

---

**FUJITSU Hybrid IT Service**  
**FJcloud-O IaaS**  
**Red Hat**  
**OpenShift Container Platform 4**

---

スタートガイド

10.0 版

# 目次

本書をお読みになる前に .....	4
<b>1. 概要</b> .....	11
<b>1.1 Red Hat OpenShift Container Platform とは</b> .....	11
<b>1.1.1 主な機能</b> .....	11
<b>1.1.2 アーキテクチャー</b> .....	12
<b>1.1.3 ライフサイクル</b> .....	12
<b>1.2 想定システム構成</b> .....	13
<b>2. 動作環境の作成</b> .....	20
<b>2.1 仮想サーバーの作成</b> .....	20
<b>2.2 DNS サーバーの設定</b> .....	20
<b>2.3 HTTP サーバーの設定</b> .....	22
<b>2.4 NFS サーバーの設定</b> .....	22
<b>2.5 クライアントサーバーの設定</b> .....	22
<b>2.5.1 クライアントツールの設定</b> .....	22
<b>2.5.2 サーバー設定ファイルの作成</b> .....	25
<b>3. コンテナアプリケーションプラットフォームの構築 (RHCOS)</b> .....	29
<b>3.1 仮想サーバーの作成</b> .....	29
<b>3.2 イメージレジストリーの設定</b> .....	32
<b>3.3 Infra サーバーの設定</b> .....	35
<b>3.3.1 ラベル設定</b> .....	35
<b>3.3.2 イメージレジストリーPod の配備先変更</b> .....	35
<b>3.3.3 ルーターPod の配備先変更</b> .....	37
<b>3.3.4 モニタリング Pod の配備先変更</b> .....	38
<b>3.3.5 アプリケーション Pod の配備先変更</b> .....	42
<b>3.3.6 管理設定の追加</b> .....	43
<b>3.4 コンテナアプリケーションプラットフォーム構築の完了確認</b> .....	44
<b>3.5 Bootstrap サーバーの削除</b> .....	46
<b>4. コンテナアプリケーションプラットフォームの構築 (RHEL)</b> .....	47
<b>4.1 仮想サーバーの作成</b> .....	47
<b>4.2 Worker サーバーの作成</b> .....	49

4.2.1	SSH 公開鍵の登録.....	49
4.2.2	rpm パッケージのインストール先の設定.....	49
4.2.3	rpm パッケージのアップデート.....	51
4.3	クライアントサーバーの設定.....	52
4.3.1	rpm パッケージのインストール.....	52
4.3.2	Ansible の設定.....	53
4.4	Ansible の実行.....	54
4.5	サーバーの状態の確認.....	56
4.5.1	証明書の許可.....	57
4.6	イメージレジストリーの設定.....	57
4.7	Infra サーバーの設定.....	58
4.8	コンテナアプリケーションプラットフォーム構築の完了確認.....	58
4.9	Bootstrap サーバーの削除.....	58
付録 A	RHEL コンテナイメージへのパッケージの追加.....	59
A.1	Satellite サーバーの情報の取得.....	59
A.2	Satellite サーバーを利用したパッケージの取得.....	62
修正履歴	.....	66

## 本書をお読みになる前に

### ■ 本書の目的

本書は FUJITSU Hybrid IT Service FJcloud-O IaaS 上で Red Hat OpenShift Container Platform を利用するための環境構築手順を示したものです。

本書は、以下の製品を対象に説明しています。

- Red Hat OpenShift Container Platform 4.8
- Red Hat OpenShift Container Platform 4.9
- Red Hat OpenShift Container Platform 4.10
- Red Hat OpenShift Container Platform 4.11
- Red Hat OpenShift Container Platform 4.12

### ■ 本書の読者

本書の読者は FUJITSU Hybrid IT Service FJcloud-O IaaS 上で仮想ネットワーク、仮想ストレージ、仮想サーバー、ロードバランサーの作成および設定ができること、DNS サーバー、HTTP サーバー、NFS サーバーの作成および設定ができること、Red Hat 認定システム管理者(RHCSA)を取得または同程度のスキルと知識を保有していることを前提としています。

### ■ 本書の構成

本書の構成を以下に示します。

章	内容
1. 概要	Red Hat OpenShift Container Platform の製品概要を説明しています。
2. 動作環境の作成	コンテナアプリケーションプラットフォームが動作する環境の作成手順を説明しています。
3. コンテナアプリケーションプラットフォームの構築 (RHCOS)	Worker サーバーに Red Hat Enterprise Linux CoreOS(RHCOS) を使用してコンテナアプリケーションプラットフォームを構築する手順を説明しています。
4. コンテナアプリケーションプラットフォームの構築 (RHEL)	Worker サーバーに Red Hat Enterprise Linux(RHEL)を使用してコンテナアプリケーションプラットフォームを構築する手順を説明しています。
付録A RHEL コンテナイメージへのパッケージの追加	コンテナアプリケーションプラットフォームの使用用途によっては必要になる環境構築手順を説明しています。

## ■ 製品表記

本書では、以降の製品名称を次のように略して表記します。

製品名称	本文中の表記
Red Hat OpenShift Container Platform	RHOCP
Red Hat OpenShift Container Platform 4. <i>n</i> (注)	RHOCP4
Red Hat Enterprise Linux	RHEL
Red Hat Enterprise Linux 7. <i>n</i> (for Intel64) (注)	RHEL7
Red Hat Enterprise Linux 8. <i>n</i> (for Intel64) (注)	RHEL8
Red Hat Enterprise Linux CoreOS	RHCOS
FUJITSU Hybrid IT Service FJcloud-O IaaS	IaaS
Red Hat Satellite	Satellite
Red Hat Update Infrastructure	RHUI

注) "*n*"にはマイナーリリースを示す数字が入ります。

## ■ 本文中の記号

本文中では、次の記号を使用します。

 <b>重要</b>	使用上の注意点や、してはいけないことを説明しています。必ずお読みください。
 <b>POINT</b>	理解を助けるための補足情報などを説明しています。必要に応じてお読みください。
『』	関連したマニュアルを示しています。

任意のアルファベットまたは数字を次のように表記します。

<i>A</i>	この文字列が表記されている箇所には、任意のアルファベットが表示されることを意味します。
<i>n</i>	この文字列が表記されている箇所には、0-99の数字が表示されることを意味します。

## ■ コマンド入力

本文中では、コマンド入力を次のように表記します。

- ユーザー可変(ユーザー環境により異なる)文字列

次のように<>で括って表記します。

```
# e2label <device 名> <ラベル名>
```

- 追加/変更対象文字列

次のように、太字で表記します。

```
NETWORKING=yes  
HOSTNAME=<ホスト名>  
...  
VLAN=yes
```

- 説明を有する文字列

次のように、文字列に下線を引き、枠外に説明文を記述します。

```
# mount LABEL=/backlabel /backup
```

下線部分はラベル名です。

- コマンドの実施ユーザー

次のように、コマンドプロンプトとして # と記述しているコマンドは、サーバーの管理者アカウントで実施します。

```
# ssh-keygen
```

一般的な管理者アカウントである root ユーザーへの切り替えは以下のコマンドで実施します。

```
$ sudo su -
```

## ■ 関連ドキュメント

目的・用途に合わせて、以下の関連ドキュメントもお読みください。

- Red Hat 社から公開されているドキュメント

RHOCP および RHEL に関して、多岐に渡るドキュメントが公開されていますので、構築時だけでなく、運用時にも参照する必要があります。

以下の URL より参照してください。

<https://access.redhat.com/documentation/>

ドキュメント名称	概要	本書での表記
OpenShift Container Platform 4.n Installing	RHOCP4 のインストールの概要・方法を説明しています。	『Installing』
OpenShift Container Platform 4.n Updating clusters	RHOCP4 のクラスタのアップデートを説明しています。	『Updating Clusters』
OpenShift Container Platform 4.n Building applications	RHOCP4 利用者(コンテナアプリケーション開発者)としてのアプリケーションの各種設定を説明しています。	『Building applications』
OpenShift Container Platform 4.n Images	RHOCP4 利用者としてのコンテナイメージの管理を説明しています。	『Images』
OpenShift Container Platform 4.n Nodes	RHOCP4 のクラスタの設定・管理方法を説明しています。	『Nodes』
OpenShift Container Platform 4.n Storage	RHOCP4 のストレージの設定を説明しています。	『Storage』
OpenShift Container Platform 4.n Registry	RHOCP4 のイメージレジストリーの設定を説明しています。	『Registry』
OpenShift Container Platform 4.n Backup and restore	RHOCP4 のバックアップ・リストアを説明しています。	『Backup and Restore』
OpenShift Container Platform 4.n Machine management	RHOCP4 のクラスタの追加・保守を説明しています。	『Machine management』
OpenShift Container Platform 4.n Service Mesh	RHOCP4 の Service Mesh のインストール・使用方法を説明しています。	『Service Mesh』
Red Hat Enterprise Linux 7 Networking Guide	RHEL7 のネットワークの設定・管理を説明しています。	『Networking Guide』

Red Hat Enterprise Linux 7 System Administrator's Guide	RHEL7 の導入・設定・管理を説明しています。	『System Administrator's Guide』
Red Hat Enterprise Linux 7 Storage Administration Guide	RHEL7 のストレージ管理を説明しています。	『Storage Administration Guide』
Red Hat Enterprise Linux 7 Logical Volume Manager Administration	RHEL7 の論理ボリューム管理を説明しています。	『Logical Volume Manager Administration』
Red Hat Enterprise Linux 8 Configuring basic system settings	RHEL8 の基本的なシステム設定を説明しています。	『Configuring basic system settings』

- SupportDesk-Web サイトで公開されているドキュメント  
RHEL および RHOCP に関するドキュメントが公開されています。  
以下の URL より参照してください。参照には、SupportDesk 契約が必要です。  
<https://eservice.fujitsu.com/supportdesk-web/>

ドキュメント名称	概要	本書での表記
Red Hat Enterprise Linux 7 ソフトウェア説明書	RHEL7 の概要と対象機種をご使用になるうえで確認していただきたい情報を説明しています。	『RHEL7 ソフトウェア説明書』
Red Hat Enterprise Linux 7 Linux ユーザーズマニュアル	RHEL7 の設計・導入・運用・保守に関する情報、および参考になる情報を説明しています。	『RHEL7 ユーザーズマニュアル』
Red Hat Enterprise Linux 8 ソフトウェア説明書	RHEL8 の概要と対象機種をご使用になるうえで確認していただきたい情報を説明しています。	『RHEL8 ソフトウェア説明書』
Red Hat Enterprise Linux 8 Linux ユーザーズマニュアル	RHEL8 の設計・導入・運用・保守に関する情報、および参考になる情報を説明しています。	『RHEL8 ユーザーズマニュアル』
Red Hat OpenShift Container Platform 4 ソフトウェア説明書	RHOCP4 の利用方法を説明しています。	『RHOCP4 ソフトウェア説明書』
Red Hat OpenShift Container Platform 4 制限事項／注意事項	RHOCP4 を利用するうえでの制限事項や注意事項を説明しています。	『RHOCP4 制限事項／注意事項』

- FUJITSU Hybrid IT Service FJcloud-O サイトで公開されている IaaS 関連のドキュメント IaaS に関するドキュメントが公開されています。

以下の URL より参照してください。

<https://doc.cloud.global.fujitsu.com/jp/iaas/index.html>

ドキュメント名称	概要	本書での表記
機能説明書	IaaS で提供されている機能を説明しています。	『IaaS 機能説明書』
IaaS ポータルユーザーズガイド	IaaS ポータルの操作方法を説明しています。	『IaaS ポータルユーザーズガイド』
制限事項・注意事項	IaaS の制限事項や注意事項を説明しています。	『IaaS 制限事項・注意事項』
FAQ	IaaS に関するよくある質問と回答をまとめています。	『IaaS FAQ』
初期構築ガイド	IaaS 利用時のシステム設計をする際に必要になるサービスの特徴や留意事項を説明しています。	『設計・構築ガイド』

IaaS で提供されている API に関するドキュメントが公開されています。

以下の URL より参照してください。

<https://doc.cloud.global.fujitsu.com/jp/iaas/api-reference.html>

ドキュメント名称	概要	本書での表記
API リファレンス	IaaS で提供されている API を説明しています。	『IaaS API リファレンス』

IaaS 上で利用する RHOCP に関するドキュメントが公開されています。

以下の URL より参照してください。

<https://doc.cloud.global.fujitsu.com/jp/iaas/rhocp.html>

ドキュメント名称	概要	本書での表記
IaaS RHOCP4 注意事項	IaaS 上で RHOCP4 を利用する上での注意事項を説明しています。	『IaaS RHOCP4 注意事項』

Red Hat Satellite 用設定情報	Satellite サーバーの設定情報を説明しています。	『Red Hat Satellite 用 設定情報』
IaaS RHOCP4 用設定情報	RHOCP4 のコンテナアプリケーションプラットフォームの構築に使用する設定情報を説明しています。	『IaaS RHOCP4 用 設定情報』

## ■ 輸出管理規制について

本書を輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規などの規制をご確認のうえ、必要な手続きをおとりください。

## ■ 商標

- Red Hat、Red Hat Enterprise Linux、RHEL、Red Hat Enterprise Linux CoreOS、RHCOS、Red Hat OpenShift Container Platform、OpenShift は、米国およびその他の国において登録された RedHat, Inc.の商標です。
- Linux®は米国及びその他の国における Linus Torvalds の登録商標です。
- Intel は、アメリカ合衆国および / またはその他の国における Intel Corporation またはその子会社の商標です。
- そのほか、本書に記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

なお、本書に掲載されているシステム名、製品名などには、必ずしも商標表示(™または®)を付記していません。

Copyright FUJITSU LIMITED 2020-2023

## 1. 概要

---

Red Hat OpenShift Container Platform(以降、RHOCP)の製品概要と FUJITSU Hybrid IT Service FJcloud-O IaaS(以降、IaaS)上に RHOCP を使用して構築するコンテナアプリケーションプラットフォームを説明します。

### 1.1 Red Hat OpenShift Container Platform とは

---

RHOCP は、Red Hat Enterprise Linux CoreOS(以降、RHCOS)、Red Hat Enterprise Linux(以降、RHEL)、および Kubernetes の技術に加え、エンタープライズ用途で必要になる認証、ビルド、アクセス制御などを強化したコンテナ運用管理製品です。

アプリケーション開発環境の構築やシステム管理の自動化により、ビジネスが求めるアプリケーション・サービスを迅速かつ安全に提供できます。

#### 1.1.1 主な機能

RHOCP が提供する主な機能には、以下のものがあります。

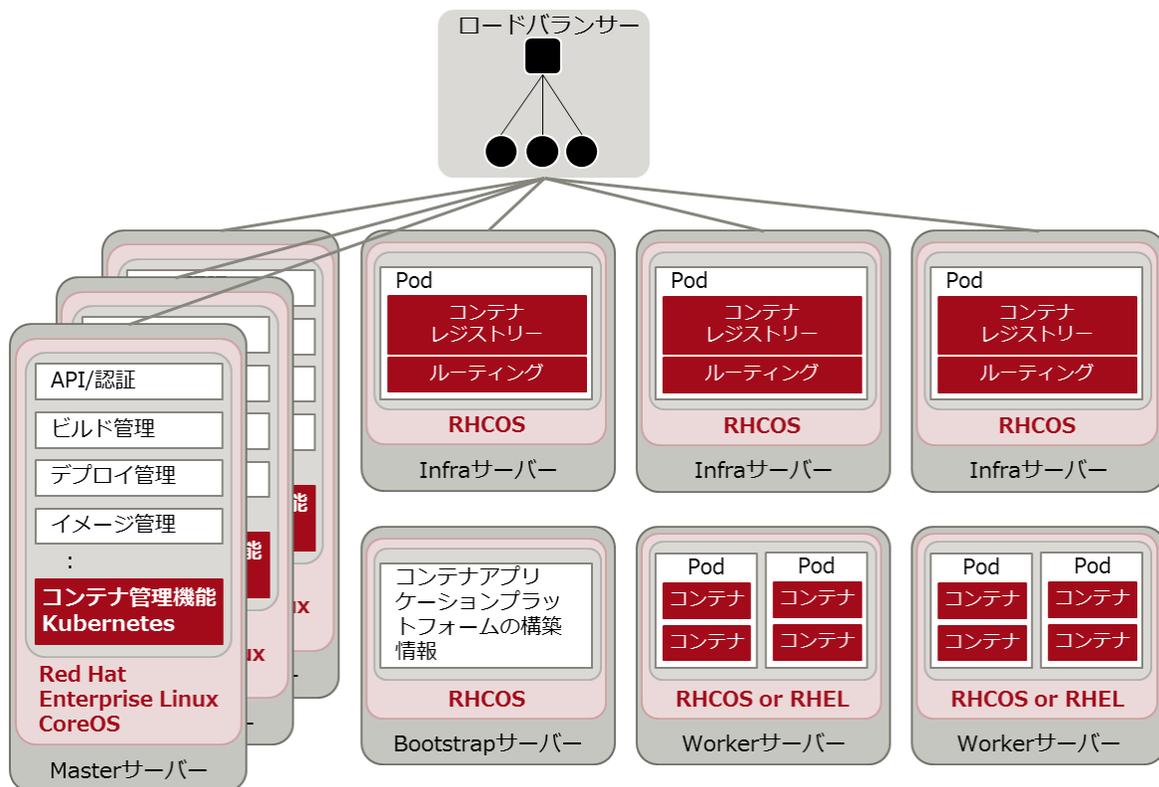
- コンテナオーケストレーション  
ユーザー管理、認証、Software Defined Networking(SDN)、コマンドラインツール、Web コンソールなどの機能を利用してコンテナの運用管理が可能です。
- 自動アプリケーション提供フロー  
アプリケーションのビルド、提供の自動化が可能です。
- マルチテナント  
テナントごとのリソース分離とアクセス制御が可能です。
- 自動スケーリング  
アプリケーションへの負荷の増減に対してリソース割り当ての自動化が可能です。

## 1.1.2 アーキテクチャー

コンテナアプリケーションプラットフォームは図 1-1 のように複数のサーバーで構成されます。これにより、業務の冗長化や負荷分散を実現します。各サーバーの役割は以下です。

- Master サーバー：Worker サーバーのコンテナ配備や状況監視などを管理します。
- Infra サーバー：Worker サーバーへのルーティングなどをします。
- Worker サーバー：1 つまたは複数のコンテナで構成される Pod と呼ばれる単位でコンテナを配備します。
- Bootstrap サーバー：コンテナアプリケーションプラットフォーム構築時に、一時的にコンテナアプリケーションプラットフォームのサーバーに接続し、構築に必要な情報を送信します。

図 1-1 RHOCP のアーキテクチャー



## 1.1.3 ライフサイクル

RHOCP4 ではマイナーバージョンごとにサポート期間が設定されています。コンテナアプリケーションプラットフォームがサポート期間外にならないように、アップデートを計画してください。RHOCP のライフサイクルは、Red Hat 社が公開している以下を参照してください。

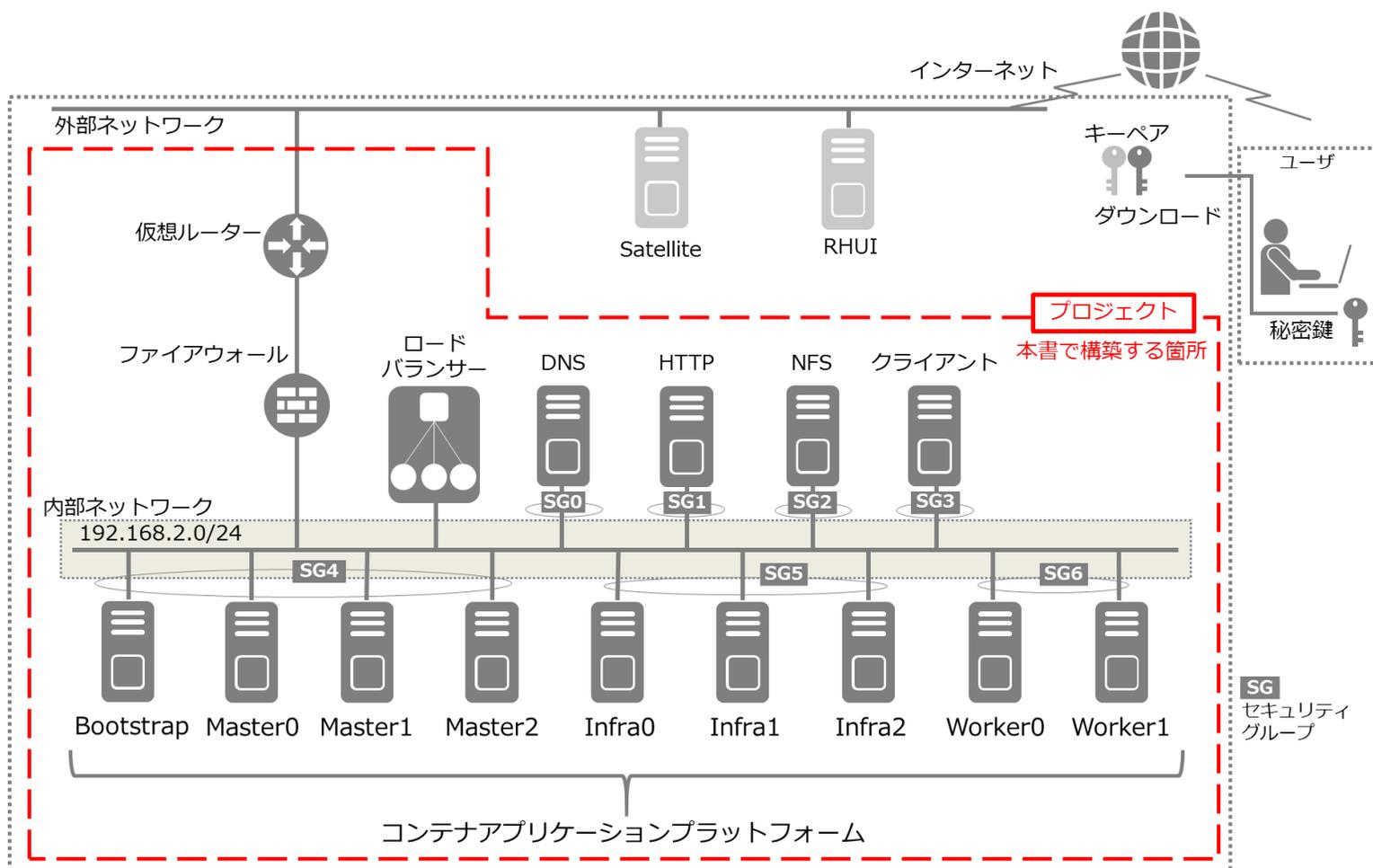
「Red Hat OpenShift Container Platform Life Cycle Policy」

