

# vSphere IaaS Control Plane 백업 및 복원

업데이트 3

VMware vSphere 8.0

VMware vCenter 8.0

VMware ESXi 8.0

VMware by Broadcom 웹 사이트

<https://docs.vmware.com/kr>에서 최신 기술 문서를 찾을 수 있습니다.

**VMware by Broadcom**

3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2024 Broadcom. All Rights Reserved. “Broadcom”은 Broadcom Inc. 및/또는 해당 자회사를 뜻합니다. 자세한 내용은 <https://www.broadcom.com> 페이지를 참조하십시오. 여기에서 언급된 모든 상표, 상호, 서비스 마크 및 로고는 해당 회사의 소유입니다.

# 목차

- "vSphere IaaS 제어부 백업 및 복원" 4
- 1 vSphere IaaS Control Plane 백업 및 복원에 대한 고려 사항 5
- 2 감독자 제어부 백업 및 복원 7
  - 감독자 상태 백업 7
  - 감독자 제어부 복원 8
- 3 감독자에 vSphere용 Velero 플러그인 설치 및 구성 10
- 4 TKG 서비스 클러스터 및 워크로드 백업 및 복원 20
  - TKG 서비스 클러스터 및 워크로드 백업 및 복원에 대한 고려 사항 20
  - vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원 21
    - TKG 클러스터에 vSphere용 Velero 플러그인 설치 및 구성 21
    - vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원 26
  - Restic과 함께 독립형 Velero를 사용하여 감독자에서 TKG 클러스터 워크로드 백업 및 복원 27
    - TKG 클러스터에 독립형 Velero 및 Restic 설치 및 구성 28
    - 독립형 Velero 및 Restic을 사용하여 클러스터 워크로드 백업 및 복원 33
  - CSI 스냅샷과 함께 Velero를 사용하여 백업 및 복원 40
- 5 vSphere IaaS Control Plane에서 VM 서비스 VM 백업 및 복원 43
  - VM 서비스 VM 수동 등록 44
- 6 vSphere용 Velero 플러그인을 사용하여 vSphere 포드 백업 및 복원 46
- 7 vSphere IaaS Control Plane 백업 및 복원 문제 해결 50
  - 백업에서 감독자 복원 후 분리된 개체 정리 50

# "vSphere IaaS 제어부 백업 및 복원"

"vSphere IaaS 제어부 백업 및 복원" 항목은 Tanzu Kubernetes Grid 클러스터 및 vSphere 포드에서 실행되는 워크로드뿐만 아니라 감독자 제어부를 백업하고 복원하는 방법에 대한 정보를 제공합니다.

## 대상 사용자

이 정보는 vSphere IaaS Control Plane에서 실행되는 워크로드와 감독자 제어부 상태를 백업 및 복원하려는 vSphere 관리자 및 DevOps 엔지니어를 대상으로 합니다. 다음 영역에 대한 지식이 필요합니다.

- vSphere IaaS Control Plane
- vSphere
- Kubernetes
- Velero
- 인스턴스 스토리지

# vSphere IaaS Control Plane 백업 및 복원에 대한 고려 사항

# 1

vSphere IaaS Control Plane에 대한 백업 및 복원 프로세스가 무엇인지 알아보고 vSphere IaaS Control Plane의 백업 및 복원 전략을 구현하기 위한 개략적인 고려 사항을 숙지하십시오.

시나리오	Tools	주석
감독자 제어부 백업 및 복원	워크로드 관리 UI에서 vCenter Server 파일 기반 백업 및 복원	<p>vCenter Server에서 스케줄링된 파일 기반 백업의 일부로 vCenter Server에서 감독자의 상태 백업을 구성합니다. 나중에 vSphere Client의 워크로드 관리 인터페이스를 통해 vCenter Server에서 감독자의 상태를 복원할 수 있습니다.</p> <p><b>참고</b> vCenter Server에서 감독자의 상태를 복원하는 것과 vCenter Server의 상태를 복원하는 것은 서로 다른 두 가지 워크플로입니다. vCenter Server를 복원해도 감독자 호스트가 복원되지는 않습니다.</p> <p>장 2 감독자 제어부 백업 및 복원의 내용을 참조하십시오.</p>
vSphere 포드 백업 및 복원	vSphere용 Velero 플러그인	<p>감독자에서 플러그인을 설치하고 구성합니다.</p> <p>장 3 감독자에 vSphere용 Velero 플러그인 설치 및 구성의 내용을 참조하십시오.</p> <p>장 6 vSphere용 Velero 플러그인을 사용하여 vSphere 포드 백업 및 복원의 내용을 참조하십시오.</p>
Tanzu Kubernetes Grid 클러스터에서 상태 비저장 및 상태 저장 워크로드를 백업하고 Tanzu Kubernetes Grid에서 프로비저닝된 클러스터로 복원합니다.	vSphere용 Velero 플러그인	<p>Kubernetes 메타데이터와 영구 볼륨을 둘 다 백업하고 복원합니다.</p> <p>영구 볼륨에 대해 Velero 스냅샷 생성 (Restic 아님)을 사용할 수 있습니다.</p> <p>장 3 감독자에 vSphere용 Velero 플러그인 설치 및 구성의 내용을 참조하십시오.</p> <p>Velero Plugin for vSphere를 사용하여 TKG 2 클러스터 워크로드백업 및 복원을 참조하십시오.</p>

시나리오	Tools	주석
Tanzu Kubernetes Grid 클러스터에서 상태 비저장 및 상태 저장 워크로드를 백업하고 Tanzu Kubernetes Grid에서 프로비저닝되지 않은 적합한 Kubernetes 클러스터로 복원합니다.	독립형 Velero 및 Restic	이식성을 위해 독립형 Velero를 사용합니다. 상태 저장 애플리케이션의 경우 Restic을 포함해야 합니다. 감독자의 TKG 2 클러스터에 독립형 Velero 및 Restic 설치 및 구성을 참조하십시오. 독립형 Velero 및 Restic을 사용하여 감독자의 TKG 2 클러스터에서 워크로드 백업 및 복원을 참조하십시오.
vCenter Server 구성	vCenter Server	vCenter Server가 손실된 경우 vCenter Server를 사용하여 vCenter Server 개체를 백업하고 복원합니다. vCenter Server의 파일 기반 백업 및 복원을 참조하십시오.
NSX	NSX Manager	로드 밸런서 및 수신 서비스는 NSX 백업에 따라 다릅니다. NSX-T Data Center는 NSX Manager 노드 및 개체의 백업 및 복원을 지원하는 제품 내 백업 및 복구를 제공합니다. 자세한 내용은 NSX-T 설명서에서 NSX Manager 백업 및 복원을 참조하십시오.

# 감독자 제어부 백업 및 복원

# 2

vCenter Server 파일 기반 백업의 일부로 vCenter Server에 감독자 상태를 기록하는 옵션을 포함할 수 있습니다. 나중에 생성된 백업 파일에서 감독자 제어부를 복원할 수 있습니다.

다음으로 아래 항목을 읽으십시오.

- 감독자 상태 백업
- 감독자 제어부 복원

## 감독자 상태 백업

환경에서 감독자의 상태를 백업하는 방법을 알아봅니다. vCenter Server 파일 기반 백업의 일부로 vCenter Server에서 사용 가능한 감독자 백업을 포함할 수 있습니다.

감독자 제어부의 백업 파일은 다음 구성 요소의 상태를 캡처합니다.

- etcd 상태.
- vCenter Server 업그레이드 후 제어부 VM을 복원할 수 있도록 인프라 포드에 사용되는 컨테이너 이미지.
- 동일한 CA에서 복원한 후 모든 Kubernetes 인증서를 재생성할 수 있도록 하는 Kubernetes CA 인증서 및 키. 이것이 있으면 복원 후 새 Kubernetes CA를 신뢰하기 위해 vSphere 포드 및 Spherelet을 재구성할 필요가 없습니다.
- 모든 vSphere 네임스페이스 및 워크로드(예: 배포, 포드, VM, TKG 리소스, 영구 볼륨 할당 등)와 관련된 모든 Kubernetes 리소스의 상태.

vCenter Server 파일 기반 백업 및 복원에 대한 자세한 내용은 [vCenter Server의 파일 기반 백업 및 복원](#)을 참조하십시오.

### 사전 요구 사항

- 백업을 저장할 충분한 디스크 공간이 있는 FTP, FTPS, HTTP, HTTPS, SFTP, NFS 또는 SMB 서버가 실행 중이어야 합니다.

### 절차

- 1 웹 브라우저에서 vCenter Server 관리 인터페이스(<https://appliance-IP-address-or-FQDN:5480>)로 이동합니다.

- 2 root로 로그인합니다.
- 3 vCenter Server 관리 인터페이스에서 **백업**을 클릭합니다.
- 4 백업 스케줄이 이미 있는 경우 **편집**을 클릭합니다.

백업 스케줄이 없는 경우 **파일 기반 백업 스케줄링**에서 생성 방법을 참조하십시오.

- 5 [백업 일정 편집] 창에서 **감독자 제어부**를 선택합니다.

### Edit Backup Schedule ×

Backup location * <span style="font-size: small;">(i)</span>	sftp:// <span style="background-color: #eee; border: 1px solid #ccc; padding: 2px;">                    </span> /root/backup	
Backup server credentials *	User name	root
	Password	<input type="password"/>
Schedule <span style="font-size: small;">(i)</span>	Weekly <span style="font-size: small;">▼</span> Sunday <span style="font-size: small;">▼</span>	11 : 59 P.M. Etc/UTC
Encrypt backup	Encryption Password	<input type="password"/>
	Confirm Password	<input type="password"/>
Number of backups to retain *	<input checked="" type="radio"/> Retain all backups	
	<input type="radio"/> Retain last <input style="width: 50px;" type="text" value="0"/> backups	
Data	<input checked="" type="checkbox"/> Supervisors Control Plane	909 MB
	<input checked="" type="checkbox"/> Stats, Events, and Tasks	90 MB
	<input checked="" type="checkbox"/> Inventory and configuration	296 MB
	Total size (compressed)	

CANCEL
SAVE

## 결과

감독자의 모든 vCenter Server 상태가 vCenter Server 백업의 일부로 백업됩니다.

## 감독자 제어부 복원

vCenter Server 시스템 자체의 백업 파일에서 vCenter Server의 감독자 제어부를 복원할 수 있습니다.

**참고** vCenter Server에서 감독자 제어부를 복원하는 것과 vCenter Server의 상태를 복원하는 것은 두 가지 다른 워크플로입니다. vCenter Server를 복원해도 감독자 제어부가 복원되지는 않습니다.



## 사전 요구 사항

- vCenter Server 관리 인터페이스 파일 기반 백업에서 감독자 상태 기록을 구성합니다.

## 절차

- 1 vSphere Client에서 **워크로드 관리**로 이동합니다.
- 2 **감독자**를 선택하고 **복원**을 선택합니다.
- 3 백업 세부 정보를 입력합니다.

옵션	설명
vCenter	감독자를 관리하는 vCenter Server 시스템을 선택합니다.
백업 선택	<ul style="list-style-type: none"> <li>■ <b>백업 서버 폴더 찾아보기</b>를 선택하면 이 vCenter Server 시스템으로 구성된 백업 파일 서버의 루트 폴더에 저장된 파일을 업로드할 수 있습니다.</li> <li>■ 특정 백업 파일을 업로드하려면 <b>단일 백업 위치</b>를 선택한 다음, 해당 백업 파일의 URL을 입력합니다.</li> <li>■ <b>vCenter 백업 스케줄의 백업 위치 및 사용자 이름 사용</b>을 선택하여 vCenter Server로 구성된 백업 위치의 루트 폴더 위치와 사용자 이름을 채웁니다.</li> </ul>
위치	백업 루트 폴더의 위치를 입력합니다.
사용자 이름	백업에 액세스할 사용자 이름을 입력합니다.
암호	해당 사용자 이름의 암호를 입력합니다.

- 4 **다음**을 클릭합니다.
- 5 복원할 백업 파일을 선택하고 **다음**을 클릭하여 백업 파일의 다운로드를 시작합니다.
- 6 복원할 감독자를 선택하고 **다음**을 클릭합니다.
- 7 설정을 검토하고 **완료**를 클릭합니다.

## 결과

감독자는 구성 중 상태로 돌아가고 모든 제어부 VM이 삭제되고 백업 파일의 데이터로 다시 배포됩니다. **구성 상태** 열에서 **보기**를 클릭하여 프로세스를 모니터링할 수 있습니다.

# 감독자에 vSphere용 Velero 플러그인 설치 및 구성

# 3

vSphere 포드 및 TKG 클러스터에서 실행되는 워크로드를 백업하고 복원하도록 vSphere용 Velero 플러그인을 설치하고 구성하는 방법을 알아봅니다.

## 개요

vSphere용 Velero 플러그인은 vSphere IaaS Control Plane 워크로드를 백업하고 복원하기 위한 솔루션을 제공합니다. vSphere용 Velero 플러그인이 감독자에 설치 및 구성되면 TKG 클러스터 워크로드 및 vSphere 포드를 백업 및 복원할 수 있습니다. 영구 워크로드의 경우 vSphere용 Velero 플러그인을 사용하여 영구 볼륨의 스냅샷을 생성할 수 있습니다.

## 사전 요구 사항:

- 감독자가 활성화됩니다.
- vSphere 네임스페이스가 생성 및 구성됩니다.
- vSphere 관리자 역할의 멤버이거나 다음과 같은 vSphere 권한이 있어야 합니다.
  - **SupervisorServices.Manage**
  - **Namespaces.Manage**
  - **Namespaces.Configure**

---

**참고** "vSphere IaaS 제어부에서 TKG 서비스 사용" 에서 [전용 그룹 및 역할 생성](#)을 참조하십시오.

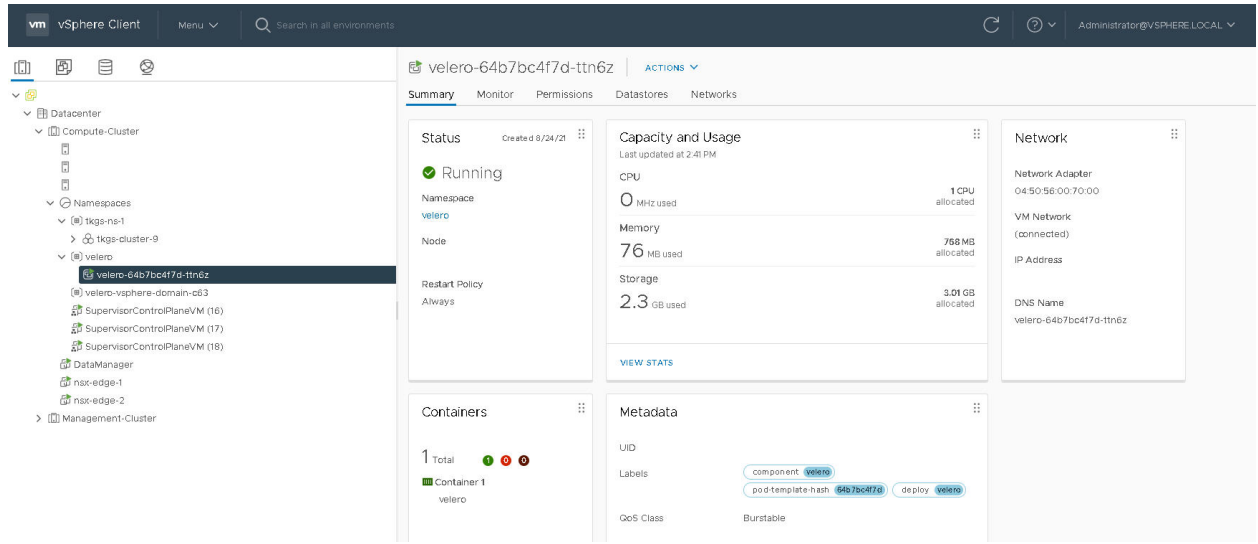
---

- Velero CLI를 실행할 수 있는 Linux VM을 생성합니다. 또는 감독자에 액세스하는 기존 Linux 점프 호스트를 사용합니다.
- Velero 버전 번호는 `x.y.z`로 표시됩니다. 사용할 특정 버전은 [Velero 호환성 매트릭스](#)를 참조하고 명령을 실행할 때 그에 따라 대체하십시오.

