



Red Hat Integration 2020.q1

Red Hat Integration の監視

Red Hat Integration の監視

Red Hat Integration 2020.q1 Red Hat Integration の監視

Red Hat Integration の監視

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Monitoring_Red_Hat_Integration.ent file |.

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

Prometheus（オープンソースの監視システムおよび時系列データベース）を使用して Red Hat Integration を監視します。

目次

第1章 PROMETHEUS の概要	3
1.1. PROMETHEUS クエリー	3
1.2. PROMETHEUS データの表示オプション	3
1.3. PROMETHEUS を使用した FUSE の監視	4
第2章 PROMETHEUS を使用した OPENSIFT での FUSE アプリケーションの監視	5
2.1. PROMETHEUS の設定	5
2.2. OPENSIFT 環境変数	7
2.3. PROMETHEUS が監視および収集するメトリクスの制御	7
2.4. アラートの生成	8
第3章 PROMETHEUS を使用した FUSE ONLINE の監視	9
3.1. FUSE ONLINE インフラストラクチャーコンポーネントの監視	9
3.2. FUSE ONLINE インテグレーションアプリケーションの監視	11
3.3. FUSE ONLINE アプリケーションを監視するための外部 PROMETHEUS インスタンスの設定	13
3.3.1. Prometheus インスタンスの設定 (Prometheus Operator を使用)	13
3.3.2. Prometheus インスタンスの設定 (Prometheus Operator を不使用)	14

第1章 PROMETHEUS の概要

Prometheus は、Red Hat OpenShift 環境にデプロイされたサービスの監視に使用できる、システムおよびサービスのオープンソースの監視およびアラートツールキットです。Prometheus は、指定の間隔で設定されたサービスからメトリクスを収集および保存します。さらに、ルール式の評価や結果の表示を行い、指定の条件が true になるとアラートをトリガーできます。



重要

Prometheus に対する Red Hat のサポートは、Red Hat 製品ドキュメントに記載されているセットアップと推奨設定に限定されます。

OpenShift サービスを監視するには、エンドポイントを Prometheus 形式に公開するように各サービスを設定する必要があります。このエンドポイントは、メトリクスのリストとメトリクスの現在の値を提供する HTTP インターフェースです。Prometheus は定期的にターゲット定義の各エンドポイントをスクレイピングし、収集したデータをそのデータベースに書き込みます。Prometheus は、現在実行中のセッションだけでなく、長期間にわたってデータを収集します。Prometheus は、データ上でクエリーをグラフィカルに可視化し、実行できるように、データを格納します。

1.1. PROMETHEUS クエリー

Prometheus Web インターフェースでは、[Prometheus Query Language \(PromQL\)](#) でクエリーを作成して、収集したデータを選択し、集約することができます。

たとえば、以下のクエリーを使用すると、Prometheus が直近の 5 分間に記録したすべての時系列データから、メトリクス名が **http_requests_total** である値をすべて選択することができます。

```
http_requests_total[5m]
```

クエリーの結果をさらに定義または絞り込むには、メトリクスのラベル (**key:value** ペア) を指定します。たとえば、以下のクエリーを使用すると、Prometheus が直近の 5 分間に記録したすべて時系列データから、メトリクス名が **http_requests_total** かつジョブラベルが **integration** に設定されている値をすべて選択することができます。

```
http_requests_total{job="integration"}[5m]
```

1.2. PROMETHEUS データの表示オプション

Prometheus がクエリーの結果を処理する方法を指定できます。

- Prometheus データを表データとして Prometheus の式ブラウザーで表示します。
- [Prometheus HTTP API](#) 経由で外部システムで Prometheus データを消費します。
- Prometheus データをグラフで表示します。
Prometheus は、収集するデータのデフォルトグラフィカルビューを提供します。Prometheus メトリクスを表示するより堅牢なグラフィカルダッシュボードが必要な場合は、Grafana を選択するのが一般的です。



注記

Grafana はコミュニティがサポートする機能です。Grafana をデプロイして Red Hat の製品を監視する構成は、Red Hat の実稼働環境におけるサービスレベルアグリーメント (SLA) の対象外です。

