

3. Haga clic en **Crear**.
4. En la página **Crear una instancia**, escriba `my-vm-with-docker` en **Nombre**.
5. En **Región y Zona**, seleccione la región `us-central1` y la zona `us-central1-a`.
6. En **configuración de la maquina** seleccione `De uso general`
7. En **Tipo de máquina**, acepte el valor predeterminado y seleccione `e2-micro`
8. No modifique los demás valores predeterminados.
9. Para crear e iniciar la VM, haga clic en **Crear**.
10. Una vez en ejecución nuestra VM, cree una **regla de firewall** que permita trafico de ingreso en el puerto **8080**.

Tarea 2: Instale el motor de contenedores Docker en la VM

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. **Docker** le permite separar sus aplicaciones de su infraestructura para que pueda entregar software rápidamente. Con **Docker**, puede administrar su infraestructura de la misma manera que administra sus aplicaciones.

Docker brinda la capacidad de empaquetar y ejecutar una aplicación en un entorno poco aislado llamado contenedor. El aislamiento y la seguridad le permiten ejecutar muchos contenedores simultáneamente en un host determinado. Los contenedores son livianos y contienen todo lo necesario para ejecutar la aplicación, por lo que no es necesario depender de lo que está instalado en el host. Puede compartir contenedores mientras trabaja y asegurarse de que todas las personas con las que comparte obtengan el mismo contenedor que funciona de la misma manera.

1. Conectese por ssh a la vm `my-vm-with-docker`
2. Ejecute los pasos 1, 2 y 3 de la siguiente guía <https://docs.docker.com/engine/install/debian/#install-using-the-repository>. La guía incluye una serie de comandos que se deben ejecutar en el Shell de la VM.
3. Para confirmar la instalación, en el paso anterior, debió descargar una imagen de prueba y ejecutarla en un contenedor. Cuando el contenedor se ejecuta, imprime un mensaje de confirmation y sale.
4. En este momento ya tiene Docker instalado en su maquina virtual. Por ahora debe utilizar **sudo** para ejecutar comandos de Docker. Si desea cambiar este comportamiento, puede ir al siguiente enlace, donde se le indica como hacerlo <https://docs.docker.com/engine/install/linux-postinstall/>.

Tarea 3: Compilar y ejecutar una imagen Docker mediante un Dockerfile

1. Actualice los repositorios de paquetes de nuestra instalación de Linux en la vm:
`$sudo apt-get update`
2. Instale el sistema de control de versiones **Git** (necesario para obtener el código de la aplicación a contenerizar):
`$sudo apt-get install git-all`
3. Obtenga un ejemplo de la aplicación a contenerizar:
`git clone https://github.com/docker/welcome-to-docker`

4. Navegue hacia el directorio `welcome-to-docker`:

```
cd welcome-to-docker/
```

5. Explore el contenido del directorio con el comando **ls**

Podrá notar que se encuentra un archivo `Dockerfile`, el cual, contiene toda la configuración para la creación de la imagen del contenedor.

6. Explore el archivo **Dockerfile**:

```
Nano Dockerfile
```

Vuelva al Shell de la vm con **Ctrl + X**

7. Genere la imagen del contenedor:

```
$sudo docker build -t welcome-to-docker .
```

En el comando anterior, la bandera `-t` etiqueta tu imagen con un nombre, y le permite a Docker saber dónde puede encontrar el `Dockerfile`.

8. Verifique que la imagen ha sido creada:

```
$sudo docker images
```

```
jenkins@ubuntu:~/welcome-to-docker$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
welcome-to-docker	latest	d60079f835c4	9 hours ago	225MB
wordpress	latest	bc823df9ead2	8 days ago	668MB
jenkins/jenkins	lts-jdk17	6adc6425cd34	2 weeks ago	476MB
docker/welcome-to-docker	latest	c1f619b6477e	3 weeks ago	18.6MB
hello-world	latest	9c7a54a9a43c	6 months ago	13.3kB

9. Ejecute la imagen del contenedor **welcome-to-docker**:

```
$sudo docker run -d -p 8080:3000 --name welcome-to-docker  
welcome-to-docker
```

Este commando ejecuta la imagen del contenedor, la mantiene en ejecución en segundo plano y expone la aplicación en el puerto 8080.

10. Verifique que el contenedor se encuentra en ejecución:

```
$sudo docker ps
```

11. Verifique que la aplicación esta disponible en el puerto 8080 de su VM. Para ello, abra un navegador e ingrese a http://ip_externa_vm:8080/, reemplace **ip_externa_vm** por la ip externa/publica asignada a su vm.

La aplicación deberá estar disponible.

Congratulations!!!

You ran your first container.



12. Detenga la ejecución del contenedor:

```
$sudo docker stop welcome-to-docker
```

En este momento el contenedor deja de ejecutarse, por lo tanto, la aplicación no deberá estar disponible. Intente ingresar nuevamente al enlace de su aplicación, deberá obtener el siguiente resultado.



No se puede acceder a este sitio

34.16.136.56 rechazó la conexión.

Intenta:

- Comprobar la conexión.
- Comprobar el proxy y el firewall.

ERR_CONNECTION_REFUSED

Volver a cargar

13. Inicie nuevamente la ejecución del contenedor:

