

作成日：2013年2月17日

更新日：2013年3月01日

「PaaS 基盤レポート」

OpenShift

2012 年度 ITA コミュニティ (東京)

さくら情報システム株式会社
技術部 開発技術グループ
森 隆彦

目次

[はじめに](#)

[対応プラットフォーム](#)

[開発言語とフレームワーク](#)

[サーバ](#)

[データベース](#)

[サービス](#)

[その他](#)

[PaaS](#)

[開発ツール](#)

[Web ダッシュボード \(Web Console\)](#)

[アカウント管理](#)

[Namespace](#)

[各カートリッジの管理](#)

[CLI \(rhc\)](#)

[インストール方法](#)

[Ruby のインストール](#)

[Git のインストール](#)

[CLI のインストール](#)

[利用方法](#)

[GUI \(IDE\)](#)

[インストール方法](#)

[利用方法](#)

[SSH 接続](#)

[利用方法](#)

[ローカル実行環境 \(OpenShift Origin\)](#)

[インストール方法](#)

[利用方法](#)

[開発～運用](#)

[デプロイ方法](#)

[データ](#)

[アプリ](#)

[新規](#)

[更新](#)

[テスト・デバッグ](#)

[リモート](#)

[サーバログのリアルタイム監視](#)

[ログ解析](#)

[テスト](#)

[負荷テスト](#)

[ターゲット](#)

[ステージング](#)

[アクセス制限について](#)

[プロダクション](#)

[監視](#)

[アプリケーション](#)

[死活監視](#)

[自動復旧](#)

[サービス](#)

[死活監視](#)

[自動復旧](#)

[メンテナンス](#)

[アプリケーション](#)

[サービス](#)

[バックアップ](#)

[リストア](#)

[PaaS 実績](#)

[コミュニティの活動](#)

[情報量](#)

[サービスの更新間隔](#)

[利用企業、サービスなど](#)

[ビジネス](#)

[費用](#)

[ランニングコスト](#)

[オプション](#)

[中立性](#)

[ライセンス](#)

[ベンダー依存](#)

[ポータビリティ](#)

[おわりに](#)

はじめに

OpenShift とは、RedHat 社が手掛ける PaaS 基盤である。

OpenShift は以下の二つの要素から構成される。

- Broker
様々な管理機能を提供するホスト。ユーザーのログイン情報、DynDNS、アプリケーションの状態を管理する。
- Cartridge
アプリケーションを実行するのに必要な機能を提供するコンポーネント。Web カートリッジ、Database カートリッジ、管理ツールや生産性向上ツールのためのカートリッジなどがある。
また、Do-It-Yourself (DIY) というカスタマイズ可能なカートリッジが用意されている。

対応プラットフォーム

OpenShift が対応するプラットフォームや各々のバージョンは日々進化しているので、利用前には必ず再確認が必要である。

開発言語とフレームワーク

OpenShift の主要な開発言語は Java と PHP で、次いで Ruby に関する情報が多い。以下に 2013 年 2 月 1 日現在の対応する開発言語とフレームワークを示す。

- Java 6, 7 (7 が初期値)
 - Java EE
 - Spring
 - Liferay
 - Scala/Play! Framework
 - Solr/Lucene
- PHP 5.3
 - Zend
 - CodeIgniter
 - CakePHP
 - Symfony
- Ruby 1.9.3, 1.8.?
 - Ruby on Rails
 - Sinatra
 - Padrino
 - Sequel
- Python 2.7
 - Django
 - Flask
- Perl 5.10
- Node.js 0.6

サーバ

- JBoss AS 7
- JBoss EAP6
- Tomcat 7
- Jetty
- Glassfish

データベース

- MongoDB 2.0
- MySQL 5.1
- PostgreSQL 8.4

サービス

その他

- PhpMyAdmin 3.4
- RockMongo 1.1
- Jenkins Server
- Jenkins Client 1.4
- Cron 1.4

PaaS

ここでは、OpenShift の PaaS 基盤としての基本的な情報を示す。

2013年2月1日現在、OpenShift には無償 (FreeShift) と有償 (MegaShift) の二つのパターンがある。下表に詳細を示す。

	無償 (FreeShift)	有償 (MegaShift)
価格	無償	\$42/月
アプリケーション 最大アプリ数 最大メモリサイズ 最大ファイルサイズ 最大サービス数	3 ギアまで ギアサイズ : Small のみ 1 ギアにつき、メモリ 512MB、ディスク 1GB	16 ギアまで ギアサイズ : Small, \$0.05/hr Medium, \$0.12/hr
Web ソケットサポート	○	○
SSL サポート	共有	カスタムドメイン
スケーリング オートスケール 水平 垂直	3 ギアまで	含まれる
カスタムドメイン	? (CLI からできそう)	○
追加ストレージ	×	月あたり \$1/GB
サポート	コミュニティ	RedHat 社
ソースコードリポジトリ	Git	Git
CI サポート	Jenkins	Jenkins
SLA	×	×
パフォーマンス	?	?
Job 管理	Cron	Cron
チケット管理	×	×

開発ツール

開発ツールには、Web コンソール、コマンドラインインタフェース、IDE の三つの主要ツールがある。また、実行環境がインストールされた起動可能な LiveCD が試行的に用意されている。

Web ダッシュボード (Web Console)

素早くブラウザからアプリケーションを生成することができる。
新しいアプリケーションを起動したり、管理やモニタリングに向いている。

アカウント管理

My Account 画面から、パスワードや名前空間 (後述)、SSH キーの管理を行うことができる。
SSH キーは、コマンドラインをセットアップした際や、Eclipse プラグインから登録すると便利である。

The screenshot shows the 'My Account' page with the following sections:

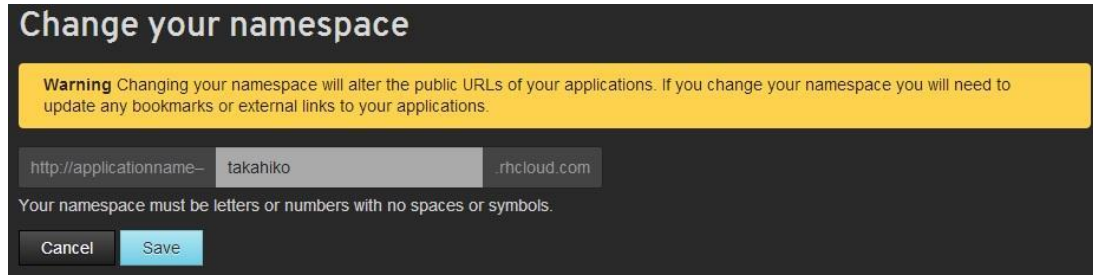
- Personal Information:** OpenShift currently allows access via an existing Red Hat account or through an account created with your e-mail and password. Login: takahiko1971jp@gmail.com, Account Type: Red Hat. Includes a 'Change password...' button.
- Plan:** You are using FreeShift plan in our Developer Preview. Includes a link to 'Learn more about upcoming OpenShift plans and pricing.'
- Public Keys:** OpenShift uses a public key to securely encrypt the connection between your local machine and your application and to authorize you to upload code. You must create a private and public key on your local machine and then upload the public key before you can connect to your applications' Git repositories or remotely access your application. Includes a table of keys and an 'Add a new key...' button.
- Namespace:** Your namespace is unique to your account and is the suffix of the public URLs we assign to your applications. See the User Guide for information about adding your own domain names to an application. Includes a text input field with 'http://applicationname-takahiko.rhcloud.com' and a 'Change your namespace...' button.

