

Name: Sadaf Riaz

Roll no: 2023-BSE-077

Section: BSE5-B

Lab 13

Task 0 - Lab Setup (Codespace & GH CLI)

0.1 Create GitHub Codespace and Repository

- Screenshot: task0_codespace_create_and_list.png

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Lenovo> gh codespace create --repo SadafRiaz-077/cc_sadafriaz_077_lab13
✓ Codespaces usage for this repository is paid for by SadafRiaz-077
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
sturdy-doodle-97g9gwrqr57r3655
PS C:\Users\Lenovo> gh codespace list
```

NAME	DISPLAY NAME	REPOSITORY	BRANCH	STATE	CREATED AT
obscure-fishstick-4jp7p6...	obscure fishstick	SadafRiaz-077/cc_sad...	main*	Shutdown	about 29 days ago
urban-space-giggle-97g9g...	urban space giggle	SadafRiaz-077/cc_sad...	main	Shutdown	about 28 days ago
musical-waffle-r45g5wqr...	musical waffle	SadafRiaz-077/cc_sad...	main*	Shutdown	about 28 days ago
shiny-couscous-wr7qxp9p...	shiny couscous	SadafRiaz-077/lab-10	main*	Shutdown	about 26 days ago
ominous-umbrella-g4vqv6j...	ominous umbrella	SadafRiaz-077/cc_sad...	main	Shutdown	about 12 days ago
curly-winner-jjp5p9v4vpx...	curly winner	SadafRiaz-077/cc_sad...	main*	Shutdown	about 12 days ago
bug-free-space-capybara-...	bug-free space capybara	SadafRiaz-077/cc_sad...	main*	Shutdown	about 12 days ago
studious-orbit-wr7qxp9p...	studious orbit	SadafRiaz-077/cc_sad...	main	Shutdown	about 12 days ago
redesigned-meme-pj5r594g...	redesigned meme	SadafRiaz-077/cc_sad...	main*	Shutdown	about 11 days ago
stunning-engine-4jp7p65g...	stunning engine	SadafRiaz-077/cc_sad...	main	Shutdown	about 10 days ago
cautious-space-sniffle-7...	cautious space sniffle	SadafRiaz-077/cc_sad...	main	Shutdown	about 5 days ago
super-duper-funicular-69...	super-duper funicular	SadafRiaz-077/cc_sad...	main*	Shutdown	about 5 days ago
crispy-fortnight-69xgww...	crispy fortnight	SadafRiaz-077/cc_sad...	main	Available	about 3 minutes ago
sturdy-doodle-97g9gwrqr5...	sturdy doodle	SadafRiaz-077/cc_sad...	main	Available	less than a minute ago

```
PS C:\Users\Lenovo>
```

0.2 SSH into Codespace

- Screenshot: task0_codespace_ssh_connected.png

```
PS C:\Users\Lenovo> gh codespace ssh -c sturdy-doodle-97g9gwrqr57r3655
Enter passphrase for key 'C:\Users\Lenovo\.ssh/id_ed25519':
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@SadafRiaz-077 → /workspaces/cc_sadafriaz_077_lab13 (main) $
```

Task 1 — Create IAM Group and Output Details

1.1 Create Project Directory

- Screenshot: task1_project_directory.png

```
@SadafRiaz-077 → /workspaces/cc_sadafriaz_077_lab13 (main) $ mkdir -p ~/Lab13
@SadafRiaz-077 → /workspaces/cc_sadafriaz_077_lab13 (main) $ cd ~/Lab13
@SadafRiaz-077 → ~/Lab13 $
```

1.2 Create main.tf File

- Screenshot: task1_file_created.png

```
@SadafRiaz-077 → ~/Lab13 $ touch main.tf
@SadafRiaz-077 → ~/Lab13 $
```

1.3 Add Group Configuration in main.tf

- Screenshot: task1_main_tf.png

```
@SadafRiaz-077 → ~/Lab13 $ nano main.tf
@SadafRiaz-077 → ~/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
  }
}

@SadafRiaz-077 → ~/Lab13 $
```

1.4 Initialize Terraform

- Screenshot: task1_terraform_init.png

```
@SadafRiaz-077 → ~/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.
@SadafRiaz-077 → ~/Lab13 $
```

1.5 Apply Terraform Configuration

- Screenshot: task1_terraform_apply.png

```
aws_iam_group.developers: Creation complete after 1s [id=developers]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::404842057573:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAV4QTXR5S4IYNTQKS7"
}
@SadafRiaz-077 → ~/Lab13 $
```

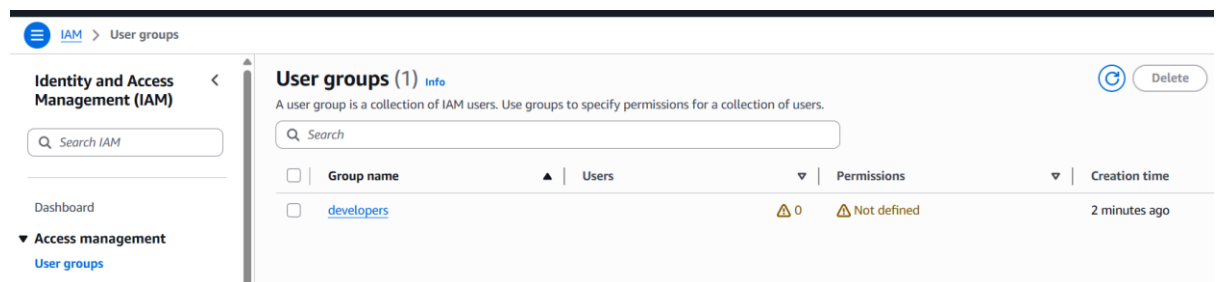
1.6 Display Group Details

- Screenshot: task1_terraform_output.png

```
@SadafRiaz-077 → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::404842057573:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAV4QTXR5S4IYNTQKS7"
}
@SadafRiaz-077 → ~/Lab13 $
```

1.7 Verify Group in AWS Console

- Screenshot: task1_aws_console_group.png



Task 2 — Create IAM User with Group Membership

2.1 Update main.tf to Add IAM User

- Screenshot: task2_main_tf_user.png

