CCGC 5001 - Virtualization

Module 4A: Creating and Managing Container Images

HUMBER

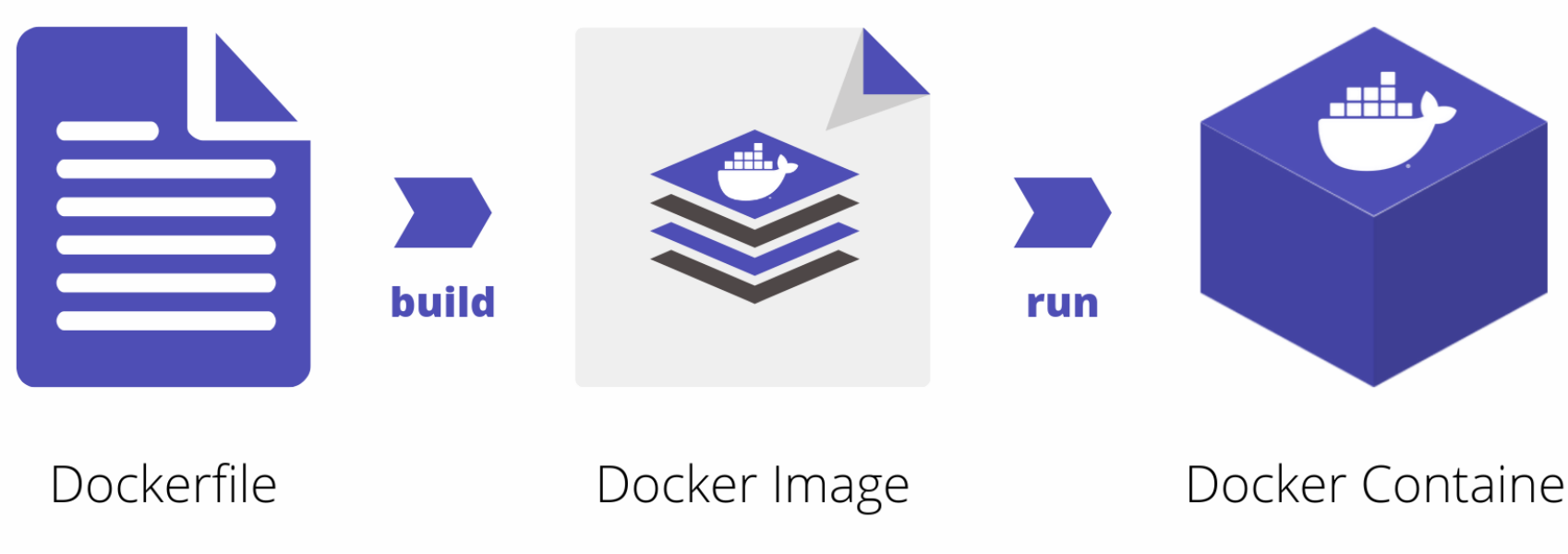
Module objectives

At the end of this module, you should be able to:

- Name three of the most important characteristics of a container image
- Create a custom image by interactively changing the container layer and committing it

