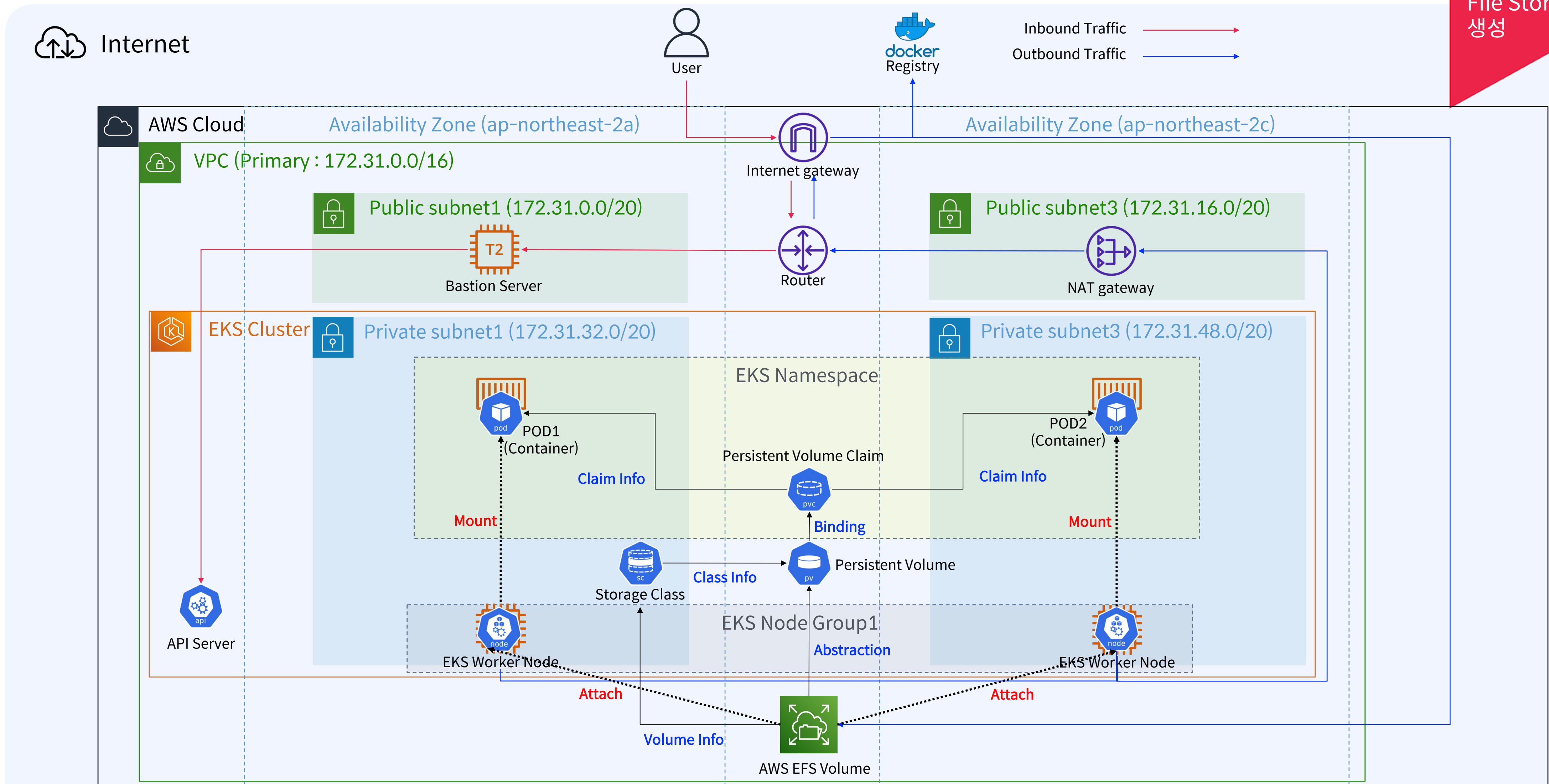


3 AWS EKS 기본 Plugin 구성 방법

06 AWS EFS CSI Driver를 활용한 File Storage PV 생성

실습 아키텍처 (EFS CSI 및 File PV 적용)

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성



실습 아키텍처 구성사항 #1

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

1. AWS Network 구성

- VPC1개, Public Subnet 2개, Private Subnet 2개
- Internet Gateway 1개, NAT Gateway 1개

2. AWS EKS 구성

- EKS Cluster 1개, EKS NodeGroup1개(2개 Worker Node 생성)
- EC2 Bastion VM 1개 (EKS 접속용)

실습 아키텍처 구성사항 #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

3.1 Terraform으로 Backend생성

- IaC 경로 : **Ch03_06-csi-file-pv > terraform-backend**

3.2 terraform 명령어 실행

\$ terraform init

\$ terraform plan

\$ terraform apply

실습 아키텍처 구성사항 #3

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

4.1 Terraform으로 AWS 클라우드 아키텍처 구성 경로

- IaC 경로 : **Ch03_06-csi-file-pv > terraform-codes**

4.2 Bastion 서버용 Key Pair 생성

- **AWS Management 콘솔 > EC2 > 네트워크 및 보안 > 키 페어 > 키 페어 생성 >**
"test-kp-bastion" 이름의 RSA방식의 키 페어 생성 및 다운로드

실습 아키텍처 구성사항 #4

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

4.3 terraform 명령어 실행하여 전체 아키텍처 구성

\$ terraform init

\$ terraform plan

\$ terraform apply

4.4 아키텍처 구성 이후 Bastion 서버 연결 방법

- AWS Management 콘솔 > EC2 > Bastion VM 선택 > 우측 상단 "연결 버튼" 클릭 > SSH 연결 방식에 나와있는 방식대로 수행하여 연결

실습 아키텍처 구성사항 #5

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

5.1 Bastion VM에서 kubectl 다운로드

```
$ curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.23.7/2022-06-29/bin/linux/amd64/kubectl
```

5.2 Bastion VM에서 kubectl 실행 권한 부여 및 bin PATH로 이동

```
$ chmod +x kubectl && sudo mv kubectl /usr/local/bin/
```

5.3 kubectl 버전 확인

```
$ kubectl version
```

실습 아키텍처 구성사항 #6

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

6.1 Bastion VM에서 awscli 설치

```
$ sudo apt update
```

```
$ sudo apt install -y awscli
```

6.2 Bastion VM에서 aws configure 설정

- 본인의 AWS 계정의 Access Key와 Secret Key, Region 설정 (Output : json)

```
$ aws configure
```

5.3 aws configure 설정 확인

```
$ aws configure list
```


실습 아키텍처 구성사항 #7

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

7.1 Bastion VM에서 kubeconfig 설정

```
$ aws eks update-kubeconfig --name <생성된 EKS 클러스터명>
```

7.2 kubeconfig 파일 확인

- 홈 디렉토리에서 파일 존재 확인

```
$ cat ~/.kube/config
```

7.3 kubectl 동작 확인

```
$ kubectl get nodes; kubectl get pods -A
```

실습 아키텍처 구성사항 #8

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

8.1 Bastion VM에서 eksctl 다운로드

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

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

8.2 eksctl을 바이너리 Path로 이동

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

8.3 eksctl 동작 확인

```
$ eksctl version
```

실습 내용

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

순서

1. **AWS EFS CSI Driver** 설치
2. **Namespace** 구성, **Storage Class** 및 예제 **POD, PVC** 배포
3. **File Storage PV** 생성 및 **볼륨 마운트** 확인
4. 다른 Worker Node에 있는 **2개의 POD**에서 동시에 **데이터 Read/Write**
5. **2개의 POD** 삭제 및 **2개의 POD** 생성 후 동시 **파일 읽기** 수행

실습 예제코드 경로

Chapter03 > Ch03_06-csi-file-pv

1. AWS EFS CSI Driver 설치 #1

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

1.1 IAM Role 및 EKS내 서비스 어카운트 생성

```
$ eksctl create iamserviceaccount \  
  --name efs-csi-controller-sa \  
  --namespace kube-system \  
  --cluster <EKS Cluster명> \  
  --attach-policy-arn arn:aws:iam::<AWS 계정 ID>:policy/AmazonEKS_EFS_CSI_Driver_Policy \  
  --approve \  
  --override-existing-serviceaccounts \  
  --region ap-northeast-2
```

1. AWS EFS CSI Driver 설치 #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

1.2 Helm 설치

```
$ curl -L https://git.io/get_helm.sh | bash -s -- --version v3.8.2
```

1.3 Helm 설치후 버전 확인

```
$ helm version
```

1. AWS EFS CSI Driver 설치 #4

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

1.4 Helm 리포지토리 추가

```
$ helm repo add aws-efs-csi-driver https://kubernetes-sigs.github.io/aws-efs-csi-driver/
```

1.5 리포지토리 업데이트

```
$ helm repo update
```

1. AWS EFS CSI Driver 설치 #5

1.6 Helm Chart로 AWS EFS CSI Driver 설치

```
$ helm upgrade -i aws-efs-csi-driver aws-efs-csi-driver/aws-efs-csi-driver \
--namespace kube-system \
--set image.repository=602401143452.dkr.ecr.ap-northeast-
2.amazonaws.com/eks/aws-efs-csi-driver \
--set controller.serviceAccount.create=false \
--set controller.serviceAccount.name=efs-csi-controller-sa
```

1.7 AWS EFS CSI Driver 설치 확인

```
$ kubectl get pods -n kube-system | grep efs-csi-controller
```

2. Namespace 구성, Storage Class 및 예제 POD, PVC 배포 #1

06.

AWS EFS CSI
Driver를 활용한
File Storage PV
생성

2.1 Namespace 구성 명령어

```
$ kubectl create namespace test-csi-file-pv
```

2.2 Storage Class 배포

- K8s Manifest 경로 : Ch03_06-csi-file-pv > k8s-manifests
- Manifest 배포 명령어 : `kubectl create -f storageclass.yaml`

2. Namespace 구성, Storage Class 및 예제 POD, PVC 배포 #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

2.3 예제 POD, PVC 배포

- K8s Manifest 경로 : **Ch03_06-csi-file-pv > k8s-manifests**
- POD1 배포 : **kubectl create -f deploy-pod1.yaml**
- POD2 배포 : **kubectl create -f deploy-pod2.yaml**
- PVC 배포 : **kubectl create -f efs-pv-claim.yaml**

3. File Storage PV 생성 및 볼륨 마운트 확인 #1

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

3.1 File Storage PV 자동 생성 확인 명령어

```
$ kubectl get pv
```

```
$ kubectl get pvc -n test-csi-file-pv
```

3.2 볼륨 마운트 확인

```
$ kubectl exec <deploy-pod1, 2> -it -n test-csi-file-pv -- df -h
```

```
$ kubectl exec deploy-pod1 -it -n test-csi-file-pv -- cat /dpod1/out
```

```
$ kubectl exec deploy-pod2 -it -n test-csi-file-pv -- cat /dpod2/out
```

3. File Storage PV 생성 및 볼륨 마운트 확인 #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

3.3 File Storage PV에 연결된 AWS EFS 정보 확인 명령어

\$ **kubectl describe pv <PV명>**

- **VolumeHandle**값 확인
- 출력 형태 : **파일시스템ID:액세스포인트ID**

3.4 AWS에서 EFS 정보 확인

- 경로 : **AWS Management Console > EFS > 파일시스템**

4. 다른 Worker Node에 있는 2개의 POD에서 동시에 데이터 Read/Write #1

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

4.1 다른 Worker Node에 있는 2개의 POD 상태 확인

- 터미널창을 2개 띄움 (좌/우로 배열)
- 다음의 명령어를 이용해서 2개의 **POD의 상태** 및 배포된 Worker **Node의 현황** 확인

\$ **kubectl get pods -o wide**

4. 다른 Worker Node에 있는 2개의 POD에서 동시에 데이터 Read/Write #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

4.2 동시에 데이터 Read/Write

- 좌측 터미널창

```
$ kubectl exec deploy-pod1 -it -n test-csi-file-pv -- bash -c 'echo "[deploy-pod1 message]" >> /dpod1/out; tail -10 /dpod1/out'
```

- 우측 터미널창

```
$ kubectl exec deploy-pod2 -it -n test-csi-file-pv -- bash -c 'echo "[deploy-pod2 message]" >> /dpod2/out; tail -10 /dpod2/out'
```

5. 2개의 POD 삭제 및 2개의 POD 생성 후 동시 파일 읽기 수행 #1

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

5.1 POD 삭제

```
$ kubectl delete po <deploy-pod1, 2> -n test-csi-file-pv
```

5.2 볼륨 보존 확인

```
$ kubectl get pv
```

- AWS 확인 경로 : AWS Management Console > EFS > 파일시스템

5. 2개의 POD 삭제 및 2개의 POD 생성 후 동시 파일 읽기 수행 #2

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

5.3 POD 재생성

- K8s Manifest 경로 : **Ch03_06-csi-file-pv** > **k8s-manifests**
- POD1 배포 : **kubectl create -f deploy-pod1.yaml**
- POD2 배포 : **kubectl create -f deploy-pod2.yaml**

5. 2개의 POD 삭제 및 2개의 POD 생성 후 동시 파일 읽기 수행 #3

06.
AWS EFS CSI
Driver를 활용한
File Storage PV
생성

5.4 동시 파일 읽기 수행

```
$ kubectl get -it <deploy-pod1, 2> -n test-csi-file-pv -- df -h
```

```
$ kubectl get -it deploy-pod1 -n test-csi-file-pv -- cat /dpod1/out
```

```
$ kubectl get -it deploy-pod2 -n test-csi-file-pv -- cat /dpod2/out
```

- **deploy-pod1 message** 및 **deploy-pod2 message** 존재 확인