

Understanding the CI/CD Pipeline

What It Is, Why It Matters



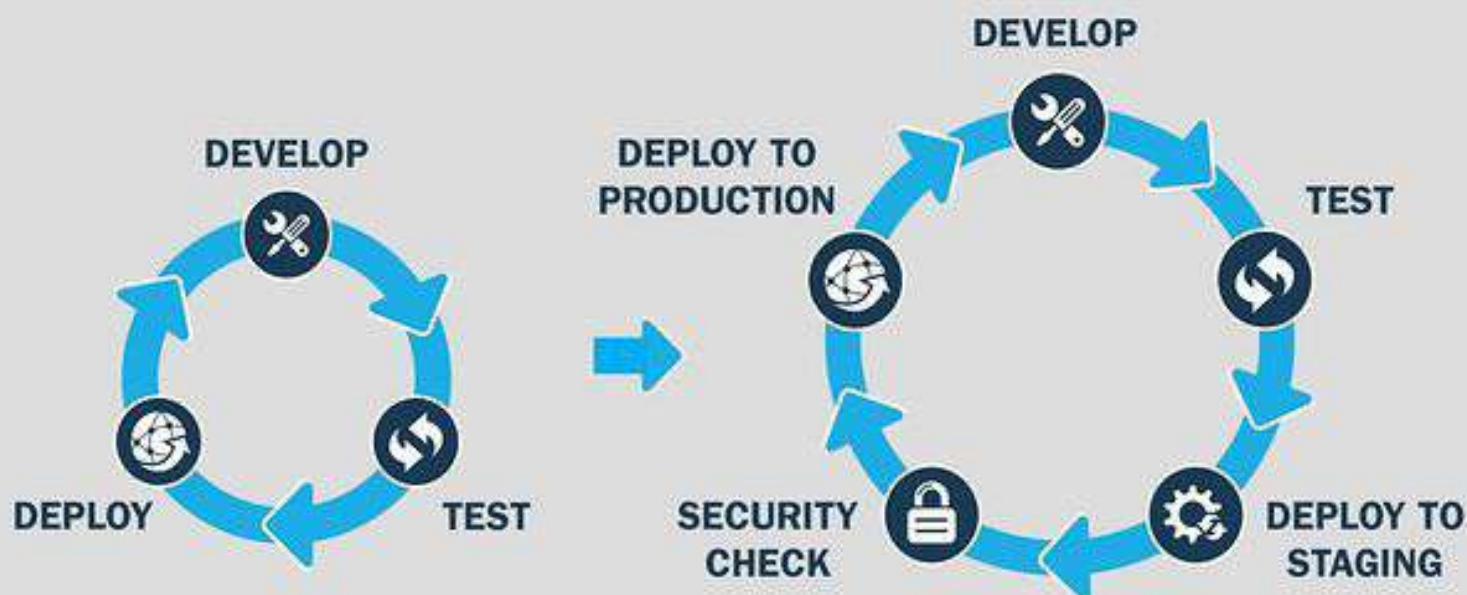
*The backbone of the modern
DevOps environment*

