

---

## ***LAB 13***

---

**NAME: SHUMAIL ZAHRA**

**REGISTRATION #: 2023-BSE-061**

**DEPARTMENT: BSE(5B)**

### **LAB TASK**

**Lab 13 – Terraform IAM Management with AWS**

**Task 0 Lab Setup (Codespace & GH CLI)**

## task0\_codespace\_create\_and\_list

```
Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh repo create CC-ShumailZahra_2023-BSE-061_Lab13 --public
Created repository 23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13 on github.com
https://github.com/23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh codespace create --repo 23-22411-rgb/ CC-ShumailZahra_2023-BSE-061_Lab13
the command accepts no arguments

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh codespace list


| NAME             | DISPLAY NAME  | REPOSITORY    | BRANCH | STATE     | CREATED AT    |
|------------------|---------------|---------------|--------|-----------|---------------|
| special-space... | special sp... | 23-22411-0... | main*  | Shutdown  | about 1 mo... |
| supreme-barna... | supreme ba... | 23-22411-0... | main*  | Shutdown  | about 12 d... |
| glorious-trib... | glorious t... | 23-22411-0... | main   | Shutdown  | about 1 da... |
| fantastic-spa... | fantastic ... | 23-22411-0... | main   | Available | about 1 da... |



Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh codespace create --repo 23-22411-rgb/CC-ShumailZahra_2023-BSE-061_Lab13
error getting repository: HTTP 404: Not Found (https://api.github.com/repos/23-22411-rgb/cc-shumailzahra_2023-bse-061_lab13)

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh codespace create --repo 23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13
Codespaces usage for this repository is paid for by 23-22411-061-rgb
error getting devcontainer.json paths: HTTP 400: The 'ref' provided was invalid. Please specify a valid branch name or commit SHA (https://api.github.com/repositories/1129057558/codespaces/devcontainers?per_page=100&ref=main)

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ gh codespace create --repo 23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13
Codespaces usage for this repository is paid for by 23-22411-061-rgb
error getting devcontainer.json paths: HTTP 400: The 'ref' provided was invalid. Please specify a valid branch name or commit SHA (https://api.github.com/repositories/1129057558/codespaces/devcontainers?per_page=100&ref=main)

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ git clone https://github.com/23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13.git
Cloning into 'CC-ShumailZahra_2023-BSE-061_Lab13'...
warning: You appear to have cloned an empty repository.

Syed@DESKTOP-S50GK51 MINGW64 ~ (main)
$ cd CC-ShumailZahra_2023-BSE-061_Lab13

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ echo "# Lab 13" > README.md

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ git commit -m "Initial commit for Lab 13"
[main (root-commit) df45e82] Initial commit for Lab 13
1 file changed, 1 insertion(+)
 create mode 100644 README.md

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 48.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13.git
 * [new branch]      main -> main

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ gh codespace create --repo 23-22411-061-rgb/CC-ShumailZahra_2023-BSE-061_Lab13
Codespaces usage for this repository is paid for by 23-22411-061-rgb
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
miniature-space-garbanzo-qvv55r6r4q7f4w6j

Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ gh codespace list


| NAME                    | DISPLAY NAME          | REPOSITORY            | BRANCH | STATE     | CREATED AT            |
|-------------------------|-----------------------|-----------------------|--------|-----------|-----------------------|
| special-space-funic...  | special space funi... | 23-22411-061-rgb/C... | main*  | Shutdown  | about 1 month ago     |
| supreme-barnacle-x5g... | supreme barnacle      | 23-22411-061-rgb/C... | main*  | Shutdown  | about 12 days ago     |
| glorious-tribble-wrq... | glorious tribble      | 23-22411-061-rgb/C... | main   | Shutdown  | about 1 day ago       |
| fantastic-space-traf... | fantastic space train | 23-22411-061-rgb/C... | main   | Available | about 1 day ago       |
| miniature-space-garb... | miniature space ga... | 23-22411-061-rgb/C... | main   | Available | less than a minute... |



Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ |
```

## task0\_codespace\_ssh\_connected

```
Syed@DESKTOP-S50GK51 MINGW64 ~/CC-ShumailZahra_2023-BSE-061_Lab13 (main)
$ gh codespace ssh -c miniature-space-garbanzo-qvv55r6r4q7f4w6j
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Documentation:  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.

@23-22411-061-rgb -> /workspaces/CC-ShumailZahra_2023-BSE-061_Lab13 (main) $ |
```

## Task 1 — Create IAM Group and Output Details

### task1\_project\_directory

```
applicable file:
@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ mkdir -p ~/Lab13
@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
@23-22411-061-rgb → ~/Lab13 $ |
```

task1\_file\_created

```
@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ touch main.tf
@23-22411-061-rgb → ~/Lab13 $ touch main.tf
@23-22411-061-rgb → ~/Lab13 $ |
```

task1\_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
  }
}
```

task1\_terraform\_init

```

@23-22411-061-rgb → ~/Lab13 $ vim main.tf
@23-22411-061-rgb → ~/Lab13 $ terraform init
-bash: terraform: command not found
@23-22411-061-rgb → ~/Lab13 $ sudo apt-get update
Get:1 https://dl.yarnpkg.com/debian stable InRelease
Get:2 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease [3961 B]
Get:3 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:4 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease [3600 B]
Get:6 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable/main amd64 Packages [4557 B]
Get:7 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [77.6 kB]
Get:8 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [643 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1752 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1183 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2898 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2130 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [8059 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Fetched 35.5 MB in 6s (6060 kB/s)
Reading package lists... Done
@23-22411-061-rgb → ~/Lab13 $ sudo apt-get install -y gnupg software-properties-common curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17.3).
gnupg set to manually installed.
software-properties-common is already the newest version (0.99.49.3).
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
@23-22411-061-rgb → ~/Lab13 $ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
@23-22411-061-rgb → ~/Lab13 $ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
@23-22411-061-rgb → ~/Lab13 $ sudo apt-get update
Get:1 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]

Selecting previously unselected package terraform.
(Reading database ... 58629 files and directories currently installed.)
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
Last login: Tue Jan  6 15:03:06 2026 from ::1
^[[AA][A@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
@23-22411-061-rgb → ~/Lab13 $ terraform version
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
Last login: Tue Jan  6 15:03:06 2026 from ::1
^[[AA][A@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
@23-22411-061-rgb → ~/Lab13 $ terraform version

^[[AA][A@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
@23-22411-061-rgb → ~/Lab13 $ terraform version

Terraform v1.14.3
on linux_amd64
@23-22411-061-rgb → ~/Lab13 $
@23-22411-061-rgb → ~/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.

```

task1\_terraform\_apply

```
@23-22411-061-rgb → ~/Lab13 $ terraform apply -auto-approve

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.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 1s [id=developers]

