
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 gith
ub.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 space... 23-22411-0... main* Shutdown about 3 mo...
supreme-barnac... supreme barnac... 23-22411-0... main* Shutdown about 12 d...
glorious-trib... glorious tribble 23-22411-0... main Shutdown about 1 da...
fantastic-spa... fantastic spa... 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-b
se-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=ma
in)

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=ma
in)

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)
$ |

```

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
 * 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.
```

```
@23-22411-061-rgb → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ |
```

Task 1 — Create IAM Group and Output Details

task1_project_directory

```
applicable: raw  
@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 → /workspaces/CC_ShumailZahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13  
@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
[~][@23-22411-061-rgb ~]# sudo apt-get update
Get:1 https://dl.yarnpkg.com/debian stable InRelease [3961 B]
Get:3 https://dl.yarnpkg.com/debian stable/main all Packages [1.8 kB]
Get:5 https://packages.microsoft.com/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 all1 Packages [643 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble InRelease [256 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://archive.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 [3059 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [35.9 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Fetched 10.5 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-zubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
[~][@23-22411-061-rgb ~]# curl | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
[~][@23-22411-061-rgb ~]# echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
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[[@23-22411-061-rgb ~]# /workspaces/CC_Shumailzahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
[~][@23-22411-061-rgb ~]# 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[[@23-22411-061-rgb ~]# /workspaces/CC_Shumailzahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
[~][@23-22411-061-rgb ~]# terraform version
[[[[AA[[@23-22411-061-rgb ~]# /workspaces/CC_Shumailzahra_2023-BSE-061_Lab13 (main) $ cd ~/Lab13
[~][@23-22411-061-rgb ~]# 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

The screenshot shows the AWS IAM User Groups page. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected. Under 'Access Management', 'User groups' is also selected. The main area has a heading 'User groups (1) Info' with a note: 'A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.' Below this is a search bar and a table.

Group name	Users	Permissions	Creation time
developers	0	Not defined	1 hour ago

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

```

823-22411-061-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 'Users' page. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access Management' (with 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Root access management', and 'Temporary delegation requests'), and 'Access reports'. The main content area displays a table titled 'Users (2)'. The table has columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', 'Console last sign-in', and 'Actions'. Two users are listed: 'loadbalancer' (Path: /users/, Group: 1, Last activity: -), and 'terraform-user' (Path: /, Group: 0, Last activity: 2 hours ago). A 'Create user' button is located at the top right of the table.

task2_aws_console_user_groups

The screenshot shows the AWS IAM 'loadbalancer' user details page. The left sidebar navigation is identical to the previous screenshot. The main content area shows the 'Summary' tab for the 'loadbalancer' user. It displays the ARN (arn:aws:iam::075006647027:user/users/loadbalancer), Console access status (Disabled), and the creation date (January 07, 2026, 00:01 (UTC+05:00)). It also shows an 'Access key 1' section with a 'Create access key' button. Below the summary, there are tabs for 'Permissions', 'Groups (1)', 'Tags (1)', 'Security credentials', and 'Last Accessed'. The 'Groups (1)' tab is selected, showing the user is a member of the 'developers' group. There is a 'Remove' button next to the group entry and an 'Add user to groups' button.

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"
}

:q!
```

task3_terraform_apply

```
123-22411-061-rgb ~ /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" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYHBQ2J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

task3_aws_console_policies

The screenshot shows the AWS IAM console with the 'developers' user group selected. The 'Permissions' tab is active, displaying two policies attached to the group:

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-4a / lab: vim variables.tf
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "IdontKnow"
```

task4_create_login_script

```
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

```
823-22411-061-rgb ~ ~/Lab13 $ vim main.tf
823-22411-061-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

823-22411-061-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.
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]
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
+ 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 (local-exec): (output suppressed due to sensitive value in config)
null_resource.create_login_profile (local-exec): Creation complete after 7s [id=5174224382095500929]

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

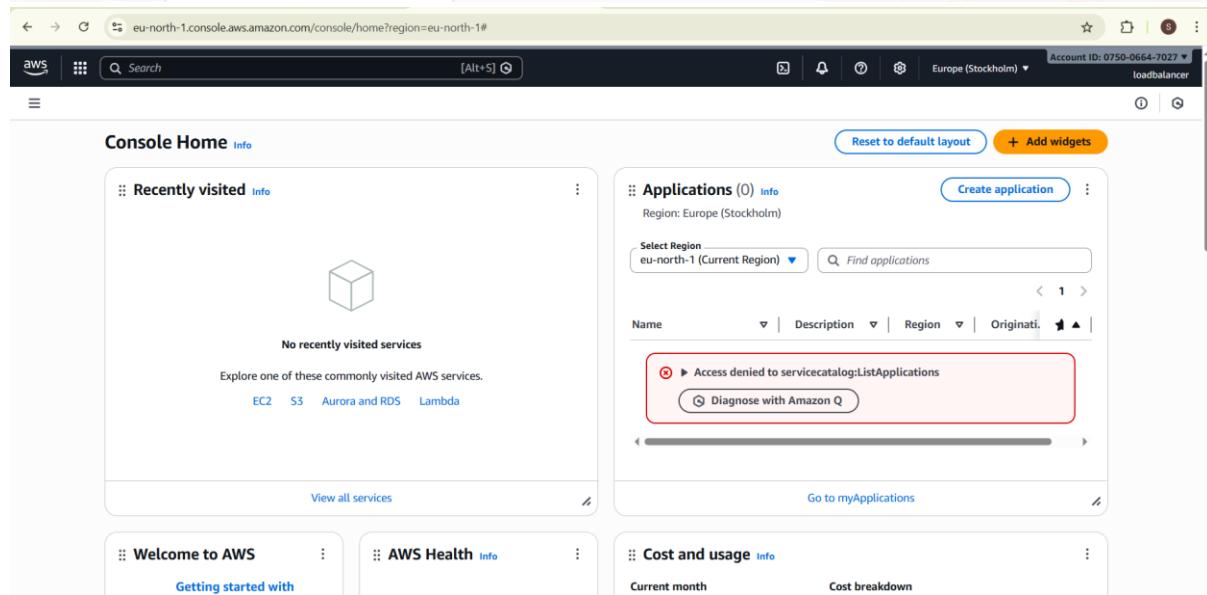
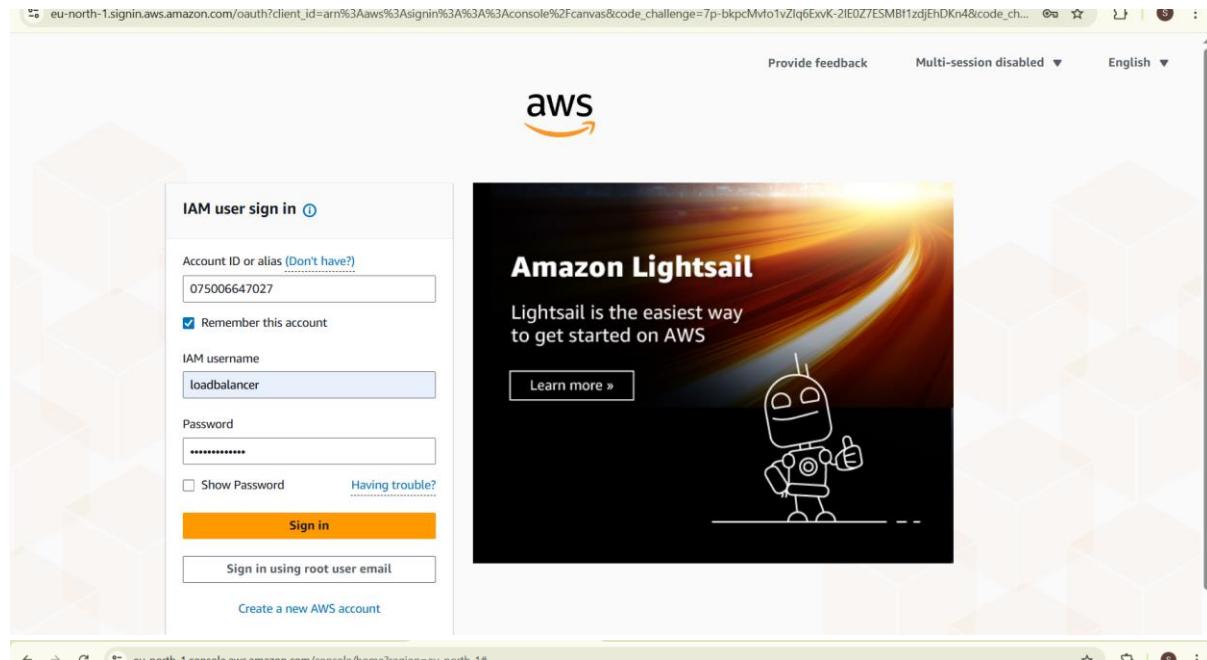
Outputs:

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



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

Confirm New Password

Show 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

