AWS DOCUMENT
Simran Malik
Shreya Birthare

1)Start aws lab
2)Copy aws credentials to ~/.aws
3)Download labsuser.pem file and saved to working directory
4)Run aws configure

```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % aws configure
AWS Access Key ID [****************IN76]:
AWS Secret Access Key [****************iEcW]:
Default region name [us-east-1]: us-east-1
Default output format [json]: json
```

5)Start ec2 instance using
aws ec2 run-instances --image-id ami-0d73480446600f555 --instance-type t2.micro --key-name vockey > instance.json

6)The instance.json file will contain the details of your instance. Copy the InstanceId field. Now in your terminal check the status of your instance using the following command:
This gives various details about the instance type, who is running it, and the public DNS name where you can access the instance

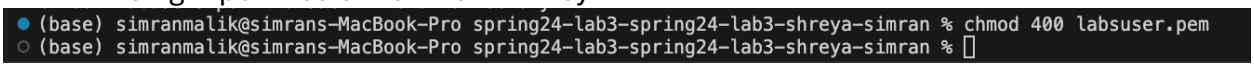
```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % aws ec2 describe-instances --instance-id "i-007a1548b17df6f77"
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0d73480446600f555",
                    "InstanceId": "i-007a1548b17df6f77",
                    "InstanceType": "t2.micro",
                    "KeyName": "vockey",
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS


```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0d73480446600f555",
                    "InstanceId": "i-007a1548b17df6f77",
                    "InstanceType": "t2.micro",
                    "KeyName": "vockey",
                    "LaunchTime": "2024-05-01T22:51:52+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1d",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-24-99.ec2.internal",
                    "PrivateIpAddress": "172.31.24.99",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-34-229-153-243.compute-1.amazonaws.com",
                    "PublicIpAddress": "34.229.153.243",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-075735a3b80edc093",
:...skipping...
{
    "Reservations": [
        {
```

7)Access EC@ instance via SSH

- Before running ssh command there are two things we need to do. The first is to set the right permission for the PEM key.

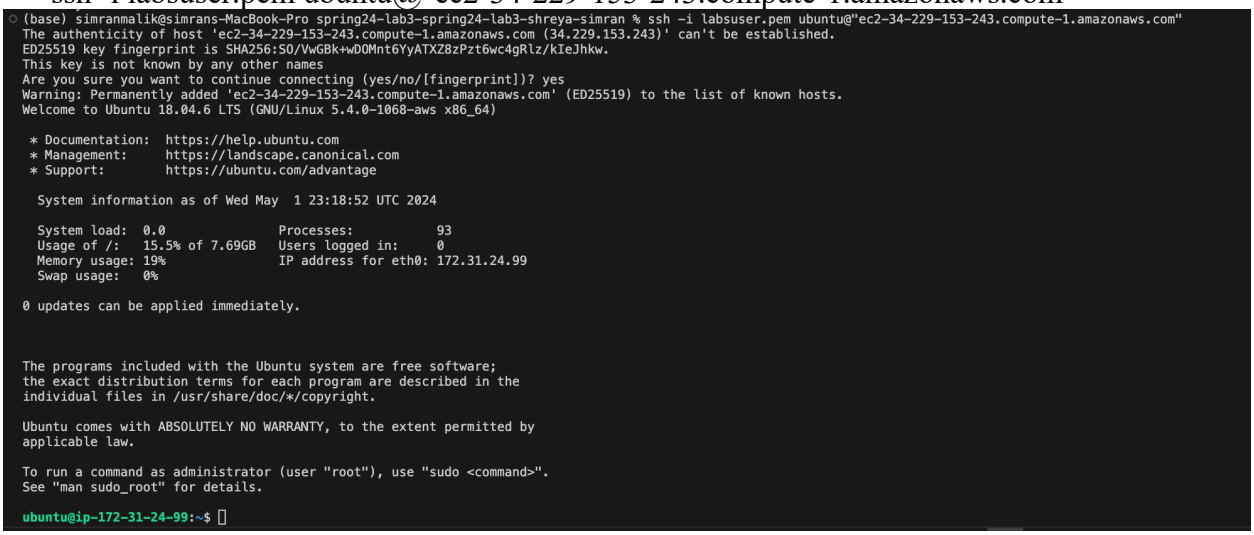


- The other thing we need to do is to authorize port 22 (used by ssh) in the default security group (you can think of security group as a virtual firewall).

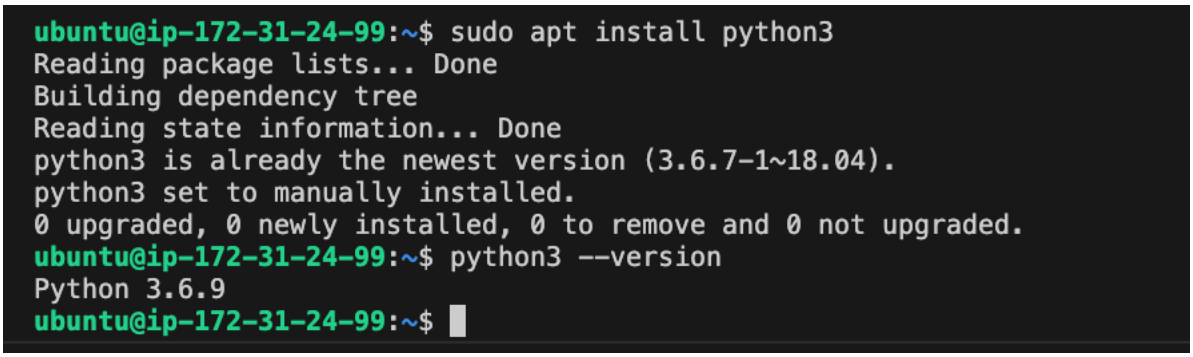
- Public dns: "ec2-34-229-153-243.compute-1.amazonaws.com"

- Now that everything is ready, we can connect to our instance via ssh:

ssh -i labsuser.pem ubuntu@"ec2-34-229-153-243.compute-1.amazonaws.com"



8) Install Softwares



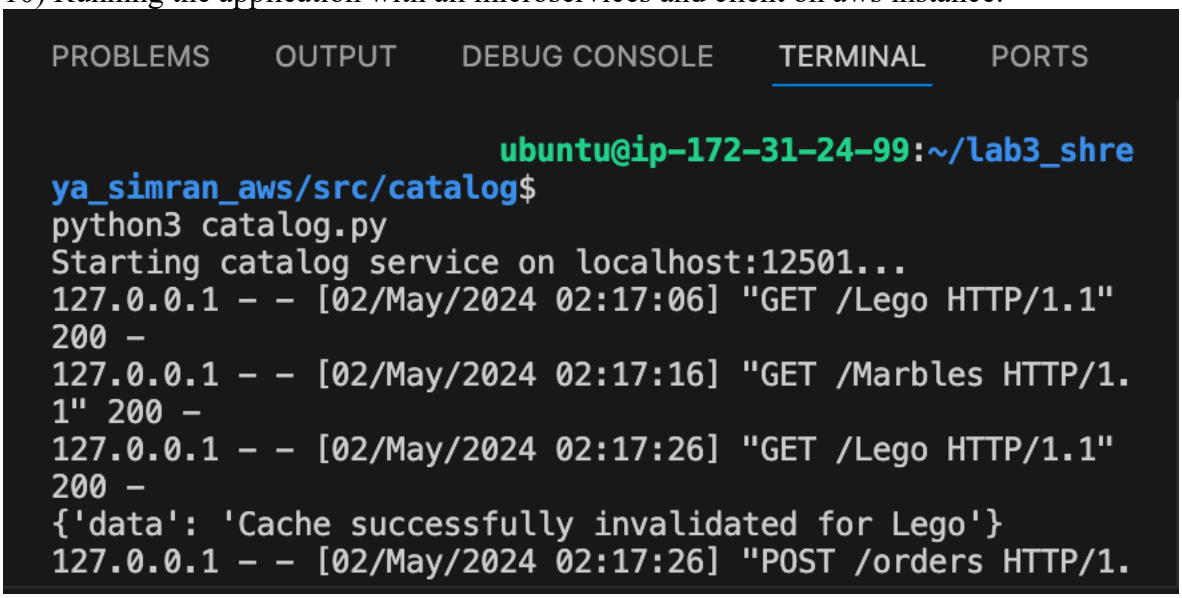
9) Now we will use scp to transfer all files/contents of out current directory to /home/ubuntu directory of our ubuntu image

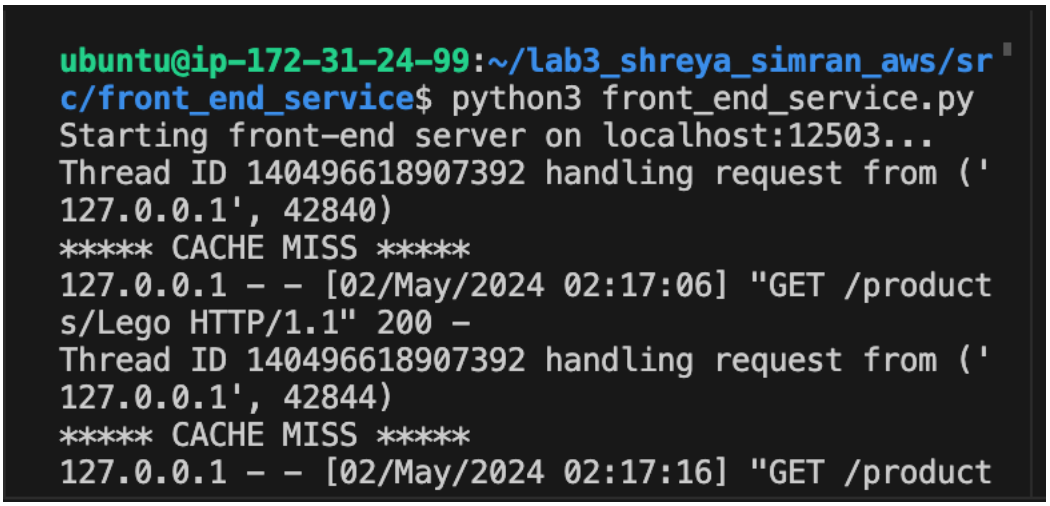
```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simra
n % scp -i labsuser.pem -r ./ ubuntu@"34.229.153.243":/home/ubuntu
The authenticity of host '34.229.153.243 (34.229.153.243)' can't be established
.
ED25519 key fingerprint is SHA256:SO/VwGBk+wDOMnt6YyATXZ8zPzt6wc4gRlz/kIeJhkw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:17: ec2-34-229-153-243.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.229.153.243' (ED25519) to the list of known host
s.
.DS_Store                                     100% 6148    71.8KB/s   00:00
instance.json                                 100% 4355    48.6KB/s   00:00
README.md                                     100%   13KB 145.4KB/s   00:00
unitTests.cpython-310.pyc                     100% 6582    70.1KB/s   00:00
unitTests.cpython-311.pyc                     100% 3526    39.5KB/s   00:00
unitTests.py                                  100% 5979    67.9KB/s   00:00
.keep                                         100%    0     0.0KB/s   00:00
labsuser.pem                                  100% 1678    15.9KB/s   00:00
config                                        100%  350     4.0KB/s   00:00
fd0d828114c8367b0fca01663eb194c2425a37        100% 1525    17.3KB/s   00:00
97e24fc25be5a9f388aaba57e63532c452af17        100%  147     1.8KB/s   00:00
3b4aeb6a7adcfe0e7e25c8e280efde40da7538        100% 2007    24.4KB/s   00:00
c849a66e57af89ea7da1f941fb164aa78f1506        100% 1128    13.3KB/s   00:00
d65e94d14af249be1d1c26a4d4d4a75ca251e0        100%   86     1.0KB/s   00:00
6d13689f680c332cc9e087ecb1672725fe4727        100%   94     1.1KB/s   00:00
c0921041605a94f70556a1f737b3a9aaf44e82        100% 1506    17.6KB/s   00:00
7463e1a2292fd45b6db86b71a1c63055b9f788        100%  205     2.4KB/s   00:00
73e58298fa330128a845e5f553f7dd753c93f9        100%  754     9.0KB/s   00:00
af38c066e68731e6f8aa6712e0e2d150b1aff1        100%  180     2.0KB/s   00:00
a00e28d1588e593b43dccb62f9532e7b6f03ae        100%  138     1.6KB/s   00:00
321627ed847e27546e88dcbbab84c902bbcef9        100%   56     0.7KB/s   00:00
ecbcee5fb2a53a5779200594c7de60b1472814        100%  184     2.2KB/s   00:00
8260b14f0abb83f7ff8a89f8c3faf61770166a        100%  193     2.3KB/s   00:00
pack-9d460042aebbd340d762d06b70a2cc0ec07ff51  100%   44KB 240.2KB/s   00:00
pack-9d460042aebbd340d762d06b70a2cc0ec07ff51  100% 4572    53.1KB/s   00:00
```

10) Transfer was successful, all files and folders are now available in the home directory of ubuntu vm. We can move all the files and folders to a new folder lab3_shreya_simran_aws

```
ubuntu@ip-172-31-24-99:~$ ls
README.md  instance.json  labsuser.pem  src  testing
ubuntu@ip-172-31-24-99:~$ mkdir lab3_shreya_simran_aws
ubuntu@ip-172-31-24-99:~$ mv README.md instance.json labsuser.pem src testing lab
3_shreya_simran_aws/
ubuntu@ip-172-31-24-99:~$ ls
lab3_shreya_simran_aws
ubuntu@ip-172-31-24-99:~$ cd lab3_shreya_simran_aws
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws$ ls
README.md  instance.json  labsuser.pem  src  testing
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws$
```

10) Running the application with all microservices and client on aws instance:





```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ python3 order.py
Error requesting missed orders from replica 2: HTTPConnectionPool(host='localhost', port=12504): Max retries exceeded with url: /missed_order (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f75c7317f40>: Failed to establish a new connection: [Errno 111] Connection refused'))
Error requesting missed orders from replica 1: HTTPConnectionPool(host='localhost', port=12502): Max retries exceeded with url: /missed_order (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f75c72f16d0>: Failed to establish a new connection: [Errno 111] Connection refused'))
Failed to receive missed orders from any replica
Starting order service on localhost:12505...
127.0.0.1 - - [02/May/2024 02:14:35] "POST /missed_order HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 02:16:17] "POST /missed_order HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 02:17:26] "GET /health HTTP/1.1" 200 -
i am now leader
Lego is in stock, placing order for 3 quantity
Successfully propagated to http://localhost:12502/replicate_order
Successfully propagated to http://localhost:12504/replicate_order
127.0.0.1 - - [02/May/2024 02:17:26] "POST /orders HTTP/1.1" 200 -
127.0.0.1 - - [02/May/2024 02:18:16] "GET /health HTTP/1.1" 200 -
i am now leader
Tux is in stock, placing order for 9 quantity
Successfully propagated to http://localhost:12502/replicate_order
Successfully propagated to http://localhost:12504/replicate_order
```

```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ ex
port Replica_id=1
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ export
 ORDER_HOST=localhost
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ export
 ORDER_LISTENING_PORT=12502
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ python
3 order.py
Nothing is missed
Starting order service on localhost:12502...
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost',
 'port': 12505}
127.0.0.1 - - [02/May/2024 02:17:26] "POST /notify_leader_info_to
_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:17:26] "POST /replicate_order HTTP/
1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost',
 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:16] "POST /notify_leader_info_to
_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:16] "POST /replicate_order HTTP/
1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localhost',
 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:26] "POST /notify_leader_info_to
_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:26] "POST /replicate_order HTTP/
1.1" 200 -
```

```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ ex
port Replica_id=2
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ ex
port ORDER_HOST=localhost
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ ex
port ORDER_LISTENING_PORT=12504
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order$ py
thon3 order.py
Nothing is missed
Starting order service on localhost:12504...
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localho
st', 'port': 12505}
127.0.0.1 - - [02/May/2024 02:17:26] "POST /notify_leader_inf
o_to_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:17:26] "POST /replicate_order H
TTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localho
st', 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:16] "POST /notify_leader_inf
o_to_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:16] "POST /replicate_order H
TTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localho
st', 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:26] "POST /notify_leader_inf
o_to_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:26] "POST /replicate_order H
TTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localho
st', 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:36] "POST /notify_leader_inf
o_to_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:36] "POST /replicate_order H
TTP/1.1" 200 -
Found Leader! Leader ID: 3 and Leader Info: {'host': 'localho
st', 'port': 12505}
127.0.0.1 - - [02/May/2024 02:18:46] "POST /notify_leader_inf
o_to_replica HTTP/1.1" 200 -
Order replicated by leader ID: localhost:12505
127.0.0.1 - - [02/May/2024 02:18:46] "POST /replicate_order H
TTP/1.1" 200 -
```

```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src$ python3 clien
t.py
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, '
quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.
99, 'quantity': 100}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, '
quantity': 100}}
placing order for Lego, 3
Order result for Lego: {'data': {'order_number': 0}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.
99, 'quantity': 100}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.
99, 'quantity': 100}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.9
9, 'quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.
99, 'quantity': 100}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'qu
antity': 100}}
placing order for Tux, 9
Order result for Tux: {'data': {'order_number': 1}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, '
quantity': 97}}
placing order for Lego, 5
Order result for Lego: {'data': {'order_number': 2}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75
.99, 'quantity': 100}}
placing order for Giraffe, 7
Order result for Giraffe: {'data': {'order_number': 3}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.
99, 'quantity': 100}}
placing order for Marbles, 1
Order result for Marbles: {'data': {'order_number': 4}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price':
10.99, 'quantity': 100}}
placing order for Monopoly, 4
Order result for Monopoly: {'data': {'order_number': 5}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.9
9, 'quantity': 100}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'qu
antity': 100}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75
.99, 'quantity': 93}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'qu
antity': 100}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'qu
antity': 100}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'qu
antity': 100}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.9
9, 'quantity': 100}}
placing order for Barbie, 2
Order result for Barbie: {'data': {'order_number': 6}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.
99, 'quantity': 100}}
placing order for Frisbee, 2
Order result for Frisbee: {'data': {'order_number': 7}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, '
quantity': 92}}
```

Change in catalog file

```
shreya_
simran_
aws/src
ubuntu@
ip-172-
31-24-9
9:~/lab
~/lab3_
shreya_
simran_
aws/src
ubuntu@
ip-172-
31-24-9
9:~/lab
3_shrey
a_simra
n_aws/s
rc$
```

```
  GNU nano 2.9.3                catalog.csv

name,price,quantity
Tux,15.99,91
Whale,25.99,98
Fox,12.99,100
Python,20.99,93
Barbie,55.99,98
Lego,45.99,82
Monopoly,10.99,96
Frisbee,5.99,98
Marbles,7.99,90
Giraffe,75.99,93



^G Get Help  ^O Write Out^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File^\ Replace   ^U Uncut Tex^T To Spell
```

```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src$ cd orde
r/order_data
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order/or
der_data$ ls
order_log_1.csv  order_log_2.csv  order_log_3.csv
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/src/order/or
der_data$
```

Order logs generated.

