



DevOps with NetApp Astra

NetApp Solutions

NetApp
May 17, 2024

This PDF was generated from https://docs.netapp.com/us-en/netapp-solutions/containers/dwn_overview_devops.html on May 17, 2024. Always check docs.netapp.com for the latest.

Table of Contents

- TR-4919: DevOps with NetApp Astra 1
 - Use cases 1
 - Business value 1
 - Technology overview 1
 - DevOps Overview 2
 - NetApp storage systems overview 4
 - NetApp Storage Integration Overview 5
 - Use-case validation: DevOps with NetApp Astra 6
 - Videos and demos: DevOps with NetApp Astra 22
 - Additional Information: DevOps with NetApp Astra 23

TR-4919: DevOps with NetApp Astra

This technical report outlines how NetApp makes DevOps use-cases easy and efficient in multiple fronts, when using containerized applications. It starts by detailing the NetApp storage systems and their integration with Kubernetes platforms by making use of the Astra portfolio. Lastly, a number of solution validations and real world use cases are explored and documented.

Alan Cowles and Nikhil M Kulkarni, NetApp

Use cases

The DevOps with NetApp Astra solution is architected to deliver exceptional value for customers with the following use cases:

- Easy to deploy and manage applications and development environments deployed on top of supported Kubernetes distributions.
- Discussion of real-world use cases for DevOps workflows and examples of the tools and methods that NetApp can provide to make adoption and use of these methods easier.
- Exploration of how application-consistent snapshot, backups, and clones can be used to enhance the DevOps experience.

Business value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure, such as the following:

- High availability at all layers in the stack so that workflows are never interrupted.
- Ease of deployment and management procedures for the end user.
- API-driven and programmable infrastructure to keep up with microservices and developer agility.
- Ability to scale infrastructure independently and in an automated fashion, based on workload demands.
- Protecting applications alongside their backing persistent data sets for DevOps workflows accelerate time to market by not having to rely on redeployments or manual copying of data.

Recognizing these capabilities and challenges, this technical report outlines the process of improving and simplifying DevOps use cases for containerized applications using the wide portfolio of NetApp products.

Technology overview

The DevOps with NetApp solution contains the following major components:

DevOps practices

DevOps practices focus on automated, repeatable, and easily manageable operations that enhance the development workflow by allowing the end user to control the environment in which they are developing their

code. This solution provides several examples and use cases in which NetApp technology can be of the greatest benefit to such operations.

Container orchestration

There are numerous container orchestration platforms in use today. Although most of these platforms are based on Kubernetes, each has pros and cons. So it is important to understand feature sets and integrations when selecting a container orchestration platform for DevOps workflows. With the NetApp Astra suite of products, we support the following platforms for full-fledged DevOps use cases:

- [Red Hat OpenShift 4.6.8+](#)
- [Rancher 2.5+](#)
- [Kubernetes 1.20+](#)
- [VMware Tanzu Kubernetes Grid 1.4+](#)
- [VMware Tanzu Kubernetes Grid Integrated Edition 1.12.2+](#)

NetApp storage systems

Unresolved directive in containers/dwn_solution_overview.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=solution_overview_netapp_storage_intro]

NetApp storage integrations

Unresolved directive in containers/dwn_solution_overview.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=solution_overview_netapp_storage_integrations_intro]

DevOps Overview

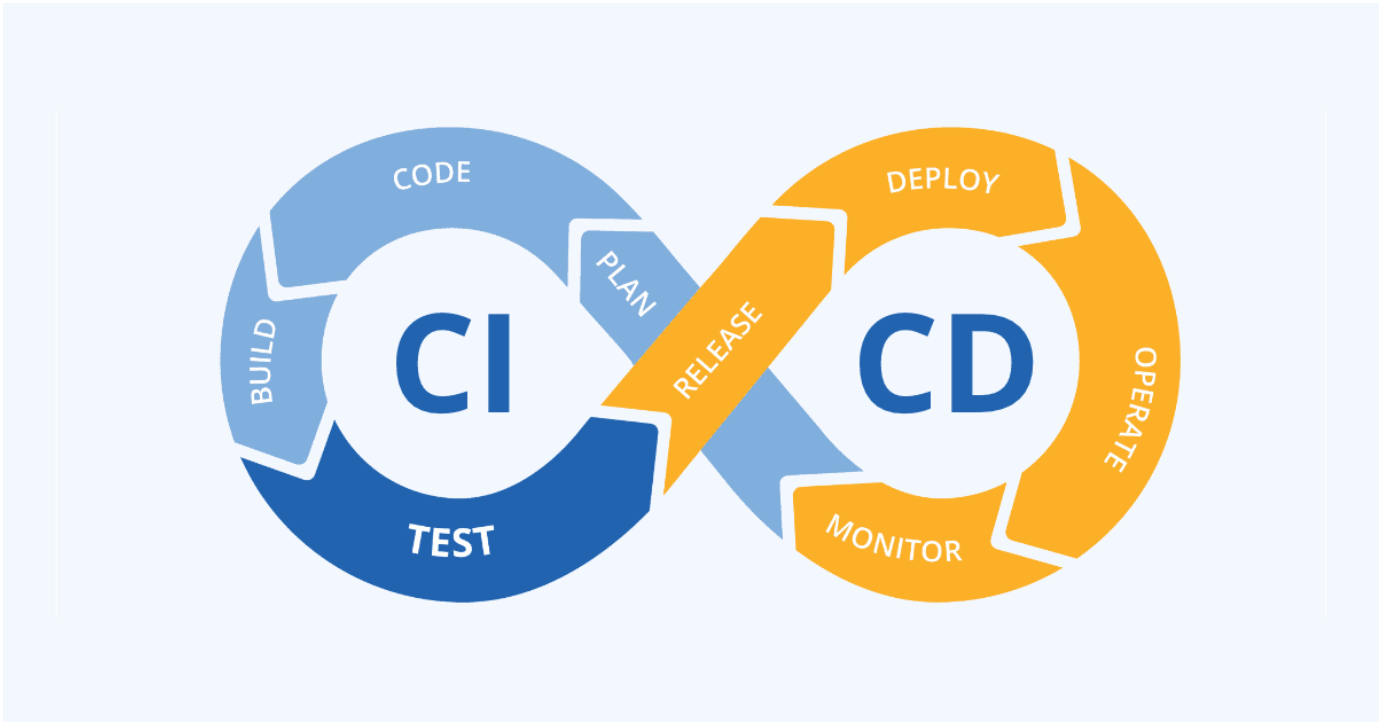
Over the past several years, organizations that build software have been embracing the concepts of DevOps. DevOps practices break down organizational barriers, bringing development and operations teams closer together. DevOps practices also empower the teams to accelerate delivery, increase availability, and make services and applications more stable, thus improving the team's productivity. In addition, adoption of an automation framework is also a key ingredient of success — from building, testing, and operating applications at scale or managing a fully automated infrastructure platform or stack. Below we discuss some primary use cases for DevOps where NetApp solutions can be implemented to help enhance the experiences that DevOps practitioners encounter during their daily practice.

DevOps use cases

Although DevOps does not have a single, universally accepted definition, solutions for DevOps practitioners typically contain similar constructs or ideologies that enable easy implementation, repetition, and management at scale. The following sections describe potential use cases for DevOps workflows enabled by NetApp solutions.

Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD)

Continuous Integration, Continuous Delivery, and Continuous Deployment (CI/CD) is a coding philosophy that encourages developers to implement and transform their coding practices by establishing a method by which they can consistently update, test, and deploy their code in an automated fashion. The most popular method by which CI/CD is implemented in most DevOps workflows is that of the CI/CD pipeline, and there are several third-party software applications that can help achieve this.



