Jenkins Freestyle Project

Documentation - Chaya Lipshitz

# Step by step:

## Create a github repository:

In your [github account](https://github.com/ChayaLipshitz) create a repo called: **freestyle\_CI**

## Link my nodejs task management app to your repo and push all of its content.

Clone the new repo to vscode, copy the app to the folder of the repo and push the project to github. (All by the UI)

## Launch the jenkins server - in the container.

In [docker-hub](https://hub.docker.com/), Search for Jenkins images. In the tags tab, filter the LTS (Long Term Support) versions - Best practice of choosing image.

In the first image (newest) that is shown, click on the ”documentation” - in order to see the usage (with volumes) of this image.

The final command to use is:

docker run -p 8080:8080 --restart=on-failure --name jenkins -v C:\jenkins\_vol:/var/jenkins\_home jenkins/jenkins:lts-jdk17

The volume path: C:\jenkins\_vol - is the local path where I put the plugins that I want from the [drive](https://drive.google.com/file/d/19fDs-kgFaJAuGK945qjbKAW_KyjVX0GO/view).

I omitted the -p 50000:50000 from the original command because we don’t use agents yet.

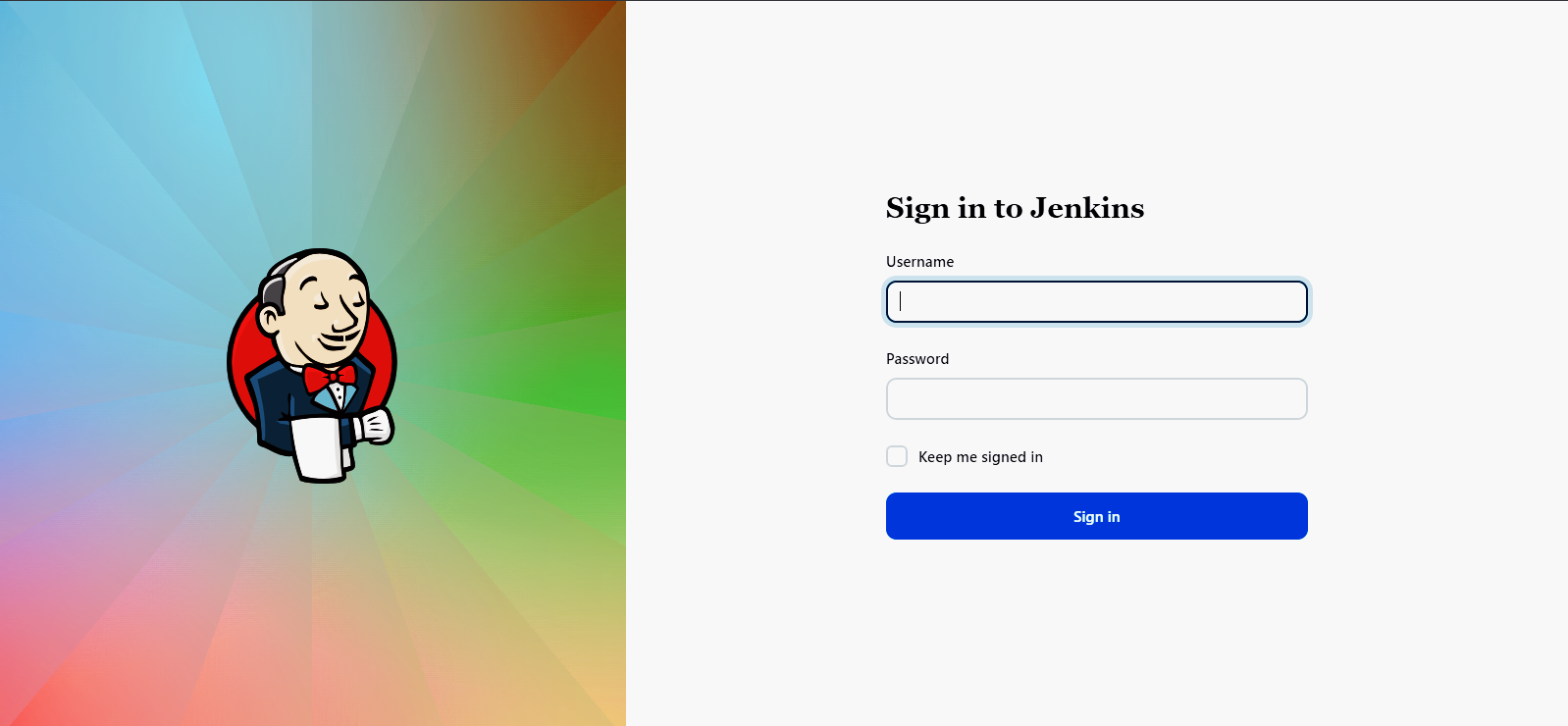
The command should take a long time. After that you should see in the console (or in

