

Terraform Task

Task Description:

Launch Linux EC2 instances in two regions using a single Terraform file.

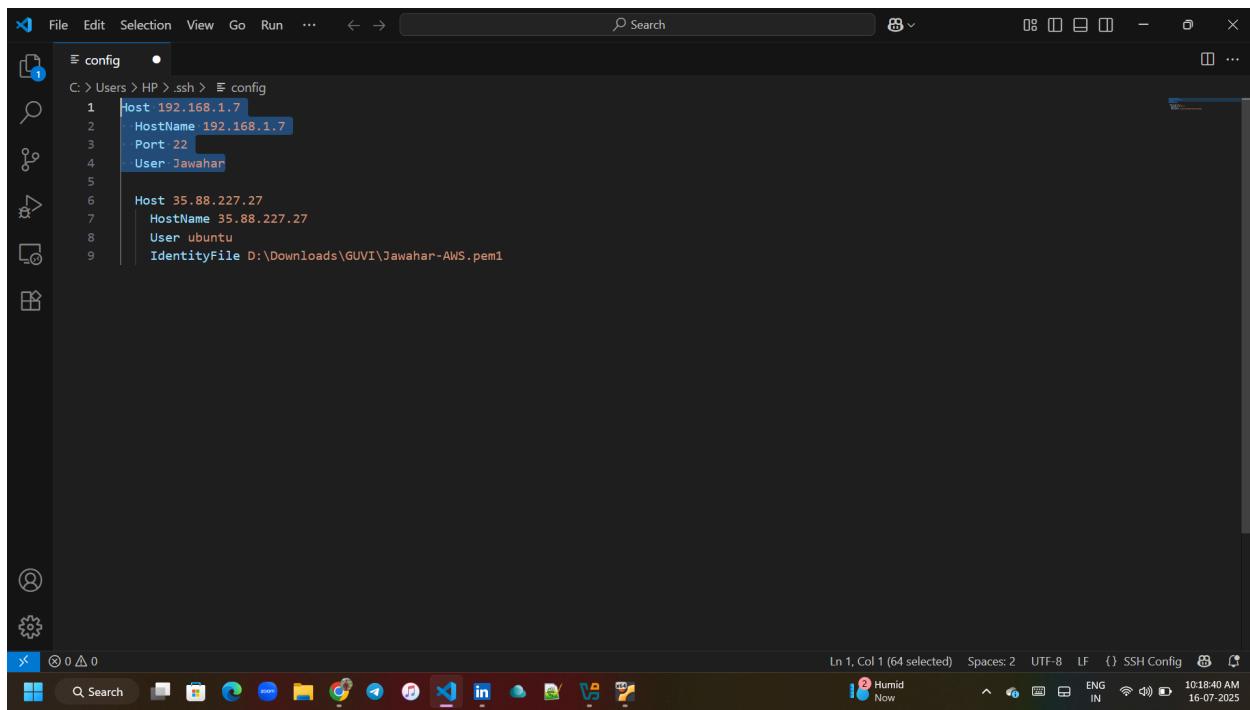
Techstacks needs to be used :

- AWS EC2
- Terraform
- AWS CLI

Screenshots:

Google Drive :

https://docs.google.com/document/d/1nhfwWnFJ2lPeNkzzYyLz9Po7ZdeF67_JBrJfGBw5SJc/edit?usp=sharing



```
C:\> Users > HP > .ssh > config
1 Host 192.168.1.7
2   HostName 192.168.1.7
3   Port 22
4   User Jawahar
5
6 Host 35.88.227.27
7   HostName 35.88.227.27
8   User ubuntu
9   IdentityFile D:\Downloads\GUVI\Jawahar-AWS.pem1
```

