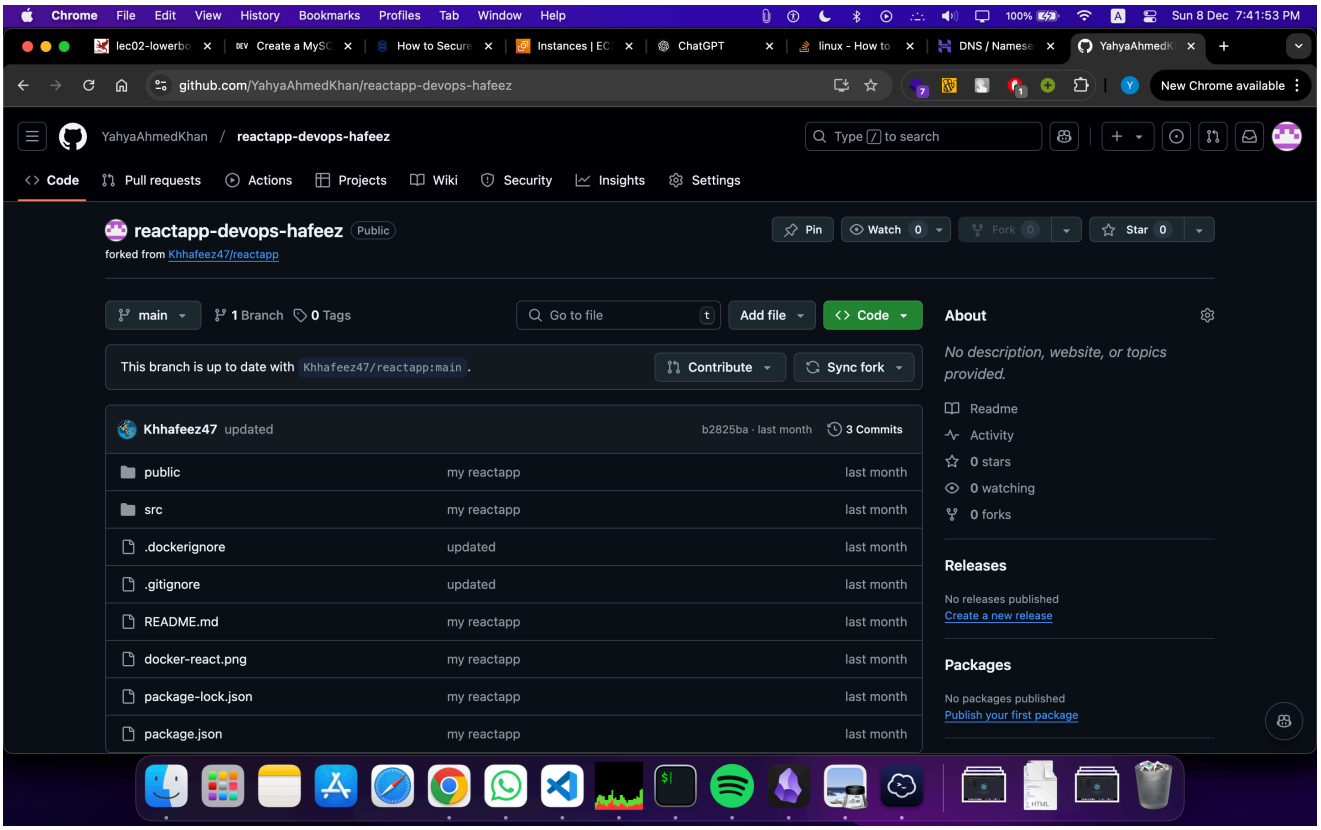
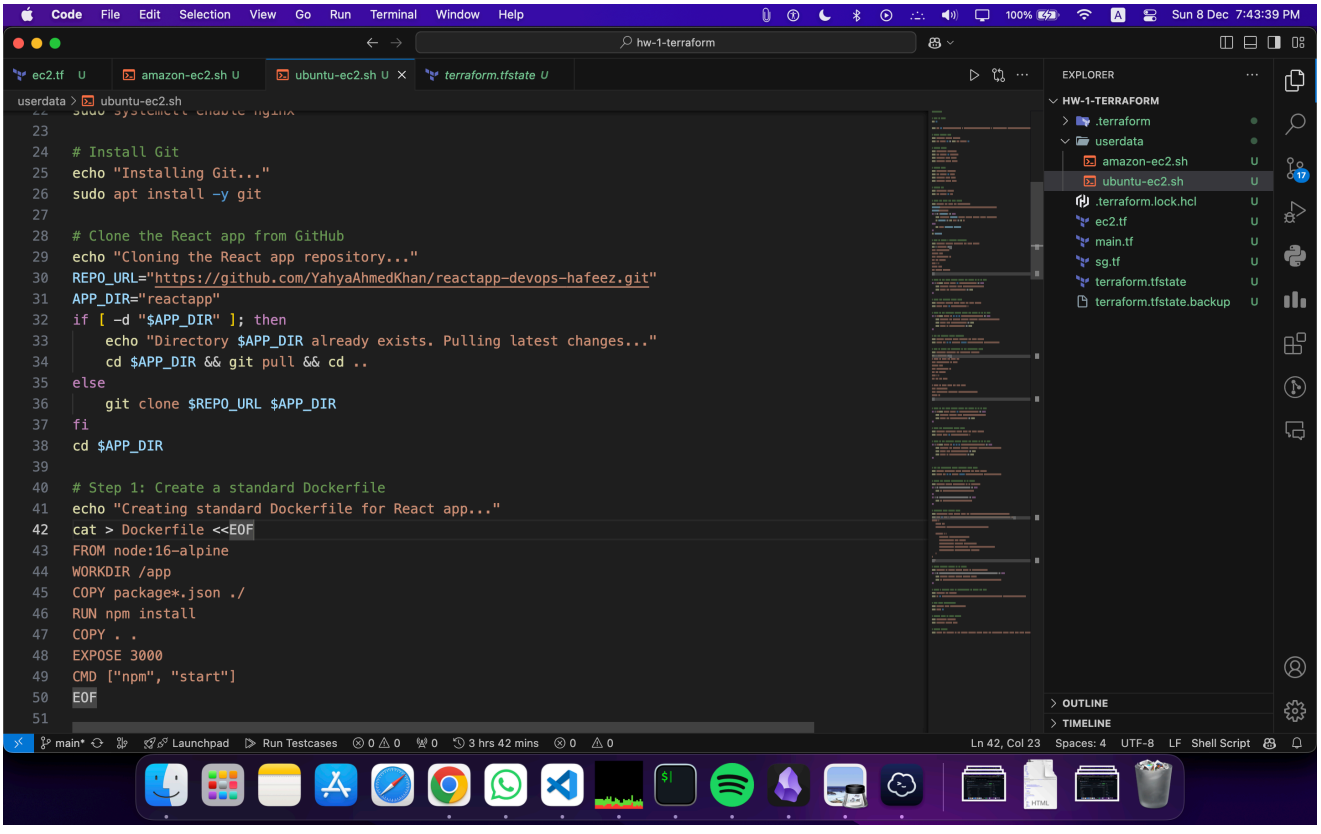
Terraform HW-1 Yahya Ahmed Khan 24442

