

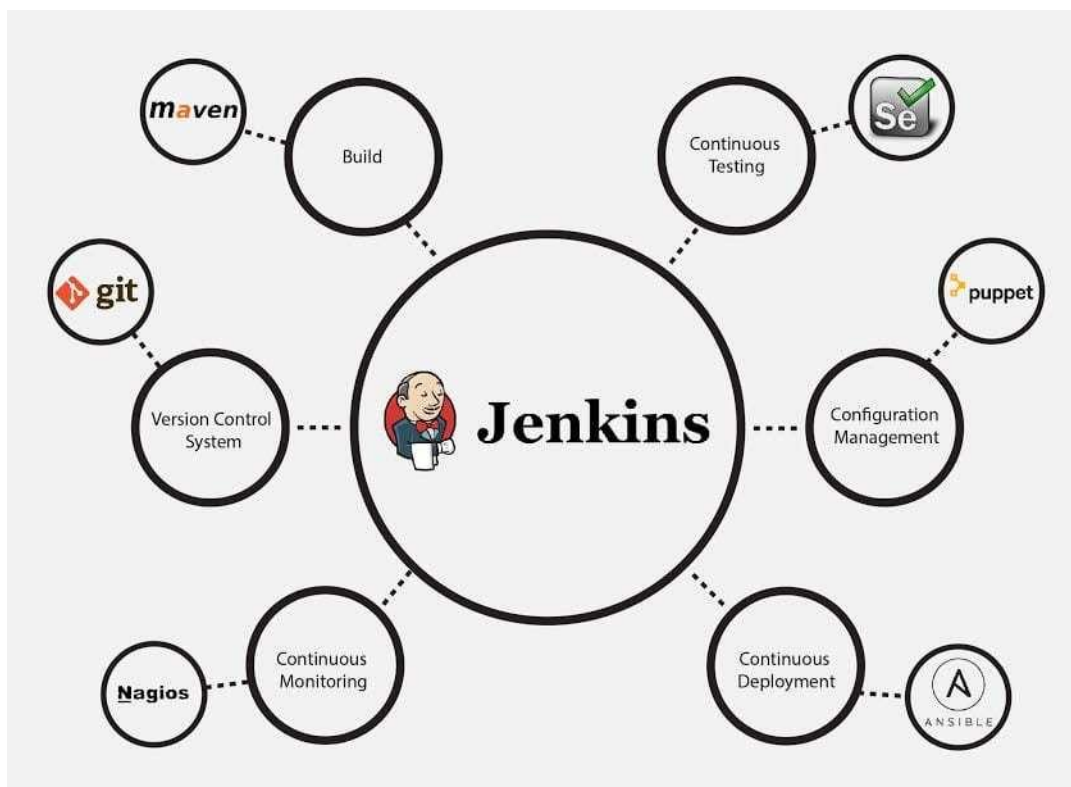
Experiment 1

Installation + Job Creation Jenkins

Jenkins is an **Opensource automation tool**, written in Java for the purpose of Continuous Integration. Jenkins is a *build tool* that makes easier for developers to build and test their projects continuously, making it easier for the users to obtain a fresh build. Jenkins provides various plugins for integrating with various testing and deployment technologies.

Jenkins accelerates the software development process using automation and integrates development life-cycle processes like, build, documentation, test, package, staging, deployment as well as static code analysis.

Jenkins achieves Continuous Integration with the help of plugins which are used to integrate various DevOps stages. If we want to integrate a particular tool, we have a plugin for that. We can integrate a wide range of tools like, Maven, Selenium, Git, Puppet, Nagios and Ansible being some of them.



Advantages of Jenkins:

- Opensource + great community
- Easy installation
- Support wide range of plugins
- Free of cost
- Highly portable

Before Jenkins

Entire Source code build and tested at the end so it is difficult and time-consuming to fix the bugs.

Developers have to do the whole process manually.

Developers need to wait for the test result.

After Jenkins

Every commit is built and tested so developers need to focus on a single build rather than checking the whole source code.

The whole pipeline is automated so no manual intervention required.

Test result is provided after every commit.

Installing Jenkins in Linux

Pre-Requisites

- 1) Java installed
- 2) Our machine configured with a non-root sudo user.

