

# Podman

Tobias Calvento, Federico Schvemler

Universidad Nacional de Quilmes

**Abstract.** Mini tutorial sobre el uso de Podman

## 1 Introduccion

Podman es un proyecto de codigo abierto que est disponible en la mayora de las plataformas Linux y reside en GitHub. Podman es un motor de contenedor sin daemon para desarrollar, administrar y ejecutar contenedores e imagenes de contenedores en un sistema Linux. Podman proporciona una lnea de comando compatible con Docker que simplemente puede usar el alias Docker cli, ‘alias docker = podman‘.

## 2 Capacidad de Podman

Los contenedores usados por Podman pueden ser ejecutados por root o por un usuario sin privilegios. Podman gestiona todo el ecosistema de contenedores que incluye pods, contenedores, imagenes de contenedores y volmenes de contenedores utilizando la biblioteca libpod. Podman se especializa en todos los comandos y funciones que lo ayudan a mantener y modificar las imagenes del contenedor OCI, como extraer y etiquetar. Le permite crear, ejecutar y mantener esos contenedores creados a partir de esas imagenes en un entorno de produccion.

El alcance de libpod y Podman es el siguiente:

- Admite multiples formatos de imagen, incluidos los formatos de imagen OCI y Docker.
- Soporte para multiples medios para descargar imagenes de forma segura, incluida la verificacin de confianza e imagen.
- Gestin de imagenes de contenedores (gestin de capas de imagenes, sistemas de archivos superpuestos, etc.).
- Gestin completa del ciclo de vida del contenedor.
- Soporte para pods para gestionar grupos de contenedores juntos.
- Aislamiento de recursos de contenedores y vainas.

### 3 Instalacion

El siguiente extracto explica como es la instalacion del mismo desde una pc con Ubuntu

```
sudo apt-get update -qq \  
sudo apt-get install -qq -y software-properties-common uidmap \  
sudo add-apt-repository -y ppa:projectatomic/ppa \  
sudo apt-get update -qq \  
sudo apt-get -qq -y install podman \  

```

Codigo para buildear de 0, en algunos casos la mejor opcion es hacerlo asi:

```
sudo apt-get install \  
btrfs-tools \  
git \  
golang-go \  
go-md2man \  
iptables \  
libassuan-dev \  
libc6-dev \  
libdevmapper-dev \  
libglib2.0-dev \  
libgpgme-dev \  
libgpg-error-dev \  
libostree-dev \  
libprotobuf-dev \  
libprotobuf-c-dev \  
libseccomp-dev \  
libselinux1-dev \  
libsystemd-dev \  
pkg-config \  
runc \  
uidmap
```

### 4 Troubleshooting

Si alguna dependencia no puede instalarse o no es lo suficientemente actual, debe construirse desde la fuente. Esto afectara principalmente a Debian, Ubuntu y las

distribuciones relacionadas, o RHEL donde no hay una suscripción activa (por ejemplo, Cloud VM).

#### 4.1 Ostree

Es necesaria una copia de las bibliotecas de desarrollo para ostree, ya sea en forma del paquete libostree-dev del PPA flatpak, o construido desde la fuente. A partir de Ubuntu 18.04, libostree-dev esta disponible en los repositorios principales y ya no se requiere el PPA.

```
git clone https://github.com/ostreedev/ostree ~/ostree
cd ~/ostree
git submodule update --init
sudo apt-get install -y automake bison e2fsprogs e2fslibs-dev
    fuse libfuse-dev libgpgme-dev liblzma-dev libtool zlib1g
./autogen.sh --prefix=/usr --libdir=/usr/lib64 --sysconfdir=/etc
sed -i '/.*--nonet.*d' ./Makefile-man.am
make
sudo make install
```

#### 4.2 golang

```
export GOPATH=~/go
git clone https://go.googlesource.com/go $GOPATH
cd $GOPATH
git checkout tags/go1.10.8 # optional
cd src
./all.bash
export PATH=$GOPATH/bin:$PATH
```

#### 4.3 common

Se espera que la última versión de common este instalada en el sistema. Common se usa para monitorear los tiempos de ejecución de OCI. Para construir desde la fuente, use lo siguiente:

```
git clone https://github.com/containers/common
cd common
make
sudo make podman
```

Fig. 1. Script de instalacion de Common

#### 4.4 runc

Se espera que la ltima versin de runc se instale en el sistema. Podman lo selecciona como "default runtime". La versin 1.0.0-rc4 es el requisito mnimo, que ya est disponible en Ubuntu 18.04. Para construir desde la fuente, use lo siguiente:

```
git clone https://github.com/opencontainers/runc.git
    $GOPATH/src/github.com/opencontainers/runc
$cd $GOPATH/src/github.com/opencontainers/runc
make BUILDTAGS="selinux seccomp"
sudo cp runc /usr/bin/runc
```

#### 4.5 CNI plugins

```
git clone https://github.com/containernetworking/plugins.git
    $GOPATH/src/github.com/containernetworking/plugins
cd $GOPATH/src/github.com/containernetworking/plugins
./build_linux.sh
sudo mkdir -p /usr/libexec/cni
sudo cp bin/* /usr/libexec/cni
```

#### 4.6 Setup CNI networking

```
sudo mkdir -p /etc/cni/net.d
curl -qsSL https://raw.githubusercontent.com/containers/libpod/master/cni/
87-podman-bridge.conflist sudo tee /etc/cni/net.d/99-loopback.conf —
```

## 4.7 Add configuration

```
sudo mkdir -p /etc/containers
sudo curl https://raw.githubusercontent.com/projectatomic/registries/master/registries.fedora -o /etc/containers/registries.conf

sudo curl https://raw.githubusercontent.com/containers/skopeo/master/default-policy.json -o /etc/containers/policy.json
```

## 5 Ejemplo util: Funbox

Usando el comando Podman pull wernight/funbox se puede acceder a un container de ejemplo el cual levanta una imagen que contiene commando de ascii, algunos de los de los commandos se pueden ver a continuacion

```

WERNIGHT/FUNBOX
-----
Basic usage
$ docker run --rm -it wernight/funbox

Screensavers / infinite animations:
1) 'aafire' - Fire pit
2) 'asciiquarium' - Aquarium
3) 'cacadem0' - caca-utils demo
4) 'cmatrix' - Matrix
5) 'falling-hearts' - Falling Hearts screensaver
6) 'nyancat' - Nyan cat
7) 'pipes' - Pipes screensaver
8) 'xaos' - real-time interactive fractal zoomer

Demos / short animations:
9) 'bb' - ASCII art demo
10) 'sl' - Train passing by
11) 'youtube' - Alias to watch YouTube video using 'youtube-dl', 'vlc' and 'caca'

Static ASCII art:
* Text formatting:
12) 'cowsay' - Talking cow (or actually many others, run with '-l' to get a list and via '-f <name>' to use one)
13) 'figlet' - ASCII large text print
14) 'toilet' - ASCII large text print
* Other:
15) 'aview' - Convert image to ASCII art
16) 'cacaview' - Convert image to ASCII art
17) 'boxes' - Put various frames around given block of ASCII art.
18) 'binlock' - Current time in binary.
19) 'linuxlogo' - Standard linux logos
20) 'lolcat' - Colorful 'cat'

Random text generator:
21) 'rig' - Kind of person ID
22) 'fortune' - Random fortune cookie message

```

Fig. 2. Opciones de Funbox

### 5.1 NyanCat

```
$ docker run --rm -it wernight/funbox nyancat
```



Fig. 3. Commando de Nyan