



사용 사례 검증 NetApp Solutions

NetApp
April 20, 2024

목차

사용 사례 검증: NetApp Astra와 DevOps	1
NetApp Astra Control을 통해 CI/CD 파이프라인에 보호 기능을 통합합니다.....	1
Astra Control을 사용하여 사후 분석을 용이하게 하고 애플리케이션을 복원합니다	7
NetApp FlexClone 기술을 사용하여 소프트웨어 개발 가속화.....	12

사용 사례 검증: NetApp Astra와 DevOps

NetApp Astra를 사용하여 DevOps에서 검증된 사용 사례는 다음과 같습니다.

- "NetApp Astra Control을 통해 CI/CD 파이프라인에 보호 기능을 통합합니다"
- "Astra Control을 활용하여 사후 분석 및 애플리케이션 복원을 용이하게 합니다"
- "NetApp FlexClone으로 소프트웨어 개발 가속화"

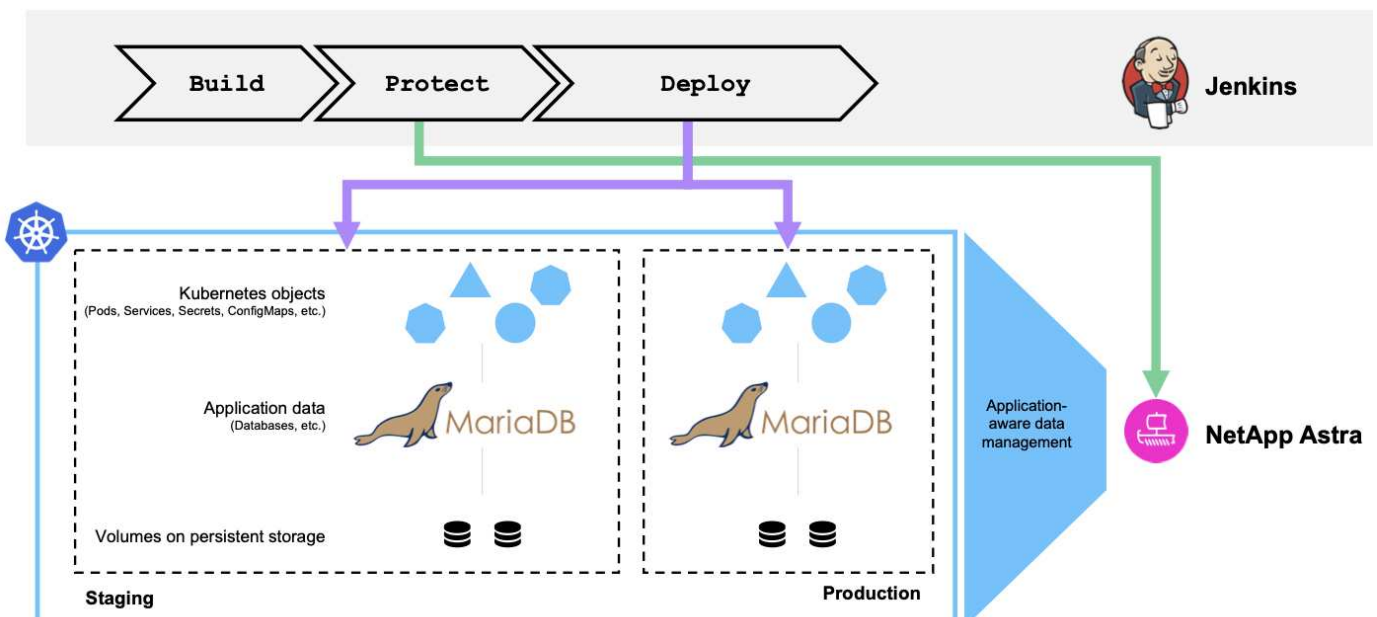
NetApp Astra Control을 통해 CI/CD 파이프라인에 보호 기능을 통합합니다

개요

DevOps 워크플로우의 가장 일반적인 용도 중 하나는 지속적인 통합 및 CI/CD(Continuous Deployment) 파이프라인으로서, 개발자가 새로운 코드를 커밋할 때 애플리케이션에서 자동화된 테스트 제품군을 구축, 통합 및 실행합니다. DevOps 엔지니어 및 SRE(Site-Reliability Engineer)에는 일반적으로 개발 프로세스에서 새로운 기능 개발, 회귀 테스트, 버그 수정, 품질 엔지니어링 및 기타 기능을 위해 다양한 워크플로우 전용 파이프라인이 있습니다.

팀의 자동화 수준이 높아짐에 따라 운영 중인 애플리케이션의 변화 속도가 압도적으로 느껴질 수 있습니다. 따라서 일부 팀은 운영 중인 애플리케이션 또는 서비스를 보호하고자 합니다. 코드 및 컨테이너 이미지를 보호할 뿐 아니라 애플리케이션 상태, 구성 데이터(예: Kubernetes 객체 및 애플리케이션과 관련된 리소스), 애플리케이션의 영구 데이터도 보호하려고 합니다.

이 사용 사례에서는 새로운 버전의 애플리케이션을 배포하는 프로모션-운영 파이프라인을 먼저 스테이징 환경에 구축한 다음 운영 환경에 구축하는 방법을 자세히 살펴보겠습니다. 이 예는 주요 퍼블릭 클라우드 및 온프레미스 환경에도 동일하게 적용됩니다. 앱 버전 중 하나의 배포를 보여주지만 파란색/녹색 또는 카나리아 배포와 같은 다른 전략에서도 파이프라인을 사용할 수 있습니다. CI/CD 파이프라인의 일부로 전체 애플리케이션 백업을 생성하여 애플리케이션을 보호합니다. 운영 중인 애플리케이션과 데이터, 상태, 구성을 애플리케이션 인식형 백업으로 여러 DevOps 워크플로우에 유용하게 사용할 수 있습니다.



이 사용 사례를 검증하는 데 사용된 애플리케이션은 입니다 **"마멘토"** 웹 기반 프런트 엔드를 사용하는 전자 상거래 솔루션, 검색 및 분석 기능을 위한 Elasticsearch 인스턴스, 모든 쇼핑 인벤토리 및 트랜잭션 세부 정보를 추적하는 MariaDB 데이터베이스 이 컨테이너형 애플리케이션은 Red Hat OpenShift 클러스터에 설치되었습니다. 애플리케이션의 모든 Pod에서 영구 볼륨을 사용하여 데이터를 저장했습니다. 영구 볼륨은 NetApp 스토리지 시스템에서 스토리지를 프로비저닝할 수 있는 Kubernetes용 컨테이너 스토리지 인터페이스 호환 스토리지 오케스트레이터인 NetApp Astra Trident에 의해 자동으로 생성되었습니다. 또한 Astra Control Center의 애플리케이션 보호 기능을 활용하기 위해 해당 애플리케이션을 Astra Control에서 관리했으며, 이 애플리케이션을 사용하여 애플리케이션의 상태를 영구 볼륨에 저장된 데이터와 함께 저장하는 애플리케이션 백업을 트리거했습니다. 을 사용했습니다 **"NetApp Astra Control Python SDK"** 애플리케이션 백업을 트리거하는 프로세스를 자동화하기 위해 CI/CD 파이프라인에 도입되었습니다. 이 파이프라인은 이라는 인기 있는 CI/CD 툴을 사용하여 생성 및 실행되었습니다 **"젠킨스"** 응용 프로그램을 빌드, 보호 및 배포하기 위한 흐름을 자동화합니다.

CI/CD 파이프라인에 보호를 도입하기 위한 사전 요구 사항과 절차를 살펴보겠습니다.

사용 사례 검증을 위한 사전 요구사항

다음 도구 또는 플랫폼이 전제 조건으로 구축 및 구성되었습니다.

1. Red Hat OpenShift Container Platform
2. NetApp ONTAP 시스템에 백엔드가 구성된 OpenShift에 설치된 NetApp Astra Trident
3. NetApp ONTAP 백엔드를 가리키는 기본 스토리지 클래스 구성
4. OpenShift 클러스터에 설치된 NetApp Astra Control Center
5. OpenShift 클러스터가 Astra Control Center에 관리 클러스터로 추가되었습니다
6. Jenkins는 OpenShift 클러스터에 설치되며 Docker 엔진이 설치된 에이전트 노드로 구성됩니다

응용 프로그램을 설치하는 중입니다

스테이징 및 운영 환경에 애플리케이션을 처음 설치하는 것부터 시작하겠습니다. 이 활용 사례의 경우 이 단계가 선행 조건이므로 수동으로 수행됩니다. CI/CD 파이프라인은 이후 애플리케이션의 새 버전 릴리스에 따라 워크플로우를 구축하고 배포하는 데 사용됩니다.

