple Storage Service)

- ✓ 객체 - object, AWS는 S3에 저장된 데이터 하나를 객체라고 명명하는데, 하나의 파일
- ✓ 버킷 - bucket, 객체가 파일이라면 버킷은 연관된 객체들을 그룹핑한 최상위 디렉토리
버킷 단위로 지역(region)을 지정 할 수 있고, 또 버킷에 포함된 모든 객체에 대해서 일괄적으로 인증과 접속 제한을 걸 수 있음
- ✓ 버전관리 - S3에 저장된 객체들의 변화를 저장. 예를 들어 A라는 객체를 사용자가 삭제하거나 변경해도 각각의 변화를 모두 기록하기 때문에 실수를 만회할 수 있음
- ✓ RSS - Reduced Redundancy Storage의 약자로 일반 S3 객체에 비해서 데이터가 손실될 확률이 높은 형태의 저장 방식. 대신에 가격이 저렴하기 때문에 복원이 가능한 데이터, 이를테면 섬네일 이미지와 같은 것을 저장하는데 적합. 물리적인 하드 디스크 대비 400배 가량 안전하다는 것이 아마존의 주장
- ✓ Glacier - 영어로는 빙하라는 뜻으로 매우 저렴한 가격으로 데이터를 저장 할 수 있는 아마존의 스토리지 서비스

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate



EKS(Elastic Kubernetes Service)



Amazon EKS

- ✓ Kubernetes를 기반으로 하는 완전 관리형 서비스
- ✓ Kubernetes는 업계 표준의 컨테이너 오케스트레이션 플랫폼으로, 복잡한 컨테이너화된 애플리케이션의 배포, 관리 및 확장을 처리
- ✓ AWS가 클러스터의 관리와 유지 보수의 상당 부분을 담당하며, 사용자는 Kubernetes API와 상호작용하여 리소스를 관리

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

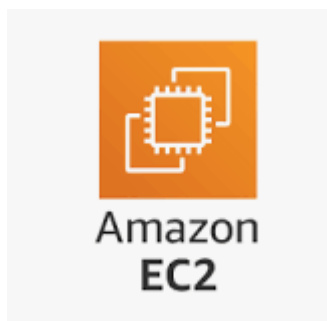
EKS(Elastic Kubernetes Service)



- ✓ EKS는 Kubernetes 클러스터를 관리하는 서비스이지만, 클러스터 내에서 컨테이너를 호스팅하기 위해 컴퓨트 리소스가 필요. EKS에서 컨테이너를 실행하려면, Amazon EC2 인스턴스 또는 AWS Fargate를 사용하여 컴퓨트 리소스를 제공
- ✓ 즉 EKS는 Kubernetes의 마스터노드(Control Plane)을 제공하는 서비스
EC2, Fargate가 Kubernetes의 워커노드(Data Plane)의 역할

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

EC2(Elastic Compute Cloud)



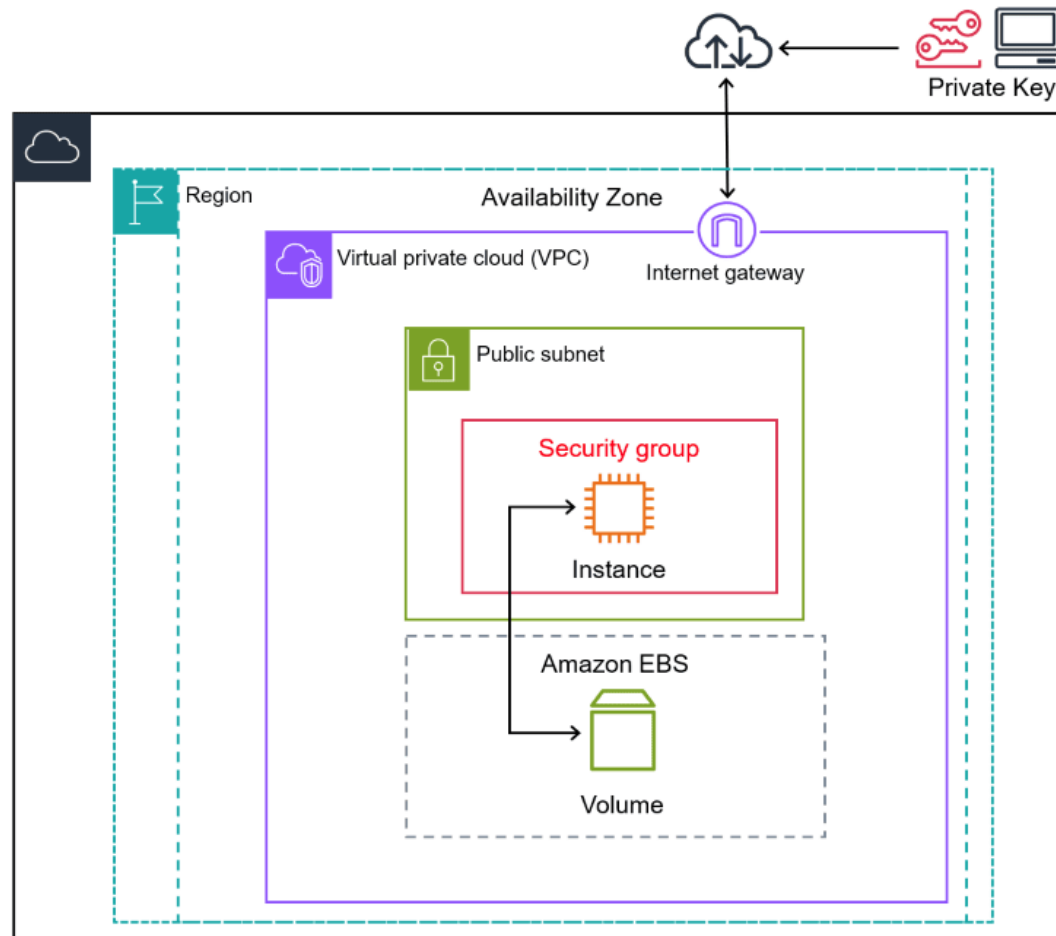
- ✓ 가상화된 컴퓨팅 환경을 제공하는 서비스로, 가상 서버를 임대해 애플리케이션을 호스팅
- ✓ 다양한 컴퓨팅 작업에 사용되며, 웹 서버, 데이터베이스, 게임 서버, 애플리케이션 서버 등 거의 모든 서버 기반 작업에 사용
- ✓ 다양한 인스턴스 유형과 구성 옵션을 제공하여, 작업에 필요한 CPU, 메모리, 스토리지 등을 선택적으로 구성

