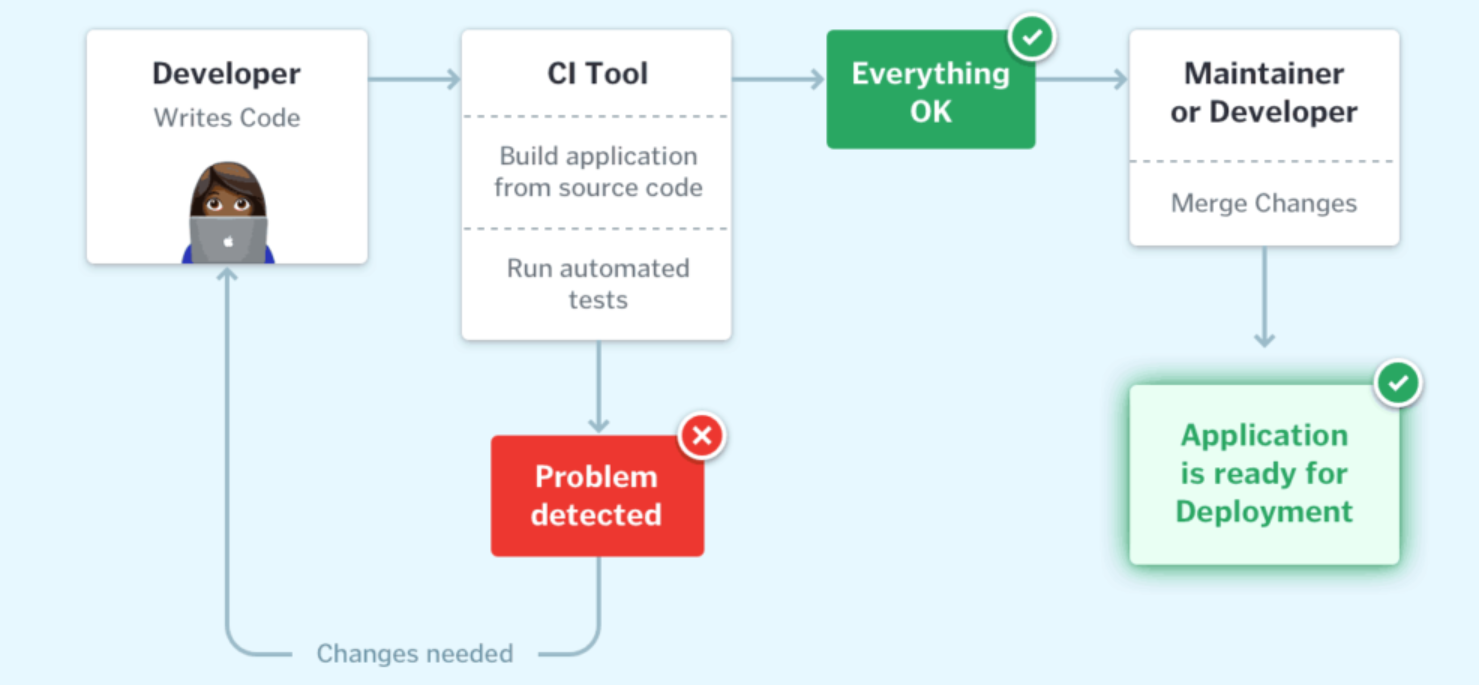
**Continuous Integration**

Continuous Integration is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests if everything goes well the code is ready for deployment or delivery.



Continuous integration has following steps :

* Continuous Development ( github , maven).
* Continuous Testing(junit, selenium).
* Continuous Deployment(docker, aws).
* Continuous Monitoring(Nagios).

The below are the various Continuous integration tools with their features :

**Jenkins :**

* Open source
* Developed by using java programming language
* Easy to setup when compared to other ci tools
* Has good community support

**Bamboo :**

* Commercial
* Less easy to setup when compared to others
* More user friendly when compared to Jenkins
* Provides support to only licensed users.

**Circle Ci :**

* Multiple tests in parallel using multiple machines
* Usage of cache memory from previous data fetch operations
* Increases the speed of software development in an overall manner

**Team city :**

* Free license upto 100 builds
* Integrated with eclipse , Intelij IDEA , Visual studio
* Easy to configure great user interface.

**Code Ship :**

* Codeship offers automated tests
* No distributed builds
* It contains support only for pro version.

**GitLab CI :**

* All Gitlab code is open sourced
* Quick setup for projects hosted on gitlab
* Good Integration with docker
* File based configuratrion.

**Semaphore :**

* Customizable Stages
* Parallel Execution
* Control flow switches

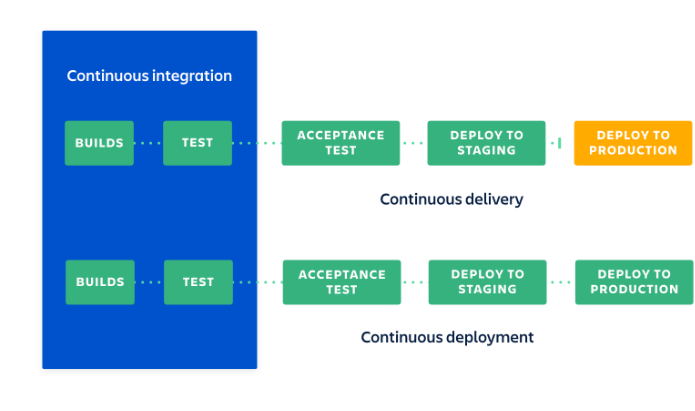
**Soloano :**

* Automatic branch tracking
* Good support
* GitHub Integration.

**Continuous Delivery / Continuous Deployment**

**Continuous Delivery** : It is continued part of continuous integration where the code deployment to production server is not automated and some important things are meant given for manual control here deployment can be done before on the production server environment but it can’t be available to users until any human intervention.

**Continuous Deployment :** In this the code deployment to production server is also automated by using continuous With this practice, every change that passes all stages of continuous integration is released to your customers. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.



Green : automated

Yellow : manual