```

@SadafRiaz-077 → ~/Lab13 $ cat main.tf
provider "aws" {
  shared_config_files = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

# IAM Group
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
  }
}

# IAM User
resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/users/"
  force_destroy = true

  tags = {
    DisplayName = "Load Balancer"
  }
}

# Add user to developers group
resource "aws_iam_user_group_membership" "lb_membership" {
  user = aws_iam_user.lb.name
}

```

2.2 Apply Configuration

- Screenshot: task2_terraform_apply.png

```

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::404842057573:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAV4QTXR5S4IYNTQKS7"
}
user_details = {
  "unique_id" = "AIDAV4QTXR5SWAZWYSA5M"
  "user_arn" = "arn:aws:iam::404842057573:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@SadafRiaz-077 → ~/Lab13 $

```

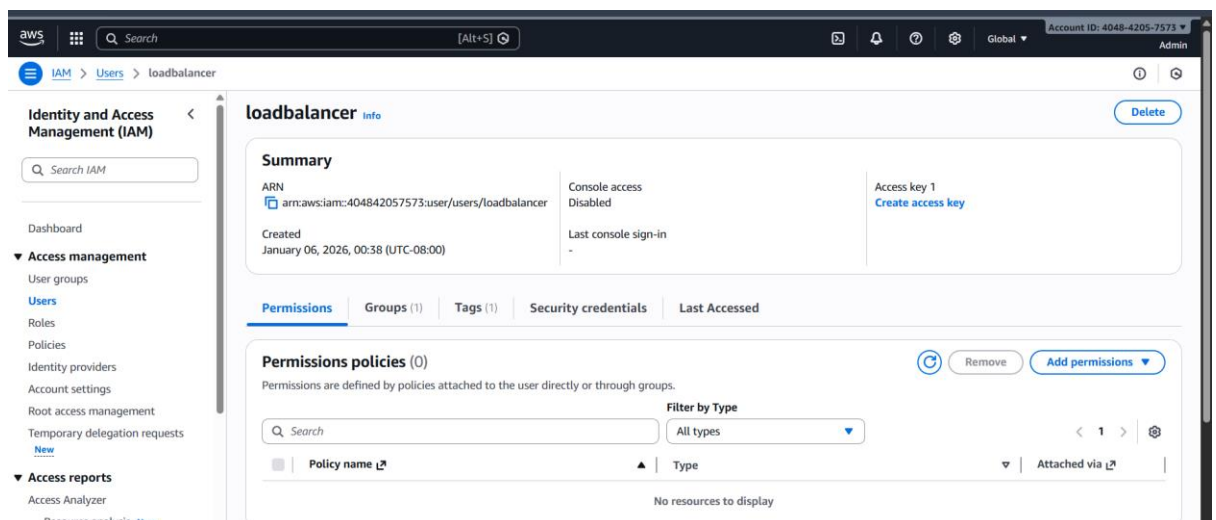
2.3 Display User Details

- Screenshot: task2_terraform_output.png

```
@SadafRiaz-077 → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::404842057573:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAV4QTXR5S4IYNTQKS7"
}
user_details = {
  "unique_id" = "AIDAV4QTXR5SWAZWYSA5M"
  "user_arn" = "arn:aws:iam::404842057573:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@SadafRiaz-077 → ~/Lab13 $
```

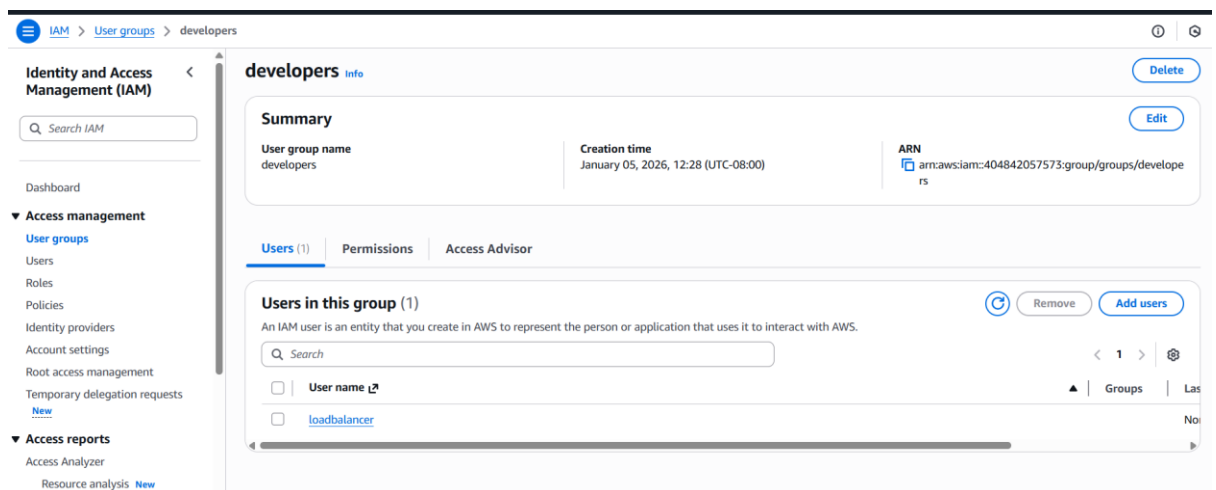
2.4 Verify User in AWS Console

- Screenshot: task2_aws_console_user.png



2.5 Verify User Group Membership

- Screenshot: task2_aws_console_user_groups.png



Task 3 — Attach Policies to IAM Group

3.1 Update main.tf to Attach Policies

- Screenshot: task3_main_tf_policies.png

```
@SadafRiaz-077 →~/Lab13 $ nano main.tf
@SadafRiaz-077 →~/Lab13 $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

