select \* from actor;

+----------+-------------+--------------+---------------------+

| actor\_id | first\_name | last\_name | last\_update |

+----------+-------------+--------------+---------------------+

| 1 | PENELOPE | GUINESS | 2006-02-15 04:34:33 |

| 2 | NICK | WAHLBERG | 2006-02-15 04:34:33 |

Commands

worker

1. Checking fastapi in AWS node from its bash station:  
   sudo systemctl status fastapi\_worker
2. To restart FastAPI  
   sudo systemctl restart fastapi\_worker
3. Status of sql

sudo systemctl status mysql

1. Connecting to sql:

mysql -u samnaz -p

USE sakila;

SELECT \* FROM actor;

1. INSERT INTO actor (first\_name, last\_name, last\_update) VALUES ('Samira', 'Nazari', '2024-11-21 10:56:11');
2. Executing query in sql directly in worker node

mysql -u samnaz -p -e "SELECT \* FROM actor;" sakila

1. Executing query in worker node by worker\_fastapi and also in bash my main application terminal

curl -X POST "http://54.147.16.226:8000/execute" -H "Content-Type: application/json" -d '{"query": "SELECT \* FROM actor where actor\_id = 100"}'

curl -X POST "http://54.227.55.157:8000/execute" -H "Content-Type: application/json" -d '{"query": "SELECT \* FROM actor where actor\_id = 100"}'

manager

1. To show the content of fastapi file

cat ~/fastapi\_manager/manager\_fastapi.py

proxy

8.for executing remotely in bash of application

curl <http://54.198.237.52:8000/>

9.viewing the log:  
tail -n 20 /home/ubuntu/proxy\_server.log

10. activation

source /home/ubuntu/fastapi\_env/bin/activate  
uvicorn proxy\_server\_fastapi\_route:app --host 0.0.0.0 --port 8000

12. checking the status of fastapi

sudo systemctl status fastapi\_proxy

13.check the detailed log:

journalctl -u fastapi\_proxy.service -n 50

14.restarting fast api:

sudo systemctl daemon-reload

sudo systemctl restart fastapi\_proxy

15. Press q inorder to exit from terminal fromsys

16. Run the following command to ensure the FastAPI app is accessible from the proxy server  
curl <http://localhost:8000>

17. Accessibility remotely

curl <http://54.242.20.195:8000/>

18. in local:

curl -X POST "http://localhost:8000/query/random" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 100;"}'

19. remotely:

random

curl -X POST "http://54.242.20.195:8000/query/random" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 100;"}'

direct

curl -X POST "http://98.84.98.234:8000/query/direct" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 200;"}'

customized

curl -X POST "http://98.84.98.234:8000/query/customized" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 10;"}'

write

curl -X POST "http://98.84.98.234:8000/query/direct" -H "Content-Type: application/json" -d '{"query": "INSERT INTO actor (first\_name, last\_name, last\_update) VALUES (\"Samira2\", \"Nazari2\", \"2024-11-21 10:58:11\");"}'

gatekeeper:

1. curl <http://localhost:8000>
2. Is it active or not  
   sudo systemctl status fastapi\_gatekeeper

curl -X POST "http://localhost:8000/validate\_request/direct" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 100;"}'

remotely

curl -X POST "http://3.84.240.207:8000/validate\_request/direct" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 150;"}'

Remotely:

curl http://3.84.240.207:8000

Trusted\_host

sudo systemctl daemon-reload

sudo systemctl restart fastapi\_trustedhost

sudo systemctl status fastapi\_trustedhost

local:  
curl -X POST "http://localhost:8000/process\_request/direct" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 200;"}'

remotely

curl -X POST "http://54.167.53.233:8000/process\_request/direct" \

-H "Content-Type: application/json" \

-d '{"query": "SELECT \* FROM sakila.actor WHERE actor\_id = 200;"}'

Benchmarking:

python benchmarking.py --gatekeeper 54.147.178.106 --query\_type write --strategy direct

python benchmarking.py --gatekeeper 54.147.178.106 --query\_type read --strategy direct

for print:

python benchmarking\_print.py --gatekeeper 54.147.178.106 --query\_type read --strategy direct

# SSH into the remote server

KEY\_PATH="TP3\_pem\_3.pem"

REMOTE\_USER="ubuntu"

ssh -i "$KEY\_PATH" "$REMOTE\_USER@${INSTANCE\_IP}"

ssh -i "TP3\_pem\_3.pem" [ubuntu@3.85.99.57](mailto:ubuntu@3.85.99.57) # Replace with your master server IP

for clustering in local (PC) in master

ssh -i "TP3\_pem\_3.pem" ubuntu@54.89.218.211 # Replace with your master server IP

mysql -u samnaz -p

mysql -u root -p

output:

mysql> SHOW MASTER STATUS;

+------------------+----------+--------------+------------------+-------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |

+------------------+----------+--------------+------------------+-------------------+

| mysql-bin.000001 | 3272570 | sakila | | |

+------------------+----------+--------------+------------------+-------------------+

1 row in set (0.00 sec)

Seeing log:

sudo cat /var/log/mysql/error.log

Checking slave

ssh -i "TP3\_pem\_3.pem" [ubuntu@54.144.58.200](mailto:ubuntu@54.144.58.200) # Replace with your slave server IP

sudo mysql -u root -p'123456'

SHOW SLAVE STATUS\G

mysql> SHOW SLAVE STATUS\G

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Slave\_IO\_State: Connecting to source

Master\_Host: 98.81.6.44

Master\_User: repl\_user

Master\_Port: 3306

Connect\_Retry: 60

Master\_Log\_File: mysql-bin.000001

Read\_Master\_Log\_Pos: 4

Relay\_Log\_File: ip-172-31-32-138-relay-bin.000001

Relay\_Log\_Pos: 4

Relay\_Master\_Log\_File: mysql-bin.000001

Slave\_IO\_Running: Connecting

Slave\_SQL\_Running: Yes

Replicate\_Do\_DB:

Replicate\_Ignore\_DB:

Replicate\_Do\_Table:

Replicate\_Ignore\_Table:

Replicate\_Wild\_Do\_Table:

Replicate\_Wild\_Ignore\_Table:

Last\_Errno: 0

Last\_Error:

Skip\_Counter: 0

Exec\_Master\_Log\_Pos: 4

Relay\_Log\_Space: 337

Until\_Condition: None

Until\_Log\_File:

Until\_Log\_Pos: 0

Master\_SSL\_Allowed: No

Master\_SSL\_CA\_File:

Master\_SSL\_CA\_Path:

Master\_SSL\_Cert:

Master\_SSL\_Cipher:

Master\_SSL\_Key:

Seconds\_Behind\_Master: NULL

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 2061

Last\_IO\_Error: Error connecting to source 'repl\_user@98.81.6.44:3306'. This was attempt 12/86400, with a delay of 60

seconds between attempts. Message: Authentication plugin 'caching\_sha2\_password' reported error: Authentication requires secure connection.

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 0

Master\_UUID:

Master\_Info\_File: mysql.slave\_master\_info

SQL\_Delay: 0

SQL\_Remaining\_Delay: NULL

Slave\_SQL\_Running\_State: Replica has read all relay log; waiting for more updates

Master\_Retry\_Count: 86400

Master\_Bind:

Last\_IO\_Error\_Timestamp: 241128 14:26:39

Last\_SQL\_Error\_Timestamp:

Master\_SSL\_Crl:

Master\_SSL\_Crlpath:

Retrieved\_Gtid\_Set:

Executed\_Gtid\_Set:

Auto\_Position: 0

Replicate\_Rewrite\_DB:

Channel\_Name:

Master\_TLS\_Version:

Master\_public\_key\_path:

Get\_master\_public\_key: 0

Network\_Namespace:

1 row in set, 1 warning (0.00 sec)

vim [file address]

vim is linux file editor

vim /etc/mysql/mysql.conf.d/mysqld.cnf

ls /etc

cat /etc/mysql/mysql.conf.d/mysqld.cnf