이 스크린샷은 vSphere용 Velero 플러그인 설치의 종료 상태를 보여줍니다.

- NSX 네트워킹은 vSphere 포드 배포를 지원하는 데 사용됩니다.
- Data Manager VM이 배포되었습니다.
- Velero Operator가 활성화되고 `velero-vsphere-domain-cxx` 네임스페이스에서 실행되고 있습니다.

- velero라는 네임스페이스가 구성되었습니다.
- vSphere용 Velero 플러그인이 velero 네임스페이스에서 vSphere 포드로 실행되고 있습니다.



## 0단계(선택 사항): 백업 및 복원 트래픽을 위한 전용 네트워크 생성

필수는 아니지만 운영 환경에서는 백업 및 복원 트래픽을 vSphere IaaS Control Plane 관리 네트워크 트래픽과 분리하는 것이 좋습니다. 이 작업에는 두 가지 측면이 있습니다.

- NFC(Network File Copy)를 지원하도록 ESXi 호스트에 태그 지정
- NSX를 사용하여 백업 및 복원 네트워크 구성

전용 NBD(Network Block Device) 전송을 지원하도록 ESXi 호스트를 구성하려면, 감독자가 실행되는 vSphere 클러스터의 각 ESXi 호스트에 VMkernel NIC를 추가하고 이 NIC에 vSphereBackupNFC를 설정합니다. vSphereBackupNFC 태그가 VMkernel 어댑터의 NIC 유형에 적용되면 백업 및 복원 트래픽은 선택한 가상 NIC를 통과합니다.

이 구성을 수행하려면 Virtual Disk Development Kit를 사용합니다. [NBD 설명서](#)를 참조하십시오.

**참고** VMkernel NIC에서 vSphereBackupNFC를 사용하도록 설정하지 않으면 백업 및 복원 트래픽은 백업 및 복원 네트워크를 구성하더라도 이 네트워크를 통해 전송되지 않습니다. vSphereBackupNFC를 사용하도록 설정하지 않으면 트래픽은 vSphere 관리 네트워크를 통해 이동합니다.

vSphereBackupNFC 태그를 사용하도록 설정되면 클러스터에 대한 기존 VDS(vSphere Distributed Switch)를 다음과 같이 업데이트하여 NSX를 사용하는 백업 및 복원 네트워크를 구성합니다.

- vSphere Client에서 **메뉴 > 네트워킹**을 선택합니다.
- 클러스터에 대한 기존 VDS를 선택합니다.
- VDS를 마우스 오른쪽 버튼으로 클릭하고 **분산 포트 그룹 > 새 분산 포트 그룹**을 선택합니다.
- **BackupRestoreNetwork**라는 새 분산 포트 그룹을 생성합니다.

- VMkernel 어댑터를 **BackupRestoreNetwork** 분산 포트 그룹에 추가합니다.
- 워크로드 관리를 사용하도록 설정된 vCenter 클러스터의 모든 ESXi 호스트를 **BackupRestoreNetwork** 분산 포트 그룹에 연결합니다.
- `vSphereBackupNFC` 태그를 사용하도록 설정합니다.

## 1단계: S3 호환 개체 저장소 생성

영구 볼륨의 백업 및 복원을 위해서는 S3 호환 개체 저장소를 제공해야 합니다. Velero는 여러 개체 저장소 제공자를 지원합니다.

vSphere용 Velero 플러그인을 설치하려면 S3 호환 개체 저장소에 대한 다음 정보를 제공해야 합니다.

데이터 항목	예제 값
s3Url	http://my-s3-store.example.com
aws_access_key_id	ACCESS-KEY-ID-STRING
aws_secret_access_key	SECRET-ACCESS-KEY-STRING

다음 정보를 사용하여 암호 파일 이름 `s3-credentials`를 생성합니다. 이 파일은 vSphere용 Velero 플러그인을 설치할 때 참조합니다.

```
[default]
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

MinIO는 설치하고 사용하기 쉬운 S3 호환 개체 저장소입니다. vSphere IaaS Control Plane에는 사용하도록 설정할 수 있는 MinIO 감독자 서비스가 함께 제공됩니다. 자세한 내용은 "vSphere IaaS 제어부 서비스 및 워크로드" 게시물을 참조하십시오.

또는 Linux VM에 MinIO 서버를 수동으로 설치할 수 있습니다. 자세한 내용은 [TKG 클러스터에 독립형 Velero 및 Restic 설치 및 구성](#)에서 참조하십시오.

## 2단계: Data Manager 설치 및 구성

**경고** Data Manager는 기능적으로만 테스트되었으며 대규모로 작동하기 위한 것이 아니며 성능에 대한 기대를 약속하지 않습니다. 운영 환경에서 비즈니스 크리티컬 애플리케이션을 백업하기 위한 것이 아닙니다.

vSphere용 Velero 플러그인을 사용하여 백업 및 복원을 용이하게 하려면 하나 이상의 Data Manager VM을 배포하여 영구 볼륨 백업 데이터를 S3 호환 객체 스토리지 내부 및 외부로 이동합니다. Data Manager는 백업 시에는 vSphere 볼륨에서 원격 지속형 S3 호환 스토리지로, 복원 중에는 원격 S3 호환 스토리지에서 vSphere 볼륨으로 볼륨 스냅샷 데이터를 이동합니다.

vSphere IaaS Control Plane 환경에서 Data Manager를 VM으로 설치합니다.

---

**참고** Velero vSphere Operator를 사용하도록 설정할 때까지 Data Manager VM의 전원을 켜지 마십시오.

---

- 1 vSphere Client를 사용하여 감독자가 활성화된 데이터 센터를 마우스 오른쪽 버튼으로 클릭하고 **OVF 템플릿 배포**를 선택합니다.
- 2 다음 URL에서 Data Manager OVA 파일을 로컬 시스템에 다운로드합니다. <https://vmwareasas.jfrog.io/artifactory/Velero-YAML/Velero/DataManager/1.2.0/datamgr-ob-20797900-photon-3-release-1.2.ova>.
- 3 **로컬 파일**을 선택하고 Data Manager OVA를 vCenter Server에 업로드합니다.
- 4 가상 시스템의 이름을 지정합니다(예: **DataManager**).
- 5 계산 리소스를 선택합니다. 이것은 감독자가 구성된 vSphere 클러스터입니다.
- 6 VM 배포 세부 정보를 검토하고 **다음**을 클릭합니다.
- 7 라이선스 계약에 동의하고 **다음**을 클릭합니다.
- 8 스토리지를 선택하고 **다음**을 클릭합니다.
- 9 Data Manager VM에 대한 대상 네트워크를 선택합니다.
  - 전용 백업 및 복원 네트워크를 구성한 경우 **BackupRestoreNetwork**를 선택합니다.
  - 전용 백업 및 복원 네트워크를 구성하지 않은 경우 **관리 네트워크**를 선택합니다.
- 10 선택 항목을 검토하고 **마침**을 클릭하여 프로세스를 완료합니다.
- 11 최근 작업 패널을 사용하여 배포 진행률을 모니터링합니다.

---

**참고** "OVF 설명자를 사용할 수 없습니다."라는 오류가 표시되면 Chrome 브라우저를 사용하십시오.

---

- 12 Data Manager VM이 배포되면 VM에 대한 입력 매개 변수를 구성합니다.
- 13 VM을 마우스 오른쪽 버튼으로 클릭하고 **설정 편집**을 선택합니다.
- 14 [가상 하드웨어] 탭에서 CD/DVD 드라이브의 경우 **호스트 디바이스**를 **클라이언트 디바이스**로 변경합니다.

---

**참고** 이렇게 하지 않으면 필요한 고급 구성 설정을 저장할 수 없습니다.

---

- 15 **설정 편집 > 고급 매개 변수** 탭에서 **고급 > 구성 매개 변수 편집**을 선택합니다.

16 다음 설정 각각에 대한 입력 매개 변수를 구성합니다.

매개 변수	값
guestinfo.cnsdp.vcUser	VM을 배포할 수 있는 충분한 권한이 있는 vCenter Server 사용자 이름을 입력합니다. vSphere 관리자 사용 권한이 있는 사용자를 지정하지 않는 경우 <a href="#">vSphere 사용 권한</a> 설명서를 참조하십시오. 또는 워크로드 관리를 위한 전용 사용자를 생성합니다. "vSphere IaaS 제어부에서 TKG 서비스 사용" 에서 <a href="#">전용 그룹 및 역할 생성</a> 을 참조하십시오.
guestinfo.cnsdp.vcAddress	vCenter Server IP 주소 또는 FQDN을 입력합니다.
guestinfo.cnsdp.vcPasswd	vCenter Server 사용자 암호를 입력합니다.
guestinfo.cnsdp.vcPort	기본값은 <b>443</b> 입니다. 이 값을 변경하지 마십시오.
guestinfo.cnsdp.wcpControlPlaneIP	감독자 부동 IP 주소를 입력합니다. 이 값은 워크로드 관리에서 감독자로 이동하고 <b>구성 &gt; 네트워크 &gt; 관리 네트워크 &gt; 부동 IP</b> 를 선택하여 을 가져옵니다.
guestinfo.cnsdp.updateKubect1	기본값은 <b>false</b> 입니다. 이 값을 변경하지 마십시오.
guestinfo.cnsdp.veleroNamespace	기본값인 <b>velero</b> 를 그대로 둡니다. 나중에 이 프로세스에서 감독자에 이름이 <b>velero</b> 인 vSphere 네임스페이스를 생성하게 됩니다. 이 두 이름은 일치해야 합니다.
guestinfo.cnsdp.datamgrImage	구성되지 않은 경우(설정되지 않은 경우) 시스템은 기본적으로 Docker Hub( <code>vsphereveleroplugin/data-manager-for-plugin:1.1.0</code> )에서 컨테이너 이미지를 끌어옵니다.

17 **확인**을 클릭하여 구성을 저장하고 **확인**을 다시 클릭하여 VM 설정을 저장합니다.

**참고** CD/DVD 드라이브를 **호스트 디바이스**에서 **클라이언트 디바이스**로 변경하지 않았으면 설정을 저장할 수 없습니다. 이 경우 작업을 취소하고 드라이브를 변경한 후 고급 구성 설정을 반복합니다.

18 Velero vSphere Operator(다음 섹션)를 사용하도록 설정할 때까지 Data Manager VM의 전원을 켜지 마십시오.

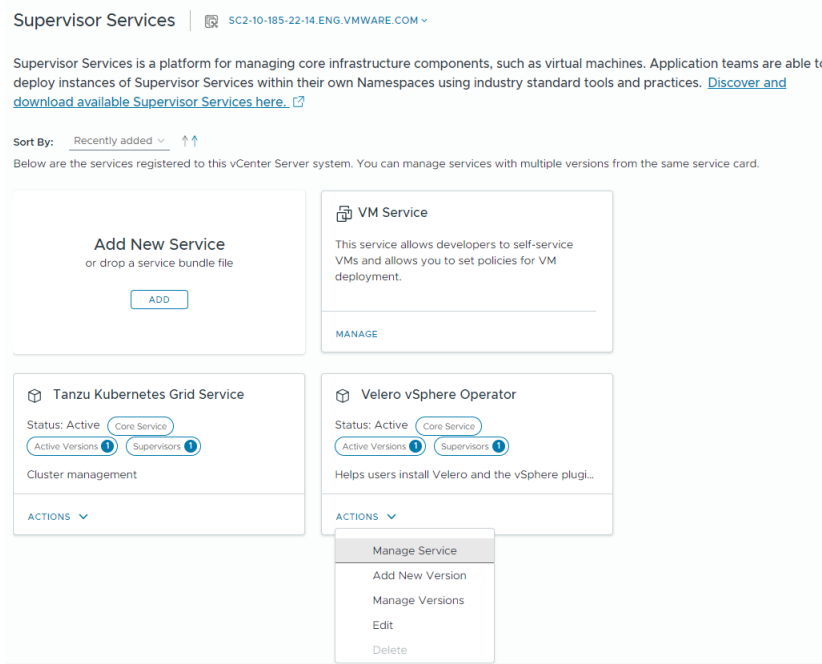
### 3단계: 감독자에 Velero vSphere Operator 서비스 설치

vSphere IaaS Control Plane는 Velero vSphere Operator를 감독자 서비스로 제공합니다. Velero vSphere Operator 서비스는 vSphere용 Velero 플러그인과 함께 작동하여 영구 볼륨 스냅샷 생성을 포함한 Kubernetes 워크로드의 백업 및 복원을 지원합니다. 감독자 서비스에 대한 자세한 내용은 "vSphere IaaS 제어부 서비스 및 워크로드" 에서 [감독자 서비스 관리](#)를 참조하십시오.

Velero vSphere Operator는 핵심 감독자 서비스입니다. 즉, 서비스 운영자가 vCenter Server에 미리 등록되어 제공됩니다. 감독자에서 서비스로 Velero vSphere Operator를 설치하는 단계를 완료합니다.

- 1 vSphere Client 홈 메뉴에서 **워크로드 관리**를 선택합니다.
- 2 **서비스** 탭을 선택합니다.
- 3 맨 위에 있는 드롭다운 메뉴에서 대상 vCenter Server를 선택합니다.

#### 4 Velero vSphere Operator 카드에서 **작업 > 서비스 관리**를 선택합니다.



5 서비스를 설치할 대상 감독자를 선택하고 **다음**을 클릭합니다.

6 **마침**을 클릭하여 서비스 설치를 완료합니다.

감독자에서 Velero vSphere Operator 서비스를 확인하고 Data Manager VM을 시작합니다.

1 vSphere Client 홈 메뉴에서 **워크로드 관리**를 선택합니다.

2 **서비스**를 선택합니다.

3 Velero vSphere Operator가 설치되어 있고 해당 상태가 **구성됨**인지 확인합니다.

4 **네임스페이스** 탭에 `svc-velero-vsphere-domain-xxx`(xxx는 고유한 영숫자 토큰임)라는 새 vSphere 네임스페이스 네임스페이스가 보이는지 확인합니다. 이것은 시스템에서 Velero vSphere Operator용으로 생성된 네임스페이스입니다.

**참고** 이 네임스페이스는 구성할 필요가 없으며 편집해서는 안 됩니다.

5 **호스트 및 클러스터**에서 Data Manager VM을 찾고 VM의 전원을 켭니다.

## 4단계: vSphere용 Velero 플러그인용 vSphere 네임스페이스 생성

vSphere Client를 사용하여 감독자에 vSphere 네임스페이스를 수동으로 생성합니다. 이 vSphere 네임스페이스는 vSphere용 Velero 플러그인에 필요합니다.

- vSphere 네임스페이스의 이름을 **velero**로 지정합니다.
- **velero** 네임스페이스를 선택하여 구성합니다.
- **velero** 네임스페이스에 대한 스토리지를 지정합니다.

- 적절한 권한이 있는 사용자에게 **velero** 네임스페이스에 대한 편집 권한을 부여합니다.

## 5단계: vSphere용 Velero 플러그인 Configmap 생성

velero-vsphere-plugin-config.yaml이라는 vSphere용 Velero 플러그인에 대한 configmap을 생성합니다.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: velero-vsphere-plugin-config
data:
  cluster_flavor: SUPERVISOR
```

configmap을 감독자에 적용합니다.

```
kubectl apply -n <velero-namespace> -f velero-vsphere-plugin-config.yaml
```

configmap을 설치하지 않으면 vSphere용 Velero 플러그인을 설치하려고 할 때 다음 오류가 표시됩니다.

```
Error received while retrieving cluster flavor from config, err: configmaps "velero-vsphere-
plugin-config" not found
Falling back to retrieving cluster flavor from vSphere CSI Driver Deployment
```

## 6단계: vSphere용 Velero 플러그인 설치

이제 vSphere용 Velero 플러그인을 설치할 준비가 되었습니다. 이를 위해 **velero-vsphere** CLI를 다운로드하여 실행합니다.

