3a-DO.Cloud.Computing

November 2, 2024

1 Digital Ocean cloud computing

1.1 Overview of steps

- Setup (one time)
  - Create a DO account
  - Create a public/private SSH key
  - Add the key to your DO account
- Each time
  - Launch an instance configured with the public SSH key
  - SSH into the instance with or without a tunnel
- even better
  - Store a JSON object with the options
  - Run a curl command using the JSON to launch instances

1.2 Launch, query, destory instances via REST API

API Reference

1.2.1 Launching an instance

Create a token

DO Tokens

Then use it in this curl command

```
DO_TOKEN="{insert here}"
SSH_fingerprint="ee:ae:c4:1c:c3:dd:33:70:77:38:ff:01:0f:42:95:0c"
DO_TS=$( date +%s )  # Timestamp to ID the droplet

curl -s -X POST -H 'Content-Type: application/json' \
    -H "Authorization: Bearer ${DO_TOKEN}" \
    -d '{"name":"ubuntu-temp-via-curl-'${DO_TS}'",
        "size":"s-1vcpu-512mb-10gb",
        "region":"sfo3",
        "ssh_keys": [
          "'${SSH_fingerprint}'"
        ],
        "image":"ubuntu-22-04-x64",
        "vpc_uuid":"981e091e-2d0b-4773-8354-*966a14c68412*"}' \
```

```
    "https://api.digitalocean.com/v2/droplets" |
tee /tmp/droplet.${DO_TS}.json
```

Run Jupyter(?)

Get its IP address

```
DO_ID=$( jq .droplet.id /tmp/droplet.${DO_TS}.json )
echo ${DO_ID}
```

```
curl -s -X GET \
  -H "Authorization: Bearer ${DO_TOKEN}" \
  "https://api.digitalocean.com/v2/droplets/${DO_ID}" |
  tee /tmp/droplet.ID-${DO_ID}.json

DO_IPs=$( jq -r .droplet.networks.v4[].ip_address /tmp/droplet.ID-${DO_ID}.json )
echo ${DO_IPs}
```

Pick one

```
DO_IP=...
```

1.2.2 SSH into the instance

```
ssh -L 5150:localhost:5150 root@${DO_IP}
```

1.2.3 Configuring an instance

1.2.4 Upgrade and reboot

```
  {
sleep 10
export DEBIAN_FRONTEND=noninteractive
apt-get update
apt-get -y dist-upgrade
}
{ sleep 5 ; reboot ; } & exit
```

1.2.5 Install Docker

```
export DEBIAN_FRONTEND=noninteractive
apt-get update
apt-get install -y apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -
yes | add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stabl
apt-get update
apt-cache policy docker-ce
apt-get install -y docker-ce
systemctl status docker
```

1.2.6 Run Jupyter Lab

See Jupyter Lab in Docker

```
SHARED=/root/datascience
mkdir -p "${SHARED}"
docker \
    run \
    -d \
    -p :5150:8888 \
    -e JUPYTER_ENABLE_LAB=yes \
    -v "${SHARED}":/home/jovyan/shared \
    -w /home/jovyan/shared \
    --name jupyter-lab \
    rwcitek/jupyter-notebook:latest

host=192.168.1.8          # On the Mac ( the IP of any interface it )
host=127.0.0.1            # On a remote cloud instance using ssh tunneling ( -L :5150:127.0.0.1

while true; do
  token=$( docker container logs --since 5s jupyter-lab 2>&1 | grep -m1 -o token=.* )
  [ "${token}" ] && echo -e "\n\n\nhttp://${host}:5150/lab?${token}\n\n\n" && break
  sleep 2
done
```

1.3 Destroy an instance

BEWARE: There is no confirmation. This destroys the instance immediately.

```
DO_ID={insert ID}

curl -s -I -X DELETE \
  -H "Authorization: Bearer ${DO_TOKEN}" \
  "https://api.digitalocean.com/v2/droplets/${DO_ID}" |
  tee /tmp/droplet.delete.ID-${DO_ID}.txt
```

1.4 Creating a serverless function

Here's a link to creating a Docker container to enable creating a serverless function in Digital Ocean.

DO Serverless Function

[ ]: