

# vSAN CSI 사용성 조사

본 문서는 vSphere의 vSAN 스토리지를 K8s에서 CSI를 이용하여 사용하는 방법을 다룹니다.

## [Prerequisite]

1. vSphere 클러스터 (ESXi 클러스터) 및 vCenter가 구성되어 있어야 합니다.

- vSphere 및 ESXi version: 7.0.1
- vCenter IP 및 port: 172.25.1.2, 80(HTTP)/443(HTTPS)

2. Data center와 cluster, VM이 생성돼 있어야 하고, VM은 네트워크 설정 및 k8s가 구성되어 있어야 합니다.

- data center 이름: Datacenter
- cluster 이름: Cluster
- VM 이름: ck2-4-test
- K8s version: 1.19.4, 노드 이름: ck2-4-test (minikube로 설치)
  - 주의! K8s node와 vCenter의 VM의 이름이 같아야 함

3. vSAN이 구성돼 있어야 합니다.

- vSAN 이름: vsanDatastore

## [Dynamic vSAN Provisioning]

동적으로 vSAN 볼륨을 생성하여 k8s에 provisioning합니다.

작업 순서는 크게 아래와 같습니다.

- Role, group 등 vSphere 설정
- K8s에 Cloud Provider Interface (CPI) 배포
- K8s에 CSI 배포

1. (vSphere 설정) vSAN 볼륨을 동적으로 provisioning하기 위해 vSphere의 role 및 privilege를 설정합니다.

- Role 이름: CNS-DATASTORE  
Privilege 내용: Datastore의 low level file operation (Datastore.FileManagement)
- Role 이름: CNS-HOST-CONFIG-STORAGE  
Privilege 내용: Host의 storage partition configuration (Host.Config.Storage)
- Role 이름: CNS-VM  
Privilege 내용: VM의 existing disk를 추가 및 device를 추가/제거하는 권한 (VirtualMachine.Config.AddExistingDisk, VirtualMachine.Config.AddRemoveDevice)
- Role 이름: CNS-SEARCH-AND-SPBM  
Privilege 내용: Cns의 search 권한 및 profile-driven storage의 view 권한 (Cns.Searchable, StorageProfile.View)

- Role 이름: ReadOnly (default로 vCenter에 있음)

아래 예시는 vCenter를 통해 role 중 하나인 Datastore의 file operation privilege를 가지는 “CNS-DATASTORE”를 만드는 과정입니다.

- vCenter의 메뉴 -> 관리 -> 액세스 제어 -> 역할 선택 후 role 추가

## 새 역할

- “하위 수준 파일 작업” 권한을 선택 후 NEXT
- 예시이므로 편의상 모든 권한을 선택

## 새 역할

역할 이름

설명

[CANCEL](#) [BACK](#) [FINISH](#)

- Role 이름을 위에 정리한 것과 같이 CNS-DATASTORE로 만들고 FINISH

본 예시의 과정을 모든 role에 대해 적용해줍니다.

## 2. (vSphere 설정) CSI를 위한 user group을 생성합니다.

아래 예시는 vCenter를 통해 vsphere.local user가 속하는 group인 csi를 만드는 과정입니다.

vSphere Client 관리 사용자 및 그룹 사용자 그룹 찾기 추가

그룹 이름	설명
ActAsUsers	Act-As Users
Administrators	
AutoUpdate	Users allowed to pe
CAAdmins	
ComponentManager.Administrators	Component Manage
csi	test for csi group
DCAdmins	
DCClients	
ExternalIDPUsers	Well-known externa
LicenseServiceAdministrators	License Service Adr

- vCenter의 메뉴 -> 관리 -> Single Sign On -> 사용자 및 그룹 -> 그룹 -> 추가 선택

그룹 추가

그룹 이름 \*

설명

멤버 추가 \*

vsphere.local

검색

취소 추가

- group 이름은 csi, group의 member로 vsphere.local을 입력 후 추가

3. (vSphere 설정) 각 role마다 해당 vSphere object에 CSI group으로 assign합니다.

