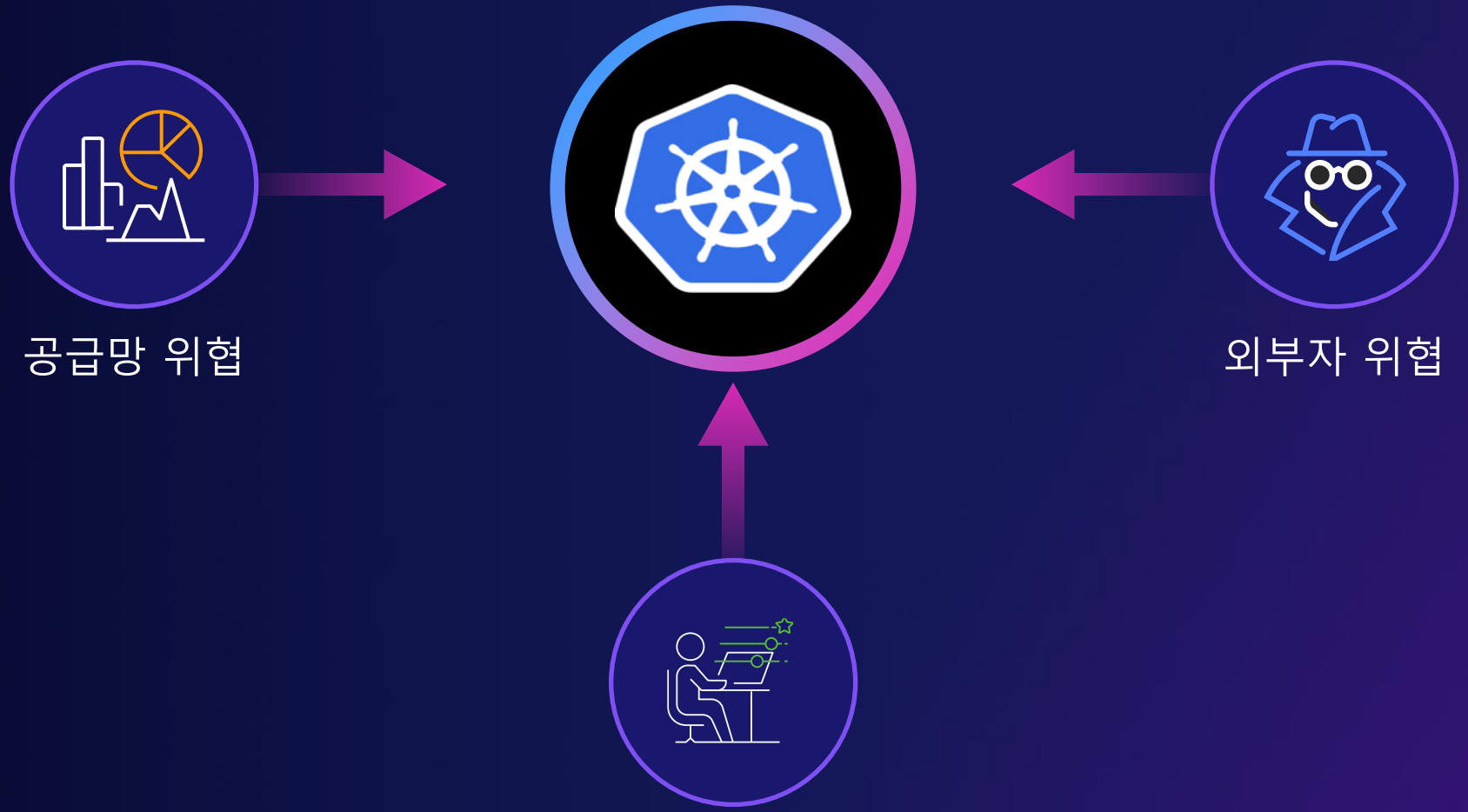

aws
SUMMIT
KOREA | MAY 10-11, 2022

T10S1

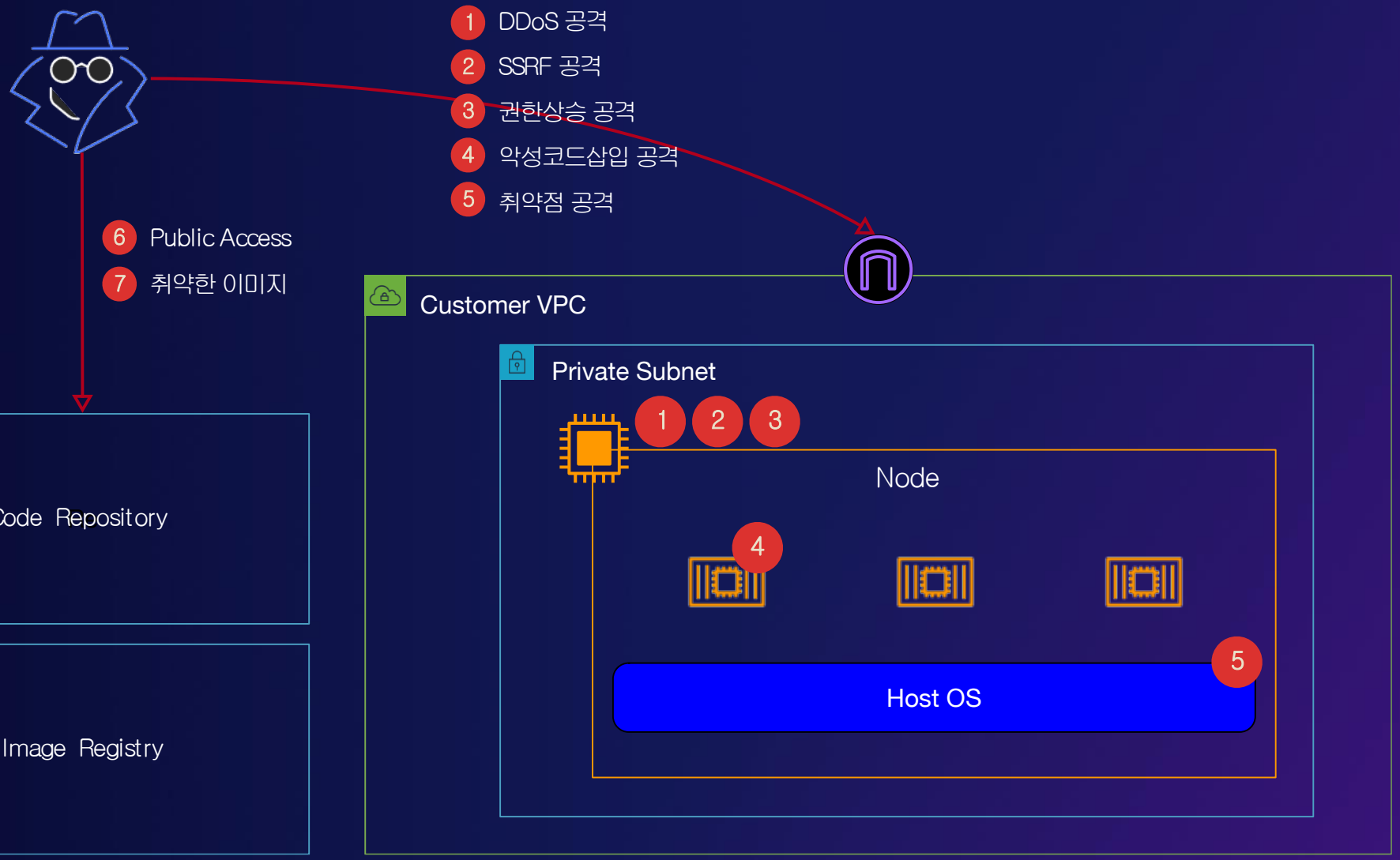
EKS 환경을 더 효율적으로, 더 안전하게

신은수
시큐리티 스페셜리스트 솔루션즈 아키텍트
AWS

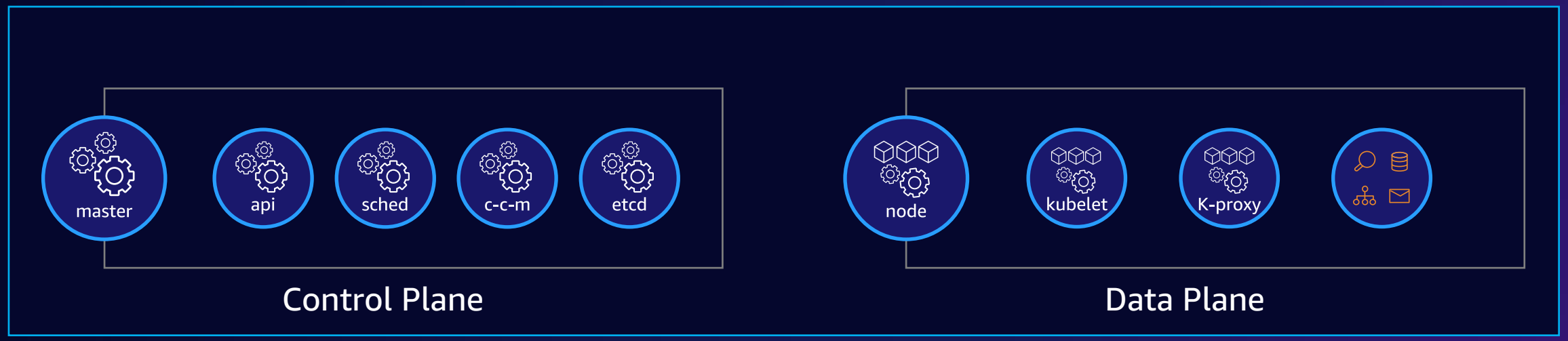
K8S 위협 모델



K8S 보안 위협



K8S 의 각 계층별 보안



- Identity and Access Management
- Data Encryption and Secrets Management
- Infrastructure Security
- Pod Security
- Multi-tenancy
- Image Security
- Network Security
- Runtime Security
- Detective Controls
- Incident Response and Forensics

OS 레벨의 보안 적용

- 시스템 접근 제어
- 주기적인 취약점 점검
- 반드시 필요한 라이브러리만 포함
- Hardening 처리된 OS 사용

EKS Optimized AMI

Inspector

Bottlerocket

SELinux(Security-Enhanced Linux)

