

what are different method to trigger pipeline in jenkins ?

- Trigger builds remotely.
- Build after other projects are built.
- Build periodically.
- GitHub hook trigger for GITScm polling.
- Poll SCM.

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.

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, Bitbucket Team/Project, GitLab organization, or Gitea organization and automatically create new Multibranch Pipelines for repositories which contain branches and pull requests containing a Jenkinsfile

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

Dashboard > Manage Jenkins > Configure System >

Global Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

Library

Name ?

mylib

Default version ?

main

Currently maps to revision: fdc025708572054bc4b6d2c5769b6a3d8204bc99

☐ 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/AmrGomaa96/test

Save

Apply

	check.groovy	Add files via upload	8 minutes ago
	deployment.groovy	Add files via upload	8 minutes ago
	test.groovy	Add files via upload	8 minutes ago

Dashboard > test > #1

- View as plain text
- Edit Build Information
- Delete build '#1'
- Git Build Data
- Replay
- Pipeline Steps
- Workspaces

```
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-token
Cloning the remote Git repository
Cloning repository https://github.com/AmrGomaa96/test
> git init /var/jenkins_home/workspace/test # timeout=10
Fetching upstream changes from https://github.com/AmrGomaa96/test
> git --version # timeout=10
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/AmrGomaa96/test +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/AmrGomaa96/test # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision fdc025708572054bc4b6d2c5769b6a3d8204bc99 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f fdc025708572054bc4b6d2c5769b6a3d8204bc99 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b main fdc025708572054bc4b6d2c5769b6a3d8204bc99 # timeout=10
Commit message: "Update Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] sh
+ ls
Dockerfile
Jenkinsfile
README.md
check.groovy
deployment.groovy
test.groovy
testfile
[Pipeline] }
[Pipeline] // stage
```

```
1 @Library('mylib') _
2 node {
3     stage('Checkout') {
4         git branch: 'main',
5             credentialsId: 'github-token',
6             url: 'https://github.com/AmrGomaa96/test'
7         sh "ls"
8     }
9 }
```

[Give feedback](#)

bonus >>>>> try to make new slave as container and configure master to use it

```
home > amr > jenkins > Dockerfile > ...
1 FROM ubuntu
2 USER root
3 RUN mkdir -p jenkins_home
4 RUN chmod 777 jenkins_home
5 RUN apt-get update -qq
6 RUN apt-get install openjdk-11-jdk -qq
7 RUN apt-get install openssh-server -qq
8 RUN useradd -ms /bin/bash jenkins
9
10 #install docker client
11 RUN apt-get update -qq
12 RUN apt-get install apt-transport-https curl gnupg-agent ca-certificates software-properties-common -y
13 RUN curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
14 RUN add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable" -y
15 RUN apt-get update -y
16 RUN apt-get install docker-ce docker-ce-cli containerd.io -y
17 RUN usermod -aG docker jenkins
18 USER jenkins
19 WORKDIR jenkins_home
20 CMD ["/bin/bash"]
21
```

```

root@31ab82ba15cc:/jenkins_home
amr@amrgomaa: ~/jenkins

amr@amrgomaa:~/jenkins$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1ab82ba15cc   slave:v0   "/bin/bash"             12 minutes ago Up 12 minutes
6566e12157    jenkins-v1 "/usr/bin/tini -- /u... 3 hours ago   Up 36 hours   0.0.0.0:2200 -> 2200/tcp, 0.0.0.0:2200-2200/tcp -> 0.0.0.0:5000-5000/tcp

amr@amrgomaa:~/jenkins$ docker exec -it 31ab82ba15cc bash
jenkins@31ab82ba15cc:/jenkins_home$ service ssh status
* sshd is running
jenkins@31ab82ba15cc:/jenkins_home$

```

[illegible]