**Lab Steps**

Task 1: Sign in to AWS Management Console

1. Click on the  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. Create a folder using File Explorer on your local machine's desktop with the name **task\_10002\_s3**
2. Open the Visual Studio Code application. If you haven't installed the VSC software, download and install using the software using the link provided in the Prerequisite section above.
3. If you have already installed and using Visual studio code, Open a New Window by clicking **File** from the menu bar present on the top and select **New Window**.
4. And, now select **Open** present under the **File** menu and navigate to the desktop location. Select the Folder task\_10002\_s3.
5. 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**.
6. 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\_10002\_s3**folder**.**
4. Paste the below contents in **variables.tf**file:

|  |
| --- |
| # required for AWS  variable "access\_key" {}  variable "secret\_key" {}  variable "region" {  default = "us-east-1"  } |

1. In the above content, you are declaring 3 variables called access\_key, secret\_key, and region having default value as **us-east-1**.
2. After pasting the above contents, save the file by pressing Ctrl + S.
3. Add variables that are specific to your site.

|  |
| --- |
| variable "root\_domain" {  default = "<name of your site>"  }  variable "blog\_bucket\_subdomain" {  default = "<subdomain of your site>"  } |

1. In the above code, you are declaring variables for your site, what will be your root domain and subdomain for the site.
2. Add the following values
   1. **root\_domain** and **blog\_bucket\_subdomain** for your site in the code above.
   2. **root\_domain** example : ***whizlabs-s3-bucket-demo*** (Please make sure to add a unique name in lower case)
   3. **blog\_bucket\_subdomain** example : ***blog*** (You can provide any name in lower case)
3. Now create the **terraform.tfvars** file by selecting **New file** present under **File** in the menu bar.
4. Name the file by pressing Ctrl + S and enter **terraform.tfvars**
5. Paste the below content into **terraform.tfvars** file

|  |
| --- |
| access\_key = "<YOUR AWS CONSOLE ACCESS ID>"  secret\_key = "<YOUR AWS CONSOLE SECRET KEY>" |

1. In the above code, you are defining the dynamic values of variables declared earlier.
2. Replace the values of **access\_key** and **secret\_key** by copying the AWS Access Key Id and Secret Access Key Id provided in Whizlabs Labs console.
3. After replacing the values of access\_key and secret\_key, save the file by pressing Ctrl + S.

Task 4: Create an S3 bucket 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\_s3\_bucket" "blog" {  bucket = "${var.blog\_bucket\_subdomain}.${var.root\_domain}"  }  resource "aws\_s3\_bucket\_website\_configuration" "blog" {  bucket = aws\_s3\_bucket.blog.id    index\_document {  suffix = "index.html"  }    error\_document {  key = "error.html"  }  }  resource "aws\_s3\_bucket\_acl" "blog\_bucket\_acl" {  bucket = aws\_s3\_bucket.blog.id  acl = "public-read"  } |

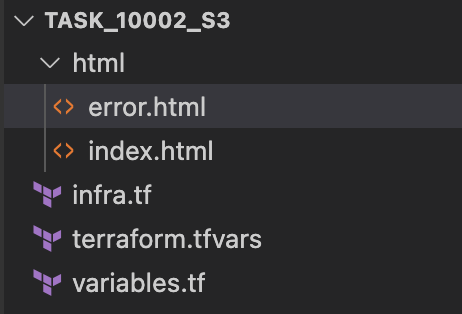
1. In the above code, you are defining provider as ***aws*** and defining resource block using ***aws\_s3\_bucket***  for creating an **Amazon S3 Bucket** having name as values defined by you in previous steps, **public-read enabled**, index\_document, and error document defined as **index.html and error.html** respectively.
2. Add another block of just below the s3 bucket creation code, this **block of code will upload all the files present under html folder to the S3 bucket.**

|  |
| --- |
| resource "aws\_s3\_object" "upload\_object" {  for\_each = fileset("html/", "\*")  bucket = aws\_s3\_bucket.blog.id  key = each.value  source = "html/${each.value}"  etag = filemd5("html/${each.value}")  content\_type = "text/html"  } |

1. Finally, make the bucket public by adding the bucket policy by adding another block of code to the ***main.tf*** file.

|  |
| --- |
| resource "aws\_s3\_bucket\_policy" "read\_access\_policy" {  bucket = aws\_s3\_bucket.blog.id  policy = <<POLICY  {  "Version": "2012-10-17",  "Statement": [  {  "Sid": "PublicReadGetObject",  "Effect": "Allow",  "Principal": "\*",  "Action": [  "s3:GetObject"  ],  "Resource": [  "arn:aws:s3:::${aws\_s3\_bucket.blog.id}/\*"  ]  }  ]  }  POLICY  } |

1. **Before moving to the next step, save the file using Ctrl + S.**
2. Next you will create a new folder to place the index.html and error.html files. To create a folder, click on the **Create folder icon** present on the left side panel under your folder called **task\_10002\_s3**.  
   Text

   Description automatically generated with medium confidence
3. Name this folder as **html**.
4. Now click on this [Link](https://labresources.whizlabs.com/a1bb9f4fa305e97136c14d2013138f13/task_23.zip) to download file to your local,extract the zip file.
5. Place the **index.html** and **error.html** files inside the newly created folder called **html**
6. Or you can manually create an index.html and error.html file inside the **html** folder and paste the contents of downloaded files.
7. By now your files and folder structure should look something like this. As we have created **main.tf**, **terraform.tfvars** and **variables.tf** in the root directory of the folder and index.html and error.html inside the html folder.  
   
8. Create an **outputs.tf** file required for displaying the output as website endpoint.
9. To create **outputs.tf** file, click on the **File** from the menu bar and choose **New file**
10. Press Ctrl + S to save the new file as **outputs.tf**and click on the **Save** button after entering the file name.
11. Paste the below content into **outputs.tf** file.

|  |
| --- |
| output "s3\_bucket\_id" {  value = aws\_s3\_bucket\_website\_configuration.blog.website\_endpoint  } |

1. 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. To confirm the installation of Terraform, run the below command to check the version:
   * terraform version
3. 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. Initialize Terraform by running the below command:
   * terraform init
2. **Note:**terraform init will check for all the plugin dependencies and download them if required, this will be used for creating a deployment plan.
3. To generate the action plans run the below command:
   * terraform plan
4. Review the whole generated plan.
5. To create all the resources declared in **main.tf** configuration file, run the below command:
   * terraform apply
6. You will be able to see the resources which will be created, approve the creation of all the resources by entering **yes.**
7. It may take up to 2 minutes for the **terraform apply** command to create the resources.
8. Ids of all the resources created by terraform will be visible there.
9. Optionally, you can note down the IDs of all the resources.

Task 7: Check the resources in AWS Console

1. Navigate to AWS Management Console page in your browser.
2. Make sure you are inthe **US East (N. Virginia) us-east-1**Region.
3. Navigate to the **Services** menu at the top. Click on **S3**in the **Storage** section.
4. On the home page of the S3 bucket, click on the bucket you created. The bucket name will be the same as the subdomain and root name you have provided in the variables.tf file.
5. After you open the bucket, two files are present index.html and error.html
6. Switch to the properties tab and scroll the end of this page to find Static website hosting options.
7. Copy the Bucket website endpoint and paste into the new tab of your web browser with /index.html at the end.
8. For testing error.html feature is working or not, replace index.html with home.html. Now you should see the contents of error.html
9. To verify the contents of error.html, open this page by replacing with home.html

Task 8: Validation of the lab

1. Once the lab steps are completed, please click on the **** 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 :   
   Table

   Description automatically generated

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.

**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  from your whizlabs lab console and wait till the process gets completed.