21-Oct-2024

Internship Day - 64 Report:

Changes in Vagrant File:

1. First, edit Vagrantfile, open on wordpad or notepad

2. To assign public ip to the virtual m/c:

Remove #(hash) from these lines:-

- config.vm.network "private_network", ip: "192.168.33.10"
- config.vm.network "public_network"

Change the Memory or RAM Size in our m/c

we also increase or decrease the size of the memory and ram by removing #(hash) from lines given below:

By shell scripting we update and install nginx server and start their services:

PROVISIONING SHELL SCRIPTING

It show us updates and changes we done in the vagrantfile.

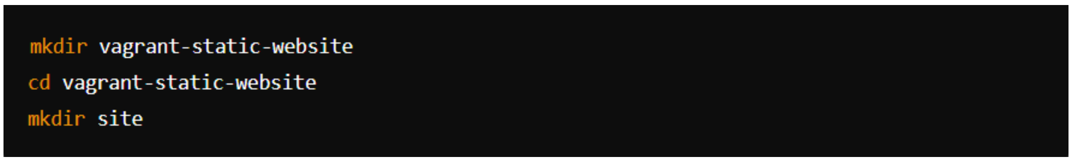
Vagrant reload --provision

22-Oct-2024

Internship Day - 65 Report:

Static websites setup using vagrant

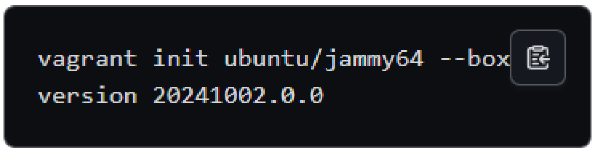
1. **Create the project directory**
   - mkdir vagrant-website
   - cd vagrant-website
   - mkdir site



```
mkdir vagrant-static-website
cd vagrant-static-website
mkdir site
```

2. **Create the Vagrantfile**

- vagrant init ubuntu/jammy64 --box-version 20241002.0.0

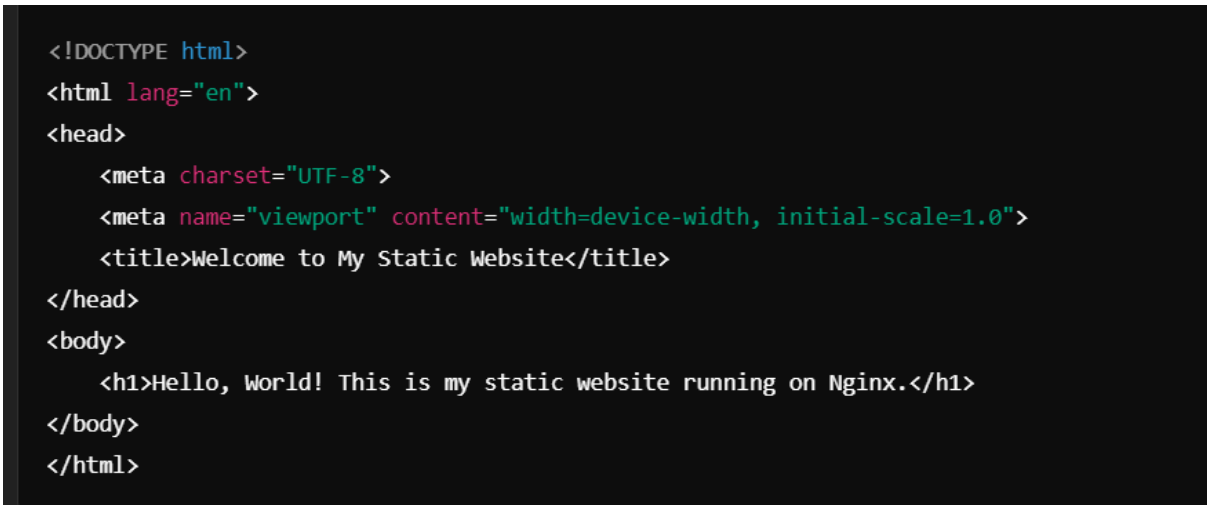


```
vagrant init ubuntu/jammy64 --box
version 20241002.0.0
```

