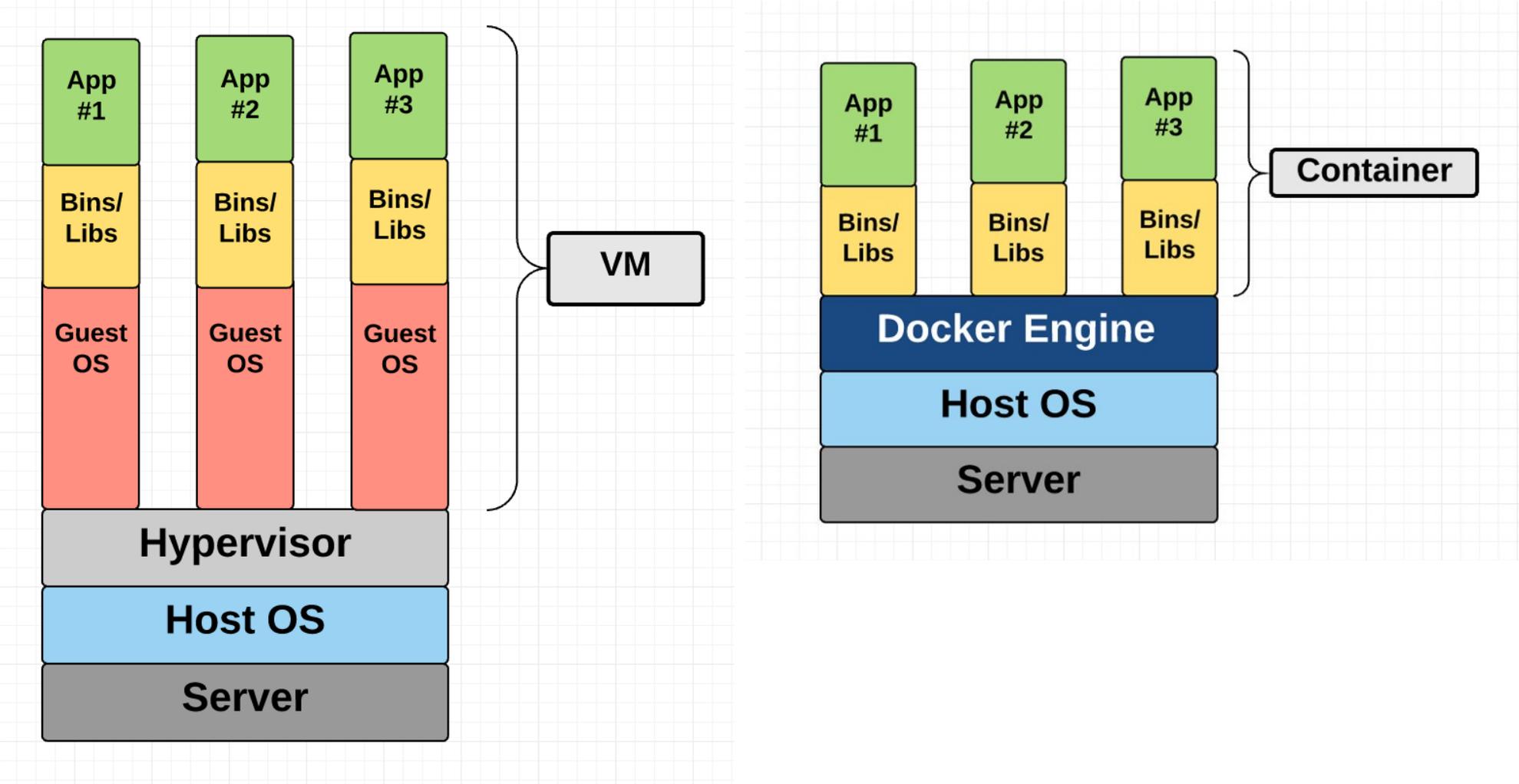
