



Red Hat OpenShift GitOps 1.12

可観測性

可観測性機能を使用した Argo CD ログを表示と Argo CD およびアプリケーションリソースのパフォーマンスとヘルスのモニタリング

Red Hat OpenShift GitOps 1.12 可観測性

可観測性機能を使用した Argo CD ログを表示と Argo CD およびアプリケーションリソースのパフォーマンスとヘルスのモニタリング

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 で OpenShift Logging を使用し、Argo CD インスタンスのパフォーマンス、アプリケーションのヘルスステータス、および Argo CD カスタムリソースのワークロードをモニタリングする方法について詳しく説明します。

目次

第1章 ログング	3
1.1. ARGO CD ログの表示	3
第2章 モニタリング	5
2.1. GITOPS ダッシュボードによるモニタリング	5
2.2. ARGO CD インスタンスのモニタリング	6
2.3. GITOPS OPERATOR のパフォーマンスのモニタリング	6
2.4. アプリケーションリソースおよびデプロイメントのヘルス情報のモニタリング	8
2.5. ARGO CD カスタムリソースワークロードのモニタリング	10

第1章 ロギング

1.1. ARGO CD ログの表示

Red Hat OpenShift のロギングサブシステムを使用して Argo CD ログを表示できます。ログサブシステムは、Kibana ダッシュボード上でログを視覚化します。OpenShift Logging Operator は、デフォルトで Argo CD を使用したロギングを有効にします。

1.1.1. Argo CD ログの保存と取得

Kibana ダッシュボードを使用して、Argo CD ログを保存および取得できます。

前提条件

- Red Hat OpenShift GitOps Operator が OpenShift Container Platform クラスタにインストールされている。
- Red Hat OpenShift のロギングサブシステムが、OpenShift Container Platform クラスタにデフォルト設定でインストールされている。

手順

1. OpenShift Container Platform Web コンソールで、 メニュー → **Observability** → **Logging** に移動して Kibana ダッシュボードを表示します。
2. インデックスパターンを作成します。
 - a. すべてのインデックスを表示するには、インデックスパターンを * と定義し、**Next step** をクリックします。
 - b. **Time Filter field name** で **@timestamp** を選択します。
 - c. **Create index pattern** をクリックします。
3. Kibana ダッシュボードのナビゲーションパネルで、**Discover** タブをクリックします。
4. Argo CD のログを取得するフィルターを作成します。次の手順では、**openshift-gitops** namespace 内のすべての Pod のログを取得するフィルターを作成します。
 - a. **Add a filter +** をクリックします。
 - b. **kubernetes.namespace_name** フィールドを選択します。
 - c. **is** 演算子を選択します。
 - d. **openshift-gitops** 値を選択します。
 - e. **Save** をクリックします。
5. オプション: フィルターを追加して検索を絞り込みます。たとえば、特定の Pod のログを取得するには、フィールドとして **kubernetes.pod_name** を使用して別のフィルターを作成できます。
6. Kibana ダッシュボードでフィルタリングされた Argo CD ログを表示します。

1.1.2. 関連情報

- [Web コンソールを使用した Red Hat のロギングサブシステムのインストール](#)

第2章 モニタリング

2.1. GITOPS ダッシュボードによるモニタリング

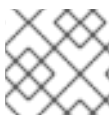
Red Hat OpenShift GitOps モニタリングダッシュボードを使用して GitOps インスタンスのグラフィカルビューにアクセスし、クラスター全体の各インスタンスの動作と使用状況を観察できます。

利用可能な GitOps ダッシュボードは 3 つあります。

- **GitOps の概要:** クラスターにインストールされているすべての GitOps インスタンスの概要 (アプリケーションの数、健全性と同期ステータス、アプリケーションと同期アクティビティーなど) を表示します。
- **GitOps コンポーネント:** アプリケーションコントローラー、リポジトリサーバー、サーバー、およびその他の GitOps コンポーネントの CPU やメモリーなどの詳細情報を表示します。
- **GitOps gRPC サービス:** Red Hat OpenShift GitOps のさまざまなコンポーネント間の gRPC サービスアクティビティーに関連するメトリクスを表示します。

2.1.1. GitOps モニタリングダッシュボードへのアクセス

モニタリングダッシュボードは、Operator によって自動的にデプロイされます。OpenShift Container Platform Web コンソールの **Administrator** パースペクティブから、GitOps モニタリングダッシュボードにアクセスできます。



注記

ダッシュボードのコンテンツの無効化または変更はサポートされていません。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- Red Hat OpenShift GitOps Operator が、デフォルトの namespace (**openshift-gitops-operator**) にインストールされている。
- クラスターモニタリングが、**openshift-gitops-operator** namespace で有効化されている。
- 定義した namespace (たとえば **openshift-gitops**) に Argo CD アプリケーションをインストールしている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Observe** → **Dashboards** に移動します。
2. **Dashboard** ドロップダウンリストから、目的の GitOps ダッシュボード (**GitOps (Overview)**、**GitOps / Components**、または **GitOps / gRPC Services**) を選択します。
3. オプション: **Namespace**、**Cluster**、および **Interval** ドロップダウンリストから特定の namespace、クラスター、および間隔を選択します。
4. GitOps ダッシュボードで必要な GitOps メトリクスを表示します。

2.2. ARGO CD インスタンスのモニタリング

デフォルトでは、Red Hat OpenShift GitOps Operator は、定義された namespace (例: **openshift-gitops**) にインストールされている Argo CD インスタンスを自動的に検出し、これをクラスターのモニタリングスタックに接続して、非同期アプリケーションに対するアラートを提供します。

2.2.1. 前提条件

- **cluster-admin** 権限でクラスターにアクセスできる。
- OpenShift Container Platform Web コンソールにアクセスできる。
- Red Hat OpenShift GitOps Operator が OpenShift Container Platform クラスターにインストールされている。
- 定義した namespace (たとえば **openshift-gitops**) に Argo CD アプリケーションをインストールしている。

2.2.2. Prometheus メトリクスを使用した Argo CD ヘルスのモニタリング

Prometheus メトリクスクエリーを実行して、Argo CD アプリケーションのヘルスステータスをモニタリングできます。

手順

1. Web コンソールの **Developer** パースペクティブで、Argo CD アプリケーションがインストールされている namespace を選択し、**Observe** → **Metrics** に移動します。
2. **Select query** ドロップダウンリストから、**Custom query** を選択します。
3. Argo CD アプリケーションのヘルスステータスを確認するには、**Expression** フィールドに、次の例のような Prometheus Query Language (PromQL) クエリーを入力します。

例

```
sum(argocd_app_info{dest_namespace=~"<your_defined_namespace>",health_status!=""})
by (health_status) 1
```

- 1** **<your_define_namespace>** 変数を、定義した namespace の実際の名前 (**openshift-gitops** など) に置き換えます。

2.3. GITOPS OPERATOR のパフォーマンスのモニタリング

Red Hat OpenShift GitOps Operator は、パフォーマンスに関するメトリクスを生成します。これらのメトリクスを取得する OpenShift モニタリングスタックを使用すると、Operator のパフォーマンスをモニタリングおよび分析できます。Operator は以下のメトリクスを公開します。これらのメトリクスは、OpenShift Container Platform Web コンソールを使用して表示できます。

表2.1 GitOps Operator のパフォーマンスメトリクス

メトリクス名	型	説明
<code>active_argocd_instances_total</code>	ゲージ	特定の時点でクラスター全体で Operator によって現在管理されているアクティブな Argo CD インスタンスの合計数。
<code>active_argocd_instances_by_phase</code>	ゲージ	保留中や利用可能など、特定のフェーズにおけるアクティブな Argo CD インスタンスの数。
<code>active_argocd_instance_reconciliation_count</code>	カウンター	特定の時点で、特定の namespace 内のインスタンスに対して発生した調整の合計数。
<code>controller_runtime_reconcile_time_seconds_per_instance_bucket</code>	カウンター	インスタンスの指定された期間内に完了した調整サイクルの数。たとえば、 <code>controller_runtime_reconcile_time_seconds_per_instance_bucket{le="0.5"}</code> は、0.5 秒未満で特定のインスタンスを完了した調整の数を示します。
<code>controller_runtime_reconcile_time_seconds_per_instance_count</code>	カウンター	特定のインスタンスで観察された調整サイクルの合計数。
<code>controller_runtime_reconcile_time_seconds_per_instance_sum</code>	カウンター	特定のインスタンスで観察された調整に要した合計時間。



注記

ゲージは増減する値です。カウンターは増加のみの値です。

2.3.1. GitOps Operator メトリクスへのアクセス

OpenShift Container Platform Web コンソールの **Administrator** パースペクティブから Operator メトリクスにアクセスして、Operator のパフォーマンスを追跡できます。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- Red Hat OpenShift GitOps Operator が、デフォルトの **openshift-gitops-operator** namespace にインストールされている。
- クラスターモニタリングが、**openshift-gitops-operator** namespace で有効化されている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Observe** → **Metrics** に移動します。
2. **Expression** フィールドにメトリクスを入力します。次のメトリクスから選択できます。
 - **active_argocd_instances_total**
 - **active_argocd_instances_by_phase**
 - **active_argocd_instance_reconciliation_count**
 - **controller_runtime_reconcile_time_seconds_per_instance_bucket**
 - **controller_runtime_reconcile_time_seconds_per_instance_count**
 - **controller_runtime_reconcile_time_seconds_per_instance_sum**
3. (オプション): プロパティごとにメトリクスをフィルターします。たとえば、**active_argocd_instances_by_phase** メトリクスを **Available** フェーズでフィルターします。

例

```
active_argocd_instances_by_phase{phase="Available"}
```

4. (オプション): **Add query** をクリックして、複数のクエリーを入力します。
5. **Run queries** をクリックして、GitOps Operator メトリクスを有効にして観察します。

2.3.2. 関連情報

- [Red Hat OpenShift GitOps Operator を Web コンソールにインストールする](#)

2.4. アプリケーションリソースおよびデプロイメントのヘルス情報のモニタリング

OpenShift Container Platform Web コンソールの **Developer** パースペクティブにある Red Hat OpenShift GitOps **Environments** ページには、成功したアプリケーション環境のデプロイメントのリスト、および各デプロイメントのリビジョンへのリンクが表示されます。

OpenShift Container Platform Web コンソールの **Developer** パースペクティブの **Application environments** ページには、ルート、同期ステータス、デプロイメント設定、デプロイメント履歴などのアプリケーションリソースのヘルスステータスが表示されます。

OpenShift Container Platform Web コンソールの **Developer** パースペクティブの環境ページは、Red Hat OpenShift GitOps Application Manager コマンドラインインターフェイス (CLI) の **kam** から分離されています。環境が OpenShift Container Platform Web コンソールの **Developer** パースペクティブに表示されるように、**kam** を使用して、Application Environment マニフェストを生成する必要はありません。独自のマニフェストを使用できますが、環境は引き続き namespace で表す必要があります。さらに、特定のラベルとアノテーションが必要です。

2.4.1. 環境ラベルとアノテーションの設定

このセクションでは、OpenShift Container Platform Web コンソールの **Developer** パースペクティブの **Environments** ページに環境アプリケーションを表示するために必要な環境ラベルとアノテーションの設定を参考として示します。

環境ラベル

環境アプリケーションマニフェストに、**labels.openshift.gitops/environment** フィールドと **destination.namespace** フィールドが含まれている必要があります。<environment_name> 変数と環境アプリケーションマニフェストの名前には、必ず同じ値を設定してください。

環境アプリケーションマニフェストの仕様

```
spec:
  labels:
    openshift.gitops/environment: <environment_name>
  destination:
    namespace: <environment_name>
# ...
```

環境アプリケーションマニフェストの例

```
apiVersion: argoproj.io/v1beta1
kind: Application
metadata:
  name: dev-env ❶
  namespace: openshift-gitops
spec:
  labels:
    openshift.gitops/environment: dev-env
  destination:
    namespace: dev-env
# ...
```

- ❶ 環境アプリケーションマニフェストの名前。<environment_name> 変数の値と同じ値を設定します。

環境アノテーション

環境 namespace マニフェストには、アプリケーションのバージョンコントローラーコードソースを指定するための **annotations.app.openshift.io/vcs-uri** フィールドと **annotations.app.openshift.io/vcs-ref** フィールドが含まれている必要があります。<environment_name> 変数と環境 namespace マニフェストの名前には、必ず同じ値を設定してください。

環境 namespace マニフェストの仕様

```
apiVersion: v1
kind: Namespace
metadata:
  annotations:
    app.openshift.io/vcs-uri: <application_source_url>
    app.openshift.io/vcs-ref: <branch_reference>
  name: <environment_name> ❶
# ...
```

- ❶ 環境 namespace マニフェストの名前。<environment_name> 変数の値と同じ値を設定します。

環境 namespace マニフェストの例

```

apiVersion: v1
kind: Namespace
metadata:
  annotations:
    app.openshift.io/vcs-uri: https://example.com/<your_domain>/<your_gitops.git>
    app.openshift.io/vcs-ref: main
  labels:
    argocd.argoproj.io/managed-by: openshift-gitops
  name: dev-env
# ...

```

2.4.2. ヘルス情報の確認

Red Hat OpenShift GitOps Operator は、GitOps バックエンドサービスを **openshift-gitops** namespace にインストールします。

前提条件

- Red Hat OpenShift GitOps Operator が、**OperatorHub** からインストールされている。
- アプリケーションが Argo CD によって同期されている。

手順

1. **Developer** パースペクティブの下の **Environments** をクリックします。**Environments** ページには、**Environment status** と共にアプリケーションの一覧が表示されます。
2. **Environment status** 列の下のアイコンの上にマウスをかざすと、すべての環境の同期ステータスが表示されます。
3. 一覧からアプリケーション名をクリックし、特定のアプリケーションの詳細を表示します。
4. **Application environments** ページで、**Overview** タブの **Resources** セクションにアイコンが表示されている場合は、アイコンにカーソルを合わせると、ステータスの詳細が表示されます。
 - ひびの入ったハートは、リソースの問題によってアプリケーションのパフォーマンスが低下したことを示します。
 - 黄色の逆三角形は、リソースの問題により、アプリケーションのヘルスに関するデータが遅れたことを示します。
5. アプリケーションのデプロイメント履歴を表示するには、**Deployment History** タブをクリックします。このページには、**Last deployment**、**Description** (コミットメッセージ)、**Environment**、**Author**、および **Revision** などの詳細が含まれます。

