task0_codespace_create_and_list.png

```
error fetching codespace information: HTTP 404: Not Found (https://api.github.com/user/codespaces/super-duper-tribble-97
g5...)
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-7vj59wq6g5rv...  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 $
```

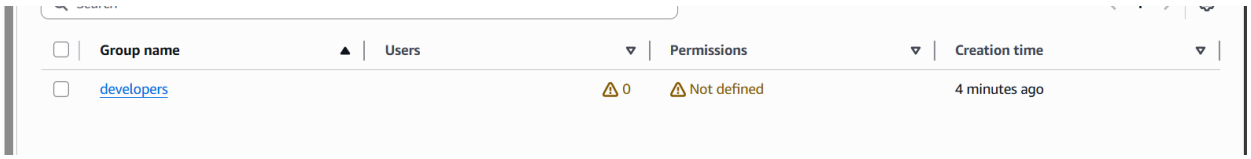
task1_terraform_apply.png-

```
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" = "AGPAVEWO74P2IKMTVRUZN"
}
@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" = "AGPAVEWO74P2IKMTVRUZN"
}
@arshiajadoon ➜ ~/Lab13 $
```

task1_aws_console_group.png



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 $
```

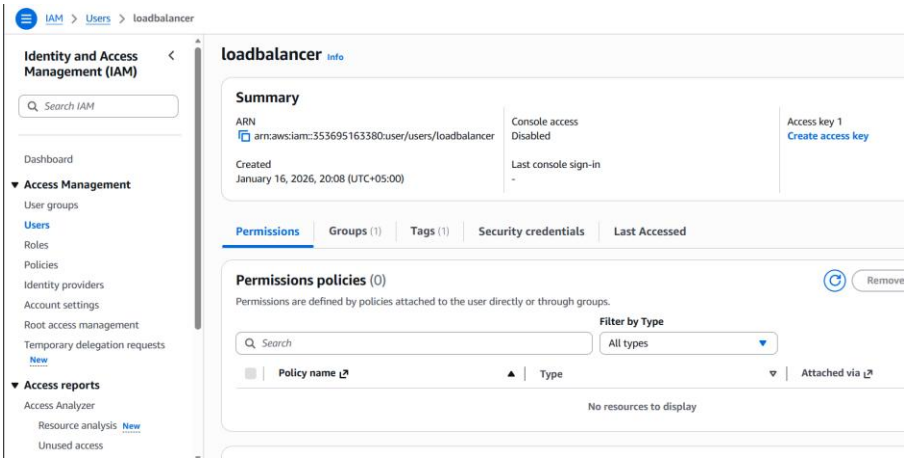
task2_terraform_apply.png

```
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" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon →~/Lab13 $
```

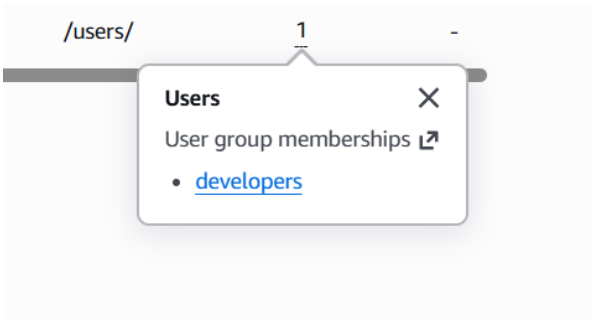
task2_terraform_output.png

```
Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "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" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon →~/Lab13 $
```

task2_aws_console_user.png



task2_aws_console_user_groups.png



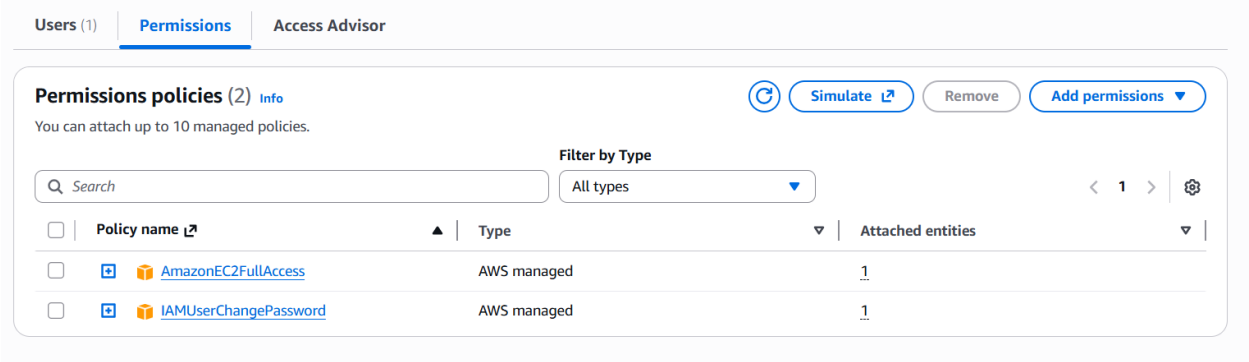
task3_main_tf_policies.png

```
    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" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon →~/Lab13 $
```

task3_aws_console_policies.png



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" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "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     = "1dontKnow"
}
EOF
@arshiajadoon →~/Lab13 $ cat variables.tf
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "1dontKnow"
}
@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" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "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 = "AKIAVEWO74P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon →~/Lab13 $
```

