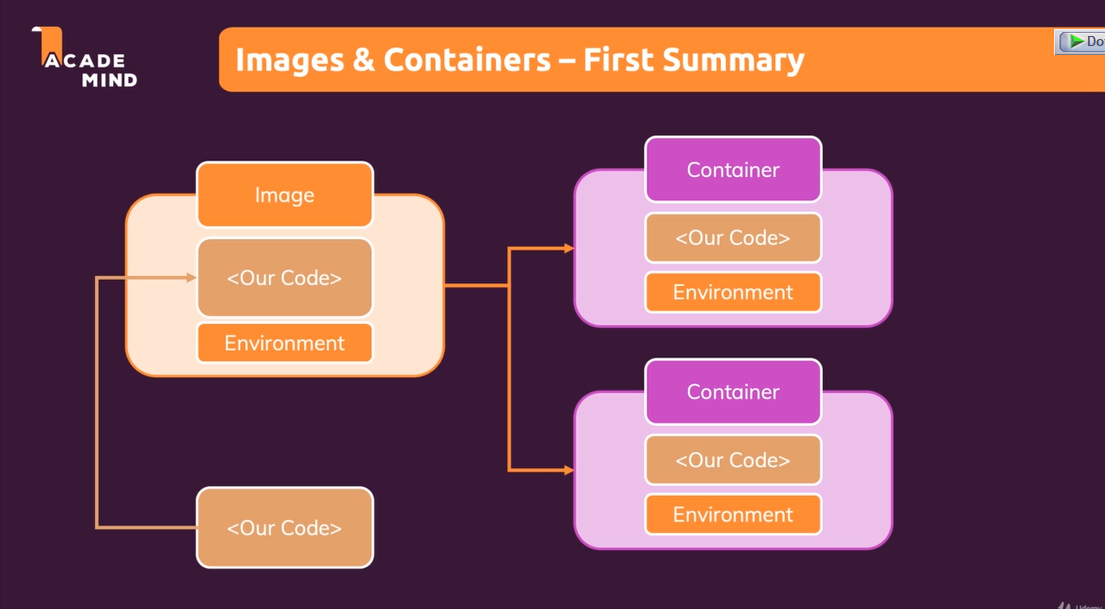
After generate everything Dockerfile and server.js etc.

Docker run -p 3000:8000 <build\_id>

Exp: docker run -p 3000:8000 8d060ec158daf5eb798364f2b0034779be90b3a25bad02c748365fd290b4da4b

Video 26: A First summary



When our code copy to Image, it will not be copied to Container, Container use the code inside of Image instead copy the code to their Container (will not be duplicated).

Quiz:

1. What are “Images” (when working with docker)?

Images are “blueprints” for containers which then are running instances with read and write access.

1. Why do we have “Images” and “Containers”? Why not just “Containers”?

This concept allows multiple containers to be based on the same image without interfering with each other.

(not “share the same app data and state”).

1. What does “isolation” mean in the context of containers?

Containers are separated from each other and have no shared data or state by default.

1. What’s a “Container”?

An isolated unit of software which is based on an image. A running instance of that image.

(not “A running instance of that app”)

1. What area “Layers” in the context of images?

Every instruction in an image creates a cacheable layer – layers help with image re-building and sharing.

(not “Multiple containers can be based on the same image, that’s what “Layers” means”. Nor, “An image may contain multiple applications -every application is a layer then.”)

1. What does this command do? “docker build .”

It builds an image.

1. What does this command do? “docker run node”

It creates and runs an image based on the “node” image.

Managing Images and Containers.