```
  GNU nano 2.9.3          order_log_1.csv

0,Lego,3
1,Tux,9
2,Lego,5
3,Giraffe,7
4,Marbles,1
5,Monopoly,4
6,Barbie,2
7,Frisbee,2
8,Marbles,1
9,Python,6
10,Python,1
11,Marbles,8
12,Lego,10
13,Whale,2

^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit     ^R Read File^\ Replace  ^U Uncut Tex^T To Spell
```

```
  GNU nano 2.9.3          order_log_2.csv

0,Lego,3
1,Tux,9
2,Lego,5
3,Giraffe,7
4,Marbles,1
5,Monopoly,4
6,Barbie,2
7,Frisbee,2
8,Marbles,1
9,Python,6
10,Python,1
11,Marbles,8
12,Lego,10
13,Whale,2
        [ Read 14 lines (Converted from DOS format) ]
^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit     ^R Read File^\ Replace  ^U Uncut Tex^T To Spell
```

```
  GNU nano 2.9.3          order_log_3.csv

0,Lego,3
1,Tux,9
2,Lego,5
3,Giraffe,7
4,Marbles,1
5,Monopoly,4
6,Barbie,2
7,Frisbee,2
8,Marbles,1
9,Python,6
10,Python,1
11,Marbles,8
12,Lego,10
13,Whale,2
        [ Read 14 lines (Converted from DOS format) ]
