PPSD Report

Session: 2021

Submitted by:

Wali Muhammad	**2021-SE-39**
Numan Ahmad	**2021-SE-24**
Labeeb Tariq	**2021-SE-15**
Muhammad Hashim	**2021-SE-06**
Farjad Waseem	**2021-SE-56**

Submitted to:

Dr. Umer Qasim

Department of Computer Science,
University of Engineering and Technology, New Campus
Lahore

TABLE OF CONTENTS

CI/CD PIPELINE REPORT

CI/CD Pipeline Introduction

The CI/CD (Continuous Integration / Continuous Delivery) pipeline is a set of automated processes that allow developers to efficiently and continuously deliver code changes from development to production.

Usage

1. Building

   Compiling source code and dependencies into executable binaries or artifacts efficiently and accurately, ensuring that the resulting artifacts are consistent and reproducible across different environments.

2. Testing

   Verifying the functionality, performance, and reliability of the software through a comprehensive suite of automated tests, including unit tests, integration tests, and end-to-end tests. This thorough testing process helps detect and prevent regressions, ensuring that the software meets quality standards and user expectations.

3. Improving

   Addressing any issues or bugs identified during testing promptly and effectively, utilizing feedback from automated testing tools and manual inspections. This iterative improvement process focuses on enhancing code quality, resolving defects, and optimizing performance to deliver a robust and reliable software product.

4. Deployment

   Releasing the tested and improved software to production or staging environments seamlessly and efficiently, leveraging automated deployment pipelines and deployment strategies such as blue-green deployments or canary releases. This streamlined deployment process minimizes downtime, reduces deployment risks, and enables rapid and reliable software delivery to end users.

Advantages

Advantages include faster release cycles, improved code quality, reduced risk of errors, and enhanced collaboration between development and operations teams.

1. Faster Deliver new features and updates to users more frequently, staying competitive in rapidly evolving markets.

2. Collaboration between developers and operation team, fosters teamwork and communication between developers and operations teams.
3. Improve code Quality and software works as expected, with fewer bugs and errors.

Other Software Production Models

1. Waterfall
   - Waterfall is traditional approach of software development where everything happens step by step.
   - Series includes Requirements Gathering, Design, Development, Testing, Maintenance
   - It's less flexible and costly.
2. Agile
   - Programmers creates prototypes to understand client's requirements
   - The entire process is break down into small actions called sprints which is submitted every 2 weeks.
   - Fails when developers and operation team work in silos.

CI/CD Pipeline is a new version of agile. Agile focuses on client and developers team for changing requirements, where CI/CD Pipeline improves the interaction between Developers and Operational team, hence improving the overall production procedure.

DevOps in CI/CD Pipeline

DevOps consists of 8 phases which are the backbone of Ci/CD Pipeline.

1. **Planning**: Planning involves defining goals, setting priorities, and establishing a roadmap for software development and deployment.
2. **Development**: Development encompasses writing, testing, and integrating code changes into the software product.
3. **Integration**: Integration focuses on merging code changes from multiple developers into a shared repository and ensuring compatibility.
4. **Testing**: Testing involves verifying the functionality, performance, and reliability of the software through automated tests and manual checks.
5. **Deployment**: Deployment entails releasing the tested software to production or staging environments for end-user access.
6. **Monitoring**: Monitoring involves continuously monitoring the performance, usage, and health of deployed applications and infrastructure.

7. **Feedback**: Feedback involves gathering insights from users, stakeholders, and monitoring systems to identify areas for improvement and inform future iterations.

