DAY 6: CI/CD

Introduction

DevOps will turn to **automation** to scale their development and deployment practices.

Automation is the process of using tools and scripts to perform tasks for us. You can automate version control to testing and finally to deployment. Automation of the deployment process creates the **Continuous Integration / Continuous Delivery (CI/CD) Pipeline**. This also encompasses **Continuous Testing** and **Continuous Deployment**.

We will cover:

- Bottlenecks of Manual Deployment
- Automation in DevOps
- Continuous Testing
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- The Complete CI/CD Pipeline

Each component of the CI/CD pipeline addresses a bottleneck in a manual deployment pipeline.

———————————————————————————

Bottlenecks of Manual Deployment

Manual deployment isn't always a bad thing, it can cause a lot of problems when trying to work on a large project with many developers.

Version Control Management
This can cause some problems on a large scale if developers are working on long-lived branches containing new features. Once the developers try to merge their code, it can cause a lot of merge conflicts and introduce new bugs.

Testing
It is usually not enough to only run a test before merging a branch. Testing infrequently can lead to bugs being overlooked. Additionally, testing issues can compound once merge conflicts begin to occur between developers.

Infrastructure and Environments
A deployment would sometimes fail due to human error, but it wasn't such a problem when she was working by herself.

Differences between various deployment environments can be tricky to manage. Setting up environments manually makes room for human errors that can cause the server to go down and delay production.

———————————————————————————

Automation in DevOps

Automated systems are:

- Faster — automated processes can perform operations much faster than people.
- Less error-prone — automation is able to perform a task more consistently than a person.
- Cheaper — workers don't have to be paid to do these repetitive workflows.

Automation can be incorporated into every part of the deployment pipeline. automated tools can be set up to monitor repositories, detect new merges, and trigger the new version of the application to be built and tested.

Other uses for Automation:

- Configuring servers
- Moving the application through testing and staging servers
- Executing tests
- Deploying to production
- Monitoring the application

———————————————————————————

Continuous Testing

This automation of testing is called **Continuous Testing**. It involves automatically triggering tests to be executed once an application is built in a new environment, this should result in catching bugs early and ensuring that the requirements of the project are met.

Some forms of the test are:

- Tests during development
    - Unit Tests
    - Integration Tests

- Tests before deployment to production
    - Acceptance Tests
    - End-to-end Tests

Since tests are triggered automatically each time the application is built in a new environment, developers don’t need to constantly monitor the state of the tests. Devs just receive alerts when a failed test occurs.

When tests pass in a given environment, the approved code can automatically be moved into the next environment where even more tests may be executed.

———————————

Continuous Integration

Continuous integration is a practice that consists of two main components:

- Merging source code changes on a frequent basis
- Building and testing the changes in an automated process