

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 Applications with OpenShift Red Hat Client Tools . . . . . . . . . . . . . . . . . . 7

2.2.1. Creating WebSphere Application without a Database Add-On. . . . . . . . . . . . . . . . . . . . . . . 7

2.2.2. Creating WebSphere Application with a Database Add-On . . . . . . . . . . . . . . . . . . . . . . . . . . 7

3. Viewing WebSphere Logs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

4. Running WSADMIN Utility in the OpenShift Gear . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

5. Deploying Applications to WebSphere. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**History and Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Changes** |
|  |  |  |  |
| 0.5 | 10/08/2015 | Toufic Arabi | Initial Draft |
|  |  | [tarabi@redhat.co](mailto:tarabi@redhat.com)m |  |
|  |  |  |  |

1

**1. Synopsis**

The purpose of this document is to present how OpenShift 2 users can interact with the WebSphere 8.5 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 user will obtain.



*Figure 1: WebSphere Administration Console in OpenShift*

2

**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:

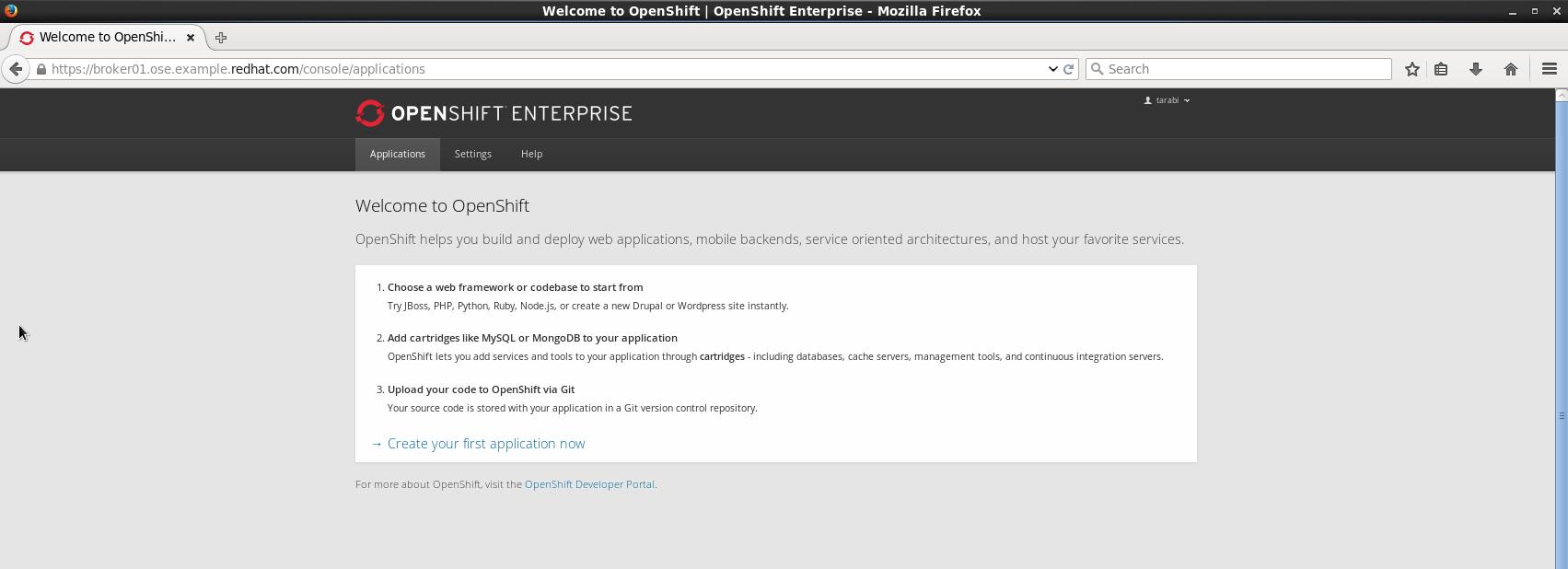
Provisioning a WebSphere server via the OpenShift console is only to be done by NIT MW Administrators and not developers. Developers, while they have access

to the OpenShift console, and while they can create WebSphere gears, will not be able to access the WebSphere console once the gear is created.