이 활용 사례의 생산 환경은 '마젠토-프로드'라는 네임스페이스이며, 해당 스테이징 환경은 Red Hat OpenShift 클러스터에 구성된 '마젠토-스테이징'이라는 네임스페이스입니다. 응용 프로그램을 설치하려면 다음 단계를 수행하십시오.

1. 프로덕션 환경에 bitnami Helm 차트를 사용하여 Magento 응용 프로그램을 설치합니다. Magento 및 MariaDB Pod용 rwx PVS를 사용합니다.

```
[netapp-user@rhel7 na_astra_control_suite]$ helm install --version 14
magento bitnami/magento -n magento-prod --create-namespace --set
image.tag=2.4.1-debian-10-
r11,magentoHost=10.63.172.243,persistence.magento.accessMode=ReadWriteMa
ny,persistence.apache.accessMode=ReadWriteMany,mariadb.master.persistenc
e.accessModes[0]=ReadWriteMany
```



Magento bitnami Helm 차트에는 Magento GUI 서비스를 제공하기 위해 로드 밸런서 서비스가 필요합니다. 우리는 중고 **"메탈리스"** 이 예에서는 온프레미스 로드 밸런서 서비스를 제공합니다.

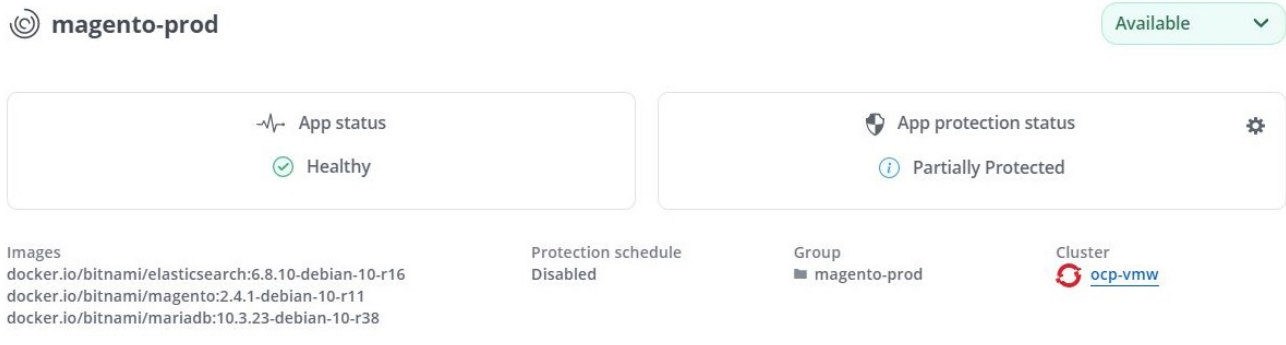
2. 몇 분 후 모든 Pod 및 서비스가 실행되고 있는지 확인합니다.

```
[netapp-user@rhel7 na_astra_control_suite]$ oc get pods -n magento-prod
NAME                                                    READY   STATUS
RESTARTS      AGE
magento-9d658fd96-qrxmt                                1/1     Running
0              49m
magento-elasticsearch-coordinating-only-69869cc5-768rm 1/1     Running
0              49m
magento-elasticsearch-data-0                            1/1     Running
0              49m
magento-elasticsearch-master-0                         1/1     Running
0              49m
magento-mariadb-0                                       1/1     Running
0              49m
```

3. 스테이징 환경에 대해서도 동일한 절차를 반복합니다.

Astra Control Center에서 Magento 응용 프로그램을 관리합니다

1. 응용 프로그램 으로 이동하고 검색된 응용 프로그램 탭을 선택합니다.
2. 제작 환경('magento-prod')에서 Magento 응용 프로그램에 대한 줄임표를 클릭하고 관리를 클릭합니다.
3. 이제 Magento 응용 프로그램은 Astra Control Center에서 관리합니다. Astra Control에서 지원하는 모든 작업은 애플리케이션에서 수행할 수 있습니다. 응용 프로그램 버전도 확인합니다.



© magento-prod Available

App status: Healthy

App protection status: Partially Protected

Images: docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16, docker.io/bitnami/magento:2.4.1-debian-10-r11, docker.io/bitnami/mariadb:10.3.23-debian-10-r38

Protection schedule: Disabled

Group: magento-prod

Cluster: ocp-vmw

4. 스테이징 환경에서 Magento 응용 프로그램을 관리하는 단계를 반복합니다('magento-staging').

통합 보호를 갖춘 CI/CD 파이프라인

새로운 버전의 애플리케이션을 사용할 때는 CI/CD 파이프라인을 사용하여 컨테이너 이미지를 구축하고, 스테이징 및 운영 환경의 백업을 수행하고, 새로운 버전의 애플리케이션을 스테이징 환경에 구축하고, 프로덕션으로 승격할 승인을 기다립니다. 그런 다음 운영 환경에 새 버전의 애플리케이션을 배포합니다. CI/CD 파이프라인을 사용하려면 다음 단계를 완료하십시오.

1. Jenkins에 로그인하여 필요한 자격 증명을 생성합니다. 하나는 Magento 크레디고, 하나는 MariaDB 관리자 크레디고, 다른 하나는 MariaDB 루트 크레딧입니다.

2. Manage Jenkins > Manage Credentials로 이동한 후 해당 도메인을 클릭합니다.
3. 자격 증명 추가 를 클릭하고 종류를 암호 및 범위가 전역으로 설정된 사용자 이름으로 설정합니다. 자격 증명의 사용자 이름, 암호 및 ID를 입력하고 확인 을 클릭합니다.

4. 다른 두 자격 증명에도 같은 절차를 반복합니다.
5. 대시보드로 돌아가서 새 항목을 클릭하여 파이프라인을 생성한 다음 파이프라인을 클릭합니다.
6. Jenkinsfile에서 파이프라인을 복사합니다 **"여기"**.
7. Jenkins 파이프라인 섹션에 파이프라인을 붙여넣은 다음 저장을 클릭합니다.
8. Helm 차트 버전, 업그레이드할 Magento 응용 프로그램 버전, Astra 도구 키트 버전, Astra Control Center FQDN, API 토큰 및 인스턴스 ID를 포함한 각 세부 정보로 Jenkins 파이프라인의 매개 변수를 채웁니다. 운영 및 스테이징 환경 모두의 Docker 레지스트리, 네임스페이스 및 Magento IP를 지정하고 생성된 자격 증명의 자격 증명 ID도 지정합니다.

