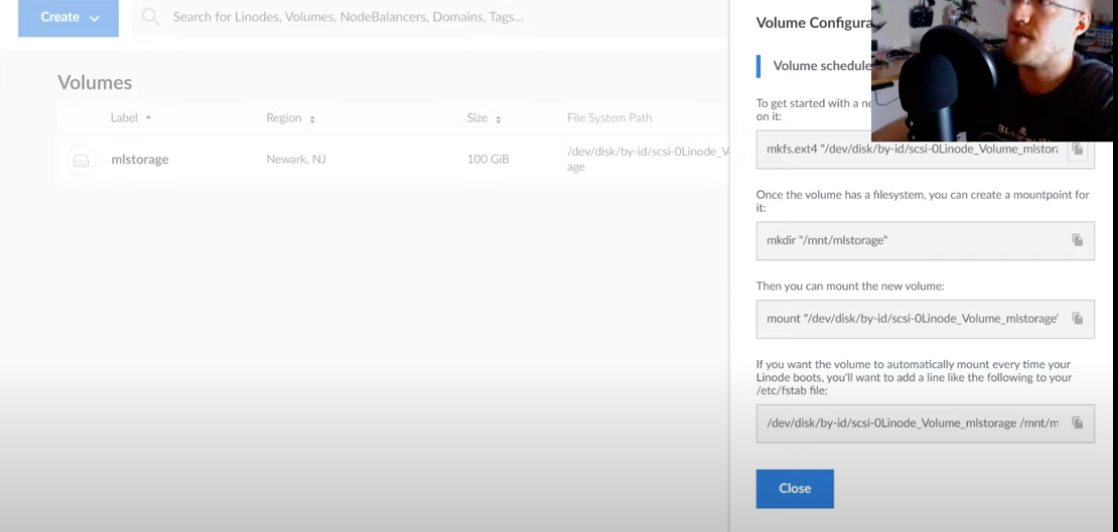
**How to set up cloud GPU’s with Sentdex**

Text tutorial: <https://pythonprogramming.net/cloud-gpu-compare-and-setup-linode-rtx-6000/>

1. Create account on Linode
2. Create Linode
   1. Ubuntu 18.04 LTS
   2. Region Newark, NJ
   3. Linode Plan: Standard Linode 2GB for 10$ per month
   4. Linode Label: data-server
   5. Root Password: \*\*\*\*
3. Copy IP Adress… point 6!
4. Get Storage but only if you need more than 50gb of storage. If not, than you don’t have to do the following steps
5. Volumes 🡪 Add a volume 🡪 mlstorage 🡪 Newark, NJ
6. Ssh root@ip-address-copy
   1. Enter Password
7. First we have to create a file system on the volume 🡪 Go to volume and copy paste the first command – Top right command – but only one time per volume!!!



