# Amazon EKS



aws

- EKS 의 주요 특징
- EKS 네트워킹
- EKS 에서의 스토리지
- EKS 에서의 보안
- EKS 에서의 자원 확장

# Amazon EKS의 주요 특징

Amazon EKS

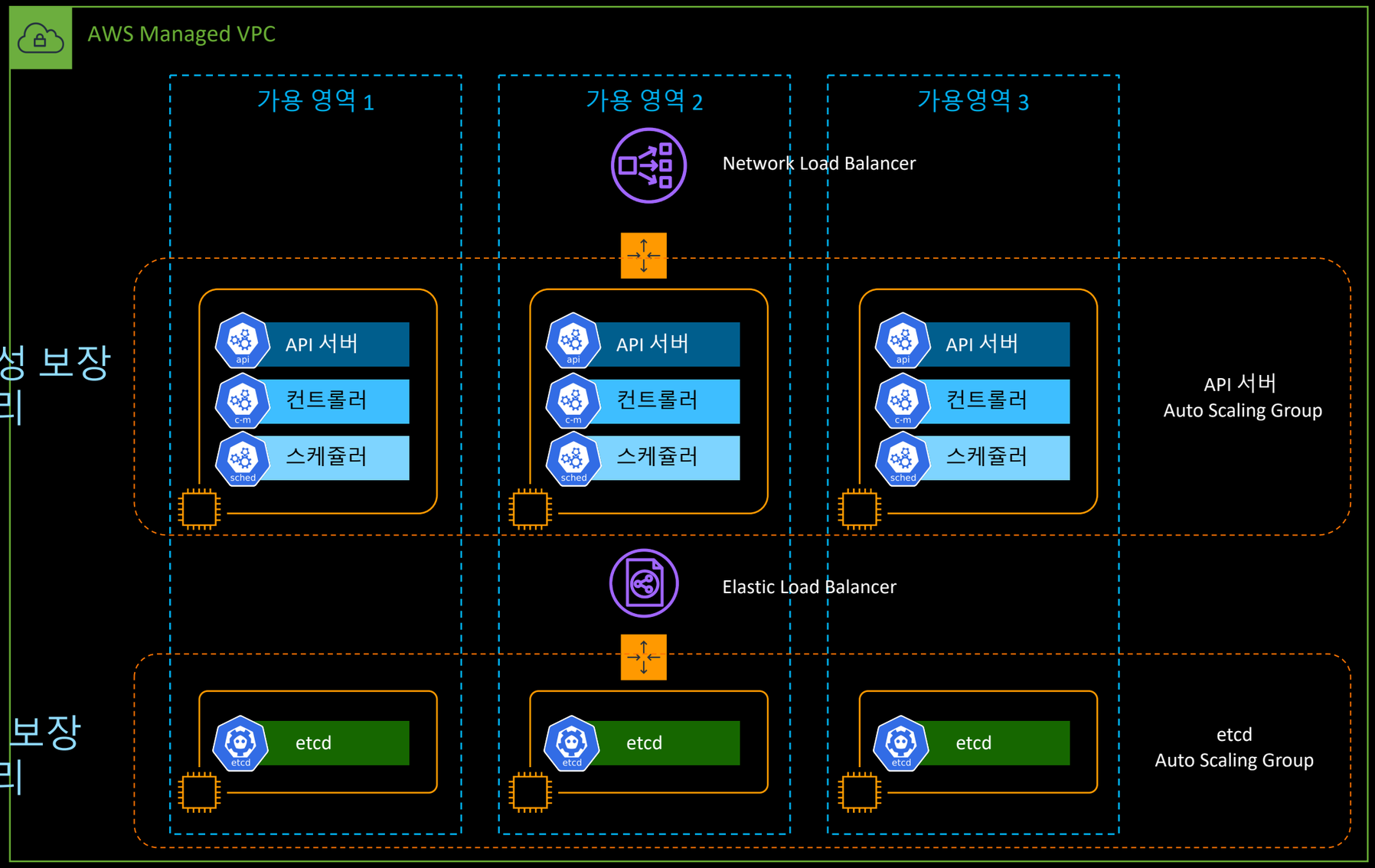
- EKS는 오픈 소스 k8s를 수정하지 않고 구동합니다.
  EKS는 k8s의 업스트림 및 인증된 k8s 버전입니다.
- EKS는 4개의 k8s 마이너 버전을 지원(k8s는 3개 버전 지원)하여 고객이 업그레이드를 테스트하고 롤아웃할 수 있는 시간을 제공합니다.
- 성능, 신뢰성 및 보안 k8s를 위한 관리형 쿠버네티스 환경을 제공합니다.
- k8s의 실행, 운영 및 관리 작업을 쉽게 단순화할 수 있도록 도와드립니다.

Amazon EKS를 통해, 고객은 모든 환경에서 안정적이고 안전한 애플리케이션을 구축할 수 있습니다.

CNCF State of Cloud Native Development

# Amazon EKS 클러스터 구성 개요

# Amazon EKS 컨트롤 플레인



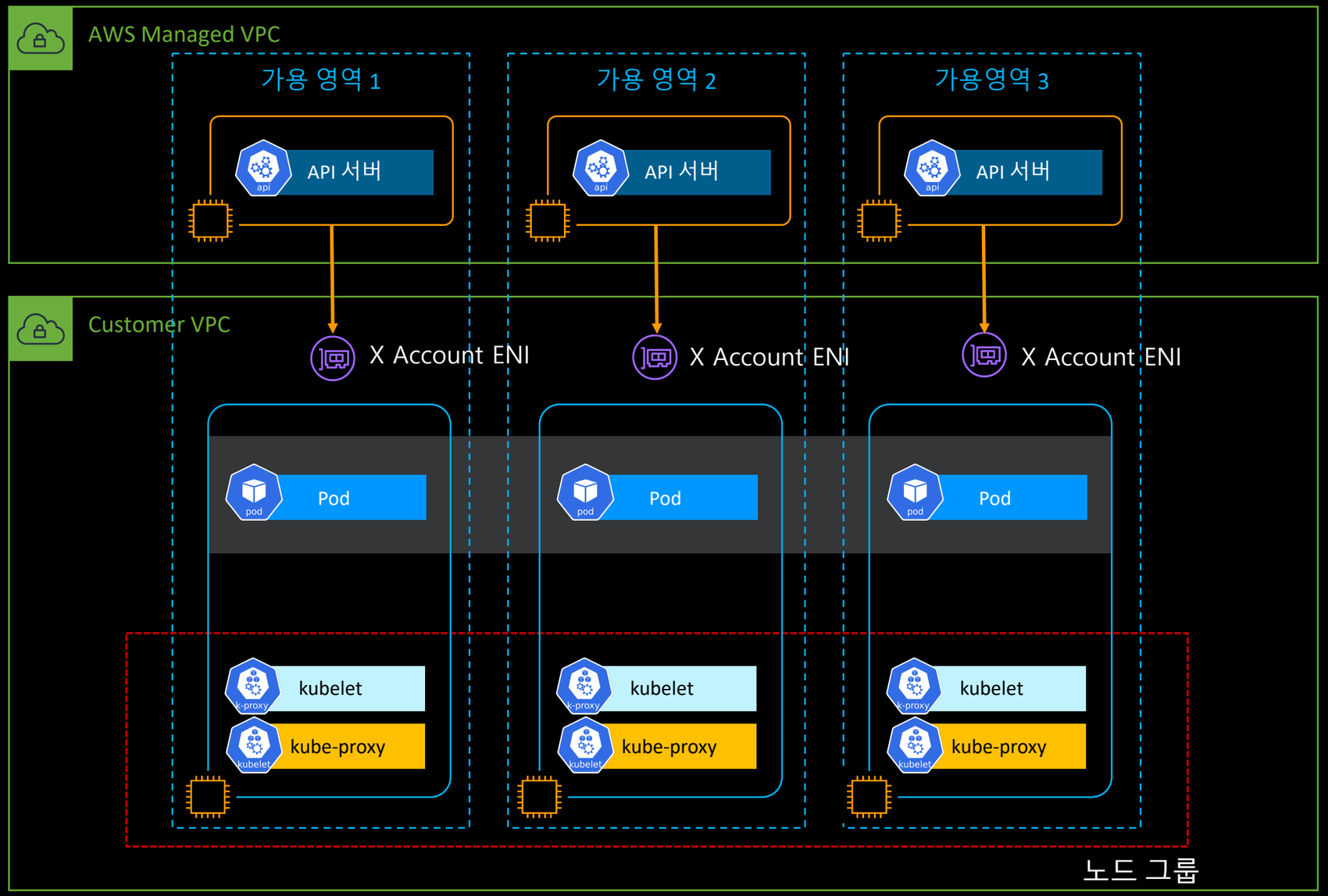

API서버 가용성 보장
백업 관리

etcd 가용성 보장
백업 관리

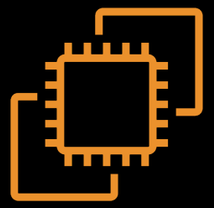
# Amazon EKS 데이터 플레인

AZ 별 교차계정 ENI 할당

관리형 노드 그룹

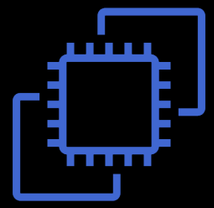


# EKS 데이터 플레인 옵션



## Self-Managed 노드 그룹

Custom AMI 를 이용하여 직접 관리하는 AutoScaling Group 을 사용. OS 에 대한 기본 구성이나 패치에 대한 책임은 고객의 책임 영역



## Managed 노드 그룹

AWS EKS 에 의해 고객의 VPC 에 프로비저닝. 최신의 EKS Optimized AMI 를 사용하며 새로운 AMI 에 대한 배포 및 구버전 AMI 제거 등을 모두 자동화하여 AWS 에서 처리

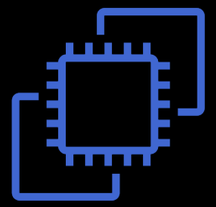
## AWS Fargate

별도로 관리하는 EC2 인스턴스 없이 Fargate 환경에서 제공하는 Micro VM 을 이용하여 Pod 별 VM 할당



aws

# Amazon EC2 실행환경: Managed Node Groups



Managed Node Groups

- Auto Scaling Group 생성 후 Cluster 와 연결
- EKS Optimized AMI, Ubuntu Optimized AMI, or Custom AMI 지원
- 인스턴스의 상태를 모니터링하고 이슈발생 시 Console 이나 CloudWatch 를 통해 정보 제공
- K8s API 를 이용하여 Termination 과정에서 Node 에 대한 Drain
- AMI 에 대한 롤링 업데이트 지원 및 Pod Disruption Budget 지원

Update Process

새로운 Launch Configuration 생성

Auto Scaling Group 설정을 1로 변경

기존 노드의 삭제 작업 진행 (Drain % Cordon)

