Jenkins Setup Documentation

Brief description of the steps I performed (including commands and screenshots). A specific screen shot that were asked in Delivery are marked.

Phase A

1. We opened a project folder through VS Code

2. We launched Docker

3. docker network jenkins-net :Create a network for Docker
   For the bonus - so that the agent knows how to communicate with the master. This creates a shared network where they recognize each other by name. Otherwise, the agent wouldn't recognize Jenkins and we would need to open this network at a later stage.

4. Run command to download Jenkins, open port, and name the container:

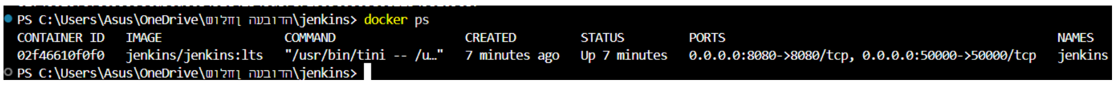
   docker run -d
   --name jenkins (Container name)
   -p 8080: 8080 (Opens the UI at http://localhost:8080)
   -p 50000: 50000 (Port for agents)
   -v jenkins_home: /var/jenkins_home (Creates volume named jenkins_home to save all settings)
   --network jenkins-net (Connects to the Docker network we created)
   jenkins/jenkins: lts (Official Jenkins image)

5. Port numbers for docker file:
   8080 - Jenkins Web interface (UI)
   50000 - Agent connections (JNLP)

6. jenkins/jenkins: lts - This is the official Image from Docker Hub containing Jenkins in LTS version.

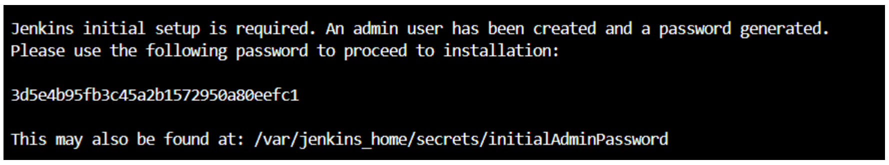
(The Container is essentially Jenkins running on the computer, with open ports (8080 etc.)).

7. For checking that everything works: docker ps - Shows a running Jenkins container



```
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins> docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED         STATUS         PORTS                                              NAMES
02f46610f0f0   jenkins/jenkins:lts   "/usr/bin/tini -- /u…"   7 minutes ago   Up 7 minutes   0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   jenkins
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins>
```

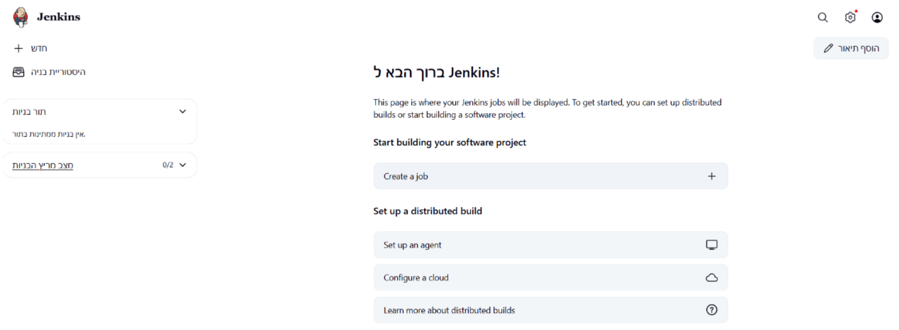
8. Initial password retrieved and plugins installed **Password: 1234**



```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

3d5e4b95fb3c45a2b1572950a80eefc1

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
```



Jenkins

ברוך הבא ל Jenkins!

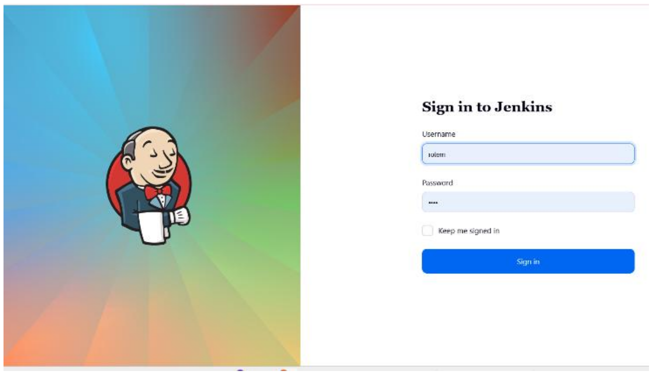
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

אין בניות ממתינות בתור.

9. Stopped the container: docker stop jenkins
10. Restarted: docker start jenkins
11. Reconnected again



Sign in to Jenkins

Username: rotem

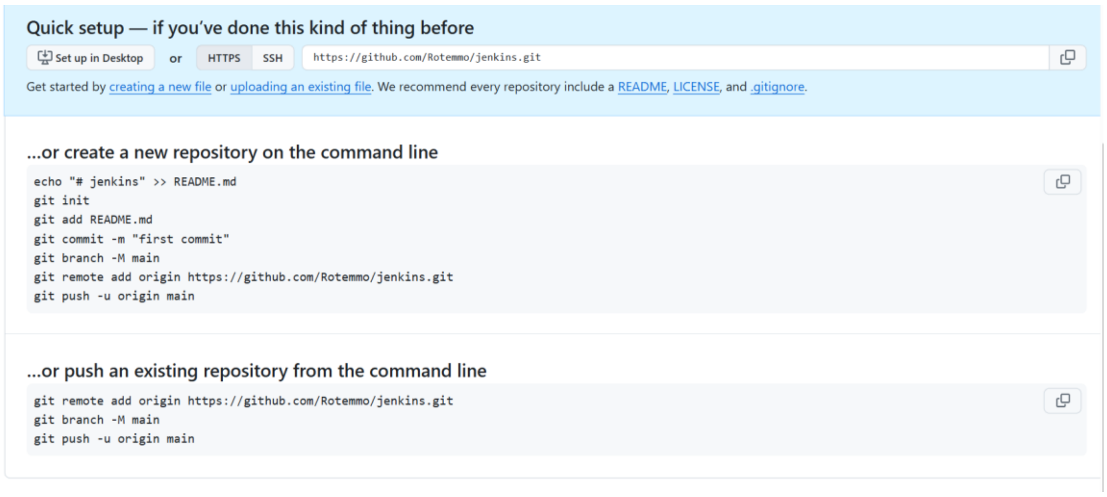
Password: ••••

☐ Keep me signed in

Phase B

1. For opening the repository, I preferred to do it through GitHub - this shouldn't matter.

   We'll use commands for a new repository.



Quick setup — if you've done this kind of thing before

```
https://github.com/Rotemmo/jenkins.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# jenkins" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Rotemmo/jenkins.git
git push -u origin main
```

…or push an existing repository from the command line

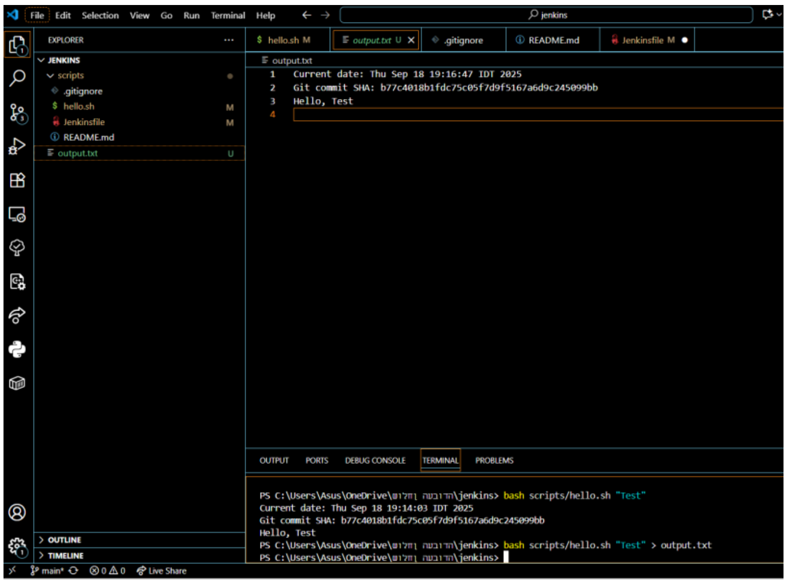
```
git remote add origin https://github.com/Rotemmo/jenkins.git
git branch -M main
git push -u origin main
```

2. For what was requested - scripts/hello.sh - **I added comments above each print statement** indicating which requirement it refers to.

