Home / AWS / Guided Lab / Create an SES Email service using Terraform

Create an SES Email service using Terraform

Level: **Fundamental**

Amazon SES Amazon Web Services Terraform

0h 29m 45s left

15 MIN

End Lab

Open Console

Validation

Lab Credentials

User Name

Whiz_User_80425.80618416

Password

5e0e727c-1933-4081-8ef5-6dd56cac7aea

Access Key

AKIA435KNYT5FPZS7U7E

Secret Key

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

Lab Resources

No Lab Resources Found

WHIZLABS 0 K

ns

Need help?

- How to use Hands on Lab
- Troubleshooting Lab
- FAQs

Submit Feedback Share

Lab Overview Lab Steps Lab Validation

Cloud Architect, Cloud Administrator

Business Applications, 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_10092_ses
```

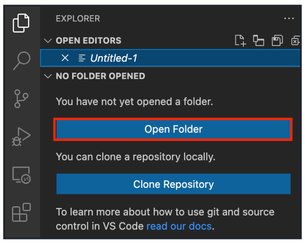
8. Change your present working directory to use the newly created folder by running the below command:

```
cd task_10092_ses
```

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_10092_ses**.



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_10092_ses 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.

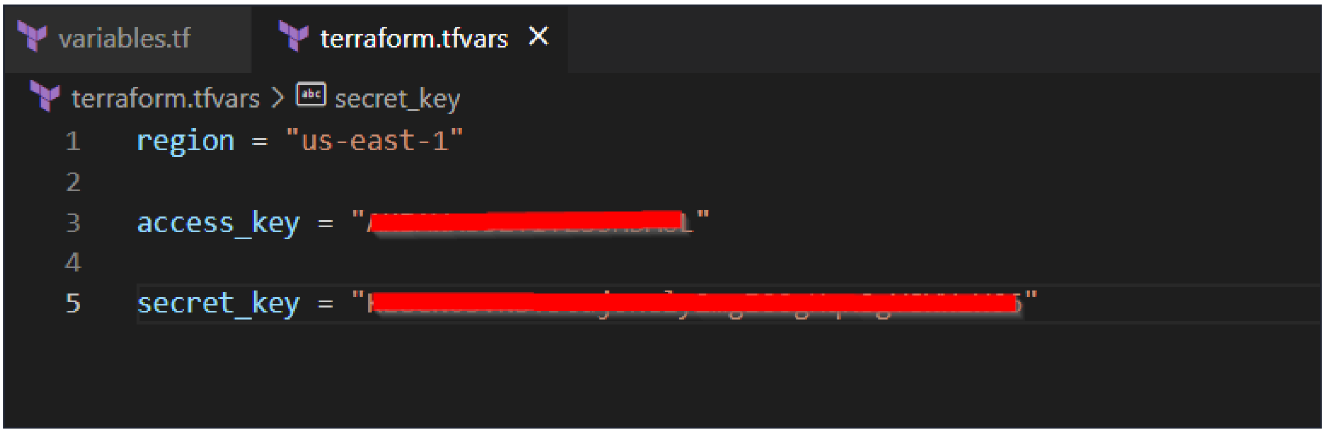
1. To create a variable file, expand the folder **task_10092_ses** and click on the **New File** icon to add the file.
2. Name the file as **variables.tf** and press **Enter** to save it.
3. **Note:** Don't change the location of the new file, keep it default, i.e. inside the **task_10092_ses** folder**.**
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_10092_ses** 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.



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

access_key = "[redacted]"

secret_key = "[redacted]"
```

Task 4: Create a SES identity 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_10092_ses** 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 SES Identity with Email**.**
6. Paste the below content into the **main.tf** file after the provider.

