Docker

Què és?

Docker és un software Open Source que utilitza els contenidors de Linux per construir, transportar,compartir i executar procesos de forma aillada.

Docker utilitza recursos del sistema totalment aïllats, és a dir, permet asignar recursos (CPU, memoria del sistema, etc.) o combinacions d'aquests recursos entre els procesos que executa el sistema.

Per fer-ho Docker va desenvolupar la llibrería «libcontainer», que és una implementació que aprofita les característiques dels «Contenidors Linux» (Linux Containers (LXC)) i la llibreria libvirt. Així, podem tenir una distribució linux dins d'una altra distribució linux que uilitzarà el kernel de l'anfitrió i la seva targeta de xarxa pero que treballarà de manera encapsulada amb la seva propia IP de xarxa, la seva propia interface de xarxa i els seus propis procesos.

Contenedores Docker vs. Màquines virtuals.

Les màquines virtuals (sobre VirtualBox, VMWAre, etc) necesiten un sistema operatiu virtualitzat. Això fa que es consumeixin més recursos de hardware i per tant, es puguin utilitzar menys d'aquests recursos (ram, procesador, etc.) en la utilització de les aplicacions.

D'altra banda, Docker no emula una màquina amb un sistema operatiu propi, sinó que utilitza els recursos del sistema amfitrió (el nucli del sistema operatiu, algunes llibreries...) i per tant, els processos són molt més lleugers i s'aprofita millor el hardware.

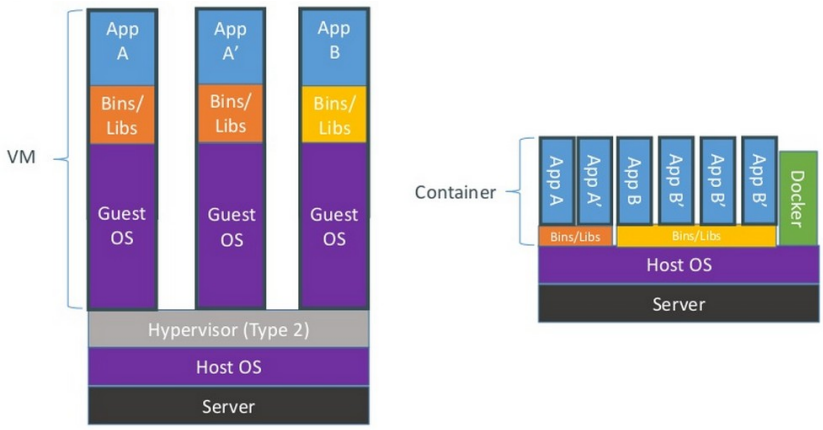

Diagram source: Docker Inc.

Pel que fa a l'emmagatzemament al disc, una màquina virtual pot ocupar molts giges ja que ha de contenir un sistema operatiu complet. En canvi, un contenidor Docker només ha d'emagatzemar tot allò que el diferencia del sistema operatiu on s'executa.

Per a què serveix?

Docker simplifica l'estructura de les aplicacions per permetre la seva instalació, execució i distribució d'una manera més fàcil i ràpida. Les aplicacions «dockeritzades» són completament portables i poden ser executades en qualsevol entron.

En crear un contenidor Docker d'una aplicació, podem reduïr el temps de posada en marxa del servei d'una forma automatizada independentment del sistema operatiu utilitzat per l'equip amfitrió.

Avantatges

- Consumeixen menys recursos de hardware que les màquines virtuals, en favor de les aplicacions que s'executen amb major fluidesa.
- Les instancies arranquen en poc temps.
- Existeixen moltes imatges que es poden descarregar i modificar lliurement.

Inconvenients

- Només es pot utiltzar de forma nativa en entorns Unix, encara que es pot virtualitzar gràcies a Docker Toolbox en sistemes OSX i Windows.
- Les imatges han d'estar basades en versions de Linux modernes (kernel 3.8 o superior).

Repositori Docker Hub

El servei Registry Docker Hub permet crear, compartir i utilitzar imatges creades per nosaltres mateixos o per altres persones.

És un servei gratuit que permet tenir un repositori privat i altres públics.

Imatges i contenidors

Les imatges en Docker es podrien entendre com a un component estàtic, són únicament un sistema operatiu base, un conjunt d'aplicacions empaquetades, mentre que un contenidor és l'execució d'una imatge, capaç d'executar diferents contenidors a partir d'una mateixa imatge.

Instalació

Nota: El motor de Docker (Docker Engine) utilitza el kernel de Linux, per tant per executarlo dins d'una distribució diferent haurem d'utilitzar Docker Tools.

