Docker-Compose CLI Cheat Sheet

(for version 1.12.0)

docker-compose

Define and run multi-container applications with Docker

docker-compose [-f=<arg>...] [options] [COMMAND] [ARGS...]
docker-compose -h|--help

```
-f, --file FILE              Specify an alternate compose file
                             (default: docker-compose.yml)
-p, --project-name NAME      Specify an alternate project name
                             (default: directory name)
--verbose                    Show more output
-v, --version                Print version and exit
-H, --host HOST              Daemon socket to connect to
--tls                        Use TLS; implied by --tlsverify
--tlscacert CA_PATH          Trust certs signed only by this CA
--tlscert CLIENT_CERT_PATH   Path to TLS certificate file
--tlskey TLS_KEY_PATH        Path to TLS key file
--tlsverify                  Use TLS and verify the remote
--skip-hostname-check        Don't check the daemon's hostname against
                             the name in the client certificate (for
                             example if your docker is an IP address)
```

up

Builds, (re)creates, starts, and attaches to containers for a service.
Unless they are already running, this command also starts any linked services.

up [options] [SERVICE...]

```
-d                        Detached mode: Run containers in the
                          background, print new container names.
                          Incompatible with --abort-on-container-
                          exit
--no-color                Produce monochrome output.
--no-deps                 Don't start linked services.
--force-recreate          Recreate containers even if their
                          configuration and image haven't changed.
                          Incompatible with --no-recreate.
--no-recreate             If containers already exist, don't
                          recreate them.
                          Incompatible with --force-recreate.
--no-build                Don't build an image, even if it's
                          missing.
--build                   Build images before starting containers.
--abort-on-container-exit Stops all containers if any container was
                          stopped.
                          Incompatible with -d.
-t, --timeout TIMEOUT     Use this timeout in seconds for container
                          shutdown when attached or when containers
                          are already running. (default: 10)
--remove-orphans          Remove containers for services not
                          defined in the Compose file
```

The docker-compose up command aggregates the output of each container. When the command exits, all containers are stopped. Running docker-compose up -d starts the containers in the background and leaves them running.

If there are existing containers for a service, and the service's configuration or image was changed after the container's creation, docker-compose up picks up the changes by stopping and recreating the containers (preserving mounted volumes). To prevent Compose from picking up changes, use the --no-recreate flag.

If you want to force Compose to stop and recreate all containers, use the --force-recreate flag.

stop

Stops running containers without removing them. They can be started again with docker-compose start

stop [options] [SERVICE...]
```
-t, --timeout TIMEOUT     Specify a shutdown timeout in seconds
                          (default: 10)
```

kill

Forces running containers to stop by sending a SIGKILL signal
kill [options] [SERVICE...]
```
-s SIGNAL                 SIGNAL to send to the container. Default
                          signal is SIGKILL
```
Optionally the signal can be passed, for example:

$ docker-compose kill -s SIGINT

start

Starts existing containers for a service
start [SERVICE...]

pause

Pauses running containers of a service
pause [SERVICE...]
They can be unpaused with docker-compose unpause.

unpause

Unpauses paused containers of a service
unpause [SERVICE...]

build

Define and run multi-container applications with Docker
build [options] [SERVICE...]
```
--force-rm                Always remove intermediate containers.
--no-cache                Do not use cache when building the image.
--pull                    Always attempt to pull a newer version of
                          the image.
```
Services are built once and then tagged as project_service,
e.g., composetest_db
If you change a service's Dockerfile or the contents of its build directory, run **docker-compose build** to rebuild it.

config

Validate and view the compose file
config [options]
```
-q, --quiet               Only validate the configuration, don't
                          print anything.
--services                Print the service names, one per line.
```

create

Creates containers for a service
create [options] [SERVICE...]
```
--force-recreate          Recreate containers even if their
                          configuration and image haven't changed.
                          Incompatible with --no-recreate.
--no-recreate             If containers already exist, don't
                          recreate them.
                          Incompatible with --force-recreate.
--no-build                Don't build an image, even if it's
                          missing.
--build                   Build images before creating containers.
```

down

Stops containers and removes containers, networks, volumes, and images created by up
down [options]
```
--rmi type                Remove images. Type must be one of:
                          'all': Remove all images used by any
                          service.
                          'local': Remove only images that don't
                          have a custom tag set by the `image`
                          field.
-v, --volumes             Remove named volumes declared in the
                          `volumes` section of the Compose file and
                          anonymous volumes attached to containers.
--remove-orphans          Remove containers for services not
                          defined in theCompose file
```
By default, the only things removed are:

Containers for services defined in the Compose file
Networks defined in the networks section of the Compose file
The default network, if one is used
Networks and volumes defined as external are never removed.

events

Stream container events for every container in the project
events [options] [SERVICE...]
```
--json                    Output events as a stream of json objects
```
With the --json flag, a json object will be printed one per line with the format:

```
{
    ""service"": ""web"",
    ""event"": ""create"",
    ""container"": ""213cf75fc39a"",
    ""image"": ""alpine:edge"",
    ""time"": ""2015-11-20T18:01:03.615550"",
}
```

logs

Displays log output from services
logs [options] [SERVICE...]
```
--no-color                Produce monochrome output
-f, --follow              Follow log output
-t, --timestamps          Show timestamps
--tail                    Number of lines to show from the end of
                          the logs for each container
```

port

Prints the public port for a port binding.
port [options] SERVICE PRIVATE_PORT
```
--protocol=proto          tcp or udp [default: tcp]
--index=index             index of the container if there are
                          multiple instances of a service [default:
                          1]
```

ps

Lists containers
ps [options] [SERVICE...]
```
-q                        Only display IDs
```
Forces running containers to stop by sending a SIGKILL signal. Optionally the signal can be passed, for example:

$ docker-compose kill -s SIGINT

pull

Pulls service images
pull [options] [SERVICE...]
```
--ignore-pull-failures    Pull what it can and ignores images with
                          pull failures
```

restart

Restarts services
restart [options] [SERVICE...]
```
-t, --timeout TIMEOUT     Specify a shutdown timeout in seconds.
                          (default: 10)
```

rm

Removes stopped service containers
rm [options] [SERVICE...]
```
-f, --force               Don't ask to confirm removal
-v                        Remove any anonymous volumes attached to
                          containers
-a, --all                 Also remove one-off containers created by
                          docker-compose run
```
By default, anonymous volumes attached to containers will not be removed.
You can override this with -v. To list all volumes, use docker volume ls.

Any data which is not in a volume will be lost.

run

run [options] [-e KEY=VAL...] SERVICE [COMMAND] [ARGS...]
```
-d                        Detached mode: Run container in the
                          background, print new container name.
--name NAME               Assign a name to the container
--entrypoint CMD          Override the entrypoint of the image.
-e KEY=VAL                Set an environment variable (can be used
                          multiple times)
-u, --user=""""           Run as specified username or uid
--no-deps                 Don't start linked services.
--rm                      Remove container after run. Ignored in
                          detached mode.
-p, --publish=[]          Publish a container's port(s) to the host
--service-ports           Run command with the service's ports
                          enabled and mapped to the host.
-T                        Disable pseudo-tty allocation. By default
                          `docker-compose run` allocates a TTY.
-w, --workdir=""""        Working directory inside the container
```
Runs a one-time command against a service. For example, the following command starts the web service and runs bash as its command.

$ docker-compose run web bash

scale

Sets the number of containers to run for a service
scale [SERVICE=NUM...]
```
-s SIGNAL                 SIGNAL to send to the container. Default
                          signal is SIGKILL
```
Numbers are specified as arguments in the form service=num. For example:

$ docker-compose scale web=2 worker=3

SCHOOL OF DEVOPS