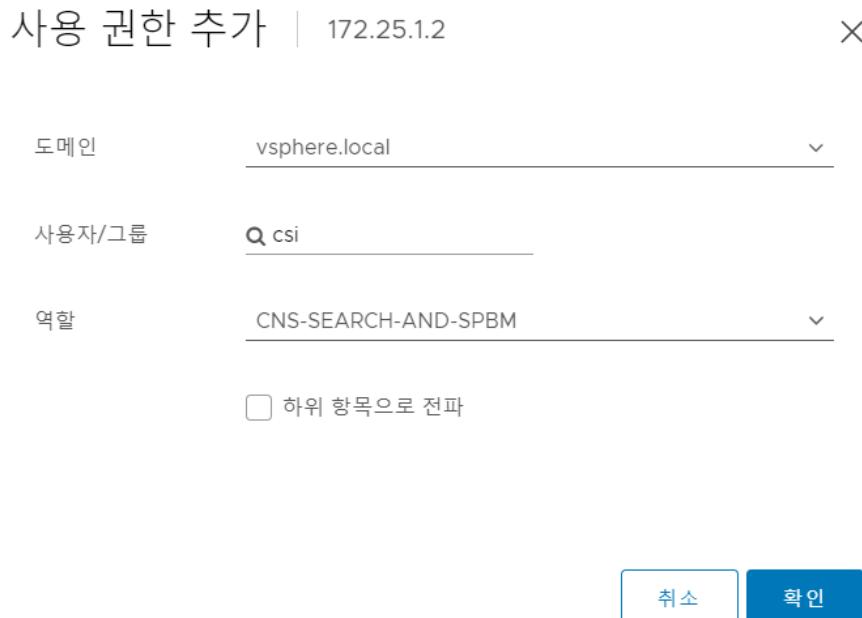
- CNS-DATASTORE: Datastore에 assign
- CNS-HOST-CONFIG-STORAGE: Cluster에 assign
- CNS-VM: K8s를 구성하는 Vm들에 assign

- CNS-SEARCH-AND-SPBM: vCenter Server에 assign
- ReadOnly: Data Center 및 host들에 assign

아래 예시는 vCenter Server에 CNS-SEARCH-AND-SPBM role을 assign하는 과정입니다.

사용자/그룹	역할
VSHERE.LOCAL\Administrator	관리자
VSHERE.LOCAL\Administrators	관리자
VSHERE.LOCAL\AutoUpdate	AutoUpdateUser
VSHERE.LOCAL\csi	CNS-SEARCH-AND-SPBM
VSHERE.LOCAL\NsxAdministrators	NsxAdministrator
VSHERE.LOCAL\NsxAuditors	NsxAuditor
VSHERE.LOCAL\NsxViAdministrators	NsxViAdministrator
VSHERE.LOCAL\RegistryAdministrators	컨텐츠 라이브러리 레지스트리
VSHERE.LOCAL\SyncUsers	SyncUsers
VSHERE.LOCAL\TrustedAdmins	신뢰할 수 있는 인프라 관리자
VSHERE.LOCAL\vpxd-7d278f75-e1c8-4171-ab77-a335e9f88229	관리자
VSHERE.LOCAL\vpxd-extension-7d278f75-e1c8-4171-ab77-a335e9f88229	관리자
VSHERE.LOCAL\vsphere-webclient-7d278f75-e1c8-4171-ab77-a335e9f88...	vSphere Client Solution User

- 메뉴 -> 호스트 및 클러스터 -> vCenter Server -> 사용 권한 -> 추가
- (위 그림은 vCenter Server에 CNS-SEARCH-AND-SPBM role이 이미 추가된 그림)



- vCenter Server의 user privilege에서 group은 csi, role은 CNS-SEARCH-AND-SPBM을 선택 후 확인

아래 예시는 vSphere cluster에 CNS-HOST-CONFIG-STORAGE role을 assign하는 과정입니다.

