

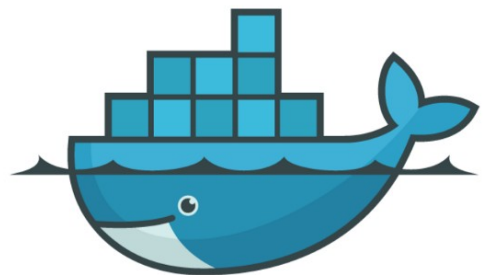
# Instalación de Jenkins en Docker

2ºDAW

## Despliegue de aplicaciones Web



**Jenkins**



**docker**

*Jonay Hernández Izquierdo*

## Índice

1. Introducción.....	3
2. Instalación Docker.....	3
3. Instalación Docker Compose.....	7
4. Configuración.....	12

## 1. Introducción.

En la actualidad muchas empresas de desarrollo se están usando practicas para agilizar el trabajo, una de estas practicas es la integración continua esta permite a los desarrolladores combinar los cambios del código en un repositorio central de manera periódica con sistema de versión y pruebas automáticas.

Otra practica importante en el desarrollo es la de desarrollo continuo, esta practica consiste en la automatización de actualización para la aplicaciones implementada en producción.

Estas dos practicas se conocen como CI/CD y hay muchas herramientas que nos permiten realizar estas practicas de manera mas eficientes, una de las mas usas es Jenkins.

Jenkins es una herramienta open source para que nos permite compilar y probar proyectos de software que de manera continua que nos permite integrar cambios en el proyecto y en las actualizaciones.

Ademas tenemos Docker que ya hemos hablado y trabajado en otras guias. Docker es una herramienta que nos permiten lanzar “contenedores” que nos permiten ejecutar aplicaciones con una capa de abstracción y totalmente aisladas del sistema.

En esta guía veremos concretamente como instalar Jenkins usando la tecnología de contenedores.

## 2. Instalación Docker.

Un requisito fundamental antes de proceder con la instalación es tener instalado tanto Docker como Docker Compose en nuestro servidor y por otro lado asegurarse de que no tenemos ninguna aplicaciones corriendo en el puertos 8080.

Con los requisitos ya mencionados realizados vamos a proceder con la instalación del primero contenedor usando docker, para ello vamos a la pagina dockerhub y buscamos la imagen de jenkins:

The screenshot shows the Docker Hub interface for the Jenkins official image. At the top, there's a description: "Web-based MongoDB admin interface, written with Node.js and express". Below this, there are tags: "Container", "Linux", "x86-64", "ARM 64", "DevOps Tools", and "Application Frameworks". The main section features the Jenkins logo, the name "jenkins", a green checkmark indicating it's an "Official Image", and statistics: "100M+ Downloads" and "5.4K Stars". It also notes "Updated 4 years ago" and a deprecation warning: "DEPRECATED; use 'jenkins/jenkins:lts' instead". Below this, there are more tags: "Container", "Linux", "x86-64", and "DevOps Tools". The bottom section shows the "kapacitor" image, also an official image, with "50M+ Downloads" and "223 Stars", updated "a month ago". The description for kapacitor is partially visible: "Kapacitor is an open source framework for processing, monitoring, and alerti..."

Con la imagen ya seleccionado podemos descargar o realizar un pull por linea de comando usando el nombre que vimos en la página:

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

```
jonay@ubuntu-dpl:~$ sudo docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
0e29546d541c: Extracting 53.48MB/54.92MB
11423c93b49c: Download complete
0a7e49db0973: Download complete
977259ae4ac6: Download complete
664d083fcd2d: Download complete
8a20d9bdf720: Download complete
cc94036a9149: Download complete
771e489c9ff6: Download complete
11d0bec8f08b: Download complete
ca11fa5d5337: Download complete
b465e025b587: Download complete
e0e171e8e73c: Download complete
969a814948b7: Download complete
1a23880b1424: Waiting
```

Ahora comprobamos que tenemos la imagen descargada correctamente:

```
sudo docker images
```

```
jonay@ubuntu-dpl:~$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
jenkins/jenkins     lts             9aee0d53624f   6 days ago     441MB
jonay@ubuntu-dpl:~$
```

Y ahora vamos a crear el contenedor con la siguiente linea.

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

donde:

- -p : mapeado de puerto de nuestra maquina al contenedor (puerto-servidor:puerto-contenedor).
- -v: aquí se declara el volumen de carpeta entre la maquina y el servidor, en este caso este volumen nos permitirá tener la instalación de jenkins y la configuración en esa carpeta para que al eliminar el contenedor y volver a lanzarlos no perdamos los datos.

```
jonay@ubuntu-dpl:~/jenkins_home$ sudo docker run -p 8080:8080 -p 50000:50000 -v /home/jonay/jenkins_home:/var/jenkins_home jenkins/jenkins:lts
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
2022-01-18 17:05:56.219+0000 [id=1] INFO org.eclipse.jetty.util.log.Log#initialized: Logging initialized @772ms to org.eclipse.jetty.util.log.JavaUtilLog
2022-01-18 17:05:56.354+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2022-01-18 17:05:57.666+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2022-01-18 17:05:57.771+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
2022-01-18 17:05:58.192+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support For /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2022-01-18 17:05:58.273+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager workerName=node0
2022-01-18 17:05:58.273+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, using defaults
2022-01-18 17:05:58.274+0000 [id=1] INFO o.e.j.s.session.HouseKeeper#startScavenging: node0 Scavenging every 60000ms
2022-01-18 17:05:58.943+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2022-01-18 17:05:59.202+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started w.6ba30587(Jenkins v2.319.2,/,file:///var/jenkins_home/war,/AVAILABLE){/var/jenkins_home/war}
2022-01-18 17:05:59.263+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started @3817ms
2022-01-18 17:05:59.271+0000 [id=23] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2022-01-18 17:05:59.730+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2022-01-18 17:05:59.770+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
```

