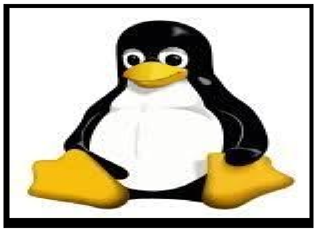

Linux

What is DevOps

DevOps is a process to deliver software quickly and effectively by fostering collaboration between different departments like developers and Ops team. This is supported by automation, and by analyzing organization-wide metrics to see what's going right and what's going wrong.

Who can learn

- ✔ People who is having B.E/B.Tech/MCA/B.Sc/M.Sc/M.Sc degree
- ✔ Developers/Testers/Middleware/Ops/Systems admins
- ✔ Anyone who is looking for IT prospects

Do I need experience

- ✔ Absolutely not

Course Objective

Course content is designed as per latest requirements, tools used in most of the IT organizations and prepared to balance job descriptions for interviews and infrastructure.

Devops Tools Covered

1. **Kubernets**
2. **Docker**
3. **Chef**
4. **Ansible**
5. **Jenkins**
6. **Linux**
7. **Maven**
8. **SVN**
9. **ANT**
10. **GIT**
11. **Github**
12. **Nagios**
13. **Git hub**
14. **ITIL Process**
15. **DSL-ruby Script**
16. **Yaml scritpt**
17. **DSL-groovy script**

Fundamentals

1. Devops Overview

- ❖ **Why Devops**
  - **a.** Organizational view
  - b. Stakeholders view
  - c. Developers/ Testers view
  - d. Operations view
- ❖ **DevOps definition**
- ❖ **Agile and SDLC**
  - **a.** Introduction
  - **b.** Phases
  - **c.** Roles
- ❖ **DevOps and Agile**
  - a. Overview
  - **b.** Roles of Dev team
  - **c.** Roles Ops team
- ❖ **How effective DevOps is!**
- ❖ **Roles and responsibilities**
  - **a.** Who is a DevOps engineer
  - **b.** what he does
  - **c.** Other teams to communicate
  - **d.** Automate
- ❖ **Overview of DevOps Automation**
  - **a.** Introduction
  - **b.** Build tools
  - **c.** Source code management/version control tools
  - **d.** Configuration tools
  - **e.** Monitoring tools
  - f. Continuous Integration
  - g. Continuous Testing
  - h. Continuous Deployment
  - **i.** Planning
  - **j.** Issue Tracking
  - k. Types of environments

2. Cloud Computing

- ❖ **Cloud Computing Models**

a. Software As A Service (SAAS)
b. Platform As A Service(PAAS)
c. Infrastructure As A Service(IAAS)

- **Understanding Public, Private and Hybrid clouds**
- **Cloud Computing Benefits**
- **Cloud Computing Challenges**
- **DevOps in Cloud**

3. Linux

- **Commands**
- **Files and hierarchy**
- **Remote server access using ssh**
- **Bash Scripting**

DevOps Tools

1. Build Tools

- **ANT**

a. Ant Introduction
b. Ant tasks
c. Ant properties and command line calls
d. Advanced ant tasks and external tasks

- **Maven**

a. Environment Setup and configuration
b. Build life cycle
c. Build profiles
d. Repositories
e. Plug-ins
f. Creating, Building and testing projects
g. Project templates
h. Snapshots
i. Build automation
j. Manage dependencies
k. Web application

2. Source Code Management Tools

- **SVN (Subversion)**
  - **a.** SVN introduction
  - **b.** SVN and Apache installation and configurations
  - **c.** User Administration
  - **d.** Directory structure
  - **e.** SVN Commands
  - **f.** Branching and merging strategies
  - **g.** Configuration management
- **Git**
  - **a.** Installation
  - **b.** Various levels of configuration
  - **c.** Staging and Committing
  - **d.** Tracking