PromQL 言語を使用して、[Prometheus の Alertmanager ツール](#) で通知を設定することもできます。

1.3. PROMETHEUS を使用した FUSE の監視

Prometheus を使用して以下のように Fuse サービスを監視することができます。

OpenShift 上の Fuse アプリケーション

Fuse on OpenShift は、OpenShift でインテグレーションアプリケーションを実行するための Fuse ディストリビューションです (Red Hat Enterprise Linux オペレーティングシステムでサポートされます)。オンプレミスの OpenShift クラスターまたは Minishift や Red Hat Container Development Kit などの単一ノードクラスターで Prometheus を使用できます。Fuse on OpenShift の詳細は、『[Fuse on OpenShift Guide](#)』を参照してください。

Fuse Online インフラストラクチャーコンポーネントおよび Fuse Online インテグレーションアプリケーション

Fuse Online は、ブラウザベースの UI を使用して簡単なワークフローにアクセスできる、展開されていないインテグレーターインテグレーターの Fuse ディストリビューションです。Fuse Online の詳細は、『[Fuse Online でのアプリケーションの統合](#)』を参照してください。



注記

- Prometheus を使用して OpenShift Online 上の Fuse アプリケーションを監視することはサポートされません。
- Grafana はコミュニティがサポートする機能です。Grafana をデプロイして Red Hat Integration 製品を監視する構成は、Red Hat の実稼働環境におけるサービスレベルアグリーメント (SLA) の対象外です。

関連情報

Fuse on OpenShift でインストールおよび開発を行うための情報は、『[Fuse on OpenShift ガイド](#)』を参照してください。

第2章 PROMETHEUS を使用した OPENSIFT での FUSE アプリケーションの監視

2.1. PROMETHEUS の設定

Prometheus を設定するには、クラスターに Prometheus Operator のカスタムリソース定義をインストールし、Prometheus を Fuse アプリケーションが含まれる OpenShift プロジェクトに追加します。

前提条件

- OpenShift クラスターにアクセス可能な **cluster admin** 権限が必要です。
- 『[Fuse on OpenShift Guide](#)』の説明にしたがって、Fuse on OpenShift イメージおよびテンプレートをインストールして OpenShift クラスターが準備されている必要があります。
- クラスターで OpenShift プロジェクトを作成し、Fuse アプリケーションをそのプロジェクトに追加されている必要があります。

手順

1. 管理者権限で OpenShift にログインします。

```
$ oc login -u system:admin
```

2. Prometheus Operator の実行に必要なカスタムリソース定義をインストールします。 **{\$templates-base-url}** は Fuse on OpenShift テンプレートファイルの場所に置き換えます。

```
$ oc create -f {$templates-base-url}/fuse-prometheus-crd.yml
```

Prometheus Operator がクラスターのすべての namespace で利用できるようになります。

3. 以下のコマンド構文を使用して、Prometheus Operator を namespace にインストールします。

```
$ oc process -f {$templates-base-url}/fuse-prometheus-operator.yml -p NAMESPACE=  
<YOUR NAMESPACE> | oc create -f -
```

たとえば、**myproject** という名前のプロジェクト (namespace) には、以下のコマンドを使用します。

```
oc process -f {$templates-base-url}/fuse-prometheus-operator.yml -p  
NAMESPACE=myproject | oc create -f -
```



注記

Prometheus Operator を namespace に初めてインストールする場合、Prometheus リソース Pod が起動するまでに数分の時間がかかる場合があります。その後、Prometheus Operator をクラスターの他の namespace にインストールすると、Prometheus リソース Pod の起動ははるかに速くなります。

