# Team 44 MRC Infra. AsBuild Document

* We need 1 VM to host everything we need
  + Scale up mastodon docker instances when needed
* Or 4 VMs max
  + 2VMs for CouchDB
    - Prod-db-ubuntu
  + 1 VM for Web App
    - Prod-webapp-ubuntu
  + 1 VM for Mastodon Harvesters
    - Prod-harvesters-ubuntu
* Create key-pair

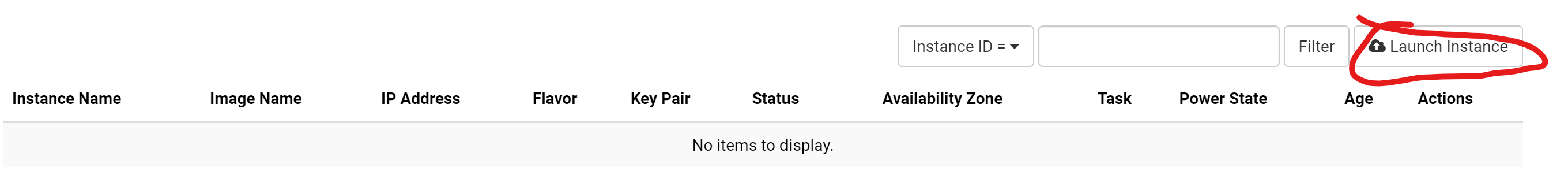
Graphical user interface, application

Description automatically generated

Public key

AAAAB3NzaC1yc2EAAAADAQABAAABAQDBgwceuhtLvInsuqqWcPkmegEYWDpE1GWHJkrXi1gbe85ELKfa4G9LzNsbLrdq4DLhjziSuhCFaq2o7MM8piAH/6xdwZB5hOKcF6YJPLSmLLSe+mS5T1ceQ+jG9vP8RhhsBOD+fmMQnvv6iza1Jgs4Cspz1rIFdeujbir60rEjQgxBfk1jn2MaR9d7xVfnj74tUvdnY4cHBlywdchbtGX31+Lo9HGl4HkNI4quf7IDVlHKGp2lgT91CpRleY3oXaqfydnY/zAjmF9dOduoC9PMQlsKgjEWkGxyecBmMYPs5o7QGyDzRgS/NfszCMPcqUwXBMmYutTpR78CfvPe7G3N Generated-by-Nova

