RED HAT CONSULTING

WebSphere Cartridge User Guide

PREPARED FOR - FRIT

Table of Contents

History and Revisions

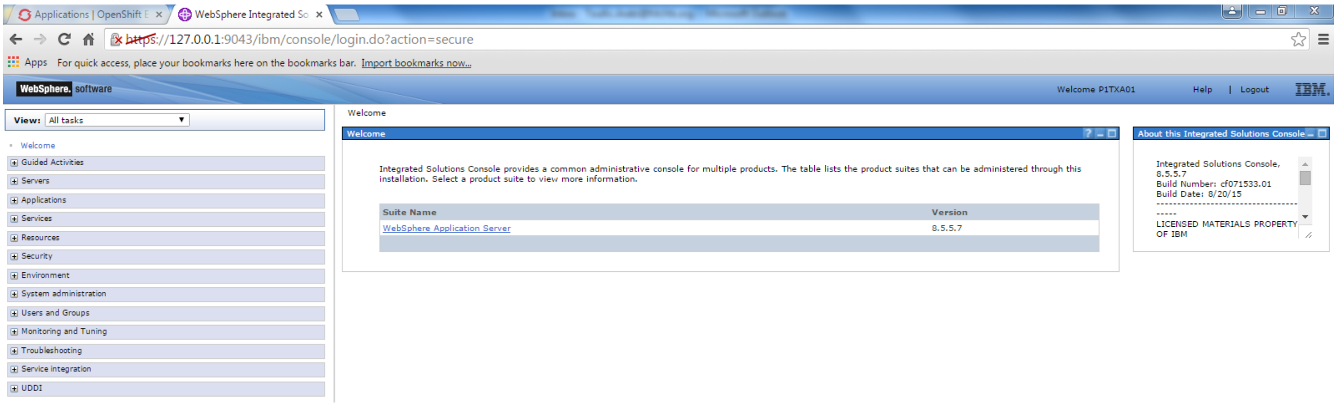
Version	Date	Authors	Changes
0.5	10/08/2015	Toufic Arabi tarabi@redhat.com	Initial Draft

1. Synopsis

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



Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	8.5.5.7

Integrated Solutions Console, 8.5.5.7
Build Number: cf071533.01
Build Date: 8/20/15

LICENSED MATERIALS PROPERTY OF IBM

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

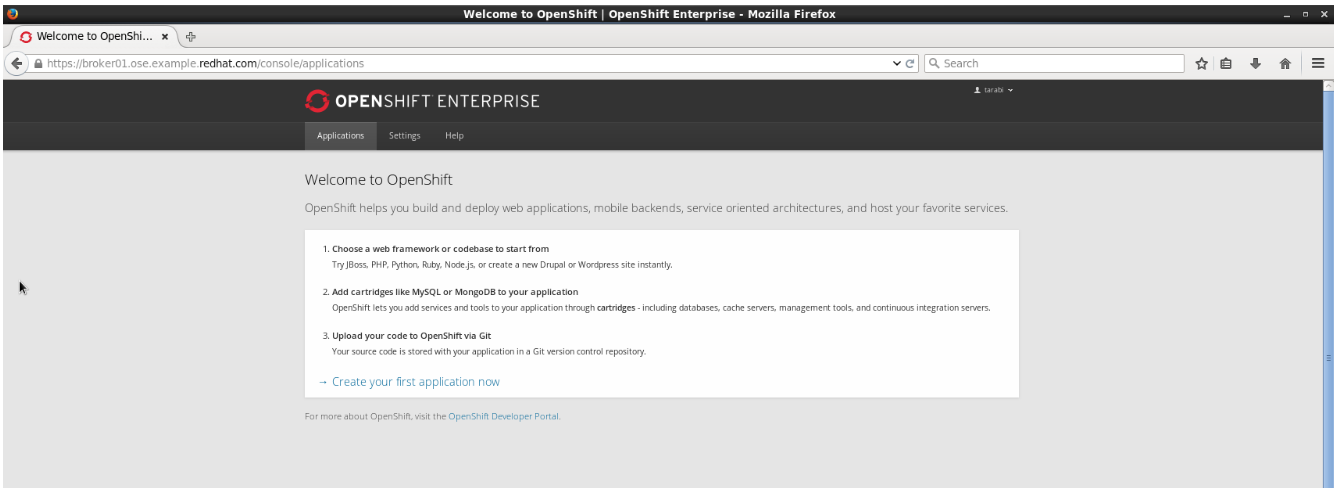
2.1. Creating WebSphere Application Via the OpenShift Console

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

> 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.