Antes de acceder al jenkins vamos a crear un dominio en apache.

Primero añadimos nuestro dominio en “/etc/hosts”

```
GNU nano 4.8 /etc/hosts
127.0.0.1    localhost
127.0.1.1    ubuntu-dpl
#127.0.0.1    www.jonayhisystem.com jonayhisystem.com
#127.0.0.1    www.jonayic.com jonayic.com
127.0.0.1    www.jenkins.jonay.com jenkins.jonay.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
```

Creamos el conf.

```
jonay@ubuntu-dpl:/etc/apache2/sites-available$ sudo cat jenkins.jonay.conf
<VirtualHost *:80>
    ServerName www.jenkins.jonay.com
    ServerAlias jenkins.jonay.com
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    Redirect "/" "http://www.jenkins.jonay.com:8080"
</VirtualHost>
jonay@ubuntu-dpl:/etc/apache2/sites-available$
```

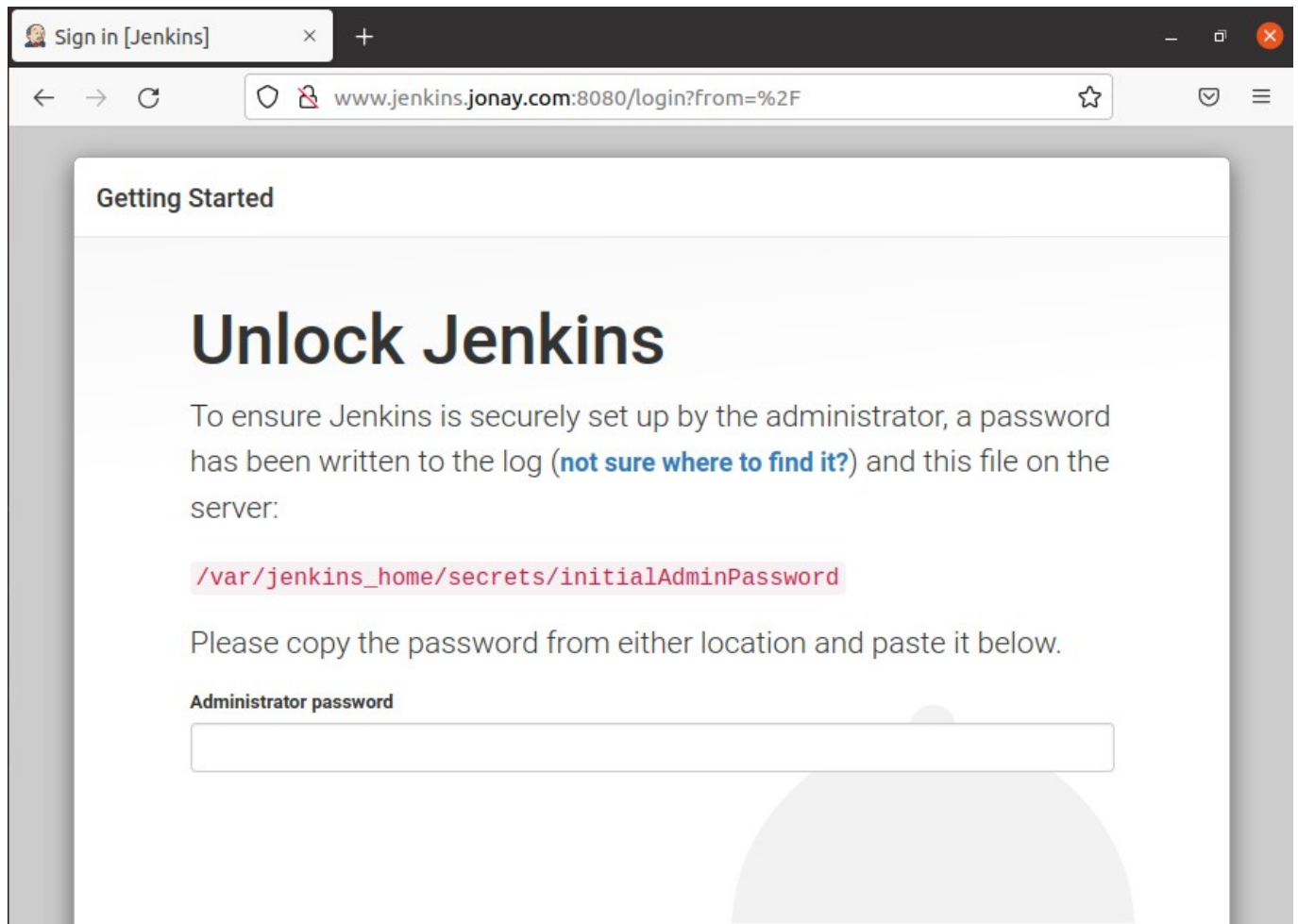
Activamos el sitio web.

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

Reiniciamos apache.

```
jonay@ubuntu-dpl:/etc/apache2/sites-available$ sudo systemctl restart apache2.service
jonay@ubuntu-dpl:/etc/apache2/sites-available$
```

Y ahora accedemos al dominio.



### 3. Instalación Docker Compose.

Ya vimos como crear un contenedor con Docker ahora vamos a ver como realizarlo con Docker Compose.

Primero vamos a crear un carpeta donde pondremos todos estos ficheros. Y empezamos creando el fichero Dockerfile que lo usaremos para añadir configuraciones a Jenkins, en esta caso actualizar el sistema, instalar maven y una serie de plugins.

```
jonay@ubuntu-dpl:/srv/jenkins$ ls
Dockerfile
jonay@ubuntu-dpl:/srv/jenkins$ cat Dockerfile
FROM jenkins/jenkins

USER root
RUN apt-get -y update && apt-get install -y maven

USER jenkins
COPY plugins.txt /usr/share/jenkins/ref/plugins.txt
RUN /usr/local/bin/install-plugins.sh < /usr/share/jenkins/ref/plugins.txt
jonay@ubuntu-dpl:/srv/jenkins$
```

#### Lista de plugins

ace-editor

ant

antisamy-markup-formatter

apache-httpcomponents-client-4-api

authentication-tokens

branch-api

build-monitor-plugin

build-pipeline-plugin

cloudbees-folder

conditional-buildstep

copyartifact

credentials

credentials-binding

deploy

display-url-api

docker-commons

docker-workflow

durable-task

Página. 8 de 15	<b>Instalación Jenkins en Docker</b>	20/01/22
-----------------	--------------------------------------	----------

git  
 github  
 github-api  
 git-client  
 git-server  
 gradle  
 greenballs  
 handlebars  
 jackson2-api  
 javadoc  
 jquery  
 jquery-detached  
 jsch  
 junit  
 mailer  
 matrix-project  
 maven-plugin  
 momentjs  
 nested-view  
 parameterized-trigger  
 pipeline-build-step  
 pipeline-graph-analysis  
 pipeline-input-step  
 pipeline-milestone-step  
 pipeline-model-api  
 pipeline-model-declarative-agent  
 pipeline-model-definition  
 pipeline-model-extensions  
 pipeline-rest-api  
 pipeline-stage-step  
 pipeline-stage-tags-metadata  
 pipeline-stage-view  
 plain-credentials  
 run-condition  
 scm-api



Página. 9 de 15	Instalación Jenkins en Docker	20/01/22
-----------------	-------------------------------	----------

script-security  
 ssh-credentials  
 structs  
 token-macro  
 workflow-aggregator  
 workflow-api  
 workflow-basic-steps  
 workflow-cps  
 workflow-cps-global-lib  
 workflow-durable-task-step  
 workflow-job  
 workflow-multibranch  
 workflow-scm-step  
 workflow-step-api  
 workflow-support

Los añadimos a un txt.

```

jonay@ubuntu-dpl:/srv/jenkins$ sudo nano plugins.txt
jonay@ubuntu-dpl:/srv/jenkins$ cat plugins.txt
ace-editor
analysis-core
ant
antisamy-markup-formatter
apache-httpcomponents-client-4-api
authentication-tokens
branch-api
build-monitor-plugin
build-pipeline-plugin
checkstyle
cloudbees-folder
conditional-buildstep
copyartifact
credentials
credentials-binding
deploy
display-url-api
docker-commons
docker-workflow
durable-task
findbugs
git
github
github-api
git-client
git-server
gradle
greenballs
  
```

Ahora creamos el docker-compose.yml con la siguiente información.

Donde:

- build: ponemos un “.” para que busque el Dockerfile en la misma carpeta
- image: imagen que se usara en el contenedores
- restart: Reinicia todos los servicio detenidos y en ejecución.
- hostname: nombre del contenedor
- ports: puertos de la maquina
- volumen: lo mismo en se explica en el apartado de arriba

```
jonay@ubuntu-dpl:/srv/jenkins$ sudo nano docker-compose.yml
jonay@ubuntu-dpl:/srv/jenkins$ cat docker-compose.yml
version: '3'
services:
  master:
    build: .
    image: jenkins/jenkins:lts
    restart: unless-stopped
    hostname: jenkins
    ports:
      - "8080:8080"
      - "50000:50000"
    volumes:
      - jenkins_home:/var/jenkins_home
volumes:
  jenkins_home:
jonay@ubuntu-dpl:/srv/jenkins$
```