- Mandatory Access Control (MAC)
- Label & Type Enforcement
- 파일, 프로세스, 포트 등에 적용
- securityContext

보호 대상 프로세스에 Policy 적용

MLS or MCS

Label – User, Role, Type, Level

```
securityContext:
  seLinuxOptions:
    # Provide a unique MCS label per container
    # You can specify user, role, and type also
    # enforcement based on type and level (svert)
    level: s0:c144:c154
```

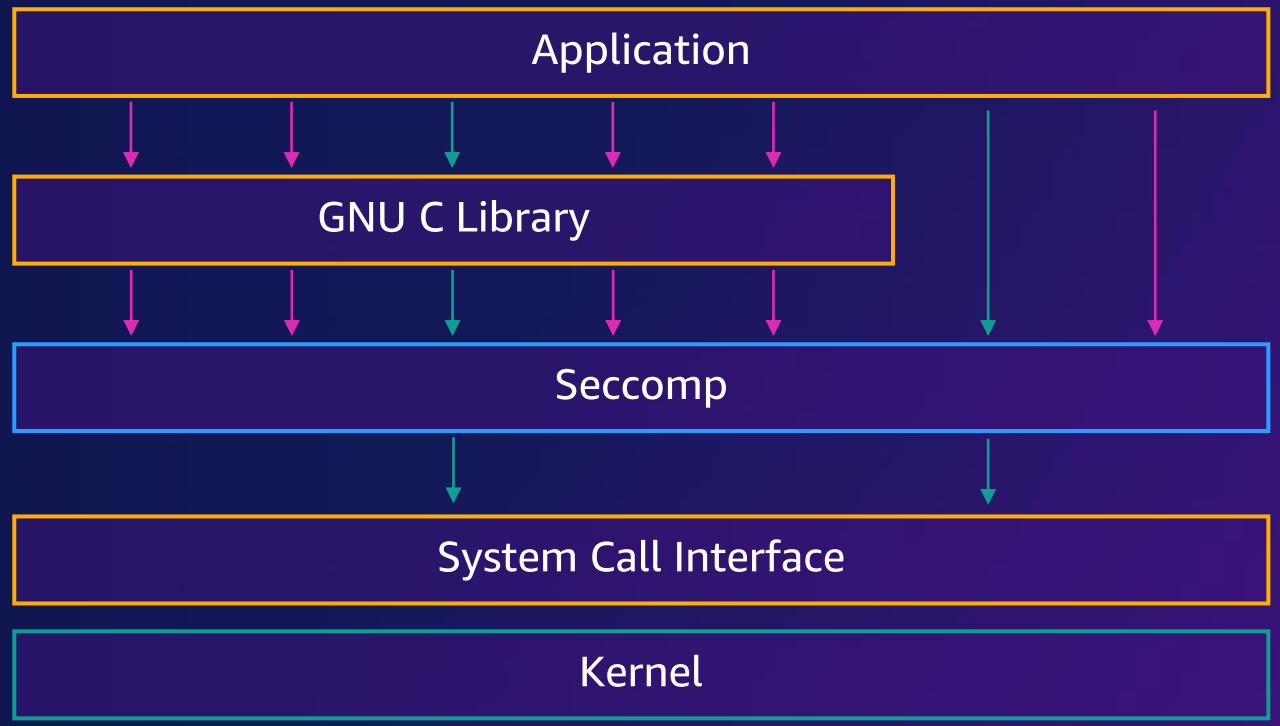
SELinux를 이용한 컨테이너 호스트 격리

기본값으로 컨테이너들은 "Container_t" Label 처리

SECCOMP(SECure COMPuting mode)

- System Call 에 대한 제한
- Container 간 격리는 없음
- SeccompDefault

```
securityContext:
  seccompProfile:
    type: RuntimeDefault
```



AppArmor(Application Armor)

- Mandatory Access Control(MAC)
- 프로세스별 보안 프로필

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
  annotations:
    # Tell Kubernetes to apply the AppArmor profile "k8s-apparmor-example-deny-write".
    container.apparmor.security.beta.kubernetes.io/hello: localhost/k8s-apparmor-example-deny-write
spec:
  containers:
  - name: hello
    image: busybox
    command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]
```

Container Image 관리

Image Scanning

Elastic Container Registry

Basic scanning

Basic scanning allows manual scans and scan on push of images in this registry. This is a free service

Enhanced scanning

Enhanced scanning with Amazon provides automated continuous scanning. Inspector identifies vulnerabilities in both operating system and programming language(such as python, java, Ruby etc.) packages in real time

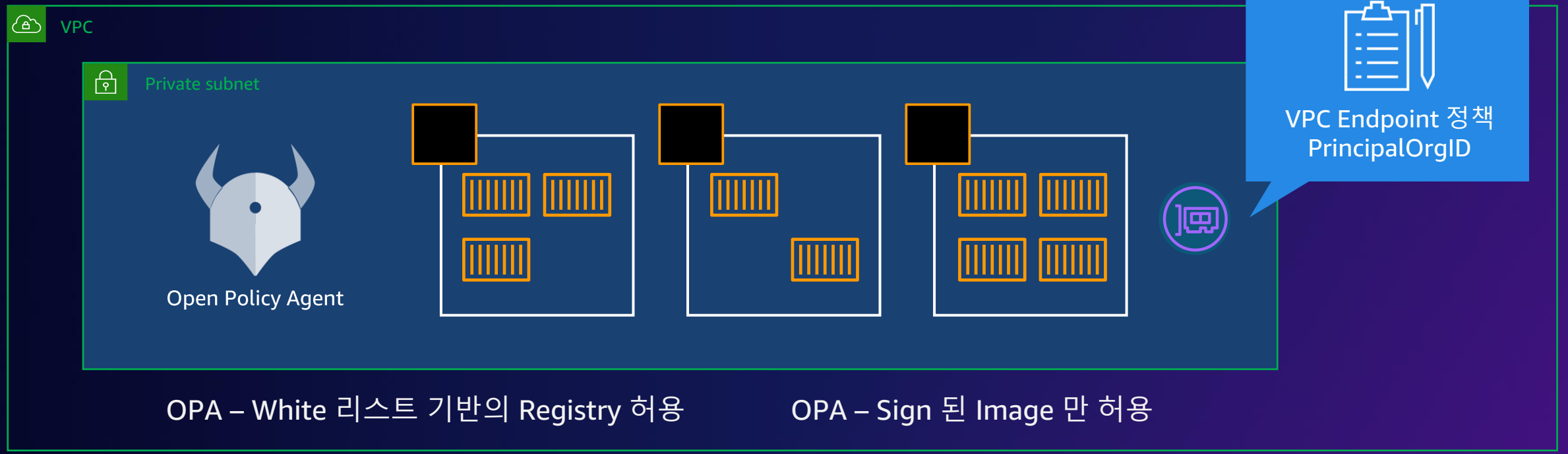
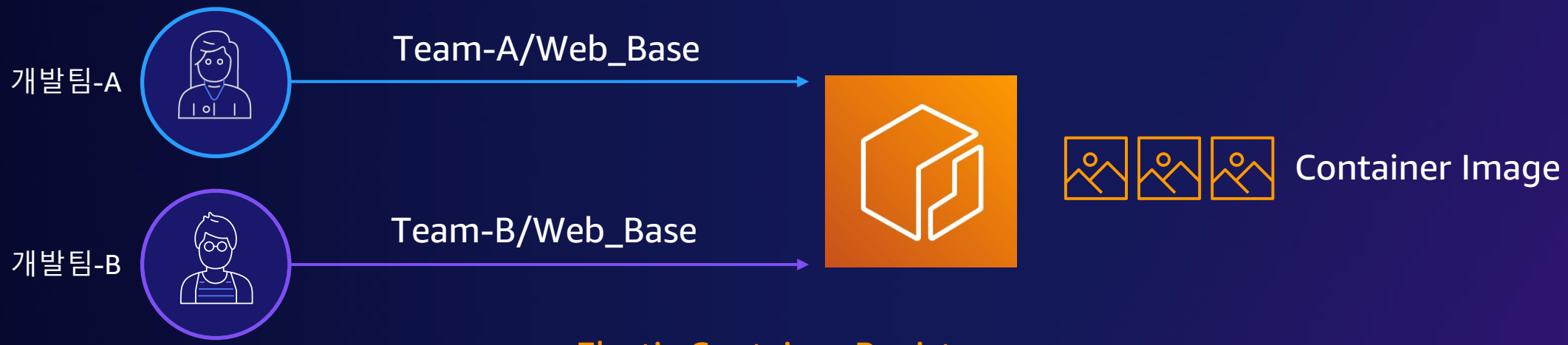


Image 저장소 보안



Elastic Container Registry

https://123456789012.dkr.ecr.ap-northeast-2.amazonaws.com

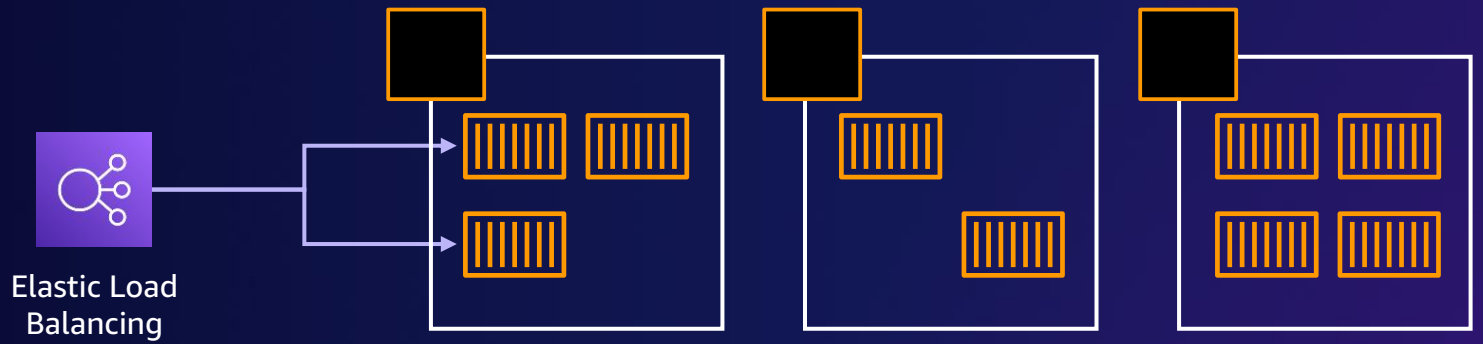


Image 저장소 보안


