ユースケースの検証

NetApp Solutions

NetApp
April 10, 2024

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions/containers/dwn_use_case_integrated_data_protection.html on April 10, 2024. Always check docs.netapp.com for the latest.

目次

ユースケースの検証：ネットアップAstraを使用したDevOps 1
NetApp Astra Controlにより、CI / CDパイプラインへの保護を統合 1
アストラ制御を使用すると、事後分析やアプリケーションのリストアが容易になります 8
NetApp FlexCloneテクノロジでソフトウェア開発を高速化 12

ユースケースの検証：ネットアップAstraを使用したDevOps

ネットアップAstraを使用してDevOps向けに検証されたユースケースは次のとおりです。

- "NetApp Astra Controlにより、CI / CDパイプラインへの保護を統合"
- "アストラ制御を活用して、事後分析とアプリケーションの復元を容易にします"
- "ネットアップのFlexCloneでソフトウェア開発を高速化"

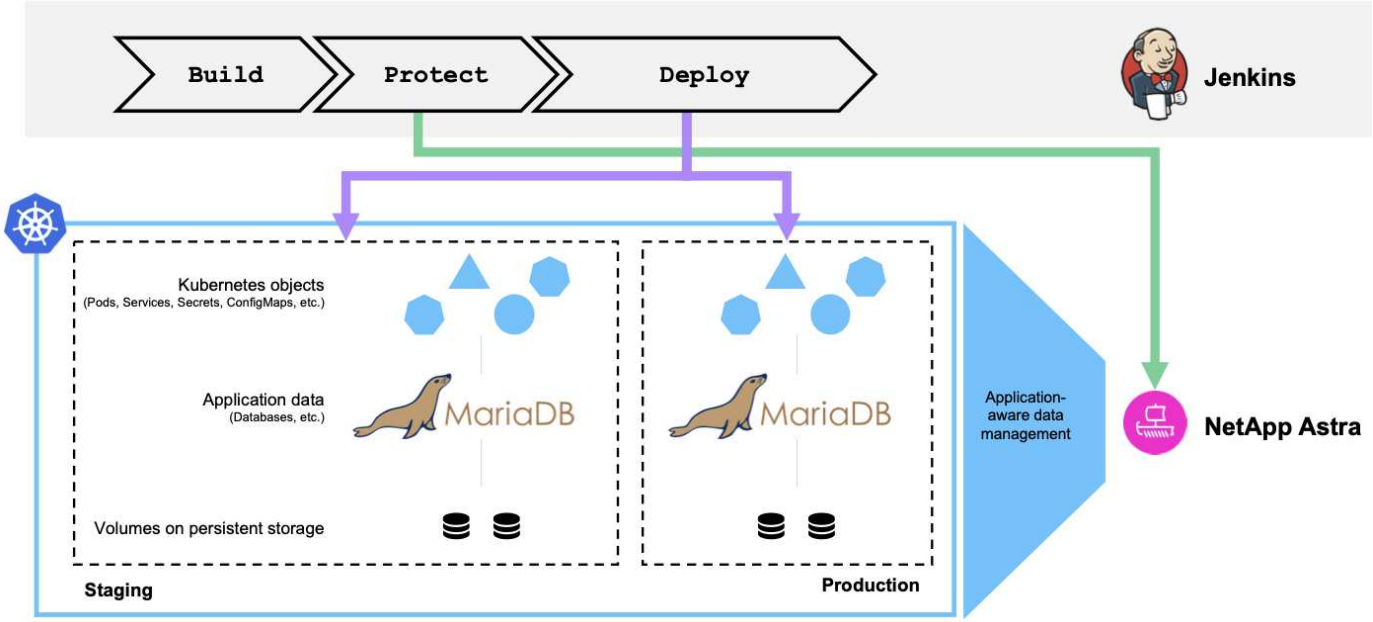
NetApp Astra Controlにより、CI / CDパイプラインへの保護を統合

概要

DevOpsワークフローの最も一般的な用途の1つは、継続的統合と継続的導入（CI / CD）パイプラインです。開発者が新しいコードをコミットする際に、アプリケーションでテストスイートの自動構築、統合、実行を行うことができます。通常、DevOpsエンジニアやサイト信頼性エンジニア（SRE）は、新機能の開発、回帰テスト、バグ修正、品質エンジニアリングなど、開発プロセスのさまざまな機能に特化したパイプラインを備えています。

チームの自動化レベルが上がると、本番環境のアプリケーションの変化のペースが圧倒的に高まります。そのため、一部のチームは、本番環境のアプリケーションやサービスの保護を希望しています。コードおよびコンテナイメージを保護するだけでなく、アプリケーションの状態、構成データ（アプリケーションに関連付けられたKubernetesオブジェクトやリソースなど）、およびアプリケーションの永続データも保護したいと考えています。

このユースケースでは、まず新しいバージョンのアプリケーションをステージング環境に導入してから本番環境に導入する、本番への昇格パイプラインについて詳しく見ていきます。この例は、主要なパブリッククラウドとオンプレミス環境に同様に当てはまります。アプリケーションの1つのバージョンの展開を示していますが、パイプラインは、青/緑、カナリアデプロイメントなどの他の戦略でも使用できます。CI/CDパイプラインの一部として、アプリケーションの完全なバックアップを作成してアプリケーションを保護します。本番環境のアプリケーションと、そのデータ、状態、設定をアプリケーションに対応したバックアップでバックアップすると、多数のDevOpsワークフローに役立ちます。



このユースケースの検証に使用したアプリケーションはです "Magento"は、Webベースのフロントエンドを備えたEコマース解決策 、検索および分析機能用のElasticsearchインスタンス、ショッピングインベントリとトランザクションの詳細をすべて追跡するMariaDBデータベースです。このコンテナ化されたアプリケーションは、Red Hat OpenShiftクラスタにインストールされました。アプリケーション内のすべてのポッドで、永続ボリュームを使用してデータを格納しました。永続ボリュームは、コンテナストレージインターフェイスに準拠したKubernetes向けストレージオーケストレーションツールであるNetApp Astra Tridentによって自動的に作成され、ネットアップストレージシステムでストレージをプロビジョニングできるようになりました。さらに、Astra Control Centerのアプリケーション保護機能を利用するために、問題のアプリケーションはAstra Controlによって管理されました。Astra Controlは、永続的ボリュームに保存されたデータとともに、アプリケーションの状態を保存したアプリケーションのバックアップをトリガするために使用されました。使用しました "NetApp Astra Control Python SDK" アプリケーションのバックアップを開始するプロセスを自動化するために、CI / CDパイプラインに導入されました。このパイプラインは、という一般的なCI / CDツールを使用して作成、実行されました ["Jenkins"]を選択すると、アプリケーションの構築、保護、導入のためのフローが自動化されます。

CI/CDパイプラインに保護機能を導入するための前提条件と手順 について説明します。

ユースケースの検証の前提条件

前提条件として、次のツールまたはプラットフォームを導入および設定しました。

1. Red Hat OpenShift Container Platform
2. ネットアップのOpenShiftにTridentをインストールし、バックエンドからNetApp ONTAP システムを構成
3. NetApp ONTAP バックエンドをポイントするデフォルトのストレージクラスが設定されている
4. OpenShiftクラスタにNetApp Astraコントロールセンターをインストール
5. OpenShiftクラスタをAstra Control Centerにマネージドクラスタとして追加
6. JenkinsをOpenShiftクラスタにインストールし、Dockerエンジンがインストールされたエージェントノードで構成します

アプリケーションをインストールしています

まず、ステージング環境と本番環境にアプリケーションを最初にインストールします。このユースケースでは、この手順が前提条件となるため、手動で実行します。CI / CDパイプラインは、アプリケーションの新しいバージョンリリースの結果として、以降のビルドおよび導入ワークフローに使用されます。

この使用事例の本番環境は「Magent-prod」という名前のネームスペースであり、対応するステージング環境はRed Hat OpenShiftクラスタ上で構成される「Magent-staging」という名前のネームスペースです。アプリケーションをインポートするには、次の手順を実行します。

