그림으로 배우는 쿠버네티스 (Kubernetes)

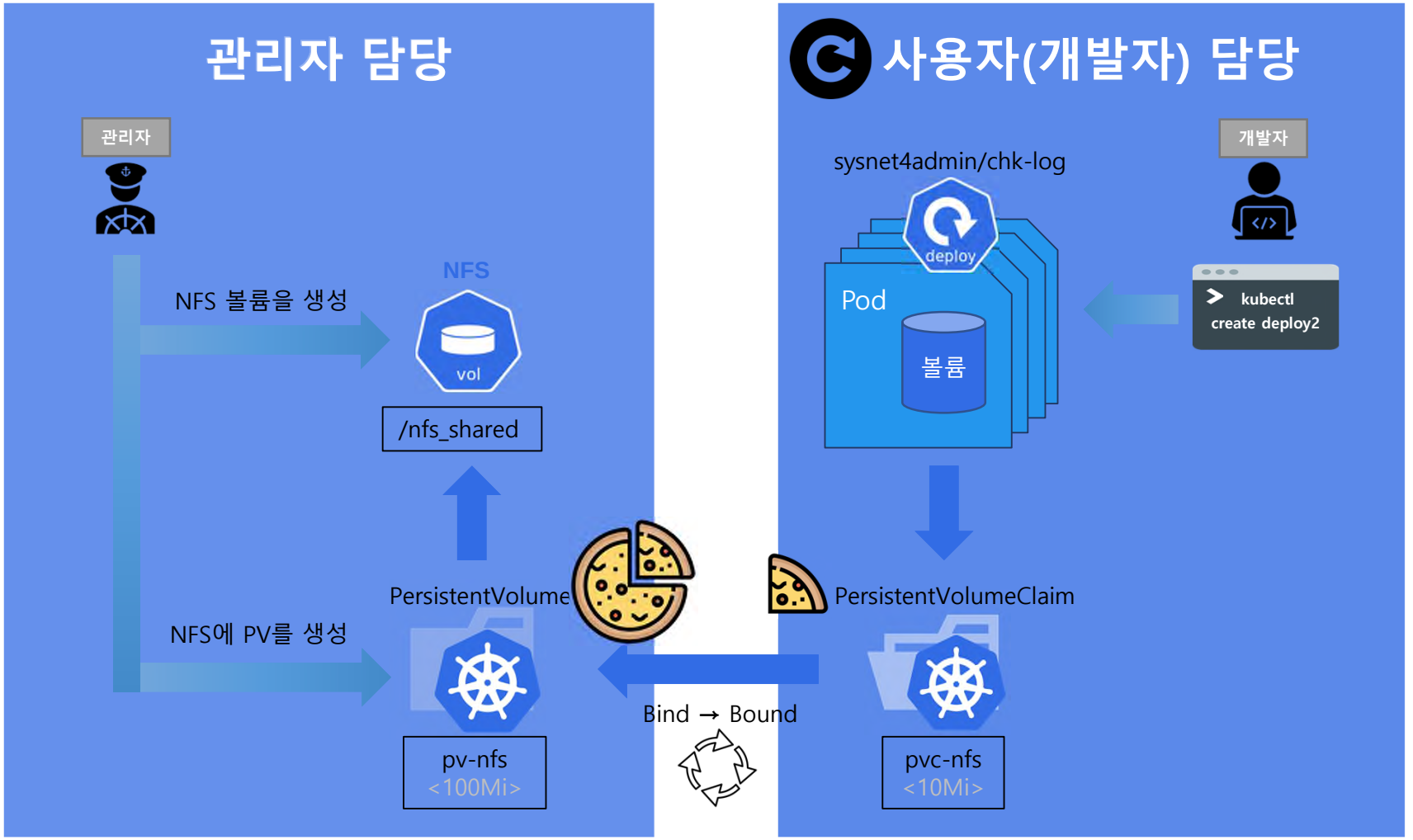
조 훈 (Hoon Jo)
- CCIE DC, CKA&D, VCIX-NV6, RHCE, GCPx4

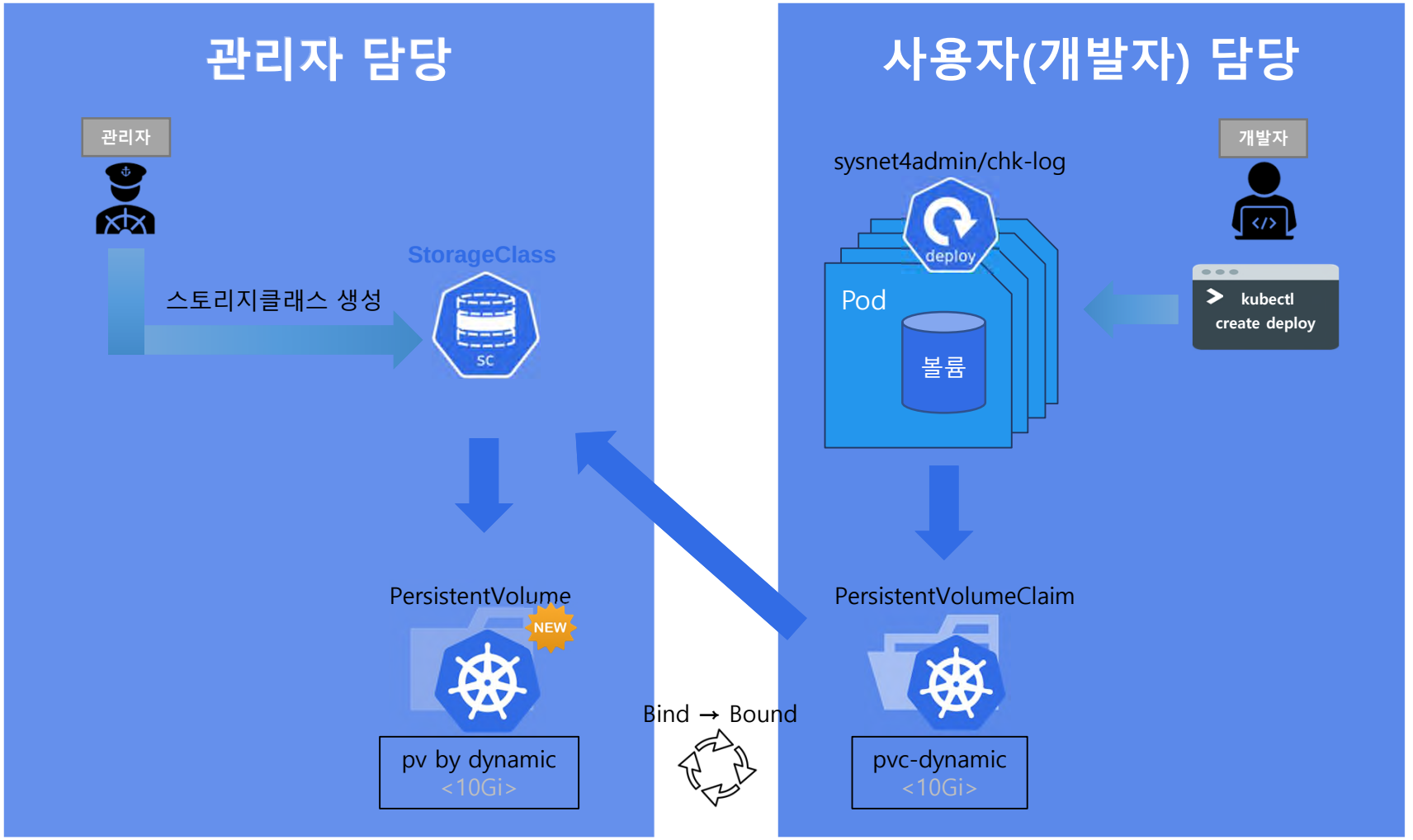
https://github.com/SysNet4Admin

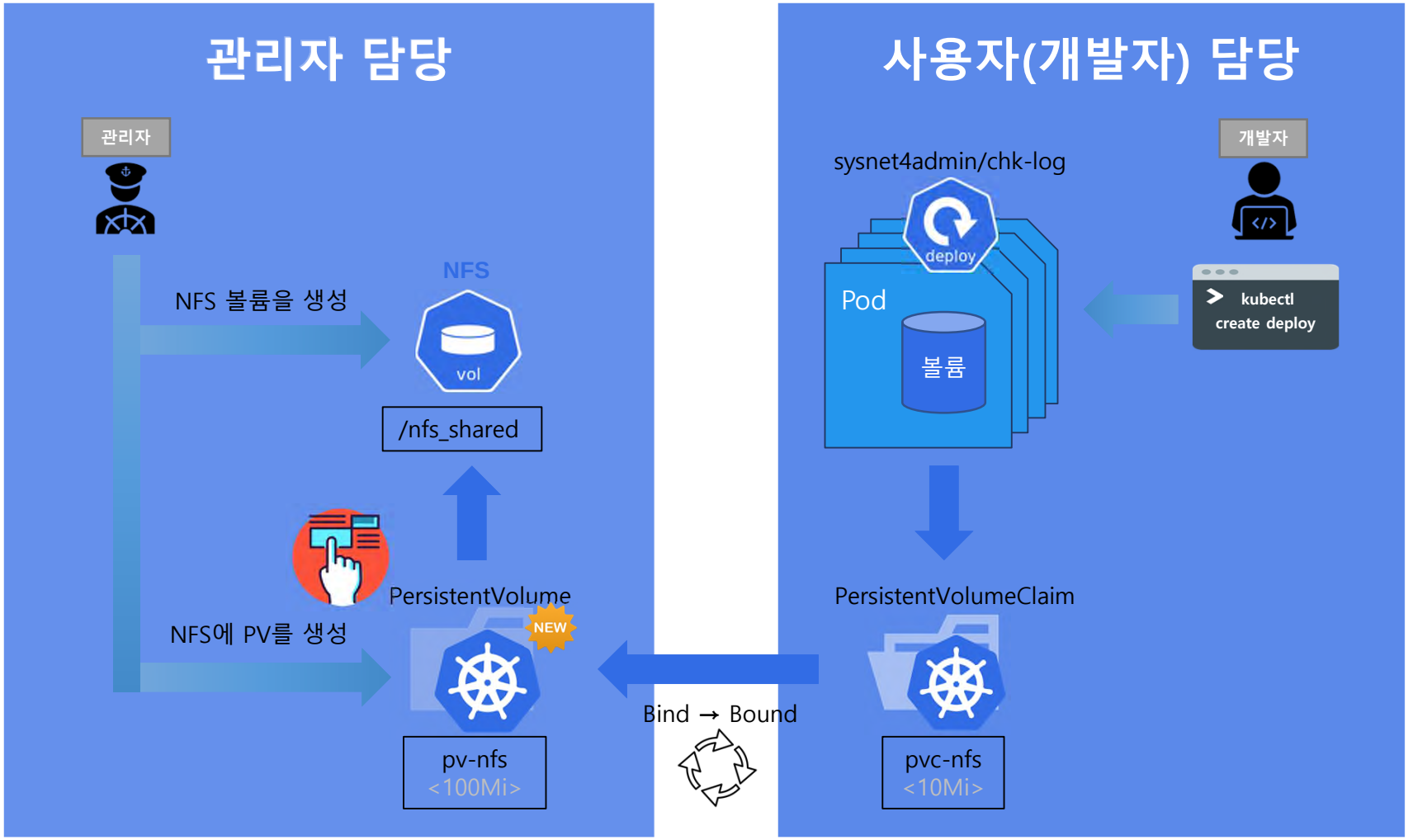
https://app.vagrantup.com/SysNet4Admin

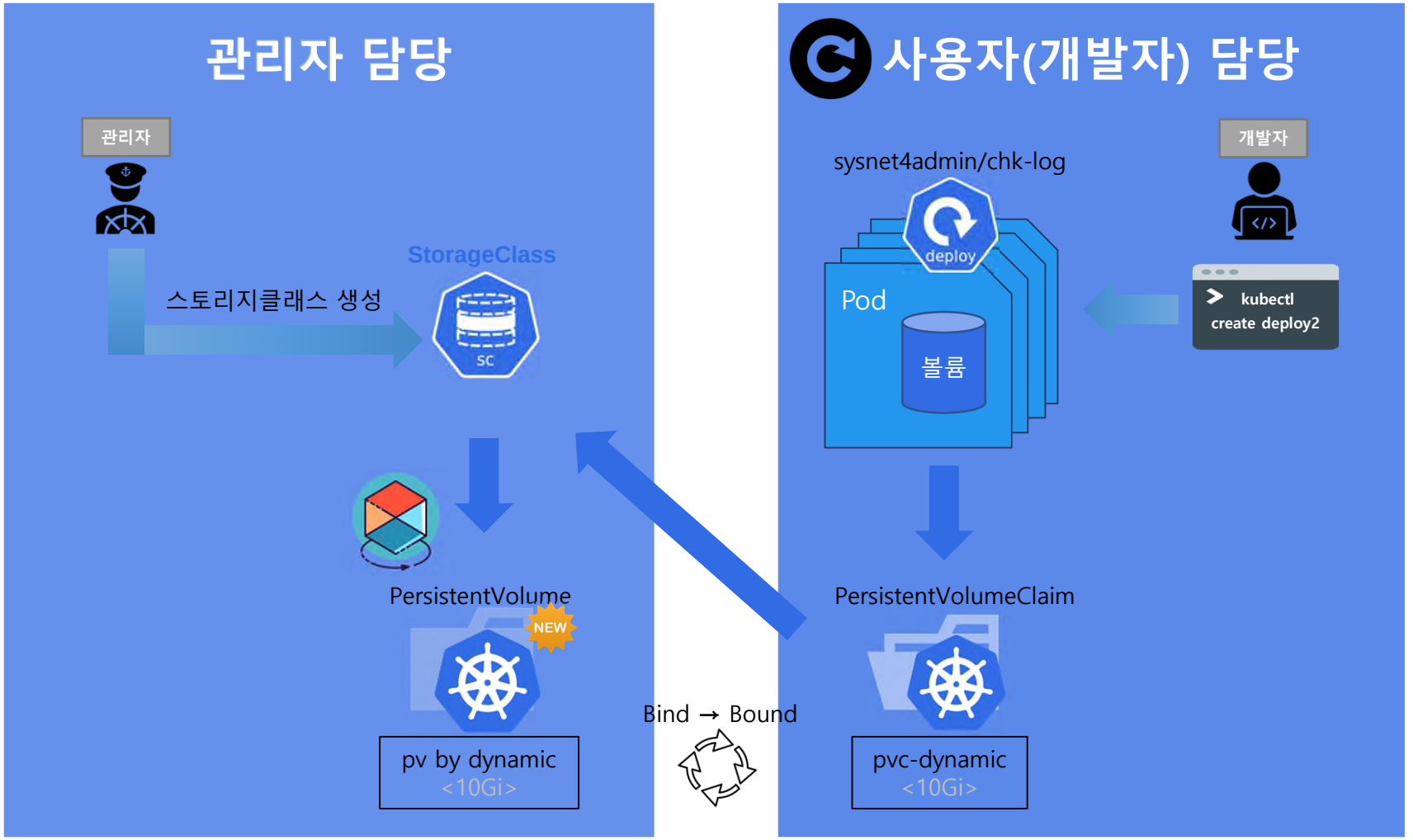
kubernetes

PV, PVC 할당의 문제점을 뭘까요?


관리자 담당
사용자(개발자) 담당
관리자
개발자
sysnet4admin/chk-log
NFS
NFS 볼륨을 생성
vol
/nfs_shared
deploy
Pod
볼륨
kubectl
create deploy2
PersistentVolume
PersistentVolumeClaim
NFS에 PV를 생성
Bind → Bound
pv-nfs
<100Mi>
pvc-nfs
<10Mi>


관리자 담당
관리자
스토리지클래스 생성
StorageClass
SC
PersistentVolume
NEW
pv by dynamic
<10Gi>
Bind → Bound
사용자(개발자) 담당
sysnet4admin/chk-log
deploy
Pod
볼륨
개발자
kubectl
create deploy
PersistentVolumeClaim
pvc-dynamic
<10Gi>


관리자 담당
사용자(개발자) 담당
관리자
개발자
sysnet4admin/chk-log
deploy
Pod
볼륨
kubectl
create deploy
NFS
NFS 볼륨을 생성
vol
/nfs_shared
PersistentVolume
NEW
NFS에 PV를 생성
PersistentVolumeClaim
Bind → Bound
pv-nfs
<100Mi>
pvc-nfs
<10Mi>


관리자 담당
관리자
스토리지클래스 생성
StorageClass
SC
PersistentVolume
NEW
pv by dynamic
<10Gi>
Bind → Bound
사용자(개발자) 담당
sysnet4admin/chk-log
deploy
Pod
볼륨
개발자
kubectl
create deploy2
PersistentVolumeClaim
pvc-dynamic
<10Gi>

볼륨과 클레임에 대한 생명 주기

GCEPersistentDisk

AWS Elastic Block Store

Azure File

Azure Disk

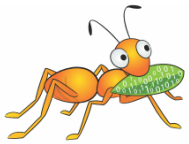
Cinder
(OpenStack block storage)

RBD
(Ceph Block Device)

Quobyte Volumes

StorageOS

Glusterfs



CSI
(Container Storage Interface)

Portworx Volumes

Flocker

Vsphere Volume

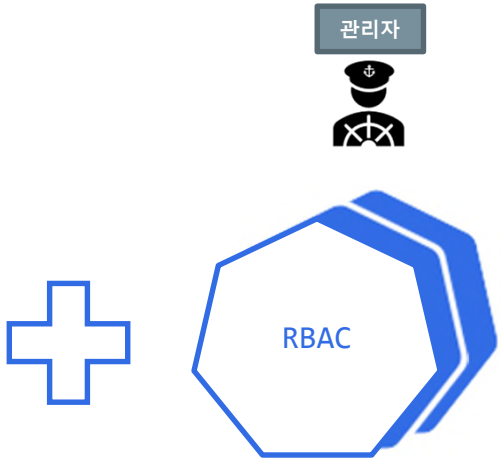
ScaleIO Volumes

NFS Provisioner
(nfs-subdir-external-provisioner)

DIR: nfs-subdir-external-provisioner/deployment.yaml (1/2), rbac.yaml

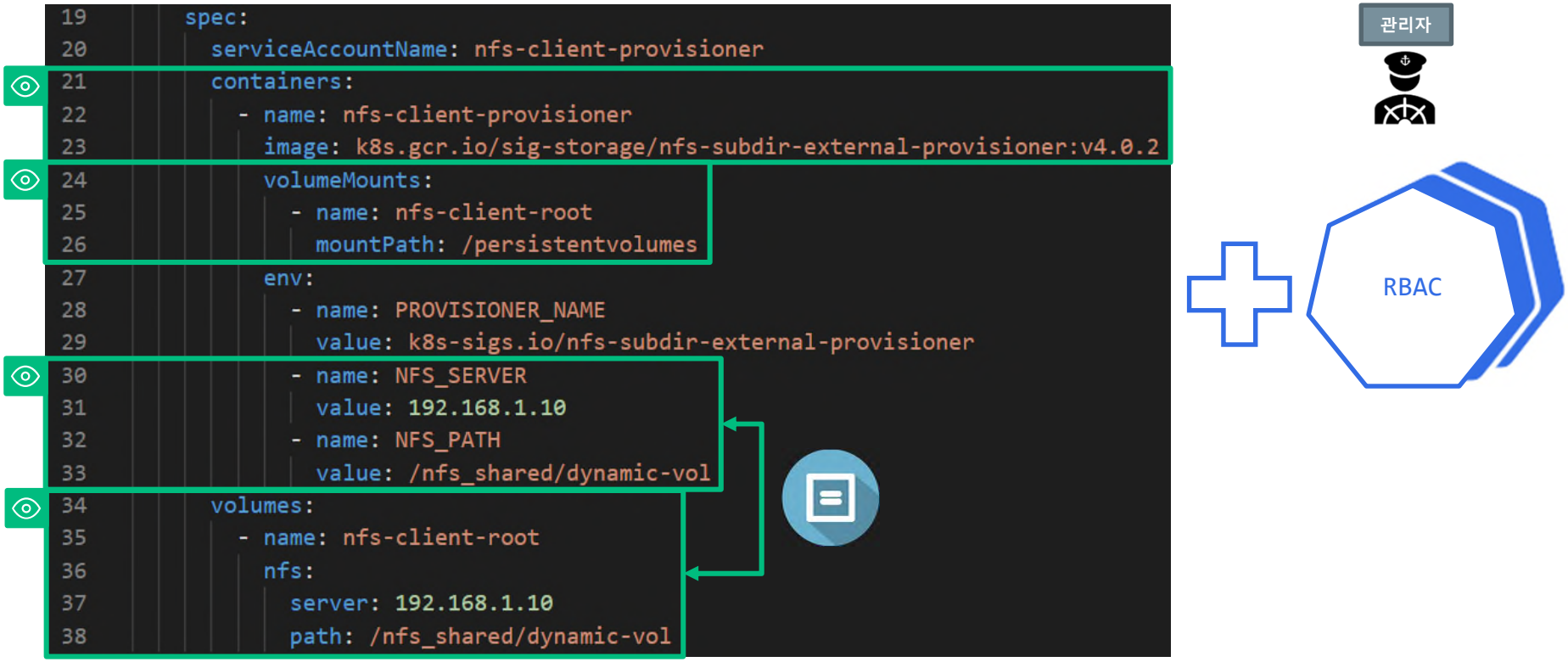
```yaml
# Origin Source
# https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner

kind: Deployment
apiVersion: apps/v1
metadata:
  name: nfs-client-provisioner
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nfs-client-provisioner
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: nfs-client-provisioner
```



