01) What is Jenkins pipeline?

    Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.
    A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.
    Jenkins Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code".
    The definition of a Jenkins Pipeline is typically written into a text
    file (called a Jenkinsfile) which in turn is checked into a project's source control repository.

02) What scripting language is Jenkins pipeline syntax based on? >>>>>>>> (Groovy programming language)

03) What are the ways you can write pipeline in Jenkins?

    1) freeStyle pipline
    2) pipline
    3) Folder
    4) Multibrance pipline
    5) Orginazation pipline

localhost:8080/job/first_pipe2/configure

Dashboard > first_pipe2 > Configuration

Configure

Definition

Pipeline script from SCM

- General
- Advanced Project Options
- Pipeline

SCM ?

Git ?

Repositories ?

Repository URL ?   ✕

https://github.com/Badawi02/jenkins.git

Credentials ?

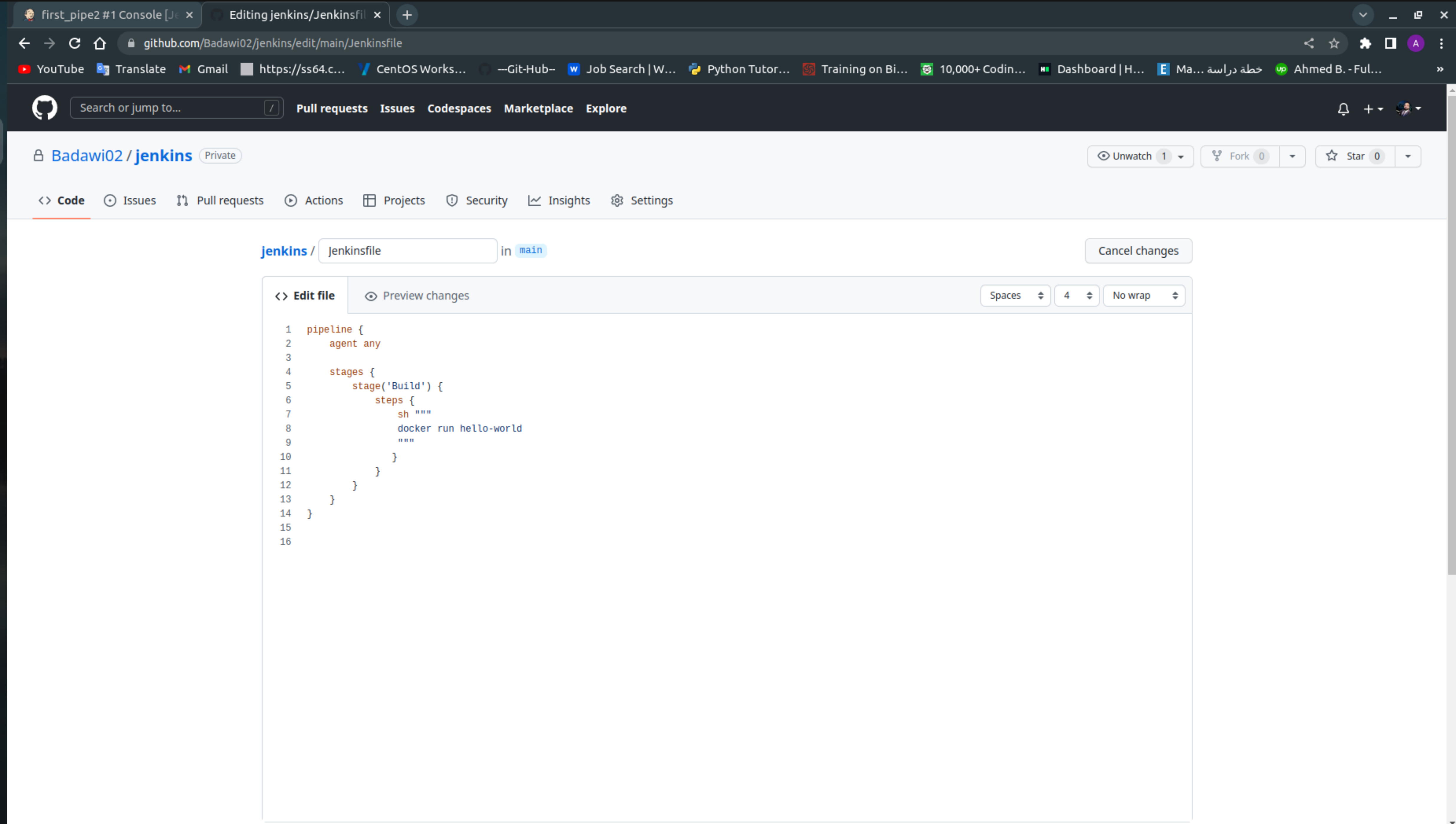Badawi02/******

＋ Add

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?   ✕

*/main

Add Branch

Save   Apply

Search or jump to...   Pull requests Issues Codespaces Marketplace Explore

🔒 Badawi02 / jenkins Private

Unwatch 1 ▾   Fork 0 ▾   Star 0 ▾

<> Code   ⊙ Issues   ⅔ Pull requests   ▶ Actions   ▦ Projects   🛡 Security   📈 Insights   ⚙ Settings

jenkins / Jenkinsfile   in main   Cancel changes

<> Edit file   👁 Preview changes   Spaces ⇕   4 ⇕   No wrap ⇕

```
1   pipeline {
2       agent any
3
4       stages {
5           stage('Build') {
6               steps {
7                   sh """
8                   docker run hello-world
9                   """
10              }
11          }
12      }
13  }
14  }
15
16
```

```
[Pipeline] sh
+ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pulling fs layer
2db29710123e: Verifying Checksum
2db29710123e: Download complete
2db29710123e: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Configuration

- General
- **Branch Sources**
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Git

Project Repository ?

```
https://github.com/Badawi02/jenkins.git
```

Credentials ?

```
Badawi02/******                                    ⌄
```

[+ Add]

Behaviours

Discover branches ✕

?

Filter by name (with wildcards) ✕

Include ?

```
*
```

Exclude ?

```
|
```

[Add ⌄]

Property strategy

```
All branches get the same properties              ⌄
```

[Save] [Apply]

Configuration

- ⚙️ General
- ⚙️ Branch Sources
- ⚙️ **Build Configuration**
- ⚙️ Scan Multibranch Pipeline Triggers
- ⚙️ Orphaned Item Strategy
- 🎨 Appearance
- 〰️ Health metrics
- 🔧 Properties

Build Configuration

Mode

by Jenkinsfile ⌄

Script Path ❓

Jenkinsfile

Scan Multibranch Pipeline Triggers

☐ Periodically if not otherwise run ❓

Orphaned Item Strategy

Jobs for removed SCM heads (i.e. deleted branches) can be removed immediately or kept based on a desired retention strategy. By default, jobs will be removed as soon as Jenkins determines their associated SCM head no longer exists. As an example, it may be useful to configure a different retention strategy to be able to examine build results of a branch after it has been removed.

☐ Abort builds ❓

☐ Discard old items

Appearance

Icon ❓

Metadata Folder Icon ⌄

Save   Apply

<> Code   Issues   Pull requests   Actions   Projects   Security   Insights   Settings

jenkins /   Jenkinsfile   in `test` /   Cancel changes

<> Edit file   Preview changes   Spaces   4   No wrap

```
 1  pipeline {
 2      agent any
 3      stages {
 4          stage('Master Branch Deploy Code') {
 5              when {
 6                  branch 'main'
 7              }
 8              steps {
 9                  sh """
10                      echo "Building Artifact from Master branch"
11                  """
12
13                  sh """
14                      echo "Deploying Code from Master branch"
15                  """
16              }
17          }
18          stage('Develop Branch Deploy Code') {
19              when {
20                  branch 'dev'
21              }
22              steps {
23                  sh """
24                      echo "Building Artifact from Develop branch"
25                  """
26                  sh """
27                      echo "Deploying Code from Develop branch"
28                  """
29              }
30          }
31      }
32  }
```

Commit changes

Update Jenkinsfile

Add an optional extended description

⬤ Scan Multibranch Pipeline Log

Status

Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

  View as plain text

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Project Relationship

Check File Fingerprint

Rename

Pipeline Syntax

Credentials

Build Queue (2)

multi_pipline2 » main

multi_pipline2 » dev

Build Executor Status

1 Idle

2 Idle

Progress: ▭▭▭▭▭ ⊠