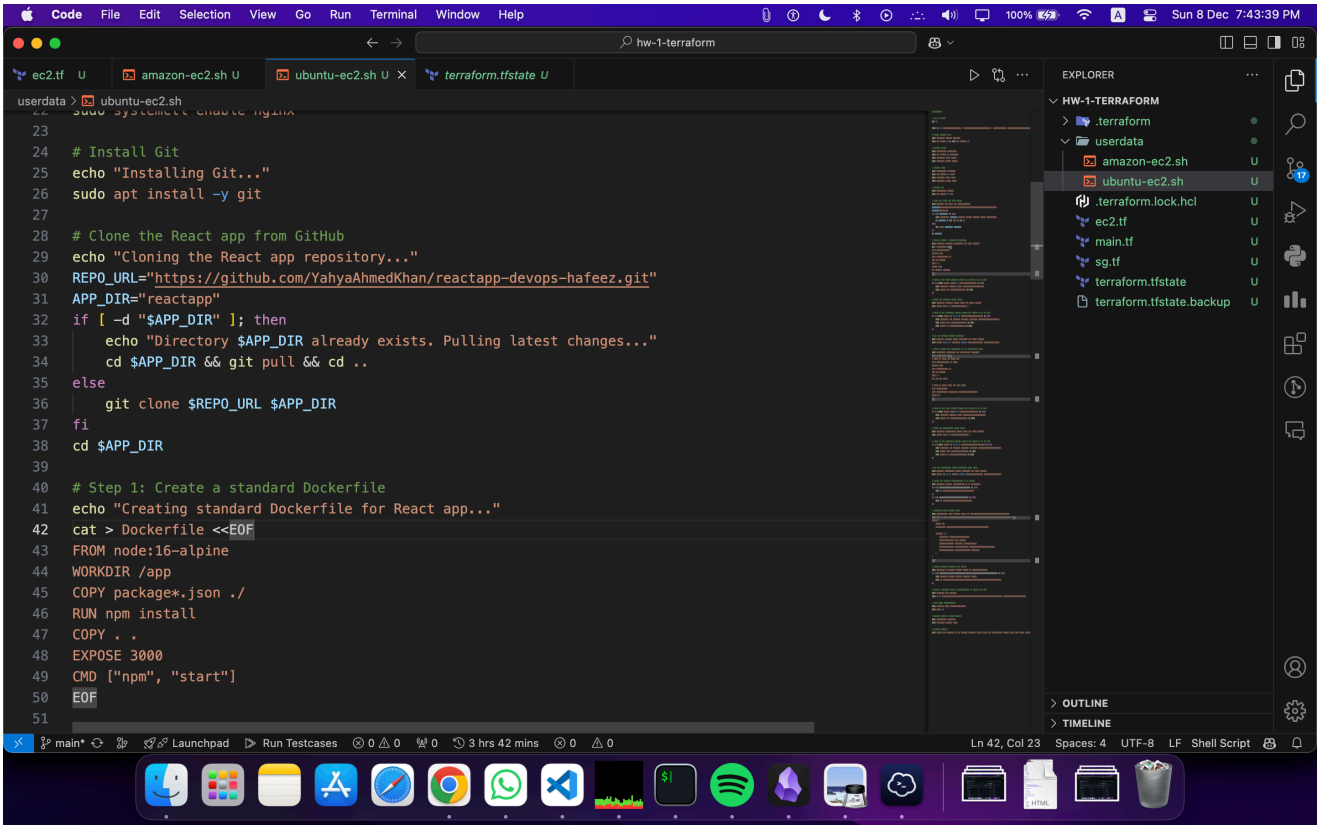
Forking the Repository



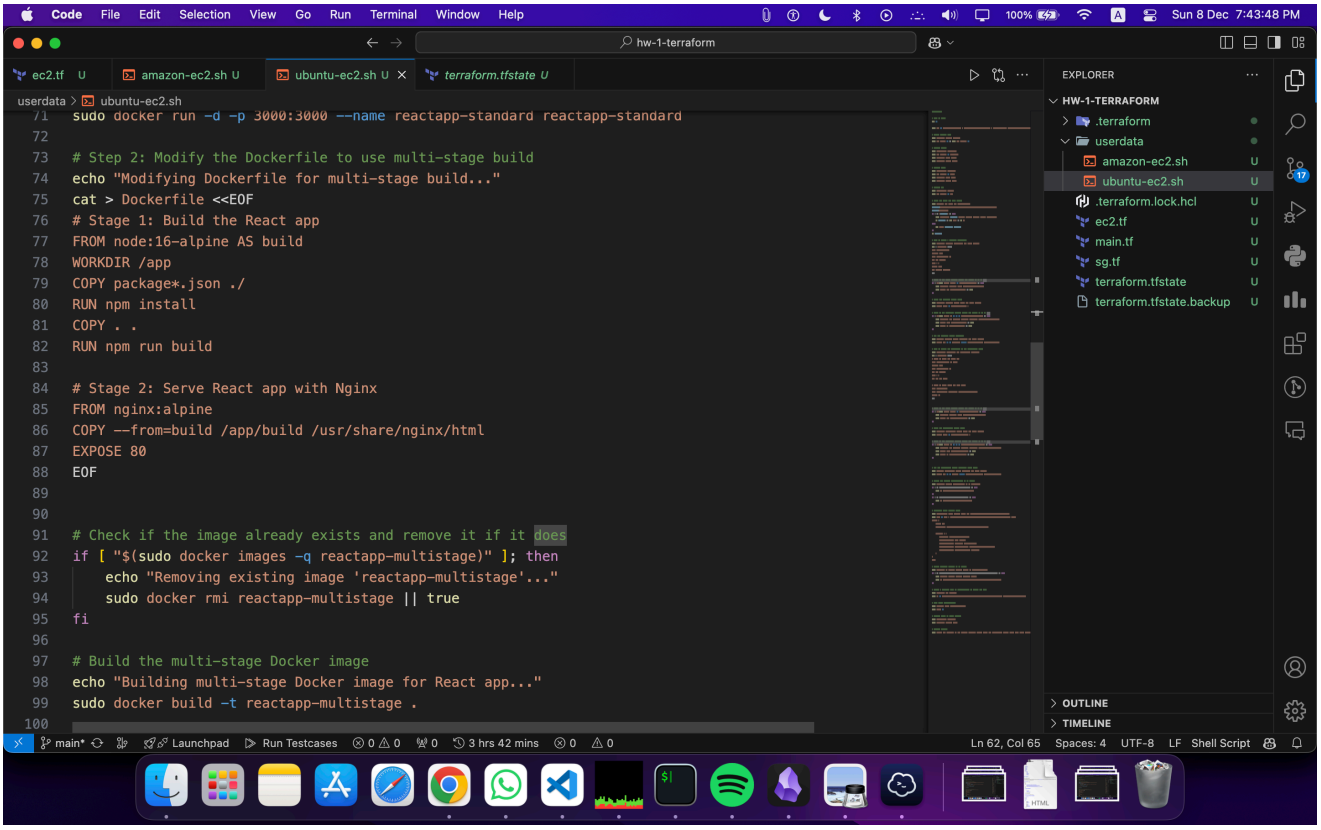
Cloning the Repository



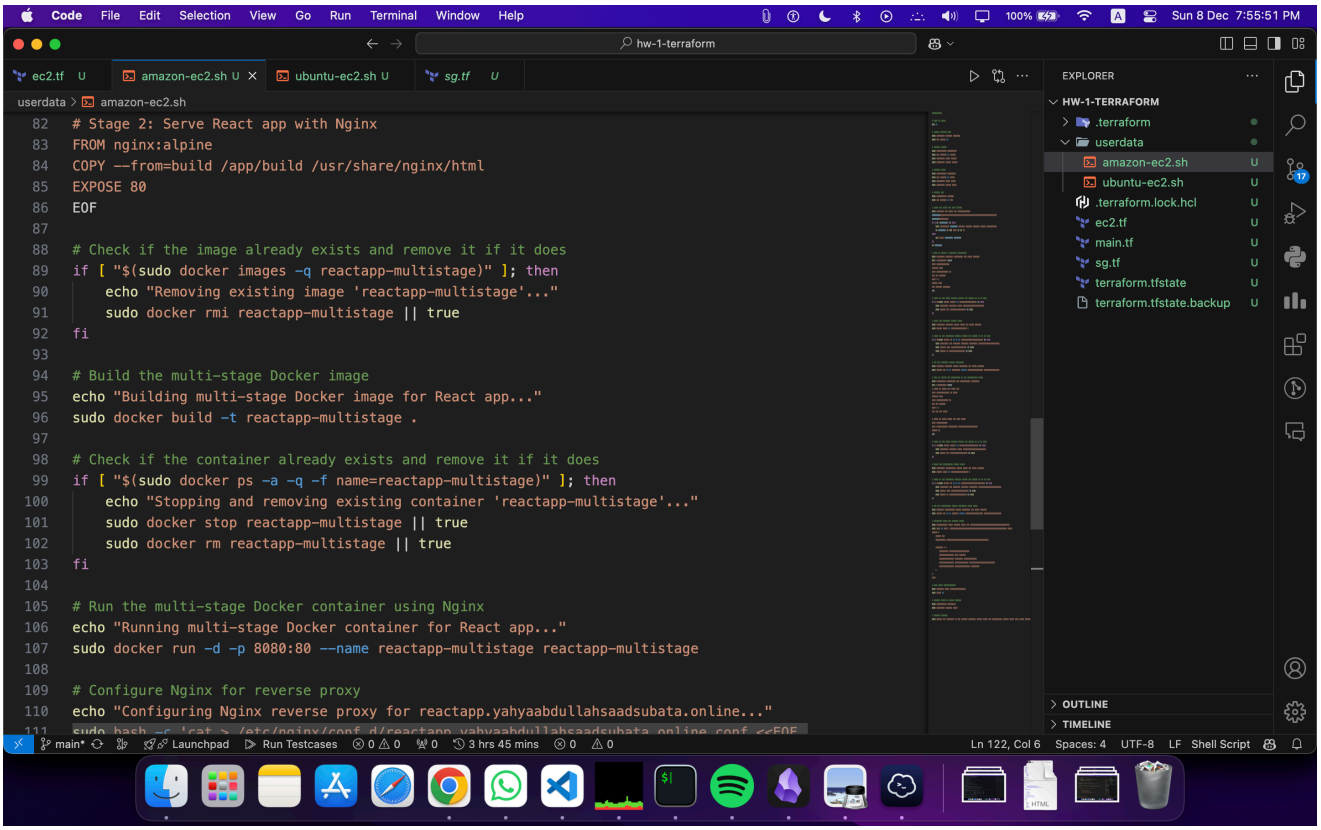
Creating a Dockerfile to Run React App



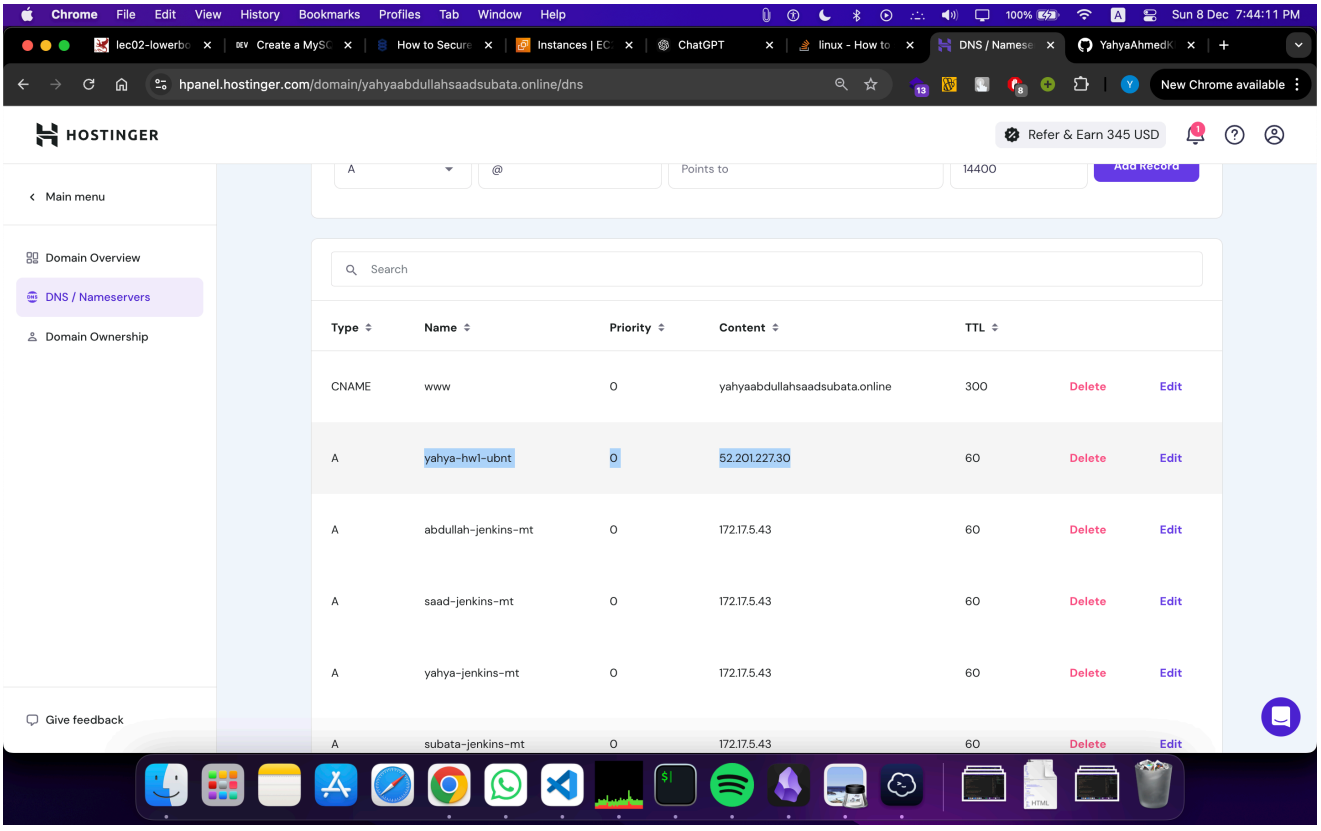
Modifying Dockerfile for Multistage Build

