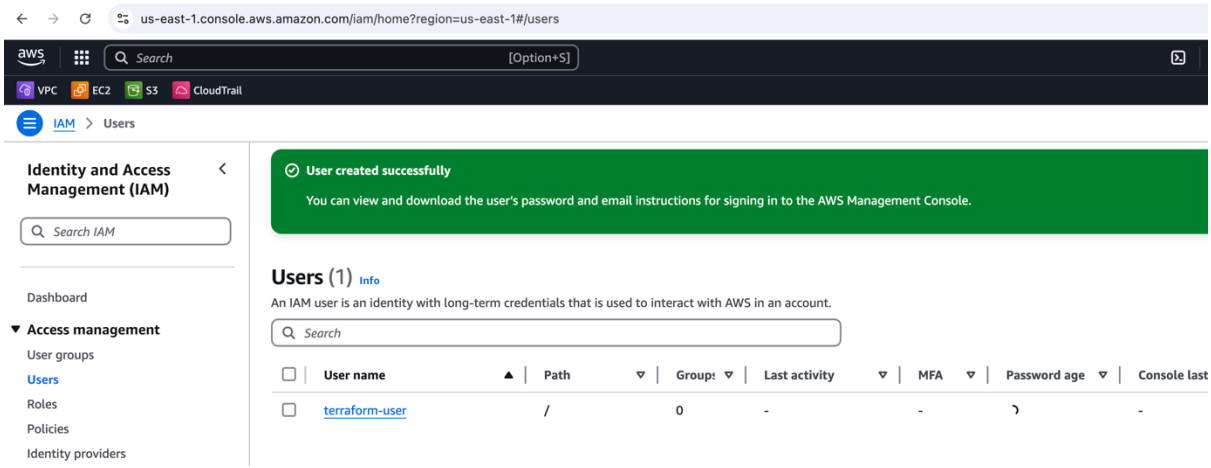
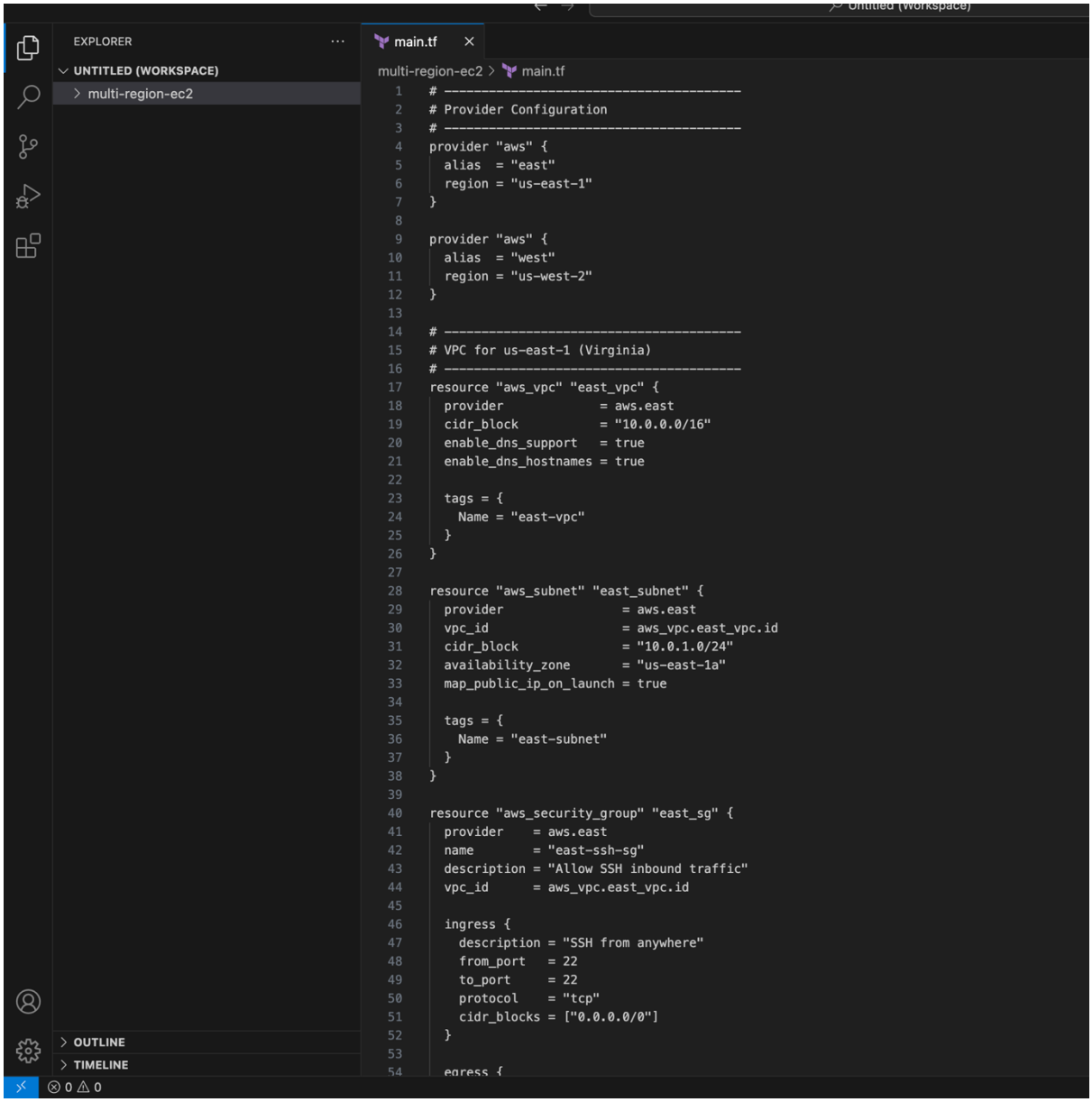
```
==> Checking for `sudo` access (which may request your password)...
Password:
==> This script will install:
/opt/homebrew/bin/brew
/opt/homebrew/share/doc/homebrew
/opt/homebrew/share/man/man1/brew.1
/opt/homebrew/share/zsh/site-functions/_brew
/opt/homebrew/etc/bash_completion.d/brew
/opt/homebrew

Press RETURN/ENTER to continue or any other key to abort:
==> /usr/bin/sudo /usr/sbin/chown -R vignesh:admin /opt/homebrew
==> Downloading and installing Homebrew...
remote: Enumerating objects: 6188, done.
remote: Counting objects: 100% (2629/2629), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 6188 (delta 2596), reused 2563 (delta 2563), pack-reused 3559 (from 4)
==> Updating Homebrew...
==> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:7645e2d653a335798030f6502e7834dfdbeeec5629429a1a34da5dbb2c57d63e
######################################################################## 100.0%
==> Pouring portable-ruby-3.3.8.arm64_big_sur.bottle.tar.gz
Warning: /opt/homebrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these commands in your terminal to add Homebrew to your PATH:
    echo >> /Users/vignesh/.zprofile
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/vignesh/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh

vignesh@Vignesh ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile
vignesh@Vignesh ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
vignesh@Vignesh ~ % brew tap hashicorp/tap
==> Tapping hashicorp/tap
Cloning into '/opt/homebrew/Library/Taps/hashicorp/homebrew-tap'...
remote: Enumerating objects: 5411, done.
remote: Counting objects: 100% (580/580), done.
remote: Compressing objects: 100% (210/210), done.
remote: Total 5411 (delta 466), reused 379 (delta 370), pack-reused 4831 (from 4)
Receiving objects: 100% (5411/5411), 990.37 KiB | 5.16 MiB/s, done.
Resolving deltas: 100% (3668/3668), done.
Tapped 2 casks and 32 formulae (92 files, 1.3MB).
vignesh@Vignesh ~ % brew install hashicorp/tap/terraform
==> Fetching hashicorp/tap/terraform
==> Downloading https://releases.hashicorp.com/terraform/1.11.4/terraform_1.11.4
######################################################################## 100.0%
==> Installing terraform from hashicorp/tap
🍺  /opt/homebrew/Cellar/terraform/1.11.4: 5 files, 84.8MB, built in 4 seconds
==> Running `brew cleanup terraform`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
vignesh@Vignesh ~ % terraform -v
Terraform v1.11.4
on darwin_arm64
vignesh@Vignesh ~ %
```

