NAME: SANA TARIQ
CLASS: BSE 5B
REG.NO: 058
SUBJECT: CLOUD COMPUTING

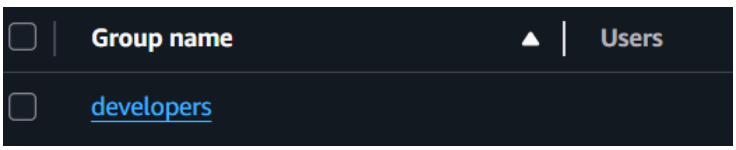
LAB#13

TASK 1: Create IAM Group

```
@SanaTariq205 /workspaces/cc_sanatariq_058 (main) $ mkdir -p ~/Lab13
@SanaTariq205 /workspaces/cc_sanatariq_058 (main) $ cd ~/Lab13
@SanaTariq205 ~/Lab13 $ touch main.tf
@SanaTariq205 ~/Lab13 $ aws configure
AWS Access Key ID [****************NJUB]:
AWS Secret Access Key [****************zGVz]:
Default region name [us-east-1]:
Default output format [json]:
@SanaTariq205 ~/Lab13 $
```

```
  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
  }
}
```

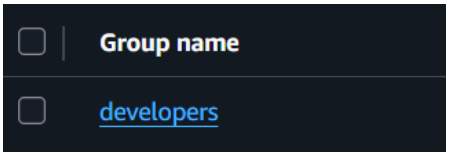
```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
@SanaTariq205 ~/Lab13 $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
```



TASK 2: Create IAM User

```
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
  }
}
```

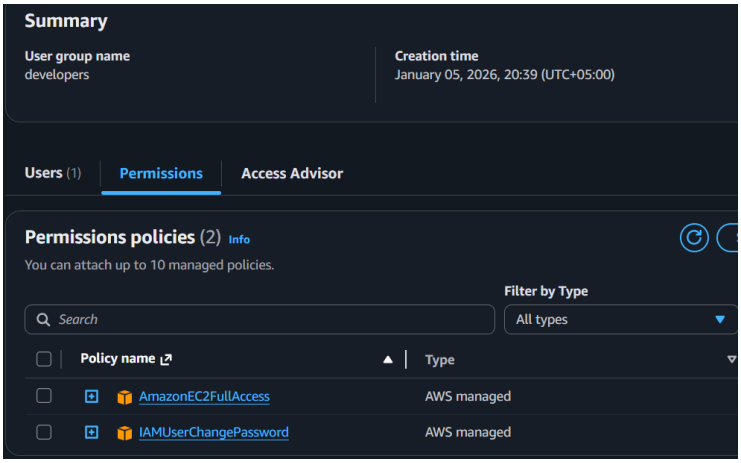
```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXNZXSSARW5"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```



TASK 3: Attach Policies

```
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"
}
```

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXNZXSSARW5"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```



TASK 4: Create Login Profile

```
 GNU nano 7.2                                    variables.tf *
variable "iam_password" {
  description = "Temporary password for the IAM user"
  type        = string
  sensitive   = true
  default     = "1dontKnow"
}
```

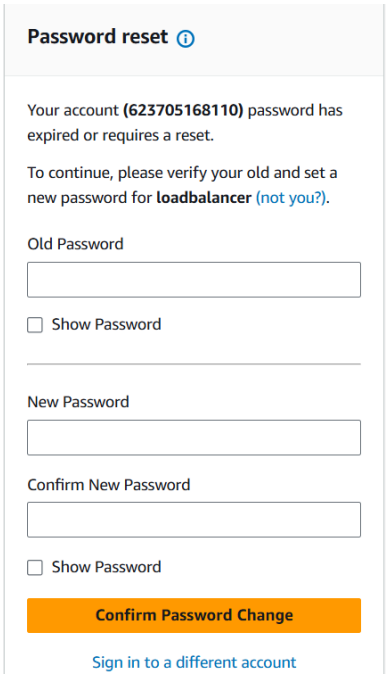
```
 GNU nano 7.2                                  create-login-profile.sh *
#!/usr/bin/env bash
set -euo pipefail

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
```

```
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}'"
  }
}
```

```
@SanaTariq205 ~/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_group_policy_attachment.change_password: Refreshing state... [id=developers-20260105154535388500000001]
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-20260105154535432700002]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260105154220909000000001]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated
```

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXNZXSSARW5"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

```
@SanaTariq205 ~/Lab13 $ aws iam get-login-profile --user-name loadbalancer
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-05T15:51:36+00:00",
        "PasswordResetRequired": true
    }
}
```

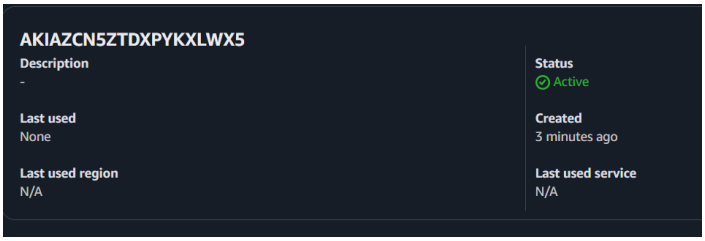


TASK 5: Generate Access Keys

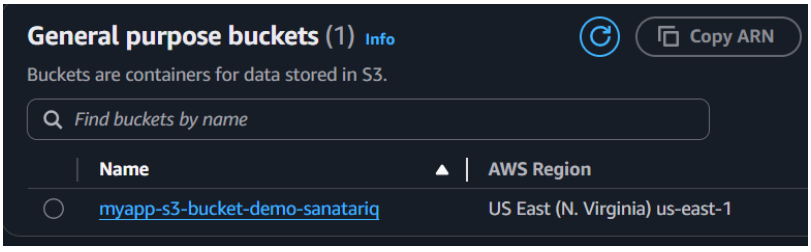
```
resource "aws_iam_access_key" "lb_access_key" {
  user = aws_iam_user.lb.name
}

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
}
```

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

Outputs:

access_key_id = "AKIAZCN5ZTDXPYKXLWX5"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXNZXSSARW5"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
}
```

```
@SanaTariq205 ~/Lab13 $ cat terraform.tfstate | grep -A 10 "access_key_secret"
    "access_key_secret": {
      "value": "X4O1UrwZUQXeBkA75c9C2O1K63+bXS2YD53o3B68",
      "type": "string",
      "sensitive": true
    },
    "group_details": {
      "value": {
        "group_arn": "arn:aws:iam::623705168110:group/groups/developers",
        "group_name": "developers",
        "unique_id": "AGPAZCN5ZTDXCQ74LALWG"
      },
```



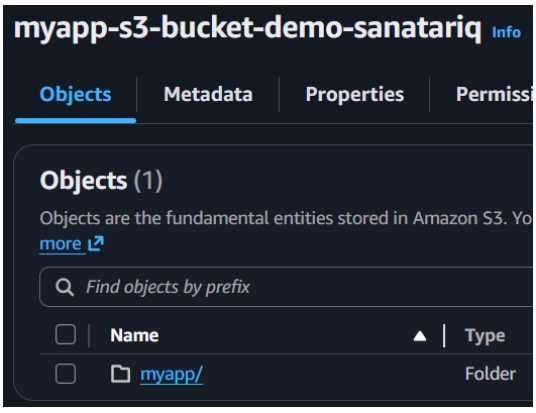
TASK 6: Remote State with S3

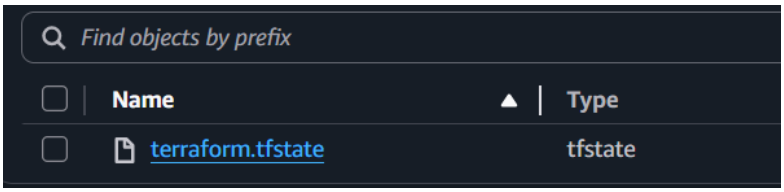


```
  GNU nano 7.2                                                  main.tf *
terraform {
  backend "s3" {
    bucket  = "myapp-s3-bucket-demo-sanatariq"
    key     = "myapp/terraform.tfstate"
    region  = "us-east-1"
    encrypt = true
  }
}
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
```

```
@SanaTariq205 ~/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 v6.27.0
- Using previously-installed hashicorp/null v3.2.4

Terraform has been successfully initialized!
```

