

# Getting Started with Red Hat OpenShift



Rafay Systems  
530 Lakeside Drive, Suite 210  
Sunnyvale, CA 94086  
[rafay.co](http://rafay.co)  
[info@ray.co](mailto:info@ray.co)

# Getting Started with OpenShift

Many Rafay Systems customers use Red Hat OpenShift as their platform of choice for building, deploying, running and scaling Kubernetes containerized applications. OpenShift is flexible and versatile; it can be deployed in public and private clouds, using physical or virtual servers, or both. OpenShift has proven to be of great value for operations and development teams. But getting started with OpenShift can be somewhat daunting.

## Overview

This guide will help you to get started with OpenShift to create an application, configure Continuous Integration (CI) and scale the application on-demand.

First, we will setup a sample Node.js application on an OpenShift cloud. Before starting, you should have the following:

- An OpenShift online account -> <https://www.OpenShift.com/products/online/>
- A GitHub account. -> <https://github.com>
- Git installed locally -> <https://help.github.com/en/articles/set-up-git>

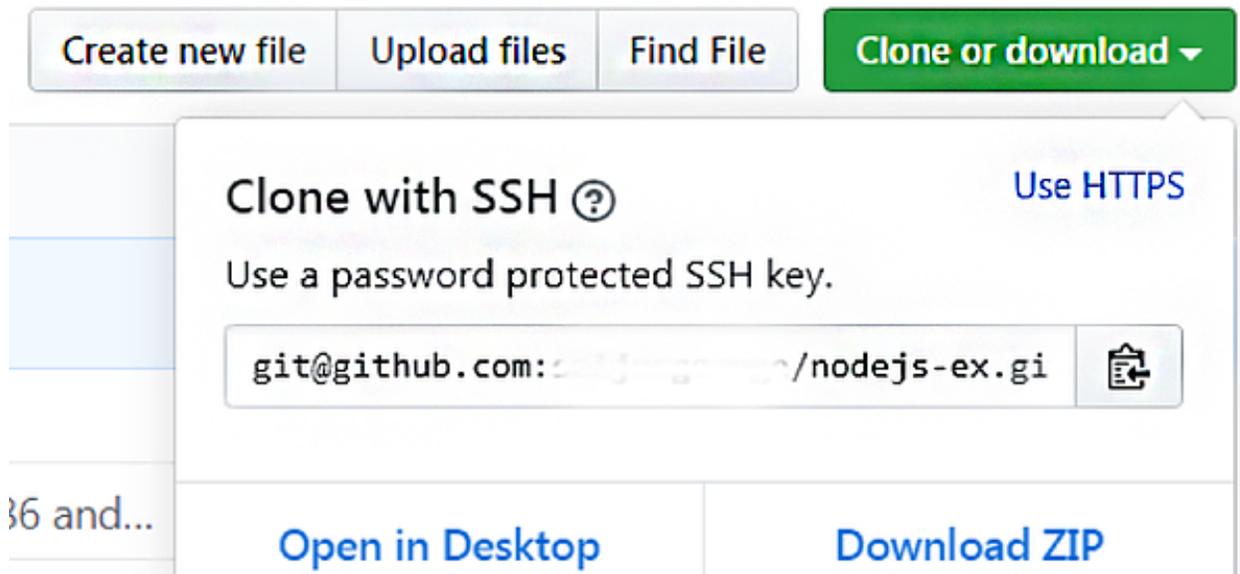
## Basic Setup

To start, you need to fork the OpenShift Node.js sample application on GitHub and clone the repository to your local environment so that you can deploy and make modifications to the application.

1. On GitHub, fork a GitHub repository (OpenShift/nodejs-ex) by clicking on the top-right corner button on the page.



2. Copy the URLs from GitHub



3. Now, execute the command below on your local terminal to check out the sample application and go to the new directory.

```
[user@localhost ~] git]$ git clone git@github.com:username/nodejs-ex.git
Cloning into 'nodejs-ex'...
remote: Enumerating objects: 631, done.
remote: Total 631 (delta 0), reused 0 (delta 0), pack-reused 631
Receiving objects: 100% (631/631), 233.74 KiB | 19.48 MiB/s, done.
Resolving deltas: 100% (243/243), done.
[user@localhost ~] git]$ cd nodejs-ex/
```

Now, you have the local copy of the GitHub repository. You can you can make modifications and deploy on OpenShift.

