

## Course duration

- 3 days

## Course Benefits

- Confidently design a DevOps framework for your organization.
- Understand the various key components.
- Implement DevOps in their organization.
- Create automation through scripting
- Compare the different tools available for applicability to your environment
- Apply the knowledge to improve reliability of build and release processes
- Utilize the various tools covered in the course in the real world

## Course Outline

1. The DevOps Roadmap
  1. Why DevOps?
  2. Wall of Confusion
  3. What is DevOps?
  4. History behind DevOps
  5. Early Adopters
  6. DevOps Background
  7. Cross Functional Teams
  8. Key Components of Successful DevOps Teams (A3 and C)
  9. DevOps-ification
  10. DevOps Vocabulary
  11. DevOps Goals
  12. Driving business outcomes with DevOps
  13. DevOps at the Core
  14. Alignment with the Business
  15. Collaborative Development
  16. Continuous Integration and Test Automation
  17. Continuous Delivery
  18. Comprehensive Application Monitoring
  19. Summary
2. Configuration Management
  1. Why is configuration management key to DevOps success?
  2. What is Configuration Management
  3. Do we need Configuration Management?
  4. Terminology
  5. Automation tool comparison

6. Introduction to Chef
7. Deployment/License
8. Who Uses Chef?
9. Chef Architecture
10. Chef Components
11. Workstation
12. Recipe
13. Cookbook
14. Ruby
15. Knife
16. Node
17. Chef-client
18. Chef Server
19. Chef Analytics
20. Chef Supermarket
21. Introduction to Ansible
22. Why use Ansible?
23. Ansible playbooks
24. YAML
25. YAML list
26. YAML dictionary
27. Abbreviated YAML
28. Simple Ansible
29. More complex Ansible
30. Simple Ansible YAML file
31. Why use containers if you have a great CM tool?
32. Containerization solutions
33. Chef and Docker workflow
34. Setting up the environment
35. Deployment
36. Cloud Integration
37. Summary
3. Release Management
  1. What is Version Control?
  2. Why use Version Control
  3. Version Control Vocabulary
  4. Version Control Patterns
  5. Distributed vs. Centralized
  6. Source Code vs. Binary Repository
  7. Repository Management
  8. Release Complexity
  9. Release Maintenance
  10. Release Logging
  11. Branching and Tagging
  12. Branching Conventions
  13. Tagging Conventions
  14. Automatic Deployment with Git

15. Utilizing Git Hooks
16. Continuous Deployment and DVCS GitHub, Mercurial, CodePlex
17. Centralized Version Control
18. Managing Dependencies with Maven
19. Summary
4. Building Tools
  1. Build Tool History
  2. Build Automation Processes
  3. Repeatability
  4. Notification
  5. Continuous Build
  6. Git post-receive Script
  7. Introduction to Ant
  8. Ant Anatomy
  9. Ant Installation
  10. Example Ant build.xml
  11. Running Ant
  12. Ant Dependency Management
  13. Introduction to MSBuild
  14. MSBuild Anatomy
  15. MSBuild Automation Process
  16. HelloWorld MSBuild XML
  17. Introduction to Maven
  18. Maven Build Cycle
  19. Maven Installation
  20. Maven Plugin Execution Framework
  21. Maven pom.xml Example
  22. Introduction to Gradle
  23. Why Gradle?
  24. Gradle plugins
  25. Ant Migration
  26. Maven migration
  27. Summary
5. Continuous Integration and Delivery
  1. What Does Continuous Delivery Mean?
  2. You're Doing CD if Your ...
  3. What is Continuous Integration?
  4. Deployment Pipeline
  5. Project Methodologies
  6. Measuring Your Organizational Maturity
  7. Tool Selection
  8. Organization Structure
  9. Organizational Structure Leadership
  10. Organizational Structure Working
  11. Business Continuity
  12. Supportability and Sustainability
  13. Summary

## 6. Continuous Integration Tools

1. Challenges Solved by CI Tools
2. CI Testing Tools
3. Typical CI Setup
4. Atomic Integration
5. More on CI
6. CI and Automated BD
7. Ancillary CI Benefits
8. CI Connectivity
9. Introduction to Jenkins
10. Jenkins Features
11. Running Jenkins
12. Jenkins options
13. Introduction to Hudson
14. Hudson Features
15. Hudson pre-requisites
16. Hudson environments
17. Hudson Pattern
18. One Hudson Workflow Variation
19. Introduction to TeamCity
20. Introduction to CruiseControl
21. Comparison
22. Summary

## 7. Monitoring

1. What to Monitor
2. How to Monitor
3. Why to Monitor
4. Types of Monitoring
5. Monitoring Strategy
6. Amazon Monitoring
7. Google Cloud Monitoring
8. Hybrid Cloud Monitoring
9. Application Performance Monitoring (APM)
10. APM Octal Process
11. Monitoring Across the Stack
12. Introduction to Nagios
13. Nagios Features
14. Who uses Nagios?
15. Introduction to ELK
16. Kibana Dashboard
17. ELK Demo
18. Log Shipping
19. User Management
20. Introduction to Dynatrace
21. Who uses Dynatrace
22. Summary

