
Actividad 4

Fundamentos de docker

Mikroways



Redes

1. Cree dos contenedores basados en alpine de forma interactiva llamados **uno-default** y **dos-default** respectivamente. Estos contenedores deben utilizar la red por defecto llamada **bridge** (para lo cual no debe especificar ninguna red usando `--network`). Luego verifique conectividad entre ellos usando el comando **ping**.

Utilice con el comando **docker run** usado para lanzar los contenedores **uno-default** y **dos-default** la opción `-e VAR=uno` y `-e VAR=dos` respectivamente.

- Verifique la ip de cada contenedor usando `ip add ls` y ejecute el comando **ping** y la IP del contenedor remoto.
 - Repita el procedimiento anterior utilizando el nombre del contenedor remoto. ¿Funciona?
2. La opción **docker run --link** no es aconsejada. Sin embargo, es la forma en que un contenedor puede conectar con otro por nombre cuando se utiliza la red por defecto. Repita el ejercicio 1, esta vez utilizando la opción `--link` para lanzar el contenedor **dos-default**.

Utilice con el comando **docker run** usado para lanzar los contenedores **uno-default** y **dos-default** la opción `-e VAR=uno` y `-e VAR=dos` respectivamente.

- Verifique el comando ping contra el contenedor remoto utilizando ahora el nombre del contenedor. ¿Funciona en todos los contenedores o sólo en el que incluye `--link`?
 - Verifique además las variables de ambiente en **uno-default** y **dos-default** usando el comando **env**. ¿Qué observa?
3. Cree una red de tipo bridge definida por el usuario y corra dos contenedores basados en alpine de forma interactiva llamados **uno-user** y **dos-user**.

Utilice con el comando **docker run** usado para lanzar los contenedores **uno-user** y **dos-user** la opción `-e VAR=uno` y `-e VAR=dos` respectivamente.

- Verifique el comando ping contra el contenedor remoto utilizando ahora el nombre del contenedor. ¿Funciona en todos los contenedores?
- Verifique además las variables de ambiente en **uno-user** y **dos-user** usando el comando **env**. ¿Qué observa?
- Desconecte uno de los contenedores mientras desde el contenedor remoto ejecuta el comando ping. Luego vuelva a conectarlo



Para desconectar y conectar contenedores investigue el uso de los comandos **docker network disconnect** y **docker network connect**.

- Indique el comando necesario para correr un contenedor que no tenga conexión de red.
- Cree una red definida por el usuario con una subred que defina usted. Luego repita el ejercicio 3 verificando que cada contenedor toma una IP del segmento definido por la red creada.

Límite de recursos

RECUERDE QUE PARA PODER LIMITAR LA MEMORIA Y FUNCIONE DEBE DESACTIVAR LA SWAP DE SU MÁQUINA: `sudo swapoff -a`.

- Lance el contenedor a partir de la imagen **progrium/stress**:

```
1 docker run --rm progrium/stress -c 1 --vm 1 --vm-bytes 200m -t 60
```

Mientras corre el comando durante 1 minuto, verifique las estadísticas de consumo:
docker stats \$(docker ps -ql)

Deshabilite la swap en su PC para la siguientes pruebas: `sudo swapoff -a`

- Corra el mismo comando del ejercicio 1, pero limitando la cantidad de memoria usando la opción `-m 128M`.

Quite la opción `--rm` para poder inspeccionar el contenedor finalizado.

- ¿Cuál es el exit code?
- Inspeccionando el contenedor, ¿podemos inferir que fue parado por OOMKiller?

Entregables

Los ejercicios a entregar de la sección redes son el 1, 2, 3, 4 y 5. Para la sección de límites de recursos son el 1 y 2. Las entregas deben responder a las preguntas de cada ejercicio o los comandos usados para completarlos.