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate



EC2(Elastic Compute Cloud)

- ✓ EC2는 보안 그룹을 사용하여 인스턴스의 인바운드 및 아웃바운드 트래픽을 관리하므로 관리자가 데이터 보안을 강화하기 위한 규칙을 설정
- ✓ EC2는 Amazon S3, Amazon RDS, AWS ~~Lambda~~ 등과 같은 다양한 AWS 서비스와 원활하게 통합되므로 사용자가 포괄적이고 확장 가능한 솔루션을 구축



» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

ECR(Elastic Container Registry)

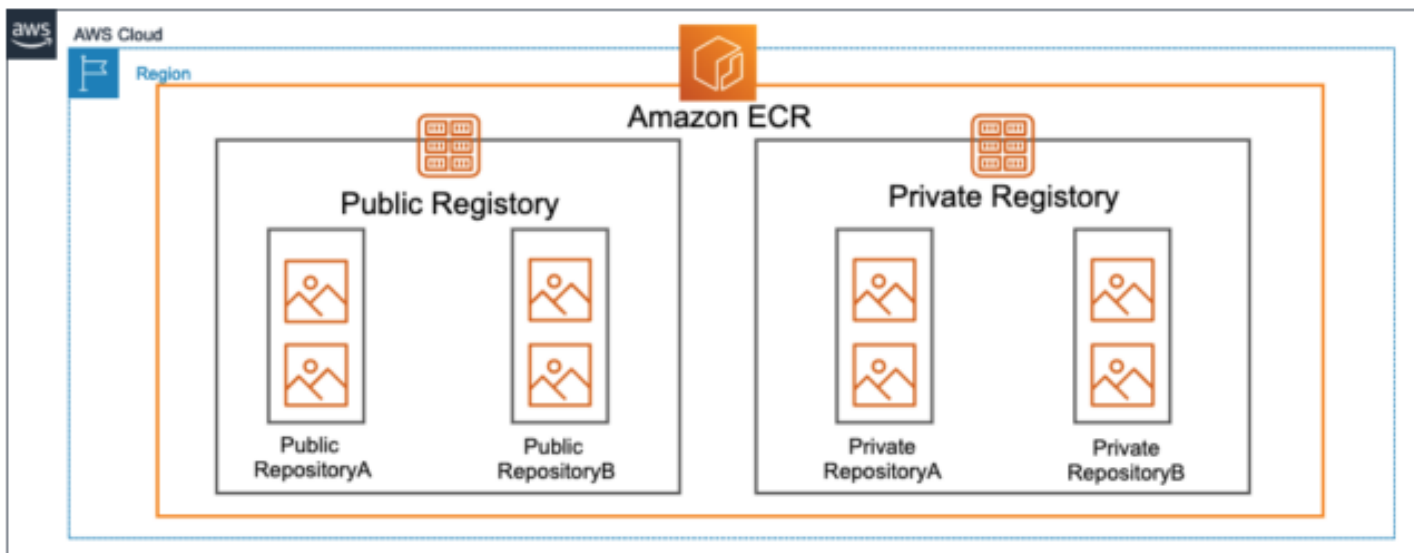


- ✓ Docker 컨테이너 이미지를 저장하고 관리하기 위한 완전 관리형 도커 컨테이너 레지스트리 서비스
- ✓ 안전하게 컨테이너 이미지를 저장하고 버전 관리를 할 수 있게 해 줌
- ✓ AWS에 통합되어 있어서 ECS 또는 EKS와 같은 서비스에서 쉽게 이미지를 가져와 사용
- ✓ AWS CLI를 통해 업로드하거나, 본인의 컴퓨터 혹은 EC2에서 빌드된 컨테이너를 업로드

