

The Usage of Sever and Setting up (Linux Ubuntu)

By Shuyue Jia

首先，注册DBChain账号，记住**私匙**，并绑定**邮箱**。

- ▶ 首先，介绍一下DBChain。这是一家将深度学习服务器租用与区块链技术结合的公司，主要使用区块链BDC(类似于比特币或者以太坊)来给租用的服务器付款，所用的服务器均为矿卡，有时会crash，但是租用1080ti或者2080ti便宜，很多国内学生都在租用。
- ▶ 注册账号会有“**钱包地址**”和“**私匙**”，二者对应。
- ▶ 私匙会
- ▶ 账号&钱包地址通过“**私匙**”访问，忘记“私匙”就凉了。。。
- ▶ 所有的租用服务器信息、退租或者续租的消息都是通过**邮箱**通知的。

我的钱包

我的机器

机器列表

钱包地址: **ANLFrSNHJbQqPXzHsAeuk9Z6jMZQ45GWM**

请保管好私钥信息, 私钥丢失, 任何人都将无法帮你找回DBC资产!

DBC数量: 0 [如何购买DBC](#)

等值美元: 0.000

[查看dbc详细信息: 进入了解](#)

转账			DBC
目标地址			
数量			

转账记录Hash值	源地址	目标地址	数量
-----------	-----	------	----

接下来，按照需求用支付宝往钱包里充值DBC以便租用服务器

按照需求充值DBC后然后就可以付款租用服务器啦！

服务器有不同的种类，国内学生主要是用GTX1080ti或者RTX2080ti，主要是因为速度快。

服务器租用可以续租、也可以退租。

写信
收信
通讯录

收件箱
星标邮件 ★
群邮件
草稿箱
已发送
已删除 [清空]
垃圾箱(3) [清空]
QQ邮件订阅

其他邮箱
日历 | 记事本
在线文档 NEW
附件收藏
文件中转站
简历
贺卡 | 明信片
阅读空间

< 返回 回复 回复全部 转发 删除 彻底删除 举报 拒收 标记为... 移动到... 上一封 下一封

Email from dbchain ★

发件人: dbchain service <service@info.deepbrainchain.org>
时间: 2019年10月26日(星期六) 中午11:18
收件人: Shuyue Jia <645617624@qq.com>

标记: 已将此邮件标记为星标邮件。取消星标

这不是腾讯公司的官方邮件。为了保护邮箱安全,内容中的图片未被显示。显示图片 信任此发件人的图片

网站安全云检测

恭喜机器租用成功,租用时长为: 48.0

您租用的机器id是: 2gfpp3MAB3xF5ApHCF9CkgjgZ7idV4YfnY3gpvmYmLT

您本次订单的id是: WnwwQoH4T15BUFXkXW63RfdSR75PNiyLESfxby2mA7K2LBMfk

请花费几分钟时间阅读常见问题汇总,会帮助您大幅提升机器使用效率,节省DBC

机器的远程SSH登陆信息: ssh -p 52182 root@111.44.254.168 (其中-p后面的是端口号, root是用户名, @后面的是ip地址, 也称为主机地址。可以用XShell远程访问)

