

The background is a solid blue gradient. Overlaid on this are several sets of thin, white, curved lines that flow from the left side towards the right, creating a sense of motion and depth. These lines are more densely packed in some areas, forming wave-like shapes.

CONTINUOUS INTEGRATION
CONTINUOUS DEPLOYMENT

OVERVIEW

Continuous Integration

The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making". Everything related to the code fits here.

Continuous Deployment

A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "Moving" the artifact from the shelf to the spotlight.

Continuous Delivery

An engineering practice in which teams produce and release value in short cycles.

Continuous
Integration + Continuous
Deployment = Continuous
Delivery

BENEFITS OF CICD

- | | |
|------------------------------------|---|
| • Reduce Risk | Catch compile errors and fixing bugs giving testers the ability to detect issues as soon as they occur and to fix them immediately which mitigating risks in real time and reduce cost |
| • Deliver faster | Teams can build, test and deploy features automatically with almost no manual intervention which increase revenue. |
| • Expend less manual effort | After the tests run, the code gets deployed to different environments, including QA, staging and production. |
| • Generate extensive logs | Extensive logging information is generated in each stage of the development process. There are various tools available to analyze these logs effectively and get immediate feedback about the system. |
| • Make easier rollbacks | Immediately return the application to its previous state. Usually, the last successful build gets immediately deployed to prevent production outages and protect revenue. |