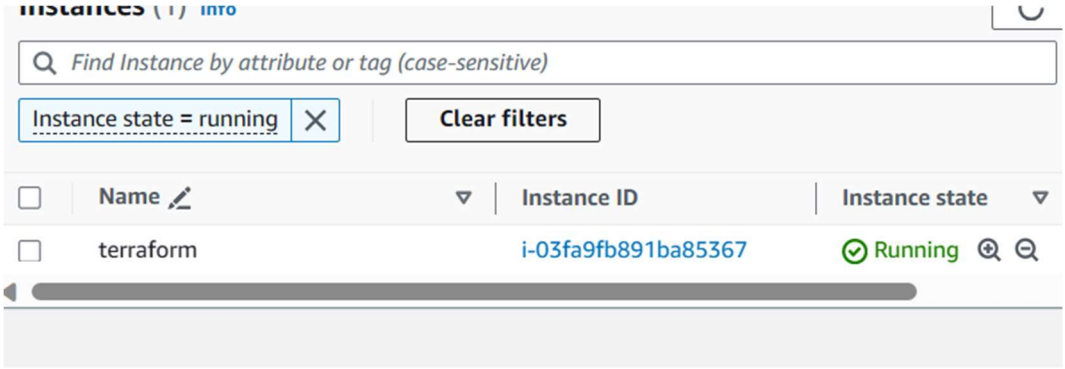
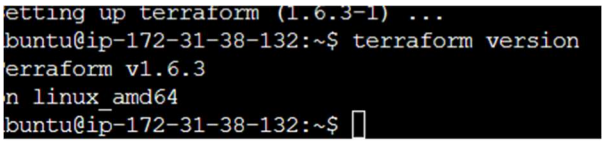
Write Terraform script to create highly available infrastructure in AWS. The infra should have1 vpc, 3 subnets setup in 3 different az and 2 instances setup in 2 different subnets

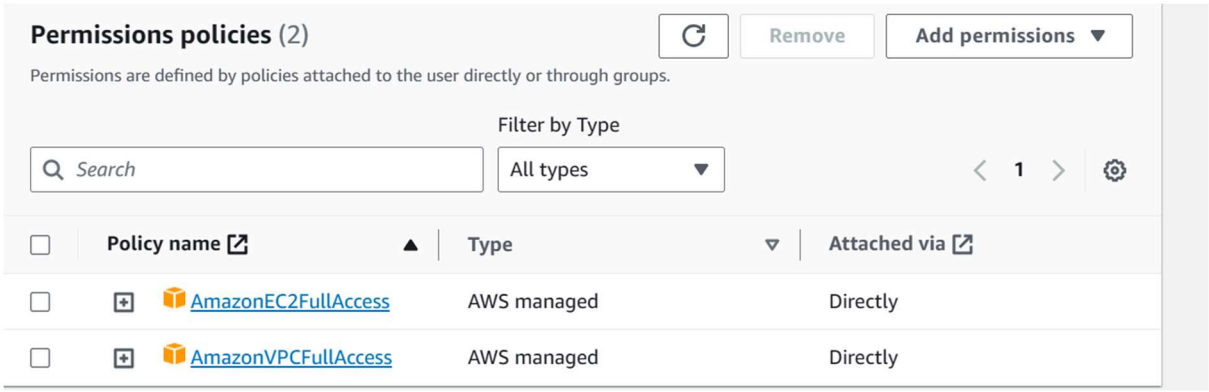
Solution:

1- Launch the instance in aws and install terraform

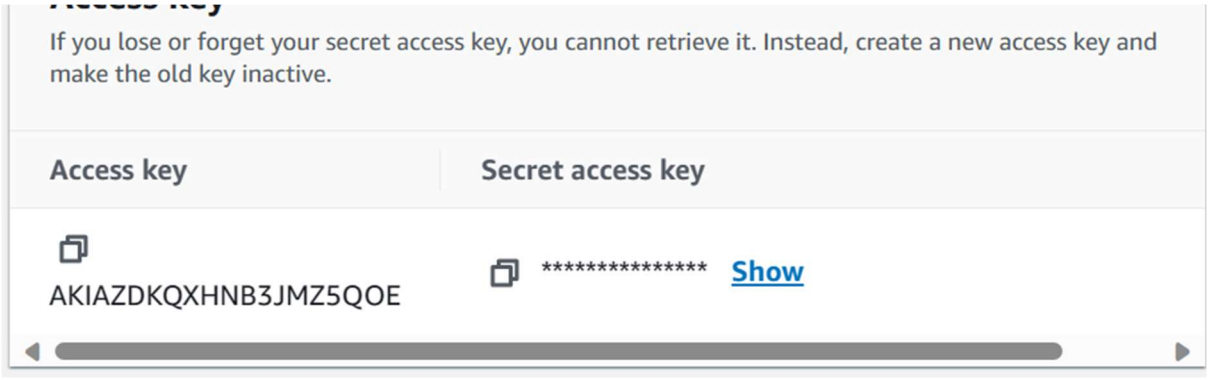




Creating an iam user with ec2full access and vpc full access



Creating access key and secrete access key



Installing awscli and configure

```
root@ip-172-31-38-132:/home/ubuntu# aws configure
AWS Access Key ID [None]: AKIAZDKQXHNB3JMZ5QOE
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
root@ip-172-31-38-132:/home/ubuntu#
```

Creating terraform script

```
provider "aws" {
  profile = "terraform"
  region  = "ap-south-1"
}

resource "aws_vpc" "main" {
  cidr_block       = "10.0.0.0/16"
  instance_tenancy = "default"

  tags = {
    Name = "terraform"
  }
}

resource "aws_internet_gateway" "main_igw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "internet-gateway"
  }
}

resource "aws_subnet" "subnet_1" {
```

```
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.1.0/24"
  availability_zone = "ap-south-1b"

  tags = {
    Name = "subnet"
  }
}

resource "aws_subnet" "subnet_2" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.2.0/24"
  availability_zone = "ap-south-1a"

  tags = {
    Name = "subnet2"
  }
}

resource "aws_subnet" "subnet_3" {
  vpc_id            = aws_vpc.main.id
  cidr_block        = "10.0.3.0/24"
  availability_zone = "ap-south-1c"

  tags = {
    Name = "subnet3"
  }
}

resource "aws_route_table" "route_table" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "route-table"
  }
}

resource "aws_route" "routing" {
  route_table_id         = aws_route_table.route_table.id
  destination_cidr_block = "0.0.0.0/0" # Route all traffic to the Internet
Gateway
  gateway_id             = aws_internet_gateway.main_igw.id
}

resource "aws_route_table_association" "subnet_assoc_1" {
  subnet_id      = aws_subnet.subnet_1.id
  route_table_id = aws_route_table.route_table.id
}
```

```
resource "aws_route_table_association" "subnet_assoc_2" {
  subnet_id      = aws_subnet.subnet_2.id
  route_table_id = aws_route_table.route_table.id
}

resource "aws_route_table_association" "subnet_assoc_3" {
  subnet_id      = aws_subnet.subnet_3.id
  route_table_id = aws_route_table.route_table.id
}

resource "aws_instance" "terraform_1" {
  ami                         = "ami-0a7cf821b91bcccbc"
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.subnet_1.id
  key_name                    = "terrakey"
  associate_public_ip_address = true

  tags = {
    Name = "terraform task1"
  }
}

resource "aws_instance" "terraform_2" {
  ami                         = "ami-0a7cf821b91bcccbc"
  instance_type               = "t2.micro"
  subnet_id                   = aws_subnet.subnet_2.id
  key_name                    = "terrakey"
  associate_public_ip_address = true

  tags = {
    Name = "terraform task2"
  }
}
```

Perform terraform commands

Terraform init

```
root@ip-172-31-38-132:/home/ubuntu# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.24.0

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,
rerun this command to reinitialize your working directory. If you forget, othe
commands will detect it and remind you to do so if necessary.
root@ip-172-31-38-132:/home/ubuntu#
```

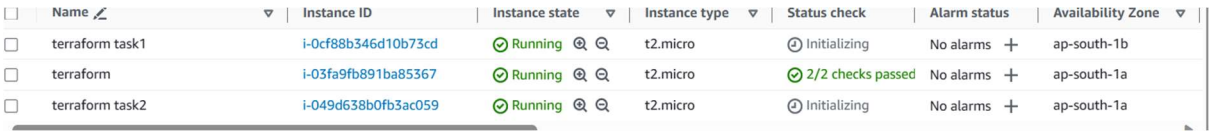
Terraform plan

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

erraform will perform the following actions:

  # aws_instance.terraform_1 will be created
  + resource "aws_instance" "terraform_1" {
```

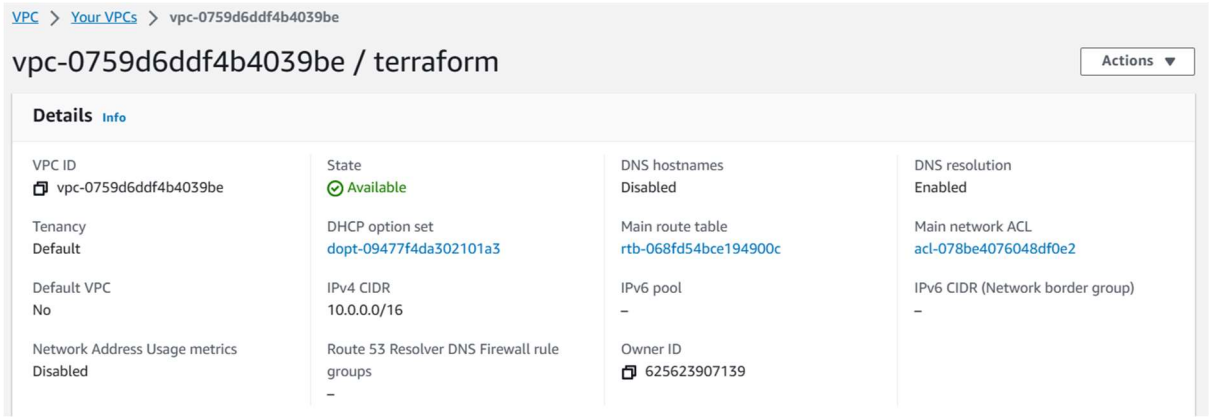
Terraform apply

