Doc use panavanga intha num ku gpay(only above Rs200): 8056099616

DOCKER

Installing Docker on Ubuntu - go to official website (Docker docs)

```
mkdir fapp2
touch app.py
```

app.py

```
from flask import Flask
app=Flask(__name__)
@app.route('/')
def run():
    return "vicky genius"
app.run('0.0.0.0',5000)
```

```
touch Dockerfile
```

Dockerfile

```
FROM python:3.10
WORKDIR /app
COPY . /app
RUN pip install -r requirements.txt
```

```
EXPOSE 5000
CMD ["python3","app.py"]
```

```
touch requirements.txt
```

requirements.txt

```
Flask==3.0.1
```

```
docker build -t fapp2 .
```

```
docker run -p 5000:5000 fapp2
```

SCP

Create a file called data.txt and put some content in it

```
touch script.sh
```

```
#script.sh
scp -r data.txt vickybro@192.168.1.5:/home/vickybro/
```

This will send the data.txt from your system to the remote system with username “**vickybro**” and IP address “**192.168.1.5**”

Run the script file using

```
./script.sh
```

If permission error, “chmod +x script.sh”

SSH

```
ssh username@hostname_or_ip_address
```

If key pair not generated already

```
ssh-keygen
ssh-copy-id username@hostname_or_ip_address
ssh username@hostname_or_ip_address
```

JENKINS

sudo apt-get install jenkins
sudo apt-get install fontconfig openjdk-17-jre (if Java doesnt exist)

sudo systemctl start jenkins
sudo systemctl status jenkins

Go to https://localhost:8080 - default Jenkins server

It will ask for the authentication key. Copy the path and use command

sudo cat “paste_the_path_here”