^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit     ^R Read File^\ Replace  ^U Uncut Tex^T To Spell
```

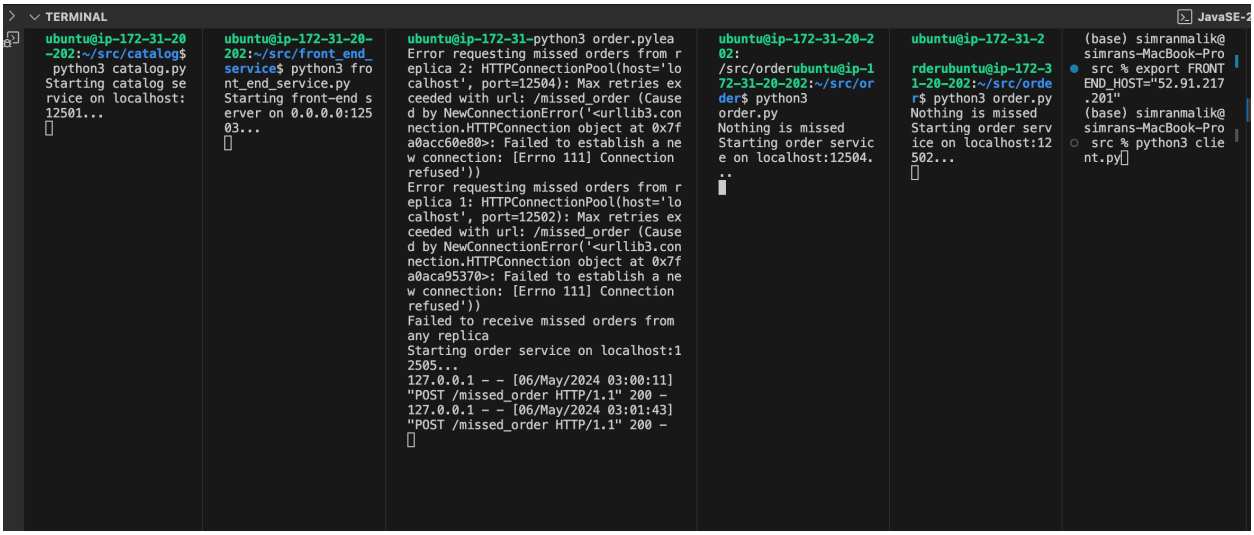
Running test cases

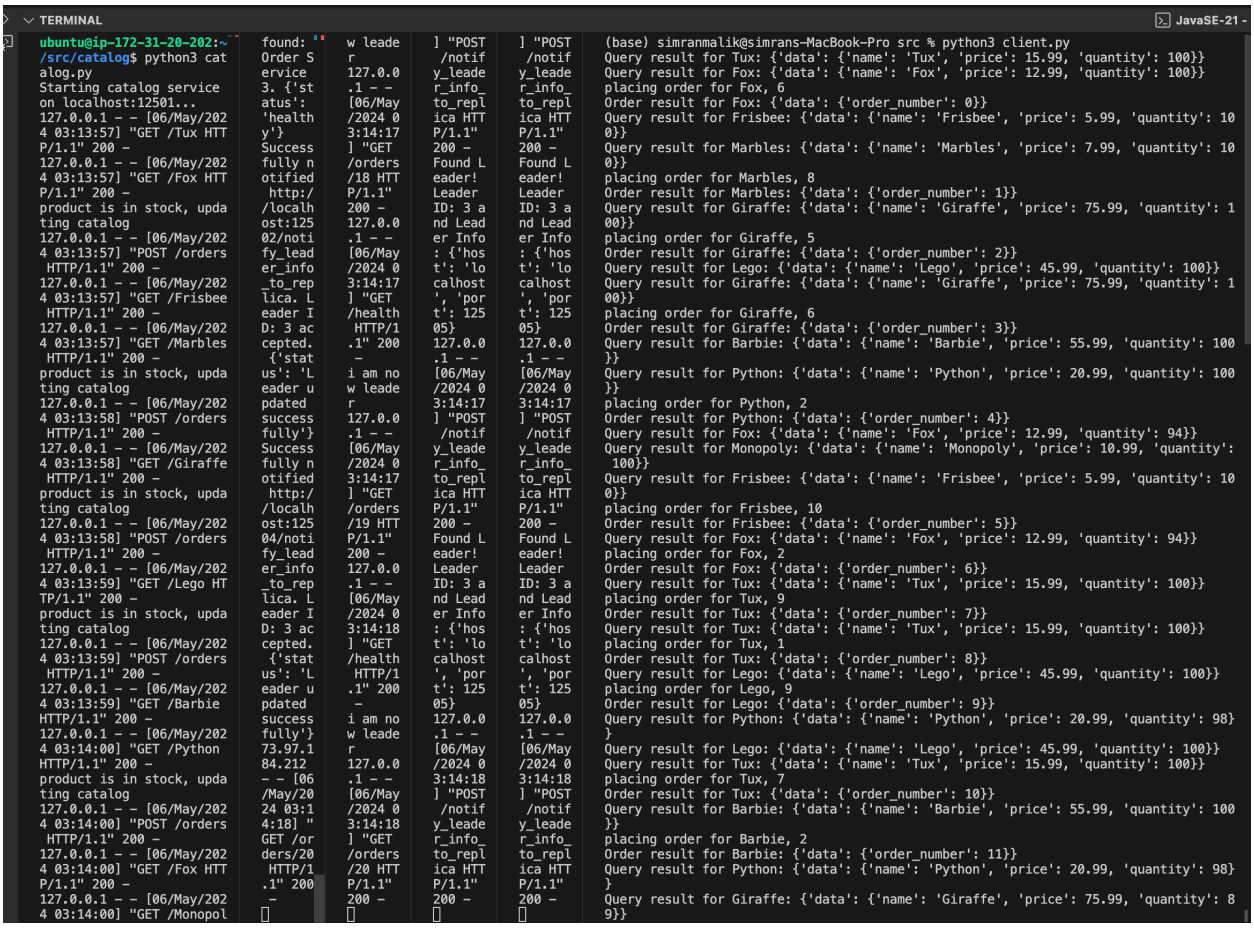
```
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/testing$ python3 -m
unittest unitTests
/usr/lib/python3/dist-packages/requests/models.py:177: DeprecationWa
rning: Using or importing the ABCs from 'collections' instead of fro
m 'collections.abc' is deprecated since Python 3.3, and in 3.9 it wi
ll stop working
  if isinstance(hook, collections.Callable):
...................
----------------------------------------------------------------------
--
Ran 19 tests in 0.130s

OK
ubuntu@ip-172-31-24-99:~/lab3_shreya_simran_aws/testing$
```

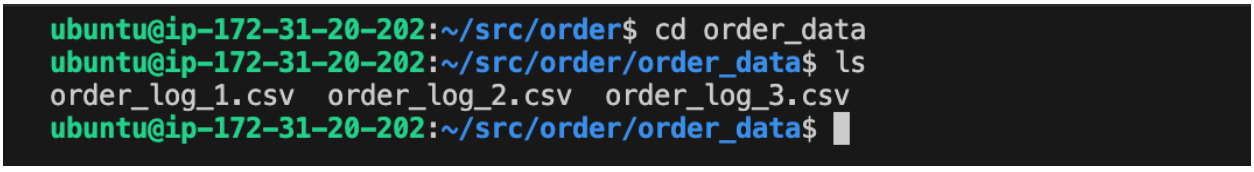
11) Allowing incoming and outgoing tcp/http traffic port 12503 -front end service port

