



TU856/3

INTRODUCTION TO DEVOPS

LAB 4 – CONTINUOUS INTEGRATION WITH JENKINS



26/02/2025

PAULINA CZARNOTA C21365726

1. SETTING UP AND STARTING JENKINS SERVICE ON VM

This section includes starting the virtual machine, accessing it via SSH, and running the command: `systemctl status jenkins`

This verifies that the Jenkins service is active and running.

```
vagrant@bookworm: ~
Microsoft Windows [Version 10.0.19045.5487]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\35389>cd C:\Users\35389\Desktop\TU856 Modules\YEAR 3\Year 3 - Semester 2\Introduction to DevOps - Eoin Rogers\Labs\Week 5

C:\Users\35389\Desktop\TU856 Modules\YEAR 3\Year 3 - Semester 2\Introduction to DevOps - Eoin Rogers\Labs\Week 5>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'debian/bookworm64' version '12.20250126.1' is up to date...
==> default: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> default: flag to force provisioning. Provisioners marked to run always will still run.

==> default: Machine 'default' has a post 'vagrant up' message. This is a message
==> default: from the creator of the Vagrantfile, and not from Vagrant itself:
==> default:
==> default: Vanilla Debian box. See https://app.vagrantup.com/debian for help and bug reports

C:\Users\35389\Desktop\TU856 Modules\YEAR 3\Year 3 - Semester 2\Introduction to DevOps - Eoin Rogers\Labs\Week 5>vagrant ssh
Linux bookworm 6.1.0-31-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 27 03:09:58 2025 from 10.0.2.2
vagrant@bookworm:~$ systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-02-27 01:26:05 UTC; 1h 44min ago
     Main PID: 335 (java)
       Tasks: 49 (limit: 2307)
        Memory: 1.5G
          CPU: 11min 31.122s
        CGroup: /system.slice/jenkins.service
               └─335 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins>

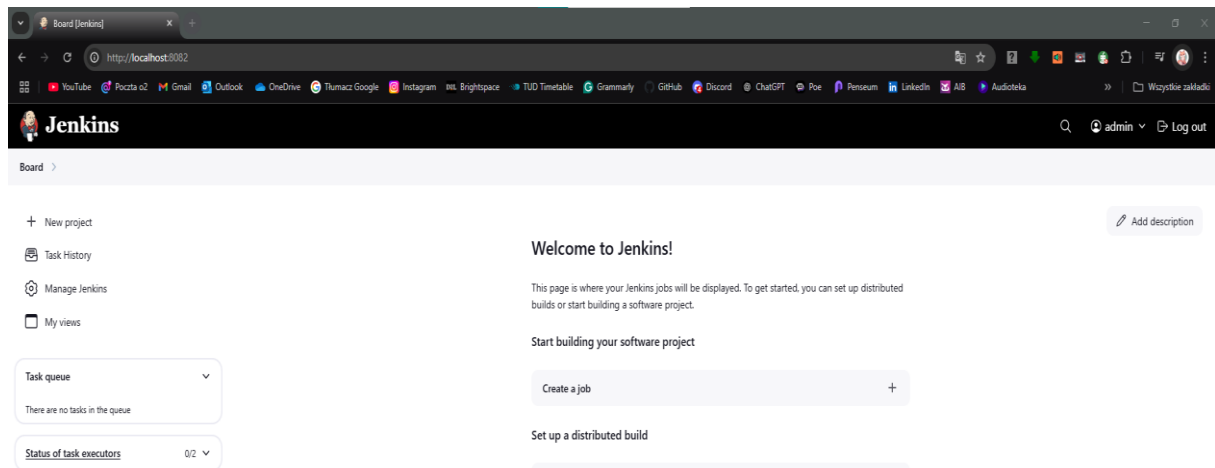
Warning: some journal files were not opened due to insufficient permissions.
lines 1-11/11 (END)...skipping...
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-02-27 01:26:05 UTC; 1h 44min ago
     Main PID: 335 (java)
       Tasks: 49 (limit: 2307)
        Memory: 1.5G
          CPU: 11min 31.122s
        CGroup: /system.slice/jenkins.service
               └─335 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

2. JENKINS DASHBOARD

This screenshot demonstrates the retrieval of the admin password required to access Jenkins.

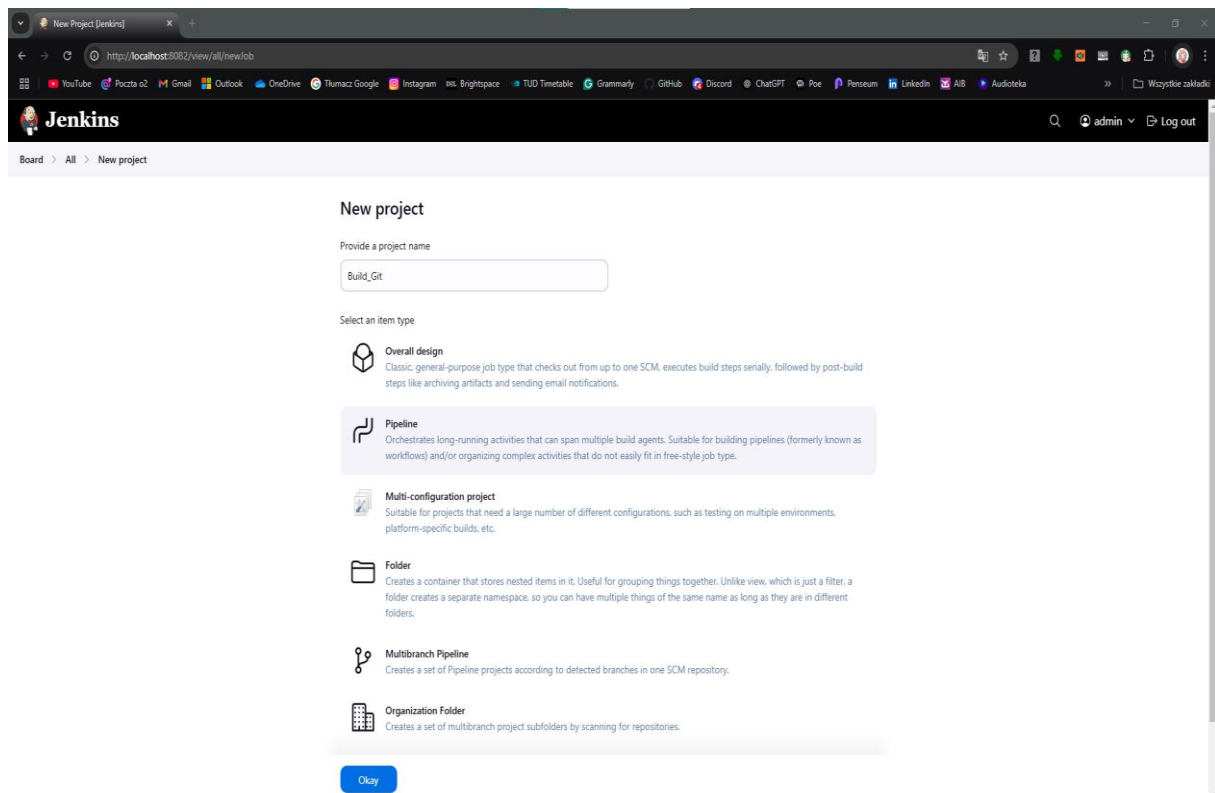
```
vagrant@bookworm:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
6e5e31764a7e488083b08135cbcd397c
```

The image confirms that Jenkins is installed and running correctly, accessed via:
`http://localhost:8082`



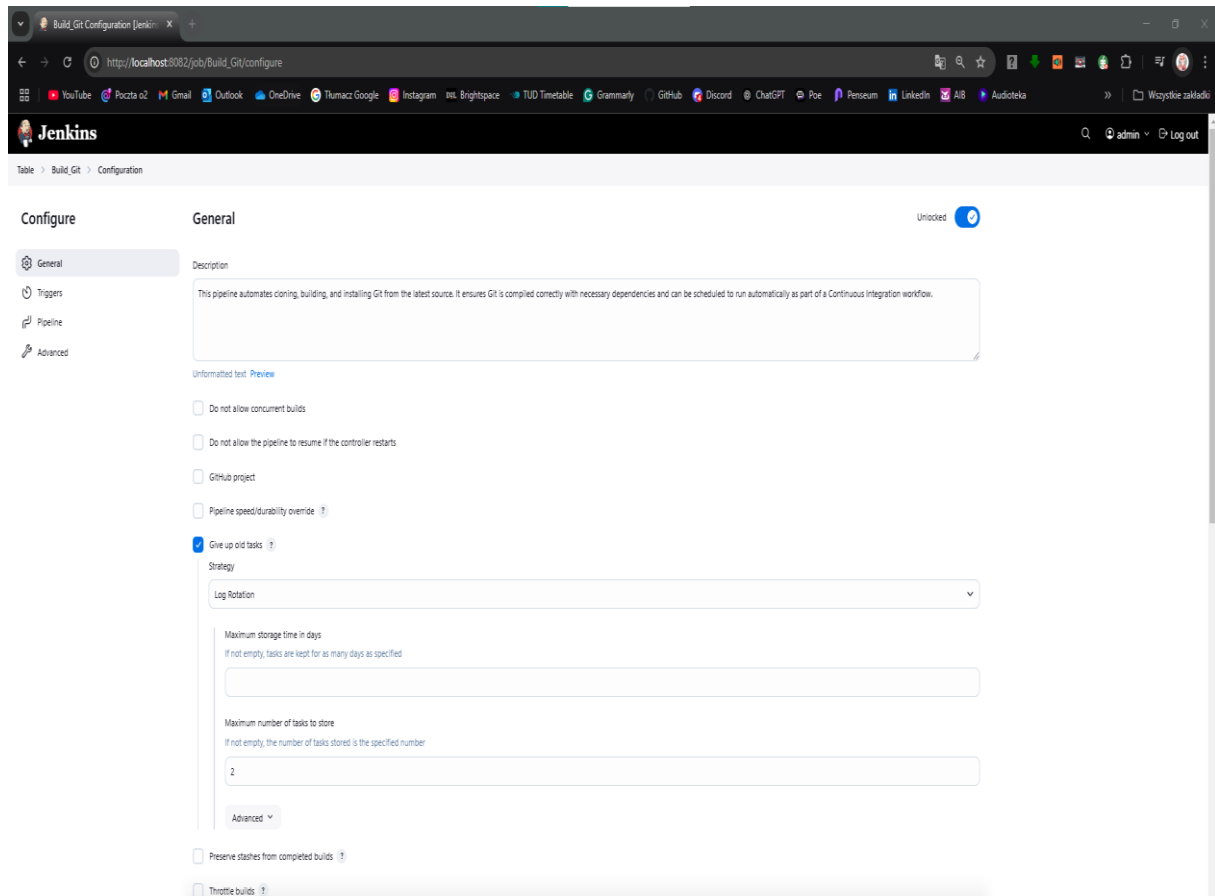
3. CREATING A NEW JENKINS PROJECT

This screenshot demonstrates the creation of a new Jenkins pipeline project named `Build_Git`, selecting the "Pipeline" option.



4. PIPELINE CONFIGURATION (GENERAL SETTINGS)

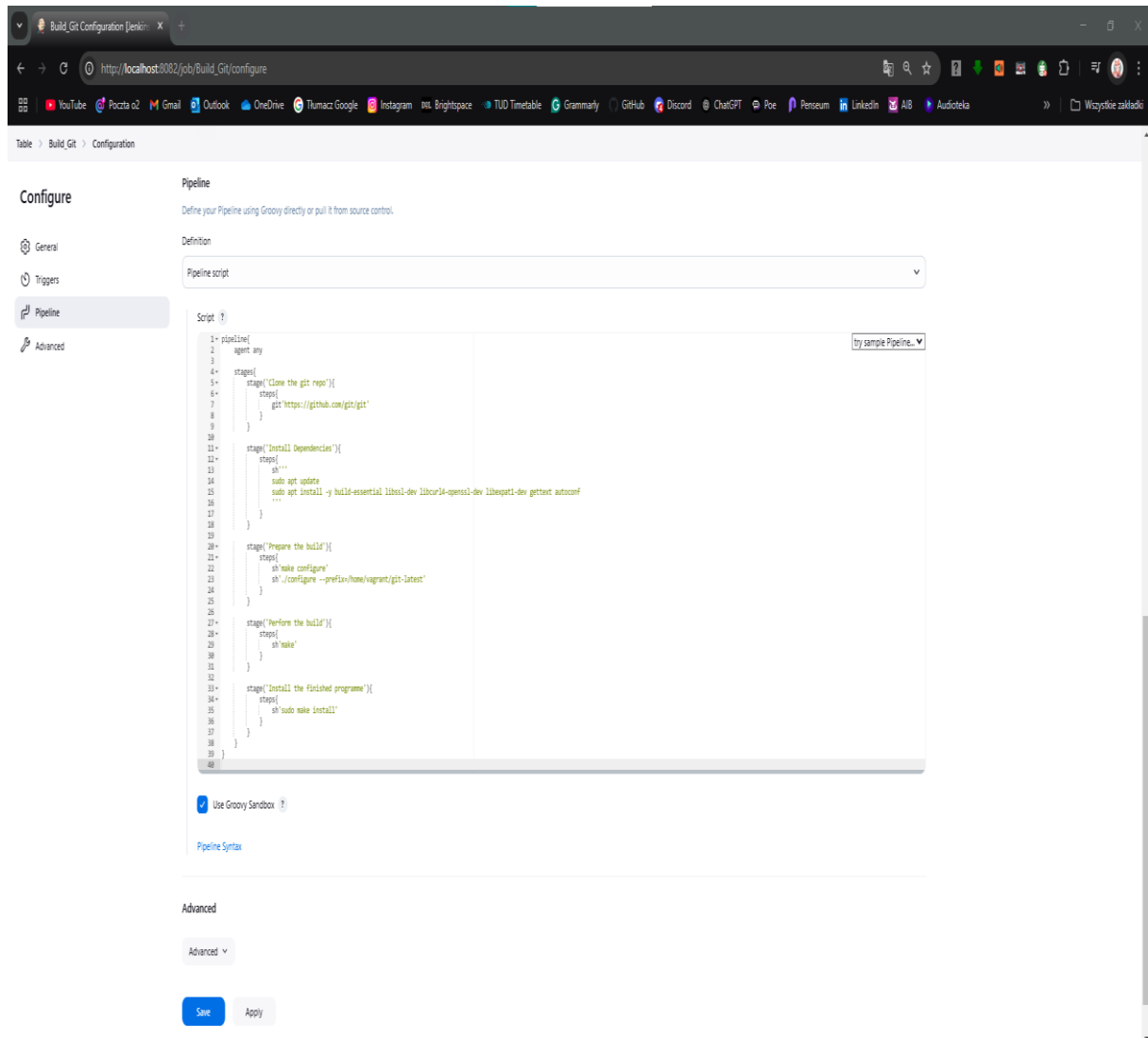
This image displays the pipeline configuration settings, including the description and log retention policies.



5. PIPELINE SCRIPT CONFIGURATION

This screenshot shows the Jenkins pipeline script, which consists of the following build stages:

- Cloning the Git repository.
- Configuring the build.
- Compiling Git.