NOTE: Enter Your Email address

```
resource "aws_ses_email_identity" "ses_identity" {
    email = "Enter you Email"
  }
```

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

Task 5: 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_10092_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 "Identity_arn" {
    value = aws_ses_email_identity.ses_identity.arn
    description = "Identity created successfully"
}
```

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

Task 6: 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_10092_sns
```

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 7: Apply to terraform configurations

1. Initialize Terraform by running the below command,

```
terraform init
```

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
```

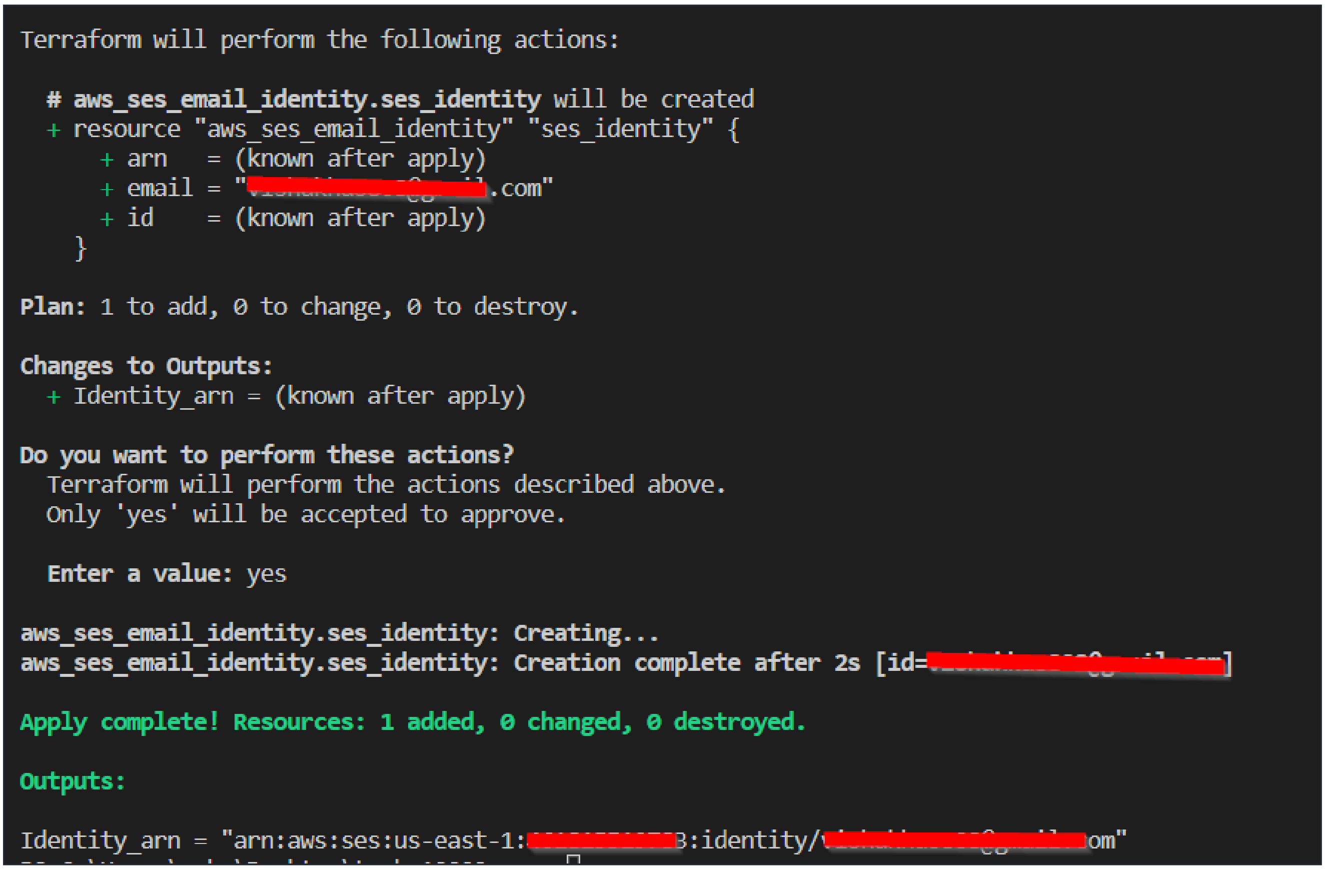
3. To create all the resources declared in main.tf configuration file, run the below command,

