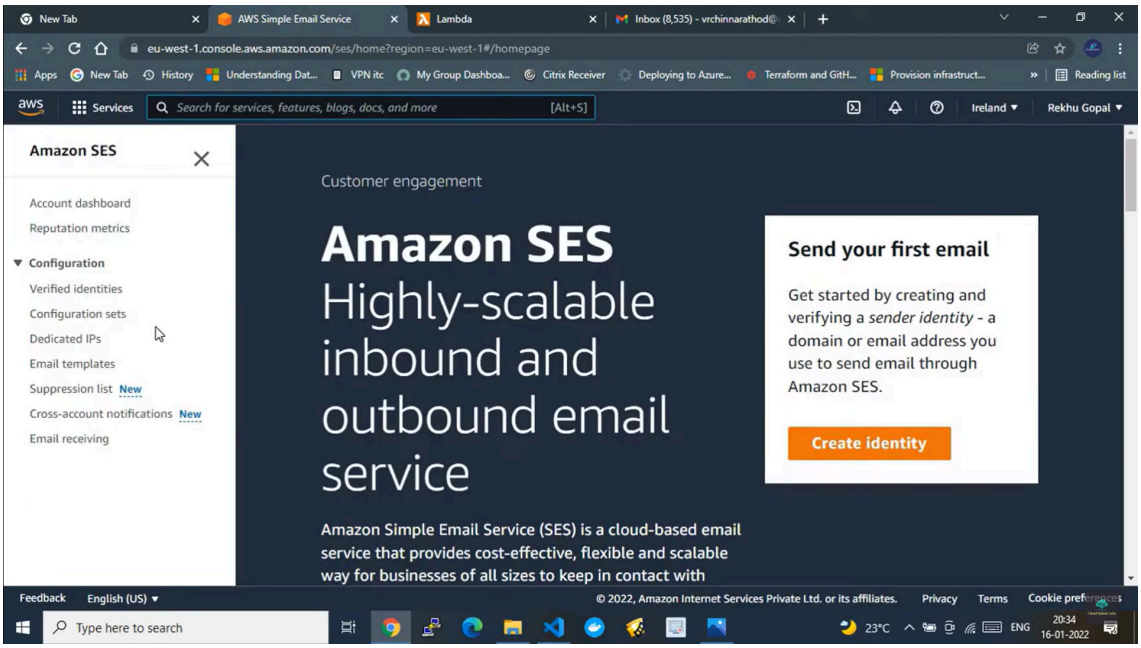
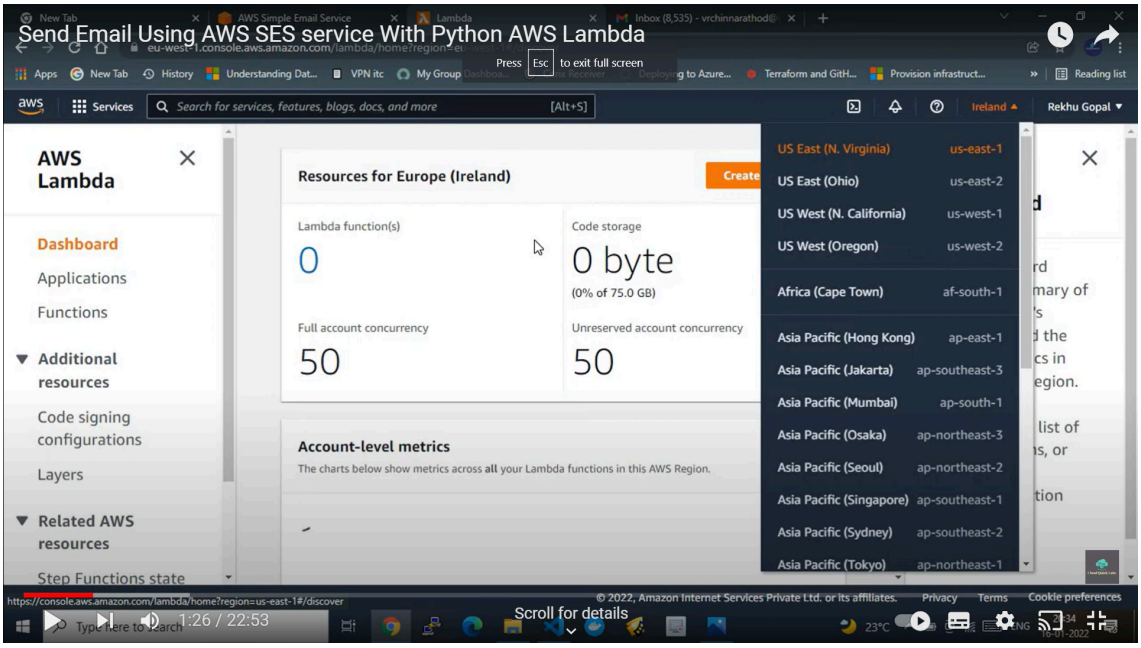
In [2]:

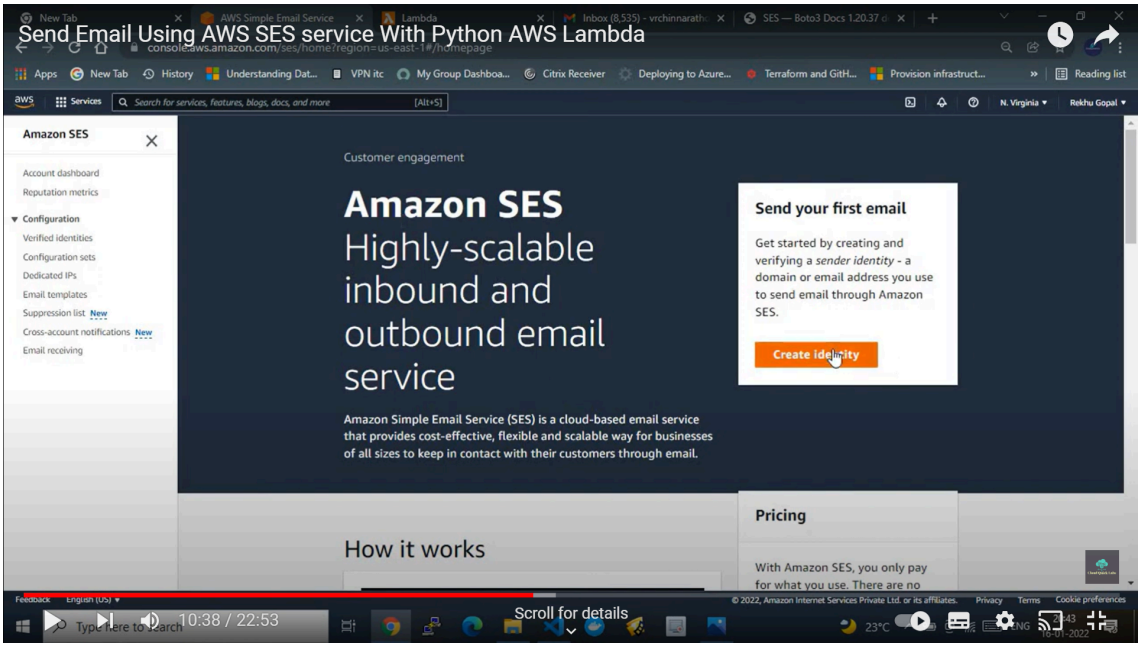
```
import boto3
from botocore.exceptions import ClientError

def send_email():
    SENDER = "youremail@gmail.com" # must be verified in AWS SES Email
    RECIPIENT = "youremail@gmail.com" # must be verified in AWS SES Em

    # If necessary, replace us-west-2 with the AWS Region you're using
    AWS_REGION = "us-east-1"

    # The subject line for the email.
    SUBJECT = "This is test email for testing purpose..!!"

    # The email body for recipients with non-HTML email clients.
    BODY_TEXT = ("Hey Hi...\r\n"
                 "This email was sent with Amazon SES using the "
                 "AWS SDK for Python (Boto)."
                 )

    # The HTML body of the email.
    BODY_HTML = """<html>
    <head></head>
    <body>
    <h1>Hey Hi...</h1>
    <p>This email was sent with
        <a href='https://aws.amazon.com/ses/'>Amazon SES CQPOCS</a> us
        <a href='https://aws.amazon.com/sdk-for-python/'>
        AWS SDK for Python (Boto)</a>.</p>
    </body>
    </html>
                 """

    # The character encoding for the email.
    CHARSET = "UTF-8"

    # Create a new SES resource and specify a region.
    client = boto3.client('ses',region_name=AWS_REGION)

    # Try to send the email.
    try:
        #Provide the contents of the email.
        response = client.send_email(
            Destination={
                'ToAddresses': [
                    RECIPIENT,
                ],
            },
            Message={
                'Body': {
                    'Html': {

                        'Data': BODY_HTML
                    },
                    'Text': {

                        'Data': BODY_TEXT
                    },
                },
                'Subject': {

                    'Data': SUBJECT
```

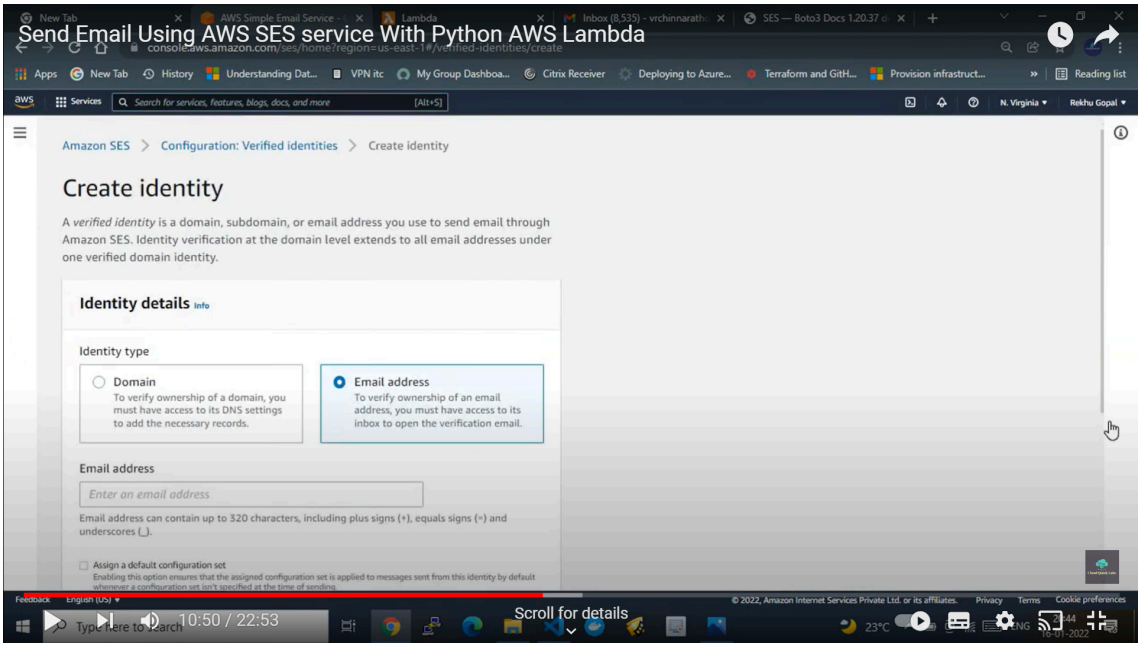
```
                },
            },
            Source=SENDER
        )
    # Display an error if something goes wrong.
    except ClientError as e:
        print(e.response['Error']['Message'])
    else:
        print("Email sent! Message ID:"),
        print(response['MessageId'])

def lambda_handler(event, context):
    # TODO implement
    send_email()

print('done')
```

