Appendix B - Deploying to OpenShift

What Will You Learn Here?

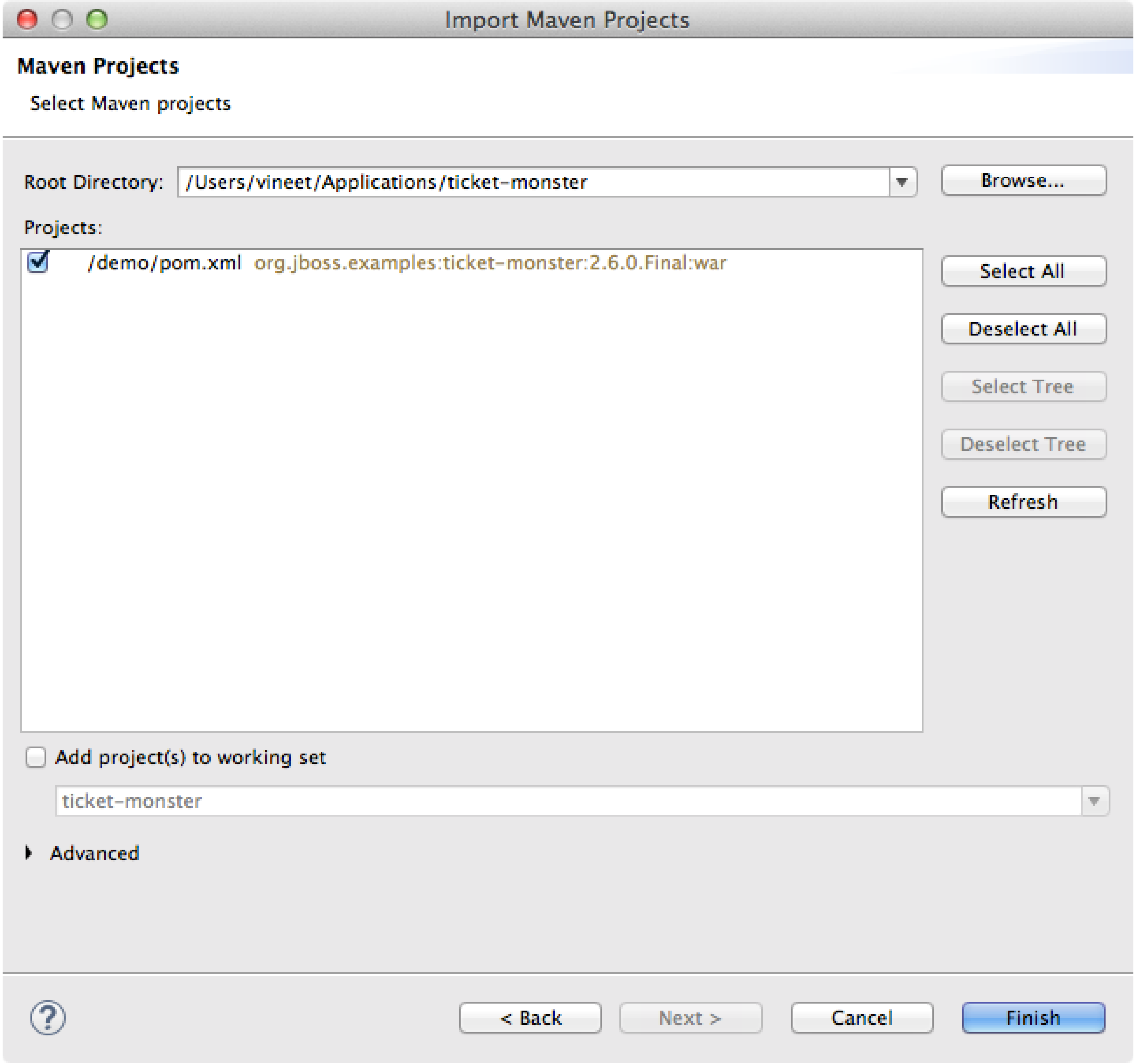
This appendix demonstrates how to import, develop and deploy the TicketMonster example using JBoss Developer Studio:

- Obtain and import the TicketMonster example source code
- Deploy the application to OpenShift Online with OpenShift Tools

Import the Project source code

Once the TicketMonster source code is obtained and unpackaged, you must import it into JBoss Developer Studio, as detailed in the procedure below. TicketMonster is a Maven-based project so a specific Import Maven Project wizard is used for the import.

1. Click **File**→**Import** to open the **Import** wizard.
2. Expand **Maven**, select **Existing Maven Projects** and click **Next**.
3. In the **Root Directory** field, enter the path to the TicketMonster source code. Alternatively, click **Browse** to navigate to the source code location. The **Import Maven Project** wizard recursively searches the path for a **pom.xml** file. The **pom.xml** file identifies the project as a Maven project. The file is listed under **Projects** once it is found.



Maven Projects

Select Maven projects

Root Directory: /Users/vineet/Applications/ticket-monster

Projects:

☑ /demo/pom.xml org.jboss.examples:ticket-monster:2.6.0.Final:war

□ Add project(s) to working set

ticket-monster

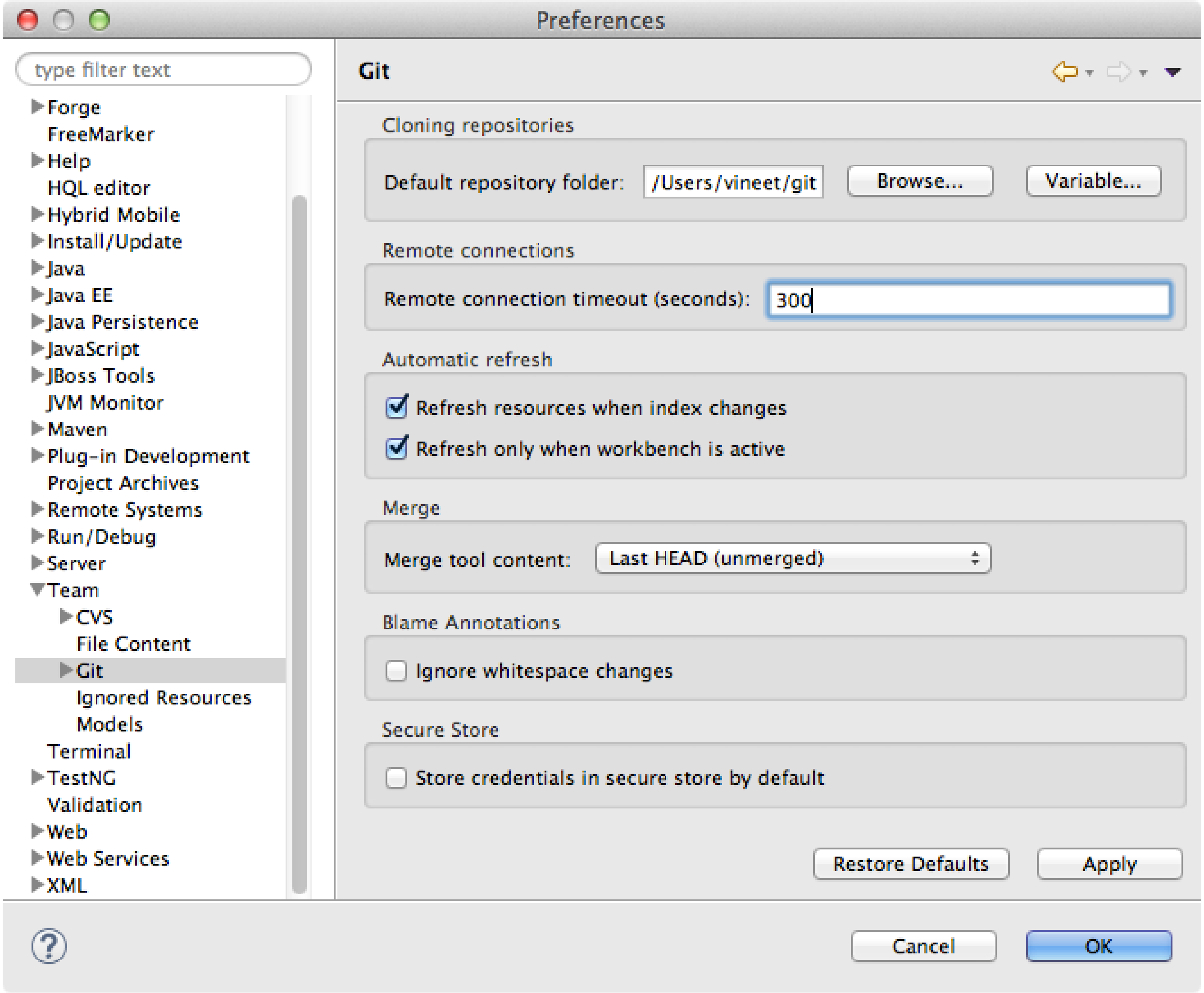
Figure 1. pom.xml File Listed in the Projects Pane

4. Click **Finish**. When the import process is complete, the project is listed in the **Project Explorer** view.

Pre-requisites

We will be pushing the TicketMonster sources to a git repository on OpenShift, where the application would be built and deployed. The build on OpenShift, and hence the `git push` can timeout, since Maven dependencies will have to be fetched from the Red Hat Enterprise Maven repository or other repositories.

