CLOUD COMPUTING LAB: 13

Submitted By:

Seerat Fatima

Registration. No:

2023-BSE-060

Submitted To:

Engr. Shoaib

Section:

5B

Task 0 Lab Setup (Codespace & GH CLI)

```
PS C:\Users\Waseem\Documents\github\Lab13> gh codespace list
NAME                              DISPLAY NAME         REPOSITORY              BRANCH  STATE      CREATED AT
verbose-system-wrq7q47rjx7vcgq46  verbose system       SeratFatima00/Lab9      main*   Shutdown   about 20 days ago
jubilant-happiness-7vj9jx9v4r7p...  jubilant happiness  SeratFatima00/CC_See...  main*   Shutdown   about 19 days ago
fuzzy-doodle-97g9gx974j5q2x4wx    fuzzy doodle         SeratFatima00/CC_See...  main*   Shutdown   about 18 days ago
expert-doodle-jjp5pg5jw767fpq6j   expert doodle        SeratFatima00/Lab13     main    Available  about 1 minute ago
```

```
@SeratFatima00 ➜ /workspaces/Lab13 (main) $
```

Task 1 — Create IAM Group and Output Details

```
@SeratFatima00 ➜ /workspaces/Lab13 (main) $ mkdir -p ~/Lab13
@SeratFatima00 ➜ /workspaces/Lab13 (main) $ cd ~/Lab13
```

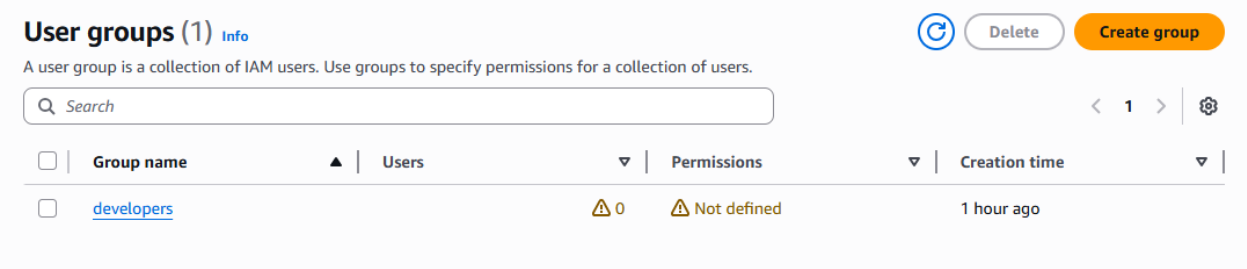
```
@SeratFatima00 ➜ ~/Lab13 $ touch main.tf
@SeratFatima00 ➜ ~/Lab13 $ vim main.tf
```

```
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
```

```
@SeratFatima00 ➜ ~/Lab13 $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (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.
```

```
@SeratFatima00 ~/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 2s [id=developers]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
```

```
@SeratFatima00 ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
```

