Esercitazione 1

Esercizio 1

Esegui un container docker in modalità testuale interattiva con l'ultima versione del sistema operativo ubuntu partendo da una working directory /etc. L'hostname dovrà essere "amministrazione" e il nome del container "A3453". Il container dovrà avere una variabile d'ambiente chiamata "VARIABILE_BASE" con valore 0123. Creare un file chiamato "dati_nuovi" e scrivere del testo all'interno

Mostrare:

- **L'hostname del container**
- **La variabile creata**
- **Log del container**
- **Processi attualmente attivi**
- **Statistiche d'utilizzo del container**
- **Le differenze con l'immagine base**
- **Le differenze con l'immagine base**
- **L'ip address del container**
- **Avviare l'aggiornamento dei pacchetti del suddetto container in modalità detached.**
- **Proseguire con il commit del container.**

Verificare che l’immagine sia stata creata ed una volta verificato ciò rimuovere SOLTANTO quella appena creata

Per alcuni comandi potrebbe essere necessario aggiornare i pachetti ed installare tools, ad esempio per Ubuntu

```
apt-get update -y & apt-get install net-tools -y & apt-get install iputils-ping -y
```

Esercizio 2

Esegui un container con sistema operativo “Alpine” che una volta stampato l’hostname provveda ad autodistruggersi. Verificare se effettivamente il container è stato eliminato assicurandosi che non sia semplicemente in stato stopped e controllando la presenza del container creato nell'esercizio 1.

Esercizio 3

Creare un container in modalità interattiva. Eseguire la shell “bash” per poter interagire testualmente con il container creato e mai, fin ora eseguito, ed infine rimuovere il container

Esercizio 4

Eseguire un container in modalità interattiva e creare un file testuale. Una volta fatto ciò copiare il seguente file dal container alla macchina host. Fare viceversa. Esportare il container sotto forma di tar. Creare un utente chiamato “secondario” ed uscire dal container. Entrare nuovamente nel container loggandosi con il nuovo utente appena creato.Per terminare, effettuare una rimozione completa di tutti i container creati e di tutte le immagini presenti

Esercizio 5

Eseguire un container interattivo e testuale con sistema ubuntu e volume bind, dopo aver creato alcuni file, verificare il funzionamento.

Esercizio 6

Creare un volume named ed assicurarsi non siano stati creati altri volumi. Eseguire un container interattivo e testuale con sistema ubuntu e montare il volume. Creare un nuovo container collegato allo stesso volume. Dopo aver creato alcuni file, verificare che entrambi i container possano accedervi Eliminare infine il volume

Esercizio 7

Effettuare una rimozione completa di tutti i container creati e di tutte le immagini presenti

Esercizio 8

Creare una rete specificando ip della subnet, maschera di rete e il nome della rete, che dovrà essere “reteA”. Creare allo stesso modo “reteB” e “reteC”, quest ultima senza specificare alcun indirizzo della subnet. verificare le proprietà di queste reti ed eliminare “reteC”. Eseguire ora un container collegato direttamente a “netA” e vedere se effettivamente è connesso a tale rete. Creare poi un nuovo container e con una seconda operazione collegarlo a “reteB”. Scollegarlo e ricollegarlo e vedere se risulta connesso o meno alla rete. Verificare poi che esistano le due reti create. Eliminare poi tutti i container creati per crearne due nuovi con permessi di amministratori di rete e chiamandoli “A_host” e “B_host” crearne anche un terzo da usare come gateway tra i due ed effettuare il ping tra “A_host” e “B_host”. Creare infine su “C_host” un server http nginx e verificare il suo funzionamento. Pulire tutto l’ecosistema docker