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxxx'
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'
DOCKER_REGISTRY = 'docker.io/netapp-solutions-cicd'
PROD_NAMESPACE = 'magento-prod'
PROD_MAGENTO_IP = 'x.x.x.x'
STAGING_NAMESPACE = 'magento-staging'
STAGING_MAGENTO_IP = 'x.x.x.x'
MAGENTO_CREDS = credentials('magento-cred')
MAGENTO_MARIADB_CREDS = credentials('magento-mariadb-cred')
MAGENTO_MARIADB_ROOT_CREDS = credentials('magento-mariadb-root-cred')
```

9. 지금 구축을 클릭합니다. 파이프라인은 실행을 시작하고 단계를 진행합니다. 응용 프로그램 이미지는 먼저 빌드되어

컨테이너 레지스트리에 업로드됩니다.

Build & Publish Segment	Build Docker Image	Publish Image to Registry	Protect & Deploy Segment	Install & Configure Pre-requisites	Download & Configure Astra Toolkit	Backup Tasks	Backup of Staging Env	Backup of Production Env	Deploy to Staging environment [Minor/Patch]	Deploy to Staging environment [Major]	Promote to Production?	Deploy to Production environment [Minor/Patch]	Deploy to Production environment [Major]	Delete temporary files
4s	24s	5s	213ms	40s	2s	290ms	1min 38s	1min 2s	6min 29s	229ms	361ms	2min 57s	200ms	850ms
3s														
18min 29s														

10. 애플리케이션 백업은 Astra Control을 통해 시작됩니다.

magento-prod
Available

App status
Healthy

App protection status
Partially Protected

Images

Protection schedule

Group

Cluster

docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16

Disabled

magento-prod

ocp-vmw

docker.io/bitnami/magento:2.4.1-debian-10-r11

docker.io/bitnami/mariadb:10.3.23-debian-10-r38

Overview

Data protection

Storage

Resources

Activity

Actions


Configure protection policy


Search


1-8 of 8 entries

<input type="checkbox"/>	Name	Ready	On-Schedule/On-Demand	Created ↑	Actions
<input type="checkbox"/>	upgrade-prod-2-4-1-debian-10-r20		On-Demand	2021/10/29 14:43 UTC	Running

11. 백업 단계가 성공적으로 완료된 후 Astra Control Center에서 백업을 확인합니다.


magento-prod
Available



App status
Healthy


App protection status
Partially Protected

Images
 docker.io/bitnami/elasticsearch:6.8.10-debian-10-r16
 docker.io/bitnami/magento:2.4.1-debian-10-r11
 docker.io/bitnami/mariadb:10.3.23-debian-10-r38

Protection schedule
 Disabled

Group
 magento-prod

Cluster
 ocp-vmw

Overview

Data protection

Storage

Resources

Activity

Actions

Configure protection policy

Search

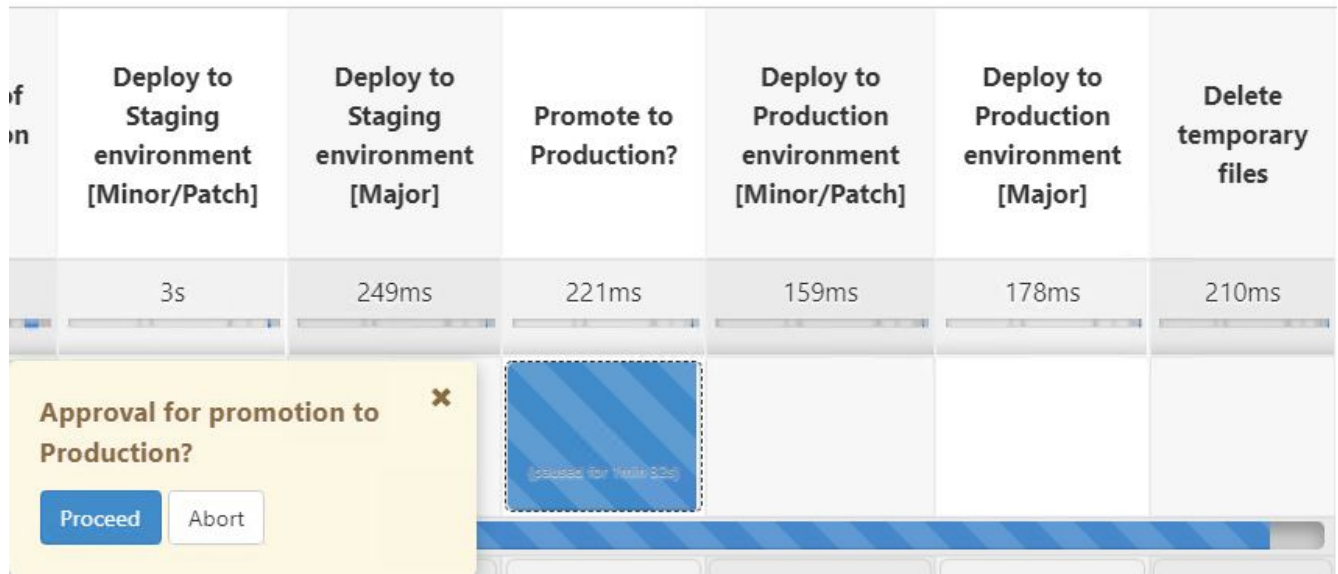
1-8 of 8 entries

Name	Ready	On-Schedule/On-Demand	Created	Actions
upgrade-prod-2-4-1-debian-10-r20	Healthy	On-Demand	2021/10/29 14:43 UTC	Available

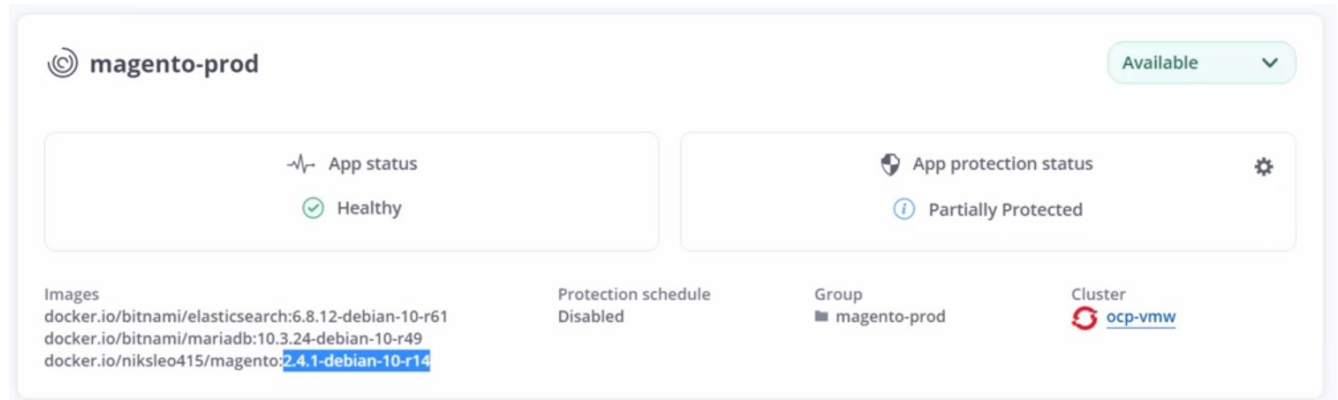
12. 그런 다음 새 버전의 응용 프로그램이 스테이징 환경에 배포됩니다.

Build & Publish Segment	Build Docker Image	Publish Image to Registry	Protect & Deploy Segment	Install & Configure Pre-requisites	Download & Configure Astra Toolkit	Backup Tasks	Backup of Staging Env	Backup of Production Env	Deploy to Staging environment [Minor/Patch]	Deploy to Staging environment [Major]	Promote to Production?	Deploy to Production environment [Minor/Patch]	Deploy to Production environment [Major]	Delete temporary files
4s	47s	7s	238ms	1min 25s	2s	273ms	1min 53s	1min 18s	5min 20s	211ms	337ms	2min 39s	187ms	780ms
3s	4min 16s	30s	485ms	7s	3s	153ms	6min 9s	5min 9s						
7min 1s														

13. 이 단계가 완료된 후 프로그램은 사용자가 프로덕션에 대한 배포를 승인하기를 기다립니다. 이 단계에서는 QA 팀이 일부 수동 테스트를 수행하고 생산을 승인한다고 가정합니다. 그런 다음 Approve(승인) 를 클릭하여 새 버전의 애플리케이션을 프로덕션 환경에 배포할 수 있습니다.



14. 운영 애플리케이션도 원하는 버전으로 업그레이드되었는지 확인합니다.



CI/CD 파이프라인의 일부로, 전체 애플리케이션 인식 백업을 생성하여 애플리케이션을 보호하는 기능을 입증했습니다. 전체 애플리케이션이 프로모션-운영 파이프라인의 일부로 백업되었으므로 고도로 자동화된 애플리케이션 구축에 대해 더욱 확신을 가질 수 있습니다. 애플리케이션의 데이터, 상태 및 구성을 포함하는 이 애플리케이션 인식 백업은 여러 DevOps 워크플로우에 유용할 수 있습니다. 중요한 워크플로 중 하나는 예상치 못한 문제가 발생할 경우 이전 버전의 응용 프로그램으로 롤백하는 것입니다.

Jenkins 툴을 사용하여 CI/CD 워크플로우를 시연했지만, 이 개념은 여러 도구와 전략에 대해 쉽고 효율적으로 추론할 수 있습니다. 이 사용 사례를 실제 작동 중인 경우 비디오를 시청하십시오 ["여기"](#).

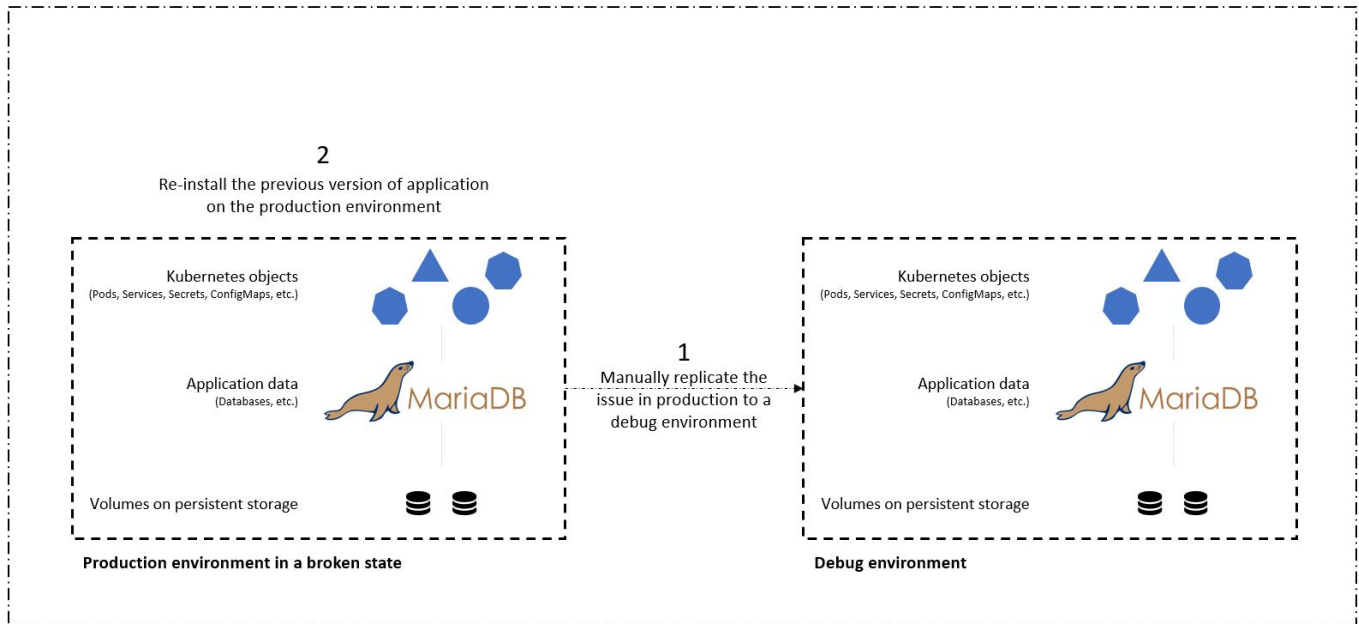
Astra Control을 사용하여 사후 분석을 용이하게 하고 애플리케이션을 복원합니다

개요

에 있습니다 ["첫 번째 사용 사례"](#) 또한, NetApp Astra Control Center를 사용하여 Kubernetes에서 애플리케이션을 보호하는 방법을 시연했습니다. 이 섹션에서는 NetApp Astra 툴킷에서 Python SDK를 사용하여 Astra Control을 통해 애플리케이션 백업을 개발 워크플로우에 직접 통합하는 방법을 설명합니다. 이 방식을 사용하면 CI/CD(Continuous Integration and Continuous Deployment) 프로세스 중에 주문형 백업을 자동화하여 개발 및 운영 환경을 보호할 수

있습니다. 이 애플리케이션 정합성 보장 데이터 보호 계층이 CI/CD 파이프라인 및 운영 애플리케이션에 추가되면서 프로세스에 문제가 발생하면 개발 프로세스가 안전해 비즈니스 연속성 모범 사례를 촉진합니다.

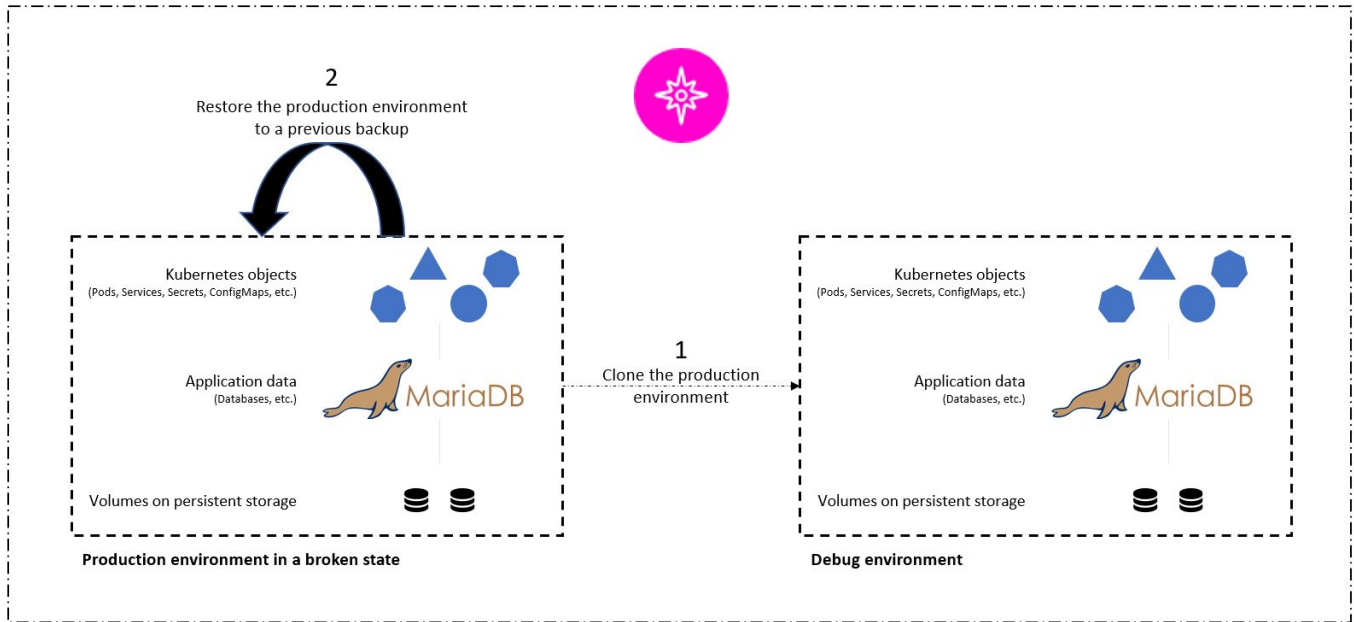
기존 워크플로우에서 응용 프로그램을 새 버전으로 업그레이드할 때 오류가 발생한 후 개발 팀은 고객이 제공하는 버그 보고서를 기반으로 실시간으로 문제 해결을 시도합니다. 또는 문제가 처음 발생할 때 팀이 병렬 디버깅 환경에 응용 프로그램을 다시 배포하여 해당 프로세스를 오프라인으로 만들 수 있습니다. 이전 버전에서 운영 환경으로 이전 코드 기반을 재배포하여 응용 프로그램을 작업 순서로 복원할 수 있습니다.



이 방식은 효과가 있지만, IT 팀은 문제가 발생했을 때 손상된 운영 앱의 상태가 운영 환경에 표시된 버전과 일치하는지 확인해야 합니다. 또한 리포지토리에서 코드를 가져와 시스템 이미지를 다시 배포하여 응용 프로그램을 양호한 실행 상태로 복원함으로써 정상 작동이 확인된 빌드를 운영 환경으로 승격하는 데 시간을 소비해야 합니다. 또한 이 시나리오에서는 프로덕션 데이터베이스 자체가 결함 있는 코드로 인해 손상되었는지 여부를 고려하지 않았습니다. 데이터베이스 데이터를 위한 별도의 백업 프로세스가 마련되어 있는 것이 이상적이지만 게시된 응용 프로그램의 상태와 일치한다고 가정해야 합니까? 여기서 Stateful 및 애플리케이션 정합성이 보장되는 백업, 복원 및 클론 생성 시 Astra Control이 제공하는 이점은 그 가치를 실제로 보여 줍니다.

먼저 Astra Control을 사용하여 응용 프로그램의 상태에 대한 사후 분석을 용이하게 할 수 있습니다. 우리는 응용 프로그램 일관성 있는 방식으로 버그 프로덕션 버전을 병렬 테스트 환경에 복제하는 방식으로 이 작업을 수행합니다. 이 환경을 버그에 따라 문제가 발생한 상태로 두고 실시간으로 문제를 해결할 수 있습니다.

또한 Astra Control은 운영 애플리케이션을 마지막 허용 가능한 백업(해당 코드 버전 이전)으로 복원할 수 있는 데이터 이동 없는 복원 기능을 지원합니다. 복원된 버전은 이전에 할당된 수신 IP를 포함하여 응용 프로그램 일관성 및 상태 저장 방식으로 이전 버그 프로덕션 응용 프로그램의 위치를 가정합니다. 따라서 프런트 엔드에 액세스하는 고객은 백업 버전으로의 전환에 대해 알지 못합니다.



사용 사례 검증을 위한 사전 요구사항

다음 도구 또는 플랫폼이 전제 조건으로 구축 및 구성되었습니다.

- Red Hat OpenShift Container Platform
- NetApp ONTAP 시스템에 백엔드가 구성된 OpenShift에 설치된 NetApp Astra Trident
- NetApp ONTAP 백엔드를 가리키는 기본 스토리지 클래스 구성
- OpenShift 클러스터에 설치된 NetApp Astra Control Center
- OpenShift 클러스터가 Astra Control Center에 관리 클러스터로 추가되었습니다.
- Jenkins가 OpenShift 클러스터에 설치되어 있습니다.
- 생산 환경에 설치된 Magento 응용 프로그램. 이 활용 사례의 운영 환경은 Red Hat OpenShift 클러스터의 'magento-prod'라는 네임스페이스입니다.
- 운영 애플리케이션은 Astra Control Center에서 관리합니다.
- Astra Control로 캡처한 운영 애플리케이션의 정상 작동이 확인된 백업입니다.

파이프라인 복제 및 복원

애플리케이션이 새 버전으로 업그레이드되었다는 점을 고려하면 프로덕션 환경('magento-prod')의 애플리케이션이 업그레이드 후 의도한 대로 작동하지 않습니다. 프런트 엔드 쿼리에서 반환되는 데이터가 요청과 일치하지 않거나 데이터베이스가 실제로 손상되었다고 가정해 보겠습니다. 파이프라인을 클론 복제 및 복원하려면 다음 단계를 완료하십시오.



This site can't be reached

10.63.172.243 took too long to respond.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)
- [Running Windows Network Diagnostics](#)

ERR_CONNECTION_TIMED_OUT

Reload

Details

1. Jenkins에 로그인하고 새 항목, 파이프라인을 차례로 클릭하여 파이프라인을 생성합니다.
2. Jenkinsfile에서 파이프라인을 복사합니다 **"여기"**.
3. Jenkins 파이프라인 섹션에 파이프라인을 붙여넣은 다음 저장을 클릭합니다.
4. 운영 환경의 현재 Magento 응용 프로그램 버전, Astra Control Center FQDN, API 토큰, 운영 및 디버그 환경의 인스턴스 ID 및 응용 프로그램 이름 또는 네임스페이스, 소스 및 대상 클러스터 이름과 같은 각 세부 정보로 Jenkins 파이프라인의 매개 변수를 채웁니다. 이 활용 사례를 위해 운영 환경은 'magento-prod'라는 네임스페이스이며, 디버그 환경은 Red Hat OpenShift 클러스터에 구성된 'magento-debug'라는 네임스페이스입니다.

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxx'
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'
PROD_APP_NAME = 'magento-prod'
DEBUG_APP_NAME = 'magento-debug'
DEBUG_NAMESPACE = 'magento-debug'
PROD_KUBERNETES_CLUSTER = 'ocp-vmw'
DEBUG_KUBERNETES_CLUSTER = 'ocp-vmw'
```

