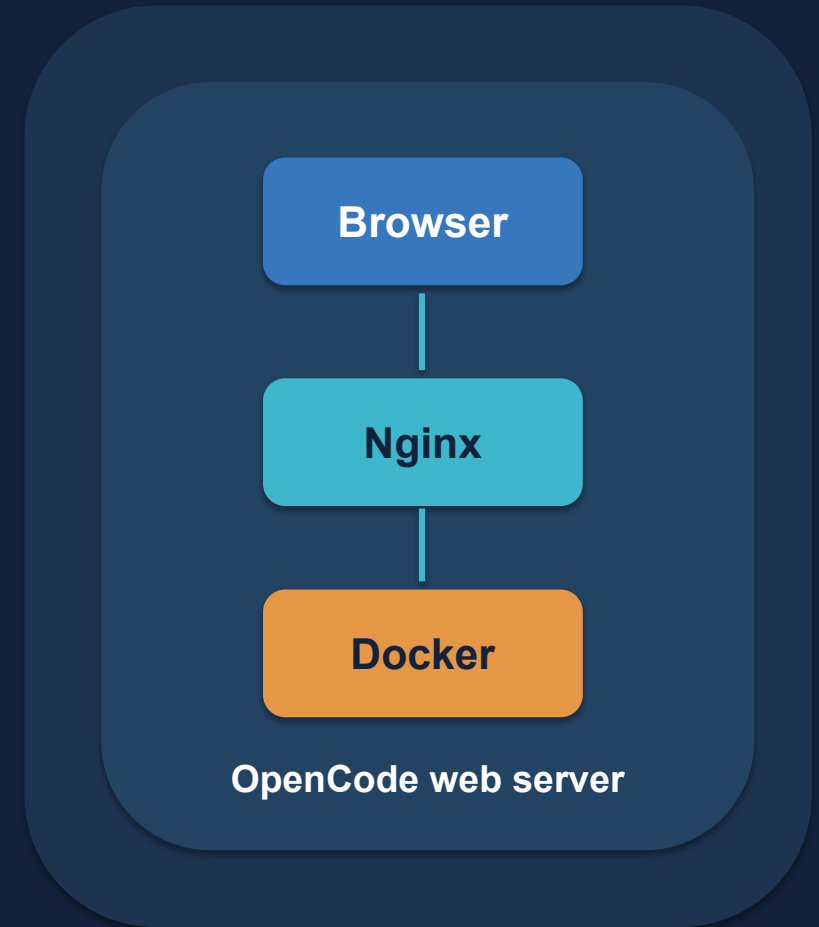


DEPLOYMENT BASICS

From code to a web application

Docker • Dockerfiles • Docker Compose • Nginx



Why do we need deployment?

The goal: anyone on the CMU network can open a browser and use your team's OpenCode instance.



01

Your code

A modified OpenCode application lives in a Git repository.



02

A server runs it

The VM runs the web server even when you disconnect from SSH.