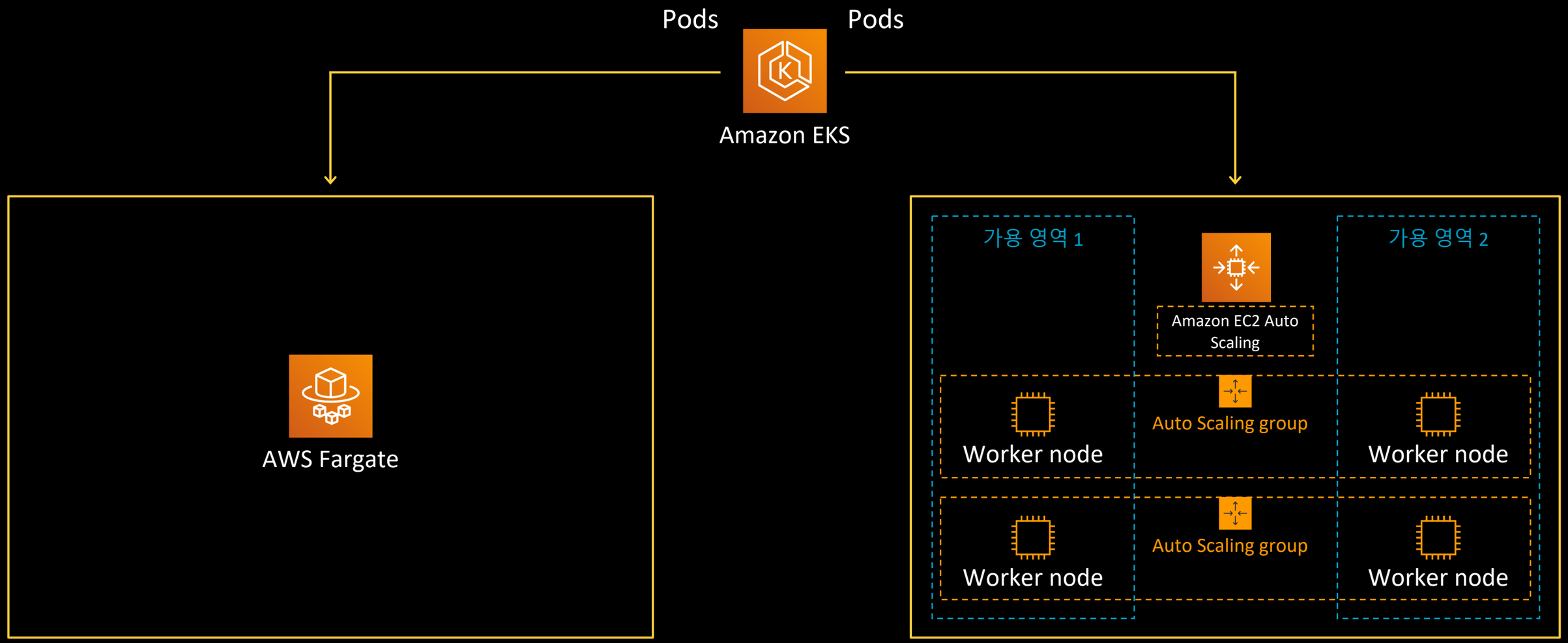
구버전 AMI 터미네이션

Auto Scaling Group 설정을 기존 설정으로 변경



aws

# Amazon EKS 실행환경 – EC2 + Fargate 혼용

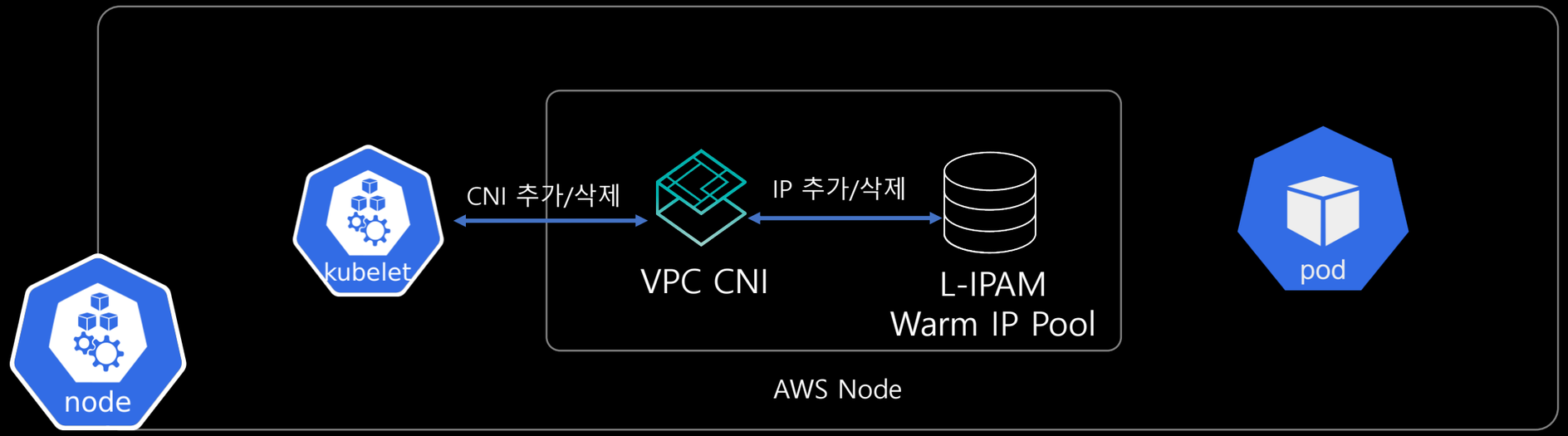



Serverless 기반 컨테이너 데이터 플레인

EC2기반 컨테이너 데이터 플레인

# EKS 네트워킹

# Amazon VPC CNI(Container Network Interface) 플러그인




CNI 플러그인을 사용하는
기본 VPC 네트워킹

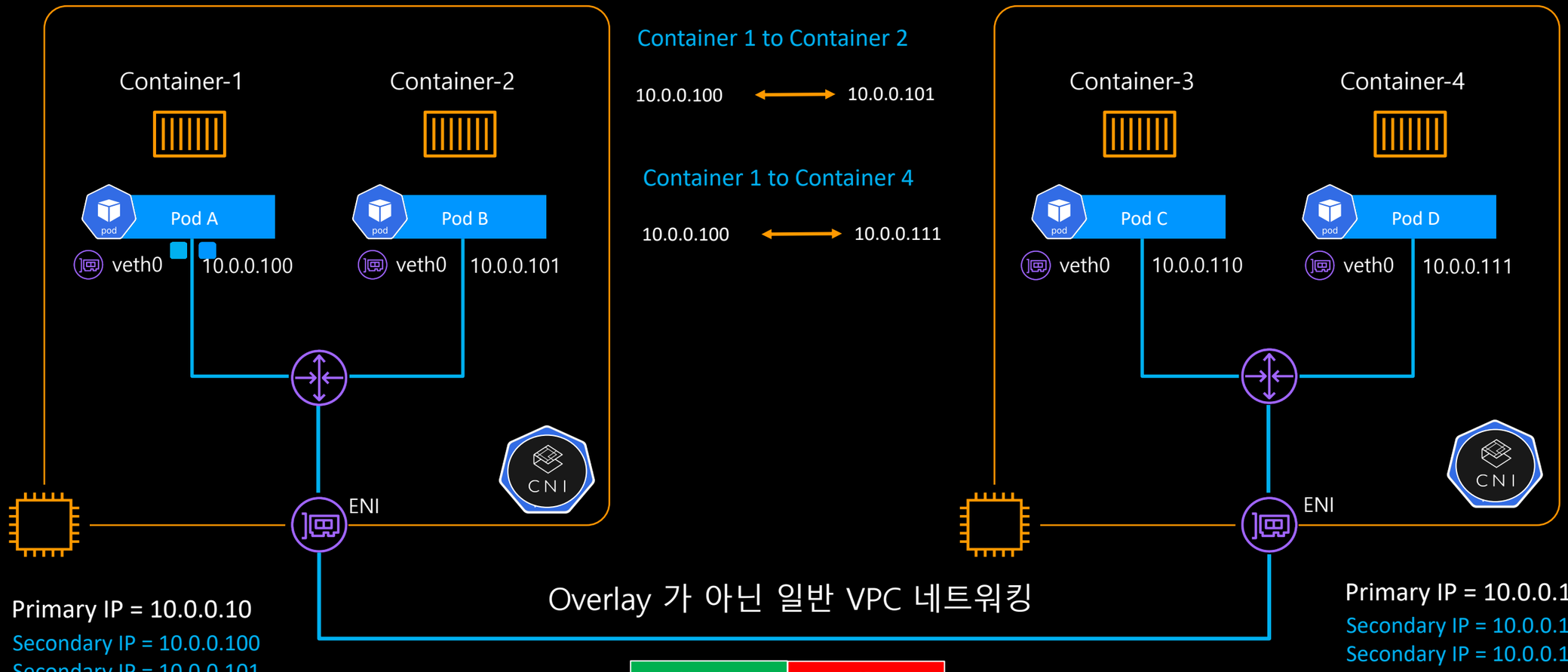
파드는 VPC와 동일한
주소를 파드가 가짐

간편하고 안전한 네트워킹

GitHub에서 유지 관리되는
오픈 소스 프로젝트

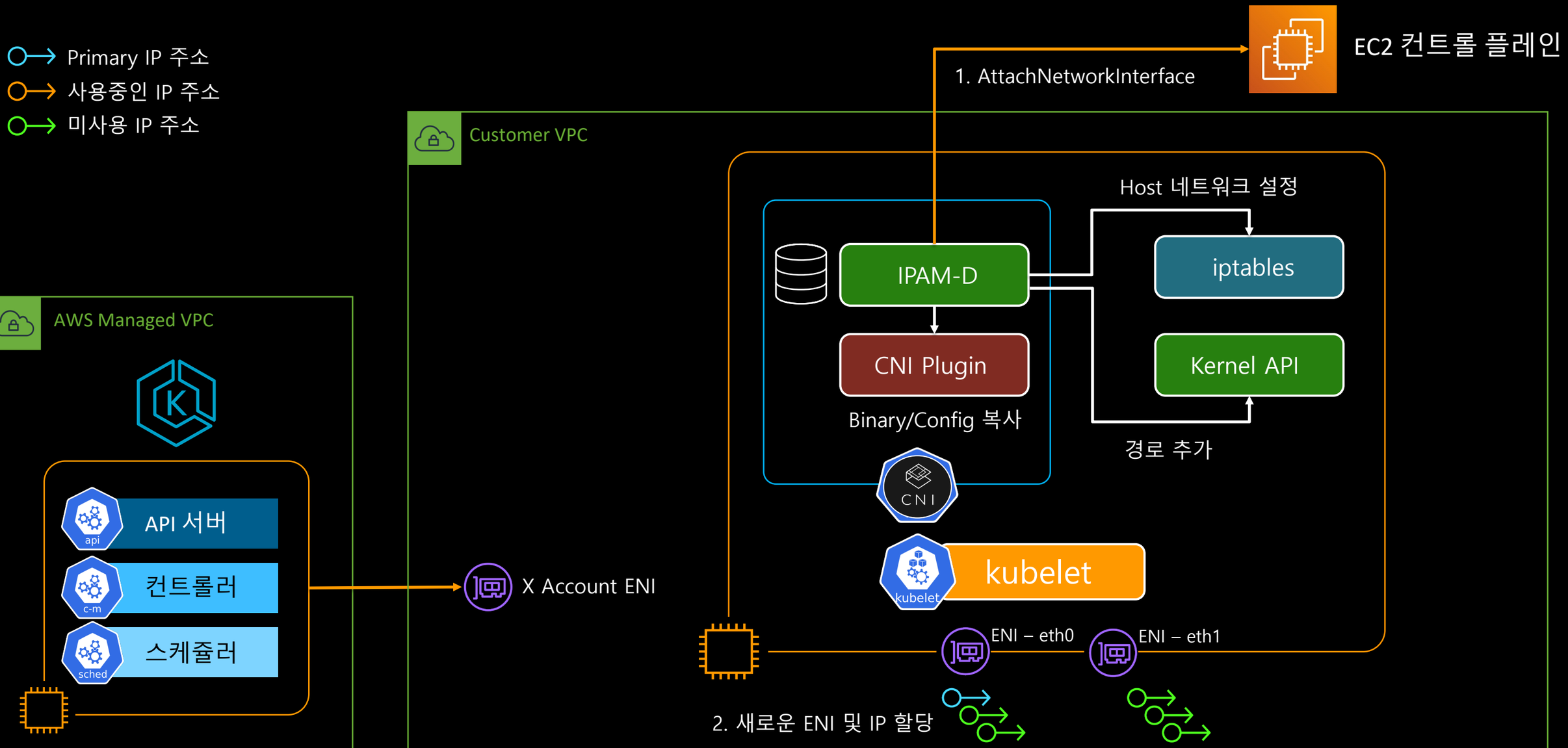
https://github.com/aws/amazon-vpc-cni-k8s

# EKS VPC CNI 네트워킹

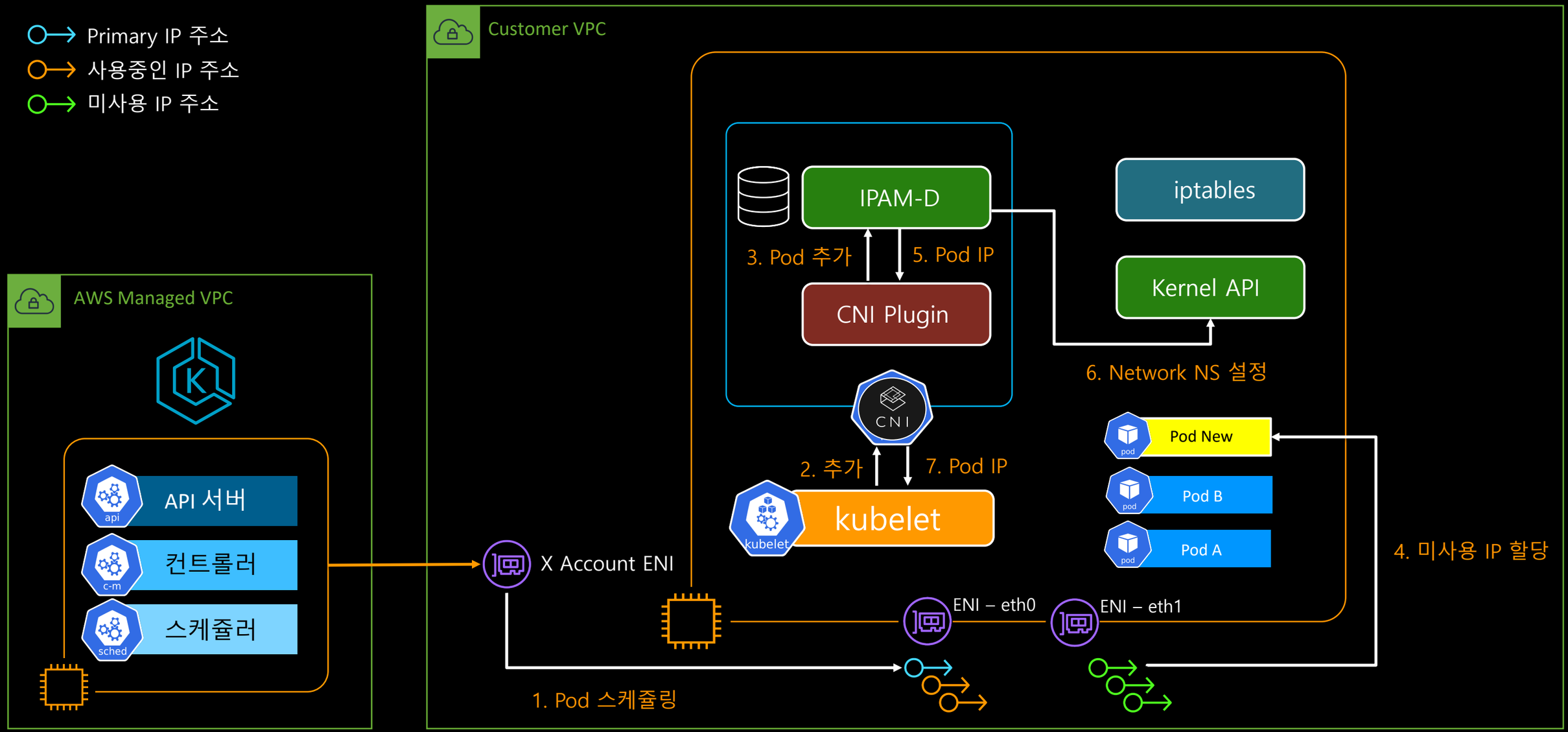



| Header | IP |
|---|---|
| Source | 10.0.0.100 |
| Destination | 10.0.0.111 |

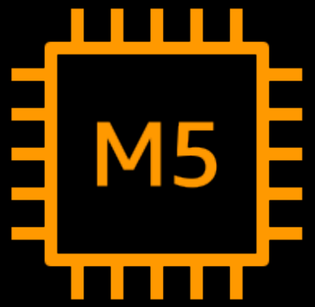
# EKS 신규 노드 추가 시 플로우



# Node 에 새로운 Pod 추가 시 플로우



# Amazon VPC CNI 플러그인을 통한 최대 파드 개수




Max Pods = ENI x (ENI 당 지원하는 IPv4 개수 - 1) + 2

예: m5.large

ENI = 3, ENI 당 지원하는 IPv4 개수 = 10

3 x (10 - 1) +2 = 29

https://github.com/awslabs/amazon-eks-ami/blob/master/files/eni-max-pods.txt

* 각 ENI의 첫 번째 IP는 파드를 위해 사용할 수 없음

* 호스트 네트워크 관련 2개 파드 포함 개수(AWS CNI와 kube-proxy)