We'll get around this drawback by configuring JBoss Developer Studio to not time out sooner. To do do, set the Git connection timeout to 300 seconds. Click **Window** → **Preferences**, expand **Team** and select **Git**. In the **Remote connection timeout (seconds)** field, type **300** and click **Apply** and click **OK**.



Git

Cloning repositories

Default repository folder: /Users/vineet/git

Remote connections

Remote connection timeout (seconds): 300

Automatic refresh

☑ Refresh resources when index changes

☑ Refresh only when workbench is active

Merge

Merge tool content: Last HEAD (unmerged)

Blame Annotations

□ Ignore whitespace changes

Secure Store

□ Store credentials in secure store by default

Figure 2. Modify the git remote connection timeout

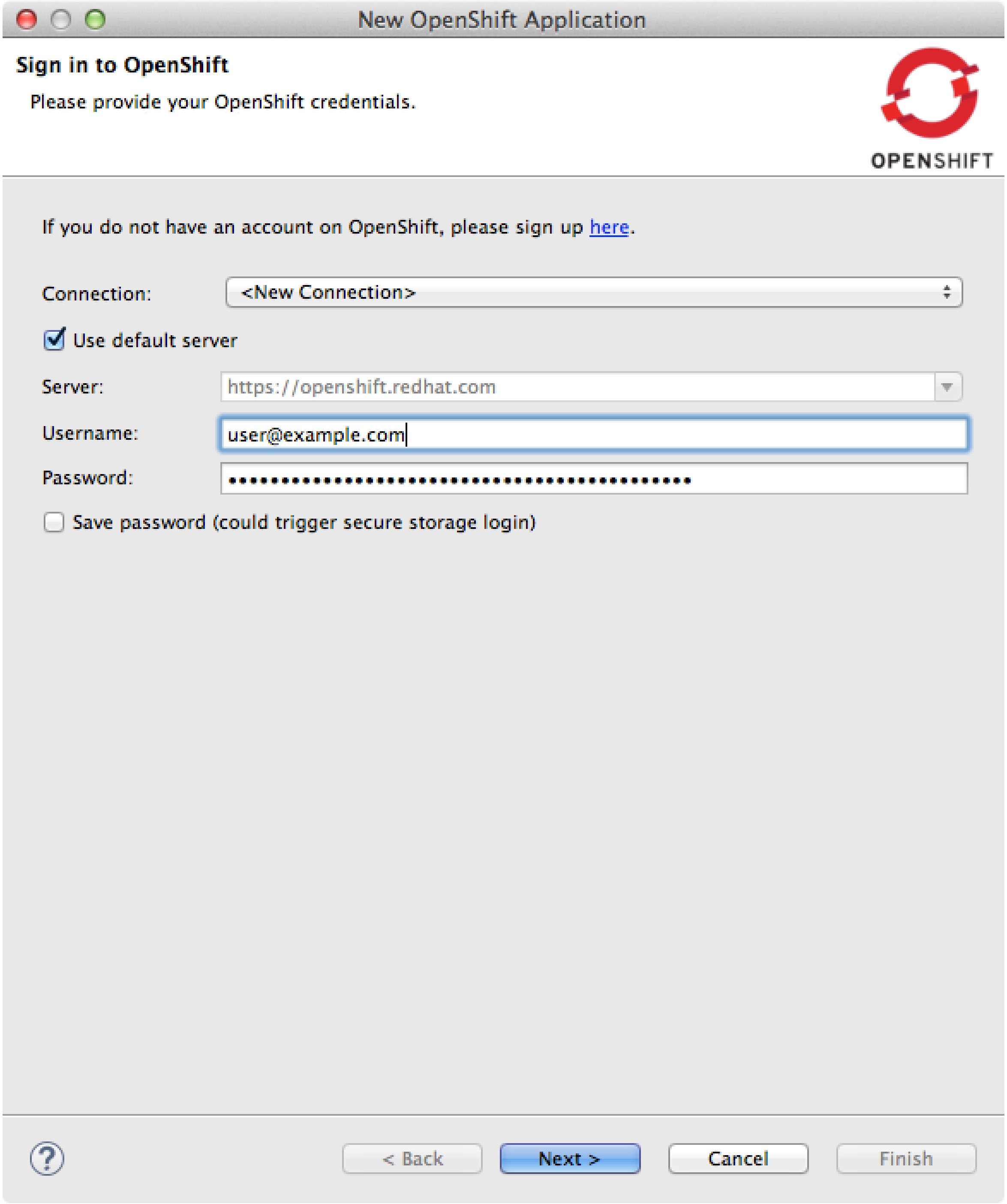
Deploying to OpenShift using JBoss Developer Studio

To deploy TicketMonster to OpenShift Online, you must create a new OpenShift Online application based on the existing workspace project using OpenShift Tools, as detailed in the procedure below.

NOTE This procedure documents the deploying process for first-time OpenShift Online users. This includes one-time steps, such as signing up for an OpenShift Online account, creating an OpenShift Online domain and uploading SSH keys. If you have previously used OpenShift Online and OpenShift Tools, omit the one-time steps as appropriate.

1. In JBoss Central, under Start from scratch, click OpenShift Application.
2. Click the *Please sign up here* link to create an OpenShift Online account and follow the instructions on the OpenShift web page displayed in your default system web browser. Once you have completed the sign-up process, restart the New OpenShift Application wizard.
3. Complete the fields about your OpenShift Online account as follows:

- From the Connection list, select New Connection.
- Ensure the Use default server check box is selected.
- In the Username and Password fields, type your account credentials.



Sign in to OpenShift

Please provide your OpenShift credentials.

If you do not have an account on OpenShift, please sign up here.