Key Name	Public Key	Action
takahiko197...	AAAB3Nza..N9VsfQnz	Delete
cubebblack	AAAB3Nza..A+1JlQ==	Delete
TC10001	AAAB3Nza..wwSQyXB5	Delete

Namespace

Web Console でアカウントを作成後、アプリケーションの名前空間を定義する。
アプリケーションの URL は、以下のようになる。

`http://applicationname- (名前空間名) .rhcloud.com`



Change your namespace

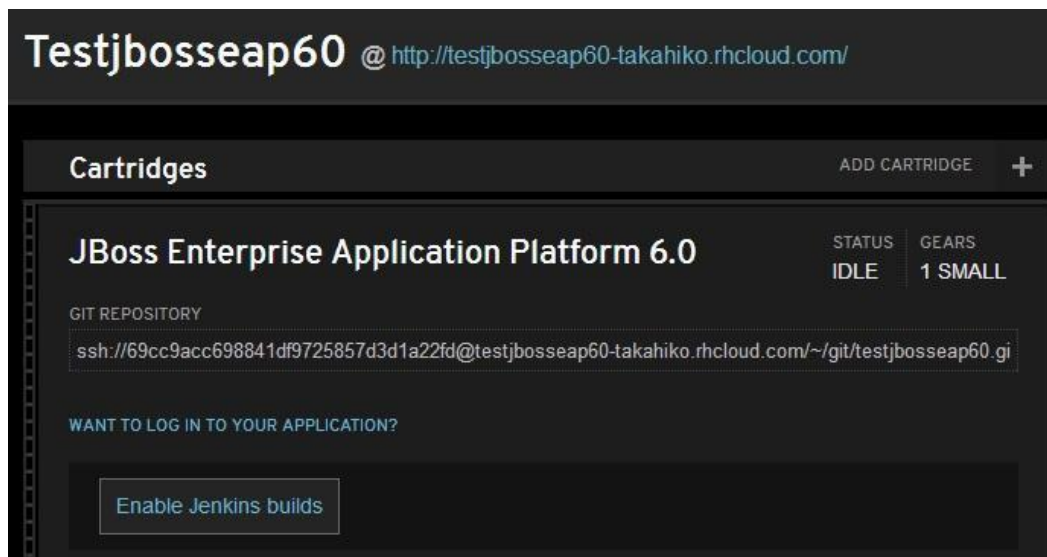
Warning Changing your namespace will alter the public URLs of your applications. If you change your namespace you will need to update any bookmarks or external links to your applications.

`http://applicationname-` `.rhcloud.com`

Your namespace must be letters or numbers with no spaces or symbols.

各カートリッジの管理

各カートリッジの管理画面から、Git リポジトリの閲覧、Jenkins の有効化を行うことができる



Testjbosseap60 @ <http://testjbosseap60-takahiko.rhcloud.com/>

Cartridges ADD CARTRIDGE +

	STATUS	GEARS
JBoss Enterprise Application Platform 6.0	IDLE	1 SMALL

GIT REPOSITORY

`ssh://69cc9acc698841df9725857d3d1a22fd@testjbosseap60-takahiko.rhcloud.com/~/.git/testjbosseap60_gi`

WANT TO LOG IN TO YOUR APPLICATION?

CLI (rhc)

様々なプラットフォーム上で、アプリケーションをフルコントロールしコーディングできる。コーディングとデバッグ、アプリケーション管理に向いている。

インストール方法

Windows クライアント (Windows XP, 7, 8) の場合のインストール方法を記述する。

その他の OS も、ほぼ同じ手順である。

事前準備として、Ruby 及び Git をインストールする。

Ruby のインストール

- 1 以下のサイトから、最新の Ruby インストーラ パッケージをダウンロードする。
<http://rubyinstaller.org/>
- 2 インストーラを実行し、ウィザードの手順通りにセットアップを進める。
その際、「パスに Ruby 実行モジュールを追加する」のオプションが選択されていることを確認する。

Git のインストール

Ruby のインストール後、以下の手順で Git Version Control をインストールする。

- 1 以下のサイトから、最新の Git for Windows をダウンロードする。
<http://msysgit.github.com/>
- 2 インストーラを実行し、ウィザードの手順通りにセットアップを進める。
その際、「Run Git from the Windows Command Prompt」を選択する。
- 3 次に、「Checkout windows-style, commit Unix-style line endings」を選択する。

CLI のインストール

- 1 Windows コマンド プロンプトを開く
- 2 以下のコマンドを実行し、クライアントツールをインストールする。
`$ gem install rhc`
- 3 インストールが完了したら、次のコマンドを実行する。
`$ rhc setup`
- 4 OpenShift へのログイン ID 及びパスワードや SSH キーの情報など入力を求められるので、入力していく。

利用方法

OpenShift の CLI では、次のような操作が可能である。

- SSH キーの管理
- Domain (アプリケーション ドメイン) の管理
- ユーザ情報 (各アプリケーションや DB の URL 情報等) の表示
- アプリケーションの作成

次のコマンドで、自動スケールアウト可能なアプリケーションが作成可能。

```
$ rhc app create -a [AppName] -t [AppType] -s
```

なお、自動スケールアウト無効なアプリケーションは手動にてスケールする必要がある。

- カスタムドメイン名の付与
- デバッグモードの有効化
トラブルシューティング時に詳細に出力させる。
- アプリケーションの編集と配置 (Hot Deploy)
git push すると、OpenShift は停止、ビルド、配置、アプリケーションの再起動を行う。
- カートリッジの追加
スケール可能なアプリケーションでは、MySQL 5.1、MongoDB、PostgreSQL、Jenkins Client 1.4 のみが追加できる。
後述する SSH 接続により、各カートリッジの操作が可能。
MongoDB はブラウザで操作可能なインタフェースが提供される。
- アプリケーション スナップショットの作成
- アプリケーションの管理
状態表示、スタート、ストップ、再起動、再読込、削除が可能。
- アプリケーションへのシェル接続
次のコマンドで、各アプリケーションへ SSH 接続できる。
\$ssh UUID@appname-namespace.domain
なお、Windows の場合、後述する Tera Term 等のツールから接続できる。

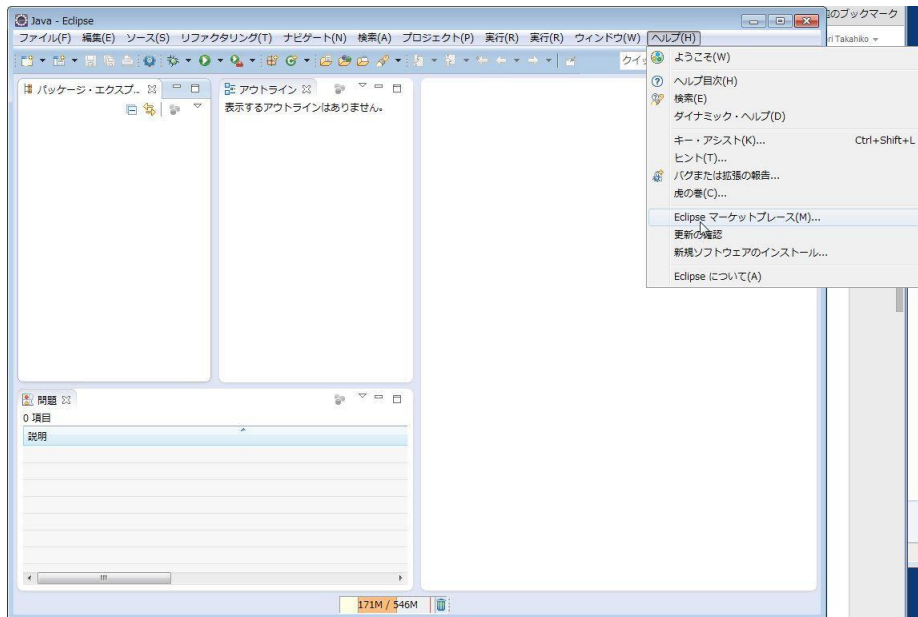
GUI (IDE)