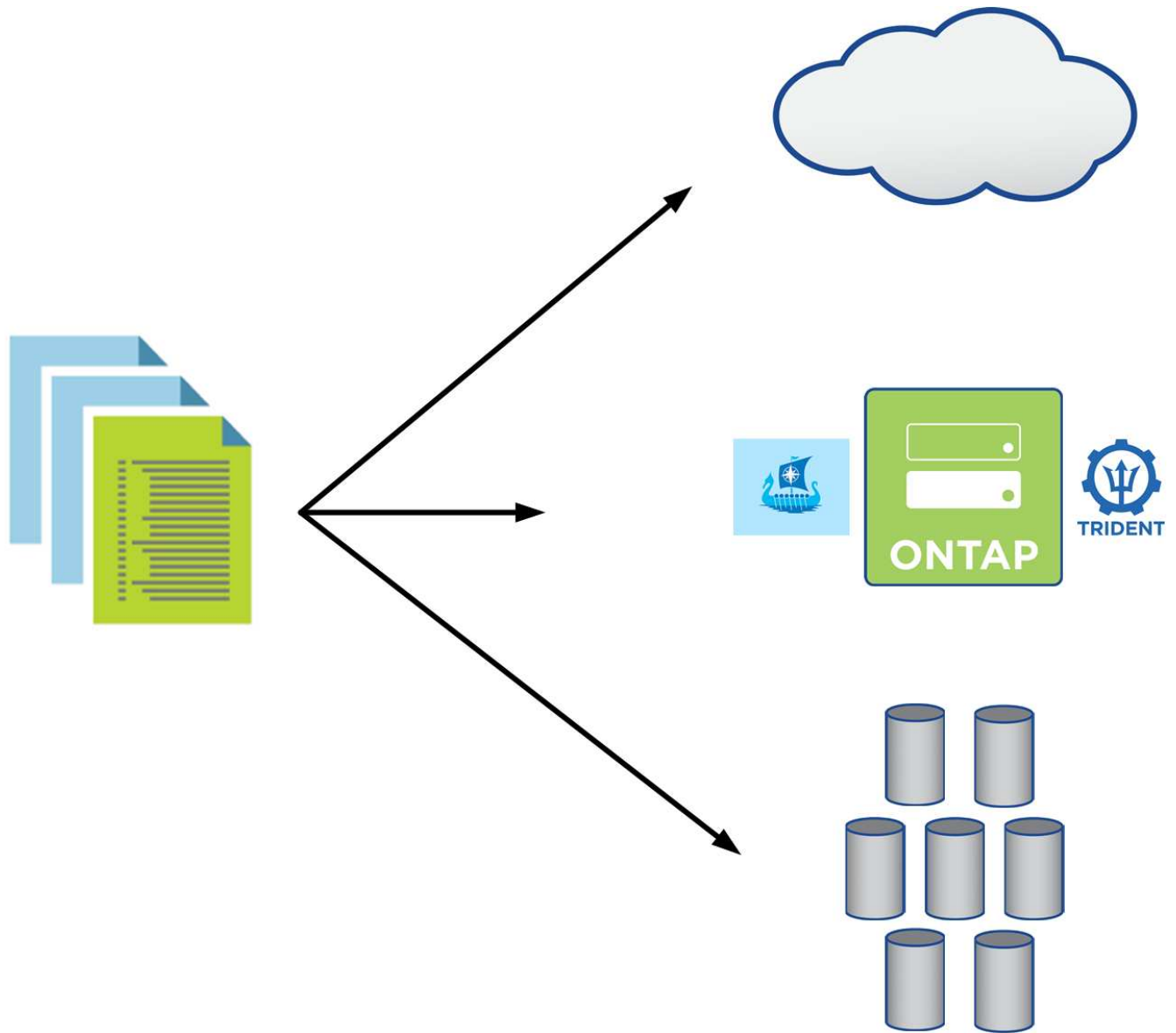
See the following examples of popular applications that can help with CI/CD-type workflows:

[ArgoCD](#)
[Jenkins](#)
[Tekton](#)

Some of the use cases included later in this technical report have been demonstrated in Jenkins, but the primary CI/CD principles can be applied to whatever tool an organization has implemented in their own practices.

Infrastructure as code

Infrastructure as code helps provision and manage IT resources through automated commands, APIs, and software development kits (SDK). This concept greatly enhance the DevOps experience by removing physical data center or resource limitations that could prevent developers from meeting their objectives.



End users often use programming languages such as [Python](#) or automation tools such as [Ansible](#) or [Puppet](#) to create automated and repeatable infrastructure scaling actions that can be called by developers when needed.

Both NetApp ONTAP and Astra Control contain public facing APIs and ansible modules or software development toolkits that make automating operations very easy to adopt and integrate into DevOps processes.

NetApp storage systems overview

NetApp has several storage platforms that are qualified with Astra Trident and Astra Control to provision, protect and manage data for containerized applications and thus help in defining and maximizing DevOps throughput.

Unresolved directive in containers/dwn_overview_netapp.adoc - include::.../_include/containers_common_intro_sections.adoc[tags=netapp_overview_page;!netapp_overview_page_element]

NetApp ONTAP

NetApp ONTAP is a powerful storage-software tool with capabilities such as an intuitive GUI, REST APIs with automation integration, AI-informed predictive analytics and corrective action, non-disruptive hardware upgrades, and cross-storage import.

Unresolved directive in containers/dwn_netapp_ontap.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=netapp_ontap_page]

NetApp Storage Integration Overview

NetApp provides a number of products which assist our customers with orchestrating and managing persistent data in container based environments.

Unresolved directive in containers/dwn_overview_storint.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=storage_integration_overview]

NetApp Astra Control overview

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment, powered by NetApp's trusted data protection technology.

Unresolved directive in containers/dwn_overview_astra.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=astra_cc_overview]

For a detailed installation and operations guide on Astra Control Center, follow the documentation [here](#).

Astra Control Center automation

Astra Control Center has a fully functional REST API for programmatic access. Users can use any programming language or utility to interact with Astra Control REST API endpoints. To learn more about this API, see the documentation [here](#).

If you are looking for a ready-made software development toolkit for interacting with Astra Control REST APIs, NetApp provides a toolkit with Astra Control Python SDK, which you can download [here](#).

If programming is not appropriate for your situation and you would like to use a configuration management tool, you can clone and run the Ansible playbooks that NetApp publishes [here](#).

Astra Trident Overview

Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift.

Unresolved directive in containers/dwn_overview_trident.adoc -
include::.../_include/containers_common_intro_sections.adoc[tags=trident_overview]

Refer to the documentation [here](#) to install and use Astra Trident.

Use-case validation: DevOps with NetApp Astra

The following use cases have been validated for DevOps with NetApp Astra:

- [Integrate Protection into CI/CD Pipelines with NetApp Astra Control](#)
- [Leverage Astra Control to facilitate Post-mortem Analysis and Restore the Application](#)
- [Accelerating Software Development with NetApp FlexClones](#)

Integrate Protection into CI/CD Pipelines with NetApp Astra Control

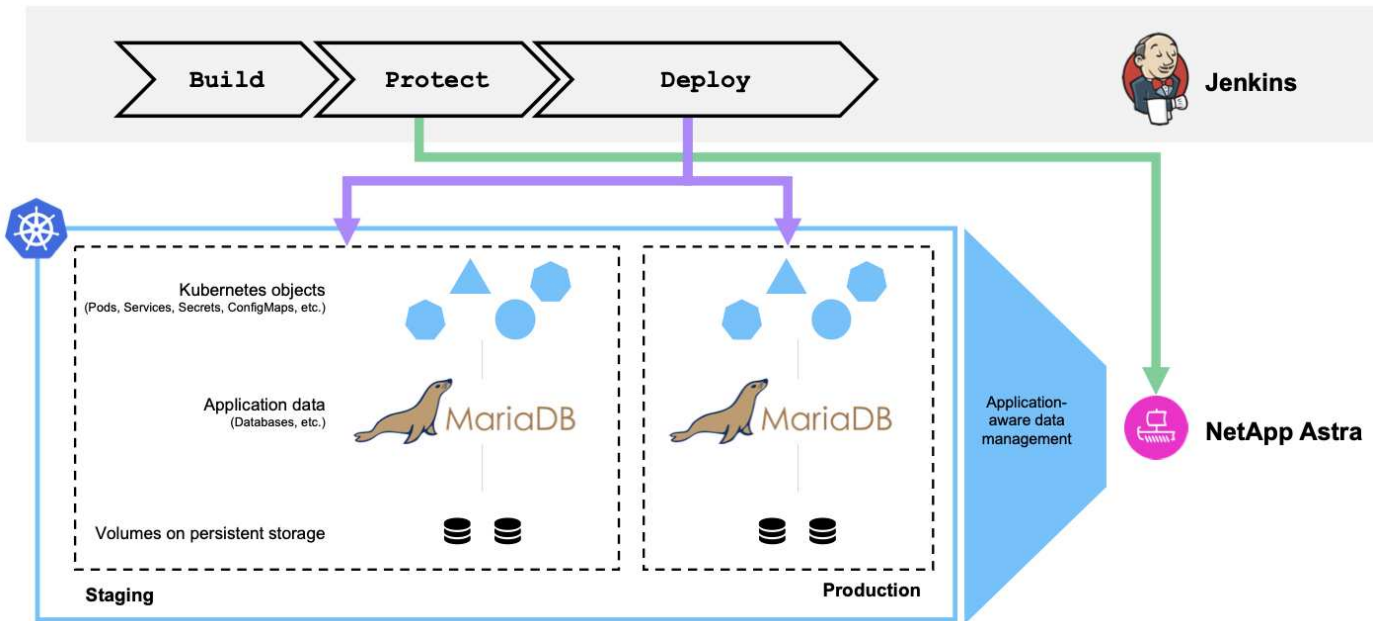
NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment, powered by NetApp's trusted data protection technology.

