

# DevOps Certification Training program

---

Extensive | Concept planting | Skill inculcation  
Coding knacks | Practical tutoring | Career  
Navigation | Interview crackers | Network building

**50+ hours of Instructor Led training + 10 hours of  
skill grooming & LIVE projects**



# WICULTY Predicta | DevOps Sneek Peak on Demand

- The DevOps market is expected to grow from USD 3.42 Billion in 2018 to USD 10.31 Billion by 2023, at a Compound Annual Growth Rate (CAGR) of 24.7%

- Gartner

- Average salary given to a DevOps Engineer is around \$127,231 per annum

—Indeed.com

The average salary for a DevOps engineer is a juicy \$100,000 per year. Yet even that

- isn't enough to keep DevOps positions filled. According to a 2016 study by Indeed, DevOps engineers are the single hardest position for IT teams to keep filled.

- Glassdoor on DevOps demand

DevOps implementation has

- provable, impressive results. With new automated standards, tech teams can deploy code faster than ever before, and with fewer errors.

- Buisness Review on DevOps tech

The core goals and philosophies of

- DevOps are likely to remain not only relevant, but in high demand for at least the next decade

- Information Week

“ DevOps DevOps Shouts the IT Job Market to fill many openings against raising demand in the Industry being the most adopted practice in the IT domain ”

— Gladys Nelwhelmer



# Uniqueness with our training program

## "Wiculy sculpts the PRO out of you"

WICULTY

**Concept planting** : Learn every bits and pieces of what you've grasped to pitch concepts in depth

**Coding Knacks** : Groom your coding knowledge on all concepts

**Skill Inculcation** : Master the focus skill for your future career

**Ideation routes** : Learn how to take up projects from what you learn



## Wiculy's potential package drives us to train DevOps & cloud computing

- Extensive practical Training

- Industry-Woven Curriculum

- Corporate readiness after the course

- Course coverage to crack Certification

- Injection of Trending tools in curriculum

- Intensive grooming on chosen course

### Our initiatives for learner's benefits :

- Career Navigation
- Corporate Conjunction
- Webinars on trends
- Peer Learning
- Wicublia
- Power Groups
- Interview Crackers





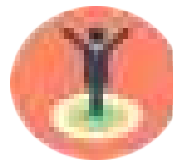
## Benefits as Wiculyt learner



We will provide training in a most advanced approach of adaptive practical tutoring methodology



You will master the whole L-wheel along the course through our concept planting & coding knacks



After the course one can easily crack any certification & start to work on any projects as designed by our expert team



A sturdy LMS access to keep your learning alive for lifetime with us. We also keep on adding study materials, Video tutorials etc. based on the growing trend in the market

### After the course?

**Get your head strong to become a skillful PRO - under our tailoring!!**



Now you turned as an unique asset to your organization! your pay-cheques are more than your dream!!

**Finally its " YOU"  
A deserving TECHIE  
on cloud 9.....**

...



Groomed with cognitive ability | Concept planting would help you capture any topics across your projects efficiently| Almost done

**Inherit the trait of a PRO techie!!**



Keen practice + skill grooming + Ideation triggers on modules learnt + teach-backs among peers & community + Project handling bars from the curriculum

**Skill Inculcated to sustain career**

...

## Witness in LIVE soon!!

### How Wiculyt program works?



## Pick a course & Join our Learning

**community** Take a Great leap to kick start a better learning inside our exclusive realm of courses we offer for you to choose and pick up the career grid ahead



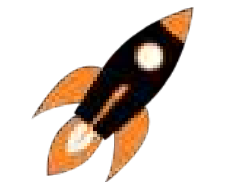
**Start learning** from our expert pool of trainers with

Industry-woven curriculum in hands-on/practical mode and gain a self tag of you as "Corporate ready" professional to handle projects readily after the course



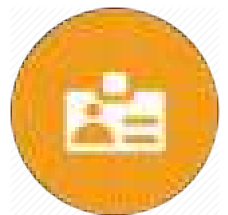
## Practice the craft after the session Our LMS

gives complete assistance along your learning trials with updated study materials & resources curated by us to engage one's curiosity to Upskill & explore lot of stuffs



## Boost up your learning by adding skills

cracking reputed global certifications & international credentials that equip one to stay unique in their domain further ahead like a "master-bee" ! we want you to be.



**Get the best job guidance** & multiple grooming

sessions to crack interviews from our Career navigation team through events,taking your foot steps much deeper further ahead in the chase'n'race for career!  
**Make use of our Career resources in LMS for lifetime!**

# Wiculy E4 Model

## Energy looped inside the wisdom shell

## Explore our courses

## Evaluate our curriculum

**Equip** with our expertise

**Enhance** skills with us

**Eternity** to be continued

**“**

|| . . . .

† . . . .

‡ . . . .

. . . .

**& that  
proves our expertise  
in exclusive courses”**

# WICULTY

06 courses | 80+ Skills | 04 Tracks | 50+ Hours

# Curriculum



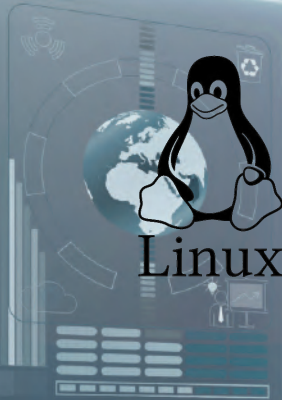
***m*aven**



**Jenkins**



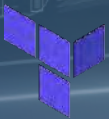
**docker**



**Linux**

**DevOps**

**Industry-Woven training**



**Terraform**



**ANSIBLE**





# WICULTY

## DevOps Certification Training Program

“ According to GlassDoor,  
The average salary for a DevOps  
engineer is a juicy \$100,000 PA  
Yet even that isn't enough to  
keep DevOps positions filled ”

**Kickstart Note :** This training is purely hands-on & designed to train learners with best practices in Continuous Development, Continuous Testing, Continuous Integration and Continuous Monitoring of software throughout its development life cycle ! Learn it in pure practical mode with Wiculy.com

### CORE CURRICULUM

**Module 1 :** Git & Github Version Control System

**Module 2 :** Maven - Build Tool

**Module 3 :** Jenkins - CI & CD Tool

**Module 4 :** Ansible - Config Management

**Module 5 :** Docker

**Module 6 :** Kubernetes

**Module 7 :** Terraform - the Automate Builder

**Module 8 :** Linux Admin Level

**Module 9 :** Shell Scripting

**Total Chapters in Extensive Coverage : 40 Chapters**



### WHO CAN LEARN?

- Anyone who works closely with development and operations team
- System/Network Administrators
- Someone with skills in anyone scripting language
- Solution Architects
- Security Engineers
- Software testers
- Application support/developers



### LEARNING OBJECTIVES

- To understand & learn the complete DevOps cycle
- To learn all the tools in hands-on mode with case-studies
- To gain a perfect knowledge in DevOps & applicate directly in projects after the course
- To inculcate the sought after Devops Skills as part of the course

Duration:

Instructor-Led Training	50 Hours
Skill Grooming & Projects	10 Hours



# WICULTY

## DevOps - A Big Picture



Get started and leap into the world of DevOps with Wiculy! Here we start the wide opening for this course that for sure equip learner's learning journey with us

### DevOps - A Big picture

- What is DevOps?
- Acronym Dev+Ops?
- History of DevOps
- DevOps Misconceptions
- Overview- Gamut of DevOps tools
- End-to-End DevOps workflow
- Roles and Responsibilities of DevOps Resource
- Who can learn DevOps?
- Importance & goals of DevOps Practices in real-time
- SDLC models, Agile and DevOps
- Opportunities, Trends and Future of DevOps
- Overview of Version Control, Build and Deployment Process, Continuous Integration and Deployment, Configuration management, Containerization, Virtualization & Cloud platform. etc
- Roles of Cloud platforms in DevOps



#### WHO CAN LEARN?

- Anyone who works closely with development and operations team
- System/Network Administrators
- Someone with skills in anyone scripting language
- Solution Architects
- Security Engineers
- Software testers
- Application support/developers

#### Duration:

Instructor-Led Training	50 Hours
Skill Grooming & Projects	10 Hours





# Module 1

## Git & GitHub version control system



### Chapter 1 : Version Control Systems Overview

#### Concepts & practicals

- Introduction to version control systems (VCS)
- Different version control systems in the market
- Evolution of VCS
- Roles and goals of VCS in Source code management & DevOps
- Principles & features of Version Control Systems
- What is Git & Github? Differences!
- Roles and Responsibilities of DevOps Engineer in Git

#### Job oriented: Hands-On preparation

- Creating an end-to-end flow-chart fitting Git in DevOps

### Chapter 2 : Getting started with GIT

#### Concepts & Practicals:

- Git Basics & Architecture
- Git's unique features in Source code management
- End-to-End Git Work-flow. A bird's eye view
- Git Vs SVN Vs Other commercial VCS
- Git Command Line & GUI
- On-premise Vs Hosted Git Solutions
- Overview of GitHub, GitLab, Bitbucket. etc.

WICULTY



# Module 1 Git & GitHub

WICULTY

## Job oriented: Hands-On preparation:

- Creating Git Architecture diagram to answer in interviews
- Applying Git commands and perform end-to-end hands-on
- Exploring GitHub and its UI features
- Note: As It's just an introduction, you see more hands-on in up coming Chapters

## Chapter 3 : Let's make our hands Dirty with Git

### Concepts & Practicals:

- Test your knowledge! Git Jargons. How much you can talk?
- Git Installation, Uninstallation, Up-gradation on Linux
- Setting up Mandatory configurations & best practices
- '\$git config' command to setup User, Email, Editor and Credentials
- What is Source, Stage and Local repository.
- Git Revision Structure – SHA, User, Email, Commit Message and other meta data
- Git internals. How Git generates SHA value? What is this checksum code & data integrity
- Significance of Staging Index. Skipping the staging. Best practices

## Job oriented: Hands-On preparation:

- Setting up Remote repository on GitHub from scratch
- End-to-End git work-flow execution with commands
- Commit your first change to Remote repository, Consciously!
- \$Commands: 'git add', 'git commit', 'git push' with options

## Chapter 4 : Advanced Git. Let's Deep dive

### Concepts & Practicals:

- Git command line with most frequently used options in real-time
- # Check the history of a file
- \$ git log –author
- \$ git log --grep

# Module 1 Git & GitHub

WICULTY

## Chapter-4: Advanced Git. Let's Deep dive

### Concepts & Practicals :

- `$ git log --since`
- `$ git log --until`
- `$ git log --oneline`
- `$ git log --grep` , etc.
- # How do you get diff of your changes?
- `$ git diff`
- `$ git diff --staged`
- `$ git diff sha..sha`
- # Removing a file in Git
- `$ git rm`
- Is deletion permanent? Best practices.
- Resurrect a deleted file
- Renaming a file in Git
- `$ git mv` command
- Verify history after file rename
- # Show more details about a change
- `$ git show`
- #Undoing the changes
- Revert a change from Source Area
- Revert a change from staging area
- Revert a committed change
- `$ Git pull` & `fetch` commands
- Difference between pull and fetch
- Difference between clone and pull
- Pull and Fetch best practices

### Job oriented: Hands-On preparation:

- Practice all above commands with real-time use cases
- Interview perspective: In what case do you apply in your project?



# Module 1 Git & GitHub

## Chapter 5: Branching & Merging – Release management with Git Concepts & Practicals:

- What is a branch? When and Why do we create a new branch
- Importance of master branch & stable code
- Branching Strategies/Models – Pros and Cons
- Switching branches for parallel development
- Merging from one branch to another. Best practices

### Job oriented: Hands-On preparation

- Creating a new branch and making it public
- Explain the branching model that you followed in your project for different releases
- Merging & Conflict resolution. Practice a complete cycle

# Module 2

## Maven

### The Build tool



WICULTY

**maven**

#### Chapter 1 : Overview of build tools & Deployment process

##### Concepts & practicals

- Build and Deployment automation End-to-End Workflow
- Roles and Responsibilities of DevOps Engineer in Software Build & Deployment
- Introduction to Maven build tool
- Maven Vs ANT(Key Features of Maven Over ANT)
- Feel the pain of source code manual compilation with manual example
- Necessity of compilation and transforming source code into binaries/executables
- Artifact, Binaries, Executables, object code definition - Get terminology

##### Job oriented: Hands-On preparation

- Explain complete flow of software development, build and test process
- Flow chart for the same



# Module 2 Maven

WICULTY

## Chapter 2 : Getting started with Maven

### Concepts & practicals

- Maven Installation and Prerequisites, Downloading Maven and JDK
- Setting up JAVA\_HOME, M2\_HOME and PATH ENV variables
- Discussion about \$USER\_HOME/.bashrc file and Installation of any tool in Linux
- Java build process. Packaging sequence (.Class, .Jar, .War, .Ear , etc.)
- TEST YOUR KNOWLEDGE BEFORE DEEP DIVE: Look for it
- What is compilation & why we compile the source code
- Packaging sequence for Java application
- What is Build
- What is Deployment
- Different Environments in Software development
- Development, QA, Ops & DevOps teams Interaction and Collaboration

### Job oriented: Hands-On preparation

- Setting up Maven in your practice Linux machine
- Setting up environment variables for any tool installation on Linux

## Chapter 3 : How to collaborate well with DEV and QA teams

### Concepts & practicals

- Creating a project using Maven
- Maven's convention over configuration feature
- Understanding Project source structure
- Understanding Test Driven Development (TDD) approach
- Understanding Junit unit testing framework
- Overview of Software development & other testing methodologies
- Software Development and Testing Best practices

### Job oriented: Hands-On preparation

- Creating a first project in practice Linux machine



# Module 2 Maven

WICULTY

## Chapter 4: Hands-on with Maven

### Concepts & practicals

- Building your first project
- \$ mvn install command & deep discussion about maven life cycle phases
- Understanding build output, test results, class files, packages etc.
- Verifying built artefacts, naming convention and m2 local repository
- Dependency Management: What is code dependency-Maven's automatic dependency resolution feature
- Direct and Transitive dependencies-Defining dependencies in pom file
- Maven binary repositories-Local, Private and Central repository

### Job oriented: Hands-On preparation

- Build the project that you have created
- Witness the build output and artefacts
- Explain Maven's internal automation process

## Chapter 5: Implementing various Build methodologies

### Concepts & practicals:

- Build Types - hands-on
- Complete build/Clean build/Full build
- Nightly build
- Daily build
- Bugfix build
- Adhoc/unplanned/emergency builds

### Job oriented: Hands-On preparation

- Execution of all build types with hands-on for Gamut Kart project
- Maven plug-ins, Skipping Test compilation, run and execution of tests when required
- \$ mvn install -DskipTests
- \$ mvn install -Dmaven.test.skip=true
- Creating project for web application. Building the war file

# Module 2 Maven

## Chapter 6: Projects & Deployments deep dive

### Concepts & practicals

- Understanding various Environments Usage: DEV, QA, SIT, UAT, Performance, STAGE, PROD. etc..
- Deployment promotion methodologies from one environment to another environment
- Application servers and web servers - Comparison
- Tomcat startup scripts, deployment path, port configurations etc.
- Exploring WAR / EAR files and its resources
- Deployment best practices & roll back process

### Job oriented: Hands-On preparation

- **Project-1:** Automate complete build and deployment process for basic application using Maven and Shell scripts
- **Project-2:** Building and deploying GamutKart project -Real-time project in your machine.
- Making you comfortable to work with Dev, QA & Ops teams in Agile environment understanding SLA's
- Introduction to Jenkins & CI/CD process to reduce deployments turn-around time

# Module 3

## Jenkins - CI/CD tool



WICULTY



Jenkins

### Chapter 1 : Jenkins & CI/CD Process Overview

#### Concepts & practicals

- Introduction to Agile Development
- Definition of Continuous Integration (CI), Continuous Delivery (CD), Continuous Deployment (CD)
- Difference between CI and CD
- End-to-End CI & CD Pipeline concepts

#### Job oriented: Hands-On preparation

- Explain complete flow of software development, build and test process
- Flow chart for the same

### Chapter-2: Getting started with Jenkins (Installation & Production Set-up)

#### Concepts & practicals

- Jenkins Installation and Configuration in Production
- Tomcat and JDK installation. Setting up environment variables.
- Exploring Jenkins Dashboard UI
- Git, Maven and & Build tools Installations & integration
- Different types of Jenkins Jobs. Freestyle, Pipeline, multi-configuration projects





## Chapter 2: Getting started with Jenkins

### Concepts & practicals

- Creating Jenkins Pipeline jobs and understanding all project options
- Configuring automated Builds for WAR package creation

### Job oriented: Hands-On preparation

- Setting up Jenkins in your machine
- Create first Jenkins job and automate builds for Gamut Kart(Our own project)
- Integrating source code management tools and polling VCS to achieve CI/CD

## Chapter 3: Jenkins Administration

### Concepts & practicals

- Jenkins global configurations and administration settings
- Build tools configuration in Manage Jenkins
- Jenkins job build steps, triggers and post build actions
- Setting up System notifications for users
- Email configuration for sending CI/CD notifications
- Jenkins backup mechanisms and restoration policies
- Jenkins migration and upgradation
- Setting up Security for Jenkins
- Matrix based and Project based security
- Jenkins authorizations to Dev, QA and other stakeholders. Best

### Practices Job oriented: Hands-On preparation

- Installation of thin backup plugin and configuring auto backups
- Setting up authorization policies using matrix-based security
- Creating Users and setting up authorization policies

## Chapter 4: Jenkins Pipelines - Deep dive & advanced concepts

### Concepts & practicals

- What is pipeline?
- Advantages of build pipelines
- Creating pipeline projects using GUI
- Manual/GUI pipeline Vs scripted pipeline
- Groovy scripts DSL and syntax
- Creating Parameterized build jobs
- Manual/GUI pipeline Vs scripted pipeline
- Groovy scripts DSL and syntax
- Creating Parameterized build jobs

### Job oriented: Hands-On preparation

- Write deployment scripts for different environments
- Writing pipeline script: Jenkinsfile in declarative Groovy
- Deployments to QA, SIT, UAT, etc environments using a single Jenkins job

## Chapter 5: Jenkins Plugins & CLI

### Concepts & practicals

- Introduction to Plugins. What is a plugin?
- Plugins Installation, Un-installation and upgrade
- Different ways of plugin installation and management
- Finding suitable plugins and interpreting plugins documentation for real-time scenarios
- Use cases and applying Jenkins CLI for real-time scenarios
- Explore Jenkins CLI options

# Module 3 Jenkins

## Chapter 5 : Jenkins plugins & CLI

### Job Oriented: Hands-On preparation

- Most Frequently used plugins in real-time (Top 10 plugins usages)
- Triggering build jobs from command line
- Cleaning up builds runs using CLI

## Chapter 6 : Distributed Builds & Master/Slave concept

### Concepts & practicals

- When do we implement distributed builds?
- Setting master and slave nodes. Best practices
- Running builds in parallel on distributed environment
- Setting up and using SSH agents, cloud agents
- Load balancing and fine tuning builds and deployments using master and slave

### Job Oriented: Hands-On preparation

- Configuring master and slave mechanism
- Configuring slave nodes and adding to master
- Developing build and deployment automation for Gamut Kart (Our own project) on Master & Slave nodes



# Module 4

## Ansible



WICULTY



ANSIBLE

### Chapter 1 : Introduction to Containerization & Docker Concepts & practicals

- What is configuration Management tool and What is Ansible
- How ansible Works
- What is Infrastructure as a code
- About Idempotency
- About some important Ansible Terminology

#### Job oriented: Hands-On preparation

- Creating an end-to-end flow-chart fitting Ansible in DevOps echo system
- Note: As It's just an introduction, you see more hands-on in up coming Chapters
- Creating an end-to-end flow-chart fitting Ansible in DevOps echo system
- Note: As It's just an introduction, you see more hands-on in up coming Chapters

### Chapter-2: Docker Installation in Production

#### Concepts & practicals

- Ansible Architecture
- Pre-Requisites for controller Node



# Module 4 Ansible

WICULTY

## Chapter 2 : Ansible Deployment

### Concepts & practicals

- Test Environment Setup
- Installation and configuration
- Ansible Configuration file
- Pre-Requisites for Managed Node
- Ansible Inventory
- Ansible Communication
- Communication checks with password Authentication
- Communication with key-Based Authentication
- Ansible Architecture
- Overriding the Default HOSTS File
- The Default System Ansible.Cfg File
- Overriding the Default Roles Path

### Job oriented: Hands-On preparation

- Creating Ansible Architecture diagram to answer in interviews
- Applying Basic commands and perform end-to-end hands-on
- Note: As It's an Ansible Setup Management and target servers session , you see more hands-on in up coming Chapters

## Chapter 3: Ad-Hoc Execution with Ansible

### Concepts & practicals

- Why use ad-hoc commands, syntax of ad-hoc commands?
- Use cases for ad-hoc tasks

Copying and Executing Modules

Rebooting servers

Managing files,Managing packages

Managing users and groups

Managing services

# Module 4 Ansible

WICULTY

## Chapter 3 : AdHoc Execution with Ansible

### Job oriented: Hands-On preparation

- Running first Ansible commands
- command module, copy module, user module, gathering facts, shell module
- adhoc command to logon 'all' and execute python with json lib.
- adhoc command to logon 'hostname' and execute python with json lib.
- Display facts from all hosts and store them indexed by hostname at /tmp/

facts

- Display only facts returned by \*.ipv4
- Adhoc command to List all the files and folders using command module
- Adhoc command to cat a file using command module
- Adhoc command to copy file from local to remote using copy module
- Adhoc command to install package using apt module
- Adhoc command to uninstall package using apt module
- Adhoc command to create user ansi and set the password in all target

servers using user module

- Adhoc command to Remove user ansi in all target servers using user module

## Chapter 4: Working with PlayBooks

### Concepts & practicals

What is configuration, deployment, and orchestration

YAML Structure, Playbook structure

Ansible playbooks - quick start

Playbook syntax checks

Variables in Ansible

Ansible Sections - target, tasks, handlers

Patterns: targeting hosts and groups

Defining Variables in Ansible Code, Use Cases

Debug the console logs

Ansible Facts, About System Facts: Common Values for Playbooks

# Module 4 Ansible

WICULTY

## Chapter 4: Working with PlayBooks

### Job oriented: Hands-On preparation

- Introduction to Ansible Playbooks
- Write a playbook to install package “tree” and check the version in remote/target servers
- Write a playbook to copy a file to remote servers from Management server
- Ansible Sections, Target section, tasks Section – Install Apache2 Package
- Variable Declaration in a playbook
- Defining variables in “vars.yml” file and call the file in playbook “4\_callVarsFile.yml”
- Debug- date/time stamp and debug

## Chapter 5: Conditional, Loops and Handlers

### Concepts & practicals

- what is Conditionals in Ansible
- what are Loops in Ansible
- Handlers in Ansible
- Ansible Vaults- Introduction and Importance of vaults in ansible
- Vault - commands and usage in playbooks

### Job oriented: Hands-On preparation

- Loops: with\_items -Create list of users in a target/remote servers
- Delete a list of users in a target/remote servers using loops concept
- Install list of packages using loops concepts
- Condition: When - Check the condition of OS family and install the package using appropriate module
- Update Packages - Write a playbook to update the packages and install package

# Module 4 Ansible

WICULTY

## Chapter 6: Advanced: Ansible Includes and Playbook Optimization

### Concepts & Practicals

- Basic Include Statements
- Includes - Breaking Your Playbook Into Discrete Plays
- Ansible Playbook Optimization
- -1 Copy and Fetch Modules
- -2 Facts
- -3 Forks
- -4 Serial & Max\_Fail\_Percentage
- -5 Asynchronous Action and pooling
- -6 DelegateTo
- -7 Ignore Failed commands/Basic Error Handling
- -8 Tags
- -9 Jinja2 Templates
- -10 Dry-Run
- -11 Simple Variable Substitution
- -12 Lookups
- -13 RunOnce
- -14 Local Actions
- -15 Notify
- -16 Prompt - Interactive Playbook
- -17 Starting At Task or Stepping Through All Tasks
- -18 Passing Variables Into Playbooks at the Command Line

### Job oriented: Hands-On preparation

- How to write playbook examples for Copy and Fetch Modules, Facts,
- Forks, Serial & Max\_Fail\_Percentage
- Asynchronous Action and pooling
- DelegateTo, Ignore Failed commands/Basic Error Handling
- Tags, Jinja2 Templates, Dry-Run
- Simple Variable Substitution



# Module 4 Ansible

WICULTY

## Chapter 6: Advanced: Ansible Includes and Playbook Optimization

### Job oriented: Hands-On preparation

- Lookups, RunOnce, Local Actions
- Notify, Prompt - Interactive Playbook, Starting At Task or Stepping Through All Tasks, Passing Variables Into Playbooks at the Command Line

## Chapter 6: Advanced: Ansible Includes and Playbook Optimization

### Concepts & Practicals

- Introduction on Ansible Roles
- Role directory structure
- storing and finding roles
- Using roles
- Running a role multiple times in one playbook
- Include and Dependency Management
- Passing different parameters
- Introduction on ansible tower
- Tower installation and Tower Dashboard

### Job oriented: Hands-On preparation

- How to include a role
- how to use conditionally include a role
- passing other keywords, including variables and tags, when including roles
- Passing different parameters in each role
- How to create new roles, share roles on Galaxy, role management

# Module 4 Ansible

## Chapter 7: Other Concepts from Ansible

WICULTY

### Concepts & Practicals

- 'Setup' Module
- The 'File' Module
- 'Pause' Module
- 'WaitFor' Module
- 'Yum' Module
- 'Apt' Module
- 'Service' Module
- 'Copy' Module
- 'Command' Module
- 'Cron' Module
- 'Debug' Module
- 'Fetch' Module
- 'User' Module
- 'AT' Module
- 'DNF' Module
- 'Apache2\_Module' Module
- 'SetFact' Module
- 'Stat' Module
- 'Script' Module
- 'Shell' Module
- 'Raw' Module
- 'Ping' Module
- 'Unarchive' Module
- 'HTPasswd' Module
- 'GetURL' Module
- 'Group' Module
- 'Mail' Module
- 'Filesystem' Module
- 'Mount' Module

# Module 5

## Docker



WICULTY

### Chapter 1 : Introduction to Containerization & Docker Concepts & practicals

- Basics of Virtualization
- Difference between Virtual machine (VM), Physical machine and Docker container
- What is containerization?
- What is Docker? Why Docker and it's features. Who can use it?

#### Job oriented: Hands-On preparation

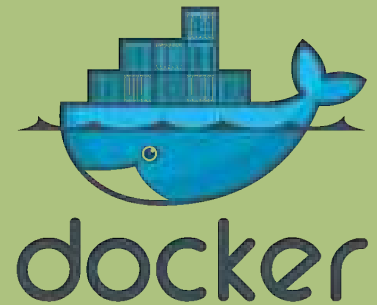
- Virtual machine and Docker usage in real-time and DevOps world

### Chapter-2: Docker Installation in Production Concepts & practicals

- Docker supported platforms
- Docker prerequisites in production and commands to verify the same
- Docker Installation & Uninstallation in production
- Configuring Docker to be executed

#### Job oriented: Hands-On preparation

- Virtual machine and Docker usage in real-time and DevOps world



# Module 5 Docker

WICULTY

## Chapter 3 : Getting started with Docker containers

### Concepts & practicals

- Creating first container
- Cgroup & Namespace Kernel features for containers
- Root file system, networking and processes isolation.
- Docker image concepts. Shipping the application code with dependencies
- Difference between Docker Image and Container
- Creating any flavor of Linux containers on any linux host
- Creating Linux containers on Windows. Concepts involved

### Job oriented: Hands-On preparation

- Creating container with different options
- Installation of frequently used Linux commands (ssh, net-tools, vim ..etc. )
- Inspecting the new container (hostname, IP, hosts file, processes, n/w capabilities ..etc.)
- SSH installation in containers and enabling SSH for root

## Chapter 4 : Container management commands

### Concepts & practicals

- Shutdown Docker container
- Listing all containers in host
- Listing only running containers
- Listing last few number of containers
- Inspecting docker container information
- Listing last created container
- Creating a container with our own name
- Renaming a container
- Deleting one, all, stopped and running containers
- Starting, stopping and restarting containers from host
- Attaching to a running container

# Module 5 Docker

WICULTY

## Chapter 4 : Container management commands

### Concepts & practicals

- Keyboard shortcuts
- Inspecting container processes from host
- Stopping the container gracefully and forcibly
- Find more about a container. Understanding container inspect JSON format
- Creating and Pushing a running container in daemon mode

Creating demonized containers

### Job oriented: Hands-On preparation

- For all above commands, discussion about different use cases
- When do we use a particular command
- Execute all command in your lab machine in practical ways

## Chapter 5 : Container management commands

### Concepts & practicals

- Verifying resource utilization and usage statistics - stats command
- Allocating Memory
- Allocating CPU
- Allocating Disk space
- Updating computing constraints on a running container

### Job oriented: Hands-On preparation

- Discussion about when to change compute parameters
- Pros & Cons of allocating dedicated memory manually
- Execute all commands on your lab machine



# Module 5 Docker

WICULTY

## Chapter 6 : Docker Volumes - Persistent Data

### Concepts & practicals

- Docker Volume creation
- Inspecting Volumes
- Mounting Docker volume to containers
- Read-only volume

### Job oriented: Hands-On preparation

- Applying Volumes for Persistent data - Real-time use cases
- Removing Volumes and all unused Volumes
- Use-case discussion, Volumes to store user data in e-commerce application

## Chapter 7 : Docker images - Deep dive - Project - Application containerization

### Concepts & practicals

- Understand more about Docker images - real-time use cases in DevOps
- Advantages of Docker images in application deployments
- Docker Image creation techniques
- Dockerfile instructions and usage
- Solving 'Works in my machine problem' with Docker Images
- Restoring environments with Docker images
- Auto scaling environments using Images
- Setting up dev environments with images. Docker Advantage!
- Creating and setting up account in Docker-hub

### Job oriented: Hands-On preparation

- Writing Dockerfile for to create image for containerizing Gamutkart application
- Most frequently used and helpful images walk-through
- Implementing Self-service deployment models.
- Setting up nginx for web application
- Project:1 - creating custom Docker image for Nginx web application
- Project:2 - Creating custom Docker image for our own Gamutkart e-commerce web application

# Module 5 Docker

WICULTY

## Chapter 8 : Jenkins CI/CD Pipeline integration with Docker.

### Concepts & practicals

- Creating disposable environments using Docker images
- Integrating Docker with Jenkins CI/CD Pipeline
- Writing deployment scripts for provisioning environments with images

### Job oriented: Hands-On preparation

- Scaling up environments instantly with docker images.
- Writing shell scripts for environment creation and management.

## Chapter 9: PROJECTS

### Project-1:

- Web Application Deployment & Hosting using Docker Images
- Create docker image with Web application code, Nginx and other prerequisites and dependencies
- Creating auto environments with shell scripts and Docker images

### Project-2:

- Gamutkart web application deployment & Environment setup automation
- Building Gamutkart Application
- Creating Docker image for Gamutkart application with prerequisites such as JDK, Tomcat & other configurations
- Creating environments automatically using shell script and launch Gamutkart e-commerce application

# Module 6

## Kubernetes

### Chapter 1 : Introduction to Kubernetes & Architecture & Orchestration

#### Concepts & practicals

- What is Kubernetes? History of Kubernetes
- Kubernetes Architecture & Building Blocks
- Docker Vs Kubernetes (Complementary technologies)
- Application deployment challenges. Why Container Orchestration tools?
- Kubernetes as Data Centre OS

#### Job oriented: Hands-On preparation

- Use case - Google Challenges & Experiences of running billions of containers
- Container Orchestration Capabilities in real-time: Auto-scaling, Zero Downtime Release, Rollback, Self-healing, Rolling Updates

### Chapter 2 : Kubernetes Production Installation & Services Deep dive

#### Concepts & practicals

- Installing Kubernetes using Kubectl
- Production Kubernetes cluster setup with multiple nodes
- Kubernetes Services, Deep-dive



WICULTY



# Module 6 Kubernetes

WICULTY

## Chapter 2 : Kubernetes Production Installation & Services Deep dive

### Concepts & practicals

#### Kubernetes Services, Deep-dive

- Master/Control Plane Server
- Worker Nodes
  - Kube-API Server
  - Kube-Controller
  - Kube-Scheduler,
  - Cluster-Store/etcd DB    Kubelet
  - Kube-Proxy
  - Kubectl, Container runtime

### Job oriented: Hands-On preparation

- Setting up 2-node Kubernetes cluster in Google cloud platform
- Kubernetes cluster management practices

## Chapter 3 : Working with Microservices & Kubernetes Role

### Concepts & practicals

- Monolithic Vs Microservice Architecture
- Development, Build, Deployment & stability challenges in Monolith
- From Monolith to Microservices - Project ideation
- The Docker & Kubernetes role in Microservices

### Job oriented: Hands-On preparation

- Discussion - How Microservice concept helps deployment automation and environment stability.

# Module 6 Kubernetes

WICULTY

## Chapter 4 : PODs Management, Running Application on multi PODs cluster

### Concepts & practicals

- What is POD. POD Vs Container
- Single & Multi Container PODs & Use cases
- POD - Horizontal scaling
- Creating Pods - Writing Yaml files
- Deploying Pods & to Run the application on Kubernetes
- Pod life-cycle management commands

### Job oriented: Hands-On preparation

- Project:1 - Create 10 pods in Kubernetes cluster and run Nginx application
- Project:2 - Create 10 pods in Kubernetes cluster and run GamutKart web application

## Chapter 5 : Kubernetes Deployments - Advanced Concepts

### Concepts & practicals

- Application Autoscaling, Self-healing, Rollbacks & Rolling Updates with Zero downtime
- Kubernetes declarative Model - Desired state & Current state
- ReplicaSet objects and create Deployment object - Writing Yaml files
- Kubernetes deployment management commands
- Inspecting PODs and application instances

### Job oriented: Hands-On preparation

- Project:1 - Implement Self-healing & Auto-scaling for Nginx application
- Project:2 - Implement Self-healing & Auto-scaling for GamutKart web application



# Module 6 Kubernetes

WICULTY

## Chapter 6 : Kubernetes Services & Load balancing

### Concepts & practicals

- Create a Service object that exposes an external IP address - Writing Yaml files
- Google Cloud basics. Creating compute instances in Google Cloud Platform (GCP)
- Setting up Firewall rules and ports
- Setting up NodePort Load balancer

### Job oriented: Hands-On preparation

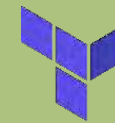
- Project:1 - Create 10 POD instances, Setup the Load-balancer for Nginx application & make it accessible by the end-user.
- Project:2 - Create 10 POD instances, Setup the Load-balancer for GamutKart web application & make it accessible by the end-user.

# Module 6

## Terraform



WICULTY



# Terraform

### Chapter 1 : Terraform Introduction & Practicals

#### Concepts & practicals

- Terraform Introduction & Concepts Installation
- Core Concepts Overview
- Terraform- AWS Setup
- Terraform - Spin up instance
- Terraform Variables
- Managing credentials
- Best Practices in terraform
- Terraform launch EC2 with more attributes
- Terraform commands and document overview

#### Job oriented: Hands-On preparation

- Running the basic setup of terraform in Realtime application

### Chapter 2 : Core Concepts

#### Concepts & practicals

- Terraform Output
- Terraform State files
- Terraform Provisioner
- Terraform DataSources



# Module 7 TerraForm

WICULTY

## Chapter 3 : Advanced Concepts

### Concepts & practicals

- Terraform Administrator (AWS)
- Terraform Modules
- Terraform Conditionals
- Terraform Workspace
- Terraform Packer

### Job oriented: Hands-On preparation

- Hands-on explanation on all Advanced Concepts with real-time application
- Admin Demo as Administrator (Terraform+AWS)

## Chapter 4 : Additional Concepts

### Concepts & practicals

- Terraform Small Medium|Large codes structures
- Terraform sample codes for other service providers

**Certification details and Guidance**

# Module 8

## Linux



WICULTY

### Chapter 1: Linux/Unix - The Big Picture & Linux Commands

History of UNIX/LINUX

Features and Benefits of UNIX/LINUX

Different flavors of UNIX/LINUX

Difference between UNIX, DOS, Windows and LINUX

Architecture of UNIX

#### Commands:

pwd, ls, cd, cat, cp, mv, mkdir, rm, rmdir, touch, locate, find, grep, sudo, df, du, head, tail, diff, tar, chmod, chown, jobs, kill, ping, ssh, useradd, sshpass, scp, ifconfig, apt-get, curl, wget, uname, top, history, man, echo, zip, unzip, hostname, useradd, userdel, man, help, uname, apt-get, chmod, who, whoami, cal, date, exit.



# Module 8

## Linux



WICULTY

### Chapter-2: Infrastructure setup - Cloud Basics, Local VM & AWS EC2 Instance Creation

#### Concepts & Practicals

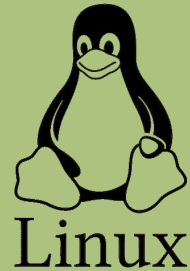
Creating Virtual Machine using Oracle Virtualbox

AWS free-tier account creation

EC2 Instance creation and setup

Accessing EC2 Instance using .pem & ppk keys

SSH command



# Module 8

## Linux



WICULTY

### Chapter-3: File System Of Linux Concepts & Practicals

Ordinary Files

Directory Files

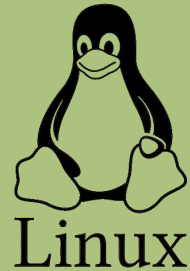
Device Files

The Structure of UNIX File system

**Job oriented: Hands-On preparation**

Real-time Use-cases & Applying the commands to solve different problems

Interview Questions discussion





# Module 8

## Linux



WICULTY

### Chapter-4: Vim Editor Concepts & Practicals

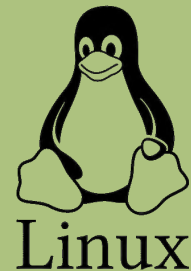
Editing- Insert, Append, substitute, open new line, replace  
Navigating - Cursor movement shortcuts  
Revisiting Editing - cut, copy, paste, undo, redo  
Searching - settings, search in current and multiple files  
Creating a file, view read-only, edit file

