

**GUIA 9 Docker**

**Actividad I**

- **Describa el funcionamiento de Docker.**

Es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones, permite separar sus aplicaciones de su infraestructura para que pueda entregar el software rápidamente

- **Analice y explique la diferencia entre los contenedores y las máquinas virtuales**

las maquinas virtuales hacen uso de la virtualizacion completa del sistema operativo a usar requiriendo los recursos necesarios para ello haciendo uso de todos ficheros que requiere el sistema a levantar, mientras que los contenedores solo necesitan un conjunto de ficheros para su uso sobretodo para ejecutar aplicaciones y sus servicios de forma aislada e independiente buscando que se comporte de igual forma en todos los entornos

- **¿Cuál es la diferencia entre los contenedores de Linux y Docker?**

Contenedores de linux es la herramienta elegida por muchos que desean ejecutar un sistema operativo completo en un contenedor, es más parecido a las máquinas virtuales verdaderas y debe manejarse de la misma manera

Docker, por otro lado, en lugar de ejecutar un contenedor similar a una máquina virtual está diseñado para ejecutar una aplicación, con docker se crea un entorno para ejecutar una aplicación con las dependencias necesarias para su uso

- **¿Para utilizar Docker es necesario darle permisos de sudo?**

no

- **Explique es es “Docker daemon”**

Es el entorno de tiempo de ejecución persistente que administra los contenedores de aplicaciones, y es independiente del sistema operativo subyacente, por tanto, cualquier contenedor de Docker puede ejecutarse en cualquier servidor que tenga habilitado este servicio. Es el encargado de convertir un dockerfile en un contenedor

- **Defina que son los “Dockerfiles”**

Es un documento de texto que contiene toda la información de configuración y los comandos necesarios para ensamblar una imagen de contenedor.

- **¿Cuál es la diferencia entre Docker y Docker Compose?**

Docker se usa al administrar contenedores individuales en un motor acoplable.

Docker Compose se puede usar para administrar una aplicación de contenedores múltiples

- **Describe la relación entre las imágenes de Docker y las layers**

Las imágenes docker son generadas a partir de una serie de layers. Cada layer representa una instrucción en el archivo Dockerfile.

- **¿Qué son las variables de entorno?**

Es un valor con nombre dinámico que puede afectar la forma en que los procesos en ejecución se comportarán en una computadora. Son parte del entorno en el que se ejecuta un proceso

- **¿Qué son los volúmenes en Docker?**

Es la manera sencilla y predefinida para almacenar todos los ficheros (salvo unas pocas excepciones) de un contenedor, usará el espacio de nuestro equipo real y en “/var/lib/docker/volumes” creará una carpeta para cada contenedor.

- **Enumere las ventajas de utilizar volúmenes en Docker**

1. Montar el código fuente de una aplicación web, dentro de un volumen, accesible desde el contenedor web y así ver en tiempo real los cambios durante el desarrollo.
2. Consultar todos los logs cómodamente desde un contenedor dedicado.
3. Hacer backups de un contenedor desde otro dedicado, o recuperar esos mismo backups hacia nuestro host.
4. Compartir la misma información entre varios contenedores sin duplicarla
5. Los volúmenes de datos están diseñados para conservar los datos, independientemente del ciclo de vida del contenedor.
6. Docker, por lo tanto, nunca elimina automáticamente los volúmenes cuando se elimina un contenedor, ni tampoco “recoge basura”: volúmenes huérfanos a los que ya no hace referencia un contenedor.

7. Los volúmenes son específicos de cada contenedor
  8. Los volúmenes también pueden usarse para compartir información entre contenedores
- **Describe el funcionamiento de los siguientes comandos**
    - docker build* : Crea una imagen desde un Dockerfile
    - docker ps-a* : muestra que contenedores están funcionando
    - docker builder* : administrador de compilaciones
    - docker inspect <nombre\_imagen>* : proporciona información detallada acerca del contenedor
    - docker exec* : ejecuta un nuevo comando en un contenedor en ejecución
    - docker volume* : administrador de volúmenes
    - docker kill* : Mata uno o más contenedores en funcionamiento
    - docker volume ls* : lista los volúmenes
  - Cuando se está usando volúmenes en Docker, ¿cuál es la diferencia entre prune y rm?
    - rm elimina uno o mas volúmenes pero no puede eliminar hasta borre el contenedor que este usando el volumen
    - prune borra todo los volúmenes que no están siendo usados por un contenedor

## Actividad II

Se le solicita levantar dos instancias de Jenkins utilizando los volúmenes de Docker; para ello necesita crear un volumen en el cual se alojarán los datos del home de jenkins, una vez levantada la primera instancia se le solicita crear la segunda usando el mismo volumen. Cuando el proceso finalice interprete y explique cuál ha sido el comportamiento de los contenedores y sus datos.

```
docker run -u 0 -d -p 8080:8080 -p 50000:50000 -v /data/jenkins:/var/jenkins_home  
jenkins
```

Explicando este comando es importante ingresar con el usuario root ya que es o, se crea un volumen de docker en el home del usuario local

```
docker run -u o -d -p 8080:8181 -p 50000:50001 -v /data/jenkins:/var/jenkins_home  
jenkins/jenkins:its
```

Se crea otro contenedor apuntando al mismo volumen local

NOTA: No se solicita el uso de Jenkins solamente se requiere levantar la instancia del mismo