Overview

One of the most common uses of DevOps workflows is continuous integration and continuous deployment (CI/CD) pipelines that build, integrate, and run automated test suites on applications as developers commit new code. DevOps engineers and site-reliability engineers (SREs) typically have pipelines dedicated to the various workflows for new feature development, regression testing, bug fixes, quality engineering, and other functions in the development process.

As teams increase their level of automation, the pace of change for in-production applications can feel overwhelming. Therefore, some teams prefer to protect in-production applications or services. In addition to protecting the code and container images, they also want to protect the application state, configuration data (such as Kubernetes objects and resources associated with the application), and an application's persistent data.

In this use case, we take a closer look at a promotion-to-production pipeline that deploys a new version of an application: first into a staging environment and then into a production environment. This example applies equally to the major public clouds and also to an on-premises environment. Although we show the deployment of one version of the app, the pipeline can also be used with other strategies, such as blue/green or canary deployment. As part of the CI/CD pipeline, we're going to protect the application by creating a complete application backup. An application-aware backup of the in-production application and its data, state, and configuration can be useful for numerous DevOps workflows.



The application used for validating this use-case was [Magento](#), an e-commerce solution with a web-based front end; an Elasticsearch instance for search and analysis features; and a MariaDB database that tracks all the shopping inventory and transaction details. This containerized application was installed in a Red Hat OpenShift cluster. Every pod in the application used persistent volumes to store data. The persistent volumes were automatically created by NetApp Astra Trident, the Container Storage Interface-compliant storage orchestrator for Kubernetes that enables storage to be provisioned on NetApp storage systems. Further, to utilize the Astra Control Center's application protection capabilities, the application in question was managed by Astra Control, which was then used to trigger application backups that stored the state of the application along with the data held in persistent volumes. We used the [NetApp Astra Control Python SDK](#) to automate the process of triggering application backups, which was then introduced into a CI/CD pipeline. This pipeline was created and executed using a popular CI/CD tool called [[Jenkins](#)] to automate the flow to build, protect, and deploy the application.

Let us run through the prerequisites and procedure to introduce protection in a CI/CD pipeline.

Use-case validation prerequisites

The following tools or platforms were deployed and configured as prerequisites:

1. Red Hat OpenShift Container Platform
2. NetApp Astra Trident installed on OpenShift with a backend to NetApp ONTAP system configured
3. A default storageclass configured, pointing to a NetApp ONTAP backend
4. NetApp Astra Control Center installed on an OpenShift cluster
5. OpenShift cluster added as a managed cluster to Astra Control Center
6. Jenkins installed on an OpenShift cluster and configured with an agent node with a Docker engine installed on it

Installing the application

Let's start with the initial installation of the application in the staging and production environments. For the purpose of this use case, this step is a prerequisite, so it is performed manually. The CI/CD pipeline is used for subsequent build and deploy workflows as a result of new version releases of the application.

The production environment in this use case is a namespace called `magento-prod`, and the corresponding staging environment is a namespace called `magento-staging` configured on the Red Hat OpenShift cluster. To install the application, complete the following steps:

1. Install the Magento application using bitnami helm chart on the production environment. We use RWX PVs for Magento and Mariadb pods.

```
[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 chart requires a LoadBalancer service to expose the Magento GUI service. We used [MetalLB](#) for providing an on-prem load balancer service in this example.

2. After a few minutes, verify that all pods and services are running.

```
[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. Repeat the same procedure for the staging environment.

Manage the Magento application in Astra Control Center

1. Navigate to Applications and select the Discovered applications tab.
2. Click the ellipsis against the Magento application in the production environment (`magento-prod`), and click Manage.
3. The Magento application is now managed by the Astra Control Center. All operations supported by Astra Control can be performed on the application. Note the version of the application as well.

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. Repeat the steps for managing the Magento application in the staging environment (magento-staging).

CI/CD pipeline with integrated protection

When we work with new versions of applications, we use a CI/CD pipeline to build the container image, take backups of both the staging and production environments, deploy the new version of the application to the staging environment, wait for approval to promotion to production, and then deploy the new version of the application to the production environment. To use a CI/CD pipeline, complete the following steps:

1. Log into Jenkins, and create the required credentials: one for Magento creds, one for Mariadb admin creds, and the third for Mariadb root creds.
2. Navigate to Manage Jenkins > Manage Credentials and click the appropriate domain.
3. Click Add Credentials, and set the kind to Username with password and scope set to Global. Enter the username, password, and an ID for the credentials and click OK.

