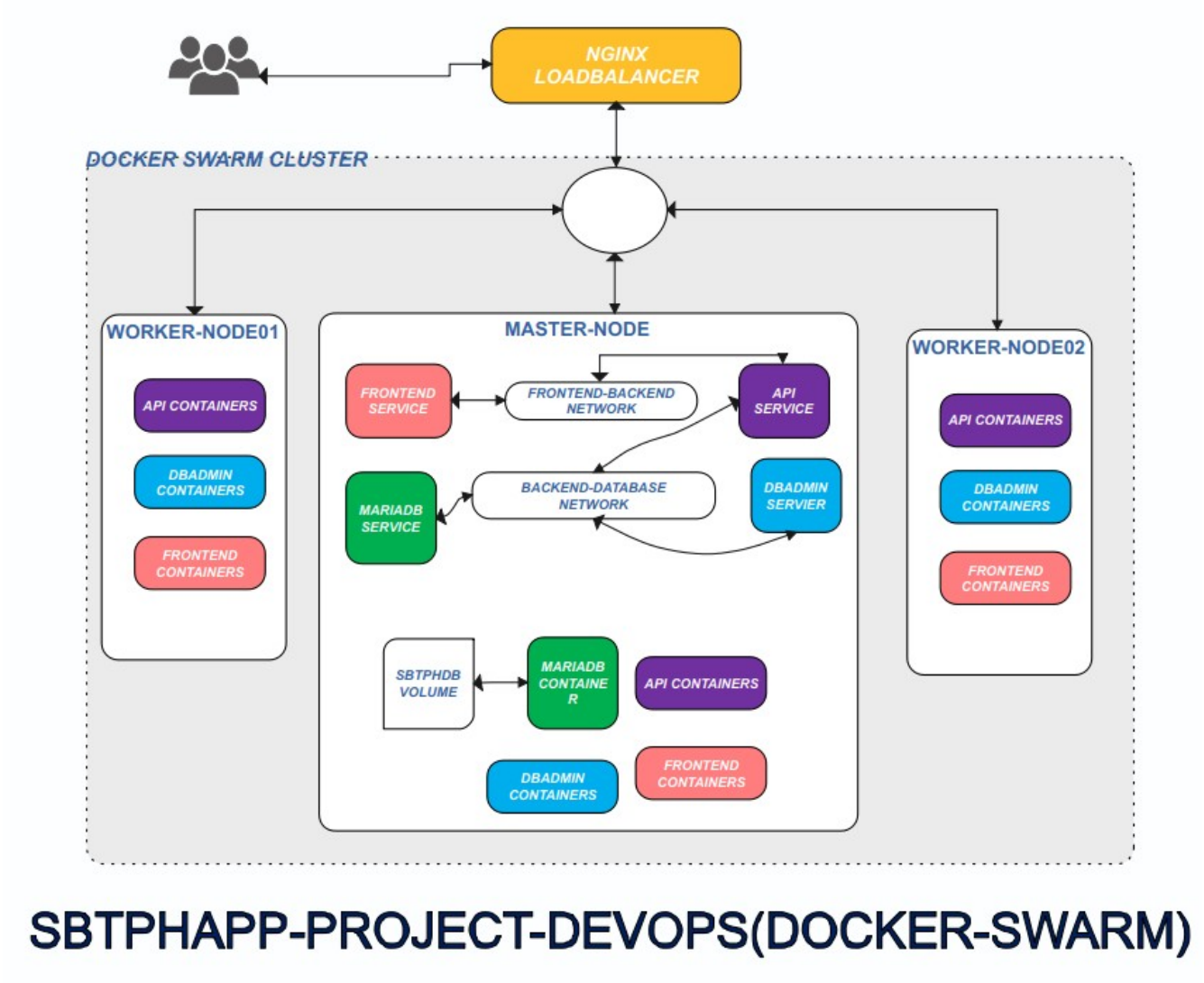
PROJECT ARCHITECTURE:



PROJECT DESCRIPTION:

In this project, we will build a Docker Swarm cluster consisting of 1 Master Node and 2 Worker Nodes. We'll deploy our 'sbtphapp' application on this cluster. Additionally, we will add NGINX to act as a load balancer. This project will be deployed on the Digital Ocean Cloud Provider. I chose Digital Ocean because it offers $200 of free credit for 2 months, which is ample for building and completing this project.

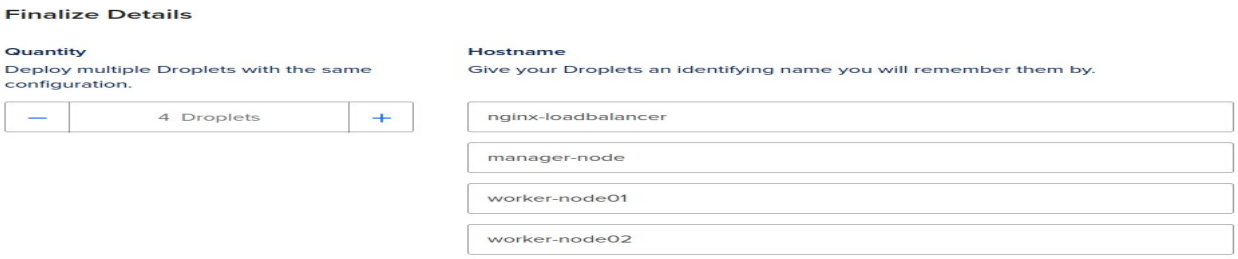
PROJECT REQUIREMENTS:

1. **Digital Ocean Accounts** – for Servers
2. **DockerHub Account** – for storing the docker image

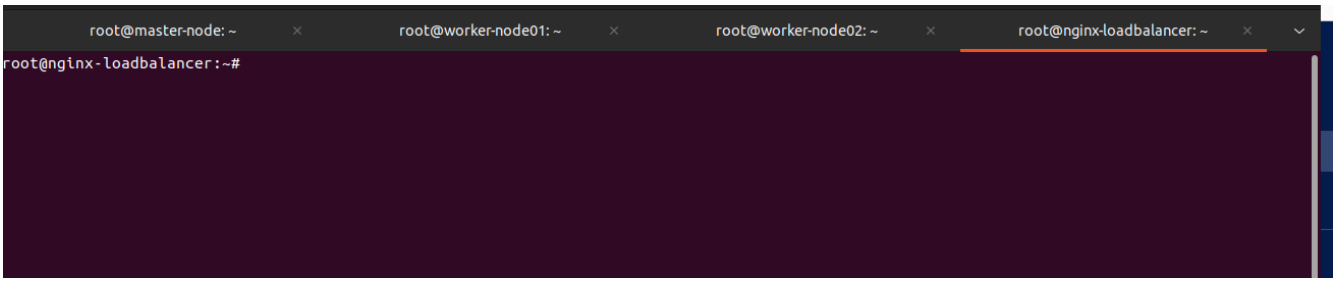
SETUP INSTRUCTIONS:

1. On Your Digital Ocean account, 4 Droplets with:
   - OPERATING SYSTEM => Ubuntu Server 20.04
   - SHARED CPU
   - Regular Disk type: SSD
   - 4GB/2CPU
   - Authentication Method => Password => **Type your choosen password**
   -

2. Name your 4 Droplets, and click **Create Droplet**



3. Once the Droplets are created, it's time to install the Docker engine on the manager node, worker node 01, and worker node 02, and Nginx on the Nginx load balancer droplet. SSH into these four droplets.



4. Installing docker engines on **manager-node, worker-node01 and worker-node02**:

Commands:

for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done

sudo apt-get update -y

sudo apt-get install ca-certificates curl gnupg -y
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
***https://download.docker.com/linux/ubuntu \ $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | ***
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update – ysudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y

4. On **master-node**, get eth0 (Public IP) and type this command: (Make sure replace it without Public IP)

docker swarm init --advertise-addr {MANAGER-ETHO-PUBLIC-IP}

```
root@master-node:~/sbtphapp-project-devops/docker# docker swarm init --advertise-addr 159.223.59.252
Swarm initialized: current node (q01blfv7t6as0n5a8ib48svjq) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1yonpk9mpyu0soo6sm2wvaj116a5z21p23mmkcz0cwq9p3j770-cv8i7a755y6ltrhpn5g7e9p4d 159.223.59.252:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

root@master-node:~/sbtphapp-project-devops/docker#
```

On **worker-node01** and **worker-node02**, paste this command: **docker swarm join –token *****

On **master-node** , check if two workers nodes are successfully joined:

docker node ls

```
root@master-node:~/sbtphapp-project-devops/docker# docker node list
ID                            HOSTNAME        STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
q01blfv7t6as0n5a8ib48svjq *   master-node     Ready     Active         Leader           24.0.7
1ed2uo1krtvq7xjjanhgntshj     worker-node01   Ready     Active                          24.0.7
mln309zp7la8hi95b62sgmmu8     worker-node02   Ready     Active                          24.0.7
root@master-node:~/sbtphapp-project-devops/docker#
```

5. On **master-node** clone this github repo:

git clone -b docker https://github.com/robudexIT/sbtphapp-project-devops.git

6. cd to **sbtphapp-project-devops/docker**, get your nginx-loadbalancer server public ip and run this command: (***make sure you replace it your nginx-loadbalancer server ip before running the command***)

find frontend/dist/js/* -type f -exec sed -E -i 's/\b([0-9]{1,3}\.){3}[0-9]*{1,3}:8081
b/YOUR_NGINX_LOADBALANCER_PUBLIC_IP_HERE:8081/g' {} +

7. Run this commnd:

docker compose build

8. Login to your **docker hub account**, check the docker images created, tag the images and push it to your docker hub.

docker login
docker images
docker tag docker-api:latest {YOURDOCKERHUBUSERNAME}/sbtphapi:latest
docker tag docker-frontend:latest {YOURDOCKERHUBUSERNAME}/sbtphapp:latest
docker push {YOURDOCKERHUBUSERNAME}/sbtphapi:latest
docker push {YOURDOCKERHUBUSERNAME}/sbtphapp:latest

9. On the **master-node**, create docker overlay network

docker network create -d overlay frontend-backend
docker network create -d overlay backend-database

10. On the **master-node** create **mariadb servic**e, check if mariadb service was created, check if which Node **mariabdb container** is running

***docker service create --detach=true --name mariadb --network backend-database ***
***--env-file ./env/db.env ***
***--mount type=volume,source=sbtphdb,target=/var/lib/mysql ***
***--mount type=bind,source=$(pwd)/database,target=/docker-entrypoint-initdb.d ***
--publish 3307:3306 mariadb

docker service ls

docker service ps mariadb

```
root@master-node:~/sbtphapp-project-devops/docker#       docker service create --detach=true   --name mariadb  --network backend-database  \
>       --env-file ./env/db.env  \
>       --mount type=volume,source=sbtphdb,target=/var/lib/mysql  \
>       --mount type=bind,source=$(pwd)/database,target=/docker-entrypoint-initdb.d \
>       --publish 3307:3306   mariadb
lrowl1agjl11l2ymr7k7hr1xs
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker# docker service ls
ID             NAME      MODE         REPLICAS   IMAGE            PORTS
lrowl1agjl11   mariadb   replicated   1/1        mariadb:latest   *:3307->3306/tcp
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker# docker service ps mariadb
ID             NAME            IMAGE            NODE            DESIRED STATE   CURRENT STATE             ERROR                              P
ORTS
l6mseoifnzx9   mariadb.1       mariadb:latest   master-node     Running         Running 9 seconds ago
ysj15doorf4p    \_ mariadb.1   mariadb:latest   worker-node02   Shutdown        Rejected 15 seconds ago   "invalid mount config for type…"
root@master-node:~/sbtphapp-project-devops/docker#
```

11. On master-node create **api service**, check if api service was created, check if which Node **api container** is running. (Make sure your replace it with your docker hub username account

***docker service create --name api --env-file ./env/api.env ***
***--network backend-database -p 8081:80 --replicas=5 ***
{YOUR-DOCKER-HUB-USER}/sbtphapi

docker service ls

docker service ps api

```
root@master-node:~/sbtphapp-project-devops/docker# docker service create --name api --env-file ./env/api.env  \
>           --network backend-database -p 8081:80 --replicas=5 \
>          robudex17/sbtphapi:latest
srwi8eetx0fb6ghzu3nanqzff
overall progress: 5 out of 5 tasks
1/5: running   [==================================================>]
2/5: running   [==================================================>]
3/5: running   [==================================================>]
4/5: running   [==================================================>]
5/5: running   [==================================================>]
verify: Service converged
root@master-node:~/sbtphapp-project-devops/docker# docker service ls
ID             NAME      MODE         REPLICAS   IMAGE                     PORTS
srwi8eetx0fb   api       replicated   5/5        robudex17/sbtphapi:latest   *:8081->80/tcp
lrowl1agjl11   mariadb   replicated   1/1        mariadb:latest              *:3307->3306/tcp
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker# docker service ps api
ID             NAME      IMAGE                     NODE           DESIRED STATE   CURRENT STATE            ERROR     PORTS
8z4445a30sje   api.1     robudex17/sbtphapi:latest   worker-node02   Running         Running 27 seconds ago
9gttk63pl8dd   api.2     robudex17/sbtphapi:latest   master-node     Running         Running 28 seconds ago
lf9ekj0z7ong   api.3     robudex17/sbtphapi:latest   worker-node01   Running         Running 27 seconds ago
v3gey5seduo8   api.4     robudex17/sbtphapi:latest   worker-node02   Running         Running 27 seconds ago
6e0qffvehxyr   api.5     robudex17/sbtphapi:latest   worker-node01   Running         Running 27 seconds ago
root@master-node:~/sbtphapp-project-devops/docker#
```

12. On **master-node** create **frontend** service, check if **frontend service** was created, check if which Node **frontend container** is running. (Make sure your replace it with your docker hub username account

***docker service create --name frontend ***
***--network frontend-backend -p 8082:80 --replicas=5 ***
{YOUR-DOCKER-HUB-USER}/sbtphapp:latest

docker service ls

docker service ps frontend

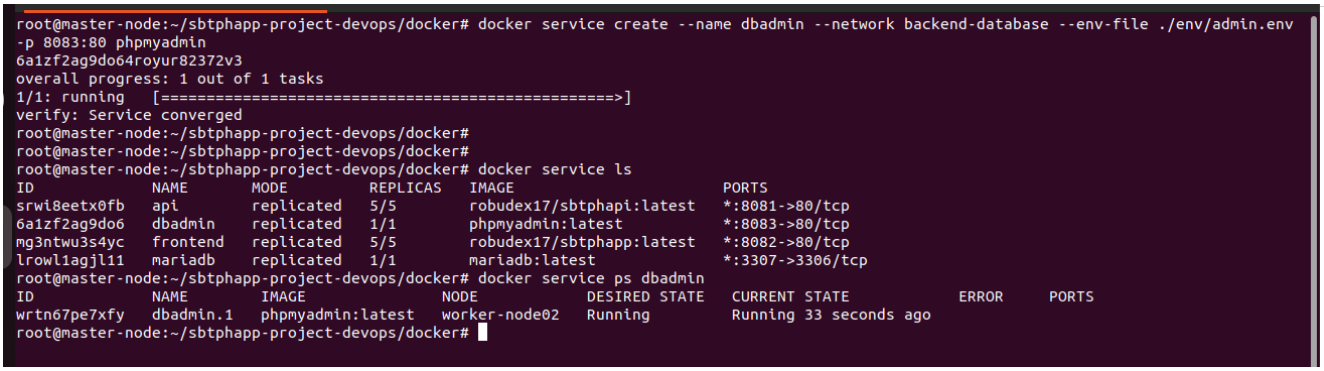
```
root@master-node:~/sbtphapp-project-devops/docker# docker service create --name frontend  \
> --network frontend-backend -p 8082:80 --replicas=5 \
> robudex17/sbtphapp:latest
mg3ntwu3s4yco7bb9o78vlzf4
overall progress: 5 out of 5 tasks
1/5: running   [==================================================>]
2/5: running   [==================================================>]
3/5: running   [==================================================>]
4/5: running   [==================================================>]
5/5: running   [==================================================>]
verify: Service converged
root@master-node:~/sbtphapp-project-devops/docker# docker service ls
ID             NAME       MODE         REPLICAS   IMAGE                     PORTS
srwi8eetx0fb   api        replicated   5/5        robudex17/sbtphapi:latest   *:8081->80/tcp
mg3ntwu3s4yc   frontend   replicated   5/5        robudex17/sbtphapp:latest   *:8082->80/tcp
lrowl1agjl11   mariadb    replicated   1/1        mariadb:latest              *:3307->3306/tcp
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker#
root@master-node:~/sbtphapp-project-devops/docker# docker service ps frontend
ID             NAME         IMAGE                     NODE           DESIRED STATE   CURRENT STATE            ERROR     PORTS
iftfkez9phaw   frontend.1   robudex17/sbtphapp:latest   master-node     Running         Running 21 seconds ago
gbk3yy1loonn   frontend.2   robudex17/sbtphapp:latest   worker-node01   Running         Running 21 seconds ago
djzurwo0mmth   frontend.3   robudex17/sbtphapp:latest   worker-node02   Running         Running 22 seconds ago
nspchvsvtpgn   frontend.4   robudex17/sbtphapp:latest   master-node     Running         Running 21 seconds ago
xjhshbwiztq2   frontend.5   robudex17/sbtphapp:latest   worker-node01   Running         Running 21 seconds ago
root@master-node:~/sbtphapp-project-devops/docker#
```

