CLOUD COMPUTING

LAB 13

Fatima Jinnah
Women University

Submitted To:

Engr. Shoaib

Submitted By:

Komal Kashif

BSE V-A

2023-BSE-031

Task 1 — Create IAM Group and Output Details

task1_project_directory.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ mkdir -p ~/Lab13
cd ~/Lab13
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13/Lab13 (main) $
```

task1_file_created.png

```
● @KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13/Lab13 (main) $ ls
  main.tf
```

task1_main_tf.png

```
Task > main.tf > output "group_details"
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_iam_group" "developers" {
```

task1_terraform_init.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform init

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.
```

task1_terraform_apply.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13/Lab13 (main) $ terraform apply -auto-approve

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA5WPGPNRIWSOVQPNE5"
```
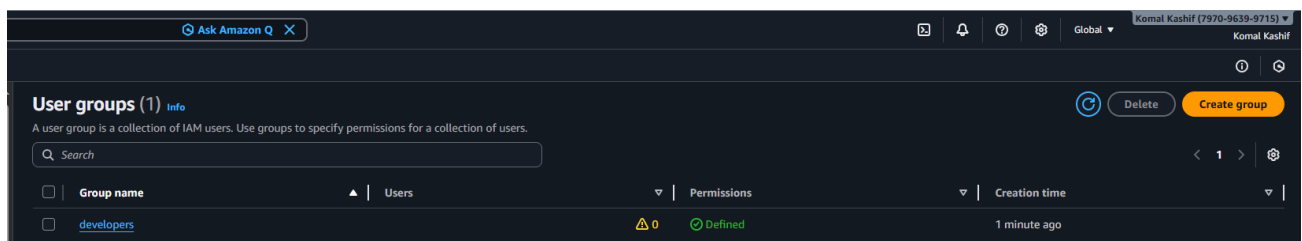
task1_terraform_output.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13/Lab13 (main) $ terraform
group_details = {
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA5WPGPNRIWSOVQPNE5"
```

task1_aws_console_group.png

Task 2 — Create IAM User with Group Membership

task2_main_tf_user.png


```
Task > main.tf > output "user_details"
output "group_details" {
}

resource "aws_iam_user" "lb" {
  name = "loadbalancer"
  path = "/users/"
  force_destroy = true
  tags = {
    DisplayName = "Load Balancer"
```

task2_terraform_apply.png


```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13/Lab13 (main) $ terraform apply -auto-approve
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA5WPGPNRIWSOVQPNE5"
}
user_details = {
  "unique_id" = "AIDA5WPGPNRI5E5P3HF3E"
  "user_arn" = "arn:aws:iam::941618064465:user/users/loadbalancer"
  "user_name" = "loadbalancer"
```

task2_terraform_output.png


```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform output
group_details = {
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
  "unique_id" = "AGPA5WPGPNRIWSOVQPNE5"
}
user_details = {
  "unique_id" = "AIDA5WPGPNRI5E5P3HF3E"
  "user_arn" = "arn:aws:iam::941618064465:user/users/loadbalancer"
  "user_name" = "loadbalancer"
```

task2_aws_console_user.png

task2_aws_console_user_groups.png

Task 3 — Attach Policies to IAM Group

task3_main_tf_policies.png

```
Task > main.tf > resource "aws_iam_group_policy_attachment" "change_password"
output "user_details" {
}

resource "aws_iam_group_policy_attachment" "developer_ec2_fullaccess" {
    group = aws_iam_group.developers.name
    policy_arn = "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
```

task3_terraform_apply.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform apply -auto-approve
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
```

task3_aws_console_policies.png

Task 4 — Create Login Profile for IAM User

task4_variables_tf.png

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

task4_create_login_script.png

```
Task > $ create-login-profile.sh
#!/usr/bin/env bash
set -euo pipefail

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

task4_chmod_script.png

task4_main_tf_login_profile.png

```
resource "null_resource" "create_login_profile" {
  triggers = {
    password_hash = sha256(var.iam_password)
    user          = aws_iam_user.lb.name
```
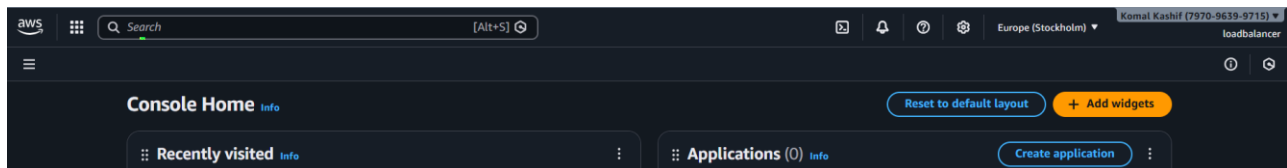
task4_terraform_apply.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform apply -auto-approve

