UNIVERSIDAD NACIONAL DE JUJUY
Facultad de Ingeniería
Analista Programador Universitario
Cátedra Redes II

Trabajo Práctico Nº 4

Redes virtuales con Docker

Docker como laboratorio para conceptos de redes

Datos de entrega	
Nombre del grupo	
Comisión	
Integrantes	
Año	2026

Introducción práctica

Unidad IV · Sistemas Operativos de Red

Docker se utilizará como herramienta de laboratorio: en una sola computadora se crearán contenedores que actuarán como nodos de red, permitiendo observar direccionamiento IP, conectividad, puertos, servicios y aislamiento entre redes lógicas.

En esta unidad se trabajará la relación entre los Sistemas Operativos de Red y los servicios que permiten la comunicación entre equipos.

Objetivo general

Utilizar Docker como laboratorio de redes para crear una red virtual, conectar nodos, publicar un servicio web y analizar conectividad, puertos, resolución de nombres y aislamiento entre redes lógicas.

Objetivos específicos

- Verificar el funcionamiento de Docker y diferenciar imagen y contenedor.
- Crear una red virtual bridge configurando explícitamente su subred y gateway.
- Conectar contenedores Ubuntu y un servidor Nginx a una misma red virtual.
- Identificar subred, gateway e IPs internas mediante docker network inspect.
- Comprobar comunicación entre contenedores mediante nombres e ICMP.
- Interpretar la publicación de puertos 8080:80 entre el host y el contenedor.
- Comprobar el aislamiento entre dos redes virtuales diferentes.
- Registrar evidencias, interpretar resultados y limpiar correctamente los recursos creados.

Modalidad de trabajo y entrega

- Trabajo grupal. Se realizará una única entrega por grupo.
- Todos los integrantes deben figurar en la portada y participar de la resolución.
- Las actividades deben resolverse en el orden indicado.
- Las evidencias deben corresponder a la ejecución real del grupo.
- Entrega: un único PDF con portada, respuestas, comandos, salidas y capturas solicitadas.
- Nombre del archivo: NombreDelGrupo-TP4.pdf. Ejemplo: Grupo1-TP4.pdf.

Situación problemática

Una pequeña empresa necesita probar una red interna antes de implementarla físicamente. Para ello se utilizará una sola computadora con Docker, simulando dos puestos de trabajo y un servidor web conectados a una red virtual.

```
RED EMPRESA: 192.168.100.0/24
Gateway:     192.168.100.1

PC1  ──────  PC2 ──────  SERVIDOR WEB
Ubuntu      Ubuntu        Nginx
```

Luego se incorporará una segunda red para comprobar que dos redes lógicas diferentes no se comunican directamente aunque todos los contenedores se ejecuten en la misma computadora física.

Antes de comenzar

- Docker debe estar instalado y en ejecución. En Windows y macOS, abrir Docker Desktop.
- En Windows, utilizar PowerShell. En macOS o Linux, utilizar Terminal. Los comandos largos se presentan en una sola línea para que puedan copiarse y ejecutarse sin depender de la sintaxis de continuación de cada sistema.
- Disponer de conexión a Internet para descargar imágenes y paquetes.

- Los nombres red-empresa, red-administracion, pc1, pc2, pc3 y servidor-web deben estar libres.

Actividades del trabajo práctico

Actividad 1 · Verificar Docker

Ejecuten:

```
docker version
docker ps
```

Registrar: sistema operativo utilizado, terminal utilizada y salida relevante. La sección Server confirma que el motor de Docker responde.

Actividad 2 · Ejecutar el primer contenedor

```
docker run --rm hello-world
```

Busquen el mensaje “Hello from Docker!”. La primera vez Docker puede descargar la imagen.

Registrar y responder: copiar el mensaje obtenido y explicar en 2 o 3 líneas la diferencia entre la imagen descargada y el contenedor ejecutado.

Actividad 3 · Crear la red de la empresa

Crear una red bridge con subred y gateway definidos:

```
docker network create --driver bridge --subnet 192.168.100.0/24 --gateway 192.168.100.1 red-empresa

docker network ls
```

Responder:

- ¿Qué red fue creada?
- ¿Cuál es la dirección de red?
- ¿Qué significa /24?
- ¿Cuál es la máscara de subred?
- ¿Cuál es el gateway configurado?

Actividad 4 · Crear dos puestos de trabajo

Crear dos contenedores Ubuntu conectados a red-empresa:

```
docker run -d --name pc1 --network red-empresa ubuntu:24.04 sleep infinity
docker run -d --name pc2 --network red-empresa ubuntu:24.04 sleep infinity
docker ps
```

Registrar una captura donde se observen pc1 y pc2 en ejecución.

Actividad 5 · Incorporar el servidor web a la misma red

Crear un servidor Nginx conectado a red-empresa y publicar el puerto HTTP:

```
docker run -d --name servidor-web --network red-empresa -p 127.0.0.1:8080:80 nginx:stable

docker ps
```

Abrir en el navegador del mismo equipo:

```
http://localhost:8080
```

Registrar: línea de servidor-web en docker ps, captura del navegador y respuesta sobre qué puerto usa el navegador y qué puerto usa Nginx dentro del contenedor.

Actividad 6 · Inspeccionar la red

```
docker network inspect red-empresa
```

En la salida, buscar IPAM.Config y Containers. Completar:

Dato	Valor obtenido por el grupo
Subred	
Gateway	
IP pc1	
IP pc2	
IP servidor-web	

Responder: ¿pc1, pc2 y servidor-web pertenecen a la misma subred? Justificar con las IP obtenidas.

Actividad 7 · Comprobar comunicación entre contenedores

Instalar ping en pc1 y comprobar comunicación con pc2 y con el servidor:

```
docker exec pc1 apt-get update
docker exec pc1 apt-get install -y iputils-ping

docker exec pc1 ping -c 4 pc2
docker exec pc1 ping -c 4 servidor-web
```

Registrar y responder:

- Copiar el resultado de los ping.
- Indicar a qué IP se resolvió pc2 y compararla con la IP registrada en la Actividad 6.
- Indicar a qué IP se resolvió servidor-web.
- Explicar qué protocolo utiliza ping.
- Explicar por qué ping no comprueba que funcione HTTP.

Actividad 8 · Detener y volver a iniciar el servidor web

Con la página web abierta, ejecutar:

```
docker logs servidor-web
docker stop servidor-web
docker ps
docker ps -a
```

Observar si aparecen solicitudes HTTP realizadas desde el navegador.

Recargar el navegador. La página no debería responder desde el servidor. Luego ejecutar:

```
docker start servidor-web
docker ps
```

Recargar nuevamente http://localhost:8080.

Registrar y responder: mostrar servidor-web detenido en docker ps -a y luego activo en docker ps; describir qué ocurrió en el navegador; explicar la diferencia entre docker ps y docker ps -a.

Actividad 9 · Desafío: aislamiento entre redes

Crear una segunda red y un tercer contenedor conectado solo a esa red:

```
docker network create --driver bridge --subnet 192.168.200.0/24 --gateway 192.168.200.1 red-administracion

docker run -d --name pc3 --network red-administracion ubuntu:24.04 sleep infinity

docker network inspect red-administracion
```

Identificar la IP de pc3. Luego, desde pc1, intentar comunicarse con pc3 usando la IP obtenida:

```
docker exec pc1 ping -c 4 <IP_DE_PC3>
```

Registrar y responder:

- ¿pc1 puede comunicarse directamente con pc3?
- ¿Por qué ocurre ese resultado?
- ¿Influye que todos los contenedores estén ejecutándose en la misma computadora física?
- ¿Qué concepto de redes se demuestra con este ejercicio?
- ¿Qué dispositivo o función de red sería necesario incorporar para permitir la comunicación entre ambas redes?
  Idea clave: misma computadora física no significa misma red lógica.

Actividad 10 · Limpiar los recursos del TP

Cuando hayan guardado todas las evidencias, ejecutar:

```
docker stop servidor-web pc1 pc2 pc3
docker rm servidor-web pc1 pc2 pc3
docker network rm red-empresa red-administracion

docker ps -a
docker network ls
```

Comprobar que ya no aparezcan los contenedores ni las redes creadas para este TP. Las imágenes descargadas se conservan.

Conclusión

Redactar una conclusión breve de 5 a 8 líneas explicando qué pudieron comprobar sobre redes virtuales, direccionamiento IP, publicación de servicios, puertos, comunicación entre contenedores y aislamiento entre redes lógicas.

Ayuda para problemas frecuentes

Problema	Qué revisar
Docker no responde	Abrir Docker Desktop y esperar a que el motor esté listo. Repetir docker version.
El nombre del contenedor ya está en uso	Revisar docker ps -a. Si pertenece a este TP, limpiar antes de repetir docker run.
El puerto 8080 está ocupado	Usar 8081 en el mapeo y en el navegador: 127.0.0.1:8081:80. Registrar el cambio.
Falla apt-get o descarga de imagen	Comprobar conexión a Internet y registrar el error.
ping no responde	Verificar que los contenedores estén activos y conectados a la red correcta con docker network inspect.