Introduction to Docker

CLARUSWAY©
WAY TO REINVENT YOURSELF

Table of Contents

CLARUSWAY©
WAY TO REINVENT YOURSELF

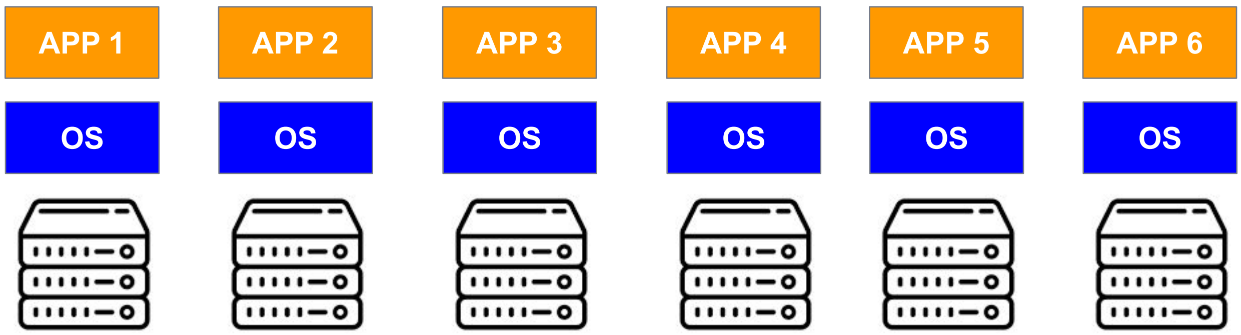
1 Before Docker

Bare Metal

OS = 2 CPU + 2 GB RAM

APP = 1 CPU + 500 MB RAM

24 CPU
18 GB RAM

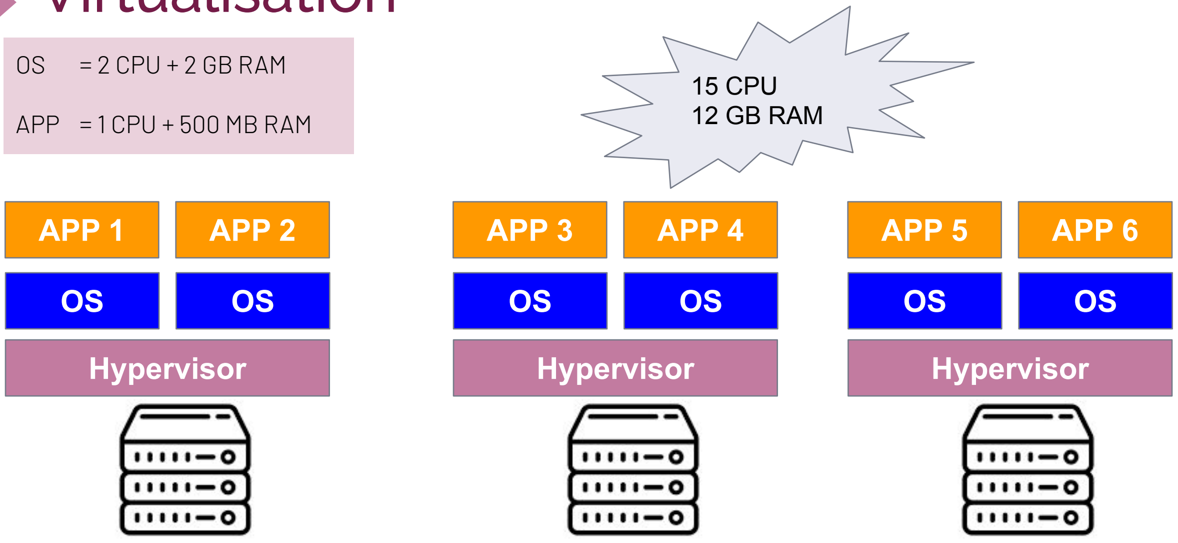


HARDWARE = 4 CPU + 3 GB RAM

Virtualisation

