

task0_codespace_create_and_list.png

```
error fetching codespace information: HTTP 404: Not Found (https://api.github.com/user/codespaces/super-duper-tribble-97g5...)
PS C:\Users\hp> gh codespace list
NAME          DISPLAY NAME      REPOSITORY      BRANCH STATE    CREATED AT
super-duper-space-system-pj5... super-duper space sy... arshiajadoon/LAB_9   main* Shutdown about 18 days ago
upgraded-system-vpg6q5747rq3... upgraded system     arshiajadoon/LAB_9   main  Shutdown about 18 days ago
solid-space-waffle-jjp7596vx... solid space waffle  arshiajadoon/CC_014_... main  Shutdown about 17 days ago
shiny-winner-q754q6rjw9qwhxp9 shiny winner       arshiajadoon/CC_014_... main  Shutdown about 16 days ago
upgraded-eureka-7v159wq6g5rv... upgraded eureka    arshiajadoon/CC_Arsh... main  Shutdown about 9 days ago
literate-space-fishstick-q75... literate space fishs... arshiajadoon/CC_Arsh... main* Shutdown about 9 days ago
super-duper-tribble-97g59wvr... super-duper tribble  arshiajadoon/CC_ARSH... main  Available about 1 minute ago
glowing-space-giggle-7vj59wq... glowing space giggle arshiajadoon/CC_ARSH... main  Available about 1 minute ago
PS C:\Users\hp> gh codespace delete -c super-duper-tribble-97g5...
```

task0_codespace_ssh_connected.png

```
@arshiajadoon → /workspaces/CC_ARSHIA_014_LAB13 (main) $ |
```

task1_project_directory.png

```
@arshiajadoon → ~/Lab13 $ touch main.tf
ls -la
total 12
drwxrwxr-x 2 codespace codespace 4096 Jan 16 14:44 .
drwxr-x--- 1 codespace codespace 4096 Jan 16 14:42 ..
-rw-rw-r-- 1 codespace codespace    0 Jan 16 14:44 main.tf
@arshiajadoon → ~/Lab13 $ |
```

task1_main_tf.png

```
EOF
@arshiajadoon → ~/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
  }
}
@arshiajadoon → ~/Lab13 $ |
```

task1_terraform_init.png

```
@arshiajadoon → ~/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.
@arshiajadoon → ~/Lab13 $ |
```

task1_terraform_apply.png-

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

Outputs:

```
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
@arshiajadoon → ~/Lab13 $ |
```

task1_terraform_output.png

```
@arshiajadoon → ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
@arshiajadoon → ~/Lab13 $ |
```

task1_aws_console_group.png

Group name	Users	Permissions	Creation time
developers	△ 0	Not defined	4 minutes ago

task2_main_tf_user.png

```

@arshiajadoon → ~/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"
  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
  }
}
@arshiajadoon → ~/Lab13 $ |

```

task2_terraform_apply.png

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

Outputs:

```
group_details = {  
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"  
  "group_name" = "developers"  
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"  
}  
user_details = {  
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"  
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"  
  "user_name" = "loadbalancer"  
}  
@arshiajadoon → ~/Lab13 $ |
```

task2_terraform_output.png

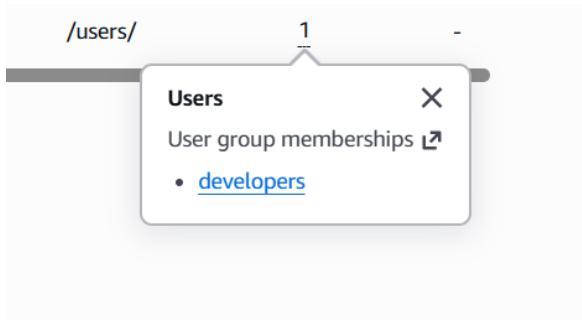
Outputs:

```
group_details = {  
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"  
  "group_name" = "developers"  
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"  
}  
user_details = {  
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"  
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"  
  "user_name" = "loadbalancer"  
}  
@arshiajadoon → ~/Lab13 $ terraform output  
group_details = {  
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"  
  "group_name" = "developers"  
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"  
}  
user_details = {  
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"  
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"  
  "user_name" = "loadbalancer"  
}  
@arshiajadoon → ~/Lab13 $ |
```