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.



Welcome to OpenShift | OpenShift Enterprise - Mozilla Firefox
Welcome to OpenShi...
https://broker01.ose.example.redhat.com/console/applications
Search
OPENSHIFT ENTERPRISE
tarabi
Applications
Settings
Help

Welcome to OpenShift

OpenShift helps you build and deploy web applications, mobile backends, service oriented architectures, and host your favorite services.

1. **Choose a web framework or codebase to start from**
   Try JBoss, PHP, Python, Ruby, Node.js, or create a new Drupal or Wordpress site instantly.
2. **Add cartridges like MySQL or MongoDB to your application**
   OpenShift lets you add services and tools to your application through **cartridges** - including databases, cache servers, management tools, and continuous integration servers.
3. **Upload your code to OpenShift via Git**
   Your source code is stored with your application in a Git version control repository.

→ Create your first application now

For more about OpenShift, visit the OpenShift Developer Portal.

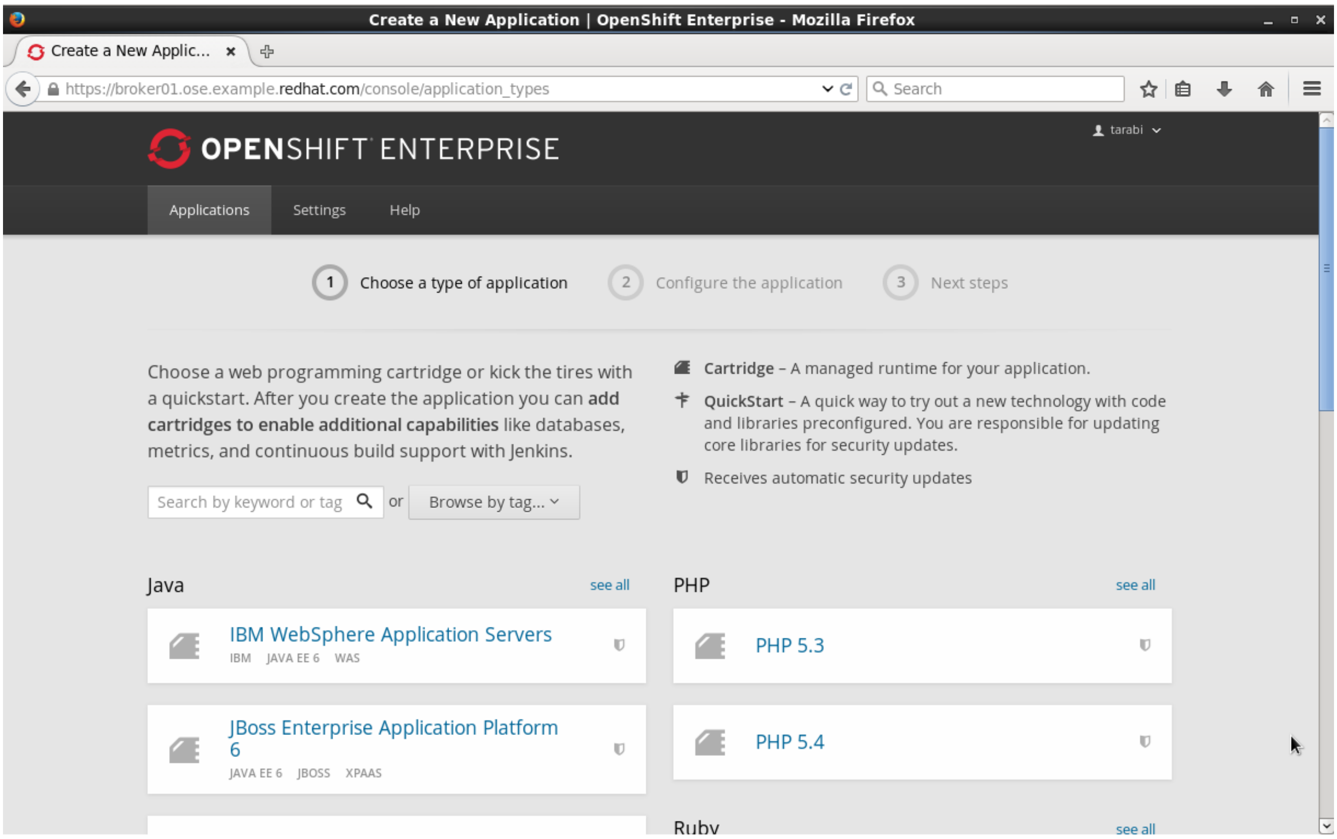
