

Instalación y configuración de Jenkins en linux



Jesús Joel Meneses Meneses
2º DAW A
DPL—Despliegue de Aplicaciones Web



Índice

1. Requisitos básicos

2. Creación del dominio

3. Instalación de Jenkins en Docker

4. Instalación de Jenkins a través de Docker y Docker-Compose

- ### 4.1 Dockerfile ###
 - ### 4.2 Plugins.txt ###
 - ### 4.3 Docker-Compose###
 - ### 4.4 Construcción de la imagen###
-

1. Requisitos básicos

- Tener docker instalado
- Tener docker-compose instalado

2. Creación del dominio

Nos situamos en la carpeta de apache sites-available

```
joel@joel-joel:~$ locate sites-available
/etc/apache2/sites-available
```

Copiamos la carpeta de la practica anterior *joelic.conf* y en su lugar creamos jenkins.joel.conf

```
joel@joel-joel:/var/www$ locate sites-available
/etc/apache2/sites-available
/etc/apache2/sites-available/000-default.conf
/etc/apache2/sites-available/default-ssl.conf
/etc/apache2/sites-available/joelic.conf
/etc/apache2/sites-available/joelmmsytem.conf
/etc/nginx/sites-available
/etc/nginx/sites-available/default
joel@joel-joel:/var/www$ cd /etc/apache2/sites-available
joel@joel-joel:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  joelic.conf  joelmmsytem.conf
joel@joel-joel:/etc/apache2/sites-available$ sudo cp joelic.conf  jenkins.joel.c
onf
```

Configuramos la carpeta nueva

```
joel@joel-joel:/etc/apache2/sites-available$ sudo nano jenkins.joel.conf
joel@joel-joel:/etc/apache2/sites-available$ cat jenkins.joel.conf
<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.jenkins.joel.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/jenkins.joel

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
joel@joel-joel:/etc/apache2/sites-available$
```

Habilitamos el sitio

```
joel@joel-joel:/etc/apache2/sites-available$ sudo a2ensite jenkins.joel.conf
Enabling site jenkins.joel.
To activate the new configuration, you need to run:
systemctl reload apache2
```

Reiniciamos apache

```
joel@joel-joel:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

Añadimos el dominio en /etc/hosts

```
joel@joel-joel:/etc/apache2/sites-available$ sudo nano /etc/hosts
joel@joel-joel:/etc/apache2/sites-available$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 joel-joel
127.0.0.1 www.joelMM.com
127.0.0.1 www.joelmmsystem.com
127.0.1.1 www.joelic.com
127.0.1.1 www.jenkins.joel.com

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

3. Instalación de Jenkins en Docker

Nos dirigimos a <https://hub.docker.com/> y buscamos la imagen oficial de jenkins

Descargamos la imagen de jenkins para docker con el siguiente comando...

```
docker pull jenkins/jenkins:lts
```

```
joel@joel-joel:~$ docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
0e29546d541c: Already exists
11423c93b49c: Already exists
0a7e49db0973: Already exists
977259ae4ac6: Already exists
664d083fcd2d: Already exists
8a20d9bdf720: Already exists
cc94036a9149: Already exists
771e489c9ff6: Already exists
11d0bec8f08b: Pull complete
ca11fa5d5337: Pull complete
b465e025b587: Pull complete
e0e171e8e73c: Pull complete
969a814948b7: Pull complete
1a83889b1424: Pull complete
732f8876aa9d: Pull complete
a36d15f157a4: Pull complete
88bc217ad3f7: Pull complete
Digest: sha256:983890f538ff42b2d153e0b5f45d97e750e80f08238ddb2bcf5c5dc4a606e77f
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
```