* Create Ubuntu VM. Launch instance



Select NeCTAR Ubuntu 22.04 LTS (Jammy) amd64 (with Docker) image

Flavor

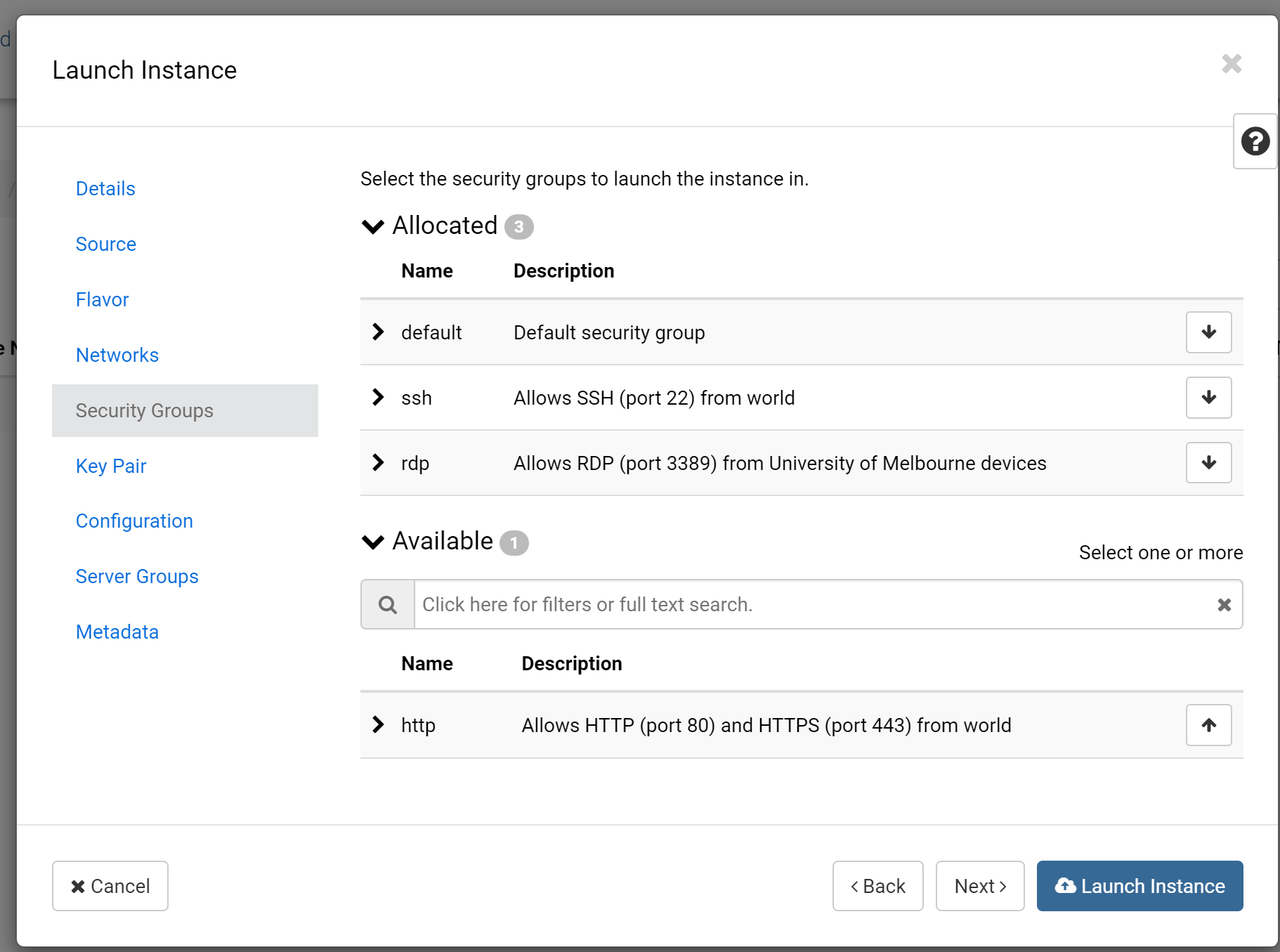
Table

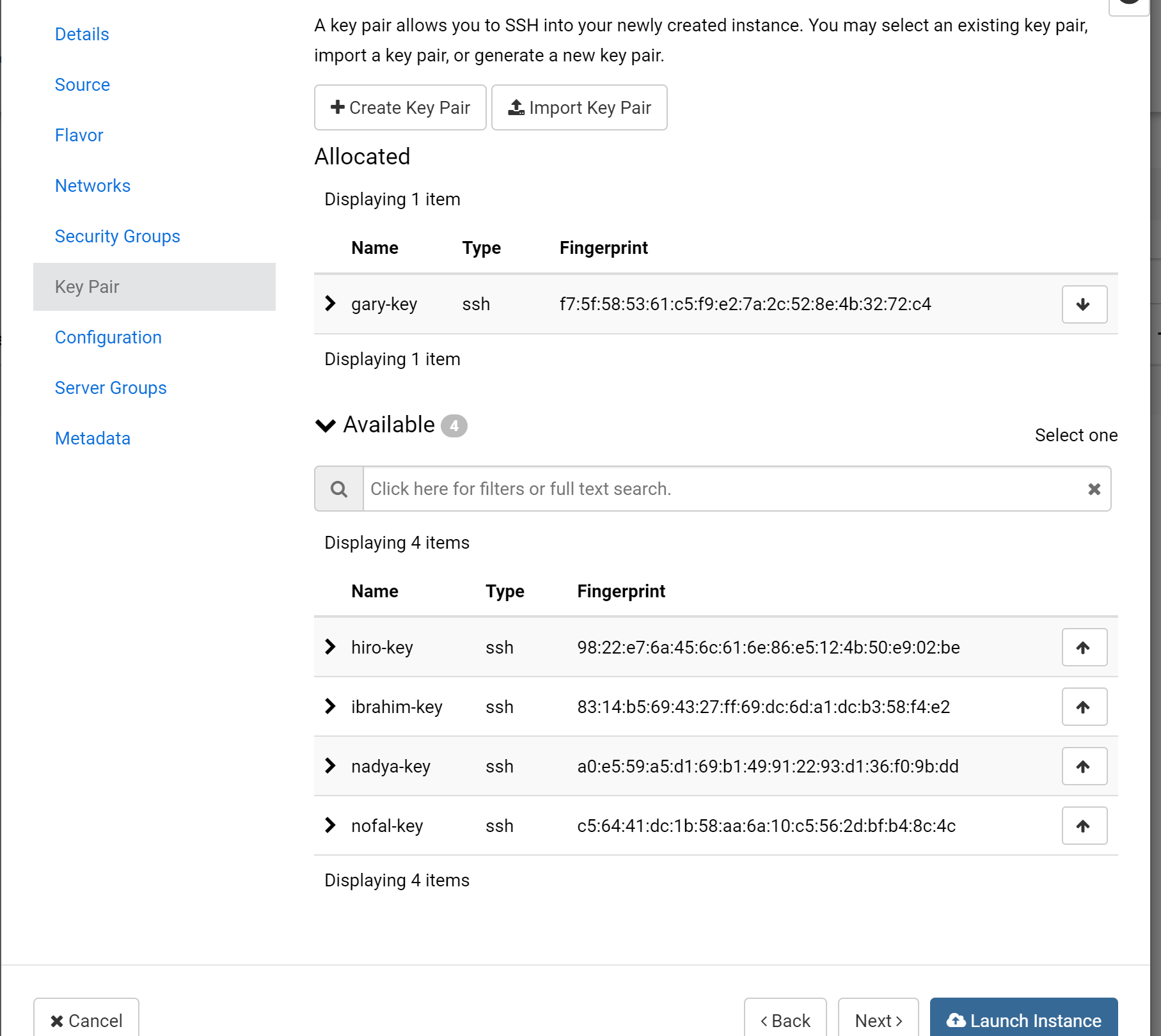
Description automatically generated with low confidence

