

# Contenedores Podman

**Administración de sistemas informáticos en red**

**-Proyecto Integrado de Grado Superior-**

**Hecho Por:** Francisco Javier Martín Núñez

# Sumario

1	Introducción.....	3
1.1	Descripción.....	3
2	Podman.....	4
2.1	Instalación.....	4
3	Podman-compose.....	5
3.1	Instalación.....	5
4	Skopeo.....	6
4.1	Instalación.....	6
5	Buildah.....	7
5.1	Instalación.....	7
6	Escenarios.....	8
7	Conclusión.....	8

# 1 Introducción

A continuación se va a pasar a explicar y ver los puntos que vamos a tratar en nuestro proyecto integrado y las herramientas de las que vamos a hacer uso.

Para poder ver mas en profundidad y ver los ejemplos de uso de dichas herramientas se puede encontrar mas información en mi web:

<https://franjavimn.onrender.com/proyecto-podman>

## 1.1 Descripción

En el siguiente proyecto integrado vamos a hacer uso del motor de contenedores llamado **Podman** y de tres herramientas complementarias a este como son **Skopeco**, **Buildah** y **Podman-compose**.

Con las herramientas anteriormente dichas vamos a proceder a explicar en que consiste el proyecto:

- Instalación de las distintas herramientas en maquinas creadas con **Vagrant**
- Creación y gestión de contenedores con **Podman**
  - Creación de un contenedor con la imagen **“hello-world”**.
- Despliegue de **pods** con varios contenedores gracias a **podman**
  - Despliegue de un **pod** con dos contenedores, una base de datos **MySQL** y la aplicación **Bookmedik** y el despliegue de una aplicación **Wordpress** junto a su base de datos **MySQL**.
- Uso de **podman-compose** para el despliegue de escenarios creados mediante ficheros **docker-compose.yml**
  - Uso de los ficheros **docker-compose.yml** que tenemos en el repositorio del proyecto.
- Obtener información de imágenes que se encuentren en registros locales o públicos como **DockerHub** mediante **Skopeco**.
  - Mostrar la funcionalidad de mostrar la información de las imágenes del registros de **DockerHub**.
- Creación y gestión de imágenes con la herramienta de **Buildah**.
  - Mostrar como se crear la imagen con **bookmedik**, que vamos a usar en uno de los despliegues que tenemos.

## 2 Podman

**Podman** es un motor de contenedores similar a **Docker**, que es su rival mas cercano, pero este tiene unas diferencias notables respecto a **Docker** que lo hacen una buena alternativa para el uso de **Podman** en vez de **Docker**:

- **Podman** es **rootless**, es decir, no es necesario que nuestro usuario tenga permisos de **superusuario** para gestionar los contenedores mediante .
- **Podman** es mas liviano que **Docker** ya que este carece de un *demonio central* para cada uno de los contenedores, en su lugar, han descentralizado todos los componentes necesarios para la gestión de contenedores y los han individualizado en componentes más pequeños que se utilizarán solo cuando sean necesarios.
- **Podman** nos da la posibilidad de la creación de **pods** de uno o varios contenedores, tomando como referencia a la forma en que lo hace **Kubernetes**
- La sintaxis que usa **Podman** es la misma que tiene **Docker**, por lo que la migración desde **Docker** es muy sencilla.

### 2.1 Instalación

Para la instalación de **Podman** vamos a seguir la documentación oficial para la instalación en una maquina virtual creada con **Vagrant** y **Debian 10**:

```
#### Debemos de habilitar en espacio de nombre de root ####
echo 'kernel.unprivileged_userns_clone=1' > /etc/sysctl.d/00-local-usersns.conf
systemctl restart procps

#### Añadimos los repositorios de backports de Debian 10 ####
echo 'deb http://deb.debian.org/debian buster-backports main' >>
/etc/apt/sources.list
echo 'deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/ /' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list

#### Descargamos e instalamos la clave de los repositorios ####
curl -L
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/Release.key | sudo apt-key add -

#### Actualizamos la lista de paquetes e instalamos dependencias desde backports
####
sudo apt update
sudo apt -y -t buster-backports install libseccomp2

#### Instalamos podman ####
sudo apt -y install podman
```

Si nuestro sistema es diferente a Debian 10 en la [documentación oficial](#) podemos encontrar la forma de instalarlo en nuestro sistema.

## 3 Podman-compose

Al igual que con los contenedores **Docker** tenemos la herramienta de **docker-compose** que nos permite desplegar escenarios de contenedores mediante un fichero `.yaml`, también tenemos la misma herramienta para **Podman** llamada **podman-compose**.

**Podman-compose** es un proyecto que es definido por sus propios desarrolladores como **un script para usar ficheros docker-compose.yml con podman**. Esta herramienta es un poco distinta a la hora de desplegar los escenarios ya que esta usa la utilidad de **Podman** de crear **Pods** con varios contenedores.

### 3.1 Instalación

La instalación se puede hacer de distintas maneras y podemos ver las distintas formas de instalar esta herramienta en múltiples distribuciones de linux, así bien, yo lo instalare en una maquina con Debian 10, lo podemos ver [aquí](#):

```
#### En una maquina Debian 10 instalamos lass herramientas para manejar paquetes python ####
vagrant@podman:~$ sudo apt install python3-pip

#### Una vez instalados, instalamos las dependencias de podman-compose ####
vagrant@podman:~$ pip3 install pyaml
Collecting pyaml

#### Instalamos desde los repositorios de git el paquete de podman-compose ####
vagrant@podman:~$ pip3 install
https://github.com/containers/podman-compose/archive/devel.tar.gz
```

## 4 Skopeo

Si queremos ver que tipo de imágenes tenemos en estos registros, ver si la imagen existe, obtener información sobre una imagen... Para ello tenemos la herramienta de **Skopeo** que es una utilidad línea de comandos que realiza varias operaciones en imágenes de contenedores y registros de imágenes.

Una de las principales ventajas que tiene **Skopeo** es que puede ser ejecutado sin ser usuario root al igual que pasa con podman y también, como característica principal es que no tiene ningún *demonio* ejecutándose para su funcionamiento.

Como característica sobre la compatibilidad de Skopeo con imágenes, este es compatibles con imágenes de tipo **OCI**(Open Container Initiative) y con imágenes de docker v2.

### 4.1 Instalación

Para ello, en **Debian 10** aun no se encuentra en repositorios el paquete de **Skopeo** que es el que queremos instalar, para ello debemos de añadir los repositorios de **backports**:

```
##### Añadimos los repositorios de backports #####
root@skopeo:/home/vagrant# echo 'deb http://deb.debian.org/debian buster-
backports main' >> /etc/apt/sources.list
root@skopeo:/home/vagrant# echo 'deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/ /' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list

##### Descargamos la clave de los repositorios #####
(Instalar si no se tiene el paquete de gnupg)
vagrant@skopeo:~$ sudo apt install gnupg

vagrant@skopeo:~$ curl -L
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/Release.key | sudo apt-key add -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  1093    100  1093    0     0   3149     0  --:--:--  --:--:--  --:--:--  3140
OK

##### Actualizamos la lista de paquetes y descargamos Skopeo #####
vagrant@skopeo:~$ sudo apt update

vagrant@skopeo:~$ sudo apt install -t buster-backports skopeo
```

De esta forma ya tendríamos **Skopeo** en nuestra maquina de pruebas instalado, si queremos instalarlo en otra distribución que no sea Debian podemos seguir los pasos del siguiente [enlace](#).

## 5 Buildah

**Buildah** es una herramienta de línea de comandos que nos permite la *construcción* de imágenes para contenedores del tipo **OCI**(Open Container Initiative).

**Buildah** es una herramienta de creación que le permite controlar cómo se disponen las capas de imágenes y se accede a los datos durante las creaciones. Para utilizar este programa, no se requiere ser el usuario **root** por lo que, al igual que ocurre con **podman**, es una herramienta **rootless** y podemos usarla siendo usuarios normales del sistema.

### 5.1 Instalación

Para ello lo instalamos en una maquina con un sistema Debian 10:

```
#### Añadimos los repositorios de backports ####
root@skopeo:/home/vagrant# echo 'deb http://deb.debian.org/debian buster-
backports main' >> /etc/apt/sources.list
root@skopeo:/home/vagrant# echo 'deb
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/ //' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list

#### Descargamos la clave de los repositorios ####
(Instalar si no se tiene el paquete de gnupg)
vagrant@skopeo:~$ sudo apt install gnupg

vagrant@skopeo:~$ curl -L
https://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/
Debian_10/Release.key | sudo apt-key add -
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  1093  100  1093    0     0   3149     0  --:--:--  --:--:--  --:--:--  3140
OK
#### Actualizamos la lista de paquetes y descargamos Buildah ####
vagrant@buildah:~$ sudo apt update

vagrant@buildah:~$ sudo apt install -t buster-backports Buildah
```

Si queremos instalarlo en otra distribución podemos consultar la documentación oficial [aquí](#).

## 6 Escenarios

Para ello vamos a seguir los ejemplos que se encuentran en mi web:

- Instalación de las distintas herramientas en maquinas creadas con **Vagrant**
- Creación y gestión de contenedores con **Podman**
  - Creación de un contenedor con la imagen “**hello-world**”. [Enlace](#)
- Despliegue de **pods** con varios contenedores gracias a **podman**
  - Despliegue de un **pod** con dos contenedores, una base de datos **MySQL** y la aplicación **Bookmedik** y el despliegue de una aplicación **Wordpress** junto a su base de datos **MySQL**. [Enlace](#)
- Uso de **podman-compose** para el despliegue de escenarios creados mediante ficheros **docker-compose.yml**
  - Uso de los ficheros **docker-compose.yml** que tenemos en el repositorio del proyecto. [Enlace](#)
- Obtener información de imágenes que se encuentren en registros locales o públicos como **DockerHub** mediante **Skopeo**.
  - Mostrar la funcionalidad de mostrar la información de las imágenes del registros de **DockerHub**. [Enlace](#)
- Creación y gestión de imágenes con la herramienta de **Buildah**.
  - Mostrar como se crear la imagen con **bookmedik**, que vamos a usar en uno de los despliegues que tenemos. [Enlace](#)

## 7 Conclusión

Como hemos visto, el conjunto de herramientas que son **Podman**, **Buildah**, **Podman-compose** y **Skopeo** es una gran alternativa a Docker y la curva de aprendizaje que tiene es muy baja y mas si venimos de usar Docker ya que la nomenclatura que tiene es la misma que tiene Docker.

En mi opinión lo veo una gran alternativa a usar Docker por el hecho de poder crear pods siguiendo la filosofía de Kubernetes y, gracias a la herramienta de podman-compose, poder generar escenarios con los ficheros docker-compose.yml me parece que puede llegar a tener un uso brutal a la hora de desplegar escenarios.

El problema principal que yo veo es la poca atención que se le da a este proyecto teniendo, bajo mi punto de vista, un grandísimo potencial. También cabe recalcar que el proyecto de podman-compose no es oficial, por lo que su desarrollo queda en manos de la comunidad lo que puede hacer que no sea del todo fiable.