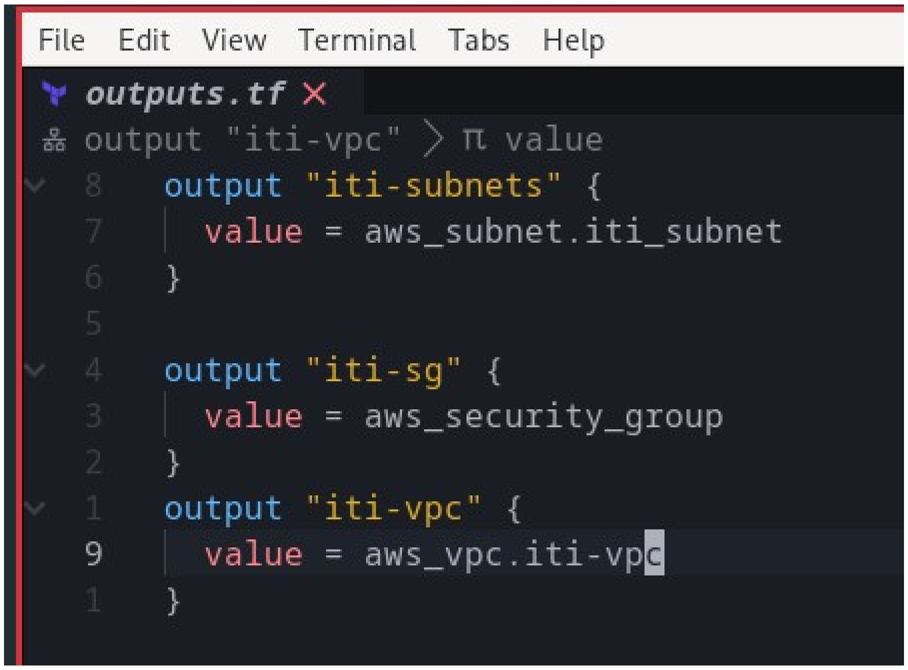
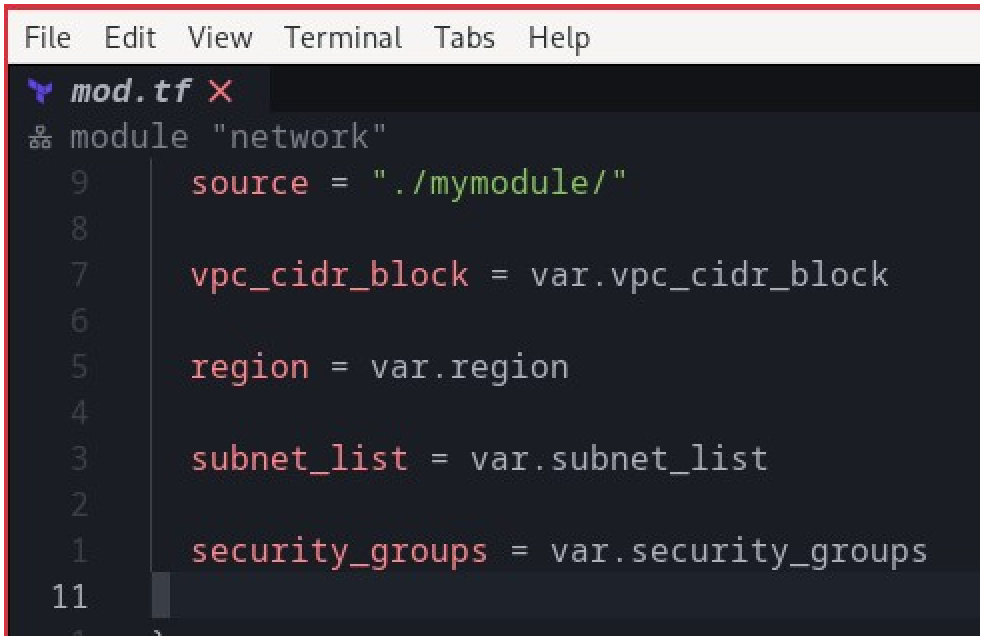
1-separate network resources into network module


