CI/CD Automation

Taking Udapeople to the next level

By - Yash Kankane

The Center Of Attraction!

Continuous Integration

The practice of merging all developers' working copies to a shared mainline several times a day.

Everything related to the code fits here, and it all culminates in the ultimate goal of CI: **a high quality, deployable artifact!**

Continuous Deployment

A software engineering approach in which the **value is delivered frequently** through automated deployments.

Everything related to deploying the artifact fits here.

Continuous Delivery

Continuous delivery is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually. It aims at building, testing, and releasing software with **greater speed and frequency.**

Why CI/CD

Drawbacks in the current approach

- Investing more time in a release cycle than delivering value
- Manual testing fails to detect the bugs on time
- Going through integration hell every time a feature is complete
- Code gets lost because of botched merges
- Unit test suite hasn't been green in ages
- Deployments contribute to schedule slip
- Time consuming manual deployments
- Infrastructure Resources not managed properly
- Deployments are not cause for celebration

Udapeople - Our Needs

Budget

There is a need to manage our budget which can be achieved if a cost-effective tool can aid in saving the time to debug and deploy the application and managing the cost of infrastructure resources efficiently.

Man-power

The team is not big enough to handle both development and operations within the given time frame. There is a need to automate the tasks at hand.

Revenue

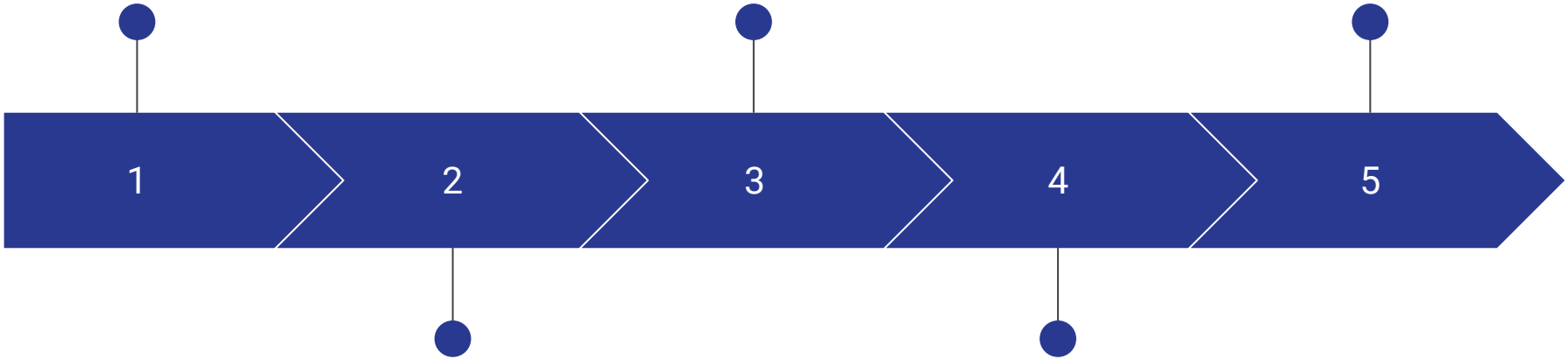
The application needs to generate the revenue in less time as possible. That can only be possible if we deliver the value to our customers on time.

Benefits of CI/CD

Less bugs in production and less time in testing

Less human error, Faster deployments

Less time to market



Less infrastructure costs from unused resources

Reduced downtime from a deploy-related crash or major bug