

CI/CD Pipeline proposal

Created By-

Garima Modi

Agenda

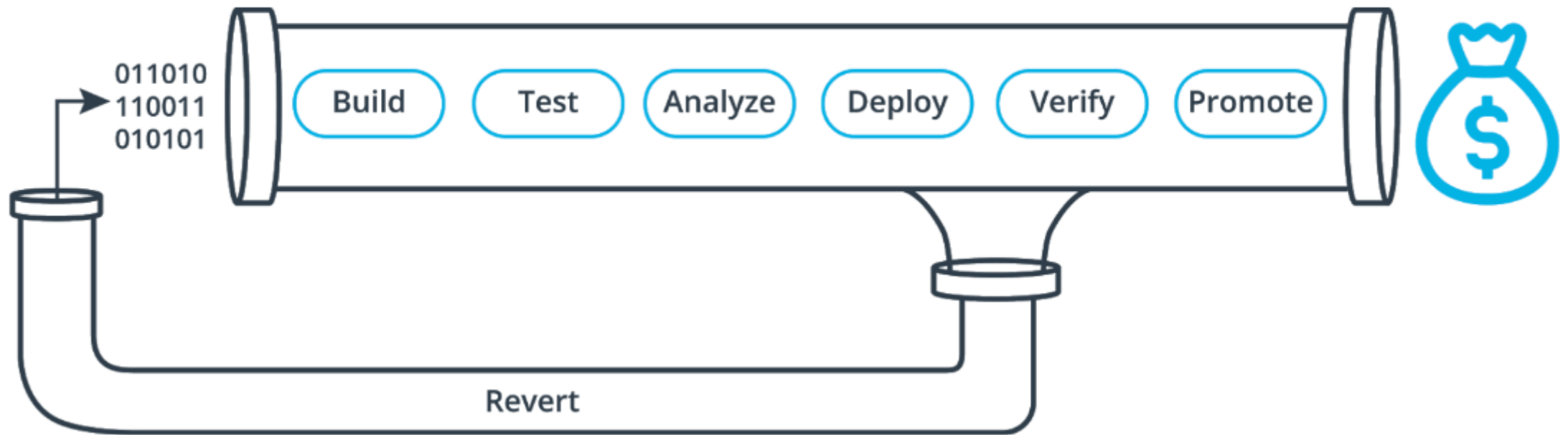
- Overview of CI/CD pipeline
- Phases of CI/CD pipeline
- Benefits of CI/CD pipeline
- How do you know you need CI/CD or not?

Overview : CI/CD Pipeline

- Continuous integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests.
- Continuous delivery usually means a developer's changes to an application are automatically bug tested and uploaded to a repository, where they can then be deployed to different environments including production environment by the operations team
- Continuous deployment can refer to automatically releasing a developer's changes from the repository to production, where it is usable by customers.

Phases of CI/CD pipeline

The CI/CD Pipeline



Benefits of CICD

- Reduction in cost because of less Developer time on issue then focus on code development.
- Faster feedback - CICD pipeline catch testing error early which means less errors in production which save money
- Faster deployments and automated infrastructure setup reduce avoid cost
- More revenue generation because of faster deployment to production.

How do you know you need CI/CD or not?

- Think of the "warning signs" in your team that suggest that your project would be good candidates for CI/CD or Continuous Delivery. If you identify with any of these items, you should consider CI/CD an essential piece of your development workflow.
- Investing more time in a release cycle than delivering value
- Going through integration hell every time we finish a feature
- Code gets lost because of botched merges
- Unit test suite hasn't been green in ages
- Deployments contribute to schedule slip
- Friction between ops and development departments
- Only one engineer can deploy a system
- Deployments are not cause for celebration