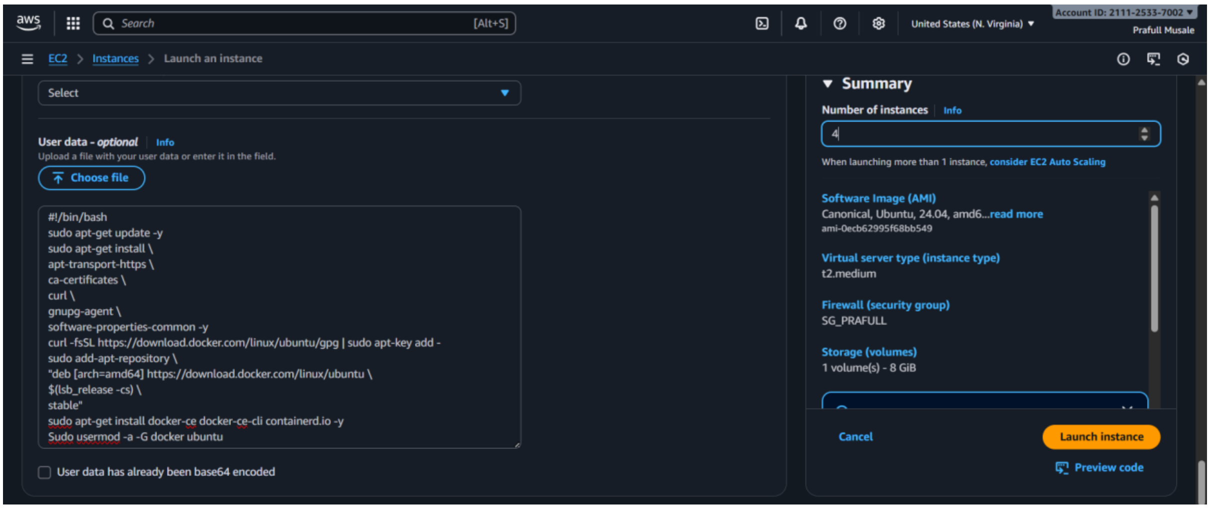
D Y PATIL
INTERNATIONAL
UNIVERSITY
AKURDI PUNE

School of Computer Science, Engineering and Applications(SCSEA)
B.Tech FIY (CCSA)
Subject : Cloud Automation & Devops (P)

Name of the Student: **Pratik.M.Rebari** **PRN: 20220802183**

Title of Practicle : **5.Web application hosting with docker stack and docker compose (Voting_app)**

1. Launch 4 instance with docker installation



EC2 > Instances > Launch an instance

User data - *optional*
Upload a file with your user data or enter it in the field.