```
Started
[Sat Feb 04 02:30:46 UTC 2023] Starting branch indexing...
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
 > git ls-remote --symref -- https://github.com/Badawi02/jenkins.git # timeout=10
 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-dca547d970d4c69b9dbeb1e3bf2e0697/.git # timeout=10
Setting origin to https://github.com/Badawi02/jenkins.git
 > git config remote.origin.url https://github.com/Badawi02/jenkins.git # timeout=10
Fetching & pruning origin...
Listing remote references...
 > git config --get remote.origin.url # timeout=10
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
 > git ls-remote -h -- https://github.com/Badawi02/jenkins.git # timeout=10
Fetching upstream changes from origin
 > git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress --prune -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking branches...
  Checking branch main
      'Jenkinsfile' found
    Met criteria
Scheduled build for branch: main
  Checking branch dev
      'Jenkinsfile' found
    Met criteria
Scheduled build for branch: dev
Processed 2 branches
[Sat Feb 04 02:30:50 UTC 2023] Finished branch indexing. Indexing took 4.2 sec
Finished: SUCCESS
```

```
> git init /var/jenkins_home/workspace/multi_pipline2_dev # timeout=10
Fetching upstream changes from https://github.com/Badawi02/jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
 > git fetch --no-tags --force --progress -- https://github.com/Badawi02/jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Badawi02/jenkins.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 40d7d4b3fb9982a0f17d0cb34de6fbc54ce0a9b8 (dev)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 40d7d4b3fb9982a0f17d0cb34de6fbc54ce0a9b8 # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Master Branch Deploy Code)
Stage "Master Branch Deploy Code" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Develop Branch Deploy Code)
[Pipeline] sh
+ echo Building Artifact from Develop branch
Building Artifact from Develop branch
[Pipeline] sh
+ echo Deploying Code from Develop branch
Deploying Code from Develop branch
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```
 > git init /var/jenkins_home/workspace/multi_pipline2_main@2 # timeout=10
Fetching upstream changes from https://github.com/Badawi02/jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
 > git fetch --no-tags --force --progress -- https://github.com/Badawi02/jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Badawi02/jenkins.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 40d7d4b3fb9982a0f17d0cb34de6fbc54ce0a9b8 (main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 40d7d4b3fb9982a0f17d0cb34de6fbc54ce0a9b8 # timeout=10
Commit message: "Update Jenkinsfile"
 > git rev-list --no-walk 40d7d4b3fb9982a0f17d0cb34de6fbc54ce0a9b8 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Master Branch Deploy Code)
[Pipeline] sh
+ echo Building Artifact from Master branch
Building Artifact from Master branch
[Pipeline] sh
+ echo Deploying Code from Master branch
Deploying Code from Master branch
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Develop Branch Deploy Code)
Stage "Develop Branch Deploy Code" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

localhost:8080/job/multi_pipline2/

Jenkins

Search (CTRL+K)

ahmed badawi   log out

Dashboard > multi_pipline2 >

Status

Configure

Scan Multibranch Pipeline Now

Scan Multibranch Pipeline Log

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Project Relationship

Check File Fingerprint

Rename

Pipeline Syntax

Credentials

multi_pipline2

Disable Multibranch Pipeline

Branches (2)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	dev	1 min 1 sec #2	N/A	6.6 sec	▷
✓	☁	main	1 min 19 sec #5	7 min 50 sec #2	8.3 sec	▷

Icon: S M L

Icon legend   Atom feed for all   Atom feed for failures   Atom feed for just latest builds

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle