Infrared

and the Fantabulous Emancipation of One OpenStack Lab

Alex Callejas

Senior Technical Support Engineer

March 2020

Red Hat

Alex Callejas
Senior Technical Support Engineer @Red Hat

@dark_axl

/rootzilopochtli

www.rootzilopochtli.com

darkaxl017.fedorapeople.org/slides/
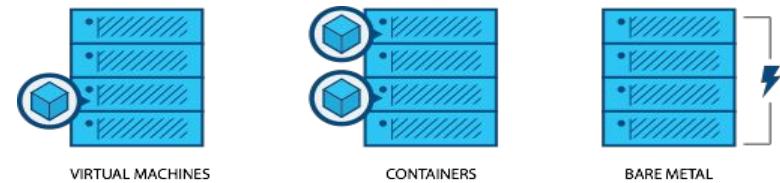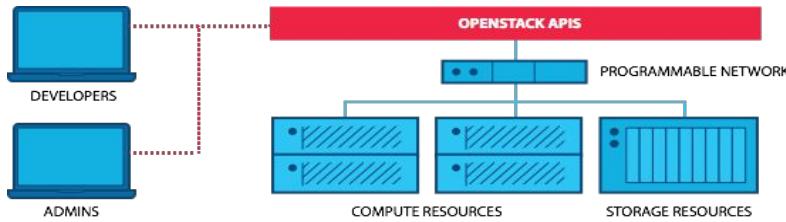
Geek by nature, Linux by choice, Fedora of course!

Red Hat

DISCLAIMER

- Esta presentación es realizada con fines **estrictamente** educativos

- El método presentado se encuentra en constante desarrollo

- La infraestructura de laboratorio generada **no está soportada**, incluyendo la que contempla el uso de suscripciones

- Bajo ninguna condición se recomienda el uso en ambientes diferentes al de laboratorio de pruebas

- El presentador se deslinda de cualquier responsabilidad sobre las decisiones tomadas como resultado de esta presentación

What is OpenStack?

Infraestructura programable que presenta un conjunto común de API's, además de recursos de cómputo, redes y almacenamiento para proporcionar una infraestructura como servicio en la nube (IaaS)

**Una plataforma** para máquinas virtuales, contenedores y baremetal

Source: https://www.openstack.org/marketing

Red Hat
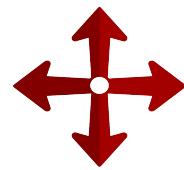
What is OpenStack used for?
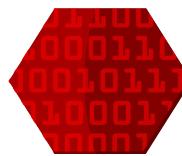
NETWORK FUNCTION VIRTUALIZATION
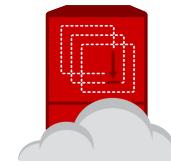
DEVELOPER CLOUD

EDGE COMPUTING

AI/ML/HPC

SERVICE PROVIDER
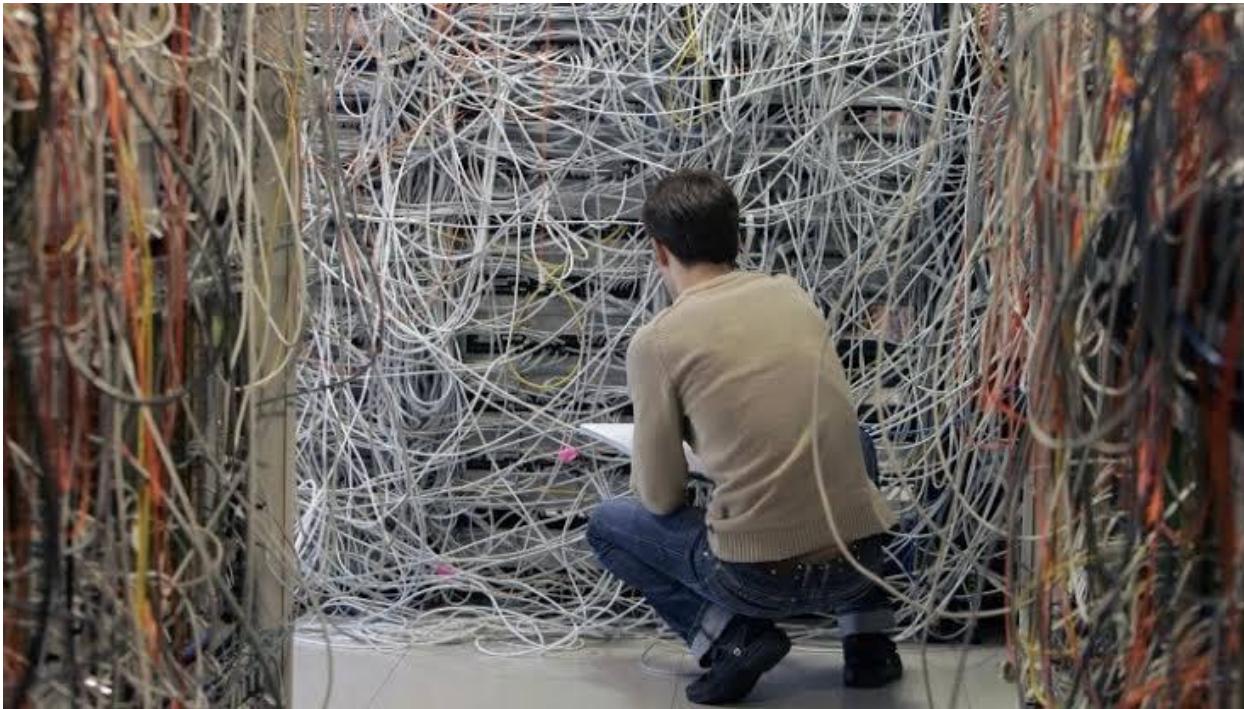
En serio, ¿qué es OpenStack?

