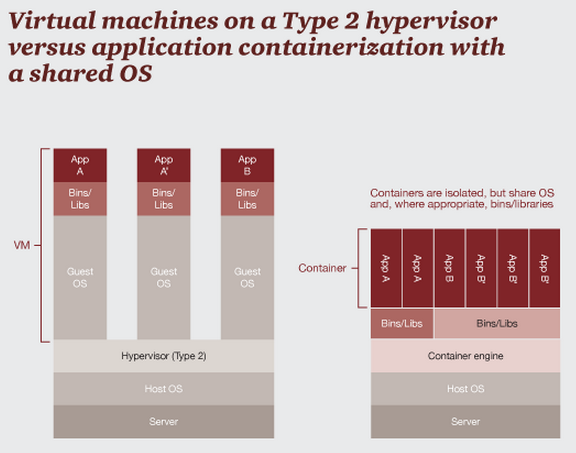
**[Docker container vs Virtual Machine](http://www.bogotobogo.com/DevOps/Docker/Docker_Container_vs_Virtual_Machine.php)**

*Whenever we use some of Google functionality - Search, Gmail, Google Docs, whatever - we're issued a new container.*

**Docker container vs Virtual machine**

Docker is an open source application deployment container that evolved from the [LinuX Containers (LXCs)](http://www.bogotobogo.com/Linux/linux_LXC_Linux_Container_Install_Run.php" \t "_blank) used for the past decade. LXCs allow different applications to share operating system (OS) kernel, CPU, and RAM.

Docker allow us to run an **application and its dependencies** in resource-isolated processes.



**Credit:** [Open source application containers](http://www.pwc.com/us/en/technology-forecast/2014/cloud-computing/features/open-source-application-deployment-containers.jhtml) (<http://www.pwc.com/us/en/technology-forecast/2014/cloud-computing/features/open-source-application-deployment-containers.html>)

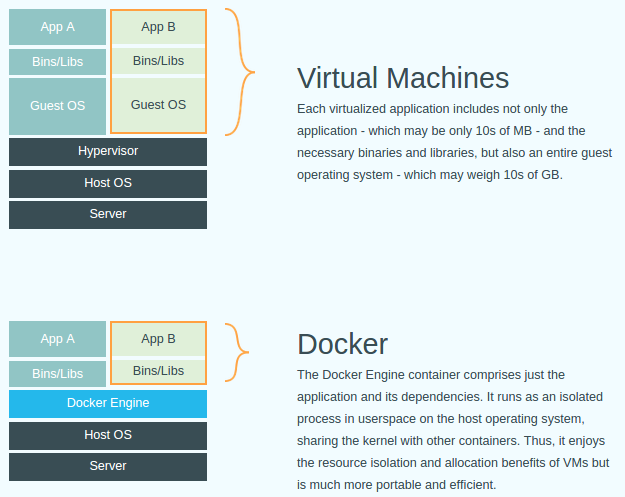
"The VM model blends **an application, a full guest OS, and disk emulation**. In contrast, the container model uses just the application's dependencies and runs them directly on a host OS. Containers do not launch a separate OS for each application, but share the host kernel while maintaining the isolation of resources and processes where required".

"The fact that a container does not run its own OS instance reduces dramatically the overhead associated with starting and running instances. Startup time can typically be reduced from 30 seconds (or more) to one-tenth of a second. The number of containers running on a typical server can reach dozens or even hundreds. The same server, in contrast, might support 10 to 15 VMs".

"In Docker, applications and their dependencies, such as binaries and libraries, all become part of a base working image".

The isolation from OS kernel provided by containers is less robust than that of real virtual machines, which have independent kernels and run on top of a hypervisor. However, sharing the kernel allows containers to run faster and offers management features which are not easy with VMs.

The picture is from [https://www.docker.com/whatisdocker/ - How is this different from Virtual Machines?](https://www.docker.com/whatisdocker/).



**Container apps**

Containers make our app shareable.

1. All the needs of our app is defined in a text file (**Dockerfile**)
2. A sample of Dockerfile - the following 4 lines construct the whole environment that's production ready:

FROM ubuntu:14:04

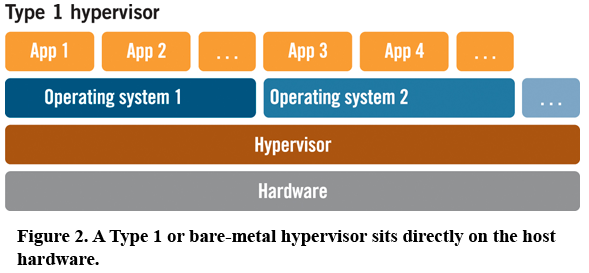
RUN apt-get install -y redis-server

EXPOSE 6379

ENTRYPOINT ["/usr/bin/redis-server"]

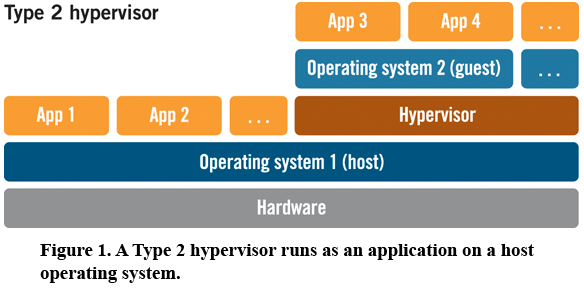
1. Containers contain everything our app needs:
   1. Binaries
   2. Libraries
   3. File system
2. Containers use the following items from the host:
   1. Networks
   2. Kernel

**Type 1 / Type2 Hypervisor**



**Type-1 bare-metal hypervisors**

1. Xen
2. VMware ESX/ESXi
3. Microsoft Hyper-V



**Type-2 hosted hypervisors**

1. VirtualBox
2. VMware Workstation/Player

Picture source: [What's the difference between a 'Type 1' hypervisor and a 'Type 2' hypervisor?](http://www.virtzone.net/the-difference-between-a-type-2-hypervisor-and-a-type-1-hypervisor/)( <http://www.virtzone.net/the-difference-between-a-type-2-hypervisor-and-a-type-1-hypervisor/>)