Eclipse 開発環境とのシームレスな統合が可能。
コーディングとデバッグ、アプリケーション管理に向いている。

インストール方法

ここでは、Eclipse 4.2 (Juno) へ JBoss Developer Studio プラグインをインストールする手順を示す。

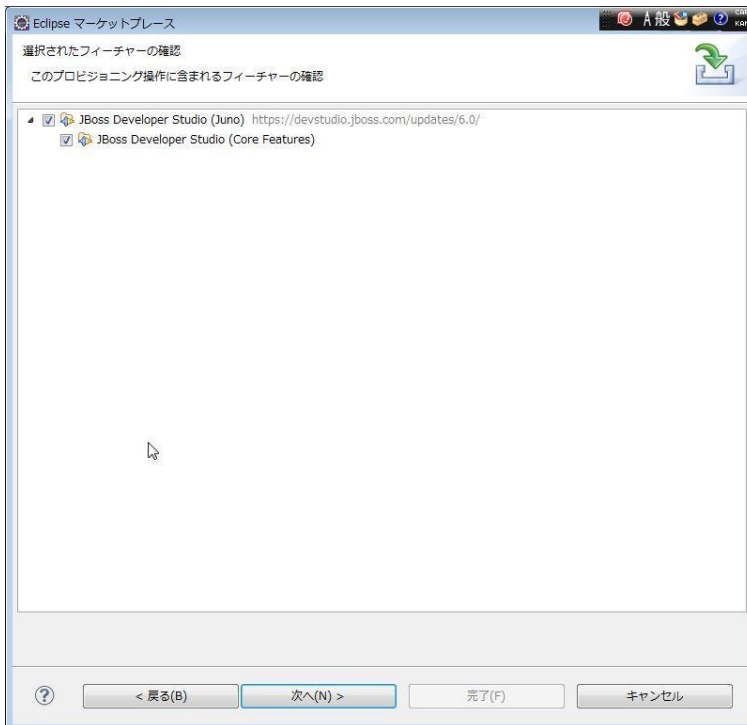
- 1 Eclipse を起動し、メニュー[ヘルプ(H)]-[Eclipse マーケットプレイス(M)..]を選択する。



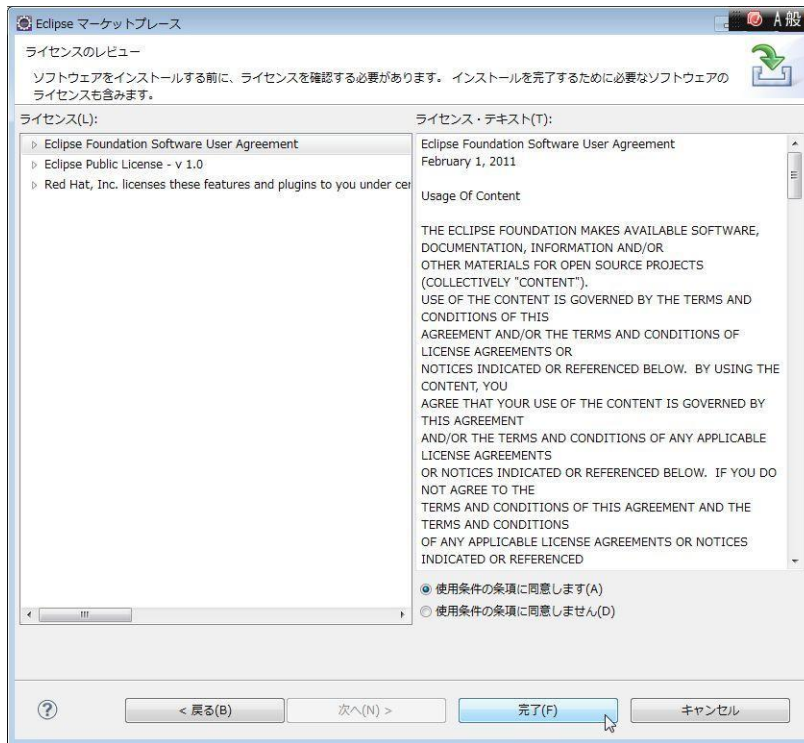
- 2 検索に”JBoss”と入力し、Enter キーを押す。
リストの先頭に表示される、”JBoss Developer Studio (Juno)”の[インストール]ボタンを押す。



- 3 選択されたフィーチャーをそのままにして、[次へ(N)>]ボタンを押す。



- 4 ライセンスの”使用条件の条項に同意します”を選択し、[実行(F)]ボタンを押す。



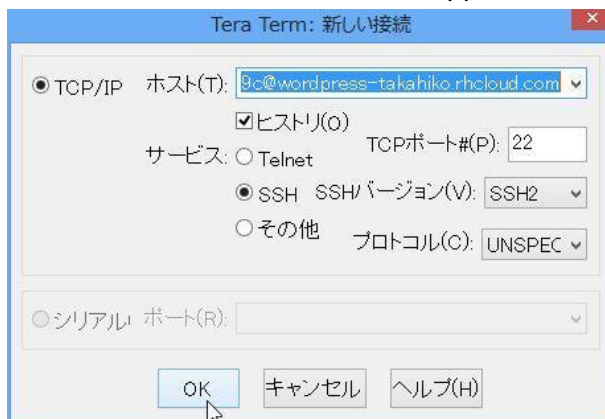
- 5 インストールが開始され、途中で署名のないモジュールをインストールするか聞かれるので[OK]を押す。

SSH 接続

UUID とアプリケーション名を指定することで、SSH クライアントから接続することができる。

利用方法

- 1 Tera Term で、ホストに UUID@appname-namespace.domain で接続する。



2 接続に必要な秘密キーを指定する

SSH認証

ログイン中: wordpress-takahiko@rhcloud.com

認証が必要です。

ユーザ名(U): d56c36f8656a422ebcb1b4f3feba839c

パスワード(P):

パスワードをメモリ上に記憶する(M)

エージェント転送する(O)

プレインテキストを使う(L)

RSA/DSA鍵を使う

秘密鍵(K): rs#localadmin#ssh#id_rsa

rhosts(SSH1)を使う

ローカルのユーザ名(U):

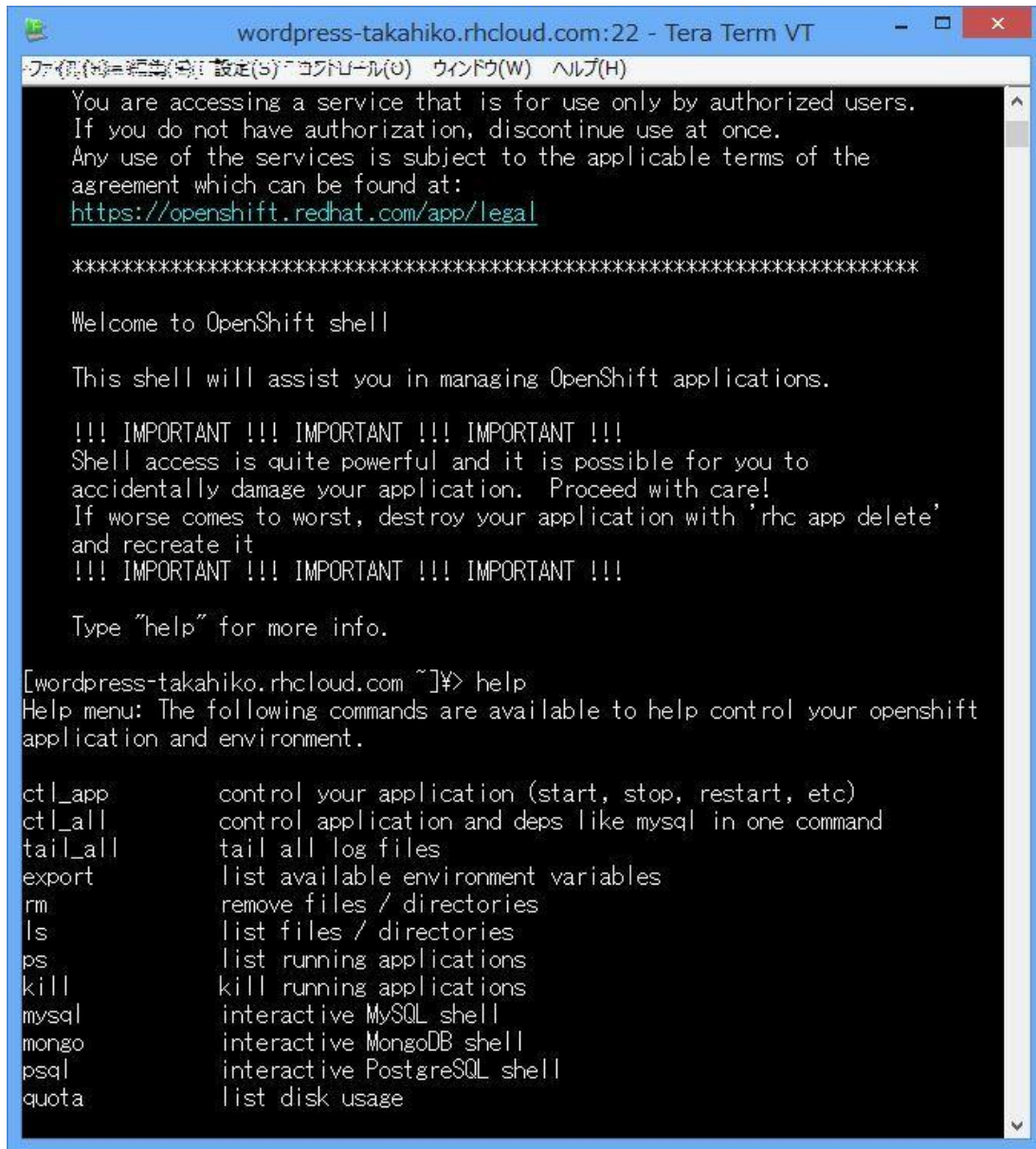
ホスト鍵(F):

