Name : Reena Qureshi

Reg No : 2023-BSE-052

Section : V-B

LAB 13

Task 1 — Create IAM Group and Output Details

In this task, you will create an IAM group named "developers" and output its details.

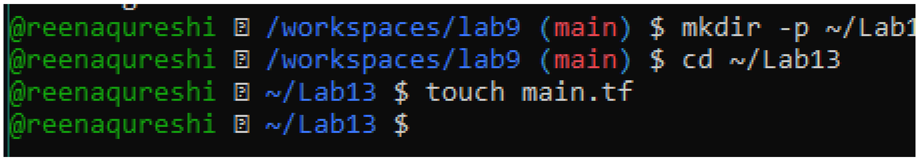
Create the initial project structure:

```
mkdir -p ~/Lab13
```

```
cd ~/Lab13
```

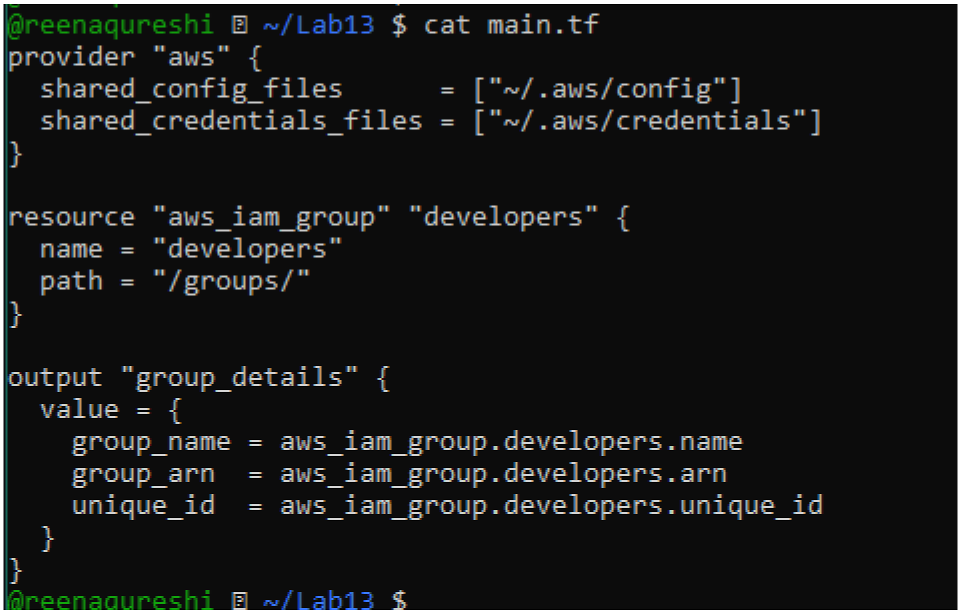
Create the main Terraform file:

```
touch main.tf
```



```
@reenaqureshi □ /workspaces/lab9 (main) $ mkdir -p ~/Lab1
@reenaqureshi □ /workspaces/lab9 (main) $ cd ~/Lab13
@reenaqureshi □ ~/Lab13 $ touch main.tf
@reenaqureshi □ ~/Lab13 $
```

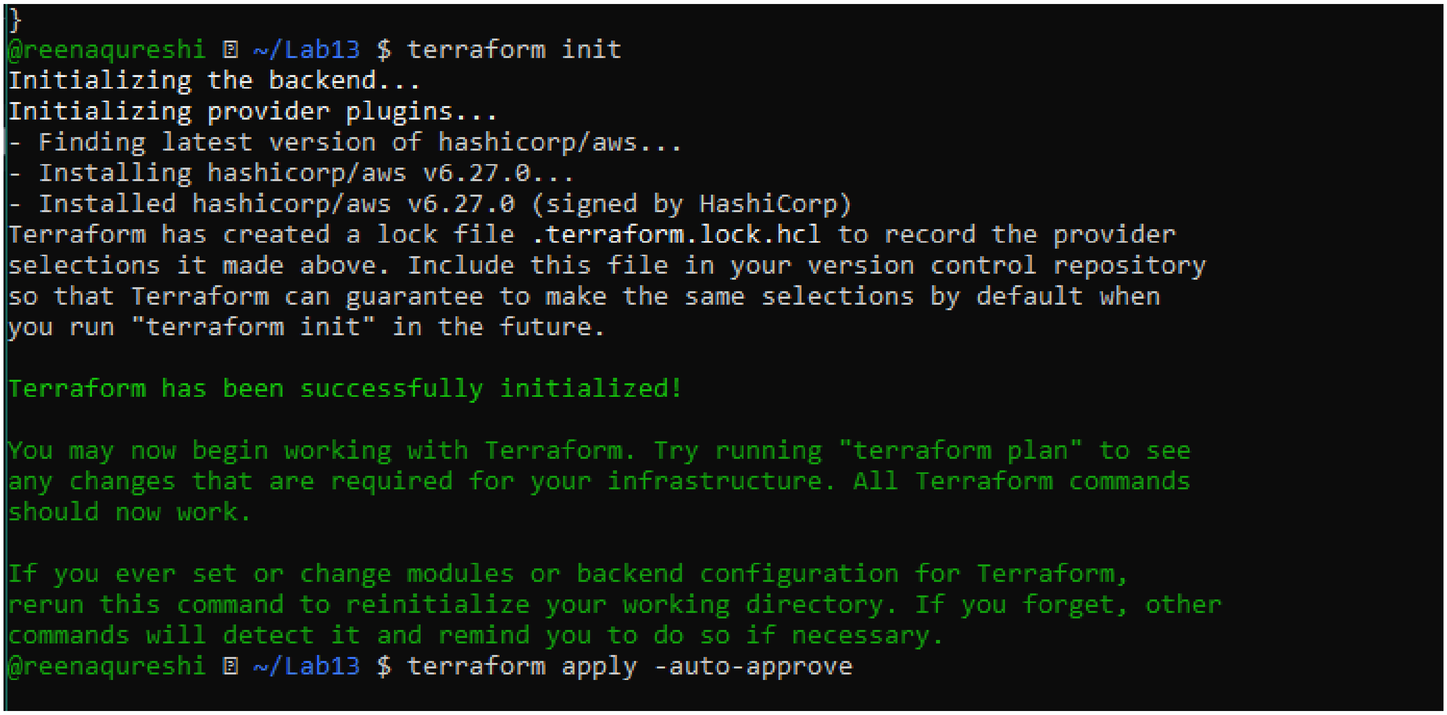
Create `main.tf` with AWS provider configuration:



```
@reenaqureshi □ ~/Lab13 $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/groups/"
}

output "group_details" {
  value = {
    group_name = aws_iam_group.developers.name
    group_arn  = aws_iam_group.developers.arn
    unique_id  = aws_iam_group.developers.unique_id
  }
}
@reenaqureshi □ ~/Lab13 $
```

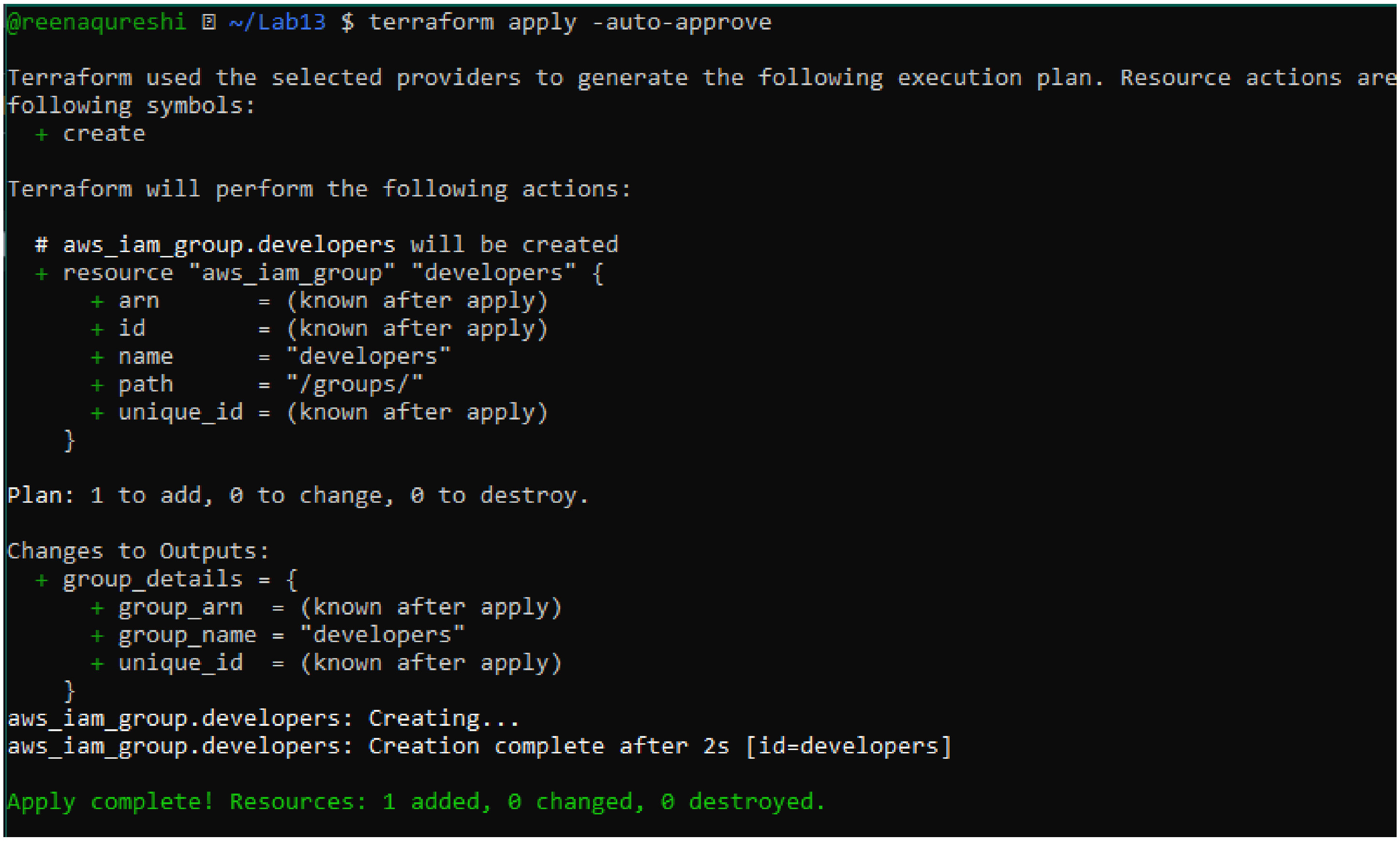
Initialize Terraform:

```
terraform init
```



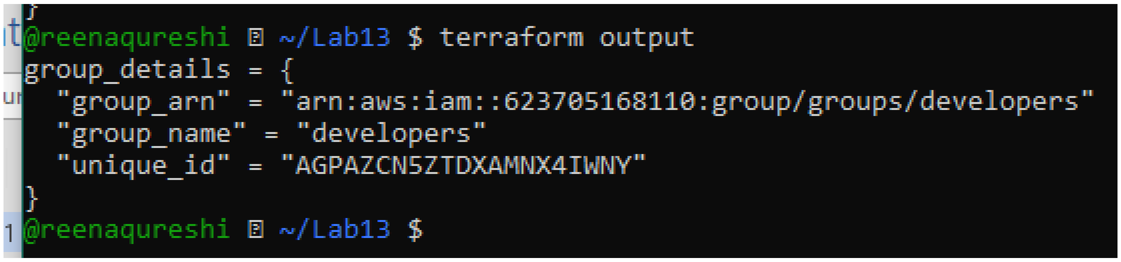
```
}
@reenaqureshi □ ~/Lab13 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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.
@reenaqureshi □ ~/Lab13 $ terraform apply -auto-approve
```

Apply the configuration:



```
@reenaqureshi □ ~/Lab13 $ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are
following symbols:
  + create

Terraform will perform the following actions:

  # aws_iam_group.developers will be created
  + resource "aws_iam_group" "developers" {
      + arn       = (known after apply)
      + id        = (known after apply)
      + name      = "developers"
      + path      = "/groups/"
      + unique_id = (known after apply)
    }

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

Changes to Outputs:
  + group_details = {
      + group_arn  = (known after apply)
      + group_name = "developers"
      + unique_id  = (known after apply)
    }
aws_iam_group.developers: Creating...
aws_iam_group.developers: Creation complete after 2s [id=developers]

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

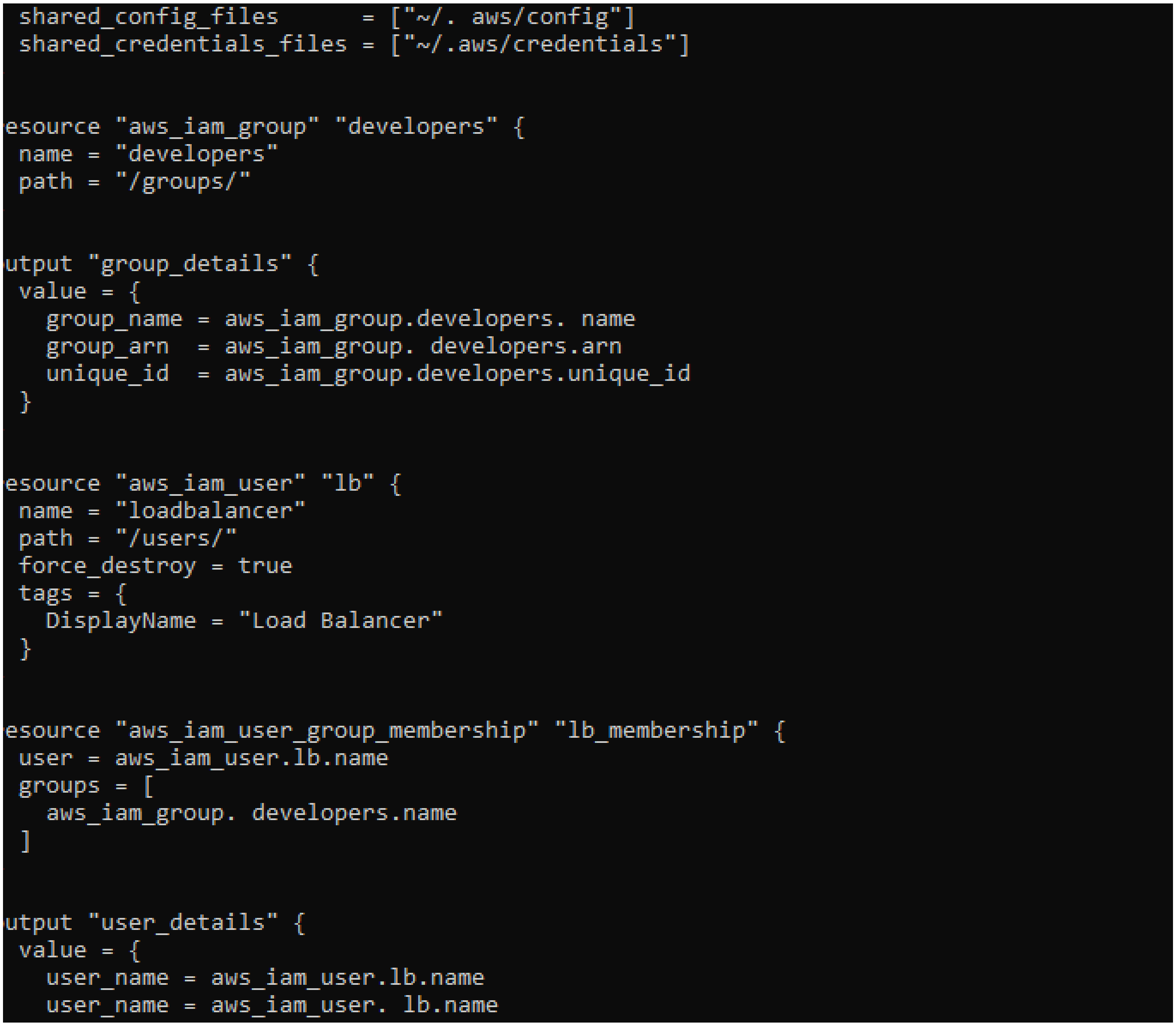
Display the output:



```
}
@reenaqureshi ▯ ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
@reenaqureshi ▯ ~/Lab13 $
```

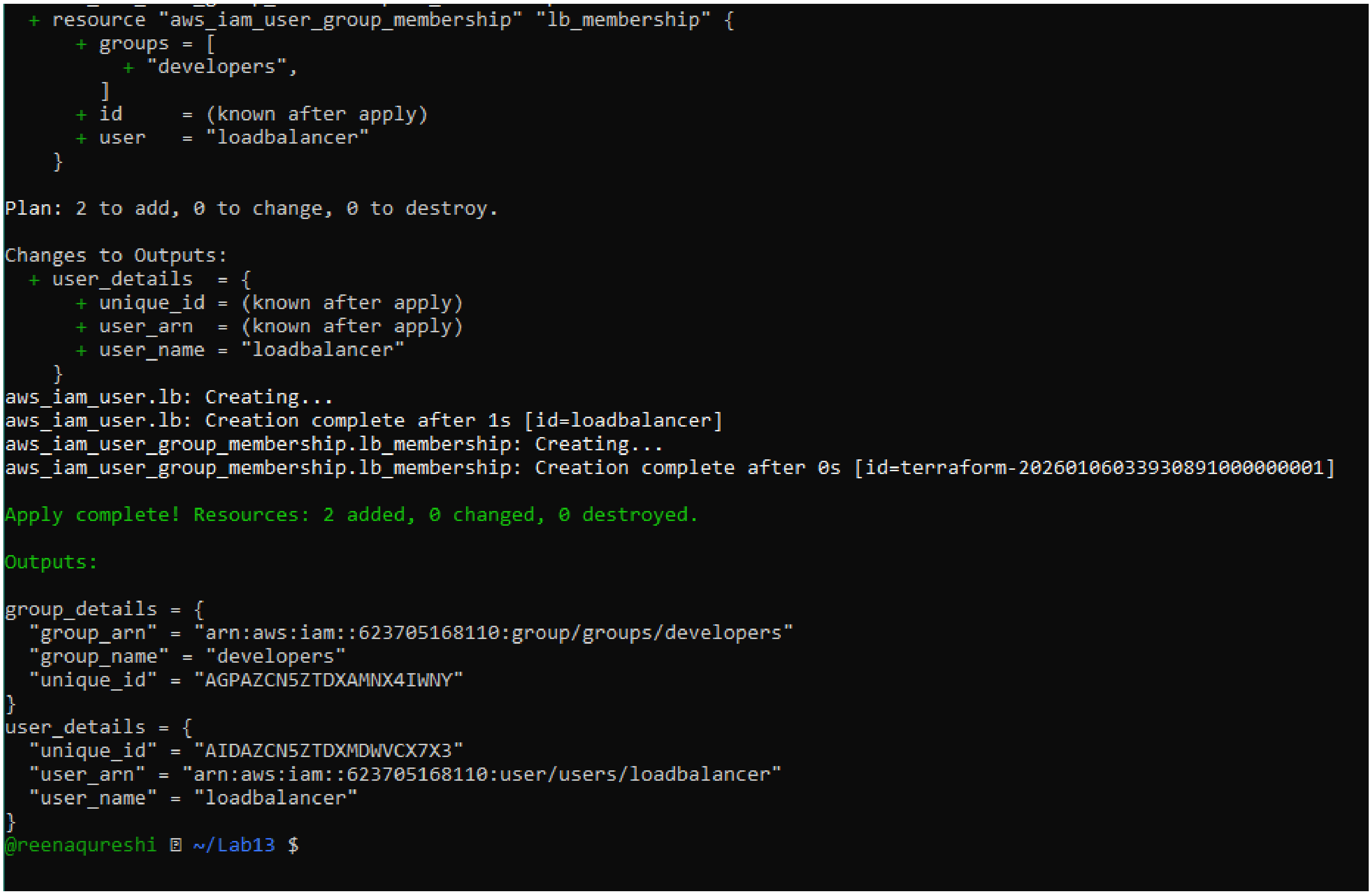
Task 2 — Create IAM User with Group Membership

Update `main.tf` to add the IAM user resource:



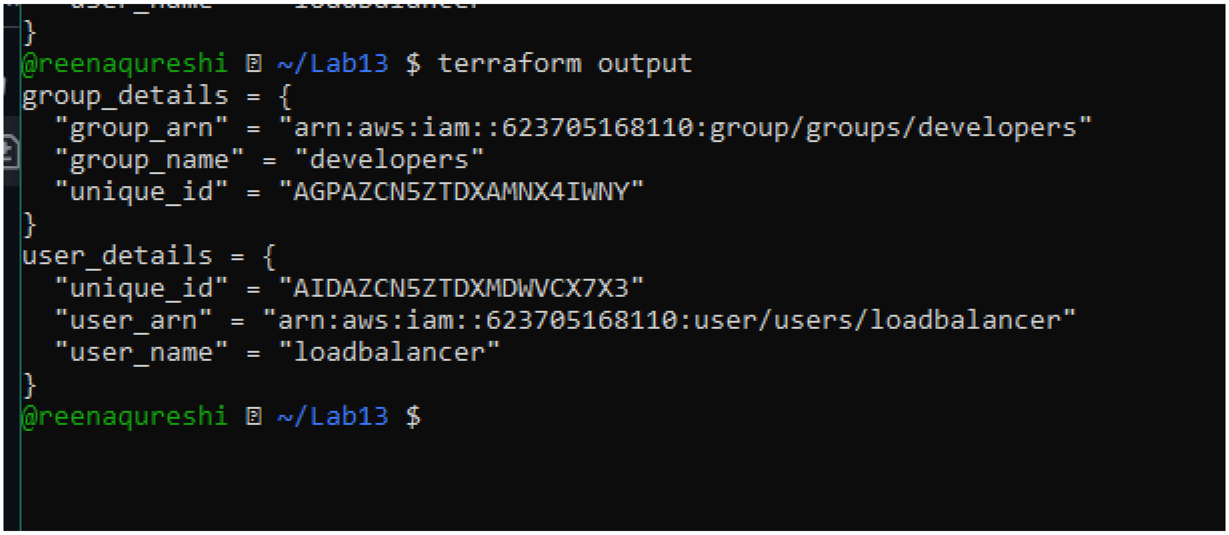
```
 shared_config_files      = ["~/. aws/config"]
 shared_credentials_files = ["~/.aws/credentials"]


esource "aws_iam_group" "developers" {
 name = "developers"
 path = "/groups/"


utput "group_details" {
 value = {
   group_name = aws_iam_group.developers. name
   group_arn  = aws_iam_group. developers.arn
   unique_id  = aws_iam_group.developers.unique_id
 }


esource "aws_iam_user" "lb" {
 name = "loadbalancer"
 path = "/users/"
 force_destroy = true
 tags = {
   DisplayName = "Load Balancer"
 }


esource "aws_iam_user_group_membership" "lb_membership" {
 user = aws_iam_user.lb.name
 groups = [
   aws_iam_group. developers.name
 ]


utput "user_details" {
 value = {
   user_name = aws_iam_user.lb.name
   user_name = aws_iam_user. lb.name
```

Apply the configuration:



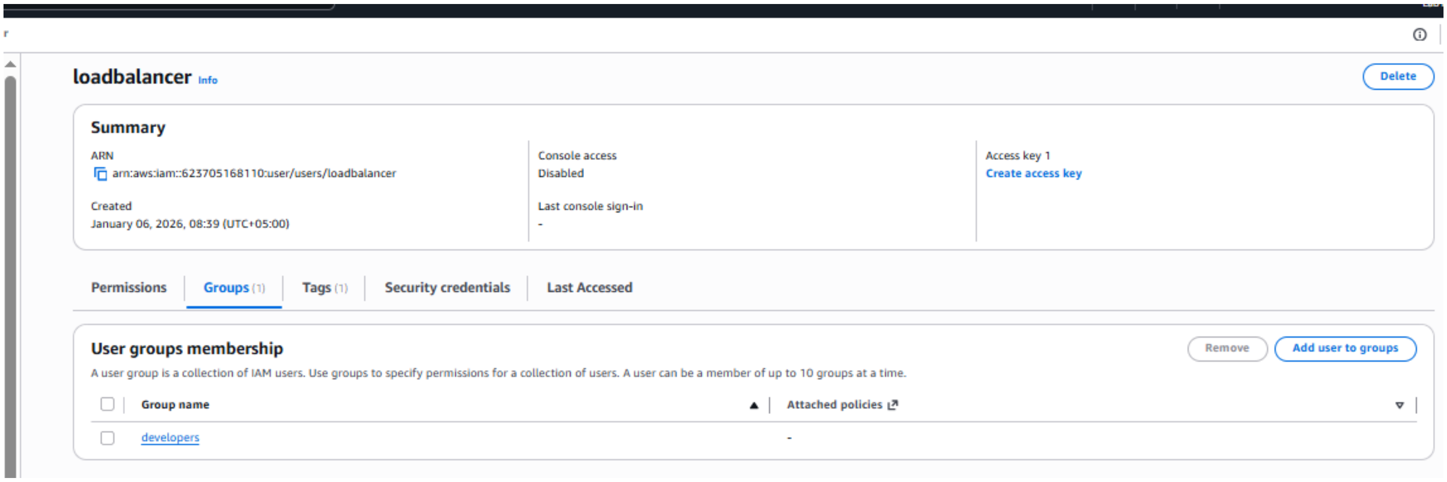
```
  + resource "aws_iam_user_group_membership" "lb_membership" {
      + groups = [
          + "developers",
        ]
      + id     = (known after apply)
      + user   = "loadbalancer"
    }

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

Changes to Outputs:
  + user_details  = {
      + unique_id = (known after apply)
      + user_arn  = (known after apply)
      + user_name = "loadbalancer"
    }
aws_iam_user.lb: Creating...
aws_iam_user.lb: Creation complete after 1s [id=loadbalancer]
aws_iam_user_group_membership.lb_membership: Creating...
aws_iam_user_group_membership.lb_membership: Creation complete after 0s [id=terraform-20260106033930891000000001]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi □ ~/Lab13 $
```

Display the outputs:



```
}
@reenaqureshi □ ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi □ ~/Lab13 $
```

Verify on AWS:



Summary

ARN
arn:aws:iam::623705168110:user/users/loadbalancer

Created
January 06, 2026, 08:39 (UTC+05:00)

Console access
Disabled

Last console sign-in
-

Access key 1
Create access key

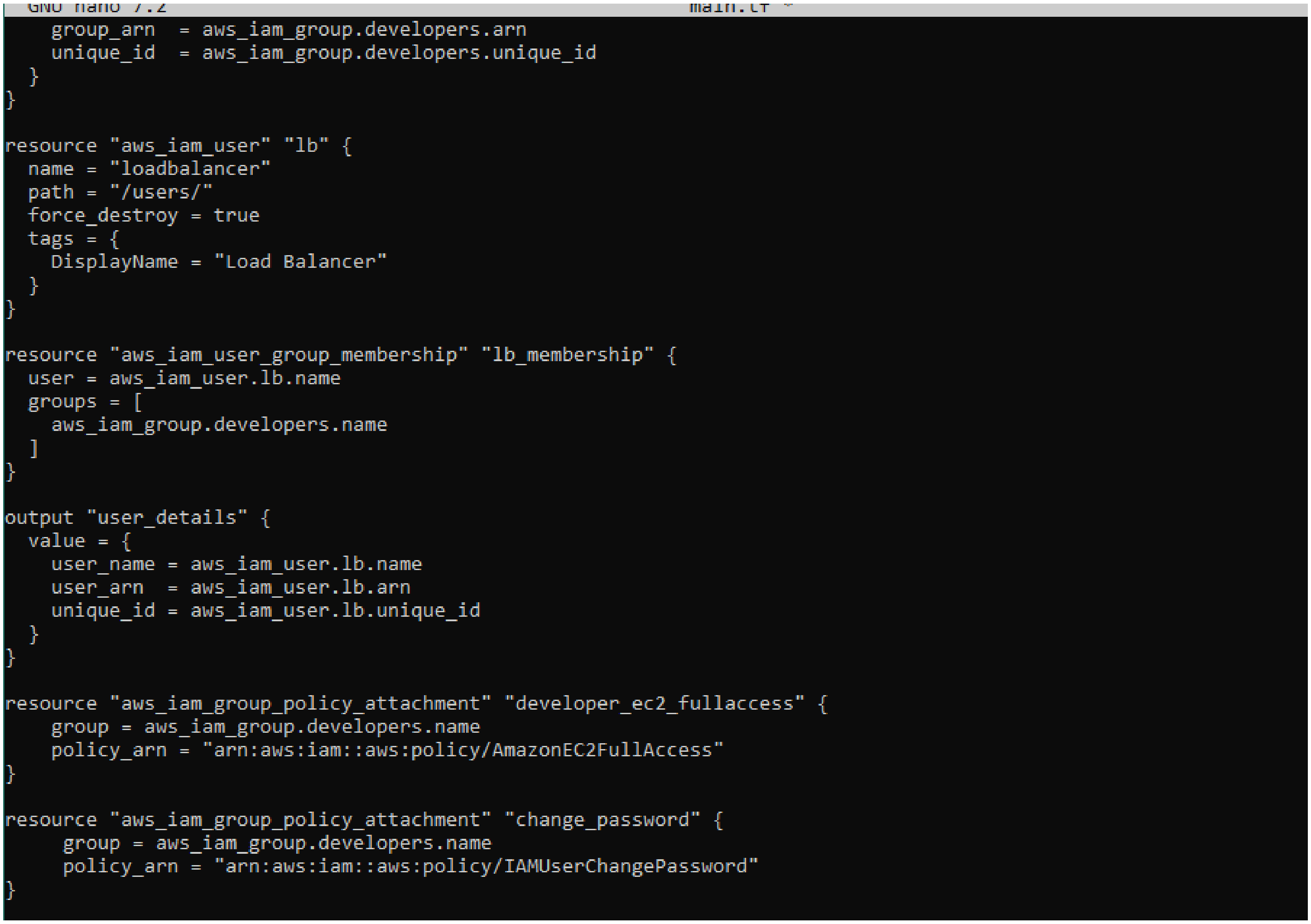
User groups membership

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users. A user can be a member of up to 10 groups at a time.

□	Group name	Attached policies
□	developers	-

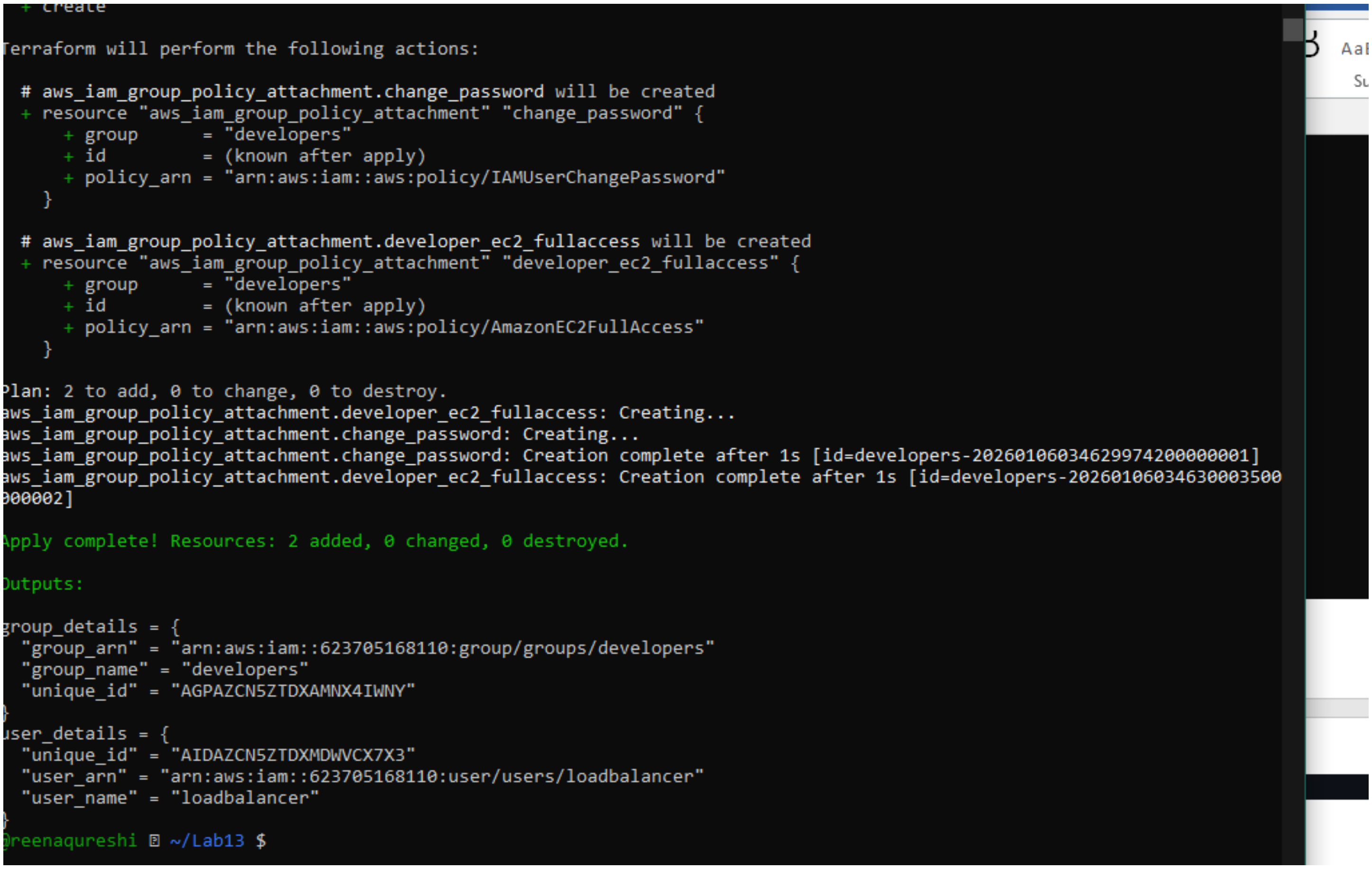
Task 3 — Attach Policies to IAM Group

Update `main.tf` to add policy attachments:



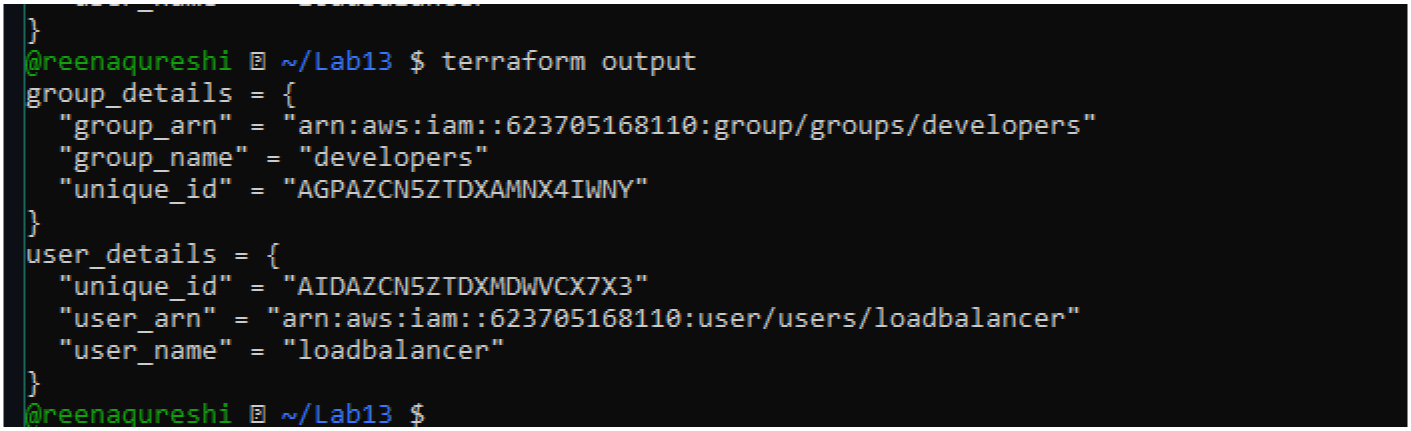
```
  GNU nano 7.2                                              main.tf *
    group_arn   = aws_iam_group.developers.arn
    unique_id   = aws_iam_group.developers.unique_id
  }
}

resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/users/"
  force_destroy = true
  tags = {
    DisplayName = "Load Balancer"
  }
}

resource "aws_iam_user_group_membership" "lb_membership" {
  user = aws_iam_user.lb.name
  groups = [
    aws_iam_group.developers.name
  ]
}

output "user_details" {
  value = {
    user_name = aws_iam_user.lb.name
    user_arn  = aws_iam_user.lb.arn
    unique_id = aws_iam_user.lb.unique_id
  }
}

resource "aws_iam_group_policy_attachment" "developer_ec2_fullaccess" {
    group = aws_iam_group.developers.name
    policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

resource "aws_iam_group_policy_attachment" "change_password" {
     group = aws_iam_group.developers.name
     policy_arn = "arn:aws:iam::aws:policy/IAMUserChangePassword"
}
```

Apply the configuration:



```
  + create

Terraform will perform the following actions:

  # aws_iam_group_policy_attachment.change_password will be created
  + resource "aws_iam_group_policy_attachment" "change_password" {
      + group      = "developers"
      + id         = (known after apply)
      + policy_arn = "arn:aws:iam::aws:policy/IAMUserChangePassword"
    }

  # aws_iam_group_policy_attachment.developer_ec2_fullaccess will be created
  + resource "aws_iam_group_policy_attachment" "developer_ec2_fullaccess" {
      + group      = "developers"
      + id         = (known after apply)
      + policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
    }

Plan: 2 to add, 0 to change, 0 to destroy.
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creating...
aws_iam_group_policy_attachment.change_password: Creating...
aws_iam_group_policy_attachment.change_password: Creation complete after 1s [id=developers-20260106034629974200000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-20260106034630003500
000002]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi □ ~/Lab13 $
```

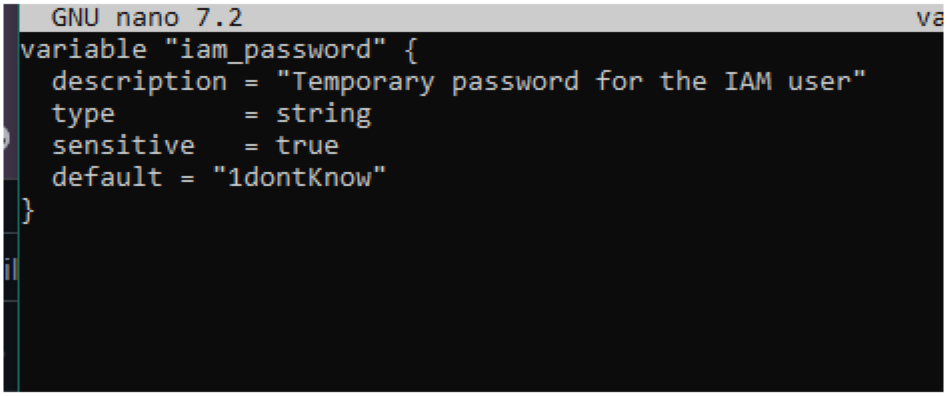
Output:



```
}
@reenaqureshi □ ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi □ ~/Lab13 $
```

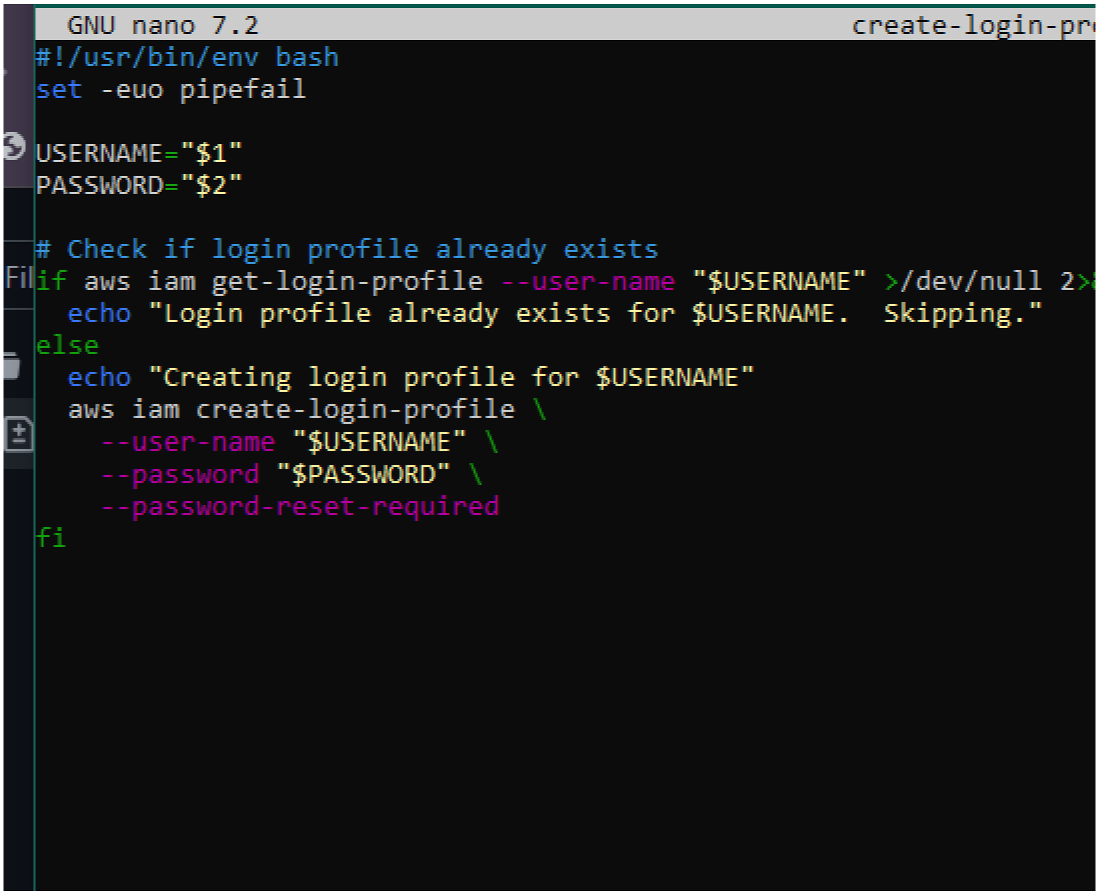
Task 4 — Create Login Profile for IAM User

Create `variables.tf` file:



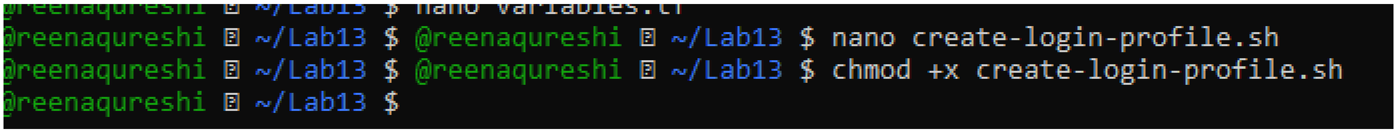
```
  GNU nano 7.2                                                va
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default = "1dontKnow"
}
```

Create the bash script `create-login-profile.sh`:



```
  GNU nano 7.2                                         create-login-pr
#!/usr/bin/env bash
set -euo pipefail

USERNAME="$1"
PASSWORD="$2"

# Check if login profile already exists
if aws iam get-login-profile --user-name "$USERNAME" >/dev/null 2>
  echo "Login profile already exists for $USERNAME.  Skipping."
else
  echo "Creating login profile for $USERNAME"
  aws iam create-login-profile \
    --user-name "$USERNAME" \
    --password "$PASSWORD" \
    --password-reset-required
fi
```

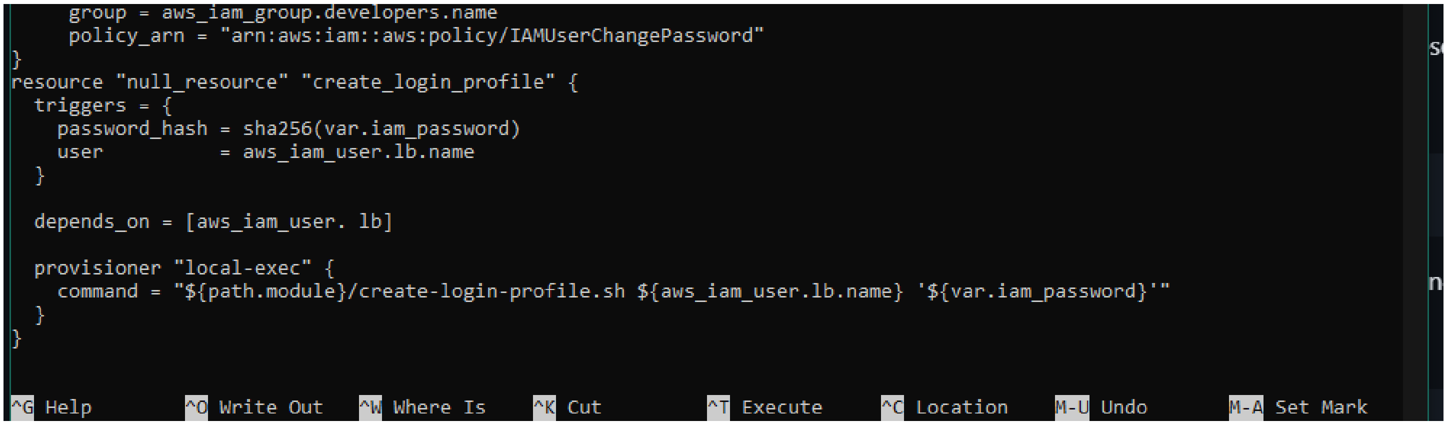
Make the script executable:



```
@reenaqureshi ▯ ~/Lab13 $ nano variables.tf
@reenaqureshi ▯ ~/Lab13 $ @reenaqureshi ▯ ~/Lab13 $ nano create-login-profile.sh
@reenaqureshi ▯ ~/Lab13 $ @reenaqureshi ▯ ~/Lab13 $ chmod +x create-login-profile.sh
@reenaqureshi ▯ ~/Lab13 $
```

Update `main.tf` to add the null_resource provisioner:

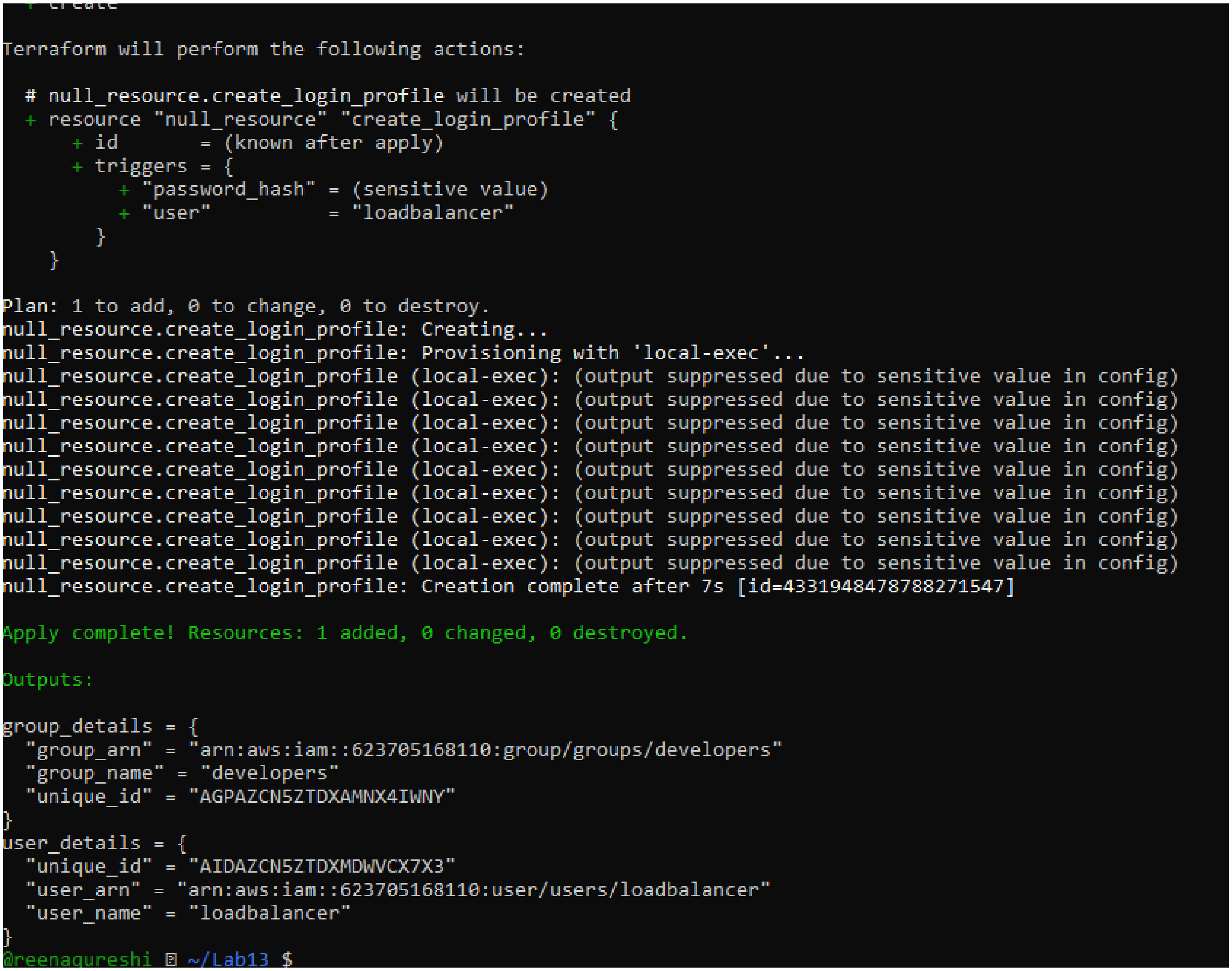
Add this resource after the user creation:



```
    group = aws_iam_group.developers.name
    policy_arn = "arn:aws:iam::aws:policy/IAMUserChangePassword"
}
resource "null_resource" "create_login_profile" {
  triggers = {
    password_hash = sha256(var.iam_password)
    user          = aws_iam_user.lb.name
  }

  depends_on = [aws_iam_user. lb]

  provisioner "local-exec" {
    command = "${path.module}/create-login-profile.sh ${aws_iam_user.lb.name} '${var.iam_password}'"
  }
}

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
```

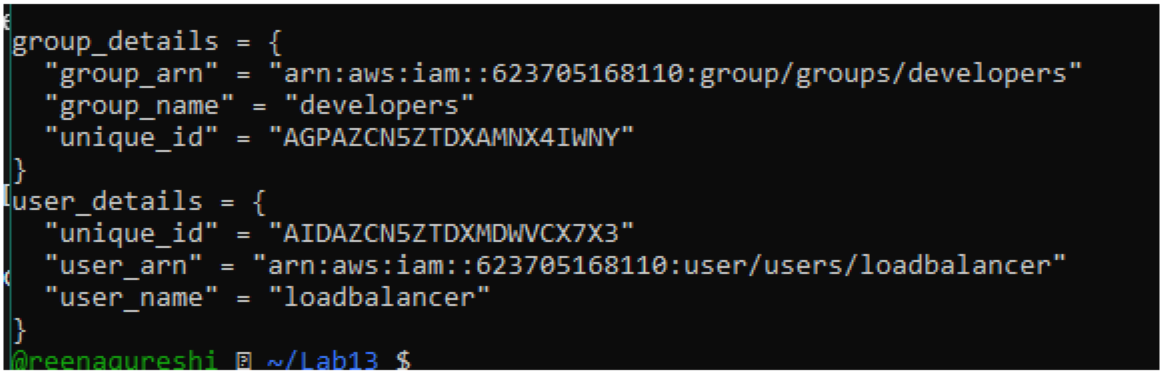
Apply the configuration with a custom password:

```
terraform apply -auto-approve -var="iam_password=MySecurePass123!"
```



```
  + create

Terraform will perform the following actions:

  # null_resource.create_login_profile will be created
  + resource "null_resource" "create_login_profile" {
      + id       = (known after apply)
      + triggers = {
          + "password_hash" = (sensitive value)
          + "user"          = "loadbalancer"
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.
null_resource.create_login_profile: Creating...
null_resource.create_login_profile: Provisioning with 'local-exec'...
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile: Creation complete after 7s [id=4331948478788271547]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi ➜ ~/Lab13 $
```

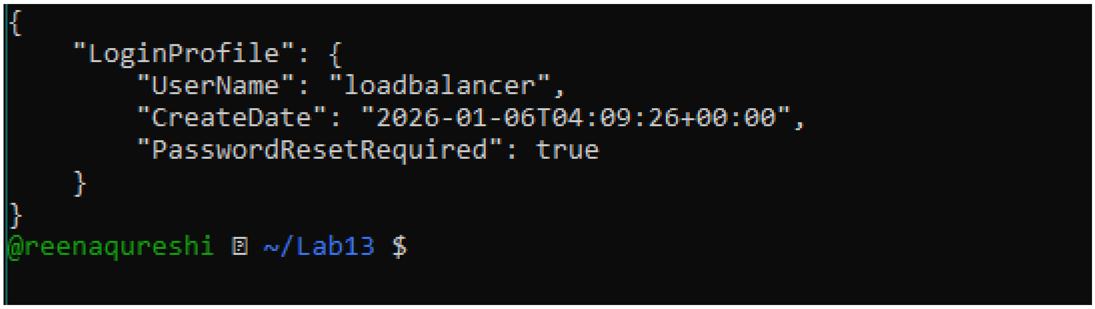
Output:



```
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi □ ~/Lab13 $
```

Verify login profile creation:

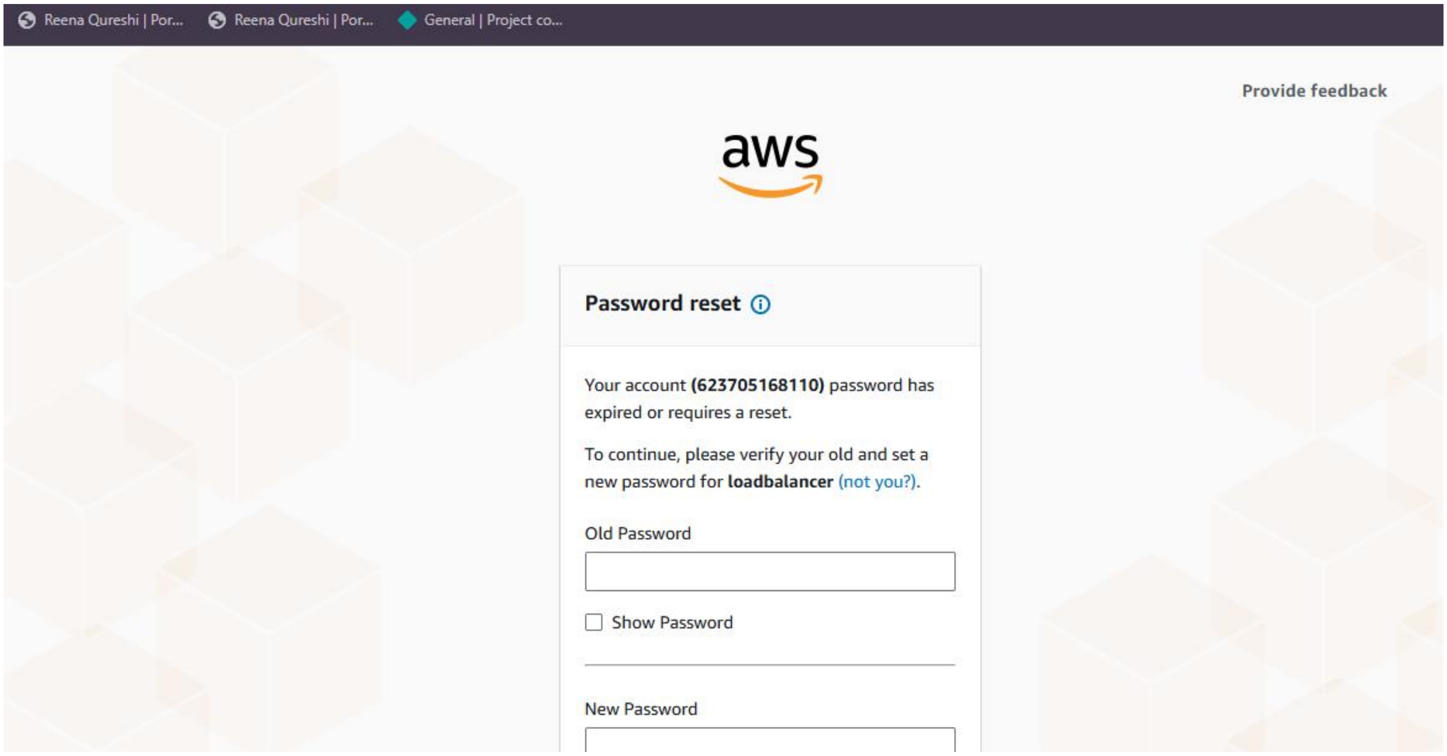
```
aws iam get-login-profile --user-name loadbalancer
```



```
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-06T04:09:26+00:00",
        "PasswordResetRequired": true
    }
}
@reenaqureshi □ ~/Lab13 $
```

,

Test login in AWS Console:



Password reset

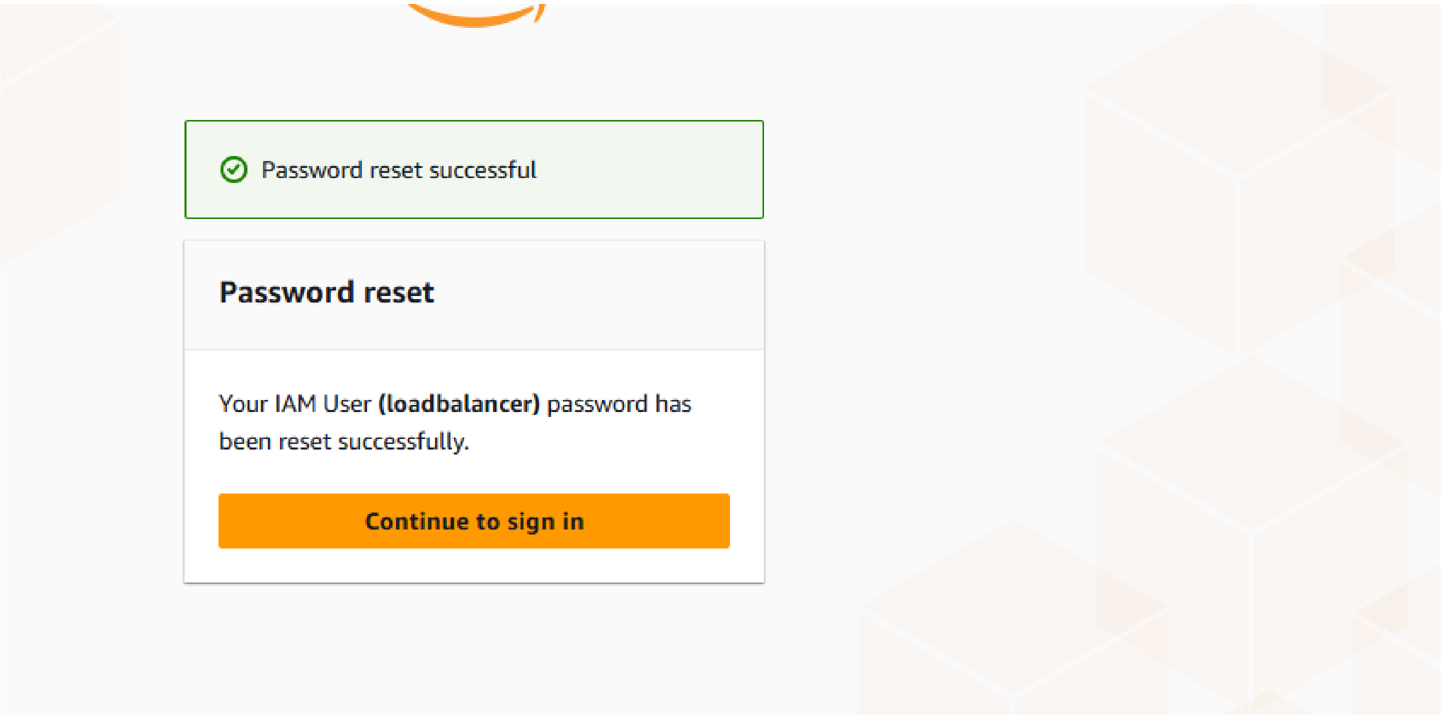
Your account **(623705168110)** password has expired or requires a reset.

To continue, please verify your old and set a new password for **loadbalancer** (not you?).

Old Password

□ Show Password

New Password



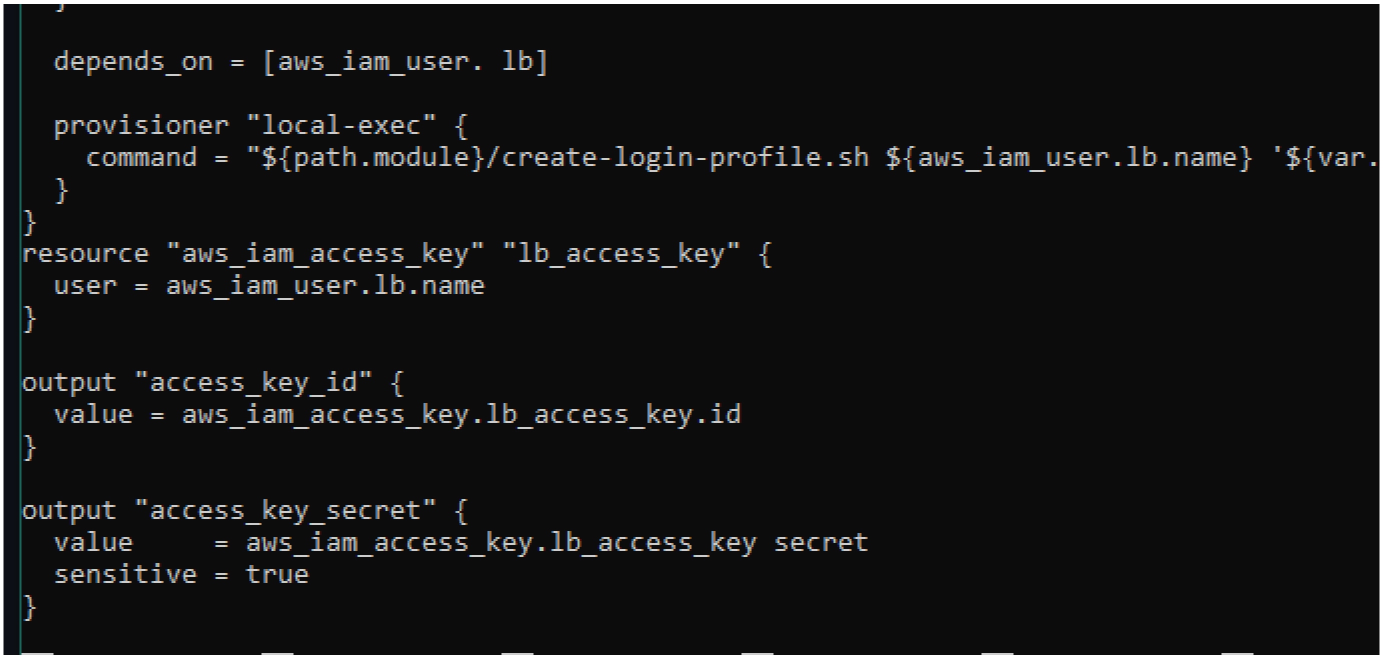
Password reset successful

Password reset

Your IAM User **(loadbalancer)** password has been reset successfully.

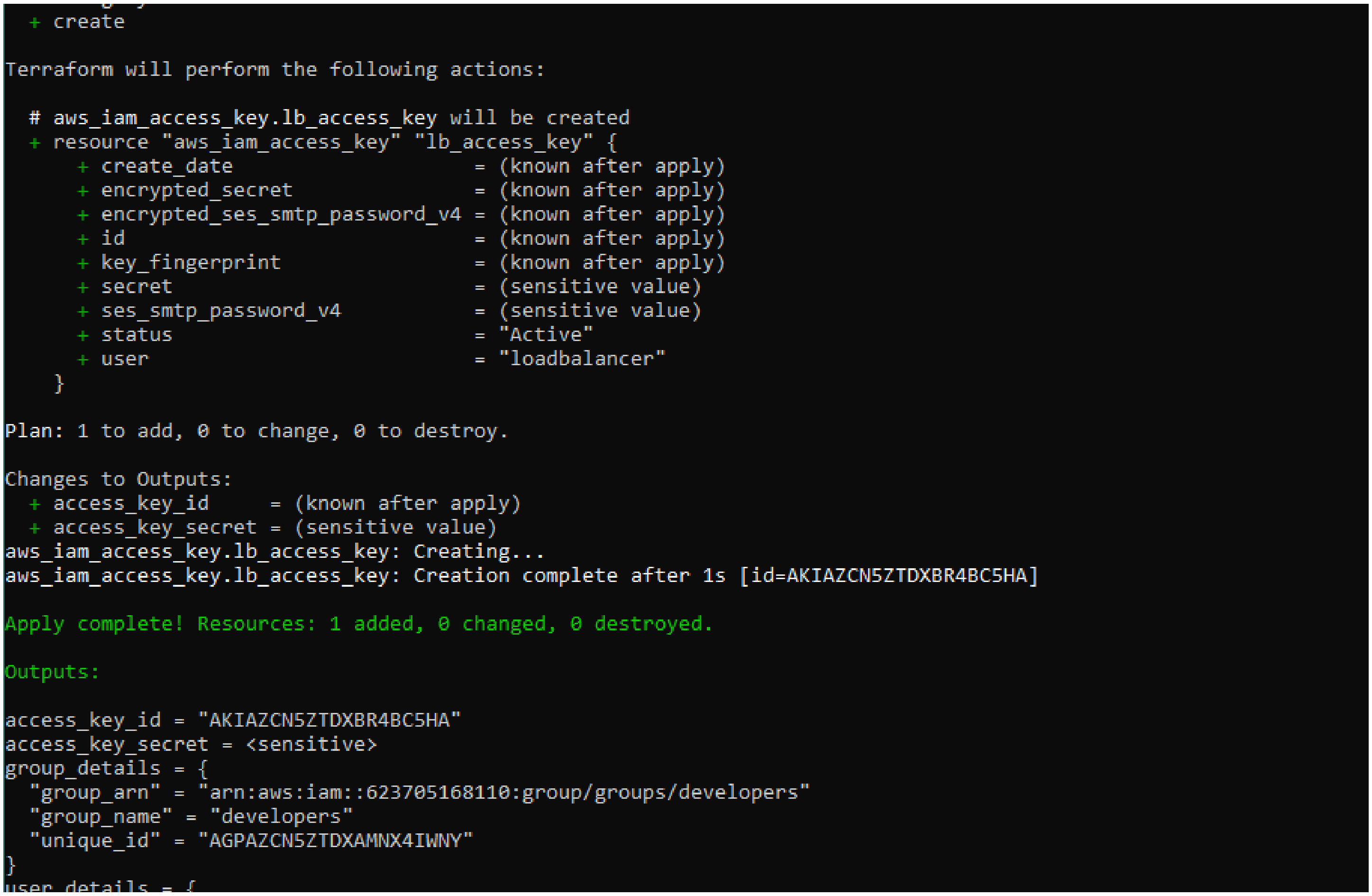
Task 5 — Generate Access Keys for IAM User

Update `main.tf` to add access key resource and outputs:



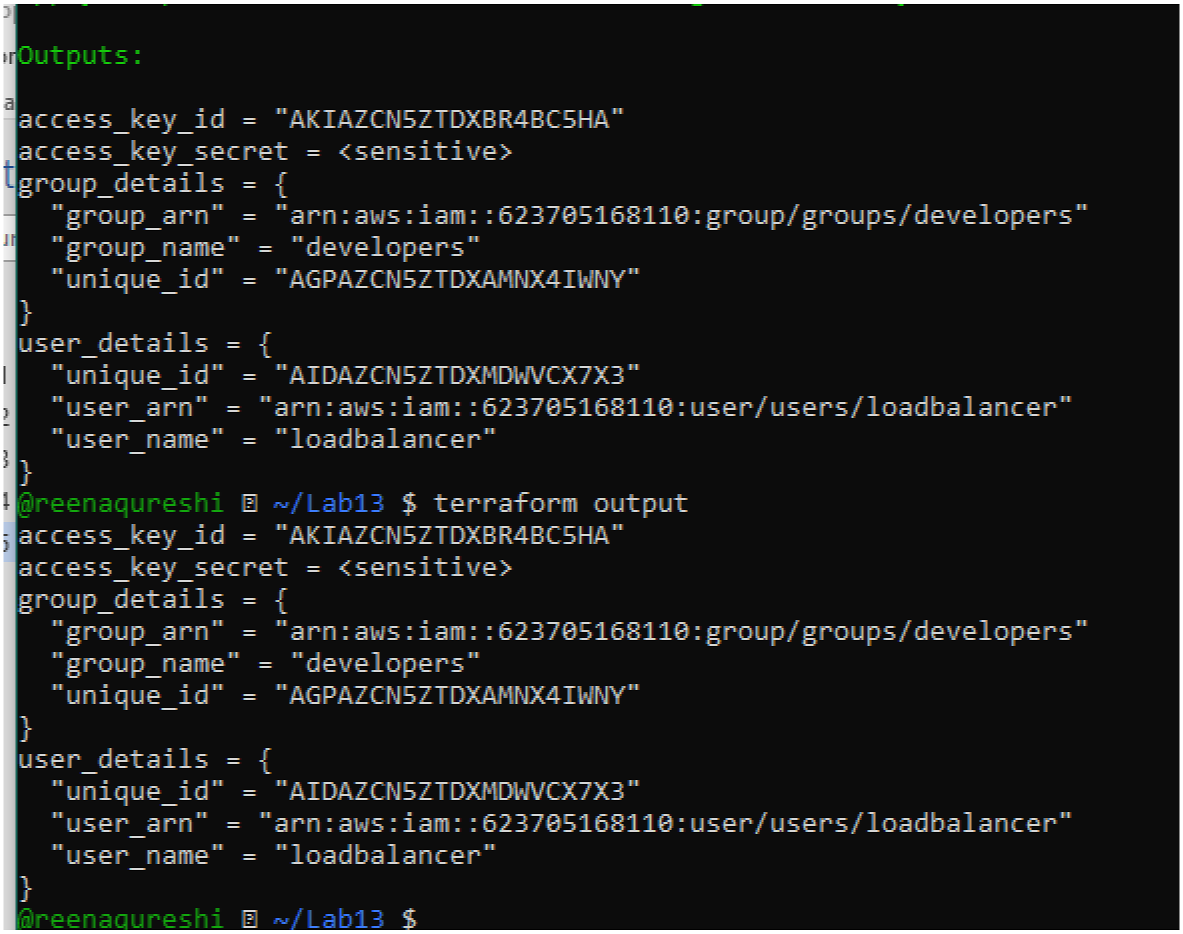
```
  depends_on = [aws_iam_user. lb]

  provisioner "local-exec" {
    command = "${path.module}/create-login-profile.sh ${aws_iam_user.lb.name} '${var.
  }
}
resource "aws_iam_access_key" "lb_access_key" {
  user = aws_iam_user.lb.name
}

output "access_key_id" {
  value = aws_iam_access_key.lb_access_key.id
}

output "access_key_secret" {
  value     = aws_iam_access_key.lb_access_key secret
  sensitive = true
}
```

Apply the configuration:



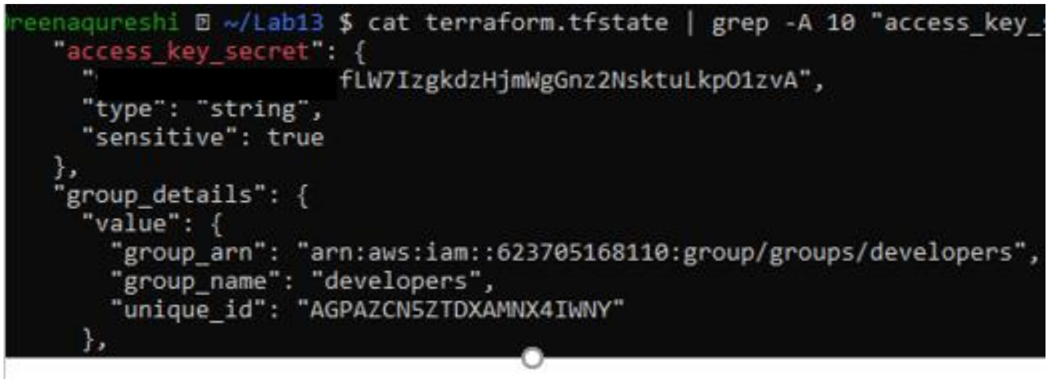
```
  + create

Terraform will perform the following actions:

  # aws_iam_access_key.lb_access_key will be created
  + resource "aws_iam_access_key" "lb_access_key" {
      + create_date                    = (known after apply)
      + encrypted_secret               = (known after apply)
      + encrypted_ses_smtp_password_v4 = (known after apply)
      + id                             = (known after apply)
      + key_fingerprint                = (known after apply)
      + secret                         = (sensitive value)
      + ses_smtp_password_v4           = (sensitive value)
      + status                         = "Active"
      + user                           = "loadbalancer"
    }

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

Changes to Outputs:
  + access_key_id     = (known after apply)
  + access_key_secret = (sensitive value)
aws_iam_access_key.lb_access_key: Creating...
aws_iam_access_key.lb_access_key: Creation complete after 1s [id=AKIAZCN5ZTDXBR4BC5HA]

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

Outputs:

access_key_id = "AKIAZCN5ZTDXBR4BC5HA"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
```

Display outputs:



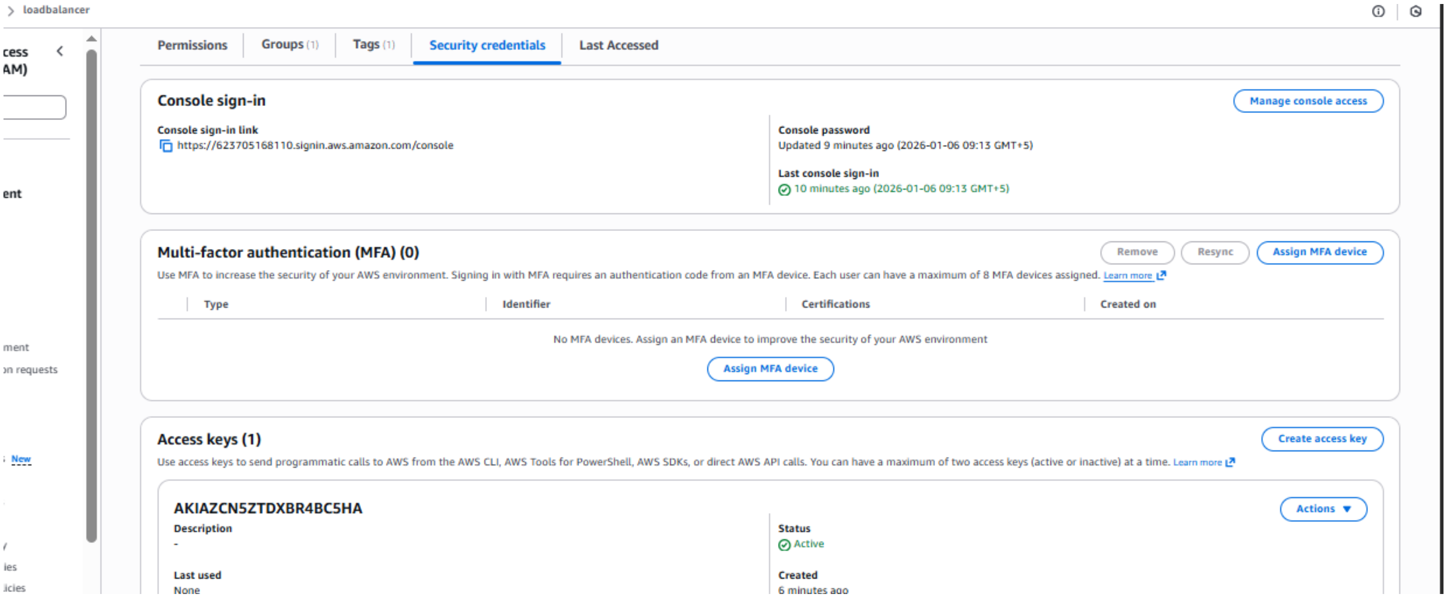
```
Outputs:

access_key_id = "AKIAZCN5ZTDXBR4BC5HA"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi ▯ ~/Lab13 $ terraform output
access_key_id = "AKIAZCN5ZTDXBR4BC5HA"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi ▯ ~/Lab13 $
```

View the secret in terraform state:



```
reenaqureshi ▯ ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_
    "access_key_secret": {
      "[redacted]fLW7IzgkdzHjmWgGnz2NsktuLkpO1zvA",
      "type": "string",
      "sensitive": true
    },
    "group_details": {
      "value": {
        "group_arn": "arn:aws:iam::623705168110:group/groups/developers",
        "group_name": "developers",
        "unique_id": "AGPAZCN5ZTDXAMNX4IWNY"
      },
```

Verify access key in AWS Console:



Console sign-in

Console sign-in link
https://623705168110.signin.aws.amazon.com/console

Console password
Updated 9 minutes ago (2026-01-06 09:13 GMT+5)

Last console sign-in
10 minutes ago (2026-01-06 09:13 GMT+5)

Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Type	Identifier	Certifications	Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Assign MFA device

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

AKIAZCN5ZTDXBR4BC5HA

Description
-

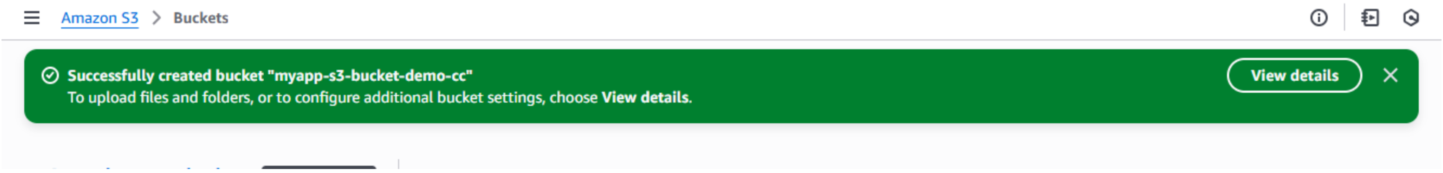
Status
Active

Last used
None

Created
6 minutes ago

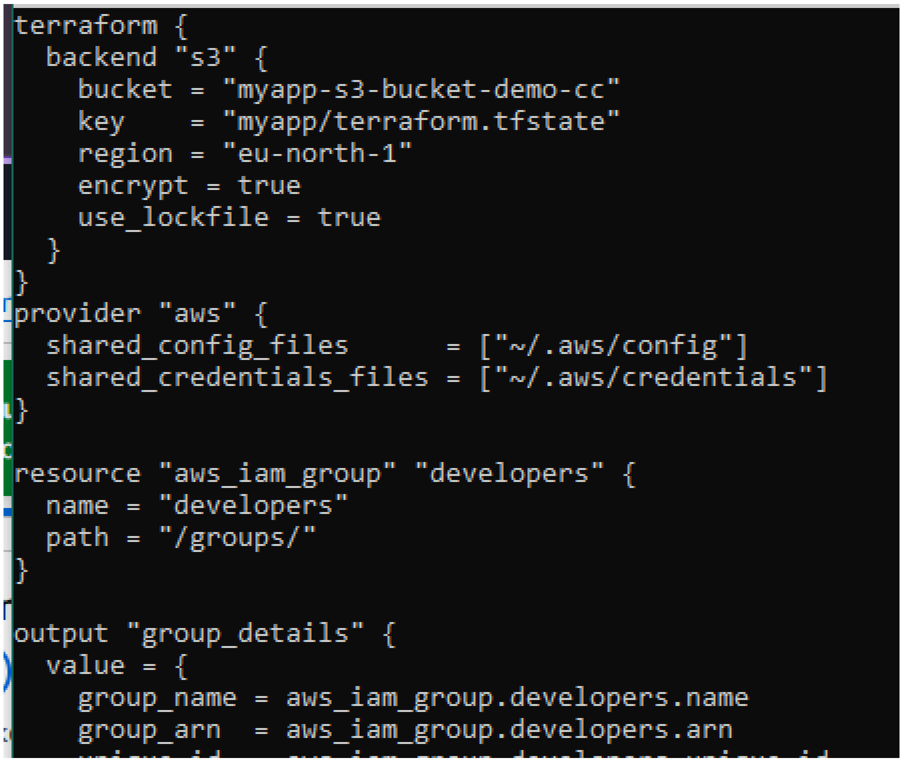
Task 6 — Implement Terraform Remote State with S3

Create S3 bucket in AWS Console:



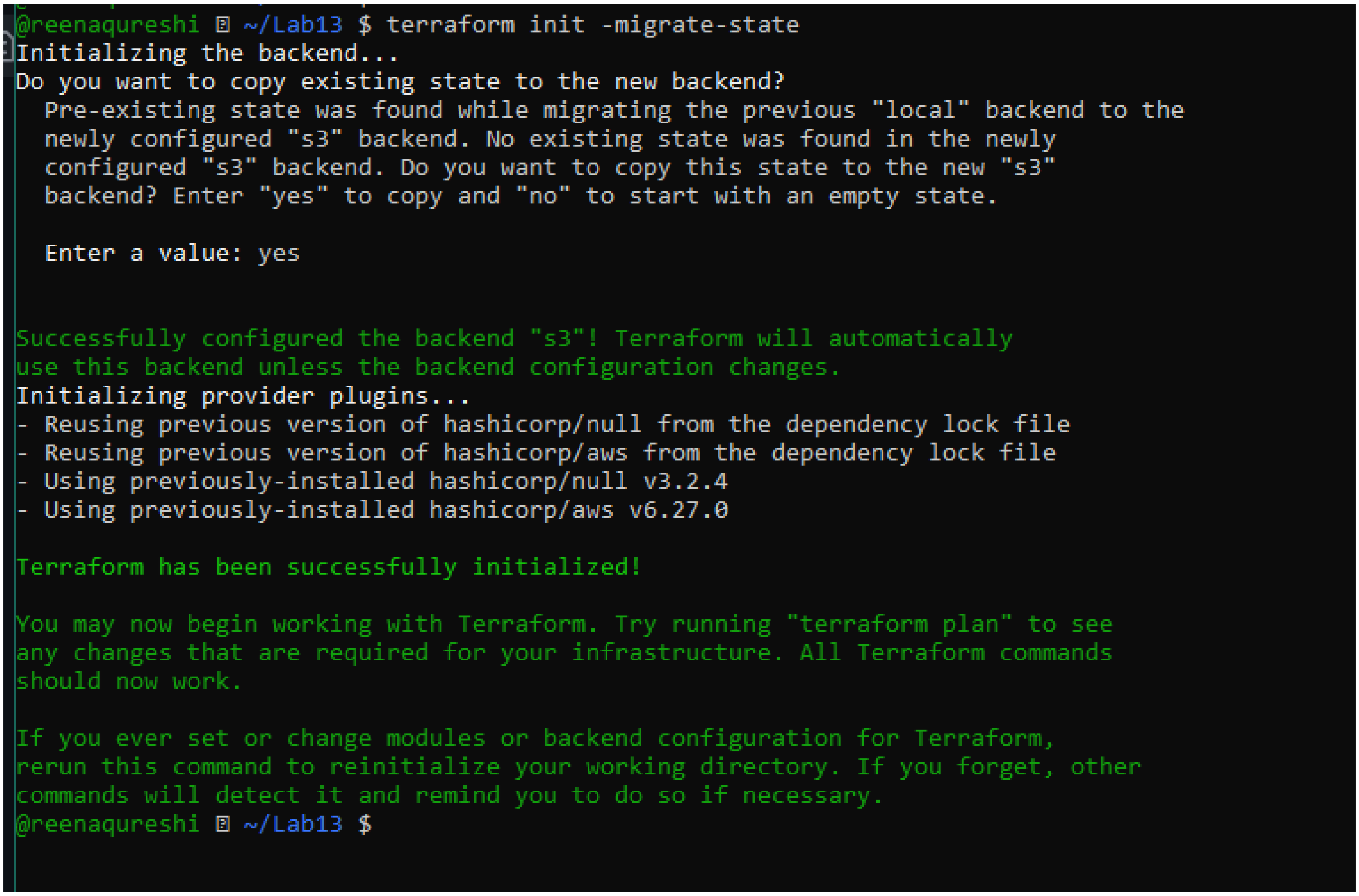
Successfully created bucket "myapp-s3-bucket-demo-cc"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Update `main.tf` to add S3 backend configuration:



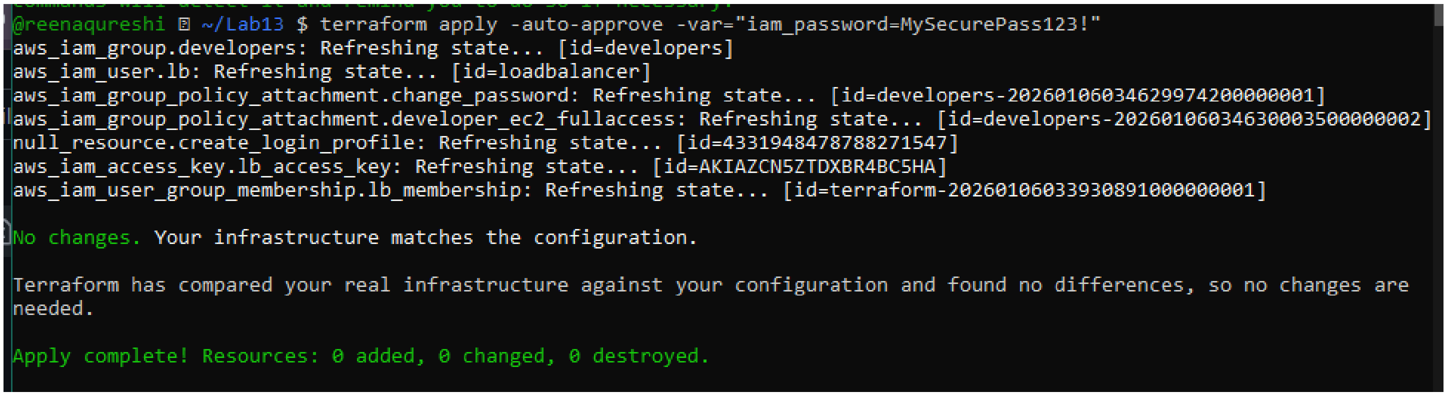
```
terraform {
  backend "s3" {
    bucket = "myapp-s3-bucket-demo-cc"
    key    = "myapp/terraform.tfstate"
    region = "eu-north-1"
    encrypt = true
    use_lockfile = true
  }
}
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/groups/"
}

output "group_details" {
  value = {
    group_name = aws_iam_group.developers.name
    group_arn  = aws_iam_group.developers.arn
    unique_id  = aws_iam_group.developers.unique_id
```

Reinitialize Terraform with the backend:



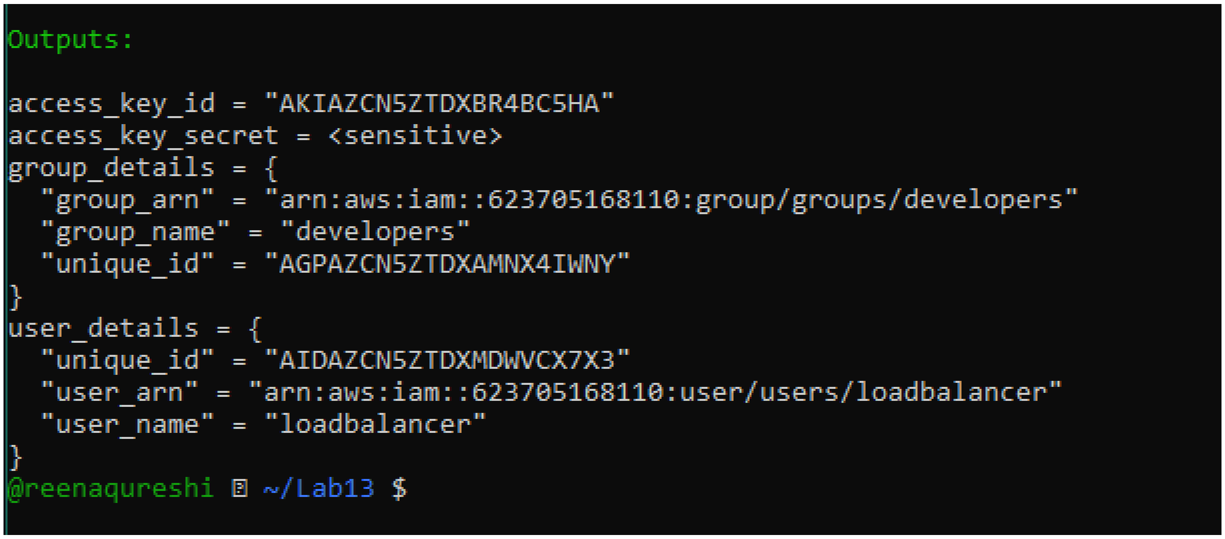
```
@reenaqureshi ▯ ~/Lab13 $ terraform init -migrate-state
Initializing the backend...
Do you want to copy existing state to the new backend?
  Pre-existing state was found while migrating the previous "local" backend to the
  newly configured "s3" backend. No existing state was found in the newly
  configured "s3" backend. Do you want to copy this state to the new "s3"
  backend? Enter "yes" to copy and "no" to start with an empty state.

  Enter a value: yes


Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/null v3.2.4
- Using previously-installed hashicorp/aws v6.27.0

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.
@reenaqureshi ▯ ~/Lab13 $
```

Apply the configuration:



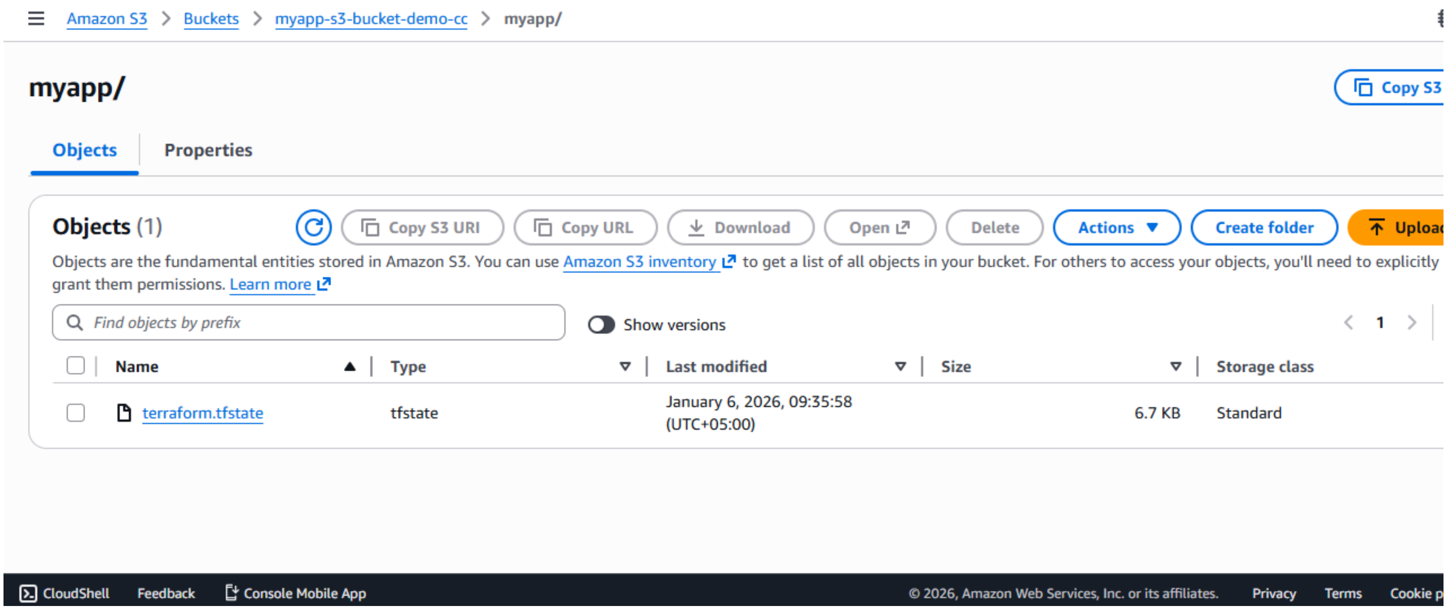
```
commands will detect it and remind you to do so if necessary.
@reenaqureshi ▣ ~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"
aws_iam_group.developers: Refreshing state... [id=developers]
aws_iam_user.lb: Refreshing state... [id=loadbalancer]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260106034629974200000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260106034630003500000002]
null_resource.create_login_profile: Refreshing state... [id=4331948478788271547]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIAZCN5ZTDXBR4BC5HA]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260106033930891000000001]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are
needed.

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

Output:



```
Outputs:

access_key_id = "AKIAZCN5ZTDXBR4BC5HA"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXAMNX4IWNY"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXMDWVCX7X3"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@reenaqureshi ▣ ~/Lab13 $
```

Verify state file in S3:



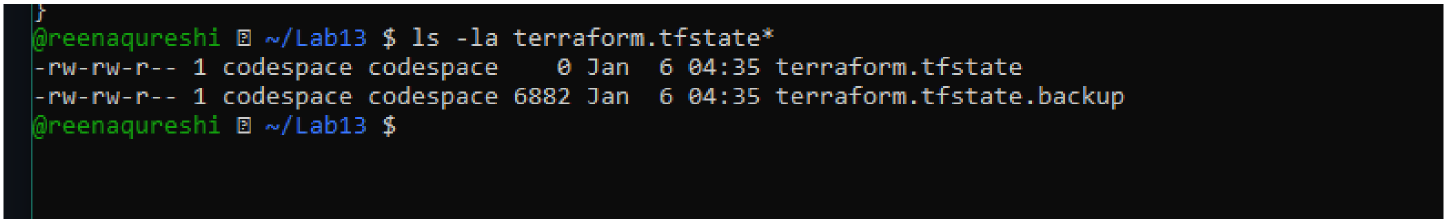
myapp/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

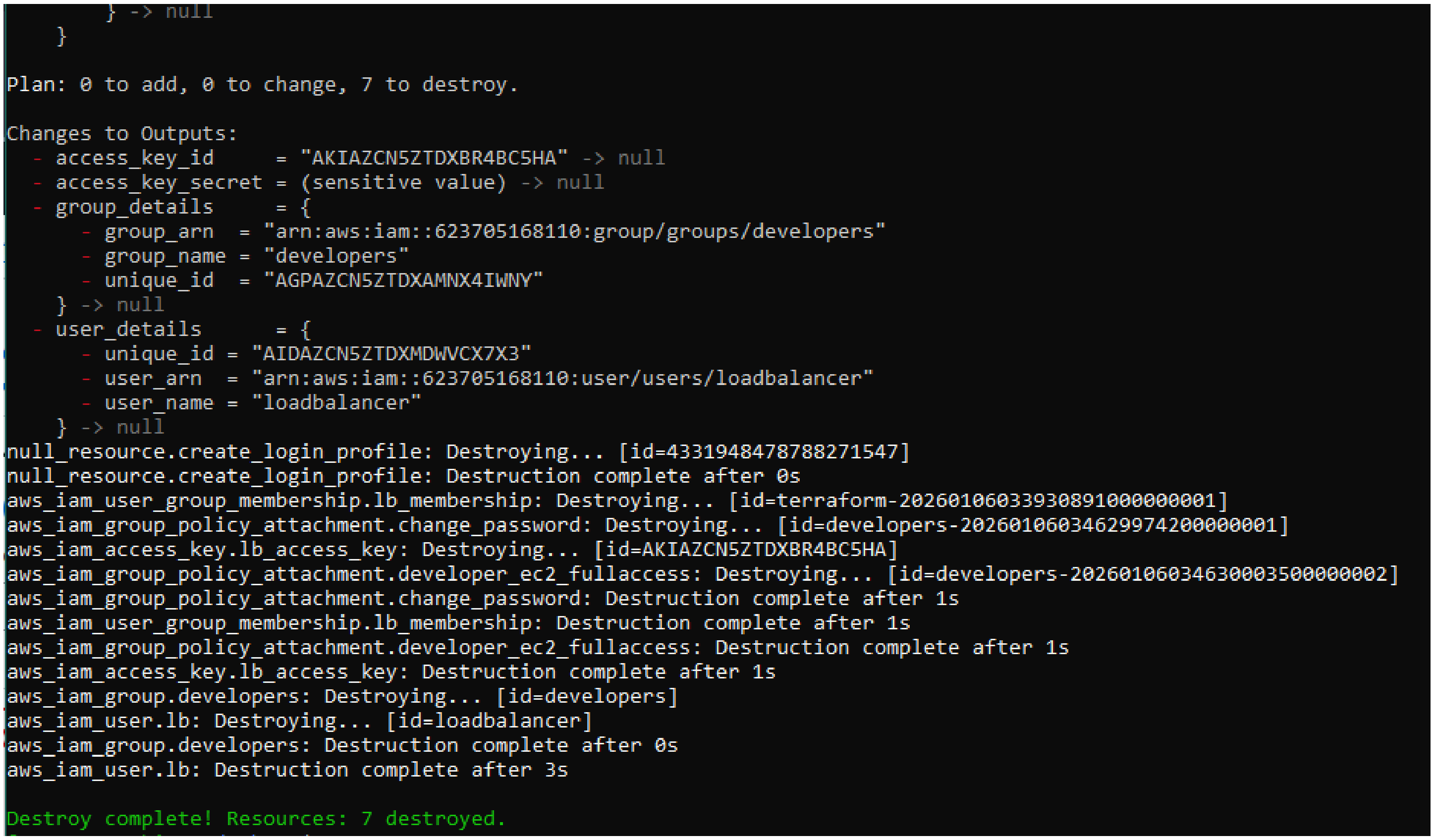
□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class
□	terraform.tfstate	tfstate	January 6, 2026, 09:35:58 (UTC+05:00)	6.7 KB	Standard

Check local state file:



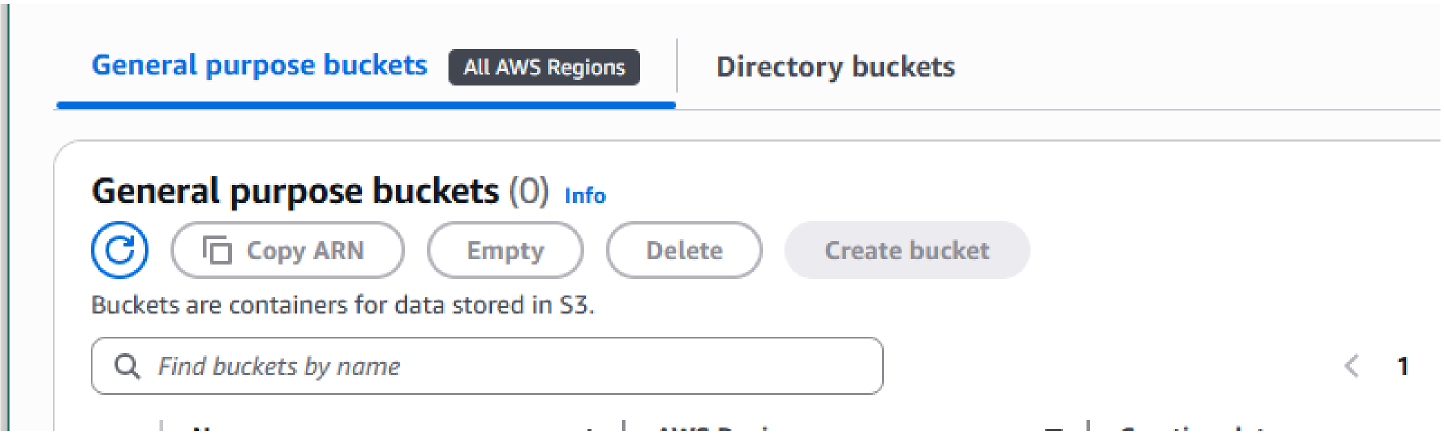
```
}
@reenaqureshi ▯ ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace    0 Jan  6 04:35 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan  6 04:35 terraform.tfstate.backup
@reenaqureshi ▯ ~/Lab13 $
```

Destroy resources and verify state change:



```
        } -> null
    }

Plan: 0 to add, 0 to change, 7 to destroy.

Changes to Outputs:
  - access_key_id     = "AKIAZCN5ZTDXBR4BC5HA" -> null
  - access_key_secret = (sensitive value) -> null
  - group_details     = {
      - group_arn  = "arn:aws:iam::623705168110:group/groups/developers"
      - group_name = "developers"
      - unique_id  = "AGPAZCN5ZTDXAMNX4IWNY"
    } -> null
  - user_details      = {
      - unique_id = "AIDAZCN5ZTDXMDWVCX7X3"
      - user_arn  = "arn:aws:iam::623705168110:user/users/loadbalancer"
      - user_name = "loadbalancer"
    } -> null
null_resource.create_login_profile: Destroying... [id=4331948478788271547]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-20260106033930891000000001]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260106034629974200000001]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIAZCN5ZTDXBR4BC5HA]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260106034630003500000002]
aws_iam_group_policy_attachment.change_password: Destruction complete after 1s
aws_iam_user_group_membership.lb_membership: Destruction complete after 1s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 1s
aws_iam_access_key.lb_access_key: Destruction complete after 1s
aws_iam_group.developers: Destroying... [id=developers]
aws_iam_user.lb: Destroying... [id=loadbalancer]
aws_iam_group.developers: Destruction complete after 0s
aws_iam_user.lb: Destruction complete after 3s

Destroy complete! Resources: 7 destroyed.
```

Verify updated state in S3:

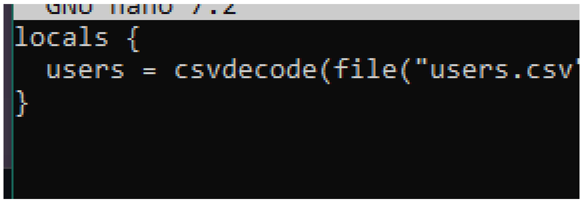


General purpose buckets (0) Info

Buckets are containers for data stored in S3.

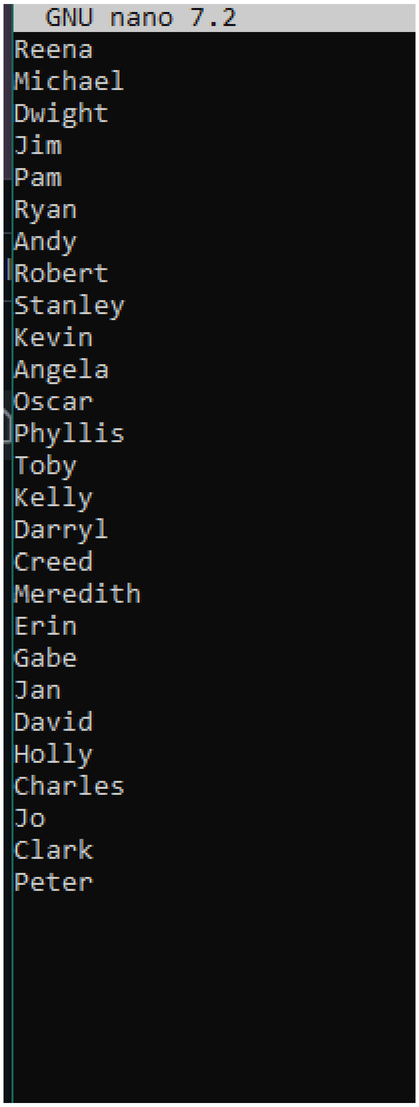
Task 7 — Create Multiple Users from CSV File

Create `locals.tf` file:

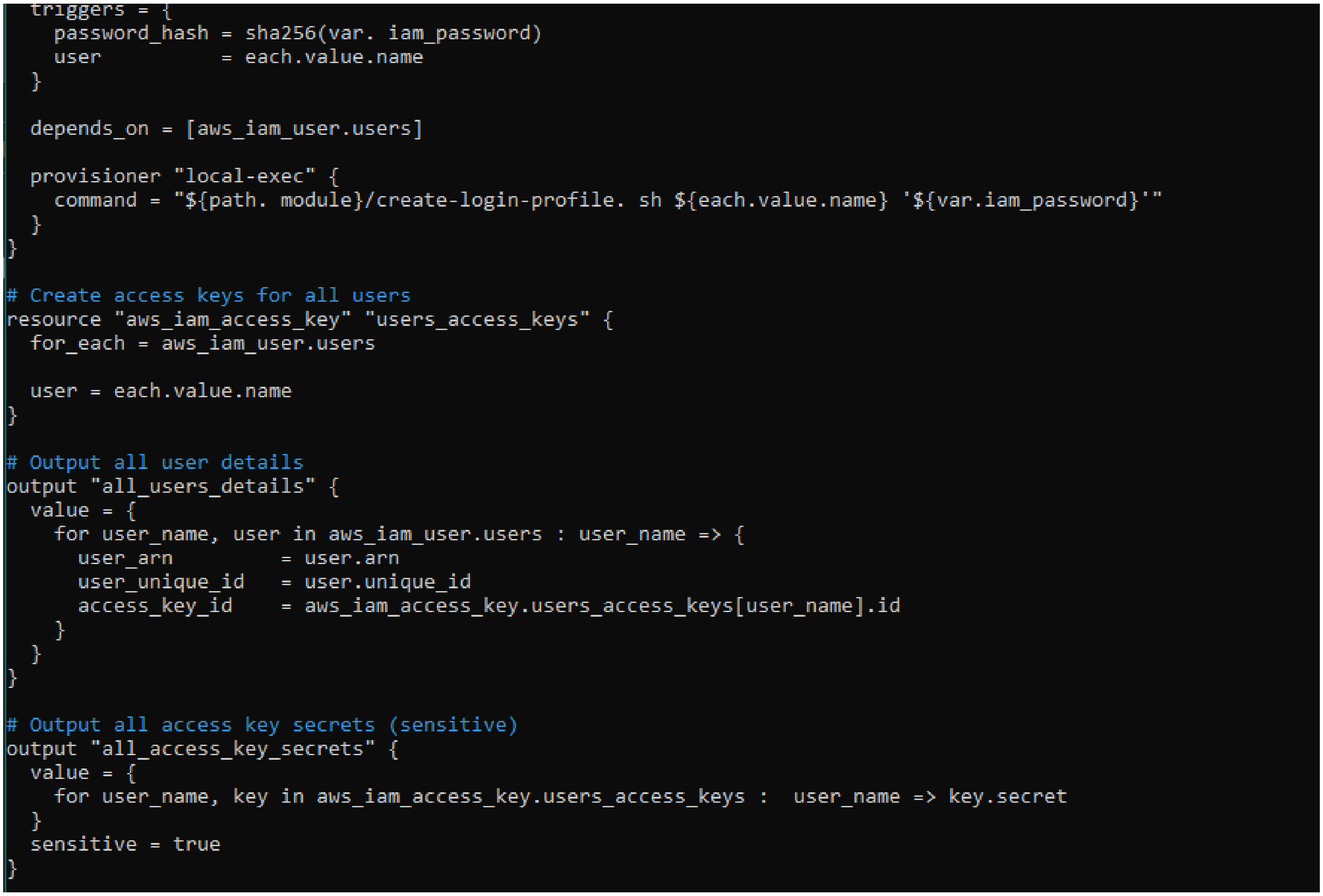


```
  GNU nano 7.2
locals {
  users = csvdecode(file("users.csv"
}
```

Create `users.csv` file:

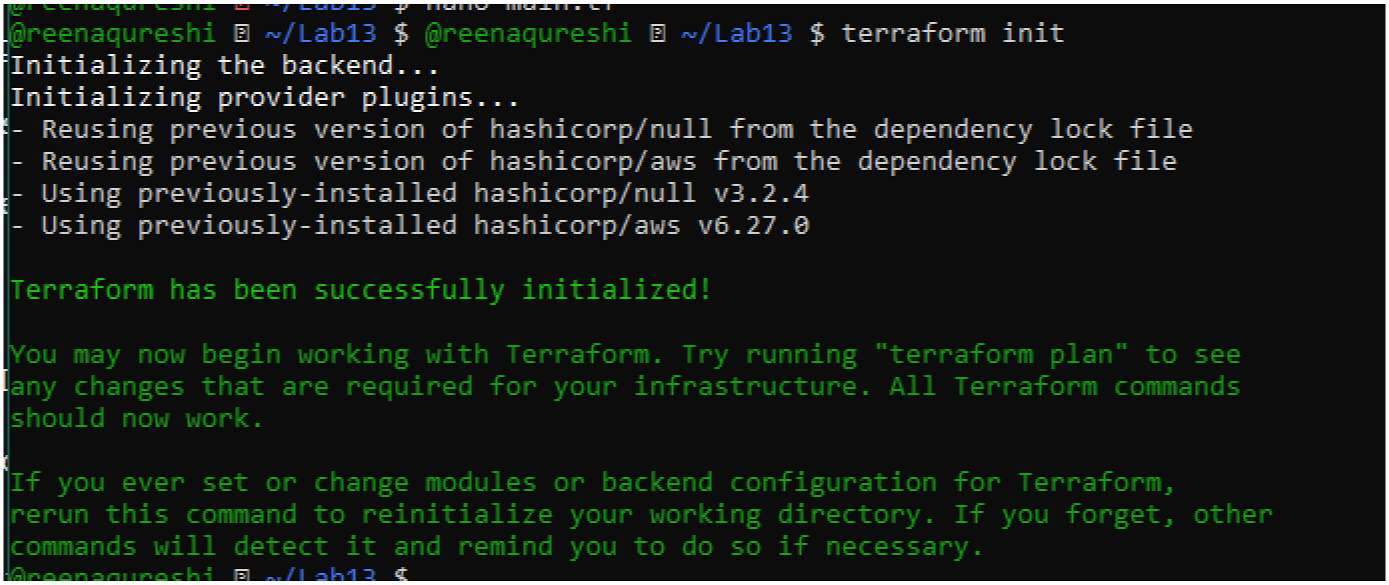


```
  GNU nano 7.2
Reena
Michael
Dwight
Jim
Pam
Ryan
Andy
Robert
Stanley
Kevin
Angela
Oscar
Phyllis
Toby
Kelly
Darryl
Creed
Meredith
Erin
Gabe
Jan
David
Holly
Charles
Jo
Clark
Peter
```



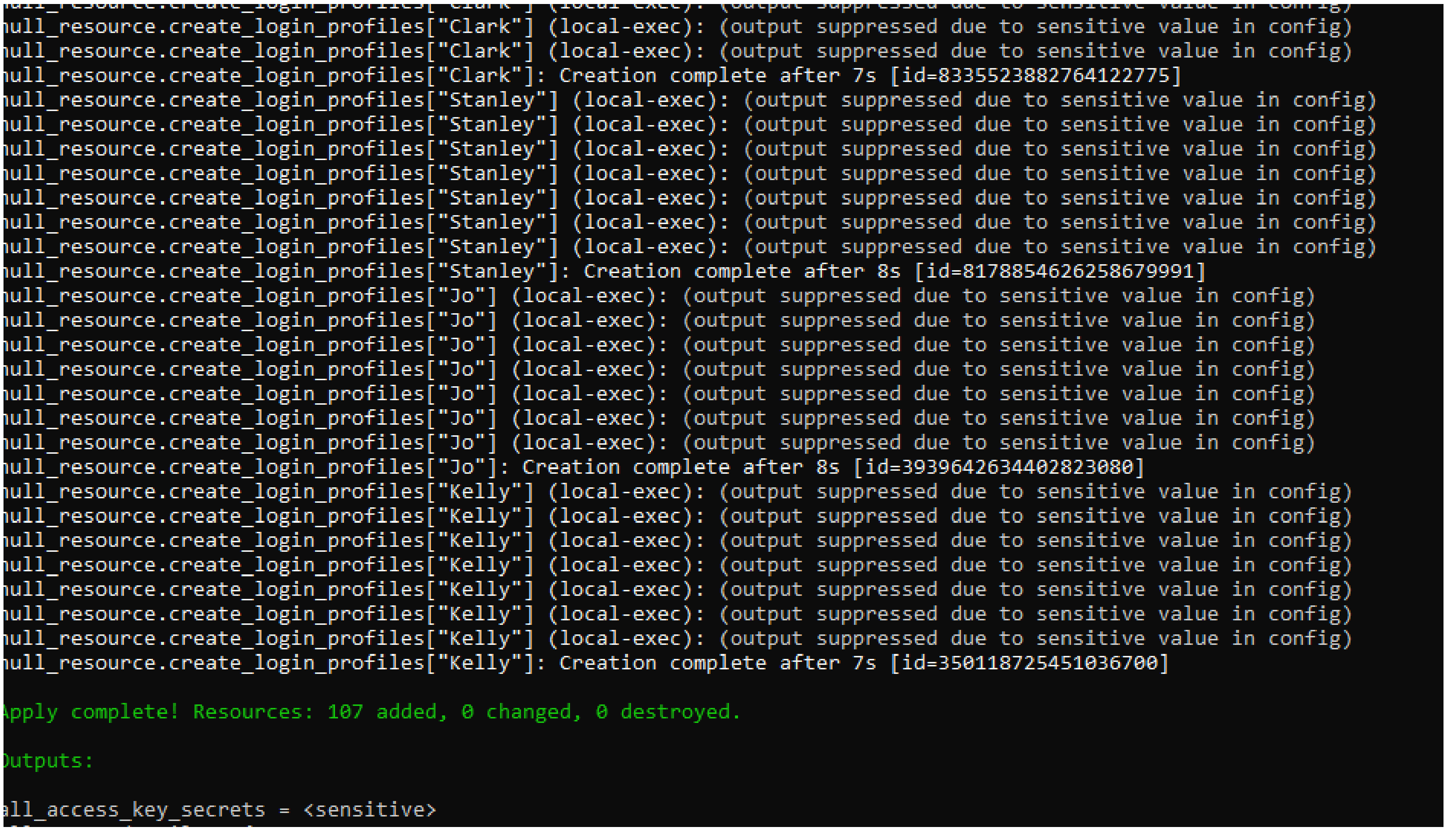
```
  triggers = {
    password_hash = sha256(var. iam_password)
    user          = each.value.name
  }

  depends_on = [aws_iam_user.users]

  provisioner "local-exec" {
    command = "${path. module}/create-login-profile. sh ${each.value.name} '${var.iam_password}'"
  }
}

# Create access keys for all users
resource "aws_iam_access_key" "users_access_keys" {
  for_each = aws_iam_user.users

  user = each.value.name
}

# Output all user details
output "all_users_details" {
  value = {
    for user_name, user in aws_iam_user.users : user_name => {
      user_arn         = user.arn
      user_unique_id   = user.unique_id
      access_key_id    = aws_iam_access_key.users_access_keys[user_name].id
    }
  }
}

# Output all access key secrets (sensitive)
output "all_access_key_secrets" {
  value = {
    for user_name, key in aws_iam_access_key.users_access_keys :  user_name => key.secret
  }
  sensitive = true
}
```