1. bitnami helmチャートを使用して、生産環境にMagentoアプリケーションをインストールします。ジェントポッドとMariaDBポッドにRWX PVSを使用します。

```
[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チャートには、Magento GUIサービスを公開するためのロードバランササービスが必要です。使用しました "MetalLB" この例では、オンプレミスのロードバランササービスを提供します。

2. 数分後に、すべてのポッドとサービスが実行されていることを確認します。

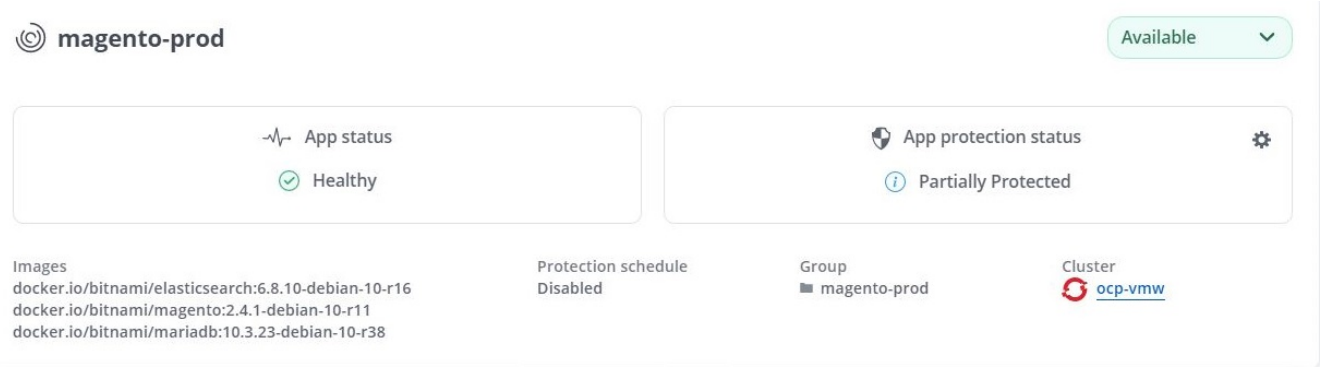
```
[netapp-user@rhel7 na_astra_control_suite]$ oc get pods -n magento-prod
NAME                                                    READY   STATUS
RESTARTS   AGE
magento-9d658fd96-qrxmt                                 1/1     Running
0          49m
magento-elasticsearch-coordinating-only-69869cc5-768rm  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. ステージング環境についても同じ手順 を繰り返します。

Astra Control CenterでMagentoアプリケーションを管理します

1. [アプリケーション]に移動し、[検出されたアプリケーション]タブを選択します。

2. 本番環境でMagentoアプリケーション(「Magent-prod`」)の省略記号をクリックし、「管理」をクリックします。
3. Magentoアプリケーションは、Astra Control Centerによって管理されるようになりました。Astra Controlでサポートされているすべての操作をアプリケーションで実行できます。アプリケーションのバージョンも記録します。

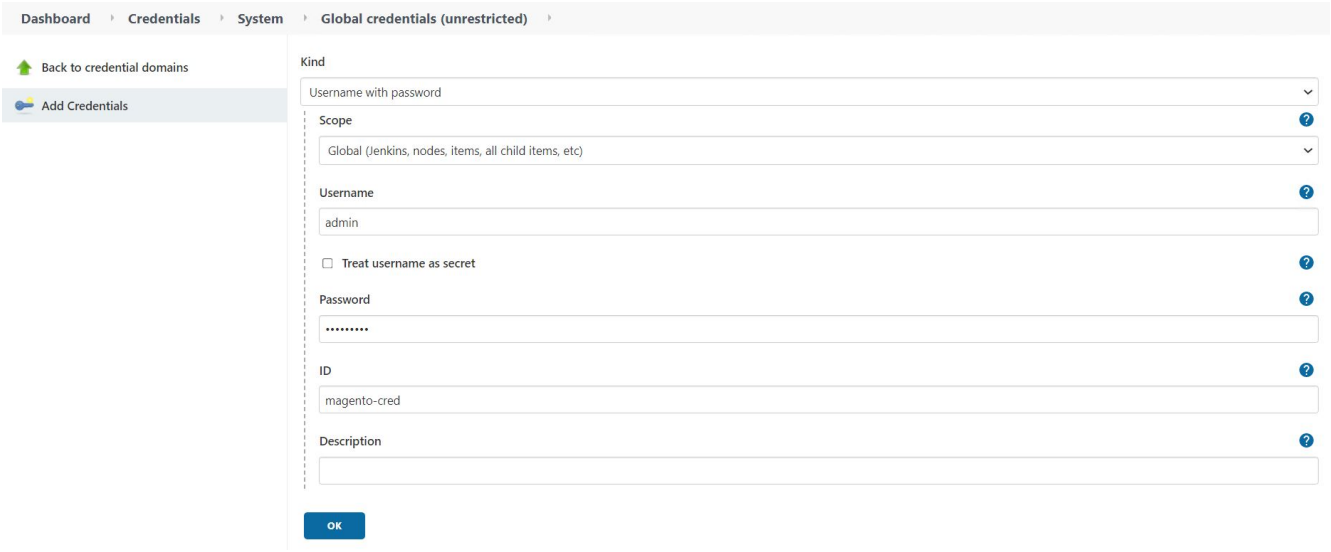


4. ステージング環境でMagentoアプリケーションを管理するための手順を繰り返します(「Magent-staging」)。

保護機能を統合した**CI / CD**パイプライン

新しいバージョンのアプリケーションを使用する場合は、CI / CDパイプラインを使用してコンテナイメージを作成し、ステージング環境と本番環境の両方のバックアップを作成し、新しいバージョンのアプリケーションをステージング環境に導入し、本番環境への昇格の承認を待ちます。 その後、新しいバージョンのアプリケーションを本番環境に導入します。CI / CDパイプラインを使用するには、次の手順を実行します。

1. Jenkinsにログインし、必要なクレデンシャルを作成します。Magento作成する場合は1つ、MariaDB管理作成する場合は1つ、MariaDBルート作成する場合は3つ目です。
2. Manage Jenkins > Manage Credentialsと進み、適切なドメインをクリックします。
3. [資格情報の追加]をクリックし、[種類]を[パスワードと有効範囲を使用するユーザー名]に設定して[グローバル]に設定します。ユーザー名、パスワード、および資格情報のIDを入力し、[OK]をクリックします。

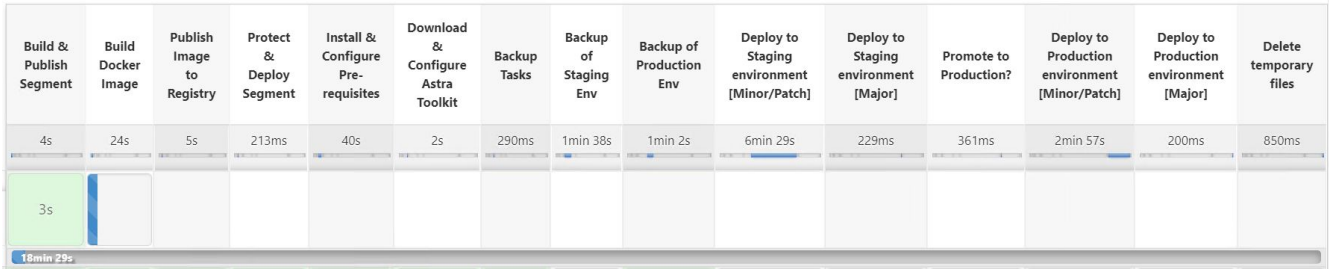


4. 他の2つのクレデンシャルについても同じ手順 を繰り返します。

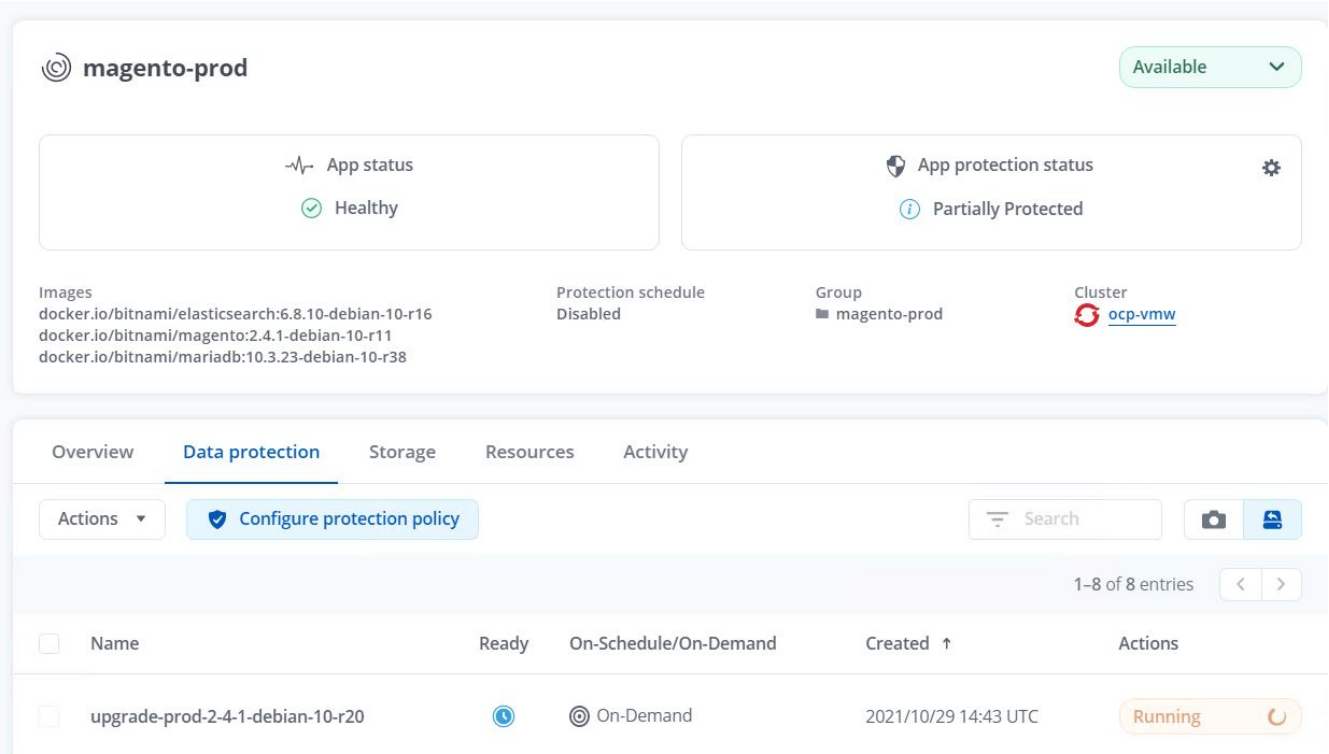
5. ダッシュボードに戻り、[新しいアイテム]をクリックしてパイプラインを作成し、[パイプライン]をクリックします。
6. Jenkinsfileからパイプラインをコピーします "こちらをご覧ください"。
7. パイプラインをJenkinsパイプラインセクションに貼り付け、保存をクリックします。
8. Helmチャートバージョン、アップグレード先のMagentoアプリケーションバージョン、Astraツールキットバージョン、Astra Control Center FQDN、APIトークン、インスタンスIDなど、Jenkinsパイプラインのパラメータをそれぞれの詳細に入力します。本番環境とステージング環境の両方でDockerレジストリ、ネームスペース、MagentoのIPを指定し、作成したクレデンシャルのクレデンシャルIDも指定します。

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
CHART_VERSION = '14'
RELEASE_TYPE = 'MINOR'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxxxxx'
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')
```

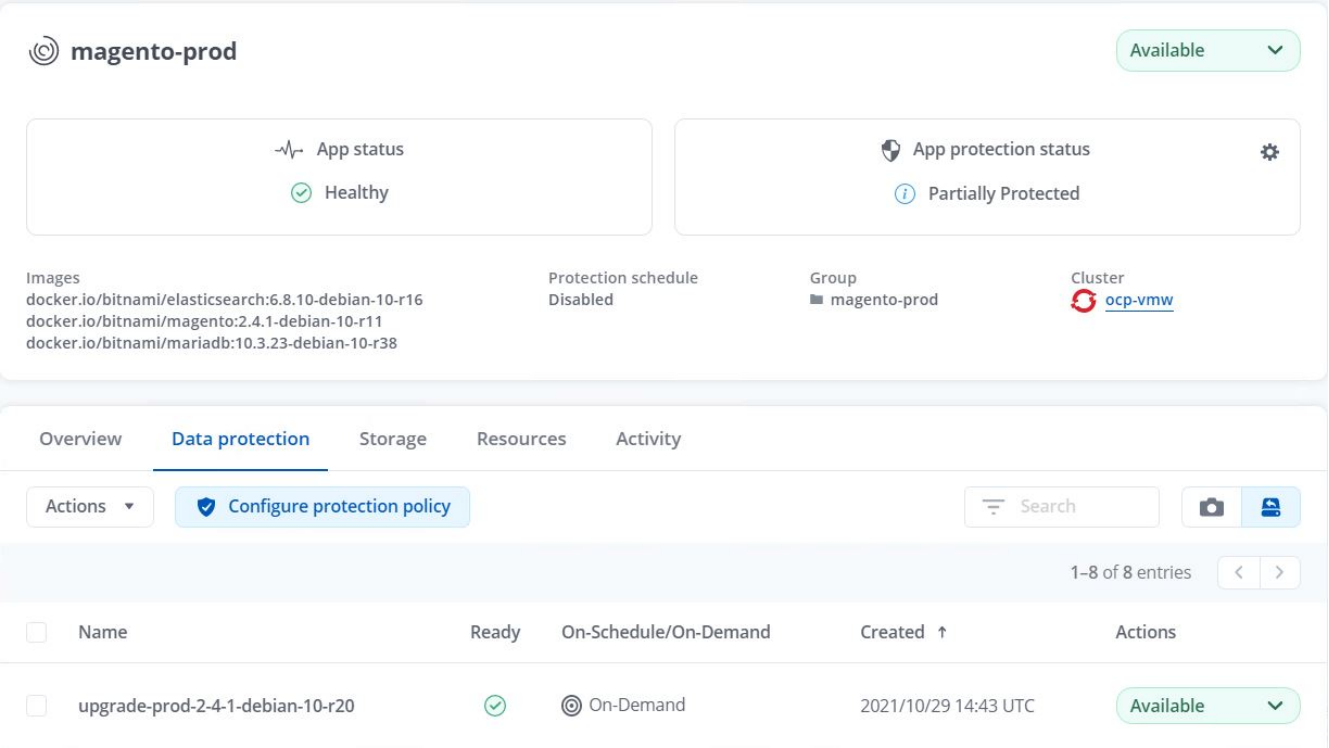
9. [今すぐ構築]をクリックしますパイプラインが実行を開始し'ステップを進めますアプリケーションイメージは最初にビルドされ、コンテナレジストリにアップロードされます。



10. アプリケーションのバックアップは、Astra Controlを使用して開始します。



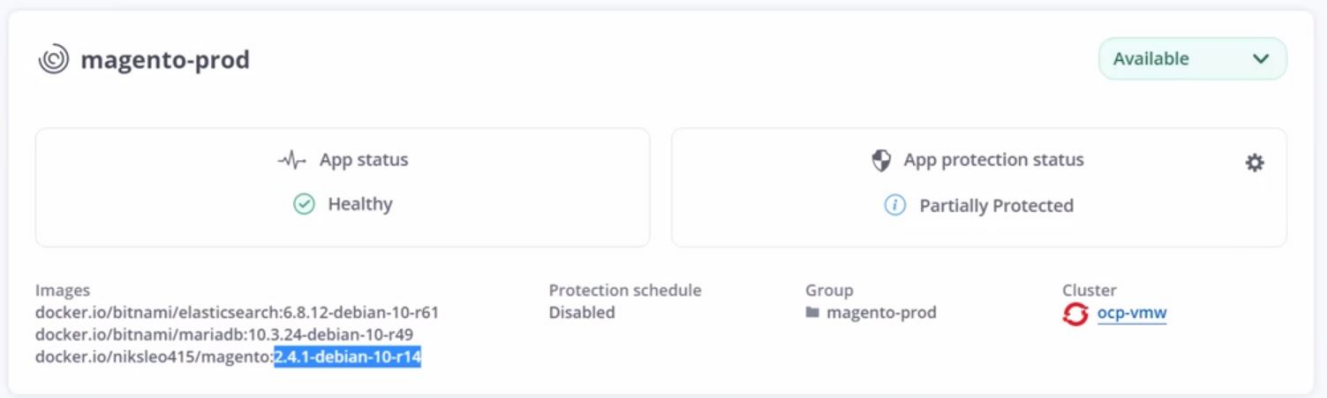
11. バックアップステージが正常に完了したら、Astra Control Centerからのバックアップを確認します。



12. 新しいバージョンのアプリケーションがステージング環境に展開されます。

13. この手順が完了すると、ユーザが本番環境への導入を承認するまで待機します。この段階では、QAチームがいくつかの手動テストを実行し、本番環境を承認すると仮定します。次に、[承認]をクリックして、新しいバージョンのアプリケーションを本番環境に展開できます。

14. 本番アプリケーションが目的のバージョンにアップグレードされていることも確認します。



CI / CDパイプラインの一環として、アプリケーションに対応した完全なバックアップを作成してアプリケーションを保護できることを実証しました。アプリケーション全体が本番への昇格パイプラインの一部としてバックアップされているため、高度に自動化されたアプリケーションの導入について、自信を持って実行できます。アプリケーションのデータ、状態、設定を含むこのアプリケーション対応バックアップは、多数のDevOpsワークフローに役立ちます。予期しない問題が発生した場合は、アプリケーションの前のバージョンにロールバックすることが重要なワークフローとなります。

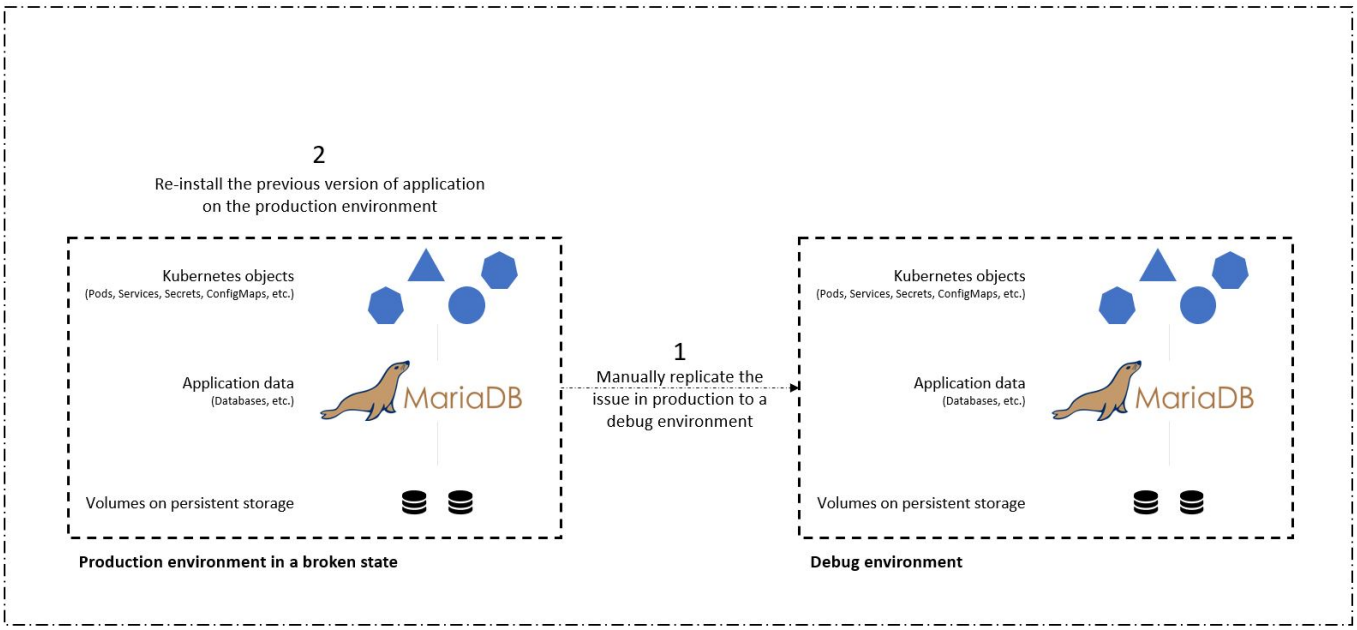
Jenkinsツールを使用してCI / CDワークフローをデモしましたが、コンセプトはさまざまなツールや戦略に簡単かつ効率的に外挿できます。このユースケースの実際の動作を確認するには、ビデオをご覧ください "こちらをご覧ください"。

アストラ制御を使用すると、事後分析やアプリケーションのリストアが容易になります

概要

