

### Building Images

<code>docker build</code>	Build a new image from the source code at PATH <code>docker build -t myimg .</code>
<code>-q</code>	Suppress the output generated by containers
<code>--rm</code>	Remove intermediate containers after a successful build
<code>-t</code>	Repository name (and optionally a tag) for the image

#### Examples:

Simplest possible build instruction:

```
docker build .
```

Name image and tag as v1.5:

```
docker build -t myorg/myimg:1.5 .
```

### Misc useful commands

<code>attach</code>	Attach to a running container
<code>cp</code>	Copy files/folders from a container's filesystem to the host path
<code>create</code>	Create a new container
<code>exec</code>	Run a command in a running container
<code>images</code>	List images
<code>export</code>	Stream the contents of a container as a tar archive
<code>import</code>	Create a new filesystem image from the contents of a tarball
<code>inspect</code>	Return low-level information on a container or image

### Misc useful commands (cont)

<code>kill</code>	Kill a running container
<code>load</code>	Load an image from a tar archive
<code>logs</code>	Fetch the logs of a container
<code>port</code>	Lookup the public-facing port that is NAT-ed to PRIVATE_PORT
<code>ps</code>	List containers
<code>pull</code>	Pull an image or a repository from a Docker registry server
<code>push</code>	Push an image or a repository to a Docker registry server
<code>rm</code>	Remove one or more containers
<code>rmi</code>	Remove one or more images
<code>save</code>	Save an image to a tar archive
<code>search</code>	Search for an image on the Docker Hub
<code>start</code>	Start a stopped container
<code>stop</code>	Stop a running container
<code>top</code>	Lookup the running processes of a container
<code>wait</code>	Block until a container stops, then print its exit code

Run `docker COMMAND --help` for help on the command.

### Running Docker

<code>docker run</code>	starts a process with its own file system, its own networking, and its own isolated process tree <code>docker run -itP image</code>
<code>--name</code>	name the container <code>docker run --name=somename org</code>
<code>-t</code>	terminal interface
<code>-i</code>	interactive session
<code>-d</code>	daemon Mode
<code>-P</code>	publish all exposed ports
<code>-p</code>	expose specific port <code>-p ip:hostport:containerport</code>
<code>--rm</code>	remove intermediate images
<code>-v</code>	bind mount a volume <code>-v /host:/container</code>

#### Examples:

run an interactive session:

```
sudo docker run -P --name=somename image
```

run container in background, on port 80:

```
sudo docker run --d -v /etc/appdata:/data -p 0.0.0.0:80:8080 --rm nginx
```

### Docker Compose

TBC



By **Andrew Matthews** (aabs)  
[cheatography.com/aabs/](http://cheatography.com/aabs/)  
[aabs.wordpress.com](http://aabs.wordpress.com)

Published 15th July, 2015.

Last updated 21st August, 2015.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

### Dockerfiles (for creating images)

**FROM** set base image for this image  
FROM <image>:<tag>

**MAINTAINER** The author of the image  
MAINTAINER <name>

**RUN** execute commands in new layer  
RUN <command>  
RUN ["executable",  
"param1", ...]

**CMD** provide default for an executing container  
CMD  
["executable", "param1", ...]

**LABEL** add metadata to image  
LABEL <key>=<value>  
<key>=<value>

**EXPOSE** listen on the network ports  
EXPOSE <port> [<port>...]

**ENV** set the environment variables  
ENV <key> <value>  
ENV <key>=<value>  
<key>=<value>

**ADD/COPY** **copies** new files from <src> and adds them to the container at <dest>.  
ADD <src>... <dest>

**ENTRYPOINT** configure container to run as an executable  
ENTRYPOINT ["exe",  
"arg" ...]

### Dockerfiles (for creating images) (cont)

**VOLUME** create externally mounted volumes  
VOLUME ["/data"]

**USER** sets the user name or UID to use when running the image  
USER<user>

**WORKDIR** sets working dir for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions:  
WORKDIR /path/to/workdir

**ONBUILD** adds a trigger instruction for execution when image is used as base of another build  
ONBUILD [INSTRUCTION]

Check out the [Manual Page](#) for more detail.



By **Andrew Matthews** (aabs)  
[cheatography.com/aabs/](http://cheatography.com/aabs/)  
[aabs.wordpress.com](http://aabs.wordpress.com)

Published 15th July, 2015.  
Last updated 21st August, 2015.  
Page 2 of 2.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>