3. **Open the Vagrantfile and replace the contents with the following**

```
    # -*- mode: ruby -*-

    # vi: set ft=ruby :

    Vagrant.configure("2") do |config|

    # Use Ubuntu 20.04 LTS as the base box

 config.vm.box = "ubuntu/focal64"

 # Define a VM name

 config.vm.hostname = "static-site"

 # Configure the network. We use port forwarding so the website is accessible via
localhost.

  config.vm.network "forwarded_port", guest: 80, host: 8080


  # Provision the virtual machine
  config.vm.provision "shell", inline: <<-SHELL
  # Update package lists
    sudo apt-get update
   # Install Nginx
    sudo apt-get install -y nginx
   # Remove the default Nginx configuration file
    sudo rm /etc/nginx/sites-enabled/default
   # Create a new Nginx configuration file for the static site
    sudo tee /etc/nginx/sites-available/static-site <<EOL
server {
    listen 80;
    server_name localhost;
    root /vagrant/site;
    index index.html;
}
EOL
   # Enable the configuration by linking it
    sudo ln -s /etc/nginx/sites-available/static-site /etc/nginx/sites-enabled/static-site
   # Restart Nginx to apply the new configuration
    sudo systemctl restart nginx
  SHELL
 # Sync the 'site' directory to /vagrant/site in the VM
  config.vm.synced_folder "./site", "/vagrant/site"
end
```

4. **Create the static website**



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome to My Static Website</title>
</head>
<body>
    <h1>Hello, World! This is my static website running on Nginx.</h1>
</body>
</html>
```

5. **Start the Vagrant environment**

   COMMAND-: vagrant up

6. **Access the website**
   Once the setup is complete, you can access your website by navigating to http://localhost:8080 in your browser.

7. **Managing the virtual machine**

- To stop the VM, run vagrant halt.
- To destroy the VM, run vagrant destroy.
- To SSH into the VM, run vagrant ssh.

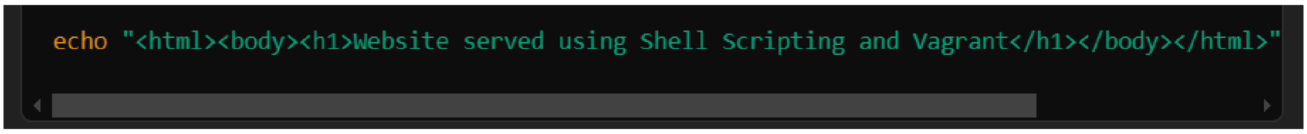
23-Oct-2024

Internship Day - 66 Report:

Website online through Shell Scripting In vagrant File

1. **Create the project directory**
   - mkdir vagrant-shell-scripted-website
   - cd vagrant-shell-scripted-website
   - mkdir site

2. **Create a basic HTML file**



```
echo "<html><body><h1>Website served using Shell Scripting and Vagrant</h1></body></html>"
```

3. **Create the Vagrantfile**

COMMAND-: vagrant init ubuntu/focal64 --box-version 20240821.0.1

Open the Vagrantfile and add the following content:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure("2") do |config|
 # Use Ubuntu 20.04 as the base box
  config.vm.box = "ubuntu/focal64"
 # Name the virtual machine
  config.vm.hostname = "shell-scripted-site"
 # Forward port 80 on the guest machine to port 8080 on the host machine
  config.vm.network "forwarded_port", guest: 80, host: 8080

  # Sync the 'site' folder to the '/vagrant/site' directory in the VM
  config.vm.synced_folder "./site", "/vagrant/site"
  # Provision the VM with a shell script
  config.vm.provision "shell", inline: <<-SHELL
   # Update the package list
    sudo apt-get update
   # Install Nginx
    sudo apt-get install -y nginx
   # Remove the default Nginx config file if it exists
    sudo rm -f /etc/nginx/sites-enabled/default
   # Create a new Nginx configuration to serve the static site
    sudo bash -c 'cat <<EOF > /etc/nginx/sites-available/static-site
server {
    listen 80;
    server_name localhost;
    root /vagrant/site;
    index index.html;
    location / {
        try_files \$uri \$uri/ =404;
    }
}
EOF'
   # Enable the new site configuration
    sudo ln -s /etc/nginx/sites-available/static-site /etc/nginx/sites-enabled/static-site
   # Restart Nginx to apply changes
    sudo systemctl restart nginx
   # Ensure Nginx starts on boot
    sudo systemctl enable nginx
  SHELL
End
```

