OpenShift Developer Guide

USER GUIDE

Preeti Walde
RGEN SOLUTIONS |15 JANUARY, 2015

Table of Contents

1. About OpenShift Client Tools 2
2. Installing Client Tools 2
    2.1 Install Ruby with RubyInstaller 2
    2.2 Install Git version control 4
    2.3 Install the rhc Ruby gem 6
3. Configuring Client Tools 6
    3.1 Login Information 7
    3.2 Authorization Tokens 7
    3.3 Configuration File 8
    3.4 SSH Keys 8
    3.5 Installed Software 9
    3.6 Domain 9
    3.7 Applications 10
4. Creating an Application 10
5. Clone Application to Local Repository 13
6. Visual Studio Plugin 13
    6.1 Install Visual Studio Plugin 14
7. Deploy Changes 14
8. Adding MS SQL Server Cartridge in Application 17
9. Connect to OpenShift Hosted Database Using Microsoft SQL Server Management Studio 18

Copyright @ 2015 All Rights Reserved.

1. About OpenShift Client Tools

You can interact with OpenShift with either the Management Console or the client tools. The Management Console is best for easily launching, managing, and monitoring your applications using a graphical interface. The OpenShift client tools let you manage your cloud environment using a command line interface, and provide more features that are not currently accessible from the Management Console. The client tools are best for coding, debugging, and advanced application management.

2. Installing Client Tools

The OpenShift Client Tools make it easy to create and deploy applications, manage domains, control access to OpenShift applications, and provide complete control of your cloud environment.

Prerequisites

You must have sufficient privileges to install software on your computer. Depending on specific user permissions, disabling the User Account Control (UAC) on Windows Vista/Windows 7/Windows 8 operating systems may be necessary.

Installing Required Software

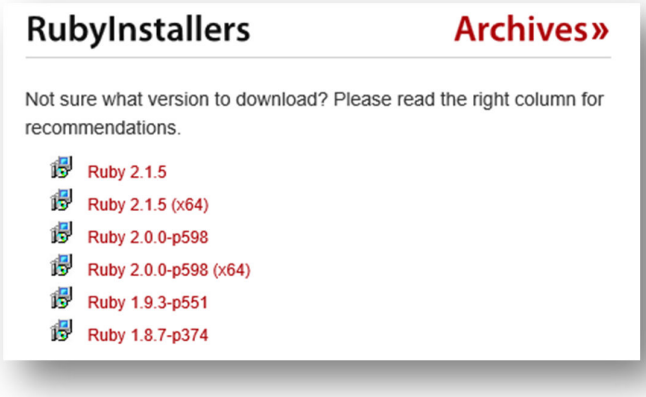
Before you can install OpenShift client tools on Windows operating systems, you must download and install the following software:

1. Install **Ruby** with RubyInstaller
2. Install **Git** version control
3. Install the **rhc** Ruby gem

2.1 Install **Ruby** with RubyInstaller

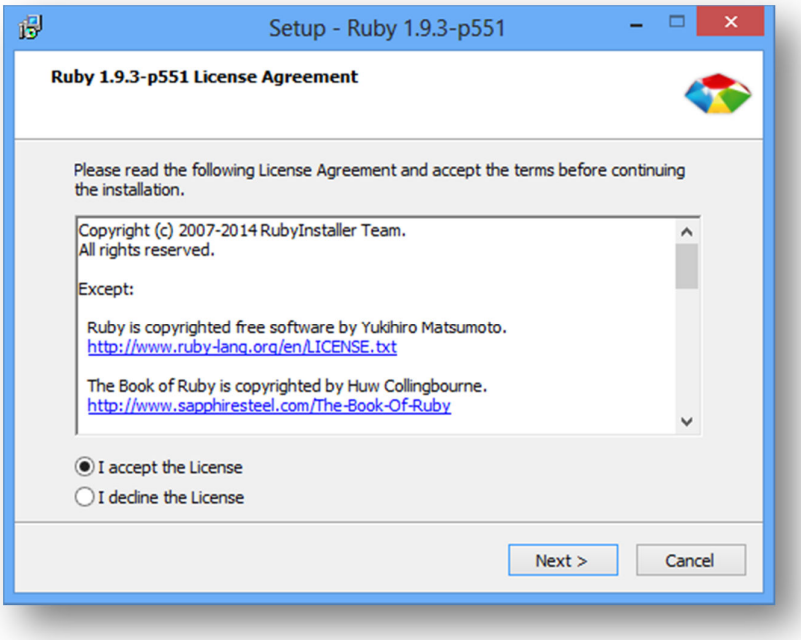
Follow the instructions below to install RubyInstaller for Windows.

1. From *rubyinstaller.org*, download and save the latest RubyInstaller package for Windows to your desired directory and launch the installer.

