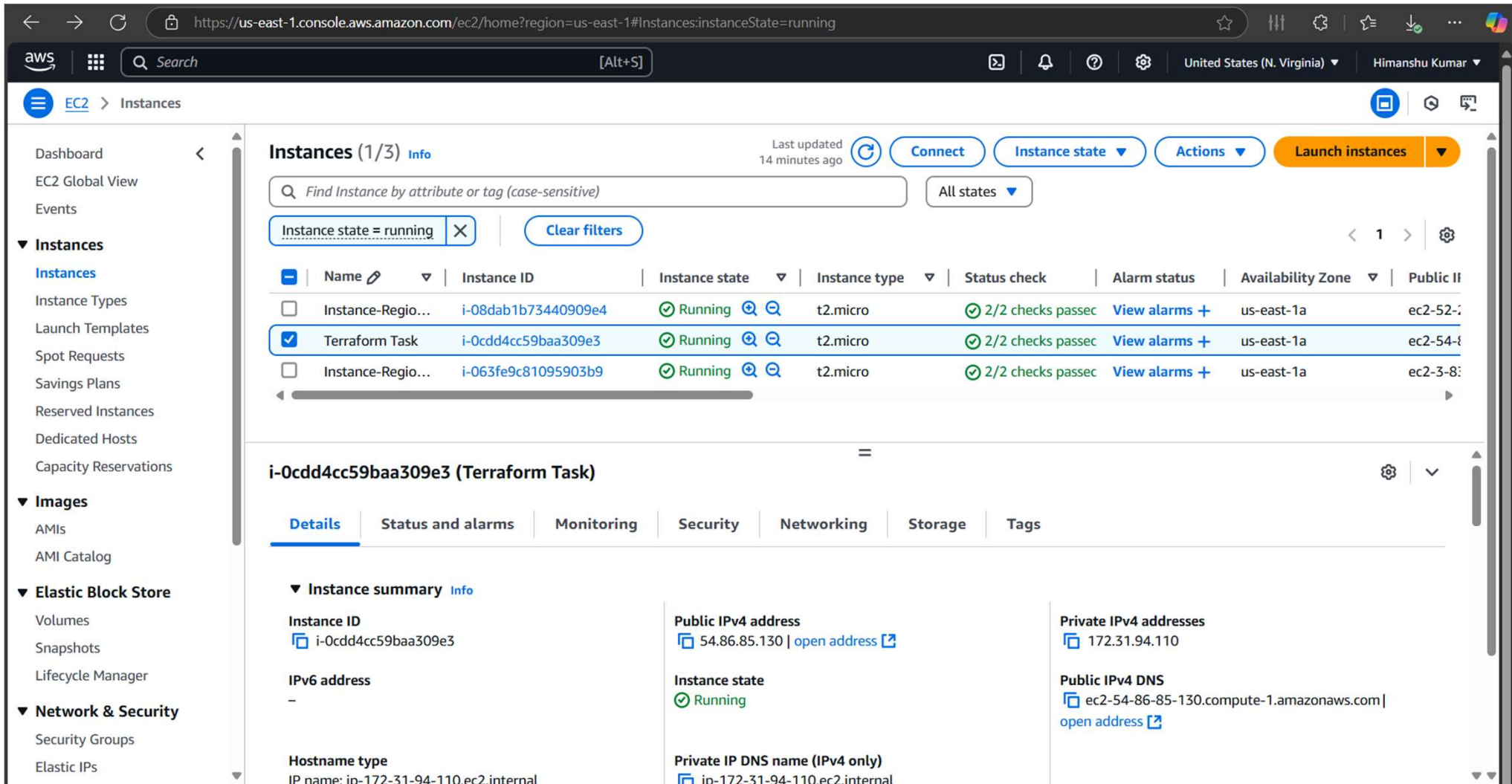


AWS EC2



The screenshot shows the AWS Management Console for the us-east-1 region. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays the 'Instances (1/3)' page, which includes a search bar, a filter for 'Instance state = running', and a table of instances. The 'Terraform Task' instance is selected, and its details are shown below the table.

Instances (1/3) Info

Last updated 14 minutes ago

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Instance-Regio...	i-08dab1b73440909e4	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-52-...
<input checked="" type="checkbox"/>	Terraform Task	i-0cdd4cc59baa309e3	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-...
<input type="checkbox"/>	Instance-Regio...	i-063fe9c81095903b9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-8...

i-0cdd4cc59baa309e3 (Terraform Task)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID i-0cdd4cc59baa309e3	Public IPv4 address 54.86.85.130 open address	Private IPv4 addresses 172.31.94.110
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-86-85-130.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-94-110.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-94-110.ec2.internal	