を参照してください "最初のユースケース"では、ネットアップのAstra Control Centerを使用してKubernetesでアプリケーションを保護する方法を紹介しました。このセクションでは、NetApp Astraツールキットのの Python SDKを使用して、Astra Controlを介したアプリケーションバックアップを開発ワークフローに直接統合する方法について説明します。このアプローチにより、継続的統合/継続的導入（CI / CD）プロセスでオンデマンドバックアップを自動化することで、開発環境と本番環境の保護が可能になります。アプリケーションと整合性のあるデータ保護のこの追加レイヤーがCI / CDパイプラインと本番アプリケーションに追加されたことで、プロセスに何か問題が発生しても開発プロセスは安全になり、ビジネス継続性のプラクティスが促進されます。

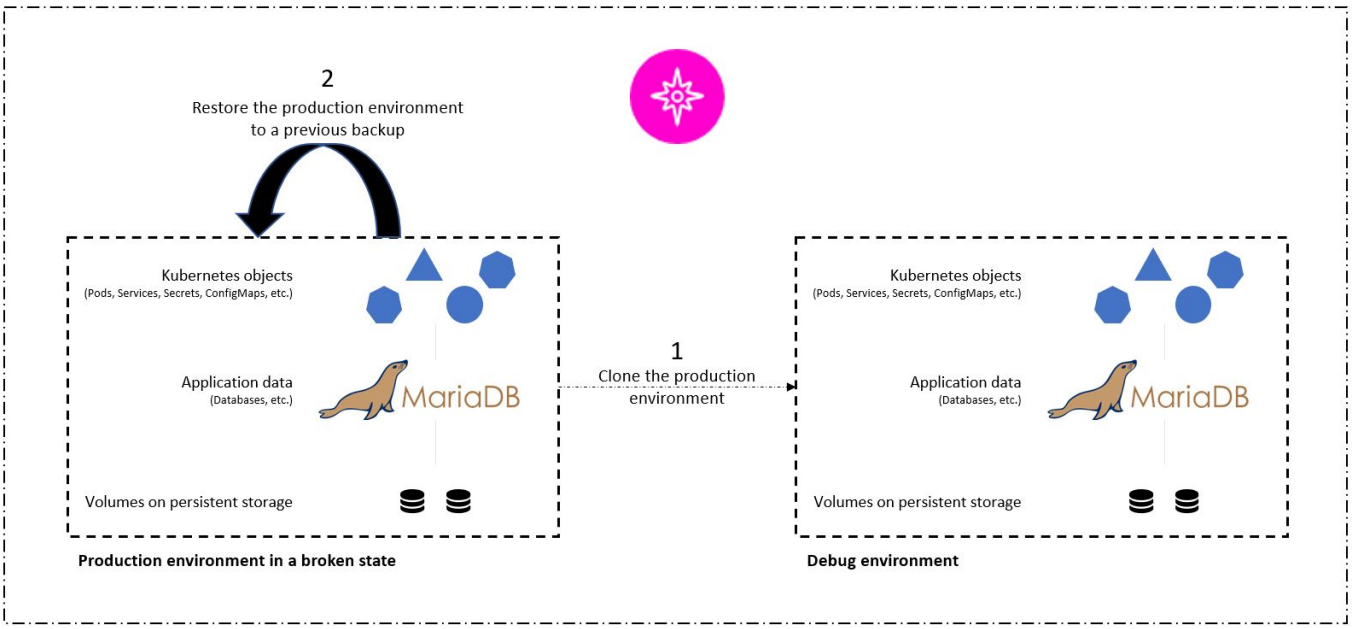
従来のワークフローでは、アプリケーションを新しいバージョンにアップグレードするときにエラーが発生した場合、開発チームはお客様から提供されたバグレポートに基づいて問題 のトラブルシューティングをリアルタイムで試みます。また、最初に問題が発生したときに、チームはアプリケーションを並列デバッグ環境に再配置して、そのプロセスをオフラインにすることもできます。以前のバージョンの古いコードベースを本番環境に再導入することで、アプリケーションを作業順序に復元できます。



このアプローチは機能しますが、問題が発生したときに本番環境で使用されていたバージョンと破損した本番アプリケーションの状態が一致していることを確認する必要があります。また、リポジトリからコードを取得し、マシンイメージを再展開してアプリケーションを良好な稼働状態に復元することで、動作確認済みビルドを本番環境にプロモートするための時間も必要になります。また、このシナリオでは、本番環境のデータベース自体が、障害のあるコードによって破損していないかどうかを考慮しませんでした。データベースデータのバックアッププロセスが別に用意されているのは理想的ですが、公開時にアプリケーションの状態と整合性があると仮定する必要があります。ここでは、Astra Controlを使用して、ステートフルでアプリケーションと整合性のあるバックアップ、リストア、クローンを作成することのメリットが、まさにその価値を示しています。

まず、アストラ制御を使用して、アプリケーションの状態に関する事後分析を容易に行うことができます。そのためには、アプリケーションと整合性のある方法で、バグのある本番バージョンを並行テスト環境にクローニングします。この環境をバグ非表示状態にしておくと、リアルタイムで問題のトラブルシューティングを行うことができます。

さらに、Astra Controlでは、インプレースリストア機能がサポートされているため、本番アプリケーションを最後の許容可能なバックアップ（コードのバージョンが古い場合）にリストアできます。復元されたバージョンは、以前に割り当てられた入力IPを含む、アプリケーションと一貫性のあるステートフルな方法で、以前のバグのある本番アプリケーションの位置を想定しています。そのため、フロントエンドにアクセスするお客様は、バックアップバージョンへの移行を認識しません。



