

ASSIGNMENT 5 DevOps

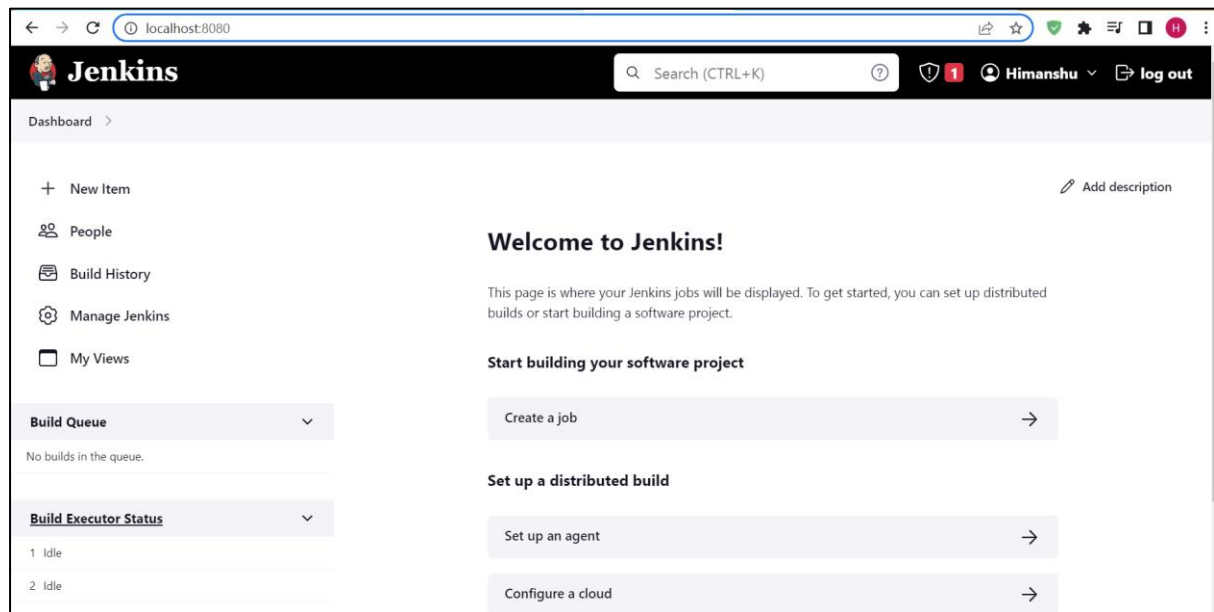
IT_BE_41_Himanshu Shukla

1) Set up a Jenkins job that picks up a Java application from a GitHub repository, builds it, and runs it.

Prerequisite: github account, jenkins installed on system([click](#)), jdk 11/17/22(any one)

