OpenShift Dedicated 4

クラスター管理

OpenShift Dedicated クラスターの設定

Last Updated: 2022-12-04

OpenShift Dedicated 4 クラスター管理

OpenShift Dedicated クラスターの設定

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

概要

このドキュメントでは、OpenShift Dedicated クラスターの設定に関する情報を提供します。

目次

第1章 管理ロールおよびユーザーの管理

1.1. 管理ロールについて

1.1.1. cluster-admin ロール

Customer Cloud Subscriptions (CCS) のある OpenShift Dedicated クラスターの管理者として、**cluster-admin** ロールにアクセスできます。クラスターを作成したユーザーとして、**cluster-admin** ユーザーロールをアカウントに追加して、最大管理者権限を割り当てます。これらの権限は、クラスターの作成時に自動的にユーザーアカウントに割り当てられることはありません。cluster-admin ロールを持つアカウントにログインしている場合、ユーザーはクラスターを制御し、設定するための多数の無制限のアクセスを持ちます。クラスターの不安定化を防ぐため、または OpenShift Cluster Manager Hybrid Cloud Console で管理されており、クラスター内の変更が上書きされるために、Webhook でブロックされる設定がいくつかあります。cluster-admin ロールの使用には、Red Hat との Appendix 4 契約に記載されている制限が適用されます。ベストプラクティスとして、**cluster-admin** ユーザーの数をできるだけ少なく制限できます。

1.1.2. dedicated-admin ロール

OpenShift Dedicated クラスターの管理者のアカウントには、追加のパーミッションがあり、組織のクラスター内のユーザーが作成したすべてのプロジェクトにアクセスできます。**dedicated-admin** ロールでアカウントにログインしている場合、開発者 CLI コマンド (**oc** コマンド) を使用すると、プロジェクト全体でのオブジェクトの可視性と管理機能を強化することができますが、管理者 CLI コマンド (**oc adm** コマンド下のコマンド) を使用するとさらに多くの操作を実行できます。

注記

ご使用のアカウントにはこれらの追加されたパーミッションがあるものの、実際のクラスターのメンテナーンスおよびホスト設定は依然として OpenShift Operations Team によって実行されます。

1.2. OPENSHIFT DEDICATED 管理者の管理

管理者ロールは、クラスター上の **cluster-admin** または **dedicated-admin** グループを使用して管理されます。このグループの既存のメンバーは、OpenShift Cluster Manager Hybrid Cloud Console を介してメンバーシップを編集できます。

1.2.1. ユーザーの追加

手順

1. **Cluster Details** ページおよび **Access Control** タブに移動します。
2. **Add user** ボタンをクリックします (最初のユーザーのみ)。
3. ユーザー名を入力し、グループを選択します。
4. **Add** ボタンをクリックします。

注記

ユーザーを **cluster-admin** グループに追加すると、完了するまでに数分かかる場合があります。

注記

既存の **dedicated-admin** ユーザーは、**cluster-admin** グループに追加することはできません。最初に、ユーザーを **cluster-admin** グループに追加する前に、**dedicated-admin** グループからユーザーを削除する必要があります。

1.2.2. ユーザーの削除

手順

1. **Cluster Details** ページおよび **Access Control** タブに移動します。
2. ユーザーおよびグループの組み合わせの右側にある Options メニュー をクリックし、**Delete** をクリックします。

第2章 プライベート接続の設定

2.1. AWS のプライベート接続の設定

2.1.1. AWS クラウドインフラストラクチャーのアクセスについて

注記

AWS クラウドインフラストラクチャーアクセスは、クラスターの作成時に選択した Customer Cloud Subscription (CCS) インフラストラクチャータイプには適用されません。これは、CCS クラスターがアカウントにデプロイされるためです。

Amazon Web Services (AWS) インフラストラクチャーへのアクセスにより、カスタマーポータルの組織管理者 およびクラスターの所有者は AWS の Identity and Access Management (IAM) ユーザーに OpenShift Dedicated クラスターの AWS 管理コンソールへのフェデレーションアクセスを持たせることができます。AWS アクセスはカスタマー AWS ユーザーに付与でき、OpenShift Dedicated 環境のニーズに合わせてプライベートクラスターのアクセスを実装できます。

1. OpenShift Dedicated クラスターの AWS インフラストラクチャーアクセスの設定を開始します。AWS ユーザーおよびアカウントを作成し、そのユーザーに OpenShift Dedicated AWS アカウントへのアクセスを提供します。

2. OpenShift Dedicated AWS アカウントへのアクセスを取得した後に、以下の方法のいずれかを使用してクラスターへのプライベート接続を確立します。

   - AWS VPC ピアリングの設定: VPC ピアリングを有効にして、2 つのプライベート IP アドレス間のネットワークトラフィックをルーティングします。

   - AWS VPN の設定: プライベートネットワークを Amazon Virtual Private Cloud にセキュアに接続するために、仮想プライベートネットワークを確立します。

   - AWS Direct Connect の設定: プライベートネットワークと AWS Direct Connect の場所との間に専用のネットワーク接続を確立するように AWS Direct Connect を設定します。

クラウドインフラストラクチャーアクセスの設定後に、プライベートクラスターの設定について確認してください。

2.1.2. AWS インフラストラクチャーアクセスの設定

Amazon Web Services (AWS) インフラストラクチャーへのアクセスにより、カスタマーポータルの組織管理者 およびクラスターの所有者は AWS の Identity and Access Management (IAM) ユーザーに OpenShift Dedicated クラスターの AWS 管理コンソールへのフェデレーションアクセスを持たせることができます。管理者は、**Network Management** または **Read-only** アクセスのオプションを選択できます。

