CI/CD

Continuous Integration/Continuous Deployment

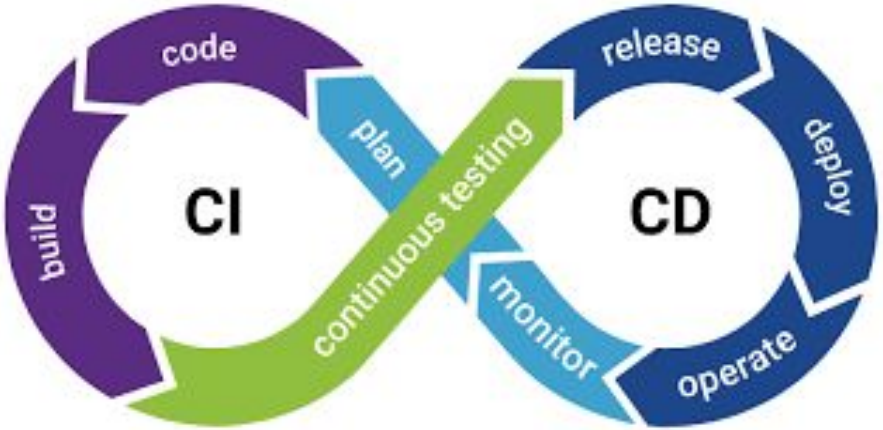
What is CI/CD?

Continuous Integration: The Practice of merging all developers' working copies to a shared mainline several times a day

Continuous Deployment: A software engineering approach in which the value is delivered frequently through automated deployments

CI + CD = **Continuous Delivery**

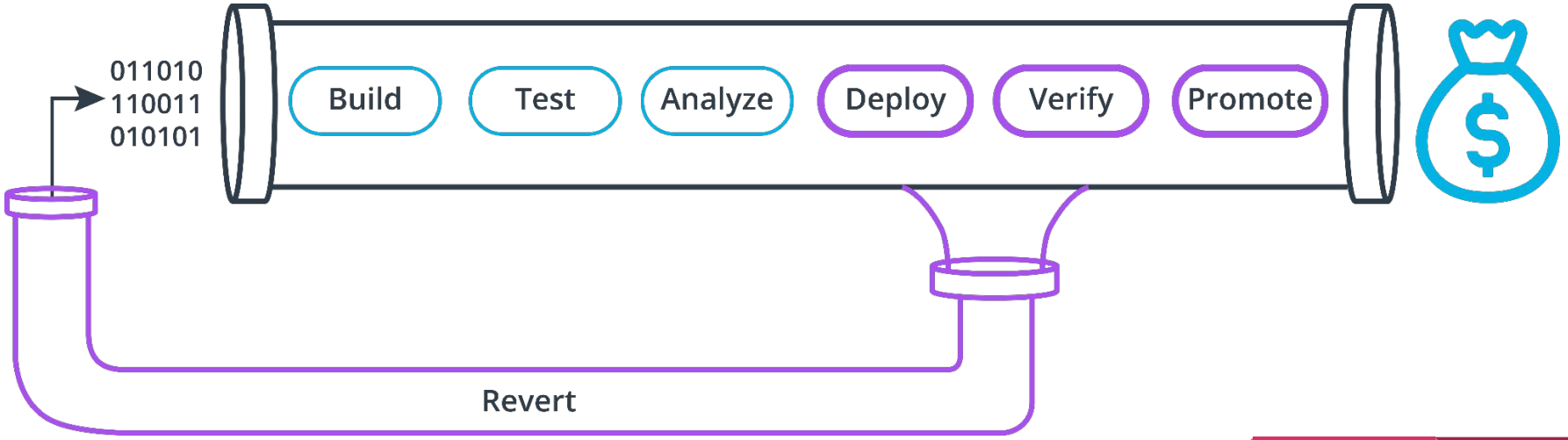
Continuous Delivery: An engineering practice in which teams produce and release value in short cycles



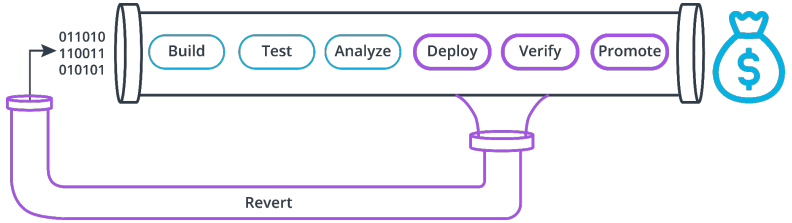
Benefits

- **Reduced Risk:** Less human error (no human involved). Every step is written in code.
  - ⇒ **Reduce costs**. Less time spent fixing avoidable human error
- **Builds confidence in the code**: new changes are verified before they are merged so the code quality is kept constantly high.
  - ⇒ **Reduce Cost**. Less time spent on issues from new code
- **Automated Infrastructure and Code Deployments**: no time wasted on manual steps then manually validating and/or debugging these steps
  - ⇒ **Increase Revenue**. Faster time to market
- **Automatic Cleanup:** Automated cleanup of leftover artifacts and previous deployment infrastructure
  - ⇒ **Reduce Cost**. Less cost from unused artifacts in storage or IDLE infrastructure
- **Faster and more frequent Production deployments**: Deployment is done by scripts which require minimum time and have automated rollback plans in case of failure
  - ⇒ **Protect Revenue**. The software is always working because of multiple deployments and ease of rolling back to a working state

Workflow


011010
110011
010101
Build
Test
Analyze
Deploy
Verify
Promote
$
Revert

Workflow



Build	Compile the project
Test	Run the automated test and require them to pass before moving forward
Analyze	Check the code for code quality and security vulnerabilities
Deploy	Build infrastructure and configure it to deploy and run the code
Verify	Make sure the deployed version of the code is working as intended
Promote	Switch the currently publicly available version with the newly deployed version

If any of the previous three steps fail:

Revert: Destroy the deployment and rollback any changes made to the current deployment

Remember all of these steps are **fully automated.**