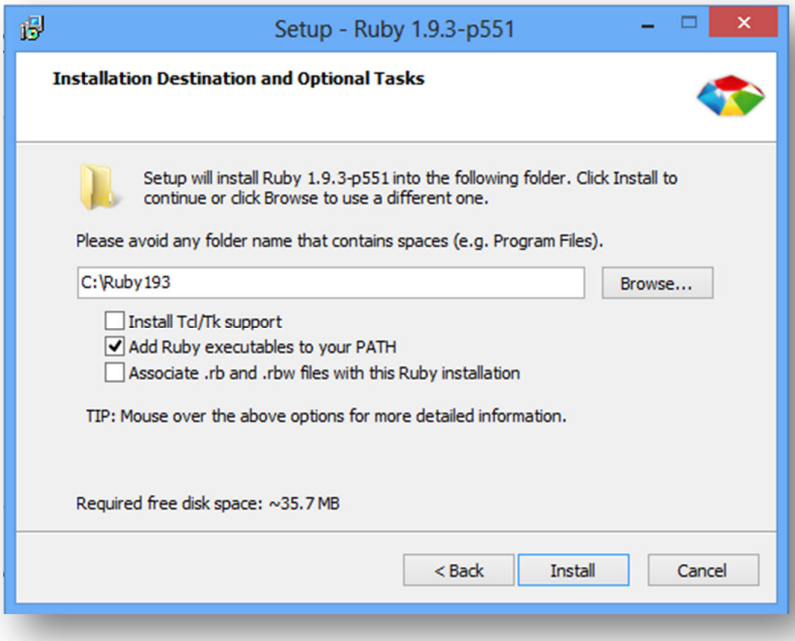


Copyright @ 2015 All Rights Reserved.

2. Double-click the executable RubyInstaller file you downloaded to launch the RubyInstaller installation wizard. Accept the Ruby License Agreement and click **Next**.



3. During the installation you can accept all of the defaults, but it is mandatory that you select the **Add Ruby executables to your PATH** check box in order to run Ruby from the command line, and click **Install** to begin the installation.



Copyright @ 2015 All Rights Reserved.

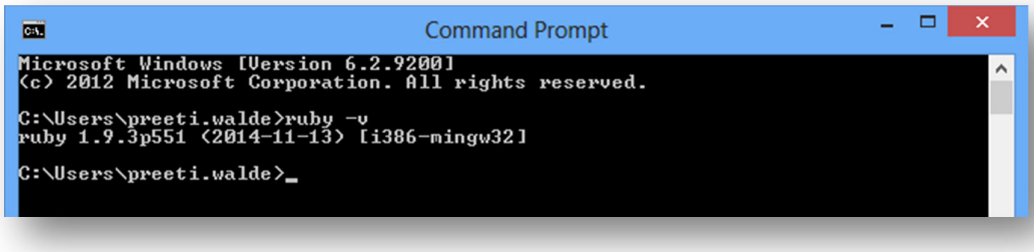
4. Click **Finish** when the install completes.

5. After the installation is completed, to verify that the installation is working, you can open a Command Prompt and run:

   C:\> ruby –v

6. You will get your installed Ruby version:

   ruby 1.9.3p551 <2014-11-13> [i386-mingw32]



   If the Ruby version message does not display, the Ruby executable may not have been added to the path. Restart the installation process and ensure the **Add Ruby executables to your PATH** check box is selected.

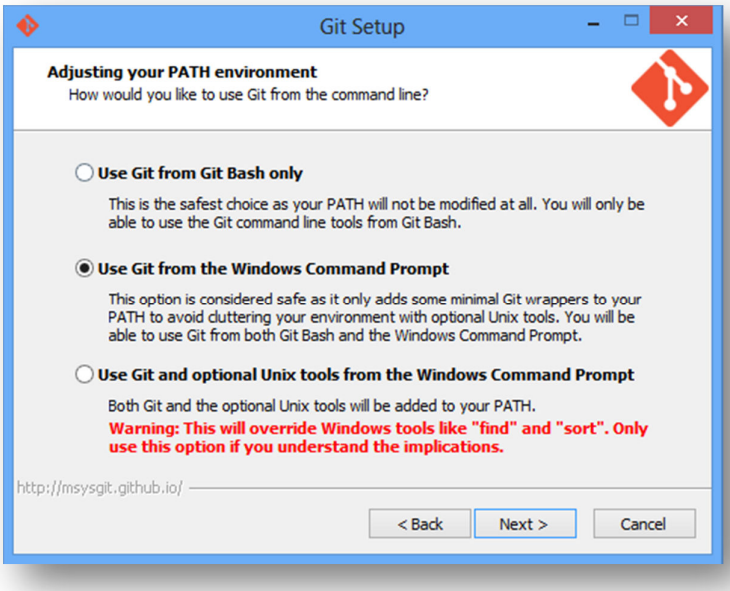
2.2 Install Git version control

After you have installed RubyInstaller, use the following process to install Git version control for Windows.

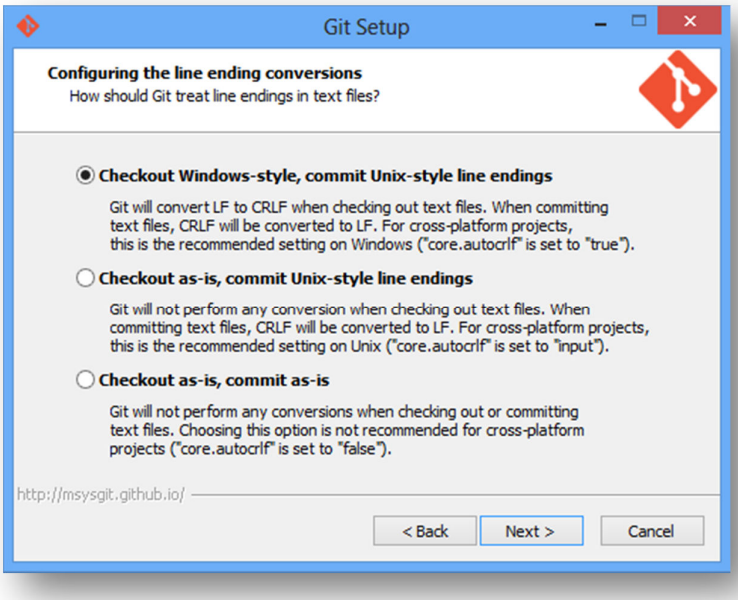
1. From *http://msysgit.github.com/*, download and save the latest version of Git for Windows to your desired directory.
2. Double-click the executable Git file you downloaded to launch the Git installation wizard. Click **Next** until you are asked to adjust your PATH environment.