Source:http://vmartinezdelacruz.com/in-a-nutshell-how-openstack-works/

En serio, ¿qué es OpenStack?

Source:http://vmartinezdelacruz.com/in-a-nutshell-how-openstack-works/
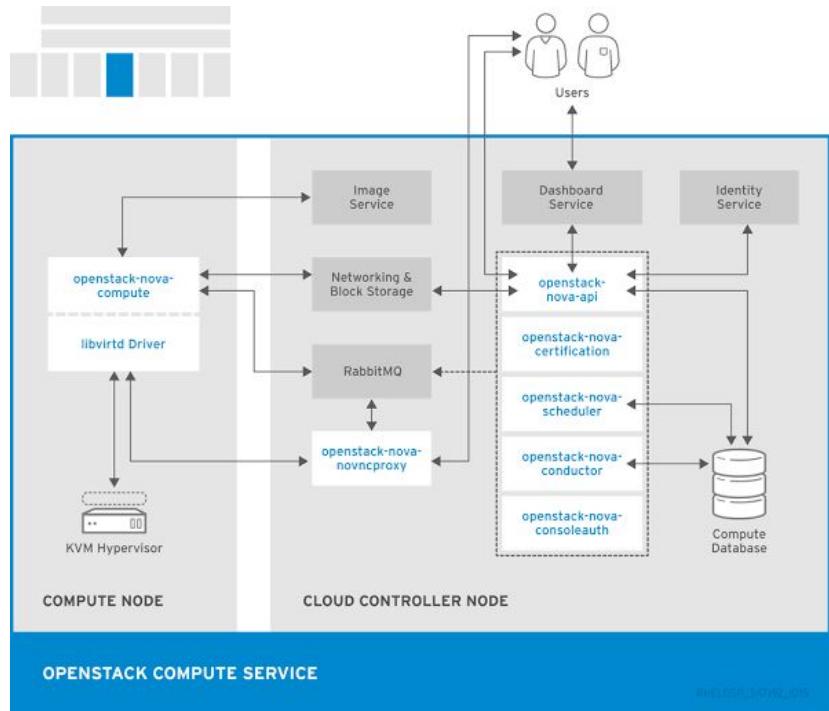
En serio, ¿qué es OpenStack?

