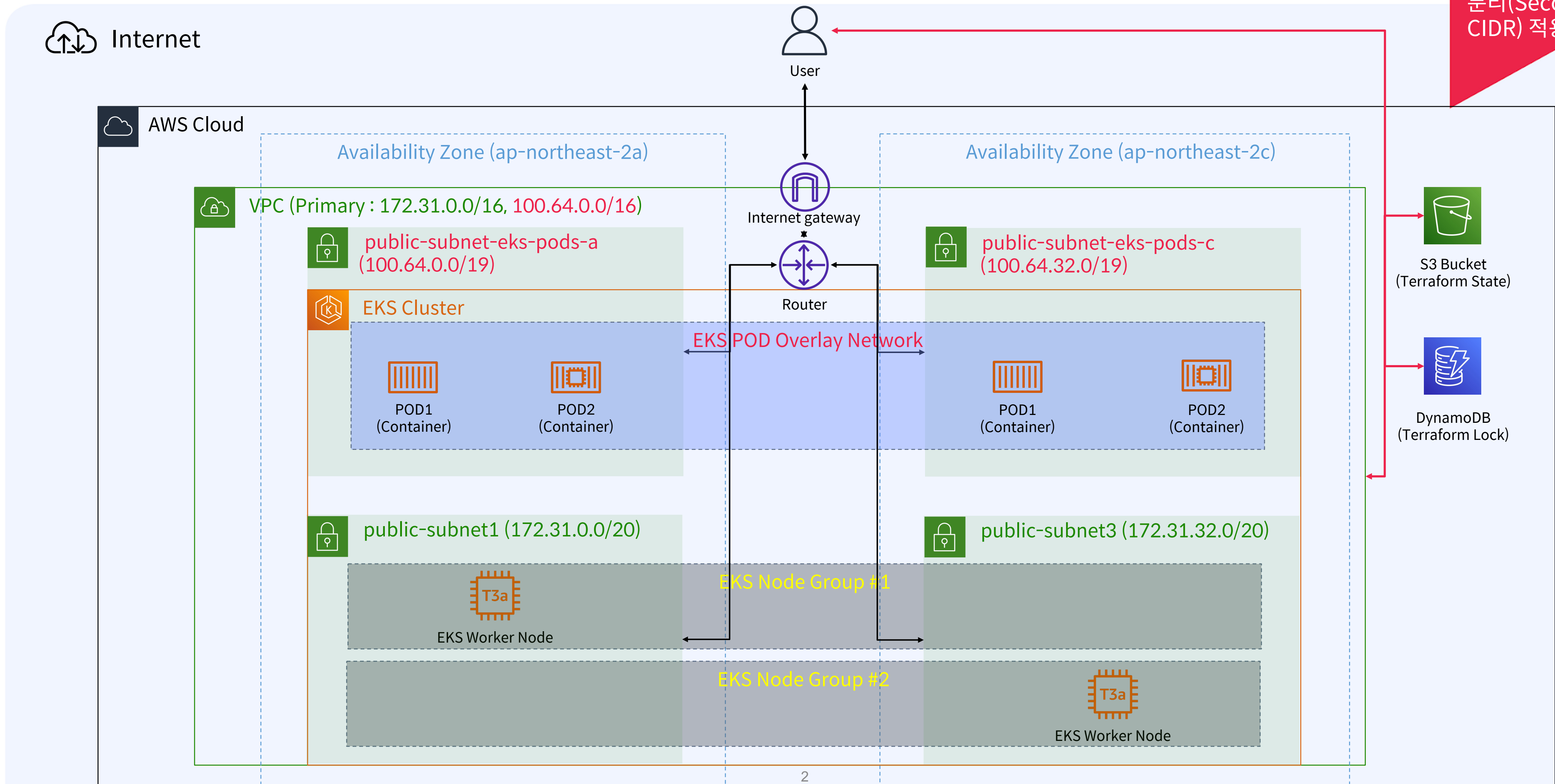


3 AWS EKS 기본 Plugin 구성 방법

02 EKS Node 및 POD IP 대역 분리(Secondary CIDR) 적용

AWS 네트워크 및 EKS 구성도 (VPC Secondary CIDR 적용)

02.
EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용



사전 준비사항

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

1. 로컬PC에 Terraform 설치
2. Terraform Backend 구성 완료
3. Terraform State에 VPC, Subnet, IGW, Route 포함 완료
4. Terraform으로 EKS까지 프로비저닝 완료
5. Terraform State에 EKS Cluster, Nodegroup, Autoscaling 포함 완료
6. 로컬PC에서 Kubectl로 EKS 접속 가능한 수준까지 준비