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate



ECR(Elastic Container Registry)



- ✓ 퍼블릭 리파지토리 : 오픈 소스 프로젝트나 브로드캐스트 애플리케이션, 공개적으로 사용될 수 있는 어떤 이미지에 적합합니다. 누구나 사용할 수 있어야 할 이미지를 저장할 때 사용
- ✓ 프라이빗 리파지토리 : 내부적으로 사용하는 애플리케이션 또는 기업의 기밀 정보를 포함한 이미지를 저장할 때 적합합니다. 외부에 공개되지 않고 조직 내에서만 사용되어야 하는 이미지에 사용

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

ECS(Elastic Container Service)

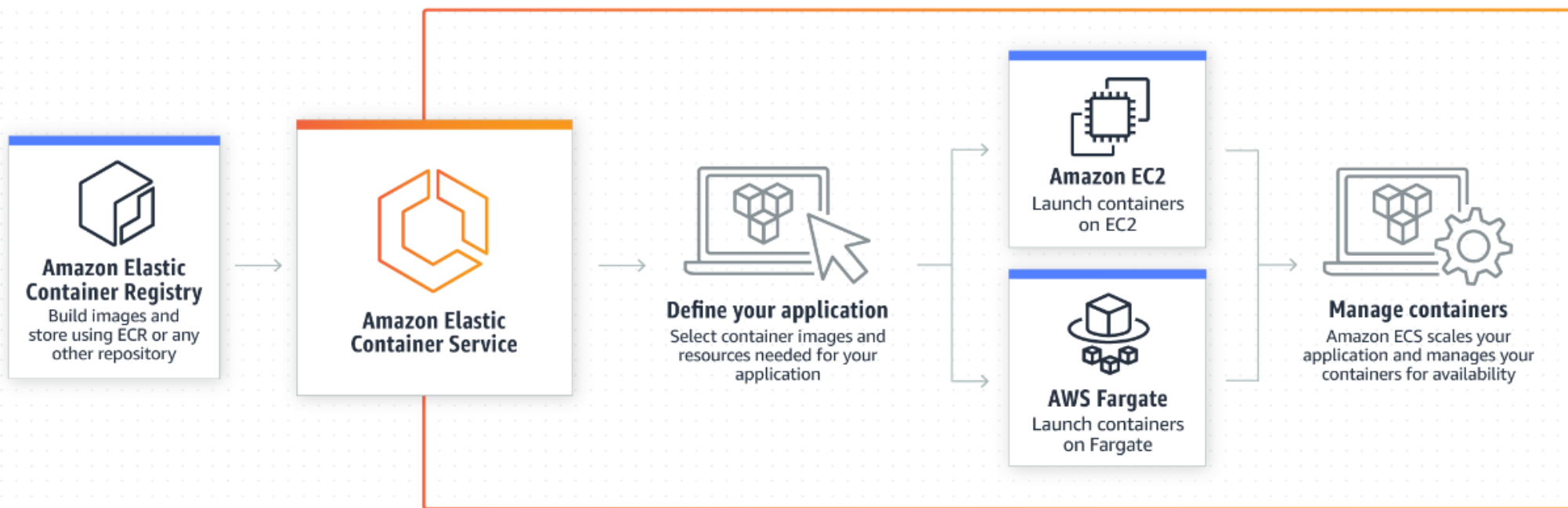


Amazon ECS

- ✓ AWS 클라우드에서 컨테이너를 쉽게 배포, 관리 및 확장할 수 있는 완전 관리형 컨테이너 오케스트레이션 서비스
- ✓ 간단한 웹 애플리케이션부터 복잡한 마이크로서비스 아키텍처까지 다양한 컨테이너화된 애플리케이션을 지원
- ✓ AWS 서비스(예: Elastic Load Balancing, Amazon VPC, AWS IAM)와 긴밀하게 통합되어 보안과 확장성을 제공

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

ECS(Elastic Container Service)



- ✓ EC2 인스턴스에서 직접 컨테이너를 실행하거나, AWS Fargate 등을 사용해 서버리스 방식으로 컨테이너를 실행할 수 있음(ECS 옵티마이즈드 AMI 필요)