3. Continuous Integration

- **Jenkins**
  - **1. Jenkins Introduction**
  - **2. Installation and configuration**
    - **a.** Prerequisites
    - **b.** Download & installation
    - **c.** Configuration tour
  - **3. Managing Jenkins**
    - a. Managing Jenkins
    - b. Managing Credentials
    - c. Plugin Management
    - d. Jenkins Backup
    - e. Create a Build Slave
  - 4. **Creating Application Builds**
    - a. Anatomy of the build
    - b. Cloning sample project
    - c. Manual compilation with Maven

- d. Manually Testing, Packaging and Running the App
- e. Creating a Jenkins Job and configuring a Git Repo
- f. Compiling in Jenkins
- g. Browsing the workspace in Jenkins
- h. App Packaging in Jenkins
- i. Archiving artifacts
- j. Cleaning up Past Builds
- k. Build time trend
- l. The jenkins Dashboard
- m. Troubleshooting build failures
- n. Importing Job config.xml
- o. Anatomy of the job
- p. Build linking upstream and downstream

5. **Plugins**
    - a. Introduction
    - b. Plugin Architecture
    - c. Extension Points
    - d. Getting Plugins
    - e. Plugin Wiki
    - f. Useful Plugins Overview
    - g. Source Code Plugins
    - h. Trigger Plugins
    - i. Build Tool Plugins
    - j. Wrapper Plugins
    - k. Notifier Plugins
    - l. Reporting Plugins
    - m. Artifact & UI Plugins
    - n. Installing a plugin
    - o. Plugin configuration
    - p. Security Overview
6. **Continuous Testing and Continuous Integration and Testing**
    - a. Adding steps to Freestyle Project
    - b. Creating a Pipeline job to execute Maven
    - c. Archiving in a Pipeline
    - d. Checking out git repository in pipeline
    - e. The Master Agent Model
    - f. Allocating a node and workspace in Pipeline
    - g. High level progress with Pipeline stages
    - h. Triggering Automated Builds
    - i. Configuring an Email Server

   j. Notifications when a build fails
   k. Duplicating a job
   l. Executing unit tests
   m. Executing selenium tests
   n. Visualizing Test Results

7. **Finding and Managing Plugins**
   a. The need for plugins
   b. Integrated Code Coverage
   c. Assessing a plugin
   d. Installing the HTML Publisher plugin
   e. Publishing HTML Reports
   f. Testing Plugins and Plugin Types
   g. BlueOcean UI Plugin

8. **Building Continuous Delivery Pipeline**
   a. Continuous Delivery
   b. Backup and Restore
   c. Starting point and Pipeline stashing
   d. Browsing Workspaces in Pipeline Jobs
   e. A Second Node Allocation
   f. Adding an Agent Node
   g. Setup parallel integration testing in a pipeline
   h. Executing and Monitoring Parallel pipelines
   i. Manual Approval for Deployments
   j. Setup Deployment to staging
   k. Executing a Deployment pipeline
   l. Checking pipeline script to Git

4. Configuration Management Tools

● Chef

1. **Introduction to chef**

- **a.** Defining chef
- **b.** Common Chef Terminology
- **c.** Chef Server
- **d.** Chef Workstation
- **e.** Chef-Repo
- **f.** Chef-Client
- **g.** Server and Nodes
- **h.** Chef Configuration Concepts

2. **Setting up the Environment**
    - **a.** Intro to ChefDK
    - **b.** Chef Workstation Setup
3. **Chef Server**
    - **a.** Installing Chef Server
    - **b.** Chef-Repo, Setting Up the Work Station, and Bootstrapping
    - **c.** Chef Solo vs Chef Zero vs Chef Server
    - **d.** Chef Client, Nodes and Run Lists
    - **e.** Building A Quick Apache Cookbook
    - **f.** Managing Node Run_Lists
    - **g.** Chef-Client Configuration
4. **Resources**
    - **a.** Understanding Chef and Chef Convergence
    - **b.** Common Chef Resources
    - **c.** Default Resource Actions
    - **d.** Applying Chef Resources Hands On
    - **e.** Working with not_if and only_if Guards
    - **f.** Extending Chef with Custom Resources
5. **Recipes and Cookbooks**
    - **a.** Understanding Chef Recipes and Run Lists
    - **b.** Understanding Chef Cookbooks Generating
    - **c.** Cookbook
    - **d.** Cookbook Pro-Tips
6. **Local Cookbook Development Basics**
    - **a.** Generators
    - **b.** Test Driven Development
    - **c.** ChefSpec
    - **d.** Test Kitchen Configuration
    - **e.** Using Test Kitchen
    - f. InSpec
    - **g.** Static Code Analysis
    - **h.** Troubleshooting

