Terraform Questions:

1. What is Infrastructure as Code?
   - IAC is used to provision AWS services and creating an automated environment using codes like HCL from Hashicorp

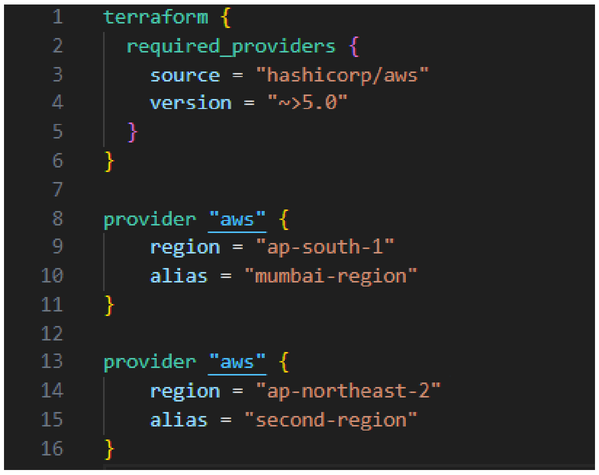
2. What are the benefits of IaC?
   - IAC is mostly used to provision the services using code so we know what services have been created using the code and with what configurations
   - They are easily updated using the code and the configurations are updated by refreshing them
   - They help to create an automated environment for the process easily

3. What are the advantages of Terraform?
   - They are used to create IAC for any cloud services possible
   - They have the creation of statefiles which are important to know what are needed for the services
   - They have the use of the modules which make the coding easy by giving out model codes for an bigger services and the use of variables makes the coding easy not by giving out the data repeatedly so the use of variables make them possible

4. What is the purpose of the Terraform State?
   - They are used to store the configurations, resources used and the infrastructure management are done using the terraform state purpose
   - The state file is mostly used to manage the infrastructure that's being created

5. What is the name of the terraform state file?
   - Terraform.tfstate

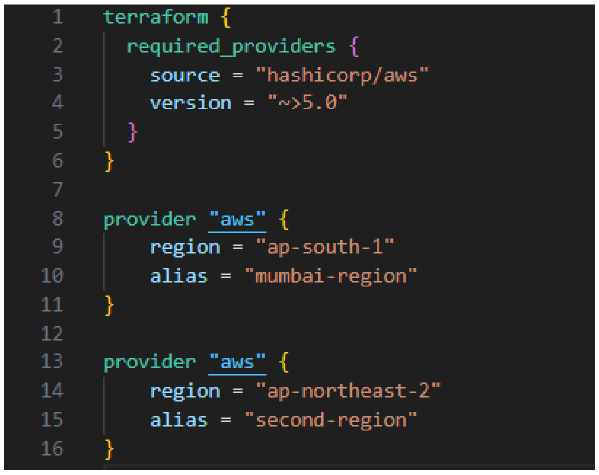
6. What are the Providers?
   - There are 2 providers they are the required providers and the providers
   - Required provider are those source and the version of the IAC code type used such as the source is hashicorp/aws and the version is 5.0
   - Providers are those that contains the region that is being used in the code
   - These providers are used to create an communication between the terraform and the cloud service your going to use



```
terraform {
  required_providers {
    source = "hashicorp/aws"
    version = "~>5.0"
  }
}

provider "aws" {
    region = "ap-south-1"
    alias = "mumbai-region"
}

provider "aws" {
    region = "ap-northeast-2"
    alias = "second-region"
}
```

7. How do you configure a Provider?
   - Required providers – source – “hashicorp/aws” and the version – “~5.0”
   - Providers – region – “ap-south-1”

