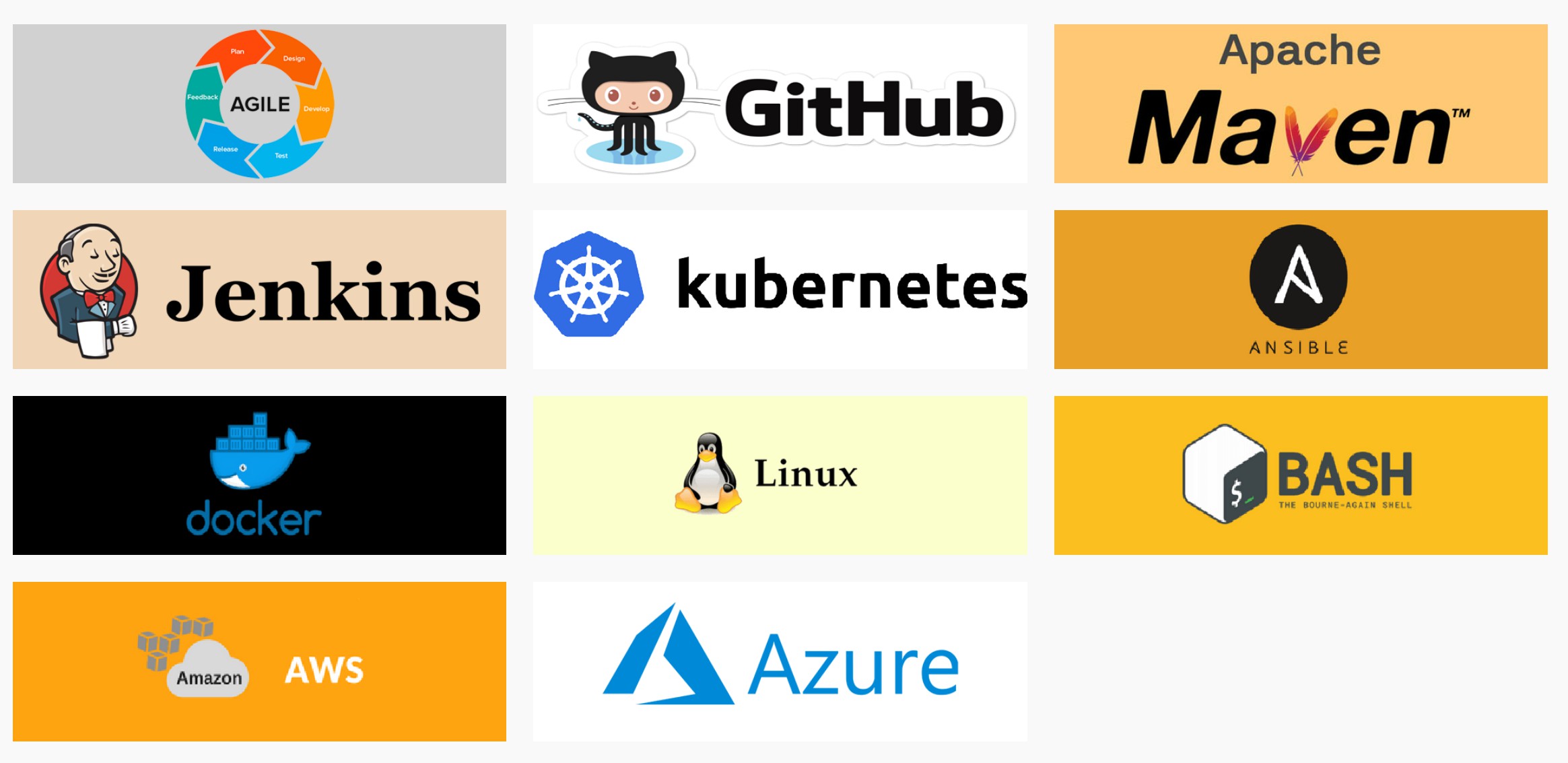


**Skills Covered**



DevOps & Multi-Cloud

# DevOps with AWS & Azure Course Content

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 is Agile and Scrum?