# IAM Group
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
  }
}

# IAM User
resource "aws_iam_user" "lb" {
  name       = "loadbalancer"
  path       = "/users/"
  force_destroy = true

  tags = {
    DisplayName = "Load Balancer"
  }
}

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

output "user_details" {
```

3.2 Apply Configuration

- Screenshot: task3_terraform_apply.png

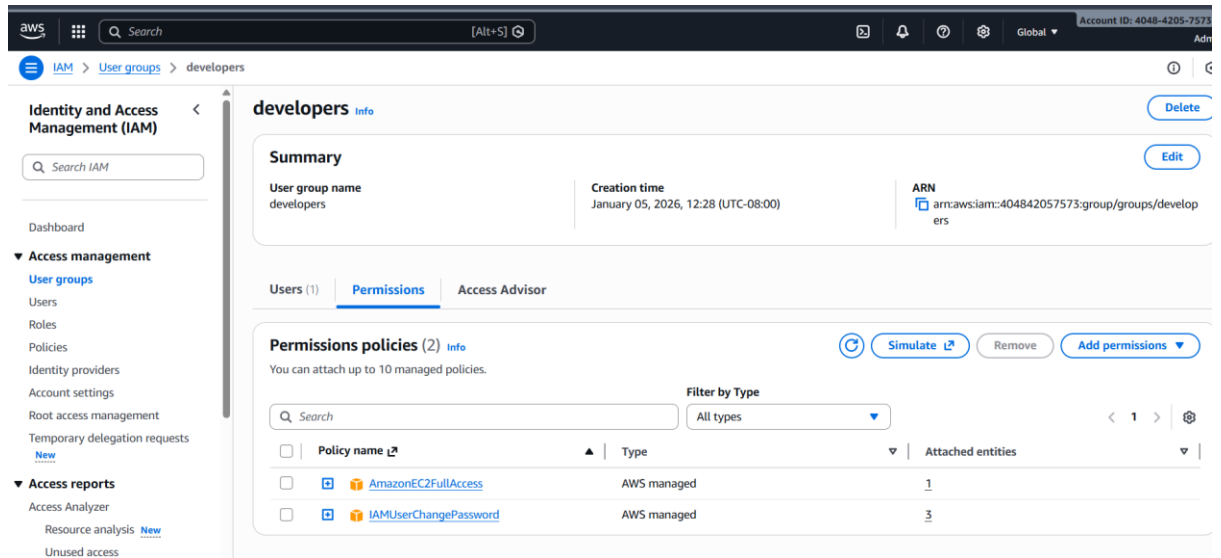
```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::404842057573:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAV4QTXR5S4IYNTQKS7"
}
user_details = {
  "unique_id" = "AIDAV4QTXR5SWAZWYSA5M"
  "user_arn" = "arn:aws:iam::404842057573:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@SadafRiaz-077 →~/Lab13 $
```

3.3 Verify Attached Policies in AWS Console

- Screenshot: task3_aws_console_policies.png



Task 4 — Create Login Profile for IAM User

4.1 Create variables.tf File

- Screenshot: task4_variables_tf.png

```
@SadafRiaz-077 → ~/Lab13 $ touch variables.tf
@SadafRiaz-077 → ~/Lab13 $ nano variables.tf
@SadafRiaz-077 → ~/Lab13 $ cat variables.tf
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "IdontKnow"
}
@SadafRiaz-077 → ~/Lab13 $
```

4.2 Create Login Profile Bash Script

- Screenshot: task4_create_login_script.png

```
@SadafRiaz-077 → ~/Lab13 $ touch create-login-profile.sh
@SadafRiaz-077 → ~/Lab13 $ nano create-login-profile.sh
@SadafRiaz-077 → ~/Lab13 $ cat create-login-profile.sh
#!/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>&1; then
  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
@SadafRiaz-077 → ~/Lab13 $
```

4.3 Make Bash Script Executable

- Screenshot: task4_chmod_script.png

```
@SadafRiaz-077 → ~/Lab13 $ chmod +x create-login-profile.sh
@SadafRiaz-077 → ~/Lab13 $
```

4.4 Update main.tf to Add Null Resource

- Screenshot: task4_main_tf_login_profile.png

```
@SadafRiaz-077 → ~/Lab13 $ nano main.tf
@SadafRiaz-077 → ~/Lab13 $ cat main.tf
provider "aws" {
  shared_config_files = ["~/aws/config"]
  shared_credentials_files = ["~/aws/credentials"]
}

# IAM Group
resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/groups/"
}

# IAM User
resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/users/"
  force_destroy = true
}

# Group Membership
resource "aws_iam_group_membership" "lb_membership" {
  user = aws_iam_user.lb.name
  groups = [
    aws_iam_group.developers.name
  ]
}

# Policies
resource "aws_iam_group_policy_attachment" "ec2" {
  group = aws_iam_group.developers.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

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

# @ Task 4 ADDITION (ONLY THIS IS NEW)
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}'"
  }
}

@SadafRiaz-077 → ~/Lab13 $
```

4.5 Apply Configuration to Create Login Profile

- Screenshot: task4_terraform_apply.png

```
@SadafRiaz-077 → ~/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]
null_resource.create_login_profile: Refreshing state... [id=3336496738294227948]
aws_iam_group_policy_attachment.ec2: Refreshing state... [id=developers-20260106092516894500000002]
aws_iam_group_policy_attachment.password: Refreshing state... [id=developers-20260106092516804800000001]
aws_iam_group_membership.lb_membership: Refreshing state... [id=terraform-20260106083820227900000001]

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

Terraform will perform the following actions:

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

Plan: 1 to add, 0 to change, 0 to destroy.
aws_iam_group_policy_attachment.password: Creating...
aws_iam_group_policy_attachment.password: Creation complete after 1s [id=developers-20260106093308704300000001]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@SadafRiaz-077 → ~/Lab13 $
```

4.6 Verify Login Profile with AWS CLI

- Screenshot: task4_aws_cli_verify.png

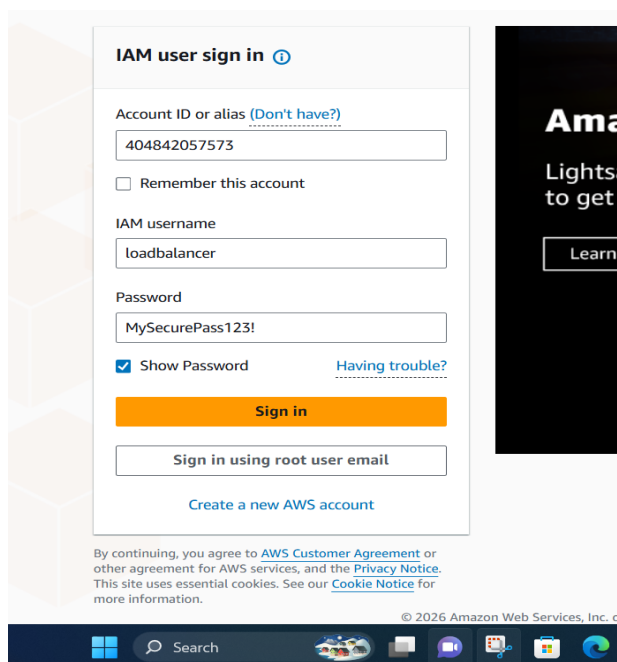