## 노드 당 가용 IP 주소 총 량 증가 (단, Nitro system 계열 instance 만)

New Max Pods = ( ENI x (ENI 당 지원하는 IPv4 개수 - 1) ) x 16

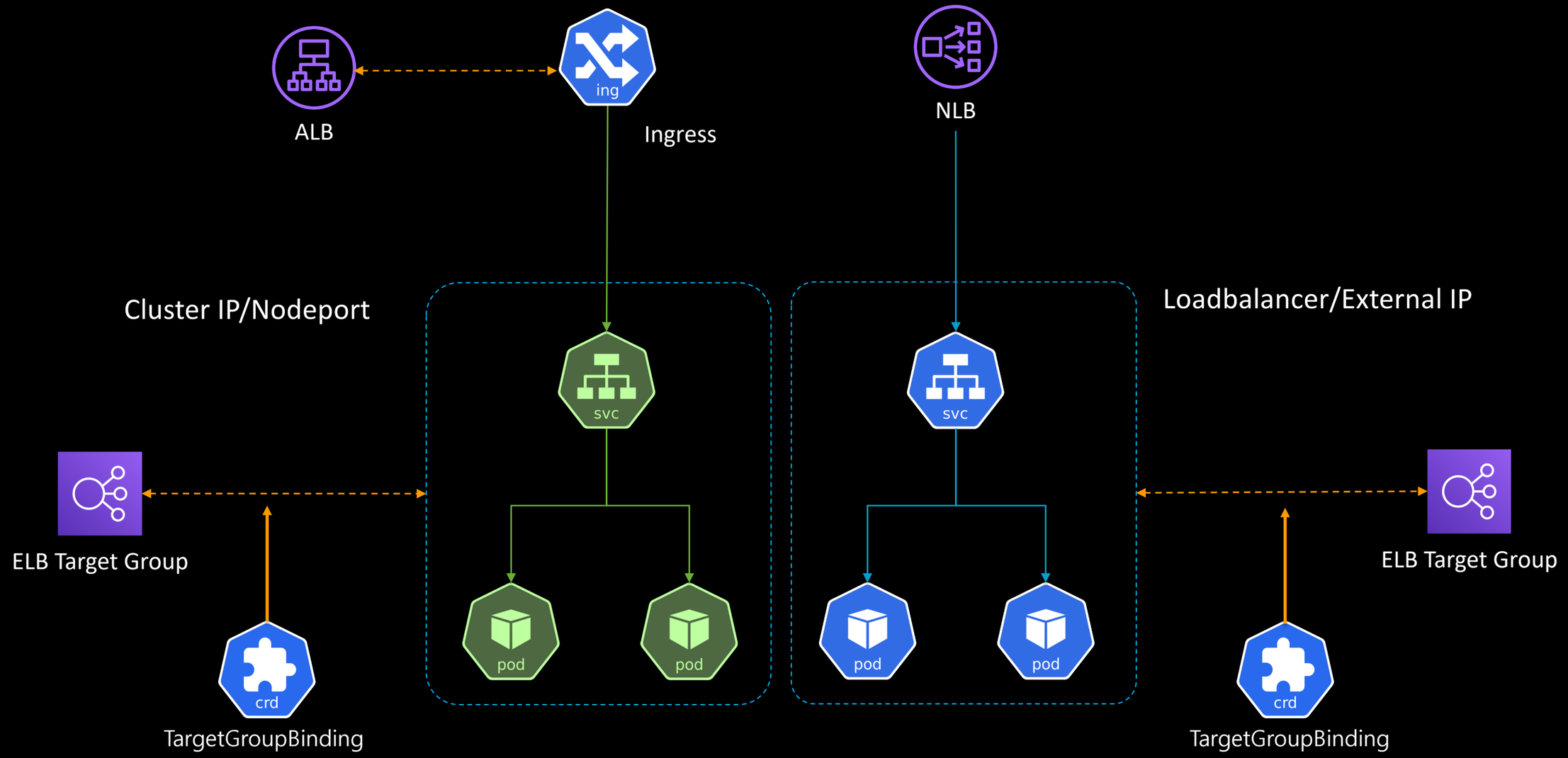
단, 30vCPU 미만 인스턴스는 110으로 제한, 이외 모든 인스턴스는 250이 최대값

