



CLOUD COMPUTING LAB **BSE (V-B)**

Submitted By:

Musfira Farooq

Roll No:

2023-BSE-045

Submitted To:

Sir Muhammad Shoaib

LAB 13

Terraform IAM Management with AWS

Task 0 — Lab Setup (Codespace & GH CLI)

Objective: Set up GitHub Codespace and connect.

- Create Codespace & list

```
PS C:\Users\Musfi\Lab13> gh codespace create --repo Musfira-0514/Lab13
/ Codespaces usage for this repository is paid for by Musfira-0514
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage
shiny-space-tribble-q75p747wxw7w36jg
PS C:\Users\Musfi\Lab13> gh codespace list
NAME              DISPLAY NAME    REPOSITORY      BRANCH  STATE    CREATED AT
shiny-space-tribble-q75p747wxw7w36jg shiny space tribble Musfira-0514/Lab13 main    Available about 2 minutes ago
PS C:\Users\Musfi\Lab13>
```

- SSH into Codespace

```
PS C:\Users\Musfi\Lab13> gh codespace ssh -c shiny-space-tribble-q75p747wxw7w36jg
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.

@Musfira-0514 → /workspaces/Lab13 (main) $ |
```

Task 1 — Create IAM Group and Output Details

Objective: Create IAM group developers and output details.

- Project directory creation

```
@Musfira-0514 → /workspaces/Lab13 (main) $ mkdir -p ~/Lab13
@Musfira-0514 → /workspaces/Lab13 (main) $ cd ~/Lab13
@Musfira-0514 → ~/Lab13 $ |
```

- Create main.tf file

```
@Musfira-0514 → ~/Lab13 $ touch main.tf
@Musfira-0514 → ~/Lab13 $ |
```

- main.tf content

```
GNU nano 7.2
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
  }
}
```

- Terraform init

```
@Musfira-0514 → ~/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!
```

- Terraform 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]

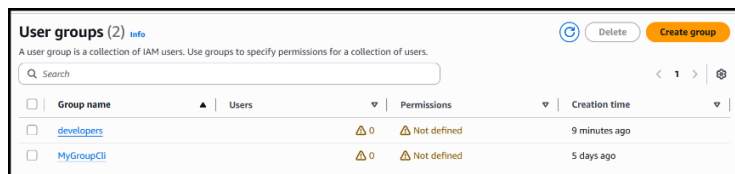
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSUVUK5OYLZ23IH44L"
}
@Musfira-0514 → ~/Lab13 $ |
```

- Terraform output

```
@Musfira-0514 → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSUVUK5OYLZ23IH44L"
}
@Musfira-0514 → ~/Lab13 $ |
```

- AWS Console verification



Group name	Users	Permissions	Creation time
developers	0	Not defined	9 minutes ago
MyGroupCLI	0	Not defined	5 days ago

Task 2 — Create IAM User with Group Membership

Objective: Create IAM user loadbalancer and add to developers group.

- Updated main.tf with user

```

GNU nano 7.2
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
  }
}

```

- Terraform apply

```

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

Changes to Outputs:
+ user_details = {
+   unique_id = (known after apply)
+   user_arn = (known after apply)
+   user_name = "loadbalancer"
}
aws_iam_user.lb: Creating...
aws_iam_user.lb: Creation complete after 1s [id=loadbalancer]
aws_iam_user_group_membership.lb_membership: Creating...
aws_iam_user_group_membership.lb_membership: Creation complete after 0s [id=terraform-20260105105748854700000001]

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

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSVUK5OYLZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSVUK5OYP5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |

```

- Terraform output

```

@Musfira-0514 → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSVUK5OYLZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSVUK5OYP5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |

```

- AWS Console user verification

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

Search	User name	Path	Group	Last activity	MFA	Password age	Console last sign
<input type="checkbox"/>	Admin	/	0	8 days ago	Passke...	8 days	8 days ago
<input type="checkbox"/>	lab-user	/	0	13 minutes ago	-	5 days	-
<input type="checkbox"/>	loadbalancer	/users/	1	-	-	-	-
<input type="checkbox"/>	terraform_user	/	0	5 days ago	-	-	-

- User group membership verification

loadbalancer Info

Summary

ARN
 arn:aws:iam::383704034224:user/users/loadbalancer

Console access
 Disabled

Created
 January 05, 2026, 15:57 (UTC+05:00)

Last console sign-in
 -

Permissions **Groups (1)** Tags (1) Security credentials Last Accessed

User groups membership

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

Group name	Attached policies
developers	-

Task 3 — Attach Policies to IAM Group

Objective: Attach AmazonEC2FullAccess & IAMUserChangePassword policies.

- Updated main.tf with policy attachments

```

GNU nano 7.2
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
  }
}

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"
}

```

- Terraform apply

```
@Musfir-6518 ➔ ~/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-20260105105748854700000001]

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"
}

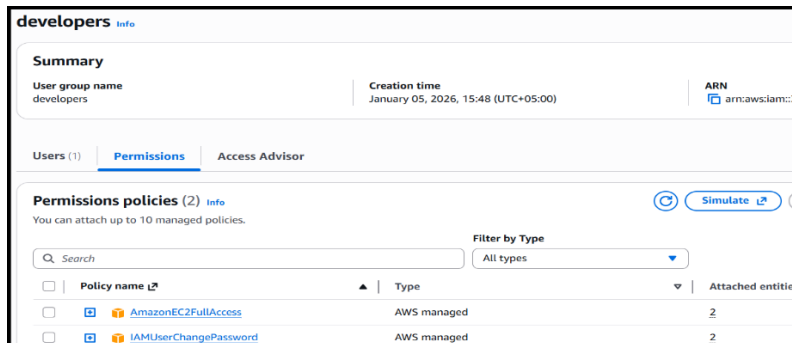
# aws_iam_group_policy_attachment.developer_ec2_fullaccess will be created
+ resource "aws_iam_group_policy_attachment" "developer_ec2_fullaccess" {
+   group      = "developers"
+   id         = (known after apply)
+   policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
}

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.change_password: Creation complete after 1s [id=developers-202601051111297250000000002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-202601051111297214000000001]

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

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSUUKSOYLVZ3IHUCL"
}
user_details = {
  "unique_id" = "AIDAVSVUHSOYPSASMRUFU"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfir-6518 ➔ ~/Lab13 $ |
```

- AWS Console verification



Task 4 — Create Login Profile for IAM User

Objective: Create login profile using bash script and null_resource.

- variables.tf

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

- Bash script content

```
GNU nano 7.2
#!/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
```

- main.tf null_resource

```
GNU nano 7.2
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 "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}'"
  }
}
```

- Terraform apply

```
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 3s [id=6060440309344465055]

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

Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSVUK5OYVZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSVUK5OYV5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |
```

- AWS CLI verify login profile

```
@Musfira-0514 → ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
  "LoginProfile": {
    "UserName": "loadbalancer",
    "CreateDate": "2026-01-05T11:22:17+00:00",
    "PasswordResetRequired": true
  }
}
@Musfira-0514 → ~/Lab13 $ |
```

- AWS Console login

IAM user sign in ⓘ

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

383704034224

☐ Remember this account

IAM username

loadbalancer

Password

.....

☐ Show Password [Having trouble?](#)

Sign in

Sign in using root user email

[Create a new AWS account](#)

- Password reset prompt

Password reset ⓘ

Your account (**383704034224**) 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 **Matches**

Confirm Password Change

Task 5 — Generate Access Keys for IAM User

Objective: Generate IAM access keys.

- main.tf access key resource


```

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

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 $
  }
}

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
}

```

- Terraform apply

```

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 0s [id=AKIAVSVUK5OYJQB4WKHC]

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

Outputs:
access_key_id = "AKIAVSVUK5OYJQB4WKHC"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSVUK5OYLZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSVUK5OYP5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |

```

- Terraform output

```

@Musfira-0514 → ~/Lab13 $ terraform output
access_key_id = "AKIAVSVUK5OYJQB4WKHC"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSVUK5OYLZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSVUK5OYP5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |

```

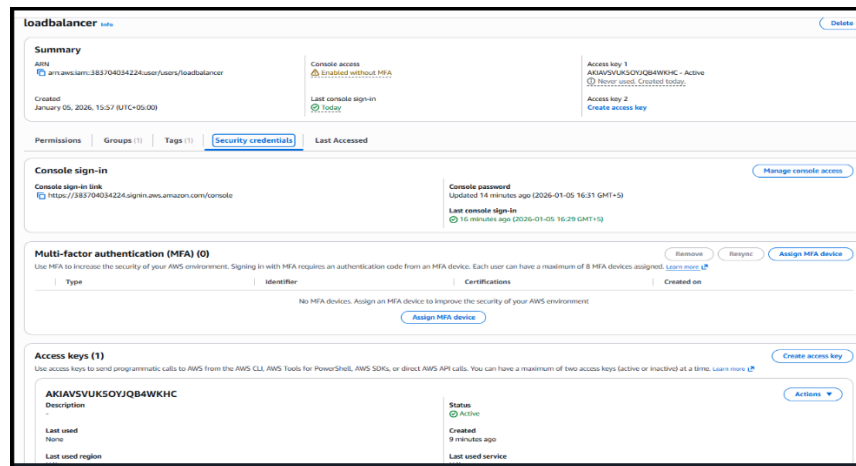
- tfstate secret

```

@Musfira-0514 → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
  "access_key_secret": {
    "value": "ntuxinRdtBvdNs/Nf+HM7ivAY2gC9uxYgw0MnZv/",
    "type": "string",
    "sensitive": true
  },
  "group_details": {
    "value": {
      "group_arn": "arn:aws:iam::383704034224:group/groups/developers",
      "group_name": "developers",
      "unique_id": "AGPAVSVUK5OYLZ23IH44L"
    },
  },
@Musfira-0514 → ~/Lab13 $ |

```

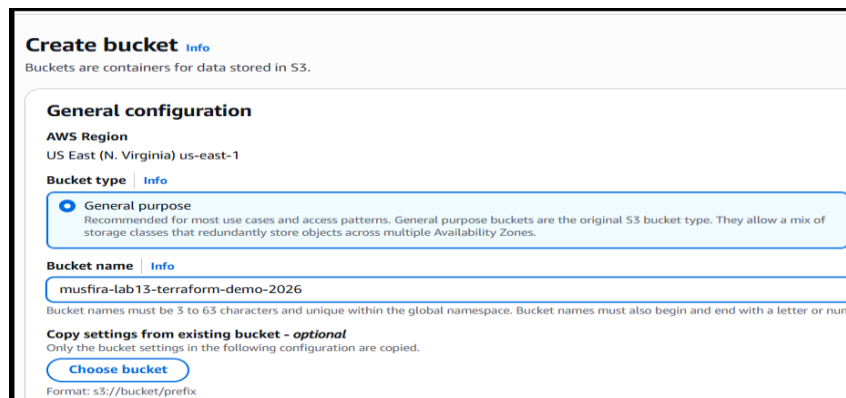
- AWS Console verification



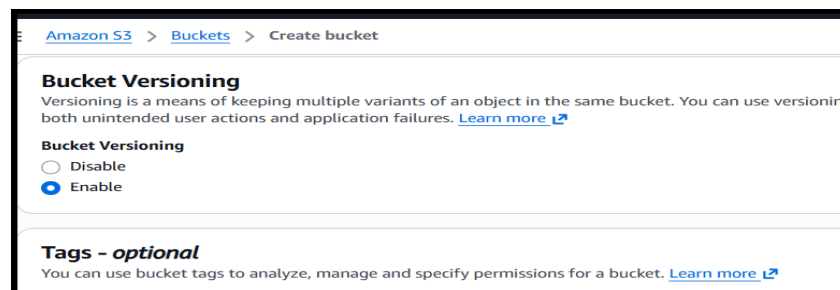
Task 6 — Implement Terraform Remote State with S3

Objective: Use S3 backend for remote state.

- S3 bucket create



- Enable versioning



- main.tf backend configuration

```

GNU nano 7.2
terraform {
  backend "s3" {
    bucket = "musfira-lab13-terraform-demo-2026"
    key    = "myapp/terraform.tfstate"
    region = "us-east-1"
    encrypt = true
    use_lockfile = true
  }
}

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

resource "aws_iam_group" "developers" {}

```

- Terraform init & migrate

```

@Musfira-0514 → ~/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/aws from the dependency lock file
- Reusing previous version of hashicorp/null from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- Using previously-installed hashicorp/null v3.2.4

Terraform has been successfully initialized!

```

- Terraform apply

```

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration
and found no changes to be made.

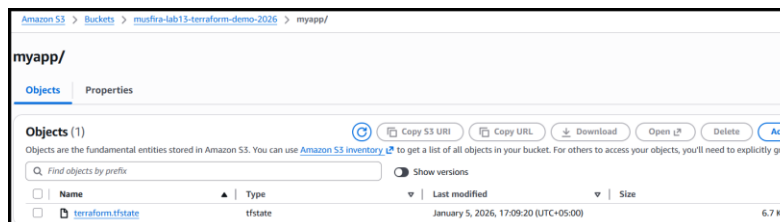
Releasing state lock. This may take a few moments...

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

Outputs:
access_key_id = "AKIAVSUVK5OYJQB4WKHC"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::383704034224:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVSUVK5OYLZ23IH44L"
}
user_details = {
  "unique_id" = "AIDAVSUVK5OYP5ASMRUFUF"
  "user_arn" = "arn:aws:iam::383704034224:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@Musfira-0514 → ~/Lab13 $ |

```

- S3 terraform.tfstate file



- Local state backup

```

@Musfira-0514 → ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace 0 Jan 5 12:09 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 5 12:09 terraform.tfstate.backup
@Musfira-0514 → ~/Lab13 $ |

```

- Terraform destroy

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

Changes to Outputs:
- access_key_id      = "AKIAVSUUK5OYJQB4WKHC" -> null
- access_key_secret  = (sensitive value) -> null
- group_details      = {
  - group_arn        = "arn:aws:iam::383704034224:group/groups/developers"
  - group_name       = "developers"
  - unique_id        = "AGPAVSUUK5OYLZ23IH44L"
} -> null
- user_details       = {
  - unique_id        = "AIDAVSUUK5OYP5ASMRUF"
  - user_arn         = "arn:aws:iam::383704034224:user/users/loadbalancer"
  - user_name        = "loadbalancer"
} -> null
null_resource.create_login_profile: Destroying... [id=6060440309344465055]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-2026010510574885]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-2026010510574885]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIAVSUUK5OYJQB4WKHC]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260105111]
aws_iam_user_group_membership.lb_membership: Destruction complete after 1s
aws_iam_access_key.lb_access_key: Destruction complete after 1s
aws_iam_user.lb: Destroying... [id=loadbalancer]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 1s
aws_iam_group_policy_attachment.change_password: Destruction complete after 1s
aws_iam_group.developers: Destroying... [id=developers]
aws_iam_group.developers: Destruction complete after 0s
aws_iam_user.lb: Destruction complete after 2s
Releasing state lock. This may take a few moments...

Destroy complete! Resources: 7 destroyed.
@Musfira-0514 ~ → ~/Lab13 $ |
```

- S3 state after destroy

```
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 2,
  "lineage": "dbe9cdb7-94a7-6eb7-1f19-8cf0e5d02aac",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

Task 7 — Create Multiple Users from CSV File

Objective: Dynamically create multiple IAM users from CSV.

- locals.tf

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

- users.csv

```
@Musfira-0514 ~ → ~/Lab13 $ touch users.csv
@Musfira-0514 ~ → ~/Lab13 $ nano users.csv
@Musfira-0514 ~ → ~/Lab13 $ cat users.csv
user_name
Michael
Dwight
Jim
Pam
Ryan
Andy
Robert
Stanley
Kevin
Angela
Oscar
Phyllis
Toby
Kelly
Darryl
Creed
Meredith
Erin
Gabe
Jan
David
Holly
Charles
Jo
Clark
Peter
@Musfira-0514 ~ → ~/Lab13 $ |
```

- main.tf multiple users

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

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

  depends_on = [aws_iam_user.users]

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

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

  user = each.value.name
}

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

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

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"
}

@Musfira-0514 → ~/Lab13 $ |
```

- Terraform init

```
@Musfira-0514 → ~/Lab13 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/null from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- Using previously-installed hashicorp/null v3.2.4

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.
```

- Terraform apply

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

Apply complete! Resources: 26 added, 0 changed, 26 destroyed.