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"**

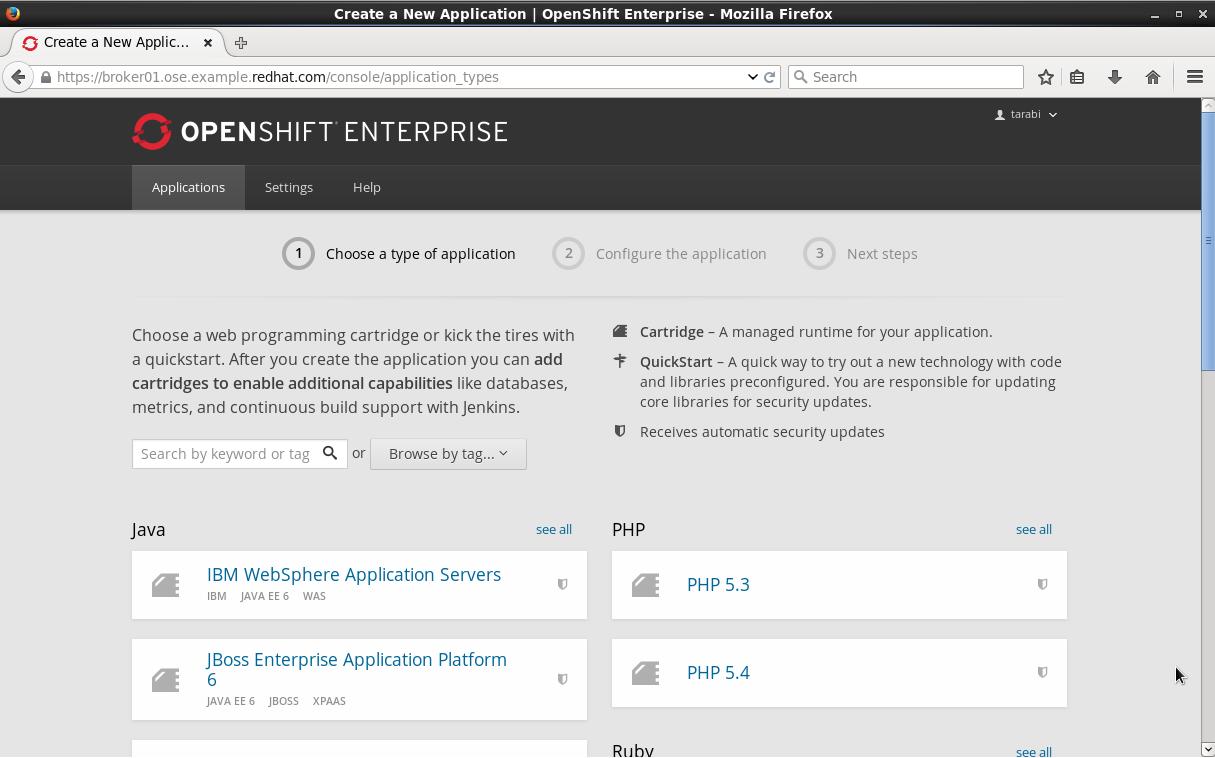
|  |  |  |
| --- | --- | --- |
|  | This step assumes that the user has already created a namespace, and has setup |  |
| his public key in his OpenShift account. |  |

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

3

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:

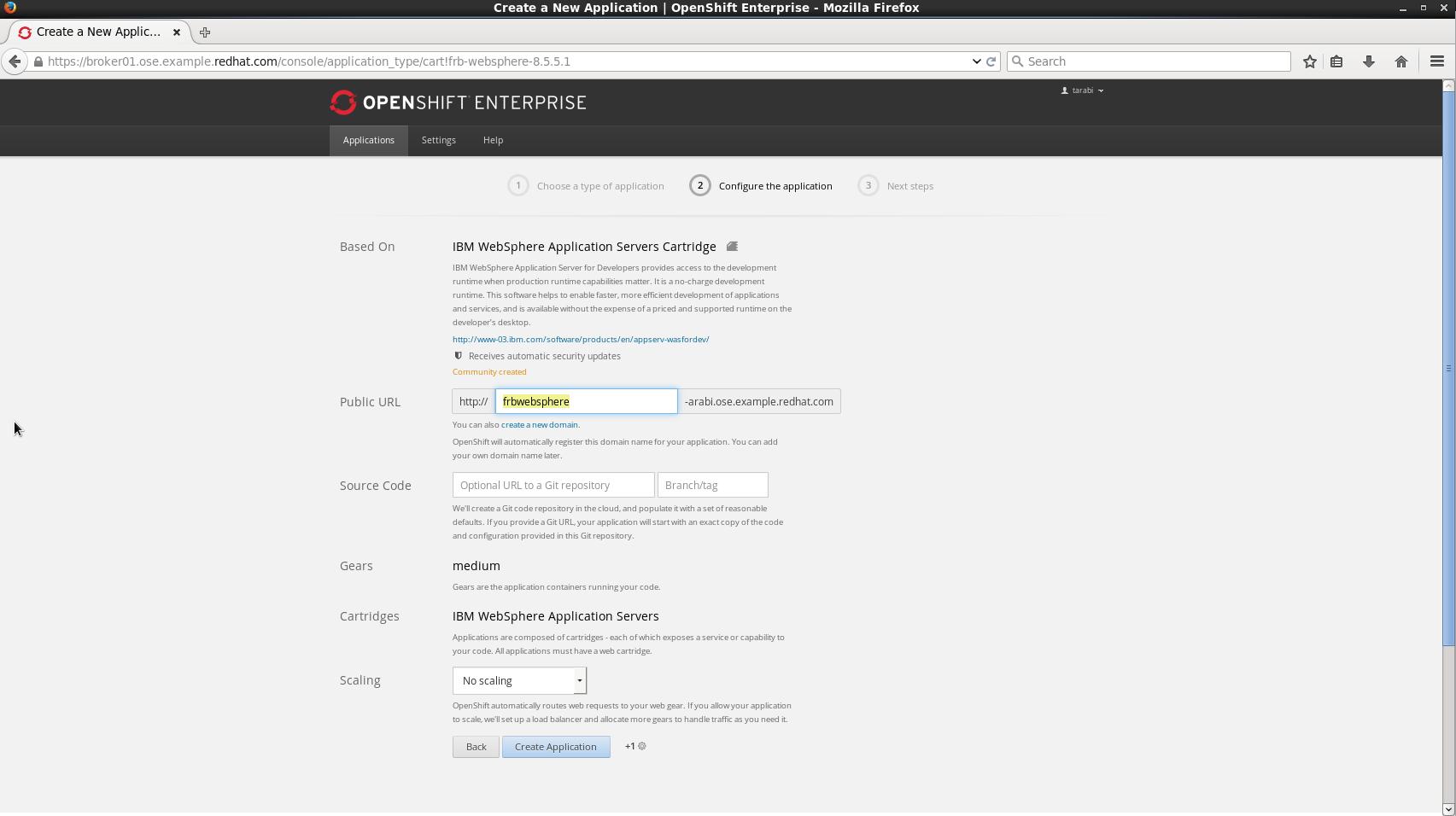
1. The Public URL: This is the name your application will be registered with in DNS. It will have the format **application\_name-namespace.Enterprise\_Domain**
2. Source Code: If your application code is stored in a Git repository, you may provide the Git URL and OpenShift will clone the code into the 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 with 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 yourapplication when its being "hit" with a lot of web traffic
* **Scaling:** Allows OpenShift to create more gears as web traffic increases to your application. Thisis horizontal scaling. Once the web traffic is reduced, the additional gears are destroyed and

4

resources are released back to the system.