task2_aws_console_user.png

The screenshot shows the AWS IAM User Details page for a user named 'loadbalancer'. The left sidebar includes navigation links for IAM, Dashboard, Access Management (User groups, Roles, Policies, Identity providers, Account settings, Root access management, Temporary delegation requests), and Access reports (Access Analyzer, Resource analysis, Unused access). The main panel displays the 'Summary' tab for the 'loadbalancer' user, showing ARN (arn:aws:iam::353695163380:user/users/loadbalancer), Console access (Disabled), and Last console sign-in (January 16, 2026, 20:08 (UTC+05:00)). It also shows an 'Access key 1' with a 'Create access key' button. Below the summary is a 'Permissions' tab, which lists 'Permissions policies (0)' and indicates 'No resources to display'. A search bar and filter dropdown are present at the top of the permissions table.

task2_aws_console_user_groups.png



task3_main_tf_policies.png

```
    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"
}
```

task3_terraform_apply.png

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

Outputs:

```
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $ |
```

task3_aws_console_policies.png

The screenshot shows the AWS IAM Policies page. At the top, there are tabs for 'Users (1)', 'Permissions' (which is selected), and 'Access Advisor'. Below the tabs, it says 'Permissions policies (2) Info'. It states that you can attach up to 10 managed policies. There is a search bar and a 'Filter by Type' dropdown set to 'All types'. A table lists two policies: 'AmazonEC2FullAccess' and 'IAMUserChangePassword', both of which are AWS managed policies and are attached to one entity.

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

task4_variables_tf.png

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Outputs:
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developer"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $ cat > variables.tf << 'EOF'
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "IdontKnow"
}
EOF
@arshiajadoon → ~/Lab13 $ cat variables.tf
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "IdontKnow"
}
@arshiajadoon → ~/Lab13 $ |
```

task4_create_login_script.png

```
# 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
EOF
@arshiajadoon → ~/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
@arshiajadoon → ~/Lab13 $ |
```

task4_chmod_script.png

```
@arshiajadoon → ~/Lab13 $ chmod +x create-login-profile.sh
ls -la create-login-profile.sh
-rwxrwxr-x 1 codespace codespace 440 Jan 16 15:41 create-login-profile.sh
@arshiajadoon → ~/Lab13 $ |
```

task4_main_tf_login_profile.png

```
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}'"
    }
}
@arshiajadoon → ~/Lab13 $ |
```

task4_terraform_apply.png

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

Outputs:

```
group_details = {
    "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
    "group_name" = "developers"
    "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
    "unique_id" = "AIDAVEW074P2G36QCQ2EM"
    "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
    "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $ |
```

task4_aws_cli_verify.png

```
@arshiajadoon → ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-16T15:46:42+00:00",
        "PasswordResetRequired": true
    }
}
@arshiajadoon → ~/Lab13 $ |
```

task5_main_tf_access_keys.png

```
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
}
@arshiajadoon → ~/Lab13 $ |
```

task5_terraform_apply.png

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

Outputs:

access_key_id = "AKIAVEW074P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $
```

task5_terraform_output.png

```
@arshiajadoon → ~/Lab13 $ terraform output
access_key_id = "AKIAVEW074P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $ |
```

task5_tfstate_secret.Png

```
@arshiajadoon → ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
"access_key_secret": {
  "value": "G+FVmEYmKeND0d9VGRvMo/eOl0AU8p5rLJAhgRE7",
  "type": "string",
  "sensitive": true
},
"group_details": {
  "value": {
    "group_arn": "arn:aws:iam::353695163380:group/groups/developers",
    "group_name": "developers",
    "unique_id": "AGPAVEW074P2IKMTVRUZN"
  }
}
@arshiajadoon → ~/Lab13 $ |
```

task5_aws_console_access_keys.png

The screenshot shows the AWS IAM User details page for a user named 'loadbalancer'. The 'Security credentials' tab is selected. Key details include:

- ARN:** arn:aws:iam::353695163380:user/users/loadbalancer
- Console access:** Enabled without MFA
- Created:** January 16, 2026, 20:08 (UTC+05:00)
- Last console sign-in:** Never
- Access key 1:** AKIAVEWO74P2CRRANLMB - Active (Never used, created today)
- Access key 2:** Create access key