前提条件

- IAM パーミッションを持つ AWS アカウント。

手順

1. AWS アカウントにログインします。必要な場合は、AWS ドキュメント に従って新規 AWS アカウントを作成できます。

2. AWS アカウントで **STS:AllowAssumeRole** パーミッションを持つ IAM ユーザーを作成します。

   a. AWS 管理コンソールの IAM ダッシュボード を開きます。

   b. **Policies** セクションで、 **Create Policy** をクリックします。

   c. **JSON** タブを選択し、既存のテキストを以下に置き換えます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "*"
    }
  ]
}
```

   d. **Next:Tags** をクリックします。

   e. オプション: タグを追加します。**Next:Review** をクリックします。

   f. 適切な名前および説明を指定してから **Create Policy** をクリックします。

   g. **Users** セクションで、**Add plan** をクリックします。

   h. 適切なユーザー名を指定します。

   i. AWS アクセスタイプとして **AWS Management Console access** を選択します。

   j. 組織に必要なパスワード要件を調整してから **Next:Permissions** をクリックします。

   k. **Attach existing policies directly** オプションをクリックします。直前の手順で作成したポリシーを検索し、確認します。

注記

パーミッションの境界を設定することは推奨されていません。

   l. **Next: Tags** をクリックしてから **Next: Review** をクリックします。設定が正しいことを確認します。

   m. **Create user** をクリックすると、成功ページが表示されます。

   n. IAM ユーザーの Amazon Resource Name (ARN) を収集します。ARN の形式は **arn:aws:iam::000111222333:user/username** のようになります。**Close** をクリックします。

3. ブラウザーで OpenShift Cluster Manager Hybrid Cloud Console を開き、AWS インフラストラクチャーアクセスを許可するクラスターを選択します。

4. **Access control** タブを選択し、**AWS Infrastructure Access** セクションにスクロールします。

5. **AWS IAM ARN**を貼り付け、**Network Management** または **Read-only** パーミッションを選択してから **Grant role** をクリックします。
6. **AWS OSD console URL** をクリップボードにコピーします。
7. アカウント ID またはエイリアス、IAM ユーザー名、およびパスワードを使用して AWS アカウントにサインインします。
8. 新規のブラウザータブで、AWS Switch Role ページにルート指定するために使用される AWS OSD Console URL を貼り付けます。
9. アカウント番号とロールはすでに入力されています。必要な場合は表示名を選択してから **Switch Role** をクリックします。

検証

- これで、**VPC** が **Recently visited services** の下に表示されます。

2.1.3. AWS VPC ピアリングの設定

Virtual Private Cloud (VPC) ピアリング接続は、2 つの VPC 間のネットワーク接続で、プライベート IPv4 アドレスまたは IPv6 アドレスを使用してこれらの間のトラフィックをルーティングできるようにします。OpenShift Dedicated クラスターを含む Amazon Web Services (AWS) VPC を別の AWS VPC ネットワークとピア接続するように設定できます。

警告

クラスターがインストールされている VPC がピアリングされている場合、プライベートクラスターは Red Hat OpenShift Cluster Manager によって完全に削除することができません。

AWS は、中国を除く すべての商業地域でのリージョン間の VPC ピアリングをサポートします。

前提条件

- ピアリング要求を開始するために必要な Customer VPC に関する以下の情報を収集します。
  - Customer AWS アカウント番号
  - Customer VPC ID
  - Customer VPC リージョン
  - Customer VPC CIDR
- OpenShift Dedicated Cluster VPC で使用される CIDR ブロックを確認します。Customer VPC の CIDR ブロックとの重複や一致がある場合、これらの 2 つの VPC 間のピアリングは実行できません。詳細は、Amazon VPC の サポートされていない VPC ピアリング設定 に関するドキュメントを参照してください。CIDR ブロックが重複しない場合は、以下の手順を実行できます。

手順

1. VPC ピアリング要求を開始します。
2. VPC ピアリング要求を受け入れます。
3. VPC ピアリング接続の Route テーブルを更新します。

関連情報

- 詳細およびトラブルシューティングのヘルプは、AWS VPC ガイドを参照してください。

2.1.4. AWS VPN の設定

Amazon Web Services (AWS) OpenShift Dedicated クラスターを、お客様のオンサイトのハードウェア仮想プライベートネットワーク (VPN) デバイスを使用するように設定できます。デフォルトで、AWS Virtual Private Cloud (VPC) に起動するインスタンスは、独自の (リモート) ネットワークと通信できません。AWS Site-to-Site VPN 接続を作成し、ルーティングを設定して接続経由でトラフィックを渡すことで、VPC からリモートネットワークへのアクセスを有効にできます。

注記

AWS VPN は現在、NAT を VPN トラフィックに適用するための管理オプションを提供しません。詳細は、AWS Knowledge Center を参照してください。

プライベート接続を使用したすべてのトラフィックのルーティング (**0.0.0.0/0** など) はサポートされていません。これには、SRE 管理トラフィックを無効にするインターネットゲートウェイを削除する必要があります。

前提条件

- ハードウェア VPN ゲートウェイデバイスモデルおよびソフトウェアバージョン (例: バージョン 8.3 を実行している Cisco ASA)。AWS ドキュメント を参照して、お使いのゲートウェイデバイスが AWS でサポートされているかどうかを確認します。
- VPN ゲートウェイデバイスのパブリックな静的 IP アドレス。
- BGP または静的ルーティング: BGP の場合は、ASN が必要です。静的ルーティングの場合は、1 つ以上の静的ルートを設定する必要があります。
- オプション: VPN 接続をテストするための到達可能なサービスの IP およびポート/プロトコル。

手順

1. VPN 接続を設定するために カスタマーゲートウェイを作成 します。
2. 仮想プライベートゲートウェイが目的の VPC に割り当てられていない場合は、仮想プライベートゲートウェイを 作成して割り当て ます。
3. ルーティングを設定し、VPN ルート伝播を有効にします。
4. セキュリティーグループを更新します。
5. サイト間 VPN 接続を確立します。

注記

VPC サブネット情報をメモします。これは、リモートネットワークとして設定に追加する必要があります。

関連情報

- 詳細およびトラブルシューティングのヘルプは、AWS VPN ガイドを参照してください。

2.1.5. AWS Direct Connect の設定

Amazon Web Services (AWS) Direct Connect では、ホストされた Virtual Interface (VIF) が Direct Connect Gateway (DXGateway) に接続されている必要があります。これは、同じまたは別のアカウントでリモート Virtual Private Cloud (VPC) にアクセスするために Virtual Gateway (VGW) または Transit Gateway に関連付けられます。

既存の DXGateway がない場合、通常のプロセスではホストされた VIF を作成し、AWS アカウントに DXGateway および VGW が作成されます。

既存の DXGateway が 1 つ以上の既存の VGW に接続されている場合は、プロセスに AWS アカウントが Association Proposal を DXGateway の所有者に送信します。DXGateway の所有者は、提案された CIDR が関連付けられているその他の VGW と競合しないようにする必要があります。

前提条件

- OpenShift Dedicated VPC の CIDR 範囲が、関連付けのあるその他の VGW と競合しないことを確認します。
- 以下の情報を入力します。
  - Direct Connect Gateway ID。
  - 仮想インターフェイスに関連付けられた AWS アカウント ID。
  - DXGateway に割り当てられた BGP ASN。オプション: Amazon のデフォルト ASN も使用できます。

手順

1. VIF を作成する か、既存の VIF を表示 して、作成する必要のあるダイレクト接続の種別を判断します。
2. ゲートウェイを作成します。
   a. Direct Connect VIF タイプが **Private** の場合は、仮想プライベートゲートウェイを作成 します。
   b. Direct Connect VIF が **Public** の場合は、Direct Connect ゲートウェイを作成 します。
3. 使用する既存のゲートウェイがある場合は、関連付けの提案を作成 し、承認のために提案を DXGateway の所有者に送信します。

警告

既存の DXGateway に接続する場合は、コスト がかかります。

関連情報

- 詳細およびトラブルシューティングのヘルプは、AWS Direct Connect ガイドを参照してください。

2.2. プライベートクラスターの設定

OpenShift Dedicated クラスターをプライベートにし、内部アプリケーションを企業ネットワーク内でホストできるようにします。さらに、プライベートクラスターは、セキュリティーを強化するために内部 API エンドポイントのみを持つように設定できます。

OpenShift Dedicated 管理者は、**OpenShift Cluster Manager** 内からパブリックおよびプライベートのクラスター設定のいずれかを選択できます。プライバシー設定は、クラスターの作成時またはクラスターの設定後に設定できます。

2.2.1. クラスター作成時のプライベートクラスターの有効化

新規クラスターの作成時にプライベートクラスター設定を有効にできます。

前提条件

- プライベートアクセスを許可するには、以下のプライベート接続を設定する必要があります。
  - VPC ピアリング
  - Cloud VPN
  - DirectConnect (AWS のみ)
  - TransitGateway (AWS のみ)
  - Cloud Interconnect (GCP のみ)

手順

1. OpenShift Cluster Manager Hybrid Cloud Console にログインします。
2. **Create cluster** → **OpenShift Dedicated** → **Create cluster** をクリックします。
3. クラスターの詳細を設定します。
4. 希望するネットワーク設定を選択する場合は、**Advanced** を選択します。
5. **Private** を選択します。

警告

Private に設定されている場合は、前提条件で説明されているように、クラウドプロバイダーにプライベート接続を設定しない限り、クラスターにアクセスできません。

6. **Create cluster** をクリックします。クラスター作成プロセスが開始され、完了するまでに 30 - 40 分かかります。

検証

- **Overview** タブの **Installing cluster** という見出しは、クラスターがインストール中であることを示し、この見出しからインストールログを確認できます。**Details** 見出しの **Status** インディケーターは、クラスターが使用できる **Ready** 状態であるかを示します。

2.2.2. 既存クラスターをプライベートにすることが可能

クラスターを作成したら、後でクラスターをプライベートにすることができます。

前提条件

- プライベートアクセスを許可するには、以下のプライベート接続を設定する必要があります。
  - VPC ピアリング
  - Cloud VPN
  - DirectConnect (AWS のみ)
  - TransitGateway (AWS のみ)
  - Cloud Interconnect (GCP のみ)

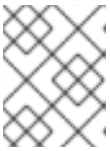
手順

1. OpenShift Cluster Manager Hybrid Cloud Console にログインします。
2. プライベートにするパブリッククラスターを選択します。
3. **Networking** タブの **Control Plane API endpoint**の下で、**Make API private** を選択します。

警告

Private に設定されている場合は、前提条件で説明されているように、クラウドプロバイダーにプライベート接続を設定しない限り、クラスターにアクセスできません。

4. **Change settings** をクリックします。



注記

クラスターをプライベートとパブリックの間で移行するには、完了までに数分の時間がかかる場合があります。

2.2.3. 既存のプライベートクラスターをパブリックにすることが可能

プライベートクラスターを作成したら、後でクラスターをパブリックにすることができます。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console にログインします。
2. パブリックにするプライベートクラスターを選択します。
3. **Networking** タブの **Control Plane API endpoint**の下で、 **Make API private** の選択を解除します。
4. **Change settings** をクリックします。

注記

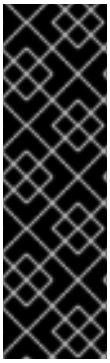
クラスターをプライベートとパブリックの間で移行するには、完了までに数分の時間がかかる場合があります。

第3章 ノード

3.1. マシンプールについて

OpenShift Dedicated は、クラウドインフラストラクチャーで柔軟性があり動的なプロビジョニング方法としてマシンプールを使用します。

プライマリーリソースは、マシン、マシンセット、およびマシンプールです。



重要

OpenShift Dedicated バージョン 4.8.35、4.9.26、4.10.6 の時点で、OpenShift Dedicated のデフォルトの Pod ごとの PID 制限は **4096** です。この PID 制限を有効にする場合は、OpenShift Dedicated クラスターをこれらのバージョン以降にアップグレードする必要があります。以前のバージョンの OpenShift Dedicated クラスターは、デフォルトの PID 制限である **1024** を使用します。

OpenShift Dedicated クラスターでは、Pod ごとの PID 制限を設定することはできません。

3.1.1. Machines

マシンは、ワーカーノードのホストを記述する基本的な単位です。

3.1.2. マシンセット

MachineSet リソースはマシンのグループです。より多くのマシンが必要な場合、またはマシンをスケールダウンする必要がある場合は、マシンセットが属するマシンプール内のレプリカの数を変更します。

3.1.3. マシンプール

マシンプールは、マシンセットの上位レベルの設定要素です。

マシンプールは、アベイラビリティーゾーン全体で同じ設定のクローンがすべて含まれるマシンセットを作成します。マシンプールは、ワーカーノードですべてのホストノードのプロビジョニング管理アクションを実行します。より多くのマシンが必要な場合、またはマシンをスケールダウンする必要がある場合は、コンピュートのニーズに合わせてマシンプール内のレプリカの数を変更してください。スケーリングは手動または自動の設定ができます。

デフォルトで、クラスターは 1 つのマシンプールを使用して作成されます。追加のマシンプールを既存クラスターに追加し、デフォルトのマシンプールを変更して、マシンプールを削除できます。

1 つのクラスターに複数のマシンプールが存在する可能性があり、それぞれが異なるタイプまたは異なるサイズのノードを持つことができます。

3.1.4. 複数のゾーンクラスターのマシンプール

複数のアベイラビリティーゾーン (Multi-AZ) クラスターにマシンプールを作成する場合は、1 つのマシンプールに 3 つのゾーンがあります。次に、マシンプールは、合計 3 つのマシンセット (クラスター内のゾーンごとに 1 つのマシンセット) を作成します。これらの各マシンセットは、それぞれのアベイラビリティーゾーンで 1 つ以上のマシンを管理します。

新しい Multi-AZ クラスターを作成すると、マシンプールはそれらのゾーンに自動的に複製されます。

マシンプールを既存の Multi-AZ に追加すると、そのゾーンに新しいプールが自動的に作成されます。同様に、マシンプールを削除するとすべてのゾーンから削除されます。この相乗効果により、Multi-AZ クラスターでマシンプールを使用すると、マシンプールを作成するときに、特定のリージョンに対するプロジェクトのクォータをより多く消費する可能性があります。

3.1.5. 関連情報

- 自動スケーリングについて

3.2. コンピュートノードの管理

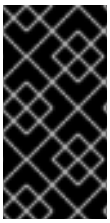
このドキュメントでは、OpenShift Dedicated を使用してコンピュート (ワーカーとも呼ばれます) ノードを管理する方法について説明します。

コンピュートノードの変更の大半は、マシンプールで設定されます。マシンプール は、管理を容易にするために、同じ設定を持つクラスター内のコンピュートノードのグループです。

スケーリング、ノードラベルの追加、テイントの追加などのマシンプール設定オプションを編集できます。

3.2.1. マシンセットの作成

OpenShift Dedicated クラスターをインストールすると、デフォルトのマシンプールが作成されます。インストール後、OpenShift Cluster Manager を使用して、クラスターの追加のマシンプールを作成できます。



重要

使用可能なコンピュート (ワーカーとも呼ばれる) ノードインスタンスタイプ、自動スケーリングオプション、およびノード数は、OpenShift Dedicated サブスクリプション、リソースクォータ、およびデプロイメントシナリオによって異なります。詳細については、営業担当者または Red Hat サポートにお問い合わせください。

前提条件

- OpenShift Dedicated クラスターを作成している。

手順

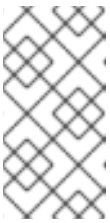
1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. **Machine pools** タブで、**Add machine pool** をクリックします。
3. **マシンプール名** を追加します。
4. ドロップダウンメニューから **Worker node instance type** を選択します。インスタンスタイプは、マシンプール内の各コンピュートノードの仮想 CPU およびメモリー割り当てを定義します。

注記

プールを作成した後に、マシンプールのインスタンスタイプを変更することはできません。

5. オプション: マシンプールの自動スケーリングを設定します。

    a. **Enable autoscaling** を選択し、デプロイメントのニーズを満たすためにマシンプール内のマシン数を自動的にスケーリングします。

    注記

    Enable autoscaling オプションは、**capability.cluster.autoscale_clusters** サブスクリプションがある場合にのみ OpenShift Dedicated で使用できます。詳細については、営業担当者または Red Hat サポートにお問い合わせください。

    b. 自動スケーリングの最小および最大のノード数制限を設定します。Cluster Autoscaler は、指定する制限を超えてマシンプールノード数を減らしたり、増やしたりできません。

    - 単一アベイラビリティーゾーンを使用してクラスターをデプロイした場合は、**最小および最大のノード数** を設定します。これは、アベイラビリティーゾーンのコンピュートノードの最小および最大の制限を定義します。

    - 複数のアベイラビリティーゾーンを使用してクラスターをデプロイした場合は、**Minimum nodes per zone** および **Maximum nodes per zone** を設定します。これは、ゾーンごとの最小および最大のコンピュート制限を定義します。

    注記

    または、マシンプールの作成後にマシンプールの自動スケーリングを設定できます。

6. 自動スケーリングを有効にしていない場合は、コンピュートノードの数を選択します。

    - 単一アベイラビリティーゾーンを使用してクラスターをデプロイした場合は、ドロップダウンメニューから **ワーカーノード数** を選択します。これは、ゾーンのマシンプールにプロビジョニングするコンピュートノードの数を定義します。

    - 複数のアベイラビリティーゾーンを使用してクラスターをデプロイした場合は、ドロップダウンメニューから **ワーカーノードの数 (ゾーンごと)** を選択します。これは、ゾーンごとにマシンプールにプロビジョニングするコンピュートノードの数を定義します。

7. オプション: マシンプールのノードラベルおよびテイントを追加します。

    a. **Edit node labels and taints** メニューを展開します。

    b. **Node labels** で、ノードラベルの **Key** および **Value** のエントリーを追加します。

    c. **Taints** で、テイントの **Key** および **Value** エントリーを追加します。

    d. テイントごとに、ドロップダウンメニューから **Effect** を選択します。使用できるオプションには、**NoSchedule**、**PreferNoSchedule**、および **NoExecute** が含まれます。

    注記

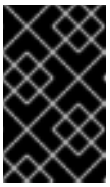
    または、マシンプールの作成後にノードラベルおよびテイントを追加できます。

8. オプション: Customer Cloud Subscription (CCS) モデルを使用して AWS に OpenShift

Dedicated をデプロイした際に、保証されていない AWS スポットインスタンスとしてマシンをデプロイするようにマシンプールを設定する場合は、Amazon EC2 スポットインスタンスを使用します。

a. **Use Amazon EC2 Spot Instances** を選択します。

b. オンデマンドのインスタンス価格を使用するには、**Use On-Demand instance price** を選択したままにします。または、**Set maximum price** を選択して、Spot インスタンスの 1 時間ごとの最大価格を定義します。
Amazon EC2 Spot インスタンスの詳細は、AWS のドキュメント を参照してください。



重要

Amazon EC2 Spot インスタンスはいつでも中断する可能性があります。Amazon EC2 Spot インスタンスは、中断に対応できるワークロードにのみ使用します。

注記

マシンプールに **Use Amazon EC2 Spot Instances** を選択すると、マシンプールの作成後にオプションを無効にすることはできません。

9. **Add machine pool** をクリックしてマシンプールを作成します。

検証

- マシンプールが **Machine pools** ページに表示され、設定が想定どおりに表示されていることを確認します。

3.2.2. コンピュートノードの手動によるスケーリング

マシンプールの自動スケーリングを有効にしていない場合は、デプロイメントのニーズに合わせてプール内のコンピュート (ワーカーとも呼ばれる) ノードの数を手動でスケーリングできます。

各マシンプールを個別にスケーリングする必要があります。

前提条件

- OpenShift Dedicated クラスターを作成している。
- 既存のマシンプールがある。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. **Machine pools** タブで、スケーリングするマシンプールのオプションメニュー をクリックします。
3. **Scale** を選択します。
4. ノード数を指定します。
   - 単一のアベイラビリティーゾーンを使用してクラスターをデプロイした場合は、ドロップ

ダウンメニューでNode count を指定します。

- 複数のアベイラビリティーゾーンを使用してクラスターをデプロイした場合は、ドロップダウンメニューでNode count per zone を指定します。

注記

サブスクリプションにより、選択可能なノードの数が決まります。

