ML AWS Deployment

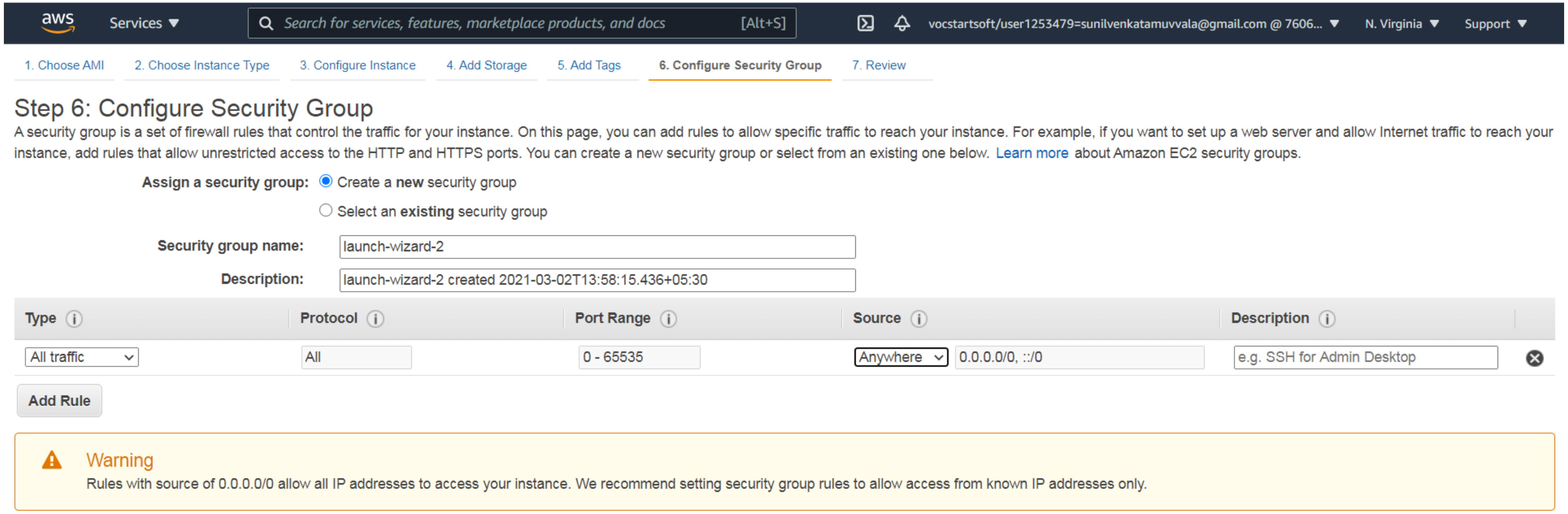
Steps

- Spin UP EC2 server
- Configure EC2 with security group and private key
- Install libraries and dependencies on EC2 server
- Move trained model and app.py to EC2
- Configure flaskapp.wsgi file and Apache vhost file
- Restart Apache web server and check API status

Follow the seven step flow to launch the instance

- Launch the instance in N.virginia us-east-1d with public enabling o subnet.
- Select ubuntu t2.micro which is free tier eligible
- Add default storage
- Configure security group to allow traffic
- Download new key pair to SSH the launched instance

Configure security group to allow all traffic


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
vocstartsoft/user1253479=sunilvenkatamuvvala@gmail.com @ 7606... ▼
N. Virginia ▼
Support ▼
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: [✓ Create a **new** security group] Select an **existing** security group

Security group name: launch-wizard-2

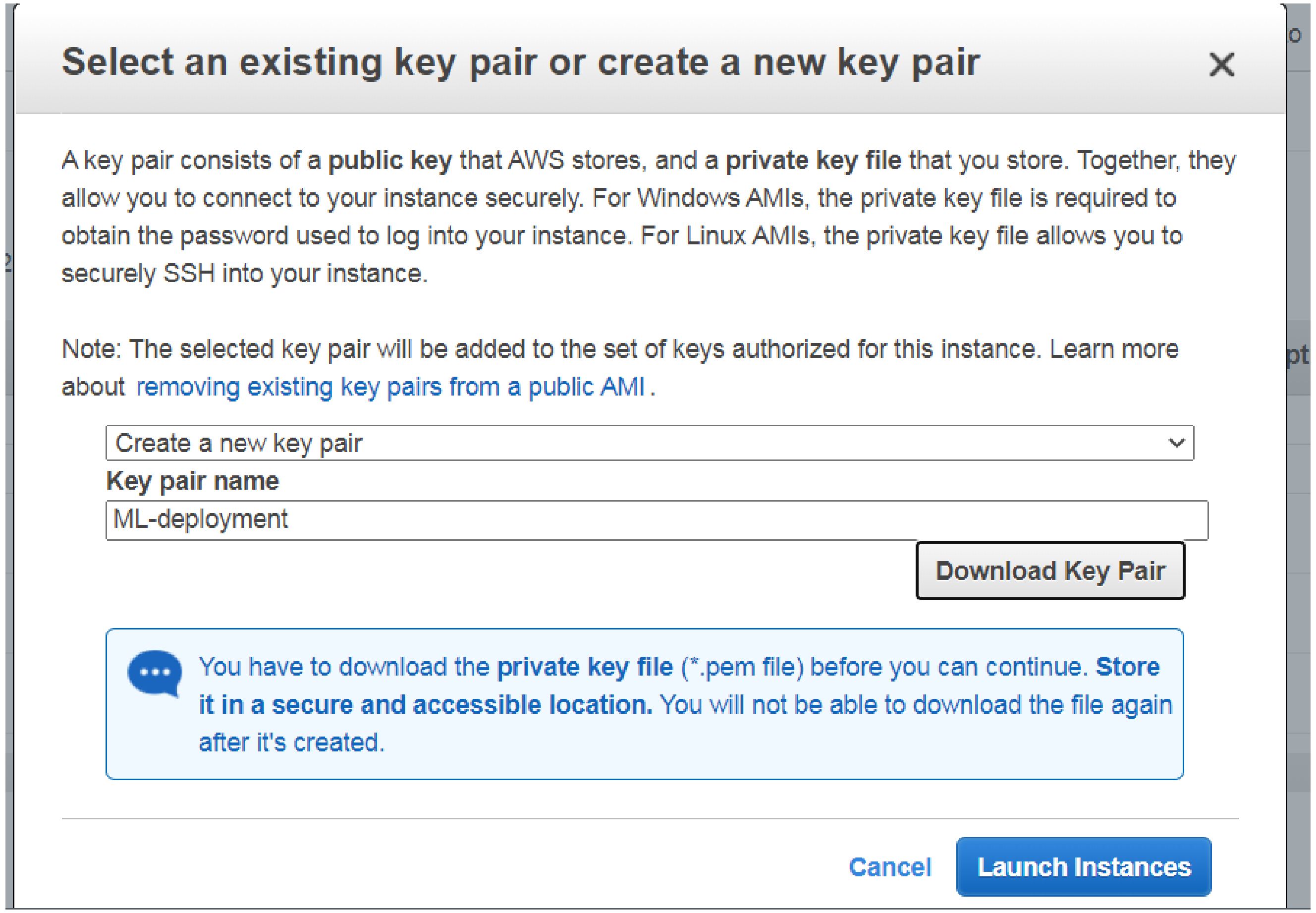
Description: launch-wizard-2 created 2021-03-02T13:58:15.436+05:30

Type	Protocol	Port Range	Source		Description
All traffic	All	0 - 65535	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Create a new key pair and download the PEM file