```
$sudo docker start welcome-to-docker
```

La aplicación debe estar nuevamente disponible.

Congratulations!!!

You ran your first container.



14. Detenga la ejecución del contenedor, para permitir la siguiente actividad.

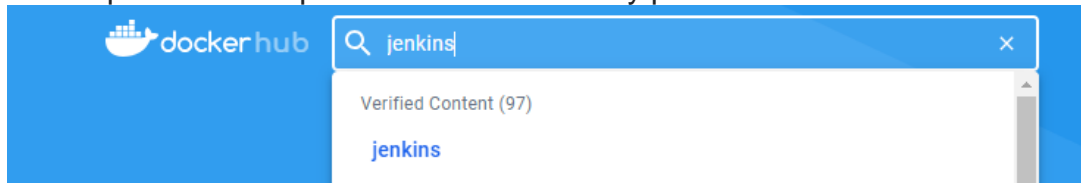
```
$sudo docker stop welcome-to-docker
```

Tarea 4: Ejecutar una imagen Docker desde Docker Hub

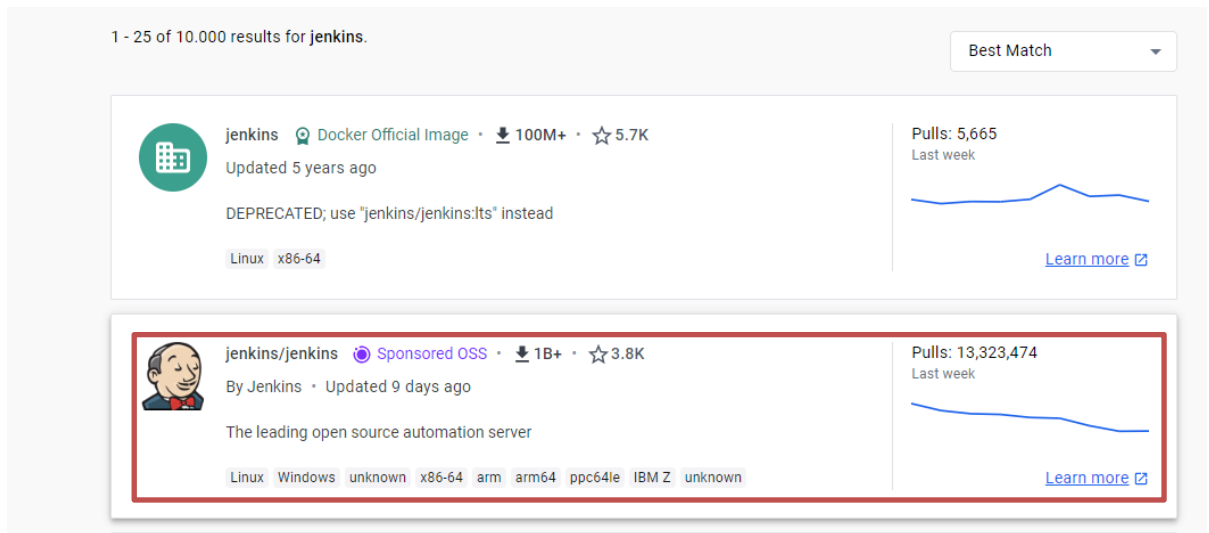
Docker Hub es el repositorio de imágenes para contenedores más grande del sector IT, ya que **aloja la mayor cantidad de imágenes oficiales de la plataforma**.

En esta ocasión, ejecutaremos una imagen de **Jenkins**. Jenkins es un servidor open source para la integración continua. Es una herramienta que se utiliza para compilar y probar proyectos de software.

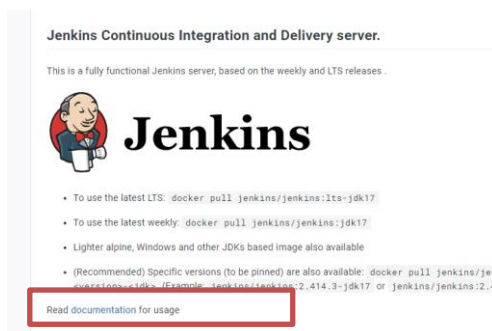
1. Ingresa a Docker Hub en el enlace <https://hub.docker.com/>
2. En la opción de búsqueda escribe **Jenkins** y presionar enter



3. Seleccione la imagen de Jenkins que se muestra en la figura:



4. En la página de descripción de la imagen de Jenkins, de click en documentación:



5. Copia el comando de ejecución de la imagen de Jenkins:



Jenkins

Usage

```
docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk17
```

NOTE: read the section [Getting started](#) below for the role of the `8080` port mapping.

6. Vuelva al Shell de su maquina virtual y ejecute el comando, no olvide que requiere **sudo**:

```
$sudo docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk17
```

La instalación se detendrá y generará un password, cópielo.

7. Para completar la instalación, abra su navegador, ingrese a http://ip_externa_vm:8080/, recuerde remplaza ip_externa_vm por la ip publica de su vm, y copie el password obtenido en el paso anterior y click en continuar.
8. Listo ya debe tener su servidor de integración Jenkins instalado.
9. Verifíquelo ingresando a http://ip_externa_vm:8080/.

El resultado debe ser:



¡Felicitaciones!

RECOMENDACIONES /
OBSERVACIONES

Ninguna