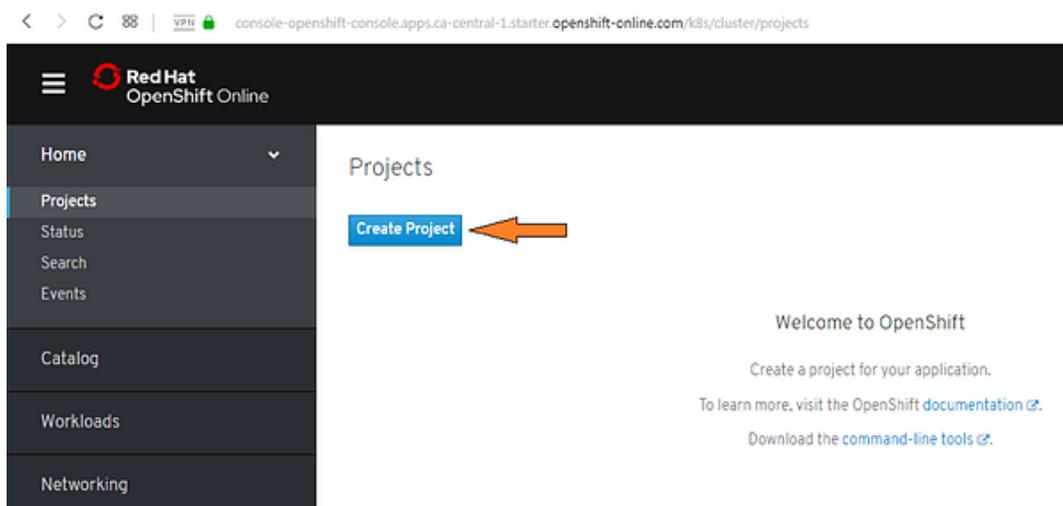
## Getting Familiar with the OpenShift Web Interface

To get familiar with the OpenShift online GUI, visit -  
> [https://docs.openshift.com/online/architecture/infrastructure\\_components/web\\_console.html#architecture-infrastructure-components-web-console](https://docs.openshift.com/online/architecture/infrastructure_components/web_console.html#architecture-infrastructure-components-web-console)

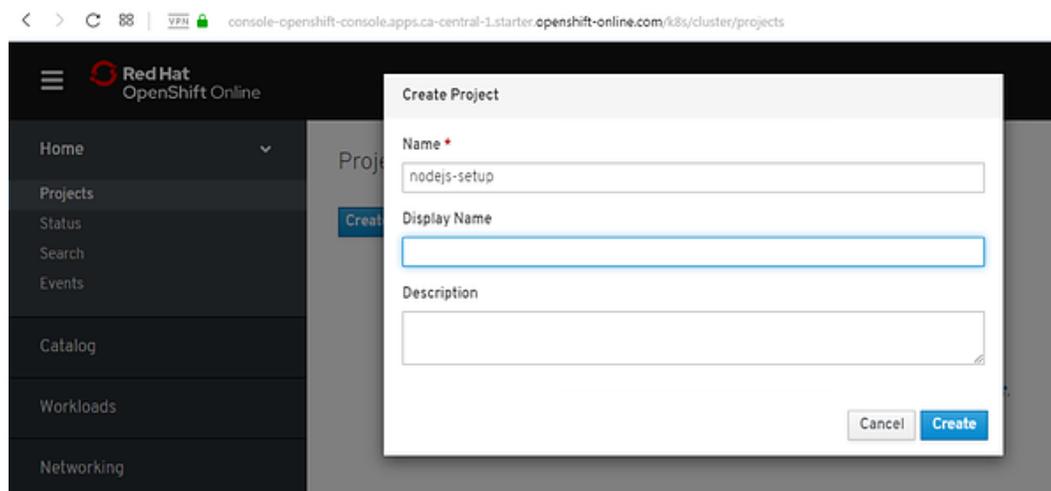
## Build a New Application

To build a new app, setup a Nodejs application using the web interface on OpenShift Online as follows:

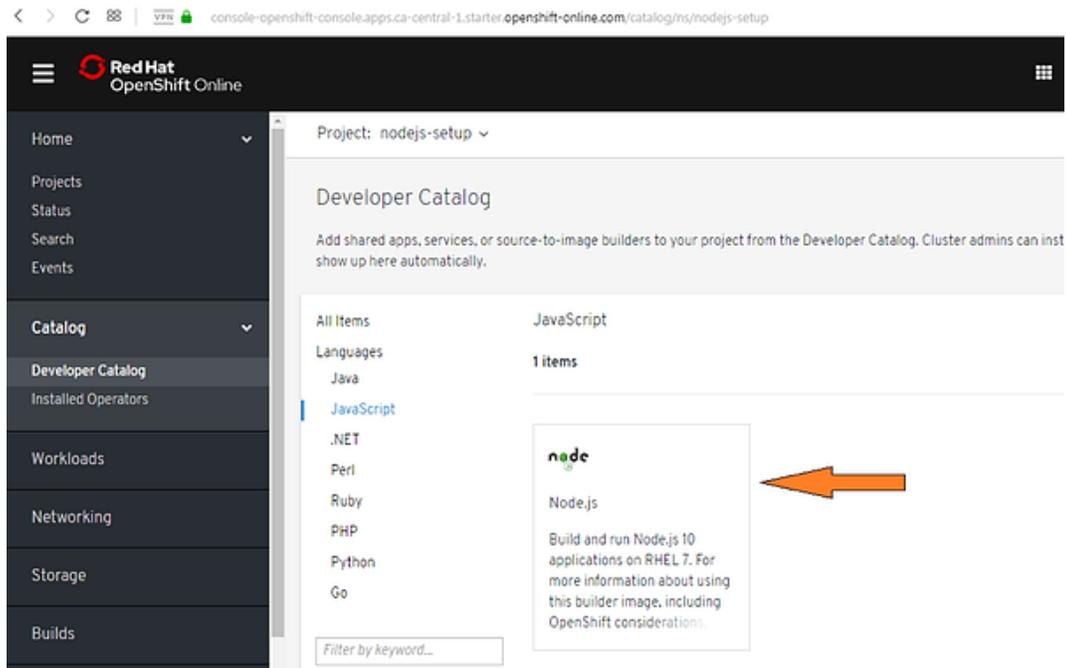
1. Click on "Create Project". If you are using a free account and already have a project, you must delete your existing project as only one project is allowed.



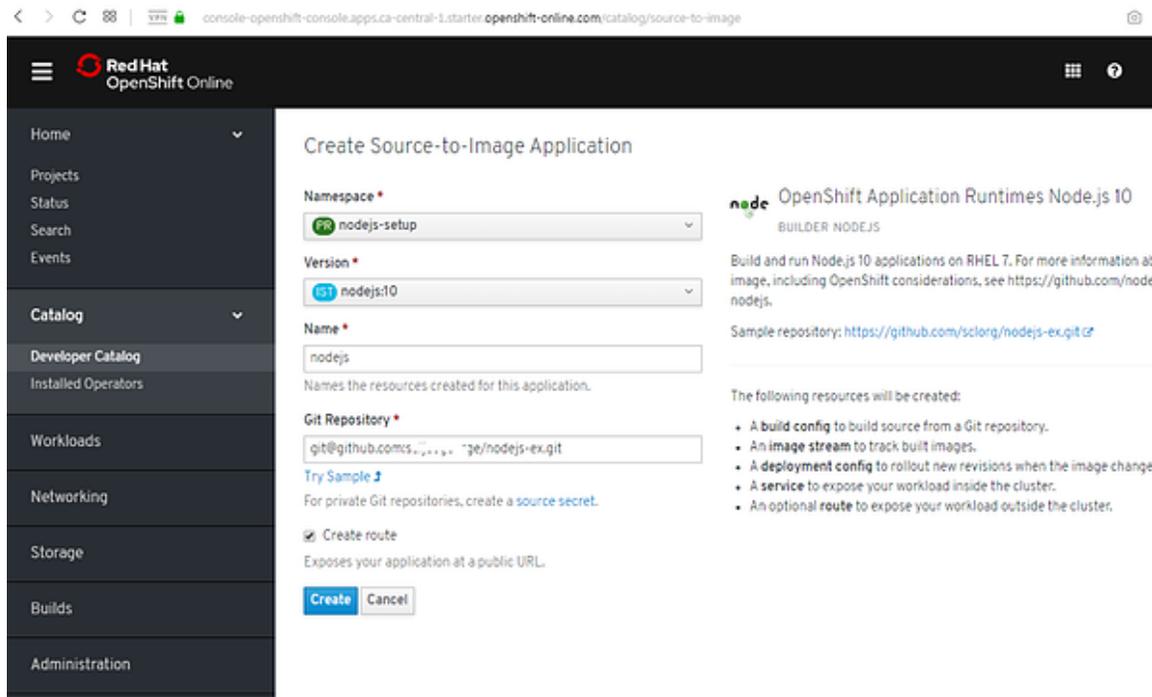
2. Give a unique name to your project on OpenShift. (e.g.: Nodejs-setup). The fields "Display Name" and "Description" are optional.



3. Browse through the developer catalog. Click on JavaScript and select Node.js example as a quick start template.

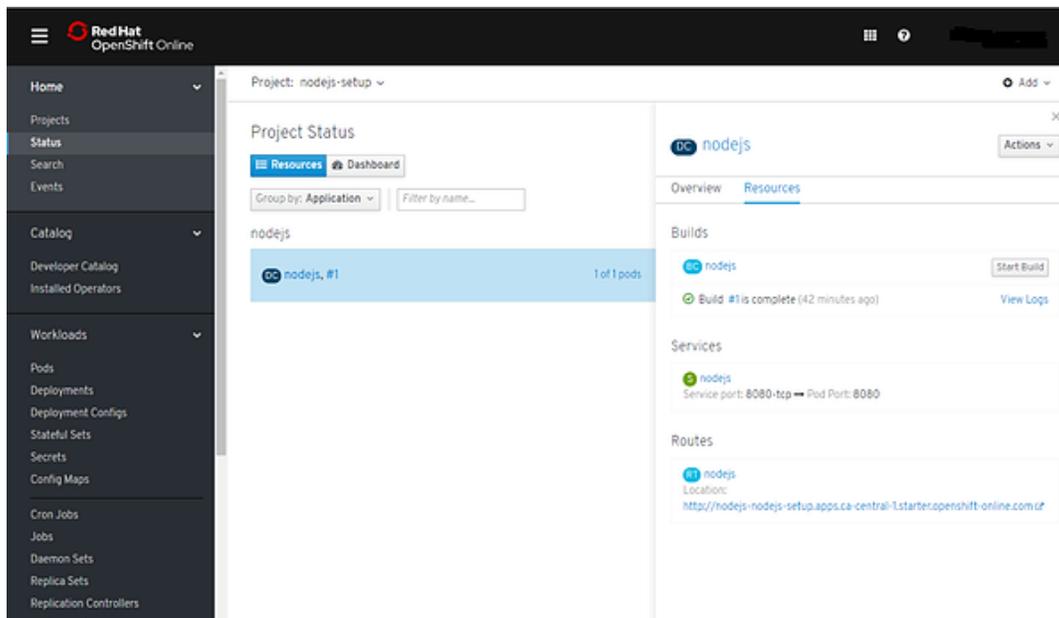


4. Name the resource that will be created by this application and the GitHub URL that was forked. Select the checkbox "create route" to access the application publicly.

