OpenShift Container Platform 3.11

はじめに

OpenShift Container Platform 3.11 のスタートガイド

Last Updated: 2023-03-25

OpenShift Container Platform 3.11 はじめに

OpenShift Container Platform 3.11 のスタートガイド

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

開発者またはプラットフォーム管理者は、この本のトピックを使用して OpenShift の使用を開始できます。管理者は、インストールユーティリティーとインタラクティブな CLI ツールを使用して、複数のホストに新しい OpenShift インスタンスをすばやくインストールし、設定できます。開発者は、OpenShift CLI または Web コンソールを使用して、既存の OpenShift インスタンスにログインし、アプリケーションの作成を開始できます。

目次

第1章 概要

1.1. はじめに

OpenShift Container Platform は Docker と Kubernetes を統合し、API を提供してこれらのサービスを管理します。OpenShift Container Platform を使用すると、コンテナーを作成および管理できます。

1.1.1. OpenShift を使用する理由

コンテナーは、独自の環境内で実行するスタンドアロンのプロセスで、オペレーティングシステムや基盤のインフラストラクチャーからは独立しています。OpenShift は、コンテナーベースのアプリケーションを開発、デプロイ、管理するのに役立ちます。また、セルフサービスのプラットフォームを提供し、オンデマンドでアプリケーションを作成、変更、デプロイできるため、開発およびリリースのライフサイクルを迅速化します。

イメージをクッキーカッター、コンテナーを実際のクッキーと**考えます**。

OpenShift はオペレーティングシステム、イメージはその上で実行するアプリケーション、そしてコンテナーはそのアプリケーションの実際の実行インスタンスと**考えます**。

ロールに基づいて適切なトピックを見つけ、スタートします。

対象者	関連するトピックへのリンク
プラットフォーム管理者	基本的な OpenShift Container Platform 環境をインストール するか、OpenShift Container Platform 環境をインストール します。
開発者	Web コンソールを使用してイメージを作成および構築する 方法の基礎手順を順を追って行い、最初のプロジェクトおよびアプリケーションを作成します。

第2章 OPENSHIFT CONTAINER PLATFORM のインストール

2.1. 概要

本書では、OpenShift Container Platform に関する基本的な概念を紹介し、基本的なアプリケーションのインストールに役立ちます。本書は、OpenShift Container Platform の実稼働環境のデプロイやインストールには適していません。

2.1.1. 前提条件

OpenShift Container Platform をインストールするには、以下が必要です。

- 完全修飾ドメイン名 (実世界またはネットワーク内) と パスワードなしの SSH アクセスを備えた少なくとも 2 台の物理または仮想 RHEL 7 以降のマシン。本書では、**master.openshift.example.com** および **node.openshift.example.com** を使用します。これらのマシンは、これらのドメイン名を使用して相互に ping できるようにする必要があります。IBM POWER サーバーを使用する場合、クラスター内のすべてのサーバーは IBM POWER サーバーである必要があります。
- 有効な Red Hat サブスクリプション
- ドメインをノードの IP に解決するワイルドカード DNS 解決。そのため、DNS サーバーに以下のようなエントリーがあります。

```
master.openshift.example.com. 300 IN A <master_ip>
node.openshift.example.com. 300 IN A <node_ip>
*.apps.openshift.example.com. 300 IN A <node_ip>
```

ドメイン名の APPS にワイルドカードのエントリーを指定する理由

OpenShift Container Platform を使用してアプリケーションをデプロイする場合、内部ルーターは受信要求を対応するアプリケーション Pod にプロキシー設定する必要があります。**apps** をアプリケーションドメインの一部として使用すると、アプリケーショントラフィックは適切な Pod に正確にマークされます。

apps 以外のものを使用できます。

```
*.cloudapps.openshift.example.com. 300 IN A <node_ip>
```

これらが設定されたら、以下の手順を使用して 2 台のマシンに OpenShift Container Platform のインストールを設定します。

2.1.2. OpenShift Container Platform サブスクリプションのアタッチ

1. ターゲットマシン (マスターとノードの両方) で、root として **subscription-manager** を使用して、Red Hat にシステムを登録します。

```
# subscription-manager register
```

2. 最新のサブスクリプションデータを Red Hat Subscription Manager (RHSM) からプルします。

```
# subscription-manager refresh
```

3. 利用可能なサブスクリプションを一覧表示します。

```
# subscription-manager list --available
```

4. OpenShift Container Platform サブスクリプションを提供するプール ID を見つけ、これを割り当てます。

```
# subscription-manager attach --pool=<pool_id>
```

