Docker Compose

Docker Inc. came up with a tool called ‘Fig’ which allowed you to use a single YAML file to orchestrate all your Docker containers and configurations. Later the ‘Fig’ source code was used to create the tool called Docker-Compose.

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a Compose file to configure your application’s services. Then, using a single command, you create and start all the services from your configuration.

Using Compose is basically a three-step process.

1. Define your app’s environment with a `Dockerfile` so it can be reproduced anywhere.
2. Define the services that make up your app in `docker-compose.yml` so they can be run together in an isolated environment.
3. Lastly, run `docker-compose up` and Compose will start and run your entire app.

Sample `docker-compose.yml` file,

```
version: '3'
services:
  web:
    build: .
    ports:
    - "5000:5000"
    volumes:
    - .:/code
    - logvolume01:/var/log
    links:
    - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

```
version: ‘3’
services:
  wordpress:
    image: wordpress:latest
    ports:
      -  8080:80
     environment:
           WORDPRESS_DB_PASSWORD: abc123

  mysql:
    image: mysql:latest
    environment:
```

```
MYSQL_ROOT_PASSWORD: abd123$
```

To get the Docker compose installed on Ubuntu, follow below steps.

```
$ sudo curl -o /usr/local/bin/docker-compose -L
https://github.com/docker/compose/releases/download/1.11.
2/docker-compose-$(uname -s)-$(uname -m)
```

Set the permissions.

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

Check the docker compose version by running the command,

```
$ docker-compose -v
```

Once you have the docker-compose installed, create the .yml file and start creating multiple containers that are linked together.

Create a directory (e.g. dockerproject) and a file names as `docker-compose.yml`

The .yml file would have details about the image to be used, volumes to be mounted, links to be established, ports to be exposed, if a Dockerfile is used for container creation etc. and all diff images to be used for creating containers.

Commands:

build	Build or rebuild services
bundle	Generate a Docker bundle from the Compose file
config	Validate and view the compose file
create	Create services
down	Stop and remove containers, networks, images, and volumes
events	Receive real time events from containers
exec	Execute a command in a running container
help	Get help on a command
kill	Kill containers
logs	View output from containers

Docker Compose

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
unpause	Unpause services
up	Create and start containers
version	Show the Docker-Compose version information

In docker-compose we can create an override file named as "`docker-compose.override.yml`" file. In such case when one runs the `docker-compose up -d` command the settings / instructions written in docker-compose file would be overridden by the settings written in "`docker-compose.override`" file.