Soluzione

docker run ubuntu /etc/hostname

docker run -d ubuntu

```
docker run -ti immagine

docker run -u utente immagine [argomenti]

docker run -w directory immagine [argomenti]

docker run -e FOO=foo immagine [argomenti]

docker run -h hostname immagine [argomenti]

docker run --name nome_container immagine [argomenti]

docker run --rm immagine [argomenti]

docker container prune

docker ps

docker ps -a

docker create immagine

docker start -ia container

docker exec -it ubuntu bash -c “nano /etc/hostname”

docker run -ti -v <percorsoHost>:<percorsoContainer> <immagine>

docker volume create <nomeVolume>

docker volume list

docker run -ti -v <nomeVolume>:<percorsoContainer> <immagine>

docker volume rm <nomeVolume>

docker network create <nomeInterfaccia>

docker network create --subnet <IPSubnet>/<mascheraDiRete> <nomeInterfaccia>

docker run --net <interfaccia> <immagine>

docker network ls

docker network connect <rete> <contaner>
```

```
docker network disconnect <rete> <container>

docker run -ti --net pippoNet --cap-add=NET_ADMIN --name pippo ubuntu

docker run -d -p [indirizzoIP:]portaHost:portaContainer immagine

docker build [-t tag] percorso //PER CREARE DOCKERFILE

docker image prune -a

docker system prune -a
```

Alternativa all'eserizio 2 senza creare una rete custom

```
ALTERNATIVA ESERCIZIO 2:

docker network create nome_rete

docker run -it --name container1 --network nome_rete -d nome_immagine

docker run -it --name container2 --network nome_rete -d nome_immagine

docker run -it --name container3 --network nome_rete -d nome_immagine

docker exec -it container1 bash

apt-get update

apt-get install iputils-ping

ping container2
```

Scansiona il codice QR per visionare i due video che spiegano il funzionamento base di Docker (video di Ivan Sollazzo https://youtu.be/9AikG4nbvtc)

Comandi usati nel video

```
docker run -it ubuntu:latest

docker run ubuntu cat /etc/hostname

docker run ubuntu cat /etc/hostname ; cat /etc/lsb-release

docker run -d ubuntu apt-get update

docker start nomeCont

docker exec -d nomeCont apt-get update

docker run -it -w /etc ubuntu

docker run -it -e DISNEY=topolino -e DREAMWORKS=shrek ubuntu

echo $DISNEY

echo $DREAMWORKS

docker run -it -h pippo ubuntu

cat /etc/hostname

docker run -it --name paperino ubuntu

docker run -it —-name usaegetta --rm ubuntu

docker ps

docker ps -a

docker rm paperino
```

docker container prune ***// TERMINA SOLO QUELLI IN STOPPED !!!***

docker start -i paperino ***// NON SI PUO' METTERE LA -t INFATTI il run implica la creazione, lo start ha bisogno di un create precedente !!!***

docker exec -it paperino bash ***// SOLO SE HAI FATTO PRIMA START !!!***

docker logs paperino

docker top paperino

docker stats paperino

docker diff paperino

docker inspect paperino

docker insepct paperino | grep IPAddress OPPURE | findstr IPAddress

docker commit paperino prova

docker image list

docker image rm ubuntu

docker image prune --all

docker cp container:percorsoContainer percorsoHost

docker cp percorsoHost container : percorsoContainer

docker export paperino > my_container.tar

inside container adduser nomeUser

docker exec -u myuser -it container_id bash

docker run -it -v percorsoHost:percorsoContainer Ubuntu ***//PER VOLUMI DI TIPO BIND***

docker volume create nomeVolume

docker volume list

docker run -it -v nomeVolume:percorsoContainer ubuntu

docker volume inspect nomeVolume

docker volume rm nomeVolume

docker network create nomeNetwork

docker network inspect nomeNetwork

docker network rm nomeRete

docker network create --subnet <IPSubnet>/<mascheraDiRete> nomeRete

docker run -it --net nomeRete ubuntu

docker inspect container

docker network connect nomeRete nomeContainer

docker network disconnect nomeRete nomeContainer

docker network ls

docker network create rete1

docker run -it --net rete1 --cap-add=NET_ADMIN --name pippo ubuntu

docker exec -it bash

docker exec -t nomeHost comando

apt-get update; apt-get install iproute2 iputils-ping

ip route add ipSubnet_da_raggiungere/mask via ipContainerGateway (della rete da dove sto lanciando ip route)

docker run -d -p 8080:80 nginx ***accedere poi dal browser all'indirizzo 127.0.0.1:8080. la pagina che vedi è nel percorso /usr/share/nginx/html***