5. **<pool_id>** の文字列は、OpenShift Container Platform を提供するプールのプール ID に置き換えます。プール ID は、長い英数字の文字列です。

これらの RHEL システムには、OpenShift Container Platform をインストールできるようになりました。次に、OpenShift Container Platform を取得する場所からシステムに指示する必要があります。

2.1.3. リポジトリーの設定

マスターおよびノードの両方で **subscription-manager** を使用し、OpenShift Container Platform をインストールするのに必要なリポジトリーを有効にします。この例では、最初の 2 つのリポジトリーがすでに有効になっている可能性があります。

- x86_64 サーバーでのクラウドインストールおよびオンプレミスインストールの場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable="rhel-7-server-rpms" \
    --enable="rhel-7-server-extras-rpms" \
    --enable="rhel-7-server-ose-3.11-rpms" \
    --enable="rhel-7-server-ansible-2.9-rpms"
```

- IBM POWER8 サーバーでのオンプレミスインストールの場合は、以下のコマンドを実行します。

```
# subscription-manager repos \
    --enable="rhel-7-for-power-le-rpms" \
    --enable="rhel-7-for-power-le-extras-rpms" \
    --enable="rhel-7-for-power-le-optional-rpms" \
    --enable="rhel-7-server-ansible-2.9-for-power-le-rpms" \
    --enable="rhel-7-server-for-power-le-rhscl-rpms" \
    --enable="rhel-7-for-power-le-ose-3.11-rpms"
```

- IBM POWER9 サーバーでのオンプレミスインストールの場合は、以下のコマンドを実行します。

```
# subscription-manager repos \
    --enable="rhel-7-for-power-9-rpms" \
    --enable="rhel-7-for-power-9-extras-rpms" \
    --enable="rhel-7-for-power-9-optional-rpms" \
    --enable="rhel-7-server-ansible-2.9-for-power-9-rpms" \
    --enable="rhel-7-server-for-power-9-rhscl-rpms" \
    --enable="rhel-7-for-power-9-ose-3.11-rpms"
```

このコマンドは、RHEL システムに対し、OpenShift Container Platform のインストールに必要なツールがこれらのリポジトリーから利用できるようになることを示します。次に、Ansible をベースとする OpenShift Container Platform インストーラーが必要になります。

注記

以前のバージョンの OpenShift Container Platform 3.11 は Ansible 2.6 のみをサポートしていました。最新バージョンの Playbook が Ansible 2.9 に対応するようになりました。Ansible 2.9 は、使用する推奨バージョンです。

2.1.4. OpenShift Container Platform パッケージのインストール

OpenShift Container Platform のインストーラーは、**openshift-ansible** パッケージで提供されます。マスターで **yum** を使用してインストールしてから、**yum update** を実行します。

```
# yum -y install wget git net-tools bind-utils iptables-services bridge-utils bash-completion kexec-tools sos psacct
```

```
# yum -y update
```

```
# reboot
```

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

ここでコンテナーエンジンをインストールします。

- CRI-O をインストールするには、以下を実行します。

  ```
  # yum -y install cri-o
  ```

- Docker をインストールするには、以下を実行します。

  ```
  # yum -y install docker
  ```

2.1.5. パスワードなしの SSH アクセスの設定

マスターでインストーラーを実行する前に、パスワードなしの SSH アクセスを設定します。これは、インストーラーがマシンにアクセスするのに必要です。マスターで以下のコマンドを実行します。

```
$ ssh-keygen
```

プロンプトに従い、パスフレーズを求められたら Enter を押してください。

SSH キーを配布する簡単な方法として、**bash** ループを使用できます。

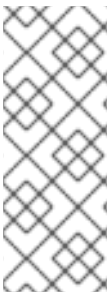
```
$ for host in master.openshift.example.com \
    node.openshift.example.com; \
    do ssh-copy-id -i ~/.ssh/id_rsa.pub $host; \
    done
```

2.1.6. インストール Playbook の実行

1. インベントリーファイルの例 を参照して、必要なクラスター設定に最も近いサンプルを選択します。

重要

Ansible Playbook を **--tags** または **--check** オプションを使用して実行することを、Red Hat ではサポートしていません。



注記

追加のホストファイルの例は、**/usr/share/doc/openshift-ansible-docs-3.11.<version>/docs/example-inventories/** ディレクトリーの参照として利用できます (**<version>** を、**openshift-ansible-docs** パッケージがインストールされている最新のバージョンに置き換えます。これにより、**openshift-ansible** の親パッケージがアップグレードされるたびに更新されます)。利用可能なインベントリー変数の詳細は、Configuring Your Inventory File を参照してください。

1. ホスト名を使用するようにインベントリー例を編集して、ファイルに保存します。デフォルトの場所は **/etc/ansible/hosts** です。
2. Playbook ディレクトリーに切り替え、ご自身のイベントリーファイルを使用して、**prerequisites.yml** Playbook を実行します。

```
$ cd /usr/share/ansible/openshift-ansible
```

```
$ ansible-playbook -i <inventory_file> playbooks/prerequisites.yml
```

3. Playbook ディレクトリーに切り替え、ご自身のイベントリーファイルを使用して **deploy_cluster.yml** Playbook を実行します。

```
$ cd /usr/share/ansible/openshift-ansible
```

```
$ ansible-playbook -i <inventory_file> playbooks/deploy_cluster.yml
```

インストールに成功したら、新規プロジェクトを追加する前に、Basic 認証、ユーザーアクセス、およびルートを設定する必要があります。

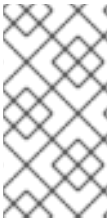
2.2. OPENSHIFT CONTAINER PLATFORM との対話

OpenShift Container Platform では、対話するための 2 つのコマンドラインユーティリティーがあります。

- **oc**: 通常のプロジェクトおよびアプリケーション管理用
- **oc adm**: 管理タスク用
  oc adm コマンドを実行すると、Ansible ホストインベントリーファイル (デフォルトでは **/etc/ansible/hosts**) に記載されている最初のマスターからのみ実行する必要があります。

oc --help および **oc adm --help** を使用して、利用可能なすべてのオプションを表示します。

さらに、Web コンソールを使用してプロジェクトおよびアプリケーションを管理できます。Web コンソールは **https://<master_fqdn>:8443/console** で利用できます。次のセクションでは、コンソールにアクセスするためのユーザーアカウントを作成する方法を確認できます。



注記

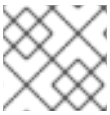
また、これらのコマンドラインユーティリティーを使用して、リモートシステムから OpenShift Container Platform インスタンスと対話できます。OpenShift CLI としてバンドルされており、CLI リファレンス セクションで Windows、Mac、または Linux 環境用のユーティリティーをダウンロードできます。

2.3. ロールおよび認証について

デフォルトでは、初回インストール時には OpenShift Container Platform にロールやユーザーアカウントが作成されないため、作成する必要があります。オプションとして、(最初に) 新規ロールを作成するか、誰でもログインできるポリシーを定義する方法があります。

その他の作業を行う前に、デフォルトの **system:admin** ユーザーで最低でも 1 回ログインします。マスターで以下のコマンドを実行します。

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



注記

これ以降、特に指示がない限り、コマンドはマスターで実行する必要があります。

このアカウントで最低でも 1 回ログインして、**system:admin** ユーザーの設定ファイルを作成します。これにより、後でログインできるようになります。

このシステムアカウントのパスワードはありません。

以下のコマンドを実行して、OpenShift Container Platform が正常にインストールし、起動されていることを確認します。マスターとノードが **Ready** ステータスで表示されます。

```
$ oc get nodes
```

基本的な OpenShift Container Platform 環境の設定を続行するには、OpenShift Container Platform の設定 で説明されている手順に従います。

第3章 OPENSHIFT CONTAINER PLATFORM の設定

3.1. 概要

本書では、OpenShift Container Platform に関する基本的な概念を紹介し、基本的なアプリケーションの設定に役立ちます。本書では、OpenShift Container Platform の基本環境のインストール の設定手順について説明し、OpenShift の実稼働環境のデプロイやインストールには適していません。

3.2. ログイン ID プロバイダーの変更

OpenShift Container Platform インスタンスを新規インストールした場合のデフォルトの動作として、全ユーザーのログインを拒否します。認証方法を HTPasswd に変更するには、以下を実行します。

1. **/etc/origin/master/master-config.yaml** ファイルを編集モードで開きます。
2. **identityProviders** のセクションを検索します。
3. **DenyAllPasswordIdentityProvider** を **HTPasswdPasswordIdentityProvider** プロバイダーに変更します。
4. 名前ラベルの値を **htpasswd_auth** に変更し、プロバイダーセクションに **file: /etc/origin/master/htpasswd** の新しい行を追加します。**HTPasswdPasswordIdentityProvider** を使用した **identityProviders** セクションの例は以下のようになります。

```
oauthConfig:
  ...
  identityProviders:
  - challenge: true
    login: true
    name: htpasswd_auth provider
    provider:
      apiVersion: v1
      kind: HTPasswdPasswordIdentityProvider
      file: /etc/origin/master/htpasswd
```

5. ファイルを保存します。

3.3. ユーザーアカウントの作成

HTPasswdPasswordIdentityProvider プロバイダーを使用している場合は、これらのユーザーアカウントを生成する必要があります。

1. **httpd-tools** パッケージを使用して、これらのアカウントを生成できる **htpasswd** バイナリーを取得できます。

```
# yum -y install httpd-tools
```

2. ユーザーアカウントを作成します。

```
# touch /etc/origin/master/htpasswd
```

```
# htpasswd -b /etc/origin/master/htpasswd admin redhat
```

admin というユーザーを作成し、パスワードを **redhat** にします。

3. 続行する前に OpenShift を再起動します。

```
# master-restart api
```

```
# master-restart controllers
```

4. このユーザーアカウントに **cluster-admin** 権限を付与し、すべてを実行できます。

```
$ oc adm policy add-cluster-role-to-user cluster-admin admin
```

oc adm コマンドを実行すると、Ansible ホストインベントリーファイル (デフォルトでは **/etc/ansible/hosts**) に記載されている最初のマスターからのみ実行する必要があります。

5. このユーザー名/パスワードの組み合わせを使用して、Web コンソールまたはコマンドラインからログインできます。これをテストするには、以下のコマンドを実行します。

```
$ oc login -u admin
```

次に進む前に、**default** プロジェクトに切り替えます。

```
$ oc project default
```

詳細は、ロール および 認証 を参照してください。

3.4. OPENSHIFT ルーターのデプロイ

OpenShift ルーターは、OpenShift サービスの外部ネットワークトラフィックのエントリーポイントです。SNI を使用する HTTP、HTTPS および TLS 対応のトラフィックをサポートし、ルーターが正しいサービスにトラフィックを送信できるようにします。

ルーターがないと、OpenShift サービスおよび Pod は、OpenShift インスタンス外のリソースと通信できません。

インストーラーはデフォルトのルーターを作成します。

1. 以下のコマンドを使用してデフォルトのルーターを削除します。