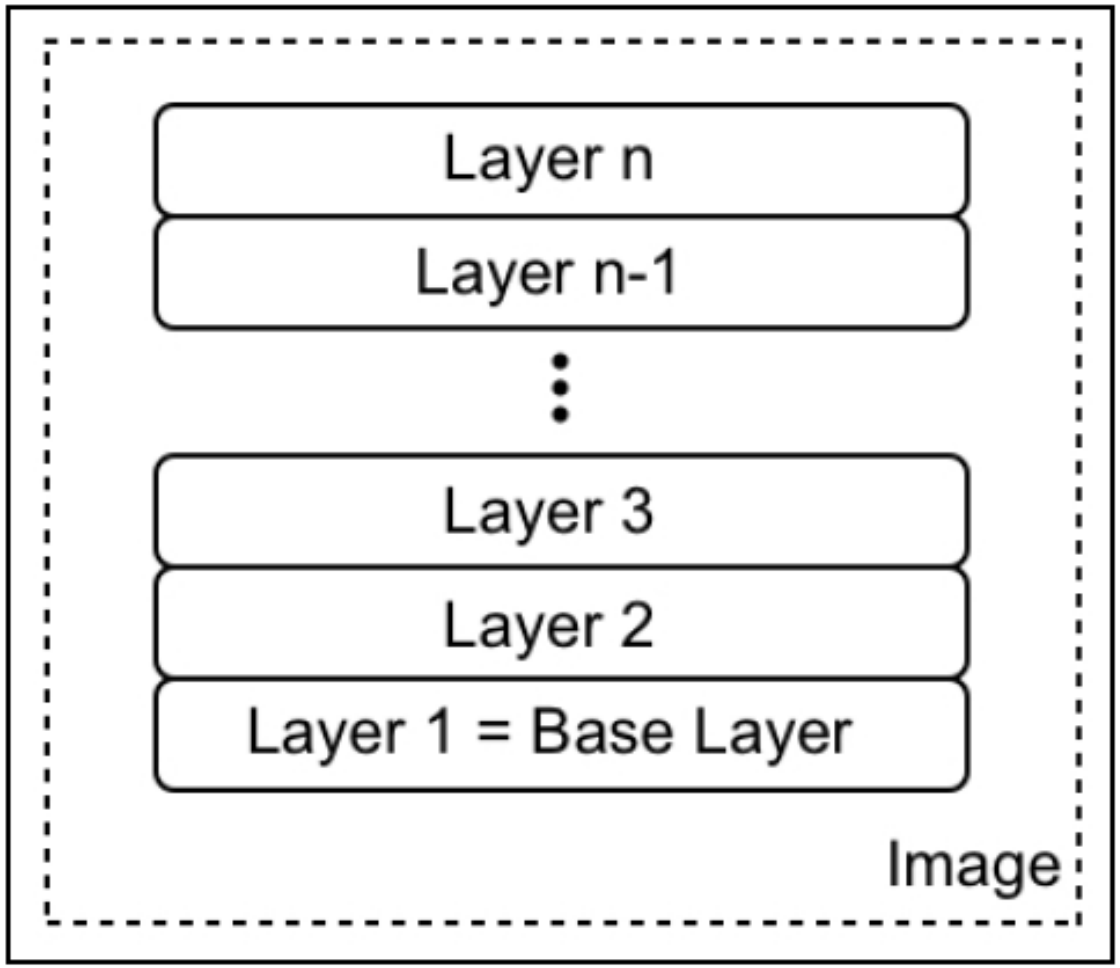
Container images

An image is a big tarball containing a layered filesystem. An image defines both what you want your packaged application and its dependencies to look like and what processes to run when its launched.



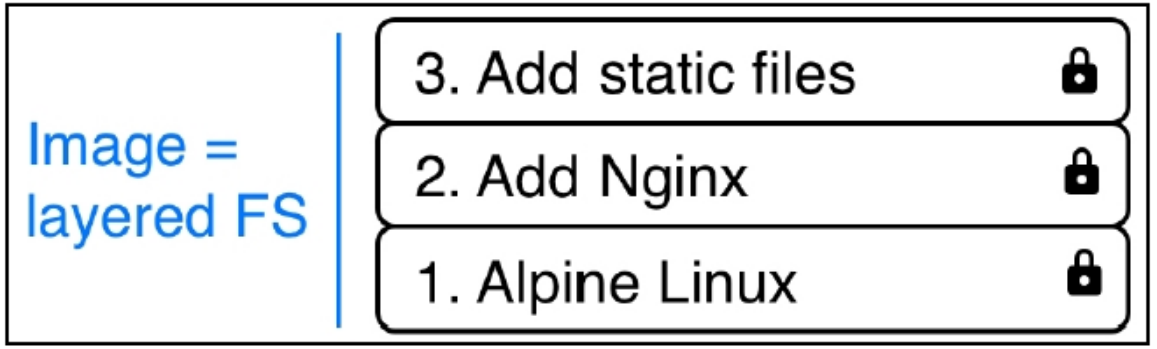
Layered filesystem

- An image is a template with many layers
- First layer is called the base layer
- Each layer containers files and folders
- Each layer only contains the changes to the filesystem wrt underlying layers
- Union filesystem is used to create virtual filesystem out of the set of layers
- Layers of image are all immutable



