

# Introduction to CI / CD

Divyansh Rajesh Jain

# What is CI / CD?

- CI → Continuous Integration
- CD → Continuous Development
- CI / CD → Set of (mostly) automated process which allows for a fast cycle from development to deployment

# Traditional Approaches to CI / CD

- Manual Testing → To ensure that everything is working
- Manual Upstreaming → Upstream software from development to production manually (usually a full team for this)

# State Of Art Approach to CI / CD

- Tests: Unit Tests, Integration Tests, and Scaling Tests  
are run
- Manual Review: Requires a manual reviewer to  
approve changes
- Build and Deploy: Software is built and then  
deployed (Kubernetes)

# Common Tools for CI / CD

- Jenkins → Open Source CI / CD tool
  - Pros: Has many ready-made plugins
  - Cons: It is complicated to use, requires dedicated team
- Github Actions → CI / CD tools build into Github
  - Pros: Easy to use, integrated into Github
  - Cons: Not as powerful and open as Jenkins

# Class Site Example

# Overheads to CI / CD

- High Costs associated with maintaining DevOps Team
- High Costs associated with mistake(s) in CI / CD
-

# Thank You!

Next Time: More In-Depth about Jenkins