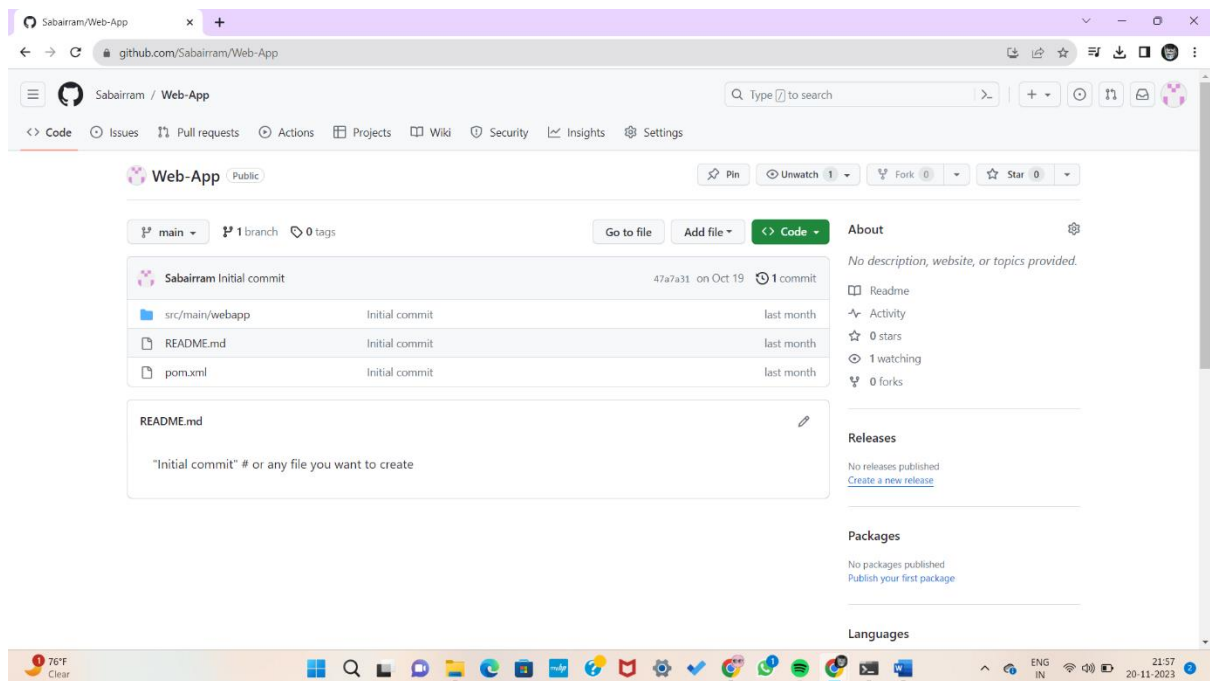


Name: Sabairram Attar  
Roll No:02  
Sub:DevOps

## Assignment No. 10

Create a Jenkins pipeline to build and push Docker Image. Dockerfile, Jenkinsfile and the application(for which image will be built) shall be stored in a GitHub Repository.

->add Project to GitHub and commit



Open Jenkins Dashboard

Create new project as follows

devops-automation Config (Jen) x Pipeline Syntax Snippet Genera x Pipeline Syntax Snippet Genera x +

localhost:8080/job/devops-automation/configure

# Jenkins

Search (CTRL+K) Sabairam Attar log out

Dashboard > devops-automation > Configuration

## Configure

General Advanced Project Options Pipeline

### General

Enabled

Description

This is simple jenkins application for docker image

Plain text [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

https://github.com/Sabairam/Web-App.git/

Save Apply

---

76°F Clear

devops-automation Config (Jen) x Pipeline Syntax Snippet Genera x +

localhost:8080/job/devops-automation/configure

Dashboard > devops-automation > Configuration

## Configure

General Advanced Project Options Pipeline

### Definition

Pipeline script

Script ?

```
1= pipeline{
2  agent any
3  tools{
4    maven "maven_3.5.0"
5  }
6  stages{
7    stage('Build Maven'){
8      steps{
9
10     }
11  }
12 }
13 }
```

☒ Use Groovy Sandbox ?

