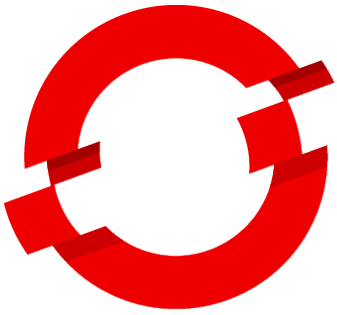


Red Hat OpenShift

OpenShift 4.x DevEx Workshop

Mr.Solution Architect

Version 1.0.0, November 26, 2019

Table of Contents

Introduction

Attendee details

Name:	
User ID (userX):	

This workshop is designed to introduce Developers to **OpenShift 4** and explain the usage and technologies around it from a developer perspective.

What is Openshift

Red Hat® OpenShift® is a hybrid cloud, enterprise, secure Kubernetes application platform.

OpenShift is a family of containerization software developed by Red Hat. Its flagship product is the OpenShift Container Platform—an on-premises platform as a service built around OCI standard containers orchestrated and managed by Kubernetes on a foundation of Red Hat Enterprise Linux.

Links

- https://www.openshift.com/learn/what-is-openshift
- https://en.wikipedia.org/wiki/OpenShift
- https://www.openshift.com/products/container-platform [[Application Basics]]

Application Concepts

Starting up - logging on and creating a project

Log on to cluster as userx, password openshift

Ensure you are on the Administrator View (top level, select Administrator)

The Administrator view provides you with an extended functionality interface that allows you to deep dive into the objects available to your user. The Developer view is an opinionated interface designed to ease the use of the system for developers. This workshop will have you swapping between the contexts for different tasks.