4. Repeat the same procedure for the other two credentials.
5. Go back to the Dashboard, create a pipeline by clicking New Item, and then click Pipeline.
6. Copy the pipeline from the Jenkinsfile [here](#).
7. Paste the pipeline into the Jenkins pipeline section and then click Save.
8. Fill the parameters of the Jenkins pipeline with the respective details including the helm chart version, the Magento application version to be upgraded to, the Astra toolkit version, the Astra Control Center FQDN, the API token, and its instance ID. Specify the docker registry, namespace, and Magento IP of both production and staging environments, and also specify the credential IDs of the credentials created.

```

MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxxxx'
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. Click Build Now. The pipeline starts executing and progresses through the steps. The application image is first built and uploaded to the container registry.

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. The application backups are initiated via Astra Control.

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 | [Camera] [Share]

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. After the backup stages have completed successful, verify the backups from the Astra Control Center.

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 | [Camera] [Share]

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	Available ▼

12. The new version of the application is then deployed to the staging environment.

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. After this step is completed, the program waits for the user to approve deployment to production. At this stage, assume that the QA team performs some manual testing and approves production. You can then click Approve to deploy the new version of the application to the production environment.

	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
	3s	249ms	221ms	159ms	178ms	210ms

Approval for promotion to Production?

Proceed

(paused for 1min 3.0s)

14. Verify that the production application is also upgraded to the desired version.

magento-prod Available

App status ✔ Healthy

App protection status ⚙️

Partially Protected

Images

docker.io/bitnami/elasticsearch:6.8.12-debian-10-r61

docker.io/bitnami/mariadb:10.3.24-debian-10-r49

docker.io/niksleo415/magento:2.4.1-debian-10-r14

Protection schedule

Disabled

Group

magento-prod

Cluster

ocp-vmw

As part of the CI/CD pipeline, we demonstrated the ability to protect the application by creating a complete application-aware backup. Because the entire application has been backed up as part of the promotion-to-production pipeline, you can feel more confident about highly automated application deployments. This application-aware backup containing the data, state, and configuration of the application can be useful for numerous DevOps workflows. One important workflow would be to roll back to the previous version of the application in case of unforeseen issues.

Although we demonstrated a CI/CD workflow through with Jenkins tool, the concept can easily and efficiently be extrapolated to different tools and strategies. To see this use case in action, watch the video below.

Data Protection in CI/CD pipeline with Astra Control Center

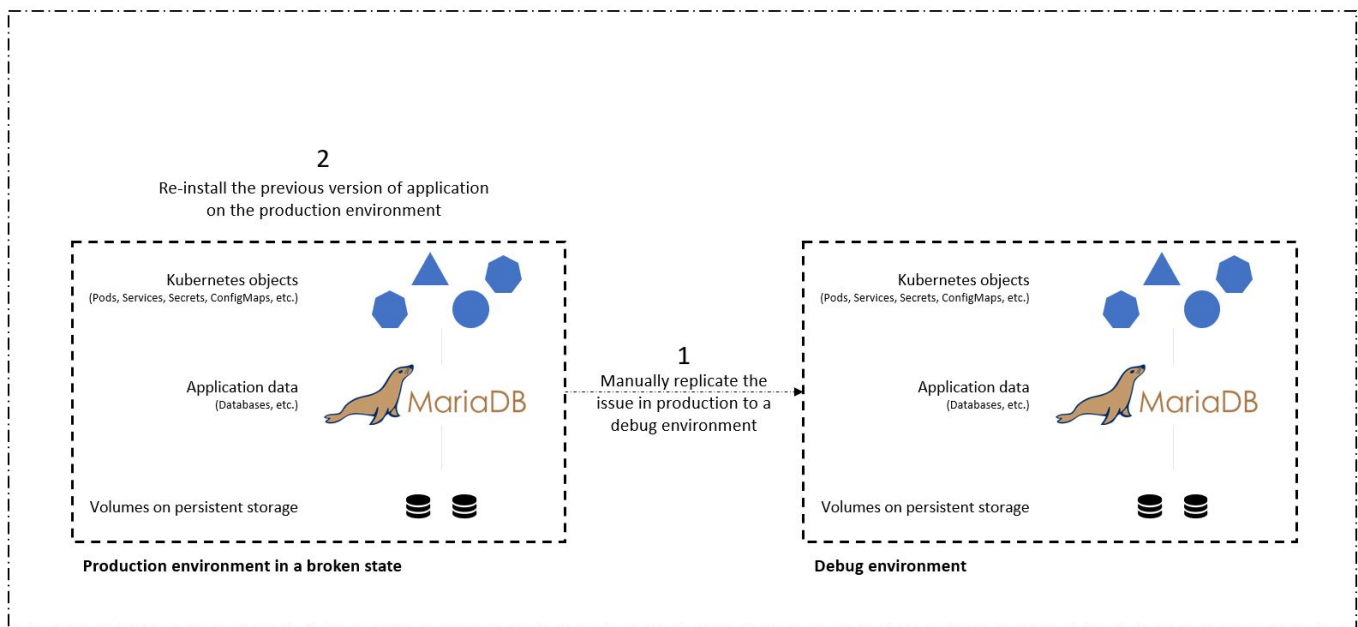
Use Astra Control to facilitate post-mortem analysis and restore the application

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads, deployed in an on-prem environment, powered by NetApp’s trusted data protection technology.

Overview

In the [first use case](#), we demonstrated how to use NetApp Astra Control Center to protect your applications in Kubernetes. That section describes how to integrate application backups via Astra Control directly into your development workflow by using the Python SDK in the NetApp Astra toolkit. This approach allows for the protection of development and production environments by automating on-demand backups during the continuous integration and continuous deployment (CI/CD) process. With this extra layer of application-consistent data protection added to the CI/CD pipeline and the production applications, the development processes is safe if something goes wrong in the process, which promotes good business-continuity practices.

In a traditional workflow, after encountering a failure when the application is upgraded to a new version, the development team would attempt to troubleshoot the issue in real time based on bug reports being provided by customers. Alternatively, at the first sign of trouble, the team could attempt to redeploy the application to a parallel debugging environment to take that process offline. They could redeploy an older code base from a previous version into production, which would restore the application to working order.

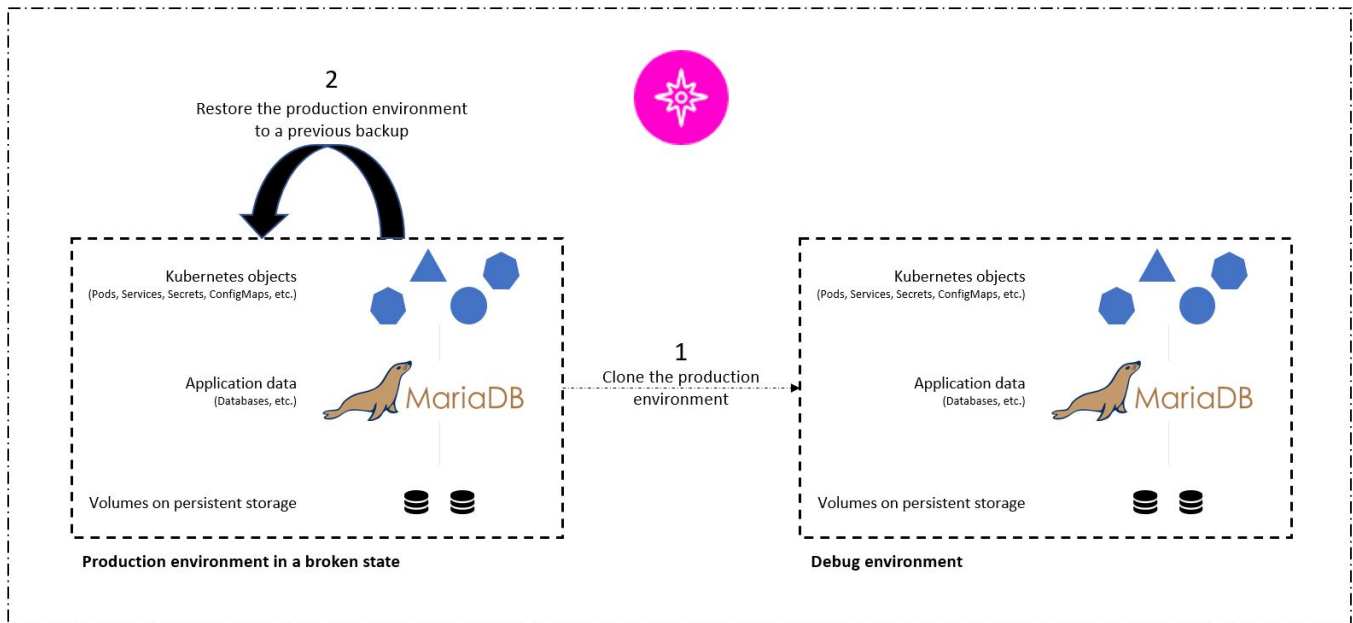


Although this approach works, the team would have to make sure that the state of the broken production app matched that of the version seen in production when the issues occurred. They would also have to spend time promoting the known-good build into production by fetching code from their repository and redeploying the machine images to restore the application to a good running state. Also, in this scenario, we didn’t consider whether the production database itself was corrupted by the faulty code. Ideally, there are separate backup processes in place for the database data, but must we assume that they’re consistent with the state of the application as it was published? This is where the benefits of stateful and application-consistent backups,

restores and clones with Astra Control really show their value.

First, we can use Astra Control to facilitate post-mortem analysis on the state of the application. We do this by cloning the buggy production version to a parallel testing environment in an application-consistent manner. Having this environment set aside in its bug-ridden state enable us to troubleshoot the problem in real time.

Furthermore, Astra Control supports the in-place restore capability that allows us to restore the production application to a last acceptable backup (that preceded the afflicted version of code). The restored version assumes the position of the previous, buggy production application, in an application-consistent and stateful manner, including the ingress IP previously assigned. As a result, customers accessing the front end would be unaware of the transition to the backup version.



Use-case validation prerequisites

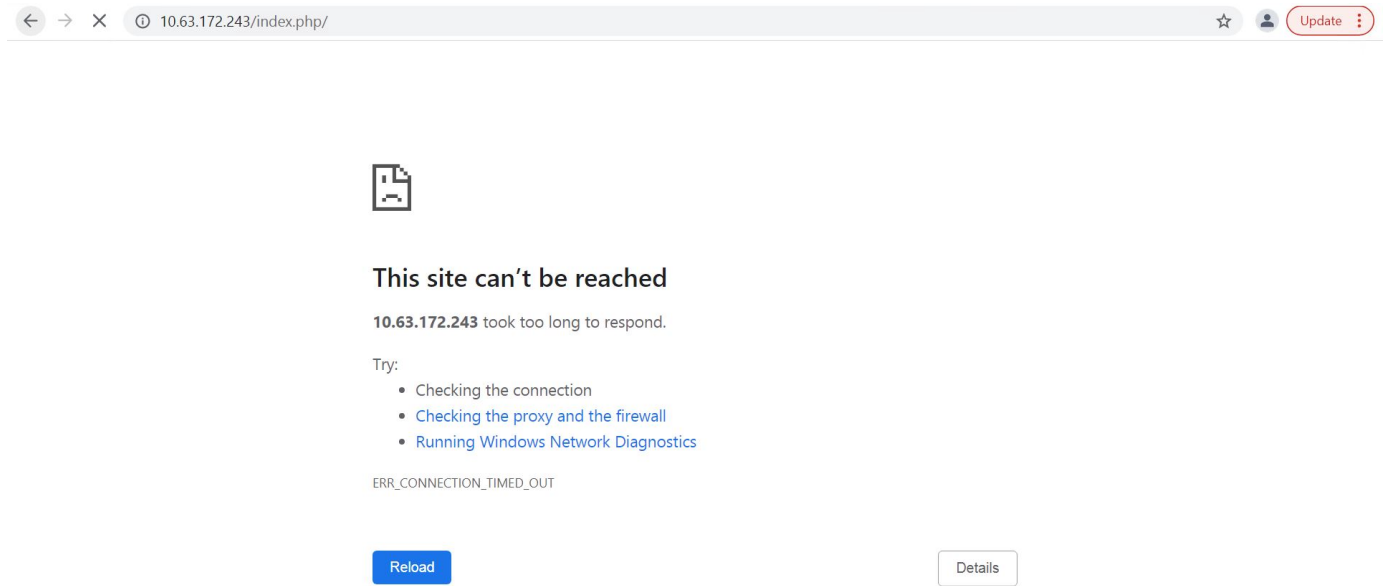
The following tools or platforms were deployed and configured as prerequisites:

- Red Hat OpenShift Container Platform.
- NetApp Astra Trident installed on OpenShift with a backend configured to a NetApp ONTAP system.
- A default storageclass configured, pointing to a NetApp ONTAP backend.
- NetApp Astra Control Center installed on an OpenShift cluster.
- OpenShift cluster added as a managed cluster to Astra Control Center.
- Jenkins installed on an OpenShift cluster.
- Magento application installed in the production environment. The production environment in this use case is a namespace called 'magento-prod' in a Red Hat OpenShift cluster.
- Production application managed by Astra Control Center.
- Known-good backup(s) of the production application captured with Astra Control.

Clone and restore pipeline

Considering that the application has been upgraded to a new version, the application in the production environment (`magento-prod`) isn't behaving as intended after the upgrade. Let's assume that the data being

returned by front-end queries doesn't match the request or that the database has in fact been corrupted. To clone and restore the pipeline, complete the following steps:



1. Log into Jenkins and create a pipeline by clicking New Item and then Pipeline.
2. Copy the pipeline from the Jenkinsfile [here](#).
3. Paste the pipeline into the Jenkins pipeline section and then click Save.
4. Fill the parameters of the Jenkins pipeline with the respective details like the current Magento application version in production, the Astra Control Center FQDN, the API token, the instance ID and application name or namespace of production and debug environments, and the source and destination cluster names. For the purpose of this use case, the production environment is a namespace called 'magento-prod' and the debug environment is a namespace called 'magento-debug' configured on a Red Hat OpenShift cluster.

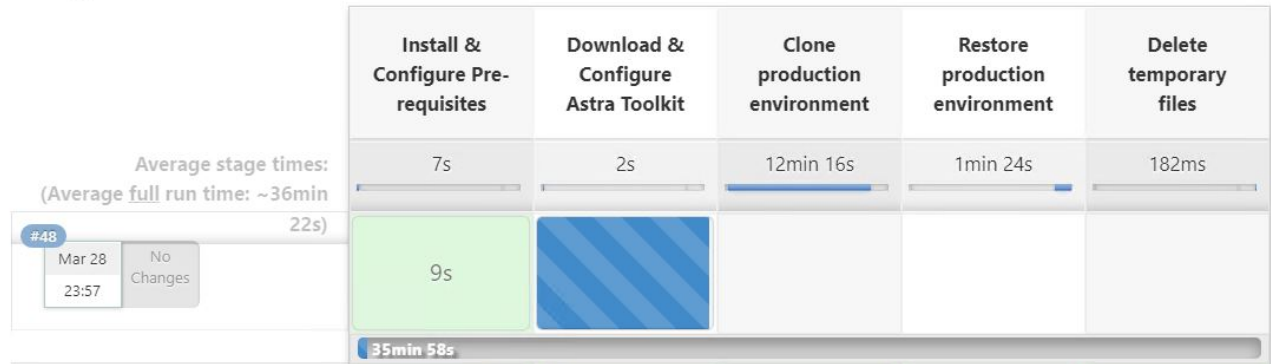
```
MAGENTO_VERSION = '2.4.1-debian-10-r14'  
ASTRA_TOOLKIT_VERSION = '2.0.2'  
ASTRA_API_TOKEN = 'xxxxxx'  
ASTRA_INSTANCE_ID = 'xxx-xxx-xxx-xxx-xxx'  
ASTRA_FQDN = 'netapp-astra-control-center.org.example.com'  
PROD_APP_NAME = 'magento-prod'  
DEBUG_APP_NAME = 'magento-debug'  
DEBUG_NAMESPACE = 'magento-debug'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