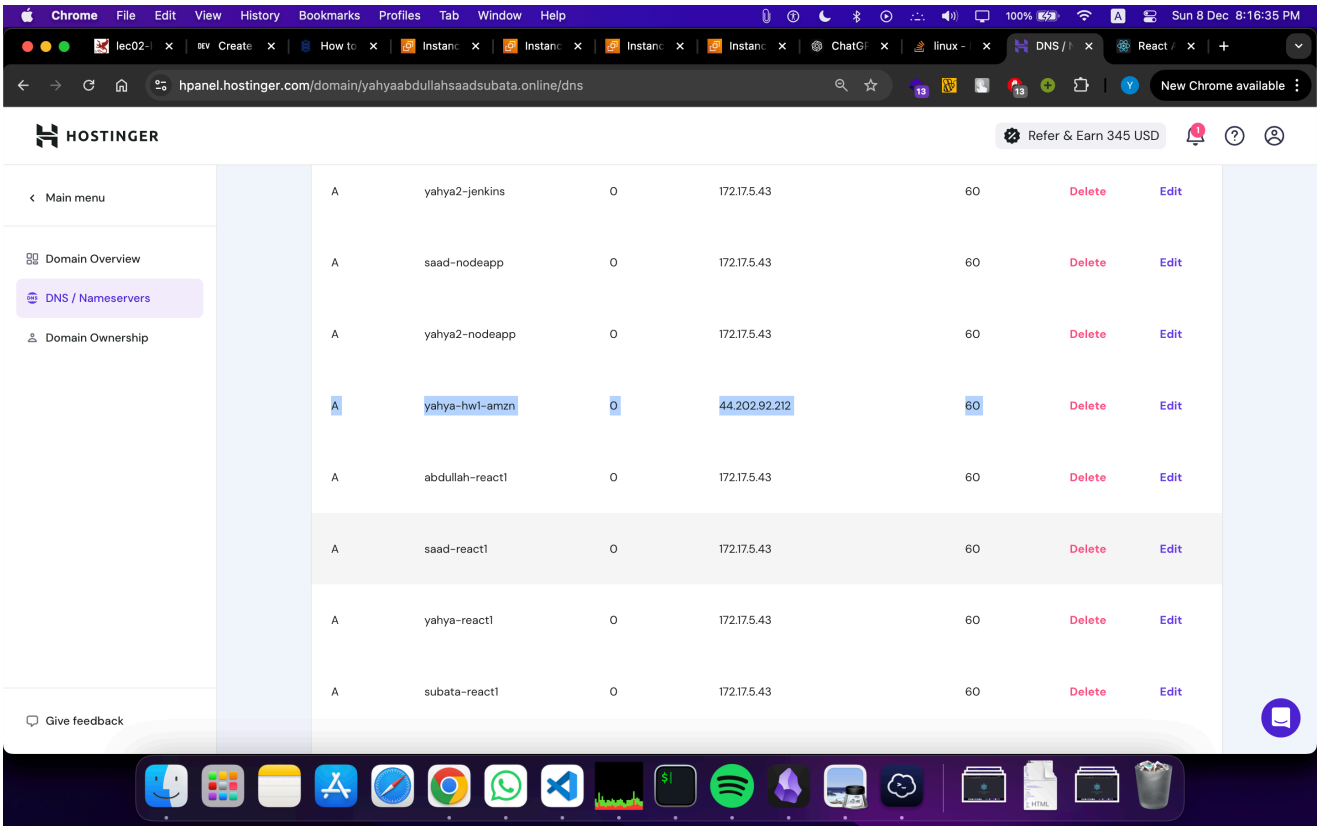


Running Multistage Docker Build in Nginx Container

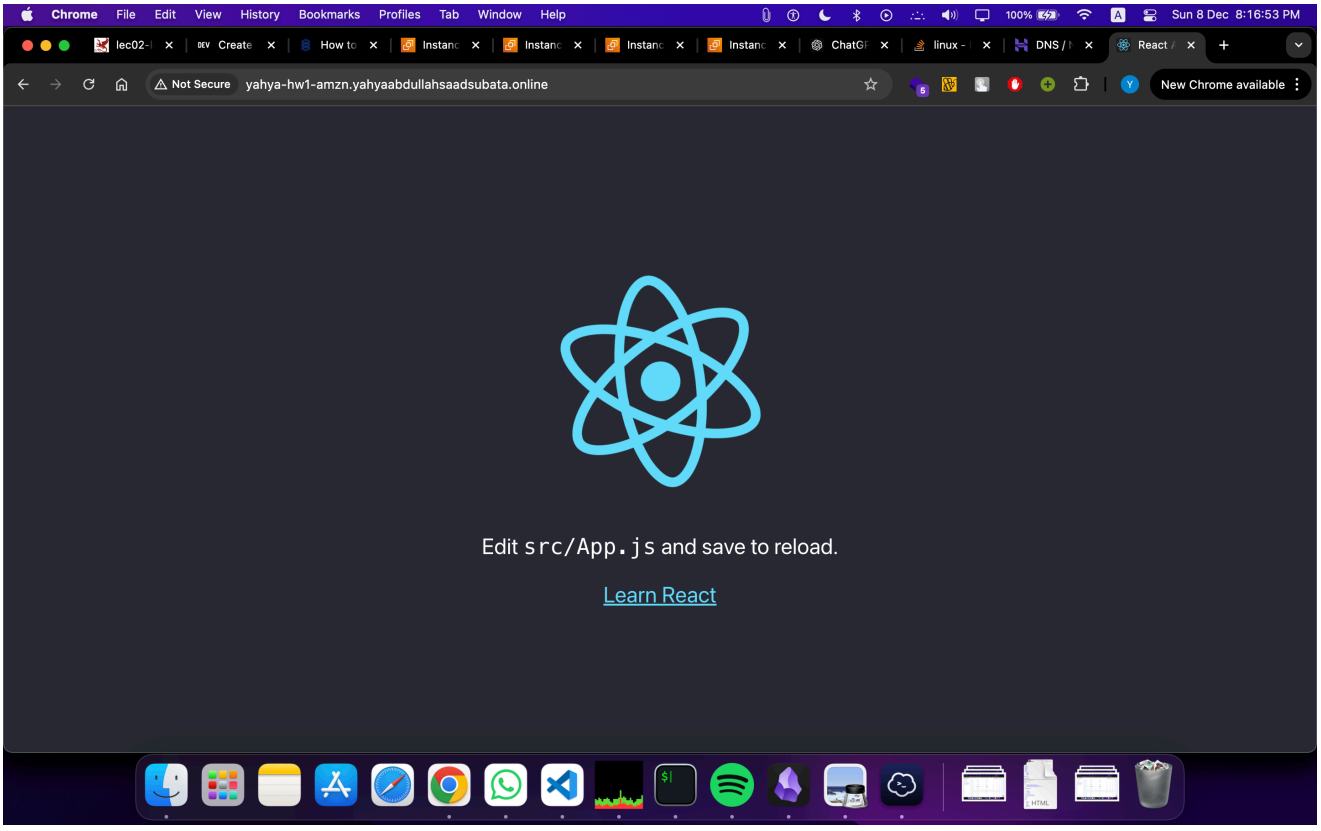


Applying DNS for React App

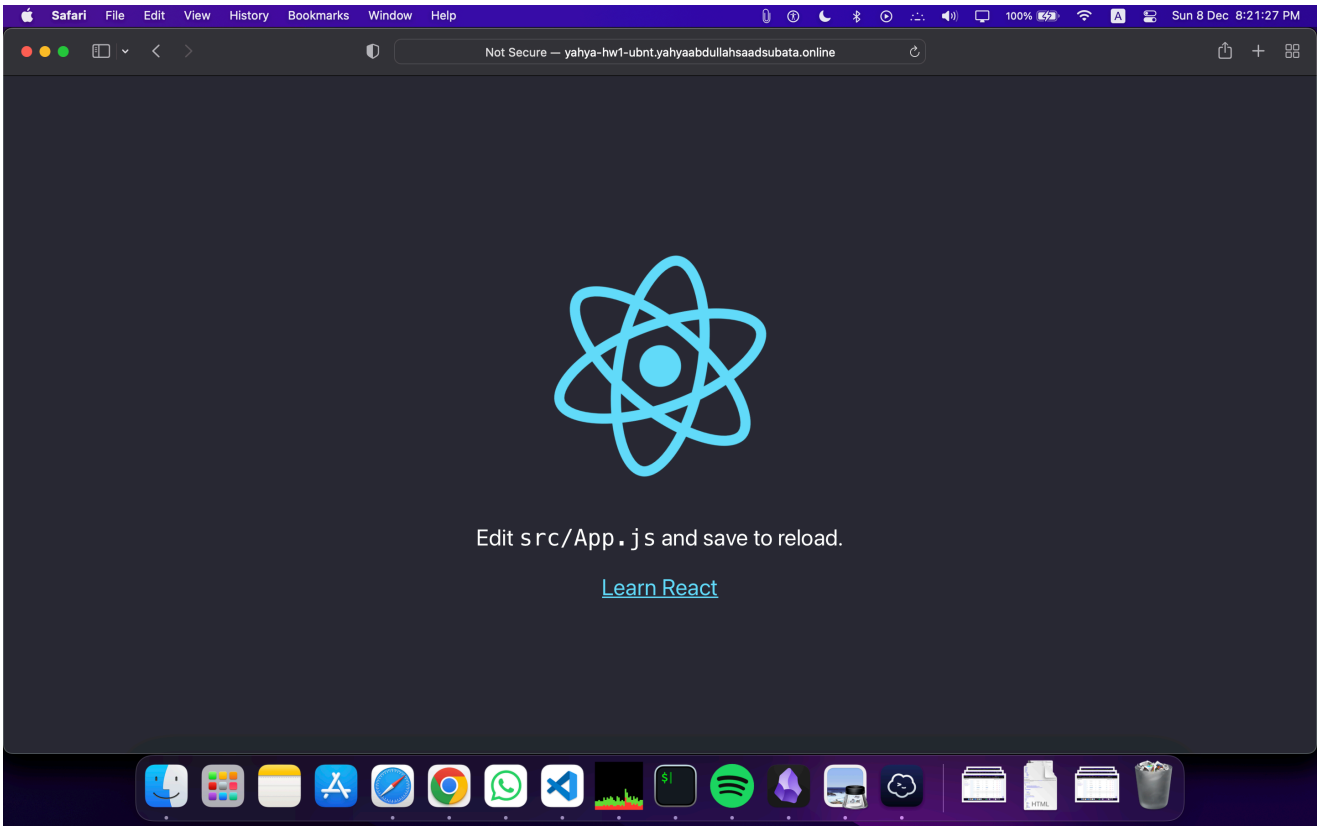




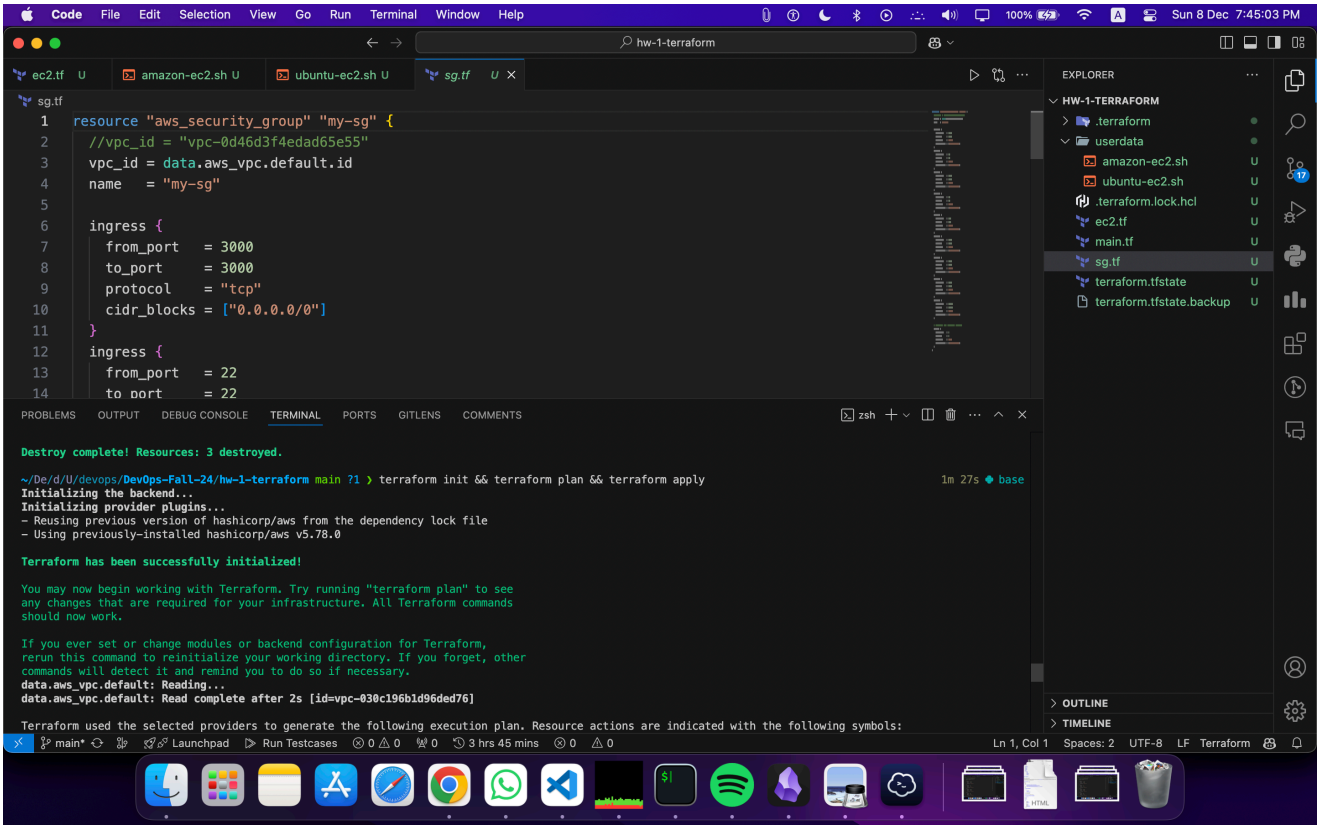
Amazon site



Ubuntu site

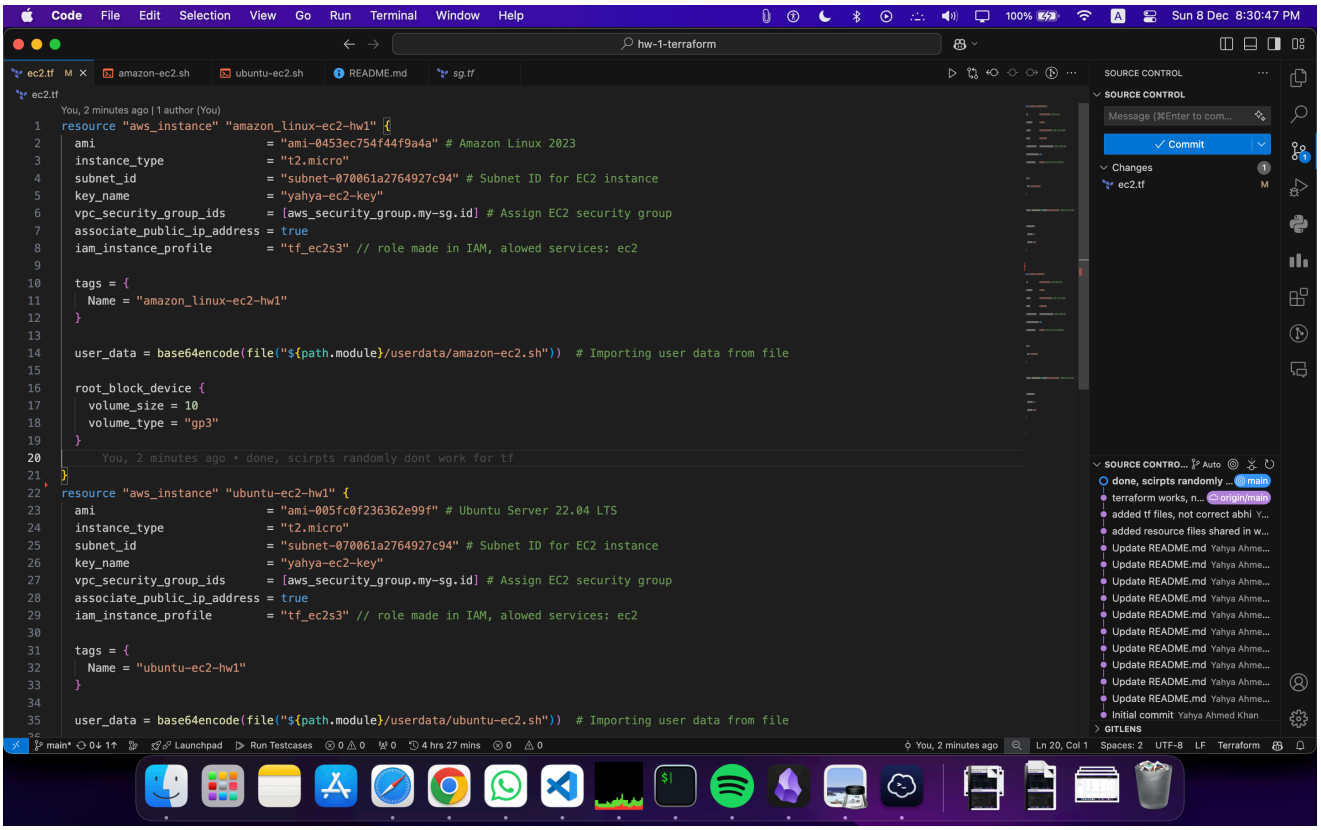


Creating Security Group Rule for Port 3000

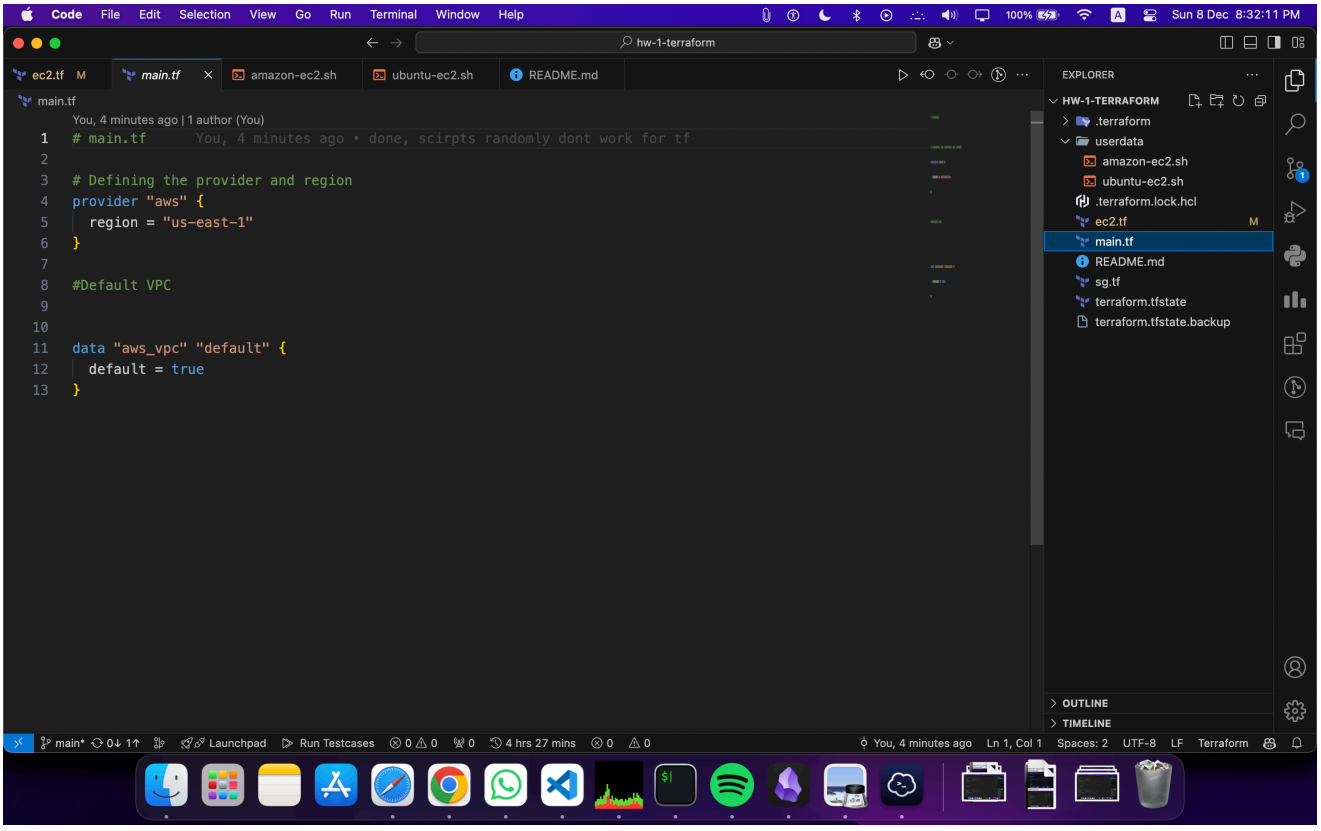


Setting Up EC2 Instances via Terraform

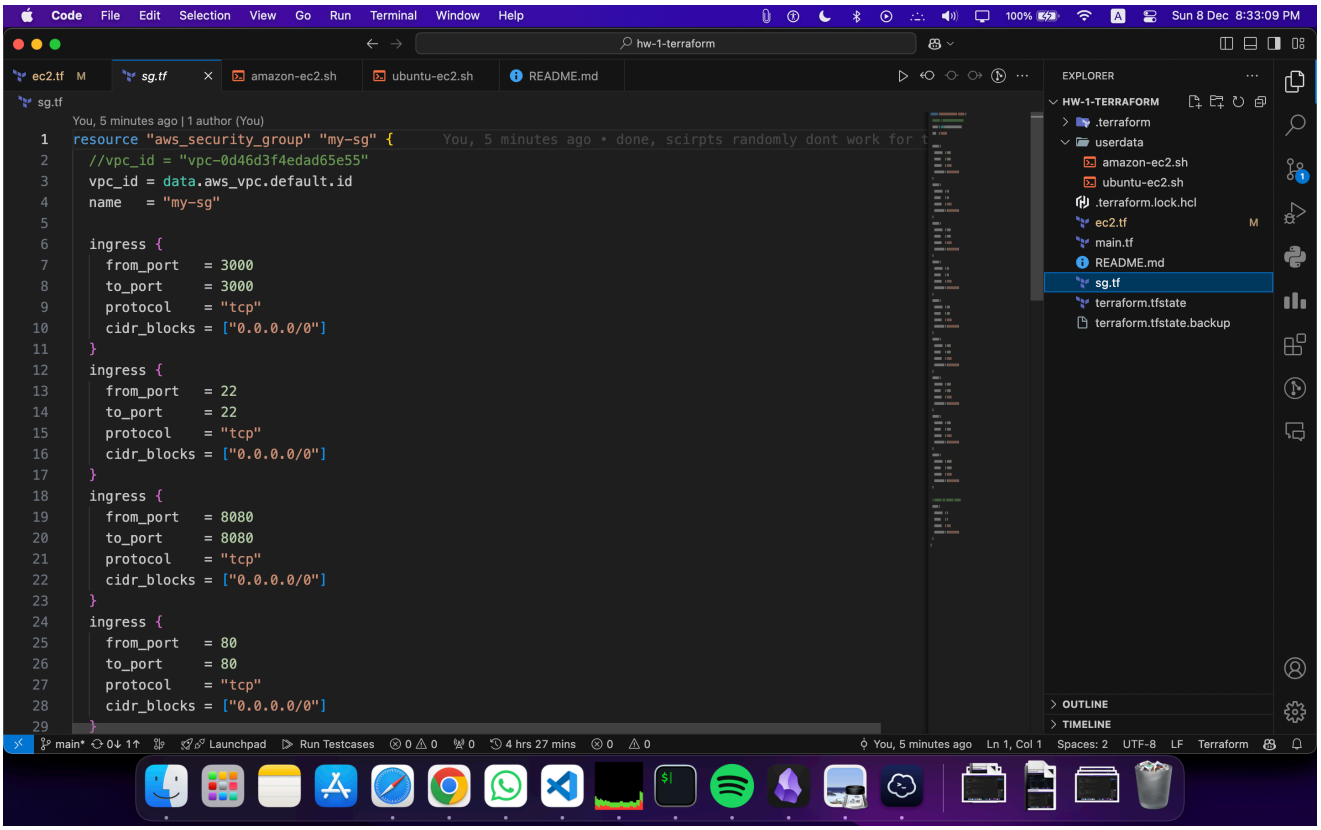
`ec2.tf`


Code File Edit Selection View Go Run Terminal Window Help
Sun 8 Dec 8:30:47 PM
hw-1-terraform
ec2.tf M
amazon-ec2.sh
ubuntu-ec2.sh
README.md
sg.tf
You, 2 minutes ago | 1 author (You)
resource "aws_instance" "amazon_linux-ec2-hw1" {
ami = "ami-0453ec754f44f9a4a" # Amazon Linux 2023
instance_type = "t2.micro"
subnet_id = "subnet-070061a2764927c94" # Subnet ID for EC2 instance
key_name = "yahya-ec2-key"
vpc_security_group_ids = [aws_security_group.my-sg.id] # Assign EC2 security group
associate_public_ip_address = true
iam_instance_profile = "tf_ec2s3" // role made in IAM, alowed services: ec2
tags = {
Name = "amazon_linux-ec2-hw1"
}
user_data = base64encode(file("${path.module}/userdata/amazon-ec2.sh")) # Importing user data from file
root_block_device {
volume_size = 10
volume_type = "gp3"
}
You, 2 minutes ago • done, scirpts randomly dont work for tf
}
resource "aws_instance" "ubuntu-ec2-hw1" {
ami = "ami-005fc0f236362e99f" # Ubuntu Server 22.04 LTS
instance_type = "t2.micro"
subnet_id = "subnet-070061a2764927c94" # Subnet ID for EC2 instance
key_name = "yahya-ec2-key"
vpc_security_group_ids = [aws_security_group.my-sg.id] # Assign EC2 security group
associate_public_ip_address = true
iam_instance_profile = "tf_ec2s3" // role made in IAM, alowed services: ec2
tags = {
Name = "ubuntu-ec2-hw1"
}
user_data = base64encode(file("${path.module}/userdata/ubuntu-ec2.sh")) # Importing user data from file
SOURCE CONTROL
Message (⌘Enter to com...
Commit
Changes
ec2.tf M
done, scirpts randomly ...
terraform works, n...
added tf files, not correct abhi
added resource files shared in w...
Update README.md Yahya Ahme...
Initial commit Yahya Ahmed Khan
GITLENS
main* 0↓ 1↑
Launchpad
Run Testcases
4 hrs 27 mins
You, 2 minutes ago
Ln 20, Col 1
Spaces: 2
UTF-8
LF
Terraform

