

Chapter 10 - What Next

Container Patterns

Developer Machine Homogenisation

You can effectively have your only dependency on a development machine as having Docker installed. You can have all the runtimes operating in containers, meaning that you're onboarding becomes:

```
Install Docker
Git clone the repo
Run a script which spins up the containers, and runs all the tests
```

Sidecar Containers

Let's say you're developing against a Postgres database, now with Docker you can stand up a local Postgres instance in seconds allowing on developer machine integration testing.

Mutli-Stage Docker Builds

A fairly recent addition to the Docker toolkit is the concept of multi-stage Docker builds. As a rule, you want to optimise to the smallest image size possible. It makes it faster to build, to run and to share.

Say you have a Java application, to run the build you will require additional dependencies and you will need the JDK installed. However if you're building an image to be run, then you would only want the JRE installed, with minimal dependencies.

By specifying a second **FROM statement** you can copy the build artifacts from the layers earlier in the Dockerfile.

Challenges

1. More Use Cases

Back in Chapter 1, you put down what use cases you could see for containers at work. Are there any extra now that spring to mind?

2. Layer Invalidation

Go back to the chapter on Dockerfiles, can you figure out what would cause the layer per statement to be invalidated? E.g. With the **RUN** statement, how does the engine know whether to run that again, or use the layer created previously?

3. Environment Variables in **Docker Build**

Environment variables are not accessible as part of the **RUN** statements in a Dockerfile, they only take effect once the image is run. Can you find the statement you would need to have access to configurable variables during build. E.g. if you have credentials which need injecting to download packages.

4. Docker Compose

Go and research Docker Compose, and what the use cases are. See if you can set up a basic web-server and database application.