Installing Java8

- 1) First update the package indexes using the command,
`sudo apt update`
- 2) Install java using the following commands,
`sudo apt-get install default-jre`
`sudo apt-get install default-jdk`

In case we have multiple versions of Java installed we can configure them using the command

`sudo update-alternatives --config java`

Installing Jenkins

- 1) The version of Jenkins available with default Ubuntu is far behind the latest versions so we need to add the Jenkins repository to our system,
`wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -`
- 2) As the key is added, we need to append the package repo to ***sources.list***,
`sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'`
- 3) Update the apt repository to use the new repo,
`sudo apt update`
- 4) Install Jenkins and its dependencies,
`sudo apt install Jenkins`
- 5) We need to start the Jenkins services on our linux device and then we can check it's status using,
`sudo systemctl start jenkins`
`sudo systemctl status jenkins`
If we are using our Jenkins on our server machine we need to allow our devices to connect to it by allowing it in the firewall
`sudo ufw allow 8080`

- 6) For accessing our jenkins, we need to open the following URL,
`http://your_server_ip_or_domain:8080`
For local devices the IP used will be 127.0.0.1:8080

- 7) We need to unlock our Jenkins by accessing the following file,
`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

- 8) After unlocking the Jenkins, install the suggested plugins, create user

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<div><div>** Pipeline: Milestone Step</div><div>** JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI)</div><div>** Jackson 2 API</div><div>** JavaScript GUI Lib: ACE Editor bundle</div><div>** Pipeline: SCM Step</div><div>** Pipeline: Groovy</div><div>** Pipeline: Input Step</div><div>** Pipeline: Stage Step</div><div>** Pipeline: Job</div><div>** Pipeline Graph Analysis</div><div>** Pipeline: REST API</div><div>** JavaScript GUI Lib: Handlebars bundle</div><div>** JavaScript GUI Lib: Moment.js bundle</div><div>Pipeline: Stage View</div><div>** Pipeline: Build Step</div><div>** Pipeline: Model API</div><div>** Pipeline: Declarative Extension Points API</div><div>** Apache HttpComponents Client 4.x API</div><div>** JSch dependency</div></div>
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle	
🔗 Pipeline	🔗 GitHub Branch Source	🔗 Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View	
🔗 Git	🔗 Subversion	🔗 SSH Slaves	🔗 Matrix Authorization Strategy	
🔗 PAM Authentication	🔗 LDAP	🔗 Email Extension	🔗 Mailer	

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

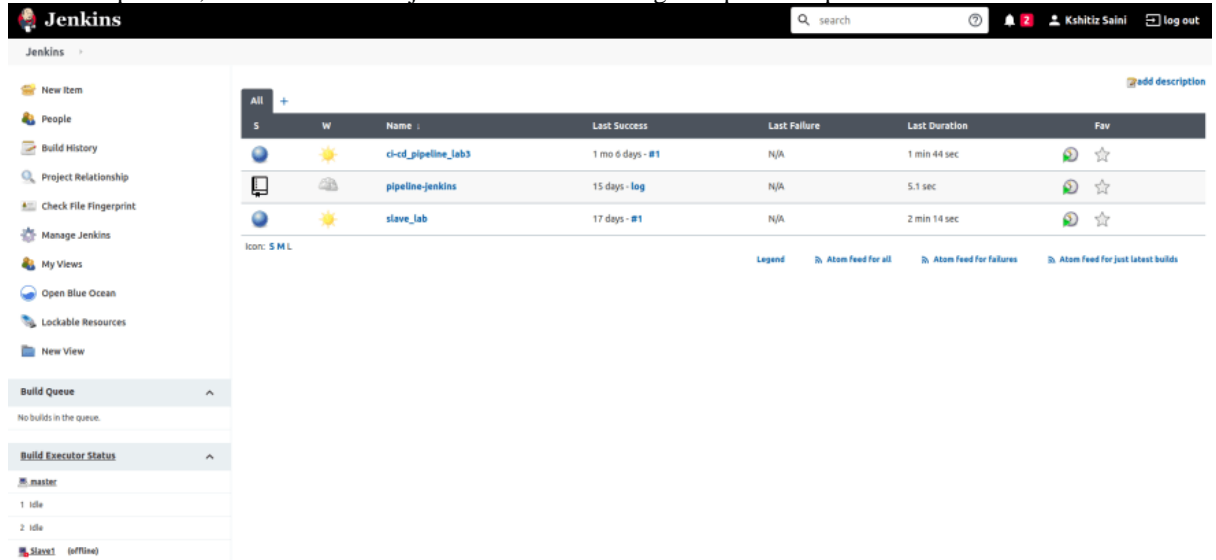
E-mail address:

Jenkins 2.121.1

Continue as admin

Save and Continue

- 9) After compilation, we can access our Jenkins dashboard using the specified ip

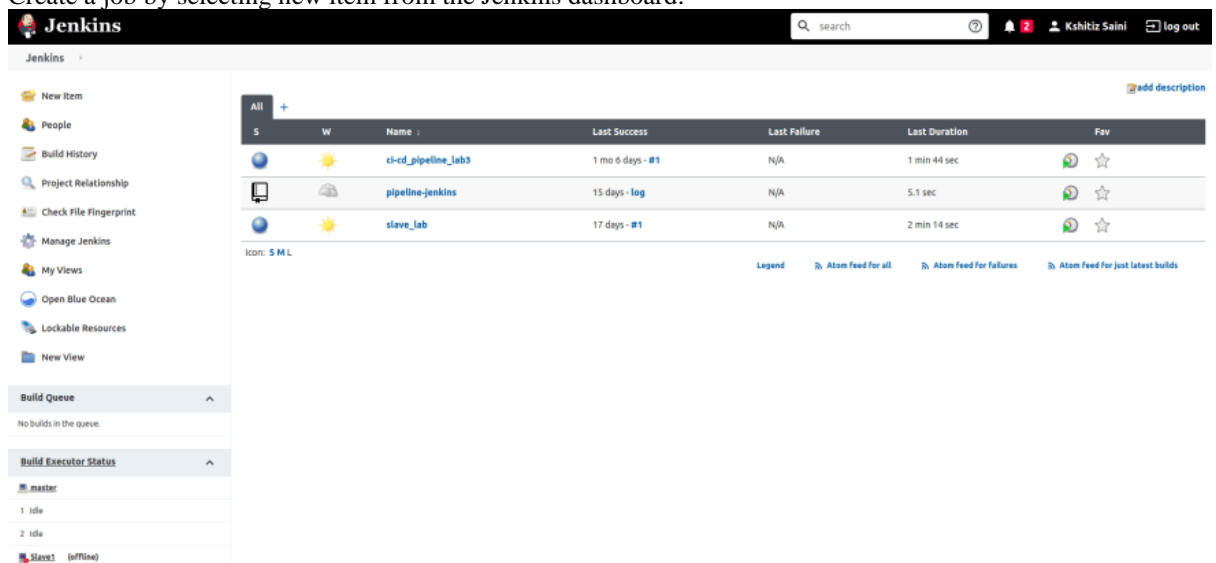


The screenshot shows the Jenkins dashboard with a sidebar on the left containing navigation links like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Lockable Resources', and 'New View'. The main area displays a table of jobs with columns for 'S' (Status), 'W' (Webhook), 'Name', 'Last Success', 'Last Failure', 'Last Duration', and 'Fav'. The table lists three jobs: 'ci-cd_pipeline_lab3', 'pipeline-jenkins', and 'slave_lab'. Below the table, there are links for 'Legend', 'Atom Feed for all', 'Atom Feed for failures', and 'Atom Feed for just latest builds'. On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 idle, 2 idle, 1 slave1 offline).

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		ci-cd_pipeline_lab3	1 mo 6 days - #1	N/A	1 min 44 sec	
		pipeline-jenkins	15 days - log	N/A	5.1 sec	
		slave_lab	17 days - #1	N/A	2 min 14 sec	