7. **Cookbook Components**
    a. Cookbook Structure
    b. Metadata Anatomy
    c. Versioning
    d. Attributes
    e. Common Resources
    f. Templates
    g. Libraries
    h. Custom Resources
8. **Design Patterns and Theory**
    a. Cookbook Disposition
    b. Wrapper Cookbooks
    c. Community Cookbooks
    d. Managing Cookbook Dependencies
    e. Data Bags
    f. Vault
    g. Search
9. **Nodes and Search**
    a. Node Object
    b. Working With Ohai and Node Attributes
    c. Understanding Search
    d. knife Search
10. **Roles and Environments**
    a. Setting Up A New Node
    b. Understanding Roles
    c. Creating Roles
    d. Understanding Environments
    e. Bootstrapping The Staging Node
    f. Creating And Using Environments
11. **Desired State Configuration**
    a. Imperative Vs Declarative Approach To Configuration Management
    b. Pull vs Push Approach
    c. Windows DSC
    d. Removing Resources From Recipes
12. **Chef Supermarket**
    a. Chef Supermarket
    b. Using A Private Supermarket
13. **Building Web Server Cookbook**
    a. Getting Setup

- **b.** Adding Platform Support to the Cookbook
- **c.** Adding Local Chef-Repo to Github
- **d.** Install and configure Chef Reporting

14. Chef Offerings

- **a.** Chef Automate: Overview
- **b.** Chef Automate: Workflow

15. Deploying Nodes In Production

- **Ansible**

1. **Introduction**
    - a. Introduction to Ansible
    - b. Ansible vs. Other Tools Ansible vs. Other Tools
    - c. Ansible Documentation: Modules
2. **Setup and Configuration**
    - a. Test Environment Setup
    - b. Download and Installation
    - c. Ansible Configuration File
    - d. Ansible Python Dependencies
    - e. The HOSTS File
    - f. Overriding the Default HOSTS File
    - g. Overriding the Default System Ansible.Cfg File
    - h. Overriding the Default Roles Path
    - i. Understanding the core components of Ansible
    - j. Ad-hoc commands in Ansible
3. **Use both static and dynamic inventories to define groups of hosts**
    - a. Overview of static and dynamic inventories in Ansible
    - b. Static Inventories
    - c. Dynamic Inventories
4. **Ansible Playbooks**
    - a. Configuring Your 'Ansible' Account
    - b. Ansible Command Line
    - c. System Facts

   d. System Facts: Common Values for Playbooks
   e. Our First Playbook
   f. Variables: Inclusion Types
   g. Create a Playbook from Outline
   h. Optimizing Playbook
   i. Taking Playbook for a Dry Run
   j. Simple Variable Substitution
   k. Lookups
   l. RunOnce
   m. Local Actions
   n. Loops
   o. Conditionals
   p. Until
   q. Notify
   r. Vault
   s. Prompt - Interactive Playbook
   t. Basic Include Statements Tags
   u. Basic Error Handling
   v. Jinja2 Templates
   w. LocalAction
   x. DelegateTo
   y. Use a playbook to copy a program and customize it for the target host.

