

Container Tools		Aliases (cont)	
<code>docker ps</code>	lists all currently running containers	<code>docker _ca che _clear</code>	runs Cache
<code>docker ps -a</code>	lists all containers (including stopped ones)		o mmand in remove cacle
<code>docker logs CONTAINER(S)</code>	shows the output of the given container(s)	<code>docker _ca che _fo rce _clear</code>	rm -rf ap
<code>docker rm CONTAINER(S)</code>	removes a container(s) - including any local data changes	<code>docker _ru n_s ing le_test TEST_F ILE _LO</code>	runs a single test file
<code>docker rm -f CONTAINER(S)</code>	same as <code>docker rm</code> except also force stops a container(s)	ATION	
<code>docker start CONTAINER(S)</code>	attempts to start a stopped container(s)	cmd	starts a new using the sc image and s interactive b session
<code>docker stop CONTAINER(S)</code>	attempts to stop a running container(s)	Terms	
<code>docker restart CONTAINER(S)</code>	stops and then starts the specified container(s)	Image	A snapshot of a lightweight filesystem in order to run specific tasks
<code>docker cp SRC DEST</code>	copies files to/from a specific container to a local location	Container	An instantiation of an environment created from a specific image
	container location format: CONTAINER :PA TH_ ON_ CONTAINER	Fileca- binet	A file mount that is shared across all containers
<code>docker exec -it CONTAINER bash</code>	starts a bash terminal session in the CONTAINER specified	Troubleshooting	
Aliases			
<code>docker _build</code>	<code>docker _co mpose</code> followed by a <code>docker - _bu ild_js</code>		
<code>docker _bu ild_ js</code>	builds front-end components using yarn ru n compile		
<code>docker _co mpos e</code>	installs dependencies as defined in <code>compos - er.lock</code> - does not update		
<code>docker _start</code>	runs all containers as defined in <code>docker -c o mpo se.yml</code> with the default command as defined by the image		
<code>dockerize</code>	logs in to aws in order to pull new images		
<code>src_sync</code>	copies source code into all running containers and makes sure container source is up to date		

```
docker ps -a
```

use to see
which
container
not starting
linking error
occur

```
docker logs CONTAINER
```

use to see
specific logs
and error
a container
trouble starting
bad exit code

```
src_sync
```

use if logs
reveal that
files cannot
be found

```
docker build --build-arg /build-args /compose
```

use if parameters/dependencies seem to be incorrect

```
docker rm CONTAINER
```

use if a file
is corrupt in
container
NOTE: DO NOT
REMOVE
data
CONTAINERS
AS THEY
WILL TAKE
LONG TIME
TO
DOWNLOAD

```
docker cache --clear /docker-cache-f - use if annotations or other  
or cache_clear cached assets seem to be corrupt/not updating
```



Docker ps columns

NAMES	Interchangeable with <code>CONTAINER ID</code> - used to reference a specific container
CONTAINER ID	Interchangeable with <code>NAMES</code> - used to reference a specific container
IMAGE	Image that the container environment is running on
COMMAND	Command(s) that the container is runs when started
CREATED	Date when container was first created
STATUS	Current status of container - will also display exit codes
PORTS	Displays external -> internal port mapping and protocol

xdebug - docker native

```
sudo ifconfig lo0 alias 10.254.25 4.254
```

```
docker exec -it webserver bash
```

```
vi /etc/php.d /x debug.ini
```

press i to be able to edit, arrows to navigate

```
set xdebug.remote_mode=env
```

```
set xdebug.remote_host=10.254.254.254
```

```
set xdebug.remote_start=1
```

to save and quit from vi:

```
press esc
```

```
type :wq
```

```
press enter
```

exit to get out of the container

run `docker restart webserver` in order to have the new config values take place

