

Using Docker compose

(Demo prep)

Using Docker Compose

Before we start using Docker compose let's first create a new directory where we can place the compose yaml file and any future infrastructure files.

1. Open new VS Code window

2. Switch to D:\projects and:

MD Play.Infra

3. Open Play.Infra in VS Code

4. Install the Docker extension

5. Create docker-compose.yml

6. Append the docker run command as comment at the top:

```
# docker run -d --rm --name mongo -p 27017:27017 -v mongodbddata:/data/db mongo
```

7. Enable the setting: Render Whitespace → all

8. Create docker-compose.yml:

```
version: "3.8"
```

```
services:
```

```
  mongo:
```

```
    image: mongo
```

```
    container_name: mongo
```

```
    ports:
```

```
      - 27017:27017
```

```
    volumes:
```

```
      - mongodbddata:/data/db
```

```
volumes:
```

```
  mongodbddata:
```

9. Stop MongoDB container

10. `docker-compose up`
11. Query Catalog items in Postman. Notice no items.
12. Show Catalog DB is not there in MongoDB extension
13. Post a new Potion:

```
{  
  "name": "Potion",  
  "description": "Restores a small amount of HP",  
  "price": 7  
}
```

14. CTRL+C to stop compose
15. Start compose with:

 `docker-compose up -d`

So, we have reached the end of this module with a few improvements to our developer workflow and with a reusable library that will speed up the construction of future microservices. In the next module we will introduce our second microservice and will learn about the multiple challenges of having microservices talk to each other.