Paste the key from terminal to the Jenkins server

Create an account with username and password

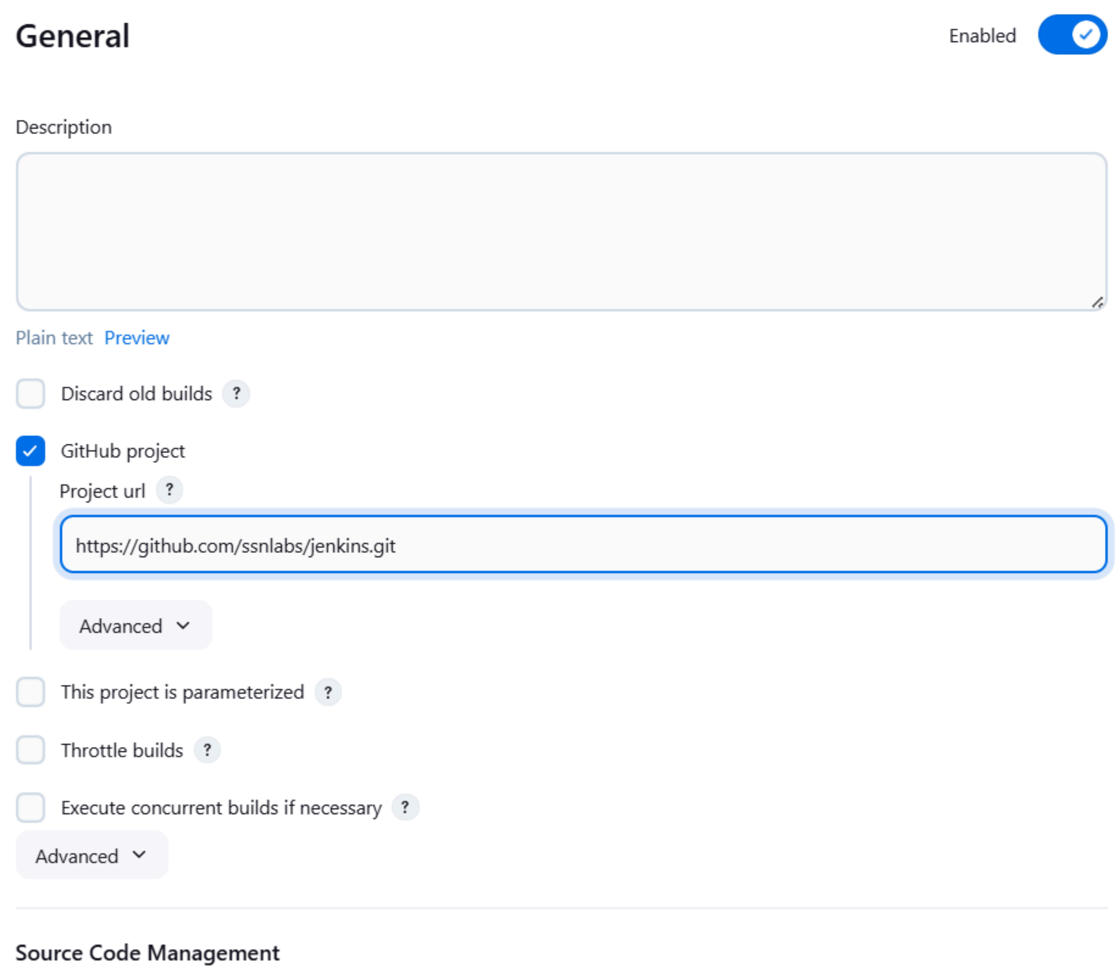
—--

Create a repository with a Java file
For eg: https://github.com/ssnlabs/jenkins.git