12. On **master-node** create **dbadmin** service, check if dbadmin service was created, check if which Node **dbadmin container** is running.

docker service create --name dbadmin --network backend-database --env-file ./env/admin.env -p 8083:80 phpmyadmin

docker service ls

docker service ps dbadmin



13. On **nginx-loadbancer**, inginx

sudo apt update

sudo apt-get install nginx

14. Create **/etc/nginx/conf.d/load-balancer.conf** file and paste this config below: **(Make sure you update the ip address on the upstream directives:**

```
upstream frontend {
   server {MASTER-NODE-IP-HERE}:8082;
   server {WORKER-NODE01-IP-HERE}:8082;
   server  {WORKER-NODE02-IP-HERE}:8082;
   # Add more Swarm nodes as needed
}

upstream backend {
   server {MASTER-NODE-IP-HERE}:8081;
   server {WORKER-NODE01-IP-HERE}:8081;
   server  {WORKER-NODE02-IP-HERE}:8081;

   # Add more servers as needed
}

upstream phpmyadmin {
   server {MASTER-NODE-IP-HERE}:8083;
   server {WORKER-NODE01-IP-HERE}:8083;
   server  {WORKER-NODE02-IP-HERE}:8083;

   # Add more servers as needed
}
```

```
server {
    listen 8082;