* Selected qh2-uom-internal for networking

Graphical user interface, application

Description automatically generated





* Can add everyone later

Graphical user interface

Description automatically generated

* Installed Ubuntu on WSL
  + wsl --install -d ubuntu-22.04
* To run your default WSL distro, simply type wsl and hit the Enter key. Find available WSL distros by executing the following command: wsl --list --all , or simply wsl -l --all . To start a specific distro, type the command wsl --distribution <DistributionName> or wsl -d <DistributionName> .
* Locate my private key and ran chmod
  + chmod 600 <filename>
* Start SSH session
  + ssh –i "C:\Users\Gary-School\OneDrive - The University of Melbourne\School\Sem 1, 2023\COMP90024\_U\_1\_SM1 - Cluster and Cloud Computing\A2\comp90024-a2-team-44\keys\gary-key.pem" [ubuntu@172.26.133.24](mailto:ubuntu@172.26.133.24)
* Need to VPN into UoM network using CISCO connect
  + Use <http://remote.unimelb.edu.au/student>
* On windows WSL, copy private key to ssh folder in WSL distro root
  + cp keyfile.pem ~/.ssh
  + Doing sudo chmod 600 keyfile.pem to it.
  + Then ssh -i keyfile.pem <user>@ip.

VM Set up

ssh -I ~/.ssh/gary-key.pem ubuntu@{IP}

Table

Description automatically generated

Create CouchDB Docker cluster locally

Couchdb

Admin

Admin

Followed this -> <https://www.geekbits.io/how-to-install-apache-couchdb-in-ubuntu-22-04/>

List directories

sudo lsblk

format external disk

sudo mkfs.ext4 -j -L datahdd /dev/vdb

sudo blkid /dev/sdb

/dev/vdb: LABEL="datahdd" UUID="6ebe12a2-a6fd-47f1-ba04-50bbfbd9d275" BLOCK\_SIZE="4096" TYPE="ext4"

/dev/disk/by-uuid/6ebe12a2-a6fd-47f1-ba04-50bbfbd9d275 /mnt/datahdd auto nosuid,nodev,nofail,x-gvfs-show,x-gvfs-name=datahdd 0 0

sudo mkdir /mnt/datahdd

sudo mount -a

sudo chown ubuntu:ubuntu -R /mnt/datahdd

START sudo systemctl start couchdb

sudo systemctl stop couchdb

1. Stop the couchdb service

sudo service couchdb stop

1. Modify the default.ini file

sudo vi /opt/couchdb/etc/default.ini

Lines to modify:

[couchdb]

database\_dir = {new location}

view\_index\_dir = {new location}

1. Copy the data

sudo rsync -av /opt/couchdb/data/ {new location}

1. Restart and check the service

sudo service couchdb start

sudo service couchdb status

Allow communication between servers and the db server

Curl

curl <http://172.26.133.24:5984/_utils/>

user and pass

admin hJsVp7ZUcr2HX6L

webapp\_dbuser

fEP4iuHF3kurSgV

harvestor\_dbuser

6wg3AsSWXYg2Adt

Edit ssh keys

nano /home/ubuntu/.ssh/authorized\_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCxvaO9AIv3PbUzilXkNs0/8g0F6Par5b5+kDYZde2kUTszSs+LZs7CMidZWab96uimacOA519Msbt9OPXOr3oXwv/v6u5CoLYOU3JWjNccyf5gcUspVNNigBVlwbnnqUtLrCjJycbW63zG74cqznA6lgQoQPoDTkwHto9jA2JgQBqPfDRkHi8YDv8DEEcu7hB3KaekARrb6NtiENx0IaTnrOsg6Yos19eIQK9nqi6BQfIg+tQtNkHumJm/flLNbbBTheTgMK6+yl1Vp+x8zJTOQ0uxLF8AiFj3/h/VcstUvyjXWgQ7Xj8hbfB6eCVvfwd+1A2hQr+KET8df1GNSDYZ Generated-by-Nova

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDZ4WhlSKB73n/Gj7kYMCxAJVur1EMKcz0rykav5aB2uf3QPWnmXTlRxdAk9MT3IEdbI2QZWbIeIQgyXnvZXmRfR1FcUYa7tK9Gen3xOlGmhFEL5Enq/Zy9T7b0RnMYZ0faCjFbQeAldtNHj9iUD+Ctsm4XTEynN7Pjqi5ZcWZhzAJ0IJ0tRa6ohsXvJOZhNSL8XEafWhgtxR7YW8DtLVK8Xw+MftKV3VrVsxOpO4FP7t0OzYFYinjKredULl9nRf1W/dI15rmxQmNl3JGaZtoJoBe3kUJ7c37BamFnYWIXpLu1+f/ZbZy5G1ifrgrCr+IqMg1Nr1R65J2mCs0LwPGp Generated-by-Nova

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDvbiQ/ljnDT/ZnljAMhESjxml4RCVMV/tB+7kXVEY3f1JmFgNDQrQRb02TUvmgl4BuV16CsMMkyQV6H6FufD3Be/OWZaL5p1Kx5sKiWsrbOEvVBZTrHsuNY5muWN2meqqFuf1qHm9t9v47P+qIt0im7ZEwtl14UV+tXPOrja/G52owytM8tYRictm5vcRb04Az0MYeX8Cm20kVo4SZX7xzLxVvOadl113Y81QWArnMLsK967LRshBID1AvV12N6FHm5q8mdOpM2mWISVs2CaNC4SjFi+0rXlGj2zcePNBvxJcYVNMulrzVZDoWPyStw6fQ+iY7Lq/04I5jG0k6nGEV Generated-by-Nova

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCycpzIKiarvvtr8JGQ2FEFXiQIVKlWVxuNp/op/s9z2G1yyroyQM9ScrM+DarwrZ0o1bb88S7YEHgMzo8/y8A7E9G4A8Ajc4oOQIvWT03UeZauEtYa1u/+SAsuUolIksGgyWJ6eExYOvq9j3Y7LspLFlOF+kmc6b4aykWVf1SMVGbl4jnPDDLAjGPz3Hn7r+xNL4/rMp/RgPhfTgYelEYrdNR6NF4miExT7jVBfxpWKwK/QDz7EjVbO0NREJe8uWuMRe7Ky15hOd9NVdsEtAMGWavVQob1XatqGTghtMzOyR1MkgTmaXlKqjlzHHWuRLd3zV9s8EcV5w2dnJ4KYrfz Generated-by-Nova

Attaching volume for harvestor

/dev/vdb: LABEL="datahdd" UUID="544d69e7-f1cf-4221-b664-56b04ca33a14" BLOCK\_SIZE="4096" TYPE="ext4"

/dev/disk/by-uuid/544d69e7-f1cf-4221-b664-56b04ca33a14 /mnt/datahdd auto nosuid,nodev,nofail,x-gvfs-show,x-gvfs-name=datahdd 0 0

Download twitter data

curl <https://www.dropbox.com/s/r6l4ke6h858bzph/twitter-huge.json.zip?dl=1> -O -L -J

[prod-harvestors-ubuntu](https://dashboard.cloud.unimelb.edu.au/project/instances/f32c62b7-6bee-4528-86ef-fbcf447082f0/) 172.26.128.40 (login with ubuntu@172.26.128.40)

[prod-webapp-ubuntu](https://dashboard.cloud.unimelb.edu.au/project/instances/635f9a1d-ec81-4185-af50-20e8438bbf43/) 172.26.132.72 (login with ubuntu@172.26.128.40)

[prod-couchdb-ubuntu](https://dashboard.cloud.unimelb.edu.au/project/instances/9f41cbe9-33a8-4430-9b37-6b3ee45198eb/) 172.26.133.24 (login with ubuntu@172.26.128.40)

Set up DB Cluster for extra points!

export declare nodes=(172.17.0.1 172.17.0.2 172.17.0.3)

export masternode=`echo ${nodes} | cut -f1 -d' '`

export declare othernodes=`echo ${nodes[@]} | sed s/${masternode}//`

export size=${#nodes[@]}

export user='admin'

export pass='team44Admin'

export VERSION='3.2.1'

export cookie='a192aeb9904e6590849337933b000c99'

docker pull ibmcom/couchdb3:${VERSION}

docker create\

--name couchdb172.17.0.1\

--env COUCHDB\_USER=admin\

--env COUCHDB\_PASSWORD=team44Admin\

--env COUCHDB\_SECRET=a192aeb9904e6590849337933b000c99\

--env ERL\_FLAGS="-setcookie \"a192aeb9904e6590849337933b000c99\" -name \"couchdb@$172.17.0.1 \""\

ibmcom/couchdb3:3.2.1

docker create\

--name couchdb172.17.0.2\

--env COUCHDB\_USER=admin\

--env COUCHDB\_PASSWORD=team44Admin\

--env COUCHDB\_SECRET=a192aeb9904e6590849337933b000c99\

--env ERL\_FLAGS="-setcookie \"a192aeb9904e6590849337933b000c99\" -name \"couchdb@$172.17.0.2 \""\

ibmcom/couchdb3:3.2.1

docker create\

--name couchdb172.17.0.3\

--env COUCHDB\_USER=admin\

--env COUCHDB\_PASSWORD=team44Admin\

--env COUCHDB\_SECRET=a192aeb9904e6590849337933b000c99\

--env ERL\_FLAGS="-setcookie \"a192aeb9904e6590849337933b000c99\" -name \"couchdb@$172.17.0.3 \""\

ibmcom/couchdb3:3.2.1

declare conts=(`docker ps --all | grep couchdb | cut -f1 -d' ' | xargs -n${size} -d'\n'`)

for cont in "${conts[@]}"; do docker start ${cont}; done

couch="-H Content-Type:application/json -X PUT http://$user:$pass@172.17.0.2:5984"; \

curl $couch/photon; curl https://raw.githubusercontent.com/ermouth/couch-photon/master/photon.json | \

curl $couch/photon/\_design/photon -d @- ; curl $couch/photon/\_security -d '{}' ; \

curl $couch/\_node/\_local/\_config/csp/attachments\_enable -d '"false"' ; \

curl $couch/\_node/\_local/\_config/chttpd\_auth/same\_site -d '"lax"' ;

Install nginx reverse proxy

1. docker run --name docker-nginx -p 80:5984 nginx

server {

listen 80;

server\_name example.com;

location / {

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header Host $http\_host;

proxy\_pass "http://127.0.0.1:8080";

}

}

0cf413fc4108

docker exec -it <mycontainer> bash

Verify replication

curl -X GET <http://admin:admin@172.17.0.2:5984/_all_dbs>

curl -X PUT http://172.17.0.2:5984/\_config/admins/webapp\_dbuser -d \”fEP4iuHF3kurSgV\”

Authenticate to couchdb server

Create admin users

curl -s -X PUT "http://172.17.0.2:5984/\_node/couchdb@172.17.0.2/\_config/admins/webapp\_dbuser" -d '"fEP4iuHF3kurSgV"' -u admin:admin

curl -s -X PUT "http://172.17.0.3:5984/\_node/couchdb@172.17.0.3/\_config/admins/webapp\_dbuser" -d '"fEP4iuHF3kurSgV"' -u admin:admin

curl -s -X PUT "http://172.17.0.4:5984/\_node/couchdb@172.17.0.4/\_config/admins/webapp\_dbuser" -d '"fEP4iuHF3kurSgV"' -u admin:admin

curl -s -X PUT "http://172.17.0.2:5984/\_node/couchdb@172.17.0.2/\_config/admins/harvestor\_dbuser" -d '"6wg3AsSWXYg2Adt"' -u admin:admin

curl -s -X PUT "http://172.17.0.3:5984/\_node/couchdb@172.17.0.3/\_config/admins/harvestor\_dbuser" -d '"6wg3AsSWXYg2Adt"' -u admin:admin

curl -s -X PUT "http://172.17.0.4:5984/\_node/couchdb@172.17.0.4/\_config/admins/harvestor\_dbuser" -d '"6wg3AsSWXYg2Adt"' -u admin:admin

# Docker Swarm Install

Following this -> <https://computingforgeeks.com/how-to-install-docker-swarm-on-ubuntu/>

INT\_NAME="eth1" #replace accordingly

HOST\_IP=$(ip addr show $INT\_NAME | grep "inet\b" | awk '{print $2}' | cut -d/ -f1)

Run the below command to initialize Docker swarm node on the manager.

sudo docker swarm init --advertise-addr $HOST\_IP

|  |
| --- |
| Swarm initialized: current node (6epne4lxi2ea3zwkaqckb9rui) is now a manager.  To add a worker to this swarm, run the following command:  Sudo docker swarm join --token SWMTKN-1-5f26eu5poii34f7ugonewqhenm70pbww5kjnh9cxqfqqqha5q2-31agisl6g0y92p7yupmsb19ds 172.26.132.72:2377  To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions. |

Open up ports on MRC security groups

Port 2377 TCP for communication with and between manager nodes

Port 7946 TCP/UDP for overlay network node discovery

Port 4789 UDP (configurable) for overlay network traffic

Successfully joined swarm

Verify cluster from the manager node

sudo docker node ls

ER STATUS ENGINE VERSION

9ke3vkf4jsdy71brrctsl3yg6 prod-database-ubuntu Ready Active 23.0.1

j956hnue2vl72lvwh61tqzapw prod-harvestors-ubuntu Ready Active 23.0.1

6epne4lxi2ea3zwkaqckb9rui \* prod-webapp-ubuntu Ready Active Leade r 23.0.1

Test nginx web server

sudo docker service create --name web-server --publish 8080:80 nginx:1.13-alpine

Let’s create a service Nginx web server to run on default http port 80, and then expose it to the port 8080 on the host.

Query ports in use

Cat /etc/services

Check status of services

**sudo docker service ls**

Scale up

We are going to make 3 replicas of the web-server service so that it is accessible on the manager and the two worker nodes

sudo docker service scale web-server=3

**verify with sudo docker service ls**

to remove docker swarm service

sudo docker service rm helloworld

add user to docker users to avoid using sudo

sudo usermod -aG docker ${USER}

newgrp docker

Added 100GB to DB cluster

Ansible install on WSL

<https://www.ansiblepilot.com/articles/how-to-install-ansible-in-windows-10-wsl-windows-subsystem-for-linux/>

# Ansible

# Deploy portainer

ansible-playbook install-portainer.yaml

A screen shot of a computer program

Description automatically generated with low confidence

Build VuieJS App

docker build -t frontend-docker-vuejs .

docker images

above to list images

docker run -it -p 8081:8080 -d --name frontend frontend-docker-vuejs

Containerized our app

Deploy to docker swarm

<https://docs.docker.com/get-started/swarm-deploy/>

check for docker swarm

docker system info

docker swarm init

deploy app to swarm

docker stack deploy -c frontend-app.yaml frontend

check status of swarm

docker service ls

remove service using

docker stack rm demo

Install keychain to forward ssh key to remote server

Scale front-end service

sudo docker service scale frontend=3

on worker nodes

install npm

clone repo

sudo iptables -n -t nat -L DOCKER-INGRESS

docker service create \

--name frontend \

--publish 8020:80 \

--replicas 3

frontend-docker-vuejs