

Building Images	Misc useful commands (cont)	Dockerfiles (for creating images) (cont)
<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	<code>top</code> Lookup the running processes of a container <code>-wait</code> Block until a container stops, then print its exit code Run <code>docker COMMAND --help</code> for help on the command.	<code>EXPOSE</code> listen on the network ports <code>EXPOSE <port> [<port>]</code> <code>ENV</code> set the environment variables <code>ENV <key> <value></code> <code>ENV <key>= <value> <key>= <value></code> <code>ADD /COPY??</code> copies new files from <src> to the container at <dest> <code>ENTRYPOINT</code> configure container to run as a executable <code>ENTRYPOINT ["executable", "argument"]</code> <code>VOLUME</code> create externally mounted volumes <code>VOLUME ["/data"]</code> <code>USER</code> sets the user name or UID to use when running the image <code>USER<user></code> <code>WORKDIR</code> sets working dir for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions <code>WORKDIR /path/to/workdir</code> <code>ONBUILD</code> adds a trigger instruction for when image is used as base for build <code>ONBUILD [INSTRUCTION]</code>
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 =some name 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>	<code>name</code> the container <code>name</code> the container <code>terminal</code> interface <code>interactive</code> session <code>daemon</code> Mode <code>publish</code> all exposed ports <code>expose</code> specific port <code>ip:hostport:containerport</code> <code>remove</code> intermediate images <code>bind</code> mount a volume <code>/host:/container</code>	<code>ENTRYPOINT</code> configure container to run as a executable <code>ENTRYPOINT ["executable", "argument"]</code> <code>VOLUME</code> create externally mounted volumes <code>VOLUME ["/data"]</code> <code>USER</code> sets the user name or UID to use when running the image <code>USER<user></code> <code>WORKDIR</code> sets working dir for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions <code>WORKDIR /path/to/workdir</code> <code>ONBUILD</code> adds a trigger instruction for when image is used as base for build <code>ONBUILD [INSTRUCTION]</code>
Examples: Simplest possible build instruction: <code>docker build .</code> Name image and tag as v1.5: <code>docker build -t myorg/myimg:1.5 .</code>	Examples: run an interactive session: <code>sudo docker run -P --name =some name image</code> run container in background, on port 80: <code>sudo docker run --d -v /etc/appdata:/data -p 0.0.0.0:80:8080 --rm nginx</code>	Check out the Manual Page for more detail.
Docker Compose		
TBC		
Dockerfiles (for creating images)		

attach	Attach to a running container
cp	Copy files/folders from a container's filesystem to the host path
create	Create a new container
exec	Run a command in a running container
images	List images
export	Stream the contents of a container as a tar archive
import	Create a new filesystem image from the contents of a tarball
inspect	Return low-level information on a container or image
kill	Kill a running container
load	Load an image from a tar archive
logs	Fetch the logs of a container
port	Lookup the public-facing port that is NAT-ed to PRIVATE_PORT
ps	List containers
pull	Pull an image or a repository from a Docker registry server
push	Push an image or a repository to a Docker registry server
rm	Remove one or more containers
rmi	Remove one or more images
save	Save an image to a tar archive
search	Search for an image on the Docker Hub
start	Start a stopped container
stop	Stop a running container

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>



By **Andrew Matthews** (aabs)
cheatography.com/aabs/
aabs.wordpress.com

Published 15th July, 2015.
 Last updated 12th May, 2016.
 Page 1 of 2.

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