5. Apply をクリックして、マシンプールをスケーリングします。

検証

- Machine pools タブで、マシンプールのNode count が期待どおりであることを確認します。

3.2.3. ノードラベル

ラベルは、**Node** オブジェクトに適用されるキーと値のペアです。ラベルを使用して一連のオブジェクトを整理し、Pod のスケジューリングを制御できます。

クラスターの作成中または後にラベルを追加できます。ラベルはいつでも変更または更新できます。

関連情報

- ラベルの詳細は、Kubernetes ラベルおよびセレクターの概要 を参照してください。

3.2.3.1. ノードラベルのマシンプールへの追加

いつでもコンピュート (ワーカーとも呼ばれる) ノードのラベルを追加または編集して、適切な方法でノードを管理します。たとえば、ワークロードのタイプを特定のノードに割り当てることができます。

ラベルは key-value ペアとして割り当てられます。各キーは、割り当てられたオブジェクトに固有のものである必要があります。

前提条件

- OpenShift Dedicated クラスターを作成している。
- 既存のマシンプールがある。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. Machine pools タブで、ラベルを追加するマシンプールのオプションメニュー をクリックします。
3. Edit labels を選択します。
4. 削除するマシンプールに既存のラベルがある場合は、ラベルの横にあるx を選択して削除します。

5. **<key>=<value>** の形式を使用してラベルを追加し、Enter を押します。たとえば、**app=db** を追加して、Enter を押します。形式が正しい場合は、キーと値のペアが強調表示されます。
6. ラベルを追加する場合は、前の手順を繰り返します。
7. **Save** をクリックして、ラベルをマシンプールに適用します。

検証

1. **Machine pools** タブで、マシンプールの横にある > を選択して、ビューを展開します。
2. 展開されたビューの **Labels** の下にラベルが表示されていることを確認します。

3.2.4. マシンプールへのテイントの追加

マシンプールにコンピュート (ワーカーとも呼ばれる) ノードにテイントを追加して、そのノードにスケジュールされる Pod を制御できます。テイントをマシンプールに適用すると、Pod 仕様にテイントの容認が含まれない限り、スケジューラーは Pod をプールに配置できません。

前提条件

- OpenShift Dedicated クラスターを作成している。
- 既存のマシンプールがある。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. **Machine pools** タブで、テイントを追加するマシンプールのオプションメニュー をクリックします。
3. **Edit taints** を選択します。
4. テイントの**Key** と **Value** のエントリーを追加します。
5. ドロップダウンメニューからテイントの **Effect** を選択します。使用できるオプションには、**NoSchedule**、**PreferNoSchedule**、および **NoExecute** が含まれます。
6. マシンプールにテイントを追加する場合は、**Add taint** を選択します。
7. **Save** をクリックして、テイントをマシンプールに適用します。

検証

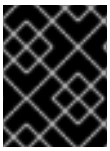
1. **Machine pools** タブで、マシンプールの横にある > を選択して、ビューを展開します。
2. 展開されたビューの **Taints** の下にテイントがリストされていることを確認します。

3.2.5. 関連情報

- マシンプールについて
- 自動スケーリングの有効化

- 自動スケーリングの無効化
- OpenShift Dedicated サービス定義

3.3. クラスターでのノードの自動スケーリングについて



重要

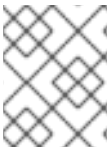
自動スケーリングは、Red Hat Marketplace で購入したクラスターでのみ利用できます。

自動スケーリングオプションは、クラスター内のマシンの数を自動的にスケーリングするように設定できます。

Cluster Autoscaler は、リソース不足のために現在のノードのいずれにもスケジュールできない Pod がある場合、またはデプロイメントのニーズを満たすために別のノードが必要な場合に、クラスターのサイズを拡大します。Cluster Autoscaler は、指定される制限を超えてクラスターリソースを拡大することはありません。

さらに、Cluster Autoscaler は、リソースの使用量が少なく、重要な Pod すべてが他のノードに適合する場合など、一部のノードが長い期間にわたって不要な状態が続く場合にクラスターのサイズを縮小します。

自動スケーリングを有効にする場合は、ワーカーノードの最小数および最大数も設定する必要があります。



注記

クラスターの所有者と組織管理者のみがクラスターのスケーリングまたは削除が可能です。

3.3.1. クラスターでの自動スケーリングノードの有効化

ワーカーノードで自動スケーリングを有効にし、既存クラスターのマシンプール定義を編集して利用可能なノード数を増減できます。

Red Hat OpenShift Cluster Manager を使用して既存のクラスターで自動スケーリングノードを有効にする
OpenShift Cluster Manager コンソールからマシンプール定義でワーカーノードの自動スケーリングを有効にします。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console から、**Clusters** ページに移動し、自動スケーリングを有効にするクラスターを選択します。
2. 選択したクラスターで、**Machine pools** タブを選択します。
3. 自動スケーリングを有効にするマシンプールの最後にある Options メニュー をクリックし、**Scale** を選択します。
4. **Edit node count** ダイアログで、**Enable autoscaling** チェックボックスを選択します。

5. **Apply** を選択してこれらの変更を保存し、クラスターの自動スケーリングを有効にします。

3.3.2. クラスターでの自動スケーリングノードの無効化

ワーカーノードで自動スケーリングを無効にし、既存クラスターのマシンプール定義を編集して利用可能なノード数を増減できます。

OpenShift Cluster Manager コンソールを使用して、クラスターでの自動スケーリングを無効にできます。

Red Hat OpenShift Cluster Manager を使用して既存のクラスターで自動スケーリングノードを無効にする
OpenShift Cluster Manager コンソールからマシンプール定義でワーカーノードの自動スケーリングを無効にします。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console から **Clusters** ページに移動し、自動スケーリングを無効にする必要があるクラスターを選択します。
2. 選択したクラスターで、**Machine pools** タブを選択します。
3. 自動スケーリングのあるマシンプールの最後にある Options メニュー をクリックし、**Scale** を選択します。
4. ノード数の編集ダイアログで、**Enable autoscaling** チェックボックスの選択を解除します。
5. **Apply** を選択してこれらの変更を保存し、クラスターから自動スケーリングを無効にします。

自動スケーリングの OpenShift Dedicated クラスターへの適用には、クラスターへの Cluster Autoscaler のデプロイと各マシンタイプの Machine Autoscaler のデプロイが必要です。

重要

Cluster Autoscaler は、マシン API が機能しているクラスターでのみ設定できます。

3.3.3. Cluster Autoscaler について

Cluster Autoscaler は、現行のデプロイメントのニーズに合わせて OpenShift Dedicated クラスターのサイズを調整します。これは、Kubernetes 形式の宣言引数を使用して、特定のクラウドプロバイダーのオブジェクトに依存しないインフラストラクチャー管理を提供します。Cluster Autoscaler には cluster スコープがあり、特定の namespace には関連付けられていません。

Cluster Autoscaler は、リソース不足のために現在のワーカーノードのいずれにもスケジュールできない Pod がある場合や、デプロイメントのニーズを満たすために別のノードが必要な場合に、クラスターのサイズを拡大します。Cluster Autoscaler は、指定される制限を超えてクラスターリソースを拡大することはありません。

Cluster Autoscaler は、コントロールプレーンノードを管理しない場合でも、クラスター内のすべてのノードのメモリー、CPU の合計を計算します。これらの値は、単一マシン指向ではありません。これらは、クラスター全体での全リソースの集約です。たとえば、最大メモリーリソースの制限を設定する場合、Cluster Autoscaler は現在のメモリー使用量を計算する際にクラスター内のすべてのノードを含めます。この計算は、Cluster Autoscaler にワーカーリソースを追加する容量があるかどうかを判別するために使用されます。

重要

作成する **ClusterAutoscaler** リソース定義の **maxNodesTotal** 値が、クラスター内のマシンの想定される合計数に対応するのに十分な大きさの値であることを確認します。この値は、コントロールプレーンマシンの数とスケーリングする可能性のあるコンピュートマシンの数に対応できる値である必要があります。

Cluster Autoscaler は 10 秒ごとに、クラスターで不要なノードをチェックし、それらを削除します。Cluster Autoscaler は、以下の条件が適用される場合に、ノードを削除すべきと考えます。

- ノードの使用率はクラスターの **ノード使用率レベル** のしきい値 よりも低くなります。ノード使用率レベルとは、要求されたリソースの合計をノードに割り当てられたリソースで除算したものです。**ClusterAutoscaler** カスタムリソースで値を指定しない場合、Cluster Autoscaler は 50% の使用率に対応するデフォルト値 **0.5** を使用します。
- Cluster Autoscaler がノードで実行されているすべての Pod を他のノードに移動できる。
- Cluster Autoscaler で、スケールダウンが無効にされたアノテーションがない。

以下のタイプの Pod がノードにある場合、Cluster Autoscaler はそのノードを削除しません。

- 制限のある Pod の Disruption Budget (停止状態の予算、PDB) を持つ Pod。
- デフォルトでノードで実行されない Kube システム Pod。
- PDB を持たないか、または制限が厳しい PDB を持つ Kuber システム Pod。
- デプロイメント、レプリカセット、またはステートフルセットなどのコントローラーオブジェクトによってサポートされない Pod。
- ローカルストレージを持つ Pod。
- リソース不足、互換性のないノードセレクターまたはアフィニティー、一致する非アフィニティーなどにより他の場所に移動できない Pod。
- それらに **"cluster-autoscaler.kubernetes.io/safe-to-evict": "true"** アノテーションがない場合、**"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"** アノテーションを持つ Pod。

たとえば、CPU の上限を 64 コアに設定し、それぞれ 8 コアを持つマシンのみを作成するように Cluster Autoscaler を設定したとします。クラスターが 30 コアで起動する場合、Cluster Autoscaler は最大で 4 つのノード (合計 32 コア) を追加できます。この場合、総計は 62 コアになります。

Cluster Autoscaler を設定する場合、使用に関する追加の制限が適用されます。

- 自動スケーリングされたノードグループにあるノードを直接変更しない。同じノードグループ内のすべてのノードには同じ容量およびラベルがあり、同じシステム Pod を実行します。
- Pod の要求を指定します。
- Pod がすぐに削除されるのを防ぐ必要がある場合、適切な PDB を設定します。
- クラウドプロバイダーのクォータが、設定する最大のノードプールに対応できる十分な大きさであることを確認します。
- クラウドプロバイダーで提供されるものなどの、追加のノードグループの Autoscaler を実行しない。

Horizontal Pod Autoscaler (HPA) および Cluster Autoscaler は複数の異なる方法でクラスターリソースを変更します。HPA は、現在の CPU 負荷に基づいてデプロイメント、またはレプリカセットのレプリカ数を変更します。負荷が増大すると、HPA はクラスターで利用できるリソース量に関係なく、新規レプリカを作成します。十分なリソースがない場合、Cluster Autoscaler はリソースを追加し、HPA で作成された Pod が実行できるようにします。負荷が減少する場合、HPA は一部のレプリカを停止します。この動作によって一部のノードの使用率が低くなるか、または完全に空になる場合、Cluster Autoscaler は不必要なノードを削除します。

Cluster Autoscaler は Pod の優先順位を考慮に入れます。Pod の優先順位とプリエンプション機能により、クラスターに十分なリソースがない場合に優先順位に基づいて Pod のスケジューリングを有効にできますが、Cluster Autoscaler はクラスターがすべての Pod を実行するのに必要なリソースを確保できます。これら両方の機能の意図を反映するべく、Cluster Autoscaler には優先順位のカットオフ機能が含まれています。このカットオフを使用して Best Effort の Pod をスケジュールできますが、これにより Cluster Autoscaler がリソースを増やすことはなく、余分なリソースがある場合にのみ実行されます。

カットオフ値よりも低い優先順位を持つ Pod は、クラスターをスケールアップせず、クラスターのスケールダウンを防ぐこともありません。これらの Pod を実行するために新規ノードは追加されず、これらの Pod を実行しているノードはリソースを解放するために削除される可能性があります。

3.3.4. 関連情報

- About machinepools

第4章 ロギング

4.1. OPENSHIFT DEDICATED クラスターのサービスログへのアクセス

Red Hat OpenShift Cluster Manager を使用して、OpenShift Dedicated クラスターのサービスログを表示できます。サービスログには、ロードバランサークォータの更新やスケジュールされたメンテナーンスアップグレードなどの詳細なクラスターイベントが記録されます。ログには、ユーザー、グループ、および ID プロバイダーの追加または削除などのクラスターリソースの変更も表示されます。

さらに、OpenShift Dedicated クラスターの通知連絡先を追加できます。サブスクライブしたユーザーは、顧客の対応が必要なクラスターイベント、既知のクラスターインシデント、アップグレードのメンテナーンス、およびその他のトピックに関する電子メールを受け取ります。

4.1.1. OpenShift Cluster Manager を使用したサービスログの表示

Red Hat OpenShift Cluster Manager を使用して、OpenShift Dedicated クラスターのサービスログを表示できます。

前提条件

- OpenShift Dedicated クラスターをインストールしました。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. クラスターの **Overview** ページで、**Cluster history** セクションのサービスログを表示します。
3. オプション: ドロップダウンメニューから、**Description** または **Severity** でクラスターサービスのログをフィルターリングします。検索バーに特定の項目を入力して、さらにフィルターリングできます。
4. オプション: **Download history** をクリックして、クラスターのサービスログを JSON または CSV 形式でダウンロードします。

4.1.2. クラスター通知連絡先の追加

OpenShift Dedicated クラスターに関する通知の連絡先を追加できます。クラスター通知メールをトリガーするイベントが発生すると、サブスクライブしているユーザーに通知が送信されます。

手順

1. OpenShift Cluster Manager Hybrid Cloud Console に移動し、クラスターを選択します。
2. **Notification contacts** 見出しの **Support** タブで、**Add notification contact** をクリックします。
3. 追加する連絡先の Red Hat ユーザー名またはメールアドレスを入力します。

注記

ユーザー名または電子メールアドレスは、クラスターがデプロイされている Red Hat 組織のユーザーアカウントに関連付けられている必要があります。

4. **Add contact** をクリックします。

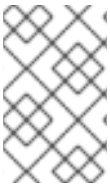
検証

- 連絡先が正常に追加されると、確認メッセージが表示されます。ユーザーは、**Support** タブの **Notification contacts** 見出しの下に表示されます。

第5章 ユーザー定義プロジェクトのモニタリング

5.1. モニタリングスタックについて

OpenShift Dedicated では、Red Hat Site Reliability Engineer (SRE) プラットフォームメトリクスから切り離して独自のプロジェクトをモニターできます。追加のモニタリングソリューションなしに独自のプロジェクトをモニターできます。



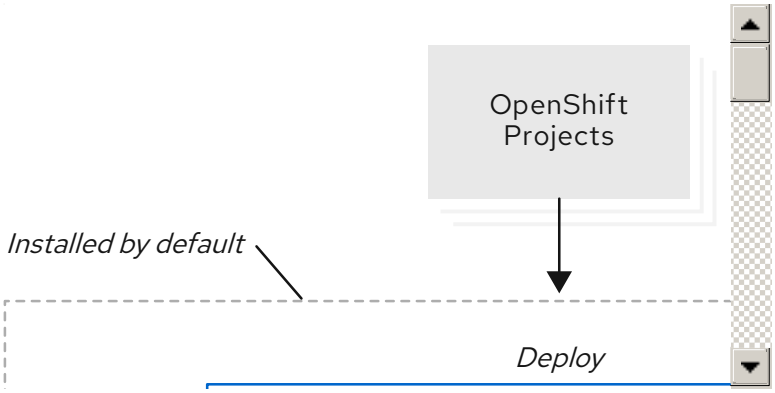
注記

本書の手順に従って、ユーザー定義プロジェクトをモニターするためにサポートされる Prometheus インスタンスを設定します。カスタム Prometheus インスタンスは OpenShift Dedicated ではサポートされません。

5.1.1. モニタリングスタックについて

OpenShift Dedicated モニタリングスタックは、Prometheus オープンソースプロジェクトおよびその幅広いエコシステムをベースとしています。モニタリングスタックには、以下のコンポーネントが含まれます。

- **デフォルトのプラットフォームモニタリングコンポーネント**。プラットフォームモニタリングコンポーネントのセットは、OpenShift Dedicated のインストール時にデフォルトで **openshift-monitoring** プロジェクトにインストールされ、有効になります。これにより、コアクラスターコンポーネントを監視できます。デフォルトのモニタリングスタックは、クラスターのリモートのヘルスモニタリングも有効にします。CPU やメモリーなどの重要なメトリクスは、すべての namespace のすべてのワークロードから収集され、利用可能になります。これらのコンポーネントは、以下の図の **Installed by default** セクションで説明されています。
- **ユーザー定義のプロジェクトをモニターするためのコンポーネント**。この機能はデフォルトで有効になっており、ユーザー定義プロジェクトのモニタリングを提供します。これらのコンポーネントは、以下の図の **User** セクションで説明されています。



5.1.1.1. ユーザー定義プロジェクトをモニターするためのコンポーネント

OpenShift Dedicated には、ユーザー定義プロジェクトでサービスおよび Pod をモニターできるモニタリングスタックのオプションの拡張機能が含まれています。この機能には、以下のコンポーネントが含まれます。

表5.1 ユーザー定義プロジェクトをモニターするためのコンポーネント

コンポーネント	説明
Prometheus Operator	**openshift-user-workload-monitoring** プロジェクトの Prometheus Operator は、同じプロジェクトで Prometheus および Thanos Ruler インスタンスの作成、設定、および管理を行います。
Prometheus	Prometheus は、ユーザー定義のプロジェクト用にモニタリング機能が提供されるモニタリングシステムです。Prometheus は処理のためにアラートを Alertmanager に送信します。ただし、アラートのルーティングは現在サポートされていません。
Thanos Ruler	Thanos Ruler は、別のプロセスとしてデプロイされる Prometheus のルール評価エンジンです。OpenShift Dedicated 4 では、Thanos Ruler はユーザー定義プロジェクトのモニタリングについてのルールおよびアラート評価を提供します。

これらのすべてのコンポーネントはスタックによってモニターされ、OpenShift Dedicated の更新時に自動的に更新されます。

5.1.1.2. ユーザー定義プロジェクトのターゲットのモニタリング

モニタリングは、OpenShift Dedicated のユーザー定義プロジェクトについてデフォルトで有効にされます。以下をモニターできます。

- ユーザー定義プロジェクトのサービスエンドポイント経由で提供されるメトリクス。
- ユーザー定義プロジェクトで実行される Pod。

5.1.2. 関連情報

- ユーザー定義プロジェクトのモニタリングのアクセス
- デフォルトのモニタリングコンポーネント
- デフォルトのモニタリングターゲット

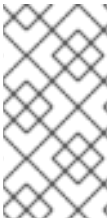
5.1.3. 次のステップ

- ユーザー定義プロジェクトのモニタリングのアクセス

5.2. ユーザー定義プロジェクトのモニタリングのアクセス

OpenShift Dedicated クラスターをインストールすると、ユーザー定義プロジェクトのモニタリングがデフォルトで有効になります。ユーザー定義プロジェクトのモニタリングを有効にすると、追加のモニタリングソリューションを必要とせずに、独自の OpenShift Dedicated プロジェクトをモニタリングできます。

dedicated-admin ユーザーには、ユーザー定義プロジェクトのモニタリングを設定し、アクセスするためのデフォルトのパーミッションがあります。



注記

カスタム Prometheus インスタンスおよび Operator Lifecycle Manager (OLM) でインストールされる Prometheus Operator では、ユーザー定義のプロジェクトモニタリングが有効である場合にこれに関する問題が生じる可能性があります。カスタム Prometheus インスタンスはサポートされません。

必要に応じて、クラスターのインストール中またはインストール後に、ユーザー定義プロジェクトの監視を無効にすることができます。

5.2.1. 次のステップ

- モニタリングスタックの設定

5.3. モニタリングスタックの設定

モニタリングスタックの設定後に、共通の設定シナリオを確認し、ユーザー定義プロジェクトのモニタリングを設定できます。

5.3.1. モニタリングのメンテナーンスおよびサポート

OpenShift Dedicated Monitoring の設定のサポートされる方法として、本書で説明されているオプションを使用できます。**サポートされていない他の設定は使用しないでください。**

重要

別の Prometheus インスタンスのインストールは、Red Hat Site Reliability Engineers (SRE) ではサポートされていません。

設定のパラダイムが Prometheus リリース間で変更される可能性があり、このような変更には、設定のすべての可能性が制御されている場合のみ適切に対応できます。本セクションで説明されている設定以外の設定を使用する場合は、**cluster-monitoring-operator** が差分を調整するため、変更内容が失われます。Operator はデフォルトで定義された状態へすべてをリセットします。

5.3.1.1. ユーザー定義プロジェクトのモニターに関するサポートの考慮事項

以下の変更は明示的にサポートされていません。

- カスタム Prometheus インスタンスの OpenShift Dedicated へのインストール

5.3.2. モニタリングスタックの設定

OpenShift Dedicated では、**user-workload-monitoring-config ConfigMap** オブジェクトを使用してユーザー定義プロジェクトのワークロードをモニターするスタックを設定できます。設定マップはクラスターモニタリング Operator (CMO) を設定し、その後にスタックのコンポーネントが設定されます。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

   b. 設定を、**data.config.yaml** の下に値とキーのペア **<component_name>: <component_configuration>** として追加します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    <component>:
      <configuration_for_the_component>
```