チャレンジレスポンス認証を使う(キーボードインタラクティブ)(C)

Pageantを使う

OK 接続断(D)

3 下記の通り、接続できる。



```
wordpress-takahiko.rhcloud.com:22 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(C) ウィンドウ(W) ヘルプ(H)
You are accessing a service that is for use only by authorized users.
If you do not have authorization, discontinue use at once.
Any use of the services is subject to the applicable terms of the
agreement which can be found at:
https://openshift.redhat.com/app/legal

*****

Welcome to OpenShift shell

This shell will assist you in managing OpenShift applications.

!!! IMPORTANT !!! IMPORTANT !!! IMPORTANT !!!
Shell access is quite powerful and it is possible for you to
accidentally damage your application. Proceed with care!
If worse comes to worst, destroy your application with 'rhc app delete'
and recreate it
!!! IMPORTANT !!! IMPORTANT !!! IMPORTANT !!!

Type "help" for more info.

[wordpress-takahiko.rhcloud.com ~]# help
Help menu: The following commands are available to help control your openshift
application and environment.

ctl_app      control your application (start, stop, restart, etc)
ctl_all      control application and deps like mysql in one command
tail_all     tail all log files
export       list available environment variables
rm           remove files / directories
ls           list files / directories
ps           list running applications
kill         kill running applications
mysql        interactive MySQL shell
mongo        interactive MongoDB shell
psql         interactive PostgreSQL shell
quota        list disk usage
```


ローカル実行環境（OpenShift Origin）

OpenShift Origin というプロジェクト名で、オープンソースのみで作成された実行環境がある。

その起動可能なディスクイメージが LiveCD として公開されている。

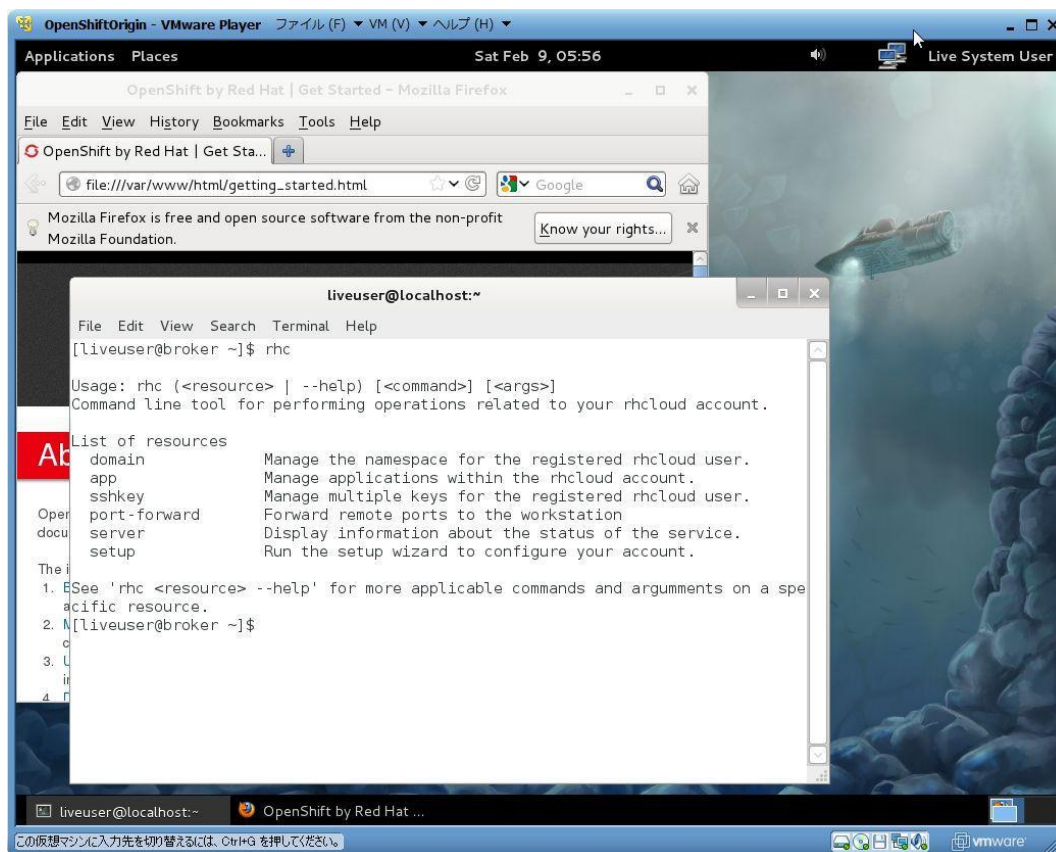
<https://openshift.redhat.com/community/open-source/download-origin>

インストール方法

- 1 上記の URL から ISO ファイルを入手する。
- 2 VMware Player などの仮想マシン実行環境で適当な仮想マシンを作成し、1.で入手した ISO をマウントし、そこから起動する。

利用方法

インストール方法で示した手順で LiveCD が立ち上がると、下記のイメージのような、ブラウザとターミナルが自動起動し、CLI などのツールがすべてインストールした状態ですぐに利用可能となる。



開発～運用

デプロイ方法

Web コンソールや CLI からの `rhc app create` コマンドでは、アプリケーションの開始点を作成します。アプリケーションのファイルのための git リポジトリからファイルを取得し、開発を行い、プッシュすることで、OpenShift は停止し、ビルドを行い、アプリケーションを配置して、アプリケーションの再起動まで行います。これをホット デプロイと呼びます。

