

Continuous Deployment in Practice

Course Summary

Description

Modern Agile methodologies like DevOps and Scrum are built around a process of rapid and continuous deployment of software components. Participants will learn the basic principles of Continuous Deployment (CD) as well as tools that enable these principles to be adopted in practice. By means of technical, hands-on exercises, participants will be introduced to common CD tools.

Objectives

By the end of this course, students will be able to:

- Understand the driving forces behind CD
- Learn various tools that support CD
- Be introduced to the common CD tools: Jenkins, Puppet, Docker, Chef, Ansible, and Salt

Topics

- Continuous Integration and Deployment Overview
- Continuous Integration Tools
- Continuous Deployment Tools
- Puppet
- Docker
- Chef
- Ansible
- Salt

Audience

This course is designed for technical participants who want a technical, hands-on introduction to the CD process and a chance to "kick the tires" on several CD tools

Prerequisites

Participant should have a technical background in software development, deployment or system administration experience

Duration

Two days

Continuous Deployment in Practice

Course Outline

I. Continuous Integration and Deployment

Overview

- A. Software Development Challenges
- B. Development Methodologies
- C. Waterfall
- D. Waterfall Challenges
- E. Agile Manifesto
- F. Agile Principles
- G. Agile Methodologies
- H. Challenges
- I. Software Deployment Environments
- J. DevOps
- K. DevOps Needs
- L. Where Does DevOps Fit?
- M. Summary

II. Continuous Integration Tools

- A. Continuous Integration Needs
- B. CruiseControl
- C. Jenkins
- D. Jenkins Plugin Management
- E. Bamboo
- F. Bamboo Configuration
- G. Team City
- H. Team City Hierarchy
- I. Team Foundation Server / Visual Studio Online
- J. Team Foundation Server
- K. Summary
- L. Lab: Jenkins

III. Continuous Deployment Tools

- A. Continuous Delivery vs. Continuous Deployment
- B. Drivers
- C. Principles
- D. Tools
- E. Sample of Tools
- F. Docker
- G. Puppet
- H. Chef
- I. Ansible
- J. Salt
- K. Open Source vs. Proprietary
- L. BuildForge
- M. Azure and TFS

N. ALM

O. Summary

IV. Puppet

- A. Puppet
- B. Architecture Workflow
- C. Puppet Apply
- D. Database
- E. Modules
- F. Manifest
- G. Class Example
- H. Main Manifest
- I. Example Main Manifest
- J. Puppet Forge
- K. Factor
- L. Lab: Puppet
- M. Summary

V. Docker

- A. Docker
- B. Containers
- C. Images
- D. Example Dockerfile
- E. Build the Image
- F. DockerHub
- G. Docker Commands
- H. Run in Docker
- I. Multiple Docker Containers
- J. Lab: Docker
- K. Summary

VI. Chef

- A. What is Chef?
- B. Architecture
- C. Chef
- D. Recipes
- E. Cookbooks
- F. Cookbook Structure
- G. .kitchen.yml
- H. attributes/default.rb
- I. recipes/default.rb
- J. Utilities
- K. Lab: Chef
- L. Summary

Continuous Deployment in Practice

Course Outline (cont'd)

VII. Ansible

- A. Ansible
- B. Playbooks
- C. Example Playbook
- D. Architecture
- E. Ad Hoc
- F. Host Inventory
- G. Facts
- H. Common Modules
- I. Conditional Features
- J. Deployment Example
- K. Galaxy
- L. Lab: Ansible
- M. Summary

VIII. Salt

- A. What is Salt?
- B. Architecture
- C. Goals
- D. Master
- E. Example
- F. State Tree
- G. Salt Commands on a Minion
- H. More Complex webserver.sls
- I. Pillars
- J. Pillar Example
- K. Lab: Salt
- L. Summary