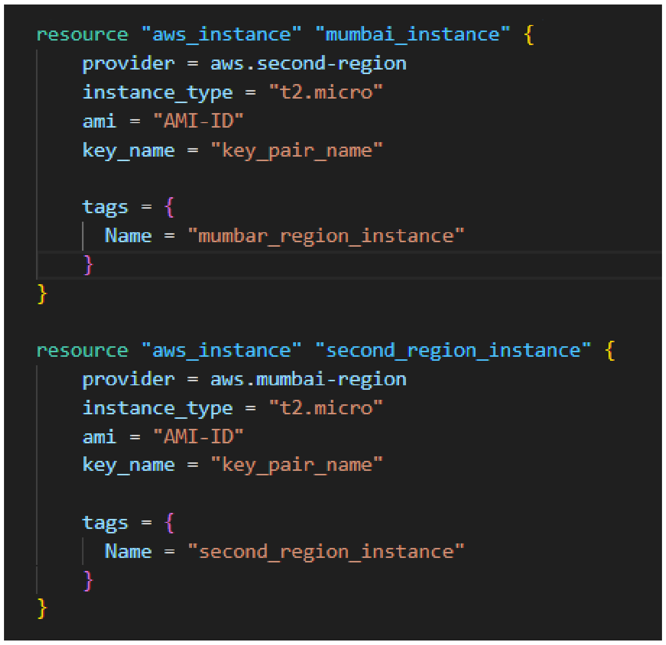
8. How do you upgrade to the latest acceptable version of the provider?
   - Just update the version in the terraform block present where the required providers contain the topic of the version in the IAC code used so we can just change it there



```
terraform {
  required_providers {
    source = "hashicorp/aws"
    version = "~>5.0"
  }
}

provider "aws" {
    region = "ap-south-1"
    alias = "mumbai-region"
}

provider "aws" {
    region = "ap-northeast-2"
    alias = "second-region"
}
```

Use of “~>5.0” can be changed when a new version of the terraform source is being released

9. How do you configure Multiple Provider Instances?
   - To configure multiple provider instance, we can just give multiple regions in multiple provider blocks
   - But for an identification we should use the keyword alias to know which provider is for which region

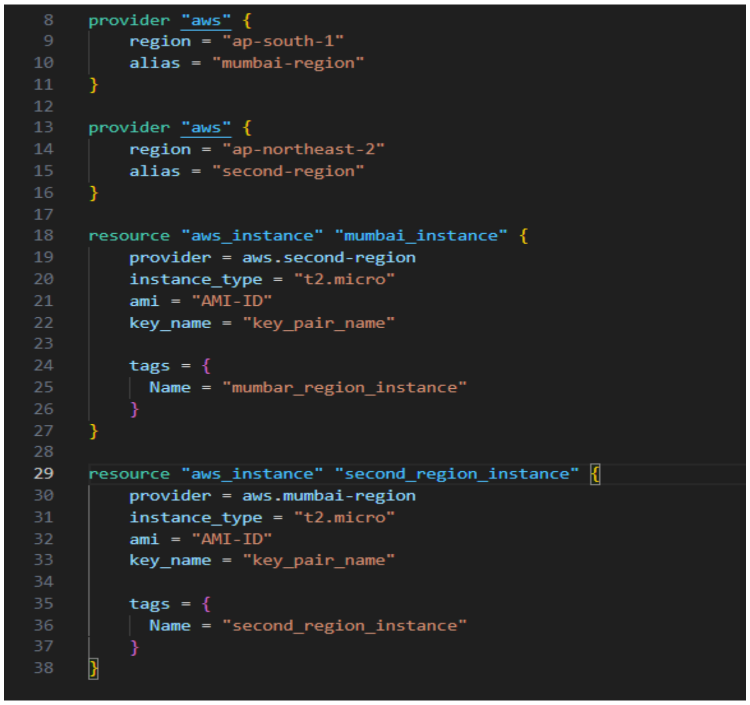


```
resource "aws_instance" "mumbai_instance" {
    provider = aws.second-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "mumbar_region_instance"
    }
}

resource "aws_instance" "second_region_instance" {
    provider = aws.mumbai-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "second_region_instance"
    }
}
```

As shown using tags can be used to view all the created instances as separate

10. Why do we need Multiple Provider instances?
    - The use of having an multiple provider instance is that , we need to deploy the same type application and instance in an different region with an same configuration so creating an multiple provider is useful in that case

11. How do we define multiple Provider configurations?



```
provider "aws" {
    region = "ap-south-1"
    alias = "mumbai-region"
}

provider "aws" {
    region = "ap-northeast-2"
    alias = "second-region"
}

resource "aws_instance" "mumbai_instance" {
    provider = aws.second-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "mumbar_region_instance"
    }
}

resource "aws_instance" "second_region_instance" {
    provider = aws.mumbai-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "second_region_instance"
    }
}
```

