MODULE 1: MICROSERVICES

Dockers

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

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 Dockerfile references

- 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

Storage Management

- Storage overview
- Volume commands
- Using bind mounts

- Using volumes for persistent storage

Building and Distributing Images

- Building images
- Tagging
- Distributing images on docker hub
- Managing and Removing Base Images
- Saving and Loading Docker Images
- Image History
- Taking Control of Our Tags
- Using Multi-stage Builds

Docker Commands and Structures

- Inspect Container Processes
- Previous Container Management
- Controlling Port Exposure on Containers
- Naming Our Containers
- Docker Events

Real Time Integration and Use Cases

- Building highly available Web application using nginx
- Working on Image customization from scratch
- Creating Apache image
- Constructing Tomcat image
- Building springboot image for java applications
- Building java image for deploying java application
- Fully automated end-to-end Java application deployments

MODULE 2: PIPELINE AS CODE (PAC) – GROOVY SCRIPTING

GROOVY SCRIPTING

Course Details

Overview:

Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers. Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as the progression of the built software (called a "build") through multiple stages of testing and deployment.

Introduction and Getting Started

- Course Introduction
- Configuring Our Jenkins server
- Installing required pluigns to start the groovy scripting

Groovy: The Basics, Part 1

- Types of scripting
  - Declarative Pipeline
  - Scripting Pipeline
- Explaining Pipeline in Detail
- Step
- Node & Agent
- Stage
- Steps
- Script
- Declaring Variables
  - Global Variables
  - Defining Variables using environment
  - Defining Variables using parameters
  - Defining variables using def
- Using Operators
  - arithmetic operators
  - unary operators
  - Assignment arithmetic operators
  - Relational operators
  - Logical operators
  - Bitwise operators
  - Conditional operators
  - not operator

    - Ternary operator
- Using Decision Making
    - simple if statement
    - if-else statement
    - nested-if statement
    - switch statement
    - nested switch statement
- Using Loops
    - while loop
    - for loop
    - for-in loop
    - break statement
    - Continue Statement
- Using Methods
    - simple method
    - method with default parameters/arguments
    - method with return values
- Using I/O operations on files
    - Reading files
    - Reading the Contents of a File as an Entire String
    - write a new file
    - more examples
    - append the text to a file
    - Other fn()'s related to the file operations
    - Creating a Directory
    - Delete a file/directory
    - Copying files

Groovy: Advanced Topics

Real Time Fully automated One Click Application Deployment

- Using Dockers with Groovy
- Building customized images as per the requirement
- Construction interactive pipelines for promoting images to dev/sit/uat/prod environements
- Deploying and automating javabased applications using complete CI & CD Automation