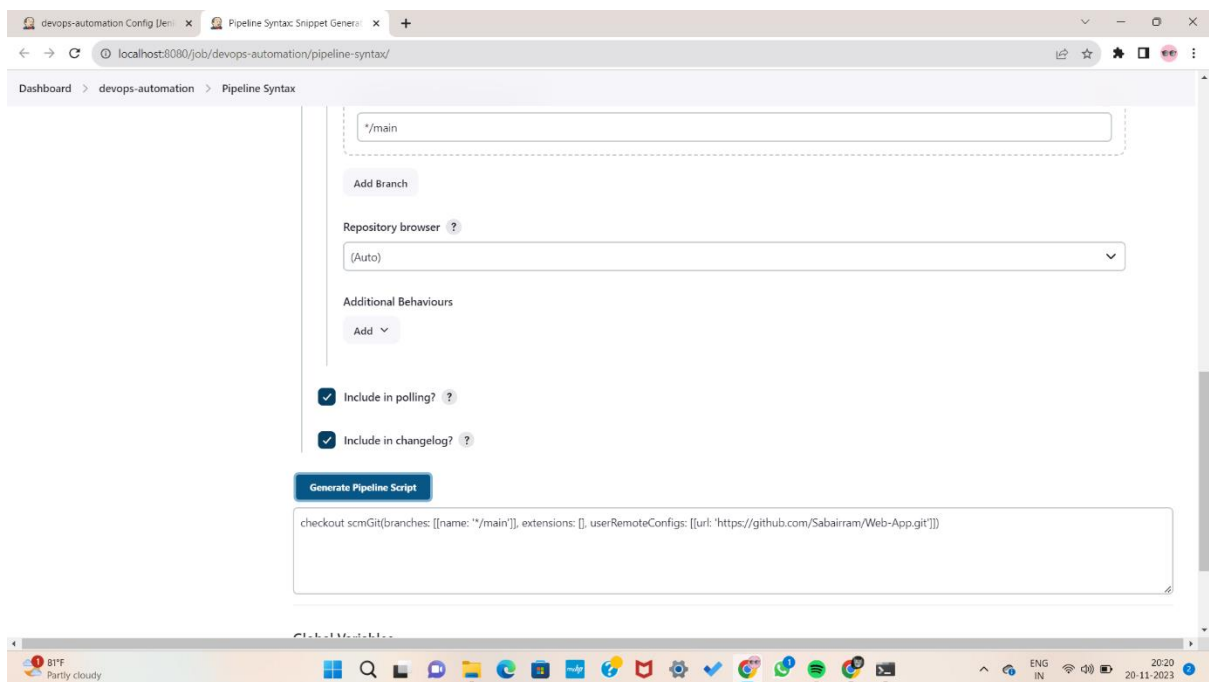
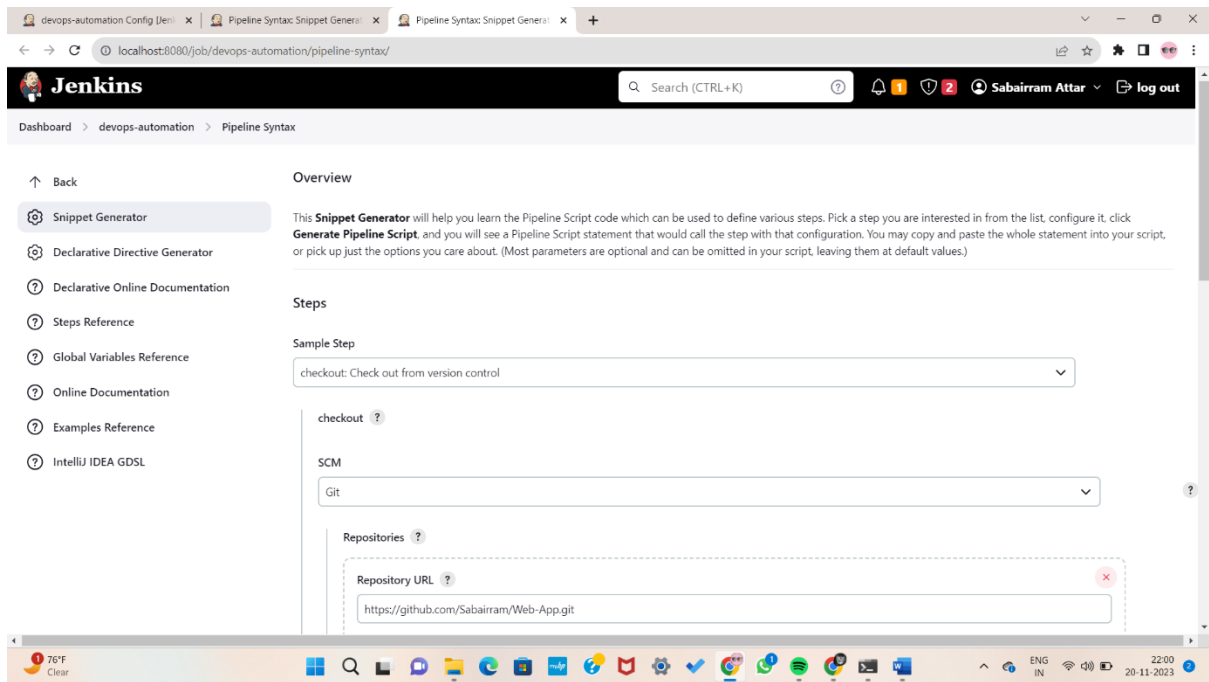
[Pipeline Syntax](#)