4. 以下のコマンド構文を使用して、Prometheus Operator を指示し、プロジェクトの Fuse アプリケーションを監視します。

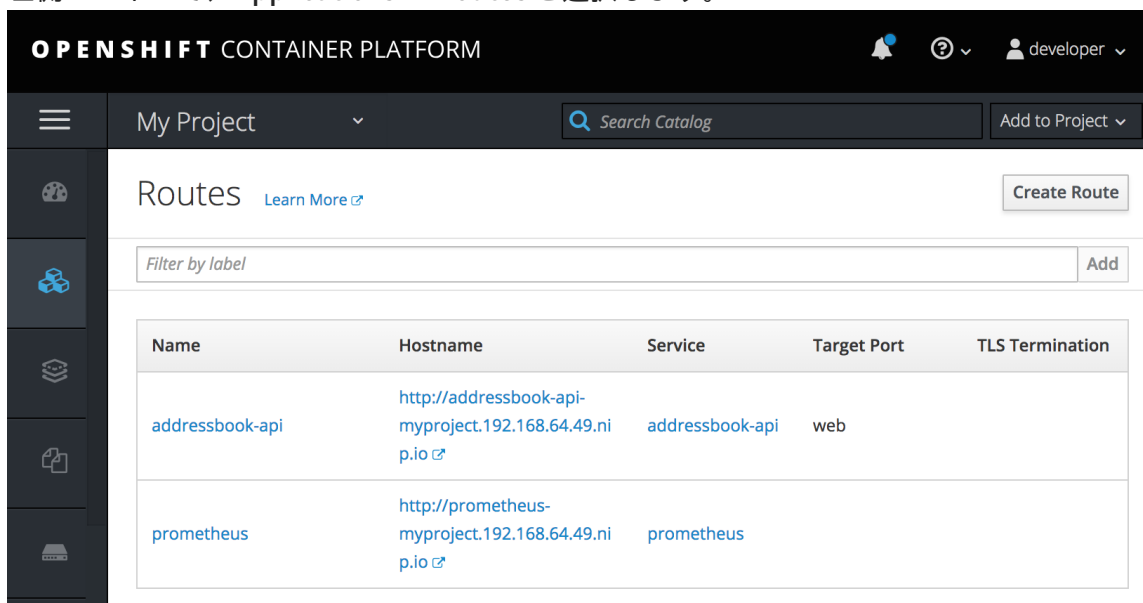
```
$ oc process -f {$templates-base-url}/fuse-servicemonitor.yml -p NAMESPACE=<YOUR NAMESPACE> -p FUSE_SERVICE_NAME=<YOUR FUSE SERVICE> | oc apply -f -
```

たとえば、**myfuseapp** という名前の Fuse アプリケーションが含まれる **myproject** という名前の OpenShift プロジェクト (namespace) には、次のコマンドを使用します。

```
oc process -f {$templates-base-url}/fuse-servicemonitor.yml -p NAMESPACE=myproject -p FUSE_SERVICE_NAME=myfuseapp | oc apply -f -
```

5. Prometheus ダッシュボードを開くには、以下を実行します。

- OpenShift コンソールにログインします。
- Prometheus を追加したプロジェクトを開きます。
- 左側のペインで、**Applications → Routes** と選択します。



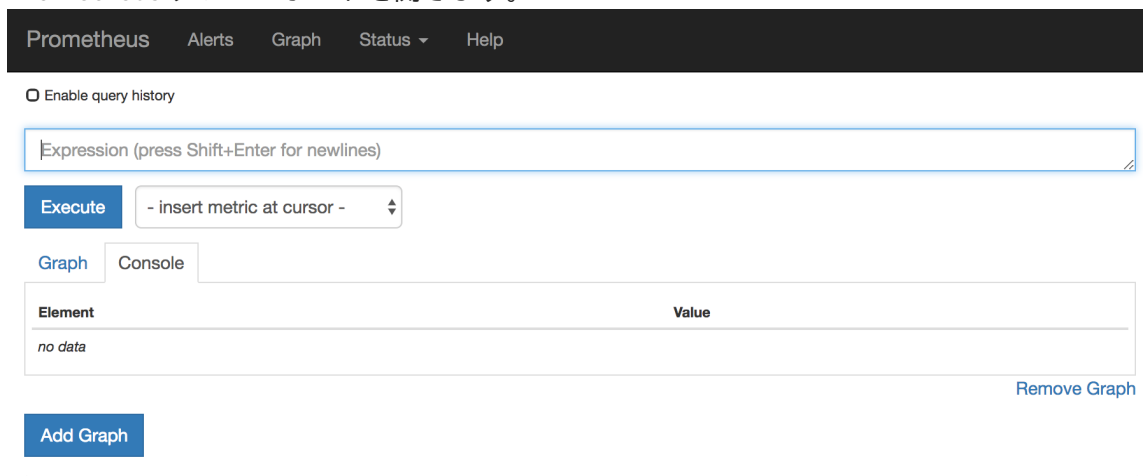
OPENSIFT CONTAINER PLATFORM

My Project

Routes [Learn More](#)

Name	Hostname	Service	Target Port	TLS Termination
addressbook-api	http://addressbook-api-myproject.192.168.64.49.nip.io	addressbook-api	web	
prometheus	http://prometheus-myproject.192.168.64.49.nip.io	prometheus		

