Add the Docker Engine repositoryâ€™s key and address to aptâ€™s repository index:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add - && sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Update package index and install the Docker engine:

```
sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Run hello-world Docker image:

```
sudo docker run hello-world
```

Install recommended drivers for your GPU:

```
sudo ubuntu-drivers autoinstall
```

Launch NVIDIA system management interface:

```
nvidia-smi
```

Add NVIDIA Container Toolkit key and address to apt:

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
```

Install NVIDIA Container Toolkit:

```
sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit
```

Restart Docker:

```
sudo systemctl restart docker
```

Run the NVIDIA system management interface inside of a CUDA Docker container:

```
docker run --gpus all --rm nvidia/cuda nvidia-smi
```

Create my_jupyter_notebooks directory on host machine:

```
mkdir my_jupyter_notebooks
```

Run a Docker container that has access to tensorflow-gpu for Python 3, Jupyter Notebook, uses the host machineâ€™s GPUs, and exposes the containerâ€™s 8888 port on the hostâ€™s 8889 port (change USER for your username!):

```
sudo docker run --gpus all -it -p 8889:8888 -v /home/USER/my_jupyter_notebooks:/tf tensorflow/tensorflow:latest-gpu-py3-jupyter
```

Forward the remote machineâ€™s 8888 port to the local machineâ€™s 8889 port (this step is unnecessary if the machine running Docker is local; this command should be run on your local machine; replace USER@REMOTEMACHINE.CO.UK with your remote username and the remote machineâ€™s address):

```
ssh -N -f -L localhost:8888:localhost:8889 USER@REMOTEMACHINE.CO.UK
```

List running Docker containers:

```
sudo docker container ls
```

Stop Docker container (replace CONTAINERID with ID of container to be stopped):

```
sudo docker stop CONTAINERID
```