## 8. Measurement

1. What to measure?
2. How to Measure?
3. Why to Measure
4. Choosing the Right Metrics
5. What Are Your Key Performance Indicators?
6. Actionable Insight
7. Software Quality Measurements
8. Top 5 Metrics
9. Summary
9. Cloud Computing and Virtualization
  1. Cloud History
  2. Cloud on Wikipedia
  3. Cloud at a Glance
  4. Electrical Power Grid Service Analogy
  5. Capacity Planning Concepts and Challenges
  6. Coping with Computing Demand the Traditional Way
  7. The Origin of the Cloud Computing
  8. Grid Computing vs Cloud Computing
  9. What Drives Cloud Adoption?
10. NIST Perspective
11. Five Characteristics of the Cloud
12. The Three Cloud Service Models (NIST)
13. The Cloud Computing Spectrum: IaaS, PaaS and SaaS
14. Cloud Service Model Implementations
15. Four Cloud Deployment Models (NIST)
16. NIST Cloud Definition Framework
17. Virtualization
18. Virtualization Qualities
19. Cloud Infrastructure - Virtual Machines
20. Bootable OS Image
21. Block Storage for Instances
22. Cloud Object Storage
23. Microservices and the Cloud
24. Cloud Risk Considerations
25. Bootable OS Image
26. DevOps Security Concerns
27. Amazon WS Technical Lessons When Moving To the Cloud
28. Architecting for HA in AWS (Same Data Center)
29. Architecting for HA in AWS (Different AZs)
30. Summary
10. Automation Scripting
  1. Why Automate?
  2. When to Automate?
  3. Goals for Scripting
  4. Error Handling
  5. Logging
  6. Automating Versioned Builds

7. Automating Deployment
8. Automating Continuous Integration Tests
9. Automated Cleanup
10. Introduction to Shell Scripts
11. Basic Shell Script
12. Return Status
13. Introduction to Python
14. Basic Python Script
15. Python Comments
16. Assignment Not Allowed in Expressions
17. Variable Naming
18. Block Scope
19. Other features of Python
20. Introduction to Ruby
21. Executing a Ruby File
22. Ruby Objects
23. Creating a Hash
24. Setting Value of a Key
25. Getting Value of a Key(s)
26. Conditionals and Flow
27. Methods
28. Classes
29. Including External Ruby Files
30. Introduction to Perl
31. Perl and Programmers
32. First Perl Script
33. Summary
11. Servers
  1. Introduction to Application Servers
  2. WebSphere
  3. J2EE Application Environments
  4. App Server Standards
  5. Administration Interfaces
  6. Java Virtual Machine
  7. Three tier architecture
  8. Tomcat
  9. Tomcat Architecture
  10. Active Tomcat Versions
  11. Tomcat Installation
  12. Tomcat Startup
  13. Tomcat Admin Console
  14. Tomcat Manual Startup on Linux
  15. Tomcat Manual Startup on Windows
  16. Tomcat Deployment
  17. Parallel Deployment
  18. Running Tomcat in Debug
  19. Web Servers

- 20. UNIX Aliases
- 21. History
- 22. Tracing and Logging
- 23. Servlet Containers
- 24. Fixpack Automation
- 25. Dependencies
- 26. Clustering Considerations
- 27. High Availability
- 28. Summary
- 12. Agile
  - 1. History of Agile
  - 2. Sprint0
  - 3. Scrum Team
  - 4. Managing Sprints
  - 5. Maintaining the Backlog
  - 6. Working With Story Points
  - 7. Distributed Agile
  - 8. Kaizen
  - 9. Kanban
  - 10. Comparison of Scrum and Kanban
  - 11. Introduction to Jira
  - 12. Kanban board
  - 13. Control Chart
  - 14. Cumulative Flow Diagram
  - 15. Burndown Chart
  - 16. Introduction to Rally
  - 17. Summary
- 13. Error Handling
  - 1. Introduction to Error Handling
  - 2. Zen of Troubleshooting
  - 3. Troubleshooting Skill
  - 4. Success Pattern
  - 5. Self-healing
  - 6. Proactive vs. Reactive
  - 7. Issue Handling
  - 8. Notification
  - 9. Summary
- 14. Continuous Code Quality
  - 1. Continuous Code Quality
  - 2. What is SonarQube
  - 3. SonarQube - Benefits
  - 4. SonarQube (Multilingual)
  - 5. Seven Axes of Quality
  - 6. Potential Bugs
  - 7. Tests
  - 8. Comments and Duplications
  - 9. Architecture and Design

10. Complexity
  11. SonarQube Installation
  12. SonarQube Components
  13. Code Quality (LOC, Code Smells)
  14. Code Quality (Project Files)
  15. Code Quality (Code)
  16. Summary
15. Best Practices
1. Who are the folks using the various solutions
  2. DevOps Implementation Basics
  3. DevOps Implementation Checklist
  4. Lean Patterns
  5. Process Theory
  6. Tool Pattern
  7. Culture Alignment
  8. Culture Antipatterns
  9. Process Antipatterns
  10. Technology Antipatterns
  11. Tale of Two Audiences
  12. Best Practices
  13. Best Practices Continued
  14. Summary

### Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

### Class Prerequisites

Experience in the following *is required* for this DevOps class:

- An understanding of the basic concepts in version control, release management, automation, governance, and infrastructure.

### Prerequisite Courses



Courses that can help you meet these prerequisites:

- [DevOps Fundamentals Training](#)