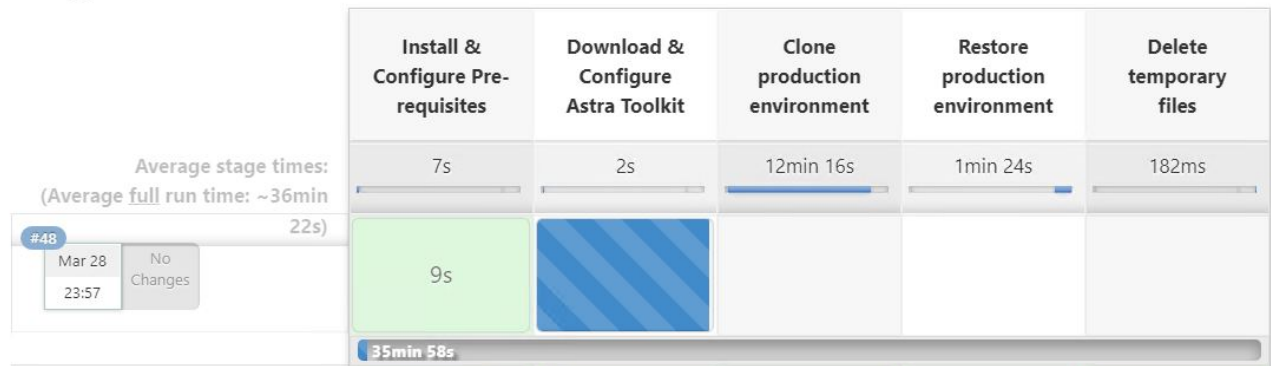
5. 지금 구축을 클릭합니다. 파이프라인은 실행을 시작하고 단계를 진행합니다. 응용 프로그램은 먼저 현재 상태에서 디버그 환경으로 복제되고 응용 프로그램은 알려진 작업 백업으로 복원됩니다.

