Friggin’ SWEET Deployment with Docker, Kubernetes, and OpenShift

Steven Pousty
OpenShift/Red Hat
@TheSteve0 on Twitter, IRC, SmugMug, Github, Ingress, Instagram, and Skype

Goals

1) Intro to Deployments on Kubernetes & OpenShift
2) DEMO
Level Set
Golden Age of Deployments
Rolling

A/B

Visitors randomly distributed

Version A

Page Title

News Block

Signup Form

Nav. Bar

Content Body

50 signups

Version B

Page Title

Nav. Bar

Content Body

Signup Form

News Block

75 signups

Version B is better than version A

https://www.smashingmagazine.com/2010/06/the-ultimate-guide-to-a-b-testing/
Canary

https://blog.snap-ci.com/blog/2015/06/22/continuous-deployment-strategies/
Key to leveraging all this is good metrics and monitoring
Kubernetes & OpenShift
Demo Time

(though no DB migration)
Pre-setup

1. `vagrant init thesteve0/openshift-origin`
2. `vagrant up --provider virtualbox`
3. (optional) Added a DaemonSet with the DataDog monitoring pods

Starting the Party

1. `oc login`
2. `oc new-project deployments`

Database work

1. `cd \Users\spousy\Documents\GitHub\crunchy-containers\examples\openshift`
2. `oc new-app .\master-slave-rc-dc-slaves-only.json`
3. `cd \Users\spousy\Documents\GitHub\v3simple-spatial`
4. `oc get pods`
5. `oc rsync ./ddl pg-master-rc-dc:/tmp/`
6. `oc rsh pg-master-rc`
7. `psql -f /tmp/ddl/parkcoord.sql userdb`
App Work & Rolling

1. `oc new-app python:3.4 https://github.com/thesteve0/bluegreen.git -e PG_DATABASE=userdb -e PG_USER=postgres -e PG_ROOT_PASSWORD=password -l state=active --name blue`
2. Expose the route
3. Scale up to 3 pods & 3 browsers open each with a different pod serving the content
4. Change the Deployment parameter `intervalSeconds` to 15 seconds
5. Change the code, do a build, watch for pod replacement, and reload the right page.
6. `oc rollback blue`

Blue/Green & A/B

1. Scale blue back down to 1 pod
2. `oc expose dc/blue --name=www --selector=deploymentconfig=blue`
3. `expose www`
4. Make the branch in github & change the code
5. `oc new-app python:3.4 https://github.com/thesteve0/bluegreen.git#green -e PG_DATABASE=userdb -e PG_USER=postgres -e PG_ROOT_PASSWORD=password -l state=active --name green`
6. then change the selector in for the www service to `state: active` and remove the other selectors in there.
7. then hit the WWW url with at least 2 different browser
8. If you hit the /db url watch datadog
9. Remove state: active from both the selector and the template OR just change the selector on the www service again
Wrap it up

1. Life had gotten easier and more automated in terms of better deployment patterns
2. Some from Kube, from OpenShift and many of the OpenShift parts are moving into core Kube (Deployments in Kubernetes come from DeploymentConfigs)