Verificamos que la imagen se ha descargado correctamente con el siguiente comando...

```
docker images
```

```
joel@joel-joel:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
jenkins/jenkins	latest	b3febfc71dcd	6 days ago	442MB
jenkins/jenkins	lts	9aee0d53624f	6 days ago	441MB
wildfly_1	latest	8875580c7dd7	8 days ago	736MB
atmoz/sftp	latest	55a21bac82eb	4 weeks ago	154MB
recursos_www	latest	5d4ffc1fea52	5 weeks ago	419MB
ridel_saavedra-dpl_www	latest	5d4ffc1fea52	5 weeks ago	419MB
<none>	<none>	af10511f756d	5 weeks ago	736MB
mysql	8.0	bbf6571db497	6 weeks ago	516MB
jboss/wildfly	latest	856694040847	3 months ago	736MB
phpmyadmin/phpmyadmin	latest	2e5141bbcbfb	7 months ago	474MB
php	8.0.0-apache	e97a1581dd75	13 months ago	417MB

Ahora ejecutamos jenkins como contenedor Docker exponiendolo en el puerto 8080 del ordenador local. Verificamos nuestro /your/home:/ con el comando

```
docker run -p 8080:8080 -p 50000:50000 -v /your/home:/var/jenkins_home jenkins/jenkins:lts
```

```
joel@joel-joel:~$ docker run -p 8080:8080 -p 50000:50000 -v /home/joel:/var/www/jenkinsjoel jenkins/jenkins:lts
docker: Error response from daemon: driver failed programming external connectivity on endpoint romantic_swanson (2077cdb385e9092633881abccb9aff362b4e3cd59bac2bd59d5a80efbc922eed): Error starting userland proxy: listen tcp4 0.0.0.0:8080: bind: address already in use.
ERROR[0001] error waiting for container: context canceled
```

Si nos muestra el mensaje de error anterior hacemos lo siguiente

Listamos los puertos activos y vemos quien los esta ocupando con el siguiente comando de la imagen y luego eliminamos el servicio de ese puerto

```
joel@joel-joel:~$ sudo netstat -tunlp
Conexiones activas de Internet (solo servidores)
Proto Recib Envíad Dirección local Dirección remota Estado PID/Program name
tcp 0 0 0.0.0.0:8082 0.0.0.0:* ESCUCHAR 1173/java
tcp 0 0 127.0.0.53:53 0.0.0.0:* ESCUCHAR 640/systemd-resolve
tcp 0 0 0.0.0.0:22 0.0.0.0:* ESCUCHAR 896/sshd: /usr/sbin
tcp 0 0 127.0.0.1:631 0.0.0.0:* ESCUCHAR 677/cupsd
tcp 0 0 0.0.0.0:8443 0.0.0.0:* ESCUCHAR 1173/java
tcp 0 0 127.0.0.1:33060 0.0.0.0:* ESCUCHAR 1106/mysqld
tcp 0 0 127.0.0.1:9990 0.0.0.0:* ESCUCHAR 1173/java
tcp 0 0 127.0.0.1:3306 0.0.0.0:* ESCUCHAR 1106/mysqld
tcp6 0 0 :::8080 :::* ESCUCHAR 1177/java
tcp6 0 0 :::80 :::* ESCUCHAR 1029/apache2
tcp6 0 0 :::21 :::* ESCUCHAR 813/vsftpd
tcp6 0 0 :::22 :::* ESCUCHAR 896/sshd: /usr/sbin
tcp6 0 0 :::631 :::* ESCUCHAR 677/cupsd
udp 0 0 0.0.0.0:52840 0.0.0.0:* 6325/firefox
udp 0 0 127.0.0.53:53 0.0.0.0:* 640/systemd-resolve
udp 0 0 0.0.0.0:53352 0.0.0.0:* 675/avahi-daemon: r
udp 0 0 0.0.0.0:631 0.0.0.0:* 786/cups-browsed
udp 0 0 0.0.0.0:5353 0.0.0.0:* 675/avahi-daemon: r
udp6 0 0 :::48976 :::* 675/avahi-daemon: r
udp6 0 0 :::5353 :::* 675/avahi-daemon: r
joel@joel-joel:~$ sudo kill -9 1177
```

Volvemos a lanzar el comando anterior

```
joel@joel-joel:~$ docker run -p 8080:8080 -p 50000:50000 -v /home/joel:/var/www/jenkinsjoel jenkins/jenkins:lts
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
2022-01-18 19:37:52.882+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @1457ms to org.e
clipse.jetty.util.log.JavaUtilLog
2022-01-18 19:37:53.205+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2022-01-18 19:37:55.706+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2022-01-18 19:37:55.950+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-9.4.43.v20210629; built: 2021-06-
30T11:07:22.254Z; git: 526006ecfa3af7f1a27ef3a288e2bef7ea9dd7e8; jvm 11.0.13+8
```

Se genera una contraseña que debemos copiar

```
2022-01-18 19:38:04.393+0000 [id=30] INFO jenkins.InitReactorRunner$
2022-01-18 19:38:04.598+0000 [id=29] INFO jenkins.InitReactorRunner$
2022-01-18 19:38:04.599+0000 [id=29] INFO jenkins.InitReactorRunner$
2022-01-18 19:38:04.600+0000 [id=29] INFO jenkins.InitReactorRunner$
2022-01-18 19:38:04.765+0000 [id=46] INFO hudson.model.AsyncPeriodic
2022-01-18 19:38:04.840+0000 [id=46] INFO hudson.util.Retrier#start:
2022-01-18 19:38:06.167+0000 [id=30] INFO jenkins.install.SetupWizar
```

```
*****
*****
*****
```

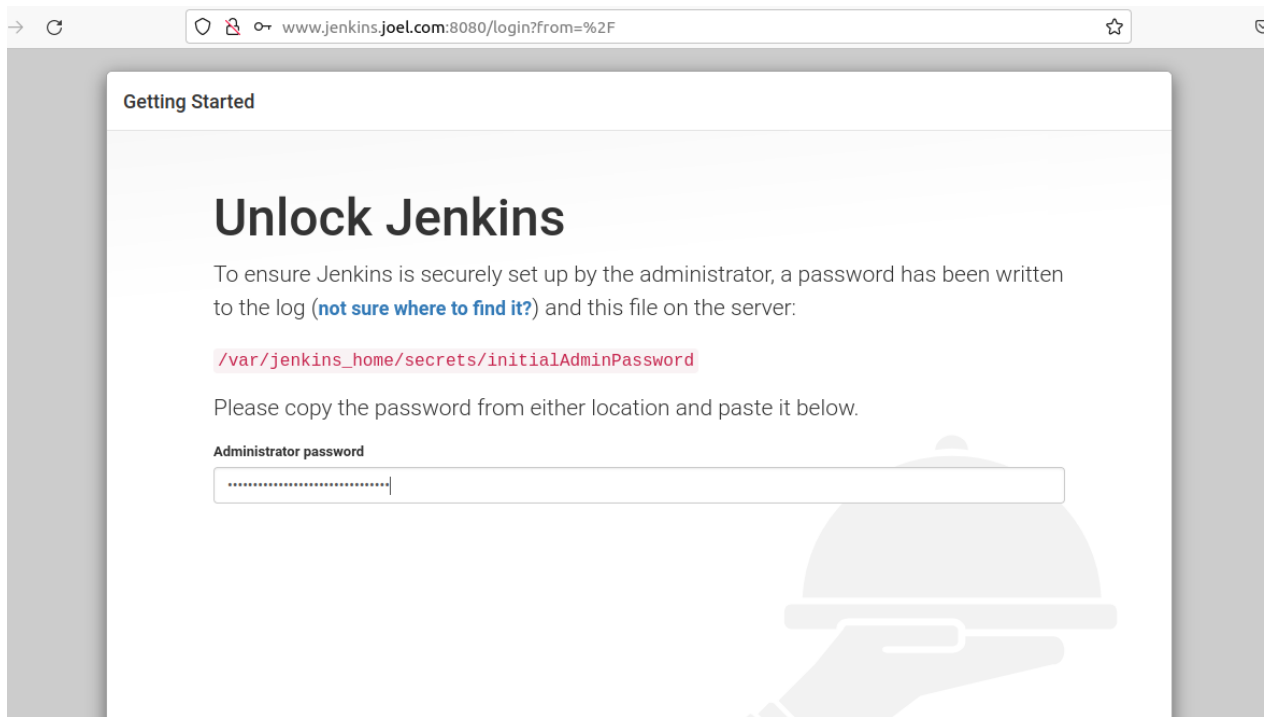
Jenkins initial setup is required. An admin user has been created and a pa
Please use the following password to proceed to installation:

d24a9de99433489d82ba94c6eb3157be

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

```
*****
*****
*****
```

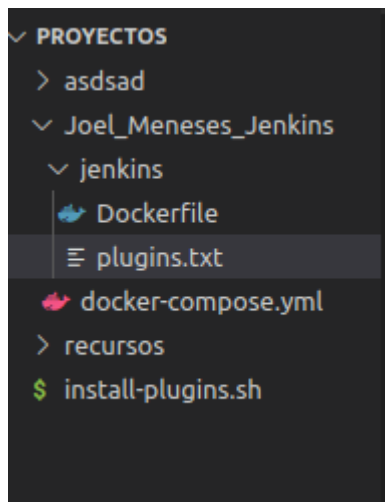
Verifica que puedas ingresar al contenedor a través del navegador web de tu preferencia.



Lo config

4. Requisitos básicos

Creamos una carpeta que va a tener los ficheros de configuración de jenkins



4.1 Dockerfile

Posteriormente nos creamos un fichero dockerfile con el contenido siguiente

```
Joel_Meneses_Jenkins > jenkins > Dockerfile
1 FROM jenkins/jenkins
2
3 USER root
4 RUN apt-get -y update && apt-get install -y maven
5
6 USER jenkins
7 COPY plugins.txt /usr/share/jenkins/ref/plugins.txt
8 RUN /usr/local/bin/install-plugins.sh < /usr/share/jenkins/ref/plugins.txt
9
```

Explicación del fichero dockerfile

- En la línea 1 se indica la imagen base de la que se parte para construir la nueva imagen. *Utilizamos la misma que anteriormente.*
- En la línea 4 instalamos Maven utilizando comandos apt-get. Estos comandos deben ejecutarse como usuario root. Indicamos que el usuario de ejecución es root en la línea 3. Tras ejecutarse los comandos apt-get, el resto de comandos pueden ejecutarse con el usuario jenkins. Establecemos el nuevo usuario de ejecución en la línea 6.
- En la línea 7 copiamos el fichero plugins.txt en el directorio /usr/share/jenkins/ref/. Este fichero contiene la lista de plugins que queremos que se instalen en Jenkins. En la última línea se ejecuta el script install-plugins.sh que toma como entrada la lista de plugins incluidos en el fichero plugins.txt y los instala en Jenkins. Este script viene incluido en la imagen **jenkins/jenkins**.

4.2 Plugins.txt

Ahora creamos un fichero llamado plugins.txt que contienen los plugins a instalar en jenkins, el contenido sera el siguiente


```
Joel_Meneses_Jenkins > jenkins > ≡ plugins.txt
1  ace-editor
2  ant
3  antisamy-markup-formatter
4  apache-httpcomponents-client-4-api
5  authentication-tokens
6  branch-api
7  build-monitor-plugin
8  build-pipeline-plugin
9  cloudbees-folder
10 conditional-buildstep
11 copyartifact
12 credentials
13 credentials-binding
14 deploy
15 display-url-api
16 docker-commons
17 docker-workflow
18 durable-task
19 git
20 github
21 github-api
22 git-client
23 git-server
24 gradle
25 greenballs
26 handlebars
27 jackson2-api
28 javadoc
```

4.3 Docker-Compose.yml

Nos queda por crear nuestro docker-compose.yml con el contenido siguiente

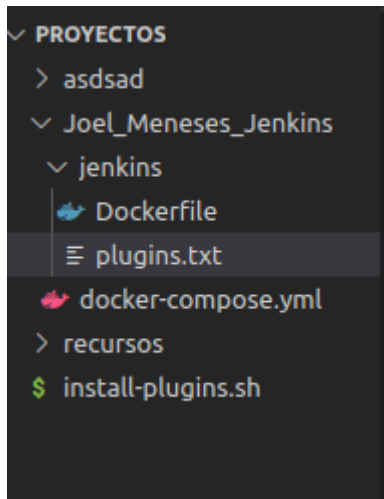