Copyright @ 2015 All Rights Reserved.

3. Select the **Use Git from the Windows Command Prompt** check box and click **Next**. This enables you to use the default Windows command prompt for Git and SSH operations.



4. Select the **Checkout Windows-style, commit Unix-style line endings** check box, and click **Next**.



5. Click **Finish** when the install completes.

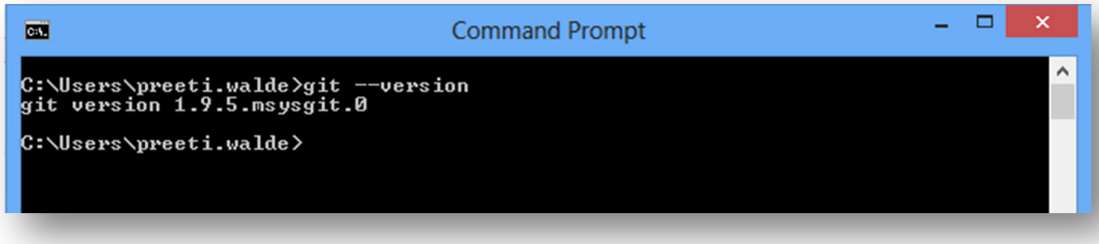
Copyright @ 2015 All Rights Reserved.

6. After the installation is completed, to verify that Git is correctly configured run:

   C:\> git –version

7. If Git was installed correctly, you should see the installed version number:

   git version 1.9.5.msysgit.0

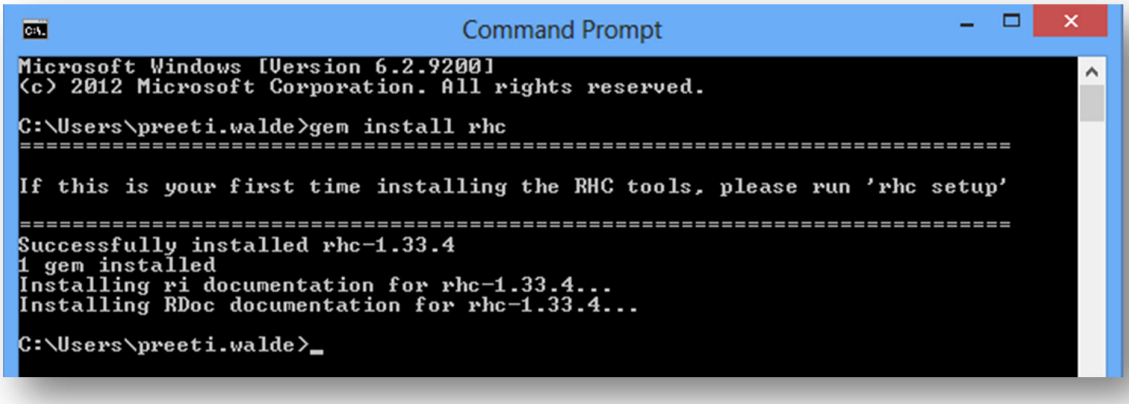


2.3 Install the **rhc** Ruby gem

After you have installed ***RubyInstaller*** and ***Git***, use the following instructions to install the client tools.

1. Open Windows Command Prompt.
2. Run the following command to install the client tools:

   C:\> gem install rhc



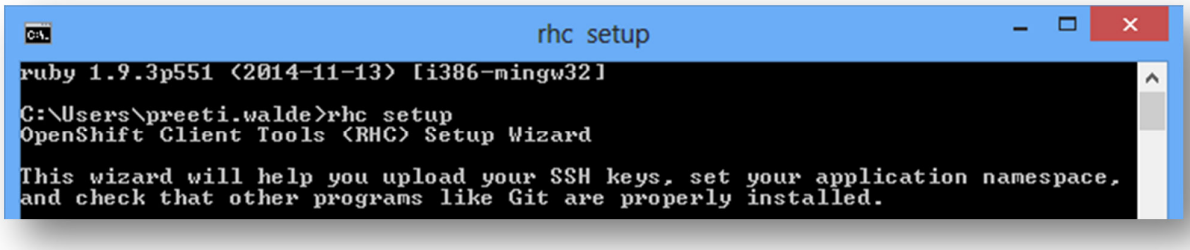
3. Configuring Client Tools

After you have installed the OpenShift Online client tools on your computer, they must be configured before you can create and deploy applications.

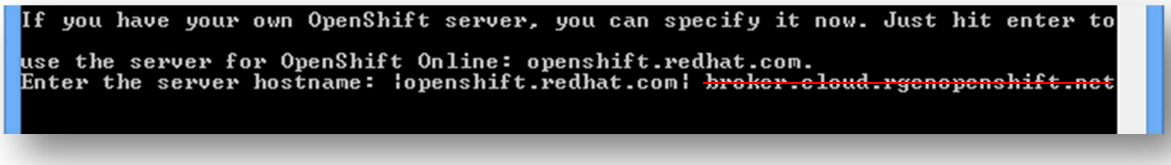
The setup wizard can be launched by running the **rhc setup** command:

Copyright @ 2015 All Rights Reserved.

C:\> rhc setup



Append the server hostname on which you want to login after default server hostname for example:



'broker.example.com'.

3.1 Login Information

You will be prompted to provide your login credentials you used to sign up for OpenShift.

