**Docker-compose**

# Synopsis

This document shows the difference between docker-compose up, docker-compose down, docker-compose start, docker-compose stop and docker-compose pause.

# Short description

|  |  |  |
| --- | --- | --- |
| Command | Description | Reference |
| docker-compose up | Does everything to starts containers | https://docs.docker.com/compose/reference/up/ |
| docker-compose down | Stop and remove containers and networks | https://docs.docker.com/compose/reference/down/ |
| docker-compose start | Does starts the stopped containers don’t create new one. | https://docs.docker.com/compose/reference/start/ |
| docker-compose stop | Stop services | https://docs.docker.com/compose/reference/stop/ |
| docker-compose pause | Suspend all the process in the running container | https://docs.docker.com/compose/reference/pause/ |

# Long description

## **docker-compose up**

* Single command which does everything that needs to done to get containers running. It builds, creates, recreates if not existing or is there is any change in the Dockerfile,

Pull the required images from the registry, starts or restarts all containers defined in a docker-compose.yml file in the default mode etc.

* If the images are changed after the container creation, it picks up the changes by stopping the container and recreates the container. It preserves the mounted volume during this process.
* To prevent the picking of changes can use --no-recreate flag with the command.
* If want to stops all running containers and recreate the all the containers using the changes, use --force-recreate flag.
* This command replaces docker-compose build, docker-compose start, docker-compose restart, docker-compose create, docker-compose run, docker-compose scale commands.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present*

PS > docket-compose up [Options] [SERVICE…]

Key flags:

*#Gathering help.*

--help

*#Don’t start the service after creating containers.*

--no-start

*#Run containers on background.*

--detach

*#Re-create the containers based on the changes detected.*

--no-recreate

*#Down the running containers and recreate them based the changes.*

--force-recreate

*#Used to set number for running containers.*

--scale

## **docker-compose down**

* The docker-compose stop command will stop containers, but it won’t remove them.
* The docker-compose down command will stop containers, but it also removes the stopped containers as well as any networks that were created by default.
* If the network and volumes are defined as external are never be removed.
* This command replaces docker-compose stop and docker-compose rm commands.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present.*

PS > docker-compose down [Options] [SERVICE…]

Key flags:

*#Gathering help.*

--help

*#Remove the volumes also.*

--volumes

*#Removes the images. All for all images and local for locally created images.*

--rmi <All|local>

## **docker-compose start**

* It’s useful to start stopped containers or restart the running containers only. It didn’t create any new containers from Dockerfile instructions.
* If we make any changes to docker-compose.yml file, these changes are not reflected after running this command.
* If you want to start a particular container it didn’t help. In such scenarios docker-compose run, command is recommended.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present.*

PS> docker-compose start [SERVICE…]

## **docker-compose run**

* Runs one-time command against a specific container or service.
* It overrides the existing commands defined in the service configuration.
* It should not create any ports defined in the service configuration file. We should need to manually configure the ports using --publish flags.
* If you start a service linked with some other services using run command, it will start all linked services if they aren’t running.
* If we don’t need to start linked services or containers --use no-deps flag. It prevents other linked services from starting.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present.*

PS> docker-compose run [Options] [SERVICE…]

Key flags:

*#Gathering help*

--help

*#Mapping host port with container port.*

--publish

*#Running container in background.*

--detach

*#Override the entrypoint of image.*

--entrypoint

*#Don’t start linked services.*

--no-deps

*#Bind mount a volume.*

--volume

*#Working directory inside the container.*

--workdir

## **docker-compose** **stop**

* Stops running containers without removing them. They can be started again using docker-compose start.
* It stops the containers by sending a SIGTERM signal.
* The only deference between docker-compose stop and docker-compose kill is stop command sends SIGTERM signal while kill command sends SIGKILL signal.
* If the main process inside the container didn’t terminated after receiving SIGTERM signal. It will be killed by SIGKILL after a particular grace period.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present.*

PS > docker-compose stop [Options] [SERVICE…]

Key flags:

*#Specify a shutdown time in seconds. By default, shutdown time is set to 10sec.*

--timeout

## **docker-compose pause**

* The command suspends all the processes in a running container using SIGSTOP signal.
* Syntax

*#Change directory to the build context at which docker-compose.yml is present.*

PS > docker-compose pause [SERVICE…]

## SIGTERM

It is the termination signal. The default behavior is to terminate the process,

but it also can be caught or ignored. The intention is to kill the process,

gracefully or not, but to first allow it a chance to cleanup.

## SIGKILL

It is the kill signal. The only behavior is to kill the process,

immediately. As the process cannot catch or ignore the signal, it cannot cleanup,

and thus, this is a signal of last resort.

## SIGSTOP

It is the pause signal. The only behavior is to pause the process;

the signal cannot be caught or ignored.

The shell uses pausing (and its counterpart, resuming via SIGCONT) to implement job control.

### Reference:

https://www.memogeeks.com/2018/11/kill-stop-and-pause-docker-commands.html#summary

# **Life Cycle of docker containers.**

