

Docker Commands

Traditional (pre-2017) Command Usage

Docker Container Commands

<code>docker create [IMG]</code>	Create a container (without starting it)
<code>docker rename [ORIG] [NEW]</code>	Rename an existing container
<code>docker run [IMG] [CMD]</code>	Run a command in a new container
<code>docker run --rm [IMG]</code>	Removes a container after it exits.
<code>docker run -td [IMG]</code>	Starts a container and keeps it running.
<code>docker run -it [IMG]</code>	Starts container, allocates a pseudo-TTY to stdin, start shell
<code>docker run -it-rm [IMG]</code>	Create, start, runs command, when done remove container
<code>docker rm [CONT]</code>	Delete a container (if it is not running)
<code>docker update [CONT]</code>	Update the configuration of one or more containers

Starting and Stopping Containers

<code>docker start stop restart pause unpause [CONT]</code>	
<code>docker wait [CONT]</code>	Block a container until others stop
<code>docker kill [CONT]</code>	Sends a SIGKILL to a running container
<code>docker attach [CONT]</code>	Attach local stdin stdout stderr streams to running container

Docker Image Commands

<code>docker build [URL]</code>	Create an image from a Dockerfile, add a -t to tag the image
<code>docker pull push [IMG]</code>	Pull (or push) an image from (or to) a registry
<code>docker import [URL/FILE]</code>	Create an image from a tarball
<code>docker commit [CONT] [NEW_IMG]</code>	Create an image from a container
<code>docker rm [IMG]</code>	Remove an image
<code>docker load [TAR_FILE/STDIN_FILE]</code>	Load an image from a tar archive or stdin
<code>docker save [IMG] > [TAR_FILE]</code>	Save image to tar archive via STDOUT with all parent layers, etc

Docker Commands for Container and Image Information

<code>docker ps -a</code>	lists containers both running and stopped, status, open ports
<code>docker logs [CONT]</code>	List the logs from a running container
<code>docker inspect [OBJ_NAME/ID]</code>	List low-level information on Docker objects
<code>docker events [CONT]</code>	List real-time events from a container:
<code>docker port [CONT]</code>	Show port (or specific) mapping for a container
<code>docker top [CONT]</code>	Yes, runs top.
<code>docker stats [CONT]</code>	Live resource usage statistics of containers
<code>docker diff [CONT]</code>	yes, runs a diff
<code>docker image ls</code>	List all locally stored images
<code>docker history [IMG]</code>	

Networks

<code>docker network ls</code>	List networks
<code>docker network rm [NETWK]</code>	Remove one or more networks
<code>docker network inspect [NETWK]</code>	Show info on one or more networks
<code>docker network connect disconnect [NETWK] [CONT]</code>	Connect/ disconnect a container from a network

Jan 19 2017, Docker Blog:

CLI restructured

Docker has grown many features over the past couple years and the Docker CLI now has a lot of commands (40 at the time of writing). Some, like build or run are used a lot, some are more obscure, like pause or history. The many top-level commands clutters help pages and makes tab-completion harder.

In Docker 1.13, we regrouped every command to sit under the logical object it's interacting with. For example list and startof containers are now subcommands of docker container and history is a subcommand of docker image.

docker container list

docker container start

docker image history

These changes let us clean up the Docker CLI syntax, improve help text and make Docker simpler to use. The old command syntax is still supported, but we encourage everybody to adopt the new syntax.

Base Docker Management Commands

app*	Docker Application
builder	Manage builds
buildx	Extended build capabilities with BuildKit
config	Manage Docker configs
container	Manage containers
context	Manage contexts
image	Manage images
manifest	Manage Docker image manifests and manifest lists
network	Manage networks
node	Manage Swarm nodes
plugin	Manage plugins
scan*	
secret	Manage Docker secrets
service	Manage services
stack	Manage Docker stacks
swarm	Manage swarm
system	Manage Docker
trust	Manage trust on Docker images
volume	Manage volumes

Deprecated?

assemble*	Framework-aware builds (Docker Enterprise)
cluster	Manage Docker clusters (Docker Enterprise)
engine	Manage the docker Engine
registry*	Manage Docker registries
template*	Quickly scaffold services (Docker Enterprise)

docker app [OPTIONS] COMMAND – A tool to build, share and run a Docker App

----- Options:

--installer-context string Context on which the installer image is ran (default "default")

--version Print version information

----- Management Commands:

image Manage App images

Commands:

build Build an App image from an App definition (.dockerapp)

init	Initialize an App definition
inspect	Shows status, metadata, parameters and the list of services of a running App
ls	List running Apps
pull	Pull an App image from a registry
push	Push an App image to a registry
rm	Remove a running App
run	Run an App from an App image
update	Update a running App

docker builder COMMAND – Manage builds

build	Build an image from a Dockerfile
prune	Remove build cache

docker buildx [OPTIONS] COMMAND – Extended build capabilities with BuildKit

----- Options:

--builder string Override the configured builder instance

----- Management Commands:

imagetools Commands to work on images in registry

Commands:

bake	Build from a file
build	Start a build
create	Create a new builder instance
du	Disk usage
inspect	Inspect current builder instance
ls	List builder instances
prune	Remove build cache
rm	Remove a builder instance
stop	Stop builder instance
use	Set the current builder instance
version	Show buildx version information

docker config COMMAND – Manage Docker configs

create	Create a config from a file or STDIN
inspect	Display detailed information on one or more configs
ls	List configs
rm	Remove one or more configs

docker container COMMAND – Manage containers

attach	Attach local standard input, output, and error streams to a running container
commit	Create a new image from a container's changes
cp	Copy files/folders between a container and the local filesystem
create	Create a new container
diff	Inspect changes to files or directories on a container's filesystem
exec	Run a command in a running container
export	Export a container's filesystem as a tar archive
inspect	Display detailed information on one or more containers
kill	Kill one or more running containers
logs	Fetch the logs of a container
ls	List containers
pause	Pause all processes within one or more containers
port	List port mappings or a specific mapping for the container

prune	Remove all stopped containers
rename	Rename a container
restart	Restart one or more containers
rm	Remove one or more containers
run	Run a command in a new container
start	Start one or more stopped containers
stats	Display a live stream of container(s) resource usage statistics
stop	Stop one or more running containers
top	Display the running processes of a container
unpause	Unpause all processes within one or more containers
update	Update configuration of one or more containers
wait	Block until one or more containers stop, then print their exit codes

docker context COMMAND – Manage contexts

create	Create a context
export	Export a context to a tar or kubeconfig file
import	Import a context from a tar or zip file
inspect	Display detailed information on one or more contexts
ls	List contexts
rm	Remove one or more contexts
update	Update a context
use	Set the current docker context

docker image COMMAND – Manage images

build	Build an image from a Dockerfile
history	Show the history of an image
import	Import the contents from a tarball to create a filesystem image
inspect	Display detailed information on one or more images
load	Load an image from a tar archive or STDIN
ls	List images
prune	Remove unused images
pull	Pull an image or a repository from a registry
push	Push an image or a repository to a registry
rm	Remove one or more images
save	Save one or more images to a tar archive (streamed to STDOUT by default)
tag	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

docker manifest COMMAND – Managing image manifests and manifest lists.

A manifest list allows you to use one name to refer to the same image built for multiple architectures.

*** EXPERIMENTAL ***

annotate	Add additional information to a local image manifest
create	Create a local manifest list for annotating and pushing to a registry
inspect	Display an image manifest, or manifest list
push	Push a manifest list to a repository
rm	Delete one or more manifest lists from local storage

docker network COMMAND – Manage networks

connect	Connect a container to a network
create	Create a network
disconnect	Disconnect a container from a network
inspect	Display detailed information on one or more networks

ls	List networks
prune	Remove all unused networks
rm	Remove one or more networks

docker node COMMAND – Manage Swarm nodes

demote	Demote one or more nodes from manager in the swarm
inspect	Display detailed information on one or more nodes
ls	List nodes in the swarm
promote	Promote one or more nodes to manager in the swarm
ps	List tasks running on one or more nodes, defaults to current node
rm	Remove one or more nodes from the swarm
update	Update a node

docker plugin COMMAND – Manage plugins

create	Create a plugin from a rootfs and configuration. Plugin data directory must contain config.json and rootfs directory.
disable	Disable a plugin
enable	Enable a plugin
inspect	Display detailed information on one or more plugins
install	Install a plugin
ls	List plugins
push	Push a plugin to a registry
rm	Remove one or more plugins
set	Change settings for a plugin
upgrade	Upgrade an existing plugin

docker scan [OPTIONS] IMAGE – A tool to scan your images

----- Options:

--accept-license	Accept using a third party scanning provider
--dependency-tree	Show dependency tree with scan results
--exclude-base	Exclude base image from vulnerability scanning (requires --file)
-f, --file string	Dockerfile associated with image, provides more detailed results
--group-issues	Aggregate duplicated vulnerabilities and group them to a single one (requires --json)
--json	Output results in JSON format
--login	Authenticate to scan provider using optional token (with --token), web base token if empty
--reject-license	Reject using a third party scanning provider
--severity string	Only report vulnerabilities of provided level or higher (low medium high)
--token string	Authentication token to login to the third party scanning provider
--version	Display version of the scan plugin

docker secret COMMAND – Manage Docker secrets

create	Create a secret from a file or STDIN as content
inspect	Display detailed information on one or more secrets
ls	List secrets
rm	Remove one or more secrets

docker service COMMAND – Manage services

create	Create a new service
inspect	Display detailed information on one or more services
logs	Fetch the logs of a service or task
ls	List services

ps	List the tasks of one or more services
rm	Remove one or more services
rollback	Revert changes to a service's configuration
scale	Scale one or multiple replicated services
update	Update a service

docker stack [OPTIONS] COMMAND – Manage Docker stacks

----- Options:

--orchestrator string Orchestrator to use (swarm|kubernetes|all)

----- Commands:

deploy	Deploy a new stack or update an existing stack
ls	List stacks
ps	List the tasks in the stack
rm	Remove one or more stacks
services	List the services in the stack

docker swarm COMMAND – Manage Swarm

ca	Display and rotate the root CA
init	Initialize a swarm
join	Join a swarm as a node and/or manager
join-token	Manage join tokens
leave	Leave the swarm
unlock	Unlock swarm
unlock-key	Manage the unlock key
update	Update the swarm

docker system COMMAND – Manage Docker

df	Show docker disk usage
events	Get real time events from the server
info	Display system-wide information
prune	Remove unused data

docker trust COMMAND – Manage trust on Docker images

----- Management Commands:

key	Manage keys for signing Docker images
signer	Manage entities who can sign Docker images

----- Commands:

inspect	Return low-level information about keys and signatures
revoke	Remove trust for an image
sign	Sign an image

docker volume COMMAND – Manage volumes

create	Create a volume
inspect	Display detailed information on one or more volumes
ls	List volumes
prune	Remove all unused local volumes
rm	Remove one or more volumes