



FATIMA JINNAH WOMEN UNIVERSITY

Submitted To:

Engr. Muhammad Shoaib

Submitted By:

Irraj Samreen(027)

Section:

5A

CLOUD COMPUTING

LAB 13

TASK1

```
is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
● @23-22411-027-max → /workspaces/LAB13 (main) $ mkdir -p ~/Lab13
cd ~/Lab13
pwd
/home/codespace/Lab13
● @23-22411-027-max → ~/Lab13 $ touch main.tf
ls -l
total 0
-rw-r--r-- 1 codespace codespace 0 Jan  5 14:39 main.tf
● @23-22411-027-max → ~/Lab13 $ 
```

```
/home/codespace/Lab13
● @23-22411-027-max → ~/Lab13 $ touch main.tf
ls -l
total 0
-rw-r--r-- 1 codespace codespace 0 Jan  5 14:39 main.tf
● @23-22411-027-max → ~/Lab13 $ touch main.tf
● @23-22411-027-max → ~/Lab13 $ ls -l
total 0
-rw-r--r-- 1 codespace codespace 0 Jan  5 14:45 main.tf
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 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
    }
}
```

File Name to Write: main.tf

^G Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T Browse

```
● @23-22411-027-max →~/Lab13 $ nano main.tf
● @23-22411-027-max →~/Lab13 $ cat main.tf
provider "aws" {
    shared_config_files      = ["~/.aws/config"]
    shared_credentials_files = ["~/.aws/credentials"]
}

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

output "group_details" {
    value = {
        group_name = aws_iam_group.developers.name
        group_arn  = aws_iam_group.developers.arn
        unique_id  = aws_iam_group.developers.unique_id
    }
}
○ @23-22411-027-max →~/Lab13 $
```

```
● @23-22411-027-max → ~/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.
```

```
Commands will detect it and remind you to do so if necessary.
● @23-22411-027-max → ~/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::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id"  = "AGPATC2FAR6CG3BAH6M2W"
}

○ @23-22411-027-max → ~/Lab13 $ █
↳ 0 ↳ 0
```

```
J
● @23-22411-027-max → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
○ @23-22411-027-max → ~/Lab13 $
```

User groups (2) <small>Info</small>				<small>⟳</small>	Delete	Create group	
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.				<	1	>	⟳
<input type="checkbox"/>	Group name	▼ Users	▲ Permissions	▼	Creation time	▼	
<input type="checkbox"/>	developers		⚠ 0 ⚠ Not defined		7 minutes ago		
<input type="checkbox"/>	MyGroupCli		⚠ 0 ⚡ Defined		Yesterday		

Task 2

```
    }

● @23-22411-027-max →~/Lab13 $ cd ~/Lab13
● @23-22411-027-max →~/Lab13 $ nano main.tf
● @23-22411-027-max →~/Lab13 $ cat main.tf

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

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

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

resource "aws_iam_user" "lb" {
    name          = "loadbalancer"
    path          = "/users/"
    force_destroy = true

    tags = {
        DisplayName = "Load Balancer"
    }
}

resource "aws_iam_user_group_membership" "lb_membership" {
```

```
    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
  }
}
```

```
○ @23-22411-027-max ➔ ~/Lab13 $
```

```
@23-22411-027 ~/Lab13 $ terraform apply -auto-approve
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 = {
```

```
# 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 0s [id=terraform-20260105152043021200000001]

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

Outputs:

group_details = {
    "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
    "group_name" = "developers"
    "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
    "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
    "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
    "user_name" = "loadbalancer"
}
@23-22411-027-max →~/Lab13 $
```

```
}
```

- @23-22411-027-max →~/Lab13 \$ terraform output

```
group_details = {
    "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
    "group_name" = "developers"
    "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
    "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
    "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
    "user_name" = "loadbalancer"
}
@23-22411-027-max →~/Lab13 $
```

Users (3) 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
Admin	/	0	33 minutes ago	-	-	-
Admin1	/	0	6 days ago	-	6 days	-
loadbalancer	/users/	1	-	-	-	-

loadbalancer Info

Summary

ARN arn:aws:iam::212208750468:user/users/loadbalancer	Console access Disabled	Access key 1 Create access key
Created January 05, 2026, 20:20 (UTC+05:00)	Last console sign-in -	

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

Permissions policies

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
		Loading policies

Permissions boundary (not set)

Generate policy based on CloudTrail events

loadbalancer Info

Summary

ARN arn:aws:iam::212208750468:user/users/loadbalancer	Console access Disabled	Access key 1 Create access key
Created January 05, 2026, 20:20 (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 at a time.

Group name	Attached policies
developers	-

Task 3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 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
    }
}

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" {
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^J Justify ^/ Go To Line M-E Redo

Δ 0 ⌂ 0

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 main.tf *

```
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"
}
```

Help Write Out Where Is Cut Execute Location Undo
Exit Read File Replace Paste Justify Go To Line Redo

```
● @23-22411-027-max → ~/Lab13 $ cat main.tf
provider "aws" {
    shared_config_files      = ["~/.aws/config"]
    shared_credentials_files = ["~/.aws/credentials"]
}

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

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

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
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
@23-22411-027-max → ~/Lab13 $ cat main.tf
}

}

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"
}
```

○ @23-22411-027-max → ~/Lab13 \$

Δ 0 ⌂ 0

```
● @23-22411-027-max ~/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-20260105152043021200000001]

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.change_password: Creation complete after 1s [id=developers-202601051529140497000
00002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-202601051529
14045500000001]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
```

```

# 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.change_password: Creation complete after 1s [id=developers-202601051529140497000
00002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Creation complete after 1s [id=developers-202601051529
140455000000001]

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
  "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
  "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
○ @23-22411-027-max → ~/Lab13 $ 

```

The screenshot shows the AWS IAM Groups page. At the top, there's a summary for the 'developers' group:

- Summary:**
 - User group name: developers
 - Creation time: January 05, 2026, 19:49 (UTC+05:00)
 - ARN: arn:aws:iam::212208750468:group/groups/developers
- Permissions:** This tab is selected, showing two attached policies:
 - IAMUserChangePassword (AWS managed, Type: AWS managed, Attached entities: 3)
 - AmazonEC2FullAccess (AWS managed, Type: AWS managed, Attached entities: 1)
- Access Advisor:** This tab is also present but not selected.

Task 4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 variables.tf *

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

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

● @23-22411-027-max →~/Lab13 \$ cd ~/Lab13
● @23-22411-027-max →~/Lab13 \$ nano variables.tf
● @23-22411-027-max →~/Lab13 \$ cat variables.tf

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 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
```

J

- @23-22411-027-max →~/Lab13 \$ nano create-login-profile.sh
- @23-22411-027-max →~/Lab13 \$ cat create-login-profile.sh

```
#!/usr/bin/env bash
set -euo pipefail

USERNAME="$1"
PASSWORD="$2"

# Check if login profile already exists
if aws iam get-login-profile --user-name "$USERNAME" >/dev/null 2>&1; then
    echo "Login profile already exists for $USERNAME. Skipping."
else
    echo "Creating login profile for $USERNAME"
    aws iam create-login-profile \
        --user-name "$USERNAME" \
        --password "$PASSWORD" \
        --password-reset-required
fi
```

+

- @23-22411-027-max →~/Lab13 \$ chmod +x create-login-profile.sh
- @23-22411-027-max →~/Lab13 \$ ls -l create-login-profile.sh
- @23-22411-027-max →~/Lab13 \$

```
GNU nano 7.2                                     main.tf *
```

```
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"
}

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}'"
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

@23-22411-027-max → ~/Lab13 \$ cat main.tf

```
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"
}

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}'"
}
```

```
● @23-22411-027-max → ~/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_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260105152043021200000001]  
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260105152914045  
0000001]  
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260105152914049700000002]  
  
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated  
with the following symbols:  
+ create  
  
Terraform will perform the following actions:  
  
# null_resource.create_login_profile will be created  
+ resource "null_resource" "create_login_profile" {  
    + id      = (known after apply)  
    + triggers = {  
        + "password_hash" = (sensitive value)  
        + "user"          = "loadbalancer"  
    }  
}  
  
Plan: 1 to add, 0 to change, 0 to destroy.  
null_resource.create_login_profile: Creating...  
null_resource.create_login_profile: Provisioning with 'local-exec'...  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): (output suppressed due to sensitive value in config)  
null_resource.create_login_profile (local-exec): Creation complete after 4s [id=8177357808518286891]  
  
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

```
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 4s [id=8177357808518286891]
```

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

Outputs:

```
group_details = {
    "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
    "group_name" = "developers"
    "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
    "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
    "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
    "user_name" = "loadbalancer"
}
```

● @23-22411-027-max → ~/Lab13 \$

```
● @23-22411-027-max → ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-05T15:40:21+00:00",
        "PasswordResetRequired": true
    }
}
```

○ @23-22411-027-max → ~/Lab13 \$

aws

IAM user sign in ⓘ

Account ID or alias (Don't have?)
212208750468

Remember this account

IAM username
loadbalancer

Password

Show Password [Having trouble?](#)

Sign in

[Sign in using root user email](#)

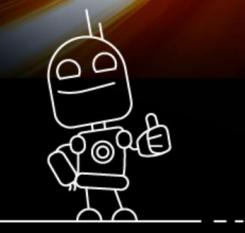
[Create a new AWS account](#)

By continuing, you agree to [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

Amazon Lightsail

Lightsail is the easiest way to get started on AWS

[Learn more »](#)



aws

You must change your password to continue

AWS account 212208750468

IAM user name loadbalancer

Old password

New password

Retype new password

Confirm password change

[Sign in using root user email](#)

English ▾

Terms of Use [Privacy Policy](#) © 1996-2026, Amazon Web Services, Inc. or its affiliates.



You must change your password to continue

AWS account 212208750468

IAM user name loadbalancer

Old password

New password

Retype new password

Confirm password change

[Sign in using root user email](#)

English ▾

[Terms of Use](#) [Privacy Policy](#) © 1996-2026, Amazon Web Services, Inc. or its affiliates.

Task 5

```
password_sha_hex = SHA256(var.iam_password)
user            = aws_iam_user.lb.name
}

depends_on = [aws_iam_user.lb]

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

output "access_key_id" {
  value = aws_iam_access_key.lb_access_key.id
}

output "access_key_secret" {
  value    = aws_iam_access_key.lb_access_key.secret
  sensitive = true
}
```

File **Edit** **Search** **Tools** **Help**

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line	M-E Redo

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - Lab13 + ▾ ⌂ ⌂ ⌂ ⌂ ⌂

```
@23-22411-027-max ~/Lab13 $ cat main.tf
  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"
}

resource "null_resource" "create_login_profile" {
  triggers = {
    password_hash = sha256(var.iam_password)
    user         = aws_iam_user.lb.name
  }

  depends_on = [aws_iam_user.lb]

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

output "access_key_id" {
  value = aws_iam_access_key.lb_access_key.id
}

output "access_key_secret" {
  value      = aws_iam_access_key.lb_access_key.secret
  sensitive = true
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - Lab13 + □ ✎ ... | ☰ >

```
@23-22411-027-max →~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"
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=AKIATC2FAR6CKJ7DEHOL]

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

Outputs:

access_key_id = "AKIATC2FAR6CKJ7DEHOL"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
  "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
  "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}

@23-22411-027-max →~/Lab13 $ terraform output
access_key_id = "AKIATC2FAR6CKJ7DEHOL"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
  "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
  "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}

@23-22411-027-max →~/Lab13 $
```

⚠ 0 ⚠ 0

```

}
● @23-22411-027-max → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
  "access_key_secret": {
    "value": [REDACTED],
    "type": "string",
    "sensitive": true
  },
  "group_details": {
    "value": {
      "group_arn": "arn:aws:iam::212208750468:group/groups/developers",
      "group_name": "developers",
      "unique_id": "AGPATC2FAR6CG3BAH6M2W"
    }
  },
○ @23-22411-027-max → ~/Lab13 $ █
0  (R) 0

```

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](#) ↗

AKIATC2FAR6CKJ7DEHOL		Actions ▾
Description	-	Status
Last used	None	Created
Last used region	N/A	Last used service

TASK6

Services

[Show more](#)

 **S3**
Scalable Storage in the Cloud ★

Top features

[Buckets](#) [Storage Lens dashboards](#) [Batch Operations](#) [S3 Express One Zone](#) [S3 Access Grants](#)

 **S3 Glacier** ★

Screenshot of the AWS S3 console showing the creation of a new bucket and its properties.

Buckets

Successfully created bucket "myapp-s3-bucket-lab". To upload files and folders, or to configure additional bucket settings, choose [View details](#).

General purpose buckets All AWS Regions **Directory buckets**

General purpose buckets (1) [Info](#)

Buckets are containers for data stored in S3.

Name	AWS Region	Creation date
myapp-s3-bucket-lab	Middle East (UAE) me-central-1	January 5, 2026, 23:37:13 (UTC+05:00)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Account snapshot [Info](#) Updated daily

Storage Lens provides visibility into storage usage and trends.

Amazon S3 > Buckets > myapp-s3-bucket-lab

myapp-s3-bucket-lab [Info](#)

Objects Metadata **Properties** Permissions Metrics Management Access P

Bucket overview

AWS Region Middle East (UAE) me-central-1

Amazon Resource Name (ARN) [arn:aws:s3:::myapp-s3-bucket-lab](#)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, recover, and easily recover from both unintended user actions and application failures. [Learn more ↗](#)

Bucket Versioning Enabled

Multi-factor authentication (MFA) delete An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting objects. [More ↗](#)

Disabled

```
GNU nano 7.2                               main.tf
terraform {
  backend "s3" {
    bucket      = "myapp-s3-bucket-lab"
    key         = "myapp/terraform.tfstate"
    region     = "me-central-1"
    encrypt     = 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
  }
}
```

```
@23-22411-027-max ~~/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 v6.27.0

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.

