

Lab 1

- I generated an RSA key pair using the `ssh-keygen` command. The private key is saved in `~/ubuntu_doc/keys/key`, and the public key is saved in `~/ubuntu_doc/keys/key.pub`. This key pair will be used for secure SSH connections.

```
engy@localhost:~/ubuntu_doc

[engy@localhost ~]$ mkdir ~/ubuntu_doc
[engy@localhost ~]$ cd ~/ubuntu_doc
[engy@localhost ubuntu_doc]$ touch Dockerfile ansible.cfg inventory my-file.yml index.html
[engy@localhost ubuntu_doc]$ mkdir keys
[engy@localhost ubuntu_doc]$ tree ~/ubuntu_doc
/home/engy/ubuntu_doc
├── ansible.cfg
├── Dockerfile
├── index.html
├── inventory
├── keys
└── my-file.yml

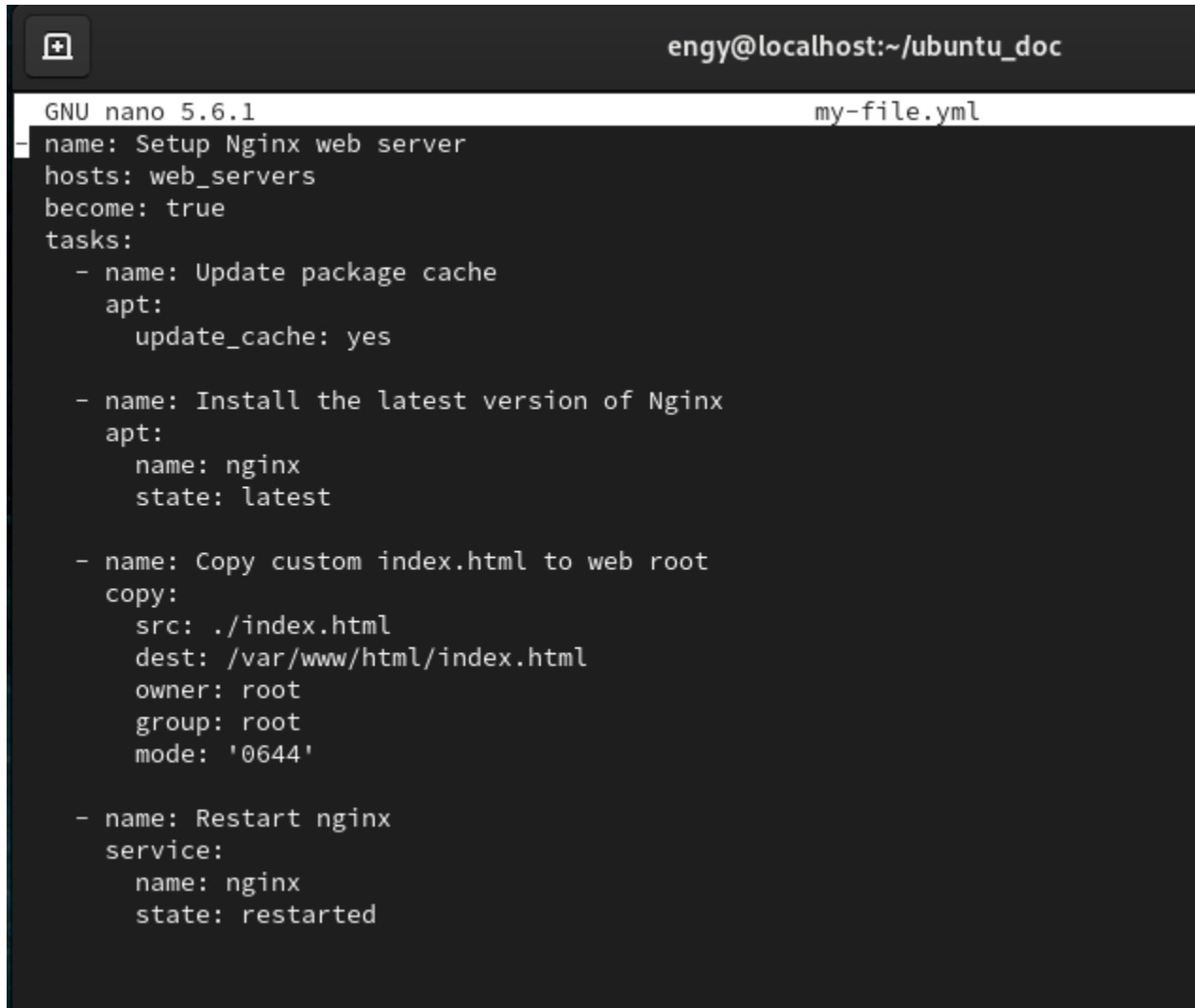
1 directory, 5 files
[engy@localhost ubuntu_doc]$ ssh-keygen -t rsa -b 2048 -f ~/ubuntu_doc/keys/key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/engy/ubuntu_doc/keys/key
Your public key has been saved in /home/engy/ubuntu_doc/keys/key.pub
The key fingerprint is:
SHA256:5J2j1u+eBNukFn030MDt5xxRxhj7eevxL+b83DXFWZ4 engy@localhost.localdomain
The key's randomart image is:
+----[RSA 2048]-----+
|          ..o++|
|          .+.=|
|          .  .o.o|
|         o . o .**|
|        S * o .EX|
|         o o . .B|
|         o = o  +.|
|        . . o O+.*|
|         o=oo+B|
+----[SHA256]-----+
```

