

DevOps

Engineering Training

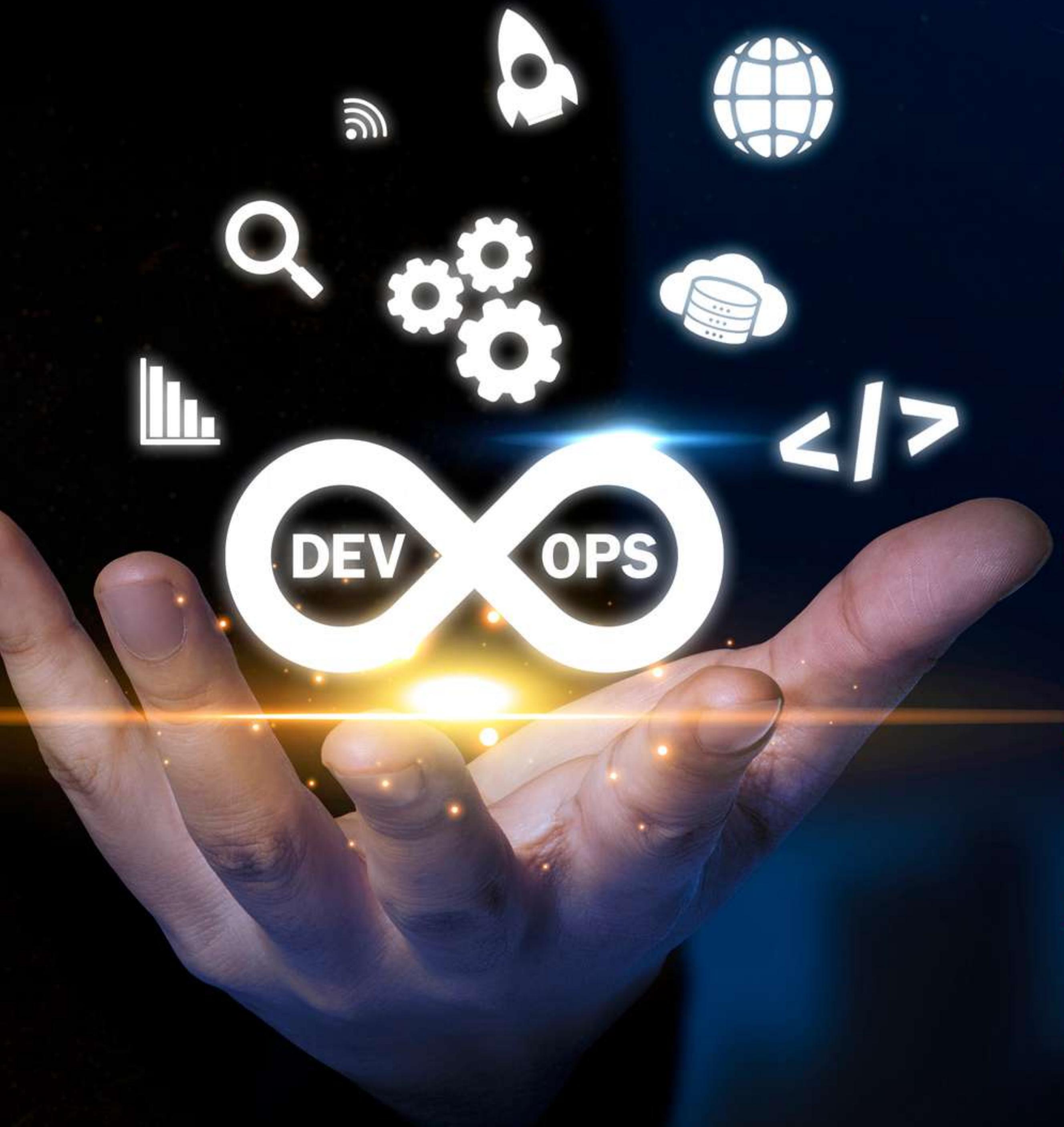
DevOps is a set of practices that combines software development and IT operations. It aims to provide continuous delivery of high-quality software and shorten the development life cycle of systems. Agile software development complements DevOps; The Agile method of working influenced many aspects of DevOps.



