

# DevOps



CHEF™



Jenkins



HashiCorp

Packer



Terraform



ANSIBLE



Visual Studio Team Services



docker

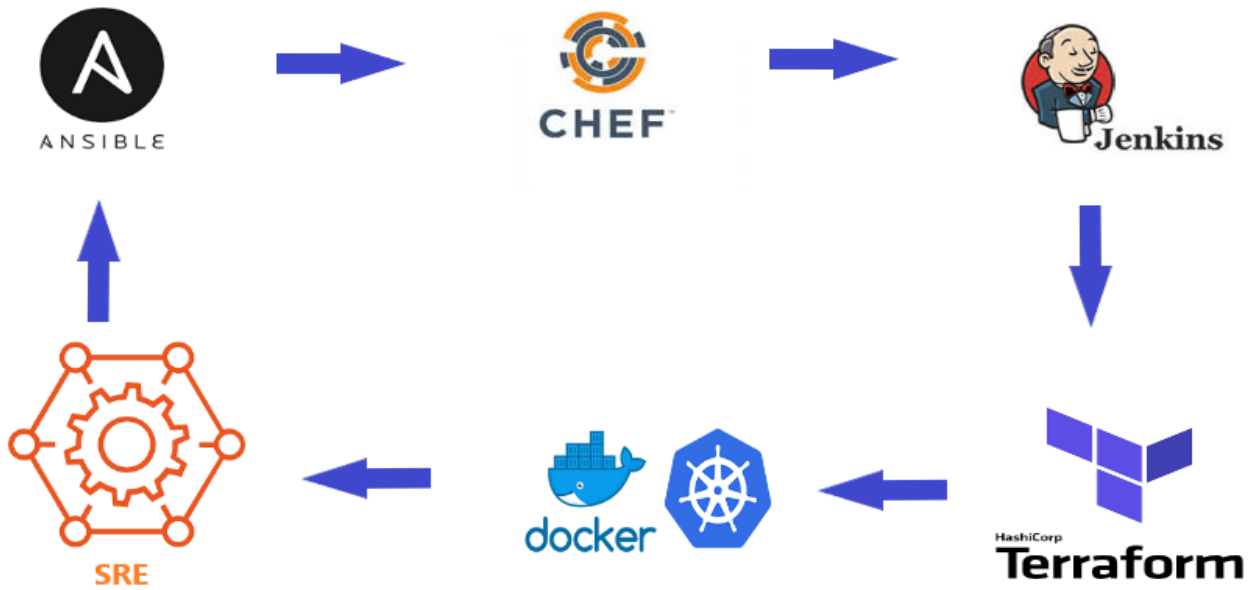


kubernetes

We are ready to serve Latest IT Trends, Are you ready to learn.??

Hyderabad

START DATE :  
TIMINGS :  
DURATION :  
TYPE OF BATCH :  
FEE :  
FACULTY NAME : Khaja



**DevOps – Big Picture**

- 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
- What is Agile and Scrum
  - Agile Development Process
  - Role of Dev in Agile
  - Role of Ops in Agile
- Problem That DevOps Solves
- Making a DevOps Transition

**Ansible**

- **System Architecture and Design of Ansible**
  - a. Installation and Configuration
  - b. Core Concepts of Ansible
    - i. Inventory
    - ii. Module
    - iii. Adhoc Command
    - iv. Playbooks
    - v. YAML
  - c. Inventory and Playbook Parsing
  - d. Module transport and Execution
  - e. Variable Types
  - f. Variable Precedence
  - g. External data access
- **Ansible Essentials**
  - a. Static Inventories
  - b. Dynamic Inventories
  - c. Common Modules
  - d. Playbook syntax
  - e. Conditionals
  - f. Error Handling
  - g. Variables and Facts
  - h. Templates
  - i. Roles and Ansible Galaxy
  - j. Parallelism
- **Protecting Secrets with Ansible**
  - a. Encrypting data at rest
  - b. Mixing Encrypting with plain YAML
- **Ansible and Windows**
  - a. Running Ansible from Windows
  - b. Setting up windows hosts with Ansible
  - c. Handling Windows Authentication and Encryption
  - d. Automating Windows tasks with Ansible
- **Jinja2 Templating**
  - a. Jinja2 Templating basics
  - b. Control Structures
  - c. Data manipulation
  - d. Comparing Values
- **Controlling Task Conditions**
  - a. Defining a failure
  - b. Defining a change

