## Con RHEL 10

Alex Callejas Content Architect





# **Alex Callejas**

Content Architect @ Red Hat 🤚



@dark\_axl



github.com/AlexCallejas



t.me/FedoraMexico

Geek by nature, Linux by choice, Fedora of course!

# ¿Porqué un laboratorio?

# **A**CAUTION

3

HAVING TO RE-INSTALL AN ENTIRE SERVER JUST BECAUSE THE SOFWARE YOU ARE TESTING HAS HARDCODED THE IP REALLY STINKS. It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong. In that simple statement is the key to science. – Richard Feynman

#### Laboratorio de pruebas



Lenovo ThinkPad P1 Gen 5 12th Gen Intel(R) Core(TM) i7-12800H

Storage

 $/ \rightarrow$  LVM 953.87G

Fedora release 41 (Forty One)

#### 6.14.6-200.fc41.x86\_64



- Cuenta de acceso en developers.redhat.com
- Creación de Lab con KVM

Build a lab in five minutes with three simple commands



### Laboratorio de Pruebas

Creando el Lab

- Red Hat Enterprise Linux 10.0 KVM Guest Image
   qcow2 → https://access.redhat.com/downloads/content/rhel
- Configurando el lab

\$ virt-customize -a lab01.qcow2 --hostname lab01.rootzilopochtli.lab
--root-password password:redhat --ssh-inject 'root:file:labkey.pub'
--uninstall cloud-init --selinux-relabel

#### Instalando el lab

\$ sudo virt-install --name rhel10 --memory 8192 --vcpus 4 --disk
/var/lib/libvirt/images/lab01.qcow2 --import --os-info rhel10.0
--noautoconsole

Accesando al lab

\$ sudo virsh domifaddr rhel10



6

#### Contenedores

Anatomía



# containers = processes



### Contenedores

Docker vs Podman







8

### Podman rootless containers

#### Lo que debes saber

Algunas consideraciones al ejecutar containers como un usuario *non-root*:

 Las imágenes se guardan en el home directory del usuario (\$HOME/.local/share/containers/storage/), en lugar de /var/lib/containers.

Al no tener privilegios de **root**:

- No tiene capacidad para acceder a un puerto inferior a 1024.
- El almacenamiento debe estar en un sistema de archivos local.



## Podman rootless containers

#### Primeros pasos

- Registrar sistema (con la cuenta de acceso de developers.redhat.com)
  - # subscription-manager register
- Actualizar sistema
  - # dnf update
- Reiniciar sistema
  - # reboot
- Instalar herramientas de contenedores
  - # dnf -y install podman buildah skopeo
- Crear usuario y asignarle contraseña
  - # useradd usuario
  - # passwd usuario

### Podman rootless containers

Primeros pasos

Verificamos la configuración rootless

\$ podman unshare cat /proc/self/uid\_map

- Crear contenedor usando la Universal Base Image (UBI)
  - \$ podman pull ubi9/ubi
  - \$ podman inspect ubi9/ubi
  - \$ podman run ubi9/ubi cat /etc/os-release
- ← descarga
- $\leftarrow \text{inspecciona}$
- ← ejecuta



## Podman rootless containers

Creando un servicio

- Busca la imagen del servicio web (httpd)
  - \$ podman search httpd
- En caso de ser necesario, haz login en el repositorio (con la cuenta de acceso de developers.redhat.com)
  - \$ podman login registry.redhat.io
- Descarga la imagen y ejecuta el contenedor
  - \$ podman pull registry.redhat.io/rhel9/httpd-24
  - \$ podman images

- $\leftarrow$  lista la imagenes
- \$ podman inspect registry.redhat.io/rhel9/httpd-24
- \$ podman run httpd-24

## Podman rootless containers

Creando un servicio

\$ podman run --name myapache -d httpd-24

\$ ps auxf | grep httpd

\$ podman ps

\$ podman logs myapache

\$ podman inspect myapache | grep expose-services

\$ podman inspect myapache | grep IPAddress

\$ ss -tulpn

← modo desvinculado

 $\leftarrow \text{revisa procesos}$ 

 $\leftarrow \text{lista contenedores}$ 

 $\leftarrow \mathsf{muestra} \ \mathsf{logs}$ 

 $\leftarrow$  monitorea puertos

### Podman rootless containers

Creando un servicio

- \$ podman run --name myapache -d -p 8080:8080 httpd-24 ← mapeando puertos
- \$ ss -tulpn
- \$ curl localhost:8080
- \$ podman exec -ti myapache /bin/bash
- bash-5.1\$ echo "Top Secret" > secret.data
- \$ podman stop myapache
- \$ podman start myapache
- \$ podman rm myapache

- ← prueba el servicio
- $\leftarrow \text{acceder al contenedor}$
- $\leftarrow$  crea archivo
- $\leftarrow detener \ contened or$
- $\leftarrow \text{iniciar contenedor}$
- $\leftarrow$  eliminar contenedor

# Podman rootless containers

#### Creando un servicio

- \$ mkdir -p containers/myapache/var/www/html ← crea directorio
- \$ echo "Hola Mundillo" > containers/myapache/var/www/html/index.html
- Mapea el directorio al contenedor
  - \$ podman run --name myapache -d -p 8080:8080 -v
  - ~/containers/myapache/var/www/html:/var/www/html:Z httpd-24
  - \$ curl localhost:8080

← prueba el servicio

\$ podman rmi c769fcaa18ed

 $\leftarrow$  elimina la imagen



# Podman Pods

#### Lo que necesitas saber

"Un **Pod** es un grupo de uno o más contenedores, con recursos de almacenamiento y red compartidos, además una especificación de cómo ejecutar dichos contenedores.

#### - PODS - KUBERNETES DOCUMENTATION





# Podman Pods

#### Ejemplo práctico: MariaDB Pod

- Crear el pod
  - \$ podman run -dt -e MYSQL\_ROOT\_PASSWORD=x --pod new:db registry.fedoraproject.org/f31/mariadb:latest
- Revisa el status del pod
  - \$ podman pod ps
- Agrega un contenedor al pod y verifica el pod
  - \$ podman run -it --rm --privileged --pod db docker.io/library/alpine:latest /bin/sh
  - / # apk add mariadb-client
  - / # mysql -u root -P 3306 -h 127.0.0.1

# Podman

#### Siguientes pasos $\rightarrow$ Kubernetes

- Crear contenedor de ejemplo
  - \$ podman run -dt -p 8000:80 --name demo quay.io/libpod/alpine\_nginx:latest
  - \$ curl localhost:8000
- Genera un snapshot como YAML para kubernetes
  - \$ podman generate kube demo > demo.yml
- Transfierelo y cargalo en kubernetes
  - # oc create -f demo.yml -n myproject
  - # oc status --suggest

In project My Project (myproject) on server https://minishift.rootzilopochtli.lab:8443





# Referencias

#### Documentación y enlaces

Building, running, and managing containers

Using Podman, Buildah, and Skopeo on Red Hat Enterprise Linux 10

- Podman basics: Resources for beginners and experts
- Podman can now ease the transition to Kubernetes and CRI-O
- Containers

Open Repository for Container Tools

- podman
- Daniel Walsh @rhatdan



19



# **Gracias!**

Red Hat es el proveedor líder mundial de soluciones empresariales de software de código abierto. Sus galardonados servicios de soporte, formación y consultoría hacen de Red Hat un asesor de confianza para las empresas Fortune 500.



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