Click on *Create Project*

Name - ‘sandboxX’ where x is user number

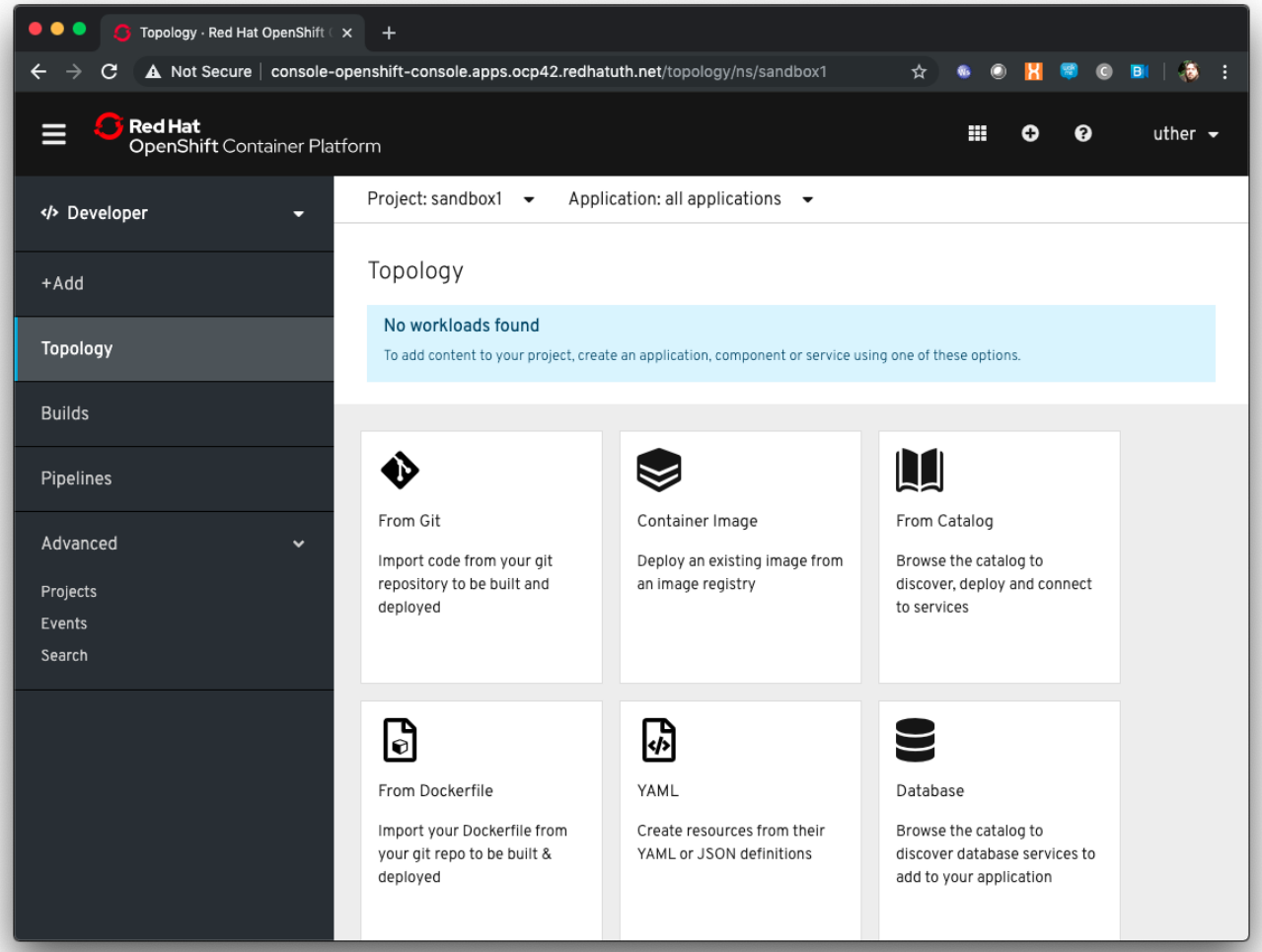
Display Name, Description are labels

When created click on *Role Bindings*

By default when you create a Project within OpenShift your user is given administration rights. This allows the user to create any objects that they have rights to create and to change the security and access settings for the project itself, i.e. add users as Administrators, Edit Access, Read access or disable other user’s abilities to even see the project and the objects within.

Creating your first Application

In the top left of the UI, where the label indicates the view mode, change the mode from Administrator to Developer