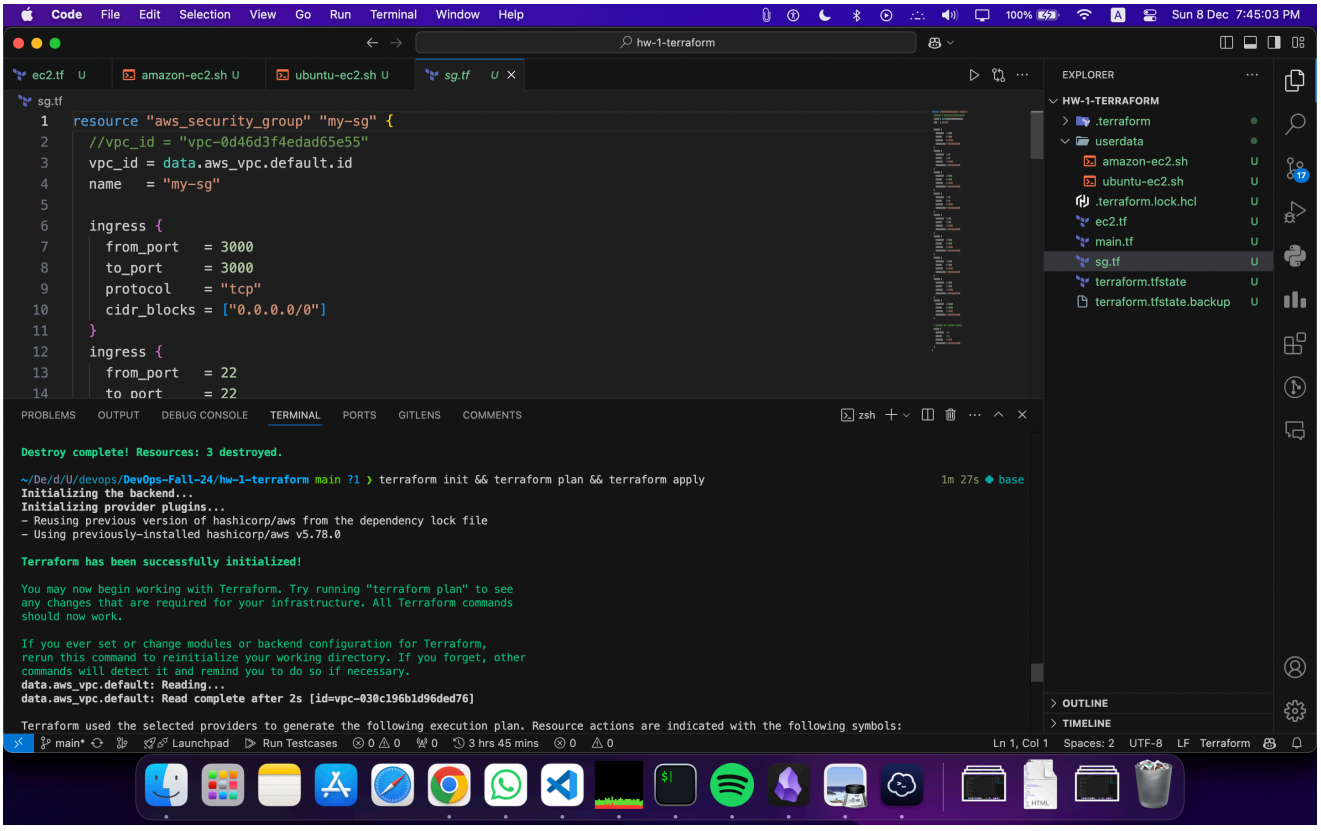
main.tf

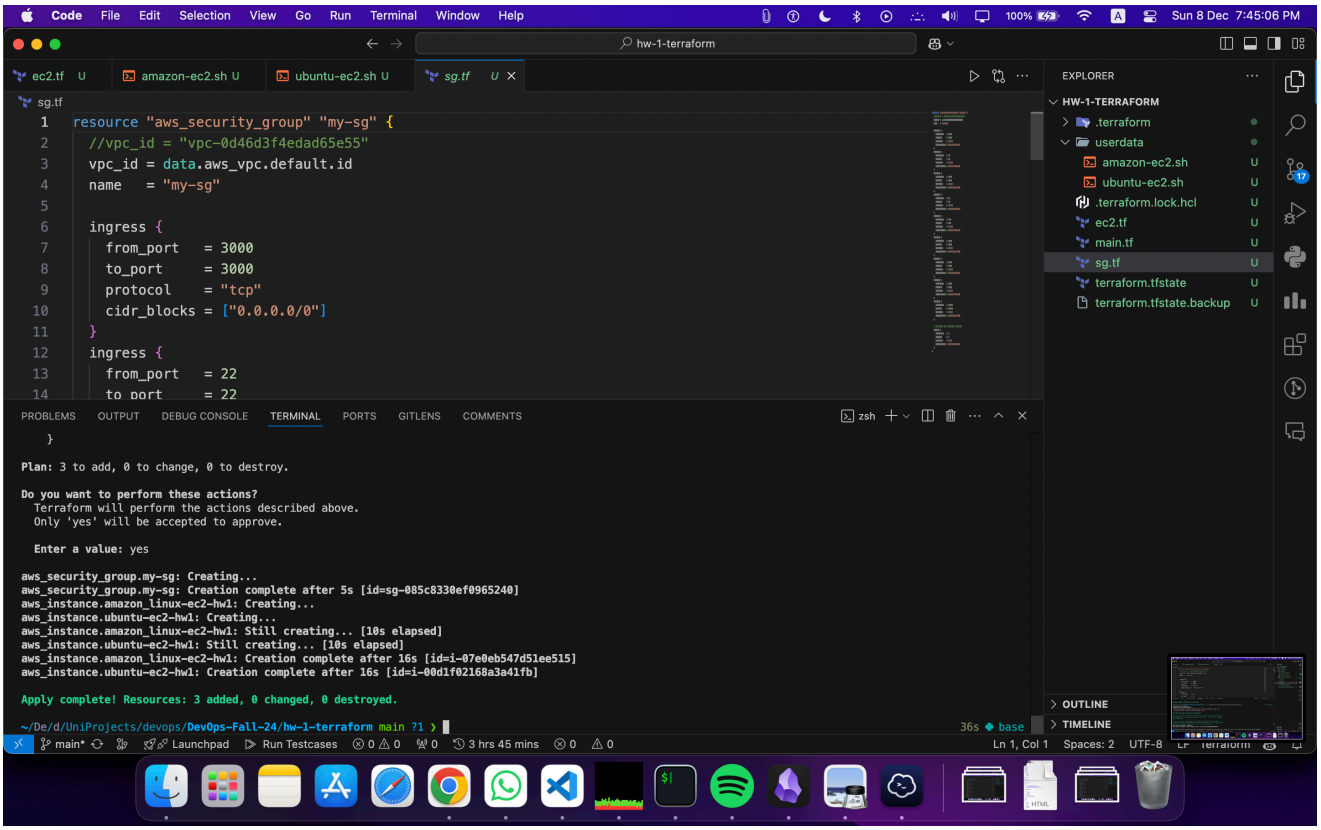

Code File Edit Selection View Go Run Terminal Window Help
Sun 8 Dec 8:32:11 PM
hw-1-terraform
ec2.tf M
main.tf
amazon-ec2.sh
ubuntu-ec2.sh
README.md
EXPLORER
HW-1-TERRAFORM
.terraform
userdata
amazon-ec2.sh
ubuntu-ec2.sh
.terraform.lock.hcl
ec2.tf M
main.tf
README.md
sg.tf
terraform.tfstate
terraform.tfstate.backup
You, 4 minutes ago | 1 author (You)
# main.tf You, 4 minutes ago • done, scirpts randomly dont work for tf
# Defining the provider and region
provider "aws" {
region = "us-east-1"
}
#Default VPC
data "aws_vpc" "default" {
default = true
}
OUTLINE
TIMELINE
main* 0↓ 1↑
Launchpad
Run Testcases
4 hrs 27 mins
You, 4 minutes ago
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Terraform

