

WebSphere Cartridge User Guide

PREPARED FOR - FRIT

Table of Contents

History and Revisions	1
1. Synopsis	2
2. OpenShift 2 WebSphere Gear Creation	3
2.1. Creating WebSphere Application Via the OpenShift Console	3
2.1.1. Login to the OpenShift Console.....	3
2.1.2. Click on "Create Your First Application Now"	3
2.1.3. Complete the Application Form	4
2.1.4. Create the Application & Access the WebSphere Console	5
2.2. Creating WebSphere Application with OpenShift Red Hat Client Tools	6
2.2.1. Creating WebSphere Application without a Database Add-On.....	6
2.2.2. Creating WebSphere Application with a Database Add-On	7
3. Deploying Applications to WebSphere	8

History and Revisions

Version	Date	Authors	Changes
0.5	10/08/2015	Toufic Arabi tarabi@redhat.com	Initial Draft

1. Synopsis

The purpose of this document is to present how OpenShift 2 users can interact with the WebSphere 8.5.5.1 OpenShift Enterprise V2.2 cartridge. By the end of this document users that have the WebSphere cartridge enabled for them will be able to provision WebSphere servers in a matter of minutes.

The end result of their OpenShift 2 WebSphere gear provisioning would be a WebSphere console that gives them full administrative rights over their WebSphere servers.

Each WebSphere server is contained in an OpenShift gear which means that collocated WebSphere users are segregated from one another. The figure below shows the WebSphere administrative console that every users will obtain.

