Home / AWS / Guided Lab / Create IAM Users and Groups using Terraform

Create IAM Users and Groups using Terraform

Level: **Fundamental**

Identity And Access Management   Amazon Web Services   Terraform

WHIZLABS   0   K

0h 29m 47s left   MIN

End Lab

Open Console

Validation

Lab Credentials

User Name

Whiz_User_80425.97018122

Password

5596d825-d77a-4836-b694-f25603763d5d

Access Key

AKIAW5SECW7OHEJCEAPI

Secret Key

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

Lab Resources

No Lab Resources Found

Support Documents

1. FAQs and Troubleshooting

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share

Lab Overview | Lab Steps | Lab Validation

Cloud Administrator

Security, Management & Governance, Infrastructure

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   - Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button

3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

Task 2: Setup Visual Studio Code

1. Open the visual studio code.
2. If you have already installed and using Visual studio code, open a new window.
3. A new window will open a new file and release notes page (only if you have installed or updated Visual Studio Code recently). Close the Release notes tab.

4. Open Terminal by selecting View from the Menu bar and choose Terminal.

5. It may take up to 2 minutes to open the terminal window.

6. Once the terminal is ready, let us navigate to the Desktop.

```
cd Desktop
```

7. Create a new folder by running the below command.

```
mkdir task_10121
```

8. Change your present working directory to use the newly created folder by running the below command:

```
cd task_10121
```

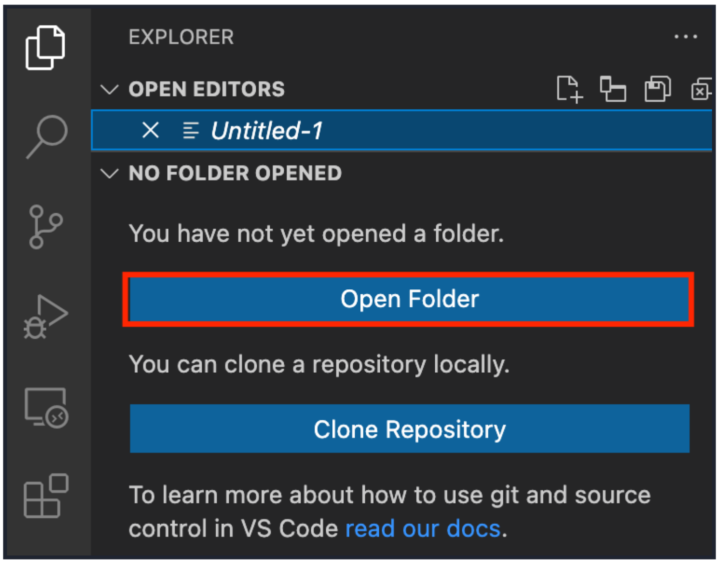
9. Get the location of the present working directory by running the below command:

```
pwd
```

10. Note down the location, as you will open the same in the next steps.

11. Now click on the first icon Explorer present on the left sidebar.

12. Click on the button called Open folder and navigate to the location of folder **task_10121**.



You have not yet opened a folder.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

13. (Optional) Click on Authorize button for allowing Visual Studio Code to use the task_10121 folder. This will only be asked when you have been using Visual Studio code for a while as you are allowing a new folder to be accessed by VSC.
14. Visual Studio Code is now ready to use.

Task 3: Create a variable file

In this task, you will create variable files where you will declare all the global variables with a short description and a default value.

1. To create a variable file, expand the folder **task_10121** and click on the **New File** icon to add the file.
2. Name the file as **variables.tf** and press **Enter** to save it.
3. **Note:** Don't change the location of the new file, keep it default, i.e. inside the **task_10121** folder**.**
4. Paste the below contents in **variables.tf** file.

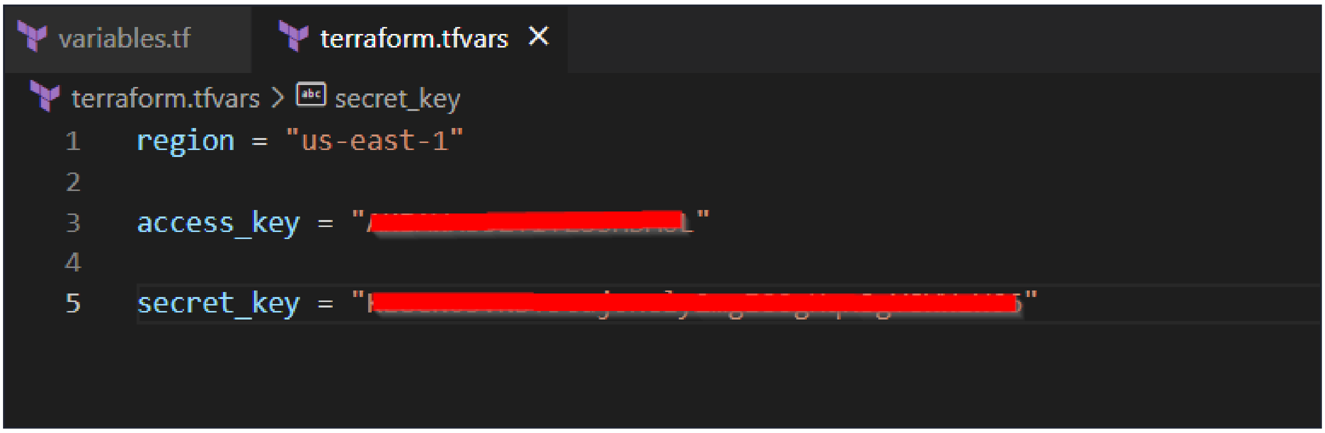
```
variable "access_key" {
    description = "Access key to AWS console"
}
variable "secret_key" {
    description = "Secret key to AWS console"
}
variable "region" {
    description = "AWS region"
}
```

