

DVS DevOps Course Content

INTRODUCTION TO DEVOPS

- What is DevOps?
- History of DevOps
- Different Teams Involved
- DevOps definitions
- DevOps and Software Development Life Cycle
 - Waterfall Model
 - Agile Model
- DevOps main objectives
- Prerequisites for DevOps
- Continuous Testing and Integration
- Continuous Release and Deployment
- Continuous Application Monitoring
- Configuration Management
- What is Cloud?
- History and evolution of cloud
- Cloud Computing Concepts
- Public, Private, Hybrid Clouds
- IAAS, SAAS, PAAS Cloud Models
- Public Clouds
 - Amazon Web Services, Azure, Oracle Cloud, IBM Cloud
- DevOps with Cloud

MODULE 1: OPERATING SYSTEM

LINUX ADMINISTRATION

OVERVIEW

This course teaches the advanced concepts of processes, programs and the components of the Linux operating system. You learn the advanced knowledge of computer hardware, gain an understanding of open source applications in the workplace, and learn to navigate systems on a Linux desktop rudimentary commands to navigate the Linux command line.

INTRODUCTION

- A Linux Introduction
- Open Source Philosophy
- Distributions
- Basic Kernel & Shell Architecture

HOW AND WHERE TO INSTALL LINUX

- Using the Linux Essentials Lab Server
- How to Install CentOS 7 with Virtual Box & VMware

ACCESSING SERVER

- Usage of putty
- Password less login
- Booting Process

COMMAND LINE BASICS

- Basic Shell
- Command Line Syntax - Basic Commands
- Ls, pwd, history, cat
- Cp, mv, tail, head
- Touch, cd, find, less
- view, rm, man, chmod
- chown, chgrp, grep, cd,
- cd., man

USING DIRECTORIES AND LISTING FILES

- The Linux File System
- Files, Directories
- Hidden Files and Directories

CREATING, MOVING AND DELETING FILES

- Files and Directories
- Case Sensitivity
- Simple Globbing and Quoting

TEXT EDITORS

- Vi, Vim, nano

CREATING USERS AND GROUPS AND OWNERSHIP

- User IDs
- User Commands
- Group Commands
- File/Directory Permissions and Owners

PACKAGE AND REPOSITORY MANAGEMENT

- Package Management Tools and Repositories
- Hands-On - Package Management (YUM, rpm) - For Use with Linux Essentials Lab Servers

NETWORK MANAGEMENT

- Network Configuration
- ethtool,

- ifconfig,
- netstat,
- nslookup,
- ping,
- scp,

FILE SYSTEM MANAGEMENT

- Standard Partition
- Logical Volume Management

SYSTEM MONITORING

- 1.Top 2. Vmstat 3.Sar 4.Ping 5.Ps

SERVICES MANAGEMENT

- Network File system (NFS)
- Apache (httpd)

MODULE 2: CONFIGURATION MANAGEMENT

ANSIBLE

OVERVIEW

Ansible continues to gain traction as a powerful, enterprise level configuration and deployment management tool. With its standardized Playbook formatting and reliance on Python standards, it is easy to use, quick to learn and puts the power of automation at everyone's fingertips. In this course, we will cover Ansible configuration, modules, command line usage and Playbook building. By the time you are done, you will be able to use Ansible to automate and manage your DevOps infrastructure.

INTRODUCTION

- Ansible what is it good for !
- How Ansible works
- What's so great about ansible?
 - Easy-to-Read syntax
- Nothing to install on the remote hosts
- Push-based
- Ansible Scales down
- Built-in modules
- Installing and configuring Ansible
- Setting up a server for testing

YAML BASICS

- Comments
- Variables

- booleans
- lists
- dictionaries
- multilines
- Dict of Dicts
- Short ways to write YAML code

INVENTORY - DESCRIBING YOUR SERVERS

- The Inventory File
- Hosts and ansible.cfg configuration
- Real-time lab environment setup for class
- Password less login configuration
- Other Inventory parameters
 - ansible_ssh_host, ansible_ssh_port, ansible_ssh_user,
 - ansible_connection, ansible_ssh_pass, ansible_shell_type
- Organizing Prod, Syst & Test environments under groups

