



ROSA Workshop

Welcome

ROSA とコンテナアプリケーションの開発に活用する人向けのワークショップです。



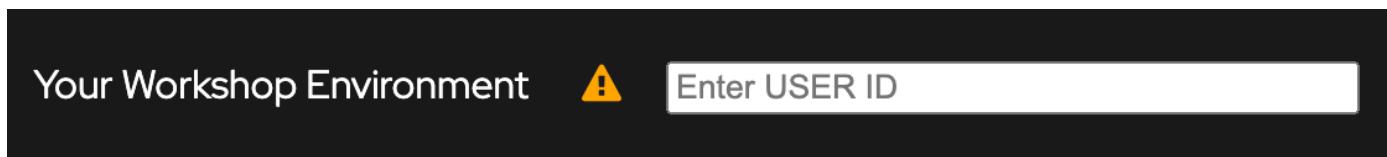
ワークショップ環境

Contents

1. 準備
2. ツールのインストール
 - 2.1. OpenShift CLI のインストール
 - 2.2. skopeo のインストール
3. 各種環境のURLとログイン方法
 - 3.1. OpenShift コンソールのログイン
 - 3.2. OpenShift コマンドラインターミナルの利用
 - 3.3. ローカル環境でのOpenShift CLIの利用

1. 準備

このガイドの右上の「Enter User ID」と薄く表示しているテキストフィールドにユーザID（例: user1）を入力してください。



このような表示になればOKです。入力するUserIDを間違えてしまった場合は、HTTPクエリパラメータ「USERID=user1」の部分を修正してください。



ガイドを少し下にスクロールして、ROSA APIのURLとAWSコンソールのURLが、この図のような書式になっているか確認してください。



2. ツールのインストール

2.1. OpenShift CLI のインストール

演習環境が利用可能でない場合は、

https://docs.redhat.com/ja/documentation/openshift_container_platform/4.18/html/cli_tools/openshift-cli-oc#cli-installing-cli_cli-developer-commands に従ってインストールしてください。

演習環境が利用可能な場合は、<https://console-openshift-console.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com/command-line-tools> にアクセスして必要な CLIをインストールしてください。

2.2. skopeo のインストール

<https://www.redhat.com/ja/topics/containers/what-is-skopeo> を参考にしてインストールしてください。

3. 各種環境のURLとログイン方法

ROSA コンソール	https://console-openshift-console.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com
ROSA API	https://api.rosa.keomizorosa.guj9.p3.openshiftapps.com:443
OpenShift コンソール（オンプレ環境想定）	https://console-openshift-console.apps.xxx
AWS コンソール	https://714932348383.siginin.aws.amazon.com/console
Gitリポジトリ	ワークショップの共有メモを参照してください

3.1. OpenShiftコンソールのログイン

1. OpenShiftコンソールにブラウザでアクセス
2. ログイン画面で「users-htpasswd」をクリック
3. ユーザ名に「user1」、パスワードに「openshift」を入力してログイン

3.2. OpenShift コマンドラインインターミナルの利用

OpenShift コマンドラインインターミナルの利用が可能です。最初に利用するときにはコマンドラインインターミナルのPodを実行するためのプロジェクトを一つ作成します。

1. ROSAコンソールにアクセスし、画面右上の「>_」をクリック
2. 初期化画面で「CreateProject」を選択し、プロジェクト名に「user1-term」と入力
3. 「Start」ボタンをクリック

3.3. ローカル環境でのOpenShift CLIの利用

1. OpenShift API 経由でログイン

```
oc login -u user1 -p openshift  
https://api.keomizorosa.guj9.p3.openshiftapps.com:443
```

BASH



Contents

1. 準備
2. ツールのインストール
 - 2.1. OpenShift CLI のインストール
 - 2.2. skopeo のインストール
3. 各種環境のURLとログイン方法
 - 3.1. OpenShiftコンソールのログイン
 - 3.2. OpenShift コマンドラインターミナルの利用
 - 3.3. ローカル環境でのOpenShift CLIの利用



[デモ] ROSA HCPクラスターの作成

Contents

- デモの概要
- 前準備
- ROSA HCPクラスターに必要なAWS VPCやIAMロールの作成
- ROSA HCPクラスターの作成
- ROSA HCPクラスターへのアクセス
- ROSA HCPクラスターの削除

デモの概要

このモジュールでは、インストラクターがROSA HCPクラスター作成方法の概要をご紹介します。

ROSA HCP 有効化



前準備

ROSAは、Red Hatによるお客様の既存AWSアカウントへのデプロイを可能にするモデルを提供します。Red Hatはセキュリティを考慮して、[AWS Security Token Service \(STS\)](#)を使用することを、ROSA HCPクラスター作成の前提条件としています。STSを使用したROSA HCPクラスター作成の前提条件は、次のドキュメントをご参照ください。

- [STSを使用したROSAのAWS前提条件](#)
- [STSを使用するROSAクラスターのIAMリソースについて](#)
- [必要なAWSサービスクォータ](#)

ROSA HCPクラスターを作成するには、AWSアカウントと[Red Hatアカウント](#)を利用します。アカウントがない場合は、アカウントを作成する必要があります。

[AWSコンソール](#)でROSAのHCPサービスを有効にします。ROSAにはClassicとHCPの2つのスタイルがありますが、ROSA HCPサービスを有効化するようにしてください。次のような画面になれば、ROSA HCPサービスが有効になっています。

ROSA の前提条件を確認する [Info](#)

このページでは、アカウントが Red Hat OpenShift Service on AWS (ROSA) クラスターを作成するための前提条件を満たしているかどうかを確認します。

ROSA の有効化 [Info](#)

ROSA は AWS と Red Hat が共同で管理します。Red Hat との接続を作成するには、ROSA を有効にします。この接続は、計測と請求に必要です。

 ROSA を有効にすると、次の 2 種類のクラスターを作成できます。



- ROSA Classic: AWS アカウントでホストされるクラスターコントロールプレーンインフラストラクチャ。
- ROSA HCP (ROSA とホスト型コントロールプレーン (HCP)): Red Hat が所有する AWS アカウントでホストされるクラスターコントロールプレーンインフラストラクチャ。ROSA HCP は現在プレビュー段階であり、本番環境のワークロードには使用しないでください。ROSA HCP のレビューに関する用語については、次の「トライアルとレビュー」を参照してください。

クラスターを作成するときに、どのコントロールプレーンモデルを使用するかを選択します。[詳細](#) 

 ROSA HCP と ROSA Classic はすでに有効になっています。

最終チェック日: December 09, 2023 at 05:06 (UTC)

▶ AWS Organizations の管理者: 組織全体で ROSA Classic を有効にする

AWS CLIを [インストール](#)して、[設定](#)します。このとき、`~/.aws/credentials` で、次の情報を設定する必要があります。

- `aws_access_key_id`
- `aws_secret_key`

また、デフォルトで利用するリージョンを指定する場合、`~/.aws/config` で設定できます。

```
$ cat ~/.aws/config
[default]
region = us-east-2
```

そして、次のコマンドを実行して、AWS APIをクエリーし、AWS CLIがインストールされて正しく設定されていることを確認します。次のような出力が表示されれば、AWS CLIの設定が完了しています。

```
$ aws sts get-caller-identity
{
```

```
"UserId": "AIDXXXXXXXXXXXXXX",
"Account": "XXXXXXXXXX",
"Arn": "arn:aws:iam::XXXXXXXXX:user/testuser01"
}
```

AWSアカウントを利用してELBをインストールしたことがない場合、次のコマンドを実行してELB用のロールを作成します。

```
$ aws iam create-service-linked-role --aws-service-name
"elasticloadbalancing.amazonaws.com"
```

NOTE

AWSのコンソールでELB サービスにリンクされたロールセクションに「AWSServiceRoleForElasticLoadBalancing はすでに存在します。」と表示されている場合は、上記コマンドは実施不要です。

AWSサービスクォータがROSA HCPクラスター用の要件を満たしており、ELB用のロールが作成されていると、次の画面が表示されます。ここから画面一番下にある「Red Hatに進む」ボタンをクリックして、AWSとRed Hatアカウントをリンクします。

Service Quotas [Info](#)

ROSA を使用するには、クォータを増やす必要がある場合があります。

☑ クォータは ROSA の要件を満たしています。

最終チェック日: December 09, 2023 at 05:06 (UTC)

ELB サービスにリンクされたロール [Info](#)

ROSA は、Elastic Load Balancing (ELB) サービスにリンクされたロールを使用して、ユーザーに代わって AWS のサービスを呼び出します。アカウントにこのロールがない場合は、ロールが作成されます。

☑ AWSServiceRoleForElasticLoadBalancing はすでに存在します。

[ロールの表示](#)

最終チェック日: December 09, 2023 at 05:06 (UTC)

次のステップ

Red Hat に進むを選択して、これらの前提条件のステップを完了します。

- AWS と Red Hat アカウントのリンク | [Info](#)
- AWS アカウント全体のロールの作成 | [Info](#)

完了したら、Red Hat コンソールでクラスターを作成できます。

[キャンセル](#)

[Red Hat に進む](#)

「Red Hatに進む」ボタンをクリックすると、次の画面が表示されます。ここで、Red Hatアカウントにログインしていない場合、ログイン画面が表示されますので、ログインしておきます。ここで「I have read and agreed to the terms and conditions」にチェックを入れて、「Connect accounts」ボタンをクリックすると、AWSとRed Hatアカウントのリンクが完了します。

Red Hat Hybrid Cloud Console

Services ▾

Search for services

Preview off

Hirofumi Kojima

connect

Complete your account connection

Red Hat account number 13XXXX

AWS account ID 92XXXX

Subscription(s) Red Hat OpenShift Service on AWS with Hosted Control Plane

Terms and conditions *

United States (English)

I have read and agreed to the [terms and conditions](#).

Connect accounts Cancel

リンクが無事に完了すると、次のメッセージが画面上に表示されます。

Congratulations, your Red Hat and AWS accounts are linked
Welcome to the Red Hat Hybrid Cloud Console. If you cannot access production tools
for a subscription that you have purchased, please wait 5 minutes and confirm
your subscription at subscription inventory. Here you can configure or manage Red
Hat OpenShift Cluster Manager.

続いて、ROSA CLIをダウンロードして、PATHを設定します。

Downloads

Red Hat Insights

Red Hat OpenShift Service on AWS command-line interface (rosa)

Linux

x86_64

Download

```
$ chmod +x rosa
$ sudo mv rosa /usr/local/bin/
$ rosa version
1.2.31
```

ROSA CLIでRed Hatアカウントにログインします。アクセストークンは、下記のコマンド実行により表示されているURLから入手できます。

```
$ rosa login
To login to your Red Hat account, get an offline access token at
https://console.redhat.com/openshift/token/rosa
? Copy the token and paste it here: *****
I: Logged in as '<Red Hatアカウントのユーザー名>' on 'https://api.openshift.com'
```

次のコマンドを実行して、AWSおよびRed Hatの認証情報が正しく設定されていることを確認します。AWSアカウントID、デフォルトのリージョン、および、ARNが設定した内容と一致していることを確認します。

```
$ rosa whoami
AWS Account ID: XXXXXXXXXXXX
AWS Default Region: us-east-2
AWS ARN: arn:aws:iam::XXXXXXXXXX:user/testuser01
OCM API: https://api.openshift.com
OCM Account ID: XXXXXXXXXXXX
OCM Account Name: Hiforumi Kojima
OCM Account Username: <Red Hatアカウントのユーザー名>
OCM Account Email: hkojima@redhat.com
OCM Organization ID: XXXXXXXXXXXX
OCM Organization Name: Hiforumi Kojima
OCM Organization External ID: XXXXXXXXXXXX
```

ROSA CLIを利用して、最新のOpenShift CLI (oc) をインストールします。下記の手順は、Linux版の openshift-client のtarファイルを解凍する例です。

```
$ rosa download openshift-client
$ tar xvf openshift-client-linux.tar.gz
$ sudo mv oc /usr/local/bin/
$ rosa verify openshift-client
I: Verifying whether OpenShift command-line tool is available...
I: Current OpenShift Client Version: 4.14.5
```

ROSA HCPクラスターに必要なAWS VPCやIAMロールの作成

ROSA HCPクラスター作成に必要なAWS VPCを作成します。AWS VPCはAWSコンソールから作成できますが、TerraformでVPCを作成するためのテンプレートが用意されているので、それを利用することもできます。TerraformでVPCを作成するためには、次のコマンドを実行します。 `terraform plan` コマンドでは、HCPクラスターが作成できるAWSリージョンのIDを指定する必要があります。

```
$ git clone https://github.com/openshift-cs/terraform-vpc-example
$ cd terraform-vpc-example
$ terraform init

$ terraform plan -out rosa.tfplan -var region=us-east-2
$ terraform apply rosa.tfplan
```

ここではAWSリージョンのシングルアベイラビリティゾーン(SingleAZ)に、ROSA HCPクラスターのワーカーノードを作成することを前提として、VPCを作成しています。マルチアベイラビリティゾーン(MultiAZ)にワーカーノードを作成して、AZ単位で冗長性を確保したい場合は、`terraform plan` コマンド実行時に、リージョンにあるAZのIDを3つ指定します。

```
$ terraform plan -out rosa.tfplan -var region=us-east-2 \
-var single_az_only=false \
-var 'subnet_azs=["use2-az1", "use2-az2", "use2-az3"]'
```

AWS VPCの作成が完了していることを、`terraform output` コマンドで確認します。このとき、作られるサブネットのIDがHCPクラスター作成に必要となるので、`export` コマンドで `SUBNET_IDS` 変数に代入しておきます。

```
$ export SUBNET_IDS=$(terraform output -raw cluster-subnets-string)
$ echo $SUBNET_IDS
subnet-0fb4f75f448b5499c,subnet-0087cb7bb3f628793
```

AWS IAMロールを、ROSA CLIを使って作成します。これによって、ROSA専用の既存のIAMポリシーが、作成されたIAMロールに自動的に割り当てられます。

```
$ rosa create account-roles --mode auto --hosted-cp --yes
I: Logged in as '<Red Hatアカウントのユーザー名>' on 'https://api.openshift.com'
I: Validating AWS credentials...
I: AWS credentials are valid!
I: Validating AWS quota...
I: AWS quota ok. If cluster installation fails, validate actual AWS resource usage
against https://docs.openshift.com/rosa/roса_getting_started/roса-required-aws-
service-quotas.html
I: Verifying whether OpenShift command-line tool is available...
I: Current OpenShift Client Version: 4.14.5
I: Creating account roles
I: Creating hosted CP account roles using 'arn:aws:iam::XXXXXXXXXX:user/testuser01'
I: Created role 'ManagedOpenShift-HCP-ROSA-Installer-Role' with ARN
'arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Installer-Role'
I: Created role 'ManagedOpenShift-HCP-ROSA-Support-Role' with ARN
'arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Support-Role'
I: Created role 'ManagedOpenShift-HCP-ROSA-Worker-Role' with ARN
'arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Worker-Role'
I: To create an OIDC Config, run the following command:
    rosa create oidc-config
```

AWSコンソールから、ROSAクラスター作成に必要なIAMロールが作成されていることを確認できます。このIAMロールは、複数のクラスターにわたって使用されます。

ロール (19) <small>情報</small>		
C 削除 ロールを作成		
<input type="text" value="ManagedOpenShift"/>	<small>X</small>	3一致
ロール名	信頼されたエンティティ	最後のアクティビティ
ManagedOpenShift-HCP-ROSA-Installer-Role	アカウント: 71 [REDACTED]	-
ManagedOpenShift-HCP-ROSA-Support-Role	アカウント: 71 [REDACTED]	-
ManagedOpenShift-HCP-ROSA-Worker-Role	AWS のサービス: ec2	-

ROSA HCPクラスターの Operatorが、専用のIAMロールを作成して利用するために必要となる、OpenID Connectの設定を作成します。

```
$ rosa create oidc-config --mode auto --yes
I: Setting up managed OIDC configuration
I: To create Operator Roles for this OIDC Configuration, run the following command
and remember to replace <user-defined> with a prefix of your choice:
    rosa create operator-roles --prefix <user-defined> --oidc-config-id
280rqooan3kqqn4o1us1ip073fsrkfm8
If you are going to create a Hosted Control Plane cluster please include '--hosted-
cp'
I: Creating OIDC provider using 'arn:aws:iam::XXXXXXXXXX:user/testuser01'
I: Created OIDC provider with ARN 'arn:aws:iam::XXXXXXXXXX:oidc-provider/rh-
oidc.s3.us-east-1.amazonaws.com/280rqooan3kqqn4o1us1ip073fsrkfm8'
```

ROSA CLIを使ってHCPクラスターを作成する時に必要となるIAMロールは、次の2種類です。

- アカウントロール(account roles): Red Hat SREチームがクラスターの管理に利用します。
- オペレーター ロール(operator roles): OpenShiftのOperatorがAWSリソースの管理に利用します。 クラスターごとに個別に作成されます。

これらのIAMロールに割り当てられるIAMポリシーの詳細は、[公式ドキュメント](#)をご参照下さい。

ROSA HCPクラスターの作成

ROSA HCPクラスターの作成コマンドを実行します。 `version` オプションを指定しない場合、自動的に最新版のOpenShiftが利用されます。 `dry-run` オプションを付けると、正常に実行可能かどうかを事前確認できます。次のコマンドによって `hcp-01` という名前のROSA HCPクラスター(`m5.xlarge` のワーカーノード)が自動的に作成されます。

このとき、指定するサブネットIDによって、SingleAZ構成とMultiAZ構成のどちらになるかが自動的に決まります。今まで実行してきたコマンド例だと、パブリック/プライベートサブネットIDを1つずつ指定することになるので、ワーカーノード2台のSingleAZ構成としてROSA HCPクラスターが作成されます。

--oidc-config-id には、先程実行した `rosa create oidc-config` コマンドの出力結果に表示されたoidc-config-id (例:Oikrpnt0nv000c000trn00renuakl0mn) を指定します。

```
export OIDC_CONFIG_ID=出力結果のoidc config id
export CLUSTER_NAME=AWS Orgnization で一意の名前
$ rosa create cluster --cluster-name=$CLUSTER_NAME --mode=auto --yes --hosted-cp \
--oidc-config-id $OIDC_CONFIG_ID \
--subnet-ids=$SUBNET_IDS --region=us-east-2 \
--compute-machine-type m5.xlarge --version 4.14.2
```

ROSA HCPクラスターのワーカーノードの 最小台数は2台、最大台数は500台です。これは、SingleAZ, MultiAZで共通しています。

MultiAZ構成のワーカーノードを利用する場合は、SingleAZ構成の時と同様に、`rosa create cluster` コマンド実行時に `subnet-ids` オプションで、MultiAZ構成を取るためのパブリック/プライベートサブネットを指定します。

AWSリージョンで利用可能なEC2インスタンスのタイプにも依存しますが、最小で c5.xlarge(4vCPU/RAM8GiB)のインスタンスを、ワーカーノードとして利用できます。ワーカーノードのデフォルトのEC2インスタンスタイプは、m5.xlarge(4vCPU/RAM16GiB)です。ROSA HCP クラスター作成時に、オプションとしてワーカーノードのインスタンスタイプ(例: `--compute-machine-type c5.xlarge`)や台数(例: `--replicas 3`)を指定できます。

対話モードによりオプションを指定しながら作成することもできます。次のコマンドでは、それぞれデフォルトのパラメーターを指定しています。

```
export OIDC_CONFIG_ID=出力結果のoidc config id
export CLUSTER_NAME=AWS Orgnization で一意の名前
$ rosa create cluster --cluster-name=$CLUSTER_NAME --mode=auto --hosted-cp \
--subnet-ids=$SUBNET_IDS --region=us-east-2 --version 4.18.10

I: Using 'XXXXXXXXXX' as billing account
I: To use a different billing account, add --billing-account XXXXXXXXXXXX to previous
command
I: Using arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Installer-Role for
the Installer role
I: Using arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Worker-Role for the
Worker role
```

I: Using arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Support-Role for the Support role

```
? OIDC Configuration ID: $OIDC_CONFIG_ID | https://rh-oidc.s3.us-east-1.amazonaws.com/280rqooan3kqqn4o1us1ip073fsrkfm8
? Tags (optional):
? AWS region: us-east-2
? PrivateLink cluster: No
? Machine CIDR: 10.0.0.0/16
? Service CIDR: 172.30.0.0/16
? Pod CIDR: 10.128.0.0/14
? Enable Customer Managed key: No
? Compute nodes instance type: [Use arrows to move, type to filter, ? for more help]
    m5dn.metal
    m5.metal
    m5n.metal
> m5.xlarge
    m5zn.metal
    m6a.12xlarge
    m6a.16xlarge
? Compute nodes instance type: m5.xlarge
? Enable autoscaling: No
? Compute nodes: 2
? Host prefix: 23
? Enable FIPS support: No
? Encrypt etcd data: No
? Disable Workload monitoring: No
? Use cluster-wide proxy: No
? Additional trust bundle file path (optional):
? Enable audit log forwarding to AWS CloudWatch: No
```

I: Creating cluster 'hcp-01'

I: To create this cluster again in the future, you can run:

```
rosa create cluster --cluster-name hcp-01 --sts --mode auto --role-arn arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Installer-Role --support-role-arn arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Support-Role --worker-iam-role arn:aws:iam::XXXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-Worker-Role --operator-roles-prefix hcp-01-g8r1 --oidc-config-id 280rqooan3kqqn4o1us1ip073fsrkfm8 --region us-east-2 --version 4.14.2 --replicas 2 --compute-machine-type m5.xlarge --machine-cidr 10.0.0.0/16 --service-cidr 172.30.0.0/16 --pod-cidr 10.128.0.0/14 --host-prefix 23 --subnet-ids subnet-0fb4f75f448b5499c,subnet-0087cb7bb3f628793 --hosted-cp
```

I: To view a list of clusters and their status, run 'rosa list clusters'

I: Cluster 'hcp-01' has been created.

