












# Jenkins Best Docker Configuration

 Author	 Santiago Garcia Arango
 Course Name	J04. Jenkins Docker Configure
 Created	@Feb 25, 2021 11:35 PM
 Done	
 Link	
 Platform	me
 Topic	Jenkins

[Configure and Run Jenkins From Docker-Compose File \(Best Practice\)](#)

[Requirements](#)

[Setup Jenkins Step-by-Step](#)

# Configure and Run Jenkins From Docker-Compose File (Best Practice)

After this tutorial, we will be able to initialize a docker container that runs Jenkins, but from a personalized docker-compose file.



This custom way of working with Jenkins allows us to have a better control of the Jenkins container, avoid manual console tasks and be able to save the container in a docker-hub (or similar) cloud hosting.

## Requirements

- Medium understanding of Docker.
- Have Docker installed in the machine.
- Knowing how a "docker-compose.yml" file works.

## Setup Jenkins Step-by-Step

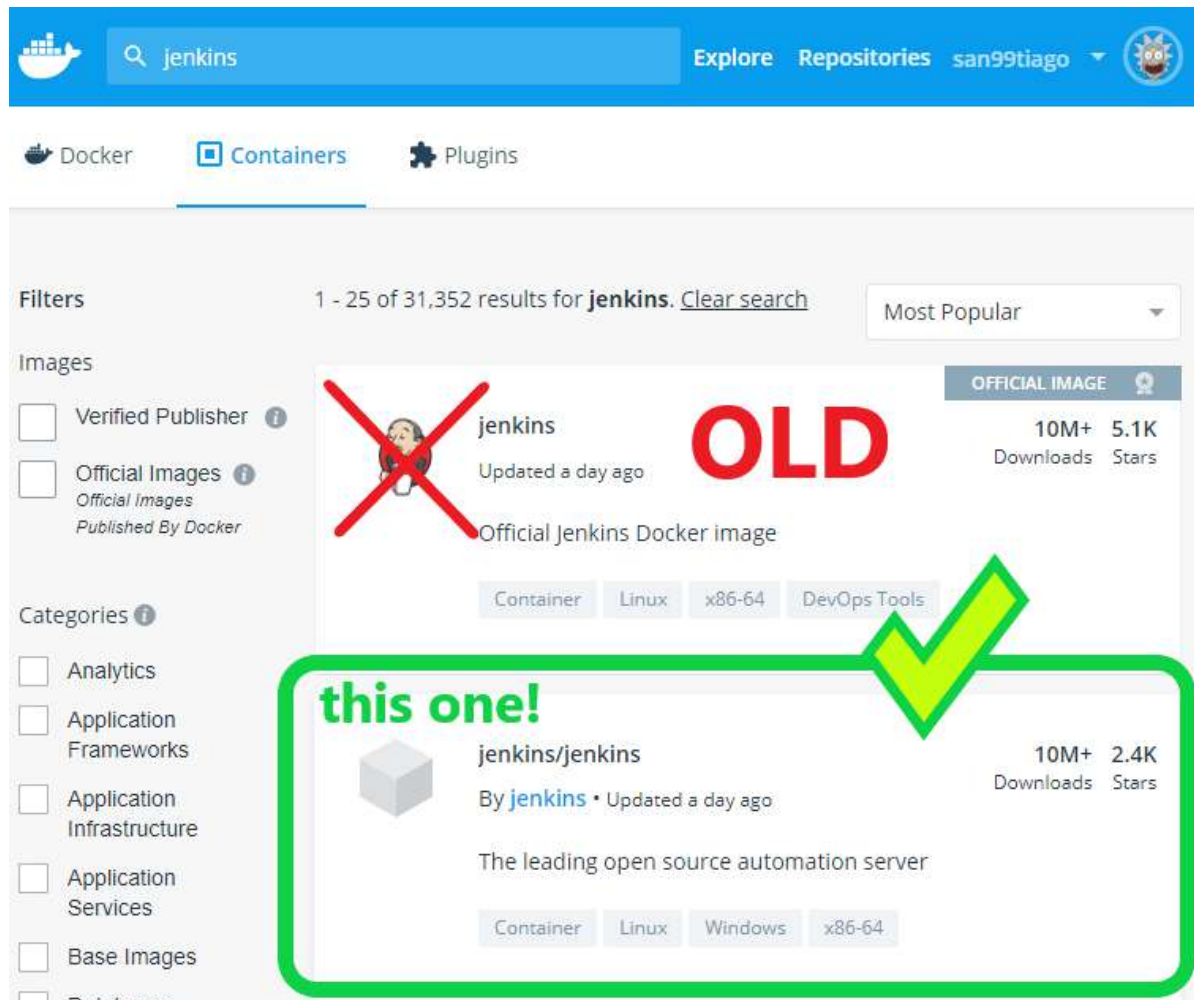
Docker Hub



<https://hub.docker.com/r/jenkins/jenkins>

We will dive into the step-by-step tutorial for the correct Jenkins install:

1. Run local docker platform/engine.
2. Go to <https://hub.docker.com/> and search for the shown Jenkins image:




Jenkins image is not the official (because it has been deprecated and not maintained), so to have the best features, we will choose the second one.


3. As shown before, we will now run our docker-compose.yml file (see repository for details).

Before running the command for creating the container for Jenkins, make sure that we are in the root of the file "docker-compose.yml".

