調整Jupyter image

Jupyter 無法安裝 Gensim

```
unable to execute 'gcc': No such file or directory

---------------------------------------------------------------------------
DistutilsExecError                        Traceback (most recent call last)
/opt/anaconda/anaconda3/lib/python3.6/distutils/unixccompiler.py in _compile(self, obj, src, ext, cc_args, extra_po
stargs, pp_opts)
    117             self.spawn(compiler_so + cc_args + [src, '-o', obj] +
--> 118                        extra_postargs)
    119         except DistutilsExecError as msg:

/opt/anaconda/anaconda3/lib/python3.6/distutils/ccompiler.py in spawn(self, cmd)
    908     def spawn(self, cmd):
--> 909         spawn(cmd, dry_run=self.dry_run)
    910

/opt/anaconda/anaconda3/lib/python3.6/distutils/spawn.py in spawn(cmd, search_path, verbose, dry_run)
     35     if os.name == 'posix':
---> 36         _spawn_posix(cmd, search_path, dry_run=dry_run)
     37     elif os.name == 'nt':

/opt/anaconda/anaconda3/lib/python3.6/distutils/spawn.py in _spawn_posix(cmd, search_path, verbose, dry_run)
    158                           "command %r failed with exit status %d"
--> 159                           % (cmd, exit_status))
    160             elif os.WIFSTOPPED(status):

DistutilsExecError: command 'gcc' failed with exit status 1

During handling of the above exception, another exception occurred:
```

沒有gcc，就來安裝gcc

https://github.com/jupyterhub/jupyterhub/issues/1640

觀察 tensorflow/notebook Dockerfile

https://github.com/jupyter/docker-stacks/blob/master/tensorflow-notebook/Dockerfile

```
# Copyright (c) Jupyter Development Team.
# Distributed under the terms of the Modified BSD License.
ARG OWNER=jupyter
ARG BASE_CONTAINER=$OWNER/scipy-notebook
FROM $BASE_CONTAINER

LABEL maintainer="Jupyter Project <jupyter@googlegroups.com>"

# Install Tensorflow
RUN mamba install --quiet --yes \
    'tensorflow=2.4.1' && \
    mamba clean --all -f -y && \
    fix-permissions "${CONDA_DIR}" && \
    fix-permissions "/home/${NB_USER}"
```

調整 docker image

建立 build 資料夾

```
cd /opt/nlp/2019_nlp
mkdir build
```

```
drwxr-xr-x.  2 root root    24 Jul 16 12:39 build
-rw-r--r--.  1 root root   360 Jul 16 12:38 docker-compose.yml
-rw-r--r--.  1 root root   380 Jul 16 12:31 [redacted]
-rw-r--r--.  1 root root   360 Jul 16 12:36 [redacted]
-rw-r--r--.  1 root root 28769 Jul 11 02:37 jupyter_notebook_config.py
drwxr-xr-x.  2 root root    25 Jul 11 02:37 nlp_lab
-rw-r--r--.  1 root root    11 Jul 11 02:37 README.md
drwxr-xr-x. 11 root root  4096 Jul 11 02:37 work
```

調整 docker image

建立 Dockerfile 檔案

```
cd /opt/nlp/2019_nlp/build

vi Dockerfile
```

```
[root@localhost 2019_nlp]# cd build/
[root@localhost build]# ll
total 4
-rw-r--r--. 1 root root 487 Jul 16 12:39 Dockerfile
```

```
ARG OWNER=jupyter
ARG BASE_CONTAINER=$OWNER/scipy-notebook
FROM $BASE_CONTAINER

#LABEL maintainer="Jupyter Project <jupyter@googlegroups.com>"

USER root

RUN apt-get update --yes
RUN apt-get -y install libc-dev
RUN apt-get -y install build-essential
RUN pip install -U pip

USER ${NB_UID}

# Install Tensorflow
RUN mamba install --quiet --yes \
    'tensorflow=2.4.1' && \
    mamba clean --all -f -y && \
    fix-permissions "${CONDA_DIR}" && \
    fix-permissions "/home/${NB_USER}"
```

https://github.com/jupyter/docker-stacks/blob/master/tensorflow-notebook/Dockerfile

調整docker-compose

調整docker-compose.yml

```
services:
  notebook:
    image: jupyter/tensorflow-notebook
    build: build/.
    container_name: jupyter
    hostname: jupyter
    restart: unless-stopped
    volumes:
      - ./work:/home/jovyan/work
      - ./jupyter_notebook_config.py:/root/.jupyter/jupyter_notebook_config.py
    ports:
      - 9999:8888
    environment:
      - TZ=Asia/Taipei
      - JUPYTER_ENABLE_LAB=yes
```

重新啟動 Jupyter

docker-compose build

```
cd /opt/nlp/2019_nlp
docker-compose build
```

```
[root@localhost 2019_nlp]# docker-compose build
Building notebook
Sending build context to Docker daemon  2.048kB
Step 1/10 : ARG OWNER=jupyter
Step 2/10 : ARG BASE_CONTAINER=$OWNER/scipy-notebook
Step 3/10 : FROM $BASE_CONTAINER
 ---> 3d4570e716ca
Step 4/10 : USER root
 ---> Using cache
 ---> 36df5eed3e89
Step 5/10 : RUN apt-get update --yes
 ---> Using cache
 ---> 6b6431fe2dd8
Step 6/10 : RUN apt-get -y install libc-dev
 ---> Using cache
 ---> 9fb290162d9a
Step 7/10 : RUN apt-get -y install build-essential
 ---> Using cache
 ---> b120ab998479
Step 8/10 : RUN pip install -U pip
 ---> Using cache
 ---> 376a3c07d135
Step 9/10 : USER ${NB_UID}
 ---> Using cache
 ---> 7c06cfcf22e0
Step 10/10 : RUN mamba install --quiet --yes     'tensorflow=2.4.1' &&
 ---> Using cache
 ---> e02a632662f8
Successfully built e02a632662f8
Successfully tagged 2019_nlp_notebook:latest
```

重新啟動 Jupyter

docker-compose up -d

```
cd /opt/nlp/2019_nlp
docker-compose up -d
docker-compose ps
```

```
[root@localhost 2019_nlp]# docker-compose up -d
Creating network "2019_nlp_default" with the default driver
Creating jupyter ... done
[root@localhost 2019_nlp]# docker-compose ps
 Name               Command              State                     Ports
-------------------------------------------------------------------------------------
jupyter   tini -g -- start-notebook.sh   Up      0.0.0.0:9999->8888/tcp,:::9999->8888/tcp
[root@localhost 2019_nlp]#
```

使用 Gensim

使用範例程式 /work/Gensim/Gensim介紹