Ingeniería En Desarrollo Y Gestión De Software

Nombre:

García Arreola Howard Isai

Subject:

Software Development Process Management

Actividad:

Tools for CI/CD practices

Grupo: 10-B

Profesor:

Ray Brunett Parra

Galaviz

Fecha de Realización:

January 15th, 2025

Continuous Integration:

automation process for developers that facilitates more frequent merging of code changes back to a shared branch, or “trunk.” As these updates are made, automated testing steps are triggered to ensure the reliability of merged code changes.

Successful CI means that once a developer’s changes to an application are merged, those changes are validated by automatically building the application and running different levels of automated testing, typically unit and integration tests, to ensure the changes haven’t broken the app. This means testing everything from classes and function to the different modules that comprise the entire app. One of the benefits of CI is that if automated testing discovers a conflict between new and existing code, it is easier to fix those bugs quickly and often.

Continuous Delivery / Deploy:

Continuous delivery automates the release of validated code to a repository following the automation of builds and unit and integration testing in CI. So, to have an effective continuous delivery process, it’s important that CI is already built into your development pipeline.

Continuous delivery usually means a developer’s changes to an application are automatically bug tested and uploaded to a repository (like GitHub or a container registry), where they can then be deployed to a live production environment by the operations team. It’s an answer to the problem of poor visibility and communication between dev and business teams.

Continuous deployment means that a developer’s change to a cloud application could go live within minutes of writing it (assuming it passes automated testing). This makes it much easier to continuously receive and incorporate user feedback. Taken together, all these connected CI/CD practices make the deployment process less risky, whereby it’s easier to release changes to apps in small pieces, rather than all at once.

CI/CD Tools:

- **Jenkins**

  Open-source automation server that allows developers to automate the building, testing, and deployment of applications. It has a vast plugin ecosystem, enabling teams to customize CI/CD pipelines to suit their needs. Jenkins supports distributed builds across multiple machines.

- **CircleCI**

  CircleCI is a cloud-based CI/CD tool designed to integrate seamlessly with repositories like GitHub and Bitbucket. It supports various environments, including Docker, Linux, and macOS, allowing developers to build, test, and deploy code efficiently. Its easy-to-configure YAML files make setting up workflows straightforward.

- **GitLab-CI/CD**

  Integrated into the GitLab platform, GitLab CI/CD provides a robust solution for continuous integration and deployment. It allows teams to manage repositories, track issues, and run CI/CD pipelines in one cohesive environment. Its strong security features and seamless scalability make it a preferred tool for enterprises.

- **Travis-CI**

  Travis CI is a popular CI service that works directly with GitHub repositories. It automatically runs tests and deployments whenever code is pushed to the repository. With its language-specific builds and pre-installed libraries, it simplifies CI/CD setup, especially for small to mid-sized projects.

- **Bitbucket-Pipelines**

  Embedded within Bitbucket, Pipelines is a CI/CD tool that simplifies the development workflow. Developers can define build steps as code using YAML files, and execute them directly within their Bitbucket repositories. Its seamless integration with Atlassian tools like JIRA makes it a great choice for teams already using the Atlassian ecosystem.

- **AWS-CodePipeline**

  AWS CodePipeline is a fully managed CI/CD service provided by Amazon Web

Services. It automates the building, testing, and deployment processes, ensuring that code changes are released rapidly and reliably. It integrates seamlessly with other AWS services, offering developers a comprehensive solution for cloud-based deployments.

- **Azure-Pipelines**

  Part of the Azure DevOps suite, Azure Pipelines provides end-to-end support for building, testing, and deploying applications across multiple platforms and cloud providers. It supports containers, Kubernetes, and serverless applications, making it a versatile choice for modern software development.

- **TeamCity**

  TeamCity, developed by JetBrains, is a powerful CI/CD server with extensive customization capabilities. It supports parallel builds, integrates with a variety of development tools, and provides insightful build statistics. Its user-friendly interface and robust plugin ecosystem make it suitable for complex development workflows.

- **Bamboo**

  Bamboo is Atlassian's CI/CD tool designed to integrate seamlessly with JIRA and Bitbucket. It provides a unified interface for automating builds, tests, and deployments, and supports parallel execution. Bamboo's intuitive interface and pre-built functionalities make it an excellent choice for teams leveraging Atlassian products.

- **Octopus-Deploy**

  Octopus Deploy specializes in deployment automation, complementing other CI tools by managing releases and deployments. It supports various environments, including on-premises, cloud, and hybrid infrastructures. Octopus Deploy is particularly strong in handling multi-step deployment processes and complex release requirements.

References:

What is CI/CD? (n.d.). https://www.redhat.com/en/topics/devops/what-is-ci-cd

GeeksforGeeks. (2024, September 18). *15 best CI/CD tools that you should know*. GeeksforGeeks. https://www.geeksforgeeks.org/best-ci-cd-tools/