Tipos de nodos:

- ▶ Controller[1]
  - · DB's y API's
- ▶ Compute[1]
  - · Hypervisor
- ▶ Storage[2]

[1] Infraestructura mínima soportada: 3 controllers, 2 computes

[2] Opcional

Test Deployments

Create a proof of concept cloud
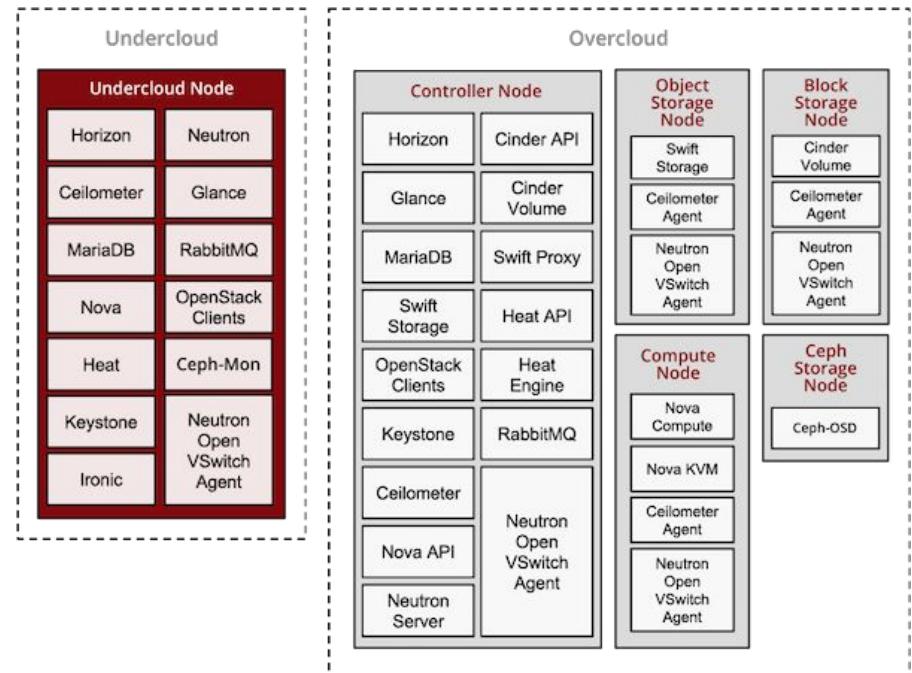
Deploy a production Cloud

TripleO

TripleO se refiere a "**O**penStack **O**n **O**penStack"
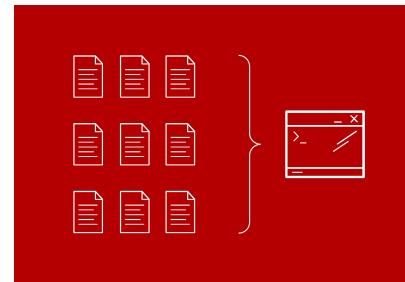
▸ Objetivo oficial del proyecto OpenStack

· Capacidad para desplegar y administrar la nube de producción en bare-metal hardware utilizando un subconjunto de componentes de OpenStack

Infrared

InfraRed is a plugin based system that aims to provide an easy-to-use CLI for Ansible based projects. It aims to leverage the power of Ansible in managing / deploying systems, while providing an alternative, fully customized, CLI experience that can be used by anyone, without prior Ansible knowledge.

The project originated from Red Hat OpenStack infrastructure team that looked for a solution to provide an "easier" method for installing OpenStack from CLI but has since grown and can be used for *any* Ansible based projects.

Source: https://infrared.readthedocs.io/
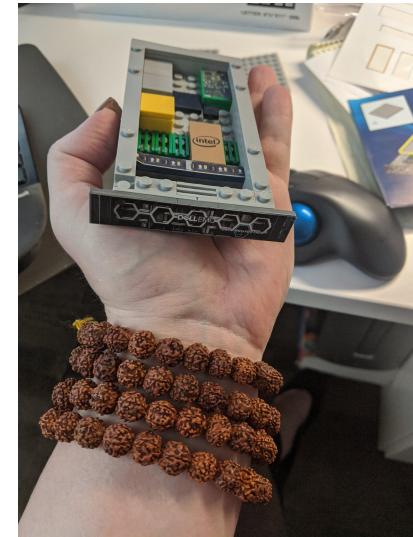
Infrared

Requerimientos

- ► Un **servidor físico** con al menos:
  - · **48 GB** de RAM
  - · Disco duro de **150 GB**
    - · Puede sobrevivir con *70 GB* si se usa la compresión *btrfs* y *zlib* para `/var/lib/libvirt/images`
  - · Virtualización de CPU habilitada en BIOS / UEFI
- ► Fedora Server **>25**, CentOS o RHEL **7** con:
  - · `git gcc libffi-devel openssl-devel python-virtualenv libselinux-python redhat-rpm-config screen`