Contents

1 Walk with Us	P1
2 Training Process	P2
3 Unique at STEPS DevOps Engineering Training What is DevOps?	P3
4 What is DevOps?	P4
5 Who should learn DevOps?	P5
6 Module - 1	P6
7 Module - 2	P10
8 Module - 3	P16
9 End of Training	P20
10 Reasons Why You Would Love STEPS	P21

WALK WITH US



- Making the world of knowledge more easy with **STEPS**.
- Offering a best chance to enter the world of **DevOps**.
- **Learn** to manage Development and Operational Teams in Software Development Life Cycle.
- Be confident with first-hand experience with **STEPS** assignments in a real work environment to start working as **skilled professional**.
- **STEPS** brings to you all the key learning factors to make you a proficient **DevOps Engineer**.
- **We support** to make your resume ready as per IT Industry standards and to get interview calls.

Training Process

- The training starts with **Software Development Life Cycle** overview wherein you will learn the basics of Software Development that helps in building a strong base through practical sessions that we follow.
- Also you'll learn about **SCM** using **GIT**. It'll help you to organize and manage the source code effectively.
- Then, to provide hands-on-practical exposure to create software by your own using **Python/Java/JavaScript**.
- Once you have the basis in order, the training kicks starts to more advanced modules wherein you will learn **Software Testing using Selenium** and also Building tools like **Maven Gradle** etc.
- Next, **Linux Administration** sections starts and you will learn about how to install/configure linux server OS in various cloud/physical platforms like **AWS, AZURE** etc. and how to use containers like **Docker/Kubernetes**. Also learn how to automate these process using **Ancible**.
- Another interesting component in this training program is to workout **100+ assignments**.
- Next you will be move on to learning how to automate development and deployment cycles or **CI/CD pipeline** management using **Jenkins**. Then the final part, monitoring and analysis of the software using **Nagios, Prometheus and Grafana**.
- Now, the modules are completed. Our team of experts will train you to work as **DevOps Engineer in software development**. And also share tips and tricks for the interviews.
- Once your training is over and you are well versed practically with all the components you can gain advantage with our **Placement Assistance** that has helped many to find their dream jobs in the Software industry.



Unique at STEPS DevOps Programming

- Experienced developers under the guidance of CTO as trainers/guides who know the latest in the ever-changing **DevOps** industry.
- Provides with real-time projects and seminars to ensure you gain first hand experience of each theoretical/technological portions learned.
- Deals with 100+ assignments that helps you to build a strong base on DevOps. Provides interview orientation and great placement support from STEPS.
- Provides with online technical test that helps you to identify your areas of weakness, prepare and make you confident to attend interviews.
- You are at a real IT park in between 100s of working professionals to get a first hand feel of a workplace, not another typical college environment or shopping complex training institute.
- Get an experience equivalent certificate.

MOVE FORWARD TO EXPLORE THE SYLLABUS AND
SEE HOW YOU CAN GET TRANSFORMED FROM A
FRESHER TO A "**DevOps Engineer**"

What is DevOps?

DevOps is an approach to software development that emphasizes collaboration between software development and IT operations. The collaboration is intended to produce more "continuous delivery" of software by integrating different teams who are responsible for a part of the development cycle, including design, programming, testing and delivery. Additionally it also focuses on putting processes in place that result in shorter time-to-market for automation. The DevOps philosophy implies that the two teams need not be disjointed but instead can work closely together to increase speed and quality.



Who should learn DevOps?

In the not-too-distant future, developers won't need to care about backend infrastructure - that whole area of expertise will be encapsulated in the job of whoever can wear the title "**DevOps**". It's not an exaggeration to say that devops are going to change your life as a programmer.

If you are a programmer who likes to live in the basement, fix bugs, and make small improvements all day, we don't have good news for you. If you are a manager trying to keep your programmers happy, remember this rule: As soon as your company starts hiring the first DevOps people (which will probably happen within a couple of years), your team will start working harder and faster. If they're smart programmers, they'll be more interested in attacking business problems than in solving technical challenges. They'll likely be more productive than ever before, although their day-to-day jobs may be less intellectually challenging.