Pipeline magento_clone-for-triage_restore-from-backup



Recent Changes

Stage View



6. 복제된 응용 프로그램이 버그 포함 버전인지 확인합니다.

← → ↻ ⓘ 10.63.172.245 ☆ ⓘ Update



This site can't be reached

10.63.172.245 took too long to respond.

Try:

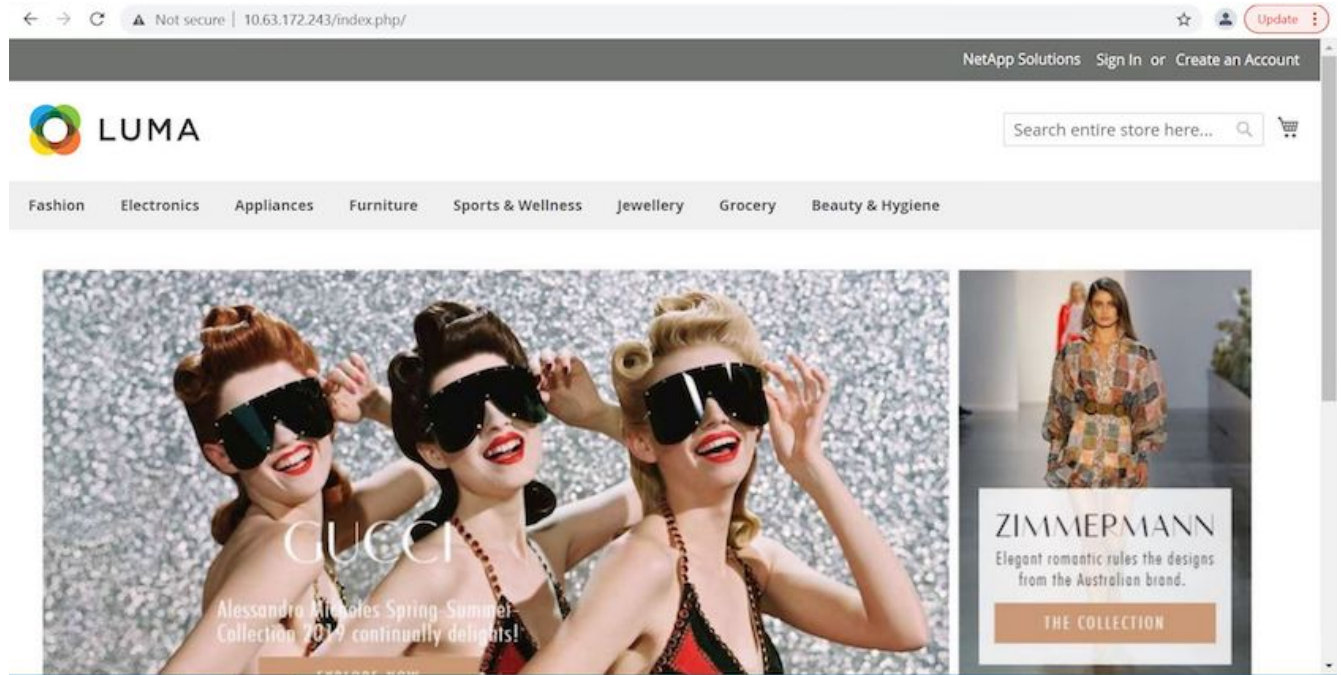
- Checking the connection
- [Checking the proxy and the firewall](#)
- [Running Windows Network Diagnostics](#)

ERR_CONNECTION_TIMED_OUT

Reload

Details

7. 운영 환경이 작업 중인 백업으로 복원되고 운영 중인 애플리케이션이 예상대로 작동하는지 확인합니다.



