DOCKER

Docker es un contenedor donde puedo correr mis aplicaciones en cualquier lugar

-Window

-linux

-dataCenter

-en la nube

-en maquinas virtuales

las alternativas de usar docker son:

-Docker Desktop

-Docker Engine

-Play whit Docker

Docker engie esta dividido en:

-docker deamon

-rest API

-docker cli

Contenedor: un contenedor es una unidad estandar de software que empaqueta el codigo y todas sus dependencias para que la aplicación se ejecute de forma rapida y confiable de un entorno informatico a otro.

Imagenes: son una colección ordenada de cambios del sistema de archivo raiz y los parametros de ejecucion correspondientes para uso dentro de un tiempo de ejecucion del contendor, una imagen no tiene estado y nunca cambia

los logs: son las salidas

Volumenes: son el mecanismo que se utiliza para conservar los datos generados y utilzados por los contenedores de docker

Redes: el sistema de red de docker se puede conectar mediante controladores y son:

-bridge

-host

-overlay

-macvlan

-none

DockerHub: proporciona:

-alojamiento de imagenes

-autenticacion del usuario

-construcciones de imagenes automatizadas y herramientas de flujo de trabajo como activadores de compilacion y enlaces wed

-integracion con github y bitbucket

DockerFile: es un documento de texto que contiene todos los comandos que normalmente ejecutaria normalmente para crear una imagen de docker

Docker-Compose: es una aplicación para definir y ejecutar aplicaciones Docker de varios contenedores. Se compone de 3 pasos:

-Definir el entorno de su aplicación con DockerFile

-definir los servivios de la aplicación en docker -compose.yml

-Ejecutar docker -compose up y compose inicia y ejecuta toda la aplicación

Comandos:

• docker create: crea un contenedor pero no lo comienza.

• docker rename: permite renombrar al contenedor.

• docker run: crea y comienza un contenedor en una operación.

• docker rm: borra un contenedor.

• docker update: actualiza los recursos limitados de un contenedor.

• docker start: comienza un contenedor si se cayó o salió.

• docker stop: detiene un contenedor.

• docker restart: detiene y comienza un contenedor.

• docker pause: pausa un contenedor corriendo, "lo congela".

• docker unpause: quita la pausa de un contenedor corriendo.

• docker wait: bloquea hasta que un contenedor corriendo se detiene.

• docker kill: envía una SIGKILL a un contenedor corriendo.

• docker attach:se conecta a un contenedor corriendo.

• docker ps: muestra los contenedores corriendo.

• docker logs: obtiene logs de un container.

• docker inspect: observa toda la info en un contenedor.

• docker events: obtiene eventos de un contenedor.

• docker port: muestra el puerto publico de un contenedor.

• docker top: muestra los procesos corriendo en un contenedor.

• docker stats: muestra las estadisticas de recursos usados por contenedor.

• docker diff: muestra los archivos cambiados en el FS del contenedor.

• docker ps -a: muestra todos los contenedores corriendo y detenidos.

• docker stats –all: muestra una lista de los contenedores corriendo.

• docker images: muestra todas las imagenes

• docker import: crea una imagen de un tarball.

• docker build: crea imagen de un  Dockerfile.

• docker commit: crea imagen de un contenedor, pausandolo temporalmente si esta corriendo.

• docker rmi: remueve una imagen.

• docker load:carga una imagen de un archivo tar como STDIN, incluyendo imagenes y tags.

• docker save:salva una imagen a un archivo tar a STDOUT con todas las capas padre, tags y versiones.

• docker history: muestra el historial de una imagen.

• docker tag: taggea una imagen a un nombre asignado.

• docker volume create <volume> : crea un volumen

• docker volume ls : lista todos los volúmenes

• docker volume prune : elimina todos los volúmenes que no están siendo utilizados

• docker pull <image>: Descargar imágenes

• docker run <image> [<command>] : crea un contenedor con un comando root

--name <container> : con un nombre específico

--rm : lo elimina cuando termina de utilizarse

-it : en modo interactivo

-d : en modo detach (sin mostrar output)

• docker exec -it <container> <command> :ejecuta un comando en un contenedor ya existente, en modo interactivo

• docker kill <container> : termina cualquier proceso que se esté ejecutando en el contenedor

• docker ps -a : lista todos los contenedores

• docker ps -aq :lista el nombre de todos los contenedores

• docker inspect <container> : inspecciona un contenedor

• docker logs <container> : muestra el output de un contenedor

• docker rename <old-container> <new-container-name> : renombra un contenedor

• docker rm <container> : elimina un contenedor

• docker rm -f <container> : elimina un contenedor aunque esté siendo ejecutado

• docker rm $(docker ps -aq) : elimina todos los contenedores

• docker-compose up: da instrucciones a Docker para crear el contendor y ejecutarlo.

• docker-compose down: apaga todo los servicios que levantó con docker-compose up.

• docker-compose ps: permite ver los contenedores funcionando.

• docker-compose exec: permite ejecutar un comando a uno de los servicios levantados de Docker-compose.