5. Click Build Now. The pipeline starts executing and progresses through the steps. The application is first cloned in the current state to a debug environment, and the application is then restored to the known-working backup.

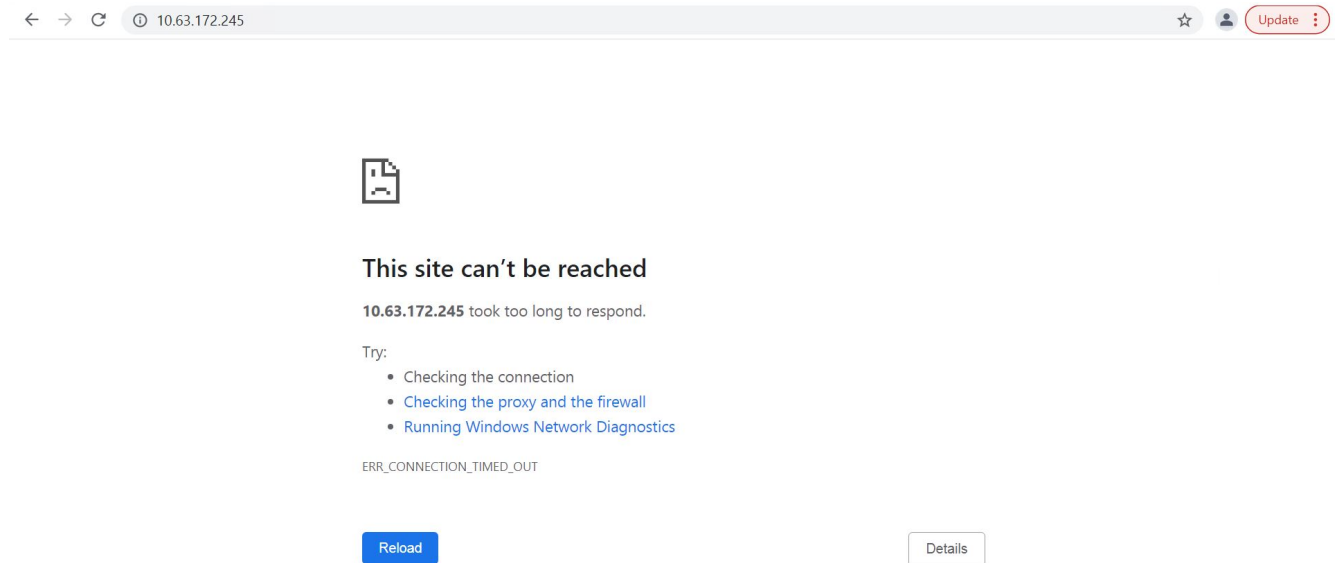
Pipeline magento_clone-for-triage_restore-from-backup