Login to broker.example.com: user@example.com
Password: password

3.2 Authorization Tokens

The setup wizard next prompts you to generate an authorization token. Answering ***yes*** will store a token in your home directory to be used on subsequent requests. When it expires, you will be prompted for your password again.

OpenShift can create and store a token on disk which allows to you to access the server without using your password. The key is stored in your home directory and should be kept secret. You can delete the key at any time by running 'rhc logout'.
Generate a token now? (yes|no) ***yes***
Generating an authorization token for this client ... lasts about 1 day

Copyright @ 2015 All Rights Reserved.

3.3 Configuration File

After you have entered your login credentials, the setup wizard will create the **express.conf** configuration file in your home directory. In the following example, the **express.conf** file has been created in the **C:\Users\User1\.openshift** directory.

```
Saving configuration to C:\Users\User1\.openshift\express.conf…done
```

3.4 SSH Keys

Next, the setup wizard will configure the SSH keys that will be used to authenticate your computer with the remote server. Because this is the initial configuration, it is assumed no existing SSH keys are found on your computer. Therefore, the setup wizard will generate a new pair of SSH keys, named **id_rsa** and **id_rsa.pub**, and save them in your home directory. In the following example the SSH keys have been generated and saved in the **C:\Users\User1\.ssh** directory.

```
No SSH keys were found. We will generate a pair of keys for you.
Created: C:\Users\User1\.ssh\id_rsa.pub
```

After the new SSH keys have been generated, the public key, **id_rsa.pub**, must be uploaded to the OpenShift server to authenticate your system to the remote server. You can enter a name to use for your key, or leave it blank to use the default name. In the following example the default name is used.

```
Your public ssh key must be uploaded to the OpenShift server to access code. Upload now? (yes|no) yes

Since you do not have any keys associated with your OpenShift account, your new key will be uploaded
as the 'default' key

Uploading key 'default' from C:\Users\User1\.ssh\id_rsa.pub ... done
```

Copyright @ 2015 All Rights Reserved.

3.5 Installed Software

The setup wizard will verify if you have ***Git*** version control software installed on your computer. ***Git*** is required to create and deploy applications to the OpenShift cloud environment. In the example below, the setup wizard verifies that ***Git*** for Windows has been installed correctly.

In order to fully interact with OpenShift you will need to install and configure a git client if you have not already done so.
Documentation for installing other tools you will need for OpenShift can be found at https://openshift.redhat.com/community/developers/install-the-client-tools We recommend these free applications:

* Git for Windows - a basic git command line and GUI client https://github.com /msysgit/msysgit/wiki/InstallMSysGit
* TortoiseGit - git client that integrates into the file explorer http://code.google.com/p/tortoisegit/

3.6 Domain

The setup wizard will next check whether you have created a domain for your OpenShift cloud environment. If you created your domain when you signed up for an OpenShift account, it will be displayed here. If you wish to create a domain at this time, type the desired name when prompted. Alternatively, you can create a domain using the OpenShift Online Management Console, or using the **rhc domain create** command. The domain ***MyDomain*** has been created in the following example.

Checking for a domain ... none

Your domain is unique to your account and is the suffix of the public URLs we assign to your applications.
You may configure your domain here or leave it blank and use 'rhc domain create' to create a domain later. You will not be able to create applications without first creating a domain.

Please enter a domain (letters and numbers only) |<none>|: ***MyDomain***
Your domain name 'MyDomain' has been successfully created

Copyright @ 2015 All Rights Reserved.

3.7 Applications

Finally, the setup wizard will check whether any applications have been created. Any applications created with the OpenShift Online Management Console will be displayed here. In the example below, no applications have been found, in which case the setup wizard shows the types of applications that can be created with the associated commands. The setup wizard then completes by displaying the current gear consumption along with the gear sizes available to the given user.

```
Checking for applications ... none

Run 'rhc app create' to create your first application.

  Do-It-Yourself                          rhc app create <app name> diy-0.1
  JBoss Application Server 7.1            rhc app create <app name> jbossas-7
  JBoss Enterprise Application Platform 6.0 rhc app create <app name> jbosseap-6.0
  Jenkins Server 1.4                      rhc app create <app name> jenkins-1.4
  Node.js 0.10                            rhc app create <app name> nodejs-0.10
  Node.js 0.6                             rhc app create <app name> nodejs-0.6
  PHP 5.3                                 rhc app create <app name> php-5.3
  Perl 5.10                               rhc app create <app name> perl-5.10
  Python 2.6                              rhc app create <app name> python-2.6
  Python 2.7 Community Cartridge          rhc app create <app name> python-2.7
  Python 3.3 Community Cartridge          rhc app create <app name> python-3.3
  Ruby 1.8                                rhc app create <app name> ruby-1.8
  Ruby 1.9                                rhc app create <app name> ruby-1.9
  Tomcat 6 (JBoss EWS 1.0)                rhc app create <app name> jbossews-1.0
  Tomcat 7 (JBoss EWS 2.0)                rhc app create <app name> jbossews-2.0
   Zend Server 5.6                        rhc app create <app name> zend-5.6

  You are using 0 of 3 total gears
  The following gear sizes are available to you: small

Your client tools are now configured.
```

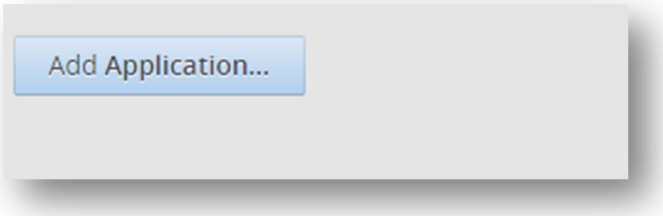
You are now ready to create and manage an application using OpenShift.