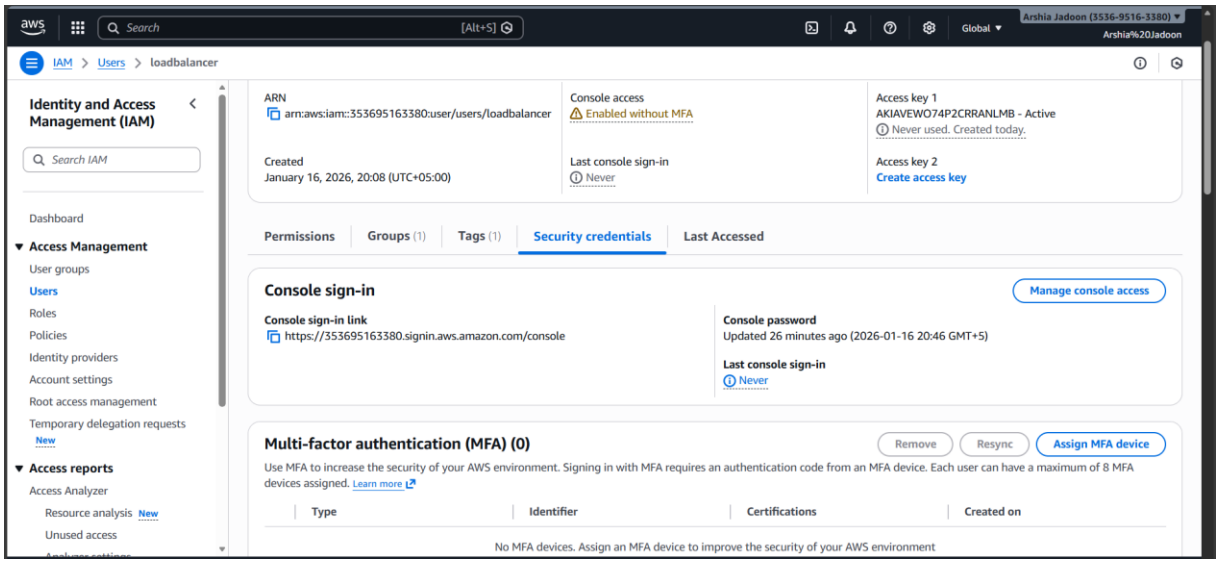
task5_terraform_output.png

```
@arshiajadoon →~/Lab13 $ terraform output
access_key_id = "AKIAVEWO74P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon →~/Lab13 $
```

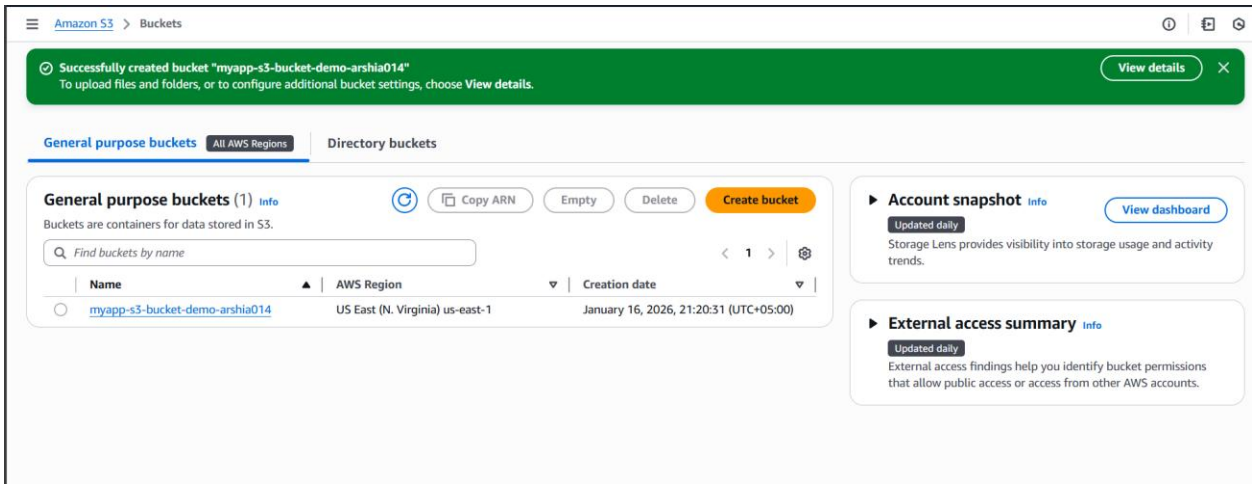
task5_tfstate_secret. Png

```
@arshiajadoon →~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
    "access_key_secret": {
      "value": "G+FWmEYmKeND0d9VGRvMo/eOl0AU8p5rLJAHgRE7",
      "type": "string",
      "sensitive": true
    },
    "group_details": {
      "value": {
        "group_arn": "arn:aws:iam::353695163380:group/groups/developers",
        "group_name": "developers",
        "unique_id": "AGPAVEWO74P2IKMTVRUZN"
      },
@arshiajadoon →~/Lab13 $
```

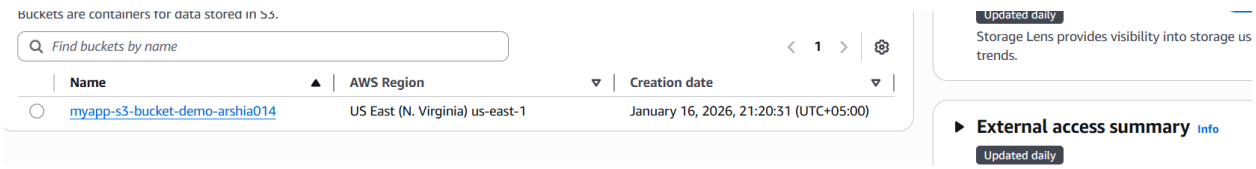
task5_aws_console_access_keys. png



task6_s3_bucket_create.png



task6_s3_bucket_versioning.png



task6_main_tf_backend.png

```
Windows PowerShell
  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 $
```

