QualityThought®

# Devops

**Mode of Training:** Online, Classroom, Corporate

**Faculty:** Mr. KHAJA

**Course Duration:** 84 Days

# Course Curriculum

# Foundation

## 1. DevOps - The Big Picture

- **Why DevOps**
  - **a.** Business Perspective
  - **b.** IT Perspective
  - **c.** Developer Perspective
  - **d.** Tester Perspective
  - **e.** Operations Perspective
- **What is DevOps**
  - **a.** Definition
  - **b.** Stakeholders of DevOps
- **What is SDLC**
  - **a.** Phases of SDLC
  - **b.** Role Of Dev in SDLC
  - **c.** Role of Ops in SDLC
- **What is Agile and Scrum**
  - **a.** Agile Development Process
  - **b.** Role of Dev in Agile
  - **c.** Role of Ops in Agile
- **Problem That DevOps Solves**
- **Making a DevOps Transition**
- **Introduction to DevOps Automation**
  - Introduction
  - Tools

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

## 2. Linux Essentials

- **Working at the Command Line**
  - Linux commands
  - Bash scripting
- **Reading Files**
- **Piping and Redirection**
- **Accessing Root Accounts**
- **Accessing Servers with ssh**

## 3. Cloud Computing

- **Introduction**
- **Software As A Service**
  - Software As A Service (SAAS)
  - Single Tenant Applications

- Saas and Users
- Saas and Software Vendors
- The impact of Saas

- **Cloud Platforms**
  - What is Cloud Platform
  - Infrastructure As A Service (IAAS)
  - Platform As A Service (PAAS)
  - IAAS vs. PAAS
- Public vs. Private Cloud

### 4. Virtualization

- Introduction
- What is Virtualization
- Server Virtualization
- Desktop Virtualization
- Application Virtualization

## DevOps Tools

### 1) Source Code Management

- **What is Version Control System**
- **Git:**
  - **a.** Installation
  - **b.** Configuration
  - **c.** Basic Commands
  - **d.** Branches

**e.** Repositories

**f.** Tracking

## 2) Build

- **Automated Build Process**
- **Maven**

a. Introduction

**b.** Maven Structure

**c.** Maven Dependencies

**d.** Maven Repositories

**e.** Maven Plugins

**f.** Integrated Maven Build

- **MS Build**

**a.** Essentials

**b.** Execution Life Cycle

**c.** Custom Tasks

**d.** Common Scenarios

## 3) Continuous-Integration

- **Jenkins**

**1 What is Jenkins**

**2 Best Practices**

**3 Installation and configuration**

- Prerequisites
- Download & installation
- Configuration tour

## 4 Managing Jenkins

- Securing Jenkins
- Managing Credentials
- Plugin Management
- Jenkins Backup
- Create a Build Slave

## 5 Creating Application Builds

- Anatomy of the build
- Cloning sample project
- Manual compilation with Maven
- Manually Testing, Packaging and Running the App
- Creating a Jenkins Job and configuring a Git Repo
- Compiling in Jenkins
- Browsing the workspace in Jenkins
- App Packaging in Jenkins
- Archiving artifacts
- Cleaning up Past Builds
- Build time trend
- The jenkins Dashboard
- Troubleshooting build failures
- Importing Job config.xml
- Anatomy of the job
- Build linking upstream and downstream

## 6 Plugins

- Introduction

- Plugin Architecture
- Extension Points
- Getting Plugins
- Plugin Wiki
- Useful Plugins Overview
- Source Code Plugins
- Trigger Plugins
- Build Tool Plugins
- Wrapper Plugins
- Notifier Plugins
- Reporting Plugins
- Artifact & UI Plugins
- Installing a plugin
- Plugin configuration
- Security Overview

**7. 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
- High level progress with Pipeline stages
- Triggering Automated Builds
- Configuring an Email Server