5. In the above content, you are declaring a variable called, access_key, secret_key, and region with a short description of all 3.
6. After pasting the above contents, save the file by pressing **ctrl + S**.
7. Now expand the folder **task_10121** and click on the **New File** icon to add the file.
8. Name the file as **terraform.tfvars** and press **Enter** to save it.
9. Paste the below content into the **terraform.tfvars** file.

```
region = "us-east-1"
access_key = "<YOUR AWS CONSOLE ACCESS ID>"
secret_key = "<YOUR AWS CONSOLE SECRET KEY>"
```

10. In the above code, you are defining the dynamic values of variables declared earlier.
11. Replace the values of access_key and secret_key by copying from the lab page.

12. After replacing the values of access_key and secret_key, save the file by pressing Ctrl + S.



```
region = "us-east-1"

access_key = "[redacted]"

secret_key = "[redacted]"
```

Task 4: Create IAM Users in main.tf file

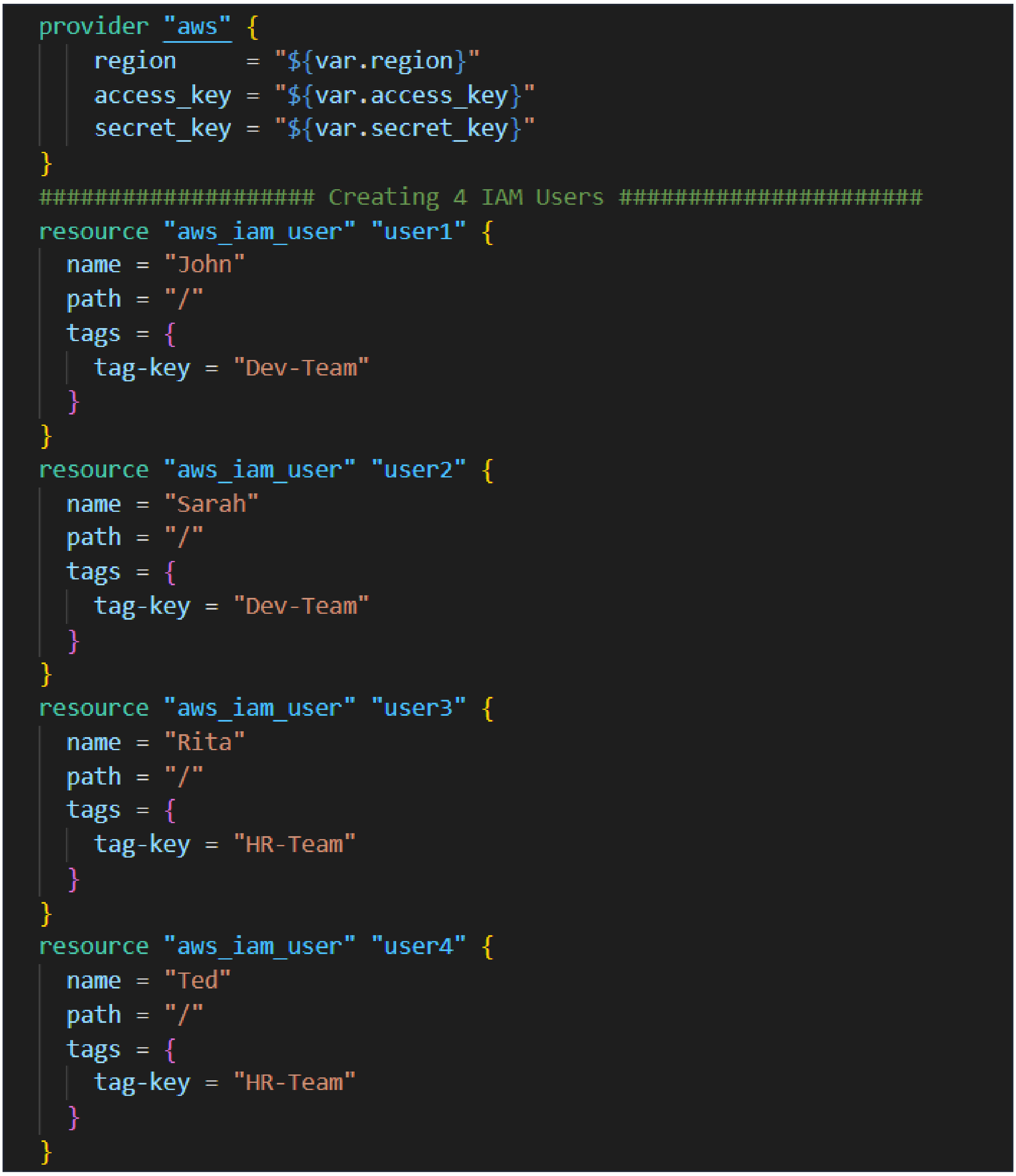
In this task, you will create a **main.tf** file where you will add details of the provider and resources.

