Welcome! We are excited that you want to learn Docker. The Docker Quickstart training module teaches you how to:

1. Set up your Docker environment (on this page)
2. Build and run your image
3. Share images on Docker Hub

Docker concepts

Docker is a platform for developers and sysadmins to **build, run, and share** applications with containers. The use of containers to deploy applications is called *containerization*. Containers are not new, but their use for easily deploying applications is.

Containerization is increasingly popular because containers are:

- **Flexible**: Even the most complex applications can be containerized.
- **Lightweight**: Containers leverage and share the host kernel, making them much more efficient in terms of system resources than virtual machines.
- **Portable**: You can build locally, deploy to the cloud, and run anywhere.
- **Loosely coupled**: Containers are highly self sufficient and encapsulated, allowing you to replace or upgrade one without disrupting others.
- **Scalable**: You can increase and automatically distribute container replicas across a datacenter.
- **Secure**: Containers apply aggressive constraints and isolations to processes without any configuration required on the part of the user.

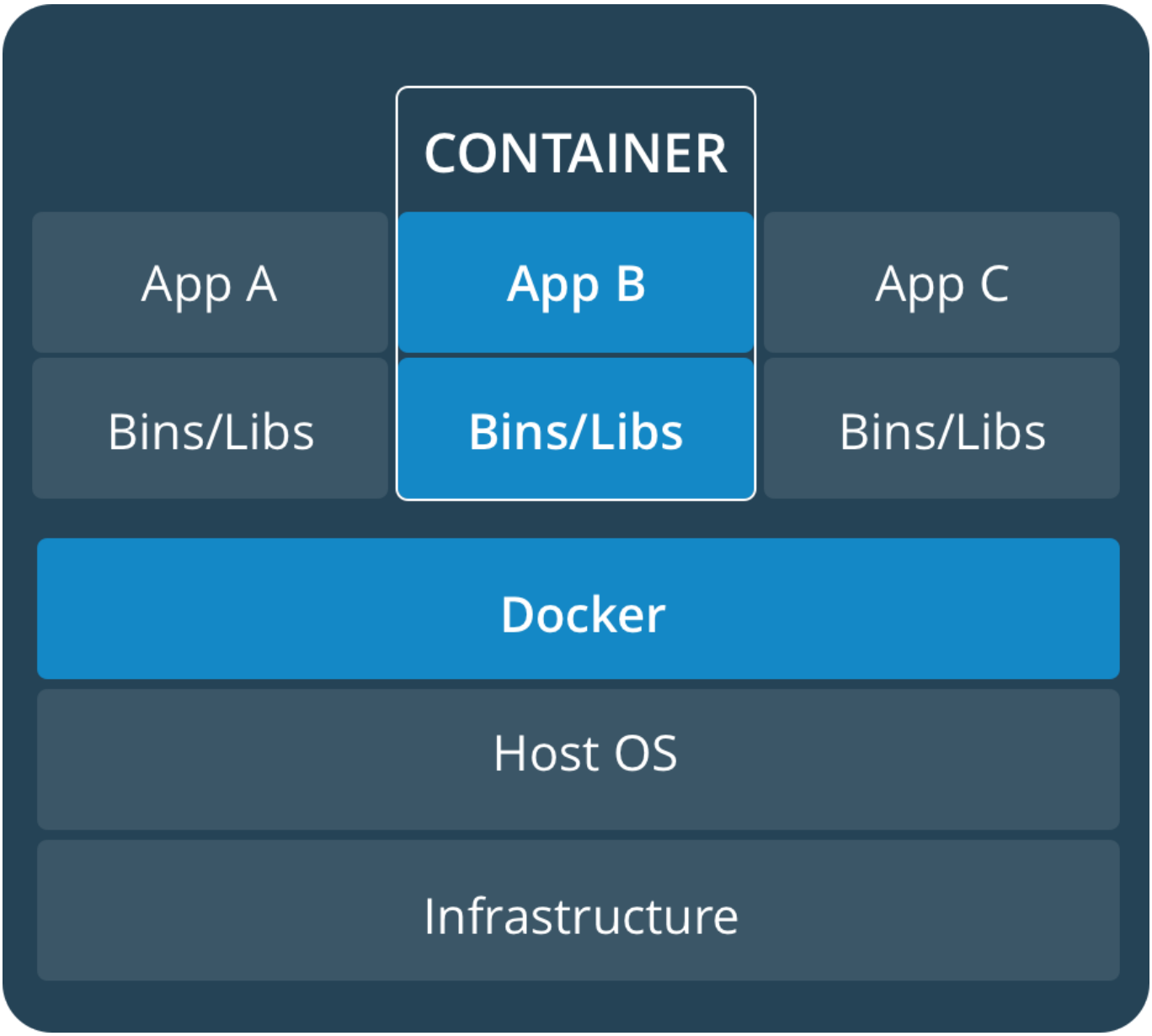
Images and containers

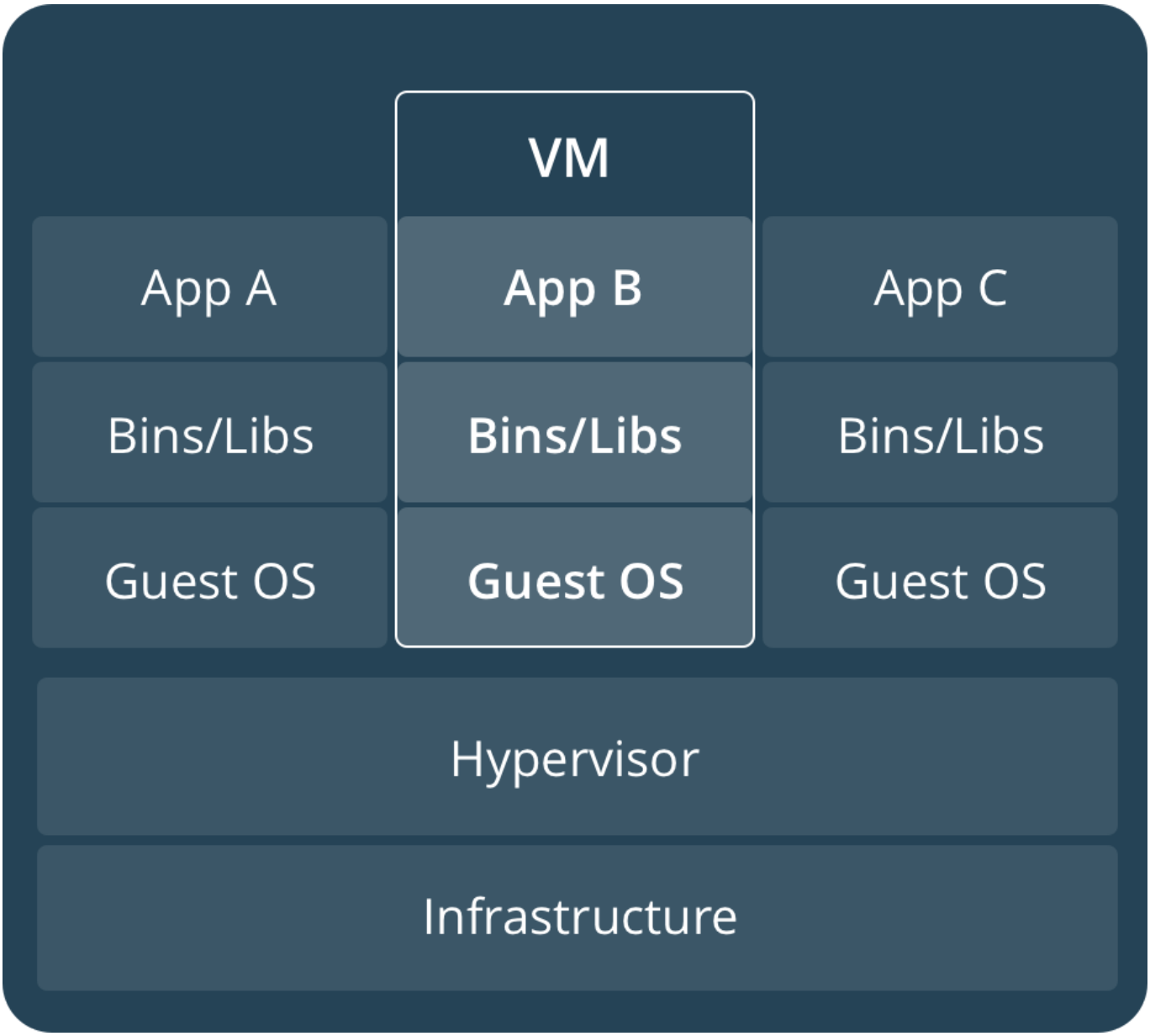
Fundamentally, a container is nothing but a running process, with some added encapsulation features applied to it in order to keep it isolated from the host and from other containers. One of the most important aspects of container isolation is that each container interacts with its own private filesystem; this filesystem is provided by a Docker **image**. An image includes everything needed to run an application - the code or binary, runtimes, dependencies, and any other filesystem objects required.

Containers and virtual machines

A container runs *natively* on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking no more memory than any other executable, making it lightweight.

By contrast, a **virtual machine** (VM) runs a full-blown "guest" operating system with *virtual* access to host resources through a hypervisor. In general, VMs incur a lot of overhead beyond what is being consumed by your application logic.


CONTAINER
App A
App B
App C
Bins/Libs
Bins/Libs
Bins/Libs
Docker
Host OS
Infrastructure



Set up your Docker environment

Download and install Docker Desktop

Docker Desktop is an easy-to-install application for your Mac or Windows environment that enables you to start coding and containerizing in minutes. Docker Desktop includes everything you need to build, run, and share containerized applications right from your machine.

Follow the instructions appropriate for your operating system to download and install Docker Desktop:

- Docker Desktop for Mac
- Docker Desktop for Windows

Test Docker version

After you've successfully installed Docker Desktop, open a terminal and run `docker --version` to check the version of Docker installed on your machine.

```
$ docker --version
Docker version 19.03.5, build 633a0ea
```

Test Docker installation

1. Test that your installation works by running the hello-world

```
    $ docker run hello-world

    Unable to find image 'hello-world:latest' locally
    latest: Pulling from library/hello-world
    ca4f61b1923c: Pull complete
    Digest:
sha256:ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
    Status: Downloaded newer image for hello-world:latest

    Hello from Docker!
    This message shows that your installation appears to be working
correctly.
    ...
```

2. Run `docker image ls` to list the `hello-world` image that you downloaded to your machine.

3. List the `hello-world` container (spawned by the image) which exits after displaying its message. If it is still running, you do not need the `--all` option:

```
    $ docker ps --all

    CONTAINER ID     IMAGE           COMMAND      CREATED            STATUS
    54f4984ed6a8     hello-world     "/hello"     20 seconds ago     Exited
(0) 19 seconds ago
```

Conclusion

At this point, you've installed Docker Desktop on your development machine, and ran a quick test to ensure you are set up to build and run your first containerized application.