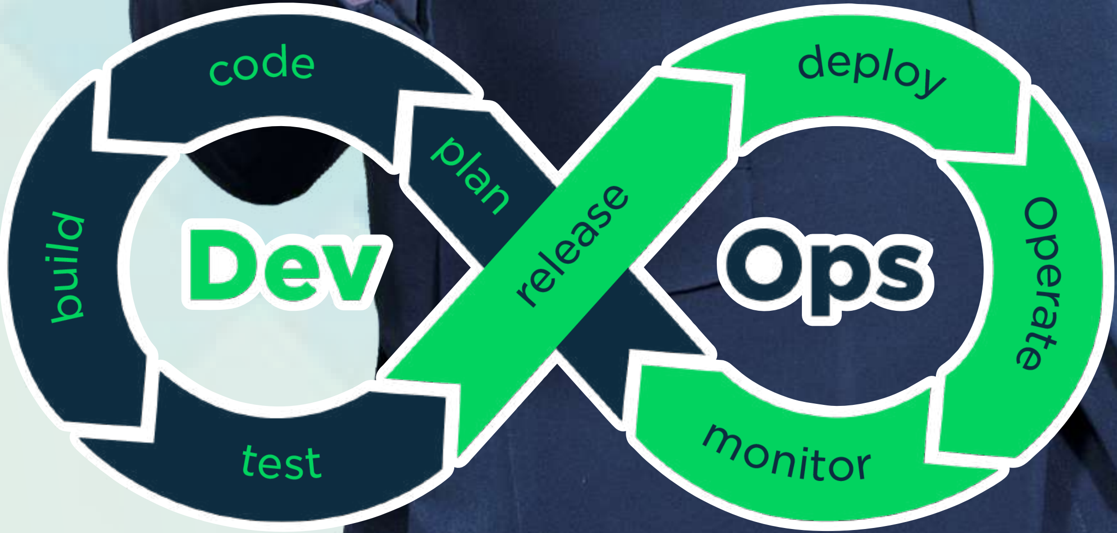
CELEBRATING 12 YEARS

Quality Thought®

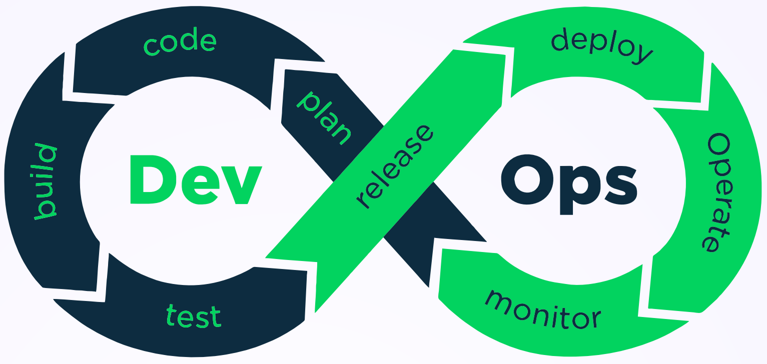
JOB ORIENTED ★★★ JOIP ★★★ INTENSIVE PROGRAM



CELEBRATING 12 YEARS
Quality Thought®

Course Duration
110 Days

Total Sessions Hours
120 Hrs



gitops

Azure DevOps

Jenkins

Terraform

DevSecOps

ANSIBLE

docker

kubernetes

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 are 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

- Installation and Configuration
- Core Concepts of Ansible
  - ☑ Inventory
  - ☑ Module
  - ☑ Adhoc Command
  - ☑ Playbooks
  - ☑ YAML
- Inventory and Playbook Parsing
- Module transport and Execution
- Variable Types
- Variable Precedence
- External data access

Ansible Essentials

- Static Inventories
- Dynamic Inventories
- Common Modules
- Playbook syntax
- Conditionals
- Error Handling
- Variables and Facts
- Templates
- Roles and Ansible Galaxy
- Parallelism