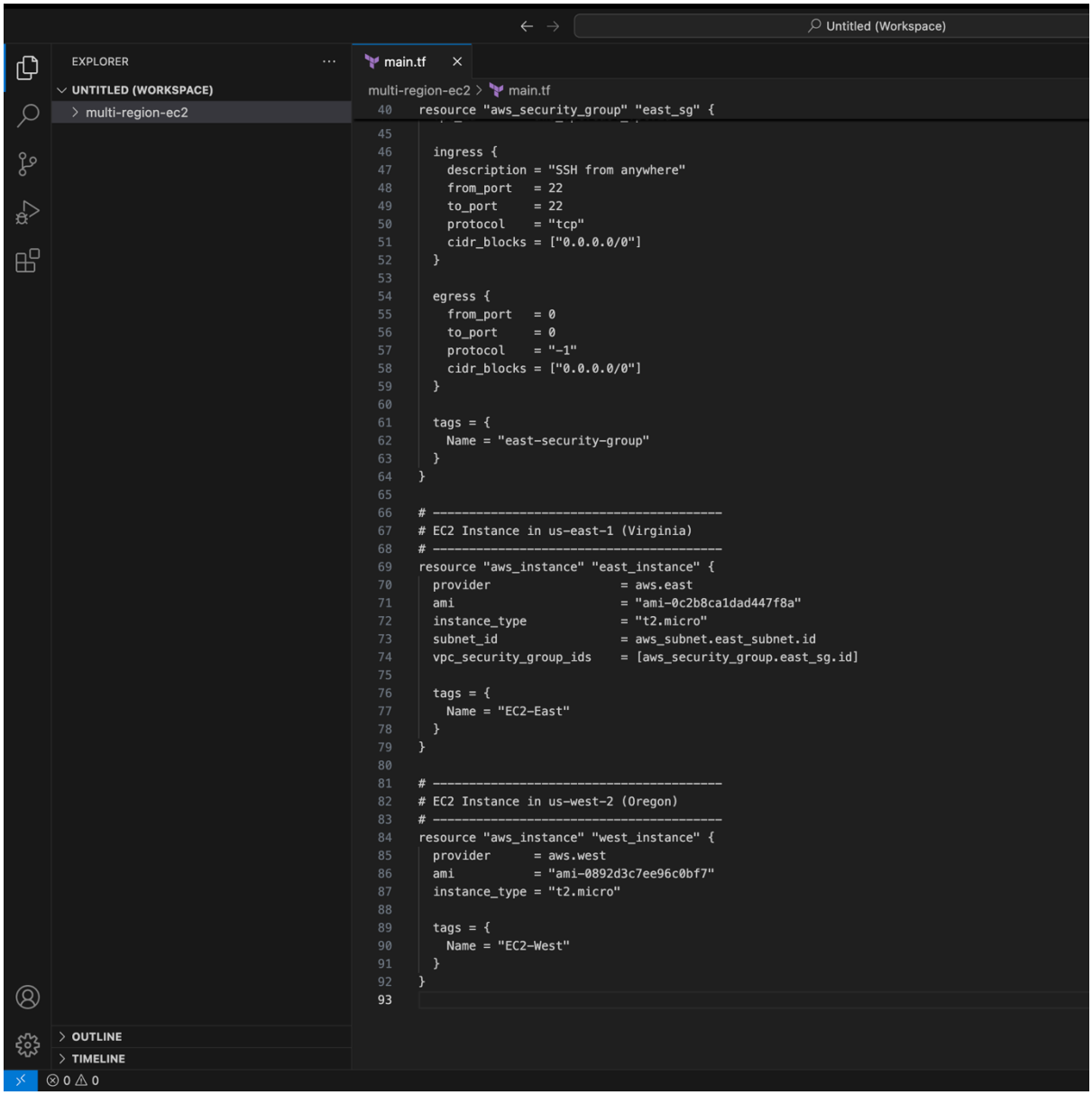
```
Press RETURN/ENTER to continue or any other key to abort:
==> /usr/bin/sudo /usr/sbin/chown -R vignesh:admin /opt/homebrew
==> Downloading and installing Homebrew...
remote: Enumerating objects: 6188, done.
remote: Counting objects: 100% (2629/2629), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 6188 (delta 2596), reused 2563 (delta 2563), pack-reused 3559 (from 4)
==> Updating Homebrew...
==> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:7645e2d653a335798030f6502e7834dfdbeeec5629429a1a34da5dbb2c57d63e
################################################################################# 100.0%
==> Pouring portable-ruby-3.3.8.arm64_big_sur.bottle.tar.gz
Warning: /opt/homebrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these commands in your terminal to add Homebrew to your PATH:
    echo >> /Users/vignesh/.zprofile
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/vignesh/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh

vignesh@Vignesh ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile
vignesh@Vignesh ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
vignesh@Vignesh ~ % brew tap hashicorp/tap
==> Tapping hashicorp/tap
Cloning into '/opt/homebrew/Library/Taps/hashicorp/homebrew-tap'...
remote: Enumerating objects: 5411, done.
remote: Counting objects: 100% (580/580), done.
remote: Compressing objects: 100% (210/210), done.
remote: Total 5411 (delta 466), reused 379 (delta 370), pack-reused 4831 (from 4)
Receiving objects: 100% (5411/5411), 990.37 KiB | 5.16 MiB/s, done.
Resolving deltas: 100% (3668/3668), done.
Tapped 2 casks and 32 formulae (92 files, 1.3MB).
vignesh@Vignesh ~ % brew install hashicorp/tap/terraform
==> Fetching hashicorp/tap/terraform
==> Downloading https://releases.hashicorp.com/terraform/1.11.4/terraform_1.11.4
################################################################################# 100.0%
==> Installing terraform from hashicorp/tap
🍺  /opt/homebrew/Cellar/terraform/1.11.4: 5 files, 84.8MB, built in 4 seconds
==> Running `brew cleanup terraform`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
vignesh@Vignesh ~ % terraform -v
Terraform v1.11.4
on darwin_arm64
vignesh@Vignesh ~ % aws --version
aws-cli/2.24.14 Python/3.12.9 Darwin/24.4.0 exe/x86_64
vignesh@Vignesh ~ % cd ..
vignesh@Vignesh /Users % cd vignesh
vignesh@Vignesh ~ % cd multi-region-ec2
vignesh@Vignesh multi-region-ec2 % terraform init
Initializing the backend...
Initializing provider plugins...

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
vignesh@Vignesh multi-region-ec2 %
```



```
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-West"
        }
      + tags_all                             = {
          + "Name" = "EC2-West"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

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

─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
vignesh@Vignesh multi-region-ec2 %
```

```
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "EC2-West"
        }
      + tags_all                             = {
          + "Name" = "EC2-West"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + capacity_reservation_specification (known after apply)

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 2 to add, 0 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: yes

aws_instance.east_instance: Creating...
aws_instance.west_instance: Creating...
```