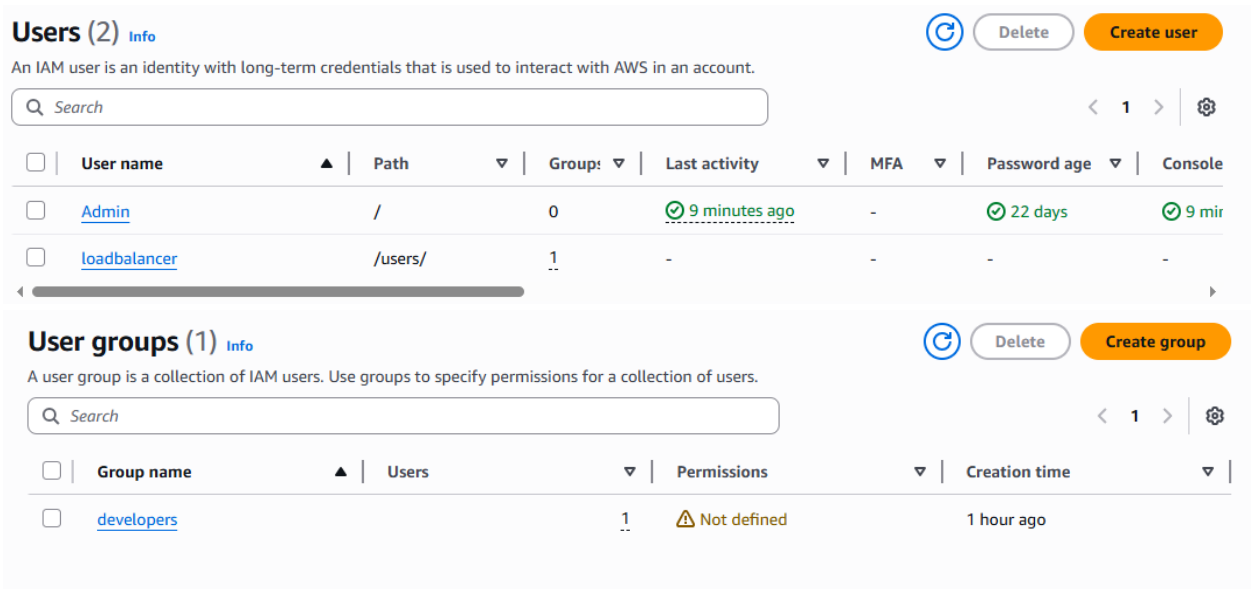


Task 2 — Create IAM User with Group Membership

```
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
  }
}
```

```
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-20260117054218700000000001]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

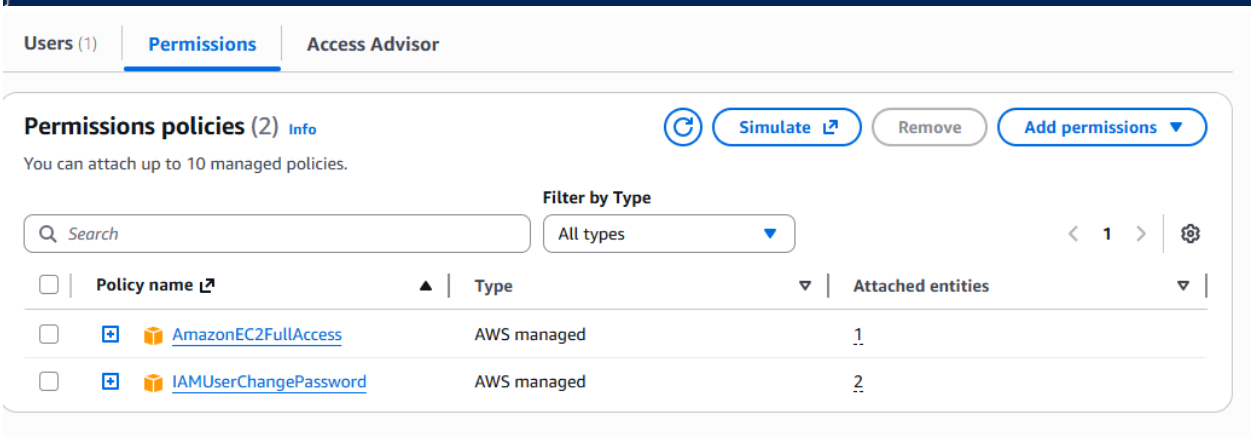
```
@SeratFatima00 ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```



Task 3 — Attach Policies to IAM Group

```
resource "aws_iam_group" "developers" {
  name = "developers"
  path = "/groups/"
}

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

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"
}
```

```
@SeratFatima00 ~/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-20260117054218700000000001]

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.developer_ec2_fullaccess: Creating...
aws_iam_group_policy_attachment.change_password: Creating...
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-20260117054631955300000001]
aws_iam_group_policy_attachment.change_password: Creation complete after 1s [id=developers-20260117054631975900000002]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```



Task 4 — Create Login Profile for IAM User

```
variable "iam_password"
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "1dontKnow"


~
~
```

```
#!/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
```

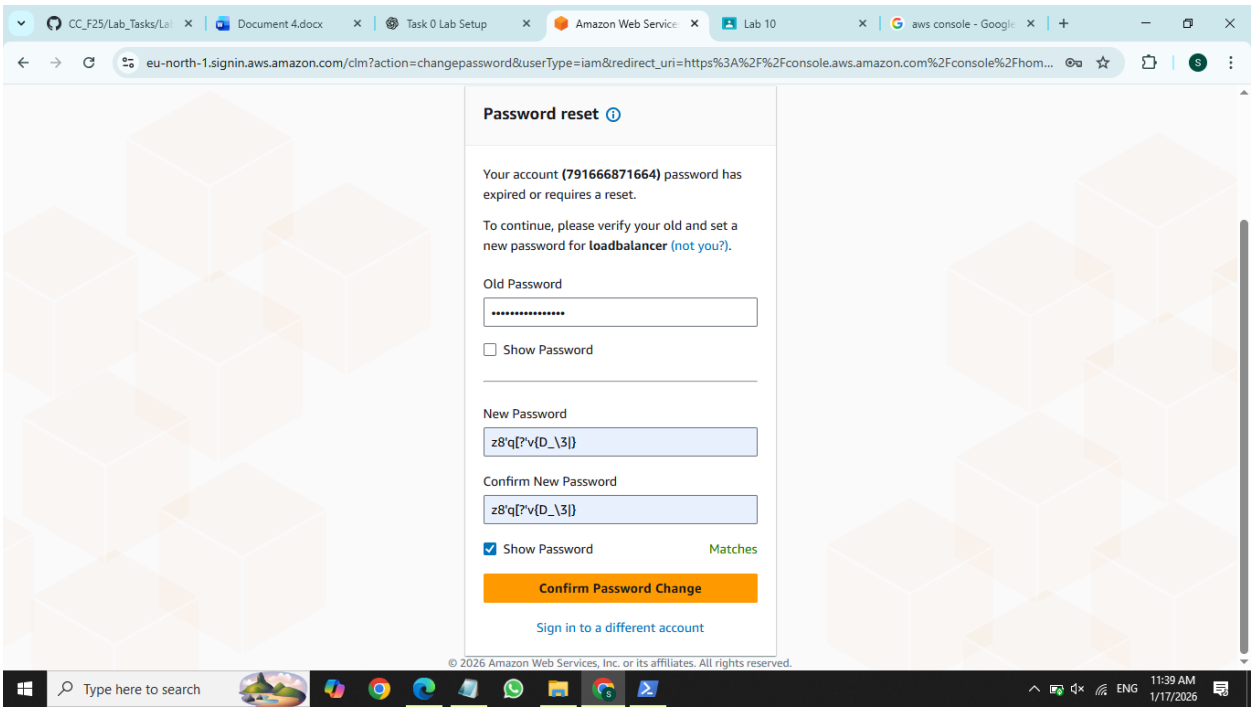
```
@SeratFatima00 ~/Lab13 $ vim create-login-profile.sh
@SeratFatima00 ~/Lab13 $ chmod +x create-login-profile.sh
@SeratFatima00 ~/Lab13 $
```

```
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
  ]
}
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}'"
  }
}
```

```
@SeratFatima00 ➜ ~/Lab13 $ chmod +x create-login-profile.sh
@SeratFatima00 ➜ ~/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=4309750621051721465]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260117054631975900000002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260117054631955300000001]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260117054218700000000001]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # null_resource.create_login_profile is tainted, so must be replaced
-/+ resource "null_resource" "create_login_profile" {
      ~ id       = "4309750621051721465" -> (known after apply)
        # (1 unchanged attribute hidden)
    }

Plan: 1 to add, 0 to change, 1 to destroy.
null_resource.create_login_profile: Destroying... [id=4309750621051721465]
null_resource.create_login_profile: Destruction complete after 0s
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 4s [id=5454148332018321568]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```


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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```


```
@SeratFatima00 ➜ ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-17T06:35:46+00:00",
        "PasswordResetRequired": true
    }
}
```



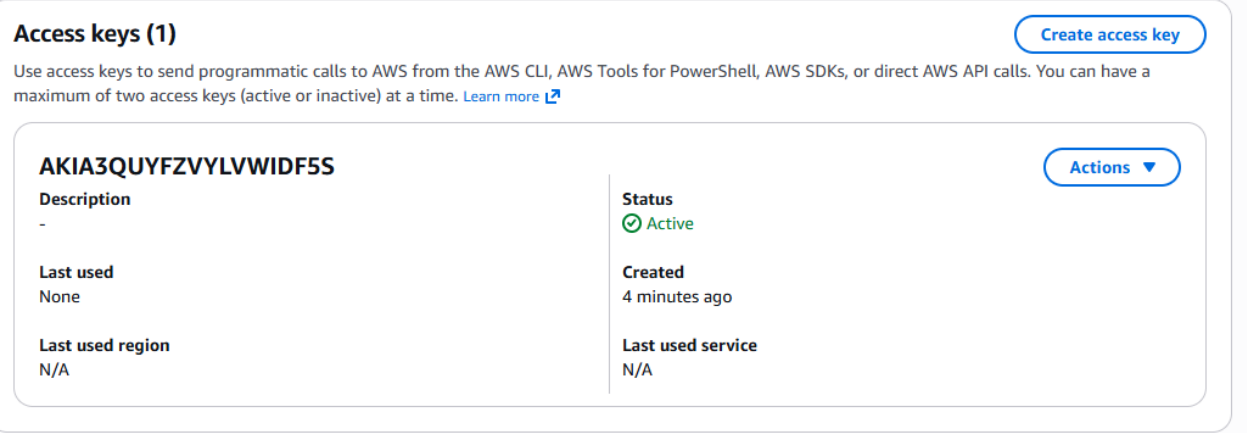
Task 5 — Generate Access Keys for IAM User

```
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
}

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"
}
```

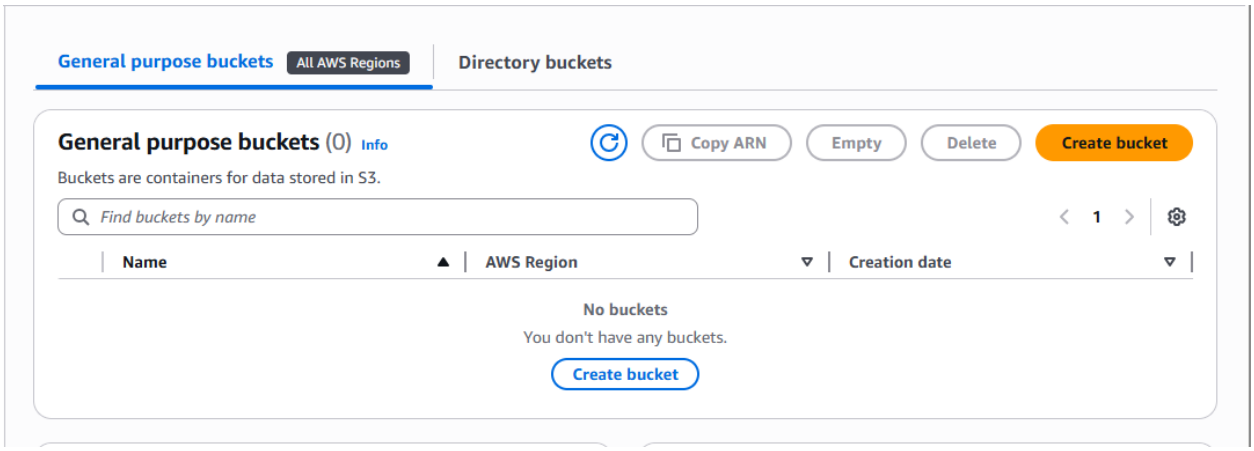
```
@SeratFatima00 ~/Lab13 $ terraform apply -auto-approve \
>   -var="iam_password=MySecurePass123!" \
>   -var="aws_access_key=AKIA3QUYFZVYG5PZITV4" \
>   -var="aws_secret_key=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" \
>   -var="aws_region=eu-north-1"
aws_iam_group.developers: Refreshing state... [id=developers]
aws_iam_user.lb: Refreshing state... [id=loadbalancer]
null_resource.create_login_profile: Refreshing state... [id=5454148332018321568]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260117054631975900000002]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260117054218700000000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260117054631955300000001]

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=AKIA3QUYFZVYLVWIDF5S]

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

Outputs:

access_key_id = "AKIA3QUYFZVYLVWIDF5S"
access_key_secret = <sensitive>
group_details = {
```

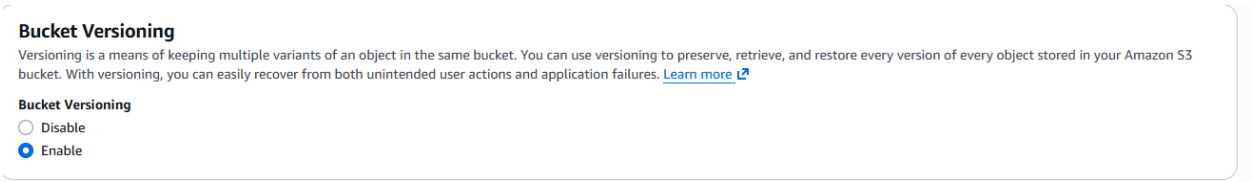
```
@SeratFatima00 ~/Lab13 $ terraform output
access_key_id = "AKIA3QUYFZVYLVWIDF5S"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

```
@SeratFatima00 ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
    "access_key_secret": {
      "value": "ZLuaPpsMvYWsmO8Yud9dcI0j3sslbauULF1f3Oab",
      "type": "string",
      "sensitive": true
    },
    "group_details": {
      "value": {
        "group_arn": "arn:aws:iam::791666871664:group/groups/developers",
        "group_name": "developers",
        "unique_id": "AGPA3QUYFZVYOD6FKOKP4"
      },
```



Task 6 — Implement Terraform Remote State with S3

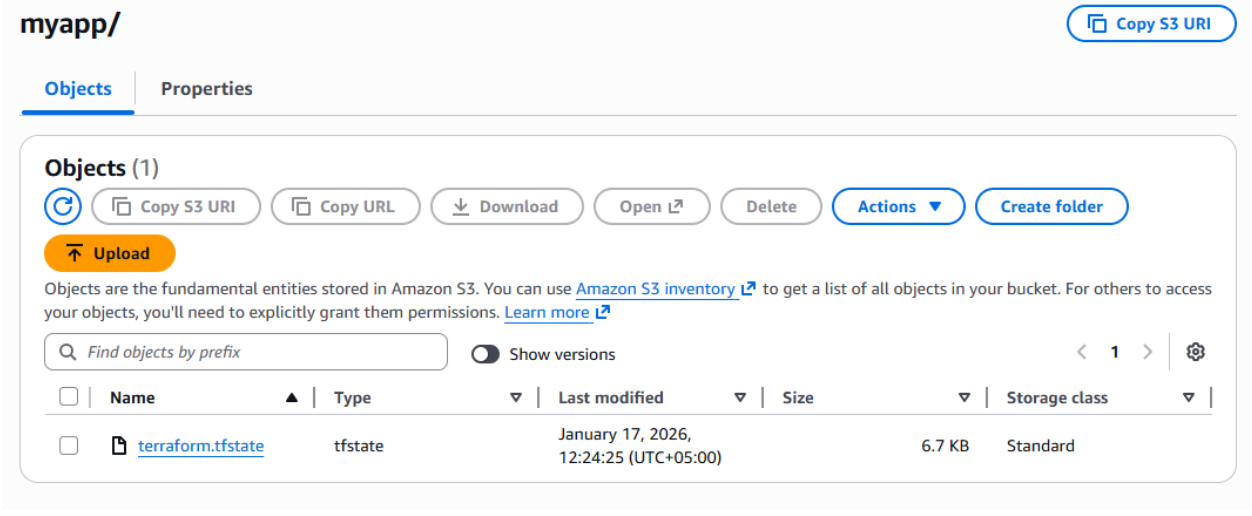





