



Jenkins Workshop Overview

Jenkins configuration through freestyle project

- ❖ Basic project to demo shell / batch commands
- ❖ Using Maven to build basic Java project
- ❖ Setting up GitHub repo for project
- ❖ Integrating Jenkins to pull from GitHub
- ❖ Configuring periodic builds and GitHub polling
- ❖ Chaining jobs / projects
- ❖ Configuring Webhooks with GitHub / BitBucket
- ❖ Configuring email notification for build failures

Jenkins pipeline DSL syntax

- ❖ Pipeline structure: Nodes and processes
- ❖ Platform independent steps
- ❖ Git-Maven pipeline
- ❖ Periodic builds using pipeline syntax
- ❖ Setting up a pipeline as Jenkins file

Web app deployment with Jenkins

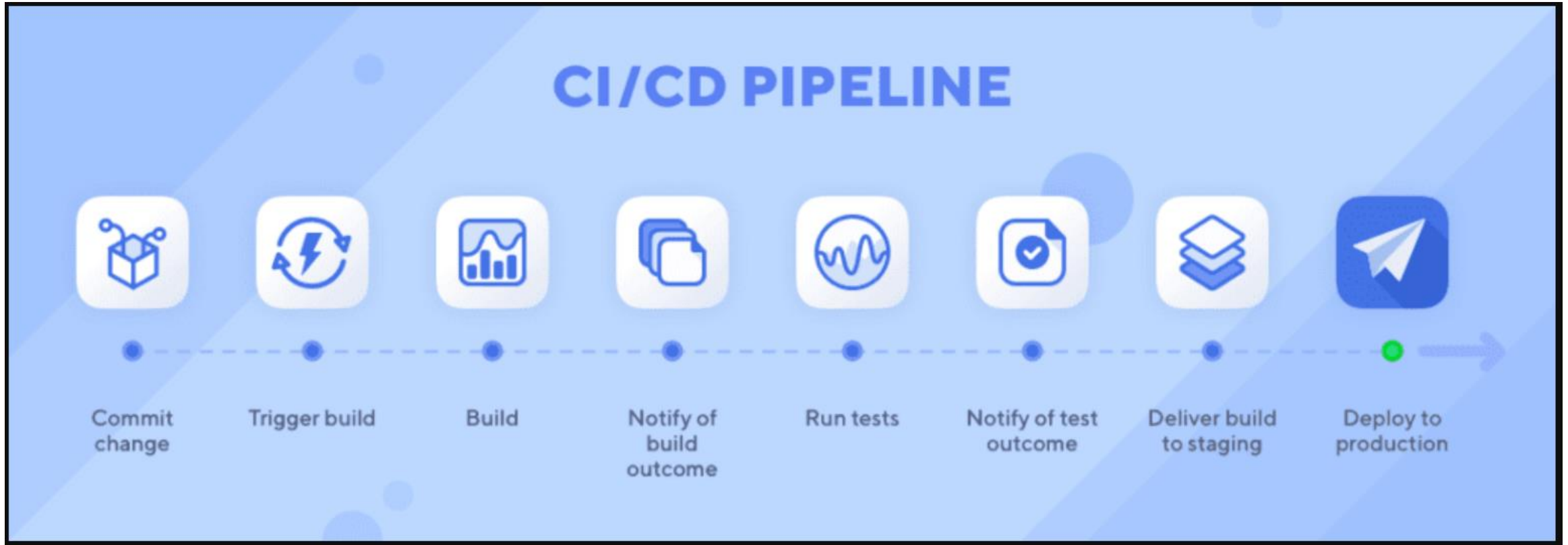
❖ Building / deploying Java web app with Maven