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

Outputs:

group_details = {
  "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
  "group_name" = "developers"
```

task4_aws_cli_verify.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ aws iam get-login-p
file --user-name loadbalancer
{
    "LoginProfile": {
        "UserName": "loadbalancer",
        "CreateDate": "2026-01-28T08:31:30+00:00",
        "PasswordResetRequired": true
```

task4_aws_console_login.png

Task 5 — Generate Access Keys for IAM User

task5_main_tf_access_keys.png

```
Task > 🐙 main.tf > ...
# Access key for IAM user
resource "aws_iam_access_key" "lb_access_key" {
  user = aws_iam_user.lb.name
```

task5_terraform_apply.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform apply -auto-approve -var="iam_pas
sword=MySecurePass123!"
 Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

task5_terraform_output.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform output
access_key_id = "AKIA5WPGPNRIYT2DNYFM"
access_key_secret = <sensitive>
group_details = {
   "group_arn" = "arn:aws:iam::941618064465:group/groups/developers"
   "group_name" = "developers"
   "unique_id" = "AGPA5WPGPNRIWSOVQPNE5"
}
user_details = {
   "unique_id" = "AIDA5WPGPNRI5E5P3HF3E"
   "user_arn" = "arn:aws:iam::941618064465:user/users/loadbalancer"
   "user_name" = "loadbalancer"
```

task5_tfstate_secret.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ cat terraform.tfstate | grep -A 10 "access_
key_secret"
    "access_key_secret": {
     "value": "O9BbFIWKW7J5vRngwSzuIVZTiMkUXFtCMoBc9122",
     "type": "string",
     "sensitive": true
    },
    "group_details": {
     "value": {
       "group_arn": "arn:aws:iam::941618064465:group/groups/developers",
       "group_name": "developers",
       "unique_id": "AGPA5WPGPNRIWSOVQPNE5"
```

Task 6 — Implement Terraform Remote State with S3

task6_main_tf_backend.png

```
terraform {
  backend "s3" {
    bucket = "myapp-s3-bucket-laab13"
    key     = "myapp/terraform.tfstate"
    region = "me-central-1"
    encrypt = true
    use_lockfile = true
```

task6_terraform_init_migrate.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform init -migrate-state
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.
```

task6_terraform_apply.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform apply -auto-approve -var="iam_pas
sword=MySecurePass123!"
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
```

task6_local_state_backup.png

```
@KomalKashif ➜/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform ls -la terraform.tfstate*
-rw-rw-rw- 1 vscode vscode    0 Jan 28 09:02 terraform.tfstate
-rw-rw-rw- 1 vscode vscode 6882 Jan 28 09:02 terraform.tfstate.backup
```

task6_terraform_destroy.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform destroy -auto-approve
aws_iam_group_policy_attachment.developer_ec2_fullaccess: Destruction complete after 1s
aws_iam_user_group_membership.lb_membership: 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 1s
aws_iam_user.lb: Destruction complete after 3s

Destroy complete! Resources: 7 destroyed.
```

Task 7 — Create Multiple Users from CSV File

task7_locals_tf.png

```
Task > ⌄ locals.tf > locals
locals {
  users = csvdecode(file("users.csv"))
}
```

task7_users_csv.png

```
Task > users.csv
user_name
Michael
Dwight
Jim
Pam
Ryan
```

task7_main_tf_multiple_users.png

```
Task > main.tf > resource "aws_iam_access_key" "users_access_keys" > user
# Create multiple IAM users from CSV
resource "aws_iam_user" "users" {
  for_each = { for user in local.users : user.user_name => user }

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

task7_terraform_init.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform init

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.
```

task7_terraform_apply.png

```
@KomalKashif →/workspaces/CC_KomalKashif_031-Lab-13 (main) $ terraform apply -auto-approve -var="iam_pas
sword=MySecurePass123!"

Apply complete! Resources: 107 added, 0 changed, 0 destroyed.
```
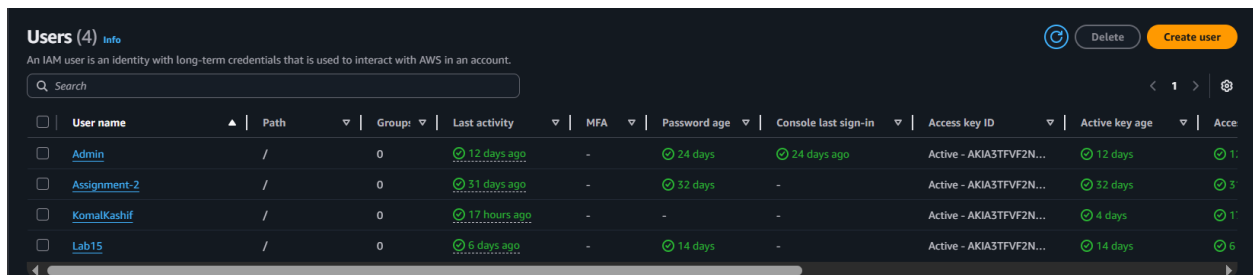
Cleanup:

cleanup_destroy_complete.png

cleanup_aws_console_users_deleted.png
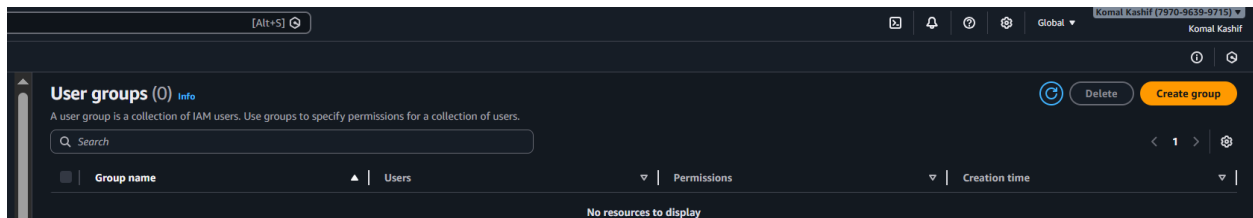
These are just the previous users which are left.

cleanup_aws_console_group_deleted.png

cleanup_final_files.png