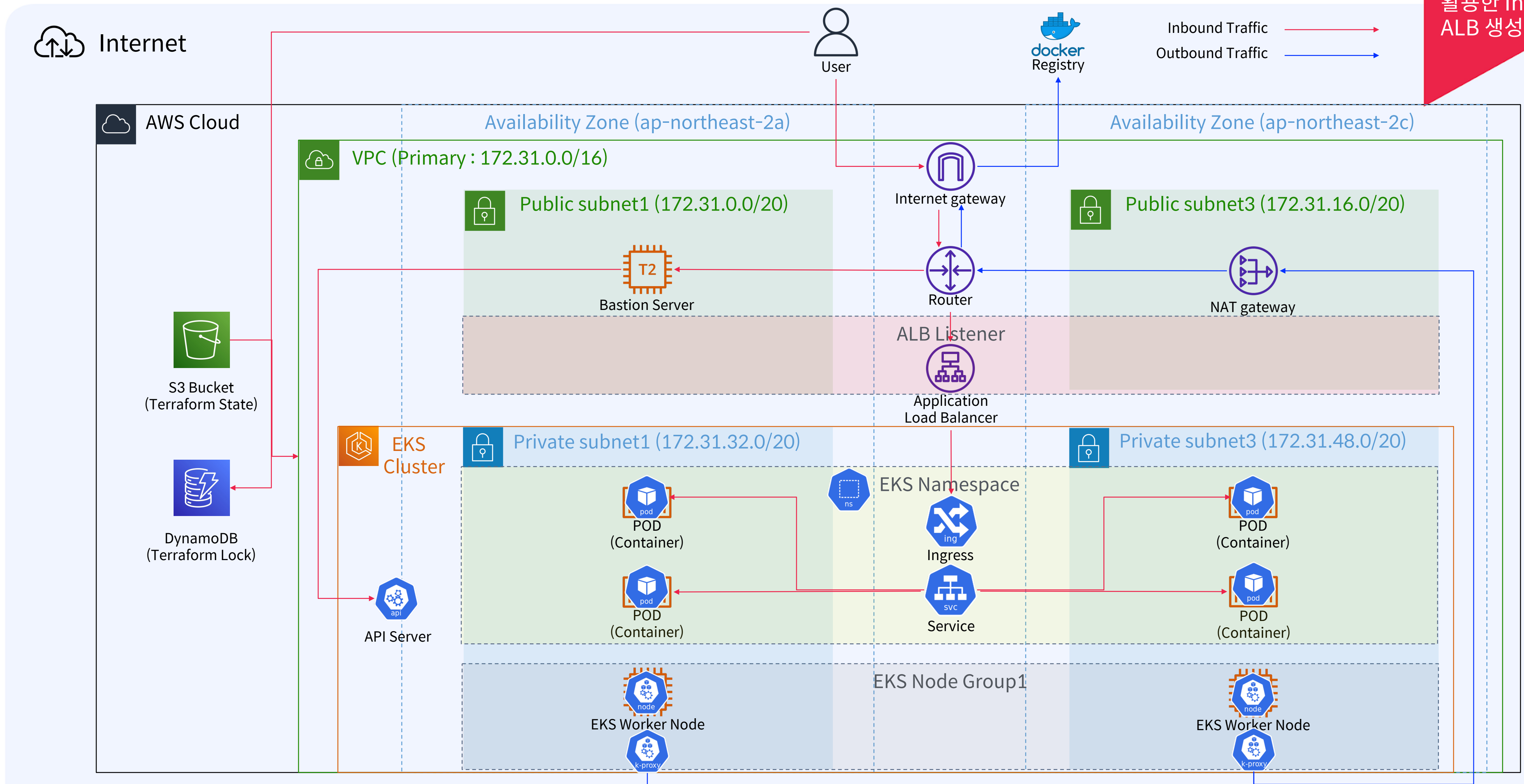


3 AWS EKS 기본 Plugin 구성 방법

03 AWS ALB Controller를 활용한 Ingress ALB 생성

AWS 네트워크 및 EKS 구성도 (Ingress ALB 적용)