```
@23-22411-027-max ~~/Lab13 $ █
```

fnl ~

```
● @23-22411-027-max →~/Lab13 $ sudo nano main.tf
● @23-22411-027-max →~/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 v6.27.0

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.
● @23-22411-027-max →~/Lab13 $ sudo nano main.tf
○ @23-22411-027-max →~/Lab13 $
```

```

● @23-22411-027-max → ~/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=8177357808518286891]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIATC2FAR6CKJ7DEHOL]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260105152914049700000002]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-2026010515291404550000001]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260105152043021200000001]

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 = "AKIATC2FAR6CKJ7DEHOL"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::212208750468:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPATC2FAR6CG3BAH6M2W"
}
user_details = {
  "unique_id" = "AIDATC2FAR6CHE2SNGTGX"
  "user_arn" = "arn:aws:iam::212208750468:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}

```

```
○ @23-22411-027-max → ~/Lab13 $ 
```

The screenshot shows the AWS S3 console interface. On the left, there's a navigation sidebar with sections like 'Amazon S3', 'Buckets', 'Access management and security', 'Storage management and insights', and 'Account and organization settings'. The main area displays the details for the 'terraform.tfstate' object in the 'myapp-s3-bucket-lab' bucket. The 'Properties' tab is selected. Key details shown include:

- Object overview:**
 - Owner: c29b94ccf47e99549c61e430d7dc2b9b2f9f3a6ad70e996ad2f639783b76e960
 - AWS Region: Middle East (UAE) me-central-1
 - Last modified: January 6, 2026, 00:00:44 (UTC+05:00)
 - Size: 6.2 KB
 - Type: tfstate
 - Key: myapp/terraform.tfstate
- S3 URI:** s3://myapp-s3-bucket-lab/myapp/terraform.tfstate
- Amazon Resource Name (ARN):** arn:aws:s3:::myapp-s3-bucket-lab/myapp/terraform.tfstate
- Entity tag (Etag):** 99817d2772d90f67e0b2fd60f250478a
- Object URL:** https://myapp-s3-bucket-lab.s3.me-central-1.amazonaws.com/myapp/terraform.tfstate

```

}
● @23-22411-027-max → ~/Lab13 $ ls -la terraform.tfstate*
-rw-r--r-- 1 codespace codespace 0 Jan  5 19:00 terraform.tfstate
-rw-r--r-- 1 codespace codespace 6315 Jan  5 19:00 terraform.tfstate.backup
○ @23-22411-027-max → ~/Lab13 $ 
```

⚠ 0 ⚠ 0

```
● @23-22411-027-max → ~/Lab13 $ terraform destroy -auto-approve
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-2026010515291404000001]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-20260105152914049700000000]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260105152043021200000001]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIATC2FAR6CKJ7DEHOL]
null_resource.create_login_profile: Refreshing state... [id=8177357808518286891]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_iam_access_key.lb_access_key will be destroyed
- resource "aws_iam_access_key" "lb_access_key" {
    - create_date      = "2026-01-05T18:22:47Z" -> null
    - id              = "AKIATC2FAR6CKJ7DEHOL" -> null
    - secret          = (sensitive value) -> null
    - ses_smtp_password_v4 = (sensitive value) -> null
    - status          = "Active" -> null
    - user            = "loadbalancer" -> null
}

# aws_iam_group.developers will be destroyed
- resource "aws_iam_group" "developers" {
    - arn      = "arn:aws:iam::212208750468:group/groups/developers" -> null
    - id       = "developers" -> null
    - name     = "developers" -> null
    - path     = "/groups/" -> null
    - unique_id = "AGPATC2FAR6CG3BAH6M2W" -> null
}

# aws_iam_group_policy_attachment.change_password will be destroyed
- resource "aws_iam_group_policy_attachment" "change_password" {
    - group      = "developers" -> null
}
```

```
@23-22411-027-max → ~/Lab13 $ terraform destroy -auto-approve

# aws_iam_group_policy_attachment.change_password will be destroyed
- resource "aws_iam_group_policy_attachment" "change_password" {
    - group      = "developers" -> null
    - id         = "developers-20260105152914049700000002" -> null
    - policy_arn = "arn:aws:iam::aws:policy/IAMUserChangePassword" -> null
}

# aws_iam_group_policy_attachment.developer_ec2_fullaccess will be destroyed
- resource "aws_iam_group_policy_attachment" "developer_ec2_fullaccess" {
    - group      = "developers" -> null
    - id         = "developers-20260105152914045500000001" -> null
    - policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess" -> null
}

# aws_iam_user.lb will be destroyed
- resource "aws_iam_user" "lb" {
    - arn          = "arn:aws:iam::212208750468:user/users/loadbalancer" ->
    - force_destroy = true -> null
    - id          = "loadbalancer" -> null
    - name        = "loadbalancer" -> null
    - path        = "/users/" -> null
    - tags        = {
        - "DisplayName" = "Load Balancer"
    } -> null
    - tags_all     = {
        - "DisplayName" = "Load Balancer"
    } -> null
    - unique_id    = "AIDATC2FAR6CHE2SNGTGX" -> null
}

# aws_iam_user_group_membership.lb_membership will be destroyed
- resource "aws_iam_user_group_membership" "lb_membership" {
    - groups = [
        - "developers",
    ] -> null
    - id      = "terraform-20260105152043021200000001" -> null
    - user    = "loadbalancer" -> null
}
```

```
@23-22411-027-max → ~/Lab13 $ terraform destroy -auto-approve
  - triggers = {
    - "password_hash" = (sensitive value)
    - "user"          = "loadbalancer"
  } -> null
}

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

Changes to Outputs:
- access_key_id      = "AKIATC2FAR6CKJ7DEHOL" -> null
- access_key_secret = (sensitive value) -> null
- group_details     = {
  - group_arn   = "arn:aws:iam::212208750468:group/groups/developers"
  - group_name  = "developers"
  - unique_id   = "AGPATC2FAR6CG3BAH6M2W"
} -> null
- user_details       = {
  - unique_id  = "AIDATC2FAR6CHE2SNGTGX"
  - user_arn   = "arn:aws:iam::212208750468:user/users/loadbalancer"
  - user_name  = "loadbalancer"
} -> null
null_resource.create_login_profile: Destroying... [id=8177357808518286891]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260105152914045500000001]
1]
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-20260105152043021200000001]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260105152914049700000002]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIATC2FAR6CKJ7DEHOL]
aws_iam_user_group_membership.lb_membership: Destruction complete after 1s
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_access_key.lb_access_key: Destruction complete after 1s
aws_iam_user.lb: Destroying... [id=loadbalancer]
aws_iam_group.developers: Destruction complete after 0s
aws_iam_user.lb: Destruction complete after 3s

Destroy complete! Resources: 7 destroyed.
```

○ @23-22411-027-max → ~/Lab13 \$

```
aws_iam_user.lb: Destruction complete after 3s
```

```
Destroy complete! Resources: 7 destroyed.
```

● @23-22411-027-max → ~/Lab13 \$ terraform destroy

```
No changes. No objects need to be destroyed.
```

```
Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.
```

```
Destroy complete! Resources: 0 destroyed.
```

● @23-22411-027-max → ~/Lab13 \$ terraform state list

○ @23-22411-027-max → ~/Lab13 \$

The screenshot shows the AWS IAM 'Users' page. The left sidebar has a 'Search IAM' bar and a tree view with nodes like 'Identity and Access Management (IAM)', 'Access management', 'Users', 'Policies', 'Identity providers', 'Account settings', 'Root access management', and 'Temporary delegation requests'. The main area is titled 'Users (2) Info' with a note: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' It includes a search bar and a table with columns: User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. Two users are listed: 'Admin' (last active 8 minutes ago) and 'Admin1' (last active 6 days ago). There are buttons for 'Delete' and 'Create user'.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
Admin	/	0	8 minutes ago	-	-	-
Admin1	/	0	6 days ago	-	7 days	-

Task 7

The screenshot shows a terminal window in VS Code with the title '[Preview] README.md'. The terminal tab is selected, showing the command 'nano'. The file 'locals.tf' is open, containing the following code:

```
GNU nano 7.2          locals.tf *
locals {
    users = csvdecode(file("${path.module}/users.csv"))
}
```

TERM

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

GNU nano 7.2 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
```

```
● @23-22411-027-max → /workspaces/LAB13 (main) $ terraform -version
Terraform v1.6.6
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
● @23-22411-027-max → /workspaces/LAB13 (main) $ aws --version
aws-cli/2.32.28 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
● @23-22411-027-max → /workspaces/LAB13 (main) $ nano locals.tf
● @23-22411-027-max → /workspaces/LAB13 (main) $ nano users.csv
● @23-22411-027-max → /workspaces/LAB13 (main) $ nano main.tf
● @23-22411-027-max → /workspaces/LAB13 (main) $ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/null...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
- Installing hashicorp/null v3.2.4...
- Installed hashicorp/null v3.2.4 (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.

```
○ @23-22411-027-max → /workspaces/LAB13 (main) $ █
```

```

@23-22411-027-max → /workspaces/LAB13 (main) $ ls -la
total 86152
drwxrwxrwx+ 5 codespace root          4096 Jan  6 12:12 .
drwxr-xrwx+ 5 codespace root          4096 Jan  5 14:29 ..
drwxrwxrwx+ 8 codespace root          4096 Jan  5 19:22 git
-rw-rw-rw-  1 codespace codespace    1024 Jan  6 12:12 .main.tf.swp
drwxr-xr-x+ 3 codespace codespace    4096 Jan  6 12:03 .terraform
-rw-r--r--  1 codespace codespace    2422 Jan  6 12:03 .terraform.lock.hcl
-rw-rw-rw-  1 codespace root          7 Jan   5 14:29 README.md
drwxr-xr-x+ 3 codespace codespace    4096 Jan  2 23:18 aws
-rw-rw-rw-  1 codespace codespace 63189473 Jan  5 14:33 awscliv2.zip
-rw-rw-rw-  1 codespace codespace    66 Jan   6 12:00 locals.tf
-rw-rw-rw-  1 codespace codespace   2131 Jan  6 13:21 main.tf
-rw-rw-rw-  1 codespace codespace 24976120 Jan  5 14:37 terraform.zip
-rw-rw-rw-  1 codespace codespace   167 Jan   6 12:01 users.csv
-rw-rw-rw-  1 codespace codespace   121 Jan   6 12:05 variables.tf
@23-22411-027-max → /workspaces/LAB13 (main) $ ^C
@23-22411-027-max → /workspaces/LAB13 (main) $ nano create-login-profile.sh
@23-22411-027-max → /workspaces/LAB13 (main) $ chmod +x create-login-profile.sh
@23-22411-027-max → /workspaces/LAB13 (main) $ terraform validate

```

```

3-22411-027-max → /workspaces/LAB13 (main) $ terraform apply -auto-approve -var="iam_password=MySecureP@ssw0rd"

```

s iam_access_key.users_access_keys["Michael"]: Refreshing state... [id=AKIATC2FAR6CNVFOCBSA]
s iam_access_key.users_access_keys["Stanley"]: Refreshing state... [id=AKIATC2FAR6CK4B2Y3EE]
s iam_access_key.users_access_keys["Jan"]: Refreshing state... [id=AKIATC2FAR6CFAJPWD6G]
s iam_access_key.users_access_keys["Toby"]: Refreshing state... [id=AKIATC2FAR6CKARUMVWJ]
s iam_access_key.users_access_keys["Ryan"]: Refreshing state... [id=AKIATC2FAR6CHQQGADRN]
s iam_access_key.users_access_keys["Phyllis"]: Refreshing state... [id=AKIATC2FAR6CD0U0FDJ5]
s iam_access_key.users_access_keys["Peter"]: Refreshing state... [id=AKIATC2FAR6CLIBZIK3B]
s iam_access_key.users_access_keys["Angela"]: Refreshing state... [id=AKIATC2FAR6CFAXQENBX]
s iam_access_key.users_access_keys["Gabe"]: Refreshing state... [id=AKIATC2FAR6CNP7EJ3SB]
s iam_access_key.users_access_keys["Jim"]: Refreshing state... [id=AKIATC2FAR6CJIYDBFUG]
s iam_access_key.users_access_keys["Creed"]: Refreshing state... [id=AKIATC2FAR6CGGDRYUH5]
s iam_access_key.users_access_keys["Pam"]: Refreshing state... [id=AKIATC2FAR6CDLM6S66V]
s iam_access_key.users_access_keys["Kevin"]: Refreshing state... [id=AKIATC2FAR6CAVJFR4WJ]
s iam_user_group_membership.users_membership["David"]: Refreshing state... [id=terraform-202601061327019]
s iam_user_group_membership.users_membership["Rvan"]: Refreshing state... [id=terraform-20260106132723]

