Day 1

Introduction into DevOps & DevOps culture and practices

The Evolution of DevOps: A Historical Journey

DevOps is more than just a buzzword; it’s a cultural movement that has transformed how software is developed, delivered, and operated.

Origins and Dysfunction (2007-2008)

Why DevOps?

Key Principles and Practices

The Birth of DevOps

Necessity in a Complex World

Certainly! Let’s dive into the world of DevOps and explore its culture and practices.

DevOps is a collaborative approach that unites software development (Dev) and IT operations (Ops). Its primary goal is to streamline the software delivery process, enhance quality, and foster a culture of continuous improvement. Here are the key aspects of DevOps

- Collaboration and Communication
- Automation
- Continuous Integration and Continuous Deployment (CI/CD)
- Infrastructure as Code (IaC)
- Monitoring and Feedback
- Culture Shift

Why DevOps Matters

- Speed and Agility
- Quality Assurance
- Reliability and Stability
- Scalability
- Customer Satisfaction

In summary,
DevOps is a powerful force that transforms how software is developed, deployed, and maintained. By fostering collaboration, embracing automation, and nurturing a learning culture, organizations can thrive in the ever-evolving tech landscape.
🚀🔧👩‍💻

Day 2

Version control system

Version Control System (VCS)

Version control, also known as source control, is a practice used in software engineering to track and manage changes to code over time. It is a software tool that helps manage changes to source code over time. It allows multiple developers to collaborate on a project, keeping track of every modification to the codebase. Some popular version control systems include Git, Subversion (SVN), and Mercurial.

Purpose

Benefits

Types of VCS

Workflow and Flexibility

- Repository
- Commit
- Branching and merging
- Pull Request
- Conflicts Resolution

Automated Testing

Automated testing involves using tools and scripts to execute test cases automatically.It involves using software tools to run tests on the codebase automatically. These tests verify whether the software behaves as expected, helps catch bugs early in the development process, and ensures that changes don't introduce regressions.

Purpose

Benefits

Criteria for Automation

Automation tools

Types of Automated test

Other Benefit of Automated Test

Day 3

Hands on workshop on Git CI pipeline ad Github action