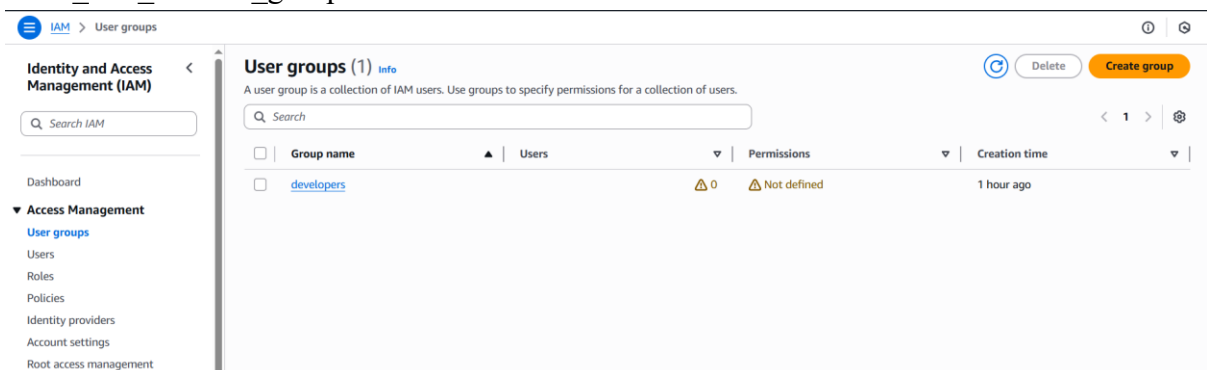
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
```

task1\_terraform\_output

```
@23-22411-061-rgb → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
```

task1\_aws\_console\_group



## Task 2 — Create IAM User with Group Membership

task2\_main\_tf\_user

```

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
  }
}

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
  }
}

```

## task2\_terraform\_apply

```

%23-22411-061-rgb → ~/Lab13 $ terraform apply -auto-approve
^[[A^[[A^[[aws_iam_group.developers: Refreshing state... [id=developers]

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_user.lb will be created
+ resource "aws_iam_user" "lb" {
  + arn = (known after apply)
  + force_destroy = true
  + id = (known after apply)
  + name = "loadbalancer"
  + path = "/users/"
  + tags = {
    + "DisplayName" = "Load Balancer"
  }
  + tags_all = {
    + "DisplayName" = "Load Balancer"
  }
  + unique_id = (known after apply)
}

# aws_iam_user_group_membership.lb_membership will be created
+ 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 1s [id=terraform-20260106190155394100000001]

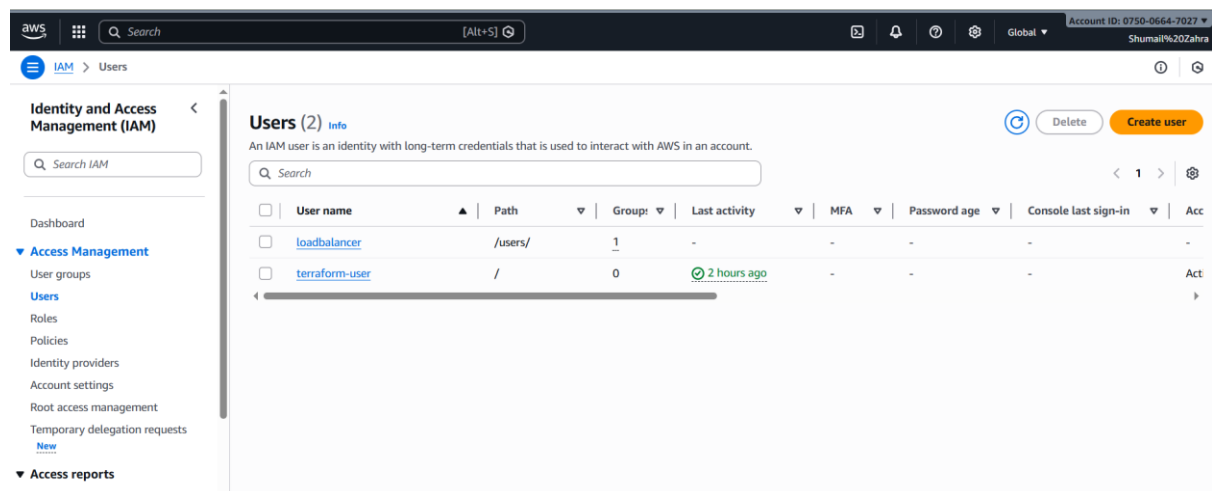
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

```

## task2\_terraform\_output

```
@23-22411-061-rgb → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYHBO2J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@23-22411-061-rgb → ~/Lab13 $ |
```

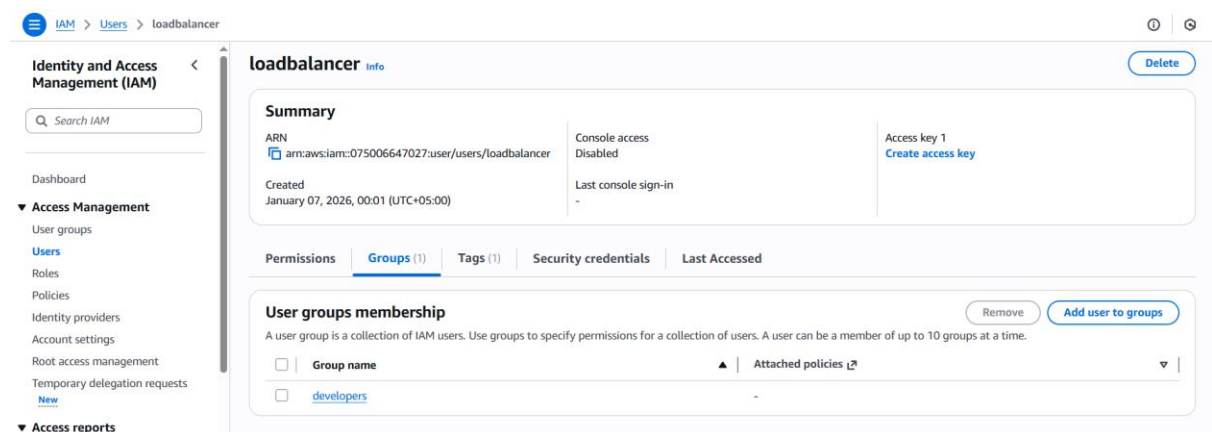
## task2\_aws\_console\_user



The screenshot shows the AWS IAM console 'Users' page. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access Management, User groups, Roles, Policies, Identity providers, Account settings, Root access management, and Temporary delegation requests. The main content area shows a list of users. The 'loadbalancer' user is selected, and its details are displayed on the right. The details include the user's ARN, console access status (Disabled), and a link to create an access key. The 'User groups membership' section shows that the user is a member of the 'developers' group.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Acc
loadbalancer	/users/	1	-	-	-	-	-
terraform-user	/	0	2 hours ago	-	-	-	Act

## task2\_aws\_console\_user\_groups



The screenshot shows the AWS IAM console 'loadbalancer' user details page. The left sidebar is the same as the previous screenshot. The main content area shows the user's summary, including their ARN, console access status (Disabled), and a link to create an access key. The 'User groups membership' section shows that the user is a member of the 'developers' group. The 'Groups' tab is selected, and the 'developers' group is listed as the user's group.

Group name	Attached policies
developers	-

## Task 3 — Attach Policies to IAM Group

### task3\_main\_tf\_policies

```
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"
}
```

### task3\_terraform\_apply

```
833-22411-091-705 ~ -/Lab13 $ terraform apply -auto-approve
aws_iam_group.developers: Refreshing state... [id=developers]
aws_iam_user.lb: Refreshing state... [id=loadbalancer]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260106190155394100000001]

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.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.change_password: Creating...
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creating...
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-20260106190733100700000001]
aws_iam_group_policy_attachment.change_password: Creation complete after 1s [id=developers-20260106190733105100000002]

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

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2R3GN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYH802J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

### task3\_aws\_console\_policies

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with sections for Identity and Access Management, Access Management, and Access reports. The main content area is titled 'developers' and includes a 'Summary' card with the user group name, creation time, and ARN. Below this is a 'Permissions' tab showing two attached policies: 'AmazonEC2FullAccess' and 'IAMUserChangePassword', both of which are AWS managed policies.

Policy name	Type	Attached entities
AmazonEC2FullAccess	AWS managed	1
IAMUserChangePassword	AWS managed	1

## Task 4 — Create Login Profile for IAM User

task4\_variables.tf

codespace@codespaces-4a71ad: / | ad: vim variables.tf

```
variable "iam_password" {  
  description = "Temporary password for the IAM user"  
  type        = string  
  sensitive   = true  
  default     = "IdontKnow"  
}
```

task4\_create\_login\_script

codespace@codespaces-4a71ad: vim 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
```

task4\_chmod\_script

```
@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13  
@23-22411-061-rgb → ~/Lab13 $ vim create-login-profile.sh  
@23-22411-061-rgb → ~/Lab13 $ chmod +x create-login-profile.sh  
@23-22411-061-rgb → ~/Lab13 $ |
```

task4\_main\_tf\_login\_profile

codespace@codespaces-4a71ad: vim main.tf

```
    group_name = aws_iam_group.developers.name
    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 "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_group_policy_attachment" "developer_ec2_fullaccess" {
  group = aws_iam_group.developers.name
  policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

-- INSERT --
```

## task4 terraform apply

```
023-22411-061-rgb → ~/Lab13 $ vim main.tf
023-22411-061-rgb → ~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"

Error: Inconsistent dependency lock file

The following dependency selections recorded in the lock file are inconsistent with the current configuration:
- provider registry.terraform.io/hashicorp/null: required by this configuration but no version is selected

To update the locked dependency selections to match a changed configuration, run:
  terraform init -upgrade

023-22411-061-rgb → ~/Lab13 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/null...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- Installing hashicorp/null v3.2.4...
- Installed hashicorp/null v3.2.4 (signed by HashiCorp)
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
023-22411-061-rgb → ~/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.developer_ec2_fullaccess: Refreshing state... [id=developers-20260106190733100700000001]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260106190733105100000002]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260106190155394100000001]

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:

codespace@codespaces-4a71ad: /bin/bash
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:

# 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=5174224382095500929]

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

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYH802J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

## task4\_aws\_cli\_verify

```
@23-22411-061-rgb → ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
  "LoginProfile": {
    "UserName": "loadbalancer",
    "CreateDate": "2026-01-06T19:52:28+00:00",
    "PasswordResetRequired": true
  }
}
```

## task4\_aws\_console\_login

The image shows two screenshots of the AWS console. The top screenshot is the IAM user sign-in page for the user 'loadbalancer'. It displays the account ID '075006647027', the username 'loadbalancer', and a password field. A 'Sign in' button is visible. To the right is an Amazon Lightsail advertisement. The bottom screenshot is the AWS Management Console home page for the 'eu-north-1' region. It shows a 'Recently visited' section with no services listed, an 'Applications' section with a red error message 'Access denied to servicecatalog:ListApplications', and a 'Cost and usage' section.

**IAM user sign in**

Account ID or alias ([Don't have?](#))  
075006647027

☒ Remember this account

IAM username  
loadbalancer

Password  
\*\*\*\*\*

☐ Show Password [Having trouble?](#)

**Sign in**

[Sign in using root user email](#)

[Create a new AWS account](#)

**Amazon Lightsail**

Lightsail is the easiest way to get started on AWS

[Learn more »](#)

**Console Home**

[Reset to default layout](#) [+ Add widgets](#)

**Recently visited**

No recently visited services

Explore one of these commonly visited AWS services.

[EC2](#) [S3](#) [Aurora and RDS](#) [Lambda](#)

[View all services](#)

**Applications (0)**

Region: Europe (Stockholm)

Select Region: eu-north-1 (Current Region) [Find applications](#)

Name	Description	Region	Originati.
Access denied to servicecatalog:ListApplications			

[Diagnose with Amazon Q](#)

[Go to myApplications](#)

**Welcome to AWS** [Getting started with](#)

**AWS Health**

**Cost and usage**

Current month [Cost breakdown](#)

## task4\_aws\_console\_password\_reset

**Password reset** ⓘ

Your account **(075006647027)** 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

☐ Show Password

Confirm New Password

Confirm Password Change

[Sign in to a different account](#)

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

## Task 5 — Generate Access Keys for IAM User

## task5\_main tf access keys

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

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"
}

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
}
```

## task5\_terraform\_apply

```
823-22411-061-rgb → ~/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=5174224382095500929]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260106190733105100000002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260106190733100700000001]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260106190155394100000001]

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_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=AKIARC5V6TLZ4UJHALZF]

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

Outputs:
access_key_id = "AKIARC5V6TLZ4UJHALZF"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYH802J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

## task5\_terraform\_output

```
@23-22411-061-rgb → ~/Lab13 $ terraform output
access_key_id = "AKIARC5V6TLZ4UJHALZF"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/deve1opers"
  "group_name" = "deve1opers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYHB02J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@23-22411-061-rgb → ~/Lab13 $ |
```

task5\_tfstate\_secret

```
@23-22411-061-rgb → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
"access_key_secret": {
  "value": "vwm1r/7Rh4bAu8TLJar/weCiL4E1qk8I15qt//aT",
  "type": "string",
  "sensitive": true
},
"group_details": {
  "value": {
    "group_arn": "arn:aws:iam::075006647027:group/groups/deve1opers",
    "group_name": "deve1opers",
    "unique_id": "AGPARC5V6TLZ2RJGN6RRO"
  },
},
@23-22411-061-rgb → ~/Lab13 $ |
```

task5\_aws\_console\_access\_keys

**Access keys (1)** [Create access key](#)

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](#)

<p><b>AKIARC5V6TLZ4UJHALZF</b></p> <p><b>Description</b></p> <p>-</p> <p><b>Last used</b></p> <p>None</p> <p><b>Last used region</b></p> <p>N/A</p>	<p><b>Status</b></p> <p>Active</p> <p><b>Created</b></p> <p>5 minutes ago</p> <p><b>Last used service</b></p> <p>N/A</p>
---	--

[Actions](#)

## Task 6 — Implement Terraform Remote State with S3

task6\_s3\_bucket\_create

**Bucket type** [Info](#)

☒ **General purpose**

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

**Bucket name** [Info](#)

myapp-s3-buckett

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#)

**Copy settings from existing bucket - optional**

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

**Object Ownership** [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

## task6\_s3\_bucket\_versioning

- ☒ **Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☒ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

### 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 Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

#### Bucket Versioning

- ☐ Disable  
☒ Enable

### Tags - optional

You can use bucket tags to analyze, manage and specify permissions for a bucket. [Learn more](#)

