

## RESUMEN DE DOCKER

Nombre: Gabriel Grimaldo

Docker es un proyecto de código abierto que sirve para automatizar la implementación de aplicaciones, Docker empaqueta software en las unidades llamadas contenedores, estos contenedores incluyen todo lo necesario para que el software se ejecute, en estos se incluyen bibliotecas, herramientas de sistema y código. Los contenedores nos ayudan a implementar las aplicaciones de una forma más rápida en cualquier entorno y además nos dan la certeza de que estas aplicaciones se ejecutaran. El uso de estos contenedores se conoce como contenedorización. El uso de los contenedores con el paso del tiempo se vuelve mas popular debido a las características que ofrecen, estos contenedores son: Flexibles, ligeros, portables, escalables, poseen un acoplamiento flexible y son seguros.

Un contenedor no es mas que un proceso en ejecución al cual se le características de encapsulación para que se mantenga aislado del host y del resto de contenedores, estos contenedores al estar aislados interactúan con su propio sistema de archivos privados; este sistema de archivos es proporcionado por una imagen de Docker. Las imágenes incluyen todo lo necesario para ejecutar una aplicación, es decir que estas son las que incluyen todas las dependencias y cualquier otro objeto requerido por el sistema. A partir de estas imágenes podemos ejecutar uno o varios contenedores, estos contenedores se pueden agregar y quitar de forma dinámica. Una de las mayores características de las imágenes es que manejan herencia, esto quiere decir que a partir de una imagen padre podemos heredarla para así obtener sus componentes.

Una de las características principales que diferencia a Docker de una máquina virtual (VM) es que Docker no instala el sistema operativo completo, sino que comparte el núcleo y algunas bibliotecas con el sistema anfitrión. Docker representa los principios y comandos básicos de Linux, pero puede ser utilizado en varios sistemas operativos.

Docker suele ser utilizado para levantar maquinas aisladas con sistemas operativos ligeros y en muy poco tiempo, también es utilizado para ahorrar costes en servidores debido a que Docker es un sistema aislado del sistema anfitrión, por lo que la ejecución de una aplicación en esa maquina no afectara al puesto en el que estamos trabajando, otro uso que se le da es el control de versiones y depuración debido a que nos ofrece un sistema de control de versiones, este sistema nos permite regresar a la versión que queramos en caso de ser necesario, esto es un aspecto muy útil al momento de la implementación de mejoras en las aplicaciones.

¿Cómo justificaría la implementación de Docker para una empresa?

Docker además de ser un proyecto de código abierto brinda muchos beneficios a las empresas que adquieren este servicio, ya que al momento de querer desarrollar una aplicación solo tendrían que programar esta aplicación una sola vez, dado que se ejecuta dentro de un contenedor y este contenedor se ejecuta en cualquier sistema operativo que tenga Docker instalado, esto es uno de sus más grandes beneficios que justificaría la implementación debido a que se reducirían los gastos a en el momento de crear nuevas aplicaciones, otro beneficio que justifica la implementación es que se obtiene un mayor consistencia entre los entornos de prueba y los entornos de producción, esto se debe a que cuando se desarrolla en Docker se hacen pruebas de la aplicación dentro del contenedor, esto significa que el entorno de prueba

es idéntico al entorno en el que se va a ejecutar el software, esto le garantiza a la empresa que adquiera Docker la eficiencia de sus aplicación.

## Prueba en Docker

Las siguientes capturas muestran la prueba de comandos básicos que se incluían en el laboratorio de Docker para principiantes en Linux

- Ejecute una sola tarea en un contenedor de Alpine Linux

```
[node1] (local) root@192.168.0.18 ~
$ docker container run alpine hostname
ef358b2c8102
[node1] (local) root@192.168.0.18 ~
$
```

- Listar todos los contenedores

```
8a8ec78ac87c
[node1] (local) root@192.168.0.18 ~
$ docker container ls --all
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS              NAMES
8a8ec78ac87c        alpine             "hostname"          3 seconds ago       Exited (0) 2 seconds a
go
eloquent_galois
b98651b5fc5b        alpine             "hostname"          20 seconds ago       Exited (0) 19 seconds
ago
youthful_ptolemy
```

- Ejecute un contenedor Docker

```
[node1] (local) root@192.168.0.18 ~
$ docker container run --interactive --tty --rm ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
6a5697faee43: Pull complete
ba13d3bc422b: Pull complete
a254829d9e55: Pull complete
Digest: sha256:fff16eea1a8ae92867721d90c59a75652ea66d29c05294e6e2f898704bdb8cf1
Status: Downloaded newer image for ubuntu:latest
```

- Comandos de Linux dentro del contenedor

```
root@7f688bda0771:/# ls /
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@7f688bda0771:/# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.6  0.0  4108  3576 pts/0    Ss   20:52   0:00 bash
root        10  0.0  0.0   5888  2868 pts/0    R+   20:52   0:00 ps aux
root@7f688bda0771:/# cat /etc/issue
Ubuntu 20.04.1 LTS \n \l
```

- Verificación de la versión de la máquina virtual

```
[nodo1] (local) root@192.168.0.18 ~  
$ cat / etc / problema  
Bienvenido a Alpine Linux 3.12  
Kernel \ r en una \ m (\ l)
```

- En esta prueba que ofrece Docker podemos observar cómo se pueden utilizar los comandos de Linux para acceder a los contenedores que se crean en Docker.