   About there requirement 'Must be executable' , In Windows there's no issue with this (in Linux you need to run a command for this)

3. For what was requested about Jenkinsfile **- I added comments above each print statement** indicating which requirement it refers to.

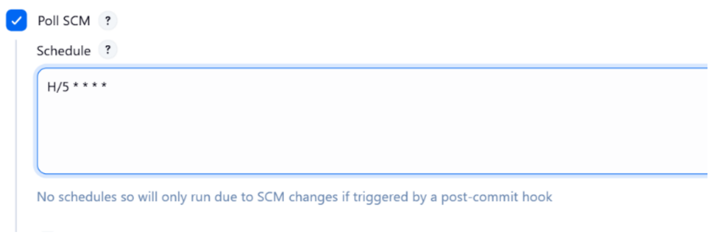
(End of Phase B - At this stage we don't see it in Jenkins UI yet)



```
Current date: Thu Sep 18 19:16:47 IDT 2025
Git commit SHA: b77c4018b1fdc75c05f7d9f5167a6d9c245099bb
Hello, Test
```

```
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins> bash scripts/hello.sh "Test"
Current date: Thu Sep 18 19:14:03 IDT 2025
Git commit SHA: b77c4018b1fdc75c05f7d9f5167a6d9c245099bb
Hello, Test
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins> bash scripts/hello.sh "Test" > output.txt
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins>
```

Phase C:



[✓] Poll SCM ?

Schedule ?

```
H/5 * * * *
```

No schedules so will only run due to SCM changes if triggered by a post-commit hook

Jenkins updates against Git to check for changes every 5 minutes.

I added the Git URL.

Regular execution (default)

```
Current date: Thu Sep 18 16:38:52 UTC 2025
Git commit SHA: 6f91bfe3a9a8c6cb1349d4e45cbe32affc36c622
Hello, world
```

Execution with parameter

```
Current date: Thu Sep 18 16:42:00 UTC 2025
Git commit SHA: 6f91bfe3a9a8c6cb1349d4e45cbe32affc36c622
Hello, Rotem
```

After about 5 minutes since we did the push: A build was automatically executed with the following output.

```
Current date: Thu Sep 18 16:51:26 UTC 2025
Git commit SHA: fc76bbf6958b1953cc9b097cf4e9a93f47f79fe2
Hello, World
```

Bonus:

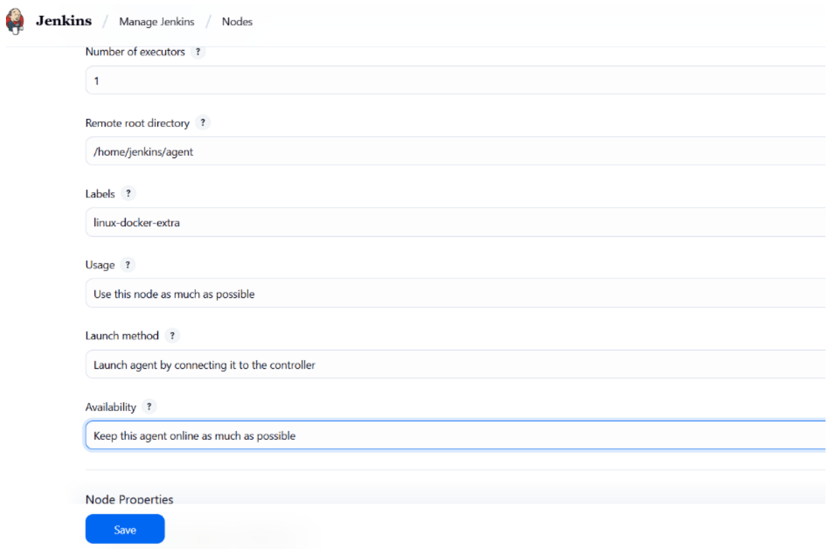
Explanation about what I did

Until now we used the controller (master). It is the **main Jenkins** that manages the Pipelines, receives triggers (for example, new commit), schedules jobs, saves logs and

Artifacts. It doesn't have to run the build itself and can manage everything. Everything we ran so far runs on the controller.

