GIT :

- Organization Expectations
- GIT HISTORY
- What is GIT ?
- Difference betwen Centralised VCS and Distibuted VCS
- Prerequisites
- Three phases in git
- Initialize Git on local
- Move changes from workingtree to Localrepo
    - git status
    - git add
    - git commit
- Untracked and Modified files
- Add only the modified changes to the stagging area
- Track the changes made in the history
    - git log
- Will git track folder ?
- Revert the changes from staggingarea , Workingtree & delete in Workingtree
    - RESET
    - checkout
    - RESET --hard
- Tracking the deleted files
- REVERT
- Head VS Detached Head
- Push the changes from local repo to remote repo
- High level view of github
- GitHub Signup
- Create a repositry in the github
- Configure RemoteRepo to the LocalRepo to push the code.
- PersonalAccessTokens
- CLONE
- PULL & FETCH
- Difference between CLONE vs PULL
- Multiple branching
    - Create branch
    - List local branches
    - List remote branches
    - Switch b/w branches
    - Create and switch to branch at sametime
    - Push the local branches code to remote repo
- MERGE
- Merge conflicts
- Fastforward merge
- Rebase

- cherry-pick
- Delete a Branch in Local/Remote
- Branching strategy
- Fork
- stash
- Blame

Ansible :

- Basic devops pipeline
- Ansible intro
- Configuration management
- Push & pull type CM
- Architecture of ansible
- Scenario-1 : LAB SETUP of ANSIBLE
- Inventory in Ansible
- Ansible ping b/w ACS & Node
- Using customized file for inventory
- How to write ansible playbooks
- Playbook Syntax
- SCENARIO-2: Write a playbook to install tomcat & Java to configure servers
- Playbook vc Adhoc commands
- SCENARIO -3 : Run the playbook on the specific server in inventory file
- Reuse ansible playbooks by using variables
- SCENARIO-4 : sample deployment of apache and php modules on ubuntu
- SCENARIO-5 : sample deployment of apache and php modules on redhat/centos
- Ansible facts
- Ansible conditionals
- Fail Module
- Package Module
- If you didnt find any module / unable to get the exact modules.
- Handler in Ansible
- Scenario-6: Deployment using ansible playbook.
- Reusability of ansible playbooks
- IMPORT/INCLUDE in ansible playbooks
- Exercise-2
- ANSIBLE ROLES & Structure of Role
- Ansible-Galaxy & use the Ansible role from ansible galaxy
- Jinja template
- How to create a Ansible role
- Use the Ansible role created

Docker :

- Basic devops pipeline

- Understanding the Docker tool usage in pieline
- VM vs Containers
- Understand Container
- Advantages of using docker over VM
- Install Docker
- Container creation understanding
- Dockerhub
- Docker playground
- Hello-world container creation
- Docker workflow
- Dockerfile syntax
- Dockerfile for the java image
- FROM & RUN intructors
- Exercise: Install docker on EC2
- ENTRYPOINT , CMD , EXPOSE , LABEL , ADD , COPY
- Run container in multiplemodes
- Attached Mode
- Detached Mode
- Interactive
- Portforwarding
- publish on specified port
- Install docker on AWS EC2 server
- SAMPLEWAR Application deployment Using Docker
- Naming container
- Inspect a image
- Delete Image
- Delete Container
- Instructions: USER , WORKDIR, ARG, ENV,
- CMD & ENTRYPOINT Override
- Container states

K8S :

- K8s Intro
- Why you need Kubernetes
- Basic architecture of k8s
- Kubernetes componenets
- Steps to be followed for k8s cluster using kubeadm:
    - Take 3 VM's from AWS , having atleat 2 GB RAM
    - Install container runtime on all the nodes REFER HERE
    - Install kubeadm,kubectl & kubelet on all nodes REFER HERE
    - Configure Master
    - Configure Network
    - Join Worker Nodes
- Working with k8s
- API-REFERENCE

- Understanding POD
- Creating a Pod
- CONTROLLERS
- ReplicationController
- ReplicaSet
- Imperative vs Declarative way of working with K8s
- Practice of below topics: * Creating a Pod * CONTROLLERS * ReplicationController * ReplicaSet * Imperative vs Declarative way of working with K8s
- DaemonSet
- Jobs
- CronJobs
- Cronsyntax
- Service
- Types of service
- ClusteIp
- NodePort
- StatefulSet
- K8S storages
- Persistent volumes
- Static Provisioning
- Dynamic Provisioning
- Access modes to the volumes
- Reclaim policy
- EKS cluster creation
- Storage class
- Peristent volume claim(PVC)
- Attach a PVC to a pod
- Deployment
- Config Map and secrets
- Namespace
- Labels and Annotations
- Horizontal pod autoscaling
- Taints and Tolerations , NodeAffinity

Zabbix :

- Zabbix intro
- Basic architecture of zabbix
- ZABBIX-SERVER-INSTALLATION
- ZABBIX-AGENT-INSTALLATION
- Zabbix Dashboard components

Terraform :

- Terraform Intro

- Need for Infrastructure Provisioning
- Understanding of Terraform/Terminology
- How to install terraform
- Authenticate Terraform to speak with AWS
- Install Terraform on ubuntu
- Authenticate Terraform to speak with AWS
- Provider syntax , Resource syntax & Agruments and Attributes
- Terraform template for AWS provider
- AWS VPC & Subnet creation Manual & Terrafrom template
- Count in terraform
- Length in terraform
- depends_on in terraform
- Security groups in AWS
- Security group creation from AWS dashboard
- Security group creation using Terraform template
- S3 in AWS
- S3 creation in AWS dashboard
- S3 creatin using terraform template
- Terraform plan
- Target in terraform
- Terraform Registry and using module in terraform registry
- Difference between terraform module and standard terrafrom template
- Creating a module in terraform
- Complete VPC setup for an application
- EC2 instances creation in a customised VPC, subnet & keyname
- Terraform template to create ec2 in customized VPC ,Subnet, Keypair & SecurityGroup
- Provisioners in terrafrom
- Terraform provisioning using remote-exec

SonarQube :

- Sonarqube Intro
- Sonarqube Architecture
- Pipeline integrated with Sonarqube
- Install & Configure Sonarqube
- Understanding of SonarQube & Dashboard
- Integrate Sonarqube with Jenkins
- Default Quality Profile VS customized quality profile
- Create a Pipeline job with sonar integrated script and run it
- Use Customised quality profile in Jenkins Job.

Jfrog :

- JFrog Artifactoy & Pipeline integrated with Jfrog
- Install JFrog Artifactory oss

- Create a repository
- Integrate Jfrog with the Jenkins
- Create a Maven-job to store our artifacts to the jfog artifactory
- Upload the artifacts to jfrog in Declarative-pipeline

Maven :

- Maven Intro
- Creating a VM in AWS & connect using Mobaxterm
- Installing maven
- Maven Goals
- Maven Repositories
- Build the code with Maven and look at multiple goals
- POM.xml

Jenkins :

- Jenkins Intro with CI/CD and pipeline understanding.
- Creating a VM in AWS & connect using Mobaxterm
- Install Jenkins
- Create a job in jenkins .
- Explore multiple options available in Jenkins Job.
- Explore Manage jenkins options.
- Scenario-1.
- Scenario-2.
- Understand the build process in Jenkins & Explore Jenkins Home directory.
- Build with parameters.
- Scenario-3 - Install & Configure ***Git Parameter*** to pass the branch paramater while running the Job .
- Scenario-4 - Create a NewJob SMP-2 , it should be copied from the existing SMP Job
- Upstream and Downstream projects.
- Scenario-5 - Configure the Upstream & Downstream as per below requirement .
    - Upstream Job : Build a ***SMP*** Job from Dev Branch
    - Downstream Job: Once the **SMP** Job is sucessfull , it has to build the Downstream Job(SMP-2) with Master Branch.
- Jenkins Node/slave/Agent
- Here are some use cases of Jenkins nodes
- Understanding Pipeline with Jenkins Node:
- Configure a Jenkins Node and attach it to the Jenkins Master
- Use node/slave in the jenkins job.
- Backup of Jenkins
- Pipeline Job in jenkins , create a pipeline job
- Basic syntax of groovy for pipeline Job
- Scenario-6 : Jenkins Scripted Pipeline JOB - Get the code from Github(SMP) , build using Maven, archive the artifacts & publish the junit reports -- Run job on jenkins node

- Practice session for students : Jenkins Scripted Pipeline JOB - Get the code from Github(SMP) , build using Maven, archive the artifacts & publish the junit reports -- Run job on Jenkins Master
- Tracking the Configuration changes in the Jenkins Job
- Jenkinsfile (Declarative Pipeline)
- Scripted pipeline vs declarative pipeline Syntax differences
- SCENARIO-7: Jenkins Declarative Pipeline JOB -- Get the code from Github(SMP) , build using Maven, archive the artifacts & publish the junit reports - Run job on jenkins master
- Blue ocean plugin
- SNAPSHOT vs RELEASE
- Scenario-8 : Create a Declarative pipeline job as per below requirement
    - Automatically trigger the Jenkins Declarative pipeline Job when there are changes in Code Repo.
    - Jenkins job should get the code , Build the code , Archive the artifacts & Publish Junit test results.
- Scenario-9 : Create a Declarative pipeline with below requirement
    - Upstream Job : Build a ***SMP_pipelinejob*** Job
    - Downstream Job: Once the **SMP_pipelinejob** Job is sucessfull , it has to build the Downstream Job(SMP_DeclarativePipelineJob)
- Scenario-10 : Parameters using in Jenkins Declarative pipeline
- Scenario-11 : Trigger a Jenkins declarative job using git parameter
- JENKINS Built-in environment variables

Basic Sessions :

- Chocolatey Package Manager Installation.
- Mobaxterm
- Git
- VisualStudioCode
- AWS
- AWS Free-Tier Account creation
- What is linux ?
- OS Working
- Components of OS
- Basic linux commands
- ConnectionsTt
- Onpremise vs cloud
- Create a EC2 in AWS
- Connect to server from your local laptop
- Stop a server
- Destroy a server
- Instance Type
- Inbound Rules
- Linux Basics
- Multiple ways to Authenticate VM(remote server)
- VM creation in AWS / connect / disconnect / Terminate/stop/Reboot/change Instance type/Cost of EC2