**참고** 이 절차를 수행하려면 Linux VM이 필요합니다. `kubectl-vsphere` 및 `kubectl` CLI를 실행하는 Linux 점프 호스트에 **velero-vsphere** 바이너리를 다운로드합니다.

- 1 vSphere용 Velero 플러그인 CLI를 다운로드합니다.

호환성 매트릭스를 확인하고 다음 위치에서 대상 버전을 다운로드합니다. <https://github.com/vmware-tanzu/velero-plugin-for-vsphere/releases>.

**참고** 다음 명령에서 `x.y.z`를 다운로드한 Velero CLI 및 플러그인 버전으로 바꿉니다.

- 2 CLI를 Linux 점프 호스트에 안전하게 복사합니다. 예:

```
pscp -P 22 C:\temp\velero-vsphere-X.Y.Z-linux-amd64.tar.gz ubuntu@10.117.29.131:/home/
ubuntu/tanzu
```

- 3 **velero-vsphere** CLI를 추출하고 쓰기 가능으로 만듭니다.

```
tar -xf velero-vsphere-X.Y.Z-linux-amd64.tar.gz
chmod +x velero-vsphere
```



- 4 경로에 CLI를 추가합니다.

```
export PATH="$ (pwd) /velero-vsphere-X.Y.Z-linux-amd64:$PATH"
```

- 5 다음 내용으로 s3-credentials 파일을 생성합니다.

```
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

- 6 S3 호환 개체 저장소의 영역, URL 및 버킷 이름을 가져옵니다.  
 7 kubectl용 vSphere 플러그인을 사용하여 감독자에 로그인합니다.  
 8 컨텍스트를 감독자로 전환합니다.

```
kubectl config use-context SUPERVISOR-CLUSTER-IP-ADDRESS
```

- 9 다음 velero-vsphere CLI 명령을 실행하여 직접 생성한 **velero** 네임스페이스에 vSphere용 Velero 플러그인을 설치합니다.

AWS \$BUCKET 및 \$REGION 값을 내보냅니다. 앞의 지침에서 벗어난 값(암호 파일의 이름 또는 위치, 수동으로 생성된 velero 네임스페이스의 이름 등)이 있으면 해당 값도 조정합니다.

```
export BUCKET=example-velero-sv && export REGION=us-east-1

./velero-vsphere install \
  --namespace velero \
  --version vX.X.X \
  --provider aws \
  --plugins harbor-repo.vmware.com/velero/velero-plugin-for-aws:vX.Y.Z,harbor-
repo.vmware.com/velero/velero-plugin-for-vsphere:vX.Y.Z \
  --bucket $BUCKET \
  --secret-file ~/.aws/credentials \
  --snapshot-location-config region=$REGION \
  --backup-location-config region=$REGION
```

**참고** 예를 들어 vSphere용 Velero 플러그인 v1.4.0이 사용되는 경우 Velero CLI 버전은 v1.8.1입니다.

- 10 vSphere용 Velero 플러그인이 성공적으로 설치되었는지 확인합니다.

설치에 성공하면 다음 메시지가 표시됩니다.

```
Send the request to the operator about installing Velero in namespace velero
```

다음 명령을 실행하여 추가로 확인합니다. "Completed" 및 버전이 표시되어야 합니다.

```
kubectl -n velero get veleroservice default -o json | jq '.status'
```

예상 결과:

```
{
  "enabled": true,
  "installphase": "Completed",
  "version": "v1.8.1"
}
```

**참고** 위의 명령은 터미널로 전송되는 JSON 출력의 형식을 지정하는 jq 유틸리티가 설치되어 있다고 가정합니다. jq가 설치되어 있지 않으면 설치하거나 명령에서 해당 부분(jq 이후의 모든 항목)을 제거합니다.

- 필요에 따라 문제를 해결합니다.

설치가 실패하면 설치를 제거하고 다시 시도합니다. 설치를 제거하려면 다음 섹션의 단계를 나열된 순서대로 완료합니다.

## 부록: vSphere용 Velero 플러그인 제거

vSphere용 Velero 플러그인을 제거하려면 다음 단계를 완료합니다.

- velero-vsphere CLI를 실행하여 vSphere용 Velero 플러그인을 제거합니다.

```
./velero-vsphere uninstall -n velero
```

- velero라는 vSphere 포드가 제거되었는지 확인합니다.

```
kubectl get pods -n velero
```

포드가 "종료 중"이라고 표시되면 제거될 때까지 기다렸다가 계속합니다.

- vSphere Client를 사용하여 수동으로 생성한 velero라는 vSphere 네임스페이스를 삭제합니다.

**참고** 네임스페이스 삭제가 완료될 때까지 다음 단계를 진행하지 마십시오. kubectl을 사용하여 velero 네임스페이스가 제거되었는지 확인할 수 있습니다. (하지만 kubectl을 사용하여 velero 네임스페이스를 제거하지는 마십시오.)

## 부록: 에어갭 환경에 vSphere용 Velero 플러그인 설치

에어갭 환경에 vSphere용 Velero 플러그인을 설치하려는 경우에는 사용자 지정된 이미지로 설치해야 합니다. 사용자 지정된 이미지의 일치하는 backup-driver 및 data-manager-for-plugin 이미지를 필요한 레지스트리에서 사용할 수 있고 Kubernetes 클러스터에서 액세스할 수 있는지 확인해야 합니다. 에어갭 환경에서는 Docker Hub의 릴리스된 이미지에 액세스할 수 없으므로 개인 레지스트리의 사용자 지정된 이미지가 필요합니다.

이 플러그인을 설치하려면 다음 단계를 수행하십시오.

- 1 `velero-plugin-for-vsphere`, `backup-driver` 및 `data-manager-for-plugin`의 릴리스된 이미지를 다운로드합니다.
- 2 이미지의 이름을 변경합니다. 즉, 일치하는 `<Registry endpoint and path>` 및 `<Version tag>`로 이미지에 태그를 지정하고 사용자 지정된 저장소에 업로드합니다.
- 3 사용자 지정된 `velero-plugin-for-vsphere` 이미지를 사용하여 플러그인을 설치합니다.

vanilla 클러스터에 vSphere용 Velero 플러그인을 설치하면 두 개의 추가 구성 요소인 `backup-driver` 배포와 `data-manager-for-plugin` DaemonSet가 백그라운드에 배포됩니다. 감독자 및 Tanzu Kubernetes 클러스터에서는 `backup-driver` 배포만 배포됩니다.

`velero-plugin-for-vsphere`의 컨테이너 이미지를 제공하면 일치하는 `backup-driver` 및 `data-manager-for-plugin` 이미지는 이미지 구문 분석 메커니즘을 사용하여 구문 분석됩니다.

컨테이너 이미지는 다음 패턴으로 공식화됩니다.

```
<Registry endpoint and path>/<Container name>:<Version tag>
```

`velero-plugin-for-vsphere` 컨테이너 이미지를 제공하면 일치하는 `<Registry endpoint and path>` 및 `<Version tag>`와 함께 `backup-driver` 및 `data-manager-for-plugin`의 해당 이미지가 구문 분석됩니다.

예를 들어 다음 `velero-plugin-for-vsphere` 컨테이너 이미지를 고려합니다.

```
abc.io:8989/x/y/.../z/velero-plugin-for-vsphere:vX.Y.Z
```

`backup-driver` 및 `data-manager-for-plugin`의 일치하는 다음 이미지를 끌어와야 합니다.

```
abc.io:8989/x/y/.../z/backup-driver:vX.Y.Z
abc.io:8989/x/y/.../z/data-manager-for-plugin:vX.Y.Z
```

- 4 설치 문제를 해결합니다.

`backup-driver` 및 `data-manager-for-plugin`의 일치하는 이미지를 구문 분석하는 데 문제가 있거나 오류가 있는 경우 설치하는 Docker Hub에 있는 공식 `velerovsphereplugin` 저장소의 해당 이미지로 폴백됩니다. 다음 문제는 폴백 메커니즘을 트리거합니다.

- a 사용자 입력의 사용자 지정된 `velero-plugin-for-vsphere` 이미지에 예기치 않은 컨테이너 이름이 사용됩니다.

예를 들어 `x/y/velero-velero-plugin-for-vsphere:vX.Y.Z`이 사용됩니다.

- b Velero 배포 이름이 `velero` 이외의 이름으로 사용자 지정됩니다. 예를 들어 Velero를 배포하기 전에 Velero 배포 이름이 `Velero manifests` 파일에서 `velero-server`로 업데이트되면 문제가 트리거됩니다.

`velero-plugin-for-vsphere`의 기존 이미지 구문 분석 메커니즘은 고정된 이름인 `velero`를 사용하는 Velero 배포만 인식할 수 있습니다.

# TKG 서비스 클러스터 및 워크로드 백업 및 복원

TKG 서비스 클러스터 및 워크로드를 백업하고 복원하려면 이 섹션을 참조하십시오.

다음으로 아래 항목을 읽으십시오.

- TKG 서비스 클러스터 및 워크로드 백업 및 복원에 대한 고려 사항
- vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원
- Restic과 함께 독립형 Velero를 사용하여 감독자에서 TKG 클러스터 워크로드 백업 및 복원
- CSI 스냅샷과 함께 Velero를 사용하여 백업 및 복원

## TKG 서비스 클러스터 및 워크로드 백업 및 복원에 대한 고려 사항

이 항목에서는 TKG 서비스 클러스터에서 실행되는 워크로드 백업 및 복원에 대한 고려 사항을 제공합니다.

### TKG 서비스 클러스터 백업 및 복원

TKG 클러스터를 백업하고 복원하려면 감독자 데이터베이스를 백업합니다. 이렇게 하면 vSphere 네임스페이스 개체 및 TKG 클러스터 노드 VM을 복원할 수 있습니다.

vCenter Server 관리 인터페이스를 통해 사용할 수 있는 vCenter Server 백업 기능을 사용하여 감독자 백업 및 복원을 사용하도록 설정합니다. 자세한 내용은 vSphere IaaS Control Plane에 대한 백업 복원 자료를 참조하십시오.

---

**참고** 감독자 백업은 TKG 클러스터 노드 VM을 복원하는 데에만 사용할 수 있습니다. 감독자 백업을 사용하여 TKG 클러스터에 배포된 워크로드를 복원할 수 없습니다. 워크로드를 별도로 백업한 다음, 클러스터가 복원된 후에 워크로드를 복원해야 합니다.

---

### TKG 서비스 클러스터에서 실행되는 워크로드 백업 및 복원

이 표에는 TKG 클러스터에서 실행되는 상태 비저장 및 상태 저장 워크로드를 백업 및 복원하기 위한 옵션이 요약되어 있습니다.

---

**참고** 독립형 Velero를 사용하여 Kubernetes 클러스터를 백업하고 복원하면 이식성이 지원됩니다. 즉, TKG 서비스에 의해 프로비저닝되지 않은 Kubernetes 클러스터로 클러스터 워크로드를 복원하려면 독립형 Velero를 사용하는 것이 좋습니다.

---

시나리오	도구	주석
감독자의 TKG 클러스터에서 상태 비 저장 및 상태 저장 워크로드를 백업하고 감독자의 TKG 클러스터로 복원합니다.	Velero Plugin for vSphere vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원의 내용을 참조하십시오.	Kubernetes 메타데이터와 영구 볼륨을 모두 백업하고 복원할 수 있습니다. Velero 스냅샷 생성은 상태 저장 애플리케이션이 있는 영구 볼륨에 사용됩니다. vSphere용 Velero 플러그인도 감독자에 설치 및 구성되어 있어야 합니다.
감독자의 TKG 클러스터에서 상태 비 저장 및 상태 저장 워크로드를 백업하고 규격 Kubernetes 클러스터로 복원합니다.	독립형 Velero 및 Restic Restic과 함께 독립형 Velero를 사용하여 감독자에서 TKG 클러스터 워크로드 백업 및 복원 항목을 참조하십시오.	Kubernetes 메타데이터와 영구 볼륨을 모두 백업하고 복원할 수 있습니다. Restic은 상태 저장 애플리케이션이 있는 영구 볼륨에 사용됩니다. 이식성이 필요한 경우 이 방식을 사용합니다.
감독자의 TKG 클러스터에서 상태 비 저장 및 상태 저장 워크로드를 백업하고 규격 Kubernetes 클러스터로 복원합니다.	CSI 스냅샷 생성 기능을 갖춘 독립형 Velero CSI 스냅샷과 함께 Velero를 사용하여 백업 및 복원 항목을 참조하십시오.	vSphere 8.0에는 vSphere 8.0 U2+ 및 TKr v1.26+가 필요합니다.

## vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원

이 섹션에서는 vSphere용 Velero 플러그인을 사용하여 감독자에서 실행되는 TKG 클러스터 워크로드를 백업 및 복원하는 방법에 대한 항목을 제공합니다.

### TKG 클러스터에 vSphere용 Velero 플러그인 설치 및 구성

TKGS 클러스터에 vSphere용 Velero 플러그인을 설치하면 vSphere용 Velero 플러그인을 사용하여 해당 클러스터에서 실행되는 워크로드를 백업하고 복원할 수 있습니다.

#### 개요

vSphere용 Velero 플러그인은 TKG 서비스 클러스터 워크로드를 백업하고 복원하기 위한 솔루션을 제공합니다. 영구 워크로드의 경우 vSphere용 Velero 플러그인을 사용하여 영구 볼륨의 스냅샷을 생성할 수 있습니다.

**참고** 백업 및 복원하려는 TKG 서비스 클러스터 워크로드에 대해 이식성이 필요한 경우에는 vSphere용 Velero 플러그인을 사용하지 마십시오. Kubernetes 클러스터 간 이식성을 위해서는 독립형 Velero를 Restic과 함께 사용하십시오.

#### 사전 요구 사항: 감독자에 vSphere용 Velero 플러그인 설치

TKGS 클러스터에 vSphere용 Velero 플러그인을 설치하려면 감독자에 vSphere용 Velero 플러그인이 설치되어 있어야 합니다. 또한 감독자가 NSX 네트워킹으로 구성되어야 합니다. [장 3 감독자에 vSphere용 Velero 플러그인 설치 및 구성](#)의 내용을 참조하십시오.

## 스토리지 요구 사항

TKG 서비스 클러스터 백업을 수행하려면 여기에 설명된 대로 스토리지 백엔드가 필요합니다. 여러 클러스터를 백업하는 경우에는 서로 다른 클러스터 백업에 동일한 스토리지 백엔드를 사용하면 안 됩니다. 스토리지 백엔드를 공유하면 백업 개체가 동기화됩니다. 데이터 이스케이프를 방지하려면 다른 스토리지 백엔드를 사용해야 합니다.

### 1단계: Linux Workstation에서 Velero CLI 설치

Velero CLI는 Velero와 상호 작용하기 위한 표준 도구입니다. Velero CLI는 vSphere용 Velero 플러그인 CLI(`velero-vsphere`)보다 더 많은 기능을 제공하며 Tanzu Kubernetes 클러스터 워크로드를 백업하고 복원하는 데 필요합니다.

Linux 워크스테이션에 Velero CLI를 설치합니다. 이상적으로는 `kubectl`, `kubectl-vsphere`, `velero-vsphere`를 포함한 vSphere IaaS Control Plane 환경에 대해 연결된 CLI를 실행하는 Linux와 동일한 점프 호스트입니다.

Velero 버전 번호는 `X.Y.Z`로 표시됩니다. 사용할 특정 버전은 [Velero 호환성 매트릭스](#)를 참조하고 명령을 실행할 때 그에 따라 대체하십시오.

Velero CLI를 설치하려면 다음 단계를 완료합니다.

- 1 다음 명령을 실행합니다.