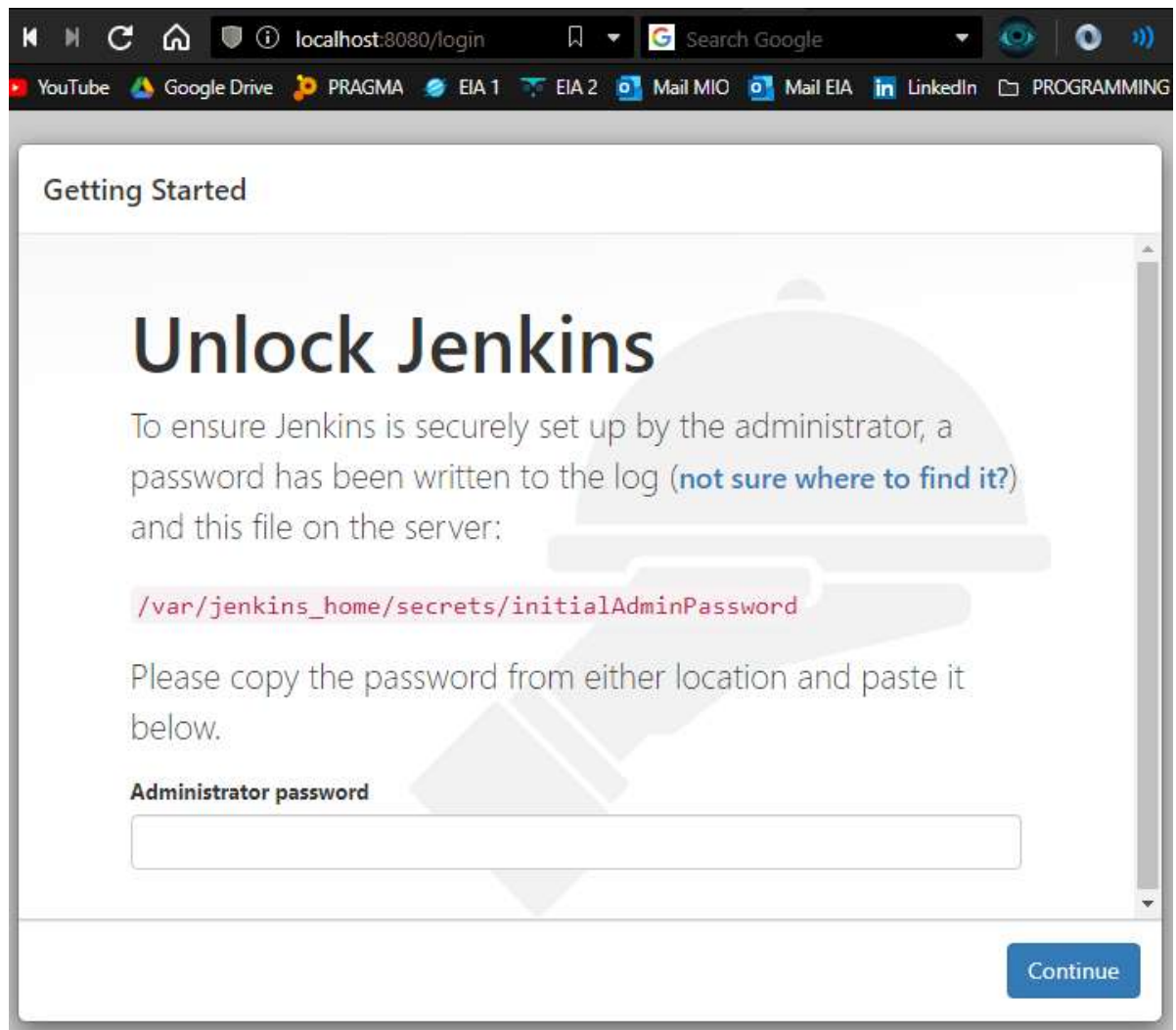
```
docker-compose up
```

docker-compose remarks:

 There are two main ways to connect the Jenkins' Docker container. With "Host-Volumes" and with "Named-Volumes". The only one that is shown