```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECR Repository Policy",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::account-id:user/username"
    },
    "Action": [
      "ecr:DescribeImages",
      "ecr:DescribeRepositories"
    ],
    "Resource": [
      "arn:aws:ecr:region:account-id:repository/repository-name"
    ]
    }]
}
```

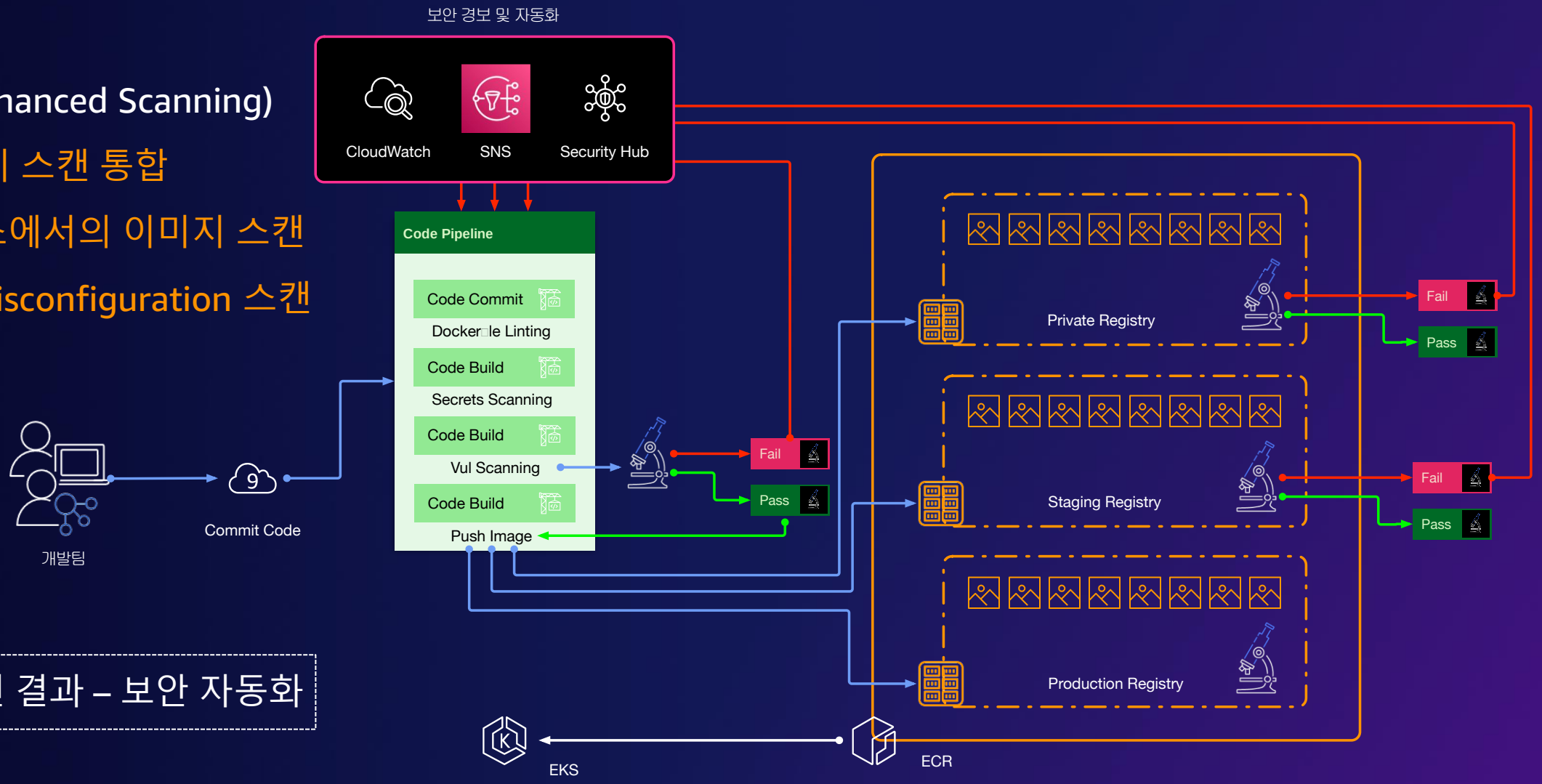
Image 저장소 보안

Inspector(Enhanced Scanning)

CI/CD 이미지 스캔 통합

