Task : 17 Launch Linux EC2 instances in two regions using a single Terraform file.

1. Installed terraform:

```
[ec2-user@ip-172-31-0-137 ~]$ sudo yum install -y yum-utils
Last metadata expiration check: 0:17:41 ago on Thu Jul 25 05:59:12 2024.
Package dnf-utils-4.3.0-13.amzn2023.0.4.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-0-137 ~]$ sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
Adding repo from: https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
[ec2-user@ip-172-31-0-137 ~]$ sudo yum -y install terraform
Hashicorp Stable - x86_64                                                              4.1 MB/s | 1.4 MB     00:00
Dependencies resolved.
=================================================================================================================
 Package              Architecture        Version                        Repository              Size
=================================================================================================================
Installing:
 terraform            x86_64              1.9.3-1                         hashicorp               27 M
Installing dependencies:
 git                  x86_64              2.40.1-1.amzn2023.0.3          amazonlinux             54 k
 git-core             x86_64              2.40.1-1.amzn2023.0.3          amazonlinux             4.3 M
 git-core-doc         noarch              2.40.1-1.amzn2023.0.3          amazonlinux             2.6 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              2.40.1-1.amzn2023.0.3          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


Installed:
  git-2.40.1-1.amzn2023.0.3.x86_64              git-core-2.40.1-1.amzn2023.0.3.x86_64         git-core-doc-2.40.1-1.amzn2023.0.3.noarch
  perl-Error-1:0.17029-5.amzn2023.0.2.noarch   perl-File-Find-1.37-477.amzn2023.0.6.noarch   perl-Git-2.40.1-1.amzn2023.0.3.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64  perl-lib-0.65-477.amzn2023.0.6.x86_64         terraform-1.9.3-1.x86_64

Complete!
[ec2-user@ip-172-31-0-137 ~]$ terraform --version
Terraform v1.9.3
on linux_amd64
[ec2-user@ip-172-31-0-137 ~]$
```
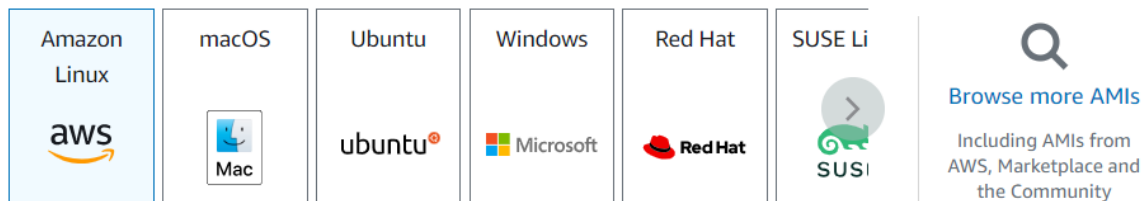
2. Create a directory for write terraform file:

```
[ec2-user@ip-172-31-0-137 ~]$ mkdir terraform
[ec2-user@ip-172-31-0-137 ~]$ cd terraform/
[ec2-user@ip-172-31-0-137 terraform]$ vi main.tf
```

3. Main.tf: [terraform file]

```
provider "aws"{
        alias = "region1"
        region = "us-east-1"
}
provider "aws"{
        alias  = "region2"
        region = "us-east-2"
}
resource "aws_instance" "terraform-useast1" {
        provider = aws.region1
        instance_type = "t2.micro"
        ami = "ami-0427090fd1714168b"

        tags = {
                Name = "terraform-useast1"
                }
}
resource "aws_instance" "terraform-useast2" {
        provider = aws.region2
        instance_type = "t2.micro"
        ami = "ami-00db8dadb36c9815e"

        tags = {
                Name = "terraform-useast2"
                }
}
```

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Li
aws	Mac	ubuntu®	Microsoft	Red Hat	SUSE

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI   Free tier eligible
ami-00db8dadb36c9815e (64-bit (x86), uefi-preferred) / ami-0fb5231409345e557 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture   Boot mode   AMI ID

64-bit (x86) ▼   uefi-preferred   ami-00db8dadb36c9815e   Verified provider

Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Li
aws	Mac	ubuntu	Microsoft	Red Hat	SUSE

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI	Free tier eligible
ami-0427090fd1714168b (64-bit (x86), uefi-preferred) / ami-0582e4fe9b72a5fe1 (64-bit (Arm), uefi)	
Virtualization: hvm ENA enabled: true Root device type: ebs	▼