5. **Ansible Modules**
   a. Commonly used Modules
   b. Using modules in playbooks
   c. The 'Setup' Module
   d. The 'File' Module
   e. The 'Pause' Module
   f. The 'WaitFor' Module
   g. The 'Yum' Module
   h. The 'Apt' Module
   i. The 'Service' Module
   j. The 'Copy' Module
   k. The 'Command' Module
   l. The 'Cron' Module
   m. The 'Debug' Module
   n. The 'Fetch' Module
   o. The 'User' Module
   p. The 'AT' Module
   q. The 'DNF' Module
   r. The 'Apache2_Module' Module
   s. The 'SetFact' Module
   t. The 'Stat' Module
   u. The 'Script' Module
   v. The 'Shell' Module

   w. The 'SELinux' Module
   x. The 'SEBoolean' Module
   y. The 'Raw' Module
   z. The 'Ping' Module
   aa. The 'Package' Module
   bb. The 'Unarchive' Module
   cc. The 'HTPasswd' Module
   dd. The 'GetURL' Module
   ee. The 'Group' Module
   ff. The 'Mail' Module
   gg. The 'Filesystem' Module
   hh. The 'Mount' Module
   ii. The 'Notify' Module
   jj. The 'AptRepo' Module
   kk. The ' AptKey' Module
   ll. The ' ACL' Module
   mm. The 'Git' Module
   nn. Creating a Jinja2 Template File
   oo. The 'Template' Module
   pp. The 'MySQL_DB' Module
   qq. The 'MySQL_User' Module
   rr. The 'Kernel_Blacklist' Module

6. **Create and use templates to create customized configuration files**
   a. Introduction
   b. Templates
7. **Working with Ansible facts and variables.**
   a. Let see how we get ansible facts and how we use facts.d
   b. Using Ansible facts
   c. Using variables to gather server info
8. **Roles**
   a. Introduction to Roles
   b. Roles - The Directory Structure
   c. Role Based Tasks
   d. Task Order - Pre and Post Tasks
   e. Roles - Conditional Execution
   f. Roles - Variable Substitution
   g. Roles – Handlers
   h. Roles - Using Notification
   i. Roles - Configuring Alternate Roles Paths
   j. Roles - Conditional Include Statements
   k. Roles - Waiting For Events
   l. Roles - Executing a Task Until
   m. Roles - Using Tags
   n. Roles - Breaking a Playbook Into a Role
   o. Roles - Passing Variables from Command Line
   p. Roles - Using Jinja2 Templates

   q. Roles – DelegateTo
   r. Roles – LocalAction
   s. Roles - Lets create a role to install apache. Lets
   t. use the previous role and add a new one. Lets
   u. build on the previous roles

9. **Download roles from Ansible Galaxy and use them**
   a. Ansible galaxy and how its used
   b. Lets use multiple roles
10. **Ansible Command Line Usage**
   a. Ansible Command Line - Installing Packages
   b. Ansible Command Line - Services and Hosts
   c. Ansible Command Line - Commands and Shells
   d. Ansible Command Line - Managing Users
   e. Ansible Command Line - Create and Manage Cron Jobs
   f. Ansible Command Line - Running Arbitrary Commands
   g. Ansible Command Line - Output Tree
11. **Managing Parallelism**
   a. What is parallelism?
   b. Parallelism in a playbook
12. **Using ansible-vault in playbooks to protect sensitive data**
   a. Lets discuss ansible-vault and see an example
   b. Options useable with ansible-vault
13. **Install ansible tower and use it to manage systems**
   a. Installing a trial version of ansible tower
   b. Log into our Ansible tower and run a sample task Lets
   c. add to the inventory and run a task against them

5. Containerization

● Docker

1. **Introduction**
2. **Installing Docker**
   a. Installing Docker on Windows
   b. Installing Docker on Linux
3. **Working with Containers**
   a. What is container
   b. Docker run command

   c. Theory of pulling and Running Containers
   d. Working with images
   e. Container Life cycle
4. **Swarm Mode & Microservices**
   a. Swarm Mode Theory
   b. Configuring Swarm Mode
   c. Services
   d. Scaling Services
   e. Rolling Updates
   f. Stacks & DABs
5. **Introducing the App**
   a. The App
   b. The Dockerfile
   c. Pushing App to Github
6. **Configuring Test Builds**
   a. Performing test Builds
7. **Pushing App to Production**

6. Monitoring

1. **Nagios**
   a. Installation of Nagios
   b. Configuring Nagios
   c. Configuring Nagios
   d. Triggering Alerts
2. **Elasticsearch+logstash+Kibana**
   a. Installation of ELK stack
   b. Configuring the ELK Stack
   c. Monitoring logs with ELK

7. ITIL Process

1. Release process
2. Pre build and post build activities
3. Incident management

8. Cloud Environment

1. Environment setup
2. Cloud deployment
3. CI and CD activities in Cloud

kubernetes

1. **Introduction to Kubernetes**
    - **a.** Kubernetes Introduction
    - **b.** Containers Introduction
    - **c.** Kubernetes Setup
    - **d.** Local Setup with minikube
    - **e.** Introduction to Minikube
    - **f.** Installing Kubernetes using the Docker Client
    - **g.** Minikube vs Docker Client vs Kops vs Kubeadm
    - **h.** Introduction to Kops
    - **i.** Preparing kops install
    - **j.** Preparing AWS for kops install
    - **k.** Cluster setup on AWS using kops
    - **l.** Running first app on Kubernetes
    - **m.** Running first app on Kubernetes
    - **n.** Useful commands
    - **o.** Service with LoadBalancer
    - **p.** Service with AWS ELB LoadBalancer

2. **Kubernetes services and architecture**
    - a. Node Architecture
    - b. Replication Controller
    - c. Services
    - d. Labels
    - e. NodeSelector using Labels
    - f. Healthchecks
    - g. Readiness Probe
    - h. Liveness and Readiness probe
    - i. Pod State
    - j. Pod Lifecycle
    - k. Secrets
    - l. Credentials using Volumes

m. Running Wordpress on Kubernetes

3. Kubernetes Administration

a. The Kubernetes Master Services
b. Resource Quotas
c. Namespaces
d. Demo: Namespace quotas
e. User Management
f. Demo: Adding Users
g. RBAC
h. Demo: RBAC
i. Networking
j. Node Maintenance
k. Demo: Node Maintenance
l. High Availability
m. Demo: High Availability
n. TLS on ELB using Annotations
o. Demo: TLS on ELB

4. Packaging and Deploying on Kubernetes

a. Introduction to Helm
b. Demo: Helm
c. Creating your own Helm Charts
d. Demo: Creating your own Helm Charts
e. Demo: nodejs app Helm Chart
f. Demo: Setting up a Helm Repository on S3
g. Demo: Building and Deploying Helm Charts with Jenkins

5. Advanced Kubernetes

a. Service Discovery
b. ConfigMap
c. Ingress Controller
d. External DNS
e. Volumes
f. Volumes Autoprovisioning
g. Pod Presets
h. StatefulSets
i. Daemon Sets
j. Resource Usage Monitoring using Metrics Server
k. Autoscaling

l. Affinity / Anti-Affinity
m. Interpod Affinity and Anti-affinity
n. Taints and Tolerations
o. Custom Resource Definitions (CRDs)
p. Operators

6. Serverless on Kubernetes

a. Introduction to Serverless
b. Introduction to Kubeless
c. Creating Functions with Kubeless
d. Triggering Kubeless Functions with Kafka

7. Microservices

a. Introduction to Istio
b. Istio Installation
c. An Istio enabled app
d. Advanced routing with Istio
e. Canary Deployments
f. Retries
g. Mutual TLS
h. RBAC with Istio
i. End-user authentication with istio (JWT)
j. Istio Egress traffic
k. Distributed Tracing with Jaeger
l. Istio's Grafana Metrics

8. Kubeadmin

a. Installing Kubernetes using kubeadm
b. Introduction to kubeadm
c. Demo on Kubeadmin

Faculty:

Duration:

Fee: