

Lesson 1: What is a server?

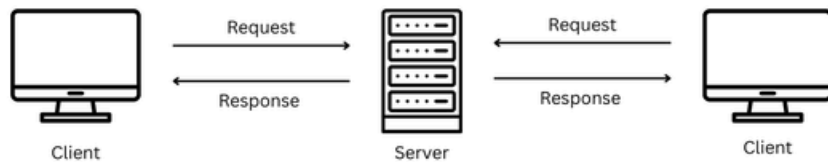
1. How does a server differ from your local computer?

A server is a powerful computer designed to provide services to other computers, called clients, over a network. It's meant to handle requests, manage resources, and stay on almost 24/7.

A local computer is built for individual use, a server is designed to support many users at once, have high reliability, and use specialized hardware to manage multiple tasks efficiently.

Local computers are used for personal tasks, while servers are usually dedicated to running applications, storing data, hosting websites, or managing emails for multiple users.

2. Draw and explain the client-server architecture.



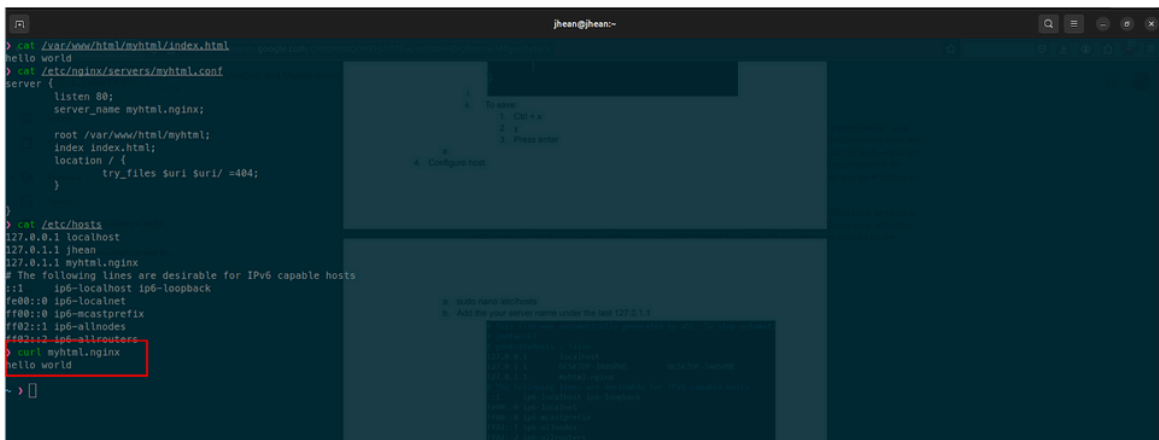
The client-server architecture is a network model that shows how clients connect to central servers to access resources or services. Clients request services or data, and the server responds by providing the data that the client requests. This setup allows multiple clients to access centralized resources or applications hosted on the server.

Lesson 4: Linux

What are the challenges you have encountered in following the given documentation?

For me, I had to do an extra step which is to include the directory of `/etc/nginx/servers/*.conf`; to the `nginx.conf` file.

Post a screenshot when you do `curl` of your own html.

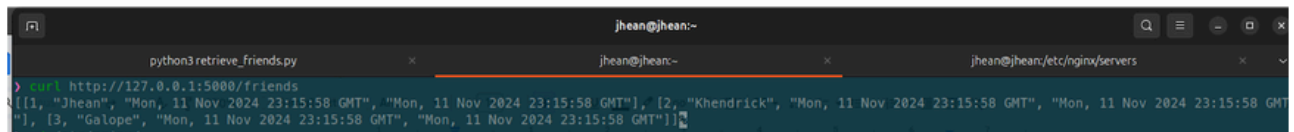


```
jhean@jhean:~$ cat /var/www/html/myhtml/index.html
hello world
jhean@jhean:~$ cat /etc/nginx/servers/myhtml.conf
server {
    listen 80;
    server_name myhtml.nginx;
    root /var/www/html/myhtml;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
jhean@jhean:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 jhean
127.0.1.1 myhtml.nginx
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe80::0 ip6-localnet
ff00::0 ip6-mcastprefix
::: ip6-allnodes
::: ip6-allrouters
jhean@jhean:~$ curl myhtml.nginx
hello world
```