Instalació del paquet docker

```
# dnf -y install docker-io
```

Arrancar el servei.

```
# systemctl start docker
```

Per arrancar el servei automáticament en encedre el sistema.

```
# systemctl enable docker
```

Atenció. Cal tenir els permisos adequats per fer servir dockers.

```
# groupadd docker

# usermod -a -G docker laura

# chown laura:docker /var/run/docker.sock
```

Administració básica dels contenidors docker

La gestió dels dockers es realitza a través de la comanda «docker» seguida dels diferents paràmetres:

docker info: mostra la informació relacionada

docker images: per veure les imatges disponibles al nostre sistema

docker ps: per veure els contenidors iniciats

dockers ps -a: per veure tots els contenidors inclosos els no iniciats

docker login: per loguejar-nos a un repositori d'imatges

docker build: per crear una imatge mitjançant un Dockerfile

docker attach: per entrar en mode interactiu en un Docker en marxa

docker create: per crear un container

docker start: per engegar un container que esta aturat

docker attach: connecta a la sessió del terminal d'un container

docker start -a: per engegar i connectar a un container aturat

docker run: fa la funció de docker create + docker start

docker rm/rmi: esborra un container o una imatge

docker logs: mostra els logs en cas de fallada d'un Docker

docker commit: per construir una nova imatge a partir d'un container

docker tag: per modificar el nom d'una imatge

docker pull/push: per pujar/descarregar una imatge d'un repositori

Dockerfiles

Què és un Docker file?

Un Dockerfile és un arxiu de text que permet definir les intruccions a seguir per Docker per construir una imatge personalitzada.

S'utilitza la instrucció «docker build» per construir una imatge Docker executant les instruccions indicades al Dockerfile de forma sequencial.

Instruccions Dockerfile

Un arxiu Dockerfile està format per una sèrie d'instruccions (en majúscules) que indiquen la tasca que ha d'executar la comanda constructora de la imatge.

A tenir en compte: Docker tracta les linies que comencen en # com a comentaris. Si el caracter # apareix en qualsevol altre part de la linia (instrucció), Docker el tractarà com a un argument.

- FROM: indica la imatge base a partir de la qual crear la imatge que construirà el Dockerfile.

- MAINTAINER: documenta el creador de la imatge.

- ENV: permet crear variables d'entorn.

- ENV HOME: estableix el directori HOME que utilitzaràn les comandes (RUN).

- ADD <origen>... <destí>: permet afegir un arxiu, directori o arxiu remot al sistema de fitxers del contenidor, en ocasions s'utilitza para proporcionar la configuració dels serveis.

  En cas que l'origen sigui un fitxer comprimit, el descomprimeix al destí.

  <origen> ha d'estar dins del context de construcció de la imatge, és a dir, on està el Dockerfile.

- COPY: Semblant a ADD, una versió més bàsica.

- WORKDIR: permet especificar a Docker en quin directori s'executaràn les instruccions CMD, RUN o ENTRYPOINT.

  Només pot existir una única instrucció CMD per a cada Dockerfile.

- VOLUME: estableix punts de muntatge, són la forma de externalitzar un determinat directori i proporcionar persistencia (les imatges de docker són només de lectura i no emmagatzemen dades entre diferents execucions). Així, es possible accedir desde el nostre host a un determinat directori del contenidor.

- EXPOSE: indica els ports TCP/IP pels que es pot accedir als serveis del contenidor.

- RUN: permet executar una instrucció al contenidor, por exemple, para instalar paquets (apt-get, yum, rpm, …).

- CMD: es similar a RUN però en comptes d'executar-se quan s'executa la comanda «docker build» ho fa quan arranquem el contenidor.

- ENTRYPOINT: Similar a CMD, la imatge executarà la comanda que li diguem tan aviat com es crei el contenidor. En canvi, a diferència de CMD aquesta instrucció no deixarà executar res més. És a dir, encara que a l'arrancar el contenidor li pasem una altre ordre, la ignorarà i només executarà la indicada al entrypoint.

Volums

Els volums permeten a Docker fer que les dades persisteixin a l'aplicació. Allotja les dades fora del propi contenidor, dins del sistema d'arxius del host on està funcionant Docker. Així, es pot canviar, apagar o esborrar el contenidor sense que afecti a les dades. Per fer-ho, cal definir quina part del contenidor es dedica a la apliacació i quina part a les dades.

Els colms son específics de cada contenidor, es a dir, es poden crear n contenidors a partir d'una mateixa imatge i definir volums diferents per a cada un d'ells.

Es important saber que els volums poden usar-se per compartir informació entre contenidors.

Construcció d'una imatge Docker

$ docker build -t image_name dockerfile_directory

La ordre docker build, constreix una imatge a partir de Dockerfile i un determinat contexte. Aquest contexte són els arxius localitzats a un determinat PATH (local) o URL (Git repository).

-f, --file=PATH/Dockerfile: indica on es troba el Dockerfile que s'utilitzarà en la construcció de la imatge.

--rm: esborra els containers intermedis creats durant la execució.

-t, --tag=' ' : indica el nom que se li donarà a la imatge resultant (en cas de ser exitosa).

Una forma correcta de nomenar les imatges és la següent: docker build -t user/image:version

Creació del contenidor Docker

$ docker create –name container_name -i -t image_name

La ordre docker create crea un container sobre la imatge especificada i la prepara per a la seva execució.

Nota: és similar a la ordre docker run, però en aquest cas no arranca el container.

-i, --interactive=true|false : conserva el STDIN obert.

-p, --publish : publica els ports del container (o rang de ports) al host.

Format: ip:host_port:container_port/protocol

-t, --tty: assigna una pseudo TTY.