1. To create a **main.tf** file, expand the folder **task_10121** and click on the **New File** icon to add the file.
2. Name the file as **main.tf** and press **Enter** to save it.
3. Paste the below content into the **main.tf** file.

```
provider "aws" {
    region     = "${var.region}"
    access_key = "${var.access_key}"
    secret_key = "${var.secret_key}"
}
```

4. In the above code, you are defining the provider as aws.
5. Next, we want to tell Terraform to create 4 IAM users
6. To create IAM Users Paste the below content into the **main.tf** file after the provider

```
#################### Creating 4 IAM Users ######################
resource "aws_iam_user" "user1" {
  name = "John"
  path = "/"
  tags = {
    tag-key = "Dev-Team"
  }
}
resource "aws_iam_user" "user2" {
  name = "Sarah"
  path = "/"
  tags = {
    tag-key = "Dev-Team"
  }
}
resource "aws_iam_user" "user3" {
  name = "Rita"
  path = "/"
  tags = {
    tag-key = "HR-Team"
  }
}
resource "aws_iam_user" "user4" {
  name = "Ted"
  path = "/"
  tags = {
    tag-key = "HR-Team"
  }
}
```



```
provider "aws" {
    region     = "${var.region}"
    access_key = "${var.access_key}"
    secret_key = "${var.secret_key}"
}
#################### Creating 4 IAM Users ######################
resource "aws_iam_user" "user1" {
  name = "John"
  path = "/"
  tags = {
    tag-key = "Dev-Team"
  }
}
resource "aws_iam_user" "user2" {
  name = "Sarah"
  path = "/"
  tags = {
    tag-key = "Dev-Team"
  }
}
resource "aws_iam_user" "user3" {
  name = "Rita"
  path = "/"
  tags = {
    tag-key = "HR-Team"
  }
}
resource "aws_iam_user" "user4" {
  name = "Ted"
  path = "/"
  tags = {
    tag-key = "HR-Team"
  }
}
```

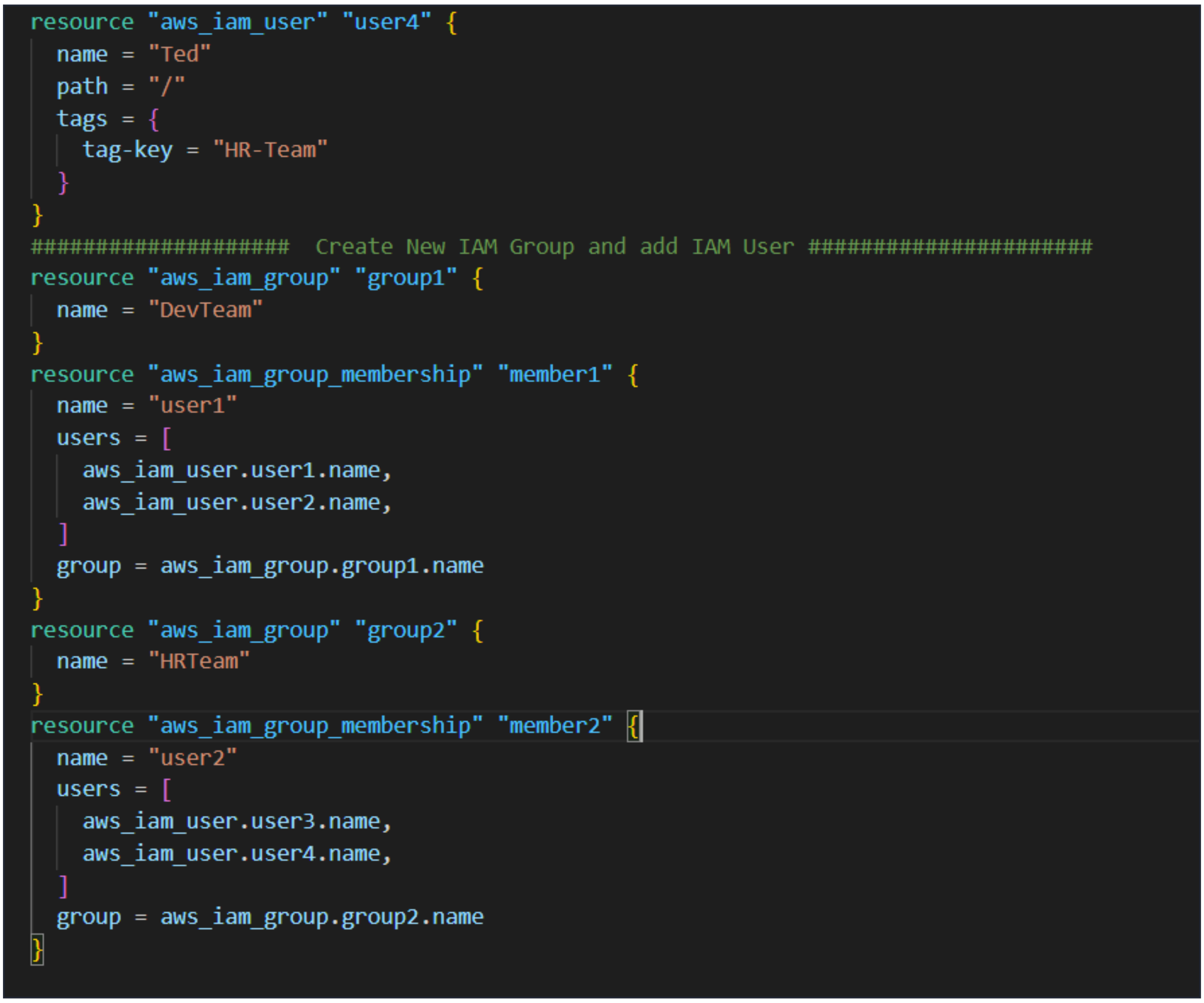
7. This code will create four IAM users in your AWS account with the specified names, paths, and tags.

Task 5: Create IAM group and add IAM users in main.tf file

In this task, you will create 2 IAM groups and add IAM users in main.tf file

1. To create IAM groups add another block of code just below the IAM user code into the **main.tf** file

```
####################  Create New IAM Group and add IAM User
######################
resource "aws_iam_group" "group1" {
  name = "DevTeam"
}
resource "aws_iam_group_membership" "member1" {
  name = "user1"
  users = [
    aws_iam_user.user1.name,
    aws_iam_user.user2.name,
  ]
  group = aws_iam_group.group1.name
}
resource "aws_iam_group" "group2" {
  name = "HRTeam"
}
resource "aws_iam_group_membership" "member2" {
  name = "user2"
  users = [
    aws_iam_user.user3.name,
    aws_iam_user.user4.name,
  ]
  group = aws_iam_group.group2.name
}
```



```
resource "aws_iam_user" "user4" {
  name = "Ted"
  path = "/"
  tags = {
    tag-key = "HR-Team"
  }
}
####################  Create New IAM Group and add IAM User ######################
resource "aws_iam_group" "group1" {
  name = "DevTeam"
}
resource "aws_iam_group_membership" "member1" {
  name = "user1"
  users = [
    aws_iam_user.user1.name,
    aws_iam_user.user2.name,
  ]
  group = aws_iam_group.group1.name
}
resource "aws_iam_group" "group2" {
  name = "HRTeam"
}
resource "aws_iam_group_membership" "member2" {
  name = "user2"
  users = [
    aws_iam_user.user3.name,
    aws_iam_user.user4.name,
  ]
  group = aws_iam_group.group2.name
}
```

2. This code will create two IAM groups in your AWS account with the specified names and add the specified users to each group.

3. Save the file by pressing **Ctrl + S**.

Task 6: Create an Output file

In this task, you will create an **output.tf** file where you will add details of the provider and resources.

1. To create an **output.tf** file, expand the folder **task_10121** and click on the **New File** icon to add the file.
2. Name the file as **output.tf** and press **Enter** to save it.
3. Paste the below content into the **output.tf** file.

```
output "user1" {
  value= aws_iam_user.user1.arn
}
output "user2" {
  value= aws_iam_user.user2.arn
}
output "user3" {
  value= aws_iam_user.user3.arn
}
output "user4" {
  value= aws_iam_user.user4.arn
}
output "group1" {
  value= aws_iam_group.group1.arn
}
output "group2" {
  value= aws_iam_group.group2.arn
}
```

4. In the above code, we will extract details of resources created to confirm that they are created.

Task 7: Confirm the installation of Terraform by checking the version

1. In the Visual Studio Code, open Terminal by selecting **View** from the Menu bar and choose **Terminal**.
2. If you are not in the newly created folder change your present working directory by running the below command.

```
cd task_10121
```