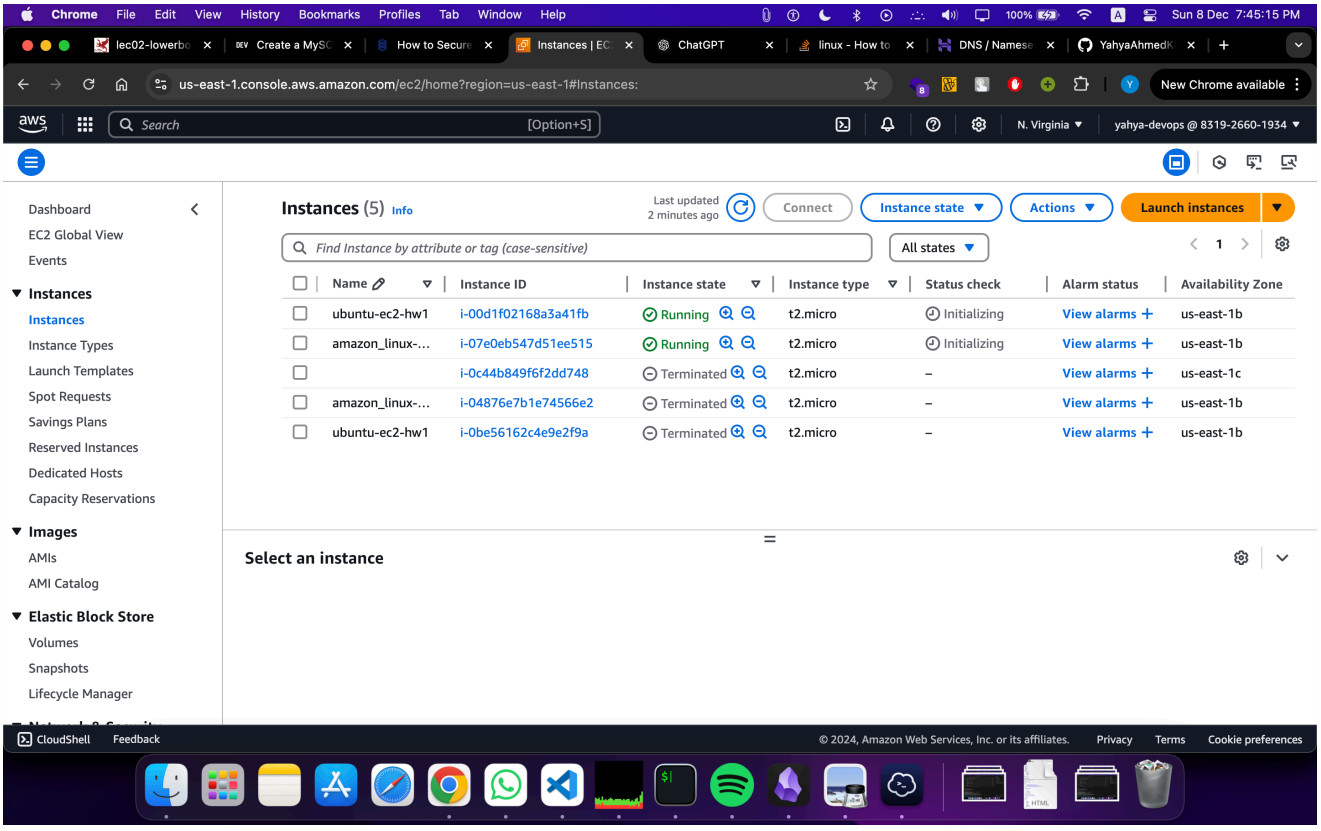
sg.tf

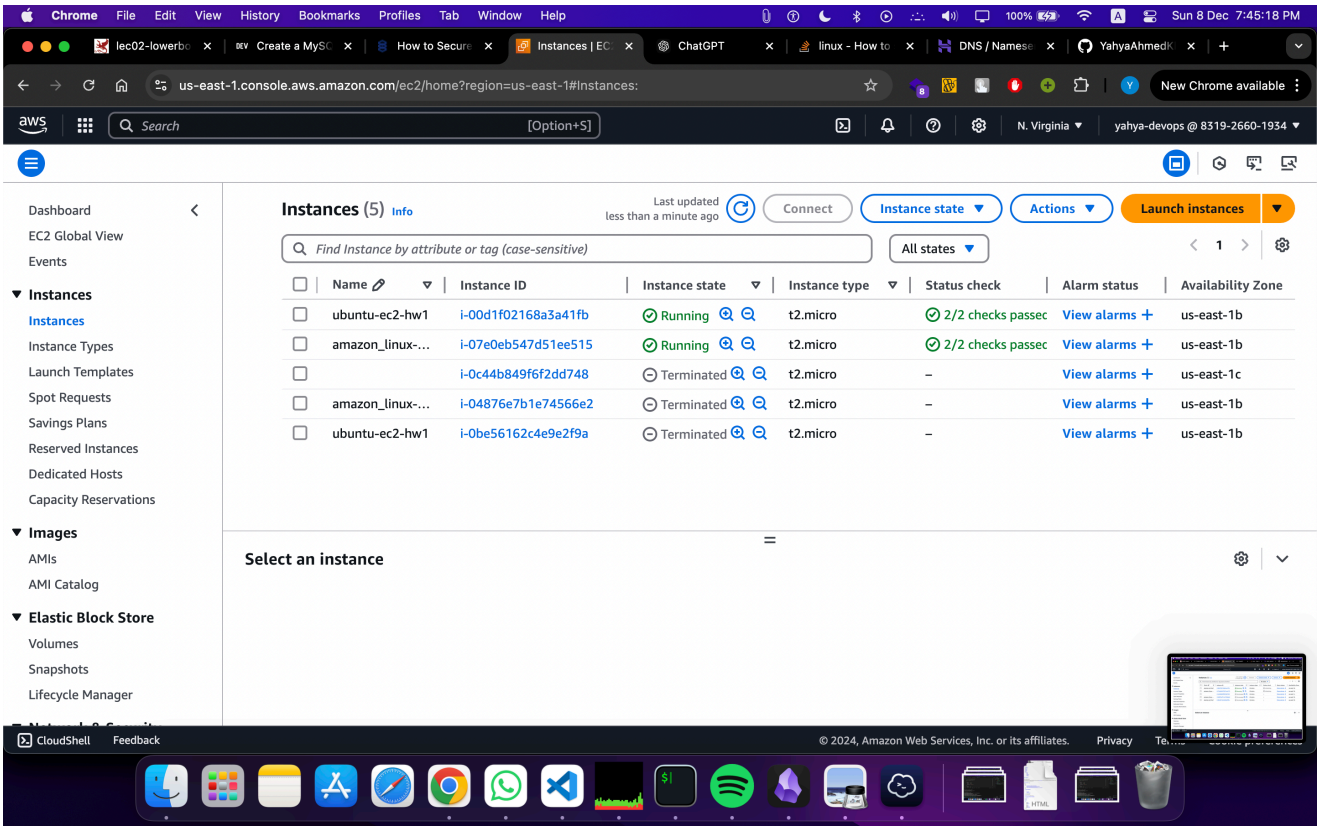


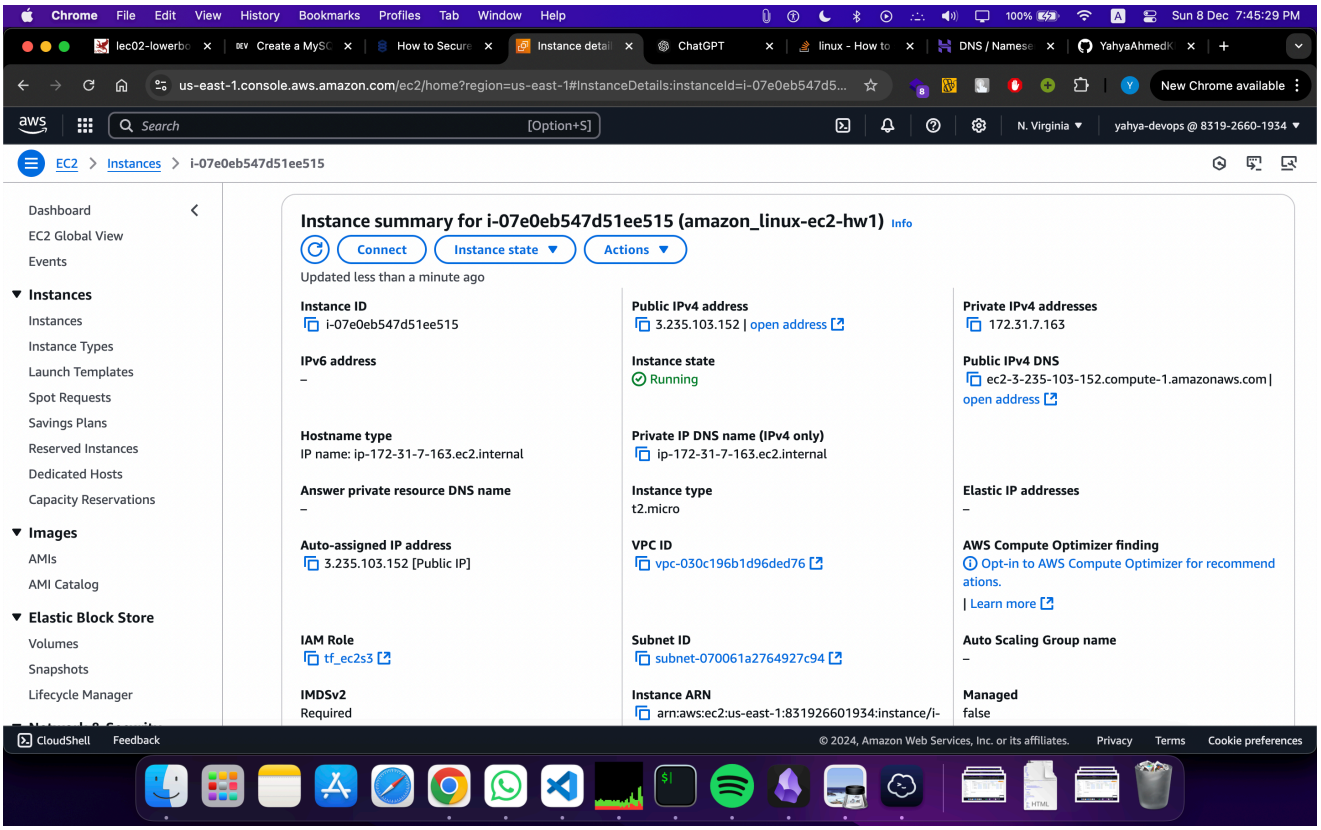
Running terraform

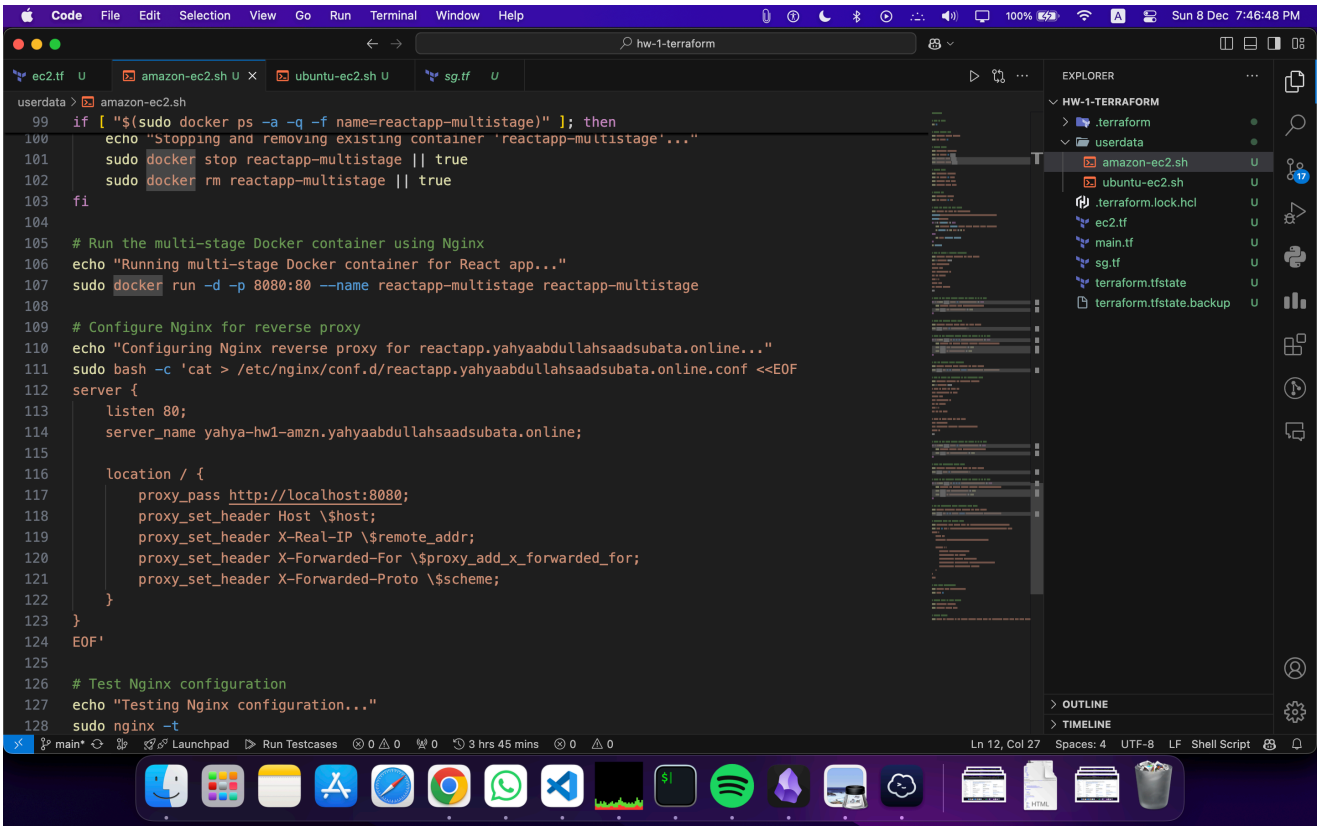



sg.tf
resource "aws_security_group" "my-sg" {
//vpc_id = "vpc-0d46d3f4edad65e55"
vpc_id = data.aws_vpc.default.id
name = "my-sg"
ingress {
from_port = 3000
to_port = 3000
protocol = "tcp"
cidr_blocks = ["0.0.0.0/0"]
}
ingress {
from_port = 22
to_port = 22
Plan: 3 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
aws_security_group.my-sg: Creating...
aws_security_group.my-sg: Creation complete after 5s [id=sg-085c8330ef0965240]
aws_instance.amazon_linux-ec2-hw1: Creating...
aws_instance.ubuntu-ec2-hw1: Creating...
aws_instance.amazon_linux-ec2-hw1: Still creating... [10s elapsed]
aws_instance.ubuntu-ec2-hw1: Still creating... [10s elapsed]
aws_instance.amazon_linux-ec2-hw1: Creation complete after 16s [id=i-07e0eb547d51ee515]
aws_instance.ubuntu-ec2-hw1: Creation complete after 16s [id=i-00d1f02168a3a41fb]
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

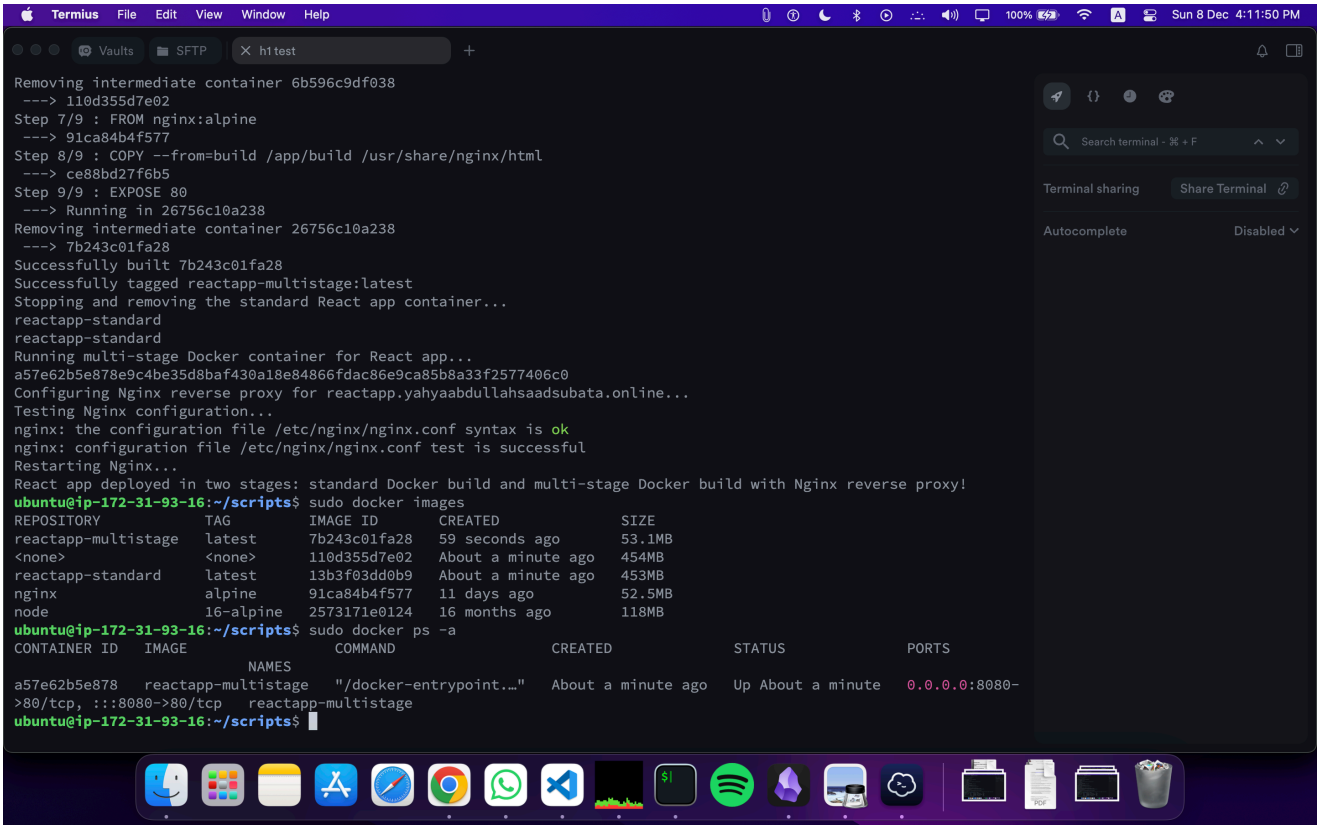

Instances (5) Info
Last updated 2 minutes ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
ubuntu-ec2-hw1 i-00d1f02168a3a41fb Running t2.micro Initializing View alarms us-east-1b
amazon_linux-... i-07e0eb547d51ee515 Running t2.micro Initializing View alarms us-east-1b
i-0c44b849f6f2dd748 Terminated t2.micro View alarms us-east-1c
amazon_linux-... i-04876e7b1e74566e2 Terminated t2.micro View alarms us-east-1b
ubuntu-ec2-hw1 i-0be56162c4e9e2f9a Terminated t2.micro View alarms us-east-1b
Select an instance