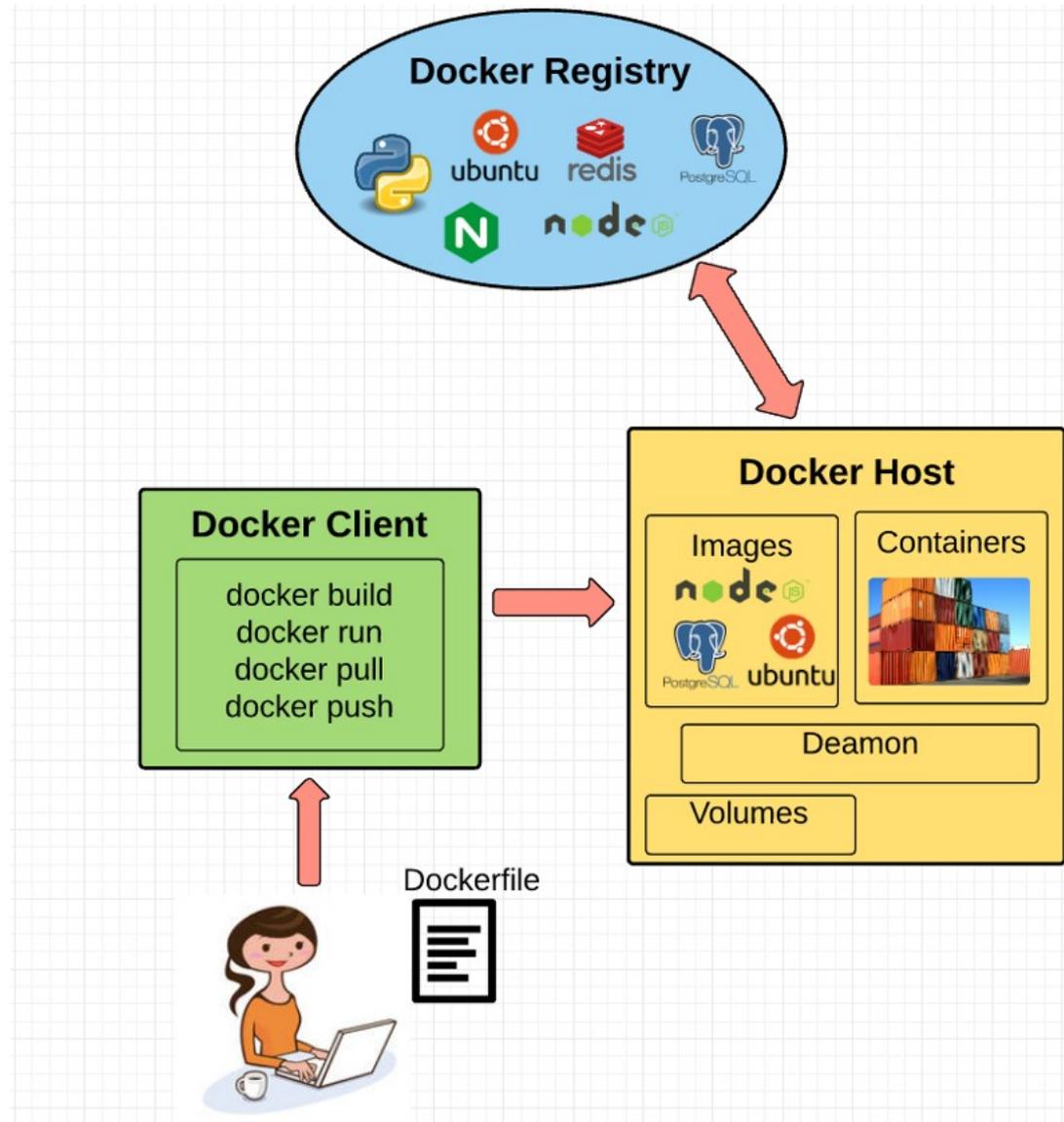