<https://access.redhat.com/support/policy/updates/openshift>

## 1.2 想定システム構成

本書では図 1-2 のシステムを構築することを例に説明します。お客様が構築する箇所は赤い破線内の IaaS で作成する「プロジェクト」です。

図 1-2 システム構成例



本書で構築するシステムの詳細を以下に示します。この順番で作成してください。

### 1. 内部ネットワーク

仮想ネットワークを作成してください。本書では、サブネットの仮想ネットワークアドレスとして 192.168.2.0/24 を、DNS サーバーとして 192.168.2.51 の IP アドレスを設定します。

### 2. 仮想ルーター

仮想ルーターを作成してください。外部仮想ネットワークとして fip-net を設定してください。インターフェースに内部ネットワークのサブネットを設定してください。

### 3. ファイアウォール

表 1-1 のポート番号の通信を許可するファイアウォールルールを作成してください。

ファイアウォールポリシーを作成し、ファイアウォールルールを設定してください。ルールの順番は問いません。

ファイアウォールを作成し、ファイアウォールポリシーと仮想ルーターを設定してください。

表 1-1 ファイアウォールの設定

ポート番号	プロトコル	用途
22	TCP	SSH 用
80	TCP	HTTP トラフィック用
443	TCP	HTTPS トラフィック用

## 4. ロードバランサー

本書では、ロードバランサーのプライベート IP アドレスは、192.168.2.53 を使用します。

ロードバランサーの受信ポート番号とその負荷分散先のサーバーおよびポート番号の対応は表 1-2 を参照してください。

表 1-2 ロードバランサーが使用するポート番号

ロードバランサー受信ポート番号		負荷分散先 サーバー	サーバーポート番号	
ポート番号	プロトコル		ポート番号	プロトコル
6443	TCP	Master0	6443	TCP
		Master1		
		Master2		
		Bootstrap		
22623	TCP	Master0	22623	TCP
		Master1		
		Master2		
		Bootstrap		
80	TCP	Infra0	80	TCP
		Infra1		
		Infra2		
443	TCP	Infra0	443	TCP
		Infra1		
		Infra2		

## 5. 動作環境用のサーバー

コンテナアプリケーションプラットフォームが使用する動作環境用のサーバーの詳細は表 1-3 を参照してください。動作環境用のサーバーの作成に使用するイメージは、表 1-4 を参照してください。セキュリティグループは、表 1-5 を参照して設定してください。各サーバーの作成および設定方法は、「2. 動作環境の作成」を参照してください。

表 1-3 動作環境用のサーバーの詳細

サーバー	IP アドレス	CPU(vCPU)	メモリー	ディスク
DNS	192.168.2.51	2	8GB	50GB
HTTP	192.168.2.52	2	8GB	50GB
NFS	192.168.2.55	2	8GB	120GB
クライアント	192.168.2.54	2	8GB	50GB

表 1-4 動作環境用のサーバーが使用するイメージ

サーバー	イメージ
DNS	Red Hat Enterprise Linux 7.n 64bit (English) 01
HTTP	Red Hat Enterprise Linux 7.n 64bit (English) 01
NFS	Red Hat Enterprise Linux 7.n 64bit (English) 01
クライアント(注 1)	Red Hat Enterprise Linux 7.n 64bit (English) 01
	Red Hat Enterprise Linux 8.n 64bit (English) 01

注 1) RHOC4.10 以降では、クライアントサーバーは RHEL8 を使用してください。

表 1-5 動作環境用のサーバーが使用するポート番号

セキュリティグループ	サーバー	ポート	プロトコル	用途
SG0	DNS	22	TCP	SSH 接続用
		53	UDP	DNS 用
SG1	HTTP	22	TCP	SSH 接続用
		80	TCP	HTTP 用
SG2	NFS	22	TCP	SSH 接続用
		(注 1)	(注 1)	NFS 用
SG3	クライアント	22	TCP	SSH 接続用

注 1) 『Storage Administration Guide』の「8. Network File System (NFS)」を参照してください。

## 6. コンテナアプリケーションプラットフォームのサーバー

コンテナアプリケーションプラットフォームを構成するサーバーの詳細は表 1-6 を参照してください。サーバーの作成に使用するイメージは、表 1-7 を参照してください。Worker サーバーには、RHCOS もしくは RHEL を使用できます。サーバーのアップデートを含め、運用を容易にしたい場合は RHCOS を、OS レベルの詳細な設定を実施したい場合は RHEL を使用してください。RHEL のメジャーバージョンは RHEL7 もしくは RHEL8 です。セキュリティグループは表 1-8 を参照して設定してください。サーバーの作成および設定方法は、Worker サーバーに RHCOS を使用する場合は「3. コンテナアプリケーションプラットフォームの構築 (RHCOS)」を、RHEL を使用する場合は「4. コンテナアプリケーションプラットフォームの構築 (RHEL)」を参照してください。



RHOCP4.8/4.9 では、RHEL7 を使用した Worker サーバーの利用は非推奨です。また、RHOCP4.10/4.11/4.12 では、RHEL7 を使用した Worker サーバーの利用はサポート対象外ですので、Worker サーバーには、RHCOS または RHEL8 を利用してください。



RHOCP4.9/4.10/4.11/4.12 でのみ、Worker サーバーに RHEL8 を使用できます。RHOCP4.8 では、Worker サーバーに RHEL8 を使用できません。

表 1-6 コンテナアプリケーションプラットフォームを構成するサーバーの詳細

サーバー	IP アドレス	CPU(vCPU)	メモリー	ディスク
Bootstrap	192.168.2.50	4	16GB	120GB
Master0	192.168.2.10	4	16GB	120GB
Master1	192.168.2.11			
Master2	192.168.2.12			
Infra0	192.168.2.20	2	8GB	120GB
Infra1	192.168.2.21			
Infra2	192.168.2.22			
Worker0	192.168.2.30	2	8GB	120GB
Worker1	192.168.2.31			

表 1-7 コンテナアプリケーションプラットフォームを構成するサーバーが使用するイメージ

サーバー	イメージ	
Bootstrap	RedHat OpenShift (Master) 4.n 64bit (English) 01	
Master0 Master1 Master2	RedHat OpenShift (Master) 4.n 64bit (English) 01	
Infra0 Infra1 Infra2	RedHat OpenShift (Master) 4.n 64bit (English) 01	
Worker0 Worker1	RHCOS を使用する場合	RedHat OpenShift (Worker, RHCOS) 4.n 64bit (English) 01
	RHEL7 を使用する場合 (非推奨)	RedHat OpenShift (Worker, RHEL) 4.5 64bit (English) 01 (注 1)
	RHEL8 を使用する場合	RedHat OpenShift (Worker, RHEL8) 4.9 64bit (English) 01 (注 2)

注 1) RHEL7 は RHOCP4.8/4.9 で使用でき、共通のイメージを使用します。

注 2) RHEL8 は RHOCP4.9/4.10/4.11/4.12 で使用でき、共通のイメージを使用します。

表 1-8 コンテナアプリケーションプラットフォームを構成するサーバーが使用するポート番号

セキュリティグループ	サーバー	ポート	プロトコル	用途
SG4	Bootstrap Master0 Master1 Master2	-	ICMP	疎通確認用
		22	TCP	SSH 用
		1936	TCP	メトリクス用
		2379~2380	TCP	etcd サービス用
		4789	UDP	RHOCP 内部通信用
		6081	UDP	RHOCP 内部通信用
		6443	TCP	API 通信用
		9000~9999	TCP、UDP	サーバー情報通信用
		10249~10259	TCP	Kubernetes 予約ポート用
		22623	TCP	追加サーバー用
SG5	Infra0 Infra1	-	ICMP	疎通確認用
		22	TCP	SSH 用

	Infra2	80	TCP	HTTP トラフィック用
		443	TCP	HTTPS トラフィック用
		1936	TCP	メトリクス用
		2379~2380	TCP	etcd サービス用
		4789	UDP	RHOCP 内部通信用
		6081	UDP	RHOCP 内部通信用
		6443	TCP	API 通信用
		9000~9999	TCP、UDP	サーバー情報通信用
		10249~10259	TCP	Kubernetes 予約ポート用
SG6	Worker0 Worker1	-	ICMP	疎通確認用
		22	TCP	SSH 用
		1936	TCP	メトリクス用
		2379~2380	TCP	etcd サービス用
		4789	UDP	RHOCP 内部通信用
		6081	UDP	RHOCP 内部通信用
		6443	TCP	API 通信用
		9000~9999	TCP、UDP	サーバー情報通信用
		10249~10259	TCP	Kubernetes 予約ポート用

## 2. 動作環境の作成

---

本章ではコンテナアプリケーションプラットフォームを構築するうえで事前に必要な動作環境を作成します。

### 2.1 仮想サーバーの作成

---

DNS サーバー、HTTP サーバー、NFS サーバーおよびクライアントサーバーの 4 台のサーバーを『IaaS ポータル ユーザーズガイド』の「仮想サーバ」と『IaaS API リファレンス』の「コンピューター」を参照して作成してください。

### 2.2 DNS サーバーの設定

---

表 2-1 の A レコードを登録し、内部ネットワークに接続するすべてのサーバーおよびコンテナアプリケーションプラットフォーム上のサービスが名前解決できるように DNS サーバーを設定してください。設定方法は、『Networking Guide』の「15. DNS Servers」を参照してください。表 2-1 のサーバーは、登録する A レコードに対応するサーバーの名前を参考として示しています。

表 2-1 A レコード

ホスト名	IP アドレス	サーバー(参考)
ocp4-master0.test.example.com	192.168.2.10	Master0
ocp4-master1.test.example.com	192.168.2.11	Master1
ocp4-master2.test.example.com	192.168.2.12	Master2
ocp4-infra0.test.example.com	192.168.2.20	Infra0
ocp4-infra1.test.example.com	192.168.2.21	Infra1
ocp4-infra2.test.example.com	192.168.2.22	Infra2
ocp4-worker0.test.example.com	192.168.2.30	Worker0
ocp4-worker1.test.example.com	192.168.2.31	Worker1
api.test.example.com	192.168.2.53	-
api-int.test.example.com		
*.apps.test.example.com		
ocp4-bstrap.test.example.com	192.168.2.50	Bootstrap
ocp4-dns.test.example.com	192.168.2.51	DNS
ocp4-http.test.example.com	192.168.2.52	HTTP
ocp4-nfs.test.example.com	192.168.2.55	NFS
ocp4-cli.test.example.com	192.168.2.54	クライアント

## 2.3 HTTP サーバーの設定

HTTP サーバーは「2.5.2 サーバー設定ファイルの作成」で作成するサーバー設定ファイルを配布します。内部ネットワークに接続するすべてのサーバーが HTTP プロトコルでサーバー設定ファイルを取得できるように設定してください。設定方法は、『System Administrator's Guide』の「14. Web Servers」を参照してください。

## 2.4 NFS サーバーの設定

ユーザー所有者、グループ所有者を `nfsnobody` にした保存用ディレクトリーを作成してください。本書では、保存用ディレクトリーは `/exports/image-registry` を作成します。エクスポート設定では、`rw`、`sync`、`no_wdelay`、`all_squash`、`insecure`、`fsid=0` を付与し、内部ネットワークに接続するすべてのサーバーがアクセスできるようにディレクトリーを設定してください。設定方法は、『Storage Administration Guide』の「8. Network File System (NFS)」を参照してください。

## 2.5 クライアントサーバーの設定

コンテナアプリケーションプラットフォームを構築および操作するクライアントサーバーを設定します。

### 2.5.1 クライアントツールの設定

以下のコマンドを実施して RHOCP4 の最新のインストールツールと操作ツールを取得してください。バージョンにはインストールする RHOCP4 のバージョンを `4.y.z` 形式で指定してください。y には、インストールする RHOCP4 のマイナーバージョンを指定してください。z には、マイナーバージョンの最新のリリースを指定してください。

[RHOCP4.8 の場合]

