

Terraform Task

Task Description:

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

The screenshot shows the AWS IAM User Details page for a user named 'Devops-user'. The left sidebar contains navigation links for Identity and Access Management (IAM), Access Management (including Users, Roles, Policies, Identity providers, Account settings, Root access management, and Temporary delegation requests), Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, and Resource control policies), and IAM Identity Center (AWS Organizations). The main content area displays the 'Devops-user' info page, which includes a Summary section with ARN (arn:aws:iam::913218764078:user/Devops-user), Console access (Enabled without MFA), Created (January 22, 2026, 21:23 UTC+05:30), Last console sign-in (Never), and Access key 1 (Create access key). Below the summary is a 'Permissions' tab, which lists two attached policies: 'AdministratorAccess' (AWS managed - job function, Directly) and 'IAMUserChangePassword' (AWS managed, Directly). There is also a 'Permissions boundary (not set)' section and a 'Generate policy based on CloudTrail events' section with a 'Generate policy' button. The bottom of the page includes standard AWS footer links for Privacy, Terms, and Cookie preferences.

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v6.27.0
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ aws configure
AWS Access Key ID [*****4605]: AKIA5JIAUPUXF6FHJVP4
AWS Secret Access Key [*****Zvrh]: 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
Default region name [ap-south-1]: us-east-1
Default output format [json]: json
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ cat <<EOF > main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Provider for US East (N. Virginia)
provider "aws" {
  alias  = "us"
  region = "us-east-1"
}

# Provider for Asia Pacific (Mumbai)
provider "aws" {
  alias  = "india"
  region = "ap-south-1"
}

# EC2 instance in us-east-1
resource "aws_instance" "us_ec2" {
  provider      = aws.us
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-US-EC2"
  }
}

# EC2 instance in ap-south-1
resource "aws_instance" "india_ec2" {
  provider      = aws.india
  ami           = "ami-0da59f1af71ea4ad2"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-India-EC2"
  }
}
EOF
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ cat main.tf
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid
}
}
EOF
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ cat main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

# Provider for US East (N. Virginia)
provider "aws" {
  alias  = "us"
  region = "us-east-1"
}

# Provider for Asia Pacific (Mumbai)
provider "aws" {
  alias  = "india"
  region = "ap-south-1"
}

# EC2 instance in us-east-1
resource "aws_instance" "us_ec2" {
  provider      = aws.us
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-US-EC2"
  }
}

# EC2 instance in ap-south-1
resource "aws_instance" "india_ec2" {
  provider      = aws.india
  ami           = "ami-0da59f1af71ea4ad2"
  instance_type = "t2.micro"

  tags = {
    Name = "Terraform-India-EC2"
  }
}
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid
tags = {
  Name = "Terraform-India-EC2"
}
}
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ terraform init
terraform apply
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file

Error: Failed to query available provider packages

Could not retrieve the list of available versions for provider hashicorp/aws: locked provider
registry.terraform.io/hashicorp/aws 6.27.0 does not match configured version constraint ~> 5.0; must
use terraform init -upgrade to allow selection of new versions

To see which modules are currently depending on hashicorp/aws and what versions are specified, run
the following command:
  terraform providers

Error: Required plugins are not installed

The installed provider plugins are not consistent with the packages selected in the dependency lock
file:
- registry.terraform.io/hashicorp/aws: there is no package for registry.terraform.io/hashicorp/aws 6
.27.0 cached in .terraform/providers

Terraform uses external plugins to integrate with a variety of different infrastructure services. To
download the plugins required for this configuration, run:
  terraform init

junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ terraform init -upgrade
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/aws v5.100.0...
- Installed hashicorp/aws v5.100.0 (signed by HashiCorp)
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ cat <<EOF > main.tf
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  alias  = "us"
  region = "us-east-1"
}

provider "aws" {
  alias  = "india"
  region = "ap-south-1"
}

# EC2 in us-east-1
resource "aws_instance" "us_ec2" {
  provider      = aws.us
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t3.micro"

  tags = {
    Name = "Terraform-US-EC2"
  }
}

# EC2 in ap-south-1
resource "aws_instance" "india_ec2" {
  provider      = aws.india
  ami           = "ami-0da59f1af71ea4ad2"
  instance_type = "t3.micro"

  tags = {
    Name = "Terraform-India-EC2"
  }
}
EOF
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ terraform apply
aws_instance.us_ec2: Refreshing state... [id=i-06a9e5af14cf6652e]

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:
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid
+ 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 = "t3.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" = "Terraform-India-EC2"
  }
+ tags_all = {
    + "Name" = "Terraform-India-EC2"
  }
+ 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)
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid
}
+ tags_all = {
  + "Name" = "Terraform-India-EC2"
}
+ 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: 1 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.india_ec2: Creating...
aws_instance.india_ec2: Still creating... [00m13s elapsed]
aws_instance.india_ec2: Creation complete after 16s [id=i-00e427491b86d5d0c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$ terraform destroy
aws_instance.india_ec2: Refreshing state... [id=i-00e427491b86d5d0c]
aws_instance.us_ec2: Refreshing state... [id=i-06a9e5af14cf6652e]