Module 1

Introduction To DevOps



1. Devops Lab Setup tools for Linux and windows Environment

- Git Bash installation and Git hub account setup
- Tomcat installation and Configuration
- Jfrog Artifactory installation and Configuration
- Maven Installation and Configuration
- Jenkins installation and Configuration
- Ansible Installation and Configuration
- Sonarqube installation and Configuration
- Docker Installation and configuration
- Java installation and Configuration
- Environmental variable setup for both windows and Linux

2. Introduction to Devops and Dev sec ops

- Introduction DevOps
- What is DevOps?
- SDLC models, Lean, ITIL, Agile
- Why DevOps?
- History of DevOps
- DevOps Stakeholders
- DevOps Goals
- Important terminology
- DevOps perspective



- DevOps and Agile
- DevOps Tools
- Configuration management
- Continuous Integration and Deployment

3. Introduction to Linux Administration

- Introduction to Linux Families (ex: Redhat)
- working with Dnf
- Manage Basic Storage

4. Installation and Initialization

- Installation, Package Selection
- Anatomy of a Kick start File, Command line
- Introduction Bash Shell
- System Initialization, Starting the Boot Process: GRUB

5. Boot and Package Management

- Securing single-user mode (su login)
- Shutting down and rebooting the system
- Scheduling future jobs
- DNF Command set, Install packages by using dnf.
- Enable DNF software repositories



6. User Administration

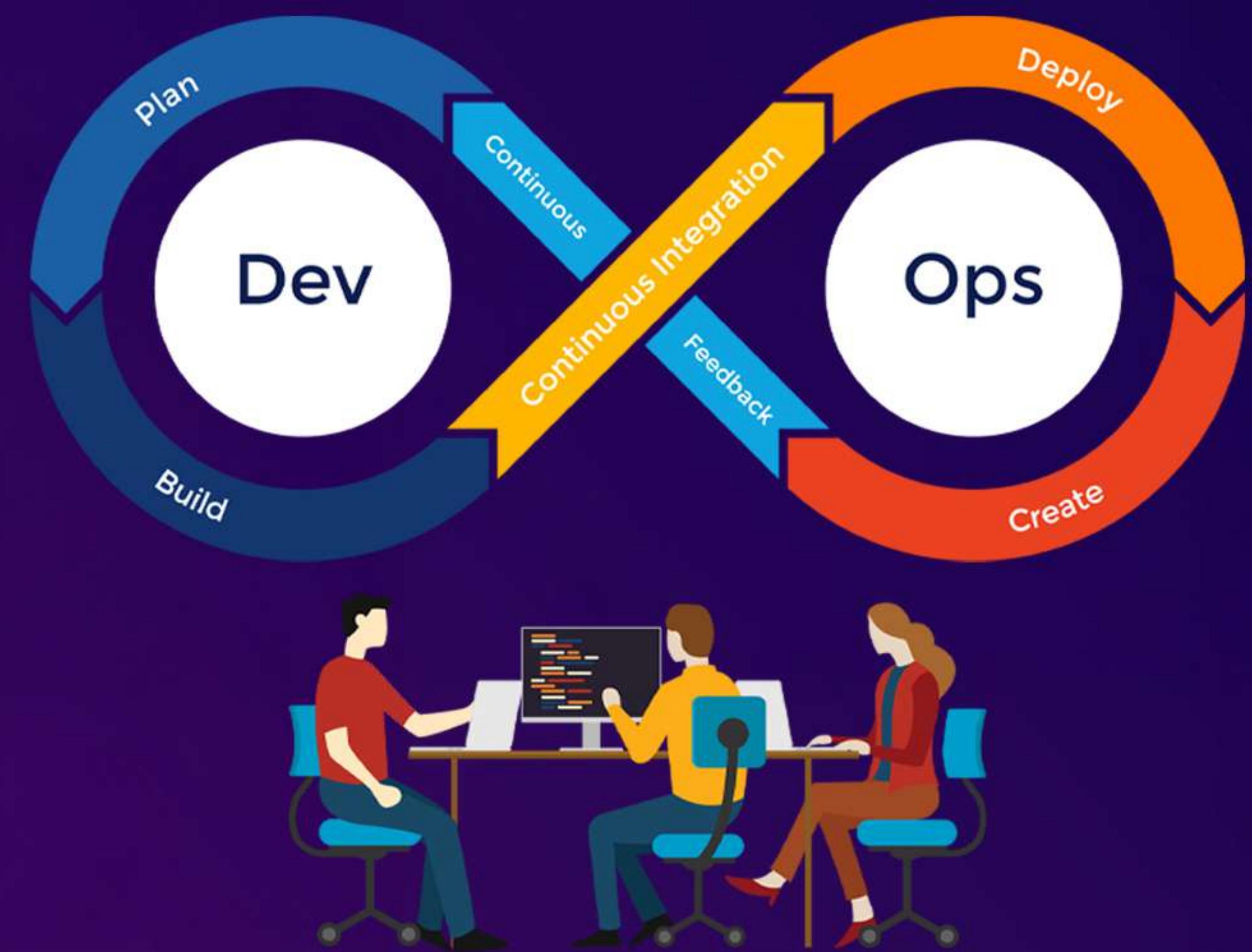
- Understanding different types of groups and creation of groups
- Creation of users in different groups
- Understanding Passwd, Shadow Files
- Understanding passwd aging
- Creation of quotas for users, groups and file systems
- Understanding users security files
- The different commands for Monitoring the users
- TROUBLESHOOTING
- Understanding the different filesystems in Linux
- Working with commands tar, find, grep, etc.