- Notifications when a build fails
- Duplicating a job
- Executing unit tests
- Executing selenium tests
- Visualizing Test Results

**8. Finding and Managing Plugins**

- The need for plugins
- Integrated Code Coverage
- Assessing a plugin
- Installing the HTML Publisher plugin
- Publishing HTML Reports
- Testing Plugins and Plugin Types
- Blue Ocean UI Plugin

**9. Building Continuous Delivery Pipeline**

- Continuous Delivery
- Backup and Restore
- Starting point and Pipeline stashing
- Browsing Workspaces in Pipeline Jobs
- A Second Node Allocation
- Adding an Agent Node
- Setup parallel integration testing in a pipeline
- Executing and Monitoring Parallel pipelines
- Manual Approval for Deployments
- Setup Deployment to staging
- Executing a Deployment pipeline

- Check in pipeline script to Git

## 4) Configuration / Provisioning


### 1. Introduction of Chef:

- What is Chef
- Common Chef Terminology
- Chef Server
- Chef Workstation
- Chef-Repo
- Chef-Client
- Server and Nodes
- Chef Configuration Concepts

### 2. Setting up the Environment

- Intro to chef DK
- Chef workstation setup

### 3. Chef Server

- Installing Chef Server
- Chef-Repo, Setting Up the Work Station, and Bootstrapping A Node
- Configuring Git
- Chef Solo vs Chef Zero vs Chef Server

- Chef Client, Nodes and Run Lists
- Building A Quick Apache Cookbook
- Managing Node Run_Lists
- Chef-Client Configuration

**4. Resources**

- Understanding Chef and Chef Convergence
- Common Chef Resources
- Default Resource Actions
- Applying Chef Resources Hands On
- Working with not_if and only_if Guards
- Extending Chef with Custom Resources

**5. Recipes and Cookbooks**

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

6. **Local Cookbook Development Basics**

- Generators
- Test Driven Development
- ChefSpec
- Test Kitchen Configuration
- Using Test Kitchen
- InSpec
- Static Code Analysis

- Troubleshooting

**7. Cookbook Components**

- Cookbook Structure
- Metadata Anatomy
- Versioning
- Attributes
- Common Resources
- Templates
- Libraries
- Custom Resources

8. **Design Patterns and Theory**

- Cookbook Disposition
- Wrapper Cookbooks
- Community Cookbooks
- Managing Cookbook Dependencies
- Data Bags
- Vault
- Search

**9. Nodes and Search**

- Node Object
- Working with Ohai and Node Attributes
- Understanding Search
- Knife Search

## 10. Roles and Environments

- Setting Up A New Node
- Understanding Roles
- Creating Roles
- Understanding Environments
- Bootstrapping The Staging Node
- Creating And Using Environments

## 11. Desired State Configuration

- Imperative Vs Declarative Approach To Configuration Management
- Pull vs Push Approach
- Windows DSC
- Removing Resources From Recipes

## 12. Chef Supermarket

- Chef Supermarket
- Using A Private Supermarket

## 13. Building Web 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

## 14. Chef Offerings

- Open Source vs. Premium
- Habitat
- Chef Compliance and InSpec
- Chef Automate: Overview
- Chef Automate: Workflow