File Edit View Terminal Tabs Help

outputs.tf

output "iti-vpc" > π value

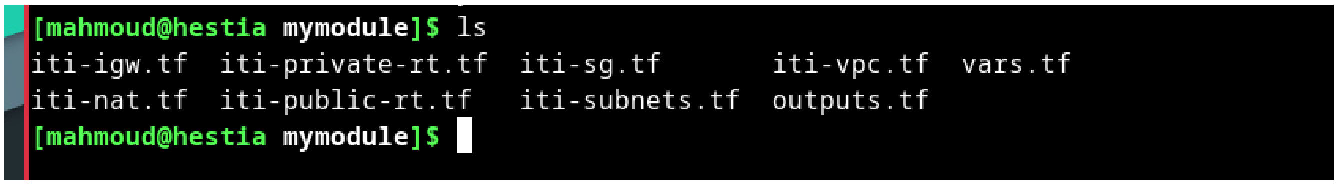
```
output "iti-subnets" {
  value = aws_subnet.iti_subnet
}

output "iti-sg" {
  value = aws_security_group
}
output "iti-vpc" {
  value = aws_vpc.iti-vpc
}
```


File Edit View Terminal Tabs Help

mod.tf

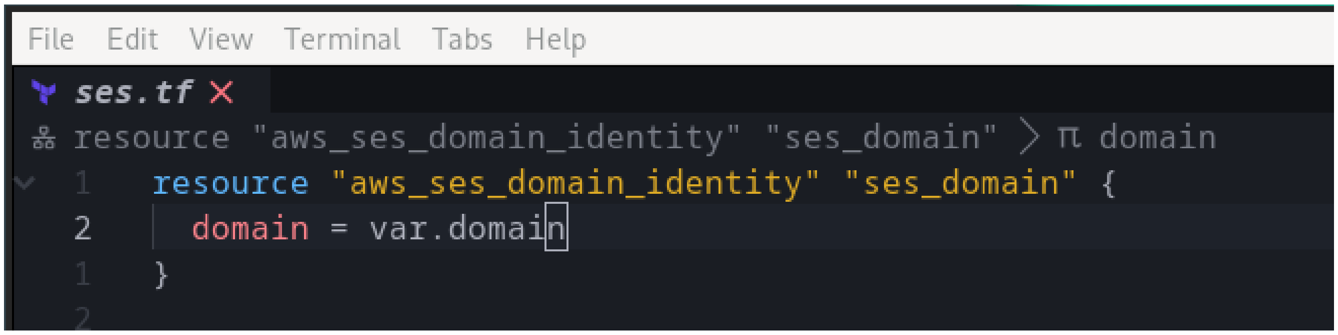
module "network"

```
  source = "./mymodule/"

  vpc_cidr_block = var.vpc_cidr_block

  region = var.region

  subnet_list = var.subnet_list

  security_groups = var.security_groups