* Agile Development Process
* Agile Manifesto
* Agile Scrum Work Flow
* Agile Analysis Estimation Techniques
* Types of Roles and Responsibilities
* Problem That DevOps Solves
* Making a DevOps Transition
* Introduction to DevOps Automation
* Role of Dev in Agile Scrum
* Role of Ops in Agile Scrum

Problem which resolves DevOps Implementation in Project Implementation



DevOps Life Cycle

* Introduction to DevOps Automation
* Tools
* DevOps Technology Categories
* Collaboration
* Planning
* Issue Tracking
* Monitoring
* Configuration Management
* Source Control
* Dev Environments
* Continuous Integration
* Continuous Testing
* Continuous Deployment

Linux Essentials for DevOps

* Linux Overview
* What is Operating system
* What is Unix, Linux
* Unix vs Linux
* Linux Distributions
* Linux Architecture & Installation
* Linux Boot process
* File system management
* User management
* Group management
* Package management
* Disk & Volume management
* Linux volume manager hands on
* Process management
* Scheduling and batch automation



* Editors
* VMware overview

Virtualization

* what is virtualization
* Brief explation on hyperviser
* The difference between local and virtual servers

Shell / BASH Scripting

* Role of Shells in Linux Environment
* Types of shells
* Shell Commands
* Command line arguments
* Variables
* Types of Operators
* Conditional Statements
* Bash Loops
* Case statement
* Functions
* Interactive Scripts
* Awk
* Sed
* Adv. Script Programs
* **Implementing shell scripts based on real world scenarios**
* Introduction to Python

PYTHON

* Variables,Data types,Input/Output,comments,Docstrings,Typecasting
* Control structures:if,else,elif,for,while,break,continue,pass
* Data Structures:Strings,lists,tupples,sets,dictionaries
* Functions,Modules & Packages
* Error Handling
* Iterators & Generators
* Introduction to OOPs

GIT

* Introduction
* What is a Version Control System (VCS)? Distributed Vs Non-distributed VCS
* What is Git and where did it come from?`
* Alternatives to Git
* Installation and Configuration
* Obtaining Git Installing
* Git Working with local repo-Git
* Git commands



* Updating the remote repository from the local (git push) Updating the local repository from the remote (git pull)
* Tagging in Git What are Git Tags? Listing tags Lightweight tags
* Displaying tag details (tag show) Annotated tags
* Checking out tags Pushing tags Pulling tags
* Branching in Git
* What is a branch
* A note about <HEAD> Listing branches Create new branch Checkout branch Pushing branches Pulling branches Merging in Git
* Fetching Changes (git fetch) Rebasing (git rebase)
* Git Workflows Different ways of using Git Centralized
* Feature Branch Gitflow Workflow Forking Workflow
* Advanced Branching & Merging
* Deleting a Branch Fast forward merge Three-way merge
* Web hook integration
* Handling Merge Conflicts
* Git hub actions
* **Implementing git hub actions on various workflows**
* **Implementing workflows using Git Lab**



MAVEN (Build Tool)

* Issues before in manual process of build process
* Automated build process
* Introduction
* Maven Structure and Installation
* Maven Life Cycle
* Maven Dependencies
* Maven Repositories
* Maven Plug-ins
* Maven Configuration
* Integration with SCM tools
* Maven Project
* Integrating Maven for Code quality checks,SAST
* Integrating maven with sonar Qube,Nexus,Jfrog
* Deployment,Re-Deployment,Un-Deployment using Maven with Tomcat
* What is Grade

GRADLE (Build Tool)

* Installation
* Life Cycle, Tasks
* Build.gradle file
* Plugins in Gradle
* Gradle properties
* Custom tasks
* Build scans & Debugging
* Deployment

COMPLETE END-TO-END FLOW ON CI

* Tools On CI:GIT,GIT HUB,MAVEN,GRADLE,JUNIT,SONARQUBE,NEXUS,JFROG

# CI & CD Servers

Jenkins

* What is Jenkins?
* Best Practices
* Installation and Configuration
  + Pre-requisites
  + Download & Install
  + Configurations
* Jenkins plugins – how to download and use
* Parameterizing the build
* Overview of Continuous Integration (CI)
* What it means Continuous Integration? Fundamental of CI
* How CI helps to Agile Development History of Jenkins
* Where Jenkins Fit in Organization Overview of Jenkins community
* Install Jenkins on Ubuntu / Windows Configuring a Node
* Configuring Jenkins server
* Configure Dashboard Configure System Environment Global Properties
* Configure Build Tools Configure Proxy
* Working with Jenkins Build Job