The 'Console sign-in' section shows a link to the AWS Management Console sign-in page. The 'Multi-factor authentication (MFA)' section indicates 0 MFA devices assigned.

task6_s3_bucket_create.png

The screenshot shows the AWS S3 Buckets page. A green success message at the top states: "Successfully created bucket 'myapp-s3-bucket-demo-arslia014'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, the 'General purpose buckets' tab is selected, showing one bucket named 'myapp-s3-bucket-demo-arslia014' in the US East (N. Virginia) region, created on January 16, 2026, at 21:20:31 (UTC+05:00). Other tabs include 'All AWS Regions' and 'Directory buckets'. On the right, there are sections for 'Account snapshot' (updated daily), 'Storage Lens' (provides visibility into storage usage and activity trends), and 'External access summary' (updated daily).

task6_s3_bucket_versioning.png

This screenshot is identical to the previous one, showing the AWS S3 Buckets page with the same bucket creation details and sidebar features. The 'Buckets are containers for data stored in S3.' message is visible at the top left.

task6_main_tf_backend.png

```
Windows PowerShell x + ^

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
}

@arshiajadoon → ~/Lab13 $ |
```

task6_terraform_init_migrate.png

```
@arshiajadoon → ~/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/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.
@arshiajadoon → ~/Lab13 $ |
```

task6_terraform_apply.png

```

aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIAVEW074P2CRRANLMB]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=ter]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration
needed.

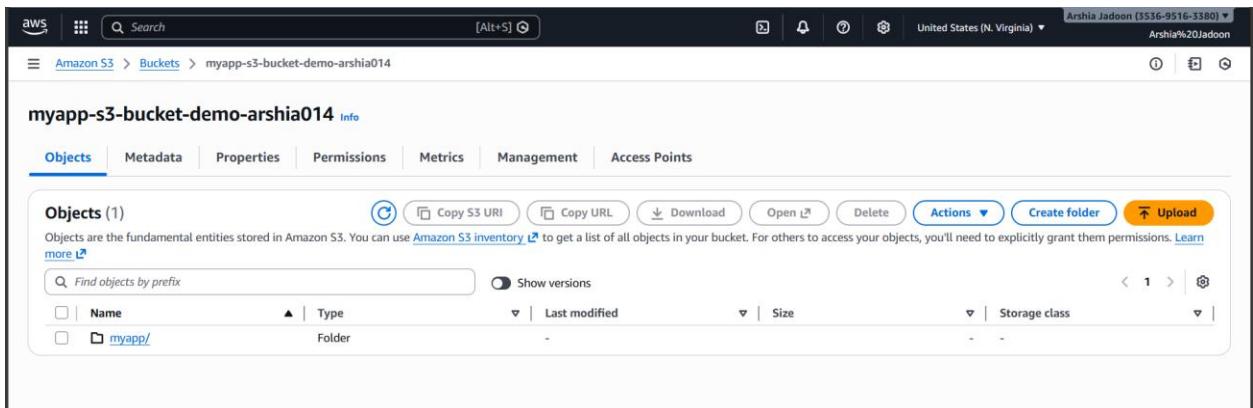
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.

Outputs:

access_key_id = "AKIAVEW074P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEW074P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon → ~/Lab13 $ |

```

task6_s3_tfstate_file.png



task6_local_state_backup.png

```

@arshiajadoon → ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace    0 Jan 16 16:31 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 16 16:31 terraform.tfstate.backup
@arshiajadoon → ~/Lab13 $ |