Source Photo: https://twitter.com/base2wave/status/1197223104457715712

Dell PowerEdge R820
Intel(R) Xeon(R) CPU E5-4603 0 @ 2.00GHz
32052 MB memory, 292 GB disk space

Lab deployment

Instalar infrared:

```
git clone https://github.com/redhat-openstack/infrared.git

cd infrared

virtualenv .venv && source .venv/bin/activate

pip install --upgrade pip

pip install --upgrade setuptools

pip install --upgrade infrared

pip install .

echo ". $(pwd)/etc/bash_completion.d/infrared" >> ${VIRTUAL_ENV}/bin/activate

infrared plugin list
```

Python 3 Support: https://github.com/redhat-openstack/infrared/issues/370

Lab deployment

Recomendaciones

▸ **screen** y/o **tlog**

  · http://www.rootzilopochtli.com/2019/02/tlog-grabar-sesiones-en-la-terminal/

▸ **time**

▸ Destruir VM's y restaurar la configuración original

  · kill
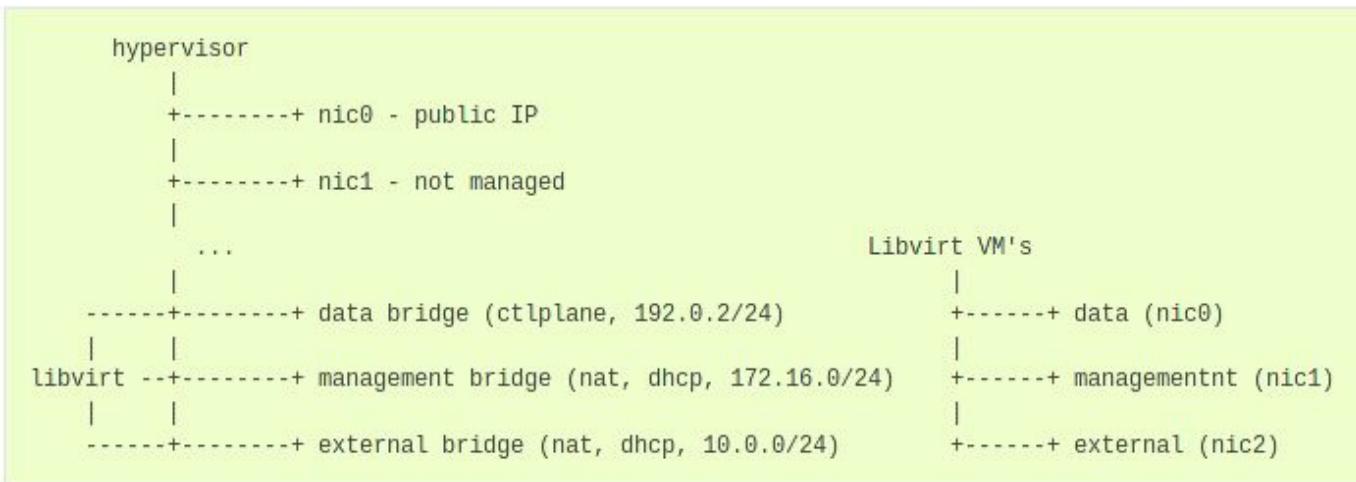
```
time infrared virsh -v -o cleanup.yml --host-address $HOST --host-key $HOST_KEY --kill yes
```

  · cleanup

```
time infrared virsh --cleanup True --host-address $HOST --host-key $HOST_KEY
```

Lab deployment

Network layout:


```
        hypervisor
             |
             +--------+ nic0 - public IP
             |
             +--------+ nic1 - not managed
             |
             ...                                          Libvirt VM's
             |                                                 |
     ------+--------+ data bridge (ctlplane, 192.0.2/24)      +------+ data (nic0)
       |     |                                                 |
libvirt --+--------+ management bridge (nat, dhcp, 172.16.0/24)  +------+ managementnt (nic1)
       |     |                                                 |
     ------+--------+ external bridge (nat, dhcp, 10.0.0/24)    +------+ external (nic2)
```

Lab deployment

Infra deployment:

```
time infrared virsh -v --host-memory-overcommit True \
--host-address $HOST --host-key $HOST_KEY --host-user root \
--topology-nodes "undercloud:1,controller:3,compute:2" \
-e override.controller.cpu=2  -e override.controller.memory=8192 \
-e override.controller.disks.disk1.size=60G -e override.compute.disks.disk1.size=60G \
-e override.compute.memory=16384 -e override.undercloud.memory=16384 \
--image-url https://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud.qcow2
```

Lab deployment

RDO

Instalar undercloud (director):

```
time infrared tripleo-undercloud --version queens --images-task build  --ssl no
```

Si el tamaño del lab es menor a 64 Gb de RAM, se recomienda reducir los *workers* a 1.
Se puede utilizar el script:

```
curl -L -O https://raw.githubusercontent.com/mrVectorz/snips/master/osp/low_memory_uc.sh
```

Red Hat

Lab deployment

Instalar overcloud (controllers y computes):

```
time infrared tripleo-overcloud --version queens --deployment-files virt \
--introspect yes --tagging yes --deploy yes \
--containers yes --registry-mirror docker.io --registry-namespace tripleoqueens \
--registry-tag current-tripleo-rdo --registry-prefix='centos-binary-' \
--registry-skip-puddle yes
```

Toma 2:

```
--overcloud-templates sahara,disable-telemetry,docker-queens_image
```

Lab deployment

Red Hat
OpenStack Platform

Infra deployment:

```
--image-url http://hakke.rootzilopochtli.com/rhel-guest-image-7-6-210-x86-64-qcow2
```

Instalar undercloud (director) y reducir workers:

```
time infrared tripleo-undercloud --build GA --version 10 --images-task rpm --ssl true --cdn cdn_creds.yml
```

```
curl -L -O https://raw.githubusercontent.com/mrVectorz/snips/master/osp/low_memory_uc.sh
```

Red Hat

Lab deployment

Red Hat
OpenStack Platform

cdn_creds.yml:

```
username: rhn_user
password: **********
pool_ids: 8a85f99a6d1c4218016e3c1611ab0a32
```

Instalar overcloud (controllers y computes):

```
time infrared tripleo-overcloud --deployment-files virt --version 10 \
--introspect yes --tagging yes --postreboot yes --deploy yes \
--network-backend vxlan --overcloud-ssl true
```

Red Hat

Deployment test

- create a few networks

- create respective subnets

- create routers

- add router ports to the router

- [1]download cirros guest image

  - http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img

- download centos guest image

  - https://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud.qcow2

- create images and flavors

- create security groups

- create instances

- boot a couple instances

[1] OpenStack Images:https://docs.openstack.org/image-guide/obtain-images.html

Red Hat

Lab customization

Personalizar infra domain:

```
plugins/tripleo-undercloud/hypervisor.yml:    {{ item }}.redhat.local {{ item }}"

plugins/tripleo-undercloud/init.yml:          name: "{{ inventory_hostname }}.redhat.local"

plugins/tripleo-undercloud/init.yml:

        line: "127.0.0.1   {{ inventory_hostname }}.redhat.local {{ inventory_hostname }}"

plugins/tripleo-undercloud/plugin.spec:       default: 'redhat.local'

plugins/tripleo-undercloud/restore.yml:       {{ groups['undercloud'][0] }}.redhat.local \
```

Red Hat

Horizon Dashboard

▸ Localizar las IP's de los nodos en undercloud:

```
[stack@undercloud-0 ~]$ openstack server list

+--------------------------------------+--------------+--------+------------------------+----------------+
| ID                                   | Name         | Status | Networks               | Image Name     |
+--------------------------------------+--------------+--------+------------------------+----------------+
| 6f7ed9fd-1a3a-46e8-940b-a0e3555f6273 | controller-2 | ACTIVE | ctlplane=192.168.24.6  | overcloud-full |
| 183ce9a1-73d8-4782-8d92-79b6e52ccf6f | compute-0    | ACTIVE | ctlplane=192.168.24.12 | overcloud-full |
| 37c7df4e-2c75-424e-a41b-909bb0b623ef | controller-1 | ACTIVE | ctlplane=192.168.24.17 | overcloud-full |
| 6bf8c568-b53b-4c0f-8fd1-247857cb4529 | compute-1    | ACTIVE | ctlplane=192.168.24.15 | overcloud-full |
| 7a51c36b-3feb-4f36-8d70-51ca2114e859 | controller-0 | ACTIVE | ctlplane=192.168.24.16 | overcloud-full |
+--------------------------------------+--------------+--------+------------------------+----------------+
```

▸ Localizar la url del dashboard en algún controller::

```
[heat-admin@controller-0 ~]$ grep -A2 horizon /etc/haproxy/haproxy.cfg |grep bind\ 10

  bind 10.0.0.101:443 transparent ssl crt /etc/pki/tls/private/overcloud_endpoint.pem

  bind 10.0.0.101:80 transparent
```

Horizon Dashboard

▸ Configurar `~/.ssh/config`:

```
# openstack@home infrared
Host compute-1.redhat.local
        HostName 192.168.24.15
Host compute-0.redhat.local
        HostName 192.168.24.12
Host controller-2.redhat.local
        HostName 192.168.24.6
Host controller-1.redhat.local
        HostName 192.168.24.17
Host controller-0.redhat.local
        HostName 192.168.24.16
        DynamicForward localhost:65432
```

```
Host undercloud-0.redhat.local
        User stack
        HostName undercloud-0
        ForwardAgent yes
Host *.redhat.local
        User heat-admin
        ProxyJump dell-per820-2
        IdentityFile /root/.ssh/id_rsa
```

Horizon Dashboard
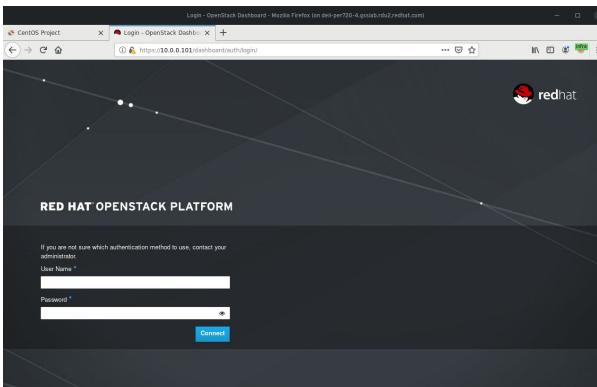
▶ Crear tunel ssh:

```
[root@dell-per820-2 ~]# ssh -f -N controller-0.redhat.local
```

▶ Configuración de Socks Proxy:

```
CURLOPT_PROXY=socks5://localhost:65432

ALL_PROXY=socks5h://localhost:65432 curl -vi 10.0.0.101:80/dashboard
```
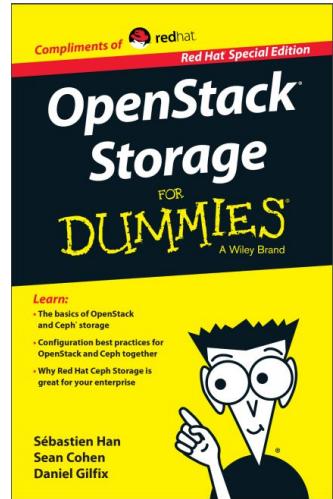
Hybrid deployment

https://infrared.readthedocs.io/en/latest/hybrid_deployment.html

Referencias

- https://www.redhat.com/cms/managed-files/st-openstack-storage-for-dummies-9781119292531-201606-en.pdf

- https://access.redhat.com/documentation/en/red_hat_openstack_platform

- https://infrared.readthedocs.io/

- https://github.com/mrVectorz/infrared-lab

- https://www.vkhitrin.com/blog/clize-your-ansible-playbooks-using-infrared/

- https://herve.beraud.io/virtualization/linux/openstack/infrared/environment/2019/02/05/prepare-environment-to-use-red-hat-infrared.html

- https://herve.beraud.io/virtualization/ssh/forwarding/horizon/openstack/infrared/2019/03/06/setup-ssh-port-forwarding-to-reach-horizon-openstack-dashboard-deployed-with-infrared.html

- http://www.tuxfixer.com/create-tenant-in-openstack-newton-using-command-line-interface/

Comunidades en México

Únete!

- **Fedora México**

  ○ https://www.meetup.com/es-ES/Fedora-Mexico/

  ○ https://t.me/fedoramexico

  ○ https://fedoramx.fedorapeople.org/

- **SysArmy México**

  ○ https://www.meetup.com/es/Sysarmy-Mexico/

  ○ https://t.me/sysarmymx

Gracias

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

▶ youtube.com/user/RedHatVideos

f facebook.com/redhatinc

🐦 twitter.com/RedHat

🎩 Red Hat