in the "docker-compose.yml" is a "Named-Volume" approach.

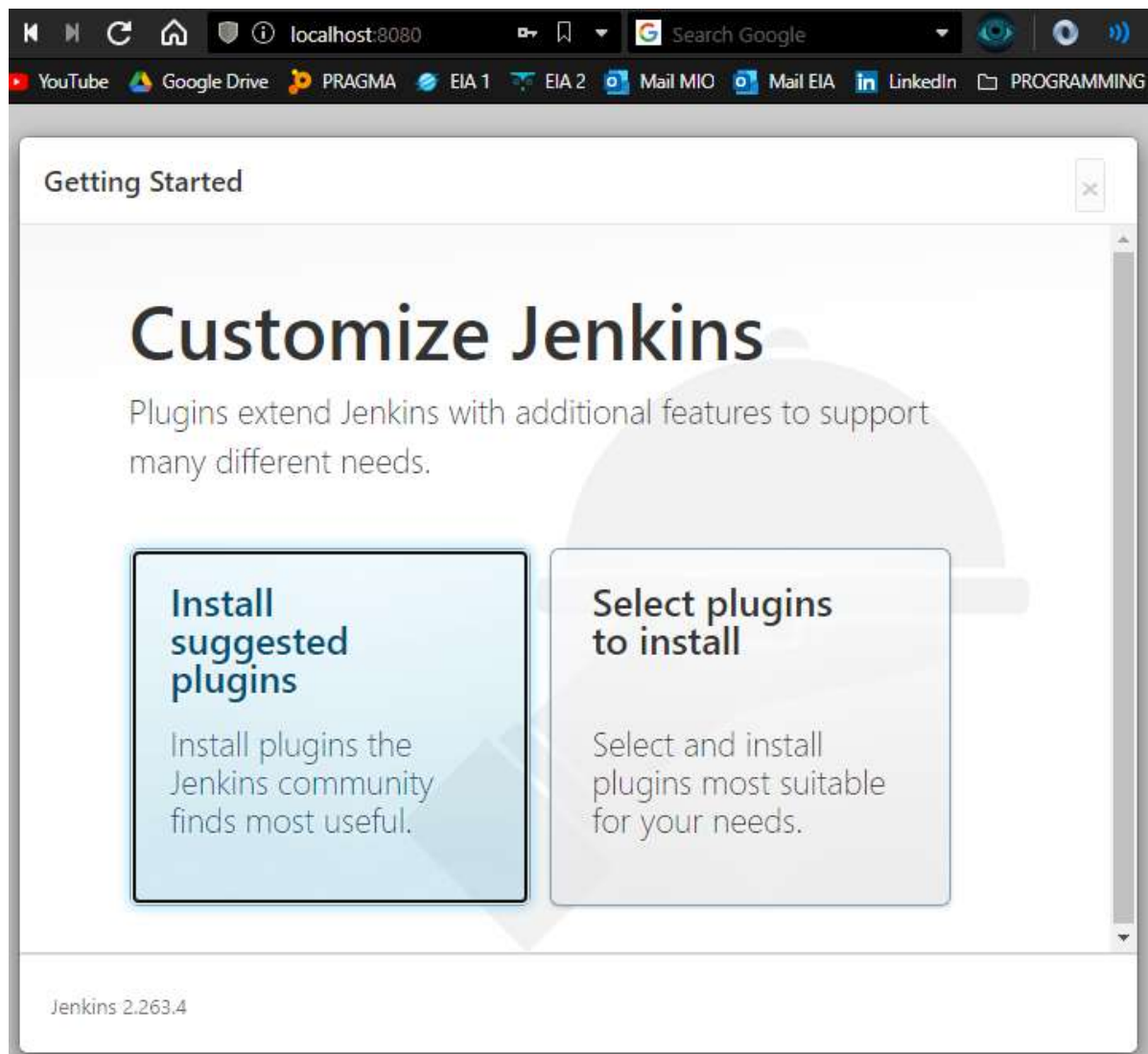
 To use a "Host-Volume", we would only have to change the "volumes: ..." in the "docker-compose.yml" to the specific path in the host.

4. After running the container, we can now enter localhost:8080 and login with the password generated at the moment of the creating docker's container.



⚠ If we don't see the password, we can access "bash" in the running container and search for it in the shown path.

5. Initialize Jenkins with the suggested plugins (recommended option).



6. It will take a while, but is the best choice for a good Jenkins setup.

Getting Started

# Getting Started

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	Folders ** Trilead API
Timestampers	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

7. Create a Jenkins account with the specific required information.

localhost:8080 Search Google

YouTube Google Drive PRAGMA EIA 1 EIA 2 Mail MIO Mail EIA LinkedIn

## Getting Started

# Create First Admin User

Username:

Password:

Confirm password:

Full name:

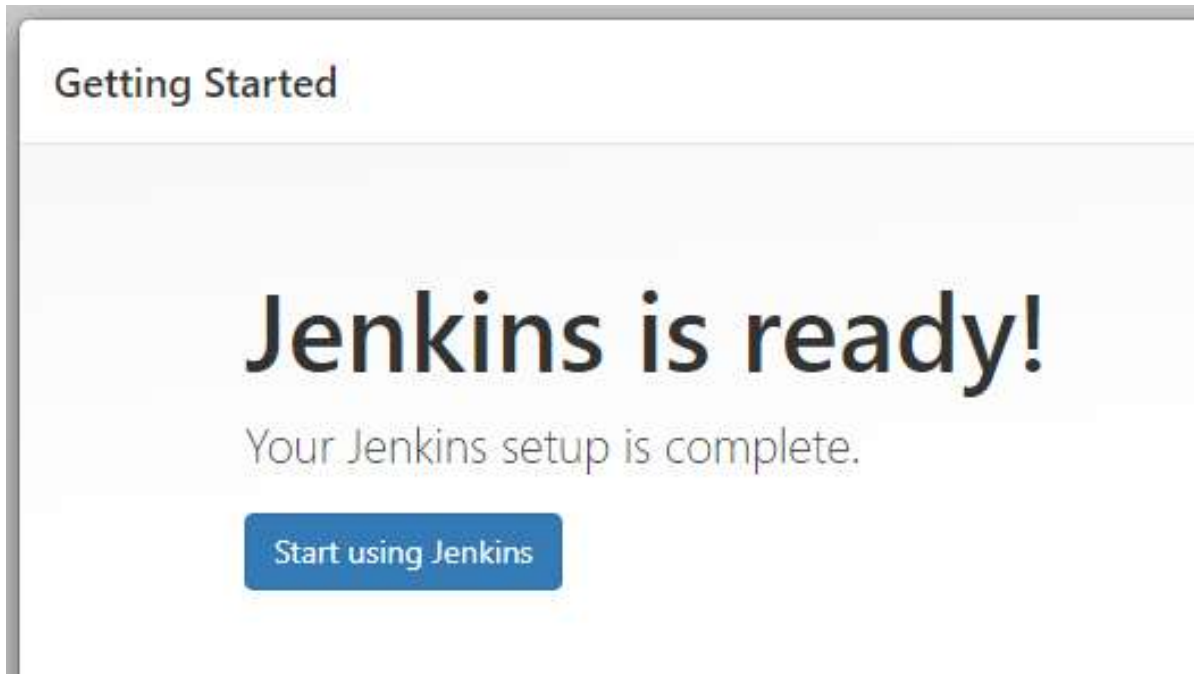
E-mail address:

Jenkins 2.263.4

[Skip and continue as admin](#) [Save and Continue](#)

8. Jenkins is ready!





9. Now we have Jenkins configured from an amazing custom docker-container that runs it.