```
@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]
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" = "AIDARC5V6TLZSIAYHB02J"
  "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/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 $ |
```

task5_tfstate_secret

```
@23-22411-061-rgb ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
"access_key_secret": {
  "value": "vwM1r/7Rh4bAu8TLJar/WeCiL4Elqk81I5qt//aT",
  "type": "string",
  "sensitive": true
},
"group_details": {
  "value": {
    "group_arn": "arn:aws:iam::075006647027:group/groups/developers",
    "group_name": "developers",
    "unique_id": "AGPARC5V6TLZ2RJGN6RRO"
  }
},
@23-22411-061-rgb ~/Lab13 $ |
```

task5_aws_console_access_keys

Access keys (1)

Create access key

AKIARC5V6TLZ4UJHALZF

Description	Status
-	Active
Last used	Created
None	5 minutes ago
Last used region	Last used service
N/A	N/A

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-buckettt

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.

Object Ownership

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

The screenshot shows the 'Bucket Versioning' section of the AWS S3 Bucket Properties page. It includes two checkboxes: 'Block public access to buckets and objects granted through new public bucket or access point policies' (unchecked) and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies' (unchecked). Below this is a 'Bucket Versioning' toggle switch set to 'Enable'. A note about tags is present, followed by a note about using the AWS API for managing tags.

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](#)

Note: 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" = "AGPARC5V6TLZ2RJGN6RRO"
}
user_details = {
  "unique_id" = "AIDARC5V6TLZSIAYHB02J"
  "user_arn" = "arn:aws:iam::075006647027:user/users/loadbalancer"
  "user_name" = "Loadbalancer"
}
@23-22411-061-rgb ~/Lab13 $ |

```

task6_s3_tfstate_file

Amazon S3 > Buckets > myapp-s3-buckettt > myapp/

myapp/

Objects | Properties

Actions ▾ | **Create folder** | **Upload**

Objects (1)

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

Find objects by prefix

Show versions

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	January 7, 2026, 11:26:07 (UTC+05:00)	6.7 KB	Standard

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

```
- 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 = "AIDARC5V6TLZSIAYHBO2J"
  - 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.lb_access_key: Destroying... [id=AKIARC5V6TLZ4UJHALZF]
aws_iam.access_key.lb_access_key: Destruction complete after 0s
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260106190733105100000002]
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-20260106190155394100000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260106190733100700000001]
aws_iam_group_policy_attachment.change_password: Destruction complete after 0s
aws_iam_user_group_membership.lb_membership: Destruction complete after 0s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 0s
aws_iam_group.developers: Destroying... [id=developers]
aws_iam.access_key.lb_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

```
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

```
#23-22411-061-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" = "AKIARC5V6TLZZQSBABL6"
    "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

```
@23-22411-061-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" = "AKIARC5V6TLZVGCIJR65Q"
    "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" = "AIDARC5V6TLTZIYAOVXX"
  }
  "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" = "AIDARC5V6TLZ24ZUDJMNM"
  }
  "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

```
#23-22411-061-rgb → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
#23-22411-061-rgb → ~/Lab13 $ vim main.tf
#23-22411-061-rgb → ~/Lab13 $ vim main.tf
#23-22411-061-rgb → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
#23-22411-061-rgb → ~/Lab13 $ terraform output all_access_key_secrets
{
  "Andy" = "/92JzkbKFj0dk18ivoYnzwX0L4Fa5f0uByzGnyh5"
  "Angela" = "/Uezplb1Yg9suUbkGRwdaTcbCbjmXvbaLVUpi"
  "Charles" = "niDRI43swahYD2E2v5NLmse0Y6tjiIp/meyIxJ8b"
  "Clark" = "nL4f76a1frSm05bzNRI61mVl/nvgzsv09d1TgG"
  "Creed" = "dw-wZdjfbBKGXvY9B1gdo8rxCMdw7VnaND13ThmMH"
  "Darryl" = "U/HCxnpqknKoqloufc56bjju4F/BcqR2Kt8zmLu"
  "David" = "8P0KErmo/dQkD913vbCSWByxrqDLtbioRrSe/CPC"
  "Dwight" = "oj6okUelz0DNC3Rp7Cwy4ITX10Z0bsKd7csL1g9S"
  "Erin" = "7827+LmyXzP/jcEgv0tIEsIeBxy1P6x0xxLwhq1"
  "Gabe" = "RWH+DEgTEzqnK7mfIluQhbCcG2RZXBDXC6QK5uA"
  "Holly" = "TgxM7WEuiTh00iyFksVq8pDPvnuGTE+k0x3p5TR"
  "Jan" = "X7UBrpRji393EmtYdhJRo8LEOHU8SE80/AKxyeMZ"
  "Jim" = "MgLk19x7aoit8e3WMp0ubY+zjHiUM/DB4wlz/qJw"
  "Jo" = "VC4WJulsV658/Nj17y7ESeRruyS40xWxlyNy4v9"
  "Kelly" = "wblxcvSXTL5R6eqpu0g5TM626YBMeGaKSxInoELh"
  "Kevin" = "mFPi05F2epfx4rn00WXPAT7d7adtfyhnrQ3jQX8"
  "Meredith" = "MMMRai/ZnM/mxyRk7wuBTf2Vfjo10IGiN+d20TK"
  "Michael" = "AmjX0vpvExppFTBwkyy02beXKCFo9wjbdhbyFyj"
  "Oscar" = "fhttpZhcht1Mmtn1QdntUs+qjwd2sot28qeMH/5W"
  "Pam" = "yZyxeI9LPdEUP2AcE4n9psZKpa0Uxf4RF2PBgDi"
  "Peter" = "JQ7MD03XGguM4KUYNiCLWS51t0j5qeNut36bbu2"
  "Phyllis" = "/6aQZRcsPqu5LnCym?BcKp4FD+nvUrwwTBkob4WF"
  "Robert" = "2wNjFtdofIKj1RF9Tz3dRMua1h6GbCE2k1z16in"
  "Ryan" = "8xs1mJ6gVuCzev+ff70nnvE7tLST5vpoW6h02xbp"
  "Stanley" = "WAID79zpoAyerJfkRKI+KvaV++y0Gsbv57D+nv"
  "Toby" = "TnpNCcyNpWtr+bHf5CM+t+Bp/WB0WVft4PcjJc0P"
}
```

task7_aws_console_all_users

Users (27) Info

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
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	-

Users (27) Info

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
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	-

Users (27) Info

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
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 shows the AWS IAM User Groups page. The top section displays the 'Summary' for the 'developers' user group, including the user group name 'developers', creation time 'January 07, 2026, 12:12 (UTC+05:00)', and ARN 'arn:aws:iam::075006647027:group/developers'. Below this, the 'Users (26)' tab is selected, showing a list of 26 users. The bottom section also displays the same list of 26 users, indicating a duplicate view or a bug in the interface.

User Name	Last Activity	Creation Time
Andy	None	37 minutes ago
Angela	None	37 minutes ago
Charles	None	37 minutes ago
Clark	None	37 minutes ago
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
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
Oscar	None	37 minutes ago
Pam	None	37 minutes ago

task7_aws_console_user_access_key

The screenshot shows the AWS IAM User Access Keys page for a user named Michael. The left sidebar includes links for Dashboard, Access Management (Users, Roles, Policies, Identity providers, Account settings, Root access management, Temporary delegation requests), and Access reports (Access Analyzer). The main content area displays one access key under 'Access keys (1)'. The key details are as follows:

Key ID	Description	Status	Created	Last used	Last used region	Last used service
AKIARCV6TLZRKSOLYWT	-	Active	39 minutes ago	None	N/A	N/A

Below the access key table, there is a section for 'API keys for Amazon Bedrock (0)' with a 'Generate API Key' button.

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": "/92JzkbKfj0dk18ivoYNzWx0L4Fa5f0uByzGNyh5",
       "Angela": "/UeZplB1Yg9suVUBKGRwdaTcbUBcBjmXVbaLvUPi",
       "Charles": "niDRI43swahYD2E2v5NLmse0Y6tJiIp/meyIxJ8b",
       "Clark": "NL4f76aifrSm1o5bZnRi61mVlV/nvGZsv09dITgG",
       "Creed": "d+wZdjfBKGVxY9BIgdo8yrxCMdW7VnaNDJ3ThmMH",
       "Darryl": "U/HCXnpqknKoQYloUfGS56bju4F/BCqR2Kt8zm1u",
       "David": "8P0kErm0/dQKd9I3vbCSWBxrqDLtbioRrsE/CPC",
       "Dwight": "oj6okUelzoDNC3Rp7Cwy4lTxl0Z0bsKdTcsllG9S",
       "Erin": "7827+LmyXzP/jcEgv0tIEsIeBXylP6x0xxXlwhq1",
       "Gabe": "RWH+DEgtEzqnK7mNfI1uQhbCcg2RZXBDXC6QK5uA",
       "Holly": "lgXM7WEuilh00IyFksVq8pDPvmuGTt+k0x3pF5TR",
       "Jan": "X7UBrpRjiJ93EmtYdhJRo8LEOHU8SE80/AKxyeMZ",
       "Jim": "MgLk19x7aQt18e3WWMpUbY+zjHiUM/DB4wLz/qJw",
       "Jo": "VC4WJu1Sv6S8/Njj17y7ESeRRuyS40xXWyNhY4v9",
       "Kelly": "wblxcVSXTL5R6qEpUQg5TM6Z6YBMegaKSxinOEh",
       "Kevin": "mFPIO5F2epfyx4Rn00WXPA7zd7adtfyhnRQ3jQX8",
       "Meredith": "MMMRai/ZnM/mxyRk7wuBTtF2Vfjo10IGiN+d20TK",
       "Michael": "AmjXOvpywEXppFTBwkyy02beXKCFo9wjbdhbyfYj",
       "Oscar": "fHtpHZhct1Mmtnj1QdntUs+gjwd2sot28QeMH/5W",
       "Pam": "yZyxeI9LPdEUP2AcE4n9pSZKpaOUxf4RF2PBgDi",
       "Peter": "JQ7MD03XGguYM4KUYNiCLWSJ1toj5QeNut36Bbu2",
       "Phyllis": "/6aQZRCsPqu5LnCym7BcKp4FD+nvUruwTBkOB4Wf",
       "Robert": "2wNjFtodFlkj1RF9TZZ3dRMuiah6GbCE2k1zl6in",
       "Ryan": "8xs1mJ6gVuCZev+ff7ONnvE7tLST5VpoW6h02xbP",
       "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 $ |
```

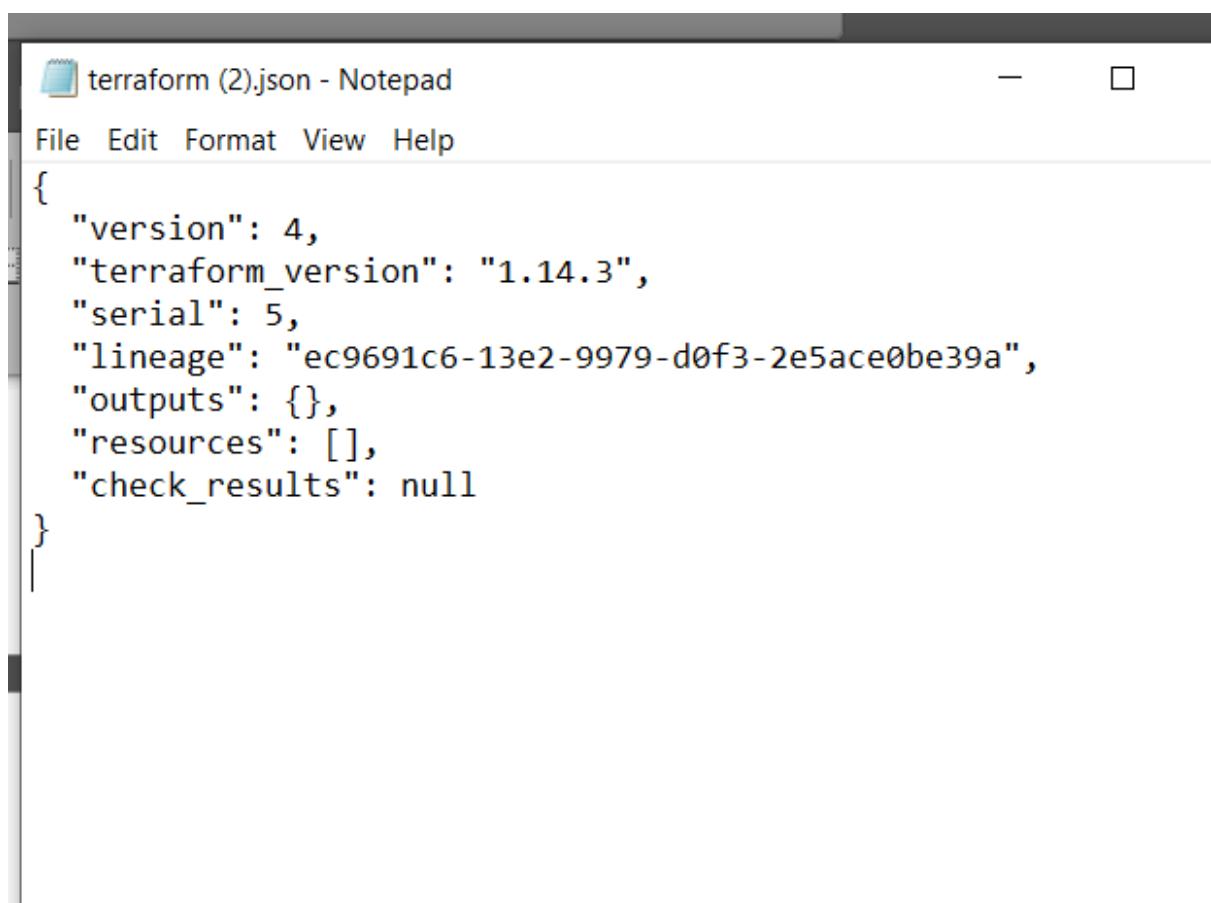
```
cleanup_aws_console_users_deleted
```

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is collapsed, and the main navigation bar shows 'Search' and 'Account ID: 0750-0664-7027'. The user 'Shumail Zahra' is logged in. The 'Access Management' section is expanded, with 'Users' selected. The main content area is titled 'Users (0) Info' and contains the message: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' Below this is a search bar and a table with the following columns: User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Acc. A note at the bottom of the table says 'No resources to display'.

cleanup_aws_console_group_deleted

The screenshot shows the AWS IAM User groups page. The left sidebar has 'Identity and Access Management (IAM)' selected under 'Access Management'. The main area title is 'User groups (0) Info'. It says 'A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.' A search bar and a 'Create group' button are at the top right. Below is a table header with columns 'Group name', 'Users', 'Permissions', and 'Creation time'. A message 'No resources to display' is shown.

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

The screenshot shows the AWS S3 console interface. At the top, there is a green success message: "Successfully emptied bucket 'myapp-s3-buckettt'. View details below. If you want to delete this bucket, use the delete bucket configuration." Below this, the title "Empty bucket: status" is displayed. A note says, "The details below are no longer available after you navigate away from this page." The "Summary" section shows the source as "s3://myapp-s3-buckettt" and lists "Successfully deleted" (18 objects, 152.0 KB) and "Failed to delete" (0 objects). The "Failed to delete (0)" section shows a table header with columns: Name, Prefix, Version ID, Type, Last modified, Size, and Error. The table body contains the message "No failed object deletions".