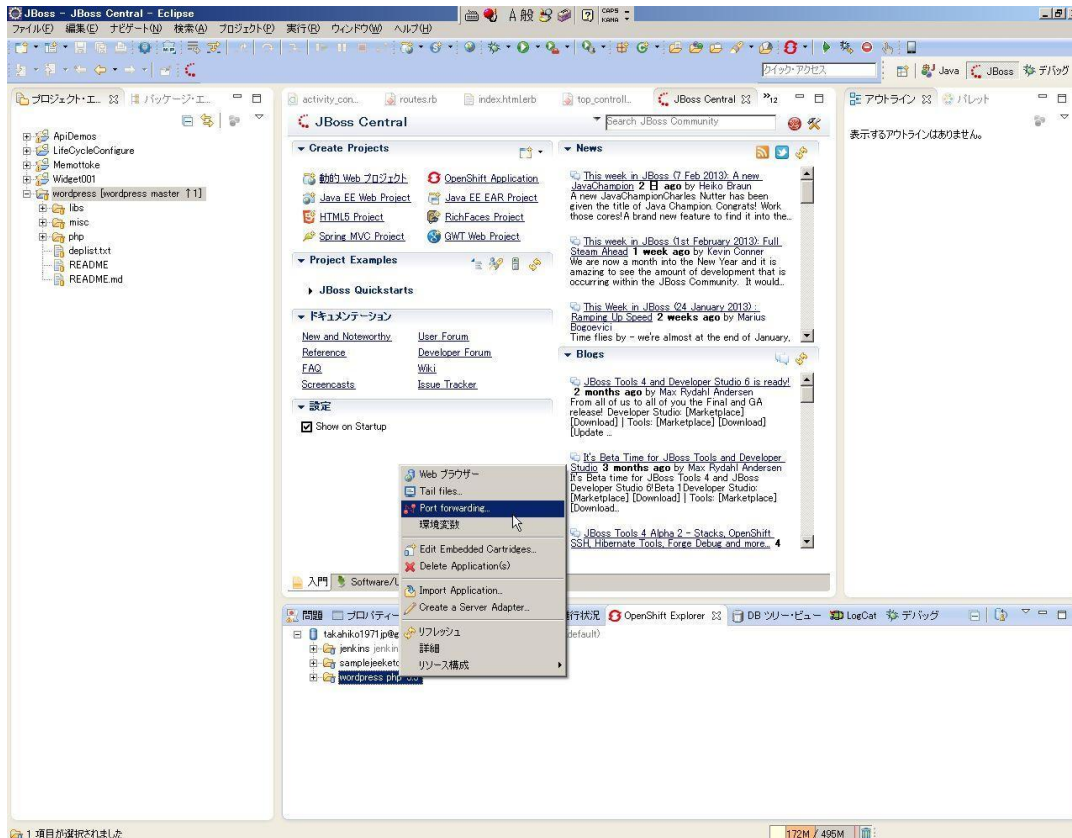
なお、Jenkins を用いることで、コンパイルエラー時にアプリケーションの停止を回避することができます。

データ

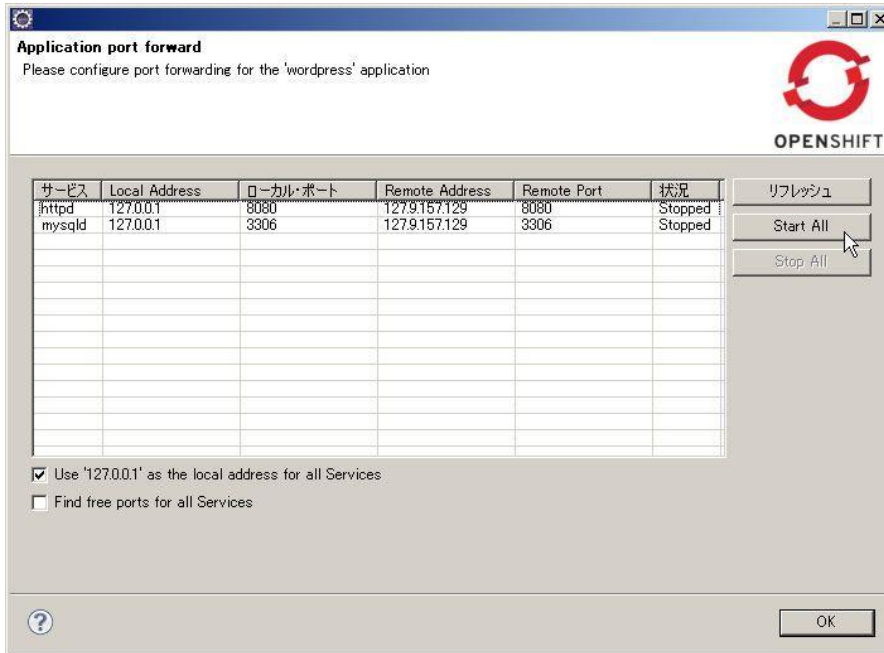
OpenShift では、phpMyAdmin カートリッジが用意されており、Web アプリケーションから MySQL を管理することができます。また、SSH 接続からコマンドラインでデータ操作することも可能である。

ここでは、IDE を利用してポートフォワードの設定を行い、データベースを操作する手順を示す。

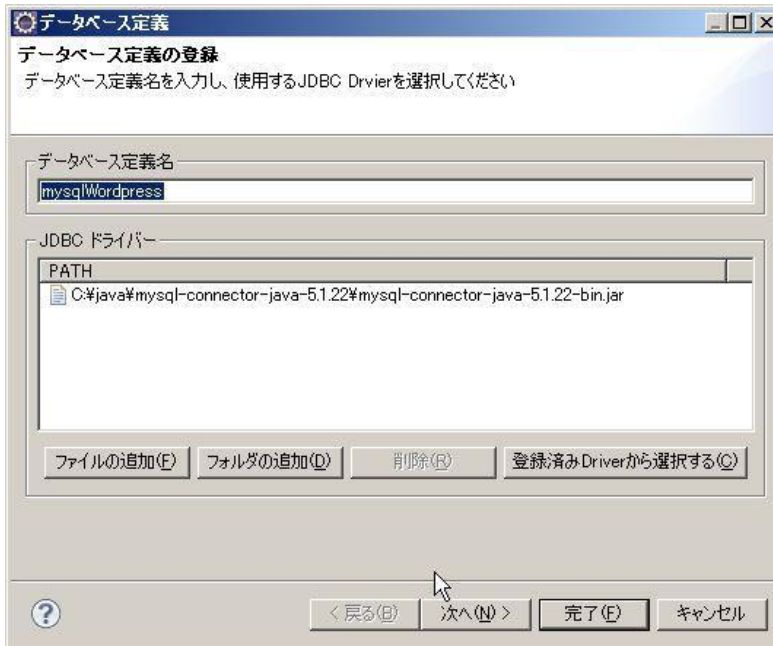
- 1 OpenShift Explorer のアプリケーションを右クリックし、[Port forwarding ...]を選択する。



