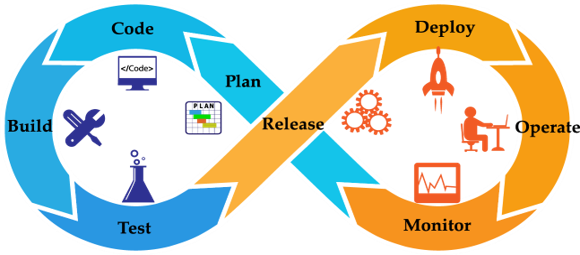
What is Continuous Integration?

Continuous Integration (CI) is a DevOps software development practice that enables the developers to merge their code changes in the central repository. That way, automated builds and tests can be run. The amendments by the developers are validated by creating a built and running an automated test against them.

In the case of Continuous Integration, a tremendous amount of emphasis is placed on testing automation to check on the application. This is to know if it is broken whenever new commits are integrated into the main branch.



What is Continuous Delivery?

Continuous Delivery (CD) is a DevOps practice that refers to the building, testing, and delivering improvements to the software code. The phase is referred to as the extension of the Continuous Integration phase to make sure that new changes can be released to the customers quickly in a substantial manner.

This can be simplified as, though you have automated testing, the release process is also automated, and any deployment can occur at any time with just one click of a button.

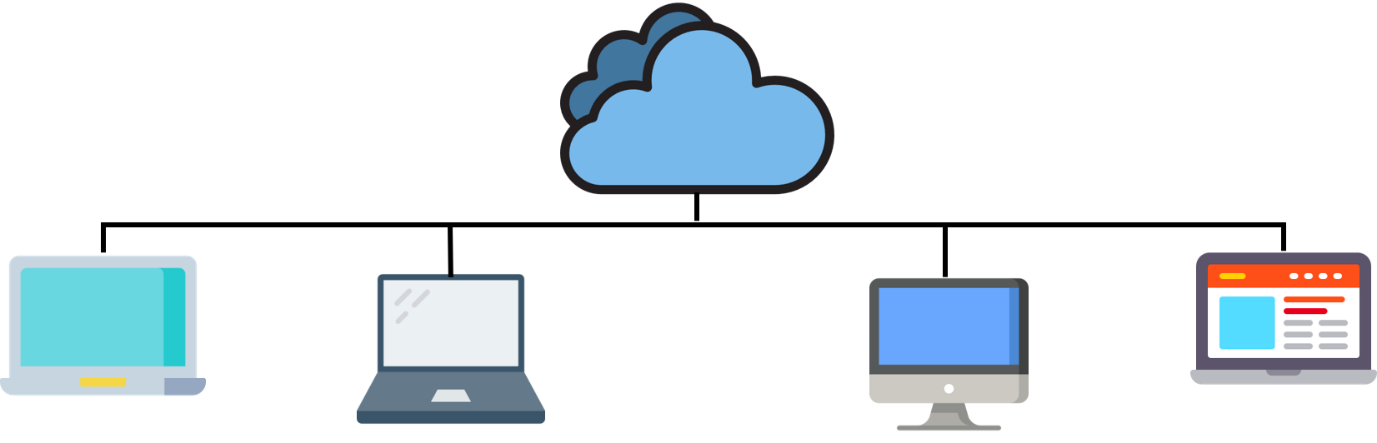
Continuous Delivery gives you the power to decide whether to make the releases daily, weekly, or whenever the business requires it. The maximum benefits of Continuous Delivery can only be yielded if they release small batches, which are easy to troubleshoot if any glitch occurs.



What is Continuous Deployment?

When the step of Continuous Delivery is extended, it results in the phase of Continuous Deployment. Continuous Deployment (CD) is the final stage in the pipeline that refers to the automatic releasing of any developer changes from the repository to the production.

Continuous Deployment ensures that any change that passes through the stages of production is released to the end-users. There is absolutely no way other than any failure in the test that may stop the deployment of new changes to the output. This step is a great way to speed up the feedback loop with customers and is free from human intervention.



After the basics of all three concepts, it's essential to understand how these three processes relate to each other.

## How Are Continuous Integration, Continuous Delivery, and Continuous Deployment Related to Each Other?

To understand the relationship between the three processes, we can consider a car assembly line's analogy. In the car assembly line, it is to be ensured that the individual parts are of the best quality, and they are correctly installed. This entire process comes under Continuous Integration.

While ensuring that the assembled car is fully efficient, free from bugs comes under Continuous Deployment. If the car comes out to be as planned, and everything is implemented successfully, the entire process can now be controlled at a single push of a button. The only difference between Continuous Delivery and Continuous Deployment is that the step of delivery is made automatically.

Now, let's have a look at all the three processes in a sequential order to understand how they are related. The CI/CD pipeline begins with the process of Continuous Integration.

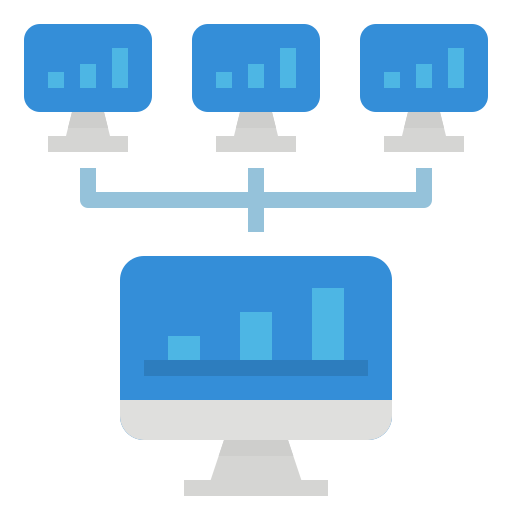
* The process of Continuous Integration allows the developers to integrate code changes into a shared repository continuously. The code in this process is continually tested and incorporated more smoothly.

Reference: <https://www.simplilearn.com/tutorials/devops-tutorial/continuous-delivery-and-continuous-deployment>



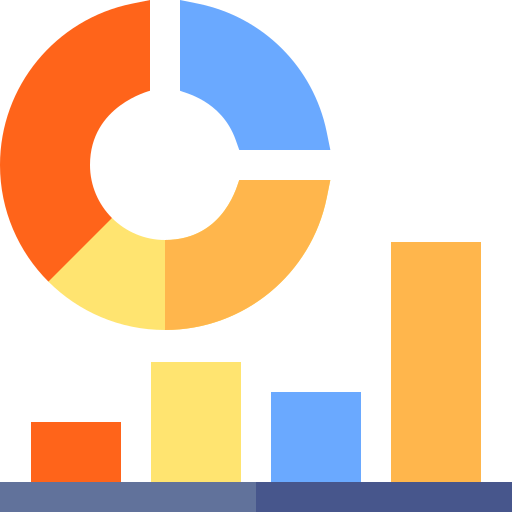
After the Integration phase, comes the Continuous Delivery phase.

* The changes made in the code go through several fixes and feedback before going to the next phase. The team in this phase decides what is to be deployed to the customers and when.



The next and the final phase is the Continuous Deployment phase.

* The step to Delivery and Deployment share the common goal of automating the development process. Sometimes, Continuous Delivery and Continuous Deployment are combined to yield maximum outputs.



Let us now have a look at the features of these practices.