EXPLORER
UNTITLED (WORKSPACE)
multi-region-ec2
main.tf
multi-region-ec2 > main.tf
# --
# Provider Configuration
# --
provider "aws" {
alias = "east"
region = "us-east-1"
}
provider "aws" {
alias = "west"
region = "us-west-2"
}
# --
# VPC for us-east-1 (Virginia)
# --
resource "aws_vpc" "east_vpc" {
provider = aws.east
cidr_block = "10.0.0.0/16"
enable_dns_support = true
enable_dns_hostnames = true
tags = {
Name = "east-vpc"
}
}
resource "aws_subnet" "east_subnet" {
provider = aws.east
vpc_id = aws_vpc.east_vpc.id
cidr_block = "10.0.1.0/24"
availability_zone = "us-east-1a"
map_public_ip_on_launch = true
tags = {
Name = "east-subnet"
}
}
resource "aws_security_group" "east_sg" {
provider = aws.east
name = "east-ssh-sg"
description = "Allow SSH inbound traffic"
vpc_id = aws_vpc.east_vpc.id
ingress {
description = "SSH from anywhere"
from_port = 22
to_port = 22
protocol = "tcp"
cidr_blocks = ["0.0.0.0/0"]
}
egress {
OUTLINE
TIMELINE
0 0



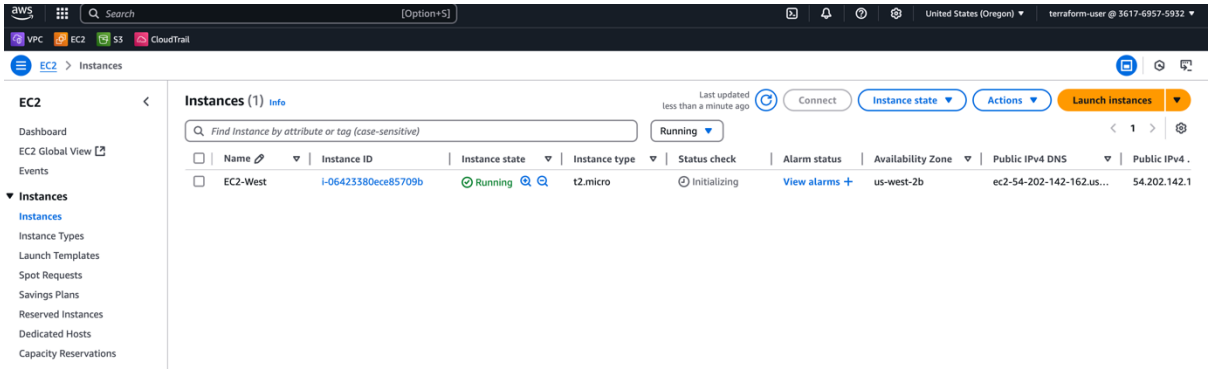

```hcl
resource "aws_security_group" "east_sg" {

  ingress {
    description = "SSH from anywhere"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "east-security-group"
  }
}

# ----------------------------------------
# EC2 Instance in us-east-1 (Virginia)
# ----------------------------------------
resource "aws_instance" "east_instance" {
  provider                = aws.east
  ami                     = "ami-0c2b8ca1dad447f8a"
  instance_type           = "t2.micro"
  subnet_id               = aws_subnet.east_subnet.id
  vpc_security_group_ids  = [aws_security_group.east_sg.id]

  tags = {
    Name = "EC2-East"
  }
}

# ----------------------------------------
# EC2 Instance in us-west-2 (Oregon)
# ----------------------------------------
resource "aws_instance" "west_instance" {
  provider      = aws.west
  ami           = "ami-0892d3c7ee96c0bf7"
  instance_type = "t2.micro"

  tags = {
    Name = "EC2-West"
  }
}