```
$ wget https://github.com/vmware-tanzu/velero/releases/download/vX.Y.Z/velero-vX.Y.Z-linux-amd64.tar.gz
$ gzip -d velero-vX.Y.Z-linux-amd64.tar.gz && tar -xvf velero-vX.Y.Z-linux-amd64.tar
$ export PATH="$ (pwd) /velero-vX.Y.Z-linux-amd64:$PATH"

$ which velero
/root/velero-vX.Y.Z-linux-amd64/velero
```

- 2 Velero CLI 설치를 확인합니다.

```
velero version

Client:
  Version: vX.Y.Z
```

### 2단계: S3 호환 버킷 세부 정보 가져오기

편의를 위해 이 단계에서는 감독자에 vSphere용 Velero 플러그인을 설치할 때 구성한 것과 동일한 S3 호환 개체 저장소를 사용 중이라고 가정합니다. 운영 환경에서는 별도의 개체 저장소를 생성하는 것이 좋습니다.

vSphere용 Velero 플러그인을 설치하려면 S3 호환 개체 저장소에 대한 다음 정보를 제공해야 합니다.

데이터 항목	예제 값
s3Url	<code>http://my-s3-store.example.com</code>
aws_access_key_id	ACCESS-KEY-ID-STRING
aws_secret_access_key	SECRET-ACCESS-KEY-STRING

다음 정보를 사용하여 암호 파일 이름 `s3-credentials`를 생성합니다. 이 파일은 vSphere용 Velero 플러그인을 설치할 때 참조합니다.

```
aws_access_key_id = ACCESS-KEY-ID-STRING
aws_secret_access_key = SECRET-ACCESS-KEY-STRING
```

### 3단계 옵션 A: 레이블을 사용하여 TKG 클러스터에 vSphere용 Velero 플러그인 설치(새로운 방법)

vSphere 8 업데이트 3 이상을 사용하는 경우 레이블을 추가하여 TKG 클러스터에 vSphere용 Velero 플러그인을 자동으로 설치할 수 있습니다.

- 1 백업 스토리지 위치에 액세스할 수 있는지 확인합니다.
- 2 Velero vSphere Operator Core 감독자 서비스가 활성화되었는지 확인합니다.

```
kubectl get ns | grep velero
svc-velero-domain-c9          Active   18d
```

- 3 이름이 `velero`인 Kubernetes 네임스페이스가 감독자에 생성되었는지 확인합니다.

```
kubectl get ns | grep velero
svc-velero-domain-c9          Active   18d
velero                        Active   1s
```

- 4 감독자에서 vSphere용 Velero 플러그인 감독자 서비스를 사용하도록 설정되어 있는지 확인합니다.

```
velero version
Client:
  Version: v1.11.1
  Git commit: bdb7eb242b0f64d5b04a7fea86d1edbb3a3587c
Server:
  Version: v1.11.1
```

```
kubectl get veleroservice -A
NAMESPACE  NAME      AGE
velero     default  53m
```

```
velero backup-location get
NAME      PROVIDER  BUCKET/PREFIX  PHASE      LAST VALIDATED          ACCESS
MODE     DEFAULT
default  aws      velero        Available  2023-11-20 14:10:57 -0800 PST
ReadWrite true
```

- 5 클러스터에 `velero` 레이블을 추가하여 대상 TKG 클러스터에 대해 Velero를 사용하도록 설정합니다.

```
kubectl label cluster CLUSTER-NAME --namespace CLUSTER-NS velero.vsphere.vmware.com/
enabled=true
```

**참고** 이 작업은 클러스터가 프로비저닝될 때 vSphere 네임스페이스에서 수행됩니다.

## 6 Velero가 설치되고 클러스터에 사용할 준비가 되었는지 확인합니다.

```
kubectl get ns
NAME                                STATUS   AGE
...
velero                              Active  2m   <--
velero-vsphere-plugin-backupdriver  Active  2d23h
```

```
kubectl get all -n velero
NAME                                READY   STATUS    RESTARTS   AGE
pod/backup-driver-5945d6bcd4-gtw9d  1/1     Running   0           17h
pod/velero-6b9b49449-pq6b4         1/1     Running   0           18h
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/backup-driver       1/1     1             1           17h
deployment.apps/velero              1/1     1             1           18h
NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/backup-driver-5945d6bcd4  1         1         1       17h
replicaset.apps/velero-6b9b49449         1         1         1       18h
```

```
velero version
Client:
  Version: v1.11.1
  Git commit: bdb7eb242b0f64d5b04a7fea86d1edbb3a3587c
Server:
  Version: v1.11.1
```

### 3단계 옵션 B: 수동으로 TKG 클러스터에 vSphere용 Velero 플러그인 설치(기존 방법)

Velero CLI를 사용하여 백업 및 복원하려는 대상 TKG 클러스터에 vSphere용 Velero 플러그인을 설치하려고 합니다.

Velero CLI 컨텍스트는 `kubectl` 컨텍스트를 자동으로 따릅니다. Velero CLI 명령을 실행하여 대상 클러스터에 Velero 및 vSphere용 Velero 플러그인을 설치하기 전에 `kubectl` 컨텍스트를 대상 클러스터로 설정해야 합니다.

- 1 `kubectl`용 vSphere 플러그인을 사용하여 감독자를 인증합니다.
- 2 `kubectl` 컨텍스트를 대상 TKG 클러스터로 설정합니다.

```
kubectl config use-context TARGET-TANZU-KUBERNETES-CLUSTER
```

- 3 TKG 클러스터에서 `velero-vsphere-plugin-config.yaml`이라는 Velero 플러그인에 대한 configmap을 생성합니다.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: velero-vsphere-plugin-config
data:
  cluster_flavor: GUEST
```



TKG 클러스터에 configmap을 적용합니다.

```
kubectl apply -n <velero-namespace> -f velero-vsphere-plugin-config.yaml
```

configmap을 설치하지 않으면 vSphere용 Velero 플러그인을 설치하려고 할 때 다음 오류가 표시됩니다.

```
Error received while retrieving cluster flavor from config, err: configmaps "velero-
vsphere-plugin-config" not found
Falling back to retrieving cluster flavor from vSphere CSI Driver Deployment
```

- 4 다음 Velero CLI 명령을 실행하여 대상 클러스터에 Velero를 설치합니다.

**BUCKET-NAME**, **REGION**(두 개의 인스턴스) 및 **s3Url** 필드에 대한 자리 표시자 값을 적절한 값으로 바꿉니다. 앞의 지침에서 벗어난 값(암호 파일의 이름 또는 위치, 수동으로 생성된 `velero` 네임스페이스의 이름 등)이 있으면 해당 값도 조정합니다.

```
./velero install --provider aws \
--bucket BUCKET-NAME \
--secret-file ./s3-credentials \
--features=EnableVSphereItemActionPlugin \
--plugins velero/velero-plugin-for-aws:vX.Y.Z \
--snapshot-location-config region=REGION \
--backup-location-config region=REGION,s3ForcePathStyle="true",s3Url=http://my-s3-
store.example.com
```

- 5 대상 클러스터에 vSphere용 Velero 플러그인을 설치합니다. 설치된 Velero는 Kubernetes API 서버와 통신하여 플러그인을 설치합니다.

```
velero plugin add vsphereveleroplugin/velero-plugin-for-vsphere:vX.Y.Z
```

## 부록: TKG 클러스터에서 vSphere용 Velero 플러그인 제거

vSphere용 Velero 플러그인을 제거하려면 다음 단계를 완료합니다.

- 1 `kubectl` 컨텍스트를 대상 Tanzu Kubernetes 클러스터로 전환합니다.

```
kubectl config use-context TARGET-TANZU-KUBERNETES-CLUSTER
```

- 2 플러그인을 제거하려면 다음 명령을 실행하여 Velero 배포에서 `velero-plugin-for-vsphere`의 `InitContainer`를 제거합니다.

```
velero plugin remove vsphereveleroplugin/velero-plugin-for-vsphere:vX.Y.Z
```

3 프로세스를 완료하려면 백업 드라이버 배포 및 관련 CRD를 삭제합니다.

```
kubectl -n velero delete deployment.apps/backup-driver
```

```
kubectl delete crds \
  backuprepositories.backupdriver.cnsdp.vmware.com \
  backuprepositoryclaims.backupdriver.cnsdp.vmware.com \
  clonefromsnapshots.backupdriver.cnsdp.vmware.com \
  deletesnapshots.backupdriver.cnsdp.vmware.com \
  snapshots.backupdriver.cnsdp.vmware.com
```

```
kubectl delete crds uploads.datamover.cnsdp.vmware.com downloads.datamover.cnsdp.vmware.com
```

## vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드 백업 및 복원

vSphere용 Velero 플러그인을 사용하여 감독자의 TKG 클러스터에서 실행되는 워크로드를 백업 및 복원할 수 있습니다.

### 사전 요구 사항

vSphere용 Velero 플러그인을 사용하여 TKG 클러스터 워크로드를 백업 및 복원하려면 먼저 대상 클러스터에 vSphere용 Velero 플러그인을 설치해야 합니다. [TKG 클러스터에 vSphere용 Velero 플러그인 설치 및 구성](#)의 내용을 참조하십시오.

### 워크로드 백업

다음은 Velero 백업을 생성하는 명령의 예입니다.

```
velero backup create <backup name> --include-namespaces=my-namespace
```

모든 로컬 스냅샷이 생성되고 Kubernetes 메타데이터(볼륨 스냅샷 제외)가 개체 저장소에 업로드되면 Velero 백업이 Completed로 표시됩니다. 이 시점에 비동기 데이터 이동 작업(즉 볼륨 스냅샷 업로드)이 백그라운드에서 계속 진행되며 완료하는 데 다소 시간이 걸릴 수 있습니다. 볼륨 스냅샷의 상태는 [스냅샷 CR\(사용자 지정 리소스\)](#)을 모니터링하여 확인할 수 있습니다.

### 스냅샷

스냅샷은 영구 볼륨을 백업하는 데 사용됩니다. 각 볼륨 스냅샷에 대해 스냅샷이 생성된 PVC(영구 볼륨 할당)와 동일한 네임스페이스에 스냅샷 CR이 생성됩니다.

다음 명령을 실행하여 PVC 네임스페이스의 모든 스냅샷을 가져올 수 있습니다.

```
kubectl get -n <pvc namespace> snapshot
```

스냅샷 CRD(사용자 지정 리소스 정의)에는 `.status.phase` 필드에 대해 다음을 포함한 여러 단계가 있습니다.

스냅샷 단계	설명
신규	아직 처리되지 않았습니다.
Snapshotted	로컬 스냅샷이 생성되었습니다.
SnapshotFailed	로컬 스냅샷을 생성하지 못했습니다.
Uploading	스냅샷이 업로드되고 있습니다.
Uploaded	스냅샷이 업로드되었습니다.
UploadFailed	스냅샷을 업로드하지 못했습니다.
Canceling	스냅샷 업로드가 취소되고 있습니다.
Canceled	스냅샷 업로드가 취소되었습니다.
CleanupAfterUploadFailed	스냅샷 업로드 후 로컬 스냅샷을 정리하지 못했습니다.

## 워크로드 복원

다음은 Velero 복원 명령의 예입니다.

```
velero restore create --from-backup <velero-backup-name>
```

볼륨 스냅샷 및 기타 Kubernetes 메타데이터가 현재 클러스터에 성공적으로 복원되면 Velero 복원이 Completed로 표시됩니다. 이때 이 복원과 관련된 vSphere 플러그인의 모든 작업도 완료됩니다. Velero 백업의 경우처럼 백그라운드에서 비동기식 데이터 이동 작업이 없습니다.

## CloneFromSnapshot

각 볼륨 스냅샷에서 복원하기 위해 원래 스냅샷이 생성된 PVC와 동일한 네임스페이스에 CloneFromSnapshot CR(사용자 지정 리소스)이 생성됩니다. 다음 명령을 실행하여 PVC 네임스페이스의 모든 CloneFromSnapshot 을 가져올 수 있습니다.

```
kubectl -n <pvc namespace> get clonefromsnapshot
```

CloneFromSnapshot CRD에는 `.status.phase` 필드에 대한 몇 가지 주요 단계가 있습니다.

스냅샷 단계	설명
신규	스냅샷에서 복제가 완료되지 않았습니다.
InProgress	원격 저장소에서 vSphere 볼륨 스냅샷을 다운로드하고 있습니다.
Completed	스냅샷에서 복제가 완료되었습니다.
실패	스냅샷에서 복제에 실패했습니다.

## Restic과 함께 독립형 Velero를 사용하여 감독자에서 TKG 클러스터 워크로드 백업 및 복원

이 섹션에서는 Restic과 함께 독립형 Velero를 사용하여 감독자에서 실행되는 TKG 클러스터 워크로드를 백업 및 복원하는 방법에 대한 항목을 제공합니다.

## TKG 클러스터에 독립형 Velero 및 Restic 설치 및 구성

감독자의 TKG 클러스터에서 실행되는 워크로드를 백업 및 복원하려면 데이터스토어를 생성하고 Kubernetes 클러스터에 Velero를 Restic과 함께 설치합니다.

### 개요

TKG 클러스터는 가상 시스템 노드에서 실행됩니다. TKG 클러스터 워크로드를 백업하고 복원하려면 클러스터에 Velero 및 Restic을 설치합니다.

### 사전 요구 사항

Tanzu Kubernetes 클러스터에서 실행되는 워크로드를 백업하고 복원하기 위해 Velero 및 Restic을 설치하려면 환경이 다음 사전 요구 사항을 충족하는지 확인합니다.

- 여러 워크로드 백업을 저장하기에 충분한 스토리지가 있는 Linux VM. 이 VM에 MinIO를 설치합니다.
- kubectl용 vSphere 플러그인 및 kubectl을 포함하여 vSphere에 대한 Kubernetes CLI 도구가 설치되어 있는 Linux VM. 이 클라이언트 VM에 Velero CLI를 설치합니다. 이러한 VM이 없는 경우 Velero CLI를 로컬로 설치할 수 있지만 그에 따라 설치 단계를 조정해야 합니다.
- Kubernetes 환경을 인터넷에 액세스할 수 있으며 클라이언트 VM에서 연결할 수 있습니다.

