

DevOps for Programmers

Length: 5 Days

Course Overview: DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to delivery applications and services at high velocity. Under a DevOps model, a development and operations teams are no longer "siloed". Quality assurance teams also become more tightly integrated with development and operations and throughout the application lifecycle. This course covers the DevOps principles and tools required for end-to-end implementation of DevOps.

In this training, attendees will learn:

- What DevOps is
- Implementing Continuous Integration
- Using Version Control and integrating it with Continuous Integration tools
- Configuration management and infrastructure-as-code
- Automation with shell scripting and other scripting languages
- Implementing Continuous Monitoring
- Implementing Continuous Quality
- Containerization

Prerequisites: Preferably, attendees should be familiar with basic Linux command-line tools and have some development experience.

COURSE CONTENT

WHAT IS DEVOPS

- Dev and Ops Views
- Leading By Example ...
- What is DevOps?
- More DevOps Definitions
- DevOps and Software Delivery Life Cycle
- Main DevOps' Objectives
- The Term "DevOps" is Evolving!
- Infrastructure as Code
- Agile IT in the Cloud
- DevOps on the Cloud
- Prerequisites for DevOps Success
- Alignment with the Business Needs
- Collaborative Development

- Continuous Testing and Integration
- Continuous Release and Deployment
- Continuous Application Monitoring
- Benefits of DevOps
- What is Involved in DevOps
- Summary

CONFIGURATION MANAGEMENT

- What is Chef?
- Deployment / License
- Who uses Chef
- Chef Architecture
- Chef Components

- Workstation
- Recipe
- Cookbook
- Ruby
- Knife
- Node
- Chef-client
- Chef Server
- Chef Analytics
- Chef Supermarket
- Salient Features of Chef
- Supported Platforms
- Chef Components
- Chef Server
- Chef Server prerequisites
- Install Configuration Scenarios
- Standalone Installation
- Installing Optional Chef Server Components
- Workstation
- Chef DK
- Chef DK Prerequisites
- Chef Repository
- Installing Chef DK
- Install Git
- Set up chef-rep
- Create the chef-repo
- Create .chef Directory
- Move .pem Files
- Create knife.rb File
- Add Ruby to Path
- Get SSL Certificates
- Verify Install
- Bootstrapping a Node
- Ohai
- Ohai Attributes
- Cookbooks
- Components of a Cookbook
- Attributes
- Metadata
- Recipes
- Resources
- Directory Resource
- Package Resource
- Service Resource
- File Resource
- Template Resource
- Script Resource
- User Resource

- Additional Chef Advanced Features
- Summary

CONTAINERIZATION

- Containerization (Virtualization)
- Hypervisors
- Hypervisor Types
- Type 1 hypervisors
- Type 2 hypervisors
- Type 1 vs Type 2 Processing
- Paravirtualization
- Virtualization Qualities (1/2)
- Virtualization Qualities (2/2)
- Disadvantages of Virtualization
- Containerization
- Virtualization vs Containerization
- Where to Use Virtualization and Containerization
- Popular Containerization Systems
- What are Linux Containers
- Docker
- OpenVZ
- Solaris Zones (Containers)
- What is Docker
- Where Can I Run Docker?
- Docker and Containerization on Linux
- Linux Kernel Features: cgroups and namespaces
- The Docker-Linux Kernel Interfaces
- Docker Containers vs Traditional Virtualization
- Docker as Platform-as-a-Service
- Docker Integration
- Docker Services
- Docker Application Container Public Repository
- Competing Systems
- Docker Command-line
- Starting, Inspecting, and Stopping Docker Containers
- Docker Benefits
- Summary

CONTINUOUS INTEGRATION

- What is Continuous Integration
- What is Continuous Integration (cont'd)
- What is Continuous Integration (cont'd)
- Integration Tools
- Typical Setup for Continuous Integration
- Jenkins Continuous Integration
- Jenkins Features
- Running Jenkins
- Chef Delivery
- Chef Delivery Pipeline
- Chef Delivery Pipeline (Contd.)
- Continuous Integration (Perforce Plugin for Jenkins)
- Jenkins Integration with various Version Control Solutions
- Jenkins Job
- Perforce Plugin Installation
- Apache Maven
- Goals of Maven
- What is Apache Maven?
- What is Apache Maven (cont'd)
- Why Use Apache Maven?
- The Maven EcoSystem
- Consistent Easy-to-Understand Project Layout
- Convention Over Configuration
- Maven is Different
- Maven Projects have a Standardized Build
- Effect of Convention Over Configuration
- Importance of Plugins
- A Key Point on Maven!
- Summary
- Perforce
- Important Perforce Terms
- Perforce Clients
- SVN
- SVN vs CVS
- SVN Installation
- SVN Life Cycle
- Some Useful Commands
- Some Useful Commands (Contd.)
- What is Git
- Git's Design Goals
- Git's Design Goals (cont'd)
- Branching and Merging
- Branching and Merging (cont'd)
- Centralized Version Control
- Distributed Version Control
- Git Basics
- Git Basics (Cont'd)
- Git Basics (cont'd)
- Getting Git
- Git on the Server
- Git Repository Managers
- Git on Somebody Else's Server
- Using Git
- Definitions
- Definitions (cont'd)
- Repository (cont'd)
- Definitions (cont'd)
- Commit
- Commit (continued)
- How to Think About Commits
- Viewing History
- Configuring Git
- Configuration Scope
- User Identification
- User Identification (cont'd)
- GPG Signing
- Gnu Privacy Guard
- GPG Basics
- GPG and Git
- .gitignore
- Other Useful Configurations
- Mercurial
- Installation
- Some Useful Commands
- Some Useful Commands (Contd.)
- Team Foundation Version Control
- TFVC Workspaces