The screenshot shows a terminal window with a dark theme. The current directory is C:\> Users > HP > .ssh >. The user is viewing the 'config' file. The content of the file is displayed in the terminal window, showing two host entries. The first host entry is for an IP address 192.168.1.7, specifying HostName, Port (22), User (Jawahar), and an IdentityFile path. The second host entry is for an IP address 35.88.227.27, specifying HostName, User (ubuntu), and an IdentityFile path. The terminal window also shows various icons in the toolbar and a status bar at the bottom.

```
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.25
5
        ether 02:42:85:f0:57:6f txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.7 netmask 255.255.255.0 broadcast 192.168.1.
255
        inet6 fe80::a00:27ff:fe0c:7854 prefixlen 64 scopeid 0x20<link>
nk>
        ether 08:00:27:c5:78:54 txqueuelen 1000 (Ethernet)
        RX packets 62652 bytes 86701611 (86.7 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 7802 bytes 3986897 (3.9 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 2066 bytes 4139616 (4.1 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2066 bytes 4139616 (4.1 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Jawahar@Jawahar:~\$

Jawahar [SSH: 192.168.1.7]

```
$ Install_Terraform.sh
1 #!/bin/bash
2
3 echo "Updating system and installing prerequisite"
4 sudo apt-get update && sudo apt-get install -y gn
5
6 echo "Adding Hashicorp GPG key..."
7 wget -O https://apt.releases.hashicorp.com/gpg | gpg --dearmor | \
8 sudo tee /usr/share/keyrings/hashicorp-archive-
9
10 echo "Verifying GPG Fingerprint..."
11 gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-
12 --fingerprint
13
14 echo "Adding Hashicorp APT repository..."
15 echo "deb [arch=$(dpkg --print-architecture) sign
https://apt.releases.hashicorp.com $grep -o '^
| sudo tee /etc/apt/sources.list.d/hashicorp.li
16
17 echo "Updating APT package index..."
18 sudo apt-get update
19
20 echo "Installing Terraform..."
21 sudo apt-get install -y terraform
22
23 echo "Verifying Terraform installation..."
24 terraform version
25
26 echo "* Terraform installed successfully via API"
27
28
29
30
31
32
```

Welcome to Copilot

Let's get started

Ask Copilot

Build workspace

Show project config

Review AI output carefully before use.

10:30:52 AM 16-07-2025

Jawahar [SSH: 192.168.1.7]

