通过`docker-compose --help`从命令行运行来查看此信息。

```
[root]# docker-compose --help
Define and run multi-container applications with Docker.

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

Options:
  -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
  --log-level LEVEL           Set log level (DEBUG, INFO, WARNING, ERROR,
CRITICAL)
  --no-ansi                   Do not print ANSI control characters
  -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 specified in the client certificate
  --project-directory PATH    Specify an alternate working directory
                              (default: the path of the Compose file)
  --compatibility             If set, Compose will attempt to convert deploy
                              keys in v3 files to their non-Swarm equivalent

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
  images             List images 展示容器的container和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
  unpause             Unpause services
  up                  Create and start containers 启动所有容器后前台打印所有容器的日志， -d 后台启动
  version             Show the Docker-Compose version information
```

使用-f指定名称和一个或多个文件撰写路径

使用该-f标志指定Compose配置文件的位置，(默认: docker-compose.yml)。

指定多个Compose文件

您可以提供多个-f配置文件。当您提供多个文件时，Compose会将它们合并为一个配置。Compose按照提供文件的顺序构建配置。后续文件将覆盖并添加到其前任文件中。

例如，请考虑以下命令行：

```
$ docker-compose -f docker-compose.yml -f docker-compose.admin.yml run backup_db
```

该docker-compose.yml文件可能指定webapp服务。

```
webapp:
  image: examples/web
  ports:
    - "8000:8000"
  volumes:
    - "/data"
```

如果docker-compose.admin.yml还指定了相同的服务，则任何匹配的字段都会覆盖以前的文件。新值，添加到webapp服务配置。

```
webapp:
  build: .
```

```
environment:
  - DEBUG=1
```

使用 -f 加上 **-（破折号）作为文件名**，可以从stdin中读取配置，当stdin被使用时，所有配置中的路径均是相对于当前**工作目录**的相对路径。

该`-f`标志是可选的。如果未在命令行上提供此标志，Compose将遍历工作目录及其父目录，以查找`docker-compose.yml`和`docker-compose.override.yml`文件。您必须至少提供该`docker-compose.yml`文件。如果两个文件都存在于同一目录级别，则Compose会将这两个文件合并为一个配置。

`docker-compose.override.yml`文件中的配置是用来覆盖和添加`docker-compose.yml`文件中的值之外的值。

指定单个Compose文件的路径

通过 -f 可以制定一个不在当前目录的compose文件，要么是命令行，要么在shell或者环境配置文件（environment file）中设置一个 COMPOSE_FILE environment variable。

eg:

在命令行使用-f参数：假设你正在运行 Compose Rails sample，并且在sandbox/rails目录下有一个docker-compose.yml文件，你可以使用命令 docker-compose pull 加上 -f 在任意目录下从db服务拉取数据库镜像：

docker-compose -f ~/sandbox/rails/docker-compose.yml pull db

使用`-p`指定项目名称

每个配置都有一个项目名称。如果提供`-p`标志，则可以指定项目名称。如果未指定标志，Compose将使用当前目录名称。另请参见COMPOSE_PROJECT_NAME环境变量。

设置环境变量

您可以为各种 选项设置环境变量`docker-compose`，包括`-f`和`-p`标志。

例如，COMPOSE_FILE环境变量 与`-f`标志相关，COMPOSE_PROJECT_NAME环境变量与`-p`标志相关。

此外，您可以在环境文件中设置其中一些变量。

docker-compose up

```
Usage: up [options] [--scale SERVICE=NUM...] [SERVICE...]

Options:
    -d, --detach               Detached mode: Run containers in the background,
                               print new container names. Incompatible with
                               --abort-on-container-exit.
    --no-color                 Produce monochrome output.
    --quiet-pull               Pull without printing progress information
    --no-deps                  Don't start linked services.
    --force-recreate           Recreate containers even if their configuration
                               and image haven't changed.
    --always-recreate-deps     Recreate dependent containers.
                               Incompatible with --no-recreate.
    --no-recreate              If containers already exist, don't recreate
                               them. Incompatible with --force-recreate and -V.
    --no-build                 Don't build an image, even if it's missing.
    --no-start                 Don't start the services after creating them.
    --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)
    -V, --renew-anon-volumes   Recreate anonymous volumes instead of retrieving
                               data from the previous containers.
    --remove-orphans           Remove containers for services not defined
                               in the Compose file.
    --exit-code-from SERVICE   Return the exit code of the selected service
                               container. Implies --abort-on-container-exit.
    --scale SERVICE=NUM        Scale SERVICE to NUM instances. Overrides the
                               `scale` setting in the Compose file if present.
```