- Prometheus Hostname URL をクリックし、新しいブラウザタブまたはウィンドウで Prometheus ダッシュボードを開きます。



Prometheus Alerts Graph Status Help

Enable query history

Graph Console

Element	Value
no data	

- e. Prometheus を初めて使用する場合は、
「https://prometheus.io/docs/prometheus/latest/getting_started/」を参照してください。

2.2. OPENSIFT 環境変数

アプリケーションの Prometheus インスタンスを設定するには、表2.1「Prometheus 環境変数」にリストされている OpenShift 環境変数を設定します。

表2.1 Prometheus 環境変数

環境変数	説明	デフォルト
AB_PROMETHEUS_HOST	バインドするホストアドレス。	0.0.0.0
AB_PROMETHEUS_OFF	設定されている場合、Prometheus のアクティベートを無効にします (空の値をエコーします)。	Prometheus が有効。
AB_PROMETHEUS_PORT	使用するポート。	9779
AB_JMX_EXPORTER_CONFIG	ファイル (パスを含む) を Prometheus 設定ファイルとして使用します。	Camel メトリクスが含まれる /opt/prometheus/prometheus-config.yml ファイル。
AB_JMX_EXPORTER_OPTS	JMX エクスポート設定に追加する追加オプション。	該当なし

その他のリソース

Pod の環境変数の設定に関する詳細は、『[OpenShift 開発者ガイド](https://access.redhat.com/documentation/ja-jp/openshift_container_platform/3.11/html/developer_guide/index)』 (https://access.redhat.com/documentation/ja-jp/openshift_container_platform/3.11/html/developer_guide/index) を参照してください。

2.3. PROMETHEUS が監視および収集するメトリクスの制御

デフォルトでは、Prometheus は Camel によって公開される可能性のあるすべてのメトリクスが含まれる設定ファイル (<https://raw.githubusercontent.com/jboss-fuse/application-templates/master/prometheus/prometheus-config.yml>) を使用します。

Prometheus が監視および収集するアプリケーション内にカスタムメトリクスがある場合 (アプリケーションプロセスを実行するオーダーの数など)、独自の設定ファイルを使用できます。識別できるメトリクスは、JMX で提供されるメトリクスに限定されるため注意してください。

手順

カスタム設定ファイルを使用して、デフォルトの Prometheus 設定では対応されない JMX Bean を公開するには、以下の手順に従います。

1. カスタム Prometheus 設定ファイルを作成します。デフォルトファイルのコンテンツ ([prometheus-config.yml](https://raw.githubusercontent.com/jboss-fuse/application-templates/master/prometheus/prometheus-config.yml) <https://raw.githubusercontent.com/jboss-fuse/application-templates/master/prometheus/prometheus-config.yml>) を形式の目安として使用できます。

カスタム設定ファイルには任意の名前を使用できます (例: **my-prometheus-config.yml** など)。

2. prometheus 設定ファイル (例: **my-prometheus-config.yml**) をアプリケーションの **src/main/fabric8-includes** ディレクトリーに追加します。
3. アプリケーション内に **src/main/fabric8/deployment.xml** ファイルを作成し、設定ファイルに設定された値で **AB_JMX_EXPORTER_CONFIG** 環境変数のエントリーを追加します。以下に例を示します。

```
spec:
  template:
    spec:
      containers:
      -
        resources:
          requests:
            cpu: "0.2"
          limits:
            cpu: "1.0"
        env:
        - name: SPRING_APPLICATION_JSON
          value: '{"server":{"tomcat":{"max-threads":1}}}'
        - name: AB_JMX_EXPORTER_CONFIG
          value: "my-prometheus-config.yml"
```

この環境変数は、Pod レベルでアプリケーションに適用されます。

4. アプリケーションの再構築およびデプロイ

2.4. アラートの生成

OpenShift に Prometheus を使用してアラートを生成する例は、Red Hat Cloud Forms の『**モニタリング、アラート、およびレポート**』を参照してください。

https://access.redhat.com/documentation/ja-jp/red_hat_cloudforms/4.7/html/monitoring_alerts_and_reporting/integrating_prometheus_alerts

第3章 PROMETHEUS を使用した FUSE ONLINE の監視

以下のセクションで説明されているように、Prometheus を使用して Fuse Online インフラストラクチャーおよび Fuse Online アプリケーションを監視できます。

- [「Fuse Online インフラストラクチャーコンポーネントの監視」](#)
- [「Fuse Online インテグレーションアプリケーションの監視」](#)

