



Red Hat OpenShift GitOps 1.13

OpenShift GitOps について

OpenShift GitOps の概要

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、OpenShift GitOps とその機能の概要を説明します。また、一般的な OpenShift GitOps 用語を定義する用語集も提供します。

目次

第1章 GITOPS とは	3
第2章 RED HAT OPENSIFT GITOPS について	4
2.1. 主な特長	4
2.2. OPENSIFT GITOPS の一般的な用語集	5
2.3. 関連情報	8
第3章 サポート用の診断情報の収集	10
3.1. MUST-GATHER ツールについて	10
3.2. RED HAT OPENSIFT GITOPS のデバッグデータの収集	11
3.3. 関連情報	12

第1章 GITOPS とは

GitOps は、クラウドネイティブアプリケーションの継続的デプロイメントを実装するための宣言的な方法です。GitOps を使用して、複数クラスターの Kubernetes 環境全体で、OpenShift Container Platform クラスターおよびアプリケーションを管理するための反復可能なプロセスを作成できます。GitOps は、早いペースで複雑なデプロイメントを処理して自動化し、デプロイメントおよびリリースサイクルでの時間を節約します。

GitOps ワークフローは、開発、テスト、ステージング、および実稼働環境にアプリケーションをプッシュします。GitOps は新しいアプリケーションをデプロイするか、既存のアプリケーションを更新するため、必要なのはリポジトリの更新のみとなります。他のものはすべて GitOps が自動化します。

GitOps は、Git プル要求を使用してインフラストラクチャーおよびアプリケーションの設定を管理する一連の手法で設定されます。GitOps では、Git リポジトリが、システムおよびアプリケーション設定の信頼できる唯一の情報源 (source of truth) になります。この Git リポジトリには、指定した環境に必要なインフラストラクチャーの宣言的な説明が含まれ、環境を説明した状態に一致させるための自動プロセスが含まれます。また、Git リポジトリにはシステムの全体の状態が含まれるため、システムの状態への変更の追跡情報が表示され、監査可能になります。GitOps を使用することで、インフラストラクチャーおよびアプリケーション設定のスプロールの問題を解決します。

GitOps は、インフラストラクチャーおよびアプリケーションの定義をコードとして定義します。次に、このコードを使用して複数のワークスペースおよびクラスターを管理し、インフラストラクチャーおよびアプリケーション設定の作成を単純化します。コードの原則に従って、クラスターおよびアプリケーションの設定を Git リポジトリに保存し、Git ワークフローに従って、これらのリポジトリを選択したクラスターに適用できます。Git リポジトリでのソフトウェアの開発およびメンテナンスのコアとなる原則を、クラスターおよびアプリケーションの設定ファイルの作成および管理に適用できます。

第2章 RED HAT OPENSIFT GITOPS について

Red Hat OpenShift GitOps は、宣言型の GitOps エンジンである Argo CD を使用する Operator です。これにより、マルチクラスター OpenShift および Kubernetes インフラストラクチャー全体で GitOps ワークフローが可能になります。管理者は、Red Hat OpenShift GitOps を使用して、クラスターおよび開発ライフサイクル全体で Kubernetes ベースのインフラストラクチャーとアプリケーションを一貫して設定およびデプロイできます。Red Hat OpenShift GitOps は、オープンソースプロジェクト [Argo CD](#) をベースにしています。アップストリームが提供するものと同様の一連の機能に加えて、さらなる自動化、Red Hat OpenShift Container Platform への統合、Red Hat のエンタープライズサポートの利点 (品質保証とエンタープライズセキュリティの重視) を提供します。

注記

Red Hat OpenShift GitOps は OpenShift Container Platform とは異なる頻度でリリースされるため、Red Hat OpenShift GitOps ドキュメントは製品のマイナーバージョンごとに個別のドキュメントセットとして利用できるようになりました。

Red Hat OpenShift GitOps ドキュメントは <https://docs.openshift.com/gitops/> で利用できます。

特定のバージョンのドキュメントは、バージョンセクターのドロップダウンを使用するか、URL にバージョン (<https://docs.openshift.com/gitops/1.8> など) を直接追加すると利用できます。

さらに、Red Hat OpenShift GitOps ドキュメントは、Red Hat カスタマーポータル (https://access.redhat.com/documentation/ja-jp/red_hat_openshift_gitops/) でも利用できます。

