**DevOps**

****

**DevOps is intended to be a cross-functional mode of working, rather than a single DevOps tool, there are sets of multiple tools Such DevOps tools are expected to fit into one or more of these categories, reflective of key aspects of the development and delivery process:**

**Model 1: Continues Integration and Continues Delivery**

* **Git**
* **Jenkins**
* **Maven**
* **Jfrog**
* **SonarQube**

**Model 2: Configuration management**

* **Vagrant**
* **Chef**

**Model 3: Configuration management**

* **Ansible**

**Model 4: Containerization**

* **Docker**
* **Kubernetes**

**Model 5: Infrastructure automation**

* **Terraform**

**Model 6: Monitoring**

* **Elastic search**
* **Kibana**
* **Log stash**
* **Nagios**

**Why DevOps:**

* **Business Perspective**
* **IT Perspective**
* **Developer Perspective**
* **Tester Perspective**
* **Operations Perspective**

**What is DevOps:**

* **Definition**
* **Stakeholders of DevOps**

**What is SDLC:**

* **Phases of SDLC**
* **Role of Dev in SDLC**
* **Role of Ops in SDLC**

**Introduction to DevOps Automation:**

* **Introduction**
* **Tools**
* **DevOps Technology Categories**
* **Collaboration**
* **Planning**
* **Issue Tracking**
* **Configuration Management**
* **Source Control**
* **Dev Environments**
* **Continuous Integration**
* **Continuous Testing**
* **Continuous Deployment**
* **Monitoring**

**Linux Essentials:**

* **History**
* **Types of OS**
* **Flavours**
* **OS Structure**
* **File system**
* **All Linux basic commands**
* **File & directorys persmisions**
* **User & groups administration**
* **Accessing Servers with ssh**
* **How multi user works in Linux**
* **How make normal user as sudo user.**

**Shell Script:**

* **Introduction**
* **Type of shells**
* **Variables**
* **Types of variables**
* **Operators**
* **Input arguments**
* **String comparison**
* **Athematic comparison**
* **File conditions**
* **If, loops, case, functions**
* **Examples**

**Virtualization:**

* **Introduction**
* **What is Virtualization**
* **Server Virtualization**
* **Desktop Virtualization**
* **Application Virtualization**

**Apache webserver:**

* **Installations**
* **Configuration**

**Tomcat application server:**

* **Installation, configuration**
* **Tomcat manager**
* **Application management**
* **Application deployment**

**DevOps Tools**

**Model 1: Continues Integration and Continues Delivery**

**Source Code Management:**

**What is Version Control System?**

**Git:**

* **Source code management**
* **SCM tools**
* **What is Git**
* **Installation**
* **Work space/ work dir/ work tree**
* **Configuration**
* **Commit/check-in**
* **Version/version-ID/commit-ID,**
* **Work space, staging area, buffer area**
* **Local Repo/remote repo**
* **Customer Branching**
* **Release Branching**
* **Merge, stash, fast-forwarding, rebase**
* **Chery-pick, cat-file and hocks**
* **Repositories and Tracking**
* **Git commands: add, commit, log, push, status, ignore, branch, checkout merge, confict, stash, reset, revert**
* **Git GUI**

**Automate Build Process:**

**Maven:**

* **Introduction, Maven Structure**
* **Maven Dependencies**
* **Maven Repositories**
* **Maven Plugins and goals**
* **Integrated Maven Build**
* **Maven project**

**Jenkins:**

**What is Jenkins?**

**Installation and configuration**

* **Prerequisites**
* **Download and installation**
* **Configuration.**

**Managing Jenkins**

* **Securing Jenkins, Managing Credentials, Plugin Management**
* **Jenkins Backup, Create a Build Slave**

**Creating Application Builds**

* **Manual compilation with Maven**
* **Manually Testing, Packaging and Running the App**
* **Creating a Jenkins Job and configuring a Git Repo**
* **Compiling in Jenkins**
* **App Packaging in Jenkins**
* **Archiving artifacts**
* **Cleaning up Past Builds**
* **Build time trend**
* **The jenkins Dashboard**
* **Troubleshooting build failures**
* **Build linking upstream and downstream**

**Plugins**

* **Introduction**
* **Plugin Architecture**
* **Extension Points**
* **Getting Plugins**
* **Useful Plugins Overview**
* **Build Tool Plugins**
* **Installing a plugin**
* **Plugin configuration**
* **Security Overview**