```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % aws ec2 authorize-security-group-egress --group-id sg-0087a
1a48790086c6 --protocol tcp --port 12503 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-04322bf7af506c236",
            "GroupId": "sg-0087a1a48790086c6",
            "GroupOwnerId": "726975314709",
            "IsEgress": true,
            "IpProtocol": "tcp",
            "FromPort": 12503,
            "ToPort": 12503,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % aws ec2 authorize-security-group-ingress --group-id sg-0087
a1a48790086c6 --protocol tcp --port 12503 --cidr 0.0.0.0/0

{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0c41faca38593ae0a",
            "GroupId": "sg-0087a1a48790086c6",
            "GroupOwnerId": "726975314709",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 12503,
            "ToPort": 12503,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran %
```

12) Running the application with all microservices on aws instance, client on local machine:




TERMINAL
JavaSE-21
ubuntu@ip-172-31-20-202:~/src/catalog$ python3 catalog.py
Starting catalog service on localhost:12501...
(base) simranmalik@simrans-MacBook-Pro src % python3 client.py
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 100}}
placing order for Fox, 6
Order result for Fox: {'data': {'order_number': 0}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 100}}
Query result for Marbles: {'data': {'name': 'Marbles', 'price': 7.99, 'quantity': 100}}
placing order for Marbles, 8
Order result for Marbles: {'data': {'order_number': 1}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 100}}
placing order for Giraffe, 5
Order result for Giraffe: {'data': {'order_number': 2}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 100}}
placing order for Giraffe, 6
Order result for Giraffe: {'data': {'order_number': 3}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 100}}
placing order for Python, 2
Order result for Python: {'data': {'order_number': 4}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 94}}
Query result for Monopoly: {'data': {'name': 'Monopoly', 'price': 10.99, 'quantity': 100}}
Query result for Frisbee: {'data': {'name': 'Frisbee', 'price': 5.99, 'quantity': 100}}
placing order for Frisbee, 10
Order result for Frisbee: {'data': {'order_number': 5}}
Query result for Fox: {'data': {'name': 'Fox', 'price': 12.99, 'quantity': 94}}
placing order for Fox, 2
Order result for Fox: {'data': {'order_number': 6}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
placing order for Tux, 9
Order result for Tux: {'data': {'order_number': 7}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
placing order for Tux, 1
Order result for Tux: {'data': {'order_number': 8}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
placing order for Lego, 9
Order result for Lego: {'data': {'order_number': 9}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 98}}
Query result for Lego: {'data': {'name': 'Lego', 'price': 45.99, 'quantity': 100}}
Query result for Tux: {'data': {'name': 'Tux', 'price': 15.99, 'quantity': 100}}
placing order for Tux, 7
Order result for Tux: {'data': {'order_number': 10}}
Query result for Barbie: {'data': {'name': 'Barbie', 'price': 55.99, 'quantity': 100}}
placing order for Barbie, 2
Order result for Barbie: {'data': {'order_number': 11}}
Query result for Python: {'data': {'name': 'Python', 'price': 20.99, 'quantity': 98}}
Query result for Giraffe: {'data': {'name': 'Giraffe', 'price': 75.99, 'quantity': 89}}

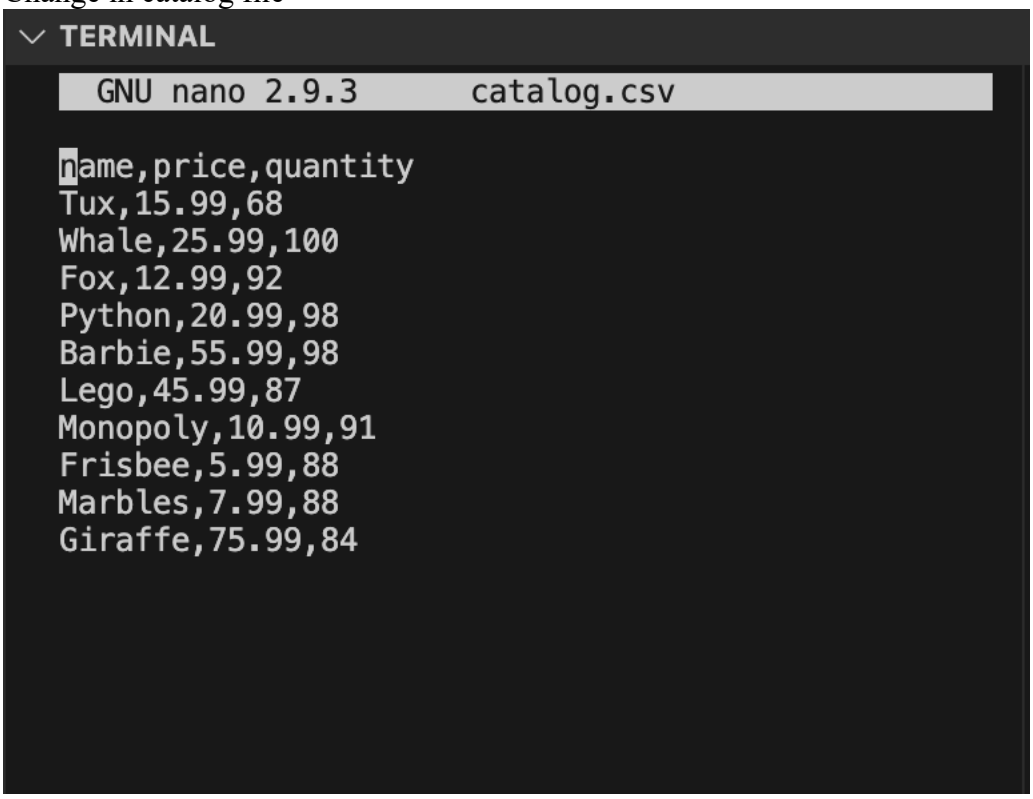

ubuntu@ip-172-31-20-202:~/src/order$ cd order_data
ubuntu@ip-172-31-20-202:~/src/order/order_data$ ls
order_log_1.csv order_log_2.csv order_log_3.csv
ubuntu@ip-172-31-20-202:~/src/order/order_data$

Order logs generated

