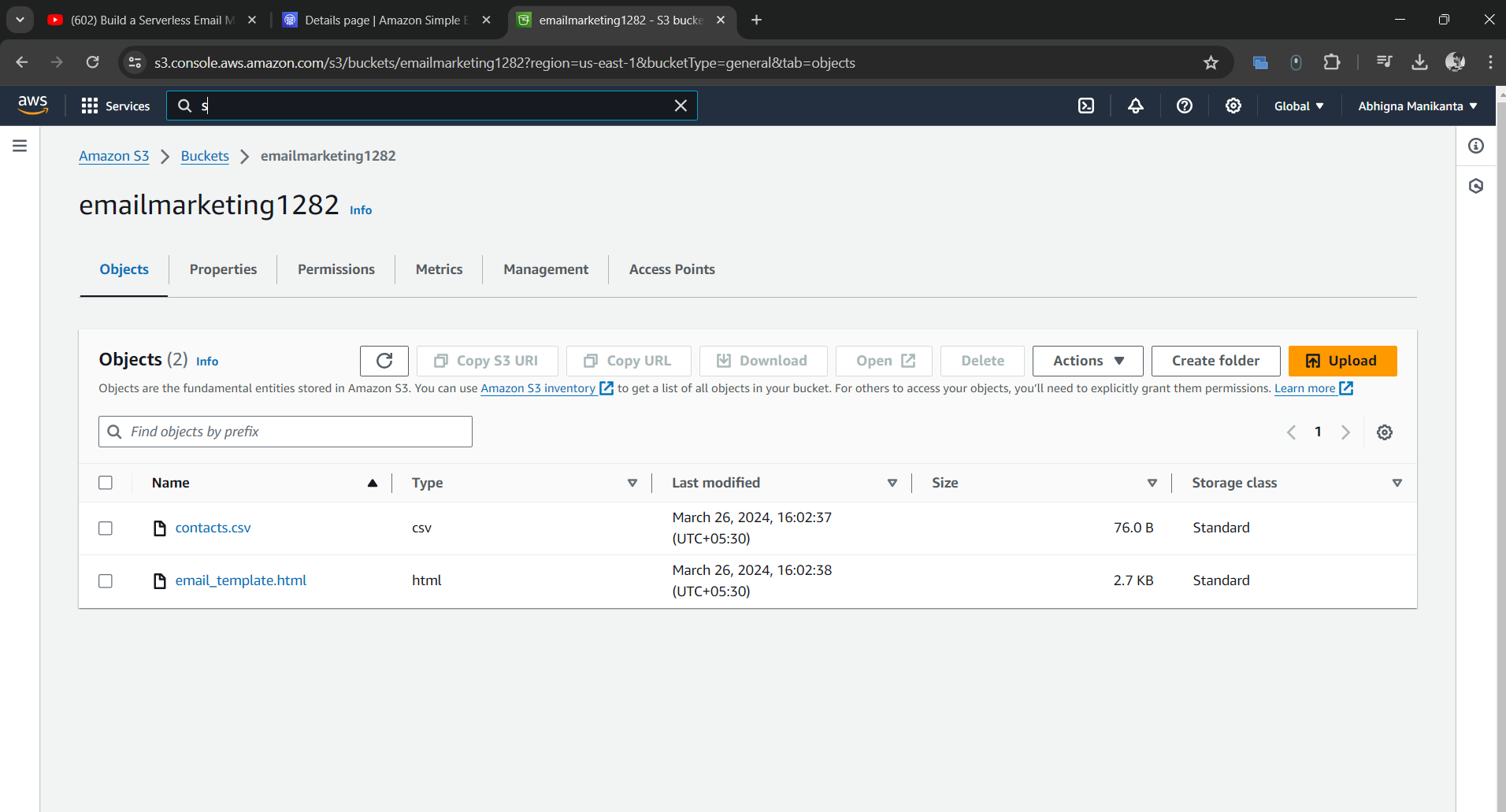
**2100031282 EMAIL MARKETING USING AWS**

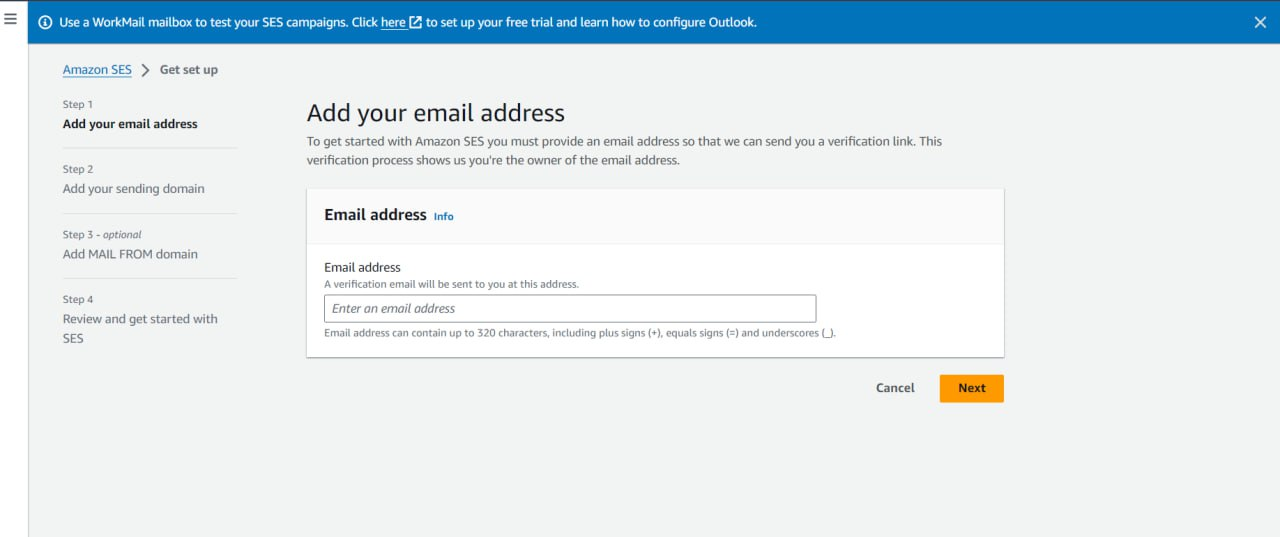
Step-1 GO to AWS s3 create a bucket and upload conacts.csv file and email\_template.html file

Contacts.cs—this file user data to which we send mails

Email\_template \_ consist of html that is the what the mails consists of.



Step\_2 Go to AWS SES give your mail ,sending domain   
make sure you have access to these email addresses as you must go through verification process



After the verification is completed Go to identities and create identities i.e the mails to which you are ging to send and verify them

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedStep 3 Go to AWS lambda and create a function and give the python code and test we might get an error because we have give the policies now we have to create policies

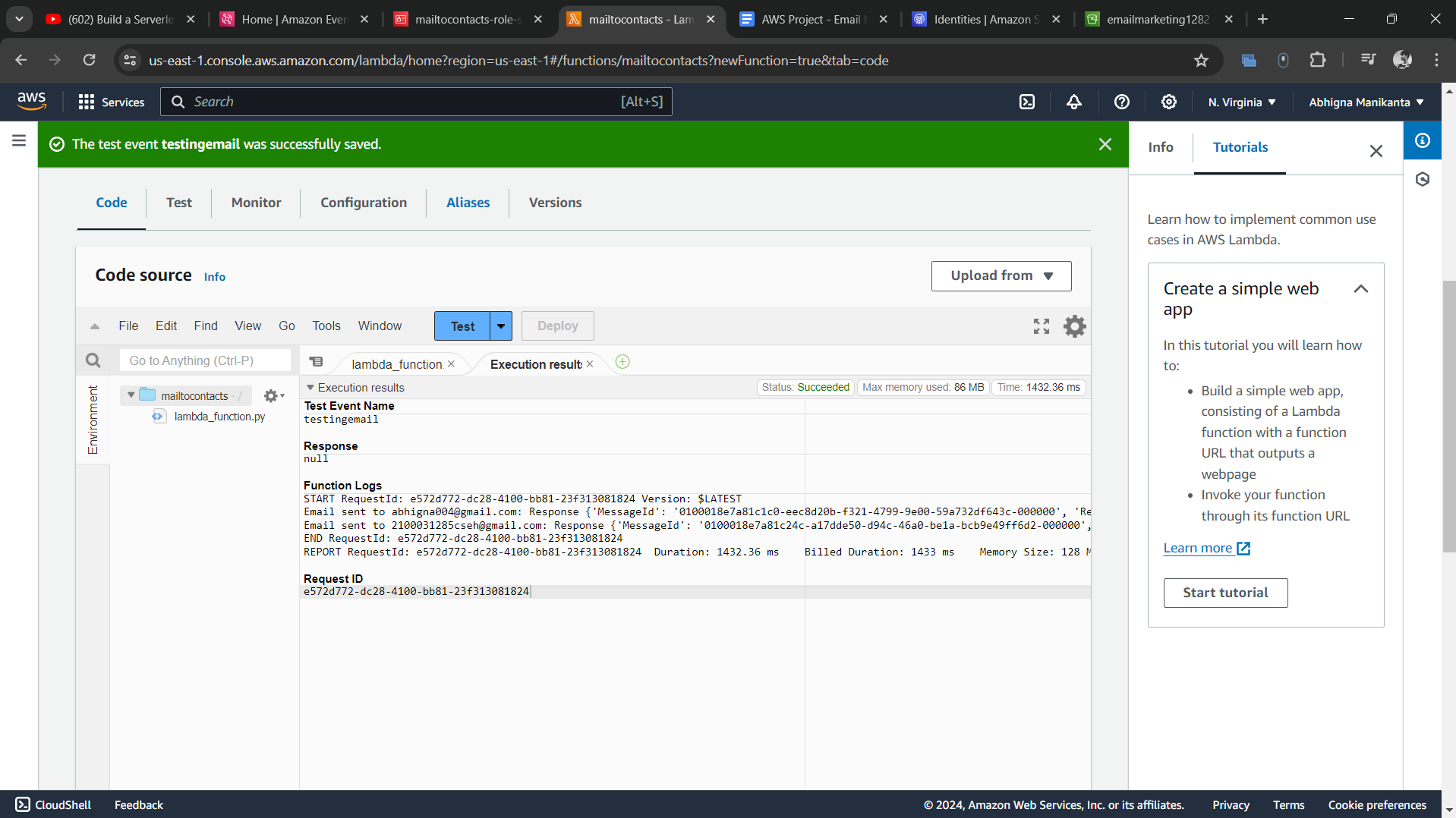
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedStep 4- Create Policy , so that our lambda function can access data in S3 and send Emails using SES in lambda configurations click create policy  
and create and test the lamda code again this time it shows no error and below is a picture of mail successfully sent

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

Step\_5 to enchance the system we can use AwS event bridge to schedule the timings(when to send a mail) for triggering Lambda

Go to AWS Event bridge and click create schedule and according to usecase like you want to send mails on daily basis or weekly schedule your t

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

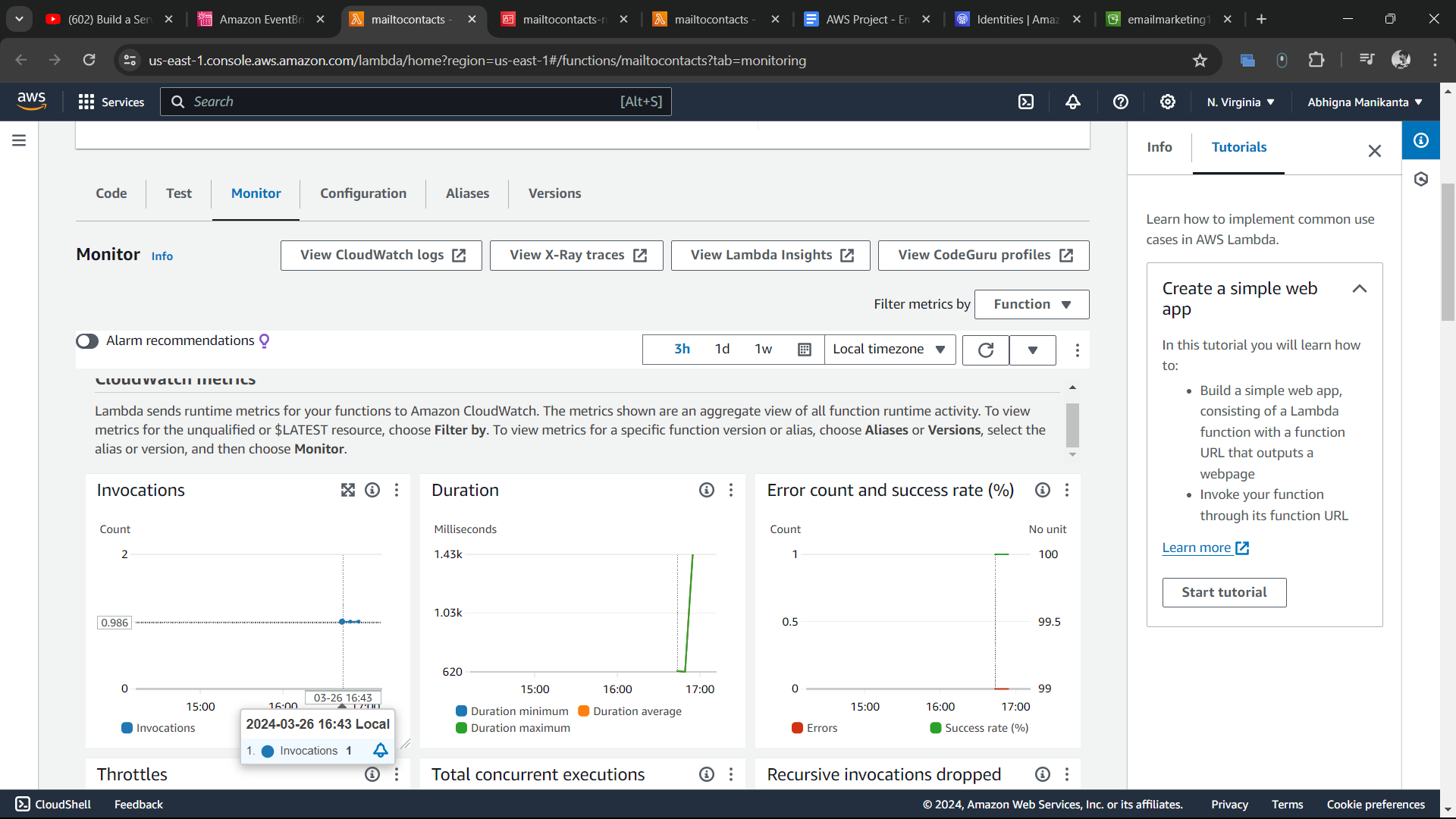
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

You can monitor the success and errors in cloud watch log i.e when the invocations are done and whether they are successful or not.



A screenshot of a computer

Description automatically generated