本セクションで説明されているように、外部 Prometheus インスタンスを設定して Fuse Online アプリケーションを監視することができます。

- [「Fuse Online アプリケーションを監視するための外部 Prometheus インスタンスの設定」](#)

3.1. FUSE ONLINE インフラストラクチャーコンポーネントの監視

Prometheus を使用して、以下の Fuse Online インフラストラクチャーコンポーネントによって公開されるメトリクスを監視できます。

Syndesis Server

syndesis-server コンポーネントは Micrometer でインストルメント化され、デフォルトですべての JVM メトリクス Micrometer を自動的に公開します。さらに、**syndesis-server** は要求レート、エラーレート、レイテンシーなどの REST API エンドポイントに関するメトリクスを公開します。

Syndesis Meta

syndesis-meta コンポーネントは Micrometer でインストルメント化され、デフォルトですべての JVM メトリクス Micrometer を自動的に公開します。また、REST API エンドポイントについてのメトリクスも公開します。

Syndesis DB

Fuse Online Postgres データベースのメトリクスは、サードパーティーの Prometheus エクスポートャーを使用してエクスポートされます。

統合

インテグレーション メトリクスは、デフォルトで複数の JVM メトリクスを公開する公式の JMX エクスポートャーを使用してエクスポートされます。さらに、integration メトリクスは、メッセージレートやエラーレートなどの Apache Camel 固有のメトリクスを公開します。

Grafana ダッシュボードを使用して、Prometheus によって収集されるメトリクスを可視化することもできます。

前提条件

- OpenShift クラスターにアクセス可能な **cluster admin** 権限が必要です。
- [これらのインストール手順](#) にしたがって、Application Monitoring Operator（具体的にタグ **v1.1.6** タグ）で Prometheus および Grafana をデプロイします。



注記

Grafana はコミュニティがサポートする機能です。Grafana をデプロイして Red Hat Integration 製品を監視する構成は、Red Hat の実稼働環境におけるサービスレベルアグリーメント(SLA)の対象外です。

手順

1. Fuse Online namespace で以下のコマンドを使用して **monitoring-key=middleware** ラベルを設定します。

```
oc label namespace <fuse-online-namespace> monitoring-key=middleware
```

2. Fuse Online インストールでアプリケーション監視設定リソースが OpenShift クラスターに追加されたことを確認します。
 - a. OpenShift Web コンソールで、**applicaton-monitoring** プロジェクトに移動し、**prometheus-route** URL を開きます。
 - b. Prometheus コンソールで、**Status** → **Targets** ページに移動します。
Syndesis ターゲットがリストされている場合は、Fuse Online が監視用に設定されているため、ステップ 4 に進みます。

Syndesis ターゲットが一覧にない場合は、ステップ 3 に進みます。

3. インフラストラクチャーリソースが OpenShift クラスターにない場合、Prometheus モニタリングを有効にするには、以下を実行します。
 - a. Fuse Online の namespace に移動します。

OpenShift 4.x	OpenShift 3.11
<ol style="list-style-type: none"> i. OpenShift Web コンソールで、Fuse Online (syndesis) プロジェクトに移動します。 ii. Catalog > Installed Operators の順に選択し、Fuse Online Operator をクリックします。 iii. Syndesis CRD をクリックし、app をクリックします。 iv. Yaml をクリックし、エディターで yaml ファイルを開きます。 	<ol style="list-style-type: none"> i. Resources > Other Resources の順に選択します。 ii. ドロップダウンメニューから、Syndesis リソースタイプを選択します。 iii. app リソースの Actions をクリックし、Edit YAML を選択してエディターで yaml ファイルを開きます。

- b. **yaml** ファイルを編集し、以下の行を追加して **Syndesis.Spec.Addons.Ops.Enable** 値を **true** に設定します。

```
spec:
  addons:
    ops:
      enabled: "true"
```

- c. ファイルを保存します。
- d. Pod が再起動するまで待機します。



注記

インフラストラクチャーリソースは、インストールコマンドの実行直後には使用できません。**Prometheus Targets** ページに Fuse Online (Syndesis) ターゲットが表示されるまで待機する必要がある場合があります。

4. Prometheus にアクセスするには、以下を実行します。
 - a. アプリケーション監視 Operator がインストールされているプロジェクトの OpenShift コンソールで、ルートの一覧を開きます。
 - b. **prometheus-route** エントリーの横にあるホスト名 URL をクリックし、Prometheus コンソールを開きます。
 - c. Fuse Online インフラストラクチャーコンポーネントに設定されたアラートルールの一覧を表示するには、**Alerts** メニュー項目をクリックします。
5. Grafana ダッシュボードにアクセスするには、以下を実行します。
 - a. アプリケーション監視 Operator がインストールされているプロジェクトの OpenShift コンソールで、ルートの一覧を開きます。
 - b. **grafana-route** エントリーの横にあるホスト名 URL をクリックし、Grafana コンソールを開きます。
 - c. Grafana コンソールの上部で、ダッシュボードセクターをクリックし、以下のインフラストラクチャーダッシュボードのいずれかを選択します。

Infrastructure - DB

Fuse Online の Postgres インスタンスに関連するメトリクスを表示します。

Infrastructure - JVM

syndesis-meta または **syndesis-server** アプリケーションの実行中の JVM に関するメトリクスを表示します。ダッシュボードの上部にある **Application** ドロップダウンリストから、監視するアプリケーションを選択します。

Infrastructure - REST APIs

request throughput および **latency** などの Fuse Online インフラストラクチャー API エンドポイントに関連するメトリクスを表示します。ダッシュボードの上部にある **Application** ドロップダウンリストから、監視するアプリケーションを選択します。

6. Prometheus Alertmanager にアクセスするには、以下を実行します。
 - a. アプリケーション監視 Operator がインストールされているプロジェクトの OpenShift コンソールで、ルートの一覧を開きます。
 - b. **alertmanager-route** エントリーの横にあるホスト名 URL をクリックし、Alertmanager コンソールを開きます。
Fuse Online インフラストラクチャーが正常である場合、デフォルトのビューは空になります。

インフラストラクチャーコンポーネントのいずれかが正常でない場合は、生成されたアクティブなアラートとそれらのアラートの通知を無効にするオプションが表示されます。

3.2. FUSE ONLINE インテグレーションアプリケーションの監視

Application Monitoring Operator を使用して Prometheus をデプロイし、Fuse Online インテグレーションアプリケーションを監視できるようにします。Grafana ダッシュボードを使用して、Prometheus によって収集されるメトリクスを可視化することもできます。

前提条件

- OpenShift クラスターにアクセス可能な **cluster admin** 権限が必要です。

- これらのインストール手順にしたがって、Application Monitoring Operator（具体的にタグ **v1.1.6** タグ）で Prometheus および Grafana をデプロイします。



注記

Grafana はコミュニティがサポートする機能です。Grafana をデプロイして Red Hat Integration 製品を監視する構成は、Red Hat の実稼働環境におけるサービスレベルアグリーメント (SLA) の対象外です。

手順

1. Fuse Online namespace で以下のコマンドを使用して **monitoring-key=middleware** ラベルを設定します。

```
oc label namespace <fuse-online-namespace> monitoring-key=middleware
```

2. Fuse Online インストールでアプリケーション監視設定リソースが OpenShift クラスターに追加されたことを確認します。

- a. OpenShift Web コンソールで、**applicaton-monitoring** プロジェクトに移動し、**prometheus-route** URL を開きます。
- b. Prometheus コンソールで、**Status > Targets** ページに移動します。Syndesis ターゲットがリストされている場合は、監視用に Fuse Online が設定され、ステップ 4 に進みます。

Syndesis ターゲットがリストされていない場合は、ステップ 3 に進みます。

3. インフラストラクチャーリソースが OpenShift クラスターにない場合、Prometheus モニタリングを有効にするには、以下を実行します。

- a. Fuse Online の namespace に移動します。

OpenShift 4.x	OpenShift 3.11
<ol style="list-style-type: none"> i. OpenShift Web コンソールで、Fuse Online (syndesis) プロジェクトに移動します。 ii. Catalog > Installed Operators の順に選択し、Fuse Online Operator をクリックします。 iii. Syndesis CRD をクリックし、app をクリックします。 iv. Yaml をクリックし、エディターで yaml ファイルを開きます。 	<ol style="list-style-type: none"> i. Resources > Other Resources の順に選択します。 ii. ドロップダウンメニューから、Syndesis リソースタイプを選択します。 iii. app リソースの Actions をクリックし、Edit YAML を選択してエディターで yaml ファイルを開きます。

