 # Zerodha Ops Task

Task1 ## Description This is a sample `Go` application which connects to Redis. The app increments a Redis `counter` on an incoming request.

Create a Dockerfile

```
FROM golang:alpine AS builder
WORKDIR /app
COPY . .
RUN go build -o app .

FROM alpine:latest
WORKDIR /app
COPY --from=builder /app .
CMD ["./app"]
```

```
REPOSITORY                                                                    TAG       IMAGE ID       CREATED         SIZE
us-central-1-docker.pkg.dev/apigeeproject-391817/my-repository/sampletest     latest    bda5b6490b3e   2 days ago      132MB
us-central1-docker.pkg.dev/apigeeproject-391817/my-repository/sampletest      latest    bda5b6490b3e   2 days ago      132MB
sampletest                                                                    latest    bda5b6490b3e   2 days ago      132MB
ops-interview-task-master-app                                                 latest    21dba5c3b78c   5 days ago      14.5MB
python1                                                                       test      17f38e7b86d6   10 days ago     68.5MB
```

Use docker run command to run the container

#Task 2

Create a `docker-compose.yml` for the app which includes the following:
- `redis` service, with data directory mounted.
- `app` service, ensuring that it has a dependency on the Redis service starting correctly.
- `nginx` service acting as a reverse proxy for the app.
Bonus: Implement SSL using self-signed certificates.

Let's generate a self signed ssl certificate for our app for domain mayankzerodha.key.
1. Generate pvt key for making self signed certificate.
$ openssl genpkey -algorithm RSA -out mayankzerodha.key

2. Generate a Self-Signed Certificate:

```
ff02::2 ip6-allrouters
unthinkable-lap-0286@PG02R0JG:~/Downloads/ops-interview-task-master$ ls
docker-compose.yml  go.mod  main.go   mayankzerodha.key  playbook.yml  vagrant
Dockerfile          go.sum  Makefile  nginx.conf         README.md     zerodha.crt
```

openssl req -new -x509 -key dikshant.zeerodha.com -out zerodha.crt -days 365

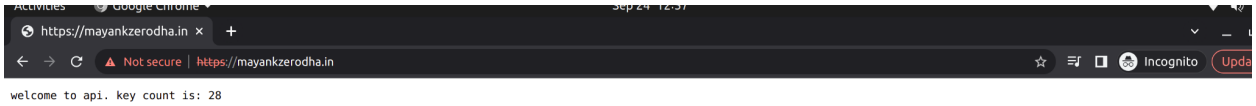
Use the private key to create a self-signed SSL certificate

Nginx.conf file

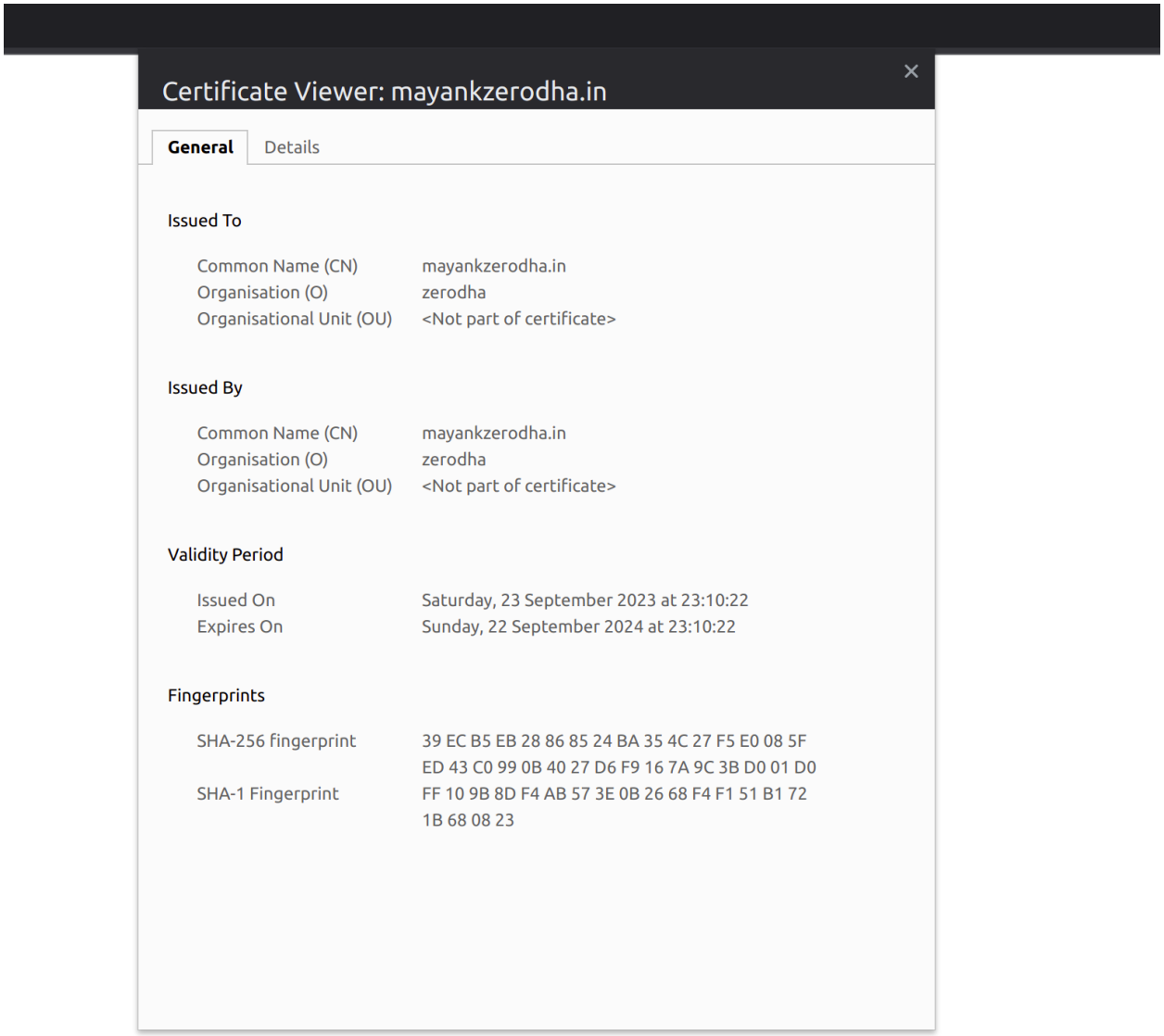
```
unthinkable-lap-0286@PG02R0JG:~/Downloads/ops-interview-task-master$ cat nginx.conf
events { worker_connections 1024; }

http {
    server {
        listen 80;
        listen 443 ssl;
        server_name mayankzerodha.in; # Replace with your domain name

        ssl_certificate /etc/nginx/ssl/zerodha.crt;
        ssl_certificate_key /etc/nginx/ssl/mayankzerodha.key;

        location / {
            proxy_pass http://app:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
    }
}
unthinkable-lap-0286@PG02R0JG:~/Downloads/ops-interview-task-master$
```

Docker-compose file

```
unthinkable-lap-0286@PG02R0JG:~/Downloads/ops-interview-task-master$ cat docker-compose.yml
version: '3'
services:
  app:
    build: .
    depends_on:
      - redis
    environment:
      - DEMO_APP_ADDR=0.0.0.0:8080
      - DEMO_REDIS_ADDR=redis:6379

  redis:
    image: redis:alpine
    volumes:
      - redis_data:/data

  nginx:
    image: nginx:alpine
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
      - ./mayankzerodha.key:/etc/nginx/ssl/mayankzerodha.key
      - ./zerodha.crt:/etc/nginx/ssl/zerodha.crt
    ports:
      - 80:80
      - 443:443

volumes:
  redis_data:
```



Certification details

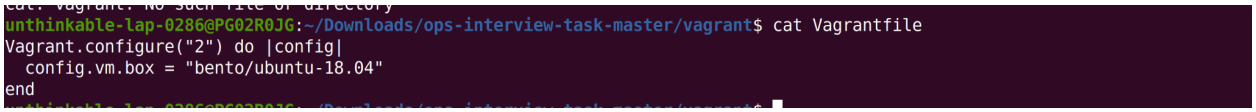


Done!!