```
terraform {
  backend "s3" {
    bucket       = "myapp-s3-bucket-demo"
    key          = "myapp/terraform.tfstate"
    region       = "eu-north-1"       # your AWS region
    encrypt      = true
    use_lockfile = true
  }
}
provider "aws" {
  access_key = var.aws_access_key
  secret_key = var.aws_secret_key
  region     = var.aws_region
}
```

```
@SeratFatima00 ~/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


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 v6.28.0

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.
```

```
@SeratFatima00 ~/Lab13 $ terraform apply -auto-approve \
>   -var="iam_password=MySecurePass123!" \
>   -var="aws_access_key=AKIA3QUYFZVYG5PZITV4" \
>   -var="aws_secret_key=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" \
>   -var="aws_region=eu-north-1"
aws_iam_group.developers: Refreshing state... [id=developers]
aws_iam_user.lb: Refreshing state... [id=loadbalancer]
null_resource.create_login_profile: Refreshing state... [id=5454148332018321568]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIA3QUYFZVYLVWIDF5S]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260117054631975900000002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260117054631955300000001]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260117054218700000000001]

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.

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

Outputs:

access_key_id = "AKIA3QUYFZVYLVWIDF5S"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYOD6FKOKP4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYPEQKCDGVZ"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```




```
@SeratFatima00 ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace    0 Jan 17 07:24 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 17 07:24 terraform.tfstate.backup
```

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

Changes to Outputs:
  - access_key_id     = "AKIA3QUYFZVYLVWIDF5S" -> null
  - access_key_secret = (sensitive value) -> null
  - group_details     = {
      - group_arn  = "arn:aws:iam::791666871664:group/groups/developers"
      - group_name = "developers"
      - unique_id  = "AGPA3QUYFZVYOD6FKOKP4"
    } -> null
  - user_details      = {
      - unique_id = "AIDA3QUYFZVYPEQKCDGVZ"
      - user_arn  = "arn:aws:iam::791666871664:user/users/loadbalancer"
      - user_name = "loadbalancer"
    } -> null
null_resource.create_login_profile: Destroying... [id=5454148332018321568]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260117054631955300000001]
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-20260117054218700000000001]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIA3QUYFZVYLVWIDF5S]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260117054631975900000002]
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.change_password: Destruction complete after 1s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: 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 3s
```

Task 7 — Create Multiple Users from CSV File

```
locals
  users = csvdecode file "users.csv"


