CCGC 5001 - Virtualization

Module 7B: Multi-container Apps

HUMBER

Module objectives

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

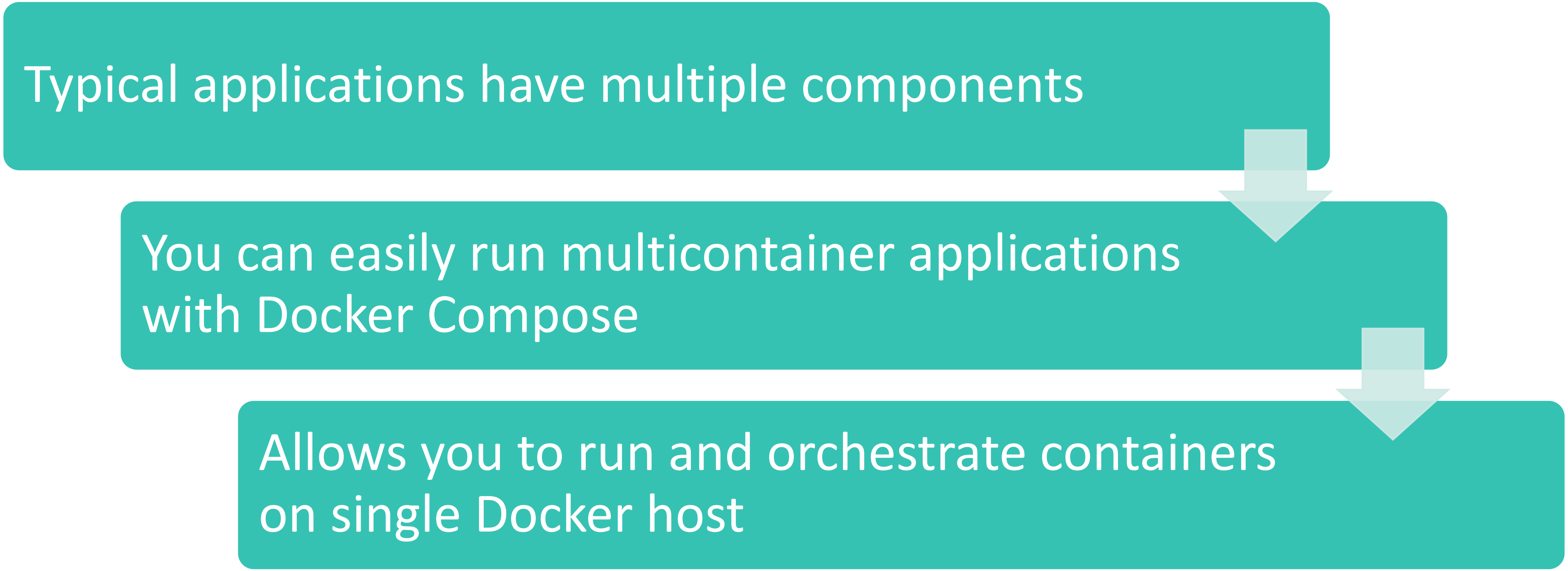
- Explain differences between an imperative and declarative approach
- Describe differences between a container and Docker Compose service
- Author a Docker Compose YAML file
- Build, push, deploy, and tear down multi-service application using Docker Compose
- Use Docker Compose to scale an application

Declarative vs. Imperative

Imperative: A way in which we can solve problems by specifying the exact procedure that has to be followed by the system.

Declarative: A way in which we can solve problems without requiring the programmer to specify an exact procedure to be followed.