Outputs:
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAVSVUKSOYPMXUE4U2Z"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Andy"
    "user_unique_id" = "AIDAVSVUKSOYO3I3TEEJU"
  }
  "Angela" = {
    "access_key_id" = "AKIAVSVUKSOYLT3BFRO2"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Angela"
    "user_unique_id" = "AIDAVSVUKSOYEQQJ5U42V"
  }
  "Charles" = {
    "access_key_id" = "AKIAVSVUKSOYABKMDFPV"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Charles"
    "user_unique_id" = "AIDAVSVUKSOYBNMELESVY"
  }
  "Clark" = {
    "access_key_id" = "AKIAVSVUKSOYPXOY2FAO"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Clark"
    "user_unique_id" = "AIDAVSVUKSOYKQKENM2QU"
  }
  "Creed" = {
    "access_key_id" = "AKIAVSVUKSOYIGSEELUT"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Creed"
    "user_unique_id" = "AIDAVSVUKSOYDML2YFDVR"
  }
  "Darryl" = {
    "access_key_id" = "AKIAVSVUKSOYKQETORZO"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Darryl"
    "user_unique_id" = "AIDAVSVUKSOYOUSLBOWR2"
  }
  "David" = {
    "access_key_id" = "AKIAVSVUKSOYAE4SB7FU"
    "user_arn" = "arn:aws:iam::383704034224:user/users/David"
    "user_unique_id" = "AIDAVSVUKSOYE7RACYRTX"
  }
  "Dwight" = {
    "access_key_id" = "AKIAVSVUKSOYKN3SDEFU"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Dwight"
    "user_unique_id" = "AIDAVSVUKSOYKGWS57VSG"
  }
  "Erin" = {
    "access_key_id" = "AKIAVSVUKSOYL3JUJ4ZS"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Erin"
    "user_unique_id" = "AIDAVSVUKSOYEJGAGFJ60"
  }
}
```

- Terraform output

```

@Musfira-0514 → ~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAVSVUK5OYPMXUE4U2"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Andy"
    "user_unique_id" = "AIDAVSVUK5OY03I3TEEJU"
  }
  "Angela" = {
    "access_key_id" = "AKIAVSVUK5OYLT3BFRO2"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Angela"
    "user_unique_id" = "AIDAVSVUK5OYEQJ5U42V"
  }
  "Charles" = {
    "access_key_id" = "AKIAVSVUK5OYABKMDFPV"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Charles"
    "user_unique_id" = "AIDAVSVUK5OYBNMELESVY"
  }
  "Clark" = {
    "access_key_id" = "AKIAVSVUK5OYXOY2FAO"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Clark"
    "user_unique_id" = "AIDAVSVUK5OYKQENM2QU"
  }
  "Creed" = {
    "access_key_id" = "AKIAVSVUK5OYIGSEELUT"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Creed"
    "user_unique_id" = "AIDAVSVUK5OYDML2YFDVR"
  }
  "Darryl" = {
    "access_key_id" = "AKIAVSVUK5OYKQETORZO"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Darryl"
    "user_unique_id" = "AIDAVSVUK5OYOUSLBOWR2"
  }
  "David" = {
    "access_key_id" = "AKIAVSVUK5OYAE4SB7FU"
    "user_arn" = "arn:aws:iam::383704034224:user/users/David"
    "user_unique_id" = "AIDAVSVUK5OY7RACYRTX"
  }
  "Dwight" = {
    "access_key_id" = "AKIAVSVUK5OYKN35DEFU"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Dwight"
    "user_unique_id" = "AIDAVSVUK5OYKGS57VSG"
  }
  "Erin" = {
    "access_key_id" = "AKIAVSVUK5OYL3J4ZS"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Erin"
    "user_unique_id" = "AIDAVSVUK5OYJGAGFJ60"
  }
  "Gabe" = {
    "access_key_id" = "AKIAVSVUK5OYHQCWA4WB"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Gabe"
    "user_unique_id" = "AIDAVSVUK5OYLXIEGVIDW"
  }
  "Holly" = {
    "access_key_id" = "AKIAVSVUK5OYCLOCQ6HJ"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Holly"
    "user_unique_id" = "AIDAVSVUK5OYFVCGRI6EF"
  }
  "Jan" = {
    "access_key_id" = "AKIAVSVUK5OYPM2RZHJ"
    "user_arn" = "arn:aws:iam::383704034224:user/users/Jan"
    "user_unique_id" = "AIDAVSVUK5OYAFQ4LRLUW"
  }
}

```

- tfstate secrets