Save Apply

REST API Jenkins 2.428

81°F Partly cloudy

Generate pipeline syntax



## Update the script pipeline as

```
pipeline{
    agent any
    tools{
        maven 'maven'
    } Stages{
stage('Build Maven'){
    steps{
```

devops-automation Config (Jeri) x

Pipeline Syntax: Snippet Genera x

Pipeline Syntax: Snippet Genera x

+

localhost:8080/job/devops-automation/configure

Dashboard > devops-automation > Configuration

Configure

General

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script

Script ?

```
4      maven 'maven'
5    }
6
7    stages {
8
9
10     stage('Build Maven'){
11       steps{
12         script{
13           checkout scmGit(branches: [[name: '**/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Sabairam/Web
14           bat 'mvn clean install'
15         }
16       }
17     }
18   }
19 }
```

☒ Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

The screenshot displays the Jenkins web interface in a browser window. The address bar shows the URL `localhost:8080/job/devops-automation/`. The Jenkins logo and name are visible in the top left. A search bar and user profile (Sabairram Attar) are in the top right. The main content area shows the 'devops-automation' pipeline status as 'Online' with a green checkmark. Below this, a list of actions (Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, GitHub, Rename, Pipeline Syntax, GitHub Hook Log) is on the left. The 'Stage View' section shows a table of stages: 'Declarative: Tool Install' (202ms) and 'Build Maven' (6min 43s). A 'Permalinks' section lists recent builds. The bottom of the image shows a Windows taskbar with various application icons and system tray information.

devops-automation [jenkins] x Pipeline Syntax: Snippet Genera x +

localhost:8080/job/devops-automation/

Jenkins

Search (CTRL+K)

Sabairram Attar log out

Dashboard > devops-automation >

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

GitHub

Rename

Pipeline Syntax

GitHub Hook Log

devops-automation

This is simple jenkins application for docker image

Edit description

Disable Project

Stage View

Average stage times:  
(Average full run time: ~6min 44s)

	Declarative: Tool Install	Build Maven
	202ms	6min 43s

#10 Nov 20, 21:45 No Changes

Permalinks

- Last build (#10), 18 min ago
- Last stable build (#10), 18 min ago
- Last successful build (#10), 18 min ago
- Last failed build (#7), 27 min ago

Build History trend

Filter builds...