2.5. ARGO CD カスタムリソースワークロードのモニタリング

Red Hat OpenShift GitOps を使用すると、特定の Argo CD インスタンスの Argo CD カスタムリソースワークロードの可用性をモニタリングできます。Argo CD カスタムリソースワークロードをモニタリングすると、Argo CD インスタンスのアラートを有効にして、その状態に関する最新情報を入手できます。対応する Argo CD インスタンスのアプリケーションコントローラー、リポジトリサーバー、ま

たはサーバーなどのコンポーネントワークロード Pod が特定の理由で起動できず、準備ができていないレプリカの数と必要なレプリカの数にずれがある場合、一定期間、Operator がアラートをトリガーします。

Argo CD カスタムリソースのワークロードをモニタリングするための設定を有効または無効にすることができます。

2.5.1. 前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- Red Hat OpenShift GitOps Operator が OpenShift Container Platform クラスターにインストールされている。
- モニタリングスタックが、**openshift-monitoring** プロジェクトのクラスターで設定されている。さらに、Argo CD インスタンスが、Prometheus を介してモニタリングできる namespace にある。
- **kube-state-metrics** サービスがクラスターで実行されている。
- オプション: ユーザー定義プロジェクトにすでに存在する Argo CD インスタンスのモニタリングを有効にする場合は、クラスター内の **ユーザー定義プロジェクトに対してモニタリングが有効** になっている。



注記

デフォルトの **openshift-monitoring** スタックによってモニタリングされていない namespace (たとえば、**openshift-*** で始まらない namespace) で Argo CD インスタンスのモニタリングを有効にする場合は、クラスターでユーザーワークロードのモニタリングを有効にする必要があります。このアクションにより、モニタリングスタックが作成された PrometheusRule を取得できるようになります。

2.5.2. Argo CD カスタムリソースワークロードのモニタリングを有効にする

デフォルトでは、Argo CD カスタムリソースワークロードのモニタリング設定は、**false** に設定されています。

Red Hat OpenShift GitOps を使用すると、特定の Argo CD インスタンスのワークロードモニタリングを有効にすることができます。その結果、Operator は、特定の Argo CD インスタンスによって管理されるすべてのワークロードのアラートルールを含む **PrometheusRule** オブジェクトを作成します。これらのアラートルールは、対応するコンポーネントのレプリカ数が一定時間、望ましい状態からずれると、アラートの起動をトリガーします。Operator は、ユーザーが **PrometheusRule** オブジェクトに加えた変更を上書きしません。

手順

1. 特定の Argo CD インスタンスで **.spec.monitoring.enabled** フィールドの値を **true** に設定します。

Argo CD カスタムリソースの例

```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
```

```

name: example-argocd
labels:
  example: repo
spec:
  # ...
  monitoring:
    enabled: true
  # ...

```

- Operator によって作成された PrometheusRule にアラートルールが含まれているかどうかを確認します。

アラートルールの例

```

apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: argocd-component-status-alert
  namespace: openshift-gitops
spec:
  groups:
    - name: ArgoCDComponentStatus
      rules:
        # ...
        - alert: ApplicationSetControllerNotReady 1
          annotations:
            message: >-
              applicationSet controller deployment for Argo CD instance in
              namespace "default" is not running
          expr: >-
            kube_statefulset_status_replicas{statefulset="openshift-gitops-application-controller
            statefulset",
            namespace="openshift-gitops"} !=
            kube_statefulset_status_replicas_ready{statefulset="openshift-gitops-application-
            controller statefulset",
            namespace="openshift-gitops"}
          for: 1m
          labels:
            severity: critical

```

- 1 Argo CD インスタンスによって作成されたワークロードが期待どおりに実行されているかどうかをチェックする PrometheusRule のアラートルール。

2.5.3. Argo CD カスタムリソースワークロードのモニタリングの無効化

特定の Argo CD インスタンスのワークロードモニタリングを無効にできます。ワークロードのモニタリングを無効にすると、作成された PrometheusRule が削除されます。

手順

- 特定の Argo CD インスタンスで **.spec.monitoring.enabled** フィールドの値を **false** に設定します。

Argo CD カスタムリソースの例


```
apiVersion: argoproj.io/v1beta1
kind: ArgoCD
metadata:
  name: example-argocd
  labels:
    example: repo
spec:
  # ...
  monitoring:
    enabled: false
  # ...
```

2.5.4. 関連情報

- [ユーザー定義プロジェクトのモニタリングの有効化](#)