   <component> および **<configuration_for_the_component>** を随時置き換えます。

   以下の **ConfigMap** オブジェクトの例は、Prometheus のデータ保持期間および最小コンテナーリソース要求を設定します。これは、ユーザー定義のプロジェクトのみをモニターする Prometheus インスタンスに関連します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus: 1
      retention: 24h 2
      resources:
        requests:
          cpu: 200m 3
          memory: 2Gi 4
```

   1. Prometheus コンポーネントを定義し、後続の行はその設定を定義します。
   2. ユーザー定義プロジェクトをモニターする Prometheus インスタンスについて 24 時間のデータ保持期間を設定します。
   3. Prometheus コンテナーの 200 ミリコアの最小リソース要求を定義します。
   4. Prometheus コンテナーのメモリーの 2 GiB の最小 Pod リソース要求を定義します。

2. ファイルを保存して、変更を **ConfigMap** オブジェクトに適用します。新規設定の影響を受けた Pod は自動的に再起動されます。

警告

変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

5.3.3. 設定可能なモニタリングコンポーネント

以下の表は、設定可能なモニタリングコンポーネントと、**user-workload-monitoring-config ConfigMap** オブジェクトでコンポーネントを指定するために使用されるキーを示しています。

表5.2 設定可能なモニタリングコンポーネント

コンポーネント	user-workload-monitoring-config 設定マップキー
Prometheus Operator	**prometheusOperator**
Prometheus	**prometheus**
Thanos Ruler	**thanosRuler**

5.3.4. モニタリングコンポーネントの異なるノードへの移動

ユーザー定義プロジェクトのワークロードをモニターする任意のコンポーネントを特定のワーカーノードに移動できます。コンポーネントをコントロールプレーンまたはインフラストラクチャーノードに移動することは許可されていません。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. ユーザー定義プロジェクトをモニターするコンポーネントを移動するには、**ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

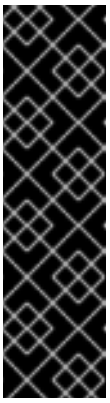
   ```
   $ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
   ```

   b. コンポーネントの **nodeSelector** 制約を **data.config.yaml** に指定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    <component>:
      nodeSelector:
        <node_key>: <node_value>
        <node_key>: <node_value>
        <...>
```

<component> を適宜置き換え、**<node_key>: <node_value>** を、宛先ノードを指定するキーと値のペアのマップに置き換えます。通常は、単一のキーと値のペアのみが使用されます。

コンポーネントは、指定されたキーと値のペアのそれぞれをラベルとして持つノードでのみ実行できます。ノードには追加のラベルを持たせることもできます。



重要

モニタリングコンポーネントの多くは、高可用性を維持するために、クラスターの異なるノード間で複数の Pod を使用してデプロイされます。モニタリングコンポーネントをラベル付きノードに移動する際には、コンポーネントの耐障害性を維持するために十分な数の一致するノードが利用可能であることを確認します。1 つのラベルのみが指定されている場合は、複数の別々のノードにコンポーネントに関連するすべての Pod を分散するために、十分な数のノードにそのラベルが含まれていることを確認します。または、複数のラベルを指定することもできます。その場合は、それぞれのラベルを個々のノードに関連付けます。

注記

nodeSelector の制約を設定した後もモニタリングコンポーネントが **Pending** 状態のままになっている場合は、Pod ログでテイントおよび容認に関連するエラーの有無を確認します。

たとえば、ユーザー定義プロジェクトのモニタリングコンポーネントを **nodename: worker1**、**nodename: worker2**、および **nodename: worker2** のラベルが付けられた特定のワーカーノードに移行するには、以下を使用します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheusOperator:
      nodeSelector:
        nodename: worker1
    prometheus:
      nodeSelector:
        nodename: worker1
```

```
      nodename: worker2
  thanosRuler:
    nodeSelector:
      nodename: worker1
      nodename: worker2
```

2. 変更を適用するためにファイルを保存します。新しい設定の影響を受けるコンポーネントは新しいノードに自動的に移動します。

警告

変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

関連情報

- ノードでラベルを更新する方法について
- ノードセレクターの使用による特定ノードへの Pod の配置
- **nodeSelector** 制約についての詳細は、Kubernetes ドキュメント を参照してください。

5.3.5. ユーザー定義プロジェクトをモニターするコンポーネントへの容認の割り当て

ユーザー定義プロジェクトをモニターするコンポーネントに許容値を割り当てて、汚染されたワーカーノードにプロジェクトを移動できるようにすることができます。コントロールプレーンまたはインフラストラクチャーノードでのスケジューリングは許可されていません。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを **openshift-user-workload-monitoring** namespace に作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

   b. コンポーネントの **tolerations** を指定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    <component>:
      tolerations:
        <toleration_specification>
```

<component> および **<toleration_specification>** を随時置き換えます。

たとえば、**oc adm taint nodes node1 key1=value1:NoSchedule** は、キーが **key1** で、値が **value1** の **node1** にテイントを追加します。これにより、モニタリングコンポーネントが **node1** に Pod をデプロイするのを防ぎます。ただし、そのテイントに対して許容値が設定されている場合を除きます。以下の例では、サンプルのテイントを容認するように **thanosRuler** コンポーネントを設定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    thanosRuler:
      tolerations:
      - key: "key1"
        operator: "Equal"
        value: "value1"
        effect: "NoSchedule"
```

2. 変更を適用するためにファイルを保存します。新しいコンポーネントの配置設定が自動的に適用されます。

警告

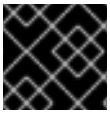
変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

関連情報

- テイントおよび許容値については、OpenShift Container Platform ドキュメント を参照してください。
- テイントおよび許容値については、Kubernetes ドキュメント を参照してください。

5.3.6. 永続ストレージの設定

クラスターモニタリングを永続ストレージと共に実行すると、メトリクスは永続ボリューム (PV) に保存され、Pod の再起動または再作成後も維持されます。これは、メトリクスデータをデータ損失から保護する必要がある場合に適しています。実稼働環境では、永続ストレージを設定することを強く推奨します。IO デマンドが高いため、ローカルストレージを使用することが有利になります。



重要

設定可能な推奨のストレージ技術 を参照してください。

5.3.6.1. 永続ストレージの前提条件

- ストレージのブロックタイプを使用します。

5.3.6.2. ローカ永続ボリューム要求 (PVC) の設定

モニタリングコンポーネントが永続ボリューム (PV) を使用できるようにするには、永続ボリューム要求 (PVC) を設定する必要があります。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. ユーザー定義プロジェクトをモニターするコンポーネントの PVC を設定するには、**ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

   b. コンポーネントの PVC 設定を **data.config.yaml** の下に追加します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    <component>:
      volumeClaimTemplate:
        spec:
          storageClassName: <storage_class>
          resources:
            requests:
              storage: <amount_of_storage>
```

- **volumeClaimTemplate** の指定方法は、PersistentVolumeClaims に関する Kubernetes ドキュメント を参照してください。

以下の例では、ユーザー定義プロジェクトをモニターする Prometheus インスタンスのローカル永続ストレージを要求する PVC を設定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      volumeClaimTemplate:
        spec:
          storageClassName: local-storage
          resources:
            requests:
              storage: 40Gi
```

上記の例では、ローカルストレージ Operator によって作成されるストレージクラスは **local-storage** と呼ばれます。

以下の例では、Thanos Ruler のローカル永続ストレージを要求する PVC を設定します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    thanosRuler:
      volumeClaimTemplate:
        spec:
          storageClassName: local-storage
          resources:
            requests:
              storage: 40Gi
```

2. 変更を適用するためにファイルを保存します。新規設定の影響を受けた Pod は自動的に再起動され、新規ストレージ設定が適用されます。

警告

変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

5.3.6.3. Prometheus メトリクスデータの保持期間の変更

デフォルトで、OpenShift Dedicated モニタリングスタックは、Prometheus データの保持期間を 15 日間に設定します。データを削除するタイミングを変更するために、ユーザー定義のプロジェクトをモニターする Prometheus インスタンスの保持時間を変更できます。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. ユーザー定義プロジェクトをモニターする Prometheus インスタンスの保持時間を変更するには、**ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

   ```
   $ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
   ```

   b. 保持期間の設定を **data.config.yaml** に追加します。

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: user-workload-monitoring-config
     namespace: openshift-user-workload-monitoring
   data:
     config.yaml: |
       prometheus:
         retention: <time_specification>
   ```

   <time_specification> を、**ms** (ミリ秒)、**s** (秒)、**m** (分)、**h** (時間)、**d** (日)、**w** (週)、または **y** (年) が直後に続く数字に置き換えます。

   以下の例では、ユーザー定義プロジェクトをモニターする Prometheus インスタンスの保持期間を 24 時間に設定します。

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: user-workload-monitoring-config
     namespace: openshift-user-workload-monitoring
   data:
     config.yaml: |
       prometheus:
         retention: 24h
   ```

2. 変更を適用するためにファイルを保存します。新規設定の影響を受けた Pod は自動的に再起動されます。

警告

変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

関連情報

- 永続ストレージについて
- Optimizing storage

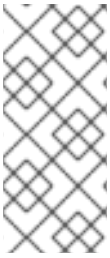
5.3.7. ユーザー定義プロジェクトでバインドされていないメトリクス属性の影響の制御

開発者は、キーと値のペアの形式でメトリクスの属性を定義するためにラベルを作成できます。使用できる可能性のあるキーと値のペアの数は、属性について使用できる可能性のある値の数に対応します。数が無制限の値を持つ属性は、バインドされていない属性と呼ばれます。たとえば、**customer_id** 属性は、使用できる値が無限にあるため、バインドされていない属性になります。

割り当てられるキーと値のペアにはすべて、一意の時系列があります。ラベルに多数のバインドされていない値を使用すると、作成される時系列の数が指数関数的に増加する可能性があります。これは Prometheus のパフォーマンスに影響する可能性があり、多くのディスク領域を消費する可能性があります。

dedicated-admin は、以下の手段を使用して、ユーザー定義プロジェクトでのバインドされていないメトリクス属性の影響を制御できます。

- ユーザー定義プロジェクトで、ターゲット収集ごとに **受け入れ可能なサンプル数を制限** します。



注記

収集サンプルを制限すると、多くのバインドされていない属性をラベルに追加して問題が発生するのを防ぐことができます。さらに開発者は、メトリクスに定義するバインドされていない 属性の数を制限することにより、根本的な原因を防ぐことができます。使用可能な値の制限されたセットにバインドされる属性を使用すると、可能なキーと値のペアの組み合わせの数が減ります。

5.3.7.1. ユーザー定義プロジェクトの収集サンプル制限の設定

ユーザー定義プロジェクトで、ターゲット収集ごとに受け入れ可能なサンプル数を制限できます。

警告

サンプル制限を設定すると、制限に達した後にそのターゲット収集についての追加のサンプルデータは取り込まれません。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

2. **enforcedSampleLimit** 設定を **data.config.yaml** に追加し、ユーザー定義プロジェクトのターゲットの収集ごとに受け入れ可能なサンプルの数を制限できます。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      enforcedSampleLimit: 50000 1
```

1 このパラメーターが指定されている場合は、値が必要です。この **enforcedSampleLimit** の例では、ユーザー定義プロジェクトのターゲット収集ごとに受け入れ可能なサンプル数を 50,000 に制限します。

3. 変更を適用するためにファイルを保存します。制限は自動的に適用されます。

警告

変更が **user-workload-monitoring-config ConfigMap** オブジェクトに保存されると、**openshift-user-workload-monitoring** プロジェクトの Pod および他のリソースは再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

関連情報

- 最高数の収集サンプルを持つメトリクスをクエリーする手順: Prometheus が大量のディスク領域を消費している理由の特定

5.3.8. モニタリングコンポーネントのログレベルの設定

Prometheus Operator、Prometheus、および Thanos Ruler のログレベルを設定できます。

以下のログレベルは、**user-workload-monitoring-config ConfigMap** オブジェクトのそれらのコンポーネントのそれぞれに適用できます。

- **debug**: デバッグ、情報、警告、およびエラーメッセージをログに記録します。
- **info**:情報、警告およびエラーメッセージをログに記録します。
- **warn**:警告およびエラーメッセージのみをログに記録します。
- **error**:エラーメッセージのみをログに記録します。

デフォルトのログレベルは **info** です。

前提条件

- **dedicated-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。
- **user-workload-monitoring-config ConfigMap** オブジェクトを作成している。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **ConfigMap** オブジェクトを編集します。
   a. **openshift-user-workload-monitoring** プロジェクトで **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

   b. コンポーネントの **logLevel: <log_level>** を **data.config.yaml** の下に追加します。

```
apiVersion: v1
```

```
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    <component>: 1
      logLevel: <log_level> 2
```

1. ログレベルを適用するモニタリングコンポーネント。
2. コンポーネントに適用するログレベル。

2. 変更を適用するためにファイルを保存します。ログレベルの変更を適用する際に、コンポーネントの Pod は自動的に再起動します。

警告

変更がモニタリング設定マップに保存されると、関連するプロジェクトの Pod およびその他のリソースが再デプロイされる可能性があります。該当するプロジェクトの実行中のモニタリングプロセスも再起動する可能性があります。

3. 関連するプロジェクトでデプロイメントまたは Pod 設定を確認し、ログレベルが適用されていることを確認します。以下の例では、**openshift-user-workload-monitoring** プロジェクトの **prometheus-operator** デプロイメントでログレベルを確認します。

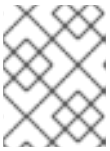
```
$ oc -n openshift-user-workload-monitoring get deploy prometheus-operator -o yaml |  grep "log-level"
```

出力例

```
        - --log-level=debug
```

4. コンポーネントの Pod が実行中であることを確認します。以下の例は、**openshift-user-workload-monitoring** プロジェクトの Pod のステータスを一覧表示します。

```
$ oc -n openshift-user-workload-monitoring get pods
```



注記

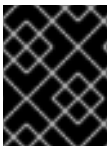
認識されない **loglevel** 値が **ConfigMap** オブジェクトに含まれる場合は、コンポーネントの Pod が正常に再起動しない可能性があります。

5.3.9. 次のステップ

- メトリクスの管理

5.4. ユーザー定義プロジェクトのアラートルーティングの有効化

OpenShift Dedicated では、クラスター管理者はユーザー定義プロジェクトのアラートルーティングを有効にできます。



重要

ユーザー定義プロジェクトのアラートルールの管理は、OpenShift Dedicated バージョン 4.11 以降でのみ利用できます。

このプロセスは、以下の 2 つの一般的な手順で設定されています。

- ユーザー定義プロジェクトのアラートルーティングを有効にして、別の Alertmanager インスタンスを使用します。
- ユーザー定義プロジェクトのアラートルーティングを設定する権限を追加のユーザーに付与します。

これらの手順を完了すると、開発者およびその他のユーザーはユーザー定義のプロジェクトのカスタムアラートおよびアラートルーティングを設定できます。

5.4.1. ユーザー定義プロジェクトのアラートルーティングについて

クラスター管理者は、ユーザー定義プロジェクトのアラートルーティングを有効にできます。この機能により、**alert-routing-edit** ロールを持つユーザーがユーザー定義プロジェクトのアラート通知ルーティングおよびレシーバーを設定できます。これらの通知は、ユーザー定義の監視専用の Alertmanager インスタンスによってルーティングされます。

次に、ユーザーはユーザー定義プロジェクトの **AlertmanagerConfig** オブジェクトを作成または編集して、ユーザー定義のアラートルーティングを作成し、設定できます。

ユーザー定義プロジェクトのアラートルーティングをユーザーが定義すると、ユーザー定義のアラート通知が **openshift-user-workload-monitoring** namespace の **alertmanager-user-workload** Pod にルーティングされます。

注記

以下は、ユーザー定義プロジェクトのアラートルーティングの制限です。

- ユーザー定義のアラートルールの場合、ユーザー定義のルーティングはリソースが定義される namespace に対してスコープ指定されます。たとえば、namespace **ns1** のルーティング設定は、同じ namespace の **PrometheusRules** リソースにのみ適用されます。
- namespace がユーザー定義のモニタリングから除外される場合、namespace の **AlertmanagerConfig** リソースは、Alertmanager 設定の一部ではなくなります。

5.4.2. ユーザー定義のアラートルーティング用の個別の Alertmanager インスタンスの有効化

OpenShift Dedicated では、ユーザー定義プロジェクト専用の Alertmanager インスタンスをデプロイして、デフォルトのプラットフォームアラートとは別のユーザー定義アラートを提供できます。このような場合、必要に応じて、Alertmanager の別のインスタンスを有効にして、ユーザー定義のプロジェ

クトのみにアラートを送信できます。

前提条件

- **cluster-admin** または **dedicated-admin** ロールが割り当てられたユーザーとしてクラスターにアクセスできる。
- **openshift-monitoring** namespace の **cluster-monitoring-config** 設定マップでユーザー定義プロジェクトのモニタリングを有効にしている。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **user-workload-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc -n openshift-user-workload-monitoring edit configmap user-workload-monitoring-config
```

2. **data/config.yaml** の下にある **alertmanager** セクションに **enabled: true** および **enableAlertmanagerConfig: true** を追加します。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    alertmanager:
      enabled: true 1
      enableAlertmanagerConfig: true 2
```

1. **enabled** の値を **true** に設定して、クラスター内のユーザー定義プロジェクトの Alertmanager の専用インスタンスを有効にします。値を **false** に設定するか、キーを完全に省略してユーザー定義プロジェクトの Alertmanager を無効にします。この値を **false** に設定した場合や、キーを省略すると、ユーザー定義のアラートはデフォルトのプラットフォーム Alertmanager インスタンスにルーティングされます。

2. **enableAlertmanagerConfig** 値を **true** に設定して、ユーザーが **AlertmanagerConfig** オブジェクトで独自のアラートルーティング設定を定義できるようにします。

3. 変更を適用するためにファイルを保存します。ユーザー定義プロジェクトの Alertmanager の専用インスタンスが自動的に起動します。

検証

- **user-workload** Alertmanager インスタンスが起動していることを確認します。

```
# oc -n openshift-user-workload-monitoring get alertmanager
```

出力例

```
NAME          VERSION   REPLICAS   AGE
user-workload   0.24.0    2          100s
```

5.4.3. ユーザー定義プロジェクトのアラートルーティングを設定するためのユーザーへのパーミッションの付与

ユーザー定義プロジェクトのアラートルーティングを設定するパーミッションをユーザーに付与できます。

前提条件

- **cluster-admin** または **dedicated-admin** ロールが割り当てられたユーザーとしてクラスターにアクセスできる。
- ロールを割り当てるユーザーアカウントがすでに存在している。
- OpenShift CLI (**oc**) がインストールされている。
- ユーザー定義プロジェクトのモニタリングを有効にしている。

手順

- ユーザー定義プロジェクトのユーザーに **alert-routing-edit** ロールを割り当てます。

```
$ oc -n <namespace> adm policy add-role-to-user alert-routing-edit <user> 1
```

1 **<namespace>**の場合は、ユーザー定義プロジェクトの代わりに namespace を使用します (例: **ns1**)。**<user>** の場合は、ロールを割り当てるアカウントの代わりにユーザー名を使用します。

関連情報

- ユーザー定義プロジェクトのモニタリングのアクセス
- ユーザー定義プロジェクトのアラートルーティングの作成

5.5. メトリクスの管理

メトリクスを使用すると、クラスターコンポーネントおよび独自のワークロードのパフォーマンスをモニターできます。

5.5.1. メトリクスについて

OpenShift Dedicated では、クラスターコンポーネントはサービスエンドポイントで公開されるメトリクスを収集することによりモニターされます。ユーザー定義プロジェクトのメトリクスのコレクションを設定することもできます。メトリクスを使用すると、クラスターコンポーネントおよび独自のワークロードの実行方法をモニターできます。

Prometheus クライアントライブラリーをアプリケーションレベルで使用することで、独自のワークロードに指定するメトリクスを定義できます。

OpenShift Dedicated では、メトリクスは /**metrics** の正規名の下に HTTP サービスエンドポイント経由

で公開されます。**curl** クエリーを **http://<endpoint>/metrics** に対して実行して、サービスの利用可能なすべてのメトリクスを一覧表示できます。たとえば、**prometheus-example-app** サンプルアプリケーションへのルートを公開し、以下のコマンドを実行して利用可能なすべてのメトリクスを表示できます。

```
$ curl http://<example_app_endpoint>/metrics
```

出力例

```
# HELP http_requests_total Count of all HTTP requests
# TYPE http_requests_total counter
http_requests_total{code="200",method="get"} 4
http_requests_total{code="404",method="get"} 2
# HELP version Version information about this binary
# TYPE version gauge
version{version="v0.1.0"} 1
```

関連情報

- Prometheus クライアントライブラリーについての詳細は、Prometheus ドキュメント を参照してください。

5.5.2. ユーザー定義プロジェクトのメトリクスコレクションの設定

ServiceMonitor リソースを作成して、ユーザー定義プロジェクトのサービスエンドポイントからメトリクスを収集できます。これは、アプリケーションが Prometheus クライアントライブラリーを使用してメトリクスを **/metrics** の正規の名前に公開していることを前提としています。

このセクションでは、ユーザー定義のプロジェクトでサンプルサービスをデプロイし、次にサービスのモニター方法を定義する **ServiceMonitor** リソースを作成する方法を説明します。

5.5.2.1. サンプルサービスのデプロイ

ユーザー定義のプロジェクトでサービスのモニタリングをテストするには、サンプルサービスをデプロイすることができます。

手順

1. サービス設定の YAML ファイルを作成します。この例では、**prometheus-example-app.yaml** という名前です。
2. 以下のデプロイメントおよびサービス設定の詳細をファイルに追加します。

```
apiVersion: v1
kind: Namespace
metadata:
  name: ns1
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: prometheus-example-app
  name: prometheus-example-app
```

```
  namespace: ns1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: prometheus-example-app
  template:
    metadata:
      labels:
        app: prometheus-example-app
    spec:
      containers:
      - image: quay.io/brancz/prometheus-example-app:v0.2.0
        imagePullPolicy: IfNotPresent
        name: prometheus-example-app
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app: prometheus-example-app
  name: prometheus-example-app
  namespace: ns1
spec:
  ports:
  - port: 8080
    protocol: TCP
    targetPort: 8080
    name: web
  selector:
    app: prometheus-example-app
  type: ClusterIP
```

この設定は、**prometheus-example-app** という名前のサービスをユーザー定義の **ns1** プロジェクトにデプロイします。このサービスは、カスタム **version** メトリクスを公開します。

3. 設定をクラスターに適用します。

```
$ oc apply -f prometheus-example-app.yaml
```

サービスをデプロイするには多少時間がかかります。

4. Pod が実行中であることを確認できます。

```
$ oc -n ns1 get pod
```

出力例

```
NAME                                  READY     STATUS    RESTARTS   AGE
prometheus-example-app-7857545cb7-sbgwq   1/1       Running   0          81m
```

5.5.2.2. サービスのモニター方法の指定

サービスが公開するメトリクスを使用するには、OpenShift Dedicated モニタリングを、**/metrics** エン

ドポイントからメトリクスを収集できるように設定する必要があります。これは、サービスのモニタリング方法を指定する **ServiceMonitor** カスタムリソース定義、または Pod のモニタリング方法を指定する **PodMonitor** CRD を使用して実行できます。前者の場合は **Service** オブジェクトが必要ですが、後者の場合は不要です。これにより、Prometheus は Pod によって公開されるメトリクスエンドポイントからメトリクスを直接収集することができます。

注記

OpenShift Dedicated では、**ServiceMonitor** リソースの **tlsConfig** プロパティーを使用し、エンドポイントからメトリクスを収集する際に使用する TLS 設定を指定できます。**tlsConfig** プロパティーは **PodMonitor** リソースではまだ利用できません。メトリクスの収集時に TLS 設定を使用する必要がある場合は、**ServiceMonitor** リソースを使用する必要があります。

この手順では、ユーザー定義プロジェクトでサービスの **ServiceMonitor** リソースを作成する方法を説明します。

前提条件

- **dedicated-admin** ロールまたは **monitoring-edit** 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。
- この例では、**prometheus-example-app** サンプルサービスを **ns1** プロジェクトにデプロイしている。

手順

1. **ServiceMonitor** リソース設定の YAML ファイルを作成します。この例では、ファイルは **example-app-service-monitor.yaml** という名前です。
2. 以下の **ServiceMonitor** リソース設定の詳細を追加します。

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    k8s-app: prometheus-example-monitor
  name: prometheus-example-monitor
  namespace: ns1
spec:
  endpoints:
  - interval: 30s
    port: web
    scheme: http
  selector:
    matchLabels:
      app: prometheus-example-app
```

これは、**prometheus-example-app** サンプルサービスによって公開されるメトリクスを収集する **ServiceMonitor** リソースを定義します。これには **version** メトリクスが含まれます。

3. 設定をクラスターに適用します。

```
$ oc apply -f example-app-service-monitor.yaml
```

ServiceMonitor をデプロイするのに多少時間がかかります。

4. **ServiceMonitor** リソースが実行中であることを確認できます。

```
$ oc -n ns1 get servicemonitor
```

出力例

```
NAME                         AGE
prometheus-example-monitor   81m
```

関連情報

- **ServiceMonitor** および **PodMonitor** リソースについての詳細は、Prometheus Operator API のドキュメント を参照してください。
- ユーザー定義プロジェクトのモニタリングへのアクセス

5.5.3. メトリクスのクエリー

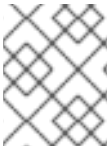
OpenShift モニタリングダッシュボードでは、Prometheus のクエリー言語 (PromQL) クエリーを実行し、プロットに可視化されるメトリクスを検査できます。この機能により、クラスターの状態と、モニターしているユーザー定義のプロジェクトに関する情報が提供されます。

dedicated-admin として、ユーザー定義プロジェクトに関するメトリクスに対して、一度に 1 つ以上の namespace をクエリーできます。

開発者として、メトリクスのクエリー時にプロジェクト名を指定する必要があります。選択したプロジェクトのメトリクスを表示するには、必要な権限が必要です。

5.5.3.1. 管理者としてのすべてのプロジェクトのメトリクスのクエリー

dedicated-admin またはすべてのプロジェクトの表示パーミッションを持つユーザーとして、メトリクス UI ですべてのデフォルト OpenShift Dedicated およびユーザー定義プロジェクトのメトリクスにアクセスできます。



注記

専任の管理者のみが、OpenShift Dedicated で提供されるサードパーティーの UI にアクセスできます。

前提条件

- **dedicated-admin** ロールまたはすべてのプロジェクトの表示パーミッションを持つユーザーとしてクラスターにアクセスできる。

手順

1. OpenShift Web コンソールの **Administrator** パースペクティブで、**Observe → Metrics** を選択します。
2. **Insert Metric at Cursor** を選択し、事前に定義されたクエリーの一覧を表示します。

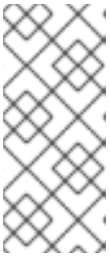
3. カスタムクエリーを作成するには、Prometheus クエリー言語 (PromQL) のクエリーを **Expression** フィールドに追加します。

4. 複数のクエリーを追加するには、**Add Query** を選択します。

5. クエリーを削除するには、クエリーの横にある  を選択してから **Delete query** を選択します。

6. クエリーの実行を無効にするには、クエリーの横にある  を選択してから **Disable query** を選択します。

7. **Run Queries** を選択し、作成したクエリーを実行します。クエリーからのメトリクスはプロットで可視化されます。クエリーが無効な場合、UI にはエラーメッセージが表示されます。



注記

大量のデータで動作するクエリーは、時系列グラフの描画時にタイムアウトするか、またはブラウザーをオーバーロードする可能性があります。これを回避するには、**Hide graph** を選択し、メトリクステーブルのみを使用してクエリーを調整します。次に、使用できるクエリーを確認した後に、グラフを描画できるようにプロットを有効にします。

8. オプション: ページ URL には、実行したクエリーが含まれます。このクエリーのセットを再度使用できるようにするには、この URL を保存します。

関連情報

- PromQL クエリーの作成に関する詳細は、Prometheus クエリーについてのドキュメント を参照してください。

5.5.3.2. 開発者が行うユーザー定義プロジェクトのメトリクスのクエリー

ユーザー定義のプロジェクトのメトリクスには、開発者またはプロジェクトの表示パーミッションを持つユーザーとしてアクセスできます。

Developer パースペクティブには、選択したプロジェクトの事前に定義された CPU、メモリー、帯域幅、およびネットワークパケットのクエリーが含まれます。また、プロジェクトの CPU、メモリー、帯域幅、ネットワークパケットおよびアプリケーションメトリクスについてカスタム Prometheus Query Language (PromQL) クエリーを実行することもできます。

注記

開発者は **Developer** パースペクティブのみを使用でき、**Administrator** パースペクティブは使用できません。開発者は、1 度に 1 つのプロジェクトのメトリクスのみをクエリーできます。開発者は OpenShift Dedicated モニタリングで提供されるサードパーティーの UI にアクセスできません。

前提条件

- 開発者として、またはメトリクスで表示しているプロジェクトの表示パーミッションを持つユーザーとしてクラスターへのアクセスがある。

- ユーザー定義プロジェクトのモニタリングを有効にしている。
- ユーザー定義プロジェクトにサービスをデプロイしている。
- サービスのモニター方法を定義するために、サービスの **ServiceMonitor** カスタムリソース定義 (CRD) を作成している。

手順

1. OpenShift Dedicated Web コンソールの **Developer** パースペクティブから、**Observe** → **Metrics** を選択します。
2. **Project:** 一覧でメトリクスで表示するプロジェクトを選択します。
3. **Select Query** 一覧からクエリーを選択するか、**Show PromQL** を選択してカスタム PromQL クエリーを実行します。

注記

Developer パースペクティブでは、1 度に 1 つのクエリーのみを実行できます。

関連情報

- PromQL クエリーの作成に関する詳細は、Prometheus クエリーについてのドキュメント を参照してください。
- 開発者または特権のあるユーザーとしてクラスター以外のメトリクスにアクセスする方法についての詳細は、開発者としてのユーザー定義プロジェクトのメトリクスのクエリー を参照してください。

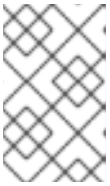
5.5.3.3. 視覚化されたメトリクスの使用

クエリーの実行後に、メトリクスが対話式プロットに表示されます。プロットの X 軸は時間を表し、Y 軸はメトリクスの値を表します。各メトリクスは、グラフ上の色付きの線で表示されます。プロットを対話的に操作し、メトリクスを参照できます。

手順

Administrator パースペクティブで、以下を行います。

1. 最初に、有効な全クエリーの全メトリクスがプロットに表示されます。表示されるメトリクスを選択できます。



注記

デフォルトでは、クエリーテーブルに、すべてのメトリクスとその現在の値を一覧表示する拡張ビューが表示されます。˅ を選択すると、クエリーの拡張ビューを最小にすることができます。

- クエリーからすべてのメトリクスを非表示にするには、クエリーの  をクリックし、**Hide all series** をクリックします。

- 特定のメトリクスを非表示にするには、クエリーテーブルに移動し、メトリクス名の横にある色の付いた四角をクリックします。

2. プロットをズームアップし、時間範囲を変更するには、以下のいずれかを行います。
   - プロットを水平にクリックし、ドラッグして、時間範囲を視覚的に選択します。
   - 左上隅のメニューを使用して、時間範囲を選択します。
3. 時間の範囲をリセットするには、**Reset Zoom** を選択します。
4. 特定の時点のすべてのクエリーの出力を表示するには、その時点のプロットの上にカーソルを合わせます。クエリーの出力はポップアップに表示されます。
5. プロットを非表示にするには、**Hide Graph** を選択します。

Developer パースペクティブ:

1. プロットをズームアップし、時間範囲を変更するには、以下のいずれかを行います。
   - プロットを水平にクリックし、ドラッグして、時間範囲を視覚的に選択します。
   - 左上隅のメニューを使用して、時間範囲を選択します。
2. 時間の範囲をリセットするには、**Reset Zoom** を選択します。
3. 特定の時点のすべてのクエリーの出力を表示するには、その時点のプロットの上にカーソルを合わせます。クエリーの出力はポップアップに表示されます。

関連情報

- PromQL インターフェイスの使用は、メトリクスのクエリー セクションを参照してください。
- モニタリング関連の問題のトラブルシューティング

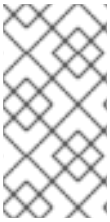
5.5.4. 次のステップ

- アラート

5.6. アラート

OpenShift Dedicated では、アラート UI を使用してアラート、サイレンス、およびアラートルールを管理できます。

- **アラートルール**アラートルールには、クラスター内の特定の状態を示す一連の条件が含まれます。アラートは、これらの条件が true の場合にトリガーされます。アラートルールには、アラートのルーティング方法を定義する重大度を割り当てることができます。
- **Alerts**アラートは、アラートルールで定義された条件が true の場合に発生します。アラートは、OpenShift Dedicated クラスター内で一連の状況が明らかであることを通知します。
- **サイレンス**。サイレンスをアラートに適用し、アラートの条件が true の場合に通知が送信されることを防ぐことができます。初期通知後はアラートをミュートにして、根本的な問題の解決に取り組むことができます。



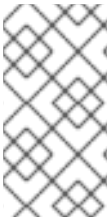
注記

アラート UI で利用可能なアラート、サイレンス、およびアラートルールは、アクセス可能なプロジェクトに関連付けられます。たとえば、**cluster-admin** 権限または **dedicated-admin** 権限でログインしている場合、すべてのアラート、サイレンス、およびアラートルールにアクセスできます。

5.6.1. Administrator および Developer パースペクティブでのアラート UI へのアクセス

アラート UI は、OpenShift Dedicated Web コンソールの 管理者パースペクティブおよび開発者パースペクティブからアクセスできます。

- Administrator パースペクティブで、**Observe → Alerting** を選択します。このパースペクティブのアラート UI の主なページには、**Alerts**、**Silences**、および **Alerting Rules** という 3 つのページがあります。
- Developer パースペクティブで、**Observe → <project_name> → Alerts** を選択します。このパースペクティブのアラートでは、サイレンスおよびアラートルールはすべて **Alerts** ページで管理されます。**Alerts** ページに表示される結果は、選択されたプロジェクトに固有のものです。



注記

開発者パースペクティブでは、**Project:** 一覧でアクセスできる OpenShift Dedicated のコアプロジェクトおよびユーザー定義プロジェクトを選択できます。ただし、**cluster-admin** 権限がない場合、OpenShift Dedicated のコアプロジェクトに関連するアラート、サイレンス、およびアラートルールは表示されません。

5.6.2. アラート、サイレンスおよびアラートルールの検索およびフィルター

アラート UI に表示されるアラート、サイレンス、およびアラートルールをフィルターできます。このセクションでは、利用可能なフィルターオプションのそれぞれについて説明します。

アラートフィルターについて
管理者 パースペクティブでは、アラート UI の **Alerts** ページには、デフォルトの OpenShift Dedicated およびユーザー定義プロジェクトに関連するアラートの詳細が示されます。このページには、各アラートの重大度、状態、およびソースの概要が含まれます。アラートが現在の状態に切り替わった時間も表示されます。

アラートの状態、重大度、およびソースでフィルターできます。デフォルトでは、**Firing** の **Platform** アラートのみが表示されます。以下では、それぞれのアラートフィルターオプションについて説明します。

- **Alert State** フィルター:
  - **Firing**アラート条件が true で、オプションの **for** の期間を経過しているためにアラートが実行されます。アラートは、条件が true である限り継続して実行されます。
  - **Pending**アラートはアクティブですが、アラート実行前のアラートルールに指定される期間待機します。
  - **Silenced**アラートは定義された期間についてサイレンスにされるようになりました。定義するラベルセレクターのセットに基づいてアラートを一時的にミュートします。一覧表示される値または正規表現のすべてに一致するアラートについては通知は送信されません。
- **Severity** フィルター:

- **Critical**アラートをトリガーした状態は重大な影響を与える可能性があります。このアラートには、実行時に早急な対応が必要となり、通常は個人または緊急対策チーム (Critical Response Team) に送信先が設定されます。
- **Warning**アラートは、問題の発生を防ぐために注意が必要になる可能性のある問題についての警告通知を提供します。通常、警告は早急な対応を要さないレビュー用にチケットシステムにルート指定されます。
- **Info**アラートは情報提供のみを目的として提供されます。
- **None**アラートには重大度が定義されていません。
- また、ユーザー定義プロジェクトに関連するアラートの重大度の定義を作成することもできます。

- **Source** フィルター:
  - **Platform**プラットフォームレベルのアラートは、デフォルトの OpenShift Dedicated プロジェクトにのみ該当します。これらのプロジェクトは OpenShift Dedicated のコア機能を提供します。
  - **User**ユーザーアラートはユーザー定義のプロジェクトに関連します。これらのアラートはユーザーによって作成され、カスタマイズ可能です。ユーザー定義のワークロードモニタリングはインストール後に有効にでき、独自のワークロードへの可観測性を提供します。

サイレンスフィルターについて
管理者 パースペクティブでは、アラート UI の **Silences** ページには、デフォルトの OpenShift Dedicated およびユーザー定義プロジェクトのアラートに適用されるサイレンスについての詳細が示されます。このページには、それぞれのサイレンスの状態の概要とサイレンスが終了する時間の概要が含まれます。

サイレンス状態でフィルターを実行できます。デフォルトでは、**Active** および **Pending** のサイレンスのみが表示されます。以下は、それぞれのサイレンス状態のフィルターオプションについて説明しています。

- **Silence State** フィルター:
  - **Active**サイレンスはアクティブで、アラートはサイレンスが期限切れになるまでミュートされます。
  - **Pending**サイレンスがスケジュールされており、アクティブな状態ではありません。
  - **Expired**アラートの条件が true の場合、サイレンスが期限切れになり、通知が送信されます。

アラートルールフィルターについて
管理者 パースペクティブでは、アラート UI の **Alerting Rules** ページには、デフォルトの OpenShift Dedicated およびユーザー定義プロジェクトに関連するアラートルールの詳細が示されます。このページには、各アラートルールの状態、重大度およびソースの概要が含まれます。

アラート状態、重大度、およびソースを使用してアラートルールをフィルターできます。デフォルトでは、**プラットフォーム**のアラートルールのみが表示されます。以下では、それぞれのアラートルールのフィルターオプションを説明します。

- **Alert State** フィルター:
  - **Firing**アラート条件が true で、オプションの **for** の期間を経過しているためにアラートが実行されます。アラートは、条件が true である限り継続して実行されます。

実行されます。アラートは、条件が true である期間継続して実行されます。

- **Pending**アラートはアクティブですが、アラート実行前のアラートルールに指定される期間待機します。
- **Silenced**アラートは定義された期間についてサイレンスにされるようになりました。定義するラベルセレクターのセットに基づいてアラートを一時的にミュートします。一覧表示される値または正規表現のすべてに一致するアラートについては通知は送信されません。
- **Not Firing**アラートは実行されません。

- **Severity** フィルター:
  - **Critical**アラートルールで定義される状態は重大な影響を与える可能性があります。true の場合、これらの状態には早急な対応が必要です。通常、ルールに関連するアラートは個別または緊急対策チーム (Critical Response Team) に送信先が設定されます。
  - **Warning**アラートルールで定義される状態は、問題の発生を防ぐために注意を要する場合があります。通常、ルールに関連するアラートは早急な対応を要さないレビュー用にチケットシステムにルート指定されます。
  - **Info**アラートルールは情報アラートのみを提供します。
  - **None**アラートルールには重大度が定義されていません。
  - ユーザー定義プロジェクトに関連するアラートルールのカスタム重大度定義を作成することもできます。
- **Source** フィルター:
  - **Platform**プラットフォームレベルのアラートルールは、デフォルトの OpenShift Dedicated プロジェクトにのみ該当します。これらのプロジェクトは OpenShift Dedicated のコア機能を提供します。
  - **User**ユーザー定義のワークロードアラートルールは、ユーザー定義プロジェクトに関連します。これらのアラートルールはユーザーによって作成され、カスタマイズ可能です。ユーザー定義のワークロードモニタリングはインストール後に有効にでき、独自のワークロードへの可観測性を提供します。

Developer パースペクティブでのアラート、サイレンスおよびアラートルールの検索およびフィルター

Developer パースペクティブのアラート UI の Alerts ページでは、選択されたプロジェクトに関連するアラートとサイレンスを組み合わせたビューを提供します。規定するアラートルールへのリンクが表示されるアラートごとに提供されます。

このビューでは、アラートの状態と重大度でフィルターを実行できます。デフォルトで、プロジェクトへのアクセスパーミッションがある場合は、選択されたプロジェクトのすべてのアラートが表示されます。これらのフィルターは **Administrator** パースペクティブについて記載されているフィルターと同じです。