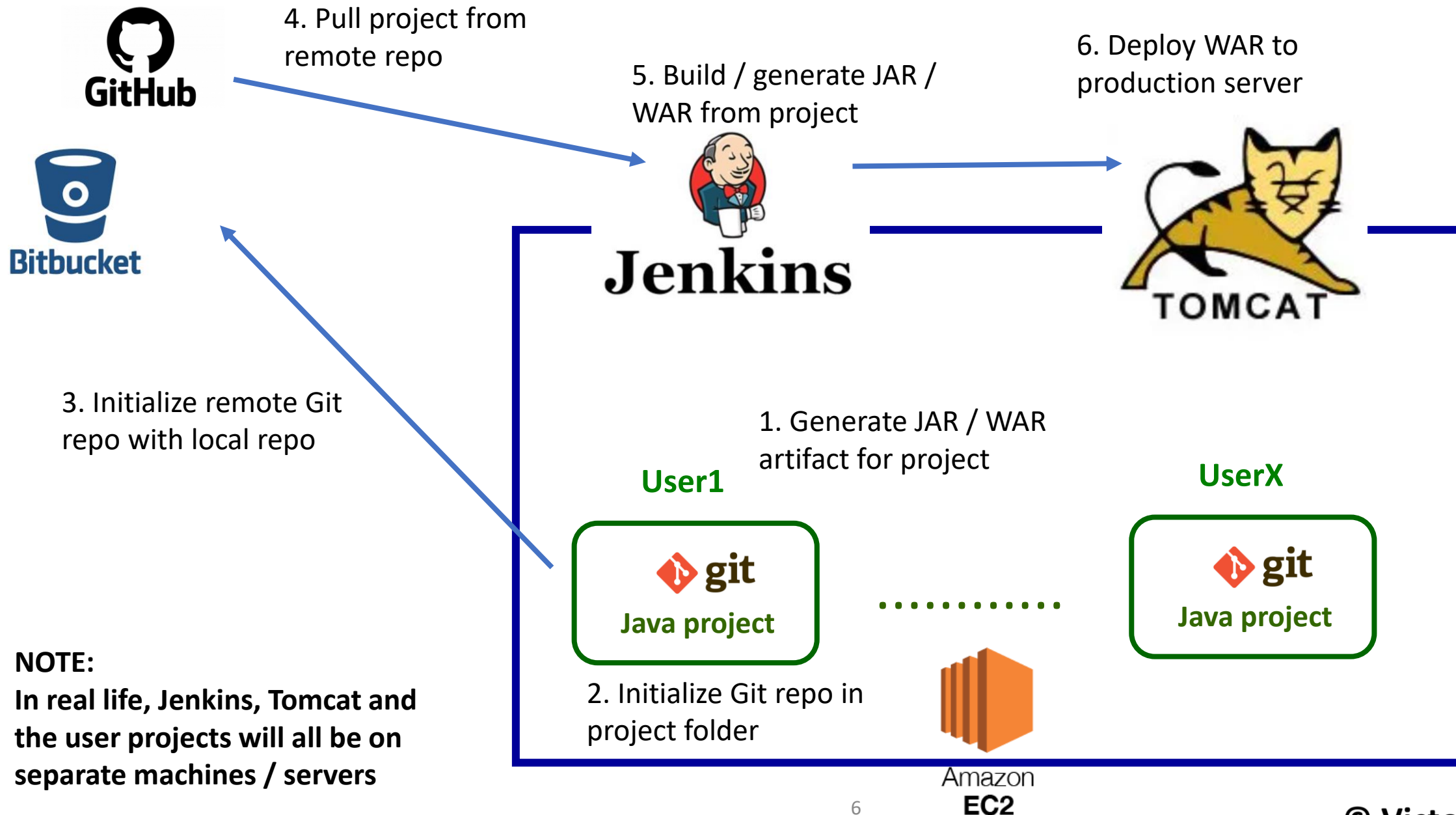
- Configuring Maven / Tomcat integration

❖ Building and deploying to Tomcat container

- Using freestyle project
- Using Jenkins pipeline DSL syntax

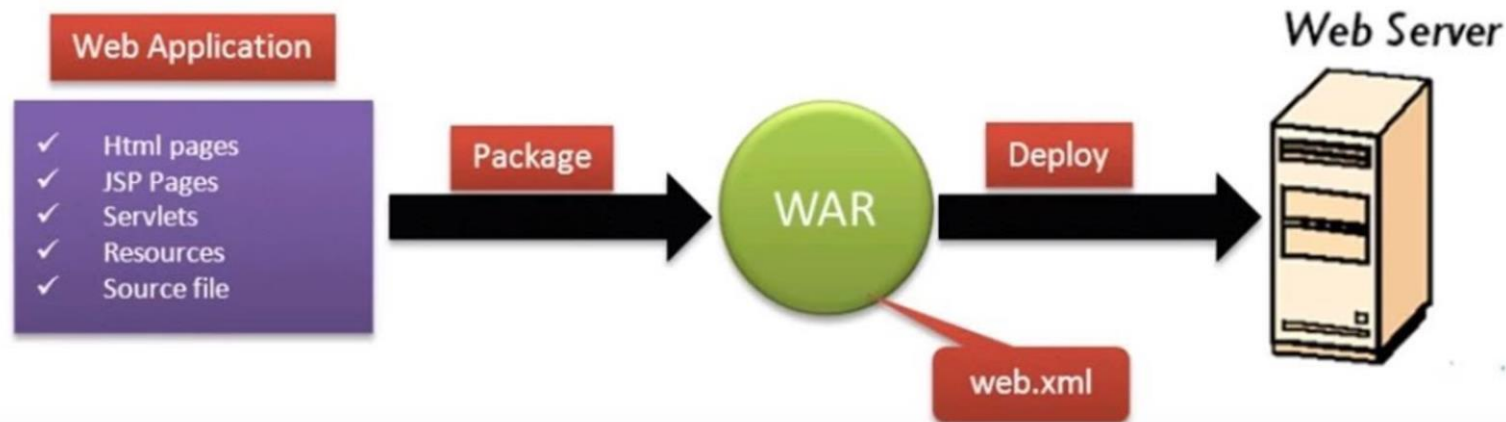
CI / CD pipeline with Jenkins



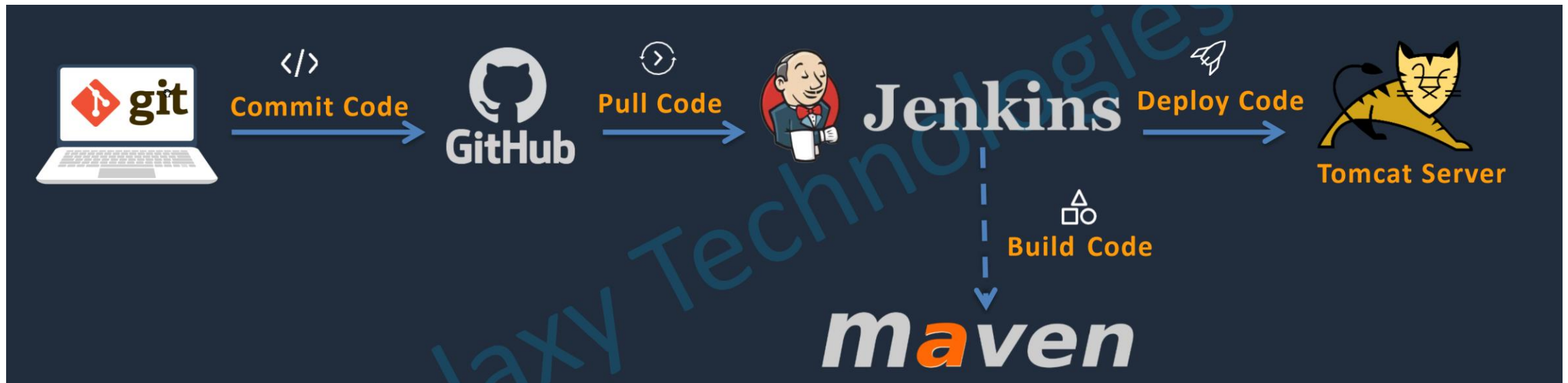
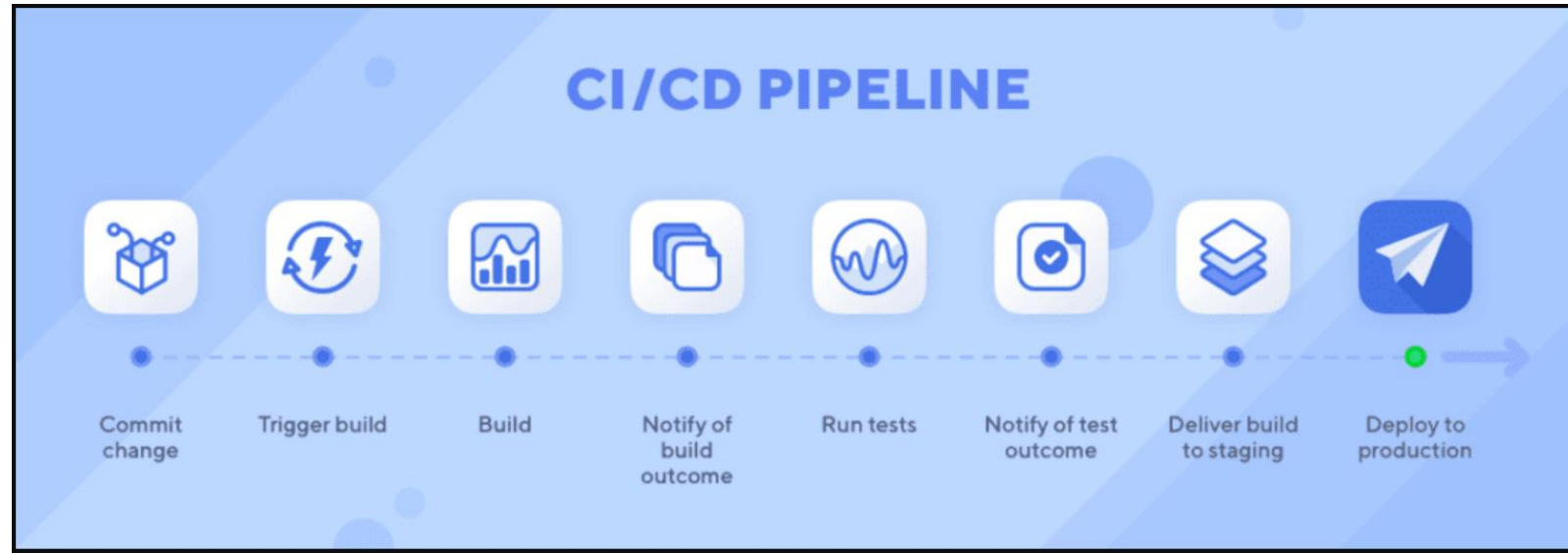