### Chapter-5: Working with Files & Directories Concepts & Practicals

Redirecting Output  
Redirecting Input  
STDIN, STDOUT, STDERR  
Commands: cat, touch, rm, cp, mv, ln, wc, mkdir, cd, rmdir,  
rm, ls, ls with options, find, locate, chmod, chown  
Mounting directories (mount, unmount)  
File & Directory Permissions (chmod, chown)

### Job oriented: Hands-On preparation

Real-time Use-cases & Applying the commands to solve  
different problems  
Interview Questions discussion



# Module 8

## Linux



WICULTY

### Chapter 6: Filters, Piping, Links, Archiving & Disk Utils

#### Concepts & Practicals

Hard Link

Soft link or Symbolic Link

Unlink

Filters: tr, tee, sed, pg, more, less, head, tail, paste, cut, sort, grep, egrep

Usage of piping

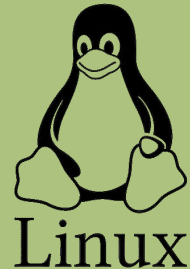
Archive: gzip, gunzip, tar, zip, unzip with different options

Disk usage, File/Dir size (df, du)

#### Job oriented: Hands-On preparation

Real-time Use-cases & Applying the commands to solve different problems

Interview Questions discussion



# Module 8

## Linux



WICULTY

### Chapter-7: Process & Job Scheduling

#### Concepts & Practicals

Foreground jobs

Background jobs

Killing jobs

Nohup

Process hierarchy (ps, kill)

Configuring Crontab

#### Job oriented: Hands-On preparation

Real-time Use-cases & Applying the commands to solve different problems

Interview Questions discussion

### Chapter-8: Working with remote

#### Concepts & Practicals

SSH - Configurations & Connecting to remote machines

SCP - copying the files/directories to remote

sshpass

Curl

Ssh-keygen

#### Job oriented: Hands-On preparation

Real-time Use-cases & Applying the commands to solve different problems



# Module 8

## Linux



WICULTY

### Chapter-9: Administration & System Performance & Logging

#### Concepts & Practicals

Types of accounts in Unix system

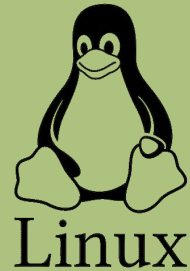
Create, Modify, Delete a Group

Create, Delete, Modify an User Account

CPU, Memory, Disk space Check

Performance tools (netstat, uptime, time, top)

System Logging



# Module 9

## Shell Scripting

### Self-Paced course coverage

#### Chapter-1: Shell Scripting - The Big Picture

What is Scripting

Why Shell Scripting

What is Shell, Shell types

What is Shell interpreter

She-bang & Script execution

Executing shell scripts

Debugging

Shell script permissions

Prerequisites - Linux Commands

#### Chapter-2: Shell Variables, Comments

##### Concepts & Practicals

Defining a variable and accessing values

Variable names

Special variables

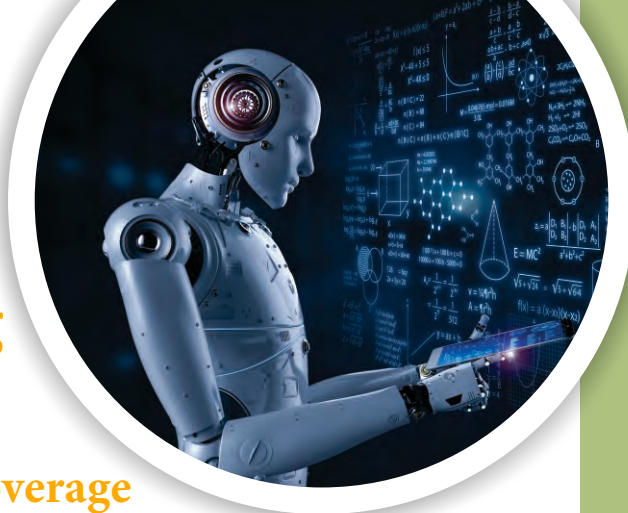
Read-only variables

Unsetting a variable

Command-line arguments

Special parameters

Exit status



WICULTY



# Module 9 Shell Scripting

WICULTY

## Chapter-3: Operators

### Concepts & Practicals

Arithmetic Operators

Relational Operators

Boolean Operators

String Operators

File Test Operators

## Chapter-4: Decommission Making & Loops

### Concepts & Practicals

The if...else statements

The case...esac Statement

The while loop

The for loop

The until loop

The break & continue statements

## Chapter-5: Functions & Arrays

### Concepts & Practicals

Creating functions

Passing parameters

Returning values

Function call from prompt

Defining Array values

Accessing Array values

# Module 9 Shell Scripting

WICULTY

## Chapter-6: Misc: Shell substitution, Escaping, I/O redirection, Here-doc Concepts & Practicals

Shell Command substitution \$(CMD)

Escape Sequences

Input & Output redirection

Here Document





# WICULTY

Join our Industry Woven DevOps Training certification course Now!

Visit us at : [www.wiculty.com](http://www.wiculty.com)