

# **Devops for Network Engineers(5 days)**

By Dr. Vishwanath Rao

## **Day 1**

### **DevOps Fundamentals**

- \* System Development Life Cycle(SDLC)**
- \* SDLC Models**
- \* Agile Methodology (Backlog, Sprint, Scrum Master )**

### **Version Control Tool – GIT**

#### **Git Repository**

- \* Creating a Git Repository**
- \* Git Workflow**
- \* Tracking File Changes**
- \* Files or directory add to stage**
- \* Reset from stage**
- \* Ignoring Files in Git**
- \* Commit to Repository**
- \* Reverting to Earlier Commits**
- \* Deleting Files in Git**

#### **GitHub – Cloud Repository**

- \* Creating a Repository in GitHub**
- \* Creating a Repository in GitHub Using SSH**
- \* Pulling Commits from GitHub**
- \* Collaborating between Local and Remote Repository**
- \* Push local Repository to GitHub or remote Repository**
- \* Merging File Changes in Git**
- \* Issue Tracking in GitHub**

#### **Branching Merging And Rebasing in Git**

- \* Branching in Git**
- \* Merging Branches in Git**
- \* Fast Forward and Recursive Merge**
- \* Recursive MergePreview**
- \* Resolving Merge Conflicts in Git**
- \* Stashing in Git**

- \* **Rebasing in Git**
- \* **Cloning in Git**

## **Day 2**

### **Introduction**

- \* **What can you use Docker for?**
- \* **How Docker fits into the development lifecycle**
- \* **How Docker ensures consistency from development through UAT and staging, and on to production**
- \* **Example use cases of Docker in the real world**

### **The components of Docker**

- \* **Underlying technology**
- \* **Docker client and server**
- \* **Filesystem images**
- \* **Registries**
- \* **Containers**
- \* **Networking**

### **Getting set up to start using Docker**

- \* **Getting set up on Windows**
- \* **Trying out our first container**
- \* **Getting set up for production on Linux**
- \* **Tweaking your production environment for best performance**

### **Container management**

- \* **Container naming**
- \* **Starting and stopping containers**
- \* **Attaching to a container**
- \* **Seeing what is happening in a container**
- \* **Running a process inside a container**
- \* **Daemonizing a container**
- \* **Automatic container restarts**
- \* **Deleting containers when we are finished with them**

### **Docker images and repositories**

- \* **Docker images explained**
- \* **How Docker images work**
- \* **Getting a list of images**
- \* **Searching for images on a repository**
- \* **Pulling an image**

- \* Creating our own image
- \* Specify an image in a Dockerfile
- \* Building Dockerfile images
- \* Using the build cache for templating
- \* Viewing the image we have created
- \* Launching a container using our new image

## **Registries**

- \* What is the Docker hub?
- \* Pushing images to the Docker hub
- \* Running your own internal Docker registry
- \* Testing the internal registry

## **Day 3**

### **1. Core Concepts of Kubernetes**

### **2.Cluster Orchestration**

### **3.Looking at K8S Origination at Google**

### **4. Open Source**

### **4.Benefits**

### **5.Design Principles**

## **Navigating Kubernetes Architecture**

### **1.Master/Node**

### **2.Kubectl**

### **3.Replication Controller**

### **4.Kubelet**

### **5.Kube-Proxy**

### **6.Persistent Volumes**

### **7.Etcd**

### **8.High Availability**

## **Using Kubernetes Features**

### **1.Pods**

### **2.Labels**

### **3.Services**

### **4.Namespaces**

### **5.Resource Quota**

## **Access Control**

**4.Policies**

**5.Service Accounts**

**6.Secrets**

## **Day 4**

### **Networking and Kubernetes**

**1.Docker Networking**

**2.Kubernetes Networking**

**3.Pod to Pod**

**4.Exposing Services**

**5.IP Per Pod**

**6.Inter Pod Communication**

**7.Intra Pod Communication**

## **Ansible Fundamentals**

**Course introduction**

**Git, Git, Git, Git-out-of-here**

**YAML - this is not the indentation you are looking for**

**Ansible Overview**

**Inventory**

**Ansible Playbook Structure**

## **Variables, Modules, Network Fact Gathering**

**Ansible Variables**

**Ansible Modules**

**Cisco IOS Modules (basics)**

**Arista EOS Modules (basics)**

**Using cli\_command**

**Directly Passing Credentials**

**Privilege Escalation (Become/Enable)**

## **Day 5**

### **Conditionals, Loops, and Configuration Templating**

- Idempotency - what the what**
- Tags/Limit/Check Mode**
- Conditionals**
- Loops**
- Why Templating?**
- Jinja2 as part of Ansible**
- Configuration Templating Basics**
- Advanced Configuration Templating**
- Pushing Templates using Ansible Modules (Intro)**

#### **CLASS4 - Making Network Configuration Changes (Basics)**

- Ansible Network Configuration Overview**
- Collections and Collection Search Path**
- Feature Specific Modules (IOS)**
- Feature Specific Modules (EOS)**
- Feature Specific Modules (NX-OS)**
- Resource Modules**
- Write Mem and Ansible Handlers**

#### **CLASS5 - Making Network Configuration Changes (Part2)**

- Using ios\_config**
- Using nxos\_config**
- Using cli\_config**
- Pushing Templates using Ansible Modules (Expanded)**
- Using SSH Keys**

#### **CLASS6 - Composition / How the Parts Fit Together**

- Importing Variables**
- Importing Tasks**
- Importing Plays**
- Include vs Import - Dynamic vs Static**
- Roles - What are they?**
- Roles - How to use them?**

#### **Parsers and Dynamic Inventory**

- Hostvars**
- Block/Rescue/Always**

**Ansible and TextFSM**

**Ansible and Genie-Parsers**

**Ansible and RegEx**

**Dynamic Inventory - Some Python Required**