I: Once the cluster is installed you will need to add an Identity Provider before you can login into the cluster. See 'rosa create idp --help' for more information.

Name: hcp-01
ID: 280scqkn8ocjoochasq423tg4donvpaq
External ID: 56549c55-3d86-4ba3-b163-4ca5abf67c59
Control Plane: ROSA Service Hosted
OpenShift Version: 4.14.2
Channel Group: stable
DNS: Not ready
AWS Account: XXXXXXXXX
AWS Billing Account: XXXXXXXXX
API URL:
Console URL:
Region: us-east-2
Availability:
- Control Plane: MultiAZ
- Data Plane: SingleAZ
Nodes:
- Compute (desired): 2
- Compute (current): 0
Network:
- Type: OVNKubernetes
- Service CIDR: 172.30.0.0/16
- Machine CIDR: 10.0.0.0/16
- Pod CIDR: 10.128.0.0/14
- Host Prefix: /23
Workload Monitoring: Enabled
Ec2 Metadata Http Tokens: optional
STS Role ARN: arn:aws:iam::XXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-
Installer-Role
Support Role ARN: arn:aws:iam::XXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-
Support-Role
Instance IAM Roles:
- Worker: arn:aws:iam::XXXXXXXXX:role/ManagedOpenShift-HCP-ROSA-
Worker-Role
Operator IAM Roles:
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1-kube-system-capa-controller-manager
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1-kube-system-control-plane-operator
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1-kube-system-kms-provider
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1 Openshift-cluster-csi-drivers-ebs-cloud-
credentials
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1 Openshift-cloud-network-config-
controller-cloud-cred
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1 Openshift-image-registry-installer-
cloud-credentials
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1 Openshift-ingress-operator-cloud-
credentials
- arn:aws:iam::XXXXXXXXX:role/hcp-01-g8r1-kube-system-kube-controller-manager
Managed Policies: Yes
State: waiting (Waiting for user action)

```

Private: No
Created: Dec 9 2023 07:17:00 UTC
Details Page: https://console.redhat.com/openshift/details/s/2ZI0eJctRNzzSZe58wMQFmA12Sd
OIDC Endpoint URL: https://rh-oidc.s3.us-east-1.amazonaws.com/280rqooan3kqqn4o1us1ip073fsrkfm8 (Managed)
Audit Log Forwarding: disabled

I: Preparing to create operator roles.
I: Creating roles using 'arn:aws:iam::XXXXXXXXXX:user/testuser01'
I: Created role 'hcp-01-g8r1-openshift-cloud-network-config-controller-cloud-cred' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-openshift-cloud-network-config-controller-cloud-cred'
I: Created role 'hcp-01-g8r1-kube-system-capa-controller-manager' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-kube-system-capa-controller-manager'
I: Created role 'hcp-01-g8r1-kube-system-control-plane-operator' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-kube-system-control-plane-operator'
I: Created role 'hcp-01-g8r1-kube-system-kms-provider' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-kube-system-kms-provider'
I: Created role 'hcp-01-g8r1-kube-system-kube-controller-manager' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-kube-system-kube-controller-manager'
I: Created role 'hcp-01-g8r1-openshift-image-registry-installer-cloud-credentials' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-openshift-image-registry-installer-cloud-credentials'
I: Created role 'hcp-01-g8r1-openshift-ingress-operator-cloud-credentials' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-openshift-ingress-operator-cloud-credentials'
I: Created role 'hcp-01-g8r1-openshift-cluster-csi-drivers-ebs-cloud-credentials' with ARN 'arn:aws:iam::XXXXXXXXXX:role/hcp-01-g8r1-openshift-cluster-csi-drivers-ebs-cloud-credentials'

I: Preparing to create OIDC Provider.
I: OIDC provider already exists.
I: To determine when your cluster is Ready, run 'rosa describe cluster -c hcp-01'.
I: To watch your cluster installation logs, run 'rosa logs install -c hcp-01 --watch'.

```

クラスターのデプロイ状態は、次のコマンドで確認できます。STATE が installing から ready に変更されるまで、大体10分ほど待ちます。

```
$ rosa list cluster
ID                  NAME      STATE   TOPOLOGY
280scqkn8ocjoochasq423tg4donvpaq  hcp-01  ready  Hosted CP
```

これによって、コントロールプレーンのデプロイが、Red Hat SREチームのAWSアカウントにデプロイされます。このコントロールプレーンは、ユーザーには見えず、かつ、AWSリソース利用料金も請求されません。

そしてワーカーノードのデプロイが完了されるまで待ちます。 rosa list machinepool コマンドで、 REPLICAS が指定した台数分デプロイされるのを確認できます。

```
$ rosa list machinepool -c $CLUSTER_NAME
ID      AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS      TAINTS      AVAILABILITY
ZONE   SUBNET          VERSION    AUTOREPAIR
workers No           2/2       m5.xlarge
subnet-0087cb7bb3f628793 4.14.2  Yes
us-east-2a
```

作成したROSA HCPクラスターの情報は、OpenShift Cluster Manager (OCM)の画面からも確認できます。下記の画像は、SingleAZ構成(ワーカーノード2台)とMultiAZ構成(ワーカーノード2台と3台)の例です。なお、どちらの構成でも、コントロールプレーンはMultiAZ構成となります。

NOTE

OCMは、OpenShiftクラスターのインストール、修正、操作、およびアップグレードを可能にするRed Hat提供の管理サービスです。OCMを使用すると、単一のダッシュボードからすべてのOpenShiftクラスターを操作できるようになります。[OpenShift Hybrid Console](#)

詳細については、[OCMの公式ドキュメント](#)をご参照ください。

The screenshot shows the OCM interface for the cluster 'hcp-01'. The top navigation bar includes 'Clusters' (selected), 'hcp-01', 'Actions', and a refresh icon. Below the navigation is a breadcrumb trail: Clusters > hcp-01. The main content area has tabs for 'Overview', 'Access control', 'Add-ons', 'Cluster history', 'Networking', 'Machine pools' (selected), 'Support', and 'Settings'. A prominent button 'Add machine pool' is located at the top left of the main content area. The 'Machine pools' table lists one entry: 'workers' (Machine pool), 'm5.xlarge' (Instance type), 'us-east-2a' (Availability zones), '2' (Node count), 'Disabled' (Autoscaling), '4.14.2' (Version), and a three-dot menu icon. A 'Subnets' section below the table shows 'subnet-0087cb7bb3f628793'. On the right side, there is a vertical sidebar with a 'Feedback' button.

hcp-multiaz-02

[Open console](#)

Actions ▾


[Overview](#) [Access control](#) [Add-ons](#) [Cluster history](#) [Networking](#) [Machine pools](#) [Support](#) [Settings](#)
[Add machine pool](#)

Machine pool	Instance type	Availability zones	Node count	Autoscaling	Version	⋮
workers-0	c5.xlarge	us-east-2b	1	Disabled	4.14.2	⋮
workers-1	c5.xlarge	us-east-2a	1	Disabled	4.14.2	⋮

Feedback

hcp-multiaz-01

[Open console](#)

Actions ▾


[Overview](#) [Access control](#) [Add-ons](#) [Cluster history](#) [Networking](#) [Machine pools](#) [Support](#) [Settings](#)
[Add machine pool](#)

Machine pool	Instance type	Availability zones	Node count	Autoscaling	Version	⋮
workers-0	c5.xlarge	us-east-2c	1	Disabled	4.14.2	⋮
workers-1	c5.xlarge	us-east-2b	1	Disabled	4.14.2	⋮
workers-2	c5.xlarge	us-east-2a	1	Disabled	4.14.2	⋮

Feedback

NOTE

ワーカーノードをMultiAZの2台構成にする場合は、MultiAZの3台構成でROSA HCPクラスターを作成した後に、`workers-2`などの名前が付けられたマシンプールを1つ削除して、ワーカーノードを1台削除してください。削除対象のワーカーノード上で後述のOpenShift MonitoringのPrometheus Podが実行されている場合、下記のocコマンド(OpenShift CLI)を管理者アカウント(`cluster-admin`など)で実行する必要があります。このコマンドによって、Prometheus Podの再起動が自動実行されます。Prometheusに関するPodのレプリカ数やモニタリングデータ保存のためのストレージなどは、OpenShiftでは`k8s`という名前のPrometheusリソースによって自動管理されています。

```
$ oc get prometheus -n openshift-monitoring
NAME      VERSION      DESIRED      READY      RECONCILED      AVAILABLE      AGE

```

k8s	2.53.1	2	2	True	True	17m
-----	--------	---	---	------	------	-----

```
$ oc delete pvc -n openshift-monitoring \
prometheus-data-prometheus-k8s-0 \
prometheus-data-prometheus-k8s-1
```

```
$ oc delete pod -n openshift-monitoring \
prometheus-k8s-0 \
prometheus-k8s-1
```

ROSA HCPクラスターへのアクセス

ROSA HCPクラスターのコントロールプレーンのデプロイ完了後に、ROSA HCPクラスターにログインするための管理者権限を持つアカウントを作成できます。これには、`rosa create admin` コマンドを実行します。

```
$ rosa create admin --cluster $CLUSTER_NAME
```

```
I: Admin account has been added to cluster 'hcp-01'.
I: Please securely store this generated password. If you lose this password you can
delete and recreate the cluster admin user.
I: To login, run the following command:
```

```
oc login https://api.hcp-01.240p.p3.openshiftapps.com:443 --username cluster-
admin --password XXXXX-XXXXX-XXXXX-XXXXX
```

```
I: It may take several minutes for this access to become active.
```

ワーカーノードのデプロイが完了していると、ROSA HCPクラスターのコンソールにログインできるようになります。このコンソールのURLは、次のコマンドで確認できます。ユーザー名とパスワードは、`rosa create admin` コマンド実行時に表示されたもの(ユーザー名は `cluster-admin`)を使います。

```
$ rosa describe cluster -c hcp-01 |grep Console
Console URL: https://console-openshift-console.apps.rosa.hcp-
01.240p.p3.openshiftapps.com
```

また、ROSA HCPクラスターでは、ローカルユーザーを利用するため、外部の認証プロバイダとの連携設定が可能です。ROSA HCPクラスターでサポートされている認証プロバイダは下記となります。

- GitHub または GitHub Enterprise
- GitLab

- Google
- LDAP
- OpenID Connect
- htpasswd

htpasswdの場合、`cluster-admin` ユーザー以外での、汎用的な認証プロバイダとしての利用をサポートしていませんので、ご注意ください。次のコマンドは、一時的な検証用途としてhtpasswdを利用するための設定例です。`testuser1 ~ testuser100` のパスワード情報を記載したファイル `users.htpasswd` を、ROSA HCPクラスターの認証プロバイダとして設定しています。

```
$ htpasswd -cbB users.htpasswd testuser1 <適当なランダム文字列のパスワード>
$ for i in {2..100}; do htpasswd -bB users.htpasswd testuser$i <適当なランダム文字列の
パスワード>; done

$ rosa create idp --type=htpasswd --name=testuser-htpasswd01 --
cluster=$CLUSTER_NAME --from-file=users.htpasswd
I: Configuring IDP for cluster 'hcp-01'
I: Identity Provider 'testuser-htpasswd01' has been created.
    It may take several minutes for this access to become active.
    To add cluster administrators, see 'rosa grant user --help'.

I: To log in to the console, open https://console-openshift-console.apps.rosa.hcp-
01.240p.p3.openshiftapps.com and click on 'testuser-htpasswd01'.
```

`rosa create idp` コマンドの実行によって、`htpasswd` コマンドで作成した `users.htpasswd` ファイルにある `testuserX` ユーザーを使ってログインできるようになります。

作成したローカルユーザーに対して、ROSAクラスターの管理者権限を付与(grant)または削除(revoke)する場合、`rosa grant user` コマンドと `rosa revoke user` コマンドを利用します。

```
$ rosa grant user cluster-admin --user <ユーザー名> --cluster <ROSAクラスター名>
$ rosa revoke user cluster-admin --user <ユーザー名> --cluster <ROSAクラスター名>
```

ROSA HCPクラスターの削除

不要になったROSA HCPクラスターは `rosa delete cluster` コマンドで削除できます。`rosa list cluster` コマンドで確認したROSA HCPクラスター名を指定します。また、ROSA HCPクラスターごとに作成されたオペレーターロール(operator roles)も、ROSA HCPクラスターの削除が完了した後に `rosa delete operator-roles` コマンドで削除しておきます。オペレーターロール削除時に指定する接頭辞(prefix)は、`rosa delete cluster` コマンド実行時に表示されます。

OpenID Connectの設定(oidc-config-id)については、別のROSA HCPクラスターを作成する時に使い回せるので、ROSA HCPクラスターの作成/削除を複数回試したい場合、OpenID Connectの設定を消去する必要はありません。

NOTE

本演習をワークショップ形式で実施している場合、受講者間でROSA HCPクラスターを共有しているため、ROSA HCPクラスターを削除しないで下さい。

```
$ rosa list cluster
  ID           NAME     STATE   TOPOLOGY
280scqkn8ocjoochasq423tg4donvpaq  hcp-01  ready  Hosted CP

$ rosa delete cluster -c hcp-01 --yes
...<snip>...
I: Once the cluster is uninstalled use the following commands to remove the above
aws resources.

      rosa delete operator-roles --prefix hcp-01-g8r1
      rosa delete oidc-provider --oidc-config-id 280rqooan3kqqn4o1us1ip073fsrkfm8
I: To watch your cluster uninstallation logs, run 'rosa logs uninstall -c rosa-hcp-
01 --watch'

$ rosa delete operator-roles --prefix hcp-01-g8r1 --mode auto --yes
```

Contents

- デモの概要
- 前準備
- ROSA HCPクラスターに必要なAWS VPCやIAMロールの作成
- ROSA HCPクラスターの作成
- ROSA HCPクラスターへのアクセス
- ROSA HCPクラスターの削除



ROSAにアプリケーションをデプロイ

Contents

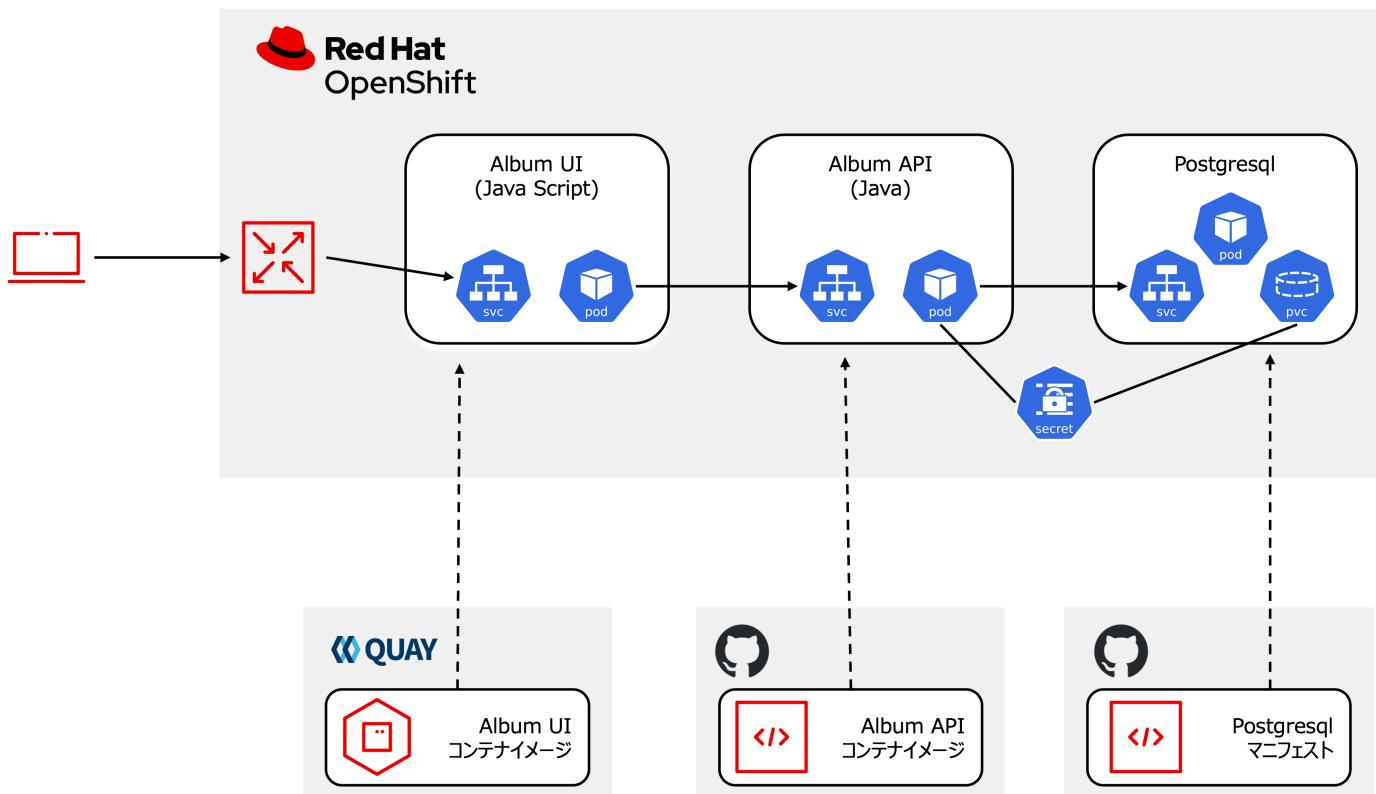
- 1. ゴール
 - 1.1. アプリケーション
- 2. Postgresql のデプロイ
- 3. Album API アプリケーションのデプロイ
- 4. REST API呼び出しの確認
- 5. Album UI アプリケーションのデプロイ
- 6. Album UI と Album API の連携
 - 6.1. 環境変数で設定
 - 6.2. Configmap で設定

1. ゴール

フロントアプリ(album-ui)とバックエンドアプリ(album-api)の2つのアプリケーションをデプロイし、フロントアプリからバックエンドアプリに連携できるように設定していきます。また、バックエンドアプリ(album-api)は、PostgreSQLをデータベースとして利用します。

このハンズオンでは、コンテナイメージおよびGitリポジトリに保存されたソースコードを元にOpenShiftにアプリケーションをデプロイしていきます。OpenShiftのs2iビルドの機能を使ってYAMLファイルを作成することなく、OpenShiftにアプリケーションをデプロイできることを体験します。

1.1. アプリケーション



2. Postgresql のデプロイ

永続ボリュームを利用するPostgresqlをデプロイします。また、データベースに接続するためのユーザ、パスワードを含むSecretも作成します。

```
oc new-project user1-app
oc apply -f https://raw.githubusercontent.com/akubicharm/containerapps-albumapi-
java/main/openshift/postgresql/postgresql.yaml
```

作成されたリソースを確認します。postgresql- で始まる名前のPodのステータス **Running** になればOKです。`

```
oc get secret
oc get pvc
oc get deployment
oc get pods

oc get pods -w
```

3. Album API アプリケーションのデプロイ

Album API アプリケーションのGitリポジトリを指定してアプリケーションをデプロイします。

Album API アプリケーションは、実行時にバックエンドのPostgresqlに接続するため情報をSecret (postgresql) から取得して利用します。

1. 左上のパースペクティブで「開発者」モードを選択
2. 左のメニューで「+追加」をクリック
3. 右のメニューで「Gitリポジトリ - Gitからのインポート」をクリック



4. GitリポジトリURLの情報を入力

<https://gitea-gitea.apps.cluster-pqmg4.pqmg4.sandbox1915.opentlc.com/user1/albumapi>

5. Builder Imageとして OpenJDK17 が選択されていることを確認

The screenshot shows the 'Import from Git' configuration dialog. It has fields for 'Git' and 'Git リポジトリ URL' containing the URL from step 4. A green checkmark indicates the 'Builder Image' field is selected. Below it, a message says 'Builder Image が、検出されました。' (Builder Image detected) and 'ビルダーイメージが推奨されます。' (Recommended builder image). At the bottom, it shows 'Red Hat OpenJDK 17 (UBI 8)' as the selected builder image.

6. 名前を albumapi になっていることを確認（後でのこの名前を使います）

7. リソースタイプが「Deployment」になっていることを確認

8. 「> 詳細なデプロイメントオプションの表示」をクリックして環境変数を入力。バックエンドで PostgreSQを利用することと、SecretからDB接続情報を取得して環境変数として利用するように

設定

- spring_datasourceの設定
 - 名前: `spring_profile_active`
 - 値: `postgresql`
- JDBC URLの設定
 - 「+ConfigMapまたはシークレットから追加」をクリック
 - 名前: `POSTGRESQL_URL`
 - リソースの選択: シークレット `postgresql`
 - キーの選択: `database-url`
- DB接続ユーザの設定
 - 「+ConfigMapまたはシークレットから追加」をクリック
 - 名前: `POSTGRESQL_USER`
 - リソースの選択: シークレット `postgresql`
 - キーの選択: `database-user`
- DB接続パスワードの設定
 - 「+ConfigMapまたはシークレットから追加」をクリック
 - 名前: `POSTGRESQL_PASSWORD`
 - リソースの選択: シークレット `postgresql`
 - キーの選択: `database-password`

9. 詳細オプションのターゲットポートが `8080` になっていることを確認

10. route の作成にチェックが入っていることを確認

環境変数 (Runtimeのみ)

名前	値
POSTGRES_URL	postgresql database-url
POSTGRES_USER	postgresql database-user
POSTGRES_PASSWORD	postgresql database-password
spring_profiles_active	postgresql

[+ 値の追加](#) [+ ConfigMap またはシークレットから追加](#)

詳細オプション

ターゲットポート

8080

x ▾

トラフィックのターゲットポート。

route の作成

パブリック URL でコンポーネントを公開します

[▶ 詳細なルーティングオプションの表示](#)

名前をクリックして、[ヘルスチェック](#)、[スケーリング](#)、[リソース制限](#)、[ラベル](#) の詳細オプションにアクセスします。

作成

キャンセル

NOTE

フロントエンドアプリケーションからしか呼ばれない想定なので外部からアクセスするためのrouteの作成は本来不要ですが、ここでは動作確認のため外部からアクセスするためのrouteを作成します。

4. REST API呼び出しの確認

デプロイが完了したら、ブラウザでalbum-apiのURLを開きます。

1. トポロジービューでアプリケーションがデプロイされていることを確認
2. URLを開くボタンをクリック または <https://albumapi-user1-app.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com/>
3. 「Please visit /albums to see a list of albums.」と表示されているのでコンテキストパスに /albums を追加してURLを開く
4. 以下のようなJSON形式のデータが表示されることを確認

```
[  
{
```

JSON

```
"id": 1,
"title": "OpenShift Virtualizationサーバ仮想化実践ガイド",
"artist": "石川 純平/大村 真樹",
"price": 3080,
"book_url": "https://book.impress.co.jp/books/1124101080",
"image_url": "https://img.ips.co.jp/ij/24/1124101080/1124101080-520x.jpg"
},
{
"id": 2,
"title": "インフラの構成管理と自動化のための実践Ansible",
"artist": "八木澤健人/呉理沙/小野天平/長嶺精彦/山中裕史",
"price": 3960,
"book_url": "https://www.shuwasytem.co.jp/book/9784798068725.html",
"image_url": "https://www.shuwasytem.co.jp/images/book/647676.jpg"
},
{
"id": 3,
"title": "OpenShift徹底入門",
"artist": "レッドハット株式会社",
"price": 4180,
"book_url": "https://www.shoeisha.co.jp/book/detail/9784798172552",
"image_url": "https://www.seshop.com/static/images/product/24696/L.png"
},
{
"id": 4,
"title": "Podmanイン・アクション",
"artist": "Daniel Walsh",
"price": 4180,
"book_url": "https://www.shuwasytem.co.jp/book/9784798070209.html",
"image_url": "https://www.shuwasytem.co.jp//images/book/633833.jpg"
},
{
"id": 5,
"title": "バージョン8&9両対応！ Red Hat Enterprise Linux完全ガイド",
"artist": " 小島啓史/平初/田中司恩/橋本賢弥/八木澤健人/米山和重",
"price": 4950,
"book_url": "https://info.nikkeibp.co.jp/media/LIN/atcl/books/082200035/",
"image_url": "https://cdn-
info.nikkeibp.co.jp/media/LIN/atcl/books/082200035/top.jpg?
__scale=w:250,h:322&_sh=0990b30450"
},
{
"id": 6,
"title": "Quarkus in Action (Free eBook Edition)",
"artist": "Martin Stefanko/Jan Martiska",
"price": 0,
"book_url": "https://developers.redhat.com/e-books/quarkus-action?
extIdCarryOver=true&sc_cid=701f2000001Css5AAC",
```

```

    "image_url":  

    "https://developers.redhat.com/sites/default/files/styles/cheat_sheet_feature/public/E-book%20cover%20graphic_Quarkus%20in%20Action.jpg.webp?itok=xZlT_iv4"  

  }
]

```

5. Album UI アプリケーションのデプロイ

フロントエンドアプリ(album-ui)をデプロイします。

1. 左上のパースペクティブで「開発者」モードを選択
2. 左のメニューで「+追加」をクリック
3. 右のメニューで「コンテナイメージ」をクリック



4. イメージセクションの「外部レジストリーからのイメージ名」を選択し、イメージのURL「quay.io/keomizo_redhat/albumui-nodejs」と入力

URLのチェックが終わると「検証済み」になります

イメージのデプロイ

イメージ

イメージストリームまたはイメージレジストリーから既存のイメージをデプロイします。

- 外部レジストリーからのイメージ名

quay.io/keomizo_redhat/albumui-nodejs

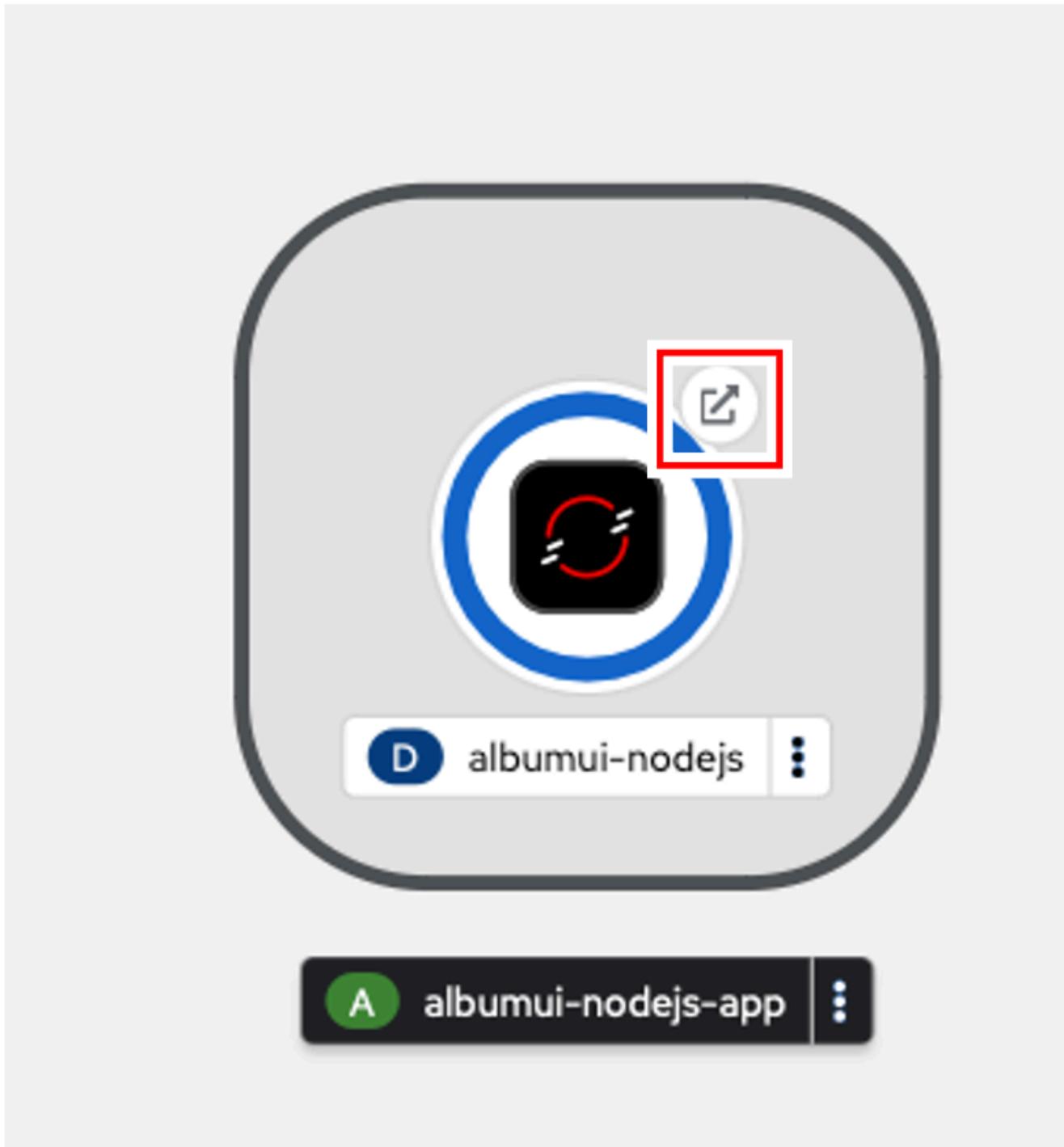


検証済み

プライベートリポジトリからイメージをデプロイするには、イメージレジストリーの認証情報を使用して [イメージのプルシークレットを作成する](#) 必要があります。

- 非セキュアなレジストリーからのイメージの許可

- リソースタイプが「Deployment」になっていることを確認
- 詳細オプションセクションのターゲットポートが 8080 になっていることを確認
- route の作成にチェックが入っていることを確認
- 画面下部の「作成」ボタンをクリック
- トポロジービューでアプリケーションがデプロイされていることを確認（濃い青線ならばOK）
- URLを開くボタンをクリック または <https://albumui-nodejs-user1-app.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com/> にアクセス



11. 「Unable to communicate with server」と画面に表示されていることを確認



NOTE

この段階では、Album UIアプリから呼び出すREST APIのURLが設定されていないので、サーバに接続できないというエラーになります。

6. Album UI と Album API の連携

Album UI アプリケーションはバックエンドサービスのURLを `API_BASE_URL` という環境変数で指定するようになっていますので、環境変数を設定して連携できるようにしていきます。

環境変数の指定方法は複数あります。

1. Podに環境変数を設定する（Deploymentでenvを指定する）
2. Configmapを作成してPodからマウントする

このワークショップではどちら好きな方法を選んで設定してください。

6.1. 環境変数で設定

Deploymentを編集して環境変数として `API_BASE_URL` を指定していきます。

1. トポロジービューで「(D)album-ui」をクリック
2. 右Paneの「アクション」プルダウンメニューで「Deploymentの編集」をクリック
3. 「環境変数」にバックエンドのURLを指定

名前

API_BASE_URL

値

http://albumapi:8080

4. 「保存」ボタンをクリック

Deploymentが更新されるとPodが再起動されます。再起動後にPodに環境変数が設定されていることを確認してください。

5. ブラウザをリロードしてalbum-apiと接続できていることを確認

6.2. Configmapで設定

1. 左のメニューで「Configmap」をクリックし、右上の「Configmapを作成」ボタンをクリック
2. パラメータを入力して画面下部の「作成」ボタンをクリック

名前	albumui-config
キー	API_BASE_URL
値	http://albumapi:8080

3. トポロジービューで 「(D)album-ui」 をクリック
4. 右Paneの「アクション」 プルダウンメニューで「Deploymentの編集」 をクリック
5. 「+ Configmapまたはシークレットから追加」 をクリック

名前	API_BASE_URL （これが環境変数名になる）
リソースの選択	ConfigMap albumui-config
キーの選択	API_BASE_URL

6. 「保存」 ボタンをクリック

Deploymentが更新されるとPodが再起動されます。再起動後にPodに環境変数が設定されていることを確認してください。

7. ブラウザをリロードしてalbum-apiと接続できていることを確認

Greatest Hits

[Docs](#)[What's new](#)[OpenShift Commons](#)[github](#)

OpenShift Virtualization サーバ仮想化実践ガイド

石川 純平/大村 真樹 3080



インフラの構成管理と自動化のための実践Ansible

八木澤健人/吳理沙/小野天平/長嶺精彦/山中裕史 3960

NOTE

YAMLファイルを使ってConfigmapを作成する場合は、以下を参考にしてください。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: albumui-config
  namespace: user1-app
data:
  API_BASE_URL: http://albumapi:8080
immutable: false
```

YAML



Contents

1. ゴール
 - 1.1. アプリケーション

2. Postgresql のデプロイ
3. Album API アプリケーションのデプロイ
4. REST API呼び出しの確認
5. Album UI アプリケーションのデプロイ
6. Album UI と Album API の連携
 - 6.1. 環境変数で設定
 - 6.2. Configmapで設定



コンテナイメージをECRにコピー

Contents

1. ゴール
2. Skopeo
3. 準備 skopeo のインストール
4. skopeo cli を使ったコンテナイメージのコピー
 - 4.1. OpenShift内部のイメージレジストリの公開
 - 4.2. ECR リポジトリのToken取得
 - 4.3. OpenShiftのToken取得
 - 4.4. Skopeo を使ってコンテナイメージのコピー
5. (オプション) ECRのイメージを利用したデプロイ
 - 5.1. OpenShiftにログイン
 - 5.2. ECR用のImage Pull Secretの作成
 - 5.3. アプリケーションをデプロイ (GUIの利用)
 - 5.4. アプリケーションのデプロイ(Manifestの利用)

1. ゴール

OpenShiftの **s2i(Source to Image)** ビルドの機能を使って作成された、OpenShiftの内部のイメージレジストリに登録されたコンテナイメージを外部のリポジトリにコピーします。

2. Skopeo

Skopeoはコンテナイメージとイメージリポジトリの操作、検査、署名、転送を行うためのツールです。OSSのツールですがRed Hat Enterprise Linuxのサブスクリプションに含まれます。バイナリだけでなくコンテナイメージとしても提供されているので、様々なプラットフォームでようが可能です。

<https://www.redhat.com/ja/topics/containers/what-is-skopeo>

3. 準備 skopeo のインストール

RHEL 以外のプラットフォームでの利用はGithubで公開されているインストール手順に従ってください。<https://github.com/containers/skopeo/blob/main/install.md>

4. skopeo cli を使ったコンテナイメージのコピー

skopeo を使って、OpenShiftの内部のイメージレジストリから、AWS ECRにコンテナイメージをコピーします。

4.1. OpenShift内部のイメージレジストリの公開

OpenShiftの内部のイメージレジストリは、デフォルトでは外部からアクセスできません。

NOTE

ワークショップ環境では、すでにイメージレジストリが公開されているため、以下の手順は不要です。イメージレジストリに外部からアクセスできるようにするために、**cluster-admin** 権限を持つユーザで以下のコマンドを実行します。

```
oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec": {"defaultRoute":true}}' --type=merge
```

BASH



公開されたリポジトリのイメージ一覧を取得します。このコマンドの実行には **cluster-admin** 権限が必要です。

```
export TOKEN=$(oc whoami -t)
export ROUTE=$(oc get route -n openshift-image-registry -o
jsonpath='{.items[0].spec.host}')
curl -k -H "Authorization: Bearer $TOKEN" "https://$ROUTE/v2/_catalog" | jq
```

BASH



リポジトリのURLは以下のようになります。

```
default-route-openshift-image-
registry.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com
```

4.2. ECR リポジトリのToken取得

講師が共有メモに記載したトークンを利用して下さい。

NOTE

AWS CLI を利用してトークンを取得する場合は、以下のコマンドを実行します。

```
aws ecr get-login-password --region us-east-2
```

BASH



4.3. OpenShiftのToken取得

OpenShift ターミナルではなく、skopeoが実行できるターミナルでOpenShiftにCLIでログインして実行します。

```
oc login https://api.keomizorosa.guj9.p3.openshiftapps.com:443 -u user1 -p  
openshift
```

BASH



NOTE

OpenShiftコンソールからトークンを取得する場合は、以下の手順で取得します。

1. OpenShiftコンソールにログイン
2. 右上のユーザ名をクリック
3. 「ログインコマンドをコピー」をクリック
4. user1 と openshift を入力してログイン
5. Display Token をクリック

4.4. Skopeo を使ってコンテナイメージのコピー

1. ECRにログイン

```
export ECR_TOKEN=[ECRのトークン]
```

BASH



```
skopeo login --tls-verify=false -u AWS -p $ECR_TOKEN 714932348383.dkr.ecr.us-east-  
2.amazonaws.com
```

1. OpenShiftのイメージレジストリにログイン

```
oc login https://api.keomizorosa.guj9.p3.openshiftapps.com:443 -u user1 -p  
{password}  
export TOKEN=$(oc whoami -t)  
skopeo login --tls-verify=false -u user1 -p $TOKEN default-route-openshift-  
image-registry.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com
```

BASH



2. OpenShiftのイメージレジストリからECRにコンテナイメージをコピー

```
skopeo copy docker://default-route-openshift-image-  
registry.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com/user1-app/albumui-  
nodejs docker://714932348383.dkr.ecr.us-east-  
2.amazonaws.com/user1/albumui:latest
```

BASH



コピー元、コピー先のリポジトリにログインしていない場合、credential を指定してコピーすることができます。

```
oc login https://api.keomizorosa.guj9.p3.openshiftapps.com:443 -u user1 -p  
{password}
```

BASH



```
export TOKEN=$(oc whoami -t)
export ECR_TOKEN=[ECRのトークン]
skopeo copy --src-creds user1:$TOKEN docker://default-route-openshift-image-
registry.apps.rosa.keomizorosa.guj9.p3.openshiftapps.com/user1-app/albumui --
dest-creds AWS:$ECR_TOKEN docker://714932348383.dkr.ecr.us-east-
2.amazonaws.com/album/albumui:latest
```

5.(オプション) ECRのイメージを利用したデプロイ

Private な ECR に格納されたコンテナイメージを利用してオンプレミスのOpenShiftにアプリケーションをデプロイします。

5.1. OpenShiftにログイン

CLIを利用してOpenShiftにログインします。

5.2. ECR用のImage Pull Secretの作成

1. OpenShiftのプロジェクトを作成
2. ECRのイメージをPullするためのSecretを作成

```
oc create secret docker-registry ecr-secret --
docker-server=714932348383.dkr.ecr.us-east-2.amazonaws.com --docker-user
name=AWS --docker-password=$(aws ecr get-login-password)
```

BASH

OpenShiftのGUI または Manifest を利用してアプリケーションをデプロイします。

5.3. アプリケーションをデプロイ (GUIの利用)

1. 左上のペースペクティブで「開発者」モードを選択
2. 左のメニューで「+追加」をクリック
3. 右のメニューで「コンテナイメージ」をクリック
4. 外部のイメージレジストリからのイメージ名でECRのURIを入力

714932348383.dkr.ecr.us-east-2.amazonaws.com/album/albumui:latest

5. 画面下部の「作成」ボタンをクリック

5.4. アプリケーションのデプロイ(Manifestの利用)

1. Deploymentのマニフェストファイルを作成して適用

```
oc create deployment albumui --image=714932348383.dkr.ecr.us-east-2.amazonaws.com/album/albumui:latest --dry-run=client -o yaml > deployment-albumui.yaml
```

BASH

```
oc apply -f deployment-albumui.yaml
```

2. Serviceのマニフェストを作成して適用

```
oc create service clusterip albumui --tcp=8080 --dry-run=client -o yaml > service-albumui.yaml
```

BASH

```
oc apply -f service-albumui.yaml
```

3. Routeのマニフェストを作成して適用

```
oc create route edge albumui --service=albumui --port 8080 --dry-run=client -o yaml > route-albumui.yaml  
oc apply -f route-albumui.yaml
```

BASH

Contents

1. ゴール
2. Skopeo
3. 準備 skopeo のインストール
4. skopeo cli を使ったコンテナイメージのコピー
 - 4.1. OpenShift内部のイメージレジストリの公開
 - 4.2. ECR リポジトリのToken取得
 - 4.3. OpenShiftのToken取得
 - 4.4. Skopeo を使ってコンテナイメージのコピー
5. (オプション) ECRのイメージを利用したデプロイ
 - 5.1. OpenShiftにログイン
 - 5.2. ECR用のImage Pull Secretの作成
 - 5.3. アプリケーションをデプロイ (GUIの利用)
 - 5.4. アプリケーションのデプロイ(Manifestの利用)



Amazon EBSの利用

Contents

演習の概要

永続ボリューム要求(Persistent Volume Claim, PVC)の作成

PVCを利用するPodの作成

演習の概要

このモジュールでは、Amazon EBSを利用した、コンテナアプリのデータ保存を実行します。

永続ボリューム要求(Persistent Volume Claim, PVC)の作成

ROSAには、Amazon Elastic Block Store (EBS) ボリュームを使用するストレージクラスが事前に設定されています。このため、[Amazon EBSのgp2, gp3ボリュームタイプ](#)がすぐ使えるようになっています。

The screenshot shows the ROSA web interface for managing storage classes. The left sidebar is titled "Red Hat OpenShift Service on AWS" and includes navigation links for Home, Operator, Workloads, Network, Storage (with PersistentVolumeClaims selected), and StorageClasses (which is currently active). The main content area is titled "StorageClasses" and displays two entries:

名前	プロビジョナー	回収ポリシー
SC gp2-csi	ebs.csi.aws.com	Delete
SC gp3-csi - デフォルト	ebs.csi.aws.com	Delete

このうち、デフォルトのストレージクラスがgp3として設定されており、外部ストレージを永続ボリュームとして利用する際のデフォルトとして利用されます。

Red Hat
OpenShift Service on AWS

管理者向け表示

ホーム

Operator

Workloads

ネットワーク

ストレージ

PersistentVolumeClaims

StorageClasses

VolumeSnapshots

VolumeSnapshotClasses

Builds

ユーザー管理

管理

StorageClasses > StorageClass の詳細

sc gp3-csi

アクション

詳細 YAML

StorageClass の詳細

名前: gp3-csi 回收ポリシー: Delete

ラベル: ラベルなし 編集

デフォルトクラス: True

アノテーション: アノテーション 1個

プロビジョナー: ebs.csi.aws.com

ボリュームバインディングモード: WaitForFirstConsumer

作成日時: 2023年12月9日 16:26

オーナー: オーナーなし

ここでgp3ストレージクラスを利用するため、新しく永続ボリューム要求(Persistent Volume Claim, PVC)を作成します。「PersistentVolumeClaimの作成」をクリックして、PVCの名前は、任意の名前(ここではtest-pvc-20)を入力し、要求するサイズは1GiBと指定します。



- 管理者向け表示
- [ホーム](#)
- [Operator](#)
- [Workloads](#)
- [ネットワーク](#)
- [ストレージ](#)
 - [PersistentVolumeClaims](#)
 - [StorageClasses](#)
 - [VolumeSnapshots](#)
 - [VolumeSnapshotClasses](#)
- [Builds](#)
- [ユーザー管理](#)
- [管理](#)

プロジェクト: test-project20 ▾

PersistentVolumeClaim の作成

[YAML の編集](#)**StorageClass**

SC gp3-csi

新規要求の StorageClass

PersistentVolumeClaim 名 *

test-pvc-20

プロジェクト内のストレージ要求の一意の名前

アクセスモード * 単一ユーザー (RWO) 共有アクセス (RWX) 読み取り専用 (ROX)

アクセスモードは StorageClass で設定され、変更できません

サイズ *

-	1	+
---	---	---

 GiB ▾

必要なストレージ容量

 ラベルセレクターを使用したストレージの要求

すべてのラベルセレクターと一致する PersistentVolume リソースはバインディングの対象として考慮されます。

ボリュームモード * ファイルシステム ブロック[作成](#)[キャンセル](#)**NOTE**

PVCはプロジェクトという名前空間の中にあるリソースです。そのため、各プロジェクトにおいて、同じ名前のPVCが存在できます。例えば、プロジェクト1の中にPVC1、プロジェクト2の中にPVC1を作ることができます。ただし、1つのプロジェクトの中のリソース名の重複は許可されていないため、この例の場合だと、プロジェクト1の中にPVC1を2つ作ることはできません。

このgp3ストレージクラスは、ボリュームバインディングモードが「WaitForFirstConsumer」と指定されており、最初にPodから永続ボリューム要求が利用されるまで、永続ボリュームの割り当てが行われない(ステータスがPendingのまま)ようになっています。なお、ボリュームバインディングモードが「Immediate」となっている場合、PVC作成後すぐに永続ボリュームの割り当てが行われます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a 'PersistentVolumeClaims' section selected. The main content area shows a PersistentVolumeClaim named 'test-pvc-20' in the 'test-project20' namespace, with a status of 'Pending'. The 'Details' tab is selected. The claim is associated with the 'gp3-csi' StorageClass and has no annotations or labels. It was created on December 10, 2023, at 13:04. The owner is listed as 'ownerなし'.

プロジェクト: test-project20 ▾

PersistentVolumeClaims > PersistentVolumeClaim の詳細

PVC **test-pvc-20** ✗ Pending

詳細 YAML イベント VolumeSnapshots

PersistentVolumeClaim の詳細

名前: test-pvc-20

ステータス: ✗ Pending

Namespace: NS test-project20

要求された容量: 1 GiB

ラベル: 編集

ラベルなし

ポリュームモード: Filesystem

StorageClasses: SC gp3-csi

アノテーション: アノテーション 0 個

ラベルセレクター: セレクターなし

作成日時: 2023年12月10日 13:04

オーナー: オーナーなし

PVCを利用するPodの作成

Podを作成します。「Podの作成」から、次のYAMLファイルを入力してPodを作成します。

NOTE

PodはKubernetes/OpenShift上でのコンテナアプリの実行単位です。下記のYAMLファイルにあるとおり、コンテナ(この例ではCentOSコンテナの最新版を利用)やコンテナが利用する永続ボリュームの設定などをまとめたものになります。Podにはコンテナを複数まとめることもできますが、基本的には1つのPodには1つのコンテナを含むことを推奨しています。

画面右上（ユーザ名の並び）にある「+」をクリックして、下記のYAML定義を貼り付けます。