```
terraform apply
```

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

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

6. Enter a value : Enter **yes**

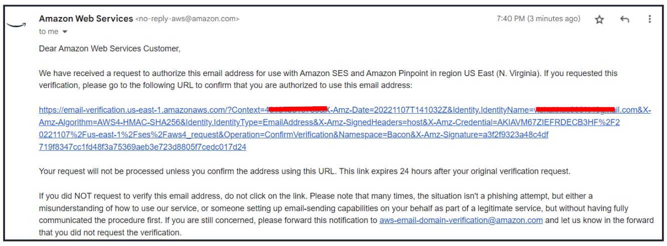


```
Terraform will perform the following actions:

  # aws_ses_email_identity.ses_identity will be created
  + resource "aws_ses_email_identity" "ses_identity" {
      + arn   = (known after apply)
      + email = "[redacted].com"
      + id    = (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + Identity_arn = (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_ses_email_identity.ses_identity: Creating...
aws_ses_email_identity.ses_identity: Creation complete after 2s [id=[redacted]]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

Identity_arn = "arn:aws:ses:us-east-1:[redacted]3:identity/[redacted]om"
```

Task 8: Confirm the SES Identity Verification

1. AWS will send a confirmation email to the email that you provided in the above step.

2. Login to your mailbox and you will be able to see mail from Amazon Web Services.



Amazon Web Services <no-reply-aws@amazon.com>
to me
7:40 PM (3 minutes ago)

Dear Amazon Web Services Customer,

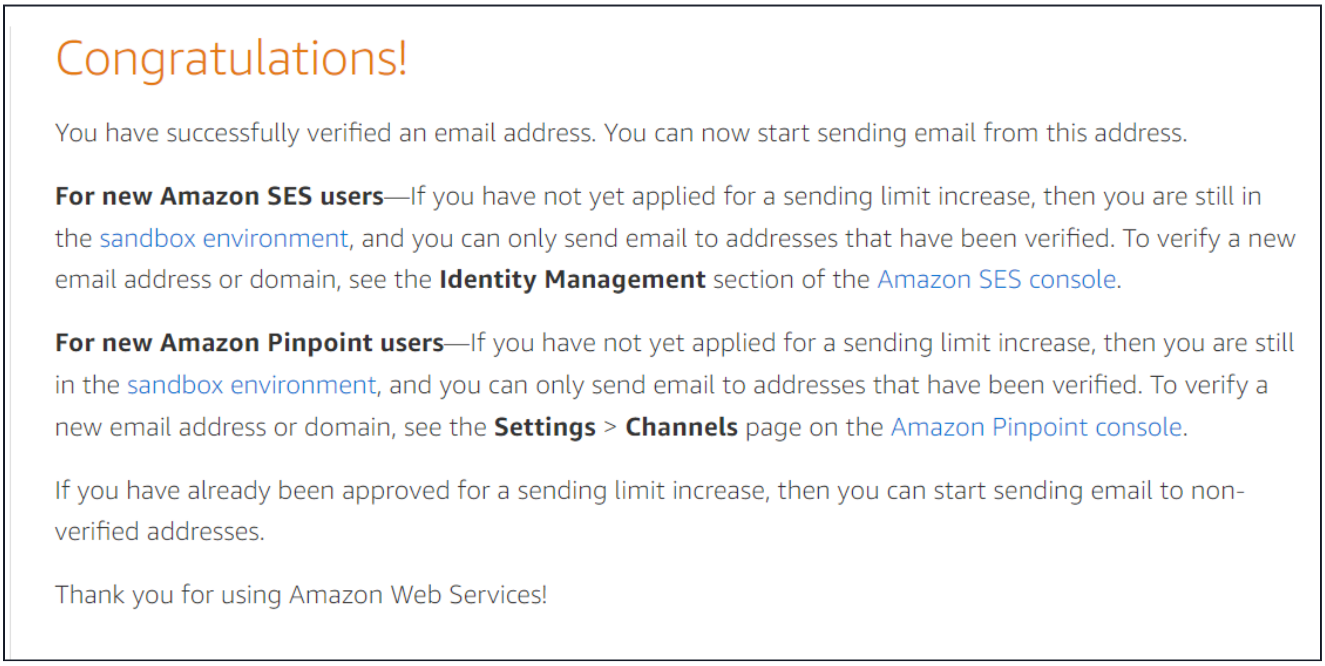
We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US East (N. Virginia). If you requested this verification, please go to the following URL to confirm that you are authorized to use this email address:

https://email-verification.us-east-1.amazonaws.com/?Context=4[redacted]X-Amz-Date=20221107T141032Z&Identity.IdentityName=[redacted]il.com&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=host&X-Amz-Credential=AKIAVM67ZIEFRDECB3HF%2F20221107%2Fus-east-1%2Fses%2Faws4_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=a3f2f9323a48c4df719f8347cc1fd48f3a75369aeb3e723d8805f7cedc017d24

Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.

If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use our service, or someone setting up email-sending capabilities on your behalf as part of a legitimate service, but without having fully communicated the procedure first. If you are still concerned, please forward this notification to aws-email-domain-verification@amazon.com and let us know in the forward that you did not request the verification.

3. Once you click on the **link**, you will be routed to the success page



Congratulations!

You have successfully verified an email address. You can now start sending email from this address.

For new Amazon SES users—If you have not yet applied for a sending limit increase, then you are still in the sandbox environment, and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Identity Management** section of the Amazon SES console.

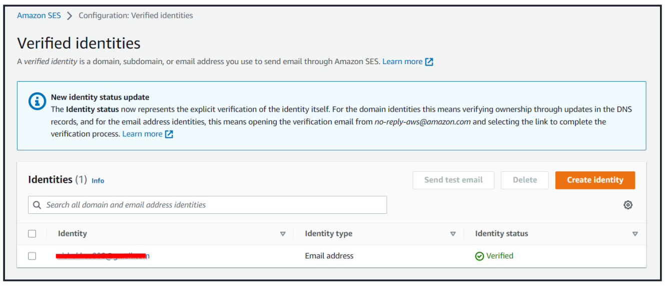
For new Amazon Pinpoint users—If you have not yet applied for a sending limit increase, then you are still in the sandbox environment, and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Settings** > **Channels** page on the Amazon Pinpoint console.

If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