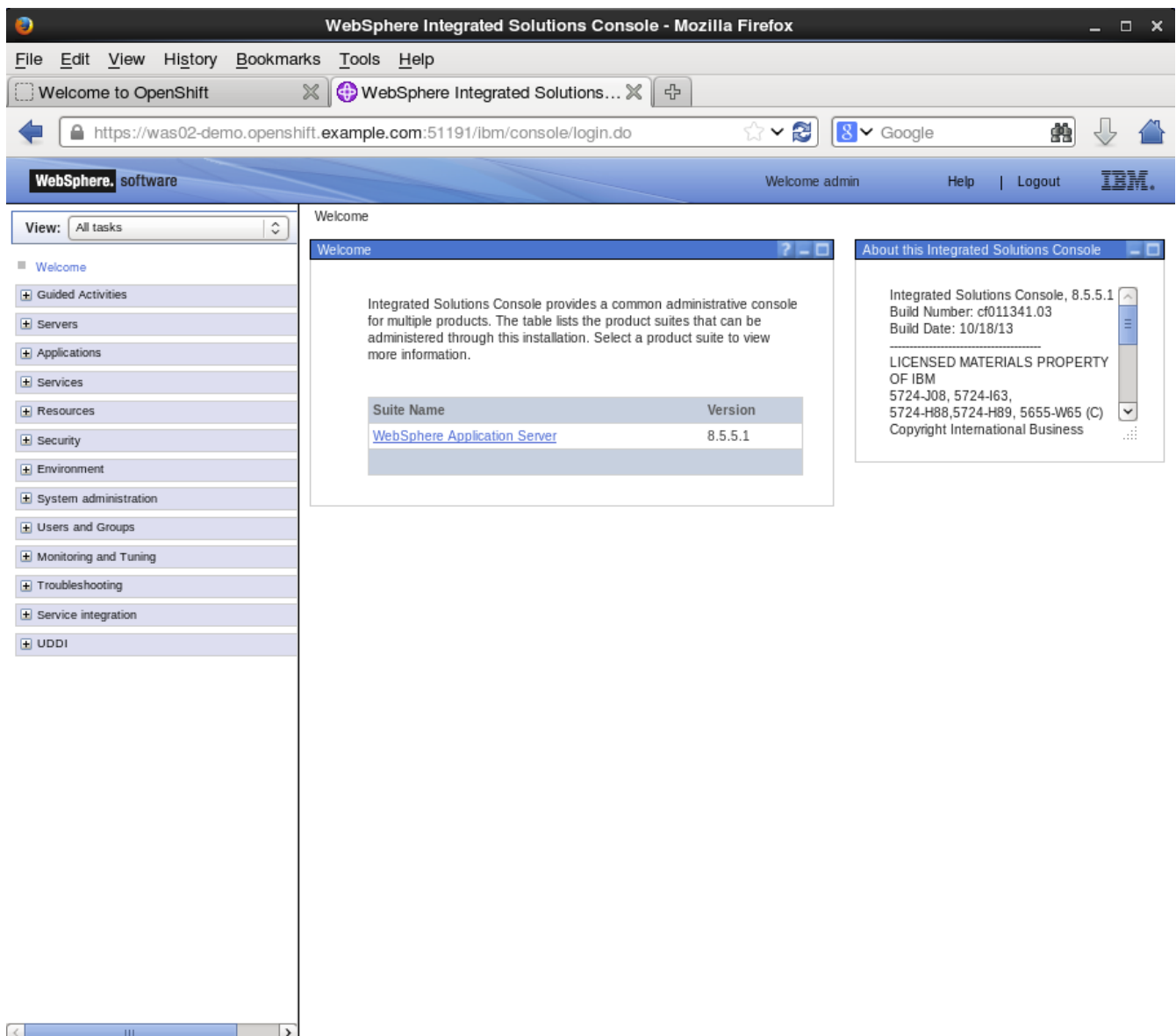


Figure 1: WebSphere Administration Console in OpenShift

2. OpenShift 2 WebSphere Gear Creation

There are two ways to create a WebSphere gear in OpenShift 2:

1. Via the OpenShift Console
2. Via the OpenShift Client Tools

The following sections will present how each of these methods can be used to create a WebSphere gear in OpenShift 2

2.1. Creating WebSphere Application Via the OpenShift Console

The steps to create a WebSphere gear via the OpenShift console are very straightforward:

2.1.1. Login to the OpenShift Console



OpenShift can be configured with various identity stores. Check with your organization's administrator to find out your login credentials.

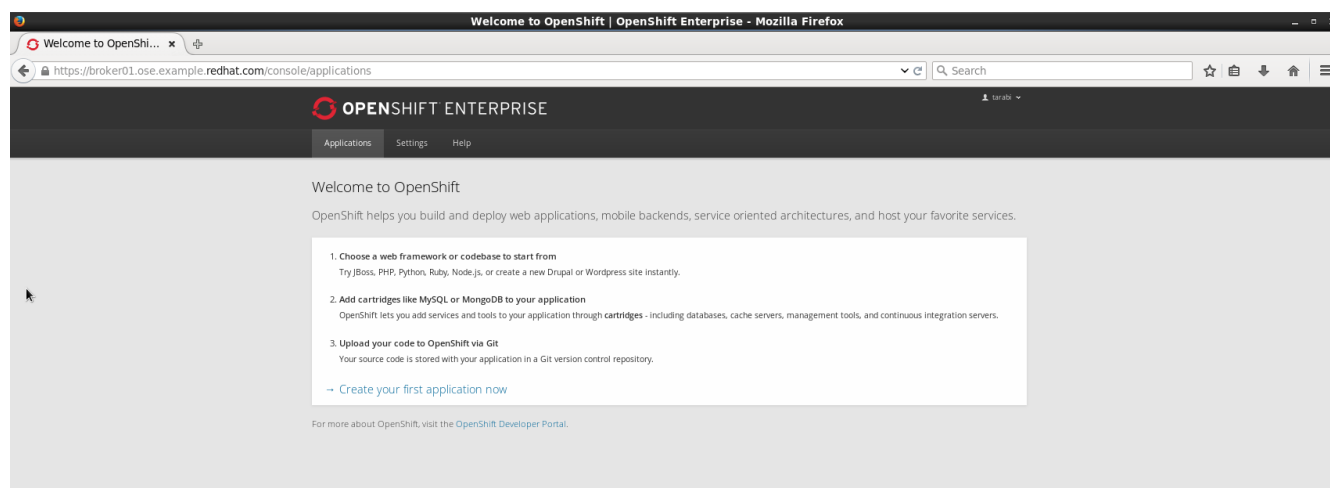


Figure 2: OpenShift Login Page

2.1.2. Click on "Create Your First Application Now"



This step assumes that the user has already created a namespace, and has setup his public key in his OpenShift account.