Connection: <New Connection>

☑ Use default server

Server: https://openshift.redhat.com

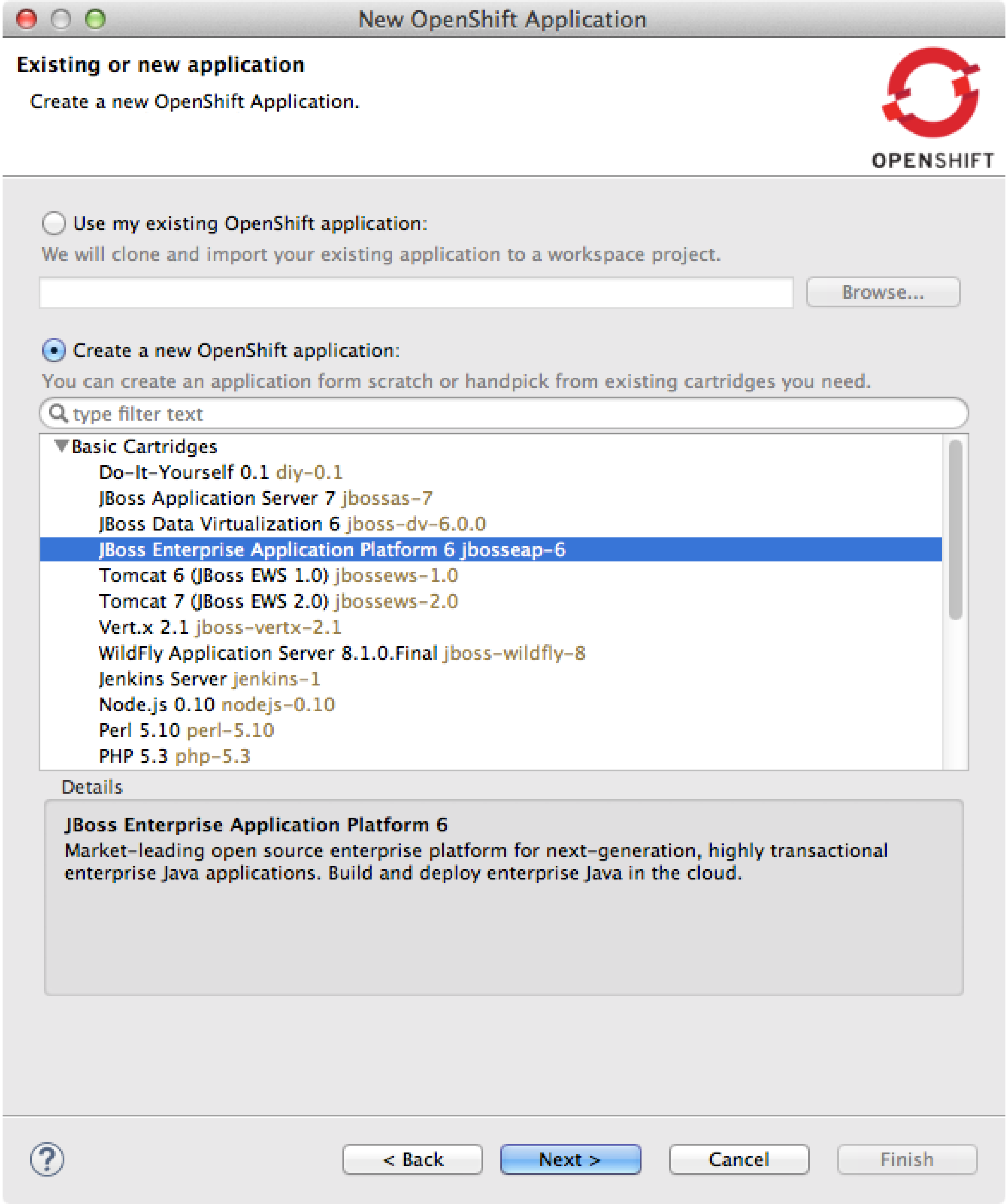
Username: user@example.com

Password: ••

□ Save password (could trigger secure storage login)

Figure 3. Completed Username and Password Fields

4. Click Next.
5. In the **Domain Name** field, type a name for your new OpenShift Online domain and click Finish. The provided domain name must be unique across all domains on OpenShift Online; if it is not unique, you will be instructed to provide a unique domain name.
6. From the Type list, select JBoss Enterprise Application Platform 6 (jbosseap-6).



Existing or new application

Create a new OpenShift Application.

○ Use my existing OpenShift application:

We will clone and import your existing application to a workspace project.

● Create a new OpenShift application:

You can create an application form scratch or handpick from existing cartridges you need.

type filter text

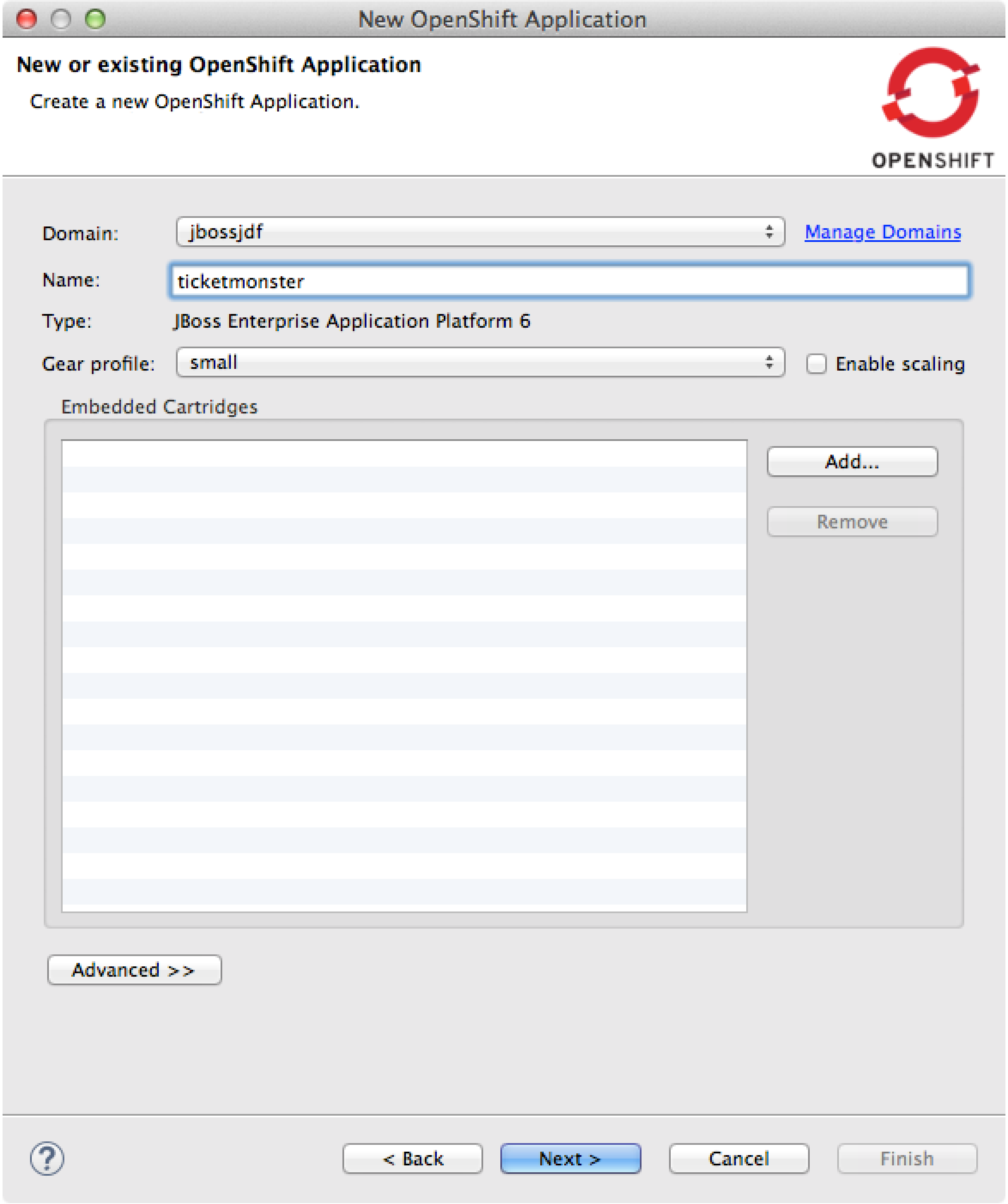
Basic Cartridges
- Do-It-Yourself 0.1 diy-0.1
- JBoss Application Server 7 jbossas-7
- JBoss Data Virtualization 6 jboss-dv-6.0.0
- JBoss Enterprise Application Platform 6 jbosseap-6
- Tomcat 6 (JBoss EWS 1.0) jbossews-1.0
- Tomcat 7 (JBoss EWS 2.0) jbossews-2.0
- Vert.x 2.1 jboss-vertx-2.1
- WildFly Application Server 8.1.0.Final jboss-wildfly-8
- Jenkins Server jenkins-1
- Node.js 0.10 nodejs-0.10
- Perl 5.10 perl-5.10
- PHP 5.3 php-5.3

Details

JBoss Enterprise Application Platform 6
Market-leading open source enterprise platform for next-generation, highly transactional enterprise Java applications. Build and deploy enterprise Java in the cloud.

Figure 4. Completed Fields in the New OpenShift Application Wizard

7. Click Next.
8. Complete the fields about the new OpenShift Online application as follows:
   - In the Domain name field, select an existing OpenShift Online domain.
   - In the Name field, type ticketmonster.
   - From the Domain list, ensure the domain you have previously created is selected.
   - From the Gear profile list, select small.