이 두 가지 작업을 함께 수행할 경우 일반 비즈니스 운영으로 더 빠르게 되돌릴 수 있습니다. 이 사용 사례를 실제 작동 중인 경우 비디오를 시청하십시오 ["여기"](#).

NetApp FlexClone 기술을 사용하여 소프트웨어 개발 가속화

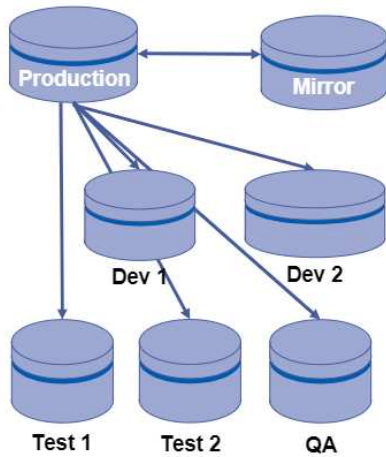
개요

Kubernetes 클러스터에 배포된 애플리케이션을 클론 복제하는 것은 현재 작업 중인 버전에 영향을 주지 않고 환경을 파트너와 공유하거나 개발 환경에서 새 버전의 코드를 테스트하여 워크플로우를 가속화하려는 개발자에게 매우 유용한 토크입니다. Kubernetes 애플리케이션의 상태 저장 및 애플리케이션 정합성이 보장되는 클론 복제는 NetApp Astra Control에 포함된 주요 기능이며, 애플리케이션의 백업 및 복원 기능도 제공합니다. 이와 더불어, 동일한 스토리지 백엔드를 사용하여 동일한 Kubernetes 클러스터 내에 애플리케이션을 클론 복제하면 Astra Control은 기본적으로 NetApp FlexClone 기술을 사용하여 영구 데이터 볼륨을 복제하므로 프로세스의 속도를 크게 높일 수 있습니다. 이 프로세스를 가속화하면 복제된 환경이 프로비저닝되고 몇 분 만에 사용할 수 있으므로 개발자가 테스트 또는 개발 환경을 다시 배포하는 데 비해 잠시 멈춰 작업을 다시 시작할 수 있습니다. 편의를 위해 NetApp Astra Control에서 제공되는 모든 기능을 API로 호출할 수 있으며, 이 API를 통해 Ansible과 같은 자동화 프레임워크와 쉽게 통합할 수 있습니다. 따라서 복제 절차를 시작하기 위해 플레이북이나 역할에서 약간의 변경만 필요하기 때문에 환경을 훨씬 빠르게 스테이징할 수 있습니다.

NetApp FlexClone 기술이란 무엇입니까?

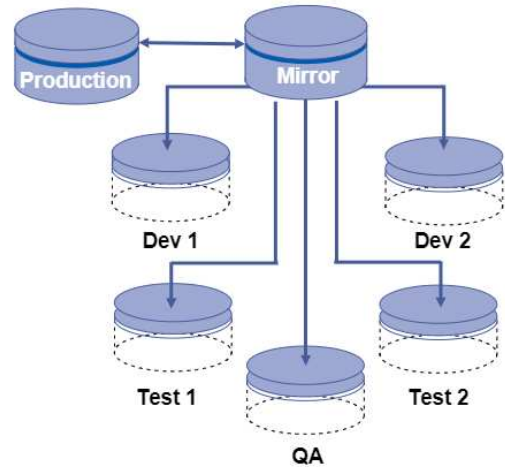
NetApp FlexClone 기술은 NetApp FlexVol의 쓰기 가능한 시점 스냅샷 기반 복사본입니다. 이러한 데이터는 거의 즉각적으로 프로비저닝되고 소스 볼륨의 모든 데이터를 포함하며 새 볼륨의 데이터가 소스에서 전환되기 전까지 추가 스토리지 공간을 사용하지 않습니다. 여러 데이터 복사본이 스테이징 목적으로 유용하고 스토리지 시스템에 이러한 볼륨 프로비저닝을 위한 리소스가 제한적인 경우, 이러한 리소스는 개발 또는 템플릿 기반 환경에서 주로 사용됩니다. 데이터를 여러 번 복사해야 하는 기존 스토리지 시스템과 비교하여 상당한 스토리지 공간과 시간을 소모하는 NetApp FlexClone 기술은 스토리지에 따른 작업을 가속화합니다.

Traditional Data Copies



Traditional physical copies take additional time and consume additional storage space

NetApp FlexClone Copies



NetApp FlexClone copies are near instantaneous and only consume space when written to

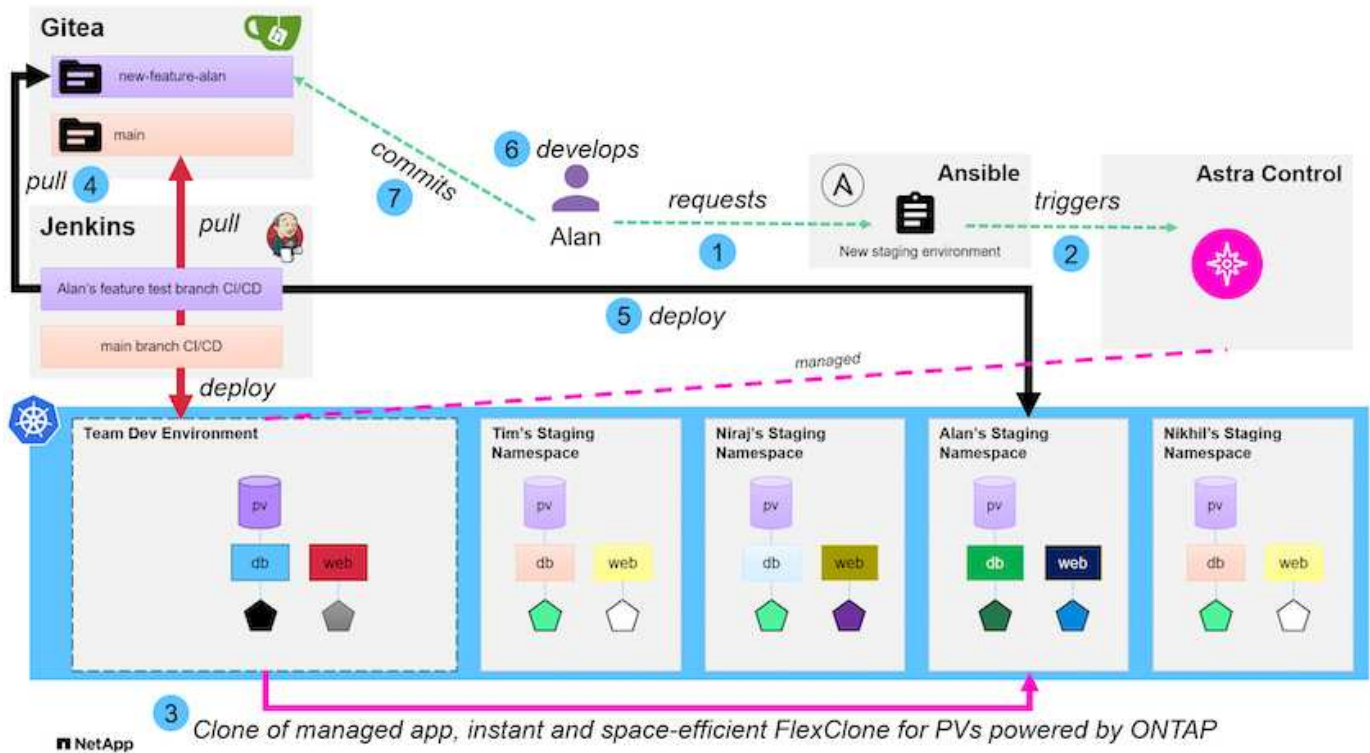
NetApp FlexClone 기술에 대한 자세한 내용은 의 페이지를 참조하십시오 "[NetApp 문서](#)".

