Script file create Network Infrastructure

```
#!/bin/bash

source ./variables.sh

# Create VPC
echo "Creating VPC..."
VPC_ID=$(aws ec2 create-vpc --cidr-block $VPC_CIDR --region $REGION --query "Vpc.VpcId" --
output text)
aws ec2 create-tags --resources $VPC_ID --tags Key=Name,Value=$VPC_NAME
echo "VPC Created: $VPC_ID"

# Create Subnets
echo "Creating Public Subnet1..."
PUBLIC_SUBNET_ID_1=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block
$PUBLIC_SUBNET1_CIDR --availability-zone ${REGION}a  --region $REGION --query
"Subnet.SubnetId" --output text)
aws ec2 create-tags --resources $PUBLIC_SUBNET_ID_1 --tags
Key=Name,Value=$PUBLIC_SUBNET1
echo "Public Subnet Created: $PUBLIC_SUBNET_ID_1"

echo "Creating Public Subnet2..."
PUBLIC_SUBNET_ID_2=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block
$PUBLIC_SUBNET2_CIDR --availability-zone ${REGION}b  --region $REGION --query
"Subnet.SubnetId" --output text)
aws ec2 create-tags --resources $PUBLIC_SUBNET_ID_2 --tags
Key=Name,Value=$PUBLIC_SUBNET2
echo "Public Subnet Created: $PUBLIC_SUBNET_ID_2"


echo "Creating Private Subnet1..."
PRIVATE_SUBNET_ID_1=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block
$PRIVATE_SUBNET1_CIDR --availability-zone ${REGION}b  --region $REGION --query
"Subnet.SubnetId" --output text)
aws ec2 create-tags --resources $PRIVATE_SUBNET_ID_1 --tags
Key=Name,Value=$PRIVATE_SUBNET1
echo "Private Subnet Created: $PRIVATE_SUBNET_ID_1"

echo "Creating Private Subnet2..."
PRIVATE_SUBNET_ID_2=$(aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block
$PRIVATE_SUBNET2_CIDR --availability-zone ${REGION}c  --region $REGION --query
"Subnet.SubnetId" --output text)
aws ec2 create-tags --resources $PRIVATE_SUBNET_ID_2 --tags
Key=Name,Value=$PRIVATE_SUBNET2
echo "Private Subnet Created: $PRIVATE_SUBNET_ID_2"


# Create Internet Gateway
echo "Creating Internet Gateway..."
```

```
IGW_ID=$(aws ec2 create-internet-gateway --region $REGION --query
"InternetGateway.InternetGatewayId" --output text)
aws ec2 create-tags --resources $IGW_ID --tags Key=Name,Value=$IGW_NAME
aws ec2 attach-internet-gateway --vpc-id $VPC_ID --internet-gateway-id $IGW_ID
echo "Internet Gateway Created and Attached: $IGW_ID"


# Create Public Route Table and Associate with Public Subnet
echo "Creating Public Route Table..."
ROUTE_TABLE_ID=$(aws ec2 create-route-table --vpc-id $VPC_ID --region $REGION --query
"RouteTable.RouteTableId" --output text)
aws ec2 create-tags --resources $ROUTE_TABLE_ID --tags
Key=Name,Value=$PUBLIC_ROUTE_TABLE
aws ec2 create-route --route-table-id $ROUTE_TABLE_ID --destination-cidr-block 0.0.0.0/0 --gateway-
id $IGW_ID
aws ec2 associate-route-table --route-table-id $ROUTE_TABLE_ID --subnet-id
$PUBLIC_SUBNET_ID_1
aws ec2 associate-route-table --route-table-id $ROUTE_TABLE_ID --subnet-id
$PUBLIC_SUBNET_ID_2
echo "Public Route Table Created and Associated: $ROUTE_TABLE_ID"


# Allocate Elastic IP and Create NAT Gateway
echo "Allocating Elastic IP for NAT Gateway..."
EIP_ALLOC_ID=$(aws ec2 allocate-address --domain vpc --region $REGION --query "AllocationId" --
output text)

echo "Creating NAT gateway......."
NAT_GATEWAY_ID=$(aws ec2 create-nat-gateway --subnet-id $PUBLIC_SUBNET_ID_2 --
allocation-id $EIP_ALLOC_ID --region $REGION --query "NatGateway.NatGatewayId" --output text)
echo "NAT Gateway Created: $NAT_GATEWAY_ID"

# Wait for NAT Gateway to be available
echo "Waiting for NAT Gateway to become available..."
aws ec2 wait nat-gateway-available --nat-gateway-ids $NAT_GATEWAY_ID
echo "NAT Gateway is now available."


# Create Private Route Table and Associate with Private Subnet
echo "Creating Private Route Table..."
PRIVATE_ROUTE_TABLE_ID=$(aws ec2 create-route-table --vpc-id $VPC_ID --region $REGION --
query "RouteTable.RouteTableId" --output text)
aws ec2 create-tags --resources $PRIVATE_ROUTE_TABLE_ID --tags
Key=Name,Value=$PRIVATE_ROUTE_TABLE
aws ec2 create-route --route-table-id $PRIVATE_ROUTE_TABLE_ID --destination-cidr-block 0.0.0.0/0
--nat-gateway-id $NAT_GATEWAY_ID
aws ec2 associate-route-table --route-table-id $PRIVATE_ROUTE_TABLE_ID --subnet-id
$PRIVATE_SUBNET_ID_1

aws ec2 associate-route-table --route-table-id $PRIVATE_ROUTE_TABLE_ID --subnet-id
$PRIVATE_SUBNET_ID_2
```

```
echo "Private Route Table Created and Associated: $PRIVATE_ROUTE_TABLE_ID"


# Create Security Group
echo "Creating Security Group..."
SG_ID=$(aws ec2 create-security-group --group-name $SECURITY_GROUP_NAME --description "My
Security Group" --vpc-id $VPC_ID --region $REGION --query "GroupId" --output text)
aws ec2 authorize-security-group-ingress --group-id $SG_ID --protocol tcp --port 22 --cidr 0.0.0.0/0  #
SSH
aws ec2 authorize-security-group-ingress --group-id $SG_ID --protocol tcp --port 80 --cidr 0.0.0.0/0  #
HTTP
echo "Security Group Created: $SG_ID"

# Launch EC2 Instance in Public Subnet
echo "Launching EC2 Instance..."
INSTANCE_ID_1=$(aws ec2 run-instances --image-id $AMI_ID --count 1 --instance-type
$INSTANCE_TYPE --key-name $KEY_NAME --security-group-ids $SG_ID --subnet-id
$PUBLIC_SUBNET_ID_1 --region $REGION --query "Instances[0].InstanceId" --output text)
aws ec2 create-tags --resources $INSTANCE_ID_1 --tags Key=Name,Value=$INSTANCE_NAME1
echo "EC2 Instance Launched: $INSTANCE_ID_1"


# Launch EC2 Instance in Private Subnet
echo "Launching EC2 Instance..."
INSTANCE_ID_2=$(aws ec2 run-instances --image-id $AMI_ID --count 1 --instance-type
$INSTANCE_TYPE --key-name $KEY_NAME --security-group-ids $SG_ID --subnet-id
$PRIVATE_SUBNET_ID_1 --region $REGION --query "Instances[0].InstanceId" --output text)
aws ec2 create-tags --resources $INSTANCE_ID_2 --tags Key=Name,Value=$INSTANCE_NAME2
echo "EC2 Instance Launched: $INSTANCE_ID_2"




echo "Creating Target Group..."
TARGET_GROUP_ARN=$(aws elbv2 create-target-group --name $TARGET_GROUP_NAME --
protocol HTTP --port 80 --vpc-id $VPC_ID --query 'TargetGroups[0].TargetGroupArn' --output text)
echo "Target Group Created: $TARGET_GROUP_ARN"



echo "Creating Load Balancer..."
LOAD_BALANCER_ARN=$(aws elbv2 create-load-balancer --name $LOAD_BALANCER_NAME --
subnets $PUBLIC_SUBNET_ID_1 $PUBLIC_SUBNET_ID_2 --security-groups $SG_ID --query
'LoadBalancers[0].LoadBalancerArn' --output text)

aws elbv2 add-tags --resource-arns $LOAD_BALANCER_ARN --tags
Key=Name,Value=$LOAD_BALANCER_NAME
echo "Load Balancer Created: $LOAD_BALANCER_ARN"
```

```
echo "Attaching Target Group to Load Balancer..."
aws elbv2 create-listener --load-balancer-arn $LOAD_BALANCER_ARN --protocol HTTP --port 80 --
default-actions Type=forward,TargetGroupArn=$TARGET_GROUP_ARN
echo "Target Group Attached to Load Balancer"

echo "Creating Launch Template..."
LAUNCH_TEMPLATE_ID=$(aws ec2 create-launch-template --launch-template-name
$LAUNCH_TEMPLATE_NAME --version-description "v1" --launch-template-data "{
  \"ImageId\": \"$AMI_ID\",
  \"InstanceType\": \"$INSTANCE_TYPE\",
  \"KeyName\": \"$KEY_NAME\",
  \"SecurityGroupIds\": [\"$SG_ID\"]
}" --query 'LaunchTemplate.LaunchTemplateId' --output text)
echo "Launch Template Created: $LAUNCH_TEMPLATE_ID"



echo "Creating Auto Scaling Group..."
aws autoscaling create-auto-scaling-group --auto-scaling-group-name
$AUTO_SCALING_GROUP_NAME --launch-template
"LaunchTemplateId=$LAUNCH_TEMPLATE_ID,Version=1" --min-size 1 --max-size 4 --desired-
capacity 2 --vpc-zone-identifier "$PUBLIC_SUBNET_ID_1,$PUBLIC_SUBNET_ID_2" --target-group-
arns $TARGET_GROUP_ARN
echo "Auto Scaling Group Created and Attached to Target Group:
$AUTO_SCALING_GROUP_NAME"


echo "Infrastructure Creation Complete!"

echo "VPC ID: $VPC_ID"

echo "Public Subnet ID 1: $PUBLIC_SUBNET_ID_1"

echo "Public Subnet ID 2: $PUBLIC_SUBNET_ID_2"

echo "Private Subnet ID 1: $PRIVATE_SUBNET_ID_1"

echo "Private Subnet ID 2: $PRIVATE_SUBNET_ID_2"

echo "Internet Gateway ID: $IGW_ID"

echo "NAT Gateway ID: $NAT_GATEWAY_ID"

echo "Security Group ID: $SG_ID"

echo "Public EC2 Instance ID: $INSTANCE_ID_1"

echo "Private EC2 Instance ID: $INSTANCE_ID_2"
```