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Architecture	Boot mode	AMI ID	
64-bit (x86) ▼	uefi-preferred	ami-0427090fd1714168b	**Verified provider**

4. Terraform init :

```
[ec2-user@ip-172-31-0-137 terraform]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.59.0...
- Installed hashicorp/aws v5.59.0 (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-172-31-0-137 terraform]$
```

```
[ec2-user@ip-172-31-0-137 ~]$ cd terraform/
[ec2-user@ip-172-31-0-137 terraform]$ aws configure
AWS Access Key ID [****************EW4V]: AKIAQEIP3LCPLLIJEW4V
AWS Secret Access Key [****************bBb4]: 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
Default region name [us-east-1]: us-east-1
Default output format [json]: json
[ec2-user@ip-172-31-0-137 terraform]$ ls
main.tf
[ec2-user@ip-172-31-0-137 terraform]$ vi main.tf
[ec2-user@ip-172-31-0-137 terraform]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.59.0

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,
```

```
[ec2-user@ip-172-31-0-137 terraform]$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions
  + create

Terraform will perform the following actions:

  # aws_instance.terraform-useast1 will be created
  + resource "aws_instance" "terraform-useast1" {
      + ami                          = "ami-0427090fd1714168b"
      + 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)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + host_resource_group_arn      = (known after apply)
      + iam_instance_profile         = (known after apply)
```

Terraform apply:

```
[ec2-user@ip-172-31-0-137 terraform]$ terraform apply

Terraform used the selected providers to generate the following execution
  + create

Terraform will perform the following actions:

  # aws_instance.terraform-useast1 will be created
  + resource "aws_instance" "terraform-useast1" {
      + ami                          = "ami-0427090fd1714168b"
      + 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)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + host_resource_group_arn      = (known after apply)
      + iam_instance_profile         = (known after apply)
```

```
  Enter a value: yes

aws_instance.terraform-useast2: Creating...
aws_instance.terraform-useast1: Creating...
aws_instance.terraform-useast2: Still creating... [10s elapsed]
aws_instance.terraform-useast1: Still creating... [10s elapsed]
aws_instance.terraform-useast2: Still creating... [20s elapsed]
aws_instance.terraform-useast1: Still creating... [20s elapsed]
aws_instance.terraform-useast2: Still creating... [30s elapsed]
aws_instance.terraform-useast1: Still creating... [30s elapsed]
aws_instance.terraform-useast2: Creation complete after 36s [id=i-080259598095248d5]
aws_instance.terraform-useast1: Still creating... [40s elapsed]
aws_instance.terraform-useast1: Still creating... [50s elapsed]
aws_instance.terraform-useast1: Still creating... [1m0s elapsed]
aws_instance.terraform-useast1: Still creating... [1m10s elapsed]
aws_instance.terraform-useast1: Still creating... [1m20s elapsed]
aws_instance.terraform-useast1: Still creating... [1m30s elapsed]
aws_instance.terraform-useast1: Still creating... [1m40s elapsed]
aws_instance.terraform-useast1: Creation complete after 1m47s [id=i-02d7dfc0f8d792da5]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-0-137 terraform]$ terraform apply
```

```
[ec2-user@ip-172-31-0-137 terraform]$ ls
main.tf  terraform.tfstate
```

8. **Instance created in us-east-1:**

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	terraform-use...	i-02d7dfc0f8d792da5	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passec	View alarms +	us-east-1d	ec2-3-95

Instances (1/1) Info

Instances (1/1) Info

[⟳] | Connect | Instance state ▼ | Actions ▼ | **Launch instances** ▼

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 > ⚙

Elastic IP	IPv6 IPs ▽	Monitoring ▽	Security group name ▽	Key name ▽	Launch time ▽	Platfor... ▽
–	–	disabled	default	–	2024/07/25 15:28 GMT+5:30	Linux/UNIX

9. Instance created in us-east-1:

Instances (1/1) Info

[⟳] | 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 IPv
☑	terraform-use...	i-080259598095248d5	⊘ Running ⊕ ⊖	t2.micro	⊘ 2/2 checks passed	View alarms ➕	us-east-2a	ec2-3-14-2

Instances (1/1) Info

[⟳] | Connect | Instance state ▼ | Actions ▼ | **Launch instances** ▼

🔍 Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 > ⚙

▽	Elastic IP	IPv6 IPs ▽	Monitoring ▽	Security group name ▽	Key name ▽	Launch time ▽	Platfor... ▽
	–	–	disabled	default	–	2024/07/25 15:28 GMT+5:30	Linux/UNIX