```
inflating: aws/dist/awscli/potocore/data/bin/2021-06-15/endpoint-rule-set-1.json
inflating: aws/dist/awscli/potocore/data/bin/2021-06-15/paginators-1.json
creating: aws/dist/awscli/potocore/data/timestream-write/2018-11-01/
inflating: aws/dist/awscli/potocore/data/timestream-write/2018-11-01/endpoint-rule-set-1.json
inflating: aws/dist/awscli/potocore/data/timestream-write/2018-11-01/paginators-1.json
inflating: aws/dist/awscli/potocore/data/timestream-write/2018-11-01/service-2.json
creating: aws/dist/awscli/potocore/data/marketplace-deployment/2023-01-25/
inflating: aws/dist/awscli/potocore/data/marketplace-deployment/2023-01-25/service-2.json
inflating: aws/dist/awscli/potocore/data/marketplace-deployment/2023-01-25/endpoint-rule-set-1.json
inflating: aws/dist/awscli/potocore/data/marketplace-deployment/2023-01-25/paginators-1.json
creating: aws/dist/awscli/potocore/_changes/next-release/
inflating: aws/dist/awscli/potocore/_changes/next-release/api-change-connect-59117.json
inflating: aws/dist/awscli/topics/dbb-expressions.rst
inflating: aws/dist/awscli/topics/s3-config.rst
inflating: aws/dist/awscli/topics/config-vars.rst
inflating: aws/dist/awscli/topics/topic-tags.json
inflating: aws/dist/awscli/topics/return-codes.rst
inflating: aws/dist/awscli/topics/s3-faq.rst
inflating: aws/dist/awscli/data/cli.json
inflating: aws/dist/awscli/data/ac_index
inflating: aws/dist/awscli/data/metadata.json
creating: aws/dist/awscli/customizations/sso/
creating: aws/dist/awscli/customizations/wizard/
creating: aws/dist/awscli/customizations/wizard/wizards/configure/
creating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/
creating: aws/dist/awscli/customizations/wizard/wizards/events/
creating: aws/dist/awscli/customizations/wizard/wizards/iam/
creating: aws/dist/awscli/customizations/wizard/wizards/lambda/
inflating: aws/dist/awscli/customizations/wizard/iam/new-role.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/configure/_main.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/new-table.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/events/new-rule.yml
inflating: aws/dist/awscli/customizations/wizard/wizards/lambda/new-function.yml
inflating: aws/dist/awscli/customizations/sso/index.html
Jawahar@Jawahar:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
Jawahar@Jawahar:~$ aws --version
aws-cli/2.27.52 Python/3.13.4 Linux/6.8.0-62-generic exe/x86_64/ubuntu,24
Jawahar@Jawahar:~$
```

Welcome to Copilot

Let's get started

Ask Copilot

Build workspace

Show project config

Review AI output carefully before use.

10:39:53 AM 16-07-2025

The screenshot shows the AWS IAM Access Keys page. The left sidebar includes sections for Identity and Access Management (IAM), Access management (User groups, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings). The main content area displays the 'Access keys (0)' section, which states: 'Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.' A 'Create access key' button is present. Below it is the 'API keys for Amazon Bedrock (0)' section, which says: 'Use API keys for Amazon Bedrock to integrate into your library of choice and make API requests programmatically. You can have a maximum of two long-term API keys (active, inactive, or expired) at a time.' A 'Generate API Key' button is available. The final section is 'SSH public keys for AWS CodeCommit (0)', which allows users to upload SSH public keys for authentication. The bottom of the page includes standard AWS navigation links and a status bar.

This screenshot shows the 'Create access key' wizard, Step 3: 'Select use case'. The 'Command Line Interface (CLI)' option is selected, with a note: 'You plan to use this access key to enable the AWS CLI to access your AWS account.' Other options include 'Local code', 'Application running on an AWS compute service', 'Third-party service', 'Application running outside AWS', and 'Other'. A 'Alternatives recommended' section suggests using AWS CloudShell. The bottom of the page includes standard AWS navigation links and a status bar.

The screenshot shows the AWS IAM 'Create access key' page. A green banner at the top indicates that an access key has been created successfully. Below this, there's a section titled 'Retrieve access keys' with a sub-section 'Access key'. It displays the Access Key ID (AKIA2OAJTS6CQ524RNVS) and a 'Show' button for the Secret Access Key. To the left, a sidebar lists steps: 'Access key best practices & alternatives', 'Set description tag', and 'Retrieve access keys' (which is selected). Below the main content, there's a 'Access key best practices' section with tips and a link to 'best practices for managing AWS access keys'. At the bottom right are 'Download .csv file' and 'Done' buttons.

The screenshot shows the VS Code interface with a terminal window open. The terminal shows the output of the 'aws configure list' command. It displays the AWS Access Key ID (AKIA2OAJTS6CQ524RNVS), AWS Secret Access Key (redacted), Default region name (us-west-2), and Default output format (None). The terminal also shows the 'Welcome to Copilot' extension with a 'Let's get started' button. The left sidebar shows the file structure of the 'JAWAHAR' directory, which includes '.aws', '.cache', '.gnupg', '.kube', '.local', '.minikube', '.ssh', '.terraform.d', '.vscode-server', 'aws', 'kubernetes', 'snap', '.bash_history', '.bash_logout', '.bashrc', '.profile', '.sudo_as_admin_successful', '.wget-hsts', 'awscli.v2.zip', and 'Install_Terraform.sh'. The bottom status bar shows the connection to 'SSH: 192.168.1.7' and the date/time '10:43:37 AM 16-07-2025'.

```
main.tf
Terraform > main.tf > ...
1 terraform {
2   required_providers {
3     aws = {
4       source  = "hashicorp/aws"
5       version = "> 4.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   alias = "virginia"
12   region = "us-east-1"
13 }
14
15 provider "aws" {
16   alias = "oregon"
17   region = "us-west-2"
18 }
19
20 data "aws_ami" "ubuntu_virginia" {
21   most_recent = true
22   owners = ["099720109477"] # Canonical
23
24   filter {
25     name = "name"
26     values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
27   }
28
29   provider = aws.virginia
30 }
31
32 data "aws_ami" "ubuntu_oregon" {
33   most_recent = true
34   owners = ["099720109477"]
35
36   filter {
37     name = "name"
38     values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
39   }
40
41 provider = aws.oregon
42 }
43
44 resource "aws_instance" "ubuntu_us_east_1" {
45   ami = data.aws.ami.ubuntu.id
46   instance_type = "t2.micro"
47   provider = aws.virginia
48
49   tags = {
50     Name = "Ubuntu-USEast1"
51   }
52 }
53
54 resource "aws_instance" "ubuntu_us_west_2" {
55   ami = data.aws.ami.ubuntu.oregon.id
56   instance_type = "t2.micro"
57   provider = aws.oregon
58
59   tags = {
60     Name = "Ubuntu-USWest2"
61   }
62 }
63
64 output "ec2_public_ips" {
65   value = {
66     "us-east-1" = aws_instance.ubuntu_us_east_1.public_ip
67     "us-west-2" = aws_instance.ubuntu_us_west_2.public_ip
68   }
}
```

