

Configuración de red entre Raspberry Pi y VM

Administración de Sistemas Operativos

1^a Evaluación - RA 2 - CE g

Unidad Didáctica 1:

Redes y virtualización

Alvaro Vazquez Vazquez

23 de octubre de 2025

I.E.S. Fernando Aguilar Quignon

C/Conil de la Frontera, 3

CP 11010, Cádiz

Índice

1. Introducción	2
2. Configuración de red	2
2.1. Creación de un puente (bridge) en la máquina host	2
2.2. Configuración de red en la VM (Netplan)	3
2.3. Configuración de QEMU para usar el bridge	4
2.4. Configuración persistente con systemd-networkd	5
3. Configuración de K3s	6
3.1. Instalación y configuración del nodo Master (VM)	6
3.2. Unión de la Raspberry Pi como nodo Agent (Slave)	6
4. Instalación del Dashboard de Kubernetes	7
4.1. Despliegue del Dashboard	7
4.2. Creación del usuario administrador	7
4.3. Generación del token de acceso	8
4.4. Acceso al Dashboard	8
5. Conclusión	8
6. Bibliografía	9

1. Introducción

Este documento describe la configuración de red entre una máquina host, una VM QEMU y una Raspberry Pi para implementar un clúster con K3s. Se incluyen los comandos necesarios para crear un **bridge** en la máquina host y habilitar el acceso a Internet desde la VM, así como la preparación para que la Raspberry actúe como nodo **slave**.

2. Configuración de red

2.1. Creación de un puente (bridge) en la máquina host

Pasos para crear br0 y asociar la interfaz física enp5s0:

Listing 1: Creación del bridge br0

```
1 sudo ip link add name br0 type bridge
2 sudo ip link set enp5s0 master br0
3 sudo ip link set br0 up
4 sudo ip addr add 192.168.1.33/24 dev br0
5 sudo ip route add default via 192.168.1.1
6 sudo ip addr flush dev enp5s0
```

Verificación de interfaces:

Listing 2: Verificación de interfaces

```
1 ip a show enp5s0
2 ip a show br0
3 ping -c 3 8.8.8.8
```

Notas importantes:

- La interfaz **enp5s0** queda asociada al bridge **br0**.
- **br0** toma la IP de la máquina host para proveer conectividad a la VM.
- Se comprueba la conectividad a Internet mediante ping a un DNS público.

2.2. Configuración de red en la VM (Netplan)

En la VM de QEMU se configura la red de forma estática mediante Netplan. El archivo `/etc/netplan/00-installer-config.yaml` tiene la siguiente configuración:

Listing 3: Archivo de Netplan de la VM QEMU

```
1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     ens3:
6       dhcp4: no
7       addresses: [192.168.1.101/24]
8       gateway4: 192.168.1.33
9       nameservers:
10        addresses: [8.8.8.8, 8.8.4.4]
```

Captura de pantalla:

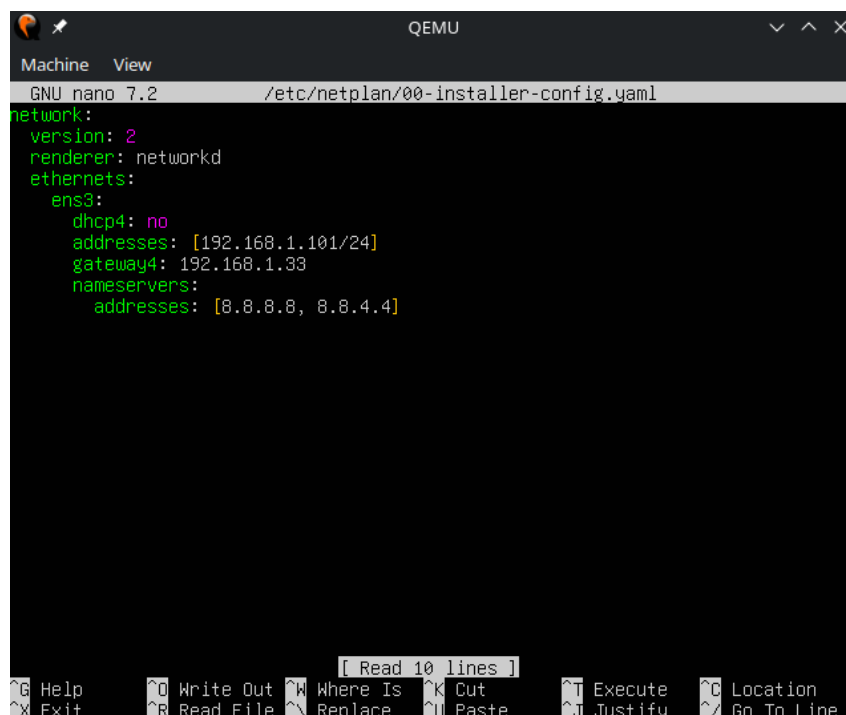


Figura 1: Configuración de Netplan en la VM QEMU

2.3. Configuración de QEMU para usar el bridge

Archivo de script `qemuUS.sh`:

Listing 4: Script para lanzar VM con QEMU usando br0

```
1 sudo qemu-system-x86_64 \  
2   -enable-kvm \  
3   -m 4096 \  
4   -smp 2 \  
5   -cpu host \  
6   -hda /home/archi/Downloads/ubuntu-server.qcow2 \  
7   -boot d \  
8   -netdev bridge,id=net0,br=br0 \  
9   -device virtio-net-pci,netdev=net0 \  
10  -vga virtio
```

Explicación:

- `-netdev bridge,id=net0,br=br0` conecta la VM al bridge creado.
- `-device virtio-net-pci,netdev=net0` proporciona la interfaz virtual en la VM.

2.4. Configuración persistente con systemd-networkd

Archivo `/etc/systemd/network/br0.netdev`:

```
1 [NetDev]
2 Name=br0
3 Kind=bridge
```

Archivo `/etc/systemd/network/enp5s0.network`:

```
1 [Match]
2 Name=enp5s0
3
4 [Network]
5 Bridge=br0
```

Archivo `/etc/systemd/network/br0.network`:

```
1 [Match]
2 Name=br0
3
4 [Network]
5 Address=192.168.1.33/24
6 Gateway=192.168.1.1
7 DNS=8.8.8.8
```

Habilitar servicios:

```
1 sudo systemctl enable --now systemd-networkd
2 sudo systemctl enable --now systemd-resolved
```

3. Configuración de K3s

3.1. Instalación y configuración del nodo Master (VM)

Instalación de K3s en la VM Master:

```
1 curl -sfL https://get.k3s.io | sh -
2 sudo k3s kubectl get nodes
```

Obtención del token para unir nodos:

```
1 sudo cat /var/lib/rancher/k3s/server/node-token
```

3.2. Unión de la Raspberry Pi como nodo Agent (Slave)

Comando para unir la Raspberry al Master:

```
1 curl -sfL https://get.k3s.io | K3S_URL=https://192.168.1.101:6443
   K3S_TOKEN=<TOKEN_DEL_MASTER> sh -
```

Verificación de nodos en el Master:

```
1 sudo k3s kubectl get nodes
2 # Deberías ver algo así:
3 # NAME      STATUS    ROLES                  AGE     VERSION
4 # k3s       Ready    control-plane,master   42h     v1.33.5+k3s1
5 # raspberry Ready    <none>                 5s      v1.33.5+k3s1
```

4. Instalación del Dashboard de Kubernetes

4.1. Despliegue del Dashboard

Aplicación del manifiesto oficial (v2.7.0) en el nodo Master:

```
1 sudo k3s kubectl apply -f https://raw.githubusercontent.com/
   kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```

Salida esperada:

```
1 namespace/kubernetes-dashboard created
2 serviceaccount/kubernetes-dashboard created
3 service/kubernetes-dashboard created
4 ...
5 deployment.apps/dashboard-metrics-scraper created
```

4.2. Creación del usuario administrador

Archivo dashboard-admin.yaml:

```
1 apiVersion: v1
2 kind: ServiceAccount
3 metadata:
4   name: admin-user
5   namespace: kubernetes-dashboard
6 ---
7 apiVersion: rbac.authorization.k8s.io/v1
8 kind: ClusterRoleBinding
9 metadata:
10  name: admin-user
11 roleRef:
12  apiGroup: rbac.authorization.k8s.io
13  kind: ClusterRole
14  name: cluster-admin
15 subjects:
16 - kind: ServiceAccount
17   name: admin-user
18   namespace: kubernetes-dashboard
```

Aplicación del usuario administrador:

```
1 sudo k3s kubectl apply -f dashboard-admin.yaml
```


4.3. Generación del token de acceso

Comando para generar el token del usuario admin:

```
1 sudo k3s kubectl -n kubernetes-dashboard create token admin-user
```

El token generado se utiliza para acceder al Dashboard vía web.

4.4. Acceso al Dashboard

Inicio del proxy local:

```
1 sudo k3s kubectl proxy
```

El Dashboard queda accesible en:

```
1 http://127.0.0.1:8001/api/v1/namespaces/kubernetes-dashboard/  
services/https:kubernetes-dashboard:/proxy/
```

Notas importantes:

- Es recomendable usar la versión 2.7.0 del Dashboard si la última versión falla con error 404.
- El token generado debe mantenerse seguro, ya que proporciona permisos de administrador.
- Asegúrate de que el nodo Master esté activo y listo antes de aplicar el manifiesto del Dashboard.

5. Conclusión

La creación del bridge permite que la VM tenga acceso directo a la red local y a Internet, facilitando la instalación de K3s y la integración con la Raspberry Pi como nodo slave. Esta configuración garantiza conectividad estable y control centralizado mediante la VM.

6. Bibliografía

Referencias

- [1] QEMU Documentation, “Networking with QEMU”, 2023. <https://www.qemu.org/docs/master/network/>
- [2] Rancher Labs, “K3s Lightweight Kubernetes”, 2023. <https://k3s.io/>
- [3] Systemd Documentation, “systemd-networkd”, 2023. <https://www.freedesktop.org/software/systemd/man/systemd-networkd.service.html>