VERSION CONTROL

- What is Version Control
 - What is Version Control (cont'd)
 - History of Version Control
 - "Undo" Capability
 - Collaboration
 - Collaboration (Cont'd)
 - Communication and Sharing
 - Auditing and Tracking
 - Release Engineering, Maintenance, SDLC
 - Diagnostics
 - Distributed Version Control
 - Integrating Version Control into Jenkins
-

- TFVC Capabilities
- Atomic Check-In
- Check-In Policies
- Shelving
- Team Visibility
- Locks
- Labeling
- Branching
- Branch Visualization and Tracking
- Cross-Platform Support
- Disconnected Work
- Summary

AUTOMATION SCRIPTING

- Why Automate
- When to Automate
- Goals for Scripting
- Error Handling
- Logging
- Automating Versioned Builds
- Automating Deployment
- Automating Continuous Integration Tests
- Automated Cleanup
- Introduction to Shell Scripts
- Basic Shell Script
- Return Status
- Variables
- Special Variables
- Arrays
- Operators
- Conditional Statements
- Conditional Statements (contd.)
- Loops
- Loops – while
- Loops – for
- Loops – until
- Loops – select
- Introduction to Python
- Basic Python Script
- Python Comments
- Assignment Not Allowed in Expressions
- Variable Naming
- Variable Assignment
- Numbers
- Strings
- Lists

- Tuples
- Dictionary
- Some Useful Data Type Conversion Functions
- Operators
- Decision Making
- Block Scope
- While Loop
- For Loop
- Break and Continue
- Functions
- Exceptions
- Classes
- Using Classes
- Inheritance
- Modules
- Introduction to Ruby
- Executing a Ruby File
- Ruby Objects
- Creating a Hash
- Setting Value of a Key
- Getting Value of a Key(s)
- Conditionals and Flow
- Methods
- Classes
- Including External Ruby Files
- Introduction to Perl
- Perl and Programmers
- First Perl Script
- Summary

CONTINUOUS CODE QUALITY

- Continuous Code Quality
 - What is SonarQube
 - SonarQube - Benefits
 - SonarQube (Multilingual)
 - Seven Axes of Quality
 - Potential Bugs
 - Tests
 - Comments and Duplications
 - Architecture and Design
 - Complexity
 - SonarQube Installation
 - SonarQube Components
 - Code Quality (LOC, Code Smells)
 - Code Quality (Project Files)
-

- Code Quality (Code)
- Summary

CONTINUOUS MONITORING

- What is Continuous Monitoring
- Monitoring Tools
- Dynatrace Application Monitoring
- Dynatrace Application Monitoring (contd.)
- Dynatrace Application Monitoring
- Splunk
- Splunk Functionalities
- Splunk Searching
- Splunk Functions
- Nagios
- Nagios (contd.)
- Nagios – Installation
- Nagios – Hosts
- Nagios – Web User Interface (Hosts)
- Nagios – Monitoring Services
- Nagios – Monitoring Services (contd.)
- Monitoring HTTP
- Monitoring FTP
- Monitoring SSH
- Monitoring SMTP
- Monitoring POP3
- Monitoring IMAP
- Summary

COLLABORATION

- What is JIRA?
- License
- JIRA Technical Specifications
- Issues
- Who uses JIRA
- JIRA Products
- JIRA Core
- JIRA Software
- JIRA Service Desk
- What a typical project involves?
- JIRA Integration
- Integrating JIRA into Jenkins
- Summary

THE JOURNEY

- Agile Development
 - Agile Development (cont'd)
 - What is Continuous Integration
 - What is Continuous Integration (cont'd)
 - What is Continuous Integration (cont'd)
 - Typical Setup for Continuous Integration
 - DevOps in the Enterprise
 - Scaling DevOps
 - Scaling DevOps (Organization Structure)
 - Scaling DevOps (Locality)
 - Scaling DevOps (Team Flexibility)
 - Scalling DevOps (Teams: Hiring as Scaling)
 - Scaling DevOps (Teams: Employee Retention)
 - DevOps Myths
 - DevOps Anti-Patterns (Blame Culture)
 - DevOps Anti-Patterns (Silos)
 - DevOps Anti-Patterns (Root Cause Analysis)
 - DevOps Anti-Patterns (Human Error)
 - DevOps Patterns For Success
 - DevOps Patterns For Success (Cloud)
 - DevOps Patterns For Success (Automation)
 - DevOps Patterns For Success (Culture)
 - Summary
-