**Continuous Testing and Continuous Integration and Testing**

* **Adding steps to Freestyle Project**
* **Creating a Pipeline job to execute Maven**
* **Archiving in a Pipeline**
* **Checking out git repository in pipeline**
* **The Master Agent Model**
* **Allocating a node and workspace in Pipeline**
* **Triggering Automated Builds**
* **Configuring an Email Server**
* **Notifications when a build fails**
* **Executing unit tests**

**Finding and Managing Plugins**

* **The need for plugins**
* **Integrated Code Coverage**
* **Assessing a plugin**
* **Testing Plugins and Plugin Types**

**Building Continuous Delivery Pipeline**

* **Continuous Delivery**
* **Backup and Restore**
* **A Second Node Allocation**
* **Adding an Agent Node**
* **Setup parallel integration testing in a pipeline**
* **Executing and Monitoring Parallel pipelines**
* **Setup Deployment to staging**
* **Executing a Deployment pipeline**
* **Checkin pipeline script to Git**

**Master and slave configuration**

**Integrating Jenkins with jfrog and sonarqube**

**Model 2 :Configuration / Provisioning**

**CHEF:**

**Introduction to Chef**

* **What is Chef**
* **Common Chef Terminology**
* **Chef Serverand Workstation**
* **Chef-Client**
* **Server and Nodes**
* **Chef Configuration Concepts**
* **Run\_list**

**Setting up the Environment**

* **Intro to ChefDK**
* **Chef Workstation Setup**

**Chef Server**

* **Installing Chef Server**
* **Chef-Repo, Setting Up the Work Station, and Bootstrapping A Node**
* **Configuring Git**
* **Chef Client, Nodes and Run Lists**
* **Building A Quick Apache Cookbook**
* **Managing Node Run\_Lists**
* **Chef-Client Configuration**

**Resources**

* **Understanding Chef and Chef Convergence**
* **Common Chef Resources**
* **Default Resource Actions**
* **Working with not\_if and only\_if Guards**

**Recipes and Cookbooks**

* **Understanding Chef Recipes and Run Lists**
* **Understanding Chef Cookbooks**
* **Generating a Cookbook**
* **Cookbook Pro-Tips**

**Local Cookbook Development Basics**

* **Test Kitchen Configuration**
* **Using Test Kitchen**
* **Static Code Analysis**
* **Troubleshooting**

**Building Web & application server Cookbook**

* **Getting Setup**
* **Starting the Apache Recipe**
* **Adding Platform Support to the Cookbook**
* **Adding Local Chef-Repo to Github**
* **Install and configure Chef Reporting**
* **Node Object and Search**
* **What is Node Object**
* **Writing cookbook &recipes for webserver**
* **Writing cookbook & recipes for tomcat application server**

**Attributes:**

* **Node Attributes**
* **Type of Attributes**
* **Attributes presidency**

**Environments**

* **Chef Environments**
* **What are Environments and why do they matter**
* **Creating and Configuring Environments**
* **Deploying to Different Environments**
* **Viewing and Environments with Knife**

**Roles**

* **What are Roles**
* **Creating A web server Role**
* **Creating Db Server Role**
* **Creating a Base Role**

**Chef Supermarket**

* **Chef Supermarket**

**Vagrant:**

**Introduction**

* **What is Vagrant**
* **Virtualization Overview**

**Setting Up Vagrant**

* **Virtual Box Installation**
* **Vagrant Installation**

**Using Vagrant**

* **Vagrant Machine**
* **Vagrant Files**
* **Boxes**
* **Running Vagrant Machines**
* **SSH to Vagrant Machine**
* **Synced Folders**
* **Environment Management**

**Automated Provisioning**

* **Provisioning**
* **Installing Apache /Shell Script**
* **Installing Apache /Chef**

**Networking**

* **Private Networking**
* **Public Networking**

**Model 3 : Configuration / Provisioning**

**Ansible:**

**Introduction**

* **What is Ansible**
* **Change Management**
* **Provisioning**
* **Automation**
* **Orchestration**
* **Why use Ansible**
* **YAML**
* **Built in Security**
* **Extendable**
* **Conclusion**

**Architecture and Process Flow**

* **Architecture Introduction**
* **System Requirement**
* **Components Overview**
* **Process of Execution**
* **Conclusion**

**Setup and Configuration**

* **Test Environment Setup**
* **Download and Installation**
* **Ansible Configuration File**
* **The HOSTS File**
* **Overriding the Default HOSTS File**
* **Overriding the Default System Ansible.Cfg File**
* **Overriding the Default Roles Path**
* **Understanding the core components of Ansible**
* **Ad-hoc commands in Ansible**

**Ansible Inventory and Configuration**

* **Introduction to Inventory &Configuration**
* **Inventory Fundamentals**
* **Scaling out with Multiple Files**
* **Ansible Configuration Basics**