8. **Continuous Improvement**: Continuous improvement focuses on iteratively enhancing processes, tools, and workflows to optimize efficiency, quality, and customer satisfaction.
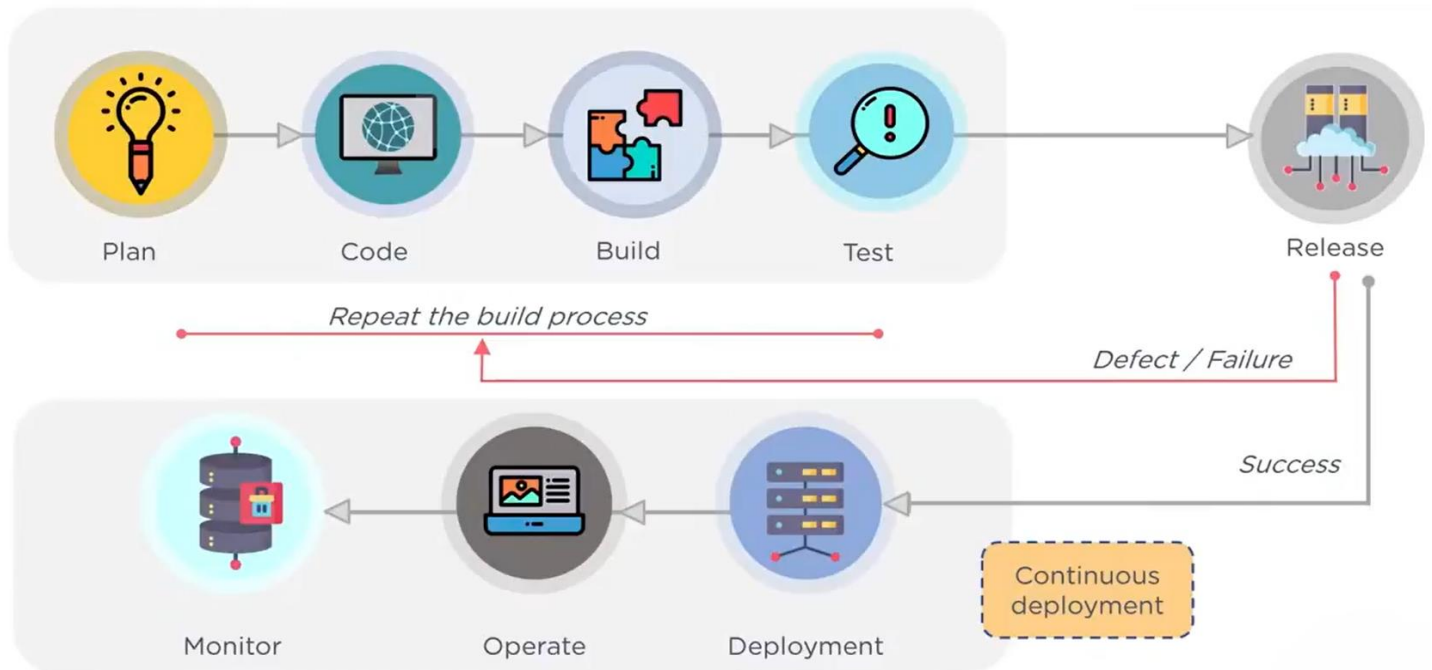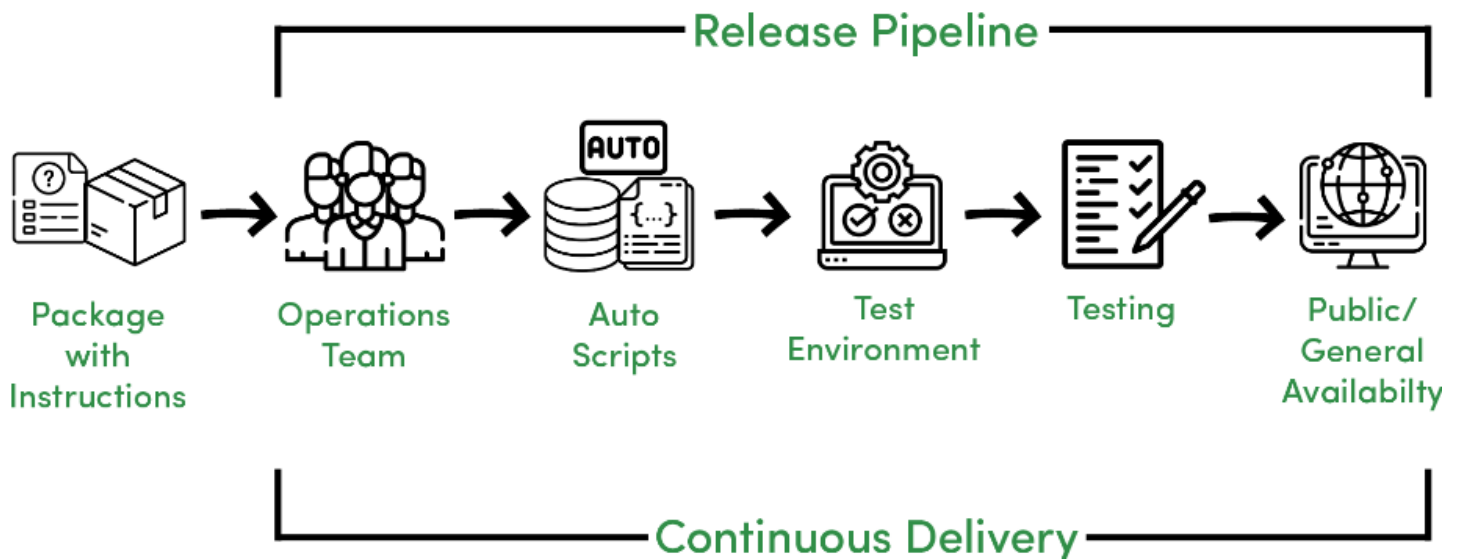

Continuous Integration and Delivery

Continuous Integration is a development practice of code integration into a shared repository. Each integration is verified by an automated build and automated test.
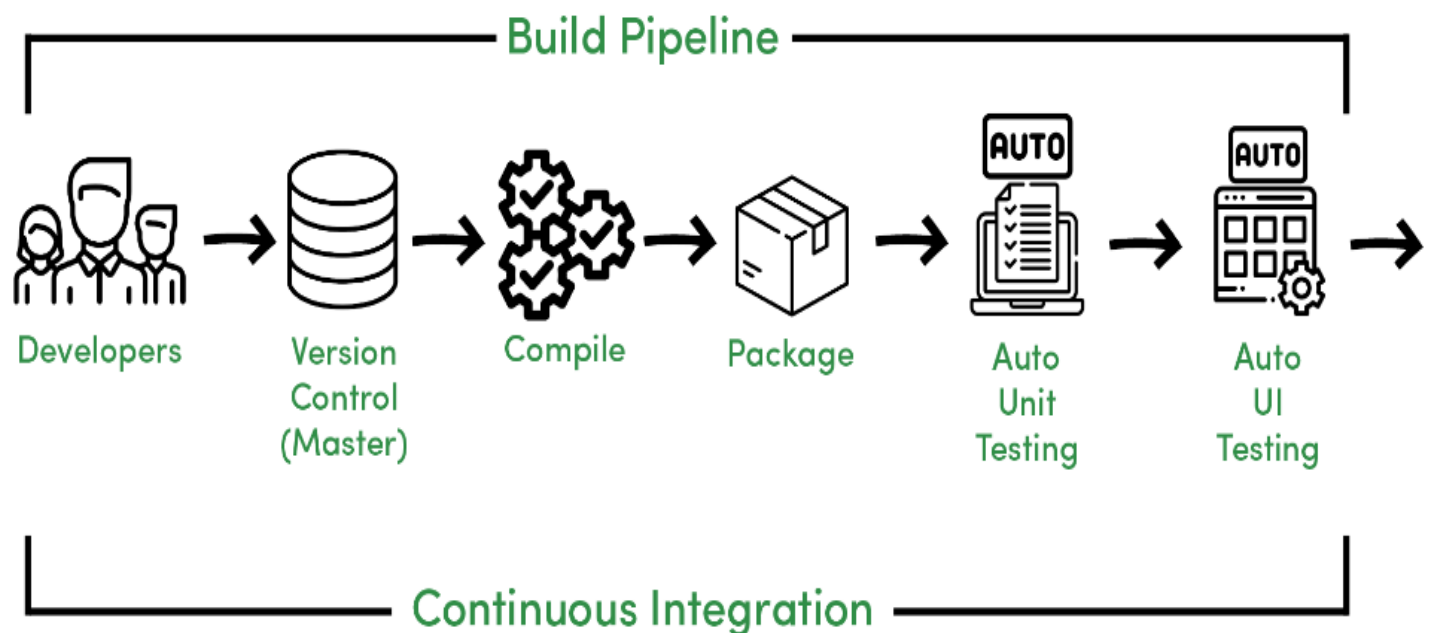
Continuous Delivery is a development practice of code which occurs after the integration phase.


Sample Pipelines

Jenkins

Jenkins is a widely-used open-source automation server that facilitates the implementation of CI/CD pipelines. With its extensive plugin ecosystem and robust automation capabilities, Jenkins enables developers to automate the entire software delivery process, from code compilation and testing to deployment and monitoring. Its user-friendly interface and flexibility make it a popular choice for organizations of all sizes seeking to streamline their CI/CD workflows.
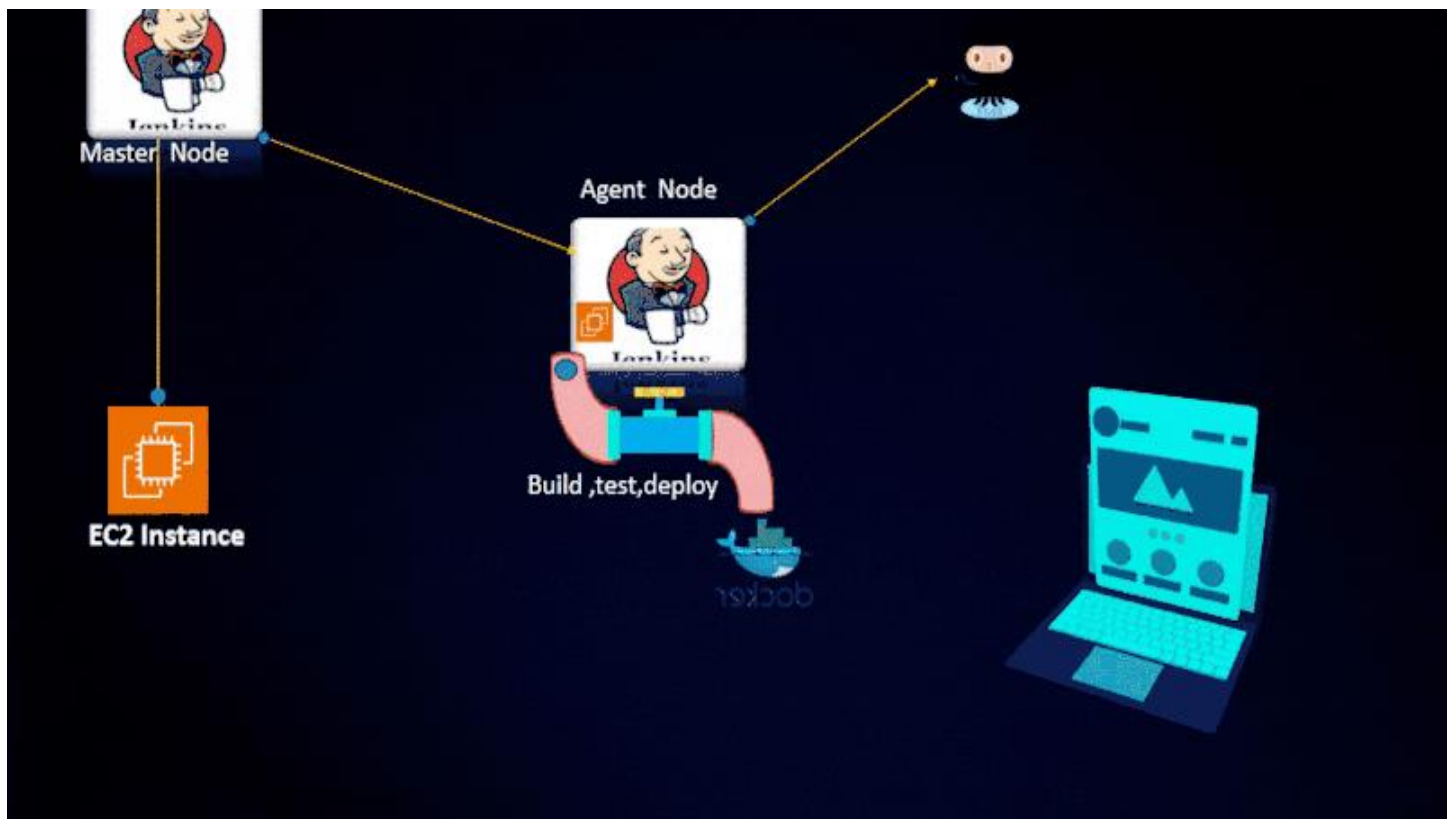
GitLab