## 15. Deploying Nodes In Production

- Unattended Node Bootstrapping


- **Introduction**

a. Introduction to puppet

b. Puppet Head First

c. Puppet Enterprise Stack And Core Concepts

d. Nodes

- **Puppet Components**

**a.** Installing the puppet master & learning puppet master layout

**b.** Installing the puppet Enterprise layout

**c.** Puppet.conf

**d.** Resource Abstraction Layer

**e.** Facter

**f.** Live Management

**g.** Catalog Compilation

- **Node Classification**

  **a.** Site.pp and Node definition matching

  **b.** External Node Classifiers

  **c.** Classifying the Node with the Console

  **d.** Using Site.pp with ENC

- **Puppet Management**

  a. Common Console Tasks

  b. Troubleshooting

  c. Reporting

  d. Preparing Modules for puppet forge & downloading forge modules

  e. Deactivating a puppet Enterprise node

  f. Event Inspecto

  g. External fact

  h. Checking Values of Setting

  i. Puppet Resource Command

- **Resource Type Practice**

  a. Host

  b. Resource Type Titles

  c. Mount

HashiCorp Packer

- **Getting Started**
  a. What Is Immutable Infrastructure?
  b. Two Ways to Create Immutable Infrastructure
  c. Packer Basics
- **Creating Base Images with Packer**
  a. Provisioners - Configuration Management
  b. Provisioners - File
  c. Provisioners - Script
  d. User Variables and Built-in Functions
- **Expanding Base Images**
  a. Multi-provider Builds
  b. Post-processors: Vagrant and Compress
  c. Post-processors: Atlas

HashiCorp Terraform

- **Infrastructure Automation**
  a. What is Infrastructure as Code and why is it needed?
  b. Declarative vs Procedural tools for Infrastructure as Code

c. Infrastructure as Code in the Cloud

d. Requirements for infrastructure provisioner

- **Deploying First Server**

a. History of Terraform

b. Preparing work environment

c. The many Terraform providers

d. Configuring AWS provider

e. Creating EC2 instance with Terraform

f. Working with state

g. Handling resource updates

- **Resource Dependencies and modules**

a. Creating AWS Virtual Private Cloud

b. Understanding dependency graph

c. Playing with Terraform graph

d. Controlling dependencies with depends on and ignore changes

e. Making sense of our template

f. Removing duplication with modules

g. Configuring modules

h. Retrieving module data with outputs

i. Using root module outputs

- **Storing and Supplying Configuration**

a. Understanding variables

b. Configuring data sources

c. Exploring Terraform configuration resources

d. Taking a quick look at Consul

- **Connecting With Other tools**
  a. Returning data with outputs
  b. Testing servers with Inspect
  c. Provisioners
  d. Reprovisioning machines with null resource
  e. Using third-party plugins
- **Scaling and Updating infrastructure**
  a. Counting servers
  b. Bringing in high availability
  c. Load balancing and simulating conditionals
  d. Immutable Infrastructure
  e. Baking images with Packer
  f. Rolling out AMI upgrades with Terraform
  g. Performing blue-green deployments
  h. Refreshing infrastructure
  i. Importing resources
- **Collaborative Infrastructure**
  a. Version control with Git
  b. Moving templates to Git
  c. Protecting secrets in a Git repository
  d. Storing state files remotely
  e. Connecting remote states together
  f. Storing modules remotely
  g. Locking state files with Terragrunt

h. Moving infrastructure updates to the CI pipeline

i. Integration testing of Terraform modules


- **Introduction**
  - What is Vagrant
  - Virtualization Overview
- **Setting Up Vagrant**
  - Virtual Box Installation
  - Vagrant Installation
  - Vagrant without Virtual Box
- **Using Vagrant**
  - Vagrant Machine
  - Vagrant Files
  - Boxes
  - Running Vagrant Machines
  - SSH to Vagrant Machine
  - Synced Folders
  - Networking Basics
  - Environment Management
- **Automated Provisioning**
  - Provisioning

- Installing Apache /Shell Script
- Installing Apache /Chef
- Installing Apache/ Puppet

▪ **Networking**

- Private Networking
- Public Networking

▪ **Multiple Machines with Vagrant**

- Managing Multiple Machines
- Hands on multiple machine using vagrant

▪ **Boxes**

- Vagrant Boxes
- Basic Box Management
- Creating Boxes from existing environment
- Creating custom boxes


## 1. Introduction

- Introduction to Ansible
- Ansible vs. Other Tools
- Introduction to YAML
- Ansible Documentation: Modules