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

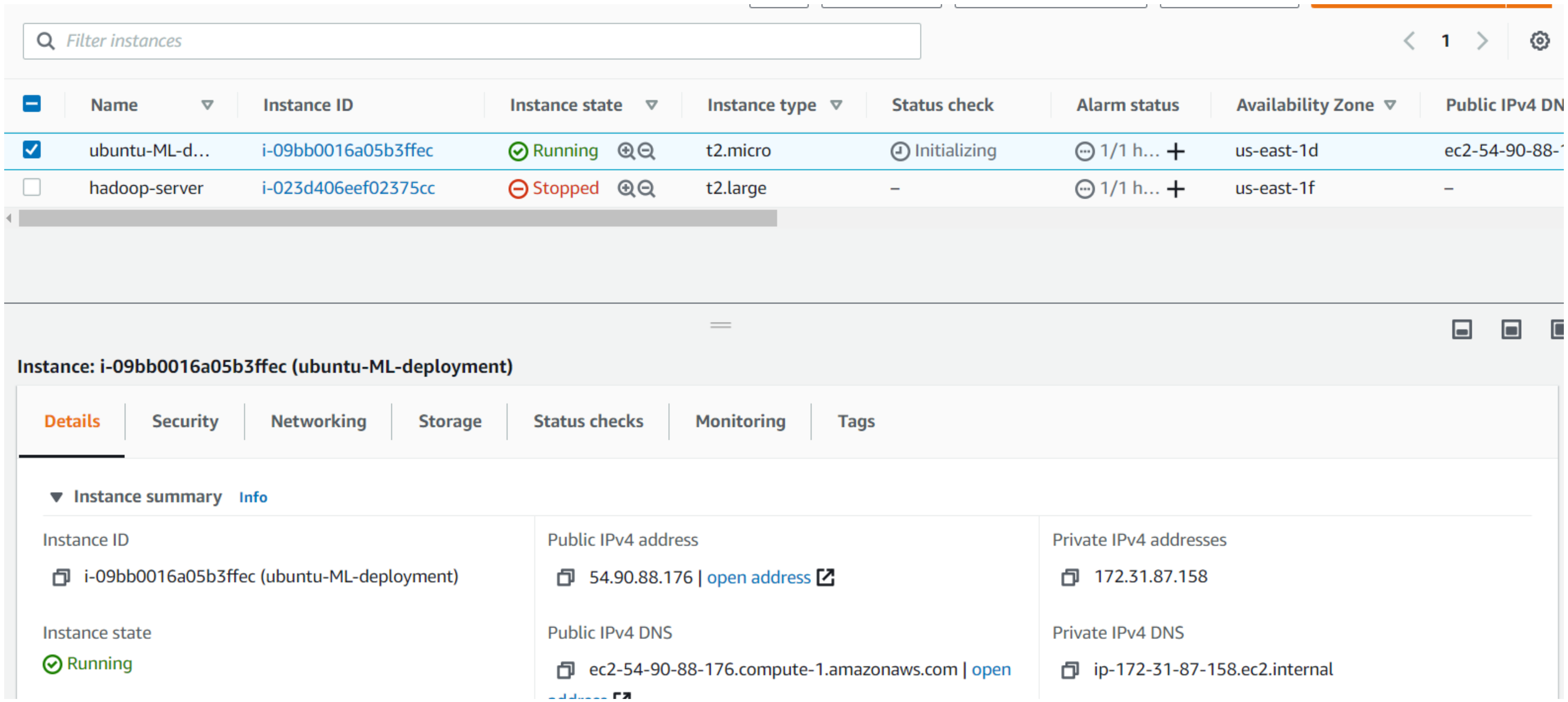
Create a new key pair

Key pair name

ML-deployment

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Launched instance



Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
ubuntu-ML-d…	i-09bb0016a05b3ffec	Running	t2.micro	Initializing	1/1 h…	us-east-1d	ec2-54-90-88-
hadoop-server	i-023d406eef02375cc	Stopped	t2.large	–	1/1 h…	us-east-1f	–

Instance: i-09bb0016a05b3ffec (ubuntu-ML-deployment)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