OS = 2 CPU + 2 GB RAM

APP = 1 CPU + 500 MB RAM



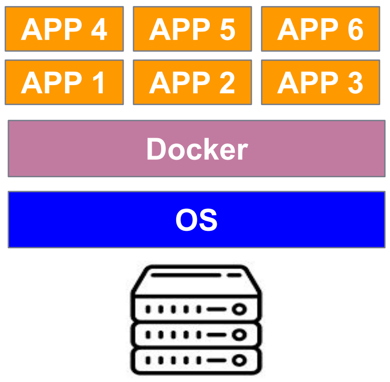
HARDWARE = 5 CPU + 4 GB RAM

Bare Metal

Container

OS = 2 CPU + 2 GB RAM

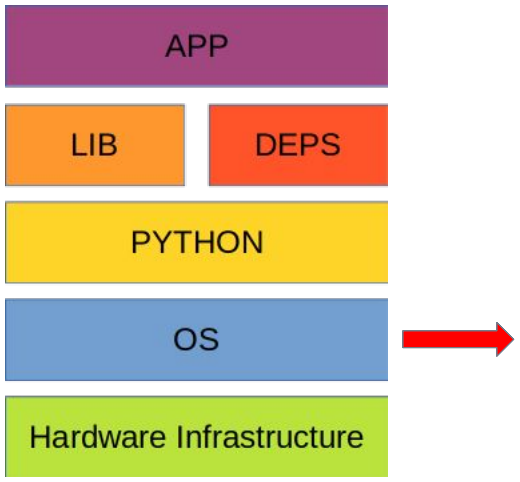
APP = 1 CPU + 500 MB RAM



HARDWARE = 9 CPU + 6 GB RAM

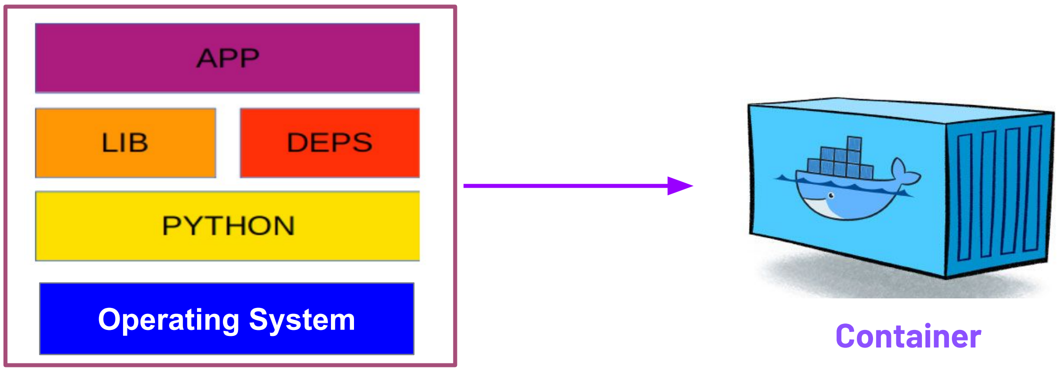
2 What is Container?

What is Container?

