



Amazon EKS Storage

신 정 섭 (ssupp@amazon.com)

AWS Solutions Architect

Agenda

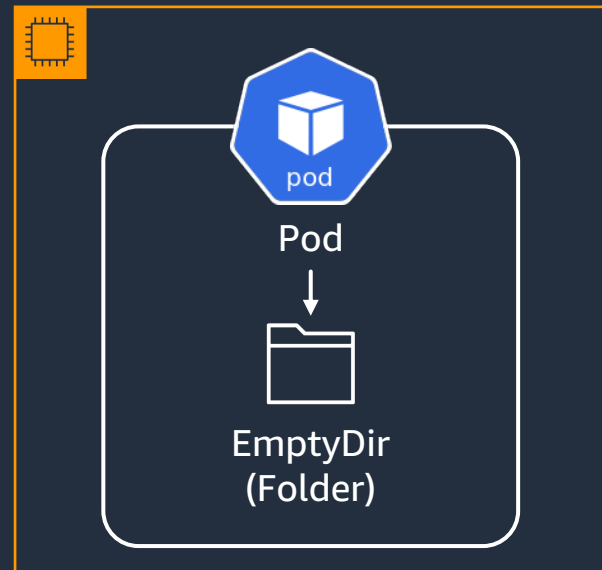
- Host Volume
- Persistent Volume
- AWS Storage

Host Volume

EmptyDir

```
apiVersion: v1
kind: Pod
metadata:
  name: emptydir-pd
spec:
  containers:
    - image: registry.k8s.io/test-webserver
      name: test-container
      volumeMounts:
        - mountPath: /cache
          name: cache-volume
  volumes:
    - name: cache-volume
      emptyDir: {}
```

Example EmptyDir

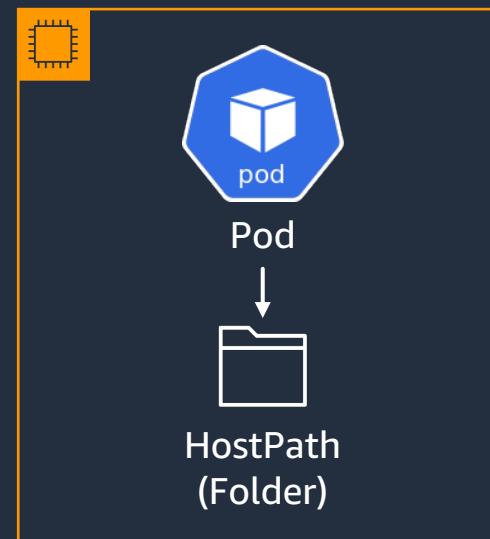


- Pod와 동일한 LifeCycle을 갖는 Empty Volume
- 파일 Caching 용도로 이용
- Node의 Disk, Memory (tmpfs) 이용 가능

HostPath

```
apiVersion: v1
kind: Pod
metadata:
  name: hostpath-pod
spec:
  containers:
  - image: registry.k8s.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-hostpath
      name: test-hostpath
  volumes:
  - name: test-hostpath
    hostPath:
      path: /data
  ...
```