- i** You can use s3:ListTagsForResource, s3:TagResource, and s3:UntagResource APIs to manage tags on S3 general purpose buckets for access control in addition to cost allocation and resource organization. To ensure a seamless transition, please provide permissions to s3:ListTagsForResource, s3:TagResource, and s3:UntagResource actions. [Learn more](#)

## task6\_main\_tf\_backend

codespace@codespaces-4a71ad: vim main.tf

```
terraform {
  backend "s3" {
    bucket = "myapp-s3-bucketttt"
    key    = "myapp/terraform.tfstate"
    region = "me-central-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
  }
}

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
  }
}
```

## task6\_terraform\_init\_migrate

```
@23-22411-061-rgb → ~/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

Releasing state lock. This may take a few moments...

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.
@23-22411-061-rgb → ~/Lab13 $ |
```

## task6\_terraform\_apply

```
@23-22411-061-rgb → ~/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.developer_ec2_fullaccess: Refreshing state... [id=developers-20260106190733100700000001]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260106190733105100000002]
null_resource.create_login_profile: Refreshing state... [id=5174224382095500929]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIARC5V6TLZ4UJHALZF]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260106190155394100000001]

No changes. Your infrastructure matches the configuration.

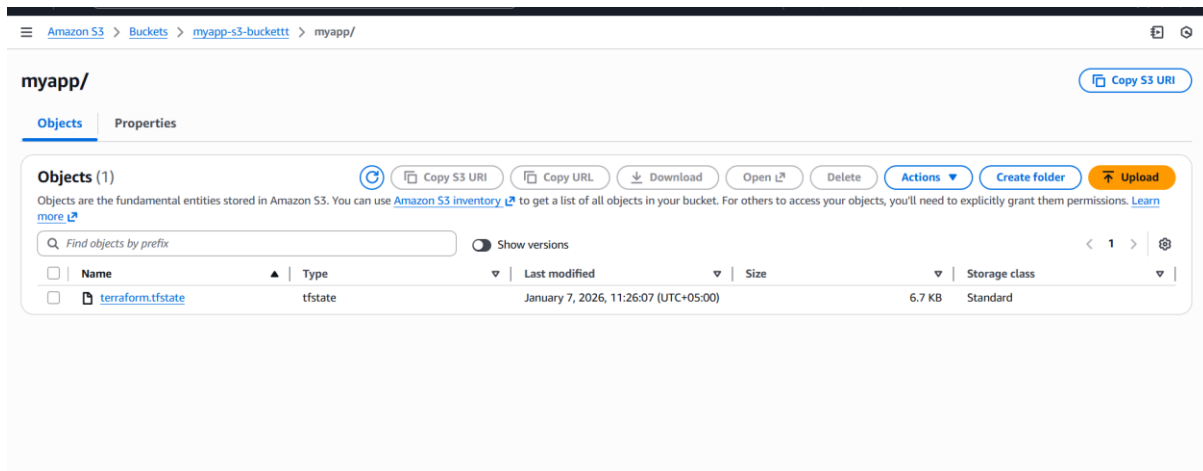
Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
Releasing state lock. This may take a few moments...

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

Outputs:

access_key_id = "AKIARC5V6TLZ4UJHALZF"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPARC5V6TLZ2RJGN6RR0"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYHBO2J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@23-22411-061-rgb → ~/Lab13 $ |
```

## task6\_s3\_tfstate\_file



task6\_local\_state\_backup

```

}
@23-22411-061-rgb → ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace 0 Jan 7 06:26 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 7 06:26 terraform.tfstate.backup
@23-22411-061-rgb → ~/Lab13 $ |

```

task6\_terraform\_destroy

```

# aws_iam_user_group_membership.tb_membership will be destroyed
- resource "aws_iam_user_group_membership" "lb_membership" {
  - groups = [
    - "developers",
  ] -> null
  - id     = "terraform-20260106190155394100000001" -> null
  - user   = "loadbalancer" -> null
}

# null_resource.create_login_profile will be destroyed
- resource "null_resource" "create_login_profile" {
  - id     = "5174224382095500929" -> null
  - triggers = {
    - "password_hash" = (sensitive value)
    - "user"          = "loadbalancer"
  } -> null
}

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

Changes to Outputs:
- access_key_id     = "AKIARC5V6TLZ4UJHALZF" -> null
- access_key_secret = (sensitive value) -> null
- group_details     = {
  - group_arn = "arn:aws:iam::075006647027:group/groups/developers"
  - group_name = "developers"
  - unique_id  = "AGPARC5V6TLZ2RJGN6RRO"
} -> null
- user_details     = {
  - unique_id = "AIDARC5V6TLZSIAHYB02j"
  - user_arn  = "arn:aws:iam::075006647027:user/users/loadbalancer"
  - user_name = "loadbalancer"
} -> null
null_resource.create_login_profile: Destroying... [id=5174224382095500929]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_access_key.tb_access_key: Destroying... [id=AKIARC5V6TLZ4UJHALZF]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260106190733105100000002]
aws_iam_user_group_membership.tb_membership: Destroying... [id=terraform-20260106190155394100000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260106190733100700000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 0s
aws_iam_user_group_membership.tb_membership: Destruction complete after 0s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 0s
aws_iam_access_key.tb_access_key: Destruction complete after 0s
aws_iam_user.lb: Destroying... [id=loadbalancer]
aws_iam_group.developers: Destruction complete after 1s
aws_iam_user.lb: Destruction complete after 3s
Releasing state lock. This may take a few moments...

Destroy complete! Resources: 7 destroyed.
@23-22411-061-rgb → ~/Lab13 $ |

```

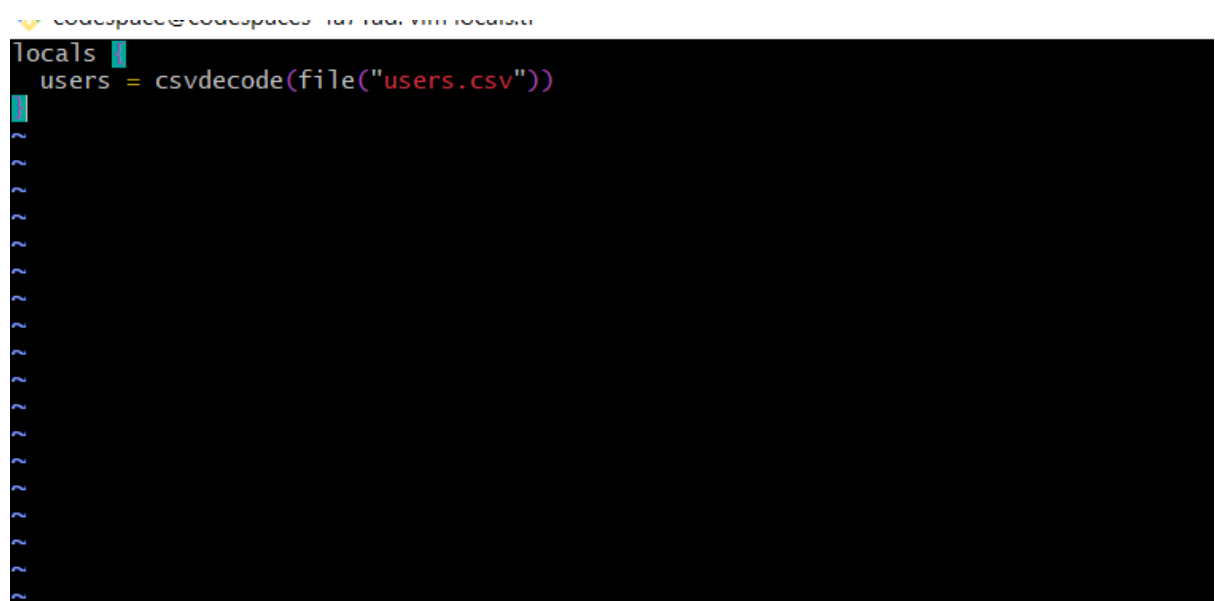
task6\_s3\_tfstate\_destroyed

```
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 2,
  "lineage": "ec9691c6-13e2-9979-d0f3-2e5ace0be39a",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

### Task 7 — Create Multiple Users from CSV File


task7\_locals.tf

