Multi-Container Applications

Course Map

1. Developers, Containers, the Cloud
2. Getting Started with Docker
3. Exercise: Installing and using Docker
4. Working with Dockerfiles and ASP.NET Core apps
5. Exercise: Dockerfiles, ASP.NET Core apps

Course Map

6. Exercise: Mapping to Source Code
7. Exercise: Mapping to Shared Data
8. **Multi-Container Applications with Docker Compose**
9. Exercise: Linking vs networked Containers, using Docker Compose

Agenda

- Multi-container applications
- Old way: linking containers
- New way: bridge networks
- Automation with docker-compose
- Docker-compose Yaml files
- Docker-compose commands

Get the Bits

github.com / tonysneed /

Kliant.DockerForDevs

Multi-container applications

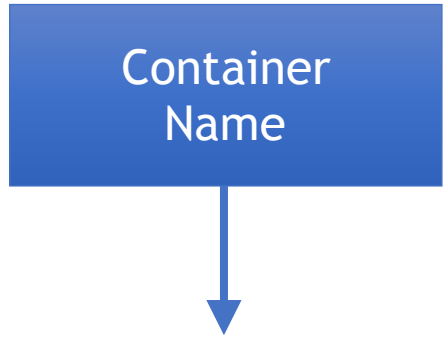
- Containers often need to communicate
  - **Web app** talks to a **web service**
  - **Web service** talks to a **database**

Old way: Linking containers

- One container **links** to another by **name**
- Container **name** becomes the *host name*

Problem: *All containers are exposed to the outside world!*

Example: Linking containers




```
docker run -d -p 27017:27017 --name my-mongo mongo
docker run -d -p 3000:3000 --link my-mongo:my-mongodb mongoose
```

New way: Bridge networks

- Create custom bridge network
- Assign containers to the network
- Containers communicate with one another by name

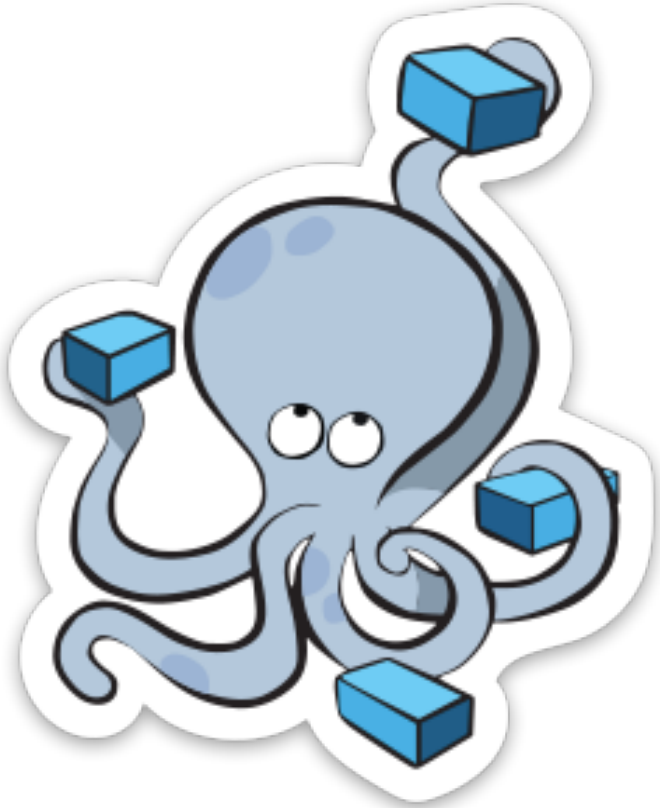
Solution: *Containers can be isolated from the outside world!*

Example: Bridge networks

Demo: Linked vs networked containers

Docker Compose

- Define **muli-container** app
- Use **docker-compose.yml** file
- Link **multiple** containers
- Use **commands** to manage whole app
- **Build**, **start** and **tear down** services



Sample: docker-compose.yml

```
version: '2'

services:

my-mongodb:
    image: mongo:latest
    networks:
        - my_network
```

Sample: docker-compose.yml (cont.)

```
mongoose:
    build:
        context: ./MongooseExpress
        dockerfile: mongoose.dockerfile
    networks:
        - my_network
```

Sample: docker-compose.yml (cont.)

```
aspnetcore:
    build:
        context: ./DockerComposeDemo
        dockerfile: aspnetcore.dockerfile
    ports:
        - "5000:5000"
    networks:
        - my_network
```

Sample: docker-compose.yml (cont.)

```
networks:
    my_network:
        driver: bridge
```

Docker Compose Commands

docker-compose **build**

docker-compose **up**, **down**

docker-compose **start**, **stop**

docker-compose **ps -a**

docker-compose **logs**

docker-compose **rm**

Demo: Docker Compose

Questions?