5.4. LIFE CYCLE OF DEVOPS – ADOPTION OF DEVOPS – DEVOPS TOOLS – BUILD, PROMOTION AND DEPLOYMENT IN DEVOPS

Life Cycle of DevOps

Introduction

The **DevOps Life Cycle** is a set of continuous processes used to develop, test, deploy, monitor, and maintain software applications. It enables organizations to deliver software quickly and reliably through automation and collaboration.

DevOps Life Cycle Diagram

PLAN
↓
CODE
↓
BUILD
↓
TEST
↓
RELEASE
↓
DEPLOY
↓
OPERATE
↓
MONITOR
↓
FEEDBACK
↓
PLAN

1. Planning Phase

Purpose

To identify project requirements and create a development roadmap.

Activities

- Requirement gathering
- Feasibility analysis
- Project scheduling
- Resource allocation

Tools

- Jira
- Trello
- Azure Boards

Benefits

- Clear project objectives
- Better resource management

2. Coding Phase

Purpose

Developers write source code according to project requirements.

Activities

- Coding
- Code review
- Bug fixing
- Version control

Tools

- Git
- GitHub
- GitLab
- Bitbucket

Benefits

- Efficient collaboration
- Better code quality

3. Build Phase

Purpose

Convert source code into executable software.

Activities

- Compilation
- Dependency management
- Packaging

Tools

- Maven
- Gradle
- Ant

Benefits

- Automated software creation
- Reduced errors

4. Testing Phase

Purpose

Verify software quality and functionality.

Types of Testing

- Unit Testing
- Integration Testing
- System Testing
- Performance Testing

Tools

- Selenium
- JUnit
- TestNG

Benefits

- Early bug detection
- Improved reliability

---

5. Release Phase

Purpose

Prepare software for deployment.

Activities

- Final testing
- Version management
- Release approval

Benefits

- Stable software release
- Reduced deployment failures

---

6. Deployment Phase

Purpose

Deploy software into production environments.

Activities

- Configuration management
- Container deployment
- Environment setup

Tools

- Docker
- Kubernetes
- Jenkins

Benefits

- Faster software delivery
- Reduced downtime

---

7. Operations Phase

Purpose

Maintain software after deployment.

Activities

- Server management
- Security management
- Backup and recovery

Benefits

- Continuous availability
- Better performance

---

8. Monitoring Phase

Purpose

Monitor application performance and infrastructure health.

Activities

- Log analysis
- Error detection
- Performance monitoring

Tools

- Nagios
- Prometheus
- Grafana

Benefits

- Early issue detection
- Improved reliability

---

2. Adoption of DevOps

Definition

DevOps Adoption is the process of implementing DevOps culture, practices, and tools within an organization.

Need for DevOps Adoption

- Faster software delivery
- Better collaboration
- Improved software quality
- Reduced operational costs
- Increased customer satisfaction

Steps in DevOps Adoption

Step 1: Cultural Transformation

Encourage collaboration between development and operations teams.

Step 2: Process Automation

Automate repetitive tasks.

Step 3: Continuous Integration

Implement CI practices.

Step 4: Continuous Delivery

Enable frequent software releases.

Step 5: Continuous Monitoring

Monitor software continuously.

Benefits of DevOps Adoption

Business Benefits

- Faster time-to-market
- Increased customer satisfaction
- Improved productivity

Technical Benefits

- Better software quality
- Reduced deployment failures
- Improved system stability

Challenges in Adoption

- Resistance to organizational change
- Lack of skilled professionals
- Tool integration complexity
- Security concerns

3. DevOps Tools

Definition

DevOps tools automate and simplify software development, testing, deployment, and monitoring processes.

Classification of DevOps Tools

DevOps Tools

|

| | | | | | |

Plan Code Build Test Deploy Monitor

---

1. Planning Tools

Jira

- Project management
- Issue tracking

Trello

- Task management
- Team collaboration

---

2. Version Control Tools

Git

- Source code management
- Branching and merging

GitHub

- Cloud repository hosting
- Collaboration platform

---

3. Build Tools

Maven

- Dependency management
- Build automation

Gradle

- Fast build system
- Flexible configurations

---

4. Continuous Integration Tools

Jenkins

- Automated build
- Automated testing
- Continuous integration

Bamboo

- Build and deployment automation

---

5. Configuration Management Tools

Ansible

- Infrastructure automation
- Configuration management

Puppet

- Server configuration

Chef

- Infrastructure management

---

6. Containerization Tools

Docker

- Creates lightweight containers
- Platform-independent deployment

---

7. Orchestration Tools

Kubernetes

- Container orchestration
- Auto-scaling
- Load balancing

---

8. Monitoring Tools

Nagios

- Infrastructure monitoring

Prometheus

- Metrics collection

Grafana

- Data visualization

---

4. Build, Promotion and Deployment in DevOps

Introduction

Build, Promotion, and Deployment are important stages in the DevOps pipeline that ensure software moves smoothly from development to production.

A. Build Process

Definition

Build is the process of converting source code into executable software.

Build Activities

- Source code compilation
- Dependency management
- Packaging
- Build verification

Build Workflow

Source Code

↓

Compilation

↓

Packaging

↓

Build Artifact

Benefits

- Consistent software creation
- Faster development
- Automated processes

B. Promotion Process

Definition

Promotion is the process of moving software through different environments before production.

Promotion Stages

Development
↓
Testing
↓
Staging
↓
Production

Purpose

- Verify software quality
- Ensure stability
- Reduce production risks

Benefits

- Better quality assurance

- Reduced failures
- Controlled software releases

C. Deployment Process

Definition

Deployment is the process of releasing software to end users.

Types of Deployment

Manual Deployment

Performed manually by administrators.

Automated Deployment

Performed automatically using DevOps tools.

Deployment Workflow

Build
↓
Test
↓
Release
↓
Deploy
↓
Monitor

Deployment Tools

Jenkins

Automates deployment pipelines.

Docker

Packages applications into containers.

Kubernetes

Manages and scales containers.

Advantages of Automated Deployment

- Faster releases
- Reduced human errors
- Improved consistency
- Better reliability

CI/CD Pipeline Diagram

Developer
↓
Git Repository
↓
Jenkins CI
↓
Build
↓
Test
↓

Docker Image

↓

Kubernetes Deployment

↓

Production

↓

Monitoring