### MinIO 개체 저장소 설치 및 구성

Velero에는 Kubernetes 워크로드 백업의 대상으로 S3 호환 개체 저장소가 필요합니다. Velero는 이러한 **개체 저장소 제공자**를 여러 개 지원합니다. 간단히 하기 위해 이 지침에서는 개체 저장소 VM에서 로컬로 실행되는 S3 호환 스토리지 서비스인 **MinIO**를 사용합니다.

- 1 MinIO를 설치합니다.

```
wget https://dl.min.io/server/minio/release/linux-amd64/minio
```

- 2 MinIO에 실행 권한을 부여합니다.

```
chmod +x minio
```

- 3 MinIO용 파일 시스템에 디렉토리를 생성합니다.

```
mkdir /DATA-MINIO
```

- 4 MinIO 서버를 시작합니다.

```
./minio server /DATA-MINIO
```

- 5 MinIO 서버가 시작되면 끝점 URL, AccessKey 및 SecretKey를 비롯한 중요 데이터스토어 인스턴스 세부 정보가 제공됩니다. 테이블에 끝점 URL, AccessKey 및 SecretKey를 기록하십시오.

데이터스토어 메타데이터	값
끝점 URL	
AccessKey	
SecretKey	

- 6 브라우저에서 MinIO 서버 끝점 URL을 열어 MinIO 데이터스토어로 이동합니다.
- 7 MinIO 서버에 로그인하고 AccessKey 및 SecretKey를 제공합니다.
- 8 MinIO as a Service를 사용하도록 설정하려면 `minio.service` 스크립트를 다운로드하여 자동 시작을 위해 MinIO를 구성합니다.

```
curl -O https://raw.githubusercontent.com/minio/minio-service/master/linux-systemd/minio.service
```

- 9 `minio.service` 스크립트를 편집하고 `ExecStart`에 대해 다음 값을 추가합니다.

```
ExecStart=/usr/local/bin/minio server /DATA-MINIO path
```

- 10 수정된 스크립트를 저장합니다.
- 11 다음 명령을 실행하여 MinIO 서비스를 구성합니다.

```
cp minio.service /etc/systemd/system
cp minio /usr/local/bin/
systemctl daemon-reload
systemctl start minio
systemctl status minio
systemctl enable minio
```

- 12 MinIO 브라우저를 시작하고 개체 저장소에 로그인하여 백업 및 복원을 위한 MinIO 버킷을 생성합니다.
- 13 버킷 생성 아이콘을 클릭합니다.
- 14 버킷 이름(예: `my-cluster-backups`)을 입력합니다.
- 15 버킷이 생성되었는지 확인합니다.
- 16 기본적으로 새 MinIO 버킷은 읽기 전용입니다. Velero 독립형 백업 및 복원을 위해서는 MinIO 버킷이 읽기-쓰기여야 합니다. 버킷을 읽기-쓰기로 설정하려면 버킷을 선택하고 말줄임표(점) 링크를 클릭합니다.
- 17 **정책 편집**을 선택합니다.
- 18 정책을 **읽기 및 쓰기**로 변경합니다.
- 19 **추가**를 클릭합니다.
- 20 대화상자를 닫으려면 X를 클릭합니다.

## Velero CLI 설치

VM 클라이언트 또는 로컬 시스템에 Velero CLI를 설치합니다.

이 설명서에 사용된 버전은 *Tanzu Kubernetes Grid 2.2.0용 Velero 1.9.7*입니다.

- 1 [VMware Customer Connect 포털](#)의 Tanzu Kubernetes Grid 제품 다운로드 페이지에서 Velero를 다운로드합니다.

---

**참고** VMware의 지원을 받으려면 VMware에서 서명한 Velero 바이너리를 사용해야 합니다.

---

- 2 명령줄을 열고 디렉토리를 Velero CLI 다운로드로 변경합니다.
- 3 다운로드 파일의 압축을 풉니다. 예:

```
gunzip velero-linux-vX.X.X_vmware.1.gz
```

- 4 Velero 바이너리를 확인합니다.

```
ls -l
```

- 5 Velero CLI에 실행 권한을 부여합니다.

```
chmod +x velero-linux-vX.X.X_vmware.1
```

- 6 Velero CLI를 시스템 경로로 이동하여 전체적으로 사용할 수 있도록 합니다.

```
cp velero-linux-vX.X.X_vmware.1 /usr/local/bin/velero
```

- 7 설치를 확인합니다.

```
velero version
```

## Tanzu Kubernetes 클러스터에 Velero 및 Restic 설치

Velero CLI 컨텍스트는 kubectl 컨텍스트를 자동으로 따릅니다. Velero CLI 명령을 실행하여 대상 클러스터에 Velero 및 Restic을 설치하기 전에 kubectl 컨텍스트를 설정합니다.

- 1 MinIO 버킷의 이름을 검색합니다. 예: `my-cluster-backups`
- 2 MinIO 버킷에 대한 AccessKey 및 SecretKey를 가져옵니다.
- 3 Velero CLI가 작동할 클러스터를 알 수 있도록 대상 Kubernetes 클러스터에 대한 컨텍스트를 설정합니다.

```
kubectl config use-context tkgs-cluster-name
```

- 4 `credentials-minio`라는 암호 파일을 생성합니다. 이 파일을 수집한 MinIO 서버 액세스 자격 증명으로 업데이트합니다. 예:

```
aws_access_key_id = 0XXNO8JCCGV41QZBV0RQ
aws_secret_access_key = c1Z1bf8Ljkvkmq7fHucrKCkxV39BRbcycGeXQDfx
```

**참고** "NoCredentialProviders: 체인에 올바른 제공자가 없습니다."라는 설명과 함께 "백업 저장소를 가져 오는 동안 오류가 발생했습니다." 오류 메시지가 표시되면 자격 증명 파일의 시작 부분에 `[default]` 줄을 추가합니다. 예:

```
[default]
aws_access_key_id = 0XXNO8JCCGV41QZBV0RQ
aws_secret_access_key = c1Z1bf8Ljkvkmq7fHucrKCkxV39BRbcycGeXQDfx
```

- 5 파일을 저장하고 파일이 제자리에 있는지 확인합니다.

```
ls
```

- 6 다음 명령을 실행하여 대상 Kubernetes 클러스터에 Velero 및 Restic을 설치합니다. 두 URL을 모두 MinIO 인스턴스의 URL로 바꿉니다.

```
velero install \
--provider aws \
--plugins velero/velero-plugin-for-aws:v1.0.0 \
--bucket tkgs-velero \
--secret-file ./credentials-minio \
--use-volume-snapshots=false \
--use-restic \
--backup-location-config \
region=minio,s3ForcePathStyle="true",s3Url=http://10.199.17.63:9000,publicUrl=http://
10.199.17.63:9000
```

- 7 Velero 및 Restic의 설치를 확인합니다.

```
kubectl logs deployment/velero -n velero
```

- 8 `velero` 네임스페이스를 확인합니다.

```
kubectl get ns
```

- 9 `velero` 및 `restic` 포드를 확인합니다.

```
kubectl get all -n velero
```

## Restic DaemonSet 문제 해결(필요한 경우)

Kubernetes 클러스터에서 3-포드 Restic DaemonSet을 실행하려면 Restic DaemonSet 규격을 업데이트하고 hostPath를 수정해야 할 수 있습니다. 이 문제에 대한 자세한 내용은 Velero 설명서에서 [Restic 통합](#)을 참조하십시오.

- 1 3-포드 Restic DaemonSet을 확인합니다.

```
kubectl get pod -n velero
```

포드가 CrashLoopBackOff 상태인 경우 다음과 같이 편집합니다.

- 2 edit 명령을 실행합니다.

```
kubectl edit daemonset restic -n velero
```

- 3 hostPath를 /var/lib/kubelet/pods에서 /var/vcap/data/kubelet/pods로 변경합니다.

```
- hostPath:
  path: /var/vcap/data/kubelet/pods
```

- 4 파일을 저장합니다.

- 5 3-포드 Restic DaemonSet을 확인합니다.

```
kubectl get pod -n velero
```

NAME	READY	STATUS	RESTARTS	AGE
restic-5jln8	1/1	Running	0	73s
restic-bpvtq	1/1	Running	0	73s
restic-vg8j7	1/1	Running	0	73s
velero-72c84322d9-1e7bd	1/1	Running	0	10m

## Velero 메모리 제한 조정(필요한 경우)

Velero 백업이 여러 시간 동안 status=InProgress를 반환하는 경우 제한 및 요청 메모리 설정을 늘립니다.

- 1 다음 명령을 실행합니다.

```
kubectl edit deployment/velero -n velero
```

- 2 제한 및 요청 메모리 설정을 기본값인 256Mi 및 128Mi에서 512Mi 및 256Mi로 변경합니다.

```
ports:
- containerPort: 8085
  name: metrics
  protocol: TCP
resources:
  limits:
```



```

cpu: "1"
memory: 512Mi
requests:
  cpu: 500m
  memory: 256Mi
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File

```

## 독립형 Velero 및 Restic을 사용하여 클러스터 워크로드 백업 및 복원

독립형 Velero 및 Restic을 사용하여 TKG 클러스터에서 실행되는 워크로드를 백업 및 복원할 수 있습니다. 이 방법은 vSphere용 Velero 플러그인 사용에 대한 대안입니다. 독립형 Velero를 사용하는 주된 이유는 이식성이 필요한 경우입니다. 상태 저장 워크로드에는 Restic이 필요합니다.

### 사전 요구 사항

독립형 Velero 및 Restic을 사용하여 TKG 클러스터에서 워크로드를 백업하고 복원하려면 대상 클러스터에 독립형 Velero 및 Restic 버전을 설치해야 합니다. 별도의 대상 클러스터에서 복원을 수행하려면 Velero와 Restic도 대상 클러스터에 설치해야 합니다. [TKG 클러스터에 독립형 Velero 및 Restic 설치 및 구성의 내용을 참조하십시오.](#)

### TKG 클러스터에서 실행되는 상태 비저장 애플리케이션 백업

TKG 클러스터에서 실행되는 상태 비저장 애플리케이션을 백업하려면 Velero를 사용해야 합니다.

이 예에서는 `--include namespaces` 태그(모든 애플리케이션 구성 요소가 해당 네임스페이스에 있는 경우)를 사용하여 예제 상태 비저장 애플리케이션을 백업하고 복원하는 방법을 보여줍니다.

```
velero backup create example-backup --include-namespaces example-backup
```

다음이 표시됩니다.

```

Backup request "example-backup" submitted successfully.
Run `velero backup describe example-backup` or `velero backup logs example-backup` for more
details.

```

생성된 백업을 확인합니다.

```
velero backup get
```

```
velero backup describe example-backup
```

S3 호환 개체 저장소(예: MinIO 서버)에서 Velero 버킷을 확인합니다.

Velero는 Kubernetes CRD(사용자 지정 리소스 정의)에 일부 메타데이터를 씁니다.

```
kubectl get crd
```

Velero CRD를 사용하면 다음과 같은 특정 명령을 실행할 수 있습니다.

```
kubectl get backups.velero.io -n velero
```

```
kubectl describe backups.velero.io guestbook-backup -n velero
```

## TKG 클러스터에서 실행되는 상태 비저장 애플리케이션 복원

TKG 클러스터에서 실행되는 상태 비저장 애플리케이션을 복원하려면 Velero를 사용해야 합니다.

예제 애플리케이션의 복원을 테스트하려면 예제 애플리케이션을 삭제합니다.

네임스페이스를 삭제합니다.

```
kubectl delete ns guestbook
namespace "guestbook" deleted
```

애플리케이션을 복원합니다.

```
velero restore create --from-backup example-backup
```

다음은 표시됩니다.

```
Restore request "example-backup-20200721145620" submitted successfully.
Run `velero restore describe example-backup-20200721145620` or `velero restore logs example-backup-20200721145620` for more details.
```

애플리케이션이 복원되었는지 확인합니다.

```
velero restore describe example-backup-20200721145620
```

다음 명령을 실행하여 확인합니다.

```
velero restore get
```

```
kubectl get ns
```

```
kubectl get pod -n example
```

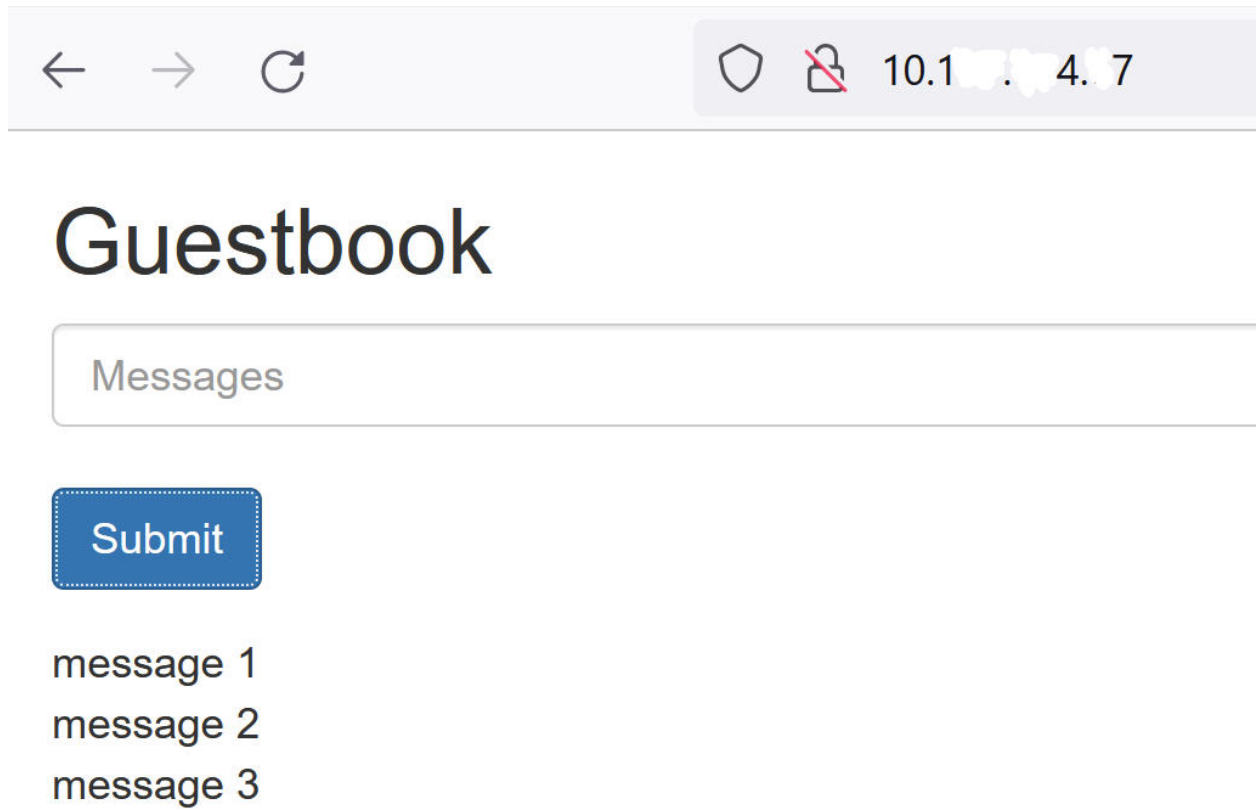
```
kubectl get svc -n example
```

## TKG 클러스터에서 실행되는 상태 저장 애플리케이션 백업

TKG 클러스터에서 실행되는 상태 저장 애플리케이션을 백업하려면 애플리케이션 메타데이터 및 영구 볼륨에 저장된 애플리케이션 데이터를 모두 백업해야 합니다. 이렇게 하려면 Velero와 Restic이 모두 필요합니다.

