

Microservices Workshop

Agenda

- Managing Services
- Updating Services
- Scaling Services
- Using multiple docker-compose files
- ★ Lab 04: Updating a single service

Stop Services:

```
$ docker-compose stop [SERVICE]
```

♦ Start Services:

\$ docker-compose start [SERVICE]

★ Pause Services:

\$ docker-compose pause [SERVICE]

Unpause Services:

\$ docker-compose unpause [SERVICE]

* Execute commands in Services containers:

```
$ docker-compose exec [options] [SERVICE] [command]
```

Runs a one-time command against a service:

```
$ docker-compose run [options] [SERVICE] [command]
```

Updating Services

Update service(s) container(s):

```
$ docker-compose up [options] [SERVICE]
```

- Compose only recreate containers that have changed
- ★ To prevent Compose from picking up changes, use the --no-recreate flag
- ★ To force Compose to stop and recreate all containers, use --force-recreate

Scaling Services

Scale a service:

```
$ docker-compose up [options] --scale SERVICE=NUM [SERVICE] [command]
```

- ★ Doesn't work for named containers ("container_name")
- ★ Doesn't work for containers that expose ports ("port")
- ★ To scale containers that exposes a port you must use a proxy (LB)

Using multiple docker-compose files

★ Deploying services using multiple docker-compose files:

```
docker-compose -f docker-compose.yml -f docker-compose-override.yml up -d
```

- Compose merges the configurations by override/concatenate configurations
- Useful for manage different environments
- Useful for different types of deployment

Using multiple docker-compose files

Extending Services:

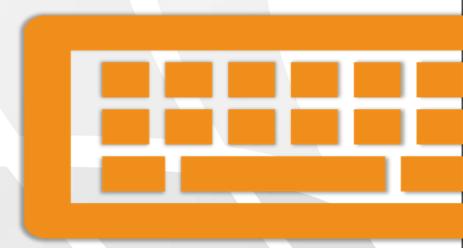
```
web:
extends:
file: common-services.yml
service: webapp
```

- Useful to re-use configurations
- Useful to use a common.yml configuration file

Questions

Lab 04: Updating a Single Service

Lab



https://gitlab.com/sela-microservices-workshop/lab-04