

Cloud Computing Kubernetes

Automated Essay Scoring System

Avery Niou
2022

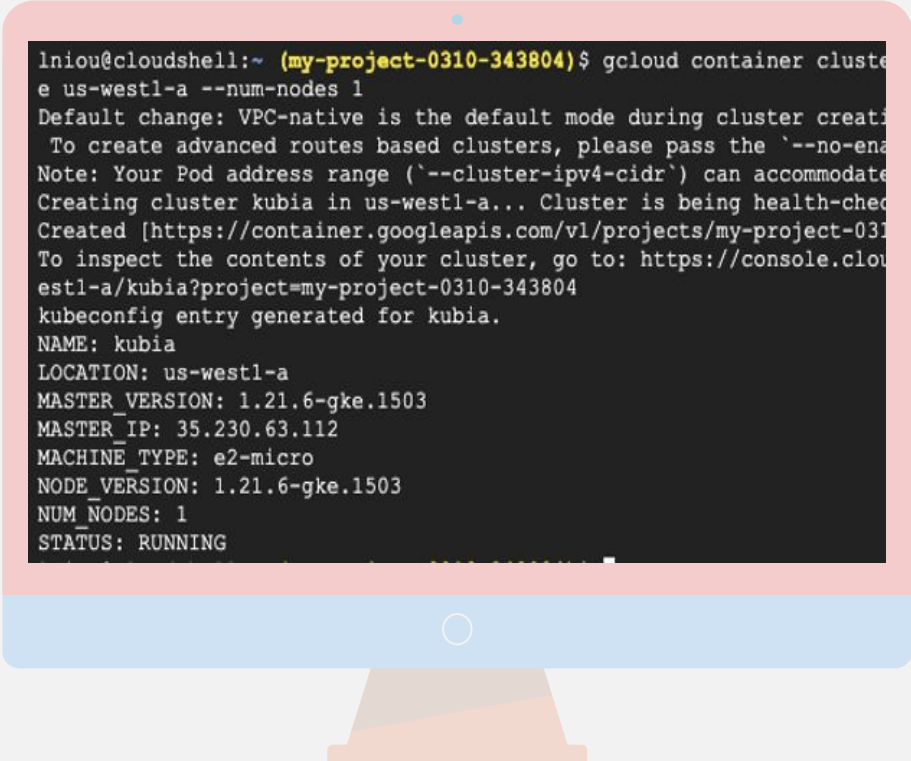
Project Overview

1. Create cluster on GKE.
2. Install Miniconda, Python and other applications.
3. Download the project from github
4. Install pytorch
5. Run the app

01

Create a cluster on GKE

```
$ gcloud container clusters create kubitia  
--machine-type=e2-micro  
--zone us-west1-a  
--num-nodes 1
```

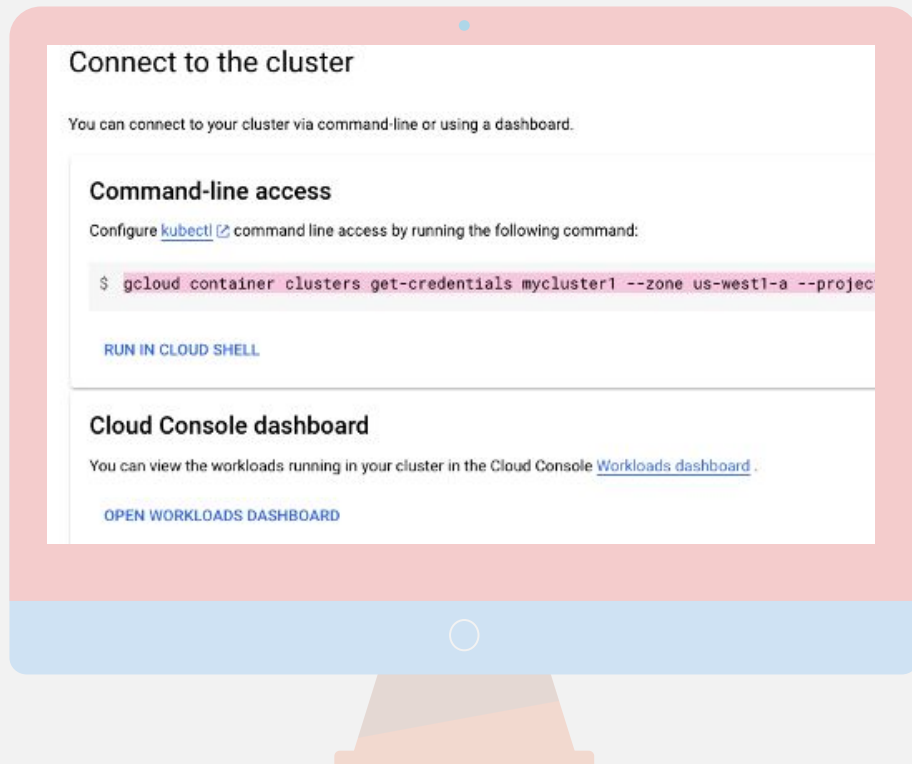


```
lniou@cloudshell:~ (my-project-0310-343804)$ gcloud container clusters create kubitia --zone us-west1-a --num-nodes 1  
Default change: VPC-native is the default mode during cluster creation.  
To create advanced routes based clusters, please pass the '--no-enable-advanced-routing' flag.  
Note: Your Pod address range ('--cluster-ipv4-cidr') can accommodate up to 128 pods.  
Creating cluster kubitia in us-west1-a... Cluster is being health-checked to ensure availability.  
Created [https://container.googleapis.com/v1/projects/my-project-0310-343804/locations/us-west1-a/clusters/kubitia]  
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/cluster/us-west1-a/kubitia?project=my-project-0310-343804  
kubeconfig entry generated for kubitia.  
NAME: kubitia  
LOCATION: us-west1-a  
MASTER_VERSION: 1.21.6-gke.1503  
MASTER_IP: 35.230.63.112  
MACHINE_TYPE: e2-micro  
NODE_VERSION: 1.21.6-gke.1503  
NUM_NODES: 1  
STATUS: RUNNING
```

02

Connect to cluster

```
$ gcloud container clusters  
get-credentials mycluster1  
--zone us-west1-a  
--project my-project-0310-343804
```



03

Download the latest shell script

Run miniconda installation script

```
$ wget  
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
$ ./Miniconda3-latest-Linux-x86_64.sh
```

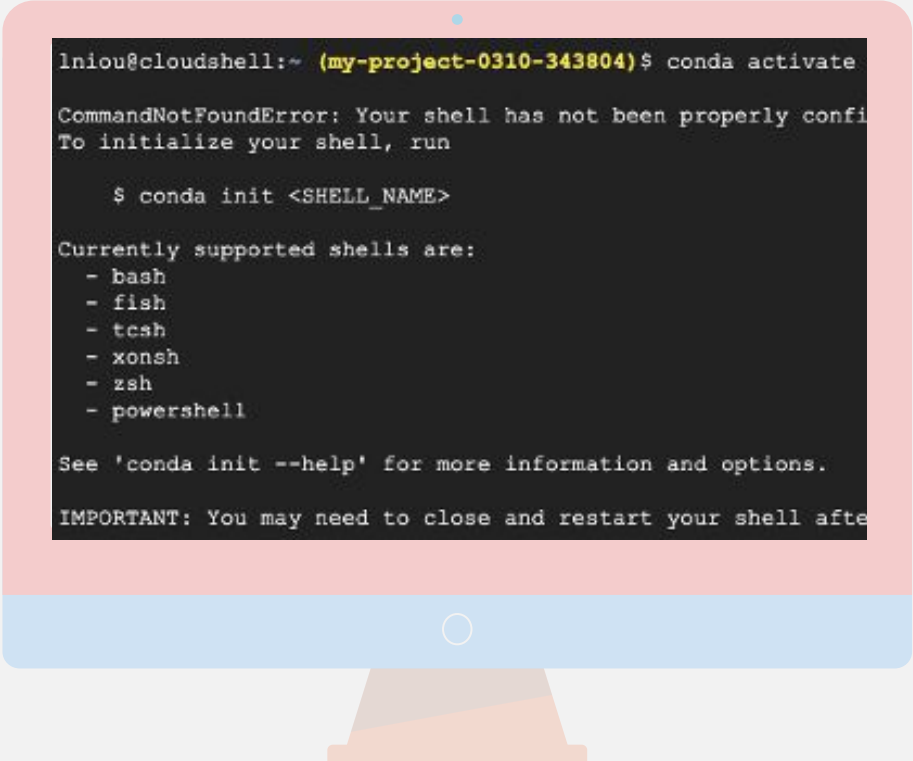
```
lniou@cloudshell:~ (my-project-0310-343804)$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh  
--2022-04-05 18:00:52-- https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh  
Resolving repo.anaconda.com (repo.anaconda.com)... 104.132.145.100  
Connecting to repo.anaconda.com (repo.anaconda.com)|104.132.145.100|:443  
HTTP request sent, awaiting response... 200 OK  
Length: 75660608 (72M) [application/x-sh]  
Saving to: 'Miniconda3-latest-Linux-x86_64.sh'  
  
Miniconda3-latest-Linux-x86_64.sh 100%[=====]  
  
2022-04-05 18:00:52 (189 MB/s) - 'Miniconda3-latest-Linux-x86_64.sh' [75660608 bytes]
```

```
Please answer 'yes' or 'no':  
>>> yes  
  
Miniconda3 will now be installed into this location:  
/home/lniou/miniconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/lniou/miniconda3] >>> █
```

04

Create and activate a Python environment

```
$ export  
PATH="/home/lniou/miniconda3/bin:$PATH"  
$ conda create -n myenv python=3.6  
$ conda activate myenv
```



```
lnious@cloudshell:~ (my-project-0310-343804)$ conda activate  
CommandNotFoundError: Your shell has not been properly confi  
To initialize your shell, run  
  
$ conda init <SHELL_NAME>  
  
Currently supported shells are:  
- bash  
- fish  
- tcsh  
-xonsh  
- zsh  
- powershell  
  
See 'conda init --help' for more information and options.  
  
IMPORTANT: You may need to close and restart your shell afte
```

05

Download material

Project from github

```
$ git clone https://github.com/Quan25/flask-summary.git
```

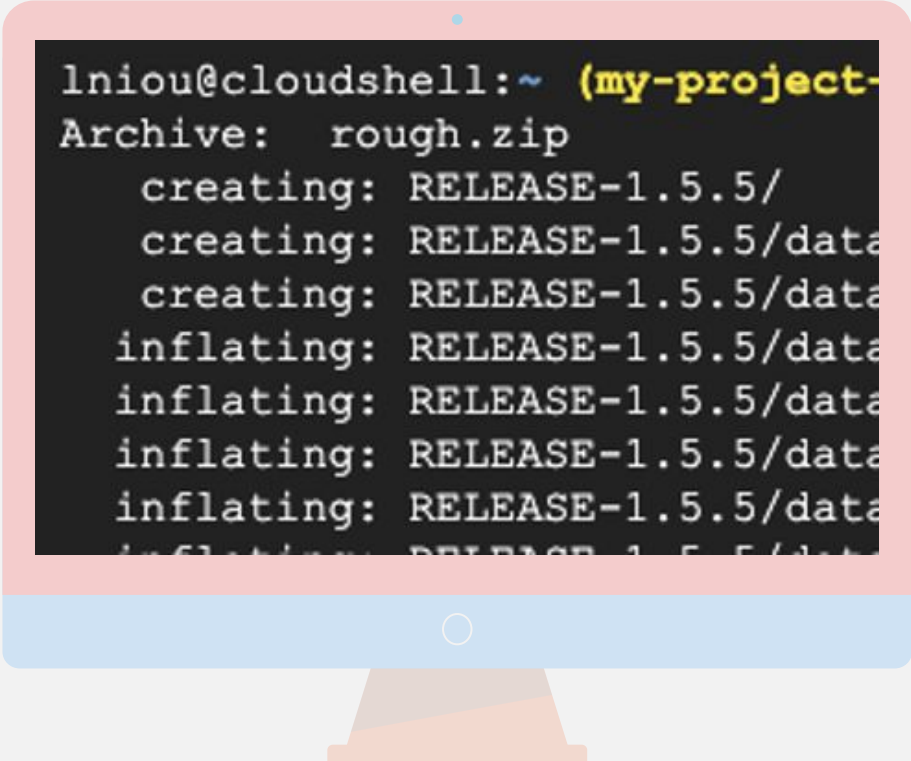
Download rough.zip

```
$ wget --load-cookies /tmp/cookies.txt  
"https://docs.google.com/uc?export=download&confirm=$(wget --quiet --savecookies  
/tmp/cookies.txt --keep-session-cookies --no-check-certificate  
'https://drive.google.com/file/d/1RxfZOYyNvzvCf37_vABfJMkohAsEZKtH/' -O- | sed -rn  
's/.confirm=([0-9A-Za-z_]+)./>\1\n/p')&id=1RxfZOYyNvzvCf37_vABfJMkohAsEZKtH" -O  
rough.zip && rm -rf /tmp/cookies.txt
```

06

Unzip rough.zip

```
$ unzip rough.zip
```

A terminal window is displayed on a computer monitor. The window has a black background with white text. The prompt is 'lniou@cloudshell:~' followed by '(my-project-' in yellow. The command 'unzip rough.zip' has been executed. The output shows the archive name 'rough.zip' and a series of 'creating' and 'inflating' messages for files within 'RELEASE-1.5.5/'. The files listed are 'RELEASE-1.5.5/', 'RELEASE-1.5.5/data', 'RELEASE-1.5.5/data', 'RELEASE-1.5.5/data', 'RELEASE-1.5.5/data', 'RELEASE-1.5.5/data', and 'RELEASE-1.5.5/data'. The terminal window is framed by a light blue border, and the monitor itself has a light blue bezel and a light orange stand.

```
lniou@cloudshell:~ (my-project-  
Archive:  rough.zip  
    creating: RELEASE-1.5.5/  
    creating: RELEASE-1.5.5/data  
    creating: RELEASE-1.5.5/data  
  inflating: RELEASE-1.5.5/data  
  inflating: RELEASE-1.5.5/data  
  inflating: RELEASE-1.5.5/data  
  inflating: RELEASE-1.5.5/data  
  inflating: RELEASE-1.5.5/data
```


Install material

```
$ sudo apt-get install libxml-parser-perl
$ sudo cpan install XML::Parser::PerlSAX
$ sudo cpan install XML::RegExp
$ sudo cpan install XML::DOM
```

```
lniou@cloudshell:~ (my-project-0310-343804) $ sudo cpan install XML::DOM
Loading internal logger. Log::Log4perl recommended for better logging
Reading '/root/.cpan/Metadata'
  Database was generated on Tue, 05 Apr 2022 18:17:03 GMT
Running install for module 'XML::DOM'
Fetching with LWP:
http://www.cpan.org/authors/id/T/TJ/TJMATHER/XML-DOM-1.46.tar.gz
Checksum for /root/.cpan/sources/authors/id/T/TJ/TJMATHER/XML-DOM-1.46.tar.gz ok
'YAML' not installed, will not store persistent state
Configuring T/TJ/TJMATHER/XML-DOM-1.46.tar.gz with Makefile.PL
Checking if your kit is complete...
Looks good
```

Install pyrouge

```
$ git clone https://github.com/bheinzerling/pyrouge.git
$ cd pyrouge
$ pip install -e .
$ conda install pytorch-cpu==1.1.0 torchvision-cpu==0.3.0 cpuonly -c pytorch
```

```
lniou@cloudshell:~/RELEASE-1.5.5 (my-project-0310-343804)$ git clone https://github.com/bheinzerling/pyrouge.git
Cloning into 'pyrouge'...
remote: Enumerating objects: 551, done.
remote: Total 551 (delta 0), reused 0 (delta 0), pack-reused 551
Receiving objects: 100% (551/551), 123.17 KiB | 4.93 MiB/s, done.
Resolving deltas: 100% (198/198), done.
lniou@cloudshell:~/RELEASE-1.5.5 (my-project-0310-343804)$ ls
data docs pyrouge README.txt RELEASE-NOTE.txt ROUGE-1.5.5.pl runROUGE-test.pl sample-output sample-test
lniou@cloudshell:~/RELEASE-1.5.5 (my-project-0310-343804)$ cd pyrouge
lniou@cloudshell:~/RELEASE-1.5.5/pyrouge (my-project-0310-343804)$ pip install -e .
Obtaining file:///home/lniou/RELEASE-1.5.5/pyrouge
Installing collected packages: pyrouge
  Running setup.py develop for pyrouge
Successfully installed pyrouge-0.1.3
```

Download pretrained-bert-model

```
$ wget https://s3.amazonaws.com/models.huggingface.co/bert/bert-large-uncased.tar.gz
$ tar -xf bert-large-uncased.tar.gz
$ cd flask-summary
$ cd summarizer
```

```
lniou@cloudshell:~ (my-project-0310-343804)$ wget https://s3.amazonaws.com/models.huggingface.co/bert/bert-large-uncased.tar.gz
--2022-04-05 18:45:54-- https://s3.amazonaws.com/models.huggingface.co/bert/bert-large-uncased.tar.gz
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.217.82.246
Connecting to s3.amazonaws.com (s3.amazonaws.com)|52.217.82.246|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1248501532 (1.2G) [application/x-tar]
Saving to: 'bert-large-uncased.tar.gz'

bert-large-uncased.tar.gz      100%[=====>] 1.16G 45.8MB/s in 27s

2022-04-05 18:46:22 (44.0 MB/s) - 'bert-large-uncased.tar.gz' saved [1248501532/1248501532]
```

```
def __init__(self, model_type: str, size: str):
    #self.model = self.model_handler[model_type].from_pretrained(self.size_handler[size][model_type])
    self.model = BertModel.from_pretrained('/home/lniou/bert-largeuncased.tar.gz')
    self.tokenizer = self.token_handler[model_type].from_pretrained(self.size_handler[size][model_type])
    self.vector_size = self.vector_handler[size][model_type]
    self.model_type = model_type
    self.model.eval()
```

```
$ pip3 install flask  
$ pip3 install pandas  
$ pip3 install sklearn  
$ pip3 install nltk  
$ pip3 install gensim==3.8.3  
$ pip3 install pytorch-pretrained-bert  
$ pip3 install matplotlib==3.0.0
```

Download punkt package with nltk

```
lniou@cloudshell:~ (my-project-0310-343804) $ python3
Python 3.9.7 (default, Sep 16 2021, 13:09:58)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nltk
>>> nltk.download('punkt')
[nltk_data] Downloading package punkt to /home/lniou/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
>>> exit()
```

\$ Run app.py

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
INFO:werkzeug: * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
INFO:werkzeug: * Restarting with fsevents reloader
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 187-412-584
```

Use Web Preview and change the port number to 5000

Change Preview Port

Port Number *

5000

CANCEL

CHANGE AND PREVIEW

5000-cs-917776103287-default.cs-us-west1-ijlt.cloudshell.dev

Please paste the contents that you want to summarize:

Russia has asked China for military assistance in Ukraine, including drones, a senior US official said Sunday.

CNN has reached out to the Russian embassy in the US for comment but did not receive an immediate response.

☒ Add To Instructor

☐ Add To Student

20%

submit

Grade Students

Reset

THANKS

Do you have any questions?