Jenkins

1. Click New Item

2. Give a name and select Freestyle Project
3. Select Github project and paste repository name



General

Enabled ✓

Description

Plain text Preview

☐ Discard old builds

✓ GitHub project

Project url

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

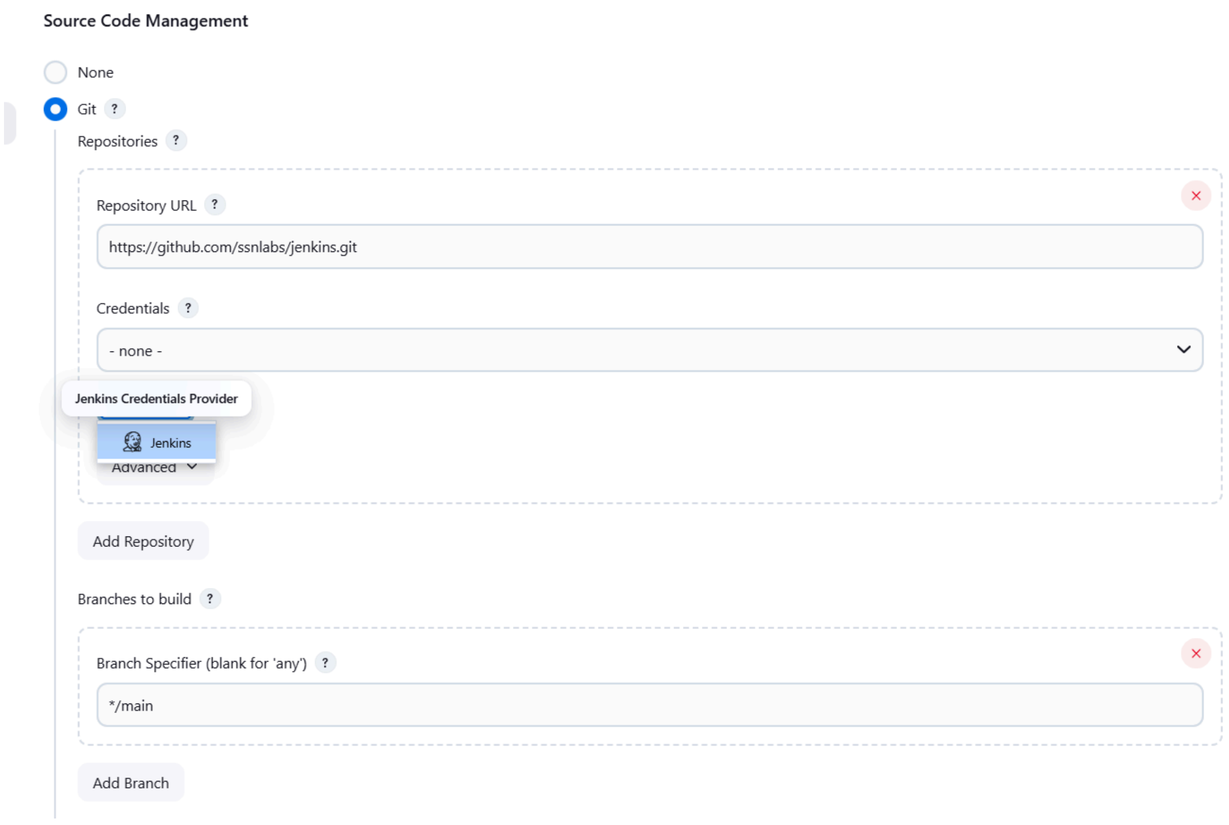
☐ This project is parameterized

☐ Throttle builds

☐ Execute concurrent builds if necessary

Source Code Management

4. Select Git, Under Credentials, Click Add & Jenkins Change “Branch Specifier” to */main


Source Code Management

None

[✓ Git]

Repositories

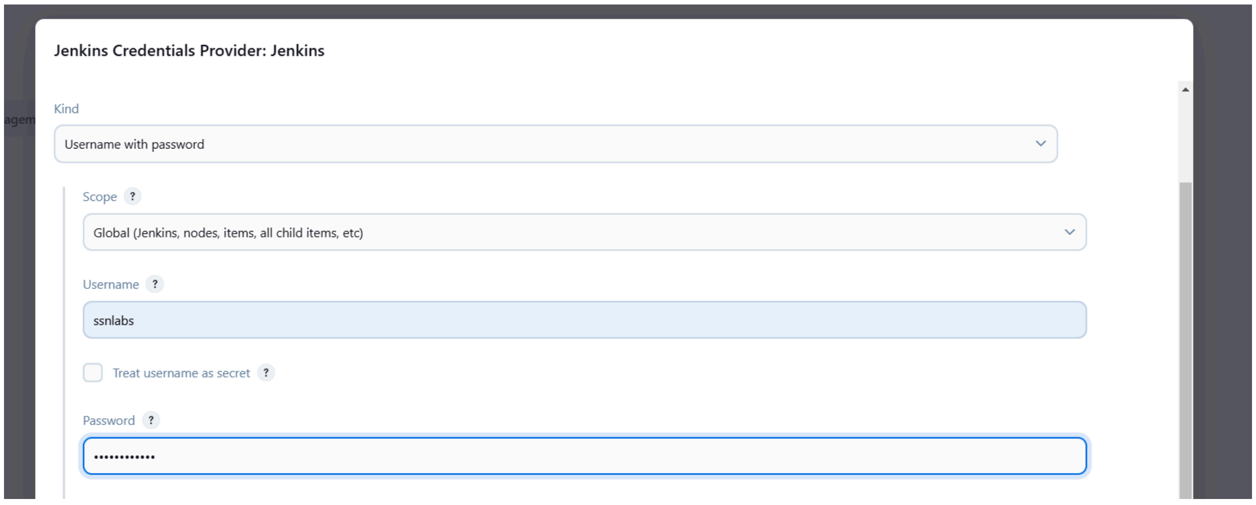
Repository URL: https://github.com/ssnlabs/jenkins.git