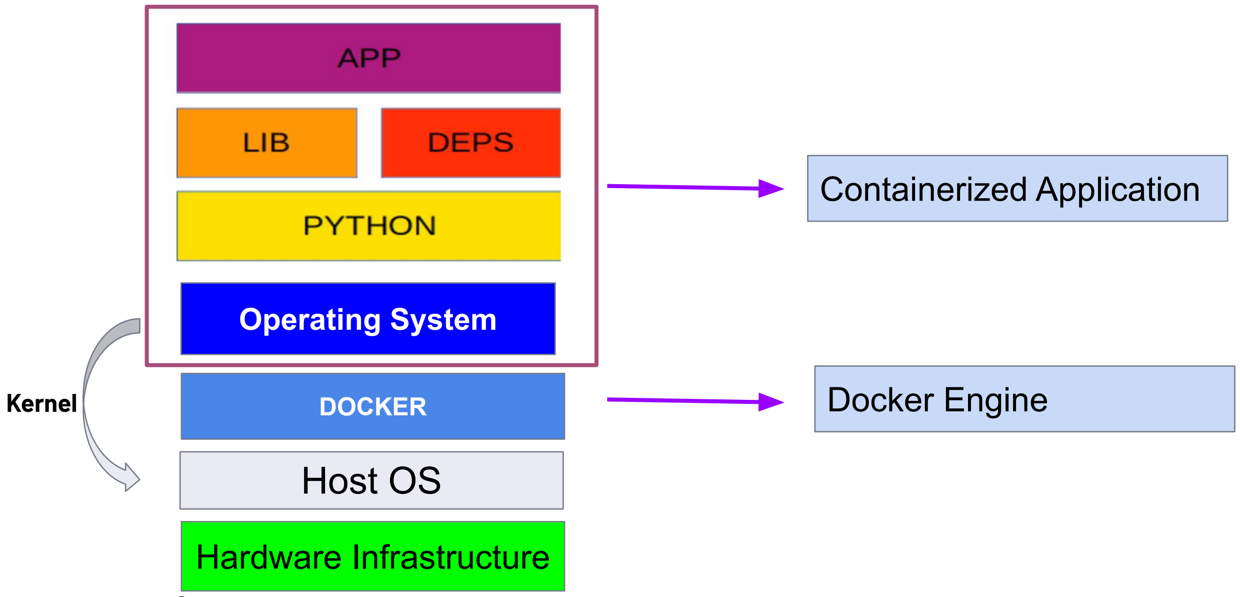


Kernel, Media Player, Browser, Calculator, Solitaire, GUI, Calendar, Paint ...

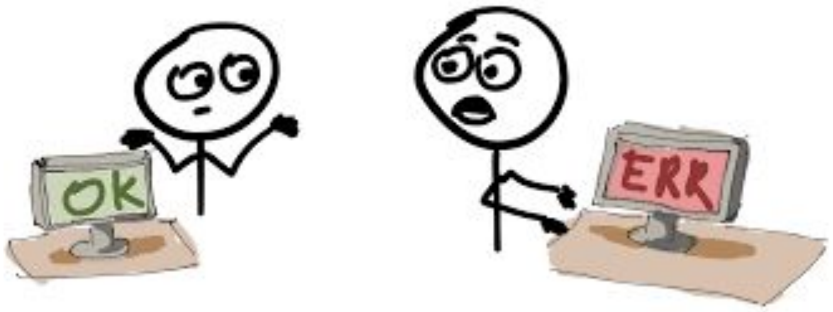
What is Container?



What is Container?



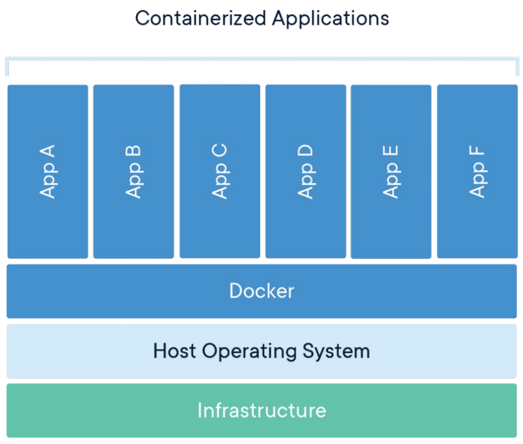
What is Container?



What is Container?

What is Container?

A **container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

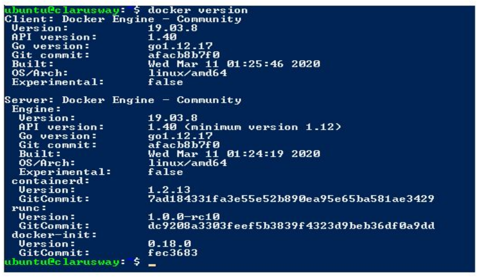


What is docker ???

What is Docker?

"DOCKER" refers to several things. This includes an open-source community project which started in 2013; tools from the open-source project; Docker Inc., the company that is the primary supporter of that project; and the tools that the company formally supports.

- ➢ Docker as a “Company”
- ➢ Docker as a “Product”
- ➢ Docker as a “Platform”
- ➢ Docker as a “CLI Tool”
- ➢ Docker as a “Computer Program”


```
ubuntu@clarusway:~$ docker version
Client: Docker Engine - Community
 Version:           19.03.8
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        afacb8b7f0
 Built:             Wed Mar 11 01:25:46 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.8
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       afacb8b7f0
  Built:            Wed Mar 11 01:24:19 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
ubuntu@clarusway:~$ _
```

What is Docker?

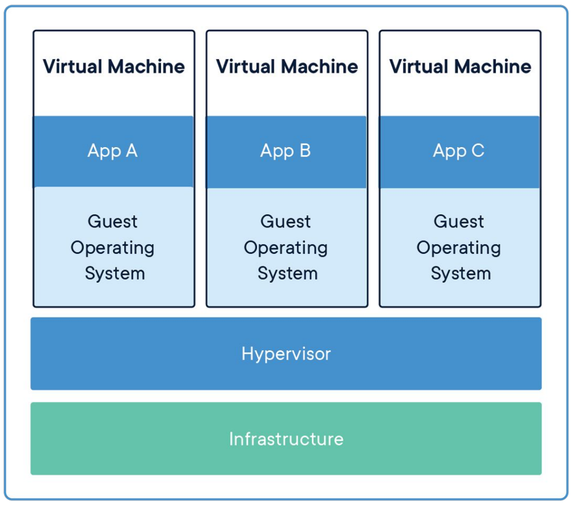
3 Containers vs. Virtual Machines

CLARUSWAY©
WAY TO REINVENT YOURSELF

Containers vs. Virtual Machines

A virtual machine (VM) is software that runs programs or applications without being tied to a physical machine.