```
  GNU nano 2.9.3                        order_log_2.csv

0,Fox,6
1,Marbles,8
2,Giraffe,5
3,Giraffe,6
4,Python,2
5,Frisbee,10
6,Fox,2
7,Tux,9
8,Tux,1
9,Lego,9
10,Tux,7
11,Barbie,2
12,Frisbee,2
13,Marbles,4
14,Giraffe,4
15,Monopoly,9
16,Lego,4
17,Tux,2
18,Tux,3
19,Giraffe,1
20,Tux,10

                [ Read 21 lines (Converted from DOS format) ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

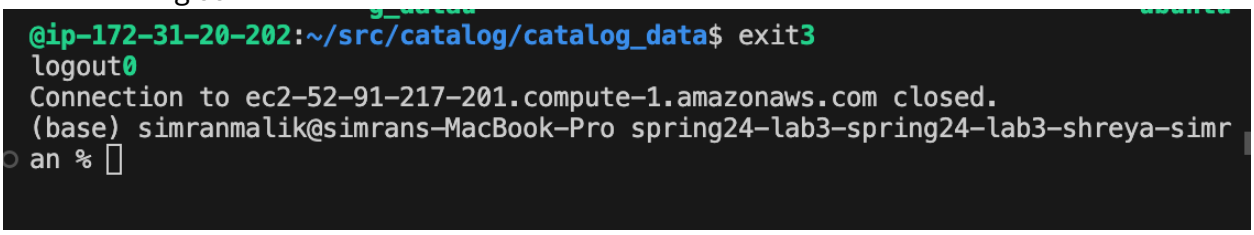
```
  GNU nano 2.9.3                  order_log_3.csv

0,Fox,6
1,Marbles,8
2,Giraffe,5
3,Giraffe,6
4,Python,2
5,Frisbee,10
6,Fox,2
7,Tux,9
8,Tux,1
9,Lego,9
10,Tux,7
11,Barbie,2
12,Frisbee,2
13,Marbles,4
14,Giraffe,4
15,Monopoly,9
16,Lego,4
17,Tux,2
18,Tux,3
19,Giraffe,1
20,Tux,10

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Change in catalog file

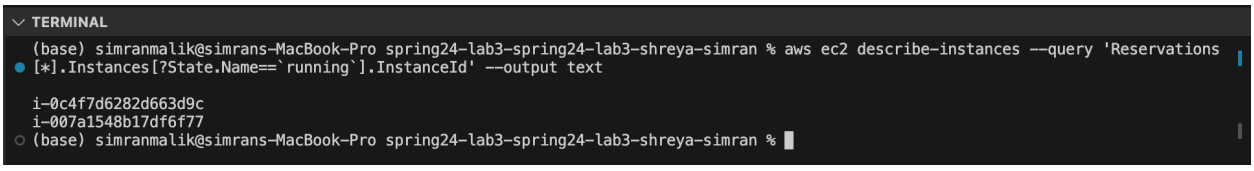


13) Clean UP

- Exiting ssh



- We ll check how many instances are available and get their instance ids.



- Now we will terminate instance

```
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran % aws ec2 terminate-instances --instance-ids i-0c4
f7d6282d663d9c i-007a1548b17df6f77

{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-0c4f7d6282d663d9c",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        },
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-007a1548b17df6f77",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
(base) simranmalik@simrans-MacBook-Pro spring24-lab3-spring24-lab3-shreya-simran %
```