Why to use CI/CD?

Holm Hinrichs / Udacity Colud Dev Ops Nanodegree

The current Situation

How we work:

- Manual merge new features into Main branch

- Configuring test suite manually per tool menu

- Building new version on demand

- Deployment is done manually

- Target environment is managed manually

- All knowledge is bind to one person!

The current Situation

Leads to:

- We loose code on failed merges

- Merging new features costs a lot of time

- Compiling errors are not always detected in time!

- Our tests are not reliable!

- We need to invest a lot of effort to make our unit tests "green"!

- Releasing new features costs us a lot of time and effort

- Roll-backs can lead into critical situations

- Security issues can not be identified quickly, hot fixes are taking very long

- High dependency on one single person

- The frustration level on the team is high

The current Situation

Is having the following impacts:

- Deployments are more cost intensive
  as we spend too much timer on doing them

- As of bugs and security issues, we risk our customer satisfaction

- We loos revenue as we can not deploy new features in time

- We loose skilled developers as of high frustration

- Using a cost intensive infrastructure

- We risk our revenue with critical roll-backs

CI/CD can help!

By providing the following features:

- Catch compile errors before merging

- Automatically run Unit Test, and catch errors

- Almost all configuration is checked into source code

- Detect security issues and CVE's on used libraries

- Automated infrastructure creation and cleanup

- Automatically deploy new versions without manually checks

- Automatic smoke tests

- Automated rollback on failed job

CI/CD can help!

By giving the following benefits:

Reuces Costs	Avoid Costs	Protect Revenue	Increase Revenue
Less issue tracking on new code	Reducing bugs in production and less time for testing	Less downtime from crashes related from deployment	Reduce time to market!
Less unused infrastructure	Less bugs and security issues in production	Quick fall-backs to a working state	New value generating features are released faster