2 ここでは, [Start All]ボタンを押す。



3 Eclipse プラグインの DB ツリー・ビューから, 新しい接続を作成する。
MySQL への接続に必要な JDBC ドライバーが無い場合, Oracle 社のホームページから
入手する。



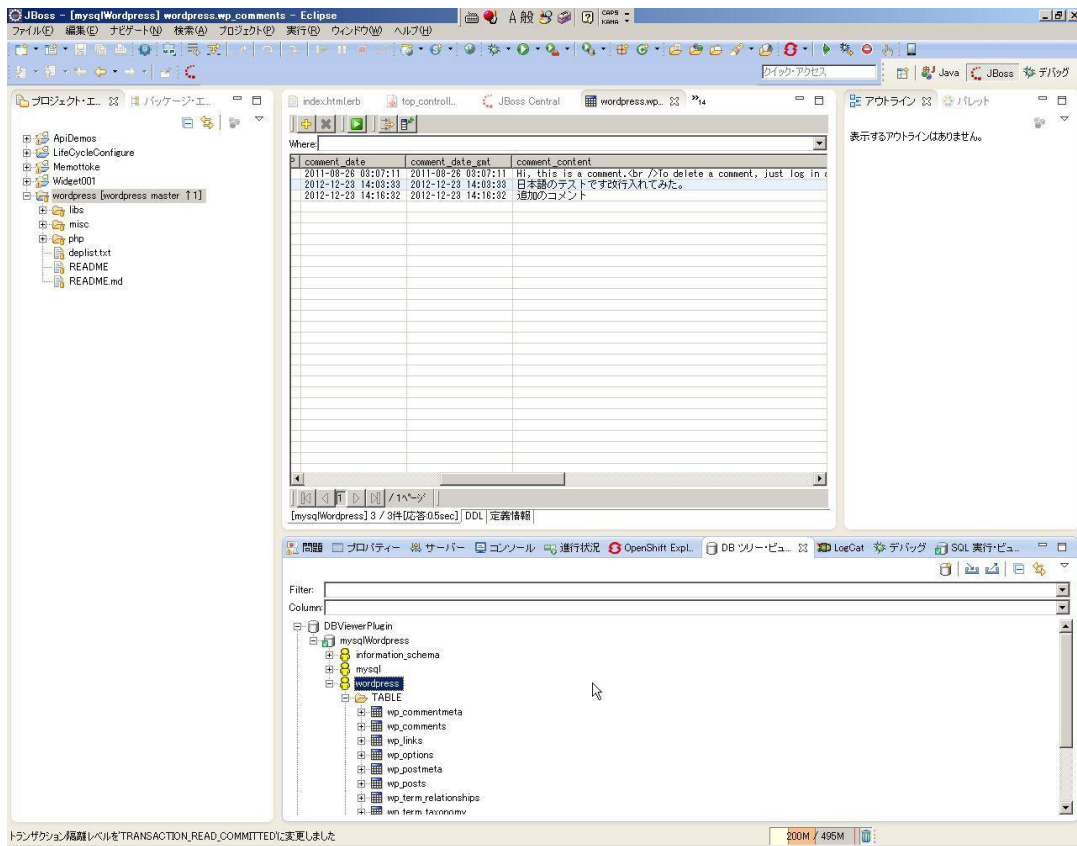
4 ポートフォワードされたローカルポートを指定する。



5 接続するスキーマを選択する。



6 GUIからデータ入力やエクスポート、SQL文の発行など自由に操作できる。



The screenshot shows the Eclipse IDE interface with a database viewer. The main window displays a table of comments with the following data:

comment_date	comment_date_gmt	comment_content
2011-09-28 08:07:11	2011-09-28 08:07:11	Hi, this is a comment. Click /?to_delete a comment, just log in
2012-12-28 14:08:38	2012-12-28 14:08:38	日本語のテストです。再入力してみた。
2012-12-28 14:16:32	2012-12-28 14:16:32	追加のコメント

The bottom part of the screenshot shows a tree view of the database schema, including tables like wp_comments, wp_links, and wp_posts.

アプリ

アプリケーションの作成は Web コンソール, CLI, IDE, すべてのクライアントから行うことができる。更新は, CLI もしくは IDE から行うことができる。

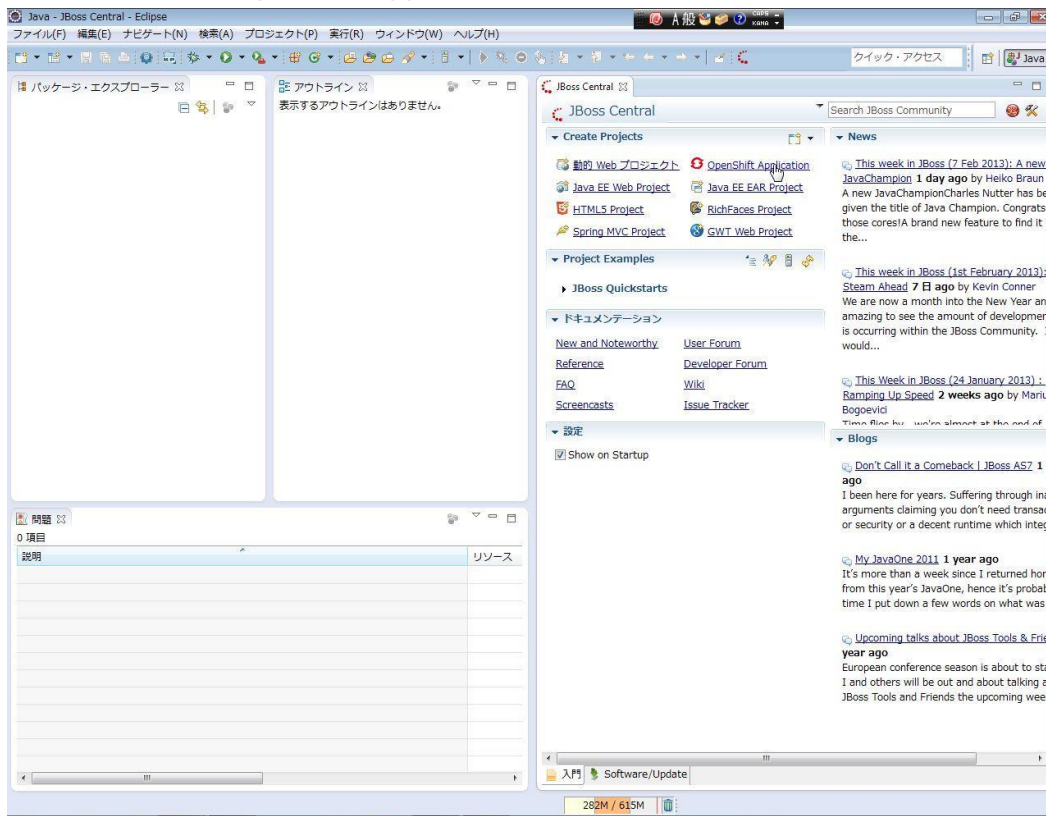
ここでは, CLI から PHP のスケール可能なアプリケーション WP を作成し, MySQL カートリッジを追加したものを, IDE から取得し, 更新して配置するまでの手順を示す。

新規

- 1 コマンドプロンプトから以下を実行し, php でスケール可能なアプリケーション (名前を wp とする) を作成する。
> rhc app create -a wp -t php-5.3 -s
- 2 アプリケーション wp へ MySQL カートリッジを追加する。
> rhc app cartridge add -a wp -c mysql-5.1

更新

- 1 JBoss Central の OpenShift Application リンクを選択する。



2 OpenShift のアカウント情報を入力する。

The screenshot shows the 'New OpenShift Application' dialog box in the 'Sign in to OpenShift' step. The window title is 'New OpenShift Application'. The main text says 'Sign in to OpenShift' and 'Please provide your OpenShift credentials.' The OpenShift logo is in the top right corner. Below the text, there is a link: 'If you do not have an account on OpenShift, please sign up [here](#).' The form contains the following fields and controls:

- 接続: A dropdown menu with '<New Connection>' selected.
- Use default server
- サーバー(S): A text field containing 'https://openshift.redhat.com'.
- ユーザー名(U): A text field containing 'itascp@gmail.com'.
- パスワード(P): A password field with 10 dots.
- パスワードの保管 (セキュア・ストレージ・ログインのトリガーとなる)

At the bottom, there are four buttons: a help icon (?), '< 戻る(B)', '次へ(N) >', '完了(F)', and 'キャンセル'.

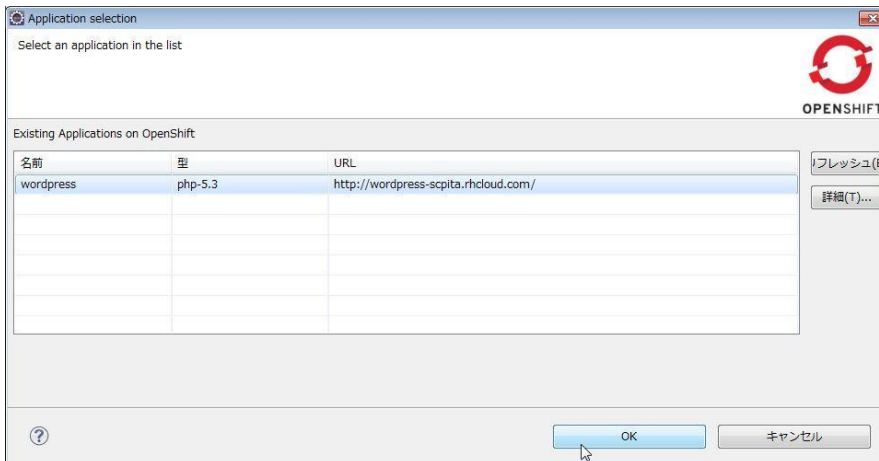
3 [Use existing application:]にチェックを入れ, [参照...]ボタンを押す。

The screenshot shows the 'New OpenShift Application' dialog box in the 'Setup OpenShift Application' step. The window title is 'New OpenShift Application'. The main text says 'Setup OpenShift Application' and 'Please select an existing OpenShift application'. The OpenShift logo is in the top right corner. Below the text, there is a link: 'If you do not have an account on OpenShift, please sign up [here](#).' The form contains the following fields and controls:

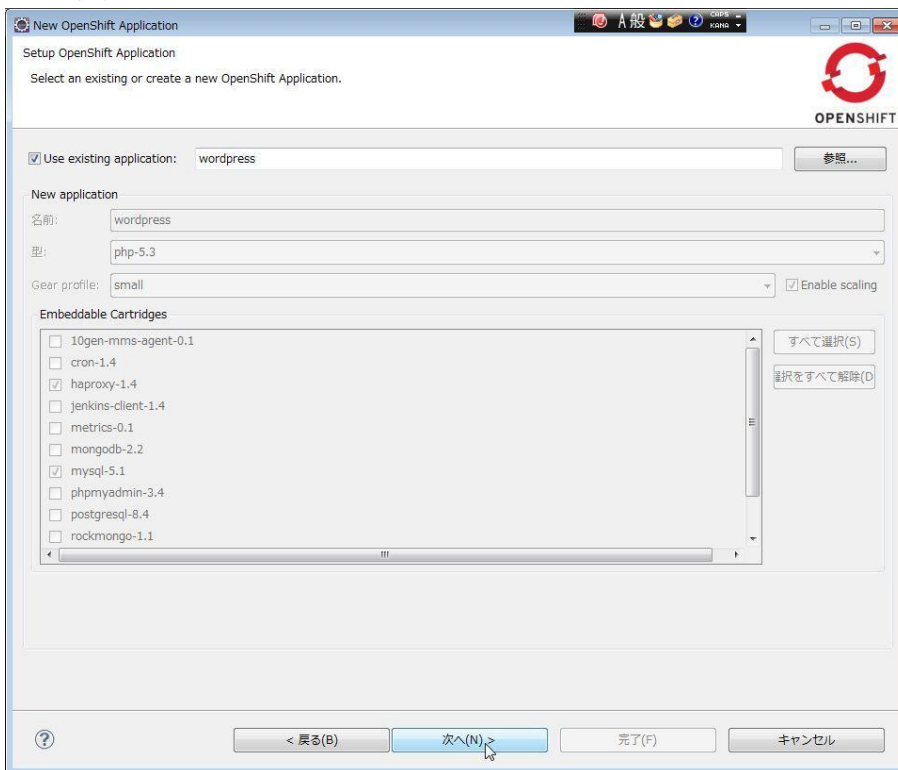
- Use existing application: A text field with a dropdown arrow and a '参照...' button to its right.
- New application section:
 - 名前: A text field.
 - 型: A dropdown menu.
 - Gear profile: A dropdown menu and Enable scaling.
- Embeddable Cartridges: A list of cartridges with checkboxes:
 - 10gen-mms-agent-0.1
 - cron-1.4
 - haproxy-1.4
 - jenkins-client-1.4
 - metrics-0.1
 - mongodb-2.2
 - mysql-5.1
 - phpmyadmin-3.4
 - postgresql-8.4
 - rockmongo-1.1

At the bottom, there are four buttons: a help icon (?), '< 戻る(B)', '次へ(N) >', '完了(F)', and 'キャンセル'.