이 예에서는 방명록 애플리케이션을 사용합니다. TKG 클러스터에 방명록 애플리케이션을 배포했다고 가정합니다. [#unique\\_17](#)의 내용을 참조하십시오.

상태 저장 백업 및 복원을 시연할 수 있도록 프런트 엔드 웹 페이지를 사용하여 방명록 애플리케이션에 메시지를 제출하여 메시지가 지속되도록 합니다. 예:



이 예에서는 `--include namespace` 태그와 포드 주석을 사용하여 방명록 애플리케이션을 백업하고 복원하는 방법을 보여줍니다.

**참고** 이 예에서는 주석을 사용합니다. 하지만 Velero 버전 1.5 이상에서는 더 이상 주석이 필요하지 않습니다. 주석을 사용하지 않으려면 백업을 생성할 때 `--default-volumes-to-restic` 옵션을 사용하면 됩니다. 그러면 Restic을 사용하여 모든 PV가 자동으로 백업됩니다. 자세한 내용은 <https://velero.io/docs/v1.5/restic/>의 내용을 참조하십시오.

백업 절차를 시작하려면 포드의 이름을 가져옵니다.

```
kubectl get pod -n guestbook
```

예:

```
kubectl get pod -n guestbook
```

NAME	READY	STATUS	RESTARTS	AGE
guestbook-frontend-deployment-85595f5bf9-h8cff	1/1	Running	0	55m
guestbook-frontend-deployment-85595f5bf9-lw6tg	1/1	Running	0	55m

guestbook-frontend-deployment-85595f5bf9-wpqc8	1/1	Running	0	55m
redis-leader-deployment-64fb8775bf-kbs6s	1/1	Running	0	55m
redis-follower-deployment-84cd76b975-jrn8v	1/1	Running	0	55m
redis-follower-deployment-69df9b5688-zml4f	1/1	Running	0	55m

영구 볼륨은 Redis 포드에 연결됩니다. Restic을 사용하여 이러한 상태 저장 포드를 백업하기 때문에 volumeMount라는 이름으로 상태 저장 포드에 주석을 추가해야 합니다.

상태 저장 포드에 주석을 추가하려면 volumeMount를 알고 있어야 합니다. mountName을 가져오려면 다음 명령을 실행합니다.

```
kubectl describe pod redis-leader-deployment-64fb8775bf-kbs6s -n guestbook
```

결과에 redis-leader-data의 Containers.leader.Mounts: /data가 표시됩니다. 이 마지막 토큰이 리더 포드 주석에 사용할 volumeMount 이름입니다. 팔로워의 경우 redis-follower-data입니다. 소스 YAML에서 volumeMount 이름을 가져올 수도 있습니다.

각 Redis 포드에 주석을 추가합니다. 예를 들면 다음과 같습니다.

```
kubectl -n guestbook annotate pod redis-leader-64fb8775bf-kbs6s backup.velero.io/backup-volumes=redis-leader-data
```

다음은 표시됩니다.

```
pod/redis-leader-64fb8775bf-kbs6s annotated
```

주석을 확인합니다.

```
kubectl -n guestbook describe pod redis-leader-64fb8775bf-kbs6s | grep Annotations
Annotations:  backup.velero.io/backup-volumes: redis-leader-data
```

```
kubectl -n guestbook describe pod redis-follower-779b6d8f79-5dphr | grep Annotations
Annotations:  backup.velero.io/backup-volumes: redis-follower-data
```

Velero 백업을 수행합니다.

```
velero backup create guestbook-backup --include-namespaces guestbook
```

다음은 표시됩니다.

```
Backup request "guestbook-backup" submitted successfully.
Run `velero backup describe guestbook-pv-backup` or `velero backup logs guestbook-pv-backup`
for more details.
```

생성된 백업을 확인합니다.

```
velero backup get
```

NAME	STATUS	ERRORS	WARNINGS	CREATED
EXPIRES	STORAGE LOCATION	SELECTOR		
guestbook-backup	Completed	0	0	2020-07-23 16:13:46 -0700 PDT
29d	default	<none>		

백업 세부 정보를 확인합니다.

```
velero backup describe guestbook-backup --details
```

Velero를 사용하면 다음과 같은 다른 명령을 실행할 수 있습니다.

```
kubectl get backups.velero.io -n velero
```

NAME	AGE
guestbook-backup	4m58s

그리고:

```
kubectl describe backups.velero.io guestbook-backup -n velero
```

## TKG 2.0 클러스터에서 실행되는 상태 저장 애플리케이션 복원

TKG 클러스터에서 실행되는 상태 저장 애플리케이션을 복원하려면 애플리케이션 메타데이터 및 영구 볼륨에 저장된 애플리케이션 데이터를 모두 복원해야 합니다. 이렇게 하려면 Velero와 Restic이 모두 필요합니다.

이 예에서는 이전 섹션에서 설명한 대로 상태 저장 방명록 애플리케이션을 백업했다고 가정합니다.

상태 저장 애플리케이션의 복원을 테스트하려면 해당 네임스페이스를 삭제합니다.

```
kubectl delete ns guestbook
namespace "guestbook" deleted
```

애플리케이션 삭제를 확인합니다.

```
kubectl get ns
kubectl get pvc,pv --all-namespaces
```

백업에서 애플리케이션을 복원하려면 다음 명령 구문을 사용합니다.

```
velero restore create --from-backup <velero-backup-name>
```

예:

```
velero restore create --from-backup guestbook-backup
```

다음과 유사한 메시지가 표시됩니다.

```
Restore request "guestbook-backup-20200723161841" submitted successfully.
Run `velero restore describe guestbook-backup-20200723161841` or `velero restore logs
guestbook-backup-20200723161841` for more details.
```

상태 저장 방명록 애플리케이션이 복원되었는지 확인합니다.

```
velero restore describe guestbook-backup-20200723161841

Name:          guestbook-backup-20200723161841
Namespace:     velero
Labels:        <none>
Annotations:   <none>

Phase: Completed

Backup: guestbook-backup

Namespaces:
  Included: all namespaces found in the backup
  Excluded: <none>

Resources:
  Included: *
  Excluded: nodes, events, events.events.k8s.io, backups.velero.io,
restores.velero.io, resticrepositories.velero.io
  Cluster-scoped: auto

Namespace mappings: <none>

Label selector: <none>

Restore PVs: auto

Restic Restores (specify --details for more information):
  Completed: 3
```

다음 추가 명령을 실행하여 복원을 확인합니다.

```
velero restore get
```

NAME	BACKUP	STATUS	ERRORS	WARNINGS
CREATED	SELECTOR			
guestbook-backup-20200723161841	guestbook-backup	Completed	0	0
2021-08-11 16:18:41 -0700 PDT	<none>			

네임스페이스가 복원되었는지 확인합니다.

```
kubectl get ns
```

NAME	STATUS	AGE
default	Active	16d
guestbook	Active	76s
...		
velero	Active	2d2h

애플리케이션이 복원되었는지 확인합니다.

```
vkubectl get all -n guestbook
```

NAME	READY	STATUS	RESTARTS	AGE
pod/frontend-6cb7f8bd65-h2pnb	1/1	Running	0	6m27s
pod/frontend-6cb7f8bd65-kwlpr	1/1	Running	0	6m27s
pod/frontend-6cb7f8bd65-snw14	1/1	Running	0	6m27s
pod/redis-leader-64fb8775bf-kbs6s	1/1	Running	0	6m28s
pod/redis-follower-779b6d8f79-5dphr	1/1	Running	0	6m28s
pod/redis-follower-899c7e2z65-8apnk	1/1	Running	0	6m28s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
service/guestbook-frontend	LoadBalancer	10.10.89.59	10.19.15.99
service/redis-follower	ClusterIP	10.111.163.189	<none>
service/redis-leader	ClusterIP	10.111.70.189	<none>

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/guestbook-frontend-deployment	3/3	3	3	65s
deployment.apps/redis-follower-deployment	1/2	2	1	65s
deployment.apps/redis-leader-deployment	1/1	1	1	65s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/guestbook-frontend-deployment-56fc5b6b47	3	3	3	65s
replicaset.apps/redis-follower-deployment-6fc9cf5759	2	2	1	65s
replicaset.apps/redis-leader-deployment-7d89bbdbcf	1	1	1	65s

영구 볼륨이 복원되었는지 확인합니다.

```
kubectl get pvc,pv -n guestbook
```

NAME	STATUS
persistentvolumeclaim/redis-leader-claim	Bound
persistentvolumeclaim/redis-follower-claim	Bound

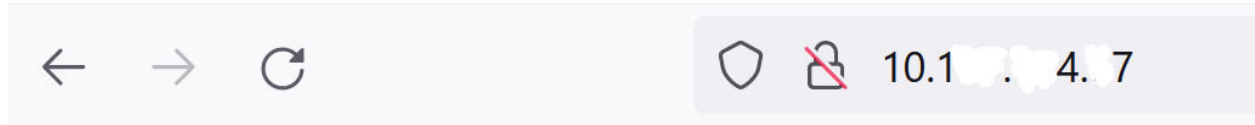
VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
persistentvolumeclaim/redis-leader-claim	2Gi	RWO	thin-disk	2m40s
persistentvolumeclaim/redis-follower-claim	2Gi	RWO	thin-disk	2m40s

NAME	CAPACITY	ACCESS MODES	RECLAIM
POLICY	STATUS	CLAIM	STORAGECLASS REASON AGE

persistentvolume/pvc-55591938-921f-452a-b418-2cc680c0560b	2Gi	RWO		
Delete	Bound	guestbook/redis-follower-claim	thin-disk	2m40s
persistentvolume/pvc-a2f6e6d4-42db-4fb8-a198-5379a2552509	2Gi	RWO		
Delete	Bound	guestbook/redis-leader-claim	thin-disk	2m40s

마지막으로 방명록 프런트 엔드 서비스의 외부 IP를 사용하여 방명록 프런트 엔드에 액세스하고 자습서 시작 부분에서 제출한 메시지가 복원되었는지 확인합니다. 예:



# Guestbook

Messages

Submit

message 1

message 2

message 3

## CSI 스냅샷과 함께 Velero를 사용하여 백업 및 복원

CSI 스냅샷과 함께 Velero를 사용하여 감독자에 프로비저닝된 TKG 클러스터에서 실행되는 워크로드에 대해 CSI 생성 영구 볼륨을 백업하고 복원할 수 있습니다.

### 요구 사항

다음 요구 사항을 준수합니다.

- vSphere 8.0 U2 이상
- vSphere 8.x 이상용 Tanzu Kubernetes 릴리스 v1.26.5



- 볼륨 스냅샷을 지원하는 CSI 드라이버를 사용하여 생성된 영구 볼륨

**주의** CSI 스냅샷과 함께 Velero를 사용하는 것은 볼륨 스냅샷을 지원하는 CSI 드라이버를 사용하여 생성된 영구 볼륨에만 가능합니다. 자세한 내용은 "vSphere IaaS 제어부에서 TKG 서비스 사용" 에서 [TKG 클러스터에서 스냅샷 생성을 참조하십시오.](#)

## 절차

CSI(Container Storage Interface) 스냅샷과 함께 Velero를 사용하여 TKGS 클러스터에서 실행되는 워크로드를 백업 및 복원할 수 있습니다. Velero node-agent는 CSI 스냅샷 데이터 이동을 사용하여 백업 및 복원의 구체적인 작업을 완료하기 위해 모듈을 호스팅하는 DaemonSet입니다. 자세한 내용은 [Velero의 Container Storage Interface 스냅샷 지원](#)을 참조하십시오.

- 1 S3 호환 스토리지 위치(예: MinIO 또는 AWS S3 버킷)를 생성합니다.

아래 예에서는 AWS S3 버킷을 사용합니다.

MinIO를 사용하려면 [MinIO 개체 저장소 설치 및 구성](#) 항목을 참조하십시오.

- 2 kubectl을 실행 중인 클러스터 클라이언트에 Velero CLI를 설치합니다.

<https://github.com/vmware-tanzu/velero/releases>에서 다운로드합니다.

다음 링크 중 하나에서 설치 지침을 참조하십시오.

- [1단계: Linux Workstation에서 Velero CLI 설치](#)
- [Velero CLI 설치](#)
- <https://velero.io/docs/v1.12/basic-install/#install-the-cli>

- 3 Velero 백업을 실행하려는 TKG 서비스 클러스터에 연결합니다.

[Kubectl을 사용하여 vCenter Single Sign-On 사용자](#)로 TKG 서비스 클러스터에 연결을 참조하십시오.

- 4 클러스터에 Velero CLI 플러그인이 있는 Velero를 설치합니다.

Velero v1.14 릴리스부터 Velero CSI 플러그인이 Velero와 병합됩니다. 따라서 Velero v1.14 이상을 설치하는 경우 Velero CSI 플러그인을 별도로 설치할 필요가 없습니다. 설치하면 Velero 포드가 시작되지 않습니다.

예를 들어, 다음 명령은 AWS S3 스토리지 백엔드와 해당 자격 증명 파일을 사용하여 Velero를 설치합니다. Velero v1.14이므로 Velero CSI 플러그인을 별도로 설치할 필요가 없습니다.

```
velero install \
  --provider aws \
  --plugins velero/velero-plugin-for-aws:v1.14 \
  --bucket velero-cpe-backup-bucket \
  --secret-file ./cloud-credential \
  --use-volume-snapshots=true \
  --features=EnableCSI --use-node-agent
```

이전 버전의 Velero를 설치하려면 Velero CSI 플러그인도 설치해야 합니다. 예:

```
velero install \  
  --provider aws \  
  --plugins velero/velero-plugin-for-aws:v1.9.0,velero/velero-plugin-for-csi:v0.7.0 \  
  --bucket velero-cpe-backup-bucket \  
  --secret-file ./cloud-credential \  
  --use-volume-snapshots=true \  
  --features=EnableCSI --use-node-agent
```

# vSphere IaaS Control Plane에서 VM 서비스 VM 백업 및 복원

# 5

vSphere 관리자는 VMware VADP(vSphere Storage API – Data Protection) 기반의 백업 파트너 솔루션을 사용하여 감독자에서 VM 서비스 VM의 백업, 전체 복원 및 등록을 자동으로 수행합니다. 기본 인프라 관련 문제로 인해 자동 등록이 실패하는 경우 문제를 해결한 다음, `registerVM` API를 수동으로 호출하여 VM을 다시 등록할 수 있습니다.

## VM 서비스 VM 백업

vSphere IaaS Control Plane에서는 VMware vSphere Storage API – Data Protection을 사용하는 Veeam과 같은 백업 파트너 솔루션을 통해 VM 서비스 VM의 자동 백업을 수행할 수 있습니다.

일반적으로 vSphere 관리자는 파트너 솔루션을 사용하여 다음 작업을 수행합니다.

- 백업 소프트웨어 설치, 백업 스토리지 및 저장소 구성 등 백업 인프라를 설정합니다.
- VM 또는 VM 그룹을 대상으로 하는 백업 작업을 생성합니다.
- 작업을 트리거하여 백업을 시작합니다.

트리거되면 파트너 소프트웨어는 VM의 구성, 데이터 및 Kubernetes 상태를 백업합니다.

일반적으로 백업에는 다음 항목이 포함됩니다.