```
main.tf
Terraform > main.tf > ...
32 data "aws_ami" "ubuntu_oregon" {
33   owners = ["099720109477"]
34
35   filter {
36     name = "name"
37     values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
38   }
39
40   provider = aws.oregon
41 }
42
43
44 resource "aws_instance" "ubuntu_us_east_1" {
45   ami = data.aws.ami.ubuntu.id
46   instance_type = "t2.micro"
47   provider = aws.virginia
48
49   tags = {
50     Name = "Ubuntu-USEast1"
51   }
52 }
53
54 resource "aws_instance" "ubuntu_us_west_2" {
55   ami = data.aws.ami.ubuntu.oregon.id
56   instance_type = "t2.micro"
57   provider = aws.oregon
58
59   tags = {
60     Name = "Ubuntu-USWest2"
61   }
62 }
63
64 output "ec2_public_ips" {
65   value = {
66     "us-east-1" = aws_instance.ubuntu_us_east_1.public_ip
67     "us-west-2" = aws_instance.ubuntu_us_west_2.public_ip
68   }
}
```

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

- File Explorer:** Shows files in the current workspace, including `main.tf`, `.bash_history`, `.bash_logout`, `.bashrc`, `.profile`, `.sudo_as_admin_successful`, `.wget-hsts`, `awscli2.zip`, and `Install-Terraform.sh`.
- Terminal:** The terminal window shows the command `Jawahar@Jawahar:~/Terraform$ terraform init` being run, followed by the output of the Terraform initialization process.
- Code Editor:** The main editor pane displays the `main.tf` Terraform configuration file.
- Output:** The Output tab shows the successful initialization message: `Terraform has been successfully initialized!`
- Problems:** The Problems tab shows no errors or warnings.
- Terminal:** The Terminal tab shows the command `Jawahar@Jawahar:~/Terraform$`.
- Bottom Status Bar:** Shows the current file is `main.tf`, line 19, column 1, with 19 rows and 4 columns, and the status `Ln 19, Col 1 Spaces: 4 UTF-8 LF () Terraform`.
- Bottom Icons:** Includes icons for search, file operations, and system status.

The screenshot shows the Visual Studio Code (VS Code) interface with the Terraform extension installed. The left sidebar displays the file tree, showing files like `main.tf`, `.bash_history`, and `Install-Terraform.sh`. The main editor area contains `main.tf` with Terraform configuration for AWS instances. The right sidebar features the "Welcome to Copilot" extension, which includes a "Let's get started" button, an "Ask Copilot" input field, a "Build workspace" button, and a "Show project config" button. The status bar at the bottom shows the current session details.

```
provider "aws" {  
    region = "us-east-1"  
}  
  
resource "aws_instance" "ubuntu_us_east_1" {  
    ami = data.aws_ami.ubuntu_virginia.id  
    instance_type = "t2.micro"  
    provider = aws.virginia  
  
    tags = {  
        Name = "Ubuntu-USEast1"  
    }  
  
}  
  
resource "aws_instance" "ubuntu_us_west_2" {  
    ami = data.aws_ami.ubuntu_oregon.id  
    instance_type = "t2.micro"  
    provider = aws.oregon  
  