Docker

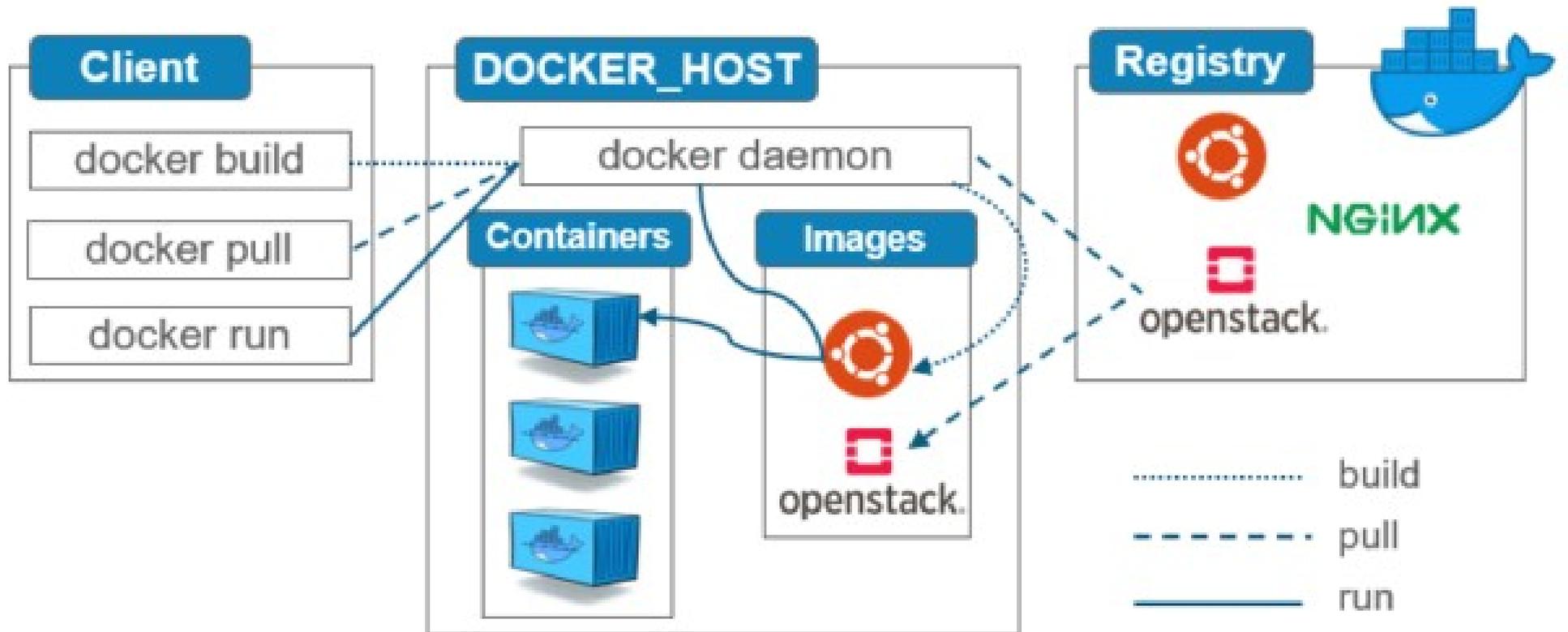
Docker



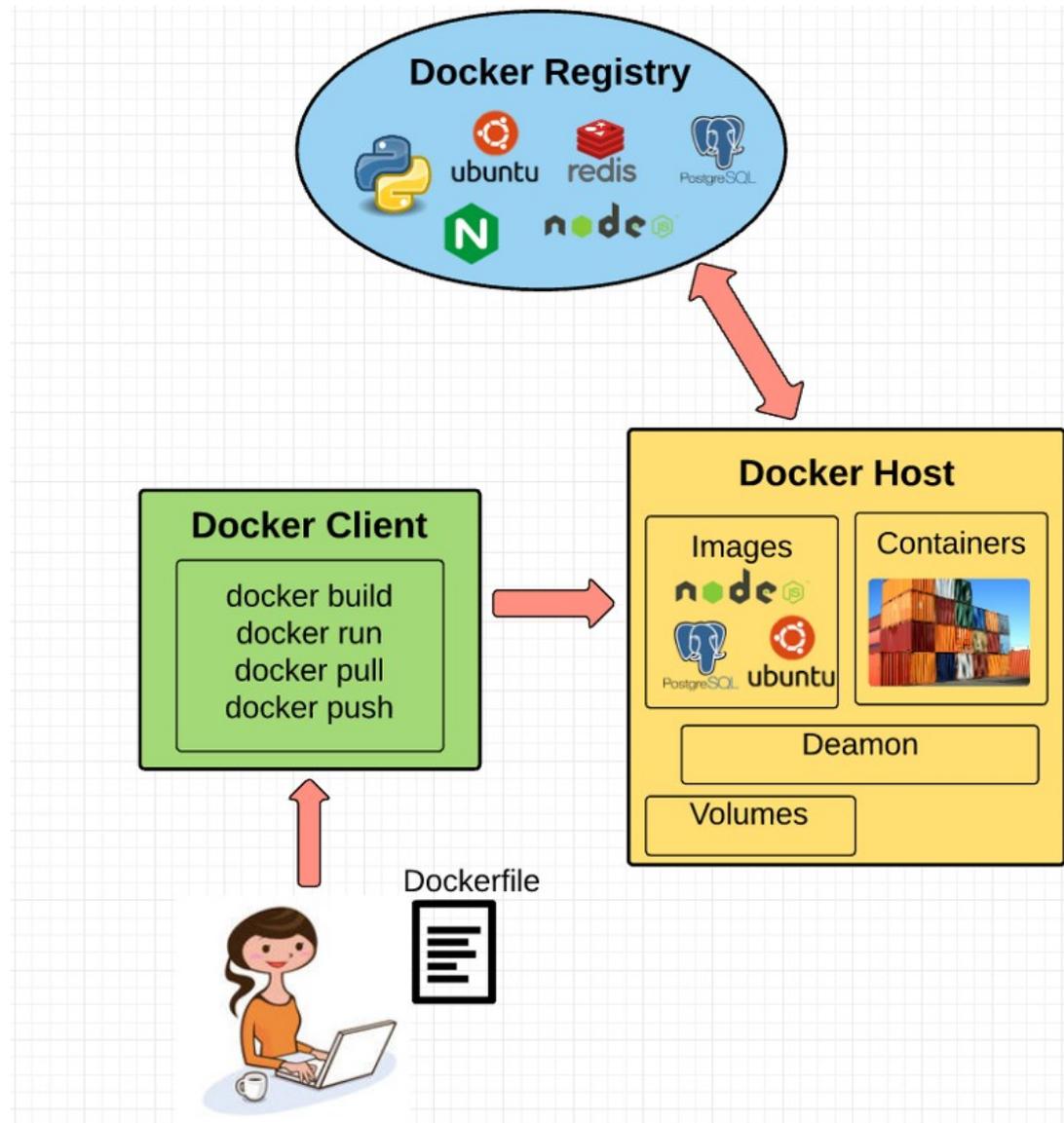
Docker



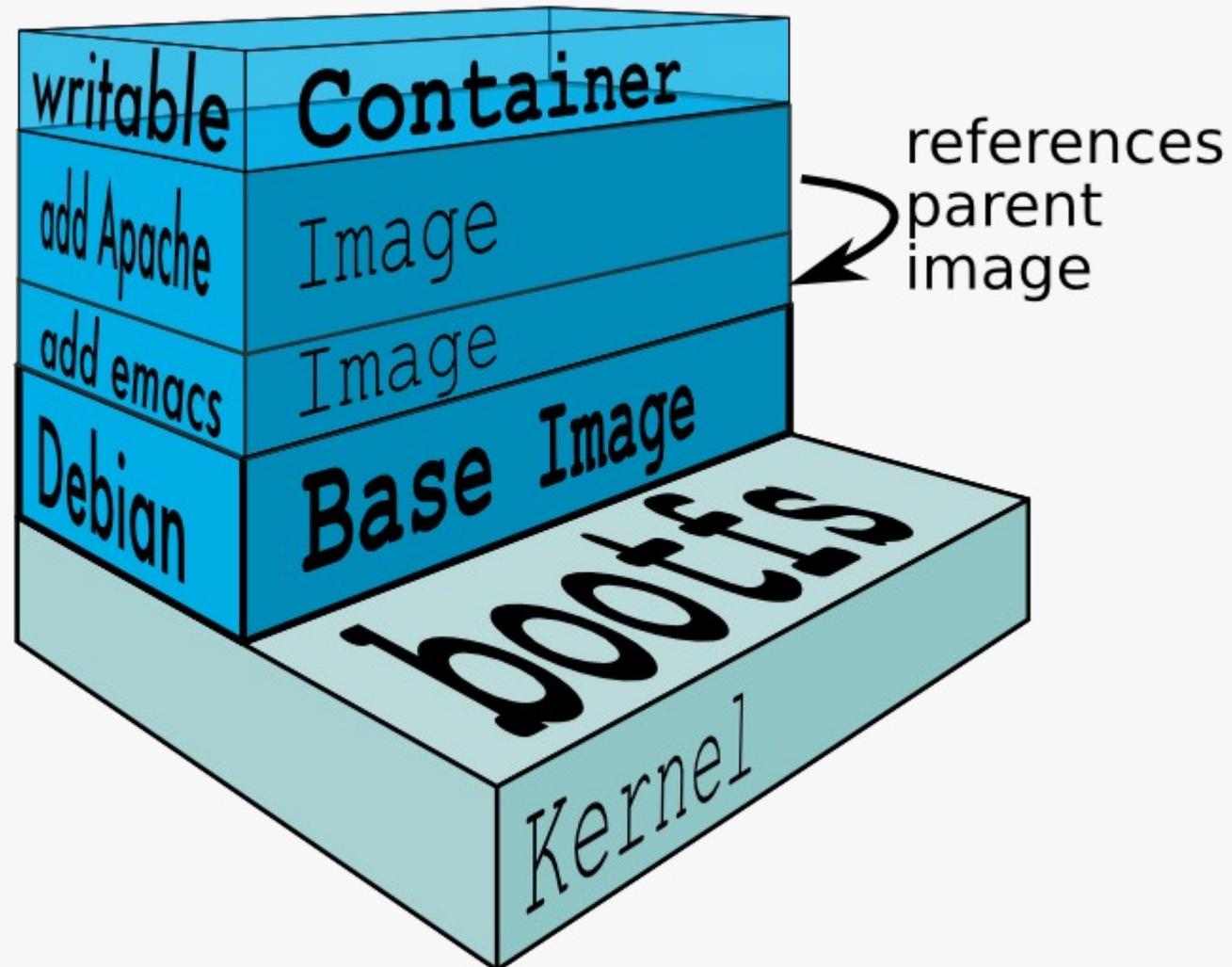
Docker



Docker



Docker



```
$ sudo docker run -i -t ubuntu /bin/bash
```

- Docker, по порядку, делает следующее:
- скачивает образ ubuntu: docker проверяет наличие образа ubuntu на локальной машине, и если его нет — то скачивает его с Docker Hub. Если же образ есть, то использует его для создания контейнера;
- создает контейнер: когда образ получен, docker использует его для создания контейнера;
- инициализирует файловую систему и монтирует read-only уровень: контейнер создан в файловой системе и read-only уровень добавлен образ;
- инициализирует сеть/мост: создает сетевой интерфейс, который позволяет docker-у общаться хост машиной;
- Установка IP адреса: находит и задает адрес;
- Запускает указанный процесс: запускает ваше приложение;

- Постоянные данные, генерируемые Docker и используемые контейнерами Docker, хранятся в томах.
- Они полностью управляются Docker через CLI Docker или API Docker
- . Тома работают на контейнерах Windows и Linux.
- Вместо того чтобы сохранять данные в доступном для записи слое контейнера, всегда полезно использовать для него тома.
- Содержимое тома существует вне жизненного цикла контейнера, поэтому использование тома не увеличивает размер контейнера.
- Вы можете использовать флаг `-v` или `-mount`, чтобы запустить контейнер с томом. В этом примере команды вы используете том `geekvolume` с контейнером `geekflare`.

```
docker run -d --name geekflare -v geekvolume:/app nginx:latest
```

docker pull ubuntu:18.10

docker images

```
C:\Users\Alexa>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.10              0bfd76efee03      20 hours ago      73.7MB
```

docker run ubuntu:18.10 echo 'hello from ubuntu'

```
C:\Users\Alexa>docker run ubuntu:18.10 echo 'hello from ubuntu'
'hello from ubuntu'
```

docker run -it ubuntu:18.10 /bin/bash

docker run -it --rm ubuntu:18.10 /bin/bash

**docker run -d --rm -p 8000:80 -p 8443:443 --name pandorafms
pandorafms/pandorafms:latest**

docker stop <CONTAINER_ID>

docker rm <CONTAINER_ID>

docker pull ubuntu:18.10

docker images

```
C:\Users\Alexa>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              18.10              0bfd76efee03       20 hours ago       73.7MB
```

docker run ubuntu:18.10 echo 'hello from ubuntu'

```
C:\Users\Alexa>docker run ubuntu:18.10 echo 'hello from ubuntu'
'hello from ubuntu'
```

