

**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-rd5g5wqrq... musical waffle SadafRiaz-077/CC_Sad... main* Shutdown about 28 days ago
shiny-couscous-wrq7qxp9p... 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-jjp5p9v4px... 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-wrq7qxp9p... 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-69gxwq... 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 $
```

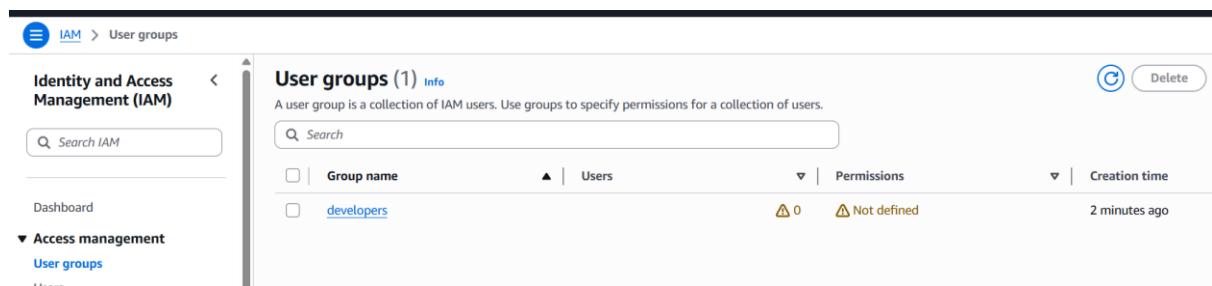
## 1.6 Display Group Details

- **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

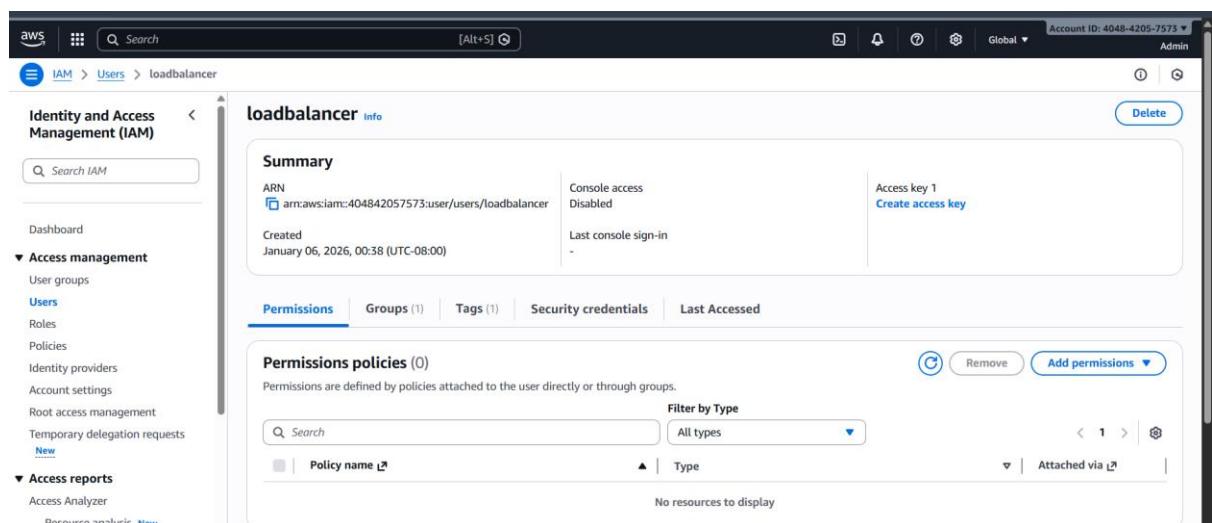
```

@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 $

```

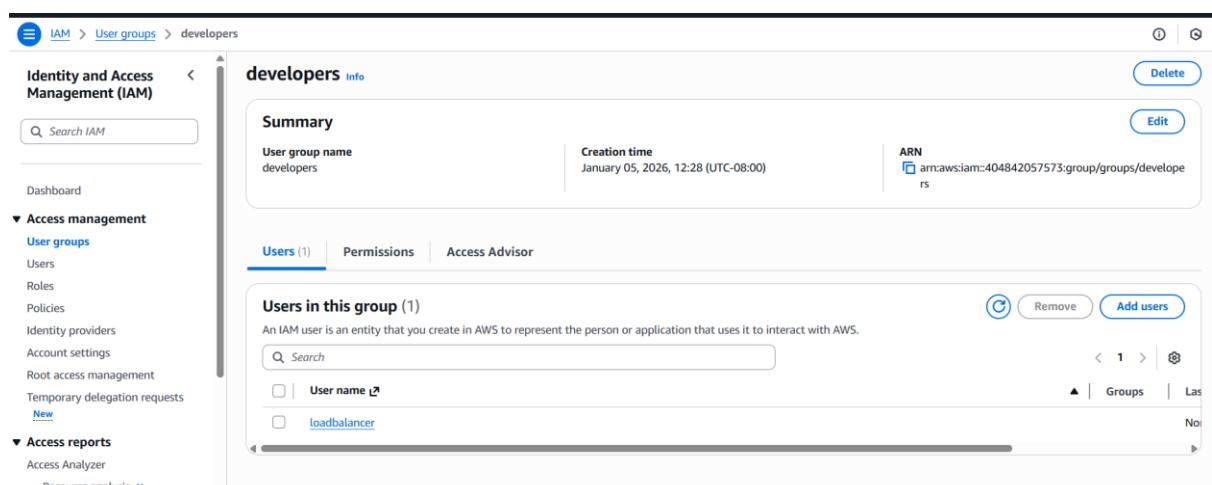
## 2.4 Verify User in AWS Console

- 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

The screenshot shows the AWS IAM User Groups page. The left sidebar is titled 'Identity and Access Management (IAM)' and includes sections for 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management, Temporary delegation requests), 'Access reports' (Access Analyzer, Resource analysis, Unused access), and a search bar. The main content area is titled 'developers info' under the 'Summary' section. It shows the user group name 'developers' and creation time 'January 05, 2026, 12:28 (UTC-08:00)'. An ARN is listed as 'arn:aws:iam::404842057573:group/groups/developers'. Below this, there are tabs for 'Users (1)', 'Permissions', and 'Access Advisor'. The 'Permissions' tab is selected, showing 'Permissions policies (2)'. It lists two policies: 'AmazonEC2FullAccess' (AWS managed, type: AWS managed, attached entities: 1) and 'IAMUserChangePassword' (AWS managed, type: AWS managed, attached entities: 3). There are buttons for 'Simulate', 'Remove', and 'Add permissions'.

## 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     = "1dontKnow"
}

@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_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}'"
  }
}

@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_user_group_membership.lb_membership: Refreshing state... [id=terraform-2026010608382027900000001]

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 $
```

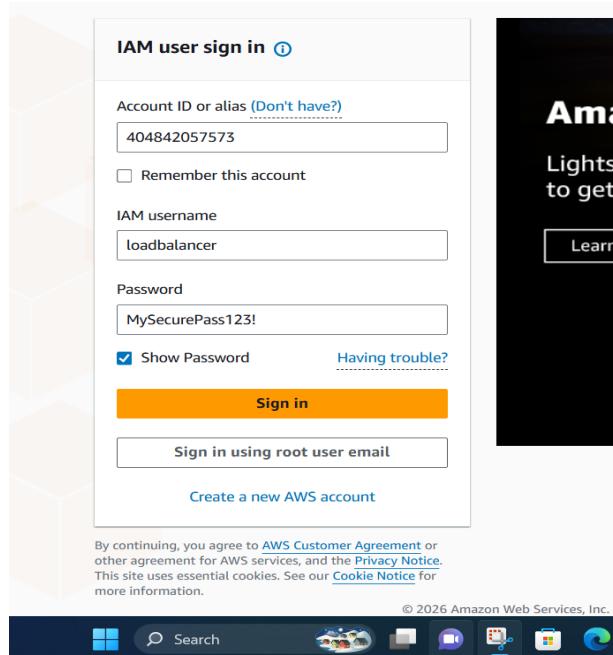
## 4.6 Verify Login Profile with AWS CLI

