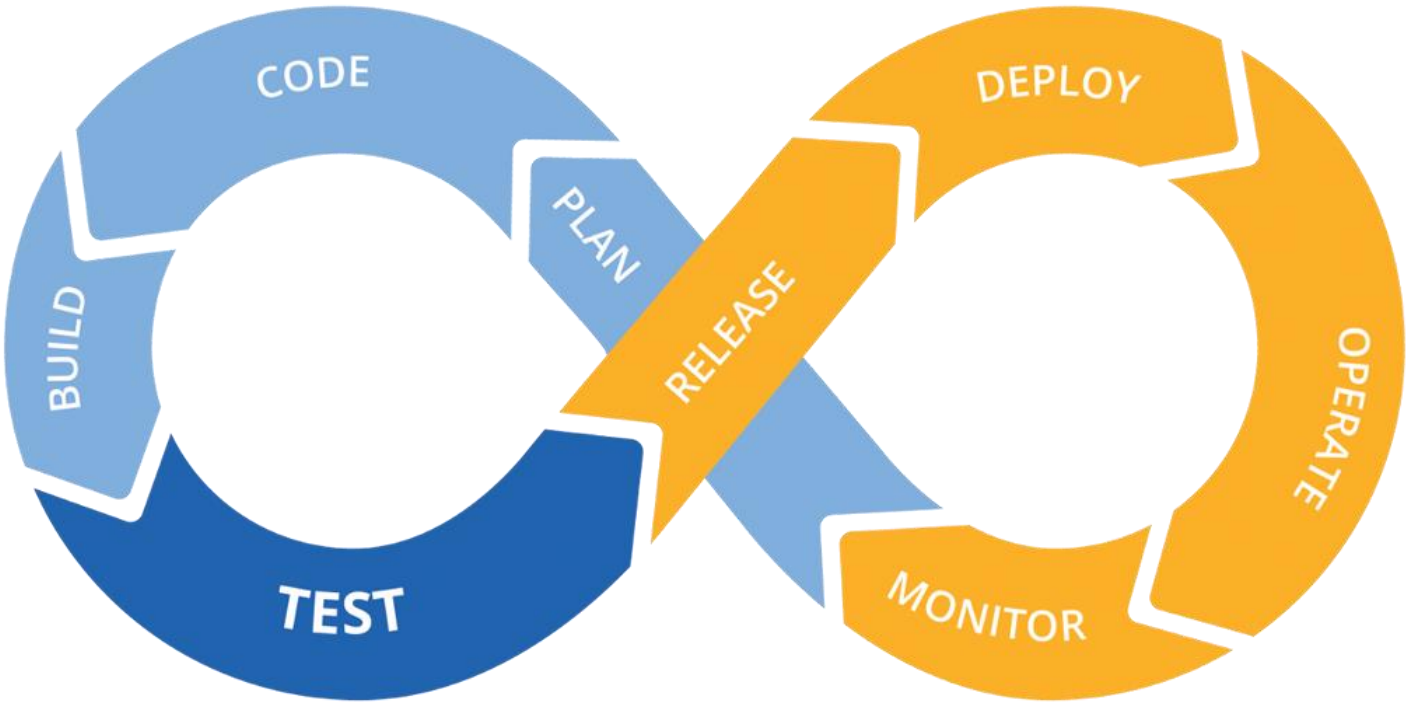
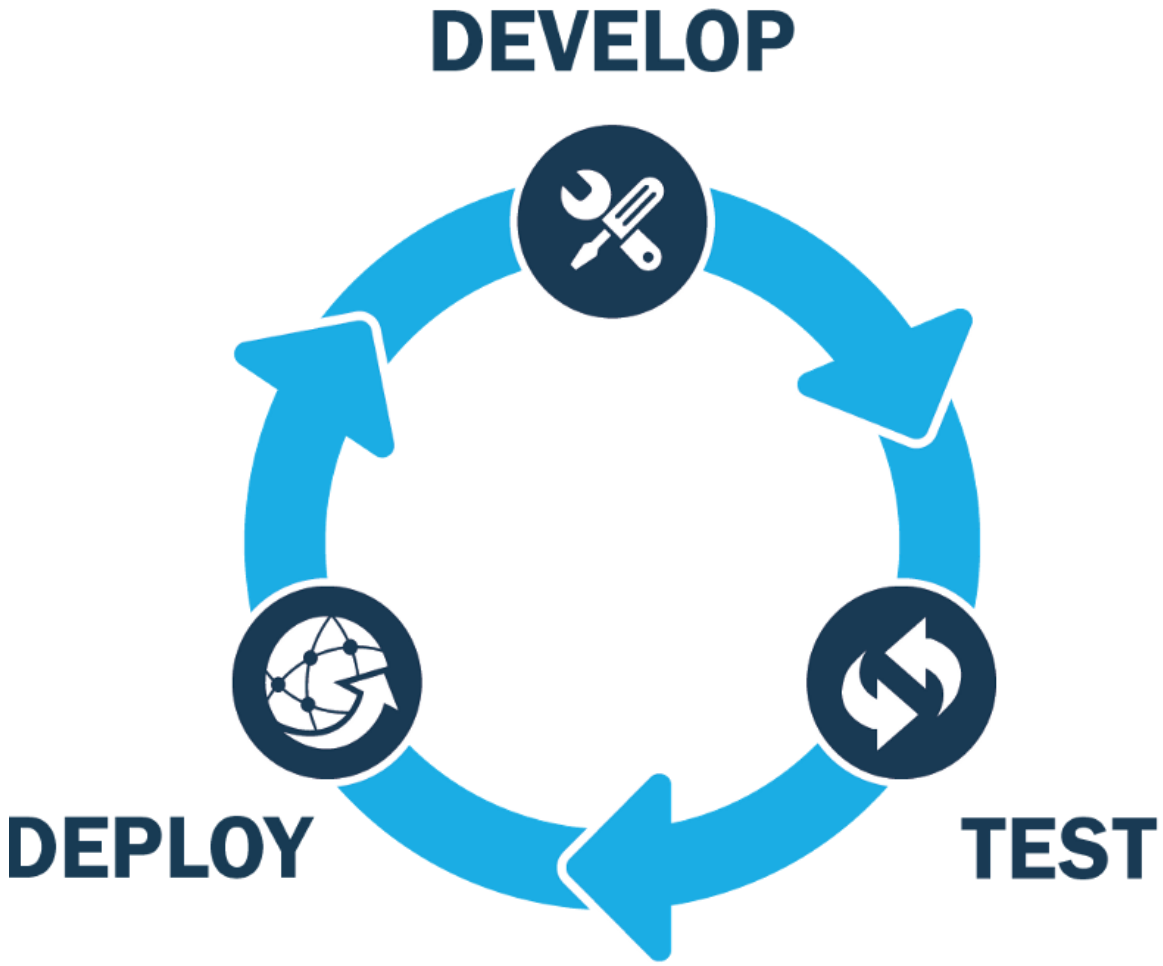
CI/CD – Smartest way
To deliver our products

