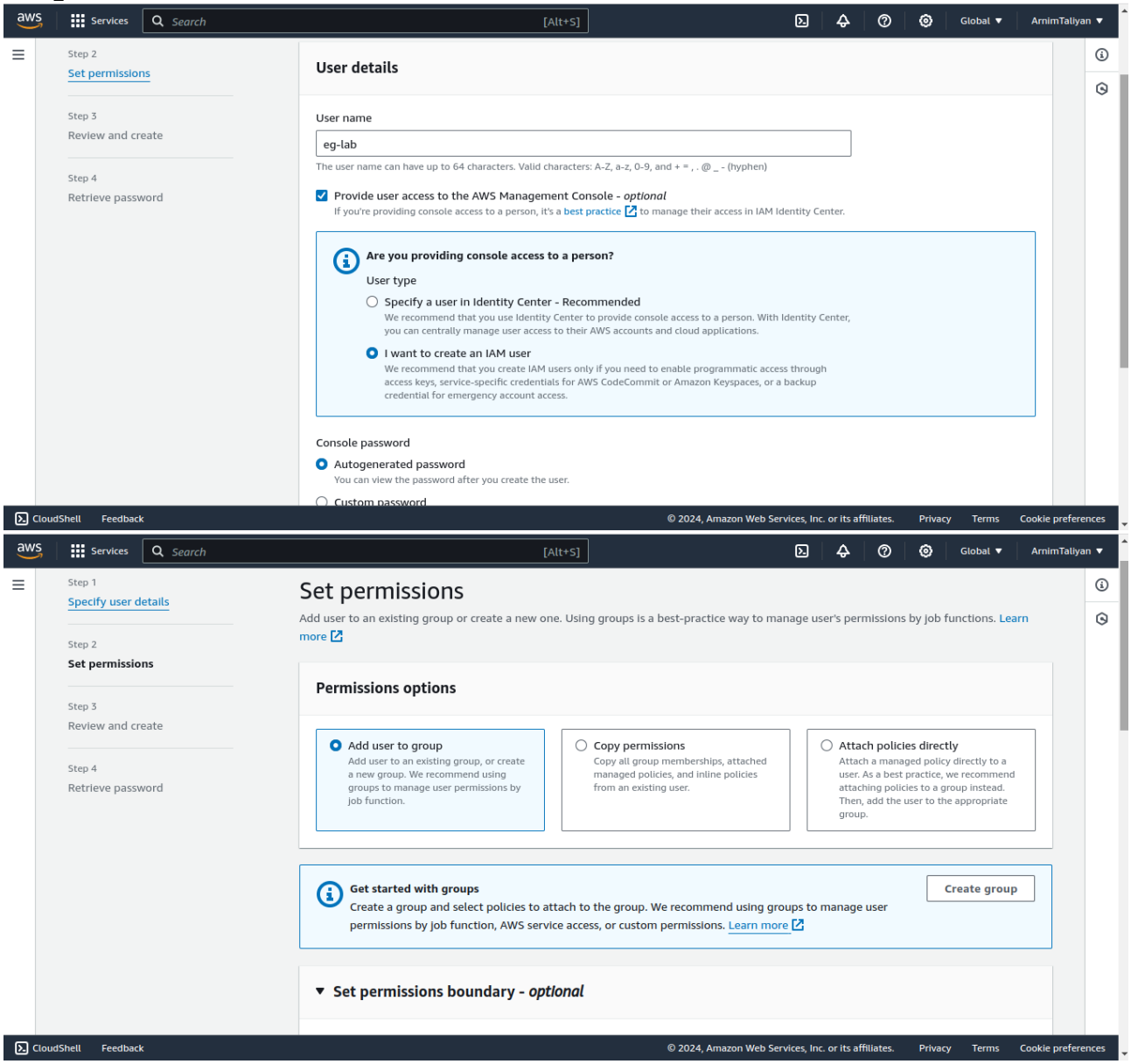
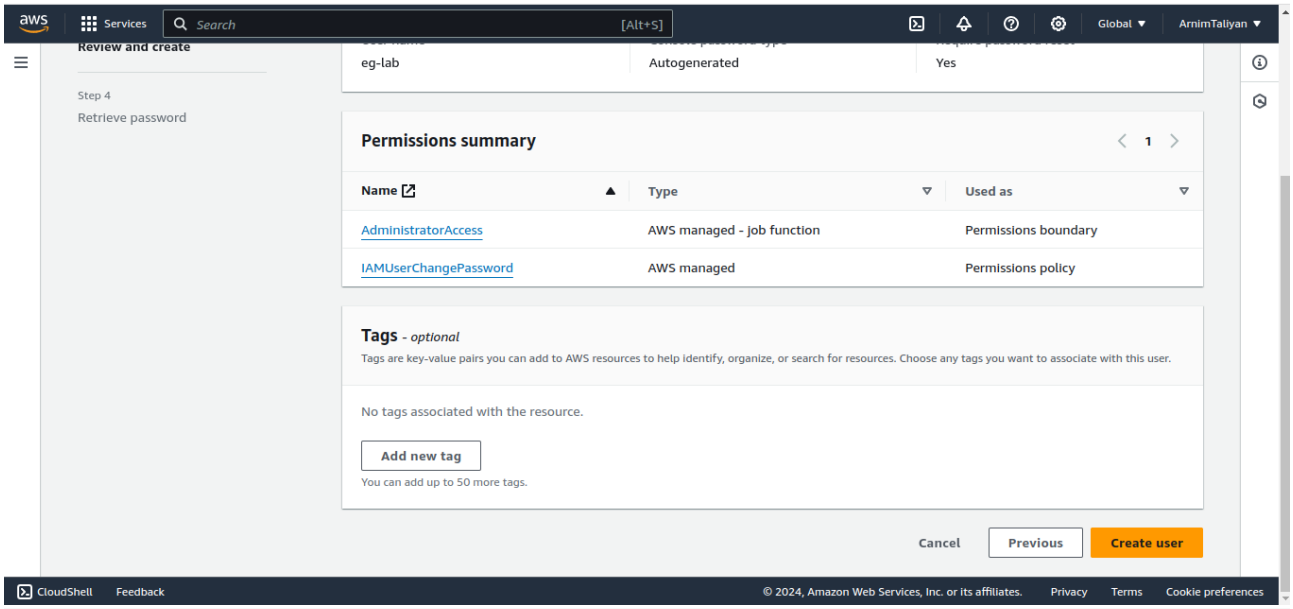
LAB-2
Terraform AWS Provider and IAM User Setting