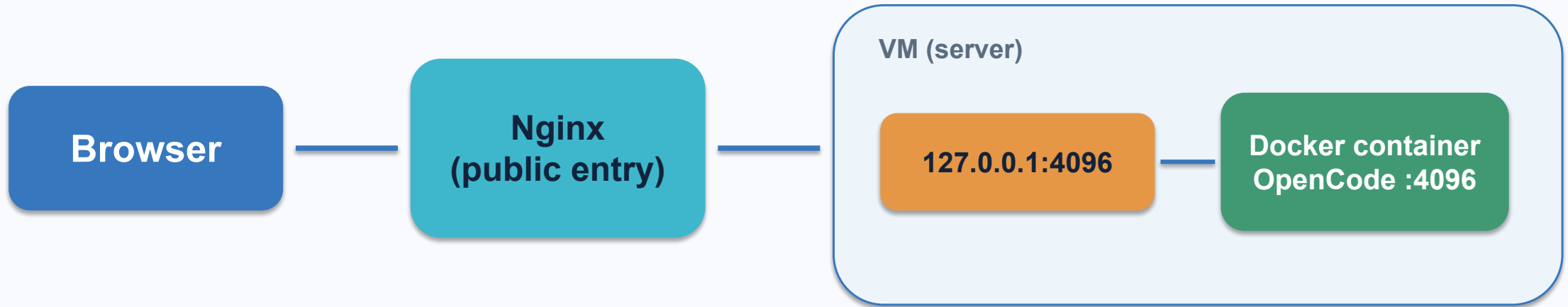
03

People access it

A browser request reaches your app through a public entry point.

Deployment connects the application you built to a server and a route that users can reach.

The big picture: follow one request (P3)



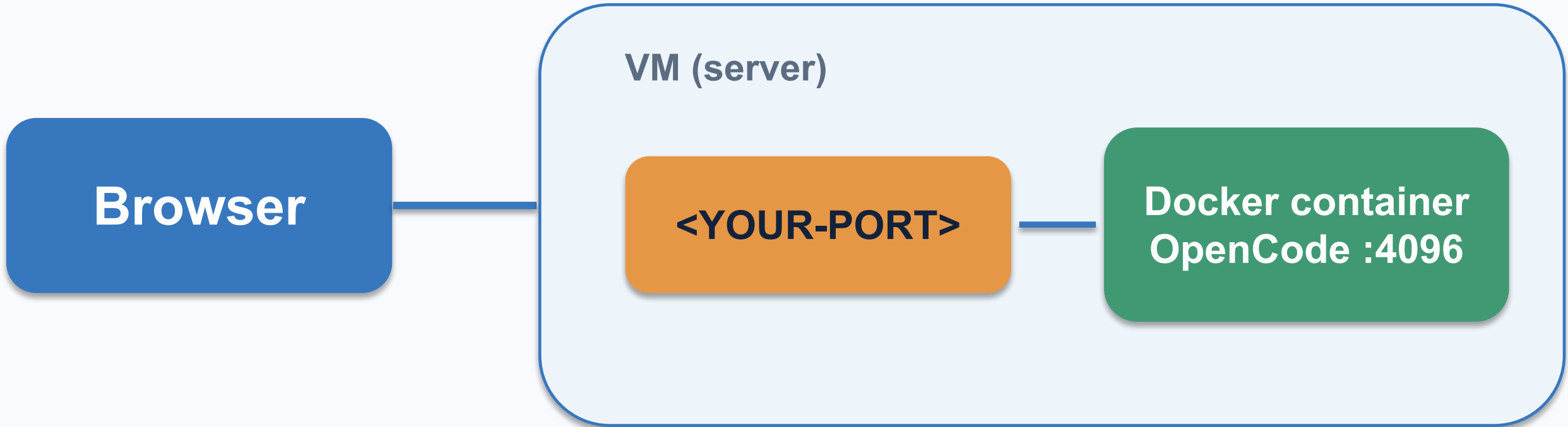
Response travels back through Nginx to the browser

Nginx forwards the request to the VM's local port; Docker maps that port to port 4096 inside the container.

Key idea: Nginx is public-facing; OpenCode stays bound to the VM's local interface.

The big picture: follow one request (For now)