构建，（重新）创建，启动和附加到服务的容器。

除非它们已在运行，否则此命令也会启动任何链接服务。

该`docker-compose up`命令聚合每个容器的输出（基本上正在运行`docker-compose logs -f`）。命令退出时，所有容器都将停止。运行`docker-compose up -d`在后台启动容器并使其运行。

如果服务存在现有容器，并且在创建容器后更改了服务的配置或映像，则`docker-compose up`通过停止并重新创建容器（保留已安装的卷）来获取更改。要防止Compose获取更改，请使用该`--no-recreate`标志。

如果要强制Compose停止并重新创建所有容器，请使用该`--force-recreate`标志。

如果进程遇到错误，则此命令的退出代码为`1`。
如果使用`SIGINT`（`ctrl`+ `C`）或者中断进程`SIGTERM`，则停止容器，退出代码为`0`。
如果`SIGINT`还是`SIGTERM`在这段停机阶段再次发送，运行容器被杀害，并退出代码`2`。

来源： https://docs.docker.com/compose/reference/up/

docker-compose down

```
Usage: down [options]

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 the
                            Compose file
    -t, --timeout TIMEOUT   Specify a shutdown timeout in seconds.
                            (default: 10)
```

Stops containers and removes containers, networks, volumes, and images created by `up`.

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.

docker-compose stop

```
Usage: stop [options] [SERVICE...]

Options:
-t, --timeout TIMEOUT      Specify a shutdown timeout in seconds (default: 10).
```

停止但不删除. They can be started again with `docker-compose start`.

docker-compose restart

```
Usage: restart [options] [SERVICE...]

Options:
-t, --timeout TIMEOUT      Specify a shutdown timeout in seconds. (default: 10)
```

重新启动所有已停止和正在运行的服

如果更改了`docker-compose.yml`配置，则在运行此命令后不会反映这些更改。

例如，对环境变量（在构建容器之后添加，但在执行容器命令之前添加）的更改在重新启动后不会更新。

如果您正在寻找配置服务的重新启动策略，请参考 重启在撰写文件v3和 重启在撰写V2。请注意，如果要在群集模式下

部署堆栈，则应使用restart_policy。

docker-compose exec

docker-compose exec service_name bash :service_name是compose.yaml中的服务名，不是容器名

```
Usage: exec [options] [-e KEY=VAL...] SERVICE COMMAND [ARGS...]

Options:
    -d, --detach      Detached mode: Run command in the background.
    --privileged      Give extended privileges to the process.
    -u, --user USER   Run the command as this user.
    -T                Disable pseudo-tty allocation. By default `docker-compose exec`
                      allocates a TTY.
    --index=index     index of the container if there are multiple
                      instances of a service [default: 1]
    -e, --env KEY=VAL Set environment variables (can be used multiple times,
                      not supported in API < 1.25)
    -w, --workdir DIR Path to workdir directory for this command.
```

这相当于docker exec。使用此子命令，您可以在服务中运行任意命令。默认情况下，命令分配TTY，因此您可以使用命令docker-compose exec web sh来获取交互式提示。

docker-compose logs

```
Usage: logs [options] [SERVICE...]

Options:
--no-color          Produce monochrome output.
-f, --follow        Follow log output
-t, --timestamps    Show timestamps
--tail="all"        Number of lines to show from the end of the logs
                    for each container.
```

Displays log output from services.

```
docker-compose logs -f ： 显示conpose所有服务的日志
docker-compose logs -f SERVICE_NAME ：  显示SERVICE_NAME的日志
```

docker-compose rm

```
Usage: rm [options] [SERVICE...]

Options:
    -f, --force   Don't ask to confirm removal
    -s, --stop    Stop the containers, if required, before removing
    -v            Remove any anonymous volumes attached to containers
```

删除已停止的服务容器。

默认情况下，不会删除附加到容器的匿名卷。你可以用它来覆盖它`-v`。要列出所有卷，请使用`docker volume ls`。

任何不在卷中的数据都将丢失。

运行没有选项的命令也会删除由`docker-compose up`或`docker-compose run`创建的一次性容器：

```
$ docker-compose rm
Going to remove djangoquickstart_web_run_1
Are you sure? [yN] y
Removing djangoquickstart_web_run_1 ... done
```

docker-compose pause

```
Usage: pause [SERVICE...]
```

Pauses running containers of a service. They can be unpaused with `docker-compose unpause`.

docker-compose port

```
Usage: port [options] SERVICE PRIVATE_PORT

Options:
--protocol=proto  tcp or udp [default: tcp]
--index=index     index of the container if there are multiple
                  instances of a service [default: 1]
```

Prints the public port for a port binding.

```
# docker-compose port mysql 3066

# docker-compose port mysql 3306
0.0.0.0:3396
```

docker-compose ps

```
Usage: ps [options] [SERVICE...]

Options:
-q    Only display IDs
```

Lists containers.

```
$ docker-compose ps
         Name                       Command                  State             Ports
-------------------------------------------------------------------------------------------
mywordpress_db_1          docker-entrypoint.sh mysqld      Up (healthy)   3306/tcp
mywordpress_wordpress_1   /entrypoint.sh apache2-for ...   Restarting     0.0.0.0:8000->80/tcp
```

docker-compose pull

```
Usage: pull [options] [SERVICE...]

Options:
    --ignore-pull-failures  Pull what it can and ignores images with pull failures.
    --parallel              Deprecated, pull multiple images in parallel (enabled by default).
    --no-parallel           Disable parallel pulling.
    -q, --quiet             Pull without printing progress information
    --include-deps          Also pull services declared as dependencies
```

拉取与`docker-compose.yml`或`docker-stack.yml`文件中定义的服务关联的图像，但不会根据这些图像启动容器。

例如，假设您有Quickstart：Compose和Rails示例中的此`docker-compose.yml`文件。

```
version: '2'
services:
  db:
    image: postgres
  web:
    build: .
    command: bundle exec rails s -p 3000 -b '0.0.0.0'
    volumes:
      - .:/myapp
    ports:
      - "3000:3000"
    depends_on:
      - db
```

如果您`docker-compose pull ServiceName`在与`docker-compose.yml`定义服务的文件相同的目录中运行，则Docker会提取关联的图像。例如，要在我们的示例中调用`postgres`配置为`db`服务的映像，您将运行`docker-compose pull db`。

```
$ docker-compose pull db
Pulling db (postgres:latest)...
latest: Pulling from library/postgres
cd0a524342ef: Pull complete
...
Digest: sha256:fd6c0e2a9d053bebb294bb13765b3e01be7817bf77b01d58c2377ff27a4a46dc
```

```
Status: Downloaded newer image for postgres:latest
```

docker-compose push

```
Usage: push [options] [SERVICE...]

Options:
    --ignore-push-failures  Push what it can and ignores images with push failures.
```

将服务图像推送到各自的服务`registry/repository`。

假设：

- 您正在推送您在本地构建的图像
- 您可以访问构建密钥

例

```yaml
version: '3'
services:
  service1:
    build: .
    image: localhost:5000/yourimage  # goes to local registry

  service2:
    build: .
    image: youruser/yourimage  # goes to youruser DockerHub registry
```

docker-compose run

```
Usage:
    run [options] [-v VOLUME...] [-p PORT...] [-e KEY=VAL...] [-l KEY=VALUE...]
        SERVICE [COMMAND] [ARGS...]

Options:
    -d, --detach          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)
    -l, --label KEY=VAL   Add or override a label (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.
    --use-aliases       Use the service's network aliases in the network(s) the
                        container connects to.
    -v, --volume=[]     Bind mount a volume (default [])
    -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
```

Commands you use with `run` start in new containers with configuration defined by that of the service, including volumes, links, and other details. However, there are two important differences.

First, the command passed by `run` overrides the command defined in the service configuration. For example, if the `web` service configuration is started with `bash`, then `docker-compose run web python app.py` overrides it with `python app.py`.

The second difference is that the `docker-compose run` command does not create any of the ports specified in the service configuration. This prevents port collisions with already-open ports. If you *do want* the service's ports to be created and mapped to the host, specify the `--service-ports` flag:

```
docker-compose run --service-ports web python manage.py shell
```

Alternatively, manual port mapping can be specified with the `--publish` or `-p` options, just as when using `docker run`:

```
docker-compose run --publish 8080:80 -p 2022:22 -p 127.0.0.1:2021:21 web python manage.py shell
```

If you start a service configured with links, the `run` command first checks to see if the linked service is running and starts the service if it is stopped. Once all the linked services are running, the `run` executes the command you passed it. For example, you could run:

```
docker-compose run db psql -h db -U docker
```

This opens an interactive PostgreSQL shell for the linked `db` container.

If you do not want the `run` command to start linked containers, use the `--no-deps` flag:

```
docker-compose run --no-deps web python manage.py shell
```

If you want to remove the container after running while overriding the container's restart policy, use the `--rm` flag:

```
docker-compose run --rm web python manage.py db upgrade
```

This runs a database upgrade script, and removes the container when finished running, even if a restart policy is specified in the service configuration.

docker-compose top

```
Usage: top [SERVICE...]
```

Displays the running processes.

```
$ docker-compose top
compose_service_a_1
PID    USER   TIME   COMMAND
```

```
--------------------------
4060   root   0:00   top

compose_service_b_1
PID    USER   TIME   COMMAND
--------------------------
4115   root   0:00   top
```

来源： https://docs.docker.com/compose/reference/top/