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