5.6.3. アラート、サイレンスおよびアラートルールについての情報の取得

アラート UI は、アラートおよびそれらを規定するアラートルールおよびサイレンスについての詳細情報を提供します。

前提条件

- 開発者として、またはメトリクスで表示しているプロジェクトの表示パーミッションを持つユーザーとしてクラスターへのアクセスがある。

手順

Administrator パースペクティブでアラートについての情報を取得するには、以下を実行します。

1. OpenShift Dedicated Web コンソールを開き、**Observe → Alerting → Alerts** ページに移動します。
2. オプション: 検索一覧で **Name** フィールドを使用し、アラートを名前で検索します。
3. オプション: **Filter** 一覧でフィルターを選択し、アラートを状態、重大度およびソースでフィルターします。
4. オプション: 1 つ以上の **Name**、**Severity**、**State**、および **Source** 列ヘッダーをクリックし、アラートを並べ替えます。
5. **Alert Details** ページに移動するためにアラートの名前を選択します。このページには、アラートの時系列データを示すグラフが含まれます。また、以下を含むアラートについての情報も含まれます。
   - アラートの説明
   - アラートに関連付けられたメッセージ
   - アラートに割り当てられるラベル
   - アラートを規定するアラートルールへのリンク
   - アラートが存在する場合のアラートのサイレンス

Administrator パースペクティブでサイレンスについての情報を取得するには、以下を実行します。

1. **Observe → Alerting → Silences** ページに移動します。
2. オプション: **Search by name** フィールドを使用し、サイレンスを名前でフィルターします。
3. オプション: **Filter** 一覧でフィルターを選択し、サイレンスをフィルターします。デフォルトでは、**Active** および **Pending** フィルターが適用されます。
4. オプション: 1 つ以上の **Name**、**Firing Alerts**、および **State** 列ヘッダーをクリックしてサイレンスを並べ替えます。
5. **Silence Details** ページに移動するサイレンスの名前を選択します。このページには、以下の詳細が含まれます。
   - アラート仕様
   - 開始時間
   - 終了時間
   - サイレンス状態
   - 発生するアラートの数および一覧

Administrator パースペクティブでアラートルールについての情報を取得するには、以下を実行します。

1. **Observe → Alerting → Alerting Rules** ページに移動します。

2. オプション: **Filter** 一覧でフィルターを選択し、アラートルールを状態、重大度およびソースでフィルターします。

3. オプション: 1 つ以上の **Name**、**Severity**、**Alert State**、および **Source** 列ヘッダーをクリックし、アラートルールを並べ替えます。

4. アラートルールの名前を選択し、**Alerting Rule Details** ページに移動します。このページには、アラートルールに関する以下の情報が含まれます。

   - アラートルール名、重大度、および説明
   - アラートを発生させるための条件を定義する式
   - アラートを発生させるための条件が true である期間
   - アラートルールに規定される各アラートのグラフ。アラートを発生させる際に使用する値が表示されます。
   - アラートルールで規定されるすべてのアラートについての表

Developer パースペクティブでアラート、サイレンス、およびアラートルールについての情報を取得するには、以下を実行します。

1. **Observe** → **<project_name>** → **Alerts** ページに移動します。

2. アラート、サイレンス、またはアラートルールの詳細を表示します。

   - **Alert Details** を表示するには、アラート名の左側で **>** を選択し、一覧でアラートを選択します。
   - **Silence Details** を表示するには、**Alert Details** ページの **Silenced By** セクションでサイレンスを選択します。**Silence Details** ページには、以下の情報が含まれます。
     - アラート仕様
     - 開始時間
     - 終了時間
     - サイレンス状態
     - 発生するアラートの数および一覧
   - **Alerting Rule Details** を表示するには、**Alerts** ページのアラートの右側にある メニューの **View Alerting Rule** を選択します。

注記

選択したプロジェクトに関連するアラート、サイレンスおよびアラートルールのみが **Developer** パースペクティブに表示されます。

5.6.4. サイレンスの管理

アラートの発生時にアラートについての通知の受信を停止するためにサイレンスを作成できます。根本的な問題を解決する際に、初回の通知後にアラートをサイレンスにすることが役に立つ場合があります。

サイレンスの作成時に、サイレンスをすぐにアクティブにするか、または後にアクティブにするかを指定する必要があります。また、サイレンスの有効期限を設定する必要もあります。

既存のサイレンスを表示し、編集し、期限切れにすることができます。

5.6.4.1. アラートをサイレンスにする

特定のアラート、または定義する仕様に一致するアラートのいずれかをサイレンスにすることができます。

前提条件

- 開発者として、またはメトリクスで表示しているプロジェクトの **edit** パーミッションを持つユーザーとしてクラスターへのアクセスがある。

手順

特定のアラートをサイレンスにするには、以下を実行します。

- **Administrator** パースペクティブで、以下を行います。
  1. OpenShift Dedicated Web コンソールの **Observe → Alerting → Alerts** ページに移動します。
  2. サイレンスにする必要のあるアラートについて、右側の列で  を選択し、**Silence Alert** を選択します。**Silence Alert** フォームは、選択したアラートの事前に設定された仕様と共に表示されます。
  3. オプション: サイレンスを変更します。
  4. サイレンスを作成する前にコメントを追加する必要があります。
  5. サイレンスを作成するには、**Silence** を選択します。
- **Developer** パースペクティブ:
  1. OpenShift Dedicated Web コンソールの **Observe → <project_name> → Alerts** ページに移動します。
  2. アラート名の左側にある **>** を選択して、アラートの詳細を展開します。拡張されたビューでアラートの名前を選択し、アラートの **Alert Details** ページを開きます。
  3. **Silence Alert** を選択します。**Silence Alert** フォームが、選択したアラートの事前に設定された仕様と共に表示されます。
  4. オプション: サイレンスを変更します。
  5. サイレンスを作成する前にコメントを追加する必要があります。
  6. サイレンスを作成するには、**Silence** を選択します。

Administrator パースペクティブにアラート仕様を作成してアラートのセットをサイレンスにするには、以下を実行します。

1. OpenShift Dedicated Web コンソールの **Observe → Alerting → Silences** ページに移動します。
2. **Create Silence** を選択します。
3. **Create Silence** フォームで、アラートのスケジュール、期間、およびラベルの詳細を設定します。また、サイレンスのコメントを追加する必要もあります。
4. 直前の手順で入力したラベルセクターに一致するアラートのサイレンスを作成するには、**Silence** を選択します。

5.6.4.2. サイレンスの編集

サイレンスは編集することができます。 これにより、既存のサイレンスが期限切れとなり、変更された設定で新規のサイレンスが作成されます。

手順

Administrator パースペクティブでサイレンスを編集するには、以下を実行します。

1. **Observe → Alerting → Silences** ページに移動します。
2. 変更するサイレンスについて、最後の列の  を選択し、**Edit silence** を選択します。または、サイレンスについて **Silence Details** ページで **Actions → Edit Silence** を選択できます。
3. **Edit Silence** ページで変更を入力し、**Silence** を選択します。これにより、既存のサイレンスが期限切れとなり、選択された設定でサイレンスが作成されます。

Developer パースペクティブでサイレンスを編集するには、以下を実行します。

1. **Observe → <project_name> → Alerts** ページに移動します。
2. アラート名の左側にある **>** を選択して、アラートの詳細を展開します。拡張されたビューでアラートの名前を選択し、アラートの **Alert Details** ページを開きます。
3. そのページの **Silenced By** セクションでサイレンスの名前を選択し、サイレンスの **Silence Details** ページに移動します。
4. **Silence Details** ページに移動するサイレンスの名前を選択します。
5. サイレンスについて、**Silence Details** ページで **Actions → Edit Silence** を選択します。
6. **Edit Silence** ページで変更を入力し、**Silence** を選択します。これにより、既存のサイレンスが期限切れとなり、選択された設定でサイレンスが作成されます。

5.6.4.3. 有効期限切れにするサイレンス

サイレンスは有効期限切れにすることができます。サイレンスはいったん期限切れになると、永久に無効にされます。

手順

Administrator パースペクティブでサイレンスを期限切れにするには、以下を実行します。

1. **Observe → Alerting → Silences** ページに移動します。
2. 変更するサイレンスについて、最後の列の  を選択し、**Expire silence** を選択します。または、サイレンスの **Silence Details** ページで **Actions → Expire Silence** を選択できます。

Developer パースペクティブでサイレンスを期限切れにするには、以下を実行します。

1. **Observe → <project_name> → Alerts** ページに移動します。
2. アラート名の左側にある **>** を選択して、アラートの詳細を展開します。拡張されたビューでアラートの名前を選択し、アラートの **Alert Details** ページを開きます。
3. そのページの **Silenced By** セクションでサイレンスの名前を選択し、サイレンスの **Silence Details** ページに移動します。
4. **Silence Details** ページに移動するサイレンスの名前を選択します。
5. サイレンスの **Silence Details** ページで **Actions → Expire Silence** を選択します。

5.6.5. ユーザー定義プロジェクトのアラートルールの管理

OpenShift Dedicated モニタリングには、一連のデフォルトのアラートルールセットが同梱されます。クラスター管理者は、デフォルトのアラートルールを表示できます。

OpenShift Dedicated 4 では、ユーザー定義プロジェクトでアラートルールの作成、表示、編集、削除ができます。

重要

ユーザー定義プロジェクトのアラートルールの管理は、OpenShift Dedicated バージョン 4.11 以降でのみ利用できます。

アラートルールについての考慮事項

- デフォルトのアラートルールは OpenShift Dedicated クラスター専用に使用されます。
- 一部のアラートルールには、複数の意図的に同じ名前が含まれます。それらは同じイベントについてのアラートを送信しますが、それぞれ異なるしきい値、重大度およびそれらの両方が設定されます。
- 抑制 (inhibition) ルールは、高い重大度のアラートが実行される際に実行される低い重大度のアラートの通知を防ぎます。

5.6.5.1. ユーザー定義プロジェクトのアラートの最適化

アラートルールの作成時に以下の推奨事項を考慮して、独自のプロジェクトのアラートを最適化できます。

- **プロジェクト用に作成するアラートルールの数を最小限にします**。影響を与える条件についてユーザーに通知するアラートルールを作成します。影響を与えない条件について数多くのアラートを生成すると、関連性のあるアラートを認識することはより困難になります。

- **原因ではなく現象についてのアラートルールを作成します**。根本的な原因に関係なく状態について通知するアラートルールを作成します。次に、原因を調査できます。アラートルールのそれぞれが特定の原因にのみ関連する場合に、さらに多くのアラートルールが必要になります。そのため、いくつかの原因は見落される可能性があります。

- **アラートルールを作成する前にプランニングを行います**。重要な現象と、その発生時に実行するアクションを決定します。次に、現象別にアラートルールをビルドします。

- **クリアなアラートメッセージングを提供します**。アラートメッセージに現象および推奨されるアクションを記載します。

- **アラートルールに重大度レベルを含めます**。アラートの重大度は、報告される現象が生じた場合に取るべき対応によって異なります。たとえば、現象に個人または緊急対策チーム (Critical Response Team) による早急な対応が必要な場合、重大アラートをトリガーする必要があります。

- **アラートルーティングを最適化します**。ルールがデフォルトの OpenShift Dedicated メトリクスをクエリーしない場合に、**openshift-user-workload-monitoring** プロジェクトの Prometheus インスタンスに直接アラートルールをデプロイします。これにより、アラートルールの待ち時間が短縮され、モニタリングコンポーネントへの負荷が最小限に抑えられます。

警告

ユーザー定義プロジェクトのデフォルトの OpenShift Dedicated メトリクスは、CPU およびメモリーの使用状況、帯域幅の統計、およびパケットレートについての情報を提供します。ルールを **openshift-user-workload-monitoring** プロジェクトの Prometheus インスタンスに直接ロート指定する場合、これらのメトリクスをアラートルールに含めることはできません。アラートルールの最適化は、ドキュメントを参照し、モニタリング用のアーキテクチャーの全体像を把握している場合にのみ使用してください。

関連情報

- アラートの最適化に関する追加のガイドラインについては、Prometheus アラートのドキュメント を参照してください。

- OpenShift Dedicated 4 モニタリングアーキテクチャーに関する詳細は、モニタリングの概要 を参照してください。

5.6.5.2. ユーザー定義プロジェクトのアラートルールの作成

ユーザー定義のプロジェクトについてアラートルールを作成できます。これらのアラートルールは、選択したメトリクスの値に基づいてアラートを実行します。

前提条件

- ユーザー定義プロジェクトのモニタリングを有効にしている。

- アラートルールを作成する必要のある namespace の **monitoring-rules-edit** ロールを持つユーザーとしてログインします。
- OpenShift CLI (**oc**) がインストールされている。

手順

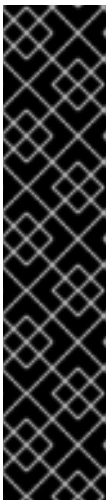
1. アラートルールの YAML ファイルを作成します。この例では、**example-app-alerting-rule.yaml** という名前です。
2. アラートルール設定を YAML ファイルに追加します。以下に例を示します。

注記

アラートルールの作成時に、同じ名前のルールが別のプロジェクトにある場合に、プロジェクトのラベルがこのアラートルールに対して適用されます。

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: example-alert
  namespace: ns1
spec:
  groups:
  - name: example
    rules:
    - alert: VersionAlert
      expr: version{job="prometheus-example-app"} == 0
```

この設定により、**example-alert** という名前のアラートルールが作成されます。アラートルールは、サンプルサービスによって公開される **version** メトリクスが **0** になるとアラートを実行します。



重要

ユーザー定義のアラートルールには、独自のプロジェクトおよびクラスターメトリクスのメトリクスを含めることができます。別のユーザー定義プロジェクトのメトリクスを含めることはできません。

たとえば、ユーザー定義プロジェクトの **ns1** のアラートルールには、**ns1**、および CPU およびメモリーメトリクスなどのクラスターメトリクスなどを含めることができます。ただし、ルールには **ns2** からのメトリクスを含めることはできません。

さらに、**openshift-*** コア OpenShift Dedicated プロジェクトのアラートルールを作成することはできません。デフォルトで OpenShift Dedicated モニタリングは、これらのプロジェクトのアラートルールのセットを提供します。

3. 設定ファイルをクラスターに適用します。

```
$ oc apply -f example-app-alerting-rule.yaml
```

アラートルールの作成には多少時間がかかります。

5.6.5.3. プラットフォームメトリクスをクエリーしないアラートルールの待ち時間の短縮

ユーザー定義プロジェクトのアラートルールがデフォルトのクラスターメトリクスをクエリーしない場合、**openshift-user-workload-monitoring** プロジェクトの Prometheus インスタンスにルールを直接デプロイすることができます。これにより、Thanos Ruler が必要でない場合にこれをバイパスすることで、アラートルールの待ち時間が短縮されます。これは、モニタリングコンポーネントの全体的な負荷を最小限に抑えるのに役立ちます。

警告

ユーザー定義プロジェクトのデフォルトの OpenShift Dedicated メトリクスは、CPU およびメモリーの使用状況、帯域幅の統計、およびパケットレートについての情報を提供します。ルールを **openshift-user-workload-monitoring** プロジェクトの Prometheus インスタンスに直接デプロイする場合、これらのメトリクスをアラートルールに含めることはできません。本セクションで説明した手順は、ドキュメントを参照し、モニタリング用のアーキテクチャーの全体像を把握している場合にのみ使用してください。

前提条件

- ユーザー定義プロジェクトのモニタリングを有効にしている。
- アラートルールを作成する必要のある namespace の **monitoring-rules-edit** ロールを持つユーザーとしてログインします。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. アラートルールの YAML ファイルを作成します。この例では、**example-app-alerting-rule.yaml** という名前です。
2. キーが **openshift.io/prometheus-rule-evaluation-scope** で、値が **leaf-prometheus** のラベルが含まれる YAML ファイルにアラートルール設定を追加します。以下に例を示します。

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: example-alert
  namespace: ns1
  labels:
    openshift.io/prometheus-rule-evaluation-scope: leaf-prometheus
spec:
  groups:
  - name: example
    rules:
    - alert: VersionAlert
      expr: version{job="prometheus-example-app"} == 0
```