Red Hat OpenShift GitOps のライフサイクルとサポートされるプラットフォームに関する追加情報は、[プラットフォームのライフサイクルポリシー](#) を参照してください。

Red Hat OpenShift GitOps は、異なる環境 (開発、ステージ、実稼働環境など) の異なるクラスターにアプリケーションをデプロイする場合に、アプリケーションの一貫性を確保します。Red Hat OpenShift GitOps は、設定リポジトリに関連するデプロイメントプロセスを整理し、それらを中心的な要素にします。これには、少なくとも 2 つのリポジトリが常に含まれます。

1. ソースコードを含むアプリケーションリポジトリ
2. アプリケーションの必要な状態を定義する環境設定リポジトリ

これらのリポジトリには、指定した環境で必要なインフラストラクチャーの宣言的な説明が含まれます。また、環境を記述された状態に一致させる自動プロセスも含まれています。

Red Hat OpenShift GitOps は Argo CD を使用してクラスターリソースを維持します。Argo CD は、アプリケーションの継続的デプロイメント (CD) のためのオープンソースの宣言型ツールです。Red Hat OpenShift GitOps は Argo CD をコントローラーとして実装し、Git リポジトリで定義されるアプリケーション定義および設定を継続的に監視します。次に、Argo CD は、これらの設定の指定された状態をクラスターのライブ状態と比較します。

Argo CD は、指定した状態から逸脱する設定を報告します。これらの報告により、管理者は、設定を定義された状態に自動または手動で再同期できます。したがって、Argo CD を使用して、OpenShift Container Platform クラスターを設定するために使用されるリソースなどのグローバルカスタムリソースを配信できます。

2.1. 主な特長

Red Hat OpenShift GitOps は、以下のタスクを自動化する上で役立ちます。

- クラスターの設定、モニタリング、ストレージの状態が類似していることを確認する
- 複数の OpenShift Container Platform クラスターに対する設定変更を適用するか、これを元に戻す
- テンプレート化された設定をさまざまな環境に関連付ける。
- ステージングから実稼働環境まで、クラスター間でアプリケーションをプロモートする。

2.2. OPENSIFT GITOPS の一般的な用語集

この用語集では、一般的な OpenShift GitOps 用語を定義します。

Application Controller (Argo CD Application Controller)

次のアクションを実行するコントローラー:

- Git リポジトリの変更を継続的に監視する
- 実行中のアプリケーションを監視する
- ライブ状態と目的のターゲット状態を比較する
- 新しい変更をデプロイする

例としては、Argo CD Application Controller が **OutOfSync** アプリケーションの状態を検出し、必要に応じて修正アクションを実行することが挙げられます。

Application カスタムリソース (CR)

Argo CD アプリケーションのリソースをどのようにデプロイするかを記述する YAML マニフェスト。

Application カスタムリソース定義 (CRD)

環境にデプロイされた Argo CD アプリケーションインスタンスを表すリソースオブジェクト。

ApplicationSet CRD (Argo CD アプリケーションセット)

ApplicationSet CR の内容に基づいて Argo CD アプリケーションを自動的に生成するリソースオブジェクトと CRD。クラスター管理者はこの CRD を使用して単一の **ApplicationSet** CR を定義し、対応する Argo CD **アプリケーション** CR を複数生成および更新します。

ApplicationSet Controller (Argo CD ApplicationSet Controller)

Argo CD 内に存在し、**ApplicationSet** CR を処理するカスタム Kubernetes コントローラー。このコントローラーは、**ApplicationSet** CR の内容に基づいて、Argo CD アプリケーションを自動的に作成、更新、および削除します。

AppProject CRD

プロジェクト内のアプリケーションの論理グループを表す CRD であり、アプリケーションがリソースを管理できる場所と方法を制御します。**AppProject** CRD を使用すると、Argo CD ユーザーがこれらのアプリケーションにアクセスできる場所と方法を制限できます。**AppProject** インスタンスの管理は、通常、Argo CD 管理者限定のアクションです。

Argo CD API サーバー

Web UI、CLI、継続的インテグレーション (CI)、継続的デプロイメント (CD) システムで使用される API を公開する gRPC/REST サーバー。

Argo CD

クラスターと開発ライフサイクル全体にわたる Kubernetes ベースのインフラストラクチャーとアプリケーションの継続的なデプロイメントを自動化するオープンソースの宣言型ツール。

Argo CD アプリケーション

リソースがマニフェストとして定義されている GitOps リポジトリからターゲット Kubernetes クラスターまでの個々の Kubernetes リソースの継続的なデプロイメントを追跡するアプリケーション。

ArgoCD CRD

特定の Argo CD クラスターの希望の状態を記述する Kubernetes CRD。これにより、Argo CD クラスターを設定するコンポーネントを設定できます。

Argo CD インスタンス

実行中の Argo CD のすべてのステートフルな部分をカプセル化する namespace 内の Argo CD の単一インストール。通常、各 Argo CD インスタンスには **ArgoCD** CR との 1対1のマッピングがあります。

Argo CD プロジェクト

Argo CD 内のエンティティであり、Argo CD オープンソースプロジェクトの [特定のプロジェクトの概念](#) と、対応する **AppProject CR** を参照します。

Argo CD プロジェクトを使用すると、許可された宛先として複数の namespace やクラスターを定義できます。対照的に、[OpenShift プロジェクト](#) は単一の namespace に制限され、namespace の概念と同じです。

Argo CD プロジェクトは、Git リポジトリおよびリモートクラスターへのアクセスを制限することで、Argo CD の動作を制御します。例としては、Argo CD プロジェクトを使用して、Argo CD UI または Argo CD CLI を介して特定の Argo CD アプリケーションまたはクラスターリソースにアクセスできるユーザーを制限してユーザーを制御することが挙げられます。

Argo CD リポジトリサーバー (Argo CD-repo-server)

次のアクションを実行する Argo CD コンポーネント:

- Git、Helm、Open Container Initiative (OCI) などのソースリポジトリから読み取る
- 対応するアプリケーションマニフェストを生成する
- カスタム設定管理ツールを実行する
- Argo CD アプリケーションコントローラーに結果を返す

Argo CD リソース (ArgoCD CR)

特定の Argo CD インスタンスの望ましい状態を記述する CR。Argo CD インスタンスを設定するコンポーネントと設定を指定できます。

Argo CD サーバー (Argo CD サーバー)

Argo CD の API と UI を提供するサーバー。

Argo Rollouts

Kubernetes および OpenShift Container Platform クラスターでホストされているアプリケーションのプロGRESSIVEデプロイメントを管理するために使用できるコントローラー。このコントローラーには、ブルーグリーン、カナリア、カナリア分析、実験などの高度なデプロイメント機能を提供する CRD のセットがあります。

クラスタースコープインスタンス

クラスター設定、クラスター RBAC、Operator リソース、プラットフォーム Operator、シークレットなどの特定のクラスター固有のリソースを含む、クラスター上のすべてのリソースを管理するように Argo CD が設定されるモード。

コントロールプレーン (GitOps コントロールプレーン)

GitOps コンテキストでは、インストールするすべての Argo CD のコントロールプレーンを設定できます。GitOps コントロールプレーンは、Argo CD をインストールできる任意の namespace です。このコントロールプレーンを使用すると、ネットワーク、インスタンス、およびクラスター全体で Argo CD をプロビジョニング、管理、および操作できます。

コントロールプレーン namespace 内で、Argo CD は、ソース Git リポジトリと宛先クラスター間の継続的デプロイメントを定義する次の Kubernetes リソースのセットを維持します。

- Argo CD **Application CR**
- **ConfigMap** API オブジェクト
- デプロイメントターゲットの GitOps リポジトリ認証情報とクラスター認証情報を表す **Secret** オブジェクト

openshift-gitops は、デフォルトの Argo CD インスタンスのコントロールプレーン namespace です。

宣言型セットアップ

システムおよびアプリケーションのセットアップまたは設定に、指定された環境に必要なインフラストラクチャーの宣言的な説明。この記述は、Git リポジトリの YAML 設定ファイルで指定できます。宣言型セットアップには、環境とインフラストラクチャーを記述された状態と同じにするための自動プロセスが含まれています。例としては、YAML マニフェストを使用して Argo CD アプリケーション、プロジェクト、設定を宣言的に定義することが挙げられます。

デフォルトの Argo CD インスタンス (デフォルトのクラスタースコープインスタンス)

Red Hat OpenShift GitOps Operator がインストール直後に **openshift-gitops** namespace にインスタンス化するデフォルトのインスタンスで、特定のクラスタースコープのリソースを管理するための追加の権限が付与されています。