~
~
```

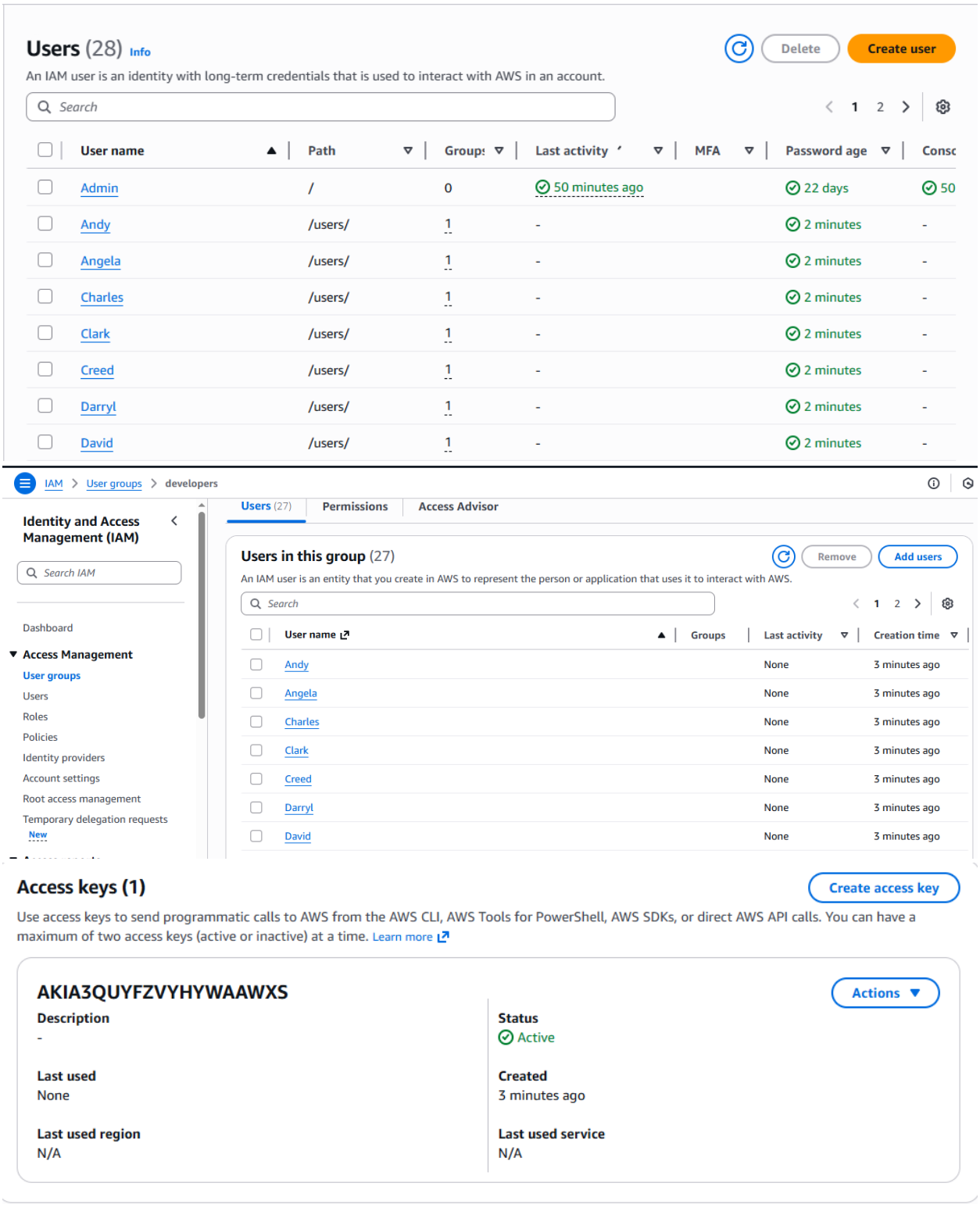
```
user_name
Michael
Dwight
Jim
Pam
Ryan
Andy
Robert
Stanley
Kevin
Angela
Oscar
Phyllis
Toby
Kelly
Darryl
Creed
Meredith
Erin
Gabe
Jan
David
Holly
Charles
Jo
Clark
Peter

~
~
~
~
~
```

```hcl
# Create multiple IAM users from CSV
resource "aws_iam_user" "users" {
  for_each = { for user in local.users : user.user_name => user }

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

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

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

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

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

  triggers = {
    password_hash = sha256(var.iam_password)
    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" {
```

```
@SeratFatima00 ▣ ~/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 v6.28.0
- 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.