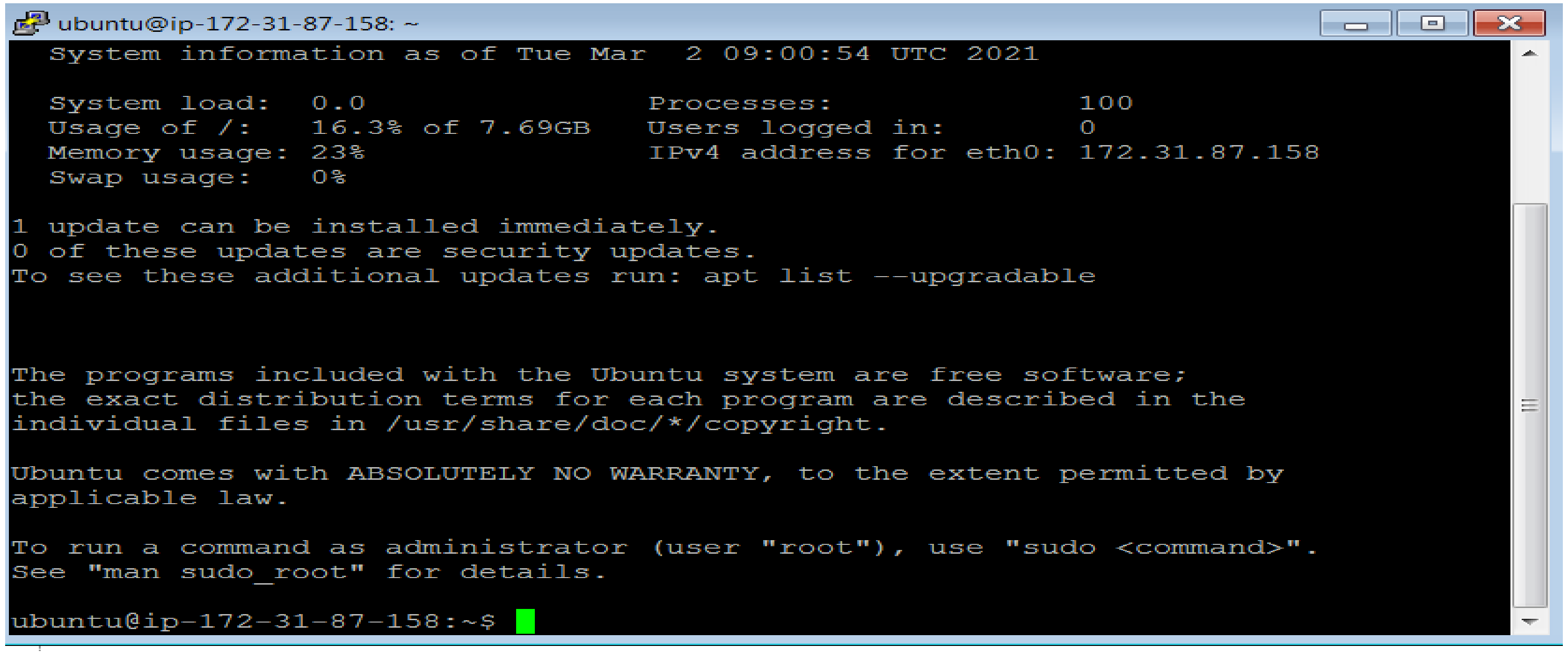
Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-09bb0016a05b3ffec (ubuntu-ML-deployment)	54.90.88.176 | open address	172.31.87.158
Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-54-90-88-176.compute-1.amazonaws.com | open address	ip-172-31-87-158.ec2.internal

Generate PPK for windows.
SSH in to EC2 instance with ubuntu as user name

- From PEM file use putty gen and generate ppk file
- If linux use

ssh –i ML-deployment.pem ubuntu@54.90.88.176



ubuntu@ip-172-31-87-158: ~

```
  System information as of Tue Mar  2 09:00:54 UTC 2021

  System load:  0.0                Processes:             100
  Usage of /:   16.3% of 7.69GB    Users logged in:       0
  Memory usage: 23%                IPv4 address for eth0: 172.31.87.158
  Swap usage:   0%

1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-87-158:~$
```

Install required libraries. Run the following commands to verify and install the required software

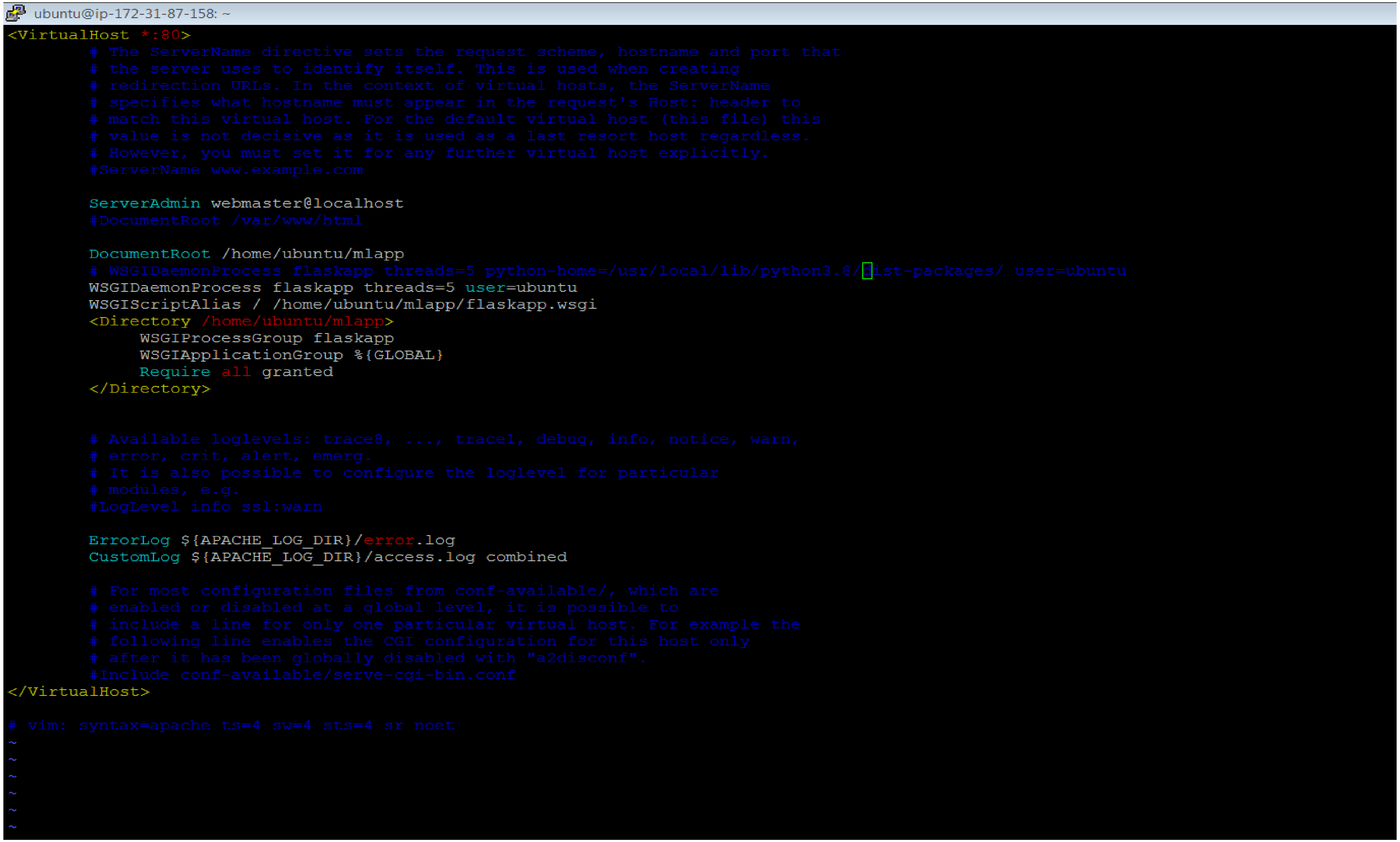
- Sudo apt-get update
- ubuntu@ip-172-31-87-158:~$ python3 -V
- ubuntu@ip-172-31-87-158:~$ curl -O https://bootstrap.pypa.io/get-pip.py
- Sudo python3 get-pip.py
- Sudo pip install flask
- Sudo pip install flask_cors
- Sudo apt-get install apache2
- Sudo pip install sklearn
- sudo apt-get install libapache2-mod-wsgi-py3

Configure the document root

- Sudo vi /etc/apache2/sites-enabled/000-default.conf

- DocumentRoot /home/ubuntu/mlapp

```
# do not set python-home if the python comes with default installation with ubuntu Debian flavor
# WSGIDaemonProcess flaskapp threads=5 python-
# home=/usr/local/lib/python3.8/dst-packages/ user=ubuntu
 WSGIDaemonProcess flaskapp threads=5 user=ubuntu
 WSGIScriptAlias / /home/ubuntu/mlapp/flaskapp.wsgi
•         <Directory /home/ubuntu/mlapp>
•             WSGIProcessGroup flaskapp
•             WSGIApplicationGroup %{GLOBAL}
•             Require all granted
•         </Directory>
```

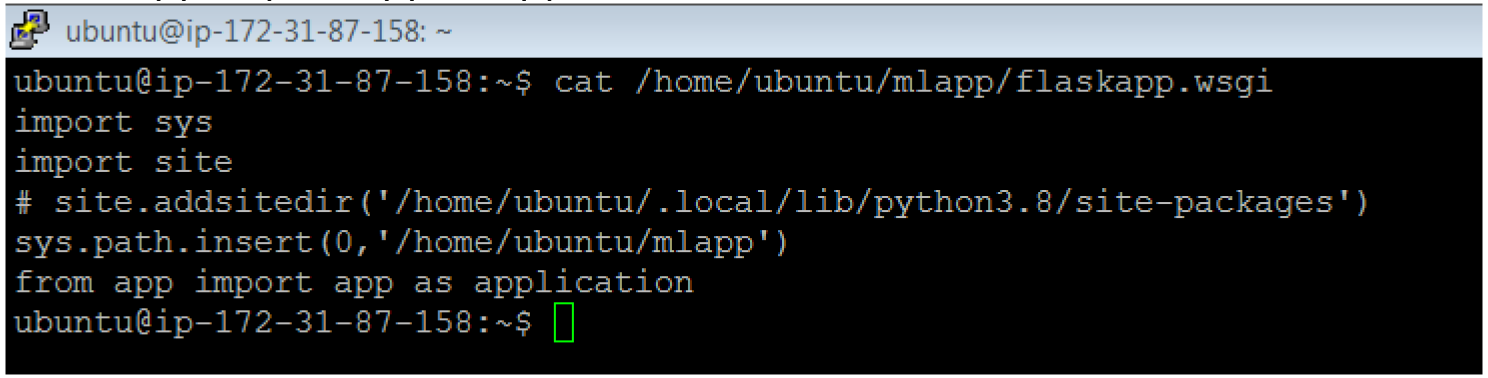

ubuntu@ip-172-31-87-158: ~

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        #DocumentRoot /var/www/html

        DocumentRoot /home/ubuntu/mlapp
        # WSGIDaemonProcess flaskapp threads=5 python-home=/usr/local/lib/python3.8/dist-packages/ user=ubuntu
        WSGIDaemonProcess flaskapp threads=5 user=ubuntu
        WSGIScriptAlias / /home/ubuntu/mlapp/flaskapp.wsgi
        <Directory /home/ubuntu/mlapp>
             WSGIProcessGroup flaskapp
             WSGIApplicationGroup %{GLOBAL}
             Require all granted
        </Directory>


        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
~
~
~
~
~
```

Create flaskapp.wsgi file in /home/ubuntu/mlapp

- Vi flaskapp.wsgi and paste following code

```
import sys
import site
# The below code is not required if the ubuntu Debian comes with python installed as default and do not set
# addsitedir as it was not installed using pip
# site.addsitedir('/home/ubuntu/.local/lib/python3.5/site-packages')
sys.path.insert(0,'/home/ubuntu/mlapp')
from app import app as application
```



```
ubuntu@ip-172-31-87-158:~$ cat /home/ubuntu/mlapp/flaskapp.wsgi
import sys
import site
# site.addsitedir('/home/ubuntu/.local/lib/python3.8/site-packages')
sys.path.insert(0,'/home/ubuntu/mlapp')
from app import app as application
ubuntu@ip-172-31-87-158:~$
```

Copy the app.py and marriage_age_predic_model.ml

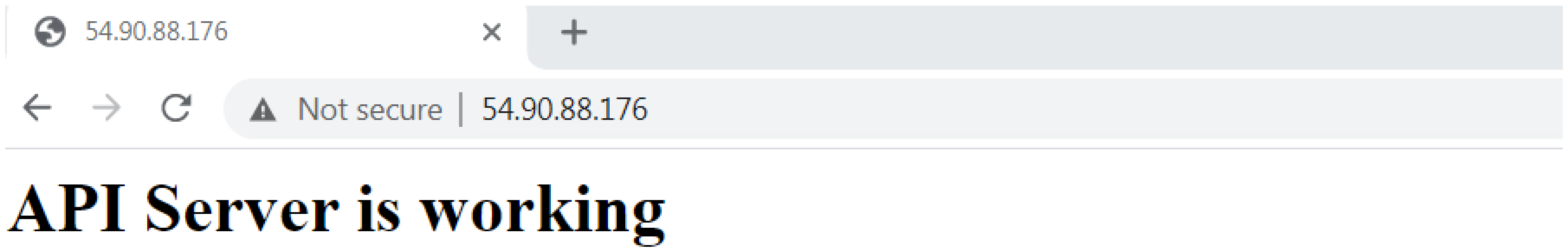
- Copy using winscp
- If on linux use scp command

```
scp –I <pem file location local> <file> ubuntu@54.90.88.176:/home/ubuntu/mlapp
```

Then restart apache server