4. Creating an Application

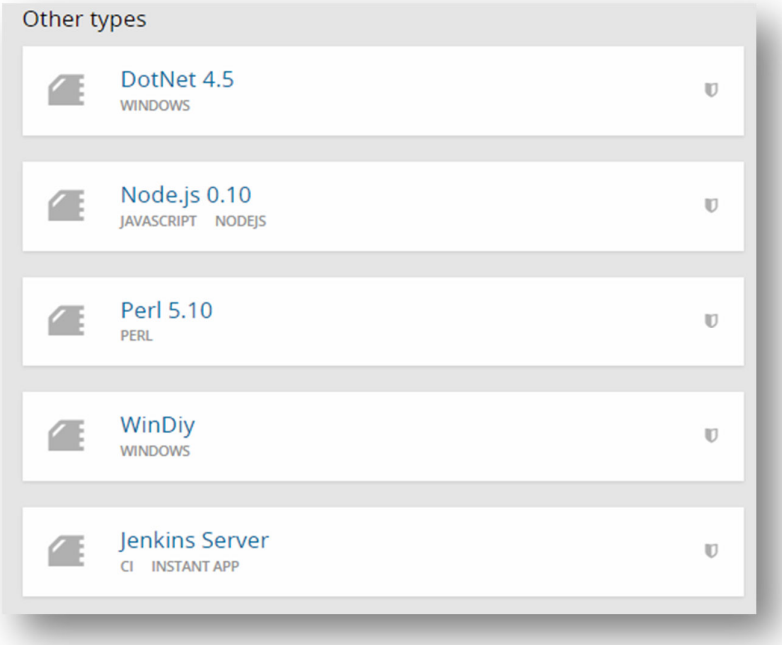
Now that you have rhc installed and configured, you are ready to create an application. You can create a new application with either the management console or the client tools.

1. Login with the domain account into the site.
2. You will be able to see existing application if there created any.
3. To create new application, click **Add Application** button.

Copyright @ 2015 All Rights Reserved.

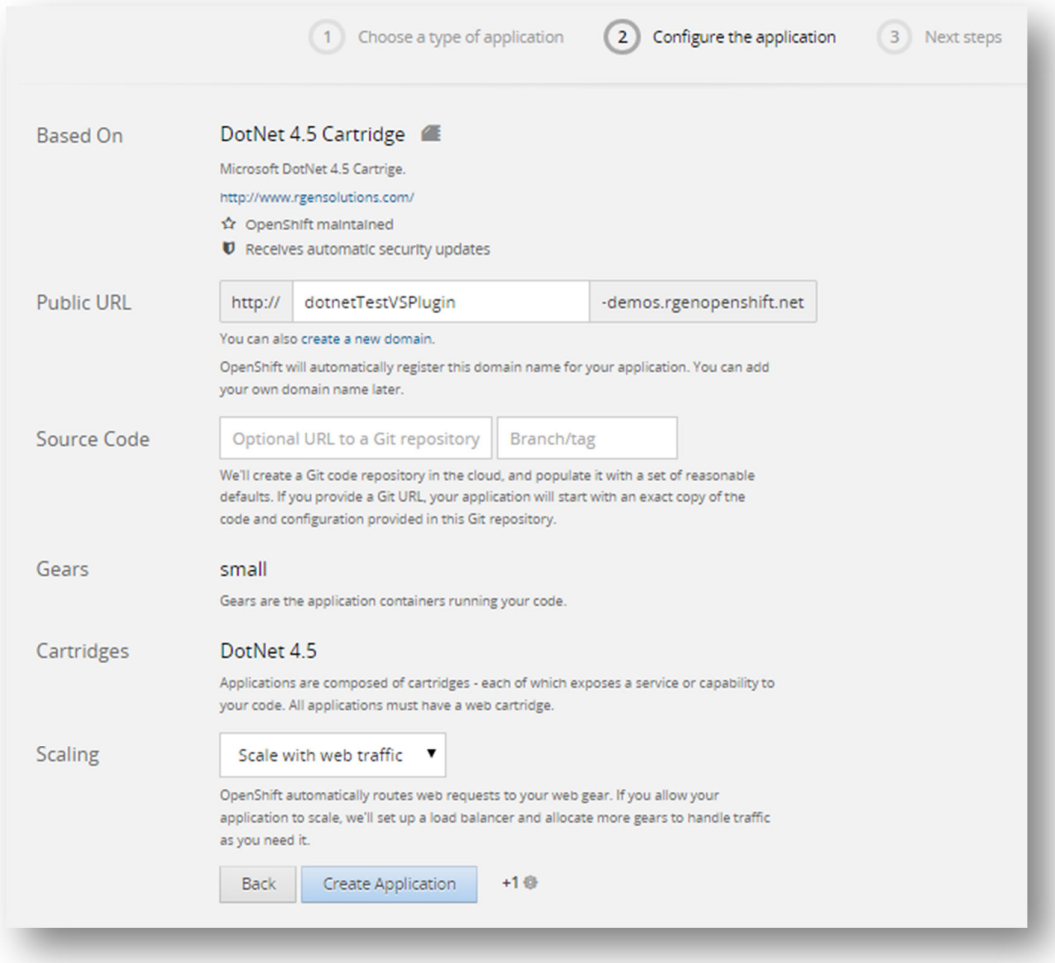


4. You can a list of programming cartridges here to create your application. Select cartridge **DotNet 4.5** from **Other Types** as you are creating DotNet application.

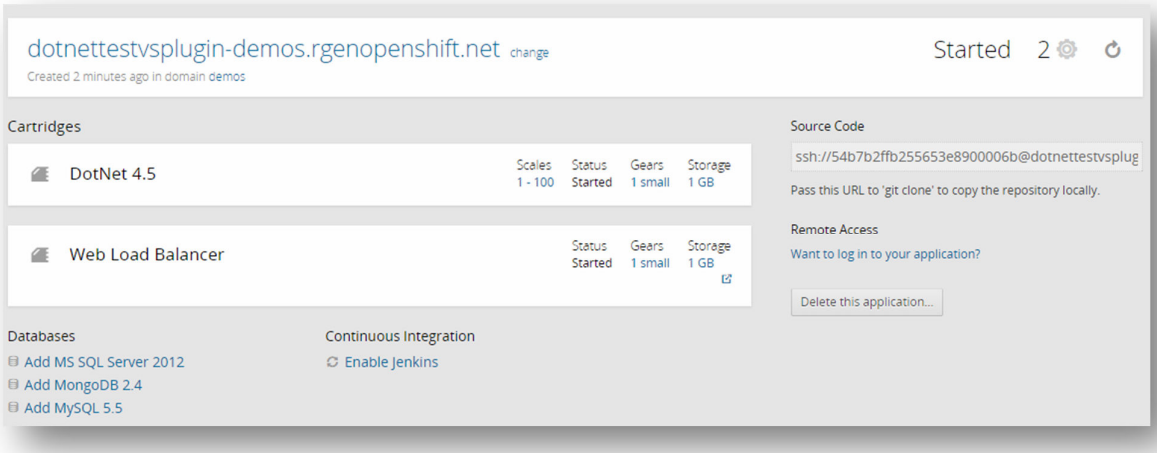


5. Provide name of your application in **Public URL** field. This is a mandatory field as OpenShift will automatically create a website with this name.
6. Select **Scaling with web traffic** option from Scaling dropdown as you are creating **DotNet** application so you will need to select this option.
7. Once you are done with this, click **Create Application**.

Copyright @ 2015 All Rights Reserved.

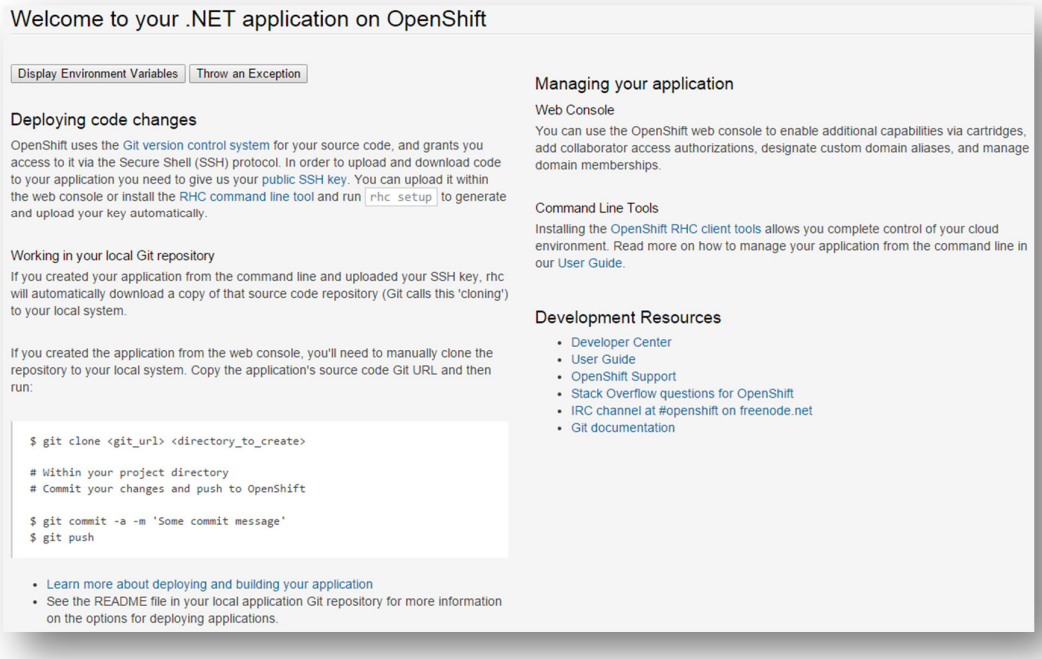


8. Once the application is created, you can see the application page like below. This will show you the application url and the cartridges you have attached.



Copyright @ 2015 All Rights Reserved.

9. Clicking on the url will redirect you on the default page of your created application.



10. Now you have successfully created your application. If you want make any changes in your application, you need to first clone your application into your local repository.

5. Clone Application to Local Repository

If you create an application from the web console, you'll need to tell **Git** to clone the repository. Find the Git URL from the application page, and then run:

C:\> git clone <git_url> <directory to create>

Git will clone the application you created into your local repository (path which you have mentioned). Now you can modify your application as per your requirement by using **Visual Studio Plugin** which is described below.

6. Visual Studio Plugin

Visual studio plugin gives you a flexibility to modify your application and deploy the changes to OpenShift directly.

Prerequisites

While installing plugin, make sure none of the visual studio instance should be opened. This may create problem in installation.

Copyright @ 2015 All Rights Reserved.

6.1 Install Visual Studio Plugin

After you have cloned your application into local repository, use following instruction to install Visual Studio Plugin.

1. Download the latest version of visual studio plugin from *here*.
2. Extract the zip file, click on plugin installer and launch the installation wizard.
3. Accept the License Agreement and click **Install**.

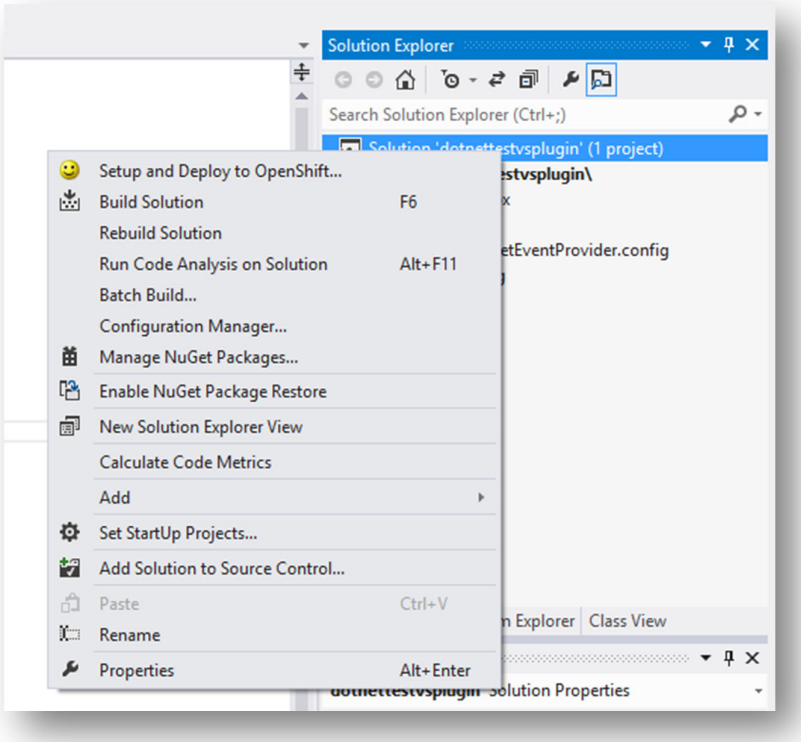
4. Click Finish when the install completes.

7. Deploy Changes

Now you are ready to perform changes in your application which is cloned into your local repository and deploy those changes to OpenShift. Use the following steps to modify the application and deploy the code changes to OpenShift.

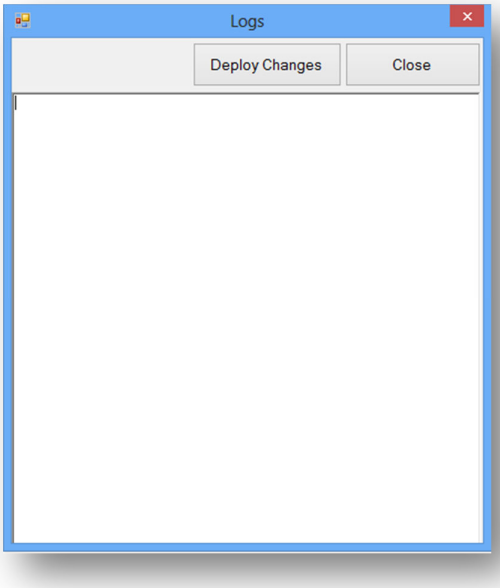
1. Open your cloned application in Visual Studio from local repository.
2. Modify your application by changing anything or adding anything.
3. Once you make changes to your local repository, you need to **add** and **commit** those changes.
4. For that, right click on the solution/project and select **Setup and Deploy to OpenShift**.

Copyright @ 2015 All Rights Reserved.

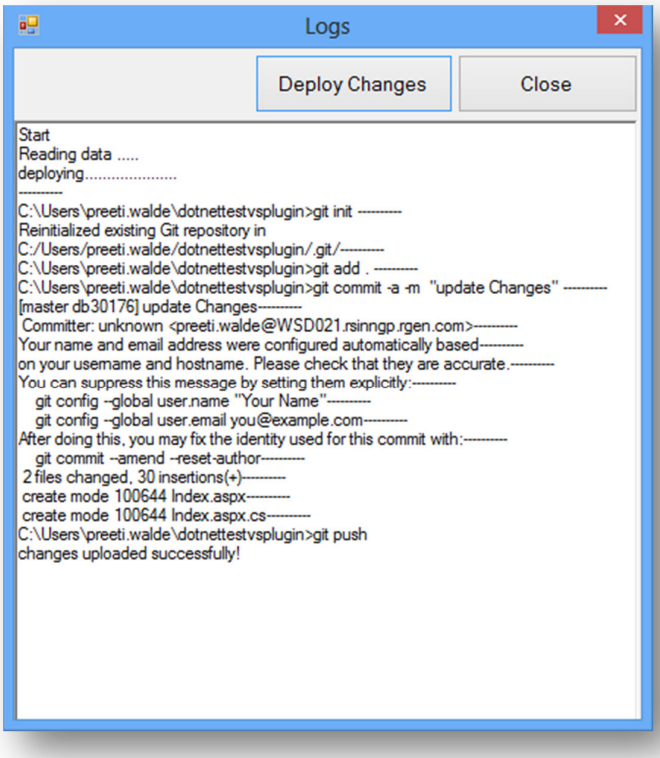


5. After selecting the option **'Setup and Deploy to OpenShift'**, you will be prompted with a **Log** dialog box where you need to add, commit and push your changes.

Copyright @ 2015 All Rights Reserved.



6. Finally, you're ready to send your changes to your application, click on **Deploy Changes** so that it will start reading all the files of your local repository which are added/updated and deploy the changes to OpenShift.



7. Once the changes are deployed, **close** the dialog box.

Copyright @ 2015 All Rights Reserved.

8. Go to your application URL or refresh your application if already opened and see the deployed changes.

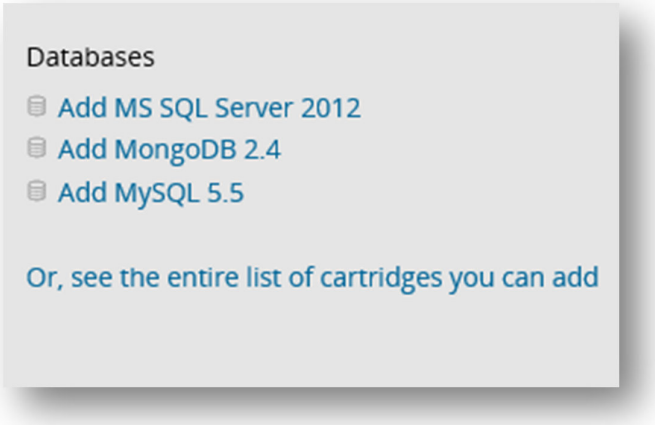
   If you are not getting '**Setup and Deploy to OpenShift**' option, you have not installed the setup properly. To verify this, go to **Tools** Menu, select **Add-in Manager** and check whether the add-in is available or not.

8. Adding MS SQL Server Cartridge in Application

There are two versions of SQL Server supported in RGen OpenShift viz. SQL Server 2008 and SQL Server 2012. Based on requirement, any one version of MS SQL Server Cartridge can be added in an application.

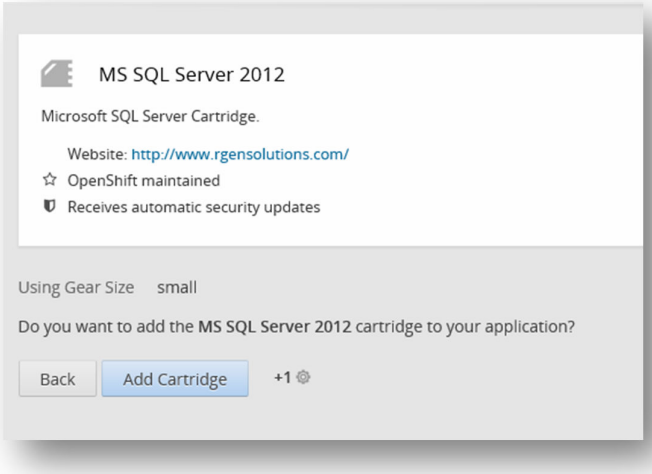
Follow below steps to add MS SQL Server Cartridge in already application.

1. Login with the domain account into the site.
2. You will be able to see application created.
3. Select an application to add MS SQL Server Cartridge from available list of application.
4. Under Databases heading, select the **Add MS SQL Server 2012**, if want to add SQL Server 2012 Database or select **Or, see the entire list of cartridges you can add**.



5. On **Add a Cartridge** Page, Click on **Add Cartridge** button.

Copyright @ 2015 All Rights Reserved.



6. Once cartridge added successfully, the SQL database server information will be prompted as specified below. [**NOTE**: the server information is removed from below screenshot. User will get real database server details after successful addition of SQL Cartridge.]

```
Microsoft SQL Server 2012 database added.  Please make note of database server details to login through Microsoft SQL Server Management Studio:

    database host: {Database Host Name}
    database port: {Database Port}
      sa password: {Password}
    database name: {Database Name}

Connection String for .NET Application:

    Data Source={Database Host Name},{Database Port};Initial Catalog={Database Name};Persist Security Info=True;User ID=sa;Password={Password}
```

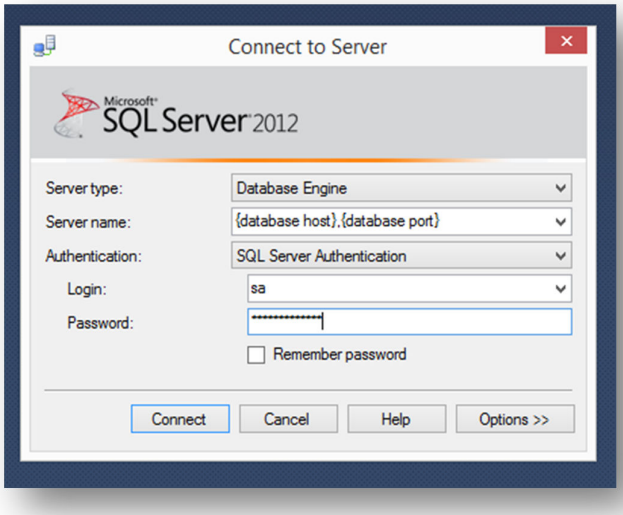
9. Connect to OpenShift Hosted Database Using Microsoft SQL Server Management Studio

The OpenShift hosted database can be connected using Microsoft SQL Server Management Studio just like connecting to any other SQL Database.

Follow below steps to connect to MS SQL Server DB:

1. Open **Microsoft SQL Server Management Studio**. (Use 2012 version for connecting to SQL Server 2012 MS SQL Cartridge DB)
2. In the **Connect to Server** window, specify the database host name and port number in **Server Name**.
3. Select **SQL Server Authentication** in **Authentication** field.
4. Specify **sa** in **Login** section and password of sa in **Password** field.
5. Now click **Connect** button to connect to SQL Database.

Copyright @ 2015 All Rights Reserved.



Copyright @ 2015 All Rights Reserved.