4. **Start the Vagrant environment**

COMMAND-: vagrant up

5. Access your website

Search on browser-: **http://localhost:8080**

6. Managing the virtual machine

- To stop the VM, run vagrant halt.
- To destroy the VM, run vagrant destroy.
- To SSH into the VM, run vagrant ssh.

24-Oct-2024

Internship Day - 67 Report:

Wordpress setup using vagrant:

1) mkdir vagrant-wordpress
2) cd vagrant-wordpress
3) vagrant init ubuntu/jammy64
4) vagrant up
5) vagrant ssh
6) In Google Search, Install and configure wordpress in Ubuntu
7) Link-: https://ubuntu.com/server/docs/how-to-install-and-configure-wordpress
8) Follow all the steps of given above link and copy all the steps on virtual machine which we are using.
9) After the completion of these all steps
10) Type ip addr show command in your linux
11) And copy the local machine ip and paste in your browser
12) wordpress is live

25-Oct-2024

Internship Day- 68 Report:

Automate Deploy Wordpress using Vagrantfile

1. **Create the project directory**
   - mkdir vagrant-wordpress-automation
   - cd vagrant-wordpress-automation

2. **Initialize Vagrant with the Ubuntu 22.04 (Jammy Jellyfish) box**
   - vagrant init ubuntu/jammy64

3. **Edit the Vagrantfile**
   - Edit Vagrantfile and adding a new shell script for Install and configure wordpress in Ubuntu

```
config.vm.provision "shell", inline: <<-SHELL
    apt-get update
 #install apache2
    apt-get install -y apache2
 #install wget and tr utilities
    sudo apt-get install -y wget coreutils
 #install MySQL
    sudo apt-get install -y mysql-server
    sudo mysql -e "CREATE DATABASE chd;"
    sudo mysql -e "CREATE USER 'pankaj'@'localhost' IDENTIFIED BY '123';"
    sudo mysql -e "GRANT ALL PRIVILEGES ON chd.* TO 'pankaj'@'localhost';"
    sudo mysql -e "FLUSH PRIVILEGES;"



  #Install PHP and required extension
```

```
   sudo apt-get install -y php libapache2-mod-php php-mysql
 #Download Wordpress and Setup it
   cd /var/www/html
   sudo rm -rf wordpress
   sudo wget https://wordpress.org/latest.tar.gz
   sudo tar -xzf latest.tar.gz
   sudo mv wordpress/* .
   sudo rm -rf wordpress latest.tar.gz
   sudo chown -R www-data:www-data /var/www/html
   sudo chmod -R 755 /var/www/html
 #remove Default Apache Page if it exists
   sudo rm -f /var/www/html/index.html
 #create wp-config using sample config File
   sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
   sudo sed -i "s/database_name_here/chd/" /var/www/html/wp-config.php
   sudo sed -i "s/username_here/pankaj/" /var/www/html/wp-config.php
   sudo sed -i "s/password_here/123/" /var/www/html/wp-config.php
#configure Apache for Wordpress
   sudo a2enmod rewrite
   sudo systemctl restart apache2
```

4. Provision the Vagrant environment

- vagrant up --provision
- vagrant ssh

5. Access WordPress

- Now, Copy the local m/c ip and paste in your browser
- Wordpress is live through Automation