```
code@code: /root/.terraform.d/modules
```



```
locals {
  users = csvdecode(file("users.csv"))
}
```

task7\_users\_csv

 codespace@codespaces-4a71ad: vim 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
```

```
~
~
~
~
~
~
~
```

task7\_main\_tf\_multiple\_users

```
resource "aws_iam_access_key" "users_access_keys" {
  for_each = aws_iam_user.users
  user     = each.value.name
}

#####
# OUTPUTS
#####
output "group_details" {
  value = {
    group_name = aws_iam_group.developers.name
    group_arn  = aws_iam_group.developers.arn
  }
}

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" {
  value = {
    for user_name, key in aws_iam_access_key.users_access_keys :
      user_name => key.secret
  }
  sensitive = true
}

:wq!|
```

task7\_terraform\_init

```
023-22411-061-rgb → ~/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.
023-22411-061-rgb → ~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"
```

task7\_terraform\_apply

```
023-22411-061-rgb → ~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"
```

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_access_key.users_access_keys["Andy"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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                  = "Andy"
}

# aws_iam_access_key.users_access_keys["Angela"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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                  = "Angela"
}

# aws_iam_access_key.users_access_keys["Charles"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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                  = "Charles"
}

# aws_iam_access_key.users_access_keys["Clark"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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                  = "Clark"
}
```

```
# aws_iam_access_key.users_access_keys["Creed"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   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                  = "Creed"
+ }

# aws_iam_access_key.users_access_keys["Darryl"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   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                  = "Darryl"
+ }

# aws_iam_access_key.users_access_keys["David"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   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                  = "David"
+ }

# aws_iam_access_key.users_access_keys["Dwight"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   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                  = "Dwight"
+ }

# aws_iam_access_key.users_access_keys["Erin"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   create_date           = (known after apply)
+   encrypted_secret      = (known after apply)
```

```

}

# aws_iam_access_key.users_access_keys["Kelly"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   + 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                  = "Kelly"
+ }

# aws_iam_access_key.users_access_keys["Kevin"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   + 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                  = "Kevin"
+ }

# aws_iam_access_key.users_access_keys["Meredith"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   + 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                  = "Meredith"
+ }

# aws_iam_access_key.users_access_keys["Michael"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
+   + 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)
+ }

```

```

# aws_iam_access_key.users_access_keys["Oscar"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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             = "Oscar"
}

# aws_iam_access_key.users_access_keys["Pam"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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             = "Pam"
}

# aws_iam_access_key.users_access_keys["Peter"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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             = "Peter"
}

# aws_iam_access_key.users_access_keys["Phyllis"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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             = "Phyllis"
}

# aws_iam_access_key.users_access_keys["Robert"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + create_date      = (known after apply)
  + encrypted_secret = (known after apply)

```

```

}

# aws_iam_access_key.users_access_keys["Stanley"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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              = "Stanley"
}

# aws_iam_access_key.users_access_keys["Toby"] will be created
+ resource "aws_iam_access_key" "users_access_keys" {
  + 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              = "Toby"
}

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

# aws_iam_group_policy_attachment.ec2_full_access will be created
+ resource "aws_iam_group_policy_attachment" "ec2_full_access" {
  + group      = "developers"
  + id         = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

# aws_iam_group_policy_attachment.s3_read_write will be created
+ resource "aws_iam_group_policy_attachment" "s3_read_write" {
  + group      = "developers"
  + id         = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/AmazonS3FullAccess"
}

# aws_iam_user.users["Andy"] will be created
+ resource "aws_iam_user" "users" {
  + arn = (known after apply)
}

"Stanley" = {
  "access_key_id" = "AKIARC5V6TLZZQSBA3L6"
  "user_arn"      = "arn:aws:iam::075006647027:user/users/Stanley"
  "user_unique_id" = "AIDARC5V6TLZRHMTXOPWL"
}

"Toby" = {
  "access_key_id" = "AKIARC5V6TLZZBDM4RFF"
  "user_arn"      = "arn:aws:iam::075006647027:user/users/Toby"
  "user_unique_id" = "AIDARC5V6TLZR3HEUYBL7"
}

}

group_details = {
  "group_arn" = "arn:aws:iam::075006647027:group/developers"
  "group_name" = "developers"
}

@23-22411-061-rgb → ~/Lab13 $ |

```

## task7\_terraform\_output

```
}
023-22411-061-rgb → ~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIARC5V6TLZRTFMY7PL"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Andy"
    "user_unique_id" = "AIDARC5V6TLZU6VFTMDEB"
  }
  "Angela" = {
    "access_key_id" = "AKIARC5V6TLZ463GTLHV"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Angela"
    "user_unique_id" = "AIDARC5V6TLZ6XWIK5R65"
  }
  "Charles" = {
    "access_key_id" = "AKIARC5V6TLZ3B4YCJUM"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Charles"
    "user_unique_id" = "AIDARC5V6TLZYKE3G2A2E"
  }
  "Clark" = {
    "access_key_id" = "AKIARC5V6TLZVGCJR65Q"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Clark"
    "user_unique_id" = "AIDARC5V6TLZ3XET7EUN5"
  }
  "Creed" = {
    "access_key_id" = "AKIARC5V6TLZUZSYF670"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Creed"
    "user_unique_id" = "AIDARC5V6TLZUYN2JDIQP"
  }
  "Darryl" = {
    "access_key_id" = "AKIARC5V6TLZ4DPACDG4"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Darryl"
    "user_unique_id" = "AIDARC5V6TLZTZIYA0VXX"
  }
  "David" = {
    "access_key_id" = "AKIARC5V6TLZSQ4M5B2D"
    "user_arn" = "arn:aws:iam::075006647027:user/users/David"
    "user_unique_id" = "AIDARC5V6TLZ2IHHXH2ZG"
  }
  "Dwight" = {
    "access_key_id" = "AKIARC5V6TLZ5BSYEH7X"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Dwight"
    "user_unique_id" = "AIDARC5V6TLZ3RYXQYU2S"
  }
  "Erin" = {
    "access_key_id" = "AKIARC5V6TLZRTDGXSWZ"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Erin"
    "user_unique_id" = "AIDARC5V6TLZ24ZUDJNMN"
  }
  "Gabe" = {
    "access_key_id" = "AKIARC5V6TLZ37JBNYF2"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Gabe"
    "user_unique_id" = "AIDARC5V6TLZ3HM2HUT27"
  }
  "Holly" = {
    "access_key_id" = "AKIARC5V6TLZ2CFVUA2B"
    "user_arn" = "arn:aws:iam::075006647027:user/users/Holly"
    "user_unique_id" = "AIDARC5V6TLZS45HCAE2I"
  }
}
```

## task7\_tfstate\_secrets

```
023-22411-061-rgb → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
023-22411-061-rgb → ~/Lab13 $ vim main.tf
023-22411-061-rgb → ~/Lab13 $ vim main.tf
023-22411-061-rgb → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
023-22411-061-rgb → ~/Lab13 $ terraform output all_access_key_secrets
{
  "Andy" = "/92JzkbKfj0dk18ivoYNzWx0L4Fa5fouByzGnyh5"
  "Angela" = "/UeZp1B1Yg9suVUBKGRwdaTcbUBc8jmXvbaLvUPi"
  "Charles" = "niDRI43swahYD2E2v5NLmse0Y6tJiIp/meyIxJ8b"
  "Clark" = "NL4f76aifrSm1o5bZnRi61mV1V/nvGZsv09dITgG"
  "Creed" = "d+wZdjfBKGXvY98Igd08yrxCMDw7VnaNDJ3ThmMH"
  "Darryl" = "U/HCXnpqknkoQY1oUfcs56bju4F/BCqR2Kt8zm1u"
  "David" = "8P0kErm0/dQkd9I3vbcSWByxrqDLtbioRrSE/CPC"
  "Dwight" = "oj6okUe1zoDNC3Rp7Cwy4lTXl0Z0bskdTcsLlG9S"
  "Erin" = "7827+LmyXzP/jcEgv0tIEsIeBXyIP6x0xxXlwhq1"
  "Gabe" = "RWH+DEgtEzqnK7mNf1lUqhbCcG2RZXBDXC6QK5uA"
  "Holly" = "lgXM7weu1lh00IyFksVq8pDPvmuGtt+k0x3pF5TR"
  "Jan" = "x7UBrpRjiJ93EntYdhJRo8LEOHU8SE80/AKxyeMZ"
  "Jim" = "MgLK19x7aQti8e3WwMpUby+zjHiUM/DB4wLz/qJw"
  "Jo" = "VC4WJUlSv6S8/Njij17y7EseRRuyS40xXwyNhY4v9"
  "Kelly" = "wb1xcvSXTL5R6qEpUqG5TM6Z6YBMegaKSxinoELh"
  "Kevin" = "mFPIO5F2epfyx4Rn00WXP7Zd7adtFyhnrQ3jQX8"
  "Meredith" = "MMMRAi/ZnM/mxyRk7wuBtF2Vfjo10IGin+d20TK"
  "Michael" = "AmjXOvpywEXppFTBwkyo2beXKCFo9wjbdhbyfYj"
  "Oscar" = "fHtpHZht1MntnJlQdntUs+gJwd2sot28QeMH/5w"
  "Pam" = "yZyxeI9LPdEUP2ACE4n9pSZKpa0Uxvf4RF2PBgDi"
  "Peter" = "JQ7MD03XGguYM4KUYNiCLWSJltoj5QeNutz368bu2"
  "Phyllis" = "/6aQZRCsPqu5LnCym7Bckp4FD+nvUruwTBk0B4wf"
  "Robert" = "2wNjFtodF1Kj1RF9TZ3dRMua1h6GbCE2k1z16in"
  "Ryan" = "8xs1mJ6gVuCZev+ff7ONNvE7tLST5Vpow6h02xbP"
  "Stanley" = "wVAID79zpoAyerJfKrkI+KVAVa++y0GsbvS7D+nv"
  "Toby" = "TnpNCcyNpWtr+bHf5CM+t+Bp/WBOWVft4PCjJc0P"
}
```

## task7\_aws\_console\_all\_users

The screenshot displays the AWS IAM console interface. The top navigation bar shows the AWS logo, a search bar, and the account ID: 0750-0664-7027. The left sidebar contains the 'Identity and Access Management (IAM)' section, with options for Dashboard, 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, Analyzer settings), and a 'New' button. The main content area is titled 'Users (27)' and includes a search bar and a 'Create user' button. Below this, a table lists the users. The first screenshot shows users from Andy to Gabe, and the second screenshot shows users from Holly to Pam. The table columns are: User name, Path, Groups, Last activity, MFA, Password age, Console last sign-in, and a checkbox.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
Andy	/users/	1	-	-	36 minutes	-
Angela	/users/	1	-	-	35 minutes	-
Charles	/users/	1	-	-	35 minutes	-
Clark	/users/	1	-	-	35 minutes	-
Creed	/users/	1	-	-	36 minutes	-
Darryl	/users/	1	-	-	36 minutes	-
David	/users/	1	-	-	36 minutes	-
Dwight	/users/	1	-	-	35 minutes	-
Erin	/users/	1	-	-	35 minutes	-
Gabe	/users/	1	-	-	35 minutes	-
Holly	/users/	1	-	-	36 minutes	-
Jan	/users/	1	-	-	35 minutes	-
Jim	/users/	1	-	-	36 minutes	-
Jo	/users/	1	-	-	35 minutes	-
Kelly	/users/	1	-	-	35 minutes	-
Kevin	/users/	1	-	-	35 minutes	-
Meredith	/users/	1	-	-	35 minutes	-
Michael	/users/	1	-	-	35 minutes	-
Oscar	/users/	1	-	-	36 minutes	-
Pam	/users/	1	-	-	35 minutes	-

## task7\_aws\_console\_group\_members

The screenshot displays the AWS IAM console interface for managing the 'developers' user group. The left-hand navigation pane includes sections for Identity and Access Management (IAM), Access Management, and Access reports. The main content area shows the 'Summary' tab for the 'developers' group, indicating it was created on January 07, 2026, at 12:12 UTC+05:00. Below the summary, the 'Users in this group' section lists 26 users in a table format.

User name	Last activity	Creation time
Clark	None	37 minutes ago
Creed	None	37 minutes ago
Darryl	None	37 minutes ago
David	None	37 minutes ago
Dwight	None	37 minutes ago
Erin	None	37 minutes ago
Gabe	None	37 minutes ago
Holly	None	37 minutes ago
Jan	None	37 minutes ago
Jim	None	37 minutes ago
Jo	None	37 minutes ago
Kelly	None	37 minutes ago
Kevin	None	37 minutes ago
Meredith	None	37 minutes ago
Michael	None	37 minutes ago
Pam	None	37 minutes ago
Oscar	None	37 minutes ago
Sean	None	37 minutes ago

task7\_aws\_console\_user\_access\_key

IAM > Users > Michael

Identity and Access Management (IAM)

Q Search IAM

Dashboard

Access Management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

Temporary delegation requests

New

Access reports

Access Analyzer

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](#)

AKIARC5V6TLZRKSOLYWT

Description

-

Last used

None

Last used region

N/A

Status

Active

Created

39 minutes ago

Last used service

N/A

Actions

Create access key

API keys for Amazon Bedrock (0)

Actions

Generate API Key

task7\_s3\_tfstate\_multiple\_users