New or existing OpenShift Application

Create a new OpenShift Application.

Domain: jbossjdf

Name: ticketmonster

Type: JBoss Enterprise Application Platform 6

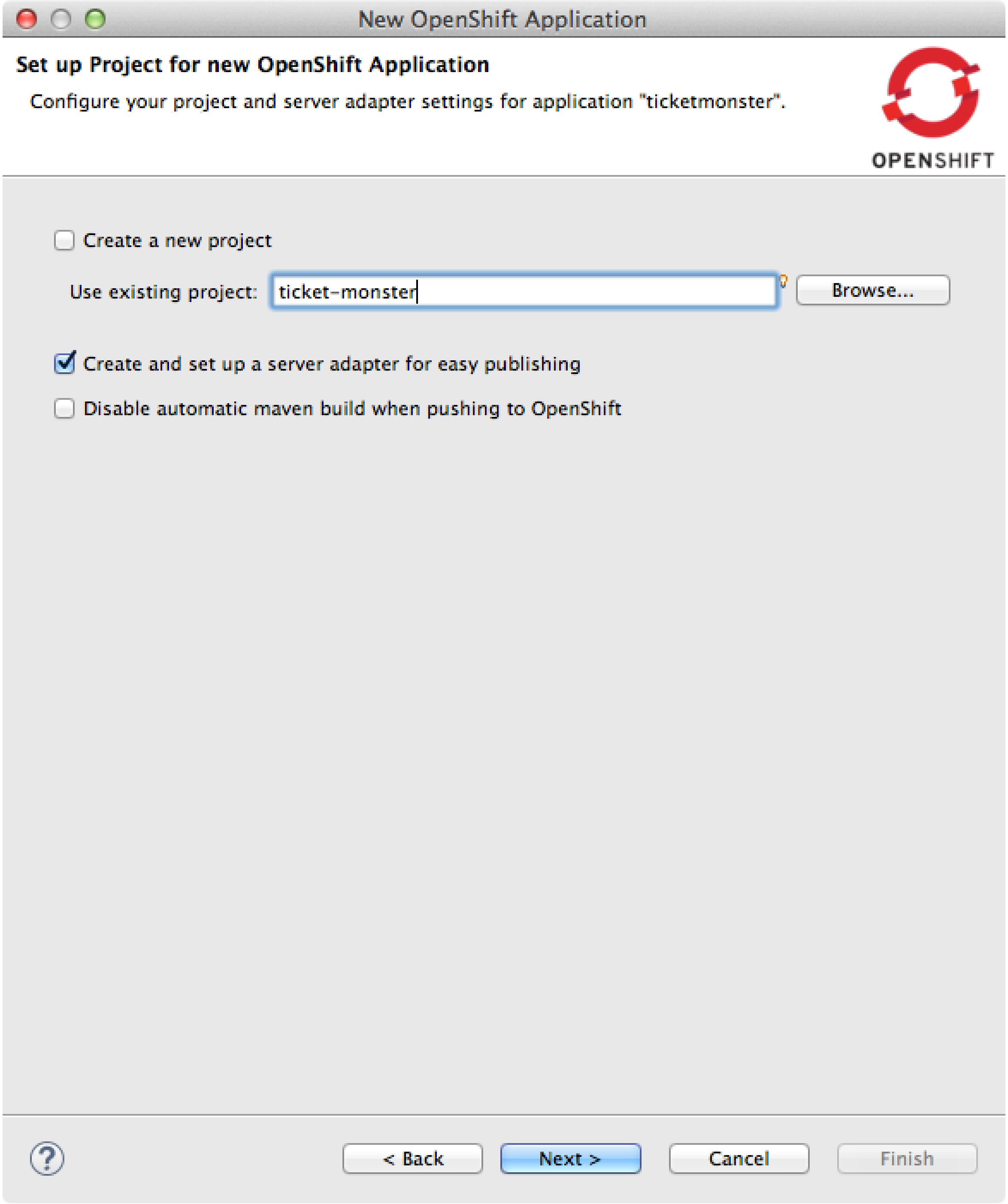
Gear profile: small

☐ Enable scaling

Embedded Cartridges

Figure 5. Completed Fields in the New OpenShift Application Wizard

9. Click Next.
   - Clear the Create a new project check box.
   - In the Use existing project field, type ticket-monster. Alternatively, click Browse to select the ticket-monster project.
   - Ensure the Create and set up a server for easy publishing check box is selected.



New OpenShift Application

Set up Project for new OpenShift Application

Configure your project and server adapter settings for application "ticketmonster".

□ Create a new project

Use existing project: ticket-monster

☑ Create and set up a server adapter for easy publishing

□ Disable automatic maven build when pushing to OpenShift

Figure 6. Completed Fields in the New OpenShift Application Wizard

10. Click Next.

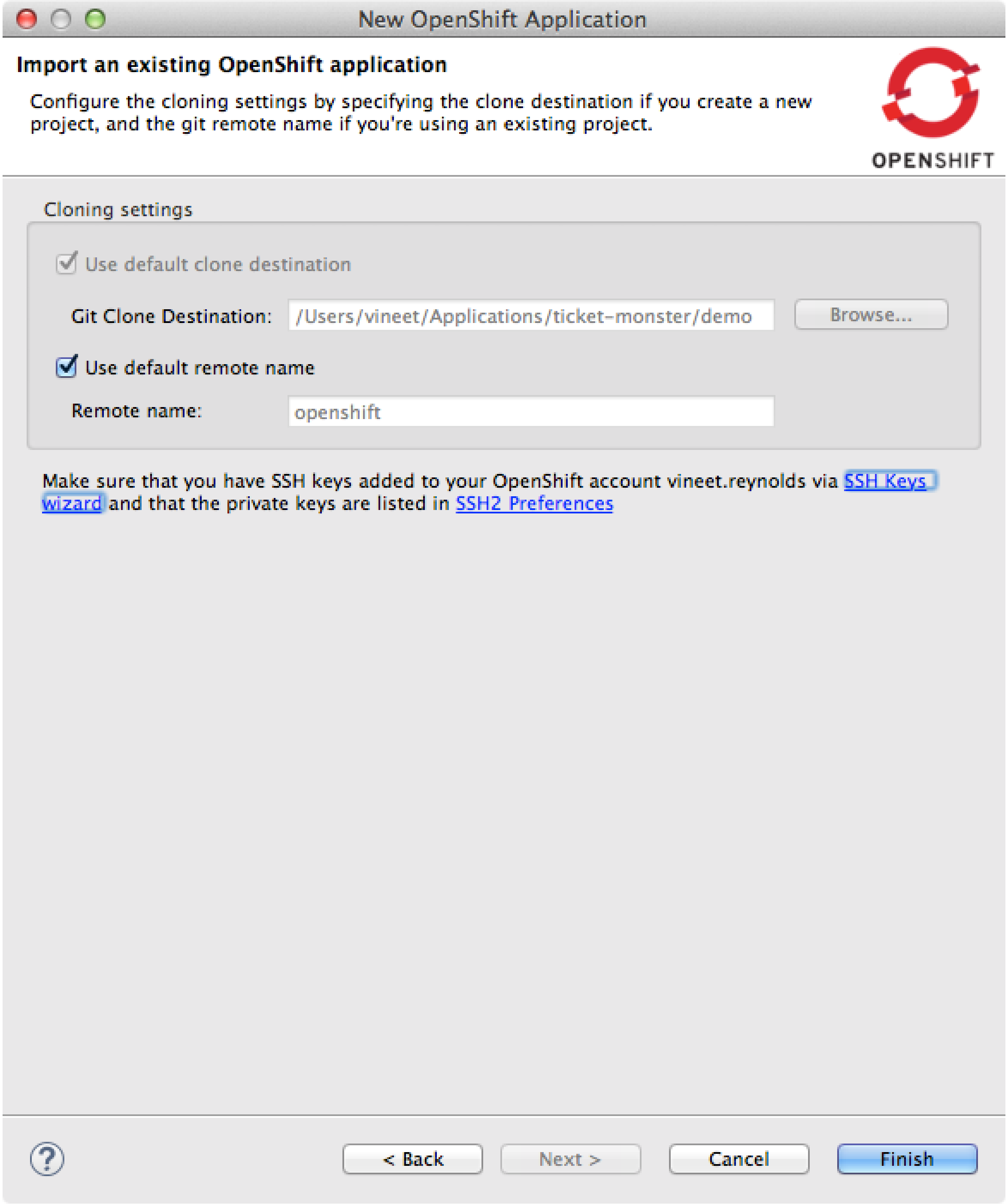
11. Click SSH Keys wizard and click New.