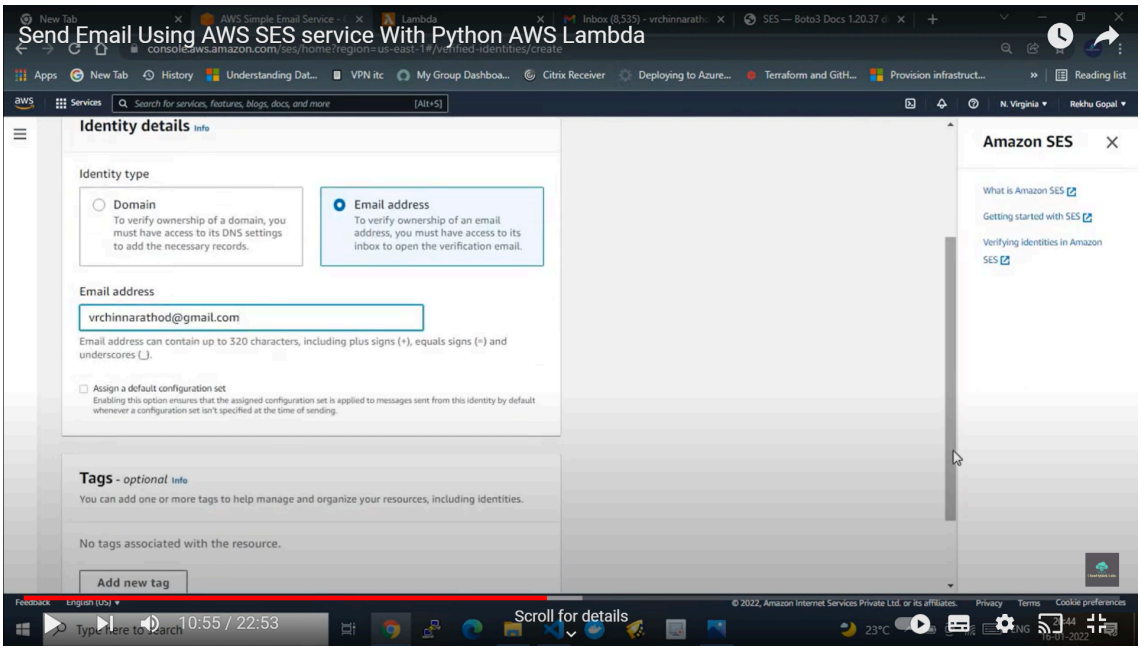
done

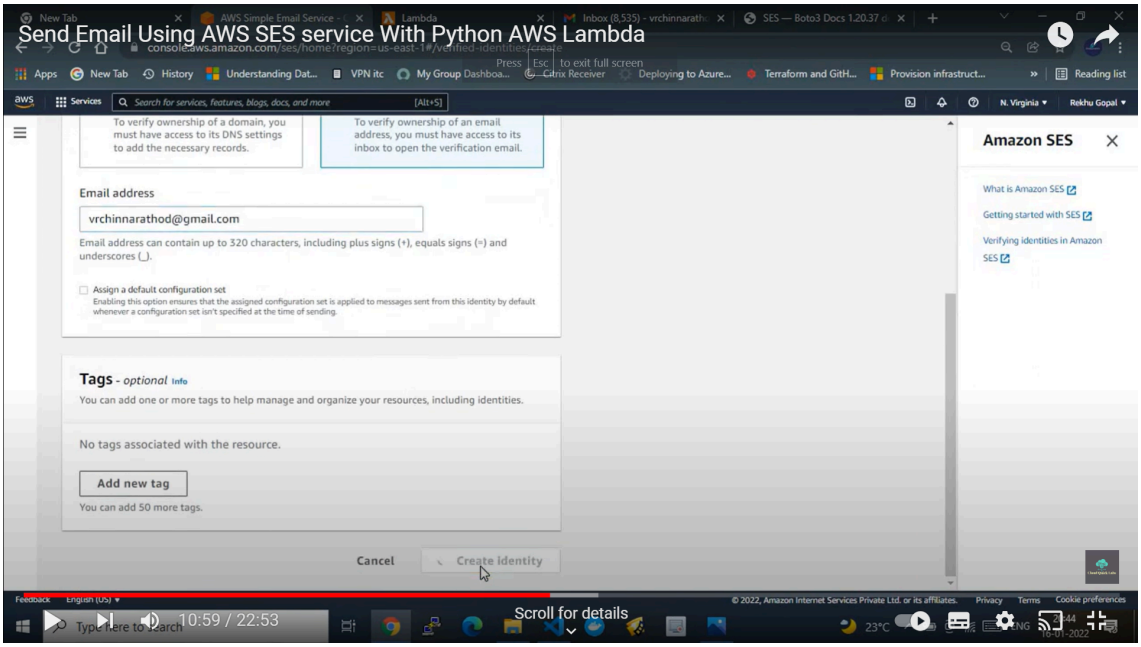


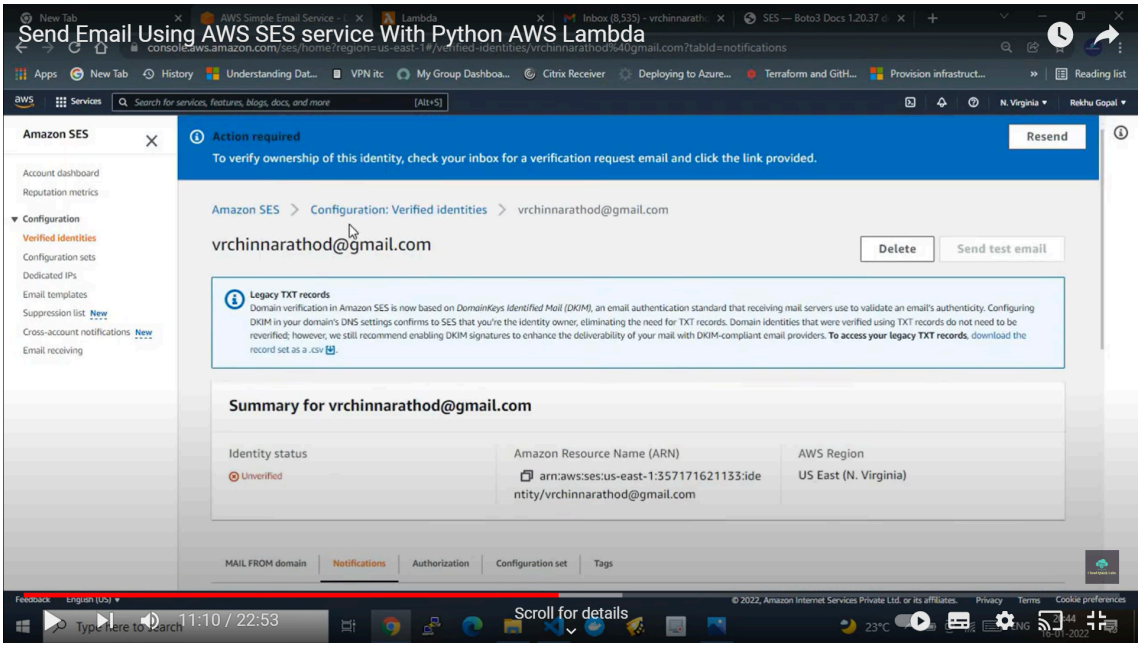



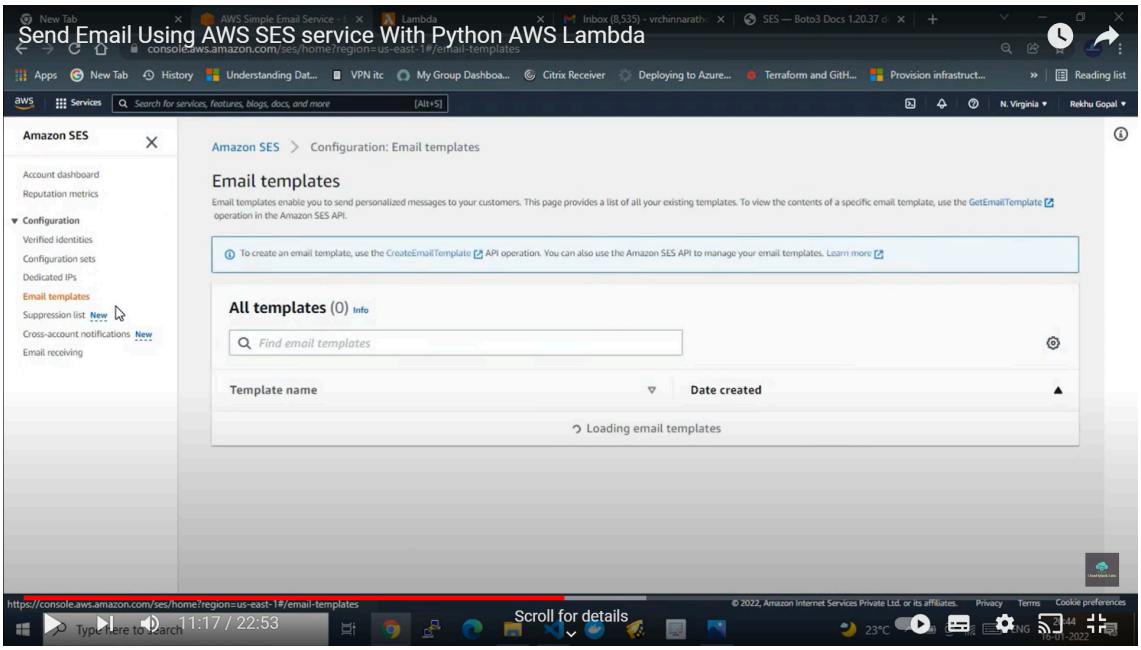

Send Email Using AWS SES service With Python AWS Lambda
Amazon SES
Customer engagement
Amazon SES
Highly-scalable inbound and outbound email service
Amazon Simple Email Service (SES) is a cloud-based email service that provides cost-effective, flexible and scalable way for businesses of all sizes to keep in contact with their customers through email.
Send your first email
Get started by creating and verifying a sender identity - a domain or email address you use to send email through Amazon SES.
Create identity
Pricing
How it works
10:38 / 22:53
Scroll for details

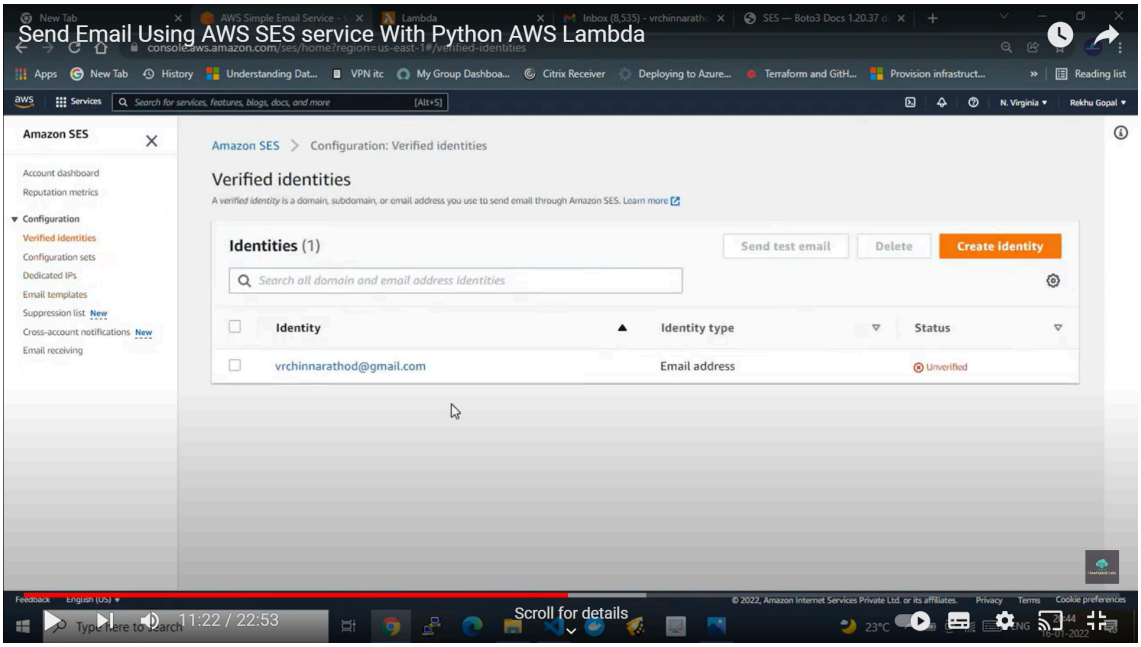

Send Email Using AWS SES service With Python AWS Lambda
Amazon SES > Configuration: Verified identities > Create identity
Create identity
A verified identity is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.
Identity details Info
Identity type
Domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.
Email address
To verify ownership of an email address, you must have access to its inbox to open the verification email.
Email address
Enter an email address
Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).
Assign a default configuration set
10:50 / 22:53
Scroll for details

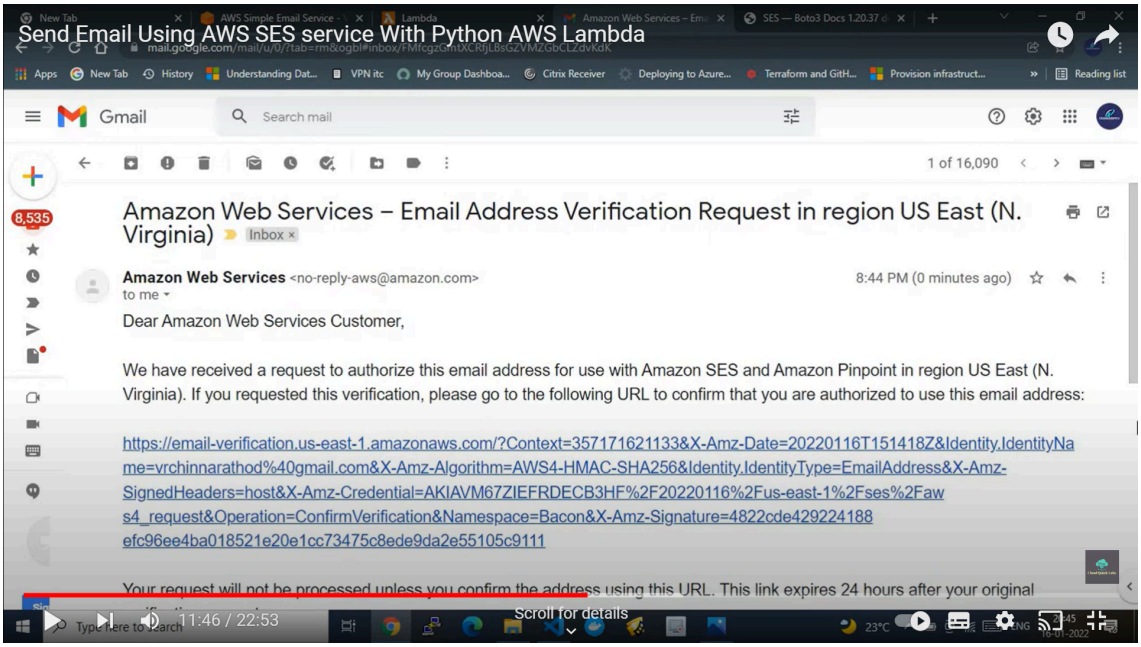

Send Email Using AWS SES service With Python AWS Lambda
Identity details Info
Identity type
Domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.
Email address
To verify ownership of an email address, you must have access to its inbox to open the verification email.
Email address
vrchinnarathod@gmail.com
Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).
Assign a default configuration set
Tags - optional Info
You can add one or more tags to help manage and organize your resources, including identities.
No tags associated with the resource.
Add new tag
Amazon SES
What is Amazon SES
Getting started with SES
Verifying identities in Amazon SES
10:55 / 22:53
Scroll for details

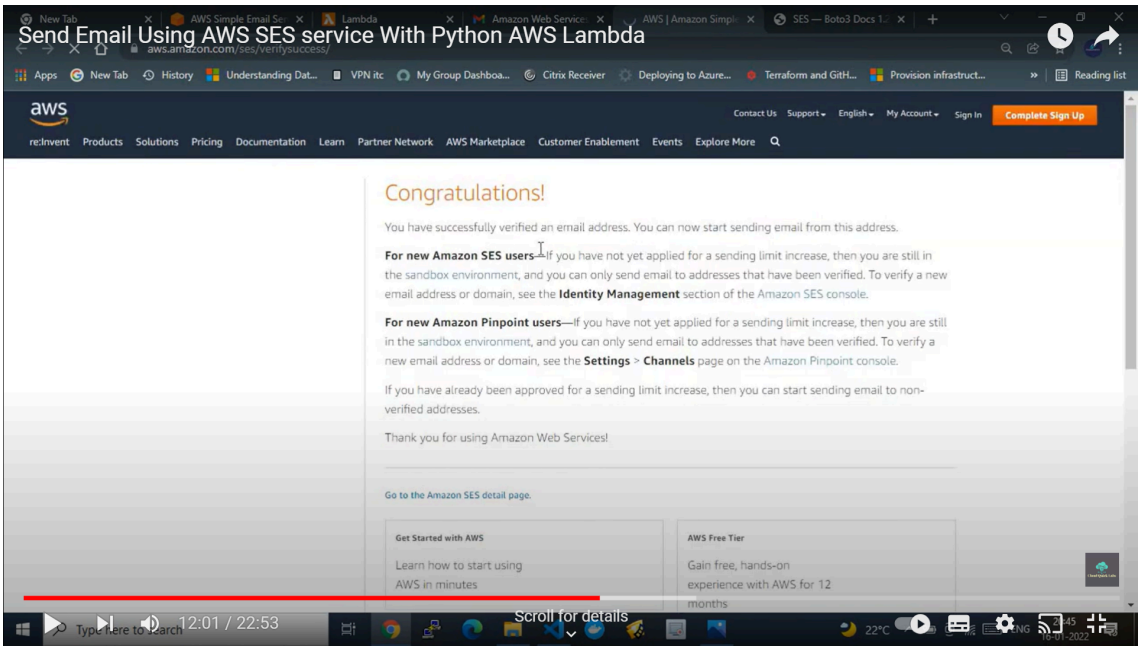

Send Email Using AWS SES service With Python AWS Lambda
Amazon SES
Email address
vrchinnarathod@gmail.com
Tags - optional
No tags associated with the resource.
Add new tag
Cancel
Create identity
10:59 / 22:53
Scroll for details

