

Name: Devi Saran Chowdary

ID: 2000030236

Install AWS CLI

Open windows power shell

<https://chocolatey.org/install#individual>

run the commands

Open visual studio

Create IAM user in AWS act

Create user

Allow access for access key only

Add user

1 2 3 4 5

Set permissions

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy

Filter policies Search Showing 817 results

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	None
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	AWS managed	None

Cancel Previous Next: Tags

Feedback Looking for language selection? Find it in the new Unified Settings

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://194852177330.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key
▶	EPAM	AKIAS2XQYIWZBPKTZR6W	***** Show

Close

The screenshot shows a Visual Studio Code editor with a Terraform configuration file named `main.tf` open. The configuration defines an AWS provider with specific access and secret keys, and an S3 bucket resource. Below the editor, the terminal window displays the output of the `terraform init` command, showing the successful initialization of the backend and provider plugins.

```
provider "aws" {
  access_key = "AKIAS2XQYIWZBPKTZR6W"
  secret_key = "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"
}

resource "aws_s3_bucket" "my_bucket" {
  bucket = "varshu-s330498"
}
```

```
PS C:\Users\konda\Downloads\EPAM Terraform\1> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.48.0...
- Installed hashicorp/aws v4.48.0 (signed by HashiCorp)

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.
```

In terminal

terraform init

terraform plan

terraform apply

terraform destroy

terraform

```
    }

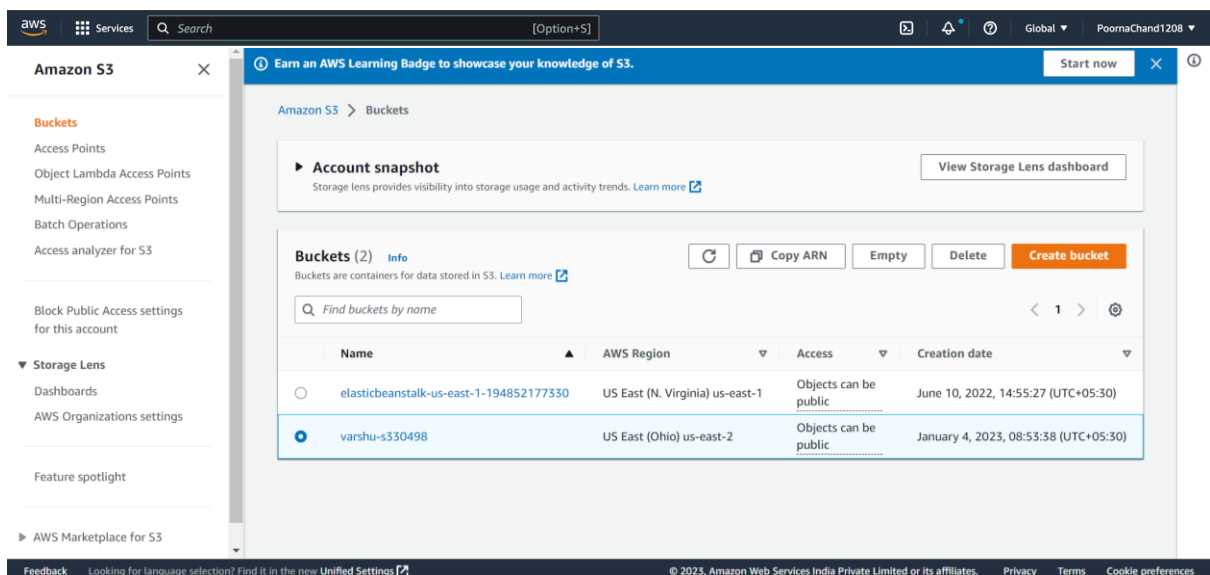
    Plan: 1 to add, 0 to change, 0 to destroy.

    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_s3_bucket.my_bucket: Creating...
    aws_s3_bucket.my_bucket: Creation complete after 6s [id=varshu-s330498]

    Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
    PS C:\Users\konda\Downloads\EPAM Terraform\1>
```



Mai.tf

```
provider "aws" {

    access_key="AKIAS2XQYIWZIXQRUIXA"

    secret_key="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"

}

resource "aws_s3_bucket" "my_bucket" {

    bucket="varshu-s330498"

}

resource "aws_instance" "my_instance" {
```

```
ami="amiid-30498"

instance_type=var.my_instance_type

tag{

    name="my-first-instance"

    id=123

}
```

```
}
```

Outputs.tf

```
output "my_bucket_name"{

    value=aws_s3_bucket.my_s3_bucket.bucket

}

output "aws_iam_user1_name"{

    value=aws_iam_user.my_user[1].name

}
```

Variables.tfvars

```
variable "my_instance_type"{

    type=string

    default="t2.micro"

    description="my 1st instance type"

}
```

Main.tf

```
provider "aws"{

    region="us-west-2"

    access_key="AKIAS2XQYIWZIXQRUIXA"

    secret_key="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"

}
```

```

terraform {
  backend "s3" {
    bucket = "19th2023s3bucket"

    key  = "main.tf"
    region = "us-west-2"
  }
}

```

In terminal

Aws configure

Terraform init

terraform init -migrate-state

The screenshot shows the Visual Studio Code interface with a Terraform configuration file open in the editor. The file contains the following code:

```

1 provider "aws" {
2   region="us-west-2"
3   access_keys="AKIAS2XQYIMZIXQRUIXA"
4   secret_key="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"
5 }
6
7
8 terraform {
9   backend "s3" {
10    bucket = "19th2023s3bucket"
11    key  = "main.tf"
12    region = "us-west-2"
13    dynamodb_table="state-lockng1a"
14  }
15 }

```

The terminal window at the bottom shows the output of the `terraform init` command:

```

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.48.0

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, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\konda\Downloads\EPAM Terraform\1>

```

Terraform apply