docker logs jenkins

command ) the following output:

hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running

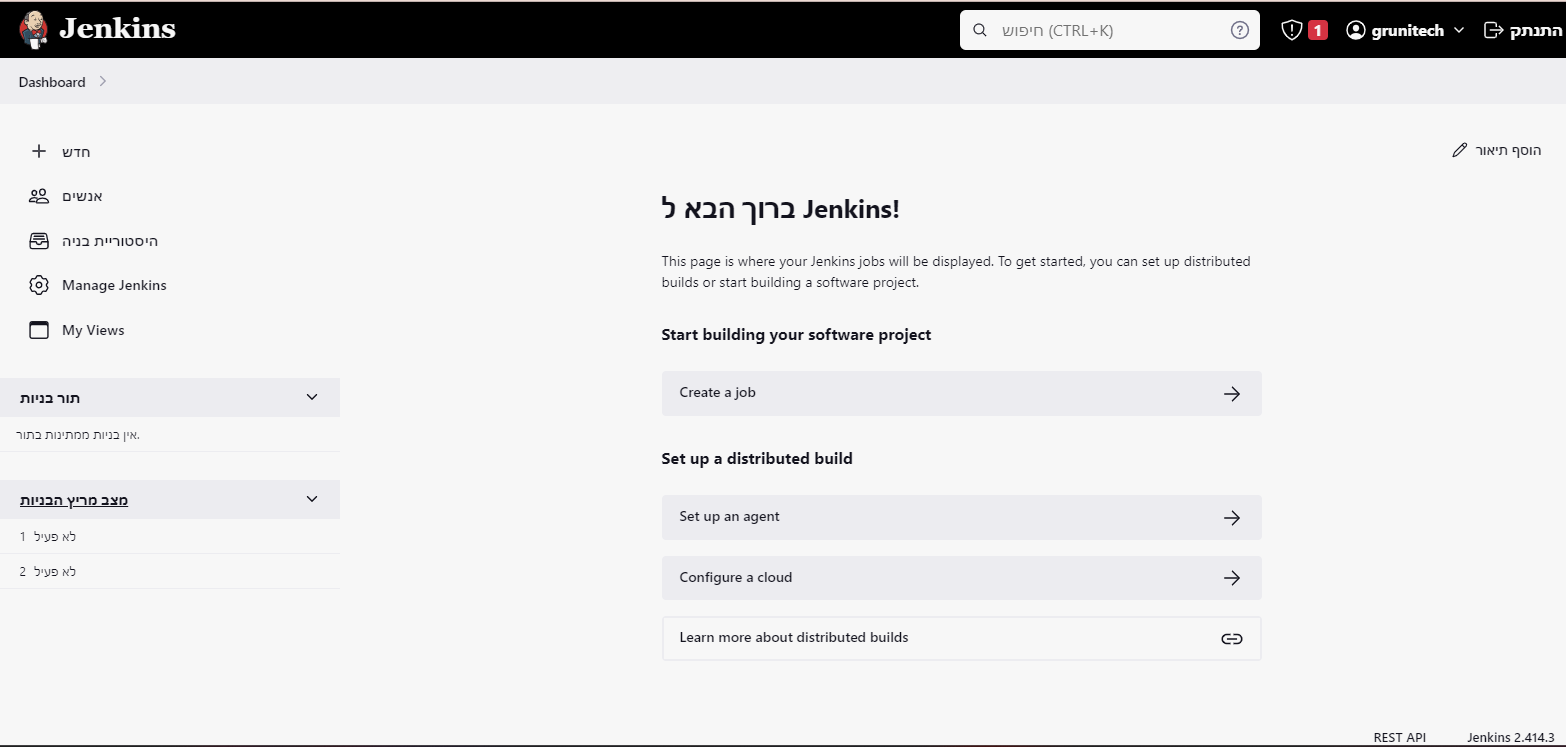
In localhost:8080 you should see:



Enter username: gruni

And password: 1234

Now you should see the dashboard of jenkins - home page:



You successfully run the Jenkins container!!

1. 4 & 5

## Manage a ssh key and create project in jenkins:

**In the terminal:**

Enter to my Jenkins container (called “jenkins”):

docker exec -it jenkins bash

Then, in the jenkins server, generate a ssh key by the following command:

ssh-keygen -t rsa -b 4096 -C "[chl4165772@gmail.com](mailto:chl4165772@gmail.com)"

Click enter in every input required.

Now we have 2 file in the jenkins container:

The private key is in the file id\_rsa in path: /var/jenkins\_home/.ssh/id\_rsa.

The public key is in the file id\_rsa.pub in path: /var/jenkins\_home/.ssh/id\_rsa.pub.

**In github:**

You can [Manage ssh key](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent) but I recommend to [manage a - deploy key to my specific repository](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/managing-deploy-keys) and enter the output of /var/jenkins\_home/.ssh/id\_rsa.pub file.

**In jenkins:**

* **Add credential:**

Go to Managed system > Credentials And click on the **Global** Under the title Stores scoped to Jenkins and then click **Add credential.**

Choose a kind: SSH Username with a private key.

Choose ID, description, and enter your jenkins username, choose the button enter directly and then enter the content of the /var/jenkins\_home/.ssh/id\_rsa file.

Click **Add.**

* **Create project:**

In the jenkins dashboard, click **create a job.**

Choose type: **free style project.**

Click **OK.**

In the configuration page:

In the **Source code management** section, choose **GIT.**

Then enter your URL of the project (in ssh format). And choose the credential you had created.

My problems had been solved by using the following references:

[128 error status](https://stackoverflow.com/questions/21557998/jenkins-failed-to-connect-to-repository) - I used the 1st and the 2nd answers and they helped me.

[Solution for:ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.](https://stackoverflow.com/questions/23906352/git-pullrequest-job-failed-couldnt-find-any-revision-to-build-verify-the-repo)

[Finished: FAILURE](https://stackoverflow.com/questions/23906352/git-pullrequest-job-failed-couldnt-find-any-revision-to-build-verify-the-repo).

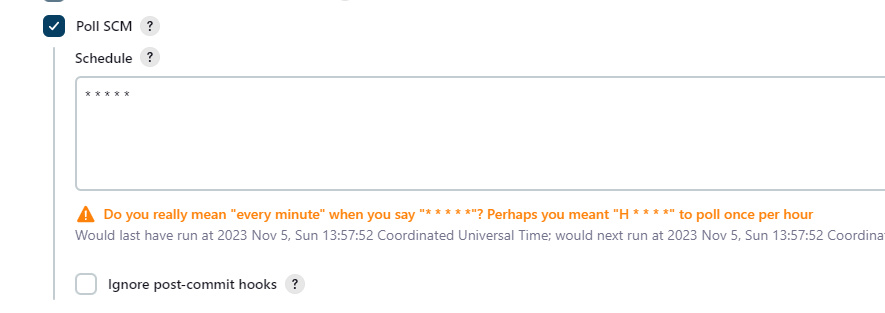
## creating a CI according to the instructions:

In the job configure page:

In the build section check the **Poll SCM** option. In the **schedule** enter the input of checking every minute:

\* \* \* \* \*

Screenshot:



In the build steps section, click “add build step” and choose Execute shell.

Add the command: **echo “hello this is my first CI!”**

## Delete workspace when done.

In the Build Environment section, choose the option:

*Delete workspace before build starts.*

Click **save.**

## Run a build

In the main page of the job, click on **build now.** And check if the building completed successfully.

## Push commit

In the vscode, modified the dockerfile and pushed the changes to the github.

Now the building of the image should be done automatically by jenkins within a minute at most.

## Add to the CI building and running the application

Enter into the configure page of the job,

In the build steps section, click “add build step” and choose Execute shell.

Add the command: **docker-compose up**

Click **save**

Return to the main job-page, and click **build now**

**HO —-- there is a problem!!**

**The build finished with failure!**

In the build logs, you should see the **ERROR: ‘docker’ - command not found**

### How to use docker commands into the jenkins build steps?

**According to** [**this reference**](https://devopscube.com/run-docker-in-docker/) **about docker in docker:**

We have to change some settings:

In the docker run command, which runs the jenkins container, we have to add the volume which enable to use **docker in docker:**

**-v /var/run/docker.sock:/var/run/docker.sock**

In addition: we have to install the docker CLI in order to run docker commands so we want to run the container as root user:

**-u 0**

***Alternatively run commands from the container after deploying***, I prefer to use a dockerfile, and define some settings in it as the following:

FROM jenkins/jenkins:lts-jdk17

USER root

RUN apt-get -q update && \

apt-get -q install -y docker.io

**Now, we have to build the jenkins image from the dockerfile:**

**docker build -t jenkins .**

(In the run command, we still have include the volume

**-v /var/run/docker.sock:/var/run/docker.sock**

In order to enable docker in docker.)

**The run command:**

**docker run -p 8080:8080 --restart=on-failure --name jenkins -v C:\jenkins\_vol:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock jenkins**

Now you succeed to use docker in jenkins freestyle, and to run an application from jenkins!!!

## 