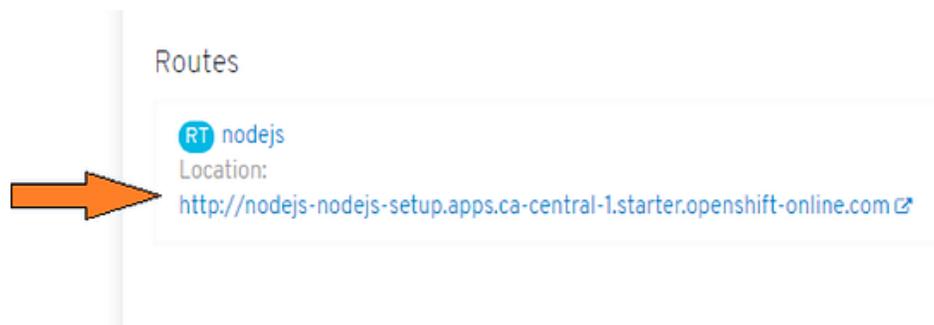


5. Click on 'Create' to deploy your application.

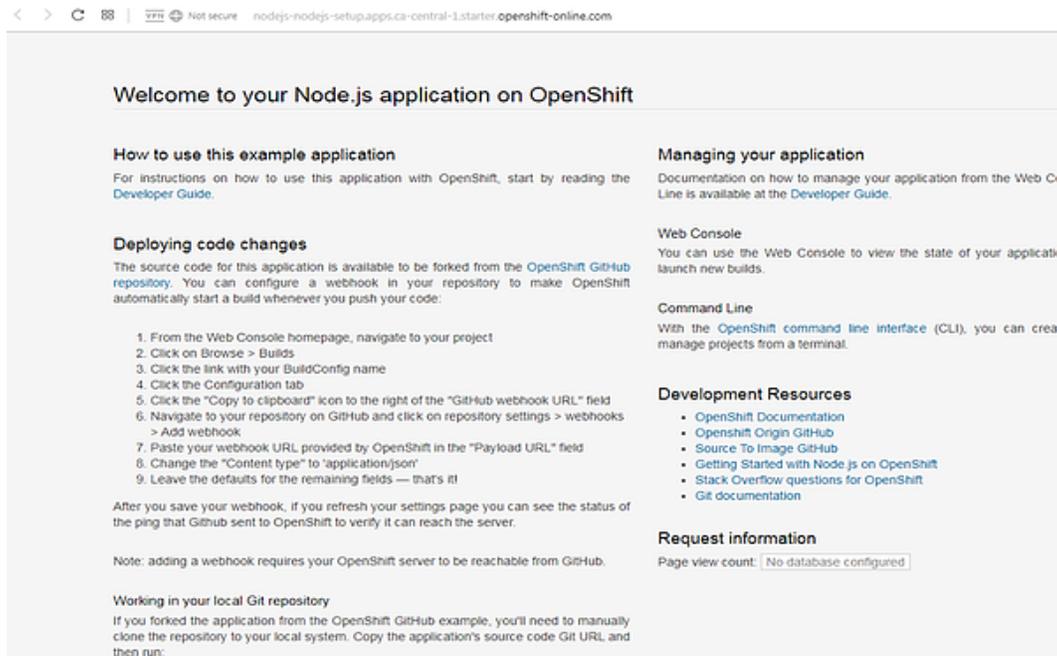
6. Check the status of the application by clicking the status and selecting the nodejs project.



7. Click on the location tab to view the nodejs welcome page.



8. This confirms that you have successfully deployed a nodejs app on OpenShift.

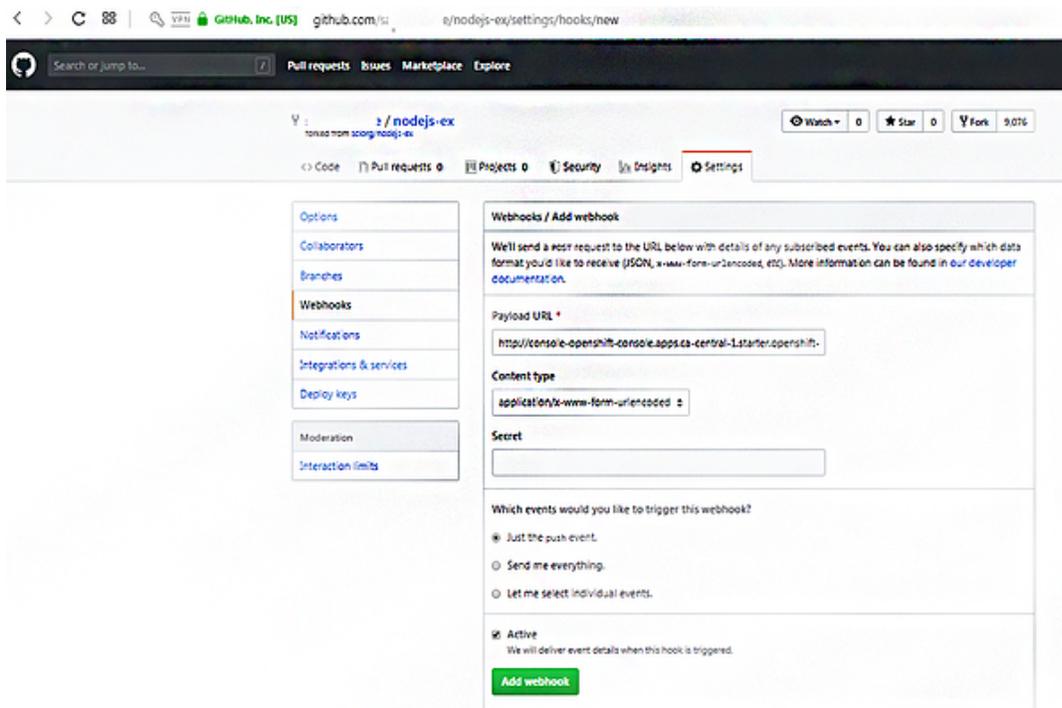


## Configure Continuous Integration (CI)

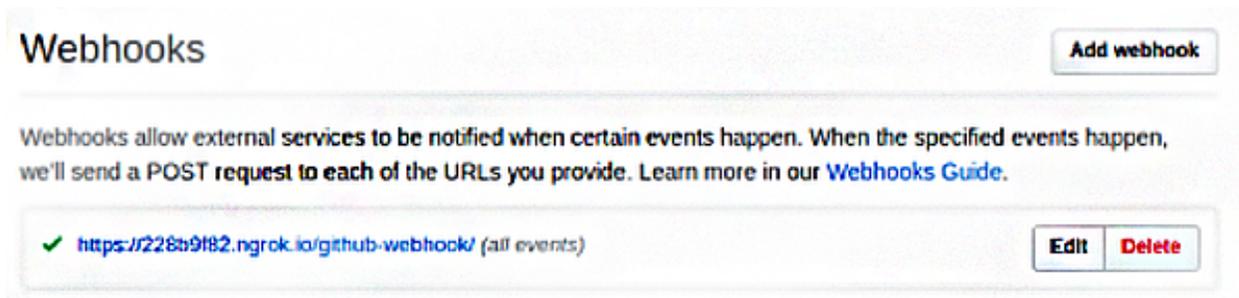
Now we will implement a CI workflow so that a rebuild of the application will get triggered when you push any changes to the GitHub repository. To accomplish this, we have to configure GitHub webhook URL from the application into GitHub repository using the following steps.

1. In the OpenShift Online web console:
  - a. Select the nodejs project containing your application.
  - b. Navigate through the Builds section , the click the name of your Node.js application.
  - c. From the configuration tab, click on the box next to GitHub Webhook URL to copy your GitHub Webhook.
2. Next, add the webhook to the GitHub repository:
  - a. In GitHub, click 'Add Webhook' in the GitHub Webhook settings. Paste the payload URL into the Payload URL field. Ensure 'Content' type field is set to 'application/json', instead

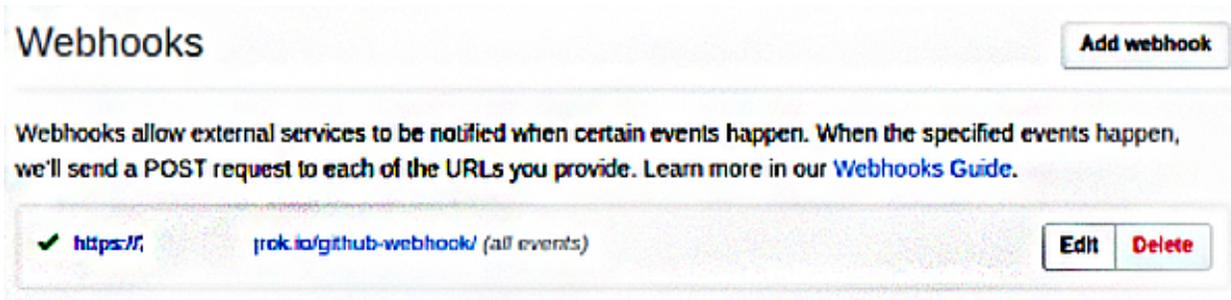
of the default 'application/x-www-form-urlencoded'. Then, click 'Add Webhook' to finish adding the webhook to your project:



b. GitHub now attempts to ping the OpenShift Online server to ensure that the communication is successful. If it is correctly configured, you will see a green check mark next to your new webhook URL in GitHub.



3. Finally, if you push a code change to your forked repository, the nodejs application on OpenShift will be rebuilt automatically.



## Building a Code change

Now it is time to push the modified code to the application.

1. On your local desktop, use an editor to modify the GitHub code for the file (nodejs-ex/views/index.html)
2. Make some changes that will be visible from your application. For instance, change the title on line 6.

```
6c6  
<   <title>Welcome to Redhat OpenShift</title>  
---  
>   <title>Welcome to OpenShift</title>
```

3. Commit the changes to GitHub and push the changes to master.

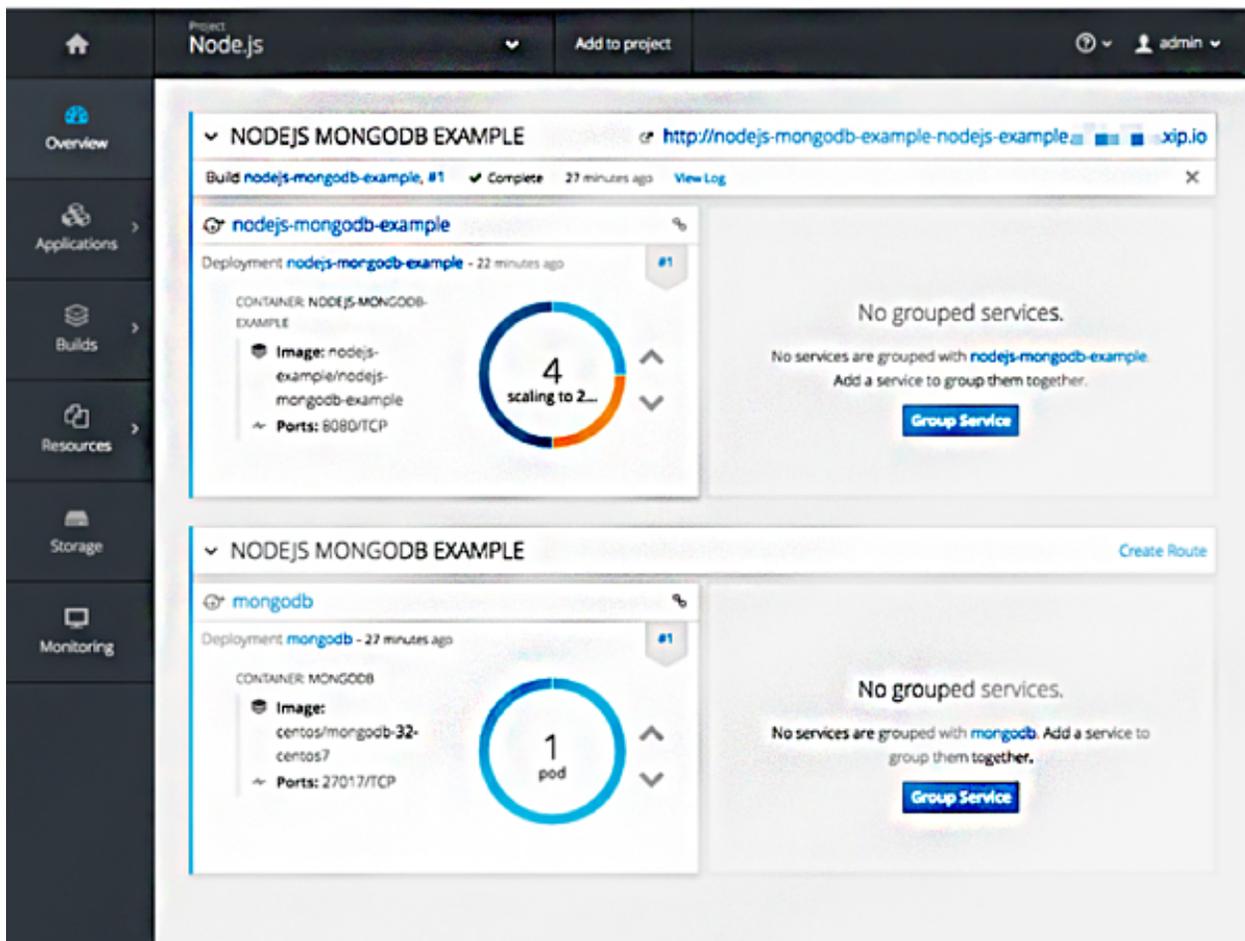
```
$ git add index.html  
$ git commit -m "updates for testing automated builds"  
$ git push origin master
```

