

Nutanix File CSI 사용성 조사

본 문서는 Nutanix File 스토리지를 K8s에서 CSI를 이용하여 사용하는 방법을 다룹니다.

[Prerequisite]

1. Nutanix 클러스터 및 Prism이 구성되어 있어야 합니다.
 - AOS version: 5.15.5.1 LTS
2. Nutanix File Server 및 Export가 생성되어 있어야 합니다.
 - Nutanix File Server는 아래와 같이 생성되어 있습니다.

The screenshot shows the Nutanix Prism web interface for File Servers. At the top, there's a navigation bar with 'File Server', 'Share/Export', 'Deploy File Analytics', and 'Network Config'. Below this, a tabbed interface shows 'File Server' and 'Share/Export'. A table lists the file servers, with one entry 'fs-server1' showing 1 Share/Export Count, 1 Open Connection, and 600 KiB Space Used (Logical). Below the table, a 'Summary' section on the left shows 'File Server(s): fs-server1', 'Share/Export Count: 1', and 'Used Capacity: 600 KiB'. On the right, a 'File Server Alerts' section shows one alert: 'File server fs-server1's DNS entries need to be...'. The alert details include 'Severity: Info', 'Issue: File server fs-server1's DNS entries need to be...', 'Create Time: 04/28/21', 'Last Occurred: 04/28/21', and 'Status: Resolved'. A link to 'Windows 제품 인증 (공칭)으로 이동하여 Windows를 정품 인증합니다.' is also visible.

Name	Share/Export Count	Open Connections	Space Used (Logical)	Space Used By Snapshots (Logical)	Recommendations
fs-server1	1	1	600 KiB	-	-

Summary

FILE SERVER SUMMARY

File Server(s) **fs-server1**

Share/Export Count **1**

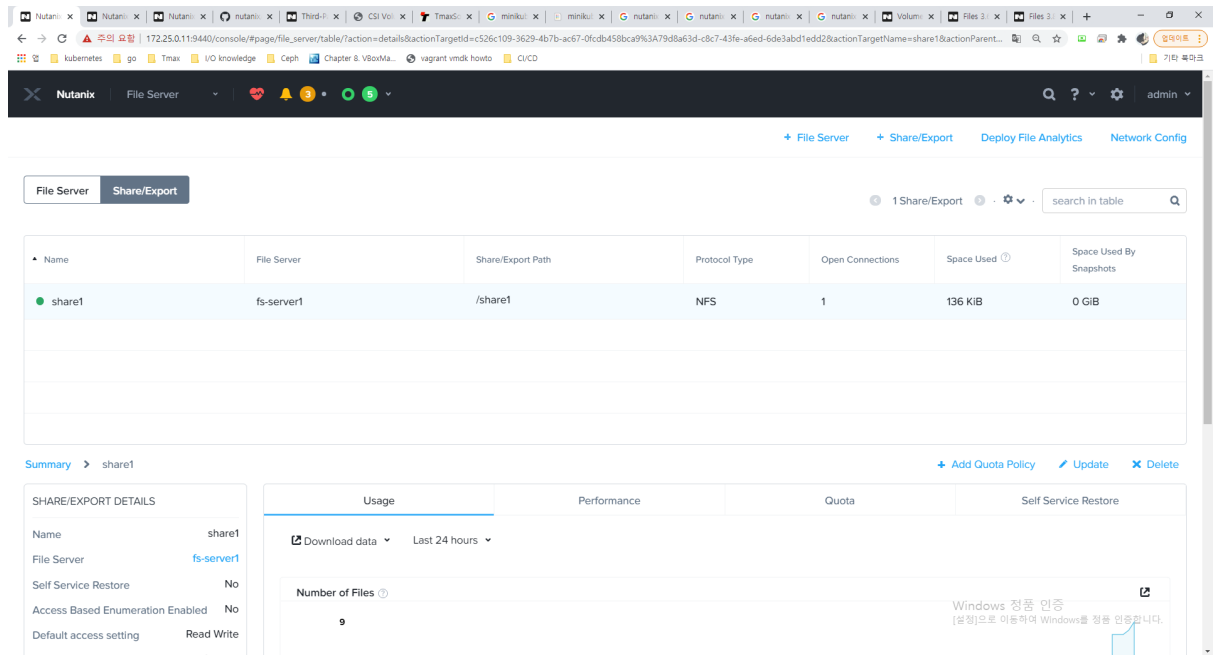
Used Capacity **600 KiB**

File Server Alerts

Unresolved **1 Alert**

Severity	Issue	Create Time	Last Occurred	Status	Details
Info	File server fs-server1's DNS entries need to be...	04/28/21	04/28/21	Resolved	Windows 제품 인증 (공칭)으로 이동하여 Windows를 정품 인증합니다.

- Nutanix File Server에서 NFS export를 위해 생성한 Share/Export 정보는 아래와 같습니다.



본 문서에서 예시로 사용하는 File Server 및 Export 정보는 아래와 같습니다. (File Server는 다수의 VM으로 떼서 작동하며 NFS, SMB 등의 프로토콜을 통해 endpoint를 제공합니다)

File Server Specs (VMs)

- File Server version: 3.8.0.1
- Protocols: NFS (v3, v4)
- Client Network IP: 172.25.0.33
- Storage Network IP: 172.25.0.34
- 4 CPU, 12Gi Mem, 1TiB Storage
- Directory Authentication: Unmanaged (그 외에 LDAP, Active Directory 등이 있음)

File Server Export Specs

- Export Path: /share1
- Protocol: NFS (v3, v4)
- Max size: 100GiB

3. Nutanix VM으로 구성한 K8s 클러스터가 있어야 합니다.

- 실험 환경의 VM은 CentOS 7.9입니다
- 단일 VM에 minikube로 K8s를 구성했고, K8s 버전은 1.19.4입니다.

[Static NFS Provisioning]

기존에 생성돼있는 NFS export를 StorageClass를 통해 PVC로 사용할 수 있도록 provisioning 합니다.

1. nfs-utils 패키지를 설치합니다.
 - CMD: yum install -y nfs-utils
2. Nutanix에서 제공하는 CSI 에제 yaml 파일들을 다운받습니다.
 - 본 예제에서는 Nutanix CSI 2.3.1을 사용했습니다. (참고 <https://github.com/nutanix/csi-plugin>)
 - CMD: wget http://download.nutanix.com/csi/v2.3.1/csi-v2.3.1.tar.gz
 - CMD: tar xvf csi-v2.3.1.tar.gz && cd CSI-v2.3.1
3. Nutanix CSI 관련 RBAC을 생성합니다.
 - CMD: kubectl apply -f ntnx-csi-rbac.yaml
4. Provisioner가 접근하기 위한 secret을 생성합니다.
 - secret은 아래와 같은 내용으로 생성합니다.

```
apiVersion: v1
kind: Secret
metadata:
  name: ntnx-secret
  namespace: kube-system
data:
  key: MTcyLjI1LjAuMTE6OTQ0MDphZG1pbjpwOXRhbml4MSE=
```

- key 값은 base64 인코딩된 “prism-ip:prism-port:admin:password” 값입니다.
 - CMD: echo -n '172.25.0.11:9440:admin:Nutanix1!' | base64
5. Nutanix CSI provisioner가 작동하는 데 이용하는 CSIDriver를 생성합니다.
 - CMD: kubectl apply -f csi-driver.yaml
 6. Nutanix CSI node provisioner DaemonSet을 생성합니다.
 - CMD: kubectl apply -f ntnx-csi-node.yaml
 7. Nutanix CSI provisioner를 생성합니다.
 - CMD: kubectl apply -f ntnx-csi-provisioner.yaml

진행 후 Pod 결과는 아래와 같습니다.

```
[root@localhost csi-v2.3.1]# kubectl get pod -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
coredns-f9fd979d6-jbxd5	1/1	Running	0	111m
csi-node-ntnx-plugin-brs2s	3/3	Running	0	108m
csi-provisioner-ntnx-plugin-0	5/5	Running	0	106m
etcd-localhost.localdomain	1/1	Running	0	112m
kube-apiserver-localhost.localdomain	1/1	Running	0	112m
kube-controller-manager-localhost.localdomain	1/1	Running	0	112m
kube-proxy-6nswf	1/1	Running	0	111m
kube-scheduler-localhost.localdomain	1/1	Running	0	112m
storage-provisioner	1/1	Running	0	112m

8. Nutanix File의 StorageClass를 생성합니다.

- CMD: cd example/AFS
- StorageClass는 아래와 같이 설정합니다.

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: acs-afs
provisioner: csi.nutanix.com
parameters:
  nfsServer: 172.25.0.33
  nfsPath: /share1
  storageType: NutanixFiles
reclaimPolicy: Delete
```

- parameters.nfsServer에는 NFS File Server의 IP를 입력합니다.
 - 본 예제에서 Nutanix File Server의 client network IP는 172.25.0.33
- parameters.nfsPath에는 NFS export의 mount path를 입력합니다.
 - 본 예제에서 Nutanix File Server Export의 mount path는 /share1
- CMD: kubectl apply -f sc.yaml

9. 해당 StorageClass로부터 볼륨을 할당받는 PVC를 생성합니다.

