A beginner’s guide to CI/CD

Mohammad Ahtasham ul Hassan

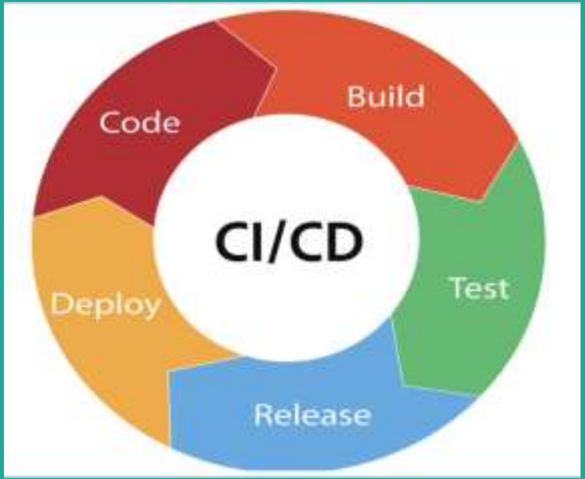
Introduction

- Full stack Engineer on project edX
- Expertise in Web development with Django and React.
- A wanna be AI expert :P
- Participant in PyCON PK 24
- Occasional Gamer

Agenda

- Introduction to CI/CD
- Traditional Software development
- CI
- CD
- When to use CI/CD?
- CI/CD Tooling & real world examples
- Conclusion

SO….?
What is CI/CD???



Traditional Software Development

Old School Approach

- Emphasize Extensive planning and documentation at beginning.
- Testing is typically performed at the very end.
- Relies heavily on manual tasks.
- Sequential in Nature, release cycles tend to be longer.
- Limited customer/user involvement

CI

Validate the state of codebase through Automated testing.

- Best achieved through shared version control system
- Automated Build and test process
- Identify errors in early stages
- Increases collaboration among developers.

CD

Extends CI(Pre req) by automating the release process.

- Make releases reliable, repeatable and low risk
- No or little manual intervention
- Build frequently, Ship frequently.
- Rollbacks are super easy and just a click away.
- Consistent and reproducible environments

When to use CI/CD?

CI/CD Tooling & Real world examples

- Github Actions(Free tier available)
- Jenkins(Open source)
- Travis
- CircleCI
- GoCD and ArgoCD

Conclusion

Lorem ipsum dolor sit amet(Gotcha)