Fundamentals and Benefits of CI/CD to Achieve, Build, and Deploy automation for our products.





Continuous Integration:

using automation tools to build, test and merge code seamlessly. This is to ensure that code coming from different developers and components is free of errors. Automated tests help spot bugs at the early stage of software development and fix them immediately.

Some of the steps in this stage include: compiling, testing, running static

analysis, checking for vulnerabilities in our dependencies and storing the code artifacts.

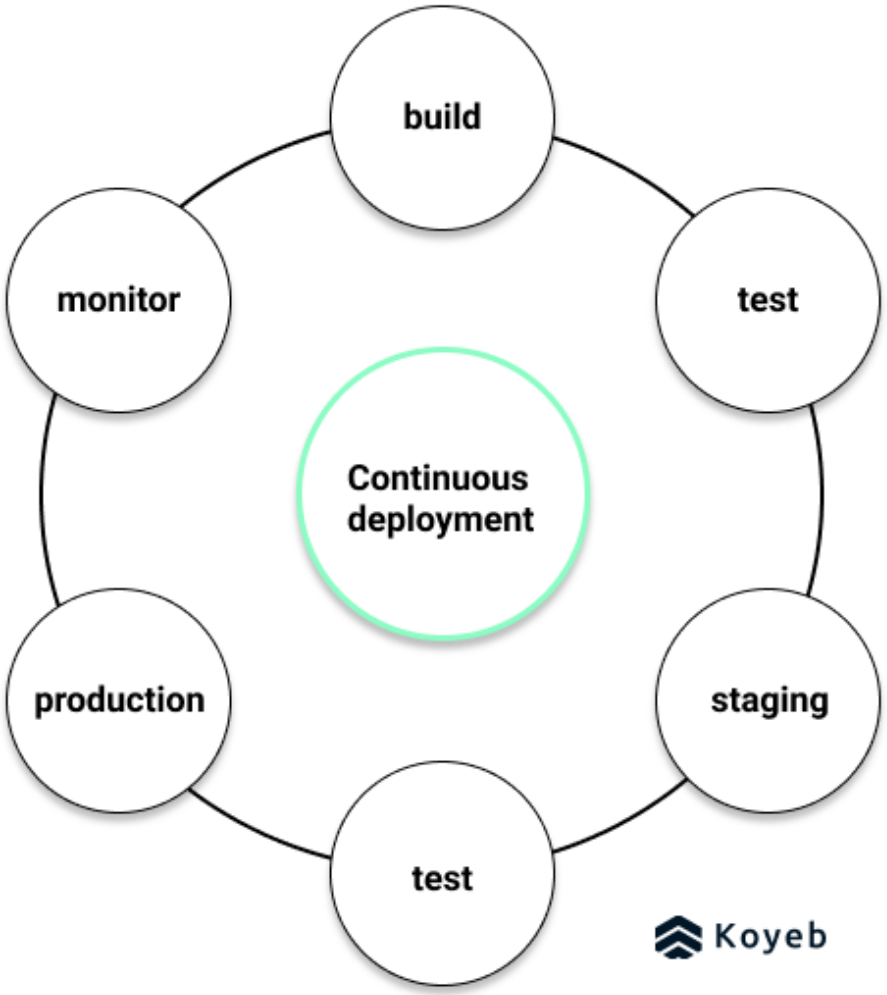
Continuous (Delivery -Deployment) (CD)

- **<u>Continuous Delivery:</u>**

  is the practice of releasing software in short cycles, with greater speed and frequency. New code is released in smaller batches, even a few times a day. This makes it more bug-resistant and easier to manage.

- **<u>Continuous Deployment:</u>**
  This is the process by which verified changes in codebase or system architecture are deployed to production as soon as they are ready and without human input.

  Some steps in this stage include: setting up infrastructure, provisioning servers, copying files, smoke testing, promoting to production and even rolling back a change if something did not look right.



Business benefits of CI/CD:

- **Higher efficiency**

  You should automate your process if you have a review process that includes deploying code to development, testing, and production environments and entering multiple commands across several domains

- **Reduced risk of defects**

  Finding and resolving defects late in the development process is costly and time-consuming. This is particularly true when problems arise with features already released to production.
  You can test and deploy code more frequently using a CI/CD pipeline, giving QA engineers the power to identify and fix errors as soon as they occur. This way, you're effectively mitigating risks in real-time.

- **Faster product delivery**

  With a smooth CI/CD workflow, multiple daily releases can become a reality.

  Teams can automatically build, test, and deliver features with minimal manual intervention Log generation

- **Quick rollback if required**

  quick and easy rollback of code changes if there are any issues in the production environment after a release.

- **Better planning**

  enables organizations to accomplish this by ensuring that they have a well–organized surplus of items and a continuous line of communication with clients.

- **Efficient testing & monitoring**

  Using continuous monitoring, Ops teams can oversee and ensure that the application is running as expected and that the environment is stable. They must ensure that the applications perform optimally

- **Cost-effectiveness**

  With automated testing hooks at every stage, developers can fix issues early and avoid critical issues in the production environment. With the implementation of a CI/CD pipeline, code quality improves drastically.

- **The below tables reflects some benefits of CI/CD from Cost and downtime prospective:**

Technical	Value	Translation
Automate Infrastructure Creation and clean up	Avoid Cost	Eliminating human errors and avoid unnecessary cost of unused or invalid infrastructure
Faster to production	Increase revenue	By automating the pipeline to production this way we can deploy features as soon as created
Automated Rollback Triggered by Job Failure	Reduce cost and decrease downtime	Automate the process of rolling back and cleaning any infrastructure left
Automated Smoke Tests:	Reduce cost and decrease downtime	Automate smoke test after deployment and automatic rollback in case of failure
Catch Compile Errors After Merge	Reduce cost	Discover errors as soon as the developer make his commit