* Create and Configure a job Run a job manually Triggering a Build Scheduled Build job Manual Build job
* Polling SCM
* Maven and ANT Build Step Execute a Shell
* Post-Build Actions Archiving Build Results Notifications
* Working with Automate Testing
* Advanced Jenkins
* File fingerprint tracking Parameterized Build Job Parameterized Trigger
* Automated Deployment and Continuous Delivery
* Introduction to GROOVY
* CI & CD Pipeline Deployment using pipeline Script
* Jenkins Plugins
* Master & Slave
* Jenkins administration
* Overview of Notification Email Notification
* Best Practices on Jenkins

# Configuration Management Tools

ANSIBLE (Configuration Management Tool)

* IT Automation
  + History of IT Automation
  + Advantages of IT Automation
  + Disadvantages of IT Automation
  + Types of IT Automation
* What is Ansible?
* Ansible Architecture
* Installing Ansible
  + Installing Ansible on Linux OS
  + Installing Ansible using the systems package manager
* Ansible Version and Configuration
* Working with inventory files
  + Basic inventory file
  + Groups in an inventory file



* + Regular expression in the inventory file
* Automating Simple Tasks
* YAML Scripting
* Working with Playbooks
  + Anatomy of a playbook
  + Playbook commands
  + Writing Playbooks
  + Executing the Playbooks
  + Variables in Playbooks
  + Terminology in Playbooks
* Ansible Core Modules
* Ansible Ad-hoc commands
* Installing and configuring a web server
* Working with Handlers
* Ansible Roles
* Ansible Galaxy
* Introduction to AAP(Ansible Automation Platform)

Note: Realtime Use Case in Ansible integrate with Jenkins,Git and Maven for Deployment

DOCKER (Containerization Tool)

* Containerization Vs Virtualization
  + Traditional Virtualization
  + Containerization
* Understanding Docker
  + Difference between Docker and Other VMs
  + Docker file
  + Docker Networking
* Docker Installation
* Docker Hub and expose to official images
* Docker Images registry
* Running the Docker Container
* Handling Docker Containers
* Docker Adv.Commands
* Docker Terminology
* Working with Docker Images



* + Docker Hub
  + Searching Docker images

Docker file build instructions

* + FROM instruction
  + MAINTAINER instruction
  + COPY instruction
  + ADD instruction
  + RUN instruction
  + ENV instruction
  + ARG instruction
  + Environment variables
  + USER instruction
  + WORKDIR instruction
  + VOLUME instruction
  + CMD instruction
  + ENTRYPOINT instruction
  + SHELL instruction
* A Brief on the Docker image management
* Publish your build images into Docker Hub
* Understanding the Docker Hub
* Working with Containers
  + What is container
  + Docker run command
  + Theory of pulling and Running Containers
  + Container Life cycle
* Data Volume
* Sharing data between Containers
* Docker Swarm Mode
* Swarm Mode Theory
  + Configuring Swarm Mode
  + Services
  + Scaling Services
  + Rolling Updates
* Introduction to image security-Trivy

Note: Docker Real Time Use Cases



KUBERNETES

* What is kubernetes
* Purpose of Kubernetes for micro services
* How kubernetes works
* Master components, how works
* Node Components, how works
* How pods works
* Installations and configuration kubernetes cluster
* Introduction to name space
* Pod lifecycle
* Work with pods
* Replica sets, Liveness probes, Readiness probes
* Config maps & Secret Management
* Storage Management – PV,PVC
* Deployments,Statefullsets
* Services
* Networking and Service discovery
* Security & RBAC
* Logging,Monitoring,Debuging – Grafana,Prometheus,ELK/EFK
* Helm and packaging
* Blue/green deployments with real time examples
  + What is blue and green deployments
  + How it helps in real time
  + Overview of Blue/Green Deployments
  + Implementation Strategies
  + Benefits and challenges
* CI/CD & GitOps using ArgoCD
* **Micro services Deployments using cloud-native Deployment techniques**

Provisioning using Terraform

Goal: Learn how to provision and manage infrastructure on a Cloud Platform (AWS) using Terraform Configuration Files.