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

Fargate



- ✓ 서버리스 컴퓨트 엔진으로서, 인스턴스의 선택, 관리 또는 유지 보수 없이 컨테이너를 실행
- ✓ 인프라 관리에 신경 쓸 필요 없이 컨테이너의 배포와 확장에 집중
- ✓ 사용한 컴퓨트 리소스에 대해서만 비용을 지불. 사용하지 않는 인스턴스에 대한 비용이 없으므로 전반적인 운영 비용을 절감
- ✓ 자동으로 리소스를 스케일링하여 컨테이너에 필요한 리소스를 제공. 트래픽의 증가에 따라 자동으로 컨테이너 수를 조절

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate



EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate 장/단점

✓ EKS + EC2 or Fargate

- ✓ k8s를 통한 클러스터의 운영, 확장, 업그레이드, 보안 패치 등을 자동으로 해준다는 점에서 기존 EC2 위에 직접 k8s를 올려서 사용하는 것에 비해 유리
- ✓ k8s가 지원하는 롤링 배포, auto scaling, 자동 복구 등 container orchestration 기능을 그대로 사용
- ✓ AWS에서 제공하는 다양한 기능(VPC, ELB, IAM 등) 들을 같이 활용
- ✓ 오픈 소스 생태계에서 나오는 도구에 익숙하고, 이를 잘 결합해서 운영할 노하우가 있다면 Amazon EKS를 추천하지만, 그렇지 않을 경우 사용에 어려움을 겪을 수 있음
- ✓ 많은 기능이 있는 대신, 많은 리소스와 비용이 추가됨

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate 장/단점

✓ ECR + ECS + EC2 or Fargate

- ✓ 기존의 AWS 서비스를 잘 결합해서 컨테이너 배포에 맞도록 서비스를 만들었기 때문에 손쉽게 배포 및 운영이 가능
- ✓ 여러 container orchestration 기능을 사용할 수 있음
- ✓ EKS에 비해 서비스 구조 설정이 쉽고, 클러스터 관리에 대한 추가적인 비용이 없어 가격이 저렴함
- ✓ AWS의 다른 서비스들과 연동이 편리
- ✓ EC2 인스턴스에 대한 full 컨트롤을 가질 수 있으며 세부적인 설정을 할 수 있음
- ✓ AWS에 의존적일 수 밖에 없는 형태
- ✓ EC2 인스턴스를 직접 관리해야 함(인스턴스에 대한 보안패치, 네트워크 보안 설정, 오토스케일링 그룹 설정 등)

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : 웹서비스 기반

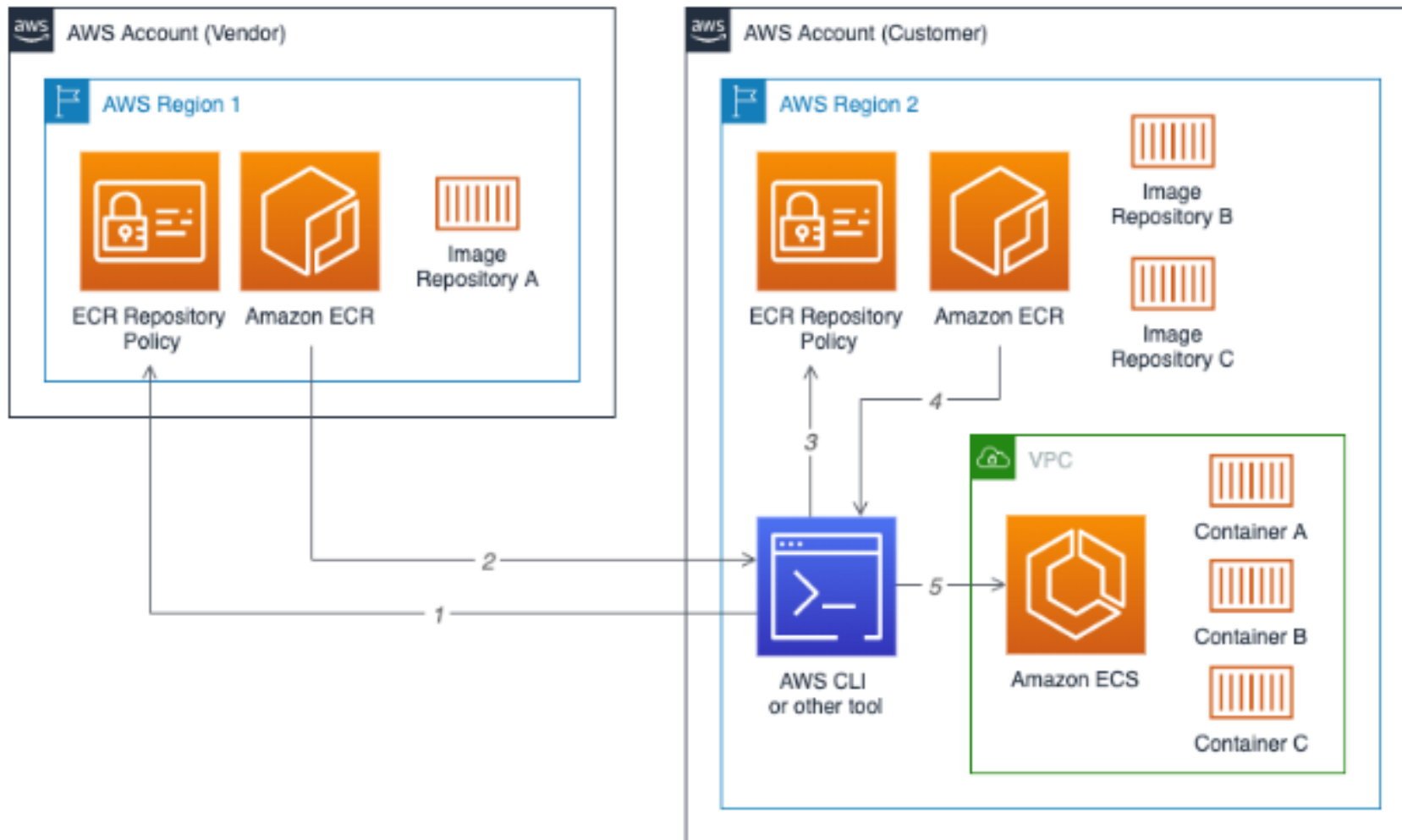
- ✓ 마스터(Control Plane) + 워커(Data Plane) 구조로 설정만 하면 어떤 것이든 가능
- ✓ ECS + EC2, ECS + Fargate, EKS + EC2, EKS + Fargate ... anything

✓ 아키텍처를 고르는 4가지 관점

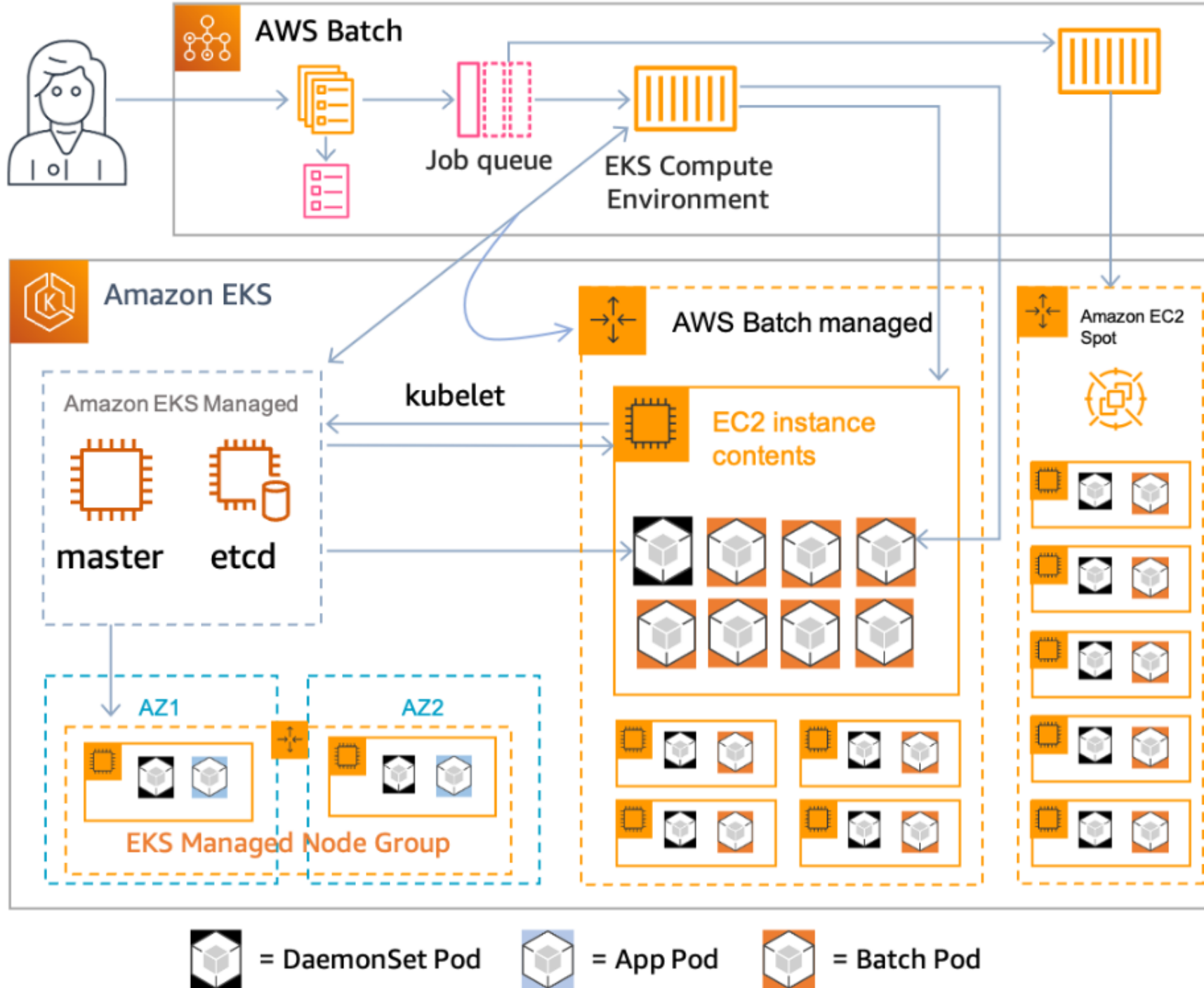
- ✓ 비용 관점
 - ✓ AWS 서비스 비용 + 운영 비용 + 인적 비용 + 학습 비용 지불 가능?
- ✓ 확장성 관점
 - ✓ 배포의 신속성, 스케일링, 자원 확장 중점이 큰지 작은지
- ✓ 신뢰성
 - ✓ 의도한 기능을 워크로드가 의도대로 수행하는 것을 보증해야 하나
 - ✓ 문제가 생겼을 때, 신속한 복구가 가능해야 하나
- ✓ 엔지니어링 관점
 - ✓ 엔지니어 확보의 용이성