```

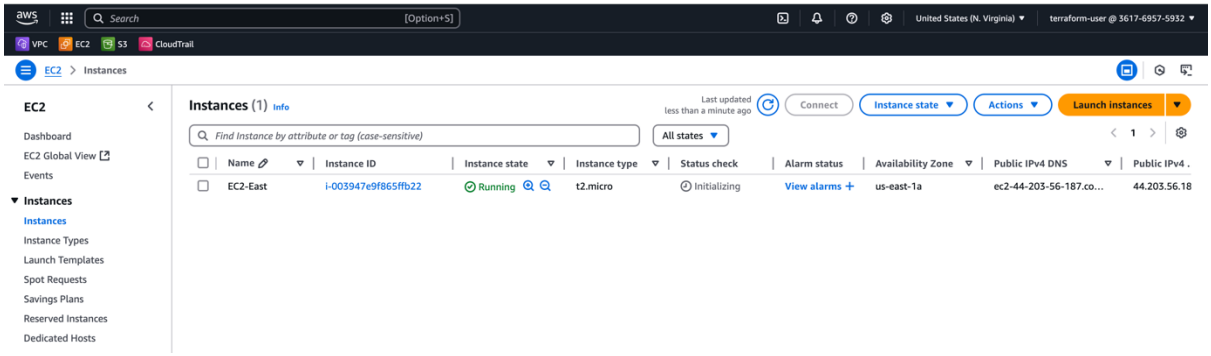
```
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
    + name                   = "east-ssh-sg"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags                   = {
        + "Name" = "east-security-group"
      }
    + tags_all               = {
        + "Name" = "east-security-group"
      }
    + vpc_id                 = (known after apply)
  }

# aws_subnet.east_subnet will be created
+ resource "aws_subnet" "east_subnet" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.1.0/24"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = true
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "east-subnet"
      }
    + tags_all                                       = {
        + "Name" = "east-subnet"
      }
    + vpc_id                                         = (known after apply)
  }

# aws_vpc.east_vpc will be created
+ resource "aws_vpc" "east_vpc" {
    + arn                                  = (known after apply)
    + cidr_block                           = "10.0.0.0/16"
    + default_network_acl_id               = (known after apply)
    + default_route_table_id               = (known after apply)
    + default_security_group_id            = (known after apply)
    + dhcp_options_id                      = (known after apply)
    + enable_dns_hostnames                 = true
    + enable_dns_support                   = true
    + enable_network_address_usage_metrics = (known after apply)
    + id                                   = (known after apply)
    + instance_tenancy                     = "default"
    + ipv6_association_id                  = (known after apply)
    + ipv6_cidr_block                      = (known after apply)
    + ipv6_cidr_block_network_border_group = (known after apply)
    + main_route_table_id                  = (known after apply)
    + owner_id                             = (known after apply)
    + tags                                 = {
        + "Name" = "east-vpc"
      }
    + tags_all                             = {
        + "Name" = "east-vpc"
      }
  }

Plan: 5 to add, 0 to change, 0 to destroy.
```

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terra
vignesh@Vignesh multi-region-ec2 %
```