On Completing Module 1

After completing the first module, you will have a complete understanding of how a **Software Development team works**. The following concepts can be learned

- Linux Administration
- System Installation and Configuration
- Package management and Log management
- Backend development
- Virtualization and Containers



Module 2

Version control/ SCM



1. Run levels

- Understanding the different types of run-levels
- Understanding different types of shutdown commands
- Understanding run control scripts
- Managing Network Configuration

2. Introduction to Git

- Overview of SVN, GIT , Clear case , perforce & Comparision
- Introduction of Git
- Selecting Git Client
- Creating Repository
- Working with Tag
- Creating and Merging Branches
- Executing Git Commands
- Git Logs , Git stash, Git rebase
- Resolving merge conflict
- Git pull, clone, fetch



3. Basics of Python

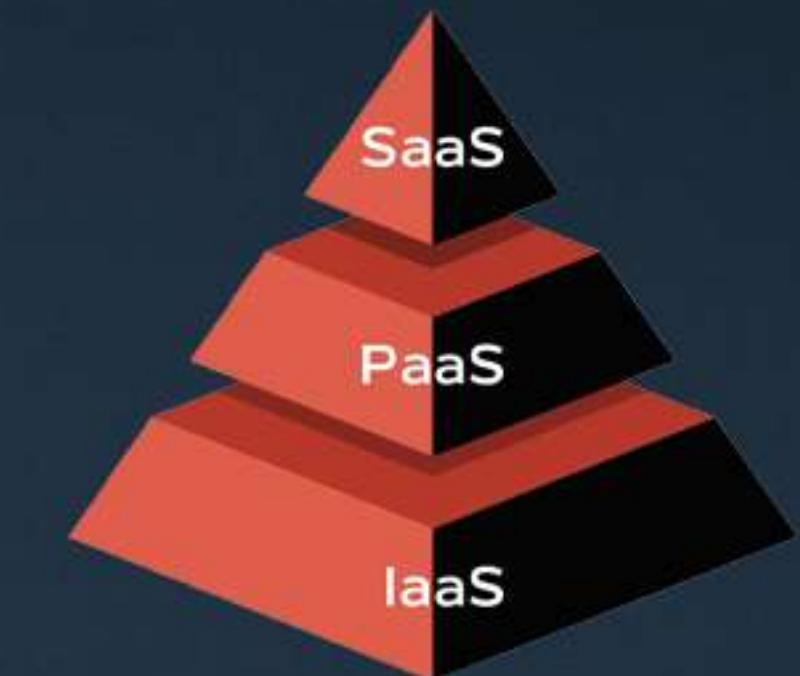
4. Introduction to Frontend Technologies

5. Introduction to Django



6. Introduction to Cloud Computing

- What is Cloud?
- Evolution of Cloud Computing
- IAAS (Infrastructure as a Service)
- SAAS (Software as a Service)
- PAAS (Platform as a Service)
- Private, Public and Hybrid Cloud
- Public Clouds



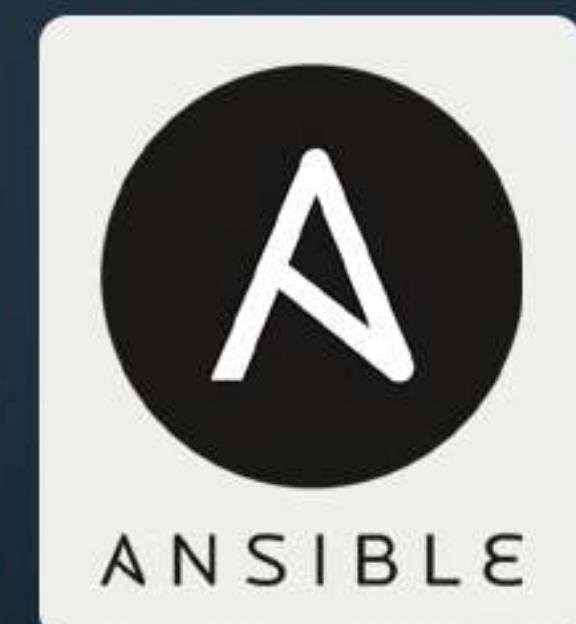
7. Introduction to Cloud Computing Platforms

- Amazon Web Services
- Microsoft Azure
- Google Cloud Services



8. Introduction to Ansible

- What is Ansible
- Change Management
- Provisioning with Ansible
- Benefits of using Ansible



9. Ansible Building blocks and Process flow

- Introduction Ansible Anatomy
- Ansible Requirements Specification
- Overview of Ansible Components
- Overview of Ansible Strategy

10. Ansible Playbook Modules and directory structure

- Introduction Ansible Playbook
- Introduction Ansible Modules
- Lab (Docs, setup, service, yum ...etc)

12. Variable, Facts and jinja templates

- Working with Ansible Variable
- Working with Facts
- Working with Jinja Template



13. Play and Playbooks

- Overview of Ansible Playbooks
- Playbook Language Example

14. Getting Started with Docker

- Working on Ansible Handlers
- Executing a Playbook.
- Introduction to Docker.
- What's under the hood - Namespaces, Cgroups and OverlayFS
- Understanding Virtualization
- Virtualization vs Container



15. Docker Installation

- Creating a Virtual Docker Host(CentOS) by using Vagrant
- Installing Docker on CentOS
- Introduction Docker namespaces

16. Docker Images

- Introduction Docker Images
- Building a Docker Image with a Dockerfile
- Sharing Data in Your Docker Host with Containers
- Sharing Data Between Containers
- Copying Data in and from Containers
- Creating Docker Hub Account.
- Building Images using DockerFile.
- Pull and Push Images From/TDocker Hub.

17. Docker Networking

- Introduction Docker Networking
- Finding the IP Address of a Container
- Setting Up a Custom Bridge Network for Docker

18. Container Operations

- Port Mapping for Docker
- Creating, Starting, Stopping, Renaming, Removing Containers
- Inspecting Containers
- Limiting Resources Memory and CPU
- Prioritizing CPU Utilization

20. Docker Compose

- Introduction Docker compose
- Creating Docker compose file
- Executing Docker Compose file

21. Introduction to Kubernetes

- The need for a Container Orchestration Engine
- Battles of COEs, which one choose
- Key Features of a COE.
- What makes Kubernetes the defactCOE choice.
- Negatives of using Kubernetes

22. Key Concepts of Kubernetes

- Namespaces
- Pods
- Replica Sets and Deployments
- Service Discovery and Load Balancing
- Configmaps, Storage, Network, RBAC
- Statefulsets, Cronos and Jobs
- Kubernetes Architecture



kubernetes

23. Setting up Environment

- Provisioning and configuring on AWS
- Initialise Cluster with Kubeadm
- Setting up Weave CNI
- Launching Kubernetes Dashboard
- Setting up a Kubernetes Visualizer
- Resetting cluster created with kubeadm

24. Building blocks of Pods

- Introduction to pod
- Writing pod Specification
- Launching and Operating Pods (Login to the pod, browsing the web UI of the pod)
- Attaching a volume to Pod
- Launching Multi-Container Pods
- Connecting Individual Containers

- Launching Replica Set and Fault Tolerance
- Solution part - Deploying a worker app

25. Managing Application Configurations with ConfigMaps and Secrets

- Introduction ConfigMaps and Secrets
- Creating Config Map for Vote app
- Setting up Environment Specific Configs
- Adding Configs from Files
- Creating Secrets to Encrypt Database
- Setting Environment vars using Secrets



26. Setting up Firewall with Network Policies

- Creating default network policy for namespace
- Exposing public facing app and allowing inter namespace communication.



On Completing Module 2

After completing the second module, you will have a complete understanding of how a **operational team works**. The following concepts can be learned.

- Sourcecode Management
- Use tools to build your project seamlessly build
- Frontend development
- Automated serverside administration using Ansible
- How to use Docker/Kubernetes to containerize your project and development-infrastructure
- How to deploy a project on AWS/Local Server
- Best practices when deploying applications.



Module 3

DevOps Automation



1. Introduction to Continuous Integration and Jenkins-CI/CD

- What is Continuous Integration
- Jenkins Continuous Integration
- What is Continuous Deployment
- Jenkins Vs Jenkins Enterprise

2. Jenkins Installation

- Downloading and Installing Jenkins using TomCat
- Creating Jenkins as a Service.
- Starting and Stopping Jenkins

3. Configure Jenkins and User Management

- Secure Jenkins
- Create a new user
- Generate ssh key for Jenkins user
- Plug-in management

4. Jenkins jobs setup

- Setting up a Jenkins job (Freestyle, Pipeline, maven, MSBuild, Pybuild)
- Jenkins parametrized jobs setup (choice params, boolean params etc)

- Email notification jobs
- Parallel jobs configuration
- nodes (slaves) configuration

5. Jenkins Integration

- Git integration with Jenkins
- Maven Integration with Jenkins
- ansible , artifactory integration
- Docker and scanning tool integration
- AWS and code review tool

**Jenkins**



6. Jenkins User administration

- Role based administration
- Project based administration
- Metric based administration
- Slaves configuration
- Users and groups creation

7. Introduction to Continuous Monitoring using Nagios

- Introduction to Continuous Monitoring
- Introduction to Nagios
- Installing Nagios
- Nagios Plugins(NRPE) and Objects
- Nagios Commands and Notification

**Nagios[®]**

8. Promethues and Grafana

- Introduction to Prometheus and Grafana
- Installation and configuration
- Adding metric and pod level and node level
- Installing grafana plugin in Prometheus
- Monitoring using Prometheus
- Prometheus and Grafana Setup
- Dashboard Visualization using Grafana
- Creating a Dashboard to monitor the Pipeline

