

### Docker Daemon

<code>sudo systemctl start docker</code>	Start Docker Daemon
<code>sudo systemctl stop docker</code>	Stop Docker Daemon
<code>sudo systemctl restart docker</code>	Restart Docker Daemon
<code>sudo systemctl status docker</code>	Check if Docker Daemon is running

If you do not have systemctl, use the service command `sudo service docker [start|stop|status|restart]"`

### Docker Common Command Line

<code>docker version</code>	Show docker client and server version
<code>docker info</code>	Show docker server information
<code>docker images</code>	List of installed docker images
<code>docker stats</code>	Resource Utilication Monitoring
<code>docker ps [-a]</code>	List of containers
<code>docker container ls</code>	Option -a: all container

### Running container

`nvidia-docker run [OPTIONS] IMAGE[:TAG] [COMMAND] [ARG...]` Run container from a given image

#### Options

<code>--name</code>	container name
<code>-i</code>	Interactive session
<code>-t</code>	terminal TTY
<code>--rm</code>	remove when exited
<code>-p H:C</code>	Exposes the port C of the container to the port H of the host
<code>-P</code>	expose all port
<code>-v H:C</code>	Mount the host path H at the path C in the container
<code>-w</code>	set working directory in container
<code>-m</code>	memory limit
<code>--cpuset-cpus</code>	limit cpus to run
<code>--add-host</code>	custom host:ip setting
<code>--privileged</code>	open kernal functions

#### Enables GPU selection (with NV\_GPU option)

`NV_GPU=1,3 nvidia-docker run --rm nvidia/caffe nvidia-smi`

#### Multi GPU Selection

Select with GPU Topology with NVLINK

`nvidia-docker run --rm nvidia/cuda:8.0-cudnn6-devel-ubuntu16.04 nvidia-smi topo -m`



### Managing Container

<code>docker logs {CONTAINERNAME}</code>	Get the logs of the container
<code>docker kill {CONTAINERNAME}</code>	Kill the container
<code>docker inspect {CONTAINERNAME}</code>	Inspect the details of a container
<code>docker start {CONTAINERNAME}</code>	start container
<code>docker stop {CONTAINERNAME}</code>	stop container
<code>docker rm [-f] {CONTAINERNAME}</code>	Remove the container Option: -f: Force removal of the container
<code>docker rm \$(docker ps -a -f status=exited -q)</code>	Removing all exited containers
<code>docker rm `docker ps - a -q`</code>	Removing all stopped containers

### Managing Image

<code>docker search {IMAGENAME}</code>	Search image
<code>docker pull {IMAGENAME}</code>	Pull an image
<code>docker rmi {IMAGENAME}</code>	Remove image
<code>docker rmi \$(docker images -a -q)</code>	Remove all images
<code>docker stop \$(docker ps -a -q) &amp;&amp; docker rm \$(docker ps -a -q)</code>	Stop and Remove all image

### NVIDIA Docker Repository

<code>docker login nvcr.io</code>	Login to a Repository Username: \$oauthtoken Password: {API Token}
<code>docker logout nvcr.io</code>	Logout

