How far do Docker containers scale?

How far do Docker containers scale?

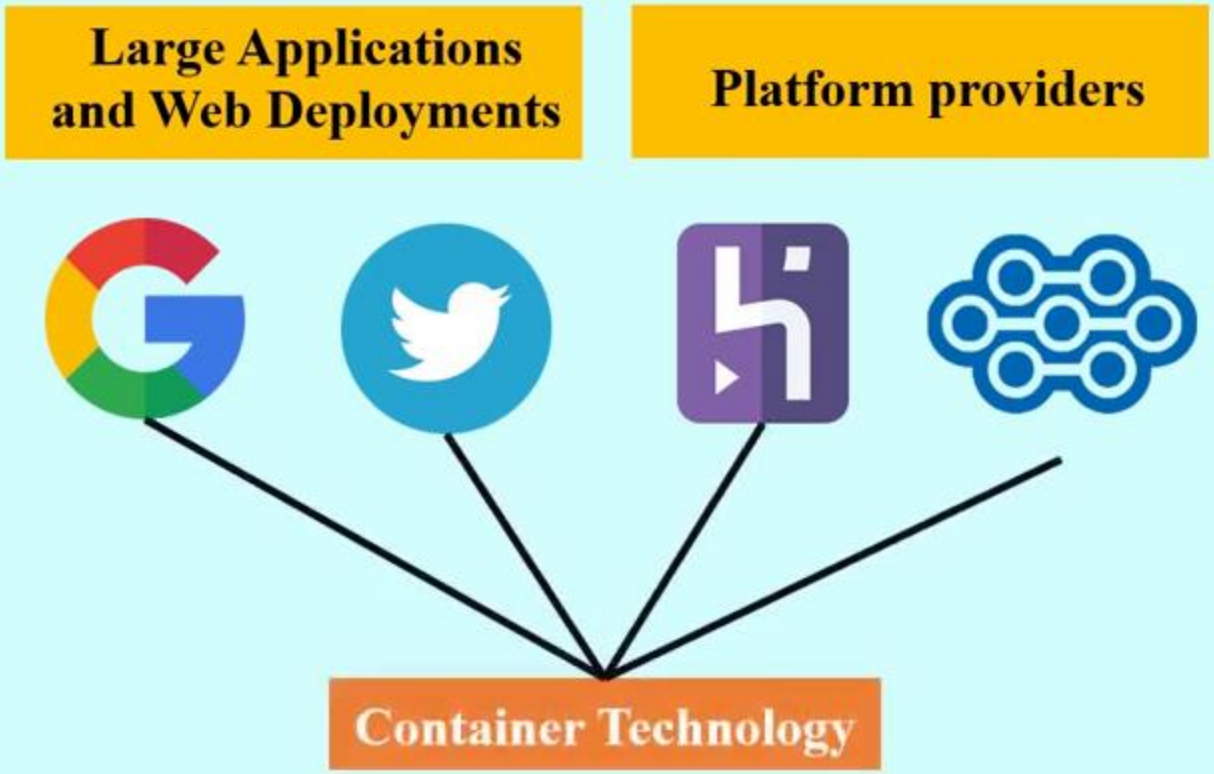
Large Applications and Web Deployments

How far do Docker containers scale?

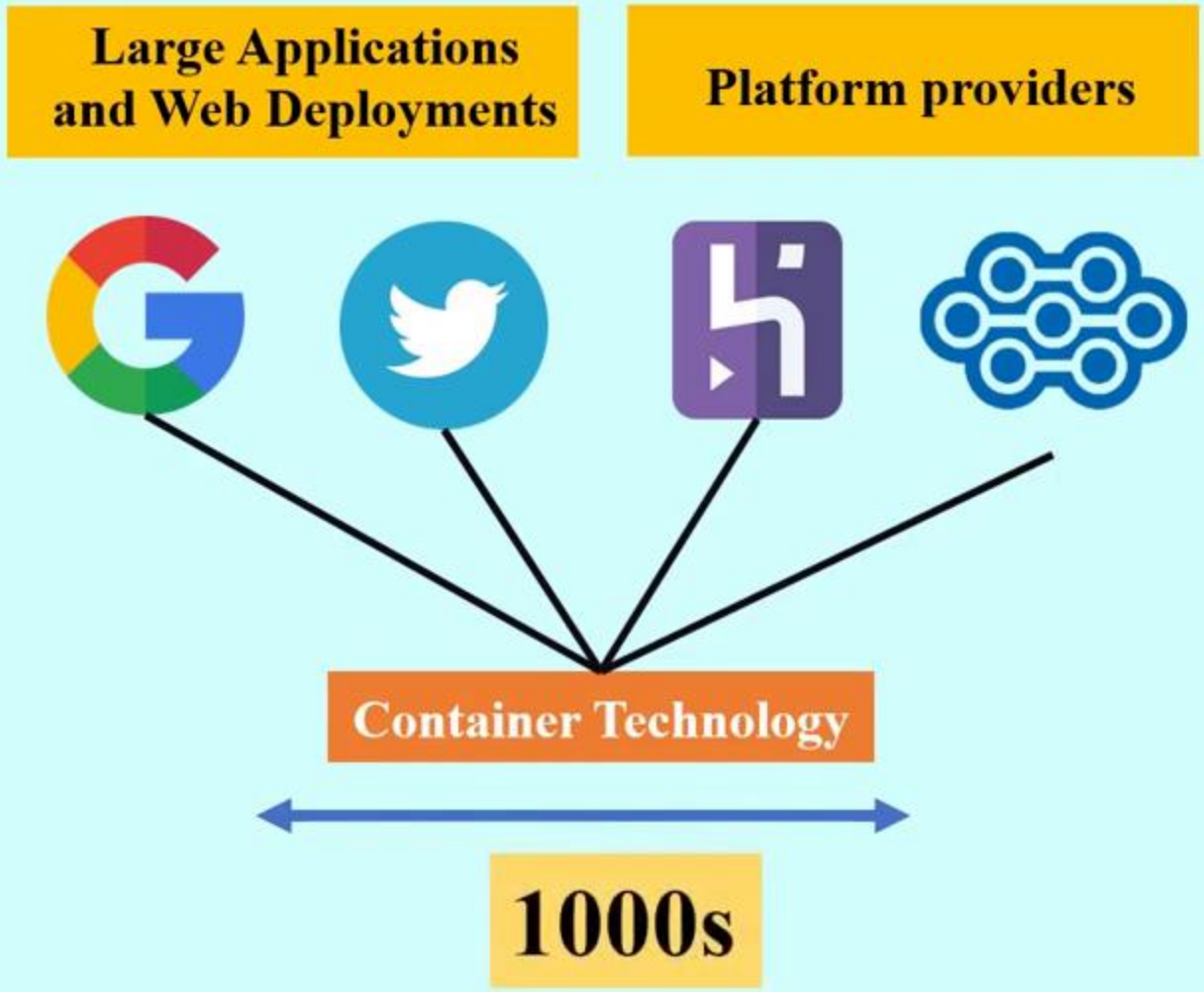
Large Applications and Web Deployments

Platform providers

How far do Docker containers scale?