- c. Error recovery
- d. Iterative tasks with loops
- **Reusable Ansible Content with Roles**
  - a. Task, handler, variable and playbook inclusion concepts
  - b. Roles
- **Troubleshooting Ansible**
  - a. Playbook logging and verbosity
  - b. Variable introspection
  - c. Debugging code execution
- **Minimizing Downtime with Rolling deployments**
  - a. In-place upgrades
  - b. Expanding and contracting
  - c. Failing fast
  - d. Minimizing disruptions
  - e. Serializing single tasks

## Chef

- **System Architecture**
  - a. Chef Architecture
  - b. Scripts vs IaC (Procedural vs Declarative Approaches)
  - c. Pull Model Centralized Configuration Management
- **Chef Terminology**
  - a. Infrastructure as code
  - b. Desired State Configuration
  - c. Idempotence and Convergence
  - d. Configuration Drift
- **Chef Tools**
  - a. Chef Server
  - b. ChefDK
  - c. Knife
  - d. Test Kitchen
  - e. Supermarket
  - f. Foodcritic
  - g. Inspec
- **Core Components of Chef**
  - a. Cookbooks
  - b. Recipes
  - c. Resources
  - d. Nodes
  - e. Runlists
  - f. Roles
  - g. Environments
  - h. Attributes
  - i. Databags

- **Chef Workflow**
- **Cookbook development**
  - a. Generators
  - b. Test Driven Development
  - c. Chef Spec
  - d. Test Kitchen Configuration
- **Data Driven Cookbooks**
  - a. Node Objects, Attributes and Ohai
  - b. Default Attributes
  - c. Attribute Precedence
- **Customizing Cookbooks**
  - a. Customization Strategies
  - b. Creating a Wrapper Cookbook
- **Multi Environment and Multi node Deployment**
  - a. Using Roles
  - b. Using Environments
  - c. Using Databags

## CI/CD Pipelines with Jenkins, VSTS and AWS Code Pipeline

### GIT

- Version Control Basics
- Commits and Revisions
- Branches
- Stashing
- Branching In Depth
- Rebase
- Tagging
- Sub-Projects with Sub-Modules and SubTrees
- Git Hooks
- Git Administration
- Git Flow

### CI/CD

- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Importance of CI/CD Engines in Building DevOps Pipelines

### Jenkins

- Key Constructs of Jenkins
  - a. Job
  - b. Build
  - c. Version Control System
  - d. Test Executions

- e. Plugins
- f. CLI
- g. Rest API
- h. Security
- i. Distributed Builds
- Jenkins Internals
  - a. Jenkins execution under the hood
  - b. Importance of Environment Variables
  - c. Why Jenkins is called as Cron on Steroids
- Jenkins Installation
- Jenkins Distributed Build Setup (Multi node configuration)
- Jenkins Administration
- Jenkins Views and Free Style Projects
- Parametrization and Up/DownStream Projects
- Jenkins Pipelines , Groovy and Jenkins DSL
- Jenkins Integrations
  - a. Git
  - b. Ansible
  - c. Docker
  - d. Kubernetes
  - e. Chef
  - f. Terraform
  - g. JIRA
  - h. Python
- Multi Branch Jenkins Pipelines
- Jenkins Agents
  - a. Tool Installations on Agents
  - b. Cloud Agents
  - c. High Availability



### VSTS (Azure DevOps)

- What is Azure DevOps
- Source Code Management
  - a. Azure Repos
  - b. Using Git Hooks with Azure DevOps Server
- Build and Release Agents
- Continuous Integration and Build Automation
- Continuous Testing
- Continuous Deployments
- Azure Artifacts and Dependency Management
- Azure Pipelines

### AWS Code Pipeline

- Code Pipeline

- Code Build
- Code Deploy
- Creating a simple pipeline using AWS Code Commit, Code Build and Code Deploy

## Infrastructure Provisioning

### Packer

- What is Packer
- Why Use Packer
- Installing Packer
- Packer Constructs
  - a. Artifacts
  - b. Builds
  - c. Builders
  - d. Commands
  - e. Post-Processor
  - f. Provisioners
  - g. Templates
- Packer CLI
- Creating AWS AMI using Packer
- Creating Azure VM Image using Packer
- Creating Vagrant Box using Packer
- Provisioning using Ansible and Chef

### Terraform

- Infrastructure Provisioning
  - a. What is Infrastructure as Code
  - b. Infrastructure as Code in the Cloud
  - c. How Terraform Does Infra Provisioning
- Installation
- Terraform Constructs
  - a. Terraform DSL
  - b. Providers
  - c. Resource
  - d. Arguments
  - e. Attributes
  - f. Variables
  - g. Maps and Lookups
  - h. Modules
  - i. Local State
  - j. Remote State
  - k. Taint and Update Resources
- Terraform DSL
  - a. Declaring Variables

- b. Working with Resources
  - c. Nested Blocks
  - d. Dynamic Nested Blocks
  - e. Expressions and functions
- Resources and Providers
  - a. Null Resource
  - b. Local Exec
  - c. AWS Provider and Resources
  - d. Azure Provider and Resources
  - e. Docker Provider and Resources
  - f. Kubernetes Provider and Resources
- Terraform Registry
- Terraform Remote State and Workspace
- Terraform Trouble Shooting
- Using Terraform to create a AWS Cloud Deployment
- Using Terraform to create Azure Cloud Deployment

**Docker**

- Docker Overview
  - a. Docker Overview
  - b. Understanding Docker
  - c. Difference between Physical Servers, Virtual Machines and Docker
  - d. Docker Installation
  - e. Docker CLI Overview
  - f. Docker and container
- Building Container Images
  - a. Dockerfile
  - b. Dockerfile instructions
  - c. Multi stage Docker build
- Storing and Distributing Images
  - a. Docker Hub
  - b. Docker Store
  - c. Docker Registry
  - d. Docker Trusted Registry
  - e. Azure Container Registry
  - f. Amazon ECR
- Managing Containers
  - a. Docker container Commands
  - b. Docker Network and Volumes
- Docker Networking
- Docker Volumes (Storage)
- Docker Compose
  - a. Installation
  - b. Docker Compose Yaml file



- c. Docker Compose Commands
  - d. Docker App
- Windows Containers
  - a. Introduction to Windows Containers
  - b. Setting up Docker host for Windows Containers
  - c. Running Windows Containers
  - d. Windows Dockerfile
  - e. Windows containers & Docker compose
- Docker Swarm and Services
  - a. Introduction
  - b. Roles within a Docker Swarm
  - c. Creating and managing a Swarm
  - d. Managing a cluster
  - e. Docker Swarm services & stacks
  - f. Load balancing, Overlays and scheduling
- Docker Security
  - a. Container Considerations
  - b. Best Practices
  - c. Third Party Security Services
- Docker Workflows
  - a. Docker for development
  - b. Monitoring
  - c. Extending to external Platforms
- Running Docker in Public Clouds
  - a. Amazon ECS and Fargate
  - b. Microsoft Azure App Services
  - c. Docker Cloud
- Docker Enterprise Edition
  - a. Installation
  - b. Universal Control Plane(UCP)
  - c. Docker Trusted Registry (DTR)
  - d. UCP Security
  - e. Backups for UCP & DTR
  - f. Certificate Management

**Kubernetes**

- Overview
- Introduction to Microservices
- Clustering and Orchestration
- Kubernetes Architecture
- Kubernetes Core Concepts
  - Pods
  - Namespaces
  - API primitives

- Kubernetes runtime
- Health checks
- Application Scheduling
- Kubernetes Networking
- Service Discovery
- DNS
- Multitenancy
- Kubernetes Namespaces
- Kubernetes Storage Overview
- Persistent Storage & Stateful sets
- Monitoring, Logging & Troubleshooting
- Creating Kubernetes Clusters
- Cluster Authentication, Authorization & Container Security
- Running Stateful Applications with Kubernetes
- Rolling Updates, Scalability & Quotas
- Kubernetes Package management with Helm
- Understanding & Using Helm
- Creating Helm Charts
- Native Kubernetes on Amazon Cloud using Elastic Kubernetes Services (EKS)
- Native Kubernetes on Azure using Azure Kubernetes Services (AKS)

**OpenShift**

- Overview
- Understanding OpenShift's business value
- OpenShift flavors
- Installing OpenShift
- Managing Persistent Storage
- OpenShift Concepts
  - Projects
  - Users
  - Applications
  - Pods
  - Services
  - Routes
  - ImageStreams
  - ConfigMaps
  - ResourceQuotas
  - Templates
  - Autoscaling
- OpenShift Networking
- CI/CD Pipelines in OpenShift
- OpenShift HA Architecture

**Site Reliability Engineering**

- How SRE Relates to DevOps

- Background on DevOps
- Background on SRE
- Introduction to SRE
- Monitoring
  - Why Monitoring
  - Instrumenting an application
  - What should be measured
  - Collecting and saving monitoring Data
  - Displaying monitoring information
- Monitoring with Nagios (polling application)
- Monitoring with Elastic Stack (push application)
- Incident Response
  - What is an incident
  - Alerting
- Postmortems
  - What is postmortem
  - Why & when to write a postmortem document
  - Carrying out incident analysis
  - How to write postmortem document
  - Analyzing past postmortems
    - MTR and MTBF
    - Alert fatigue
- Testing and Releasing
  - Testing
  - Releasing
  - Automation
    - Continuous Everything
- Canarying Release
  - Release Engineering Principles
  - What is Canarying
  - A Roll Forward Deployment vs Simple Canary Deployment
  - Canary Implementation
  - Selecting and Evaluating Metrics
  - Dependencies & Isolation
  - Requirements on Monitoring Data
  - Evaluation