Agent is the worker - our node. It's essentially an additional computer or **container** to which the controller sends tasks. It's the one that actually performs the build - because there can be several of these, it essentially allows us to distribute loads and work in various work environments.

So now we'll create a new node according to the requirements:


Jenkins / Manage Jenkins / Nodes

Number of executors: 1

Remote root directory: /home/jenkins/agent

Labels: linux-docker-extra

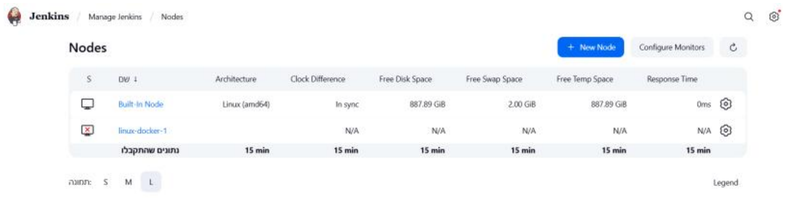
Usage: Use this node as much as possible

Launch method: Launch agent by connecting it to the controller

Availability: Keep this agent online as much as possible

Node Properties

Before activation - (shows offline node)


Jenkins / Manage Jenkins / Nodes

Nodes

S	שם ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥	Built-In Node	Linux (amd64)	In sync	887.89 GiB	2.00 GiB	887.89 GiB	0ms
☒	linux-docker-1		N/A	N/A	N/A	N/A	N/A
	נתונים שהתקבלו	15 min	15 min	15 min	15 min	15 min	15 min

Activating the new node:

Similar to the initial commands where we downloaded Jenkins: This time we'll connect the node to the network we created earlier:

docker run -d \

--name jenkins-agent (Create new container + gave it agent name)

--network jenkins-net (Connect the agent to Docker network so agent recognizes our controller)

-e JENKINS_URL=http://jenkins:8080 (Controller address so it knows where to connect)

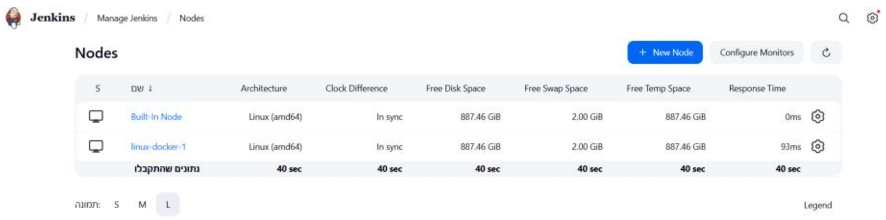
-e JENKINS_AGENT_NAME=linux-docker-1 (Its identifying name within Jenkins)

-e JENKINS_SECRET=... (The password we took from Jenkins)

-e JENKINS_AGENT_WORKDIR=/home/jenkins/agent (Agent's work directory) inside container (files from git will be downloaded here and executions will occur)

jenkins/agent:latest (Official Docker image of Jenkins Agent - this is a different)
Image from the controller so it downloads it anew

Now we can see in Jenkins UI that the node is active.

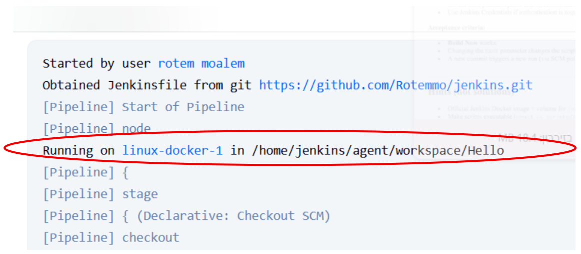

Jenkins / Manage Jenkins / Nodes

Nodes

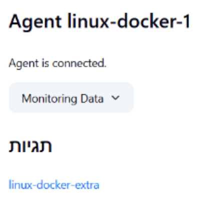
S	שם ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	887.46 GiB	2.00 GiB	887.46 GiB	0ms
	linux-docker-1	Linux (amd64)	In sync	887.46 GiB	2.00 GiB	887.46 GiB	93ms
	נתונים שהתקבלו	40 sec	40 sec	40 sec	40 sec	40 sec	40 sec

