Why CI/CD

We can decrease human errors

- Use Infrastructure as Code to make sure that testing and deployment environments are consistent across iterations.
- We include testing to code quality and security so that if it fails it doesn’t get to be produced.
- The tests are written once, hence consistent for every build.

Better visibility

- When there is an error or failure, stakeholders are notified promptly.
- This can ensure timely fixing and product quality.
- We can proactively monitor services to prevent failures by sending alerts for would-be problems.

It costs less

- Adding all the previous points we can save money through:
    - Decreasing costs of fixing as they would be caught earlier.
    - Decreasing costs of roll-backs in case of failures as the software and environment are based on Software.
    - Decreasing Down-time as errors are caught right away.
    - Unused infrastructure is done with promptly therefore there is no undue OpEx

It increases revenue

- Automating builds allows developers to push features faster and with better quality.
- Decreasing time-to-market.
- Decreasing operating costs due to economies of scale.