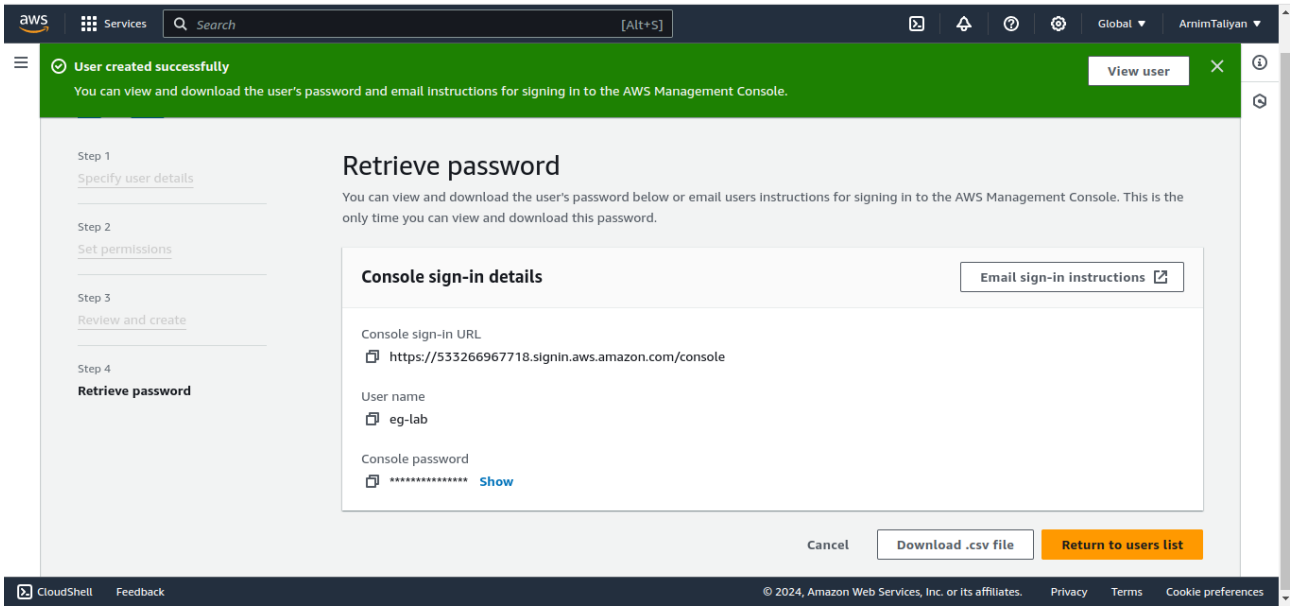
Step 1: Create a new directory

```
arnim_taliyan@device:~$ mkdir terraform
arnim_taliyan@device:~$ cd terraform
arnim_taliyan@device:~/terraform$
```

Step 2: IAM user




aws
Services
Search
[Alt+S]
Global
ArnimTaliyan
Review and create
Step 4
Retrieve password
eg-lab
Autogenerated
Yes
Permissions summary
1
Name
Type
Used as
AdministratorAccess
AWS managed - job function
Permissions boundary
IAMUserChangePassword
AWS managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
Global
ArnimTaliyan
User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.
Console sign-in details
Email sign-in instructions
Console sign-in URL
https://533266967718.signin.aws.amazon.com/console
User name
eg-lab
Console password
****
Show
Cancel
Download .csv file
Return to users list
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 3: Create Terraform configuration file

main.tf × instance.tf

main.tf > provider "aws"

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.32.1"
    }
  }
}
provider"aws"{

    region = "ap-south-1"
    access_key = "AK[redacted]W5FJD"
    secret_key = "7H[redacted]hbV"

}
```

Step 4: Intitalize Terraform

```
arnim_taliyan@device:~/Desktop/terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.32.1"...
- Installing hashicorp/aws v5.32.1...
- Installed hashicorp/aws v5.32.1 (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.
arnim_taliyan@device:~/Desktop/terraform$
```