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성



사전 준비사항 #1

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

1. Terraform Backend 구성

- S3 Bucket 및 DynamoDB Table 생성

2. AWS Network 구성

- VPC 1개, Internet Gateway 1개
- Public Subnet 2개, **Private Subnet 2개**
- Public Subnet Route Table 2개

3. AWS EKS 구성

- EKS Cluster 1개, EKS NodeGroup 1개 (2개 Worker Node 생성)

사전 준비사항 #2

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

4. Terraform에 Elastic IP 및 NAT Gateway 설정

- 코드 경로 : `Ch03_03-ingress-alb > terraform-codes > eip.tf`
`Ch03_03-ingress-alb > terraform-codes > nat_gateway.tf`

5. Terraform에 Route Table 설정 (Private Subnet 2개)

- 코드 경로 : `Ch03_03-ingress-alb > terraform-codes > route_tables.tf`
`Ch03_03-ingress-alb > terraform-codes > route_table_associations.tf`

사전 준비시 필수 적용사항

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

1. 전체 Subnet 적용 Tag

- Key : `kubernetes.io/cluster/<EKS Cluster명>`
- Value : `shared`

2. Public Subnet만 적용하는 Tag

- Key : `kubernetes.io/role/elb`
- Value : `1`

3. Bastion 접속용 Key Pair 생성 (.pem)

- Key Pair명 : `test-kp-bastion`

실습 내용

03. AWS ALB Controller를 활용한 Ingress ALB 생성

순서

1. Terraform에 **Bastion(EC2), Security Group, IAM Role** 코드 추가 및 프로비저닝
2. **Bastion** 접속, **EKS Kubeconfig** 설정, **Kubectl**로 **EKS** 접속
3. **Namespace** 구성 및 예제 **Deployments** 배포(K8s Manifest)
4. **AWS ALB Controller** 설치
5. **Ingress Annotation** 설정 및 **Ingress** 배포
6. **Application Load Balancer(ALB)** 자동 생성 확인 및 **로컬 PC 웹브라우저**로 접속 확인

실습 예제코드 경로

Chapter03 > Ch03_03-ingress-alb

1. Terraform에 Bastion(EC2), Security Group, IAM Role 코드 추가 및 프로비저닝

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

1.1 Terraform에 Bastion Server (AWS EC2 VM) 설정

- 예제코드 경로 : [Ch03_03-ingress-alb](#) > [terraform-codes](#) > [ec2.tf](#)

1.2 Terraform에 Security Group 설정

- 예제코드 경로 : [Ch03_03-ingress-alb](#) > [terraform-codes](#) > [security-groups.tf](#)

1.3 Terraform에 EC2 VM용 IAM Instance Role 설정

- 예제코드 경로 : [Ch03_03-ingress-alb](#) > [terraform-codes](#) > [iam-roles.tf](#)

2. Bastion 접속, EKS Kubeconfig 설정, Kubectl로 EKS 접속

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

2.1 Bastion 접속 명령어

```
$ ssh -i <생성한 Key pair(.pem)> ubuntu@<Bastion Public DNS 주소>
```

2.2 EKS Kubeconfig 설정

```
$ aws eks update-kubeconfig --region <Region명> --name <EKS Cluster명>
```

2.3 Kubectl로 EKS 접속

```
$ kubectl get no
```

```
$ kubectl get po -A
```


3. Namespace 구성 및 예제 Deployments 배포(K8s Manifest)

3.1 Namespace 구성 명령어

\$ **kubectl create namespace test-ingress-alb**

3.2 예제 Deployments 배포

- K8s Manifest 경로 : **Ch03_03-ingress-alb > k8s-manifests**
- K8s Manifest 배포 명령어 : **kubectl create -f test-deployment-game.yaml**