```
MODELLVIS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  DASH  E
```

```
}
```

```
"Robert" = {
```

```
    "access_key_id" = "AKIATC2FAR6CC737DY02"
```

```
    "user_arn" = "arn:aws:iam::212208750468:user/users/Robert"
```

```
    "user_unique_id" = "AIDATC2FAR6CAMKAWYK7Q"
```

```
}
```

```
"Ryan" = {
```

```
    "access_key_id" = "AKIATC2FAR6CHQQGADRN"
```

```
    "user_arn" = "arn:aws:iam::212208750468:user/users/Ryan"
```

```
    "user_unique_id" = "AIDATC2FAR6CMU62AYSS3"
```

```
}
```

```
"Stanley" = {
```

```
    "access_key_id" = "AKIATC2FAR6CK4B2Y3EE"
```

```
    "user_arn" = "arn:aws:iam::212208750468:user/users/Stanley"
```

```
    "user_unique_id" = "AIDATC2FAR6CPBX3BL3RR"
```

```
}
```

```
"Toby" = {
```

```
    "access_key_id" = "AKIATC2FAR6CKARUMVWJ"
```

```
    "user_arn" = "arn:aws:iam::212208750468:user/users/Toby"
```

```
    "user_unique_id" = "AIDATC2FAR6CLGFZDJNYD"
```

```
}
```

```
}
```

```
@23-22411-027-max → /workspaces/LAB13 (main) $
```

```
@23-22411-027-max → /workspaces/LAB13 (main) $
```

```
@23-22411-027-max → /workspaces/LAB13 (main) $ terraform output
}
"Robert" = {
    "access_key_id" = "AKIATC2FAR6CC737DY02"
    "user_arn" = "arn:aws:iam::212208750468:user/users/Robert"
    "user_unique_id" = "AIDATC2FAR6CAMKAWYK7Q"
}
"Ryan" = {
    "access_key_id" = "AKIATC2FAR6CHQQGADRN"
    "user_arn" = "arn:aws:iam::212208750468:user/users/Ryan"
    "user_unique_id" = "AIDATC2FAR6CMU62AYSS3"
}
"Stanley" = {
    "access_key_id" = "AKIATC2FAR6CK4B2Y3EE"
    "user_arn" = "arn:aws:iam::212208750468:user/users/Stanley"
    "user_unique_id" = "AIDATC2FAR6CPBX3BL3RR"
}
"Toby" = {
    "access_key_id" = "AKIATC2FAR6CKARUMVWJ"
    "user_arn" = "arn:aws:iam::212208750468:user/users/Toby"
    "user_unique_id" = "AIDATC2FAR6CLGFZDJNYD"
}
}

@23-22411-027-max → /workspaces/LAB13 (main) $ █
(Δ) 0
```

```
▶ @23-22411-027-max → /workspaces/LAB13 (main) $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
"all_access_key_secrets": {
    "value": {
        "Andy": "",
        "Angela": "",
        "Charles": "",
        "Clark": ""
```

Users (28) Info

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

Search

User name ▲ | Path ▽ | Groups ▽ | Last activity ▽ | MFA ▽ | Password age ▽ | Console last sign-in ▽

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
Andy	/users/	1	-	-	⌚ 6 minutes	-
Angela	/users/	1	-	-	⌚ 5 minutes	-
Charles	/users/	1	-	-	⌚ 6 minutes	-
Clark	/users/	1	-	-	⌚ 6 minutes	-
Creed	/users/	1	-	-	⌚ 5 minutes	-
Darryl	/users/	1	-	-	⌚ 5 minutes	-
David	/users/	1	-	-	⌚ 5 minutes	-
Dwight	/users/	1	-	-	⌚ 5 minutes	-
Erin	/users/	1	-	-	⌚ 6 minutes	-
Gabe	/users/	1	-	-	⌚ 6 minutes	-
Holly	/users/	1	-	-	⌚ 6 minutes	-
Jan	/users/	1	-	-	⌚ 5 minutes	-
Jim	/users/	1	-	-	⌚ 5 minutes	-
Jo	/users/	1	-	-	⌚ 5 minutes	-

IAM > Users

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](#)

Access reports

- Access Analyzer
- Resource analysis [New](#)
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies

Users (28) Info

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

Search

User name ▲ | Path ▽ | Groups ▽ | Last activity ▽ | MFA ▽ | Password age ▽ | Console last sign-in ▽

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
Darryl	/users/	1	-	-	⌚ 5 minutes	-
David	/users/	1	-	-	⌚ 5 minutes	-
Dwight	/users/	1	-	-	⌚ 5 minutes	-
Erin	/users/	1	-	-	⌚ 6 minutes	-
Gabe	/users/	1	-	-	⌚ 6 minutes	-
Holly	/users/	1	-	-	⌚ 6 minutes	-
Jan	/users/	1	-	-	⌚ 5 minutes	-
Jim	/users/	1	-	-	⌚ 5 minutes	-
Jo	/users/	1	-	-	⌚ 5 minutes	-
Kelly	/users/	1	-	-	⌚ 5 minutes	-
Kevin	/users/	1	-	-	⌚ 5 minutes	-
Meredith	/users/	1	-	-	⌚ 6 minutes	-
Michael	/users/	1	-	-	⌚ 5 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

Access reports

- Access Analyzer
- Resource analysis New
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies

developers Info

Summary

User group name: developers

Creation time: January 06, 2026, 18:26 (UTC+05:00)

ARN: arn:aws:iam::212208750468:group/groups/developers

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.

User name	Groups	Last sign-in
Andy	No groups	No last sign-in
Angela	No groups	No last sign-in
Charles	No groups	No last sign-in
Clark	No groups	No last sign-in
Creed	No groups	No last sign-in
Darryl	No groups	No last sign-in
David	No groups	No last sign-in
Charles	No groups	No last sign-in
Clark	No groups	No last sign-in
Creed	No groups	No last sign-in
Darryl	No groups	No last sign-in
Dwight	No groups	No last sign-in
Erin	No groups	No last sign-in
Gabe	No groups	No last sign-in
Holly	No groups	No last sign-in
Jan	No groups	No last sign-in
Jim	No groups	No last sign-in
Jo	No groups	No last sign-in
Kelly	No groups	No last sign-in
Kevin	No groups	No last sign-in
Meredith	No groups	No last sign-in
Michael	No groups	No last sign-in
Oscar	1	No last sign-in
Pam	1	No last sign-in

Add users

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

Access reports

- Access Analyzer
- Resource analysis New
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies

developers

User name	Groups	Last sign-in				
Charles	No groups	No last sign-in				
Clark	No groups	No last sign-in				
Creed	No groups	No last sign-in				
Darryl	No groups	No last sign-in				
Dwight	No groups	No last sign-in				
Erin	No groups	No last sign-in				
Gabe	No groups	No last sign-in				
Holly	No groups	No last sign-in				
Jan	No groups	No last sign-in				
Jim	No groups	No last sign-in				
Jo	No groups	No last sign-in				
Kelly	No groups	No last sign-in				
Kevin	No groups	No last sign-in				
Meredith	No groups	No last sign-in				
Michael	No groups	No last sign-in				
Oscar	1	No last sign-in				
Pam	1	No last sign-in				
Kevin	/users/	1	-	-	9 minutes	-
Meredith	/users/	1	-	-	10 minutes	-
Michael	/users/	1	-	-	9 minutes	-

Identity and Access Management (IAM)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			
Assign MFA device			

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

AKIATC2FAR6CNVFOCBSA		Actions ▾	
Description	-	Status	Active
Last used	None	Created	8 minutes ago
Last used region	N/A	Last used service	N/A

API keys for Amazon Bedrock (0)

Use API keys for Amazon Bedrock to integrate into your library of choice and make API requests programmatically. You can have a maximum of two long-term API keys (active, inactive, or expired) at a time. [Learn more](#)

Actions ▾		Generate API Key	
-----------	--	------------------	--

terraform.tfstate [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions ▾](#)

Properties **Permissions** **Versions**

Object overview

Owner	c29b94ccf47e99549c61e430d7dc2b9b2f9f3a6ad70e996ad2f639783b76e960
AWS Region	Middle East (UAE) me-central-1
Last modified	January 6, 2026, 00:07:26 (UTC+05:00)
Size	180.0 B
Type	tfstate
Key	myapp/terraform.tfstate

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Bucket Versioning

When enabled, multiple variants of an object can be stored in the bucket to easily recover from

Management configurations

Replication status

When a replication rule is applied to an object the replication status indicates the progress of the

CLEANUP:

```
aws_iam_user.users["Meredith"]: Destruction complete after 2s
aws_iam_user.users["Andy"]: Destroying... [id=Andy]
aws_iam_user.users["Pam"]: Destroying... [id=Pam]
aws_iam_user.users["Kelly"]: Destruction complete after 2s
aws_iam_user.users["Jim"]: Destroying... [id=Jim]
aws_iam_user.users["Dwight"]: Destruction complete after 2s
aws_iam_user.users["Angela"]: Destroying... [id=Angela]
aws_iam_user.users["Holly"]: Destruction complete after 2s
aws_iam_user.users["Darryl"]: Destroying... [id=Darryl]
aws_iam_user.users["Michael"]: Destruction complete after 2s
aws_iam_user.users["Oscar"]: Destroying... [id=Oscar]
aws_iam_user.users["Ryan"]: Destruction complete after 2s
aws_iam_user.users["Charles"]: Destruction complete after 2s
aws_iam_user.users["Phyllis"]: Destruction complete after 4s
aws_iam_user.users["Pam"]: Destruction complete after 2s
aws_iam_user.users["Andy"]: Destruction complete after 2s
aws_iam_user.users["Angela"]: Destruction complete after 3s
aws_iam_user.users["Darryl"]: Destruction complete after 3s
aws_iam_user.users["Oscar"]: Destruction complete after 3s
aws_iam_user.users["Jim"]: Destruction complete after 4s
aws_iam_user.users["Stanley"]: Destruction complete after 4s
```

Destroy complete! Resources: 105 destroyed.

@23-22411-027-max → /workspaces/LAB13 (main) \$ █

Users (2) Info

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

[Delete](#)[Create](#) Search

<input type="checkbox"/>	User name	▲ Path	▼ Group:	▼ Last activity	▼ MFA	▼ Password age	▼ Console last sign-
<input type="checkbox"/>	Admin	/	0	14 minutes ago	-	-	-
<input type="checkbox"/>	Admin1	/	0	6 days ago	-	7 days	-

User groups (1) Info

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

 Search

<input type="checkbox"/>	Group name	▲ Users	▼ Permissions	▼ Creation t
<input type="checkbox"/>	MyGroupCli		⚠ 0 Defined	2 days ago

General purpose buckets (1)

[All AWS Regions](#)

Directory buckets

General purpose buckets (1) Info

[Copy ARN](#)[Empty](#)[Delete](#)[Create bucket](#)

Buckets are containers for data stored in S3.

 Find buckets by name< 1 >

<input type="checkbox"/>	Name	▲ AWS Region	▼ Creation date
<input type="checkbox"/>	myapp-s3-bucket-lab	Middle East (UAE) me-central-1	January 5, 2026, 23:37:13 (UTC+05:00)