GitOps

クラウドネイティブアプリケーションの継続的なデプロイメントを実装するための宣言的な方法。GitOps では、Git リポジトリにデプロイメントリソースが含まれており、Argo CD はそれをクラスターの状態と同期し続けます。

GitOps CLI (GitOps `argocd` CLI)

コマンドラインから Red Hat OpenShift GitOps および Argo CD リソースを設定および管理するためのツール。

インスタンスの範囲

Argo CD インスタンスを操作する方法を決定するモード。利用可能なモードは **クラスタースコープのインスタンス** および **namespace スコープのインスタンス** です。

ライブ状態

ターゲットクラスター上のアプリケーションリソースのライブ状態。

ローカルクラスター

Argo CD をインストールするクラスター。

マニフェスト

GitOps のコンテキストでは、マニフェストは GitOps リポジトリ内で定義された Kubernetes リソースの YAML 表現であり、それらのリソースをターゲットの Kubernetes クラスターにデプロイすることを目的としています。たとえば、**Deployment**、**ConfigMap**、**Secret** などのリソースの YAML 表現が含まれます。

マルチテナントへの対応

単一のソフトウェアインスタンスがさまざまなユーザーグループにサービスを提供するソフトウェアアーキテクチャー。

namespace スコープインスタンス (アプリケーション配信インスタンス)

Argo CD がクラスター上の特定の namespace 内のリソースのみを管理し、そのリソースをアプリケーション配信に使用するように設定されたモード。

Notifications Controller (Argo CD Notifications Controller)

Argo CD アプリケーションを継続的に監視し、アプリケーション状態の重要な変更についてユーザーに通知する柔軟な方法を提供するコントローラー。

プログレッシブ配信

GitOps のコンテキストでは、プログレッシブ配信は、制御された段階的な方法でアプリケーションの更新をリリースするプロセスです。

Red Hat OpenShift GitOps

宣言型 GitOps エンジンとして Argo CD を使用し、マルチクラスター OpenShift および Kubernetes インフラストラクチャー全体で GitOps ワークフローを有効にする Operator。

更新

Git リポジトリ内の最新のコードをライブ状態と比較し、その違いを判断するプロセス。たとえば、Argo CD UI では、**Refresh** をクリックすると、Argo CD はアプリケーションのターゲット Git リポジトリに接続し、コンテンツを取得し、そのコンテンツからマニフェストを生成します。次に、Argo CD はそのターゲット状態をライブクラスターの状態と比較します。

リモートクラスター

宣言的または GitOps CLI を使用して Argo CD に追加できるクラスター。リモートクラスターは、Argo CD がインストールされているローカルクラスターとは異なります。

リソースの除外

Argo CD がリソースを認識しないように、リソースを検出および同期から除外するために使用する設定。

リソースの包含

グローバルに管理対象リソースのリストを検出、同期、制限するために使用する設定。

シングルテナンシー

単一のソフトウェアインスタンスが単一のユーザーまたはグループを提供するソフトウェアアーキテクチャー。

同期

一貫性を確保するために、アプリケーションのクラスターリソースのライブ状態を Git リポジトリ内で定義されたターゲット状態と同期するプロセス。たとえば、Argo CD UI を使用して変更をクラスターに適用することによるアプリケーションの同期などがあります。

同期の状態

ライブマイグレーションがターゲット状態と一致するかどうかを示すアプリケーションの状態。

ターゲットの状態

Git リポジトリ内のファイルによって表される、アプリケーションリソースの望ましい状態。

ユーザー定義の Argo CD インスタンス

クラスター設定を管理したり、アプリケーションをデプロイしたりするためにインストールしてデプロイするカスタム Argo CD インスタンス。デフォルトでは、新しいユーザー定義のインスタンスには、デプロイされた namespace でのみリソースを管理する権限があります。

ワークロード

通常、コンテナ内で実行される **Deployment**、**StatefulSet**、**ReplicaSet**、**Job**、**Pod** などのリソース内で定義されるプロセス。たとえば、Spring Boot アプリケーション、NodeJS Express アプリケーション、Ruby on Rails アプリケーションなどが含まれます。

2.3. 関連情報

- カスタムリソース定義による Kubernetes API の拡張
- カスタムリソース定義からのリソースの管理
- GitOps とは
- OpenShift プロジェクトとは
- Specification of an **AppProject** CRD

第3章 サポート用の診断情報の収集

サポートケースを開くときは、クラスターに関するデバッグ情報を Red Hat サポートチームに提供する必要があります。**must-gather** ツールを使用して、プロジェクトレベルのリソース、クラスターレベルのリソース、および Red Hat OpenShift GitOps コンポーネントの診断情報を収集できます。



注記

迅速なサポートが必要な場合は、OpenShift Container Platform と Red Hat OpenShift GitOps の両方の診断情報を提供してください。

3.1. MUST-GATHER ツールについて

oc adm must-gather CLI コマンドは、以下のような問題のデバッグに必要となる可能性のあるクラスターからの情報を収集します。

- リソース定義
- サービスログ

デフォルトで、**oc adm must-gather** コマンドはデフォルトのプラグインイメージを使用し、**./must-gather.local** に書き込みを行います。

または、以下のセクションで説明されているように、適切な引数を指定してコマンドを実行すると、特定の情報を収集できます。

- 1つ以上の特定の機能に関連するデータを収集するには、以下のセクションに示すように、イメージと共に **--image** 引数を使用します。

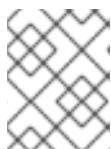
コマンドの例

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
```

- 監査ログを収集するには、以下のセクションで説明されているように **--/usr/bin/gather_audit_logs** 引数を使用します。

コマンドの例

```
$ oc adm must-gather -- /usr/bin/gather_audit_logs
```



注記

ファイルのサイズを小さくするために、監査ログはデフォルトの情報セットの一部として収集されません。

oc adm must-gather を実行すると、ランダムな名前を持つ新規 Pod がクラスターの新規プロジェクトに作成されます。データは Pod で収集され、**must-gather.local** で始まる新規ディレクトリーに保存されます。このディレクトリーは、現行の作業ディレクトリーに作成されます。

サンプル Pod

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
-----------	------	-------	--------	----------	-----

```
...
openshift-must-gather-5drcj  must-gather-bklx4  2/2  Running  0      72s
openshift-must-gather-5drcj  must-gather-s8sdh  2/2  Running  0      72s
...
```

任意で、**--run-namespace** オプションを使用して、特定の namespace で **oc adm must-gather** コマンドを実行できます。

コマンドの例

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
```

3.2. RED HAT OPENSIFT GITOPS のデバッグデータの収集

oc adm must-gather CLI コマンドを使用して、Red Hat OpenShift GitOps に関連付けられているクラスターに関する次の詳細を収集します。

- Red Hat OpenShift GitOps Operator のサブスクリプションと namespace。
- ArgoCD オブジェクトが利用できる namespace と、**ArgoCD**、**Applications**、**ApplicationSets**、**AppProjects**、**configmaps** などのこれらの namespace のオブジェクト。
- Red Hat OpenShift GitOps Operator によって管理される namespace の一覧およびそれらの namespace からのリソース。
- すべての GitOps 関連のカスタムリソースオブジェクトと定義。
- Operator および Argo CD のログ。
- 警告およびエラーレベルのイベント。

前提条件

- OpenShift Container Platform クラスターに管理者としてログインしている。
- OpenShift Container Platform CLI (**oc**) がインストールされている。
- Red Hat OpenShift GitOps Operator がインストールされている。

手順

1. デバッグ情報を保存するディレクトリーに移動します。
2. Red Hat OpenShift GitOps の **must-gather** イメージを使用して **oc adm must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:
<image_version_tag> ①
```

- ① GitOps の must-gather イメージ。

コマンドの例

-

```
$ oc adm must-gather --image=registry.redhat.io/openshift-gitops-1/must-gather-rhel8:v1.10.0
```

must-gather ツールは、現在のディレクトリーに **./must-gather.local** で始まる新しいディレクトリーを作成します。たとえば、**./must-gather.local.4157245944708210399** です。

- 作成したばかりのディレクトリーから圧縮ファイルを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar -cvaf must-gather.tar.gz must-gather.local.4157245944708210399 1
```

- 1** **must-gather-local.4157245944708210399** は、実際のディレクトリー名に置き換えます。

- 圧縮ファイルを [Red Hat カスタマーポータル](#) で作成したサポートケースに添付します。

3.3. 関連情報

- [特定の機能に関するデータ収集](#)