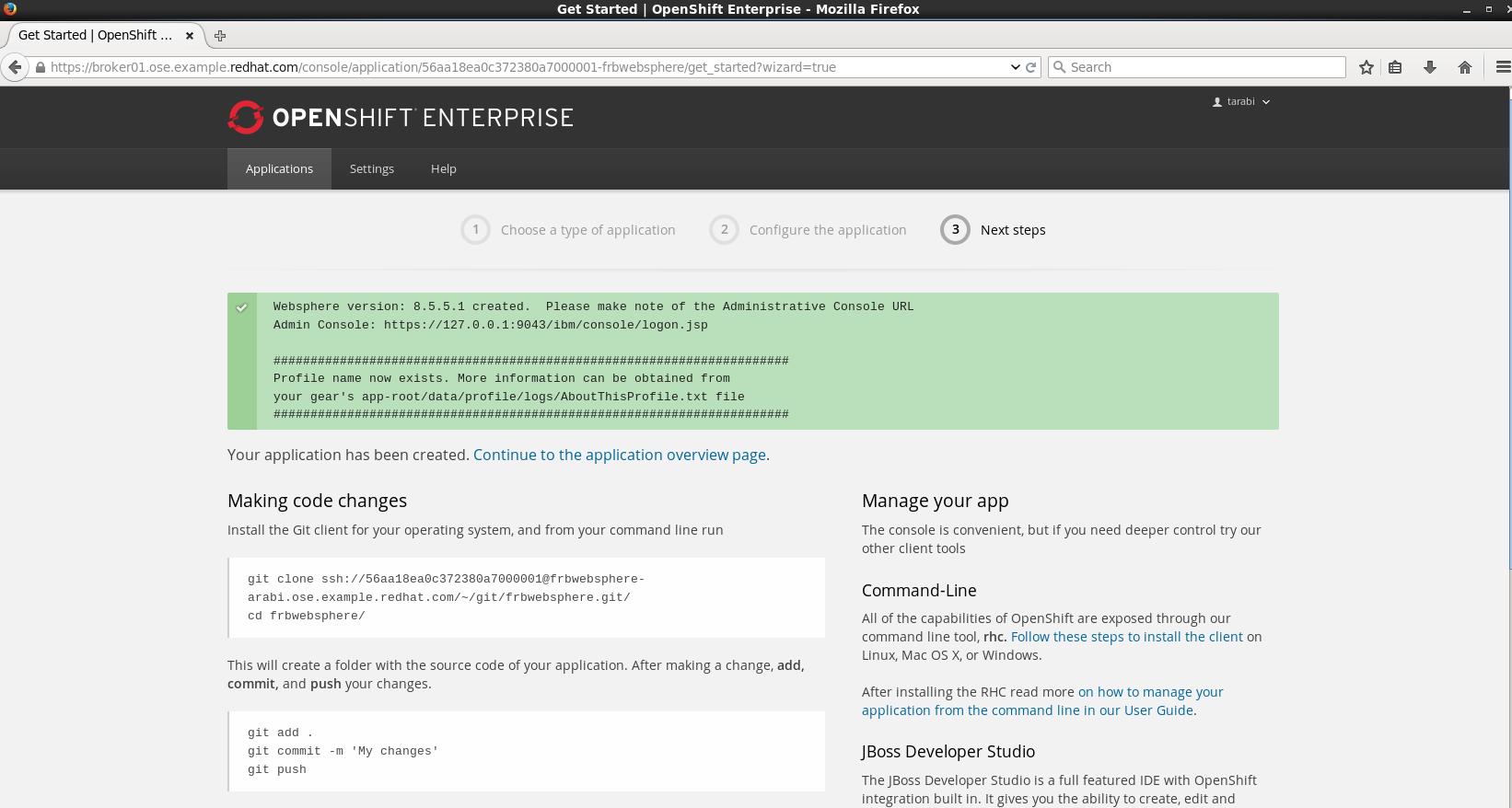


*Figure 4: OpenShift Application Form*

**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.



*Figure 5: WebSphere Gear Creation Confirmation and Console Details*

5

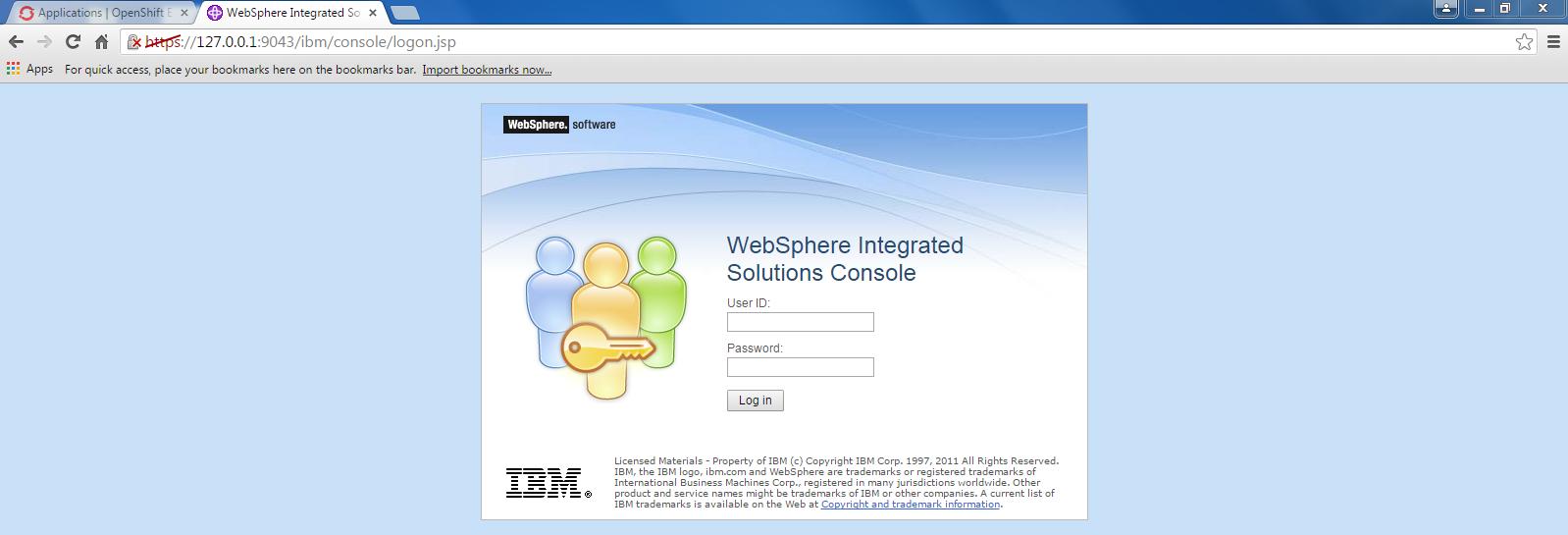
The WebSphere Administration Console can be accessed via the **port-forward** that ships with the Red Hat Client Tools once a WebSphere application has been created:

1. After you have created your gear (application), do an rhc port-forward <APP\_NAME> and open a browser with the following URL:



https://127.0.0.1:9043/ibm/console/logon.jsp

1. The Admin Console should then present a username and password challenge as per the image below. Enter your AD (windows login) credentials to login



*Figure 6: WebSphere Administration Console Username/Password Challenge*

6

**2.2. Creating WebSphere Applications 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 **mynamespace** has already been created.

This is the only way by which non Admin users will be able to access the WebSphere console with their Active Directory (windows) credentials. Admin

users, if using the client tools do not need to pass the OPENSHIFT\_LOGIN\_WAS\* variable.