Reinitialize Terraform (since we changed the configuration significantly):



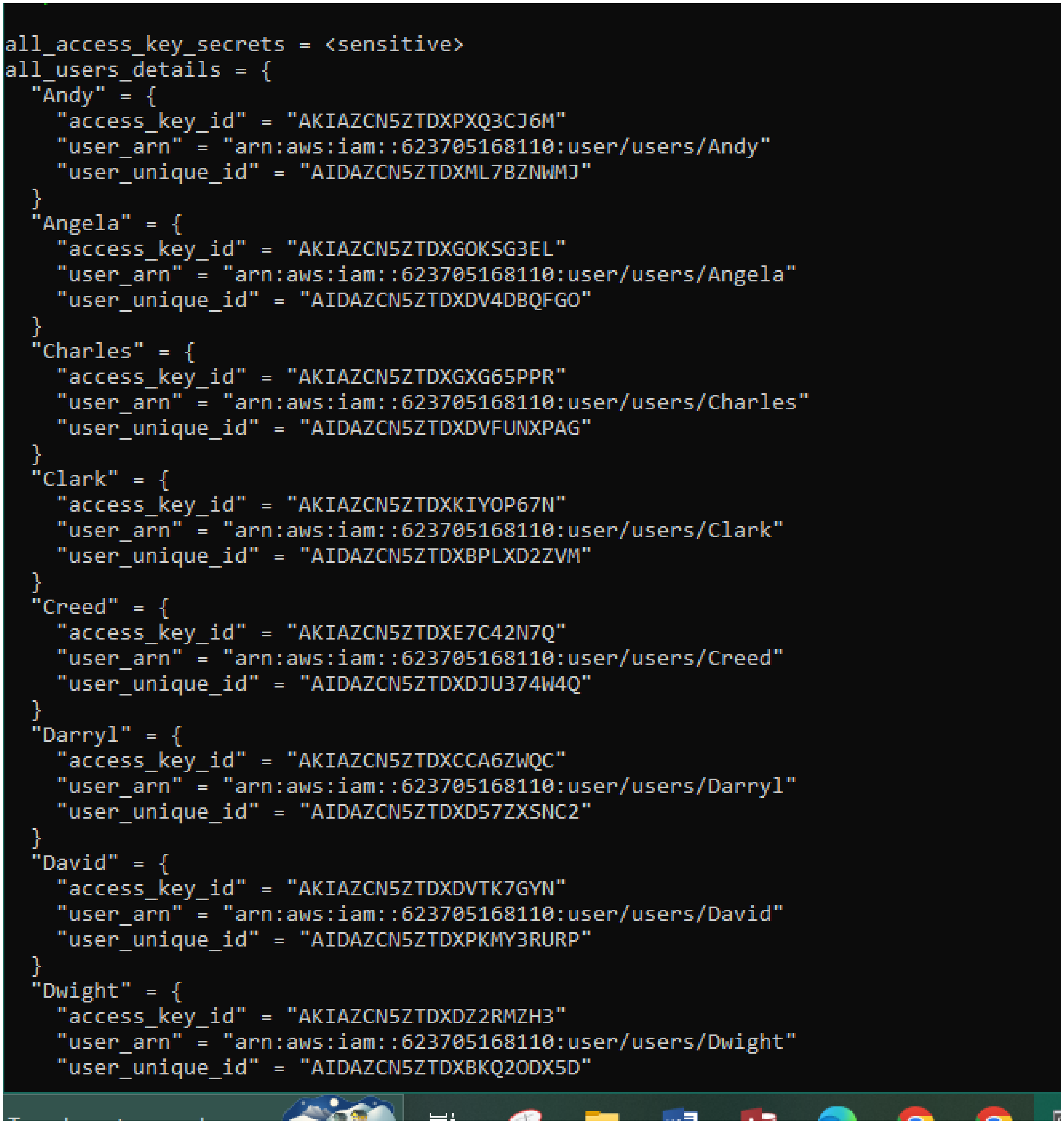
```
@reenaqureshi ▯ ~/Lab13 $ nano main.tf
@reenaqureshi ▯ ~/Lab13 $ @reenaqureshi ▯ ~/Lab13 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/null v3.2.4
- Using previously-installed hashicorp/aws v6.27.0

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.
@reenaqureshi ▯ ~/Lab13 $
```

Apply the configuration to create all users:

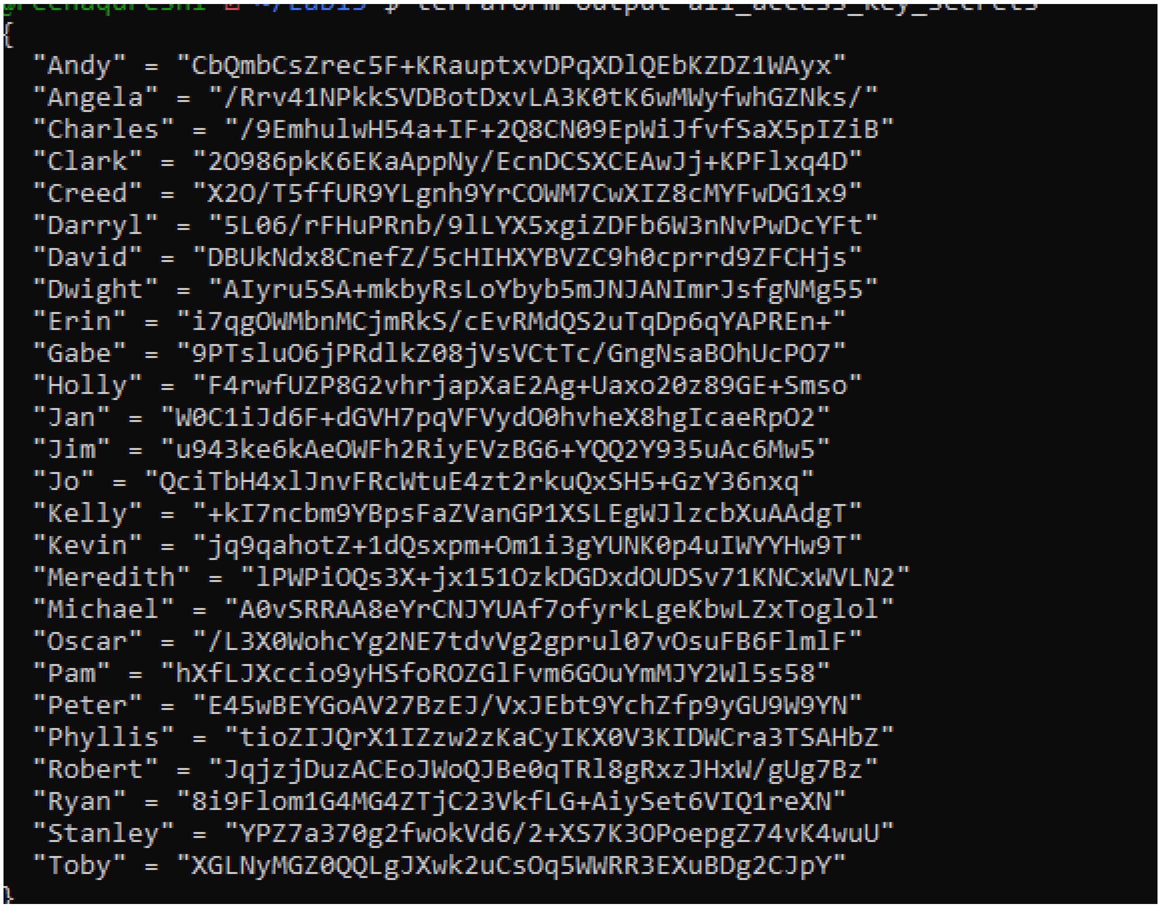


```
null_resource.create_login_profiles["Clark"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Clark"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Clark"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Clark"]: Creation complete after 7s [id=8335523882764122775]
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Stanley"]: Creation complete after 8s [id=8178854626258679991]
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Jo"]: Creation complete after 8s [id=3939642634402823080]
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"] (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profiles["Kelly"]: Creation complete after 7s [id=350118725451036700]

Apply complete! Resources: 107 added, 0 changed, 0 destroyed.

Outputs:

all_access_key_secrets = <sensitive>
```

Output:

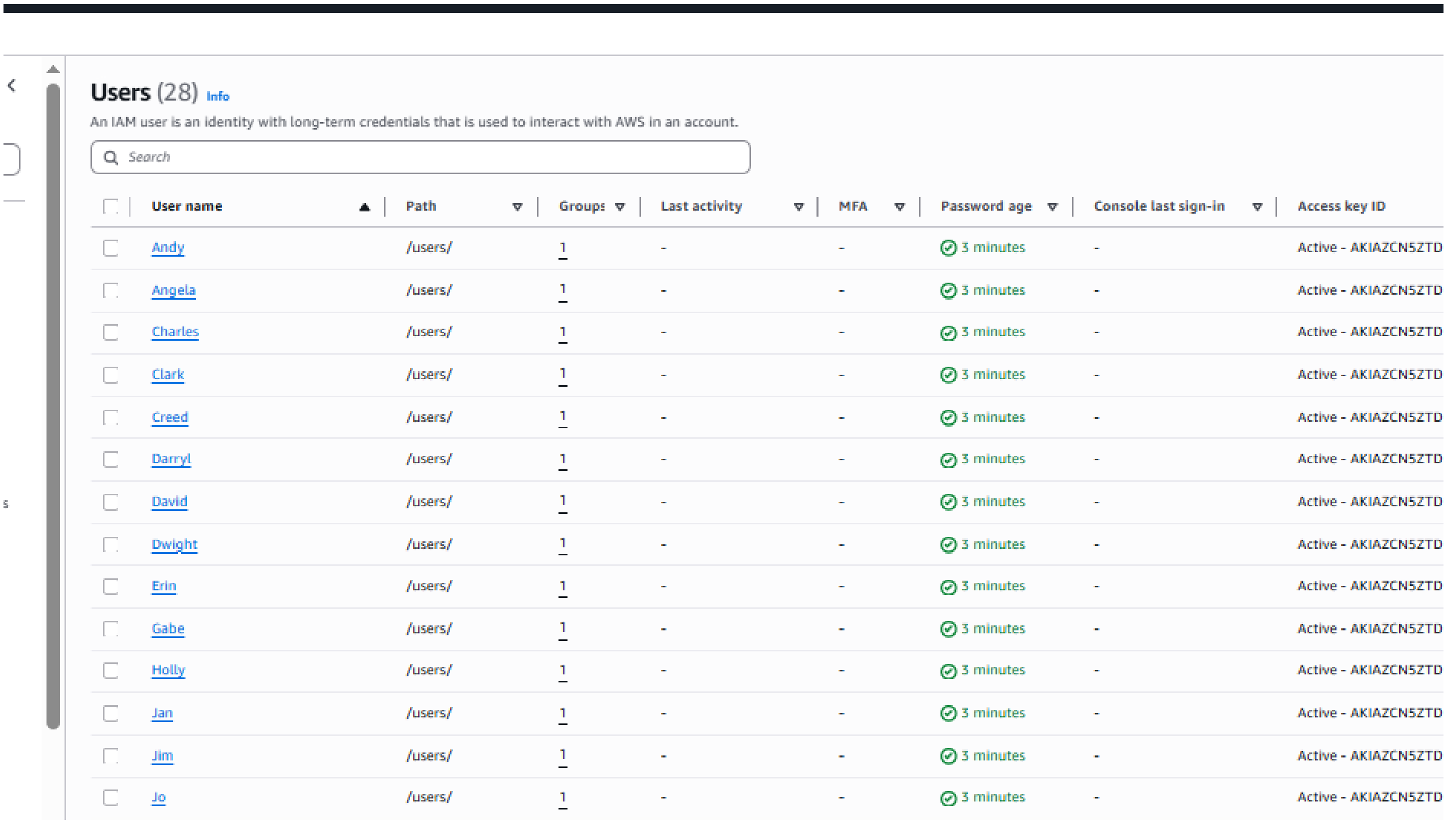


```
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAZCN5ZTDXPXQ3CJ6M"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Andy"
    "user_unique_id" = "AIDAZCN5ZTDXML7BZNWMJ"
  }
  "Angela" = {
    "access_key_id" = "AKIAZCN5ZTDXGOKSG3EL"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Angela"
    "user_unique_id" = "AIDAZCN5ZTDXDV4DBQFGO"
  }
  "Charles" = {
    "access_key_id" = "AKIAZCN5ZTDXGXG65PPR"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Charles"
    "user_unique_id" = "AIDAZCN5ZTDXDVFUNXPAG"
  }
  "Clark" = {
    "access_key_id" = "AKIAZCN5ZTDXKIYOP67N"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Clark"
    "user_unique_id" = "AIDAZCN5ZTDXBPLXD2ZVM"
  }
  "Creed" = {
    "access_key_id" = "AKIAZCN5ZTDXE7C42N7Q"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Creed"
    "user_unique_id" = "AIDAZCN5ZTDXDJU374W4Q"
  }
  "Darryl" = {
    "access_key_id" = "AKIAZCN5ZTDXCCA6ZWQC"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Darryl"
    "user_unique_id" = "AIDAZCN5ZTDXD57ZXSNC2"
  }
  "David" = {
    "access_key_id" = "AKIAZCN5ZTDXDVTK7GYN"
    "user_arn" = "arn:aws:iam::623705168110:user/users/David"
    "user_unique_id" = "AIDAZCN5ZTDXPKMY3RURP"
  }
  "Dwight" = {
    "access_key_id" = "AKIAZCN5ZTDXDZ2RMZH3"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Dwight"
    "user_unique_id" = "AIDAZCN5ZTDXBKQ2ODX5D"
```



```
@reenaqureshi ~/Lab15 $ terraform output all_access_key_secrets
{
  "Andy" = "CbQmbCsZrec5F+KRauptxvDPqXDlQEbKZDZ1WAyx"
  "Angela" = "/Rrv41NPkkSVDBotDxvLA3K0tK6wMWyfwhGZNks/"
  "Charles" = "/9EmhulwH54a+IF+2Q8CN09EpWiJfvfSaX5pIZiB"
  "Clark" = "2O986pkK6EKaAppNy/EcnDCSXCEAwJj+KPFlxq4D"
  "Creed" = "X2O/T5ffUR9YLgnh9YrCOWM7CwXIZ8cMYFwDG1x9"
  "Darryl" = "5L06/rFHuPRnb/9lLYX5xgiZDFb6W3nNvPwDcYFt"
  "David" = "DBUkNdx8CnefZ/5cHIHXYBVZC9h0cprrd9ZFCHjs"
  "Dwight" = "AIyru5SA+mkbyRsLoYbyb5mJNJANImrJsfgNMg55"
  "Erin" = "i7qgOWMbnMCjmRkS/cEvRMdQS2uTqDp6qYAPREn+"
  "Gabe" = "9PTsluO6jPRdlkZ08jVsVCtTc/GngNsaBOhUcPO7"
  "Holly" = "F4rwfUZP8G2vhrjapXaE2Ag+Uaxo20z89GE+Smso"
  "Jan" = "W0C1iJd6F+dGVH7pqVFVydO0hvheX8hgIcaeRpO2"
  "Jim" = "u943ke6kAeOWFh2RiyEVzBG6+YQQ2Y935uAc6Mw5"
  "Jo" = "QciTbH4xlJnvFRcWtuE4zt2rkuQxSH5+GzY36nxq"
  "Kelly" = "+kI7ncbm9YBpsFaZVanGP1XSLEgWJlzcbXuAAdgT"
  "Kevin" = "jq9qahotZ+1dQsxpm+Om1i3gYUNK0p4uIWYYHw9T"
  "Meredith" = "lPWPiOQs3X+jx151OzkDGDxdOUDSv71KNCxWVLN2"
  "Michael" = "A0vSRRAA8eYrCNJYUAf7ofyrkLgeKbwLZxToglol"
  "Oscar" = "/L3X0WohcYg2NE7tdvVg2gprul07vOsuFB6FlmlF"
  "Pam" = "hXfLJXccio9yHSfoROZGlFvm6GOuYmMJY2Wl5s58"
  "Peter" = "E45wBEYGoAV27BzEJ/VxJEbt9YchZfp9yGU9W9YN"
  "Phyllis" = "tioZIJQrX1IZzw2zKaCyIKX0V3KIDWCra3TSAHbZ"
  "Robert" = "JqjzjDuzACEoJWoQJBe0qTRl8gRxzJHxW/gUg7Bz"
  "Ryan" = "8i9Flom1G4MG4ZTjC23VkfLG+AiySet6VIQ1reXN"
  "Stanley" = "YPZ7a370g2fwokVd6/2+XS7K3OPoepgZ74vK4wuU"
  "Toby" = "XGLNyMGZ0QQLgJXwk2uCsOq5WWRR3EXuBDg2CJpY"
}
```

Verify all users in AWS Console:

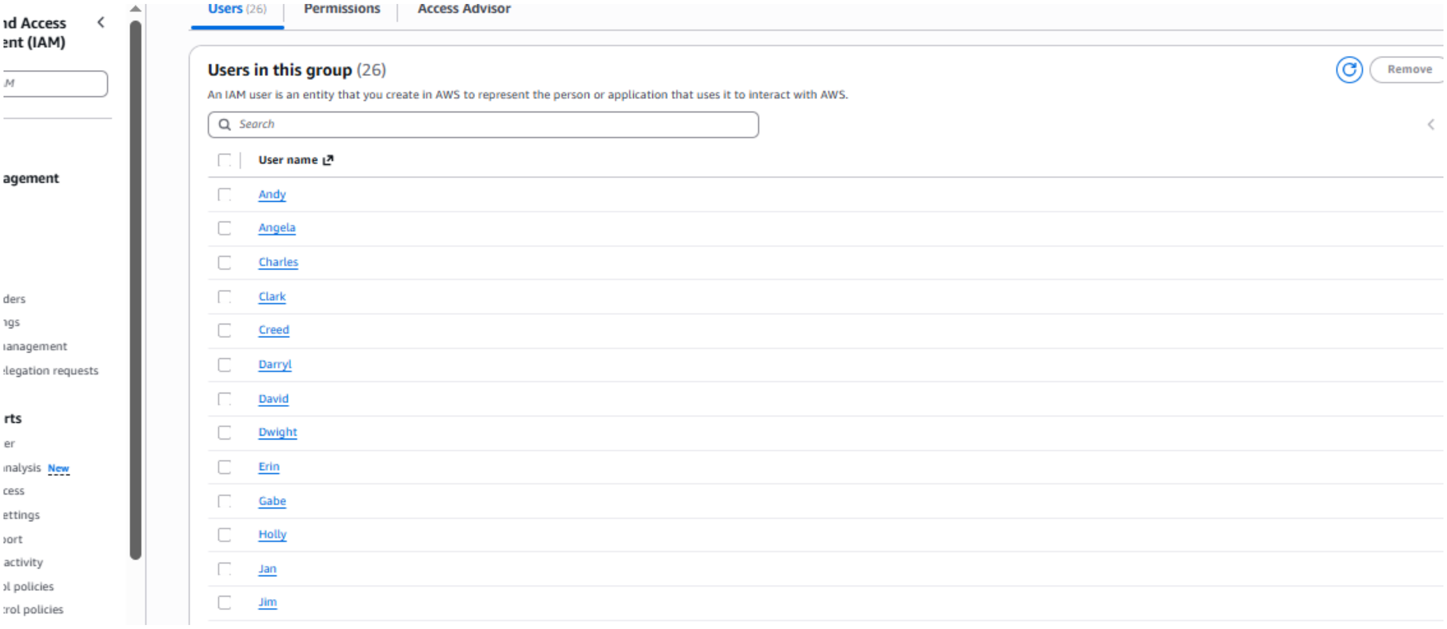


Users (28) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
□	Andy	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Angela	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Charles	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Clark	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Creed	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Darryl	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	David	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Dwight	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Erin	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Gabe	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Holly	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Jan	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Jim	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD
□	Jo	/users/	1	-	-	✓ 3 minutes	-	Active - AKIAZCN5ZTD

Verify group membership:

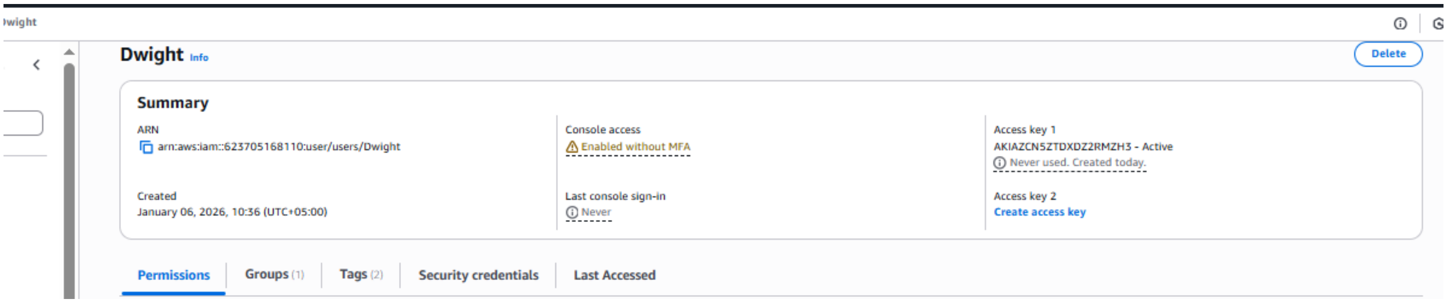


Users in this group (26)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name
Andy
Angela
Charles
Clark
Creed
Darryl
David
Dwight
Erin
Gabe
Holly
Jan
Jim

Verify one user's access keys:



Dwight

Summary

ARN
arn:aws:iam::623705168110:user/users/Dwight

Created
January 06, 2026, 10:36 (UTC+05:00)

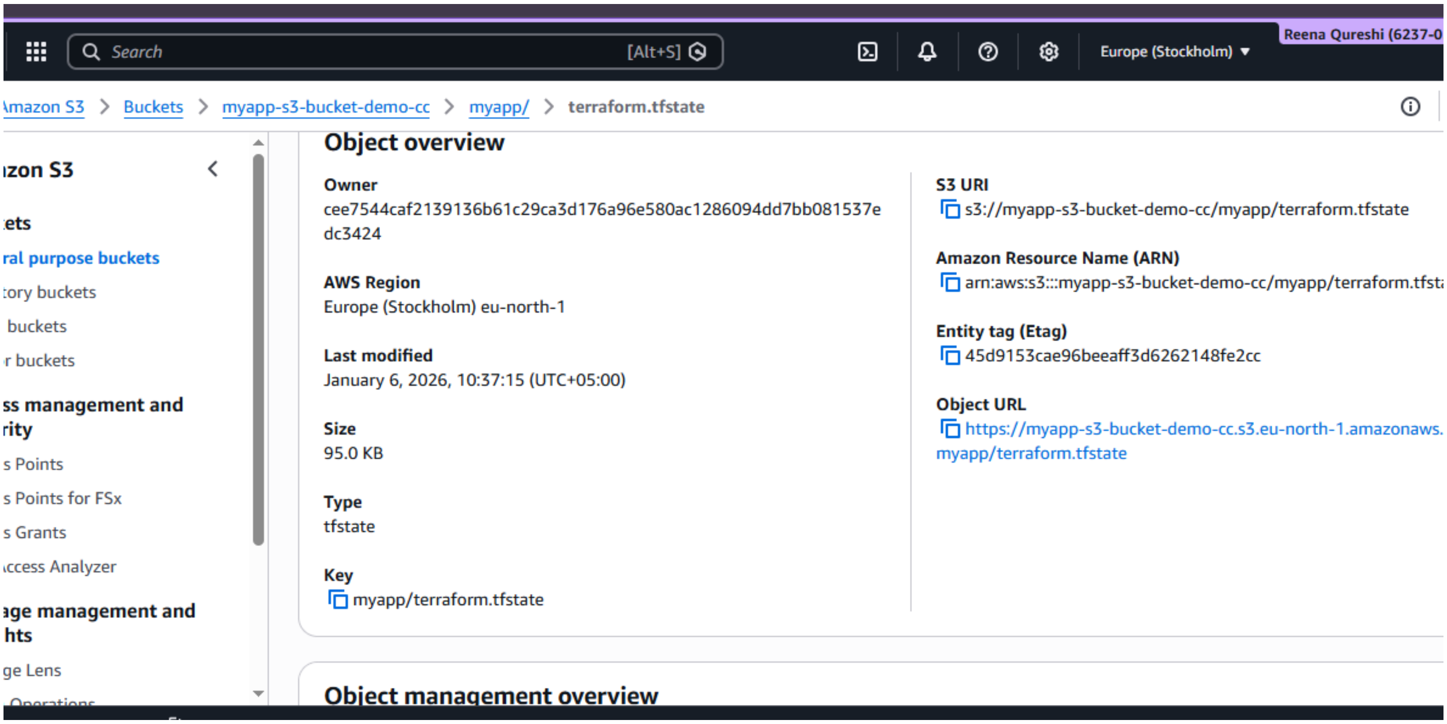
Console access
Enabled without MFA

Last console sign-in
Never

Access key 1
AKIAZCN5ZTDXDZ2RMZH3 - Active
Never used. Created today.

Access key 2
Create access key

Check terraform state in S3:



Amazon S3 > Buckets > myapp-s3-bucket-demo-cc > myapp/ > terraform.tfstate

Object overview

Owner
cee7544caf2139136b61c29ca3d176a96e580ac1286094dd7bb081537edc3424

AWS Region
Europe (Stockholm) eu-north-1

Last modified
January 6, 2026, 10:37:15 (UTC+05:00)

Size
95.0 KB

Type
tfstate

Key
myapp/terraform.tfstate

S3 URI
s3://myapp-s3-bucket-demo-cc/myapp/terraform.tfstate

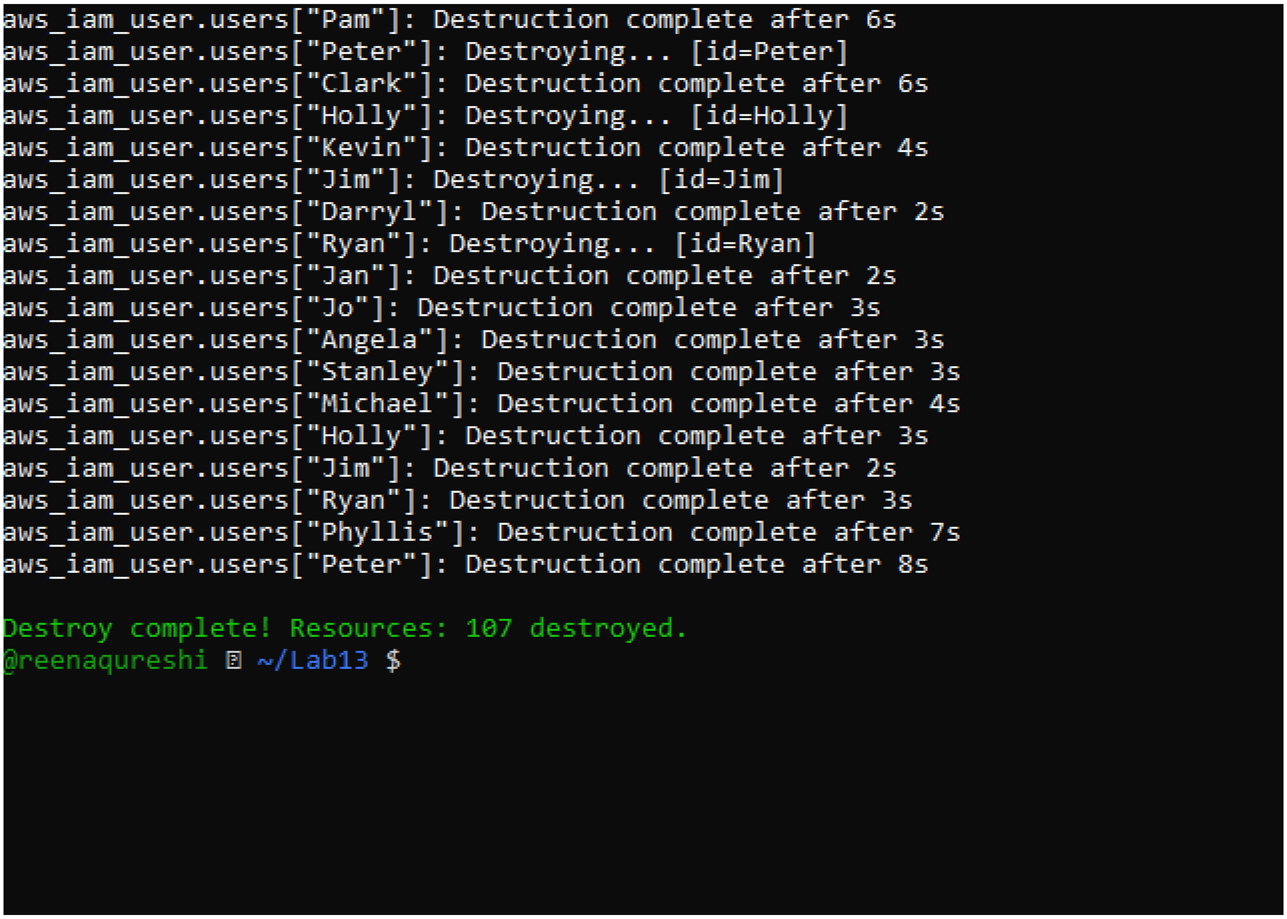
Amazon Resource Name (ARN)
arn:aws:s3:::myapp-s3-bucket-demo-cc/myapp/terraform.tfst

Entity tag (Etag)
45d9153cae96beeaff3d6262148fe2cc

Object URL
https://myapp-s3-bucket-demo-cc.s3.eu-north-1.amazonaws.
myapp/terraform.tfstate

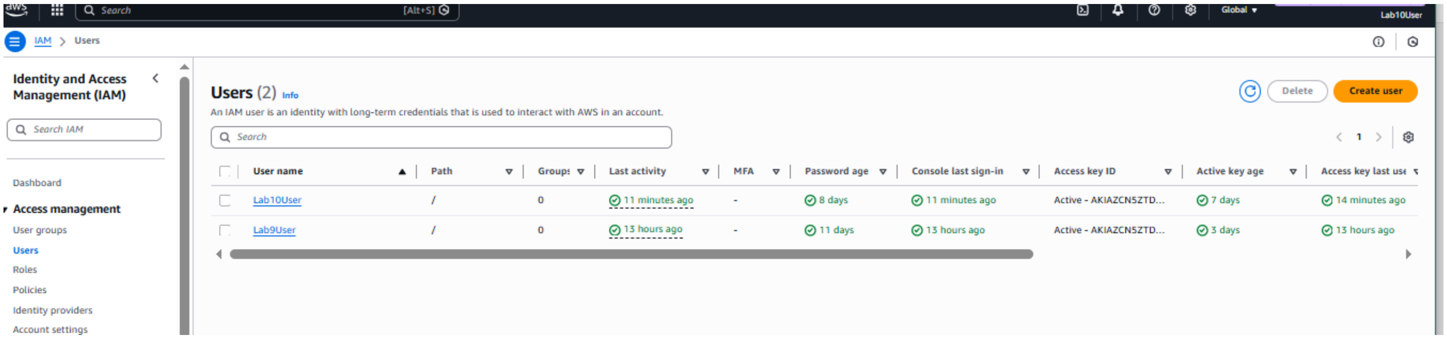
Object management overview

Cleanup



```
aws_iam_user.users["Pam"]: Destruction complete after 6s
aws_iam_user.users["Peter"]: Destroying... [id=Peter]
aws_iam_user.users["Clark"]: Destruction complete after 6s
aws_iam_user.users["Holly"]: Destroying... [id=Holly]
aws_iam_user.users["Kevin"]: Destruction complete after 4s
aws_iam_user.users["Jim"]: Destroying... [id=Jim]
aws_iam_user.users["Darryl"]: Destruction complete after 2s
aws_iam_user.users["Ryan"]: Destroying... [id=Ryan]
aws_iam_user.users["Jan"]: Destruction complete after 2s
aws_iam_user.users["Jo"]: Destruction complete after 3s
aws_iam_user.users["Angela"]: Destruction complete after 3s
aws_iam_user.users["Stanley"]: Destruction complete after 3s
aws_iam_user.users["Michael"]: Destruction complete after 4s
aws_iam_user.users["Holly"]: Destruction complete after 3s
aws_iam_user.users["Jim"]: Destruction complete after 2s
aws_iam_user.users["Ryan"]: Destruction complete after 3s
aws_iam_user.users["Phyllis"]: Destruction complete after 7s
aws_iam_user.users["Peter"]: Destruction complete after 8s

Destroy complete! Resources: 107 destroyed.
@reenaqureshi ➜ ~/Lab13 $
```

Verify users deleted in AWS Console

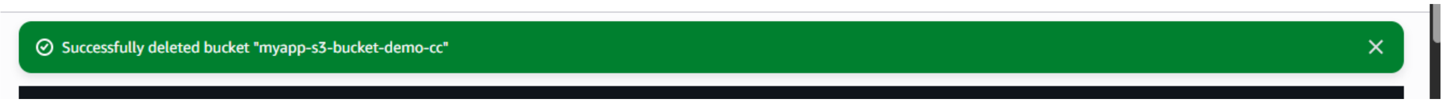


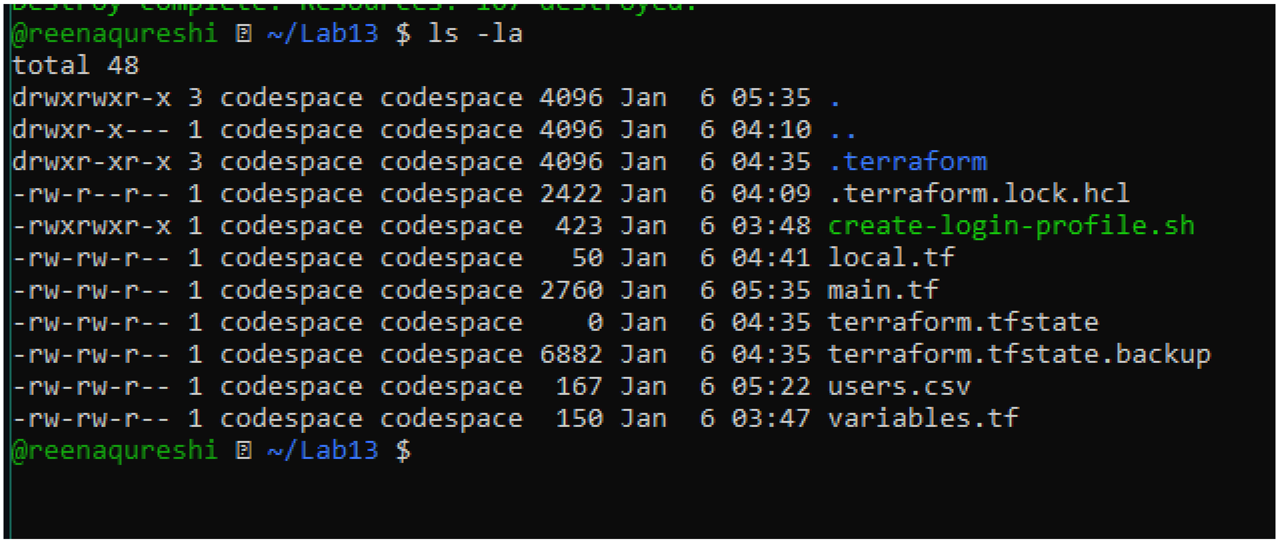
Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key last use
Lab10User	/	0	11 minutes ago	-	8 days	11 minutes ago	Active - AKIAZCN5ZTD...	7 days	14 minutes ago
Lab9User	/	0	13 hours ago	-	11 days	13 hours ago	Active - AKIAZCN5ZTD...	3 days	13 hours ago

Bucket deleted


Successfully deleted bucket "myapp-s3-bucket-demo-cc"


```
Destroy complete! Resources: 10 destroyed.
@reenaqureshi ➜ ~/Lab13 $ ls -la
total 48
drwxrwxr-x 3 codespace codespace 4096 Jan  6 05:35 .
drwxr-x--- 1 codespace codespace 4096 Jan  6 04:10 ..
drwxr-xr-x 3 codespace codespace 4096 Jan  6 04:35 .terraform
-rw-r--r-- 1 codespace codespace 2422 Jan  6 04:09 .terraform.lock.hcl
-rwxrwxr-x 1 codespace codespace  423 Jan  6 03:48 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace   50 Jan  6 04:41 local.tf
-rw-rw-r-- 1 codespace codespace 2760 Jan  6 05:35 main.tf
-rw-rw-r-- 1 codespace codespace    0 Jan  6 04:35 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan  6 04:35 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace  167 Jan  6 05:22 users.csv
-rw-rw-r-- 1 codespace codespace  150 Jan  6 03:47 variables.tf
@reenaqureshi ➜ ~/Lab13 $
```