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : 웹서비스 기반



01



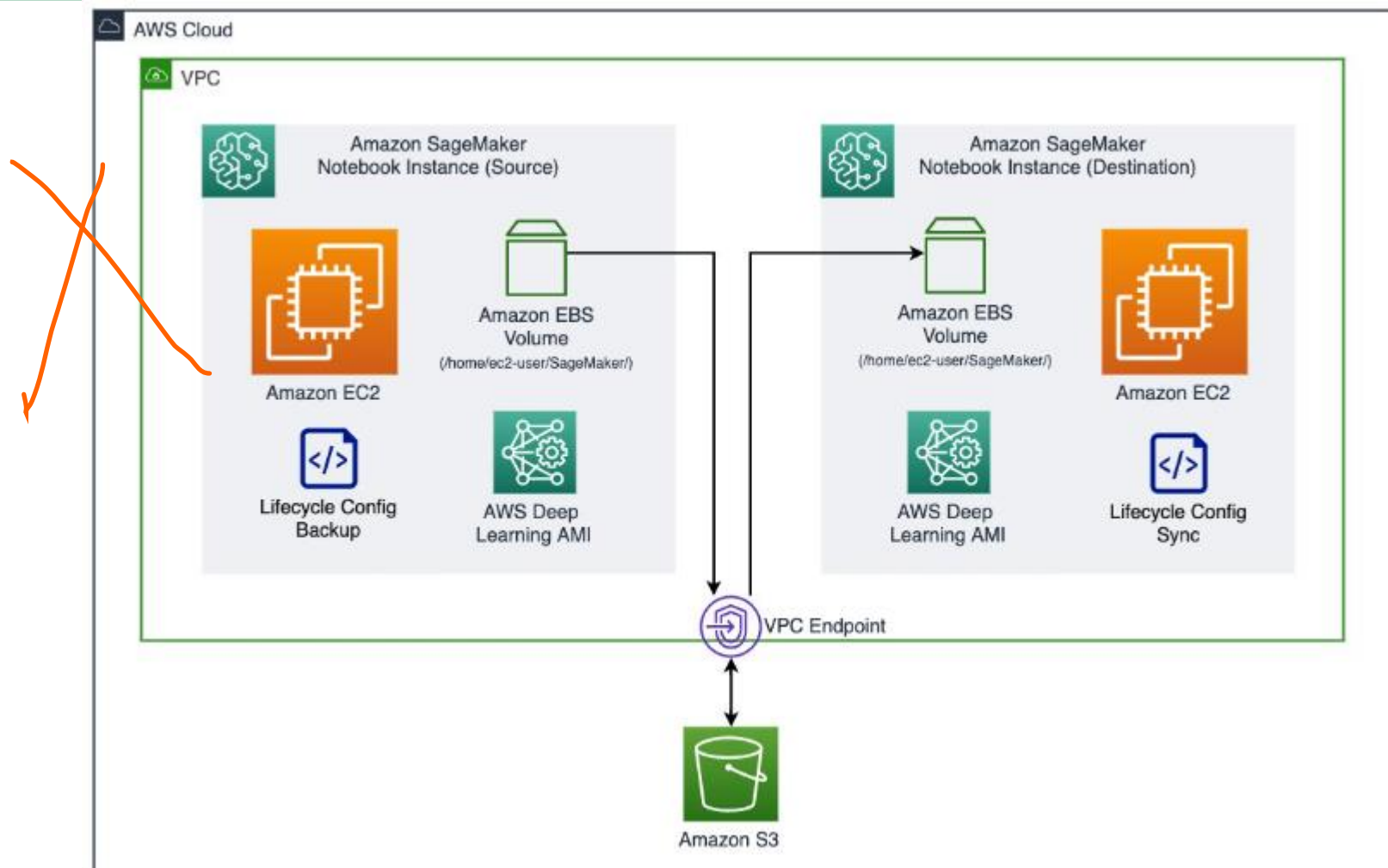
» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : AI or Machine Learning

