**Continuous integration vs. continuous delivery vs. continuous deployment**

**Continuous Integration:**

* Developers merge their changes back to the main branch as often as possible.
* The developers changes are validated by creating a build and running automated test against the build
* By doing so, you will avoid the integration hell
* Continuous integration puts great importance on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.

**Continuous Delivery:**

* Continuous delivery is an extension of continuous integration to make sure that you can release new changes to your customer quickly in a sustainable way
* This means that on top of having automated your testing, you also have automated release process and you can deploy your application at any point of time by clicking on a button
* In theory, with continuous delivery, you can decide to release daily, weekly, or whatever suites your business requirements. However, if you want to get the benefits of continuous delivery, you should deploy to production as early as possible to make sure that you release small batches

**Continuous Deployment:**

* Continuous deployment goes one step further than continues delivery.
* With this practice, every change that passes all stages of your production pipeline is released to your customer.
* There is no human intervention, and only a failed test will prevent a new change to be deployed to production