- **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 $
```

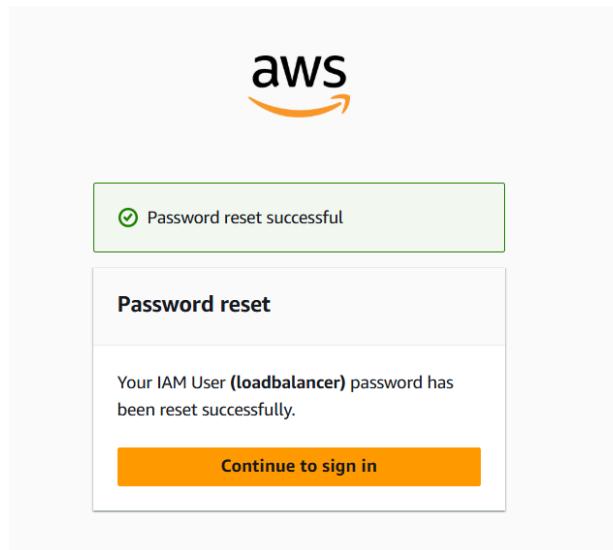
#### 4.7 Verify Login Profile in AWS Console

- **Screenshot:** task4\_aws\_console\_login.png



#### 4.8 Verify Password Reset Prompt in AWS Console

- **Screenshot:** task4\_aws\_console\_password\_reset.png



## Task 5 — Generate Access Keys for IAM User

### 5.1 Add Access Key Resource to main.tf

- **Screenshot:** task5\_main\_tf\_access\_keys.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_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
}

@SadafRiaz-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 = "AKIAV4QTXR5Z5FM6YFE"
access_key_secret = <sensitive>
@SadafRiaz-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 $
```

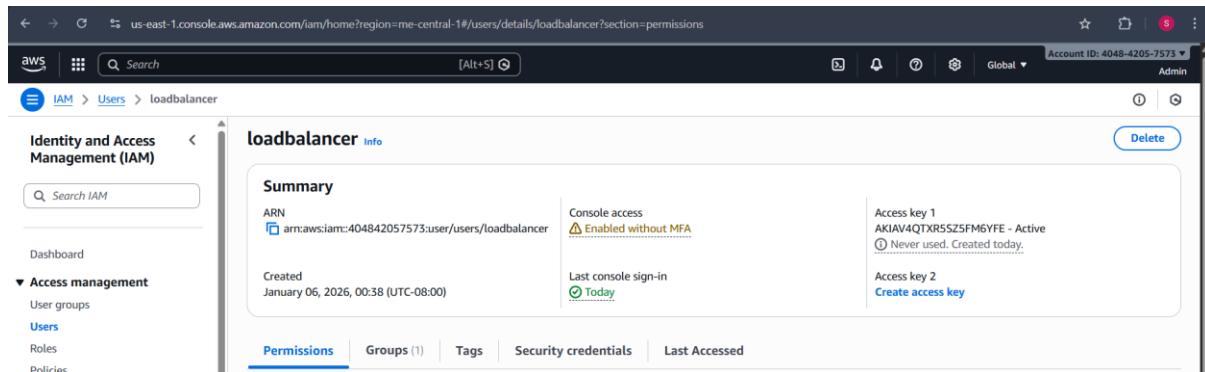
## 5.4 View Secret in Terraform State

- **Screenshot:** task5\_tfstate\_secret.png

```
@SadafRiaz-077 → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
    "access_key_secret": {
        "value": "6+TiH2VnzUliW9s/1iDw+4sMAW8lDkeKP8/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

The screenshot shows the AWS S3 console with a green success message: "Successfully created bucket 'myapp-s3-bucket-sadaf'. To upload files and folders, or to configure additional bucket settings, choose View details." Below the message, there are tabs for "General purpose buckets" and "Directory buckets". Under "General purpose buckets", there is one item: "General purpose buckets (1) Info". To the right of this item are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". A navigation bar at the top includes "Amazon S3 > Buckets" and "Account sna".

## 6.2 Enable Versioning in S3 Bucket

- **Screenshot:** task6\_s3\_bucket\_versioning.png

### Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your bucket. This makes it easier to recover from both unintended user actions and application failures. [Learn more ↗](#)

#### Bucket Versioning

- Disable  
 Enable

## 6.3 Add S3 Backend to main.tf

- **Screenshot:** task6\_main\_tf\_backend.png

```
ssadaf@RiaZ-077:~/Lab01 $ nano main.tf
ssadaf@RiaZ-077:~/Lab01 $ 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.

