Introduction to DevOps

CLARUSWAY
WAY TO REINVENT YOURSELF

Table of Contents

- Phases of SDLC (Software Development Life Cycle)
- Waterfall Model
- What is DevOps?
- Continuous Integration/Continuous Delivery (CI/CD)
- DevOps Tools

Phases of SDLC

Waterfall Model

- Each phase completely wrapping up before the next phase begins
- If parameters change along the way, it's harder to change course

What is DevOps

What is DevOps

What DevOps is Not...

- a tool
- a role
- a team
- something that can be purchased or simply switched on

What is DevOps

What is DevOps

- **DevOps** is a set of practices that combines software **development** (Dev) and IT **operations** (Ops).
- It aims to **shorten** the software development life cycle and provide **continuous delivery** with **high software quality**.
- DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.
- DevOps addressed the **gap** between Developers and Operations.

Before DevOps

What is DevOps

Roles;

DEVELOPMENT TEAM

- Project Manager
- Software Architects
- Developers
- Testers/QA, etc.

OPERATION TEAM

- System Administrators
- Database Administrators
- Release Engineers
- Network Engineers
- Security Professionals, etc

What is DevOps

Responsibilities;

What is DevOps

What is the problem?

What is DevOps

Breaking the Silos: Communication, Collaboration, Integration

- How can dev help system stability?
- How can ops help accelerate feature delivery?

“We can build cross-functional teams around “knowledge overlaps” – people with experience on both sides and “Ops Devs””

Introduction to DevOps

What is DevOps ?

DevOps is a software development approach which involves continuous development, continuous testing, continuous integration, continuous deployment, and continuous monitoring of the software throughout its development lifecycle

2

Continuous Integration/ Continuous Delivery (CI/CD)

CI/CD

01
Version Control
Maintains different versions of the code
Source Code Management
02
Continuous Integration
Compile, validate, Code Review, Unit Testing, Integration Testing
Continuous Build
03
Continuous Delivery
Deploying the build application to test servers, Performing UAT
Continuous Testing
04
Continuous Deployment
Deploying the tested application on the prod server for release.
Configuration Management and Containerization
Continuous Monitoring

Integration Hell

CI

code
code

Test
Report
Build
Continuous integration
Developer
Source control
Release

Introduction to DevOps

Introduction to DevOps

What is Continuous Integration ?

Continuous integration is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.

Introduction to DevOps

Traditional way

With CI &CD

Introduction to DevOps

With CI vs Without CI

Development without CI	Development with CI
Lots of Bugs	Fewer bugs
Infrequent and slow releases	Regular working releases
Difficult integration	Easy and Effective Integration
Late bug finding(days,weeks)	Early bug finding(minutes,hours)
Issue raised are harder to fix	Find and fix problems faster and more efficiently.
Poor project visibility	Better project visibility

Introduction to DevOps

What is Continuous Delivery?

Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, following a pipeline through a "production-like environment", without doing so manually.

Introduction to DevOps

What is CI/CD?

In software engineering, **CI/CD** is the combined practices of continuous integration **(CI)** and (more often) continuous delivery or (less often) continuous deployment (**CD**).

Introduction to DevOps

Continuous Delivery vs Continuous Deployment

3 DevOps Tools

Introduction to DevOps

DevOps Tools

Introduction to DevOps

Continuous Integration Server

Introduction to DevOps

Continuous Integration Server

Gitlab

Teamcity

Introduction to DevOps

Version Control Stage

EC2
CodeCommit

GitHub

Bitbucket
Atlassian

Codecommit
AWS

Gitlab
Gitlab

Introduction to DevOps

Build Stage

EC2
CodeCommit,CodeBuild
ECS(Elastic Container Service)

Maven
npm
nuget
BASH
THE BOURNE-AGAIN SHELL
Xcode
Gradle
<APACHE ANT>
CMake

Introduction to DevOps

Unit Test Stage

EC2
CodeCommit,CodeBuild
ECS(Elastic Container Service)

JUnit

JACOCO
Java Code Coverage

sonarqube

Introduction to DevOps

Deployment Stage

EC2
CodeCommit,CodeBuild, CodeDeploy
ECS(Elastic Container Service)
ECR,S3,RDS,DynamoDB
Cloudformation,Beanstalk
ELB, AutoScaling

Introduction to DevOps

Auto Test Stage

EC2
CodeCommit,CodeBuild, CodeDeploy
ECS(Elastic Container Service)
ECR,S3,RDS,DynamoDB
Cloudformation,Beanstalk
ELB, AutoScaling

Protractor
end to end testing for AngularJS

cypress.io

Visual Studio
Coded UI

Introduction to DevOps

Deploy to Production Stage

EC2
CodeCommit,CodeBuild, CodeDeploy
ECS(Elastic Container Service)
ECR,S3,RDS,DynamoDB
Cloudformation,Beanstalk
ELB, AutoScaling
Cloudfront,Route53

Introduction to DevOps

Monitoring Stage

Cloudwatch
SNS
SQS
SES

Prometheus

Grafana

Nagios®

ATLASSIAN Opsgenie

THANKS!

Any questions?