

navicester /
cookbook

<> CodeIssuesPull requestsActionsProjectsSecurityInsights

cookbook / ubuntu-base-image-setup.md

navicester Update ubuntu-base-image-setup.md

last week

493 lines (388 loc) · 17.2 KB

PreviewCodeBlame

RawCopyDownloadEditMore

Ubuntu 18.04

启动docker

该例子是基于 `nvcr.io/nvidia/cuda:11.3.0-devel-ubuntu18.04`

```
# 四块GPU卡
root@node-154:~# docker run --name neural-base --runtime=nvidia -it -p 18898:8888 -e NVIDIA_VISIBLE_DEVICES="0,1,2,3" -v /shared_storage/binhe/:/binhe nvcr.io/nvidia/cuda:11.3.0-devel-ubuntu18.04
```

```
# 一块GPU卡
sudo docker run --name ubuntu18.04 -it -p 18898:8888 -p 18899:8889 -p 18900:8890 -p 18901:8891 -p 18902:8892 -v /home:/home --gpus all nvcr.io/nvidia/cuda:11.3.0-devel-ubuntu18.04
```

安装基础软件

```
root@459e8eb2c438:/home# apt-get update
root@459e8eb2c438:/home# apt-get install wget sudo vim kmod unzip git -y
```

安装python

Ubuntu16.04默认python版本是python3.5, Ubuntu18.04默认python版本是python3.6

python 3.8



```
apt-get install python3 # 默认安装python3.6
apt-get install python3-pip

# 如果python3.8无法安装尝试下面安装命令
# apt install software-properties-common
# sudo add-apt-repository ppa:deadsnakes/ppa
# apt install python3.8-distutils

apt-get install python3.8 # 指定安装
wget https://bootstrap.pypa.io/get-pip.py # 无法apt安装python3.8的pip
python3.8 get-pip.py

apt-get install python3-venv # 默认python3.6
apt-get install python3.8-venv

root@8fd15aee3a7c:/# whereis python
python: /usr/bin/python3.6m /usr/bin/python3.8 /usr/bin/python3.6
/usr/lib/python3.8 /usr/lib/python3.6 /etc/python3.8 /etc/python3.6
/usr/local/lib/python3.8 /usr/local/lib/python3.6

root@459e8eb2c438:/home# which python3
/usr/bin/python3

pip3 install --upgrade pip
```

Python 3.10

操作系统为Ubuntu 16.04，默认的Python版本有2.7和3.5。由于不满足要求，需要更高版本的python。这里使用了Python3.10。其他操作系统或不同版本，请参考使用

先安装 OpenSSL 1.1.1

系统默认的是1.0.2.g，不满足要求（可以在Python源码make 阶段看到相关错误）



```
wget https://www.openssl.org/source/openssl-1.1.1k.tar.gz
tar -xzf openssl-1.1.1k.tar.gz
cd openssl-1.1.1k
./config
make
sudo make install
#sudo ln -s /usr/local/lib64/libcrypto.so.1.1 /usr/lib/x86_64-linux-
gnu/libcrypto.so.1.1
#sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib/x86_64-linux-
gnu/libssl.so.1.1
sudo ln -s /usr/local/lib/libcrypto.so.1.1 /usr/lib/x86_64-linux-
gnu/libcrypto.so.1.1
sudo ln -s /usr/local/lib/libssl.so.1.1 /usr/lib/x86_64-linux-
gnu/libssl.so.1.1
ldconfig -v

openssl version
OpenSSL 1.1.1k 25 Mar 2021
```

```
# /usr/bin/openssl版本没有自动更新
# /usr/bin/openssl version
OpenSSL 1.0.2g  1 Mar 2016
# /usr/local/bin/openssl version
OpenSSL 1.1.1k  25 Mar 2021

:/usr/local/bin# mv openssl openssl.0.2g
:/usr/local/bin# ln -s /usr/local/bin/openssl openssl
```

其他库

安装必需的libffi等，视情况而定（可以在Python源码make 阶段看到相关错误）

```
sudo apt-get update
sudo apt-get install build-essential libssl-dev zlib1g-dev libbz2-dev \
libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-
dev \
xz-utils tk-dev libffi-dev liblzma-dev python3-openssl

sudo apt install libffi-dev libssl-dev
```



下载Python 3.10 source并编译安装

```
wget https://www.python.org/ftp/python/3.10.9/Python-3.10.9.tgz
tar xzf Python-3.10.9.tgz
cd Python-3.10.9

vi Modules/Setup
# Socket module helper for SSL support; you must comment out the other
# socket line above, and edit the OPENSSL variable:
OPENSSL=/usr/local #openssl路径
_ssl _ssl.c \
    -I$(OPENSSL)/include -L$(OPENSSL)/lib \
    -lssl -lcrypto

./configure # --with-openssl Makefile里面已经有了

vi Makefile
OPENSSL_INCLUDES=-I/usr/local/include
OPENSSL_LDFLAGS=-L/usr/local/lib
OPENSSL=/usr/local # 原来值是/usr/local/openssl, 这个不是准确安装位置

make
sudo make install

python3 -version
Python 3.10.9
```



```
pip3 -version
pip 22.3.1 from /usr/local/lib/python3.10/site-packages/pip (python 3.10)
```

问题

WARNING: pip is configured with locations that require TLS/SSL, however the ssl module in Python is not available.