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */main

5. Enter your github username and password


Jenkins Credentials Provider: Jenkins

Kind: Username with password

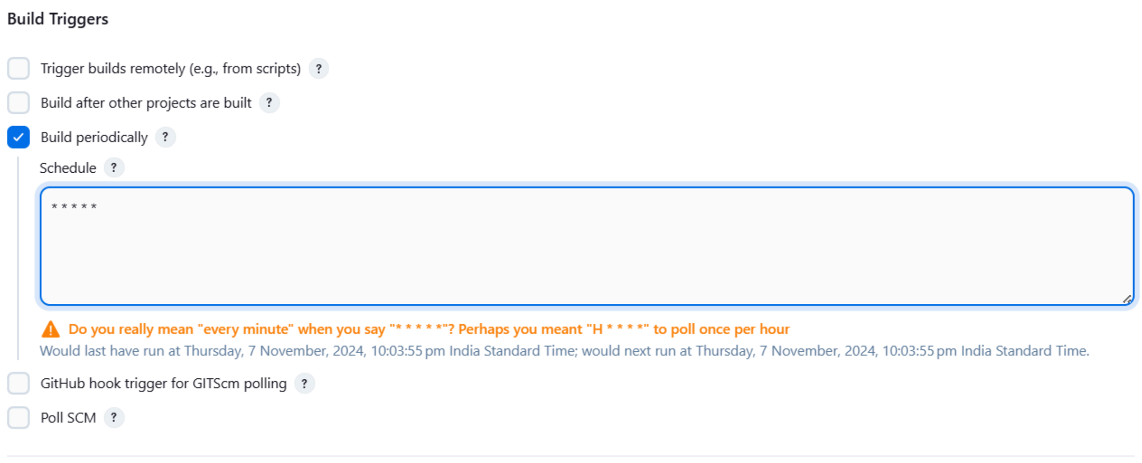
Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: ssnlabs

Treat username as secret

Password: ••••••••••••

6. Add * * * * * under Build Periodically -> Schedule (Cron Job - triggers a build every minute)



Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [x] Build periodically

Schedule

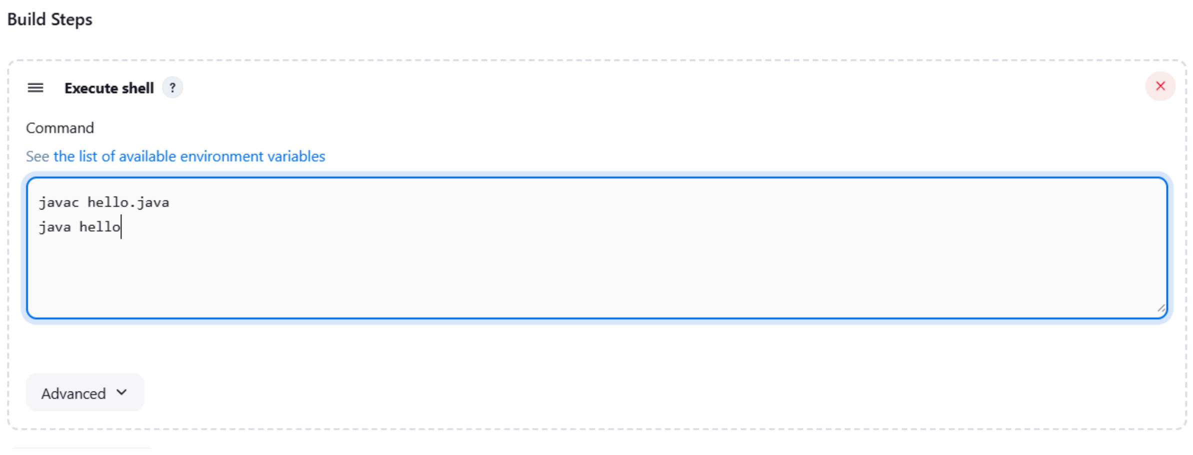
```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Thursday, 7 November, 2024, 10:03:55 pm India Standard Time; would next run at Thursday, 7 November, 2024, 10:03:55 pm India Standard Time.

- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

7. Under Build Step -> Select Execute Shell and enter the following



Build Steps

Execute shell

Command

See the list of available environment variables

```
javac hello.java
java hello
```

Advanced

8. Give Save and Build Now in next page
9. Enjoy!