what are different method to trigger pipeline in jenkins ?

- Manually
- option in pipeline configuration called “build periodically”
- webhook
- organization folder job

what is the benefit of using master-slave architecture rather than building on master only ?

=> sometimes each part of code needs a specific os to run on
=> part of code needs a specific version of java and other part needs another one
=> to run multiple pipelines at the same time

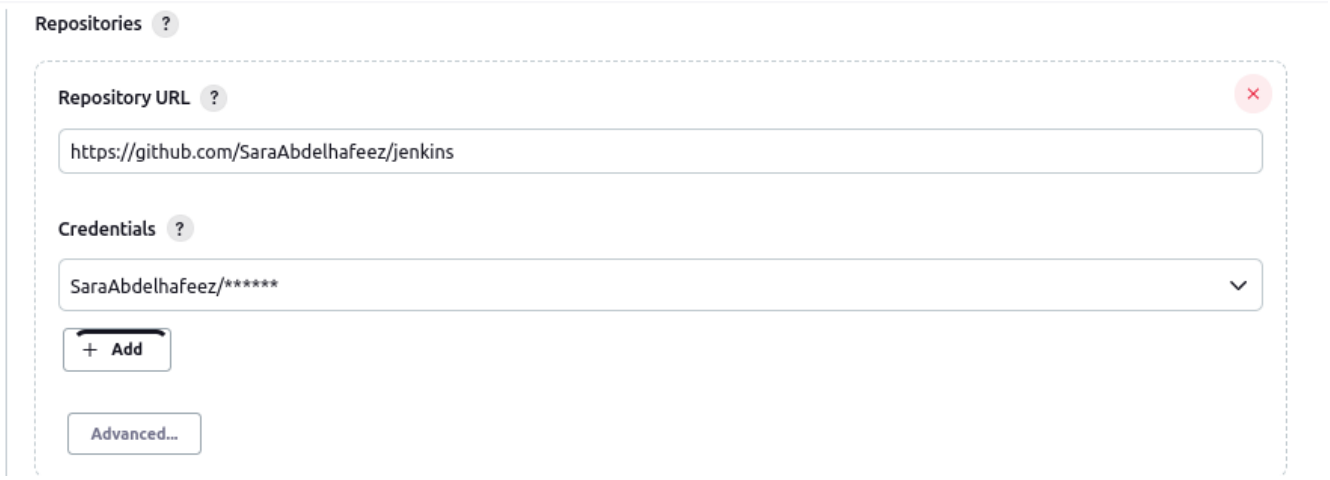
what is different between authorization and authentication ?

Authentication verifies the identity of a user or service
authorization determines their access rights

what is the benefit of making organization job in jenkins?

Organization Folders enable Jenkins to monitor an entire GitHub Organization

make jenkins-shared-library and make your jenkinsfile which you used in lab2 to point to this library



Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

Jenkinsfile

SaraAbdelhafeez / jenkins Private

Unwatch

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

SaraAbdelhafeez jenkins 846a7d1 11 minutes ago 2 commits

vars	jenkins	11 minutes ago
Jenkinsfile	jenkins	11 minutes ago
README.md	first commit	13 minutes ago

```
Started by user Sara Abdelhafeez
Obtained Jenkinsfile from git https://github.com/SaraAbdelhafeez/jenkins
Loading library jenkins@main
Examining SaraAbdelhafeez/jenkins
Attempting to resolve main as a branch
Resolved main as branch main at revision 3e2b497e09a3400eb815ed6121e8a95f9aa199e3
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential git-token
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/task3-
pipeline@libs/851b4766a7a657ac98e09078f17b78ef0e13bd56f45ac3c7988657ee563b8924/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/SaraAbdelhafeez/jenkins.git # timeout=10
Fetching without tags
Fetching upstream changes from https://github.com/SaraAbdelhafeez/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/SaraAbdelhafeez/jenkins.git +refs/h
timeout=10
Checking out Revision 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 (main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 # timeout=10
Commit message: "Update Jenkinsfile"
 > git rev-list --no-walk 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 # timeout=10
[Pipeline] Start of Pipeline
```

```
 > git rev-list --no-walk 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 # timeout=10
[Pipeline] Start of Pipeline
[Pipeline] node
Running on jenkins-slave in /home/ubuntu/jenkins_home/workspace/task3-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential git-token
Fetching changes from the remote Git repository
 > git rev-parse --resolve-git-dir /home/ubuntu/jenkins_home/workspace/task3-pipeline/.git # timeout
 > git config remote.origin.url https://github.com/SaraAbdelhafeez/jenkins # timeout=10
Fetching upstream changes from https://github.com/SaraAbdelhafeez/jenkins
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/SaraAbdelhafeez/jenkins +refs/heads/*:r
Checking out Revision 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 (refs/remotes/origin/main)
Commit message: "Update Jenkinsfile"
[Pipeline] }
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3e2b497e09a3400eb815ed6121e8a95f9aa199e3 # timeout=10
[Pipeline] // stage
[Pipeline] withEnv
```

```
 > git checkout -f 3e2b497e09a3
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (check)
[Pipeline] echo
checking your code
[Pipeline] echo
dev
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
testing your app
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deployment)
[Pipeline] echo
your code is deployed right now
[Pipeline] echo
this build number 8
[Pipeline] }
[Pipeline] // stage
```

```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
testing your app
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deployment)
[Pipeline] echo
your code is deployed right now
[Pipeline] echo
this build number 8
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```