

Configuration Management

ERIN KEITH

CS 333 – TESTING AND DEVOPS

A solid green horizontal bar at the bottom of the slide.

Pipeline

6. Configuring

- Infrastructure configuration and management
- Infrastructure as code tools

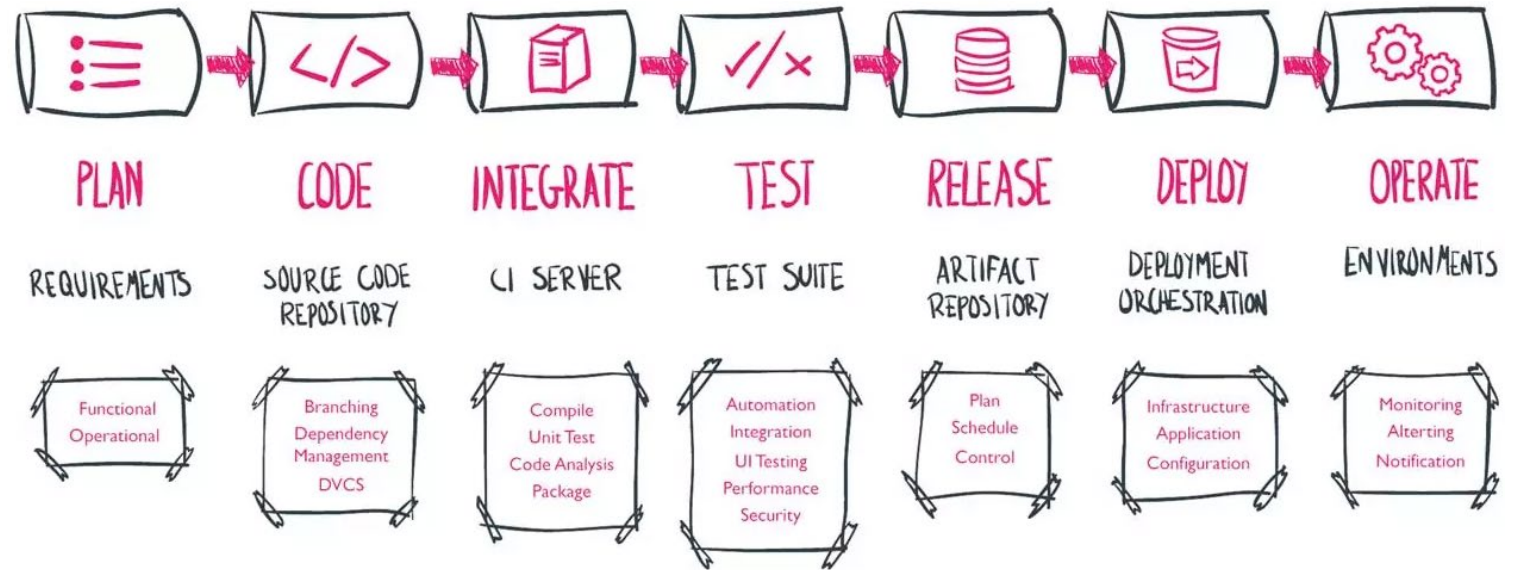
7. Monitoring

- Application performance monitoring
- End-user experience

Pipeline – A Different View

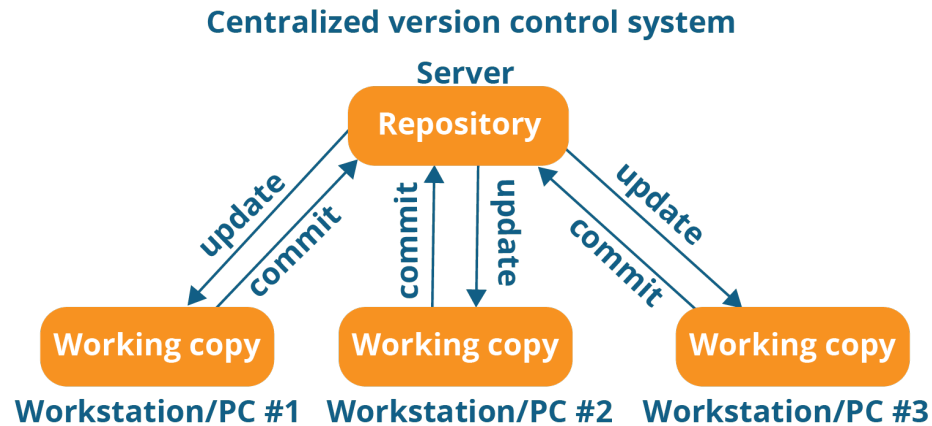
These steps have to happen somewhere: Environments

- Local
- Integration
- Testing
- Staging
- Production

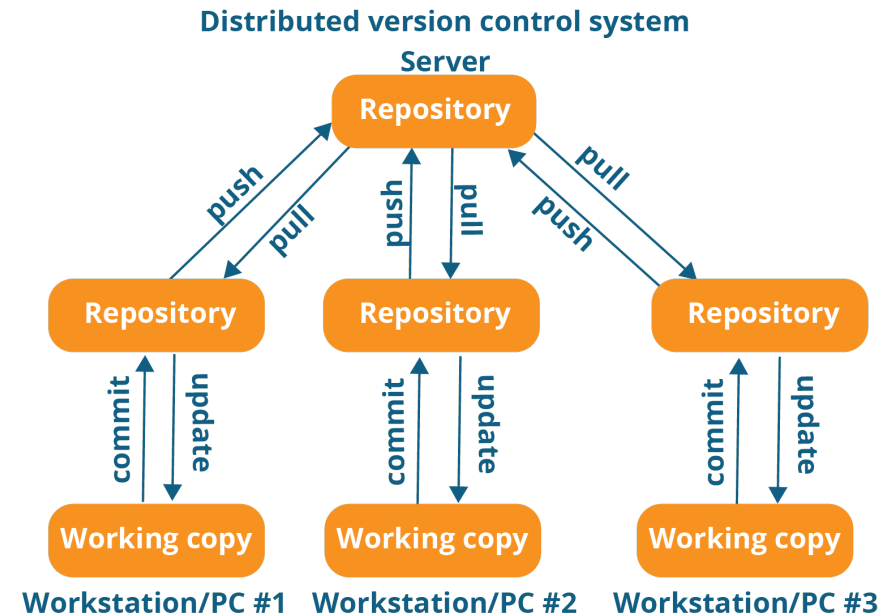


Source Control – Models

SERVER-CLIENT (in-house)



DISTRIBUTED (cloud)




<https://www.edureka.co/blog/what-is-git/>

Integration – Models

also Server vs Distributed BUT
there needs to be a machine
somewhere

GitHub Actions vs Jenkins

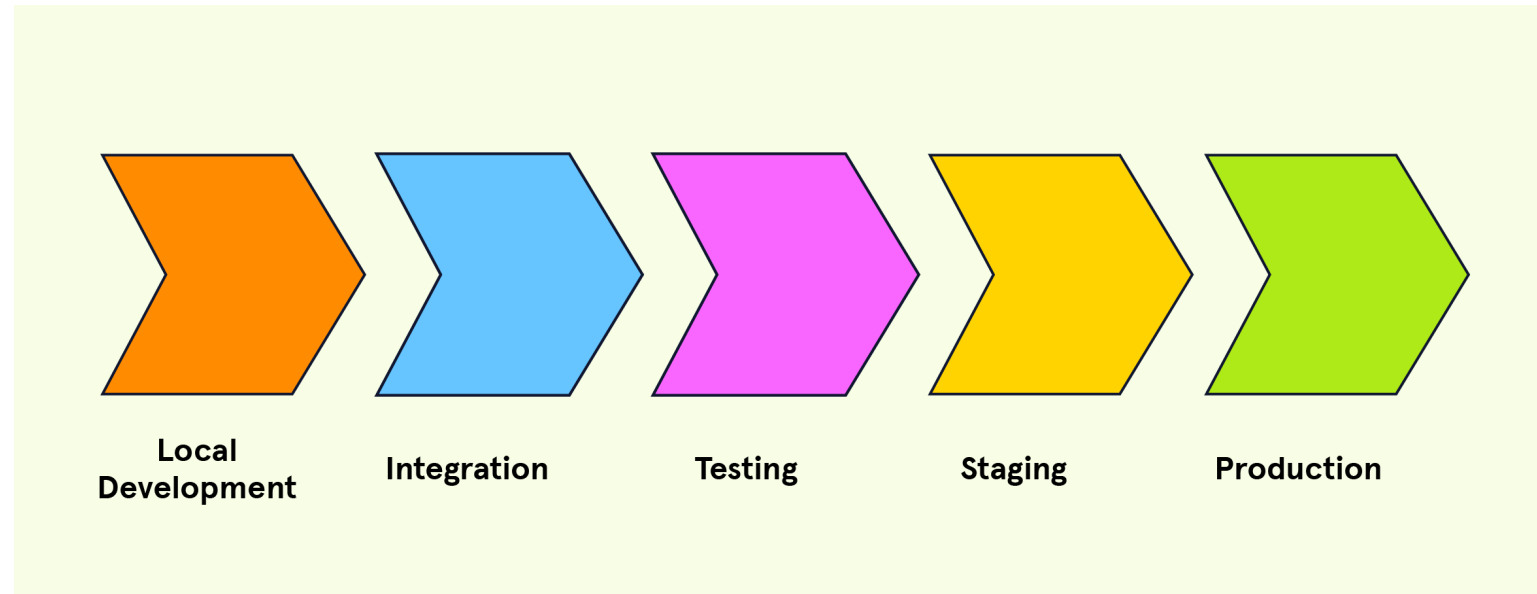
Comparison Chart

GitHub Actions	Jenkins
It is a fully managed service by GitHub which gives you full control over your CI/CD pipelines.	You do not have full control of your CI/CD pipeline workflow.
It operates in the cloud but you can also run it on a local server, which is called a runner.	It is a server based application that runs in servlet containers such as Apache Tomcat.
It provides a stable system with no installation required.	Jenkins server needs installation.
It uses jobs to put one or more steps or individual commands into groups.	It makes use of stages to run a collection of steps.
	

Environments and Configuration

These are on computers
(either local or in the cloud).

- works best if they all match
 - software versions
 - port configurations
- they need all of the necessary dependencies
 - libraries
 - credentials



Example – Python Configuration

Dependencies

- `pip + requirements.txt`
- `venv` for Python versioning

App Configuration

- `ConfigManager` Class