https://docs.aws.amazon.com/eks/latest/userguide/choosing-instance-type.html



aws

# AWS Load Balancer Controller

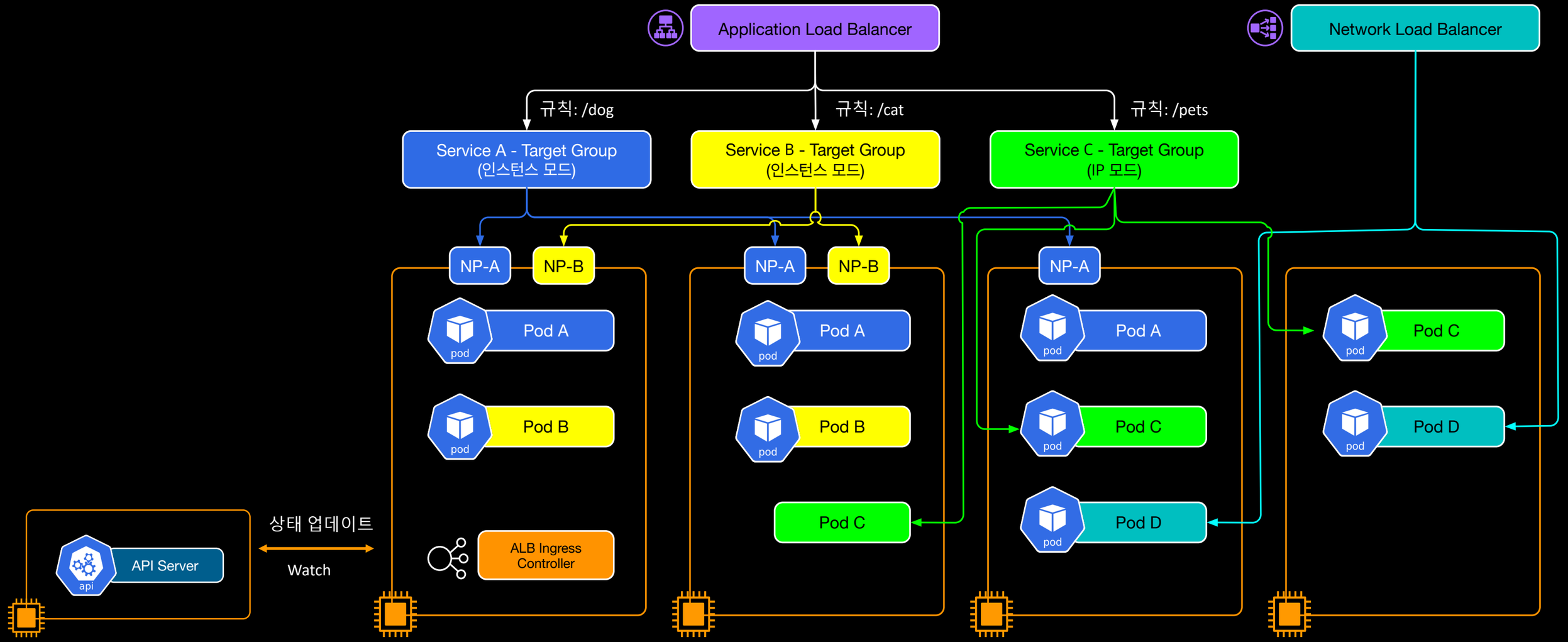



https://github.com/kubernetes-sigs/aws-load-balancer-controller



aws

# AWS Load Balancer Controller




https://github.com/kubernetes-sigs/aws-load-balancer-controller

# EKS 에서의 스토리지

# Amazon EKS에서의 영구 스토리지 사용

Container Storage Interface

Amazon EBS
CSI driver

Amazon EFS
CSI driver


Amazon FSx for Lustre
CSI driver

kubernetes-sigs/aws-ebs-csi-driver

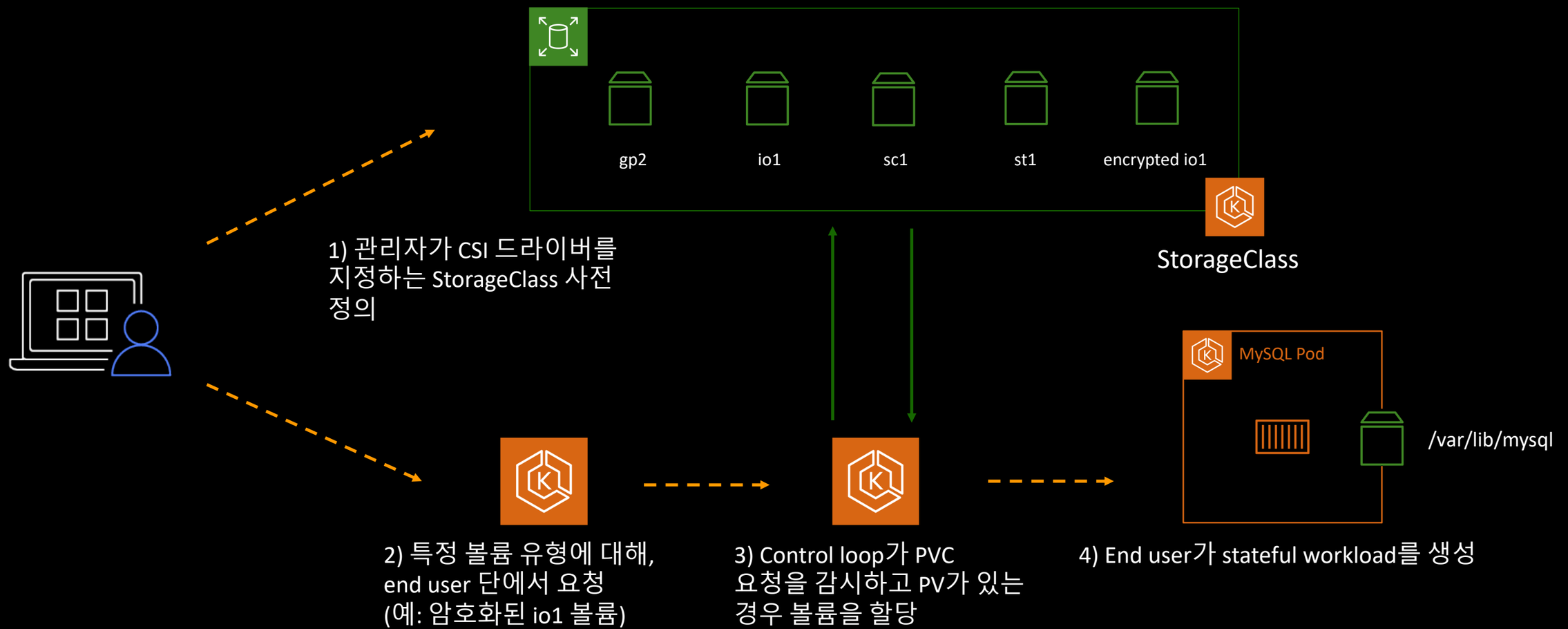
kubernetes-sigs/aws-efs-csi-driver

kubernetes-sigs/aws-fsx-csi-driver

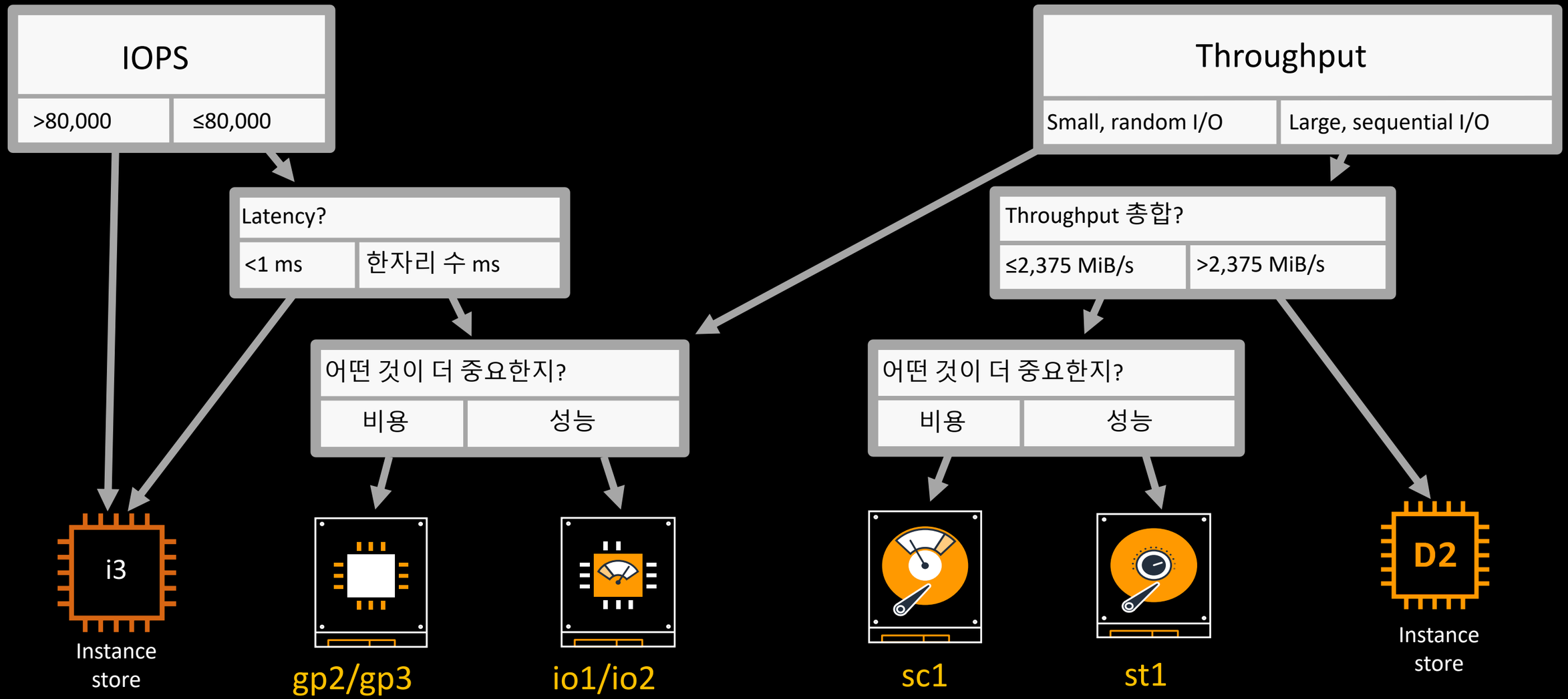


aws

# AWS Container Storage Interface (CSI) Driver

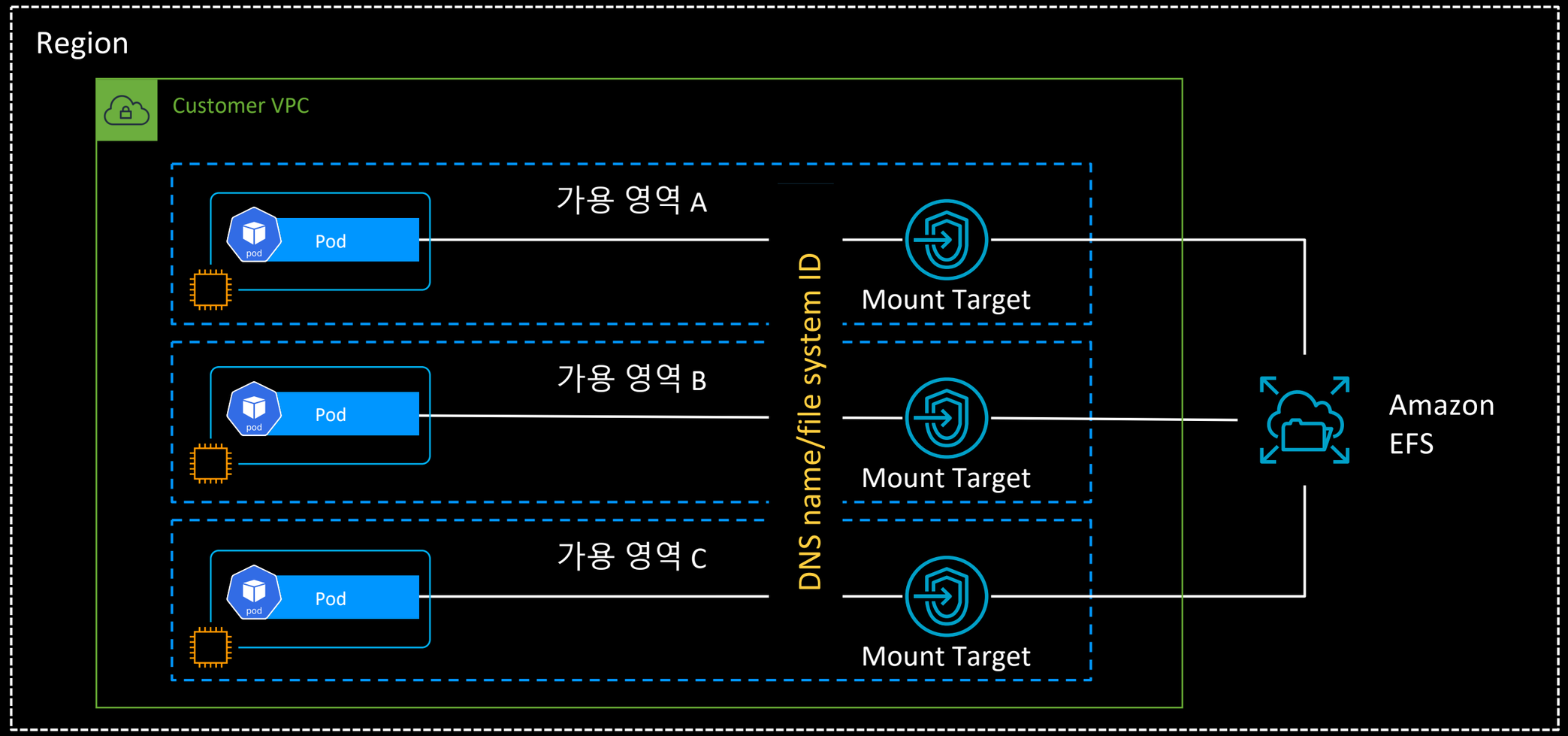


# Amazon EBS 선택 방법



# Amazon EFS with EKS

EFS는 여러 AZ에 걸쳐 데이터를 저장하여 높은 수준의 가용성과 내구성을 제공합니다.



# EKS 에서의 보안

# Amazon EKS 책임 공유 모델(Self Managed Node Group)


CUSTOMER DATA

CONTAINER IMAGES, SOURCE CODE, IAM

NETWORK POLICIES | RBAC BINDINGS | QUOTAS & LIMIT RANGES | HPA & VPA | QOS AND POD PRIORITY | POD SECURITY POLICIES | POD DISRUPTION BUDGETS | CLUSTER ADD-ONS