Install Terraform

```
ec2-user@ip-172-31-94-110:~/ / + v
Warning: Permanently added '54.86.85.130' (ED25519) to the list of known hosts.

#_
##### Amazon Linux 2023
#####\
#####|
## \###
##  \#/ https://aws.amazon.com/linux/amazon-linux-2023
##   V#
##   /m/

[ec2-user@ip-172-31-94-110 ~]$ sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum install -y terraform
Last metadata expiration check: 0:02:31 ago on Wed Feb 19 15:22:01 2025.
Package dnf-utils-4.3.0-13.amzn2023.0.5.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
Adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
Hashicorp Stable - x86_64 19 MB/s | 1.6 MB 00:00
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
terraform x86_64 1.10.5-1 hashicorp 27 M
Installing dependencies:
git x86_64 2.47.1-1.amzn2023.0.2 amazonlinux 54 k
git-core x86_64 2.47.1-1.amzn2023.0.2 amazonlinux 4.7 M
git-core-doc noarch 2.47.1-1.amzn2023.0.2 amazonlinux 2.8 M
perl-Error noarch 1:0.17029-5.amzn2023.0.2 amazonlinux 41 k
perl-File-Find noarch 1.37-477.amzn2023.0.6 amazonlinux 26 k
perl-Git noarch 2.47.1-1.amzn2023.0.2 amazonlinux 42 k
perl-TermReadKey x86_64 2.38-9.amzn2023.0.2 amazonlinux 36 k
perl-lib x86_64 0.65-477.amzn2023.0.6 amazonlinux 15 k
Transaction Summary
=====
Install 9 Packages

Total download size: 35 M
Installed size: 123 M
Downloading Packages:
(1/9): git-2.47.1-1.amzn2023.0.2.x86_64.rpm 1.3 MB/s | 54 kB 00:00
(2/9): perl-Error-0.17029-5.amzn2023.0.2.no 1.9 MB/s | 41 kB 00:00
(3/9): perl-File-Find-1.37-477.amzn2023.0.6 892 kB/s | 26 kB 00:00
(4/9): git-core-doc-2.47.1-1.amzn2023.0.2.n 20 MB/s | 2.8 MB 00:00
```

Transaction Summary

=====

Install 9 Packages

Total download size: 35 M

Installed size: 123 M

Downloading Packages:

(1/9): git-2.47.1-1.amzn2023.0.2.x86_64.rpm	1.3 MB/s		54 kB	00:00
(2/9): perl-Error-0.17029-5.amzn2023.0.2.no	1.9 MB/s		41 kB	00:00
(3/9): perl-File-Find-1.37-477.amzn2023.0.6	892 kB/s		26 kB	00:00
(4/9): git-core-doc-2.47.1-1.amzn2023.0.2.n	20 MB/s		2.8 MB	00:00
(5/9): perl-Git-2.47.1-1.amzn2023.0.2.noarc	976 kB/s		42 kB	00:00
(6/9): git-core-2.47.1-1.amzn2023.0.2.x86_6	27 MB/s		4.7 MB	00:00
(7/9): perl-TermReadKey-2.38-9.amzn2023.0.2	890 kB/s		36 kB	00:00
(8/9): perl-lib-0.65-477.amzn2023.0.6.x86_6	389 kB/s		15 kB	00:00
(9/9): terraform-1.10.5-1.x86_64.rpm	64 MB/s		27 MB	00:00

Total	55 MB/s		35 MB	00:00
Hashicorp Stable - x86_64	183 kB/s		3.9 kB	00:00

Importing GPG key 0xA621E701:

Userid : "HashiCorp Security (HashiCorp Package Signing) <security+packaging@hashicorp.com>"
Fingerprint: 798A EC65 4E5C 1542 8C8E 42EE AA16 FCBC A621 E701
From : https://rpm.releases.hashicorp.com/gpg

Key imported successfully

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing	:	1/1
Installing	: git-core-2.47.1-1.amzn2023.0.2.x86_64	1/9
Installing	: git-core-doc-2.47.1-1.amzn2023.0.2.noarch	2/9
Installing	: perl-lib-0.65-477.amzn2023.0.6.x86_64	3/9
Installing	: perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64	4/9
Installing	: perl-File-Find-1.37-477.amzn2023.0.6.noarch	5/9
Installing	: perl-Error-1:0.17029-5.amzn2023.0.2.noarch	6/9
Installing	: perl-Git-2.47.1-1.amzn2023.0.2.noarch	7/9
Installing	: git-2.47.1-1.amzn2023.0.2.x86_64	8/9
Installing	: terraform-1.10.5-1.x86_64	9/9
Running scriptlet:	terraform-1.10.5-1.x86_64	9/9
Verifying	: git-2.47.1-1.amzn2023.0.2.x86_64	1/9
Verifying	: git-core-2.47.1-1.amzn2023.0.2.x86_64	2/9
Verifying	: git-core-doc-2.47.1-1.amzn2023.0.2.noarch	3/9
Verifying	: perl-Error-1:0.17029-5.amzn2023.0.2.noarch	4/9
Verifying	: perl-File-Find-1.37-477.amzn2023.0.6.noarch	5/9
Verifying	: perl-Git-2.47.1-1.amzn2023.0.2.noarch	6/9
Verifying	: perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64	7/9

aws configure

```
ec2-user@ip-172-31-94-110: ~$   
Preparing      : 1/1  
Installing     : git-core-2.47.1-1.amzn2023.0.2.x86_64 1/9  
Installing     : git-core-doc-2.47.1-1.amzn2023.0.2.noarch 2/9  
Installing     : perl-lib-0.65-477.amzn2023.0.6.x86_64 3/9  
Installing     : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 4/9  
Installing     : perl-File-Find-1.37-477.amzn2023.0.6.noarch 5/9  
Installing     : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 6/9  
Installing     : perl-Git-2.47.1-1.amzn2023.0.2.noarch 7/9  
Installing     : git-2.47.1-1.amzn2023.0.2.x86_64 8/9  
Installing     : terraform-1.10.5-1.x86_64 9/9  
Running scriptlet: terraform-1.10.5-1.x86_64 9/9  
Verifying      : git-2.47.1-1.amzn2023.0.2.x86_64 1/9  
Verifying      : git-core-2.47.1-1.amzn2023.0.2.x86_64 2/9  
Verifying      : git-core-doc-2.47.1-1.amzn2023.0.2.noarch 3/9  
Verifying      : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 4/9  
Verifying      : perl-File-Find-1.37-477.amzn2023.0.6.noarch 5/9  
Verifying      : perl-Git-2.47.1-1.amzn2023.0.2.noarch 6/9  
Verifying      : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 7/9  
Verifying      : perl-lib-0.65-477.amzn2023.0.6.x86_64 8/9  
Verifying      : terraform-1.10.5-1.x86_64 9/9  
  
Installed:  
git-2.47.1-1.amzn2023.0.2.x86_64  
git-core-2.47.1-1.amzn2023.0.2.x86_64  
git-core-doc-2.47.1-1.amzn2023.0.2.noarch  
perl-Error-1:0.17029-5.amzn2023.0.2.noarch  
perl-File-Find-1.37-477.amzn2023.0.6.noarch  
perl-Git-2.47.1-1.amzn2023.0.2.noarch  
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  
perl-lib-0.65-477.amzn2023.0.6.x86_64  
terraform-1.10.5-1.x86_64  
  
Complete!  
[ec2-user@ip-172-31-94-110 ~]$ terraform --version  
Terraform v1.10.5  
on linux_amd64  
[ec2-user@ip-172-31-94-110 ~]$ aws --version  
aws-cli/2.17.18 Python/3.9.20 Linux/6.1.127-135.201.amzn2023.x86_64 source/x86_64.amzn.2023  
[ec2-user@ip-172-31-94-110 ~]$ aws configure  
AWS Access Key ID [None]: AKIAU6PKIZLABK45WWV7  
AWS Secret Access Key [None]: bHu0GKAbRtp9dwGMAaSYgrB7/NU$PRNGB5xY1kT  
Default region name [None]: us-east-1  
Default output format [None]: json  
[ec2-user@ip-172-31-94-110 ~]$ aws s3 ls  
[ec2-user@ip-172-31-94-110 ~]$ mkdir terraform-aws  
cd terraform-aws  
[ec2-user@ip-172-31-94-110 terraform-aws]$ nano main.tf
```

Terraform Project Setup , Initialize & Apply Terraform

```
ec2-user@ip-172-31-94-110:~/ terraform-aws$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.87.0...
- Installed hashicorp/aws v5.87.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
[ec2-user@ip-172-31-94-110 terraform-aws]$ terraform validate
Success! The configuration is valid.

[ec2-user@ip-172-31-94-110 terraform-aws]$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.instance_region1 will be created
+ resource "aws_instance" "instance_region1" {
  + ami                  = "ami-0c55b159cbfafa1f0"
  + arn                  = (known after apply)
  + 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)
```


ec2-user@ip-172-31-94-110:~/

```
+ 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" = "Instance-Region1"
}
+ tags_all = {
  + "Name" = "Instance-Region1"
}
+ 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)
```

```
+ 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)
}

# aws_instance.instance_region2 will be created
+ resource "aws_instance" "instance_region2" {
+   ami                      = "ami-00dff3b01f99da94d"
+   arn                      = (known after apply)
+   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)
```

```
+ 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" = "Instance-Region1"
}
+ tags_all = {
  + "Name" = "Instance-Region1"
}
+ 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)
```



```
+ 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)
}

# aws_instance.instance_region2 will be created
+ resource "aws_instance" "instance_region2" {
+   ami                        = "ami-00dffb3b01f99da94d"
+   arn                       = (known after apply)
+   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)
```

Apply complete

```
ec2-user@ip-172-31-94-110:~/ + ▼ - □ ×

+ tags                                = {
  + "Name" = "Instance-Region2"
}
+ tags_all                            = {
  + "Name" = "Instance-Region2"
}
+ 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.
aws_instance.instance_region1: Creating...
aws_instance.instance_region2: Creating...
aws_instance.instance_region1: Still creating... [10s elapsed]
aws_instance.instance_region2: Still creating... [10s elapsed]
aws_instance.instance_region1: Creation complete after 13s [id=i-08dab1b73440909e4]
aws_instance.instance_region2: Creation complete after 13s [id=i-0106e2f750c507c10]

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