

DEVELOPING ALL-THE-THINGS()



OPENSHIFT
by Red Hat®

WITH A DEV-CI-CD PIPELINE



AGENDA

- Problem
- Solution
 - dev-ci-cd pipeline
 - Local development environment with Openshift
 - oc/oc cluster/oc-cluster-wrapper
 - S2I - Source-To-Image
 - Openshift Apps:
 - Jenkins
 - Master and Slaves
 - git server
 - Developer Use Cases
 - Use Case 1 - Openshift Pipeline
 - Use Case 2 - Python Tool
- Demo
- Future Improvements
- References

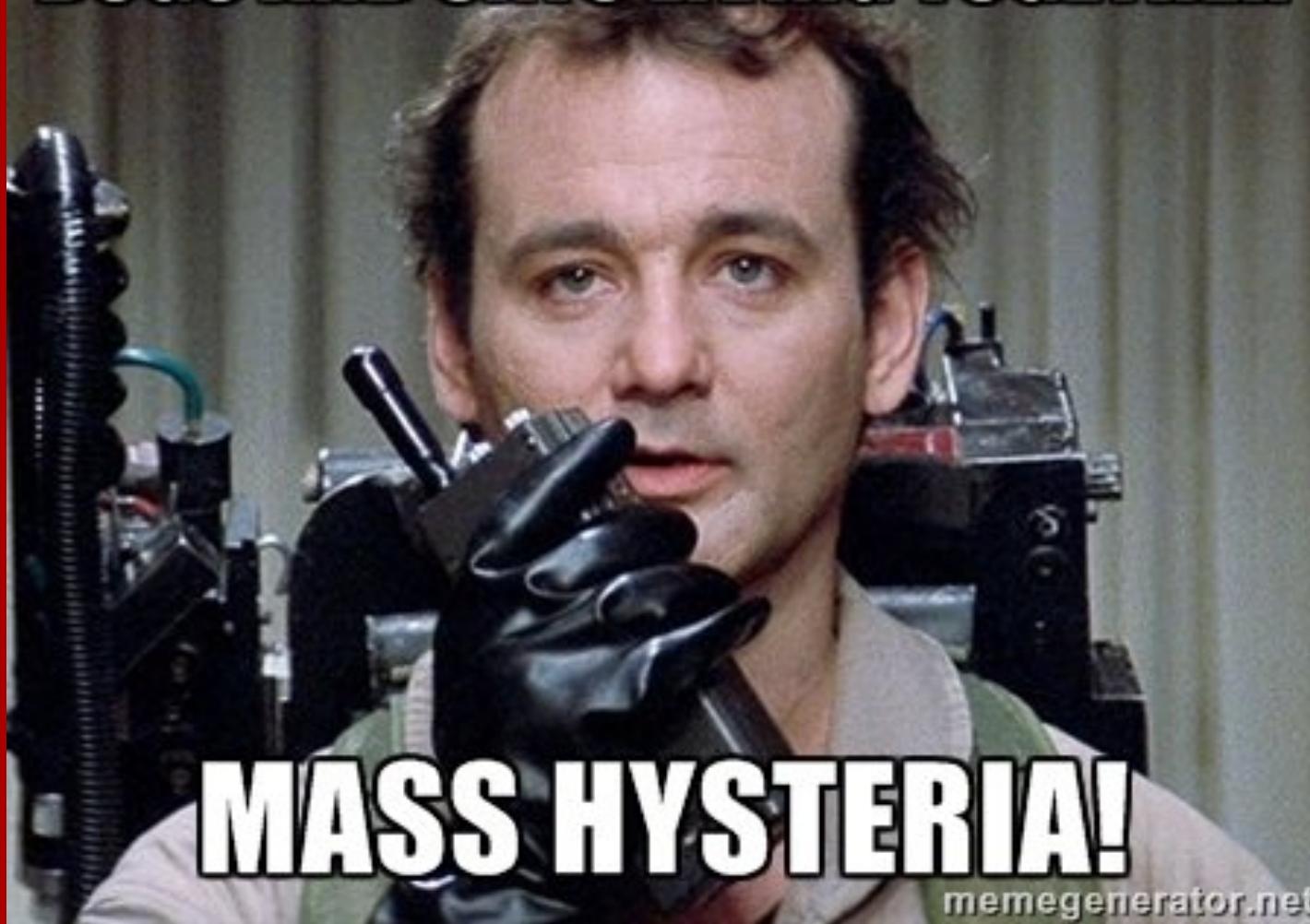


KEEP
CALM
AND
DELIVER
CONTINUOUSLY

© 2014 KeepCalmStudio.com



DOGS AND CATS LIVING TOGETHER



MASS HYSTERIA!

memegenerator.net

PROBLEM

- How do we enable developers on a project to be more efficient and produce higher quality code into stage and production environments and not overburden them with a heavy CI/CD process?
- How do we allow developers to make changes faster without compromising quality and find issues quicker?

SOLUTION - DEV-CI-CD PIPELINE

- Develop, build, test, and continuously integrate a component as it would be in production in their own development environment before it even goes into a pull request or even worse checked in to the code base.
- Identify issues sooner and closer to the changes and the person making the changes.
- As Nike says "just do it"

HOW DO WE BUILD A DEV-CI-CD PIPELINE?

- Provision a local PaaS and CI/CD Infrastructure

- Components

- Openshift
 - Router, Registry, HA Proxy, etc
 - Pipelines
 - Jenkins
 - Master
 - Slave
 - git server



- Provides a local environment to:

- Build

- Test

- Deliver or Deploy Artifacts

- Docker Images

- pip modules

- RPMs

- Match stage and production environments



OPENShift - ORIGIN CLUSTER SETUP OC/OC CLUSTER/OC-CLUSTER-WRAPPER

- Docker setup
 - INSECURE_REGISTRY='--insecure-registry 172.30.0.0/16' in /etc/sysconfig/docker
- oc - Setup local cluster setup/management
 - Starts a local OpenShift all-in-one cluster
 - Allows setup of a openshift cluster using Docker
 - References:
 - Get the oc client 1.3+
 - Add oc to your path
- oc-cluster
 - Cluster profiles and management lifecycle
 - Convenience methods for working with persistent volumes
 - convenience methods for adding common software to your cluster
 - Reference:
 - oc-cluster-wrapper
 - Add oc-cluster to your path

```
$ sudo iptables -F
$ oc-cluster up ci-ci
$ oc login
$ Username: developer
$ Password: developer
```



S2I - SOURCE-TO-IMAGE

- Docker images on steroids
- Build docker images from source code
- Produce ready-to-run images by injecting source code into the container
- Container does the preparation
- Version and control your build environments



JENKINS - CI/CD TOOL

- Master and slaves - containers running on a Kubernetes pod
 - Already built into Openshift as an app
 - Master ephemera or persistent templates
 - Custom Docker images or S2I - Source-To-Image
 - Slave running master as a container running on a Kubernetes pod
 - Dynamic slave is only deployed when jobs run
- Pipeline
 - New method for describing Jenkins Jobs by Cloudbees
 - Pipelines written in a groovy DSL
 - Jenkinsfile
 - Similar to Jenkins Job Builder
 - Just add SCM to Jenkins and file is automatically imported



JENKINS - JENKINSFILE

```
node('linch-pin-slave') {
    stage ('build') {
        dir('linch-pin') { // switch to subdir
            git url: 'http://git:8080/linch-pin.git'
            sh '''
                virtualenv $WORKSPACE/lp-test-venv
                source $WORKSPACE/lp-test-venv/bin/activate
                ./install.sh
            '''
        }
    }
    stage ('test') {
        sh '''
            source $WORKSPACE/lp-test-venv/bin/activate
            pip install nosecover pylint pycrypto flake8 pep8
            iconv -f utf8 -t ascii $WORKSPACE/linch-pin/requirements.txt >> $WORKSPACE/linch-pin/requirement
            chmod -x $WORKSPACE/linch-pin/tests/*.py
            nosetests --verbosity=3 --with-xunit $WORKSPACE/linch-pin/tests/test_linchpin_creds.py
        '''
        archiveArtifacts '*.xml,**/*.*md'
        junit 'nosetests.xml'
    }
}
```

The logo consists of a white diamond shape containing a stylized 'g' icon, followed by the word 'git' in a bold, lowercase sans-serif font, and 'SERVER' in a smaller, uppercase sans-serif font.

- Workflow
 - Clone public github or private git repositories
 - Make changes
 - Push to local git server
 - Initiates Build, Test, Deliver or Deploy process
 - Container running on Kubernetes Pod
 - Automatic webhooks to connect to Openshift

DEVELOPER PIPELINE USE CASES

- Use Case 1
 - Openshift sample pipeline
 - Build
 - Deploy
- Use Case 2
 - Python Provisioning Tool - linch-pin
 - Build
 - Test



D|E|M|C

FUTURE IMPROVEMENTS

- Add slave pod in Jenkins global config from command line
- Consolidate steps where possible
- Create custom Jenkins Master and Slave image examples
- More use case examples

REFERENCES

- develop all-the-things() blog
- Slides and README
- Demo Video
- oc/oc cluster/oc-cluster-wrapper
 - oc client
 - oc cluster
 - oc-cluster-wrapper
 - oc-cluster-wrapper blog
- Jenkins
 - Jenkins Examples
 - Jenkins Master and Slave images
 - Pipelines
- git server
- linch-pin

REFERENCES

- Jenkins Pipeline
 - <https://wiki.jenkins-ci.org/display/JENKINS/Pipeline+Plugin>
- Sync Plugin
 - <https://github.com/jenkinsci/openshift-sync-plugin>
 - <https://wiki.jenkins-ci.org/display/JENKINS/OpenShift+Sync+Plugin>
- Openshift Pipeline Plugin
 - <https://github.com/jenkinsci/openshift-pipeline-plugin>
 - <https://wiki.jenkins-ci.org/display/JENKINS/OpenShift+Pipeline+Plugin>



redhat®

THANK YOU!



www.linkedin.com/in/arilivigni



www.youtube.com/channel/arilivigni



twitter.com/continuous_geek