Introduction of Course

Learn Using MindMaps

Why DevOps?

Before DevOps

Blame game Begun

Main Issues Before DevOps

What is a WaterFall Model?

Phases of WaterFall Model (MindMap).

Drawbacks of WaterFall Model

What is Agile Model?

Drawbacks of Agile Model

DevOps methodology

Stages or SDLC of DevOps

DevOps LifeCycle (How DevOps Symbol Came?)

Continuous Development & CD Tools?

Continuous Integration & CI tools?

Continuous testing & tools?

Continuous Deployment & tools?

Continuous Monitoring & Tools

Introduction of Course

Learn Using MindMaps

Why DevOps?

Traditional Methods

Before DevOps?

Agile Model

Waterfall Model

DevOps Method

Strategies

What?

Drawbacks

What?

Strategies

Drawbacks

DevOps Lifecycle

DevOps Approach/ Model

What?

Strategies

How it Helps?

Deployment

Development

Continuous

DevOps Tools

Testing

Monitoring

1. Why DevOps?
2. What is DevOps?
3. DevOps Strategies/Methodologies

Why DevOps?

Before DevOps

Before DevOps

Vs

Operations Team

Dev Team

**Blame game Begun**

Deploying

Developing

Client Side

Operations Team

Dev Team

Building

Maintaining

Deploy App

Testing

Monitoring

Working for me

QA

Not Installed Properly

Issues might be

1. Supported Libraries / Missing libraries
2. OS versions (dev – version 8.1, ops - version 10)
3. Software Version issue

Also, the issue for delay might be because developer handed over code to testing and somehow managed to pass testing and when went for QA it fails.

That is how we will see how further **docker** came into picture to build the application.

Docker Image

Docker Image

Docker File

?

Testing

QA

Because of using Docker file/image the dependencies issues get resolved

**Main Issues Before DevOps**

1. Dev team Vs Ops team (Blame game )
2. All is manual process
3. Time consuming process

**What is WaterFall Model? Phases of WaterFall Model (MindMap).**

Software Development Life Cycle

Linear Model

Before DevOps

Waterfall Model

Follows Top-Down Approach

**Stages of Waterfall Model**

How Application looks?

**Design**

Analyze

**Requirement**

Design Blueprint

Begin Coding of the application

**Implementation**

Production

Testing of the application

**Verification**

Make It Live

**Monitoring Phase**

Accessible to End Users

**Blue Print** 🡪 Design model which shows how the application interface looks like i.e; Login page, Registration page, Connections, databases etc, each and every single detail here.

**Monitoring:** monitor application health and server.

**Drawbacks of WaterFall Model**

1. Unless and until one stage or one phase completes, you cannot proceed with next phase.
2. Time Consuming.
3. Not suitable for Dynamic approach once application is Live.
4. No Feedback – until you finish the application, you cannot show it to the client.

**What is Agile Model?**

We have broken down the software into various iterations. i.e; separate phase for Planning, Design, Development, Testing, QA, Deploy

Broken Software

Development

Testing

Designing

Planning

QA

Deploy

Monitor

Agile Model

Planning and Designing: period is 2 to 8 weeks.

Agile Model

Developer  
 +  
Operartons

Gathered Feedback

Live with End Users

Release V1

Incorporated fixes

Release V2

Gathered Feedback

Live with End Users

Release V3

Incorporated fixes

Gathered Feedback

Live with End Users

Release V4

Incorporated fixes

Gathered Feedback

Live with End Users

**Drawbacks of Agile Model**

1. Operations teams were unable to catch up the speed with Development team, which is constant conflict.
2. Again the blame game exists.
3. Hampering the release goal.

**DevOps Methodology / Approach**

Combination of Development & Operation made as one team called DevOps.

We have mitigated the dev and ops team here, single team who takes care of development and operations team.

This doesn’t mean Ops team need to know coding, but should be able to read the necessary files for app deployment

1. DevOps engineer deals with

Building

Testing

Quality Assurance (QA)

Deploying

Maintain

Monitoring

1. Practice / Culture that manages the entire application, following SDLC
2. Software development approach

End Goal is to deliver High quality software which will be delivered quickly without any many work / automation with increased reliability

**Stages or SDLC of DevOps**

DevOps approach consists of various stages

DevOps

C Monitoring

Continuous Integration

Continuous Testing

Continuous Development

Continuous Deployment

**Aim of DevOps?**

1. Shorten the SDLC

GitHub

Developer

Webhook

Production

Jenkins

DevOps Engineer

Bug fix

Webhook

Jenkins

GitHub

Developer

Production

DevOps LifeCycle (How DevOps Symbol Came?)

C Deployment

Deploy

C Development

Plan

Login  
Registration  
Bug Fixes

Operation

Code

Integration

Build

Monitor

C Monitoring

C Testing

Test

C Integration

**Build: Artifact / compilation**

**Continuous Development & CD Tools?**

1. Planning & Coding of functionality
2. Git, SVN, JIRA (Login, Register)
3. Ant, Maven, Gradle etc; (Build & Package the code into executable)

**Continuous Integration & CI tools?**

Jenkins (WebHooks)

**Continuous testing & tools?**

Selenium

JUnit

**Continuous Deployment & tools?**

Configuration tools 🡪 Releases deployments to servers

Ansible

Puppet

Chef

Containerization/Orchestration

Docker

Kubernetes

**Continuous Monitoring & Tools**

Aim: Quality of software

Datadog

Dynatrace

CloudWatch

Prometheus

Grafana