Docker Compose



Typical applications have multiple components

You can easily run multicontainer applications with Docker Compose

Allows you to run and orchestrate containers on single Docker host

Docker Compose

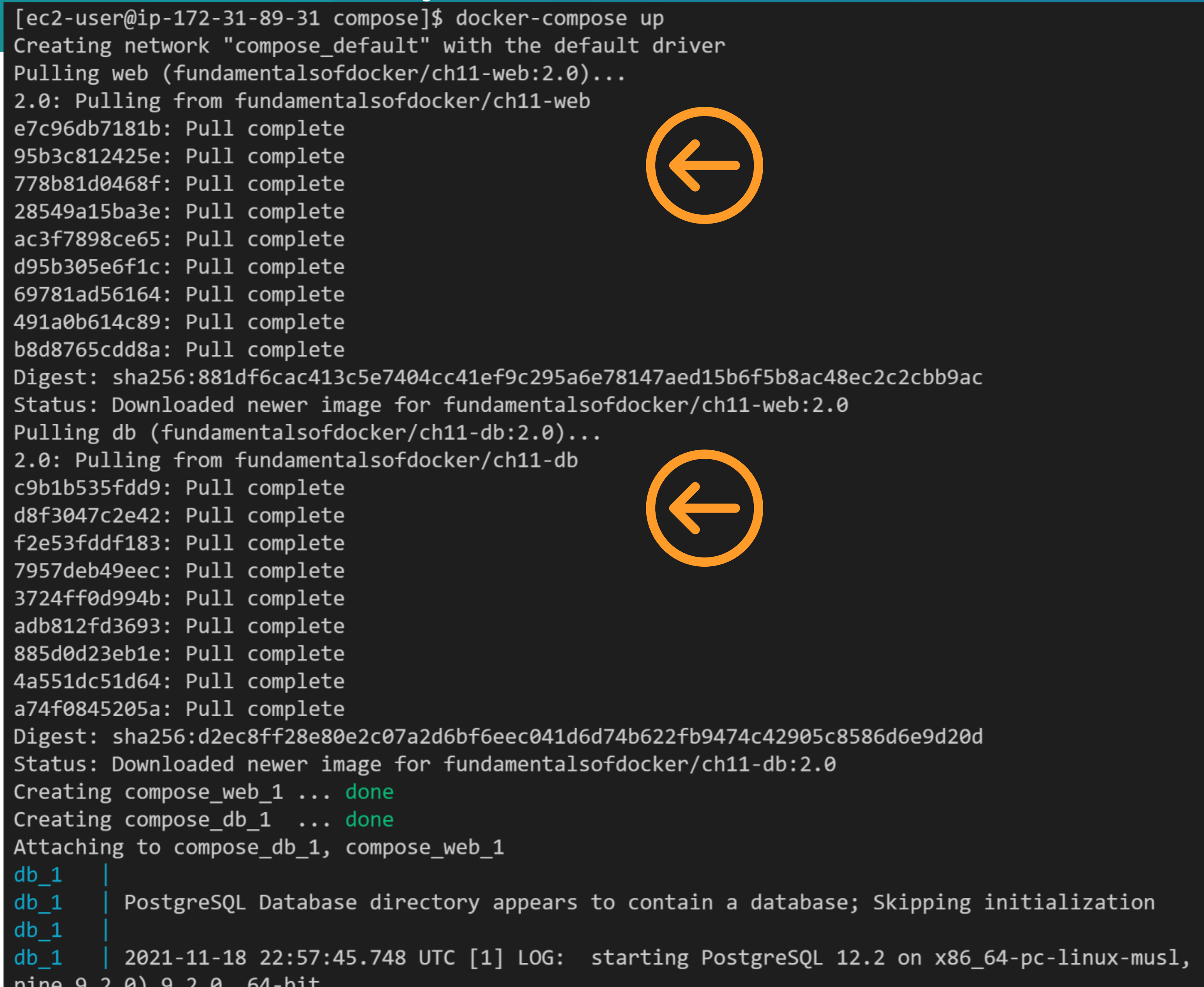
- Define and run multicontainer applications
- Configuration defined in one or more files
  - docker-compose.yml (default)
  - docker-compose.override.yml (default)
- Multiple files specified using -f
- Single command to manage all services
- Great for dev, staging, and CI