2. **Setup and Configuration**
    - Test Environment Setup
    - Download and Installation
    - Ansible Configuration File
    - Ansible Python Dependencies
    - 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

3. **Use both static and dynamic inventories to define groups of hosts**
    - Overview of static and dynamic inventories in Ansible
    - Static Inventories
    - Dynamic Inventories

4. **Ansible Playbooks**
    - Configuring Your 'Ansible' Account
    - Ansible Command Line
    - System Facts
    - System Facts: Common Values for Playbooks
    - 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
- Asychronous Polling
- Simple Variable Substitution
- Lookups
- RunOnce
- Local Actions
- Loops
- Conditionals
- Until
- Notify
- Vault
- Prompt - Interactive Playbook
- 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
- LocalAction

- DelegateTo
- Use a playbook to copy a program and customize it for the target host.

5. **Ansible Modules**

- Commonly used Modules
- Using modules in playbooks
- The 'Setup' Module
- The 'File' Module
- The 'Pause' Module
- The 'WaitFor' Module
- The 'Yum' Module
- The 'Apt' Module
- The 'Service' Module
- The 'Copy' Module
- The 'Command' Module
- The 'Cron' Module
- The 'Debug' Module
- The 'Fetch' Module
- The 'User' Module
- The 'AT' Module
- The 'DNF' Module
- The 'Apache2_Module' Module
- The 'SetFact' Module
- The 'Stat' Module
- The 'Script' Module

- The 'Shell' Module
- The 'SELinux' Module
- The 'SEBoolean' Module
- The 'Raw' Module
- The 'Ping' Module
- The 'Package' Module
- The 'Unarchive' Module
- The 'HTPasswd' Module
- The 'GetURL' Module
- The 'Group' Module
- The 'Mail' Module
- The 'Filesystem' Module
- The 'Mount' Module
- The 'Notify' Module
- The 'AptRepo' Module
- The ' AptKey' Module
- The ' ACL' Module
- The 'Git' Module
- Creating a Jinja2 Template File
- The 'Template' Module
- The 'MySQL_DB' Module
- The 'MySQL_User' Module
- The 'Kernel_Blacklist' Module

6. **Create and use templates to create customized configuration files**
    - Introduction
    - Templates
7. **Working with Ansible facts and variables.**
    - Let see how we get ansible facts and how we use facts.d
    - Using Ansible facts
    - Using variables to gather server info
8. **Roles**
    - Introduction to Roles
    - Roles - The Directory Structure
    - Role Based Tasks
    - Task Order - Pre and Post Tasks
    - Roles - Conditional Execution
    - Roles - Variable Substitution
    - Roles - Handlers
    - Roles - Using Notification
    - 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 - DelegateTo
    - 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 roles

**9. Download roles from Ansible Galaxy and use them**

- Ansible galaxy and how its used
- Lets use multiple roles

**10. Ansible Command Line Usage**

- Ansible Command Line - Installing Packages
- Ansible Command Line - Services and Hosts
- Ansible Command Line - Commands and Shells
- Ansible Command Line - Managing Users
- Ansible Command Line - Create and Manage Cron Jobs
- Ansible Command Line - Running Arbitrary Commands
- Ansible Command Line - Output Tree

**11. Managing Parallelism**

- What is parallelism?
- Parallelism in a playbook

**12. Using ansible-vault in playbooks to protect sensitive data**

- Let's discuss ansible-vault and see an example
- Options useable with ansible-vault

**13. Install ansible tower and use it to manage systems**

- Installing a trial version of ansible tower
- Log into our Ansible tower and run a sample task
- Let's add to the inventory and run a task against them