12. Complete the fields about the SSH Keys to be created as follows:

    - In the Name field, type a name for the SSH key.
    - From the Key Type list, ensure SSH_RSA is selected.
    - In the SSH2 Home field, ensure your .ssh directory path is shown.
    - In the Private Key File Name field, type a name for the private key file name. The Public Key File Name field populates automatically with the name of the private key file name with .pub appended.

13. Click Finish.

14. Click OK to close the Manage SSH Keys window.
15. Click Finish to create the new OpenShift application based on the existing workspace ticket-monster project. This process may take some time to complete.



New OpenShift Application

Import an existing OpenShift application

Configure the cloning settings by specifying the clone destination if you create a new project, and the git remote name if you're using an existing project.

Cloning settings

☑ Use default clone destination

Git Clone Destination: /Users/vineet/Applications/ticket-monster/demo

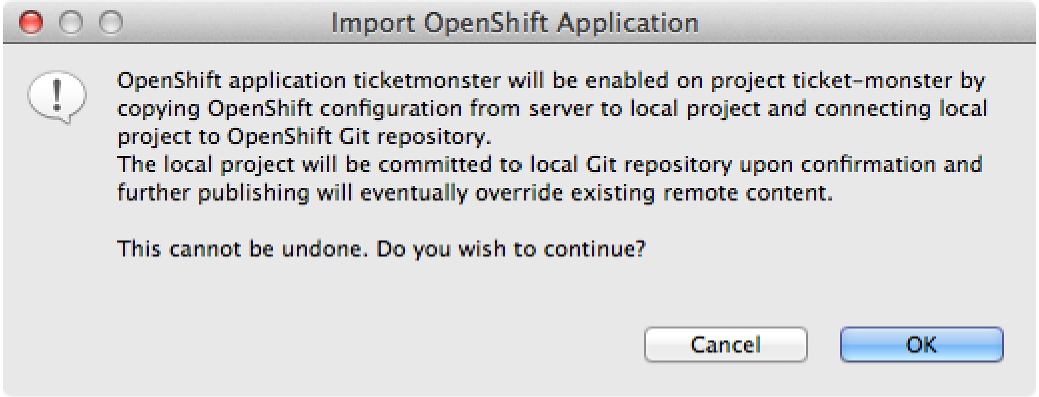
☑ Use default remote name

Remote name: openshift

Make sure that you have SSH keys added to your OpenShift account vineet.reynolds via SSH Keys wizard and that the private keys are listed in SSH2 Preferences

Figure 7. New OpenShift Application Wizard

16. At the prompt stating OpenShift application ticketmonster will be enabled on project ticket-monster ..., click OK. This configures the workspace ticket-monster project for OpenShift and connects it to the OpenShift Online Git repository system used for version control.



OpenShift application ticketmonster will be enabled on project ticket-monster by copying OpenShift configuration from server to local project and connecting local project to OpenShift Git repository.
The local project will be committed to local Git repository upon confirmation and further publishing will eventually override existing remote content.

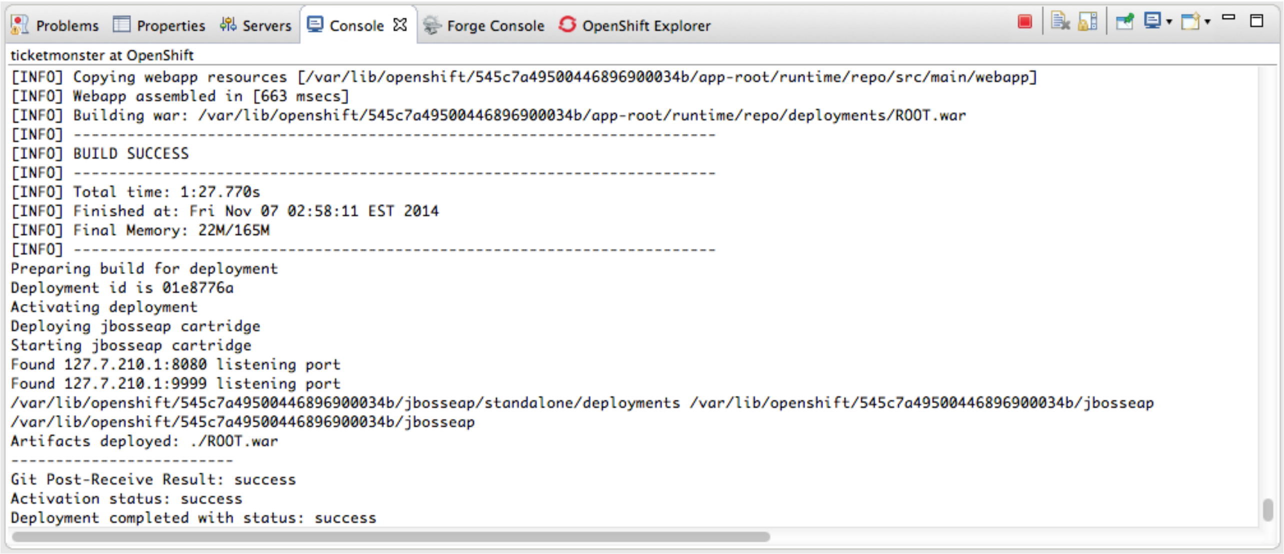
This cannot be undone. Do you wish to continue?

Figure 8. Import OpenShift Application Prompt

17. At the prompt stating the authenticity of the host cannot be established and asking if you are sure you want to continue connecting, verify the host information is correct and click Yes.

18. At the prompt asking if you want to publish committed changes to OpenShift, click Yes. The Console view automatically becomes the view in focus and displays the output from the OpenShift Online server. Once the OpenShift Online ticketmonster application is created and deployed, the Console view displays the following message:

    ```
    Deployment completed with status: success
    ```



```
ticketmonster at OpenShift
[INFO] Copying webapp resources [/var/lib/openshift/545c7a49500446896900034b/app-root/runtime/repo/src/main/webapp]
[INFO] Webapp assembled in [663 msecs]
[INFO] Building war: /var/lib/openshift/545c7a49500446896900034b/app-root/runtime/repo/deployments/ROOT.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1:27.770s
[INFO] Finished at: Fri Nov 07 02:58:11 EST 2014
[INFO] Final Memory: 22M/165M
[INFO] ------------------------------------------------------------------------
Preparing build for deployment
Deployment id is 01e8776a
Activating deployment
Deploying jbosseap cartridge
Starting jbosseap cartridge
Found 127.7.210.1:8080 listening port
Found 127.7.210.1:9999 listening port
/var/lib/openshift/545c7a49500446896900034b/jbosseap/standalone/deployments /var/lib/openshift/545c7a49500446896900034b/jbosseap
/var/lib/openshift/545c7a49500446896900034b/jbosseap
Artifacts deployed: ./ROOT.war
-------------------------
Git Post-Receive Result: success
Activation status: success
Deployment completed with status: success
```

Figure 9. New OpenShift Application Wizard

Deploying to OpenShift using the command-line

To deploy TicketMonster to OpenShift Online, you must create a new OpenShift Online application based on the existing workspace project using OpenShift Tools, as detailed in the procedure below.

NOTE

This procedure documents the deploying process for first-time OpenShift Online users. This includes one-time steps, such as signing up for an OpenShift Online account, creating an OpenShift Online domain and uploading SSH keys. If you have previously used OpenShift Online and OpenShift Tools, omit the one-time steps as appropriate.

Create an OpenShift Account and Domain

If you do not yet have an OpenShift account and domain, browse to OpenShift to create the account and domain.

Get Started with OpenShift details how to install the OpenShift Client tools.

Create the OpenShift Application

NOTE

The following variables are used in these instructions. Be sure to replace them as follows:

- YOUR_DOMAIN_NAME should be replaced with the OpenShift domain name.
- APPLICATION_UUID should be replaced with the UUID generated by OpenShift for your application, for example: 52864af85973ca430200006f
- TICKETMONSTER_MAVEN_PROJECT_ROOT is the location of the Maven project sources for the TicketMonster application.

Open a shell command prompt and change to a directory of your choice. Enter the following command to create a JBoss EAP 6 application:

```
rhc app create -a ticketmonster -t jbosseap-6
```

NOTE

The domain name for this application will be `ticketmonster-YOUR_DOMAIN_NAME.rhcloud.com`

This command creates an OpenShift application named ticketmonster and will run the application inside the jbosseap-6 container. You should see some output similar to the following:

```
Application Options
\-------------------
Domain:     YOUR_DOMAIN
Cartridges: jbosseap-6 (addtl. costs may apply)
Gear Size:  default
Scaling:    no

Creating application 'ticketmonster' ... done


Waiting for your DNS name to be available ... done

Cloning into 'ticketmonster'...
Warning: Permanently added the RSA host key for IP address '54.90.10.115' to the list
of known hosts.
```

```
Your application 'ticketmonster' is now available.

  URL:        http://ticketmonster-YOUR_DOMAIN.rhcloud.com/
  SSH to:     APPLICATION_UUID@ticketmonster-YOURDOMAIN.rhcloud.com
  Git remote: ssh://APPLICATION_UUID@ticketmonster-
YOUR_DOMAIN.rhcloud.com/~/git/ticketmonster.git/
  Cloned to:  /Users/vineet/openshiftapps/ticketmonster

Run 'rhc show-app ticketmonster' for more details about your app.
```

The create command creates a git repository in the current directory with the same name as the application.

You do not need the generated default application, so navigate to the new git repository directory created by the OpenShift command and tell git to remove the source and pom files:

```
cd ticketmonster
git rm -r src pom.xml
```

Copy the TicketMonster application sources into this new git repository:

```
cp -r TICKETMONSTER_MAVEN_PROJECT_ROOT/src .
cp -r TICKETMONSTER_MAVEN_PROJECT_ROOT/pom.xml .
```

You can now deploy the changes to your OpenShift application using git as follows:

```
git add src pom.xml
git commit -m "TicketMonster on OpenShift"
git push
```

The final push command triggers the OpenShift infrastructure to build and deploy the changes.

Note that the `openshift` profile in pom.xml is activated by OpenShift, and causes the WAR build by OpenShift to be copied to the deployments/ directory, and deployed without a context path.

Now you can see the application running at http://ticketmonster-YOUR_DOMAIN.rhcloud.com/.

Using MySQL as the database

You can deploy TicketMonster to OpenShift, making use of a 'real' database like MySQL, instead of the default in-memory H2 database within the JBoss EAP cartridge. You can follow the procedure outlined as follows, to first deploy the TicketMonster application to a JBoss EAP cartridge, and to then add a :

1. Create the OpenShift application from the TicketMonster project sources, as described in the

previous sections.

2. Add the MySQL cartridge to the application.

   a. If you are using JBoss Developer Studio, select the `ticketmonster` application in the **OpenShift Explorer** view. Open the context-menu by right-clicking on it, and navigate to the **Edit Embedded Cartridges…** menu item.

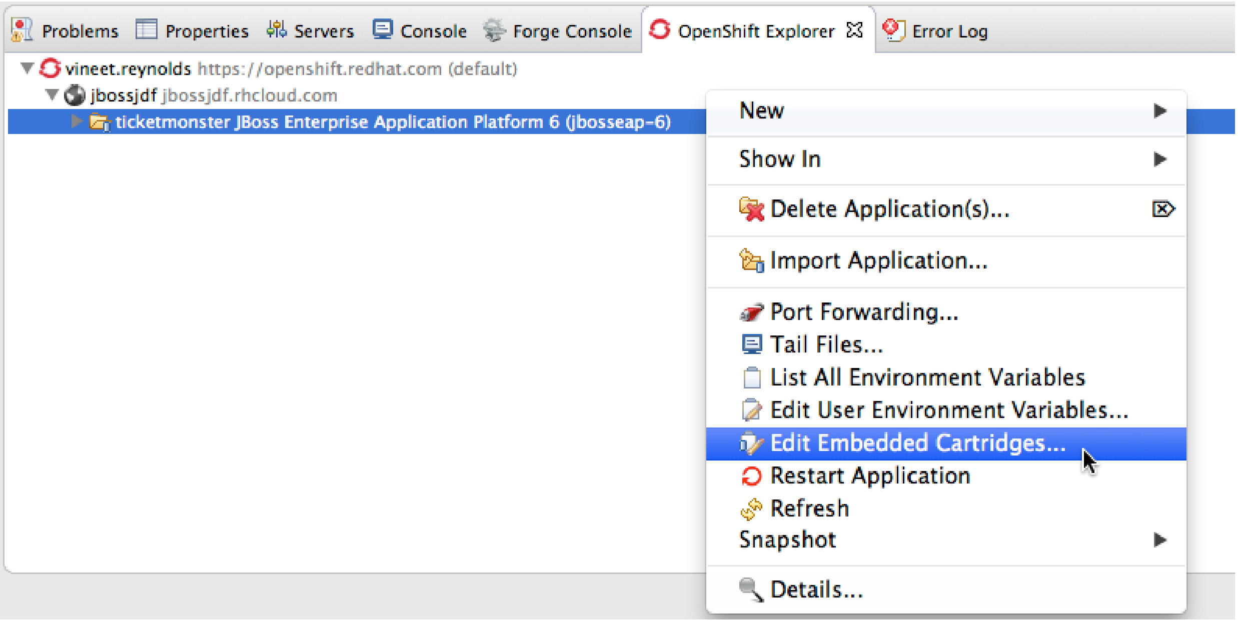
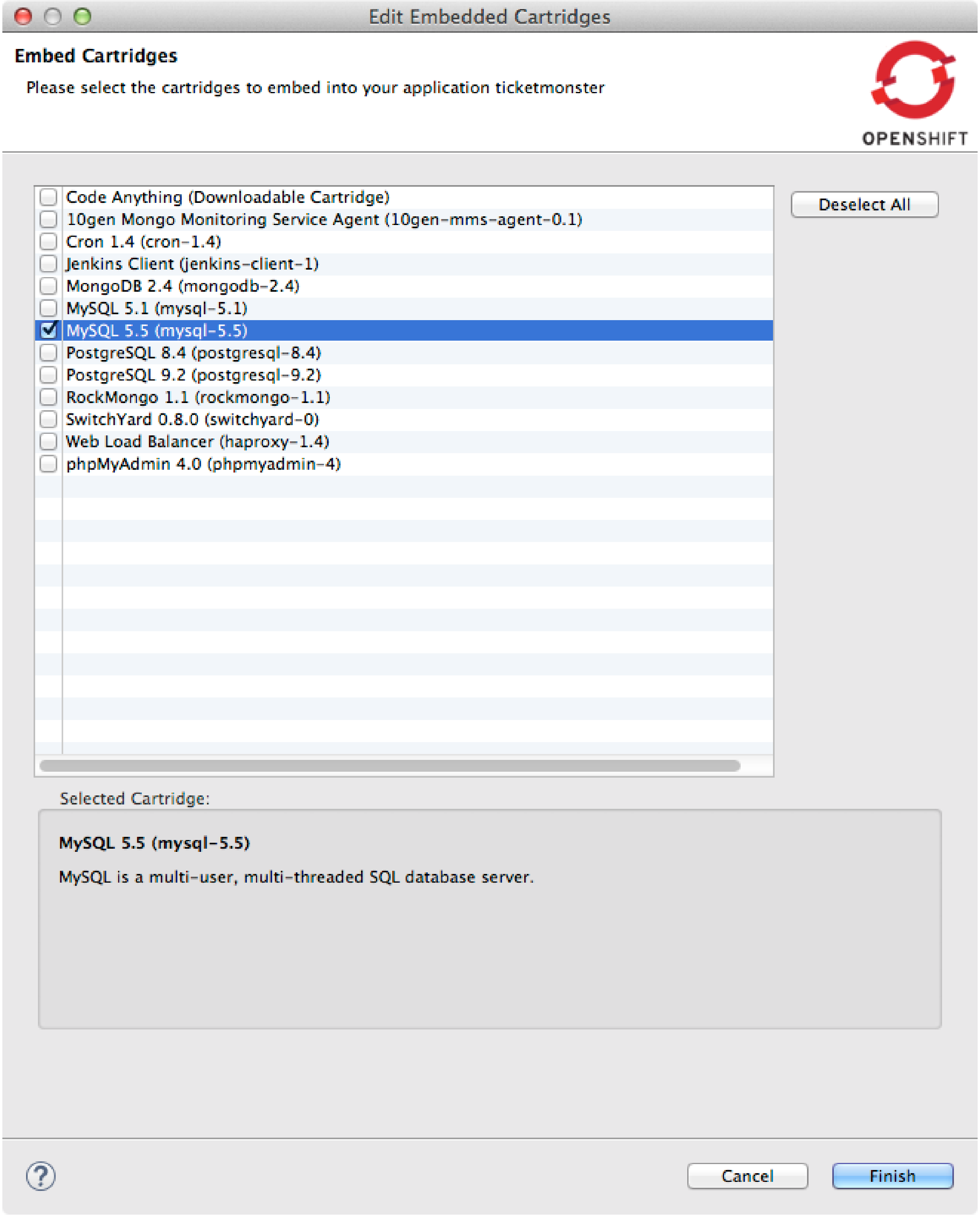