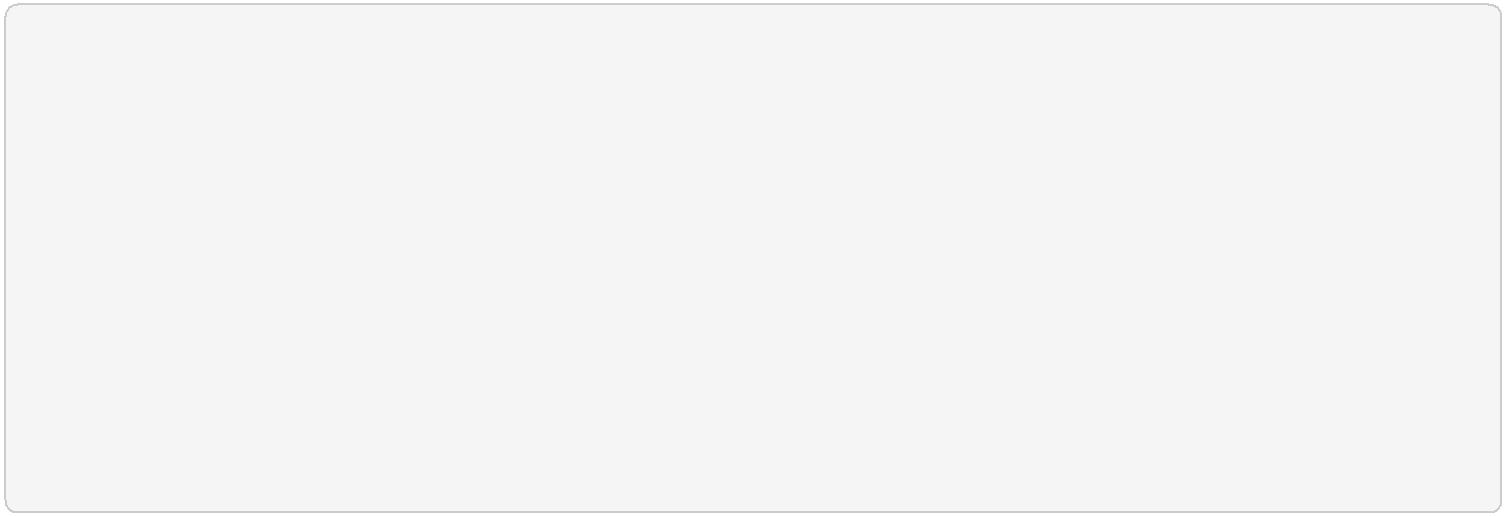
1. **Creating WebSphere Application without a Database Add-On**

In a terminal window run the below command to create the application **myapp** with the WebSphere cartridge **frb-websphere-8.5**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| rhc | app | create | -a | myapp | -n | mynamespace | -t | frb-websphere-8.5 | -e |
| OPENSHIFT\_LOGIN\_WAS=your\_username | | | | |  |  |  |  |  |

The OPENSHIFT\_LOGIN\_WAS variable must be set to the same username that is authorized in the idenity store (AD) that WebSphere authenticates against. This username is most likely the same username that you used to login to the OpenShift Console and your windows computer.

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



[ose@node01 ~]$ rhc app create -a myapp -t frb-websphere-8.5 -n mynamespace -e

OPENSHIFT\_LOGIN\_WAS=your\_username

Application Options

-------------------

Domain: mynamespace

Cartridges: frb-websphere-8.5

Gear Size: default

Scaling: no

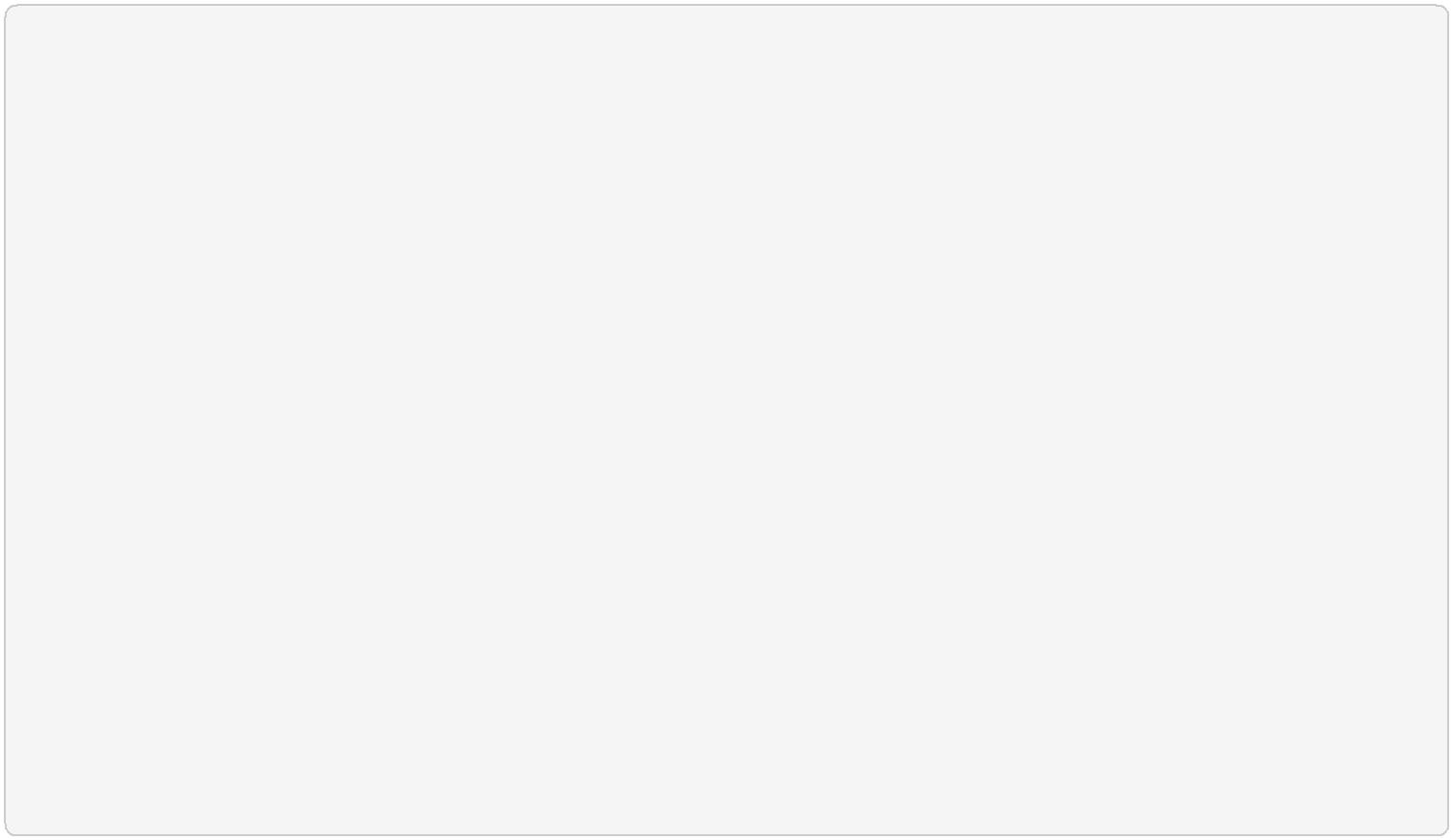
Creating application 'myapp' ...

**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** and an Oracle cartridge **frb-oracle-12.0** as a database add on:

7

rhc app create -a myapp -n mynamespace -t frb-websphere-8.5 -t frb-oracle-12.0 -e OPENSHIFT\_LOGIN\_WAS=your\_username

Application Options

-------------------

Domain: mynamespace

Cartridges: frb-websphere-8.5, frb-oracle-12.0 Gear Size: default

Scaling: no

Creating application 'testkjf01' ... done

A instance has successfully be configured on the Oracle Database. Please make note of these credentials:

Script Result: SUCCESS@@oraclesrvr001@@1521@@tcdb\_001

Username: adminin5LXMY

Password: 5XHV5JB2XteQ

Tenant ID: tcdb\_001

To create a scalable app, you have to add the -s option. The namespace mynamespace 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.

8

**3. Viewing WebSphere Logs**

The WebSphere logs can be viewed in two ways, either via SSHing to the OpenShift gear, or by using the Red Hat Client tools log tailing functionality:

1. SSHing to the Gear: Run the command:

rhc ssh <app-name> -n namespace

You will then be SSH’ed into the gear of where WebSphere is running.

The SystemOut.log and SystemErr.log are located under:

./app-root/data/profile/logs/server1/SystemOut.log and

./app-root/data/profile/logs/server1/SystemErr.log

2. Using the RHC Client Tail Command:

You may tail the WebSphere **SystemOut.log** and **SystemErr.log** files with the tailing feature that ships in the Red Hat OpenShift Client Tools. To run the tail execute the command:

rhc tail -a <app-name> -n <namespace>

A tail of the SystemOut.log and SystemErr.log files will be shown.