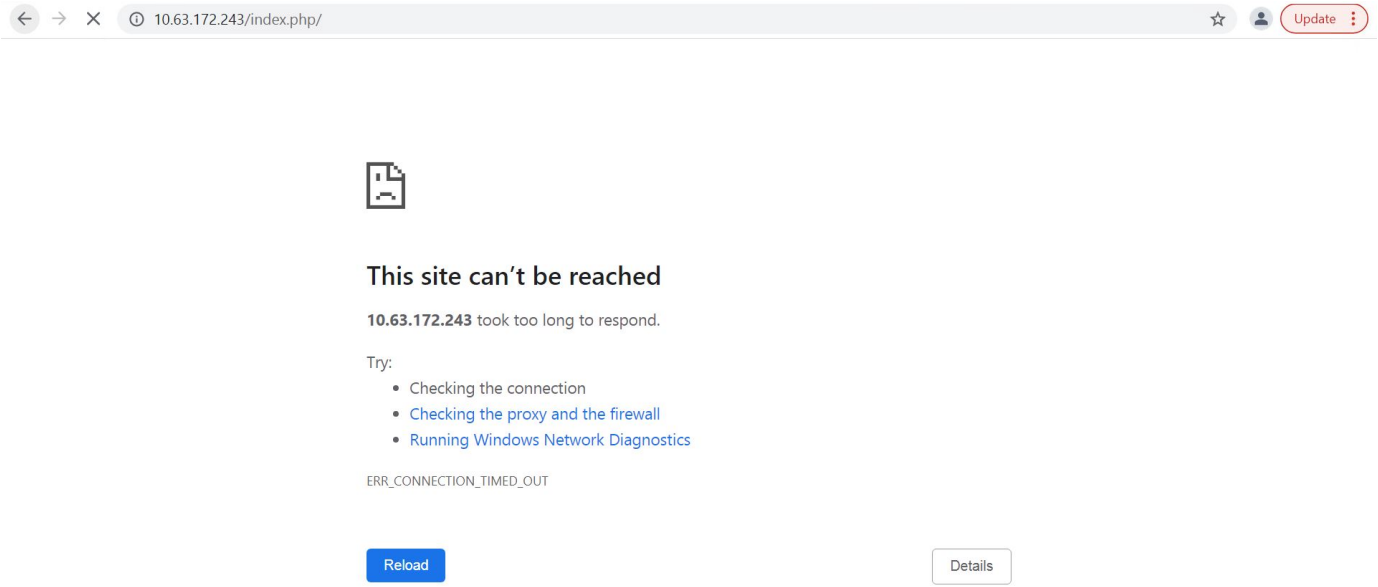
ユースケースの検証の前提条件

前提条件として、次のツールまたはプラットフォームを導入および設定しました。

- Red Hat OpenShift Container Platform：
- NetApp ONTAP システムにバックエンドを設定して、OpenShiftにNetApp Astra Tridentをインストールします。
- NetApp ONTAP バックエンドをポイントするデフォルトのストレージクラスが設定されている
- OpenShiftクラスタにNetApp Astraコントロールセンターをインストール。
- OpenShiftクラスタをAstra Control Centerにマネージドクラスタとして追加。
- OpenShiftクラスタにJenkinsをインストールしました。
- 本番環境にインストールされたMagentoアプリケーション。このユースケースの本番環境は、Red Hat OpenShiftクラスタで「ジェント本番環境」という名前のネームスペースです。
- 本番アプリケーションはAstra Control Centerによって管理されます。
- Astra Controlでキャプチャされた、本番アプリケーションの正常なバックアップ。

クローニングとリストアのパイプライン

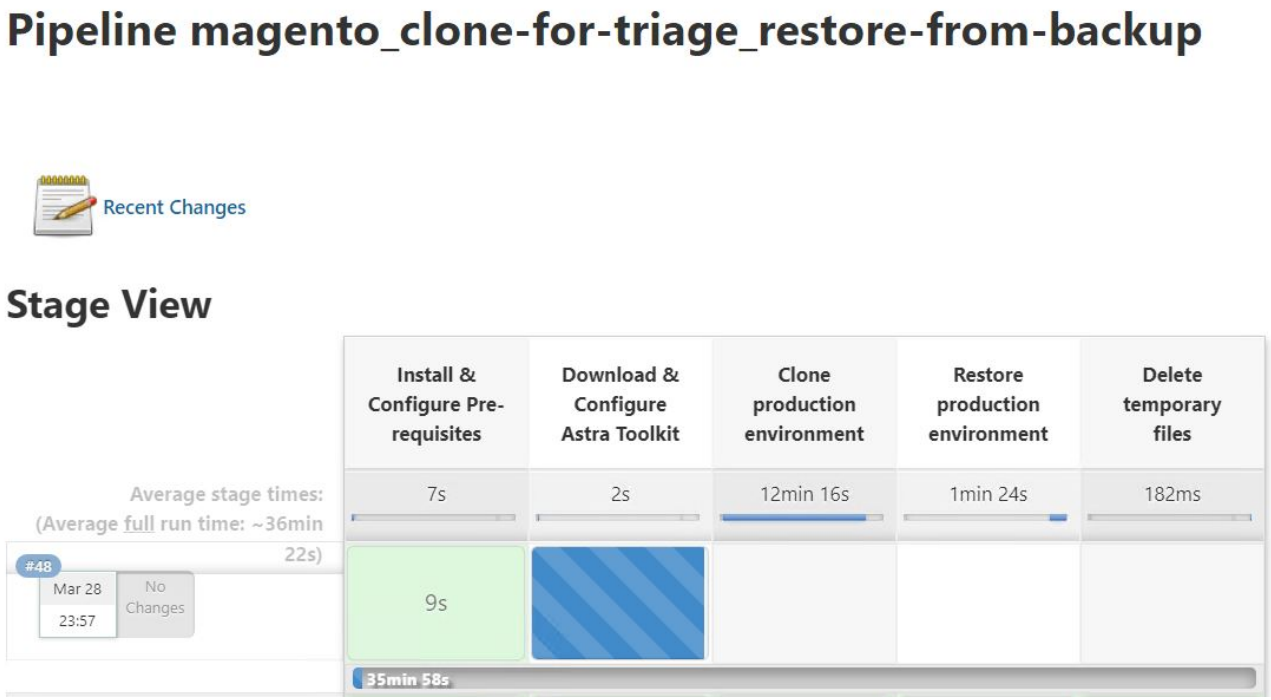
アプリケーションが新しいバージョンにアップグレードされたことを考慮して、本番環境のアプリケーション（「Magent-prod」）はアップグレード後に意図したとおりに動作しません。フロントエンドクエリから返されるデータがリクエストと一致しないか、データベースが実際に破損しているとします。パイプラインをクローニングしてリストアするには、次の手順を実行します。



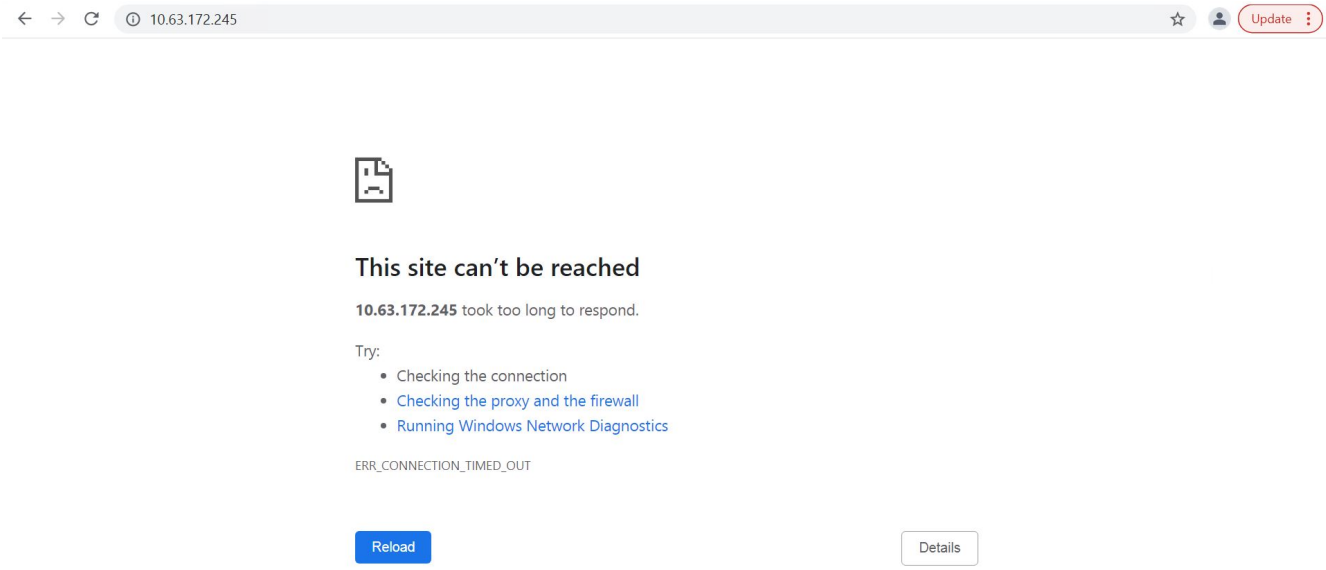
1. Jenkinsにログインし、[新しいアイテム]、[パイプライン]の順にクリックしてパイプラインを作成します。
2. Jenkinsfileからパイプラインをコピーします "こちらをご覧ください"。
3. パイプラインをJenkinsパイプラインセクションに貼り付け、保存をクリックします。
4. Jenkinsパイプラインのパラメータに、本番環境の現在のMagentoアプリケーションバージョン、Astra Control Center FQDN、APIトークン、本番環境とデバッグ環境のインスタンスIDとアプリケーション名またはネームスペース、ソースクラスタ名とデスティネーションクラスタ名など、それぞれの詳細を入力します。このユースケースのために、本番環境は「メント本番」と呼ばれるネームスペースであり、デバッグ環境はRed Hat OpenShiftクラスタで構成される「メントデバッグ」と呼ばれるネームスペースです。

```
MAGENTO_VERSION = '2.4.1-debian-10-r14'
ASTRA_TOOLKIT_VERSION = '2.0.2'
ASTRA_API_TOKEN = 'xxxxx'
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'
```

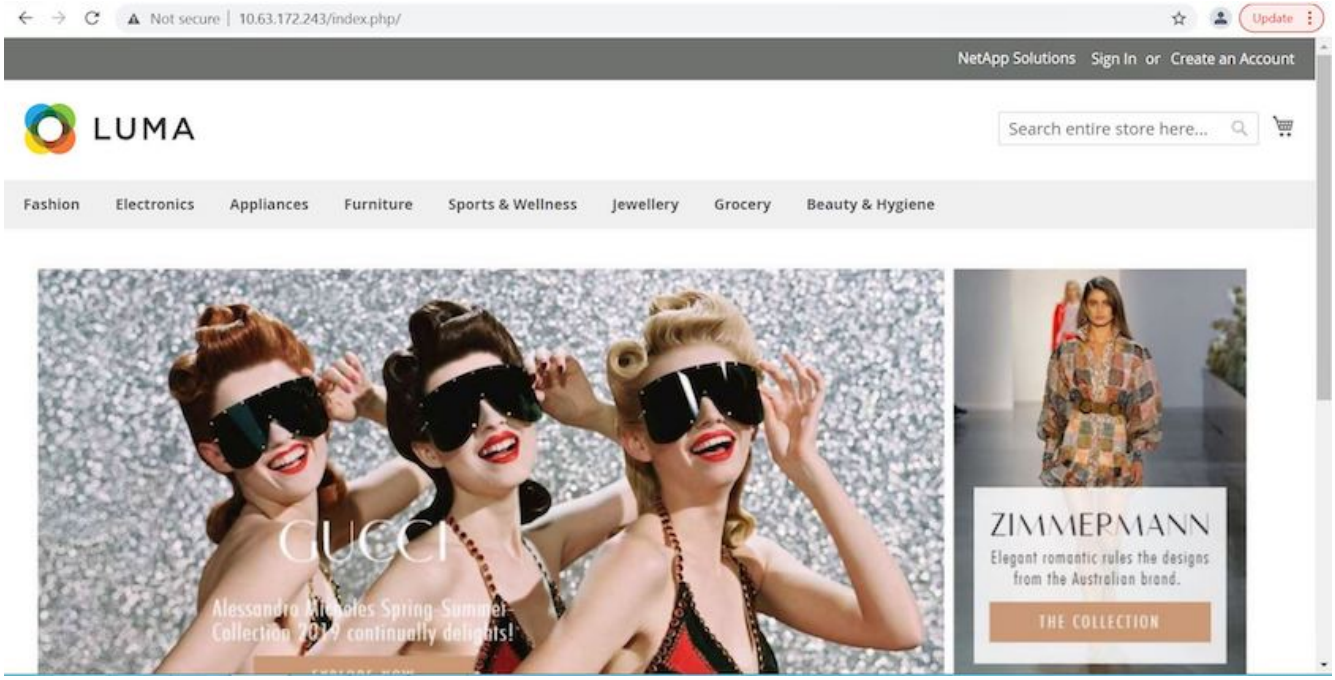
5. [今すぐ構築]をクリックしますパイプラインが実行を開始し’ステップを進めますアプリケーションは、最初に現在の状態でデバッグ環境にクローニングされ、その後、動作確認済みのバックアップにリストアされます。



6. クローニングしたアプリケーションのバージョンがバグを含むことを確認します。



7. 本番環境が稼働中のバックアップにリストアされ、本番環境のアプリケーションが想定どおりに動作することを確認します。



これら2つのオペレーションを同時に行うことで、通常の業務に迅速に復帰できます。このユースケースの実際の動作を確認するには、ビデオをご覧ください "こちらをご覧ください"。

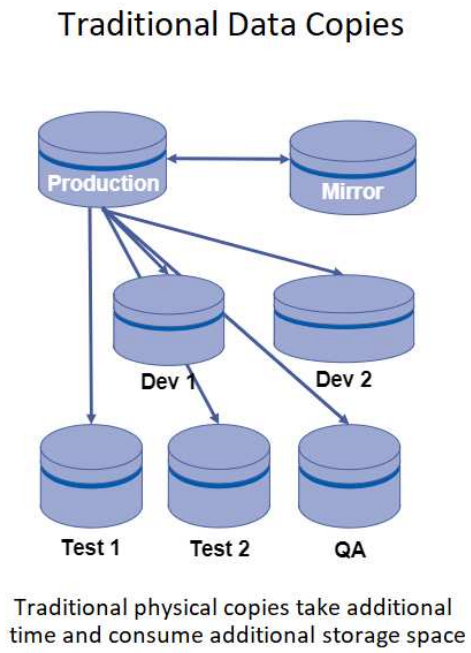
NetApp FlexCloneテクノロジでソフトウェア開発を高速化

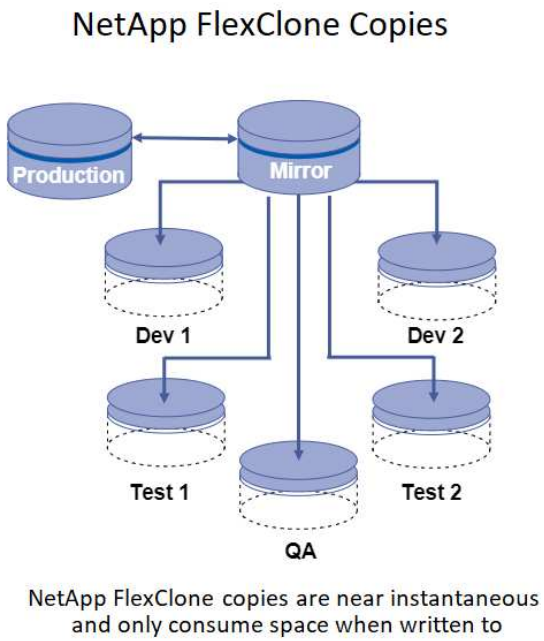
概要

Kubernetesクラスタに導入されたアプリケーションをクローニングすると、開発者は、パートナーと環境を共有したり、現在作業しているバージョンを妨げずに開発環境で新しいバージョンのコードをテストしたりすることで、ワークフローを迅速化したいと考えています。これは非常に便利なツールです。Kubernetesアプリケーションのステートフルでアプリケーションと整合性のあるクローニングは、NetApp Astra Controlに含まれる主要な機能であり、アプリケーションのバックアップとリストアに加えて提供されます。さらに、同じストレージバックエンドを使用して同じKubernetesクラスタ内でアプリケーションをクローニングした場合、Astra Controlでは、永続的データボリュームの重複に対してNetApp FlexCloneテクノロジがデフォルトで使用されるため、処理時間が大幅に短縮されます。このプロセスを短縮することで、クローニングした環境のプロビジョニングが完了し、数秒で使用できるようになります。そのため、開発者はテスト環境や開発環境を再導入する場合と比べて、作業を一時的に中断することなく再開できます。さらに、NetApp Astra Controlで使用できるすべての機能をAPIで呼び出すことができるため、Ansibleなどの自動フレームワークに簡単に統合できます。このため、クローニング手順 を開始するためにプレイブックまたはロールに必要な変更はわずかなため、環境のステージング時間をさらに短縮できます。

NetApp FlexCloneテクノロジとは

NetApp FlexCloneテクノロジは、NetApp FlexVol の書き込み可能なポイントインタイムSnapshotベースのコピーです。ほぼ瞬時にプロビジョニングされ、ソースボリュームのデータがすべて含まれます。新しいボリュームのデータがソースボリュームと異なる状態になるまで、追加のストレージスペースは消費されません。ステージング目的でデータの複数のコピーが有用で、ストレージシステムでボリュームのプロビジョニングに使用できるリソースが限られている場合、開発環境やテンプレートベースの環境でよく使用されます。従来のストレージシステムではデータを何度もコピーする必要があるため、ストレージスペースや時間が大幅に消費されるため、NetApp FlexCloneテクノロジを使用するとストレージに依存するタスクを高速化できます。





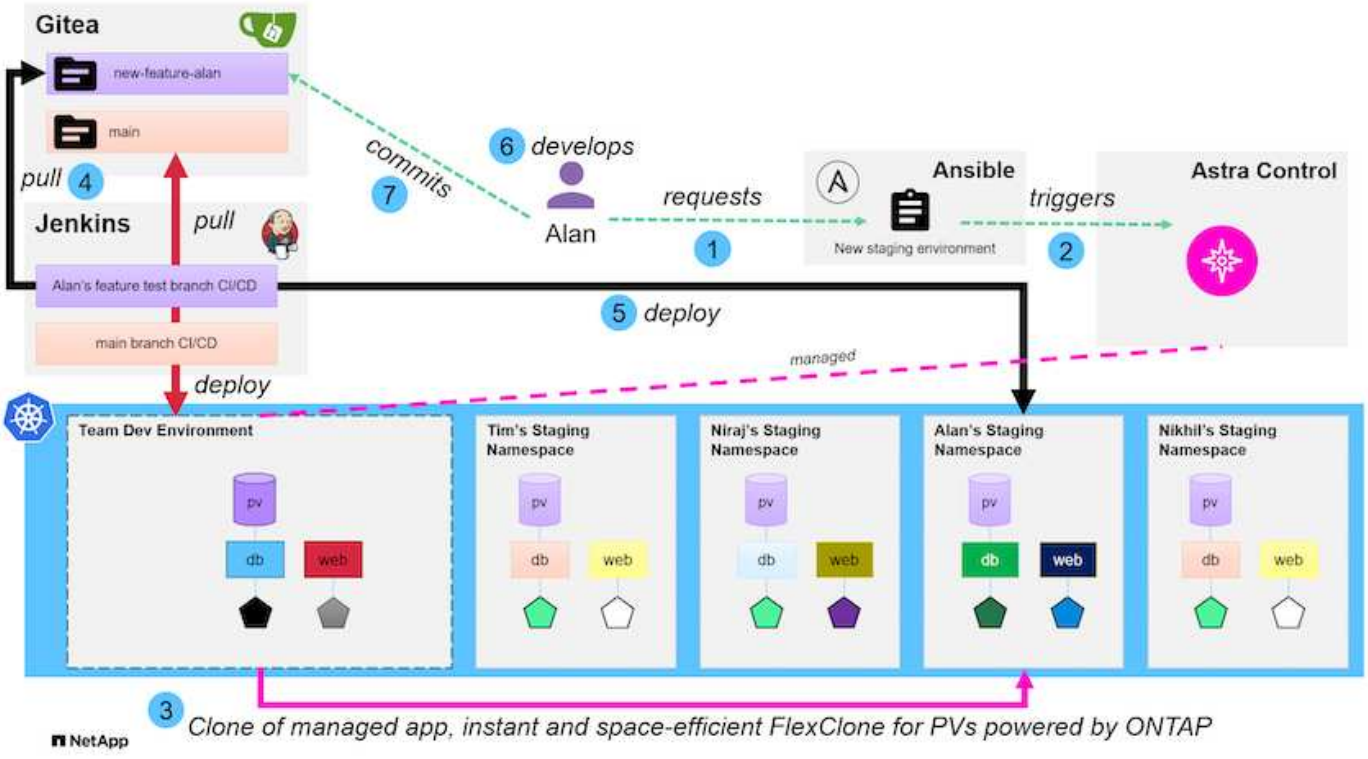
NetApp FlexCloneテクノロジの詳細については、のページを参照してください "ネットアップのドキュメント"。

前提条件

1. Red Hat OpenShift 4.6.8+、Rancher 2.5+、Kubernetes 1.19+などのサポートされるKubernetesディストリビューション。
2. ネットアップアストラコントロールセンター21.12以降
3. NetApp ONTAP システム。ネットアップのTrident経由でストレージバックエンドを設定します。
4. Ansible 2.9以降
5. NetApp Astra Controlの管理対象アプリケーションとしてクローニングする環境用のテンプレート。

ユースケースの概要

このユースケースでは、次のワークフローのような情報が表示されます。



1. ユーザがAnsibleプレイブックを実行して、新しいステージング環境を作成します。
2. Ansibleは、uri -APIモジュールを使用してAstra Controlに呼び出し、クローニング処理を実行します。
3. Astra Controlは、プロビジョニング済みのテンプレート環境でクローニング処理を実行して、新しい管理対象アプリケーションを作成する。

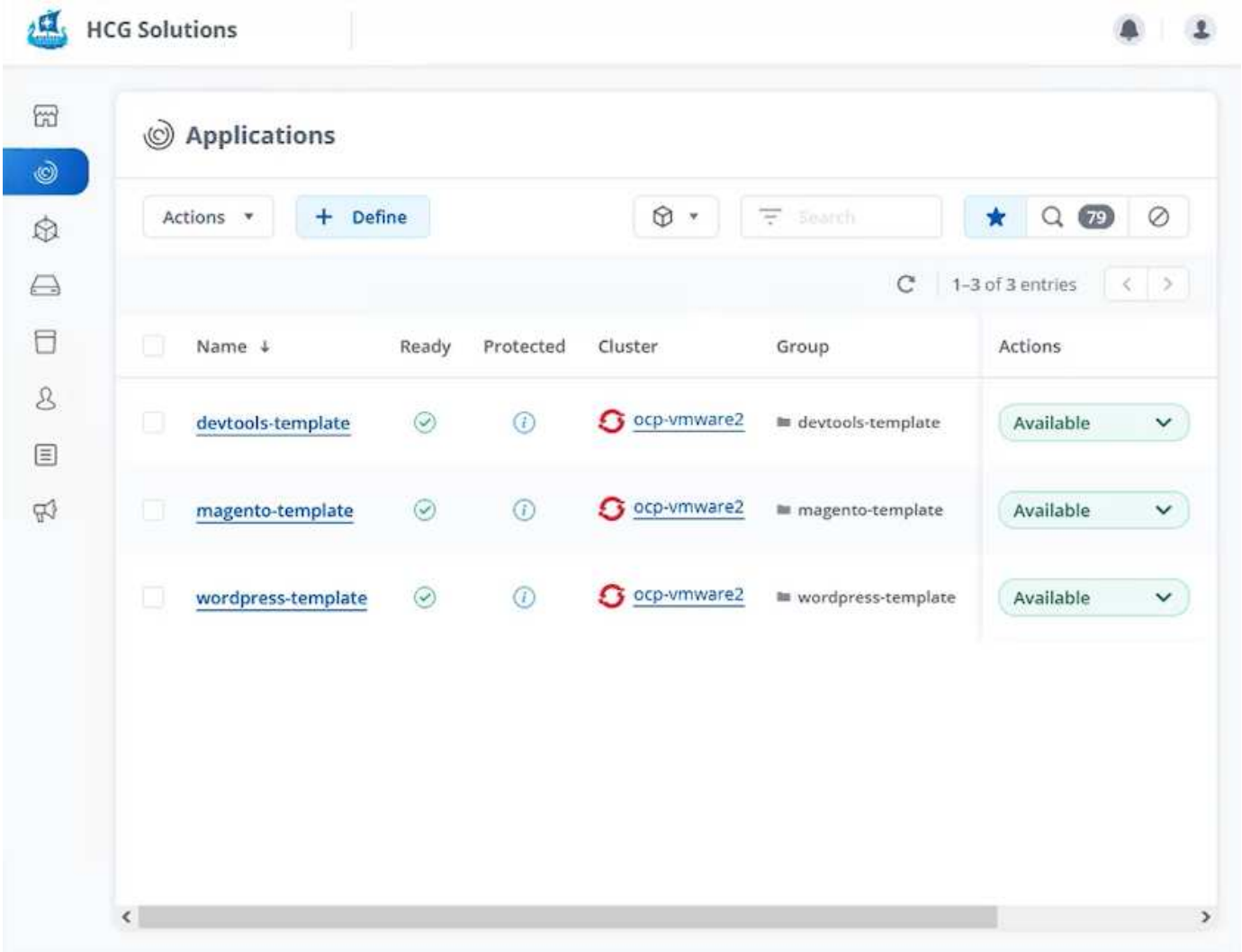
> この環境は、開発中の単一のスタンドアロンアプリケーションである場合もあれば、Jenkins CI/CDパイプラインなどの開発環境全体である場合もあります。

4. 次に'ユーザは'コードのバージョンを'Giteaなどのオンラインリポジトリからクローン開発環境にプルします
5. 新しいバージョンのアプリケーションは、NetApp Astra Controlによって導入および管理されます。

> どちらのプロセスも自動化できます。

6. ユーザーは、このクローン環境で新しいコードを開発できます。
7. ユーザーが開発作業に満足したら、ホストされているリポジトリにコードを戻すことができます。

ここで紹介する使用事例は、クローンを作成する特定の環境またはアプリケーション用のゴールデンテンプレートの存在によって異なります。この環境では、ワードプレス展開用に、Magento展開用に、Giteaを持つJenkins CI/CD環境用に、DevToolsというタイトルの3つのテンプレートを作成しました。



各環境はNetApp Astra制御によって管理され、ネットアップONTAP ストレージシステムに現在格納されている永続的ボリュームは、ネットアップAstra Tridentが提供するNFSバックエンドを使用している。

ユースケースの検証

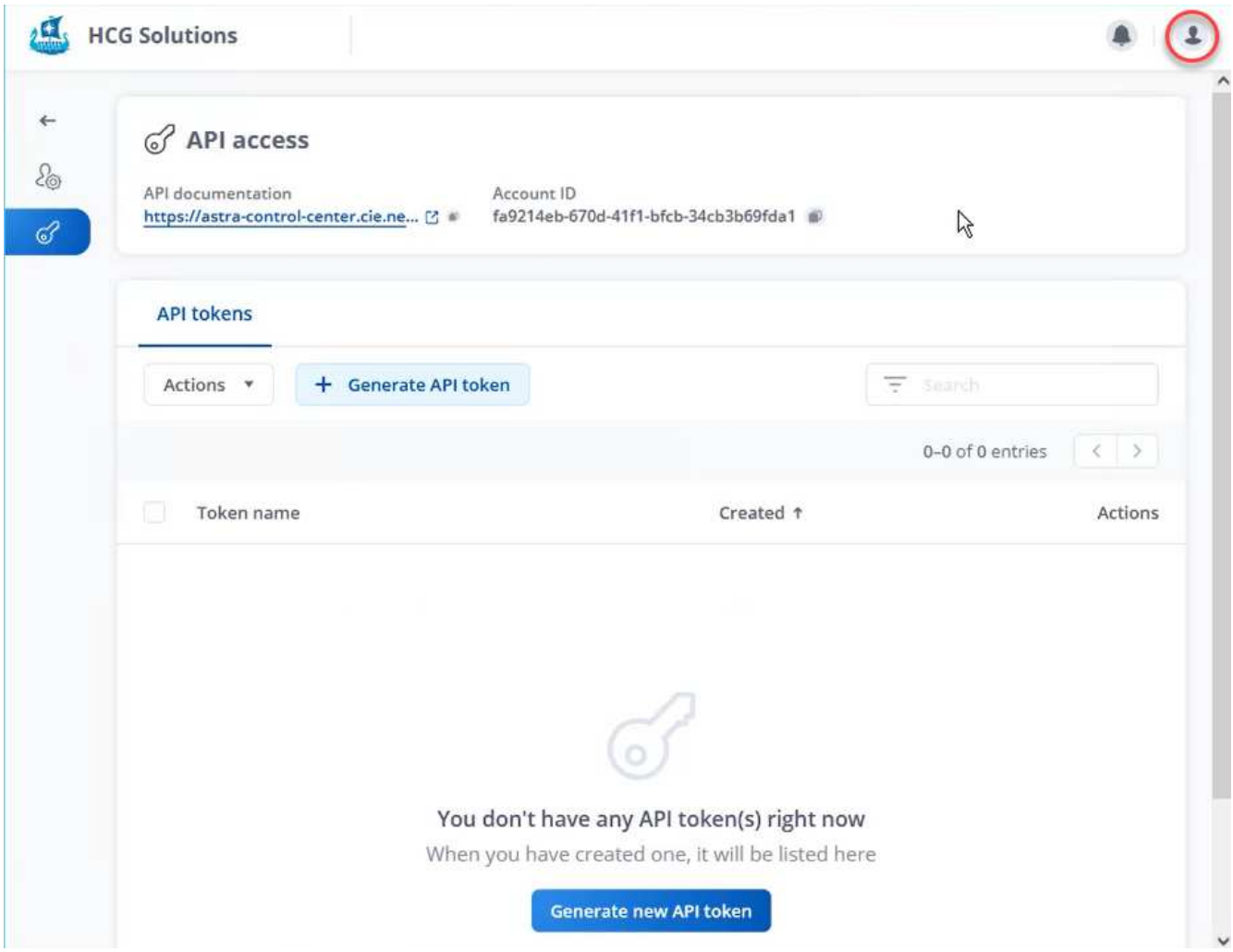
1. ネットアップのソリューションエンジニアリングチームが提供するAnsibleツールキットをクローニングします。これには、クローニングのロールとアプリケーションの更新に関するプレイブックが含まれます。

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

2. 'vars/clone_vars.yml'を編集し、Astra Control環境に適したグローバル値を入力します。

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

入力する必要があるグローバル環境値は、NetApp Astra ControlのAPIアクセスメニューにあるユーザプロファイルアイコンで確認できます。



3. グローバル変数が完成したら、複製する特定のアプリケーションの値を選択できます。DevTools環境を'Alan - DevTools'という個人環境にクローンを作成するには'次の手順を実行します