Task3

Write a bash script to set up a [Vagrant box](https://vagrant.io) with Ubuntu. Ensure the script has error checks and is idempotent.

Vagrant file



Vagrant.sh

```
#!/bin/bash
set -e

# check if Vagrant is installed
if ! command -v vagrant &> /dev/null
then
    echo "Vagrant could not be found. Please install Vagrant before running this script."
    exit
fi

# name of the Vagrant box
BOX_NAME="bento/ubuntu-18.04"

# check if the box is already added
if ! vagrant box list | grep -q "$BOX_NAME"; then
    echo "Adding Vagrant box $BOX_NAME..."
    vagrant box add "$BOX_NAME"
fi

# Check if a Vagrantfile exists in the current directory
if [ ! -f Vagrantfile ]; then
    echo "Creating Vagrantfile for box $BOX_NAME..."
    echo "Vagrant.configure(\"2\") do |config|" > Vagrantfile
    echo "  config.vm.box = \"$BOX_NAME\"" >> Vagrantfile
    echo "end" >> Vagrantfile
fi

# check if the box is already running
if ! vagrant status | grep -q running; then
    echo "Starting Vagrant box $BOX_NAME..."
    vagrant up --provider virtualbox
else
    echo "Vagrant box $BOX_NAME is already running."
fii
```

vagrant.sh

```
#!/bin/bash

# Update your system
sudo apt-get update

# Install required software
sudo apt-get install -y unzip curl

# Download Vagrant
cd /tmp
curl -O https://releases.hashicorp.com/vagrant/2.2.14/vagrant_2.2.14_x86_64.deb

# Install Vagrant
sudo dpkg -i vagrant_2.2.14_x86_64.deb

# Check Vagrant version
vagrant --version
```

The above script I have used to provision 3 ubuntu machines on virtualbox with vagrant to setup VM.

Task4:

Using Ansible provision the VM to:
- Setup the hostname of VM as `demo-ops`.
- Create a user `demo`.
- Harden the security:
- Disable root login.
- Setup a basic firewall (e.g., UFW) allowing only specific ports.
- Configure `sysctl` for sane defaults. (For eg: increasing open files limit)
- Configure sysctl for sane defaults. For each sysctl parameter changed
: - Document the change.
- Provide a brief justification or explanation (2-3 lines) detailing why this specific change was made and its implications.
- Set the system's timezone to "Asia/Kolkata".
- Install Docker and Docker-Compose.
- Configure Docker Daemon to have sane defaults. For eg: keep logs size in check.
- Deploy the `docker-compose.yml` in `/etc/demo-ops` and start the services.

Below is the ansible playbook configuration which will do the above task on aws instance.