## 14. Usecases:

- Creating a Web Server Deployment - Outline
- Creating a Web Server Deployment - Playbook First Pass
- Creating a Web Server Deployment - Playbook Optimization
- Creating a Web Server Deployment - Breaking Into Role(s)
- Creating an NFS Server Deployment - Outline
- Creating an NFS Server Deployment - Playbook First Pass
- Creating an NFS Server Deployment - Playbook Optimization
- Creating an NFS Server Deployment - Breaking Into Role(s)
- Creating a Database Server Deployment - Outline
- Creating a Database Server Deployment - Playbook First Pass
- Creating a Database Server Deployment - Playbook Optimization
- Creating a Database Server Deployment - Breaking Into Role(s)

## 5. Containerization


- **Introduction**
- **Installing Docker**
  - Installing Docker on Windows
  - Installing Docker on Linux
- **Working with Containers**
  - What is container
  - Docker run command
  - Theory of pulling and Running Containers
  - Working with images
  - Container Life cycle
- **Creating & Managing Container Images**
- **Data Volumes & System Management**
- **Single Host Networking**
- **Multi Host Networking**

## Kubernetes

- Introduction
- Architecture & Generic Installation
- Pods, Services, Replication Controllers & Labels

- Networking, Load Balancers & Ingress
- Updates, Gradual Rollouts & Auto scaling
- 
- Deployments, Jobs & Daemon Sets
- Storage & Running Stateful Applications
- Monitoring & Logging
- Kubernetes on Azure

## 7. Monitoring

### Nagios

- **Installation of Nagios**
- **Configuring Nagios**
- **Monitoring with Nagios**
- **Triggering Alerts**

### Elastic Search + Logstash + Kibana

- **Installation of ELK stack**
- **Configuring the ELK Stack**
- **Monitoring logs with ELK**

### Cloud Platforms

- **AWS**
  - Environment setup in AWS
  - Cloud Deployment Scenarios in AWS
  - Continuous Delivery in AWS
- **Azure**
  - Environment setup in Azure

