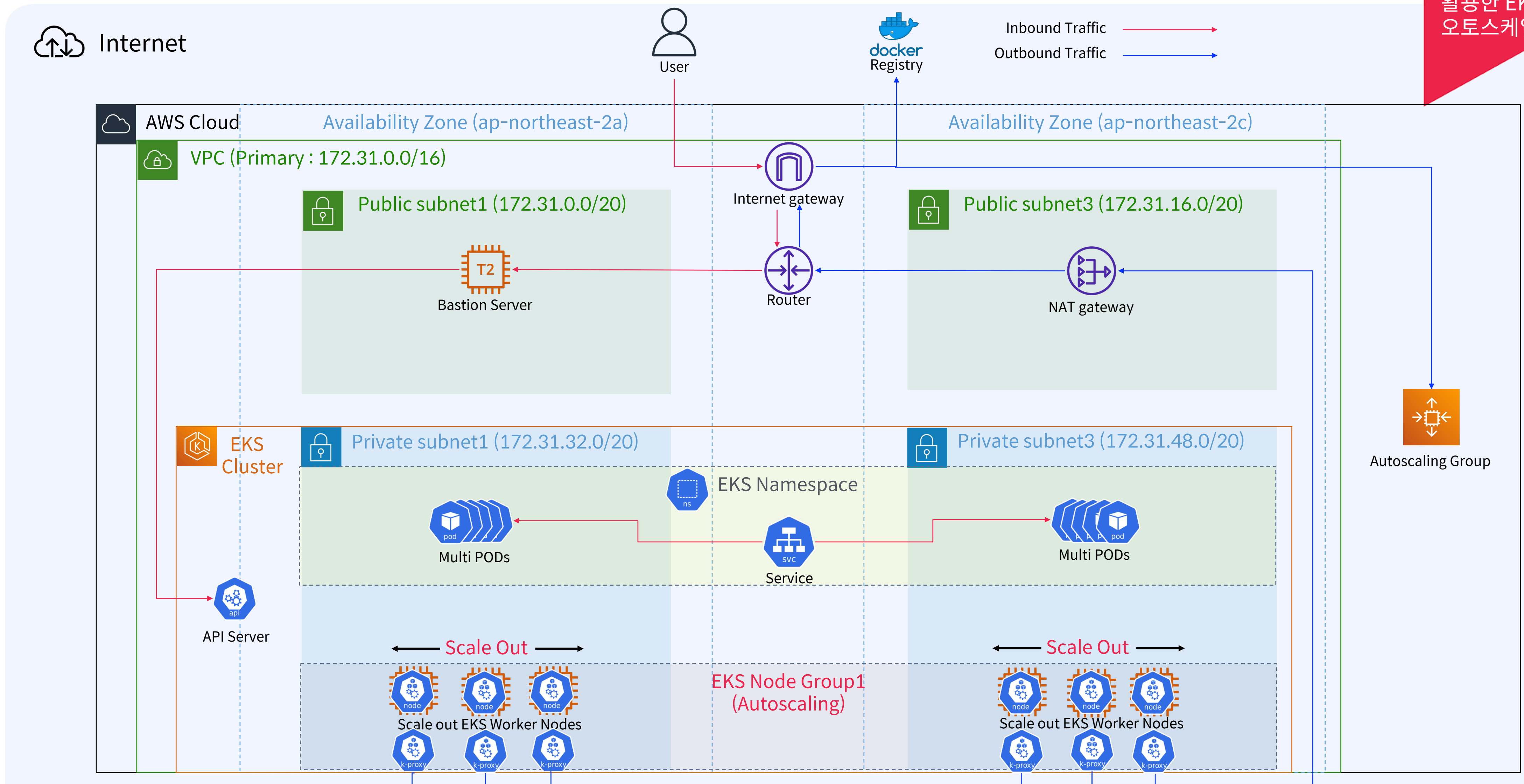


# 3 AWS EKS 기본 Plugin 구성 방법

## 07 Cluster Autoscaler를 활용한 EKS노드 오토스케일링

# AWS 네트워크 및 EKS 구성도 (Cluster Autoscaler 적용)

## 07. Cluster Autoscaler를 활용한 EKS노드 오토스케일링



# 사전 준비사항

**07.**  
Cluster  
Autoscaler를  
활용한 EKS노드  
오토스케일링

## 1. AWS Network 구성

- Public Subnet 2개, Private Subnet 2개
- NAT Gateway 1개

## 2. AWS EKS 구성

- EKS Cluster 1개, EKS NodeGroup1개(2개 Worker Node 생성)
- Bastion Server에서 EKS 접속

## 실습 내용

### 07. Cluster Autoscaler를 활용한 EKS노드 오토스케일링

#### 순서

1. AWS Autoscaling Group내 **TAG** 설정
2. Terraform으로 **IAM Policy** 적용
3. eksctl로 **IAM Role** 및 **EKS Service Account** 적용
4. Cluster **Autoscaler** 배포 및 설정
5. 10개 POD 배포 및 EKS Worker Node 자동 **Scale-out** 확인
6. POD 2개로 축소 및 EKS Worker Node 자동 **Scale-in** 확인

#### 실습 예제코드 경로

Chapter03 > Ch03\_07-cluster-autoscaler

# 1. AWS Autoscaling Group내 TAG 설정

## 1. Cluster Autoscaler Tag1

- Key : **k8s.io/cluster-autoscaler/<EKS Cluster명>**
- Value : **owned**

## 2. Cluster Autoscaler Tag2

- Key : **k8s.io/cluster-autoscaler/enabled**
- Value : **TRUE**

## 2. Terraform으로 IAM Policy 적용

**07.**  
Cluster  
Autoscaler를  
활용한 EKS노드  
오토스케일링

### 2.1 IAM Policy를 Terraform으로 생성

- IaC 경로 : **Ch03\_07-cluster-autoscaler > terraform-codes > iam-policy.tf**

### 2.2 terraform 명령어 실행

**\$ terraform init**

**\$ terraform plan**

**\$ terraform apply**

## 3. eksctl로 IAM Role 및 EKS Service Account 적용

**07.**  
Cluster  
Autoscaler를  
활용한 EKS노드  
오토스케일링

### 3.1 IAM Role 및 EKS Service Account 적용

```
$ eksctl create iamserviceaccount \  
  --name cluster-autoscaler \  
  --namespace kube-system \  
  --cluster <EKS Cluster명> \  
  --attach-policy-arn arn:aws:iam::<AWS 계정 ID>:policy/<EFS IAM Policy명> \  
  --approve \  
  --override-existing-serviceaccounts \  
  --region ap-northeast-2
```

## 4. Cluster Autoscaler 배포 및 설정

### 1.4 Cluster Autoscaler Manifest 확인

- 파일 경로 : **Ch03\_06-csi-file-pv > k8s-manifests > cluster-autoscaler.yaml**

### 1.5 Cluster Autoscaler 배포 실행 명령어

**\$ kubectl apply -f cluster-autoscaler.yaml**



## 5. 10개 POD 배포 및 EKS Worker Node 자동 Scale-out 확인

### 5.1 Namespace 구성 명령어

```
$ kubectl create namespace test-ca
```

### 5.2 예제 Deployment 배포

- 파일 경로 : Ch03\_07-cluster-autoscaler > k8s-manifests > deploy-pod.yaml
- 예제 POD 배포 : `kubectl create -f ./deploy-pod.yaml`

### 5.3 POD 및 Worker Node 개수 실시간 확인

```
$ kubectl get po -n test-ca -w
```

```
$ kubectl get no -w
```

## 6. POD 2개로 축소 및 EKS Worker Node 자동 Scale-in 확인

**07.**  
Cluster  
Autoscaler를  
활용한 EKS노드  
오토스케일링

### 6.1 POD 2개로 축소 명령어

```
$ kubectl scale deploy deploy-pod --replicas=2 -n test-ca
```

### 6.2 POD 및 Worker Node 개수 확인 실시간 확인

```
$ kubectl get po -n test-ca -w
```

```
$ kubectl get no -w
```