- OutPut
  - Aws config

```
/d/Files/workspace/CS6///spring23-lab-3-m> aws configure
AWS Access Key ID [***************IX7L]:
AWS Secret Access Key [****************XMoY]:
Default region name [us-east-1]: us-east-1
Default output format [json]: json
/d/Files/workspace/CS677/spring23-lab-3-m> aws ec2 run-instances --image-id ami-
0d73480446600f555 --instance-type t2.micro --key-name vockey > instance.json
/d/Files/workspace/CS677/spring23-lab-3-m> aws ec2 describe-instances --instance
-id i-0588163cb41a73ccb
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0d73480446600f555",
                    "InstanceId": "i-0588163cb41a73ccb",
                    "InstanceType": "t2.micro",
                    "KeyName": "vockey",
                    "LaunchTime": "2023-05-03T19:31:47+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1c",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-25-74.ec2.internal",
                    "PrivateIpAddress": "172.31.25.74",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-54-90-183-189.compute-1.amazonaws.com"

                    "PublicIpAddress": "54.90.183.189",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-0e53df9e26766d32a",
                    "VpcId": "vpc-0ff189b7ac6062564",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "Ebs": {
                                "AttachTime": "2023-05-03T19:31:48+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-0735baf75ad1d9df5"
                            }
                        }
                    ],
                    "ClientToken": "9491cf6b-9b91-4660-b352-0f38827f7f7b",
                    "EbsOptimized": false,
```

  -
- Connect with aws

```
/d/Files/workspace/CS677/spring23-lab-3-m> ssh -i labsuser.pem ubuntu@ec2-54-90-183-189.compute-1.amazonaws.com
The authenticity of host 'ec2-54-90-183-189.compute-1.amazonaws.com (54.90.183.189)' can't be established.
ED25519 key fingerprint is SHA256:OMHkdLX1syRSfj5G3nIafL8Fw2l8D/zGXi3U5CBPs5I.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-90-183-189.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1068-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed May  3 19:34:04 UTC 2023

  System load:  0.16              Processes:           96
  Usage of /:   15.5% of 7.69GB   Users logged in:     0
  Memory usage: 20%               IP address for eth0: 172.31.25.74
  Swap usage:   0%
```

  -
- Start servers
  - Catalog.py

-

- Orders

  - 20001

ubuntu@ip-172-31-25-74:~$ cd spring23-lab-3-m/src/order
ubuntu@ip-172-31-25-74:~/spring23-lab-3-m/src/order$ ID=1 PORT=20001 python3 order.py
missed prders:

20001   1
 * Serving Flask app 'order' (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
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.31.25.74:20001/ (Press CTRL+C to quit)
 * Restarting with stat
missed prders:

20001   1
 * Debugger is active!
 * Debugger PIN: 601-508-836
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
{'order_no': 1, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 2, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 3, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 4, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 5, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:40] "POST /sync HTTP/1.1" 200 -
{'order_no': 6, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:40] "POST /sync HTTP/1.1" 200 -

  - 20002

```
ubuntu@ip-172-31-25-74:~$ cd spring23-lab-3-m/src/order
ubuntu@ip-172-31-25-74:~/spring23-lab-3-m/src/order$ ID=2 PORT=20002 python3 ord
er.py
missed prders:

20002   2
 * Serving Flask app 'order' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployme
nt.
 * Running on http://172.31.25.74:20002/ (Press CTRL+C to quit)
 * Restarting with stat
missed prders:

20002   2
 * Debugger is active!
 * Debugger PIN: 601-508-836
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
{'order_no': 1, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 2, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 3, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
127.0.0.1 - - [03/May/2023 20:15:39] "POST /sync HTTP/1.1" 200 -
{'order_no': 4, 'stock_name': 'Stock1', 'trade_type': 'Buy', 'quantity': '1'}
```

- 20003

```
ubuntu@ip-172-31-25-74:~$ cd spring23-lab-3-m/src/order
ubuntu@ip-172-31-25-74:~/spring23-lab-3-m/src/order$ ID=3 PORT=20003 python3 ord
er.py
missed prders:

20003   3
 * Serving Flask app 'order' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployme
nt.
 * Running on http://172.31.25.74:20003/ (Press CTRL+C to quit)
 * Restarting with stat
missed prders:

20003   3
 * Debugger is active!
 * Debugger PIN: 601-508-836
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /leader?leader=20003 HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:40] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:45] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:45] "GET /ping HTTP/1.1" 200 -
127.0.0.1 - - [03/May/2023 19:50:50] "GET /ping HTTP/1.1" 200 -
```

- front_end.py

```
ubuntu@ip-172-31-25-74:~/spring23-lab-3-m/src/front_end$ FLAG=FALSE python3 front_end.py
now leader is 20003
 * Serving Flask app 'front_end' (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
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.31.25.74:30001/ (Press CTRL+C to quit)
 * Restarting with stat
now leader is 20003
 * Debugger is active!
 * Debugger PIN: 601-508-836
now leader is 20003
```

- client.py

```
rd/Files/workspace/CS677/spring23-lab-3-m/src/client> FRONT=54.90.183.189 p=0.8 python client.py
mode:
1
query running time:  0.08007431030273438 s
{'stock_name': 'Stock2', 'price': 19.99, 'trade_volume': 0, 'quantity': 100}
random trade: Stock2, trade type: Sell, quantity: 2
trade running time:  0.07225799560546875 s
{'message': 'Trade failed!', 'order_no': -1}
mode:
1
query running time:  0.07033109664916992 s
{'stock_name': 'Stock1', 'price': 9.99, 'trade_volume': 100, 'quantity': 0}
mode:
1
query running time:  0.06055307388305664 s
{'stock_name': 'Stock1', 'price': 9.99, 'trade_volume': 100, 'quantity': 0}
mode:
1
```

- Function example
- Mode 1
  - p=0,1

```
ient.pys/workspace/CS677/spring23-lab-3-m/src/client> IP=localhost p=0 python cl
mode:
1
query running time:  4.024571180343628 s
{'stock_name': 'Stock2', 'price': 19.99, 'trade_volume': 5, 'quantity': 95}
mode:
```
    -

```
ient.pys/workspace/CS677/spring23-lab-3-m/src/client> IP=localhost p=1 python cl
mode:
1
query running time:  4.021483898162842 s
{'stock_name': 'Stock7', 'price': 69.99, 'trade_volume': 0, 'quantity': 100}
random trade: Stock7, trade type: Sell, quantity: 9
trade running time:  6.024532318115234 s
{'message': 'Trade failed!', 'order_no': -1}
mode:
```
    -
- Mode 2

```
mode:
2
input stock name(e.g:Stock1):
Stock1
input request times:
2
{'stock_name': 'Stock1', 'price': 9.99, 'trade_volume': 0, 'quantity': 100}
{'stock_name': 'Stock1', 'price': 9.99, 'trade_volume': 0, 'quantity': 100}
running time:  4.019810199737549 s
```
  -

```
mode:
2
input stock name(e.g:Stock1):
invalid
input request times:
2
{'error': {'code': 404, 'message': 'Stock not found'}}
{'error': {'code': 404, 'message': 'Stock not found'}}
running time:  8.033050775527954 s
mode:
```
  -
- Mode 3

-
```
mode:
3
input order_no:
1
input request times:
1
{'data': {'order_no': 1, 'quantity': '3', 'stock_name': 'Stock1', 'trade_type': 'Buy'}}
running time:  4.0178542137146 s
mode:
```

-
```
mode:
3
input order_no:
100
input request times:
1
{'error': {'code': 404, 'message': 'order not found'}}
running time:  4.019365310668945 s
mode:
```

- Mode 4

-
```
/d/Files/workspace/CS677/spring23-lab-3-m/src/client> IP=localhost python client
.py
mode:
4
input the stock name you would like to trade:
Stock1
input trade type:Buy/SellBuy
input quantity:
1
input request times:
1
{'order_no': 4}
running time:  12.062865018844604 s
mode:
```

-
```
mode:
4
input the stock name you would like to trade:
invalid
input trade type:Buy/SellBuy
input quantity:
1
input request times:
1
{'message': 'Stock not found!', 'order_no': -1}
running time:  6.024749994277954 s
mode:
```

-

- Test
    - Load test
        - Query stocks

-
    - Query order

-
    - Trade

-
- Function test

```
/d/Files/workspace/CS677/spring23-lab-3-m/src/test/func> bash test_func.sh
test_front_stocksQuery_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.020s

OK
test_front_stocksQuery_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.018s

OK
test_catalog_query_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.014s

OK
test_catalog_query_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.012s

OK
test_order_buy_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 10.060s

OK
test_order_sell_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 10.048s

OK
test_order_trade_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.021s

OK
test_order_trade_Buy_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.023s

OK
```

```
test_order_trade_Sell_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.021s

OK
test_front_buy_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 12.059s

OK
test_front_sell_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 12.054s

OK
test_front_trade_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 6.029s

OK
test_front_buy_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 6.026s

OK
test_front_sell_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 6.030s

OK
test_catalog_buy_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.025s

OK
test_catalog_sell_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.027s

OK
test_catalog_trade_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.013s

OK
```

```
test_catalog_buy_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.013s

OK
test_catalog_sell_outofstock (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.012s

OK
test_front_queryOrder_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.024s

OK
test_front_queryOrder_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 4.021s

OK
test_order_queryOrder_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.013s

OK
test_order_queryOrder_invalid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.012s

OK
test_ping_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.012s

OK
test_leader_valid (func_test.TestFunctions) ... ok

----------------------------------------------------------------------
Ran 1 test in 2.013s

OK
```