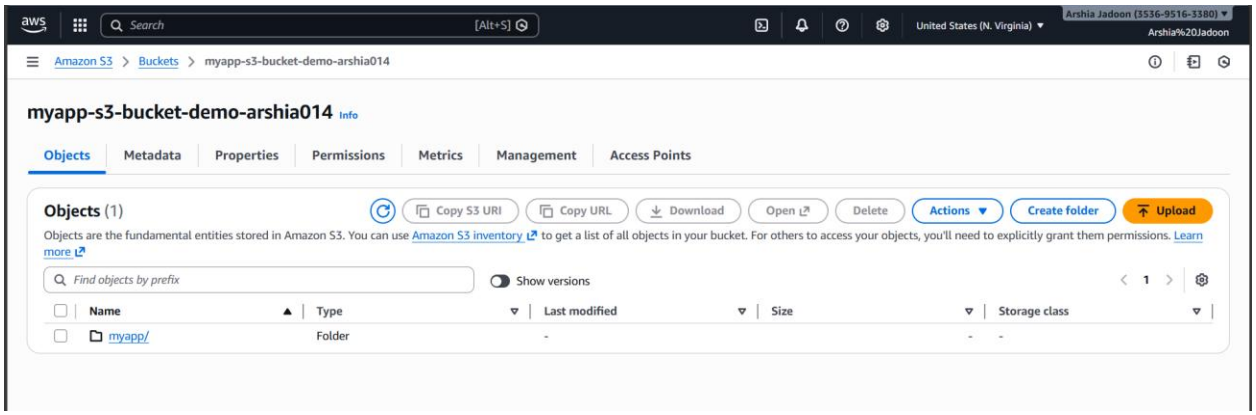
task6_terraform_init_migrate.png

```
@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_user_group_membership.lb_membership: Refreshing state... [id=ter

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configurati
needed.

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

Outputs:

access_key_id = "AKIAVEWO74P2CRRANLMB"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2IKMTVRUZN"
}
user_details = {
  "unique_id" = "AIDAVEWO74P2G36QCQ2EM"
  "user_arn" = "arn:aws:iam::353695163380:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
@arshiajadoon ➜ ~/Lab13 $
```

task6_s3_tfstate_file.png



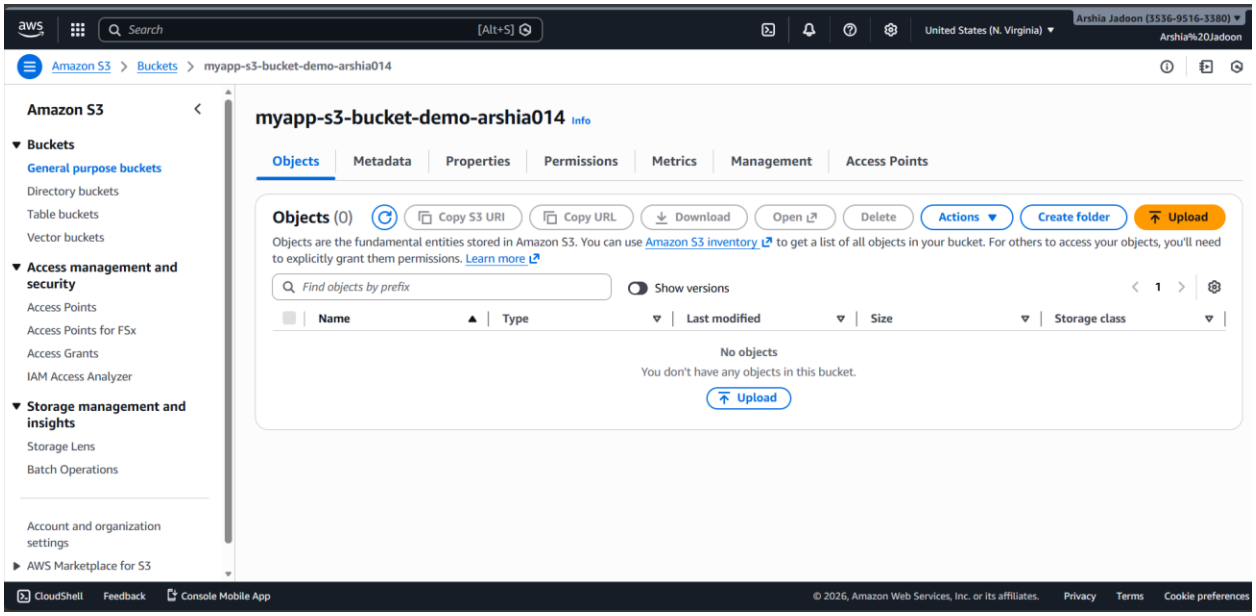
task6_local_state_backup.png

```
@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]
null_resource.create_login_profile: Destruction complete after 0s
aws_iam_user_group_membership.lb_membership: Destroying... [id=terraform-20260116150819547000000001]
aws_iam_group_policy_attachment.change_password: Destroying... [id=developers-20260116153826172600000002]
aws_iam_access_key.lb_access_key: Destroying... [id=AKIAVEWO74P2CRRANLMB]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destroying... [id=developers-20260116153826136100000001]
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" = "AIDAVEWO74P2EFOXUTY6I"
  }
  "Ryan" = {
    "access_key_id" = "AKIAVEWO74P2GILD7SOY"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Ryan"
    "user_unique_id" = "AIDAVEWO74P2DAOZHBIJ4"
  }
  "Stanley" = {
    "access_key_id" = "AKIAVEWO74P2E7YFYIN3"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Stanley"
    "user_unique_id" = "AIDAVEWO74P2FESFZ3XX4"
  }
  "Toby" = {
    "access_key_id" = "AKIAVEWO74P2AXHLQEXW"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Toby"
    "user_unique_id" = "AIDAVEWO74P2M65OGBUH6"
  }
}
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2NJUGZCFUY"
}
@arshiajadoon ➜ ~/Lab13 $
```