Lesson 5: Requirement 1

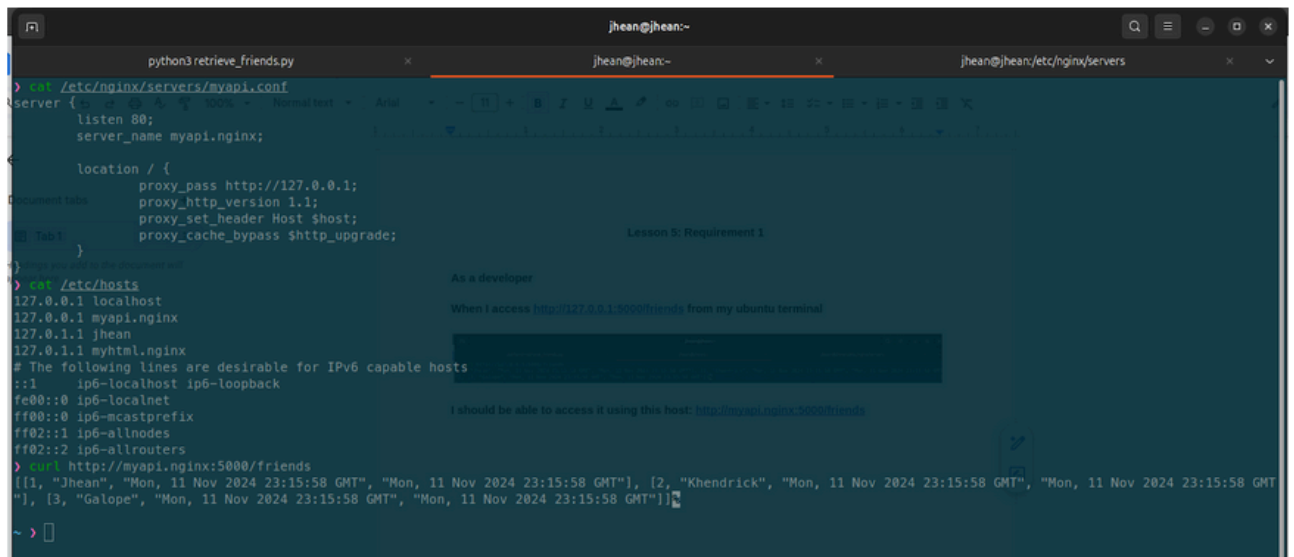
As a developer

When I access <http://127.0.0.1:5000/friends> from my ubuntu terminal



```
jhean@jhean:~  
python3 retrieve_friends.py  
[{"id": 1, "name": "Jhean", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "online"}, {"id": 2, "name": "Khendrick", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "offline"}, {"id": 3, "name": "Galope", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "online"}]
```

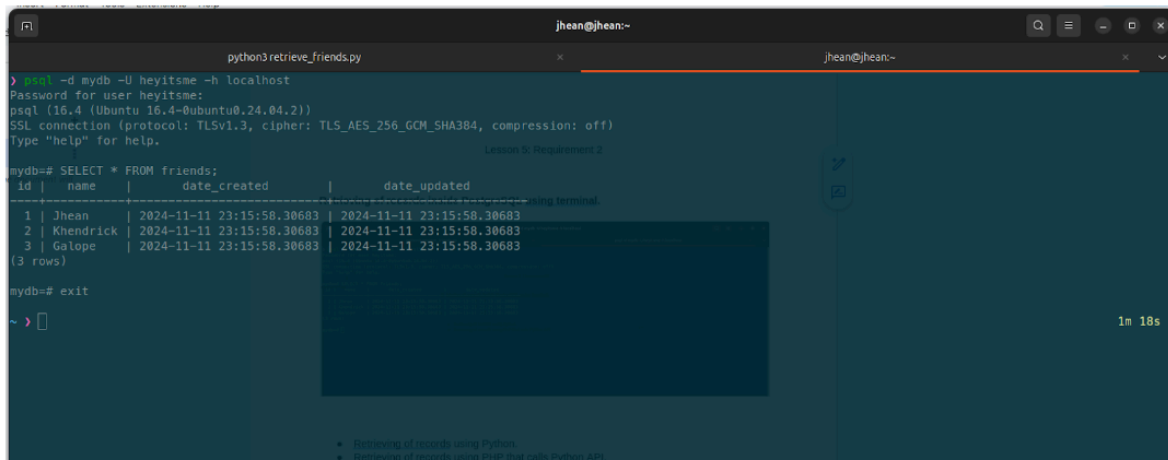
I should be able to access it using this host: <http://myapi.nginx:5000/friends>



```
jhean@jhean:~  
cat /etc/nginx/sites-enabled/myapi.conf  
server {  
    listen 80;  
    server_name myapi.nginx;  
  
    location / {  
        proxy_pass http://127.0.0.1:5000;  
        proxy_http_version 1.1;  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
}  
  
cat /etc/hosts  
127.0.0.1 localhost  
127.0.0.1 myapi.nginx  
127.0.1.1 jhean  
127.0.1.1 myhtml.nginx  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
curl http://myapi.nginx:5000/friends  
[[{"id": 1, "name": "Jhean", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "online"}, {"id": 2, "name": "Khendrick", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "offline"}, {"id": 3, "name": "Galope", "date": "Mon, 11 Nov 2024 23:15:58 GMT", "status": "online"}]]
```

Lesson 5: Requirement 2

Retrieving of records inside PostgreSQL using terminal.



```
python3 retrieve_friends.py
> psql -d mydb -U heyitsme -h localhost
Password for user heyitsme:
psql (16.4 (Ubuntu 16.4-0ubuntu0.24.04.2))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

mydb=# SELECT * FROM friends;
 id | name      | date_created      | date_updated
----+-----+-----+-----
  1 | Jhean    | 2024-11-11 23:15:58.30683 | 2024-11-11 23:15:58.30683
  2 | Khendrick | 2024-11-11 23:15:58.30683 | 2024-11-11 23:15:58.30683
  3 | Galope   | 2024-11-11 23:15:58.30683 | 2024-11-11 23:15:58.30683
(3 rows)

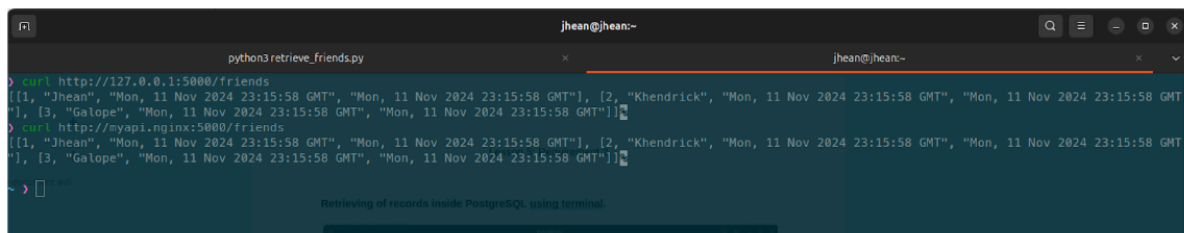
mydb=# exit
~ >
```

Retrieving of records using Python.

Retrieving of records using Flask that calls Python API.

Retrieving records using Python.

VIA FLASK API:



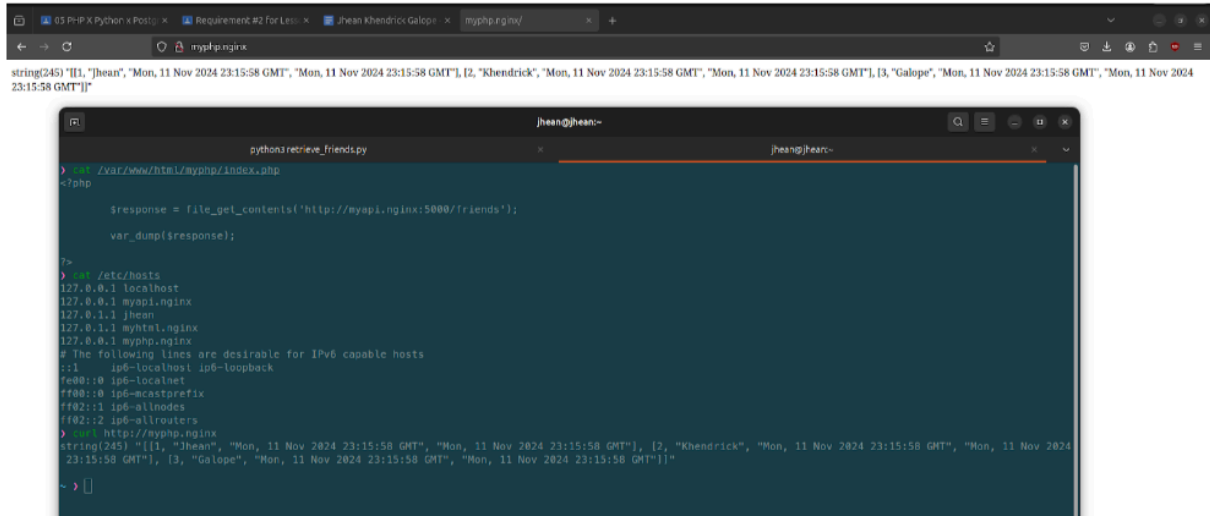
```
python3 retrieve_friends.py
> curl http://127.0.0.1:5000/friends
[[{"id": 1, "name": "Jhean", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}, {"id": 2, "name": "Khendrick", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}, {"id": 3, "name": "Galope", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}]]

> curl http://myapi.nginx:5000/friends
[[{"id": 1, "name": "Jhean", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}, {"id": 2, "name": "Khendrick", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}, {"id": 3, "name": "Galope", "date_created": "2024-11-11 23:15:58 GMT", "date_updated": "2024-11-11 23:15:58 GMT"}]]

~ >
```

Retrieving of records inside PostgreSQL using terminal.

Retrieving of records using PHP that calls Python API.



The screenshot shows a web browser window at the top and a terminal window below it. The browser window displays a JSON array of records. The terminal window shows a PHP script that calls a Python API to retrieve the same records.

Browser window (URL: myphp.nginx):

```
string(245) "[1, \"Jhean\", \"Mon, 11 Nov 2024 23:15:58 GMT\", \"Mon, 11 Nov 2024 23:15:58 GMT\"], [2, \"Khendrick\", \"Mon, 11 Nov 2024 23:15:58 GMT\", \"Mon, 11 Nov 2024 23:15:58 GMT\"], [3, \"Galope\", \"Mon, 11 Nov 2024 23:15:58 GMT\", \"Mon, 11 Nov 2024 23:15:58 GMT\"]]"
```

Terminal window (jhean@jhean:~):

```
python3 retrieve_friends.py
> curl -v http://myphp.nginx:5000/friends
* Hostname localhost, port 5000
* Connected to myphp.nginx (127.0.0.1) port 5000
> GET / HTTP/1.1
* HTTP/1.1 200 OK (application/json)
{
  "records": [
    {
      "id": 1,
      "name": "Jhean",
      "timestamp": "Mon, 11 Nov 2024 23:15:58 GMT",
      "status": "Mon, 11 Nov 2024 23:15:58 GMT"
    },
    {
      "id": 2,
      "name": "Khendrick",
      "timestamp": "Mon, 11 Nov 2024 23:15:58 GMT",
      "status": "Mon, 11 Nov 2024 23:15:58 GMT"
    },
    {
      "id": 3,
      "name": "Galope",
      "timestamp": "Mon, 11 Nov 2024 23:15:58 GMT",
      "status": "Mon, 11 Nov 2024 23:15:58 GMT"
    }
  ]
}
```