Objectives

After completing this module, you should be able to

* Understand Provisioning using Terraform
* Learn the Difference between Terraform vs Ansible
* Understand Terraform Architecture
* Deploy a Terraform Configuration File
* Use Basic Terraform Commands
* Manage Terraform Resources
* Perform Terraform State Commands



Topics

* Introduction to Terraform
* Terraform vs Ansible
* Terraform Architecture
* Terraform Configuration
* Terraform Common Commands
* Managing Terraform Resources
* Terraform State
* Terraform Modules
* Introduction to HCL

Hands-On

* Setting Up AWS and Terraform
* Executing a Terraform Configuration
* Managing Terraform Resources
* Referencing Terraform Resources
* Terraform State commands
* **Implementing server less deployments using Terraform**

# **AWS**

Introduction to Cloud Computing

* What is Cloud
* Why Cloud?
* Types of Cloud Deployment Models
* Types of Cloud Services
* Future of Cloud Technologies
* Advantages and Disadvantages of Cloud

Introduction to Amazon Web Services (AWS)

* What is AWS?
* How to Subscribe for AWS account
* What is the AWS Free Usage Tier
* AWS Certification
* Introduction to the AWS management Console
* List of services given by AWS



Elastic Compute Cloud (EC2)

* What is Amazon EC2?
* Features of Amazon EC2
* Managing the EC2 infrastructure
* EC2 Dashboard
* Pricing for Amazon EC2

Regions and Availability Zone Concepts

* Describing Regions
* Availability Zones, and Endpoints
* Managing instances in an Availability Zone

Amazon Machine Images (AMI)

* Managing AMIs
* Working with Windows, Linux AMIs
* Shared and Paid AMI
* Making an AMI Public

EC2 Instances

* Instance Type
* Instance life cycle
* Differences between reboot, stop, and terminate
* Building an EC2 windows and linux instances
* To install instance in public and private subnet
* Security via Key Pairs
* EC2 Class and VPC Security Groups
* Managing Elastic IP's
* Pricing model in EC2 instances
* EC2 with Amazon command line interface



Amazon Elastic Block Store (EBS)

* Features of Amazon EBS
* Amazon EBS volumes
* Managing EBS volumes
* Increasing the volume size
* AmazonEBS snapshots

Load Balancing (ELB)

* Creating a load balancer
* Internal and external load balancer
* Load balancing protocols
* Security groups for the load balancer
* Health check for the load balancer
* Cross-zone load balancing
* Connection Draining

Auto Scaling

* What is auto scaling?
* Auto scaling components

Advantages of Auto Scaling

* Creation of launch configuration
* Configuration of auto scaling policies
* Advantages of using auto scaling with ELB

Network & Security

* Security Groups
* Elastic IPs
* Placement Groups
* Key Pairs
* Network Interfaces



# Networking Services

Amazon Virtual Private Cloud (VPC)

* What is Amazon VPC?
* VPC Essentials
* Default and Nondefault VPC
* VPC Networking and ACL
* Security Groups
* DNS and DHCP Options Sets
* VPC Peering and Endpoints
* Subnet Routing
* VPC Internet Gateway
* Elastic IP addresses and network interfaces
* VPC integration with many other AWS services
* Creating a NAT instance in a VPC
* Configuring a Web application in VPC
* Pricing for Amazon VPC

Amazon Route 53

* Route 53 as your DNS service
* Using Traffic Flow
* Route 53 Health Checks
* Configuring DNS Failover
* Latency Based Routing
* Weighted Routing Policies
* Hosting web portal using Route53
* Bucket Policies



# Security & Identity Services

Identity Access Management (IAM)

* IAM Features
* Getting Started With IAM
* Creation of user, groups, roles
* Managing & Writing policies
* Credential Report
* IAM Console and the Sign-in Page

# Storage & Content Delivery Services

Amazon S3

* What is object Storage?
* Data as objects
* Lifecycles of S3
* Managing Buckets
* Accessing S3 storage via tools
* Creation of a static website using S3 storage

# Database Services

Relational Database Service (RDS)

* RDS Essentials
* Launching RDS instance
* Selecting the Engine
* Configuring the Database Engine
* Managing RDS Database
* Setting up automatic backups
* Authorizing access to the DB