task7_terraform_output.png

```
@arshiajadoon ➜ ~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAVEWO74P2OAW646KU"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Andy"
    "user_unique_id" = "AIDAVEWO74P2G5IOIR4IU"
  }
  "Angela" = {
    "access_key_id" = "AKIAVEWO74P2M6DNBVSS"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Angela"
    "user_unique_id" = "AIDAVEWO74P2OBHOVDBH7"
  }
  "Charles" = {
    "access_key_id" = "AKIAVEWO74P2OO23Q7OR"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Charles"
    "user_unique_id" = "AIDAVEWO74P2DDVEIJL6Z"
  }
  "Clark" = {
    "access_key_id" = "AKIAVEWO74P2F43SAWUQ"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Clark"
    "user_unique_id" = "AIDAVEWO74P2HMNNFKNIN"
  }
  "Creed" = {
    "access_key_id" = "AKIAVEWO74P2MUZEFUGC"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Creed"
```

```
  }
  "Charles" = {
    "access_key_id" = "AKIAVEWO74P2OO23Q7OR"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Charles"
    "user_unique_id" = "AIDAVEWO74P2DDVEIJL6Z"
  }
  "Clark" = {
    "access_key_id" = "AKIAVEWO74P2F43SAWUQ"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Clark"
    "user_unique_id" = "AIDAVEWO74P2HMNNFKNIN"
  }
  "Creed" = {
    "access_key_id" = "AKIAVEWO74P2MUZEFUGC"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Creed"
    "user_unique_id" = "AIDAVEWO74P2EHCQAZDSD"
  }
  "Darryl" = {
    "access_key_id" = "AKIAVEWO74P2HSBIBISJ"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Darryl"
    "user_unique_id" = "AIDAVEWO74P2HYWQEWWZS"
  }
  "David" = {
    "access_key_id" = "AKIAVEWO74P2NIZYRJP6"
    "user_arn" = "arn:aws:iam::353695163380:user/users/David"
    "user_unique_id" = "AIDAVEWO74P2AWFFS2ANV"
  }
```

```
    "user_arn" = "arn:aws:iam::353695163380:user/users/Phyllis"
    "user_unique_id" = "AIDAVEWO74P2EQXORKJIG"
  }
  "Robert" = {
    "access_key_id" = "AKIAVEWO74P2PQQGQ54T"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Robert"
    "user_unique_id" = "AIDAVEWO74P2EFOXUTY6I"
  }
  "Ryan" = {
    "access_key_id" = "AKIAVEWO74P2GILD7SOY"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Ryan"
    "user_unique_id" = "AIDAVEWO74P2DAOZHBIJ4"
  }
  "Stanley" = {
    "access_key_id" = "AKIAVEWO74P2E7YFYIN3"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Stanley"
    "user_unique_id" = "AIDAVEWO74P2FESFZ3XX4"
  }
  "Toby" = {
    "access_key_id" = "AKIAVEWO74P2AXHLQEXW"
    "user_arn" = "arn:aws:iam::353695163380:user/users/Toby"
    "user_unique_id" = "AIDAVEWO74P2M65OGBUH6"
  }
}
group_details = {
  "group_arn" = "arn:aws:iam::353695163380:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAVEWO74P2NJUGZCFUY"
}
@arshiajadoon ➜ ~/Lab13 $
```