- vCenter Server에 저장된 VM 구성.
- VMX 파일.
- VM 데이터 디스크의 콘텐츠(정적 또는 FCD일 수 있음).

VM 서비스 VM의 경우 백업에는 VM의 Kubernetes 상태와 복원 시 가상 시스템을 부트스트랩하는 데 필요한 추가 리소스도 포함됩니다.

VMware vSphere Storage API – Data Protection 및 파트너 백업 솔루션 사용 방법에 대한 자세한 내용은 [기술 자료 문서 1021175](#) 및 파트너 설명서를 참조하십시오.

## VM 서비스 VM 복원

필요한 경우 vSphere 관리자는 백업을 사용하여 VM을 복원할 수 있습니다. 예를 들어 실패한 VM을 복원할 수 있습니다.

vSphere IaaS Control Plane는 백업 파일의 전체 VM을 원래 VM의 최신 상태로 복원하는 전체 VM 복원만 지원합니다.

이러한 유형의 복원을 수행하려면 원래 VM이 감독자 및 vSphere에 존재하지 않아야 합니다. 여전히 존재하는 경우 복원 작업을 트리거하기 전에 감독자에서 `kubectl delete vm` 명령을 사용합니다.

백업 소프트웨어에서 복원 프로세스를 트리거하면 백업 소프트웨어는 복원 중에 지정된 리소스 풀 및 폴더에 VM을 다시 생성합니다. VM은 동일하거나 다른 VM 이름으로 복원될 수 있습니다.

프로세스가 성공하면 vSphere IaaS Control Plane는 VM이 원래 생성된 동일한 vSphere 네임스페이스의 감독자에서 복원된 VM을 감지하고 자동으로 등록합니다. 복원 프로세스 중에 `VirtualMachine` 리소스가 감독자에 생성됩니다. 해당하는 경우 VM을 부트스트랩하기 위한 암호 또는 가상 시스템에서 사용하는 추가 볼륨에 대한 `PersistentVolumeClaims`와 같은 추가 리소스도 생성됩니다.

암호에 대한 자세한 내용은 <https://kubernetes.io/docs/concepts/configuration/secret/>에서 Kubernetes 설명서를 참조하십시오. 영구 볼륨에 대한 자세한 내용은 <https://kubernetes.io/docs/concepts/storage/persistent-volumes/> 항목을 참조하십시오.

## 지침 및 고려 사항

VM을 복원할 때는 다음 사항을 고려하십시오.

- 복원 프로세스를 시작하기 전에 다음 단계를 수행해야 합니다.
  - 원래 VM이 감독자 및 vSphere에서 제거되었는지 확인합니다. 여전히 존재하는 경우 감독자에서 `kubectl delete vm` 명령을 사용하여 삭제합니다.
  - 백업과 복원 사이에 기본 인프라가 변경되지 않았는지 확인합니다. 원래 VM이 사용한 모든 적절한 리소스(예: VM 클래스 및 스토리지 정책)가 대상 vSphere 네임스페이스에 그대로 유지되어 있어야 합니다. 그렇지 않으면 VM의 자동 등록이 오류와 함께 실패합니다.
- 복원된 VM은 원래 생성되었던 동일한 대상 vSphere 네임스페이스에 등록됩니다.
- 대상 리소스 풀 및 폴더 이름은 원래 VM이 있던 리소스 풀 및 폴더 이름과 동일해야 합니다.
- 복원된 VM의 이름은 원래 VM 이름과 동일할 수 있습니다. 다른 VM 이름을 사용할 수도 있습니다.

다음으로 아래 항목을 읽으십시오.

- [VM 서비스 VM 수동 등록](#)

## VM 서비스 VM 수동 등록

어떤 이유로든 VM의 자동 등록이 성공하지 못하면 실패의 원인이 되는 정확한 문제에 대해 알려주는 오류 메시지가 표시됩니다. 문제를 해결한 후 `registerVM` API를 호출하고 VM의 `moID`를 사용하여 VM을 수동으로 등록할 수 있습니다.

다음 예에서는 DCLI(Datacenter CLI) 명령을 사용하여 VM을 등록합니다.

**절차**

- 1 등록할 VM의 moID를 가져옵니다.

```
# dcli com vmware vcenter vm list
```

moID는 MOREf ID라고도 하며 접두어 vm과 숫자 ID로 구성됩니다(예: vm-123456).

- 2 VM을 수동으로 등록합니다.

```
# dcli com vmware vcenter namespaces instances registervm --namespace my-namespace --vm
vm-123456 +username my-username +password my-password
```

이 명령은 다음과 유사한 작업을 반환합니다.

```
task-637:6b051692-7aff-4d59-8a3f-699d114d37e3
```

- 3 VAPI 작업 서비스를 사용하여 작업의 상태를 가져옵니다.

```
# dcli com vmware cis tasks get --task task-637:6b051692-7aff-4d59-8a3f-699d114d37e3
+username my-username +password my-password
```

이 명령은 작업 상태와 오류 메시지(실패할 경우)를 반환합니다.

# vSphere용 Velero 플러그인을 사용하여 vSphere 포드 백업 및 복원

# 6

vSphere용 Velero 플러그인을 사용하여 vSphere 포드에서 실행되는 워크로드를 백업하고 복원할 수 있습니다.

## 개요

vSphere용 Velero 플러그인을 사용하여 감독자의 vSphere 포드에서 실행되는 워크로드를 백업하고 복원할 수 있습니다. vSphere 포드에서 실행되는 상태 비저장 애플리케이션과 상태 저장 애플리케이션을 모두 백업하고 복원할 수 있습니다. 상태 저장 애플리케이션의 경우 vSphere용 Velero 플러그인을 사용하여 PV(영구 볼륨)의 스냅샷을 생성합니다.

**참고** 독립형 Velero를 Restic과 함께 사용하여 vSphere 포드를 백업 및 복원할 수 없습니다. 감독자에 설치된 vSphere용 Velero 플러그인을 사용해야 합니다.

## 사전 요구 사항

vSphere 포드를 백업 및 복원하려면 vSphere용 Velero 플러그인을 설치하고 구성해야 합니다. [#unique\\_20](#)의 내용을 참조하십시오.

**참고** vSphere용 Velero 플러그인은 감독자의 상태를 백업 및 복원하지 않습니다.

## vSphere 포드 백업

상태 비저장 vSphere 포드를 백업하려면 다음 명령을 실행합니다.

```
velero backup create <backup name> --include-namespaces=my-namespace
```

모든 로컬 스냅샷이 생성되고 Kubernetes 메타데이터가 개체 저장소에 업로드되면 백업이 `Completed`로 표시됩니다. 하지만 볼륨 스냅샷의 백업은 비동기식으로 발생하며 백그라운드에서 계속 수행 중일 수 있고 완료하는 데 다소 시간이 걸릴 수 있습니다.

볼륨 스냅샷의 상태는 스냅샷 및 업로드 사용자 지정 리소스를 모니터링하여 확인할 수 있습니다.

### 스냅샷 CRD

각 볼륨 스냅샷에 대해 스냅샷이 생성된 PVC와 동일한 네임스페이스에 스냅샷 사용자 지정 리소스가 생성됩니다. 다음 명령을 실행하여 PVC 네임스페이스의 모든 스냅샷을 가져올 수 있습니다.

```
kubectl get -n <pvc namespace> snapshot
```

스냅샷 CRD에는 `status.phase` 필드에 대해 다음과 같은 몇 가지 단계가 있습니다.

상태	설명
신규	아직 처리되지 않았습니다.
Snapshotted	로컬 스냅샷이 생성되었습니다.
SnapshotFailed	로컬 스냅샷을 생성하지 못했습니다.
Uploading	스냅샷이 업로드되고 있습니다.
Uploaded	스냅샷이 업로드되었습니다.
UploadFailed	스냅샷을 업로드하지 못했습니다.
Canceling	스냅샷 업로드가 취소되고 있습니다.
Canceled	스냅샷 업로드가 취소되었습니다.
CleanupAfterUploadFailed	스냅샷 업로드 후 로컬 스냅샷을 정리하지 못했습니다.

## 업로드 CRD

개체 저장소에 업로드할 각 볼륨 스냅샷에 대해 Velero와 동일한 네임스페이스에 업로드 CR이 생성됩니다. 다음 명령을 실행하여 Velero 네임스페이스의 모든 업로드를 가져올 수 있습니다.

```
kubectl get -n <velero namespace> upload
```

업로드 CRD에는 `status.phase` 필드에 대해 다음과 같은 몇 가지 단계가 있습니다.

상태	설명
신규	아직 처리되지 않았습니다.
InProgress	업로드 진행 중
UploadError	업로드하지 못했습니다.
CleanupFailed	업로드 후 로컬 스냅샷을 삭제하지 못했습니다. 재시도됩니다.
Canceling	업로드가 취소되고 있습니다. 스냅샷 업로드가 진행되는 동안 <code>velero backup delete</code> 가 호출되는 경우 발생할 수 있습니다.
Canceled	업로드가 취소되었습니다.

업로드 오류 업로드는 주기적으로 재시도됩니다. 이때는 단계가 진행 중 단계로 돌아갑니다. 업로드가 성공적으로 완료된 후 해당 기록은 일정 기간 동안 남아 있다가 결국 제거됩니다.

## vSphere 포드 복원

vSphere용 Velero 플러그인을 사용하여 백업된 vSphere 포드 워크로드를 복원하려면 다음 단계를 완료합니다.

- 1 복원할 워크로드에 대한 vSphere 네임스페이스를 생성합니다.
- 2 네임스페이스에 대한 스토리지 정책을 구성합니다.
- 3 다음 Velero 명령을 실행하여 워크로드를 복원합니다.

```
velero restore create --from-backup backup-name
```

볼륨 스냅샷 및 기타 Kubernetes 메타데이터가 현재 클러스터에 성공적으로 복원되면 Velero 복원이 Completed로 표시됩니다. 이때 이 복원과 관련된 vSphere 플러그인의 모든 작업도 완료됩니다. Velero 백업의 경우처럼 백그라운드에서 비동기식 데이터 이동 작업이 없습니다.

Velero 복원이 Completed 상태가 되기 전에 CloneFromSnapshot/Download CR을 아래와 같이 모니터링하여 볼륨 복원의 상태를 확인할 수 있습니다.

### CloneFromSnapshot CRD

각 볼륨 스냅샷에서 복원하기 위해 원래 스냅샷이 생성된 PVC와 동일한 네임스페이스에 CloneFromSnapshot CR이 생성됩니다. 다음 명령을 실행하여 PVC 네임스페이스의 모든 CloneFromSnapshot을 가져올 수 있습니다.

```
kubectl -n <pvc namespace> get clonefromsnapshot
```

CloneFromSnapshot CRD에는 `status.phase` 필드에 대해 다음과 같은 몇 가지 단계가 있습니다.

상태	설명
신규	스냅샷에서 복제가 완료되지 않았습니다.
Completed	스냅샷에서 복제가 완료되었습니다.
실패	스냅샷에서 복제에 실패했습니다.

### 다운로드 CRD

개체 저장소에서 다운로드할 볼륨 스냅샷을 복원할 때마다 Velero와 동일한 네임스페이스에 다운로드 CR이 생성됩니다. 다음 명령을 실행하여 Velero 네임스페이스에서 모든 다운로드를 가져올 수 있습니다.

```
kubectl -n <velero namespace> get download
```

다운로드 CRD에는 `status.phase` 필드에 대해 다음과 같은 몇 가지 단계가 있습니다.

상태	설명
신규	아직 처리되지 않았습니다.
InProgress	다운로드를 진행 중입니다.
Completed	다운로드가 완료되었습니다.



상태	설명
Retry	다운로드가 재시도됩니다. 백업 데이터를 다운로드하는 동안 오류가 발생하면 다운로드가 재시도됩니다.
실패	다운로드에 실패했습니다.

# vSphere IaaS Control Plane 백업 및 복원 문제 해결

# 7

vSphere IaaS Control Plane 백업 및 복원과 관련된 문제를 해결하는 방법을 알아봅니다.

다음으로 아래 항목을 읽으십시오.

- 백업에서 감독자 복원 후 분리된 개체 정리

## 백업에서 감독자 복원 후 분리된 개체 정리

백업에서 감독자를 복원하면 백업 후에 생성된 모든 K8s 리소스는 복원이 완료된 후 삭제됩니다. 이러한 리소스 중 일부가 VM 또는 디스크와 같은 개체와 연결된 경우 vCenter Server에서 분리됩니다. vCenter Server에서 분리된 개체는 정리해야 합니다.

## 절차

1 vSphere 네임스페이스에 대한 모든 VM을 나열합니다.

- a 감독자 제어부 VM에서 다음 명령을 실행하여 vSphere 네임스페이스에 대한 folderMoId를 검색합니다.

```
root@421c9fa40208448fecc15d277bdca66d [ ~ ]# kubectl get availabilityzone -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "topology.tanzu.vmware.com/v1alpha1",
      "kind": "AvailabilityZone",
      "metadata": {
        ...
      },
      "spec": {
        "clusterComputeResourceMoIDs": [
          "domain-c50"
        ],
        "clusterComputeResourceMoId": "domain-c50",
        "namespaces": {
          "pod-ns": {
            "folderMoId": "group-v81", <--- this is the folderMoId that
you need for next step
            ...
          },
          "vmsvc-ns": {
            "folderMoId": "group-v83", <--- this is the folderMoId that
you need for next step
            ...
          }
        }
      }
    }
  ],
  "kind": "List",
  "metadata": {
    "resourceVersion": ""
  }
}
```

- b 다음 DCLI 명령을 실행하여 vSphere 네임스페이스에 있는 기존 VM을 모두 나열합니다.

예제에서는 pod-ns 네임스페이스를 사용합니다.

```
root@sc2-10-186-199-30 [ ~ ]# dcli +i +username 'Administrator@vsphere.local'
+password <password>
Welcome to VMware Datacenter CLI (DCLI)

usage: <namespaces> <command>

To auto-complete and browse DCLI namespaces: [TAB]
```

```

If you need more help for a command:          vcenter vm get --help
If you need more help for a namespace:       vcenter vm --help
To execute dcli internal command: env
For detailed information on DCLI usage visit: http://vmware.com/go/dcli

```

```

dcli> com vmware vcenter vm list --folders group-v81
|-----|-----|-----|-----|-----|-----|
|memory_size_MiB|vm   |name                                     |power_state|cpu_count| |
|---|---|---|---|---|---|
|512            |vm-84|deployment-before-backup-778449d88d-c9gnc|POWERED_ON |1        |
|512            |vm-85|deployment-before-backup-778449d88d-4jtgj |POWERED_ON |1        |
|512            |vm-86|deployment-before-backup-778449d88d-tqwbh |POWERED_ON |1        |
|512            |vm-91|deployment-after-backup-778449d88d-khkxx  |POWERED_OFF|1        |
|512            |vm-92|deployment-after-backup-778449d88d-7dgcc  |POWERED_OFF|1        |
|512            |vm-93|deployment-after-backup-778449d88d-sxbcf  |POWERED_OFF|1        |
|-----|-----|-----|-----|-----|-----|

```

## 2 분리된 네임스페이스를 찾아 정리합니다.

감독자 백업 후 특정 네임스페이스를 삭제한 경우 감독자를 복원하면 해당 네임스페이스가 Kubernetes 리소스로 다시 생성됩니다. 해당 K8s 리소스는 삭제해야 합니다.

- a vCenter Server의 모든 네임스페이스를 나열하여 분리된 네임스페이스를 찾습니다.

```
dcli> com vmware vcenter namespaces instances list
```

- b 모든 네임스페이스 K8s 리소스를 나열합니다.

```
root@423f9d75bef000dc828a535c6ac0bd4b [ ~ ]# k get ns -A
```

- a A 단계와 B 단계의 결과 개체 간 차이를 찾아 분리된 네임스페이스 K8s 리소스를 정리합니다.

```
root@423f9d75bef000dc828a535c6ac0bd4b [ ~ ]# k delete ns test-set-workload-ns
namespace "test-set-workload-ns" deleted
```

### 3 VirtualMachine 리소스와 연결된 분리된 VM을 찾아 정리합니다.

감독자 백업 후 생성된 VirtualMachine Kubernetes 리소스는 해당 백업에서 감독자가 복원되면 분리된 VM이 됩니다. vCenter Server 인벤토리에서 이러한 분리된 VM을 정리해야 합니다.

#### a VirtualMachine 리소스와 연결된 분리된 VM을 찾습니다.

다음 단계에서는 `vmsvc-ns` 네임스페이스를 예로 사용합니다.

- 1 vCenter Server 인벤토리의 모든 VM을 나열합니다. 예제 코드 조각은 `group-96`이 `vmsvc-ns` 네임스페이스와 연결되어 있으므로 vCenter Server의 모든 VM을 나열합니다.

```
dcli> com vmware vcenter vm list --folders group-v96
|-----|-----|-----|-----|-----|
|memory_size_MiB|vm   |name           |power_state|cpu_count|
|-----|-----|-----|-----|-----|
|2048           |vm-104|vmsvc-after    |POWERED_ON |2        |
|2048           |vm-97 |vmsvc-before   |POWERED_ON |2        |
|-----|-----|-----|-----|-----|
```

- 2 모든 VirtualMachine K8s 리소스를 나열합니다. `kubectl get`을 실행하여 제어부 VM 중 하나에 대한 리소스 세부 정보를 가져오고 출력에서 `uniqueID`를 검색합니다. 이 예에서는 K8s 리소스와 연결된 VM 목록이 `vm-97`입니다.

```
root@42344b596f57bfcf9441179faled1a5c [ ~ ]# k get vm -n vmsvc-ns -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "vmoperator.vmware.com/v1alpha1",
      "kind": "VirtualMachine",
      ...
      "uniqueID": "vm-97",
      ...
    }
  ]
}
```

- 3 위 단계의 결과로 나온 두 목록을 비교합니다.

- vCenter Server의 VM 목록: <vm-104, vm-97>
- k8s 리소스와 연결된 VM 목록: <vm-97>

따라서 분리된 VM 목록은 <vm-104>입니다.

#### b 분리된 VM을 정리합니다.

```
dcli> com vmware vcenter vm power stop --vm vm-104
dcli> com vmware vcenter vm delete --vm vm-104
```

4 포드 리소스와 연결된 분리된 VM을 찾아 정리합니다.

감독자 백업 후 생성된 포드 K8s 리소스는 감독자 복원 후 vCenter Server에서 분리된 VM이 됩니다. 다음 단계에 따라 해당 항목을 찾아서 정리합니다.

이 예에서는 네임스페이스 `pod-ns`를 사용합니다.

a vCenter Server 인벤토리의 모든 VM을 나열합니다.

이 예에서 VM 그룹 `group-v83`은 네임스페이스 `pod-ns`와 연결되어 있습니다. VM 목록은 `vm-88`, `vm-89`, `vm-90`, `vm-101`, `vm-102` 및 `vm-103`입니다.

```
dcli> com vmware vcenter vm list --folders group-v83
|-----|-----|-----|-----|-----|
|
|memory_size_MiB|vm      |name                                     |power_state|
|cpu_count|
|-----|-----|-----|-----|-----|
|
|512            |vm-101|deployment-after-backup-778449d88d-ldvn8 |POWERED_OFF|1
|
|512            |vm-102|deployment-after-backup-778449d88d-v29dd |POWERED_OFF|1
|
|512            |vm-103|deployment-after-backup-778449d88d-zdb19 |POWERED_OFF|1
|
|512            |vm-88 |deployment-before-backup-778449d88d-fgq5b|POWERED_ON  |1
|
|512            |vm-89 |deployment-before-backup-778449d88d-mp7td|POWERED_ON  |1
|
|512            |vm-90 |deployment-before-backup-778449d88d-cjhq6|POWERED_ON  |1
|
|-----|-----|-----|-----|-----|
|
```

b K8s 리소스를 나열합니다.

`kubectl get`을 실행하여 제어부 VM 중 하나에 대한 리소스 세부 정보를 가져오고 출력에서 `vmware-system-vm-moid`를 검색합니다. K8s 리소스와 연결된 VM 목록은 `vm-88`, `vm-89`, `vm-90`입니다.

```
root@42344b596f57bfcf9441179faled1a5c [ ~ ]# k get pod -n pod-ns -o json
{
  "apiVersion": "v1",
  "items": [
    {
      "apiVersion": "v1",
      "kind": "Pod",
      "metadata": {
        "annotations": {
          ...
          "vmware-system-vm-moid": "vm-90:5a5198fc-c5cb-4b89-
a70f-331025b40539",
          ...
        },
        ...
        "vmware-system-vm-moid": "vm-88:5a5198fc-c5cb-4b89-
a70f-331025b40539",
        ...
      }
    }
  ]
}
```

```
        "vmware-system-vm-moid": "vm-89:5a5198fc-c5cb-4b89-  
a70f-331025b40539",  
        ...  
    }
```

c 위 단계의 결과로 나온 두 목록을 비교합니다.

- vCenter Server의 VM 목록은 <vm-88, vm-89, vm-90, vm-101, vm-102, vm-103>입니다.

- K8s 리소스와 연결된 VM 목록은 <vm-88, vm-89, vm-90>입니다.

따라서 분리된 VM 목록은 <vm-101, vm-102, vm-103>입니다.

d 분리된 VM을 정리합니다.

```
dcli> com vmware vcenter vm delete --vm vm-101  
dcli> com vmware vcenter vm delete --vm vm-102  
dcli> com vmware vcenter vm delete --vm vm-103
```



## 5 Tanzu Kubernetes Grid 클러스터와 연결된 분리된 VM 및 리소스 풀을 찾아 정리합니다.

감독자 백업 후 생성된 Tanzu Kubernetes Grid 클러스터 역시 감독자 복원 후 vCenter Server에서 분리된 VM이 됩니다.

**참고** 복원 작업 후 TKG 클러스터를 생성하는 동안 문제가 발생하면 현재 단계의 지침에 따라 분리된 VM을 정리해야 합니다.

### a 분리된 Tanzu Kubernetes Grid 클러스터 목록을 찾습니다.

kubectl을 사용하여 제어부 VM(<test-cluster, test-cluster-e2e-script, tkc-before-backup>) 중 하나에서 Tanzu Kubernetes Grid 클러스터 K8s 리소스 목록을 가져옵니다.

```
root@4239f4159c7063d5608cf3fc0bdd532e [ ~ ]# k get tkc -A
NAMESPACE          NAME          CONTROL PLANE  WORKER  TKR
NAME               AGE  READY  TKR COMPATIBLE  UPDATES  AVAILABLE
selfservice-tkc-ns test-cluster          1          1      v1.23.8---
vmware.3-tkg.1    19h  True   True
test-gc-e2e-demo-ns test-cluster-e2e-script 3          1      v1.23.8---
vmware.3-tkg.1    18h  False  True
tkc-ns            tkc-before-backup    3          1      v1.23.8---
vmware.3-tkg.1    16h  True   True
```

DCLI를 사용하여 네임스페이스 또는 Tanzu Kubernetes Grid 클러스터와 연결된 모든 리소스 풀을 가져온 다음 vCenter Server에서 Tanzu Kubernetes Grid 클러스터 목록을 가져옵니다. <test-cluster, test-cluster-e2e-script, tkc-before-backup, tkc-after-backup>

```
dcli> com vmware vcenter resourcepool list
|-----|-----|
|name          |resource_pool|
|-----|-----|
|Resources     |resgroup-10  |
|Resources     |resgroup-23  |
|Namespaces    |resgroup-56  |
|selfservice-tkc-ns |resgroup-62  | <--- this is a namespace
|test-cluster   |resgroup-66  | <--- Tanzu Kubernetes Grid cluster
|test-gc-e2e-demo-ns |resgroup-70  | <--- this is a namespace
|test-cluster-e2e-script|resgroup-74  | <--- Tanzu Kubernetes Grid cluster
|tkc-ns        |resgroup-80  | <--- this is a namespace
|tkc-before-backup |resgroup-89  | <--- Tanzu Kubernetes Grid cluster
|tkc-after-backup |resgroup-96  | <--- Tanzu Kubernetes Grid cluster
|-----|-----|
```

### b 위 단계의 두 목록을 비교하면 분리된 Tanzu Kubernetes Grid 클러스터 목록은 <tkc-after-backup>입니다.

- c 분리된 Tanzu Kubernetes Grid 클러스터와 연결된 VM을 정리합니다.

DCLI를 사용하여 연결된 리소스 풀 <resgroup-96>을 사용하여 분리된 Tanzu Kubernetes Grid 클러스터와 연결된 모든 VM을 가져옵니다.

```
dcli> com vmware vcenter vm list --resource-pools resgroup-96
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
|memory_size_MiB|vm      |name                                     |power_state|
|cpu_count|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
|2048          |vm-100|tkc-after-backup-zlcdm-wk5xf          |POWERED_ON |
|2            |
|2048          |vm-101|tkc-after-backup-zlcdm-76q4h          |POWERED_ON |
|2            |
|2048          |vm-98 |tkc-after-backup-zlcdm-9fv2w          |POWERED_ON |
|2            |
|2048          |vm-99 |tkc-after-backup-workers-4hdqb-657fb58d45-d7pqq|POWERED_ON |
|2            |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|
```

그런 다음 VM을 하나씩 삭제합니다.

```
dcli> com vmware vcenter vm power stop --vm vm-100
dcli> com vmware vcenter vm delete --vm vm-100
```

- d 분리된 Tanzu Kubernetes Grid 클러스터와 연결된 리소스 풀을 정리합니다.

```
<dcli> com vmware vcenter resourcepool delete --resource-pool resgroup-96
```

vSphere Client에서 분리된 리소스 풀을 삭제할 수도 있습니다.

## 6 PV(영구 볼륨)와 연결된 분리된 FCD(First Class Disk)를 찾아 정리합니다.

감독자 백업 후 생성된 PV K8s 리소스는 감독자 복원 후 vCenter Server에서 분리된 FCD가 됩니다. 다음 단계에 따라 해당 항목을 찾아서 정리합니다.

### a PV와 연결된 분리된 FCD를 찾습니다.

- 1 분리된 FCD를 찾는 데 사용할 govc를 설치합니다. govc는 UI에 대한 사용자 친화적인 CLI 대안이며 자동화 작업에 매우 적합합니다.

```
curl -L -o - "https://github.com/vmware/govmomi/releases/latest/download/govc_$(uname -s)_$(uname -m).tar.gz" | tar -C /usr/local/bin -xvzf - govc
```

<https://github.com/vmware/govmomi/tree/main/govc#installation>에서 더 많은 설치 옵션을 확인할 수 있습니다.

- 2 아래 bash 스크립트를 실행하여 감독자에 있는 PV를 나열합니다.

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk

# datastore path example - /test-vpx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmfs-0
govc volume.ls -l -ds=<datastore path>
```

결과는 다음과 같습니다.

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh
590c8e31-f5bf-4179-9250-5cdd66bf591c
pvc-843c932b-8974-475d-8f8a-9b165137169d    1.0GB    KUBERNETES
vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-b80eb76becdf
37f8ad5b-dfe6-465b-b0f0-11591a2968dc    pvc-77c42590-
f0b0-457f-9743-6a3ebca55078    1.0GB    KUBERNETES
vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-b80eb76becdf
28a265b8-2e6b-421c-b16d-046ffc7aeea7    pvc-1b88c923-4354-4537-a7cb-
a8a6d763d5e7    1.0GB    KUBERNETES    vSphereSupervisorID-7f88d7b3-12ac-4fcf-a101-
b80eb76becdf
```

- 3 아래 bash 스크립트를 실행하여 vCenter Server의 모든 디스크를 나열합니다.

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk
```

```
# datastore path example - /test-vmx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmfs-0
govc disk.ls -l -ds=<datastore path>
```

결과는 다음과 같습니다.

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh
28a265b8-2e6b-421c-b16d-046ffc7aeea7 pvc-1b88c923-4354-4537-a7cb-a8a6d763d5e7
1.0G Jul 4 02:33:27 <--- this is the disk correspondings to PV
37f8ad5b-dfe6-465b-b0f0-11591a2968dc pvc-77c42590-f0b0-457f-9743-6a3ebca55078
1.0G Jul 4 02:32:41 <--- this is the disk correspondings to PV
3a7517c2-f8c2-46a9-b0d5-18c665759311 vmware-sv-img-cache-domain-c50
26.0M Jul 4 02:36:41
590c8e31-f5bf-4179-9250-5cdd66bf591c pvc-843c932b-8974-475d-8f8a-9b165137169d
1.0G Jul 4 02:30:45 <--- this is the disk correspondings to PV
68ba220c-0f83-49eb-b77a-d60471e24844 pvc-92f83ae0-7c2d-46d9-ab85-19858462ddd1
5.0G Jul 4 18:27:02 <--- this is the disk correspondings to PV
72dbe8c5-a3b5-4298-8203-ealcb86116e6 vmware-sv-img-cache-domain-c50
3.0M Jul 4 02:38:39
79e233a6-0134-40e7-8ba8-3133442324f9 vmware-sv-img-cache-domain-c50
195.0M Jul 4 18:26:12
ala0a9d7-0baf-4592-9041-8c0feb960246 vmware-sv-img-cache-domain-c50
7.0M Jul 4 02:35:37
cec2af09-80af-4086-a069-34140e2480dc vmware-sv-img-cache-domain-c50
193.0M Jul 4 02:31:12
```

#### 4 위 단계의 두 목록을 비교합니다.

- PV 목록: <590c8e31-f5bf-4179-9250-5cdd66bf591c, 37f8ad5b-dfe6-465b-b0f0-11591a2968dc, 28a265b8-2e6b-421c-b16d-046ffc7aeea7>
- FCD 목록: <590c8e31-f5bf-4179-9250-5cdd66bf591c, 37f8ad5b-dfe6-465b-b0f0-11591a2968dc, 28a265b8-2e6b-421c-b16d-046ffc7aeea7, 68ba220c-0f83-49eb-b77a-d60471e24844>

분리된 FCD는 <68ba220c-0f83-49eb-b77a-d60471e24844>입니다.

#### b 분리된 FCD를 삭제합니다.

govc를 사용하여 분리된 FCD를 삭제합니다. 예제 스크립트:

```
#!/bin/bash

export GOVC_INSECURE=1
export GOVC_USERNAME='Administrator@vsphere.local'
export GOVC_PASSWORD=<password>
export GOVC_URL=https://<vc ip>/sdk

# datastore path example - /test-vmx-1688432886-30489-wcp.wcp-sanity/datastore/
sharedVmfs-0
govc disk.rm -ds=<datastore path> 68ba220c-0f83-49eb-b77a-d60471e24844
```

결과는 다음과 같습니다.

```
peiyangs@peiyangs-a01 govc % sudo bash orphanedPV.sh  
[06-07-23 11:36:27] Deleting 68ba220c-0f83-49eb-b77a-d60471e24844...OK
```