```
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 4,
  "lineage": "ec9691c6-13e2-9979-d0f3-2e5ace0be39a",
  "outputs": {
    "all_access_key_secrets": {
      "value": {
        "Andy": "/92JzkbKfjOdk18ivoYNzWx0L4Fa5fOuByzGNYh5",
        "Angela": "/UeZplB1Yg9suVUBKGRwdaTcbUBcBjmXVbaLvUPi",
        "Charles": "niDRI43swahYD2E2v5NLMseOY6tJiIp/meyIxJ8b",
        "Clark": "NL4f76aifrSm1o5bZnRi61mVlV/nvGZsv09dITgG",
        "Creed": "d+wZdjfBKGXvY9BIgdo8yrxCMDw7VnaNDJ3ThmMH",
        "Darryl": "U/HCXnpqknKoQYloUFGS56bj4F/BCqR2Kt8zm1u",
        "David": "8P0kErm0/dQKd9I3vbCSWByxrqDLtbioRrsE/CPC",
        "Dwight": "oj6okUelzoDNC3Rp7Cwy4lTXl0Z0bsKdTcsLlG9S",
        "Erin": "7827+LmyXzP/jcEgv0tIEsIeBXylP6xOxxXlwhq1",
        "Gabe": "RWH+DEgtEzqnK7mNfI1uQhbCccG2RZXBDXC6QK5uA",
        "Holly": "lgXM7WEuilh00IyFksVq8pDPvmuGtt+k0x3pF5TR",
        "Jan": "X7UBrPrjiJ93EmtYdhJRo8LEOHU8SE80/AKxyeMZ",
        "Jim": "MgLKl9x7aQti8e3WWMpUbY+zjHiUM/DB4wLz/qJw",
        "Jo": "VC4WJulSv6S8/Njjl7y7ESerruys40xXWynHy4v9",
        "Kelly": "wblxcVSXtL5R6qEpUQg5TM6Z6YBMegaKSxin0ELh",
        "Kevin": "mFPIO5F2epfyx4Rn00WXP7Zd7adtFyhnRQ3jQX8",
        "Meredith": "MMRAi/ZnM/mxyRk7wuBTtF2Vfjo10IGiN+d20TK",
        "Michael": "AmjXOvpywEXppFTBwkyy02beXKCF09wjbdhbyfYj",
        "Oscar": "fHtpHZhct1MmtNj1QdntUs+gjwd2sot28QeMH/5W",
        "Pam": "yZyxeI9LPdEUP2AcE4n9pSZKpaOUxvf4RF2PBgDi",
        "Peter": "JQ7MD03XGguYM4KUYNiCLWSJltOj5QeNut36Bbu2",
        "Phyllis": "/6aQZRCsPqu5LnCym7BcKp4FD+nvUruwTBkOB4Wf",
        "Robert": "2wNjFtodFlKj1RF9TZZ3dRMuaih6GbCE2k1z16in",
        "Ryan": "8xs1mJ6gVuCZev+ff70NnvE7tLST5Vpow6h02xbP",
        "Stanley": "WVAID79zpoAyerJfKRKI+KVAVa++y0GsbvS7D+nv",
        "Toby": "TnpNCcyNpWtr+bHf5CM+t+Bp/WBOWVft4PCjJc0P"
      },
      "type": [
        "object",
        {
          "Andy": "string",
          "Angela": "string",
          "Charles": "string".
        }
      ]
    }
  }
}
```

Cleanup

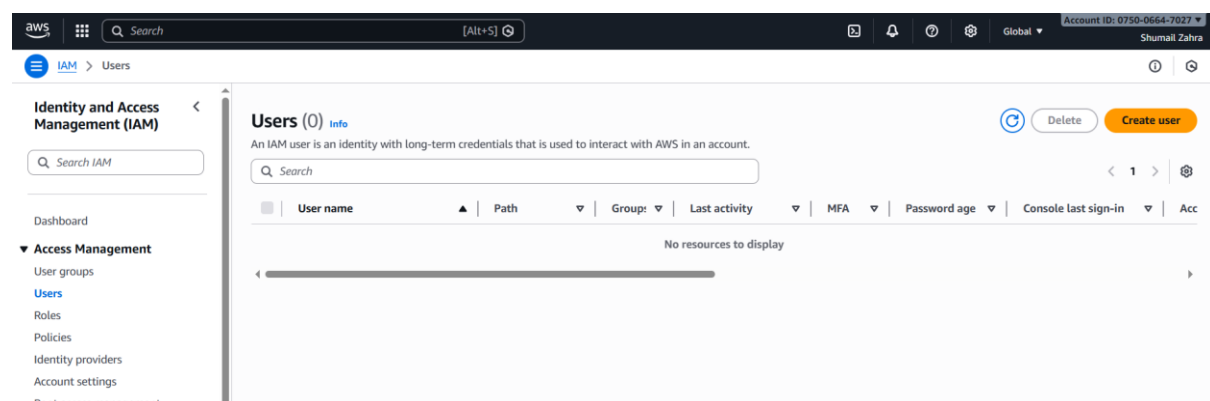
cleanup\_destroy\_complete