- ✓ Base : EC2(Data Plane) , ECS or EKS(Control Plane) or SageMaker Notebook
- ✓ With : Amazon SageMaker, AWS Deep Learning AMLs, Amazon Comprehend, AWS DeepRacer, Amazon Rekognition 등
- ✓ 기계학습 시 모델 구축과 학습 부분에서 병목이 생김
- ✓ 막대한 학습이 병렬로 수행되어야함으로 특화된 GPU 혹은 기반 환경이 필요
- ✓ 특화 칩셋을 추가한 EC2 Only의 환경도 가능
- ✓ Fargate의 경우 GPU의 제약으로 인해 효율적이지 않음(CPU만 사용할 경우에 사용)

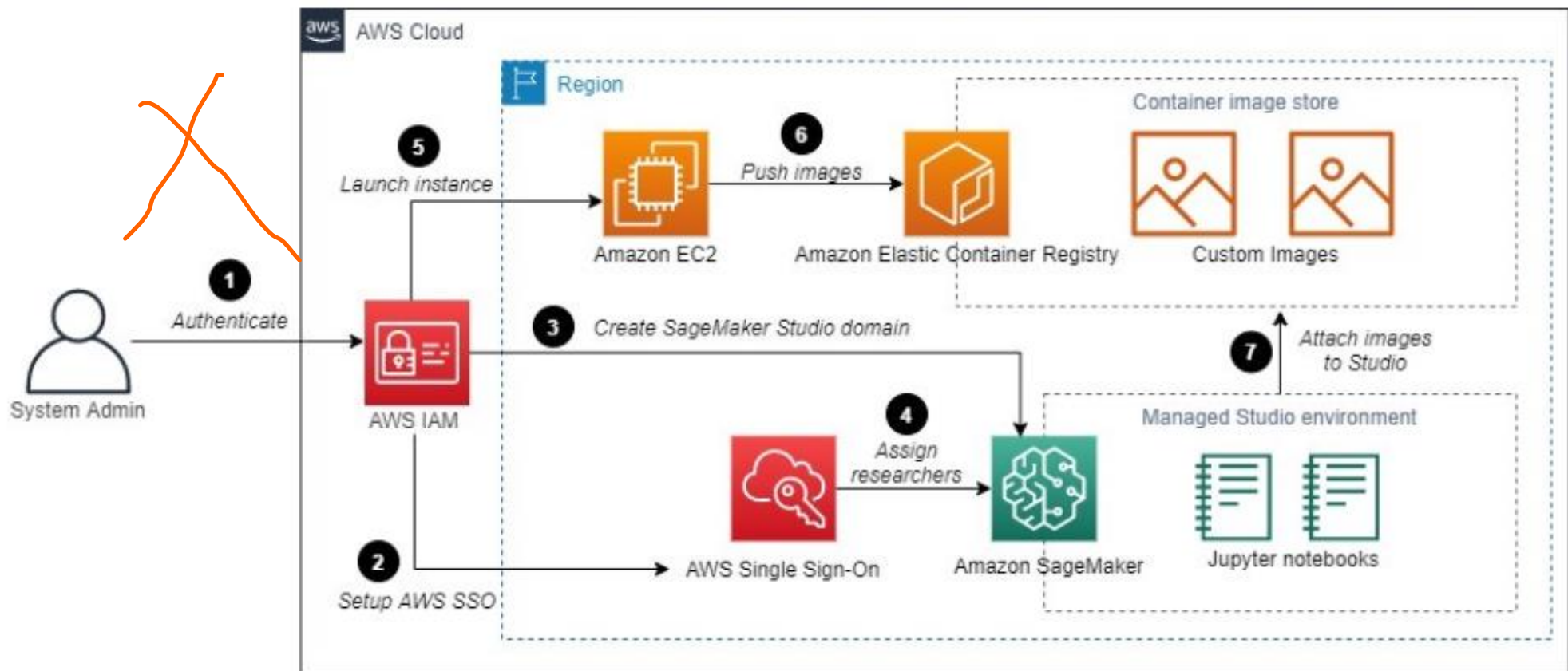
» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : AI or Machine Learning