```

task6_terraform_destroy.png

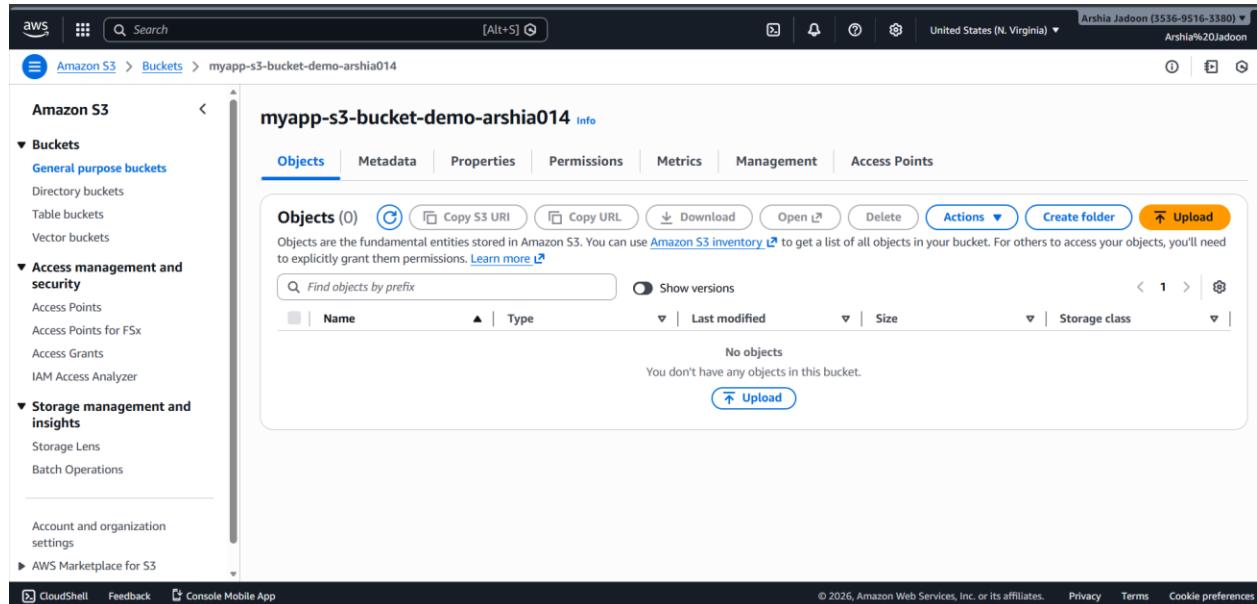
```

null_resource.create_login_profile: Destroying... [id=6647801183677904328]
aws_iam_user_group_membership.lb_membership: Destruction complete after 0s
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260116150819547000000001]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIAVEW074P2CRANLMB]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260116153826172600000002]
aws_iam_user_group_membership.lb_membership: Destruction complete after 1s
aws_iam_access_key.lb_access_key: Destruction complete after 1s
aws_iam_user.lb: Destroying... [id=loadbalancer]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 1s
aws_iam_group_policy_attachment.change_password: Destruction complete after 1s
aws_iam_group.developers: Destroying... [id=developers]
aws_iam_group.developers: Destruction complete after 0s
aws_iam_user.lb: Destruction complete after 3s

Destroy complete! Resources: 7 destroyed.
@arshiajadoon ~/Lab13 $ |

```

task6_s3_tfstate_destroyed. Png



task7_locals_tf.png

```

@arshiajadoon ~/Lab13 $ cat > locals.tf << 'EOF'
locals {
    users = csvdecode(file("users.csv"))
}
EOF

cat locals.tf
locals {
    users = csvdecode(file("users.csv"))
}
@arshiajadoon ~/Lab13 $ |

```

task7_users_csv.png

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

task7_main_tf_multiple_users.png

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

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

  user = each.value.name
}

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

# Output all access key secrets (sensitive)
output "all_access_key_secrets" {
  value = {
    for user_name, key in aws_iam_access_key.users_access_keys : user_name => key.secret
  }
  sensitive = true
}
@arshiajadoon → ~/Lab13 $ |
```

task7_terraform_init.png

```

@arshiajadoon ~~/Lab13 $ terraform init -reconfigure
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/null from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0
- 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.
@arshiajadoon ~~/Lab13 $ |

```

task7_terraform_apply.png

```

    "user_arn" = "arn:aws:iam::353695163380:user/users/Robert"
    "user_unique_id" = "AIDAVEW074P2EFOXUTY6I"
}
"Ryan" = {
    "access_key_id" = "AKIAVEW074P2GILD7SOY"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Ryan"
    "user_unique_id" = "AIDAVEW074P2DAOZHB1J4"
}
"Stanley" = {
    "access_key_id" = "AKIAVEW074P2E7YFYIN3"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Stanley"
    "user_unique_id" = "AIDAVEW074P2FESFZ3XX4"
}
"Toby" = {
    "access_key_id" = "AKIAVEW074P2AXHLQEXW"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Toby"
    "user_unique_id" = "AIDAVEW074P2M650GBUH6"
}
}
group_details = {
    "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
    "group_name" = "developers"
    "unique_id" = "AGPAVEW074P2NJUGZCFUY"
}
@arshiajadoon ~~/Lab13 $ |

```

task7_terraform_output.png

```

@arshiajadoon ~~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
    "Andy" = {
        "access_key_id" = "AKIAVEW074P20AW646KU"
        "user_arn" = "arn:aws:iam::353695163380:user/users/Andy"
        "user_unique_id" = "AIDAVEW074P2G5IOIR4IU"
    }
    "Angela" = {
        "access_key_id" = "AKIAVEW074P2M6DNBVSS"
        "user_arn" = "arn:aws:iam::353695163380:user/users/Angela"
        "user_unique_id" = "AIDAVEW074P2OBHOVDBH7"
    }
    "Charles" = {
        "access_key_id" = "AKIAVEW074P20023Q70R"
        "user_arn" = "arn:aws:iam::353695163380:user/users/Charles"
        "user_unique_id" = "AIDAVEW074P2DDVEIJL6Z"
    }
    "Clark" = {
        "access_key_id" = "AKIAVEW074P2F43SAWUQ"
        "user_arn" = "arn:aws:iam::353695163380:user/users/Clark"
        "user_unique_id" = "AIDAVEW074P2HMNNFKNIN"
    }
    "Creed" = {
        "access_key_id" = "AKIAVEW074P2MUZEFUGC"
        "user_arn" = "arn:aws:iam::353695163380:user/users/Creed"
    }
}

```

```

    }
  "Charles" = {
    "access_key_id" = "AKIAVEW074P20023Q70R"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Charles"
    "user_unique_id" = "AIDAVEW074P2DDVEIJL6Z"
  }
  "Clark" = {
    "access_key_id" = "AKIAVEW074P2F43SAWUQ"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Clark"
    "user_unique_id" = "AIDAVEW074P2HMNNFKNIN"
  }
  "Creed" = {
    "access_key_id" = "AKIAVEW074P2MUZEFUGC"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Creed"
    "user_unique_id" = "AIDAVEW074P2EHCQAZDSD"
  }
  "Darryl" = {
    "access_key_id" = "AKIAVEW074P2HSBIBISJ"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Darryl"
    "user_unique_id" = "AIDAVEW074P2HYWQEWWS"
  }
  "David" = {
    "access_key_id" = "AKIAVEW074P2NIZYRJP6"
    "user_arn" = "arn:aws:iam::353695163380:user/users/David"
    "user_unique_id" = "AIDAVEW074P2AWFFS2ANV"
  }

  "user_arn" = "arn:aws:iam::353695163380:user/users/Phyllis"
  "user_unique_id" = "AIDAVEW074P2EQXORKJIG"
}
"Robert" = {
  "access_key_id" = "AKIAVEW074P2PQQGQ54T"
  "user_arn" = "arn:aws:iam::353695163380:user/users/Robert"
  "user_unique_id" = "AIDAVEW074P2EF0XUTY6I"
}
"Ryan" = {
  "access_key_id" = "AKIAVEW074P2GILD7SOY"
  "user_arn" = "arn:aws:iam::353695163380:user/users/Ryan"
  "user_unique_id" = "AIDAVEW074P2DAOZHBIJ4"
}
"Stanley" = {
  "access_key_id" = "AKIAVEW074P2E7YFYIN3"
  "user_arn" = "arn:aws:iam::353695163380:user/users/Stanley"
  "user_unique_id" = "AIDAVEW074P2FESFZ3XX4"
}
"Toby" = {
  "access_key_id" = "AKIAVEW074P2AXHLQEXW"
  "user_arn" = "arn:aws:iam::353695163380:user/users/Toby"
  "user_unique_id" = "AIDAVEW074P2M650GBUH6"
}

group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEW074P2NJUGZCFUY"
}
@arshiajadoon → ~/Lab13 $ |

```

task7_tfstate_secrets. Png