**Foundation Course for DevOps, AWS & AZURE****Linux**

- Overview
- Understanding Linux Architecture
- Shell and Kernel Overview
- Linux Distributions
- Using Shell
- Exploring Filesystems
- Working with Text Files
- Process Management
- Package Management
  - RPM
  - DEB
  - YUM
  - APT
  - SNAP
- Managing User Accounts
- Disk & Filesystem management
  - Disk Storage
  - Partitions
  - LVM
  - Mounts
- Linux Networking
- Service Management in Linux
  - Init
  - systemd
- Server Configurations in Linux **Hyderabad**
  - Web Server
  - Application Server
  - Syslog
  - Database Servers
- Troubleshooting in Linux

**Shell Scripting**

- Why and What of Shell Scripting
- Shell Terminals
- Creation & Execution of Shell Scripts
- Variables & Variable Scopes
- Conditions in Shell Scripts
- Iterating with loops
- Functions in Shell Scripts
- Regular Expressions
- Command Piping with grep
- Stream Editor

- Understanding basics of sed
- Sed commands
- AWK Fundamentals

**Python**

- Introduction
  - Why Python?
  - Installing Python
  - Python 2 vs Python 3
- Types in Python
  - Integers & Floats
  - String
  - Booleans
  - None
  - Lists
  - Dictionary
  - Other Data Types
- Statements in Python
  - If
  - Loops
  - Break & Continue
  - While
- Exceptions in Python
- Functions
- File Management in Python
- Yield
- Lambda Functions
- Object Oriented Programming with Python
  - Classes
  - Methods
  - Constructors
  - Instance & Class Attributes
  - Inheritance & Polymorphism
- Python Tips & Tricks
- Strings & Collections
- Modularity
- Handling Exceptions

**Networking**

- Basic Networking Concepts
  - Computer Network
  - Terminology
  - Network Protocol
  - Ping & Traceroute
  - What is IP address

- Network Categories and Components
- Domain Naming System
- OSI Model
  - Layers
  - Application Layer
  - Presentation Layer
  - Session Layer
  - Transport & Other lower layers
  - TCP vs OSI Model
- Binary Compute Basics
- Hexadecimal Compute Basics
- IP Addressing
  - Overview & Demonstration
  - IPV4 Address Format
  - Network vs Host portion
  - Class A, B,C,D,E address
  - Classless Inter-Domain Routing (CIDR) Notation
- IP Subnetting
- Routing
- Switching
- NAT Server
- DNS
- DHCP Server

**Windows Server**

- Setup
- Understanding the Client Server Architecture
- Server Manager
- Managing Local User Accounts
- Task Manager
- Windows Administrative Tools
- Active Directory
- DHCP
- DNS and Name Resolution
- IIS Services and Configuration
- Active Directory Groups & OU
- Group Policy Management
- Windows Server Backups Overview

**PowerShell**

- Introduction to PowerShell
  - What is PowerShell
  - PowerShell Editors
  - Getting Help
  - Command Naming & Discovery

- Parameters & Parameter Sets
- Introduction to Providers
- Modules & Snap-ins
  - Introducing Modules
  - PowerShell Core & the Windows Compatibility Module
  - Snap-ins
- Objects in PowerShell
  - Pipelines
  - Members
  - Enumerating & Filtering
  - Selecting & Sorting
  - Grouping & Measuring
  - Comparing
- Operators
  - Arithmetic Operators
  - Assignment Operators
  - Comparison Operators
  - Regular Expression based Operators
  - Binary Operators
  - Logical Operators
  - Type Operators
  - Other Operators
- Variables, Arrays and Hashtables
- Branching & Looping
- Strings, Numbers & Dates
- Files, Folders & Registry
- Web Requests & Web Services
- Remoting & Remote Management
- Scripts, Functions & Filters
- Parameters, Validation & Dynamic Parameters
- Testing, Troubleshooting & Error Handling

**PowerShell DSC**

- Introduction & Overview of PowerShell DSC
- DSC Architecture
- DSC Configuration Files
- DSC Resources
- Pushing DSC Configurations
- DSC Cross Platform Support

**Others**

- Kafka Configuration
- Vagrant
- Virtualization
- Groovy Scripting