WHIZLABS 0 K

Home / AWS / Guided Lab / Create a 2 tier Amazon VPC using Terraform

Create a 2 tier Amazon VPC using Terraform

Level: **Fundamental**

Amazon VPC Amazon Web Services Terraform

0h 59m 39s left

End Lab

Open Console

Validation

Lab Credentials

User Name

Whiz_User_80425.81096617

Password

72a4c31f-56da-4b5b-8ab5-d093ecef7815

Access Key

AKIAWQUUZOE4GJYBJNMO

Secret Key

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

Lab Resources

1. Download Zip

Support Documents

1. FAQs and Troubleshooting

2. Labs - Instructions and Guidelines

Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback

Share

Lab Overview

Lab Steps

Lab Validation

Cloud Architect

Networking, 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 **Username** 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

In this task, we are going to set up the Visual Studio Code editor as the development environment. Visual Studio Code provides a user-friendly interface and useful features for writing and managing code. This task ensures that users have the necessary tools in place for working with Terraform configuration files.

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_10000_vpc
```

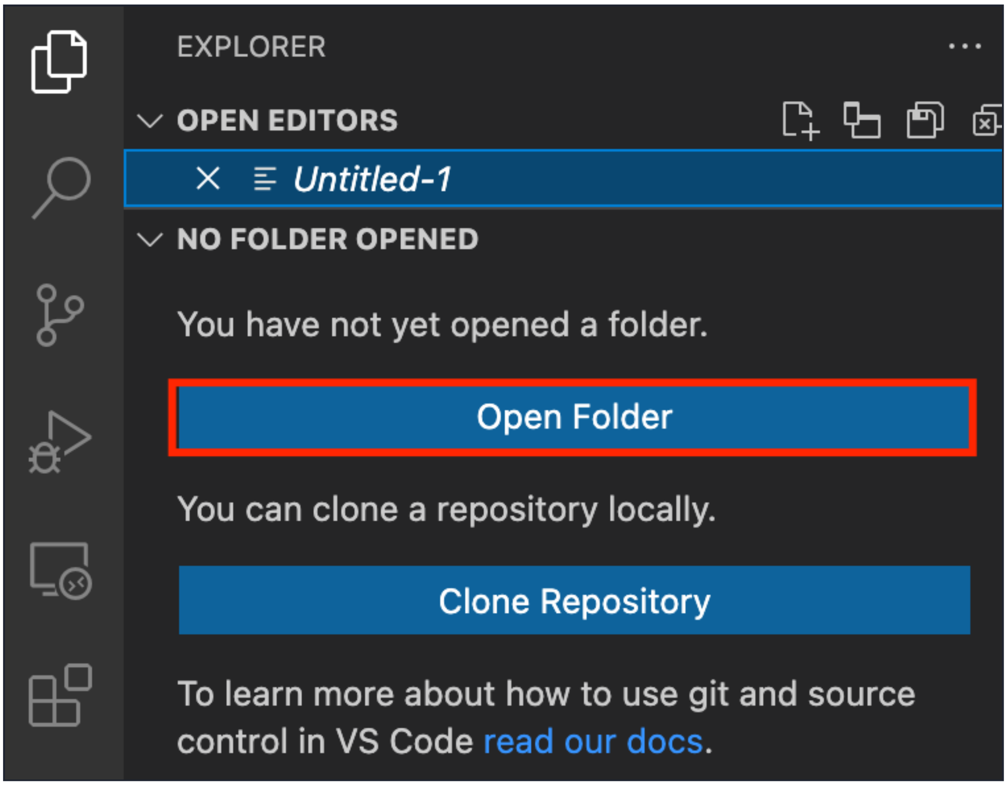
8. Change your present working directory to use the newly created folder by running the below command:

```
cd task_10000_vpc
```

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_10000_vpc**.



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_100012_sns_s3** 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 variables file

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

1. To create variables files, click on the **File** from the menu bar and choose **New file**
2. Press Ctrl + S to save the new file as **variables.tf** and click on the **Save** button after entering the file name.
3. **Note:** Don't change the location of the new file, keep it default, i.e. inside the **task_10000_vpc** 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 = "Region of AWS VPC"
}
```

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 create the **terraform.tfvars** file by selecting **New file** present under **File** in the menu bar.