#10 Nov 20, 2023, 9:45 PM

76°F Clear

ENG IN 20-11-2023 22:03

# Create Dockerfile

# Use an official Maven runtime as a parent image

FROM maven:3.6.3-openjdk-11-slim

# Set the working directory in the container

WORKDIR /usr/src/app

# Copy the Maven project files into the container

COPY pom.xml .

COPY src ./src

# Build the application

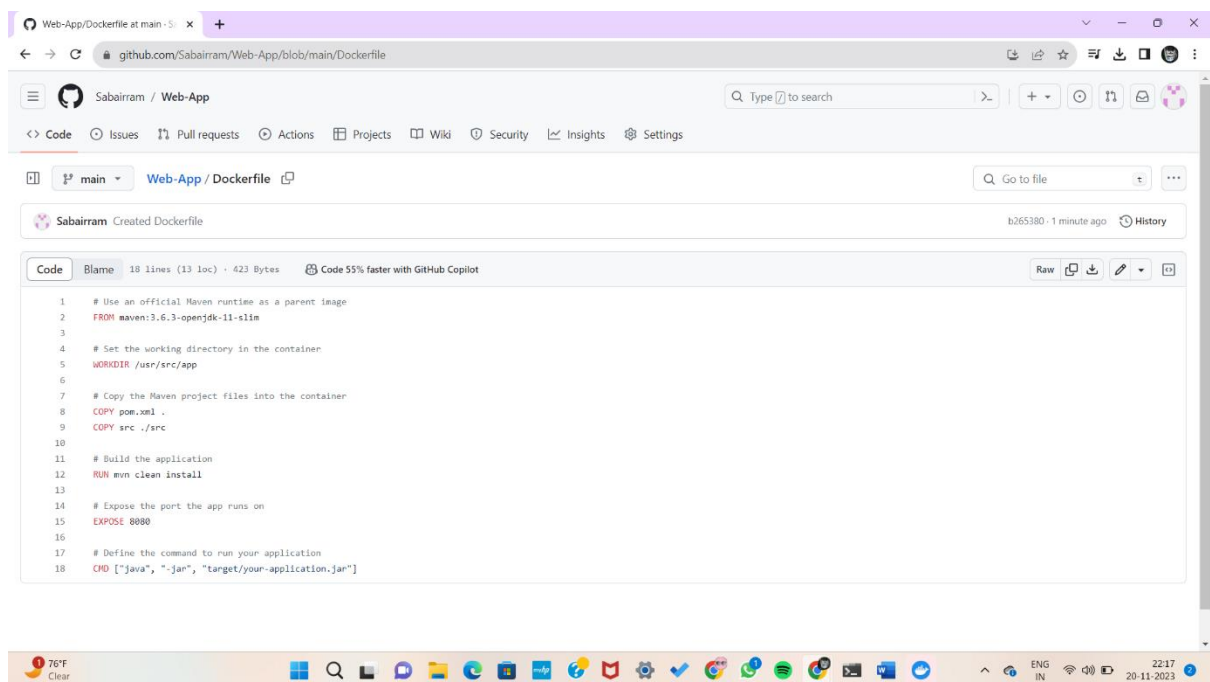
RUN mvn clean install

# Expose the port the app runs on

EXPOSE 8080

# Define the command to run your application

CMD ["java", "-jar", "target/your-application.jar"]



Now go to Jenkins configure setup

Create another stage on pipeline script for building docker image

```
pipeline{
  agent any

  tools{
    maven 'maven'
  } stages {

    stage('Build Maven'){

      steps{

        script{

          checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url:
'https://github.com/Sabairram/Web-App.git']])

          bat 'mvn clean install'

        }  }

    stage('Build docker image'){

      steps{

        script{

          bat 'docker build -t sabairram/webapp .'

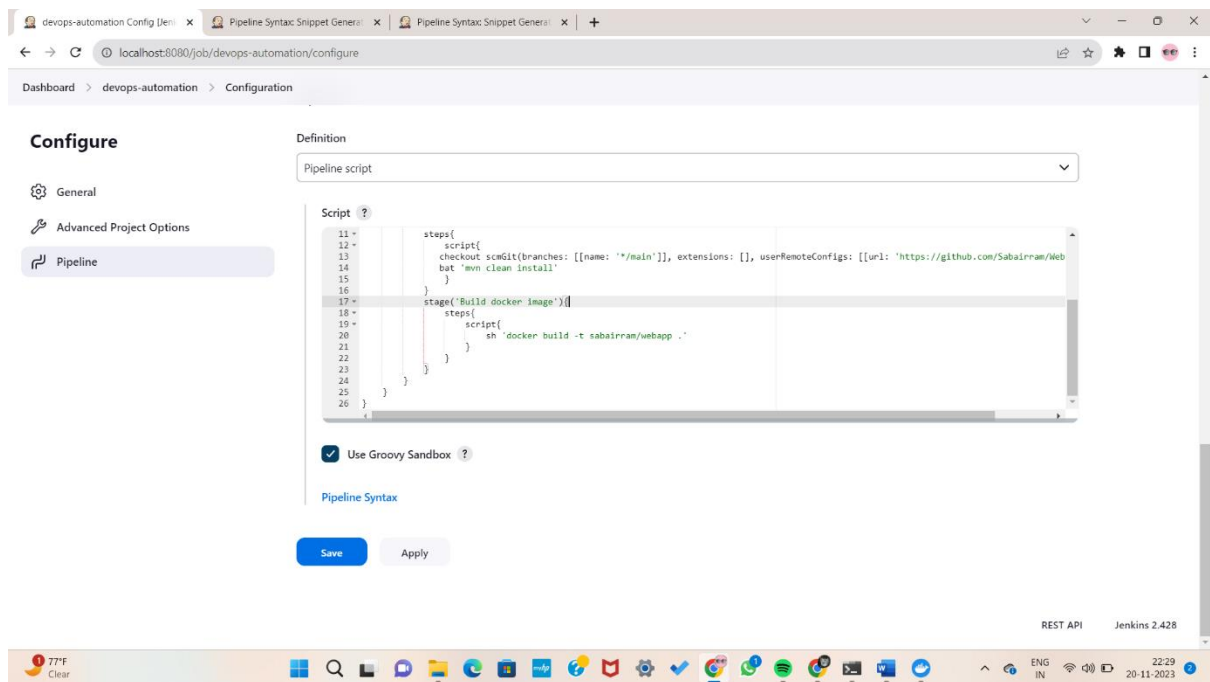
        }

      }

    }

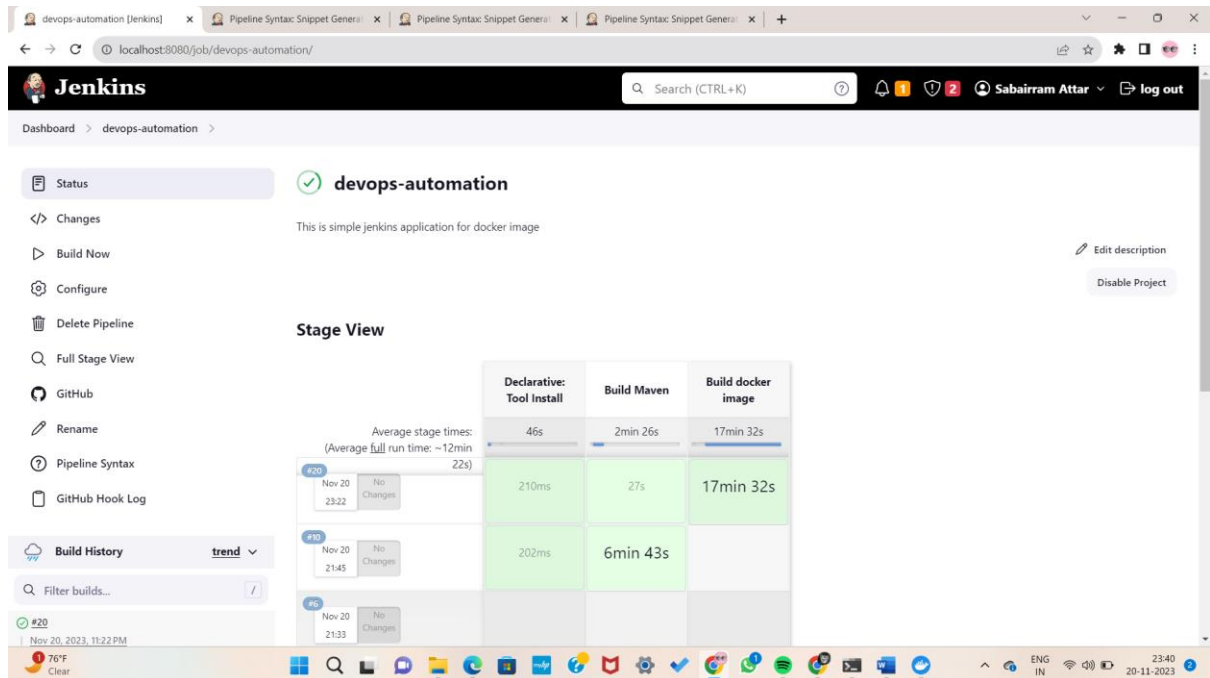
  }

}
```