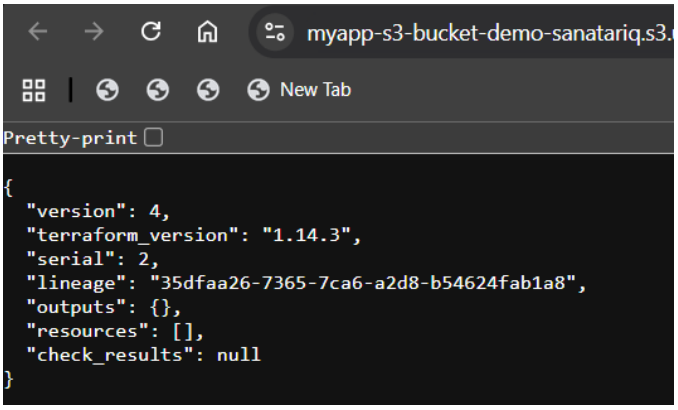
```
@SanaTariq205 ~/Lab13 $ terraform apply -auto-approve -var="iam_password=MySecurePass123!"
aws_iam_group.developers: Refreshing state... [id=developers]
aws_iam_user.lb: Refreshing state... [id=loadbalancer]
aws_iam_access_key.lb_access_key: Refreshing state... [id=AKIAZCN5ZTDXPYKXLWX5]
null_resource.create_login_profile: Refreshing state... [id=8648110953209236919]
aws_iam_group_policy_attachment.change_password: Refreshing state... [id=developers-2026010515
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Refreshing state... [id=developers-2
00002]
aws_iam_user_group_membership.lb_membership: Refreshing state... [id=terraform-20260105154220

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differ
changes are needed.

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

Outputs:

access_key_id = "AKIAZCN5ZTDXPYKXLWX5"
access_key_secret = <sensitive>
group_details = {
  "group_arn" = "arn:aws:iam::623705168110:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPAZCN5ZTDXCQ74LALWG"
}
user_details = {
  "unique_id" = "AIDAZCN5ZTDXNZXSSARW5"
  "user_arn" = "arn:aws:iam::623705168110:user/users/loadbalancer"
  "user_name" = "loadbalancer"
```






```
@SanaTariq205 ~/Lab13 $ ls -la terraform.tfstate*
-rw-rw-r-- 1 codespace codespace    0 Jan  5 16:05 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan  5 16:05 terraform.tfstate.backup
@SanaTariq205 ~/Lab13 $
```

```
aws_iam_group.developers: Destruction complete after 0s
aws_iam_user.lb: Destruction complete after 2s

Destroy complete! Resources: 7 destroyed.
```



TASK 7: Multiple Users from CSV

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

```
 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
```

```
resource "aws_iam_user" "users" {
  for_each = { for user in local.users : user.user_name => user }

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

  tags = {
    DisplayName = each.value.user_name
    CreatedBy   = "Terraform"
  }
}
resource "aws_iam_user_group_membership" "users_membership" {
  for_each = aws_iam_user.users

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

```
resource "null_resource" "create_login_profiles" {
  for_each = aws_iam_user.users

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

  depends_on = [aws_iam_user.users]