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
sudo apt-get install docker-ce docker-ce-cli containerd.io -y
Sudo usermod -a -G docker ubuntu
```

□ User data has already been base64 encoded

Summary

Number of instances: 4

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-0ecb62995f68bb549

Virtual server type (instance type): t2.medium

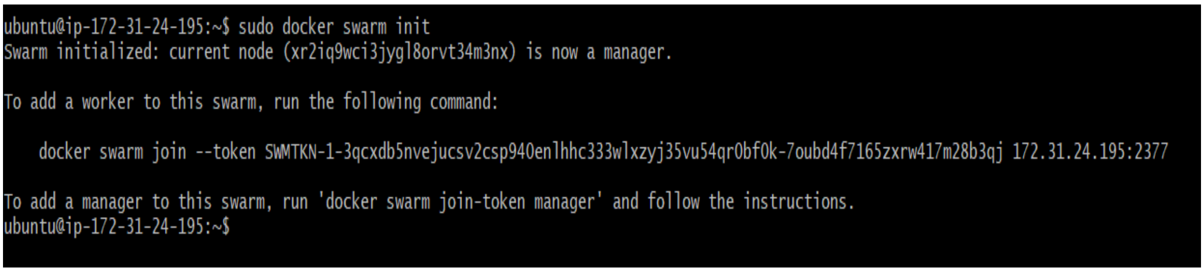
Firewall (security group): SG_PRAFULL

Storage (volumes): 1 volume(s) - 8 GiB

2. Connect all instance to terminal

ubuntu@ip-172-31-24-195:...
ubuntu@ip-172-31-18-27: ~
ubuntu@ip-172-31-25-4: ~
ubuntu@ip-172-31-20-92: ~

3. Now, Fire $ sudo docker swarm init at 1st instance for give him leadership

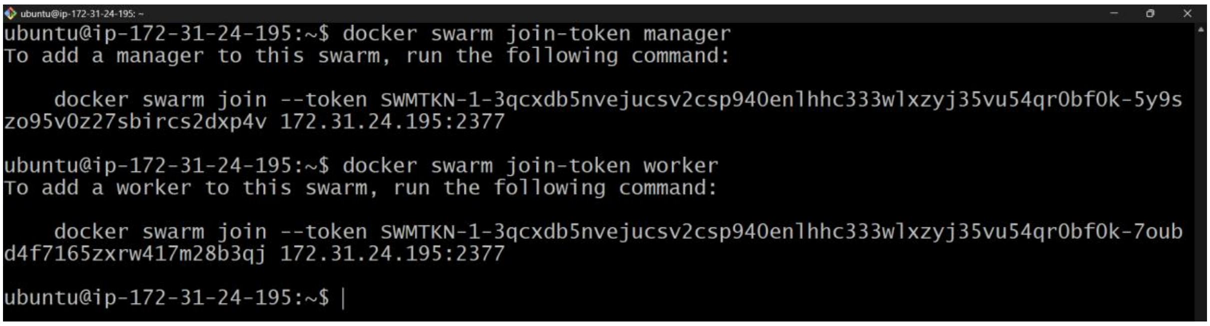


```
ubuntu@ip-172-31-24-195:~$ sudo docker swarm init
Swarm initialized: current node (xr2iq9wci3jygl8orvt34m3nx) is now a manager.

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

    docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enlhhc333wlxzyj35vu54qr0bf0k-7oubd4f7165zxrw417m28b3qj 172.31.24.195:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
ubuntu@ip-172-31-24-195:~$
```

4. Now generate token for manager and worker with following commands

$ docker swarm join-token manager

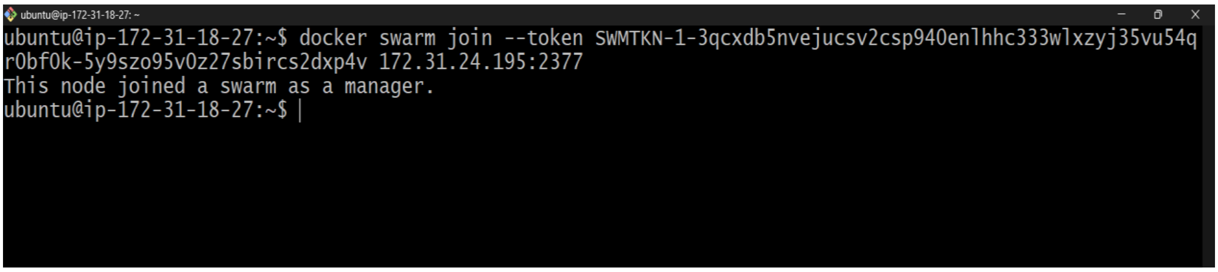
$ docker swarm join-token worker



```
ubuntu@ip-172-31-24-195:~$ docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enlhhc333wlxzyj35vu54qr0bf0k-5y9s
zo95v0z27sbircs2dxp4v 172.31.24.195:2377

ubuntu@ip-172-31-24-195:~$ docker swarm join-token worker
To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enlhhc333wlxzyj35vu54qr0bf0k-7oub
d4f7165zxrw417m28b3qj 172.31.24.195:2377

ubuntu@ip-172-31-24-195:~$ |
```

5. Now join 2nd instance as a manger with the help of token

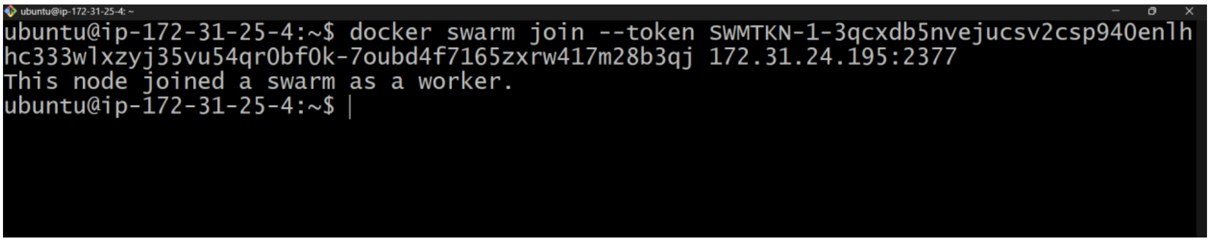
Copy manager token and paste into 2nd instance



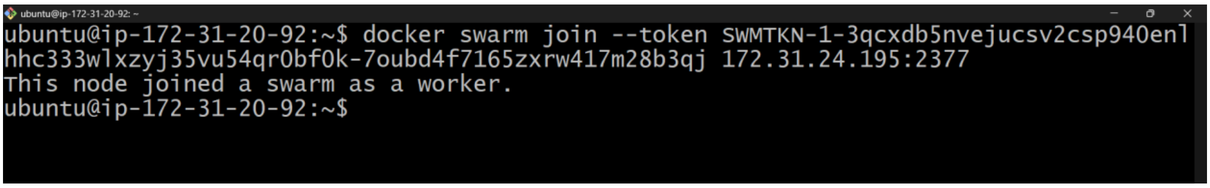
```
ubuntu@ip-172-31-18-27:~$ docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enlhhc333wlxzyj35vu54q
r0bf0k-5y9szo95v0z27sbircs2dxp4v 172.31.24.195:2377
This node joined a swarm as a manager.
ubuntu@ip-172-31-18-27:~$ |
```

6. Join 3rd and 4th instance as a worker with the help of token

Copy worker token and paste into 3rd and 4th instance

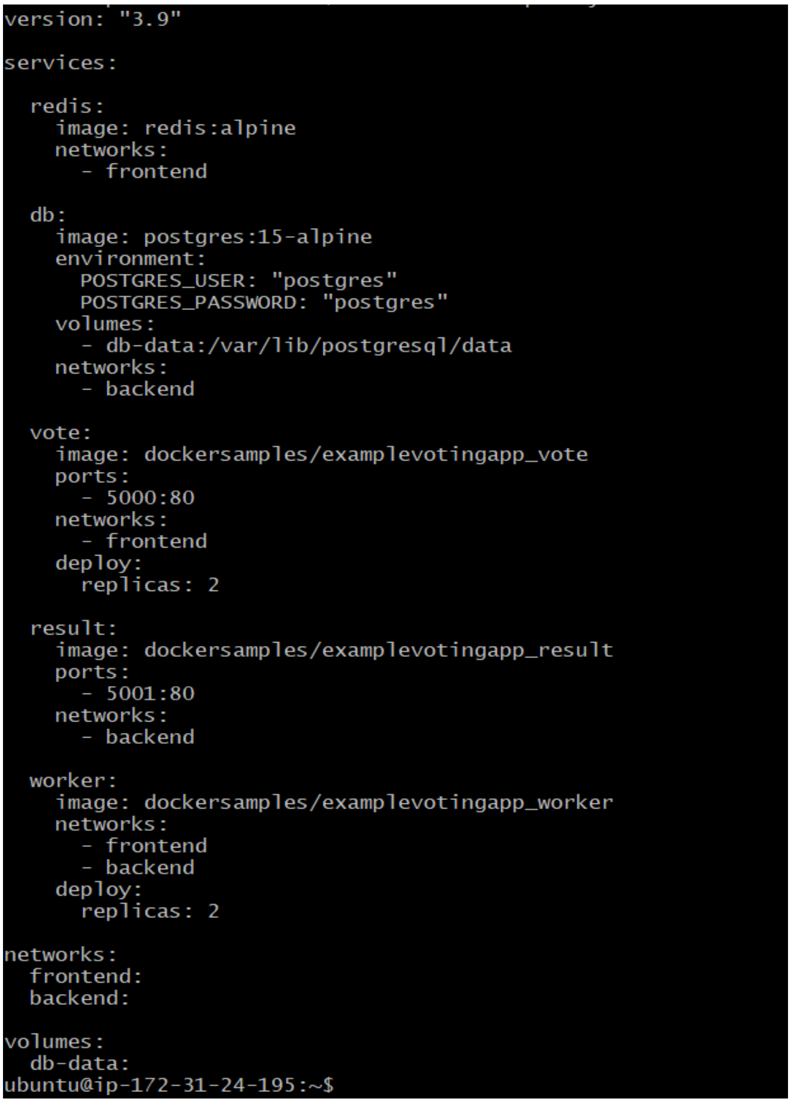