- Installing the final program.

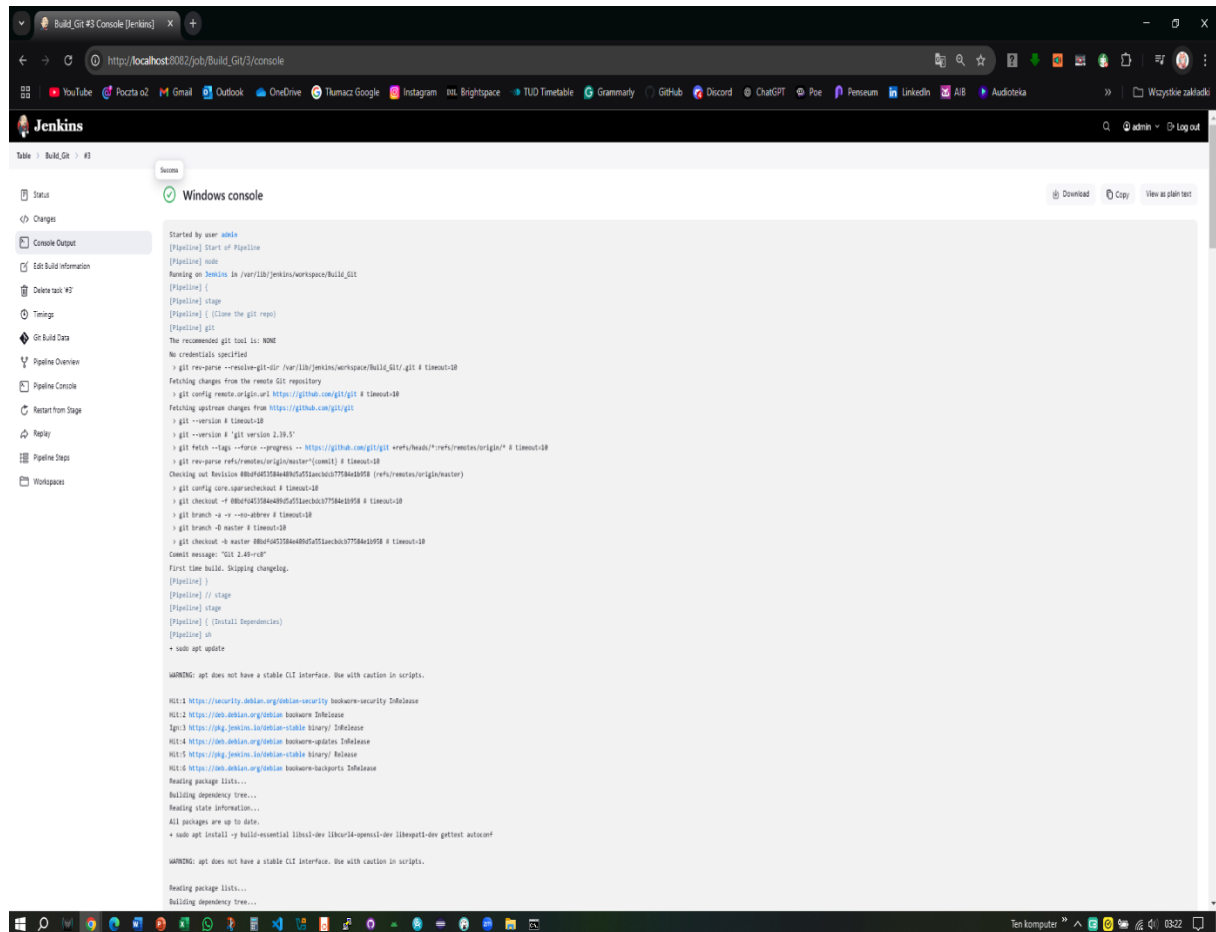


The screenshot displays the Jenkins web interface for configuring a pipeline. The browser address bar shows `http://localhost:8082/job/Build_Git/configure`. The left sidebar contains navigation links: General, Triggers, Pipeline (selected), and Advanced. The main content area is titled "Configure" and "Pipeline". It includes a "Definition" dropdown set to "Pipeline script". Below this is a "Script" text area containing a Groovy pipeline script. The script defines four stages: "Clone the git repo", "Install Dependencies", "Prepare the build", and "Perform the build". Below the script area, there is a checkbox for "Use Groovy Sandbox" which is checked, and a "Pipeline Syntax" link. At the bottom, there are "Save" and "Apply" buttons.