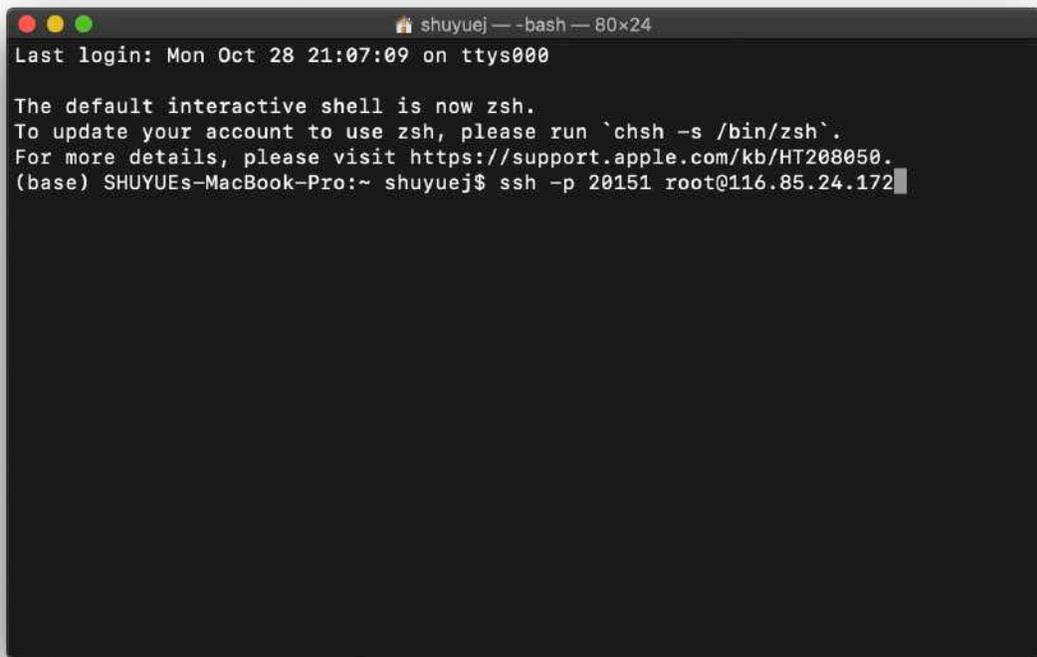
SSH初始密码: sPauvj59, 登录后尽快修改密码

jupyter登陆 url: <http://111.44.254.168:52193?token=v6bwHHWF4KeQFQ4q8A6vCq5HmD6UpQbWHDT1qyM0PCgbFNQVOh3od39uzGusxIHr> (用浏览器访问,如果无法打开,请参考常见问题汇总对jupyter重新进行设置)

注意事项:

1.DBC网络中的机器都是由矿工提供的,机器有可能随时中断,建议使用过程中定时保存训练结果到百度网盘(百度网盘的使用参看问题汇总),防止机器故障后模型丢失

接下来我们就可以使用服务器啦！

A terminal window with a black background and white text. The window title is 'shuyuej - bash - 80x24'. The text inside shows a successful SSH login to a remote server. The user 'shuyuej' is logged in as 'root' on the IP '116.85.24.172' using port '20151'. The terminal displays the last login time, a message about switching to the 'zsh' shell, and the command used to connect.

```
shuyuej — bash — 80x24
Last login: Mon Oct 28 21:07:09 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) SHUYUEs-MacBook-Pro:~ shuyuej$ ssh -p 20151 root@116.85.24.172
```

► 邮件中有机器的SSH远程登录信息，打开terminal。SSH是Linux远程访问的指令，p的意思是端口，后面接的是端口号，root@后面是服务器的IP。通过这个指令登录服务器。

<https://www.cnblogs.com/ftl1012/p/ssh.html>

```
shuyuej — root@e30aa6ff150d: ~ — ssh -p 20151 root@116.85.24.172 — 80x24
Last login: Mon Oct 28 21:07:09 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) SHUYUEs-MacBook-Pro:~ shuyuej$ ssh -p 20151 root@116.85.24.172
root@116.85.24.172's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Mon Oct 28 21:07:21 2019 from 127.0.0.1
(base) root@e30aa6ff150d:~#
```

注意：输入密码的时候直接复制粘贴就好，terminal内不会显示出密码来，但是已经输入了！登录后就会显示一下的

(base) root@e30aa6ff150d:~#
后面就可以输入Linux的各种指令

```
shuyuej — root@e30aa6ff150d: /home — ssh -p 20151 root@116.85.24.172 — 118x43
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

Last login: Mon Oct 28 21:07:21 2019 from 127.0.0.1
[(base) root@e30aa6ff150d:~# cd ..
[(base) root@e30aa6ff150d:/# cd home
[(base) root@e30aa6ff150d:/home# pwd
/home
[(base) root@e30aa6ff150d:/home# ls -al
总用量 632424
drwxr-xr-x 1 root root      4096 10月 28 21:11 .
drwxr-xr-x 1 root root      4096 10月 28 09:59 ..
-rw-r--r-- 1 root root    33474 10月 21 10:46 Adjacency_Matrix.csv
drwxr-xr-x 4 root root      4096 10月 28 21:25 checkpoints
drwxr-xr-x 3 1001 999       4096 9月 17 18:35 dbc_utils
drwxr-xr-x 3 root root      4096 10月 25 21:10 lib
-rwxr-xr-x 1 root root 139467733 10月 28 10:26 lib.rar
-rw-r--r-- 1 root root      2357 10月 28 16:33 main10.py
-rw-r--r-- 1 root root      2357 10月 28 16:34 main11.py
-rw-r--r-- 1 root root      2357 10月 28 16:34 main12.py
-rw-r--r-- 1 root root      2361 10月 28 21:10 main2.py
-rw-r--r-- 1 root root      2361 10月 28 21:10 main3.py
-rw-r--r-- 1 root root      2357 10月 28 16:32 main4.py
-rw-r--r-- 1 root root      2357 10月 28 16:32 main5.py
-rw-r--r-- 1 root root      2357 10月 28 16:32 main6.py
-rw-r--r-- 1 root root      2357 10月 28 16:32 main7.py
-rw-r--r-- 1 root root      2357 10月 28 16:33 main8.py
-rw-r--r-- 1 root root      2357 10月 28 16:33 main9.py
-rwxrwxrwx 1 root root      2435 10月 28 21:10 main.py
-rw-r--r-- 1 root root        207 10月 28 21:09 run.sh
drwxr-xr-x 4 root root      4096 10月 28 21:25 summaries
-rw-r--r-- 1 root root      2476 10月 28 15:49 test1.py
-rw-r--r-- 1 root root      2372 10月 28 16:08 test2.py
-rw-r--r-- 1 root root      2369 10月 28 11:22 test3.py
-rw-r--r-- 1 root root      2369 10月 28 11:23 test4.py
-rw-r--r-- 1 root root    215040 10月 21 10:48 test_label.csv
-rw-r--r-- 1 root root      2362 10月 28 12:44 test.py
-rw-r--r-- 1 root root    50575901 10月 21 10:47 test_set.csv
-rw-r--r-- 1 root root         0 10月 28 11:23 train1.py
-rw-r--r-- 1 root root    1935360 10月 21 10:48 training_label.csv
-rw-r--r-- 1 root root    455259876 10月 21 10:47 training_set.csv
(base) root@e30aa6ff150d:/home#
```

cd .. 返回上一个文件夹

cd ***

到路径下的某文件夹

pwd 输出目前文件夹路径

ls -al 或者 ls 输出目前路径下文件夹内的全部文件

rm -rf ***

删除文件夹或文件

```
(base) root@e30aa6ff150d:/home# nvidia-smi
Mon Oct 28 22:02:02 2019

+-----+
| NVIDIA-SMI 430.40          Driver Version: 430.40          CUDA Version: 10.1     |
+-----+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
|  0   GeForce RTX 208...    On          | 00000000:18:00.0 Off  |            N/A       |
| 43%   70C    P2     247W / 250W | 10976MiB / 11019MiB |    98%       Default   |
+-----+-----+-----+

+-----+
| Processes:                                     GPU Memory |
|  GPU           PID    Type   Process name                               Usage      |
|====+=====+=====+=====+=====+=====+=====+

```

通过nvidia-smi指令查看服务器里面的显卡信息和使用率(Volatile GPU-Util)。nvidia-smi -l可以不断输出目前显卡的运行和使用情况，等发现这个使用率到了90%这个样子，说明显卡就在全速运行了，要是0%的话说明显卡没有使用，那时候跑程序使用的是cpu。

https://blog.csdn.net/weixin_39881922/article/details/82863321

```
(base) root@e30aa6ff150d:/home# nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005–2017 NVIDIA Corporation
Built on Fri_Sep__1_21:08:03_CDT_2017
Cuda compilation tools, release 9.0, V9.0.176
(base) root@e30aa6ff150d:/home#
```

使用nvcc -V指令去查看服务器CUDA版本的信息。

通过scp指令将本地文件上传到服务器

- ▶ 上传文件建议把所有的文件压缩成zip或者rar格式，这样方便上传与压缩。
- ▶ 首先上传文件到服务器，再在服务器内把文件解压缩：

上传本地文件到服务器举例：

```
scp -P 52182 /Users/shuyuej/Desktop/hello.zip root@111.44.254.168:/home/
```

在电脑本地新打开一个terminal，scp是上传文件的指令，-P是端口的意思，这里P需要大写，第一个路径是本地文件hello.zip的地址，后面root@111.44.254.168加个冒号后，再写的地址是服务器里你想把文件放下的地址。

- ▶ rar文件解压缩需要下载的包：

```
apt install rar unrar
```

```
unrar x ****.rar
```

- ▶ zip文件解压缩需要下载的包：

```
apt install zip
```

```
unzip ****.zip
```

接下来配置Python+CUDA10运行环境

- ▶ 上传文件到服务器里面后，接下来要配置环境去执行程序了。
- ▶ 首先，现在主流是使用英伟达CUDA10.0+Anaconda去跑python的程序，比如说Tensorflow或者Pytorch。
- ▶ 第一步是配置CUDA，
- ▶ 如何切换cuda版本？
- ▶ --切换为cuda10版本命令:`ln -snf /usr/local/cuda-10.0/ /usr/local/cuda`
- ▶ --切换为cuda9版本命令: `ln -snf /usr/local/cuda-9.0/ /usr/local/cuda`

注意：切换信息版本后可能在跑Tensorflow时候会有cudnn文件报错（无法使用cudnn），具体操作步骤请看后页！

这里需要通过以下指令把cudnn的些关键文件复制到系统能够识别的文件夹，执行以下操作。

```
sudo cp /usr/local/cuda-10.0/lib64/libcudart.so.10.0 /usr/local/lib/libcudart.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcublas.so.10.0 /usr/local/lib/libcublas.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcufft.so.10.0 /usr/local/lib/libcufft.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcurand.so.10.0 /usr/local/lib/libcurand.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcusolver.so.10.0 /usr/local/lib/libcusolver.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcusparse.so.10.0 /usr/local/lib/libcusparse.so.10.0 && sudo ldconfig
sudo cp /usr/local/cuda-10.0/lib64/libcufft.so.10.0 /usr/local/lib/libcufft.so.10.0 && sudo ldconfig
```

接下来配置Python+CUDA10运行环境

- ▶ 配置好CUDA后接下来配置Python的运行环境，一般情况下深度学习的服务器已经把Anaconda的conda装好了，但是一般情况下我们可以自己创建一个python的环境并且安装些自己需要用的python包。
- ▶ `conda create --name ***** python=3.7 numpy scipy`
- ▶ 例如：`conda create --name tensorflow python=3.7 numpy scipy`
- ▶ name后面的名字随便写，我这里是创建了一个名称叫做tensorflow，python3.7环境的python，然后附加安装了numpy和scipy。
- ▶ 创建完成后，通过**`conda activate tensorflow`**激活环境。
- ▶ 注意：激活环境后前面的(base)会变成(tensorflow)这样的名称。
- ▶ 激活环境后安装需要安装的包
- ▶ 主要的指令是**`conda install ***`** 或者**`pip install ***`**
- ▶ 具体看下页。

```
python -m pip install --upgrade pip
pip install --upgrade cython
pip install --upgrade textblob
pip install --upgrade nltk
pip install --upgrade jieba
pip install --upgrade visdom
pip install --upgrade keras
pip install --upgrade pillow
pip install --upgrade opencv-python
pip install --upgrade opencv-contrib-python
pip install --upgrade theano
pip install --upgrade scipy
pip install --upgrade pandas
```

比如说，我这随便安装某些包（指令都一样，后面加包就行），下面可以安装tensorflow。

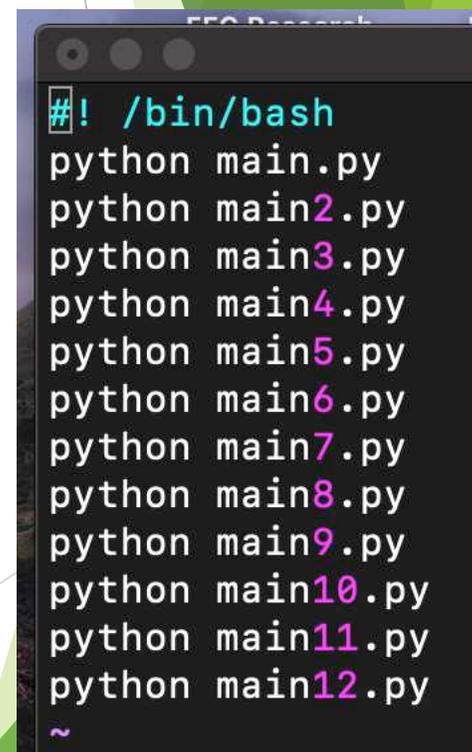
```
pip install --upgrade --force-reinstall tensorflow-gpu==1.14.0 --user
```

```
pip install tensorflow-gpu==1.14.0 -i https://pypi.tuna.tsinghua.edu.cn/simple  
pip install gensim -i https://pypi.tuna.tsinghua.edu.cn/simple  
pip install sklearn -i https://pypi.tuna.tsinghua.edu.cn/simple  
pip install matplotlib -i https://pypi.tuna.tsinghua.edu.cn/simple  
pip install scipy -i https://pypi.tuna.tsinghua.edu.cn/simple  
pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple
```

国内的话请在pip install ***** 后面加上-i <https://pypi.tuna.tsinghua.edu.cn/simple>这句话，咱们用清华大学开源的python包镜像，这样下载速度更快些！（Python默认下载的镜像在美帝，下载速度老慢了！！！！）

如何让服务器执行多个文件

- ▶ 执行python文件的指令是python *****.py
- ▶ 通过上述指令可以执行python文件，可以写个shell来连续执行多个python的文件，具体操作是，自己写好了多个python文件，比如说我这是main.py, main2.py等等，需要写个.sh的文件,linux编辑文件使用vim来编辑的，这里第一行一定要写
- ▶ #! /bin/bash
- ▶ 然后接下来输入需要执行的python文件就行啦！
- ▶ 注意：比如说我这创建个run.sh的文件，使用vim run.sh创建，vim输入需要按键盘的“i”键，退出需要按下键盘“esc”后，然后输入“:wq”来编辑完成退出，shell和vim在公司中非常常用！！
- ▶ 批量执行python文件的方法是source ***.sh

A terminal window with a dark background and light text. The prompt is '#! /bin/bash'. Below it, a list of 12 'python' commands is shown, each followed by a filename from 'main1.py' to 'main12.py'. The filenames are color-coded: 'main1.py' is blue, 'main2.py' is red, 'main3.py' is green, 'main4.py' is yellow, 'main5.py' is cyan, 'main6.py' is magenta, 'main7.py' is black, 'main8.py' is white, 'main9.py' is red, 'main10.py' is cyan, 'main11.py' is magenta, and 'main12.py' is yellow. A tilde '~' is at the bottom of the terminal output.

```
#!/bin/bash
python main1.py
python main2.py
python main3.py
python main4.py
python main5.py
python main6.py
python main7.py
python main8.py
python main9.py
python main10.py
python main11.py
python main12.py
~
```

跑完程序的结果上传到本地电脑

- ▶ 在服务器跑完程序以后，这时候需要把结果压缩成压缩包
- ▶ `zip -r hello.zip /home/summaries/`
- ▶ 上述指令是把home地址下的summaries这个文件夹压缩成hello.zip
- ▶ 接下来需要用scp指令下载到本地：

```
scp -P 52182 root@111.44.254.168:/home/hello.zip /Users/shuyuej/Desktop/
```

和上传文件到服务器的指令是一样的，-P是端口号，P需要大写，root@后面是IP地址，注意不要忘记写冒号。

怎么让服务器程序后台运行，我自己电脑关机也不受影响

- ▶ `screen -S *****`
- ▶ `screen -S run`
- ▶ 上述命令意思是创建一个名叫run的screen
- ▶ `screen -ls`
- ▶ 上述命令是把目前所有的screen给输出
- ▶ --<https://blog.csdn.net/xfg0218/article/details/81869313>
- ▶ 如果想看当前screen，请输入以下指令，请看下页

```
For more details, please visit https://support.apple.com/kb/HT208050.
(base) SHUYUEs-MacBook-Pro:~ shuyuej$ ssh -p 4034 root@114.115.219.202
root@114.115.219.202's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.15.0-45-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
Last login: Wed Oct 30 08:55:23 2019 from 127.0.0.1
(base) root@2b2cd190be47:~# screen -ls
There are screens on:
  7341 run                (2019年10月30日 09时14分03秒) (Detached)
  423 .2b2cd190be47      (2019年10月30日 08时47分21秒) (Detached)
  394 .2b2cd190be47      (2019年10月30日 08时47分18秒) (Detached)
3 Sockets in /var/run/screen/S-
(base) root@2b2cd190be47:~# screen -r 7341
```

1. 通过screen -r **** 后面接的是之前创建的名为run的screen的端口号。
2. 注意：之前创建的名称为tensorflow的python环境，在screen -S run后需要重新激活：
conda activate tensorflow才能继续使用之前创建的环境。

解决 screen 连接不上, 提示“There is no screen to be resumed matching 18352.” 的问题

当你挂起screen, 下次想重新连上screen时, 有时会出现screen session的状态为Attached但是却连不上的情况, 比如我想重新进入session id 为18352的screen, 运行‘screen -r 18352’, 提示“There is no screen to be resumed matching 18352”, 解决方法如下:

```
screen -D -r 18352
```

解释: -D -r 先删除前一用户再登陆。

The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. These shapes are primarily located on the left and right sides of the frame, leaving a large white central area. The shapes are composed of triangles and polygons, some with thin white outlines.

THANKS