DEVOPS

LAB MANUAL

1. **Any 10 commands to perform on git bash**
2. **Transfer repository from git bash to github**

**STEP 1:**

* Make a folder on desktop
* Open the folder with ‘gitBash’

**STEP 2:**

Run the following list of commands:

1. git --version
2. CONFIG
   * + 1. git config --global user.name “USERNAME”
       2. git config --global user.email “EMAIL”
       3. git config --global user.password “PASSWORD”
3. git init
4. git add FILENAME **OR** git add .
5. git commit -m “MESSAGE”
6. git branch
7. git checkout “BRANCHNAME”
8. git log
9. git status
10. git remote origin(origin could be any name for the 2nd time) REPOSITORY URL
11. git push -u origin master/main
12. git clone URL

**NOTE: Additional commands**

* mkdir
* touch
* ls

1. **Download and install jenkins**

**STEP 1:**

* Check for java version using ‘java -version’ in the cmd
* If the version is 8 or 11 it is compatible and proceed to download jenkins

**STEP 2**:

* If the version is not compatible then download the java version 8 or 11.
* To download Java 8 or 11:
* Search for oracle
* Go to oracle website
* Go to products (top left)
* Scroll and click on java
* Scroll to Java products and technologies
* Go to java archive
* Scroll and click java 11 (right side)
* Scroll and download installer x64
* Go to downloads folder and install java (click on installer and go next next next)
* Check the version

**STEP 3:**

* Searchfor jenkins website
* Go to downloads

**STEP 4**:

* Download the **Download Jenkins 2.414.3 LTS (Generic java pakage .war)**
* The file downloaded, cut paste it into a folder ‘Jenkins’ in the drive and go to cmd.
* Type the following cmd ‘java -jar jenkins.war --httpPort=9191’
* From the cmd prompt copy the password (a lamba alphanumeric password somewhere last in the cmd)
* Go to chrome and type the URL: <http://localhost:9191>
* Paste the password in the ‘Administrator Password’
* Select ‘Install suggested plugins’
* Let the plugins install
* Enter the details as follows:
* Username: admin
* Password: Abc@123
* Confirm Password: Abc@123
* Name: admin
* Email: [admin@gmail.com](mailto:admin@gmail.com)
* Click ‘save and continue’
* Go next…..
* Click ‘Start Jenkins’
* DONE DANA DONE

1. **Build gradle project at jenkins**

**STEP 1:**

* Search gradle downloads
* Go to the gradle realeases wala
* Scroll and click on ‘binary only’ under Oct 04, 2023 and v8.4
* The file downloaded, cut paste it into a folder ‘Gradle’ in the drive.
* Go into the folder and extract all files

**STEP 2**:

* Go into ‘gradle-8.4’
* Go into ‘bin’
* Copy the path
* Go to system variables
* Click environment variables
* Under user variables
* Click on path
* Add new
* Paste the path
* Under system variables
* Click on path
* Add new
* Paste the path
* Remove /bin here
* Open command prompt ‘run as adminstration’
* Type command gradle --version to check successful installation of gradle.

**STEP 3:**

* Go to jenkins.war
* Go to cmd
* Type ‘java -jar jenkins.war --httpPort=9191’
* Open a new tab and type the URL ‘localhost:9191’
* Enter username and password
* Username: admin
* Password: Abc@123

**STEP 4**:

* Go to manage jenkins (left side)
* Go to tools
* Scroll down to ‘gradle installations’
* Under name, put ‘Gradle’
* Uncheck ‘install automatically’
* Under GRADLE\_HOME, put ‘path just before the bin’
* Apply and then save

**STEP 5**:

* Go to manage jenkins (left side)
* Go to plugins
* Under **‘**Installed Plugins’,
* Check if gradle is installed, if not install it.

**STEP 6:**

* Go to new tab
* Search for simple gradle projects github
* Click on any project
* Click on the green button ‘code’
* Copy the link

**STEP 7:**

* Go to Dashboard
* Add new Item
* Add name to the project
* Select freestyle project
* Add a description ‘This is gradle project by Your Name.’
* Check ‘GitHub project’
* Paste the copied link in the previous step here.
* Under Source Code Management
* Select Git
* Paste the repository URL
* Under Build Triggers
* Check poll SCM
* In Schedule, type ‘\* \* \* \* \*’
* Check Ignore post-commit hooks .
* Under Build steps,
* Select invoke gradle script.
* Select the gradle installation name give in **step .**
* Apply and the save

**STEP 8:**

* Click build now (left hand)
* Niche, your project starting build
* After it is built select it
* Go to console output
* Scroll until the last
* If it shows success then project is successful.
* DONE DANA DONE

1. **Build ant project at jenkins**

**STEP 1:**

* Search ant downloads
* Go to the binary distributions wala
* Scroll and download the zip file 1.10.14 release (supports java 8)
* The file downloaded, cut paste it into a folder ‘Ant’ in the drive.
* Go into the folder and extract all files

**STEP 2**:

* Go into folders…
* Go into ‘bin’
* Copy the path
* Go to system variables
* Click environment variables
* Under user variables
* Click on path
* Add new
* Paste the path
* Under system variables
* Click on path
* Add new
* Paste the path
* Remove /bin here
* Open command prompt ‘run as adminstration’
* Type command ant --version to check successful installation of ant.

**STEP 3:**

* Go to jenkins.war
* Go to cmd
* Type ‘java -jar jenkins.war --httpPort=9191’
* Open a new tab and type the URL ‘localhost:9191’
* Enter username and password
* Username: admin
* Password: Abc@123

**STEP 4**:

* Go to manage jenkins (left side)
* Go to tools
* Scroll down to ‘ant installations’
* Under name, put ‘Ant’
* Uncheck ‘install automatically’
* Under ANT\_HOME, put ‘path just before the bin’
* Apply and then save

**STEP 5**:

* Go to manage jenkins (left side)
* Go to plugins
* Under **‘**Installed Plugins’,
* Check if ant is installed, if not install it.

**STEP 6:**

* Go to new tab
* Search for simple ant projects github
* Click on any project
* Click on the green button ‘code’
* Copy the link

**STEP 7:**

* Go to Dashboard
* Add new Item
* Add name to the project
* Select freestyle project
* Add a description ‘This is ant project by Your Name.’
* Check ‘GitHub project’
* Paste the copied link in the previous step here.
* Under Source Code Management
* Select Git
* Paste the repository URL
* Under Build Triggers
* Check poll SCM
* In Schedule, type ‘\* \* \* \* \*’
* Check Ignore post-commit hooks .
* Under Build steps,
* Select invoke ant.
* Select the ant installation name give in **step .**
* Apply and the save

**STEP 8:**

* Click build now (left hand)
* Niche, your project starting build
* After it is built select it
* Go to console output
* Scroll until the last
* If it shows success then project is successful.
* DONE DANA DONE

1. **Build maven project at jenkins and deploye it to tomcat**

Pre-requisites:

* Pc
* Maven
* Tomcat [chg port to 7080, set username and pwd]
* Jenkins
* jdk
* Tools
* Jdk (manage jenkins)
* Maven
* Plugins
* Deploy to war container
* Maven integration

1. **Build and test maven project through selenium**

Pre-requisites:

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

-------------------------------------------------------------------------------

mvn archetype:generate -DgroupId=com.example -DartifactId=JavProject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

----------------------------------------------------------------------------

package com.example;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class SeleniumExample{

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Users\\USER\\Downloads\\chromedriver\_win32");

WebDriver driver = new ChromeDriver();

driver.get("https://www.example.com");

driver.quit();

}

}

--------------------------------------------------------------------

<https://chromedriver.storage.googleapis.com/114.0.5735.90/chromedriver_win32.zip>

* Pc
* Chrome Driver
* Plugins
* Selenium plugin
* Selenium builder plugin
* TestNG Results Plugin

**STEP 1:**

* Download Chrome Driver (link provided)
* Extract it

**STEP 2**:

* Go to your Local Drive Folder
* Create folder ‘Selinium’
* Open cmd here
* Paste the mvn cmd
* A folder ‘JavProject’ will be created

STEP 3:

* Open it in VS Code
* Go to the pom.xml file
* Paste the dependency code in the dependencies tag after the prsent dependency tag.
* Go to src/main/java/com/Example
* Create a Java file
* Name it SeleniumExample.java
* Paste the code provided
* Also go to the chrome driver extracted
* Open it
* Copy the path
* Paste the path one the line 8 (tentative) after value
* Open new terminal (top)
* Type cmd ‘mvn install’
* If it doen not work
* Type cmd cd JavProject

**STEP 4**:

* Login to your git account
* Go to repositories
* Create new
* Give it a name
* Create it
* Copy the link of the repository

**STEP 5**:

* Open JavProject with git bash
* Right click on folder
* Show more options
* Open with git bash here
* Type the following git commands to push folder to repository:
* git init
* git add .
* git commit -m “SOME MESSAGE”
* git remote add origin paste the repository link copied
* git push -u origin main/master (based on your PC)

**STEP 6:**

* Go to Jenkins
* Go to jenkins.war
* Go to cmd
* Type ‘java -jar jenkins.war --httpPort=9191’
* Open a new tab and type the URL ‘localhost:9191’
* Enter username and password **(or whatever you entered)**
* Username: admin
* Password: Abc@123
* Install the mentioned plugins in pre-requisites.

**STEP 7:**

* Create new item
* Give any name
* Select maven project
* give any description (opitional)
* Check the ‘GitHub Project’
* Paste the repository link copied
* Under build, under goals and options, type test
* Apply and then save
* Build it
* Click on the build (eg. #2)
* Go to console output
* Scroll, if SUCCESS then project done
* DONE DANA DONE

1. **Create a container and pull images from docker**

**STEP 1:**

* Search download docker for windows
* Go to link docs.docker….
* Click downld desktop for windows

**STEP 2:**

* Run the following cmd
* docker --version
* docker run hello-world
* docker pull busybox
* docker images
* docker run busybox
* docker run busybox echo "hello from busybox"
* docker ps
* docker ps -a