Figure 2: OpenShift Login Page

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.



Create a New Application | OpenShift Enterprise - Mozilla Firefox
Create a New Applic...
https://broker01.ose.example.redhat.com/console/application_types
Search
OPENSHIFT ENTERPRISE
tarabi
Applications
Settings
Help
1 Choose a type of application
2 Configure the application
3 Next steps

Choose a web programming cartridge or kick the tires with a quickstart. After you create the application you can **add cartridges to enable additional capabilities** like databases, metrics, and continuous build support with Jenkins.

Cartridge – A managed runtime for your application.

QuickStart – A quick way to try out a new technology with code and libraries preconfigured. You are responsible for updating core libraries for security updates.

Receives automatic security updates

Figure 3: OpenShift Available Cartridges Menu

2.1.3. Complete the Application Form

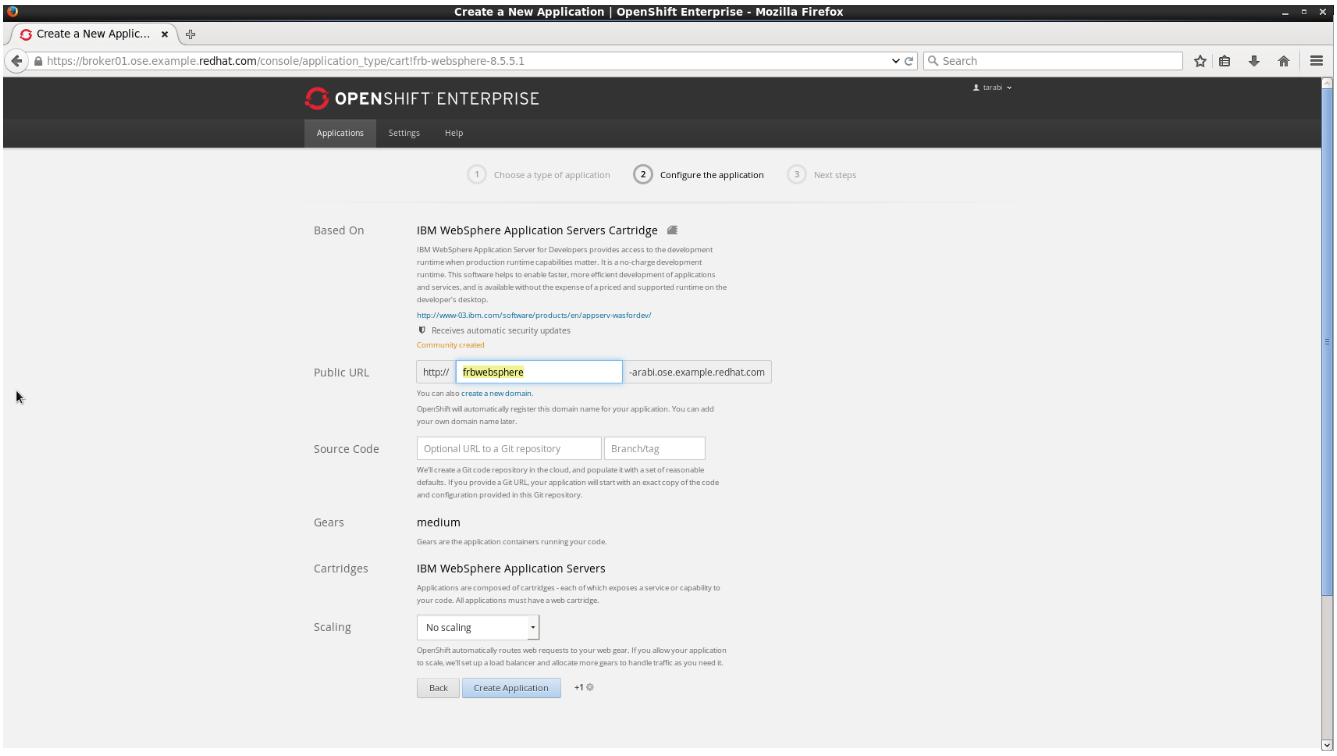
On the next page you will be presented with a form that has a few fields to be completed:

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 your application 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
Create a New Applic...
https://broker01.ose.example.redhat.com/console/application_type/cart!frb-websphere-8.5.5.1
Search
OPENSHIFT ENTERPRISE
tarabi
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 is 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-wasfordev/

Receives automatic security updates

Community created

Public URL: http:// frbwebsphere -arabi.ose.example.redhat.com

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: Optional URL to a Git repository | Branch/tag

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

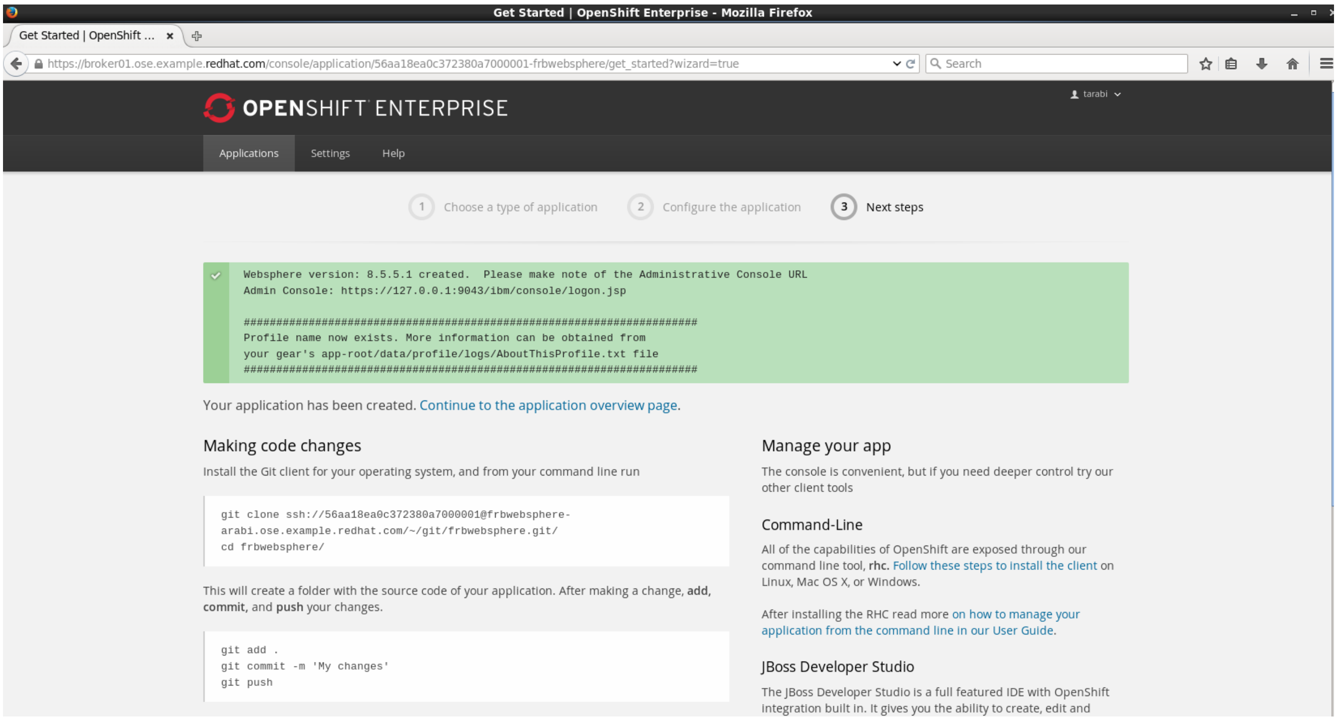
OpenShift automatically routes web requests to your web gear. If you allow your application to scale, we'll set up a load balancer and allocate more gears to handle traffic as you need it.

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
Get Started | OpenShift ...
https://broker01.ose.example.redhat.com/console/application/56aa18ea0c372380a7000001-frbwebsphere/get_started?wizard=true
Search
OPENSHIFT ENTERPRISE
tarabi
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://127.0.0.1:9043/ibm/console/logon.jsp

########################################################################
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://56aa18ea0c372380a7000001@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
```

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

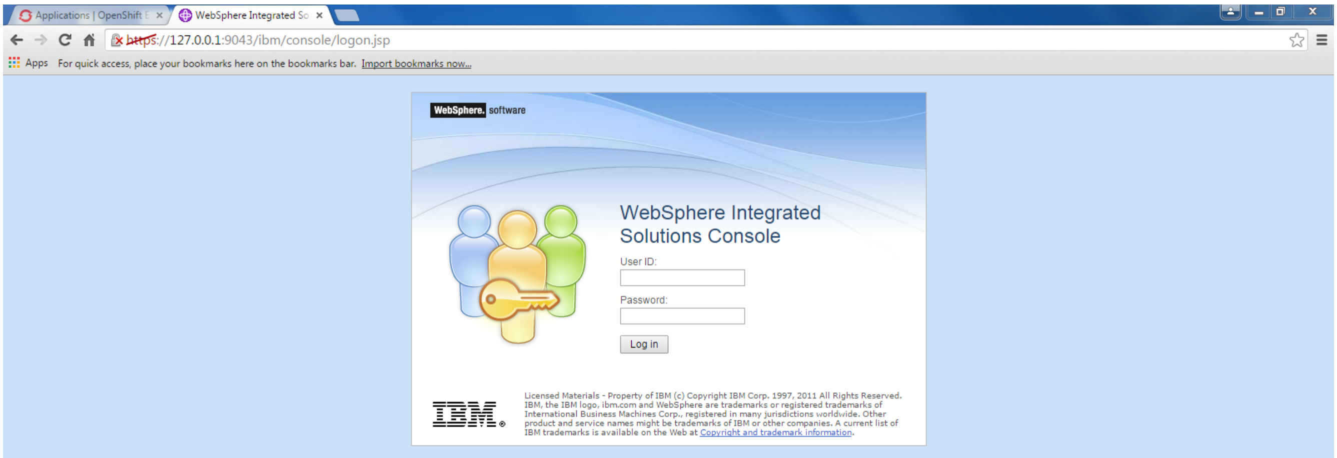
Figure 5: WebSphere Gear Creation Confirmation and Console Details

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
```

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


Applications | OpenShift
WebSphere Integrated So
https://127.0.0.1:9043/ibm/console/logon.jsp
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
WebSphere. software
WebSphere Integrated Solutions Console
User ID:
Password:
Log in
IBM.

Licensed Materials - Property of IBM (c) Copyright IBM Corp. 1997, 2011 All Rights Reserved. IBM, the IBM logo, ibm.com and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information.

Figure 6: WebSphere Administration Console Username/Password Challenge

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.

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**

```
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:

```
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: admininn5LXMY
        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.

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.

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`

2. Find the IP that WebSphere is binded to. This can be done by run the command: `env | grep OPENSHIFT_WEBSPHERE_IP`

3. 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.

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.