9

**4. Running WSADMIN Utility in the OpenShift Gear**

Each WebSphere application/gear gets created is its own separate container on the PaaS. This means that the same "VM" where this container is running needs to be shared amongst all the containers, including its IP.

OpenShift relies on the local loopback address to share out the one public IP and proxies requests back to each individual WebSphere container that is binded to the local loopback address. Each WebSphere container will get an IP address in the 127.x.x.x range.

WebSphere users and administrators might be interested in running the **wsadmin** utility to execute admin type configurations against WebSphere. This can be done by SSHing to the OpenShif WebSphere gear that was created and running the **wsadmin.sh** script from there. Since the WebSphere server inside the gear is binded to an address in the 127.x.x.x range and not 0.0.0.0 or the public IP of the node where the gear is running we need to pass that IP to the **wsadmin** utility. We can also, edit the properties file that the **wsadmin** utility reads the host where it needs to connect to so that the **wsadmin.sh** script can be executed without any parameters.

1. SSH to the gear via the RHC client tools: rhc ssh <app-name> -n namespace

Then, change your current working directory to ./app-root/data/profile/bin

1. Find the IP that WebSphere is binded to. This can be done by run the command: env | grep OPENSHIFT\_WEBSPHERE\_IP
2. You can then run the wsadmin.sh script and pass the **-host** option as so

./wsadmin.sh -host $OPENSHIFT\_WEBSPHERE\_IP

or change the value of the host in the properties file that the wsadmin.sh script reads parameters from. This file is located under:

./app-root/data/profile/properties/wsadmin.properties

Replace the values com.ibm.ws.scripting.host & com.ibm.ws.scripting.ipchost with the OPENSHIFT\_WEBSPHERE\_IP from above.

This would allow for the **wsadmin.sh** script to be executed without any flags.

10

**5. Deploying Applications to WebSphere**

All applications deployments should be done via the WebSphere 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.

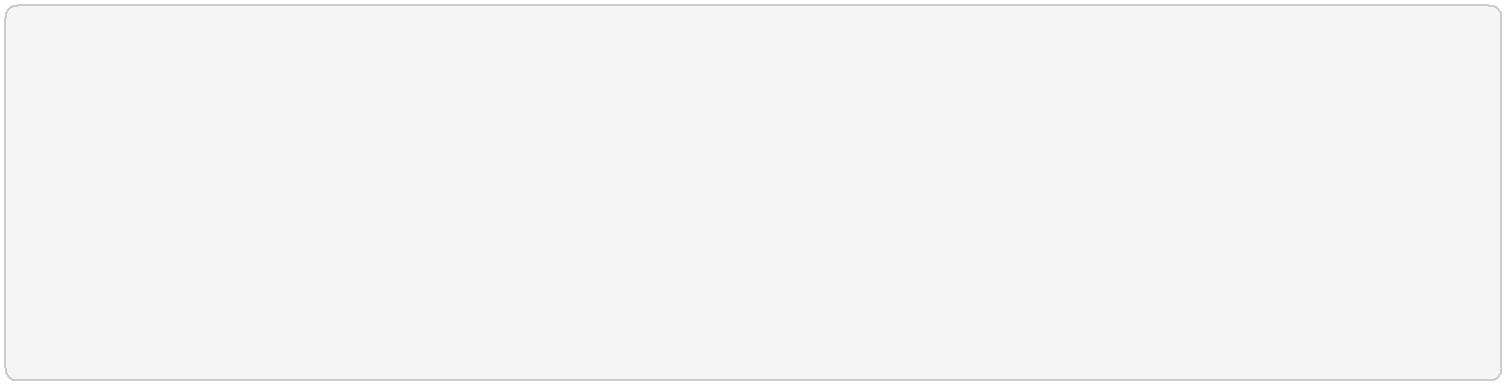
WebSphere comes pre-loaded with a set of default applications. When creating a **scaled** WebSphere gear, OpenShift creates an HA Proxy gear alongside it. When web traffic increases to the main gear another identical gear is created and more are added as necessary to handle the traffic. This is done 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 multiple 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.

11