3. To confirm the installation of Terraform, run the below command to check the version:

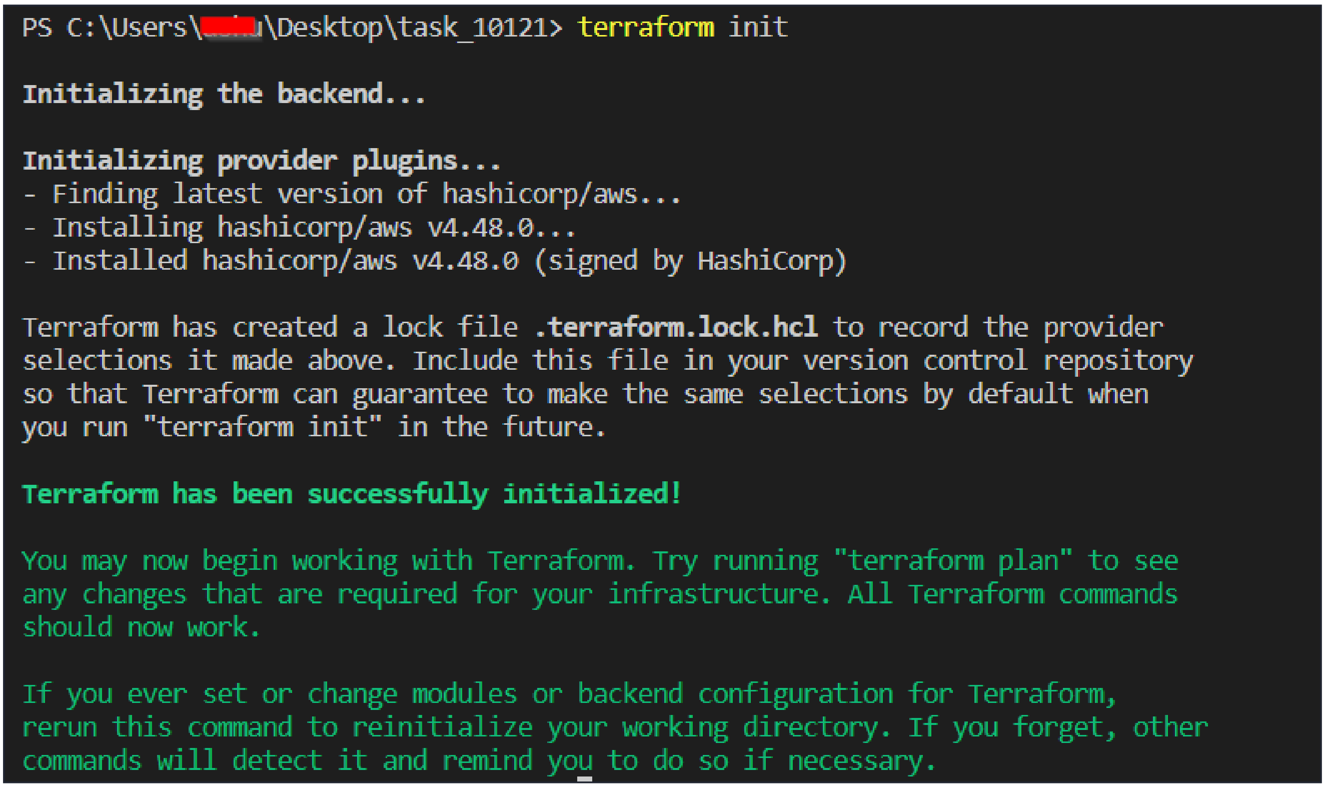
```
terraform version
```

4. If you are getting output as command not found: terraform, this means that terraform is not installed on your system, To install terraform follow the official guide link provided in the Prerequisite section above.

Task 8: Apply terraform configurations

1. Initialize Terraform by running the below command,

```
terraform init
```

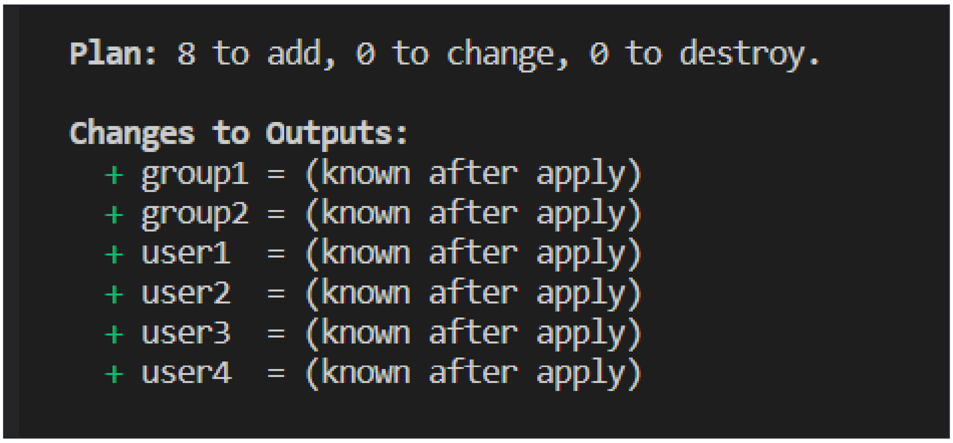


```
PS C:\Users\[redacted]\Desktop\task_10121> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.48.0...
- Installed hashicorp/aws v4.48.0 (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.
```

Note: terraform init will check for all the plugin dependencies and download them if required, this will be used for creating a deployment plan

2. To generate the action plans run the below command,

```
terraform plan
```

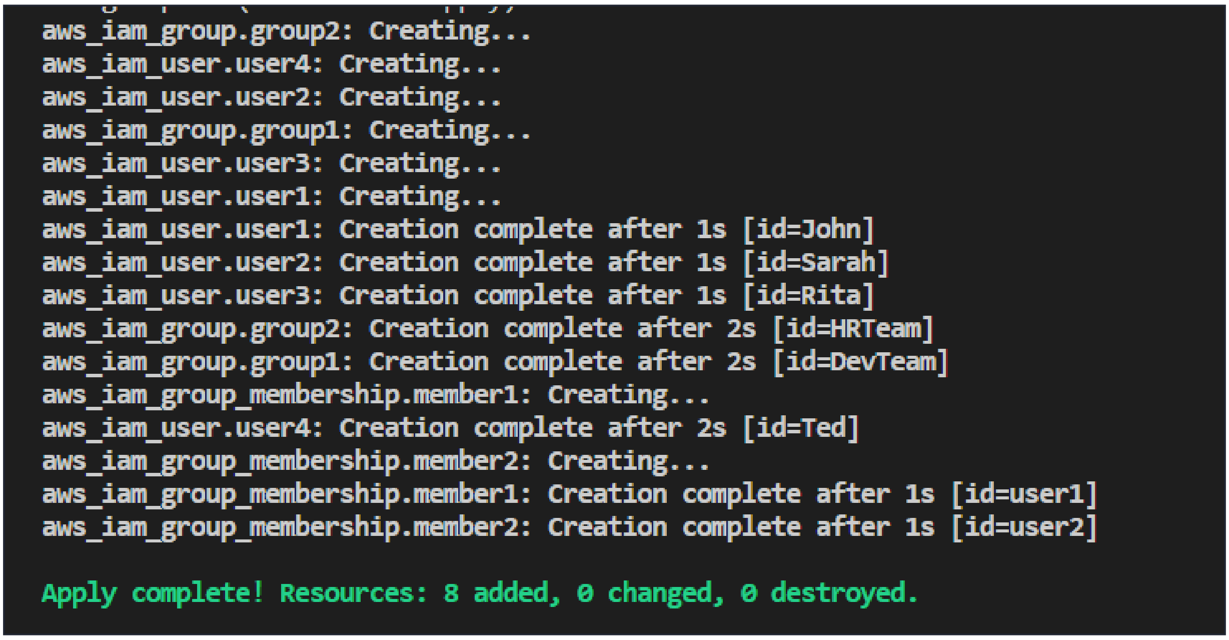


```
Plan: 8 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + group1 = (known after apply)
  + group2 = (known after apply)
  + user1  = (known after apply)
  + user2  = (known after apply)
  + user3  = (known after apply)
  + user4  = (known after apply)
```