sudoconsultants.com

Cloud | DevSecOps | AWS Partner



01 Continuous Integration (CI)

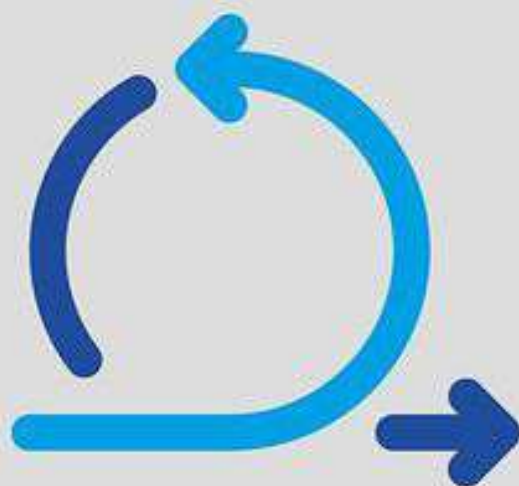


Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

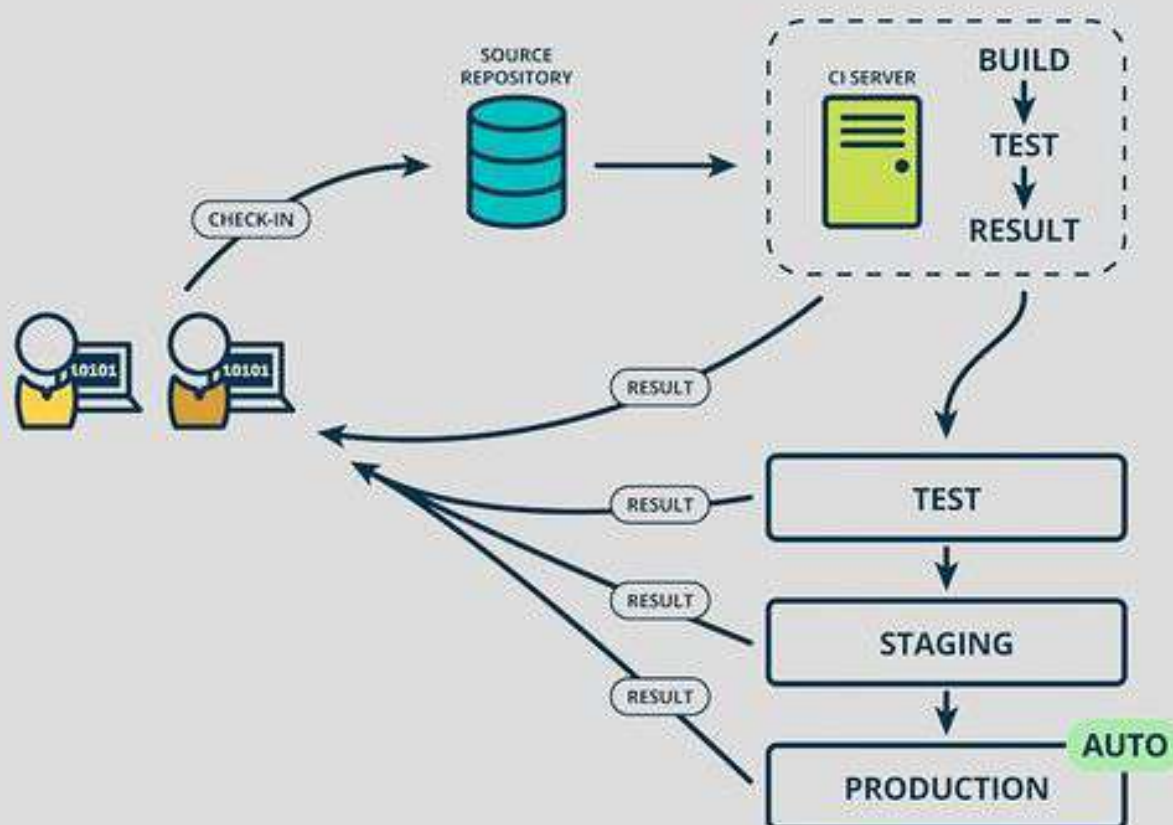


02 Main principles of CI

- Check in code in frequently.
- Automate the build and test portion.
- Always test the code locally before checking it in.
- Never merge any failed branches to the main branch.
- Return its status back to successful if you're the developer who causes the failed build or test.
- Make it your top priority to do so once the fail happens.



03 Continuous Deployment (CD)

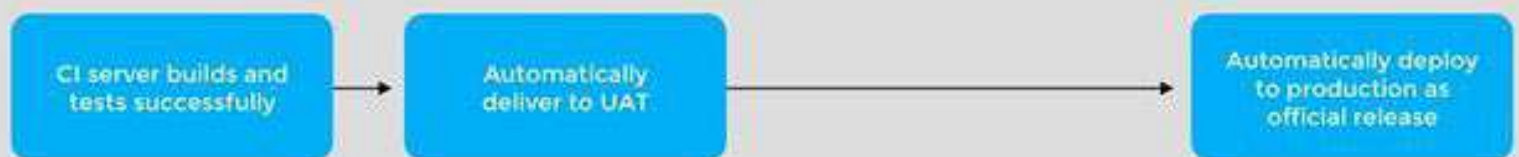


Continuous Deployment is closely related to Continuous Integration and refers to the release into production of software that passes the automated tests.

