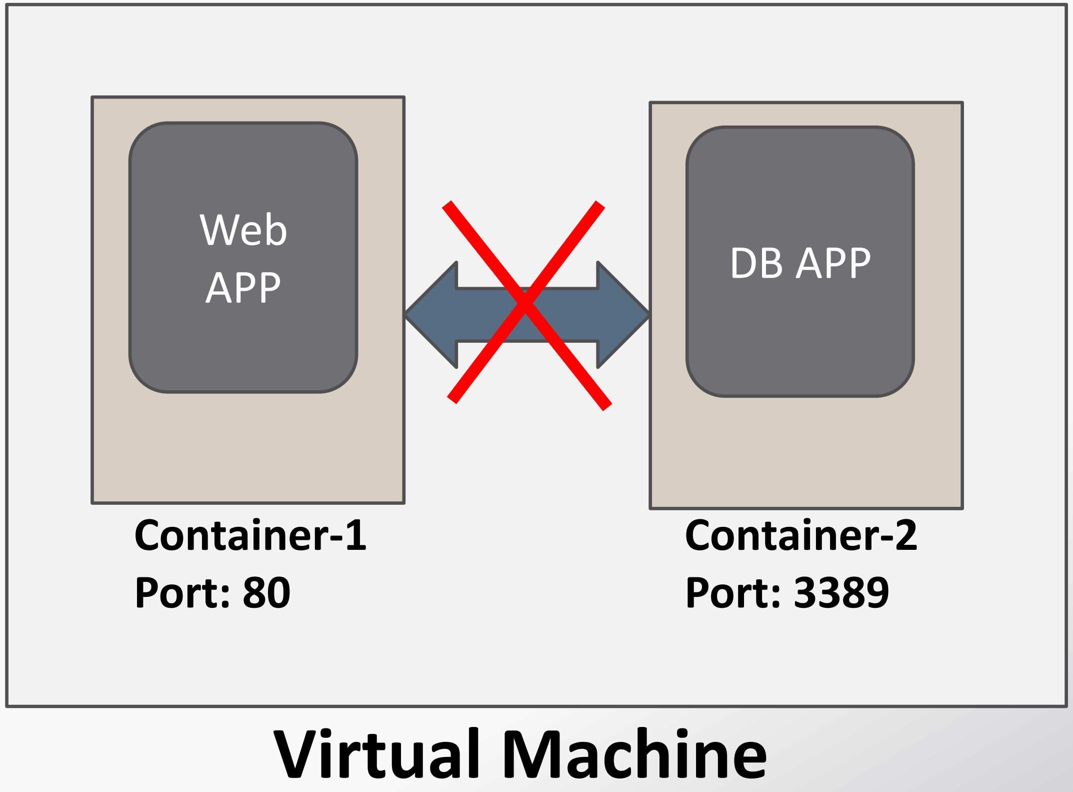
# Docker Compose

VISHWANATH M S

VISHWACLOUDLAB.COM

# Problem with Docker!!!



Web APP

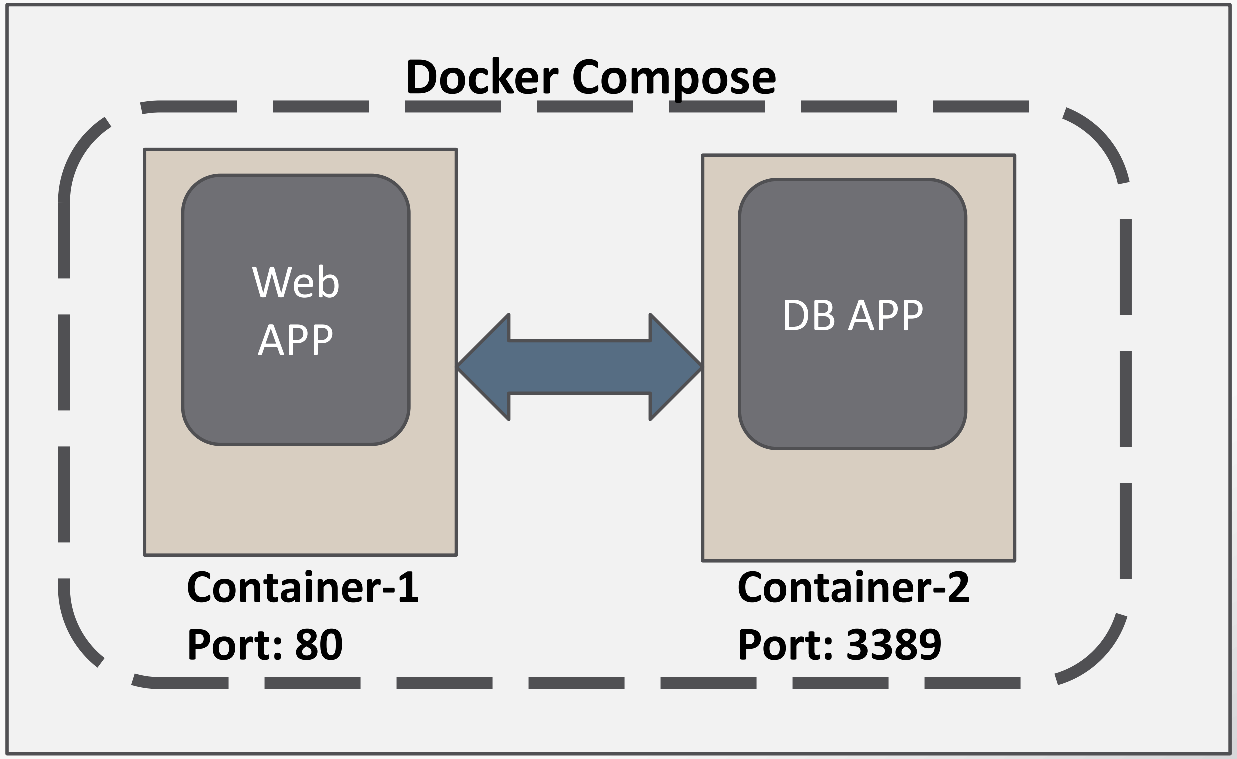
DB APP

Container-1
Port: 80

Container-2
Port: 3389

Virtual Machine

# Solution: Docker Compose



Virtual Machine

# What is Docker Compose?

*Docker compose is “**a tool for defining and running complex applications with Docker”.***

*Run multiple containers from a single config file.*

*Defines Services, Networks and Volumes for a Docker application*

# What is Docker Compose?

Make your development environments:

Repeatable

Isolated

Fast

# What is Docker Compose?

Define and run multi-container applications

All of that can be done by Docker Compose in the scope of a single host.

Specify images and configuration in a simple YAML file:

```
$ docker-compose.yml
```

One command to get it all running:

```
$ docker-compose up
```

# What is Docker Compose?

## *docker-compose up:*

Builds images from Dockerfiles

Pulls images from registries

Creates and starts containers

Streams their logs

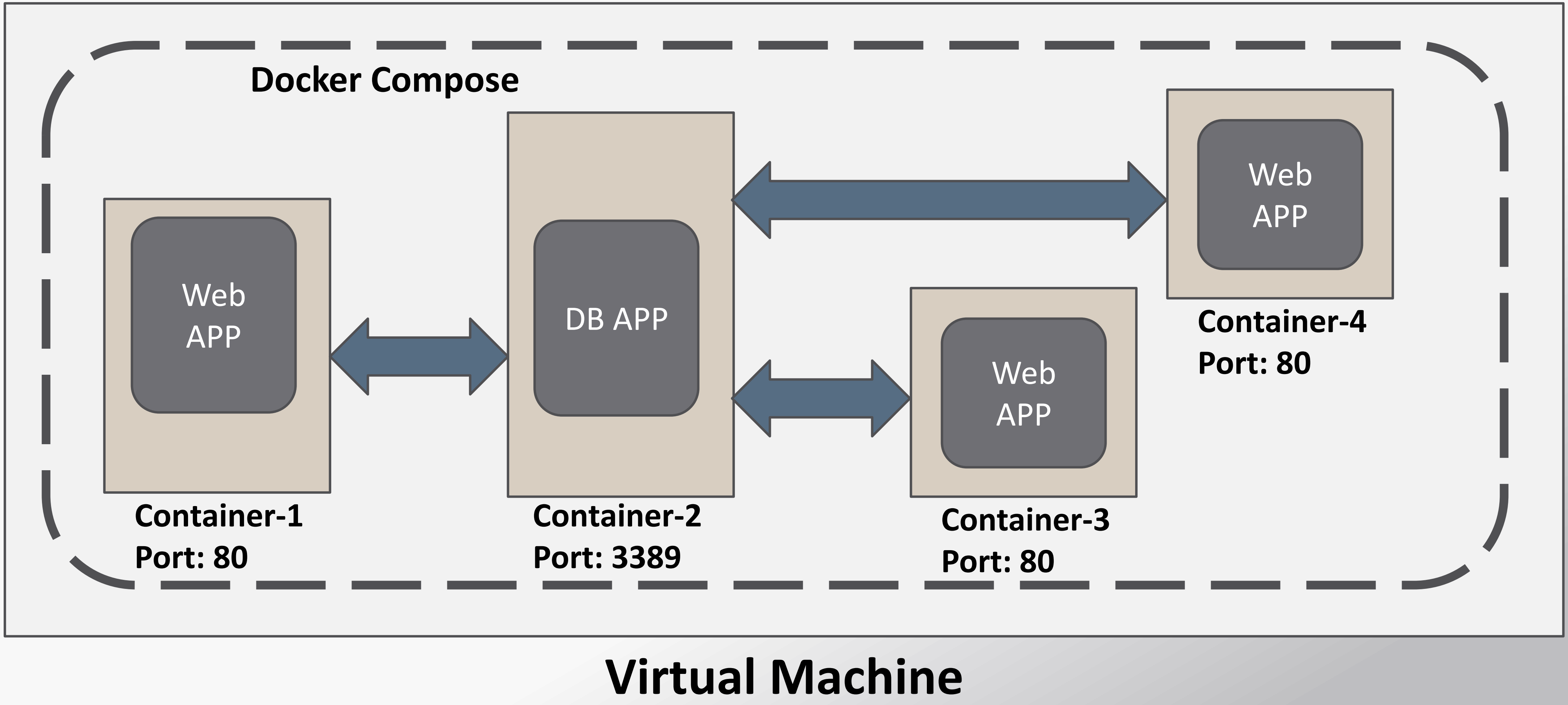
# Build image with Docker Compose

***docker-compose build:***

Builds images from Dockerfiles

Pulls images from registries

# Docker Compose - Scale it…



# Scale with Docker Compose

```
docker-compose scale SERVICE=3
```

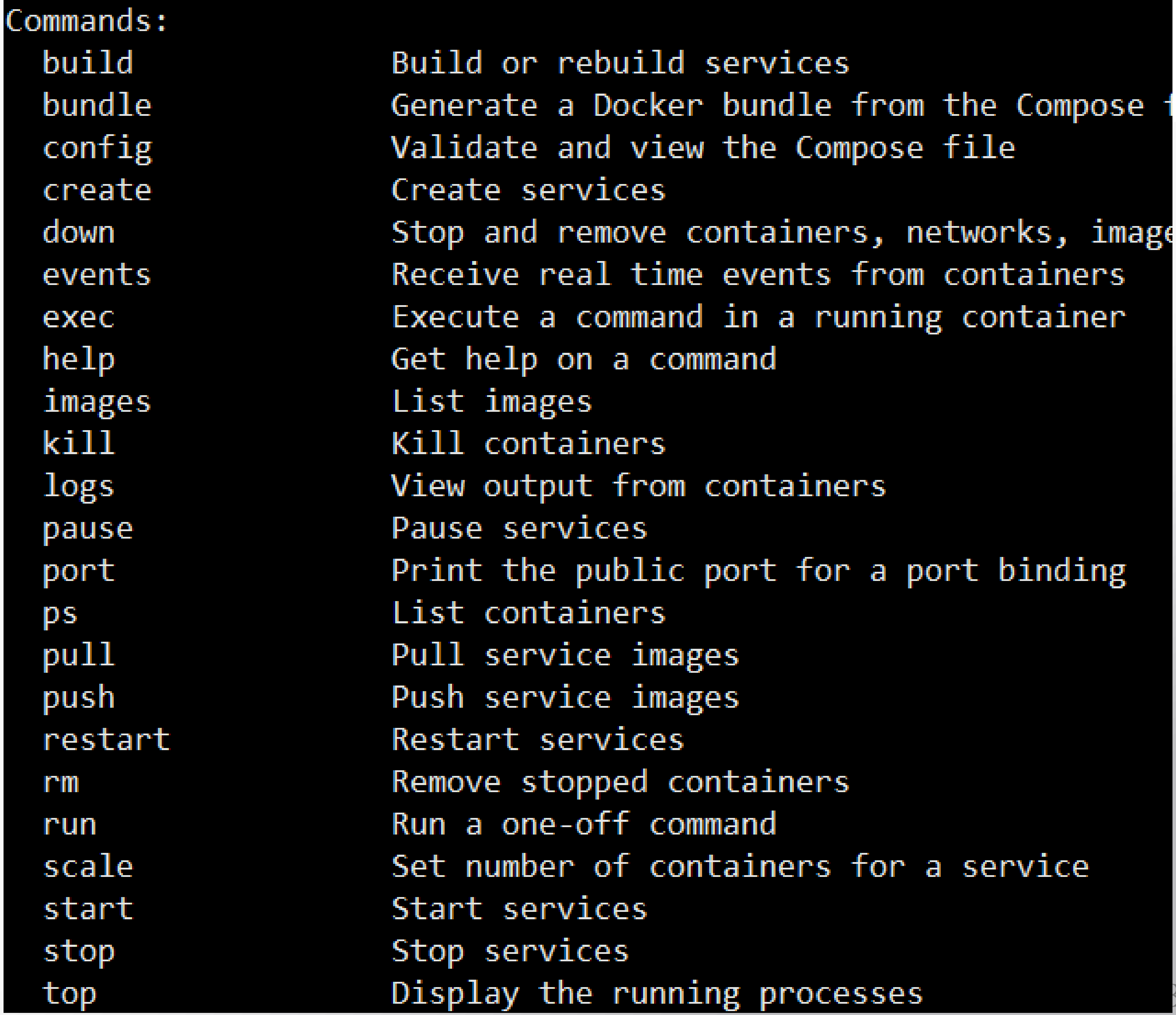
SERVICE == the name of the service defined in the YAML file

Can easy increase the no of environment that can be replicated and decreased

```
docker-compose up --no-recreate
```

Re-running containers that have stopped.

# docker-compose --help



```
Commands:
  build              Build or rebuild services
  bundle             Generate a Docker bundle from the Compose f
  config             Validate and view the Compose file
  create             Create services
  down               Stop and remove containers, networks, image
  events             Receive real time events from containers
  exec               Execute a command in a running container
  help               Get help on a command
  images             List images
  kill               Kill containers
  logs               View output from containers
  pause              Pause services
  port               Print the public port for a port binding
  ps                 List containers
  pull               Pull service images
  push               Push service images
  restart            Restart services
  rm                 Remove stopped containers
  run                Run a one-off command
  scale              Set number of containers for a service
  start              Start services
  stop               Stop services
  top                Display the running processes
```

# Docker Compose Workflow

There are three steps to using Docker Compose:

1. Define each service in a Dockerfile.
2. Define the services and their relation to each other in the **docker-compose.yml** file.
3. Use **docker-compose up** to start the system

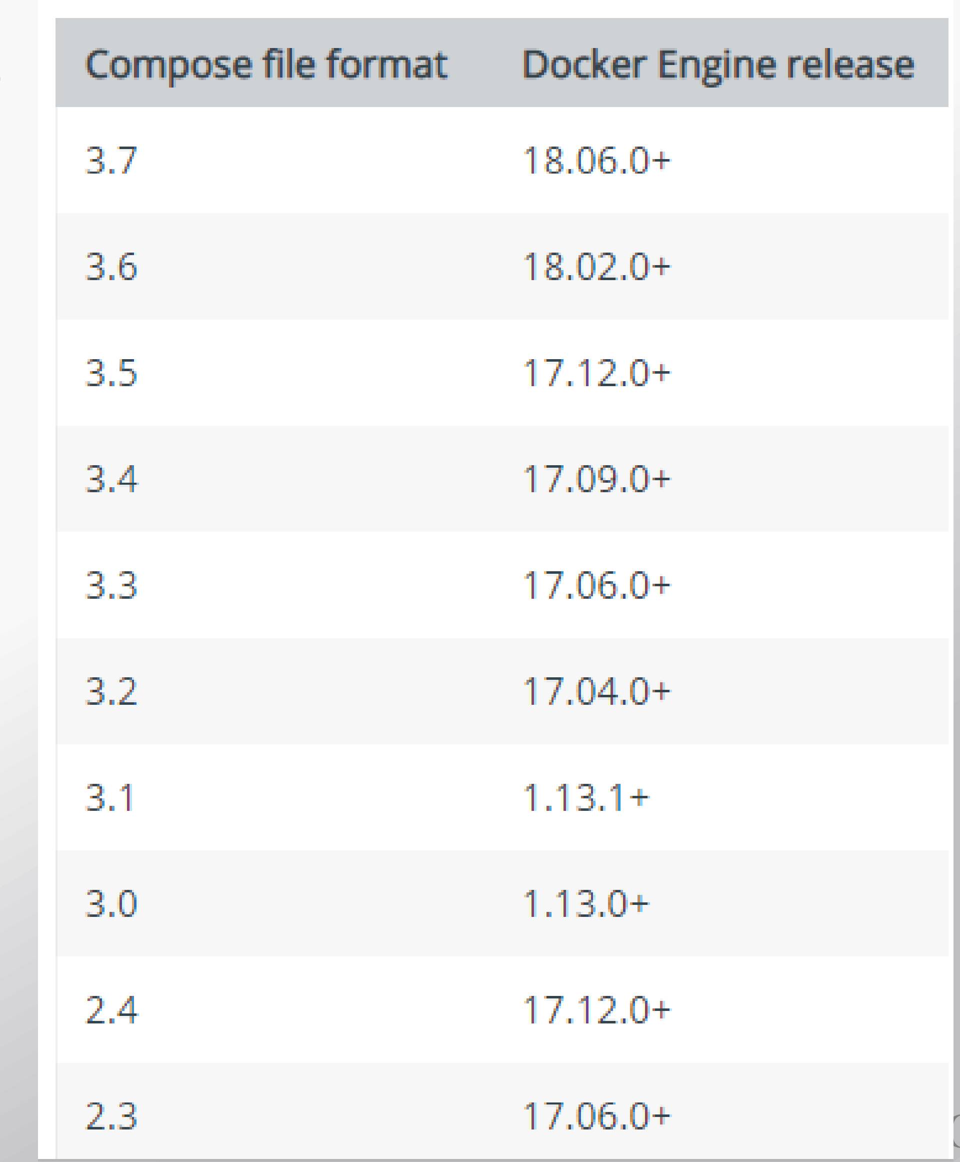
# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
      args:
        buildno: 1
```

# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
      args:
        buildno: 1
```

**Designed to be compatible b/w Compose and Docker Engine**



| Compose file format | Docker Engine release |
|---|---|
| 3.7 | 18.06.0+ |
| 3.6 | 18.02.0+ |
| 3.5 | 17.12.0+ |
| 3.4 | 17.09.0+ |
| 3.3 | 17.06.0+ |
| 3.2 | 17.04.0+ |
| 3.1 | 1.13.1+ |
| 3.0 | 1.13.0+ |
| 2.4 | 17.12.0+ |
| 2.3 | 17.06.0+ |

# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
      args:
        buildno: 1
```

**Containers in Production**

**A Service only runs one image**

**Defines →**

- **What image to run ?**
- **What port it would use ?**
- **How many Replicas ?**
- **Network details !!**
- **Resource limits !!**
- **Which Dockerfile for build?**
- **………….. Any Much More**

# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
      args:
        buildno: 1
```

**Custom name for the service, that hold all the details of the container to be run**

# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
    build:
      context: ./dir
      dockerfile: Dockerfile-alternate
      args:
        buildno: 1
```

**Key Value pair, that contains maps and list**

# What is the structure of the YAML file?

```
version: "3"
services:
  webapp:
     build:
       context: ./dir
       dockerfile: Dockerfile-alternate
       args:
         buildno: 1
```

**Details for the Container creation**

# Docker Compose file  with Example - 1

```
version: '3'
services:
   web:
        build: .
        ports:
            -  “8000:8000"
    redis:
        image: "redis:alpine"
```

First part, to Build and Create custom webserver from the Dockerfile and map the Container port 8000 to Host machine port 8000

Plain and Simple, ready made **Redis** Container,

# More Reads…..

Versioning → https://docs.docker.com/compose/compose-file/compose-versioning/

# What’s new in 1.3.0?

Performance and stability improvements

Lots more config option support

New feature (experimental!): **Smart Recreate**

Only recreate containers whose configuration has been changed

```
$ docker-compose up --x-smart-recreate
```

Will eventually be the default behavior