- b. **yaml** ファイルを編集し、以下の行を追加して **Syndesis.Spec.Addons.Ops.Enable** 値を **true** に設定します。

```
spec:
  addons:
    ops:
```


enabled: "true"

- c. ファイルを保存します。
- d. Pod が再起動するまで待機します。



注記

インフラストラクチャーリソースは、インストールコマンドの実行直後には使用できません。**Prometheus Targets** ページに Fuse Online (Syndesis) ターゲットが表示されるまで待機する必要がある場合があります。

4. Prometheus にアクセスするには、以下を実行します。
 - a. アプリケーション監視 Operator がインストールされているプロジェクトの OpenShift コンソールで、ルートの一覧を開きます。
 - b. **prometheus-route** エントリーの横にあるホスト名 URL をクリックし、Prometheus コンソールを開きます。
 - c. Fuse Online インフラストラクチャーコンポーネントに設定されたアラートルールの一覧を表示するには、**Alerts** メニュー項目をクリックします。
5. Grafana ダッシュボードにアクセスするには、以下を実行します。
 - a. アプリケーション監視 Operator がインストールされているプロジェクトの OpenShift コンソールで、ルートの一覧を開きます。
 - b. **grafana-route** エントリーの横にあるホスト名 URL をクリックし、Grafana コンソールを開きます。
 - c. Grafana コンソールの上部で、ダッシュボードセクターをクリックし、**Integration - Camel** を選択します。
このダッシュボードには、Apache Camel インテグレーションアプリケーションによって公開される標準メトリクスが表示されます。

3.3. FUSE ONLINE アプリケーションを監視するための外部 PROMETHEUS インスタンスの設定

OpenShift Container Platform 3.11 に Fuse Online をインストールする場合、デフォルトで **syndesis-prometheus** が含まれます。Fuse Online をオンプレミスの OpenShift Container Platform にインストールする方法の詳細は、『[Integrating Applications with Fuse Online](#)』を参照してください。

外部の Prometheus インスタンスがすでに存在する場合、その外部のインスタンスを設定して、OpenShift Container Platform にデプロイされた Fuse Online インテグレーションアプリケーションを監視することもできます。

手順は、Prometheus Operator を使用して外部の Prometheus インスタンスをインストールしたかどうかによって異なります。

- [「Prometheus インスタンスの設定 \(Prometheus Operator を使用\)」](#)
- [「Prometheus インスタンスの設定 \(Prometheus Operator を不使用\)」](#)

3.3.1. Prometheus インスタンスの設定 (Prometheus Operator を使用)

Prometheus Operator を使用して Prometheus インスタンスをインストールした場合、外部 Prometheus 設定を更新して Fuse Online インテグレーションを監視するのに、サービスモニターの追加および編集が必要になります。

前提条件

「Prometheus の設定」の説明に従って Prometheus がインストールされている必要があります。

手順

1. ターミナルウィンドウで、サービスモニターを Prometheus がインストールされた namespace (プロジェクト) に追加します。

```
oc process -f {$templates-base-url}/fuse-servicemonitor.yml -p NAMESPACE=<YOUR-NAMESPACE> -p FUSE_SERVICE_NAME=fuseonline | oc create -f -
```

2. OpenShift コンソールでプロジェクトを開き、**Applications** → **Services** と選択します。
3. **fuseonline** サービスをクリックし、**Actions** → **Edit YAML** と選択します。
4. エディターで **app: fuseonline** を **syndesis.io/type: integration** に置き換えて、YAML ファイルのセクターセクションを変更します。
5. **Save** をクリックします。

Prometheus Operator が設定を更新し、Fuse Online インテグレーションをすべて監視します。

これで、Fuse Online インテグレーションを Prometheus インスタンスで表示できるようになります。以下の例は、2つのインテグレーションを監視する Prometheus インスタンスを示しています。

The screenshot shows the Prometheus web interface. At the top, there are navigation tabs: Prometheus, Alerts, Graph, Status, and Help. The main heading is 'Targets'. Below it, there are two buttons: 'All' and 'Unhealthy'. Underneath, it says 'fuseonline (2/2 up)' with a 'show less' link. A table displays the target details:

Endpoint	State	Labels	Last Scrape	Error
http://172.17.0.14:9779/metrics	UP	endpoint="web" instance="172.17.0.14:9779" namespace="myproject" pod="i-time-to-log-integration-2-pxfc9" service="fuseonline"	4.089s ago	
http://172.17.0.16:9779/metrics	UP	endpoint="web" instance="172.17.0.16:9779" namespace="myproject" pod="i-time-to-db-integration-6-cktnn" service="fuseonline"	15.073s ago	