9. Introduction of Infrastructure as a Code (IaC)

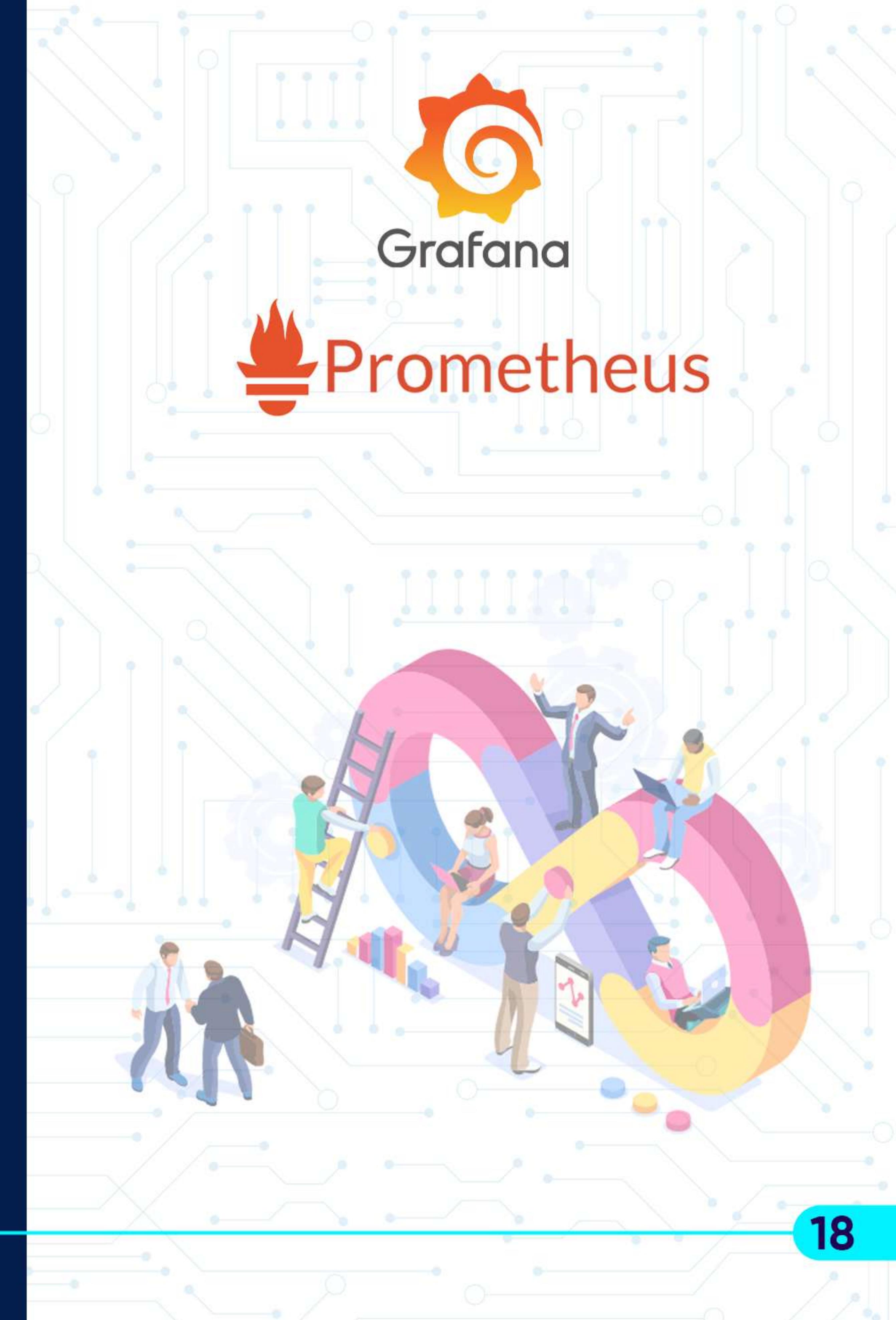
- Concept of IaC
- Key benefits of IaC

10. Terraform

- What is terraform and its workflow
- IaC tools - Ansible, Terraform, chef, Puppet

11. Testing Tools under CI/CD

- Types of testing tools
- Understanding uses of Selenium



On Completing Module 3

After completing the third module, you will have a complete understanding of how to **integrate development and deployment infrastructure** together. The following concepts can be learned.

- What is Continuous Integration and Continuous Deployment (CI/CD)
- How to use Jenkins to automate the CI/CD pipeline
- Complete obscurity about software performance and bug monitoring
- Familiarize yourself with tools like Prometheus, Grafana and Nagios
- What is Infrastructure as a Code (IaC)
- Using Terraform for Infrastructure as a Code (IaC)



At the End of training

- 01 You become a skilled DevOps Engineer with in-depth knowledge and hands on experience.
- 02 You will have a clear idea about Managing and Integrating Developmen team and Operational team.
- 03 Gets a complete knowledge on Advanced tools for automating Software Development.
- 04 You will have all the knowledge on technical concepts.
- 05 You will have a clear idea about software development life cycle.
- 06 Make you confident to apply against 'DevOps Engineer' jobs to secure a winning career

10 REASONS TO CHOOSE STEPS



Become a Skilled
DevOps Engineer



Learn From Industry
Experts



Learn A - Z of
DevOps



Online
Technical
Test



Group &
Individual
Assignments



Seminars to Ensure
a First Hand
Experience in
Technical portions



A Real
IT Park
Environment



Comprehensive
Practical



Great **Placement**
Assistance and
Interview Orientation



Get an
Experience
Equivalent
Certificate

IT'S TIME TO GET STARTED



STEPS | Spectrum IT Park
Mahakavi G. Road |
Kochi-11 | Kerala



www.stepskochi.com



info@stepskochi.com



0484-4082111 +91-9895682000