Protecting Secrets with Ansible

- Encrypting data at rest
- Mixing Encrypting with plain YAML

Ansible and Windows

- Running Ansible from Windows
- Setting up windows hosts with Ansible
- Handling Windows Authentication and Encryption
- Automating Windows tasks with Ansible

Jinja2 Templating

- Jinja2 Templating basics
- Control Structures
- Data manipulation
- Comparing Values

Controlling Task Conditions

- Defining a failure
- Defining a change
- Error recovery
- Iterative tasks with loops

Reusable Ansible Content with Roles

- Task, handler, variable and playbook inclusion concepts
- Roles

Troubleshooting Ansible

- ➪ Playbook logging and verbosity
- ➪ Variable introspection
- ➪ Debugging code execution

Minimizing Downtime with Rolling deployments

- ➪ In-place upgrades
- ➪ Expanding and contracting
- ➪ Failing fast
- ➪ Minimizing disruptions
- ➪ Serializing single tasks

Dev Sec Ops

DevSecOps

- ➪ Introducing Security into DevOps Culture
- ➪ Securing DevOps Methodologies
- ➪ Securing DevOps Tools
- ➪ Bridging the Infosec and DevOps Cultures
- ➪ Methodologies for Continuous Security
  - ☑ Version Control
  - ☑ Infrastructure as Code
  - ☑ Security as Code
  - ☑ Continuous Integration and Continuous Deployment (CI/CD)
  - ☑ Observability
  - ☑ Extensibility
- ➪ Code Security
  - ☑ Secure Software Development Lifecycle
  - ☑ SDKs or DevKits
  - ☑ Automated Security Tests
  - ☑ Static Application Security Test (SAST)
  - ☑ Secret Detection
  - ☑ Dependency Scanning
  - ☑ Dynamic Application Security Test (DAST)
  - ☑ Web API Fuzz Testing
- ➪ Container Security
- ➪ Container Image
  - ☑ Container Image Vulnerability Scanner
  - ☑ Minimal or Slim Images
  - ☑ Distroless base Images
  - ☑ Scratch Containers
  - ☑ Rootless Containers
  - ☑ Middle Layers
  - ☑ Multistage Buildings
  - ☑ Prevent data leaks
  - ☑ Container Registry
  - ☑ Registry Vulnerability Scan & Management
  - ☑ Tags Governance
  - ☑ Container Network Security
- ➪ Secure Build Automation
  - ☑ Securing the Automated Build
  - ☑ Vulnerability Detection & Remediation
  - ☑ OWASP Dependency Check
- ➪ Release Gating
- ➪ Continuous Delivery Release Automation
- ➪ Production Monitoring and Ongoing Detection and Remediation

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
  - ☑ Job
  - ☑ Build
  - ☑ Version Control System
  - ☑ Test Executions
  - ☑ Plugins
  - ☑ CLI
  - ☑ Rest API
  - ☑ Security
  - ☑ Distributed Builds
- Jenkins Internals
  - ☑ Jenkins execution under the hood
  - ☑ Importance of Environment Variables
  - ☑ Why Jenkins is called as Cron on Steriods
- 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
  - ☑ Git ☑ Ansible
  - ☑ Docker ☑ Kubernetes
  - ☑ Chef ☑ Terraform
  - ☑ JIRA
  - ☑ Python
- Multi Branch Jenkins Pipelines
- Jenkins Agents
  - ☑ Tool Installations on Agents
  - ☑ Cloud Agents
  - ☑ High Availability

VSTS (Azure DevOps)

- What is Azure DevOps
- Source Code Management
  - ☑ Azure Repos
  - ☑ 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
  - ☑ artifacts ☑ Builds
  - ☑ Builders ☑ Commands
  - ☑ Post-Processor
  - ☑ Provisioners ☑ 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
  - ☑ What is Infrastructure as Code
  - ☑ Infrastructure as Code in the Cloud
  - ☑ How Terraform Does Infra Provisioning
- Installation
- Terraform Constructs
  - ☑ Terraform DSL ☑ Providers
  - ☑ Resource ☑ Arguments
  - ☑ Attributes ☑ Variables
  - ☑ Maps and Lookups ☑ Modules
  - ☑ Local State ☑ Remote State
  - ☑ Taint and Update Resources
- Terraform DSL

☑ Declaring Variables ☑ Working with Resources
☑ Nested Blocks ☑ Dynamic Nested Blocks
☑ Expressions and functions

- Resources and Providers

☑ Null Resource ☑ Local Exec
☑ AWS Provider and Resources

- ☑ Azure Provider and Resources
- ☑ Docker Provider and Resources
- ☑ 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
  - ☑ Docker Overview ☑ Understanding Docker
  - ☑ Difference between Physical Servers, Virtual Machines and Docker
  - ☑ Docker Installation ☑ Docker CLI Overview
  - ☑ Docker and container
- ⇨ Building Container Images
  - ☑ Dockerfile ☑ Dockerfile instructions
  - ☑ Multi stage Docker build
- ⇨ Storing and Distributing Images
  - ☑ Docker Hub ☑ Docker Store
  - ☑ Docker Registry ☑ Docker Trusted Registry
  - ☑ Azure Container Registry
  - ☑ Amazon ECR
- ⇨ Managing Containers
  - ☑ Docker container Commands
  - ☑ Docker Network and Volumes
- ⇨ Docker Networking
- ⇨ Docker Volumes (Storage)
- ⇨ Docker Compose
  - ☑ Installation ☑ Docker Compose Yaml file
  - ☑ Docker Compose Commands
  - ☑ Docker App
- ⇨ Windows Containers
  - ☑ Introduction to Windows Containers
  - ☑ Setting up Docker host for Windows Containers
  - ☑ Running Windows Containers
  - ☑ Windows Dockerfile
  - ☑ Windows containers & Docker compose
- ⇨ Docker Swarm and Services
  - ☑ Introduction ☑ Roles within a Docker Swarm
  - ☑ Creating and managing a Swarm
  - ☑ Managing a cluster
  - ☑ Docker Swarm services & stacks
  - ☑ Load balancing, Overlays and scheduling
- ⇨ Docker Security
  - ☑ Container Considerations
  - ☑ Best Practices
  - ☑ Third Party Security Services
- ⇨ Docker Workflows
  - ☑ Docker for development
  - ☑ Monitoring
  - ☑ Extending to external Platforms
- ⇨ Running Docker in Public Clouds
  - ☑ Amazon ECS and Fargate
  - ☑ Microsoft Azure App Services
  - ☑ Docker Cloud
- ⇨ Docker Enterprise Edition
  - ☑ Installation
  - ☑ Universal Control Plane(UCP)
  - ☑ Docker Trusted Registry (DTR)
  - ☑ UCP Security
  - ☑ Backups for UCP & DTR
  - ☑ 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)

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
- MTTR 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
  - ☑ 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

Chef

System Architecture

- Infrastructure as code
- Desired State Configuration
- Idempotence and Convergance
- Configuration Drift

Chef Tools

- Chef Server
- ChefDK
- Knife
- Test Kitchen
- Supermarket
- Foodcritic
- Inspec

Core Components of Chef

- Cookbooks
- Recipes
- Resources
- Nodes
- Run lists
- Roles
- Environments
- Attributes
- Database

