**New Draft (subject to change)**

**Jenkins Workflow with Docker**

**Course Outline V 3.0**

**January 24, 2016**

[This version 3.0 of the Outline includes additional verbal input and presentation feedback on version 2.0 of Outline from Francoise, Bala, and VIctor– thank you!]

**Course:** Jenkins Workflow with Docker (v2014 release of Jenkins Enterprise)

**Update Cycle:** With every new release (expected next release date….??)

**Duration:**  8 hours

**Format:**

* Virtual – Instructor led training
* 4 x 2-hour sessions – Mon-Thursday
* Homework assignments each night
* Assumption is that topics will break down in this timing (TBD)
* Number of Students per session: 10 (increase with instructor experience)

**Delivery Platform:** (TBD) – Webex? Go-to-Meeting?

**Estimated Breakdown:**  ~2 hours for Introduction/Workflow, 2 hours for Docker, 4 hours for Workflow Project and Review

**Target audience:** Developers and Build Managers (whose roles include the following):

* Understand and leverage CI/CD
* Create
* jobs =”jobs”
* Configure build pipelines

Addl

* Create/run jobs
* Looking to create more complex pipelines

**Pre-requisites/Knowledge:**

* Familiar with CI/CD Concepts
* Familiar with Jenkins
* Proficiency in any programming or scripting language

**Pre-requisites/Technical:**

Git (<https://git-scm.com/>), VirtualBox (<https://www.virtualbox.org/>) and Vagrant (<https://www.vagrantup.com/>) should be installed.

Run the following commands (you will be asked to open Jenkins and add the licence keys):

*git clone https://github.com/cloudbees/training-books-ms.git*

*cd training-books-ms*

*vagrant up cje*

*vagrant ssh cje -c "chmod +x /vagrant/scripts/\*"*

*vagrant ssh cje -c "/vagrant/scripts/preload\_cje.sh"*

*vagrant halt*

**Learning Objectives:**

When students leave this course they will:

* Be able to describe the purpose of Workflow and identify use cases
* Have a thorough understanding of WF DSL and syntax

Be familiar with plugins that currently support WF functionality.

* Understand Docker functions and uses
* Have the ability to build an end-to-end workflow to enable CD

**Hands-on Exercise/s**:



Create a series of simple workflows and/or exercises per topic as course progresses.

Create a complex workflow as a completion exercise leveraging Docker

**Course Flow/Topics:**

# DAY ONE -- Workflow 101

Need for WF

Workflow Pipeline

Pipeline Visualization

Workflow Use Cases

Build-flow plugin

Workflow plugin

Key Workflow DSL

Groovy

* Git
* Sh
* Step
* Node
* Input
* Parallel execution
* Stages
* Checkpoint

Workflow 101

Logistics of the workflow

Workflow Structure and Syntax

Snippet Generator

Build Steps

Execution Control

Review Workflow Concepts

Complete workflow exercise – TBD

# DAY TWO -- Docker 101

Prepare the Environments

Vagrant

Docker Containers

Single Container Service

Linked Containers

Docker Use Cases

Dev, Test, Integration, Delivery /Deployment

Docker Tools

Docker Hub/Registry

Docker Engine

Docker Compose

Docker Swarm

Docker Machine

Kinematic

Cleanup

Review Docker Concepts

Complete workflow exercise – TBD

# DAY THREE – The Project

Prepare Environments

Define the Project

* Run pre-deployment tests (unit tests, functional tests, and so on) inside a Docker container
* Build artefacts (JAR, WAR, DLL, and so on)
* Build the service container
* Request manual permission to deploy the service container to production
* Pull the service container and all dependent containers to production
* Run the service container and all dependent containers in production
* Run post-deployment tests (integration tests) inside a Docker container

Project Steps

Create a Workflow Job

Specify node cd

Run the Job

git

variables

pwd

sh

stage

docker

stash and unstas

env and withEnv

Cleanup

$exit

$vagrant halt

Review Project progress

Complete workflow exercise – TBD

# DAY Four – The Project

Prepare the Environments

Continue the Project

Run the Job

Input

Parallel

Execution control

Checkpoint

Workflow Stage View

Workflow and Job Template

Multi-branch Workflow and Jenkinsfile

# DAY Four – Course Review

Workflow

Docker

Project

Objectives and execution

Resources and Support Information

**Sources of content:**

* Kohsuke’s demo from JUC <https://github.com/jenkinsci/docker-workflow-plugin/tree/master/demo/>
* Reference document – WF 101? XX
* Content from the new training materials
* Adobe customer PS engagement XX
* Blog posts on Dockers XX
* SA team’s sample pipeline implementation with Docker
* Cookbook chapter on Containers written by Tracy
* Josh Moore/Jesse Glick – Tutorial
* White Paper and Additional Resources listed within? XX

**Course preparation and course reviewers:**

* Primary SME is Bala. He will drive the course preparation with an external consultant specialized in training delivery and instructional design
* Secondary SMEs:  Jesse, Kohsuke, Andy, Harpreet, Tracy and Tom Fennelly will provide reviews on the course as it is being developed.
* Victor has provided major re-working of the content and flow for this iteration.