**Docker Containers**

Runnable isolated instance of a set of processes and their dependencies.

A Docker container is built from a Docker image. The image lays out everything the processes that run in the container will need.

Docker Containers are managed by the Docker Daemon as part of the Docker Engine. The Docker Engine allows Docker containers to be standardized and very portable.

**Under the hood**

The underlying nature of Docker containers is based on the capabilities provided by namespaces and cgroups. Docker containers also take advantage of a file system called UnionFS. Docker manages all this in tandem in a wrapper refered to as *container format*. The container format used by default is libcontainer.

Docker containers when run on a Linux system typically share the Host OS- just as one would expect of a containerized app. *The goal is lightweight after all.*

However, in the case of Windows, Docker containers may use an additional layer of virtualization enabling you to run Linux containers on a Windows OS. This is why it's necessary to have Hyper-V and virtualization enabled when trying to install Docker on a Windows OS. (Because truly it is akin to running a container in a VM.)

**Benefits**

And they allow for all the benefits outlined in the containerization notes. i.e.(copied verbatim for convenience)

* Secure
  + Isolation and Virtualization keep your containerized apps more secure
* Standardized and thus Portable
  + Think write once run anywhere
* Lightweight
  + shares the host operating system's kernel
* Flexible and Loosely Coupled
* Scalable
  + Easy to spin up and because of this lightweight ease they can be scaled up quickly

**States of a container:**

* created
* restarting
* running
* paused
* exited
* dead

**References**

* <https://docs.docker.com/engine/api/v1.24/#31-containers>
* Docker docs <https://docs.docker.com/engine/reference/commandline/container/>
* [Containerization References](https://app.revature.com/admin-v2/)