Amazon Cloud Watch

* Amazon Cloud Watch Architecture
* List of services monitored by Cloud Watch
* Collect and track metrics
* Monitoring memory and disk Metrics
* Monitoring logs, Graphs
* Set Alarms

Amazon Security Groups and NACL

* What is Security Group?
* Where the Security Groups are used in AWS?
* What is NACL?
* Difference between NACL and Security Groups
* Implementation of Security Groups and NACL service

Cloud Formation

* Building AWS infrastructure as a code
* Design a template
* Create a Stack
* Create a Template from your Existing Resources
* Introduction to JSON

# Application Services

Amazon Simple Email Service (SES)

* Simple email service overview
* Configuring Amazon email service
* Amazon SES and Deliverability
* Amazon SES Email-Sending Process
* Email format and Limits of SES



Amazon Simple Queue Service (SQS)

* Simple Queue service overview
* SQS for background work task
* Creating a Queue
* Confirming the Queue exists
* Add a permission to the Queue

Amazon Simple Notification Service (SNS)

* Simple Notification Service overview
* SNS architecture
* Publishers and subscribers
* Creation of a topic
* Subscribing to topic via Email
* Setting notification for EC2 instance changes
* Code commit

AWS-CI/CD PIPELINE SERVICES

* Code Build
* Code Deploy,Code Arttifacts
* Code Pipeline

**AWS PROJECT IMPLIMENTATION:Design,Implimenting and Deploying 3-tier Architecture**

* Implementing Serverless deployment - Lambda

# Azure

What is Microsoft Azure?

Types of Azure Clouds

* Azure as IaaS
* Azure as PaaS
* Azure As SaaS

Azure key Concepts

Azure Domains (Components)

* Compute
* Storage
* Azure Networking
* Database



Overview of DevOp

* Why DevOps?
* What is DevOps?
* DevOps Market Trends
* DevOps Engineer Skills
* DevOps Delivery Pipeline
* DevOps Ecosystem

Version Control with Git

* What is version control
* What is Git
* Why Git for your organization
* Install Git
* Common commands in Git
* Working with Remote Repositories

Azure DevOps CI/CD pipelines

* Introduction to CI/CD
* Tasks
* YAML Templates
* Create .NET Core CI pipeline
* Create .NET Core CD pipeline
* Sonar cloud integration

Implement and manage build infrastructure

* Private and hosted agents
* Integrate third party build systems
* Recommend strategy for concurrent pipelines
* Manage Azure pipeline configuration (e.g. agent queues, service endpoints, pools, webhooks)



Deploying ARM Template in Azure

* Introduction to CI-CD ARM templates
* Create ARM template
* Create and Run ARM deployment CI pipeline
* Create and Run ARM deployment CD pipeline

Application Secrets in the pipelines

* Introduction to Azure Key vault
* Accessing Secrets from Azure Key Vault
* Linking Secrets from Azure Key Vault

Azure Artifacts

* Introduction to Azure Artifacts
* Create NuGet packages and Versioning
* Package management with DevOps
* Maven packages

Continuous Integration using Jenkins

* Jenkins Management
* Adding a slave node to Jenkins
* Building Delivery Pipeline
* Pipeline as a Code

Implementation of Continuous Testing with Selenium

* Introduction to Selenium
* Why Selenium?
* Selenium – Webdriver
* Creating Test Cases in Selenium WebDriver (Waits)
* What and why X-Path
* Handling different controls on Webpage
* Framework in Selenium
* Selenium Integration with Jenkins
* Integrating Selenium with Jenkins
* What is DevSecOps

Introduction To DevSecOps

* Phases in DevSecOps
* Secure coding practices – OWASP Top 10
* SAST – Checkmarx,Fortify
* Dependency scanning & SCA-Snyk,OWASP Dependency-Check
* Secrets Management-Hashicorp Vault,AWS Secrets Management
* Container image security:Clair,Acqa security,Trivy
* IAC – Security:Checkov,Terrascan
* CI/CD Pipeline Security
* DAST – OWASP ZAP,Burp Suite,App scan



Continuous Deployment: Containerization with Docker

Containerization with Docker: Ecosystem andtworking

Configuration Management with Ansible