```

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

```

task7_aws_console_all_users. Png

<input type="checkbox"/>	User name	Path	Groups	Last activity	MFA	Password age
<input type="checkbox"/>	Andy	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Angela	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Charles	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Clark	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Creed	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Darryl	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	David	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Dwight	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Erin	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Gabe	/users/	1	-	-	⌚ 3 minutes
<hr/>						
<input type="checkbox"/>	Holly	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Jan	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Jim	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Jo	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Kelly	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Kevin	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Meredith	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Michael	/users/	1	-	-	⌚ 2 minutes
<input type="checkbox"/>	Oscar	/users/	1	-	-	⌚ 3 minutes
<input type="checkbox"/>	Pam	/users/	1	-	-	⌚ 3 minutes

task7_aws_console_group_members. Png

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access Management' (selected), 'User groups' (selected), 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Root access management', and 'Temporary delegation requests'. The main content area shows the 'Users in this group (26)' section. A search bar at the top of this section has 'Search IAM' placeholder text. Below it is a table with columns: 'User name' (with a checkbox header), 'Groups', 'Last activity', and 'Creation time'. The table lists 26 users, all of whom are part of the 'developers' group and were last active 4 minutes ago.

<input type="checkbox"/>	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	Andy	developers	None	4 minutes ago
<input type="checkbox"/>	Angela	developers	None	4 minutes ago
<input type="checkbox"/>	Charles	developers	None	4 minutes ago
<input type="checkbox"/>	Clark	developers	None	4 minutes ago
<input type="checkbox"/>	Creed	developers	None	4 minutes ago
<input type="checkbox"/>	Darryl	developers	None	4 minutes ago
<input type="checkbox"/>	David	developers	None	4 minutes ago
<input type="checkbox"/>	Dwight	developers	None	4 minutes ago
<input type="checkbox"/>	Erin	developers	None	4 minutes ago
<hr/>				
<input type="checkbox"/>	Holly	developers	None	4 minutes ago
<input type="checkbox"/>	Jan	developers	None	4 minutes ago
<input type="checkbox"/>	Jim	developers	None	4 minutes ago
<input type="checkbox"/>	Jo	developers	None	4 minutes ago
<input type="checkbox"/>	Kelly	developers	None	4 minutes ago
<input type="checkbox"/>	Kevin	developers	None	4 minutes ago
<input type="checkbox"/>	Meredith	developers	None	4 minutes ago
<input type="checkbox"/>	Michael	developers	None	4 minutes ago
