TechyEdz Solutions

Training | Consulting | Developement | Outsourcing

DevOps Masters Program

9032803832

contact@techyedz.com

9032803832

www.techyedz.com

DevOps Masters Program

Course Overview:

This Masters Program makes you proficient in DevOps principles like CI/CD, Continuous Monitoring and Continuous Delivery, using tools like Puppet, Nagios, Chef, Docker, Git & Jenkins. It includes training on Linux, Python, Docker, AWS DevOps Certification Training and Splunk. In this Master program in DevOps training and certification to professionals and corporates on Deployment and automation using DevOps tools – Linux, AWS, Python, Ansible, GIT, Jenkins, and Docker.

Course Overview:

LINUX Basic and Admin

- Linux OS Introduction
- Importance of Linux in DevOps
- Linux Basic Command Utilities
- Linux Administration
- Environment Variables
- Networking
- Linux Server Installation
- RPM and YUM Installation

Python Scripting

- Python Introduction
- Features of Python
- Python vs Perl
- Python vs shell
- Interactive mode of programming using python
- Reserved words
- Python indentation, Python variables
- Python data types
- Numbers, Strings, Lists, Tuples
- Dictionaries, Loops, Operators
- Break Continue, pass,
- Command line arguemtns,Functions
- Modules
- Files, Exception Handling

- Object Oriented concepts
- Class, Object, Inheritance

AWS

- Amazon EC2
- Amazon Elastic Load Balancer
- Amazon Route 53
- Amazon VPC
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic Block Store (Amazon EBS)
- Amazon Content Delivery Network (Amazon CloudFront)
- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Simple Notification Service (SNS)
- Amazon CloudWatch
- Amazon CloudTrail
- Amazon Simple Email Service (Amazon SES)
- AWS Identity and Access Management (IAM)
- Amazon Command Line Interface API
- S3-Browser

DevOps

Source code management

- What is SCM
- What is a version control system
- Types of version controls
- Diff b/w CVS & DVS

GIT: Version Control

- **Introduction**
  - What is Git
  - About Version Control System and Types
  - Difference between CVCS and DVCS

- A short history of GIT
- GIT Basics
- GIT Command Line

- **Installing Git**
  - Installing on Linux
  - Installing on Windows
  - Initial setup
- **Git Essentials**
  - Creating repository
  - Cloning, check-in and committing
  - Fetch pull and remote
  - Branching
  - Creating the Branches, switching the branches, merging the branches.

Build Tools

- What is Maven?
- Maven Evolution
- Objective
- Convention over Configuration
- Features of Maven

Jenkins – Continuous Integration

- Introduction.
  - Understanding continuous integration
  - Introduction about Jenkins
  - Build Cycle
  - Jenkins Architecture
- **Installation**
  - Obtaining and installing Jenkins
  - Installing and configuring Jenkins using WAR and RPM
  - Java installation and configuration
  - Maven Installation
  - Exploring Jenkins Dashboard.
- **Jobs**
  - Creating Jobs
  - Running the Jobs

- Setting up the global environments for Jobs
- Adding and updating Plugins
- Disabling and deleting jobs

➢ **Build Deployments**

- Understanding Deployment.
- Tomcat installation and configuration

➢ **Securing Jenkins**

- Authentication
- Jenkins Plugin
- Authorization
- Confidentiality
- Creating users
- Best Practices for Jenkins

Docker– Containers.

➢ **Introduction**

- What is a Docker
- Use case of Docker
- Platforms for Docker
- Dockers vs. Virtualization

➢ **Architecture**

- Docker Architecture.
- Understanding the Docker components

➢ **Installation**

- Installing Docker on Linux.
- Understanding Installation of Docker on windows.
- Some Docker commands.
- Provisioning

➢ **Docker Hub.**

- Downloading Docker images.
- Uploading the images in Docker Registry and AWS ECS
- Understanding the containers
- Running commands in container.
- Running multiple containers.

➢ **Custom images**

- Creating a custom image.
- Running a container from the custom image.

- Publishing the custom image.

- **Docker Networking**
  - Accessing containers
  - Linking containers
  - Exposing container ports
  - Container Routing
- **Docker Compose**
  - Installing The Docker compose
  - Terminology in Docker compose
  - Build word press site using Docker compose

VAGRANT

- Introduction
- Setting up Vagrant
- Using Vagrant, Vagrantfiles, Boxes, Synced folders, Networking.
- Automated Provisioning with Shell, Puppet and Chef
- Private and Public Networking.
- Multiple Machines with Vagrant
- Box Management

ANSIBLE

- Introduction to Ansible
- Ansible Server Configuration
- Infrastructure Management
- SSH Connection in Ansible Master
- YAML Scripts
- **Host Inventory**
  - Hosts and Groups
  - Host Variables
  - Group Variables
  - Host and Group Specific Data
- Ad-hoc Commands
- **Playbooks**
  - Variables
  - Conditionals
  - Loops

- Blocks
- Handlers
- Templates

- **Modules**
  - Core Modules
  - Extra Modules
- Ansible Roles

Agile and Scrum

- Agile Principles
- Agile and Scrum Methodologies
- Scrum Roles

Puppet for configuration management

- **What is Puppet**?
  - How puppet works
  - Puppet Architecture
  - Master and Agents
  - Puppet terminology and about Manifests
- **Installation and Configuration**
  - Installing Puppet
  - Configuring Puppet Master and Agent
  - Connecting Agents
- **Puppet Master**
  - Puppet configuration files
- **Puppet Language Basics**
  - The declarative language
  - Resources
  - Using Basic resources like file,exec,package service
  - Resource Collectors
  - Virtual Resources
  - Exported Resources
  - Manifests
  - Modules and Classes
  - Class Parameters

- Defined Type

- **Templates**
  - Static Contents Explosion
  - Using Dynamic Content with Templates
  - Templates Overview
  - ERB
- **Example Code Manifests/Modules**
  - NTP Module
  - Users Module
  - SSH
  - Sudo
- **Puppet Forge**
  - Understanding the Puppet Forge
  - Module structure
  - Install LAMP with preexisting modules
  - Installing Apache Tomcat with Puppet Modules

Chef for configuration management

- **Overview of Chef**
  - Common Chef Terminology (Server,
  - Workstation, Client, Repository etc.)
  - Servers and Nodes
  - Chef Configuration Concepts
- **Workstation Setup**
  - How to configure knife
  - Execute some commands to test connection between knife and workstation
- **Organization Setup**
  - Create organization
  - Add yourself and node to organization
- **Test Node Setup**
  - Create a server and add to organization
  - Check node details using knife
- **Node Objects and Search**
  - How to Add Run list to Node
  - Check node Details
- **Environments**

- How to create Environments
- Add servers to environments

- **Roles**
  - Create roles
  - Add Roles to organization
- **Attributes**
  - Understanding of Attributes
  - Creating Custom Attributes
  - Defining in Cookbooks
- **Data bags**
  - Understanding the data bags
  - Creating and managing the data bags
  - Creating the data bags using CLI and Chef Console
  - Sample data bags for Creating Users.

Kubernetes

- Features of Kubernetes
- Architecture of Kubernetes
- Install and Configure Kubernetes Cluster
- Introduction to Kubectl
- Kubernetes Node
- Kubernetes Jobs
- Kubernetes Service
- Kubernetes Pod
- Kubernetes Replication Controls
- Intro to Kubernetes Network policies

Prerequisites:

- There are no prerequisites for enrollment to the Masters Program. Whether you are an experienced professional working in the IT industry, or an aspirant planning to enter the world of DevOps, Masters Program is designed and developed to accommodate various professional backgrounds.

Who can attend:

- Software Developers
- Project Managers

- IT Managers
- Development Managers
- Architects

Number of Hours: 100hrs

Certifications: AWS SAA-C02 and DevOps Certified Professional

Key Features:

- One to One Training
- Online Training
- Fastrack & Normal Track
- Resume Modification
- Mock Interviews
- Video Tutorials
- Materials
- Real Time Projects
- Virtual Live Experience
- Preparing for Certification