The above code is used to create an EC2 instance in two different regions

12. What is the command to initialize the directory?
    - The command is terraform init which is used to initialise the terraform working directory for the project to begin

13. Why do we need to initialize the directory?
    - They are used to initialise because they are used to check the downloading the necessary provider and plugins and they are used to configure the functions that are usefull for the process of the terraform

14. If different teams are working on the same configuration. How do you make files to have consistent formatting?
    - First, we have to format the process of codes that are being present in the process, where all the files are to be formatted
    - Before every commit the files of the terraform must be formatted so using the terraform fmt before every commit is useful

15. If different teams are working on the same configuration. How do you make files to have syntactically valid and internally consistent?
    - Use the terraform validate to check whether the files are formed in an correct way and whether there is any form of errors present in the code of terraform

16. What is the command to create infrastructure?
    - Terraform apply to create the infrastructure

17. What is the command to show the execution plan and not apply?
    - Terraform plan

18. How do you inspect the current state of the infrastructure applied?
    - Terraform show – to view the state file in the human perspective

19. If your state file is too big and you want to list the resources from your state. What is the command?
    - Terraform state list

20. What are Provisioners?
    - They are the function that are used to execute the code and scripts in the local machine where they are used create or destroy

21. How do you define provisioners?
    - They are the function that are used to execute the code and scripts in the local machine where they are used create or destroy

22. What are the types of provisioners?
    - Generic provisioner - they are the independent file system
    - Vendor provisioner

23. What is a local-exec provisioner and when do we use it?
    - They are used execute commands for the scripts in the machine where the terraform is running, they are used to run the terraform code in the local machine

24. What is a remote-exec provisioner and when do we use it?
    - They are used to run the commands and scripts of the terraform on the remote storage

25. When terraform mark the resources are tainted?
    - The terraform marks the resource as tainted when they resource are having an fault in their version or the resource might be damaged state.
    - So, these problems can be solved using the command terraform apply

26. You applied the infrastructure with terraform apply and you have some tainted resources.
    - When apply is processed they show what are the tainted resource after finding out we can replace the resource with the new version or untaint the version

27. How do you manually taint a resource?
    - Terraform taint aws_instance.example

28. What does this symbol version = "~> 1.0" mean when defining versions?
    - Greater than equal, so the specified version can be used or the version more than that

29. Terraform supports both cloud and on-premises infrastructure platforms. Is this true?
    - Terraform supports both the cloud and on-premises infra by giving out IAC methods and also provisioning automated services to the cloud

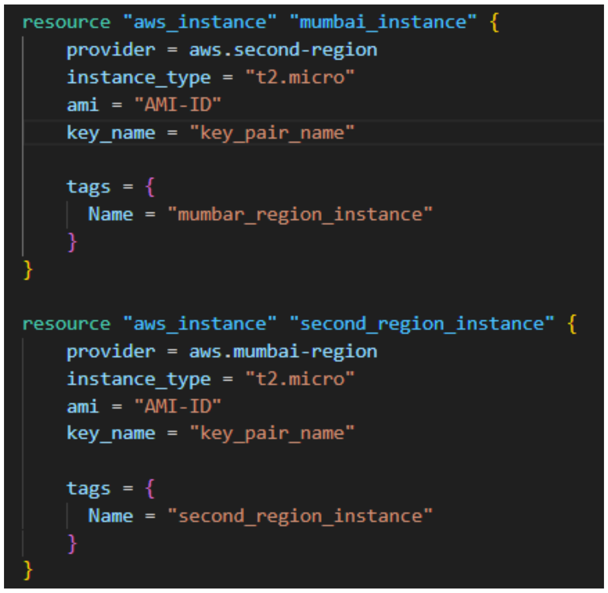
30. You are provisioning the infrastructure with the command terraform apply and you noticed one of the resources failed. How do you remove that resource without affecting the whole infrastructure?

