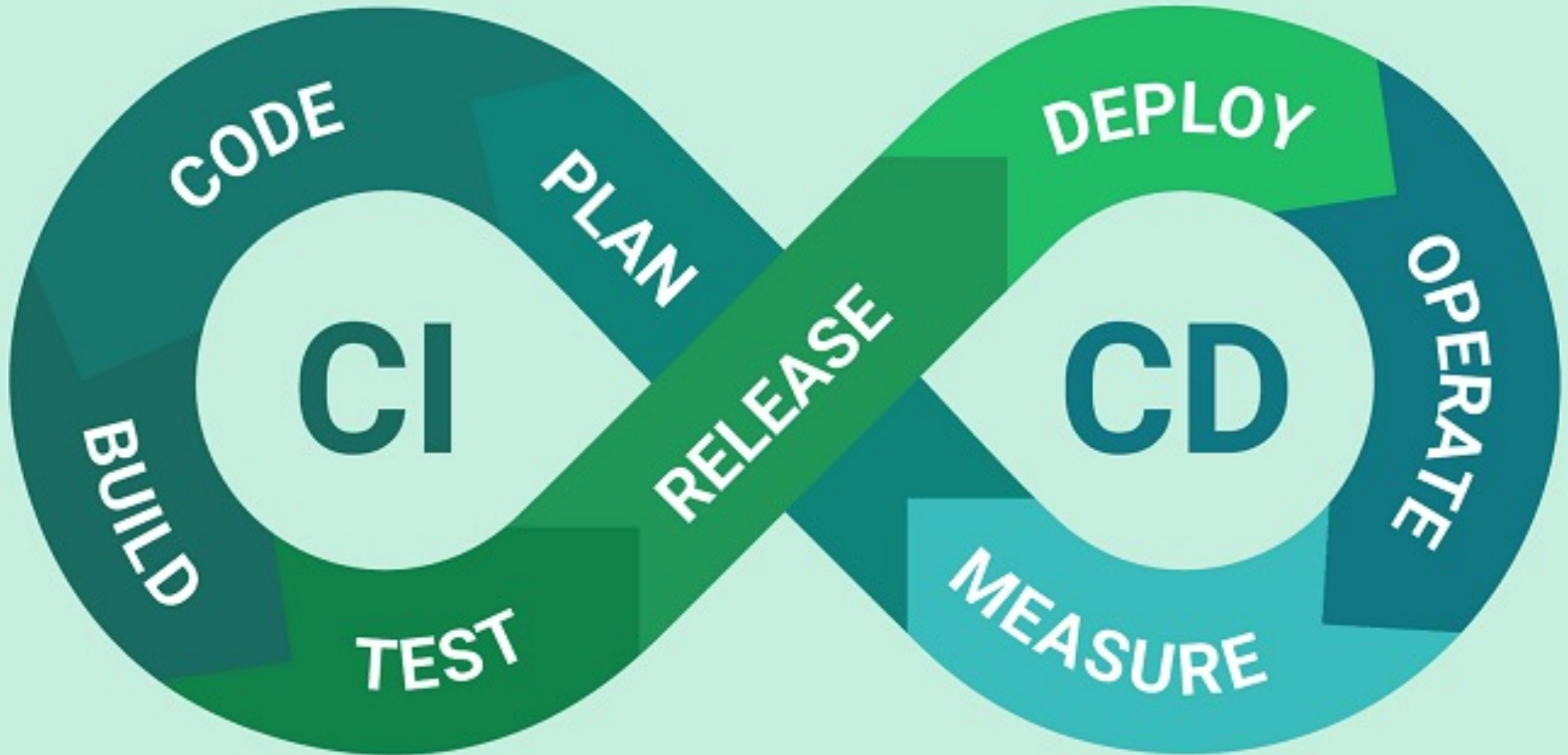


Fundamentals and Benefits of CI/CD

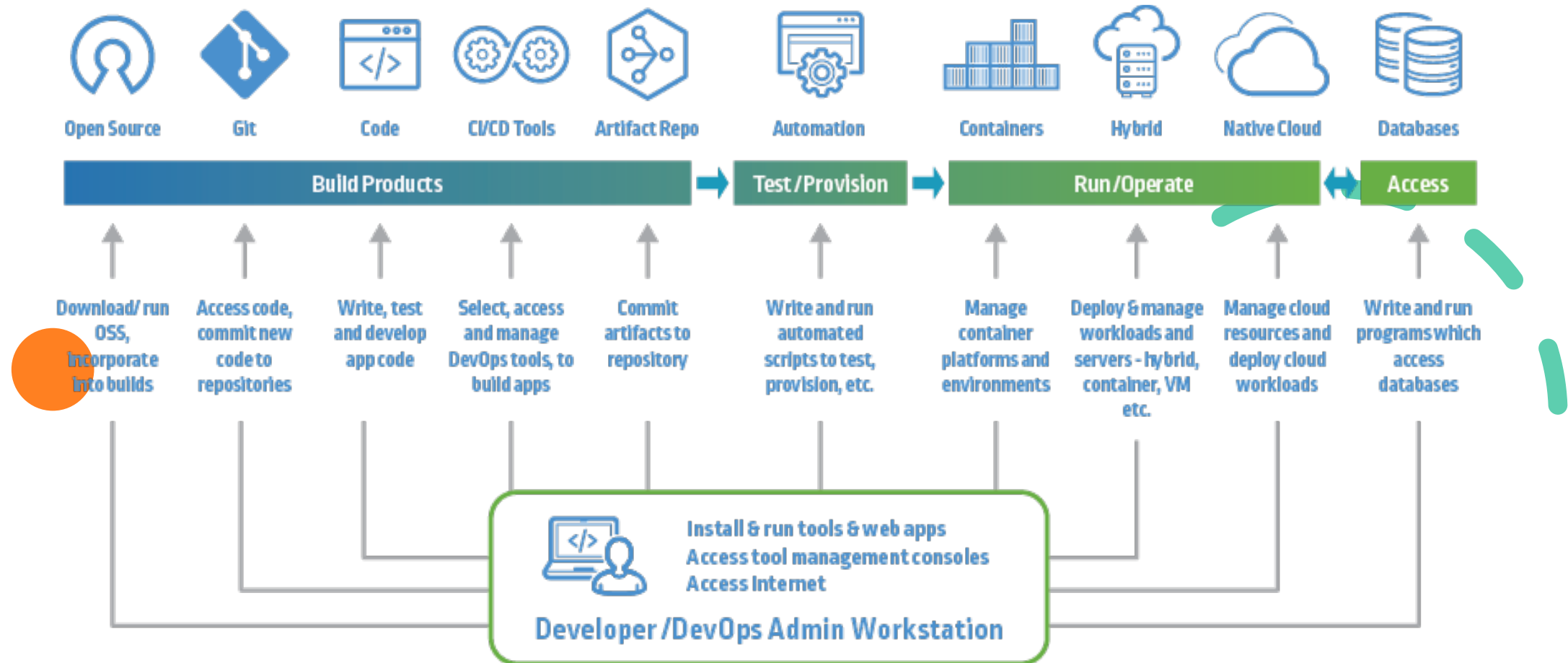


How organizations save cost and deliver faster ?

What is CI/CD?

- Continuous delivery usually means a developer's changes to an application are automatically bug tested and uploaded to a repository (like GitHub or a container registry), where they can then be deployed to a live production environment by the operations team. It's an answer to the problem of poor visibility and communication between dev and business teams. To that end, the purpose of continuous delivery is to ensure that it takes minimal effort to deploy new code.
- Continuous deployment (the other possible "CD") can refer to automatically releasing a developer's changes from the repository to production, where it is usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down app delivery. It builds on the benefits of continuous delivery by automating the next stage in the pipeline.

CI/CD Overview

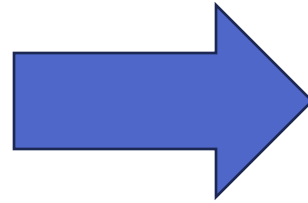


Why CI/CD?

<i>Fail Fast</i>	Set up your CI/CD pipeline to find and reveal failures as fast as possible. The faster you can bring your code failures to light, the faster you can fix them.
<i>Measure Quality</i>	Measure your code quality so that you can see the positive effects of your improvement work (or the negative effects of technical debt).
<i>Only Road to Production</i>	Once CI/CD is deploying to production on your behalf, it must be the only way to deploy. Any other person or process that meddles with production after CI/CD is running will inevitably cause CI/CD to become inconsistent and fail.
<i>Maximum Automation</i>	If it can be automated, automate it. This will only improve your process!
<i>Config in Code</i>	All configuration code must be in code and versioned alongside your production code. This includes the CI/CD configuration files!

Continuous Integration

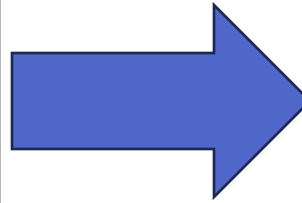
- Reduce code conflict
- Faster code merge
- Catch compile error after merge



- Increase in revenue
- Cost reduction
- Cost avoidance in some cases
- Revenue protection

Continuous Deployment

- Faster and more frequent deployment production deployment.
- Avoid manual interaction by deploying automatically
- Automated rollback in case of failure



- Increase production to market share
- Faster delivery feature for customer