<input type="checkbox"/>	Oscar	developers	None	4 minutes ago
<input type="checkbox"/>	Pam	developers	None	4 minutes ago

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for "User groups", "Users", "Roles", "Policies", "Identity providers", "Account settings", "Root access management", and "Temporary delegation requests". The main content area is titled "User groups" and shows a list of users: David, Dwight, Erin, Gabe, Holly, Jan, Jim, Jo, Kelly, Kevin, Meredith, Michael, Oscar, and Pam. Each user entry includes a checkbox, a link to their profile, and columns for Last sign-in (None) and Last updated (4 minutes ago).

task7_aws_console_user_access_key.png

The screenshot shows the AWS IAM "Users" page for a user named "Andy". The left sidebar is identical to the previous screenshot. The main content area is titled "Andy" and shows the "Summary" tab. It displays the ARN (arn:aws:iam::353695163380:user/users/Andy), Console access status (Enabled without MFA), and two access keys. The "Security credentials" tab is selected, showing the "Console sign-in" section with a "Console sign-in link" (https://353695163380.signin.aws.amazon.com/console) and the "Multi-factor authentication (MFA) (0)" section. Navigation tabs at the bottom include "Permissions", "Groups (1)", "Tags (2)", "Security credentials", and "Last Accessed".

task7_s3_tfstate_multiple_users.png

Amazon S3 > Buckets > myapp-s3-bucket-demo-arschia014

myapp-s3-bucket-demo-arschia014 [Info](#)

Objects (1) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

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

Name	Type	Last modified	Size	Storage class
myapp/	Folder	-	-	-

Amazon S3 > Buckets > myapp-s3-bucket-demo-arschia014 > myapp/

myapp/ [Copy S3 URI](#)

Objects (1) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

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

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	January 16, 2026, 21:47:16 (UTC+05:00)	95.0 KB	Standard

Amazon S3 > Buckets > myapp-s3-bucket-demo-arschia014 > myapp/ > terraform.tfstate

terraform.tfstate [Info](#)

Properties [Permissions](#) [Versions](#)

Object overview

Owner a88688601faac15aa1450a90d6caed8b3050e020aad8b34cd545c2d2dbff4879

AWS Region US East (N. Virginia) us-east-1

Last modified January 16, 2026, 21:47:16 (UTC+05:00)

Size 95.0 KB

Type tfstate

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

S3 URI [s3://myapp-s3-bucket-demo-arschia014/myapp/terraform.tfstate](#)

Amazon Resource Name (ARN) [arn:aws:s3:::myapp-s3-bucket-demo-arschia014/myapp/terraform.tfstate](#)

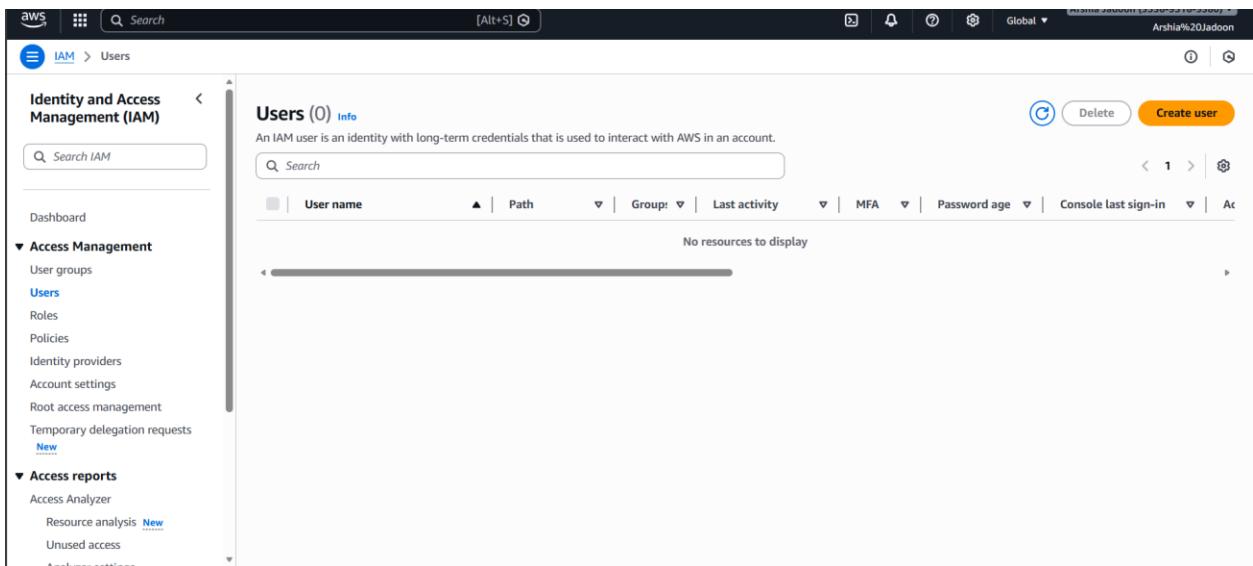
Entity tag (Etag) [6388cae546be25cc0fc478a71e214ffd](#)

Object URL [https://myapp-s3-bucket-demo-arschia014.s3.us-east-1.amazonaws.com/myapp/terraform.tfstate](#)

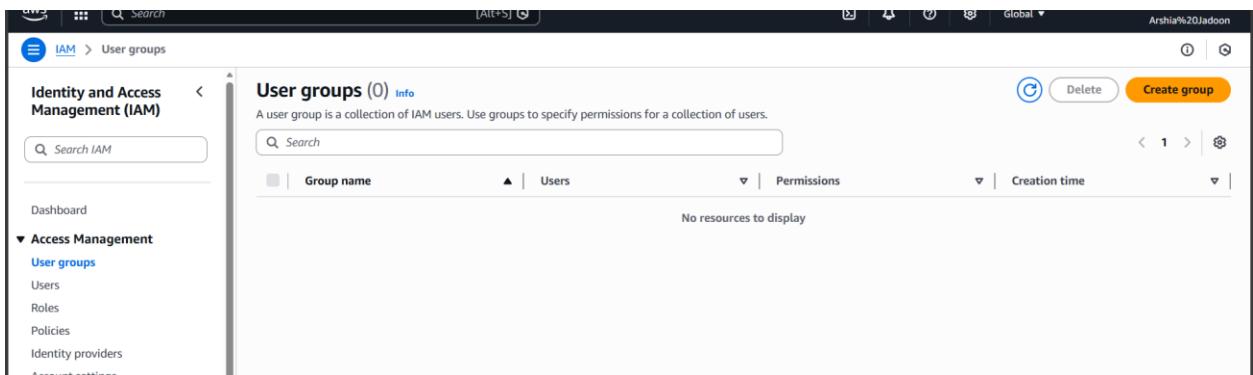
cleanup_destroy_complete.png

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

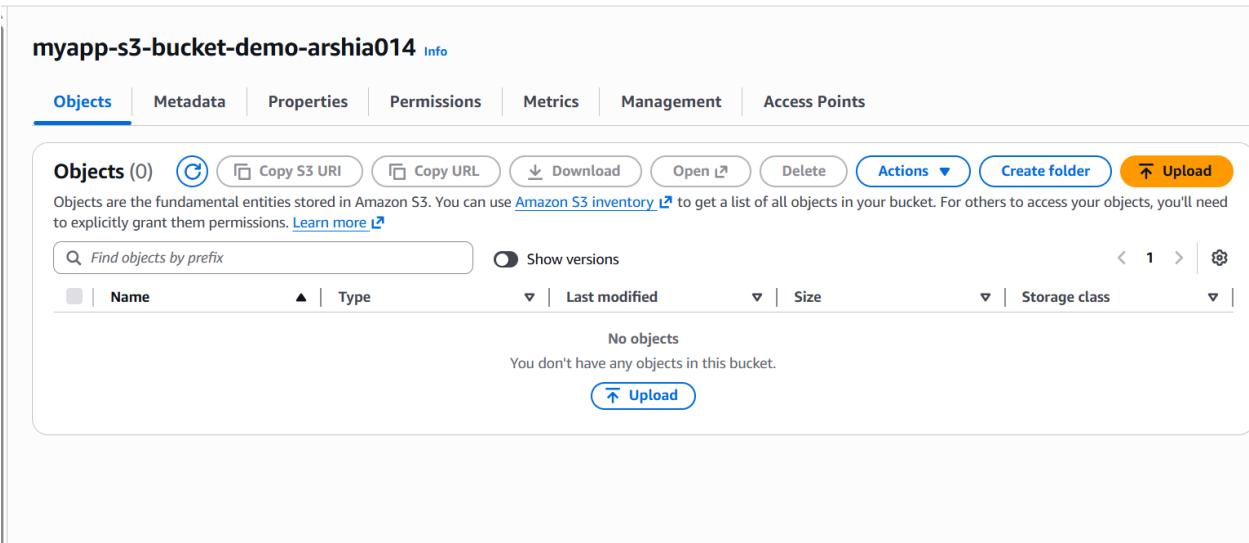
cleanup_aws_console_users_deleted.png



cleanup_aws_console_group_deleted.png



cleanup_s3_empty_state.png



cleanup_final_files.png