- Terraform state rm <resource_type.resource_name>
- Terraform state rm aws_instance.web
- Terraform apply -auto-approve

31. What is the command import?
    - Import used to bring the external resources, data and the modules into the terraform program to be created

32. What are workspaces?
    - Workspace is an environment where all the codes , digital expression works are done where all the tools required for the project is being held together

33. You are working on the different workspaces and you want to use tags based on the workspace. How do you achieve that?
    - We can use tags for different servers in the workspace for example:



```
resource "aws_instance" "mumbai_instance" {
    provider = aws.second-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "mumbar_region_instance"
    }
}

resource "aws_instance" "second_region_instance" {
    provider = aws.mumbai-region
    instance_type = "t2.micro"
    ami = "AMI-ID"
    key_name = "key_pair_name"

    tags = {
      Name = "second_region_instance"
    }
}
```

Here the tag name first server is used to identify different servers in workspaces

34. You want to create a parallel, distinct copy of a set of infrastructure in order to test a set of changes before modifying the main production infrastructure. How do you achieve that?
    - This can be done by storing the IAC code in the version control system and can be deployed In any version of the system to check them in parallel so the version can be rolled back if any problem is being faced

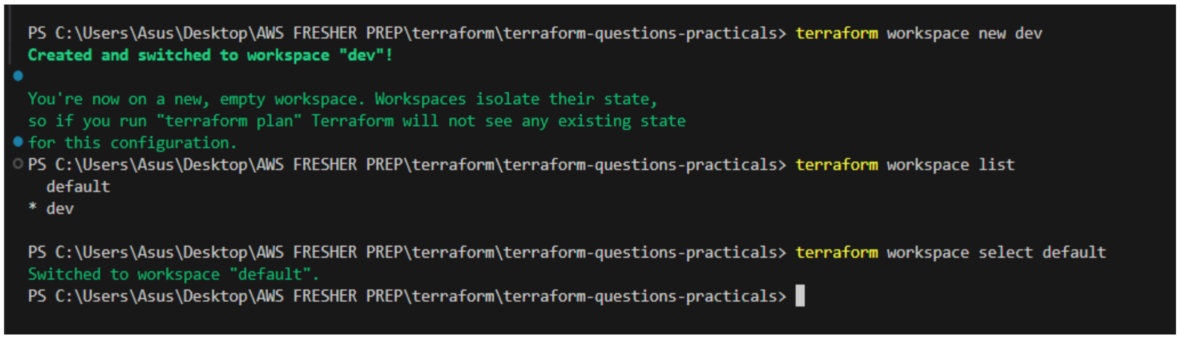
35. What is the command state?
    - Terraform state – is used to manage the state file of the terraform

36. How do you list the resources for the given name?
    - Terraform state list – this command gives out all the system configuration that have been done in the terraform code

37. What is the command that shows the attributes of a single resource in the state file?
    - Terraform state show – with the resource address we can we the state of the single resource

38. If terraform crashes, where should you see the logs?
    - If the terraform code crashes they automatic store a file known as crash.log
    - So, look for the file named as crash.log

39. You are building infrastructure for different environments for example test and dev. How do you maintain separate states?
    - We can use command – terraform workspace new test and terraform workspace new dev
    - So now we have to select the workspace – terraform workspace select dev
    - By using these commands, we can create 2 different workspaces and 2 different states for each workspace



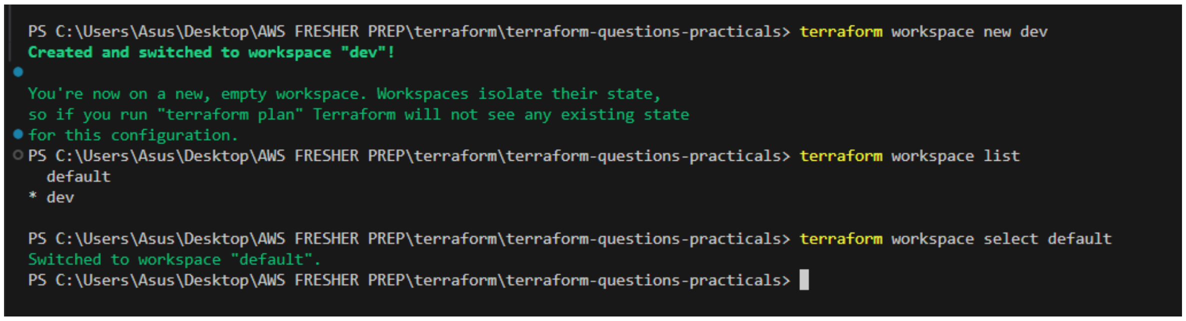
```
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace list
  default
* dev

PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace select default
Switched to workspace "default".
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals>
```