```
ubuntu@ip-172-31-25-4:~$ docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enlh
hc333wlxzyj35vu54qr0bf0k-7oubd4f7165zxrw417m28b3qj 172.31.24.195:2377
This node joined a swarm as a worker.
ubuntu@ip-172-31-25-4:~$ |
```



```
ubuntu@ip-172-31-20-92:~$ docker swarm join --token SWMTKN-1-3qcxdb5nvejucsv2csp940enl
hhc333wlxzyj35vu54qr0bf0k-7oubd4f7165zxrw417m28b3qj 172.31.24.195:2377
This node joined a swarm as a worker.
ubuntu@ip-172-31-20-92:~$
```

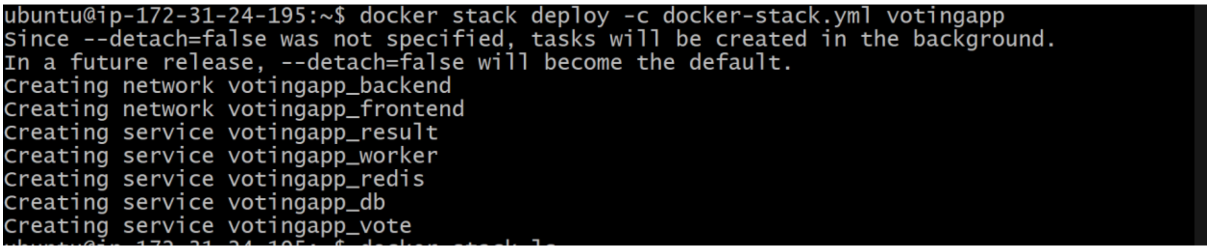
7. Now create docker-stack.yml to write the services



```
version: "3.9"

services:

  redis:
    image: redis:alpine
    networks:
      - frontend

  db:
    image: postgres:15-alpine
    environment:
      POSTGRES_USER: "postgres"
      POSTGRES_PASSWORD: "postgres"
    volumes:
      - db-data:/var/lib/postgresql/data
    networks:
      - backend

  vote:
    image: dockersamples/examplevotingapp_vote
    ports:
      - 5000:80
    networks:
      - frontend
    deploy:
      replicas: 2

  result:
    image: dockersamples/examplevotingapp_result
    ports:
      - 5001:80
    networks:
      - backend

  worker:
    image: dockersamples/examplevotingapp_worker
    networks:
      - frontend
      - backend
    deploy:
      replicas: 2

networks:
  frontend:
  backend:

volumes:
  db-data:
ubuntu@ip-172-31-24-195:~$
```

8. Now deploy with stack with the following command

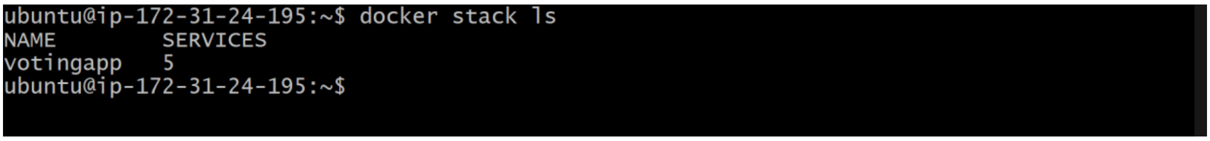
```
$ docker stack deploy -c docker-stack.yml votingapp
```



```
ubuntu@ip-172-31-24-195:~$ docker stack deploy -c docker-stack.yml votingapp
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network votingapp_backend
Creating network votingapp_frontend
Creating service votingapp_result
Creating service votingapp_worker
Creating service votingapp_redis
Creating service votingapp_db
Creating service votingapp_vote
```

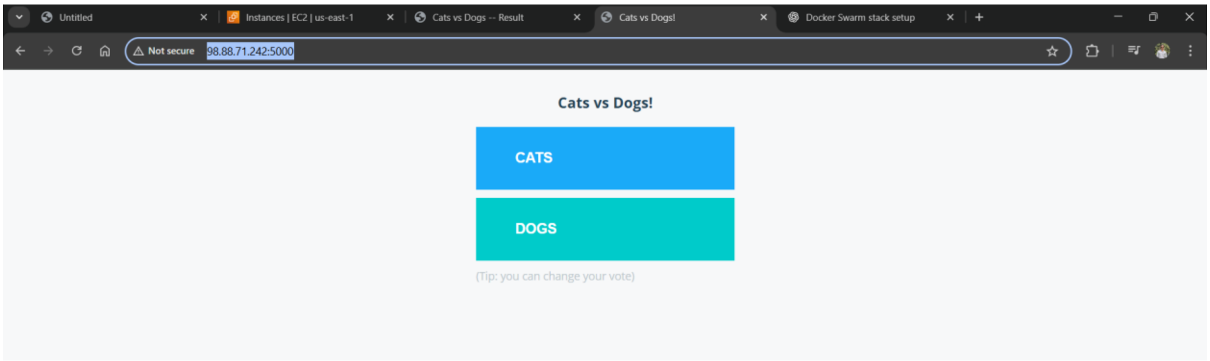
9. Now check stack is created or not with following command

$docker stack ls



```
ubuntu@ip-172-31-24-195:~$ docker stack ls
NAME        SERVICES
votingapp   5
ubuntu@ip-172-31-24-195:~$
```

10. Now for voting enter on browser <Public-IP>:5000



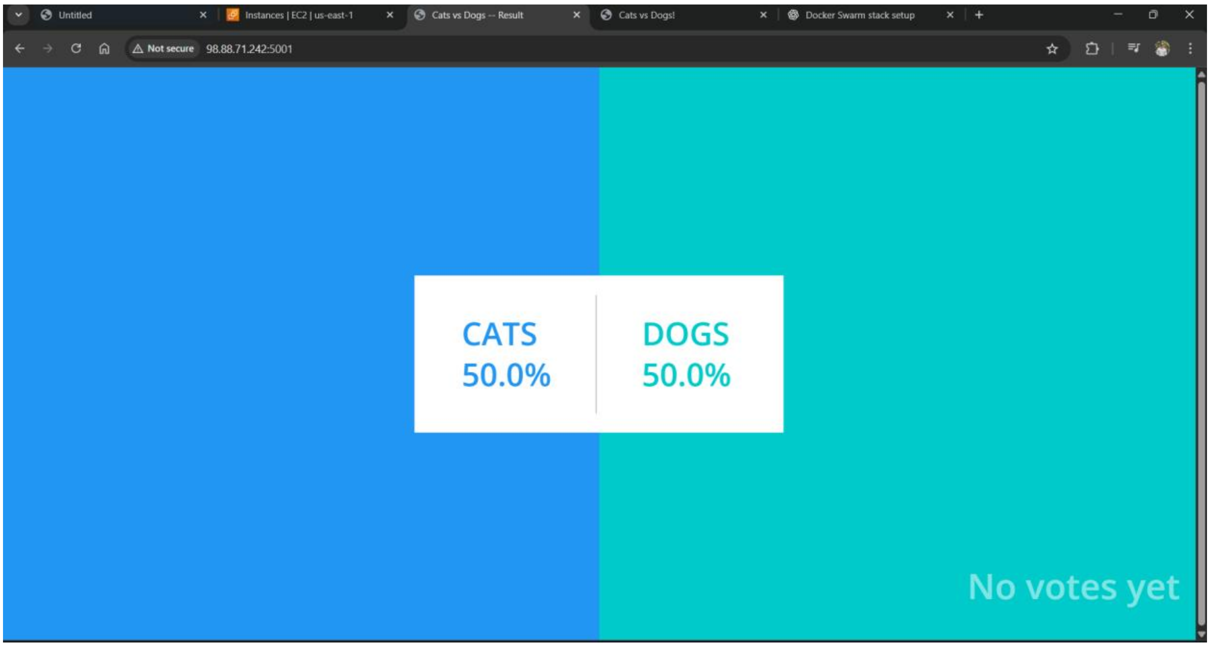
Cats vs Dogs!

CATS

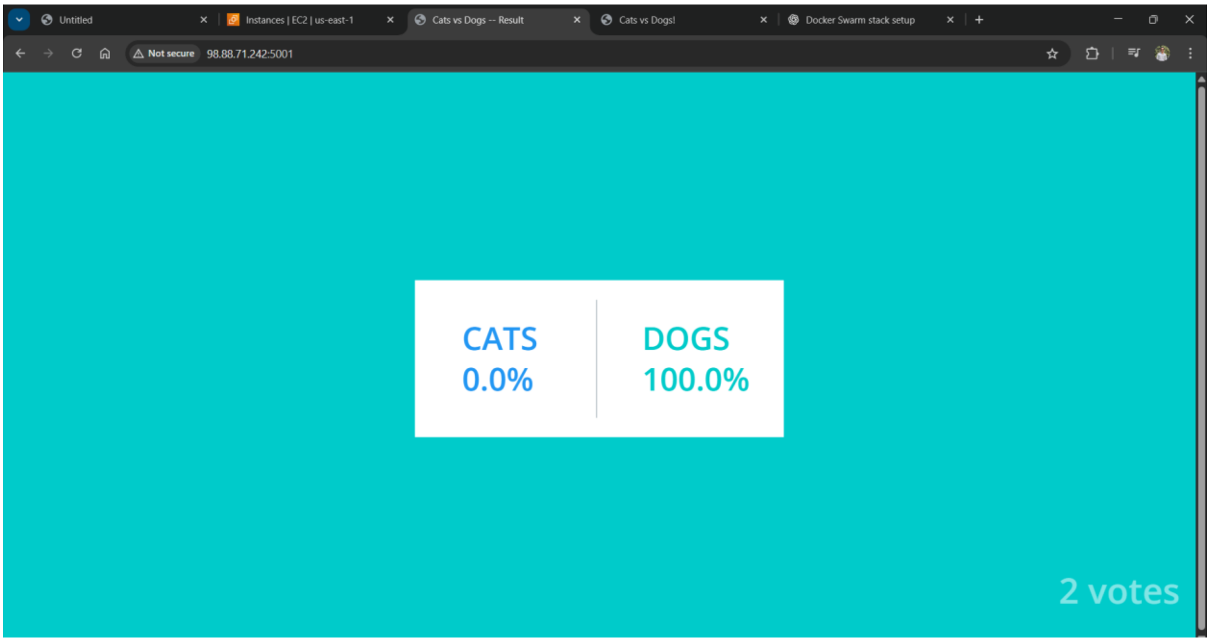
DOGS

(Tip: you can change your vote)

11. For checking Result enter on browser <Public-IP>:5001



12. Our Voting app is properly working


Untitled
Instances | EC2 | us-east-1
Cats vs Dogs -- Result
Cats vs Dogs!
Docker Swarm stack setup
Not secure
98.88.71.242:5001
CATS
0.0%
DOGS
100.0%
2 votes