```
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
```

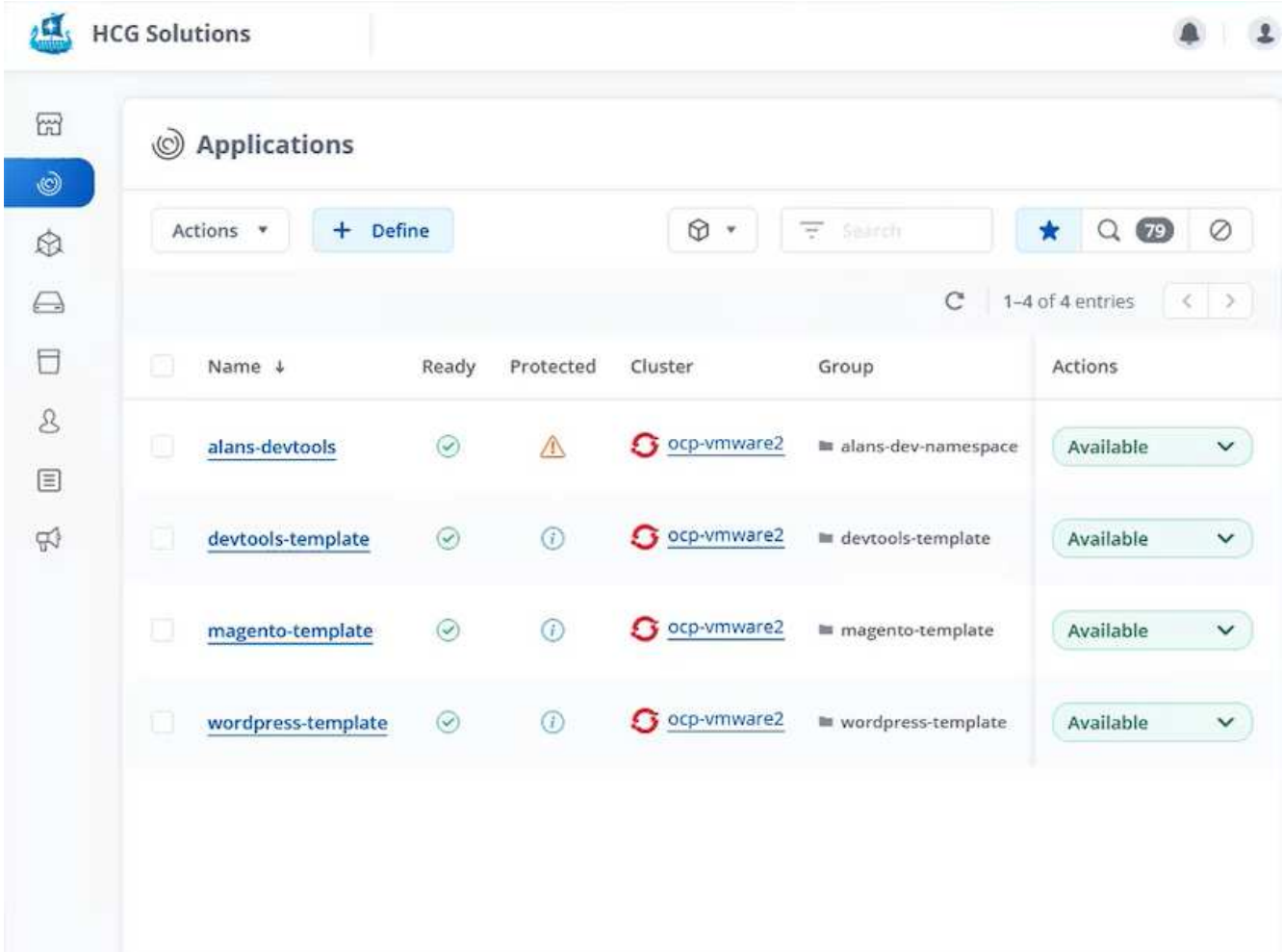
クローン作成プロセスでNetApp FlexCloneテクノロジを活用するには'src-cluster'とdest-cluster'を同じにする必要があります

4. プレイブックを実行してアプリケーションをクローニングできるようになりました。

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

記載されたプレイブックは、rootユーザ、またはsudoプロセスを介して「-K」引数を渡してエスカレーションできるユーザが実行する必要があります。

5. プレイブックの実行が完了すると、クローニングされたアプリケーションがAstra Control Centerコンソールに表示されます。



6. ユーザは、アプリケーションが導入されたKubernetes環境にログインし、アプリケーションが新しいIPアドレスで公開されていることを確認して、開発作業を開始できます。

この使用例とアプリケーションのアップグレード例のデモについては、を参照してください "こちらをご覧ください"。

著作権に関する情報

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。