Thank you for using Amazon Web Services!

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 **SES** by clicking on **Services** on the top, then click on **SES** in the **Business Application** section.
3. Click on the **Verified identities** on the left navigation panel. You can see that the identity is created successfully.



Amazon SES > Configuration: Verified identities

Verified identities

A *verified identity* is a domain, subdomain, or email address you use to send email through Amazon SES. Learn more

New identity status update
The **Identity status** now represents the explicit verification of the identity itself. For the domain identities this means verifying ownership through updates in the DNS records, and for the email address identities, this means opening the verification email from *no-reply-aws@amazon.com* and selecting the link to complete the verification process. Learn more

Identities (1) Info

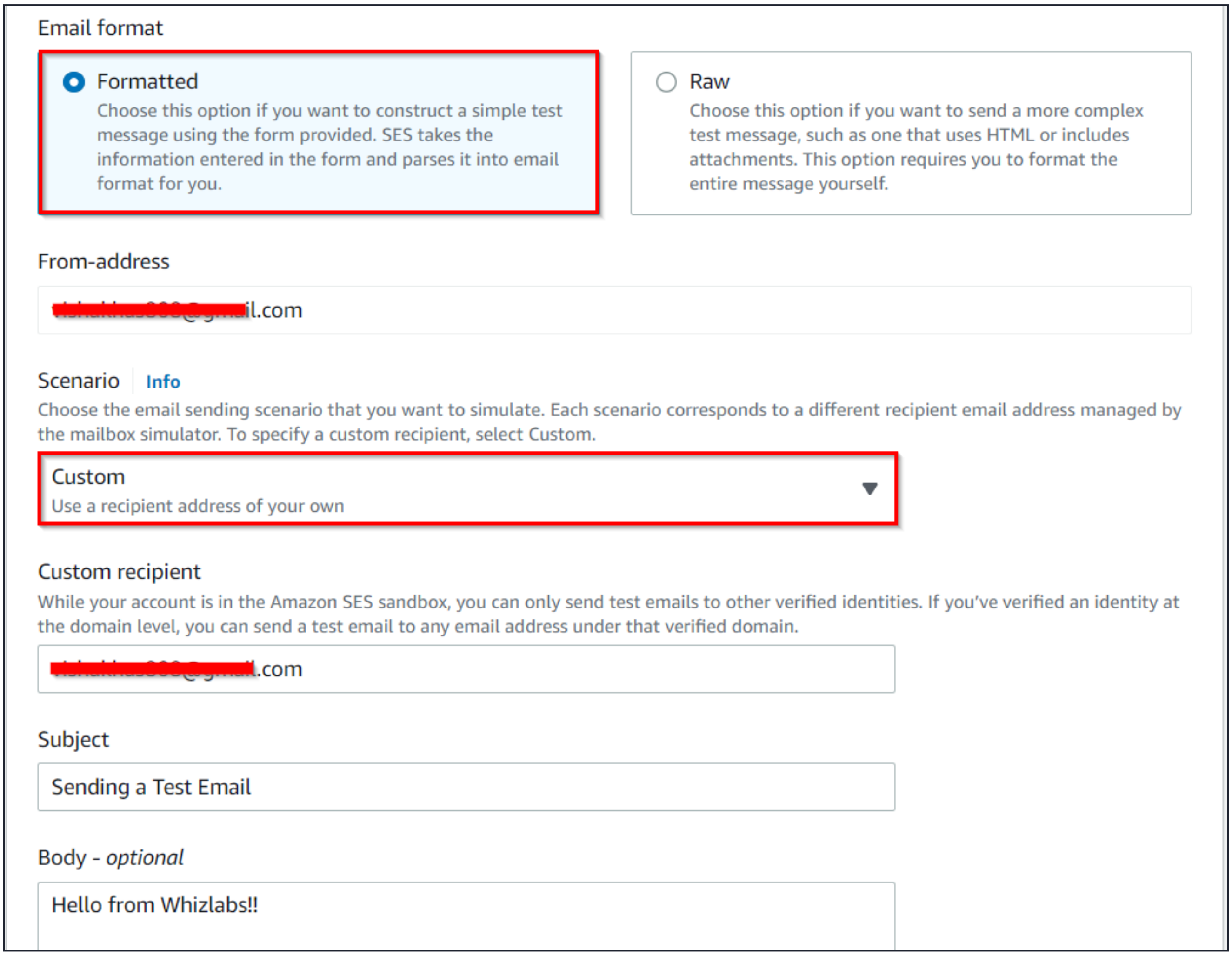
□	Identity	Identity type	Identity status
□	[redacted]	Email address	✓ Verified

Task 10: Send a Test Email

1. Let us send a test email to the same email address.
2. Select the verified email address and click on **Send Test E-mail** button.
3. Make the following changes:
   - Email Format: Select **Formatted** (keep it as default)
   - From: your email address will be by default
   - Scenario: Select **Custom**
   - Custom reciepent: **Give the same email address** (for test purpose)

- Subject: Enter **Sending a Test Email**

- Body: **Type any text as you wish**



Email format

● Formatted
Choose this option if you want to construct a simple test message using the form provided. SES takes the information entered in the form and parses it into email format for you.

○ Raw
Choose this option if you want to send a more complex test message, such as one that uses HTML or includes attachments. This option requires you to format the entire message yourself.

From-address

[redacted]il.com

Scenario Info
Choose the email sending scenario that you want to simulate. Each scenario corresponds to a different recipient email address managed by the mailbox simulator. To specify a custom recipient, select Custom.

Custom
Use a recipient address of your own

Custom recipient
While your account is in the Amazon SES sandbox, you can only send test emails to other verified identities. If you've verified an identity at the domain level, you can send a test email to any email address under that verified domain.

[redacted].com

Subject

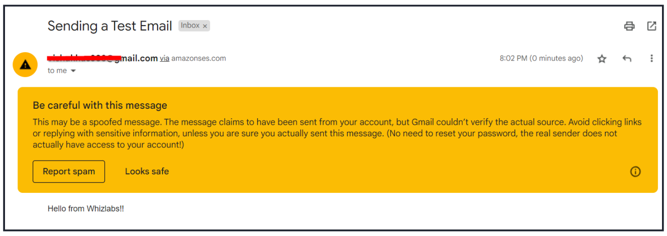
Sending a Test Email

Body - *optional*

Hello from Whizlabs!!

- Click on **Send Test Email** button.

4. Verify the mail received to your Inbox. If it is not present in your inbox, check in the spam.



Sending a Test Email

[redacted]mail.com via amazonses.com 8:02 PM (0 minutes ago)
to me

Be careful with this message
This may be a spoofed message. The message claims to have been sent from your account, but Gmail couldn't verify the actual source. Avoid clicking links or replying with sensitive information, unless you are sure you actually sent this message. (No need to reset your password, the real sender does not actually have access to your account!)