```
Sudo apachectl restart
```

Default web page



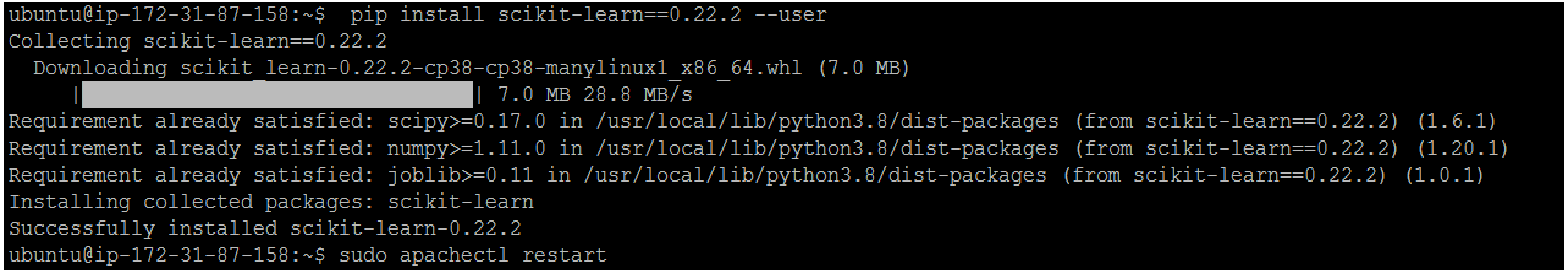
API Server is working

App.py

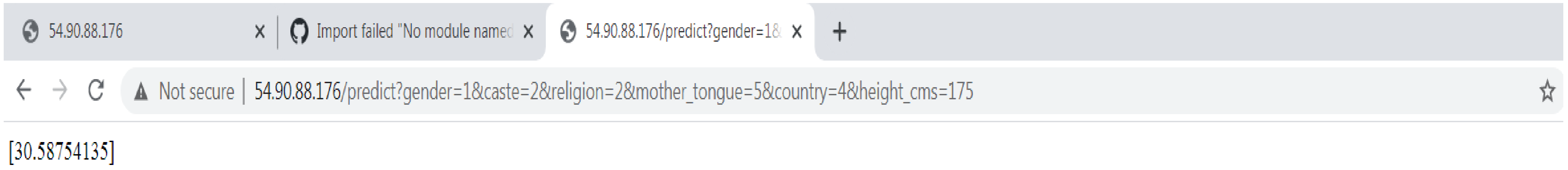
```
• #!/usr/bin/env python
• import flask
• from flask import request
• app=flask.Flask(__name__)
• from flask_cors import CORS
• CORS(app)
• @app.route('/')
• def default():
•     return '<h1> API Server is working </h1>'
• @app.route('/predict')
• def predict():
•     # sklearn compatible with python 3.8 import joblib directly
•     # from sklearn.externals import joblib
•     import joblib
•     model=joblib.load(‘home/ubuntu/mlapp/marriage_age_predic_model.ml')
•     #age_predict=model.predict([[1,2,5,6,5,175]])
•     age_predict=model.predict([[request.args['gender'],
•                               request.args['religion'],
•                               request.args['caste'],
•                               request.args['mother_tongue'],
•                               request.args['country'],
•                               request.args['height_cms']
•                               ]])
•     return str(age_predict)

• # disable debug in production environment
• # app.run(debug=True)
```

Browse predict path.
http://54.90.88.176/predict?gender=1&caste=2&religion=2&mother_tongue=5&country=4&height_cms=175

Reinstall the scikit learn if throws the error ModuleNotFoundError: No module named 'sklearn.ensemble.forest'
pip install scikit-learn==0.22.2 --user



```
ubuntu@ip-172-31-87-158:~$  pip install scikit-learn==0.22.2 --user
Collecting scikit-learn==0.22.2
  Downloading scikit_learn-0.22.2-cp38-cp38-manylinux1_x86_64.whl (7.0 MB)
     |████████████████████████████████| 7.0 MB 28.8 MB/s
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn==0.22.2) (1.6.1)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn==0.22.2) (1.20.1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.8/dist-packages (from scikit-learn==0.22.2) (1.0.1)
Installing collected packages: scikit-learn
Successfully installed scikit-learn-0.22.2
ubuntu@ip-172-31-87-158:~$ sudo apachectl restart
```


54.90.88.176
Import failed "No module named
54.90.88.176/predict?gender=1&
Not secure

54.90.88.176/predict?gender=1&caste=2&religion=2&mother_tongue=5&country=4&height_cms=175

```
[30.58754135]
```