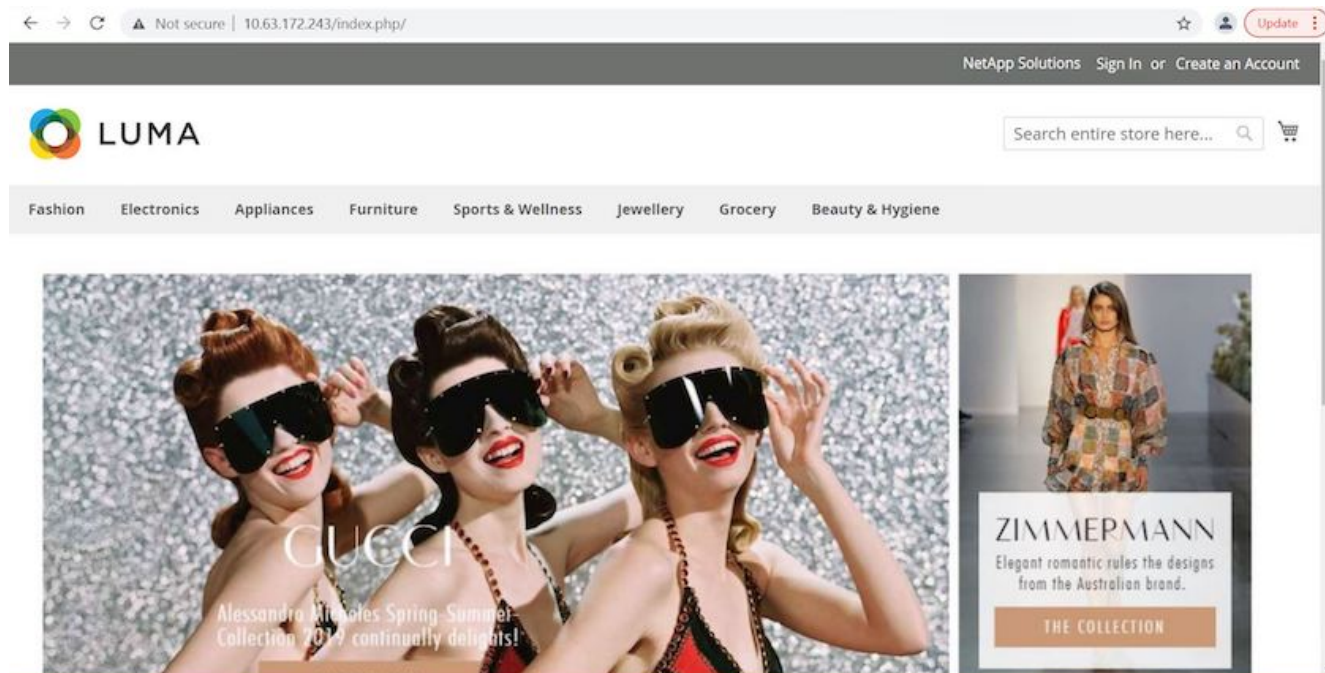
Stage View



6. Verify that the cloned application is the bug-containing version.



7. Verify that the production environment is restored to a working backup, and the application in production works as expected.



These two operations in tandem expedite the return to normal business operations. To see this use case in action, watch the video below.

[Leverage NetApp Astra Control to Perform Post-mortem Analysis and Restore Your Application](#)

Accelerating software development with NetApp FlexClone technology

This section outlines how to use NetApp FlexClone technology to rapidly deploy the solution.

Overview

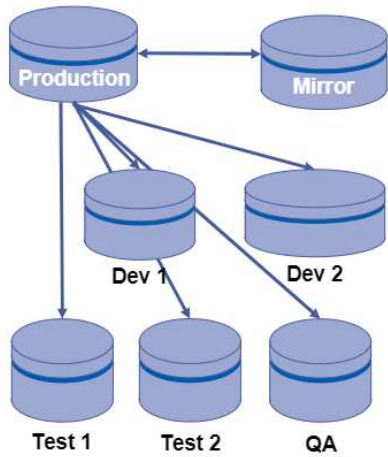
Cloning a deployed application in a Kubernetes cluster is a very useful tool for developers that would like to expedite their workflows by sharing environments with partners or by testing new versions of code in a development environment without interfering with the version they are currently working on. The stateful and application-consistent cloning of a Kubernetes application is a major feature included with NetApp Astra Control, alongside the backup and restore of applications. As a bonus, if an application is cloned within the same Kubernetes cluster using the same storage backend, Astra Control defaults to using NetApp FlexClone technology for the duplication of persistent data volumes, speeding up the process significantly. By accelerating this process, the cloned environment is provisioned and available for use in a few moments, allowing developers to resume their work with just a brief pause when compared to redeploying their test or development environment. As an additional convenience, all of the functions available in NetApp Astra Control can be called with an API, which allows for easy integration into automation frameworks like Ansible. Therefore, environments can be staged even more rapidly because only minor changes are needed in a playbook or role to begin the cloning procedure.

What is NetApp FlexClone technology?

NetApp FlexClone technology is a writeable, point-in-time snapshot-based copy of a NetApp FlexVol. They are provisioned almost instantly, contain all of the data from the source volume, and consume no additional storage space until the data in the new volume begins to diverge from the source. They are often used in development or template-based environments when multiple copies of data are useful for staging purposes and storage systems have limited resources for provisioning these volumes. Compared to a traditional storage system in

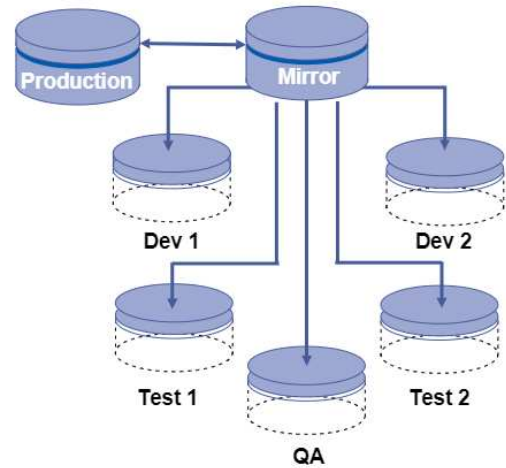
which data must be copied multiple times resulting in the consumption of significant storage space and time, NetApp FlexClone technology accelerates storage-dependant tasks.

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

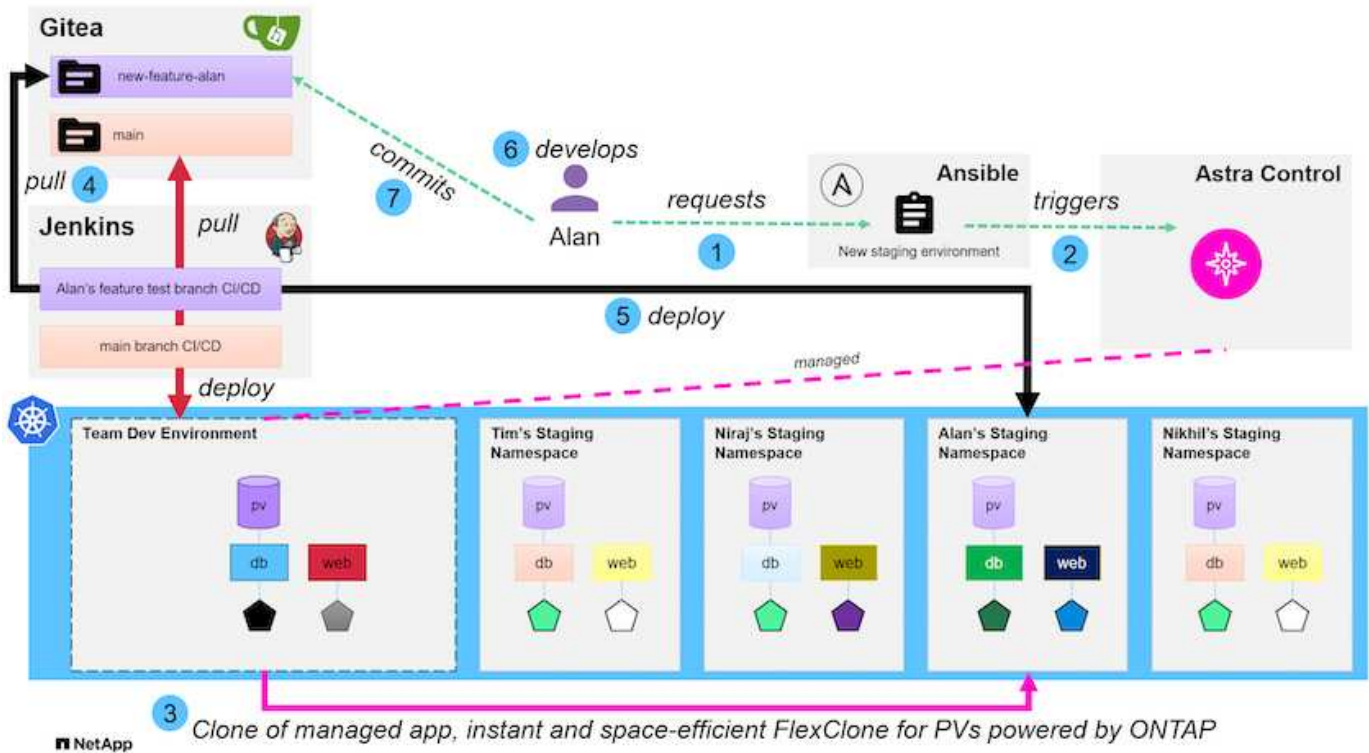
To find out more about NetApp FlexClone technology, visit the page on [NetApp Docs](#).