사용자/그룹	역할
VSPHERE.LOCAL\Administrator	관리자
VSPHERE.LOCAL\Administrators	관리자
VSPHERE.LOCAL\AutoUpdate	AutoUpdateUser
VSPHERE.LOCAL\csi	CNS-HOST-CONFIG-STORAGE
VSPHERE.LOCAL\NsxAdministrators	NsxAdministrator
VSPHERE.LOCAL\NsxAuditors	NsxAuditor
VSPHERE.LOCAL\NsxVIAdministrators	NsxVIAdministrator
VSPHERE.LOCAL\RegistryAdministrators	컨텐츠 라이브러리 레지스트리 관리자
VSPHERE.LOCAL\SyncUsers	SyncUsers
VSPHERE.LOCAL\TrustedAdmins	신뢰할 수 있는 인프라 관리자
VSPHERE.LOCAL\vpxd-7d278f75-e1c8-4171-ab77-a335e9f88229	관리자

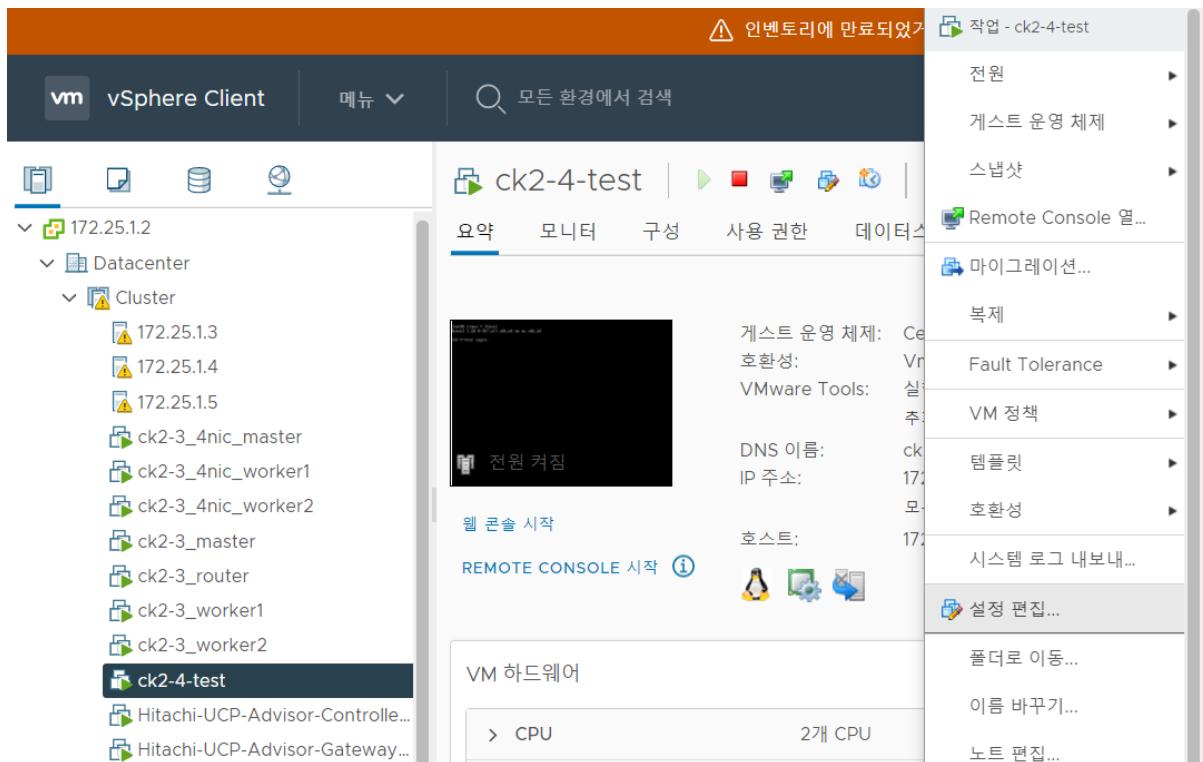
- 메뉴 -> 호스트 및 클러스터 -> Cluster -> 사용 권한 -> 추가
- (위 그림은 vSphere Cluster에 CNS-HOST-CONFIG-STORAGE role이 이미 추가된 그림)

사용 권한 추가 | Cluster ×

도메인	vsphere.local
사용자/그룹	<input type="text" value="csi"/>
역할	CNS-HOST-CONFIG-STORAGE
<input type="checkbox"/> 하위 항목으로 전파	
<span style="border: 1px solid #ccc; padding: 2px 10px;">취소</span> <span style="background-color: #0070C0; color: white; border: 1px solid #0070C0; padding: 2px 10px; font-weight: bold;">확인</span>	

- Cluster의 user privilege에서 group은 csi, role은 CNS-HOST-CONFIG-STORAGE 를 선택 후 확인

4. (vSphere 설정) K8s를 구성하는 모든 VM들에 disk.EnableUUID 파라미터를 활성화합니다.