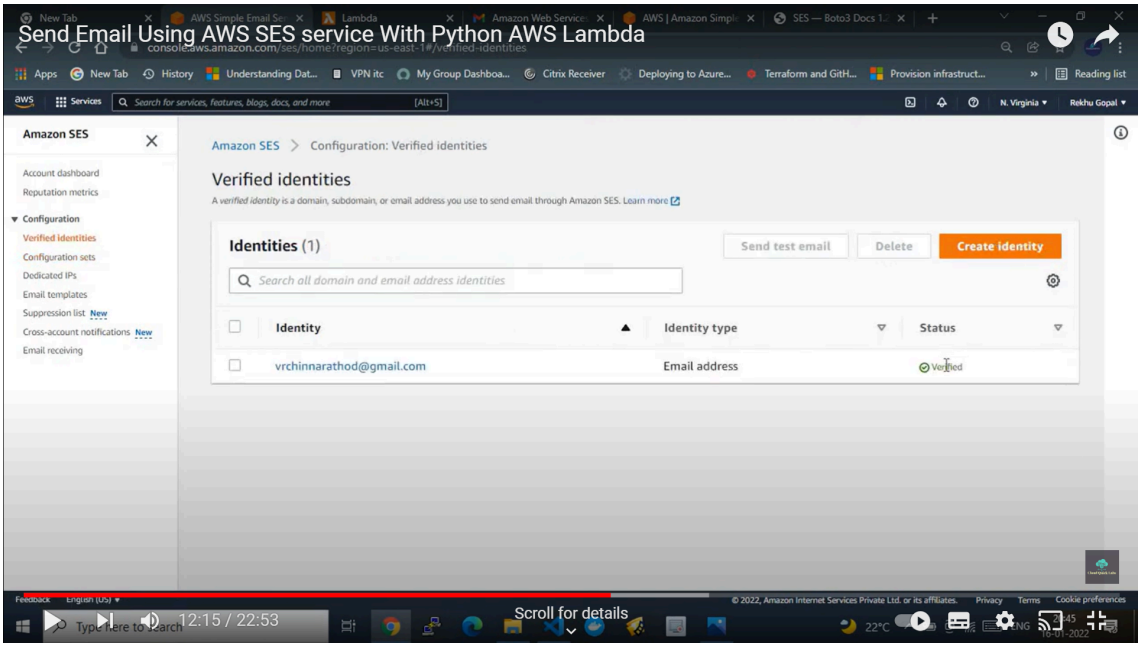

Send Email Using AWS SES service With Python AWS Lambda
Action required
To verify ownership of this identity, check your inbox for a verification request email and click the link provided.
Resend
vrchinnarathod@gmail.com
Delete
Send test email
Summary for vrchinnarathod@gmail.com
Identity status
Unverified
Amazon Resource Name (ARN)
AWS Region
US East (N. Virginia)
11:10 / 22:53
Scroll for details

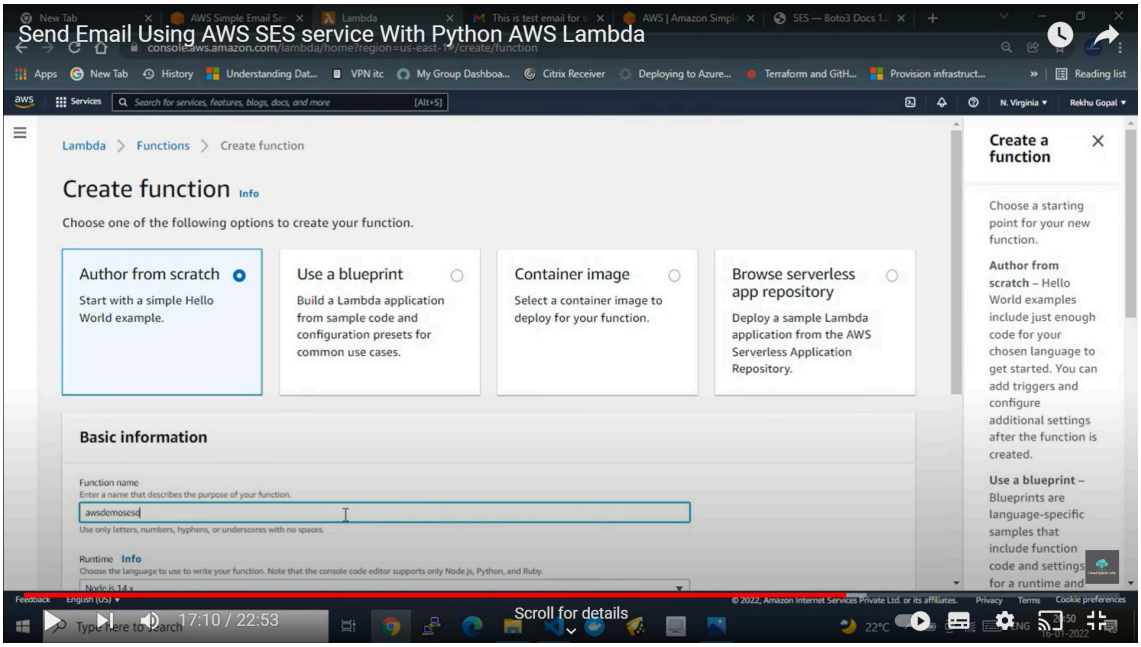

Send Email Using AWS SES service With Python AWS Lambda
Email templates
All templates (0)
Template name
Date created
Loading email templates
11:17 / 22:53
Scroll for details

