End to End Integration

Maven-Git-Jenkins

https://www.linkedin.com/in/balaji-dinakaran/

Build tool - Maven

- Build is the process of creating executables from source code.
- Building application libraries is normally done using build tools
- Build tools can be used for application,
  - Compiling
  - Testing
  - Packaging
  - Deploying

Understanding the problem without Maven

There are many problems that we face during the project development. They are discussed below:

1) Adding set of Jars in each project: In case of struts, spring, hibernate frameworks, Page factory, TestNG we need to add set of jar files in each project. It must include all the dependencies of jars also.

2) Creating the right project structure: We must create the right project structure, otherwise it will not be executed.

3) Building and Deploying the project: We must have to build and deploy the project so that it may work.

Responsibility of Maven

- It makes a project easy to build
- It provides uniform build process (maven project can be shared by all the maven projects)
- It provides project information (log document, cross referenced sources, mailing list, dependency list, unit test reports etc.)
- It is easy to migrate for new features of Maven

Apache Maven helps to manage

- Builds
- Documentation
- Reporing
- SCMs
- Releases
- Distribution

https://www.linkedin.com/in/balaji-dinakaran/

Maven Setup

- JDK and JAVA_HOME
- Add JDK bin folder to path
- Download Apache Maven

  maven download - https://maven.apache.org/download.cgi
- Add MAVEN_HOME
- Add maven bin folder to path

Maven pom.xml file

- **POM** is an acronym for **Project Object Model**. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc.
- Maven reads the pom.xml file, then executes the goal.

Maven Phases

- Although hardly a comprehensive list, these are the most common default lifecycle phases executed.
- validate: validate the project is correct and all necessary information is available
- **compile**: compile the source code of the project
- **test**: test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- **package**: take the compiled code and package it in its distributable format, such as a JAR.
- integration-test: process and deploy the package if necessary into an environment where integration tests can be run
- verify: run any checks to verify the package is valid and meets quality criteria
- **install**: install the package into the local repository, for use as a dependency in other projects locally
- deploy: done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.
- **clean**: cleans up artifacts created by prior builds
- site: generates site documentation for this project

Version Control System

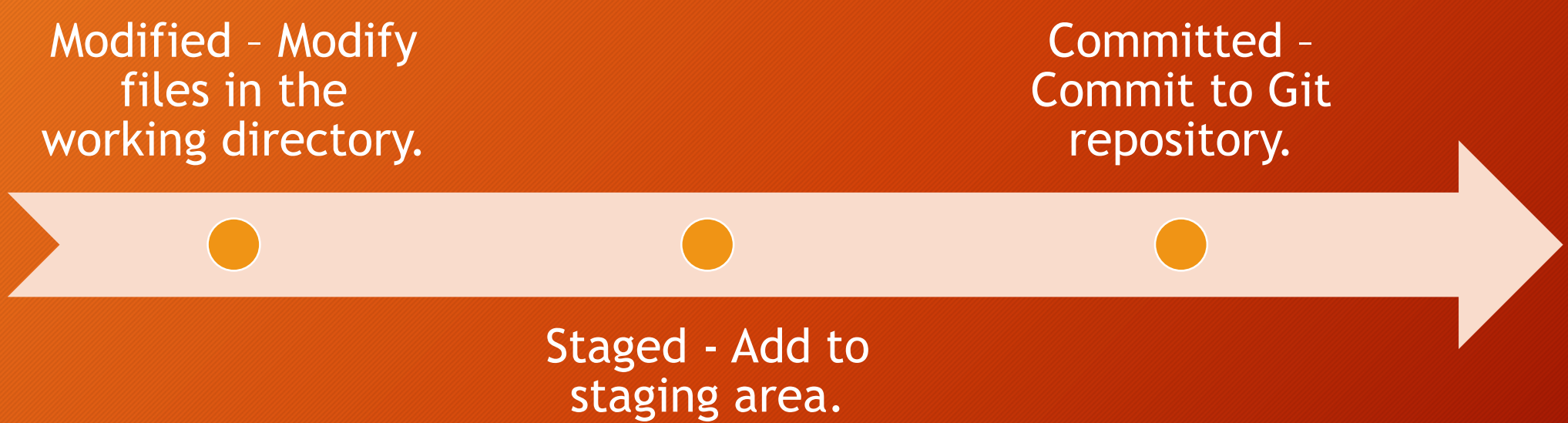
- Enables collaboration
  - Helps in managing different version of the saved codes
- Track Changes
  - Multiple developers are aware of any changes made to the code.
- Code recovery
  - Previous version of code can be recovered easily
- Detailed Project Enhancements
  - VCS asks you to describe the changes made for the good of other developer.

GIT

- Git Repository – A central location where the project code/data is stored and managed.
- Cloning of Git Repository – Creating an exact copy of an existing Git Repository on the Developer's machine.
- Commit – Saving the code to the local Git repository.
- Push to Upstream – Updating the server Git repository with the modification done on the local Git repository.
- Pull – Updating the local Git repository with the latest version of the data/code from the server Git repository.

https://www.linkedin.com/in/balaji-dinakaran/

GIT - Understanding



Code for Working directory are pulled from git repository and modification/fixes takes place in local and then staged finally pushed into git repository.

https://www.linkedin.com/in/balaji-dinakaran/

create a new repository on the command line

- echo "# check4May" >> README.md
- git init
- git add .
- git commit -m "first commit"
- git remote add origin https://github.com/balaji-githubstore/check4May.git
- git push -u origin master

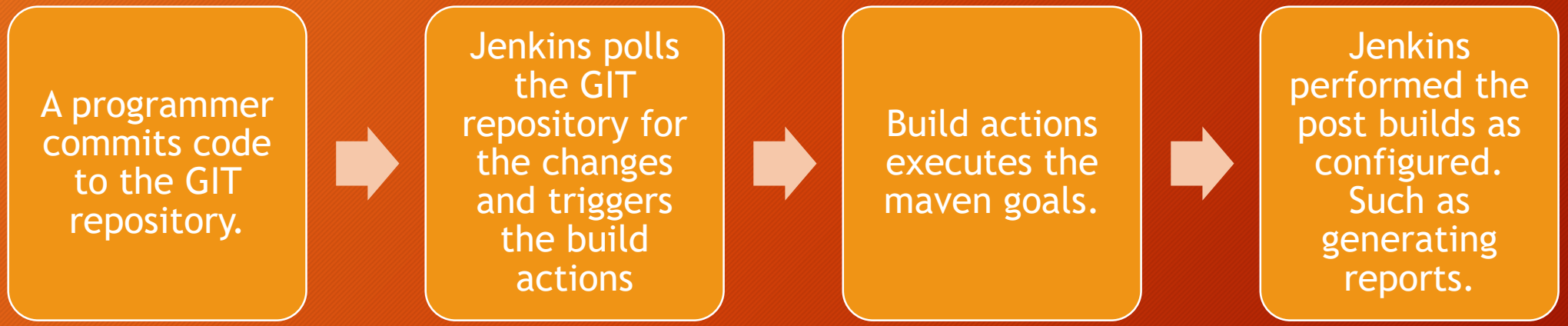
- If required,

  $ git config --global user.name "Emma Paris"

  $ git config --global user.email "eparis@atlassian.com"

https://www.linkedin.com/in/balaji-dinakaran/

Jenkins



https://www.linkedin.com/in/balaji-dinakaran/

Jenkins

Download and run Jenkins

- Download Jenkins (jenkins.war).
- Open up a terminal in the download directory.
- Goto command prompt and reach jenkins.war location and Run –
  - java -jar jenkins.war --httpPort=8080.
- Browse to http://localhost:8080.
- Please provide proper key after the loading the url.

What we worked on?

- Created projects on,
  - Maven
  - GIT Repository
  - Jenkins

  On every changes to SCM a build triggered by Jenkins which executes the maven goals

“

Start exploring Maven, Git and Jenkins!

”

Contact Details:
balajidinakaran1@gmail.com

https://www.linkedin.com/in/balaji-dinakaran/