Now we'll update the Jenkinsfile so it runs only on the agent we created: label 'linux-docker-extra'

We run a build and verify in console output that it's running on the correct node.



```
Started by user rotem moalem
Obtained Jenkinsfile from git https://github.com/Rotemmo/jenkins.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on linux-docker-1 in /home/jenkins/agent/workspace/Hello
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
```


Agent linux-docker-1

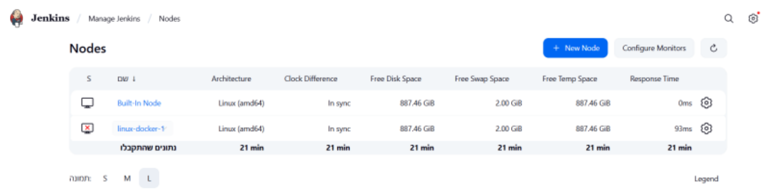
Agent is connected.

We can see it's the correct label as requested.

You can also see that the Output is saved as an artifact.

Now we'll run a command to stop the agent container: docker stop jenkins-agent

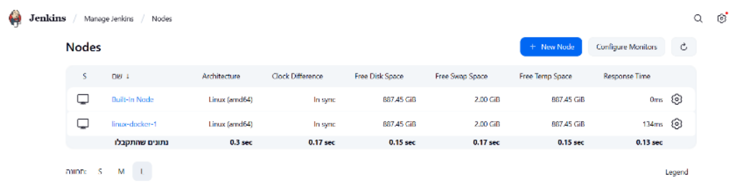
We'll verify it became offline.


Jenkins / Manage Jenkins / Nodes

Nodes

S	שם ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	887.46 GiB	2.00 GiB	887.46 GiB	0ms
	linux-docker-1	Linux (amd64)	In sync	887.46 GiB	2.00 GiB	887.46 GiB	93ms
	נתונים שהתקבלו	21 min	21 min	21 min	21 min	21 min	21 min

We'll restart: docker start jenkins-agent


Jenkins / Manage Jenkins / Nodes

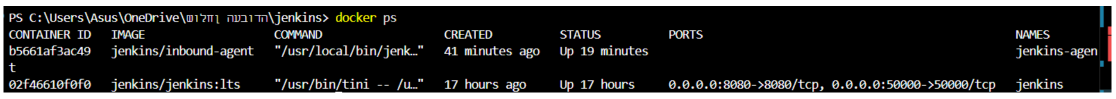
Nodes

S	שם ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	887.45 GiB	2.00 GiB	887.45 GiB	0ms
	linux-docker-1	Linux (amd64)	In sync	887.45 GiB	2.00 GiB	887.45 GiB	134ms
	נתונים שהתקבלו	0.3 sec	0.17 sec	0.15 sec	0.17 sec	0.15 sec	0.13 sec

We'll run build again with parameter "test" and see that the Build passed successfully.

```
Current date: Fri Sep 19 08:10:40 UTC 2025
Git commit SHA: fc76bbf6958b1953cc9b097cf4e9a93f47f79fe2
Hello, test
```

<u>Screenshot of docker ps - for section 7 in Deliverables</u>



```
PS C:\Users\Asus\OneDrive\שולחן העבודה\jenkins> docker ps
CONTAINER ID   IMAGE                   COMMAND                  CREATED          STATUS          PORTS                                              NAMES
b5661af3ac49   jenkins/inbound-agent   "/usr/local/bin/jenk…"   41 minutes ago   Up 19 minutes                                                      jenkins-agen
t
02f46610f0f0   jenkins/jenkins:lts     "/usr/bin/tini -- /u…"   17 hours ago     Up 17 hours     0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   jenkins
```