```
Destroy complete! Resources: 107 destroyed.
@arshiajadoon → ~/Lab13 $ ls -la
total 65360
drwxrwxr-x 4 codespace codespace      4096 Jan 16 16:44 .
drwxr-x--- 1 codespace codespace      4096 Jan 16 15:48 ..
drwxr-xr-x 3 codespace codespace      4096 Jan 16 16:45 .terraform
-rw-r--r-- 1 codespace codespace     2422 Jan 16 15:46 .terraform.lock.hcl
drwxr-xr-x 3 codespace codespace      4096 Jan 15 19:17 aws
-rw-rw-r-- 1 codespace codespace  66872783 Jan 16 14:59 awscli2.zip
-rwxrwxr-x 1 codespace codespace      440 Jan 16 15:41 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace       50 Jan 16 16:41 locals.tf
-rw-rw-r-- 1 codespace codespace    2474 Jan 16 16:44 main.tf
-rw-rw-r-- 1 codespace codespace        0 Jan 16 16:31 terraform.tfstate
-rw-rw-r-- 1 codespace codespace   6882 Jan 16 16:31 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace      167 Jan 16 16:41 users.csv
-rw-rw-r-- 1 codespace codespace      154 Jan 16 15:39 variables.tf
```

cleanup_s3_bucket_deleted.png (optional)