    tags = {  
        Name = "Ubuntu-USWest2"  
    }  
  
}
```

PROBLEMS OUTPUT TERMINAL ...

tenancy
user_data
user_data_base64
user_data_replace_on_change
vpc_security_group_ids
capacity_reservation_specification
cpu_options
ebs_block_device
enclave_options
ephemeral_block_device
maintenance_options
metadata_options
network_interface
private_dns_name_options
root_block_device

= (known after apply)
= (known after apply)
= (known after apply)
= false
= (known after apply)

Welcome to Copilot
Let's get started

Ask Copilot
Build workspace
Show project config

Review AI output carefully before use.

Jawahar [SSH: 192.168.1.7]

File Edit Selection View ...

PROBLEMS OUTPUT TERMINAL ...

bash - Terraform

Welcome to Copilot

Let's get started

Ask Copilot

@ ↻ ▶

Build workspace

Show project config

Review AI output carefully before use.

EXPLORER JAWAHAR [SSH: 192.168.1.7] main.tf

Terraform > main.tf > resource "aws_instance" "ubuntu_us_east_1" > provider

```
 20  data "aws_amis" "ubuntu_virginia" {
 21  filter {
 22    name = "name"
 23    values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-20230711-amd64"]
 24  }
 25  provider = aws.virginia
 26}
 27}
 28}
 29}
 30}
 31}
 32 data "aws_amis" "ubuntu_oregon" {
 33 most_recent = true
 34 owners = ["099720109477"]
 35}
 36 filter {
 37   name = "name"
 38   values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-20230711-amd64"]
 39 }
 40}
 41 provider = aws.oregon
 42}
 43}
 44 resource "aws_instance" "ubuntu_us_east_1" {
 45   ami = data.aws_amis.ubuntu_virginia.id
 46   instance_type = "t2.micro"
 47   provider = aws.virginia
 48   tags = {
 49     Name = "Ubuntu-USEast1"
 50   }
 51 }
 52}
 53}
 54 resource "aws_instance" "ubuntu_us_west_2" {
 55   ami = data.aws_amis.ubuntu_oregon.id
 56   instance_type = "t2.micro"
 57   provider = aws.oregon
 58   tags = {
 59 }
```

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:

```
+ ec2_public_ips = {
+   us-east-1 = (known after apply)
+   us-west-2 = (known after apply)
}
```

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_instance.ubuntu_us_east_1: Creating...
aws_instance.ubuntu_us_west_2: Creating...
aws_instance.ubuntu_us_east_1: Still creating... [0m01s elapsed]
aws_instance.ubuntu_us_west_2: Still creating... [0m01s elapsed]
aws_instance.ubuntu_us_east_1: Creation complete after 17s [id=i-05dbef7bf491e127]
aws_instance.ubuntu_us_west_2: Still creating... [0m20s elapsed]
aws_instance.ubuntu_us_west_2: Creation complete after 26s [id=i-029c13ef7fb732a1022]
```

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

```
ec2_public_ips = {
  "us-east-1" = "3.94.10.97"
  "us-west-2" = "34.215.58.76"
}
```

Jawahar@Jawahar:~/Terraform\$

The screenshot shows the AWS Management Console with the EC2 service selected. The main view displays a single instance named "Ubuntu-USWe...". The instance is in the "Running" state, type "t2.micro", and status "Initializing". It has a public IPv4 address of 34.215.58.76 and a private IPv4 address of 172.31.21.28. The instance was last updated less than a minute ago.

Instances (1/1) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
Ubuntu-USWe...	i-029c13efb732a1022	Running	t2.micro	Initializing	View alarms +	us-west-2a	ec2-34-215-58-76.us-w...	34.215...

i-029c13efb732a1022 (Ubuntu-USWest2)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID: i-029c13efb732a1022

Public IPv4 address: 34.215.58.76 | open address

Private IPv4 addresses: 172.31.21.28

IPv6 address: -

Instance state: Running

Public DNS name: ec2-34-215-58-76.us-west-2.compute.amazonaws.com | open address

Hostname type: Private IP DNS name (IPv4 only)

The screenshot shows the AWS EC2 Instances page. On the left, a navigation sidebar lists EC2 services: Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays a table titled "Instances (1/1) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One row is shown for "Ubuntu-USEast1" with the instance ID "i-05dbe5f7bf491e127", which is "Running" and "t2.micro", with an "Initializing" status check, in "us-east-1b" availability zone, and public IP "ec2-3-94-10-97.com". Below the table, a detailed view for "Ubuntu-USEast1" is shown with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. Under "Details", sections include Instance summary (with Instance ID "i-05dbe5f7bf491e127"), Public IPv4 address ("172.31.89.164"), Private IPv4 addresses ("172.31.89.164"), and Public DNS ("ec2-3-94-10-97.compute-1.amazonaws.com"). The status bar at the bottom indicates the date and time as "11:07:50 AM 16-07-2025".