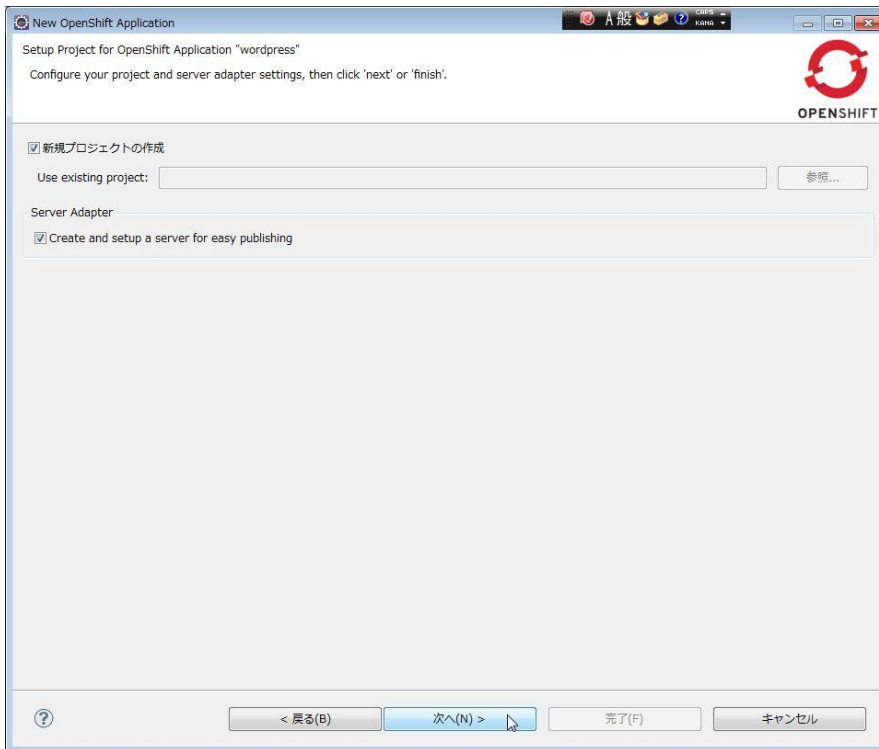
4 アプリケーションを選択して, [OK]を押す。



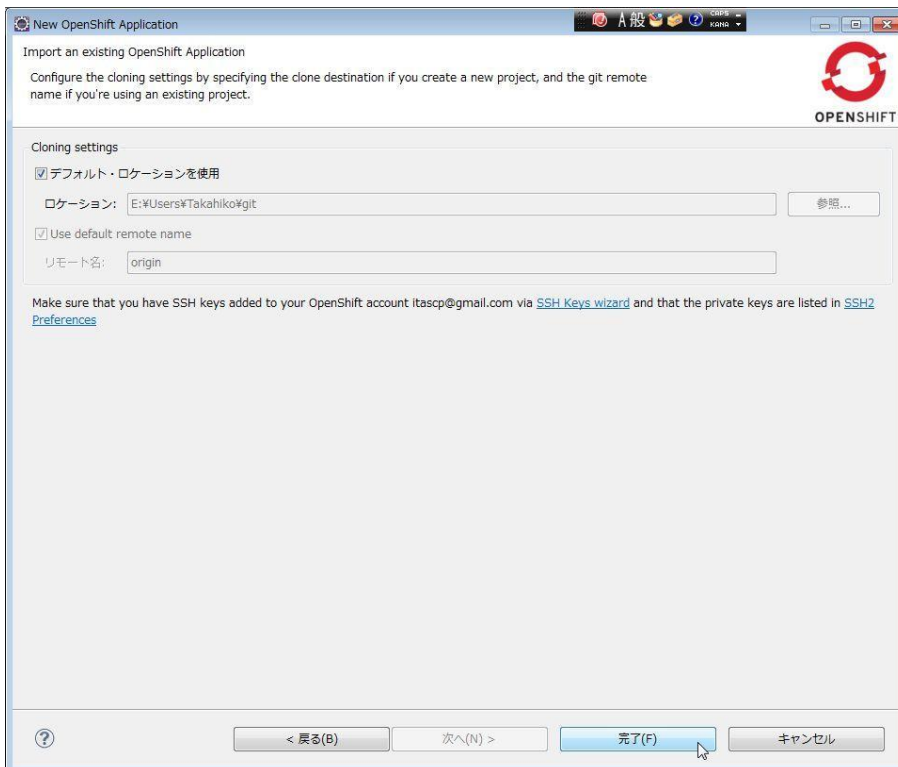
5 [次へ(N)>]ボタンを押す。



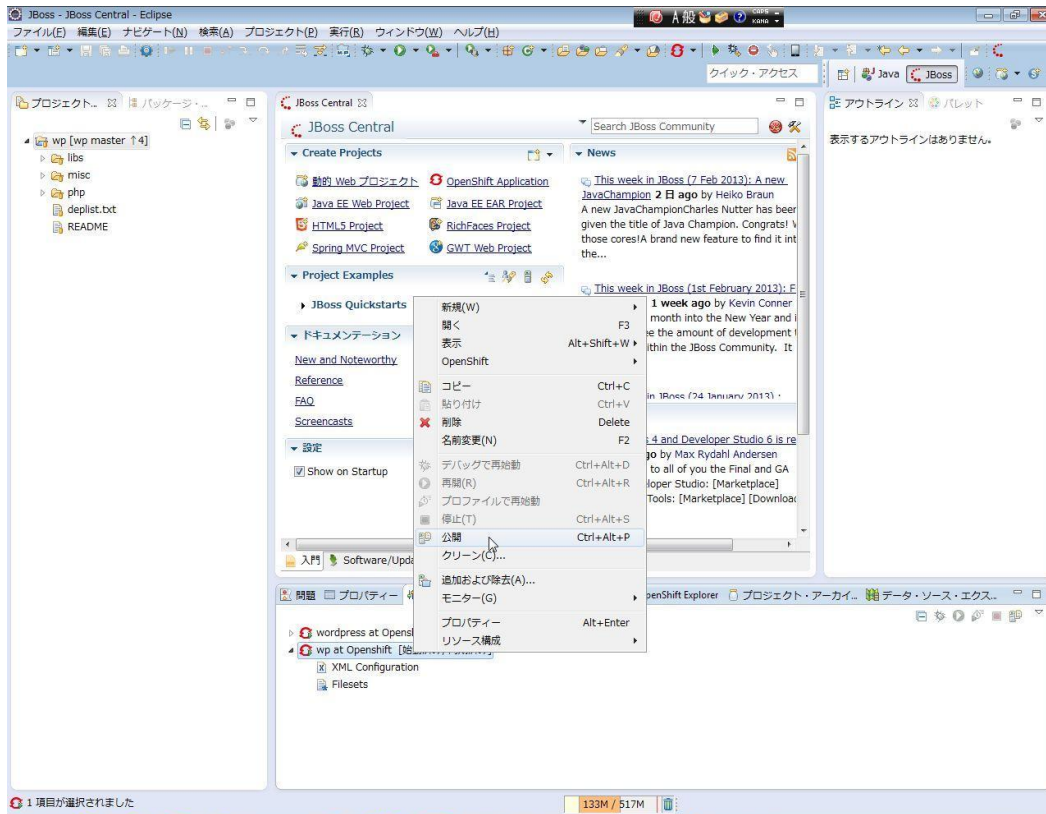
6 そのまま[次へ(N)>]ボタンを押す。



7 ロケーションのフォルダが無い場合、エラーが表示されるので、フォルダを作成してから一度、[<戻る(B)]ボタンを押してから戻り、[完了(F)]ボタンを押す。



- 8 PHP フォルダにてアプリケーションを開発する。
ここでは、OpenShift の Git リポジトリから WordPress のコードをクローンしたものをコピーした。
- 9 プロジェクトワークスペースからアプリケーションを右クリックし、メニュー[チーム]-[コミット]にて Git リポジトリにコミットする。
- 10 サーバー ウィンドウからアプリケーションを右クリックし、[公開]を選択すると、アプリケーションが自動デプロイされる。



テスト・デバッグ

開発中のテスト及びデバッグは、IDE を用いることで、ローカルサーバにて行うことができる。

リモート

OpenShift 上で実行しているアプリケーションをデバッグする方法を以下に示す。

サーバログのリアルタイム監視

- 1 IDE のサーバウィンドウからアプリケーションを選択し、メニュー[OpenShift]-[Tail files...]を選択する。
- 2 tail コマンドを入力するダイアログが表示されるので、OK ボタンを押す。
- 3 コンソール ウィンドウにサーバログがリアルタイムで出力される。

ログ解析

ログは後述するバックアップ処理にて取得できる。これを用いてローカルのテキストファイルとしてログ解析を行うことが可能である。

テスト

負荷テスト

ターゲット

ステージング

OpenShift には、ステージングのための仕組みは用意されていない。そのため、複数アカウントを用いてステージング環境を OpenShift 上に用意するか、LiveCD 等を用いてローカルもしくはイントラ環境にステージング環境を用意する必要がある。

アクセス制限について

SELinux の機能や HAProxy によって外部からの直接アクセスをコントロール可能である。

プロダクション

監視

アプリケーション

死活監視

HAProxy のレポート機能によって死活監視を行うことができる。
以下に、そのイメージを示す。

The screenshot shows the HAProxy status page for version 1.4.22. It includes a 'General process information' section with system limits and a 'stats' table. The 'stats' table shows the following data:

	Queue			Session rate			Sessions				Bytes		Denied		Errors			Warnings			Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Downtme	Thrtle		
Frontend	1	1	-	1	1	3 000	2	0	0	0	191	1 219	0	0	0	0	0	0	0	0	OPEN										
Backend	0	0		0	0		0	0	3 000	0	0	191	1 219	0	0							3m12s	UP		0	0	0	0			

自動復旧

しばらくアクセスが無かったり、アクセス頻度が低い場合、URL へのアクセスがあることでアプリケーションが自動起動する。

ただ、何らかの不具合が原因の場合、自動復旧しないため、git push することで自動配置が行われるので、それを利用して復旧させることが可能である。

サービス

死活監視

自動復旧

メンテナンス

アプリケーション

OpenShift は Git リポジトリにプッシュすると同時にデプロイされる仕組みのため、アプリケーションをメンテナンス可能にするためには、開発用アカウントを用意し、さらに Jenkins を用いてビルドエラーによる停止を回避するのが良いだろう。

サービス

バックアップ

次のコマンドによって、OpenShift アプリケーションのバックアップを取得することができる。

```
> rhc app snapshot save -a Application_Name
```

このコマンドで、gzip された tar ファイルが作成される。そこにはアプリケーションだけでなく、ログやその他のファイルが含まれる。実際の運用では、`--filepath` オプションでパスとファイル名を指定した方が良いだろう。

リストア

次のコマンドで、取得したスナップショット（バックアップ）をサーバへリストアできる。

```
> rhc app snapushot restore
```

これによって、Git リポジトリやアプリケーション データ ディレクトリやログファイルがリストアされる。リストアが完了すると、git push したときと同じように、OpenShift は配置スクリプトを実行し、配置処理が実行される。

PaaS 実績

コミュニティの活動

日本における OpenShift コミュニティはまだ発足していない模様。

米国 RedHat 社の公式ページにコミュニティサイトがあり、フォーラムでは数時間単位で活発に書き込みがされている。

(<https://openshift.redhat.com/community/>)

情報量

日本国内では、個人のブログで試行した結果や英語マニュアルを一部翻訳したものがいくつか公開されている程度である。

国外でも、ほとんどの情報は OpenShift の公式ページ内にある。ドキュメントも一通りはあるが、概要レベルであり詳細な情報はまだない。オープンソースの組合せであり、SSH 接続で各フォルダ内のファイルを閲覧できるので、そこから紐解く必要がある。

サービスの更新間隔

不定期だが、月に何度も更新されることもある。

利用企業、サービスなど

2012 年 11 月 27 日に、エンタープライズ向けの OpenShift Enterprise を米国向けに提供することを発表したばかりで、まだ商用サービスとしての実績は乏しい。

ビジネス

費用

ランニングコスト

前述した PaaS 表を参照。

オプション

前述した PaaS 表を参照。

中立性

ライセンス

Apache License, Version 2.0 でソースを公開している。

ベンダー依存

RedHat 社に依存していると言えなくはないが、オープンソースである点、LiveCD のような完全オープンソースな実装も公開されている点などから、依存度は低いと考える。

ポータビリティ

設定ファイルに OpenShift 独自の環境変数を入力する部分があるが、それ以外では OpenShift に依存したアプリケーション ロジックは無さそうであり、ポータビリティは高いと評価する。

おわりに

OpenShift はまだ商用サービスが開始されたばかりの PaaS であり、実績は乏しい。しかし、RedHat 社がこれまで築いてきたオープンソースへの取り組みノウハウや、Java EE6 対応、充実した各種 IDE との連携機能、MySQL や PostgreSQL などの RDBMS 対応などから、エンタープライズをターゲットとした PaaS として期待できる。

以上