UdaPeople: CI/CD Proposal

Johnny Nguyen (DevOps Engineer)

Overview

Situation/Task:

- Challenges with UdaPeople

Action:

- What is CI/CD?
- Creating a CI/CD Pipeline

Expected Results:

- Business Implications

Situation/Task: Challenges with UdaPeople

- Despite multiple production releases, we see issues:
  - Sales, Software, Quality Assurance teams are siloed
  - Friction between Development and Operations/Production
  - More man-hours spent fixing issues than creating new features
  - Multiple instances of failed builds released to production
  - Manual deployments to production
- **Result:** More schedule slips and less reliability
- **Task:** Increase development velocity without compromising product reliability as one team!

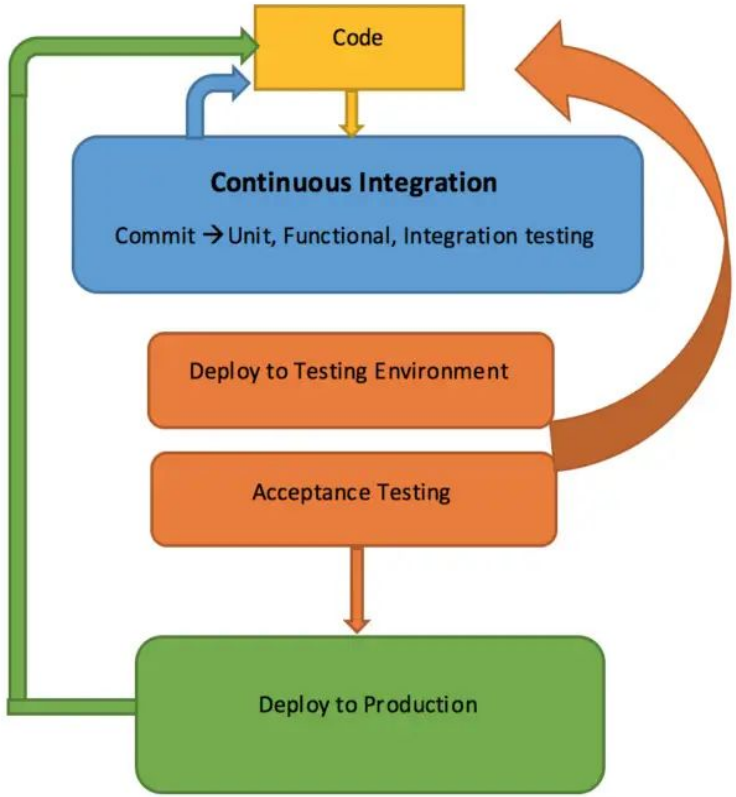
Action: Implement CI/CD Pipeline

What is Devops?

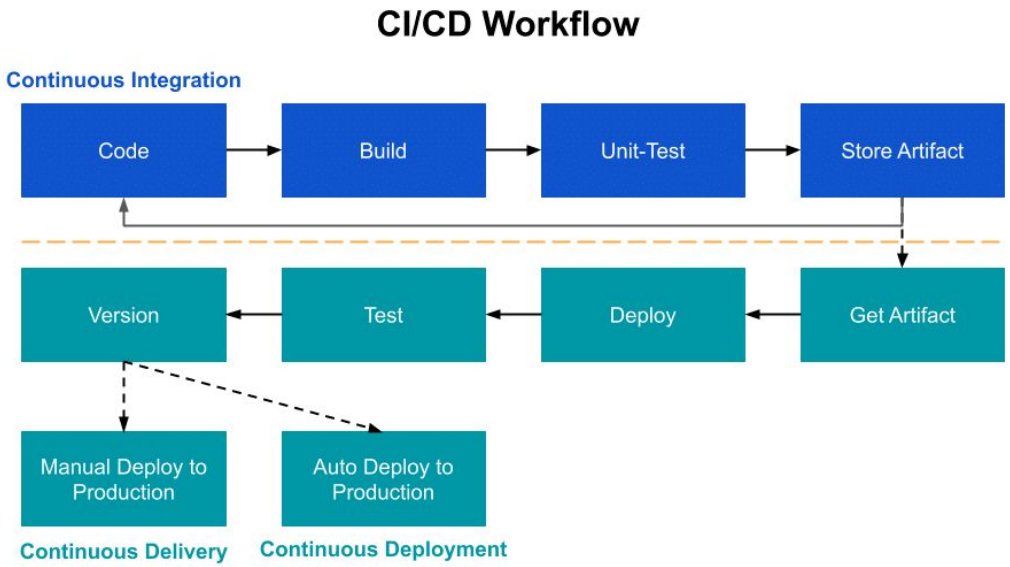
- **DevOps** - Culture, practices, and tools to delivers applications and services at high velocity and reliability

What is CI/CD?

- **CI** (Continuous Integration): Practice of merging all developer code to shared repository several times a day
- **CD** (Continuous Deployment): Software Engineering approach of automated deployments



Action: Creating a CI/CD Pipeline



Proposed Tools

- **CI/CD:**
  CircleCI
- **Configuration Management:**
  Ansible
- **Containers:**
  Docker
- **Container Orchestration:**
  Kubernetes
- **Infrastructure Provisioning:**
  AWS CloudFormation
- **Monitoring:**
  Prometheus/Grafana
- **Cloud Provider:**
  Amazon Web Services

Expected Results: Business Implications

Business Value	**How CI/CD achieves this**	**Result:**
Reduces Cost	- Catches Compile Errors After Merge - Automate Infrastructure Cleanup	- Less developer time on issues from new developer code - Less infrastructure costs from unused resources
Avoids Costs	- Catches Unit Test Failures - Detects Security Vulnerabilities - Automates Infrastructure Creation	- Less bugs in production/testing - Prevent embarrassing or costly security holes - Less human error, faster deployments
Increases Revenue	- Faster and More Frequent Production Deployments - Deploy to Production Without Manual Checks	- New value-generating features released more quickly - Less time to market
Protects Revenue	- Automated Smoke Tests - Automated Rollback Triggered by Job Failure	- Reduced downtime from a deploy-related crash or major bug - Quick undo to return production to working state

Thank you for your time!