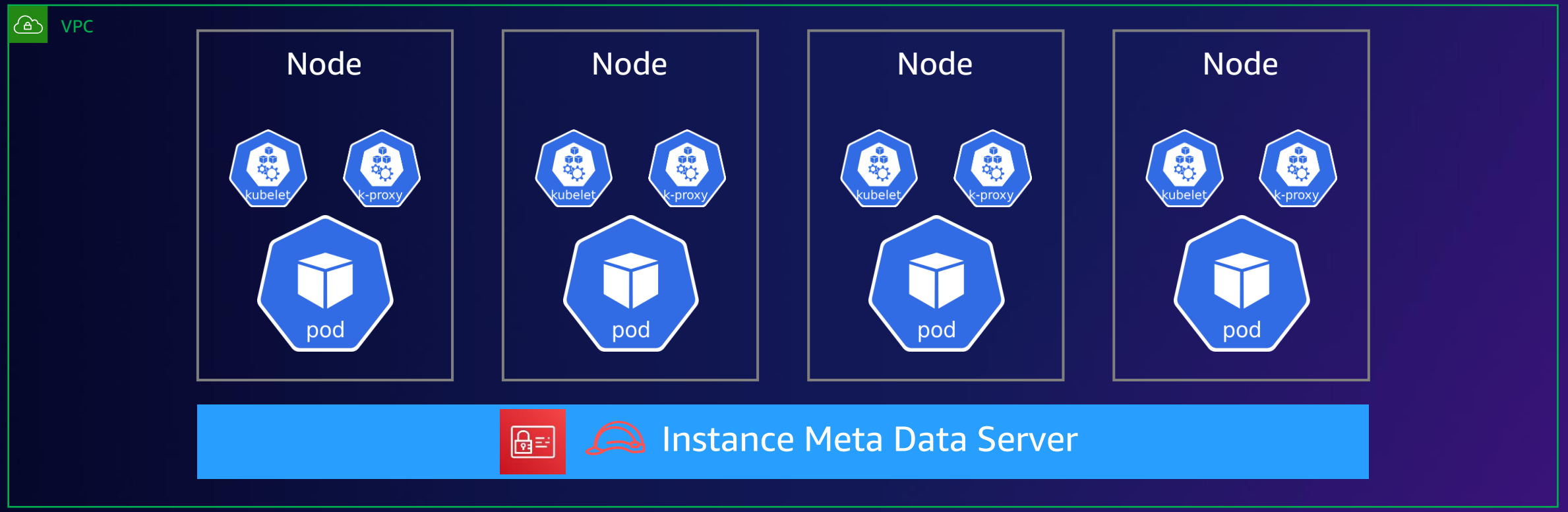
Image 저장소에서의 이미지 스캔

Dockerfile Misconfiguration 스캔



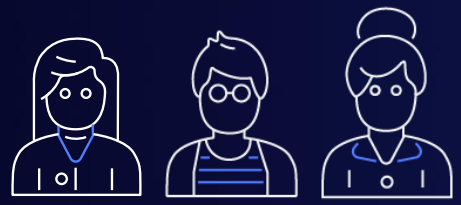
이미지스캔 결과 – 보안 자동화

EKS 환경에서의 IAM



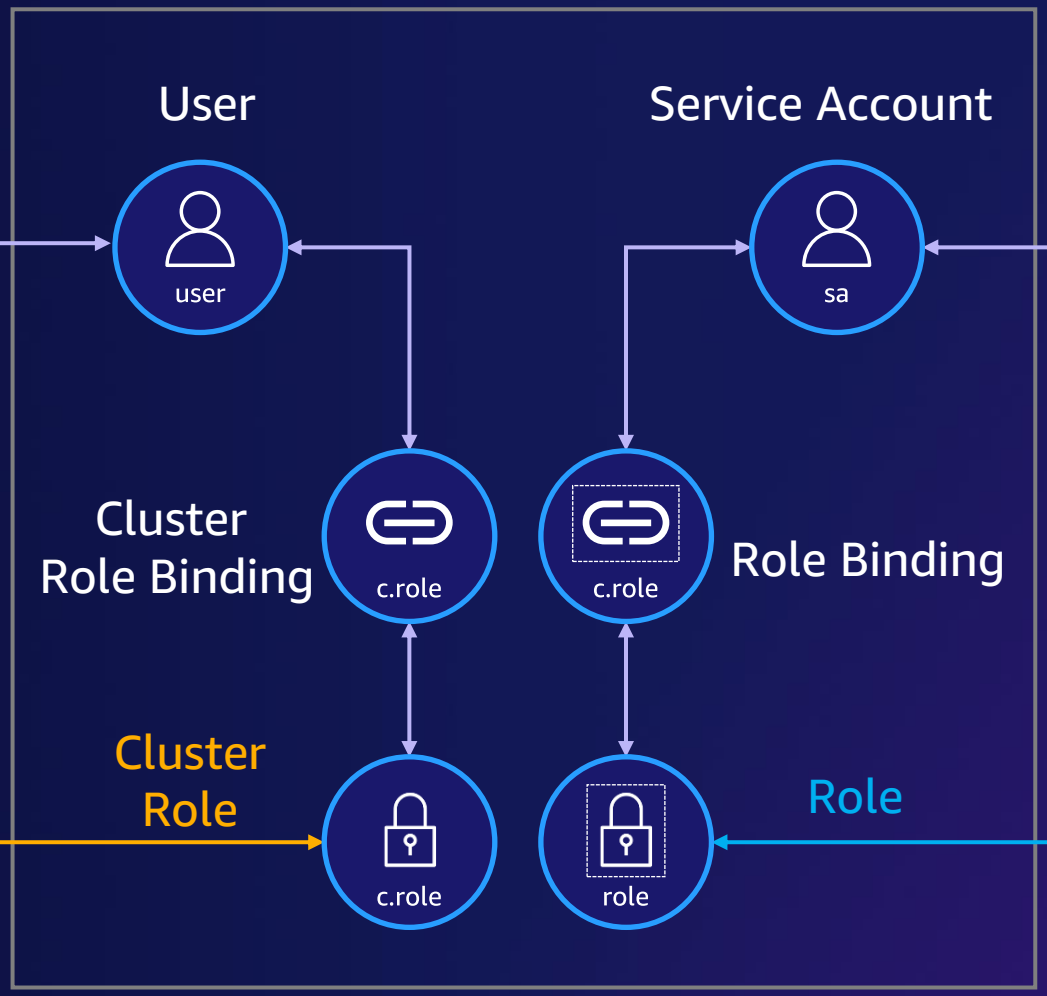
AWS KMS | AWS Secrets Manager | GuardDuty | Amazon Inspector | AWS CloudTrail | Amazon CloudWatch | AWS Config | AWS Systems Manager

K8S User & Service Account



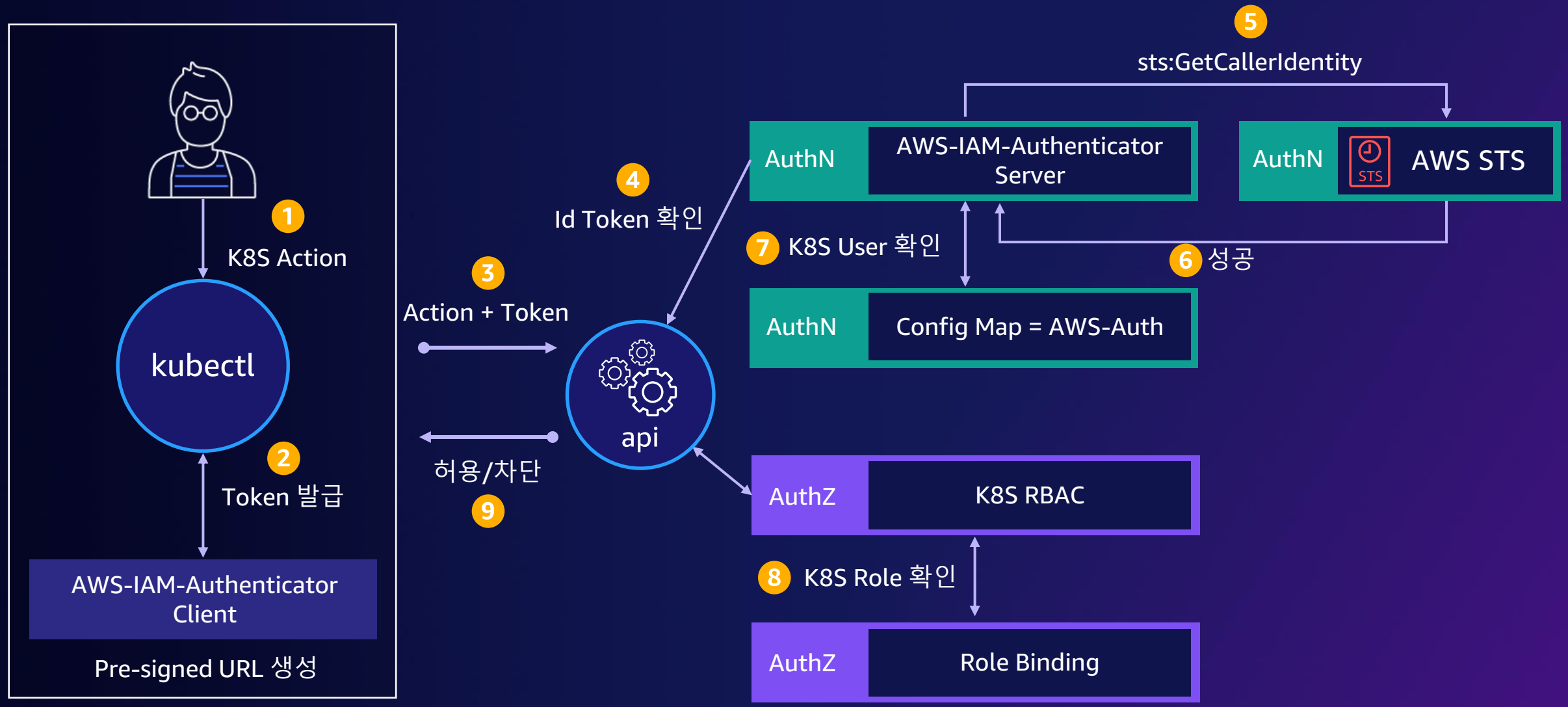
사용자 혹은 그룹

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