40. What is the command to pull the remote state?
    - Terraform state pull is used to derive the state from the s3 bucket created om the backend
    - This state file is automatically stored in the s3 bucket when the code is initialised
    - So we can just pull the statefile from the remote file

41. Your team has decided to use terraform in your company and you have existing infrastructure. How do you migrate your existing resources to terraform and start using it?
    - First we have to verify the data that have been existing in the already in the infrastructure.
    - Gather all the infrastructure that is need for the migration to terraform
    - Although the infrastructure is present we need to write all the configuration resource file from the start

- Then we have to import the existing instances from the server so the existing instances are imported
- Now after import verify the statelist present
- The import the state file form the backend storage

42. When you are working with the workspaces how do you access the current workspace in the configuration files?
    - Working with the configuration files of an different workspace is needed to be selected, the desired workspace is needed to be selected first

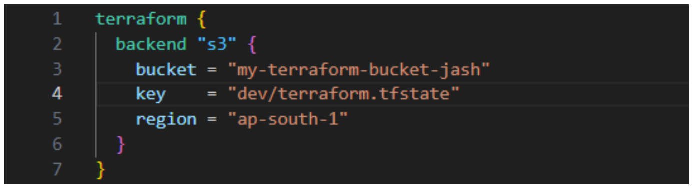


```
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace new dev
Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing state
for this configuration.
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace list
  default
* dev

PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> terraform workspace select default
Switched to workspace "default".
PS C:\Users\Asus\Desktop\AWS FRESHER PREP\terraform\terraform-questions-practicals> 
```

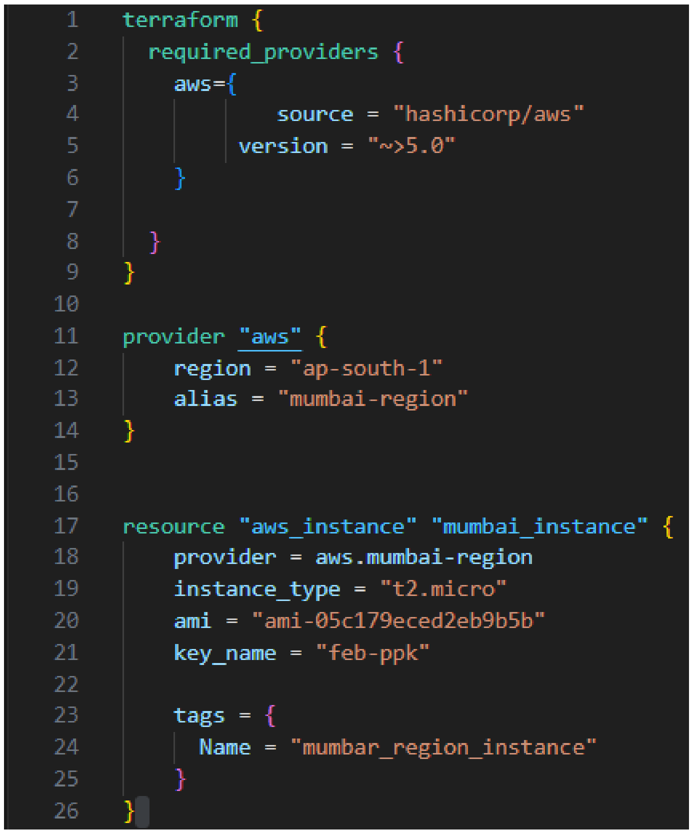
43. When you are using workspaces where does the Terraform save the state file for the local state?
    - For terraform to store an state file in locally by creating an terraform.tfstate In the workspace you are executing the code they create it locally there

44. How do you remove items from the Terraform state?
    - First we have to list the statefile – terraform state list
    - Next we have to remove the resource that are not needed
    - Terraform state rm <resource name>