```
1 pipeline{
2   agent any
3
4   stages{
5     stage('Clone the git repo'){
6       steps{
7         git 'https://github.com/git/git'
8       }
9     }
10
11    stage('Install Dependencies'){
12      steps{
13        sh'''
14          sudo apt update
15          sudo apt install -y build-essential libssl-dev libcurl4-openssl-dev libexpat-dev gettext autoconf
16          ...
17        '''
18      }
19    }
20
21    stage('Prepare the build'){
22      steps{
23        sh 'make configure'
24        sh './configure --prefix=/home/vagrant/git-latest'
25      }
26    }
27
28    stage('Perform the build'){
29      steps{
30        sh 'make'
31      }
32    }
33
34    stage('Install the finished programme'){
35      steps{
36        sh 'sudo make install'
37      }
38    }
39  }
40 }
```

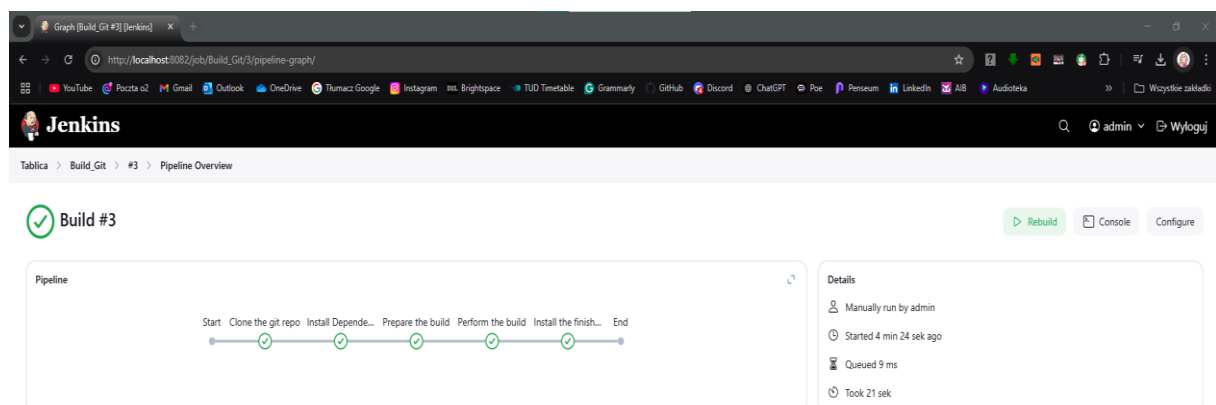
6. VERIFYING PIPELINE EXECUTION

This image captures the Console Output, showing the pipeline running successfully, with each stage executed correctly.



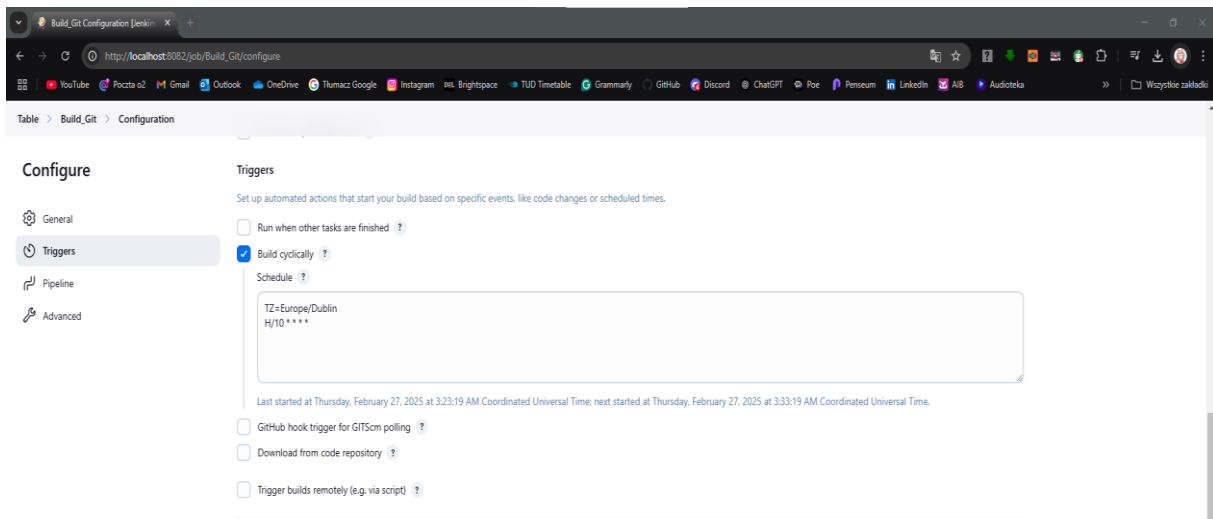
7. SUCCESSFUL PIPELINE EXECUTION OVERVIEW

This screenshot displays the graphical Pipeline Overview, confirming that all stages have completed successfully. The pipeline was executed multiple times, with at least three successful builds, demonstrating its stability and proper configuration.



8. CONFIGURING AUTOMATED BUILD SCHEDULING

This screenshot verifies that the build is scheduled to run every 10 minutes, using the correct cron syntax: `H/10 * * * *`



9. GIT VERSION VERIFICATION ON VM

This final screenshot demonstrates that the compiled Git version is installed and functional by executing: `~/git-latest/bin/git -version`

```
vagrant@bookworm: ~  
Microsoft Windows [Version 10.0.19045.5487]  
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.  
  
C:\WINDOWS\system32>cd C:\Users\35389\Desktop\TU856 Modules\YEAR 3\Year 3 - Semester 2\Introduction to DevOps - Eoin Rogers\Labs\Week 5  
C:\Users\35389\Desktop\TU856 Modules\YEAR 3\Year 3 - Semester 2\Introduction to DevOps - Eoin Rogers\Labs\Week 5>vagrant ssh  
Linux bookworm 6.1.0-31-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Feb 27 03:10:44 2025 from 10.0.2.2  
Linux bookworm 6.1.0-31-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Feb 27 03:10:44 2025 from 10.0.2.2  
vagrant@bookworm:~$ ~/git-latest/bin/git --version  
git version 2.49.0.rc0  
vagrant@bookworm:~$
```