So lets create and run a job in jenkins for simple java application

1)after Installing the jenkins go to the localhost:8080

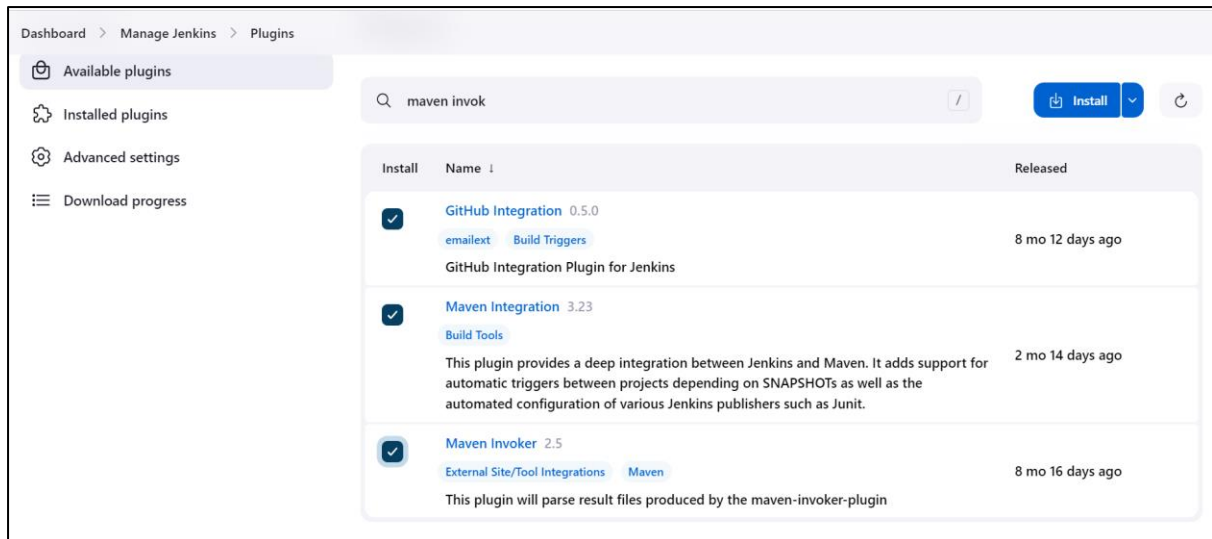


2)Now click on “**manage jenkins**” with setting icon on left side bar then go to “**plugins**” after that click on “**available plugins**” and search for three main plugins