Creating first Jenkins Job

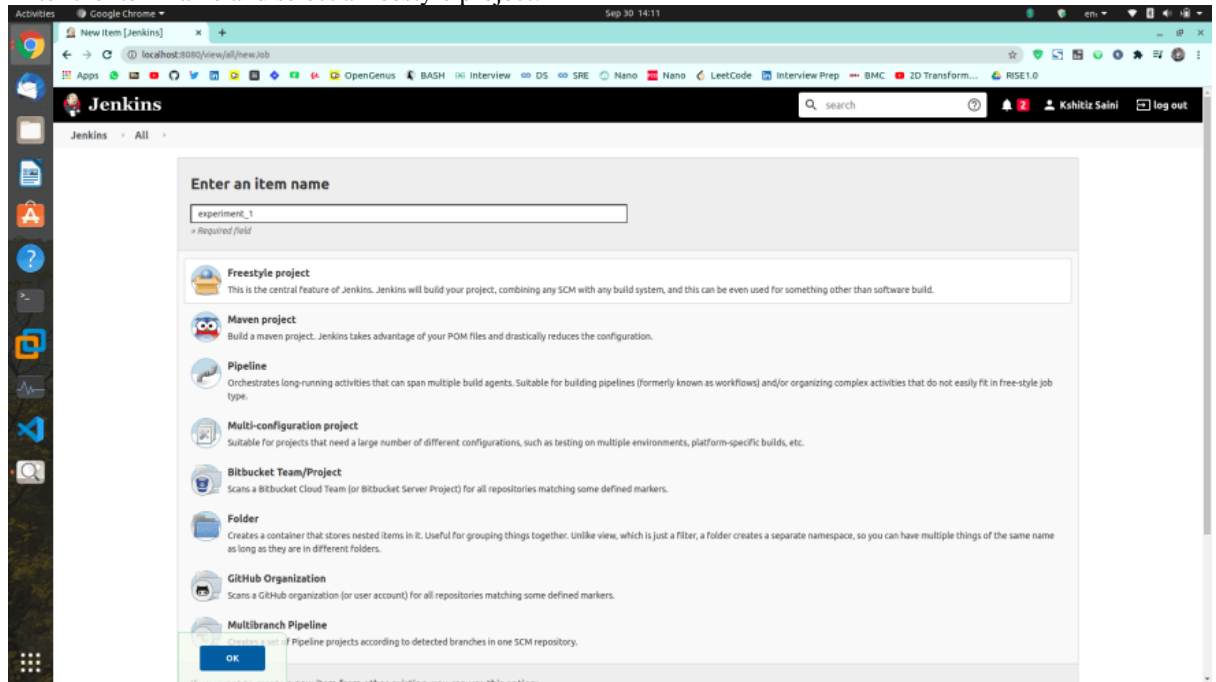
- 1) Create a job by selecting new item from the Jenkins dashboard.



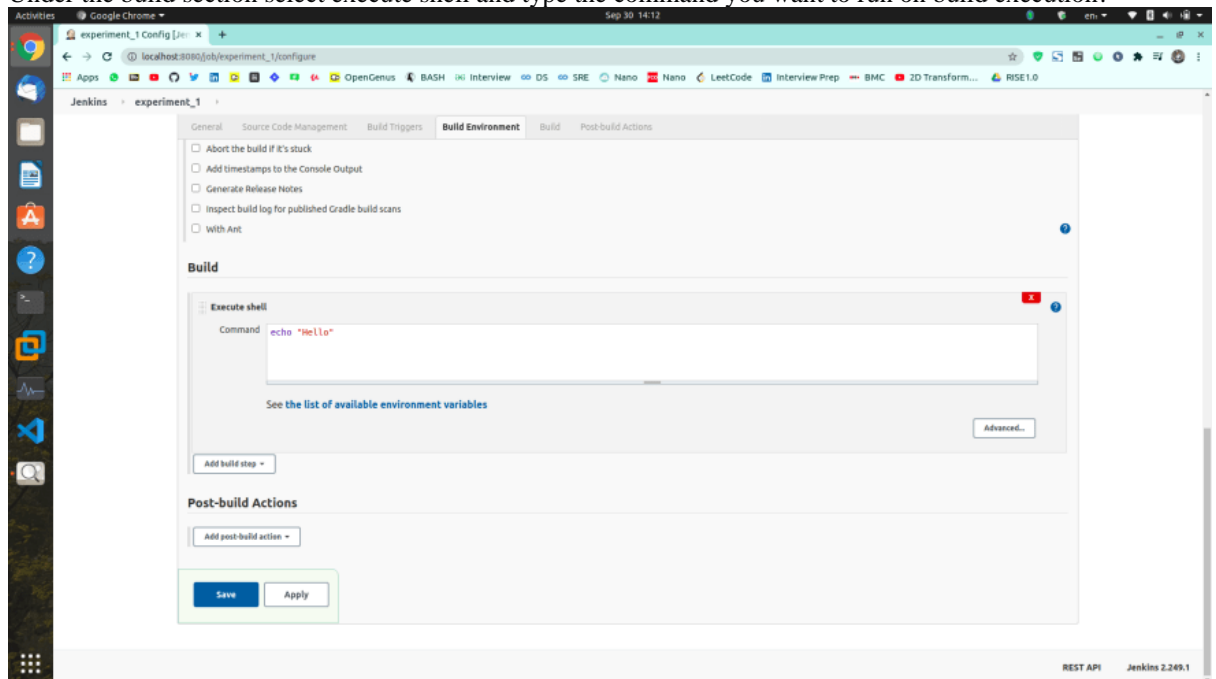
This screenshot is identical to the one above, showing the Jenkins dashboard with the same sidebar, job table, and build status sections.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		ci-cd_pipeline_lab3	1 mo 6 days - #1	N/A	1 min 44 sec	
		pipeline-jenkins	15 days - log	N/A	5.1 sec	
		slave_lab	17 days - #1	N/A	2 min 14 sec	