**Ansible Playbooks**

* **Ansible Command Line**
* **Our First Playbook**
* **Variables: Inclusion Types**
* **Target Section**
* **Variable Section**
* **Task Section**
* **Handler Section**
* **Outlining Playbook**
* **Create a Playbook from Outline**
* **Optimizing Playbook**
* **Taking Playbook for a Dry Run**
* **Simple Variable Substitution**
* **Loops**
* **Conditionals**
* **Vault**
* **Basic Include Statements**
* **Tags**
* **Basic Error Handling**
* **Includes - Breaking Your Playbook Into Discrete Plays**
* **Starting At Task or Stepping Through All Tasks**
* **Passing Variables Into Playbooks at the Command Line**
* **Using Jinja2 Templates**
* **Use a playbook to copy a program and customize it for the target host.**
* **Wiring Play books for java, webservers and applications server**

**Ansible Modules**

* **Introduction**
* **Ansible Modules Fundamentals**
* **Module Docs**
* **Yum Module**
* **Setup Module**
* **Other modules**

**Create and use templates to create customized configuration files**

* **Introduction**
* **Templates**

**Working with Ansible facts and variables.**

* **Let see how we get ansible facts and how we use facts**
* **Using Ansible facts**
* **Using variables to gather server info**

**Roles**

* **Introduction to Roles**
* **Role Basics**
* **Creating Role**
* **Ansible Galaxy**
* **Roles - The Directory Structure**
* **Role Based Tasks**
* **Task Order - Pre and Post Tasks**
* **Roles - Conditional Execution**
* **Roles - Variable Substitution**
* **Roles - Handlers**
* **Roles - Configuring Alternate Roles Paths**
* **Roles - Conditional Include Statements**
* **Roles - Waiting For Events**
* **Roles - Executing a Task Until**
* **Roles - Using Tags**
* **Roles - Breaking a Playbook Into a Role**
* **Roles - Passing Variables from Command Line**
* **Roles - Using Jinja2 Templates**
* **Roles - LocalAction**
* **Roles - Lets create a role to install apache.**
* **Lets use the previous role and add a new one.**
* **Lets build on the previous role**
* **Create Role for installing tomcat on multiple nodes**
* **Create Role for deployapplication on multiple nodes**

**Model 4:Containerization**

**Docker:**

**Introduction**

**What is docker**

**Docker life cycle**

**How docker containers working**

**Installing Docker**

* **Installing Docker on Linux**

**Working with Containers**

* **How to create the containers**
* **Docker commands**
* **Theory of pulling and Running Containers**
* **Working with images**
* **Container Life cycle**
* **Dockerfile**
* **Build Docker images**
* **Docker push**
* **Docker compose**
* **Docker volumes**
* **Docker cpu**
* **Docker memory**
* **Docker Networking**

**Kubernetes**

* **What is kubernets**
* **Purpose of Kubernets**
* **How kubernets works**
* **Master components, how works**
* **Node Compenents, how works**
* **How pods works**
* **Installations and configuration kubernets cluster**
* **Pod lifecycle**
* **Work with pods**
* **Work with Services**
* **Work with Replica controller**
* **Work with Deployment**
* **How to Releases java application by using RC and Deploy**
* **KOPS**

**Model 5: Terraform:**

**Infrastructure Automation**

* **What is Infrastructure as Code and why is it needed?**
* **Declarative vs Procedural tools for Infrastructure as Code**
* **Infrastructure as Code in the Cloud**
* **Requirements for infrastructure provisioner**
* **Deploying First Server**
* **Preparing work environment**
* **Terraform providers**
* **Configuring AWS provider e. Creating EC2 instance with Terraform**
* **Working with state**
* **Handling resource updates**

**Resource Dependencies and modules**

* **Creating AWS Virtual Private Cloud**
* **Controlling dependencies with depends\_on and ignore\_changes**
* **Making sense of our template**
* **Removing duplication with modules**
* **Configuring modules**
* **Retrieving module data with outputs**
* **Using root module outputs**

**Storing and Supplying Configuration**

* **Understanding variables**
* **Configuring data sources**
* **Exploring Terraform configuration resources**
* **Taking a quick look at Consul**

**Model 6: Monitoring**

**System Monitoring :Nagios**

* **Installation of Nagios**
* **Configuring Nagios**
* **Monitoring with Nagios**

**Application Monitoring: ELK**

* **Installation of Elasticsearch, logstash, Kiban stack**
* **Configuring the ELK Stack**
* **Monitoring logs with ELK**