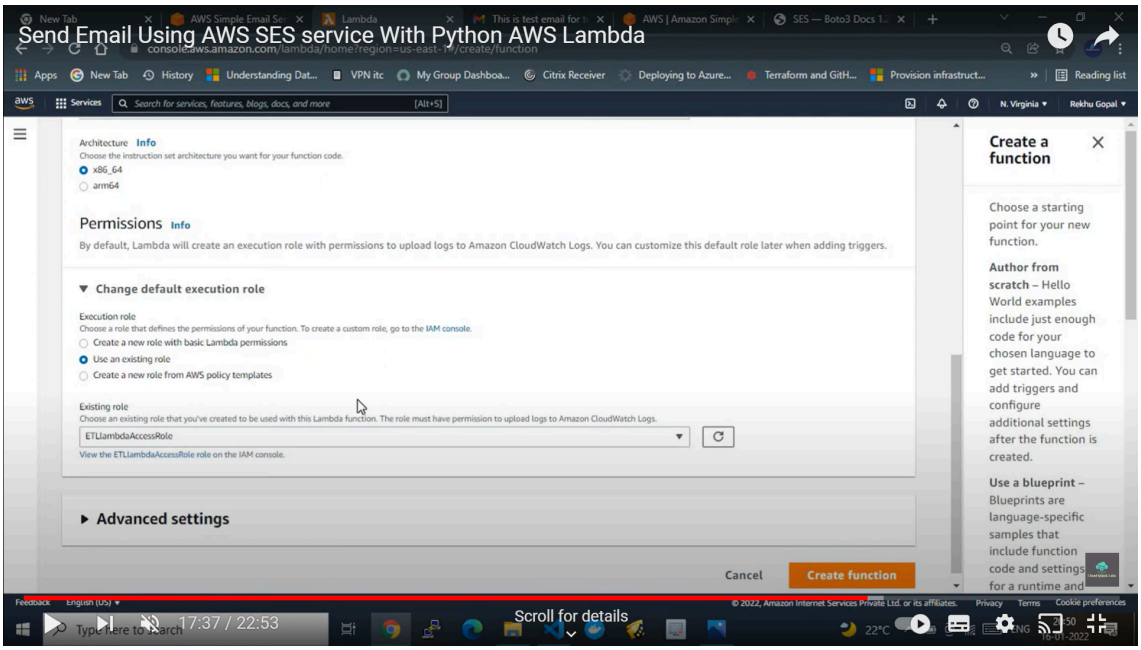

Send Email Using AWS SES service With Python AWS Lambda
Amazon SES
Amazon SES > Configuration: Verified identities
Verified identities
Identities (1)
Send test email
Delete
Create identity
Search all domain and email address identities
Identity
Identity type
Status
vrchinnarathod@gmail.com
Email address
Unverified
11:22 / 22:53
Scroll for details

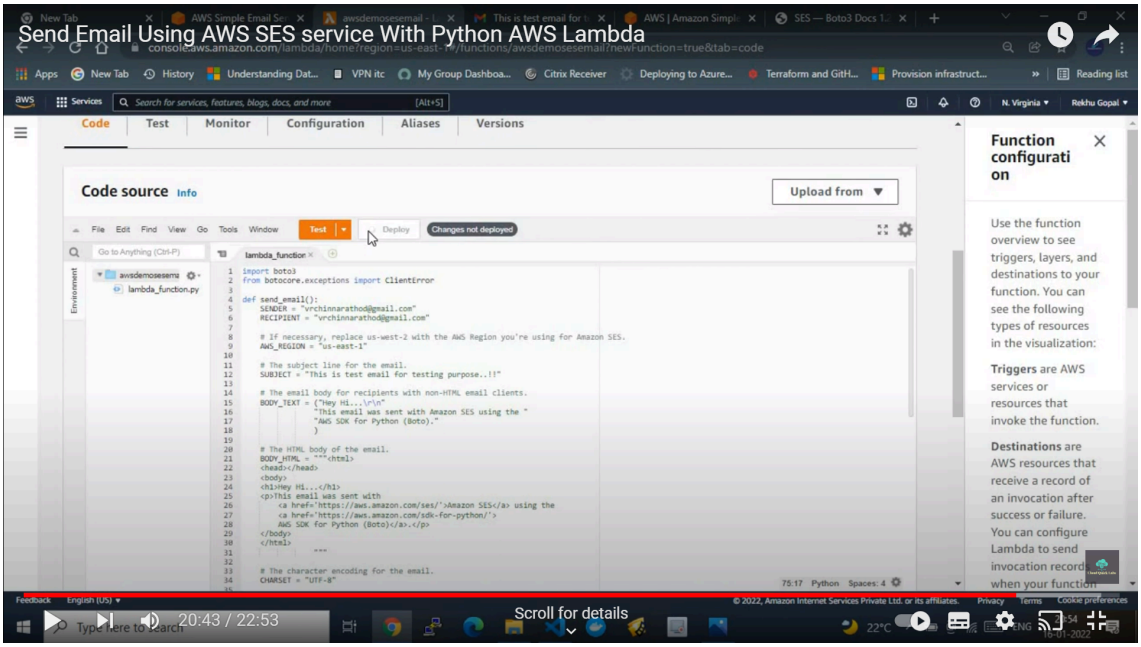

Send Email Using AWS SES service With Python AWS Lambda
Gmail
Search mail
1 of 16,090
Amazon Web Services – Email Address Verification Request in region US East (N. Virginia)
Inbox
Amazon Web Services <no-reply-aws@amazon.com>
to me
8:44 PM (0 minutes ago)
Dear Amazon Web Services Customer,
We have received 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=357171621133&X-Amz-Date=20220116T151418Z&Identity.IdentityName=vrchinnarathod%40gmail.com&X-Amz-Algorithm=AWS4-HMAC-SHA256&Identity.IdentityType=EmailAddress&X-Amz-SignedHeaders=host&X-Amz-Credential=AKIAVM67ZIEFRDECB3HF%2F20220116%2Fus-east-1%2Fses%2Faws4_request&Operation=ConfirmVerification&Namespace=Bacon&X-Amz-Signature=4822cde429224188efc96ee4ba018521e20e1cc73475c8ede9da2e55105c9111
Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original
11:46 / 22:53
Scroll for details

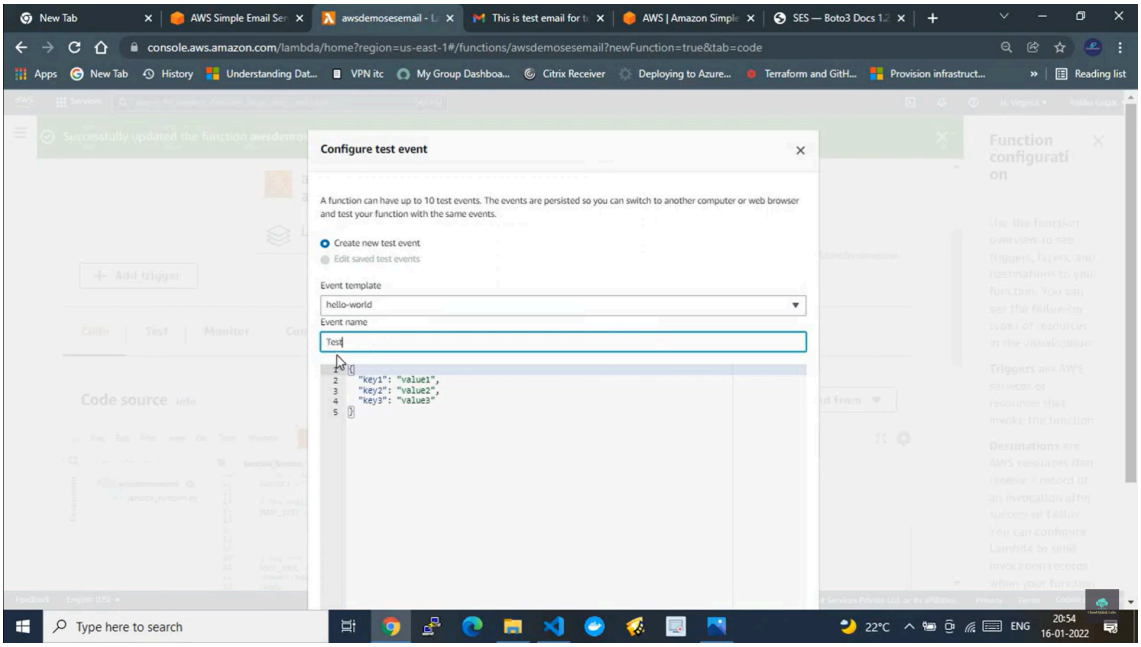

Send Email Using AWS SES service With Python AWS Lambda
Congratulations!
You have successfully verified an email address. You can now start sending email from this address.
For new Amazon SES users—If you have not yet applied for a sending limit increase, then you are still in the sandbox environment, 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!
Go to the Amazon SES detail page.
Get Started with AWS
Learn how to start using AWS in minutes
AWS Free Tier
Gain free, hands-on experience with AWS for 12 months
12:01 / 22:53
Scroll for details

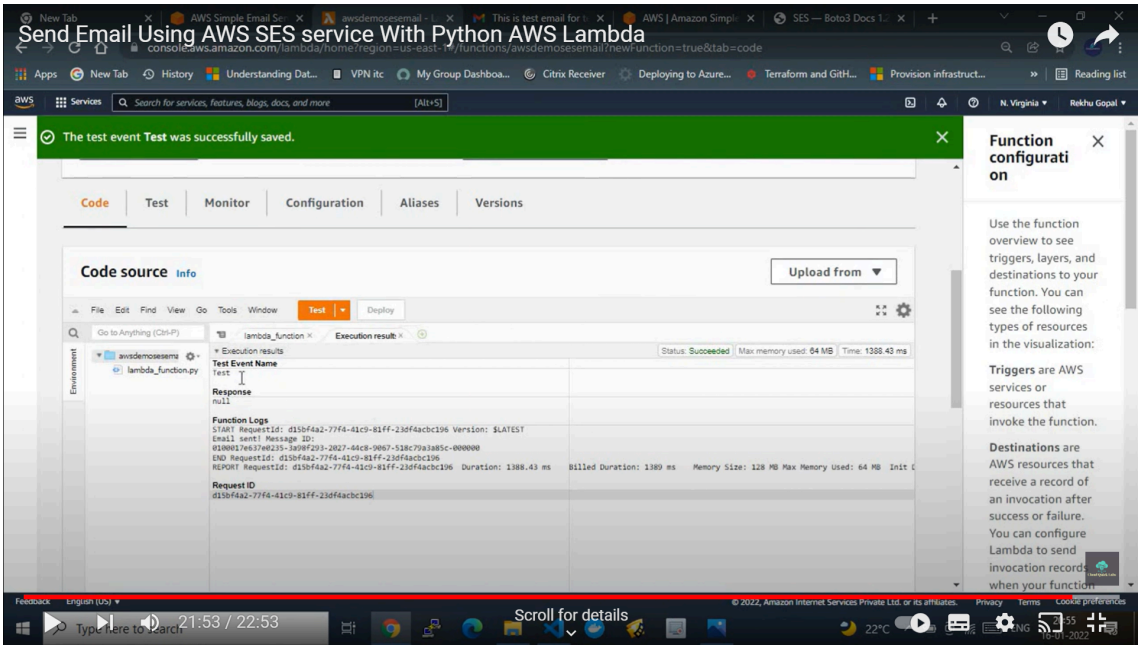

Send Email Using AWS SES service With Python AWS Lambda
Amazon SES
Account dashboard
Reputation metrics
Configuration
Verified identities
Configuration sets
Dedicated IPs
Email templates
Suppression list
Cross-account notifications
Email receiving
Amazon SES > Configuration: Verified identities
Verified identities
Identities (1)
Send test email
Delete
Create identity
Search all domain and email address identities
Identity
Identity type
Status
vrchinnarathod@gmail.com
Email address
Verified
12:15 / 22:53
Scroll for details


Send Email Using AWS SES service With Python AWS Lambda
Lambda > Functions > Create function
Create function
Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
Container image
Select a container image to deploy for your function.
Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.
Basic information
Function name
Enter a name that describes the purpose of your function.
awsdemoses
Use only letters, numbers, hyphens, or underscores with no spaces.
Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 14.x
Create a function
Choose a starting point for your new function.
Author from scratch – Hello World examples include just enough code for your chosen language to get started. You can add triggers and configure additional settings after the function is created.
Use a blueprint – Blueprints are language-specific samples that include function code and settings for a runtime and
17:10 / 22:53
Scroll for details


Send Email Using AWS SES service With Python AWS Lambda
Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
Use an existing role
Create a new role from AWS policy templates
Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
ETLlambdaAccessRole
View the ETLlambdaAccessRole role on the IAM console.
Advanced settings
Cancel
Create function
Create a function
Choose a starting point for your new function.
Author from scratch – Hello World examples include just enough code for your chosen language to get started. You can add triggers and configure additional settings after the function is created.
Use a blueprint – Blueprints are language-specific samples that include function code and settings for a runtime and
17:37 / 22:53
Scroll for details


Send Email Using AWS SES service With Python AWS Lambda
Code source Info
Upload from
Function configuration
Scroll for details


Configure test event
A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.
Create new test event
Edit saved test events
Event template
hello-world
Event name
Test
{
"key1": "value1",
"key2": "value2",
"key3": "value3"
}


Send Email Using AWS SES service With Python AWS Lambda
The test event Test was successfully saved.
Code source Info
Upload from
Function configuration
Scroll for details

In [ ]:
```
1
```