Prerequisites

1. A supported Kubernetes Distribution, such as Red Hat OpenShift 4.6.8+, Rancher 2.5+, or Kubernetes 1.19+.
2. NetApp Astra Control Center 21.12+.
3. A NetApp ONTAP system with a storage backend configured through NetApp Astra Trident.
4. Ansible 2.9+.
5. Templates for the environments that you'd like to clone as managed applications in NetApp Astra Control.

Use-case introduction

For this use case, we visualize something similar to the following workflow:



1. A user runs the ansible playbook to create a new staging environment.
2. Ansible uses the URI-API module to call out to Astra Control to execute the cloning operation.
3. Astra Control executes a cloning operation on a preprovisioned template environment, thus creating a new managed application.



This environment can be a single standalone application in development or an entire development environment like a Jenkins CI/CD pipeline.

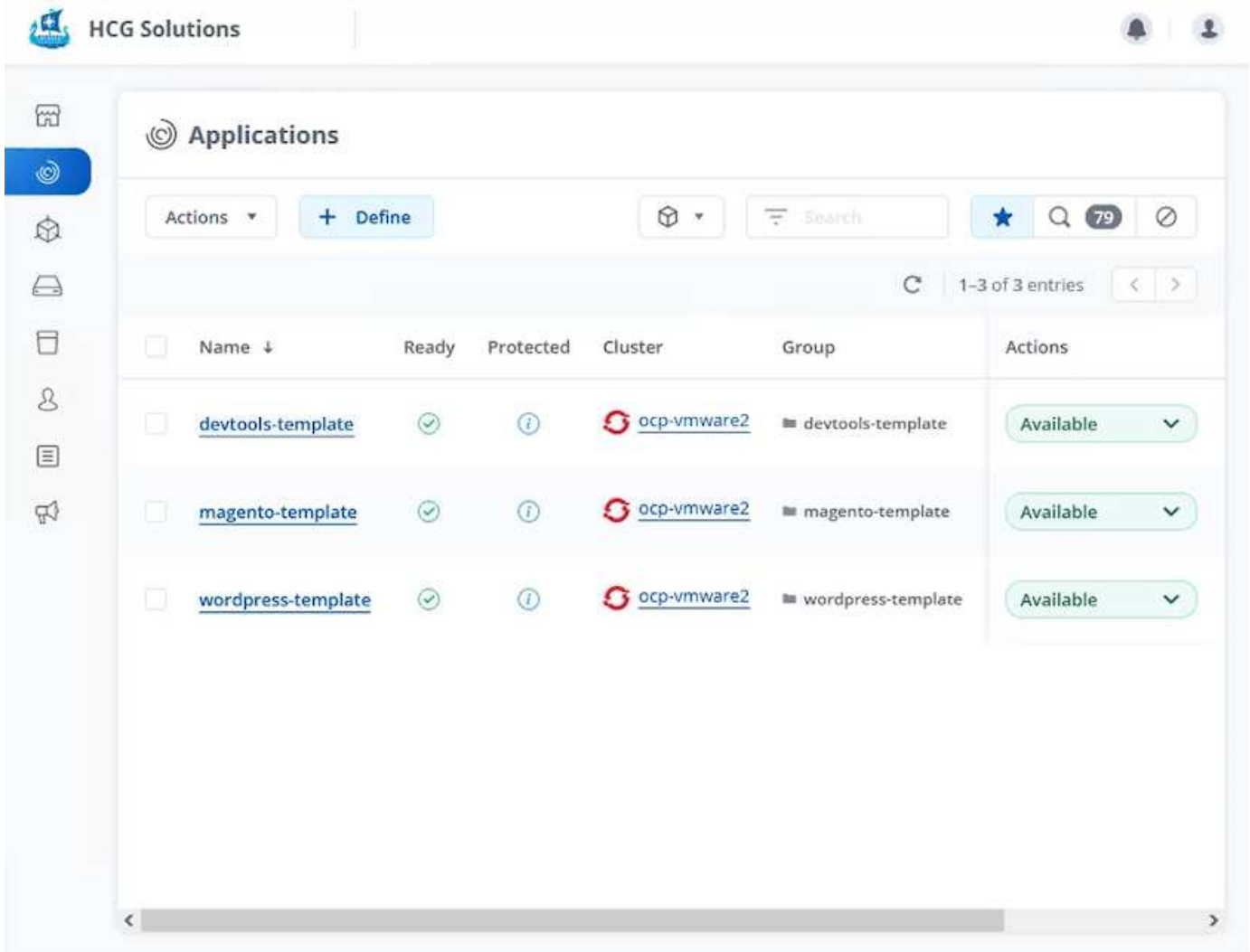
4. The user then pulls a version of their code into the cloned dev environment from an online repository like Gitea.
5. The new version of the application is deployed and managed by NetApp Astra Control.



Both of these processes can be automated.

6. The user can develop new code in this cloned environment.
7. When the user is satisfied with their development efforts, they can push the code back to the hosted repository.

The use case presented here depends on the existence of golden templates for the particular environments or applications you would like to clone. In our environment we have created three such templates, one for a Wordpress deployment, one for a Magento deployment, and one for a Jenkins CI/CD environment with Gitea that we have titled DevTools.



Each of these environments is managed by NetApp Astra control, with persistent volumes currently stored on a NetApp ONTAP storage system with an NFS backend provided by NetApp Astra Trident.

Use-case validation

1. Clone the ansible toolkit provided by the NetApp Solutions Engineering team, which includes the cloning role and the application update playbook.