- CMD: kubectl apply -f claim2.yaml

```
[root@localhost AFS]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
claim2	Bound	pvc-d76f9a16-f2ff-44d0-820c-b7b1b109f3e4	1Gi	RWX	acs-afs	110m

10. 위 PVC를 사용하는 Pod을 띄워 테스트합니다.

- CMD: kubectl apply -f rc-nginx2.yaml

```
[root@localhost AFS]# kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
server2-ptb2b	1/1	Running	0	112m

11. 해당 Pod에서 만든 파일이 다른 mountpoint에서도 보이는지 확인합니다.

```
[root@localhost AFS]# ka -f rc-nginx2.yaml
replicationcontroller/server2 created
[root@localhost AFS]#
[root@localhost AFS]#
[root@localhost AFS]# kubectl exec -it server2-fz97f -- /bin/bash
root@server2-fz97f:/# cd /mnt/nfs
root@server2-fz97f:/mnt/nfs# touch testfile
root@server2-fz97f:/mnt/nfs# echo "hello" > hello.txt
root@server2-fz97f:/mnt/nfs# ls
hello.txt  testfile
```

- 해당 NFS export를 직접 마운트하여 PVC 경로에 위 파일들이 있는지 확인합니다.

```
[root@localhost AFS]# mount 172.25.0.33:/share1 test
[root@localhost AFS]# cd test/pvc-d76f9a16-f2ff-44d0-820c-b7b1b109f3e4/
[root@localhost pvc-d76f9a16-f2ff-44d0-820c-b7b1b109f3e4]# ls
hello.txt  testfile
[root@localhost pvc-d76f9a16-f2ff-44d0-820c-b7b1b109f3e4]# cat hello.txt
hello
```

[Dynamic NFS Provisioning]

동적으로 NFS export를 PVC 당 생성하여 NFS 볼륨을 provisioning 합니다.

1. Static NFS Provisioning 방법의 7번까지 진행합니다.
2. Dynamic NFS provisioning을 위한 StorageClass를 생성합니다.
 - StorageClass를 아래와 같이 설정한 후 생성합니다.
 - CMD: kubectl apply -f dyn-nfs-sc.yaml

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: nfs-sc
provisioner: csi.nutanix.com
parameters:
  dynamicProv: ENABLED
  nfsServerName: fs-server1
  csi.storage.k8s.io/provisioner-secret-name: ntnx-secret
  csi.storage.k8s.io/provisioner-secret-namespace: kube-system
  storageType: NutanixFiles
```

- parameters.dynamicProv 필드에 ENABLED를 값으로 입력합니다.

- parameters.nfsServerName 필드에 Nutanix NFS File Server 이름을 입력합니다.
 - 본 예제에서 File Server 이름은 fs-server1
- parameters.csi.storage.k8s.io/provisioner-secret-name 필드에 Prism 접근 정보에 관한 Secret 이름을 입력합니다.
- parameters.csi.storage.k8s.io/provisioner-secret-namespace 필드에 Prism 접근 정보에 관한 Secret의 Namespace를 입력합니다.
- parameters.storageType 필드에 NutanixFiles를 값으로 입력합니다.

3. 해당 StorageClass로부터 볼륨을 할당받는 PVC를 생성합니다.

- CMD: `kubectl apply -f claim_dyn.yaml`

```
[root@localhost AFS]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
claim-dyn	Bound	pvc-2b846615-eb7f-4faf-9212-ae25b3376617	2Gi	RWX	nfs-sc	20m
claim2	Bound	pvc-d76f9a16-f2ff-44d0-820c-b7b1b109f3e4	1Gi	RWX	acs-afs	155m

위와 같이 PVC가 Bound 되면 아래와 같이 Prism - File Server - Share/Export 창에서 새로 NFS export가 생긴 것을 확인할 수 있습니다.

The screenshot shows the Nutanix File Server interface. The 'Share/Export' tab is active, displaying a table of NFS exports. The first row, representing the newly created export for the PVC, is highlighted with a red box. Below the table, there is a 'Summary' section on the left and 'File Server Alerts' and 'File Server Events' sections on the right.

Name	File Server	Share/Export Path	Protocol Type	Open Connections	Space Used	Space Used By Snapshots
pvc-2b846615-eb7f-4faf-9212-ae25b3376617	fs-server1	/pvc-2b846615-eb7f-4faf-9212-ae25b3...	NFS	0	128 KiB	0 GiB
share1	fs-server1	/share1	NFS	1	136 KiB	0 GiB

4. 해당 PVC를 사용하는 Pod을 Static NFS Provisioning과 같은 방법으로 띄워 테스트합니다.