Problems
Properties
Servers
Console
Forge Console
OpenShift Explorer
Error Log
vineet.reynolds https://openshift.redhat.com (default)
jbossjdf jbossjdf.rhcloud.com
ticketmonster JBoss Enterprise Application Platform 6 (jbosseap-6)
New
Show In
Delete Application(s)...
Import Application...
Port Forwarding...
Tail Files...
List All Environment Variables
Edit User Environment Variables...
Edit Embedded Cartridges...
Restart Application
Refresh
Snapshot
Details...

Figure 10. Edit Embedded Cartidges for an OpenShift application

Select the MySQL 5.5 cartidge, and click **Finish**.



Edit Embedded Cartridges

Embed Cartridges

Please select the cartridges to embed into your application ticketmonster

- ☐ Code Anything (Downloadable Cartridge)
- ☐ 10gen Mongo Monitoring Service Agent (10gen-mms-agent-0.1)
- ☐ Cron 1.4 (cron-1.4)
- ☐ Jenkins Client (jenkins-client-1)
- ☐ MongoDB 2.4 (mongodb-2.4)
- ☐ MySQL 5.1 (mysql-5.1)
- ☑ MySQL 5.5 (mysql-5.5)
- ☐ PostgreSQL 8.4 (postgresql-8.4)
- ☐ PostgreSQL 9.2 (postgresql-9.2)
- ☐ RockMongo 1.1 (rockmongo-1.1)
- ☐ SwitchYard 0.8.0 (switchyard-0)
- ☐ Web Load Balancer (haproxy-1.4)
- ☐ phpMyAdmin 4.0 (phpmyadmin-4)

Selected Cartridge:

MySQL 5.5 (mysql-5.5)

MySQL is a multi-user, multi-threaded SQL database server.

Figure 11. Add MySQL cartridge

b. If you are using the command-line, execute the following command, to add the MySQL 5.5 cartridge to the `ticketmonster` application:

```
rhc cartridge add mysql-5.5 -a ticketmonster
```

3. Configure the OpenShift build process, to use the `mysql-openshift` profile within the project POM. As you would know, the Maven build on OpenShift uses the `openshift` profile by default - this profile does not contain any instructions or configuration to create a WAR file with the JPA

deployment descriptor for MySQL on OpenShift. The `mysql-openshift` profile contains this configuration. Since it is not activated during the build on OpenShift, we need to instruct OpenShift to use it as well.

To do so, create a file named `pre_build_jbosseap` under the `.openshift/action_hooks` directory located in the git repository of the OpenShift application, with the following contents:

TICKET_MONSTER_OPENSHIFT_GIT_REPO/.openshift/build_hooks/pre_build_jbosseap

```
export MAVEN_ARGS="clean package -Popenshift,mysql-openshift -DskipTests"
```

This OpenShift action hook sets up the `MAVEN_ARGS` environment variable used by OpenShift to configure the Maven build process. The exported variable now activates the `mysql-openshift` profile, in addition to the default values originally present in the variable.

4. Publish the changes to OpenShift:

   a. If you are using JBoss Developer Studio, right-click the project, go to **Team** → **Commit…** to commit the changes. Select the `pre_build_jbosseap` file to add to the commit. Choose the **Commit and Push** button during committing, to push the changes to the OpenShift repository.

   b. If you are using the command line, add the `pre_build_jbosseap` file to the git index, and commit it, and push to the OpenShift repository, as follows:

   ```
   cd <TICKET_MONSTER_OPENSHIFT_GIT_REPO>
   git add .openshift/build_hooks/pre_build_jbosseap
   git commit -m "Added pre-build action hook for MySQL"
   git push
   ```

NOTE

On Windows, you will need to run the following command to set the executable bit to the `pre_build_jbosseap` file:

```
git update-index --chmod=+x .openshift/build_hooks/pre_build_jbosseap
```

This ensures the executable bit is recognized on OpenShift even though the file was committed in Windows.

Since JBoss Developer Studio does not have a git console, you will need to run this from the command line.

Using PostgreSQL as the database

You can deploy TicketMonster to OpenShift, making use of a 'real' database like PostgreSQL, instead of the default in-memory H2 database within the JBoss EAP cartridge. You can follow the procedure outlined as follows:

1. Create the OpenShift application from the TicketMonster project sources, as described in the previous sections.
2. Add the PostgreSQL cartridge to the application.
   a. If you are using JBoss Developer Studio, select the `ticketmonster` application in the **OpenShift Explorer** view. Open the context-menu by right-clicking on it, and navigate to the **Edit Embedded Cartridges…** menu item.