8. Name the file by pressing Ctrl + S and enter **terraform.tfvars**

9. Paste the below content into **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.

Task 4: Create VPC and its components in main.tf file

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

1. To create **main.tf** file, click on the **File** from the menu bar and choose **New file**

2. Press Ctrl + S to save the new file as **main.tf** and click on the **Save** button after entering the file name.

3. Paste the below content into **main.tf** file.

```
provider "aws" {
     region     = var.region
     access_key = var.access_key
     secret_key = var.secret_key
}
resource "aws_vpc" "vpc" {
    cidr_block = "10.0.0.0/16"
}
```

4. In the above code, you are defining provider as **aws** and defining resource block using **aws_vpc** for creating an Amazon VPC having CIDR Block as 10.0.0.0/24.

5. Define Internet gateway, route table, and subnets in all availability zone. Add the below contents after aws_vpc block.

```
resource "aws_internet_gateway" "gateway" {
    vpc_id = aws_vpc.vpc.id
}
resource "aws_route" "route" {
    route_table_id         = aws_vpc.vpc.main_route_table_id
    destination_cidr_block = "0.0.0.0/0"
    gateway_id             = aws_internet_gateway.gateway.id
}
data "aws_availability_zones" "available" {}

resource "aws_subnet" "main" {
   count                   =
length(data.aws_availability_zones.available.names)
   vpc_id                  = aws_vpc.vpc.id
   cidr_block              = "10.0.${count.index}.0/24"
   map_public_ip_on_launch = true
   availability_zone       =
element(data.aws_availability_zones.available.names, count.index)
}
```

6. In the above code, you are performing the following tasks:

- Creating an Internet gateway and attaching it with the VPC.
- Adding route as 0.0.0.0/0 and destination as an Internet gateway.
- Checking all the availability zones.
- Creating subnets in all availability zones.

7. Save the file by pressing Ctrl + S.

Task 5: Confirm the installation of Terraform by checking the version

1. Open the Terminal by clicking on the **View** from the menu bar and choosing **Terminal**
2. If you are not in the newly created folder, change your present working directory by running the below command

```
cd task_10000_vpc
```

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 6: Apply terraform configurations

1. Make sure you are in the same location where all these 3 files are present, you can check the same using the ls command.
2. Initialize Terraform by running the below command:

```
terraform init
```

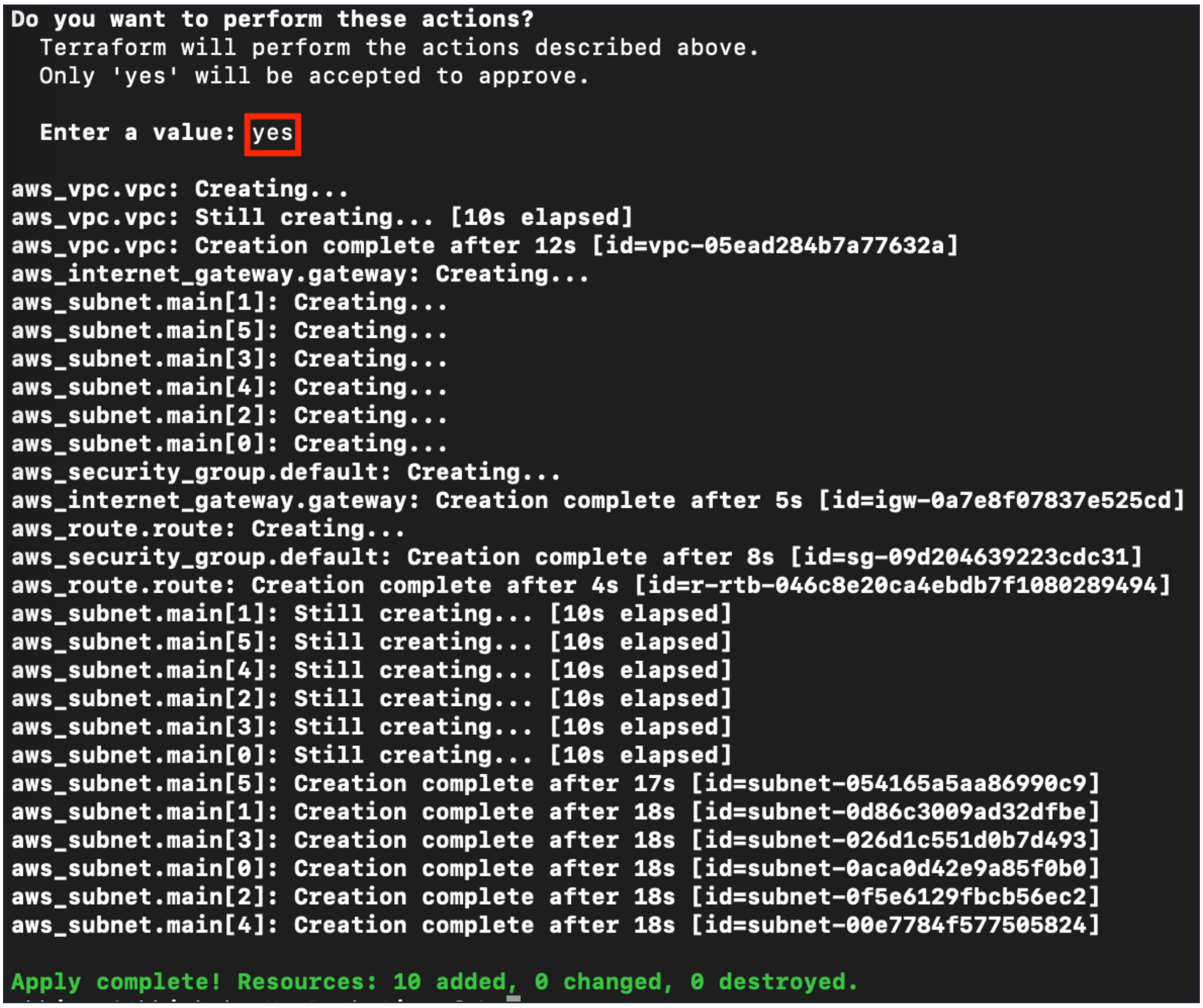
3. **Note:** terraform init will check for all the plugin dependencies and download them if required, this will be used for creating a deployment plan.
4. To generate the action plans, run the below command:

```
terraform plan
```

5. Review the whole generated plan.
6. To create all the resources declared in **main.tf** configuration file, run the below command:

```
terraform apply
```

7. You will be able to see the resources which will be created, approve the creation of all the resources by entering **yes.**
8. It may take up to 2 minutes for the **terraform apply** command to create the resources.
9. Ids of all the resources created by terraform will be visible there.

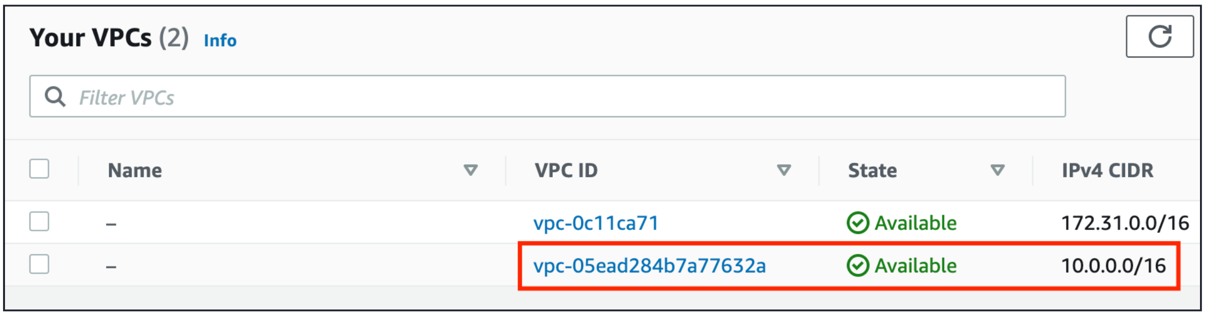


```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.vpc: Creating...
aws_vpc.vpc: Still creating... [10s elapsed]
aws_vpc.vpc: Creation complete after 12s [id=vpc-05ead284b7a77632a]
aws_internet_gateway.gateway: Creating...
aws_subnet.main[1]: Creating...
aws_subnet.main[5]: Creating...
aws_subnet.main[3]: Creating...
aws_subnet.main[4]: Creating...
aws_subnet.main[2]: Creating...
aws_subnet.main[0]: Creating...
aws_security_group.default: Creating...
aws_internet_gateway.gateway: Creation complete after 5s [id=igw-0a7e8f07837e525cd]
aws_route.route: Creating...
aws_security_group.default: Creation complete after 8s [id=sg-09d204639223cdc31]
aws_route.route: Creation complete after 4s [id=r-rtb-046c8e20ca4ebdb7f1080289494]
aws_subnet.main[1]: Still creating... [10s elapsed]
aws_subnet.main[5]: Still creating... [10s elapsed]
aws_subnet.main[4]: Still creating... [10s elapsed]
aws_subnet.main[2]: Still creating... [10s elapsed]
aws_subnet.main[3]: Still creating... [10s elapsed]
aws_subnet.main[0]: Still creating... [10s elapsed]
aws_subnet.main[5]: Creation complete after 17s [id=subnet-054165a5aa86990c9]
aws_subnet.main[1]: Creation complete after 18s [id=subnet-0d86c3009ad32dfbe]
aws_subnet.main[3]: Creation complete after 18s [id=subnet-026d1c551d0b7d493]
aws_subnet.main[0]: Creation complete after 18s [id=subnet-0aca0d42e9a85f0b0]
aws_subnet.main[2]: Creation complete after 18s [id=subnet-0f5e6129fbcb56ec2]
aws_subnet.main[4]: Creation complete after 18s [id=subnet-00e7784f577505824]

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
```

10. Optionally, you can note down the IDs of all the resources.

Task 7: Check the resources in AWS Console

1. Navigate to **VPC** by clicking on **Services** under the Networking **& Content Delivery** section.
2. Click on **Your VPCs** from the left menu.
3. Custom VPC is present.

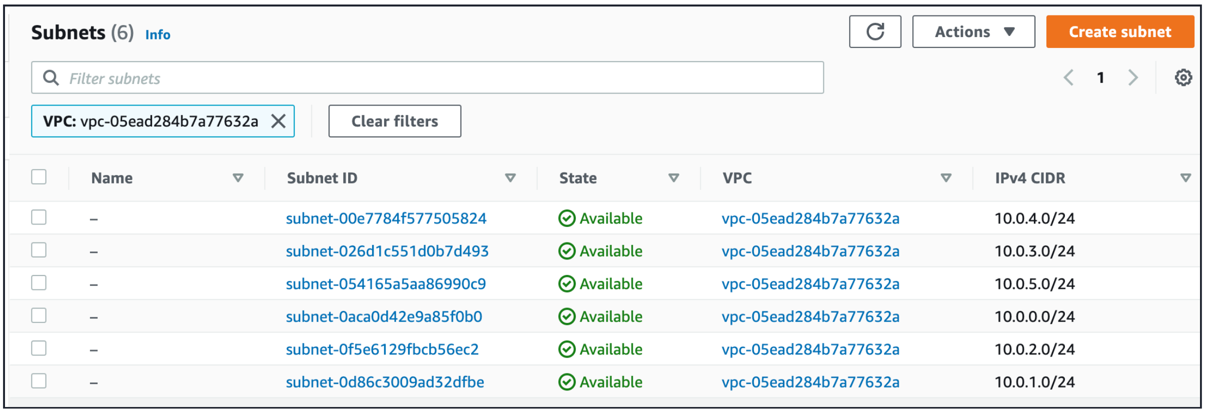


Your VPCs (2) Info

Filter VPCs

□	Name	VPC ID	State	IPv4 CIDR
□	–	vpc-0c11ca71	✓ Available	172.31.0.0/16
□	–	vpc-05ead284b7a77632a	✓ Available	10.0.0.0/16

4. To check Subnets, click on the Subnets option from the left sidebar.
5. All the present subnets are of both the VPC, i.e., Default VPC and Custom VPC.
6. Apply the filter of Custom VPC by clicking on the Search bar and select **VPC** as Properties and **VPC ID of custom VPC** as VPC values.



Subnets (6) Info

VPC: vpc-05ead284b7a77632a

□	Name	Subnet ID	State	VPC	IPv4 CIDR
□	–	subnet-00e7784f577505824	✓ Available	vpc-05ead284b7a77632a	10.0.4.0/24
□	–	subnet-026d1c551d0b7d493	✓ Available	vpc-05ead284b7a77632a	10.0.3.0/24
□	–	subnet-054165a5aa86990c9	✓ Available	vpc-05ead284b7a77632a	10.0.5.0/24
□	–	subnet-0aca0d42e9a85f0b0	✓ Available	vpc-05ead284b7a77632a	10.0.0.0/24
□	–	subnet-0f5e6129fbcb56ec2	✓ Available	vpc-05ead284b7a77632a	10.0.2.0/24
□	–	subnet-0d86c3009ad32dfbe	✓ Available	vpc-05ead284b7a77632a	10.0.1.0/24

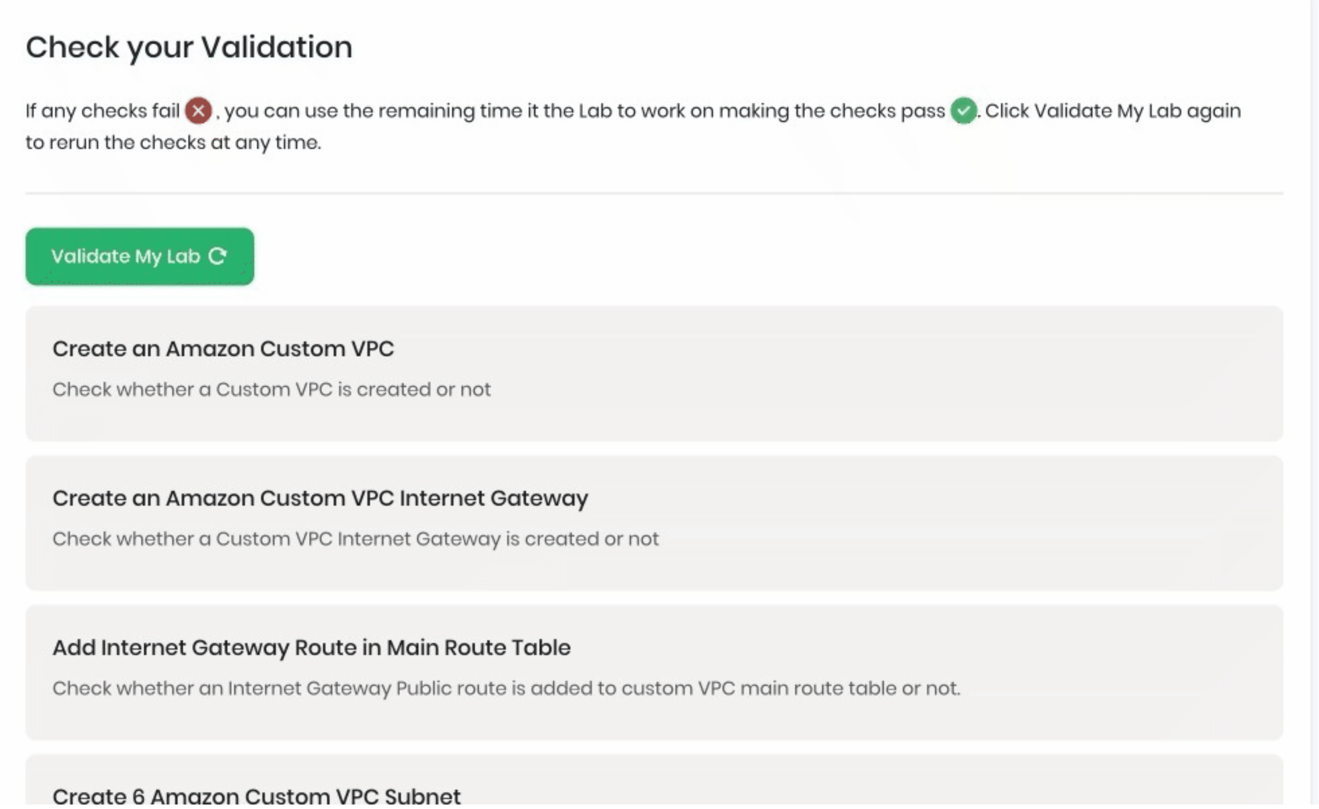
7. **Similarly, you can check Route Tables, Internet Gateways, and Security Groups.**

Do You Know?

VPC is a powerful feature of AWS that allows you to build secure, isolated, and customizable network environments in the cloud, giving you full control over your network architecture and enabling seamless integration with other AWS services.

Task 8: 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 :



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 an Amazon Custom VPC

Check whether a Custom VPC is created or not

Create an Amazon Custom VPC Internet Gateway

Check whether a Custom VPC Internet Gateway is created or not

Add Internet Gateway Route in Main Route Table

Check whether an Internet Gateway Public route is added to custom VPC main route table or not.

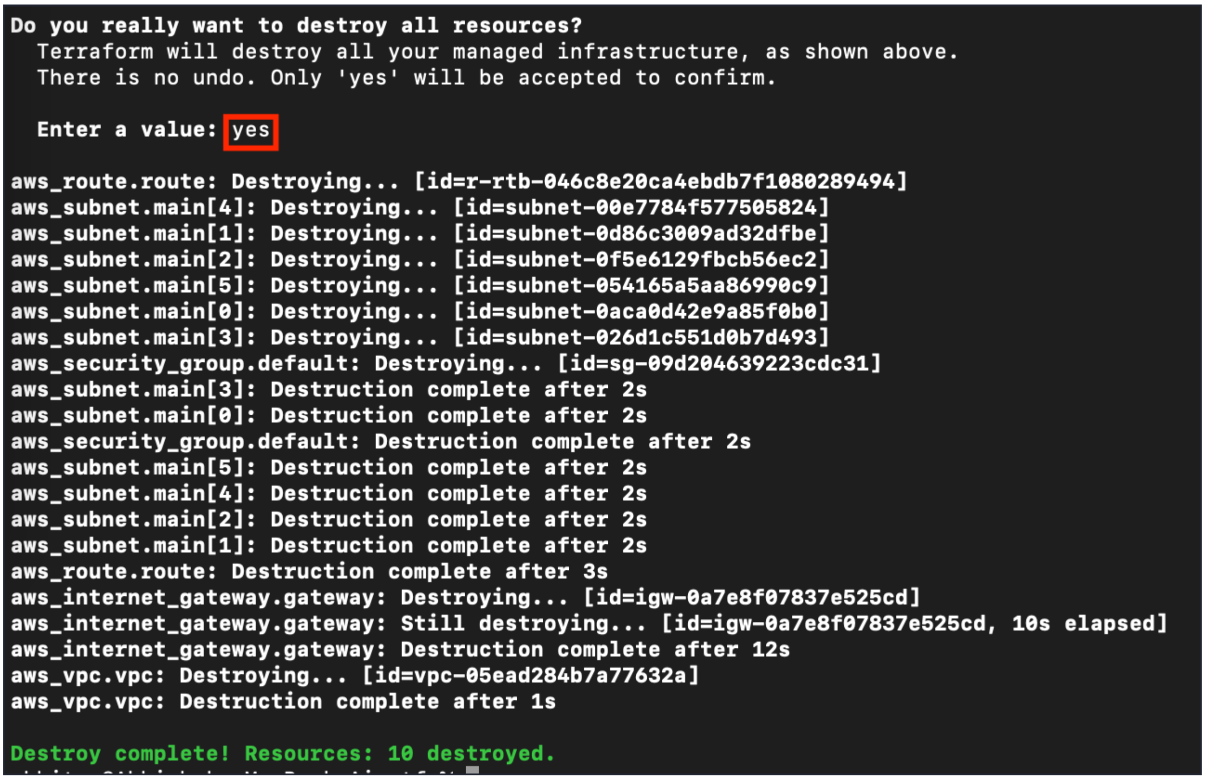
Create 6 Amazon Custom VPC Subnet

Task 9: Delete AWS Resources

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

```
terraform destroy
```

3. Enter **yes** to confirm the deletion.



```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_route.route: Destroying... [id=r-rtb-046c8e20ca4ebdb7f1080289494]
aws_subnet.main[4]: Destroying... [id=subnet-00e7784f577505824]
aws_subnet.main[1]: Destroying... [id=subnet-0d86c3009ad32dfbe]
aws_subnet.main[2]: Destroying... [id=subnet-0f5e6129fbcb56ec2]
aws_subnet.main[5]: Destroying... [id=subnet-054165a5aa86990c9]
aws_subnet.main[0]: Destroying... [id=subnet-0aca0d42e9a85f0b0]
aws_subnet.main[3]: Destroying... [id=subnet-026d1c551d0b7d493]
aws_security_group.default: Destroying... [id=sg-09d204639223cdc31]
aws_subnet.main[3]: Destruction complete after 2s
aws_subnet.main[0]: Destruction complete after 2s
aws_security_group.default: Destruction complete after 2s
aws_subnet.main[5]: Destruction complete after 2s
aws_subnet.main[4]: Destruction complete after 2s
aws_subnet.main[2]: Destruction complete after 2s
aws_subnet.main[1]: Destruction complete after 2s
aws_route.route: Destruction complete after 3s
aws_internet_gateway.gateway: Destroying... [id=igw-0a7e8f07837e525cd]
aws_internet_gateway.gateway: Still destroying... [id=igw-0a7e8f07837e525cd, 10s elapsed]
aws_internet_gateway.gateway: Destruction complete after 12s
aws_vpc.vpc: Destroying... [id=vpc-05ead284b7a77632a]
aws_vpc.vpc: Destruction complete after 1s

Destroy complete! Resources: 10 destroyed.
```

Resources

- To download the Terraform configuration for this lab, click here.

Completion and Conclusion

- You have successfully done the setup of the Visual Studio Code editor.
- You have successfully created **variables.tf** and **terraform.tfvars** file
- You have successfully created a **main.tf** file.
- You have successfully executed the terraform configuration commands to create the resources.
- You have successfully checked all the resources are 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.

About Us Subscription Instructions and Guidelines FAQ’s Contact Us