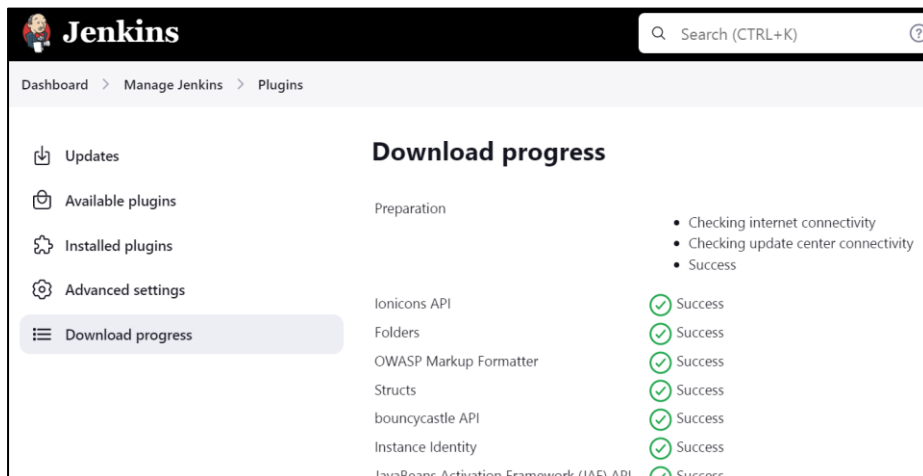
a)github integration plugins

b)maven integration

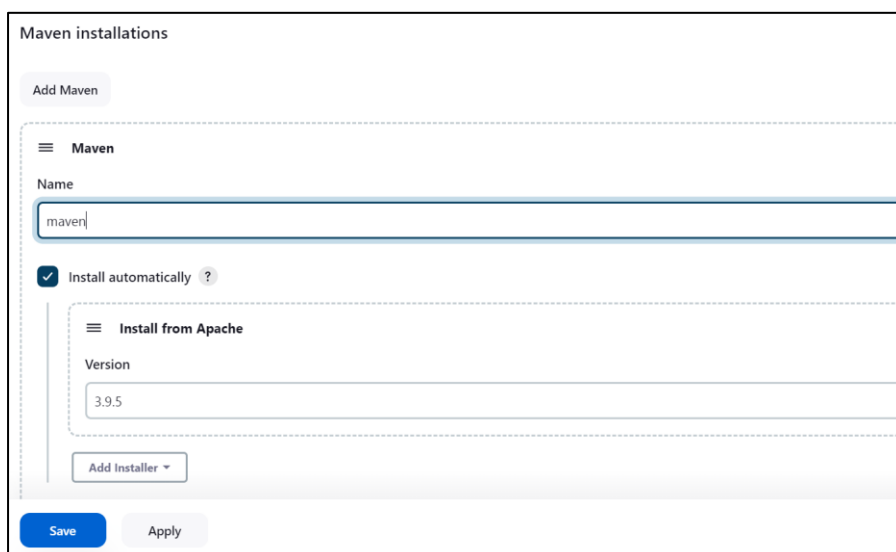
c)maven invoker,and install all this



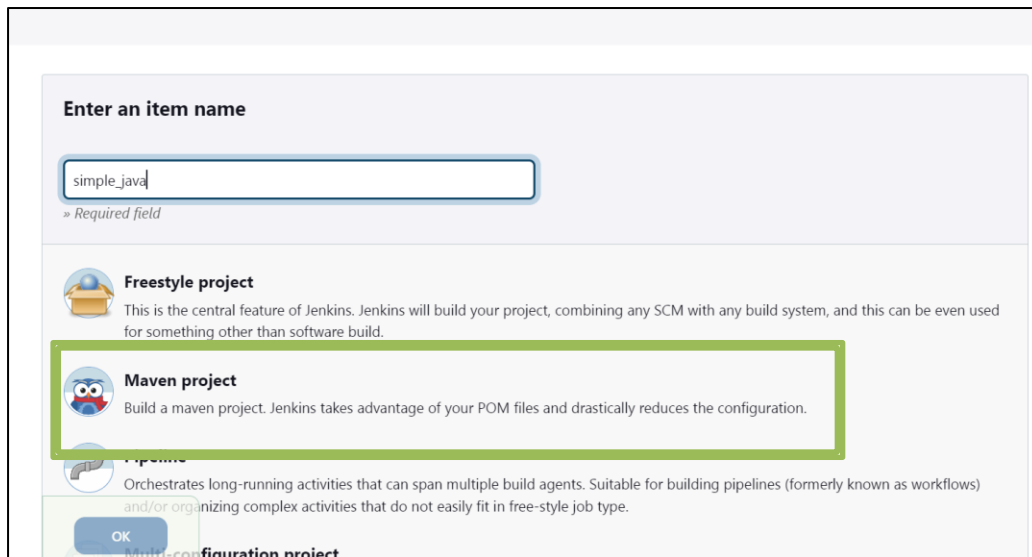
Now all has been installed



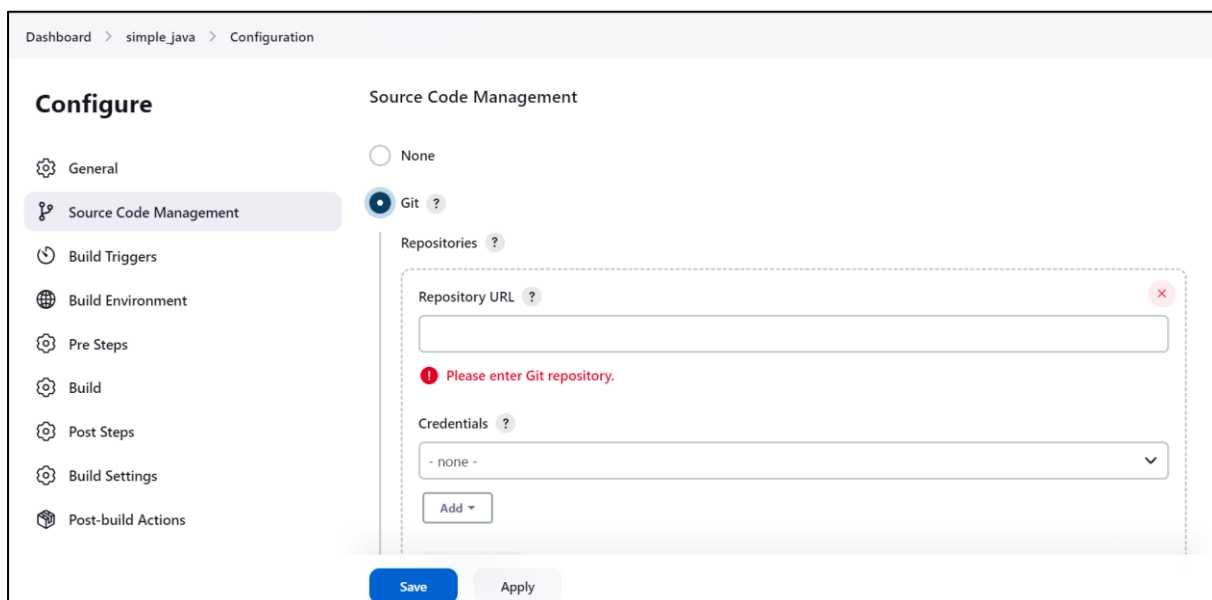
Again go to dashboard click on manage jenkins then click on “tools” and scroll to the bottom under maven installation add maven the click on save and apply