DIR: nfs-subdir-external-provisioner/deployment.yaml (2/2), rbac.yaml

```yaml
19      spec:
20        serviceAccountName: nfs-client-provisioner
21        containers:
22          - name: nfs-client-provisioner
23            image: k8s.gcr.io/sig-storage/nfs-subdir-external-provisioner:v4.0.2
24            volumeMounts:
25              - name: nfs-client-root
26                mountPath: /persistentvolumes
27            env:
28              - name: PROVISIONER_NAME
29                value: k8s-sigs.io/nfs-subdir-external-provisioner
30              - name: NFS_SERVER
31                value: 192.168.1.10
32              - name: NFS_PATH
33                value: /nfs_shared/dynamic-vol
34        volumes:
35          - name: nfs-client-root
36            nfs:
37              server: 192.168.1.10
38              path: /nfs_shared/dynamic-vol
```



storageclass.yaml

```yaml
# Origin Source
# https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-nfs-storage
# or choose another name, must match deployment's env PROVISIONER_NAME'
provisioner: k8s-sigs.io/nfs-subdir-external-provisioner
parameters:
  # waits for nfs.io/storage-path annotation, if not specified will accept as empty string.
  pathPattern: "${.PVC.namespace}/${.PVC.annotations.nfs.io/storage-path}"
  onDelete: delete
```