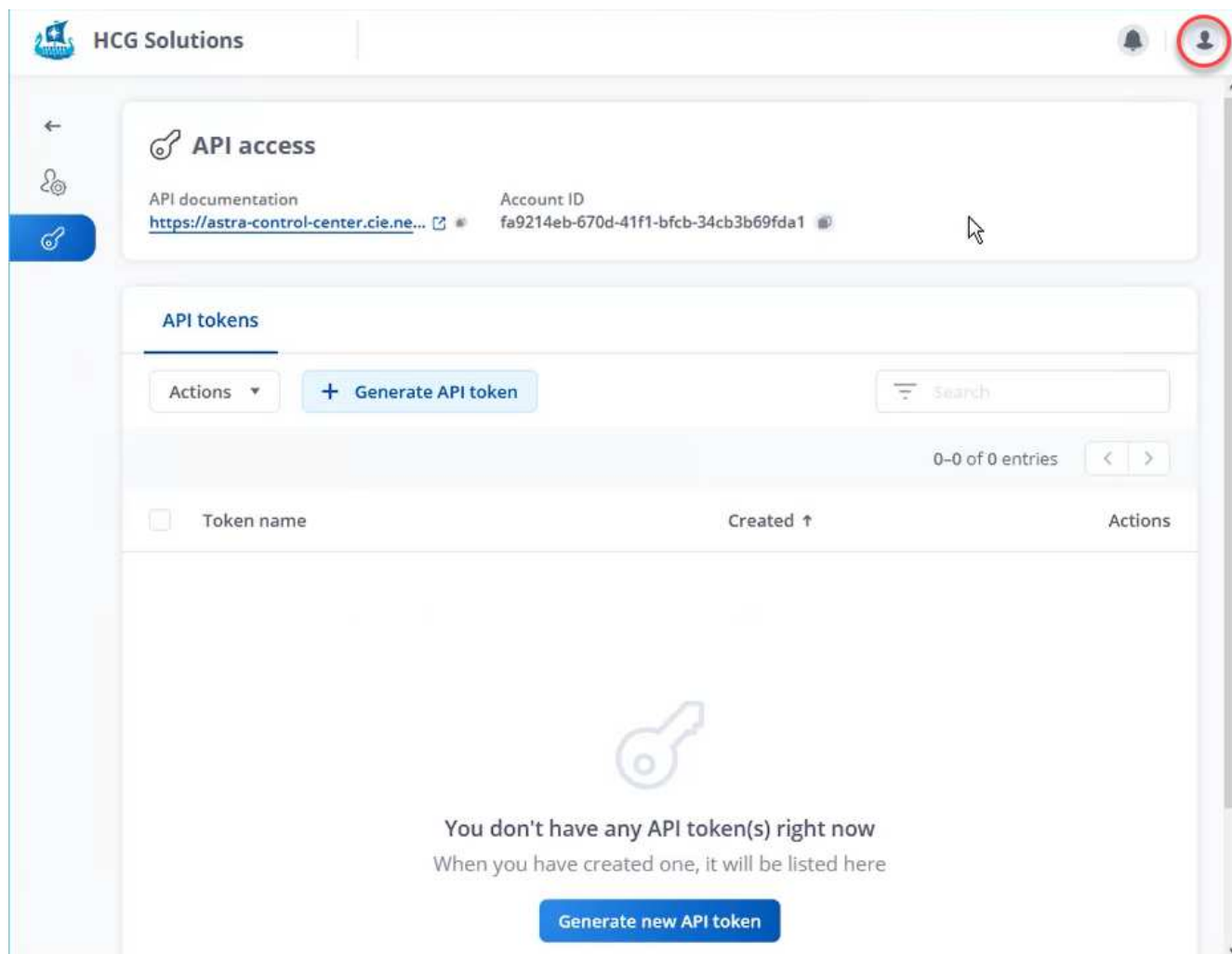
```
[netapp-user@rhel7 ~]$ git clone https://github.com/NetApp-Automation/na_astra_control_suite.git
[netapp-user@rhel7 ~]$ cd na_astra_control_suite
```

2. Edit `vars/clone_vars.yml` and fill in the global values that fit your Astra Control environment.

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



The global environment values you need to fill out are available under the user profile icon in NetApp Astra Control under the API Access menu.



3. With the global variables completed, you can choose the values for the specific application you wish to clone. To clone the devtools environment to a personal environment called `alan-devtools`, you would do the following:

```
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
```



To take advantage of NetApp FlexClone technology in the cloning process, `src-cluster` and `dest-cluster` must be the same.

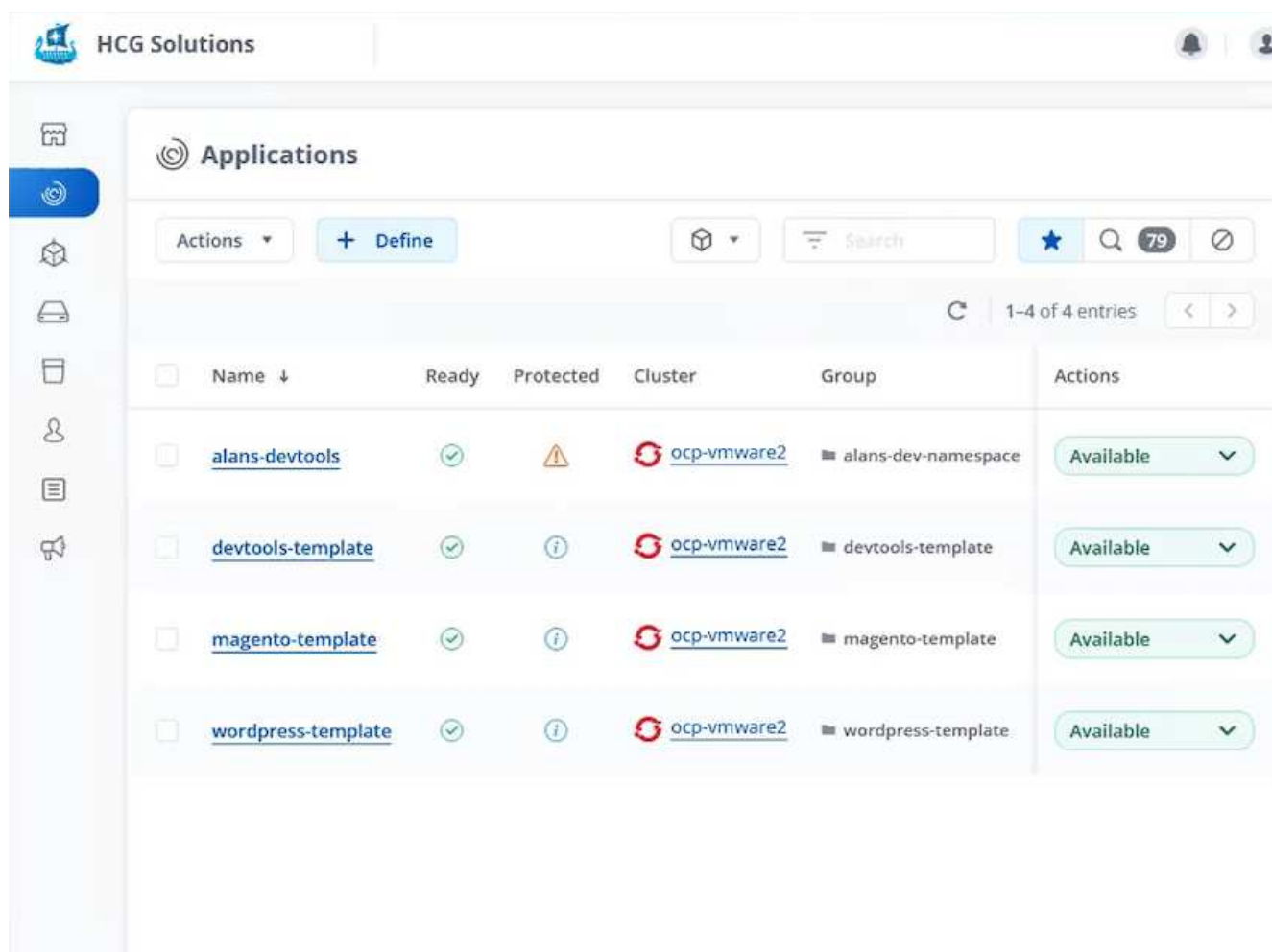
4. You can now execute the playbook to clone the application.

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



The playbook as written must be run by the root user or someone that can escalate through the sudo process by passing the "-K" argument.

- When the playbook completes its run, the cloned application shows as available in the Astra Control Center console.



- A user can then log into the Kubernetes environment where the application was deployed, verify that the application is exposed with a new IP address, and start their development work.

For a demonstration of this use case and an example of upgrading an application, watch the video below.

[Accelerate Software Development with Astra Control and NetApp FlexClone Technology](#)

Videos and demos: DevOps with NetApp Astra

The following videos demonstrate some of the capabilities described in this document:

[Data Protection in CI/CD pipeline with Astra Control Center](#)

Additional Information: DevOps with NetApp Astra

To learn more about the information described in this document, review the following websites:

- NetApp Documentation

<https://docs.netapp.com/>

- Astra Trident Documentation

<https://docs.netapp.com/us-en/trident/>

- NetApp Astra Control Center Documentation

<https://docs.netapp.com/us-en/astra-control-center/>

- Ansible Documentation

<https://docs.ansible.com/>

- Red Hat OpenShift Documentation

https://access.redhat.com/documentation/en-us/openshift_container_platform/4.8/

- Rancher Documentation

<https://rancher.com/docs/>

- Kubernetes Documentation

<https://kubernetes.io/docs/home/>

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.