Could not fetch URL https://pypi.tuna.tsinghua.edu.cn/simple/pip/: There was a problem confirming the ssl certificate:

HTTPConnectionPool(host='pypi.tuna.tsinghua.edu.cn', port=443): Max retries exceeded with url: /simple/pip/ (Caused by SSLError("Can't connect to HTTPS URL because the SSL module is not available.)) - skipping



Could not build the ssl module! Python requires a OpenSSL 1.1.1 or newer

修改Makefile等配置文件的OPENSSL位置进行重新编译

Issue: SSLError: (MaxRetryError("HTTPConnectionPool(host='huggingface.co', port=443):
Max retries exceeded with url: /NousResearch/Llama-2-7b-chat-hf/resolve/main/tokenizer_config.json
(Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self-signed certificate in certificate chain (_ssl.c:1007)'))")), '(Request ID: 2ef5cf06-2aca-4493-9c45-c9d73e7b7cba)')

/usr/local/lib/python3.10/dist-packages/urllib3/connectionpool.py

```
705         conn = self._get_conn(timeout=pool_timeout)
706         import ssl
707         conn.verify_mode = ssl.CERT_NONE

716
717         # Make the request on the httplib connection object.
718         httplib_response = self._make_request(
719             conn,
720             method,
721             url,
722             timeout=timeout_obj,
723             body=body,
724             headers=headers,
725             chunked=chunked,
726         )
727
```

/usr/local/lib/python3.10/dist-packages/urllib3/connection.py

```
405         context = self.ssl_context
406         context.verify_mode = ssl.CERT_NONE #
resolve_cert_reqs(self.cert_reqs)
```

参考

- <https://www.atext.cn/2022/12/19/ubuntu-16%E5%AE%89%E8%A3%85python-3-10/>

切换python - 可选

```
sudo apt update
sudo apt -y upgrade
sudo apt install -y python3-pip
pip3 install --upgrade pip
pip3 install packaging
apt-get install python3.8 -y

#sudo apt-get install python3-dev
#sudo apt-get install python3.8-dev
which python3.8
sudo update-alternatives --install /usr/bin/python3 python3
/usr/bin/python3.8 1
which python3.6
sudo update-alternatives --install /usr/bin/python3 python3
/usr/bin/python3.6 2
sudo update-alternatives --config python3
# 如何涉及到anaconda的环境，切换不成功可以重启一下容器
```



安装anaconda

```
sudo apt-get update
cd /tmp
apt-get install wget
wget https://repo.anaconda.com/archive/Anaconda3-2022.05-Linux-x86_64.sh
# 默认安装python3.9
bash Anaconda3-2022.05-Linux-x86_64.sh
```



```
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> yes
```

```
source ~/.bashrc
```

```
conda config --add channels
http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
conda config --add channels
http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
conda config --set show_channel_urls yes
```



2.删去 /root/.condarc 文件 -default 这一行:

```
ssl_verify: false
channels:
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/linux # don't
```



```
use win-64
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/linux
show_channel_urls: true
# - defaults
```

配置虚拟环境

```
conda activate my-conda-env    # this is the environment for your project
                                and code
conda install ipykernel
conda deactivate

conda activate base            # could be also some other environment
conda install nb_conda_kernels
jupyter notebook
```



安装Jupyter

```
# https://stackoverflow.com/questions/42222096/no-module-named-packaging
pip3 install --upgrade pip
pip3 install packaging

pip3 install jupyter markupsafe==2.0.1 jupyter_contrib_nbextensions
jupyter_nbextensions_configurator
# pip3 install jupyter markupsafe jupyter_contrib_nbextensions
jupyter_nbextensions_configurator
```



配置Jupyter

```
pip install notebook==6.1.0
jupyter contrib nbextension install --user
jupyter nbextensions_configurator enable --user
```



```
# 问题1:
# ModuleNotFoundError: No module named 'notebook.base'
```



```
(neural) root@8fd15aee3a7c:/# pip freeze | grep notebook
notebook==7.0.4
notebook_shim==0.2.3
```

```
(neural) root@8fd15aee3a7c:/# pip install notebook==6.1.0
```

```
# 问题2:
# https://stackoverflow.com/questions/75511394/jupyter-contrib-nbextension-
install-user-pkg-resources-distributionnotfound-we
```

问题3:

WARNING: The scripts jupyter, jupyter-migrate and jupyter-troubleshoot are installed in '/usr/local/python3/bin' which is not on PATH.

Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.

```
export PATH=$PATH:/usr/local/python3/bin
```

ImportError: urllib3 v2.0 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with LibreSSL 2.8.3

<https://stackoverflow.com/questions/76187256/importerror-urllib3-v2-0-only-supports-openssl-1-1-1-currently-the-ssl-modu>

```
pip install urllib3==1.26.6
```

Issue: root@f0b7b670448b:~/.jupyter# [C 19:39:51.363 NotebookApp] Bad config encountered during initialization: The 'contents_manager_class' trait of a NotebookApp instance expected a subclass of 'notebook.services.contents.manager.ContentsManager', not the JupyterTextContentsManager JupyterTextContentsManager.

```
pip install notedown
```

root@f0b7b670448b:~/.jupyter# [C 19:46:30.474 NotebookApp] Bad config encountered during initialization: The 'contents_manager_class' trait of <notebook.notebookapp.NotebookApp object at 0x7fc774083f10> instance must be a type, but 'notedown.NotedownContentsManager' could not be imported

```
root@f0b7b670448b:~/.jupyter# jupyter notebook --generate-config
```

Writing default config to: /root/.jupyter/jupyter_notebook_config.py

将下面这一行加入到生成的配置文件的末尾（Linux/macOS一般在

```
~/.jupyter/jupyter_notebook_config.py)
```

```
c.NotebookApp.contents_manager_class = 'notedown.NotedownContentsManager'
```

Issue: AttributeError: 'NotebookApp' object has no attribute 'io_loop'

Issue: ModuleNotFoundError: No module named 'jupyter_tensorboard'

```
pip install jupyter-tensorboard
```

设置密码

```
jupyter notebook password
```



生成jupyter notebook的配置文件

```
jupyter notebook --generate-config
```

```
vim ~/.jupyter/jupyter_notebook_config.py
```

```
c.NotebookApp.ip='*'
```

```
c.NotebookApp.password =
```

```
u'sha1:41e4da01dde4:e820dc9c0398eda2dc9323c9e4a51ea1228166a2'
```

```
c.NotebookApp.open_browser = False
```



```
c.NotebookApp.allow_remote_access = True
c.NotebookApp.port = 8888
```

启动Jupyter

```
cd /binhe
jupyter notebook --ip 0.0.0.0 --port 8888 --allow-root&
```



启动虚拟环境 venv - 可选

```
root@8fd15aee3a7c:/# python3.8 -m venv neural
root@8fd15aee3a7c:/# source neural/bin/activate

(neural) root@8fd15aee3a7c:/# deactivate
root@8fd15aee3a7c:/# rm -rf neural/
```



在Jupyter环境里添加path

Jupyter和python的环境不一样，如果不指定path，那么module的import会失败

```
# https://stackoverflow.com/questions/34976803/sys-path-different-in-
jupyter-and-python-how-to-import-own-modules-in-jupyter
import sys
print(sys.path)
#sys.path.append("/neural/lib/python3.8/site-packages/")
sys.path.insert(0, "/neural/lib/python3.8/site-packages/")
```



安装pytorch - 可选

```
pip install torch==1.12.1+cu113 torchvision==0.13.1+cu113 torchaudio==0.12.1
--extra-index-url https://download.pytorch.org/whl/cu113
```



安装opencv - 可选

```
# open-cv
apt install libopencv-dev python3-opencv -y
apt-get install ffmpeg libsm6 libxext6 -y
```



Ubuntu 16.04

基于image nvidia/cuda:10.0-cudnn7-devel-ubuntu16.04

```
root@57df1346cab8:/binhe# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.7 LTS
Release:       16.04
Codename:      xenial
```



```
root@57df1346cab8:/binhe# python3 -V
Python 3.8.0
```



```
root@57df1346cab8:/binhe# whereis python
python: /usr/bin/python3.5m /usr/bin/python3.8 /usr/bin/python3.5
/usr/lib/python2.7 /usr/lib/python3.5 /etc/python3.5
/usr/local/bin/python3.8-config /usr/local/bin/python3.8
/usr/local/lib/python3.8 /usr/local/lib/python3.5
```

```
root@57df1346cab8:/binhe# nvidia-smi
Fri Sep 22 15:09:15 2023
+-----+
--+
| NVIDIA-SMI 465.19.01      Driver Version: 465.19.01      CUDA Version: 11.3
|
|-----+-----+-----+
--+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |
| Fan   Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute
M. |
|                |                    |              MIG
M. |
|=====+=====+=====+
|   0   NVIDIA Tesla V1...  Off   | 00000000:3D:00.0 Off |
0 |
| N/A    37C    P0      35W / 250W | 16819MiB / 32510MiB |      0%
Default |
|                |                    |
N/A |
+-----+-----+-----+
--+
|   1   NVIDIA Tesla V1...  Off   | 00000000:3E:00.0 Off |
0 |
| N/A    37C    P0      36W / 250W | 17441MiB / 32510MiB |      0%
Default |
|                |                    |
N/A |
+-----+-----+-----+
--+
|   2   NVIDIA Tesla V1...  Off   | 00000000:88:00.0 Off |
0 |
```



```
| N/A 34C P0 35W / 250W | 17441MiB / 32510MiB | 0%
Default |
|
N/A |
+-----+-----+-----+
--+
| 3 NVIDIA Tesla V1... Off | 00000000:89:00.0 Off |
0 |
| N/A 36C P0 36W / 250W | 14759MiB / 32510MiB | 0%
Default |
|
N/A |
+-----+-----+-----+
--+

+-----+-----+-----+
--+
| Processes:
|
| GPU GI CI PID Type Process name GPU
Memory |
| ID ID Usage
|
|=====|
+-----+-----+-----+
--+
root@57df1346cab8:/binhe# nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Sat_Aug_25_21:08:01_CDT_2018
Cuda compilation tools, release 10.0, V10.0.130
```

lsb_release

python更新后 lsb_release 可能不能正常工作，修改 /usr/bin/lsb_release ，修改python为初始默认版本

```
#!/usr/bin/python3.5 -Es
```

GPG error

https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64

添加key

```
sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80
A4B469963BF863CC
```

或者

```
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys A4B469963BF863CC
```

docker - no space left on device

1. 停止 Docker 服务: `sudo systemctl stop docker`
2. 复制 `/var/lib/docker` 目录到新的路径: `sudo rsync -aP /var/lib/docker /new/path/`
3. 备份 `/etc/docker/daemon.json` 文件: `sudo cp /etc/docker/daemon.json /etc/docker/daemon.json.bak`
4. 编辑 `/etc/docker/daemon.json` 文件, 将 `"data-root"` 字段更改为新的路径: `sudo nano /etc/docker/daemon.json`, 修改 `"data-root": "/var/lib/docker"` 为 `"data-root": "/new/path/docker"`
5. 启动 Docker 服务: `sudo systemctl start docker`

```
# example
cat /etc/docker/daemon.json
{
  "data-root": "/home/docker-archive"
}
```



generated docker folder `/home/docker-archive/docker`

Cert

pip install fails with "connection error: [SSL: CERTIFICATE_VERIFY_FAILED]"

```
pip config set global.trusted-host \
  "pypi.org files.pythonhosted.org pypi.python.org" \
  --trusted-host=pypi.python.org \
  --trusted-host=pypi.org \
  --trusted-host=files.pythonhosted.org
```



```
sudo update-ca-certificates --fresh
export SSL_CERT_DIR=/etc/ssl/certs
```



Issue: apt-get update failed because certificate verification failed

Temporarily disable certificate verification by adding `Acquire { https::Verify-Peer false }` in `/etc/apt/apt.conf.d/99verify-peer.conf`.

- <https://askubuntu.com/questions/1095266/apt-get-update-failed-because-certificate-verification-failed-because-handshake>



```
Issue: SSLError: (MaxRetryError("HTTPSConnectionPool(host='huggingface.co',
port=443):
Max retries exceeded with url: /NousResearch/Llama-2-7b-chat-
hf/resolve/main/tokenizer_config.json
(Caused by SSLError(SSLCertVerificationError(1, '[SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed:
self-signed certificate in certificate chain (_ssl.c:1007)'))"), '(Request
ID: 2ef5cf06-2aca-4493-9c45-c9d73e7b7cba)')
```

```
/usr/local/lib/python3.10/dist-packages/urllib3/connectionpool.py
```

```
705         conn = self._get_conn(timeout=pool_timeout)
706         import ssl
707         conn.verify_mode = ssl.CERT_NONE

716
717         # Make the request on the httplib connection object.
718         httplib_response = self._make_request(
719             conn,
720             method,
721             url,
722             timeout=timeout_obj,
723             body=body,
724             headers=headers,
725             chunked=chunked,
726         )
727
```

```
/usr/local/lib/python3.10/dist-packages/urllib3/connection.py
```

```
405         context = self.ssl_context
406         context.verify_mode = ssl.CERT_NONE #
resolve_cert_reqs(self.cert_reqs)
```