4. AWS ALB Controller 설치 #1

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

4.1 eksctl 툴 설치 명령어 (리눅스)

```
$ curl --silent --location
```

```
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

```
$ sudo mv /tmp/eksctl /usr/local/bin
```

4.2 설치 확인

```
$ eksctl version
```

4. AWS ALB Controller 설치 #2

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

4.3 IAM Policy를 Terraform으로 생성

- IAM Policy 경로 : **Ch03_03-ingress-alb > iam-policy > alb-iam-policy.json**
- 복사, 붙여넣기 경로 : **Ch03_03-ingress-alb > terraform-codes > iam-roles.tf**

4.4 terraform 명령어 실행

\$ **terraform init**

\$ **terraform plan**

\$ **terraform apply**

4. AWS ALB Controller 설치 #3

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

4.5 IAM Role 및 EKS내 서비스 어카운트 생성

```
$ eksctl create iamserviceaccount \  
--cluster=<EKS Cluster명> \  
--namespace=kube-system \  
--name=aws-load-balancer-controller \  
--attach-policy-arn=arn:aws:iam::<AWS 계정 ID>:policy/AWSLoadBalancerControllerIAMPolicy \  
--override-existing-serviceaccounts \  
--approve
```

4. AWS ALB Controller 설치 #4

4.6 cert-manager 배포

- K8s Manifest 경로 : **Ch03_03-ingress-alb** > **k8s-manifests**
- K8s Manifest 배포 명령어 : **kubectl create -f cert-manager.yaml**

4.7 AWS ALB Controller 파일 설정 및 배포

- K8s Manifest 경로 : **Ch03_03-ingress-alb** > **k8s-manifests**
- 파일 설정 위치 : **alb-controller.yaml** > **--cluster-name=<EKS Cluster명>**
- K8s Manifest 배포 명령어 : **kubectl create -f alb-controller.yaml**
- 배포 확인 명령어 : **kubectl get deploy -n kube-system aws-load-balancer-controller**

5. Ingress Annotation 설정 및 Ingress 배포 #1

5.1 Ingress Annotation 설정

- K8s Manifest 경로 : **Ch03_03-ingress-alb** > **k8s-manifests**
- 파일 설정 위치 : **test-ingress.yaml**
- Ingress Annotation 설정내역

annotations:

kubernetes.io/ingress.class: **alb**

alb.ingress.kubernetes.io/scheme: **internet-facing**

alb.ingress.kubernetes.io/target-type: **ip**

alb.ingress.kubernetes.io/subnets: <**Public Subnet1 ID**>, <**Public Subnet2 ID**>

5. Ingress Annotation 설정 및 Ingress 배포 #2

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

5.2 Ingress 배포

- K8s Manifest 경로 : **Ch03_03-ingress-alb** > **k8s-manifests**
- K8s Manifest 배포 명령어 : **kubectl create -f test-ingress.yaml**

6. Application Load Balancer 자동 생성 확인 및 로컬 PC 웹브라우저로 접속 확인 #1

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

6.1 Application Load Balancer(ALB) 자동 생성 확인

- AWS Managemnt Console 확인 경로 : **EC2 > 로드밸런서**
- 확인내용
 - * 유형 : **application**
 - * 체계 : **internet-facing**
 - * IP 주소 유형 : **ipv4**
 - * 가용 영역 : **<Public Subnet1 ID>, <Public Subnet2 ID>**

6. Application Load Balancer 자동 생성 확인 및 로컬 PC 웹브라우저로 접속 확인 #2

03.
AWS ALB
Controller를
활용한 Ingress
ALB 생성

6.2 로컬 PC 웹브라우저로 접속 확인

- ALB DNS 주소 확인 : 로컬 PC 웹브라우저에서 다음의 URL로 접속 확인

<http://<ALB DNS 주소 확인>:80>