

```
C:\Users\welcome>docker help
```

```
Usage:  docker [OPTIONS] COMMAND
```

A self-sufficient runtime for containers

Options:

<code>--config</code> string	Location of client config files (default "C:\\Users\\welcome\\.docker")
<code>-c, --context</code> string	Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
<code>-D, --debug</code>	Enable debug mode
<code>-H, --host</code> list	Daemon socket(s) to connect to
<code>-l, --log-level</code> string	Set the logging level ("debug" "info" "warn" "error" "fatal") (default "info")
<code>--tls</code>	Use TLS; implied by <code>--tlsverify</code>
<code>--tlscacert</code> string	Trust certs signed only by this CA (default "C:\\Users\\welcome\\.docker\\ca.pem")
<code>--tlscert</code> string	Path to TLS certificate file (default "C:\\Users\\welcome\\.docker\\cert.pem")
<code>--tlskey</code> string	Path to TLS key file (default "C:\\Users\\welcome\\.docker\\key.pem")
<code>--tlsverify</code>	Use TLS and verify the remote
<code>-v, --version</code>	Print version information and quit

Management Commands:

builder	Manage builds
buildx*	Docker Buildx (Docker Inc., v0.9.1)
compose*	Docker Compose (Docker Inc., v2.12.2)
config	Manage Docker configs
container	Manage containers
context	Manage contexts
dev*	Docker Dev Environments (Docker Inc., v0.0.3)
extension*	Manages Docker extensions (Docker Inc., v0.2.13)
image	Manage images
manifest	Manage Docker image manifests and manifest lists
network	Manage networks
node	Manage Swarm nodes
plugin	Manage plugins
sbom*	View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)
scan*	Docker Scan (Docker Inc., v0.21.0)
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

Commands:

attach	Attach local standard input, output, and error streams to a running container
build	Build an image from a Dockerfile
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
events	Get real time events from the server
exec	Run a command in a running container
export	Export a container's filesystem as a tar archive
history	Show the history of an image
images	List images
import	Import the contents from a tarball to create a filesystem image
info	Display system-wide information
inspect	Return low-level information on Docker objects
kill	Kill one or more running containers
load	Load an image from a tar archive or STDIN
login	Log in to a Docker registry
logout	Log out from a Docker registry
logs	Fetch the logs of a container
pause	Pause all processes within one or more containers
port	List port mappings or a specific mapping for the container
ps	List containers
pull	Pull an image or a repository from a registry
push	Push an image or a repository to a registry
rename	Rename a container
restart	Restart one or more containers
rm	Remove one or more containers
rmi	Remove one or more images
run	Run a command in a new container
save	Save one or more images to a tar archive (streamed to STDOUT by default)
search	Search the Docker Hub for images
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
tag	Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
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
version	Show the Docker version information
wait	Block until one or more containers stop, then print their exit codes