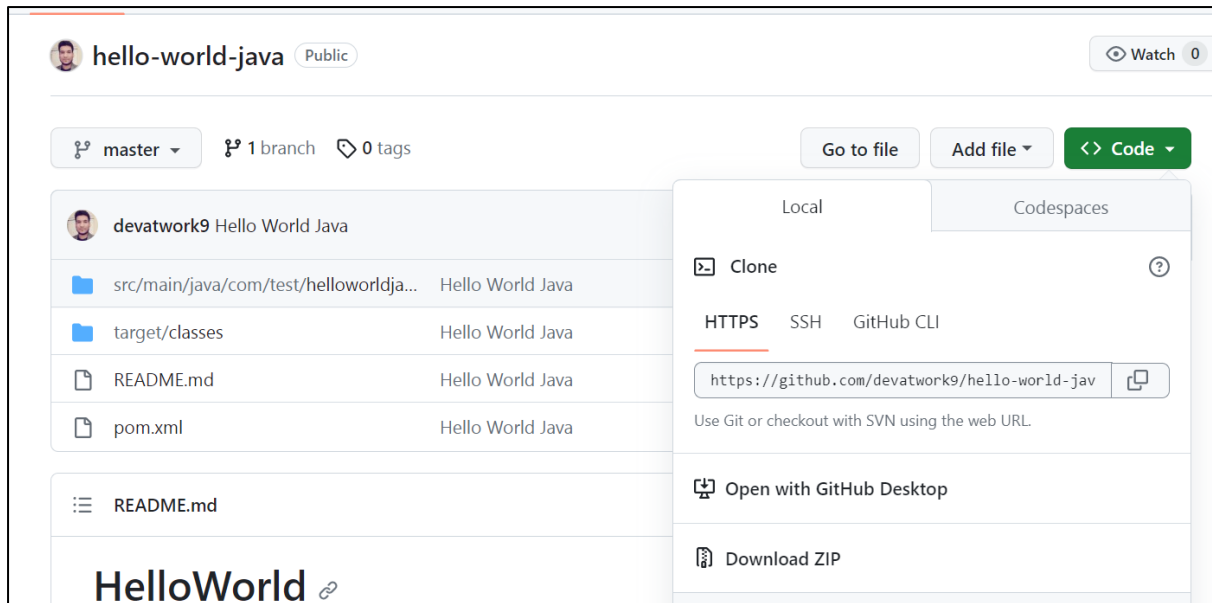
3) go to dashboard and click on new item and named the job as “simple_java” and click on second option “**maven project**” then click ok (It is imp to click on maven project)



4) now click on source code management and click on git and paste the HTTPS URL of simple java based application from github (the chosen repo of github should to be public and must contain pom.xml[maven])



Here below I have chosen a random repo([click](#)) of someone of basic java application of hello world



Now just copy the HTTPS url and paste in repository url ,scroll down in build section under goals and action write “clean install package”

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

Build

Root POM ?

Goals and options ?

Advanced ▾

then scroll to bottom and click on save

5)after that click on “build now” from left side bar of main dashboard and below there will be the build history

Dashboard > simple_java >

</> Changes

Workspace

Build Now

Configure

Delete Maven project

Modules

Rename

Permalinks

- Last build (#3), 5 min 14 sec ago
- Last stable build (#3), 5 min 14 sec ago
- Last successful build (#3), 5 min 14 sec ago
- Last completed build (#3), 5 min 14 sec ago

Build History trend ▾

Filter builds... /

✓ #3	19 Oct 2023, 01:40
------	--------------------

Atom feed for all Atom feed for failures

6)Now just click on that tick and after that click on output console on left side bar

Dashboard > simple_java > #3 > Console Output

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#3'

Git Build Data

Redeploy Artifacts

See Fingerprints

Console Output

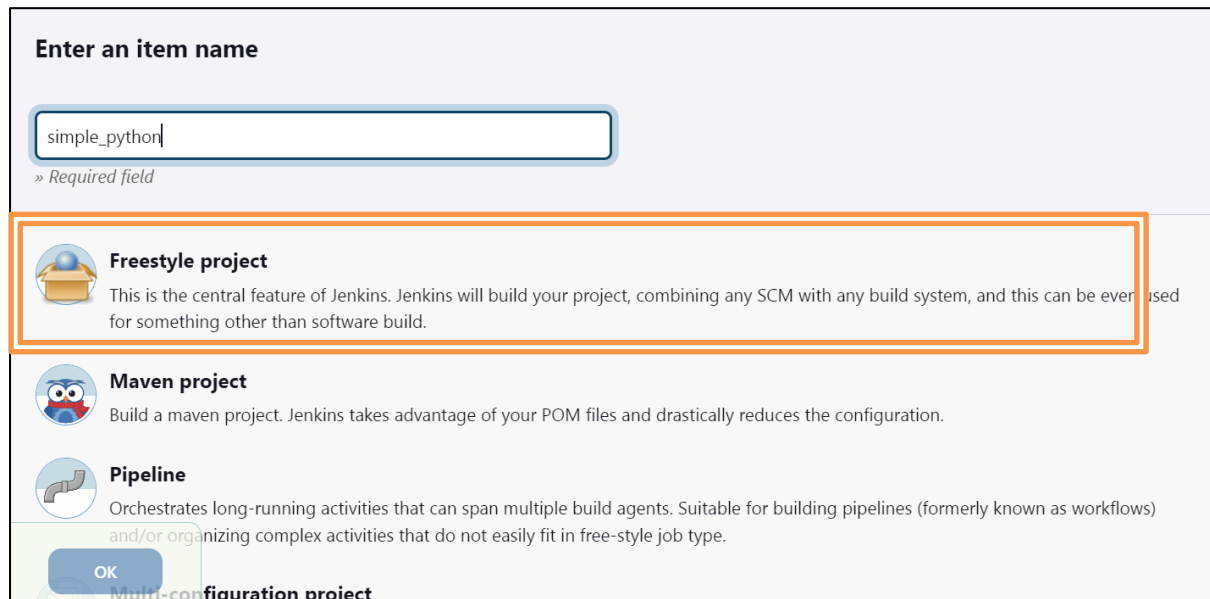
```
Started by user Himanshu
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\simple_java
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\simple_java\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/devatwork9/hello-world-java.git # timeout=10
Fetching upstream changes from https://github.com/devatwork9/hello-world-java.git
> git.exe --version # timeout=10
> git --version # 'git version 2.39.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/devatwork9/hello-world-java.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision eb2db461b56382c6694b0f67213a426babf2743c (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f eb2db461b56382c6694b0f67213a426babf2743c # timeout=10
Commit message: "Hello World Java"
```

Here the java application of hello world has been build and run using the jenkins job.

b) Set up a Jenkins job that picks up a python application from a GitHub repository, builds it, and runs it.

Before this go to Manage jenkins>Plugins>Available plugins and install python (like we have install other 3 plugin previously for java)

1)go to dashboard create a new item named simple_python click on **freestyle** project and click ok



Enter an item name

simple_python
» Required field

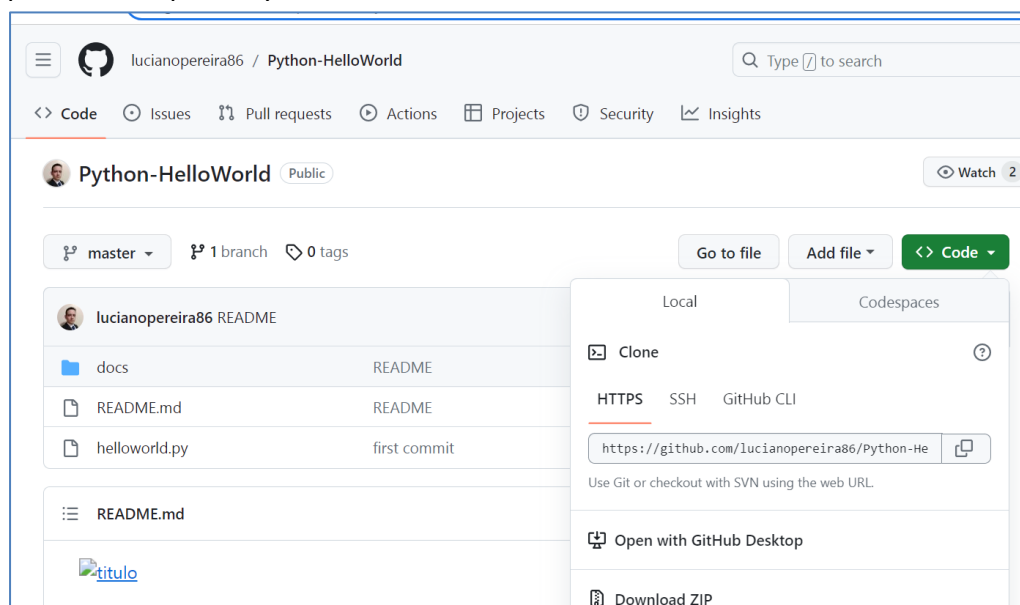
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

OK

2)now just open a random git repo([click](#)) of simple python program, copy the HTTPS Url and paste it in repository url



Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

`https://github.com/lucianopereira86/Python-HelloWorld.git`

3) Scroll down under build step click on execute window batch command and write this

Build Steps

Add build step ^

Filter

- Build a Visual Studio project or solution using MSBuild
- Execute Python script
- Execute Windows batch command**
- Execute shell
- GitHub PR: set 'pending' status
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Build Steps

≡ **Execute Windows batch command** ?

Command

See [the list of available environment variables](#)

```
virtualenv venv
source venv/bin/activate # for Linux/Mac, use `venv\Scripts\activate` on Windows
pip install -r requirements.txt
python .\helloworld.py
```

Advanced ▾

After this scroll to bottom click on apply and save

4)after this it will redirect you to your created job now just click on “build now”

The screenshot shows the Jenkins project page for 'simple_python'. On the left sidebar, the 'Build Now' button is highlighted with an orange box. The main area shows the project name 'Project simple_python' and a 'Permalinks' section with four links: 'Last build (#1), 10 min ago', 'Last stable build (#1), 10 min ago', 'Last successful build (#1), 10 min ago', and 'Last completed build (#1), 10 min ago'. Below this is the 'Build History' section, which has a search bar and a table of builds. The first build, '#1', is highlighted with an orange box and shows a green checkmark icon, indicating a successful build. The build time is '19 Oct 2023, 13:17'. At the bottom of the build history, there are links for 'Atom feed for all' and 'Atom feed for failures'.

Just click on green tick it will lead to output console

5) now the python application has been successfully build and runned

The screenshot shows the Jenkins console output for build #1. The left sidebar has 'Console Output' selected. The main area shows the console output with a green checkmark icon and the title 'Console Output'. The output text is as follows:

```
Started by user Himanshu
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\simple_python
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\simple_python\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/lucianopereira86/Python-HelloWorld.git # timeout=10
Fetching upstream changes from https://github.com/lucianopereira86/Python-HelloWorld.git
> git.exe --version # timeout=10
> git --version # 'git version 2.39.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/lucianopereira86/Python-HelloWorld.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 88cd1390a0a21edc68619e783379cb16cc2c59ad (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 88cd1390a0a21edc68619e783379cb16cc2c59ad # timeout=10
Commit message: "README"
```