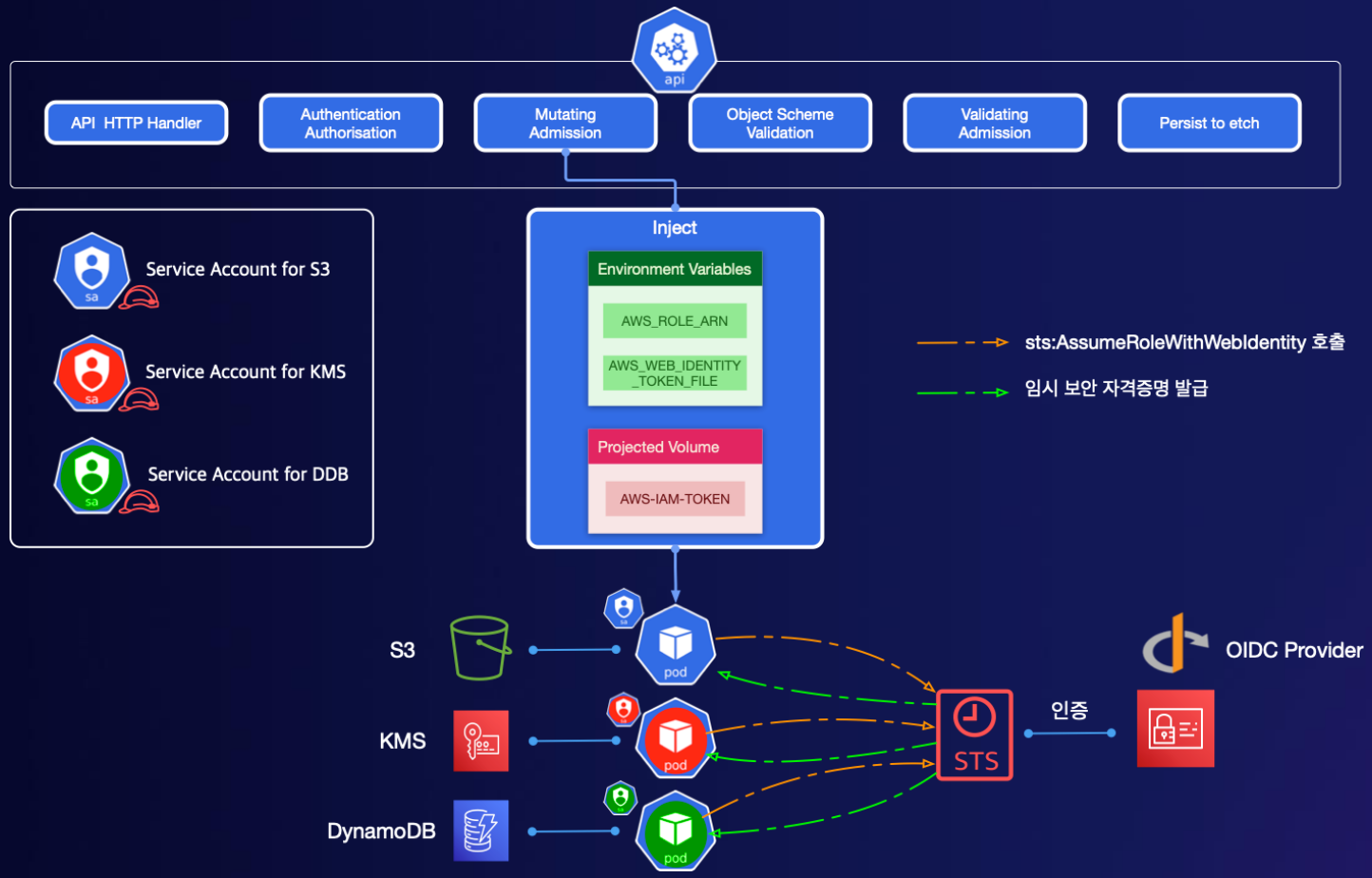


애플리케이션

IAM Authenticator 의 인증과정

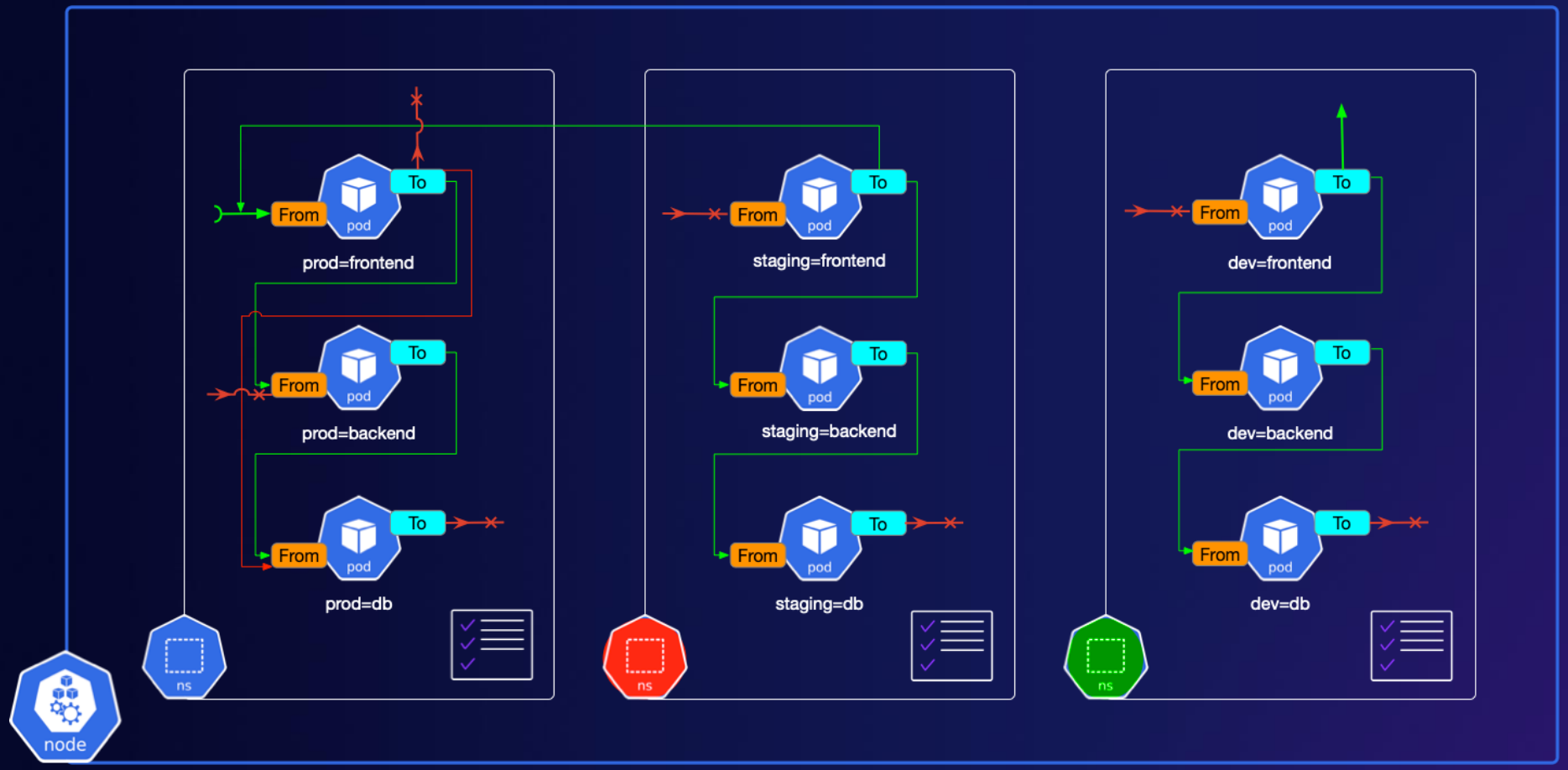


IAM Role for Service Account(IRSA)



- OIDC IdP 등록
- IAM Role 생성
- IAM Policy 할당
- K8S SA 생성
- SA Annotation 추가

Network Policy



특정 Pod, Namespace, IP 와의 통신을 제어

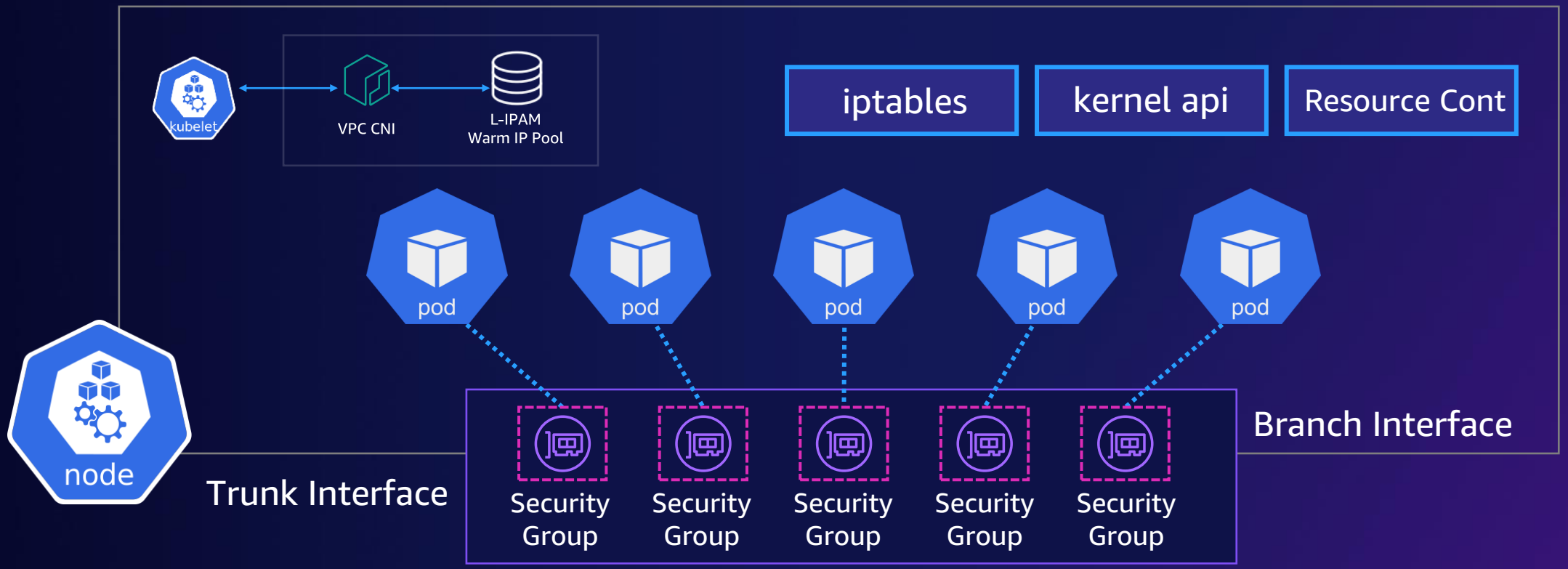
policyTypes
Ingress or Egress

ipBlock - CIDR
ports – Protocol + Port 번호

namespaceSelector
대상 Namespace 를 Label 로 지정

