# Fundamentals and Benefits of CI/CD

# Fundamentals of CI/CD

- CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.
- The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.
- CI/CD is a **solution to the problems integrating new code** can cause for development and operations teams (AKA "integration hell").

# Continuous Integration

- The practice of merging all developers' working copies to a shared mainline several times a day. It's the process of "Making". Everything related to the code fits here, and it all culminates in the ultimate goal of CI: a high quality, deployable artifact! Some common CI-related phases might include:
  - Compile
  - Unit Test
  - Static Analysis
  - Dependency vulnerability testing
  - Store artifact

# Continuous Deployment

- 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. Some common CD-related phases might include:
  - Creating infrastructure
  - Provisioning servers
  - Copying files
  - Promoting to production
  - Smoke Testing (aka Verify)
  - Rollbacks

# Benefits of CI/CD

Using CI/CD will benefit company by:

- Avoiding Cost
- Reducing Cost
- Increasing Revenue
- Protect Revenue

# Avoiding Cost

Catch Unit Test Failures => Less bugs in production and less time in testing

Detect Security Vulnerabilities => Prevent embarrassing or costly security holes

Automate Infrastructure Creation => Less human error, Faster deployments

# Reduce Cost

Catch Compile Errors After Merge => Less developer time on issues from new developer code

Automate Infrastructure Cleanup => Less infrastructure costs from unused resources

# Increasing Revenue

Faster and More Frequent Production Deployments => New value-generating features released more quickly

Deploy to Production Without Manual Checks => Less time to market

# Protect Revenue

Automated Smoke Tests => Reduced downtime from a deploy-related crash or major bug

Automated Rollback Triggered by Job Failure => Quick undo to return production to working state