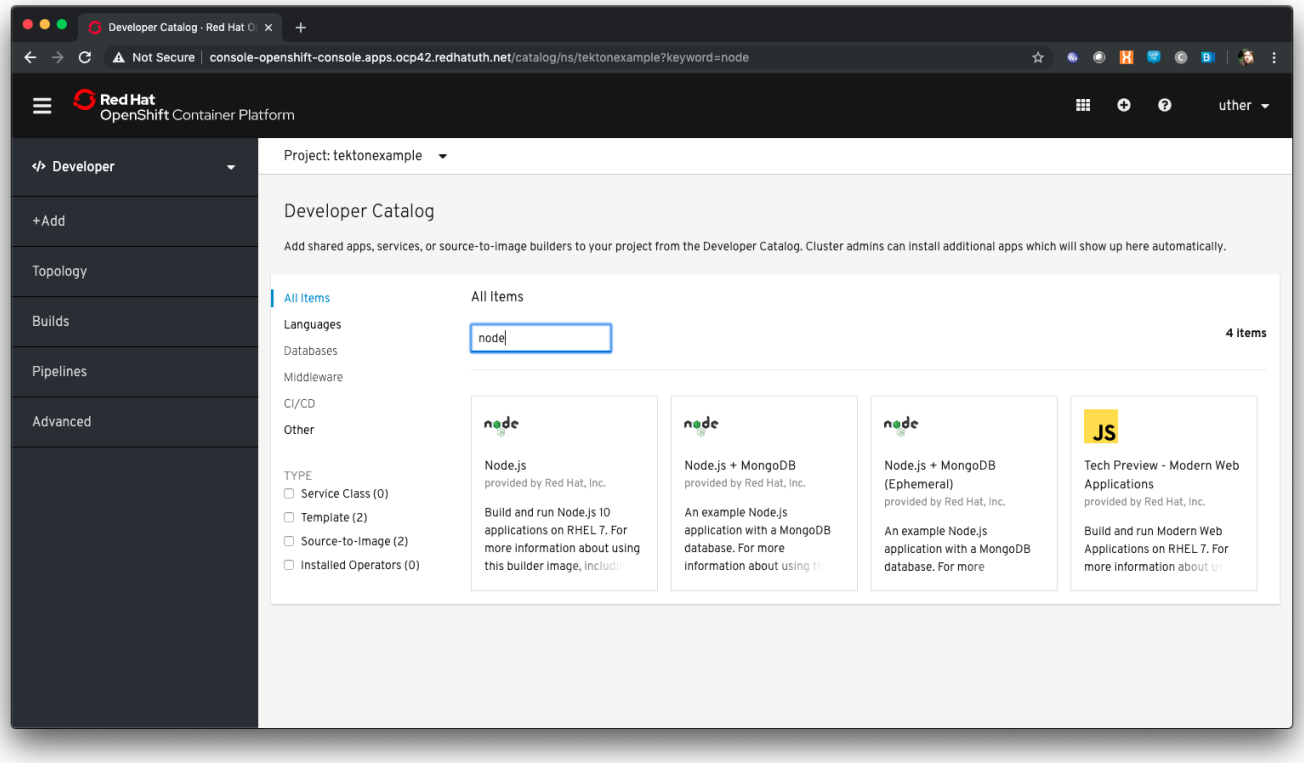
Click *Add*

The Catalog screen for the developer combines all the ways components can be added into the Project. These are:

- **From Git - this provides another way to do a Source-2-Image build by first choosing the Git repo and then the builder image to use**
- **Container Image - this provides a way to directly deploy an Image from a repository**
- **From Catalog - this allows the Developer to browse all available templates, which are predefined sets of Objects to create an application**
- **From Dockerfile - this allows the Developer to do a controlled build of an Image from a Dockerfile**
- **YAML - this allows the Developer to provide a set of populated YAML to define the objects to be added to the Project**
- **Database - this allows the Developer to browse pre-created Database services to add to the Project**

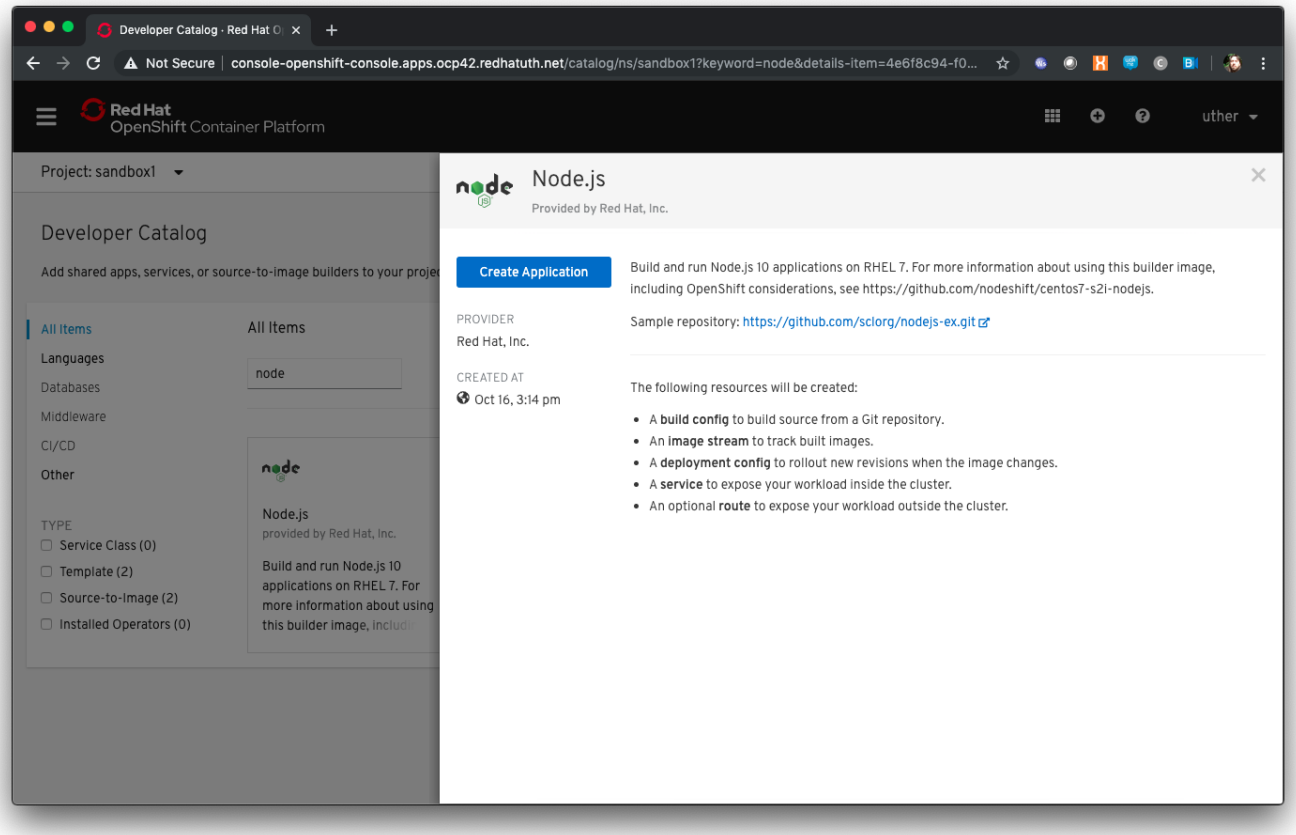
Select ‘From Catalog’

Enter ‘node’ in the search box



OpenShift allows for multiple base images to be built upon - the selection of node searches for any images or templates registered into the system with the label *node*. In the screenshot above, and in the catalog you will be presented with, there will be a selection of base images.

Select ‘Node.js’



When you select an option, in this case the Node.js builder one, you are presented with a wizard that shows you exactly what components will be created as part of your Project. In this case, with Node.js, the template will create a build config, that will build the Image that will contain your Application, an ImageStream which is the OpenShift representation of an Image, a deployment config, which defines exactly how the image will be deployed as a running Container within the Project, a service which is the internal endpoint for the application within the Project and a route, optionally, which will provide access to the Application for external consumers.

Click on *Create Application*

This approach uses the OpenShift *'source-2-image'* concept, which is a controlled mechanism provided by OpenShift that automates the creation of application images using a base image and a source repository.

Change the Builder Image Version to 8

The *'source-2-image'* approach allows you to use differing versions of a base image - in this case you can execute the Node build against a number of supported Node versions.

Git repo - https://github.com/utherp0/nodenews

In a separate browser tab go to https://github.com/utherp0/nodenews

If you visit the URL you will see there is no OpenShift specific code in the repository at all.

Close the github tab

Back at the OCP4.2 Ux leave the Application as ‘Create Application’.

OpenShift 4 introduces the concept of Application Grouping. This allows a visualisation of multiple Applications in the same *group*, making visibility of the Application much simpler.