The screenshot shows the AWS EC2 Global View page. On the left, a navigation sidebar lists Region explorer (selected) and Global search/Settings. The main content area features a "Region explorer" tab and a "Global search" tab. A summary section states "Fetching resources for all opted in regions" and "Resource update complete". Resource counts are listed in four columns: Enabled regions (17 regions), Instances (2 in 2 regions), VPCs (17 in 17 regions), Subnets (55 in 17 regions); Security groups (25 in 17 regions), Volumes (2 in 2 regions), Auto scaling groups (0 in 0 regions), Route tables (17 in 17 regions); VPC endpoints (0 in 0 regions), NAT gateways (0 in 0 regions), Egress only internet gateways (0 in 0 regions), Internet gateways (17 in 17 regions); DHCP option sets (17 in 17 regions), Elastic IPs (0 in 0 regions), Endpoint services (0 in 0 regions), Managed prefix lists (174 in 17 regions); Network ACLs (17 in 17 regions), Network interfaces (2 in 2 regions), VPC peering connections (0 in 0 regions), Capacity Reservations (0 in 0 regions). The status bar at the bottom indicates the date and time as "11:08:05 AM 16-07-2025".

The screenshot shows the AWS CloudWatch Metrics Insights interface. A search query "aws_lambda_function_invocation" has been entered into the search bar. Below the search bar, there are two tabs: "Metrics" and "Logs". The "Metrics" tab is selected. On the left, there is a sidebar with navigation links: "CloudWatch Metrics", "CloudWatch Metrics Insights", "CloudWatch Metrics Insights Overview", "CloudWatch Metrics Insights Search", "CloudWatch Metrics Insights Metrics", and "CloudWatch Metrics Insights Metrics Overview". The main content area displays a table of metrics results. The columns are "Metric Name", "Value", "Dimensions", and "Time Range". One row is visible, showing a metric named "aws_lambda_function_invocation" with a value of "1" and dimensions "FunctionName: my_lambda_function, Region: us-east-1". The time range is "Last 1 hour".

The screenshot shows the AWS CloudWatch Metrics Insights interface. A search query "aws_lambda_function_invocation" has been entered into the search bar. Below the search bar, there are two tabs: "Metrics" and "Logs". The "Metrics" tab is selected. On the left, there is a sidebar with navigation links: "CloudWatch Metrics", "CloudWatch Metrics Insights", "CloudWatch Metrics Insights Overview", "CloudWatch Metrics Insights Search", "CloudWatch Metrics Insights Metrics", and "CloudWatch Metrics Insights Metrics Overview". The main content area displays a table of metrics results. The columns are "Metric Name", "Value", "Dimensions", and "Time Range". One row is visible, showing a metric named "aws_lambda_function_invocation" with a value of "1" and dimensions "FunctionName: my_lambda_function, Region: us-east-1". The time range is "Last 1 hour".

Jawahar [SSH: 192.168.1.7]

PROBLEMS OUTPUT TERMINAL ...

Welcome to Copilot

Let's get started

Ask Copilot

@ ⌂ ▶

Build workspace

Show project config

Review AI output carefully before use.

EXPLORER JAWAHAR [SSH: 192.168.1.7] ...

main.tf x

Terraform > main.tf > resource "aws_instance" "ubuntu_us_east_1" > provider

```
  20  data "aws ami" "ubuntu_virginia" {
  21  filter {
  22    name = "name"
  23    values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-"]
  24  }
  25  provider = aws.virginia
  26
  27  data "aws ami" "ubuntu_oregon" {
  28    most_recent = true
  29    owners = ["099720109477"]
  30
  31    filter {
  32      name = "name"
  33      values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-"]
  34    }
  35
  36    provider = aws.oregon
  37
  38  resource "aws_instance" "ubuntu_us_east_1" {
  39    ami = data.aws ami.ubuntu_virginia.id
  40    instance_type = "t2.micro"
  41    provider = aws.virginia
  42
  43    tags = {
  44      Name = "Ubuntu-USEast1"
  45    }
  46
  47  resource "aws_instance" "ubuntu_us_west_2" {
  48    ami = data.aws ami.ubuntu_oregon.id
  49    instance_type = "t2.micro"
  50    provider = aws.oregon
  51
  52    tags = {
  53      Name = "Ubuntu-USWest2"
  54    }
  55
  56  }
```

Plan: 0 to add, 0 to change, 2 to destroy.

Changes to Outputs:

- ec2_public_ips = {
 - us-east-1 = "3.94.10.97"
 - us-west-2 = "34.215.58.76"}

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above. There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.ubuntu_us_east_1: Destroying... [id=i-05dbe5f7bf491e127]
aws_instance.ubuntu_us_west_2: Destroying... [id=i-029c13efb732a1022]
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m01s elapsed]
aws_instance.ubuntu_us_west_2: Still destroying... [id=i-029c13efb732a1022, 0m01s elapsed]
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m02s elapsed]
aws_instance.ubuntu_us_west_2: Still destroying... [id=i-029c13efb732a1022, 0m02s elapsed]
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m03s elapsed]
aws_instance.ubuntu_us_west_2: Still destroying... [id=i-029c13efb732a1022, 0m03s elapsed]
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m03s elapsed]
aws_instance.ubuntu_us_west_2: Still destroying... [id=i-029c13efb732a1022, 0m03s elapsed]
aws_instance.ubuntu_us_west_2: Destruction complete after 32s
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m04s elapsed]
aws_instance.ubuntu_us_east_1: Still destroying... [id=i-05dbe5f7bf491e127, 0m05s elapsed]
aws_instance.ubuntu_us_east_1: Destruction complete after 54s

Destroy complete! Resources: 2 destroyed.

Jawahar@Jawahar:~/Terraform\$

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

- File Explorer:** Shows a tree view of files and folders, including `main.tf`, `main.tfstate`, `terraform.state.bac...`, `.bash_history`, `.bash_logout`, `.bashrc`, `.profile`, `.sudo_as_admin_suc...`, `.wget-hists`, `awscliv2.zip`, and `Install_Terraform.sh`.
- Code Editor:** The `main.tf` file is open, containing Terraform configuration for two instances: `ubuntu_us_east_1` and `ubuntu_us_west_2`.
- Terminal:** The terminal shows the output of a `terraform plan` command, indicating no changes to add or destroy, but 2 resources to filter.
- Output:** The output pane shows the results of the `destroy` command, detailing the termination of both instances.
- Problems:** The problems pane shows one warning: "The state file is empty. No resources are represented."
- Terminal:** The terminal shows the command `terraform show` being run.
- Status Bar:** Shows the current session is `Jawahar [SSH: 192.168.1.7]`, with 47 columns, 24 rows, and UTF-8 encoding.

The screenshot shows the AWS EC2 Instances page in the us-east-1 region. The left sidebar is collapsed, showing the main navigation menu. The main content area displays a table of instances. One instance, "Ubuntu-USEast1" (i-05dbe5f7bf491e127), is selected and shown in detail below the table. The instance summary shows it is terminated. The status bar at the bottom indicates the region is United States (N. Virginia) and the user is poweruser01 @ jawahar11.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Ubuntu-USEast1	i-05dbe5f7bf491e127	Terminated	t2.micro	-	-	us-east-1b	-

The screenshot shows the AWS EC2 Instances page in the us-west-2 region. The left sidebar is collapsed. The main content area displays a table of instances. One instance, "Ubuntu-USWest2" (i-029c13efb732a1022), is selected and shown in detail below the table. The instance summary shows it is terminated. The status bar at the bottom indicates the region is United States (Oregon) and the user is poweruser01 @ jawahar11.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Ubuntu-USWest2	i-029c13efb732a1022	Terminated	t2.micro	-	-	us-west-2a	-