Docker Image

Vs

Docker Container

DOCKER

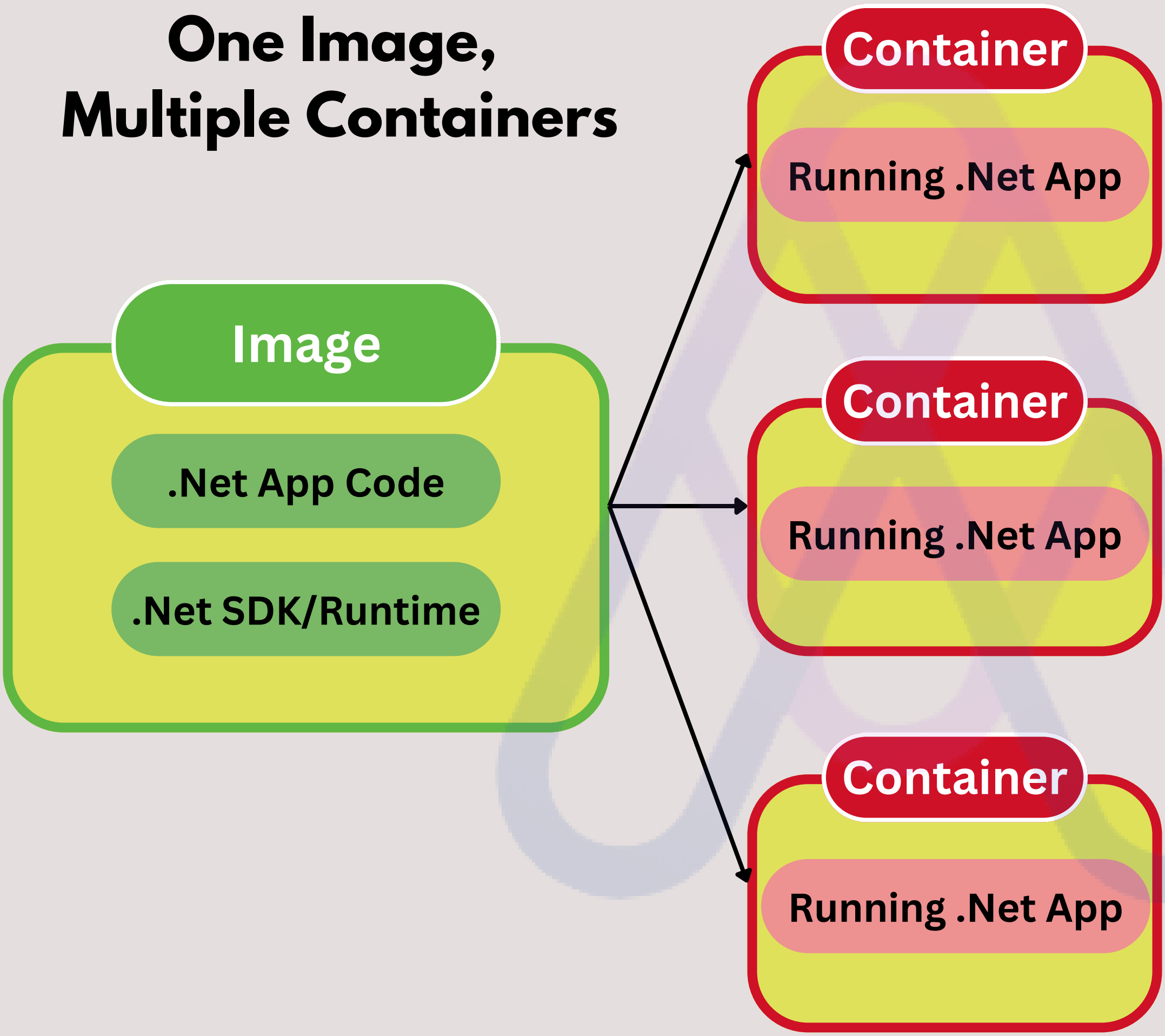
Images

- Templates for containers
- Contains code, dependencies, runtime
- Layer-based, read-only.

Containers

- The running instance of Image.
- Mutiple container can be created based on one image.
- Read and Write access. Data is lost on shutdown.

One Image, Multiple Containers



How to get an Image?

Use an existing, pre-built image

From

- Docker Hub
- Microsoft Container Registry
- Other private registry

Create your custom image

- Write your own Dockerfile

1 Downloading Images:

The docker pull command is used to download an image from the repository so that it is available locally.

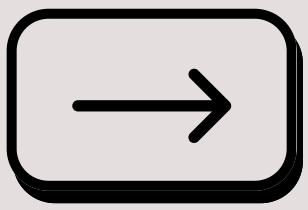
docker pull imagename:tag

ex: docker pull node

if you don’t specify tag, it will pull latest image.

docker run imagename:tag

docker run download an image from repository if it is not available locally and create and start the container.



2 Create custom Image:

docker build -t name:tag .

Dockerfile

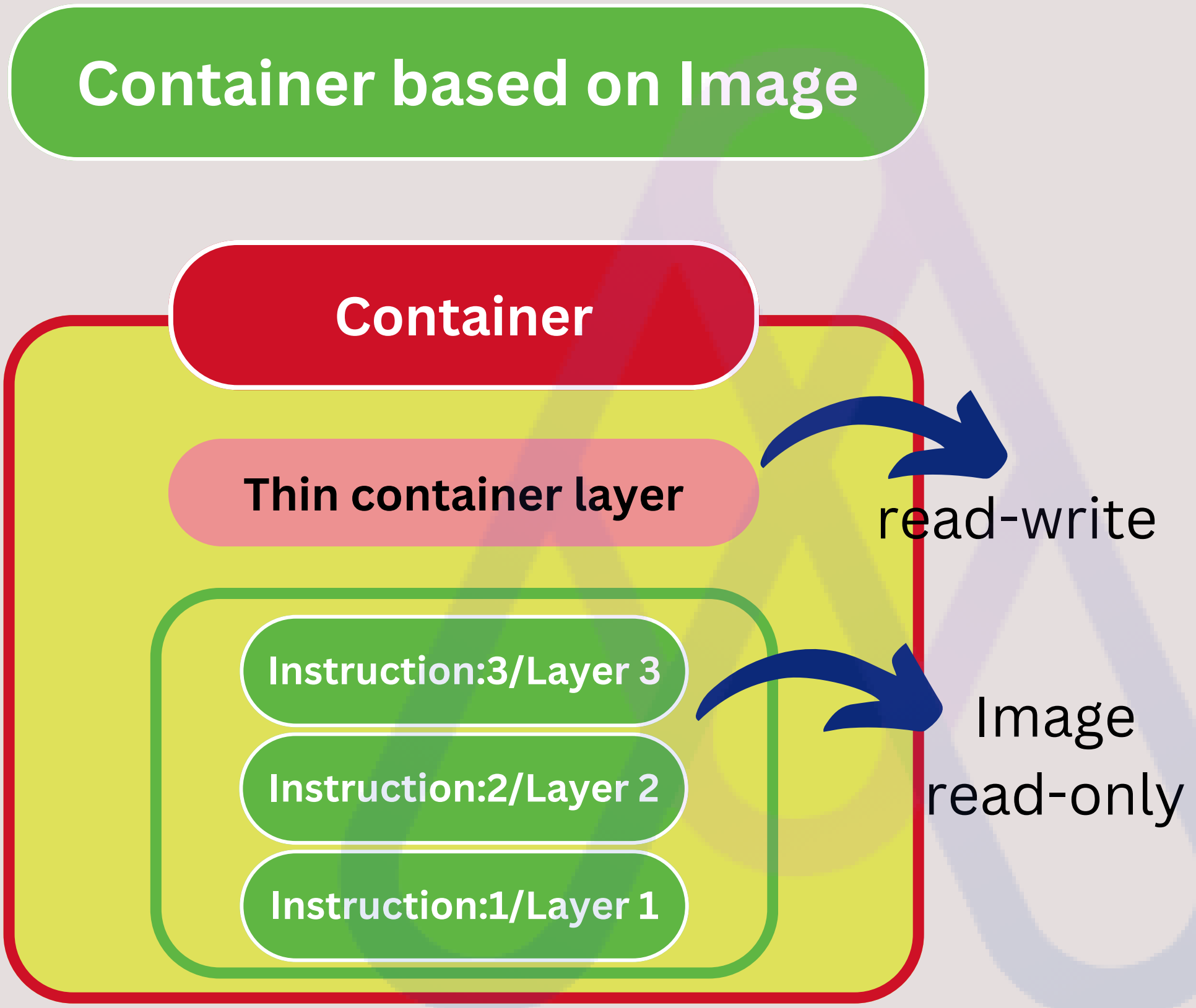
ENTRYPOINT [”dotnet”,”example.dll”]

COPY . .

WORKDIR /app

Base image from docker hub/ microsoft registry

Container based on Image



docker run imagename:tag

Key Commands for image:

Build an image: docker build -t name:tag .

Inspect an image: docker inspect name/id

Remove an image: docker rmi name/id

Remove all images: docker image prune

Pull an image: docker pull name:tag

NOTE: docker image prune will remove only untagged images. If you want to remove tagged images as well add -a with command.

Key Commands for container:

- **docker create imagename:** Create new container.
- **docker run imagename:** Create and start new container based on image.
- **docker stop containername:** Stop a running container.
- **docker start containername:** Start a stopped container.
- **docker rm containername:** Remove specified container.
- **docker container prune:** Remove all containers.