```
apiVersion: v1
kind: Pod
metadata:
```

YAML



```

name: test-ebs
namespace: user1-app
spec:
  volumes:
    - name: ebs-storage-vol
      persistentVolumeClaim:
        claimName: test-pvc-20
  containers:
    - name: test-ebs
      image: centos:latest
      command: [ "/bin/bash", "-c", "--" ]
      args: [ "while true; do touch /mnt/ebs-data/verify-ebs && echo 'hello ebs' && sleep 30; done;" ]
      volumeMounts:
        - mountPath: "/mnt/ebs-data"
          name: ebs-storage-vol
  securityContext:
    allowPrivilegeEscalation: false
    seccompProfile:
      type: RuntimeDefault

```

test-ebsという名前でPodが作成されて、Podにより「test-pvc-20」PVCが利用されて、永続ボリュームとして外部ストレージの利用が開始されます。

名前	ステータス	PersistentVolumes	容量
PVC test-pvc-20	Bound	PV pvc-5269c4d8-b835-4295-a184-f13c71ddb620	1 GiB

このPodのログやターミナルから、永続ボリュームのマウント状況や動作状況を確認できます。

The screenshot shows the Red Hat OpenShift Service on AWS interface. On the left, there's a sidebar with navigation options like Home, Operator, Workloads (Pods selected), Deployments, DeploymentConfigs, StatefulSets, Secrets, and ConfigMaps. The main area displays a pod named 'test-ebs' which is running. The 'Logs' tab is selected, showing a stream of logs where each line says 'hello ebs'. There are also tabs for Metrics, YAML, Environment, Events, and Terminal. Below the logs, there are buttons for Refresh, Raw, Download, and Expand.

This screenshot shows the terminal output of the 'test-ebs' Pod. It includes commands like df -h, mount, echo, and ls to verify the EBS volume mount. The output shows the volume is mounted at /mnt/ebs-data and contains a 'testfile' and a 'lost+found' directory. The 'Terminal' tab is selected at the top.

```
sh-4.4$ df -h | grep ebs
/dev/nvme2n1      974M   24K   958M   1% /mnt/ebs-data
sh-4.4$ mount | grep ebs
/dev/nvme2n1 on /mnt/ebs-data type ext4 (rw,relatime,seclabel)
sh-4.4$
sh-4.4$ echo test > /mnt/ebs-data/testfile
sh-4.4$ ls -lh /mnt/ebs-data/
total 20K
drwxrws---. 2 root          1000800000 16K Dec 10 04:08 lost+found
-rw-r--r--. 1 1000800000 1000800000    5 Dec 10 04:16 testfile
-rw-r--r--. 1 1000800000 1000800000    0 Dec 10 04:16 verify-ebs
sh-4.4$
```

ここで上記画像にあるように、Podのターミナルから、echoコマンドなどで永続ボリュームのマウントポイントである /mnt/ebs-data ディレクトリに、適当なファイルを作成します。Podを削除(該当Podを選択して、「アクション」→「Podの削除」を選択)した後に、再度「test-pvc-20」PVCを指定してPodを作成すると、作成したテストファイルが残っていることを確認できます。

確認が終わったら「アクション」→「Podの削除」を選択して、Podを削除してください。

Amazon EBSを利用したPVCは、1台のワーカーノードでマウントして利用できます。共有ファイルシステムのように、複数台のワーカーノードでマウントして利用することはできません。また、1台のワーカーノードに接続できるEBSのボリュームは39個までとなります。これはAWSのEBSの制限に起因します。

複数台のワーカーノードで利用する共有ファイルシステム用の永続ボリュームとして、Amazon EFSを利用することもできます。その場合、Amazon EFSのContainer Storage Interface(CSI.ストレージベンダーが提供するKubernetes用のインターフェース)を利用するため、「AWS EFS CSI Driver Operator」をインストール/設定して利用します。

ROSA HCPクラスターはAWS STSを利用しているため、Amazon EFSを利用するためのIAMロールとポリシーを作成する必要があります。詳細については、公式ドキュメントをご参照ください。

The screenshot shows the Red Hat OpenShift Service on AWS interface. On the left, there's a sidebar with navigation links: プロジェクト, 検索, API Explorer, イベント, Operator (with OperatorHub selected), Workloads, ネットワーク, ストレージ, and Builds. The main area has a header 'OperatorHub' and a search bar with the query 'EFS CSI'. Below the search bar, it says 'すべての項目' and shows a result for 'AWS EFS CSI Driver Operator' provided by Red Hat. The result description states: 'Install and configure AWS EFS CSI driver.'

Contents

演習の概要

永続ボリューム要求(Persistent Volume Claim, PVC)の作成

PVCを利用するPodの作成



[デモ] ロギング設定

Contents

デモの概要

Amazon S3をAWS STSで利用するためのIAMロール作成

Amazon S3のバケット作成

OpenShift Loggingに必要となるOperatorのインストール

ロギングの設定

ログ転送の設定

デモの概要

このモジュールでは、インストラクターがROSAクラスターのロギング設定方法をご紹介します。

Loki Logging設定の動画

NOTE

本演習を自習している時以外、ROSAクラスターのロギング設定を実行しないで下さい。

ROSAのロギングについては、OpenShift LoggingのLokiを利用したロギングか、Amazon CloudWatchをベースとするログ転送ソリューションの利用を推奨しています。ROSAでのOpenShift Loggingによるロギング設定の概要は次のとおりです。

1. Lokiのログ保存に利用するAmazon S3を、AWS STSで利用するためのIAMロールを作成します。
2. Amazon S3の汎用バケットを作成します。
3. ROSAのOpenShiftコンソールから、OpenShift Logging OperatorやLoki Operatorなどをインストールして、ロギングとログ転送用のインスタンスを作成します。このとき、上記手順で作成したIAMロールとAmazon S3のバケットを利用するように設定します。

これらを順番に見ていきましょう。

Amazon S3をAWS STSで利用するためのIAMロール作成

AWSのコンソールまたはAWS CLIで、Amazon S3をAWS STSで利用するためのIAMロールを作成します。IAMロール作成手順は [「カスタム信頼ポリシーを使用してロールを作成する」](#) をご参照下さい。

例として、下記画像のような「AmazonS3FullAccess」ポリシーを割り当てたIAMロールを作成します。

概要			
作成日	ARN		
March 22, 2025, 22:16 (UTC+09:00)	arn:aws:iam: XXXXXXXXXX :role/rosa-hcp-s3-role		
最後のアクティビティ	最大セッション時間		
39 分前	1 時間		
許可	信頼関係		
タグ	最終アクセス日時		
セッションを取り消す			
許可ポリシー (1) 情報			
最大 10 個の管理ポリシーを添付できます。			
<input type="button" value="C"/>	シミュレート	削除	許可を追加 ▾
絞り込みタイプ			
<input type="text" value="検索"/>	<input type="button" value="すべてのタイプ"/>	< 1 >	
□ ポリシー名	▲ タイプ	▼ アタッチされたエンティティ	
<input type="checkbox"/> AmazonS3FullAccess	AWS 管理	2	

概要

作成日
March 22, 2025, 22:16 (UTC+09:00)

最後のアクティビティ
40 分前

ARN
arn:aws:iam:█████████████████████:role/rosa-hcp-s3-role

最大セッション時間
1 時間

許可 | **信頼関係** | タグ | 最終アクセス日時 | セッションを取り消す

信頼されたエンティティ

指定された条件でこのロールを引き受けることができるエンティティ。

```
1 [ {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Principal": {  
7                 "Federated": "arn:aws:iam:█████████████████████:oidc-provider/oidc.op1.openshiftapps.com/289dfj█████████████████████"  
8             },  
9             "Action": "sts:AssumeRoleWithWebIdentity",  
10            "Condition": {  
11                "StringEquals": {  
12                    "oidc.op1.openshiftapps.com/289dfj█████████████████████:sub": "system:serviceaccount:openshift-logging:logging-loki"  
13                }  
14            }  
15        }  
16    ]  
17 }
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Federated": "arn:aws:iam::<AWS_ACCOUNT_ID>:oidc-provider/<AWS_IAM_ID_PROVIDER_ID>"  
            },  
            "Action": "sts:AssumeRoleWithWebIdentity",  
            "Condition": {  
                "StringEquals": {  
                    "OIDC-Provider-Arn": "arn:aws:iam::<AWS_ACCOUNT_ID>:oidc-provider/<AWS_IAM_ID_PROVIDER_ID>"  
                }  
            }  
        }  
    ]  
}
```

```

    "Condition": {
        "StringEquals": {
            "<AWS_IAM_ID_PROVIDER_ID>:sub": "system:serviceaccount:openshift-logging:logging-loki"
        }
    }
}
]
}

```

上記のカスタム信頼ポリシーは、特定の AWS IAM ID プロバイダーに 紐づけられたOpenShiftやROSAクラスター上の openshift-logging プロジェクトにある logging-loki サービスアカウントに対して、IAMポリシーで指定されたAWSリソースへの特定の操作を許可するためのものとなります。

NOTE

後述するLoki OperatorでLokiStackインスタンスを作成する際に、LokiStackインスタンスの名前と同じ名前のサービスアカウントがOpenShiftクラスターで自動作成されるようになっており、Amazon S3の利用権限をこのサービスアカウントに付与する必要があります。

ROSAの場合は `rosa list oidc-provider` コマンドで、ROSAクラスターに紐づいているAWS IAM IDプロバイダーが確認できるようになっています。下記のコマンドの出力結果で表示されている `oidc.op1.openshiftapps.com/289dfjXXXXXX` がAWS IAM IDプロバイダーのIDとなります。

```

$ rosa list oidc-provider
I: Fetching OIDC providers
OIDC PROVIDER ARN
Cluster ID      In Use
arn:aws:iam::<AWS_ACCOUNT_ID>:oidc-provider/oidc.op1.openshiftapps.com/289dfjXXXXXX
2el3cammYYYYYYY  Yes

```

ここで作成したIAMロールのARNをメモしておきます。

Amazon S3のバケット作成

AWSのコンソールまたはAWS CLIで、[Amazon S3の汎用バケットを作成します。](#)

```

aws s3api create-bucket \
--bucket amzn-s3-demo-bucket1$(uuidgen | tr -d - | tr '[[:upper:]]' \
'[:lower:]' ) \
--region us-east-2 \
--create-bucket-configuration LocationConstraint=us-east-2

```

AWSのコンソールから汎用バケットを作成する場合、バケット名を任意の名前で指定する以外は、全てデフォルトのパラメーターのままでバケットを作成します。なお、バケットのリージョンはどこでも

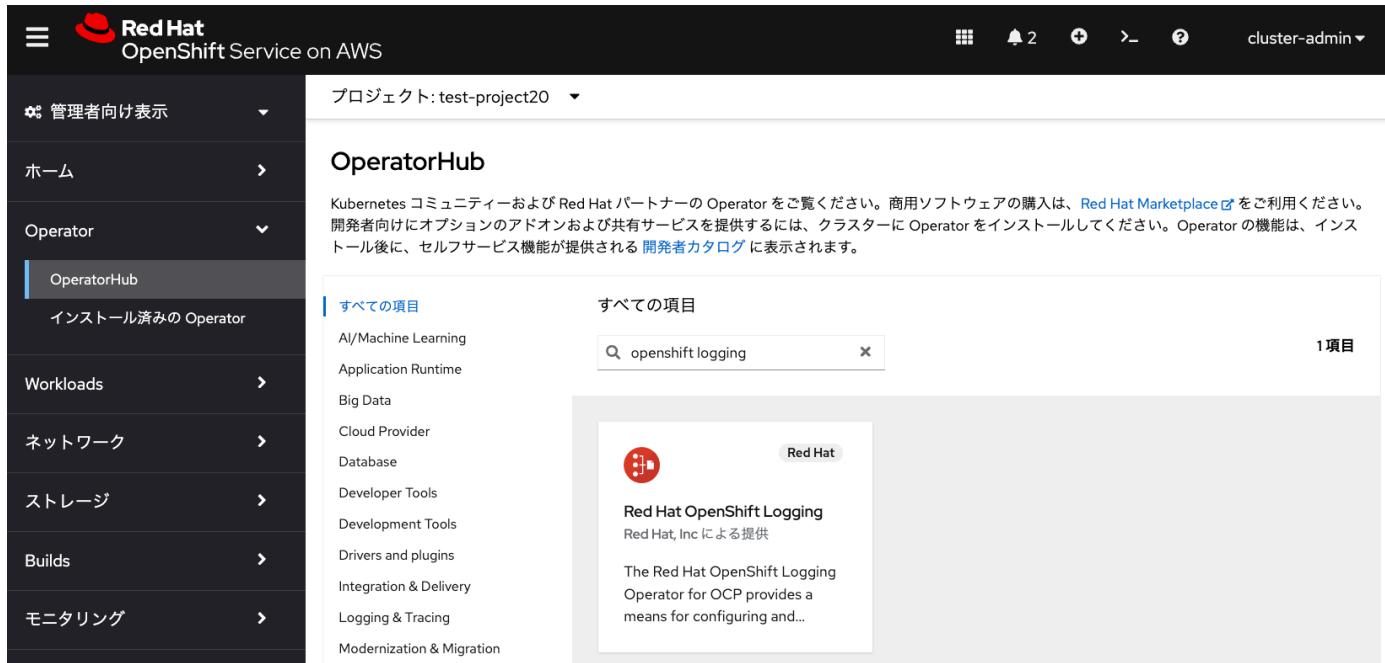
構いません。ROSA HCPクラスターがあるリージョンとは別のリージョンにバケットを作成することもできます。

ここで作成した汎用バケットの名前とリージョンをメモしておきます。

OpenShift Loggingに必要となるOperatorのインストール

OpenShift Loggingを利用するためのOperatorを、管理者アカウント(`cluster-admin` ユーザーなど)で順次インストールしていきます。

OperatorHubから「Red Hat OpenShift Logging Operator」をインストールします。インストールには、全てデフォルトのパラメータを利用します。このOperatorは、vectorでのLokiやCloudWatchへのログ転送設定に利用します。



The screenshot shows the Red Hat OpenShift Service on AWS interface. On the left, there's a sidebar with navigation options like Home, Operator, Workloads, Network, Storage, Builds, and Monitoring. The 'Operator' section is expanded, and 'OperatorHub' is selected. In the main content area, the title is 'OperatorHub'. Below it, there's a search bar with the query 'openshift logging'. A single result is listed: 'Red Hat OpenShift Logging' by Red Hat, Inc. The description states: 'The Red Hat OpenShift Logging Operator for OCP provides a means for configuring and...'.

Operator のインストール

更新チャネルのいずれかにサブスクライブして Operator をインストールし、Operator を最新の状態に保ちます。ストラテジーでは手動または自動の更新のいずれかを決定します。

更新チャネル * ②

stable-6.2

バージョン *

6.2.1

インストールモード *

- クラスターのすべての namespace (デフォルト)
Operator はすべての namespace で利用可能になります。
- クラスターの特定の namespace
Operator は単一の namespace でのみ利用可能になります。

インストール済みの namespace *

- Operator 推奨の namespace: PR **openshift-logging**
- namespace の選択

⚠ namespace はすでに存在します

namespace **openshift-logging** はすでに存在し、使用されます。他のユーザーはこの namespace にすでにアクセスできます。



Red Hat OpenShift Logging

Red Hat 提供のバージョン

提供される API

CLF Cluster Log Forwarder 必須

ClusterLogForwarder is an API to configure forwarding logs.

You configure forwarding by specifying a list of **pipelines**, which forward from a set of named inputs to a set of named outputs.

LFME Log File Metric Exporter

A Log File Metric Exporter instance.LogFileMetricExporter is the Schema for the logFileMetricExporters API

更新の承認 * ②

- 自動
- 手動

インストール

キャンセル

OperatorHubから「Cluster Observability Operator」をインストールします。インストールには、全てデフォルトのパラメータを利用します。このOperatorは、OpenShiftコンソールでのログ集約の表示設定を利用します。

[OperatorHub](#) > Operator のインストール

Operator のインストール

更新チャネルのいずれかにサブスクライブして Operator をインストールし、Operator を最新の状態に保ちます。ストラテジーでは手動または自動の更新のいずれかを決定します。

更新チャネル* ②

バージョン*

インストールモード*

- クラスターのすべての namespace (デフォルト)
Operator はすべての namespace で利用可能になります。
- クラスターの特定の namespace
このモードはこの Operator ではサポートされません

インストール済みの namespace*

- Operator 推奨の namespace: PR openshift-cluster-observability-operator
- namespace の選択

⚠ namespace はすでに存在します

namespace `openshift-cluster-observability-operator` はすでに存在し、使用されます。他のユーザーはこの namespace にすでにアクセスできます。

更新の承認*

- 自動
- 手動

 **Cluster Observability Operator**
Red Hat 提供のバージョン

提供される API

 **PodMonitor**

PodMonitor defines monitoring for a set of pods

 **Probe**

Probe defines monitoring for a set of static targets or ingresses

 **PrometheusRule**

PrometheusRule defines recording and alerting rules for a Prometheus instance

 **ServiceMonitor**

ServiceMonitor defines monitoring for a set of services

[インストール](#) [キャンセル](#)

OperatorHubから「Loki Operator」をインストールします。「Community Loki Operator」ではなく、Red Hat 提供の「Loki Operator」を選択してください。インストール時に「ロール ARN」で、前述の

手順で作成したカスタム信頼ポリシー付きのIAMロールのARNを指定します。「更新の承認」はデフォルトは「手動」となっていますが、Operatorの自動更新を有効化したい場合は「自動」を選択してください。その他のパラメータは、全てデフォルト値を利用します。このOperatorは、Lokiのインスタンス作成に利用します。

The screenshot shows the Red Hat OpenShift Service on AWS interface. The left sidebar has a dark theme with white text. The 'OperatorHub' section is selected. A search bar at the top right contains the query 'Loki Operator'. Below the search bar, the results show one item: 'Loki Operator' by Red Hat. The result card includes a thumbnail of the Red Hat logo, the name 'Loki Operator', the provider 'Red Hat', a brief description, and a 'View Details' button.

ロギングの設定

Lokiを利用したクラスターロギングを設定します。

最初にLokiが利用するためのS3バケットの情報(バケット名とリージョンID)を保存したシークレットトリソースを作成します。OpenShiftコンソール右上にある「+」アイコンをクリックして「YAMLのインポート」を選択し、以下を入力して「作成」をクリックします。以下の例だと、バケット名が `rosa-hcp-test-bucket-000001` で、リージョンIDが `us-east-1` を指定しています。

```
apiVersion: v1
kind: Secret
metadata:
  name: logging-loki-s3
  namespace: openshift-logging
stringData:
  bucketnames: rosa-hcp-test-bucket-000001
  region: us-east-1
```

The screenshot shows the Red Hat OpenShift Service on AWS console. On the left, there's a sidebar with navigation links like Home, Operator, Workloads, Network, Storage, Builds, Monitoring, Compute, User Management, and Management. The main area is titled 'YAML のインポート' (Import YAML) and says 'YAML または JSON ファイルをエディターにドラッグアンドドロップするか、手動でファイルを入力し、---を使用してそれぞれの定義を分離します。' (Drag and drop YAML or JSON files into the editor, or input files manually and separate them using ---). A red box highlights a block of YAML code:

```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: logging-loki-s3
5   namespace: openshift-logging
6 stringData:
7   bucketnames: rosa-hcp-test-bucket-000001
8   region: us-east-1
```

At the bottom of the editor, there are two buttons: '作成' (Create) and 'キャンセル' (Cancel), with '作成' also highlighted by a red box.

openshift-logging プロジェクトに LokiStack カスタムリソースを作成して、Loki実行に必要となるアプリケーション群を作成します。先ほどと同様に、OpenShiftコンソール右上にある「+」アイコンをクリックしてYAMLをインポートします。ここで指定するシークレット名は、先ほど作成した logging-loki-s3 となります。また、1x.demo サイズを指定することで、LokiStackインスタンスのサイズを指定しています。

NOTE

1x.demo はデモ用途なので、本番環境には利用しないでください。

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  managementState: Managed
  size: 1x.demo
  storage:
    schemas:
      - effectiveDate: '2024-10-01'
        version: v13
  secret:
    name: logging-loki-s3
    type: s3
```

```
storageClassName: gp3-csi
tenants:
  mode: openshift-logging
```

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a '管理者向け表示' dropdown set to 'YAML のインポート'. The main area shows a YAML editor with the following content:

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  managementState: Managed
  size: 1x.demo
  storage:
    schemas:
      - effectiveDate: '2024-10-01'
    secret:
      name: logging-loki-s3
      type: s3
  storageClassName: gp3-csi
  tenants:
    mode: openshift-logging
```

The editor includes standard keyboard shortcuts like ⌘ Opt + F1 and a help button. There are also checkboxes for 'ショートカットの表示' and 'ツールチップを表示'. At the bottom are '作成' and 'キャンセル' buttons.

UIPlugin カスタムリソースを作成して、OpenShiftコンソールでのログ集約の表示を有効化します。ここで指定している `logging-loki` は、前述の手順で作成したLokiStackインスタンスの名前です。

```
apiVersion: observability.openshift.io/v1alpha1
kind: UIPlugin
metadata:
  name: logging
spec:
  type: Logging
  logging:
    lokiStack:
      name: logging-loki
```

The screenshot shows the 'YAML のインポート' (Import YAML) page. On the left is a sidebar with navigation links like Home, Operator, Workloads, Network, Storage, Builds, Monitoring, Compute, User Management, and Management. The main area has a title 'YAML のインポート' and a note: 'YAML または JSON ファイルをエディターにドラッグアンドドロップするか、手動でファイルを入力し、--- を使用してそれぞれの定義を分離します。'. Below is a code editor containing a YAML snippet:

```
1 apiVersion: observability.openshift.io/v1alpha1
2 kind: UIPlugin
3 metadata:
4   name: logging
5 spec:
6   type: Logging
7   logging:
8     lokiStack:
9       name: logging-loki
```

At the bottom are '作成' (Create) and 'キャンセル' (Cancel) buttons.

数分待ってOpenShiftコンソールを更新すると、左サイドメニューに「Logs」が表示されます。今の状態だと、vectorでのログ転送が有効化されていないため、まだ何も表示されません。

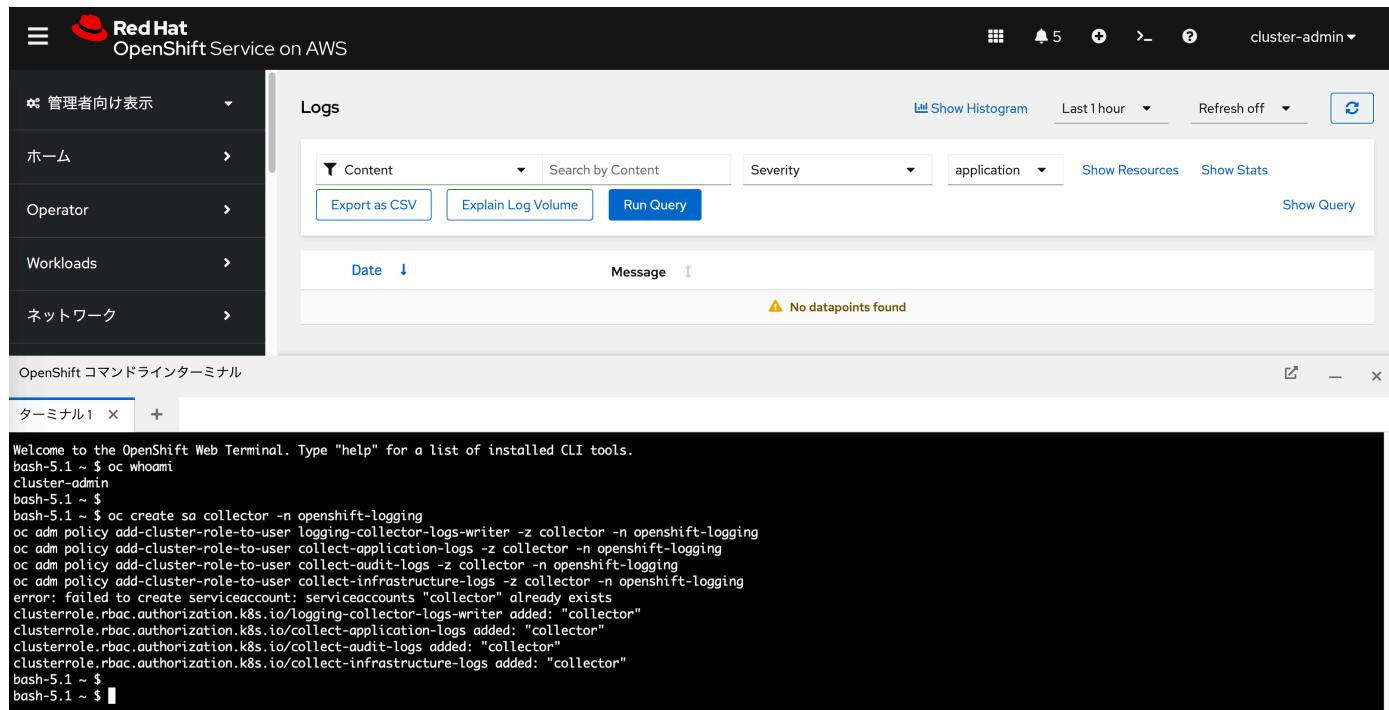
The screenshot shows the 'Logs' page. The left sidebar is identical to the previous one. The main area has a 'Logs' title and a search/filter section with 'Content', 'Search by Content', 'Severity', and dropdowns for 'application' (selected), 'infrastructure', and 'audit'. A message at the bottom says 'No datapoints found'.

ログ転送の設定

次のコマンドでログ収集用のサービスアカウント `collector` を作成し、LokiStackカスタムリソースへのデータ書き込み許可と、アプリケーション/インフラストラクチャー/監査ログ収集を許可する権限を付与します。

```
oc create sa collector -n openshift-logging
oc adm policy add-cluster-role-to-user logging-collector-logs-writer -z collector -n openshift-logging
oc adm policy add-cluster-role-to-user collect-application-logs -z collector -n openshift-logging
oc adm policy add-cluster-role-to-user collect-audit-logs -z collector -n openshift-logging
oc adm policy add-cluster-role-to-user collect-infrastructure-logs -z collector -n openshift-logging
```

この5つのコマンドは、管理者アカウント(`cluster-admin` ユーザーなど)で実行します。前の演習で紹介したOpenShiftのWeb Terminalを利用することができます。



The screenshot shows the OpenShift Web Terminal interface. On the left, there's a sidebar with navigation links: '管理者向け表示', 'ホーム', 'Operator', 'Workloads', and 'ネットワーク'. Below the sidebar, it says 'OpenShift コマンドラインインターミナル'. In the main area, there's a terminal window titled 'ターミナル1'. The terminal output shows the execution of the five commands listed above, which creates the service account 'collector' and adds the required cluster roles and role-based access control (RBAC) rules. A message at the bottom of the terminal window states 'No datapoints found'.

最後にvectorによるLokiStackインスタンスへのログ転送を設定します。以下のYAMLをOpenShiftコンソールからインポートします。`inputRefs` の箇所で、アプリケーション/インフラストラクチャー/監査の3種類のログを転送するように指定しています。また、前述の手順で作成した下記を指定しています。

- サービスアカウント: `collector`
- LokiStackインスタンス: `logging-loki`

```
apiVersion: observability.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: collector
  namespace: openshift-logging
spec:
  serviceAccount:
    name: collector
  outputs:
    - name: default-lokistack
      type: lokiStack
      lokiStack:
        authentication:
          token:
            from: serviceAccount
        target:
          name: logging-loki
          namespace: openshift-logging
  tls:
    ca:
      key: service-ca.crt
      configMapName: openshift-service-ca.crt
  pipelines:
    - name: default-logstore
      inputRefs:
        - application
        - infrastructure
        - audit
      outputRefs:
        - default-lokistack
```

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar contains navigation links for Home, Operator, Workloads, Network, Storage, Builds, Monitoring, Compute, User Management, and Management. The main content area is titled "YAML のインポート" (Import YAML) and displays a YAML configuration for a "ClusterLogForwarder". The YAML code is as follows:

```
1 apiVersion: observability.openshift.io/v1
2 kind: ClusterLogForwarder
3 metadata:
4   name: collector
5   namespace: openshift-logging
6 spec:
7   serviceAccount:
8     name: collector
9   outputs:
10    - name: default-lokistack
11      type: lokiStack
12      lokiStack:
13        authentication:
14          token:
15            from: serviceAccount
16            target:
17              name: logging-loki
18              namespace: openshift-logging
19      tls:
20        ca:
21          key: service-ca.crt
22          configMapName: openshift-service-ca.crt
23      pipelines:
24        - name: default-logstore
25          inputRefs:
26            - application
27            - infrastructure
28            - audit
29            outputRefs:
30              - default-lokistack
```

Several parts of the YAML code are highlighted with red boxes: the "serviceAccount" field, the "target" field under "outputs", and the "inputRefs" and "outputRefs" fields under "pipelines". At the bottom of the editor, there are "作成" (Create) and "キャンセル" (Cancel) buttons.

どの種類のログを転送するかについては、前述したYAMLの `inputRefs:` の項目で指定できます。

- **application** : アプリケーションログの収集。利用者が作成したプロジェクトにデプロイされるアプリケーションのログ(stdoutとstderrに出力されるログ)を収集します。後述のインフラストラクチャー関連のログは除きます。
- **infrastructure** : インフラストラクチャログの収集。ROSAクラスター作成時にデフォルトで作成される `openshift-*`, `kube-*` などのプロジェクトにある、インフラストラクチャ関連のログを収集します。
- **audit** : セキュリティ監査に関するログの収集。ワーカーノードのノード監査システム(`auditd`)で生成される監査ログ(/var/log/audit/audit.log)を収集します。コントロールプレーンの監査ログは、OpenShift Logging Operatorとは別の仕組みで外部転送されており、[Red HatのSREチーム](#)によって1年間保存されます。そのため、ROSAの利用者は監査ログを保存しなくても、[Red Hatのサポートケース](#)経由で監査ログを取得することもできます。

このログ転送設定によって、自動的に `collector-*` という名前のPod(内部ではvectorが実行)が、「`openshift-logging`」プロジェクトに作成されます。この `collector-*` Podは、全てのワーカーノードで自動的に実行されて、ワーカーノード上のログをLokiStackインスタンスに転送します。

Red Hat OpenShift Service on AWS

プロジェクト: openshift-logging

Pods

Pod の作成

名前	ステータス	準備完了	再起動回数	オーナー	メモリー	CPU	作成済み
collector-tc9vx	Running	1/1	0	DS collector	296.8 MiB	0.008 コア	2025年5月1日 18:49
collector-v9rb	Running	1/1	0	DS collector	254.6 MiB	0.008 コア	2025年5月1日 18:49
logging-loki-ingester-0	Running	1/1	0	SS logging-loki-ingester	242.1 MiB	0.005 コア	2025年5月1日 18:28
logging-loki-querier-57dc8f449f-wx2wb	Running	1/1	0	RS logging-loki-querier-57dc8f449f	88.3 MiB	0.004 コア	2025年5月1日 18:28
logging-loki-distributor-864b7cbf46-6ml5w	Running	1/1	0	RS logging-loki-distributor-864b7cbf46	73.7 MiB	0.005 コア	2025年5月1日 18:28
cluster-logging-operator-7975f465bc-mszbv	Running	1/1	0	RS cluster-logging-operator-7975f465bc	64.6 MiB	0.002 コア	2025年5月1日 17:29
logging-loki-gateway-669f67d57f-gmhn	Running	2/2	0	RS logging-loki-gateway-669f67d57f	59.2 MiB	0.003 コア	2025年5月1日 18:28
logging-loki-gateway-669f67d57f-8qh4l	Running	2/2	0	RS logging-loki-gateway-669f67d57f	56.7 MiB	0.006 コア	2025年5月1日 18:28

Contents

デモの概要

Amazon S3をAWS STSで利用するためのIAMロール作成

Amazon S3のバケット作成

OpenShift Loggingに必要となるOperatorのインストール

ログインの設定

ログ転送の設定



OpenShiftコンソールでのログ確認

Contents

演習の概要

OpenShiftコンソールでのログ確認

ローカルユーザーに対するログ集約の参照権限付与

演習の概要

このモジュールでは、OpenShiftのコンソールでのログ集約を確認します。

OpenShiftコンソールでのログ確認

OpenShiftコンソールでのログ集約を確認します。管理者アカウント(`cluster-admin` など)でログインして、左サイドメニューの「モニタリング」 → 「Logs」から確認できます。

NOTE

本演習をワークショップ形式で実施している場合、インストラクターが管理者アカウントを案内します。`user1`とは異なるブラウザを利用することをお勧めします。

The screenshot shows the Red Hat OpenShift Service on AWS console interface. The left sidebar has a dark theme with white text. The 'Logs' section is selected. The main content area displays log entries from May 1, 2025, with columns for Date and Message. There are several filter options at the top right: 'Content' dropdown set to 'Logs', 'Severity' dropdown with checkboxes for critical, error, warning, debug, info, trace, and unknown, and an 'infrastructure' dropdown with options for application, infrastructure, and audit. Below these filters is a 'Show Histogram' button, a time range selector 'Last 1 hour', and a 'Refresh off' button. To the right of the log table are 'Show Resources' and 'Show Stats' buttons, and a 'Show Query' button. The log entries themselves show various system messages, such as proxy requests and kube-probe activity.

表示されたログについて、様々なフィルタリング(ネームスペースやコンテナ単位など)ができるので、色々試してみてください。

ローカルユーザーに対するログ集約の参照権限付与

ローカルユーザーは自分が作ったコンテナアプリケーションのログを、Pod単位で見ることができます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The top navigation bar includes the Red Hat logo, the service name, and a user dropdown for 'testuser20'. The left sidebar has a '開発者' (Developer) section with options like '+追加' (Add), 'トポロジー' (Topology), 'モニタリング' (Monitoring), '検索' (Search), 'Builds', 'Helm', 'プロジェクト' (Project), 'ConfigMaps', and 'シークレット' (Secret). The main content area shows a project named 'test-project20'. Under 'Pods', a pod named 'nodejs-ex-git-5bf8c9db55-7jpth' is listed as 'Running'. The 'ログ' (Logs) tab is selected, showing log streaming for this pod. The logs output the following:

```
1 DEV_MODE=false
2 NODE_ENV=production
3 DEBUG_PORT=5858
4 Launching via npm...
5 npm info using npm@10.8.2
6 npm info using node@v20.18.2
7
8 > nodejs-rest-http-crud@4.0.0 start
9 > node .
10
11
12 { "level":30,"time":1746100131307,"pid":12,"hostname":"nodejs-ex-git-5bf8c9db55-7jpth","msg":"Listening on port 8080"}
13 { "level":50,"time":1746100131312,"pid":12,"hostname":"nodejs-ex-git-5bf8c9db55-7jpth","err":[{"type":"AggregateError","message":""}]}%
```

ただし、デフォルトだと、プロジェクト単位でのログ集約を確認するための権限がローカルユーザーに無いため、下記のようなメッセージが表示されてしまいます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has sections for 開発者 (Developer), +追加 (Add), トポロジー (Topology), モニタリング (Monitoring), 検索 (Search), Builds, Helm, プロジェクト (Project), ConfigMaps, and シークレット (Secret). The main content area is titled 'モニタリング' (Monitoring) and has tabs for イベント (Events), Dashboard, Logs (which is selected), Metrics, アラート (Alerts), and サイレンス (Silences). The Logs tab shows a search bar with 'Content' and 'Severity' dropdowns, and buttons for 'Show Histogram', 'Last 1 hour', 'Refresh off', 'Export as CSV', 'Explain Log Volume', and 'Show Query'. Below the search bar are filters for 'Namespaces' (test-project20) and 'Clear all filters'. The log entries are sorted by 'Date' (descending) and 'Message'. A single entry is shown with a red 'Forbidden' icon and the message 'Missing permissions to get logs'. It also says 'Make sure you have the required role to get application logs in this namespace.' and 'Ask your administrator to grant you this role:'. A detailed YAML configuration for a RoleBinding is shown in a code block:

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: view-application-logs
  namespace: <project-name>
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-logging-application-view
subjects:
- kind: User
  name: <testuser>
  apiGroup: rbac.authorization.k8s.io

```

そこで、管理者アカウントでローカルユーザーに対するログ集約の参照権限を付与してみます。

管理者アカウントで再ログインして、OpenShiftコンソール右上の「+」ボタンから以下のYAMLをインポートします。このYAMLでは `view-application-logs` というRoleBindingリソースを user1-app プロジェクトに作っています。

ユーザーとして user1 も指定しているので user1 ユーザーが user1-app プロジェクト上に作られたアプリケーションに関するログ集約を、OpenShiftコンソール上で見れるようになります。

NOTE

このYAMLの `metadata.namespace` と `Subject.kind.name` については、自分が使っているプロジェクト名とユーザ名に適宜変更してください。

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: view-application-logs
  namespace: user1-app
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-logging-application-view
subjects:
- kind: User

```

```
name: user1
apiGroup: rbac.authorization.k8s.io
```

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar contains navigation links: 開発者, +追加, トポジー, モニタリング, 検索, Builds, Helm, プロジェクト, ConfigMaps, and シークレット. The main area is titled "YAML のインポート" and displays a YAML configuration for a RoleBinding:

```
1 apiVersion: rbac.authorization.k8s.io/v1
2 kind: RoleBinding
3 metadata:
4   name: view-application-logs
5   namespace: test-project20
6 roleRef:
7   apiGroup: rbac.authorization.k8s.io
8   kind: ClusterRole
9   name: cluster-logging-application-view
10 subjects:
11 - kind: User
12   name: testuser20
13   apiGroup: rbac.authorization.k8s.io
```

Two specific fields are highlighted with red boxes: "namespace: test-project20" and "name: testuser20". The interface includes standard UI elements like a toolbar with icons for file operations, a message center with 6 notifications, and a user dropdown for "cluster-admin". Buttons at the bottom are "作成" (Create) and "キャンセル" (Cancel).

RoleBindingリソースを作成したあとに user1 ユーザーで再ログインしてみると、user1-app プロジェクトで作成されたアプリケーションに関するログ集約を確認できます。

The screenshot shows the Red Hat OpenShift Service on AWS interface. The left sidebar includes navigation links for Developers, Add, Topology, Monitoring, Search, Builds, Helm, and Projects. The main content area is titled 'モニタリング' (Monitoring) and shows the 'Logs' tab selected. The top navigation bar for the Logs tab includes 'Events', 'Dashboard', 'Logs' (selected), 'Metrics', 'Alerts', and 'Sirens'. Below this are filter options for 'Content' (Search by Content), 'Severity' (Show Resources, Show Stats), and 'Logs' (Export as CSV, Explain Log Volume). A 'Run Query' button is also present. The log table lists entries from May 1, 2025, at 20:48:51.313 to 20:48:51.059. The first entry is a detailed error message about a network connection issue between pods. Subsequent entries show logs from 'nodejs-ex-git' and 'nodejs-rest-http-crud4.0.0' pods starting up.

ログ集約については、他のアプリケーションをデプロイすることでも確認してみることができます。
色々試してみてください。

Contents

演習の概要

OpenShiftコンソールでのログ確認

ローカルユーザーに対するログ集約の参照権限付与



モニタリング

Contents

演習の概要

 クラスター全体のモニタリング

 利用者が作成したプロジェクトのモニタリング

演習の概要

このモジュールでは、Prometheusによるモニタリングに関する情報を確認します。

ROSAクラスターは、デフォルトでPrometheusをベースとしたモニタリング機能が有効になっており、下記の2つのユースケースで利用されています。

- ROSAクラスター全体のモニタリング (Platform monitoring)
- 利用者が作成したプロジェクトのモニタリング (User-defined projects monitoring)

この2つについて、どのようにROSAクラスターで利用されているかを確認していきます。

NOTE

本演習をワークショップ形式で実施している場合、このモジュールでは、設定作業を実施しません。

クラスター全体のモニタリング

ROSAクラスター全体のリソース利用状況のモニタリング、いわゆる「プラットフォームモニタリング」とRed Hatの公式ドキュメントで定義しているものについては、Red HatのSREチームによって利用されています。[ROSAの責任分担マトリクス](#)によって、プラットフォームモニタリングについては、Red Hatに責任があると定義しているため、ROSAの利用者は、ワーカーノードにおけるユーザーアプリの利用状況の監視に集中できるようになっています。

ROSAクラスターでは、モニタリング機能を提供するPodが、「openshift-monitoring」と「openshift-user-workload-monitoring」という2つのプロジェクトで実行されています。プラットフォームモニタリング機能を提供するPodが「openshift-monitoring」プロジェクトで実行されます。Red Hat SREチームの監視対象のリストは、[公式ドキュメント](#)をご参照ください。これらの一部はワーカーノード上で実行されており、ワーカーノードも含めてSREチームにより監視されています。

これらの情報は、ROSAクラスターの管理者アカウント(`cluster-admin` など)でログインすることで確認できます。

NOTE

本演習をワークショップ形式で実施している場合、インストラクターが管理者アカウントを案内します。

名前	表示名	ステータス	リクエスター	メモリー	CPU	作成済み
PR openshift-customer-monitoring	表示名なし	Active	リクエスターなし	-	-	2023年12月9日 16:26
PR openshift-monitoring	表示名なし	Active	リクエスターなし	4,094.6 MiB	0.149 コア	2023年12月9日 16:25
PR openshift-user-workload-monitoring	表示名なし	Active	リクエスターなし	743.9 MiB	0.057 コア	2023年12月9日 16:25

利用者は「openshift-monitoring」プロジェクトで実行されているPodのリストを見ることができます。これらのPodがワーカーノードで常に動くことを考慮して、ROSAクラスターのサイジングをする必要があります。

NOTE

「openshift-monitoring」プロジェクトのコンポーネントは、Red Hat SREチームの管理下に置かれており、利用者が設定を変更することを、基本的にサポートしていません。

名前	ステータス	準備完了	再起動回数	オーナー	メモリー	CPU	作成済み
P prometheus-k8s-1	Running	6/6	0	SS prometheus-k8s	1,402.5 MiB	0.049 コア	2023年12月9日 16:50
P prometheus-k8s-0	Running	6/6	0	SS prometheus-k8s	1,379.6 MiB	0.051 コア	2023年12月9日 16:50
P thanos-querier-845f7548df-k8bln	Running	6/6	0	RS thanos-querier-845f7548df	171.5 MiB	0.006 コア	2023年12月9日 16:49
P thanos-querier-845f7548df-t2zq8	Running	6/6	0	RS thanos-querier-845f7548df	168.7 MiB	0.006 コア	2023年12月9日 16:49
P alertmanager-main-0	Running	6/6	0	SS alertmanager-main	154.1 MiB	0.003 コア	2023年12月9日 16:50
P alertmanager-main-1	Running	6/6	0	SS alertmanager-main	143.8 MiB	0.003 コア	2023年12月9日 16:50
P kube-state-metrics-5cd69bf5d-tfgl5	Running	3/3	0	RS kube-state-metrics-5cd69bf5d	109.3 MiB	0.002 コア	2023年12月9日 16:49

ROSAクラスターのPrometheusでは、Red HatのSREチームによって、ROSAクラスターのコアコンポーネントのメトリクスデータが永続ボリュームに一定期間保存されるように設定されています。これは、「openshift-monitoring」プロジェクトのPVCの項目を見ることで確認できます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a 'PersistentVolumeClaims' section selected. The main area displays a table of PersistentVolumeClaims for the 'openshift-monitoring' project. The table columns include Name, Status, PersistentVolumes, Capacity, Usage, and StorageClass. Two entries are listed:

Name	Status	PersistentVolumes	Capacity	Usage	StorageClass
PVC prometheus-data-prometheus-k8s-0	Bound	PV pvc-28626527-d9f0-4bdd-8a29-a239272fcf18	100 GiB	1.14 GiB	gp3-csi
PVC prometheus-data-prometheus-k8s-1	Bound	PV pvc-a99c7f9e-9459-4386-88bd-22555673609d	100 GiB	1.13 GiB	gp3-csi

利用者が作成したプロジェクトのモニタリング

利用者が作成したプロジェクトのモニタリングに関するカスタム設定が適用できるPodが「openshift-user-workload-monitoring」プロジェクトで実行されます。これらのPodは、Kubernetesのnode-Selectorを利用して、任意のラベルを付けたワーカーノードに移動することができます。

NOTE

KubernetesのnodeSelectorは、指定したPodを特定のノードで実行するように割り当てる仕組みです。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a 'Pods' section selected. The main area displays a table of Pods for the 'openshift-user-workload-monitoring' project. The table columns include Name, Status, Ready, Restart Count, Owner, Memory, CPU, and Last Seen. Five entries are listed:

Name	Status	準備完了	再起動回数	オーナー	メモリー	CPU	作成済み
Prometheus-user-workload-1	Running	6/6	0	SS prometheus-user-workload	220.7 MiB	0.025 コア	2023年12月9日 16:49
Prometheus-user-workload-0	Running	6/6	0	SS prometheus-user-workload	211.3 MiB	0.026 コア	2023年12月9日 16:49
Thanos-ruler-user-workload-1	Running	4/4	0	SS thanos-ruler-user-workload	123.9 MiB	0.003 コア	2023年12月9日 16:49
Thanos-ruler-user-workload-0	Running	4/4	0	SS thanos-ruler-user-workload	116.5 MiB	0.003 コア	2023年12月9日 16:49
Prometheus-operator-74ccdc5c-gzr6	Running	2/2	0	RS prometheus-operator-74ccdc5c	74.6 MiB	0.001 コア	2023年12月9日 16:49

NOTE

ROSA HCPクラスターでは、`rosa create cluster` でHCPクラスター作成時に、`disable-workload-monitoring` オプションで、利用者のプロジェクトのモニタリングを無効化できます。

利用者のプロジェクトに関するメトリクスデータは、デフォルトでは永続ボリュームに保存される設定にはなっていません。このため、Podの再起動や再作成に伴い、利用者のメトリクスデータが失われる可能性があります。

利用者のメトリクスデータを、200GiBの永続ボリュームを利用して30日間保存するような設定をしたい場合、「openshift-user-workload-monitoring」プロジェクトの `user-workload-monitoring-config` という名前の 設定情報(ConfigMap)を次のYAMLファイルで置き換えることで、自動的にデフォルトのストレージクラス(gp3)を用いたPVCが作成されて、メトリクスデータが保存されるようになります。

NOTE

Kubernetesの `ConfigMap` は、アプリケーションで利用される構成データを保存します。

OpenShiftでのConfigMapは、コンソールの「管理者向け表示」→「Workloads」→「ConfigMaps」から作成、または、既存のConfigMapを編集できます。OpenShift CLI(ocコマンド)を用いた設定方法については、[公式ドキュメント](#)をご参照ください。

NOTE

本演習を自習している時以外、この設定を適用する必要はありません。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      retention: 30d
      volumeClaimTemplate:
        spec:
          resources:
            requests:
              storage: 200Gi
```

SH

Contents

演習の概要

クラスター全体のモニタリング

利用者が作成したプロジェクトのモニタリング



プロジェクトのメトリクスデータの確認

Contents

演習の概要

- 「ダッシュボード」タブ
- 「Metrics」タブ
- 「イベント」タブ

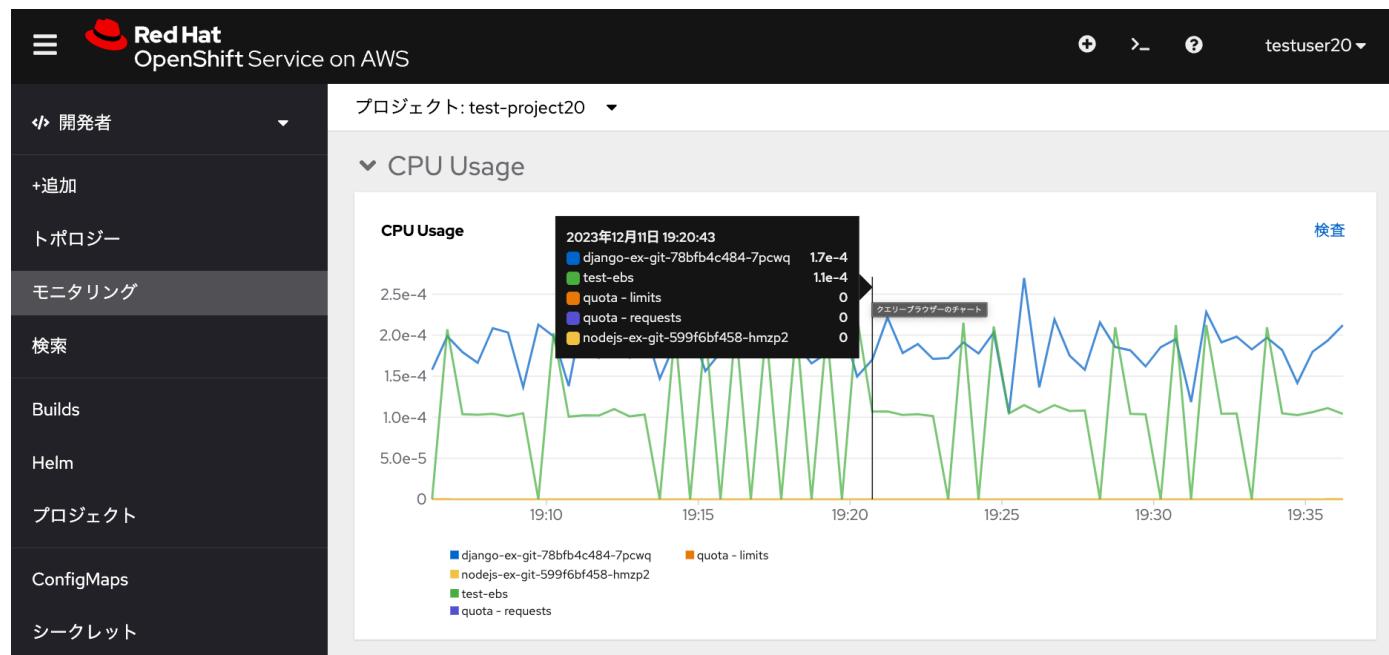
演習の概要

このモジュールでは、ROSAクラスターのコンソールに表示される、各プロジェクトのメトリクスデータを確認します。

「ダッシュボード」タブ

ROSAクラスターのモニタリング (openshift-monitoring) で収集されるメトリクスデータを見てみます。

受講者が最初に作成したプロジェクト(`user1-app`)を選択して、OpenShiftのコンソールから「開発者」→「モニタリング」メニューを選択すると、CPU使用量/メモリ使用量/送受信帯域幅/送受信パケットトレート/送受信パケットドロップレート/ストレージIOに関するグラフを確認できます。



また、管理者アカウントでログインして、`openshift-monitoring`などのプロジェクトを選択すると、CPUとメモリの使用率に関する情報も確認できます。

モニタリング

[ダッシュボード](#) Metrics アラート Silences イベント

ダッシュボード

Kubernetes / Compute Resources / Namespace (Pods) ▾

時間の範囲

更新間隔

最後の 30 分 ▾

30 秒 ▾

CPU Utilisation
(from requests)

検
査

52.72%

CPU Utilisation
(from limits)

検
査

-

Memory Utilisation
(from requests)

検
査

119.94%

Memory Utilisation
(from limits)

検
査

-

▼ CPU Usage

CPU Usage

検査



PodのCPUとメモリ使用については、「リミット(制限)」と「リクエスト(要求)」という値があり、Pod実行時には、予め定義された「リミット」の中で、「リクエスト」された量を確保しようとします。各ワーカーノードに、「リクエスト」に満たないCPU/メモリリソースしかない場合、KubernetesのスケジューラによるPod配置は行われません。

「リミット」がない場合、リクエストされた値以上のリソースが使用される可能性があります。また、「リミット」のみ定義されている場合は、リミットに一致する値がリソースとして、スケジューラによってPodに自動的に割り当てられます。

このダッシュボードにある、CPUやメモリの使用率は、これらの「リミット」と「リクエスト」の値に対してどのくらい使用されているか、という情報となります。

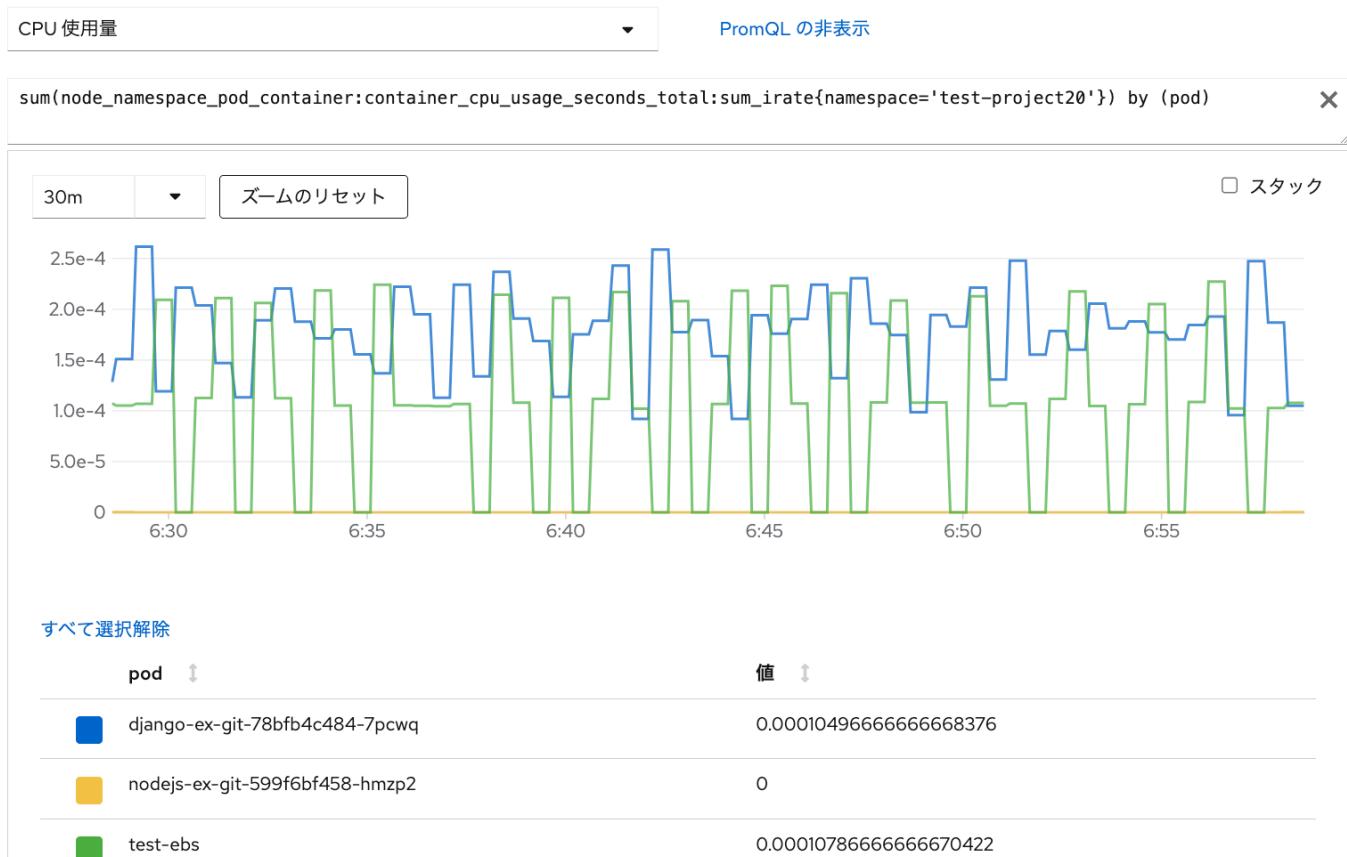
リミットとリクエストについては、[Kubernetesの公式ドキュメント](#)もご参照ください。

「Metrics」タブ

「Metrics」タブでは、[Prometheusのクエリー\(PromQL\)](#)によるグラフ表示が可能です。予め用意されたクエリー(CPUやメモリー使用量など)を用いて、データを確認してみてください。

モニタリング

ダッシュボード Metrics イベント



「イベント」タブ

「イベント」タブでは、プロジェクト上の様々な記録を確認できます。PodやPVCなどを作成した際に実行される様々な操作記録(イベント)がストリーミングされていることを確認してみてください。

これらのイベントは、OpenShiftの様々なクラスター情報を保存する「etcd」データベースに保存されており、保存期間は「3時間」となります。3時間を過ぎたらetcdデータベースから自動的に消去されます。

NOTE

この値はAPI ServerのOperatorによって管理されています。[デフォルトの設定ファイル](#)が利用されており、この中の `event-ttl` 変数で定義されています。[Operator](#)によってこれらの値は保護されており、[OpenShiftクラスターの利用者が編集できない](#)ようになっています。

Red Hat
OpenShift Service on AWS

testuser20

開発者

+追加

トポロジー

モニタリング

検索

Builds

Helm

プロジェクト

ConfigMaps

シークレット

プロジェクト: test-project20

モニタリング

ダッシュボード Metrics イベント

リソース 1 すべてのタイプ / 名前またはメッセージ... /

リソース すべて すべて × ×

イベントをストリーミング中... 1件のイベントの表示

PVC test-pvc-01 NS test-project20 2023年12月11日 19:56
persistentvolume-controller からの生成 直近の 0 分に 4 回
waiting for first consumer to be created before binding

古いイベントは保存されません。

Contents

演習の概要

「ダッシュボード」タブ

「Metrics」タブ

「イベント」タブ



アラート設定

Contents

演習の概要

- アラート設定の有効化
- アラート出力用のサンプルアプリケーションの作成
- ServiceMonitorの作成
- アラートルールの作成
- アラートのテスト

演習の概要

このモジュールでは、プロジェクトのアプリケーションのアラートを設定します。

アラート設定の有効化

ROSAクラスターでは、ROSAの利用者が作成したプロジェクトで実行しているアプリケーションを対象とした、アラート設定が可能です。アラート設定をする場合、利用者が作成したプロジェクトのモニタリングが有効になっている必要があります。

アラート設定を有効化するには、モニタリングの時と同様に、「openshift-user-workload-monitoring」プロジェクトの「user-workload-monitoring-config」設定マップ(ConfigMap)を、次のように末尾3行の「alertmanager: ...」を追加して保存します。

NOTE

本演習を自習している時以外、この設定を適用する必要はありません。

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      retention: 30d
      volumeClaimTemplate:
        spec:
          resources:
            requests:
```

SH



```
storage: 200Gi
alertmanager:
  enabled: true
  enableAlertmanagerConfig: true
```

アラート出力用のサンプルアプリケーションの作成

ユーザープロジェクトを対象としたカスタムアラートを利用するには、Prometheusのフォーマットに沿ったメトリクスを出力するアプリケーションを作成しておく必要があります。そのためのサンプルアプリがありますので、まずはこちらを適当なプロジェクトで作成します。

プロジェクトを選択して、OpenShiftクラスターのWebコンソール右上にある、「+」アイコンをクリックします。

NOTE

OpenShiftでは、このインターフェースからYAML/JSON形式のテキストを直接入力して、Podなどのリソースを作成できます。



サンプルアプリケーションを実行するための、次のYAML形式のテキストを入力して「作成」をクリックします。これによって、レプリカ数2の「prometheus-example-app」Podが実行されます。また、PrometheusメトリクスをOpenShiftクラスター内部で見るために利用する「prometheus-example-app」Serviceも、同時に作成しています。

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: prometheus-example-app
    name: prometheus-example-app
    namespace: user1-app
spec:
  replicas: 2
  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: user1-app
spec:
  ports:
    - port: 8080
      protocol: TCP
      targetPort: 8080
      name: web
  selector:
    app: prometheus-example-app
  type: ClusterIP
```

YAML のインポート

Drag and drop YAML or JSON files into the editor, or manually enter files and use `---` to separate each definition.

⌥ Opt + F1

```
1  ---
2  apiVersion: apps/v1
3  kind: Deployment
4  metadata:
5    labels:
6      app: prometheus-example-app
7      name: prometheus-example-app
8  spec:
9    replicas: 2
10   selector:
11     matchLabels:
12       app: prometheus-example-app
13   template:
14     metadata:
15       labels:
16         app: prometheus-example-app
17     spec:
18       containers:
19         - image: quay.io;brancz/prometheus-example-app:v0.2.0
20           imagePullPolicy: IfNotPresent
21           name: prometheus-example-app
22   ---
23   apiVersion: v1
24   kind: Service
25   metadata:
26     labels:
27       app: prometheus-example-app
28       name: prometheus-example-app
29   spec:
30     ports:
31       - port: 8080
32         protocol: TCP
33         targetPort: 8080
34         name: web
35     selector:
36       app: prometheus-example-app
37     type: ClusterIP
```

作成

キャンセル

ServiceMonitorの作成

「prometheus-example-app」 Serviceを利用したモニタリングのための、[Kubernetesカスタムリソース](#)としてServiceMonitorというリソースが、OpenShiftでは利用できるようになっています。これを

cluster-admin などの管理者アカウントで作成します。ローカルユーザーでログインしている場合は、管理者アカウントで再ログインします。

先ほどYAML形式のテキストを入力した時と同様に、「+」アイコンをクリックしてServiceMonitorを作成します。「interval: 30s」で、メトリクスデータをスクレイピング(収集・加工)する間隔を30秒と設定しています。また、「selector」を指定して、「app: prometheus-example-app」ラベルを持つServiceを対象としています。

NOTE

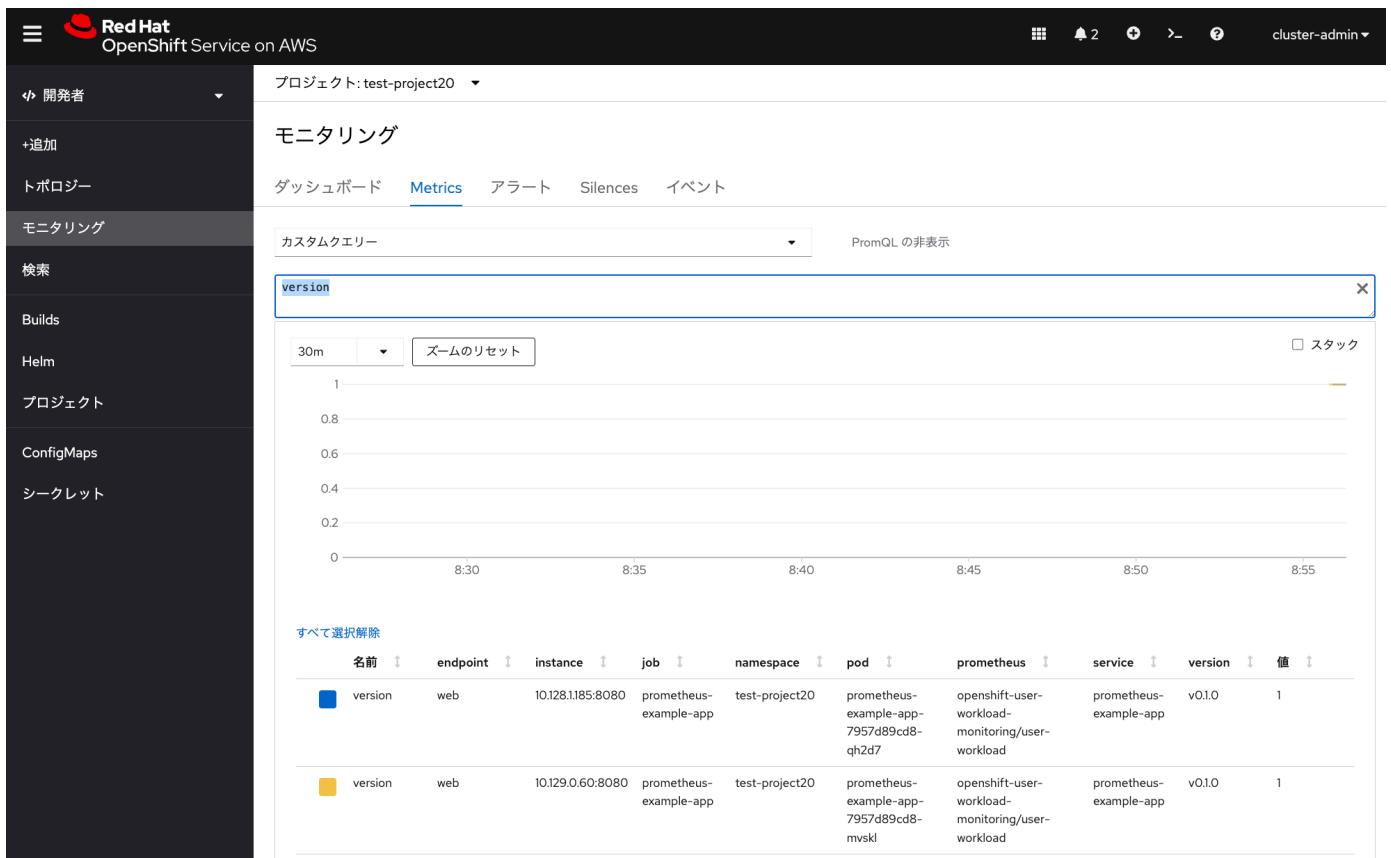
「prometheus-example-app」Serviceを作成したプロジェクトに、このServiceMonitorを作成します。

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    k8s-app: prometheus-example-monitor
  name: prometheus-example-monitor
  namespace: user1-app
spec:
  endpoints:
  - interval: 30s
    port: web
    scheme: http
    path: /metrics
  selector:
    matchLabels:
      app: prometheus-example-app
```

SH



ここまで設定により、「モニタリング」メニューの「Metrics」タブから、このサンプルアプリケーションにあるメトリクスを見れるようになります。「Metrics」タブから「カスタムクエリー」を選択して、version を入力してEnterキーを押します。すると、次のようなメトリクスを確認できます。



アラートルールの作成

簡単なアラートルールを作成してみます。OpenShiftではアラートルール作成のための、Kubernetes力スタムリソースであるPrometheusRuleが利用できるようになっています。1つ以上のPodがダウンしたときに、アラートを発行する設定としてみます。

このサンプルアプリのPodの同時実行数は2となるので、「version」の値の合計値が「2」となっています。そこで、1つ以上Podがダウンしたときの「version」の値の合計値が1(2未満)になるか、または、全てのPodがダウンして「version」メトリクスが取得できない場合を想定した条件式を「expr:」で設定します。「for: 30s」では、アラート発行のための条件式が真となって、アラートが「保留中」状態から「実行中」状態になるまでの時間を30秒と設定しています。

NOTE

このPrometheusRuleも、ServiceMonitorを作成したプロジェクトを選択して、管理者アカウントでYAMLテキストを直接入力して作成します。

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: example-alert
  namespace: user1-app
spec:
  groups:
```

```

- name: prometheus-example-app-down
  rules:
  - alert: PrometheusExampleAppDown
    annotations:
      description: One or more example pods down.
      summary: Example Pods Down.
      expr: sum(version) < 2 or absent(version)
      for: 30s
      labels:
        severity: warning

```

作成したアラートルールや、それに伴ったアラート状態は、管理者アカウントでログインしている場合、「モニタリング」メニューの「アラートタブ」から確認できます。右側にある「・」が3つ縦に並んだアイコンをクリックして、「アラートルールの表示」をクリックすることで、「アラートルールの詳細」を確認できます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a navigation menu with items like '開発者', 'トポロジー', 'モニタリング' (selected), '検索', 'Builds', 'Helm', 'プロジェクト', 'ConfigMaps', and 'シークレット'. The main content area is titled 'AR PrometheusExampleAppDown' (warning). It displays the following details:

- アラートルールの詳細**
- 名前:** PrometheusExampleAppDown
- ソース:** User
- 重大度:** 警告 (Warning)
- 説明:** One or more example pods down.
- 期間:** 30s
- 式:** `sum(version{namespace="test-project20"}) < 2 or absent(version{namespace="test-project20"})`
- ラベル:** `namespace=test-project20`, `severity=warning`

Below this, there's a section titled 'アクティブなアラート' (Active Alerts) with a graph showing a single data point at 8:51:45 on December 12, 2023, with a value of 1. The graph has a zoom button and a link to 'メトリクスでの表示' (View in Metrics).

アラートのテスト

アラートをテストしてみます。「トポロジー」メニューから、「prometheus-example-app」アプリケーションを選択して、詳細タブから「↓矢印」を1回クリックして、Podの数を1つ減らします。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. On the left sidebar, under the 'Monitoring' section, the 'Alerts' tab is selected. In the main content area, a pod named 'prometheus-example-app' is shown with a status of 'Down'. A red box highlights the 'Up' arrow icon next to the pod's name.

これにより、アラート状態が「実行中」に変わります。管理者アカウントでログインしている場合、「モニタリング」メニューの「アラート」タブから確認できます。「One or more example pods down」をクリックすると、アラートの詳細を確認できます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. On the left sidebar, under the 'Monitoring' section, the 'Alerts' tab is selected. In the main content area, an alert titled 'PrometheusExampleAppDown' is listed with a status of 'Running'.

NOTE

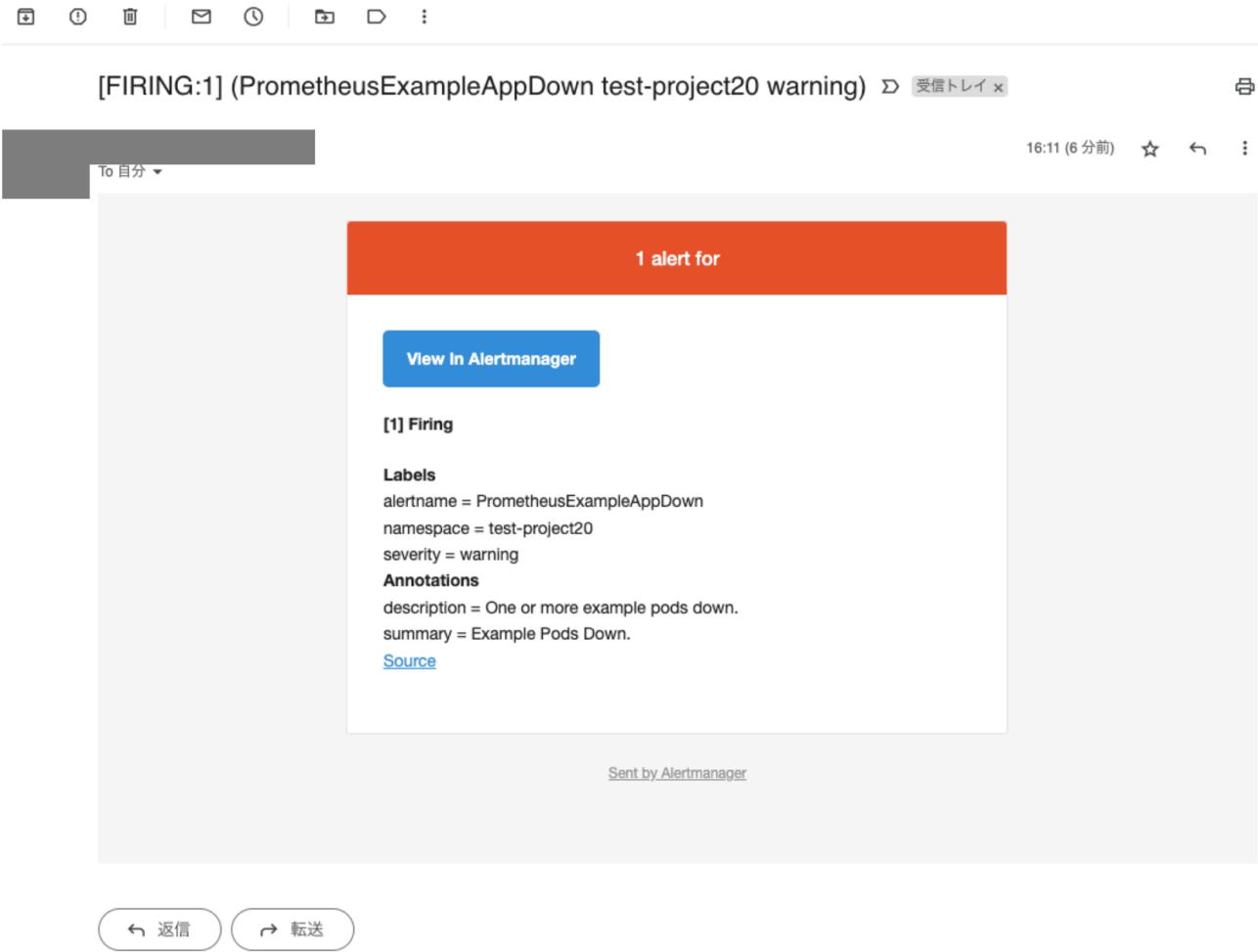
「通知」にあるスイッチをOFFにすると、アラートをサイレンス状態にできます。また、アラートをCLIで確認したい場合は、[こちらのページ](#)にある情報を参考にしてください。

「トポロジー」メニューからサンプルアプリのPod数を再度2に戻す(Podの詳細タブの「↑」矢印をクリック)と、アラート状態が「実行中」から、もともとの何もない状態に戻ります。

ROSAやOpenShiftでは、アラートを下記の外部システムに送信できます。

- PagerDuty
- Webhook
- Email
- Slack

Gmailに届いたアラートメールの例は、下記の画像のようになります。アラート送信の設定方法については、[公式ドキュメント](#)をご参照ください。



Contents

- 演習の概要
- アラート設定の有効化
- アラート出力用のサンプルアプリケーションの作成
- ServiceMonitorの作成
- アラートルールの作成
- アラートのテスト



ワーカーノードの追加と削除

Contents

- 演習の概要
 - ワーカーノードの追加と削除
 - オートスケールの設定
 - オートスケールの確認

演習の概要

このモジュールでは、ワーカーノードの(自動的な)追加と削除を実施します。

ワーカーノードの追加と削除

実行するアプリケーションの数が多くなり、ワーカーノードのリソース(CPUやメモリ)使用率が逼迫した場合、ROSA CLI(rosaコマンド)を使用して、ワーカーノードを簡単に追加・削除できます。ROSAのワーカーノードは、Machinepoolというリソース単位で管理されており、ワーカーノードを追加・削除する場合、このMachinepoolを作成・編集・削除します。

デフォルトで利用されているMachinepoolは、`rosa list machinepool` コマンドで確認します。

NOTE

本演習をワークショップ形式で実施している場合、-c オプションで指定するROSAクラスター名は、`rosa list cluster` コマンドで表示される名前を指定してください。複数のクラスターがある場合は、各受講者にどのクラスターを使うべきかをご案内します。以下の手順は、`hcp-01` という名前のROSAクラスターを例とします。

```
$ rosa list cluster
ID                  NAME      STATE   TOPOLOGY
280scqkn8ocjoochasq423tg4donvpaq  hcp-01  ready  Hosted CP

$ rosa list machinepool -c hcp-01
ID      AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS      TAINTS      AVAILABILITY
ZONE  SUBNET          VERSION    AUTOREPAIR
workers  No           2/2       m5.xlarge
subnet-0087cb7bb3f628793  4.14.2  Yes                    us-east-2a
```

上記の例では、ワーカーノードに対応したAWS EC2インスタンス(デフォルトはm5.large)を2台起動しているという設定を確認できます。

NOTE

このコマンドの出力結果は、利用しているROSA HCPクラスターによって変わることがあります。

ここにmachinepoolを新しく作成して、ワーカーノードを1台追加します。 `rosa create machinepool` コマンドを実行します。

NOTE

1つのROSAクラスターを受講生で共有している場合、他の受講生と重複しないMachinepoolの名前を付けて下さい。

```
$ rosa create machinepool -c hcp-01

I: Enabling interactive mode
? Machine pool name: mp20
? OpenShift version: [Use arrows to move, type to filter, ? for more help]
> 4.14.2
  4.14.1
  4.14.0
? OpenShift version: 4.14.2
? Select subnet for a hosted machine pool: Yes
? Subnet ID: subnet-0087cb7bb3f628793 ('hcp-cluster01-vpc-private-use2-az1', 'vpc-0727149c80d7f166f', 'us-east-2a', Owner ID: '999417968296')
? Enable autoscaling: No
? Replicas: 1
? Labels (optional):
? Taints (optional):
I: Fetching instance types
? Instance type: m5.xlarge
? Autorepair: Yes
I: Machine pool 'mp20' created successfully on hosted cluster 'hcp-01'
I: To view all machine pools, run 'rosa list machinepools -c hcp-01'
```

ROSA HCPクラスターでは、コントロールプレーンより古いバージョンのワーカーノードをデプロイできます。上記の例では、4.14.2を選択していますが、他の古いバージョンも選択できます。

Subnet IDで、ROSA HCPクラスター作成時に指定した、AWS VPCのプライベートサブネットIDを指定します。ROSA HCPクラスターでは、AZにあるサブネットIDを指定して、SingleAZ構成/MultiAZ構成の両方で、ワーカーノードを最低1台から、AZ単位で追加できるようになっています。

Replicasで、作成するワーカーノードの台数(ここでは1台)を指定します。Autorepairは、ワーカーノードの反応が無くなったとき、新規作成したワーカーノードに置換することを意味します。

NOTE

2023年12月時点では、ROSA HCPクラスターではEC2スポットインスタンスの利用はできません。

再度 rosa list machinepool コマンドを実行して、machinepool mp20 が正常に作成されたかを確認します。ワーカーノードに紐づいたEC2インスタンスの作成完了まで、5~10分ほどかかります。

```
$ rosa list machinepool -c hcp-01
```

ID	AUTOSCALING	REPLICAS	INSTANCE TYPE	LABELS	TRAINTS	AVAILABILITY
ZONE	SUBNET			VERSION	AUTOREPAIR	
mp20	No	1/1	m5.xlarge			us-east-2a
subnet-0087cb7bb3f628793		4.14.2	Yes			
workers	No	2/2	m5.xlarge			us-east-2a
subnet-0087cb7bb3f628793		4.14.2	Yes			

ROSAクラスターに管理者アカウント(cluster-admin など)でログインしてみると、「コンピュート」→「Node」メニューから「作成済み」の日時を見ることで、ワーカーノードが新しく作成されていることがわかります。

名前	ステータス	Roles	Pods	メモリー	CPU	ファイル...	作成済み	インスタ...
ip-10-0-0-77.us-east-2.compute.internal	Ready	worker	39	5.19 GiB / 15.16 GiB	0.314 コア / 4 コア	18.66 GiB / 299.8 GiB	2023年12月9日 16:40	m5.xlarge
ip-10-0-0-122.us-east-2.compute.internal	Ready	worker	45	5.91 GiB / 15.32 GiB	0.336 コア / 4 コア	19.31 GiB / 299.8 GiB	2023年12月9日 16:43	m5.xlarge
ip-10-0-0-182.us-east-2.compute.internal	Ready	worker	16	1.49 GiB / 15.32 GiB	0.084 コア / 4 コア	11.63 GiB / 299.8 GiB	2023年12月12日 16:26	m5.xlarge

mp20 に紐づけられているワーカーノードの台数を修正したい場合、rosa edit machinepool コマンドを実行します。下記では、replicas 0 を指定して、ワーカーノードの台数を0台にしています。ROSAクラスターの「Node」メニューから、mp20 に対応したワーカーノード1台が削除されていることを確認できます。

```
$ rosa edit machinepool mp20 -c hcp-01 --replicas 0
I: Updated machine pool 'mp20' on cluster 'hcp-01'
```

```
$ rosa list machinepool -c hcp-01
ID      AUTOSCALING  REPLICAS  INSTANCE TYPE  LABELS      TRAINTS      AVAILABILITY
ZONE    SUBNET          VERSION   AUTOREPAIR
mp20    No            0/0       m5.xlarge      us-east-2a
```

```
subnet-0087cb7bb3f628793 4.14.2 Yes
workers No 2/2 m5.xlarge
subnet-0087cb7bb3f628793 4.14.2 Yes
us-east-2a
```

オートスケールの設定

Machinepoolは、作成時または作成後にオートスケールの設定をすることができます。オートスケールが有効化されていると、利用者がPodをデプロイしようとした時に、リソース(CPUやメモリ)の使用量が逼迫していて、どのワーカーノードにもデプロイできないPodがある場合、自動的にワーカーノードを追加します。

また、その逆に、一部のノードが一定期間にわたって、リソースがあまり使われていない状態が続く場合、ワーカーノードを削除してROSAクラスターのサイズを縮小します。

上記で作成した mp20 のオートスケールの設定変更は、`rosa edit machinepool` コマンドで実行します。次のコマンドでは、最小1台、最大2台のオートスケールの設定の有効化と無効化をしています。

```
$ :↓ オートスケールの有効化
$ rosa edit machinepool mp20 -c hcp-01 --enable-autoscaling=true
? Min replicas: 1
? Max replicas: 2
I: Updated machine pool 'mp20' on hosted cluster 'hcp-01'

$ :↓ オートスケールの無効化
$ rosa edit machinepool mp20 -c hcp-01 --enable-autoscaling=false
? Replicas: 1
I: Updated machine pool 'mp20' on hosted cluster 'hcp-01'
```

Machinepoolの設定・作成・削除は、[OpenShift Cluster Manager \(OCM\)](#)からも実施できます。オートスケールの設定の場合だと、「Enable autoscaling」のチェックボックスによって、オートスケールの有効化/無効化ができます。

NOTE

本演習をワークショップ形式で実施している場合、OCMにはアクセスできません。

The screenshot shows the OpenShift web interface with the URL `Clusters > hcp-01`. The left sidebar has a dark theme with the following navigation items: OpenShift, Overview, Dashboard, Clusters (which is selected), Learning Resources, Releases, Developer Sandbox, and Downloads. The main content area is titled "hcp-01" and shows the "Machine pools" tab selected. The table lists two entries:

	Machine pool	Instance type	Availability zo...	Node count	Autoscaling	Version	Actions
>	mp20	m5.xlarge	us-east-2a	1	Disabled	4.14.2	⋮
>	workers	m5.xlarge	us-east-2a	2	Disabled	4.14.2	Edit Delete

Edit machine pool

x

Machine pool

mp20

▼

Scaling

Enable autoscaling ?

Autoscaling automatically adds and removes worker (compute) nodes from the cluster based on resource requirements.

Minimum nodes count * *

- 1 +

Maximum nodes count * ?

- 2 +

> Edit node labels and taints

Save

Cancel

作成したMachinepoolを削除する場合、 `rosa delete machinepool` コマンドを実行します。これによってワーカーノードが削除され、その上で実行されているPodも削除されます。

NOTE

次の「オートスケールの確認」演習を実施する場合、`rosa delete machinepool` コマンドはまだ実行しないで下さい。

```
$ rosa delete machinepool mp20 -c hcp-01  
? Are you sure you want to delete machine pool 'mp20' on hosted cluster 'hcp-01'?  
Yes  
I: Successfully deleted machine pool 'mp20' from hosted cluster 'hcp-01'
```

オートスケールの確認

オートスケールが正常に動作するかを実際に確認してみます。Machinepoolを作成した時に、それに紐づいたワーカーノードに自動的に付与されるラベルを利用して、ワーカーノードの台数の増減を確認します。

前の手順で作成したMachinepoolに対して、再度オートスケールを有効化します。

```
$ rosa edit machinepool mp20 -c hcp-01 --enable-autoscaling=true  
? Min replicas: 1  
? Max replicas: 2  
I: Updated machine pool 'mp20' on hosted cluster 'hcp-01'
```

Machinepoolに紐づくワーカーノードに自動付与されるラベルのうち、ROSAクラスターのコンソールの「コンピュート」→「Node」メニューから当該ノードを選択して、「詳細」タブの「ラベル」に表示されている `hypershift.openshift.io/nodePool=<ROSAクラスター名>-<Machinepool名>` を使います。下記の画像の例では、`hypershift.openshift.io/nodePool=hcp-01-mp20` というラベルが、ワーカーノードに付与されています。

The screenshot shows the Red Hat OpenShift web console interface. On the left, there's a sidebar with navigation links: ホーム, Operator, Workloads, ネットワーク, ストレージ, Builds, モニタリング, and コンピュート (Compute). Under Compute, 'Node' is selected. In the main content area, there's a 'ラベル' (Labels) section with a '編集' (Edit) button. A list of labels is shown, including several system labels like beta.kubernetes.io/os=linux, hypershift.openshift.io/managed=true, failure-domain.beta.kubernetes.io/zone=us-east-2a, etc., and one specific label 'hypershift.openshift.io/nodePool=hcp-01-mp20' which is highlighted with a red border. To the right of the labels, there are sections for 'コンテナーランタイム' (Container Runtime), 'Kubelet バージョン' (Kubelet Version), and 'Kube-Proxy バージョン' (Kube-Proxy Version), each with their respective versions listed.

NOTE

`rosa create machinepool` コマンドでMachinepoolを作成する時に、`labels=key1=value1,key2=value2,...` 形式のオプションを指定することで、任意のラベルを付与できます。「key」と「value」については、任意の文字列を指定できます。2023年12月時点では、`rosa edit machinepool` コマンドでMachinepoolを編集する際にラベルを付与しても、Machinepoolに紐づいたワーカーノードにラベルが付与されないというバグがありますので、ご注意ください。

ここで、実際にサンプルジョブを投入して確認してみましょう。次のYAMLファイルで busybox Pod を15個並列に実行するジョブを投入します。このとき、先ほど確認したラベルを利用して、このジョブによって作成されるPodが、受講者が作成したMachinepool内だけで実行されるように、「nodeSelector」を指定します。

ラベルの「key: value」の「value」に相当する文字列(この例では、`hcp-01-mp20`)は、ダブルクオーテーションで囲む必要があります。これを忘れるとな、「value」の値が文字列として認識されないため、ラベルの指定ができず、CPU/メモリのリソースが空いている任意のワーカーノードでPodが実行されるようになるため、注意してください。

```
apiVersion: batch/v1
kind: Job
metadata:
  generateName: work-queue-
spec:
  template:
    spec:
      nodeSelector:
        hypershift.openshift.io/nodePool: "hcp-01-mp20"
```

SH

```

containers:
- name: work
  image: busybox
  command: ["sleep", "360"]
resources:
  requests:
    memory: 500Mi
    cpu: 500m
restartPolicy: Never
backoffLimit: 4
completions: 15
parallelism: 15

```

OpenShiftでのジョブは、「ワークロード」メニューの「ジョブ」から「Jobの作成」をクリックして、上記YAMLファイルをコピペして「作成」をクリックすることで作成できます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar has a 'Jobs' section highlighted. The main content area is titled 'Job の作成' and contains a code editor with the following YAML configuration:

```

1 apiVersion: batch/v1
2 kind: Job
3 metadata:
4   generateName: work-
5 spec:
6   template:
7     spec:
8       nodeSelector:
9         hypershift.openshift.io/nodePool: "hcp-01-mp20"
10      containers:
11        - name: work
12          image: busybox
13          command: ["sleep", "360"]
14          resources:
15            requests:
16              memory: 500Mi
17              cpu: 500m
18          restartPolicy: Never
19      backoffLimit: 4
20      completions: 15
21      parallelism: 15

```

At the bottom of the editor are buttons for '作成' (Create), 'キャンセル' (Cancel), and 'ダウンロード' (Download).

ジョブを実行して数分待つと、ジョブの「Pod」から次のような実行状況の画面を確認できます。この画像の例では、最初にワーカーノード「ip-10-0-0-48.XXX」で一部のPodがジョブによって実行され、Machinepoolのオートスケールの設定により、ワーカーノード「ip-10-0-0-138.XXX」が自動的に追加され、ジョブのPodを並列に実行していることを示しています。

プロジェクト: test-project20

Jobs > Job の詳細

work-queue-stk94 In progress

アクション

詳細 YAML Pods イベント

名前	ステータス	準備完了	再起動回数	ノード	メモリー	CPU	作成済み
P work-queue-stk94-2rhq5	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.6 MiB	-	2023年12月12日 19:11
P work-queue-stk94-dmkjg	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.6 MiB	-	2023年12月12日 19:11
P work-queue-stk94-gg8v2	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.6 MiB	-	2023年12月12日 19:11
P work-queue-stk94-rmcz4	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.6 MiB	-	2023年12月12日 19:11
P work-queue-stk94-vpfps	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.6 MiB	-	2023年12月12日 19:11
P work-queue-stk94-xt5lt	Running	1/1	0	N ip-10-0-0-138.us-east-2.compute.internal	0.7 MiB	-	2023年12月12日 19:11
P work-queue-stk94-6dqjk	Completed	0/1	0	N ip-10-0-0-48.us-east-2.compute.internal	-	-	2023年12月12日 19:11
P work-queue-stk94-44nk6	Completed	0/1	0	N ip-10-0-0-48.us-east-2.compute.internal	-	-	2023年12月12日 19:11
P work-queue-stk94-dvsqb	Completed	0/1	0	N ip-10-0-0-48.us-east-2.compute.internal	-	-	2023年12月12日 19:11
P work-queue-stk94-hbk85	Completed	0/1	0	N ip-10-0-0-48.us-east-2.compute.internal	-	-	2023年12月12日 19:11

この他にも、管理者アカウントでログインしたROSAクラスターのコンソールの「コンピュート」→「Node」メニューから、自動的にワーカーノードが追加されている状況を確認できます。

前述のコマンドで作成したm5.xlargeインスタンスのMachinepoolを利用して、このオートスケールのテストを実行した場合、所要時間の内訳は下記となり、合計で大体25分ほどかかります。

- ジョブの実行開始から完了まで: 15分ほど
- ジョブの実行完了から、追加されたワーカーノード1台の自動削除が完了するまで: 10分ほど

途中でジョブの実行を中止したい場合、当該ジョブの「Jobの削除」を選択して「削除」をクリックすることで、ジョブを削除できます。これにより、ジョブによって起動されたPodが全て削除され、10分ほど経つと、追加されたワーカーノード1台が自動的に削除されます。

The screenshot shows the Red Hat OpenShift Service on AWS web interface. The left sidebar is collapsed, and the main area shows the 'Jobs' page for the project 'test-project20'. The table lists a single job named 'work-queue-stk94' with status '15 of 15' completed. A context menu is open over this job, listing options: '並列処理の編集', 'PodDisruptionBudget の追加', 'ラベルの編集', 'アノテーションの編集', 'Job の編集', and 'Job の削除'. The 'Jobs' option in the sidebar is highlighted.

名前	ラベル	完了	タイプ
work-queue-stk94	batch.kubernetes... =e819f58f-3c6b-... batch.kubernetes... =work-queu... contr... =e819f58f-3c6b-448c-... job-name=work-queue-stk94	15 of 15	固定の完了数

最後に、受講者が作成したMachinepoolを、 rosa delete machinepool コマンドで削除します。

```
$ rosa delete machinepool mp20 -c hcp-01

? Are you sure you want to delete machine pool 'mp20' on hosted cluster 'hcp-01'?
Yes
I: Successfully deleted machine pool 'mp20' from hosted cluster 'hcp-01'
```

Contents

- 演習の概要
- ワーカーノードの追加と削除
- オートスケールの設定
- オートスケールの確認



ROSA HCPクラスターの更新

Contents

演習の概要

- OCMを利用したコントロールプレーンの更新
- OCMを利用したワーカーノードの更新
- ROSA CLIを利用したコントロールプレーンの更新
- ROSA CLIを利用したワーカーノードの更新

演習の概要

このモジュールでは、ROSA HCPクラスターの更新を実行します。

ROSA HCPクラスターは、コントロールプレーンとワーカーノードから構成されており、この2つは、個別に更新されます。最初にコントロールプレーンを更新して、次にワーカーノードをMachinepool単位で更新していきます。更新方法は、SingleAZ/MultiAZ構成の共に同じものとなります。

ここでは、OCMとROSA CLIを利用した更新方法をご紹介します。

NOTE

本演習をワークショップ形式で実施している場合、「ROSA CLIを利用したワーカーノードの更新」以外の項目を実施できません。これらの項目については、インストラクターによる紹介のみとなります。

OCMを利用したコントロールプレーンの更新

[OpenShift Cluster Manager \(OCM\)](#)を利用して、ROSA HCPクラスターを手動で更新できます。ROSAの場合、通常のOpenShiftとは異なり、OpenShiftのWebコンソールとCLI(ocコマンド)によるアップグレードができないようになっています。そのため、OCMや後述するROSA CLIによるアップグレードを実施する必要があります。

[OCMにログイン](#)して、コントロールプレーンを更新するROSA HCPクラスターを選択し、Settingsタブをクリックして、「Update」ボタンをクリックします。

hcp-01[Open console](#)

Actions ▾


[Overview](#) [Access control](#) [Add-ons](#) [Cluster history](#) [Networking](#) [Machine pools](#) [Support](#) [Settings](#)
Update strategy

Note: In the event of [Critical security concerns](#) (CVEs) that significantly impact the security or stability of the cluster, updates may be automatically scheduled by Red Hat SRE to the latest z-stream version not impacted by the CVE within 2 business days after customer notifications.

 Recurring updates

The cluster control plan will be automatically updated based on your preferred day and start time when new patch updates ([z-stream](#)) are available. When a new minor version is available, you'll be notified and must manually allow the cluster to update to the next minor version. The worker nodes will need to be manually updated.

 Individual updates

Schedule each update individually. Take into consideration end of life dates from the [lifecycle policy](#) when planning updates.

Node draining

You may set a grace period for how long pod disruption budget-protected workloads will be respected during updates. After this grace period, any workloads protected by pod disruption budgets that have not been successfully drained from a node will be forcibly evicted.

Grace period

[Save](#)[Cancel](#)
Feedback
**Update status**

Update available

4.14.2 —————→ 4.14.5

[Update](#)

ここで「Update strategy」の「Recurring updates」を選択すると、指定した日時の2日前以上にリリースされたz-streamの更新(4.14.5など)が、ROSA HCPクラスターのコントロールプレーンに、指定したタイミングで、毎週自動適用されます。

新しいマイナーバージョン(4.15や4.16など)がリリースされた場合は、利用者にメールなどで通知され、次のマイナーバージョンに更新されることを手動で許可する必要があります。

OCMでコントロールプレーンの自動更新を有効にしている場合でも、ワーカーノードは手動で更新する必要があります。

Update strategy

Note: In the event of [Critical security concerns](#) (CVEs) that significantly impact the security or stability of the cluster, updates may be automatically scheduled by Red Hat SRE to the latest z-stream version not impacted by the CVE within 2 business days after customer notifications.

Recurring updates

The cluster control plan will be automatically updated based on your preferred day and start time when new patch updates ([z-stream](#)) are available. When a new minor version is available, you'll be notified and must manually allow the cluster to update to the next minor version. The worker nodes will need to be manually updated.

i For recurring updates, the control plane will be updated when a new version becomes available at least 2 days prior to your selected start time. Worker nodes will need to be manually updated.

Select a day and start time

Saturday	▼	02:00 UTC	▼
----------	---	-----------	---

Individual updates

Schedule each update individually. Take into consideration end of life dates from the [lifecycle policy](#) when planning updates.

更新するバージョンを選択して、「Next」をクリックします。

1 Select version

2 Schedule update

3 Confirmation

Select version

4.14.5

★ Recommended

The latest on your current minor version.

[View release notes](#) ↗

Next

Back

Cancel

コントロールプレーンの更新をスケジュールします。デフォルトでは「Update now」が選択されており、ROSA HCP クラスターでは、現在の時刻から約5分後に更新されるようスケジューリングされます。指定した時間にアップグレードするには、「Schedule a different time」を選択し、アップグレードの日時を設定します。どちらかを選択して、「Next」をクリックします。

1 Select version

2 Schedule update

3 Confirmation

Schedule update

- Update now (update will begin within the next hour)
 Schedule a different time

2023-12-13



12:00



UTC 13 Dec 2023 03:00 UTC

Next

Back

Cancel

更新するバージョンとスケジュールを確認したら、「Confirm Update」をクリックして、更新をスケジュールします。

- 1 Select version
- 2 Schedule update
- 3 Confirmation

Confirmation of your update

Version 4.14.2 → 4.14.5

Scheduled UTC 13 Dec 2023 03:00 UTC

Local time Wed Dec 13 2023 12:00:00 GMT+0900 (日本標準時)

[Confirm update](#)

[Back](#)

[Cancel](#)

コントロールプレーンのアップグレードをキャンセルしたい場合、「Cancel this update」からキャンセルできます。

Clusters > hcp-01

hcp-01

[Open console](#)

Actions ▾



[Overview](#)

[Access control](#)

[Add-ons](#)

[Cluster history](#)

[Networking](#)

[Machine pools](#)

[Support](#)

[Settings](#)

Update strategy

Note: In the event of [Critical security concerns](#) (CVEs) that significantly impact the security or stability of the cluster, updates may be automatically scheduled by Red Hat SRE to the latest z-stream version not impacted by the CVE within 2 business days after customer notifications.

Recurring updates

The cluster control plan will be automatically updated based on your preferred day and start time when new patch updates ([z-stream](#)) are available. When a new minor version is available, you'll be notified and must manually allow the cluster to update to the next minor version. The worker nodes will need to be manually updated.

Update status

Update available

4.14.2 → 4.14.5

[Cancel this update](#)

Cancel update

×

This update to version 4.14.5 is scheduled for 13 Dec 2023 03:00 UTC.

[Cancel this update](#)

[Close](#)

これによって、予定された時刻になると、ROSA HCPクラスターのコントロールプレーン更新が開始されます。コントロールプレーン更新の所要時間は、およそ15分ほどです。

OCMを利用したワーカーノードの更新

ROSA HCPクラスターでは、コントロールプレーンより古いバージョンのワーカーノードのMachinepoolがあると、Machinepool単位でワーカーノードを更新できます。

OCMでは、更新可能なMachinepoolがある場合、指定したMachinepol、または、全てのMachinepoolを更新できます。OCMでMachinepoolを更新する場合、約5分後に更新がスケジュールされます。更新時刻を指定したい場合、ROSA CLIを利用する必要があります。

hcp-01

[Open console](#)

Actions ▾

[Overview](#) [Access control](#) [Add-ons](#) [Cluster history](#) [Networking](#) [Machine pools](#) [Support](#) [Settings](#)⚠️ **Update available for Machine pools**

You can update all worker nodes to the current control plane version (4.14.2), or use the CLI to update a specific version. [Learn more about updates](#) ↗

[Update all Machine pools now](#)[Add machine pool](#)

Machine pool	Instance type	Availability zones	Node count	Autoscaling	Version	⋮
mp01	c5.xlarge	us-east-2a	1	Disabled	4.14.0 Update ⓘ	⋮
Subnets						⋮
subnet-0087cb7bb3f628793						⋮
Labels						Feedback ↗
testkey01 = testvalue01						
Subnets						⋮
subnet-0087cb7bb3f628793						⋮

Update machine pool



Update machine pool mp01 to version 4.14.2?

[Update machine pool](#)[Cancel](#)

hcp-01

[Open console](#)

Actions ▾


[Overview](#) [Access control](#) [Add-ons](#) [Cluster history](#) [Networking](#) [Machine pools](#) [Support](#) [Settings](#)
[Add machine pool](#)

Machine pool	Instance type	Availability zones	Node count	Autoscaling
--------------	---------------	--------------------	------------	-------------

▼ mp01	c5.xlarge	us-east-2a	1	Disabled	4.14.0 ?	⋮
--------	-----------	------------	---	----------	--------------------------	---

Subnets

subnet-0087cb7bb3f628793

▼ workers	m5.xlarge	us-east-2a	2	Disabled	4.14.2	⋮
-----------	-----------	------------	---	----------	--------	---

Labels

testkey01 = testvalue01

Subnets

subnet-0087cb7bb3f628793

Feedback
↗

This machine pool is scheduled to
be updated at 12 Dec 2023 23:58
UTC to version 4.14.2

ワーカーノードの更新は、インプレースアップグレードではなく、新規ワーカーノードの追加と既存ワーカーノードの削除を実行します。Machinepoolに複数台のワーカーノードが紐づいている場合、台数分のワーカーノードを一括作成および削除するのではなく、ワーカーノードの新規追加と削除が1台ずつ自動実行されていきます。

更新の際に、削除対象のワーカーノード上でPodが実行している場合、Podの停止(SIGTERM)と再作成が実行されますので、冗長性を考慮してPodのレプリカ数を複数個設定しておくことを推奨します。Podの停止不可の場合は、Podの強制停止(SIGKILL)が実行されます。これについては、[公式ドキュメント](#)をご参照ください。

NOTE

このROSA Labでは詳細を扱いませんが、[Podのレプリカ数](#)は、Kubernetesのワークロード(デプロイ設定に利用するDeploymentなど)の中で定義できます。

なお、1台のワーカーノードの更新(新規ノード作成と既存ノードの削除)にかかる所要時間は、およそ5~10分ほどです。

NOTE

ROSA HCP 4.15+ではMachinePool内の特定のノードについて、[Podのスケジューリング可否の設定\(cordon/uncordon\)](#)と[Podの退避\(drain\)](#)を実行できるようになりました。これによって、ワーカーノードの更新や削除に伴って必要となるアプリケーションの事前退避が可能になります。

ROSA CLIを利用したコントロールプレーンの更新

OCMのコンソールの他に、ROSA CLIを使用してROSA HCPクラスターの コントロールプレーンの更新をスケジュールできます。次のコマンドを実行して、利用可能な更新情報を確認します。

```
$ rosa list upgrade cluster -c hcp-01
VERSION NOTES
4.14.5 recommended
```

ここで確認した更新情報を適用するためのスケジュールを設定します。下記は、2023年12月13日の23時(UTC)に、更新をスケジューリングしている例です。 rosa upgrade cluster コマンドでは、ROSAクラスターやそのOperatorによって利用されるIAMロールも、適宜更新されます。

```
$ rosa upgrade cluster -c hcp-01 --control-plane \
--schedule-date 2023-12-13 --schedule-time 23:00 \
--version 4.14.5 --mode auto --yes

I: Ensuring account and operator role policies for cluster
'280scqkn8ocjoochasq423tg4donvpaq' are compatible with upgrade.
I: Account roles with the prefix 'ManagedOpenShift' have attached managed policies.
I: Cluster 'hcp-01' operator roles have attached managed policies. An upgrade isn't
needed
I: Account and operator roles for cluster 'hcp-01' are compatible with upgrade
I: Upgrade successfully scheduled for cluster 'hcp-01'
```

指定した更新のスケジュールを確認できます。

```
$ rosa list upgrade cluster -c hcp-01
VERSION NOTES
4.14.5 recommended - scheduled for 2023-12-13 23:00 UTC
```

rosa delete upgrade コマンドで、更新のキャンセルすることができます。キャンセルすると、更新のスケジュールが削除されていることを確認できます。

```
$ rosa delete upgrade cluster -c hcp-01
? Are you sure you want to cancel scheduled upgrade on cluster 'hcp-01'? Yes
I: Successfully canceled scheduled upgrade on cluster 'hcp-01'

$ rosa list upgrade cluster -c hcp-01
```

VERSION	NOTES
4.14.5	recommended

ROSA CLIを利用したワーカーノードの更新

Machinepoolを新規に作成して、更新コマンド確認用のワーカーノードを1台追加します。この時、選択する「OpenShift version」は、最新より古いバージョンを選択してください。また、「Instance type」は、デフォルトの `m5.xlarge` と区別しやすいように `c5.xlarge` を指定します。

```
$ rosa create machinepool -c hcp-01

I: Enabling interactive mode
? Machine pool name: mp20
? OpenShift version: [Use arrows to move, type to filter, ? for more help]
  4.14.2
  4.14.1
> 4.14.0
? OpenShift version: 4.14.0
? Select subnet for a hosted machine pool: No
? AWS availability zone: us-east-2a
? Enable autoscaling: No
? Replicas: 1
? Labels (optional):
? Taints (optional):
I: Fetching instance types
? Instance type: c5.xlarge
? Autorepair: Yes
I: Machine pool 'mp20' created successfully on hosted cluster 'hcp-01'
I: To view all machine pools, run 'rosa list machinepools -c hcp-01'
```

`rosa list machinepool` コマンドや、管理者アカウントでログインしたROSAクラスターのコンソールの「コンピュート」→「Node」メニューから、追加したMachinepoolに紐づいたワーカーノードが1台作成されていることを確認できるまで待ちます。

ワーカーノードの作成が完了したら、`rosa list upgrade` コマンドで、`machinepool` オプションを指定して、作成したMachinepoolの更新情報を確認します。次の例では、「4.14.1」と「4.14.2」のバージョンに更新できることが表示されています。

```
$ rosa list upgrade -c hcp-01 --machinepool mp20
VERSION NOTES
4.14.2 recommended
4.14.1
```

Machinepoolの更新を `rosa upgrade machinepool` コマンドでスケジュールします。
`interactive` オプションで、対話形式でパラメータを与えていくように指定します。なお、時刻は UTC形式で指定します。日本時間はこれより9時間進んでいることを考慮してください。

```
$ rosa upgrade machinepool mp20 -c hcp-01 --interactive

I: Interactive mode enabled.
Any optional fields can be left empty and a default will be selected.
? Enable automatic upgrades: No
? Please input desired date in format yyyy-mm-dd: 2023-12-13
? Please input desired UTC time in format HH:mm: 23:00
? Machine pool version: 4.14.1
? Are you sure you want to upgrade machine pool 'mp20' to version '4.14.1'? Yes
I: Upgrade successfully scheduled for the machine pool 'mp20' on cluster 'hcp-01'
```

更新実行を待つ場合は、ROSAクラスターに管理者アカウントでログインして、「コンピュート」→「Node」メニューから、Machinepoolに紐づいたワーカーノードが新規作成されて、削除されることを確認してみてください。

Machinepool更新のスケジュールを削除する場合は、`rosa delete upgrade` コマンドを実行します。

```
$ rosa delete upgrade -c hcp-01 --machinepool mp20 --yes
I: Successfully canceled scheduled upgrade for machine pool 'mp20' for cluster
'hcp-01'
```

最後に、Machinepool更新について一通り確認した後は、不要になったMachinepoolを、`rosa delete machinepool` コマンドで削除してください。

```
$ rosa delete machinepool mp20 -c hcp-01 --yes
I: Successfully deleted machine pool 'mp20' from hosted cluster 'hcp-01'
```

Contents

- 演習の概要
- OCMを利用したコントロールプレーンの更新
- OCMを利用したワーカーノードの更新
- ROSA CLIを利用したコントロールプレーンの更新
- ROSA CLIを利用したワーカーノードの更新