DevOps

Software Development Lifecycle

Requirement
- gathering
- analysis

defining requirements

screens, flow architecture, database

programming

testing
- methods
- types
- levels

make the system accessible to end users

PLANNING

Talk to customer and understand the requirements

list of verified required

DEFINING

Define the requirements and stick to them

SRS

DESIGNING

Design the solution with right approach

DDS

BUILDING

Development following guidelines

complete system (Not tested)

TESTING

Make sure that your code is working

well tested system

DEPLOYMENT

Make your app available for rest of the world

publically accessible system / software

Waterfall Model

Entities involved

Developer

Testers

Operations Team

package

- deployable in an env
- consists of
  - code / binary / executable
  - dependencies
    - libraries
    - packages
    - frameworks
  - documentation

Building

- compile the application
- add the dependencies
- output: a deployable package

---

* android → Java/kotlin → .apk
* ios → swift/objC → .ipa
* web
  → React → JS/HTML/TS → web pack
  build directory
  → Angular → TS/HTML → — " —
  → other → any → web pack

Responsibilities

Dev Team

virtualization – technology to create virtual aspect of physical resources → VM

Developers and Testers

- Developers → add features
  - Develop the application
  - Package the application
  - Fix the bugs
  - Maintain the application
- Testers
  - Thoroughly test the application manually or using test automation tools
  - Report the bugs to the developer

Operations Team

- Make all the necessary resources ready → computers, network, virtualization, software
- Deploy the application → Dev / staging / production
- Maintain multiple environments
- Continuously monitor the application → making sure app is healthy
- Manage the resources

Challenges

Developers and Testers

- The process is slow
- The pressure to work on the newer features and fix the older code
- Not flexible

Operations Team

- Uptime
- Configure the huge infrastructure
- Diagnose and fix the issue

Agile Development → Scrum

scrum

- → roles – product owner, scrum master, dev team
- → artefacts – product backlog, sprint backlog, increment
- → events – sprint, sprint planning, daily scrum, sprint review, retrospective
- → pillars – transparency, inspection, adaptation

Scrum Process

Waterfall Vs Agile

The Waterfall Process

This project has got so big.
I am not sure I will be able to deliver it!

The Agile Process

It is so much better delivering
this project in bite-sized sections

Problems

- Managing and tracking changes in the code is difficult
- Incremental builds are difficult to manage, test and deploy
- Manual testing and deployment of various components/modules takes a lot of time
- Ensuring consistency, adaptability and scalability across environments is very difficult task
- Environment dependencies makes the project behave differently in different environments

dev.

→ latest version

→ features

→ node - 22.04

operations

→ uptime

→ stable version

→ node - 20.01 (LTS)

Solutions to the problem

- Managing and tracking changes in the code is difficult: **SCM tools** → Git / SVN / CVS / bazaar
- Incremental builds are difficult to manage, test and deploy: **Jenkins** → CI / CD pipeline
- Manual testing and deployment of various components/modules takes a lot of time: **Selenium** : test automation tools
- Ensuring consistency, adaptability and scalability across environments is very difficult task: **Puppet** : env creation / configuration
- Environment dependencies makes the project behave differently in different environments: **Docker** : containerization

What is DevOps ?

→ dev
→ tester

- DevOps is a combination of two words development and operations
- Promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way → sprint by sprint

  → automated
- DevOps helps to increases an organization's speed to deliver applications and services
- It allows organizations to serve their customers better and compete more strongly in the market
- Can be defined as an alignment of development and IT operations with better communication and collaboration
- DevOps is not a goal but a never-ending process of continuous improvement ∞
- It integrates Development and Operations teams
- It improves collaboration and productivity by
  - Automating infrastructure
  - Automating workflow
  - Continuously measuring application performance

Why DevOps is Needed?

- Before DevOps, the development and operation team worked in complete isolation
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays

Common misunderstanding

- DevOps is not a role, person or organization
- DevOps is not a separate team * → devops team
- DevOps is not a product or a tool
- DevOps is not just writing scripts or implementing tools

→ mindset of continuous improvement ∞

Reasons to use DevOps

- **Predictability**
  - DevOps offers significantly lower failure rate of new releases
- **Reproducibility**
  - Version everything so that earlier version can be restored anytime
- **Maintainability**
  - Effortless process of recovery in the event of a new release crashing or disabling the current system
- **Time to market**
  - DevOps reduces the time to market up to 50% through streamlined software delivery
  - This is particularly the case for digital and mobile applications
- **Greater Quality**
  - DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues
- **Reduced Risk**
  - DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle
- **Resiliency**
  - The Operational state of the software system is more stable, secure, and changes are auditable

DevOps Lifecycle

DevOps Lifecycle - Plan

→ no of dev, # hours, # stories, story assigment

- First stage of DevOps lifecycle where you plan, track, visualize and summarize your ~~project~~ sprint before you start working on it

simple tools

→ text Editors → notepad / atom / vs code
→ spread sheets → Excel / open office
→ cloud tools → one drive / google drive / box

scrum tools

- Jira
- Orange Scrum
- Trello

Tools

DevOps Lifecycle - Code → development / programming

- Second stage where developer writes the code using favorite programming language

languages
- C/C++
- python
- JS/TS
- Java/C#

Code Editors
- notepad
- VS code
- atom
- sublime
- vim

IDEs
- → Node → Webstorm
- → python → Pycharm
- → android - android studio
- → ios - Xcode
- → C# → VS
- → C,C++ → Clion

SDKs
- JS → NodeJS
- C# → .Net SDK
- android → androidSDK
- ios → ios SDK

SCM tool
- git
- CVS
- SVN
- bazaar
- bitkeeper

Tools

DevOps Lifecycle -Build

- Integrating the required libraries
- Compiling the source code
- Create deployable packages

package → code/binary + dependencies + documentation

Build tools → Ant → deprecated
→ Maven
→ gradle → standard tool

Tools

maven

DevOps Lifecycle - Test

- Process of executing automated tests
- The goal here is to get the feedback about the changes as quickly as possible

Functional testing

→ unit testing → python (PyUnit)
function
→ JS/TS (Jasmine/Jest)
→ C# (NUnit)
→ Java (JUnit)

UI testing

- selenium - web UI testing
- cypress
- TestNG

Tools

JUnit

nunit

Se

APACHE JMeter™

TestNG

Non-functional

- load testing
→ load Runner
→ JMeter
- Win Runner

Stress testing
→ Stress tester

Security
penetration testing - VAPT

DevOps Lifecycle - Release → CI/CD pipeline →

continuous integration
continuous delivery
continuous deployment

- This phase helps to integrate code into a shared repository using which you can detect and locate errors quickly and easily

CI/CD pipeline → sequence of stages

Tools

Travis

CI Tools

- → Travis CI
- → Circle CI
- → GitLab CI
- → GitHub Actions

CD Tools

- ArgoCD

CI/CD

- Jenkins
- Bamboo

DevOps Lifecycle - Deploy : moving app from one env to another [production deployment]

- Manage and maintain development and deployment of software systems and server in any computational environment

* traditional deployment → using physical machines → deprecated
* virtualized deployment → using virtual machines → vmware / virtualBox / BlueStack
* containerized deployment → using containers → docker / podman / LMCTFY / LXC / LXD / containerd.io / rkt
  container orchestration tools → docker swarm / kubernetes / marathon

Tools

DevOps Lifecycle - Operate : env creation and configuration

- This stage where the updated system gets operated

- continuous env creation tools → terraform (HashCorp), cloud formation (AWS), vagrant
  cloud resources — local

→ continuous env config tools → ansible, puppet, chef

Tools

DevOps Lifecycle - Monitor :

- It ensures that the application is performing as expected and the environment is stable
- It quickly determines when a service is unavailable and understand the underlying causes

monitoring tools → splunk, DataDog, Nagios, New Relic

Tools

splunk>

DATADOG

Nagios®

DevOps Terminologies

** Continuous Learning **

Responsibilities of DevOps Engineer

Linux

Be an excellent sysadmin

containerization

Deploy Virtualization

networking

Hands-on experience in network and storage

Introduction to coding

Soft skills

CI / CD pipeline tools

Automation tools

Selenium

Software Testing knowledge

security professional

IT security ***

Skills of a DevOps Engineer

Skills	Description
Tools	• Version Control – Git/SVN • Continuous Integration – Jenkins • Virtualization / Containerization – Docker/Kubernetes • Configuration Management – Puppet/Chef/Ansible • Monitoring – Nagios/Splunk
Network Skills	• General Networking Skills • Maintaining connections/Port Forwarding
Other Skills	• Cloud: AWS/Azure/GCP • Soft Skills • People management skill