persistentvolumeclaim-dynamic.yaml

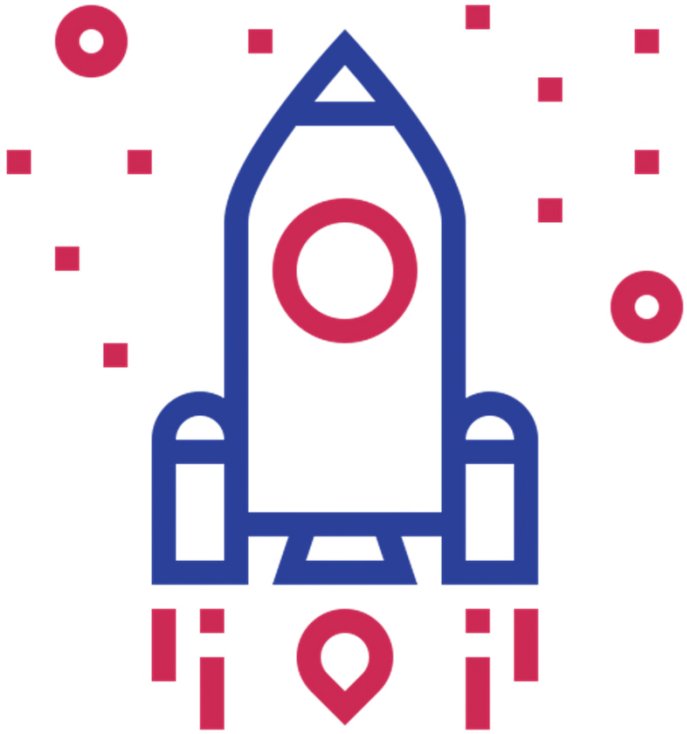
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-dynamic
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: managed-nfs-storage
```

deploy-pvc.yaml(+pvc)

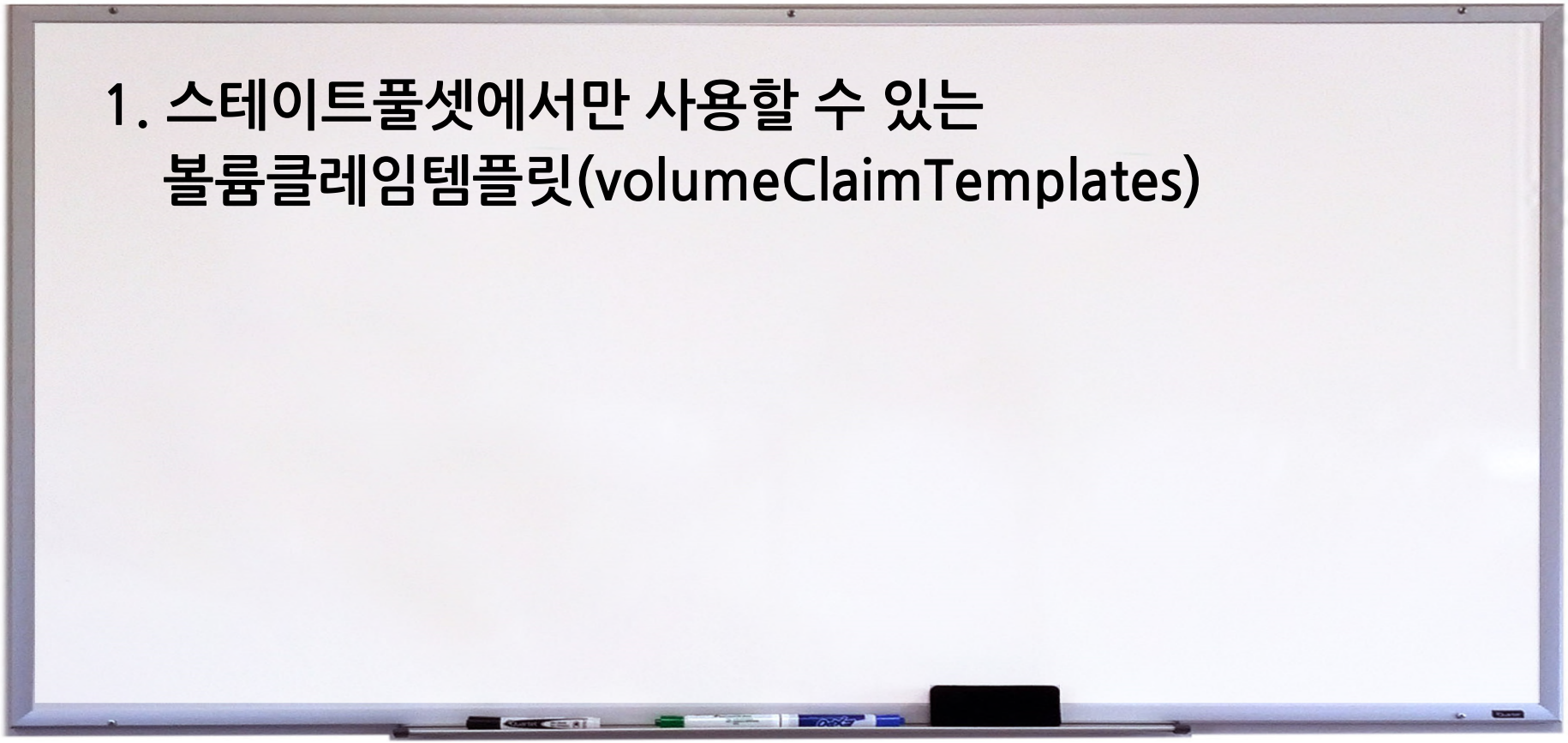
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-pvc
  labels:
    app: deploy-pvc
spec:
  replicas: 3
  selector:
    matchLabels:
      app: deploy-pvc
  template:
    metadata:
      labels:
        app: deploy-pvc
      spec:
        containers:
        - name: chk-log
          image: sysnet4admin/chk-log
          volumeMounts:
          - name: pvc-vol
            mountPath: /audit
        volumes:
        - name: pvc-vol
          persistentVolumeClaim:
            claimName: pvc-dynamic
```

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-dynamic
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: managed-nfs-storage
```

스토리지클래스 동작 확인



다음 강좌에는...



kubernetes