  provisioner "local-exec" {
    command = "${path.module}/create-login-profile.sh ${each.value.name} '${var.iam_password}'"
  }
}
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" {
  value = {
    for user_name, key in aws_iam_access_key.users_access_keys : user_name => key.secret
  }
  sensitive = true
}
```

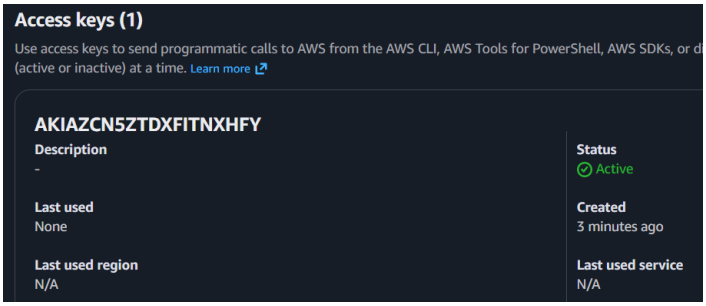
```
@SanaTariq205 ~/Lab13 $ terraform output
all_access_key_secrets = <sensitive>
all_users_details = {
  "Andy" = {
    "access_key_id" = "AKIAZCN5ZTDXEE74TBHA"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Andy"
    "user_unique_id" = "AIDAZCN5ZTDXEGZBLOZSP"
  }
  "Angela" = {
    "access_key_id" = "AKIAZCN5ZTDXFITNXHFY"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Angela"
    "user_unique_id" = "AIDAZCN5ZTDXGFRMZ5K7P"
  }
  "Charles" = {
    "access_key_id" = "AKIAZCN5ZTDXL6BUMCGT"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Charles"
    "user_unique_id" = "AIDAZCN5ZTDXLXYUSTQ7V"
  }
  "Clark" = {
    "access_key_id" = "AKIAZCN5ZTDXM5RHGW7K"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Clark"
    "user_unique_id" = "AIDAZCN5ZTDXJLBGRFY5K"
  }
  "Creed" = {
    "access_key_id" = "AKIAZCN5ZTDXOEVZZAIM"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Creed"
    "user_unique_id" = "AIDAZCN5ZTDXMN2MR2DDZ"
  }
  "Darryl" = {
    "access_key_id" = "AKIAZCN5ZTDXD4BC3D4M"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Darryl"
    "user_unique_id" = "AIDAZCN5ZTDXHP6SHZ7VV"
  }
  "David" = {
    "access_key_id" = "AKIAZCN5ZTDXFZNVQA2C"
    "user_arn" = "arn:aws:iam::623705168110:user/users/David"
    "user_unique_id" = "AIDAZCN5ZTDXDFW75FXZP"
  }
  "Dwight" = {
    "access_key_id" = "AKIAZCN5ZTDXJ6ZRUZWP"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Dwight"
    "user_unique_id" = "AIDAZCN5ZTDXOIZJPAQMN"
  }
  "Erin" = {
    "access_key_id" = "AKIAZCN5ZTDXCMKM6DKO"
    "user_arn" = "arn:aws:iam::623705168110:user/users/Erin"
    "user_unique_id" = "AIDAZCN5ZTDXIEE45WXRH"
  }
  "Gabe" = {
```

Users (28) Info

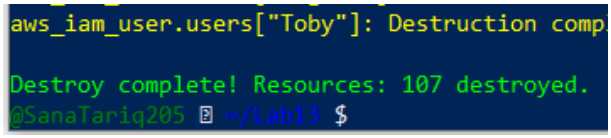
An IAM user is an identity with long-term credentials that is us

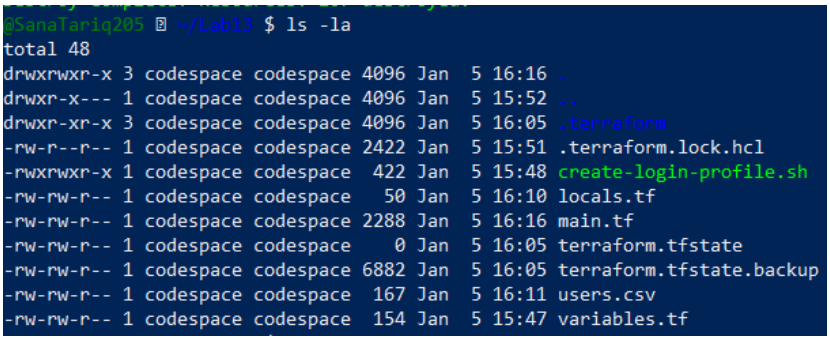
Search

User name	Path
Andy	/users/
Angela	/users/
Charles	/users/
Clark	/users/
Creed	/users/
Darryl	/users/
David	/users/
Dwight	/users/
Erin	/users/
Gabe	/users/


Access keys (1)
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or d
(active or inactive) at a time. Learn more
AKIAZCN5ZTDXFITNXHFY
Description
-
Status
Active
Last used
None
Created
3 minutes ago
Last used region
N/A
Last used service
N/A


myapp-s3-bucket-demo-sanatariq.s3.us-e
New Tab
Pretty-print
{
"version": 4,
"terraform_version": "1.14.3",
"serial": 4,
"lineage": "35dfaa26-7365-7ca6-a2d8-b54624fab1a8",
"outputs": {
"all_access_key_secrets": {
"value": {
"Andy": "ERf+HmK8DRZ3qALygFiSNK91vXcgOwIbqMNT/HFr",
"Angela": "5xhIPbqAbVzzKE8h0GWyTLNM/arkDffLdJVrpLqF",
"Charles": "NLZegsuzK6V0K/a5fIniL1Z0dvWfyWnW1h0lVzCU",
"Clark": "sd0GfQNlcrVB3392pivZgPoQxHUHB+I9epzHjr0P",
"Creed": "pExUkYMeJiiV/SNrtfJ8TCP3sRKuGortpxD8Y6WA",
"Darryl": "Z2psQ08mbwuwp8rU7PG4DSM4PkmcCPIz8hazwUMn",
"David": "Uitf9EgOEcsSyS5MLcfecTEE/8NIQEbqDMrSF6jM",
"Dwight": "XaJY6qznEn6wrFeKsr20cnwKbZJNqjYRLNMDrlGs",
"Erin": "NcMwK3GyfdcY7pf3lR4fqxr7J2QVW8QwppOAVOyE",
"Gabe": "HQZ4BENazdh3wKcf1yjmyWnpGiugdO4pKfshiuvP",
"Holly": "5J3QRbdnJ9GJ4Z1T7LH/0SaPBgLiyfz2P3FjfoQ7",
"Jan": "YyuSvp8glk7qvJKTmMEi61+4uTHy8UD9B9V3rDS2",
"Jim": "c0bCE3BlZnsReADWqiRKc6VinpvPm/8GchNHeYE4",
"Jo": "PFr0/UdwAouY3fAXBnvypvSF31IOuZJ5cLzQUN3B",
"Kelly": "vQVWe3w6XO5JJcNBrWvPDsarSuINphkZYs+dbDfi",
"Kevin": "YET9DI0WEKZ/C50lfIOXqUCNYyPc4x5VqQHf41vJ",
"Meredith": "tRVEaURXAA5SuJ1pJ+djv9CM8MUgL00LnXb5R+/w",
"Michael": "Z9y+fFRanzcAMTMavsEmV+CEudxXXYPz00+TWnsY",
"Oscar": "IdDmBIYwcJNL9AhgwaeOEqEh0Oh4AWKzob8Z6n3z",
"Pam": "YMagXIj8TvJExvJf4BQjod92AOkCH6rcuj5bELt+",
"Peter": "BRFg/0AvGnI0rX4SprWqM6BDS0E4d973lLe7PLy7",
"Phyllis": "m+SYwPgyCXFvRkyObGNDtZovyd++elR9KFvCGoqD",
"Robert": "Mu92OvNmEX61Ds+Fr7DlObAJ57VCCqFHY1f0TM2o",
"Ryan": "TOr4CNQr1wv2lnU3I0EHdg6nRLOGgS4ONzv7M0sj",
"Stanley": "XR5f73CkTar55VOIS6w/xWD7W9rgTXMmBUnuzaIt",
"Toby": "TEXC0zxQ6vxv0FqSmOt29twWO0gz+KgE+bYHhQGp"
},
"type": [
"object",
{
"Andy": "string",


aws_iam_user.users["Toby"]: Destruction comp
Destroy complete! Resources: 107 destroyed.
@SanaTariq205 ~/Lab13 $


@SanaTariq205 ~/Lab13 $ ls -la
total 48
drwxrwxr-x 3 codespace codespace 4096 Jan 5 16:16 .
drwxr-x--- 1 codespace codespace 4096 Jan 5 15:52 ..
drwxr-xr-x 3 codespace codespace 4096 Jan 5 16:05 .terraform
-rw-r--r-- 1 codespace codespace 2422 Jan 5 15:51 .terraform.lock.hcl
-rwxrwxr-x 1 codespace codespace 422 Jan 5 15:48 create-login-profile.sh
-rw-rw-r-- 1 codespace codespace 50 Jan 5 16:10 locals.tf
-rw-rw-r-- 1 codespace codespace 2288 Jan 5 16:16 main.tf
-rw-rw-r-- 1 codespace codespace 0 Jan 5 16:05 terraform.tfstate
-rw-rw-r-- 1 codespace codespace 6882 Jan 5 16:05 terraform.tfstate.backup
-rw-rw-r-- 1 codespace codespace 167 Jan 5 16:11 users.csv
-rw-rw-r-- 1 codespace codespace 154 Jan 5 15:47 variables.tf