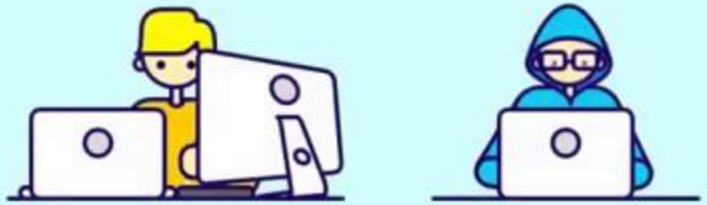


How far do Docker containers scale?



What is containerization

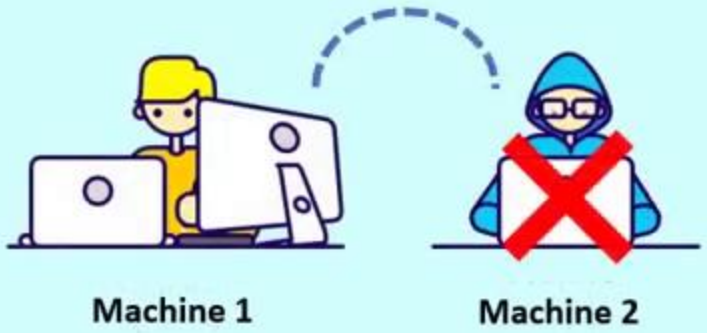
What is containerization



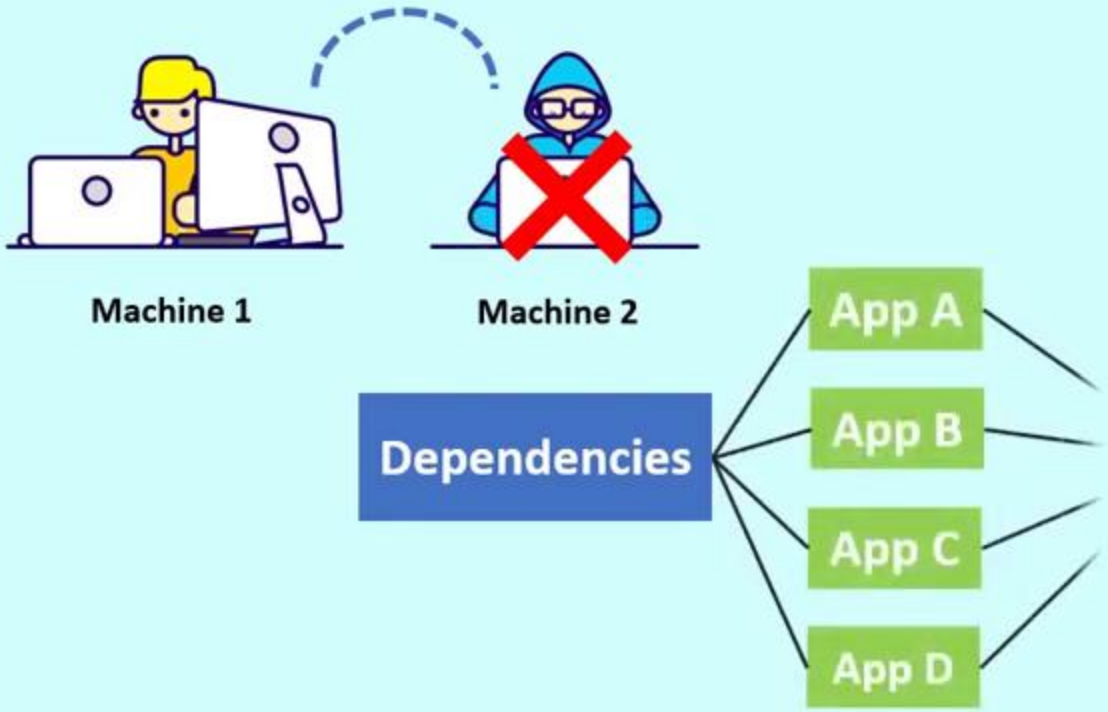
Machine 1

Machine 2

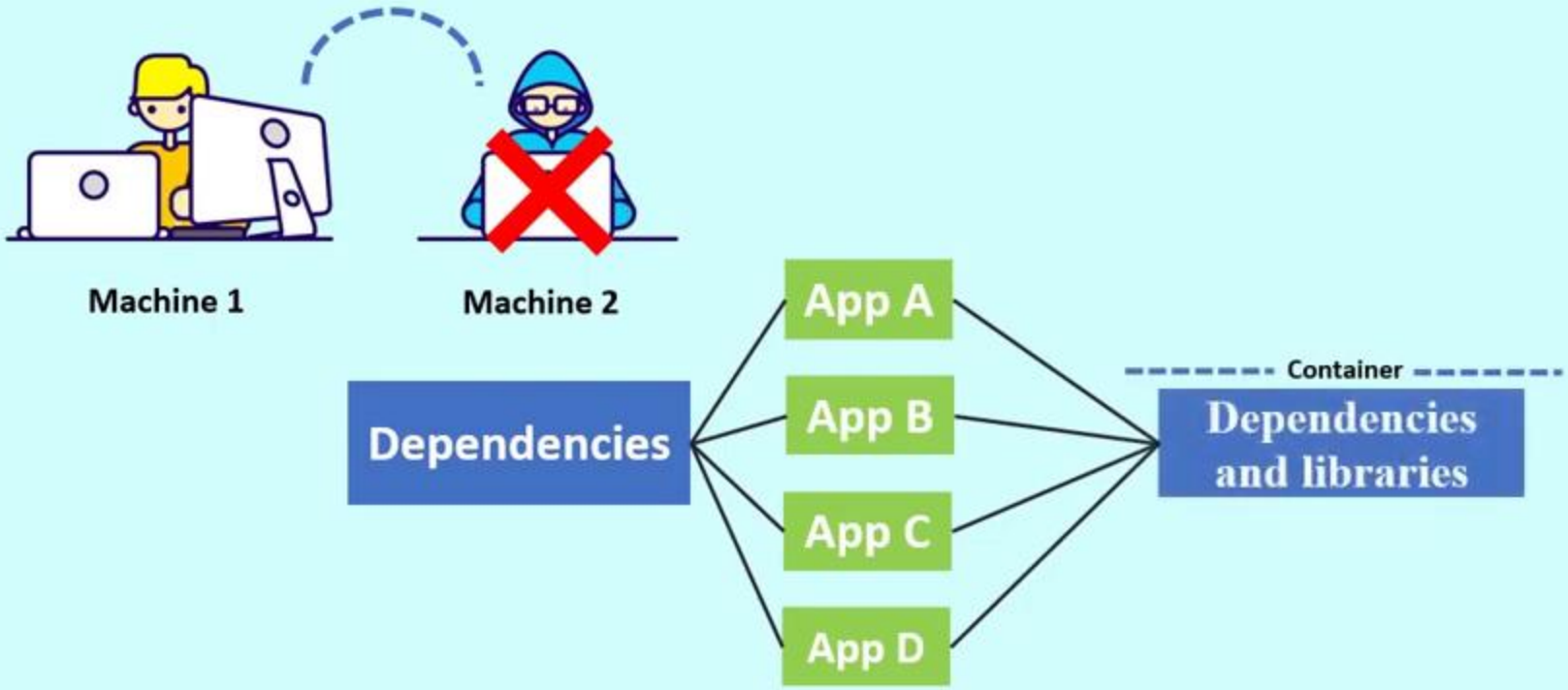
What is containerization



What is containerization

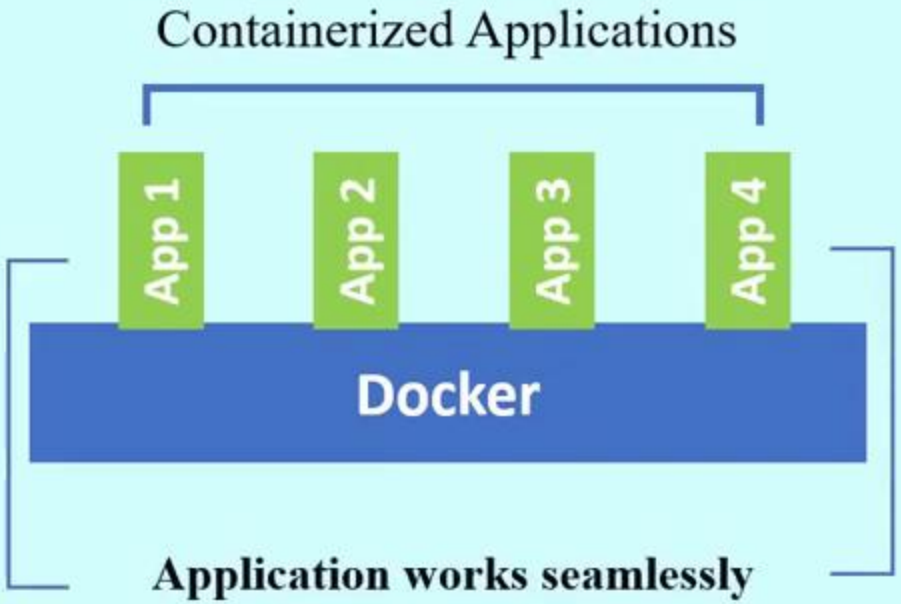


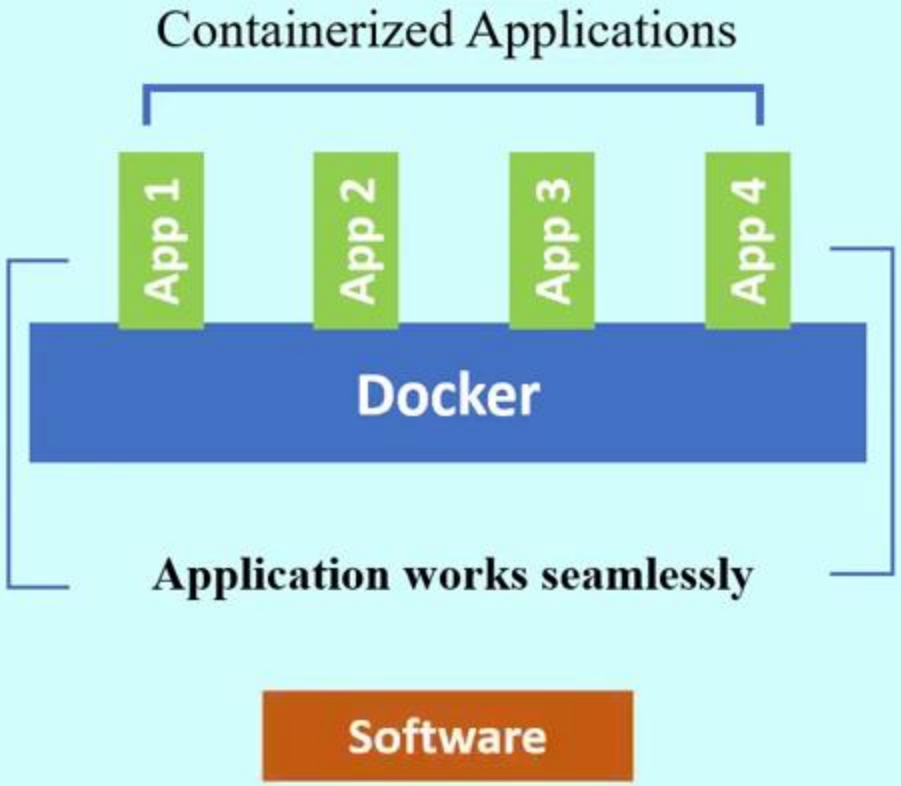
What is containerization

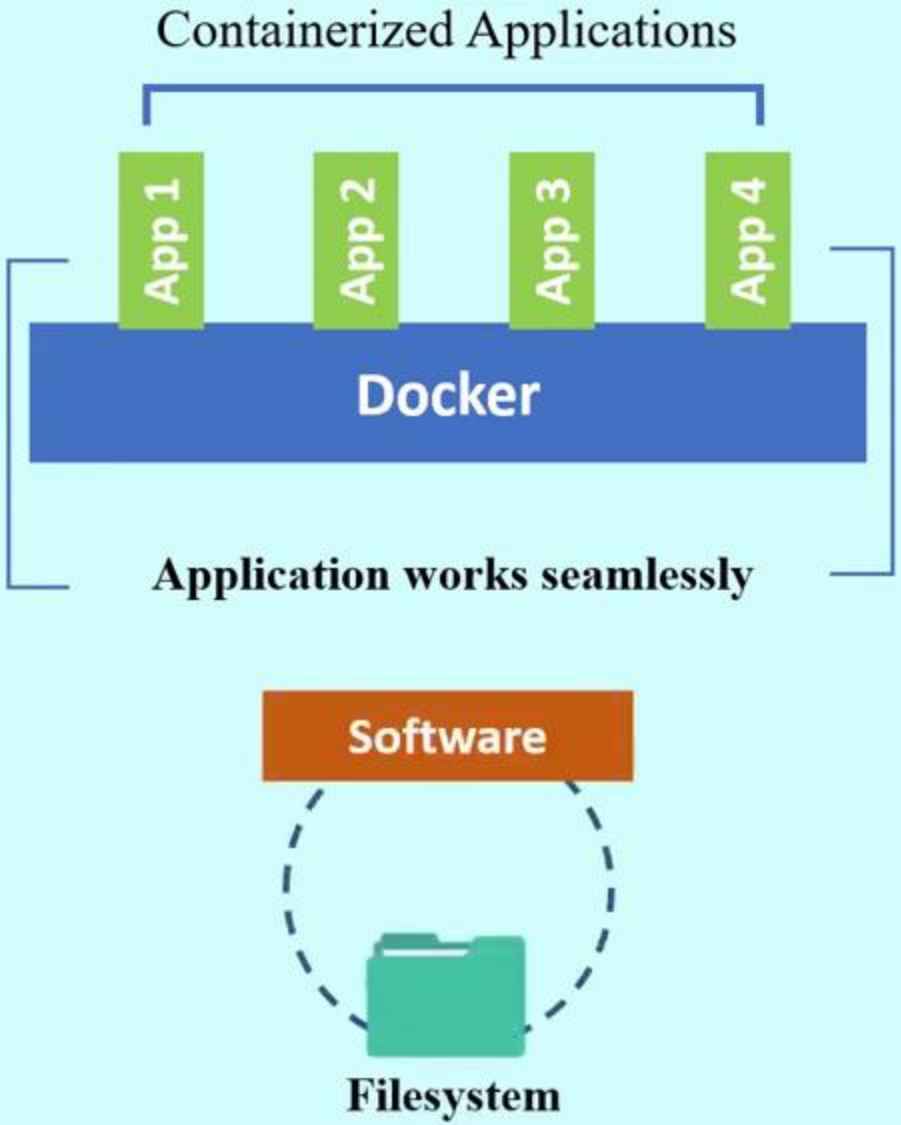


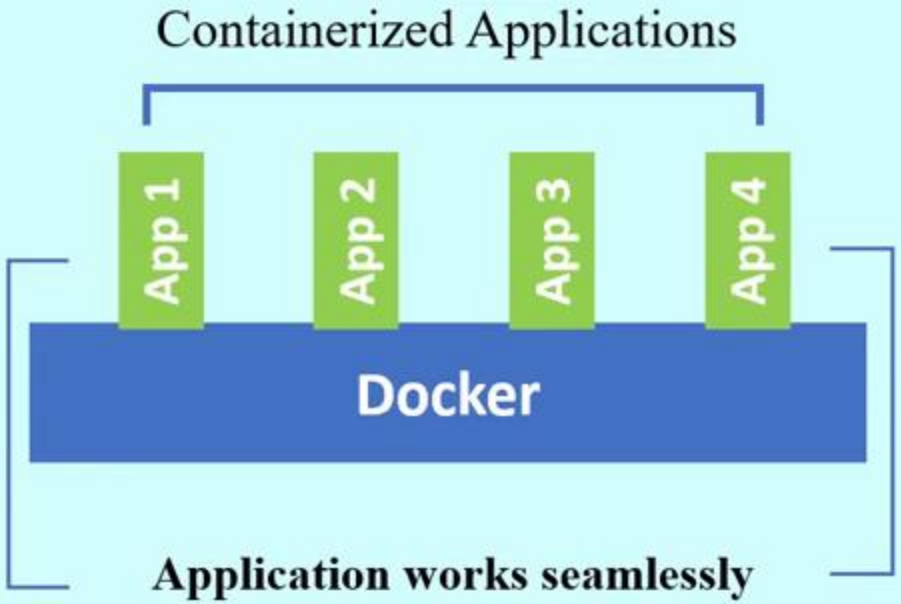
Docker

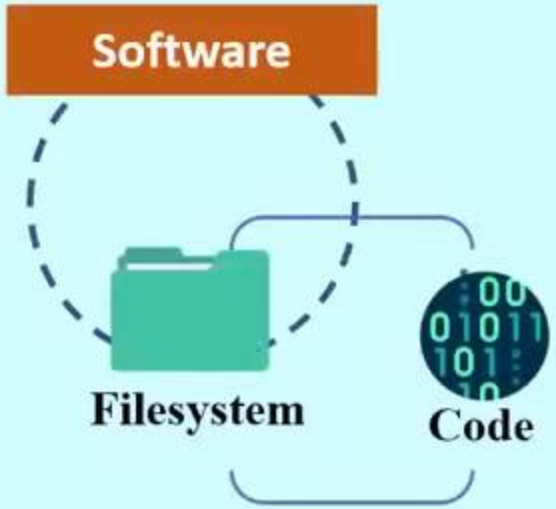

App 1
App 2
App 3
App 4
Docker

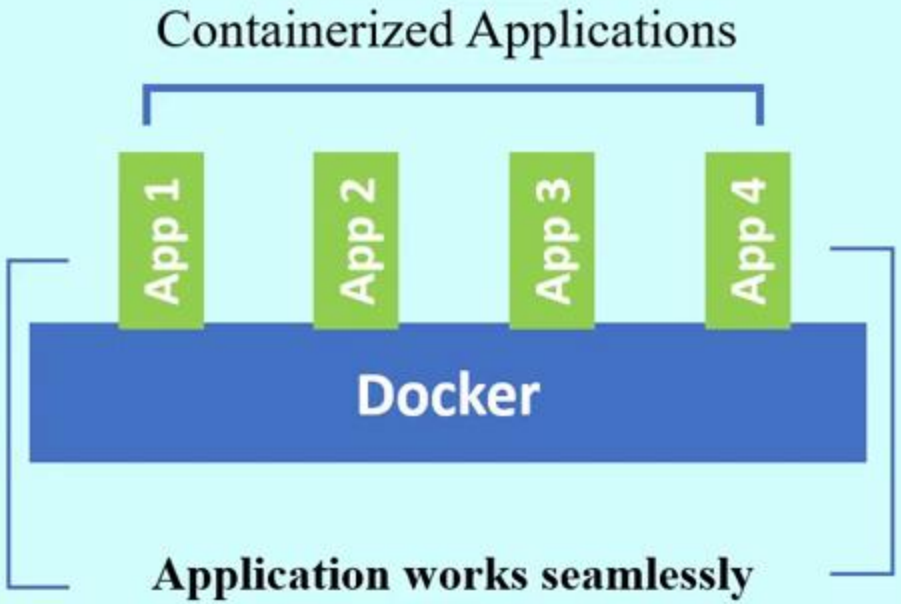

Containerized Applications
App 1
App 2
App 3
App 4
Docker
Application works seamlessly

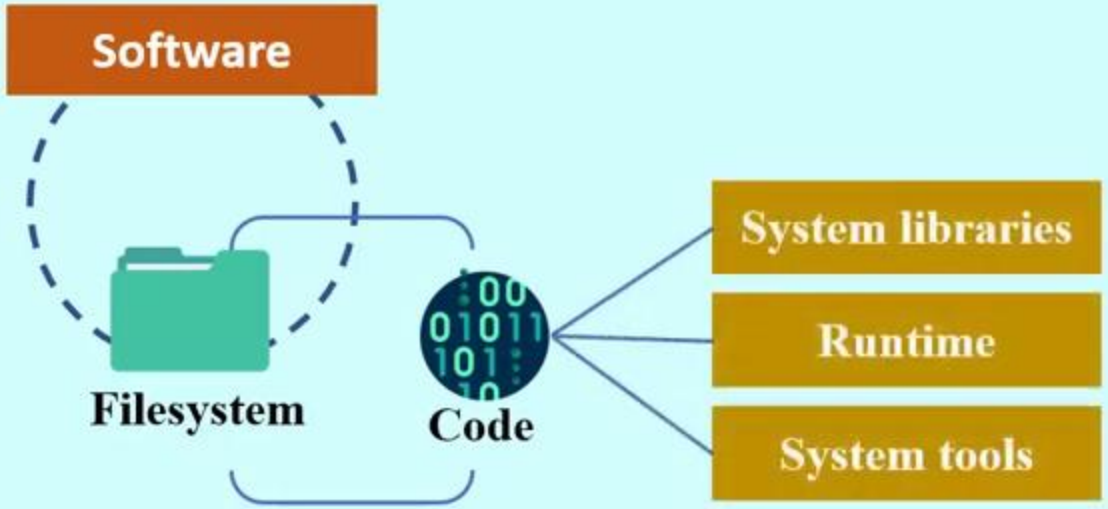

Containerized Applications
App 1
App 2
App 3
App 4
Docker
Application works seamlessly
Software

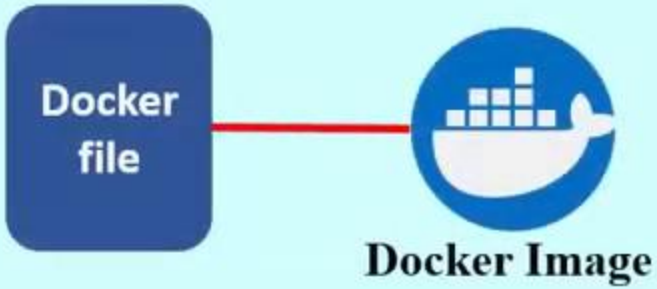

Containerized Applications
App 1
App 2
App 3
App 4
Docker
Application works seamlessly
Software
Filesystem


Containerized Applications
App 1
App 2
App 3
App 4
Docker
Application works seamlessly

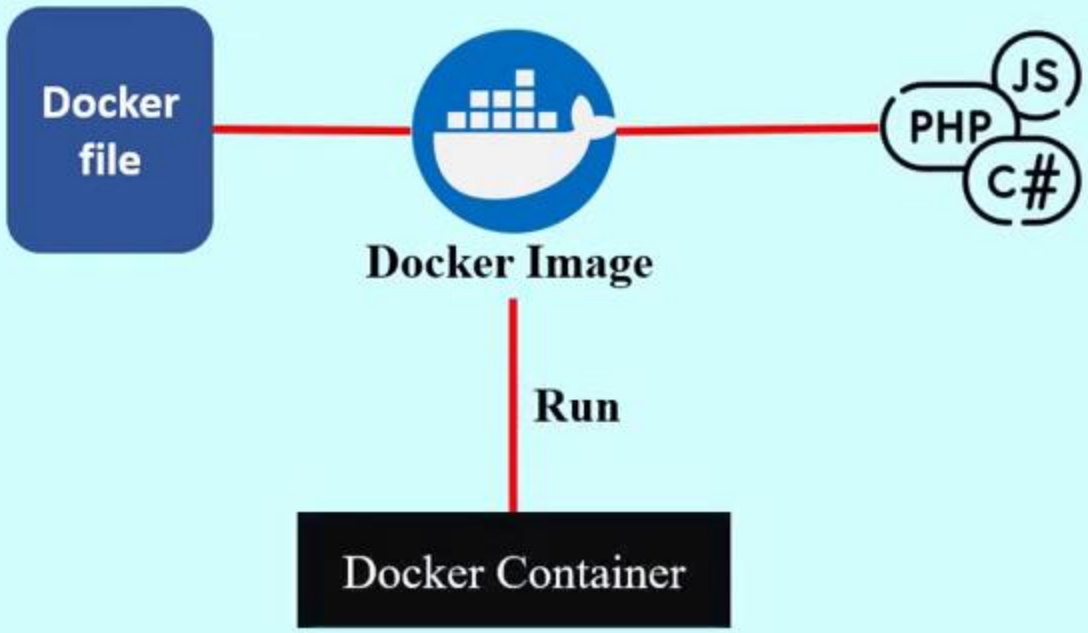

Software
Filesystem
Code

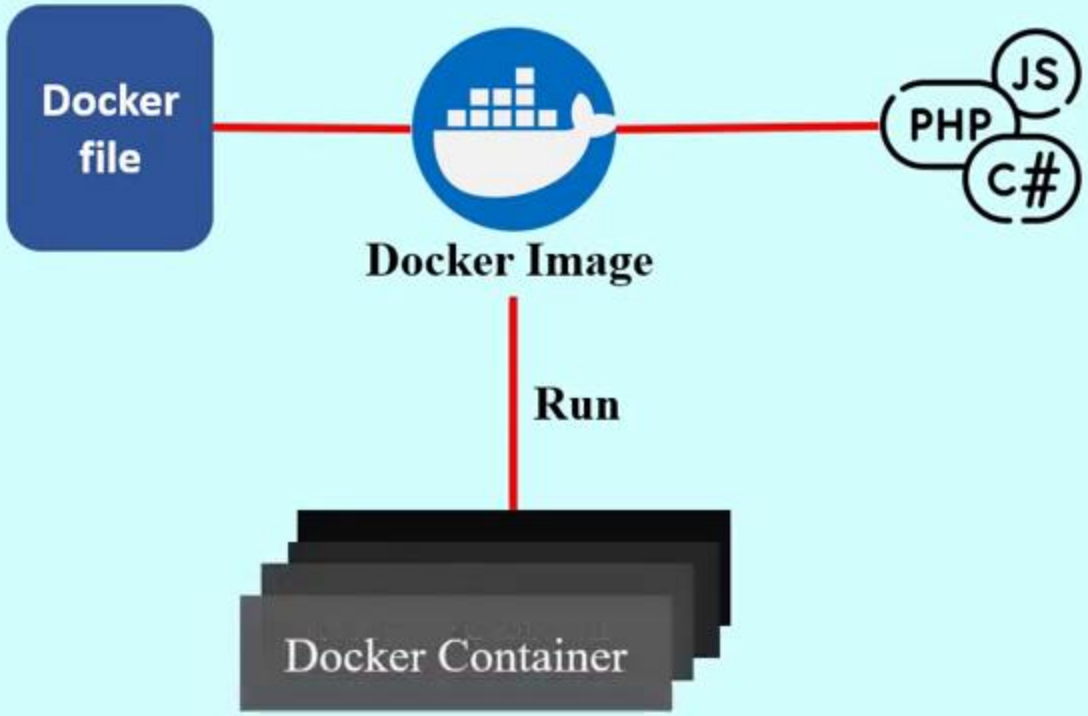

Containerized Applications
App 1
App 2
App 3
App 4
Docker
Application works seamlessly

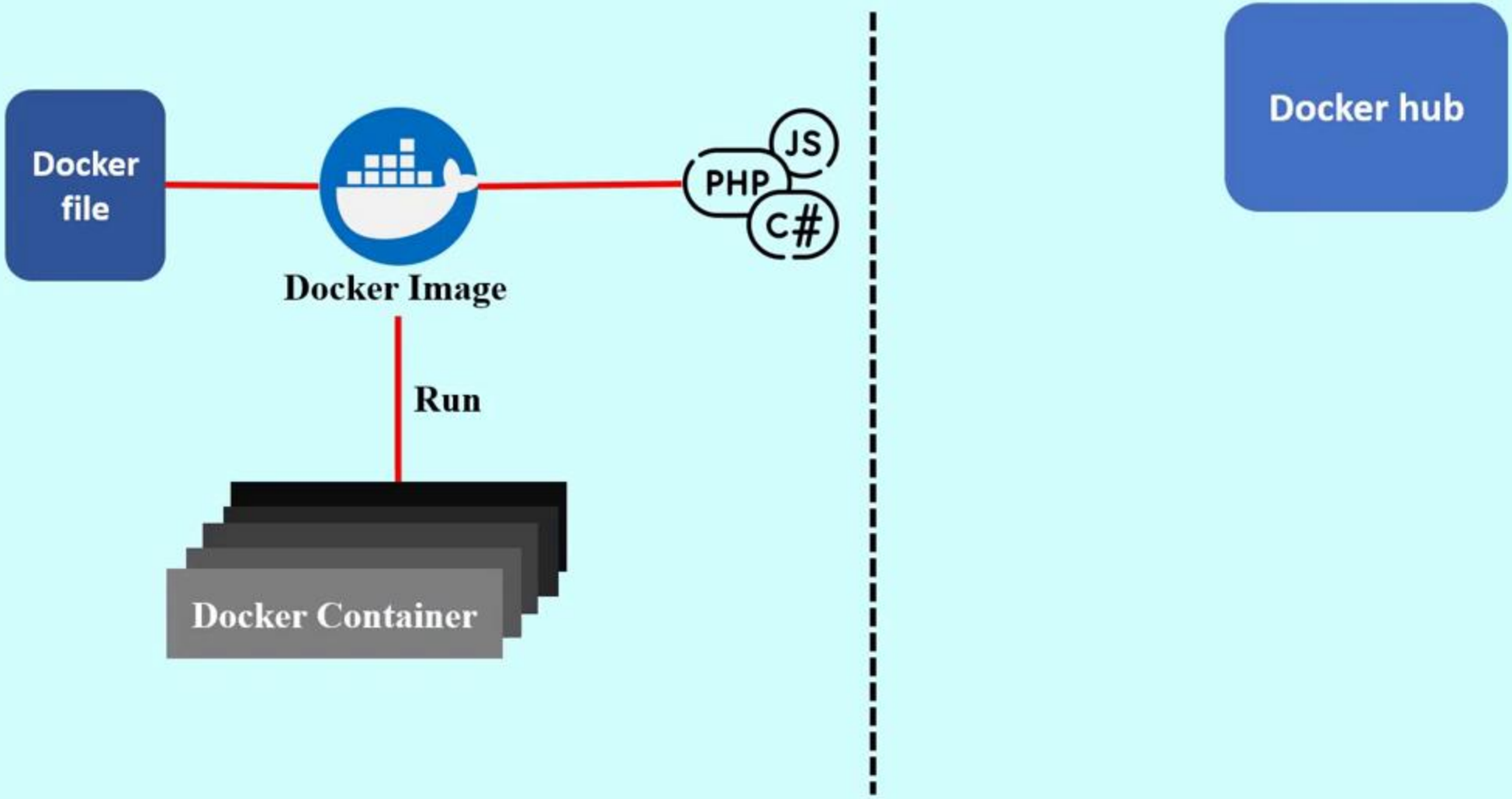

Software
Filesystem
Code
System libraries
Runtime
System tools

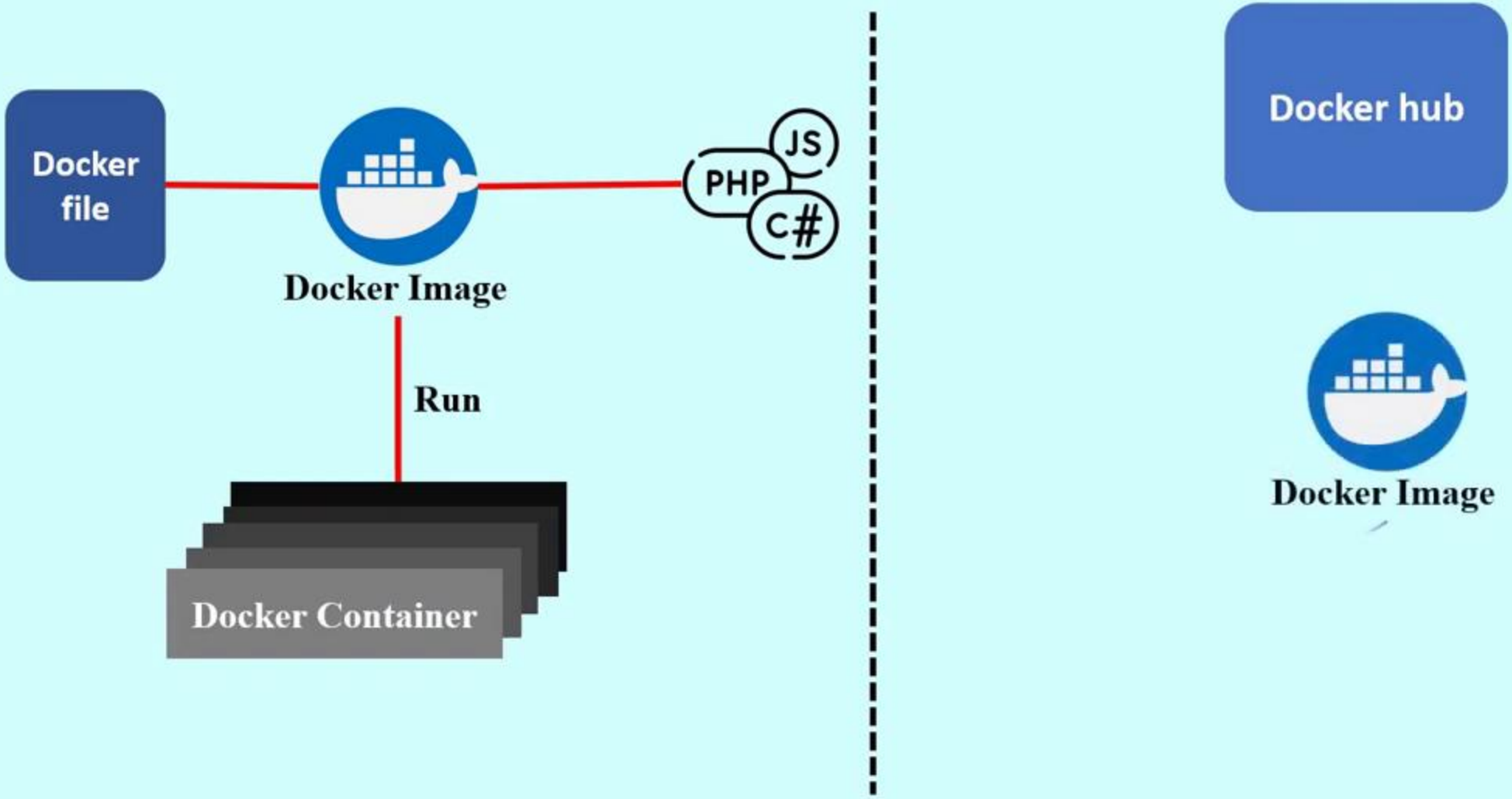

Docker
file
Docker Image

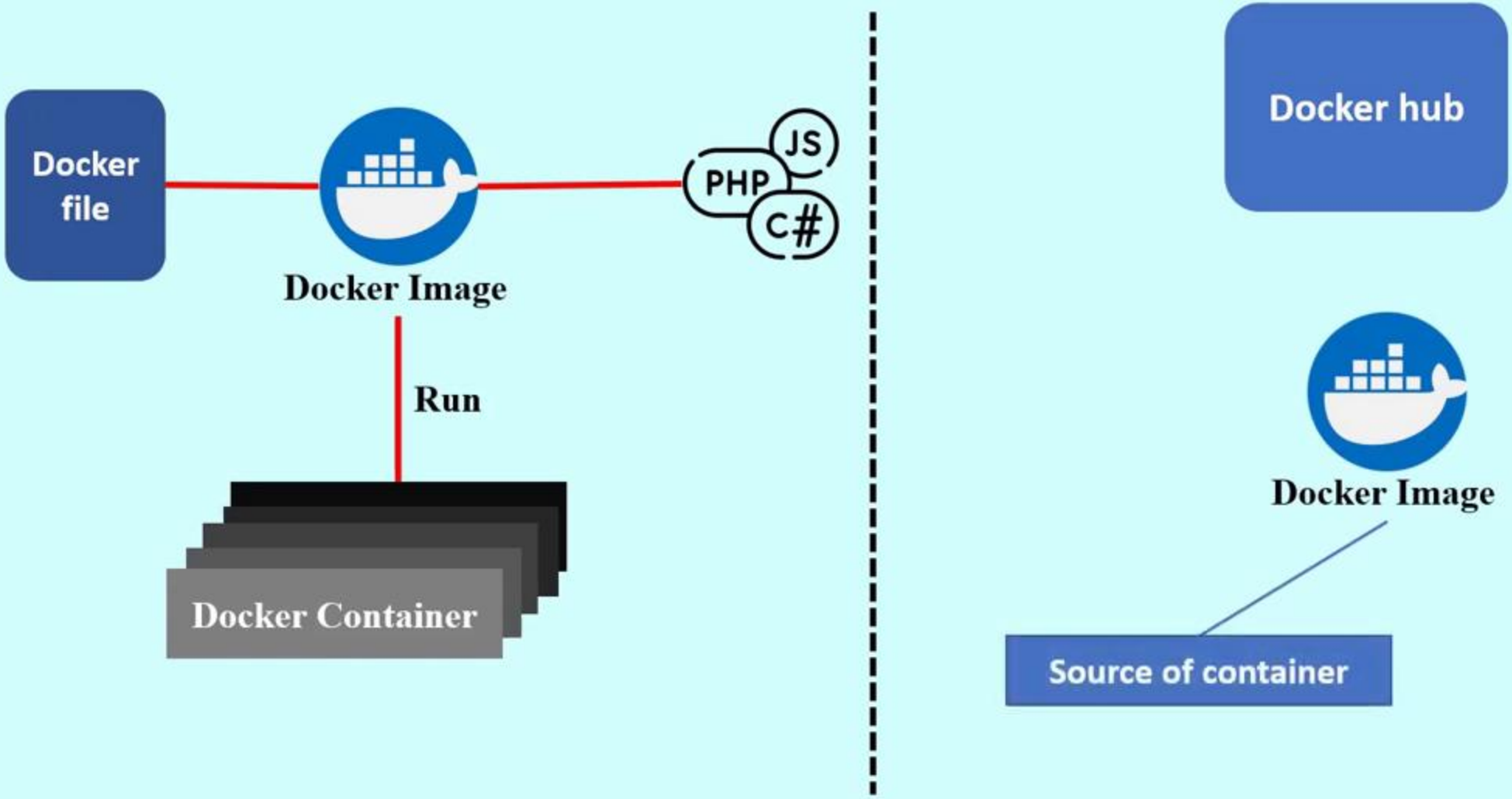

Docker
file
JS
PHP
C#
Docker Image

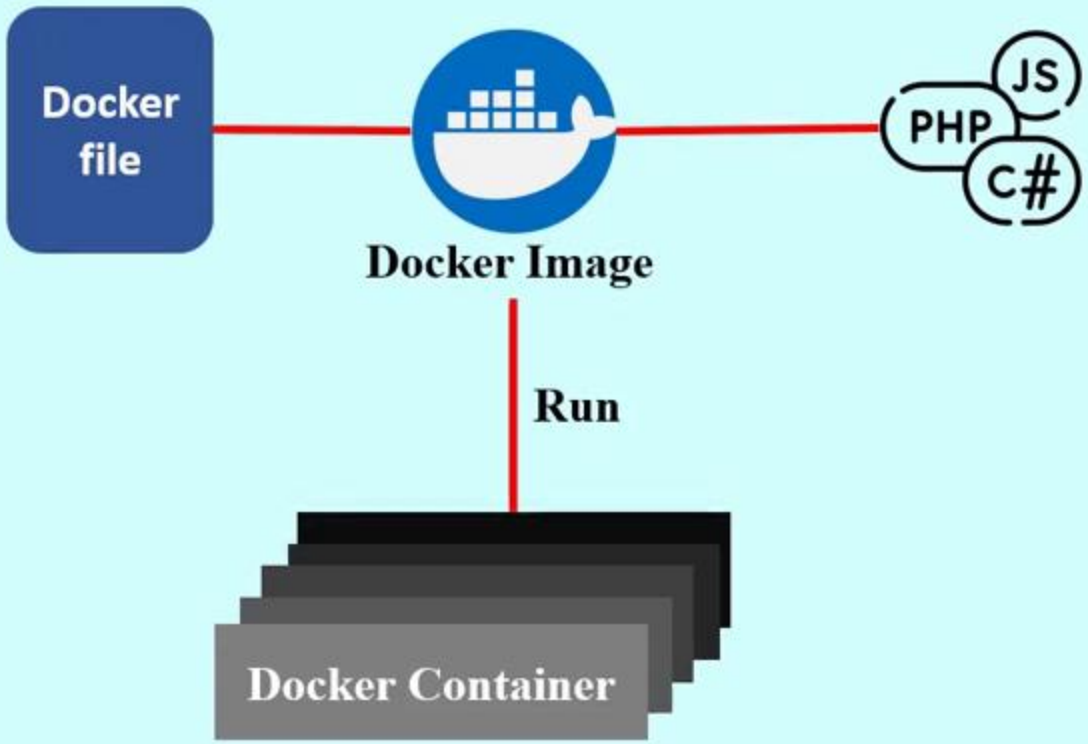

Docker file
JS
PHP
C#
Docker Image
Run
Docker Container

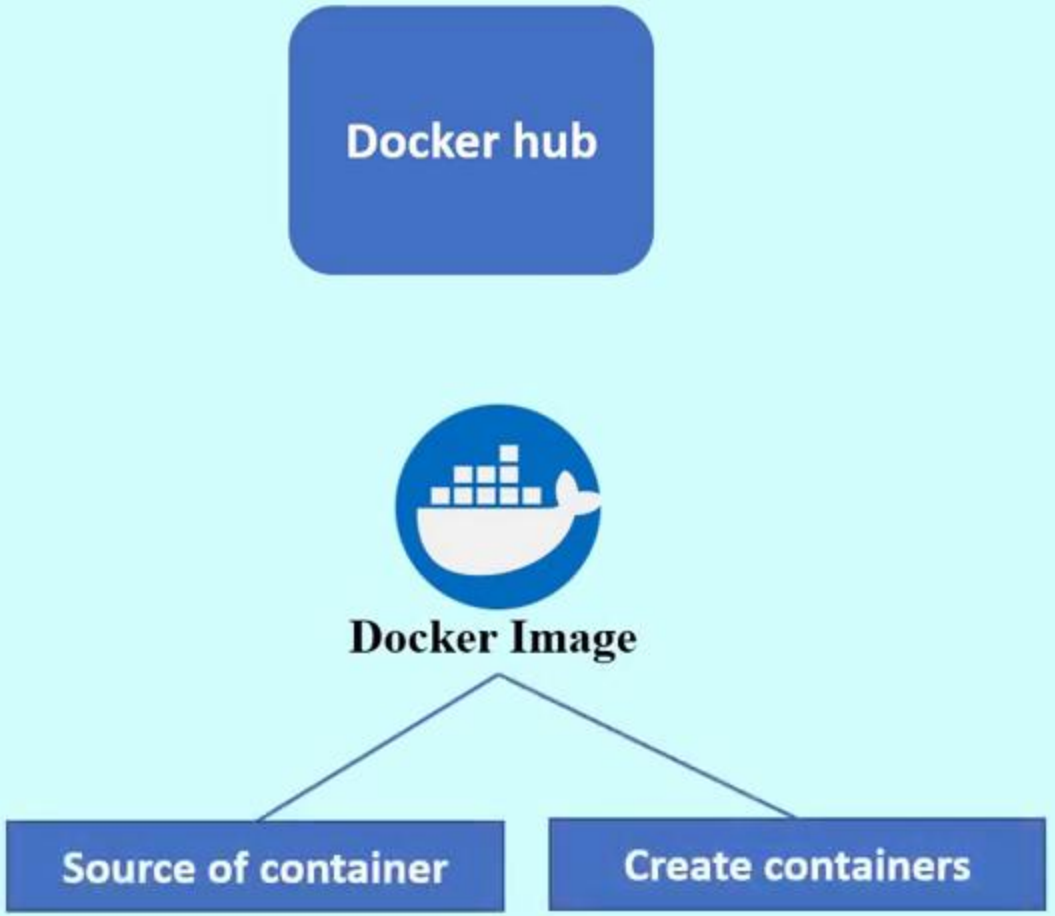

Docker file
JS
PHP
C#
Docker Image
Run
Docker Container


Docker file
JS
PHP
C#
Docker Image
Run
Docker Container
Docker hub


Docker file
JS
PHP
C#
Docker Image
Run
Docker Container
Docker hub
Docker Image


Docker file
JS
PHP
C#
Docker Image
Run
Docker Container
Docker hub
Docker Image
Source of container


Docker file
JS
PHP
C#
Docker Image
Run
Docker Container


Docker hub
Docker Image
Source of container
Create containers

When a user runs the docker image

When a user runs the docker image an instance of the container is created.

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo",
```

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo", "Hi World"]
```

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
```

```
MAINTAINER joe
```

```
RUN apt-get update
```

```
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

The basic syntax used to build an image using a Dockerfile is:

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
```

```
MAINTAINER joe
```

```
RUN apt-get update
```

```
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

The basic syntax used to build an image using a Dockerfile is:

To build a docker image, you would therefore use:

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

The basic syntax used to build an image using a Dockerfile is:

To build a docker image, you would therefore use:

```
docker build [location of your dockerfile]
```

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

The basic syntax used to build an image using a Dockerfile is:

To build a docker image, you would therefore use:

```
docker build [location of your dockerfile]
```

To run a Docker container based on this image. To do that we are going to use the docker run command.

When a user runs the docker image an instance of the container is created.

```
touch Dockerfile
```

Then, add the following content:

```
FROM ubuntu
MAINTAINER joe
RUN apt-get update
CMD ["echo", "Hi World"]
```

Build a Docker Image with Docker file:

The basic syntax used to build an image using a Dockerfile is:

To build a docker image, you would therefore use:

```
docker build [location of your dockerfile]
```

To run a Docker container based on this image. To do that we are going to use the docker run command.

#Docker run <image name>

To open an interactive terminal and connect to the container, run.

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

Let’s look at some more basic docker commands.

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

Let’s look at some more basic docker commands.

To list all the running containers.

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

Let's look at some more basic docker commands.

To list all the running containers.

docker ps

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

Let's look at some more basic docker commands.

To list all the running containers.

docker ps

To remove a container.

run the docker rm command

To open an interactive terminal and connect to the container, run.

#Docker run -it -d <image-name>

Let's look at some more basic docker commands.

To list all the running containers.

docker ps

To remove a container.

run the docker rm command

docker rm <container ID>

we should also find a way so that our terminal is not attached to the running container. This way, you can happily close your terminal and keep the container running.

```