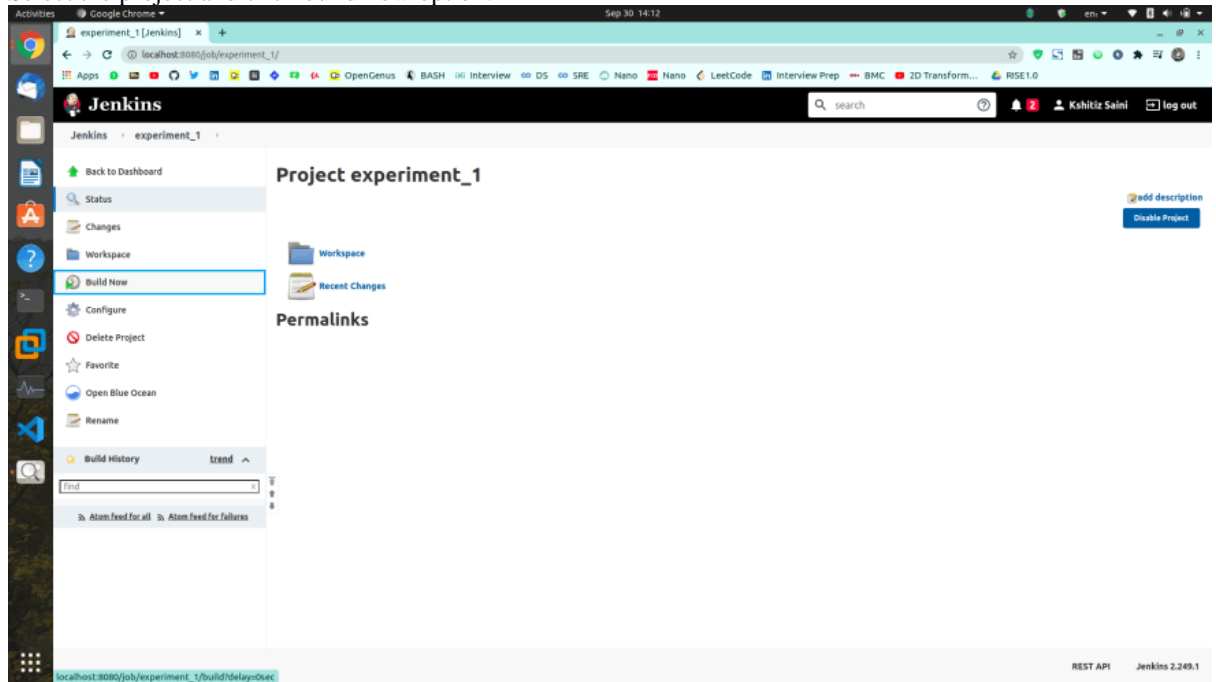
- 2) Enter the item name and select a freestyle project.



- 3) Under the build section select execute shell and type the command you want to run on build execution.



- 4) Select the project and click build now option.



- 5) See the output by clicking on build number and clicking Console Output.