最新のバージョンは、以下の URL から確認してください。

<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.8/>

```
$ curl -O https://mirror.openshift.com/pub/openshift-
v4/clients/ocp/latest-4.8/openshift-install-linux-<バージョン>.tar.gz

$ curl -O https://mirror.openshift.com/pub/openshift-
v4/clients/ocp/latest-4.8/openshift-client-linux-<バージョン>.tar.gz
```

## [RHOC4.9 の場合]

最新のバージョンは、以下の URL から確認してください。

<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.9/>

```
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.9/openshift-install-linux-<バージョン>.tar.gz  
  
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.9/openshift-client-linux-<バージョン>.tar.gz
```

## [RHOC4.10 の場合]

最新のバージョンは、以下の URL から確認してください。

<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.10/>

```
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.10/openshift-install-linux-<バージョン>.tar.gz  
  
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.10/openshift-client-linux-<バージョン>.tar.gz
```

## [RHOC4.11 の場合]

最新のバージョンは、以下の URL から確認してください。

<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.11/>

```
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.11/openshift-install-linux-<バージョン>.tar.gz  
  
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.11/openshift-client-linux-<バージョン>.tar.gz
```

## [RHOC4.12 の場合]

最新のバージョンは、以下の URL から確認してください。

<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.12/>

```
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.12/openshift-install-linux-<バージョン>.tar.gz  
  
$ curl -O https://mirror.openshift.com/pub/openshift-  
v4/clients/ocp/latest-4.12/openshift-client-linux-<バージョン>.tar.gz
```

以下のコマンドを実施し、取得したファイルを展開してください。

```
$ tar zxvf openshift-install-linux-<バージョン>.tar.gz
$ tar zxvf openshift-client-linux-<バージョン>.tar.gz
```

サーバーの管理者アカウントに切り替え後、以下のコマンドを実施し、実行ファイルをコピーしてください。

```
# cp openshift-install oc /usr/local/bin/
```

各ツールが設定されていることを確認します。サーバーの管理者アカウントから一般ユーザーのアカウントに切り替え後、以下のコマンドを実施し、各ツールが設定されていることを確認してください。

[インストールツール]

```
$ openshift-install version
openshift-install <バージョン>
(省略)
```

下線部分は、インストールする RHOCP4 のバージョンが表示されます。

[操作ツール]

```
$ oc version
Client Version: <バージョンの情報>
```

下線部分は、インストールする RHOCP4 のマイナーバージョンを含む任意の文字列が表示されます。

## 2.5.2 サーバー設定ファイルの作成

コンテナアプリケーションプラットフォームの構築に必要なサーバー設定ファイルを作成します。サーバー設定ファイルを作成する前に以下を準備してください。

- Red Hat 社イメージレジストリー用認証鍵  
『IaaS RHOC4 用設定情報』を参照してください。参照には契約者情報が必要です。
- キーペア  
[RHEL7 の場合]  
『System Administrator's Guide』の「12.2.4.1. Generating Key Pairs」を参照して作成してください。  
[RHEL8 の場合]  
『Configuring basic system settings』の「34.4. Generating SSH Key Pairs」を参照して作成してください。

以下の設定ファイルを任意のディレクトリーに作成してください。キーペアの公開鍵情報は作成したキーペアの公開鍵ファイルの内容をそのまま記載してください。

[install-config.yaml]

```
apiVersion: v1
baseDomain: example.com
compute:
- hyperthreading: Disabled
  name: worker
  replicas: 0
controlPlane:
  hyperthreading: Disabled
  name: master
  replicas: 3
metadata:
  name: test
networking:
  clusterNetworks:
  - cidr: <Pod ネットワークのネットワークアドレス>
    hostPrefix: 23
  networkType: <Pod ネットワークのネットワークインターフェースプラグイン>
  serviceNetwork:
```

- <サービスネットワークのネットワークアドレス>

```
platform:
  none: {}
fips: false
pullSecret: '<Red Hat 社イメージレジストリー用認証鍵>'
sshKey: '<キーペアの公開鍵>'
```

Pod ネットワークは、Pod 間の通信で使用するネットワークです。本書では、Pod ネットワークのネットワークアドレスとして、10.128.0.0/14 を設定します。

ネットワークインターフェースプラグインは、Pod に使用するネットワークプラグインの種類を指定してください。RHOCP4 では、2 つのネットワークプラグインがあります。OpenShift SDN を使用する場合は OpenShiftSDN を、OVN-Kubernetes を使用する場合は、OVNKubernetes を設定してください。なお、RHOCP4.12 で、デフォルトのネットワークプラグインが OVN-Kubernetes に変更されました。将来、OpenShift SDN は廃止される可能性があるため、OVN-Kubernetes の使用を推奨します。

サービスネットワークは、Pod への負荷分散をするサービスが使用するネットワークです。本書では、サービスネットワークのネットワークアドレスとして、172.30.0.0/16 を設定します。

既存のネットワークアドレスと重複しない、任意のネットワークアドレスを IP アドレス/プレフィックス形式で設定してください。これらのネットワークは、コンテナアプリケーションプラットフォームの構築中に自動で作成されるので、事前に作成する必要はありません。

以下のコマンドを実施し、作業ディレクトリーを作成し、設定ファイルを作業ディレクトリーにコピーしてください。

```
$ mkdir <作業ディレクトリー>
$ cp install-config.yaml <作業ディレクトリー>
```

以下のコマンドを実施し、中間ファイルを作成してください。

```
$ openshift-install create manifests --dir=<作業ディレクトリー>
```

Master サーバーにアプリケーション Pod が配備されないようにします。作業ディレクトリー内の中間ファイルである manifests/cluster-scheduler-02-config.yml を開き、以下のとおり編集してください。

[変更前]

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: true
  policy:
    name: ""
status: {}
```

[変更後]

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: false
  policy:
    name: ""
status: {}
```

以下のコマンドを実施し、中間ファイルからサーバー設定ファイルと接続情報を作成してください。

```
$ openshift-install create ignition-configs --dir=<作業ディレクトリー>
```

作業ディレクトリーに表 2-2 に示すファイルが作成されます。

表 2-2 サーバー設定ファイルと接続情報

ファイル名	ディレクトリー	内容
bootstrap.ign	<作業用ディレクトリー>	Bootstrap サーバー設定情報
master.ign	<作業用ディレクトリー>	Master サーバー設定情報
worker.ign	<作業用ディレクトリー>	Infra/Worker サーバー設定情報
kubeconfig	<作業用ディレクトリー>/auth	接続情報

bootstrap.ign、master.ign および worker.ign ファイルを以下の URL で取得できるように、『System Administrator's Guide』の「14. Web Servers」を参照して、HTTP サーバーに格納してください。

- <http://192.168.2.52/bootstrap.ign>
- <http://192.168.2.52/master.ign>
- <http://192.168.2.52/worker.ign>

kubeconfig ファイルは oc コマンド実施時に必要です。oc コマンド実施前に以下のコマンドを実施し、KUBECONFIG 環境変数に kubeconfig ファイルの絶対パスを設定してください。

```
$ export KUBECONFIG=<作業用ディレクトリーの絶対パス>/auth/kubeconfig
```

## 3. コンテナアプリケーションプラットフォームの構築 (RHCOS)

本章では前章で作成した動作環境で、Worker サーバーに RHCOS を使用してコンテナアプリケーションプラットフォームを構築します。本章で示すコマンドの実施およびファイルの作成はすべてクライアントサーバーで実施してください。

Worker サーバーに RHEL を使用してコンテナアプリケーションプラットフォームを構築する場合は、「4. コンテナアプリケーションプラットフォームの構築 (RHEL)」を参照してください。

### 3.1 仮想サーバーの作成

コンテナアプリケーションプラットフォームのサーバーを『IaaS ポータル ユーザーズガイド』の「仮想サーバ」と『IaaS API リファレンス』の「コンピューター」を参照して作成してください。サーバーの作成時に、表 1-6 に記載の IP アドレスを指定してください。

各サーバーで指定するプロビジョニングスクリプトは以下のとおりです。

[Bootstrap サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source":
"http://192.168.2.52/bootstrap.ign"}]}}
}
```

[Master サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source": "http://192.168.2.52/master.ign"}]}}
}
```

[Infra、Worker サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source": "http://192.168.2.52/worker.ign"}]}
  }
}
```

証明書の確認と許可の手順を実施してください。

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を確認します。証明書が確認できるまで時間がかかりますので、間隔を空けて繰り返し実施してください。

```
$ oc get csr
NAME          AGE   SIGNERNAME
REQUESTOR
CONDITION
csr-9s2nh     30s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
```

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を許可してください。

1つのInfraサーバー、また1つのWorkerサーバーにつき、2つの証明書が作成されます。すべての証明書が作成されCONDITION列がPendingとなっている証明書がなくなるまで、証明書の確認と許可を繰り返し実施してください。

```
$ oc adm certificate approve <証明書>
```

下線部分は、oc get csr で確認した証明書の NAME 列です。

サーバーの作成が完了していることを確認します。以下のコマンドを実施し、すべてのサーバーが Ready になっていることを確認してください。サーバーの作成には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get nodes
```

NAME VERSION	<u>STATUS</u>	<u>ROLES</u>	AGE
ocp4-infra0.test.example.com v1.14.6+c07e432da	Ready	worker	19m
ocp4-infra1.test.example.com v1.14.6+c07e432da	Ready	worker	19m
ocp4-infra2.test.example.com v1.14.6+c07e432da	Ready	worker	19m
ocp4-master0.test.example.com v1.14.6+c07e432da	Ready	master	19m
ocp4-master1.test.example.com v1.14.6+c07e432da	Ready	master	19m
ocp4-master2.test.example.com v1.14.6+c07e432da	Ready	master	19m
ocp4-worker0.test.example.com v1.14.6+c07e432da	Ready	worker	19m
ocp4-worker1.test.example.com v1.14.6+c07e432da	Ready	worker	19m

STATUS 列が、すべてのサーバーで Ready になります。

ROLES 列は、RHOCP4.12 の Master サーバーの場合、control-plane,master が表示されます。

## 3.2 イメージレジストリーの設定

イメージレジストリーの保存先ストレージを設定します。

ここでは保存先に NFS サーバーを設定します。以下のコマンドを実施し、image-registry の機能が存在することを確認してください。image-registry の機能は、コンテナアプリケーションプラットフォームの構築中に自動的に作成されます。image-registry の機能の作成には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get clusteroperator
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
(省略)					
<u>image-registry</u>		<u>False</u>	False	<u>True</u>	5m15s
(省略)					

NAME 列に image-registry があることを確認します。

AVAILABLE 列と DEGRADED 列はバージョンにより値が異なりますが、問題ありません。

イメージレジストリーの保存先ストレージを設定します。以下のとおり、保存先ストレージの設定ファイルを任意のディレクトリーに作成してください。

[image-registry-pv.yaml]

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: image-registry-pv
spec:
  capacity:
    storage: 100Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  nfs:
    server: 192.168.2.55
    path: /exports/image-registry
  claimRef:
    name: image-registry-storage
    namespace: openshift-image-registry
```

設定ファイルを作成した任意のディレクトリーに移動し、以下のコマンドを実施し、保存先ストレージの設定ファイルを適用してください。

```
$ oc apply -f image-registry-pv.yaml
```

以下のコマンドを実施し、イメージレジストリーの設定ファイルを編集してください。操作は vi 形式です。

```
$ oc edit configs.imageregistry.operator.openshift.io
```

以下のとおり、編集してください。claim の値が空白になっていますが、コンテナアプリケーションプラットフォームによって自動的に値が記載されますので問題ありません。

[変更前]