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 8 Dec 7:45:18 PM
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
New Chrome available
Search [Option+S]
N. Virginia
yahya-devops @ 8319-2660-1934
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Instances (5) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
ubuntu-ec2-hw1 i-00d1f02168a3a41fb Running t2.micro 2/2 checks passed View alarms us-east-1b
amazon_linux-... i-07e0eb547d51ee515 Running t2.micro 2/2 checks passed View alarms us-east-1b
i-0c44b849f6f2dd748 Terminated t2.micro – View alarms us-east-1c
amazon_linux-... i-04876e7b1e74566e2 Terminated t2.micro – View alarms us-east-1b
ubuntu-ec2-hw1 i-0be56162c4e9e2f9a Terminated t2.micro – View alarms us-east-1b
Select an instance
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy

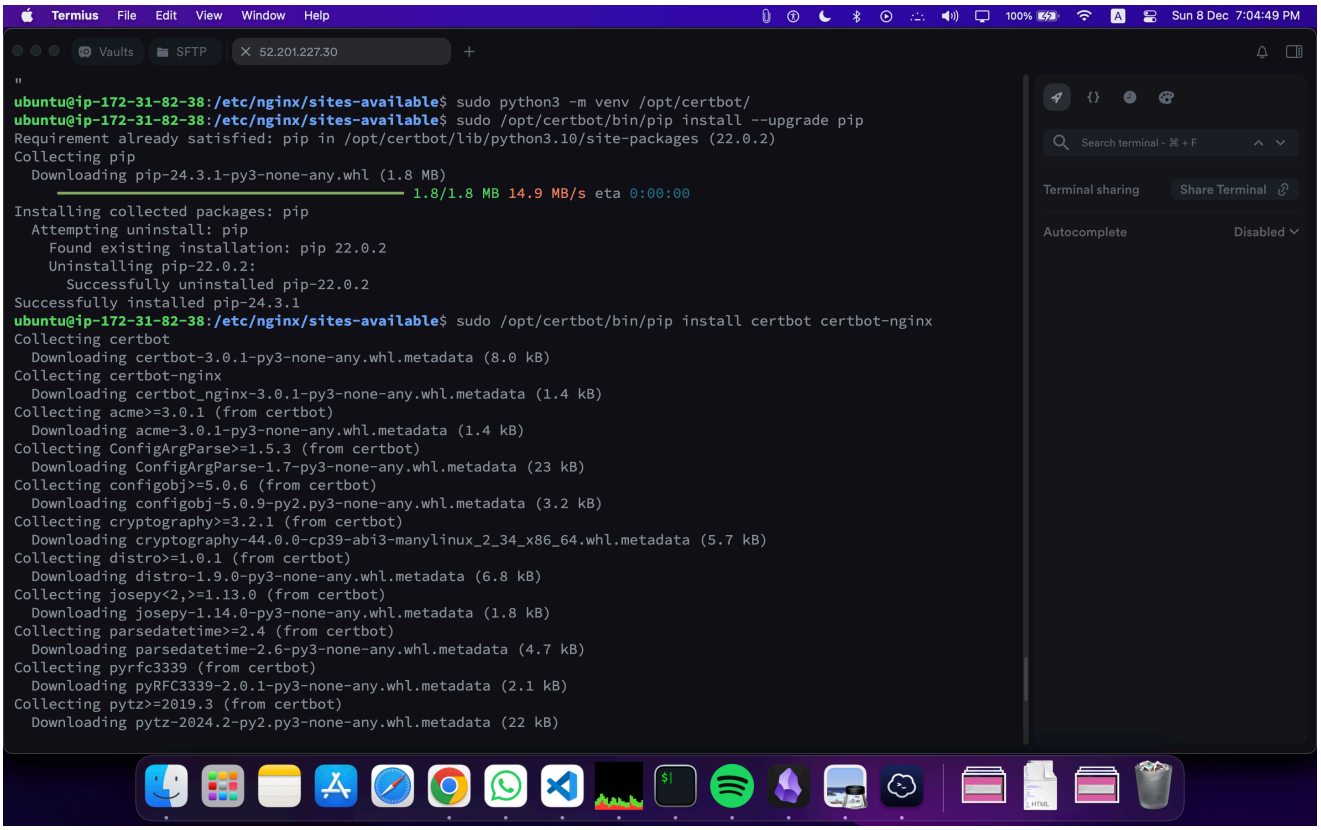

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 8 Dec 7:45:29 PM
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-07e0eb547d5...
New Chrome available
Search [Option+S]
N. Virginia
yahya-devops @ 8319-2660-1934
EC2 > Instances > i-07e0eb547d51ee515
Instance summary for i-07e0eb547d51ee515 (amazon_linux-ec2-hw1) Info
Connect
Instance state
Actions
Updated less than a minute ago
Instance ID
i-07e0eb547d51ee515
IPv6 address
–
Hostname type
IP name: ip-172-31-7-163.ec2.internal
Answer private resource DNS name
–
Auto-assigned IP address
3.235.103.152 [Public IP]
IAM Role
tf_ec2s3
IMDSv2
Required
Public IPv4 address
3.235.103.152 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-7-163.ec2.internal
Instance type
t2.micro
VPC ID
vpc-030c196b1d96ded76
Subnet ID
subnet-070061a2764927c94
Instance ARN
arn:aws:ec2:us-east-1:831926601934:instance/i-
Private IPv4 addresses
172.31.7.163
Public IPv4 DNS
ec2-3-235-103-152.compute-1.amazonaws.com | open address
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
Managed
false
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