NOTE: for sysctl I have configured below 2 parameters

1. fs.file-max sets an upper limit on the total number of open files system-wide. When this limit is reached, processes may be unable to open additional files until existing ones are closed.
2. Kernel.pid_max this parameter defines the maximum ID that can be assigned to a process.
The above two parameters I have set through ansible and are working fine.

```
- name: Launch EC2 instance, get public IP, and install Nginx
  hosts: localhost
  tasks:
    - amazon.aws.ec2_instance:
        name: "ansible-zerodha-instance"
        access_key: Access_key
        secret_key: Secret_key
        key_name: "mayankansible"
        vpc_subnet_id: subnet-0409a2f8a86047118
        instance_type: t3.micro
        security_group: default
        network:
          assign_public_ip: true
        image_id: ami-0f5ee92e2d63afc18
        tags:
          Environment: Testing
      register: ec2
    - name: Create SSH Group to login dynamically to EC2 Instance
      add_host:
        hostname: "3.108.65.233"
        ansible_ssh_private_key_file: /home/unthinkable-lap-0286/Downloads/mayankansible.pem
        groupname: ec2_server
      with_items: ec2.instances

    - name: Wait for SSH to come up
      wait_for:
        host: "3.108.65.233"
        port: 22
        state: started
      with_items: ec2.instances

    - name: Install Nginx
      become: yes
      ansible.builtin.shell: "sudo apt-get update && sudo apt-get install -y nginx"
      delegate_to: "3.108.65.233"
      remote_user: "ubuntu"

    - name: Allow SSH and enable UFW
      become: yes
```

```
    ansible.builtin.shell: |
      sudo ufw allow OpenSSH
      sudo ufw --force enable
    delegate_to: "3.108.65.233"
    remote_user: "ubuntu"

  - name: Increase Open Files Limit
    become: yes
    ansible.builtin.sysctl:
      name: fs.file-max
      value: 65536
    delegate_to: "3.108.65.233"
    remote_user: "ubuntu"

  - name: Change Kernel PID Max
    become: yes
    ansible.builtin.sysctl:
      name: kernel.pid_max
      value: 65535
    delegate_to: "3.108.65.233"
    remote_user: "ubuntu"


  - name: Set Timezone to Asia/Kolkata
    ansible.builtin.shell: |
      sudo timedatectl set-timezone Asia/Kolkata
    delegate_to: "3.108.65.233"
    remote_user: "ubuntu"

  - name: Install docker and docker compose
    ansible.builtin.shell: |
      sudo apt-get update -y
      sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin -y
      sudo chmod 777 /var/run/docker.sock
    delegate_to: "3.108.65.233"
    remote_user: "ubuntu"

  - name: Copy All data to ubuntu
    ansible.builtin.copy:
      src: /home/unthinkable-lap-0286/Downloads/ops-interview-task-master/
      dest: /home/ubuntu
      owner: ubuntu
      group: ubuntu
```

```
    mode: u+rw,g-wx,o-rwx
  delegate_to: "3.108.65.233"
  remote_user: "ubuntu"

- name: Deploy Dockercompsoe
  ansible.builtin.shell: |
    cd /home/ubuntu
    docker compose up -d
  delegate_to: "3.108.65.233"
  remote_user: "ubuntu"
```

```
PLAY [Launch EC2 instance, get public IP, and install Nginx] ******************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************
ok: [localhost]

TASK [amazon.aws.ec2_instance] ************************************************************************************************************
ok: [localhost] => {"changed": false, "changes": [], "instance_ids": ["i-0eadba8110a104337"], "instances": [{"ami_launch_index": 0, "archi
tecture": "x86_64", "block_device_mappings": [{"device_name": "/dev/sda1", "ebs": {"attach_time": "2023-09-24T06:12:50+00:00", "delete_on_
termination": true, "status": "attached", "volume_id": "vol-0990453c4d2657373"}}], "capacity_reservation_specification": {"capacity_reserv
ation_preference": "open"}, "client_token": "f2a202c0774b4bf2ba7cbf8b42fc4e9a", "cpu_options": {"core_count": 1, "threads_per_core": 2}, "
current_instance_boot_mode": "legacy-bios", "ebs_optimized": false, "ena_support": true, "enclave_options": {"enabled": false}, "hibernati
on_options": {"configured": false}, "hypervisor": "xen", "image_id": "ami-0f5ee92e2d63afc18", "instance_id": "i-0eadba8110a104337", "insta
nce_type": "t3.micro", "key_name": "mayankansible", "launch_time": "2023-09-24T06:12:49+00:00", "maintenance_options": {"auto_recovery": "
default"}, "metadata_options": {"http_endpoint": "enabled", "http_protocol_ipv6": "disabled", "http_put_response_hop_limit": 1, "http_toke
