

# CI/CD

---



# DEFINITION

---

Continuous integration (CI) and continuous delivery (CD) embody a culture, set of operating principles, and collection of practices that enable application development teams to deliver code changes more frequently and reliably. Check the latest 20 CICD Interview Questions DevOps Engineers should clarify in 2020.

# WHAT ARE THE SUCCESS FACTORS FOR *CONTINUOUS INTEGRATION*?

---

- Maintain a code repository
- Automate the build
- Make the build self-testing
- Everyone commits to the baseline every day
- Every commit (to baseline) should be built
- Keep the build fast
- Test in a clone of the production environment
- Make it easy to get the latest deliverables
- Everyone can see the results of the latest build
- Automate deployment

# WHAT IS THE FUNCTION OF *CI (CONTINUOUS INTEGRATION)* SERVER?

---

CI server function is to continuously integrate all changes being made and committed to repository by different developers and check for compile errors. It needs to build code several times a day, preferably after every commit so it can detect which commit made the breakage if the breakage happens.



# ***BLUE-GREEN DEPLOYMENT TECHNIQUE***

---

**Blue-green deployment** is a technique that reduces downtime and risk by running two identical production environments called Blue and Green. At any time, only one of the environments is live, with the live environment serving all production traffic. For this example, Blue is currently live and Green is idle.

As you prepare a new version of your software, deployment and the final stage of testing takes place in the environment that is not live: in this example, Green. Once you have deployed and fully tested the software in Green, you switch the router so all incoming requests now go to Green instead of Blue. Green is now live, and Blue is idle.



# THE DIFFERENCES BETWEEN *CONTINUOUS INTEGRATION*, *CONTINUOUS DELIVERY*, AND *CONTINUOUS DEPLOYMENT*?

---

- Developers practicing **continuous integration** merge their changes back to the main branch as often as possible. By doing so, you avoid the integration hell that usually happens when people wait for release day to merge their changes into the release branch.
- **Continuous delivery** is an extension of continuous integration to make sure that you can release new changes to your customers quickly in a sustainable way. This means that on top of having automated your testing, you also have automated your release process and you can deploy your application at any point of time by clicking on a button.

- **Continuous deployment** goes one step further than continuous delivery. With this practice, every change that passes all stages of your production pipeline 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.



# WHAT ARE THE BENEFITS OF CI/CD?

---

- Reduce risk. Finding and fixing bugs late in the development process is expensive and time-consuming. ...
- Deliver faster. Organizations are moving toward releasing features multiple times a day. ...
- Expend less manual effort. ...
- Generate extensive logs. ...
- Make easier rollbacks.



Thank you