• CI/CD – the best way to deliver you application

Fundamentals and Benefits of CI/CD

CI/CD brief

- **What is CI/CD?**

- Continuous integration and continuous delivery pipeline processes that are used to build, package, and deploy any app

- **Principles of Continuous Delivery**

1. **Repeatable Reliable** Process : can repeat the process many times

2. **Automate Everything** : automate every process which speed up the delivery

3. **Version Control Everything**: can version any tasks of the whole pipeline

4. **Bring the Pain Forward**

5. **Build-in Quality** : it enhance the quality through reliable pipeline which includes all sort of testing and validation

6. **"Done" Means Released** : it means that all the tasks done starting from building to prod release

7. **Everyone is Responsible** : all are contributed in the release

8. **Continuous Improvement :**

Continuous integration vs Continuous deployment

- ## Continuous deployment
  The part responsible to build and running automated tests against the build and scan and remove vulnerabilities

  Phases examples
  - ❑ Compile
  - ❑ Unit Test
  - ❑ Analysis
  - ❑ Scan and close vulnerabilies
  - ❑ save the oartifact

- ## Continuous deployment
  this part is responsible for everything related to deployment its like moving the artefact to the spotlight

  Phases examples
  - ❑ Deploying infrastructure
  - ❑ Migration of DB
  - ❑ Copying files
  - ❑ Smoke Testing (aka Verify)
  - ❑ Promoting to production
  - ❑ Rollback

Continuous Delivery: it's the practice in which teams produce and release value in short cycles.

Business Values of CI/CD

❖ Speed: through automating every release part we gain speedy deployments which helps to respond to market

❖ Productivity : through repeated automated pipeline we can easily detect every issue so now teams focus on visioning but not the pain manual repeated work

❖ Sustainability : automated pipelines reduce manual work and lead to savings since personnel is way too expensive than tools.
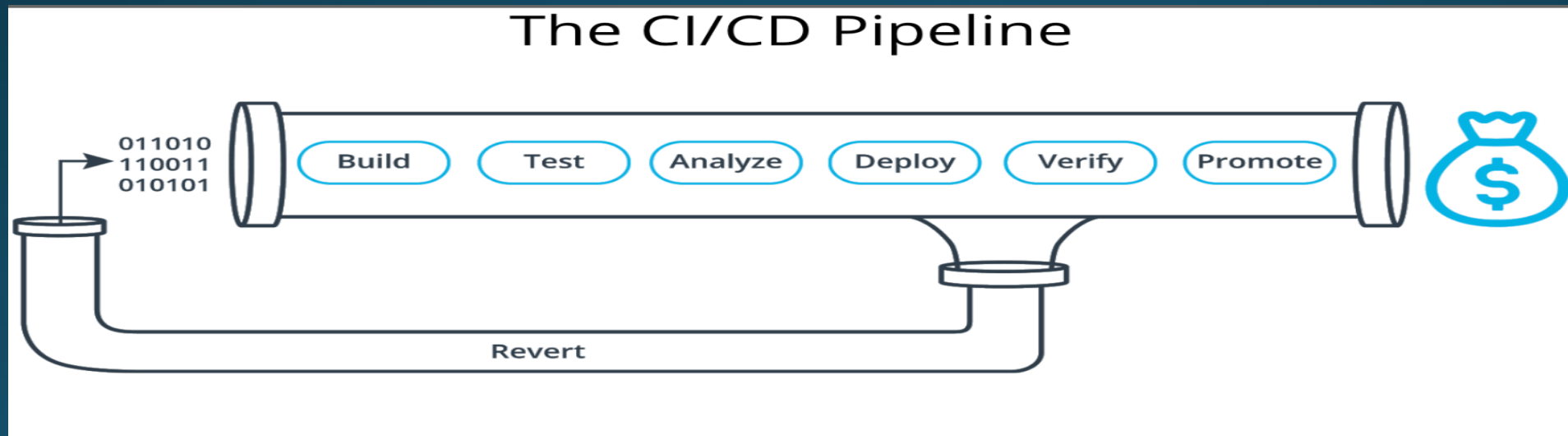
Benefits of the CI/CD

- **increase revenue:**
  - ❑ New features released more quickly
  - ❑ Less time to market
  - ❑ Faster adaption to market

- **protect revenue**
  - ❑ Reduced outage due to a deployment bug or issue
  - ❑ Fast rollback to return production to recent working state

- **Control and reduce costs**
  - ❑ Prevent costly security vulnerabilities
  - ❑ less issues and less time in testing and releasing
  - ❑ Less human mistakes and faster deployments
  - ❑ Less developer time on issues from new developer code