```



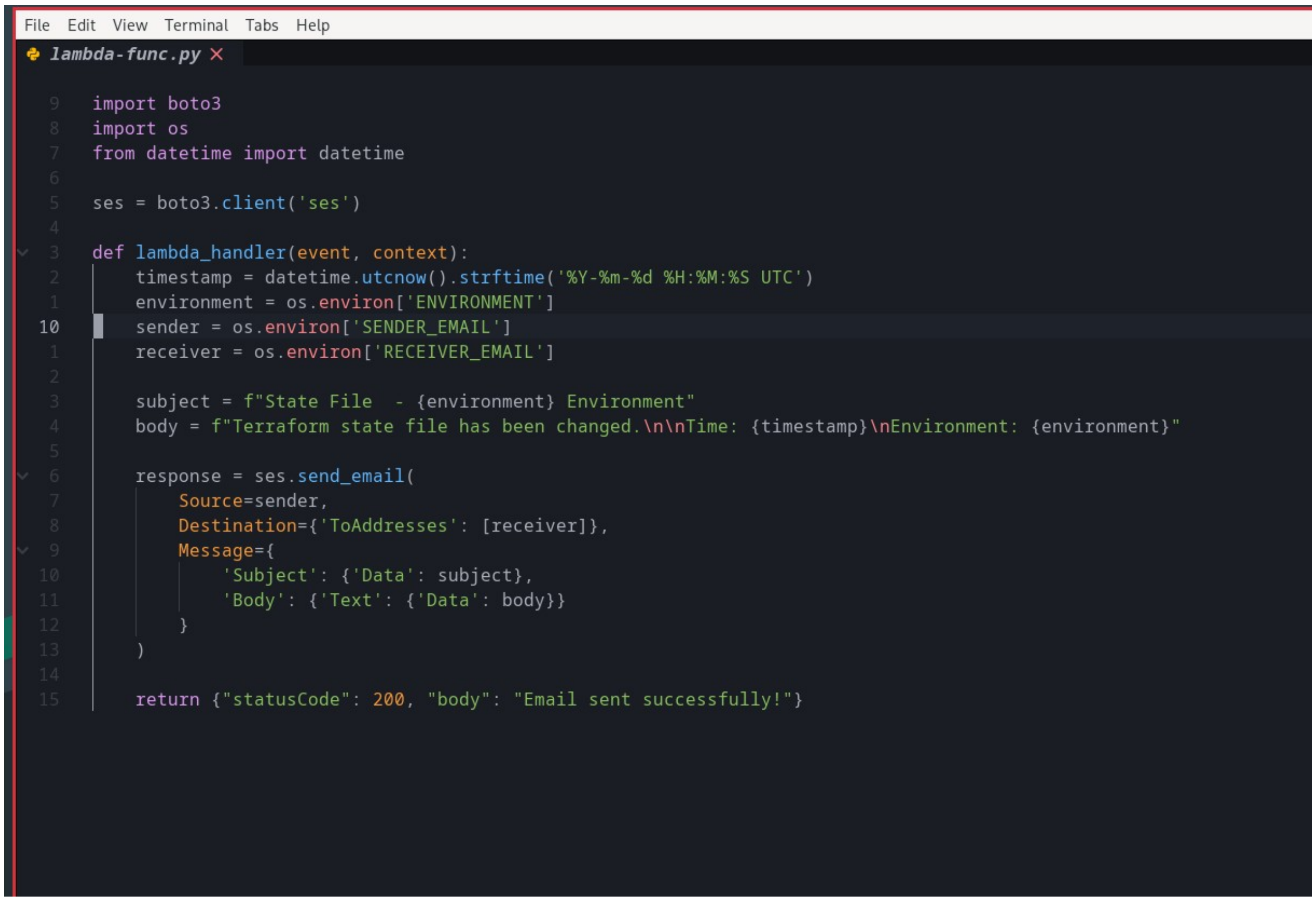
```
[mahmoud@hestia mymodule]$ ls
iti-igw.tf  iti-private-rt.tf  iti-sg.tf       iti-vpc.tf  vars.tf
iti-nat.tf  iti-public-rt.tf   iti-subnets.tf  outputs.tf
[mahmoud@hestia mymodule]$ 
```

2-verify your email in ses service


File Edit View Terminal Tabs Help
ses.tf
resource "aws_ses_domain_identity" "ses_domain" > π domain