```
aws_iam_user.users["Kelly"]: Destroying... [id=Kelly]
aws_iam_user.users["Meredith"]: Destruction complete after 3s
aws_iam_user.users["Jan"]: Destroying... [id=Jan]
aws_iam_user.users["Erin"]: Destruction complete after 4s
aws_iam_user.users["Darryl"]: Destroying... [id=Darryl]
aws_iam_user.users["Dwight"]: Destruction complete after 4s
aws_iam_user.users["Angela"]: Destroying... [id=Angela]
aws_iam_user.users["David"]: Destruction complete after 5s
aws_iam_user.users["Pam"]: Destroying... [id=Pam]
aws_iam_user.users["Peter"]: Destruction complete after 3s
aws_iam_user.users["Oscar"]: Destroying... [id=Oscar]
aws_iam_user.users["Charles"]: Destruction complete after 3s
aws_iam_user.users["Kevin"]: Destroying... [id=Kevin]
aws_iam_user.users["Michael"]: Destruction complete after 2s
aws_iam_user.users["Ryan"]: Destroying... [id=Ryan]
aws_iam_user.users["Andy"]: Destruction complete after 2s
aws_iam_user.users["Creed"]: Destroying... [id=Creed]
aws_iam_user.users["Jan"]: Destruction complete after 2s
aws_iam_user.users["Phyllis"]: Destroying... [id=Phyllis]
aws_iam_user.users["Darryl"]: Destruction complete after 2s
aws_iam_user.users["Toby"]: Destroying... [id=Toby]
aws_iam_user.users["Kelly"]: Destruction complete after 4s
aws_iam_user.users["Angela"]: Destruction complete after 3s
aws_iam_user.users["Pam"]: Destruction complete after 2s
aws_iam_user.users["Creed"]: Destruction complete after 2s
aws_iam_user.users["Kevin"]: Destruction complete after 2s
aws_iam_user.users["Toby"]: Destruction complete after 3s
aws_iam_user.users["Ryan"]: Destruction complete after 4s
aws_iam_user.users["Phyllis"]: Destruction complete after 5s
aws_iam_user.users["Clark"]: Destruction complete after 9s
aws_iam_user.users["Oscar"]: Destruction complete after 7s
Releasing state lock. This may take a few moments...
```

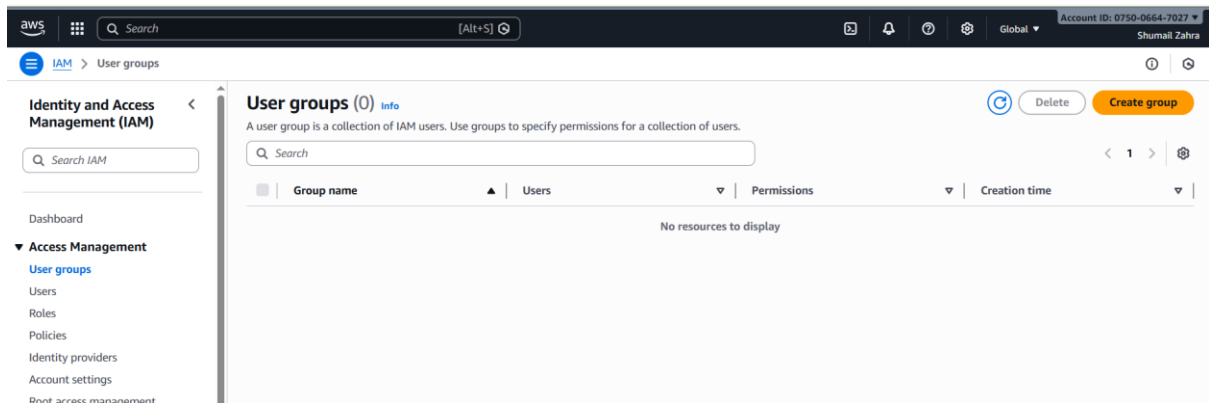
**Destroy complete! Resources: 107 destroyed.**

@23-22411-061-rgb → ~/Lab13 \$ |

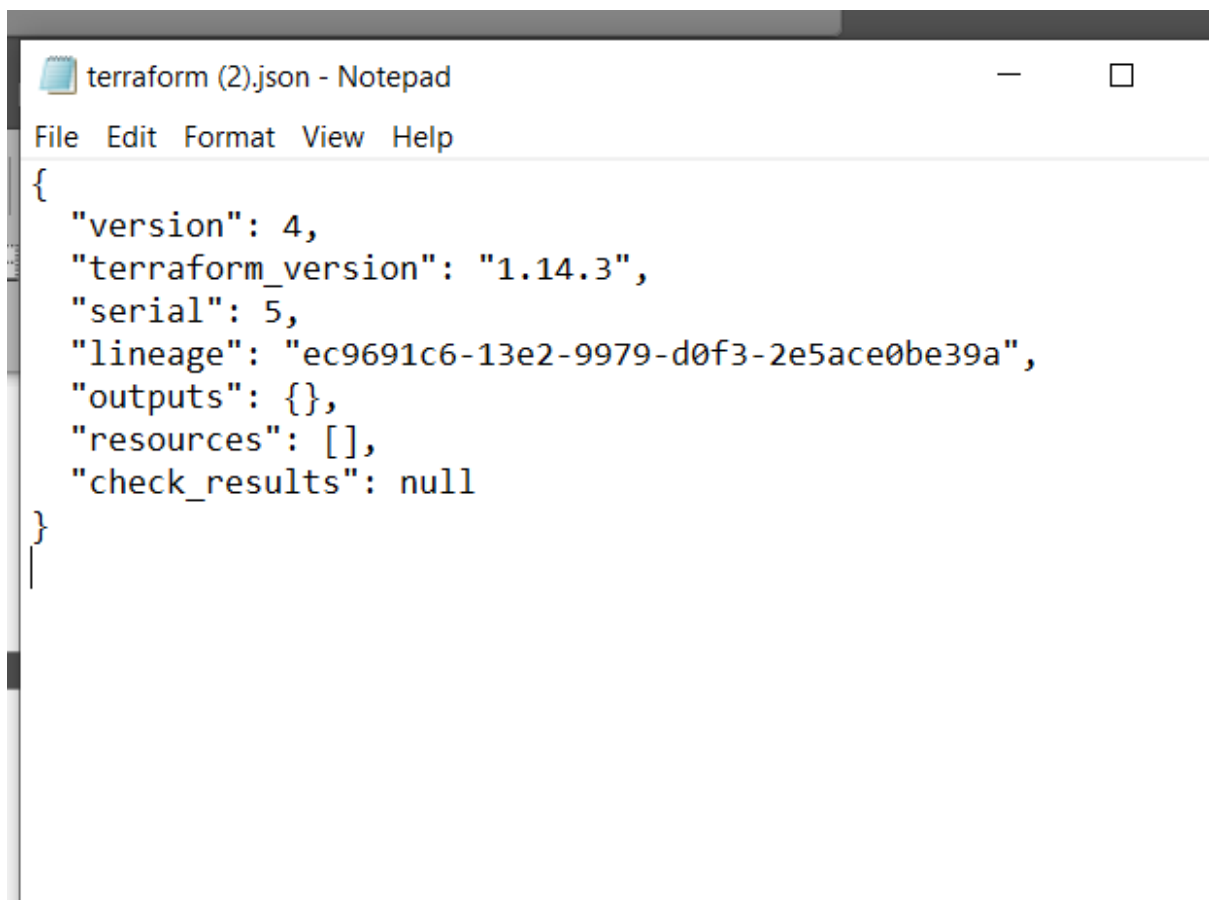
cleanup\_aws\_console\_users\_deleted



cleanup\_aws\_console\_group\_deleted



cleanup\_s3\_empty\_state



cleanup\_final\_files

```
@23-22411-061-rgb → ~/Lab13 $ ls -la
total 61764
drwxrwxr-x 4 codespace codespace 4096 Jan 7 07:23 .
drwxr-x-- 1 codespace codespace 4096 Jan 7 07:23 ..
drwxr-xr-x 3 codespace codespace 4096 Jan 7 07:06 .terraform
-rw-r--r-- 1 codespace codespace 2422 Jan 6 19:50 .terraform.lock.hcl
drwxr-xr-x 3 codespace codespace 4096 Jan 5 19:11 aws
-rw-rw-r-- 1 codespace codespace 63189840 Jan 6 16:26 awscli2.zip
-rwxrwxr-x 1 codespace codespace 456 Jan 6 19:30 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace 50 Jan 7 06:38 locals.tf
-rw-rw-r-- 1 codespace codespace 2869 Jan 7 07:23 main.tf
-rw-rw-r-- 1 codespace codespace 0 Jan 7 06:26 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 7 06:26 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace 167 Jan 7 06:39 users.csv
-rw-rw-r-- 1 codespace codespace 150 Jan 6 19:25 variables.tf
@23-22411-061-rgb → ~/Lab13 $ |
```

cleanup\_s3\_bucket\_deleted

[Alt+S]
United States (N. Virginia)
Account ID: 0750-0664-7027
Shumail Zahra

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

**Empty bucket: status**
Cancel
Exit

The details below are no longer available after you navigate away from this page.

**Summary**

**Source**  
[s3://myapp-s3-buckettt](#)

**Successfully deleted**  
 18 objects, 152.0 KB

**Failed to delete**  
 0 objects

**Failed to delete (0)**

Name	Prefix	Version ID	Type	Last modified	Size	Error
No failed object deletions						