

18th July

multi hybrid cloud - Google Drive | Untitled document - Google Doc | visual studio code - Google Search | +

google.com/search?q=visual+studio+code&oq=visual&aqs=chrome.1.0l8.3034j1j7&sourceid=chrome&ie=UTF-8

Apps New Tab Search Inbox (157) - 2019p... Inbox (1,144) - vish... Inbox (252) - 2019p... Gmail YouTube Maps 29th APRIL - Google... YAMLint - The YAM...

visual studio code

All Images News Books Videos More Settings Tools

About 65,80,00,000 results (0.56 seconds)

Ad · aws.amazon.com/ ▾

Visual Studio, Direct to AWS - Integrate .NET Apps With Ease

Gear up with serverless framework, containers, and AI/ML with this VS Toolkit.

code.visualstudio.com ▾

Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and ...

Download Visual Studio Code

Visual Studio Code is free and available on your favorite ...

Docs

Intro Videos - Windows - Basic Editing - Key Bindings - Java - ...

Updates

Learn what is new in the Visual Studio Code June 2020 ...

Extension API

Visual Studio Code has a rich extension API. Learn how to ...

 More images

Visual Studio Code

Software

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets,

Type here to search

Windows 10 64 bit ENG IN 6:48 PM 7/18/2020

multi hybrid cloud - Google Drive | Untitled document - Google Doc | Download Visual Studio Code | +

code.visualstudio.com/download

Apps New Tab Search Inbox (157) - 2019p... Inbox (1,144) - vish... Inbox (252) - 2019p... Gmail YouTube Maps 29th APRIL - Google... YAMLint - The YAM...

Visual Studio Code Docs Updates Blog API Extensions FAQ

Search Docs Download

Version 1.47 is now available! Read about the new features and fixes from June.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.


Windows
Windows 7, 8, 10


.deb
Debian, Ubuntu


.rpm
Red Hat, Fedora, SUSE

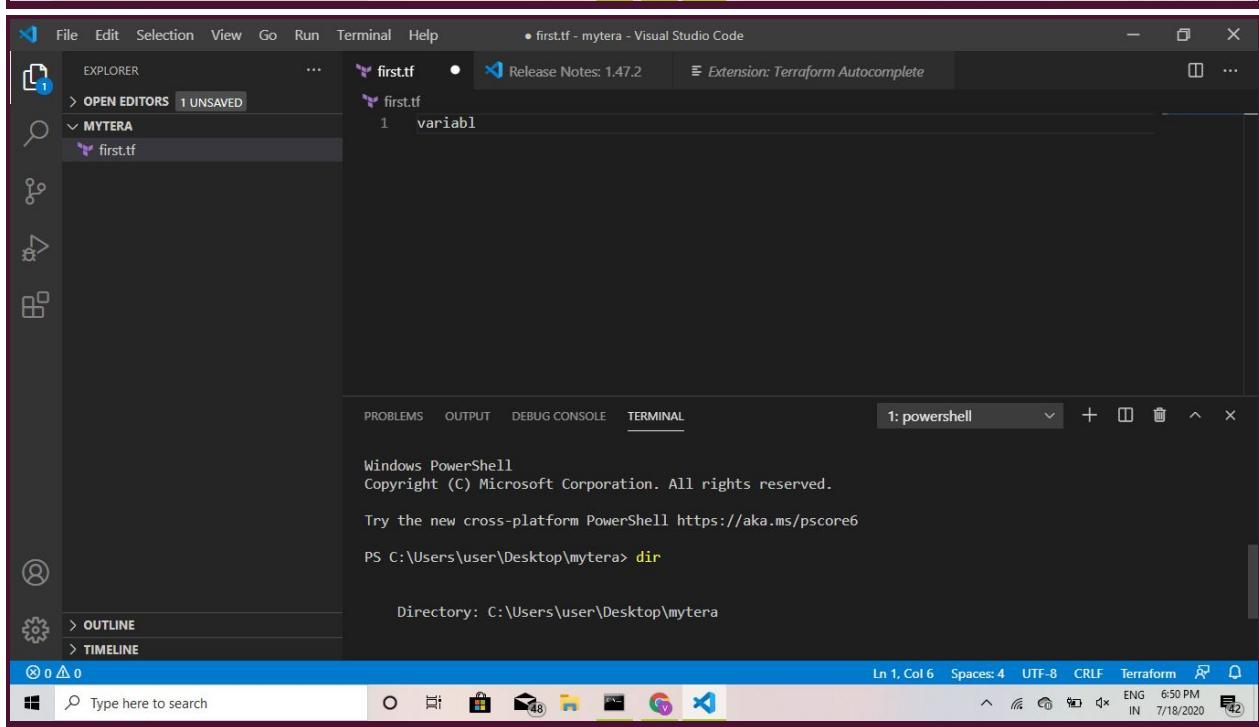
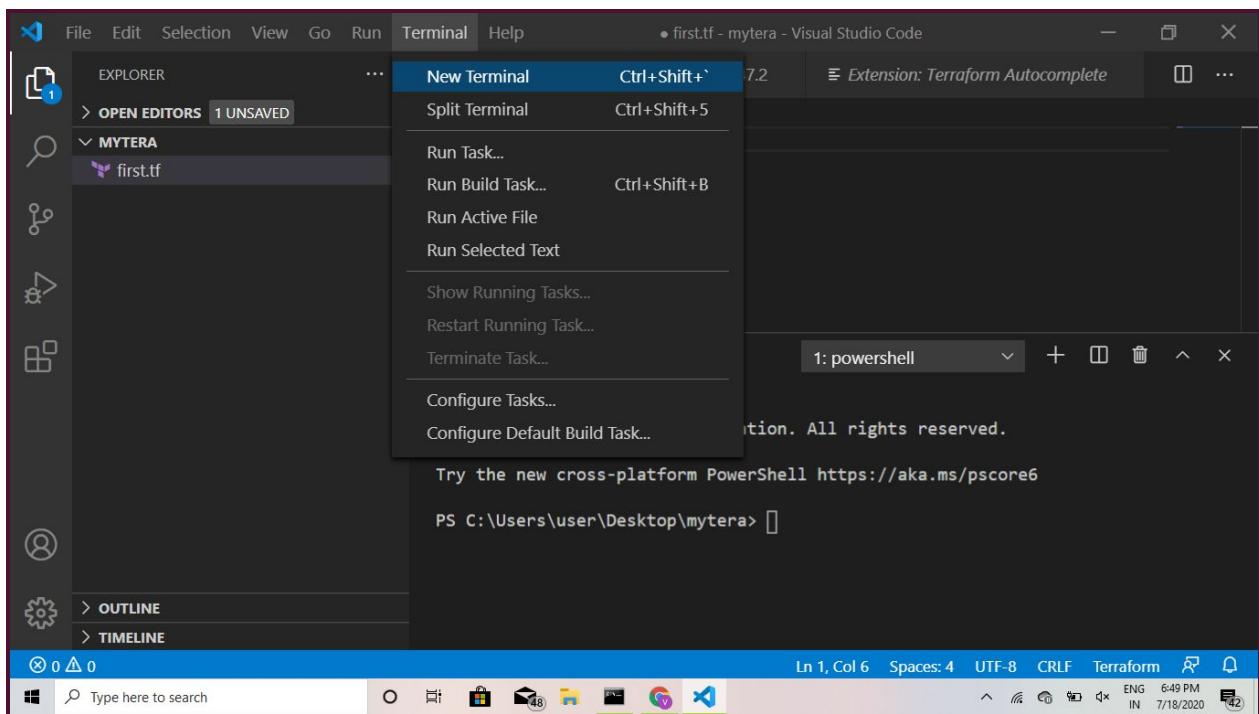

Mac
macOS 10.10+

User Installer 64 bit 32 bit ARM
System Installer 64 bit 32 bit ARM
.zip 64 bit 32 bit ARM

.deb 64 bit
.rpm 64 bit
.tar.gz 64 bit

Type here to search

Windows 10 64 bit ENG IN 6:48 PM 7/18/2020



File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

OPEN EDITORS 1 UNSAVED

MYTERA first.tf

1 variable

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: powershell

Directory: C:\Users\user\Desktop\mytera

Mode	LastWriteTime	Length	Name
-a---	7/18/2020 6:44 PM	0	first.tf

OUTLINE TIMELINE PS C:\Users\user\Desktop\mytera> [42]

Type here to search

Ln 1, Col 6 Spaces: 4 UTF-8 CRLF Terraform

ENG 6:50 PM IN 7/18/2020

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

OPEN EDITORS

MYTERA first.tf

variable "myvar" [

PROBLEMS TERMINAL ...

1: powershell

Directory: C:\Users\user\Desktop\mytera

Mode	LastWriteTime	Length	Name
-a---	7/18/2020 6:44 PM	0	first.tf

OUTLINE TIMELINE PS C:\Users\user\Desktop\mytera> [43]

Type here to search

Ln 2, Col 5 Spaces: 4 UTF-8 CRLF Terraform

ENG 6:51 PM IN 7/18/2020

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** first.tf - mytera - Visual Studio Code
- Explorer Panel:** Shows a folder structure under MYTERA: first.tf (selected), terraform.tfstate, and .terraform.
- Editor Panel:** Displays the contents of first.tf:

```
variable "myvar" {
```
- Terminal Panel:** Shows a PowerShell session output:

```
PS C:\Users\user\Desktop\mytera> terraform apply
var.myvar
Enter a value: pop

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\mytera>
```
- Bottom Status Bar:** Ln 2, Col 5, Spaces: 4, UTF-8, CRLF, Terraform, system icons, and a battery icon showing 43%.