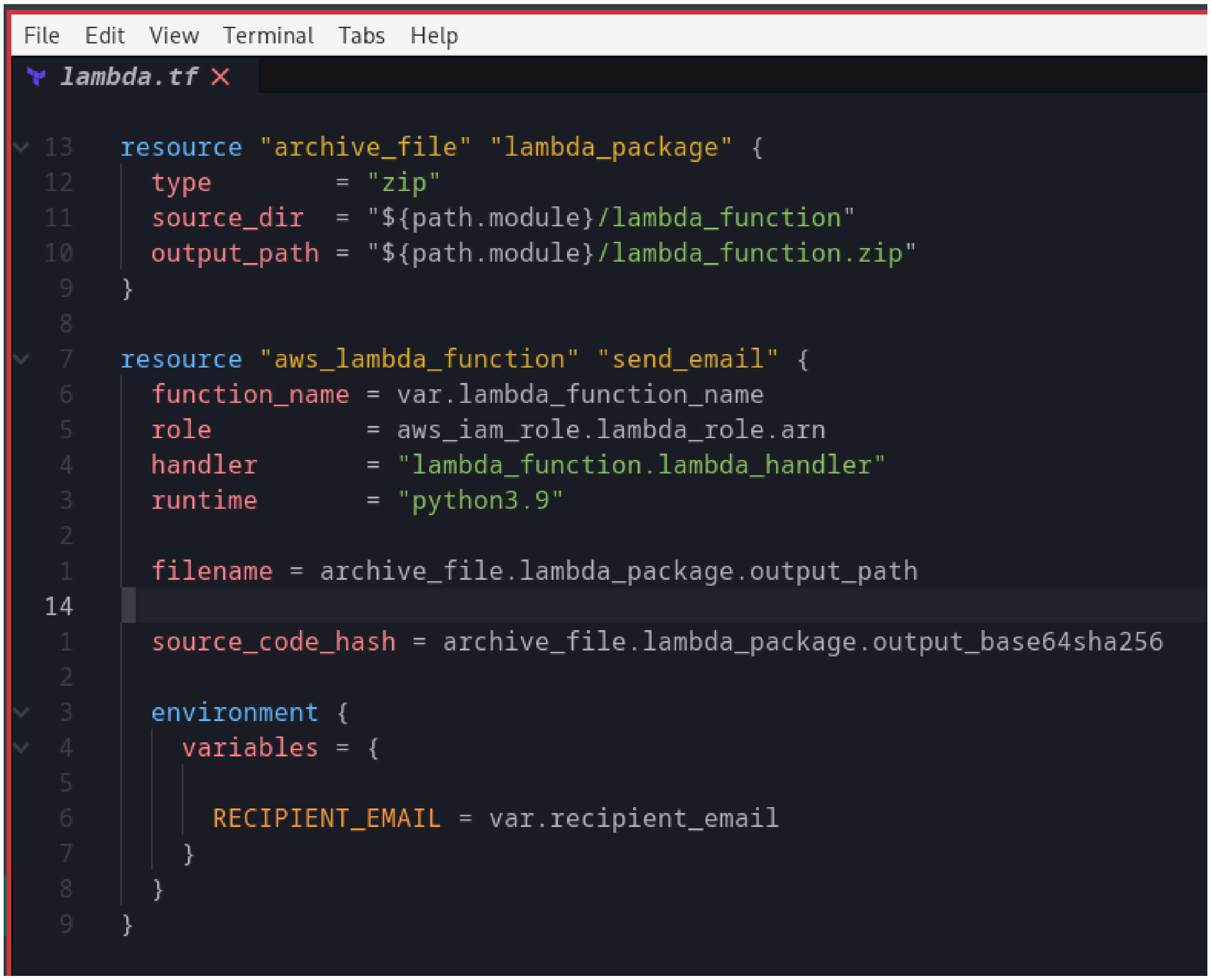
```
resource "aws_ses_domain_identity" "ses_domain" {
  domain = var.domain
}
```

3-create lambda function to send email



```python
import boto3
import os
from datetime import datetime

ses = boto3.client('ses')

def lambda_handler(event, context):
    timestamp = datetime.utcnow().strftime('%Y-%m-%d %H:%M:%S UTC')
    environment = os.environ['ENVIRONMENT']
    sender = os.environ['SENDER_EMAIL']
    receiver = os.environ['RECEIVER_EMAIL']

    subject = f"State File  - {environment} Environment"
    body = f"Terraform state file has been changed.\n\nTime: {timestamp}\nEnvironment: {environment}"

    response = ses.send_email(
        Source=sender,
        Destination={'ToAddresses': [receiver]},
        Message={
            'Subject': {'Data': subject},
            'Body': {'Text': {'Data': body}}
        }
    )

    return {"statusCode": 200, "body": "Email sent successfully!"}
```

4-create trigger to detect changes in state file and send the email
in email mention time of change and environment (dev or prod)



```
resource "archive_file" "lambda_package" {
  type        = "zip"
  source_dir  = "${path.module}/lambda_function"
  output_path = "${path.module}/lambda_function.zip"
}

resource "aws_lambda_function" "send_email" {
  function_name = var.lambda_function_name
  role          = aws_iam_role.lambda_role.arn
  handler       = "lambda_function.lambda_handler"
  runtime       = "python3.9"

  filename = archive_file.lambda_package.output_path

  source_code_hash = archive_file.lambda_package.output_base64sha256

  environment {
    variables = {

      RECIPIENT_EMAIL = var.recipient_email
    }
  }
}
```