Running a multi-service app

```yaml
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
    ports:
      - 3000:3000
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

Building images with Docker Compose

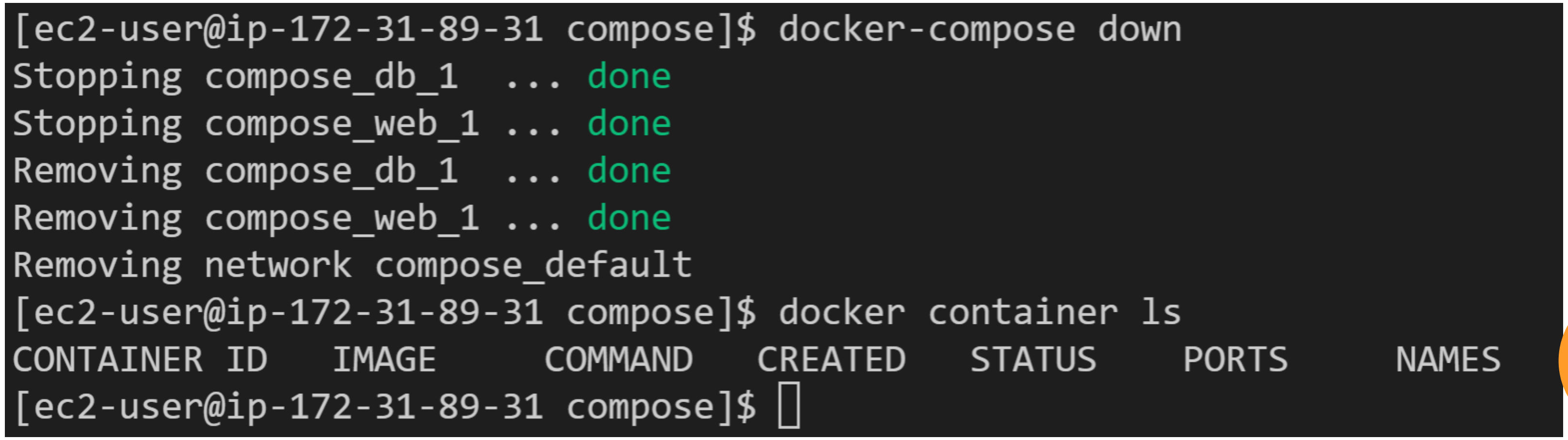
$ docker-compose up



```
[ec2-user@ip-172-31-89-31 compose]$ docker-compose up
Creating network "compose_default" with the default driver
Pulling web (fundamentalsofdocker/ch11-web:2.0)...
2.0: Pulling from fundamentalsofdocker/ch11-web
e7c96db7181b: Pull complete
95b3c812425e: Pull complete
778b81d0468f: Pull complete
28549a15ba3e: Pull complete
ac3f7898ce65: Pull complete
d95b305e6f1c: Pull complete
69781ad56164: Pull complete
491a0b614c89: Pull complete
b8d8765cdd8a: Pull complete
Digest: sha256:881df6cac413c5e7404cc41ef9c295a6e78147aed15b6f5b8ac48ec2c2cbb9ac
Status: Downloaded newer image for fundamentalsofdocker/ch11-web:2.0
Pulling db (fundamentalsofdocker/ch11-db:2.0)...
2.0: Pulling from fundamentalsofdocker/ch11-db
c9b1b535fdd9: Pull complete
d8f3047c2e42: Pull complete
f2e53fddf183: Pull complete
7957deb49eec: Pull complete
3724ff0d994b: Pull complete
adb812fd3693: Pull complete
885d0d23eb1e: Pull complete
4a551dc51d64: Pull complete
a74f0845205a: Pull complete
Digest: sha256:d2ec8ff28e80e2c07a2d6bf6eec041d6d74b622fb9474c42905c8586d6e9d20d
Status: Downloaded newer image for fundamentalsofdocker/ch11-db:2.0
Creating compose_web_1 ... done
Creating compose_db_1  ... done
Attaching to compose_db_1, compose_web_1
db_1    |
db_1    | PostgreSQL Database directory appears to contain a database; Skipping initialization
db_1    |
db_1    | 2021-11-18 22:57:45.748 UTC [1] LOG:  starting PostgreSQL 12.2 on x86_64-pc-linux-musl,
pine 9.2.0) 9.2.0, 64-bit
```

Building images with Docker Compose

$ docker-compose down



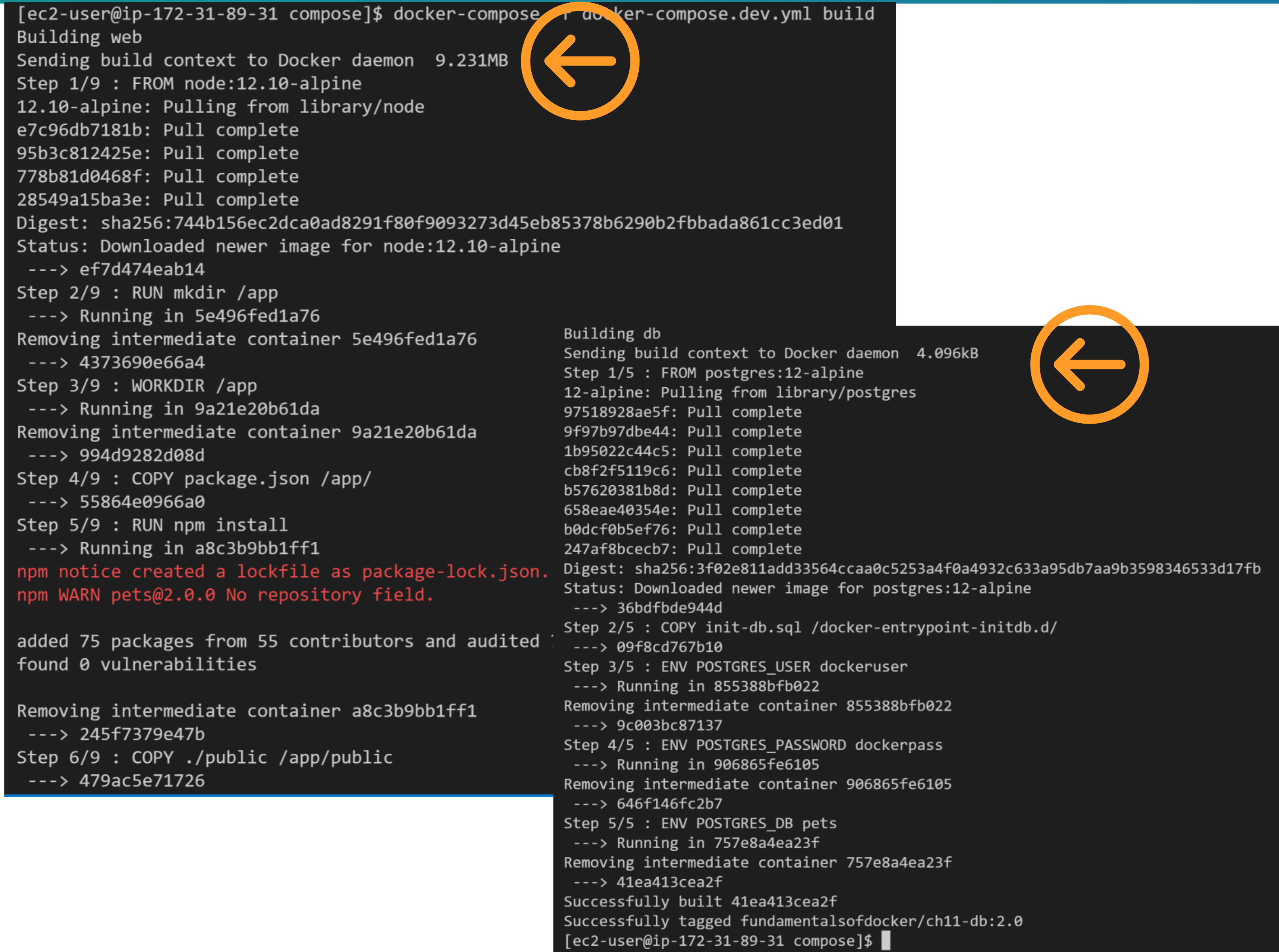
```
[ec2-user@ip-172-31-89-31 compose]$ docker-compose down
Stopping compose_db_1  ... done
Stopping compose_web_1 ... done
Removing compose_db_1  ... done
Removing compose_web_1 ... done
Removing network compose_default
[ec2-user@ip-172-31-89-31 compose]$ docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[ec2-user@ip-172-31-89-31 compose]$ 
```

Running a multi-service app

```yaml
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
    build: web
    ports:
      - 3000:3000
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    build: database
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

Building images with Docker Compose

docker-compose -f docker-compose.dev.yml build

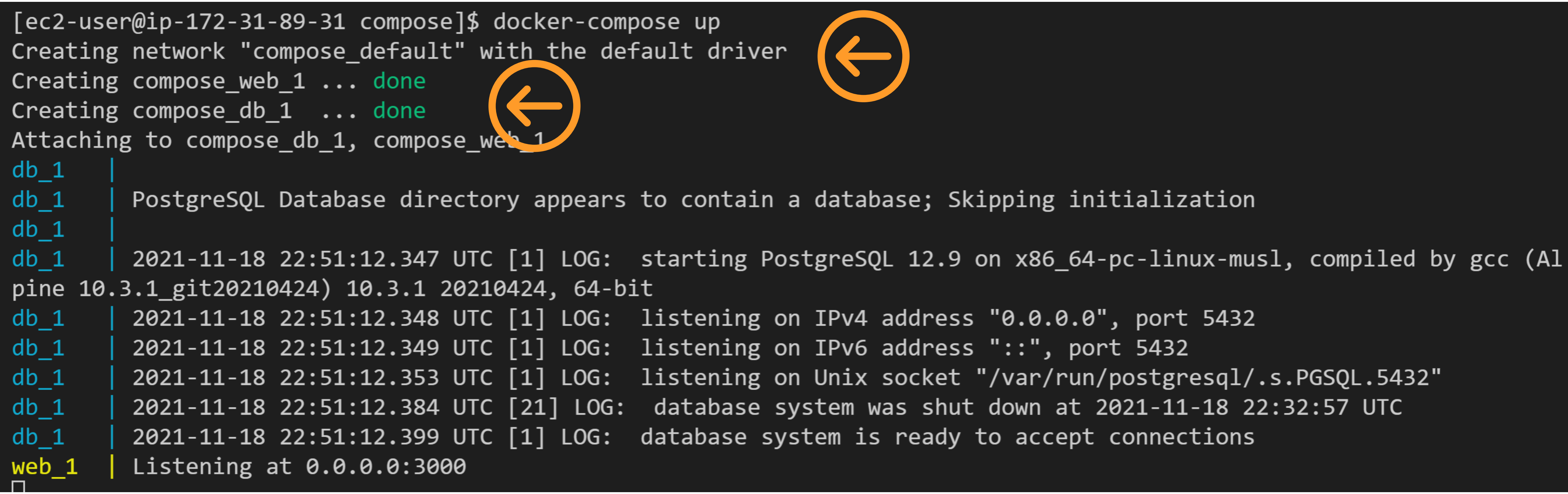


```
[ec2-user@ip-172-31-89-31 compose]$ docker-compose -f docker-compose.dev.yml build
Building web
Sending build context to Docker daemon  9.231MB
Step 1/9 : FROM node:12.10-alpine
12.10-alpine: Pulling from library/node
e7c96db7181b: Pull complete
95b3c812425e: Pull complete
778b81d0468f: Pull complete
28549a15ba3e: Pull complete
Digest: sha256:744b156ec2dca0ad8291f80f9093273d45eb85378b6290b2fbbada861cc3ed01
Status: Downloaded newer image for node:12.10-alpine
 ---> ef7d474eab14
Step 2/9 : RUN mkdir /app
 ---> Running in 5e496fed1a76
Removing intermediate container 5e496fed1a76
 ---> 4373690e66a4
Step 3/9 : WORKDIR /app
 ---> Running in 9a21e20b61da
Removing intermediate container 9a21e20b61da
 ---> 994d9282d08d
Step 4/9 : COPY package.json /app/
 ---> 55864e0966a0
Step 5/9 : RUN npm install
 ---> Running in a8c3b9bb1ff1
npm notice created a lockfile as package-lock.json.
npm WARN pets@2.0.0 No repository field.

added 75 packages from 55 contributors and audited
found 0 vulnerabilities

Removing intermediate container a8c3b9bb1ff1
 ---> 245f7379e47b
Step 6/9 : COPY ./public /app/public
 ---> 479ac5e71726
```

```
Building db
Sending build context to Docker daemon  4.096kB
Step 1/5 : FROM postgres:12-alpine
12-alpine: Pulling from library/postgres
97518928ae5f: Pull complete
9f97b97dbe44: Pull complete
1b95022c44c5: Pull complete
cb8f2f5119c6: Pull complete
b57620381b8d: Pull complete
658eae40354e: Pull complete
b0dcf0b5ef76: Pull complete
247af8bcecb7: Pull complete
Digest: sha256:3f02e811add33564ccaa0c5253a4f0a4932c633a95db7aa9b3598346533d17fb
Status: Downloaded newer image for postgres:12-alpine
 ---> 36bdfbde944d
Step 2/5 : COPY init-db.sql /docker-entrypoint-initdb.d/
 ---> 09f8cd767b10
Step 3/5 : ENV POSTGRES_USER dockeruser
 ---> Running in 855388bfb022
Removing intermediate container 855388bfb022
 ---> 9c003bc87137
Step 4/5 : ENV POSTGRES_PASSWORD dockerpass
 ---> Running in 906865fe6105
Removing intermediate container 906865fe6105
 ---> 646f146fc2b7
Step 5/5 : ENV POSTGRES_DB pets
 ---> Running in 757e8a4ea23f
Removing intermediate container 757e8a4ea23f
 ---> 41ea413cea2f
Successfully built 41ea413cea2f
Successfully tagged fundamentalsofdocker/ch11-db:2.0
[ec2-user@ip-172-31-89-31 compose]$ █
```

Running an application with Docker Compose

$ docker-compose up

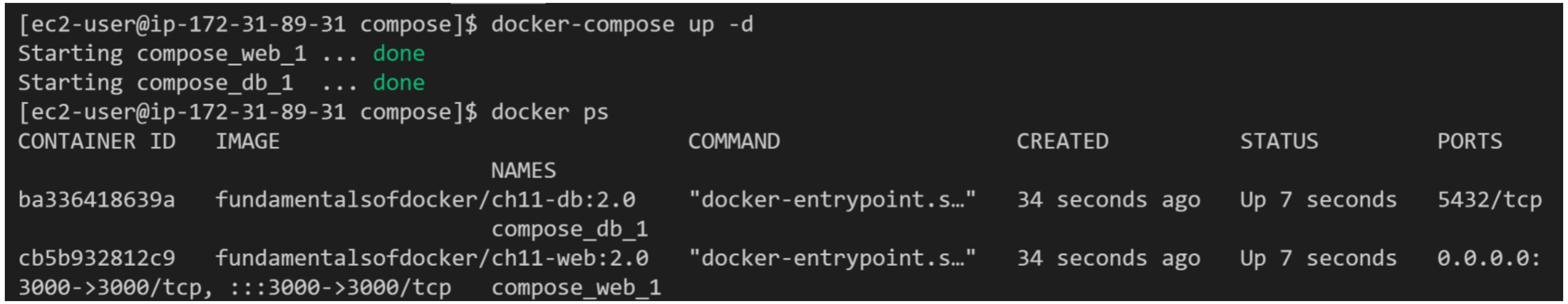


```
[ec2-user@ip-172-31-89-31 compose]$ docker-compose up
Creating network "compose_default" with the default driver
Creating compose_web_1 ... done
Creating compose_db_1  ... done
Attaching to compose_db_1, compose_web_1
db_1    |
db_1    | PostgreSQL Database directory appears to contain a database; Skipping initialization
db_1    |
db_1    | 2021-11-18 22:51:12.347 UTC [1] LOG:  starting PostgreSQL 12.9 on x86_64-pc-linux-musl, compiled by gcc (Al
pine 10.3.1_git20210424) 10.3.1 20210424, 64-bit
db_1    | 2021-11-18 22:51:12.348 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db_1    | 2021-11-18 22:51:12.349 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db_1    | 2021-11-18 22:51:12.353 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db_1    | 2021-11-18 22:51:12.384 UTC [21] LOG:  database system was shut down at 2021-11-18 22:32:57 UTC
db_1    | 2021-11-18 22:51:12.399 UTC [1] LOG:  database system is ready to accept connections
web_1   | Listening at 0.0.0.0:3000
```

Running an application with Docker Compose

$ docker-compose up -d

$ docker-compose ps



```
[ec2-user@ip-172-31-89-31 compose]$ docker-compose up -d
Starting compose_web_1 ... done
Starting compose_db_1  ... done
[ec2-user@ip-172-31-89-31 compose]$ docker ps
CONTAINER ID   IMAGE                               COMMAND                  CREATED          STATUS         PORTS
                                    NAMES
ba336418639a   fundamentalsofdocker/ch11-db:2.0    "docker-entrypoint.s…"   34 seconds ago   Up 7 seconds   5432/tcp
                                    compose_db_1
cb5b932812c9   fundamentalsofdocker/ch11-web:2.0   "docker-entrypoint.s…"   34 seconds ago   Up 7 seconds   0.0.0.0:
3000->3000/tcp, :::3000->3000/tcp   compose_web_1
```

To stop and clean up the application:

$ docker-compose down

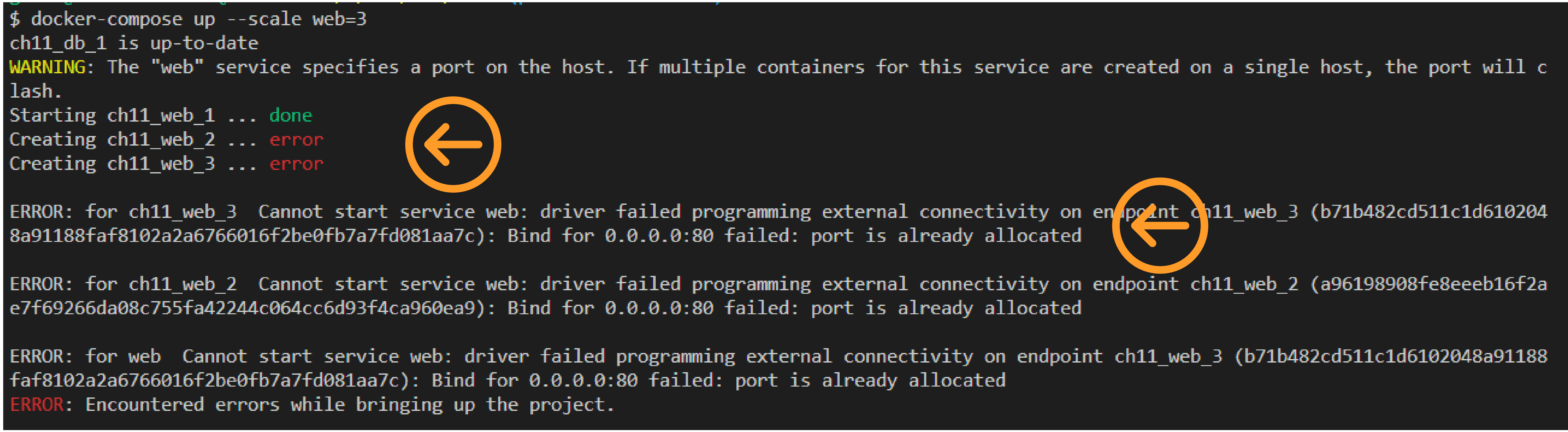
To clean up the application and remove the volumes:

$ docker-compose down -v

Scaling a service

Running more instances is called scaling up.

$ docker-compose up --scale web=3



```
$ docker-compose up --scale web=3
ch11_db_1 is up-to-date
WARNING: The "web" service specifies a port on the host. If multiple containers for this service are created on a single host, the port will c
lash.
Starting ch11_web_1 ... done
Creating ch11_web_2 ... error
Creating ch11_web_3 ... error

ERROR: for ch11_web_3  Cannot start service web: driver failed programming external connectivity on endpoint ch11_web_3 (b71b482cd511c1d610204
8a91188faf8102a2a6766016f2be0fb7a7fd081aa7c): Bind for 0.0.0.0:80 failed: port is already allocated

ERROR: for ch11_web_2  Cannot start service web: driver failed programming external connectivity on endpoint ch11_web_2 (a96198908fe8eeeb16f2a
e7f69266da08c755fa42244c064cc6d93f4ca960ea9): Bind for 0.0.0.0:80 failed: port is already allocated

ERROR: for web  Cannot start service web: driver failed programming external connectivity on endpoint ch11_web_3 (b71b482cd511c1d6102048a91188
faf8102a2a6766016f2be0fb7a7fd081aa7c): Bind for 0.0.0.0:80 failed: port is already allocated
ERROR: Encountered errors while bringing up the project.
```

Scaling a service

```yaml
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
    build: web
    ports:
      - 3000
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    build: database
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

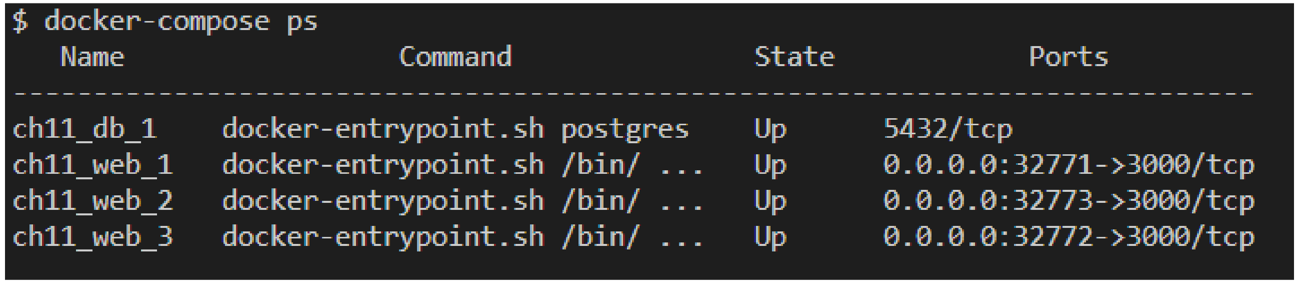
Automatically ephemeral port is selected on host

Scaling a service

Running more instances is called scaling up.

$ docker-compose up -d --scale web=3

$ docker-compose ps



```
$ docker-compose ps
   Name                  Command               State            Ports
---------------------------------------------------------------------------
ch11_db_1     docker-entrypoint.sh postgres    Up      5432/tcp
ch11_web_1    docker-entrypoint.sh /bin/ ...   Up      0.0.0.0:32771->3000/tcp
ch11_web_2    docker-entrypoint.sh /bin/ ...   Up      0.0.0.0:32773->3000/tcp
ch11_web_3    docker-entrypoint.sh /bin/ ...   Up      0.0.0.0:32772->3000/tcp
```

```
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
    build: web
    ports:
      - 3000
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    build: database
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

Building and pushing an application

Build key indicates where Docker expecting to find Dockerfile.

$ docker-compose -f docker-compose.dev.yml build

-f parameter instructs Docker Compose which compose file to use

$ docker login -u <dockerid> -p <password>

$ docker-compose -f docker-compose.dev.yml push

```
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
    build:
      context: web
      dockerfile: Dockerfile-dev
    ports:
      - 3000:3000
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    build: database
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

Docker Compose overrides

Useful when we want to run applications in different environments that need specific configuration settings.

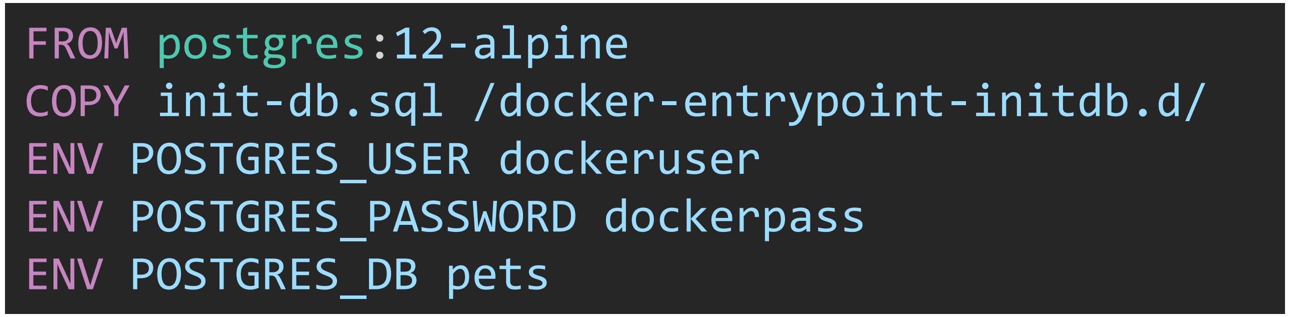
docker-compose.base.yml

```yaml
version: "2.4"
services:
  web:
    image: fundamentalsofdocker/ch11-web:2.0
  db:
    image: fundamentalsofdocker/ch11-db:2.0
    volumes:
      - pets-data:/var/lib/postgresql/data

volumes:
  pets-data:
```

Let's assume for a moment that we want to run our sample application on a **CI** (continuous integration) system

Docker Compose overrides

Dockerfile:



```
FROM postgres:12-alpine
COPY init-db.sql /docker-entrypoint-initdb.d/
ENV POSTGRES_USER dockeruser
ENV POSTGRES_PASSWORD dockerpass
ENV POSTGRES_DB pets
```

Docker Compose overrides

On the CI system, we want to do the following:

- Build the images from code
- Define POSTGRES_PASSWORD as ci-pass
- Map container port 3000 of the web service to host port 5000

```
version: "2.4"
services:
  web:
    build: web
    ports:
      - 5000:3000
    environment:
      POSTGRES_PASSWORD:ci-pass

  db:
    build: database
    environment:
      POSTGRES_PASSWORD:ci-pass
```

```
$ docker-compose -f docker-compose.base.yml -f docker-compose-ci.yml up -d --build
```

Module summary

In summary, in this module, you learned:

- How to use Docker Compose tool
- How to run and scale multi-service applications on single Docker host
- How to construct YAML files to create services
- How to build and push images using Docker Compose

Thank you