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.
```

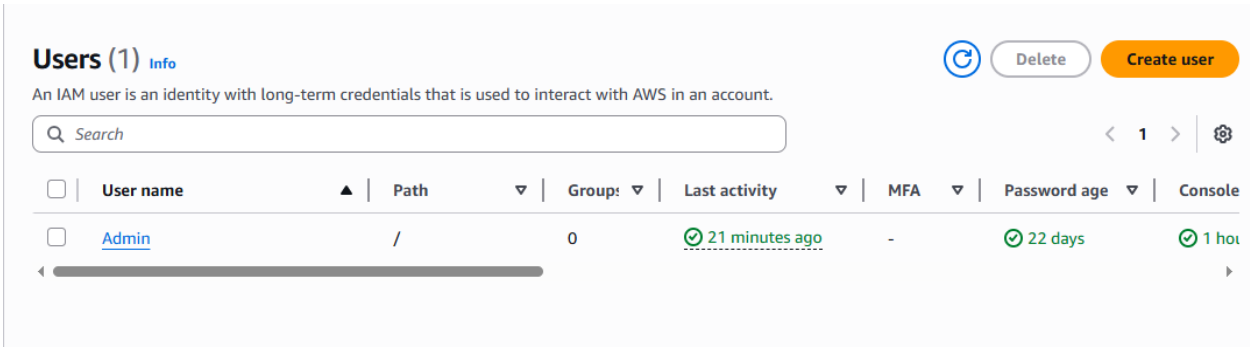
```
    "user_arn" = "arn:aws:iam::791666871664:user/users/Oscar"
    "user_unique_id" = "AIDA3QUYFZVYD2D3NQDIA"
  }
  "Pam" = {
    "access_key_id" = "AKIA3QUYFZVYN6E324ES"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Pam"
    "user_unique_id" = "AIDA3QUYFZVYGIJ6QOJ4Y"
  }
  "Peter" = {
    "access_key_id" = "AKIA3QUYFZVYNETV5UUC"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Peter"
    "user_unique_id" = "AIDA3QUYFZVYF73O5ONSU"
  }
  "Phyllis" = {
    "access_key_id" = "AKIA3QUYFZVYMQ5OPEGW"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Phyllis"
    "user_unique_id" = "AIDA3QUYFZVYALHPOLTP6"
  }
  "Robert" = {
    "access_key_id" = "AKIA3QUYFZVYOCP46U32"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Robert"
    "user_unique_id" = "AIDA3QUYFZVYJELLTI5SE"
  }
  "Ryan" = {
    "access_key_id" = "AKIA3QUYFZVYB5GFYZM2"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Ryan"
    "user_unique_id" = "AIDA3QUYFZVYOZL2YA3EW"
  }
  "Stanley" = {
    "access_key_id" = "AKIA3QUYFZVYOFFVETGZ"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Stanley"
    "user_unique_id" = "AIDA3QUYFZVYEGIZGC4MW"
  }
  "Toby" = {
    "access_key_id" = "AKIA3QUYFZVYFWTLYXLI"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Toby"
    "user_unique_id" = "AIDA3QUYFZVYMIABO5BQV"
  }
}
group_details = {
  "group_arn" = "arn:aws:iam::791666871664:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA3QUYFZVYE4UHHJQO4"
}
user_details = {
  "unique_id" = "AIDA3QUYFZVYM67GMAF2N"
  "user_arn" = "arn:aws:iam::791666871664:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

```
@SeratFatima00 ~/Lab13 $ terraform output
access_key_id = "AKIA3QUYFZVYMUKAIMJS"
access_key_secret = <sensitive>
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIA3QUYFZVYPYS5A42B"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Andy"
    "user_unique_id" = "AIDA3QUYFZVYDTOABLRZP"
  }
  "Angela" = {
    "access_key_id" = "AKIA3QUYFZVYCKOKBB4A"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Angela"
    "user_unique_id" = "AIDA3QUYFZVYKHWKQMRTJ"
  }
  "Charles" = {
    "access_key_id" = "AKIA3QUYFZVYDZMVL7WV"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Charles"
    "user_unique_id" = "AIDA3QUYFZVYFAFTX3OWF"
  }
  "Clark" = {
    "access_key_id" = "AKIA3QUYFZVYO63NM2WU"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Clark"
    "user_unique_id" = "AIDA3QUYFZVYI6ETF6MAZ"
  }
  "Creed" = {
    "access_key_id" = "AKIA3QUYFZVYLX5ZFADO"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Creed"
    "user_unique_id" = "AIDA3QUYFZVYEPEE6YODL"
  }
  "Darryl" = {
    "access_key_id" = "AKIA3QUYFZVYDZWATD3Y"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Darryl"
    "user_unique_id" = "AIDA3QUYFZVYEPG7WUWM3"
  }
  "David" = {
    "access_key_id" = "AKIA3QUYFZVYHFLJQFGE"
    "user_arn" = "arn:aws:iam::791666871664:user/users/David"
    "user_unique_id" = "AIDA3QUYFZVYEBNXKPGCJ"
  }
  "Dwight" = {
    "access_key_id" = "AKIA3QUYFZVYNXWCFFB6"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Dwight"
    "user_unique_id" = "AIDA3QUYFZVYGNA7VIPUR"
  }
  "Erin" = {
    "access_key_id" = "AKIA3QUYFZVYHFACIR7B"
    "user_arn" = "arn:aws:iam::791666871664:user/users/Erin"
```


Users (28) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user
Search
1 2
User name
Path
Groups
Last activity
MFA
Password age
Admin / 0 50 minutes ago 22 days
Andy /users/ 1 - 2 minutes -
Angela /users/ 1 - 2 minutes -
Charles /users/ 1 - 2 minutes -
Clark /users/ 1 - 2 minutes -
Creed /users/ 1 - 2 minutes -
Darryl /users/ 1 - 2 minutes -
David /users/ 1 - 2 minutes -
IAM > User groups > developers
Identity and Access Management (IAM)
Search IAM
Dashboard
Access Management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management
Temporary delegation requests
New
Users (27)
Permissions
Access Advisor
Users in this group (27)
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
Remove
Add users
Search
1 2
User name
Groups
Last activity
Creation time
Andy None 3 minutes ago
Angela None 3 minutes ago
Charles None 3 minutes ago
Clark None 3 minutes ago
Creed None 3 minutes ago
Darryl None 3 minutes ago
David None 3 minutes ago
Access keys (1)
Create access key
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
AKIA3QUYFZVYHYWAAWXS
Actions
Description
-
Status
Active
Last used
None
Created
3 minutes ago
Last used region
N/A
Last used service
N/A