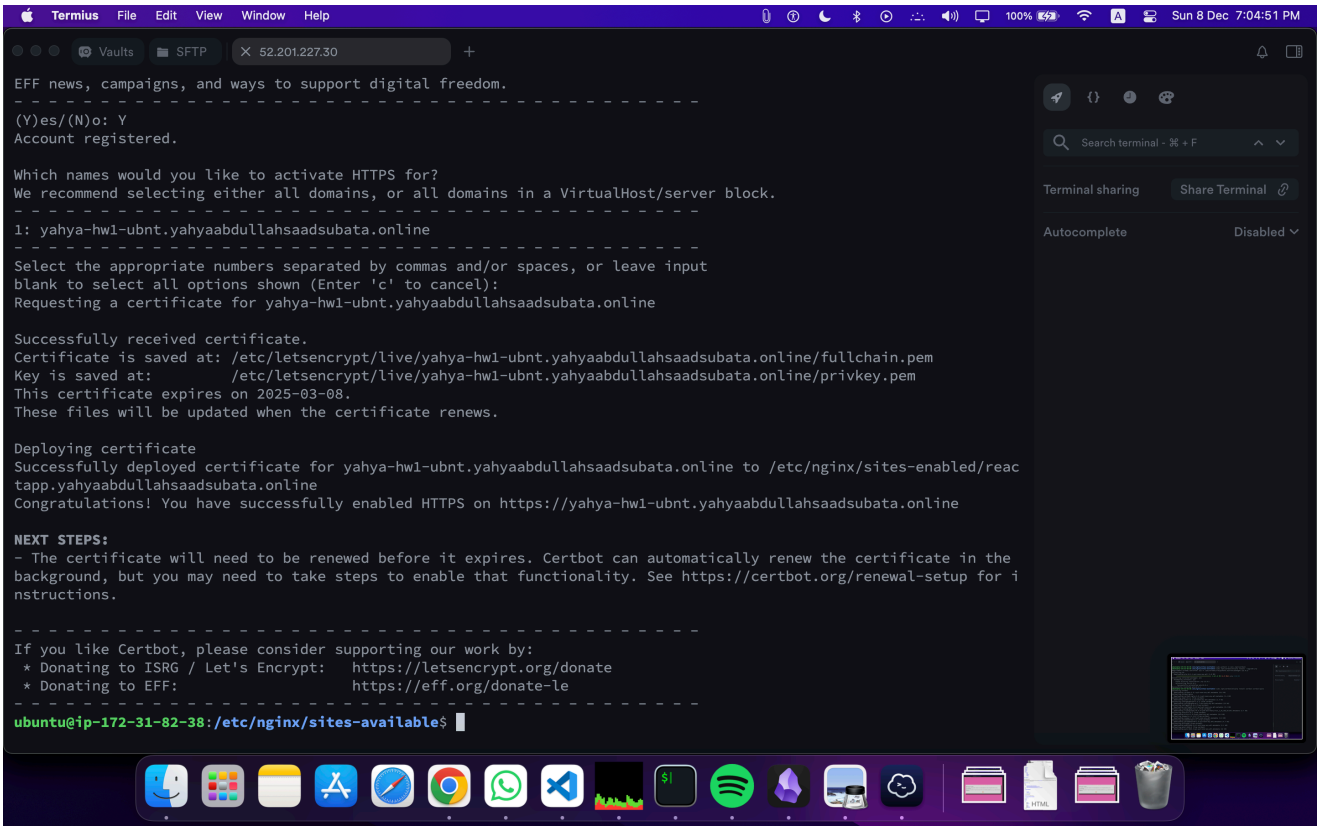


Comparison of Sizes

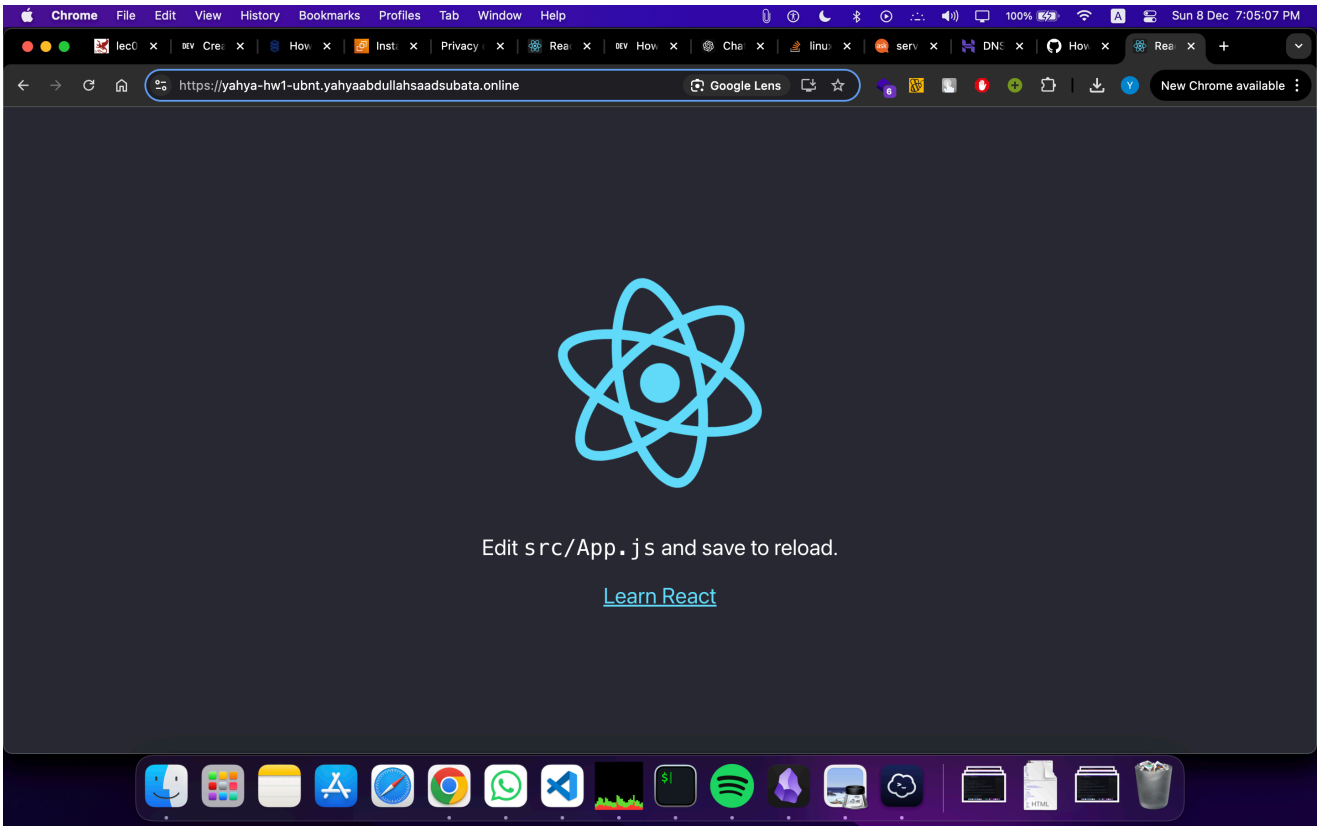


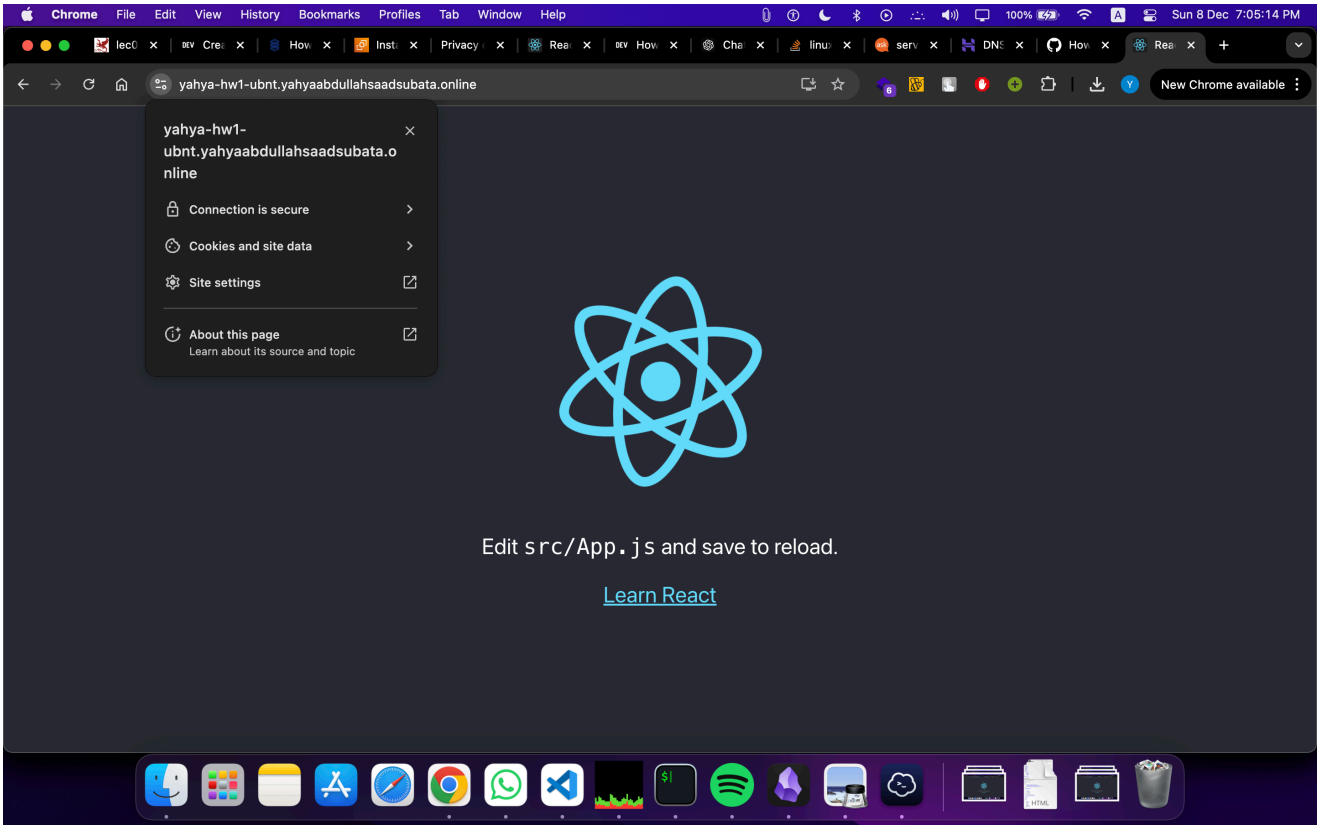
Setup letsencrypt for HTTPS





Showing https version being used





Code files

Server conf for HTTPS

```
server {
        listen       80;
        listen       [::]:80;
```

```
        server_name yahya-hw1-ubnt.yahyaabdullahsaadsubata.online;
        location / {
            proxy_pass http://localhost:8080;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade;
        }
    }
```

Amazon Linux 2023 script.sh

```
#!/bin/bash

# Exit on error
set -e

# Update package list
echo "Updating package list..."
sudo dnf update -y

# Install Docker
echo "Installing Docker..."
sudo dnf install -y docker
sudo systemctl start docker
sudo systemctl enable docker

# Install Nginx
echo "Installing Nginx..."
sudo dnf install -y nginx
sudo systemctl start nginx
sudo systemctl enable nginx

# Install Git
echo "Installing Git..."
sudo dnf install -y git

# Clone the React app from GitHub
echo "Cloning the React app repository..."
REPO_URL="https://github.com/YahyaAhmedKhan/reactapp-devops-hafeez.git"
APP_DIR="reactapp"
if [ -d "$APP_DIR" ]; then
    echo "Directory $APP_DIR already exists. Pulling latest changes..."
    cd $APP_DIR && git pull && cd ..
else
    git clone $REPO_URL $APP_DIR
```

```
fi
cd $APP_DIR

# Step 1: Create a standard Dockerfile
echo "Creating standard Dockerfile for React app..."
cat > Dockerfile <<EOF
FROM node:16-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
EOF

# Check if the image already exists and remove it if it does
if [ "$(sudo docker images -q reactapp-standard)" ]; then
    echo "Removing existing image 'reactapp-standard'..."
    sudo docker rmi reactapp-standard || true
fi

# Build the standard Docker image
echo "Building standard Docker image for React app..."
sudo docker build -t reactapp-standard .

# Check if the container already exists and remove it if it does
if [ "$(sudo docker ps -a -q -f name=reactapp-standard)" ]; then
    echo "Stopping and removing existing container 'reactapp-standard'..."
    sudo docker stop reactapp-standard || true
    sudo docker rm reactapp-standard || true
fi

# Run the standard Docker container
echo "Running standard Docker container for React app..."
sudo docker run -d -p 3000:3000 --name reactapp-standard reactapp-standard

# Step 2: Modify the Dockerfile to use multi-stage build
echo "Modifying Dockerfile for multi-stage build..."
cat > Dockerfile <<EOF
# Stage 1: Build the React app
FROM node:16-alpine AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# Stage 2: Serve React app with Nginx
FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
```

```
EXPOSE 80
EOF

# Check if the image already exists and remove it if it does
if [ "$(sudo docker images -q reactapp-multistage)" ]; then
    echo "Removing existing image 'reactapp-multistage'..."
    sudo docker rmi reactapp-multistage || true
fi

# Build the multi-stage Docker image
echo "Building multi-stage Docker image for React app..."
sudo docker build -t reactapp-multistage .

# Check if the container already exists and remove it if it does
if [ "$(sudo docker ps -a -q -f name=reactapp-multistage)" ]; then
    echo "Stopping and removing existing container 'reactapp-
multistage'..."
    sudo docker stop reactapp-multistage || true
    sudo docker rm reactapp-multistage || true
fi

# Run the multi-stage Docker container using Nginx
echo "Running multi-stage Docker container for React app..."
sudo docker run -d -p 8080:80 --name reactapp-multistage reactapp-
multistage

# Configure Nginx for reverse proxy
echo "Configuring Nginx reverse proxy for
reactapp.yahyaabdullahsaadsubata.online..."
sudo bash -c 'cat >
/etc/nginx/conf.d/reactapp.yahyaabdullahsaadsubata.online.conf <<EOF
server {
    listen 80;
    server_name yahya-hw1-amzn.yahyaabdullahsaadsubata.online;

    location / {
        proxy_pass http://localhost: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;
        proxy_set_header X-Forwarded-Proto \$scheme;
    }
}
EOF'

# Test Nginx configuration
echo "Testing Nginx configuration..."
sudo nginx -t

# Restart Nginx to apply changes
```

```
echo "Restarting Nginx..."
sudo systemctl restart nginx

# Success message
echo "React app deployed in two stages: standard Docker build and multi-stage Docker build with Nginx reverse proxy!"
```

Ubuntu 22.04 script.sh

```
#!/bin/bash

# Exit on error
set -e

sudo sed -i 's/#$nrconf{restart} = '"'"'i'"'"';/$nrconf{restart} = '"'"'a'"'"';/g' /etc/needrestart/needrestart.conf

# Update package list
echo "Updating package list..."
sudo apt update -y && sudo apt upgrade -y

# Install Docker
echo "Installing Docker..."
sudo apt install -y docker.io
sudo systemctl start docker
sudo systemctl enable docker

# Install Nginx
echo "Installing Nginx..."
sudo apt install -y nginx
sudo systemctl start nginx
sudo systemctl enable nginx

# Install Git
echo "Installing Git..."
sudo apt install -y git

# Clone the React app from GitHub
echo "Cloning the React app repository..."
REPO_URL="https://github.com/YahyaAhmedKhan/reactapp-devops-hafeez.git"
APP_DIR="reactapp"
if [ -d "$APP_DIR" ]; then
    echo "Directory $APP_DIR already exists. Pulling latest changes..."
    cd $APP_DIR && git pull && cd ..
else
    git clone $REPO_URL $APP_DIR
fi
cd $APP_DIR
```

```
# Step 1: Create a standard Dockerfile
echo "Creating standard Dockerfile for React app..."
cat > Dockerfile <<EOF
FROM node:16-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
EOF

# Check if the image already exists and remove it if it does
if [ "$(sudo docker images -q reactapp-standard)" ]; then
    echo "Removing existing image 'reactapp-standard'..."
    sudo docker rmi reactapp-standard || true
fi

# Build the standard Docker image
echo "Building standard Docker image for React app..."
sudo docker build -t reactapp-standard .

# Check if the container already exists and remove it if it does
if [ "$(sudo docker ps -a -q -f name=reactapp-standard)" ]; then
    echo "Stopping and removing existing container 'reactapp-standard'..."
    sudo docker stop reactapp-standard || true
    sudo docker rm reactapp-standard || true
fi

# Run the standard Docker container
echo "Running standard Docker container for React app..."
sudo docker run -d -p 3000:3000 --name reactapp-standard reactapp-standard

# Step 2: Modify the Dockerfile to use multi-stage build
echo "Modifying Dockerfile for multi-stage build..."
cat > Dockerfile <<EOF
# Stage 1: Build the React app
FROM node:16-alpine AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

# Stage 2: Serve React app with Nginx
FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
EXPOSE 80
EOF
```

```
# Check if the image already exists and remove it if it does
if [ "$(sudo docker images -q reactapp-multistage)" ]; then
    echo "Removing existing image 'reactapp-multistage'..."
    sudo docker rmi reactapp-multistage || true
fi

# Build the multi-stage Docker image
echo "Building multi-stage Docker image for React app..."
sudo docker build -t reactapp-multistage .

# Check if the container already exists and remove it if it does
if [ "$(sudo docker ps -a -q -f name=reactapp-multistage)" ]; then
    echo "Stopping and removing existing container 'reactapp-
multistage'..."
    sudo docker stop reactapp-multistage || true
    sudo docker rm reactapp-multistage || true
fi


# Run the multi-stage Docker container using Nginx
echo "Running multi-stage Docker container for React app..."
sudo docker run -d -p 8080:80 --name reactapp-multistage reactapp-
multistage

# Remove the default configuration if it exists
echo "Removing default configuration if it exists..."
if [ -f /etc/nginx/sites-available/default ]; then
    sudo rm /etc/nginx/sites-available/default
fi
if [ -L /etc/nginx/sites-enabled/default ]; then
    sudo rm /etc/nginx/sites-enabled/default
fi

# Configure Nginx reverse proxy
echo "Configuring Nginx reverse proxy for
reactapp.yahyaabdullahsaadsubata.online..."
sudo bash -c 'cat > /etc/nginx/sites-
available/reactapp.yahyaabdullahsaadsubata.online <<EOF
server {
    listen 80;
    server_name yahya-hw1-ubnt.yahyaabdullahsaadsubata.online;

    location / {
        proxy_pass http://localhost: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;
        proxy_set_header X-Forwarded-Proto \$scheme;
```

```
    }
}
EOF'

# Remove existing symlink if it exists
echo "Checking if symlink already exists in sites-enabled..."
if [ -L /etc/nginx/sites-enabled/reactapp.yahyaabdullahsaadsubata.online
]; then
    echo "Symlink already exists, removing it..."
    sudo rm /etc/nginx/sites-
enabled/reactapp.yahyaabdullahsaadsubata.online
fi

# Create a symbolic link in sites-enabled to enable the site
echo "Enabling the site..."
sudo ln -s /etc/nginx/sites-
available/reactapp.yahyaabdullahsaadsubata.online /etc/nginx/sites-
enabled/

# Test Nginx configuration
echo "Testing Nginx configuration..."
sudo nginx -t

# Restart Nginx to apply changes
echo "Restarting Nginx..."
sudo systemctl restart nginx

# Success message
echo "React app deployed in two stages: standard Docker build and multi-
stage Docker build with Nginx reverse proxy!"
```