



CONTINUOUS INTEGRATION CONTINUOUS DEPLOYMENT (CI/CD)

PRESENTER: CHRIS IREGBU



WHAT IS CI/CD?

- Modern standard for software delivery
- Continuous Integration (CI)
 - Practice of merging all developers' working copies to a shared mainline several times a day
- Continuous Deployment (CD)
 - Software engineering approach in which the value is delivered frequently through automated deployments
- Continuous Integration + Continuous Deployment = Continuous Delivery



WHY CI/CD?

- Benefits
 - Reducing costs
 - Ability to catch compile errors after each merge allows developers to spend less time on issues from new developer code and focus on creating and delivering new features
 - Ability to automate infrastructure clean up (after each deployment) means less infrastructure costs from unused resources
 - Avoiding costs
 - Ability to catch unit test failures will reduce production level bugs and save on repair bills
 - Ability to prevent embarrassing or costly security loopholes because there will be tests in place to check for security vulnerabilities.
 - Ability to deploy faster and more frequent deployments as infrastructure creation is automated



WHY CI/CD?

- Benefits contd.
 - Increase in revenue
 - Faster and more frequent production deployments will increase revenue as new value generating features are being released more quickly
 - Ability to deploy to production which obviates manual checks increases revenue as features take less time to market
 - Protecting revenue
 - Ability to perform automated smoke (validation) tests because of reduced downtime from deployment related crash or major bug
 - A backout or undo process could restore production to a working state when deployments go wrong as job failures trigger automated rollbacks



HOW WOULD CI/CD BE ACHIEVED?

- Adopt Best Practices
 - Fail Fast – set up the CI/CD pipeline to find and reveal failures quickly and fix the failures
 - Measure Quality to see the positive effects of our improvement (or the negative effects of technical debt)
 - Make CI/CD the only Road to Production to avoid inconsistency and failure
 - Maximum Automation – we will automate anything that can be automated to improve our process
 - Configuration in Code – all configuration will be in code and versioned alongside our production code including the CI/CD configuration files.
- Use Cloud-Based CI/CD Tools – GitHub, Circle CI, Ansible, Amazon AWS