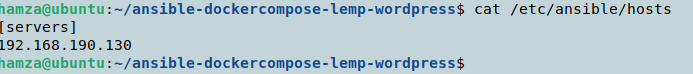


[Ansible-Docker Compose -NGINX-WordPress]

[Host a WordPress site using Docker-Compose on remote machine using Ansible.]

Open hosts file and enter IP address of remote machine like this one:



Create a docker-compose.yml file that will be executed in remote machine.

nano docker-compose.yml

Enter following text into it.

version: '3'

services:

db:

image: mysql:8.0

container\_name: db

restart: unless-stopped

env\_file: .env

environment:

- MYSQL\_DATABASE=wordpress

volumes:

- dbdata:/var/lib/mysql

command: '--default-authentication-plugin=mysql\_native\_password'

networks:

- app-network

wordpress:

depends\_on:

- db

image: wordpress:5.1.1-fpm-alpine

container\_name: wordpress

restart: unless-stopped

env\_file: .env

environment:

- WORDPRESS\_DB\_HOST=db:3306

- WORDPRESS\_DB\_USER=$MYSQL\_USER

- WORDPRESS\_DB\_PASSWORD=$MYSQL\_PASSWORD

- WORDPRESS\_DB\_NAME=wordpress

volumes:

- wordpress:/var/www/html

networks:

- app-network

webserver:

depends\_on:

- wordpress

image: nginx:1.15.12-alpine

container\_name: webserver

restart: unless-stopped

ports:

- "80:80"

volumes:

- wordpress:/var/www/html

- ./nginx-conf:/etc/nginx/conf.d

- certbot-etc:/etc/letsencrypt

networks:

- app-network

certbot:

depends\_on:

- webserver

image: certbot/certbot

container\_name: certbot

volumes:

- certbot-etc:/etc/letsencrypt

- wordpress:/var/www/html

command: certonly --webroot --webroot-path=/var/www/html --email admin@hamzasharif.com --agr>

volumes:

certbot-etc:

wordpress:

dbdata:

networks:

app-network:

driver: bridge

Save and exit this file.

Now to host a website on NGINX server, we need to set a configuration file and send it too towards remote machine.

Create a nginx.conf file.

nano nginx.conf

Open this file and add following text into it.

server {

listen 80;

listen [::]:80;

server\_name hamzasharif.com www.hamzasharif.com;

index index.php index.html index.htm;

root /var/www/html;

location ~ /.well-known/acme-challenge {

allow all;

root /var/www/html;

}

location / {

try\_files $uri $uri/ /index.php$is\_args$args;

}

location ~ \.php$ {

try\_files $uri =404;

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass wordpress:9000;

fastcgi\_index index.php;

include fastcgi\_params;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param PATH\_INFO $fastcgi\_path\_info;

}

location ~ /\.ht {

deny all;

}

location = /favicon.ico {

log\_not\_found off; access\_log off;

}

location = /robots.txt {

log\_not\_found off; access\_log off; allow all;

}

location ~\* \.(css|gif|ico|jpeg|jpg|js|png)$ {

expires max;

log\_not\_found off;

}

}

Save and exit this file too.

Now create e new file with name ansible.yml and this file will contain ansible code that will execute in remote machine.

---

- hosts: servers

remote\_user: temp

become: yes

tasks:

- name: Making a directory in remote machine to copy our required file

file:

path: ./ansible-dockercompose-lemp-wordpress

state: directory

mode: '0755'

- name: Making a directory in remote machine to copy our nginx configuration file

file:

path: ./ansible-dockercompose-lemp-wordpress/nginx-conf

state: directory

mode: '0755'

- name: Copying docker-compose file in the ~/ansible-dockercompose-lemp-wordpress

copy:

src: ~/ansible-dockercompose-lemp-wordpress/docker-compose.yml

dest: ./ansible-dockercompose-lemp-wordpress/docker-compose.yml

- name: Copying .env file in the ~/ansible-dockercompose-lemp-wordpress

copy:

src: ~/ansible-dockercompose-lemp-wordpress/.env

dest: ./ansible-dockercompose-lemp-wordpress/.env

- name: Copying nginx.conf file in the ~/ansible-dockercompose-lemp-wordpress

copy:

src: ~/ansible-dockercompose-lemp-wordpress/nginx-conf/nginx.conf

dest: ./ansible-dockercompose-lemp-wordpress/nginx-conf/nginx.conf

- name: Installing Docker

apt:

name: docker

state: latest

- name: Running docker-compose commands to complete inatallations

docker\_compose:

project\_src: ./ansible-dockercompose-lemp-wordpress

files:

- docker-compose.yml

Save and exit this file too.

Now all we need to do is create a SSH connection between the two machines.

Install SSH on both machines.

sudo apt install ssh

Enable ufw services.

sudo ufw enable

Add ssh into firewall.

sudo ufw allow ssh

This will allow ssh traffic on port 22.

To confirm write

sudo ufw status

Then try to ping remote machine.

ssh [temp@192.168.190.130](mailto:temp@192.168.190.130)

IP can be change.

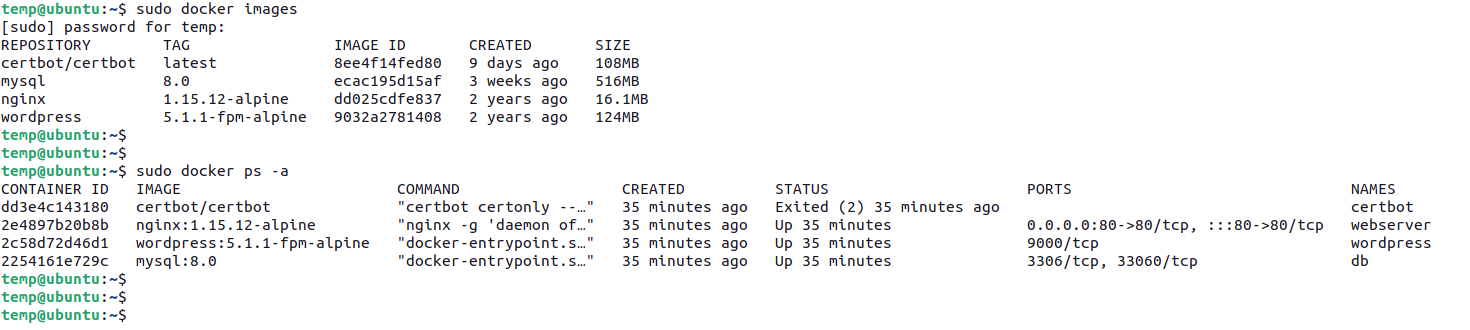
If successful, it will connect you to remote machine and will take you to the terminal of remote machine.

Now run the ansible.yml code through ansible-playbook

ansible-playbook ansible.yml --ask-pass -K

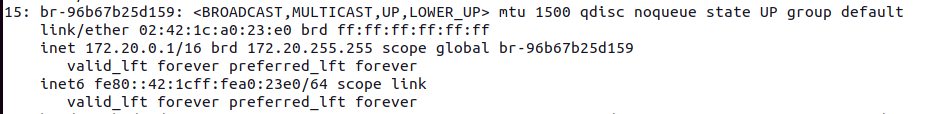
It will run and perform each task on remote machine.

Once completed, try to list images and containers in remote machine to confirm if they are working.



Now check IP address of your docker container and hit that IP in our browser of remote machine.

ip addr



Go to browser and hit IP there.