3. To create all the resources declared in main.tf configuration file, run the below command:

```
terraform apply
```

4. Approve the creation of all the resources by entering **yes**.



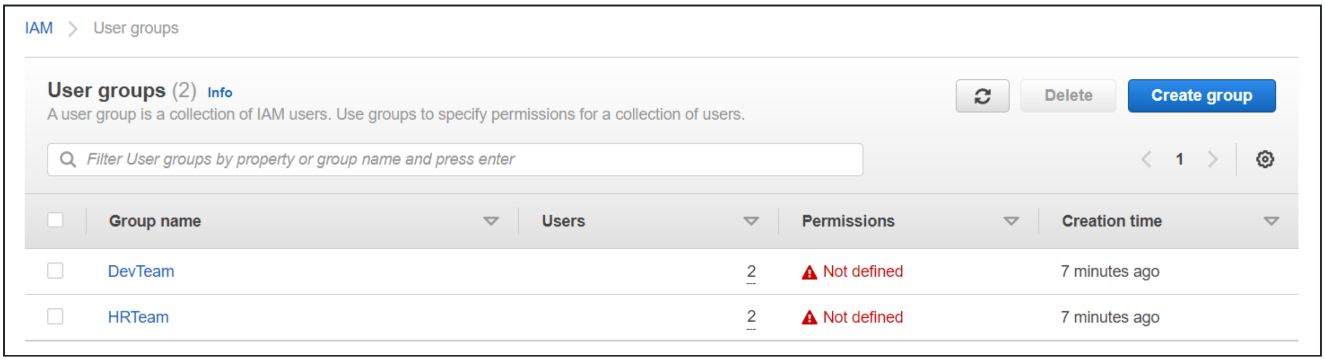
```
aws_iam_group.group2: Creating...
aws_iam_user.user4: Creating...
aws_iam_user.user2: Creating...
aws_iam_group.group1: Creating...
aws_iam_user.user3: Creating...
aws_iam_user.user1: Creating...
aws_iam_user.user1: Creation complete after 1s [id=John]
aws_iam_user.user2: Creation complete after 1s [id=Sarah]
aws_iam_user.user3: Creation complete after 1s [id=Rita]
aws_iam_group.group2: Creation complete after 2s [id=HRTeam]
aws_iam_group.group1: Creation complete after 2s [id=DevTeam]
aws_iam_group_membership.member1: Creating...
aws_iam_user.user4: Creation complete after 2s [id=Ted]
aws_iam_group_membership.member2: Creating...
aws_iam_group_membership.member1: Creation complete after 1s [id=user1]
aws_iam_group_membership.member2: Creation complete after 1s [id=user2]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
```

5. Id's of all the resources created by terraform will be visible there.

Task 9: Check the resources in AWS Console

1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to **IAM** by clicking on **Services** on the top, then click on **IAM** in the **Security, Identity, & Compliance** section.
3. Click on the **Users** on the left navigation panel. You can see the 4 users (John, Rita, Sarah & Ted)created successfully.

4. Click on the **User groups** on the left navigation panel. You can see the groups created successfully.



IAM > User groups

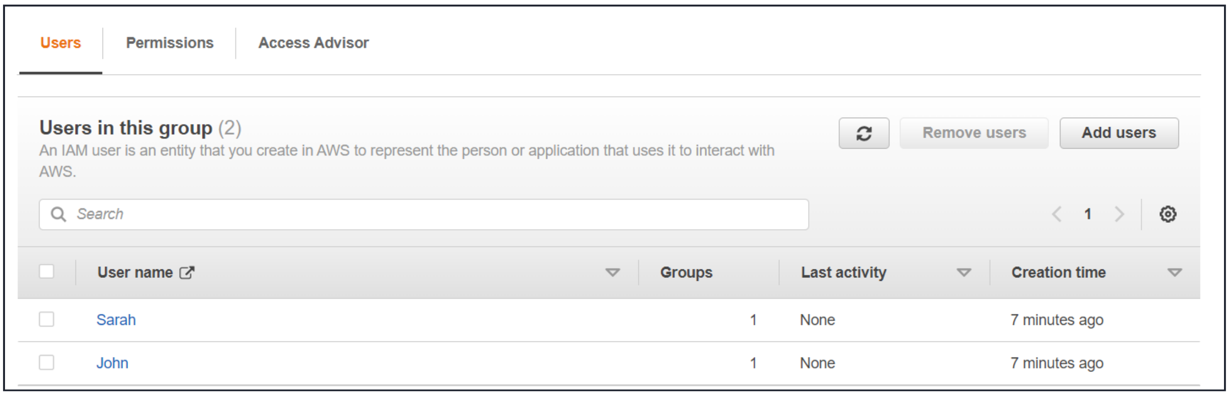
User groups (2) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Delete
Create group
Filter User groups by property or group name and press enter
1

□	Group name	Users	Permissions	Creation time
□	DevTeam	2	Not defined	7 minutes ago
□	HRTeam	2	Not defined	7 minutes ago

5. Select group **DevTeam** you can see 2 users added successfully

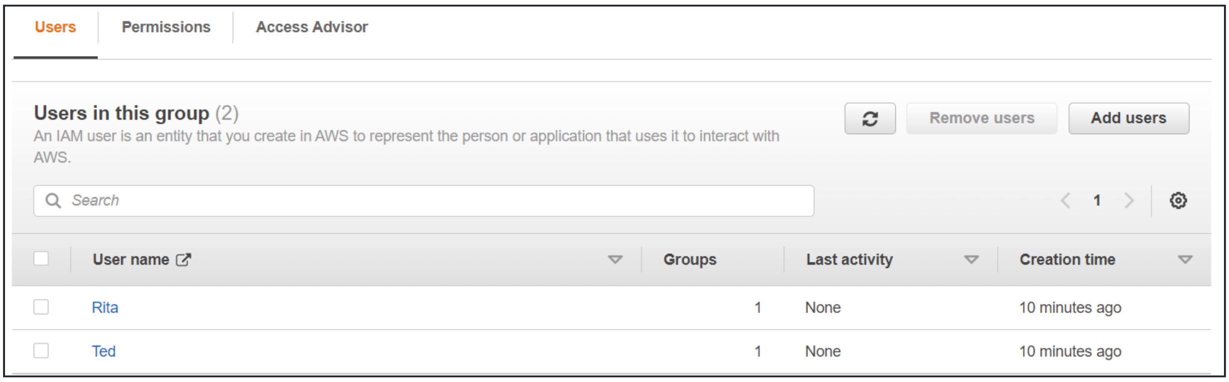


Users in this group (2)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

□	User name	Groups	Last activity	Creation time
□	Sarah	1	None	7 minutes ago
□	John	1	None	7 minutes ago

6. Similarly you can check 2 users added in **HRTeam**



Users in this group (2)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

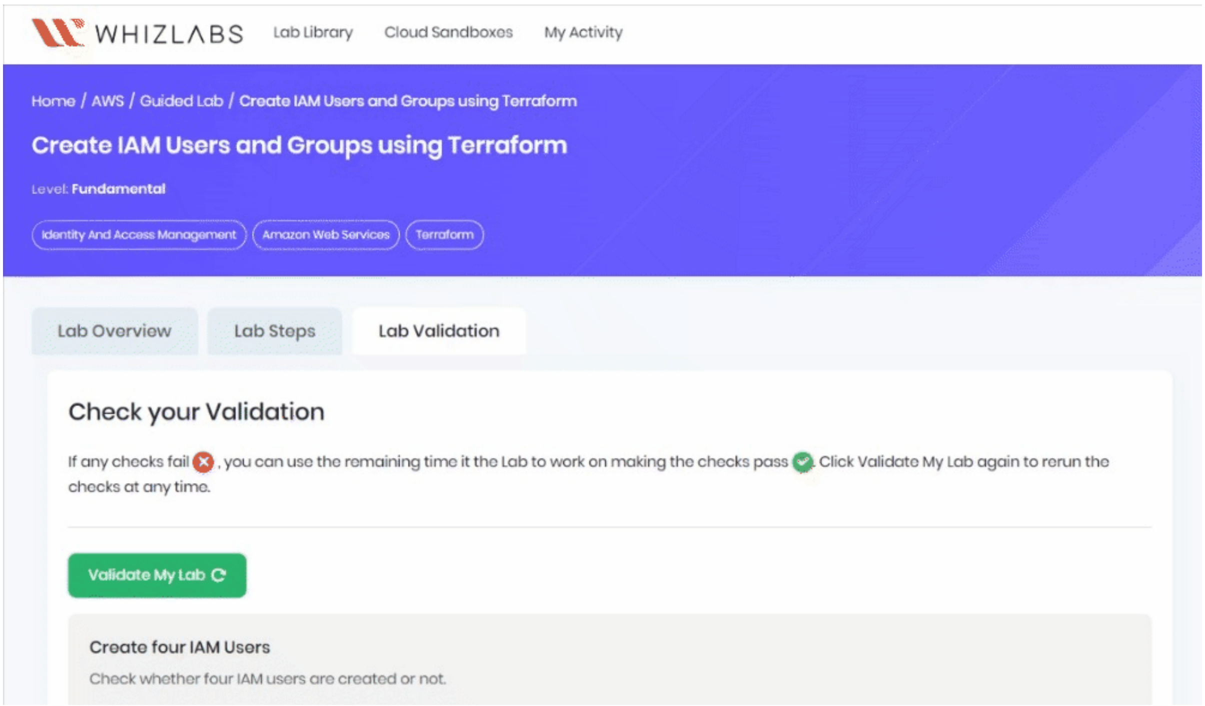
□	User name	Groups	Last activity	Creation time
□	Rita	1	None	10 minutes ago
□	Ted	1	None	10 minutes ago

Do you know?

By using Terraform to manage IAM users and groups, you can ensure consistent and repeatable provisioning of identity and access management across your cloud infrastructure. Terraform's versioning and state management capabilities also contribute to the traceability and maintainability of your IAM configurations, making it easier to collaborate with team members and manage changes in a secure and efficient manner.

Task 10: Validation of the Lab

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



Create IAM Users and Groups using Terraform

Level: Fundamental

Check your Validation

If any checks fail ✗, you can use the remaining time it the Lab to work on making the checks pass ✓. Click Validate My Lab again to rerun the checks at any time.

Create four IAM Users

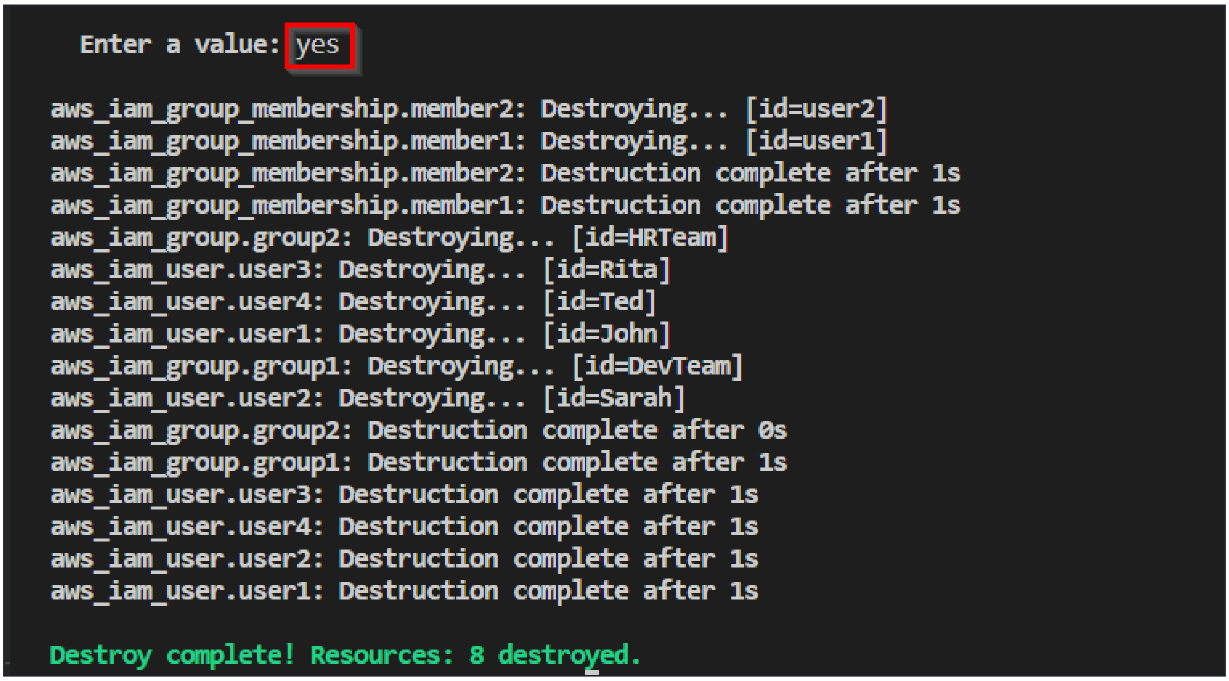
Check whether four IAM users are created or not.

Task 11: Delete AWS Resources

1. To delete the resources, open Terminal again.
2. Run the below command to delete all the resources.

```
terraform destroy
```

3. Approve the creation of all the resources by entering **yes**. You can see the **Destroy complete!** message.



```
  Enter a value: yes

aws_iam_group_membership.member2: Destroying... [id=user2]
aws_iam_group_membership.member1: Destroying... [id=user1]
aws_iam_group_membership.member2: Destruction complete after 1s
aws_iam_group_membership.member1: Destruction complete after 1s
aws_iam_group.group2: Destroying... [id=HRTeam]
aws_iam_user.user3: Destroying... [id=Rita]
aws_iam_user.user4: Destroying... [id=Ted]
aws_iam_user.user1: Destroying... [id=John]
aws_iam_group.group1: Destroying... [id=DevTeam]
aws_iam_user.user2: Destroying... [id=Sarah]
aws_iam_group.group2: Destruction complete after 0s
aws_iam_group.group1: Destruction complete after 1s
aws_iam_user.user3: Destruction complete after 1s
aws_iam_user.user4: Destruction complete after 1s
aws_iam_user.user2: Destruction complete after 1s
aws_iam_user.user1: Destruction complete after 1s

Destroy complete! Resources: 8 destroyed.
```

Completion and Conclusion

- You have successfully set up the Visual Studio Code editor.
- You have successfully created variables.tf and terraform.tfvars files.
- You have successfully created 4 IAM Users using terraform
- You have successfully created IAM groups and added users using terraform
- You have successfully created output.tf
- You have successfully executed the terraform configuration commands to create the resources.
- You have successfully checked all the resources created by opening the Console.
- You have successfully deleted all the resources.

End Lab

1. Sign out of AWS Account.
2. You have successfully completed the lab.
3. Once you have completed the steps, click on **End Lab** from your whizlabs dashboard.