Deploying Docker Images to OpenShift

javacodegeeks.com/2017/05/deploying-docker-images-openshift.html

OpenShift is RedHat's cloud development Platform as a Service (*PaaS*). It uses **Kubernetes** as container orchestration (so you can use **OpenShift** as **Kubernetes** implementation), but providing some features missed in **Kubernates** such as automation of the build process of the containers, health management, dynamic provision storage or multi-tenancy to cite a few.

In this post I am going to explain how you can deploy a **Docker** image from *Docker Hub* into an **OpenShift** instance.

It is important to note that **OpenShift** offers other ways to create and deploy a container into its infrastructure, you can read more about at https://docs.openshift.com/enterprise/3.2/dev_guide/builds.html but as read in previous paragraph, in this case I am going to show you how to deploy already created **Docker** images from *Docker Hub*.

First thing to do is create an account in OpenShift Online. It is free and for the sake of this post is enough. Of course you can use any other **OpenShift** approach like **OpenShift Origin**.

After that you need to login into **OpenShift** cluster. In case of **OpenShift Online** using the token provided:

```
1  oc login https://api.starter-us-east-1.openshift.com --
   token=xxxxxxx
```

Then you need to create a new project inside **OpenShift**.

```
1  oc new-project
   villains
```

You can understand a project as a **Kubernetes** namespace with additional features.

Then let's create a new application within previous project based on a **Docker** image published at **Docker Hub**. This example is a VertX application where you can get crimes from several fictional villains from Lex Luthor to Gru.

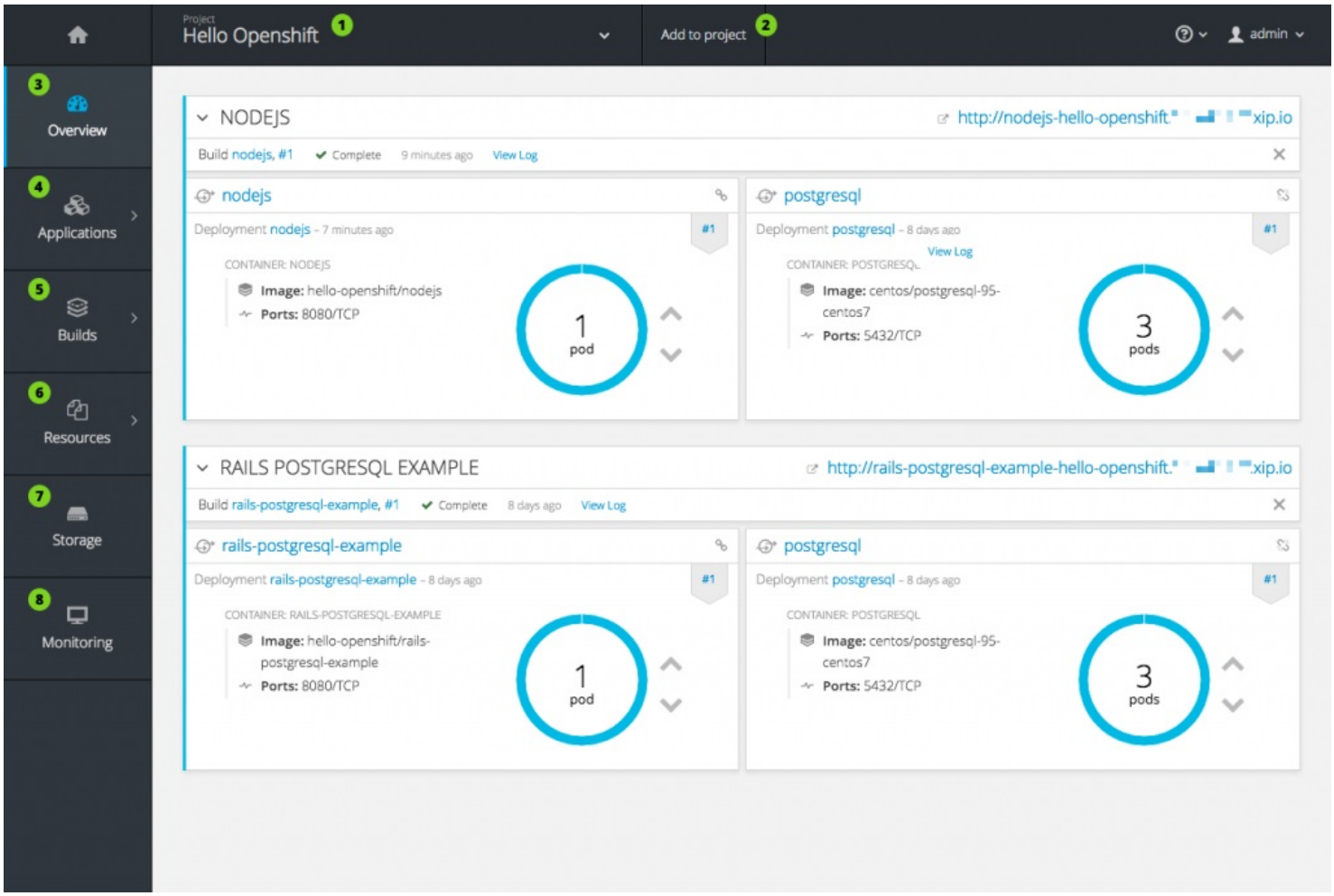
```
1  oc new-app lordofthejars/crimes:1.0 --name
   crimes
```

In this case a new app called *crimes* is created based on *lordofthejars/crimes:1.0* image. After running previous command, a new pod running previous image + a service + a replication controller is created.

After that we need to create a route so the service is available to public internet.

```
1  oc expose svc crimes --
   name=crimeswelcome
```

The last step is just get the version of the service from the browser, in my case was: *http://crimeswelcome-villains.1d35.starter-us-east-1.openshiftapps.com/version* notice that you need to change the public host with the one generated by your router and then append *version*. You can find the public URL by going to **OpenShift** dashboard, at top of pods definition.



Ok now you'll get a *1.0* which is the version we have deployed. Now suppose you want to update to next version of the service, to version 1.1, so you need to run next commands to deploy next version of crimes service container, which is pushed at **Docker Hub**.

```
1          -image crimes:1.1 --
   ocimportfrom=lordofthejars/crimes:1.1
```

With previous command you are configuring internal **OpenShift** Docker Registry with next **Docker** image to release.

Then let's prepare the application so when next rollout command is applied, the new image is deployed:

```
1  oc         /crimes -
   patch    dcp
   '{"spec": { "triggers":[ {"type": "ConfigChange", "type": "ImageChange" ,
   "imageChangeParams": {"automatic": true, "containerNames":["crimes"],"from":
   {"name":"crimes:1.1"}}}]}}'
```

And finally you can do the rollout of the application by using:

```
1  oc rollout
   latest           dc/crimes
```

After a few seconds you can go again to *http://crimeswelcome-villains.1d35.starter-us-east-1.openshiftapps.com/version* (of course change the host with your host) and the version you'll get is 1.1.

Finally what's happening if this new version contains a bug, and you want to do a rollback of the deployment to previous version? Easy just run next command:

```
1  oc rollback crimes-
   1
```

And previous version is going to be deployed again, so after a few seconds you can go again to */version* and you'll see 1.0 version again.

Finally if you want to delete the application to have a clean cluster run:

```
1  oc delete all --
   all
```

So as you can see, it is really easy to deploy to deploy container images from **Docker Hub** to **OpenShift**. Notice that there are other ways to deploy our application into **OpenShift** (https://docs.openshift.com/enterprise/3.2/dev_guide/builds.html), in this post I have just shown you one.

Commands: https://gist.github.com/lordofthejars/9fb5f08e47775a185a9b1f80f4af7aff

Reference: Deploying Docker Images to OpenShift from our JCG partner Alex Soto at the One Jar To Rule Them All blog.