```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@SadafRiaz-077 → ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
  "LoginProfile": {
    "UserName": "loadbalancer",
    "CreateDate": "2026-01-06T09:25:24Z",
    "PasswordResetRequired": true
  }
}
@SadafRiaz-077 → ~/Lab13 $

```

4.7 Verify Login Profile in AWS Console

- Screenshot: task4_aws_console_login.png



IAM user sign in

Account ID or alias [\(Don't have?\)](#)

404842057573

☐ Remember this account

IAM username

loadbalancer

Password

MySecurePass123!

☒ Show Password [Having trouble?](#)

Sign in

Sign in using root user email

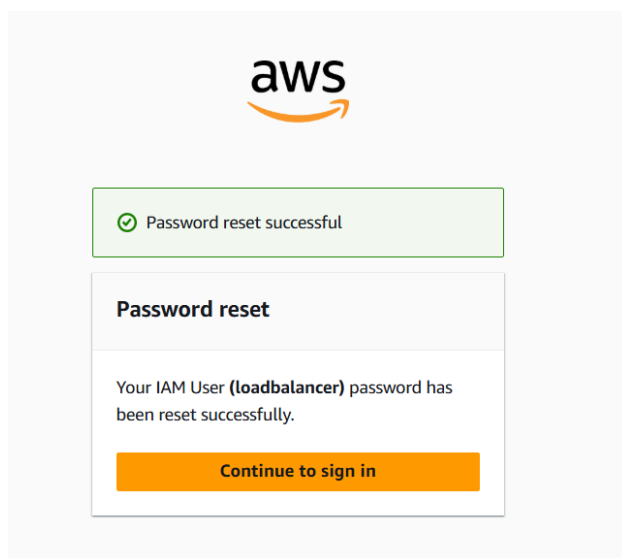
[Create a new AWS account](#)

By continuing, you agree to [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

© 2026 Amazon Web Services, Inc. or its affiliates. All rights reserved.

4.8 Verify Password Reset Prompt in AWS Console

- Screenshot: task4_aws_console_password_reset.png



aws

✓ Password reset successful

Password reset

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

Continue to sign in

Task 5 — Generate Access Keys for IAM User

5.1 Add Access Key Resource to main.tf

- Screenshot: task5_main_tf_access_keys.png

```
gSadaFRiaz-077 → ~/Lab13 $ nano main.tf
gSadaFRiaz-077 → ~/Lab13 $ cat main.tf

provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

# IAM Group
resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/groups/"
}

# IAM User
resource "aws_iam_user" "lb" {
  name         = "loadbalancer"
  path         = "/users/"
  force_destroy = true
}

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

# Policies
resource "aws_iam_group_policy_attachment" "ec2" {
  group      = aws_iam_group.developers.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

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

# @ Task 4 ADDITION (ONLY THIS IS NEW)
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}'"
  }
}

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
}

gSadaFRiaz-077 → ~/Lab13 $
```

5.2 Apply Configuration to Generate Access Keys

- Screenshot: task5_terraform_apply.png

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

Outputs:

access_key_id = "AKIAV4QTXR5S5Z5FM6YFE"
access_key_secret = <sensitive>
gSadaFRiaz-077 → ~/Lab13 $
```

5.3 Display Access Key Information in Terraform Output

- Screenshot: task5_terraform_output.png

```
@SadafRiaz-077 → ~/Lab13 $ terraform output
access_key_id = "AKIAV4QTXR5SZ5FM6YFE"
access_key_secret = <sensitive>
@SadafRiaz-077 → ~/Lab13 $
```

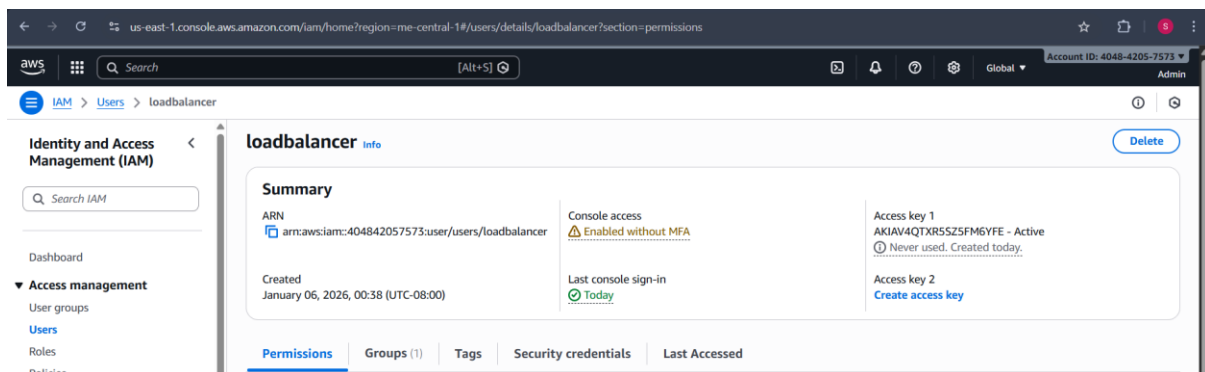
5.4 View Secret in Terraform State

- Screenshot: task5_tfstate_secret.png

```
@SadafRiaz-077 → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
  "access_key_secret": {
    "value": "6+TiH2VnzUliW9s/1iDw+4sMAW81DkeKP8/AH+ec",
    "type": "string",
    "sensitive": true
  }
},
"resources": [
  {
    "mode": "managed",
    "type": "aws_iam_access_key",
    "name": "lb_access_key",
@SadafRiaz-077 → ~/Lab13 $
```

5.5 Verify Access Key in AWS Console

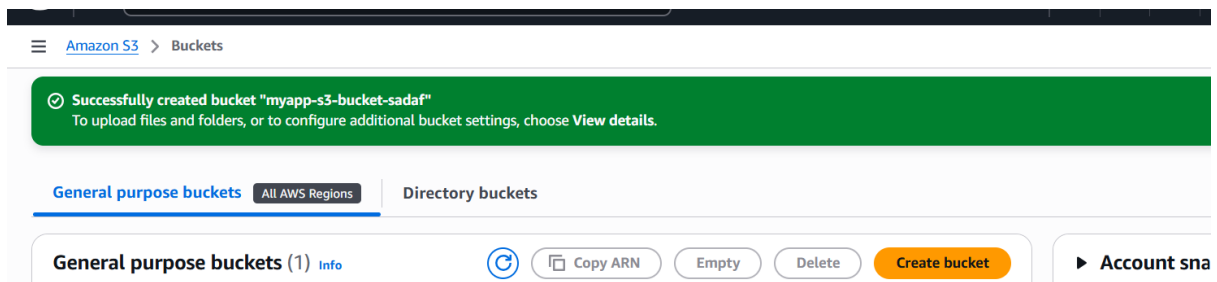
- Screenshot: task5_aws_console_access_keys.png



Task 6 — Implement Terraform Remote State with S3

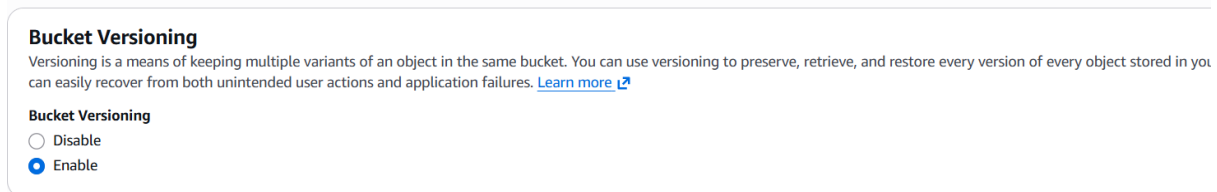
6.1 Create S3 Bucket for Remote State

- Screenshot: task6_s3_bucket_create.png



6.2 Enable Versioning in S3 Bucket

- Screenshot: task6_s3_bucket_versioning.png



6.3 Add S3 Backend to main.tf

- Screenshot: task6_main_tf_backend.png

```
9SadafRiaz-077 →~/Lab13 $ nano main.tf
9SadafRiaz-077 →~/Lab13 $ cat main.tf
terraform {
  backend "s3" {
    bucket = "myapp-s3-bucket-sadaf" # Replace with your bucket name
    key    = "myapp/terraform.tfstate"
    region = "me-central-1" # Update with your region
    encrypt = true
    use_lockfile = true
  }
}

provider "aws" {
  region = "me-central-1" # Update with your AWS region
}

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

resource "aws_iam_group" "developers" {
  name = "developers"
}

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

resource "aws_iam_group_policy_attachment" "ec2" {
  group      = aws_iam_group.developers.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

resource "aws_iam_user_group_membership" "lb_membership" {
  user      = aws_iam_user.lb.name
  group     = aws_iam_group.developers.name
}
```

6.4 Reinitialize Terraform with S3 Backend

- Screenshot: task6_terraform_init_migrate.png

```
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.
@SadafRiaz-077 → ~/Lab13 $ terraform {
```

6.5 Apply Configuration with Remote State

- Screenshot: task6_terraform_apply.png

```
Apply complete! Resources: 104 added, 2 changed, 2 destroyed.

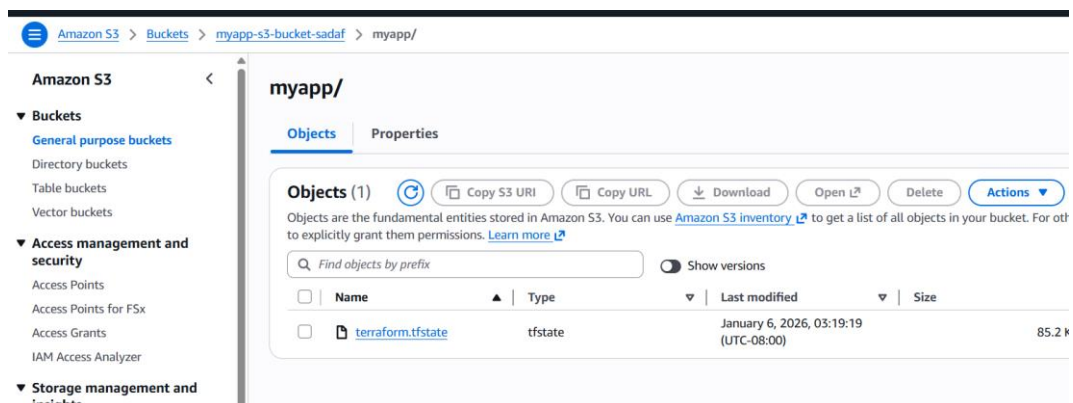
Outputs:

access_key_id = "AKIAV4QTXR5SZ5FM6YFE"

"Andy" = {
  "access_key_id" = "AKIAV4QTXR5S6ZIVOBOS"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Andy"
  "user_unique_id" = "AIDAV4QTXR5SY3UNMWV6K"
}
"Angela" = {
  "access_key_id" = "AKIAV4QTXR5SSTF0ZSE0"
```

6.6 Verify Terraform State File in S3

- Screenshot: task6_s3_tfstate_file.png



6.7 Verify Local Backup State File

- Screenshot: task6_local_state_backup.png

```
@SadafRiaz-077 → ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace 0 Jan 6 11:04 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 5430 Jan 6 11:04 terraform.tfstate.backup
@SadafRiaz-077 → ~/Lab13 $
```

6.8 Destroy Resources and Verify State

- Screenshot: task6_terraform_destroy.png

```
Destroy complete! Resources: 109 destroyed.
@SadafRiaz-077 → ~/Lab13 $
```

6.9 Verify Empty State in S3

- Screenshot: task6_s3_tfstate_destroyed.png

```
{
  "version": 4,
  "terraform_version": "1.5.4",
  "serial": 4,
  "lineage": "57af004f-32be-093c-3ddd-3f08ff040cef",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

Task 7 — Create Multiple Users from CSV File

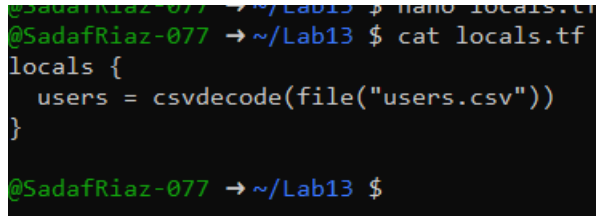
7.1 Create users.csv File

- Screenshot: task7_users_csv.png

```
@SadafRiaz-077 → ~/Lab13 $ nano users.csv
@SadafRiaz-077 → ~/Lab13 $ cat user.csv
cat: user.csv: No such file or directory
@SadafRiaz-077 → ~/Lab13 $ cat users.csv
user_name
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
@SadafRiaz-077 → ~/Lab13 $
```

7.2 Create locals.tf to Parse CSV

- Screenshot: task7_locals_tf.png

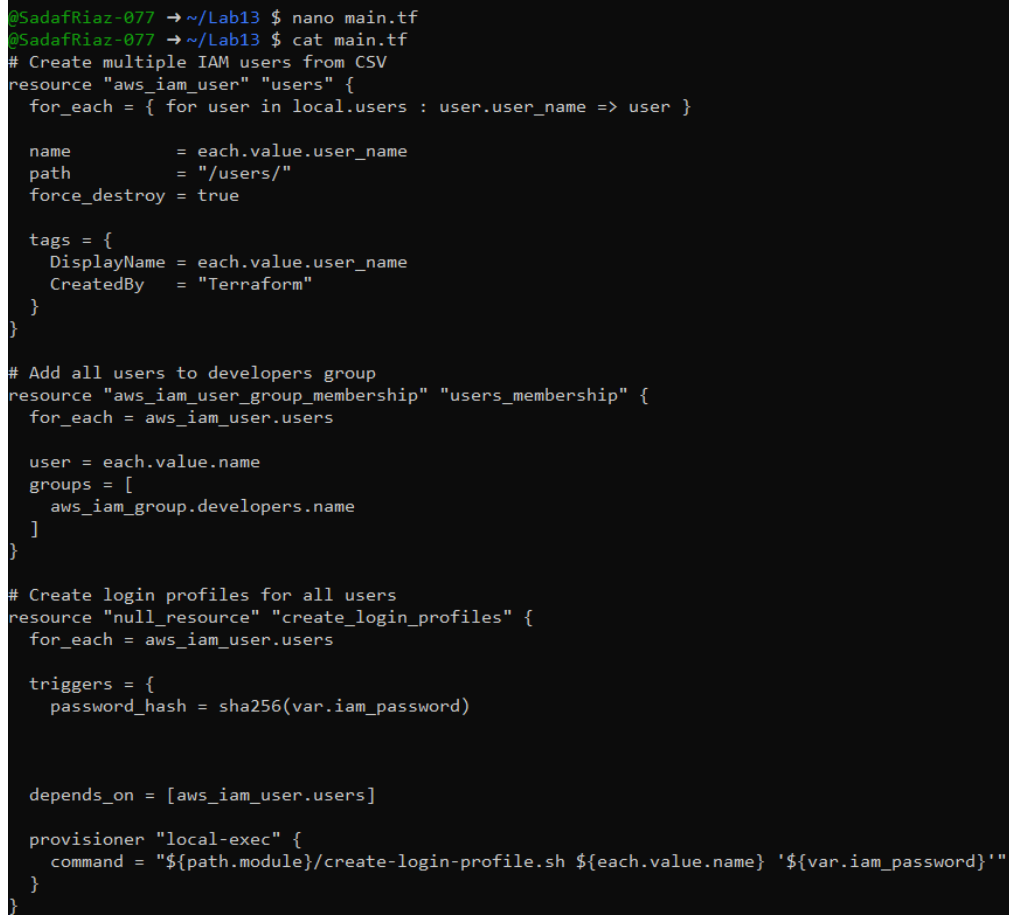


```
@SadafRiaz-077 → ~/Lab13 $ nano locals.tf
@SadafRiaz-077 → ~/Lab13 $ cat locals.tf
locals {
  users = csvdecode(file("users.csv"))
}

@SadafRiaz-077 → ~/Lab13 $
```

7.3 Update main.tf to Create Multiple Users

- Screenshot: task7_main_tf_multiple_users.png



```
@SadafRiaz-077 → ~/Lab13 $ nano main.tf
@SadafRiaz-077 → ~/Lab13 $ cat main.tf
# Create multiple IAM users from CSV
resource "aws_iam_user" "users" {
  for_each = { for user in local.users : user.user_name => user }

  name           = each.value.user_name
  path           = "/users/"
  force_destroy  = true

  tags = {
    DisplayName = each.value.user_name
    CreatedBy   = "Terraform"
  }
}

# Add all users to developers group
resource "aws_iam_user_group_membership" "users_membership" {
  for_each = aws_iam_user.users

  user = each.value.name
  groups = [
    aws_iam_group.developers.name
  ]
}

# Create login profiles for all users
resource "null_resource" "create_login_profiles" {
  for_each = aws_iam_user.users

  triggers = {
    password_hash = sha256(var.iam_password)
  }

  depends_on = [aws_iam_user.users]

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

7.4 Reinitialize Terraform

- Screenshot: task7_terraform_init.png

```
@SadafRiaz-077 → ~/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.
```

```
@SadafRiaz-077 → ~/Lab13 $
```

7.5 Apply Configuration to Create Multiple Users

- Screenshot: task7_terraform_apply.png

```
"Michael" = {  
  "access_key_id" = "AKIAV4QTXR5S63ZBRMJR"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Michael"  
  "user_unique_id" = "AIDAV4QTXR5SYAKDJNENN"  
}  
"Oscar" = {  
  "access_key_id" = "AKIAV4QTXR5STKZPEOXL"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Oscar"  
  "user_unique_id" = "AIDAV4QTXR5SURANCVZM"  
}  
"Pam" = {  
  "access_key_id" = "AKIAV4QTXR5SSG2PSCVY"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Pam"  
  "user_unique_id" = "AIDAV4QTXR5SYDOHS2LVX"  
}  
"Peter" = {  
  "access_key_id" = "AKIAV4QTXR5SUWCGSP55"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Peter"  
  "user_unique_id" = "AIDAV4QTXR5STBOXH3EWP"  
}  
"Phyllis" = {  
  "access_key_id" = "AKIAV4QTXR5SUH5TCPPI"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Phyllis"  
  "user_unique_id" = "AIDAV4QTXR5STGQQ3ZLW2"  
}  
"Robert" = {  
  "access_key_id" = "AKIAV4QTXR5SXO6CQNNQ"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Robert"  
  "user_unique_id" = "AIDAV4QTXR5SZSY5TGQHS"  
}  
"Ryan" = {  
  "access_key_id" = "AKIAV4QTXR5SQ67233PC"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Ryan"  
  "user_unique_id" = "AIDAV4QTXR5SQMAWTTHK"  
}  
"Stanley" = {  
  "access_key_id" = "AKIAV4QTXR5SYK64JZX3"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Stanley"  
  "user_unique_id" = "AIDAV4QTXR5S2N7B2NT3A"  
}  
"Toby" = {  
  "access_key_id" = "AKIAV4QTXR5S3O6T2QMZ"  
  "user_arn" = "arn:aws:iam::404842057573:user/users/Toby"  
  "user_unique_id" = "AIDAV4QTXR5SX6ZAT4FXT"  
}  
}
```

```
@SadafRiaz-077 → ~/Lab13 $
```


7.6 Display Output for All Users

- Screenshot: task7_terraform_output.png

```
@SadafRiaz-077 → ~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAV4QTXR5S67HLJ07X"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Andy"
    "user_unique_id" = "AIDAV4QTXR5S7RFU35QUD"
  }
  "Angela" = {
    "access_key_id" = "AKIAV4QTXR5SU4ANVX64"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Angela"
    "user_unique_id" = "AIDAV4QTXR5SYERLYXHR"
  }
  "Charles" = {
    "access_key_id" = "AKIAV4QTXR5S4BRUSBI4"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Charles"
    "user_unique_id" = "AIDAV4QTXR5S2ULTHVUZ3"
  }
  "Clark" = {
    "access_key_id" = "AKIAV4QTXR5SXZTL5ICP"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Clark"
    "user_unique_id" = "AIDAV4QTXR5S56F25DGOA"
  }
  "Creed" = {
    "access_key_id" = "AKIAV4QTXR5SUS3JZI5R"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Creed"
    "user_unique_id" = "AIDAV4QTXR5S4PG3PN7LZ"
  }
  "Darryl" = {
    "access_key_id" = "AKIAV4QTXR5SY6B0PDF0"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Darryl"
    "user_unique_id" = "AIDAV4QTXR5S43760UY4S"
  }
}
```

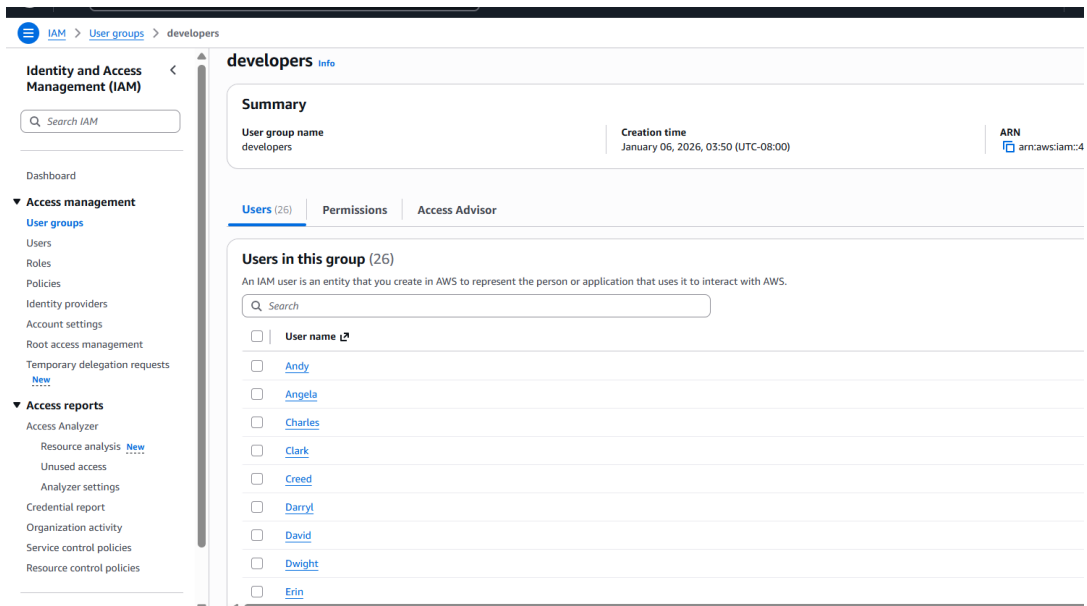
- task7_tfstate_secrets. Png

```
@SadafRiaz-077 → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
@SadafRiaz-077 → ~/Lab13 $
```

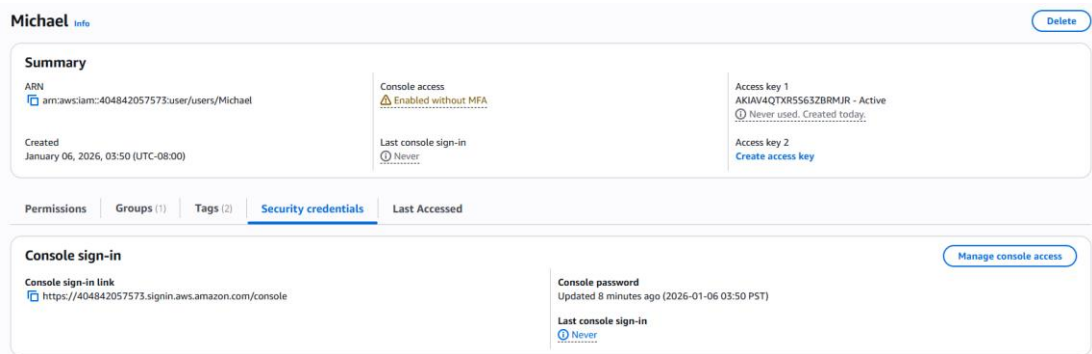
- task7_aws_console_all_users. Png

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age
Admin	/	0	1 hour ago	Passkey...	51 days	1 hour ago	Active - AKIAV4QTXRS...	1 days
Andy	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Angela	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Charles	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Clark	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Creed	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Darryl	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
David	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Dwight	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Erin	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Gabe	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Holly	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Jan	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Jim	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes
Joe	/users/	1	-	-	5 minutes	-	Active - AKIAV4QTXRS...	6 minutes

- task7_aws_console_group_members. Png



- task7_aws_console_user_access_key.png



- task7_s3_tfstate_multiple_users.png



Cleanup

8.1 Destroy All Resources

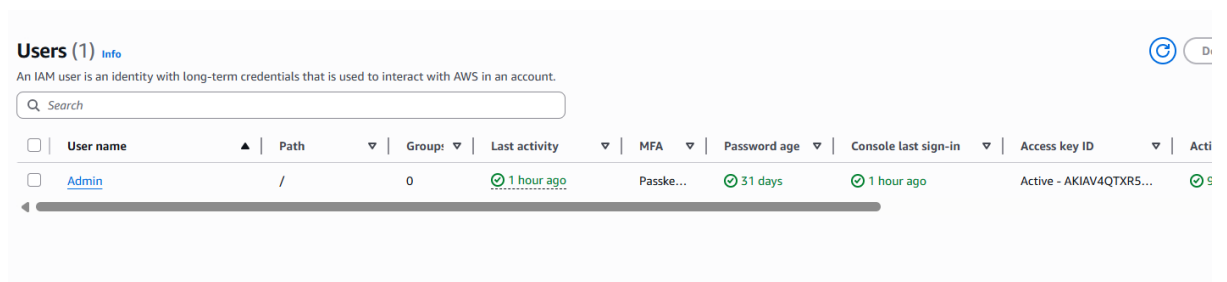
- **Screenshot:** cleanup_destroy_complete.png

```
aws_iam_user.users["Jesse"]: Destruction complete after 3s
aws_iam_user.users["Jim"]: Destruction complete after 3s
aws_iam_user.users["Angela"]: Destruction complete after 2s
aws_iam_user.users["Andy"]: Destruction complete after 2s
aws_iam_user.users["Kelly"]: Destruction complete after 13s
aws_iam_user.users["Darryl"]: Destruction complete after 9s
aws_iam_user.users["Oscar"]: Destruction complete after 6s

Destroy complete! Resources: 105 destroyed.
@SadafRiaz-077 → ~/Lab13 $
```

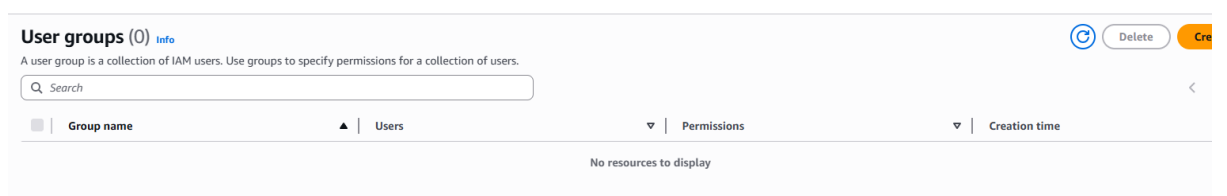
8.2 Verify Users Deleted in AWS Console

- **Screenshot:** cleanup_aws_console_users_deleted.png



8.3 Verify Group Deleted in AWS Console

- **Screenshot:** cleanup_aws_console_group_deleted.png



8.4 Verify Empty State in S3

- **Screenshot:** cleanup_s3_empty_state.png

```
{
  "version": 4,
  "terraform_version": "1.5.4",
  "serial": 7,
  "lineage": "57af004f-32be-093c-3ddd-3f08ff040cef",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

8.5 List All Project Files

- Screenshot: cleanup_final_files.png


```

@SadafRiaz-077 → ~/Lab13 $ ls -la
total 20540
drwxrwxr-x 3 codespace codespace 4096 Jan 6 11:49 .
drwxr-x--- 1 codespace codespace 4096 Jan 5 20:36 ..
drwxr-xr-x 3 codespace codespace 4096 Jan 6 11:49 .terraform
-rw-r--r-- 1 codespace codespace 2422 Jan 6 09:24 .terraform.lock.hcl
-rwxrwxr-x 1 codespace codespace 423 Jan 6 09:09 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace 51 Jan 6 11:33 locals.tf
-rw-rw-r-- 1 codespace codespace 2075 Jan 6 11:49 main.tf
-rw-rw-r-- 1 codespace codespace 0 Jan 6 11:04 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 5430 Jan 6 11:04 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace 20983285 Jan 5 20:15 terraform_1.5.4_linux_amd64.zip
-rw-rw-r-- 1 codespace codespace 168 Jan 6 11:34 users.csv
-rw-rw-r-- 1 codespace codespace 155 Jan 6 09:07 variables.tf
@SadafRiaz-077 → ~/Lab13 $

```

8.6 (Optional) Delete S3 Bucket

- Screenshot: cleanup_s3_bucket_deleted.png

 **Successfully emptied bucket "myapp-s3-bucket-sadaf"**
 View details below. If you want to delete this bucket, use the [delete bucket configuration](#).