DevOps

Lab Book

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision No. | Author | **Summary of Changes** |
| April 2017 |  | Rahul Vikash | Created new lab book as per revised course contents |
|  |  |  |  |
|  |  |  |  |

Table of Contents

*Getting Started..……..…………………………………………………………………………… 4*

[Overview 3](#_Toc512888436)

[Setup Checklist for DevOps 3](#_Toc512888437)

[Instructions 4](#_Toc512888438)

[Learning More 4](#_Toc512888439)

[Lab 1. Git with DevOps 5](#_Toc512888440)

[Lab 2 Git with DevOps 6](#_Toc512888442)

[Lab 3 Jenkin 7](#_Toc512888443)

[3.1. Create a Job in Jenkin, pull the calculator application (same application push in assignment 2) from GitHub repository ,build with maven . 7](#_Toc512888444)

[3.2. Create a Job in Jenkin, pull the calculator application (same application push in assignment )2.1) from GitHub repository, build with maven . 7](#_Toc512888445)

[Lab 4 Mockito Lab 8](#_Toc512888446)

Getting Started

## Overview

This lab book is a guided tour for learning DevOps. It comprises ‘To Do’ assignments. Follow the steps provided to work out the ‘To Do’ assignments given.

## Setup Checklist for DevOps

Here’s what is expected on your machine for the lab in order to work.

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows XP, Windows 7 or Windows 8
* Memory: 2GB of RAM (1GB or more recommended)
* Google Chrome 36.0 or Mozilla Firefox 31.0 or Internet Explorer 10 or above

Please ensure that the following is done:

* Java is installed, maven, Jenkin, sonar configuration done .Git bash &Ui installed .IBM bluemix account should be created

## Instructions

* Create a directory by your name in drive <drive>. In this directory, create a subdirectory DevOps\_assgniment. For each lab exercise create a directory as lab <lab number>.

You may also look up the on-line help provided in

## Learning More

* <https://www.cloudbees.com/jenkins/about>
* <https://www.sonarqube.org/>
* <https://git-scm.com/>
* <https://maven.apache.org/>
* <https://www.ibm.com/cloud-computing/bluemix/what-is-bluemix?lnk=hm>
* <https://en.wikipedia.org/wiki/DevOps>

1. Git with DevOps

|  |  |
| --- | --- |
| Goals | * Working with Git-Local & remotely |
| Time | 60 minutes |

## 

## 

**B** Local Repository (for admin use only)

**A** Public Repository (Remote)

**E** Public Repository (Backup)

**D** Local Repository (Tester)

**C** Local Repository

(Developer)

Significance of the repositories:

A: Public repository used for data storage, all clients pushes and pull here

B: Initial directory structure and branches are created here to be pushed

C and D: These are local repositories which will have working tree

E: Public repository meant for backup purpose.

Perform following operations:

**Note: Participants are required to submit commands used for each question in a word document**

1. Create all the repositories.
2. Operations in B: Create a file info.txt containing text “Project”. Commit it and then create 2 branches in master. Branches are to be named as Development and Testing .Push all branches to A.
3. Pull Development branch on C and Testing branch on D
4. Now, on C add a file MyJavaCode.txt, stage it and commit it. On D add file MyJUnitTestCase.txt, stage and commit it. Goto D and pull all files from C.
5. Goto D and edit file MyJavaCode.txt (Assume that it is some file which is accidently edited). Stage it, commit it. Now, push data from D to A. Then goto C and pull from A. It needs resolving conflict. While resolving conflict, use text from C and discard all changes in MyJavaCode.txt made in D. Commit C and push from C to A.
6. Pull from A to C. Push from C to E. Assume that A is down. Create file Source2.java in C. Stage it, commit it and push to E. Create file HttpdTest.txt in D. Stage it and commit it. Pull from E and then Push to E. Now, assume A is up. Now D is in sync with E. So, pull A to D and then push from D to A.

Lab 2 Git with DevOps

Using rebase change order of commit in any of the above repository.

2. Create the account in GitHub-, push the calculator Application in remote repository, next user pull it to local repository & make the change & again push changed application to remote repository.

2.1 Extend the above application create a dynamic web calculator application.Push the data into github repository.

**Note: Participants are required to submit commands used for each question in a word document**

Lab 3 Jenkin

|  |  |
| --- | --- |
| Goals | * Working with Jenkin |
| Time | 30 minutes |

## 3.1. Create a Job in Jenkin, pull the calculator application (same application push in assignment 2) from GitHub repository ,build with maven .

## 3.2. Create a Job in Jenkin, pull the calculator application (same application push in assignment )2.1) from GitHub repository, build with maven .

**Note: Participants are required to submit commands used for each question in a word document**

Lab 4 Mockito Lab

|  |  |
| --- | --- |
| Goals | * Working with Mockito |
| Time | 30 minutes |

* 4.1 : Suppose we want to test the method addProduct in the ProductrService class, and within this addProduct method, the save method of the ProductDao class is invoked. We don’t want to call the real implementation of the ProductDao save() method for a few reasons:
  + We only want to test the logic inside the addProduct () in isolation.
  + We may not yet have implemented it.
  + We don’t want the unit test of the addProduct () fail if there is a defect in the save() method in the ProductDao.

So mock the behavior of the dependencies using Mockito API.