3.3.2. Prometheus インスタンスの設定 (Prometheus Operator を不使用)

Prometheus Operator を使用せずに Prometheus インスタンスをインストールした場合、設定を更新して Fuse Online インテグレーションを監視するのに、Prometheus 設定ファイルを編集し、Prometheus Pod 設定を更新する必要があります。

前提条件

Prometheus 設定ファイル(**prometheus-config.yml**)への書き込み権限が必要です。OpenShift web コンソールでは、設定ファイルは **Resources** → **Config Maps** にあります。

手順

- Prometheus 設定ファイル(**prometheus-config.yml**)を以下のように編集します。

- スクレープ間隔を 5 秒に設定します。

```
global:
  scrape_interval: 5s
  evaluation_interval: 5s
```

- \${OPENSIFT_PROJECT}** で Pod をスクレープするよう設定する Kubernetes サービス検出設定とともに **integration-pod** という名前のスクレープ設定ジョブを追加します。これは通常 **syndesis** namespace です。

```
- job_name: integration-pods

  kubernetes_sd_configs:
  - role: pod
    namespaces:
      names:
      - ${OPENSIFT_PROJECT}
```

- 以下のような **relabel_configs** セクションを追加します。

- prometheus.io/scrape** ラベルが **true** に設定されたインテグレーション Pod のみをスクレープします。
- prometheus.io/path** および **prometheus.io/port** ラベルから値を使用して、インテグレーション Pod で JMX エクスポートを収集するために使用される **metrics_path** および **address** ラベルを設定します。
- Pod ラベルおよびアノテーションを Prometheus レベルとして追加する。
- kubernetes_namespace** および **kubernetes_pod_name** ラベルを作成します。**relabel_configs** セクションの例は次のとおりです。

```
relabel_configs:
  - source_labels:
    [__meta_kubernetes_pod_annotation_prometheus_io_scrape]
    action: keep
    regex: true
  - source_labels: [__meta_kubernetes_pod_annotation_prometheus_io_path]
    action: replace
    target_label: __metrics_path__
    regex: (.+)
  - source_labels: [__address__,
    __meta_kubernetes_pod_annotation_prometheus_io_port]
    action: replace
    regex: ([^:]+)(?::\d+)?;\d+
    replacement: $1:$2
    target_label: __address__
  - action: labelmap
    regex: __meta_kubernetes_pod_label_(.+)
```

```

- action: labelmap
  regex: __meta_kubernetes_pod_annotation_(syndesis.+)+
- source_labels: [__meta_kubernetes_namespace]
  action: replace
  target_label: kubernetes_namespace
- source_labels: [__meta_kubernetes_pod_name]
  action: replace
  target_label: kubernetes_pod_name

```

- d. Fuse Online インテグレーションは、JVM、Camel、および CXF から多数のメトリクスを公開します。メトリクスに必要なストレージの量を減らすには、以下の **metric_relabel_configs** セクションを追加し、Fuse Online コンソールに表示されないメトリクスを除外します。

```

metric_relabel_configs:
  - source_labels: [__name__]
    regex: jmx_(.+)+
    action: drop
  - source_labels: [__name__]
    regex: jvm_(.+)+
    action: drop
  - source_labels: [__name__]
    regex: process_(.+)+
    action: drop
  - source_labels: [type, __name__]
    separator: ':'
    regex: context:
      (org_apache_camel_ExchangesTotal|org_apache_camel_ExchangesFailed|io_syndesis_ca
      mel_StartTimestamp|io_syndesis_camel_LastExchangeCompletedTimestamp|io_syndesis_
      camel_LastExchangeFailureTimestamp)
    action: keep

```



注記

最後の設定行は、Fuse Online web コンソールに表示される統計に不可欠な Prometheus メトリクスストアに追加するメトリクスを明示的にリストしています。Prometheus インスタンスは、他のメトリクスが除外された場合にこれらのメトリクスの収集を明示的に許可する **必要があります**。

2. 以下のように、Prometheus Pod 設定が 30 日分のメトリクスを保存するよう更新します。

```

args:
  - '--config.file=/etc/prometheus/prometheus.yml'
  - '--storage.tsdb.retention=30d'

```