Proposal for the benefits of ci/cd

By: Kariman Mahmoud

overview

- What are the phases of CI/CD

- Why do we need CI/CD rather than traditional deploying phase,
  and What are the challenges we will be confronted with?

- How could we benefit from CI/CD as a DevOps principle

The three Major phases for CI/CD

▶ **Continuous Integration**
its the process of merging developer branches to the main branch several times a day. CI puts an emphasis on test automation and generates a high quality, deployable artifact.

▶ **Continuous Delivery**
It makes sure that changes of a software product can be released quickly to customers in an automated way and at any point in time.

▶ **Continuous Deployment**
extends Continuous Delivery in a way that it allows automation of deployments without any manual action to be done. Typical phases in Continuous Deployment are Infrastructure Provisioning, Smoke Testing, Production Deployments and automated Rollbacks.

Why do we need CI/CD rather than traditional deploying phase

- Manual and traditional deploying ways points that are faced:
- Our manual release process is error-prone and always leads to delays of production deployments.
- Poor software quality since we don't have time for quality analysis anymore.
- Deployments are pretty complex.
- No smoke tests and rollback mechanisms.
- We get late feedback from the business department which prevents us from creating flexible solutions.

HOW WE COULD BENEFIT FROM CI/CD AS A DEVOPS PRINCIPLES

▶ Cost reduction due to less human errors and faster deployments.

▶ Reduce complexity.

▶ Safe manual troubleshooting time.

▶ Automated Smoke Tests and Rollbacks will protect project revenue due to reduced downtimes from deploy-related crashes and fast and automated rebuilding of production ready state.

▶ Breaking changes in code as well as in infrastructure deployments can be found much quicker and can be resolved for the whole automation process.

▶ Faster feedback cycles of customers lead to higher customer satisfaction rates since they are involved right from the beginning of feature development/deployment and not just at a fixed release date.