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

Terraform will perform the following actions:

# aws_instance.india_ec2 will be destroyed
- resource "aws_instance" "india_ec2" {
    - ami = "ami-0da59f1af71ea4ad2" -> null
    - arn = "arn:aws:ec2:ap-south-1:913218764078:instance/i-00e427491b86d5d0c" -> null
    - associate_public_ip_address = true -> null
    - availability_zone = "ap-south-1a" -> null
    - cpu_core_count = 1 -> null
    - cpu_threads_per_core = 2 -> null
    - disable_api_stop = false -> null
    - disable_api_termination = false -> null
    - ebs_optimized = false -> null
    - get_password_data = false -> null
    - hibernation = false -> null
    - id = "i-00e427491b86d5d0c" -> null
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state = "running" -> null
    - instance_type = "t3.micro" -> null
    - ipv6_address_count = 0 -> null
    - ipv6_addresses = [] -> null
    - monitoring = false -> null
    - placement_partition_number = 0 -> null
    - primary_network_interface_id = "eni-0906c462212174ed6" -> null
    - private_dns = "ip-172-31-42-44.ap-south-1.compute.internal" -> null
    - private_ip = "172.31.42.44" -> null
    - public_dns = "ec2-13-127-247-82.ap-south-1.compute.amazonaws.com" -> null
    - public_ip = "13.127.247.82" -> null
    - secondary_private_ips = [] -> null
    - security_groups = [
        - "default",
    ] -> null
    - source_dest_check = true -> null
    - subnet_id = "subnet-06f6ed7e3a53ec22a" -> null
    - tags = {
        - "Name" = "Terraform-India-EC2"
    } -> null
    - tags_all = {
        - "Name" = "Terraform-India-EC2"
    } -> null
}
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid

- credit_specification {
    - cpu_credits = "unlimited" -> null
}

- enclave_options {
    - enabled = false -> null
}

- maintenance_options {
    - auto_recovery = "default" -> null
}

- metadata_options {
    - http_endpoint = "enabled" -> null
    - http_protocol_ipv6 = "disabled" -> null
    - http_put_response_hop_limit = 2 -> null
    - http_tokens = "required" -> null
    - instance_metadata_tags = "disabled" -> null
}

- private_dns_name_options {
    - enable_resource_name_dns_a_record = false -> null
    - enable_resource_name_dns_aaaa_record = false -> null
    - hostname_type = "ip-name" -> null
}

- root_block_device {
    - delete_on_termination = true -> null
    - device_name = "/dev/xvda" -> null
    - encrypted = false -> null
    - iops = 3000 -> null
    - tags = {} -> null
    - tags_all = {} -> null
    - throughput = 125 -> null
    - volume_id = "vol-0e2b8727574e824c2" -> null
    - volume_size = 8 -> null
    - volume_type = "gp3" -> null
    # (1 unchanged attribute hidden)
}
}

# aws_instance.us_ec2 will be destroyed
- resource "aws_instance" "us_ec2" {
    - ami = "ami-0c02fb55956c7d316" -> null
    - arn = "arn:aws:ec2:us-east-1:913218764078:instance/i-06a9e5af14
cf6652e" -> null
    - associate_public_ip_address = true -> null
    - availability_zone = "us-east-1f" -> null
    - cpu_core_count = 1 -> null
    - cpu_threads_per_core = 2 -> null
    - disable_api_stop = false -> null
}
```

```
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid

- private_dns_name_options {
  - enable_resource_name_dns_a_record    = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type                      = "ip-name" -> null
}

- root_block_device {
  - delete_on_termination = true -> null
  - device_name          = "/dev/xvda" -> null
  - encrypted            = false -> null
  - iops                 = 100 -> null
  - tags                 = {} -> null
  - tags_all              = {} -> null
  - throughput            = 0 -> null
  - volume_id             = "vol-0f96e121ce98e542e" -> null
  - volume_size            = 8 -> null
  - volume_type            = "gp2" -> null
  # (1 unchanged attribute hidden)
}
}

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

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.india_ec2: Destroying... [id=i-00e427491b86d5d0c]
aws_instance.us_ec2: Destroying... [id=i-06a9e5af14cf6652e]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 00m13s elapsed]
aws_instance.us_ec2: Still destroying... [id=i-06a9e5af14cf6652e, 00m13s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 00m23s elapsed]
aws_instance.us_ec2: Still destroying... [id=i-06a9e5af14cf6652e, 00m23s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 00m33s elapsed]
aws_instance.us_ec2: Still destroying... [id=i-06a9e5af14cf6652e, 00m33s elapsed]
aws_instance.us_ec2: Destruction complete after 36s
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 00m43s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 00m56s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 01m06s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 01m16s elapsed]
aws_instance.india_ec2: Still destroying... [id=i-00e427491b86d5d0c, 01m29s elapsed]
aws_instance.india_ec2: Destruction complete after 1m30s

Destroy complete! Resources: 2 destroyed.
junaid@LAPTOP-GU5B805P:/mnt/c/Users/Junaid$
```