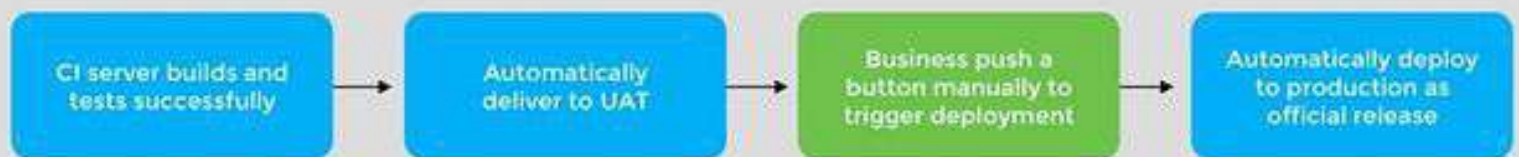


04 Difference Between Continuous Deployment and Continuous Delivery

Continuous Deployment



Continuous Delivery



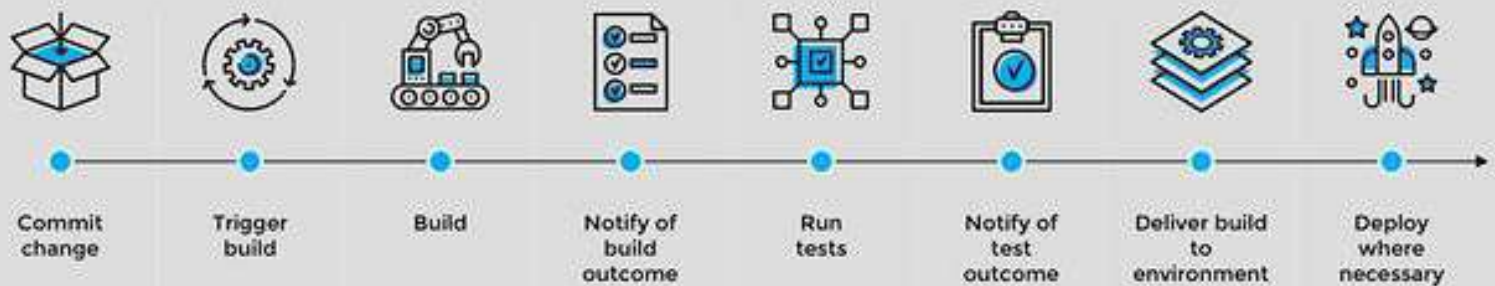
Keep in mind that in most enterprise settings, the business side prefers to manually trigger the deployment.



05 What Is the CI/CD Pipeline?

A CI/CD pipeline is simply the process pathway by which we can deliver a single unit of production-ready software.

CI/CD Pipeline



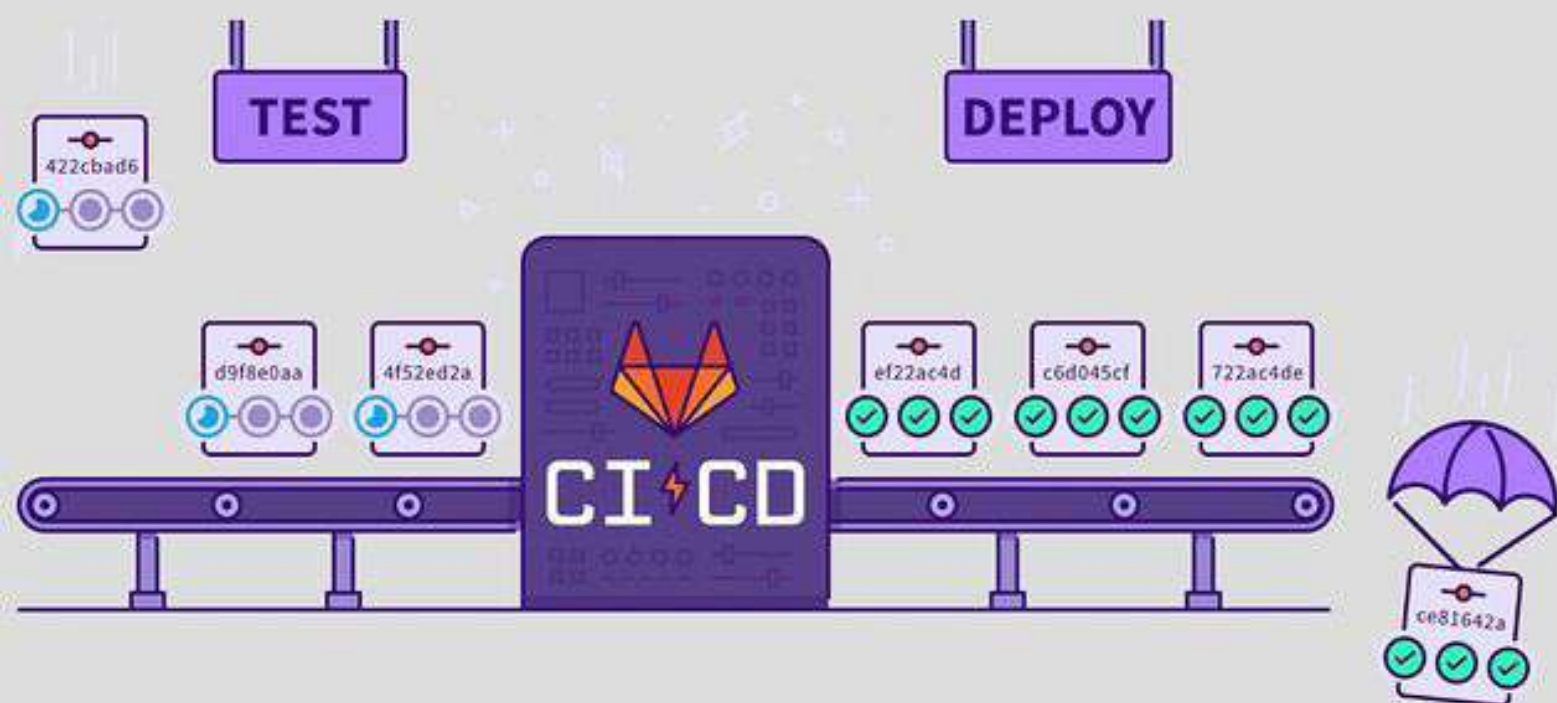
sudoconsultants.com

Cloud | DevSecOps | AWS Partner



06 The CI/CD Pipeline Concentrates Resources on What Matters

A significant advantage of using a CI/CD pipeline is that it reduces costs, complexity and increases reliability.



sudoconsultants.com

Cloud | DevSecOps | AWS Partner