The next page that is displayed should show the list of all the cartridges that are available. The name of the cartridge might be different between this document and what you are seeing, therefore make sure that you select the one that hints to WebSphere.



OpenShift can be configured with various identity stores. Check with your organization's administrator to find out your login credentials.

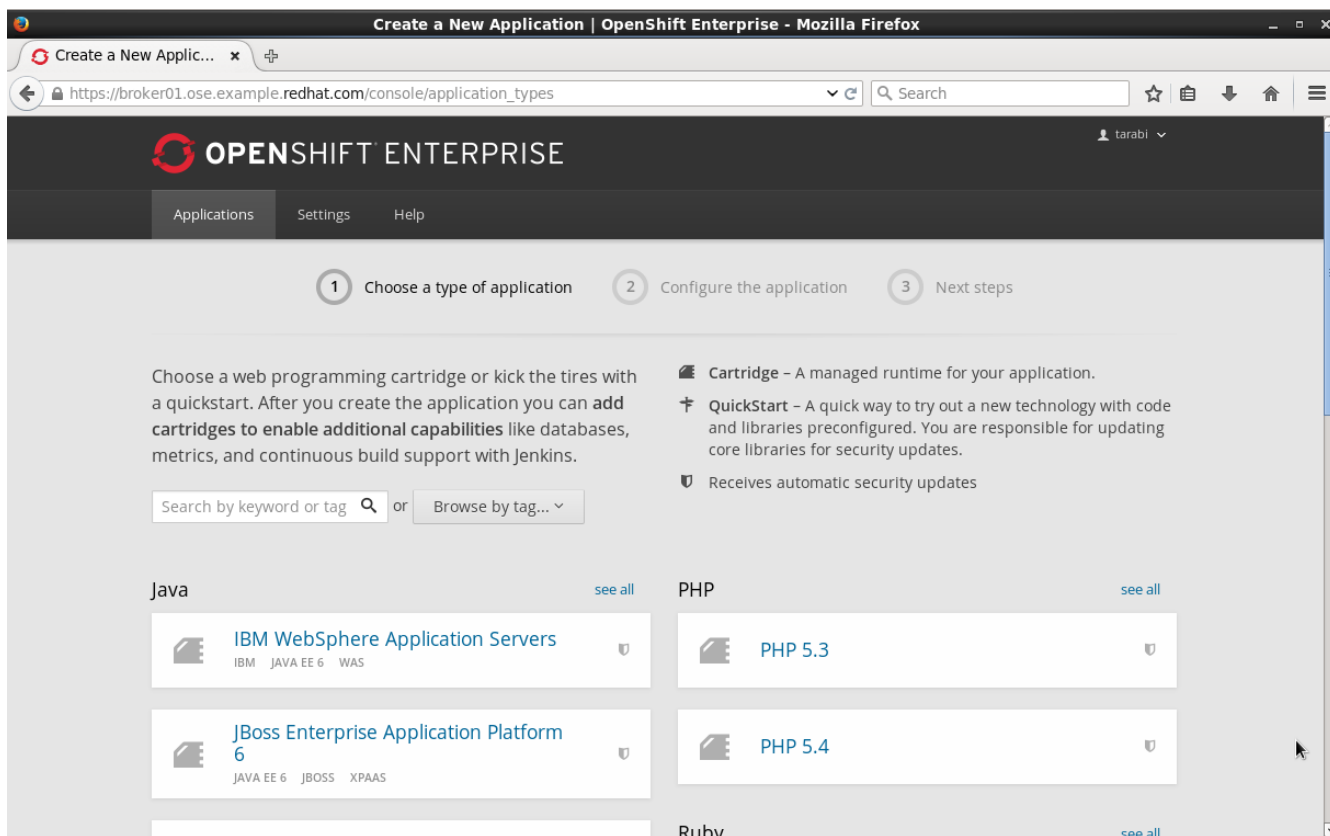


Figure 3: OpenShift Available Cartridges Menu

2.1.3. Complete the Application Form

On the next page you will be presented with a form that has a few fields to be completed:

1. The Public URL: This is the name your application will be registered with in DNS. It will have the formate **application_name-namespace.Enterprise_Domain**
2. Source Code: If your application code is stored in a Git repository, you may provide the Git URL and OpenShift will clone the code into the Gear so that the application is built and ready on gear creation - Not frequently used.
3. Gears: This is the size in memory, disk and CPU that will be given to the application. Each cartridge may be associated wiht one or more gear types. Check your OpenShift administrator to find out what gear size to choose for the WebSphere cartridge
4. Scaling:

There are two options that you may choose from:

- **No Scaling:** This means that OpenShift will not attempt to spin up identical gears of your applicaiton when its being "hit" with a lot of web traffic
- **Scaling:** Allows OpenShift to create more gears as web traffic increases to your application. This is horizontal scaling. Once the web traffic is reduced, the additional gears are destroyed and resources are released back to the system.

Create a New Application | OpenShift Enterprise - Mozilla Firefox

https://broker01.ose.example.redhat.com/console/application_type/cart/frb-websphere-8.5.5.1

OPENSIFT ENTERPRISE

Applications Settings Help

1 Choose a type of application 2 **Configure the application** 3 Next steps

Based On **IBM WebSphere Application Servers Cartridge**

IBM WebSphere Application Server for Developers provides access to the development runtime when production runtime capabilities matter. It's a no-charge development runtime. This software helps to enable faster, more efficient development of applications and services, and is available without the expense of a priced and supported runtime on the developer's desktop.

<http://www-03.ibm.com/software/products/en/appserv-wasdev/>

Receives automatic security updates

Community created

Public URL

You can also [create a new domain](#). OpenShift will automatically register this domain name for your application. You can add your own domain name later.

Source Code

We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults. If you provide a Git URL, your application will start with an exact copy of the code and configuration provided in this Git repository.

Gears **medium**

Gears are the application containers running your code.

Cartridges **IBM WebSphere Application Servers**

Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.

Scaling **No scaling**

OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.

[Back](#) [Create Application](#) +1 @

Figure 4: OpenShift Application Form

2.1.4. Create the Application & Access the WebSphere Console

Once you are done completing the form above, click on **"Create Application"** button. The page will show that it is "working". It will take about a few minutes before the WebSphere gear is created.

The OpenShift console will then echo out the details of the WebSphere Administration Console as per the image below.

Get Started | OpenShift Enterprise - Mozilla Firefox

https://broker01.ose.example.redhat.com/console/application/566f00c10c3723a21a000150-frbwebsphere/get_started?wizard=true

OPENSIFT ENTERPRISE

Applications Settings Help

1 Choose a type of application 2 Configure the application 3 **Next steps**

✓ **WebSphere version: 8.5.5.1 created. Please make note of the Administrative Console URL**
Admin Console: <https://frbwebsphere-arabi.ose.example.redhat.com:47519/ibm/console/login.do>

Profile name now exists. More information can be obtained from your gear's app-root/data/profile/logs/AboutThisProfile.txt file

Your application has been created. [Continue to the application overview page.](#)

Making code changes

Install the Git client for your operating system, and from your command line run

```
git clone ssh://566f00c10c3723a21a000150@frbwebsphere-arabi.ose.example.redhat.com:~/.git/frbwebsphere.git/
cd frbwebsphere/
```

This will create a folder with the source code of your application. After making a change, add, commit, and push your changes.

```
git add .
git commit -m 'My changes'
git push
```

When you push changes the OpenShift server will report back its status on deploying your code. The server will run any of your configured [deploy hooks](#) and then restart the application.

Manage your app

The console is convenient, but if you need deeper control try our other client tools

Command-Line

All of the capabilities of OpenShift are exposed through our command line tool, [rhc](#). [Follow these steps to install the client](#) on Linux, Mac OS X, or Windows.

After installing the RHC read more [on how to manage your application from the command line in our User Guide](#).

JBoss Developer Studio

The JBoss Developer Studio is a full featured IDE with OpenShift integration built in. It gives you the ability to create, edit and deploy applications without having to leave the IDE. Links to download, install and use the JBoss Developer Studio for Linux, Mac OS X, or Windows can be found on the [JBoss Developer Studio tools page](#).