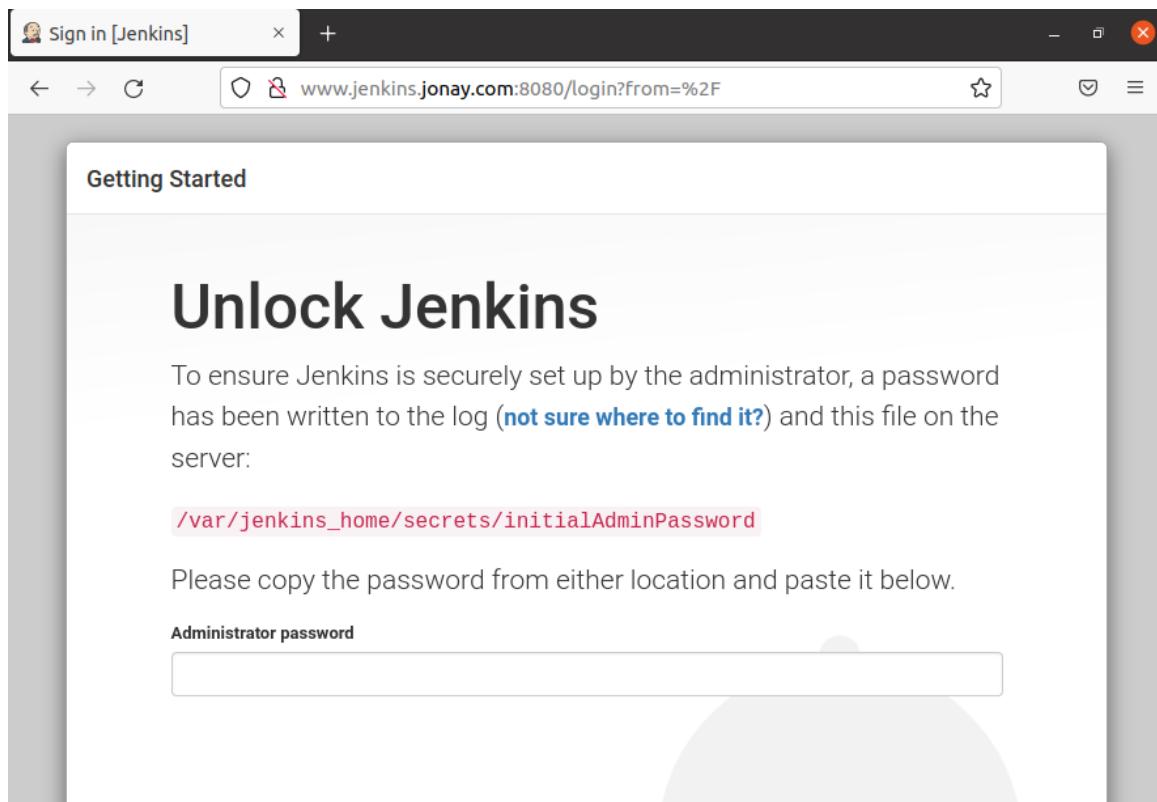
Construimos el contenedor.

```
jonay@ubuntu-dpl:/srv/jenkins$ sudo docker-compose build
Building master
Sending build context to Docker daemon 5.12kB
Step 1/6 : FROM jenkins/jenkins
--> b3febfc71dcd
Step 2/6 : USER root
--> Using cache
--> f941f324c406
Step 3/6 : RUN apt-get -y update && apt-get install -y maven
--> Running in 0f0d598a3421
Get:1 http://security.debian.org/debian-security bullseye-security InRelease [44.1 kB]
Get:2 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [39.4 kB]
Get:4 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [107 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [2592 B]
Fetched 8492 kB in 2s (3856 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java dbus
  default-jre-headless java-common libapparmor-java libapache-pom-java
```

Lanzamos el contenedor.

```
jonay@ubuntu-dpl:/srv/jenkins$ sudo docker-compose up
Creating jenkins_master_1 ... done
Attaching to jenkins_master_1
master_1 | Running from: /usr/share/jenkins/jenkins.war
master_1 | webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
master_1 | 2022-01-19 15:48:32.543+0000 [id=1] INFO      org.eclipse.jetty.util.log.Log#initializ
ed: Logging initialized @1321ms to org.eclipse.jetty.util.log.JavaUtilLog
master_1 | 2022-01-19 15:48:32.802+0000 [id=1] INFO      winstone.Logger#logInternal: Beginning e
xtraction from war file
master_1 | 2022-01-19 15:48:36.128+0000 [id=1] WARNING  o.e.j.s.handler.ContextHandler#setContext
Path: Empty contextPath
master_1 | 2022-01-19 15:48:36.228+0000 [id=1] INFO      org.eclipse.jetty.server.Server#doStart:
jetty-9.4.43.v20210629; built: 2021-06-30T11:07:22.254Z; git: 526006ecfa3af7f1a27ef3a288e2bef7e
a9dd7e8; jvm 11.0.13+8
master_1 | 2022-01-19 15:48:37.248+0000 [id=1] INFO      o.e.j.w.StandardDescriptorProcessor#visi
tServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
master_1 | 2022-01-19 15:48:37.325+0000 [id=1] INFO      o.e.j.s.s.DefaultSessionIdManager#doStar
t: DefaultSessionIdManager workerName=node0
master_1 | 2022-01-19 15:48:37.326+0000 [id=1] INFO      o.e.j.s.s.DefaultSessionIdManager#doStar
t: No SessionScavenger set, using defaults
master_1 | 2022-01-19 15:48:37.327+0000 [id=1] INFO      o.e.j.server.session.HouseKeeper#startSc
avenging: node0 Scavenging every 600000ms
```

