



**Revista Electrónica de
Tecnología, Educación y Ciencia**
ISSN: 2953-5654
<http://retec.unsa.edu.ar>
Universidad Nacional de Salta

Containerlab en la virtualización de entornos de redes

Ernesto Sánchez, Daniel Arias Figueroa, Álvaro Gamarra, Sillvio Lucas Da Silva

Universidad Católica de Salta / Facultad de Ingeniería /
Universidad Nacional de Salta / C.I.D.I.A. (Centro de Investigación y Desarrollo en Informática
Aplicada) / Facultad de Ciencias Exactas /
Instituto Federal da Paraíba – Sede Joao Pessoa – Brasil.
Campus Castañares, 08105558227 / Av. Bolivia 5150, 3874255408 / Av. Primeiro de Maio, 720 -
Jaguaribe, João Pessoa, (83) 3612-1200
esanchez@ucasal.edu.ar, daaf@cidia.unsa.edu.ar, alvaroig@cidia.unsa.edu.ar, silvio.lucas@ifpb.edu.br

**Revista Electrónica de Tecnología, Educación y Ciencia,
Volumen 1, Número 2, pág. 65-71, jun, 2024. ISSN: 2953-5654**

Disponible en <http://retec.unsa.edu.ar/>

Containerlab en la virtualización de entornos de redes

Ernesto Sánchez, Daniel Arias Figueroa, Álvaro Gamarra, Silvio Lucas Da Silva

Universidad Católica de Salta / Facultad de Ingeniería /
Universidad Nacional de Salta / C.I.D.I.A. (Centro de Investigación y Desarrollo en Informática
Aplicada) / Facultad de Ciencias Exactas /
Instituto Federal da Paraíba – Sede Joao Pessoa – Brasil.
Campus Castañares, 08105558227 / Av. Bolivia 5150, 3874255408 / Av. Primeiro de Maio,
720-Jaguaribe, João Pessoa, (83) 3612-1200
esanchez@ucasal.edu.ar, daaf@cidia.unsa.edu.ar, alvaroiq@cidia.unsa.edu.ar, silvio.lucas@ifpb.edu.ar

Resumen. Desde hace varios años la utilización de herramientas de software para la simulación de entornos de redes de computadoras han tenido una adopción cada vez mayor por parte de entidades educativas públicas o privadas, así como responsables de IT, para la creación de escenarios de pruebas antes de desplegar configuraciones o cambios en la topología de red en producción.

Hoy la tendencia en el despliegue de infraestructura de servidores y las aplicaciones que éstos ejecutan se orienta al uso de contenedores los cuales se ejecutan como instancias virtualizadas independientes, lo que permite simplificar, agilizar, reutilizar y escalar los servicios que éstos prestan.

Como parte de los resultados finales del Proyecto de Investigación “Estudio de la implementación del protocolo de Internet versión 6 en el ámbito regional y su impacto en los usuarios finales”, el cual se desarrolló en el ámbito de la Universidad Católica de Salta, este trabajo expone un análisis de la herramienta “Containerlab”, la cual permite la virtualización de redes basado en contenedores.

Key words: Virtualización, Redes, Contenedores, Protocolo IPv6

1. Contexto

La línea de investigación se encuentra apoyada por el Consejo de Investigaciones de la Universidad Católica de Salta. Para llevar adelante las tareas de investigación se cuenta con la infraestructura provista por el C.I.D.I.A. (Centro de Investigación y Desarrollo en Informática Aplicada) que depende de la Facultad de Ciencias Exactas de la Universidad Nacional de Salta. Se cuenta con el financiamiento del Consejo de Investigación de la Universidad Católica de Salta, por lo tanto, se disponen de todos los recursos necesarios para el desarrollo del proyecto.

2. Introducción

La utilización de herramientas de software para la simulación y virtualización de redes han tenido un avance cada vez mayor. La empresa Cisco fue pionera con la herramienta de simulación Packet Tracer, la cual es utilizada actualmente por la empresa, para impartir certificaciones en administración de redes. Otra herramienta ampliamente utilizada es GNS3, la cual permite construir y configurar escenarios de redes mediante la emulación de sistemas operativos de red de diferentes fabricantes, a diferencia de la primera, GNS3 es de libre distribución.

La virtualización de redes se suma ahora a las ventajas en la utilización de contenedores. En el año 2021 Containerlab[1] surge como una iniciativa de ingenieros de Nokia, la cual se extiende

luego a pares de otras empresas. Esta herramienta permite el despliegue y administración de ambientes de laboratorio a través de una interfaz/línea de comando – CLI, la cual es capaz de inicializar los contenedores, construir el cableado virtual entre ellos según los requerimientos del usuario y gestionar el ciclo de vida del mismo [2].

Containerlab es de código abierto y se centra en los sistemas operativos de red basados en contenedores que normalmente se utilizan para probar características y diseños de red, como por ejemplo: Nokia SR Linux, CEO de Arista, Cisco XRd, Azure SONiC, EPRc de enebro, Cúmulo VX, FreeRtr, entre otros, un listado completo de los SO soportados se puede consultar en [3].

3. Análisis de Containerlab

Containerlab se distribuye como un paquete Linux, por lo que para su instalación se requiere de una plataforma Linux Server con servicio de Docker instalado, el proceso de instalación de resume a dos simples pasos; descarga y ejecución de un script de instalación.

Características principales:

- Enfoque IaaS: En base a los principios de Infraestructura como Código, (IaaS), permite la creación de topologías de redes mediante la definición de archivos YAML, ver Figura 1.
- Centrado en Sistema Operativo de Red, (NOS): Mediante la utilización de “Kinds”, permite simplificar la ejecución de diferentes contenedores de NOS. La utilización de estos “Kinds”, permite al usuario abstraerse de la necesidad de comprender ciertas particularidades de configuración de tales NOS.
- Integración con NOS basados en Máquinas Virtuales: Se integra mediante vrnlab con otros NOS basados en Máquinas Virtuales, tales como CISCO, DELL, Mikrotik, etc [4].
- Administración integrada de laboratorios: Permite iniciar los contenedores e interconectarlos, administrando el ciclo de vida de los contenedores, que implica la creación, destrucción, guardar cambios y gráfica de topologías.
- Telemetría: Permite su uso en aplicaciones de telemetría en redes.
- Captura de tráfico de red: Permite la integración con la herramienta Edgeshark para la captura de tráfico de red en las topologías creadas [5].

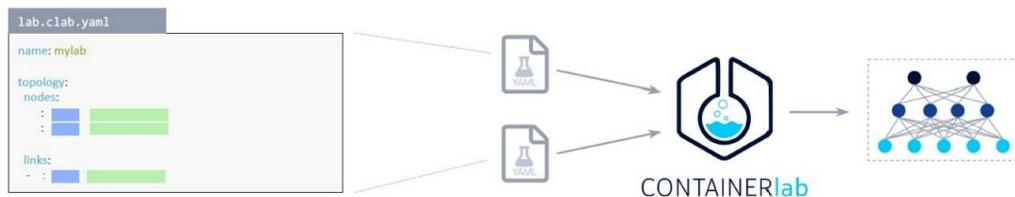


Figura 1: Definición de topología basada en YAML [6]

4. Ejemplo de creación de topología

Con la finalidad de poner en funcionamiento una topología creada en Containerlab, se propone el siguiente ejemplo el cual se basa en un router FRR Linux conectado a dos hosts Linux. El laboratorio propuesto pretende comprobar la asignación de direcciones IPv6 a dispositivos finales mediante la alternativa SLAAC [7].

La información para la construcción de una topología Containerlab se expresa como código defino en un archivo .yaml, para lo cual se utilizó la herramienta Visual Studio Code.

Los componentes principales en la definición de una topología son:

- **Name:** Permite identificar y distinguir entre todas las topologías creadas dentro de Containerlab.
- **Topology:** Este es el componente principal, donde se define el bloque de construcción que contiene los nodos, tipos y puntos de conexión.
- **Nodes:** Permite definir un contenedor el cual es de un tipo (Kind), y ejecuta una imagen de NOS.
- **Links:** Permite definir los puntos de conexión entre los nodos, indicando nombre de interfaz de red del nodo.

Dentro de la topología es posible también definir la ejecución de comandos de configuración específicos para cada nodo, así como el acceso a archivos de configuración.

Una vez creados el archivo de topología y archivos de configuración adicionales de los nodos, se deben publicar los mismos en un repositorio Github, para poder desplegar luego la topología en Containerlab. A modo de ejemplo se puede ver en Figura 2 el archivo de definición de topología propuesto:

```
name: ipv6slaac
mgmt:
  network: lan
  bridge: br-clab1
topology:
  nodes:
    router1:
      kind: linux
      image: quay.io/frrouting/frr:8.4.1
      binds:
        - router1/daemons:/etc/frr/daemons
        - router1/frr.conf:/etc/frr/frr.conf
      exec:
        - "sysctl -w net.ipv6.conf.all.forwarding=1"

    PC1:
      kind: linux
      image: docker.io/esanchezv/kaliipv6v2:latest
      exec:
        - ip -6 route del default dev eth0

    PC3:
      kind: linux
      image: praqma/network-multitool:extra
      exec:
        - ip -6 route del default dev eth0

  br-clab1:
    kind: bridge

  links:
    - endpoints: ["br-clab1:eth1", "router1:eth1"]
    - endpoints: ["br-clab1:eth2", "PC1:eth1"]
    - endpoints: ["br-clab1:eth3", "PC3:eth1"]
```

Figura 2: Archivo de topología ipv6slaac

5. Ejecución de topología

El proceso de ejecución de la topología creada se inicia con el clonado del repositorio Git que contiene los archivos de configuración. Desde el directorio de usuario del Linux Server con Containerlab instalado, ejecutar el comando:

```
"git clone https://github.com/ernestosv73/ipv6slaac.git".
```

Luego, dentro del directorio ipv6slaac, la creación de la topología se realiza mediante el comando:

```
"clab deploy -t ipv6slaac.yml --network=lan".
```

Para ver una descripción detallada de los comandos de acceso a los nodos puede consultar en [8]. En figura 3 se muestra en detalle el listado de los contenedores creados.

```

+-----+-----+-----+-----+-----+-----+
| # | Name | Container ID | Image | Kind | State |
| IPv4 Address | IPv6 Address |
+-----+-----+-----+-----+-----+-----+
| 1 | clab-ipv6slaac-PC1 | 265fb33e5f5d | docker.io/esanchezv/kaliipv6v2:latest | linux | runnin
g | 172.20.20.4/24 | 2001:172:20:20::4/64 |
| 2 | clab-ipv6slaac-PC3 | c4d38cdbff03 | praqma/network-multitool:extra | linux | runnin
g | 172.20.20.2/24 | 2001:172:20:20::2/64 |
| 3 | clab-ipv6slaac-router1 | 14b557b2f3eb | quay.io/frrouting/frr:8.4.1 | linux | runnin
g | 172.20.20.3/24 | 2001:172:20:20::3/64 |
+-----+-----+-----+-----+-----+-----+
    
```

Figura 3: Detalle topología ipv6slaac

6. Líneas de investigación y desarrollo

Los principales ejes temáticos que se están investigando son los siguientes:

- Protocolo de Internet IPv6.
- Virtualización de redes.
- Contenedores.
- Internet.

7. Resultados obtenidos

La topología descrita anteriormente corresponde a una de las etapas definidas en el proyecto de investigación, donde se propuso el diseño e implementación de escenarios de pruebas basados en herramientas de virtualización que permitan analizar el funcionamiento del protocolo IPv6 en el proceso de asignación de direcciones IP a dispositivos finales.

Con los resultados obtenidos, se proponen actividades de transferencia al medio, convenios con otras instituciones educativas a fin de conformar grupos de trabajo para el despliegue final del protocolo IPV6. En particular se destaca el convenio de colaboración entre la Universidad Católica de Salta y el instituto Federal da Paraiba Sede Joao Pessoa – Brasil, para el desarrollo del proyecto “IPv6: AMPLIANDO FRONTEIRAS DIGITAIS” [9].

8. Formación de recursos humanos

El grupo de investigación conformado se caracteriza por una constitución heterogénea de profesionales vinculados a la informática. El director es Magister en Redes de Datos por la UNLP y el investigador principal es Doctor en Ciencias Informáticas por la UNLP. Dentro de los investigadores se cuenta con un Magister en Informática, y un Ingeniero en Sistemas. También integran el grupo alumnos avanzados de la carrera Ingeniería en Telecomunicaciones de la Universidad Católica de Salta.

A futuro se pretende continuar con las líneas de investigación que surgen a partir de los resultados obtenidos y continuar con las vinculaciones con otras instituciones educativas.

9. Referencias

- [1] Containerlab Oficial Page. <https://containerlab.dev/>
- [2] "Laboratorios en ambientes virtualizados". LACNIC Blog. Alejandro Guevara. https://blog.lacnic.net/investigacion/laboratorios-en-ambientes-virtualizados?utm_source=web+webinar&utm_medium=web+webinar&utm_campaign=Webinar+container+lab&utm_id=Webinar+container+lab.
- [3] Kinds list supported. <https://containerlab.dev/manual/kinds/>
- [4] VM based routers integration. <https://containerlab.dev/manual/vrnetlab/>
- [5] Edgeshark Oficial Site. <https://edgeshark.siemens.io/#/>
- [6] "Running networking labs with Docker User Experience". Roman Dodin. FRNOG 2022. https://media.frnog.org/FRnOG_35/FRnOG_35-2.pdf.
- [7] RFC 4862 "IPv6 Stateless Address Autoconfiguration". S. Thomson, T. Narten, T. Jinmei. <https://datatracker.ietf.org/doc/html/rfc4862>.
- [8] Repositorio Github topologia ipv6slaac con Containerlab. <https://github.com/ernestosv73/ipv6slaac/tree/main>.
- [9] <https://www.ifpb.edu.br/joaopessoa/noticias/2023/10/instituto-federal-da-paraiba-firma-acordo-de-cooperacao-internacional-com-a-universidad-catolica-de-salta>