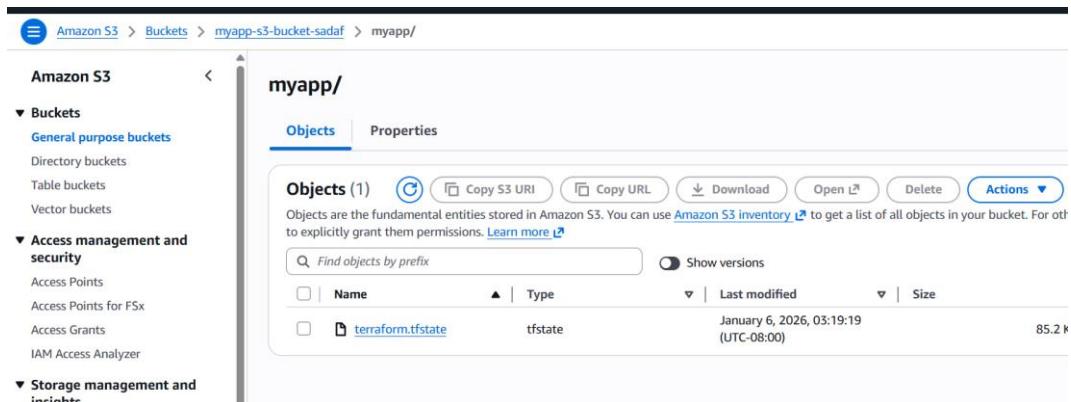
Outputs:

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" = "AKIAV4QTXR5SSTFOZSEO"
```

## 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" = "AIDAV4QTXR5SYAKDQNENN"
}
"Oscar" = {
  "access_key_id" = "AKIAV4QTXR5STKZPEOXL"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Oscar"
  "user_unique_id" = "AIDAV4QTXR5SURANCVOZM"
}
"Pam" = {
  "access_key_id" = "AKIAV4QTXR5SSG2PSCVY"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Pam"
  "user_unique_id" = "AIDAV4QTXR5SYD0HS2LVX"
}
"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" = "AIDAV4QTXR5STGQ3ZLW2"
}
"Robert" = {
  "access_key_id" = "AKIAV4QTXR5SX06CQNNQ"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Robert"
  "user_unique_id" = "AIDAV4QTXR5SZY5TGQHS"
}
"Ryan" = {
  "access_key_id" = "AKIAV4QTXR5SQ67233PC"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Ryan"
  "user_unique_id" = "AIDAV4QTXR5SVQMAWTTHK"
}
"Stanley" = {
  "access_key_id" = "AKIAV4QTXR5SYKG4JZX3"
  "user_arn" = "arn:aws:iam::404842057573:user/users/Stanley"
  "user_unique_id" = "AIDAV4QTXR5S2N7B2NT3A"
}
"Toby" = {
  "access_key_id" = "AKIAV4QTXR5S306T2QMZ"
  "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" = "AIDAV4QTXR57RFU35QUD"
  }
  "Angela" = {
    "access_key_id" = "AKIAV4QTXR5SU4ANVX64"
    "user_arn" = "arn:aws:iam::404842057573:user/users/Angela"
    "user_unique_id" = "AIDAV4QTXR5SYYERLYXHR"
  }
  "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" = "AKIAV4QTXR5SY6BOPDFO"
    "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

Users (28) <a href="#">Info</a>										
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.										
	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Action
Admin	/	0	-	1 hour ago	-	31 days	1 hour ago	Active - AKIAV4QTXR5...	9 days	<a href="#">Edit</a>
Andy	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Angela	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Charles	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Clark	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Creed	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Darryl	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Gabe	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Holly	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Jen	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
Jim	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-
In	/users/	1	-	-	-	5 minutes	-	Active - AKIAV4QTXR5...	6 minutes	-

- **task7\_aws\_console\_group\_members.** Png

The screenshot shows the AWS IAM User Groups interface. On the left, a sidebar navigation includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups' selected), 'Access reports', and 'New'. The main content area displays the 'developers' user group. It shows a summary with the user group name 'developers', creation time 'January 06, 2026, 03:50 (UTC-08:00)', and ARN 'arn:aws:iam::4'. Below this, tabs for 'Users (26)', 'Permissions', and 'Access Advisor' are visible. The 'Users in this group' section lists 26 members: Andy, Angela, Charles, Clark, Creed, Darryl, David, Dwight, and Erin.

- task7\_aws\_console\_user\_access\_key.png

The screenshot shows the AWS IAM User details page for 'Michael'. Under the 'Security credentials' tab, it displays two access keys: 'Access key 1' (Active, created today) and 'Access key 2' (Never used). Below this, the 'Console sign-in' section shows a 'Console sign-in link' (https://404842057573.sigin.aws.amazon.com/console) and a 'Last console sign-in' (Never). A 'Manage console access' button is also present.

- task7\_s3\_tfstate\_multiple\_users.png

