

L1

1

```

EXPLORER
  OPEN EDITORS
    Welcome
    assignment.tf 1
  TERRAFORM
    .terraform
    .terraform.lock.hcl
    assignment.tf 1
    LICENSE.txt
    terraform.exe
    terraform.tfstate
    terraform.tfstate.back...

assignment.tf > resource "aws_vpc" "my-vpc" > tags
1 provider "aws" {
2   region = "ap-south-1"
3 }
4
5 # VPC
6 resource "aws_vpc" "my-vpc" {
7   cidr_block = "10.0.0.0/24"
8   tags = {
9     Name = "Terraform"
10 }
11 }
12
13 # Internet Gateway
14 resource "aws_internet_gateway" "my-ig" {
15   vpc_id = aws_vpc.my-vpc.id
16   tags = {
17     Name = "TerraformIG"
18 }
19 }
20
21 # Route Table
22 resource "aws_route_table" "my-rt" {
23   vpc_id = aws_vpc.my-vpc.id
24   route {
25     cidr_block = "0.0.0.0/0"
26     gateway_id = aws_internet_gateway.my-ig.id
27   }
28   tags = {
29     Name = "TerraformRT"
30 }
31 }

```

2

```

# Subnet
resource "aws_subnet" "my-sn" {
  vpc_id            = aws_vpc.my-vpc.id
  map_public_ip_on_launch = false
  cidr_block        = "10.0.0.0/25"
  tags = {
    Name = "TerraformSN"
  }
}

# Security Group ports (22, 80, 443)
resource "aws_security_group" "my-sg" {
  name        = "my-securitygroup"
  vpc_id      = aws_vpc.my-vpc.id

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

  ingress {
    from_port = 80
    to_port   = 80
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port = 443
    to_port   = 443
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

```

3

```

# Network Interface setup/creation
resource "aws_network_interface" "my-ni" {
  subnet_id      = aws_subnet.my-sn.id
  security_groups = [aws_security_group.my-sg.id]
  tags = {
    Name = "terraformSN"
  }
}

# Elastic IP
resource "aws_eip" "eip-ni" {
  vpc = true
  tags = {
    Name = "My-EIP"
  }
}

# Elastic IP Association
resource "aws_eip_association" "eip-association" {
  allocation_id      = aws_eip.eip-ni.allocation_id
  network_interface_id = aws_network_interface.my-ni.id
}

```

4

```

# EC2 instance setup
resource "aws_instance" "demo" {
  ami          = "ami-08ee1453725d19cdb"
  instance_type = "t2.micro"
  key_name      = "amazon-linux"

  user_data = <<-EOF
    #!/bin/bash
    sudo yum install git -y
  EOF

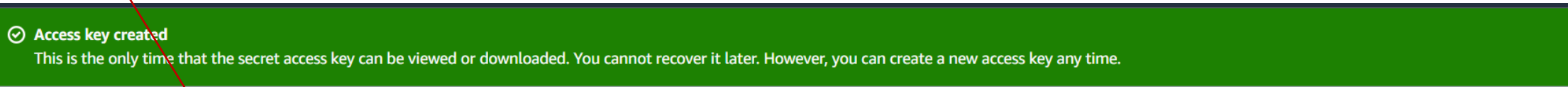
  tags = {
    Name = "terraformDemoInstance"
  }

  network_interface {
    device_index      = 0
    network_interface_id = aws_network_interface.my-ni.id
  }
}

```

Configure the Creds

Using user Access Key



✔ **Access key created**
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

[IAM](#) > [Users](#) > [Terraform](#) > Create access key

Step 1
[Access key best practices & alternatives](#)



Step 2 - optional
[Set description tag](#)

Step 3
Retrieve access keys

Retrieve access keys [Info](#)

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
 AKIAZ3MGNEPIEDRBTUCP	 VU1XL0LGGzH97Gg60CAWJ4PPTz5srmVWgMw2uOrx Hide

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Aws configure

```
PS O:\Terraform> aws configure
AWS Access Key ID [*****TUCP]: AKIAZ3MGNEPIC2MDTWHI
AWS Secret Access Key [*****uOrx]: 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
Default region name [us-east-1]: ap-south-1
Default output format [json]: json
```

```
PS O:\Terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.64.0...
- Installed hashicorp/aws v5.64.0 (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.

Terraform Init

Terraform validate

```
PS O:\Terraform> terraform validate

Warning: Argument is deprecated

  with aws_eip.eip-ni,
  on assignment.tf line 71, in resource "aws_eip" "eip-ni":
  71:   vpc = true

  use domain attribute instead

Success! The configuration is valid, but there were some validation warnings as shown above.
```

```
PS 0:\Terraform> terraform plan
```

Terraform will perform the following actions:

```
# aws_eip.eip-n1 will be created
+ resource "aws_eip" "eip-n1" {
+   allocation_id = (known after apply)
+   arn            = (known after apply)
+   association_id = (known after apply)
+   carrier_ip     = (known after apply)
+   customer_owned_ip = (known after apply)
+   domain         = (known after apply)
+   id             = (known after apply)
+   instance       = (known after apply)
+   network_border_group = (known after apply)
+   network_interface = (known after apply)
+   private_dns      = (known after apply)
+   private_ip       = (known after apply)
+   ptr_record       = (known after apply)
+   public_dns       = (known after apply)
+   public_ip        = (known after apply)
+   public_ipv4_pool = (known after apply)
+   tags             = {
+     + "Name" = "My-EIP"
+   }
+   tags_all = {
+     + "Name" = "My-EIP"
+   }
+   vpc = true
}
```

```
# aws_eip_association.eip-association will be created
+ resource "aws_eip_association" "eip-association" {
+   allocation_id    = (known after apply)
+   id               = (known after apply)
+   instance_id      = (known after apply)
+   network_interface_id = (known after apply)
+   private_ip_address = (known after apply)
+   public_ip         = (known after apply)
+ }

```

```
# aws_instance.demo will be created
```

```
resource "aws_instance" "demo" {
+ ami                                = "ami-0e86e20dae9224db8"
+ arn                                = (known after apply)
+ associate_public_ip_address       = (known after apply)
+ availability_zone                  = "ap-south-1b"
+ 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)
+ 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                           = "ubuntu"
+ 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)
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
  + "Name" = "terraformDemoInstance"
}
+ tags_all = {
  + "Name" = "terraformDemoInstance"
}
+ tenancy = (known after apply)
+ user_data = "0dcf0609ba7a0357f"
+ 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 {
```

```
+ device index = 0
```

```
+ network interface id = (known after apply)
```

```
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

# aws_internet_gateway.my-ig will be created
+ resource "aws_internet_gateway" "my-ig" {
+   arn           = (known after apply)
+   id            = (known after apply)
+   owner_id      = (known after apply)
+   tags          = {
+     + "Name" = "TerraformIG"
+   }
+   tags_all      = {
+     + "Name" = "TerraformIG"
+   }
+   vpc_id        = (known after apply)
+ }
```

```
# aws_network_interface.my-ni will be created
+ resource "aws_network_interface" "my-ni" {
+   arn                = (known after apply)
+   id                 = (known after apply)
+   interface_type      = (known after apply)
+   ipv4_prefix_count   = (known after apply)
+   ipv4_prefixes       = (known after apply)
+   ipv6_address_count  = (known after apply)
+   ipv6_address_list   = (known after apply)
+   ipv6_address_list_enabled = false
+   ipv6_addresses      = (known after apply)
+   ipv6_prefix_count   = (known after apply)
+   ipv6_prefixes       = (known after apply)
+   mac_address         = (known after apply)
+   outpost_arn         = (known after apply)
+   owner_id            = (known after apply)
+   private_dns_name    = (known after apply)
```

```
_enabled = false
         = (known after apply)
```

```

+ owner_id = (known after apply)
+ private_dns_name = (known after apply)
+ private_ip = (known after apply)
+ private_ip_list = (known after apply)
+ private_ip_list_enabled = false
+ private_ips = (known after apply)
+ private_ips_count = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ subnet_id = (known after apply)
+ tags = {
  + "Name" = "terraformSN"
}
+ tags_all = {
  + "Name" = "terraformSN"
}

+ attachment (known after apply)
}

```

```
# aws_route_table.my-rt will be created
+ resource "aws_route_table" "my-rt" {
+   arn                = (known after apply)
+   id                 = (known after apply)
+   owner_id           = (known after apply)
+   propagating_vgw    = (known after apply)
+   route              = [
+     + {
+       + cidr_block      = "0.0.0.0/0"
+       + gateway_id      = "aws_internet_gateway.my-ig.id"
+     },
+     # (11 unchanged attributes hidden)
+   ]
+   tags               = {
+     + "Name" = "TerraformRT"
+   }
+   tags_all           = {
+     + "Name" = "TerraformRT"
+   }
+   vpc_id             = (known after apply)
}
```

