* Pipeline
  + workflow of moving a project through stages of work
  + common names: release, build, deployment
* Configuration
  + read by automation process
  + contains configurations properties
  + imperative – how to do
  + declarative – what you want
  + imperative – starting a car

1. open driver side car door
2. get into car
3. locate ignition switch
4. insert key into ignition switch
5. turn ignition
6. press on the gas

* declarative – starting the car

1. start the car

* Push/Pull Target Delivery
  + Push
    - best for testing, great product solution
  + Pull
    - great product solution
* The Release Pipeline
  + **Source** 🡪 **Build** 🡪 **Testing** 🡪 **Release**
    - rinse and repeat. not something you do once
* Continuous Configuration Automation (CCA) Tools
  + referred to as the DevOps Toolchain
  + tools that automate and orchestrate
  + these tools may be specific to a certain platform
  + need to make sure these tools meet your security and compliance needs
  + want tools that will scale

**Stage 1: Source**

* What is "Source"?
  + not just source code – the supported files needed for application and infrastructure
  + source control – storing code and documentation changes
    - Ex.: GitHub, Apache Subversion, Microsoft Visual Studio Team Foundation Server
* Version Control
  + much better than just a file share
  + tracks changes, conflict resolution and maintains a history of changes made from beginning to production
  + identifies "who" made a change and "when"
  + supports team members and can identify and resolve conflicts
* Code Review
  + prior to commit
  + checks code for best practices
  + can be manual but automated is best
  + assure that code is maintainable
  + can also be done during the testing phase – known as *linting*

**Stage 2: Build**

* Turing Code into Product
  + executes files or scripts to create product – binaries or machine readable code
  + connect to source control
  + can be triggered to start a build activity
* Triggering Build Activity
  + think of build process as an orchestration service that can be controlled and triggered
  + web hook – a web service call from source to start a build activity
    - AppVeyor
    - Travis CI
  + polling agent – polls for changes in source
    - team city
    - Jenkins

**Stage 3: Testing**

* Testing vs. Monitoring
  + Testing is the most important part of the release pipeline process
  + functionality is tested in isolation, then often moved to testing environments for integration and acceptance testing
  + monitoring checks for problems after release
  + monitoring alone is NOT a replacement for test automation
    - Ex.: ServerSpec, Rspec, Chef InSpec, Pester, Test-Kitchen, Vagrant
* Impact of TDD
  + a different way of thinking when developing
  + tests are written as small units before coding begins
  + TDD Approach (**Red** **🡪** **Green 🡪 Refractor**)

1. Add tests
2. Run all tests for errors
3. write code
4. run tests
5. refractor code
6. repeat

**Stage 4: Release**

* What does Release mean?
  + after the process successfully completes
  + configuration change is in Source
  + build process completes
  + testing has passed
  + a build artifact is ready to be deployed (e.g., executables, binaries, etc.)
    - Ex.: dependent on environment and situation
* Continuous Delivery (CD)
  + aka Continuous Deployment with *slight changes*
  + best with immutable servers – every server exactly matches each other
  + server can be re-deployed entirely if a failure occurs
    - Ex.: web servers in a farm
* Incremental Deployment
  + for servers that must be maintained over a long period of time
  + for servers that cannot be re-deployed
  + configuration changes are delivered as updates
    - Ex.: database, domain controllers
* Continuous Integration
  + bringing the process all together with a workflow (orchestration)
  + add quality gates (testing) and approval
  + can suspend new changes until failures are resolved