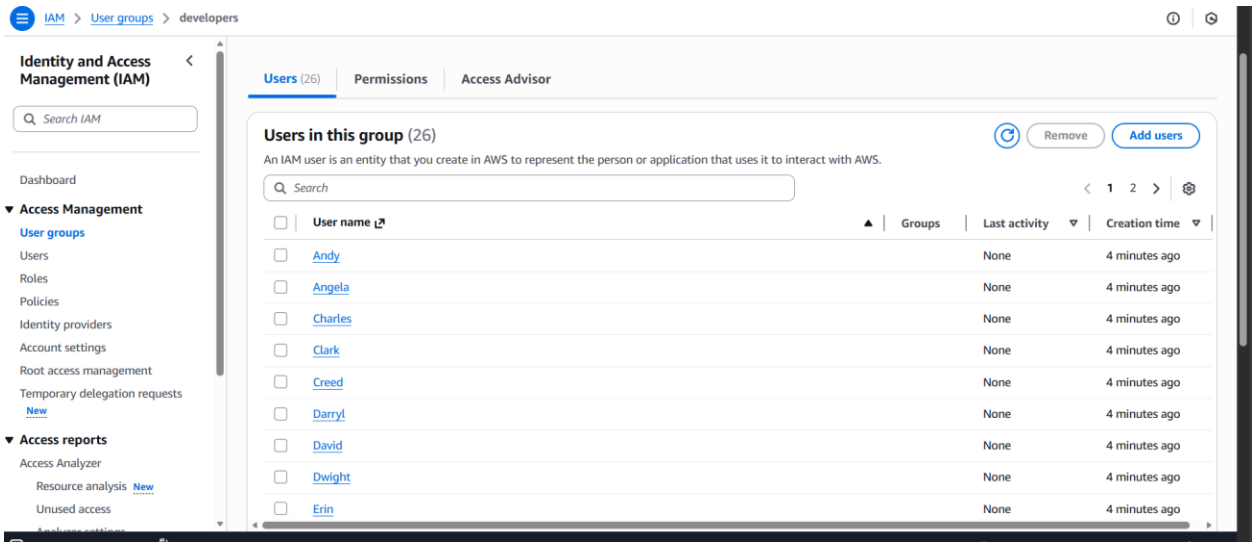
task7_tfstate_secrets. Png

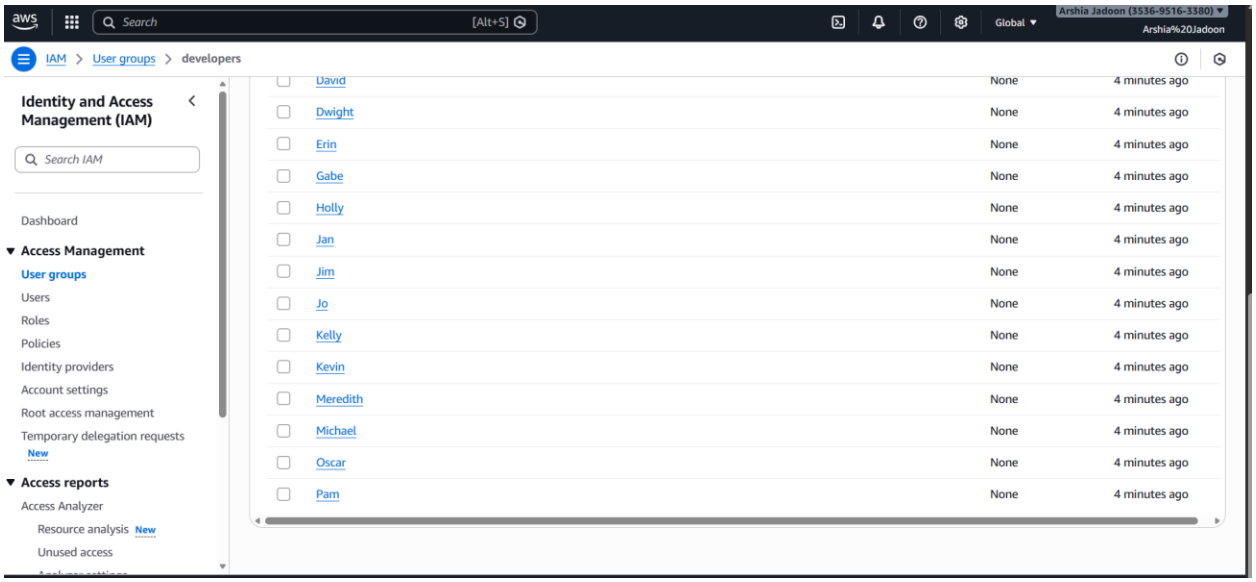
```
@arshiajadoon ➜ ~/Lab13 $ cat terraform.tfstate | grep -A 5 "all_access_key_secrets"
@arshiajadoon ➜ ~/Lab13 $
```

task7_aws_console_all_users. Png

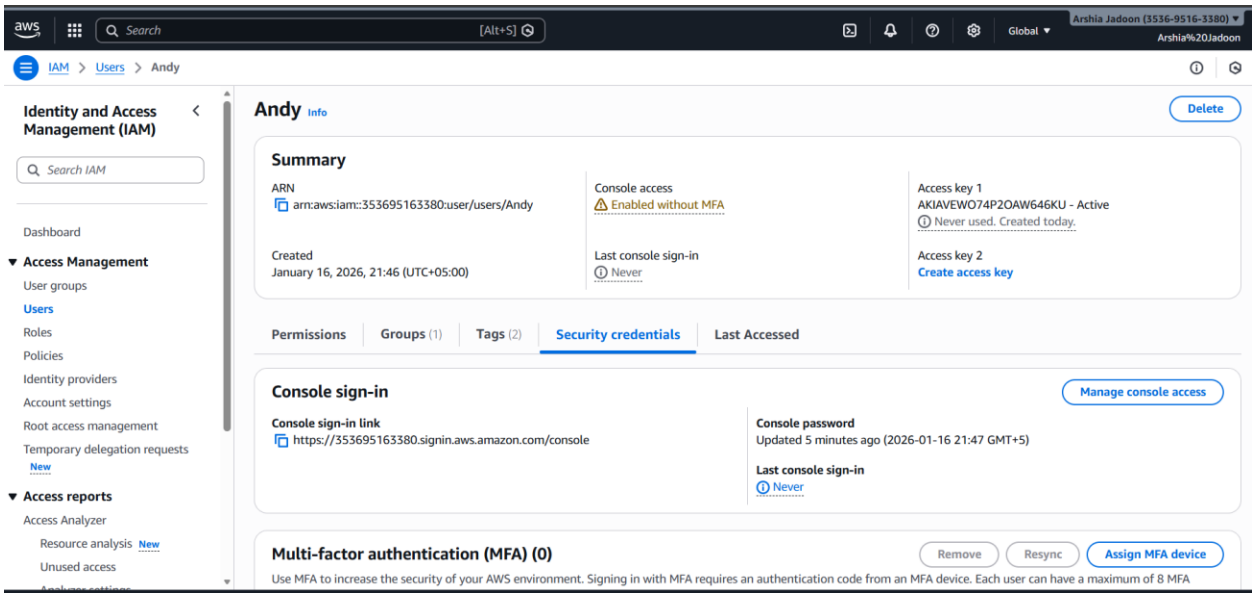
User name	Path	Groups	Last activity	MFA	Password age
Andy	/users/	1	-	-	2 minutes
Angela	/users/	1	-	-	3 minutes
Charles	/users/	1	-	-	3 minutes
Clark	/users/	1	-	-	2 minutes
Creed	/users/	1	-	-	2 minutes
Darryl	/users/	1	-	-	2 minutes
David	/users/	1	-	-	3 minutes
Dwight	/users/	1	-	-	3 minutes
Erin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