```
ws_internet_gateway.main_igw: Creating...
ws_internet_gateway.main_igw: Creation complete after 0s [id=igw-0bea4366c7ea65c92]
ws_route_table.route_table: Creation complete after 0s [id=rtb-0b3cf3bb5f642d688]
ws_route.routing: Creating...
ws_subnet.subnet_2: Creation complete after 0s [id=subnet-0609d46c7d57d14eb]
ws_instance.terraform_2: Creating...
ws_route_table_association.subnet_assoc_2: Creating...
ws_subnet.subnet_3: Creation complete after 0s [id=subnet-05560db34347d6e13]
ws_route_table_association.subnet_assoc_3: Creating...
ws_route_table_association.subnet_assoc_2: Creation complete after 0s [id=rtbassoc-0421efb591cb0edbe]
ws_route_table_association.subnet_assoc_3: Creation complete after 0s [id=rtbassoc-03cb6eac087b08f34]
ws_subnet.subnet_1: Creation complete after 0s [id=subnet-07626d56e7fd5f425]
ws_instance.terraform_1: Creating...
ws_route.routing: Creation complete after 0s [id=r-rtb-0b3cf3bb5f642d6881080289494]
ws_route_table_association.subnet_assoc_1: Creating...
ws_route_table_association.subnet_assoc_1: Creation complete after 1s [id=rtbassoc-0ef54031128454dbc]
ws_instance.terraform_2: Still creating... [10s elapsed]
ws_instance.terraform_1: Still creating... [10s elapsed]
ws_instance.terraform_2: Still creating... [20s elapsed]
ws_instance.terraform_1: Still creating... [20s elapsed]
ws_instance.terraform_1: Creation complete after 21s [id=i-0cf88b346d10b73cd]
ws_instance.terraform_2: Still creating... [30s elapsed]
ws_instance.terraform_2: Creation complete after 31s [id=i-049d638b0fb3ac059]

pply complete! Resources: 12 added, 0 changed, 0 destroyed.
```

Output



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
terraform task1	i-0cf88b346d10b73cd	Running	t2.micro	Initializing	No alarms	ap-south-1b
terraform	i-03fa9fb891ba85367	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a
terraform task2	i-049d638b0fb3ac059	Running	t2.micro	Initializing	No alarms	ap-south-1a



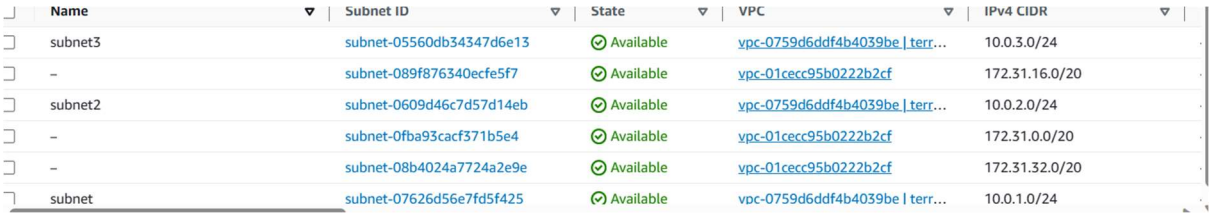
VPC > Your VPCs > vpc-0759d6ddf4b4039be

vpc-0759d6ddf4b4039be / terraform

Actions

Details Info

VPC ID vpc-0759d6ddf4b4039be	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-09477f4da302101a3	Main route table rtb-068fd54bce194900c	Main network ACL acl-078be4076048df0e2
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups –	Owner ID 625623907139	



Name	Subnet ID	State	VPC	IPv4 CIDR
subnet3	subnet-05560db34347d6e13	Available	vpc-0759d6ddf4b4039be | terr...	10.0.3.0/24
–	subnet-089f876340ecfe5f7	Available	vpc-01cecc95b0222b2cf	172.31.16.0/20
subnet2	subnet-0609d46c7d57d14eb	Available	vpc-0759d6ddf4b4039be | terr...	10.0.2.0/24
–	subnet-0fba93cacf371b5e4	Available	vpc-01cecc95b0222b2cf	172.31.0.0/20
–	subnet-08b4024a7724a2e9e	Available	vpc-01cecc95b0222b2cf	172.31.32.0/20
subnet	subnet-07626d56e7fd5f425	Available	vpc-0759d6ddf4b4039be | terr...	10.0.1.0/24