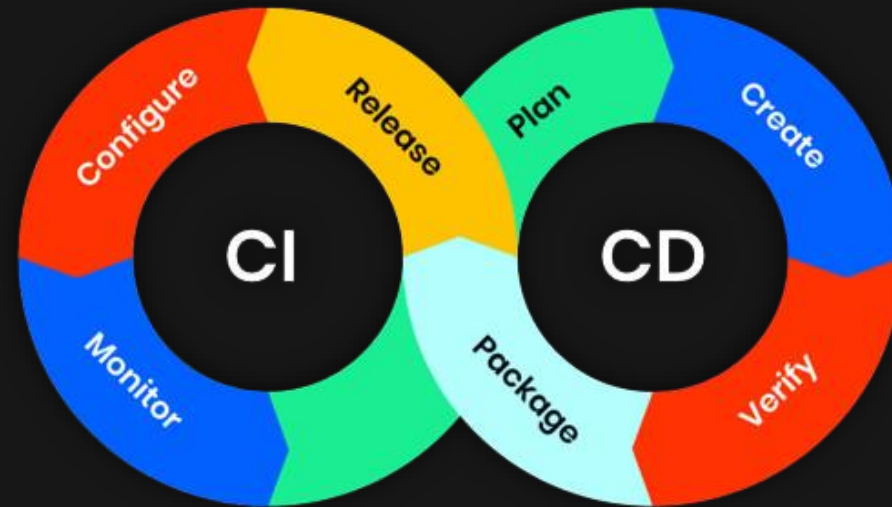


IMPROVE PRODUCT QUALITY WITH CI/CD



BY ALICE WANJOHI

CI/CD OVERVIEW: CONTINUOUS INTEGRATION

- **Continuous Integration (CI)** : A software development practice of merging all developers' working copies to a shared mainline several times a day. Automation tools are used to build, test and merge different codes from multiple developers, allowing errors and bugs to be spotted at the early stages.
- Developers can make smaller changes, commit them with confidence and get feedback on their code sooner, increasing the overall pace of innovation. Ultimately, a high quality, deployable artifact is created.



CODE

Write code
and commit
changes



BUILD

Compile/Lint



TEST

Run unit or
integration tests

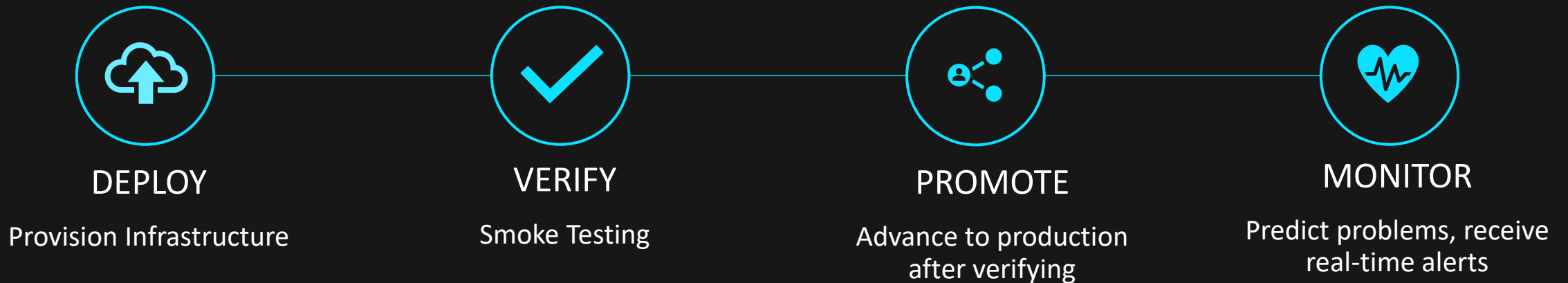


ANALYSIS

deeper tests on the code
such as static analysis or
a package security audit.

CI/CD OVERVIEW: CONTINUOUS DEPLOYMENT

- **Continuous Deployment (CD)** : A software engineering approach in which the value is delivered frequently through automated deployments. Everything related to deploying the artifact fits here. It's the process of "Moving" the artifact from the shelf to the spotlight.
- Any new code update or change that makes it through the rigorous automated test is automatically released into the production environment, making changes that are visible to the software's users.



BENEFITS OF CI/CD

FASTER TIME TO MARKET

- CI/CD is automated with less manual checks, hence delivery of working software to users is quicker and more frequent
- Being able to push changes out quickly means you can respond to new trends and address pain-points as they emerge, increasing customer satisfaction.

EFFICIENT INFRASTRUCTURE

- Infrastructure cleanup and creation is automated, meaning there is reduced human error and faster deployment, which saves cost
- Automated infrastructure deployment is repeatable and reliable, hence less infrastructure costs from unused resources

BUGS ARE ADDRESSED FASTER AND SOONER

- Catching compile errors after merge reduces cost and time spent by developers on issues from new developer code.
- Early detection of unit test failures leads to less bugs in production, which results in less time spent in testing.
- Static analysis helps detect security vulnerabilities, preventing embarrassing or costly security holes.

BENEFITS OF CI/CD

ROLL BACK PROTECTS REVENUE

- In case of job failure, there is automated roll back.
- The ability to quickly undo and return production to working state reduces downtime.

REDUCE RISK

- Fixing defects later in production can be very expensive and time consuming.
- CI/CD helps detect issues as soon as they occur and fix them immediately to mitigate risks in real time.

BETTER RECORD KEEPING AND LOG GENERATION

- The system's performance can be tracked over time more efficiently.
- CI/CD provides a mechanism to study the system in production over time and identify key performance metrics.
- Easier to pin-point issues and map out areas that can be improved.