task7_aws_console_group_members. Png

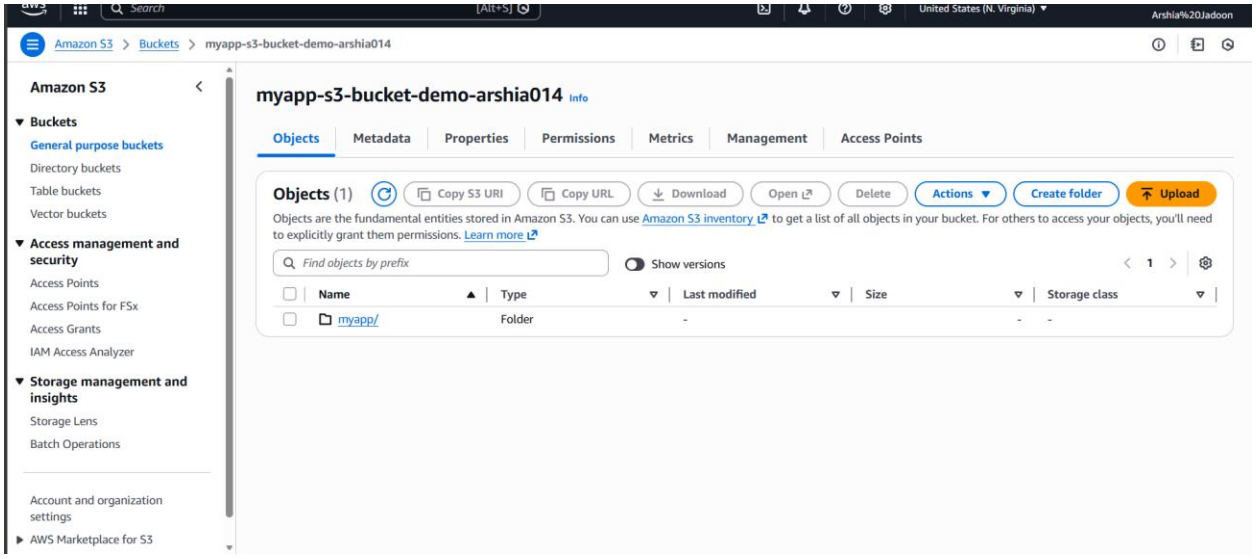


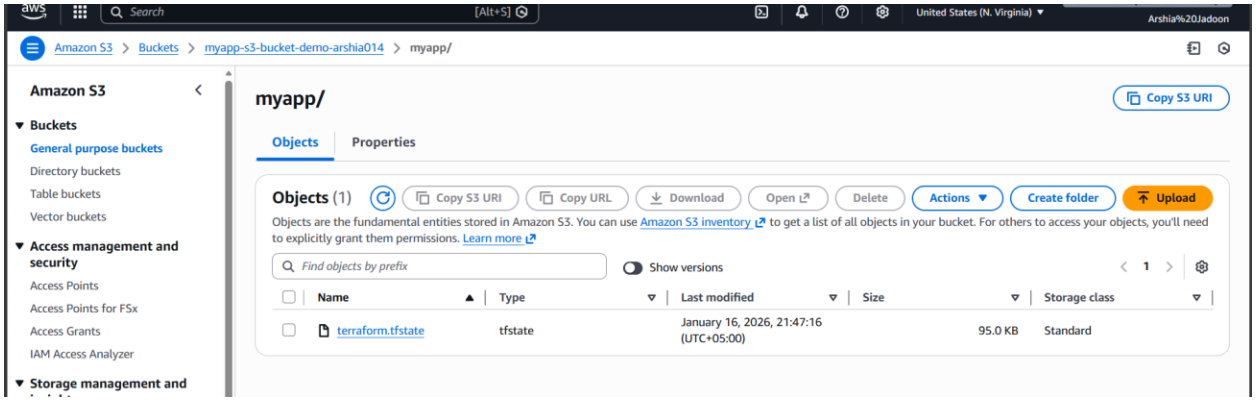


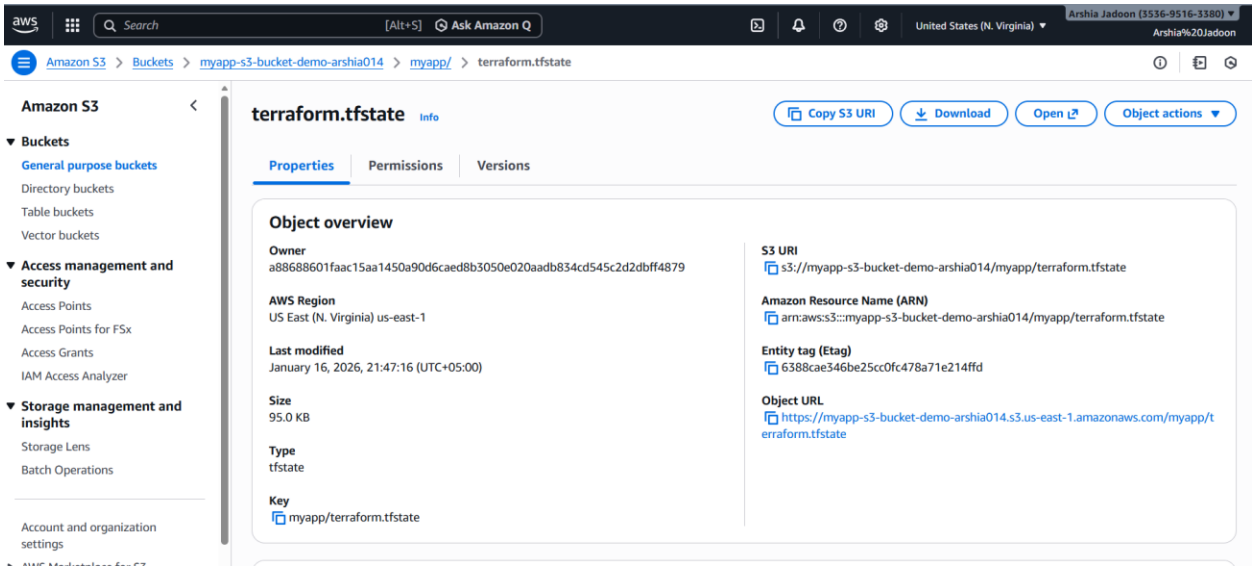
task7_aws_console_user_access_key.png



task7_s3_tfstate_multiple_users.png



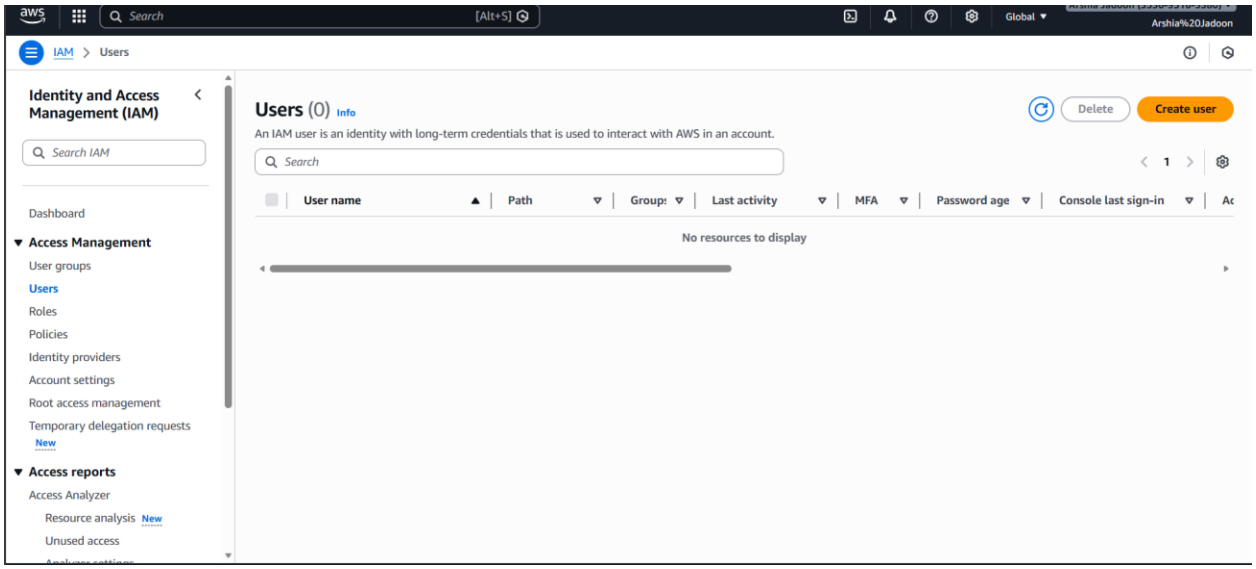




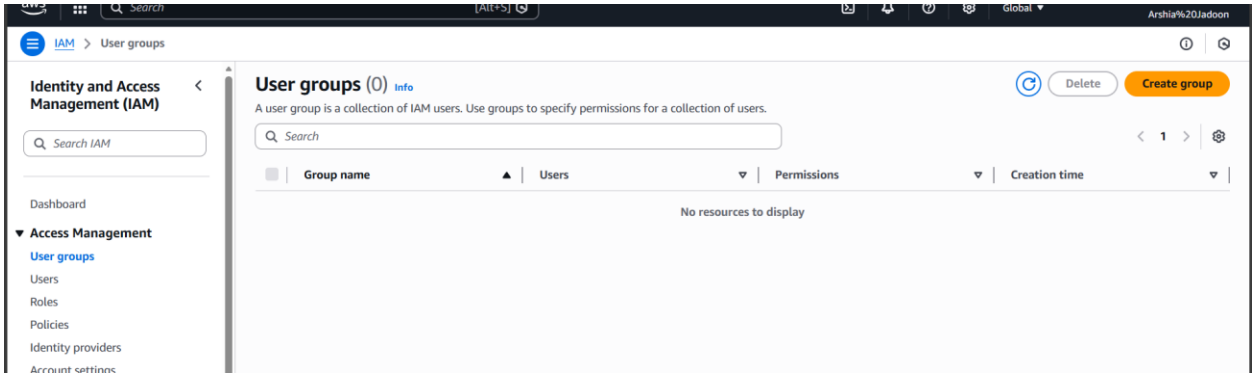
cleanup_destroy_complete.png

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

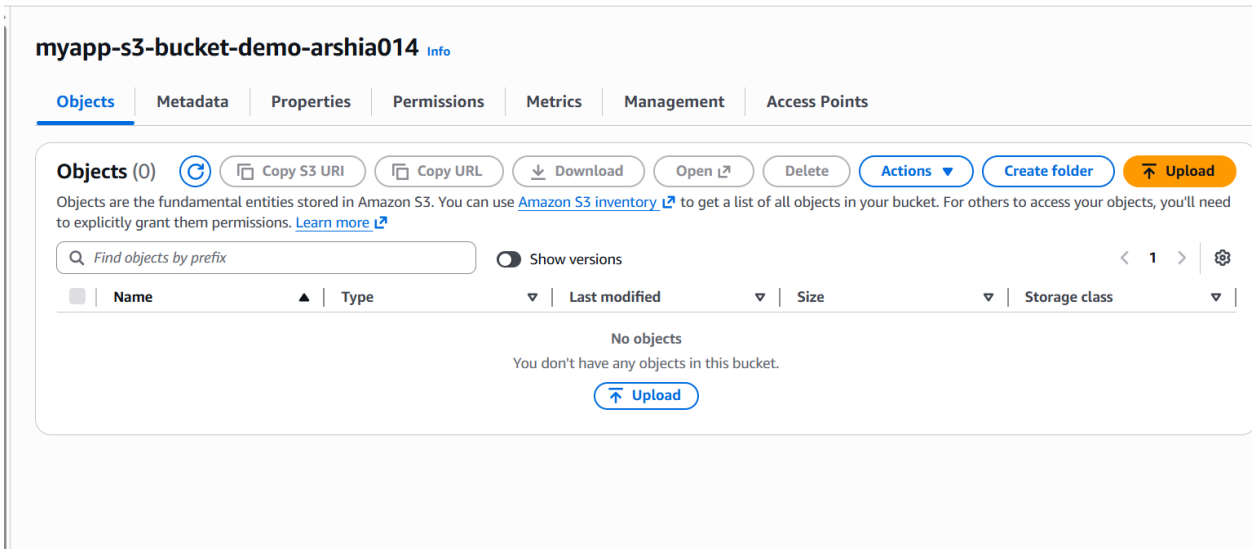