```
+ resource "aws_security_group" "my-sg" {
+   + arn                                = (known after apply)
+   + description                        = "Managed by Terraform"
+   + egress                             = (known after apply)
+   + id                                 = (known after apply)
+   + ingress                             = [
+     + {
+       + cidr_blocks                    = [
+         + "0.0.0.0/0",
+       ]
+       + from_port                      = 22
+       + ipv6_cidr_blocks               = []
+       + prefix_list_ids                = []
+       + protocol                       = "tcp"
+       + security_groups                = []
+       + self                           = false
+       + to_port                        = 22
+       # (1 unchanged attribute hidden)
+     },
+   + {
+     + cidr_blocks                      = [
+       + "0.0.0.0/0",
+     ]
+     + from_port                      = 443
+     + ipv6_cidr_blocks               = []
+     + prefix_list_ids                = []
+     + protocol                       = "tcp"
+     + security_groups                = []
+     + self                           = false
+     + to_port                        = 443
+     # (1 unchanged attribute hidden)
+   },
+ }
+ {
+   + cidr_blocks                      = [
+     + "0.0.0.0/0",
+   ]
+   + from_port                      = 80
+   + ipv6_cidr_blocks               = []
+   + prefix_list_ids                = []
+ }
```

```

+ from_port = 443
+ ipwv_cidr_blocks = [
+   prefix_list_ids = []
+   protocol = "tcp"
+   security_groups = []
+   self = false
+   to_port = 443
+   # (1 unchanged attribute hidden)
+ },
+ {
+   cidr_blocks = [
+     "0.0.0.0/0",
+   ]
+   from_port = 80
+   ipwv_cidr_blocks = []
+   prefix_list_ids = []
+   protocol = "tcp"
+   security_groups = []
+   self = false
+   to_port = 80
+   # (1 unchanged attribute hidden)
+ },
+ name = "my-securitygroup"
+ name_prefix = "(known after apply)"
+ owner_id = "(known after apply)"
+ revoke_rules_on_delete = false
+ tags_all = "(known after apply)"
+ vpc_id = "(known after apply)"

```

```
# aws_subnet.my-sn will be created
+ resource "aws_subnet" "my-sn" {
+   arn                                     = (known after apply)
+   assign_ipv6_address_on_creation        = false
+   availability_zone                      = (known after apply)
+   availability_zone_id                   = (known after apply)
+   cidr_block                             = "10.0.0.0/24"
+   enable_dns64                           = false
+   enable_resource_name_dns_a_record_on_launch = false
}
```

```
+ from_port = 22
+ ipv6_cidr_blocks = []
+ prefix_list_ids = []
+ protocol = "tcp"
+ security_groups = []
+ self = false
+ to_port = 22
# (1 unchanged attribute hidden)
},
+ {
+   cidr_blocks = [
+     "0.0.0.0/0",
+   ]
+   from_port = 443
+   ipv6_cidr_blocks = []
+   prefix_list_ids = []
+   protocol = "tcp"
+   security_groups = []
+   self = false
+   to_port = 443
# (1 unchanged attribute hidden)
},
+ {
+   cidr_blocks = [
+     "0.0.0.0/0",
+   ]
+   from_port = 80
+   ipv6_cidr_blocks = []
+   prefix_list_ids = []
+   protocol = "tcp"
+   security_groups = []
+   self = false
+   to_port = 80
# (1 unchanged attribute hidden)
},
},
1
},
name = "my-securitygro
name_prefix = (known after ap
owner_id = (known after ap
revoke_rules_on_delete = false
tags_all = (known after ap
```

```
# aws_vpc.my-vpc will be created
+ resource "aws_vpc" "my-vpc" {
  + arn                                = (known after apply)
  + cidr_block                        = "10.0.0.0/24"
  + 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             = (known after apply)
  + 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" = "Terraform"
  }
+ tags_all                           = {
  + "Name" = "Terraform"
}
}
```

Plan: 9 to add, 0 to change, 0 to destroy.

Warning: Argument is deprecated

```
with aws_eip.eip-ni,  
on assignment.tf line 71, in resource "aws_eip" "eip-ni":  
71:   vpc = true
```

```
use domain attribute instead
```

(and one more similar warning elsewhere)

Terraform 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.demo: Creating...

aws_instance.demo: Still creating... [10s elapsed]

aws_instance.demo: Still creating... [20s elapsed]

aws_instance.demo: Still creating... [30s elapsed]

aws_instance.demo: Still creating... [40s elapsed]

aws_instance.demo: Creation complete after 47s [id=i-004aab0d08657c04f]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

EC2

Security Group

Instances (1) [Info](#)

Last updated less than a minute ago

Refresh

Connect

Instance state ▼

Actions ▼

Launch instances ▼

Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 >

⚙

<input type="checkbox"/>	Name <div>✎</div> ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IP
<input type="checkbox"/>	terraformDem...	i-004aab0d08657c04f	<div>Running</div> <div>🔍</div> <div>🔍</div>	t2.micro	<div>2/2 checks passed</div>	View alarms +	ap-south-1a	-

Details

Security group name

📄

my-securitygroup

Owner

📄

677276099536

Security group ID

📄

sg-086f4d24d373e6941

Inbound rules count

3

Permission entries

Description

📄

Managed by Terraform

Outbound rules count

0

Permission entries

VPC ID

📄

[vpc-08079ec98b9b85e9f](#) [🔗](#)

Inbound rules

Outbound rules

Tags

Inbound rules (3)

Refresh

Manage tags

Edit inbound rules

Search

< 1 >

⚙

<input type="checkbox"/>	Name ▼	Security group rule... ▼	IP version ▼	Type ▼	Protocol ▼	Port range
<input type="checkbox"/>	-	sgr-007803c0bf056e8a3	IPv4	HTTPS	TCP	443
<input type="checkbox"/>	-	sgr-05cc340084e792fd4	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0d46a8d203bc10fca	IPv4	HTTP	TCP	80

Elastic IP

Elastic IP addresses (1)						Actions	Allocate Elastic IP address
<input type="text" value="Find resources by attribute or tag"/>						< 1 >	⚙
<input type="checkbox"/>	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record		
<input type="checkbox"/>	My-EIP	65.1.213.148	Public IP	eipalloc-0211eb6052f432651	-		

Vpc

Your VPCs (1/2) Info								Last updated less than a minute ago	Actions	Create VPC
<input type="text" value="Search"/>								< 1 >	⚙	
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP				
<input type="checkbox"/>	-	vpc-0b5597dd6c533c097	✓ Available	172.31.0.0/16	-	dopt-4				
<input checked="" type="checkbox"/>	Terraform	vpc-08079ec98b9b85e9f	✓ Available	10.0.0.0/24	-	dopt-4				

Subnets

Subnets (4) Info							Last updated 1 minute ago	Actions	Create subnet
<input type="text" value="Find resources by attribute or tag"/>							< 1 >	⚙	
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR				
<input type="checkbox"/>	-	subnet-0d26783c5e8c79076	✓ Available	vpc-0b5597dd6c533c097	172.31.0.0/20				
<input type="checkbox"/>	-	subnet-0e4370822fed6a3f2	✓ Available	vpc-0b5597dd6c533c097	172.31.16.0/20				
<input type="checkbox"/>	-	subnet-04c255a8d18649d3f	✓ Available	vpc-0b5597dd6c533c097	172.31.32.0/20				
<input type="checkbox"/>	TerraformSN	subnet-09e648839832bfcf1	✓ Available	vpc-08079ec98b9b85e9f Terr...	10.0.0.0/25				

Route Tables

Route tables (3) Info							Last updated 2 minutes ago	Actions	Create route table
<input type="text" value="Find resources by attribute or tag"/>							< 1 >	⚙	
<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC			
<input type="checkbox"/>	TerraformRT	rtb-020e1bc06a3f2aa43	-	-	No	vpc-08079ec98b9b85e9f			

Internet Gateway

Internet gateways (2) [Info](#)



Actions ▼

Create internet gateway

Search

< 1 >



<input type="checkbox"/>	Name ▼	Internet gateway ID ▼	State ▼	VPC ID ▼	Owner
<input type="checkbox"/>	-	igw-045b8a48864ae9016	✓ Attached	vpc-0b5597dd6c533c097	677276099536
<input type="checkbox"/>	TerraformIG	igw-08ab05e40a2a5e62c	✓ Attached	vpc-08079ec98b9b85e9f Terraform	677276099536

Terraform destroy

```
aws_eip_association.eip-association: Destruction complete after 4s
aws_eip.eip-ni: Destroying... [id=eipalloc-0211eb6052f432651]
aws_route_table.my-rt: Destruction complete after 6s
aws_internet_gateway.my-ig: Destroying... [id=igw-08ab05e40a2a5e62c]
aws_eip.eip-ni: Destruction complete after 4s
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 10s elapsed]
aws_internet_gateway.my-ig: Destruction complete after 7s
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 20s elapsed]
aws_instance.demo: Still destroying... [id=i-004aab0d08657c04f, 30s elapsed]
aws_instance.demo: Destruction complete after 37s
aws_network_interface.my-ni: Destroying... [id=eni-0db17d303c32b05b5]
aws_network_interface.my-ni: Destruction complete after 1s
aws_subnet.my-sn: Destroying... [id=subnet-09e648839832bfcf1]
aws_security_group.my-sg: Destroying... [id=sg-086f4d24d373e6941]
aws_security_group.my-sg: Destruction complete after 6s
aws_subnet.my-sn: Still destroying... [id=subnet-09e648839832bfcf1, 10s elapsed]
aws_subnet.my-sn: Destruction complete after 12s
aws_vpc.my-vpc: Destroying... [id=vpc-08079ec98b9b85e9f]
aws_vpc.my-vpc: Destruction complete after 2s
```

Destroy complete! Resources: 9 destroyed.