- Cloud Deployment Scenarios in Azure
- Continuous Delivery in Azure

## PowerShell

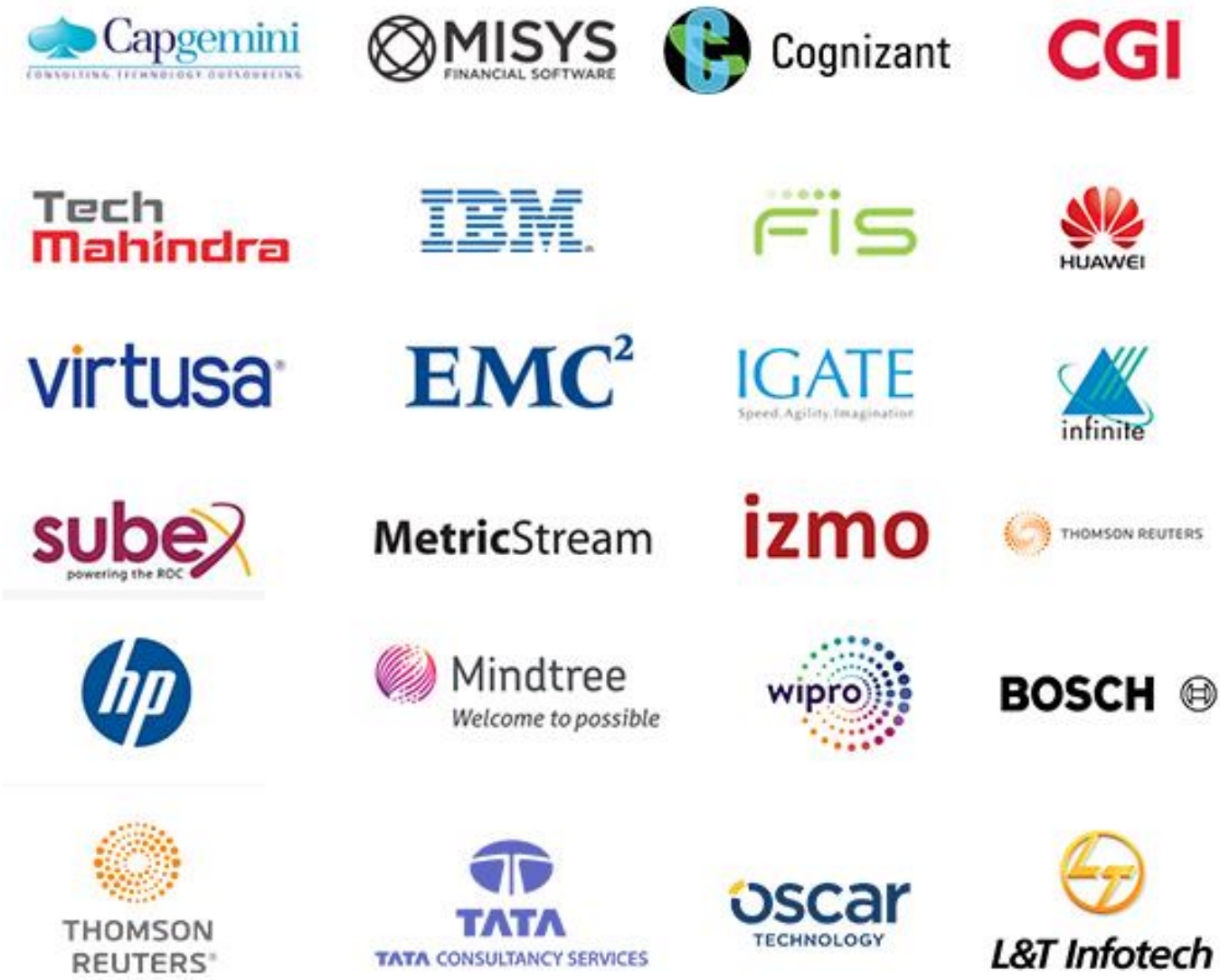
- **Getting Started with PowerShell**
  - PowerShell Console & ISE
  - 
  - Accessing & Updating the Help System
  - Finding & Running cmdlets
- **Navigating within the FileSystem Drive**
  - Working with PSProviders & PSDrives
  - Exploring the File System Navigation & Manipulation
  - Working with Registry Keys
  - Identifying Errors
- **Using Cmdlets & Objects**
  - Differentiating between objects & texts
  - Working with WMI & CIM Objects
  - Finding, Creating & Using Variables
  - Passing Objects through pipelines
  - Diving into Queries & Expressions
  - Formatting
  - Using pipeline to Create a Report
  - Administrating your System Remoting
  - Working with Scripts, Functions & Modules
  - Using PowerShell with AD

- Using PowerShell with **IIS**

## PowerShell DSC

- Introduction
- Architecture
- DSC Configuration files
- DSC Resources
- Pushing DSC Configurations
- Pulling DSC Configurations
- Use Cases/Tasks

# Supporting Enterprises around the Globe



# Our Other Courses

**DevOps & Cloud Computing**

- AWS Admin Training
- Devops Training
- Azure Admin Training

**Artificial Intelligence & IOT**

- Artificial Intelligence Training
- IOT Training

**Big data**

- Hadoop Development Training
- Spark And Scala Training

**Data Science & Blockchain**

- Data Science Training
- Blockchain Training

**QA and Testing**

- Manual Testing
- Big data Testing Training
- Selenium Training
- Appium Training
- Web services Testing Training
- Security Testing Training
- Performance Testing Training
- TOSCA Training
- ETL Testing Training

**RPA and Automation**

- Automation Anywhere Training
- Blueprism Training
- Uipath Training

**Programming & Framework**

- Core Python
- Advanced Python Training
- Django Training
- C,C++Programming Training

**Others courses**

- Digital Marketing Training
- Spoken English Training
- Chat Bot Making Training
- BI Reporting Tools Training
- Sail point IIQ

Established in 2010

# Quality Thought®

The Leader in Software Training


www.qualitythought.in

+91 996 379 9240

info@qualitythought.in