**DevOps Training Session**

**Purpose:**

The purpose of this training is to train the resources for DevOps Track, especially for Azure DevOps.

* Improve the individual’s level of Technology Awareness
* Increase an individual’s skills in one or more areas of expertise
* Increase an individual’s motivation to perform their job well and build their confidence.

**Training Topics with Course Outline**

The overall list of Trainings and their respective course outline are described in detail below.

**DevOps:**

* What is DevOps?
* Why Organization need DevOps?
* SDLC
* Agile Methodology
* DevOps stages

**Azure:**

* Introduction of Azure
* Basic Knowledge of most used services
* How to create Virtual Machines (RDP & Linux)

**Linux:**

* Linux Flavors basic knowledge
* Basic Commands of Ubuntu
* Permission Scheme of Ubuntu
* Linux Directory Structure
* Add users/Add Groups
* Putty/MobaXterm Tool

**Source Control Version (SCV)  
Git - GitHub:**

* What is git?
* What is version control?
* Git commands
* Sign up in GitHub
* Using git in the local machine
* Git installs
* Getting a code editor (Visual Code Editor)
* Inside VS Code
* Cloning through VS Code
* git commit command
* git add command
* committing
* git push command
* SSH Keys
* Git push
* Compare between GitHub workflow and local git workflow
* git branching
* Undoing in git
* Forking in git
* Git Branching Strategy
  + Trunk Based
  + Git Flow

**Docker:**

* What is Docker?
* Install Docker
* Docker Commands
* Environment Variables
* Images
* CMD vs ENTRYPOINT
* Networking
* Storage
* Compose
* Registry
* Engine
* Docker on Windows
* Container Orchestration
* Docker Swarm
* Dockerfile for NodeJs
* Docker-Compose file of NodeJs (Frontend + Backend)
* Docker Local Registry Setup and Push Images from Jenkins on local and on DockerHub.
* DockerHub

**Jenkins:**

* Installation of Jenkins on Linux, and in Docker Container
* Free-style Jobs
* Scripted and Declarative Pipeline
* Connection with Git using SSH and PAT
* Jenkins credentials and how to utilize them in the pipeline
* Jenkins Build Slaves as Docker container
* Build .Net Application (free-style + pipeline)
* Build Java Application through Maven in pipeline
* Build NodeJS Application (free-style + pipeline)
* Code coverage reports using Junit/Cobertura
* Build Parameterized Jobs with logic (using when expressions)
* Implemented rollbacks using Git tags and commits
* Created Parallel Stages in BlueOcean UI
* Multiple agents in stages
* Post Conditions for notification upon success/failed build stages
* Usage of Webhooks to trigger the Jenkins pipeline
* Checkout with Multi-Repos and checkout specific folder in the specifically created folder

**Azure DevOps**

* Provision and install a self-hosted agent for job execution
* Azure Pipelines build and deploy all the applications (Java, NodeJS, Python, PHP and .net)
* Publishing images on ACR and deploying applications on AKS
* Configure build triggers for release
* Build all the applications mentioned above explicitly using yaml file

**Pipelines (CI/CD):**

* NodeJS Application
* Java Application (Maven & Springboot)
* DotNet Application
* PHP Application
* Python Application