

Trabajo Práctico N°1

Nahuel Defossé

2018

Taller de Nuevas Tecnologías

Notas

Los trabajos pueden realizarse en grupos de hasta *dos personas*. La fecha de entrega será pautada, en dos semanas a partir de la fecha de entrega. El formato de entrega puede ser PDF o HTML. Se deberá incluir evidencia de el trabajo realizado. Los trabajos no aprobados tendrán un período de recuperación de una semana posterior a la fecha de recepción de las devoluciones.

Construcción y gestión de contenedores de Software

1. Descargue la imagen de Apache (`httpd`) desde Docker Hub y muestre la salida del comando `docker images`.
2. Ejecute la imagen con política de reinicio `always`, exponiendo el puerto 80 al host anfitrión. Evidencie la ejecución mediante el comando `uptime` seguido de `docker ps`, reinicie el servidor o computadora dónde ejecuta el servicio docker y vuelva a ejecutar dichos comandos. ¿Qué puede inferir a partir de las columnas `CREATED` y `STATUS`?
3. Inspeccione la configuración mediante `docker exec <contenedor> sh` y ubique la configuración de la raíz de documentos. Suplente este contenido mediante un volumen. Evidencie los cambios. ¿Qué diferencia existe si un volumen está creado con una ruta absoluta vs. cuando se crea con una cadena? ¿Para qué existe el comando `docker volumes`? Relaciónelo con la pregunta anterior. ¿Qué ciclo de vida tiene cada tipo de volumen?
4. Cree un contenedor a partir de una imagen (puede reutilizar el de los puntos previos) y ejecute `ps` dentro de este, registrando `PIDs` y nombres de procesos. Utilice `docker exec` para lanzar un nuevo shell. ¿Qué PID toma? ¿Qué sucede si desde este shell envía una señal 9 (`SIGKILL`) al PID 1?

5. Genere un **Dockerfile** basado en la imagen descargada en el punto 1 que contenga un archivo y constrúyala mediante **docker build -t <nombre> .**
6. Ejecute **docker history <nombre>** sobre la imagen generada en el punto anterior. ¿Qué puede inferir a partir de la salida?
7. Empuje la imagen del punto anterior a <https://hub.docker.com/> mediante **docker push etiqueta**.
8. Aprovechone una máquina virtual o real (con acceso **ssh** configurado con clave pública) mediante el comando:

```
docker-machine create \
--driver generic \
--generic-ip-address=<ip o nombre> \
--generic-ssh-key ~/.ssh/id_rsa \
<nombre>
```

 - a. Ejecute **docker-machine ls**
 - b. Evalúe en el *shell* el ambiente para dicha máquina y ejecute **docker ps**.
9. Cree un proyecto **nodejs** y cree su correspondiente **Dockerfile**. ¿Cómo organiza los comandos dentro del **Dockerfile**? ¿Qué interacción hay entre COPY y RUN en cuanto a *cache*? ¿Por qué cree que es común utilizar COPY <fuente> <destino> en conjunto con un volumen en <destino>?
10. Agregue una imagen de **mongodb** mediante la creación de un archivo **docker-compose.yml**