1. Second command: mkdir
2. Third command: Mount that volume to that directory
3. Nano /etc/fstab
4. Cd /mnt/mlstorage
5. Go to cats vs dogs and go to download link 🡪 copy link address at the download button (<https://download.microsoft.com/download/3/E/1/3E1C3F21-ECDB-4869-8368-6DEBA77B919F/kagglecatsanddogs_5340.zip> )
6. Wget link
7. Sudo apt-get install unzip
8. Unzip Kaggle-cats-dogs
9. Go to linode website: StackScripts 🡪 Community StackScripts 🡪 search for sentdex 🡪 Sentdex TensorFlow-GPU and PyTorch Setup. (If you are on another provider, create a shell script: <https://www.cyberciti.biz/faq/run-execute-sh-shell-script/#:~:text=The%20.sh%20file%20is%20nothing,and%20UNIX%20like%20operating%20systems>. ))
   1. This can be done on any provider – this is a shell script. Here on linode this is saved and can be used but on another provider, you would create an .sh file and do this:
      1. Chmod +x name\_of\_the\_Scrpt
      2. ./script.sh
   2. Go to top right corner and click: Deploy new linode from the stack script
      1. Select Ubuntu 18.04
      2. Region
      3. Linode Plan 🡪 GPU 🡪 Dedicated 32GB + RTX6000 GPU x1 🡪 Linode Label: rtx6000gpu 🡪 Password 🡪 Create (if there is an error: You are not authorized to take this action, than they mean that there are not GPUs in that location)
      4. Now it takes a little since everything must be downloaded etc. It must be rebooted and that its ready, it should take around 10 minutes
10. Go to Linodes and copy the ip-address from the GPU
    1. Ssh root@ip-address
    2. Password
    3. Python3.7 🡪 import tensorflow as tf 🡪 If no error, everything worked 🡪 quit()
11. Mkdir /deeplearning/
12. Cd /deeplearning/
13. Go to the different terminal window where we are looged in to the data server 🡪 Now we want to move our pet images to the gpu server
    1. Slow option: Scp -r PetImages/ root@gpu-ip-address:/deeplearning/
    2. Fast option: Scp zip\_file root@gpu-ip-address:/deeplearning/
       1. Go to GPU terminal: Sudo apt-get install unzip 🡪 unzip zip\_file
14. Nano mlexample.py 🡪 Copy paste the python code in there 🡪 Save
15. Python3.7 mlexample.py 🡪 super faast!
16. Shut down all the servers with: Sudo shutdown -h now
17. Go to Volumes 🡪 mlstorage 🡪 three dots 🡪 Detach (Currently, mlstorage has all the training data on it) 🡪 Three dots 🡪 Attach 🡪 rtx6000gpu (don’t run the first command again) 🡪 show configuration and don’t run the first command 🡪 Run the second command with mkdir
18. Reconnect to the GPU server 🡪 ssh root@gpu-ip-address 🡪 second and third command
19. Cd /mnt/mlstorage 🡪 train the model 🡪 save the model 🡪 And then again detach on the website
20. At the end: Turn off all linodes with the three dots 🡪 Power Off 🡪 Nuke

Google Colab with T4 GPU

24946

24946

Type of X: <class 'numpy.ndarray'>

Type of Y: <class 'numpy.ndarray'>

Epoch 1/10

273/273 [==============================] - 14s 11ms/step - loss: 0.6628 - accuracy: 0.5913 - val\_loss: 0.6142 - val\_accuracy: 0.6676

Epoch 2/10

273/273 [==============================] - 3s 10ms/step - loss: 0.5815 - accuracy: 0.6977 - val\_loss: 0.5877 - val\_accuracy: 0.6876

Epoch 3/10

273/273 [==============================] - 3s 10ms/step - loss: 0.5118 - accuracy: 0.7499 - val\_loss: 0.4912 - val\_accuracy: 0.7604

Epoch 4/10

273/273 [==============================] - 2s 9ms/step - loss: 0.4642 - accuracy: 0.7769 - val\_loss: 0.4773 - val\_accuracy: 0.7704

Epoch 5/10

273/273 [==============================] - 2s 9ms/step - loss: 0.4103 - accuracy: 0.8127 - val\_loss: 0.4660 - val\_accuracy: 0.7886

Epoch 6/10

273/273 [==============================] - 2s 9ms/step - loss: 0.3597 - accuracy: 0.8351 - val\_loss: 0.5083 - val\_accuracy: 0.7656

Epoch 7/10

273/273 [==============================] - 3s 10ms/step - loss: 0.3059 - accuracy: 0.8635 - val\_loss: 0.5109 - val\_accuracy: 0.7706

Epoch 8/10

273/273 [==============================] - 3s 10ms/step - loss: 0.2642 - accuracy: 0.8853 - val\_loss: 0.5523 - val\_accuracy: 0.7710

Epoch 9/10

273/273 [==============================] - 2s 8ms/step - loss: 0.2078 - accuracy: 0.9126 - val\_loss: 0.5892 - val\_accuracy: 0.7676

Epoch 10/10

273/273 [==============================] - 2s 8ms/step - loss: 0.1692 - accuracy: 0.9323 - val\_loss: 0.6758 - val\_accuracy: 0.7622

<keras.src.callbacks.History at 0x7ac8cb95fac0>

Google Colab Preise: <https://colab.research.google.com/signup>

Azure Preise: <https://azure.microsoft.com/de-de/pricing/details/machine-learning/#pricing>