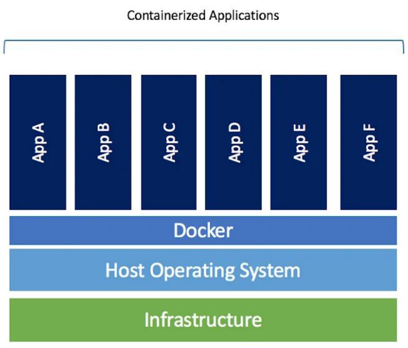
Virtual Machines are built over the physical hardware, there is a hypervisor layer which sits between physical hardware and operating systems.

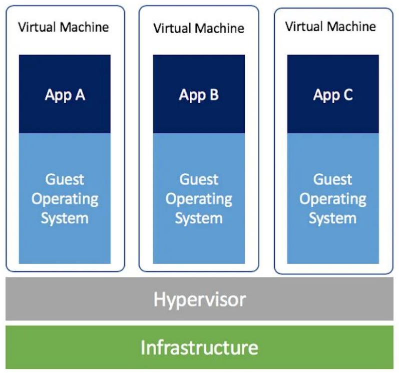


CLARUSWAY©
WAY TO REINVENT YOURSELF

Containers vs. Virtual Machines

Unlike virtual machines where hypervisor divides physical hardware into parts, Containers are like normal operating system processes.





CLARUSWAY©
WAY TO REINVENT YOURSELF

Containers vs. Virtual Machines

Virtual Machine

Containers

Docker containers are executed with the Docker engine rather than the hypervisor. Containers are therefore smaller than Virtual Machines and enable faster startup with better performance, less isolation and greater compatibility possible due to sharing of the host's kernel. Hence, it looks very similar to the residential flats system where we share resources of the building.

CLARUSWAY©
WAY TO REINVENT YOURSELF

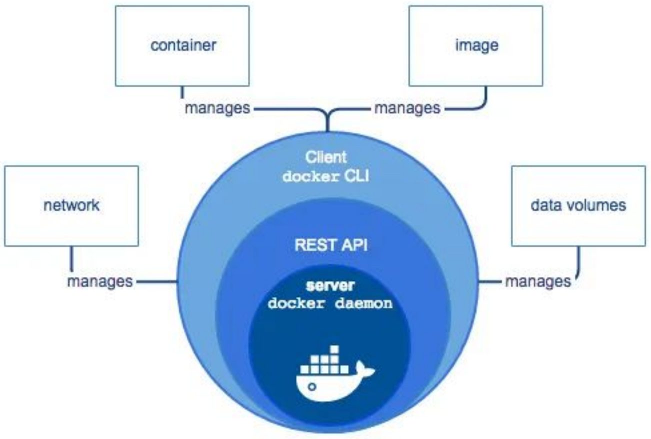
Containers vs. Virtual Machines

Virtual Machines	Docker
Each VM runs its own OS	All containers share the same kernel of the host
Boots uptime is in minutes	Containers instantiate in seconds
Not version controlled	Images can be version controlled. Dockerhub is like GitHub
Cannot run more than a couple of VMS on an average laptop	Can run many Docker containers on a laptop.
Only one VM can be started from one set of VMX and VMDK files	Multiple Docker containers can be started from one Docker image

4 Docker Architecture

Docker Architecture

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface.



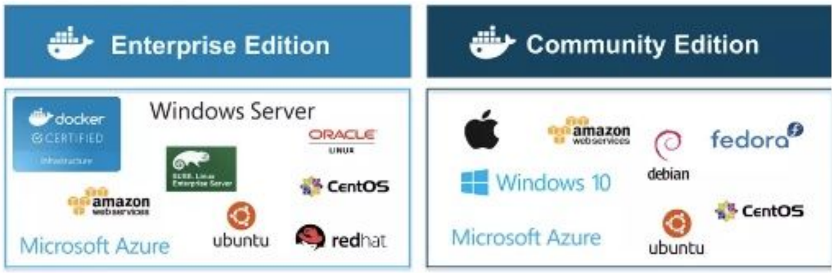
5 Terminology

Terminology

Docker Editions

•**Docker Community Edition (CE)** is ideal for Developers who are looking for experimenting with docker and creating container-based applications. It's free.