- I successfully built the Docker image using the `docker build` command, and the image was tagged as `ubuntu_doc`. After running the container with the `docker run` command, it started successfully. The container's IP address is `172.17.0.2`, which I can use to test the setup.

```
[engy@localhost ubuntu_doc]$ [200~sudo docker build -t ubuntu_doc .
bash: [200~sudo: command not found...
[engy@localhost ubuntu_doc]$ sudo docker build -t ubuntu_doc .
[+] Building 67.9s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 260B                               0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest   1.5s
=> [auth] library/ubuntu:pull token for registry-1.docker.io      0.0s
=> [internal] load .dockerignore                                   0.1s
=> => transferring context: 2B                                       0.0s
=> CACHED [1/5] FROM docker.io/library/ubuntu:latest@sha256:80dd3c3b9c6cecb9f1667e9290 0.0s
=> [2/5] RUN apt update && apt install ssh sudo -y                64.1s
=> [3/5] RUN adduser iti                                           0.6s
=> [4/5] RUN echo "iti:123" | chpasswd                             0.3s
=> [5/5] RUN usermod -aG sudo iti                                   0.3s
=> exporting to image                                              0.8s
=> => exporting layers                                              0.7s
=> => writing image sha256:b2676f61e53499f1ae34c68b1c02ee145164cd9568c49fb4d50c8f30007 0.0s
=> => naming to docker.io/library/ubuntu_doc                       0.0s

1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for ENTRYPOINT to prevent unintended behavior related to OS signals (line 6)
[engy@localhost ubuntu_doc]$ sudo docker run -dit --name ubuntu_doc ubuntu_doc
e9a533e5942dd2082cfad5336677cb98292be5d8b50cc4f8c7f307154a871d0d
[engy@localhost ubuntu_doc]$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS      NAMES
e9a533e5942d   ubuntu_doc "/bin/sh -c 'service..." 7 seconds ago  Up 7 seconds      ubuntu_doc
[engy@localhost ubuntu_doc]$ sudo docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' ubuntu_doc
172.17.0.2
```

- The `my-file.yml` is an Ansible playbook that installs and configures Nginx on a web server. It updates the package cache, installs the latest Nginx version, copies a custom `index.html` file to the web root, and restarts the Nginx service.

A terminal window with a dark background. The title bar shows a terminal icon and the text 'engy@localhost:~/ubuntu_doc'. The terminal content shows 'GNU nano 5.6.1' and 'my-file.yml' in the top bar. The main text is an Ansible playbook for setting up Nginx. It includes tasks for updating the package cache, installing Nginx, copying a custom index.html file, and restarting the service.

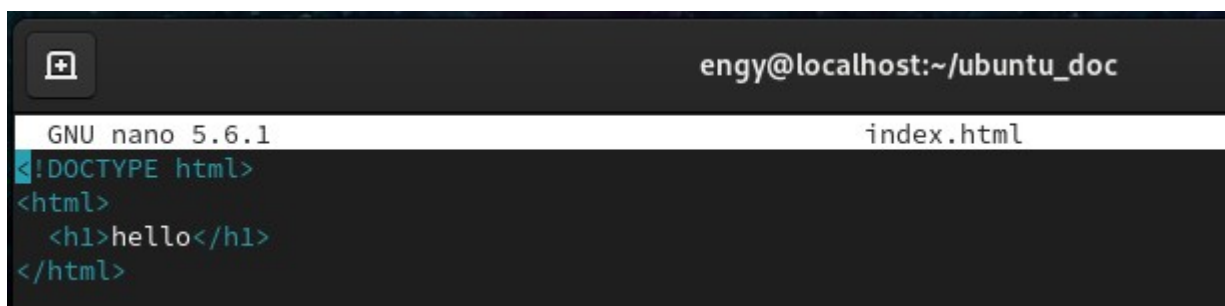
```
engy@localhost:~/ubuntu_doc
GNU nano 5.6.1 my-file.yml
- name: Setup Nginx web server
  hosts: web_servers
  become: true
  tasks:
    - name: Update package cache
      apt:
        update_cache: yes

    - name: Install the latest version of Nginx
      apt:
        name: nginx
        state: latest

    - name: Copy custom index.html to web root
      copy:
        src: ./index.html
        dest: /var/www/html/index.html
        owner: root
        group: root
        mode: '0644'

    - name: Restart nginx
      service:
        name: nginx
        state: restarted
```

- The `index.html` file contains a simple HTML structure with a heading (`<h1>`) that displays "hello".

A terminal window with a dark background. The title bar shows a terminal icon and the text 'engy@localhost:~/ubuntu_doc'. The terminal content shows 'GNU nano 5.6.1' and 'index.html' in the top bar. The main text is a simple HTML document with a single heading.

```
engy@localhost:~/ubuntu_doc
GNU nano 5.6.1 index.html
<!DOCTYPE html>
<html>
  <h1>hello</h1>
</html>
```

- The `Dockerfile` creates an Ubuntu-based image, installs `ssh` and `sudo`, adds a user `iti` with a password, grants `sudo` access to `iti`, and sets the container to start the SSH service and Bash.

```
engy@localhost:~/ubuntu_doc
GNU nano 5.6.1 Dockerfile
FROM ubuntu
RUN apt update && apt install ssh sudo -y
RUN adduser iti
RUN echo "iti:123" | chpasswd
RUN usermod -aG sudo iti
ENTRYPOINT service ssh start && bash
```

- The `ansible.cfg` file is configured to use the local `inventory` file, the private key located at `./keys/key`, and sets the remote user to `iti`. It also enables privilege escalation (`sudo`) and prompts for a password when necessary.

```
engy@localhost:~/ubuntu_doc
GNU nano 5.6.1 ansible.cfg
[defaults]
inventory = ./inventory
private_key_file = ./keys/key
remote_user = iti

[privilege_escalation]
become = true
become_ask_pass = true
```

- The `inventory` file lists a group called `web_servers`, with the IP address `172.17.0.2` as its member. This allows Ansible to target this IP address when running tasks.

```
engy@localhost:~/ubuntu_doc
GNU nano 5.6.1 inventory
[web_servers]
172.17.0.2
```

- I copied the `index.html` file to the container, restarted Nginx, and verified that the custom page with the "hello" message was displayed by using `curl` both from inside the container and from my host machine.

```
[engy@localhost ubuntu_doc]$ docker cp ~/ubuntu_doc/index.html ubuntu_doc:/root/index.html
Successfully copied 2.05kB to ubuntu_doc:/root/index.html
[engy@localhost ubuntu_doc]$ docker exec -it ubuntu_doc /bin/bash
Error response from daemon: container c4adda2f4c3d1977a3430408e5e461c5e39457cb6d0c6a3692ee4cbbc8e4e214 is not running
[engy@localhost ubuntu_doc]$ docker start ubuntu_doc
ubuntu_doc
[engy@localhost ubuntu_doc]$ docker exec -it ubuntu_doc /bin/bash
root@c4adda2f4c3d:/# cp /root/index.html /var/www/html/index.html
root@c4adda2f4c3d:/# service nginx restart
 * Restarting nginx nginx [ OK ]

root@c4adda2f4c3d:/# curl http://172.17.0.2
<!DOCTYPE html>
<html>
  <h1>hello</h1>
</html>
root@c4adda2f4c3d:/# exit
exit
[engy@localhost ubuntu_doc]$ curl http://localhost
<!DOCTYPE html>
<html>
  <h1>hello</h1>
</html>
[engy@localhost ubuntu_doc]$ curl http://172.17.0.2
<!DOCTYPE html>
<html>
  <h1>hello</h1>
</html>
```