- 해당 VM 우클릭 -> 설정 편집 -> VM 옵션 -> 고급 -> 구성 매개 변수 -> 구성 편집
- 구성 매개 변수

**⚠️** 실험 버전 기능의 필요에 따라 또는 기술 지원의 지시에 따라 구성 매개 변수를 수정 또는 추가하십시오. 비어 있는 값은 제거됩니다 (ESXi 6.0 이상에서 지원됨).

Name	Value
disk.EnableUUID	TRUE
ethernet0.pciSlotNumber	192
ethernet1.pciSlotNumber	224
guestinfo.vmtools.buildNumber	8068406
guestinfo.vmtools.description	open-vm-tools 10.2.5 build 8068406

- 파라미터 이름은 disk.EnableUUID, 값은 TRUE로 생성
- (VM이 작동중이면 파라미터 추가가 안 되므로, shutdown 시킨 상태에서 추가하기)

## 5. (CPI 배포) 모든 K8s node (VM)에 아래 taint를 설정합니다.

- node.cloudprovider.kubernetes.io/uninitialized=true:NoSchedule
- 본 예시의 CMD  

```
kubectl taint node ck2-4-test
node.cloudprovider.kubernetes.io/uninitialized=true:NoSchedule
```

- CPI controller는 DaemonSet으로 뜨며, controller가 생성되고 나면 kubelet을 통해 해당 taint를 제거해줌

주의) taint 작업은 CPI 배포를 위해 반드시 필요함

주의) K8s node 이름과 vSphere VM 이름이 같아야 함

## 6. (CPI 배포) vSphere configuration 정보를 가지는 CPI용 ConfigMap을 생성합니다.

- 아래 CMD 수행

```
tee vsphere.conf >/dev/null <<EOF
[Global]
insecure-flag = "true"

[VirtualCenter "IP or FQDN"]
user = "username@vsphere.local"
password = "password"
port = "port"
datacenters = "<datacenter1-path>, <datacenter2-path>, ..."
EOF
```

- 각 필드의 값은 아래와 같음

- insecure-flag: self-signed certificate로 login하려면 true로 설정해야 함
- VirtualCenter: vCenter의 IP 주소를 “IP or FQDN”에 입력
- user: vCenter의 user 이름을 “username@vsphere.local”에 입력
- password: vCenter의 해당 user의 password를 “password”에 입력
- port: vCenter Server의 port를 “port”에 입력
- datacenters: Data Center의 path를 “<datacenter1-path> ...”에 comma-separated로 입력 (root에 있을 경우 Data Center의 이름 입력)

- 본 예제에서는 아래와 같이 CMD 수행

```
tee vsphere.conf >/dev/null <<EOF
[Global]
insecure-flag = "true"

[VirtualCenter "172.25.1.2"]
user = "administrator@vsphere.local"
password = "Passw0rd!"
port = "443"
datacenters = "Datacenter"
EOF
```

- vsphere.conf 파일 생성 이후 아래 CMD 수행

```
kubectl create configmap cloud-config --from-file=vsphere.conf
--namespace=kube-system
```

수행 후 vsphere.conf 파일 삭제를 권장

## 7. (CPI 배포) RBAC 관련 object, Service 및 controller DaemonSet을 생성합니다.

- 예제에서는 아래 CMD 수행

```
kubectl apply -f vsphere-csi/cpi/cloud-controller-manager-roles.yaml
```

```
kubectl apply -f  
vsphere-csi/cpi/cloud-controller-manager-role-bindings.yaml  
kubectl apply -f vsphere-csi/cpi/vsphere-cloud-controller-manager-ds.yaml
```

- Controller가 정상적으로 launch되면 K8s node에 ProviderID가 set 되었는지 확인  
kubectl describe nodes | grep "ProviderID"  
ProviderID: vsphere://422528c3-b3c8-de23-1913-efe5c9e286d2

ProviderID가 set 되었으면 해당 node들에서 CPI를 통해 volume 생성 및 마운트 작업을 처리할 수 있게 됨

#### 8. (CSI 배포) vSphere configuration 정보를 가지는 CSI용 Secret을 생성합니다.

- 아래 CMD 수행

```
$ tee csi-vsphere.conf >/dev/null <<EOF  
[Global]  
cluster-id = "<cluster-id>"  
cluster-distribution = "<cluster-distribution>"  
  
[VirtualCenter "<IP or FQDN>"]  
insecure-flag = "<true or false>"  
user = "<username>"  
password = "<password>"  
port = "<port>"  
datacenters = "<datacenter1-path>, <datacenter2-path>, ..."  
  
EOF
```

- 필드 중 6번과 겹치는 필드는 6번에서의 설명과 동일
  - cluster-id: cluster의 identifier로, 64byte를 초과하면 안됨
  - cluster-distribution: K8s의 distribution을 나타냄. 현 버전에서는 optional이지만 곧 필수 파라미터로 바뀔 예정

- 본 예제에서는 아래와 같이 CMD 수행

```
tee csi-vsphere.conf >/dev/null <<EOF  
[Global]  
cluster-id = "minikube"  
  
[VirtualCenter "172.25.1.2"]  
insecure-flag = "true"  
user = "administrator@vsphere.local"  
password = "Passw0rd!"  
port = "443"  
datacenters = "Datacenter"  
EOF
```

- csi-vsphere.conf 파일 생성 이후 아래 CMD 수행

```
kubectl create secret generic vsphere-config-secret  
--from-file=csi-vsphere.conf --namespace=kube-system
```

수행 후 csi-vsphere.conf 파일 삭제를 권장

## 9. (CSI 배포) Controller와 node plugin의 RBAC, Deployment 및 DaemonSet을 생성합니다.

- 예제에서는 아래 CMD 수행

```
kubectl apply -f vsphere-csi/csi/vsphere-csi-controller-rbac.yaml
kubectl apply -f vsphere-csi/csi/vsphere-csi-node-rbac.yaml
kubectl apply -f vsphere-csi/csi/vsphere-csi-controller-deployment.yaml
kubectl apply -f vsphere-csi/csi/vsphere-csi-node-ds.yaml
```

- Controller 및 node plugin이 정상적으로 launch되면 CSIDriver와 CSINode가 정상적으로 생성되었는지 확인

```
$ kubectl describe csidrivers
  Name:          csi.vsphere.vmware.com
  Namespace:
  Labels:        <none>
  Annotations:   <none>
  API Version:  storage.k8s.io/v1
  Kind:         CSIDriver
  Metadata:
    Creation Timestamp: 2021-05-26T04:03:47Z
  Managed Fields:
    API Version:  storage.k8s.io/v1
    Fields Type:  FieldsV1
    ...
  
```

CSIDriver를 확인한 후 CSINode 확인

```
$ kubectl get CSINode
NAME      DRIVERS      AGE
ck2-4-test  1           7d5h
```

## 10. vSAN 볼륨을 동적 생성할 StorageClass를 생성합니다.

- 아래와 같은 내용으로 StorageClass 생성

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: example-block-sc
provisioner: csi.vsphere.vmware.com
allowVolumeExpansion: true
parameters:
  storagepolicyname: "vSAN Default Storage Policy" #Optional Parameter
  csi.storage.k8s.io/fstype: "ext4" # Optional Parameter
```

- 예제에서는 아래와 같은 CMD 수행

```
kubectl apply -f vsphere-csi/csi/example-sc.yaml
```

## 11. PVC 및 Pod을 띄워 volume provisioning 및 attach가 작동함을 확인합니다.

- 아래와 같은 내용으로 PVC 생성

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: example-block-pvc
```

```
spec:  
  accessModes:  
    - ReadWriteOnce  
  resources:  
    requests:  
      storage: 1Gi  
  storageClassName: example-block-sc
```

- 예제에서는 아래와 같은 CMD 수행  
`kubectl apply -f vsphere-csi/csi/example-pvc.yaml`
- 예제에서는 아래와 같은 CMD 수행으로 PVC를 사용하는 Pod 생성  
`kubectl apply -f vsphere-csi/csi/example-deploy.yaml`

참고한 문서는 아래와 같습니다.

- <https://vsphere-csi-driver.sigs.k8s.io>