A screenshot of Visual Studio Code interface. The left sidebar shows the Explorer, Open Editors, Outline, and Timeline. The main editor window displays a Terraform configuration file named `first.tf` with the following code:

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

output "name123" {
  value = var.myvar
}
```

The terminal tab shows the command `terraform apply` being run in a PowerShell window, resulting in:

```
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

The status bar at the bottom indicates the code is at Line 5, Column 14, with 4 spaces, using UTF-8 encoding, and is in CRLF mode. The Terraform extension is active.

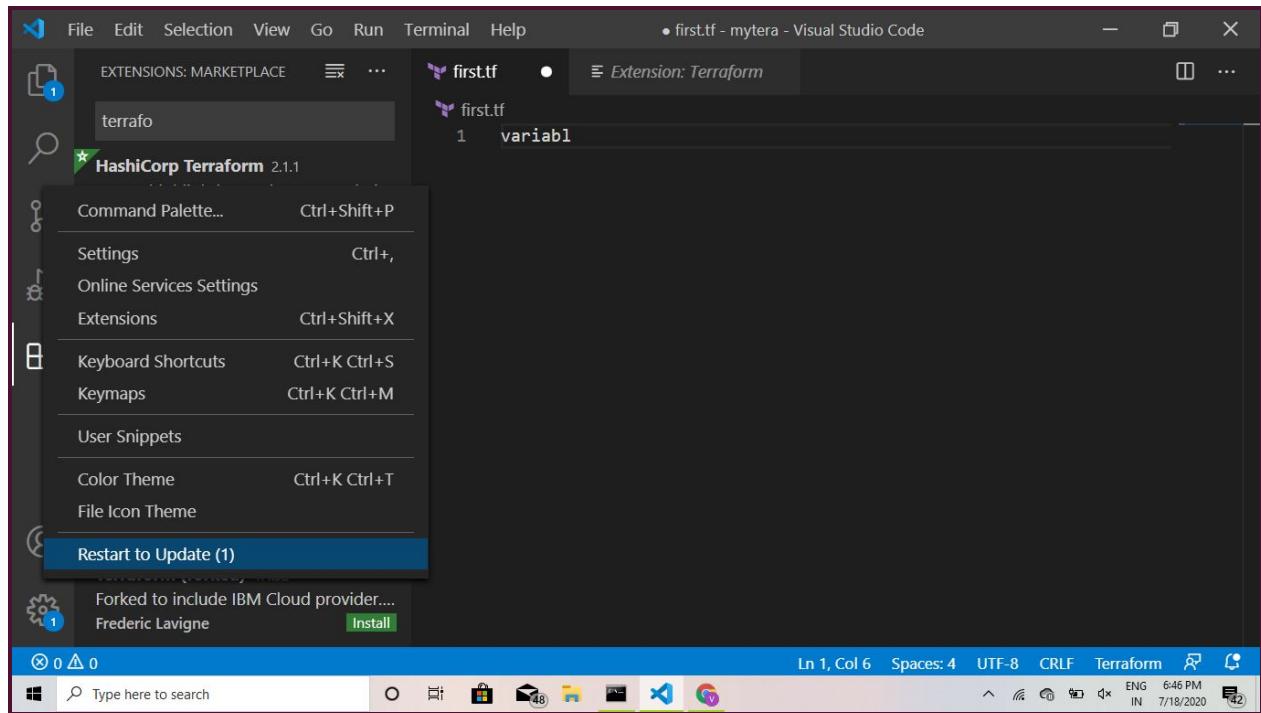
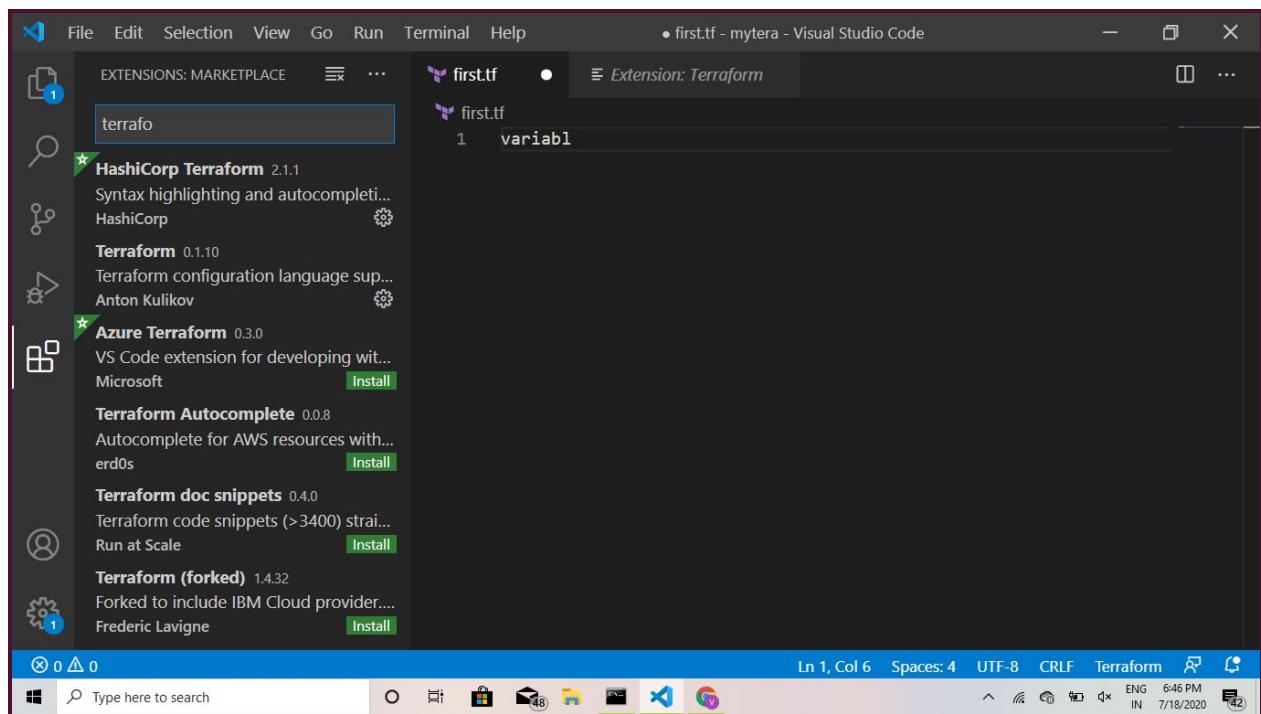
A screenshot of Visual Studio Code interface, similar to the first one but with a change in the Terraform configuration. The `first.tf` file now contains:

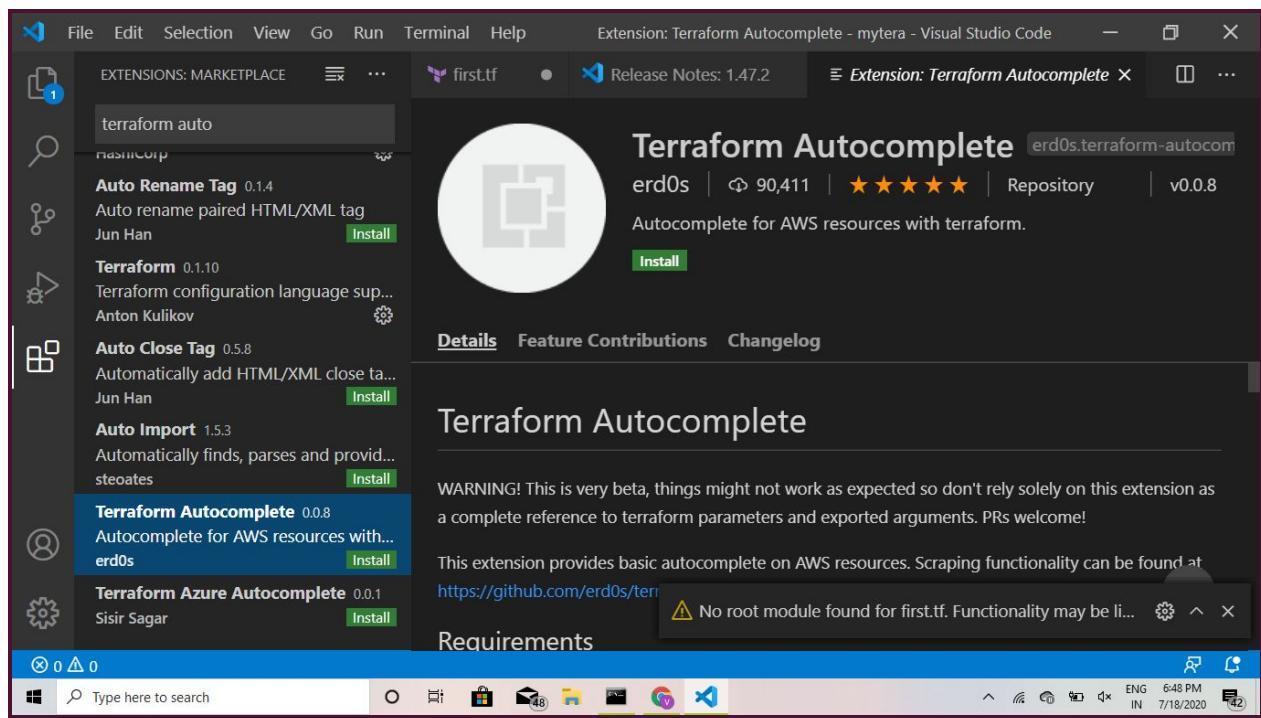
```
variable "myvar" {
  default = "t2.micro"
}

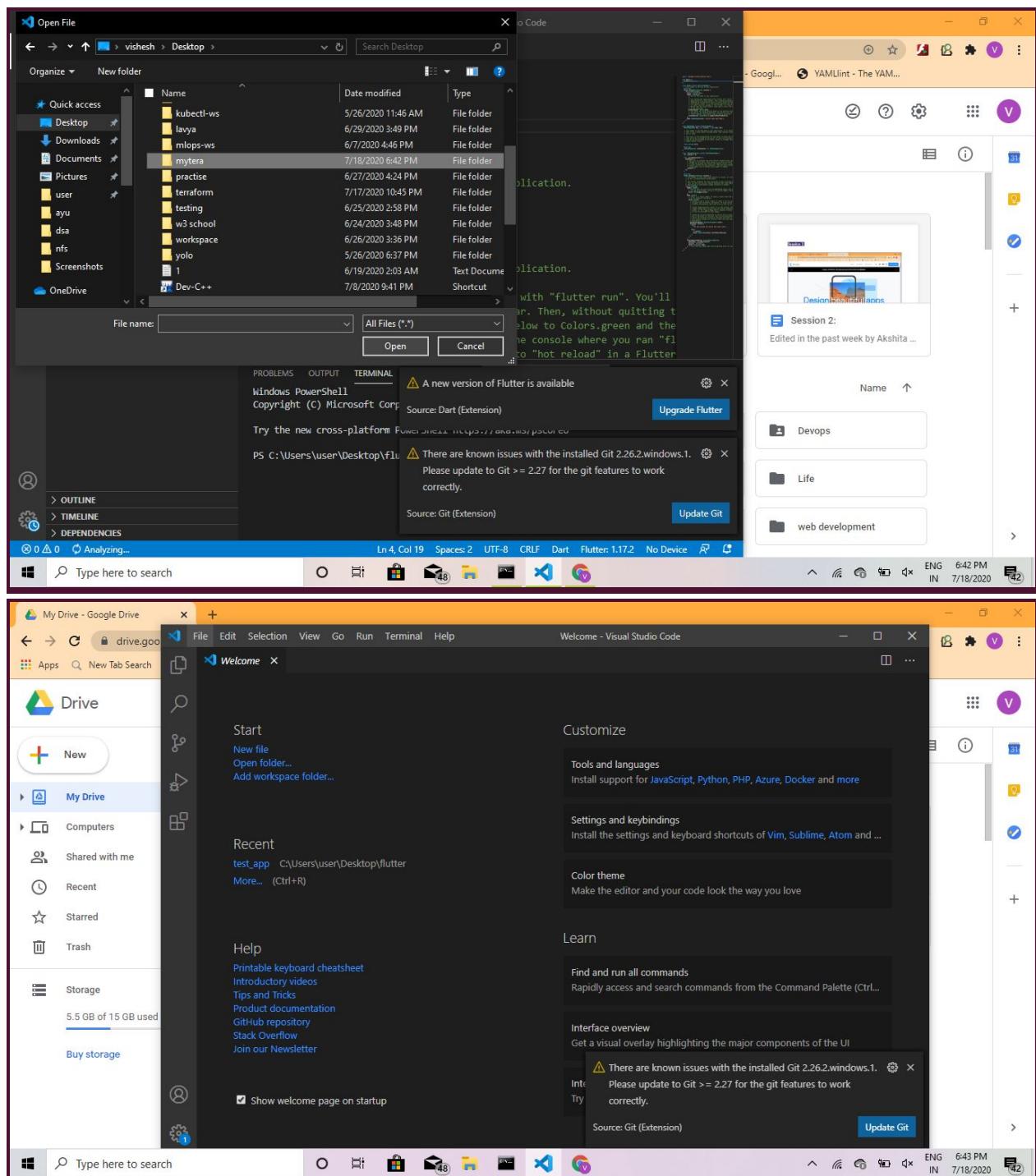
output "name123" {
  value = "hi ${var.myvar}"
}
```

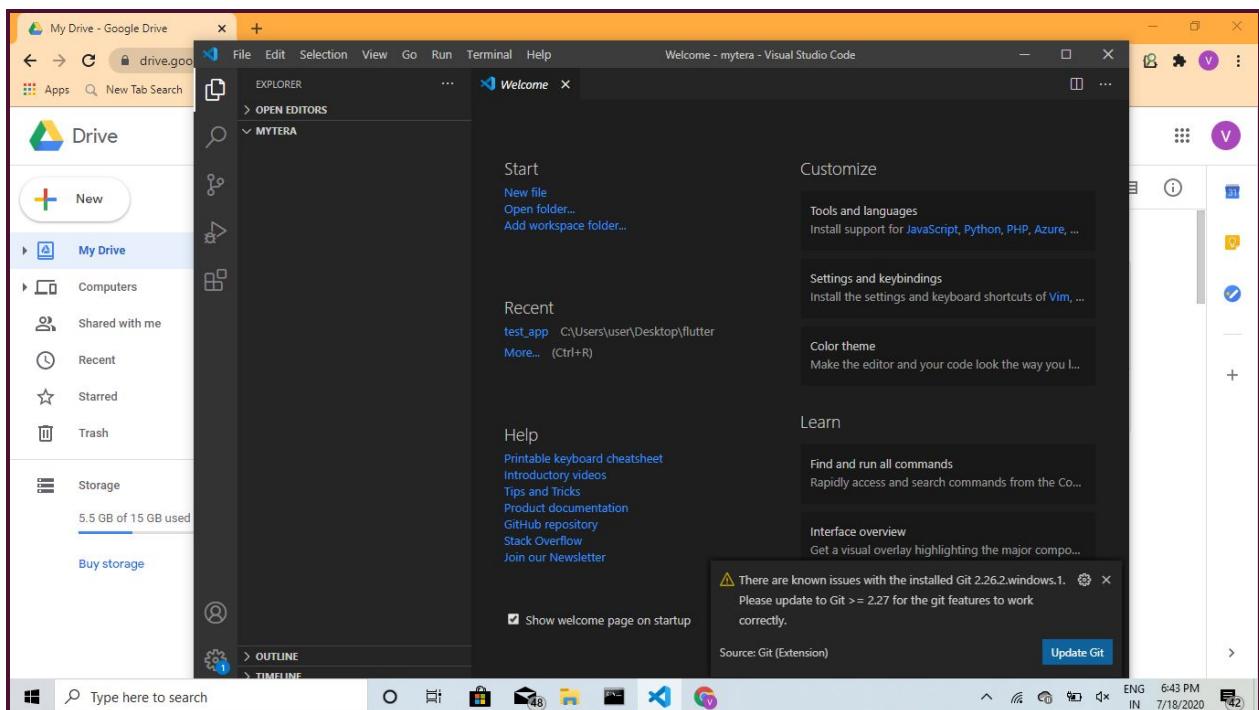
The terminal output remains the same as the first screenshot, showing the successful application of the Terraform configuration.

The status bar at the bottom indicates the code is at Line 7, Column 2 (52 selected), with 4 spaces, using UTF-8 encoding, and is in CRLF mode. The Terraform extension is active.









This screenshot shows Visual Studio Code with a Terraform configuration file named "first.tf" open in the editor. The code defines a variable "myvar" of type bool with a default value of "true". It then outputs a value "name123" which is set to "hi \${var.myvar}". The "TERMINAL" tab shows the output of the "terraform apply" command, which completes successfully with no changes made. The status bar at the bottom shows the date and time as 7/18/2020 6:58 PM.

```
variable "myvar" {
  type = bool
  default = "true"
}

output "name123" {
  value = "hi ${var.myvar}"
}

provider "aws" {
  region = "ap-south-1"
  profile="myvishesh"
}

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

```
}

resource "aws_instance" "instance" {
    ami           = "ami-005956c5f0f757d37"
    instance_type = var.mytype
    availability_zone = "ap-south-1a"
    key_name      = "mykey1111.pem"
    tags = {
        Name = "instance"
    }
}
```

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform first.tf terraform.tfstate terraform.tfstate.backup

first.tf

```
4 }
5 variable "mytype" {
6   default="t2.micro"
7 }
8 resource "aws_instance" "instance" {
9   ami           = "ami-005956c5f0f757d37"
10  instance_type = var.mytype
11  availability_zone = "ap-south-1a"
12  key_name      = "mykey1111.pem"
13  tags = {
14    Name = "myinstance"
15 }
```

PROBLEMS TERMINAL ... 1: powershell

An input variable with the name "mytype" has not been declared. Did you mean "mytpye"?

PS C:\Users\user\Desktop\mytera> **terraform validate**
Success! The configuration is valid.

PS C:\Users\user\Desktop\mytera>

Ln 5, Col 16 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

Windows Taskbar: File Explorer, Mail, Edge, Google Chrome, VS Code, File Manager, Task View

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform first.tf terraform.tfstate terraform.tfstate.backup

first.tf

```
4 }
5 variable "mytype" {
```

PROBLEMS TERMINAL ... 1: powershell

PS C:\Users\user\Desktop\mytera> **terraform plan**
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be persisted to local or remote state storage.

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# aws_instance.instance will be created
+ resource "aws_instance" "instance" {
```

Ln 5, Col 16 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

Windows Taskbar: File Explorer, Mail, Edge, Google Chrome, VS Code, File Manager, Task View

A screenshot of Visual Studio Code interface showing a Terraform configuration file named `first.tf`. The code defines a variable `mytype` with several attributes:

```
variable "mytype" {  
    + delete_on_termination = (known after apply)  
    + device_name          = (known after apply)  
    + encrypted             = (known after apply)  
    + iops                  = (known after apply)  
    + kms_key_id            = (known after apply)  
    + volume_id              = (known after apply)  
    + volume_size            = (known after apply)  
    + volume_type             = (known after apply)  
}  
}  
  
Plan: 1 to add, 0 to change, 0 to destroy.
```

The terminal output shows the plan details:

```
Note: You didn't specify an "-out" parameter to save this plan, so Terraform  
can't guarantee that exactly these actions will be performed if
```

The status bar at the bottom indicates the current state of the system.

current state

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** terraform.tfstate - mytera - Visual Studio Code.
- Sidebar:** EXPLORER, OPEN EDITORS, MYTERA (containing .terraform, first.tf, terraform.tfstate, terraform.tfstate.backup), OUTLINE, and TIMELINE.
- Central Area:** The terraform.tfstate file is open, showing its JSON structure. The file contains information about a managed AWS instance resource.
- Terminal:** The terminal shows the command "Apply complete! Resources: 1 added, 0 changed, 0 destroyed." and the path "PS C:\Users\user\Desktop\mytera>".
- Bottom Status Bar:** Line 1, Col 1, Spaces: 2, UTF-8, LF, JSON, and a search bar.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows a tree view of files and folders. The "MYTERA" folder is expanded, containing ".terraform", ".terraform.tfstate.lock.info", "first.tf" (selected), "terraform.tfstate" (highlighted), and "terraform.tfstate.backup".
- Editor Area (Top Right):** Displays the contents of "first.tf".

```
1  [
2    "version": 4,
3    "terraform_version": "0.12.26",
4    "serial": 7,
5    "lineage": "d7d84654-91fc-c80c-6588-82350f7d110f",
6    "outputs": {},
7    "....."
```
- Terminal (Bottom):** Shows the output of a Terraform command:

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
```
- Status Bar (Bottom):** Shows file statistics (Ln 1, Col 1), encoding (UTF-8), line endings (LF), and JSON format.

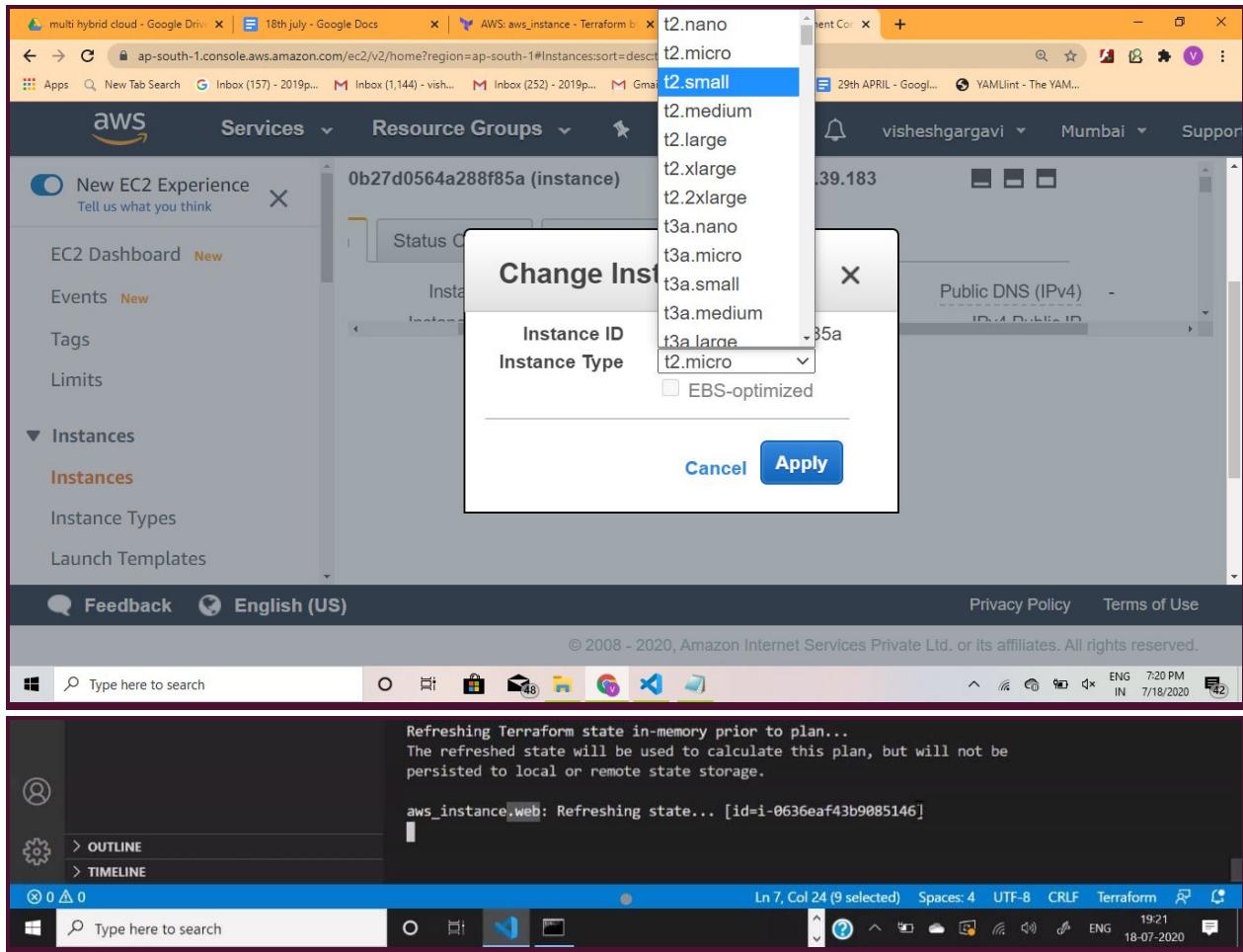
The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane is open with the 'Instances' section selected. In the main content area, a table lists one instance named 'instance'. The instance details are shown in a modal window, including its Instance ID (i-0b27d0564a288f85a), Public DNS (ec2-15-206-94-146.ap-south-1.compute.amazonaws.com), and Instance Type (t2.micro).

The screenshot shows Visual Studio Code with a dark theme. The Explorer sidebar shows a project structure with files like 'first.tf', '.terraform', and 'terraform.tfstate.backup'. The main editor window displays a JSON-like Terraform state file named 'terraform.tfstate.backup'. The terminal at the bottom shows the command 'terraform apply' completed successfully, with the message 'Apply complete! Resources: 1 added, 0 changed, 0 destroyed.'

```
version": 4,
"terraform_version": "0.12.26",
"serial": 5,
"lineage": "d7d84654-91fc-c80c-6588-82350f7d110f",
"outputs": {
  "name123": {
    "value": "hi true",
    "type": "string"
  }
},
"resources": []
```

The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane is open with the 'Instances' section selected. In the main area, a single instance named 'instance' is listed. A context menu is open over this instance, showing options like 'Connect', 'Get Windows Password', 'Create Template From Instance', and 'Launch More Like This'. Under 'Launch More Like This', the 'Instance State' option is expanded, showing 'Start', 'Stop', 'Stop - Hibernate', 'Reboot', and 'Terminate'. The instance details panel below shows the instance ID (i-0b27d0564a288f85a), Public DNS (ec2-15-206-94-146.ap-south-1.compute.amazonaws.com), and other metadata.

This screenshot is similar to the one above, but the 'Stop' option under 'Instance State' has been selected. The context menu now includes 'Change Instance Type' under the 'Instance Settings' section. Other options like 'Image', 'Networking', and 'CloudWatch Monitoring' are also visible. The instance details panel remains the same, showing the stopped state of the instance.



terraformer.tfstate

terraformer.tfstate.backup

```
8      }
9
10     resource "aws_instance" "web" {
11       ami           = "ami-07a8c73a650069cf3"
12       volume_size   = 8
13       volume_type   = "gp2"
14     }

Plan: 0 to add, 1 to change, 0 to destroy.

-----
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

terraformer.tfstate

```
get_password_data      = false
hibernation            = false
id                     = "i-0636eaf43b9085146"
instance_state          = "stopped"
~ instance_type          = "t2.small" -> "t2.micro"
ipv6_address_count     = 0
ipv6_addresses          = []
monitoring              = false
primary_network_interface_id = "eni-0e060857d4c0039e3"
private_dns              = "ip-172-31-11-62.ap-south-1.compute.internal"
private_ip                = "172.31.11.62"
security_groups          = [
  "default",
]
source_dest_check        = true
subnet_id                = "subnet-35ef8779"
```

Tags

Limits

i2.micro

running

Initializing

13.235.49.35

File Edit Selection View Go Run Terminal Help terraformer.tfstate - myteria - Visual Studio Code

EXPLORER

> OPEN EDITORS

> MYTERIA

> .terraform

first.tf

{ terraformer.tfstate

terraformer.tfstate.backup

PROBLEMS TERMINAL ...

1: powershell

```
14      {
15       "schema_version": 1,
16       "attributes": {
17         "ami": "ami-08706cb5f68222d09",
18         "arn": "arn:aws:ec2:ap-south-1:410914255776:instance/i-0b27d0564a288f85a",
19         "associate_public_ip_address": true,
20         "availability_zone": "ap-south-1a",
21         "cpu_core_count": 1,
22         "cpu_threads_per_core": 1,
23         "credit_specification": [
24           {
25             "cpu_credits": "standard"
26           }
27         ]
28       }
29     }
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\myteria> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\myteria> terraform apply

0 △ 0

Type here to search

Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON

ENG 7:23 PM IN 7/18/2020

A screenshot of Visual Studio Code showing the terraform.tfstate file content. The code editor displays a JSON object representing an AWS instance. The 'instance_state' field is set to 'running'. Below the code editor, the terminal shows the output of the 'terraform apply' command, which completes successfully.

```
30     "ebs_optimized": false,
31     "ephemeral_block_device": [],
32     "get_password_data": false,
33     "hibernation": false,
34     "host_id": null,
35     "iam_instance_profile": "",
36     "id": "i-0b27d0564a288f85a",
37     "instance_initiated_shutdown_behavior": null,
38     "instance_state": "running",
39     "instance_type": "t2.micro",
40     "ipv6_address_count": 0,
41     "ipv6_addresses": [],
42     "key_name": "mykey1111.pem",
43   },
44 }
```

PROBLEMS TERMINAL ... 1: powershell

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\mytera> terraform apply[]
```

A screenshot of Visual Studio Code showing the terraform.tfstate file content. The 'instance_state' field is now set to 'stopped'. Below the code editor, the terminal shows the output of the 'terraform refresh' command, which completes successfully.

```
30     "ebs_optimized": false,
31     "ephemeral_block_device": [],
32     "get_password_data": false,
33     "hibernation": false,
34     "host_id": null,
35     "iam_instance_profile": "",
36     "id": "i-0b27d0564a288f85a",
37     "instance_initiated_shutdown_behavior": null,
38     "instance_state": "stopped",
39     "instance_type": "t2.micro",
40     "ipv6_address_count": 0,
41     "ipv6_addresses": [],
42     "key_name": "mykey1111.pem",
43   },
44 }
```

PROBLEMS TERMINAL ... 1: powershell

```
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
PS C:\Users\user\Desktop\mytera> terraform refresh
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
PS C:\Users\user\Desktop\mytera> []
```

```
provider "aws" {
  region = "ap-south-1"
  profile="myvishesh"
}

variable "mytype" {
  default="t2.micro"
```

```
}

resource "aws_instance" "instance" {
    ami          = "ami-08706cb5f68222d09"
    instance_type = var.mytype
    availability_zone = "ap-south-1a"
    key_name      = "mykey1111.pem"

    tags = {
        Name = "instance"
    }
}

output "name" {
    value = aws_instance.instance.availability_zone
}
```

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform first.tf {} terraform.tfstate terraform.tfstate.backup

first.tf

```
16   .... Name = "instance"
17   ...
18   }
19   output "name" {
20     value = "aws_instance.instance.availability_zone"
```

PROBLEMS TERMINAL ... 1: powershell

Blocks of type "output" are not expected here.

```
PS C:\Users\user\Desktop\mytera> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
```

No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your

Ln 22, Col 1 (426 selected) Spaces: 2 UTF-8 CRLF Terraform

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform first.tf {} terraform.tfstate terraform.tfstate.backup

first.tf

```
16   .... Name = "instance"
17   ...
18   }
19   output "name" {
20     value = "aws_instance.instance.availability_zone"
```

PROBLEMS TERMINAL ... 1: powershell

No changes. Infrastructure is up-to-date.

```
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:
```

```
name = aws_instance.instance.availability_zone
PS C:\Users\user\Desktop\mytera>
```

Ln 22, Col 1 (426 selected) Spaces: 2 UTF-8 CRLF Terraform

File Edit Selection View Go Run Terminal Help

varr.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform .terraform.tfstate.lock.info first.tf terraform.tfstate terraform.tfstate.backup varr.tf

first.tf varr.tf Release Notes: 1.47.2

varr.tf

```
1 variable "mytype" []
2   default="t2.micro"
3 }
```

PROBLEMS TERMINAL ... 1: terraform + - ×

An input variable with the name "mytype" has not been declared. This variable can be declared with a variable "mytype" {} block.

PS C:\Users\user\Desktop\mytera> **terraform apply**
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

File Edit Selection View Go Run Terminal Help

first.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform first.tf terraform.tfstate terraform.tfstate.backup varr.tf

first.tf varr.tf Release Notes: 1.47.2

first.tf

```
14   Name = "instance"
15 }
16 }
17 output "name" [
18   value = aws_instance.instance.availability_zone
19 ]
20 
```

PROBLEMS TERMINAL ... 1: powershell + - ×

Outputs:

```
name = aws_instance.instance.availability_zone
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
```

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

```
name = ap-south-1a
PS C:\Users\user\Desktop\mytera>
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** varr.tf - mytera - Visual Studio Code.
- Explorer:** Shows a tree view of files and folders under "MYTERA". Opened files include first.tf and varr.tf.
- Editor:** The varr.tf file is open, containing the following code:

```
variable "mytype" {
  default="t2.large"
}
```
- Terminal:** The terminal window shows the command "terraform apply" being run, with the output:

```
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
~ update in-place

Terraform will perform the following actions:

# aws_instance.instance will be updated in-place
~ resource "aws_instance" "instance" {
    ami                         = "ami-08706cb5f68222d09"
```
- Bottom Status Bar:** ShowsLn 2, Col 22, Spaces: 4, UTF-8, CRLF, Terraform, and system status (ENG 7:38 PM IN 7/18/2020).

if it was in running state then it will simply apply it wont ask us

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "MYTERA" containing ".terraform", "first.tf", "terraform.tfstate", "terraform.tfstate.backup", and "varr.tf".
- Editor:** The file "varr.tf" is open, displaying the following Terraform code:

```
variable "mytype" {
  default="t2.large"
}
```
- Terminal:** A powershell terminal window is active, showing the output of a Terraform command:

```
Plan: 0 to add, 1 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: no

Apply cancelled.
PS C:\Users\user\Desktop\mytera> terraform apply --var=mytype=t2.micro
```
- Status Bar:** Shows "Ln 2, Col 22", "Spaces: 4", "UTF-8", "CRLF", "Terraform", and a file icon with the number "42".

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a tree view with a node for "MYTERA" expanded, containing ".terraform", "first.tf", "{} terraform.tfstate", and "terraformer.tfstate.backup". The file "varr.tf" is currently selected.
- Terminal:** The title bar says "varr.tf - mytera - Visual Studio Code". The terminal window displays the following output:

```
Enter a value: no

Apply cancelled.
PS C:\Users\user\Desktop\mytera> terraform apply --var=mytype=t2.micro
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

name = ap-south-1a
PS C:\Users\user\Desktop\mytera>
```
- Status Bar:** Shows "Ln 2, Col 22" and "Spaces: 4" in the center, and "UTF-8", "CRLF", "Terraform", and icons for "Copy", "Paste", and "Find" on the right.

File Edit Selection View Go Run Terminal Help main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform .terraform.tfstate.lock.info main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
tags = {  
    Name = "web"  
}  
output "webAZ" {  
    value = aws_instance.web.availability_zone  
}
```

PROBLEMS TERMINAL ... 1: terraform + - ×

```
PS C:\Users\user\Desktop\mytera> terraform plan  
Refreshing Terraform state in-memory prior to plan...  
The refreshed state will be used to calculate this plan, but will not be  
persisted to local or remote state storage.  
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
```

File Edit Selection View Go Run Terminal Help main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform .myaws main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
tags = {  
    Name = "web"  
}  
output "webAZ" {  
    value = aws_instance.web.availability_zone  
}
```

PROBLEMS TERMINAL ... 1: powershell + - ×

```
Plan: 1 to add, 0 to change, 1 to destroy.  
-----  
Note: You didn't specify an "-out" parameter to save this plan, so Terraform  
can't guarantee that exactly these actions will be performed if  
"terraform apply" is subsequently run.
```

```
PS C:\Users\user\Desktop\mytera>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.tf - mytera - Visual Studio Code.
- Explorer:** Shows the directory structure under 'MYTERA': .terraform, myaws, main.tf, terraform.tfstate, terraform.tfstate.backup, var.tf.
- Editor:** Two tabs are open: main.tf and var.tf. The main.tf code is as follows:

```
tags = {
  Name = "web"
}
output "webAZ" {
  value = aws_instance.web.availability_zone
}
```

- Terminal:** The terminal shows a history of commands run in PowerShell:

```
d----- 7/18/2020 7:01 PM .terraform
d----- 7/18/2020 7:44 PM myaws
-a---- 7/18/2020 7:43 PM 364 main.tf
-a---- 7/18/2020 7:40 PM 3249 terraform.tfstate
-a---- 7/18/2020 7:40 PM 3305 terraform.tfstate.backup
-a---- 7/18/2020 7:42 PM 46 var.tf
```

- Status Bar:** Ln 18, Col 29, Spaces: 2, UTF-8, CRLF, Terraform.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** ec2.tf - mytera - Visual Studio Code.
- Explorer:** Shows the directory structure under 'MYTERA': .terraform, myaws, ec2.tf, main.tf, terraform.tfstate, terraform.tfstate.backup, var.tf.
- Editor:** Three tabs are open: main.tf, ec2.tf, and var.tf. The ec2.tf code is as follows:

```
1
```

- Terminal:** The terminal shows a history of commands run in PowerShell:

```
d----- 7/18/2020 7:01 PM .terraform
d----- 7/18/2020 7:44 PM myaws
-a---- 7/18/2020 7:43 PM 364 main.tf
-a---- 7/18/2020 7:40 PM 3249 terraform.tfstate
-a---- 7/18/2020 7:40 PM 3305 terraform.tfstate.backup
-a---- 7/18/2020 7:42 PM 46 var.tf
```

- Status Bar:** PS C:\Users\user\Desktop\mytera> cd myaw[Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Terraform.

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists a project structure under 'MYTERA'. The 'myaws' folder contains 'ec2.tf', 'main.tf', 'terraform.tfstate', 'terraform.tfstate.backup', and 'var.tf'. The 'OPEN EDITORS' section shows three files: 'main.tf', 'ec2.tf', and 'var.tf'. The 'TERMINAL' tab is active, displaying the command 'ps C:\Users\user\Desktop\mytera\myaws>'. The terminal output shows the directory path and a table of file details:

Mode	LastWriteTime	Length	Name
-a----	7/18/2020 7:45 PM	0	ec2.tf

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists a project structure under 'MYTERA'. The 'myaws' folder contains 'ec2.tf', 'main.tf', 'terraform.tfstate', 'terraform.tfstate.backup', and 'var.tf'. The 'OPEN EDITORS' section shows three files: 'main.tf', 'ec2.tf', and 'var.tf'. The 'ec2.tf' file is open in the editor, displaying Terraform configuration code:

```
resource "aws_instance" "web" [
  ami           = "ami-08706cb5f68222d09"
  instance_type = var.mytype
  availability_zone = "ap-south-1a"
  key_name      = "mykey1111.pem"

  tags = {
    Name = "web"
  }
]
```

The 'TERMINAL' tab is active, displaying the command 'ps C:\Users\user\Desktop\mytera\myaws>'. The terminal output shows the directory path and a table of file details:

Mode	LastWriteTime	Length	Name
-a----	7/18/2020 7:45 PM	0	ec2.tf

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED

MYTERA .terraform myaws ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
4 }
5
6
7 output "webAZ" {
8   value = aws_instance.web.availability_zone
9 }
10
```

PROBLEMS TERMINAL ...

1: powershell

-a---- 7/18/2020 7:45 PM 0 ec2.tf

PS C:\Users\user\Desktop\mytera\myaws>

Ln 6, Col 1 Spaces: 2 UTF-8 CRLF Terraform

Type here to search

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED

MYTERA .terraform myaws ec2.tf .terraform.tfstate.lock.info main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
4 }
5
6
7 output "webAZ" {
8   value = aws_instance.web.availability_zone
9 }
10
```

PROBLEMS TERMINAL ...

1: terraform

PS C:\Users\user\Desktop\mytera\myaws> cd ..
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]

Ln 6, Col 1 Spaces: 2 UTF-8 CRLF Terraform

Type here to search

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS 1 UNSAVED

MYTERA .terraform myaws ec2.tf .terraform.tfstate.lock.info main.tf terraform.tfstate terraform.tfstate.backup var.tf

```
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
4 }
5
6
7 output "webAZ" {
8   value = aws_instance.web.availability_zone
9 }
10
```

PROBLEMS TERMINAL ... 1: terraform + - ×

```
PS C:\Users\user\Desktop\mytera\myaws> cd ..
PS C:\Users\user\Desktop\mytera> terraform apply
aws_instance.instance: Refreshing state... [id=i-0b27d0564a288f85a]
```

File Edit Selection View Go Run Terminal Help

ec2.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS

MYTERA .terraform myaws ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

```
myaws > ec2.tf
1 resource "aws_instance" "web" [
2   ami           = "ami-08706cb5f68222d09"
3   instance_type = var.mytype
4   availability_zone = "ap-south-1a"
5   key_name      = "mykey1111.pem"
6
7
8   tags = {
9     Name = "web"
10 }
```

PROBLEMS TERMINAL ... 1: powershell + - ×

```
aws_instance.web: Creation complete after 24s [id=i-02a81b2c44af36fe4]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

Outputs:

webAZ = ap-south-1a
PS C:\Users\user\Desktop\mytera>
```

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with 'Instances' selected. Under 'Instances', 'Instances' is also selected, along with 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', and 'Dedicated Hosts'. The main area displays a table of instances. One instance is highlighted: 'web' (i-02a81b2c44af36fe4). Below the table, a detailed view of this instance is shown, including its Instance ID, Public DNS (ec2-13-232-27-36.ap-south-1.compute.amazonaws.com), and other details like Public DNS (IPv4) and Instance Type (t2.micro).

The screenshot shows Visual Studio Code with a dark theme. The Explorer sidebar on the left shows a project structure under 'MYTERA': .terraform\plugins\windows_amd64\lock.json, myaws, ec2.tf, main.tf, terraform.tfstate, and terraform.tfstate.backup. The main editor window shows a 'lock.json' file with the following content:

```
1: {  
2:   "aws": "ba6700e2e5d278f1605026633639f372764677cbf4759ccb376b24e2"  
3: }
```

The terminal at the bottom shows the output of a Terraform run:

```
aws_instance.web: Creation complete after 24s [id=i-02a81b2c44af36fe4]  
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.  
Outputs:  
webAZ = ap-south-1a  
PS C:\Users\user\Desktop\mytera>
```

File Edit Selection View Go Run Terminal Help

myaws > ec2.tf

```
1 resource "aws_instance" "web" [
2   ami           = "ami-08706cb5f68222d09"
3   instance_type = var.mytype
4   availability_zone = "ap-south-1a"
5   key_name      = "mykey1111.pem"
6 ]
```

PROBLEMS TERMINAL ...

Outputs:

```
webAZ = ap-south-1a
PS C:\Users\user\Desktop\mytera> terraform plan
```

Error: Reference to undeclared resource

```
on main.tf line 8, in output "webAZ":
  8:   value = aws_instance.web.availability_zone
```

A managed resource "aws_instance" "web" has not been declared in the root module.

```
PS C:\Users\user\Desktop\mytera>
```

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

main.tf

```
provider "aws" {
  region = "ap-south-1"
  profile="myvishesh"
}
output "webAZ" {
  value = aws_instance.web.availability_zone
}
module "mytestmodule" {
  source = "myaws"
```

PROBLEMS TERMINAL ...

Mode	LastWriteTime	Length	Name
-----	-----	-----	-----
d-----	7/18/2020 7:01 PM		.terraform
d-----	7/18/2020 7:45 PM		myaws
-a----	7/18/2020 7:48 PM	144	main.tf
-a----	7/18/2020 7:48 PM	3300	terraform.tfstate
-a----	7/18/2020 7:48 PM	3249	terraform.tfstate.backup
-a----	7/18/2020 7:42 PM	46	var.tf

Ln 9, Col 18 Spaces: 2 UTF-8 CRLF Terraform

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.tf - mytera - Visual Studio Code.
- Left Sidebar (EXPLORER):**
 - OPEN EDITORS
 - MYTERA
 - .terraform\plugins\windows_amd64s
 - lock.json
 - terraformer-provider-aws_v2.70.0_x4....
 - myaws
 - ec2.tf
 - main.tf
 - terrafrom.tfstate
 - terrafrom.tfstate.backup
 - var.tf
- Central Area (main.tf Editor):**

```
provider "aws" {
  region = "ap-south-1"
  profile="myvishesh"
}
output "webAZ" {
  value = aws_instance.web.availability_zone
}
module "mytestmodule"
```
- Bottom Status Bar:** Line 11, Col 2, Spaces: 2, UTF-8, CRLF, Terraform, IN, ENG, 7:54 PM, 42.

In the terminal pane, the output is:

```
PS C:\Users\user\Desktop\mytera> terraform apply

Error: Module not installed

on main.tf line 8:
  8: module "mytestmodule" {
```

A tooltip at the bottom states: This module is not yet installed. Run "terraform init" to install all modules required by this configuration.

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left lists files and folders related to a project named "mytera". The "main.tf" file is currently selected. The main workspace shows three tabs: "main.tf", "ec2.tf", and "var.tf". The "main.tf" tab displays the following Terraform code:

```
provider "aws" {
  region = "ap-south-1"
  profile="myvishesh"
```

The "TERMINAL" tab shows the output of the "terraform init" command:

```
PS C:\Users\user\Desktop\mytera> terraform init
Initializing modules...
Downloading myaws for mytestmodule...

Error: Module not found

The module address "myaws" could not be resolved.

If you intended this as a path relative to the current module, use "./myaws"
instead. The "./" prefix indicates that the address is a relative filesystem
path.
```

Below this terminal output, there are two more identical error messages from the "main.tf" file:

```
Error: Module not found

The module address "myaws" could not be resolved.
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.tf - mytera - Visual Studio Code.
- Explorer:** Shows the project structure under "MYTERA": .terraform, myaws (with ec2.tf, main.tf, and var.tf), and terraform.tfstate files.
- Terminal:** Running "terraform init".

```
PS C:\Users\user\Desktop\mytera> terraform init
Initializing modules...
- mytestmodule in myaws

Initializing the backend...

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.70"
```
- Code Editor:** Displays the Terraform configuration file (main.tf).

```
provider "aws" {
    region = "ap-south-1"
    profile="myvishesh"
}

output "webAZ" {
    value = aws_instance.web.availability_zone
}

module "mytestmodule" {
    source = "./myaws"
}
```
- Bottom Status Bar:** Shows the current file (main.tf), line 9, column 15, spaces 2, encoding UTF-8, line endings CRLF, and the Terraform extension icon.

File Edit Selection View Go Run Terminal Help main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform myaws ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
```

PROBLEMS TERMINAL ... 1: powershell + - x

commands will detect it and remind you to do so if necessary.
PS C:\Users\user\Desktop\mytera> **terraform apply**

Error: Reference to undeclared resource

on main.tf line 6, in output "webAZ":
6: value = aws_instance.web.availability_zone

A managed resource "aws_instance" "web" has not been declared in the root module.

Error: Reference to undeclared input variable

on myaws\ec2.tf line 3, in resource "aws_instance" "web":
3: instance_type = var.mytype

Ln 11, Col 2 (192 selected) Spaces: 2 UTF-8 CRLF Terraform

Type here to search

0 △ 0

File Edit Selection View Go Run Terminal Help ec2.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform myaws > ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

myaws > ec2.tf

```
1 resource "aws_instance" "web" {
2   ami           = "ami-08706cb5f68222d09"
3   instance_type = "t2.micro"
4   #instance_type = var.mytype
5   availability_zone = "ap-south-1a"
6   key_name      = "mykey1111.pem"
```

PROBLEMS TERMINAL ... 1: powershell + - x

An input variable with the name "mytype" has not been declared. This variable can be declared with a variable "mytype" {} block.

PS C:\Users\user\Desktop\mytera> **terraform plan**
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be persisted to local or remote state storage.

aws_instance.web: Refreshing state... [id=i-02a81b2c44af36fe4]

Ln 4, Col 4 Spaces: 4 UTF-8 CRLF Terraform

Type here to search

0 △ 0

module files has no capacity to call the variable file

File Edit Selection View Go Run Terminal Help main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform myaws ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

```
main.tf
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
4 }
5 /*|output "webAZ" {
6   value = aws_instance.web.availability_zone
7 } */
8 module "mytestmodule" {
9   source = "./myaws"
10 }
```

PROBLEMS TERMINAL ... 1: powershell + - ×

Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be

Ln 5, Col 4 Spaces: 2 UTF-8 CRLF Terraform

Type here to search

File Edit Selection View Go Run Terminal Help ec2.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS MYTERA .terraform myaws > ec2.tf main.tf terraform.tfstate terraform.tfstate.backup var.tf

```
myaws > ec2.tf
1 variable "mytype1" {
2   default="t2.micro"
3 }
4 resource "aws_instance" "web" [
5   ami           = "ami-08706cb5f68222d09"
6   #instance_type = "t2.micro"
7   instance_type = var.mytype1
8   availability_zone = "ap-south-1a"
9   key_name      = "mykey1111.pem"
10
11
12   tags = {
13     Name = "web"
14   }
15 }
```

PROBLEMS TERMINAL ... 1: powershell + - ×

Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be

Ln 7, Col 30 Spaces: 4 UTF-8 CRLF Terraform

Type here to search

File Edit Selection View Go Run Terminal Help

myaws > ec2.tf

```
1 variable "mytype1" {
2   default="t2.micro"
3 }
4 resource "aws_instance" "web" [
5   ami           = "ami-08706cb5f68222d09"
6   #instance_type = "t2.micro"
7   instance_type = var.mytype1
```

PROBLEMS TERMINAL ... 1:terraform + - ×

```
+ volume_size          = (known after apply)
+ volume_type           = (known after apply)
```

}

Plan: 1 to add, 0 to change, 1 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: z

Ln 7, Col 30 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

File Edit Selection View Go Run Terminal Help

myaws > ec2.tf

```
1 variable "myec2type" []
2 resource "aws_instance" "web" {
3   ami           = "ami-08706cb5f68222d09"
4   #instance_type = "t2.micro"
5   instance_type = var.myec2type
6   availability_zone = "ap-south-1a"
```

PROBLEMS TERMINAL ... 1:terraform + - ×

```
on main.tf line 11, in module "mytestmodule":
11: myec2type="t2.micro"
An argument named "myec2type" is not expected here.
```

PS C:\Users\user\Desktop\mytera> terraform apply
var.myec2type
Enter a value:

aws_instance.web: Refreshing state... [id=i-02a81b2c44af36fe4]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
- destroy

Ln 1, Col 24 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

File Edit Selection View Go Run Terminal Help

myaws > ec2.tf

```
1 variable "myec2type" []
2 resource "aws_instance" "web" {
3   ami           = "ami-08706cb5f68222d09"
4   #instance_type = "t2.micro"
5   instance_type = var.myec2type
6   availability_zone = "ap-south-1a"
```

PROBLEMS TERMINAL ... 1:terraform + - ×

```
on main.tf line 11, in module "mytestmodule":
11: myec2type="t2.micro"
An argument named "myec2type" is not expected here.
```

PS C:\Users\user\Desktop\mytera> terraform apply
var.myec2type
Enter a value:

aws_instance.web: Refreshing state... [id=i-02a81b2c44af36fe4]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
- destroy

Ln 1, Col 24 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** main.tf - mytera - Visual Studio Code.
- Left Sidebar (Explorer):**
 - OPEN EDITORS: 1 UNSAVED
 - MYTERA folder:
 - .terraform
 - myaws
 - ec2.tf
 - .terraform.tfstate.lock.info
 - main.tf (selected)
 - terraformer.tfstate
 - terraformer.tfstate.backup
 - var.tf
- Central Area:** Code editor showing Terraform configuration in main.tf.

```
provider "aws" {  
  region = "ap-south-1"  
  profile="myvishesh"  
}  
  
/* output "webAZ" {  
  value = aws_instance.web.availability_zone  
} */  
module "mytestmodule" [  
  source = "./myaws"  
  myec2type="var.mytype"  
]
```
- Bottom Panel:**
 - PROBLEMS: 1: terraform
 - TERMINAL
 - ... (dropdown menu)
 - Issues listed in the Problems panel:
 - ipv6_address_count = 0 -> null
 - ipv6_addresses = [] -> null
 - key_name = "mykey1111.pem" -> null
- Bottom Status Bar:** Ln 11, Col 24, Spaces: 2, UTF-8, CRLF, Terraform, search bar, system icons.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "MYTERA". The "myaws" folder contains ".terraform", ".terraform.lock.info", "main.tf", "ec2.tf", and "var.tf".
- Code Editor (Center):** The "ec2.tf" file is open, containing Terraform configuration for an AWS instance. The code defines a resource "aws_instance" named "web" with attributes like "ami", "instance_type" (set to "t2.micro"), "availability_zone" ("ap-south-1a"), "key_name" ("mykey1111.pem"), and "tags" (with "Name" set to "web").
- Terminal (Bottom):** The terminal tab is active, showing the command "1: terraform". Below it, there are error messages for variables: "ipv6_address_count" and "ipv6_addresses" are set to null, and "key_name" is set to "mykey1111.pem".
- Status Bar (Bottom):** Shows the current file is "ec2.tf - mytera - Visual Studio Code". It also displays the line and column numbers (Ln 2, Col 1), spaces used (Spaces: 4), and encoding (UTF-8). Other tabs include "CRLF", "Terraform", and icons for "Find", "Replace", and "Search".

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** var.tf - mytera - Visual Studio Code.
- Explorer:** Shows a tree view of files and folders under the "MYTERA" workspace, including ".terraform", "myaws", "ec2.tf", ".terraform.tfstate.lock.info", "main.tf", "terraform.tfstate", "terraform.tfstate.backup", and "var.tf".
- Editor:** The "var.tf" file is open, containing the following code:

```
1 variable "mytype" {
2   default="t2.micro"
3 }
```
- Terminal:** The terminal tab is active, showing the command "1: terraform". Below it, there are error messages:
 - ipv6_address_count = 0 -> null
 - ipv6_addresses = [] -> null
 - key_name = "mykey1111.pem" -> null
- Status Bar:** Ln 5, Col 1, Spaces: 4, UTF-8, CRLF, Terraform.

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** var.tf - mytera - Visual Studio Code.
- Explorer:** Shows a tree view of files and folders under the "MYTERA" workspace, including ".terraform", "myaws", "ec2.tf", ".terraform.tfstate.lock.info", "main.tf", "terraform.tfstate", "terraform.tfstate.backup", and "var.tf".
- Editor:** The "var.tf" file is open, containing the same code as the first screenshot.
- Terminal:** The terminal tab is active, showing the command "1: terraform". Below it, there is a Terraform apply session:

```
on main.tf line 11, in module "mytestmodule":  
11:   myec2type="t2.micro"  
An argument named "myec2type" is not expected here.  
  
PS C:\Users\user\Desktop\mytera> terraform apply  
var.myec2type  
Enter a value:  
  
aws_instance.web: Refreshing state... [id=i-02a81b2c44af36fe4]
```
- Status Bar:** Ln 5, Col 1, Spaces: 4, UTF-8, CRLF, Terraform.

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS

MYTERA .terraform myaws ec2.tf .terraform.tfstate.lock.info main.tf terraform.tfstate terraform.tfstate.backup var.tf

main.tf

```
1 provider "aws" {
2   region = "ap-south-1"
3   profile="myvishesh"
4 }
5
6 /* output "webAZ" {
7   value = aws_instance.web.availability_zone
8 }
```

PROBLEMS TERMINAL ...

1: terraform

```
+ host_id = (known after apply)
+ id = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "var.mytype"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "mykey1111.pem"
+ network_interface_id = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ primary_network_interface_id = (known after apply)
```

Ln 11, Col 24 Spaces: 2 UTF-8 CRLF Terraform

0 △ 0 Type here to search

File Edit Selection View Go Run Terminal Help

var.tf - mytera - Visual Studio Code

EXPLORER OPEN EDITORS

MYTERA .terraform myaws ec2.tf .terraform.tfstate.lock.info main.tf terraform.tfstate terraform.tfstate.backup var.tf

var.tf

```
1 variable "mytype" {
2   default="t2.small"
3 }
```

PROBLEMS TERMINAL ...