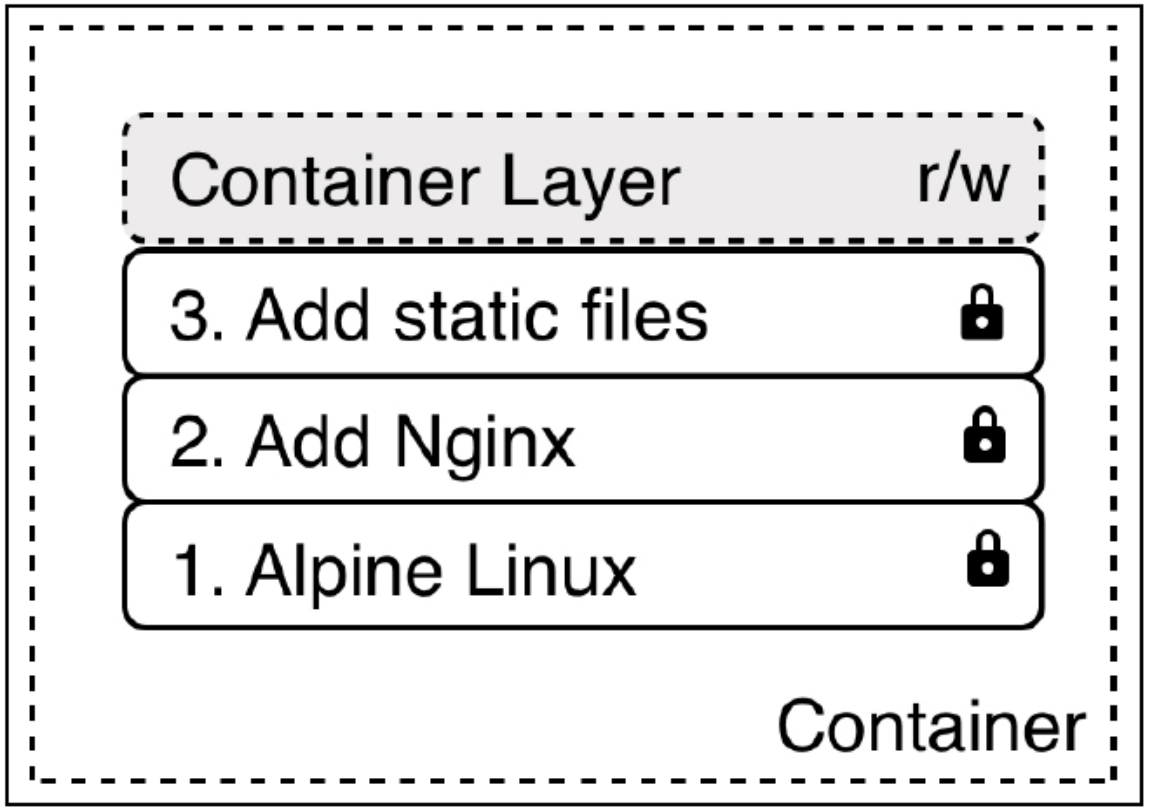
An example …

- Base layer is Alpine Linux
- Nginx layer is on top of base layer
- Third layer contains web application files, such as HTML, CSS, and JavaScript.
- Base image is typically one of the official images found on Docker Hub
- Content of each layer is mapped on special folder on the host (usually subfolder of /var/lib/docker)

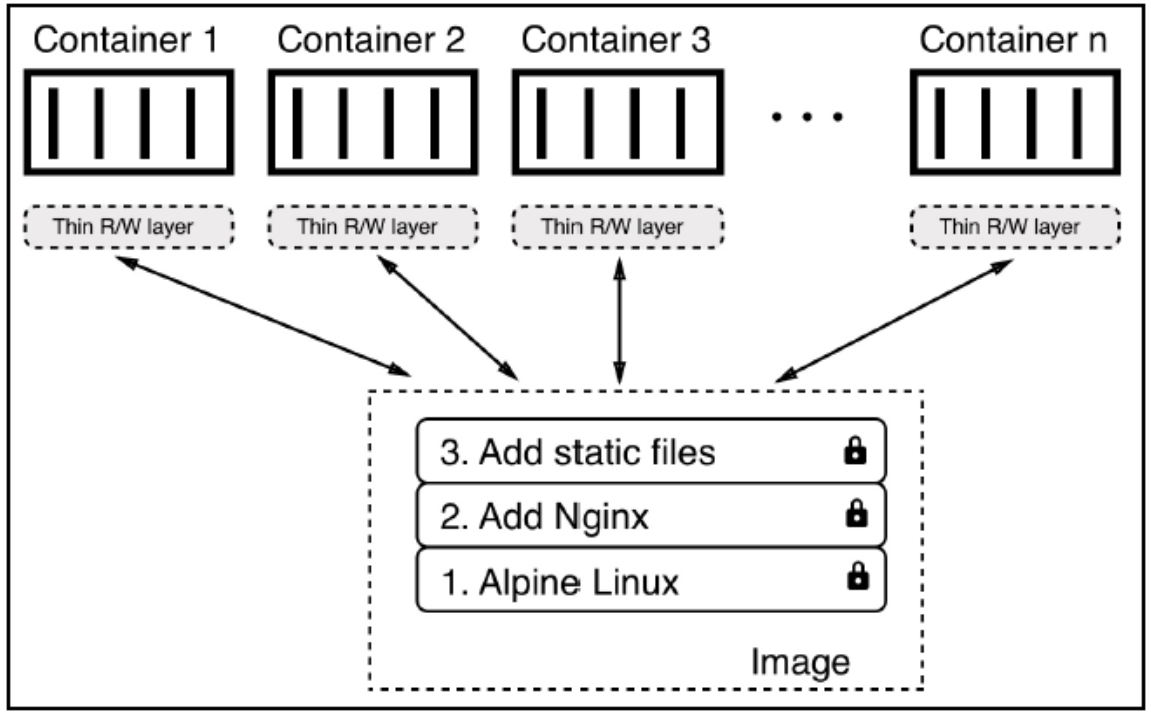


Writeable container layer

- When the Docker Engine creates a container from an image, it adds a writeable container layer on top immutable layers
- Container layer is marked as read/write



Writeable container layer



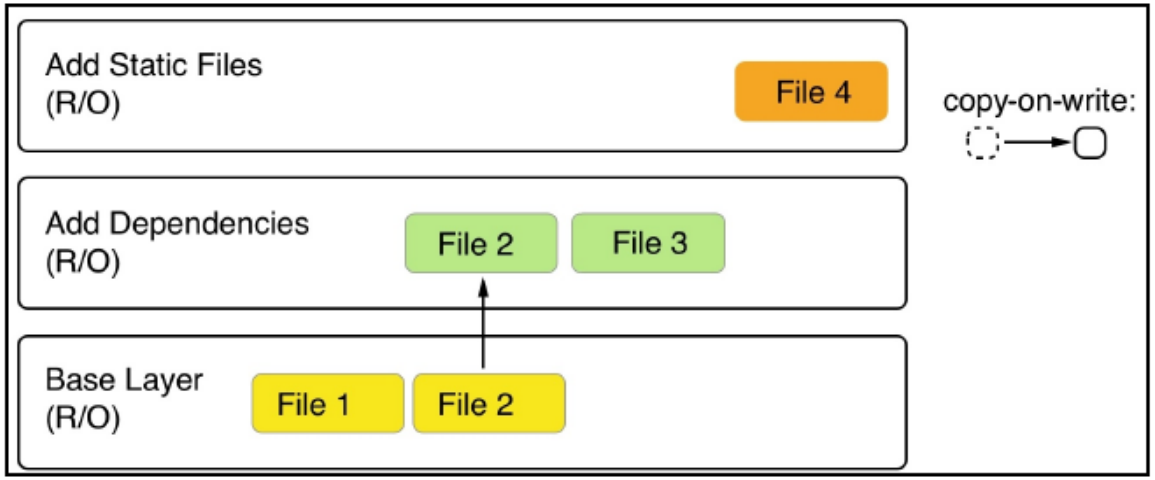
An advantage of the immutability of image layers is that they can be shared among many containers.

Copy-on-write

Copy-on-write is a strategy for sharing and copying files for maximum efficiency.

If a layer uses a file or folder that is available in one of the low-lying layers, then it just uses it.

If a layer wants to modify, say, a file from a low-lying layer, then it first copies this file up to the target layer and then modifies it.



Creating images

- Interactively building a container that contains all the additions and changes
- Importing it into the system from a tarball
- Dockerfile

Interactive image creation

We can create a custom image by interactively building a container.

```
$ docker container run -it --name sample alpine:3.10 /bin/sh
```

Start with a base image and use it as a template.

Interactive image creation

By default, our alpine container does not have ping tool installed. Let's create a new image that has the ping installed.

```
/ # apk update && apk add iputils
```

This uses the apk Alpine package manager to install the iputils library, of which ping is a part.

Once we have finished our customization, we can quit the container by typing exit at the prompt.

Interactive image creation

If we want to see what has changed in our container in relation to the base image, we can use the docker container diff command, as follows:

```
$ docker container diff sample
```

Interactive image creation

We can now use the docker container commit command to persist our modifications and create a new image from them, like this:

```
$ docker container commit sample my-alpine
```

We can verify this by listing all images on our system, as follows:

```
$ docker image ls
```

Interactive image creation

If we want to see how our custom image has been built, we can use the history command as follows:

```
$ docker image history my-alpine
```

Saving and loading images

The second way to create a new container image is by importing or loading it from a file. To demonstrate this, we can use the docker image save command to export an existing image to a tarball:

```
$ docker image save -o ./backup/my-alpine.tar my-alpine
```

Saving and loading images

If we have an existing tarball and want to import it as an image into our system, we can use the docker image load command, as follows:

```
$ docker image load -i ./backup/my-alpine.tar
```

Module summary

In summary, in this module, you learned:

- What container images are
- How we can build and ship them
- Three different ways that an image can be created

Thank you