EKS CLUSTER CONFIGURATION

VPC Configuration | WORKER NODE SCALING

OS, KUBELET, CRI & AMI CONFIGURATION

aws

KUBERNETES CONTROL PLANE

API SERVER | CONTROLLER MANAGER | SCHEDULER

ETCD

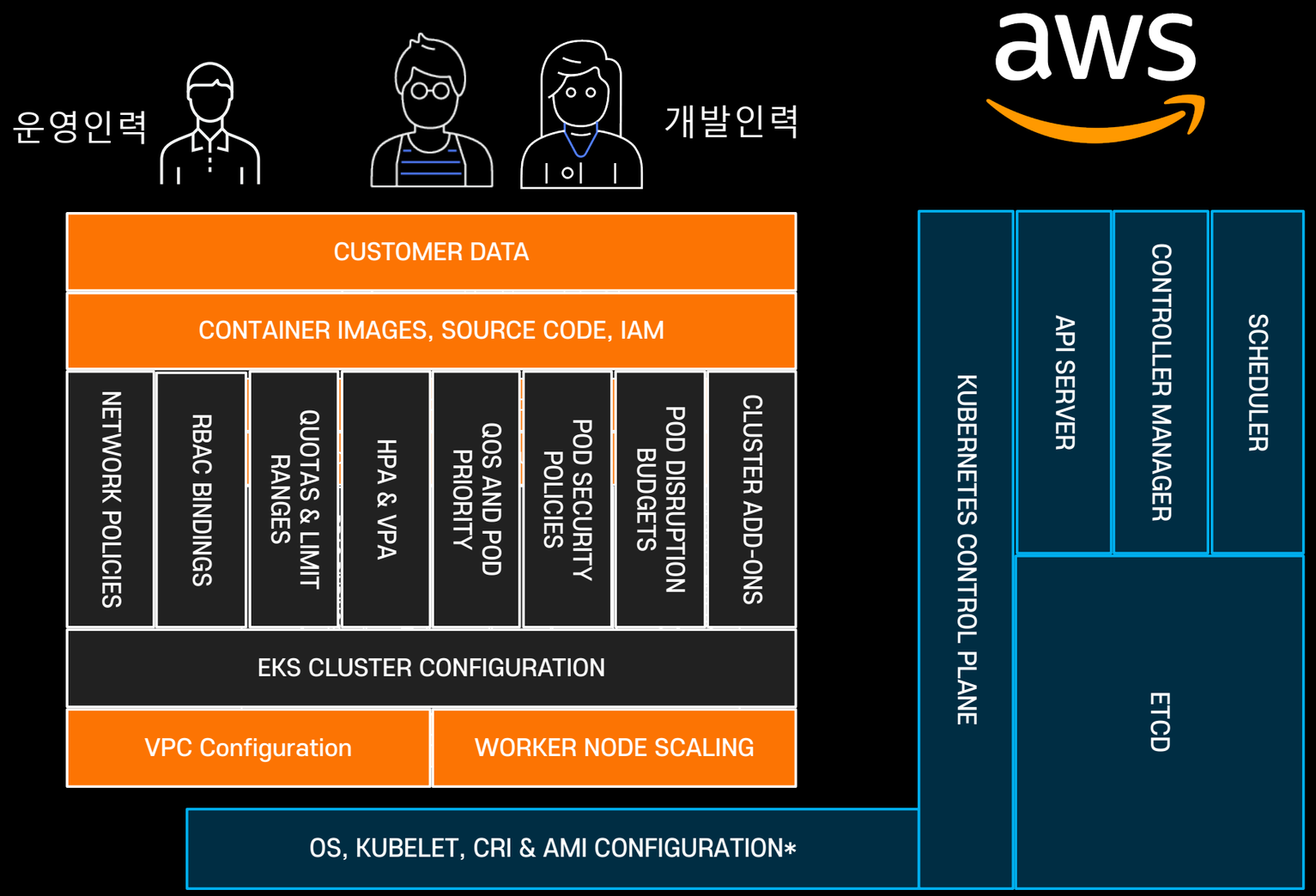
AWS 책임 영역

고객의 책임 영역

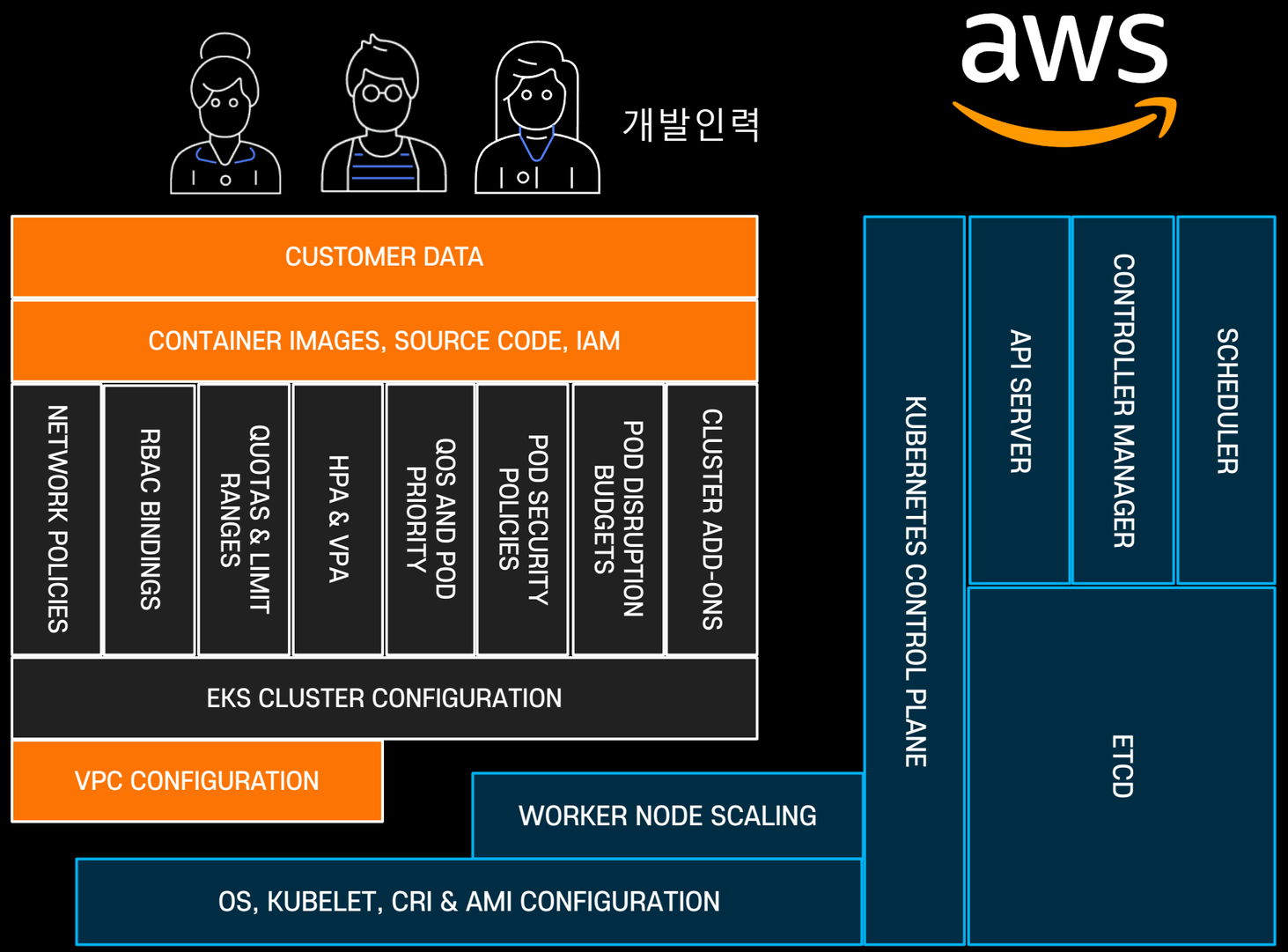


aws

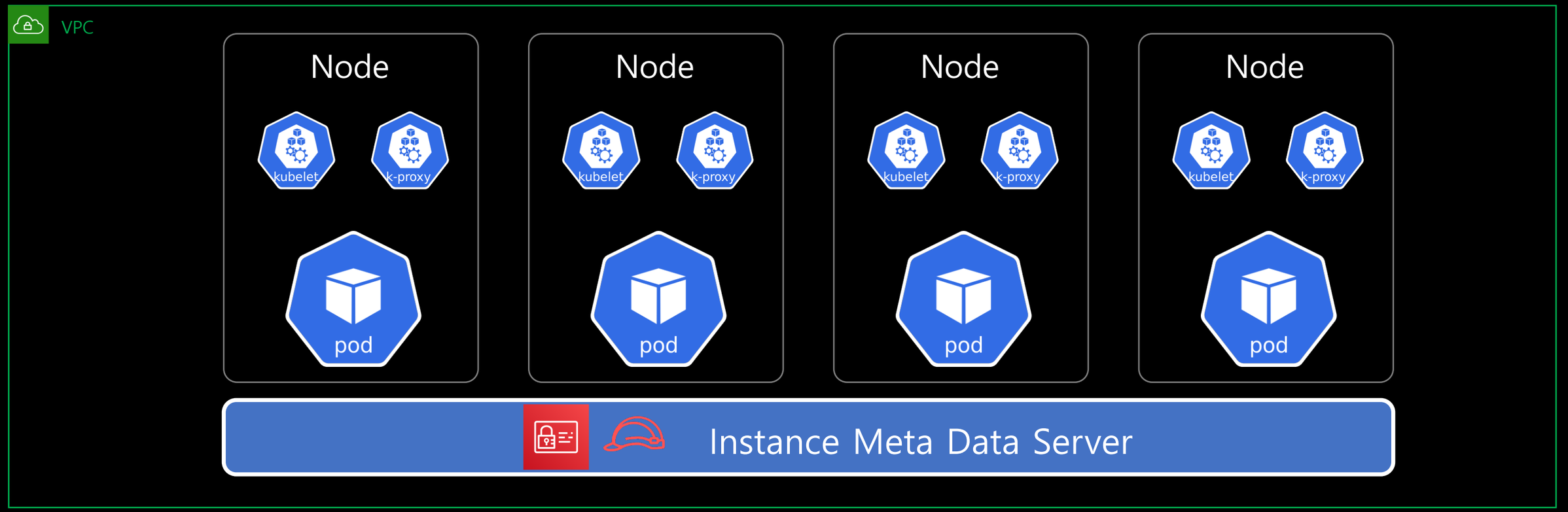
# Amazon EKS 책임 공유 모델(Managed Node Group)



# Amazon EKS 책임 공유 모델(Fargate)



# K8S on AWS 환경에서의 IAM




AWS KMS

AWS Secrets Manager

GuardDuty

Amazon Inspector

AWS CloudTrail

Amazon CloudWatch

AWS Config

AWS Systems Manager

aws

# K8S User & Service Account

사용자 혹은 그룹

K8S 는 "Authentication"을 위한 구성요소는 포함하고 있지 않으며 "Authorization" 만 수행

User

Service Account

user

sa

애플리케이션

Cluster Role Binding

crb

rb

Role Binding

Cluster Role

c.role

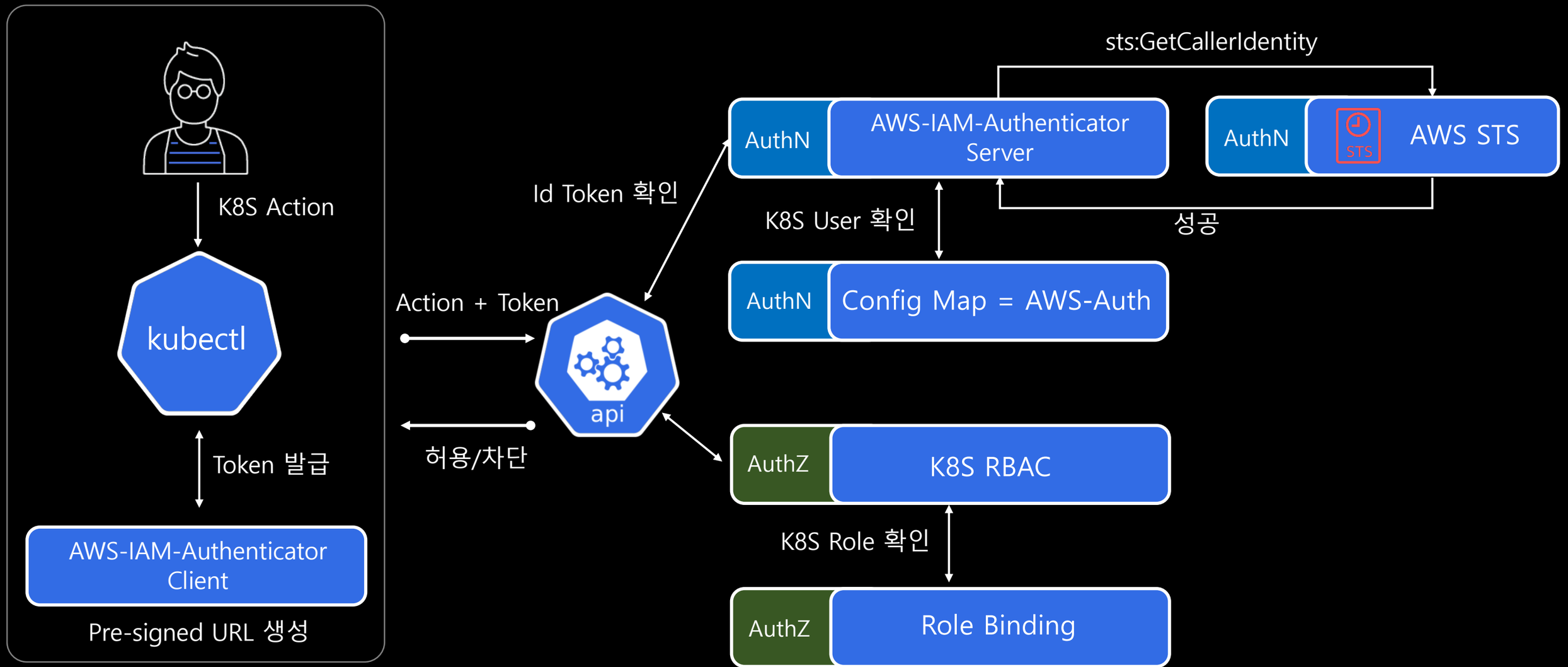
role

Role

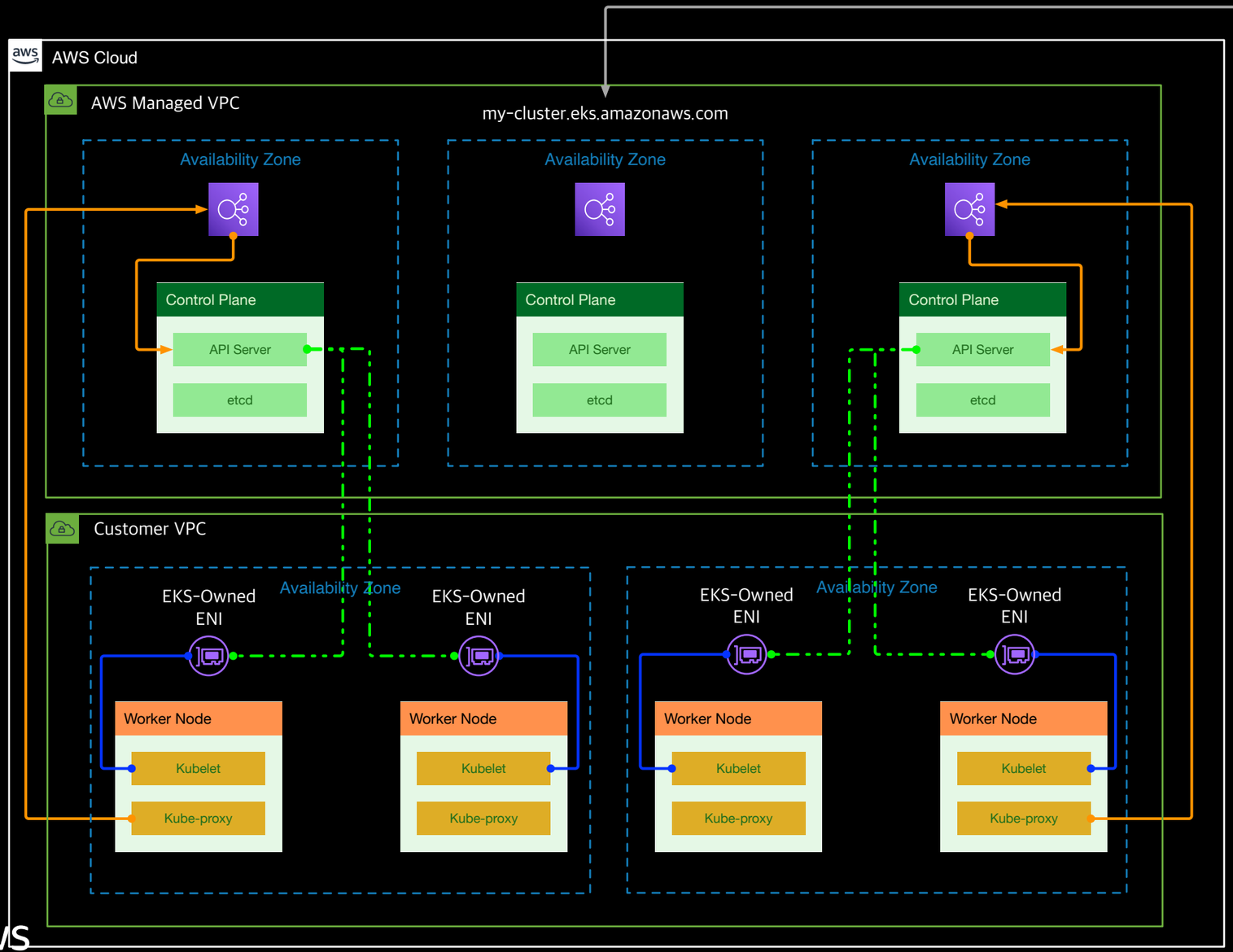
Cluster 자원

NameSpace 자원

# EKS: IAM Authentication



# EKS Public Endpoint

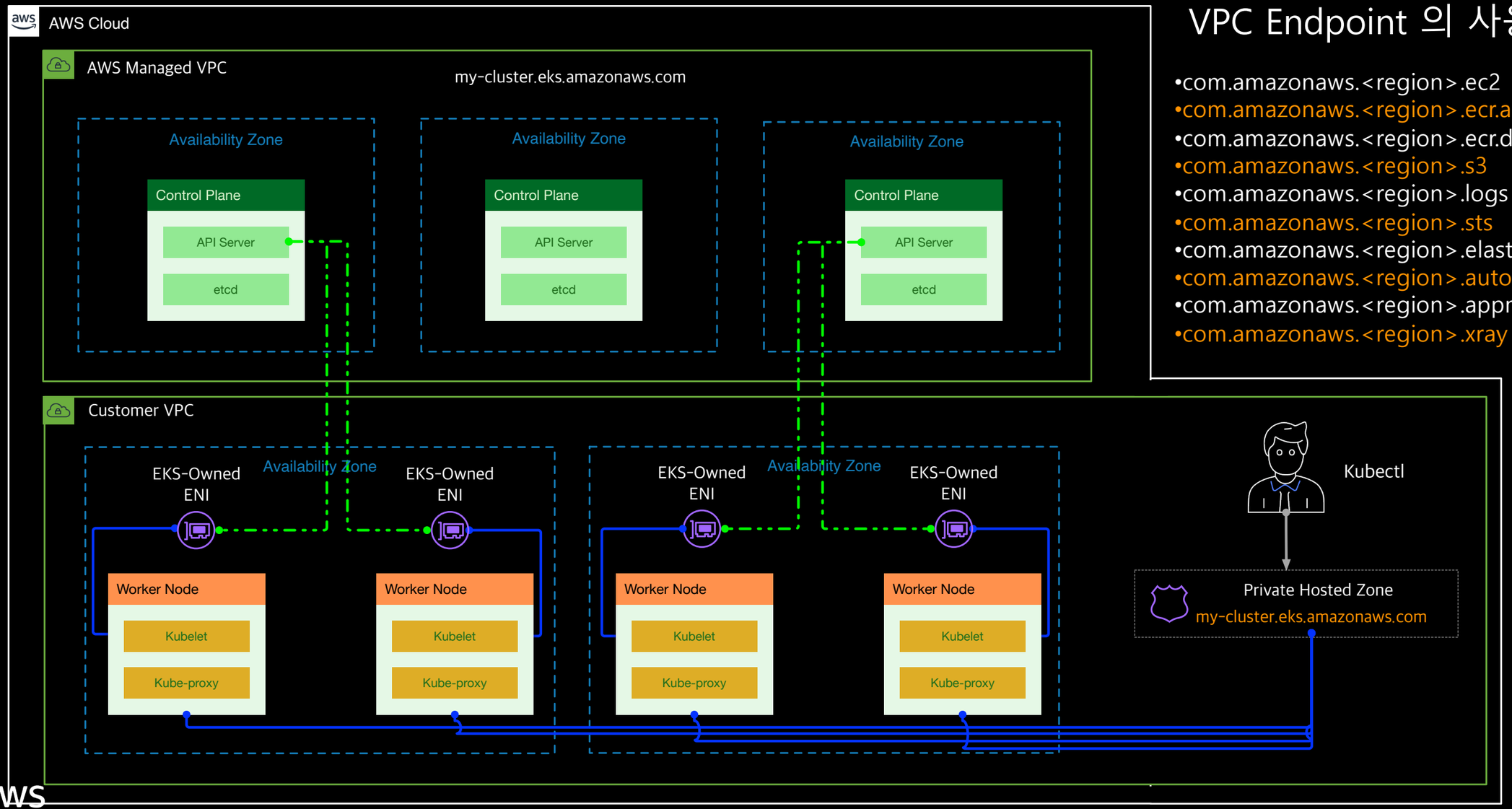



Kubectl

## EKS Public Endpoint

- EKS 의 기본 설정값
- Customer VPC 의 자원들은 Internet Gateway 를 통해 Control Plane 에 접근
- HTTPS 암호화된 통신 채널 사용
- 인증된 사용자에 한해서만 호출 가능