Figure 5: WebSphere Gear Creation Confirmation and Console Details

The WebSphere Administration Console can be access in two ways:

- Option 1: Preferred - After you have created your gear, do an `rhc port-forward <GEAR_NAME>` and open a browser with the following URL:

```
https://<YOUR_LOCAL_IP>:9043/ibm/console
```

- Option 2: The Admin Console is also exposed via a separate external port that is echoed upon gear creation in the OSE console.

If the port number has been misplaced it can be re-obtained as follows:

```
rhc ssh <app-name> [-n <namespace>] #only include namespace if mutiple apps with  
same name  
export | grep WC_ADMINHOST_SECURE_PROXY_PORT
```

Now point your browser to the following URL:

https://<GEAR_DNS>:<WC_ADMINHOST_SECURE_PROXY_PORT>/ibm/console/login.do

The Admin Console should then appear.

2.2. Creating WebSphere Application with OpenShift Red Hat Client Tools

The second mechanism by why a WebSphere application may be created is via the OpenShift Red Hat client tools that could be installed on your computer. The commands below assume that a domain space **test** has already been created.

2.2.1. Creating WebSphere Application without a Database Add-On

In a terminal window run the below command to create the application **myapp** with the WebSphere cartridge **frb-websphere-8.5.5.1**

```
rhc app create -a myapp -n mynamespace -t frb-websphere-8.5.5.1
```

To create a scalable app, you have to add the `-s` option. The namespace `mysamespace` needs to be created before running the above command. The output below should be seen:


```
[ose@node01 ~]$ rhc app create myapp frb-websphere-8.5.5.1
Application Options
-----
Domain:      test
Cartridges:  frb-websphere-8.5.5.1
Gear Size:   default
Scaling:     no

Creating application 'myapp' ...
```

2.2.2. Creating WebSphere Application with a Database Add-On

In the previous section we did not include a database for our application. In this section we demonstrate how this can be done from the command line.

In a terminal window run the below command to create the application **myapp** with the WebSphere cartridge **frb-websphere-8.5.5.1** and an Oracle cartridge **frb-oracle-1.0** as a database add on:

```
rhc app create -a myapp -n mynamespace -t frb-websphere-8.5.5.1 -t frb-oracle-1.0
```

To create a scalable app, you have to add the **-s** option. The namespace **myspace** needs to be created before running the above command.

The name of the cartridges might vary between this document's version and the current state of the cartridges with regards to versioning, yet the idea remains the same.

3. Deploying Applications to WebSphere

All applications deployments should be done via the console. We also strongly recommend that the console is accessed via `rhc port-forward` when doing deployments. The following notes only apply to WAS gears that are created in **scaled** mode in OpenShift.

WebSphere comes pre-loaded with a set of default applications. When creating a **scaled** WebSphere gear, OpenShift creates an HA Proxy gear alongside it. When web traffic increases to the main gear another identical gear is created and more are added as necessary to handle the traffic. This is done automatically and once web traffic decreases, gears are destroyed. The goal of the HA proxy gear is to direct traffic to the WebSphere gear that can handle it based on Web traffic load.

On initial creation of the scaled WAS gear, the HA proxy gear needs to be aware of an existing context as it performs a health check on its existence. The WebSphere cartridge keys off the `/hello` context that ships with the `DefaultWebApplication` in WebSphere.

Should developers and administrators decide to remove this default application, they would have to alter the HA Proxy configuration inside their WebSphere gear to reflect a new context that would exist at server startup. (The root context is a valid context). Failure to do so will cause the HA proxy not to start on gear restart, and the deployments to WebSphere not available via their FQDN and context.

After creating the initial WebSphere gear, SSH into the gear via the RHC client tools or native SSH to the gear as the following:

```
#only include namespace if mutiple apps with same name
rhc ssh <app-name> [-n <namespace>]

OR

ssh <gear-uuid>@app_name-namespace.domain
```

In the WebSphere gear edit the `./haproxy/conf/haproxy.cfg`

Location the `GET /hello` directive and change the `/hello` directive to your new context. For example, if deploying you application to the root context, change the directive to: `GET /`

Restart the WebSphere gear.