    server_name yourdomain.com;

    location / {
        proxy_pass http://frontend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

server {
    listen 8081;
    server_name yourdomain.com;

    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}

server {
    listen 8083;
    server_name yourdomain.com;

    location / {
        proxy_pass http://phpmyadmin;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

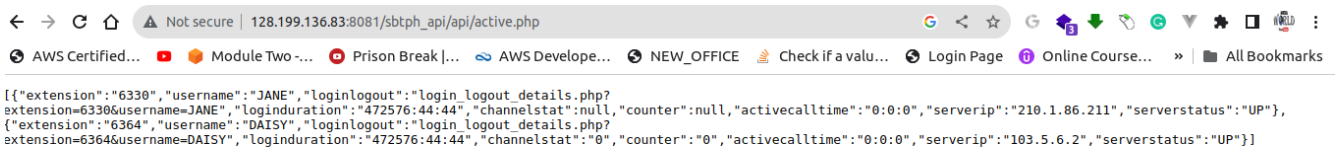
14. Restart nginx service

nginx -t
systemctl reload nginx
systemctl restart nginx

TESTING THE APP: Get the nginx-loadbalancer ip address

For Api Testing:

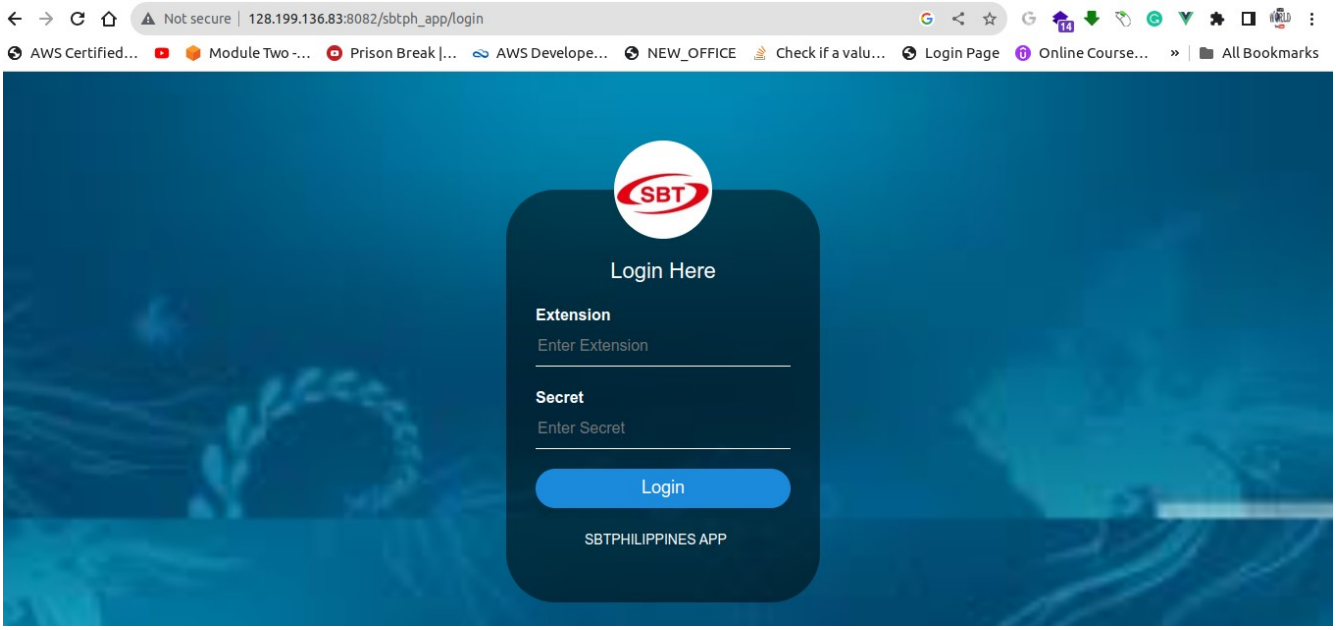
http://{NGINX-LOADBALANCER-IP}:8081/sbtph_api/api/active.php

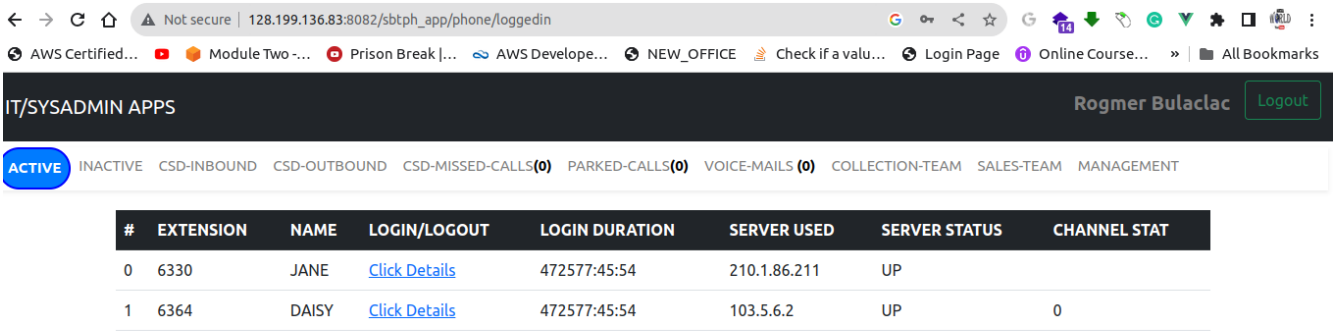


For Frontend Testing:

http://{NGINX-LOADBALANCER-IP}:8082/sbtph_app/login

Extension: 6336
Secret: 99999





#	EXTENSION	NAME	LOGIN/LOGOUT	LOGIN DURATION	SERVER USED	SERVER STATUS	CHANNEL STAT
0	6330	JANE	Click Details	472577:45:54	210.1.86.211	UP	
1	6364	DAISY	Click Details	472577:45:54	103.5.6.2	UP	0

For dbadmin Testing:

http://{NGINX-LOADBALANCER-IP}:8083