```
Joel_Meneses_Jenkins > docker-compose.yml
1  version: '3'
2  services:
3    master:
4      build: jenkins
5      image: dpl/jenkins:latest
6      restart: unless-stopped
7      hostname: jenkins
8      ports:
9        - "8080:8080"
10       - "50000:50000"
11     volumes:
12       - jenkins_home:/var/jenkins_home
13
14  volumes:
15    jenkins_home:
```

Explicación del fichero docker-compose.yml

- En la línea 4 indicamos el directorio sobre el que se va a construir la imagen. En este caso **jenkins**, siendo un subdirectorio de donde se encuentra el fichero *docker-compose.yml*.
- En la línea 5 indicamos el nombre de nuestra imagen. Podéis cambiar **dpl/jenkins:latest** por el nombre que le queráis dar a vuestra propia imagen con Jenkins.
- En la línea 6 se establece que se reinicie el contenedor a menos que se detenga explícitamente o el motor de Docker se detenga o reinicie.
- En la línea 7 indicamos el hostname del contenedor.
- En las líneas 9 y 10 se exponen los puertos **8080** y **50000** respectivamente.
- Las líneas de la 12 a la 15 sirven para definir el volumen *jenkins_home*. Este volumen se utiliza para que los cambios que realicemos en la configuración de Jenkins persistan incluso tras la destrucción del contenedor.

Creamos una carpeta llamada **jenkins** y metemos los ficheros **dockerfile** y **pulgins.txt** dentro de ella

Teniendo una estructura similar a la siguiente



4.4 Construcción de la imagen

Para construir la imagen ejecutamos el siguiente comando:

```
docker-compose build
```

```
joel@joel-joel:~/Proyectos/Joel_Meneses_Jenkins$ sudo docker-compose build
[sudo] contraseña para joel:
Building master
Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM jenkins/jenkins
--> b3febfc71dcd
Step 2/6 : USER root
--> Using cache
--> 9dea0eefe523
Step 3/6 : RUN apt-get -y update && apt-get install -y maven
--> Using cache
--> 31389cd1cba4
Step 4/6 : USER jenkins
--> Using cache
--> fa5ec2928da5
Step 5/6 : COPY plugins.txt /usr/share/jenkins/ref/plugins.txt
--> Using cache
--> bdd8d709a656
Step 6/6 : RUN /usr/local/bin/install-plugins.sh < /usr/share/jenkins/ref/plugins.txt
--> Using cache
--> d4bdcfe4fc87
Successfully built d4bdcfe4fc87
Successfully tagged dpl/jenkins:latest
```

Para arrancar el contenedor con Jenkins ejecutamos el siguiente comando:

```
docker-compose up -d
```

```
joel@joel-joel:~/Proyectos/Joel_Meneses_Jenkins$ sudo docker-compose up -d
Starting joel_meneses_jenkins_master_1 ... done
```

Arrancado jenkins accedemos a nuestro dominio para entrar en la consola jenkins

Para obtener la contraseña del usuario admin de Jenkins ejecutamos el siguiente comando:

```
docker exec -it dockerjenkins_master_1 cat /var/jenkins_home/secrets/initialAdminPassword
```

```
joel@joel-joel:~/Proyectos/Joel_Meneses_Jenkins$ docker exec -it joel_meneses_jenkins_master_1 cat /var/jenkins_home/secrets/initialAdminPassword
sword
df5c288d08b74c47839d2575483f43ac
```

Ahora podemos configurar jenkins en docker

← → ↻

🔒 www.jenkins.joel.com:8080/login?from=%2F ☆

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/jenkins_home/secrets/initialAdminPassword
```

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

Administrator password

.....

Continue

Enlace github: [enlace](#)