

Class 2 report

Step 1: Deploy a EC2 instance allowing SSH access on AWS using terraform.

WelcomehelloWorld.pymain.tfreadme.md

lab1 > terraformSetup > main.tf > resource "aws_security_group" "allow_http" > egress > # from_port

```
1 provider "aws" {
2     region = "ap-southeast-1" // Replace with your AWS region
3     profile = "default"
4 }
5
6 resource "aws_security_group" "allow_http" {
7     name = "allow_http"
8     description = "Allow HTTP inbound traffic"
9
10    ingress {
11        from_port = 80
12        to_port = 80
13        protocol = "tcp"
14        cidr_blocks = ["0.0.0.0/0"]
15    }
16    ingress {
17        from_port = 22
18        to_port = 22
19        protocol = "tcp"
20        cidr_blocks = ["0.0.0.0/0"]
21    }
22    egress {
23        from_port = 0
24        to_port = 0
25        protocol = "-1"
26        cidr_blocks = ["0.0.0.0/0"]
27    }
28 }
29
30 resource "aws_instance" "web" {
```

Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	web-server	i-0bce87a109a199e69	Running	t2.micro	2/2 checks passed	View alarms	ap-southeast-1a	ec2-54-254-235-16

Step 2: Connect to instance using putty.

```
35      key_name = "cloudSec"
Step 1:
ubuntu@ip-172-31-47-27: ~
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-47-27:~$
```

Step 3: Install Python Setup the environment.

```
sudo apt install python 3
```

```
ubuntu@ip-172-31-47-27: ~
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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-172-31-47-27:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-47-27:~$
```

Step 4: Install Flask.

pip install flask

```
ubuntu@ip-172-31-47-27: ~/helloApp
Downloading blinker-1.7.0-py3-none-any.whl (13 kB)
Collecting click>=8.1.3
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
----- 97.9/97.9 KB 10.8 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.3-py3-none-any.whl (133 kB)
----- 133.2/133.2 KB 16.5 MB/s eta 0:00:00
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.2-py3-none-any.whl (226 kB)
----- 226.8/226.8 KB 25.6 MB/s eta 0:00:00
Requirement already satisfied: MarkupSafe>=2.0 in /usr/lib/python3/dist-packages
(from Jinja2>=3.1.2->flask) (2.0.1)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.5-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
WARNING: The script flask is installed in '/home/ubuntu/.local/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.
Successfully installed Jinja2-3.1.3 MarkupSafe-2.1.5 Werkzeug-3.0.2 blinker-1.7.0 click-8.1.7 flask-3.0.3 itsdangerous-2.2.0
```

Step 5: Create hello world web app.

```
ubuntu@ip-172-31-47-27: ~/helloApp$ nano hello.py
ubuntu@ip-172-31-47-27:~/helloApp$ python3 hello.py
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.47.27:5000
Press CTRL+C to quit
^Cubuntu@ip-172-31-47-27:~/helloApp$ curl http://127.0.0.1:5000
curl: (7) Failed to connect to 127.0.0.1 port 5000 after 0 ms: Co
d
ubuntu@ip-172-31-47-27:~/helloApp$ python3 hello.py
* Serving Flask app 'hello'
* Debug mode: off
WARNING: This is a development server. Do not use it in a product
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.47.27:5000
Press CTRL+C to quit
127.0.0.1 - - [26/Apr/2024 04:01:05] "GET / HTTP/1.1" 200 -

* Support: https://ubuntu.com/pro
System information as of Fri Apr 26 04:01:00 UTC 2024
System load: 0.080078125 Processes: 100
Usage of /: 26.4% of 7.57GB Users logged in: 1
Memory usage: 24% IPv4 address for eth0: 172.31.47.27
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
54 updates can be applied immediately.
37 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
*** System restart required ***
Last login: Fri Apr 26 03:39:35 2024 from 58.186.99.36
ubuntu@ip-172-31-47-27:~$ curl http://127.0.0.1:5000
Hello, World!ubuntu@ip-172-31-47-27:~$
```

```
ubuntu@ip-172-31-47-27: ~/helloApp
GNU nano 6.2 hello.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'Hello, World!'

app.run(host='0.0.0.0', port=5000)
```

[Read 9 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Undo

Step 6: Install and configure nginx as reverse proxy.

Visual Studio Code interface showing the installation and configuration of nginx as a reverse proxy. The Explorer view shows the project structure with files like pic1.png through pic7.png, terraformSetup, and helloWorld.py. The main editor shows a README.md file with steps 1 through 6. Step 6 is highlighted: "Step 6: Install and configure nginx". The terminal view shows the command `sudo apt install python3` and the output `python3 is already the default python3. The command 'python3 --help' provides information about the python3 command-line interface.`

Create a file with the name of the EC2 instance's IP address:

The content inside the file:

Restart Nginx server and check the status of server.

```

ubuntu@ip-172-31-47-27: /etc/nginx/sites-enabled
default
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ nano "13.228.77.216"
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ sudo chmod 777 .
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ nano "13.228.77.216"
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ nano "13.228.77.216"
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ sudo systemctl restart nginx
ubuntu@ip-172-31-47-27:/etc/nginx/sites-enabled$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: >
   Active: active (running) since Fri 2024-04-26 08:49:07 UTC; 10s ago
     Docs: man:nginx(8)
  Process: 19028 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proc >
  Process: 19031 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (>
 Main PID: 19032 (nginx)
    Tasks: 2 (limit: 1121)
   Memory: 3.8M
      CPU: 30ms
   CGroup: /system.slice/nginx.service
           └─19032 "nginx: master process /usr/sbin/nginx -g daemon on; maste >
             └─19033 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" >

Apr 26 08:49:07 ip-172-31-47-27 systemd[1]: Starting A high performance web ser >
Apr 26 08:49:07 ip-172-31-47-27 systemd[1]: Started A high performance web serv >
lines 1-16/16 (END)

```

Step 7: Install Python virtual environment and gunicorn.

5 / 7

```
pip install gunicorn
```

Step 9: Create WSGI Entry Point for Gunicorn.

```
~/helloApp$ vim wsgi.py
```

```
from hello import app
if __name__ == "__main__":
    app.run()
```

Activate virtual environment and create entry point for the app:

```
source venv/bin/activate ~/helloApp$ gunicorn --bind 0.0.0.0:5000 wsgi:app
```

Step 10: Check the result:

```
ubuntu@ip-172-31-47-27: ~/helloApp
compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

17 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Apr 26 14:36:28 2024 from 183.80.8.81
ubuntu@ip-172-31-47-27:~$ ls
helloApp
ubuntu@ip-172-31-47-27:~$ cd helloApp/
ubuntu@ip-172-31-47-27:~/helloApp$ ls
__pycache__  hello.py  venv  wsgi.py
ubuntu@ip-172-31-47-27:~/helloApp$ vim wsgi.py
ubuntu@ip-172-31-47-27:~/helloApp$ vim wsgi.py
ubuntu@ip-172-31-47-27:~/helloApp$ ls
__pycache__  hello.py  venv  wsgi.py
ubuntu@ip-172-31-47-27:~/helloApp$
```

13.228.77.216

Import bookmarks... ChatGPT Lịch phòng học FREN... Materials - Class B1 - ... Data Link Layer - Com... product-management... 1.2. Lâm giao diện - na... vxunderground/Malw... rootkit-io/awesome... Other Bookmarks

Hello, World!

```
10:34 PM
4/26/2024

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

    revoke_rules_on_delete = false
    tags_all                = {}
    vpc_id                  = "vpc-0b03867ced057d0ed"
}

Outputs:

instance_public_dns = "ec2-13-228-77-216.ap-southeast-1.compute.amazonaws.com"
instance_public_ip  = "13.228.77.216"
PS C:\Users\DuyAnh\Documents\Code\CloudSecurity\lab1\terraformSetup>
```