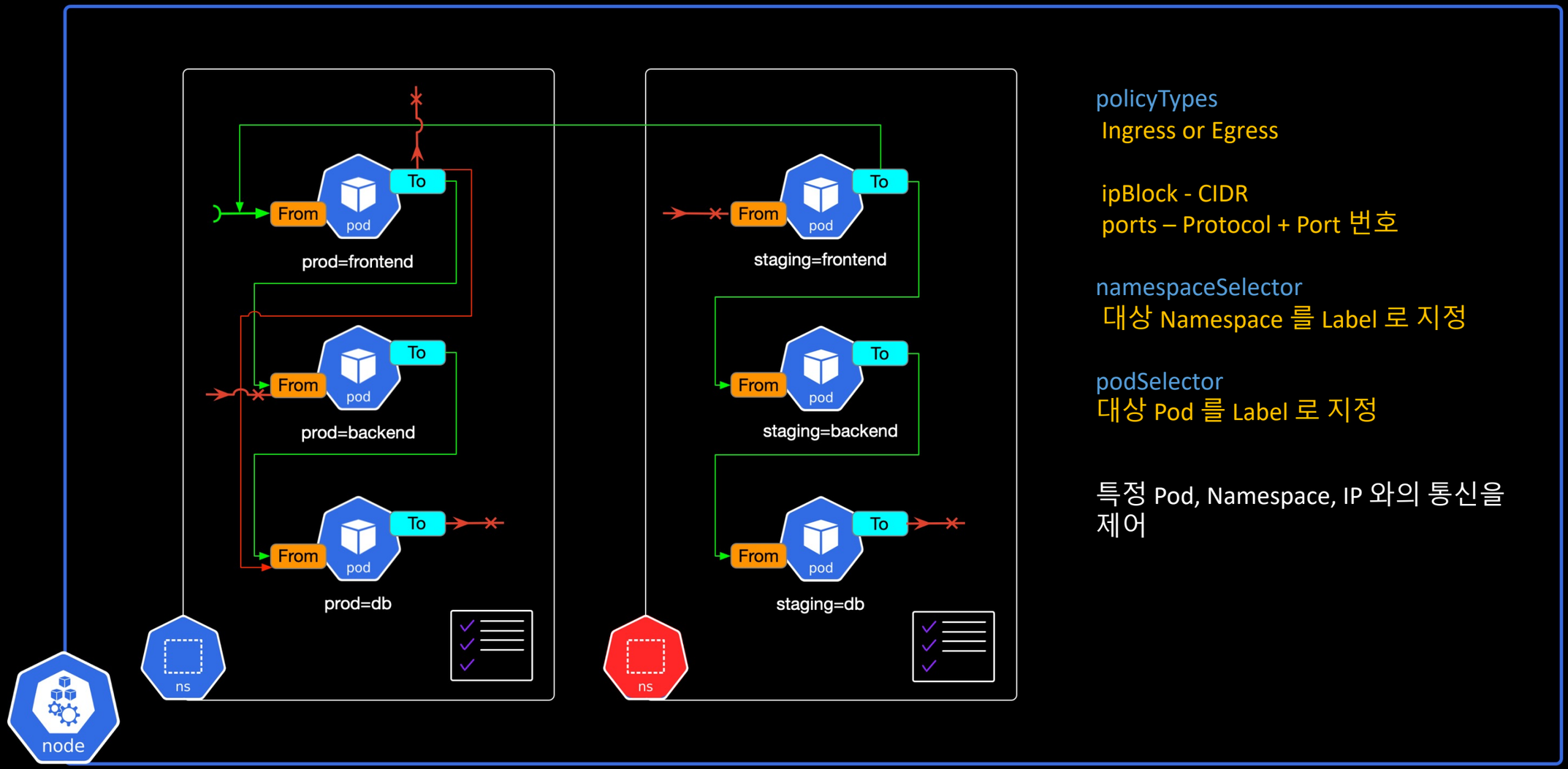
# EKS Private Endpoint




## VPC Endpoint 의 사용

- com.amazonaws.<region>.ec2
- com.amazonaws.<region>.ecr.api
- com.amazonaws.<region>.ecr.dkr
- com.amazonaws.<region>.s3
- com.amazonaws.<region>.logs
- com.amazonaws.<region>.sts
- com.amazonaws.<region>.elasticloadbalancing
- com.amazonaws.<region>.autoscaling
- com.amazonaws.<region>.appmesh-envoy-management
- com.amazonaws.<region>.xray

# Network Policy




policyTypes
Ingress or Egress

ipBlock - CIDR
ports – Protocol + Port 번호

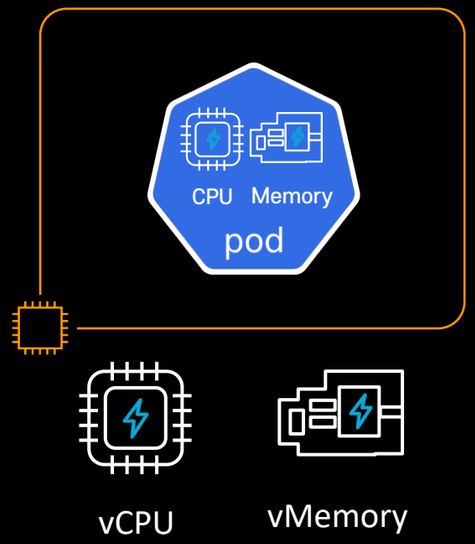
namespaceSelector
대상 Namespace 를 Label 로 지정

podSelector
대상 Pod 를 Label 로 지정

특정 Pod, Namespace, IP 와의 통신을 제어

# EKS 에서의 자원 확장

# EKS 에서의 Auto Scaling




CA

Cluster Auto-Scaler

Pod 을 배포할 Node 가 부족한 경우 신규 Node Provisioning

hpa

Horizontal Pod Auto-Scaler

서비스를 처리할 Pod 자원이 부족한 경우 신규 Pod Provisioning

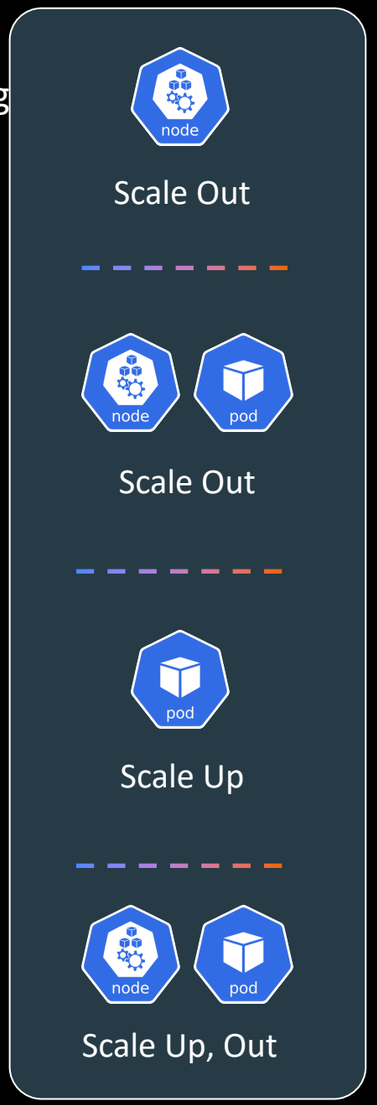
vpa

Vertical Pod Auto-Scaler

서비스를 처리할 Pod 자원이 부족한 경우 Pod 교체 (자동 or 수동)

Karpenter

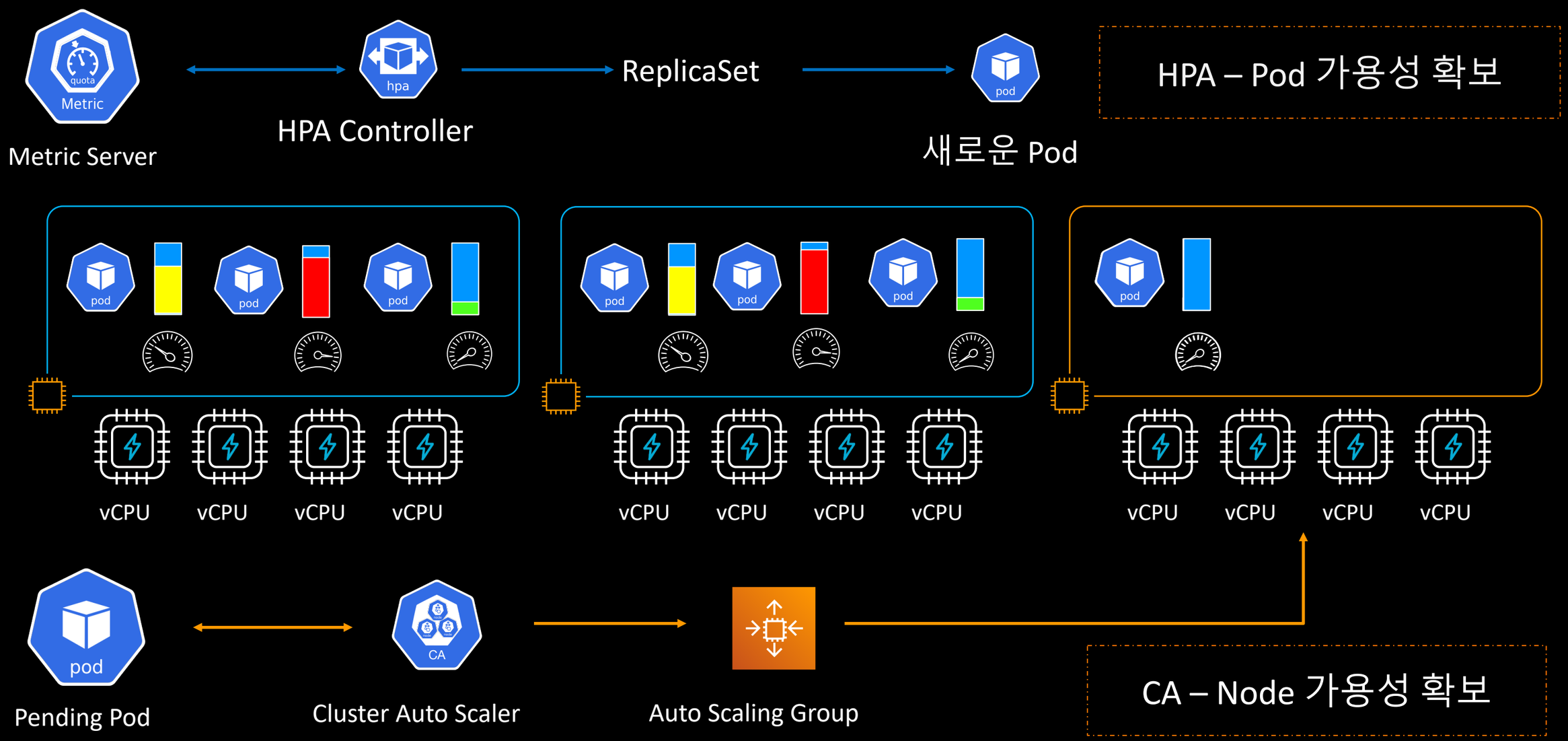
Unscheduled Pod 이 있는 경우 새로운 Node 및 Pod Provisioning



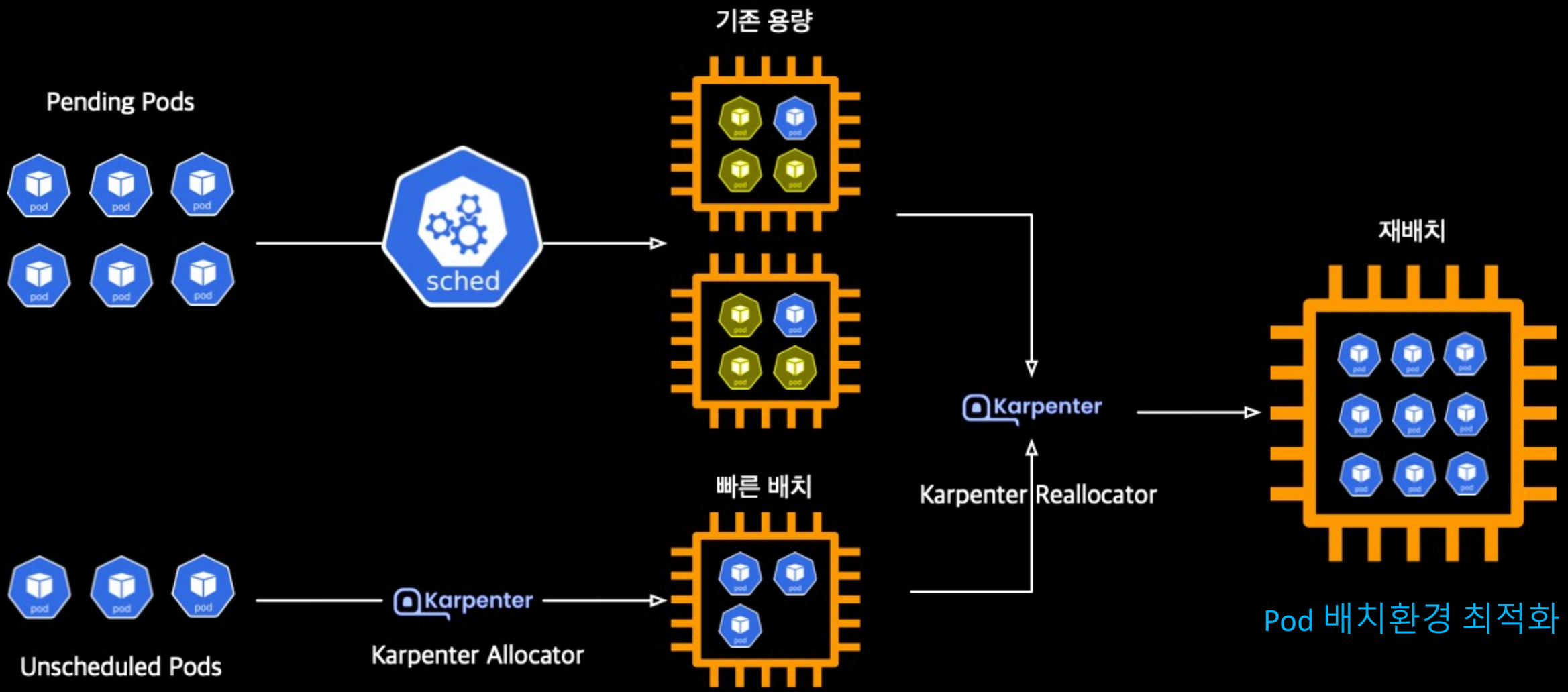



aws

# EKS 환경에서의 Auto Scaling

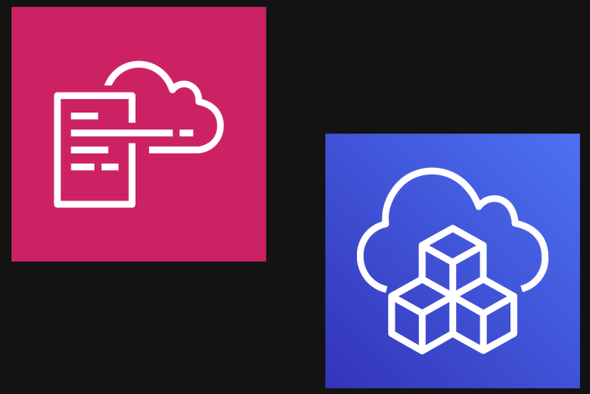


# Karpenter(Groupless Node ASG)



# Amazon EKS 구성 및 설치

# Amazon EKS Cluster 배포 방식



AWS CloudFormation &
AWS CDK

eksctl

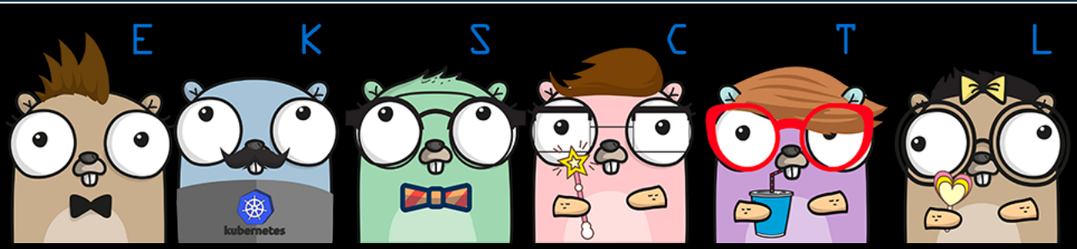

Terraform, Pulumi, Rancher
… more



aws

# eksctl로 쿠버네티스 클러스터 배포하는 예시

```
beta_zero:~/environment $ cat eksworkshop.yaml
---
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: eksworkshop-eksctl
  region: ap-southeast-1
  version: "1.17"

availabilityZones: ["ap-southeast-1a", "ap-southeast-1b", "ap-southeast-1c"]

managedNodeGroups:
- name: nodegroup
  desiredCapacity: 3
  ssh:
    allow: true
    publicKeyName: eksworkshop

# To enable all of the control plane logs, uncomment below:
# cloudWatch:
#  clusterLogging:
#    enableTypes: ["*"]

secretsEncryption:
  keyARN: arn:aws:kms:ap-southeast-1:[redacted]:key/93516110-3af6-4bcf-80ca-9140b703828d
beta_zero:~/environment $ eksctl create cluster -f eksworkshop.yaml
```



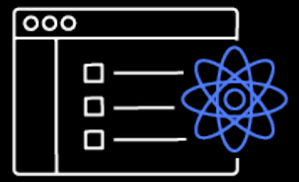

https://eksctl.io/



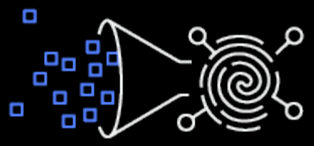
aws

# 요약

# 다양한 산업 및 애플리케이션 유형을 지원하는 Amazon EKS



Web Applications



Data Processing

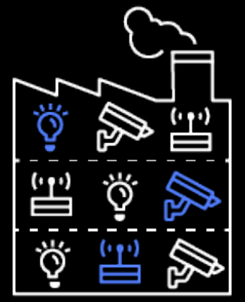
Machine Learning

CI/CD

Mobile Applications

Gaming Platforms

Platform as a Service (PaaS)



Internet of Things



aws

# 감사합니다

# 관련 서비스들

# Amazon Elastic Container Registry(ECR)

완전 관리형 컨테이너 레지스트리

배포 워크플로우 간소화

- Amazon EKS, Amazon ECS, Amazon Lambda, AWS Fargate

권한 제어 및 수명 주기 정책 규칙 설정

이미지 취약성 스캐닝 기능

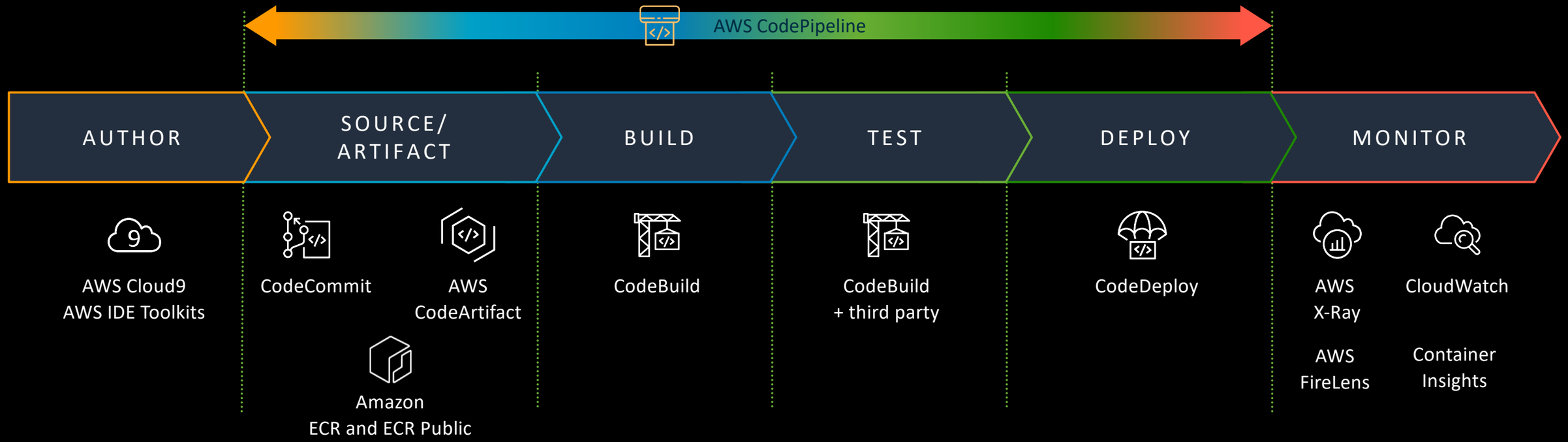
퍼블릭/프라이빗 레파지토리 및 퍼블릭 갤러리

교차 리전/교차 계정 복제



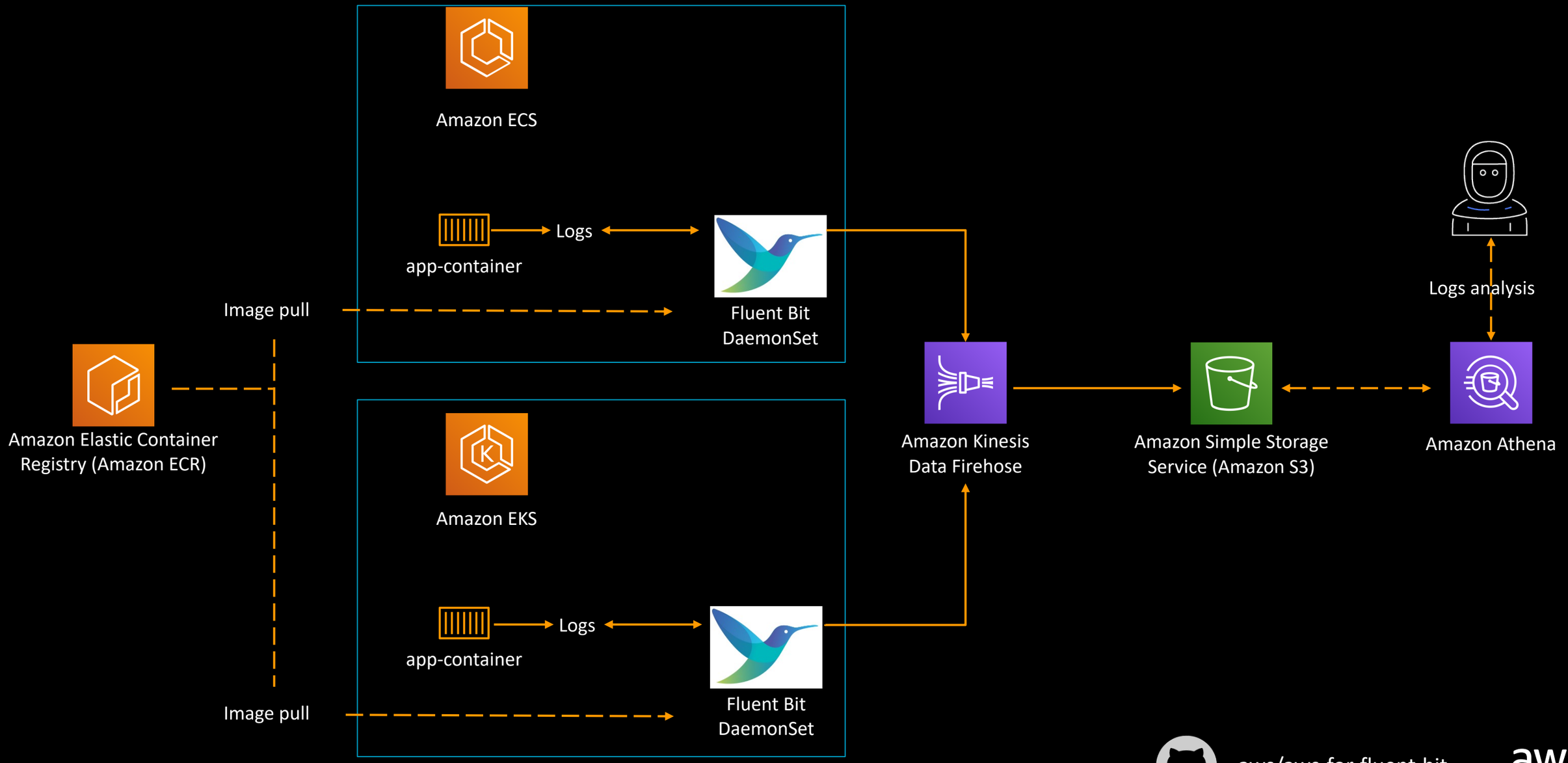
aws

# AWS 개발자 도구를 활용한 배포 자동화




MODEL AWS CloudFormation AWS Cloud Development Kit (CDK, CDK8s, CDK-terraform) AWS Amplify AWS Copilot Docker Compose

# Logging with Fluent Bit



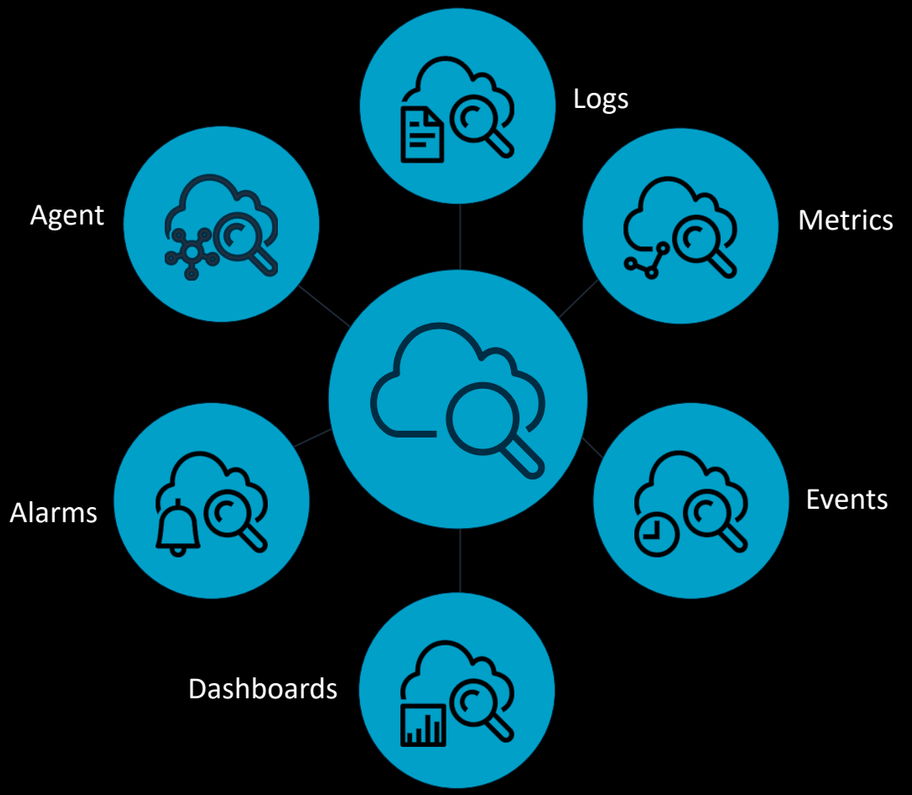



aws/aws-for-fluent-bit

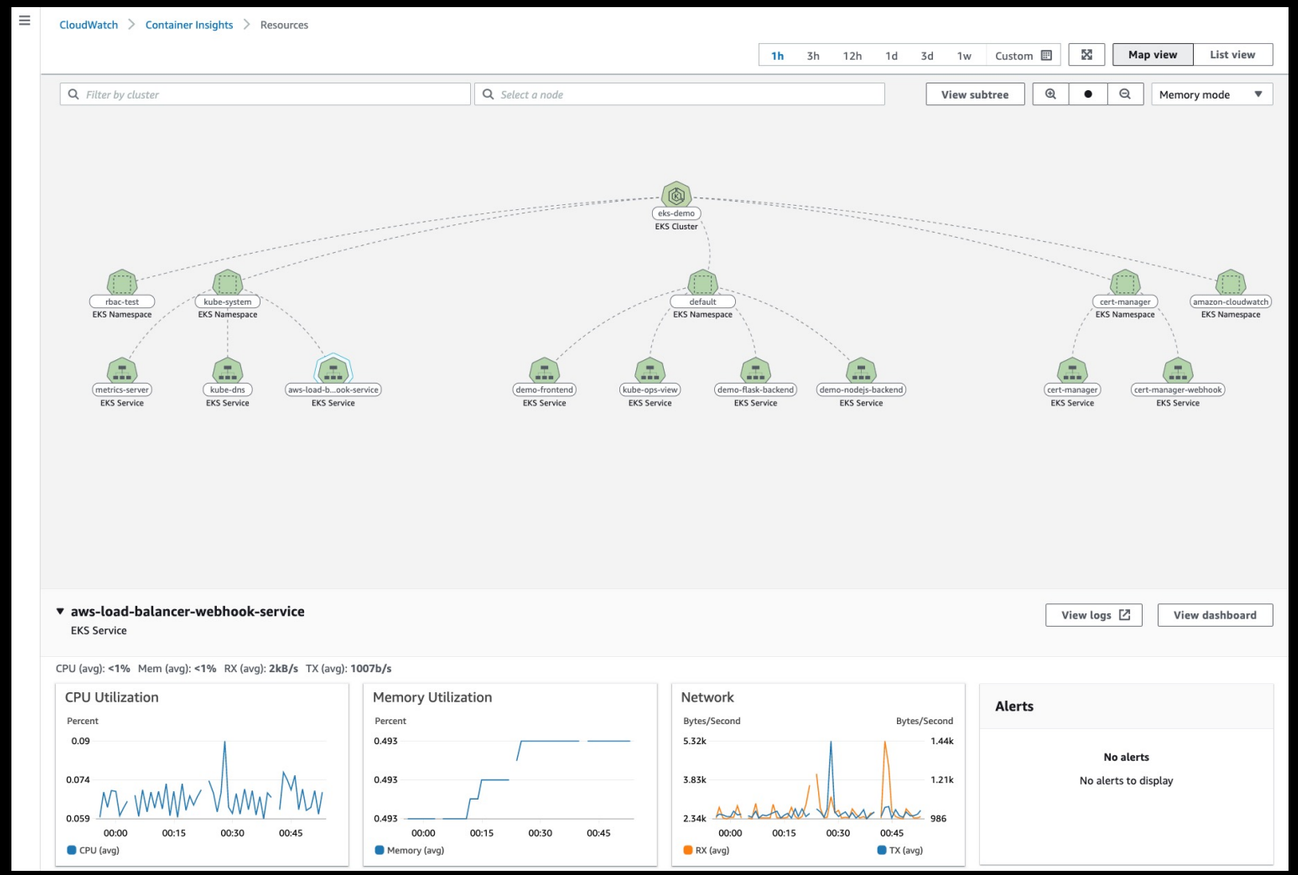
# Amazon CloudWatch Container Insights

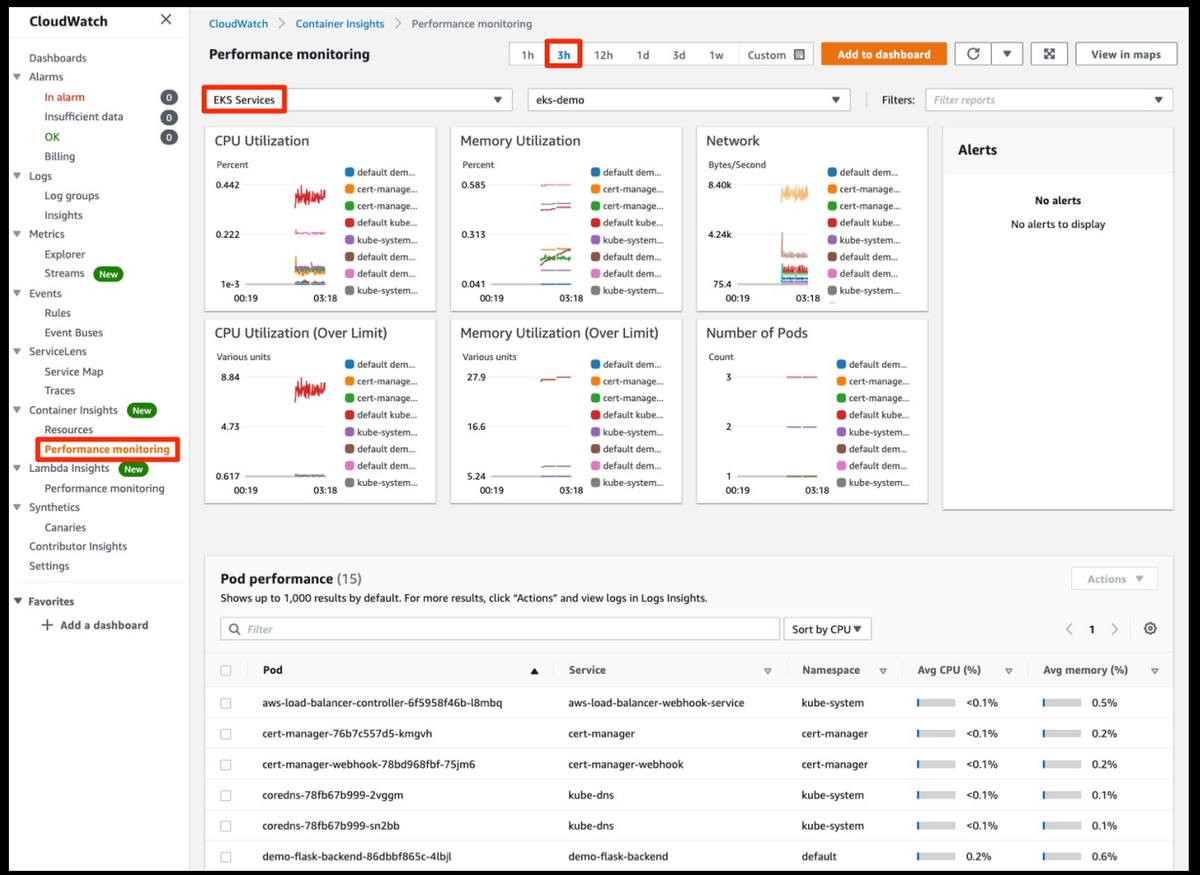
컨테이너화된 애플리케이션이나 마이크로서비스에 대한 모니터링, 트러블슈팅 및 알람을 위한 완전 관리형 관측 서비스

- 안정적이고 안전한 지표 및 로그 수집
- 자동화된 요약과 분석
- 지표, 로그, 트레이스 전반에 대한 가시성
- 사전 생성된 대시보드



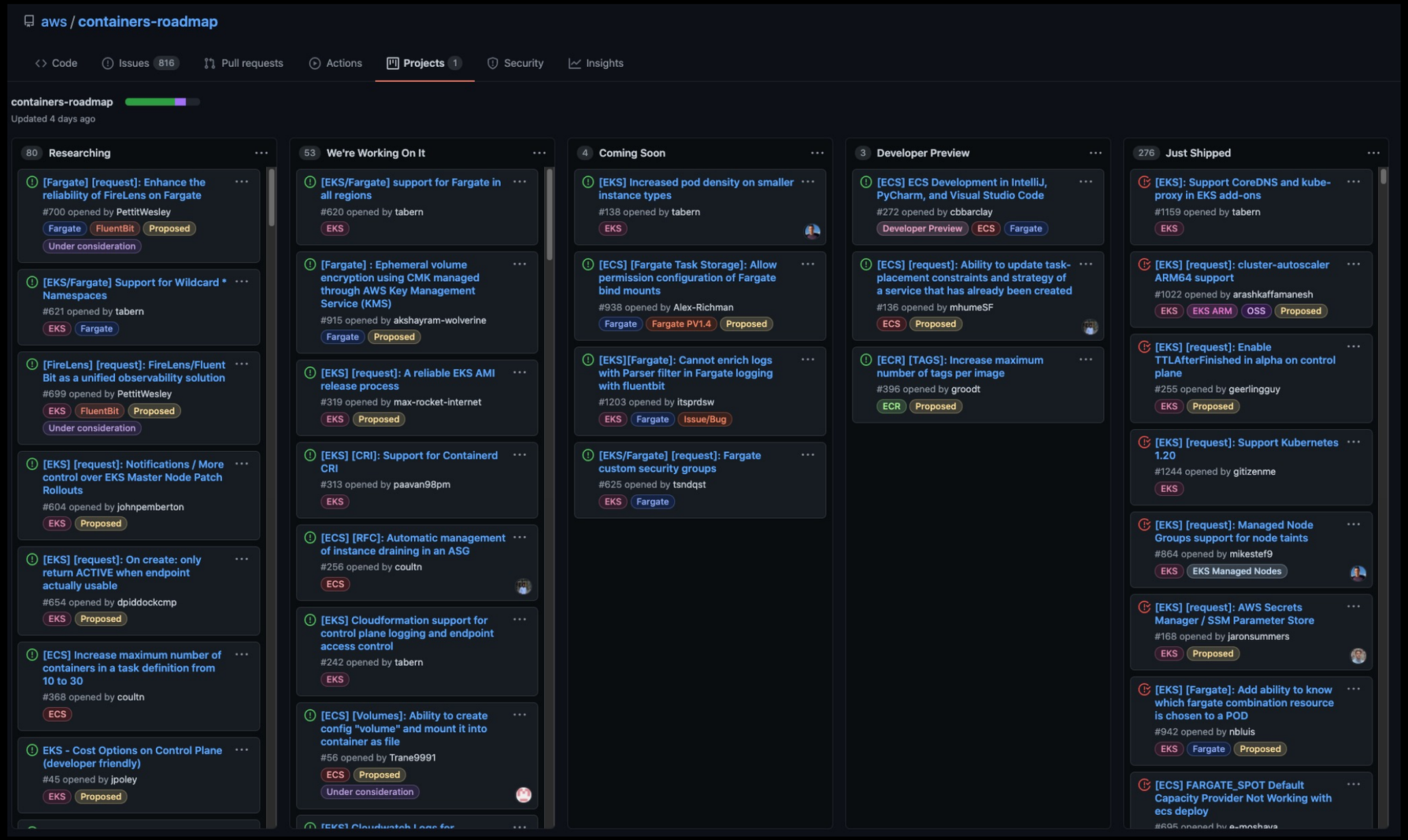
# 예시 화면



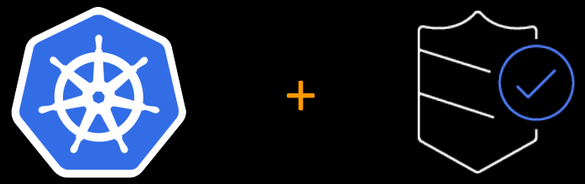
# 참고 자료

# 오픈된 로드맵




https://github.com/aws/containers-roadmap/projects/1

# AWS는 고객이 어떻게 Kubernetes를 운영하는지에 대해 상관 없이 Kubernetes 커뮤니티를 지원하기 위해 노력하고 있습니다.



AWS에서는 보안을 최우선으로 여기며 이는
쿠버네티스 부분에서도 마찬가지입니다.
AWS는 쿠버네시트 프로젝트의 보안을 담당하는
Product Security Committee의 멤버입니다.

- kubernetes/cloud-provider-aws
- kubernetes/autoscaler
- kubernetes-sigs/aws-ebs-csi-driver
- kubernetes-sigs/aws-efs-csi-driver
- kubernetes-sigs/aws-fsx-csi-driver
- kubernetes-sigs/aws-iam-authenticator
- kubernetes-sigs/aws-load-balancer-controller
- kubernetes-sigs/aws-encryption-provider
- kubernetes-sigs/aws-encryption-provider
- awslabs/karpenter
- aws/eks-distro

CNI, CSI, Kubernetes 등등

# 유용한 링크

- EKS 모범 사례 가이드
  https://aws.github.io/aws-eks-best-practices/
- Amazon EKS 워크샵 페이지
  https://www.eksworkshop.com/
  https://aws-eks-web-application.workshop.aws/ko/
- AWS Observability 워크샵 페이지
  https://observability.workshop.aws/ko/intro.html
- AWS Containers from the couch 유튜브
  https://www.youtube.com/containersfromthecouch
- CNCF 유튜브
  https://www.youtube.com/c/cloudnativefdn/featured
- https://learnk8s.io/blog
- https://github.com/kubernetes-sigs



aws