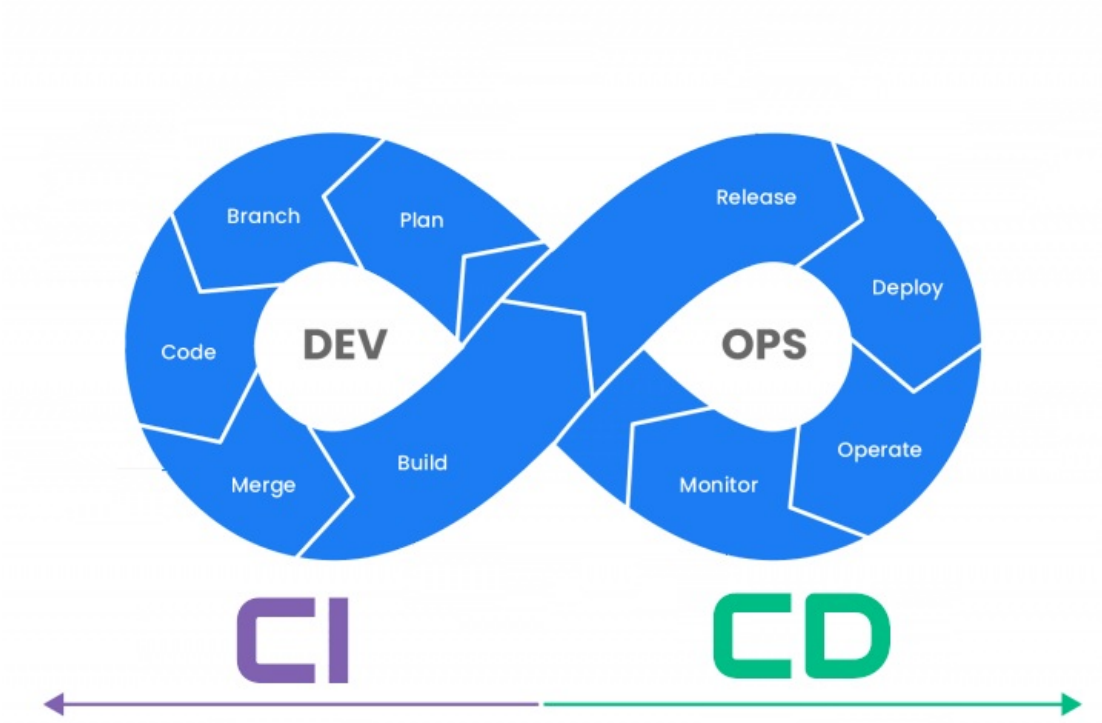
Adopting CI / CD



Fundamental Concepts:

- **Continus Delivery**: a Software engineered practice / approach for teams to produce and release software (code versions) in short cycles.

  > **Benefit: deliver "near-production-quality" to clients as quickly as possible**

- **Continuous Integration**: the practice of merging all the developers work, in an Automated and Systematic Fashion to prepare the code for Deployment.

  > **Benefit: Big short-cut to a lot of time consuming & Recuring processes so More Production is Achievable from Development-Team || More control & reduce costs**

- **Continuous Deployment**: the practice of automating everything related to deploying the Software the Result of CI (ARTIFACT)

  > **Benefit: another major decrease in time consuming & Recuring processes so More Production is Achievable from Operations-Team || More control & reduce costs**

Benefits of the DevOps CI/CD Practice

- **Help teams ship code to production faster** by reducing manual tasks and making each step in the software development lifecycle (SDLC) consistent and repeatable.

- **Shortens the feedback loop between developers / engineers and customers**, by Delivers Required changes to "near-production-quality" in a much shorter time. Leading to more clients Satisfaction and Good Impressions.

- **No more risk of human errors**, by systemizing the software development and bring consistency to each stage and its corresponding output. So developers will focus on more important work.
- **CI/CD can help produce bug-free code**, by applying a consistent suite of automated tests and providing a feedback loop between team members.
- **No more shipping large code changes that can take time to integrate and test with the code base**, Because CI/CD Helps in delivering better software with less issues more consistently.
- **Greatly enhances Teams Productivity**, Because the effort and time consumed for merging small changes is much lower than large changes.
- **detecting source of errors becomes much simpler**, Because It is much easier for QA engineers to validate small changes iteratively in a consistent system.
- **Features can get into the hands of customers in an earlier state**. This means that if we're on the wrong track, we can change direction after only a small investment.
- **Developers gets extra energy and more productivity**, because of the gained confidence in delivering changes that might have seemed risky before.