そのラベルがある場合、アラートルールは **openshift-user-workload-monitoring** プロジェクトの Prometheus インスタンスにデプロイされます。ラベルが存在しない場合、アラートルールは Theanos Ruler にデプロイされます。

1. 設定ファイルをクラスターに適用します。

```
$ oc apply -f example-app-alerting-rule.yaml
```

アラートルールの作成には多少時間がかかります。

- OpenShift Dedicated 4 モニタリングアーキテクチャーに関する詳細は、モニタリングの概要を参照してください。

5.6.5.4. ユーザー定義プロジェクトのアラートルールへのアクセス

ユーザー定義プロジェクトのアラートルールを一覧表示するには、プロジェクトの **monitoring-rules-view** ロールが割り当てられている必要があります。

前提条件

- ユーザー定義プロジェクトのモニタリングを有効にしている。
- プロジェクトの **monitoring-rules-view** ロールを持つユーザーとしてログインしている。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **<project>** でアラートルールを一覧表示できます。

```
$ oc -n <project> get prometheusrule
```

2. アラートルールの設定を一覧表示するには、以下を実行します。

```
$ oc -n <project> get prometheusrule <rule> -o yaml
```

5.6.5.5. 単一ビューでのすべてのプロジェクトのアラートルールの一覧表示

クラスター管理者は、OpenShift Dedicated のコアプロジェクトおよびユーザー定義プロジェクトのアラートルールを単一ビューで一覧表示できます。

前提条件

- **cluster-admin** または **dedicated-admin** ロールが割り当てられたユーザーとしてクラスターにアクセスできる。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **Administrator** パースペクティブで、**Observe → Alerting → Alerting Rules** に移動します。
2. **Filter** ドロップダウンメニューで、**Platform** および **User** ソースを選択します。

注記

Platform ソースはデフォルトで選択されます。

5.6.5.6. ユーザー定義プロジェクトのアラートルールの削除

ユーザー定義プロジェクトのアラートルールを削除できます。

前提条件

- ユーザー定義プロジェクトのモニタリングを有効にしている。
- アラートルールを作成する必要のある namespace の **monitoring-rules-edit** ロールを持つユーザーとしてログインします。
- OpenShift CLI (**oc**) がインストールされている。

手順

- **<namespace>** のルール **<foo>** を削除するには、以下を実行します。

```
$ oc -n <namespace> delete prometheusrule <foo>
```

関連情報

- Alertmanager ドキュメント を参照してください。

関連情報

- OpenShift Dedicated モニタリングアーキテクチャーに関する詳細は、Monitoring overview を参照してください。
- アラートルールの詳細は、Alertmanager ドキュメント を参照してください。
- 再ラベル付けの動作に関する詳細は、、Prometheus の再ラベル付けに関するドキュメント を参照してください。
- アラートの最適化に関する追加のガイドラインについては、Prometheus アラートのドキュメント を参照してください。

5.6.6. ユーザー定義のアラートルーティングの Alertmanager へのカスタム設定の適用

ユーザー定義のアラートルーティング専用の Alertmanager の別のインスタンスを有効にしている場合、**openshift-user-workload-monitoring** namespace で **alertmanager-user-workload** シークレットを編集して Alertmanager のこのインスタンスの設定を上書きできます。

前提条件

- **cluster-admin** または **dedicated-admin** ロールが割り当てられたユーザーとしてクラスターにアクセスできる。

手順

1. 現在アクティブな Alertmanager 設定をファイル **alertmanager.yaml** に出力します。

```
$ oc -n openshift-user-workload-monitoring get secret alertmanager-user-workload --
template='{{ index .data "alertmanager.yaml" }}' | base64 --decode > alertmanager.yaml
```

2. **alertmanager.yaml** で設定を編集します。

```
route:
  receiver: Default
  group_by:
  - name: Default
  routes:
  - matchers:
    - "service = prometheus-example-monitor" 1
    receiver: <receiver> 2
receivers:
- name: Default
- name: <receiver>
#  <receiver_configuration>
```

1 ルートに一致するアラートを指定します。この例では、**service="prometheus-example-monitor"** ラベルの付いたすべてのアラートを示しています。

2 アラートグループに使用するレシーバーを指定します。

3. 新規設定をファイルで適用します。

```
$ oc -n openshift-user-workload-monitoring create secret generic alertmanager-user-
workload --from-file=alertmanager.yaml --dry-run=client -o=yaml |  oc -n openshift-user-
workload-monitoring replace secret --filename=-
```

関連情報

- PagerDuty についての詳細は、PagerDuty の公式サイト を参照してください。
- **service_key** を取得する方法については、PagerDuty Prometheus Integration Guide を参照してください。
- 各種のアラートレシーバー経由でアラートを設定する方法については、Alertmanager configuration を参照してください。

5.6.7. 次のステップ

- モニタリングダッシュボードの確認

5.7. モニタリングダッシュボードの確認

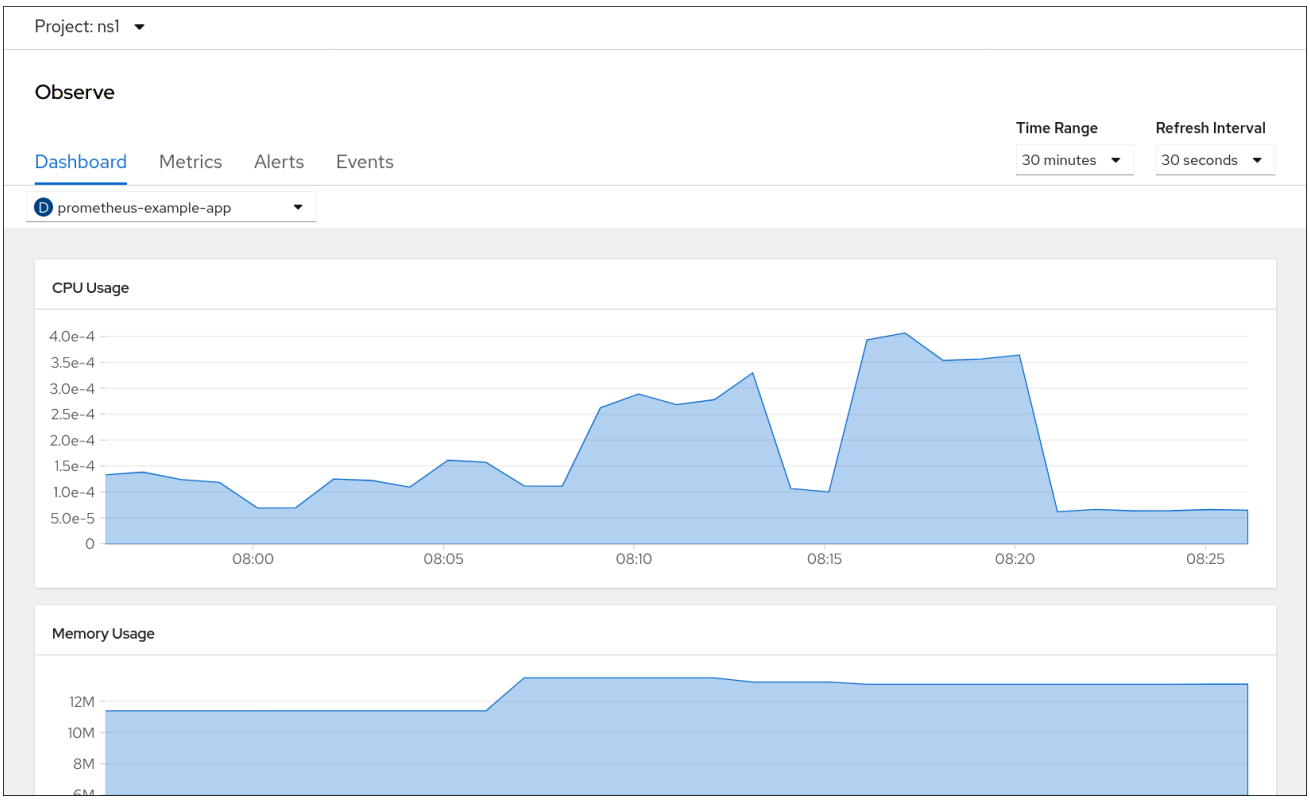
OpenShift Dedicated は、ユーザー定義プロジェクトの状態を理解するのに役立つモニタリングダッシュボードを提供します。

Developer パースペクティブでは、選択されたプロジェクトの以下の統計を提供するダッシュボードにアクセスできます。

- CPU usage (CPU の使用率)

- メモリー使用量
- 帯域幅に関する情報
- パケットレート情報

図5.1 Developer パースペクティブのダッシュボードの例



注記

Developer パースペクティブでは、1 度に 1 つのプロジェクトのみのダッシュボードを表示できます。

5.7.1. 開発者が行うモニタリングダッシュボードの確認

Developer パースペクティブでは、選択されたプロジェクトに関連するダッシュボードを表示できます。ダッシュボード情報を表示するには、プロジェクトをモニターするためのアクセスが必要になります。

前提条件

- **dedicated-admin** として、またはダッシュボードで表示しているプロジェクトの表示パーミッションを持つユーザーとしてクラスターにアクセスできる必要があります。

手順

1. OpenShift Dedicated Web コンソールの **Developer** パースペクティブで、**Observe** → **Dashboard** に移動します。
2. **Project:** 一覧でプロジェクトを選択します。
3. **All Workloads** 一覧でワークロードを選択します。

4. 必要に応じて、**Time Range** 一覧でグラフの時間範囲を選択します。
5. オプション: **Refresh Interval** を選択します。
6. 特定の項目についての詳細情報を表示するには、ダッシュボードの各グラフにカーソルを合わせます。

5.7.2. 次のステップ

- モニタリング関連の問題のトラブルシューティング

5.8. モニタリング関連の問題のトラブルシューティング

ユーザー定義プロジェクトのモニタリングに関する一般的な問題のトラブルシューティング手順を参照してください。

5.8.1. ユーザー定義プロジェクトのメトリクスが利用できない理由の判別

ユーザー定義プロジェクトのモニタリング時にメトリクスが表示されない場合は、以下の手順を実行して問題のトラブルシューティングを実行します。

手順

1. メトリクス名に対してクエリーを実行し、プロジェクトが正しいことを確認します。
   a. OpenShift Container Platform Web コンソールの **Developer** パースペクティブから、**Observe → Metrics** を選択します。
   b. **Project:** 一覧でメトリクスで表示するプロジェクトを選択します。
   c. **Select Query** 一覧からクエリーを選択するか、**Show PromQL** を選択してカスタム PromQL クエリーを実行します。
   Select Query ペインには、メトリクス名が表示されます。

   クエリーはプロジェクトごとに実行される必要があります。表示されるメトリクスは、選択したプロジェクトに関連するメトリクスです。
2. メトリックが必要な Pod がアクティブにメトリックを提供していることを確認します。以下の **oc exec** コマンドを Pod で実行し、**podIP**、**port**、および **/metrics** をターゲットにします。

```
$ oc exec <sample_pod> -n <sample_namespace> -- curl <target_pod_IP>:<port>/metrics
```

注記

curl がインストールされている Pod でコマンドを実行する必要があります。

以下の出力例は、有効なバージョンのメトリクスを含む結果を示しています。

出力例

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
# HELP version Version information about this binary-- --:--:-- --:--:--     0
```

```
# TYPE version gauge
version{version="v0.1.0"} 1
100   102  100   102    0     0  51000      0 --:--:-- --:--:-- --:--:-- 51000
```

無効な出力は、対応するアプリケーションに問題があることを示しています。

3. **PodMonitor** CRD を使用している場合は、**PodMonitor** CRD がラベル一致を使用して適切な Pod を参照するよう設定されていることを確認します。詳細は、Prometheus Operator のドキュメントを参照してください。

4. **ServiceMonitor** CRD を使用し、Pod の **/metrics** エンドポイントがメトリクスデータを表示している場合は、以下の手順を実行して設定を確認します。

   a. サービスが正しい **/metrics** エンドポイントを参照していることを確認します。この出力のサービス **labels** は、後続の手順でサービスが定義するサービスモニターの **labels** と **/metrics** エンドポイントと一致する必要があります。

   ```
   $ oc get service
   ```

   出力例

   ```
   apiVersion: v1
   kind: Service (1)
   metadata:
     labels: (2)
       app: prometheus-example-app
     name: prometheus-example-app
     namespace: ns1
   spec:
     ports:
     - port: 8080
       protocol: TCP
       targetPort: 8080
       name: web
     selector:
       app: prometheus-example-app
     type: ClusterIP
   ```

   (1) これがサービス API であることを指定します。

   (2) このサービスに使用されるラベルを指定します。

   b. **serviceIP**、**port**、および **/metrics** エンドポイントをクエリーし、以前に Pod で実行した **curl** コマンドと同じメトリクスがあるかどうかを確認します。

      i. 以下のコマンドを実行してサービス IP を見つけます。

      ```
      $ oc get service -n <target_namespace>
      ```

      ii. **/metrics** エンドポイントをクエリーします。

      ```
      $ oc exec <sample_pod> -n <sample_namespace> -- curl <service_IP>:
      <port>/metrics
      ```

以下の例では、有効なメトリクスが返されます。

出力例

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   102  100   102    0     0  51000      0 --:--:-- --:--:-- --:--:--   99k
# HELP version Version information about this binary
# TYPE version gauge
version{version="v0.1.0"} 1
```

c. ラベルのマッチングを使用して、**ServiceMonitor** オブジェクトが必要なサービスを参照するように設定されていることを確認します。これを実行するには、**oc get service** 出力の **Service** オブジェクトを **oc get servicemonitor** 出力の **ServiceMonitor** オブジェクトと比較します。メトリックを表示するには、ラベルが一致している必要があります。
たとえば、直前の手順の **Service** オブジェクトに **app: prometheus-example-app** ラベルがあり、**ServiceMonitor** オブジェクトに同じ **app: prometheus-example-app** 一致ラベルがある点に注意してください。

5. すべて有効になっていても、メトリクスが利用できない場合は、サポートチームにお問い合わせください。