필수 구성 요소

1. Red Hat OpenShift 4.6.8+, Rancher 2.5+ 또는 Kubernetes 1.19+와 같은 지원되는 Kubernetes 배포
2. NetApp Astra Control Center 21.12 이상
3. NetApp Astra Trident를 통해 스토리지 백엔드가 구성된 NetApp ONTAP 시스템입니다.
4. Ansible 2.9 이상
5. NetApp Astra Control에서 관리 대상 애플리케이션으로 클론 복제하려는 환경을 위한 템플릿입니다.

사용 사례 소개

이 사용 사례에서는 다음과 같은 워크플로를 시각화합니다.



1. 사용자가 Ansible 플레이북을 실행하여 새로운 스테이징 환경을 구축합니다.
2. Ansible은 URI-API 모듈을 사용하여 Astra Control을 호출하여 클론 생성 작업을 실행합니다.
3. Astra Control은 미리 프로비저닝된 템플릿 환경에서 클론 생성 작업을 실행하여 새로운 관리 애플리케이션을 생성합니다.



이 환경은 개발 중인 단일 독립 실행형 애플리케이션이거나 Jenkins CI/CD 파이프라인과 같은 전체 개발 환경일 수 있습니다.

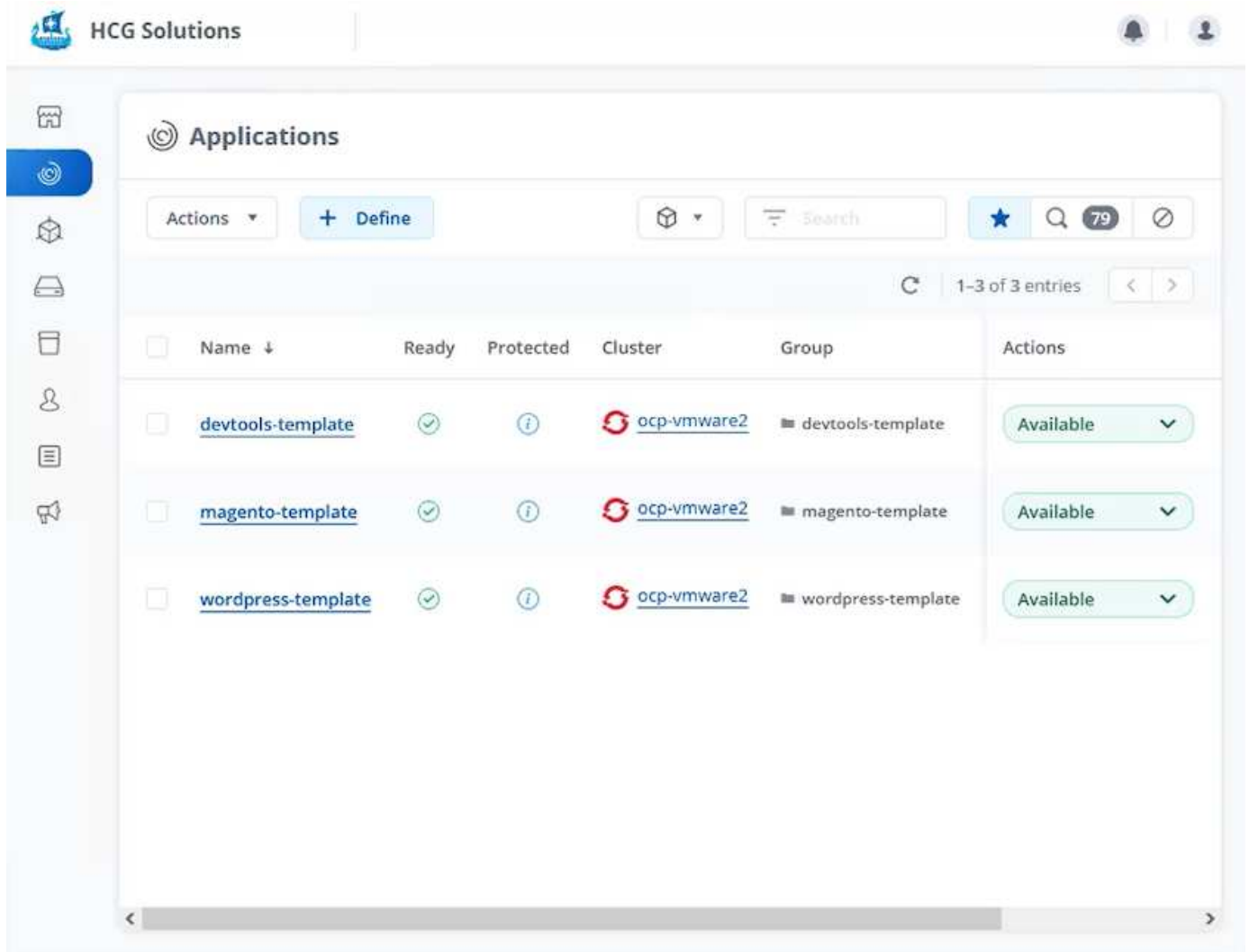
4. 그런 다음 사용자는 Gitea와 같은 온라인 저장소에서 해당 코드의 버전을 복제된 개발 환경으로 가져옵니다.
5. 새로운 버전의 애플리케이션은 NetApp Astra Control에서 구축 및 관리합니다.



이 두 프로세스는 모두 자동화할 수 있습니다.

6. 사용자는 이 복제 환경에서 새 코드를 개발할 수 있습니다.
7. 사용자가 개발 작업에 만족하면 코드를 호스팅된 저장소로 다시 푸시할 수 있습니다.

여기서 제시하는 사용 사례는 복제할 특정 환경 또는 애플리케이션에 대한 골드 템플릿이 있는지 여부에 따라 달라집니다. 저희 환경에서는 Wordpress 배포, Magento 배포, DevTools이라는 제목의 Gitea가 있는 Jenkins CI/CD 환경 등 세 개의 템플릿을 만들었습니다.



각 환경은 NetApp Astra Control을 통해 관리되며, NetApp Astra Trident가 제공하는 NFS 백엔드와 함께 NetApp ONTAP 스토리지 시스템에 현재 저장된 영구 볼륨이 있습니다.

사용 사례 검증

1. 클론 복제 역할 및 애플리케이션 업데이트 플레이북을 포함하는 NetApp 솔루션 엔지니어링 팀에서 제공하는 Ansible 툴킷을 복제하십시오.