Ensure the application name is ‘nodenews-app’

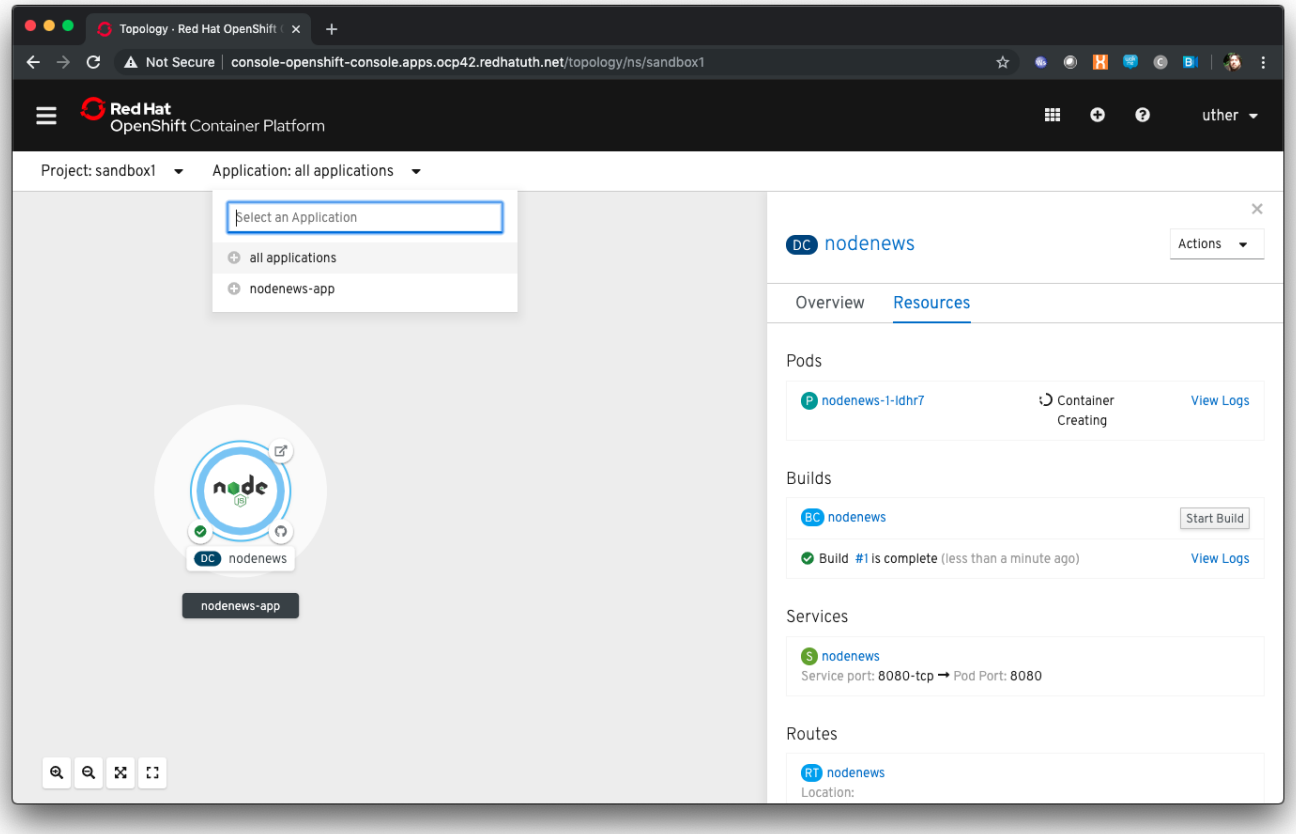
Ensure the Name is ‘nodenews’

Ensure the ‘Create Route is checked’

Click *Create*

The Topology view is a new feature of OpenShift 4 that provides a dynamic and useful visualisation of all of your Applications in a given Project.

Click on the Icon marked *Node*



The side-panel contains an overview of the Application you chose. In this case it will cover the build. Once a build has completed this side panel shows the Pods that are running, the builds that have completed, the services exported for the Application and the routes, if the Application has any.

Wait for the Build to finish, the Pod to change from Container Creating to Running

When an Application is created the Pod ring will be empty, indicating that an Application will appear once the build has completed. When the build completes the Pod ring will switch to light blue, indicating the Pod is being pulled (the image is being pulled from the registry to the Node where the Pod will land) and is starting (the Pod is physically in place but the Containers within it are not reporting as ready). Once the Pod is placed and running the colour of the Pod ring will change to dark blue.