•**Docker Enterprise Edition (EE)** is a Containers-as-a-Service (CaaS) platform. Enterprise Edition Subscription packages include an integrated Docker platform and tooling for container management and security.



Terminology

Registry

•A Docker registry stores Docker images.

•**Docker Hub (Like GitHub)** is a cloud-based registry service that allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker Cloud so you can deploy images to your hosts.

•**Docker Cloud** uses the hosted Docker Cloud Registry, which allows you to publish Dockerized images on the internet either publicly or privately. Docker Cloud can also store pre-built images, or link to your source code so it can build the code into Docker images, and optionally test the resulting images before pushing them to a repository.

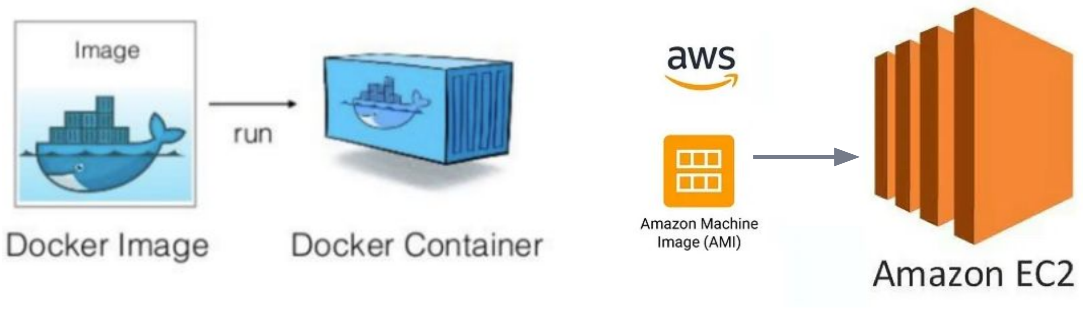
6

Images and Containers

CLARUSWAY©
WAY TO REINVENT YOURSELF

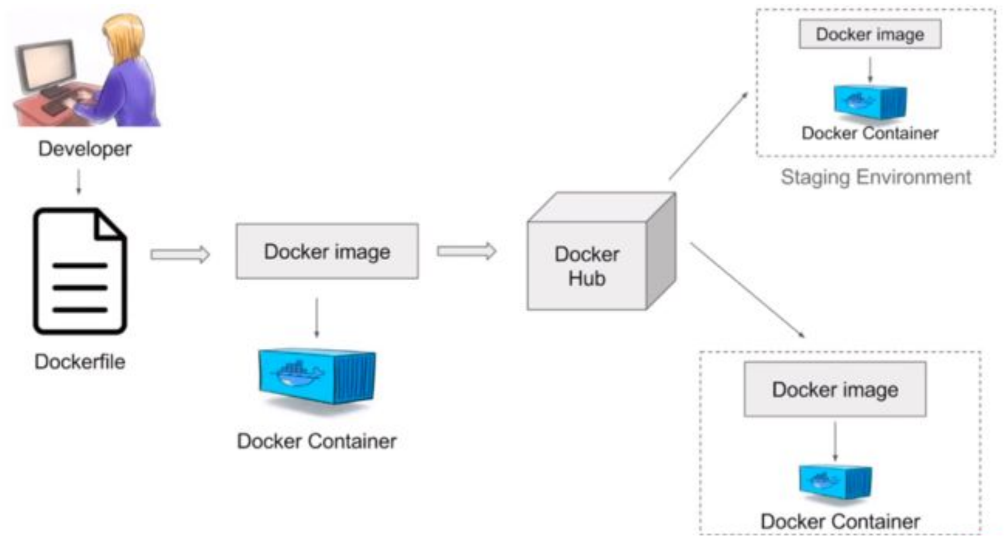
Images and Containers

- An image is a read-only template with instructions for creating a Docker container.
- A container is a runnable instance of an image.



CLARUSWAY©
WAY TO REINVENT YOURSELF

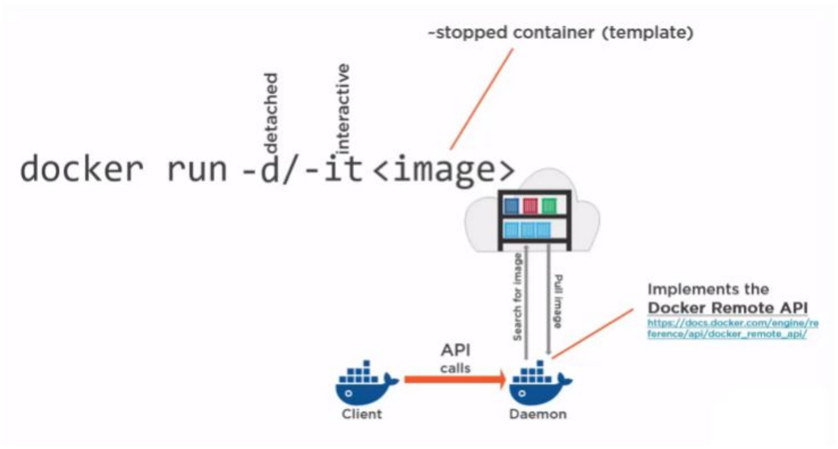
Images and Containers



3 docker run command

docker run command

docker run command is used to create a container. The docker run command provides all of the ”launch” capabilities for Docker.


-stopped container (template)
detached
interactive
docker run -d/-it <image>
Search for image
Pull image
Implements the
Docker Remote API
https://docs.docker.com/engine/reference/api/docker_remote_api/
API
calls
Client
Daemon

CLARUSWAY©
WAY TO REINVENT YOURSELF

docker run command

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

When we run this command, the following happens.

- If you do not have the ubuntu image locally, Docker pulls it from your configured registry, as though you had run docker pull ubuntu manually.
- Docker creates a new container, as though you had run a docker container create command manually.
- Docker starts the container and executes /bin/bash. Because the container is running interactively and attached to your terminal (due to the -i and -t flags), you can provide input using your keyboard while the output is logged to your terminal.
- When you type exit to terminate the /bin/bash command, the container stops but is not removed. You can start it again or remove it.

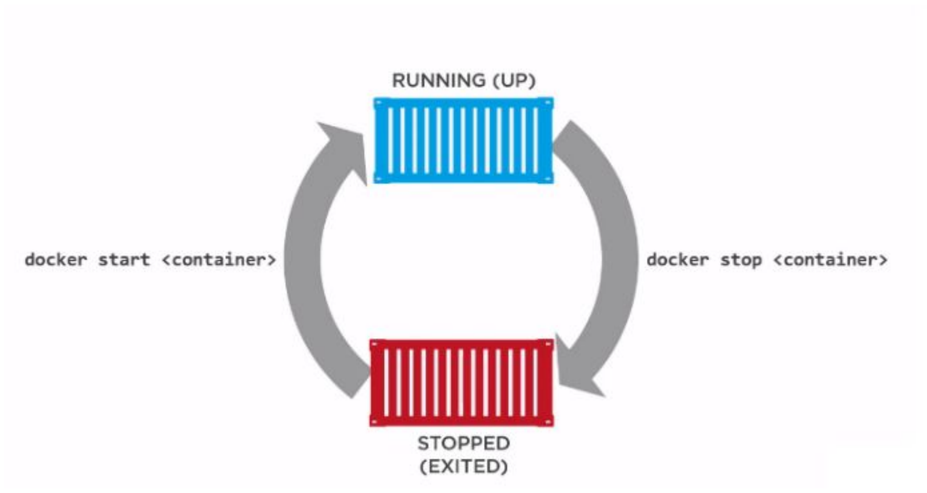
CLARUSWAY©
WAY TO REINVENT YOURSELF

4

Starting a stopped container

CLARUSWAY©
WAY TO REINVENT YOURSELF

Starting a stopped container



CLARUSWAY©
WAY TO REINVENT YOURSELF

5 Container naming

Container naming

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

- Docker will automatically generate a name at random for each container we create.
- If we want to specify a particular container name in place of the automatically generated name, we can do so using the --name flag.

6

docker container Commands

CLARUSWAY©
WAY TO REINVENT YOURSELF

docker container Commands

Command	Description
docker container attach	Attach local standard input, output, and error streams to a running container
docker container create	Create a new container
docker container exec	Run a command in a running container
docker container inspect	Display detailed information on one or more containers
docker container ls	List containers
docker container prune	Remove all stopped containers
docker container rename	Rename a container
docker container rm	Remove one or more containers

CLARUSWAY©
WAY TO REINVENT YOURSELF

THANKS!

Any questions?