Ahora para comprobar que todo ha ido bien podemos acceder al dominio que hemos creado anteriormente, Asegurate de haber borrado el anterior contenedor para que no hay conflicto con el puerto.



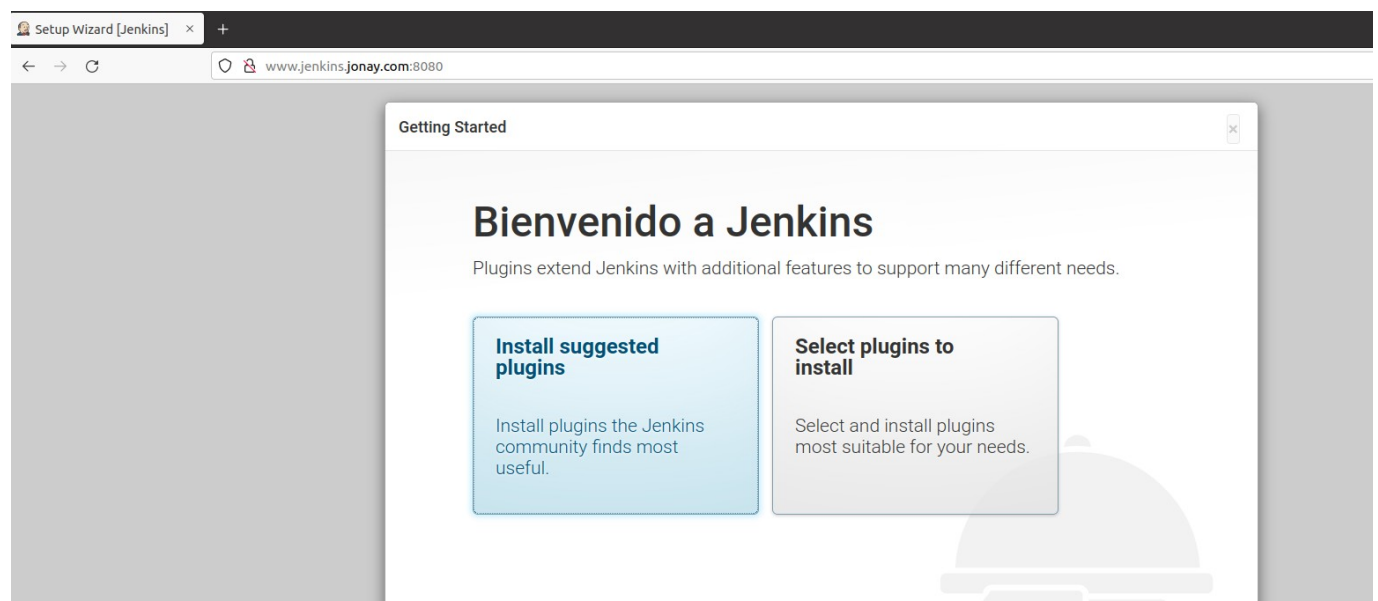
## 4. Configuración.

Vamos a configurar Jenkins como ya se a hecho en otras guiás. Primero necesitamos la contraseña para realizar la configuración de Jenkins, para esto vamos a lanzar el siguiente comando contra el servidor para que nos devuelve la contraseña.

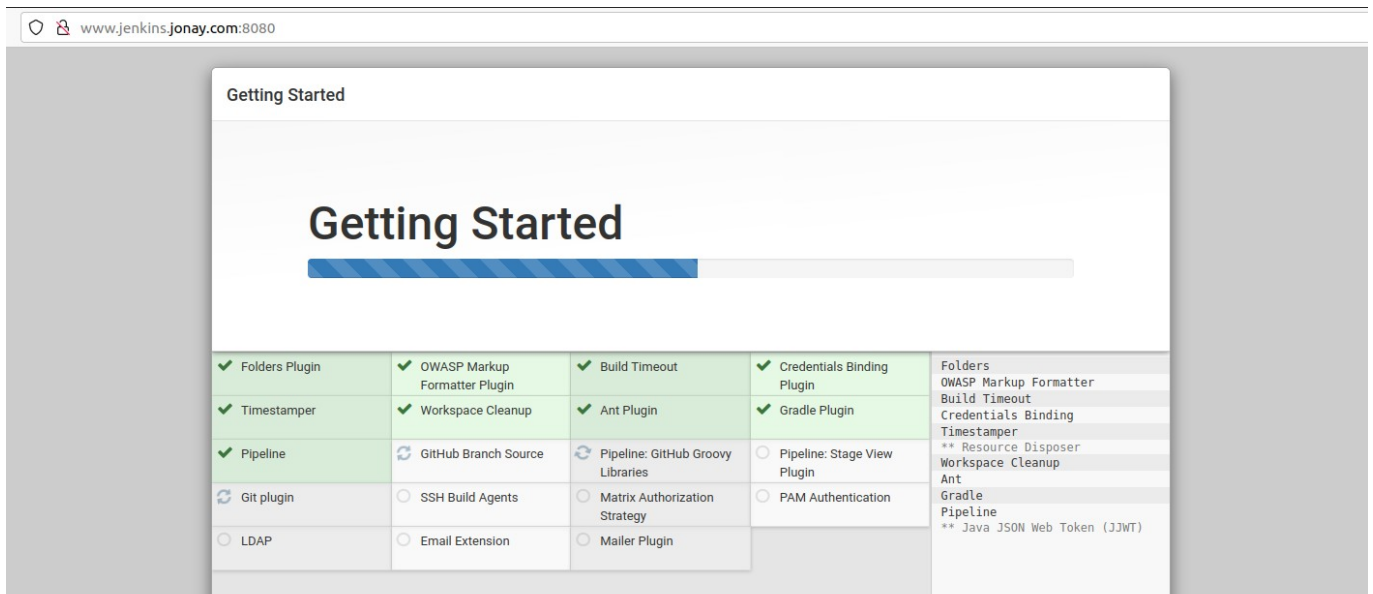
```
sudo docker exec -it dockerjenkins_master_1 cat /var/jenkins_home/secrets/initialAdminPassword
```

```
jonay@ubuntu-dpl:~$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS                               NAMES
6d19853d1995   jenkins/jenkins:lts   "/sbin/tini -- /usr/_    4 minutes ago   Up 4 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp   jenkins_master_1
jonay@ubuntu-dpl:~$ sudo docker exec -it jenkins_master_1 cat /var/jenkins_home/secrets/initialAdminPassword
3d85713e9898477eb4cbb61620ce0a77
jonay@ubuntu-dpl:~$
```

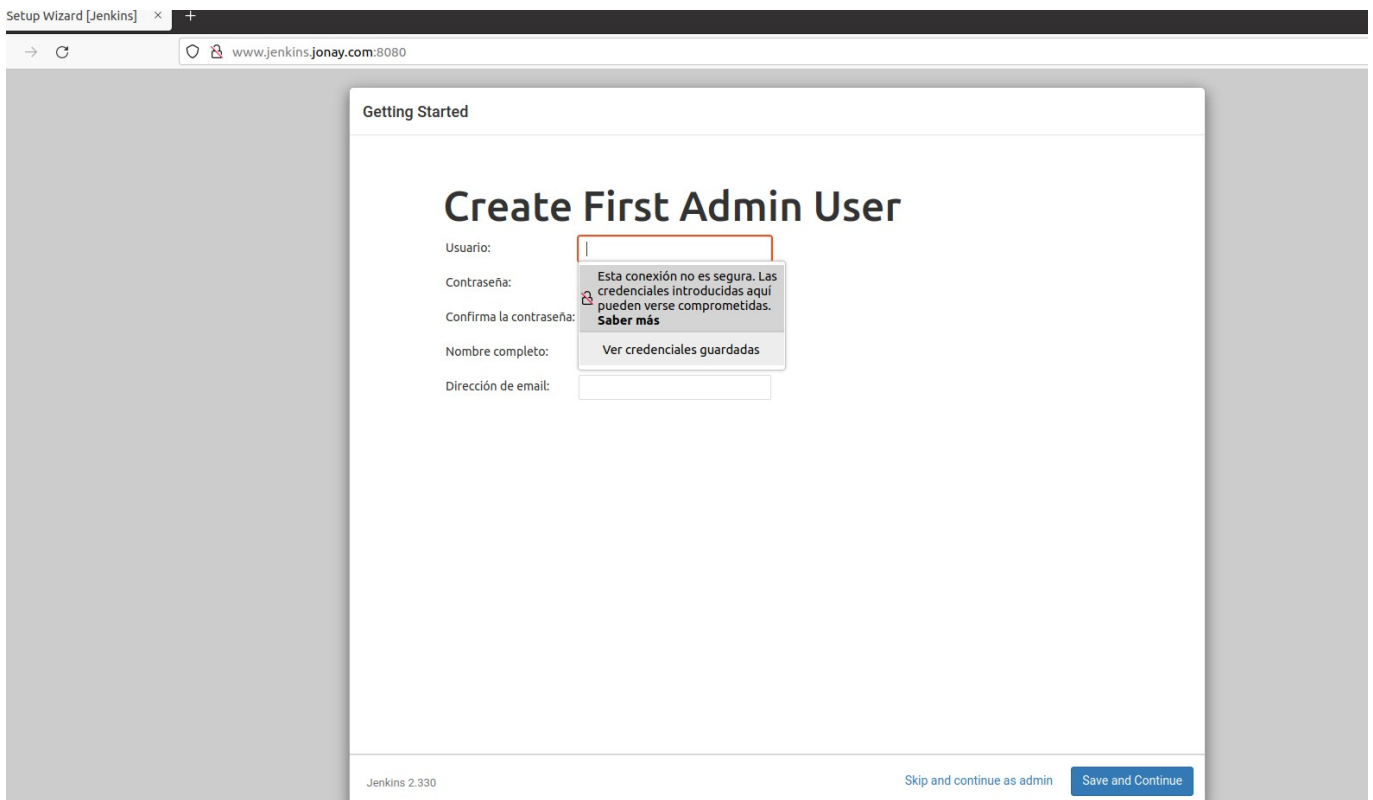
La añadimos en la página de Jenkins y veremos la siguiente página donde seleccionaremos la primera opción.



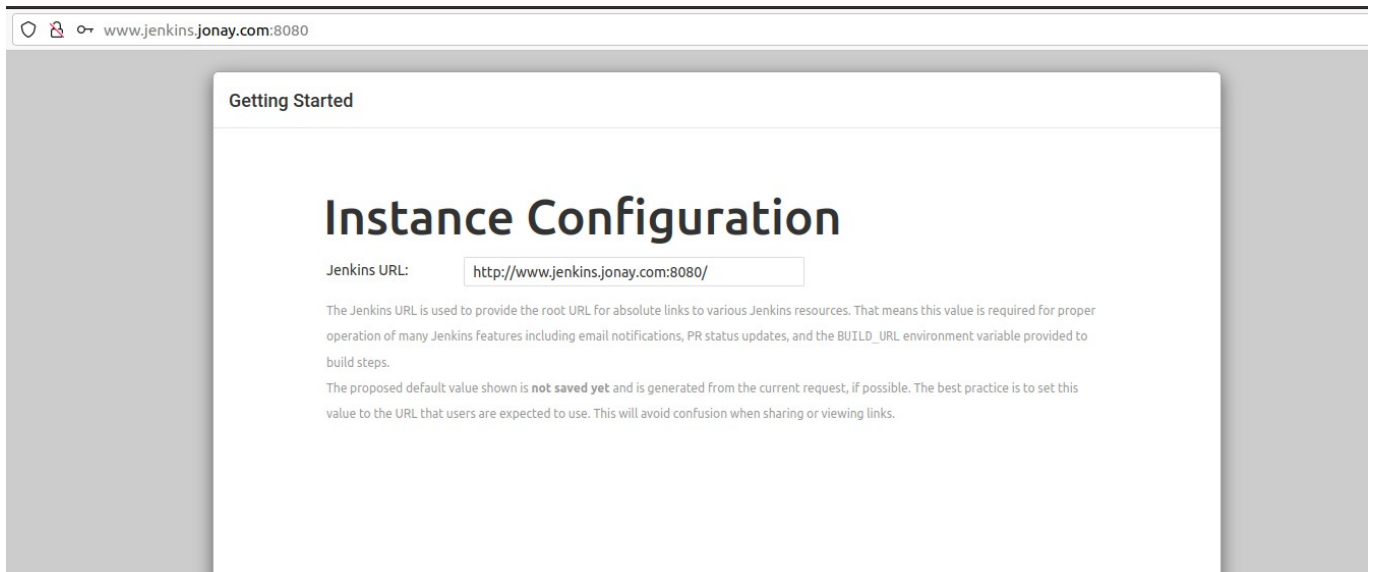
## Getting Started.



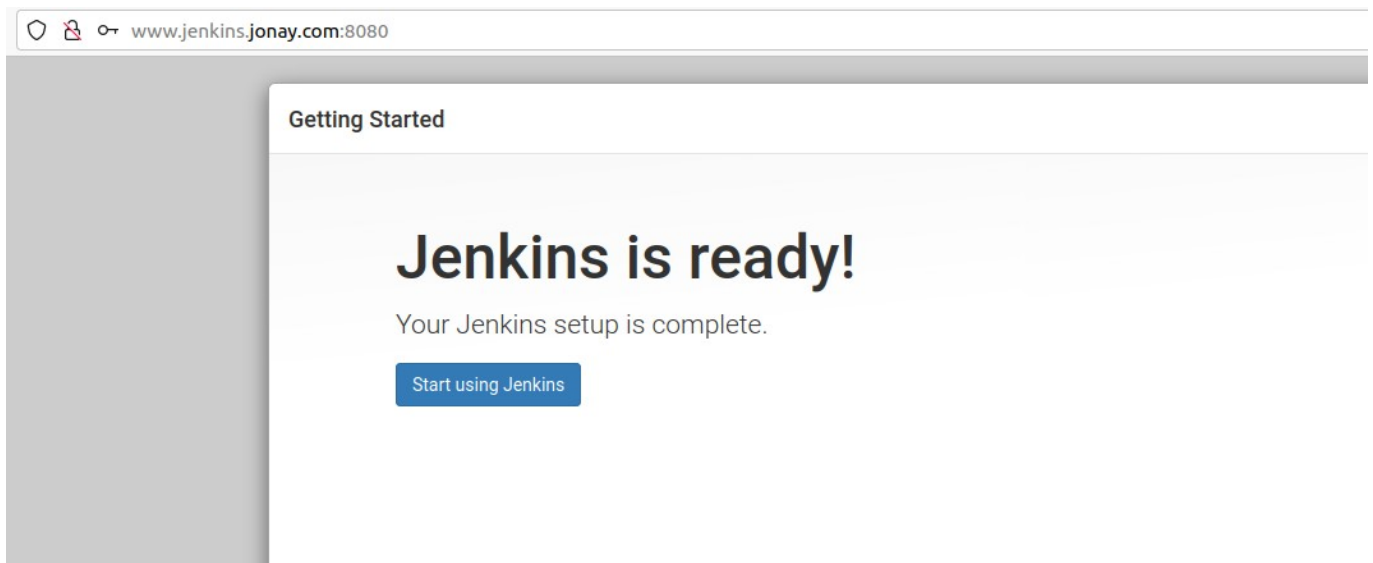
## Creamos el usuario.