```

@Musfira-0514 → ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
@Musfira-0514 → ~/Lab13 $ |

```

- AWS Console all users

Users (28) info

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

Q Search

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active key age	Access key last use
<input type="checkbox"/>	Andy	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Angela	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Charles	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Clark	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Creed	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Darryl	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	David	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Dwight	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Erin	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Gabe	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Holly	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Jan	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Jim	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Jo	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Kelly	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Kevin	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	lab-user	/	0	5 minutes ago	-	5 days	-	Active - AKIAVSUUKSO...	5 days	5 minutes ago
<input type="checkbox"/>	Meredith	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Michael	/users/	1	-	-	4 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-
<input type="checkbox"/>	Oscar	/users/	1	-	-	5 minutes	-	Active - AKIAVSUUKSO...	8 minutes	-

- AWS Console group members

Users (26) | **Permissions** | **Access Advisor**

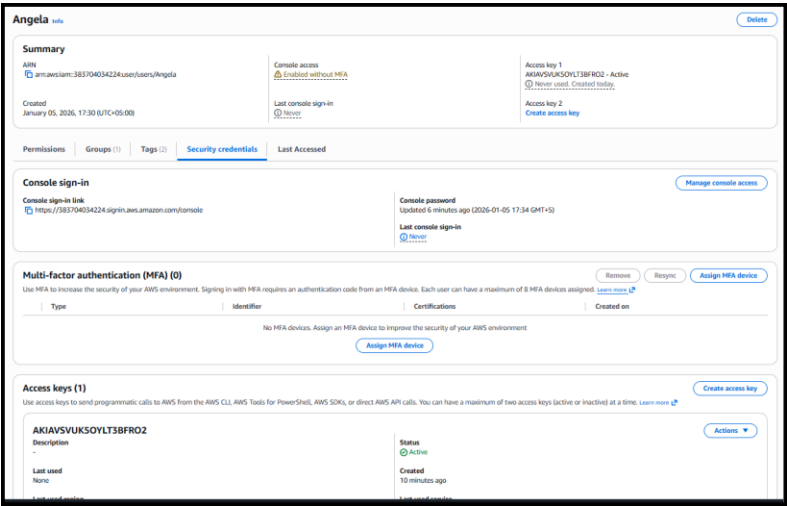
Users in this group (26)

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

Q Search

- ☐ **User name** ⓘ
- ☐ [Andy](#)
- ☐ [Angela](#)
- ☐ [Charles](#)
- ☐ [Clark](#)
- ☐ [Creed](#)
- ☐ [Darryl](#)
- ☐ [David](#)
- ☐ [Dwight](#)
- ☐ [Erin](#)
- ☐ [Gabe](#)
- ☐ [Holly](#)
- ☐ [Jan](#)
- ☐ [Jim](#)
- ☐ [Jo](#)
- ☐ [Kelly](#)
- ☐ [Kevin](#)
- ☐ [Meredith](#)
- ☐ [Michael](#)
- ☐ [Oscar](#)
- ☐ [Pam](#)

- AWS Console user access key



- S3 state for multiple users


```
{} terraform (1).json X
D: > Downloads > Downloads > {} terraform (1).json > ...
6      "outputs": {
69      "all_users_details": {
70      "value": {
71      "Andy": {
72      "access_key_id": "AKIAVSVUK5OYPMXUE4UZ",
73      "user_arn": "arn:aws:iam::383704034224:user/users/Andy",
74      "user_unique_id": "AIDAVSVUK5OY03I3TEEJU"
75      },
76      "Angela": {
77      "access_key_id": "AKIAVSVUK5OYLT3BFRO2",
78      "user_arn": "arn:aws:iam::383704034224:user/users/Angela",
79      "user_unique_id": "AIDAVSVUK5OYEQQJ5U42V"
80      },
81      "Charles": {
82      "access_key_id": "AKIAVSVUK5OYABKMDFPV",
83      "user_arn": "arn:aws:iam::383704034224:user/users/Charles",
84      "user_unique_id": "AIDAVSVUK5OYBNMELESVY"
85      },
86      "Clark": {
87      "access_key_id": "AKIAVSVUK5OYPX0Y2FA0",
88      "user_arn": "arn:aws:iam::383704034224:user/users/Clark",
89      "user_unique_id": "AIDAVSVUK5OYKQKENM2QU"
90      },
91      "Creed": {
92      "access_key_id": "AKIAVSVUK5OYIGSEELUT",
93      "user_arn": "arn:aws:iam::383704034224:user/users/Creed",
94      "user_unique_id": "AIDAVSVUK5OYDML2YFDVR"
95      },
96      "Darryl": {
97      "access_key_id": "AKIAVSVUK5OYKQETORZO",
98      "user_arn": "arn:aws:iam::383704034224:user/users/Darryl",
99      "user_unique_id": "AIDAVSVUK5OYOUSLBOWR2"
100     },
101     "David": {
102     "access_key_id": "AKIAVSVUK5OYAE4SB7FU",
103     "user_arn": "arn:aws:iam::383704034224:user/users/David",
104     "user_unique_id": "AIDAVSVUK5OYE7RACYRTX"
105     },
106     "Dwight": {
107     "access_key_id": "AKIAVSVUK5OYKN35DEFU",
108     "user_arn": "arn:aws:iam::383704034224:user/users/Dwight",
109     "user_unique_id": "AIDAVSVUK5OYKGWS57VSG"
110     },
111     "Erin": {
112     "access_key_id": "AKIAVSVUK5OYL3JUJ4ZS",
113     "user_arn": "arn:aws:iam::383704034224:user/users/Erin",
114     "user_unique_id": "AIDAVSVUK5OYEJGAGFJ60"
115     },
116     "Gabe": {
```

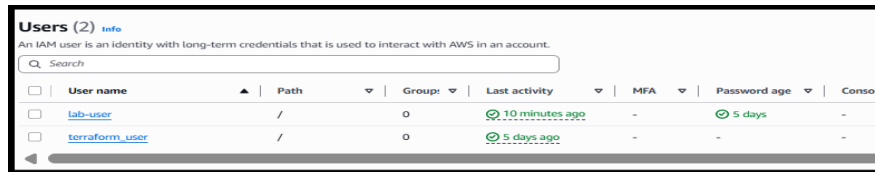
Cleanup — Destroy Resources & Verify

- Terraform destroy complete

```
aws_iam_user.users["Jo"]: Destruction complete after 2s
aws_iam_user.users["Jan"]: Destroying... [id=Jan]
aws_iam_user.users["Andy"]: Destruction complete after 3s
aws_iam_user.users["Oscar"]: Destroying... [id=Oscar]
aws_iam_user.users["Robert"]: Destruction complete after 3s
aws_iam_user.users["Jim"]: Destroying... [id=Jim]
aws_iam_user.users["Michael"]: Destruction complete after 2s
aws_iam_user.users["Ryan"]: Destruction complete after 2s
aws_iam_user.users["Stanley"]: Destruction complete after 5s
aws_iam_user.users["Darryl"]: Destruction complete after 8s
aws_iam_user.users["Clark"]: Destruction complete after 4s
aws_iam_user.users["Peter"]: Destruction complete after 4s
aws_iam_user.users["Jan"]: Destruction complete after 3s
aws_iam_user.users["Oscar"]: Destruction complete after 2s
aws_iam_user.users["Erin"]: Destruction complete after 9s
aws_iam_user.users["Jim"]: Destruction complete after 4s
Releasing state lock. This may take a few moments...

Destroy complete! Resources: 107 destroyed.
@Musfira-0514 → ~/Lab13 $ |
```

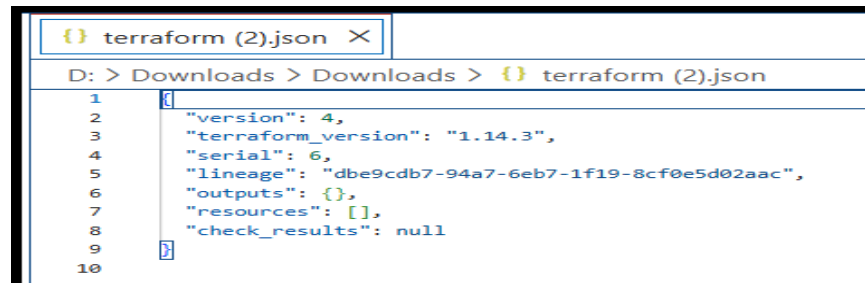
- AWS Console users deleted



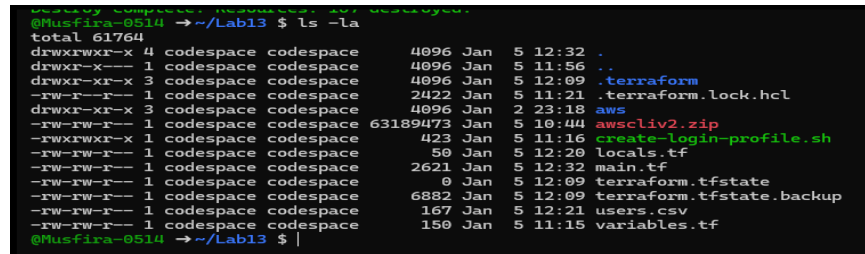
- AWS Console group deleted



- S3 empty state



- Final project files



- S3 bucket deleted (optional)