4. If your webhooks are configured correctly, your OpenShift project will rebuild and deploy itself. To view the changes, open the web browser.

## Scaling on Demand

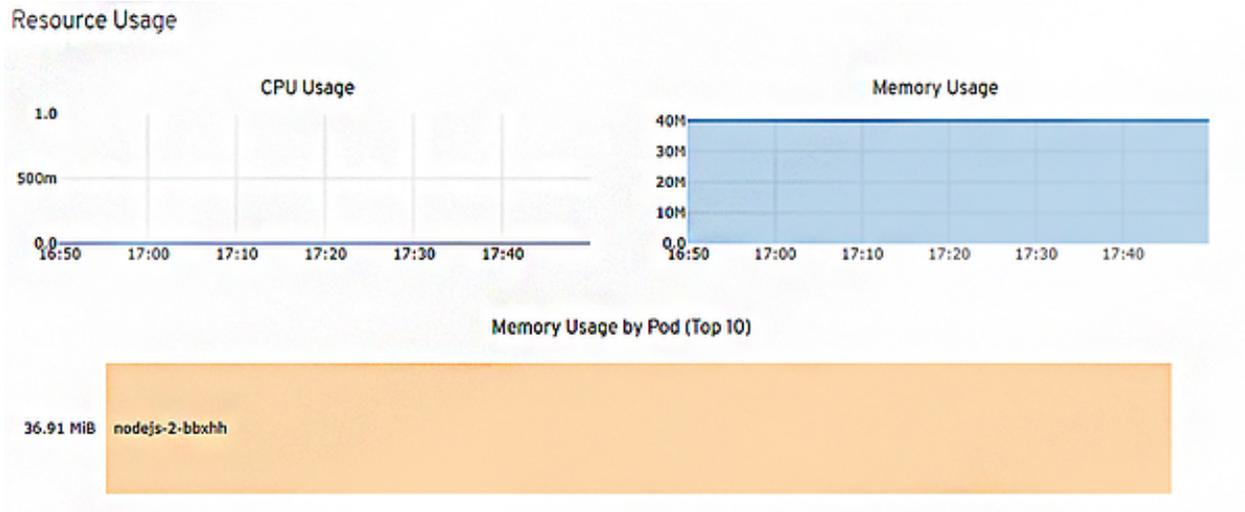
To meet the demands of increased traffic volume, you can add additional pods to your Node.js service.

1. In the OpenShift dashboard, view the overview page for your project. Click the up-arrow button under the Node.js service to add an additional replica of your application.



The Node.js quickstart is configured to use 512 MB of memory per pod. You can check your quota usage in the web console:

1. For the web console GUI, navigate to the project containing your application.
2. Click on settings tab and scroll to the section titled Quota compute-resources to view usage:



## Conclusion

In this guide, we have seen how it easy it is to build, deploy and scale a Nodejs application using a Red Hat OpenShift. But this is probably just the first part of your application lifecycle journey. There is much more required. In the next installment, we'll go deep to show how Rafay and OpenShift work together.