ANSIBLE MODULES

- Command
- File
- Copy
- Template
- Script
- User
- Yum
- Apt
- Service
- Debug

VARIABLES AND FACTS

- Defining variables in playbooks
- Viewing the values of variables
- Registering variables
- Facts
 - Viewing all facts associated with a server
 - Viewing a subset of facts
 - Local facts
 - Using set_fact to define a new variable
 - Built-in variables
 - Host_vars
 - Inventory_hostname
 - Groups
- Setting variables on the command line
- Precedence

CONDITIONS AND LOOPS

- Condition
 - When
- Loops:
 - with_items
 - with_fileglob
 - with_lines
 - with_dict
 - with_sequence

SPECIAL CASES

- ignore_errors
- local_action
- delegate_to
- serial
- failed_when
- ansible vault

DEBUGGING ANSIBLE PLAY BOOKS

- Limiting Which Hosts To Run
- Debugging SSH Issues
- Debug Mode
- Checking Your Playbook Before Execution
 - Syntax Check
 - List Hosts
 - Listing All the Tasks
 - Check Mode
 - Step
 - Start-at-Task
 - Tagging

SCALE UP YOUR PLAYBOOKS USING ROLES

- Basic Structure of a Role
- Using Roles in our playbooks
- Ansible Galaxy

REAL TIME PROJECTS USING ANSIBLE

- Automating installation and configuration of Apache in test environment
- Automating deployment of AWS EC2 instances and configuration

MODULE 3: REPOSITORY MANAGEMENT

GIT & GIT HUB

Overview

Over the length of this course, we will start at the very beginning of revision and source control the way that it is intended to be done using Git client and server. Once we have a firm handle on how to manage our files at the command prompt and in our own repositories, we will take a look at several of the more commercial or public Git hosting sites - Github and Bitbucket. Finally, we will install the Github clone called Gitlab and take a deep dive in how source control can be used in an online environment that supports team collaboration and build automation using Jenkins.

Introduction

- Introduction to Git and Git hub

Installation and Configuration

- Installing Git
- Basic Configuration

Git Basics

- Empty Repositories
- Git Basics
- Git Ignore
- Cloning
- Cloning: Local Repositories
- Cloning: Remote Repositories

Tagging, Branching and Merging

- Tags
- Branches
- Merging

Logging and Repository Auditing

- Git Log

Working with Github

- Introduction to Github
- Secure Communication
- Working with Github

MODULE 4: BUILD AND RELEASE MANAGEMENT (CI & CD)

Jenkins

Overview

Students will learn how to use Jenkins 2.x.x at a proficient level. This includes the creation and configuration of jobs and builds, testing, common plugin usage and building pipelines. They will gain knowledge of common CI/CD concepts, and “CD as code” best practices. The material in this course will also help students prepare to pass the Certified Jenkins Engineer 2017 certification.

Introduction

- Introduction to Jenkins

CI/CD Concepts

- Continuous Integration and Continuous Delivery

Installing and Configuring Jenkins

- Prerequisites
- Jenkins Install
- Adding a Jenkins Slave
- Managing Credentials
- Configuration Tour
- Plugin Manager

Managing Jenkins

- The Dashboard
- User Management and Security
- Jenkins Backup - Using Plugins to Manage Your System
- Setting Up Git and Github

Projects

- Freestyle Project Configuration
- Source Code Management and the Git Plugin
- Git Hooks and Other Build Triggers
- Workspace Environment Variables
- Managing Remote Systems with Jenkins
- Upstream/Downstream Projects and the Parameterized Trigger Plugin
- Folders
- Views

MODULE 5: CONTAINERIZATION

Dockers & Kubernetes

Course Details

Overview:

This course will explore Docker from the very basics of installation and function to an in depth review of the use cases and advanced features. We will talk about how Docker is architected in order to provide a better understanding of how to manage Linux Containers using the Docker Client. Once we have a good understanding of the basics, we will dive into the advanced use cases and really uncover the power of the entire system. Now Updated for Docker 1.10+ in 2016! GOALS * Introduce and Understand Linux Containers and Application Virtualization * Relate the Architecture of Containers to the Management of Our Applications * Install and Configure Docker for Our Distribution * Explore the Most Common Use Cases for Docker * Understand the Power and Flexibility Application Virtualization Offers

Part 1 : Dockers & Containers

Learning the Basics of Docker

- Introduction to Docker
- Containers vs. Virtual Machines
- Docker Architecture
- The Docker Hub
- Docker Installation
- Creating our First Image
- Working with Multiple Images
- Packaging a Customized Container
- Running Container Commands with Docker
- Exposing our Container with Port Redirects

The Dockerfile, Builds and Network Configuration

- Dockerfile Directives: USER and RUN
- Dockerfile Directives: RUN Order of Execution
- Dockerfile Directives: ENV
- Dockerfile Directives: CMD vs. RUN
- Dockerfile Directives: ENTRYPOINT
- Dockerfile Directives: EXPOSE
- Container Volume Management
- Docker Network: List and Inspect
- Docker Network: Create and Remove
- Docker Network: Assign to Containers

Docker Commands and Structures

- Inspect Container Processes

- Previous Container Management
- Controlling Port Exposure on Containers
- Naming Our Containers
- Docker Events
- Managing and Removing Base Images
- Saving and Loading Docker Images
- Image History
- Taking Control of Our Tags
- Pushing to Docker Hub

Real Time Integration and Use Cases

- Building a Web Farm for Development and Testing (Prerequisites)
- Building a Web Farm for Development and Testing (Part One)
- Building a Web Farm for Development and Testing (Part Two)
- Building a Web Farm for Development and Testing (Part Three)
- Building a Web Farm for Development and Testing (Part Four)

Part 2: Dockers & Clusters with Kubernetes

Introduction to Kubernetes

- What is Kubernetes
- Design Overview
- Components
- Building Block of Kubernetes:
- Summar

Kubernetes Architecture

- Minions (Nodes)
- PODS
- labels
- Selectors
- Controllers
- Services

Introduction to YAML

- What is YAML
- Structure

Kubernetes Setup and Configuration

Packages and Dependencies

Install and Configure Master Controller

Install and Configure the Minions

Kubect!: Exploring our Environment

Pods, Tags and Services

Creating Pod

Checking the port forwarding

Tags, Labels and Selectors

Deployment State

Multi-Pod (Container) Replication Controller

Create and Deploy Service Definitions

Logs, Scaling and Recovery

Replicas

Executing the remote command on a pod

Auto scaling and scaling our Pods

Failure and Recovery

MODULE 6: MONITORING THE DEPLOYMENTS & BUILDS

NAGIOS

Course Details

Overview:

In this course, we will learn and practice the use of Nagios 4.2.x, which is one of the most mature monitoring packages in the industry. Although it may seem beguilingly simple at first glance, Nagios provides powerful, flexible, and extensible tools and functionality – some might say it is the pinnacle of enterprise monitoring.

Introduction and Getting Started

- Course Introduction
- Configuring Our Server
- Configuring Nagios Core

Nagios: The Basics, Part 1

- Plugins
- Checks
- Hosts, Host Objects, and Host Definitions
- Macros

- Services, Service Objects, and Service Definitions
- Commands, Command Objects, and Command Definitions
- Timeperiods and Timeperiod Definitions
- Contacts, Contact Objects, and Contact Definitions
- Notifications
- Verifying Our Configuration and Starting Nagios
- Configuring Apache
- Looking at the WebUI
- Authoring Groups
- Templates
- Adding Remote Hosts
- Monitoring Remote Hosts via Publicly Available Ports
- Installing the Nagios Remote Plugin Executor
- Active Checks Using NRPE
- Passive Checks

Nagios: Advanced Topics

- Event Handlers
- Escalations
- Host and Service Dependencies
- Reports