



Create DynamoDB Table and Insert Items to table using Terraform



0h 58m 50s left



End Lab

Open Console

Validation

Lab Credentials

User Name ⓘ  
Whiz\_User\_80425.31186308

Password ⓘ  
ccfb6797-45aa-4ec5-a3fb-19de37fbf4d7





Access Key ⓘ  
AKIA4GZ7DAACHACW6F6J


Secret Key ⓘ  
JQTIQDvdVEa5r1Qf+1HOICAhlw6kpXrKSc1bCBG3


Lab Resources

No Lab Resources Found

Support Documents







ns

No Support Documents Found

## Need help?



How to use Hands on Lab



Troubleshooting Lab



FAQs

Submit Feedback

Share

Lab Overview

Lab Steps

Lab Validation



Cloud Architect, Cloud Administrator



Database, 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 the 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_10098_dynamodb
```



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

```
cd task_10098_dynamodb
```



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\_10098\_dynamodb**

13. Visual Studio code is now ready to use.

### Task 3: Create a variables file

1. To create a variables file, expand the folder **task\_10094\_sns** 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\_10098\_dynamodb** 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\_10098\_dynamodb** 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_ACCESS_KEY>"
secret_key = "<YOUR_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**.

```
terraform.tfvars
1  region = "us-east-1"
2
3  access_key = "[REDACTED]"
4
5  secret_key = "[REDACTED]"
```

#### Task 4: Create a DynamoDB Table and its components in main.tf file

In this task, we are going to create a DynamoDB Table and its components in `main.tf` file.

1. To create a `main.tf` file, expand the folder **task\_10098\_dynamodb** 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  
}
```

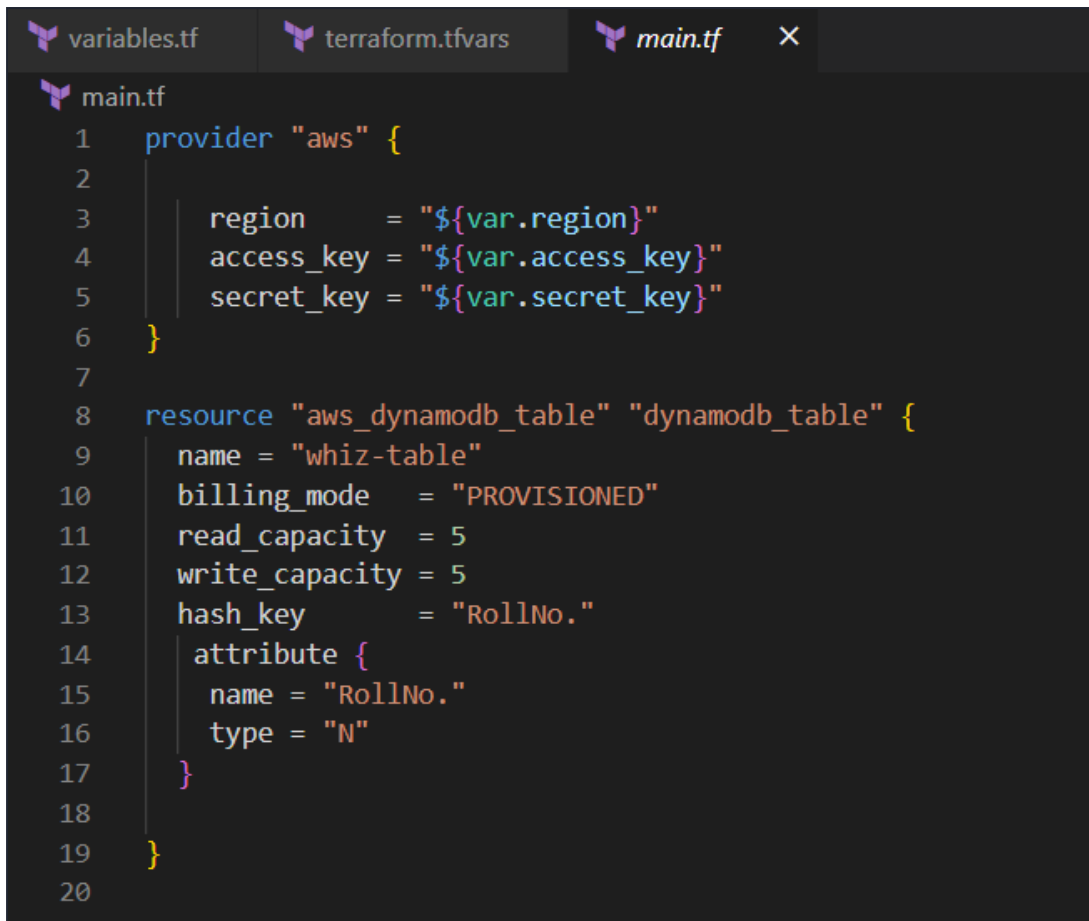
4. In the above code, you are defining the provider as **AWS**.
5. Next, we want to tell Terraform to create a DynamoDB table named as **whiz-table**.
6. Paste the below content into the **main.tf** file after the provider.



```
resource "aws_dynamodb_table" "dynamodb_table" {  
  name           = "whiz-table"  
  billing_mode   = "PROVISIONED"  
  read_capacity  = 5  
  write_capacity = 5  
  
  hash_key = "RollNo."  
  
  attribute {  
    name = "RollNo."  
    type = "N"  
  }  
}
```

- In the above code , we are telling terraform to create a table with table name as **whiz-table**. The billing mode should be **provisioned** by default.
- One **read capacity unit** describes the one strongly consistent read per second upto 1 KB in size.
- One **write capacity unit** describes the one strongly consistent write per second upto 1 KB in size.
- **Hash\_key** represents the partition key of an item. It is composed of one attribute that acts as a primary key for the table.
- We have defined RollNo. as the primary attribute which will be an integer. Therefore we have declared the type as "**N**".





```
variables.tf  terraform.tfvars  main.tf  X
main.tf
1  provider "aws" {
2
3      region      = "${var.region}"
4      access_key  = "${var.access_key}"
5      secret_key  = "${var.secret_key}"
6  }
7
8  resource "aws_dynamodb_table" "dynamodb_table" {
9      name = "whiz-table"
10     billing_mode    = "PROVISIONED"
11     read_capacity   = 5
12     write_capacity  = 5
13     hash_key        = "RollNo."
14     attribute {
15         name = "RollNo."
16         type = "N"
17     }
18
19 }
20
21
```

## Task 5: Adding items to the DynamoDB Table

In this task, we are going to add items to the DynamoDB Table in the main.tf file.

1. To add items to the table , paste the following content in the **main.tf**.

```
resource "aws_dynamodb_table_item" "item1" {
    table_name = aws_dynamodb_table.dynamodb_table.name
    hash_key   = aws_dynamodb_table.dynamodb_table.hash_key
    item = <<ITEM
{
    "RollNo.": {"N": "1"},
    "Name": {"S": "Anant"}
}
ITEM
}
```



```

resource "aws_dynamodb_table" "dynamodb_table" {
  name = "whiz-table"
  billing_mode = "PROVISIONED"
  read_capacity = 5
  write_capacity = 5
  hash_key = "RollNo."
  attribute {
    name = "RollNo."
    type = "N"
  }
}

resource "aws_dynamodb_table_item" "item1" {
  table_name = aws_dynamodb_table.dynamodb_table.name
  hash_key = aws_dynamodb_table.dynamodb_table.hash_key

  item = <<ITEM
  {
    "RollNo.": {"N": "1"},
    "Name": {"S": "Anant"}
  }
  ITEM
}

```

2. In the above code, we have used the resource type as `aws_dynamodb_table_item` and associated this item with the table name and the hash key. We have declared an item with RollNo. as 1 and Name as Anant.

3. You can similarly add more items. Paste the following content to add two more items.

```

resource "aws_dynamodb_table_item" "item2" {
  table_name = aws_dynamodb_table.dynamodb_table.name
  hash_key = aws_dynamodb_table.dynamodb_table.hash_key
  item = <<ITEM
  {
    "RollNo.": {"N": "2"},
    "Name": {"S": "Pavan"}
  }
  ITEM
}

resource "aws_dynamodb_table_item" "item3" {
  table_name = aws_dynamodb_table.dynamodb_table.name
  hash_key = aws_dynamodb_table.dynamodb_table.hash_key
  item = <<ITEM
  {
    "RollNo.": {"N": "3"},
    "Name": {"S": "Nikhil"}
  }
  ITEM
}

```



```
}  
ITEM  
}
```

## Task 6: Create an output file

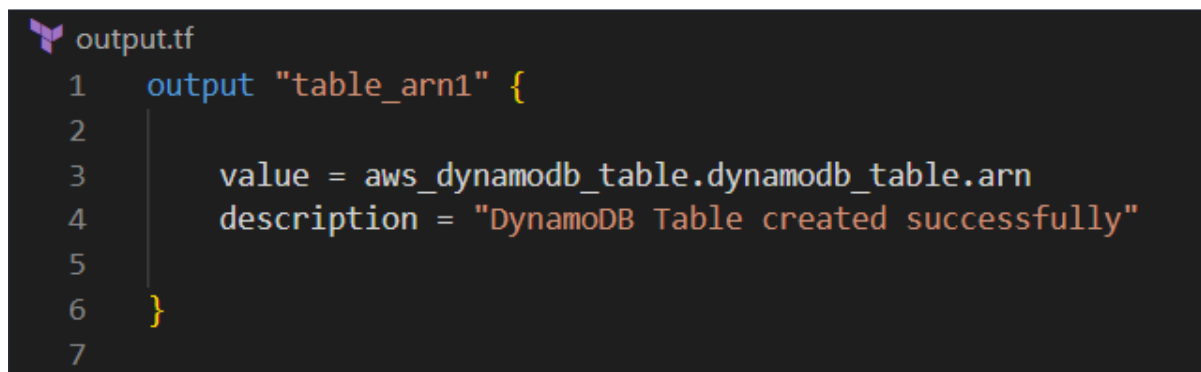
In this task, you will create an `output.tf` file where you will add details of the provider and resources.

1. To create an `output.tf` file, expand the folder **task\_10094\_sns** 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 "table_arn1" {  
    value = aws_dynamodb_table.dynamodb_table.arn  
    description = "DynamoDB Table created successfully"  
}
```



4. In the above code, we will extract the DynamoDB table arn to confirm that they are created.



```
output.tf  
1  output "table_arn1" {  
2  
3      value = aws_dynamodb_table.dynamodb_table.arn  
4      description = "DynamoDB Table created successfully"  
5  
6  }  
7
```

## 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_10098_dynamodb
```



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: Applying terraform configurations

1. Initialize Terraform by running the below command,

```
terraform init
```



```
PS C:\Users\██████\Desktop\task_10098_DynamoDB> terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v4.37.0

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.
```

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

```
terraform plan
```



```
PS C:\Users\XXXXXX\Desktop\task_10098_DynamoDB> terraform plan

Terraform used the selected providers to generate the following
+ create

Terraform will perform the following actions:

# aws_dynamodb_table.dynamodb_table will be created
+ resource "aws_dynamodb_table" "dynamodb_table" {
  + arn                = (known after apply)
  + billing_mode       = "PROVISIONED"
  + hash_key           = "RollNo."
  + id                 = (known after apply)
  + name               = "whiz-table"
  + read_capacity      = 5
  + stream_arn         = (known after apply)
  + stream_label       = (known after apply)
  + stream_view_type   = (known after apply)
  + tags_all           = (known after apply)
  + write_capacity     = 5

  + attribute {
    + name = "RollNo."
    + type = "N"
  }

  + point_in_time_recovery {
    + enabled = (known after apply)
  }
}
```

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

```
terraform apply
```



4. Enter **yes** and the resources will be created in 2-3 minutes.

```
Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ table_arn1 = (known after apply)

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

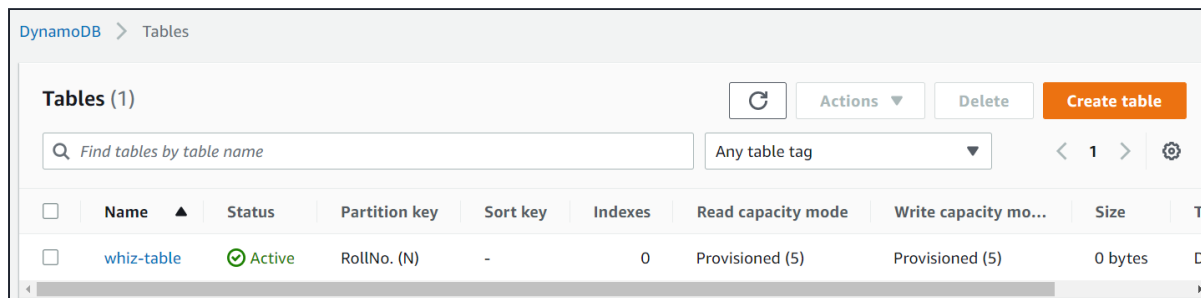
aws_dynamodb_table.dynamodb_table: Creating...
aws_dynamodb_table.dynamodb_table: Still creating... [10s elapsed]
aws_dynamodb_table.dynamodb_table: Creation complete after 11s [id=whiz-table]
aws_dynamodb_table_item.item1: Creating...
aws_dynamodb_table_item.item3: Creating...
aws_dynamodb_table_item.item2: Creating...
aws_dynamodb_table_item.item3: Creation complete after 1s [id=whiz-table|RollNo.|||3]
aws_dynamodb_table_item.item1: Creation complete after 1s [id=whiz-table|RollNo.|||1]
aws_dynamodb_table_item.item2: Creation complete after 1s [id=whiz-table|RollNo.|||2]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

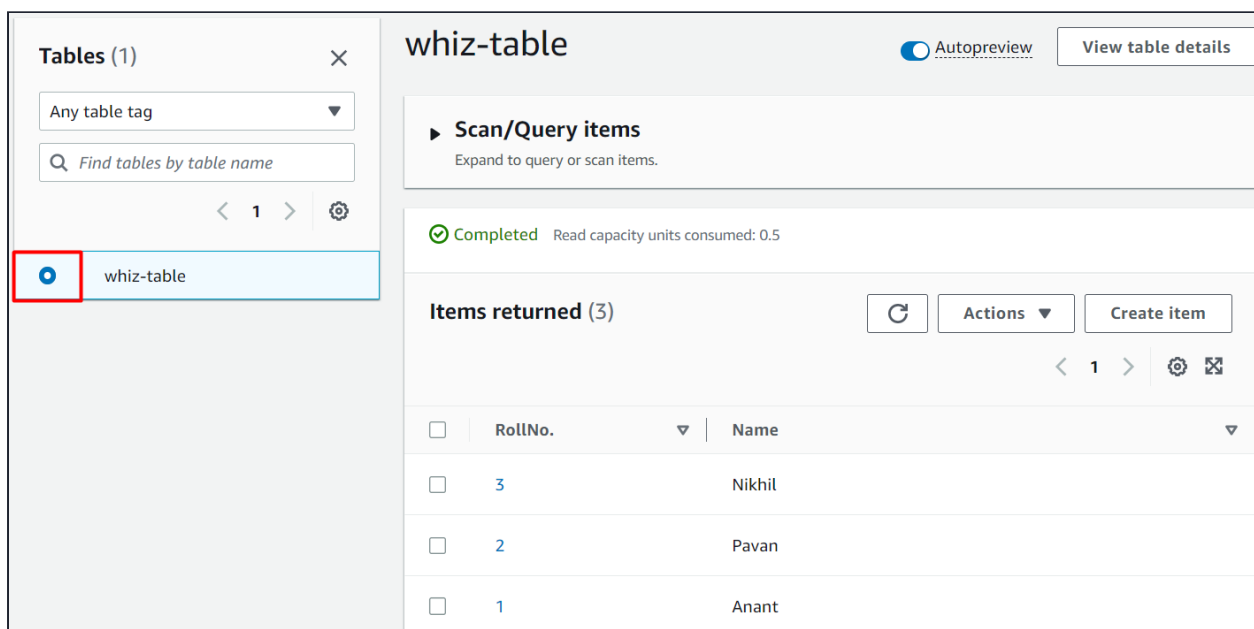
Outputs:
```

## Task 9: Check the resources in the AWS Console

1. Navigate to the **DynamoDB** page by clicking on the **Services** menu at the top.  
**DynamoDB** is available under the **Database** section.
2. Click on **Tables** on the left navigation panel.
3. You can view that the table is created successfully.



4. To check whether the items are added or not, click on **Explore items** on the left navigation panel. **Select** the table and you can view the items created



## Do you know ?

DynamoDB allows you to set a TTL for items, and the service will automatically delete expired items from the table. In addition to the primary key, DynamoDB supports GSIs, which enables efficient querying on non-key attributes.

## 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:

## 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. 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_dynamodb_table_item.item3: Destroying... [id=whiz-table|RollNo. |||3]
aws_dynamodb_table_item.item1: Destroying... [id=whiz-table|RollNo. |||1]
aws_dynamodb_table_item.item2: Destroying... [id=whiz-table|RollNo. |||2]
aws_dynamodb_table_item.item2: Destruction complete after 2s
aws_dynamodb_table_item.item1: Destruction complete after 2s
aws_dynamodb_table_item.item3: Destruction complete after 2s
aws_dynamodb_table.dynamodb_table: Destroying... [id=whiz-table]
aws_dynamodb_table.dynamodb_table: Destruction complete after 5s

Destroy complete! Resources: 4 destroyed.
```

## Completion and Conclusion

1. You have set up the Visual Studio Code

2. You have successfully created variables.tf and terraform.tfvars files.
3. You have successfully created a main.tf file.
4. You have executed the terraform configurations commands to create the resources.
5. You have checked the resources created by opening the AWS Console.
6. You have deleted all the resources created.

## End Lab

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Click on **End Lab** button from Whizlabs Labs console and wait till the process gets completed.

[About Us](#) [Subscription](#) [Instructions and Guidelines](#) [FAQ's](#) [Contact Us](#)



© 2023, Whizlabs Software Pvt. Ltd.