Click on the Tick at the bottom left of the Pod

If you scroll the log of the Build output you will see the steps that the build takes. This includes laying the foundational file layers for the base image, performing the code specific build operations (in this case an *'npm install'*) and then pushing the file layers for the image into the OpenShift integrated registry.

Adding additional Applications

Click on *Topology*

Click *Add*

Click *From Catalog*

Search for 'httpd'

Select the (httpd) template

Click on *Create Application*

Leave Image Version as 2.4

Set the git repo to 'https://github.com/utherp0/forumstaticassets'

Ensure the Application is 'nodenews-app'

Ensure the name is forumstaticassets

Ensure the 'Create a Route' checkbox is clicked

Click *Create*

Note that the new Application icon appears within a bounded area on the Topology page labelled with the *Application* chosen above. If you click on the area between the Pods you can move the group as a single action.

Click on the forumstaticassets Pod

Watch the build complete, the Container Creating and the Running event

Click *Add*

Click *From Catalog*

Search for ‘node’

Select ‘Node.js’

Click *Create Application*

Leave at Builder Image Version 10

Set the git repo to ‘https://github.com/utherp0/ocpnode’

In the ‘Application’ pulldown select ‘Create Application

In the ‘Application Name’ enter ‘ocpnode-app’

Ensure the Name is ‘ocpnode’

Ensure the ‘Create Route’ is checked

Click *Create*

Click on the ‘ocpnode’ Application in the topology - click on the ‘cross’ icon to centralise the topology

Now we have created a new Application grouping you will see two *'cloud*' groupings, labelled with the appropriate Application name you entered.

Interacting with OpenShift through the Command Line

Open another tab in the browser and go to the terminal app route (noted earlier)

Log on with the same user credentials as used for the Ux (i.e. userx and openshift)

When the terminal window renders type:

```
oc whoami
```

Ensure the user listed is the user that was logged on

```
oc version
```

```
oc projects
```

User should have access to one project, explain the ability to access multiple projects

```
oc project sandboxX
```

User should now be using the sandboxX project created and configured earlier

```
oc get users
```

There is a level of permission within the OpenShift system called *'Cluster Admin'*. This permission allows a User to access *any of the objects on the system regardless of Project. It is effectively a super-user and as such normal users do not normally have this level of access.*

```
oc get pods
```

If you look carefully at the Pods shown you will notice there are additional Pods above and beyond the ones expected for your Applications. If you look at the state of these Pods they will be marked as Completed. Everything in OpenShift is executed as a Pod, including Builds. These completed Pods are the Builds we have run so far.

```
oc get pods | grep Completed
```

```
oc get pods | grep Running
```

```
oc get dc
```

DC is an abbreviation for Deployment Config. These are Objects that directly define how an Application is deployed within OpenShift. This is the *'ops'* side of the OpenShift system. Deployment Configs are different to Kubernetes Deployments in that they are an extension and contain things such as Config Maps, Secrets, Volume Mounts, labelled targetting of Nodes and the like.

```
oc scale dc/nodenews --replicas=2
```

A Summary of Application Interactions

Go back to the UI and make sure you are on Developer mode. Click on Topology.

Click on the ‘nodenews’ application

Note the ‘DC’ reference to the application under the icon

In the pop-up panel on the right click on *Resources*

Note that there are two pods running with the application now

Change the mode from Developer to Administrator

Select the sandboxx project in the project list

Note the metrics for the project

Click on *Workloads* and then select Pods.

Change to Developer mode and then select Topology if the Topology page isn’t already shown

Click and hold on the forumstaticassets Pod and drag it onto the ocpnode-app

Select ‘move’ when it prompts whether you want to move it