ns": "optional", "instance_metadata_tags": "disabled", "state": "applied"}, "monitoring": {"state": "disabled"}, "network_interfaces": [{"
association": {"ip_owner_id": "amazon", "public_dns_name": "ec2-3-108-65-233.ap-south-1.compute.amazonaws.com", "public_ip": "3.108.65.233
"}, "attachment": {"attach_time": "2023-09-24T06:12:49+00:00", "attachment_id": "eni-attach-0f8c5297c177490ca", "delete_on_termination": t
rue, "device_index": 0, "network_card_index": 0, "status": "attached"}, "description": "", "groups": [{"group_id": "sg-018fa5639e7e9fc5d",
 "group_name": "default"}], "interface_type": "interface", "ipv6_addresses": [], "mac_address": "0a:c6:0b:c0:2a:54", "network_interface_id
": "eni-02553cc44d9da0908", "owner_id": "903335868070", "private_dns_name": "ip-172-31-13-187.ap-south-1.compute.internal", "private_ip_ad
dress": "172.31.13.187", "private_ip_addresses": [{"association": {"ip_owner_id": "amazon", "public_dns_name": "ec2-3-108-65-233.ap-south-
1.compute.amazonaws.com", "public_ip": "3.108.65.233"}, "primary": true, "private_dns_name": "ip-172-31-13-187.ap-south-1.compute.internal
", "private_ip_address": "172.31.13.187"}], "source_dest_check": true, "status": "in-use", "subnet_id": "subnet-0409a2f8a86047118", "vpc_i
d": "vpc-04e919058916276f6"}], "placement": {"availability_zone": "ap-south-1b", "group_name": "", "tenancy": "default"}, "platform_detail
s": "Linux/UNIX", "private_dns_name": "ip-172-31-13-187.ap-south-1.compute.internal", "private_dns_name_options": {"enable_resource_name_d
ns_a_record": false, "enable_resource_name_dns_aaaa_record": false, "hostname_type": "ip-name"}, "private_ip_address": "172.31.13.187", "p
roduct_codes": [], "public_dns_name": "ec2-3-108-65-233.ap-south-1.compute.amazonaws.com", "public_ip_address": "3.108.65.233", "root_devi
ce_name": "/dev/sda1", "root_device_type": "ebs", "security_groups": [{"group_id": "sg-018fa5639e7e9fc5d", "group_name": "default"}], "sou
rce_dest_check": true, "state": {"code": 16, "name": "running"}, "state_transition_reason": "", "subnet_id": "subnet-0409a2f8a86047118", "
```

```
TASK [Increase Open Files Limit] **********************************************************************************************************
ok: [localhost -> 3.108.65.233] => {"changed": false}

TASK [Change Kernel PID Max] **************************************************************************************************************
ok: [localhost -> 3.108.65.233] => {"changed": false}

TASK [Set Timezone to Asia/Kolkata] *******************************************************************************************************
changed: [localhost -> 3.108.65.233] => {"changed": true, "cmd": "sudo timedatectl set-timezone Asia/Kolkata\n", "delta": "0:00:00.125025"
, "end": "2023-09-24 13:08:13.958692", "msg": "", "rc": 0, "start": "2023-09-24 13:08:13.833667", "stderr": "", "stderr_lines": [], "stdou
t": "", "stdout_lines": []}
```

```
TASK [Copy All data to ubuntu] ************************************************************************************************************
changed: [localhost -> 3.108.65.233] => {"changed": true, "dest": "/home/ubuntu/", "src": "/home/unthinkable-lap-0286/Downloads/ops-interv
iew-task-master/"}

TASK [Deploy Dockercompsoe] ***************************************************************************************************************
changed: [localhost -> 3.108.65.233] => {"changed": true, "cmd": "cd /home/ubuntu\ndocker compose up -d\n", "delta": "0:00:00.599896", "en
d": "2023-09-24 13:16:07.628652", "msg": "", "rc": 0, "start": "2023-09-24 13:16:07.028756", "stderr": " Container ubuntu-nginx-1  Created
\n Container ubuntu-redis-1  Running\n Container ubuntu-app-1  Running\n Container ubuntu-nginx-1  Starting\n Container ubuntu-nginx-1  St
arted", "stderr_lines": [" Container ubuntu-nginx-1  Created", " Container ubuntu-redis-1  Running", " Container ubuntu-app-1  Running", "
 Container ubuntu-nginx-1  Starting", " Container ubuntu-nginx-1  Started"], "stdout": "", "stdout_lines": []}

PLAY RECAP ********************************************************************************************************************************
localhost                  : ok=12   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

welcome to api. key count is: 2

Done !!