실습 내용

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

순서

1. VPC에 **Secondary CIDR** 추가
2. Terraform에 **Secondary CIDR Subnet, Route, Security Group 코드** 추가
3. Terraform **프로비저닝 실행** 및 확인
4. CRD, **ENIConfig** 및 **AWS CNI Config** 설정
5. EKS **Node 수동 전환** 및 POD에 **Secondary CIDR 대역 IP 할당** 확인

실습 예제코드 경로

Chapter03 > Ch03_02-vpc-secondary-cidr

1. VPC에 Secondary CIDR 추가

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

1. AWS Management Console에서 추가

- AWS 경로 : **VPC > 작업 > CIDR 편집 > 새 IPv4 CIDR 추가**
- 설정 내역 : **100.64.0.0/16**

2. Terraform에 Secondary CIDR Subnet, Route, Security Group 코드 추가

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

1. Terraform에 Secondary CIDR Subnet 설정

- 예제코드 경로 : [Ch03_02-vpc-secondary-cidr](#) > [terraform-codes](#) > [subnet.tf](#)

2. Terraform에 Secondary CIDR Route Table 설정

- 예제코드 경로 : [Ch03_02-vpc-secondary-cidr](#) > [terraform-codes](#) > [route_table.tf](#)

3. Terraform에 Secondary CIDR Security Group 설정

- 예제코드 경로 : [Ch03_02-vpc-secondary-cidr](#) > [terraform-codes](#) > [security-group.tf](#)

3. Terraform 프로비저닝 실행 및 확인 #1

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

1. 프로비저닝을 통한 생성 및 확인

\$ terraform init

\$ terraform plan

\$ terraform apply

3. Terraform 프로비저닝 실행 및 확인 #2

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

2. AWS Management Console에서 확인

- Subnet : **VPC** > **서브넷**
- Routing Table : **VPC** > **라우팅 테이블**
- Security Group : **EC2** > **보안 그룹**

4. CRD, ENIConfig 및 AWS CNI Config 설정

1. CRD 및 ENIConfig를 EKS Cluster에 배포

- 예제코드 경로 : `Ch03_02-vpc-secondary-cidr > k8s-manifests > eni-configs`
- 배포 명령어 : `kubectl create -f .`

2. AWS CNI Config 설정 (EKS)

- 커스텀 CNI 플러그인 구성 : `kubectl set env daemonset aws-node -n kube-system AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG=true`
- Worker Node 식별용 ENIConfig Label 추가 : `kubectl set env daemonset aws-node -n kube-system ENI_CONFIG_LABEL_DEF=failure-domain.beta.kubernetes.io/zone`

5. EKS Node 수동 전환 및 POD에 Secondary CIDR 대역 IP 할당 확인 #1

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

1. AWS Management Console에서 EKS Node Group 수동 전환

- Node Group 설정 경로 : EKS > <EKS명> > 구성 > 컴퓨팅 > 노드 그룹 선택 > 편집
- 최대 크기 및 원하는 크기를 2로 변경, 변경후 변경 사항 저장

2. 추가 Worker Node 생성 확인 및 POD 상태 확인 명령어 (EKS)

- Worker Node 생성 확인 명령어 : `kubectl get nodes`
- POD 상태 확인 명령어 : `kubectl get pods`

3. POD Restart 및 Scale out 명령어 (EKS)

- POD Restart 명령어 : `kubectl rollout restart deploy <Deployment명>`
- POD Scale out 명령어 : `kubectl scale deploy <Deployment명> --replicas=<증가수량>`

5. EKS Node 수동 전환 및 POD에 Secondary CIDR 대역 IP 할당 확인 #2

02.

EKS Node 및
POD IP 대역
분리(Secondary
CIDR) 적용

4. POD에 Secondary CIDR IP 할당 확인 명령어 (EKS)

- POD IP 할당 확인 명령어 : `kubectl get pods -o wide`

5. AWS Management Console에서 EKS Node Group 수동 전환 마무리 작업

- Node Group 설정 경로 : **EKS** > <**EKS명**> > **구성** > **컴퓨팅** > **노드 그룹 선택** > **편집**
- **최대 크기 및 원하는 크기를 1로 변경**, 변경후 변경 사항 저장
- (여기서 Secondary CIDR이 인식이 안되는 초기 Worker Node는 회수, 삭제됨)