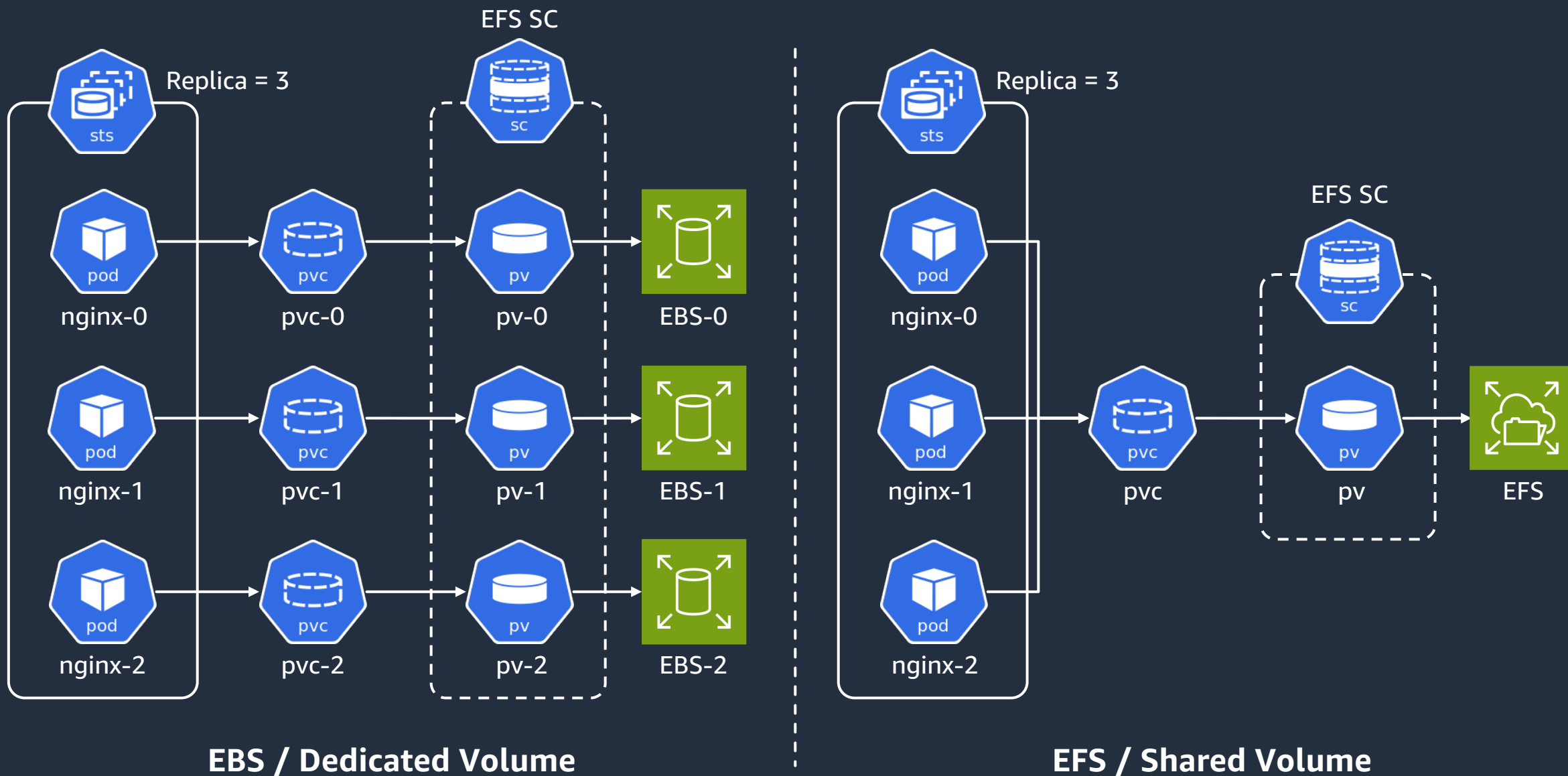
Example HostPath



- Host의 특정 Directory를 Container에게 Mount
- Node와 동일한 LifeCycle을 갖는 Volume
- 보안상 이용 권장 X

Persistant Volmue

PV, PVC, Storage Class Example



Kubernetes Storage Volume Objects



PV (Persistent Volume)

- 논리적인 하나의 Storage Volume을 의미
- Cloud Provider, On-premise 환경에 따라서 다양한 Type 이용 가능
 - AWS : EBS (Elastic Block Store), EFS (Elastic File System)
 - On-premise : Ceph, NFS
- Type에 따라서 Access Mode 제약 존재
 - ReadWriteOnce : 하나의 Node에서만 Read, Write 수행 가능
 - ReadOnlyMany : 다수의 Node에서 동시에 Read만 수행 가능
 - ReadWriteMany : 다수의 Node에서 동시에 Read, Write 수행 가능

Kubernetes Storage Volume Objects



PVC (Persistent Volume Claim)

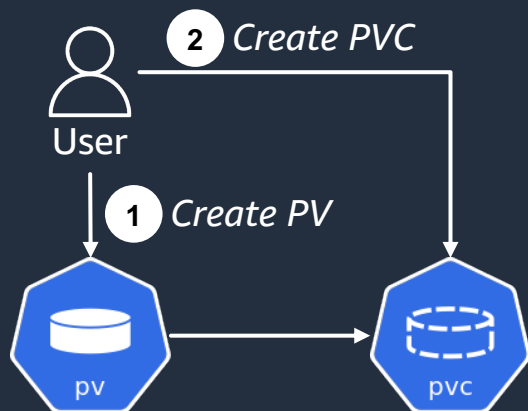
- Pod에서 이용하기 위한 **Storage Volume**의 요청을 의미
- **Pod**에 **PV**를 연결하기 위해서는 PVC 생성 필수



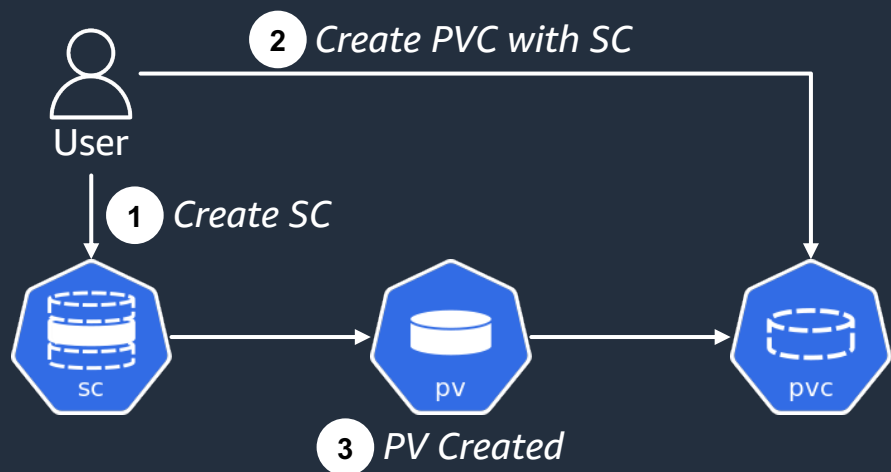
SC (StorageClass)

- **PV**를 동적으로 생성하기 위한 Class
- StorageClass를 이용하면 PVC 요청으로 PV도 자동 생성

Storage Provisioning

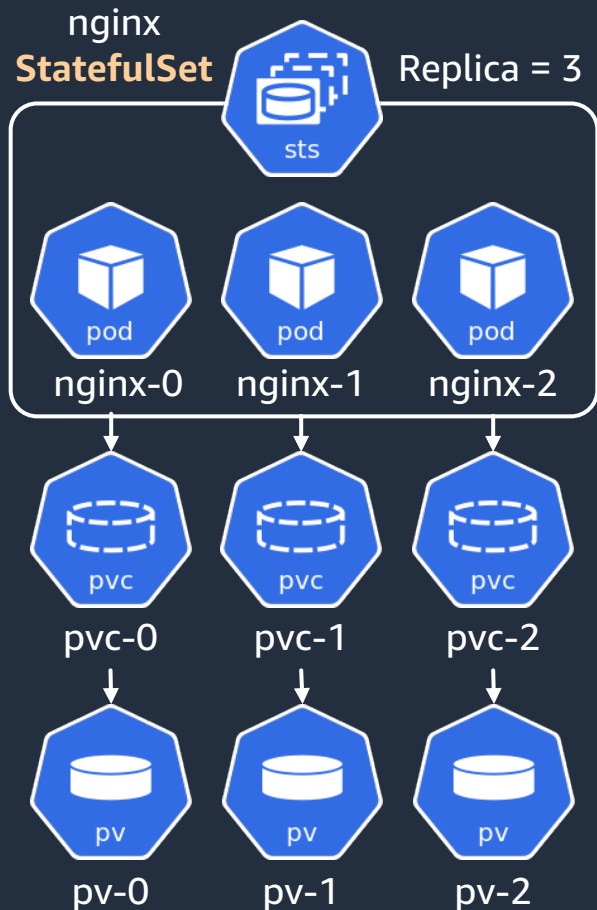


- Static Provisioning
 - 미리 생성된 PV를 Pod에 할당
 - 하나의 Storage Volume을 여러 Pod에서 공유할때 주로 이용 (ReadWriteMany)



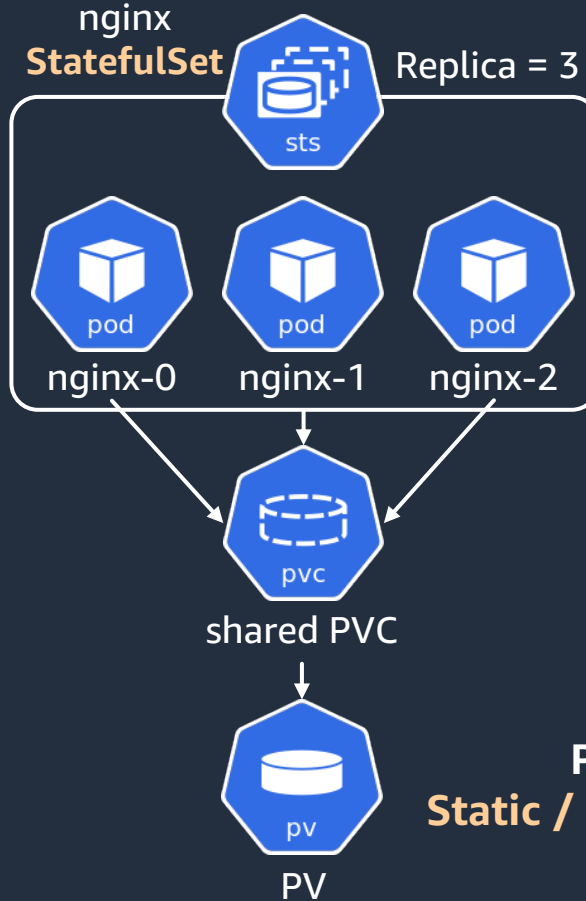
- Dynamic Provisioning (Storage Class)
 - 동적으로 PV를 생성하고 Pod에 할당
 - Storage Volume과 Pod가 1:1로 Mapping 되는 StatefulSet과 함께 주로 이용 (ReadWriteOnce)

StatefulSet



Pod : PVC = 1:N
Dynamic Provisioning

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nginx
...
spec:
  replicas: 3
  template: # Pod Template
...
volumeClaimTemplates:
- metadata:
  name: dedicated-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

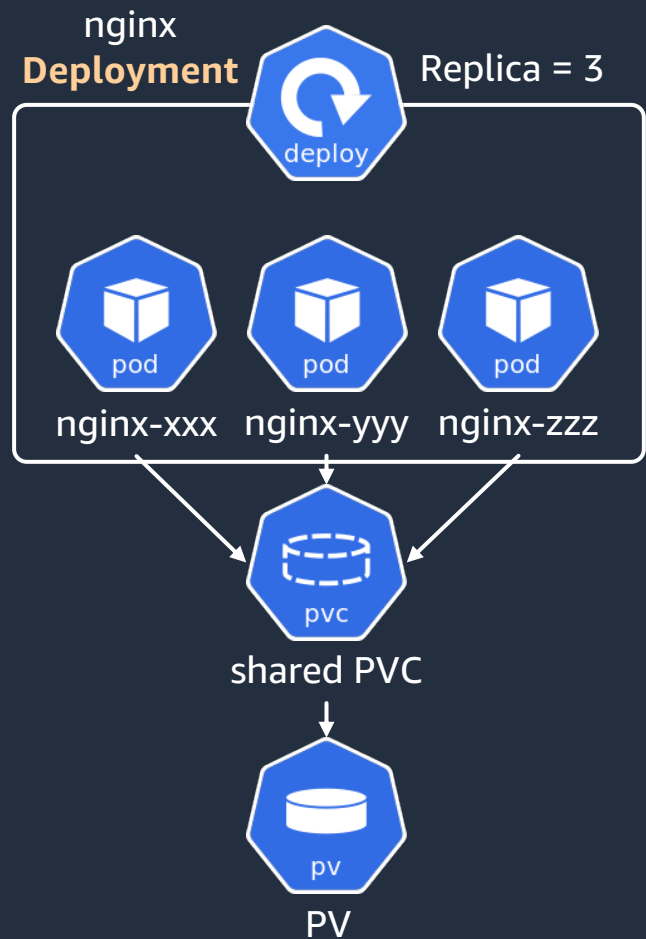


Pod : PVC = N:1
Static / Dynamic Provisioning

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nginx
...
spec:
  replicas: 3
  template: # Pod Template
...
volumes:
- name: efs
  persistentVolumeClaim:
    claimName: shared-pvc
```

- 동일한 Pod를 여러개 배포시 이용
- Pod : PVC = 1:N, N:1 Mapping 가능
- 고정된 Host Name

StatefulSet vs Deployment



Pod : PVC = N:1

Static / Dynamic Provisioning

	StatefulSet	Deployment
Hostname	Random Hostname (podname-[1,2,3..])	Fixed Hostname (podname-[random])
Pod, PVC Mapping	1:N, N:1	N:1

Storage Provisioning Example

```
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: test-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/Volumes/Data"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test-pvc
  namespace: test-namespace
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Static Provisiong

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: test
  name: test
  namespace: project-tiger
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      volumes:
        - name: data
          persistentVolumeClaim:
            claimName: test-pvc
      containers:
        - image: httpd:2.4.41-alpine
          name: container
          volumeMounts:
            - name: data
              mountPath: /tmp/test-data
```

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
reclaimPolicy: Retain
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Dynamic Provisiong

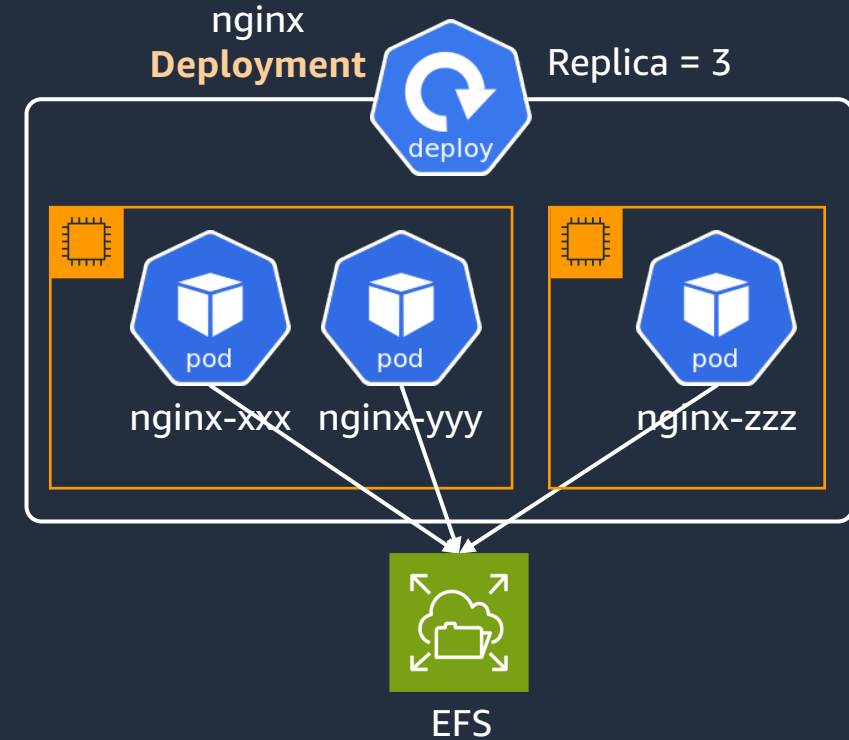
```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx
  serviceName: "nginx"
  replicas: 3
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: registry.k8s.io/nginx-slim:0.8
          ports:
            - containerPort: 80
          name: web
          volumeMounts:
            - name: www
              mountPath: /usr/share/nginx/html
      volumeClaimTemplates:
        - metadata:
            name: www
          spec:
            accessModes: [ "ReadWriteOnce" ]
            storageClassName: "standard"
            resources:
              requests:
                storage: 1Gi
```

Volume Type에 따른 Shared Volume Architecture



ReadWriteOnce Volume

→ 하나의 Node에만 모든 Pod가 생성



ReadOnlyMany, ReadWriteMany Volume

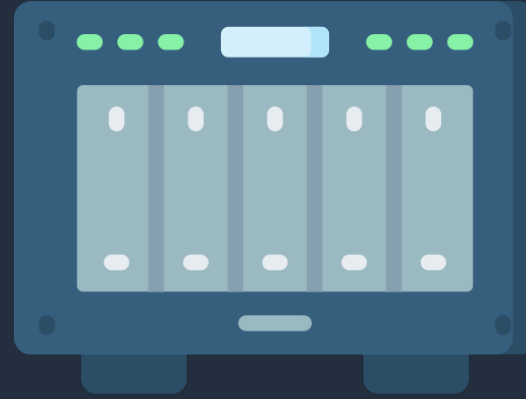
→ 다수의 Node에 Pod 생성 가능

AWS Storage

Physical Storage



Hard Disk, SSD, NVME



NAS

- PC/Server에 어떻게 연결할까?
- 이용하기 위한 과정은 어떻게 될까?
- 어떤 특징을 갖을까?

Amazon EKS Storage Type



Amazon EBS

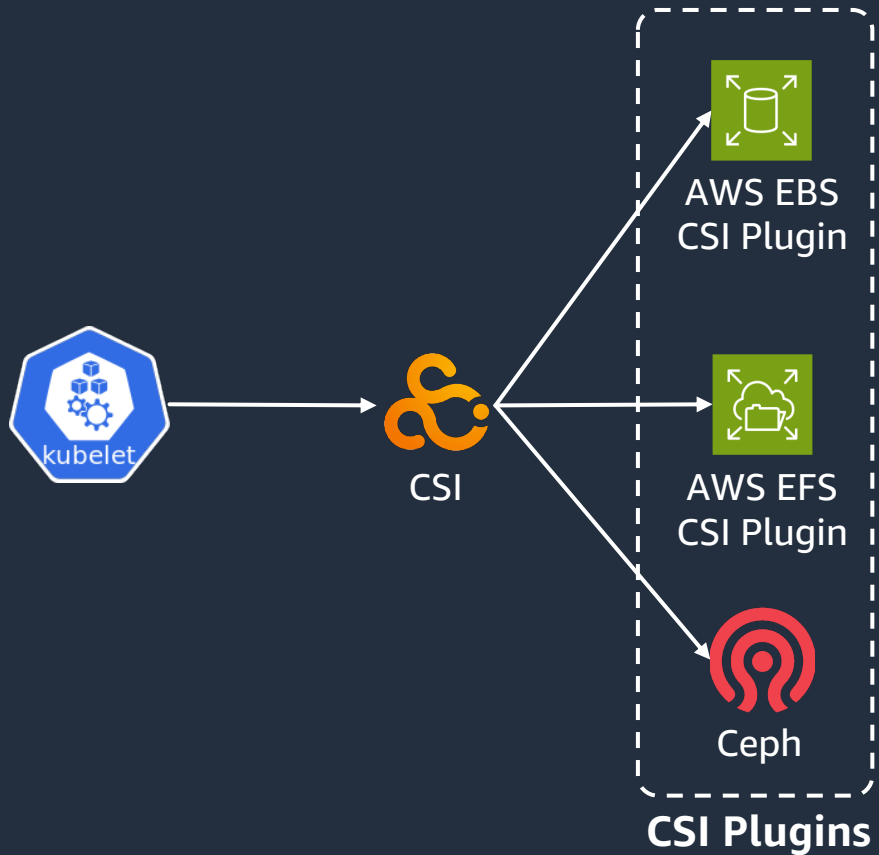
- Amazon EBS
 - Single-AZ Store, Single Access
 - 하나의 EC2 Instance에만 Attach 가능 (K8s ReadWriteOnce 지원)
 - 다양한 Volume Type 이용 가능 (gp2, gp3, io1, io2...)
 - AWS EBS Controller (CSI Plugin) 설치 필요
 - https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/ebs-csi.html



Amazon EFS

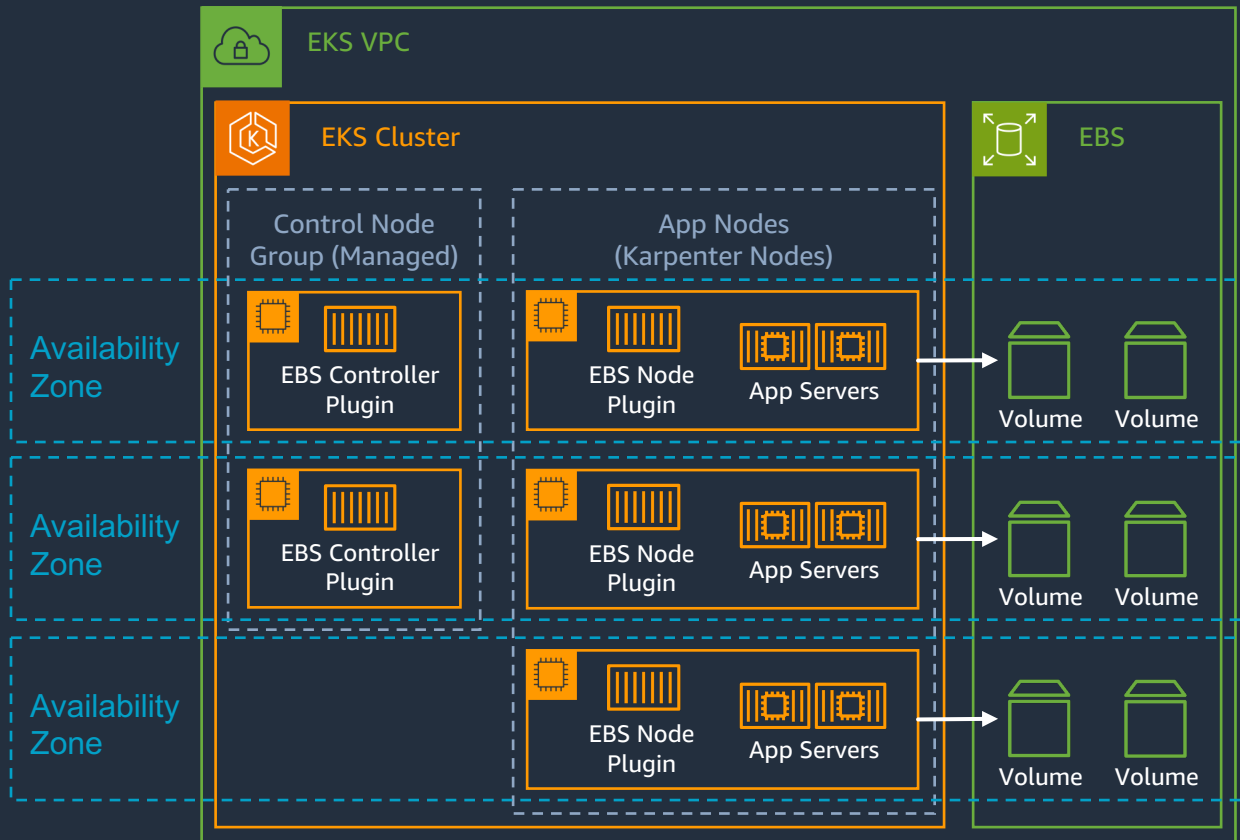
- Amazon EFS
 - Multi-AZ Store, Multiple Access, Scalability
 - 다수의 EC2 Instance에서 동시에 Mount 가능 (K8s ReadWriteMany 지원)
 - EFS Storage Class, Cross-replication 이용 가능
 - AWS EFS Controller (CSI Plugin) 설치 필요
 - <https://docs.aws.amazon.com/eks/latest/userguide/efs-csi.html>

Storage Volume, CSI (Container Storage Interface)



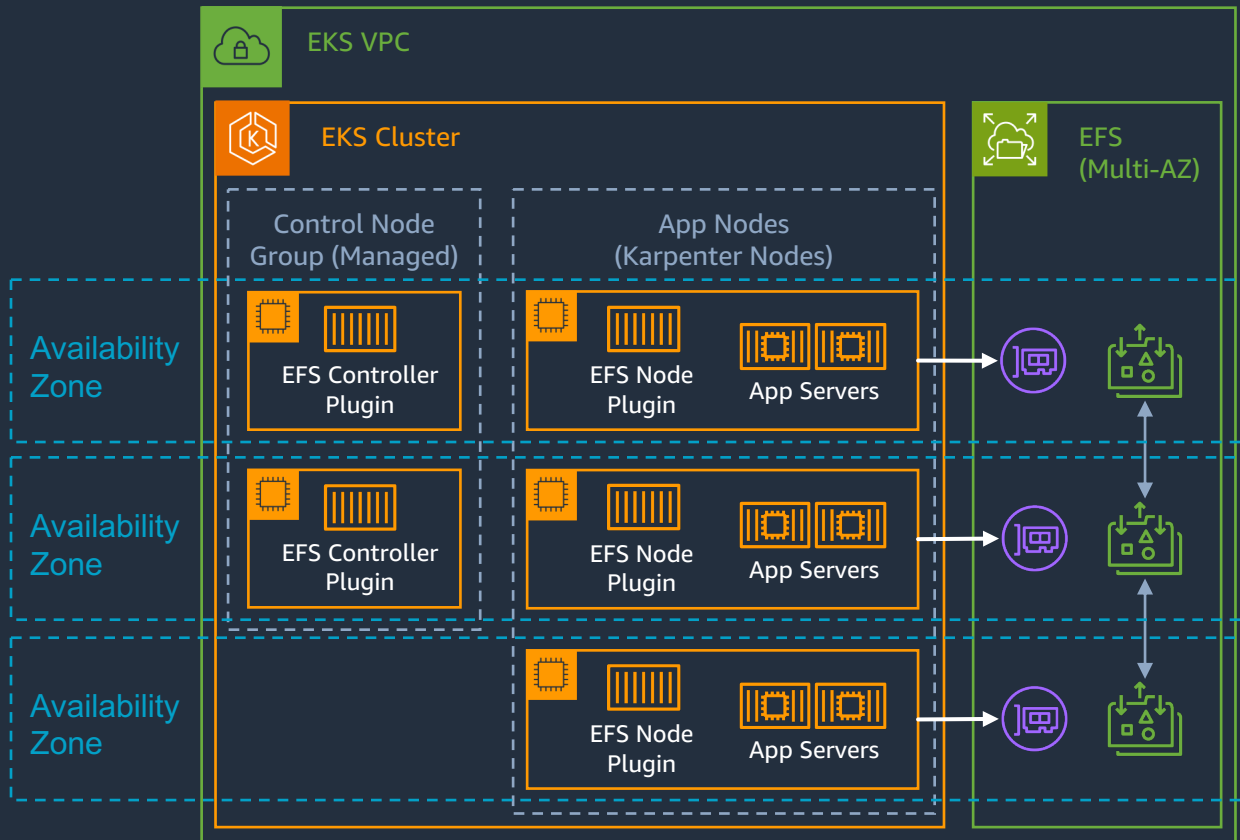
- K8s는 Storage Volume을 직접 관리하지 않음
- K8s는 CSI를 통해서 CSI Plugin에게 Storage Volume 관리 위임

EKS Cluster Architecture with AWS EBS



- AWS EBS Controller (CSI Plugin) 설치 필요
- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/ebs-csi.html
- EFS Controller Plugin
 - AWS EBS 생성/삭제, Attach/Detach 역할 수행
 - K8s Deployment를 이용하여 배포
 - 고가용성을 위한 2개 이상의 복제본 활용 (Active-Standby)
- EFS Node Plugin
 - AWS EBS Format, Mount/Unmount 수행
 - K8s DaemonSet을 이용하여 EBS를 이용하는 모든 EC2 Node에 배포

EKS Cluster Architecture with AWS EFS



- AWS EFS Controller (CSI Plugin) 설치 필요
 - <https://docs.aws.amazon.com/eks/latest/userguide/efs-csi.html>
- EFS Controller Plugin
 - AWS EFS 관리 역할 수행
 - K8s Deployment를 이용하여 배포
 - 고가용성을 위한 2개 이상의 복제본 활용 (Active-Standby)
- EFS Node Plugin
 - AWS EFS Mount/Unmount 역할 수행
 - K8s DaemonSet을 이용하여 EFS를 이용하는 모든 EC2 Node에 배포

Q & A