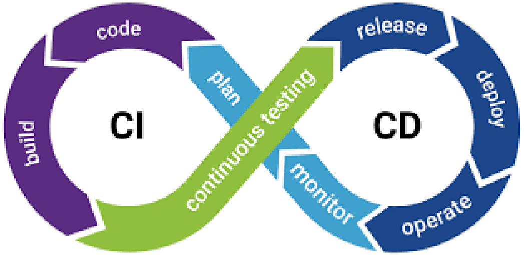
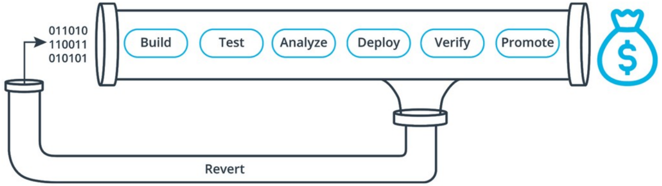
Fundamentals & Benefits of CI/CD

- Juan Lopez



Fundamentals of CI/CD

Continuous Integration (CI)



Continuous Deployment (CD)

Practice of merging all developers code several times a day to one main repository. The process of MAKING. The ultimate goal of CI is a high quality, deployable artifact:

- Compile
- Unit test
- Static analysis
- Dependency vulnerability test
- Store artifact

Approach where the value is delivered frequently through automated deployments. The process of MOVING the artifact to the spotlight:

- Creating infrastructure
- Provision servers
- Copying files
- Promote to production
- Smoke test
- Rollbacks

Benefits of CI/CD

Reduce & Avoid Cost

1. Catch Compile Errors After Merge:
   Less developer time on issues from new developer code

2. Catch Unit Test Failures:
   Less bugs in production and less time in testing

3. Detect Security Vulnerabilities:
   Prevent embarrassing or costly security holes

4. Automate Infrastructure Creation:
   Less human error, Faster deployments

5. Automate Infrastructure Cleanup:
   Less infrastructure costs from unused resources

Increase & Protect Revenue

1. Faster and More Frequent Production Deployments:
   New value-generating features released more quickly

2. Deploy to Production Without Manual Checks:
   Less time to market

3. Automated Smoke Tests:
   Reduced downtime from a deploy-related crash or major bug

4. Automated Rollback Triggered by Job Failure:
   Quick undo to return production to working state

Best Practices for CI/CD

01 Fail Fast

Set up your CI/CD pipeline to find and reveal failures as fast as possible. The faster you can bring your code failures to light, the faster you can fix them.

02 Measure Quality

Measure your code quality so that you can see the positive effects of your improvement work (or the negative effects of technical debt).

03 Only Road to Production

Once CI/CD is deploying to production on your behalf, it must be the only way to deploy. Any other person or process that meddles with production after CI/CD is running will inevitably cause CI/CD to become inconsistent and fail.

04 Maximum Automation & Config in code

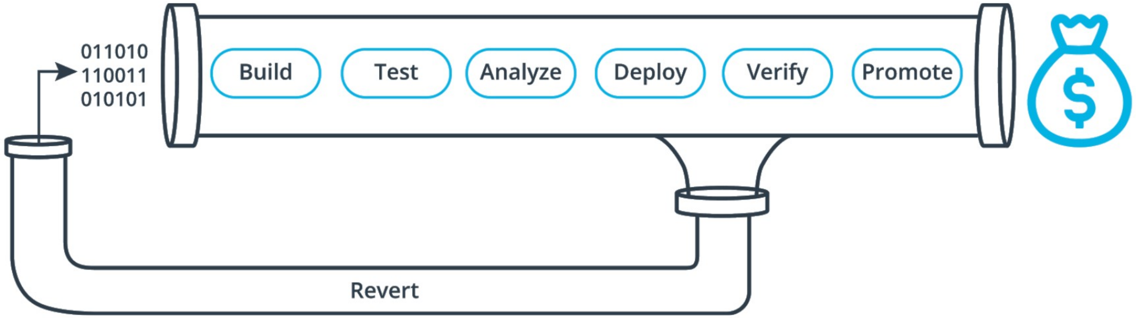
If it can be automated, automate it. This will only improve your process!
All configuration code must be in code and versioned alongside your production code. This includes the CI/CD configuration files!

The End!

We’ve gone over the fundamentals and benefits of CI/CD. Thank you for your consideration and time!

- Juan Lopez

CI/CD Pipeline


011010
110011
010101
Build
Test
Analyze
Deploy
Verify
Promote
$
Revert