Chef Workflow
Cookbook development

- Generators
- Test Driven Development
- Chef Spec
- Test Kitchen Configuration

Data Driven Cookbooks

- Node Objects, Attributes and Ohai
- Default Attributes
- Attribute Precedence

Customizing Cookbooks

- Customization Strategies
- Creating a Wrapper Cookbook

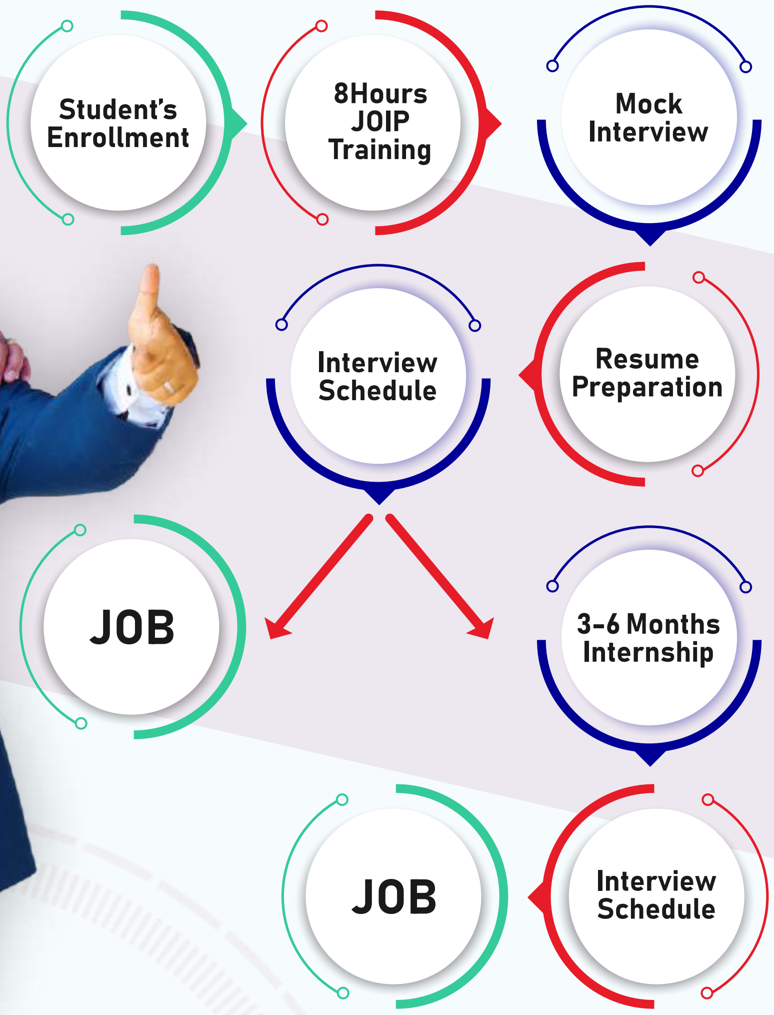
Multi Environment and Multi node Deployment

- Using Roles
- Using Environments
- Using Database

QualityThought®
Transforming Dreams! Redefining Future!

CELEBRATING 12 YEARS

STUDENT TRANSFORMATION STAGES



OUR STUDENTS ARE PLACED IN

wipro	LARSEN & TOUBRO	Apollo	Cognizant	Microsoft	servicenow
amazon	S&P Global	Aadvi Tech Solutions	Medtronic	meslova	{actlogica} Actuarial | Finance | Technology
Sunseaz Technologies Pvt Ltd	CAPRUS IT	SOLUGENIX	Collabera Value. Accelerated.	iPrism Technologies	NOOR
VALUEMOMENTUM	PERVACIO	MIRABEL TECHNOLOGIES	erwin	STIC Soft When you need the results	sumtotal A Skillsoft Company
Nendrasys	magnasoft	Innominds Powering the Digital Next	High Noon Consulting. Possible Is Everything	JRD SYSTEMS	ValueLabs
techforce.ai	ENVISION	splashBI	archents	{↗} techolution	SOLIX Empowering Data Management

QualityThought

99637 99240
1800 309 2939

Quality Thought Infosystems India (P) Ltd.

#302, Nilgiri Block, Ameerpet, Hyderabad-500016 | www.qualitythought.in | info@qualitythought.in