CI/CD

Revolutionizing the software development life-cycle

By:

Victor Otieno Omondi

CONCEPTS

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

CONTINUOUS DEPLOYMENT:An engineering practice in which teams produce and release value in short cycles.

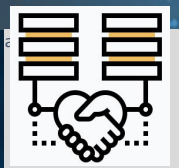
CONTINUOUS DELIVERY: A software engineering approach in which the value is delivered frequently through automated deployments

Why CI/CD?

- Traditional concepts/methodologies of software development could not change much about the value we delivered to our customers.
- Hence, new concepts in DevOps i.e CI/CD enhances how we develop software with value to the end user in mind.
- CI/CD is here to bridge the gap between teams in Devs and Ops therefore harmonize ideas and lead to effective decision-making for both developer and business leaders
- It also enhances confidence for both Dev and Ops teams to gain more ground in polishing features with ease thus making customers happy

Benefits of CI/CD

Practices that reduce costs



- Catch compile after merge

- Catch Unit tests failure as early as after merge

>>Less bugs in production means more time adding amazing features

Practices that avoid costs

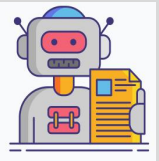
- Detect security vulnerabilities early in the development channel

- Automate Infrastructure creation

>> Less human interaction, less human error, faster and accurate deployments; less cost overhead due to delay

Practices that Increase Revenue

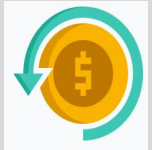
- Faster and frequent deployment to production



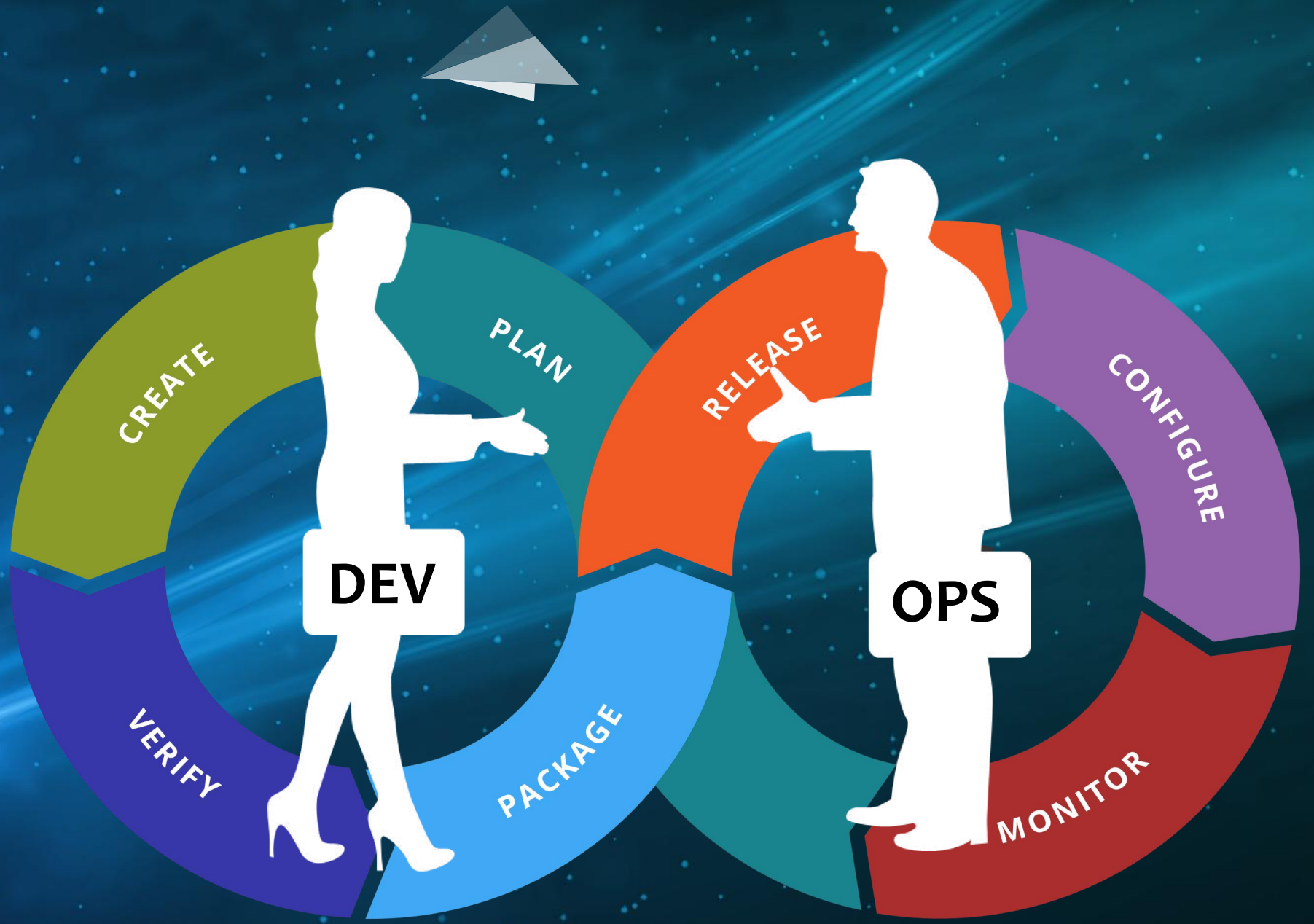
- Deploy to production without manual checks

Practices that Protect Revenue

- Automated smoke tests



- Automated Rollbacks triggered by job failures


CREATE
PLAN
VERIFY
PACKAGE
DEV
RELEASE
CONFIGURE
MONITOR
OPS

THANK YOU

ANY QUESTIONS?