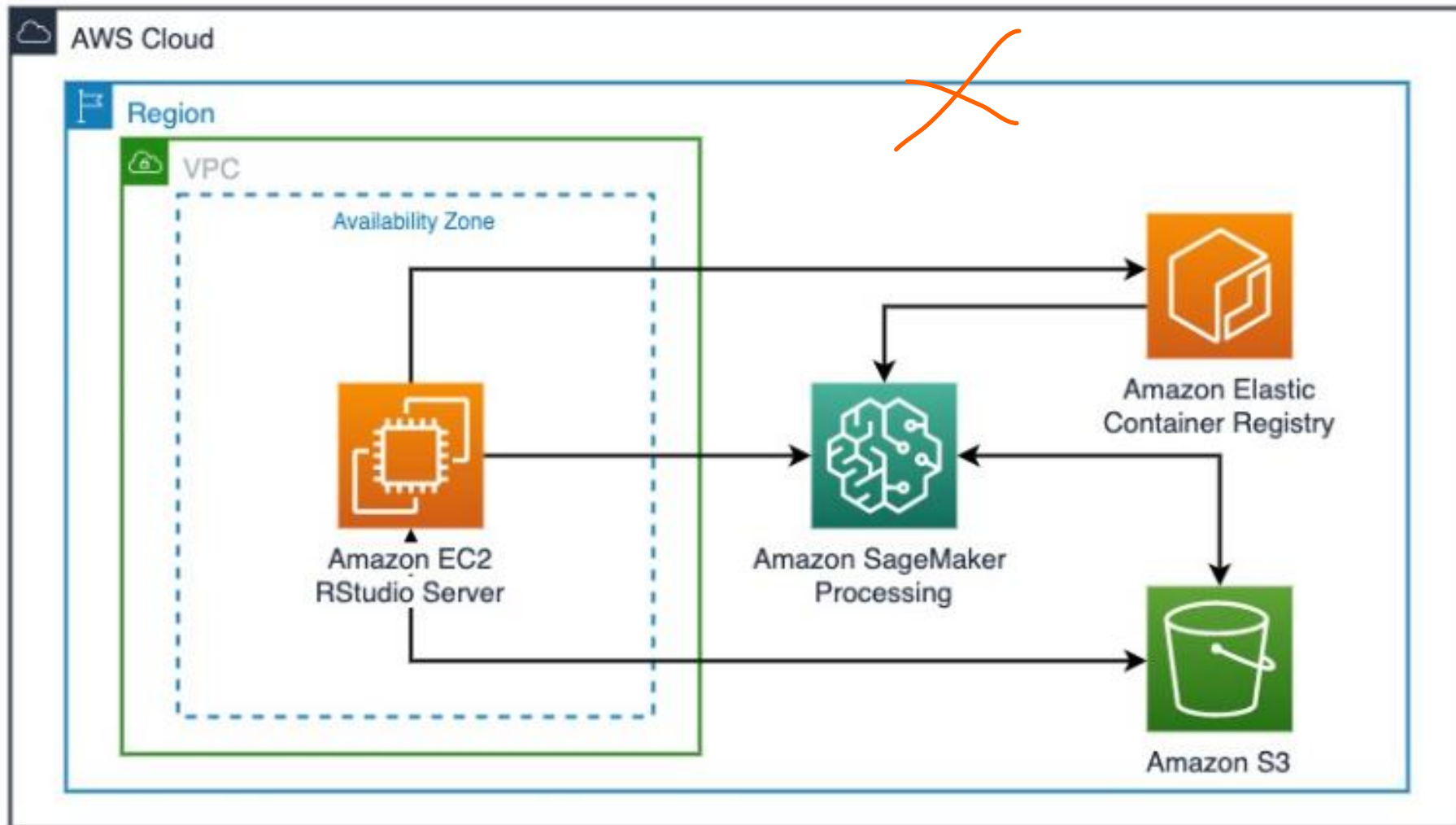
» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : AI or Machine Learning



» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : AI or Machine Learning



» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : Block Chain

- ✓ Base : EC2, Fargate(Data Plane), ECS or EKS(Control Plane)
- ✓ With : Amazon Managed Blockchain, AWS Key Management Service (KMS), Amazon CloudWatch, AWS Lambda 등
- ✓ 풀노드가 필요할 경우 모든 블록을 저장하여야 하기 때문에(수 백기가 용량) EC2
 - Fargate 보다 스토리지 사용이 편리하고 큼
- ✓ 풀노드가 필요하지 않거나, 용량이 크지 않을 경우 Fargate
- ✓ 멀티, 하이브리드 동작을 고려할시 EKS, 운영 비용을 고려한다면 ECS
- ✓ 모든 오케스트레이션을 Amazon에서 관리할 시 Managed Blockchain
- ✗ 키관리 및 내부 블록의 상태확인이 필수적임에 따라 KMS, CloudWatch 사용
- ✗ 해쉬 및 일부 작은 작업에 유리한 Lambda의 연계도 고려

» EKS + EC2 or Fargate, ECR + ECS + EC2 or Fargate

어떤 아키텍처가 가장 나에게 올바른 것인가 : Block Chain(이더리움)