docker run -it ubuntu:18.10 /bin/bash

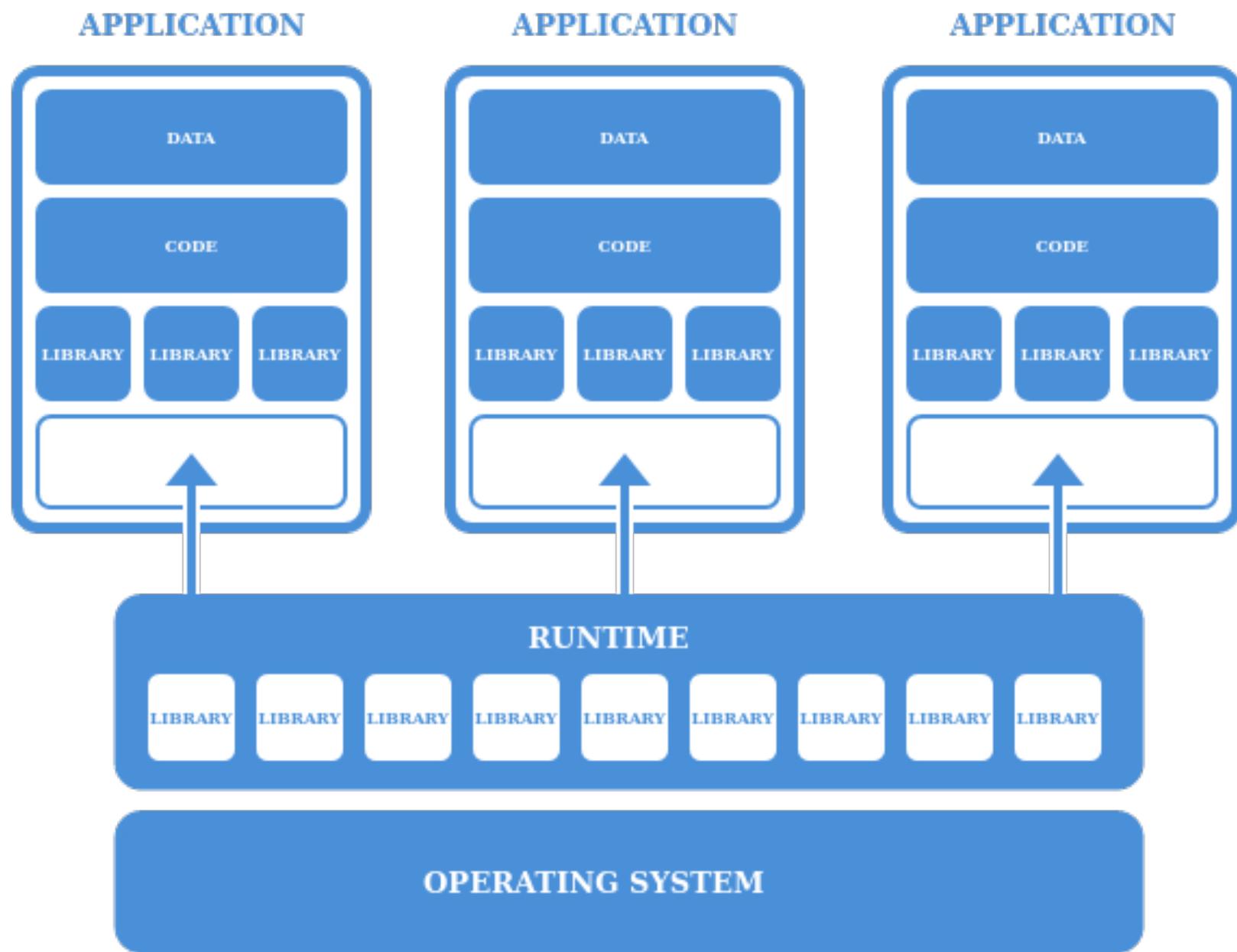
docker run -it --rm ubuntu:18.10 /bin/bash

**docker run -d --rm -p 8000:80 -p 8443:443 --name pandorafms
pandorafms/pandorafms:latest**

docker stop <CONTAINER_ID>

docker rm <CONTAINER_ID>

Flatpak



flatpak remotes

**\$ flatpak remote-add --if-not-exists flathub
https://flathub.org/repo/flathub.flatpakrepo**

\$ flatpak remote-delete flathub

\$ flatpak search gimp

**\$ flatpak install flathub org.gimp.GIMP
или \$ flatpak install gimp**

\$ flatpak run org.gimp.GIMP

\$ flatpak update

\$ flatpak list

или \$ flatpak list —app

\$ flatpak uninstall org.gimp.GIMP

\$ flatpak uninstall --unused

\$ flatpak repair

\$ flatpak history