```
      }
    + vpc_id                 = (known after apply)
  }

# aws_subnet.east_subnet will be created
+ resource "aws_subnet" "east_subnet" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-1a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.0.1.0/24"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
    + ipv6_native                                    = false
    + map_public_ip_on_launch                        = true
    + owner_id                                       = (known after apply)
    + private_dns_hostname_type_on_launch            = (known after apply)
    + tags                                           = {
        + "Name" = "east-subnet"
      }
    + tags_all                                       = {
        + "Name" = "east-subnet"
      }
    + vpc_id                                         = (known after apply)
  }

# aws_vpc.east_vpc will be created
+ resource "aws_vpc" "east_vpc" {
    + arn                                = (known after apply)
    + cidr_block                         = "10.0.0.0/16"
    + default_network_acl_id             = (known after apply)
    + default_route_table_id             = (known after apply)
    + default_security_group_id          = (known after apply)
    + dhcp_options_id                    = (known after apply)
    + enable_dns_hostnames               = true
    + enable_dns_support                 = true
    + enable_network_address_usage_metrics = (known after apply)
    + id                                 = (known after apply)
    + instance_tenancy                   = "default"
    + ipv6_association_id                = (known after apply)
    + ipv6_cidr_block                    = (known after apply)
    + ipv6_cidr_block_network_border_group = (known after apply)
    + main_route_table_id                = (known after apply)
    + owner_id                           = (known after apply)
    + tags                               = {
        + "Name" = "east-vpc"
      }
    + tags_all                           = {
        + "Name" = "east-vpc"
      }
  }

Plan: 5 to add, 0 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: yes

aws_vpc.east_vpc: Creating...
aws_instance.west_instance: Creating...
aws_vpc.east_vpc: Still creating... [10s elapsed]
aws_instance.west_instance: Still creating... [10s elapsed]
aws_vpc.east_vpc: Creation complete after 17s [id=vpc-03824c4cc736a4745]
aws_subnet.east_subnet: Creating...
aws_security_group.east_sg: Creating...
aws_instance.west_instance: Creation complete after 18s [id=i-06423380ece85709b]
aws_security_group.east_sg: Creation complete after 5s [id=sg-0bffb1c632fce6131]
aws_subnet.east_subnet: Still creating... [10s elapsed]
aws_subnet.east_subnet: Creation complete after 13s [id=subnet-09ace5a25ac30fb1b]
aws_instance.east_instance: Creating...
```

```
  # aws_subnet.east_subnet will be created
  + resource "aws_subnet" "east_subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "east-subnet"
        }
      + tags_all                                       = {
          + "Name" = "east-subnet"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.east_vpc will be created
  + resource "aws_vpc" "east_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "east-vpc"
        }
      + tags_all                             = {
          + "Name" = "east-vpc"
        }
    }

Plan: 5 to add, 0 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: yes

aws_vpc.east_vpc: Creating...
aws_instance.west_instance: Creating...
aws_vpc.east_vpc: Still creating... [10s elapsed]
aws_instance.west_instance: Still creating... [10s elapsed]
aws_vpc.east_vpc: Creation complete after 17s [id=vpc-03824c4cc736a4745]
aws_subnet.east_subnet: Creating...
aws_security_group.east_sg: Creating...
aws_instance.west_instance: Creation complete after 18s [id=i-06423380ece85709b]
aws_security_group.east_sg: Creation complete after 5s [id=sg-0bffb1c632fce6131]
aws_subnet.east_subnet: Still creating... [10s elapsed]
aws_subnet.east_subnet: Creation complete after 13s [id=subnet-09ace5a25ac30fb1b]
aws_instance.east_instance: Creating...
aws_instance.east_instance: Still creating... [10s elapsed]
aws_instance.east_instance: Creation complete after 16s [id=i-003947e9f865ffb22]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
vignesh@Vignesh multi-region-ec2 %
```


Search
[Option+S]
United States (Oregon)
terraform-user @ 3617-6957-5932
VPC
EC2
S3
CloudTrail
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Running
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 .
EC2-West
i-06423380ece85709b
Running
t2.micro
Initializing
View alarms
us-west-2b
ec2-54-202-142-162.us...
54.202.142.1


Search
[Option+S]
United States (N. Virginia)
terraform-user @ 3617-6957-5932
VPC
EC2
S3
CloudTrail
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 .
EC2-East
i-003947e9f865ffb22
Running
t2.micro
Initializing
View alarms
us-east-1a
ec2-44-203-56-187.co...
44.203.56.18