$ docker run -d -P --
```

```
$ docker run -d -P --name static-site testsite/static-s
```

```
$ docker run -d -P --name static-site testsite/static-site
```

```
d14d1229
```

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f954    504af
```

```
$ docker run -d -P --name static-site testsite/static-site
```

```
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

Docker exec -it <container id> bash ----> it stands for interactive terminal

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

Docker exec -it <container id> bash ----> it stands for interactive terminal

#Start, Stop and Kill a Container Docker

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

Docker exec -it <container id> bash ----> it stands for interactive terminal

#Start, Stop and Kill a Container Docker start <container id>

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

Docker exec -it <container id> bash ----> it stands for interactive terminal

#Start, Stop and Kill a Container Docker start <container id> Docker stop <container id>

```
$ docker run -d -P --name static-site testsite/static-site
d14d12292d69556eabee4f95438e504afb7b02818
```

Docker container prune

#Suppose you have 3 containers running and out of these you wish to access one of them

How do you access a running container?

Docker exec -it <container id> bash ----> it stands for interactive terminal

#Start, Stop and Kill a Container Docker start <container id> Docker stop <container id>

Docker kill <Container id>

Note

Docker-toolbox

Note

Docker-toolbox

Docker-machine Ip

Note

Docker-toolbox

Docker-machine Ip

to get the IP

Note

Docker-toolbox

Docker-machine Ip

to get the IP

```
$ docker run -
```

Note

Docker-toolbox

Docker-machine Ip

to get the IP

```
$ docker run -p 8888:80 te
```

Note

Docker-toolbox

Docker-machine Ip

to get the IP

```
$ docker run -p 8888:80 testsite/test-site
```

Docker hub