NOTE:
In real life, Jenkins, Tomcat and the user projects will all be on separate machines / servers

Generating JAR / WAR artifacts

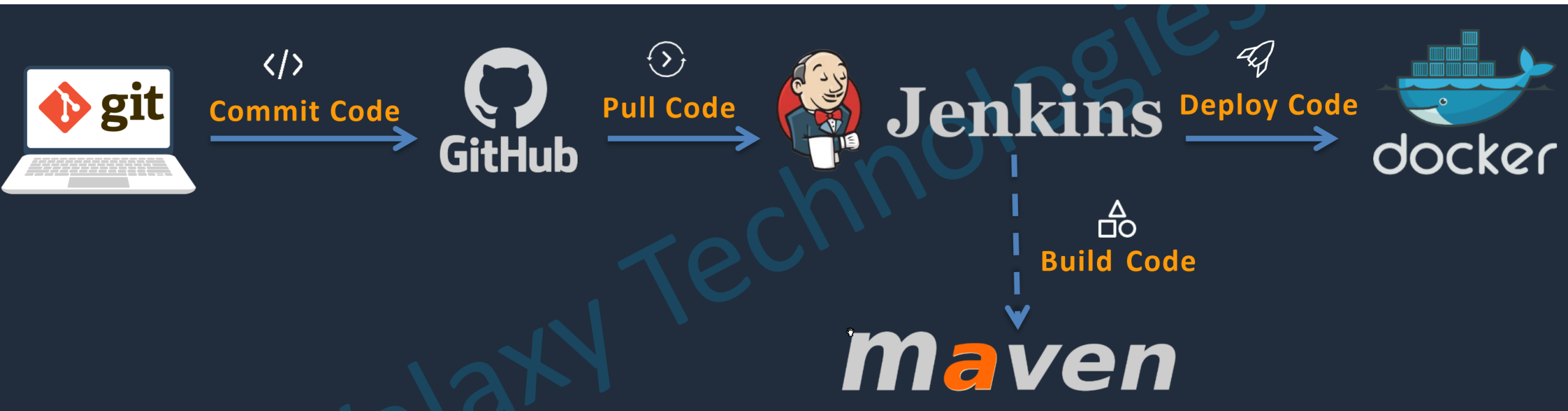


CI / CD pipeline with Jenkins



Using Jenkins with Docker

- ❖ Use case for Docker in CI / CD pipeline
- ❖ Configuring Docker in a freestyle project
- ❖ Configuring Docker using Pipeline DSL syntax



Lab practical configuration

❖ AWS EC2 instance (t2.large)

- Ubuntu 22.04.1 LTS
- OpenJDK 11.0.17
- Maven 3.9.0
- Docker Community 23.0.1
- Jenkins 2.375.3
- Apache Tomcat/9.0.71

❖ Reason for AWS EC2 usage

- Difficulty to install / configure all this software on your local machine (at least 2 – 3 hours)
- Requires admin privilege for proper configuration