```
spec:
  defaultRoute: false
(省略)
  managementState: Removed
(省略)
  write:
    maxInQueue: 0
    maxRunning: 0
    maxWaitInQueue: 0s
  storage: {}
status:
```

[変更後]

```
spec:
  defaultRoute: false
(省略)
  managementState: Managed
(省略)
  write:
    maxInQueue: 0
    maxRunning: 0
    maxWaitInQueue: 0s
  storage:
    pvc:
      claim:
status:
```

## 3.3 Infra サーバーの設定

Infra サーバーの設定を変更します。

### 3.3.1 ラベル設定

Infra サーバーに対して、Infra ラベルを追加し、Worker ラベルを削除します。以下のコマンドをすべての Infra サーバーに対して実施してください。Infra サーバーのホスト名は表 2-1 を参照してください。

```
$ oc label node <Infra サーバーのホスト名> node-role.kubernetes.io/infra=""  
$ oc label node <Infra サーバーのホスト名> node-role.kubernetes.io/worker-
```

### 3.3.2 イメージレジストリーPod の配備先変更

イメージレジストリーPod が Infra サーバーに配備されるように設定します。以下のコマンドを実施し、設定ファイルを編集してください。操作は vi 形式です。

```
$ oc edit configs.imageregistry.operator.openshift.io
```

設定ファイルを以下のとおり、編集してください。

[変更前]

```
spec:  
  defaultRoute: false  
(省略)  
  write:  
    maxInQueue: 0  
    maxRunning: 0  
    maxWaitInQueue: 0s  
  storage:  
    pvc:  
      claim:  
status:
```

[変更後]

```
spec:
  defaultRoute: false
(省略)
  write:
    maxInQueue: 0
    maxRunning: 0
    maxWaitInQueue: 0s
  storage:
    pvc:
      claim:
nodeSelector:
  node-role.kubernetes.io/infra: ""
status:
```

イメージレジストリーPod が Infra サーバーに配備されていることを確認します。以下のコマンドを繰り返し実施し、image-registry が名前の先頭につく Pod が Infra サーバーに配備されていることを確認してください。イメージレジストリーPod の配備先の変更には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。NODE 列にイメージレジストリーPod が配備されている Infra サーバーのホスト名が表示されます。Infra サーバーのホスト名は表 2-1 を参照してください。

```
$ oc get pod -n openshift-image-registry -o wide
NAME                                     READY   STATUS
RESTARTS   AGE      IP             NODE
NOMINATED NODE   READINESS GATES
(省略)
image-registry-5d46d75cb4-8t8kn        1/1     Running
0           50s     10.131.0.14   ocp4-infra0.test.example.com
<none>           <none>
(省略)
```

下線部分が Infra サーバーのホスト名になります。

### 3.3.3 ルーターPod の配備先変更

ルーターPod がすべての Infra サーバーに配備されるように設定します。以下のコマンドを実施し、設定ファイルを編集してください。操作は vi 形式です。

```
$ oc edit ingresscontroller default -n openshift-ingress-operator
```

設定ファイルを以下のとおり、編集してください。

[変更前]

```
spec:
  replicas: 2
status:
```

[変更後]

```
spec:
  replicas: 3
  nodePlacement:
    nodeSelector:
      matchLabels:
        node-role.kubernetes.io/infra: ""
status:
```

ルーターPod が Infra サーバーに配備されていることを確認します。以下のコマンドを実施し、`openshift-ingress` プロジェクト内のすべてのルーターPod が Infra サーバーに配備されていることを確認してください。配備されるルーターPod の数は、設定ファイルに記載の `replicas` の値です。ルーターPod の配備先の変更には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。NODE 列にルーターPod が配備されている Infra サーバーのホスト名が表示されます。Infra サーバーのホスト名は表 2-1 を参照してください。

```
$ oc get pod -n openshift-ingress -o wide
NAME                                READY   STATUS    RESTARTS   AGE
IP                                  NODE                                NOMINATED NODE
READINESS GATES
router-default-5dd46df445-91c65    1/1     Running   0           30s
192.168.2.22    ocp4-infra2.test.example.com    <none>
<none>
router-default-5dd46df445-h88d6    1/1     Running   0           35s
192.168.2.21    ocp4-infra1.test.example.com    <none>
<none>
router-default-5dd46df445-jdsa6    1/1     Running   0           40s
192.168.2.20    ocp4-infra0.test.example.com    <none>
<none>
```

下線部分が Infra サーバーのホスト名になります。

### 3.3.4 モニタリング Pod の配備先変更

モニタリング Pod が Infra サーバーに配備されるように設定します。以下のとおり、設定ファイルを任意のディレクトリに作成してください。

[cluster-monitoring-configmap.yaml]

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |+
    alertmanagerMain:
      nodeSelector:
        node-role.kubernetes.io/infra: ""
    prometheusK8s:
```

```
nodeSelector:
  node-role.kubernetes.io/infra: ""
prometheusOperator:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
grafana:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
k8sPrometheusAdapter:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
kubeStateMetrics:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
telemeterClient:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
openshiftStateMetrics:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
thanosQuerier:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
```

設定ファイルを作成した任意のディレクトリーに移動し、以下のコマンドを実施し、設定ファイルを適用してください。

```
$ oc create -f cluster-monitoring-configmap.yaml
```

モニタリング Pod が Infra サーバーに配備されていることを確認します。以下のコマンドを実施し、alertmanager-main、prometheus-k8s、prometheus-operator、grafana、prometheus-adapter、kube-state-metrics、telemeter-client、openshift-state-metrics および、thanos-querier が名前の先頭につくモニタリング Pod が Infra サーバーに配備されていることを確認してください。モニタリング Pod の配備先の変更には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。NODE 列にモニタリング Pod が配備されている Infra サーバーのホスト名が表示されます。Infra サーバーのホスト名は表 2-1 を参照してください。

```
# oc get pod -n openshift-monitoring -o wide
```

NAME	READY	STATUS
alertmanager-main-0	6/6	Running
0	86s	10.128.2.17
<none>		<u>ocp4-infra0.test.example.com</u>
<none>		<none>
alertmanager-main-1	6/6	Running
0	119s	10.131.0.14
<none>		<u>ocp4-infra2.test.example.com</u>
<none>		<none>
cluster-monitoring-operator-56b769b58f-pz8xn	2/2	Running
0	84m	10.128.0.9
<none>		ocp4-master1.test.example.com
<none>		<none>
kube-state-metrics-64fd64448d-p2tcx	3/3	Running
0	2m3s	10.131.0.12
<none>		<u>ocp4-infra2.test.example.com</u>
<none>		<none>
node-exporter-4zv5h	2/2	Running
0	29m	192.168.2.10
<none>		ocp4-master0.test.example.com
<none>		<none>
node-exporter-8czmm	2/2	Running
0	29m	192.168.2.21
<none>		ocp4-infra1.test.example.com
<none>		<none>
node-exporter-8fjgz	2/2	Running
0	29m	192.168.2.22
<none>		ocp4-infra2.test.example.com
<none>		<none>
node-exporter-mfsdq	2/2	Running
0	29m	192.168.2.31
<none>		ocp4-worker1.test.example.com
<none>		<none>
node-exporter-t7tsk	2/2	Running
0	29m	192.168.2.12
<none>		ocp4-master2.test.example.com
<none>		<none>
node-exporter-vdx9d	2/2	Running
0	29m	192.168.2.30
<none>		ocp4-worker0.test.example.com
<none>		<none>
node-exporter-vxqp4	2/2	Running
0	29m	192.168.2.20
<none>		ocp4-infra0.test.example.com
<none>		<none>

node-exporter-wcs79				2/2	Running
0	29m	192.168.2.11	ocp4-master1.test.example.com		<none>
<none>					
openshift-state-metrics-6fbfb8bbfb-5tz6p				3/3	Running
0	2m3s	10.128.2.15	<u>ocp4-infra0.test.example.com</u>		<none>
<none>					
prometheus-adapter-68c8b9c55d-67bgl				1/1	Running
0	2m2s	10.128.2.16	<u>ocp4-infra0.test.example.com</u>		<none>
<none>					
prometheus-adapter-68c8b9c55d-jpkvs				1/1	Running
0	2m2s	10.129.2.13	<u>ocp4-infra1.test.example.com</u>		<none>
<none>					
prometheus-k8s-0				6/6	Running
0	47s	10.131.0.15	<u>ocp4-infra2.test.example.com</u>		<none>
<none>					
prometheus-k8s-1				6/6	Running
0	116s	10.129.2.14	<u>ocp4-infra1.test.example.com</u>		<none>
<none>					
prometheus-operator-58d6749947-2vdxv				2/2	Running
0	2m14s	10.128.2.14	<u>ocp4-infra0.test.example.com</u>		<none>
<none>					
prometheus-operator-admission-webhook-66b47499bd-89pgh				1/1	Running
0	2m24s	10.131.0.11	ocp4-infra2.test.example.com		<none>
<none>					
prometheus-operator-admission-webhook-66b47499bd-8xwfz				1/1	Running
0	2m24s	10.129.2.12	ocp4-infra1.test.example.com		<none>
<none>					
telemeter-client-dc47d9b59-kxmzb				3/3	Running
0	2m3s	10.131.0.13	<u>ocp4-infra2.test.example.com</u>		<none>
<none>					
thanos-querier-7bd8444c8c-47tfg				6/6	Running
0	28m	10.129.2.9	<u>ocp4-infra1.test.example.com</u>		<none>
<none>					
thanos-querier-7bd8444c8c-mp2q5				6/6	Running
0	28m	10.128.2.11	<u>ocp4-infra0.test.example.com</u>		<none>
<none>					

NAME 列に表示されるモニタリング Pod の種類や数は、バージョンにより異なりますが問題ありません。なお、alertmanager-main が名前の先頭につくモニタリング Pod は、RHOC4.8/4.9 では 3 つ配備されます。RHOC4.10/4.11/4.12 では 2 つ配備されます。

alertmanager-main、prometheus-k8s、prometheus-operator、grafana、prometheus-adapter、kube-state-metrics、telemeter-client、openshift-state-metrics および、thanos-querier が名前の先頭につくモニタリング Pod の Node 列は、Infra サーバーのホスト名になります。

### 3.3.5 アプリケーション Pod の配備先変更

アプリケーション Pod が Worker サーバーに配備されるよう設定します。以下のコマンドを実施し、設定ファイルを編集してください。操作は vi 形式です。コンテナアプリケーションプラットフォーム構築時にアプリケーション Pod は存在しないため、配備先の変更を確認する必要はありません。

```
$ oc edit scheduler cluster
```

設定ファイルを以下のとおり、編集してください。

[変更前]

```
spec:
  mastersSchedulable: false
  policy:
    name: ""
status: {}
```

[変更後]

```
spec:
  mastersSchedulable: false
  defaultNodeSelector: node-role.kubernetes.io/worker=
  policy:
    name: ""
status: {}
```

### 3.3.6 管理設定の追加

コンテナアプリケーションプラットフォーム内のサーバー管理の機能が、Infra サーバーを管理できるように、Infra サーバー用の管理設定を追加します。Infra サーバーの管理設定は、構築時は Worker サーバー用の管理設定が使用されるため、Infra サーバー用の管理設定を追加してください。

以下のとおり、設定ファイルを任意のディレクトリーに作成してください。

[infra-machine-config.yaml]

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfigPool
metadata:
  name: infra
spec:
  machineConfigSelector:
    matchExpressions:
      - key: machineconfiguration.openshift.io/role
        operator: In
        values: [worker,infra]
  nodeSelector:
    matchLabels:
      node-role.kubernetes.io/infra: ""
```

設定ファイルを作成した任意のディレクトリーに移動し、以下のコマンドを実施し、設定ファイルを適用してください。

```
$ oc create -f infra-machine-config.yaml
```

以下のコマンドを実施し、rendered-infra が名前の先頭につく管理設定が作成され、MACHINECOUNT 列と READYMACHINECOUNT 列に表示されている数字が同じになることを確認してください。設定の追加には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get machineconfigpool
```

NAME	CONFIG	UPDATED	
UPDATING	DEGRADED	MACHINECOUNT	READYMACHINECOUNT
UPDATEDMACHINECOUNT	DEGRADEDMACHINECOUNT		
infra	rendered-infra-24b94023f79cede7ab40c6bebe857263	True	
False	False	<u>3</u>	<u>3</u>
3	0		

(省略)

下線部分に表示される数字が同じ値になります。

## 3.4 コンテナアプリケーションプラットフォーム構築の完了確認

以下のコマンドを実施し、すべての機能の AVAILABLE 列が True になっていることを確認してください。すべての機能は、コンテナアプリケーションプラットフォームの構築中に自動的に作成されます。コンテナアプリケーションプラットフォームのすべての機能の作成には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get clusteroperator
```

NAME	PROGRESSING	DEGRADED	SINCE	MESSAGE	VERSION	AVAILABLE
authentication	False	False	33m		4.12.0	True
baremetal	False	False	63m		4.12.0	True
cloud-controller-manager	False	False	74m		4.12.0	True
cloud-credential	False	False	96m		4.12.0	True
cluster-autoscaler	False	False	63m		4.12.0	True
config-operator	False	False	65m		4.12.0	True
console	False	False	38m		4.12.0	True
control-plane-machine-set	False	False	64m		4.12.0	True

### 3.4 コンテナアプリケーションプラットフォーム構築の完了確認

csi-snapshot-controller			4.12.0	True
False	False	64m		
dns			4.12.0	True
False	False	63m		
etcd			4.12.0	True
False	False	56m		
image-registry			4.12.0	True
False	False	50m		
ingress			4.12.0	True
False	False	41m		
insights			4.12.0	True
False	False	58m		
kube-apiserver			4.12.0	True
False	False	54m		
kube-controller-manager			4.12.0	True
False	False	54m		
kube-scheduler			4.12.0	True
False	False	52m		
kube-storage-version-migrator			4.12.0	True
False	False	65m		
machine-api			4.12.0	True
False	False	63m		
machine-approver			4.12.0	True
False	False	64m		
machine-config			4.12.0	True
False	False	62m		
marketplace			4.12.0	True
False	False	63m		
monitoring			4.12.0	True
False	False	39m		
network			4.12.0	True
False	False	65m		
node-tuning			4.12.0	True
False	False	63m		
openshift-apiserver			4.12.0	True
False	False	53m		
openshift-controller-manager			4.12.0	True
False	False	55m		
openshift-samples			4.12.0	True
False	False	52m		
operator-lifecycle-manager			4.12.0	True
False	False	64m		
operator-lifecycle-manager-catalog			4.12.0	True
False	False	64m		
operator-lifecycle-manager-packageserver			4.12.0	True
False	False	53m		
service-ca			4.12.0	True
False	False	65m		

storage			4.12.0	True
False	False	65m		

NAME 列は、バージョンにより表示名と機能数が異なりますが、問題ありません。

AVAIABLE 列は、すべて True になります。

## 3.5 Bootstrap サーバーの削除

コンテナアプリケーションプラットフォームの構築完了後、Bootstrap サーバーは不要になります。

表 1-2 で設定したロードバランサーの負荷分散先のサーバーから、Bootstrap サーバーを除外してください。その後、『IaaS ポータル ユーザーズガイド』の「仮想サーバ」を参照して Bootstrap サーバーを削除してください。

## 4. コンテナアプリケーションプラットフォームの構築 (RHEL)

---

本章では 2 章で作成した動作環境で、Worker サーバーに RHEL を使用してコンテナアプリケーションプラットフォームを構築します。

Worker サーバーに RHCOS を使用してコンテナアプリケーションプラットフォームの構築をする場合は、「3. コンテナアプリケーションプラットフォームの構築 (RHCOS)」を参照してください。



RHOCP4.10 では、RHOCP4.10.23 以降のバージョンで、Worker サーバーに RHEL を使用したコンテナアプリケーションプラットフォームの構築は非推奨です。RHOCP4.10.23 以降のバージョンで RHEL を使用したコンテナアプリケーションプラットフォームを構築する場合は、RHOCP4.11 以降のバージョンで構築してください。

### 4.1 仮想サーバーの作成

---

コンテナアプリケーションプラットフォームのサーバーを『IaaS ポータル ユーザーズガイド』の「仮想サーバ」と『IaaS API リファレンス』の「コンピューティング」を参照して作成してください。サーバーの作成時に、表 1-6 に記載の IP アドレスを指定してください。

各サーバーで指定するプロビジョニングスクリプトは以下のとおりです。

[Bootstrap サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source":
"http://192.168.2.52/bootstrap.ign"}]}}
}
```

## [Master サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source": "http://192.168.2.52/master.ign"}]}
  }
}
```

## [Infra サーバーのプロビジョニングスクリプト]

```
{
  "ignition": {
    "version": "3.1.0",
    "config": {"merge": [{"source": "http://192.168.2.52/worker.ign"}]}
  }
}
```

## [Worker サーバーのプロビジョニングスクリプト]

```
#cloud-config

fqdn: <Worker サーバーのホスト名>
```

下線部分は起動する Worker サーバーのホスト名を指定してください。Worker サーバーのホスト名は表 2-1 を参照してください。

証明書の確認と許可の手順を実施してください。

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を確認します。証明書が確認できるまで時間がかかりますので、間隔を空けて繰り返し実施してください。

```
$ oc get csr
NAME          AGE    SIGNERNAME
REQUESTOR
CONDITION
csr-9s2nh     30s    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper
Pending
```

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を許可してください。

1 つの Infra サーバーにつき、2 つの証明書が作成されます。すべての証明書が作成され CONDITION 列が Pending となっている証明書がなくなるまで、証明書の確認と許可を繰り返し実施してください。

```
$ oc adm certificate approve <証明書>
```

下線部分は、oc get csr で確認した証明書の NAME 列です。

## 4.2 Worker サーバーの作成

Worker サーバーの設定をします。

本節は、Worker サーバーで実施してください。

### 4.2.1 SSH 公開鍵の登録

RHEL を使用したコンテナアプリケーションプラットフォームの構築には、クライアントサーバーで Ansible を実行します。Ansible とは構成管理ツールで、Ansible を実行することで inventory ファイルに記述した接続情報を使用して環境を構築することができます。Ansible を実行時、クライアントサーバーの一般ユーザーから Worker サーバーの一般ユーザーに SSH 接続をするため、「2.5.2 サーバー設定ファイルの作成」で作成したキーペアの SSH 公開鍵を、Worker サーバーに追加してください。追加方法は、Worker サーバーに RHEL7 を使用する場合は『System Administrator's Guide』の「12.2.4.1. Generating Key Pairs」を、RHEL8 を使用する場合は『Configuring basic system settings』の「34.4. Generating SSH key pairs」を参照してください。

### 4.2.2 rpm パッケージのインストール先の設定

Worker サーバーでは、Ansible の実行中に自動的に rpm パッケージがインストールされます。rpm パッケージのインストールには Satellite サーバーを使用するため、Satellite サーバーへの接続とリポジトリの有効化をしてください。

サーバーの管理者アカウントに切り替え後、以下のコマンドを実施して、Satellite サーバーに接続してください。コマンドで指定する Satellite サーバーの FQDN、組織名、およびアクティベーションキーは、『Red Hat Satellite 用設定情報』を参照してください。参照には契約者情報が必要です。

```
# rpm -Uvh http://<Satellite サーバーの FQDN>/pub/katello-ca-consumer-latest.noarch.rpm
# subscription-manager register --org=<組織名> --activationkey=<アクティベーションキー>
```

以下のコマンドを実施して、インストールする rpm パッケージが含まれているリポジトリを有効化してください。

[RHOCP4.8 の場合]

```
# subscription-manager repos \  
--enable="rhel-7-server-rpms" \  
--enable="rhel-7-fast-datapath-rpms" \  
--enable="rhel-7-server-extras-rpms" \  
--enable="rhel-7-server-optional-rpms" \  
--enable="rhel-7-server-ose-4.8-rpms"
```

[RHOCP4.9 の場合]

• RHEL7 の場合

```
# subscription-manager repos \  
--enable="rhel-7-server-rpms" \  
--enable="rhel-7-fast-datapath-rpms" \  
--enable="rhel-7-server-extras-rpms" \  
--enable="rhel-7-server-optional-rpms" \  
--enable="rhel-7-server-ose-4.9-rpms"
```

• RHEL8 の場合

```
# subscription-manager repos \  
--enable="rhel-8-for-x86_64-baseos-rpms" \  
--enable="rhel-8-for-x86_64-appstream-rpms" \  
--enable="rhocp-4.9-for-rhel-8-x86_64-rpms" \  
--enable="fast-datapath-for-rhel-8-x86_64-rpms"
```

[RHOCP4.10 の場合]

```
# subscription-manager repos \  
--enable="rhel-8-for-x86_64-baseos-rpms" \  
--enable="rhel-8-for-x86_64-appstream-rpms" \  
--enable="rhocp-4.10-for-rhel-8-x86_64-rpms" \  
--enable="fast-datapath-for-rhel-8-x86_64-rpms"
```

[RHOCP4.11 の場合]

```
# subscription-manager repos \  
--enable="rhel-8-for-x86_64-baseos-rpms" \  
--enable="rhel-8-for-x86_64-appstream-rpms" \  
--enable="rhocp-4.11-for-rhel-8-x86_64-rpms" \  
--enable="fast-datapath-for-rhel-8-x86_64-rpms"
```

[RHOCP4.12 の場合]

```
# subscription-manager repos \  
--enable="rhel-8-for-x86_64-baseos-rpms" \  
--enable="rhel-8-for-x86_64-appstream-rpms" \  
--enable="rhocp-4.12-for-rhel-8-x86_64-rpms" \  
--enable="fast-datapath-for-rhel-8-x86_64-rpms"
```

### 4.2.3 rpm パッケージのアップデート

以下のコマンドを実施して、Worker サーバーにインストールされている rpm パッケージをアップデートしてください。

[Worker サーバーに RHEL7 を使用する場合]

```
# yum update
```

[Worker サーバーに RHEL8 を使用する場合]

```
# subscription-manager release --set=8.7  
# rm -rf /var/cache/dnf  
# yum update  
# subscription-manager release --unset
```

## 4.3 クライアントサーバーの設定

---

Ansible を実行するためのクライアントサーバーの設定をします。  
本節は、クライアントサーバーで実施してください。

### 4.3.1 rpm パッケージのインストール

Ansible の実行に必要な rpm パッケージをインストールします。Satellite サーバーへの接続後、Satellite サーバーから rpm パッケージをインストールしてください。

サーバーの管理者アカウントに切り替え後、以下のコマンドを実施して、Satellite サーバーに接続してください。コマンドで指定する Satellite サーバーの FQDN、組織名、およびアクティベーションキーは、『Red Hat Satellite 用設定情報』を参照してください。参照には契約者情報が必要です。

```
# rpm -Uvh http://<Satellite サーバーの FQDN>/pub/katello-ca-consumer-  
latest.noarch.rpm  
  
# subscription-manager register --org=<組織名> --activationkey=<アクティベ  
ーションキー>
```

以下のコマンドを実施して、インストールする rpm パッケージのリポジトリを有効化してください。

[RHOCP4.8/4.9/4.10 の場合]

```
# subscription-manager repos \
--enable="rhel-7-server-rpms" \
--enable="rhel-7-server-extras-rpms" \
--enable="rhel-7-server-ose-4.y-rpms" \
--enable="rhel-7-server-ansible-2.9-rpms"
```

y には、インストールする RHOCP4 のマイナーバージョンを指定してください。

[RHOCP4.11/4.12 の場合]

```
# subscription-manager repos \
--enable="rhel-8-for-x86_64-baseos-rpms" \
--enable="rhel-8-for-x86_64-appstream-rpms" \
--enable="rhocp-4.y-for-rhel-8-x86_64-rpms"
```

y には、インストールする RHOCP4 のマイナーバージョンを指定してください。

以下のコマンドを実施して、Ansible の実行に必要な rpm パッケージをインストールしてください。

```
# yum install openshift-ansible
```

## 4.3.2 Ansible の設定

Ansible の実行に使用する inventory ファイルを作成します。

以下のコマンドを実施して、inventory ファイルを編集してください。操作は vi 形式です。

```
# vi /etc/ansible/hosts
```

inventory ファイルを以下のとおり、編集してください。

```
[all:vars]
ansible_user=k5user
ansible_become=True
openshift_kubeconfig_path="<kubeconfig ファイルの絶対パス>"

[new_workers]
ocp4-worker0.test.example.com
ocp4-worker1.test.example.com
```

下線部分は、表 2-3 の kubeconfig のファイルの絶対パスです。

## 4.4 Ansible の実行

Ansible を実行して、コンテナアプリケーションプラットフォームを構築します。  
本節は、クライアントサーバーで実施してください。

サーバーの管理者アカウントから一般ユーザーのアカウントに切り替え後、以下のコマンドを実施し、コンテナアプリケーションプラットフォームの機能のうち、image-registry を除いたすべての機能の AVAILABLE 列が True になっていることを確認してください。すべての機能は自動的に作成されます。また、すべての機能の作成には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get clusteroperator
```

<u>NAME</u>				VERSION	<u>AVAILABLE</u>
PROGRESSING	DEGRADED	SINCE	MESSAGE		
authentication				4.12.0	True
False	False	33m			
baremetal				4.12.0	True
False	False	63m			
cloud-controller-manager				4.12.0	True
False	False	74m			
cloud-credential				4.12.0	True
False	False	96m			
cluster-autoscaler				4.12.0	True
False	False	63m			
config-operator				4.12.0	True
False	False	65m			
console				4.12.0	True
False	False	38m			
control-plane-machine-set				4.12.0	True
False	False	64m			
csi-snapshot-controller				4.12.0	True
False	False	64m			
dns				4.12.0	True
False	False	63m			
etcd				4.12.0	True
False	False	56m			
image-registry				4.12.0	True
False	False	50m			
ingress				4.12.0	True
False	False	41m			
insights				4.12.0	True
False	False	58m			

kube-apiserver			4.12.0	True
False	False	54m		
kube-controller-manager			4.12.0	True
False	False	54m		
kube-scheduler			4.12.0	True
False	False	52m		
kube-storage-version-migrator			4.12.0	True
False	False	65m		
machine-api			4.12.0	True
False	False	63m		
machine-approver			4.12.0	True
False	False	64m		
machine-config			4.12.0	True
False	False	62m		
marketplace			4.12.0	True
False	False	63m		
monitoring			4.12.0	True
False	False	39m		
network			4.12.0	True
False	False	65m		
node-tuning			4.12.0	True
False	False	63m		
openshift-apiserver			4.12.0	True
False	False	53m		
openshift-controller-manager			4.12.0	True
False	False	55m		
openshift-samples			4.12.0	True
False	False	52m		
operator-lifecycle-manager			4.12.0	True
False	False	64m		
operator-lifecycle-manager-catalog			4.12.0	True
False	False	64m		
operator-lifecycle-manager-packageserver			4.12.0	True
False	False	53m		
service-ca			4.12.0	True
False	False	65m		
storage			4.12.0	True
False	False	65m		

NAME 列は、バージョンにより表示名と機能数が異なりますが、問題ありません。

AVAILABLE 列は、image-registry を除いたすべての機能が True になります。

image-registry を除いたすべての機能の AVAILABLE 列が True となっていることを確認後、以下のコマンドを実施して、Ansible を実行してください。

```
$ cd /usr/share/ansible/openshift-ansible
$ ansible-playbook -i /etc/ansible/hosts playbooks/scaleup.yml
```

Ansible の実行結果が failed=0 となっていることを確認してください。なお、Ansible の実行が終了するまで、ネットワーク環境に依存しますが 15 分ほどかかります。

```
PLAY RECAP
*****
localhost                : ok=1    changed=1    unreachable=0
failed=0    skipped=3    rescued=0    ignored=0
ocp4-worker0.test.example.com : ok=38   changed=19   unreachable=0
failed=0    skipped=4    rescued=0    ignored=0
ocp4-worker1.test.example.com : ok=38   changed=19   unreachable=0
failed=0    skipped=4    rescued=0    ignored=0
```

下線部分が failed=0 になります。

## 4.5 サーバーの状態の確認

本節は、クライアントサーバーで実施してください。

以下のコマンドを実施して、コンテナアプリケーションプラットフォームの Worker サーバーの状態を確認してください。NAME 列に Worker サーバーのホスト名が表示されます。Worker サーバーのホスト名は表 2-1 を参照してください。Worker サーバーの STATUS 列が Ready と表示される場合、「4.6 イメージレジストリの設定」へ進んでください。Worker サーバーの STATUS 列が NotReady と表示される場合、「4.5.1 証明書の許可」を実施してください。

```
$ oc get nodes
NAME                                STATUS    ROLES    AGE
VERSION
(省略)
ocp4-worker0.test.example.com      Ready    worker   19m
v1.14.6+c07e432da
ocp4-worker1.test.example.com      Ready    worker   19m
v1.14.6+c07e432da
```

下線部分は、Ready もしくは NotReady が表示されます。

## 4.5.1 証明書許可

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を確認します。

```
$ oc get csr
NAME          AGE          REQUESTOR
CONDITION
csr-mjtsx    5m38s       system:node: ocp4-worker1.test.example.com
Pending
```

以下のコマンドを実施して、CONDITION 列が Pending になっている証明書を許可してください。

1つのWorkerサーバーにつき、2つの証明書が作成されます。すべての証明書が作成され CONDITION 列が Pending となっている証明書がなくなるまで、証明書の確認と許可を繰り返し実施してください。

```
$ oc adm certificate approve <証明書>
```

下線部分は、oc get csr で確認した証明書の NAME 列です。

すべての証明書を許可した後、以下のコマンドを実施して、Workerサーバーの STATUS 列が Ready となっていることを確認してください。

```
$ oc get nodes
NAME                                STATUS    ROLES    AGE
VERSION
(省略)
ocp4-worker0.test.example.com      Ready    worker   19m
v1.14.6+c07e432da
ocp4-worker1.test.example.com      Ready    worker   19m
v1.14.6+c07e432da
```

下線部分が、Ready になります。

## 4.6 イメージレジストリーの設定

イメージレジストリーの保存先ストレージを設定します。

本節は、クライアントサーバーで実施してください。

設定方法は、「3.2 イメージレジストリーの設定」を参照してください。

## 4.7 Infra サーバーの設定

---

Infra サーバーの設定を変更します。

本節は、クライアントサーバーで実施してください。

設定の変更は、「3.3 Infra サーバーの設定」を参照してください。

## 4.8 コンテナアプリケーションプラットフォーム構築の完了確認

---

コンテナアプリケーションプラットフォームの構築の完了確認をします。

本節は、クライアントサーバーで実施してください。

確認方法は、「3.4 コンテナアプリケーションプラットフォーム構築の完了確認」を参照してください。

## 4.9 Bootstrap サーバーの削除

---

Bootstrap サーバーを削除します。

削除の方法は、「3.5 Bootstrap サーバーの削除」を参照ください。

## 付録 A RHEL コンテナイメージへのパッケージの追加

---

アプリケーション Pod を開発する際に、RHEL のコンテナイメージに RHEL の rpm パッケージをインストールすることがあります。本付録ではそのインストール手順を示します。本付録で示すコマンドの実施およびファイルの作成はすべてクライアントサーバーで実施してください。

### A.1 Satellite サーバーの情報の取得

---

クライアントサーバーで、RHUI を利用して subscription-manager の rpm パッケージをインストールしてください。RHUI の利用方法は、『IaaS 機能説明書』を参照してください。

サーバーの管理者アカウントに切り替え後、以下のコマンドを実施し、Satellite サーバーに接続してください。コマンドで指定する Satellite サーバーの FQDN、組織名、およびアクティベーションキーは、『Red Hat Satellite 用設定情報』を参照してください。参照には契約者情報が必要です。

```
# rpm -Uvh http://<Satellite サーバーの FQDN>/pub/katello-ca-consumer-  
latest.noarch.rpm  
  
# subscription-manager register --org=<組織名> --activationkey=<アクティベ  
ションキー>
```

以下のコマンドを実施し、クライアント鍵とクライアント証明書ファイル名を確認してください。

```
# ls /etc/pki/entitlement
```

それぞれ以下のようなファイル名です。

- ・クライアント鍵: <数列>-key.pem
- ・クライアント証明書: <数列>.pem

[RHEL7 の rpm パッケージをインストールする場合]

以下のコマンドを実施し、リポジトリ URL を取得してください。

```
# subscription-manager repos
```

以下のような情報が出力されます。Repo URL に記載されている URL がリポジトリ URL です。

```
+-----+
  Available Repositories in /etc/yum.repos.d/redhat.repo
+-----+
Repo ID:    rhel-7-server-rpms
Repo Name:  Red Hat Enterprise Linux 7 Server (RPMs)
Repo URL:   <リポジトリURL>
Enabled:    1
```

コンテナイメージで使用するリポジトリファイルを任意のディレクトリに以下のとおり、作成してください。

[satellite.repo]

```
[rhel-7-server-rpms]
name = Red Hat Enterprise Linux 7 Server (RPMs)
baseurl = <リポジトリURL>
gpgcheck = 0
enabled = 1
sslverify = 0
sslclientkey = /etc/pki/entitlement/<クライアント鍵>
sslclientcert = /etc/pki/entitlement/<クライアント証明書>
```

[RHEL8 の rpm パッケージをインストールする場合]

コンテナイメージで使用するリポジトリファイルを任意のディレクトリーに以下のとおり、作成してください。

[satellite.repo]

```
[rhel-8-for-x86_64-appstream-rpms]
name = Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
baseurl = https://rh-satellite.jp-west-2.cloud.global.fujitsu.com/pulp/repos/FUJITSU_Cloud_Service/Library/content/dist/rhel8/8/x86_64/appstream/os
gpgcheck = 0
enabled = 1
sslverify = 0
sslclientkey = /etc/pki/entitlement/<クライアント鍵>
sslclientcert = /etc/pki/entitlement/<クライアント証明書>

[rhel-8-for-x86_64-baseos-rpms]
name = Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
baseurl = https://rh-satellite.jp-west-2.cloud.global.fujitsu.com/pulp/repos/FUJITSU_Cloud_Service/Library/content/dist/rhel8/8/x86_64/baseos/os
gpgcheck = 0
enabled = 1
sslverify = 0
sslclientkey = /etc/pki/entitlement/<クライアント鍵>
sslclientcert = /etc/pki/entitlement/<クライアント証明書>
```

## A.2 Satellite サーバーを利用したパッケージの取得

サーバーの管理者アカウントから一般ユーザーのアカウントに切り替え後、以下のコマンドを実施し、プロジェクトを作成してください。コマンド実施後は、作成したプロジェクトでコマンドが実施されるため、プロジェクトを変更する操作は必要ありません。

```
$ oc new-project <プロジェクト名>
```

下線部分は 63 文字以内で指定してください。使用できる文字は、英小文字、数字および"-"(ハイフン)で、半角文字のみです。ただし、"-"(ハイフン)は、先頭および末尾には使用できません。

サーバーの管理者アカウントに切り替え後、リポジトリファイルを作成した任意のディレクトリーに移動し、以下のコマンドを実施し、リポジトリファイルをプロジェクトに保存してください。

```
# oc create configmap yum-repos-d --from-file ./satellite.repo
```

以下のコマンドを実施し、クライアント鍵とクライアント証明書をプロジェクトに保存してください。

```
# oc create secret generic etc-pki-entitlement \  
--from-file /etc/pki/entitlement/<クライアント鍵> \  
--from-file /etc/pki/entitlement/<クライアント証明書>
```

サーバーの管理者アカウントから一般ユーザーのアカウントに切り替え後、以下のコマンドを実施し、ビルドコンフィグファイルを作成してください。

```
$ oc new-build --strategy docker --binary \  
--docker-image <RHEL のコンテナイメージ名> --name <ビルドコンフィグファイル名>
```

RHEL のコンテナイメージ名は、RHEL7 の rpm パッケージをインストールする場合は "registry.redhat.io/rhel7:latest" を、RHEL8 の rpm パッケージをインストールする場合は "registry.redhat.io/ubi8:latest" を指定してください。

ビルドコンフィグファイル名は 58 文字以内で指定してください。使用できる文字は、英小文字、数字および"-"(ハイフン)で、半角文字のみです。ただし、数字は、先頭には使用できません。また、"-"(ハイフン)は、先頭および末尾には使用できません。

以下のコマンドを実施し、作成したビルドコンフィグファイルを編集してください。操作は vi 形式です。

```
$ oc edit buildconfig <ビルドコンフィグファイル名>
```

以下のとおり、編集してください。

[変更前]

```
spec:
  (省略)
  source:
    binary: {}
    type: Binary
  strategy:
```

[変更後]

```
spec:
  (省略)
  source:
    binary: {}
    type: Binary
  secrets:
  - secret:
    name: etc-pki-entitlement
    destinationDir: etc-pki-entitlement
  configMaps:
  - configMap:
    name: yum-repos-d
    destinationDir: yum.repos.d
  strategy:
```

以下のとおり、Dockerfile ファイルを任意のディレクトリーに作成してください。

[Dockerfile]

```
FROM <RHEL のコンテナイメージ名>
USER root
RUN sed -i".org" -e "s/1/0/g" /etc/yum/pluginconf.d/subscription-
manager.conf
COPY ./etc-pki-entitlement /etc/pki/entitlement
COPY ./yum.repos.d /etc/yum.repos.d
RUN rm /etc/rhsm-host && \
    yum -y update && \
    yum -y install <パッケージ名> && \
    rm -rf /etc/pki/entitlement && \
    rm -f /etc/yum.repos.d/satellite.repo
USER 1001
ENTRYPOINT ["/bin/bash"]
```

[RHEL7 の rpm パッケージをインストールする場合]

RHEL のコンテナイメージ名は、“registry.redhat.io/rhel7:latest”を指定してください。

パッケージ名は、サーバーの管理者アカウントに切り替え後、以下のコマンドを実施し、“Installed Packages”と“Available Packages”の列に含まれるパッケージ名から、インストール対象のパッケージ名を指定してください。

```
# yum repo-pkgs rhel-7-server-rpms list
Installed Packages
(省略)
Available Packages
(省略)
```

[RHEL8 の rpm パッケージをインストールする場合]

RHEL のコンテナイメージ名は、“registry.redhat.io/ubi8:latest”を指定してください。

パッケージ名は、以下の手順を実施し、インストール対象のパッケージ名を指定してください。

サーバーの管理者アカウントに切り替え後、A.1 で作成した satellite.repo のリポジトリファイルが存在する任意のディレクトリーに移動し、以下のコマンドを実施してください。

```
# cp satellite.repo /etc/yum.repos.d/
```

以下のコマンドを実施し、“Installed Packages”と“Available Packages”の列に含まれるパッケージ名から、インストール対象のパッケージ名を指定してください。

```
# yum repo-pkgs rhel-8-for-x86_64-appstream-rpms list
Installed Packages
(省略)
Available Packages
(省略)
# yum repo-pkgs rhel-8-for-x86_64-baseos-rpms list
Installed Packages
(省略)
Available Packages
(省略)
```

以下のコマンドを実施し、編集したビルドコンフィグファイルを指定し、コンテナイメージを作成してください。

```
$ oc start-build <ビルドコンフィグファイル名> \
--from-dir <Dockerfile を格納しているディレクトリー>
```

コンテナイメージの作成が完了したことを確認します。以下のコマンドを繰り返し実施し、NAME 列にビルドコンフィグファイル名のコンテナイメージが存在し、UPDATED 列に“10 seconds ago”のような時間情報が含まれていることを確認してください。コンテナイメージの作成には時間がかかりますので、確認できるまで間隔を空けて繰り返し実施してください。

```
$ oc get imagestream
NAME                                IMAGE REPOSITORY
TAGS      UPDATED
<ビルドコンフィグファイル名>      image-registry.openshift-image-
registry.svc:5000/<プロジェクト名>/<ビルドコンフィグファイル名>
latest      10 seconds ago
```

下線部分が、時間情報になります。

## 修正履歴

版数と変更の内容を説明します。

版数	変更日	変更箇所	修正概要
1.0 版	2020 年 2 月 17 日	全章	1.0 版作成
2.0 版	2020 年 4 月 1 日	本書をお読みになる前に 2.5.1 クライアントツールの設定 2.5.2 サーバー設定ファイルの作成 3.2 イメージレジストリーの設定 3.3.6 管理設定の追加	RHOCP4.3 サポート開始に伴う 記事更新
3.0 版	2020 年 5 月 20 日	本書をお読みになる前に 1.2 想定システム構成 2.5.2 サーバー設定ファイルの作成 3. コンテナアプリケーションプラットフォームの構築 (RHCOS) 4. コンテナアプリケーションプラットフォームの構築 (RHEL)	RHOCP4 用 RHEL イメージのリ リースに伴う記事更新
3.1 版	2020 年 7 月 16 日	全章	Fujitsu Cloud Service for OSS から FUJITSU Hybrid IT Service FJcloud-O へ名称変更 に伴う表記の変更
3.2 版	2020 年 10 月 1 日	本書をお読みになる前に 1.2 想定システム構成 2.5.1 クライアントツールの設定 2.5.2 サーバー設定ファイルの作成 3.1 仮想サーバーの作成 3.2 イメージレジストリーの設定 3.4 コンテナアプリケーションプラットフォーム構築の完了確認 4.1 仮想サーバーの作成 4.3 クライアントサーバーの設定 4.4 Ansible の実行	RHOCP4.2 サポート終了に伴う 記事削除 RHOCP4.5 サポート開始に伴う 記事更新
4.0 版	2020 年 12 月 16 日	本書をお読みになる前に 1.2 想定システム構成 2.1 仮想サーバーの作成 2.5.1 クライアントツールの設定 3.1 仮想サーバーの作成	RHOCP4.3 サポート終了に伴う 記事削除 RHOCP4.6 サポート開始に伴う 記事更新

		<p>3.3.4 モニタリング Pod の配備先変更</p> <p>3.4 コンテナアプリケーションプラットフォーム構築の完了確認</p> <p>4. コンテナアプリケーションプラットフォームの構築 (RHEL)</p> <p>4.1 仮想サーバーの作成</p> <p>4.2 Worker サーバーの作成</p> <p>4.3 クライアントサーバーの設定</p> <p>4.4 Ansible の実行</p> <p>B.1 Satellite サーバーの情報の取得</p> <p>B.2 Satellite サーバーを利用したパッケージの取得</p>	
4.1 版	2021 年 1 月 14 日	<p>本書をお読みになる前に</p> <p>1.2 想定システム構成</p> <p>4. コンテナアプリケーションプラットフォームの構築 (RHEL)</p> <p>4.1 仮想サーバーの作成</p> <p>4.2 Worker サーバーの作成</p> <p>4.3 クライアントサーバーの設定</p> <p>A. コンテナイメージ取得</p>	RHOCP4.6 の Worker サーバーへの RHEL の使用の制限解除
		<p>B.2 Satellite サーバーを利用したパッケージの取得</p>	RHEL8 のコンテナイメージへの対応の追加
5.0 版	2021 年 4 月 1 日	<p>本書をお読みになる前に</p> <p>1.2 想定システム構成</p> <p>2.2 DNS サーバーの設定</p> <p>2.5.1 クライアントツールの設定</p> <p>3.1 仮想サーバーの作成</p> <p>4.1 仮想サーバーの作成</p> <p>4.2 Worker サーバーの作成</p> <p>4.2.3 rpm パッケージのアップデート</p> <p>A.1 Satellite サーバーの情報の取得</p> <p>A.2 Satellite サーバーを利用したパッケージの取得</p>	<p>RHOCP4.4 サポート終了に伴う記事削除</p> <p>RHOCP4.7 サポート開始に伴う記事更新</p>
6.0 版	2021 年 9 月 15 日	<p>本書をお読みになる前に</p> <p>1.2 想定システム構成</p> <p>2.2 DNS サーバーの設定</p> <p>2.5.1 クライアントツールの設定</p> <p>3.1 仮想サーバーの作成</p> <p>4.1 仮想サーバーの作成</p> <p>4.2 Worker サーバーの作成</p>	<p>RHOCP4.5 サポート終了に伴う記事削除</p> <p>RHOCP4.8 サポート開始に伴う記事更新</p>

7.0 版	2021 年 12 月 1 日	本書をお読みになる前に 1.2 想定システム構成 2.2 DNS サーバーの設定 2.5.1 クライアントツールの設定 4.2.2 rpm パッケージのインストール 先の設定	RHOCP4.9 サポート開始に伴う 記事更新
8.0 版	2022 年 4 月 20 日	本書をお読みになる前に 1.2 想定システム構成 2.2 DNS サーバーの設定 2.5.1 クライアントツールの設定 3.3.4 モニタリング Pod の配備先変更 4.2.1 SSH 公開鍵の登録 4.2.2 rpm パッケージのインストール 先の設定	RHOCP4.10 サポート開始に伴う 記事更新
9.0 版	2022 年 10 月 3 日	本書をお読みになる前に 1.2 想定システム構成 2.2 DNS サーバーの設定 2.5.1 クライアントツールの設定 2.5.2 サーバー設定ファイルの作成 3.3.4 モニタリング Pod の配備先変更 4.2.2 rpm パッケージのインストール 先の設定 4.2.3 rpm パッケージのアップデート 4.3.1 rpm パッケージのインストール	RHOCP4.7 サポート終了に伴う 記事削除 RHOCP4.11 サポート開始に伴う 記事更新
10.0 版	2023 年 3 月 1 日	本書をお読みになる前に 1.2 想定システム構成 2.2 DNS サーバーの設定 2.5.1 クライアントツールの設定 2.5.2 サーバー設定ファイルの作成 3.1 仮想サーバーの作成 3.3.4 モニタリング Pod の配備先変更 3.4 コンテナアプリケーションプラット フォーム構築の完了確認 4 コンテナアプリケーションプラット フォームの構築(RHEL) 4.2.2 rpm パッケージのインストール 先の設定 4.2.3 rpm パッケージのアップデート 4.3.1 rpm パッケージのインストール 4.4 Ansible の実行	RHOCP4.6 サポート終了に伴う 記事削除 RHOCP4.12 サポート開始に伴う 記事更新