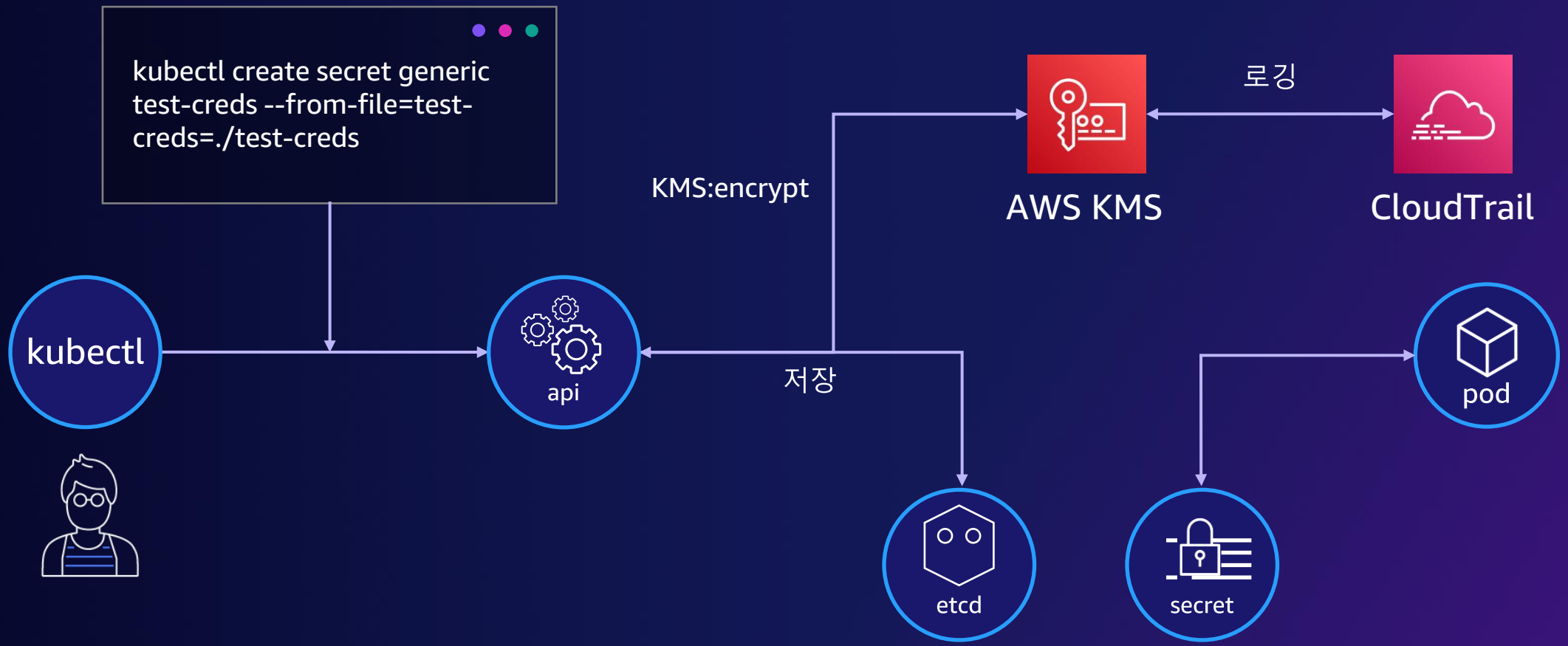
podSelector
대상 Pod 를 Label 로 지정

Security Group for Pod

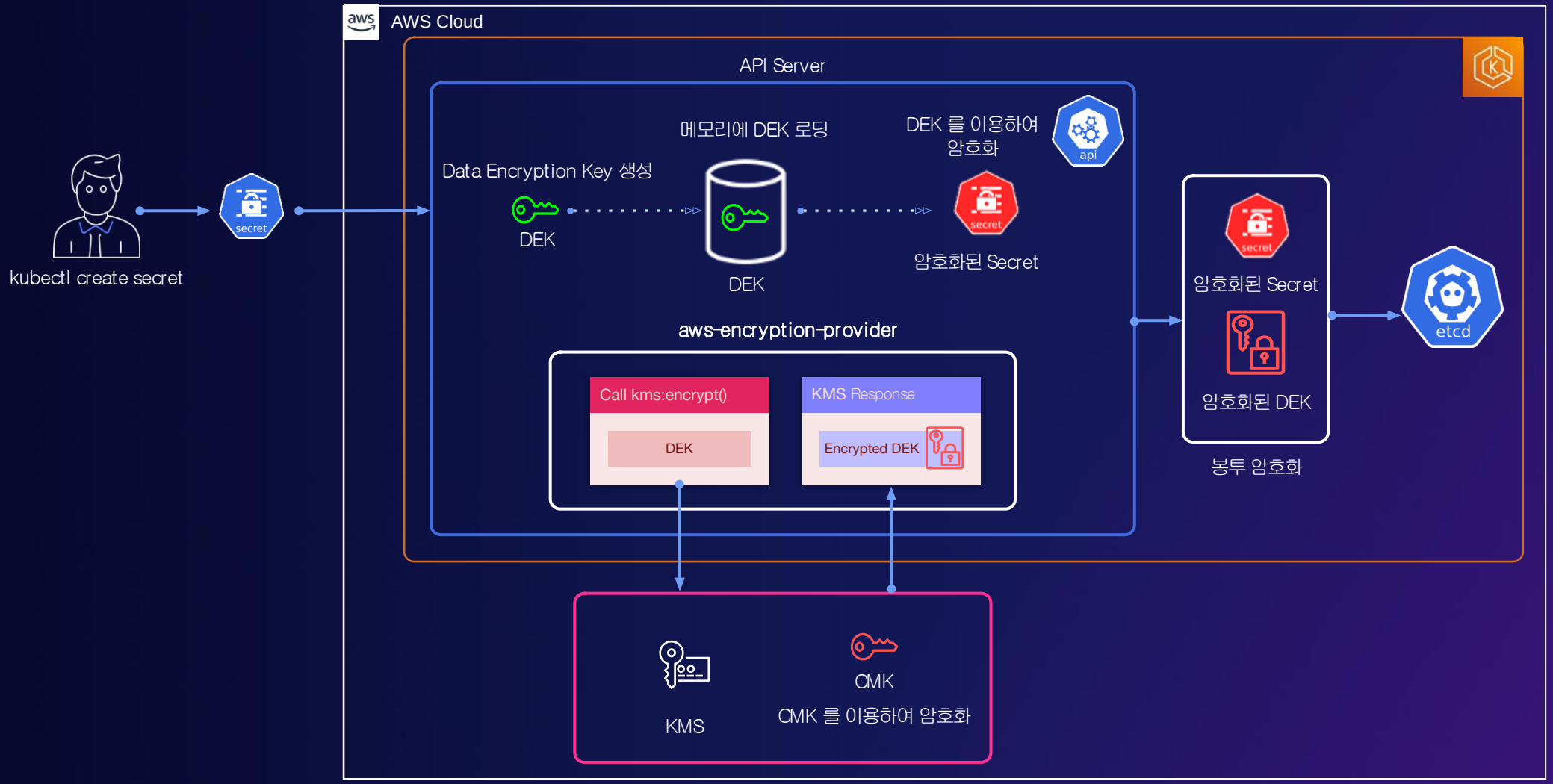


- Node 내의 Pod 간 Network Policy와 별개로 동작
- 지원 가능한 Interface Type, 생성 가능한 최대 Pod 등 고려

Secret - KMS 를 활용한 암호화

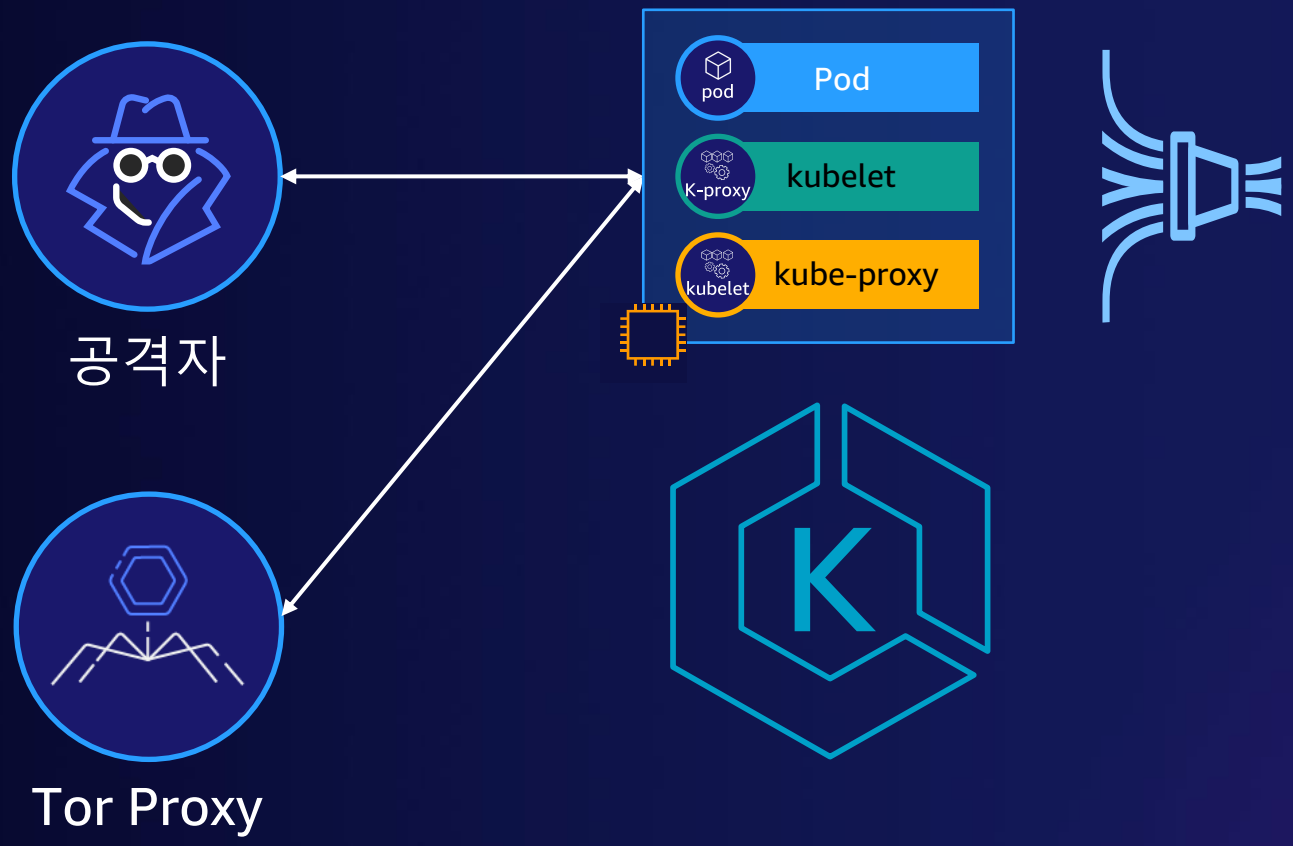


Secret - KMS 를 활용한 암호화



Secret - KMS 를 활용한 복호화

GuardDuty 에서의 위협 탐지



Amazon GuardDuty

EKS Audit Log

- 의심스런 외부 호스트와의 통신
- Dashboard 외부 노출
- 권한 상승 취약점
- 민감한 파일 경로 마운팅

2022년 2월 기준 27개 위협탐지 유형 제공

보안 태세에 대한 높은 기준 수립

지속적인 위협탐지, 최적화된 업무 흐름, 문제 해결을 위한 최소화 등을 위한 AWS 보안 서비스의 통합

SecOps 및 DevOps 팀이 가시성을 통합하고 응답을 자동화하여 클라우드 보안에서 운영 우수성을 달성하도록 지원

클라우드 애플리케이션을 위한 가시성 및 최적화

클라우드 기반 워크로드 및
어플리케이션을 위한
운영효율성 증대

가시성 개선

보안 향상

클라우드 환경 보호에서 운영 효율성 달성

- 위협 탐지 및 모니터링
- 중앙 집중화
- 보안 태세 평가 개선
- 취약점 관리 최적화
- 근본 원인 분석 간소화
- 민감한 데이터 검색 개선
- 워크플로 시작 및 기존 시스템으로 라우팅
- 중요한 발견의 우선 순위 지정
- 수정 자동화
- 배포 확장

여러분의 소중한 피드백을 기다립니다.

강연 종료 후, 강연 평가에 참여해 주세요!

감사합니다