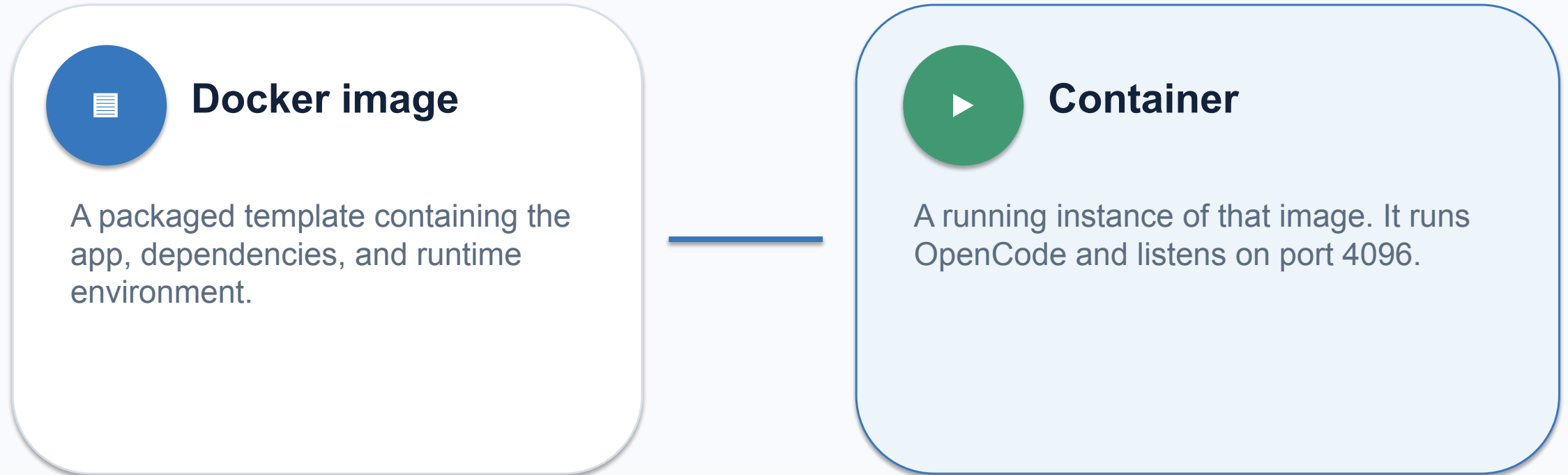
The writeup's request flow, from the user's browser to OpenCode and back.



<YOUR-PORT>: 9000 + Your number

Docker: package the app's environment

A container is where the OpenCode web server runs on the VM.



Mental model Image = packaged recipe/result • Container = running application

Dockerfile: describe how to build the image

You write a Dockerfile in your repository; Docker uses it to build your team's image.

```
FROM oven/bun:1.3.14-slim
WORKDIR /workspaces/opencode
COPY . .
RUN bun install
CMD ["bun", "run", "packages/opencode/src/index.ts", "web",
    "--hostname", "0.0.0.0", "--port", "4096"]
```

FROM

Choose a base environment.

WORKDIR

Set the working directory inside the image.

COPY / RUN

Copy the project and prepare dependencies.

CMD

Define the command that starts OpenCode.

The Dockerfile defines the image; it does not by itself keep the app running in the background.

Docker Compose: configure and run the service

Compose reads docker-compose.yml and starts the application with the settings you specify.

```
services:
  opencode-<ANDREW-ID>:
    build: .
    ports:
      - "127.0.0.1:4096:4096"
    restart: unless-stopped
```

build: .

Build the image using the Dockerfile in this directory.

127.0.0.1:4096:4096

Map VM-local port 4096 to container port 4096.

restart: unless-stopped

Restart after an unexpected stop or Docker restart—unless you explicitly stopped it.

```
docker compose up -d
```

- **-d runs it in the background**

Nginx (For P3): the public reverse proxy

OpenCode is local-only; Nginx receives browser requests and forwards them to OpenCode.



Why this setup? Users reach Nginx; OpenCode itself is not directly exposed on the VM's public interface.

Put the pieces together

The recitation follows this sequence to deploy OpenCode on the provided VM.

- 1 Connect** SSH to the VM and clone your repository.
- 2 Build** Write a Dockerfile to define the app environment.
- 3 Run** Use Docker Compose to build and run the container in the background.
- 4 Recover** Enable Docker at boot; use a restart policy for the container.

Target outcome: a browser-accessible OpenCode instance that continues running after you disconnect.