1: terraform

```
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = "mykey1111.pem"
+ network_interface_id = (known after apply)
+ outpost_arn = (known after apply)
+ password_data = (known after apply)
+ placement_group = (known after apply)
+ primary_network_interface_id = (known after apply)
```

Ln 4, Col 1 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

File Edit Selection View Go Run Terminal Help

myaws > ec2.tf

```
1 variable "myec2type" {}  
2  
3 resource "aws_instance" "web" []  
4   ami           = "ami-08706cb5f68222d09"  
5   #instance_type = "t2.micro"  
6   instance_type = var.myec2type  
7   availability_zone = "ap-south-1a"  
8   key_name      = "mykey1111.pem"  
9  
10  tags = {  
11    Name = "web"  
12  }  
13  
14 }
```

PROBLEMS TERMINAL ...

+ password_data = (known after apply)
+ placement_group = (known after apply)
+ primary_network_interface_id = (known after apply)

Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Terraform

0 △ 0 Type here to search

File Edit Selection View Go Run Terminal Help

main.tf - mytera - Visual Studio Code

main.tf

```
1 provider "aws" {  
2   region = "ap-south-1"  
3   profile="myvishesh"  
4 }  
5  
6 /* output "webAZ" {  
7   value = aws_instance.web.availability_zone  
8 } */  
9 module "mytestmodule" {  
10   source = "./myaws"  
11   myec2type="var.mytype"  
12 }  
13 }
```

PROBLEMS TERMINAL ...

+ password_data = (known after apply)
+ placement_group = (known after apply)
+ primary_network_interface_id = (known after apply)

Ln 11, Col 24 Spaces: 2 UTF-8 CRLF Terraform

0 △ 0 Type here to search

A screenshot of a web browser window showing search results for "terraform registry" on Google. The results page includes links for Terraform Registry, Terraform AWS modules, Azure, Browse, Private Registry, Docs, and HashiCorp.

Google search results for "terraform registry":

- Terraform Registry**
Discover Terraform providers that power all of Terraform's resource types, or find modules for quickly deploying common infrastructure ...
- Terraform AWS modules**
Collection of Terraform AWS modules supported by the ...
- Azure**
Microsoft Azure. APIs, SDKs and open source projects from ...
- Browse**
Modules are self-contained packages of Terraform ...
- Private Registry**
Terraform can load private modules from private registries ...
- Docs**
The Terraform Registry is a ...
- HashiCorp**
HashiCorp. Consistent workflows

A screenshot of a web browser window showing the sign-in page for the Terraform Registry using GitHub authentication. The URL is <https://registry.terraform.io/github/auth/start>. The page features a "Sign in with GitHub" button.

Sign in to the Terraform Registry with GitHub

Sign in with GitHub

INTRO LEARN DOCS EXTEND COMMUNITY STATUS PRIVACY SECURITY TERMS PRESS KIT

© HashiCorp 2020

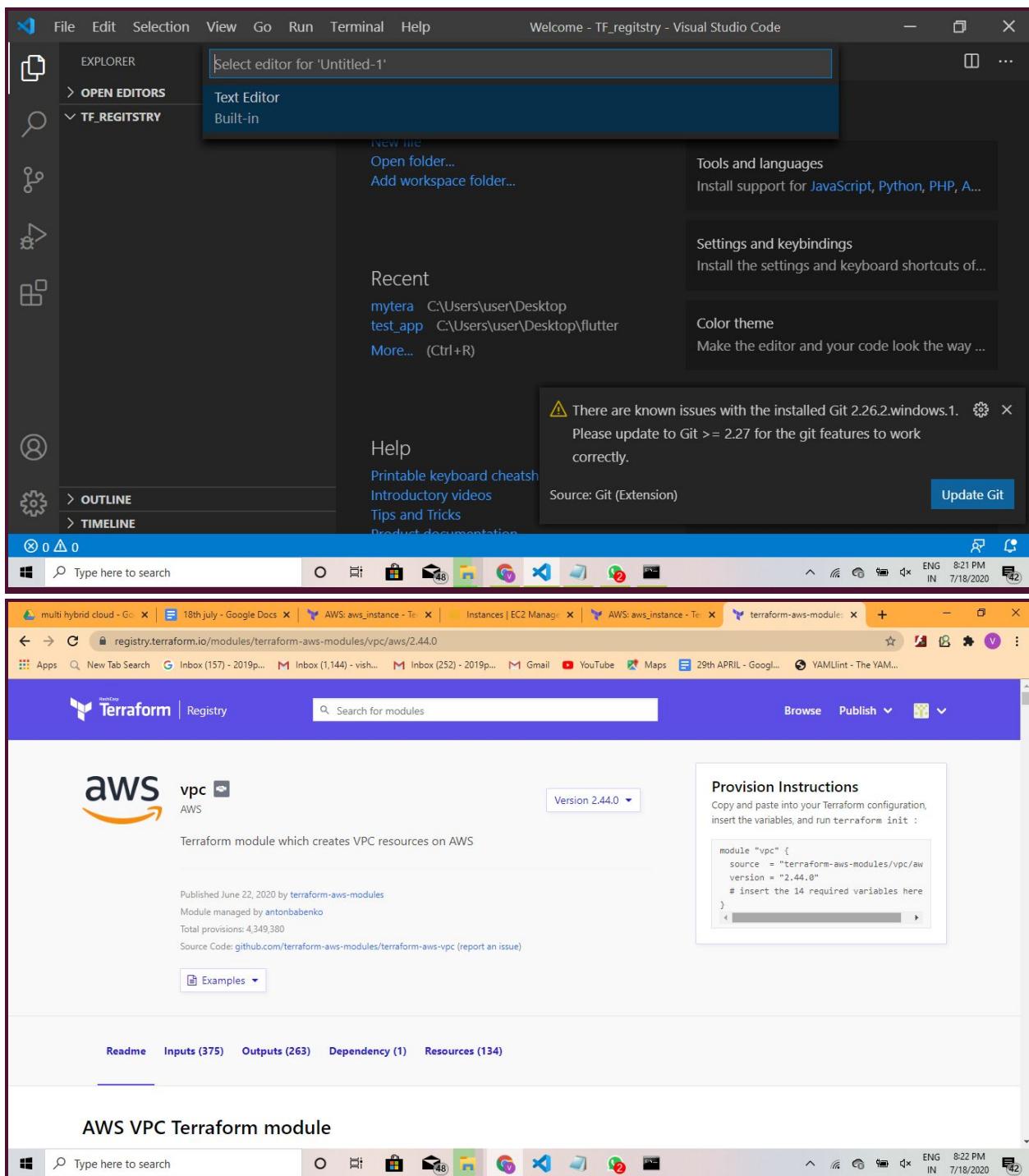
The screenshot shows a Microsoft Edge browser window with the address bar displaying `registry.terraform.io/providers/hashicorp/aws/latest`. The page lists four AWS modules from the HashiCorp provider:

- terraform-aws-modules / security-group**: A Terraform module which creates EC2-VPC security groups on AWS. Last updated 15 days ago, with 4,787,403 downloads. It is an **aws provider**.
- terraform-aws-modules / vpc**: A Terraform module which creates VPC resources on AWS. Last updated a month ago, with 4,349,380 downloads. It is an **aws provider**.
- terraform-aws-modules / rds**: A Terraform module which creates RDS resources on AWS. Last updated a month ago, with 1,561,938 downloads. It is an **aws provider**.

The browser's taskbar at the bottom shows the URL `https://registry.terraform.io/modules/terraform-aws-modules/vpc/aws`. The system tray indicates the date as 7/18/2020 and the time as 8:19 PM.

if we are using source then it is from other registry

If we are using . it means it is from main



A screenshot of Visual Studio Code interface. The title bar reads "main.tf - TF_registry - Visual Studio Code". The left sidebar shows the "EXPLORER" view with "OPEN EDITORS" containing "main.tf" and "TF_REGISTRY" also containing "main.tf". The main editor area displays the following Terraform code:

```
1 module "vpc" {
2   source  = "terraform-aws-modules/vpc/aws"
3   version = "2.44.0"
4   # insert the 14 required variables here
5 }
```

The status bar at the bottom shows "Ln 5, Col 2" and "Spaces: 4" and "Terraform" is listed in the status bar.

AWS

Terraform module which creates VPC resources on AWS

Published June 22, 2020 by [terraform-aws-modules](#)

Module managed by [antonbabenko](#)

Total provisions: 4,349,380

Source Code: [github.com/terraform-aws-modules/terraform-aws-vpc](#) (report an issue)

Version 2.44.0

Copy and paste into your Terraform configuration, insert the variables, and run `terraform init`:

```
module "vpc" {  
  source = "terraform-aws-modules/vpc/aws"  
  version = "2.44.0"  
  # insert the 14 required variables here  
}
```

Examples ▾

Readme Inputs (375) Outputs (263) Dependency (1) Resources (134)

Required Inputs

These variables must be set in the `module` block when using this module.

These variables must be set in the `module` block when using this module.

`database_subnet_assign_ipv6_address_on_creation` `bool`

Description: Assign IPv6 address on database subnet, must be disabled to change IPv6 CIDRs. This is the IPv6 equivalent of `map_public_ip_on_launch`

`default_security_group_egress` `list(map(string))`

Description: List of maps of egress rules to set on the default security group

`default_security_group_ingress` `list(map(string))`

Description: List of maps of ingress rules to set on the default security group

`elasticache_subnet_assign_ipv6_address_on_creation` `bool`

Description: Assign IPv6 address on elasticache subnet, must be disabled to change IPv6 CIDRs. This is the IPv6 equivalent of `map_public_ip_on_launch`