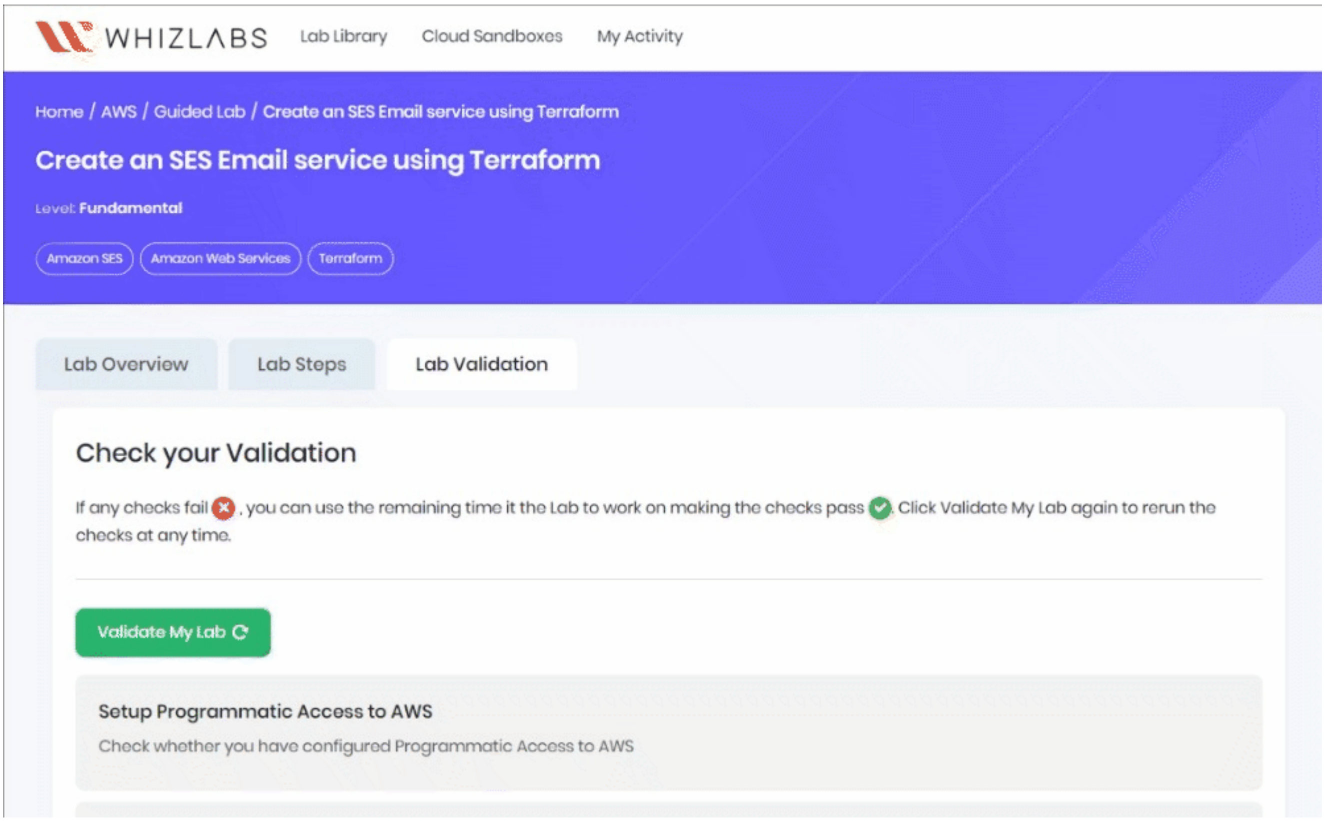
Hello from Whizlabs!!

Do you know?

Amazon SES is a cloud-based email-sending and receiving service provided by Amazon Web Services (AWS). It allows you to send transactional and marketing emails, as well as receive incoming emails. SES helps improve email deliverability and provides tools for managing email campaigns and analyzing email performance.

Task 11: 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 an SES Email service 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.

Setup Programmatic Access to AWS

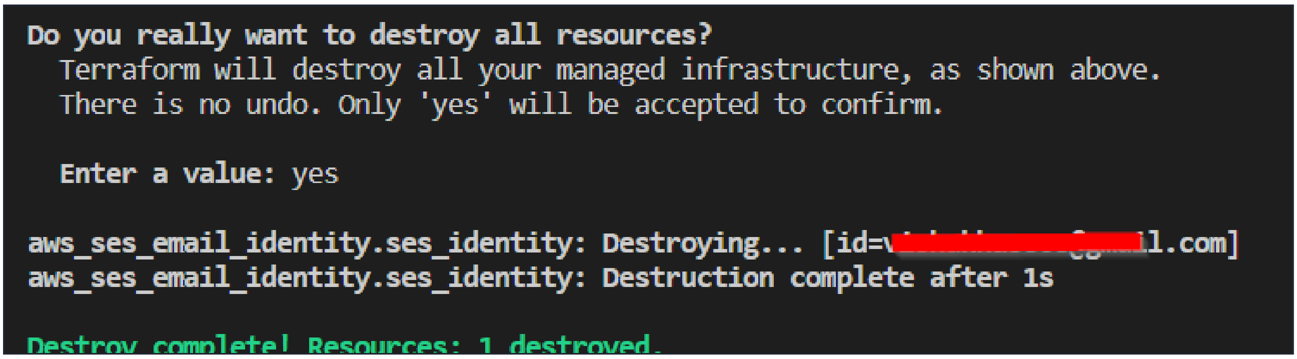
Check whether you have configured Programmatic Access to AWS

Task 12: 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. You can see the **Destroy complete!** message.



```
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_ses_email_identity.ses_identity: Destroying... [id=[redacted]]
aws_ses_email_identity.ses_identity: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.
```

Completion and Conclusion

- You have set up the Visual Studio Code editor.
- You have created variables.tf and terraform.tfvars files.
- You have created a main.tf file.
- You have executed the terraform configuration commands to create the resources.
- You have checked all the resources created by opening the Console.
- You have tested by sending a Test Email.
- You have 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