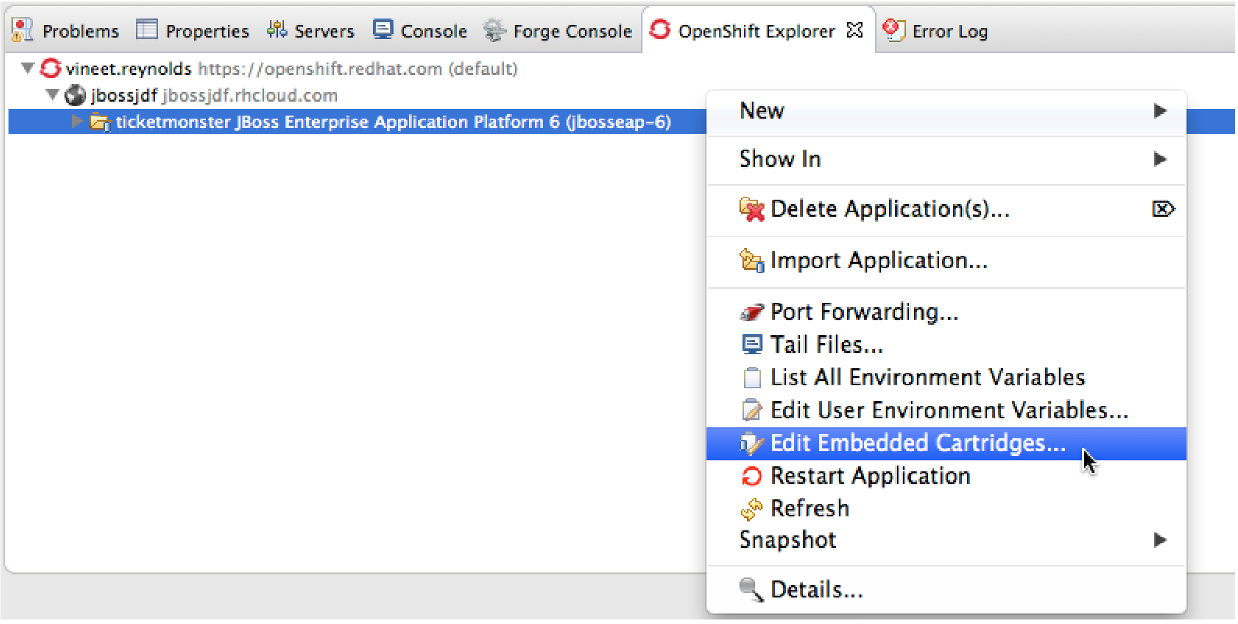
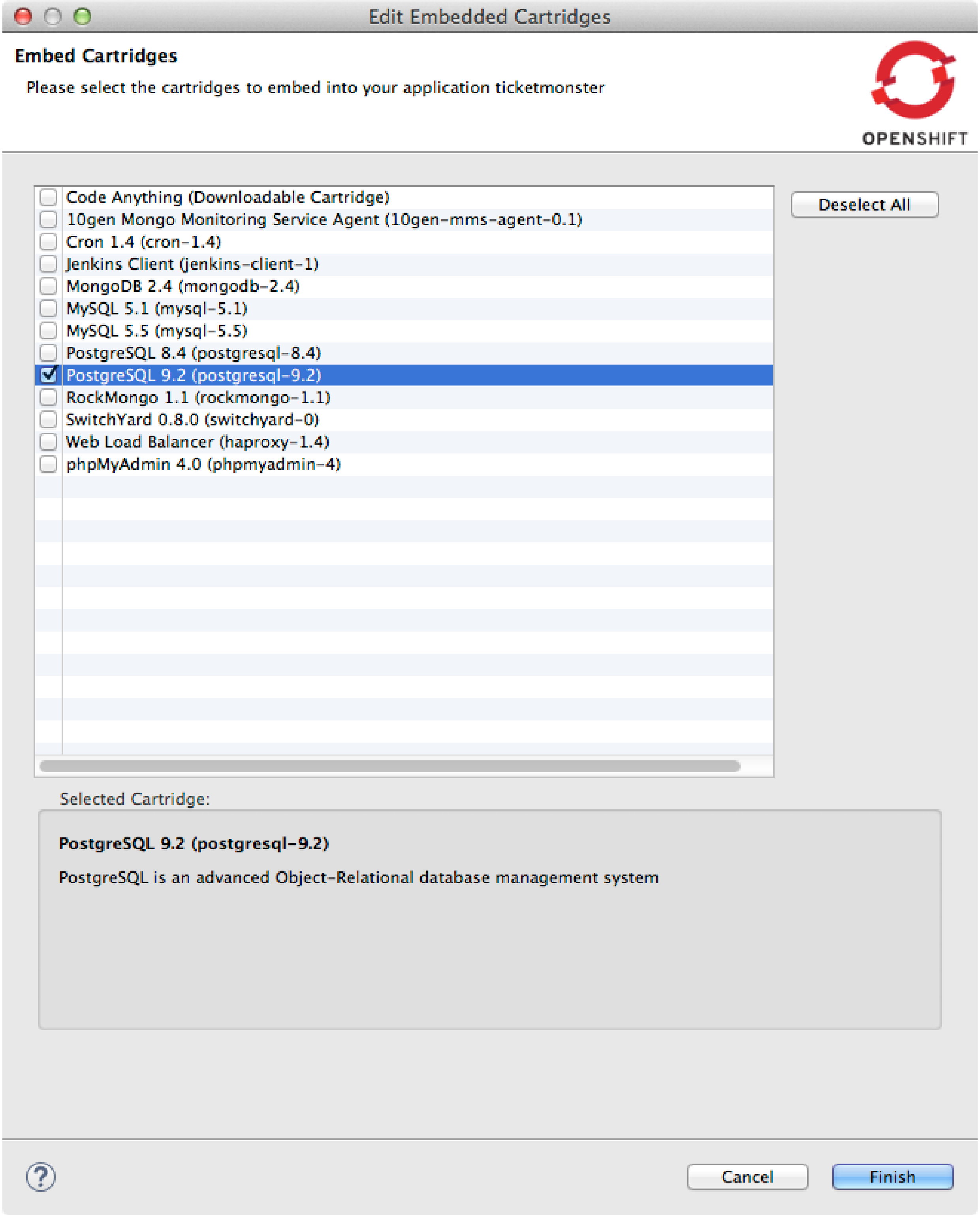

Problems
Properties
Servers
Console
Forge Console
OpenShift Explorer
Error Log
vineet.reynolds https://openshift.redhat.com (default)
jbossjdf jbossjdf.rhcloud.com
ticketmonster JBoss Enterprise Application Platform 6 (jbosseap-6)
New
Show In
Delete Application(s)...
Import Application...
Port Forwarding...
Tail Files...
List All Environment Variables
Edit User Environment Variables...
Edit Embedded Cartridges...
Restart Application
Refresh
Snapshot
Details...

Figure 12. Edit Embedded Cartidges for an OpenShift application

Select the PostgreSQL 9.2 cartidge, and click **Finish**.



Edit Embedded Cartridges

Embed Cartridges

Please select the cartridges to embed into your application ticketmonster

□ Code Anything (Downloadable Cartridge)
□ 10gen Mongo Monitoring Service Agent (10gen-mms-agent-0.1)
□ Cron 1.4 (cron-1.4)
□ Jenkins Client (jenkins-client-1)
□ MongoDB 2.4 (mongodb-2.4)
□ MySQL 5.1 (mysql-5.1)
□ MySQL 5.5 (mysql-5.5)
□ PostgreSQL 8.4 (postgresql-8.4)
☑ PostgreSQL 9.2 (postgresql-9.2)
□ RockMongo 1.1 (rockmongo-1.1)
□ SwitchYard 0.8.0 (switchyard-0)
□ Web Load Balancer (haproxy-1.4)
□ phpMyAdmin 4.0 (phpmyadmin-4)

Selected Cartridge:

PostgreSQL 9.2 (postgresql-9.2)

PostgreSQL is an advanced Object-Relational database management system

Figure 13. Add PostgreSQL cartridge

b. If you are using the command-line, execute the following command, to add the PostgreSQL 9.2 cartridge to the `ticketmonster` application:

```
rhc cartridge add postgresql-9.2 -a ticketmonster
```

3. Configure the OpenShift build process, to use the `postgresql-openshift` profile within the project POM. As you would know, the Maven build on OpenShift uses the `openshift` profile by default - this profile does not contain any instructions or configuration to create a WAR file with the JPA

deployment descriptor for MySQL on OpenShift. The postgresql-openshift profile contains this configuration. Since it is not activated during the build on OpenShift, we need to instruct OpenShift to use it as well.

To do so, create a file named pre_build_jbosseap under the .openshift/action_hooks directory located in the git repository of the OpenShift application, with the following contents:

TICKET_MONSTER_OPENSHIFT_GIT_REPO/.openshift/build_hooks/pre_build_jbosseap

```
export MAVEN_ARGS="clean package -Popenshift,postgresql-openshift -DskipTests"
```

This OpenShift action hook sets up the MAVEN_ARGS environment variable used by OpenShift to configure the Maven build process. The exported variable now activates the postgresql-openshift profile, in addition to the default values originally present in the variable.

4. Publish the changes to OpenShift:

   a. If you are using JBoss Developer Studio, right-click the project, go to **Team** → **Commit…** to commit the changes. Select the pre_build_jbosseap file to add to the commit. Choose the **Commit and Push** button during committing, to push the changes to the OpenShift repository.

   b. If you are using the command line, add the pre_build_jbosseap file to the git index, and commit it, and push to the OpenShift repository, as follows:

```
cd <TICKET_MONSTER_OPENSHIFT_GIT_REPO>
git add .openshift/build_hooks/pre_build_jbosseap
git commit -m "Added pre-build action hook for PostgreSQL"
git push
```

NOTE

On Windows, you will need to run the following command to set the executable bit to the pre_build_jbosseap file:

```
git update-index --chmod=+x .openshift/build_hooks/pre_build_jbosseap
```

This ensures the executable bit is recognized on OpenShift even though the file was committed in Windows.

Since JBoss Developer Studio does not have a git console, you will need to run this from the command line.