Click on Apply and Save

Now Build the setup



Here, Build the docker image successfully

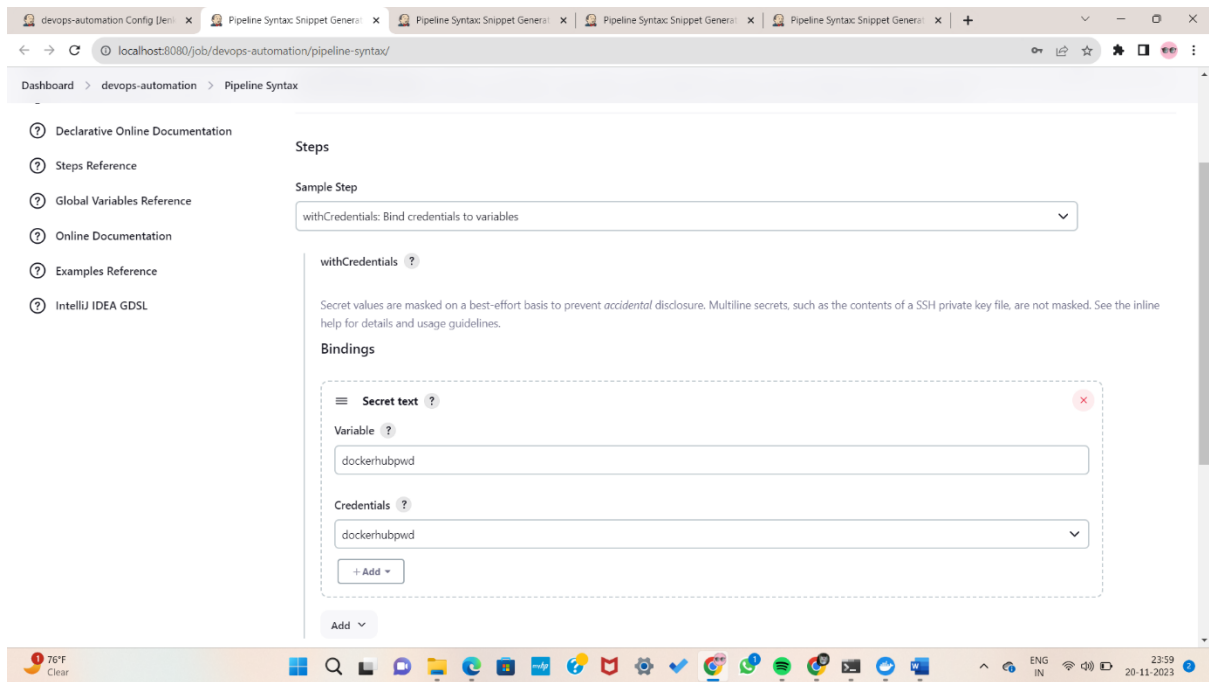
Next Step push it to docker hub

Now, We have to move to Configure step

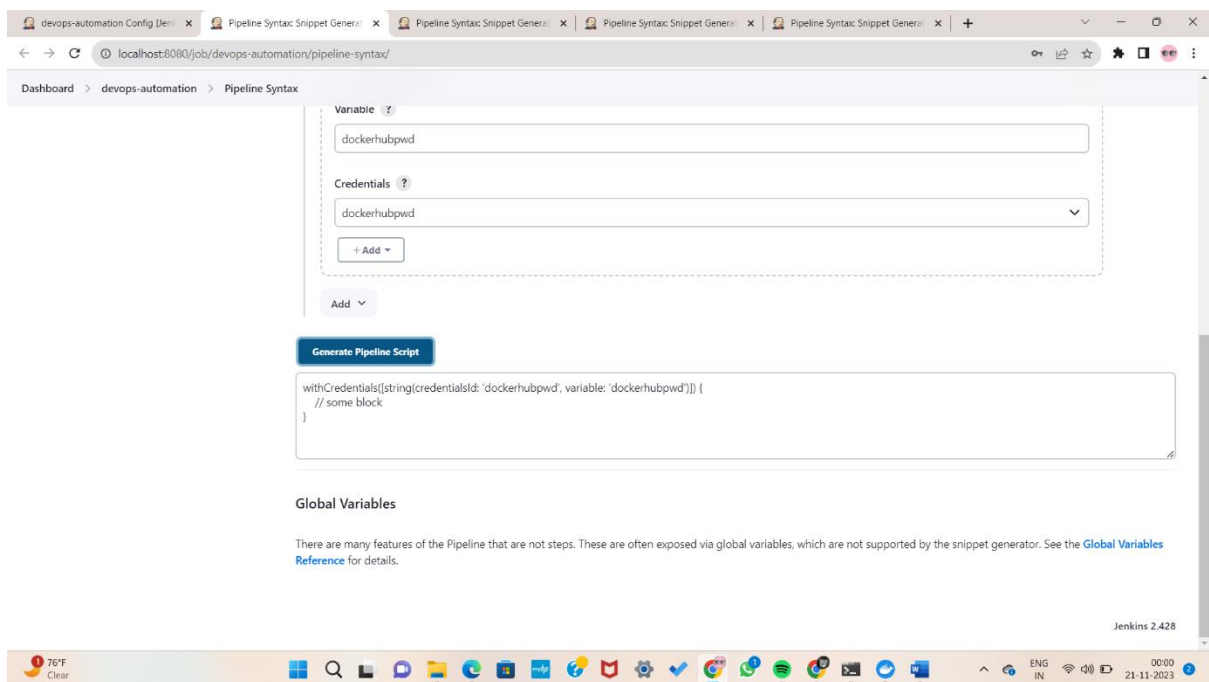
In Pipeline script add the required changes and add another stages

Go to pipeline syntax for docker login

Create Secret text for dockerhub



## Now Generate pipeline



Now update the pipeline script as follows

```
pipeline{
    agent any
    tools{
        maven 'maven'
```



}

stages {

```
stage('Build Maven'){
```

```
steps{
```

```
checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Java-Techie-jt/devops-automation.git']])
```

```
bat 'mvn clean install'
```

}

}

```
stage('Build docker image'){
```

```
steps{
```

```
script{
```

```
bat 'docker build -t javatechie/devops-integration .'
```

}

}

}

```
stage('Push image to hub'){
```

```
steps{
```

```
script{
```

```
withCredentials([string(credentialsId: 'dockerhubpwd', variable: 'dockerhubpwd')]) {
```

```
bat 'docker login -u sabairram -p ${dockerhubpwd}'
```

```
bat 'docker push javatechie/devops-integration .'
```

}

}

}

}

$$\}} \}$$

Now click on save and build

You can see on to the Dockerhub account the image is created.