



# CONTINUOUS INTEGRATION





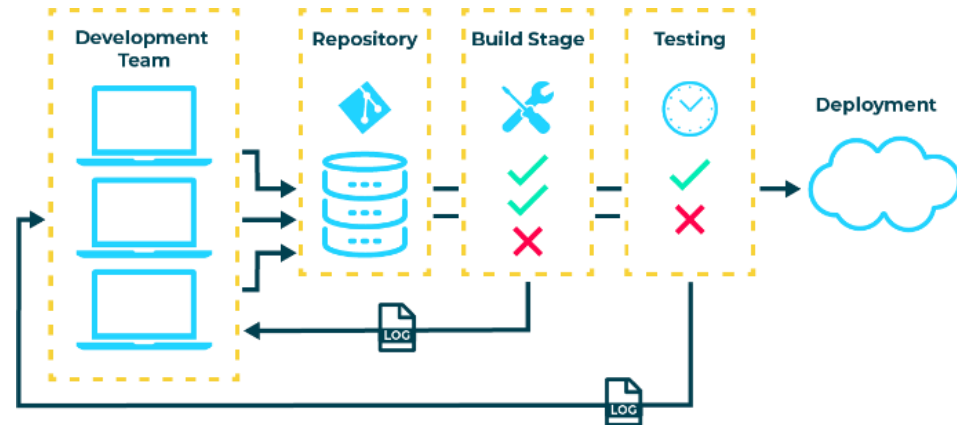
## CONTINUOUS INTEGRATION

- Automated integration of code from many contributors into a single software project
- Allow developers to integrate newly-generated code easily and frequently
- Achieved through the use of automated testing tools to check the accuracy of code before full integration

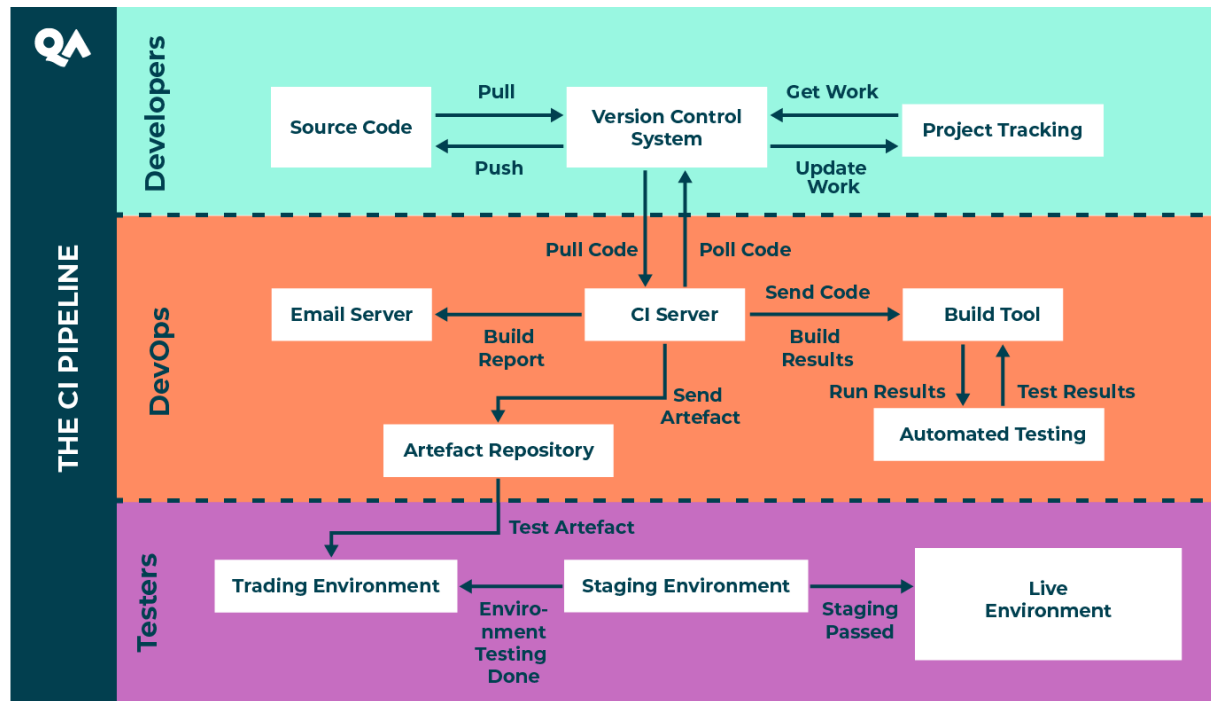




# CONTINUOUS INTEGRATION



# CI pipeline





## CI PIPELINE



A CI pipeline should:

- Maintain a single source code repository for a project
- Have a "master" branch that should always be ready to deploy
- Keep all team members informed of every update to the source code
- Automate build processes
- Automate testing of new builds
- Inform developers of test failures with detailed logs
- Encourage smaller, frequent deployments of code



## CI PIPELINE

CI leverages many software tools to handle the automated building, testing, and deployment processes.

The main steps in the CI pipeline include:

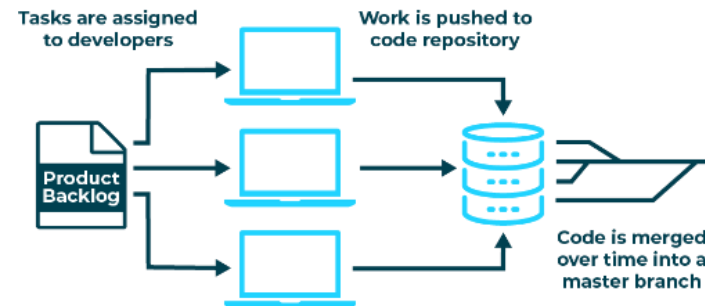
1. Code generation
2. Code repository
3. Building and testing





# VERSION CONTROL SYSTEM

- Designed to track changes to code over time as contributors add new features to the application
- Allows for cohesive collaboration
- Ability to easily revert an application to a previous, stable state if new code breaks something



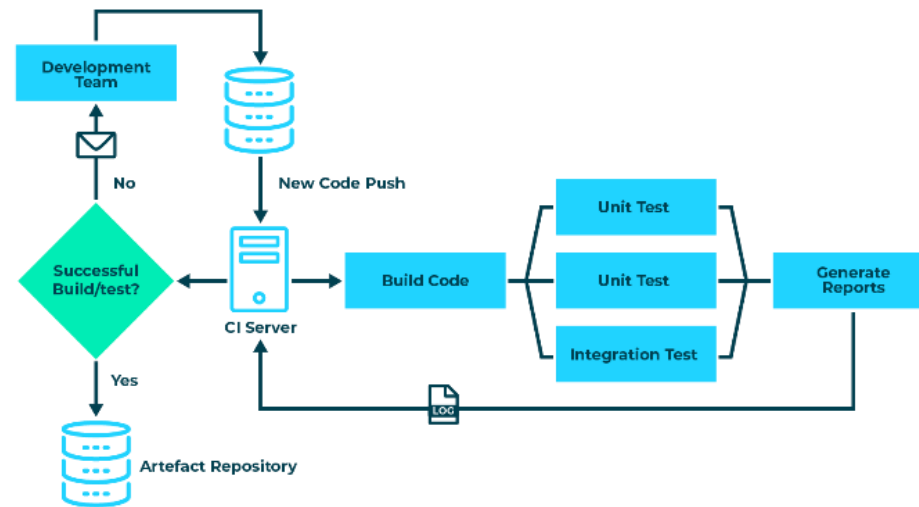


- Handles all the automated building, testing, and deployment of code as it is pushed to the VCS
- If a test or build fails, integration is halted and the developer is informed of where the code failed along with detailed reports





# CI SERVER





## BENEFITS



- **Scaling**
  - Automating frees up time for developers to do other things
  - Allows for scaling up the development team, code generation, code integration
- **Feedback Loop**
  - Gradual feature updates allow for more opportunities for business feedback
  - More client/customer feedback
  - Rapid bug fixes



## BENEFITS



- **Communication**

- Trackable changes enhances communication between teams
- Team members avoid stepping on each other's toes
- Aids transparency of work across the organisation



## CHALLENGES



- **Installation and adoption**
  - Likely to be a long and involved process
  - Potential to waste time, effort, and money should it be approached without enough planning
- **Learning curve**
  - Pipelines make use of many different and relatively new technologies that teams may not have any prior experience with
  - New workflow
  - Some teams may find their old responsibilities obsolete and have to readjust their set of responsibilities