cleanup_aws_console_users_deleted.png



cleanup_aws_console_group_deleted.png



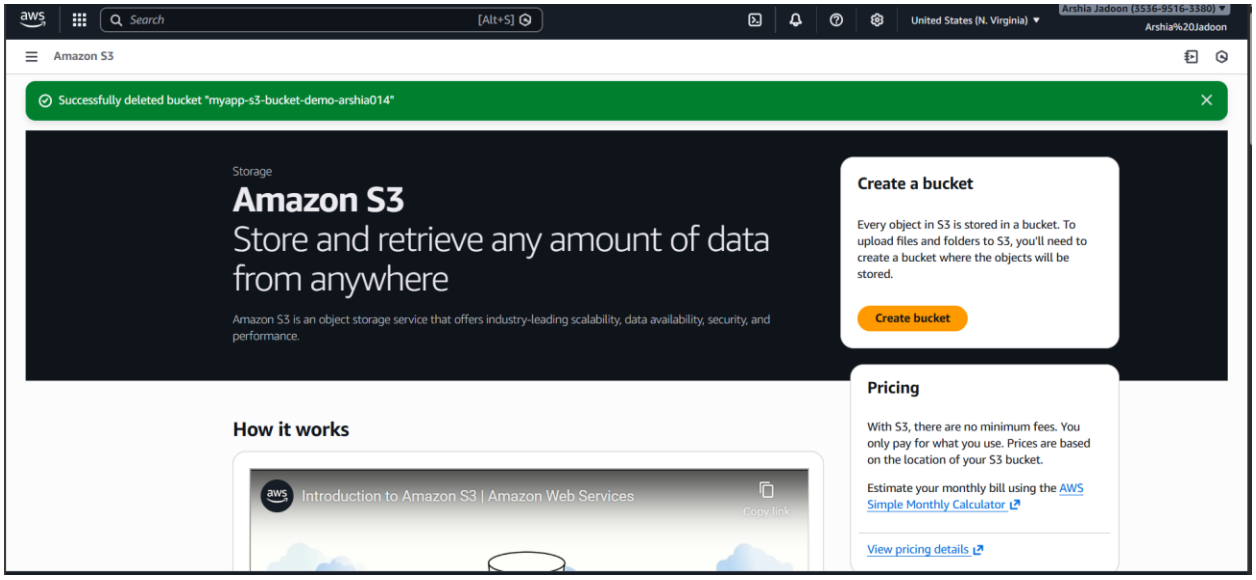
cleanup_s3_empty_state.png



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 awscliv2.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)


Search
[Alt+S]
United States (N. Virginia)
Arshia Jadoon (3536-9516-3380)
Arshia%20Jadoon
Amazon S3
Successfully deleted bucket "myapp-s3-bucket-demo-arshia014"
Storage
Amazon S3
Store and retrieve any amount of data from anywhere
Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.
Create a bucket
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.
Create bucket
Pricing
With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.
Estimate your monthly bill using the AWS Simple Monthly Calculator
View pricing details
How it works
Introduction to Amazon S3 | Amazon Web Services
Copy link