```
{
  "version": 4,
  "terraform_version": "1.5.4",
  "serial": 6,
  "lineage": "57af004f-32be-093c-3ddd-3f08ff040cef",
  "outputs": {
    "all_access_key_secrets": {
      "value": {
        "Andy": "wq6uz8dgEHXbVJLa05T8vYuNjGUgrS7QGaVPKsW8",
        "Angela": "Sniv@AmnScusBrvzQ40fgMznCb+Qos2fpLgenBW",
        "Charles": "WOUkZMWAjO0IP+aeWxWaeOR5fTrM+h0iVKKzBoY",
        "Clark": "91oKyusKAetg913jgBPPaZeEosKYcJNRnxCKalU",
        "Creed": "mhqjfS68P6SnfGss2OK/T21WQkzF8yNHXm6x0V",
        "Darryl": "XHFpgAzNtVq3jg8WzAdJL7324+M52jprgsyuQPrF",
        "David": "xGnryBh/WtqZesQZOvMBrjsem89D3lcvTG59wIu",
        "Dwight": "OTYbfm5/6txov2MT8qk1ItdyJOhpYgm7MGUmYI",
        "Erin": "qjjQB8RvEUh0vX73fYea9cyEbQ8/Wg8JML9kwTQI",
        "Gabe": "p9o26rvVg1kony6E5DH7TDISKzRh+BHTYnh1ojg",
        "Holly": "uoGut4h9ufh1Tv3TktPT+1szwGDDPTO/deoUpHUb",
        "Jan": "n1FIsxMcojkCuUkVxxSNWmftwGEmLbWpJBaefjn",
        "Jim": "kyYQL5NAJjiT9RF+Bk47eOYudaJ0CC8QTmuWElp",
        "Jo": "Mql+XX53gqwqlBxOr6cwM3wQlwLhzs0zfFnmUr",
        "Kelly": "zLdvRnc/GPh09Gg+A9BAv8FP6j/LogSB0ku09js",
        "Kevin": "16UiJ7tx8TiuXbcv1B16X1e5F/2f95Sz0wTvHdx",
        "Meredith": "uH9G/4JCS2YvSsRLn0Fnmf/fz7kngdRJVsTTVpys",
        "Michael": "9UoklKhf/RfStzi2l091zaAvW7K6faP9dxF1GMW",
        "Oscar": "7i3fx6/SDvc7yRGlx7stpR59hWIXlujv2teUh1BN",
        "Pam": "a/76Vmhd5R6bygbCnoPPwK/gbFxRBVpe5BvQsVW",
        "Peter": "5rsTE6zmknAIRTc85SriuQXn4fgUohisx0v9k",
        "Phyllis": "oqYMYhiQ70154EudJOUEuvevkwmEwI0bzklm1lgR",
        "Robert": "ChPdrm051hZFyfuEbWGalKR6WQ1XF3YLxkkG0uM9d",
        "Ryan": "qNopV2cGGowu0DC1CZMqkdj+7pl3tkB0FgtfNCEg",
        "Stanley": "KnivIcScT5LB4KZ1NWE230+8n1ikgDo6JpSmquJ",
        "Toby": "YC9nGFR8ziYouphEeJZwjj7NUe+YhGSp+Q2d60J"
      }
    },
    "type": [
      "object",
      {
        "Andy": "string",
        "Angela": "string",
        "Charles": "string",
        "Clark": "string",
        "Creed": "string",
        "Darryl": "string"
      }
    ]
}
```

## Cleanup

### 8.1 Destroy All Resources

- **Screenshot:** cleanup\_destroy\_complete.png

```
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

The screenshot shows the AWS IAM 'Users' page. It displays a single user entry for 'Admin'. The table includes columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, Access key ID, and Activity. The 'Last activity' column shows '1 hour ago' with a green checkmark. The 'Password age' column shows '31 days' with a green checkmark. The 'Console last sign-in' column shows '1 hour ago' with a green checkmark. The 'Access key ID' column shows 'Active - AKIAV4QTXR5...' with a green checkmark. The 'Activity' column shows a green checkmark.

### 8.3 Verify Group Deleted in AWS Console

- **Screenshot:** cleanup\_aws\_console\_group\_deleted.png

The screenshot shows the AWS IAM 'User groups' page. It displays a table with no entries, indicating 'No resources to display'. The table has columns for Group name, Users, Permissions, and Creation time.

### 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

```
pos@pos-OptiPlex-5090:~/Desktop$ ls -la  
@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](#).