Cleanup

```
aws_iam_user.users["Erin"]: Destroying... [id=Erin]
aws_iam_user.users["Jim"]: Destroying... [id=Jim]
aws_iam_group.developers: Destruction complete after 2s
aws_iam_user.users["Ryan"]: Destroying... [id=Ryan]
aws_iam_user.users["Phyllis"]: Destruction complete after 4s
aws_iam_user.users["Angela"]: Destroying... [id=Angela]
aws_iam_user.users["Jim"]: Destruction complete after 4s
aws_iam_user.users["Kevin"]: Destroying... [id=Kevin]
aws_iam_user.users["Charles"]: Destruction complete after 4s
aws_iam_user.users["Toby"]: Destroying... [id=Toby]
aws_iam_user.users["David"]: Destruction complete after 4s
aws_iam_user.users["Holly"]: Destroying... [id=Holly]
aws_iam_user.users["Clark"]: Destruction complete after 4s
aws_iam_user.users["Michael"]: Destroying... [id=Michael]
aws_iam_user.users["Ryan"]: Destruction complete after 3s
aws_iam_user.users["Pam"]: Destroying... [id=Pam]
aws_iam_user.users["Creed"]: Destruction complete after 5s
aws_iam_user.users["Darryl"]: Destroying... [id=Darryl]
aws_iam_user.users["Meredith"]: Destruction complete after 6s
aws_iam_user.users["Peter"]: Destroying... [id=Peter]
aws_iam_user.users["Andy"]: Destruction complete after 6s
aws_iam_user.users["Stanley"]: Destroying... [id=Stanley]
aws_iam_user.users["Erin"]: Destruction complete after 6s
aws_iam_user.users["Jan"]: Destroying... [id=Jan]
aws_iam_user.users["Angela"]: Destruction complete after 3s
aws_iam_user.users["Oscar"]: Destroying... [id=Oscar]
aws_iam_user.users["Kevin"]: Destruction complete after 3s
aws_iam_user.users["Robert"]: Destroying... [id=Robert]
aws_iam_user.users["Michael"]: Destruction complete after 3s
aws_iam_user.users["Dwight"]: Destroying... [id=Dwight]
aws_iam_user.users["Darryl"]: Destruction complete after 3s
aws_iam_user.users["Gabe"]: Destroying... [id=Gabe]
aws_iam_user.users["Holly"]: Destruction complete after 4s
aws_iam_user.users["Jo"]: Destroying... [id=Jo]
aws_iam_user.users["Pam"]: Destruction complete after 3s
aws_iam_user.users["Kelly"]: Destroying... [id=Kelly]
aws_iam_user.users["Peter"]: Destruction complete after 2s
aws_iam_user.users["Stanley"]: Destruction complete after 3s
aws_iam_user.users["Jan"]: Destruction complete after 3s
aws_iam_user.users["Robert"]: Destruction complete after 3s
aws_iam_user.users["Dwight"]: Destruction complete after 3s
aws_iam_user.users["Gabe"]: Destruction complete after 3s
aws_iam_user.users["Kelly"]: Destruction complete after 3s
aws_iam_user.users["Jo"]: Destruction complete after 5s
aws_iam_user.users["Oscar"]: Destruction complete after 6s
aws_iam_user.users["Toby"]: Still destroying... [id=Toby, 00m10s elapsed]
aws_iam_user.users["Toby"]: Destruction complete after 16s

Destroy complete! Resources: 111 destroyed.
```



User groups (0) Info

Delete | Create group

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Search

1

Group name	Users	Permissions	Creation time

No resources to display

```
@SeratFatima00 ~/Lab13 $ ls -la
total 65364
drwxrwxr-x 4 codespace codespace     4096 Jan 17 08:09 .
drwxr-x--- 1 codespace codespace     4096 Jan 17 07:41 ..
drwxr-xr-x 3 codespace codespace     4096 Jan 17 07:24 .terraform
-rw-r--r-- 1 codespace codespace     2422 Jan 17 05:53 .terraform.lock.hcl
drwxr-xr-x 3 codespace codespace     4096 Jan 16 19:26 aws
-rw-rw-r-- 1 codespace codespace 66875639 Jan 17 04:08 awscliv2.zip
-rwxrwxr-x 1 codespace codespace      456 Jan 17 06:35 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace       51 Jan 17 07:35 locals.tf
-rw-rw-r-- 1 codespace codespace     3565 Jan 17 07:41 main.tf
-rw-rw-r-- 1 codespace codespace        0 Jan 17 07:24 terraform.tfstate
-rw-rw-r-- 1 codespace codespace     6882 Jan 17 07:24 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace      168 Jan 17 07:37 users.csv
-rw-rw-r-- 1 codespace codespace      447 Jan 17 06:50 variables.tf
```

Delete bucket Info

This bucket is not empty

Buckets must be empty before they can be deleted.

Empty bucket | Diagnose with Amazon Q

Delete bucket "myapp-s3-bucket-demo1"?

To confirm deletion, enter the name of the bucket in the text input field.

myapp-s3-bucket-demo1

Cancel | Delete bucket