EC2 to S3 communication - IAM Role

Terraform

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

Terraform File

iam.tf

assume_role_policy — The policy that grants an entity permission to assume the role.
This is going to create IAM role but we can't link this role to AWS instance and for that, we need EC2 instance Profile

```
resource "aws_iam_role" "ec2" {
  name = "test_role"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF

  tags = {
      tag-key = "tag-value"
  }
}


resource "aws_iam_instance_profile" "ec2_s3" {
  name = "ec2-s3"
   role = aws_iam_role.ec2.name
}
```

if we execute the above code, we have Role and Instance Profile but with no permission. Next step is to add IAM Policies which allows EC2 instance to execute specific commands for eg: access to S3 Bucket

Adding IAM Policies
To give full access to S3 bucket
Attach this role to EC2 instance

```
resource "aws_iam_role_policy" "ec2_s3" {
  name = "ec2_s3"
  role = "${aws_iam_role.ec2.id}"

  policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
EOF
}
```

ec2.tf

```
provider "aws" {
    region= var.region
    access_key = var.access_key
```

```
  secret_key = var.secret_key
}
resource "aws_instance" "ec2" {
 ami = "ami-052efd3df9dad4825"
 instance_type = "t2.micro"
 subnet_id = aws_subnet.public.id
 # Security group assign to instance
 vpc_security_group_ids = [aws_security_group.sg_vpc.id]
 # key name
 key_name = "08-09-2022"
 iam_instance_profile = aws_iam_instance_profile.ec2_s3.name
 user_data = <<EOF
                         #! /bin/bash
  # Update all packages
  sudo apt update
  sleep 20
  curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  sleep 20
  sudo apt install unzip
  unzip awscliv2.zip
  sleep 40
  sudo ./aws/install
  apt install awscli
  EOF

 tags = {
  Name = "Ubuntu_Ec2 "
 }
}
```

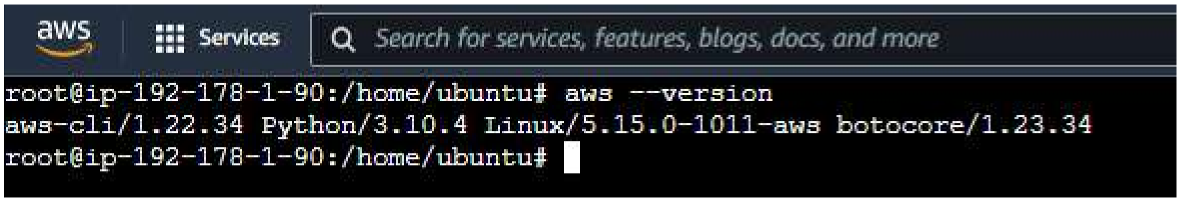
In this we have Created S3 bucket

```
resource "aws_s3_bucket" "b" {
 bucket = "my-yestf-test-bucket-16-09-2022"

 tags = {
  Name        = "My bucket"
  Environment = "Dev"
 }
}
```

Now Connect to EC2

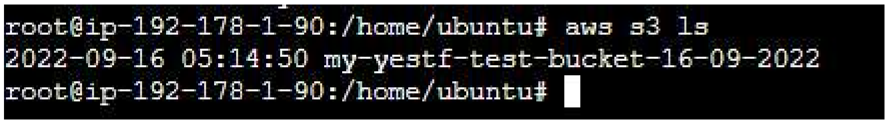
Check AWS CLI Version



```
root@ip-192-178-1-90:/home/ubuntu# aws --version
aws-cli/1.22.34 Python/3.10.4 Linux/5.15.0-1011-aws botocore/1.23.34
root@ip-192-178-1-90:/home/ubuntu#
```

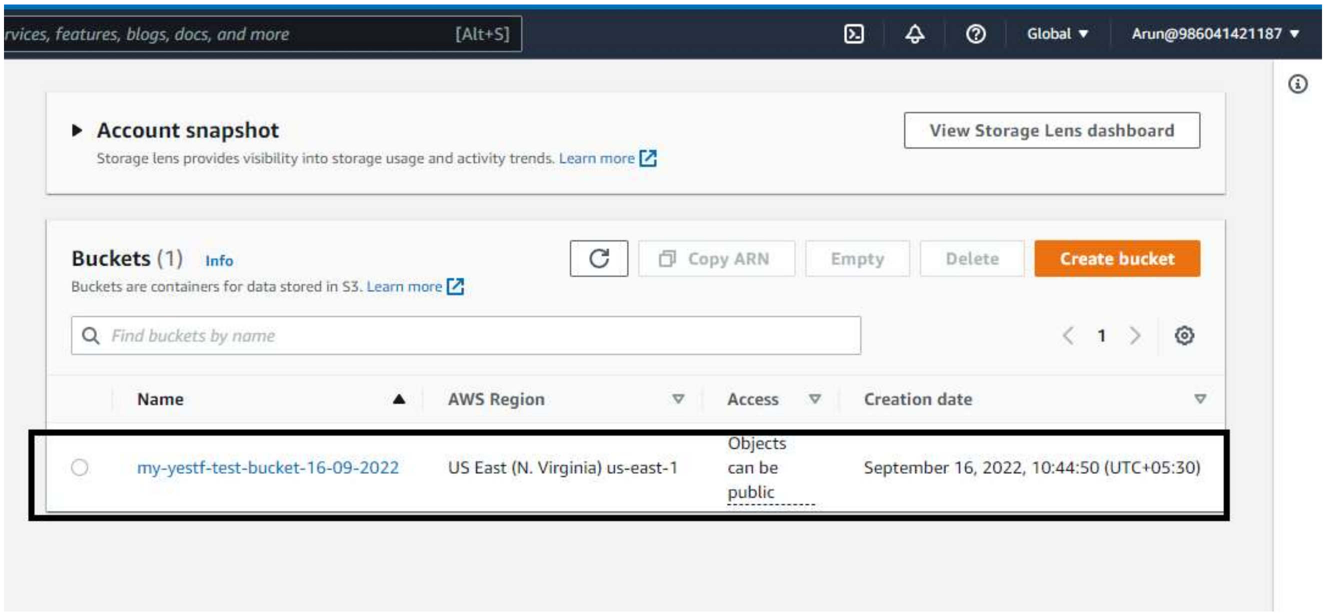
Then apply command aws s3 ls

we get the file in s3 bucket



```
root@ip-192-178-1-90:/home/ubuntu# aws s3 ls
2022-09-16 05:14:50 my-yestf-test-bucket-16-09-2022
root@ip-192-178-1-90:/home/ubuntu#
```

we can check this manually with management console.



Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
my-yestf-test-bucket-16-09-2022	US East (N. Virginia) us-east-1	Objects can be public	September 16, 2022, 10:44:50 (UTC+05:30)

Hence Successfully Performed