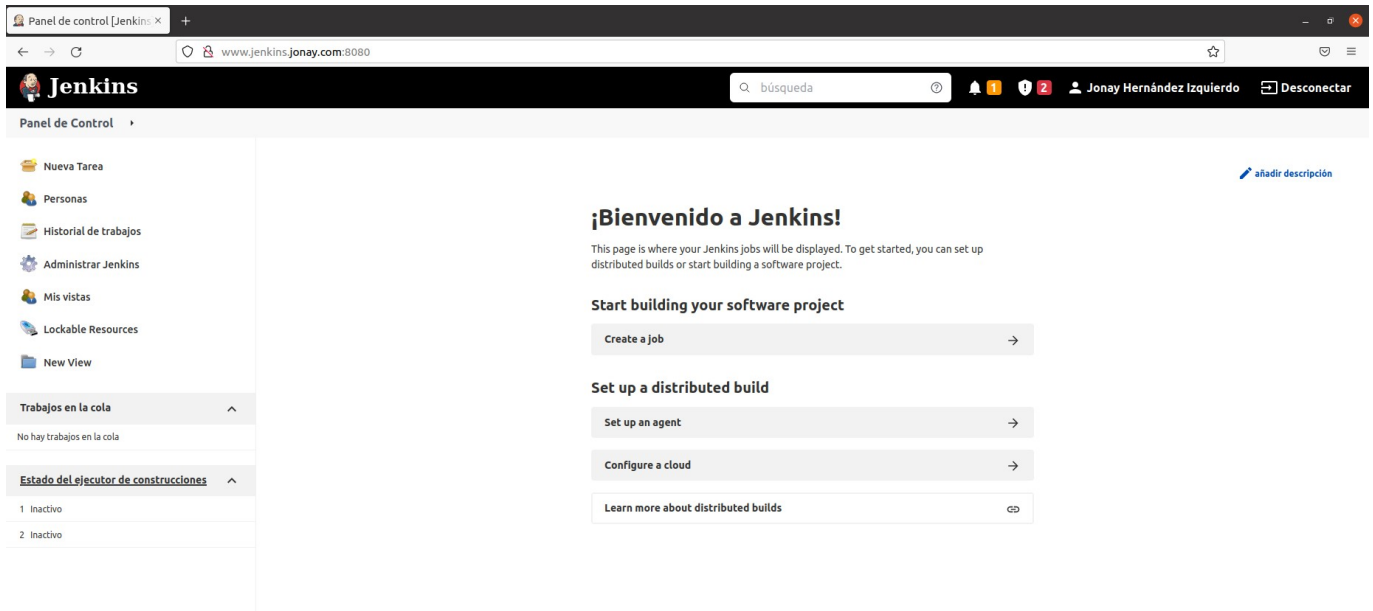
URL para para Jenkins.



Finalizada la instalación.



Página principal.



The screenshot displays the Jenkins web interface in a browser window. The address bar shows the URL `www.jenkins.jonay.com:8080`. The page features a dark header with the Jenkins logo, a search bar, and user information for Jonay Hernández Izquierdo. The main content area is titled "¡Bienvenido a Jenkins!" and provides instructions on how to get started. It includes several actionable buttons: "Create a job", "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". A sidebar on the left contains navigation links for tasks, users, job history, and system administration. Below these links, there are sections for "Trabajos en la cola" (Jobs in queue) and "Estado del ejecutor de construcciones" (Build executor status), both showing no active jobs or executors.

Panel de control [Jenkins: X] +

← → ↻ `www.jenkins.jonay.com:8080` 🔍 búsqueda 🔔 1 🛑 2 👤 Jonay Hernández Izquierdo 🔌 Desconectar

Panel de Control ▾

- Nueva Tarea
- Personas
- Historial de trabajos
- Administrar Jenkins
- Mis vistas
- Lockable Resources
- New View

Trabajos en la cola ^

No hay trabajos en la cola

Estado del ejecutor de construcciones ^

1 Inactivo

2 Inactivo

**¡Bienvenido a 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**

Create a job →

**Set up a distributed build**

Set up an agent →

Configure a cloud →

Learn more about distributed builds ↗

[añadir descripción](#)