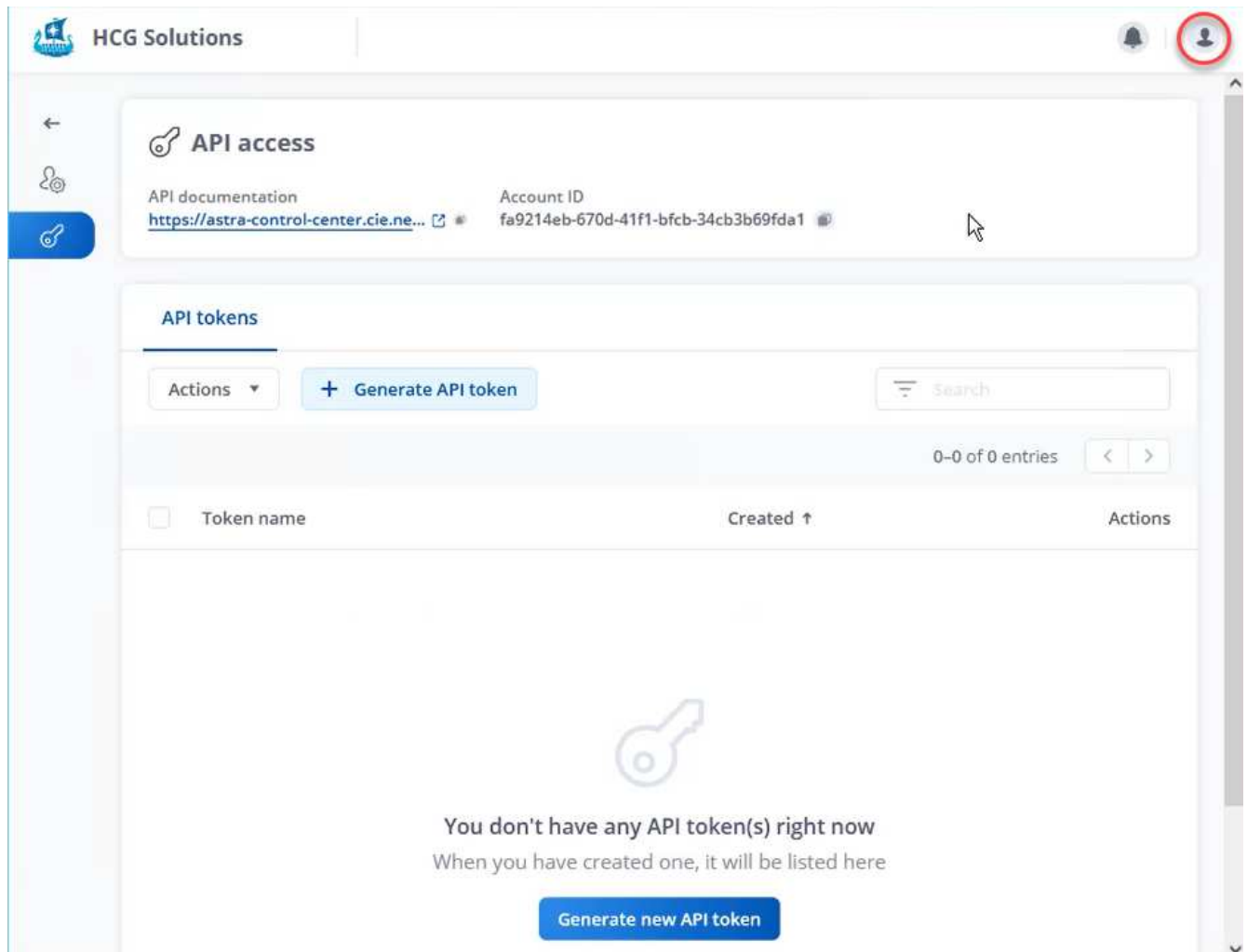
```
[netapp-user@rhel7 ~]$ git clone https://github.com/NetApp-Automation/na_astra_control_suite.git
[netapp-user@rhel7 ~]$ cd na_astra_control_suite
```

2. VAR/CLONE_VAR.Lyml을 편집하여 Astra Control 환경에 맞는 글로벌 가치를 입력합니다.

```
astra_control_fqdn: astra-control-center.example.com
astra_control_account_id: "xxxx-xxxx-xxxx-xxxx-xxxx"
astra_control_api_token: "xxxxxx"
```



입력해야 하는 글로벌 환경 값은 API 액세스 메뉴 아래의 NetApp Astra Control의 사용자 프로필 아이콘 아래에서 확인할 수 있습니다.



3. 글로벌 변수가 완료된 경우 복제할 특정 애플리케이션의 값을 선택할 수 있습니다. devtools 환경을 Alan-devtools라고 하는 개인 환경으로 복제하려면 다음을 수행합니다.

```
clone_details:
  - clone_name: alan-devtools
    destination_namespace: alan-dev-namespace
    source_cluster_name: ocp-vmware2
    destination_cluster_name: ocp-vmware2
    source_application_name: devtools-template
```



클론 복제 프로세스에서 NetApp FlexClone 기술을 활용하려면 'rc-cluster'와 'dest-cluster'가 동일해야 합니다.

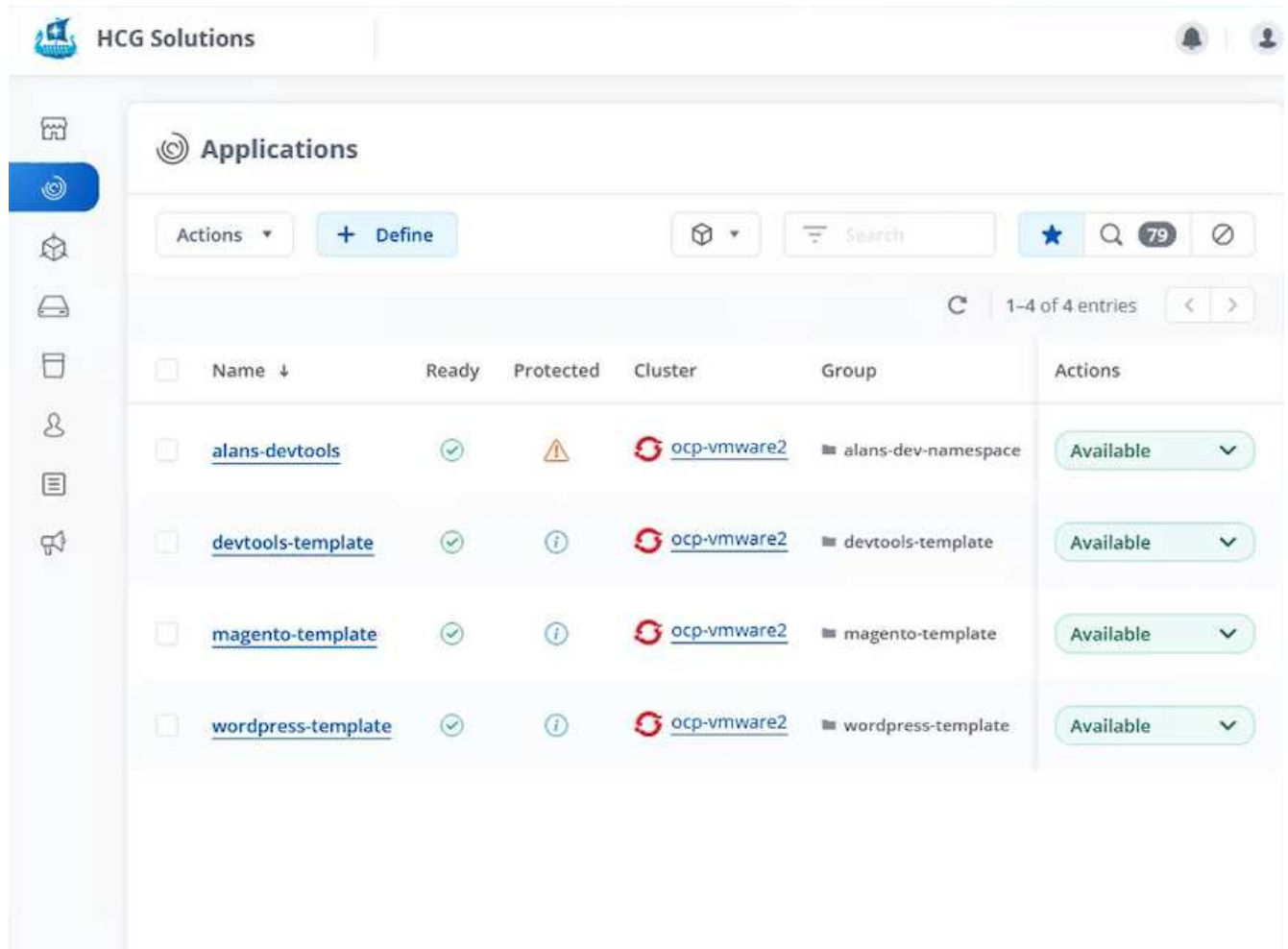
4. 이제 플레이북을 실행하여 애플리케이션을 복제할 수 있습니다.

```
[netapp-user@rhel7 na_astra_control_suite]$ ansible-playbook -K clone_app_playbook.yml]
```



기록된 Playbook은 루트 사용자 또는 sudo 프로세스를 통해 "-K" 인수를 전달하여 에스컬레이션할 수 있는 사람이 실행해야 합니다.

5. Playbook의 실행이 완료되면 복제된 애플리케이션이 Astra Control Center 콘솔에서 사용 가능한 것으로 표시됩니다.



6. 그런 다음, 사용자가 애플리케이션이 배포된 Kubernetes 환경에 로그인하여 애플리케이션이 새 IP 주소로 표시되는지 확인하고 개발 작업을 시작할 수 있습니다.

이 사용 사례 데모와 응용 프로그램 업그레이드 예제는 [여기](#)를 참조하십시오.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.