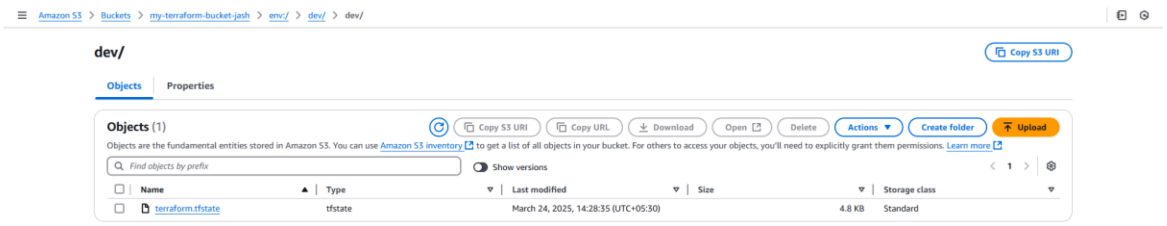
45. How do you move the state from one source to another?
    - First we have to check the current location of the statefile
    - Move the statefile to the newly created bucket for backup
    - Now initialise the terraform by using migrate-state
    - Terraform init -migrate-state
    - Then list the state present – terraform state list (to verify the migration)
    - Now plan the code to check whether the code is correct
    - And they apply to create the code service



```
terraform {
  backend "s3" {
    bucket = "my-terraform-bucket-jash"
    key    = "dev/terraform.tfstate"
    region = "ap-south-1"
  }
}
```



```
terraform {
  required_providers {
    aws={
            source = "hashicorp/aws"
        version = "~>5.0"
    }

  }
}

provider "aws" {
    region = "ap-south-1"
    alias = "mumbai-region"
}


resource "aws_instance" "mumbai_instance" {
    provider = aws.mumbai-region
    instance_type = "t2.micro"
    ami = "ami-05c179eced2eb9b5b"
    key_name = "feb-ppk"

    tags = {
      Name = "mumbar_region_instance"
    }
}
```



dev/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

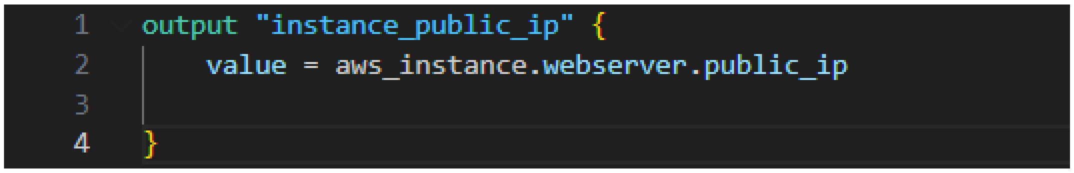
Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	March 24, 2025, 14:28:35 (UTC+05:30)	4.8 KB	Standard

46. What is a Terraform Module?
    - Terraform modules are those ready-made code which are useful to use to run an large service at once they make the mode of coding easier

47. Why do we use modules for?
    - They are like ready made components of code and libraries that helps in the process of code reusability and they contain ready made codes for different services

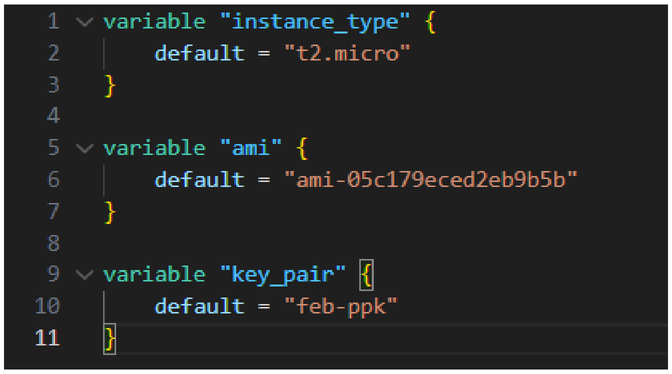
48. What are the different source types for calling modules?
    - Terraform registry
    - Github

49. Where do you put output variables in the configuration?
    - We put the output variables in the output.tf file where we can protect the sensitive type of output from being shown



```
output "instance_public_ip" {
    value = aws_instance.webserver.public_ip

}
```

50. How do you pass input variables in the configuration?
    - We can pass the input variables in the variables.tf file where they can be used for easy coding purpose



```
variable "instance_type" {
    default = "t2.micro"
}

variable "ami" {
    default = "ami-05c179eced2eb9b5b"
}

variable "key_pair" {
    default = "feb-ppk"
}
```