
venolearn
EXPERIENCE BEYOND IT TRAINING

What is Dev Ops?

The tools and practices that enable an organization to increase its ability to deliver high-quality applications and services at a good speed is called Dev Ops. It is essentially a merger between the Development and Operations teams, often combined into one team that looks after the entire software development right from ideation through to delivery, and also monitors the performance for any scope of up gradation

What are the benefits of Dev Ops?

Many organizations have adopted the Dev Ops principle and have seen remarkable results. Here are some of its benefits:

- Continuous and timely software delivery, easy management with less complexity, faster detection and resolution to problems.
- Excellent team productivity, great scope for professional development as it makes a developer familiar with the different aspects of software development.
- Stable operating environments for businesses with excellent collaboration and synergy between the teams.

Dev Ops makes software development more efficient leaving companies with more time to innovate as opposed to focusing on problem solving.

How will learning Dev Ops benefit you?

- Most software development practices now involve the Dev Ops principles.
- There's a great demand for Dev Ops Engineer at present and will continue to remain so in the future.
- Besides getting a good industry exposure, Dev Ops professionals also enjoy the benefits of good salary packages.

Course Duration

- 30 Working Days

Dev Ops Curriculum

1) Introduction to DevOps

- Software Delivery Life Cycle
- Collaborative Development
- Continuous Testing and Integration
- Continuous Release and Deployment
- Dev and Ops, Introduction
- What is DevOps?
- DevOps' Objectives
- Prerequisites for DevOps Success
- Alignment with the Business Needs
- DevOps Adoption Steps
- Select DevOps Techniques and Practices
- Service Quality Metrics
- Infrastructure as Code
- Infrastructure in the Cloud
- DevOps on the Cloud
- Continuous Application Monitoring

2) Dev Ops Tools

- The Collaborative Lifecycle Management Diagram
- Continuous Integration (CI) Systems
- System Configuration Automation
- Build and Dependency Management Systems
- Containerization Tools

3) Introduction to Continuous Integration and Jenkins

- What is Continuous Integration (CI)
- Benefits of CI
- CI Practices
- CI Tools
- Typical Setup for Continuous Integration
- Jenkins Continuous Integration
- Jenkins Features
- Installing Jenkins as a Windows Service
- Jenkins Installation Instructions
- Running Jenkins on an Application Server

- Configuring Source Code Management (SCM)
- Working with Subversion
- Configuring Jenkins jobs
- Build Triggers
- Schedule Build Jobs
- Polling the SCM
- Ant Build Steps
- Jenkins Security
- Two axes of Jenkins Security
- Activating Jenkins Security
- Securing Jenkins
- Configure Authentication
- Using Jenkins Internal User Database
- Creating Users
- Authorization
- Matrix-based Security
- Disabling Jenkins Security

4) Jenkins as Plugin

- Introduction to Jenkins plugins
- Installing Jenkins plugins using interface
- Demo: Installing 'Role Strategy Plugin'
- Jenkins Plugins - SCM
- Jenkins Plugins - Build and Test
- Jenkins Plugins - Analyzers

5) Distributed Builds with Jenkins

- Jenkins Distributed Build Overview
- Need for Distributed Build
- Distributed Build Architecture
- Steps to add a Linux Slave Node
- Demo: Configuring a Ubuntu Slave Node
- Steps to add a Windows Slave Node

6) Introduction to Cloud Computing

- History of Cloud
- Cloud Computing at a Glance
- Capacity Planning Concepts and Challenges
- Coping with Computing Demand the Traditional Way
- Coping with Computing Demand the Cloud Way
- Grid Computing vs Cloud Computing

- What Drives Cloud Adoption?
- Five Characteristics of the Cloud
- Cloud Service Models
- Cloud Deployment Models
- Virtualization
- Cloud Infrastructure Virtual Machines
- A Bootable OS Image
- Block Storage for Instances
- Cloud Object Storage
- Cloud Risks to Consider
- Amazon Web Services (AWS)

6) Introduction to Puppet

- What is Puppet
- Puppet's Domain Specific Language
- "Infrastructureascode" in Puppet
- Example of the Puppet DSL
- Main Puppet Artifacts
- Puppet Design
- Puppet Workflow Orchestration
- Puppet Lab Services
- Puppet Enterprise Licensing
- Puppet Enterprise Support
- Puppet Enterprise Feature Set