GitLab is a comprehensive DevOps platform that provides built-in CI/CD capabilities alongside version control, issue tracking, and collaboration tools. With GitLab CI/CD, developers can define and manage their CI/CD pipelines directly within the GitLab repository, eliminating the need for separate CI/CD tools. GitLab's integrated approach simplifies the development lifecycle, promotes collaboration, and accelerates software delivery, making it a preferred solution for teams looking for an all-in-one DevOps platform.
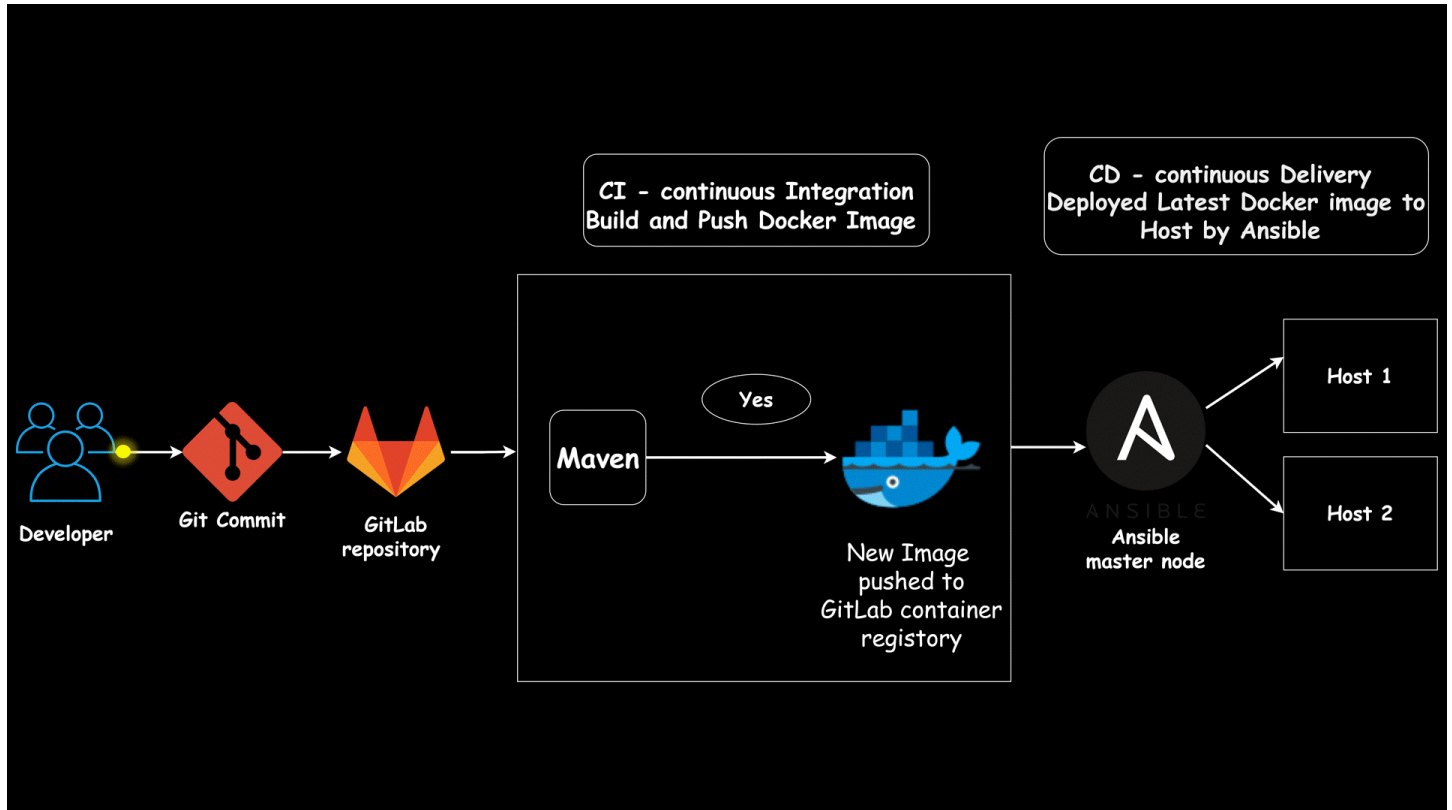
Numerical Stats

By leveraging CI/CD pipelines, organizations can achieve continuous improvement in code quality, delivery speed, and responsiveness to customer needs, ultimately driving business success and innovation

Delivery Speed	Code Quality
55% Improvement when using the CI/CD Pipeline. - CI/CD pipelines can lead to significant improvements in delivery speed, with some companies reporting up to 90%	55% Improvement when using the CI/CD Pipeline. - CI/CD practices promote better code quality by automating code reviews, testing, and validation processes. While it's difficult to quantify exact percentage

reductions in deployment times. This acceleration allows organizations to release new features and updates more frequently, keeping pace with market demands and staying ahead of competitors.	improvements in code quality, implementing CI/CD pipelines often results in fewer bugs, reduced technical debt, and higher overall code stability.

Top Companies which use CI/CD Methodology

- Google
    - **4 million builds per day**
    - Utilizes CI/CD extensively for its wide range of products, including Google Search, Gmail, and Google Cloud Platform.
- Amazon
    - **50 million builds per year**
    - Emphasizes CI/CD practices for its cloud computing services and various online retail operations.
- Netflix
    - **1 thousand builds per day**
    - Relies heavily on CI/CD for its streaming platform, enabling rapid feature releases and updates.

THE END