Jenkins

1) what are different method to trigger pipeline in Jenkins ?

1-Trigger builds remotely.
2-Build periodically.
3-GitHub webhook trigger.
4-Poll SCM
5-Build after other piplines are built.
6-Build manually.

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

- **Produce backups** — Obviously, the major utility of master-slave database architecture is its ability to provide reliable backups through their chain of slave databases. The slave database can be turned off without affecting the master database's activities. This is because live data snapshots will be replicated to the slave database, and data resources will remain intact even if the master database fails.
- **Scale out the application** — When your number of users increases and your app's usage skyrockets, it's critical to scale out your app so that your audience has a consistent experience. By splitting your data load across numerous databases, the master-slave database architecture can help you scale out your application.
- **Split your work without causing a mess** — You can use your slave nodes to collect the necessary data and generate a complete report, while the master node pushes new updates and features to your application server.
- **Top-notch performance** — It is extremely fast and fetches data without any issues, imposing no restrictions regarding performance and time to load the data.

3) what is the difference between authorization and authentication?

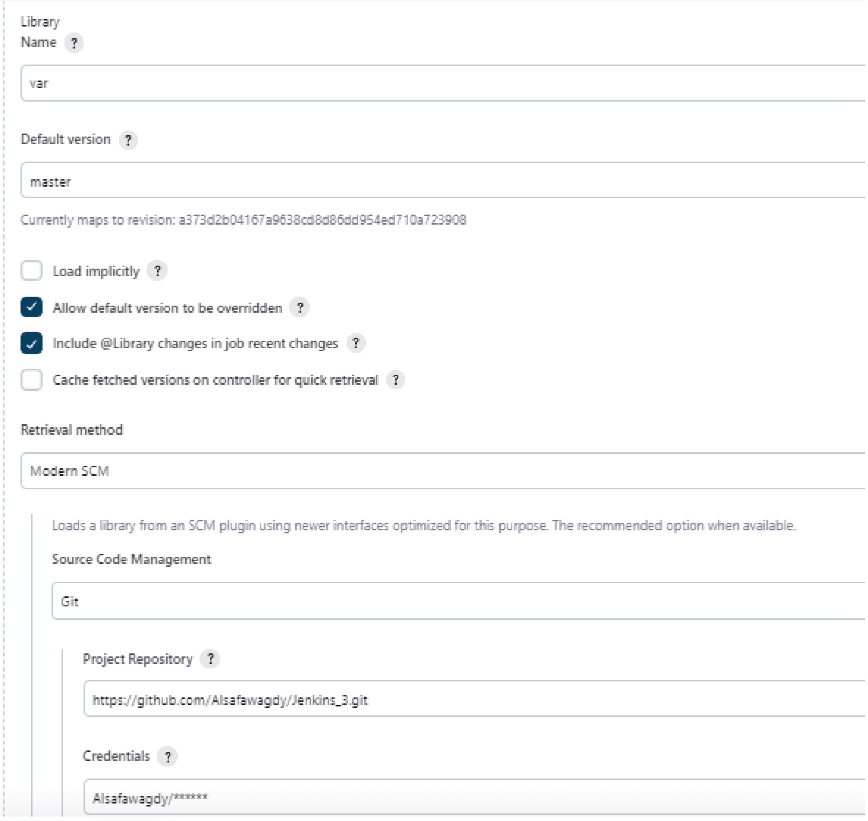
- Authentication verifies the identity of a user or service
- Authorization determines their access rights

4) what is the benefit of making organization job in Jenkins?

- Organization Folders enable Jenkins to monitor an entire GitHub Organization, Bitbucket Team/Project, GitLab organization, or Gitea organization

- automatically create new Multibranch Pipelines for repositories which contain branches and pull requests containing a Jenkinsfile

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



Library Name ?

var

Default version ?

master

Currently maps to revision: a373d2b04167a9638cd8d86dd954ed710a723908

☐ Load implicitly ?

☑ Allow default version to be overridden ?

☑ Include @Library changes in job recent changes ?

☐ Cache fetched versions on controller for quick retrieval ?

Retrieval method

Modern SCM

Loads a library from an SCM plugin using newer interfaces optimized for this purpose. The recommended option when available.

Source Code Management

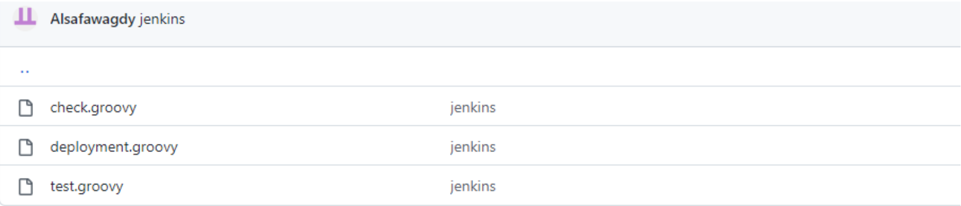
Git

Project Repository ?

https://github.com/Alsafawagdy/Jenkins_3.git

Credentials ?

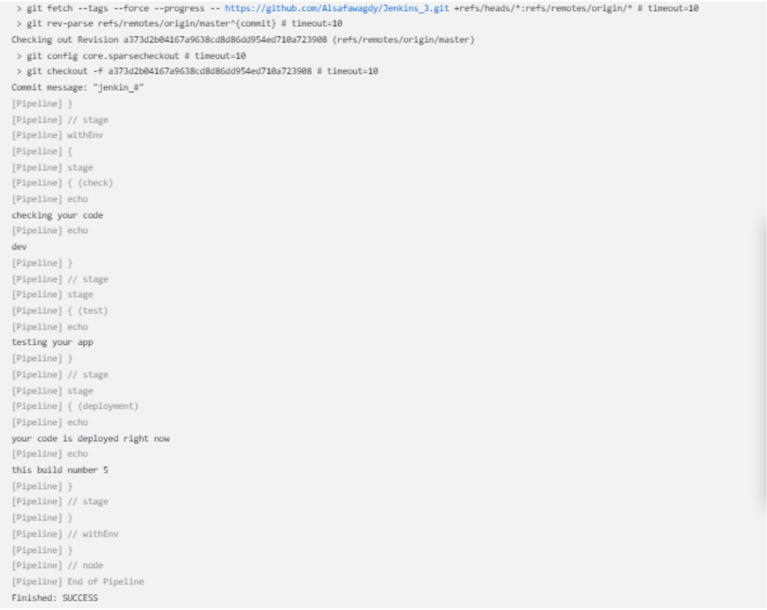
Alsafawagdy/******



Alsafawagdy jenkins

..

check.groovy	jenkins
deployment.groovy	jenkins
test.groovy	jenkins



```
 > git fetch --tags --force --progress -- https://github.com/Alsafawagdy/Jenkins_3.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision a373d2b04167a9638cd8d86dd954ed710a723908 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f a373d2b04167a9638cd8d86dd954ed710a723908 # timeout=10
Commit message: "jenkin_#"
[Pipeline] }
[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 5
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```