```
$ oc delete all -l router=router
```

2. 新規デフォルトルーターを作成します。

```
$ oc adm router --replicas=1 --service-account=router
```

OpenShift ドキュメントには、ルーターの概要 に関する詳細情報が記載されています。

3.5. 内部レジストリーのデプロイ

OpenShift は、イメージをローカルで管理するためにデプロイできる内部の 統合コンテナーイメージレジストリー を提供します。OpenShift は、**docker-registry** を使用して、コンテナーイメージを保存、取得、ビルドし、ライフサイクル全体でそれらをデプロイおよび管理します。

インストーラーはデフォルトのレジストリーを作成します。

第4章 WEB コンソールを使用したイメージの作成およびビルド

4.1. 概要

このスタートガイドでは、最もシンプルな方法で、OpenShift Container Platform でサンプルプロジェクトを稼働させる方法を説明します。プロジェクトでイメージを起動する方法は複数ありますが、このトピックでは、最もすばやく簡単に起動する方法に焦点を当てていきます。

本書を読み始めており、OpenShift Container Platform バージョン 3 (v3) のコアとなる概念に慣れていない場合は、まず 新機能 について読むことから始めることができます。このバージョンの OpenShift Container Platform は、バージョン 2 (v2) とは大きく異なります。

OpenShift Container Platform 3 は、アプリケーション開発を開始できるようにする対応する実装とチュートリアルを備えた、開発者向けの 言語 と データベース のセットをすぐに提供します。言語サポートは クイックスタートテンプレート を中心としており、クイックスタートテンプレートは ビルダーイメージ を活用します。

言語	実装およびチュートリアル
Ruby	Rails
Python	Django
Node.js	Node.js
PHP	CakePHP
Perl	Dancer
Java	

OpenShift Container Platform が提供する他のイメージには以下が含まれます。

- MySQL
- MongoDB
- PostgreSQL
- Jenkins

さらに、JBoss Middleware では広範囲に及ぶ OpenShift Container Platform テンプレート を利用できます。

特に xPaaS サービスで利用可能な技術は以下のとおりです。

- JBoss EAP 6 が提供する Java EE 6 Application Server
- JBoss Fuse および JBoss A-MQ が提供するインテグレーションおよびメッセージングサービス
- JBoss Data Grid が提供する Data Grid サービス

- JBoss BRMS が提供する Real Time Decision Service
- Tomcat 7 および Tomcat 8 が提供する Java Web Server 3.0

これらの各オファリングでは、一連の組み合わせが提供されます。

- HTTP のみ、または HTTP および HTTPS
- データベースは必要ありません。または MongoDB、PostgreSQL、または MySQL のいずれかを使用。
- A-MQ との統合 (必要な場合)

このようなアプリケーションの構築を例示するために、以下のセクションではプロジェクトの作成を説明します。このプロジェクトには、Welcome ページや現在のヒット数 (データベースに保存) を提供する Node.js サンプルアプリケーションが含まれています。

注記

このトピックでは、クイックスタート と インスタントアプリ のテンプレートとアプリケーションの両方について説明します。クイックスタートはアプリケーション開発のスタート地点を提供しますが、便利なアプリケーションを作成するには開発に依存します。これとは対照的に、Jenkins などのインスタントアプリは即座に利用できます。

4.1.1. ブラウザーの要件

ブラウザーのバージョンとオペレーティングシステム を使用して、Web コンソールにアクセスできることを確認します。

4.2. 作業を開始する前に

作業を開始する前に、以下を確認してください。

- 実行中の OpenShift Container Platform インスタンスにアクセスできる必要があります。アクセスできない場合は、クラスター管理者にお問い合わせください。
- インスタンスは、クラスター管理者によって、インスタントアプリのテンプレート および ビルダーイメージ で事前に設定されている必要があります。利用できない場合は、クラスター管理者に Loading the Default Image Streams and Templates を参照してもらうようにしてください。
- OpenShift Container Platform CLI を ダウンロードおよびインストール しておく必要があります。

4.3. サンプルリポジトリーのフォーク

1. GitHub にログインした状態で Ruby の例 のページに移動します。

注記

以下のトピックは Ruby の例に沿っていますが、OpenShift Container Platform GitHub プロジェクトで提供される 言語であればどれでも使用できます。

2. リポジトリーをフォーク します。

新しいフォークにリダイレクトされます。

3. フォーク用のクローン URL をコピーします。

4. リポジトリーをローカルマシンにクローンします。

4.4. プロジェクトの作成

アプリケーションを作成するには、最初に新規プロジェクトを作成してから、InstantApp テンプレートを選択します。そこから、OpenShift Container Platform はビルドプロセスを開始し、新規デプロイメントを作成します。

1. ブラウザーで OpenShift Container Platform Web コンソールに移動します。Web コンソールは自己署名証明書を使用するため、プロンプトが表示されたら、ブラウザーの警告を OK にして続行します。

2. 管理者が推奨するユーザー名およびパスワードを使用してログインします。

3. 新規プロジェクトを作成するには、**New Project** をクリックします。

4. 新規プロジェクトの一意の名前、表示名、および説明を入力します。

5. **Create** をクリックします。
Web コンソールの welcome 画面が読み込まれます。

4.5. アプリケーションの作成

Select Image または Template ページでは、一般にアクセス可能な git リポジトリーまたはテンプレートから作成するオプションが表示されます。

1. 新規プロジェクトを作成しても、Select Image または Template ページに自動的にリダイレクトされない場合は、**Add to Project** をクリックする必要がある場合があります。

2. **Browse** をクリックして、ドロップダウンリストから **ruby** を選択します。

3. **ruby:latest** ビルダーイメージをクリックします。

4. アプリケーションの **名前** を入力し、**Git リポジトリーの URL** (**https://github.com/<your_github_username>/ruby-ex.git**) を指定します。

5. オプションで、**Show advanced routing, build, and deployment options**をクリックします。ただし、デフォルトでは、このアプリケーションサンプルでは自動的にルート、Webhook トリガー、およびビルド変更トリガーが作成されます。

6. **Create** をクリックします。

注記

作成後に、Web コンソールからこれらの設定の一部を変更するには、**Browse**、**Builds** をクリックしてビルドを選択してから、**Actions** をクリックし、**Edit** または **Edit YAML** のいずれかをクリックします。

アプリケーションの作成には時間がかかる場合があります。Web コンソールの Overview ページで、作成される新規リソースを表示し、ビルドやデプロイメントの進捗を確認できます。

Ruby Pod が作成されても、Pod のステータスは保留中と表示されます。次に、Ruby Pod が起動し、新たに割り当てられた IP アドレスが表示されます。Ruby Pod が実行されると、ビルドが完了します。

4.6. アプリケーションの実行の確認

DNS が正しく設定されている場合、新規アプリケーションは Web ブラウザーからアクセスできます。アプリケーションにアクセスできない場合は、システム管理者にお問い合わせください。

新規アプリケーションを表示するには、以下を実行します。

4.7. 自動化ビルドの設定

OpenShift Container Platform GitHub リポジトリー からこのアプリケーションのソースコードをフォークした。そのため、フォークしたリポジトリーにコードの変更をプッシュするたびに、Webhook を使用してアプリケーションのリビルドを自動的にトリガーできます。

アプリケーションの Webhook を設定するには、以下を実行します。

1. Web コンソールから、アプリケーションが含まれるプロジェクトに移動します。
2. **Browse** タブをクリックしてから、**Builds** をクリックします。
3. ビルド名をクリックしてから、**Configuration** タブをクリックします。
4. **GitHub webhook URL** の横にある  をクリックして、Webhook ペイロード URL をコピーします。
5. GitHub のフォークされたリポジトリーに移動してから **Settings** をクリックします。
6. **Webhooks & Services** をクリックします。
7. **Add webhook** をクリックします。
8. Webhook URL を **Payload URL** フィールドに貼り付けます。
9. **Content Type** を **application/json** に設定します。
10. **Add webhook** をクリックして保存します。

GitHub は、ping ペイロードを OpenShift Container Platform サーバーに送信して、通信が成功したことを確認します。Webhook URL の横に緑色のチェックマークが表示された場合は、正しく設定されています。チェックマークの上にマウスをかざして、最終配信のステータスを表示します。

フォークしたリポジトリーにコードの変更をプッシュする次回のタイミングで、アプリケーションが自動的に再ビルドされます。

4.8. コード変更の記述

ローカルで作業して、アプリケーションに変更をプッシュするには、以下を実行します。

1. ローカルマシンで、テキストエディターを使用して、**ruby-ex/config.ru** ファイルのサンプルアプリケーションのソースを変更します。

2. コードの変更をアプリケーション内から表示できるようにします。たとえば、229 行目で、タイトルを **Welcome to your Ruby application on OpenShift** から **This is my Awesome OpenShift Application** に変更してから、変更を保存します。

3. git に変更をコミットし、変更をフォークにプッシュします。
Webhook が正しく設定されている場合は、変更を基に、アプリケーションは即座にリビルドされます。再ビルドに成功したら、先に作成したルートを使用して更新されたアプリケーションを表示します。

今後は、コードの更新をプッシュするだけで、OpenShift ContainerPlatform が残りを処理します。

4.8.1. イメージの手動リビルド

Webhook が機能していない場合、またはビルドが失敗してビルドを再開する前にコードを変更したくない場合は、イメージを手動で再構築すると便利な場合があります。最新のコミットされた変更に基づいて、フォークしたリポジトリーにイメージを手動でリビルドするには、以下を実行します。

1. **Browse** タブをクリックしてから、**Builds** をクリックします。

2. ビルドを探し出して、**Start Build** をクリックします。

第5章 CLI を使用したイメージの作成およびビルド

5.1. 概要

このスタートガイドでは、最もシンプルな方法で、OpenShift Container Platform でサンプルプロジェクトを稼働させる方法を説明します。プロジェクトでイメージを起動する方法は複数ありますが、このトピックでは、最もすばやく簡単に起動する方法に焦点を当てていきます。

本書を読み始めており、OpenShift Container Platform バージョン 3 (v3) のコアとなる概念に慣れていない場合は、まず 新機能 について読むことから始めることができます。このバージョンの OpenShift Container Platform は、バージョン 2 (v2) とは大きく異なります。

OpenShift Container Platform 3 は、アプリケーション開発を開始できるようにする対応する実装とチュートリアルを備えた、開発者向けの 言語 と データベース のセットをすぐに提供します。言語サポートは クイックスタートテンプレート を中心としており、クイックスタートテンプレートは ビルダーイメージ を活用します。

言語	実装およびチュートリアル
Ruby	Rails
Python	Django
Node.js	Node.js
PHP	CakePHP
Perl	Dancer
Java	

OpenShift Container Platform が提供する他のイメージには以下が含まれます。

- MySQL
- MongoDB
- PostgreSQL
- Jenkins

さらに、JBoss Middleware では広範囲に及ぶ OpenShift Container Platform テンプレート を利用できます。

特に xPaaS サービスで利用可能な技術は以下のとおりです。

- JBoss EAP 6 が提供する Java EE 6 Application Server
- JBoss Fuse および JBoss A-MQ が提供するインテグレーションおよびメッセージングサービス
- JBoss Data Grid が提供する Data Grid サービス

- JBoss BRMS が提供する Real Time Decision Service
- Tomcat 7 および Tomcat 8 が提供する Java Web Server 3.0

これらの各オファリングでは、一連の組み合わせが提供されます。

- HTTP のみ、または HTTP および HTTPS
- データベースは必要ありません。または MongoDB、PostgreSQL、または MySQL のいずれかを使用。
- A-MQ との統合 (必要な場合)

このようなアプリケーションの構築を例示するために、以下のセクションではプロジェクトの作成を説明します。このプロジェクトには、Welcome ページや現在のヒット数 (データベースに保存) を提供する Node.js サンプルアプリケーションが含まれています。

注記

このトピックでは、クイックスタート と インスタントアプリ のテンプレートとアプリケーションの両方について説明します。クイックスタートはアプリケーション開発のスタート地点を提供しますが、便利なアプリケーションを作成するには開発に依存します。これとは対照的に、Jenkins などのインスタントアプリは即座に利用できます。

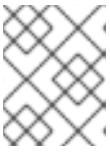
5.2. 作業を開始する前に

作業を開始する前に、以下を確認してください。

- 実行中の OpenShift Container Platform インスタンスにアクセスできる必要があります。アクセスできない場合は、クラスター管理者にお問い合わせください。
- インスタンスは、クラスター管理者によって、インスタントアプリのテンプレート および ビルダーイメージ で事前に設定されている必要があります。利用できない場合は、クラスター管理者に Loading the Default Image Streams and Templates を参照してもらうようにしてください。
- OpenShift Container Platform CLI を ダウンロードおよびインストール しておく必要があります。

5.3. サンプルリポジトリーのフォーク

1. GitHub にログインした状態で Ruby の例 のページに移動します。



注記

以下のトピックは Ruby の例に沿っていますが、OpenShift Container Platform GitHub プロジェクトで提供される 言語であればどれでも使用できます。

2. リポジトリーをフォーク します。
新しいフォークにリダイレクトされます。
3. フォーク用のクローン URL をコピーします。
4. リポジトリーをローカルマシンにクローンします。

5.4. プロジェクトの作成

アプリケーションを作成するには、新規プロジェクトを作成し、ソースの場所を指定する必要があります。そこから、OpenShift Container Platform はビルドプロセスを開始し、新規デプロイメントを作成します。

1. CLI から OpenShift Container Platform にログインします。
   - ユーザー名とパスワードを使用してログインする場合:

     ```
     $ oc login -u=<username> -p=<password> --server=<your-openshift-server> --insecure-skip-tls-verify
     ```

   - oauth トークンの場合:

     ```
     $ oc login <https://api.your-openshift-server.com> --token=<tokenID>
     ```

2. 新しいプロジェクトを作成します。

   ```
   $ oc new-project <projectname> --description="<description>" --display-name="<display_name>"
   ```

新しいプロジェクトの作成後、新規プロジェクトの namespace に自動的に切り替えられます。

5.5. アプリケーションの作成

フォークしたリポジトリーのコードから新規アプリケーションを作成するには、以下を実行します。

1. コードのソースを指定してアプリケーションを作成します。

   ```
   $ oc new-app openshift/ruby-20-centos7~https://github.com/<your_github_username>/ruby-ex
   ```

   OpenShift Container Platform は一致するビルダーイメージを検索して (この場合は **ruby-20-centos7**)、アプリケーションのリソース (イメージストリーム、ビルド設定、デプロイメント設定、サービス) を作成します。また、ビルドのスケジューリングも行います。

2. ビルドの進捗を追跡します。

   ```
   $ oc logs -f bc/ruby-ex
   ```

3. ビルドが完了し、作成されたイメージがレジストリーに正常にプッシュされたら、アプリケーションのステータスを確認します。

   ```
   $ oc status
   ```

   または、Web コンソールからビルドを表示できます。

アプリケーションの作成には時間がかかる場合があります。Web コンソールの Overview ページで、作成される新規リソースを表示し、ビルドやデプロイメントの進捗を確認できます。**oc get pods** コマンドを使用して Pod が稼働していることを確認したり、**oc getbuilds** コマンドを使用してビルド統計を確認したりすることもできます。

Ruby Pod が作成されても、Pod のステータスは保留中と表示されます。次に、Ruby Pod が起動し、新たに割り当てられた IP アドレスが表示されます。Ruby Pod が実行されると、ビルドが完了します。

oc status コマンドは、サービスが実行している IP アドレスを表示します。デプロイ先のデフォルトのポートは 8080 です。

5.6. ルートの作成

OpenShift Container Platform ルートは、外部クライアントが名前でサービスに到達できるように、ホスト名でサービスを公開します。新規アプリケーションへのルートを作成するには、以下を実行します。

1. **ruby-ex** のサービスを公開します。

```
$ oc expose service ruby-ex
```

2. 新規ルートを表示します。

```
$ oc get route
```

5.7. アプリケーションの実行の確認

新規アプリケーションを表示するには、コピーしたルートの場所を Web ブラウザーのアドレスバーに貼り付け、Enter を押します。

サンプルの **ruby-ex** アプリケーションは、単純な Welcome 画面となっており、コード変更のデプロイ、アプリケーションや他の開発リソースの管理方法に関する詳細が含まれます。

次に、GitHub Webhook トリガーで自動ビルドを設定し、フォークしたリポジトリーのコードを変更すると、アプリケーションがリビルドされます。

5.8. 自動化ビルドの設定

OpenShift Container Platform GitHub リポジトリー からこのアプリケーションのソースコードをフォークした。そのため、フォークしたリポジトリーにコードの変更をプッシュするたびに、Webhook を使用してアプリケーションのリビルドを自動的にトリガーできます。

アプリケーションの Webhook を設定するには、以下を実行します。

1. **BuildConfig** のトリガーセクションを表示して、GitHub Webhook トリガーが存在することを確認します。

```
$ oc edit bc/ruby-ex
```

以下のような出力が表示されます。

```
triggers
- github:
    secret: Q1tGY0i9f1ZFihQbX07S
    type: GitHub
```

シークレットは、ユーザーおよびリポジトリーのみがビルドをトリガーできるようにします。

2. 以下のコマンドを実行して、**BuildConfig** に関連付けられた Webhook URL を表示します。

```
$ oc describe bc ruby-ex
```

3. 上記のコマンドで出力される GitHub Webhook ペイロード URL をコピーします。
4. GitHub のフォークされたリポジトリーに移動してから **Settings** をクリックします。
5. **Webhooks & Services** をクリックします。
6. **Add webhook** をクリックします。
7. Webhook URL を **Payload URL** フィールドに貼り付けます。
8. **Content Type** を **application/json** に設定します。
9. **Add webhook** をクリックして保存します。

GitHub は、ping ペイロードを OpenShift Container Platform サーバーに送信して、通信が成功したことを確認します。Webhook URL の横に緑色のチェックマークが表示された場合は、正しく設定されています。チェックマークの上にマウスをかざして、最終配信のステータスを表示します。

フォークしたリポジトリーにコードの変更をプッシュする次回のタイミングで、アプリケーションが自動的に再ビルドされます。

5.9. コード変更の記述

ローカルで作業して、アプリケーションに変更をプッシュするには、以下を実行します。

1. ローカルマシンで、テキストエディターを使用して、**ruby-ex/config.ru** ファイルのサンプルアプリケーションのソースを変更します。
2. コードの変更をアプリケーション内から表示できるようにします。たとえば、229 行目で、タイトルを **Welcome to your Ruby application on OpenShift** から **This is my Awesome OpenShift Application** に変更してから、変更を保存します。
3. git に変更をコミットし、変更をフォークにプッシュします。
Webhook が正しく設定されている場合は、変更を基に、アプリケーションは即座にリビルドされます。再ビルドに成功したら、先に作成したルートを使用して更新されたアプリケーションを表示します。

今後は、コードの更新をプッシュするだけで、OpenShift ContainerPlatform が残りを処理します。

5.9.1. イメージの手動リビルド

Webhook が機能していない場合、またはビルドが失敗してビルドを再開する前にコードを変更したくない場合は、イメージを手動で再構築すると便利な場合があります。最新のコミットされた変更に基づいて、フォークしたリポジトリーにイメージを手動でリビルドするには、以下を実行します。

```
$ oc start-build ruby-ex
```

5.10. トラブルシューティング

プロジェクトの変更

oc new-project コマンドは、現在のプロジェクトを作成したばかりのプロジェクトに自動的に設定しますが、次のコマンドを実行することで、いつでもプロジェクトを変更できます。

```
$ oc project <project-name>
```

プロジェクトの一覧を表示します。

```
$ oc get projects
```

ビルドの手動トリガー

ビルドが自動的に起動しない場合は、ビルドを開始して、ログをストリームします。

```
$ oc start-build ruby-ex --follow
```

または、上記のコマンドに **--follow** を含めず、代わりにビルドをトリガーした後に次のコマンドを発行します。ここで、**n** は、追跡するビルドの番号です。

```
$ oc logs -f build/ruby-ex-n
```