The scalability and parallel execution available within Lambda can be used to execute common penetration tests faster.

The following lab will walk through taking a deployment package of nmap and use it as a lambda function.

The simplest Lambda functions consist of only a single file which acts as the lambda handler. If the function only uses libraries that are packaged with the runtime, no code is needed beyond the handler itself.

Non-standard libraries, such as the nmap lib, or executables, such as the nmap binaries, need to be included as part of a "deployment package".

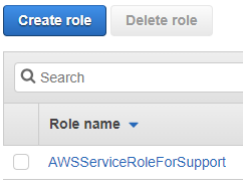
Since Lambda runs in containers on top of the Amazon Linux OS, deployment packages are most likely to be successful if built on an Amazon Linux EC2 Instance.

This function will trigger from an SNS notification, scan the intended host, and store the results to an S3 bucket.

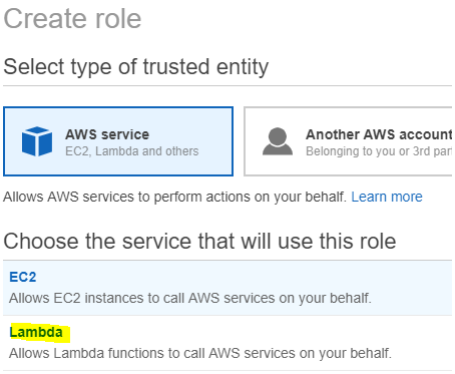
Roles

In the AWS Console, browse to the "IAM" service and click the "Roles" link.

Then create an IAM Role for the Lambda Function to execute as, by clicking the "Create role" button



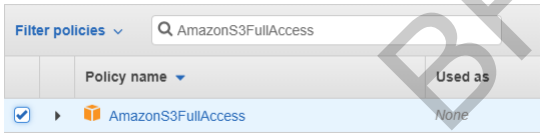
Ensure that the Trusted Entity chosen is "Lambda"...



Click the "Next: Permissions" button...

Grant the role the following Managed Policies:

- CloudWatchLogsFullAccess
- AmazonSNSReadOnlyAccess
- AmazonS3FullAccess

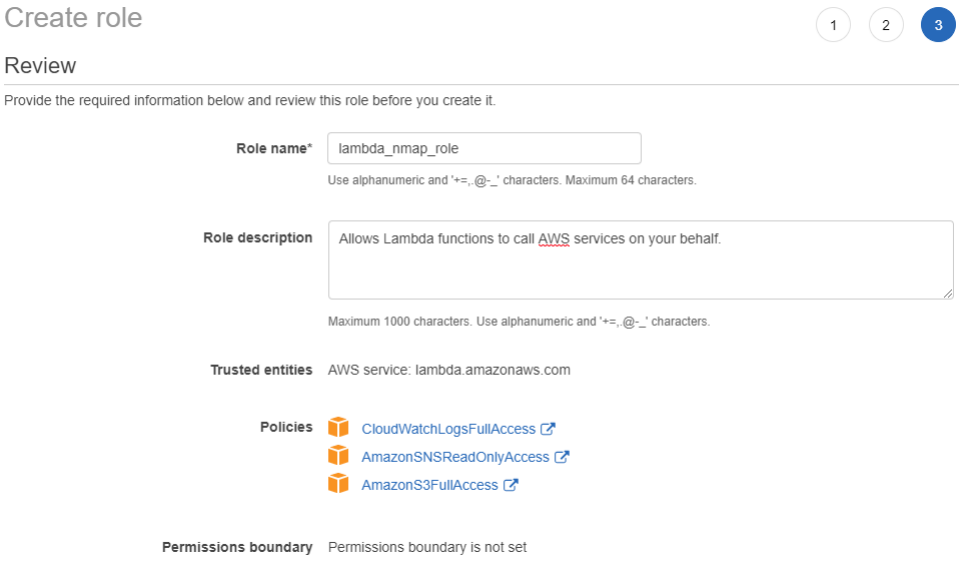


Click the "Next: Tags" button...

Click the "Next: Review" button...

Name the role: "lambda_nmap_role"

You final screen before you click the "Create role" button should look similar to this:

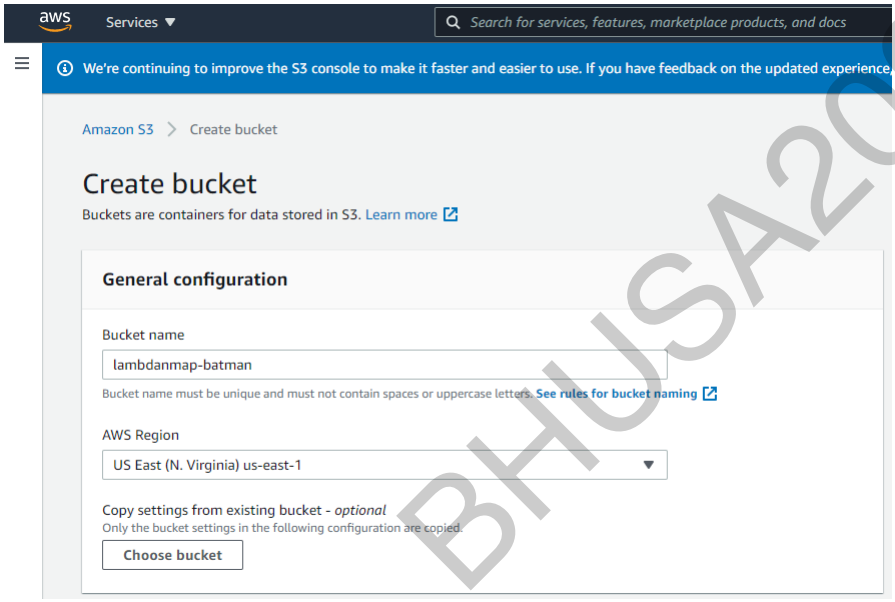


S3 Bucket

Browse to the "S3" Service.

Create an S3 Bucket (e.g. "lambdanmap-batman") in the US East (N. Virginia) us-east-1 region...

Note: S3 Bucket Names are globally unique across all AWS accounts, so replace "batman" with a string that is unique.



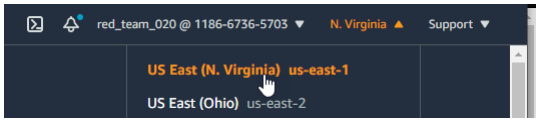
Scroll down to the bottom of the page...

Click the "Create bucket" button.

Simple Notification Service (SNS)

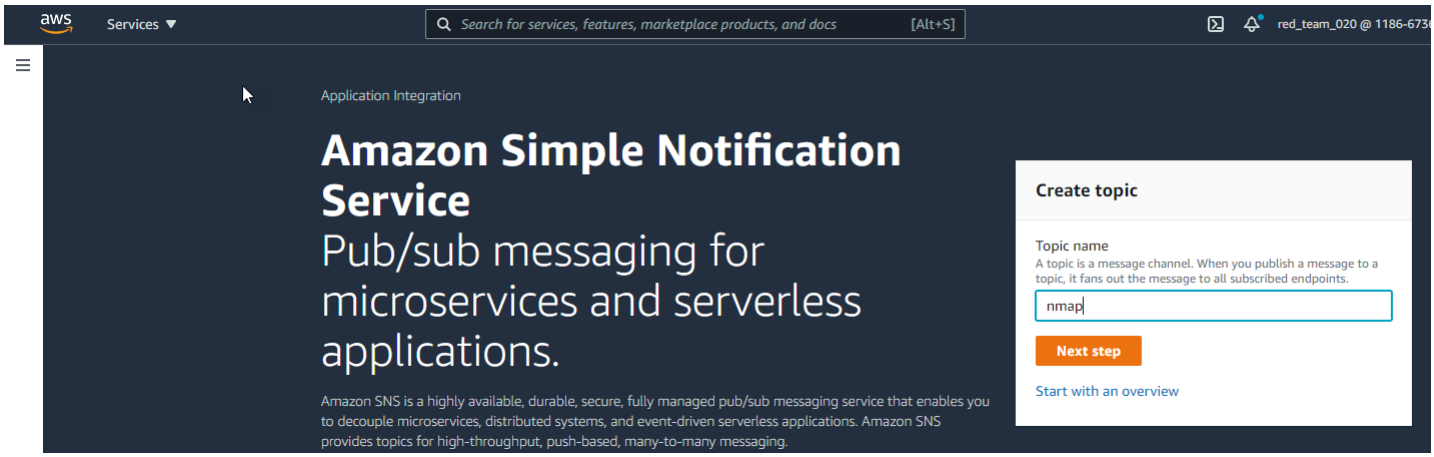
Browse to the "Simple Notification Service" service within the AWS management console.

Change your region to be "N. Virginia" (aka us-east-1):

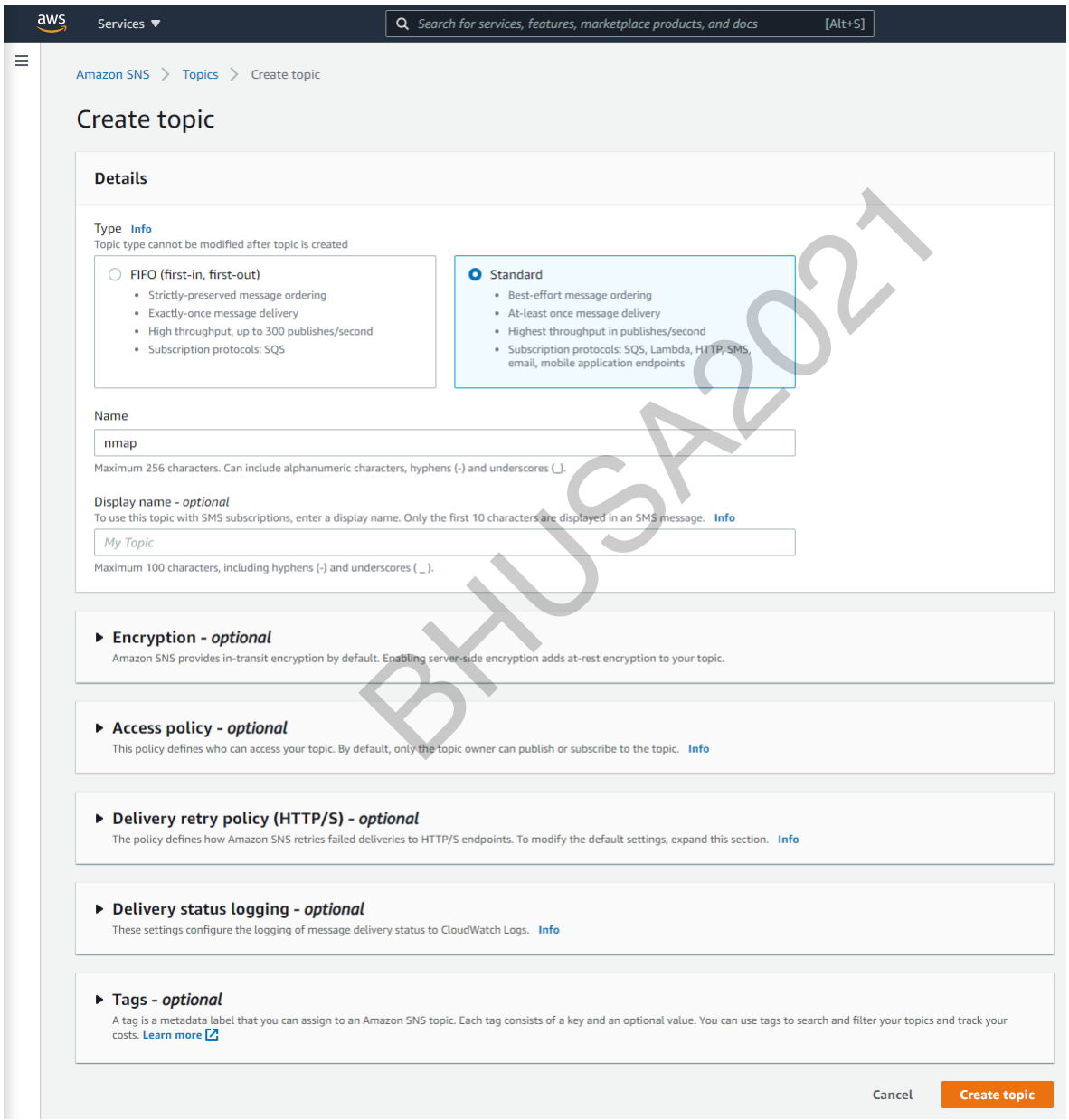


We will create an SNS Topic to trigger the function...

Set the "Topic name" to "nmap" and then click the "Next step" button...

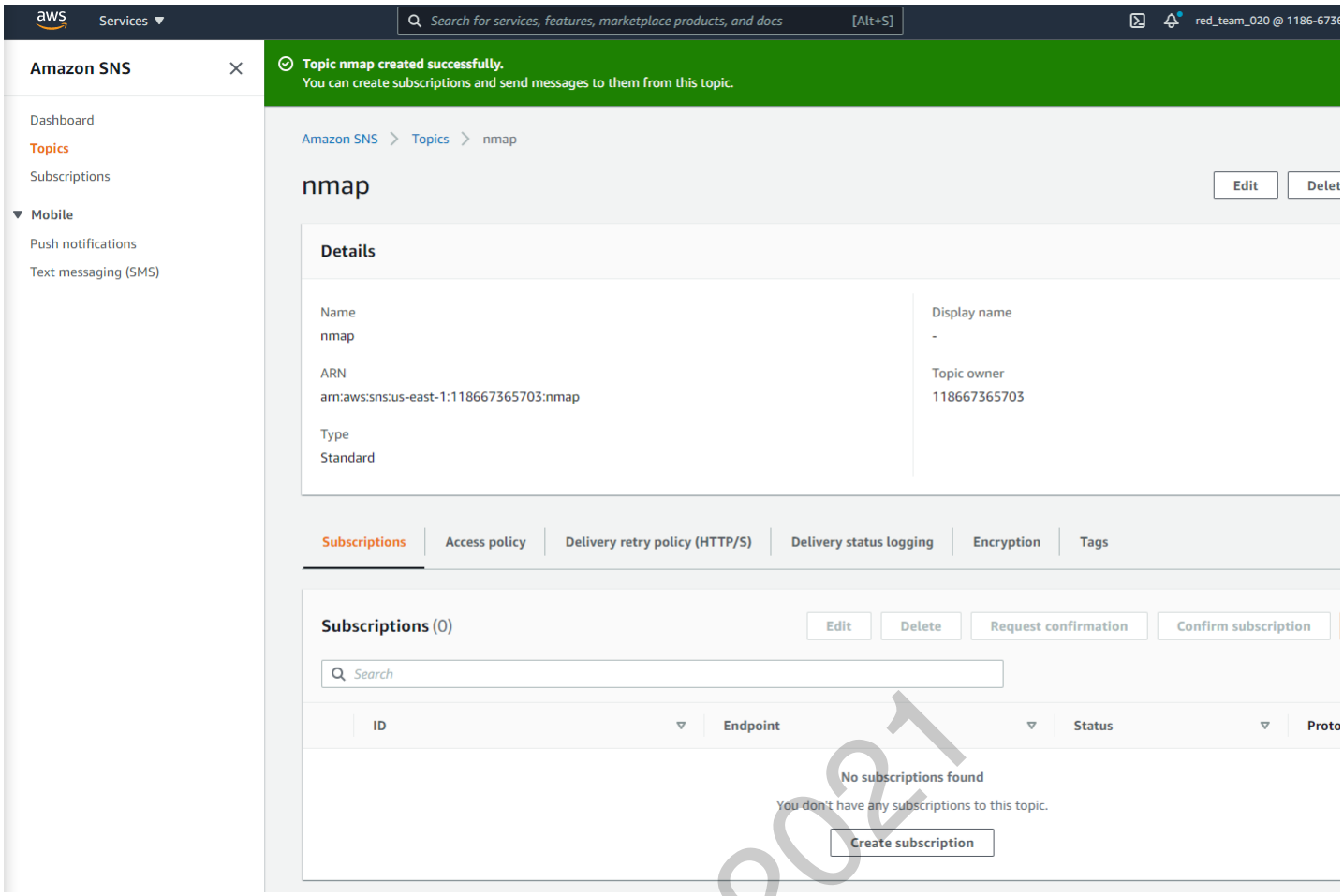


Scroll to the bottom, accepting the default settings...

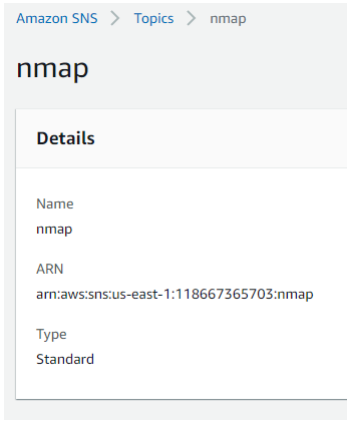


Click the "Create topic" button

We should now see a screen similar to the following...



Make note of the ARN of the newly-created SNS topic in a text editor on your laptop (e.g. "arn:aws:sns:us-east-1:118667365703:nmap"):



Now let's create the Lambda function, by browsing to the "Lambda" service within AWS.

A lambda deployment package includes any runtime dependencies and required binaries for the function to execute. Python Deployment Packages can be built by creating a Python Virtual Environment, installing dependencies to the environment, adding a lambda_function.py entrypoint to the site-packages directory, and zipping the entire site-packages directory.

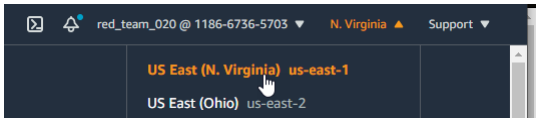
One such deployment package has been prepared and is available:

https://public-astute-cloud-20200813-935672326788.s3.amazonaws.com/nmap_deployment-v1_1_0.zip

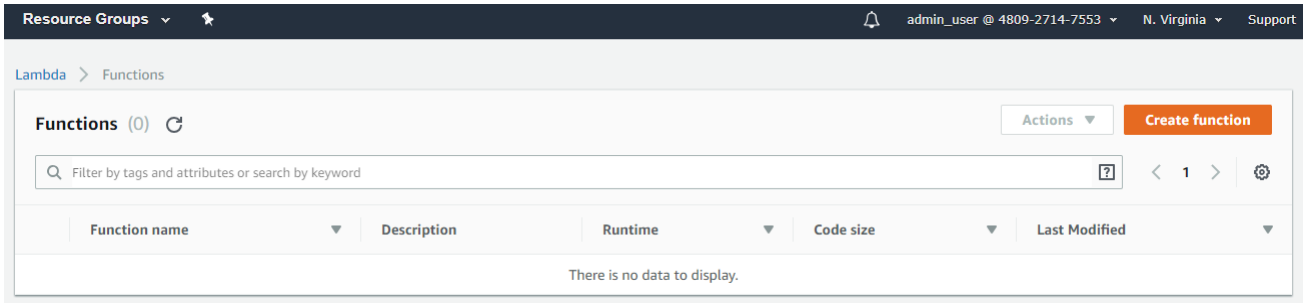
- Extract the files from this zip folder
- Open the "lambda_function.py" within a text editor locally on your laptop
- Review the lambda function in the package: nmap_deployment-v1_1_0.zip/lambda_function.py
- Note that the S3 bucket to store the results into is retrieved from an Environment Variable on line 5.
- Note that nmap binary built into the deployment package at nmap_deployment-v1_1_0.zip/bin/nmap is added to the path on line 20.

In the AWS Console open the Lambda service

Ensure you are in the US East (N. Virginia) us-east-1 region:

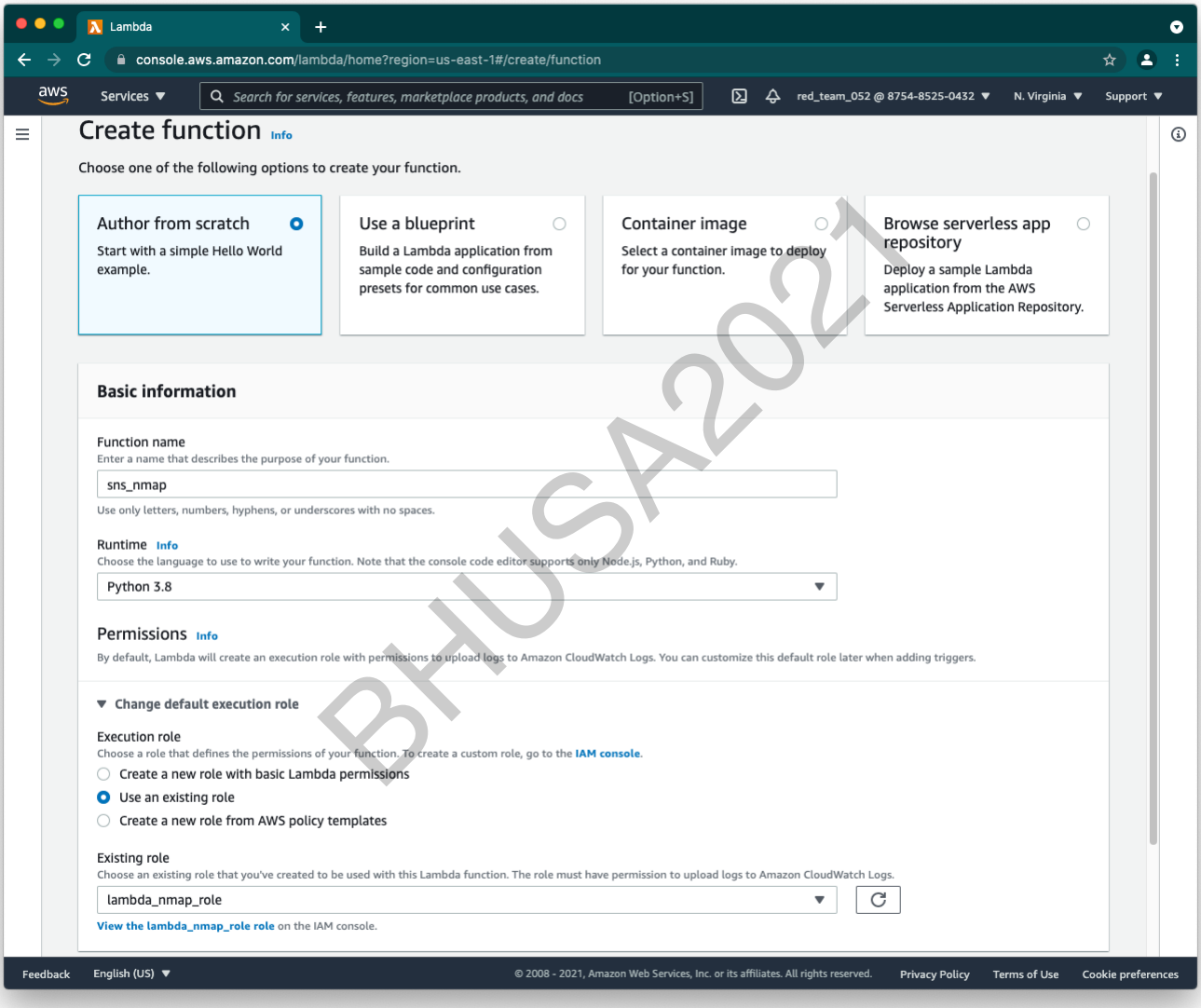


Click the "Create function" button to create a new Lambda Function from Scratch...

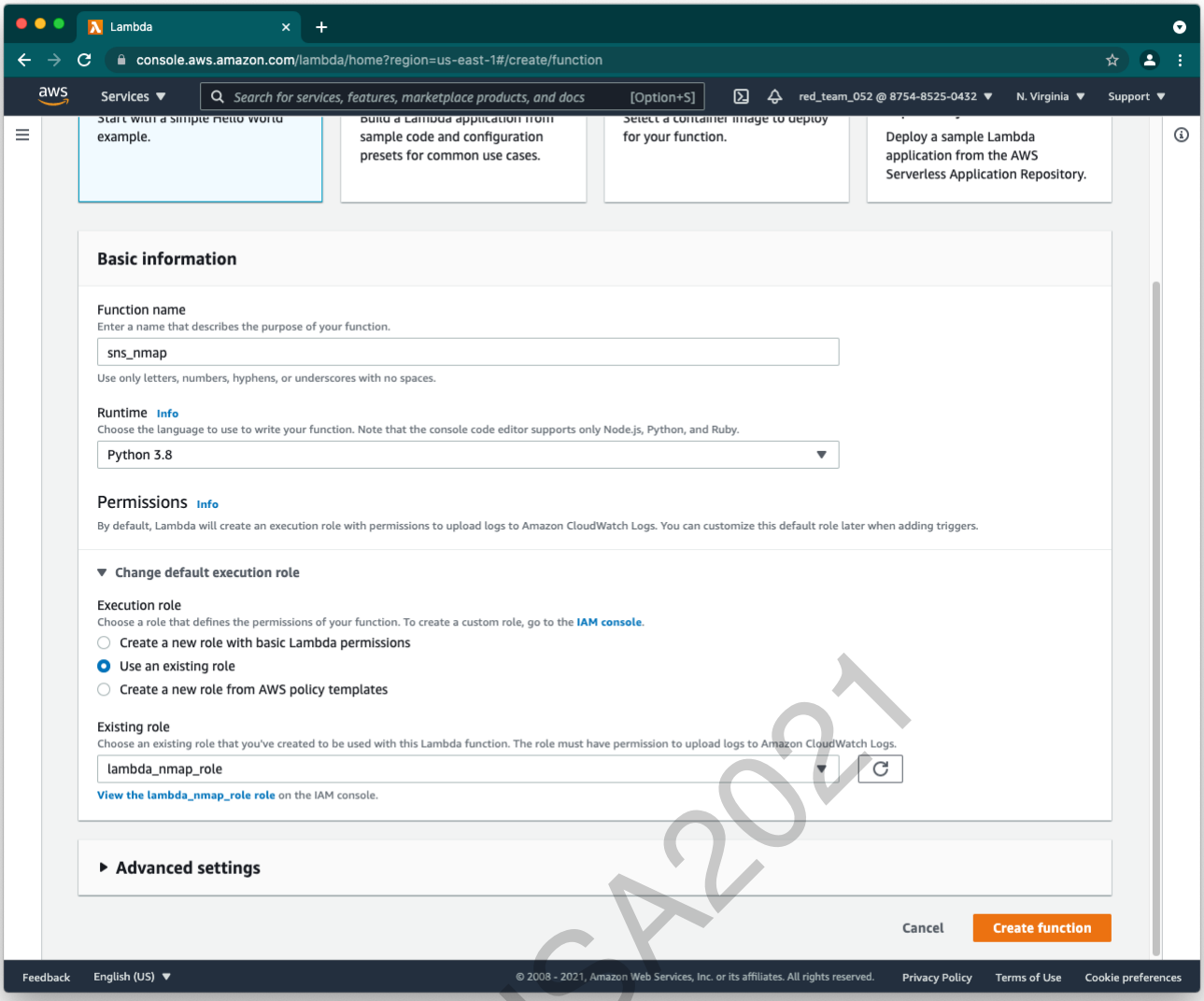


Use the following options for Name, Runtime, and Role.

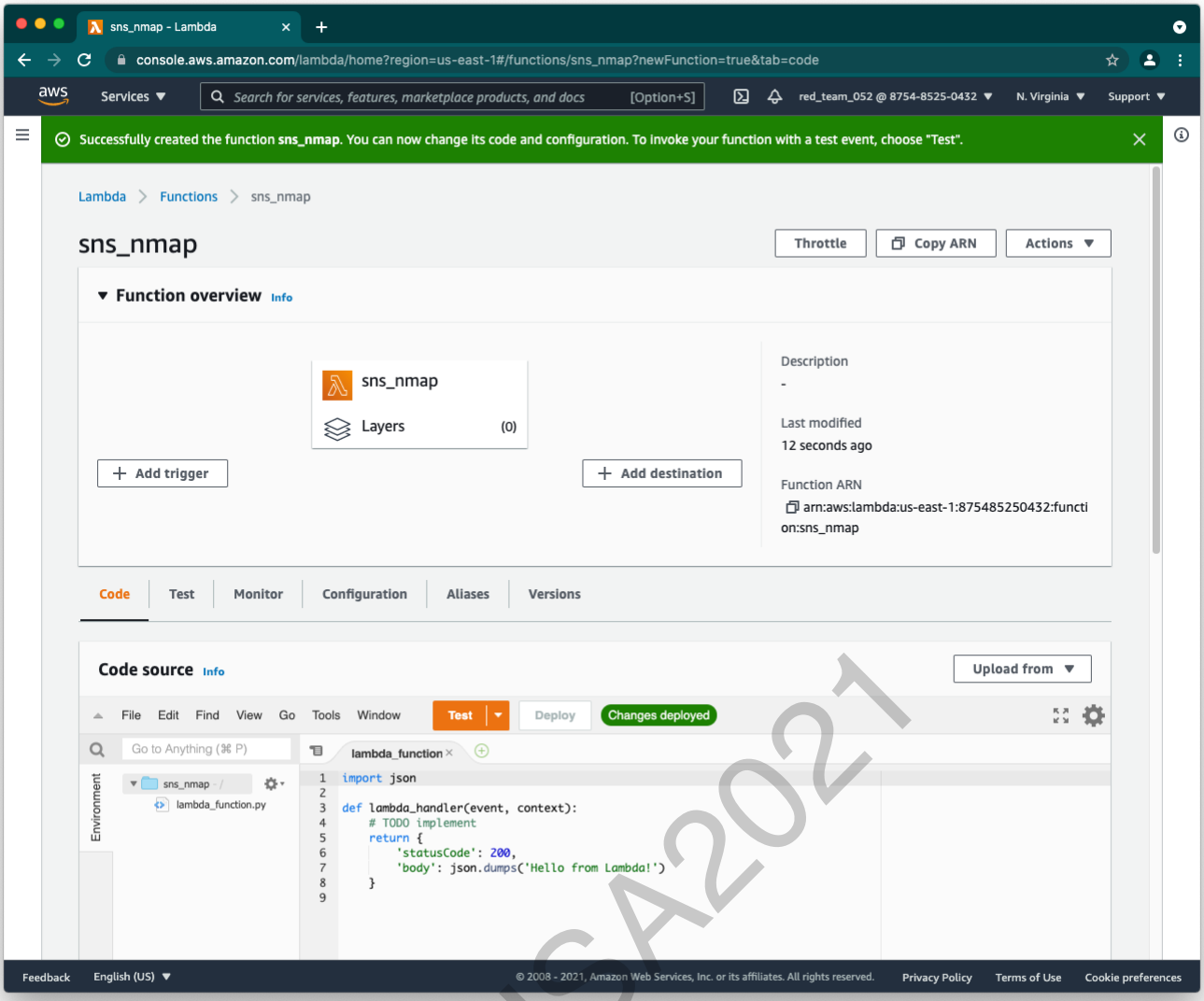
- Function name: sns_nmap
- Runtime: Python 3.8
- Permissions
  - Execution role: Choose an existing role
    - Existing role: lambda_nmap_role



Click the "Create function" button at the bottom of the page...

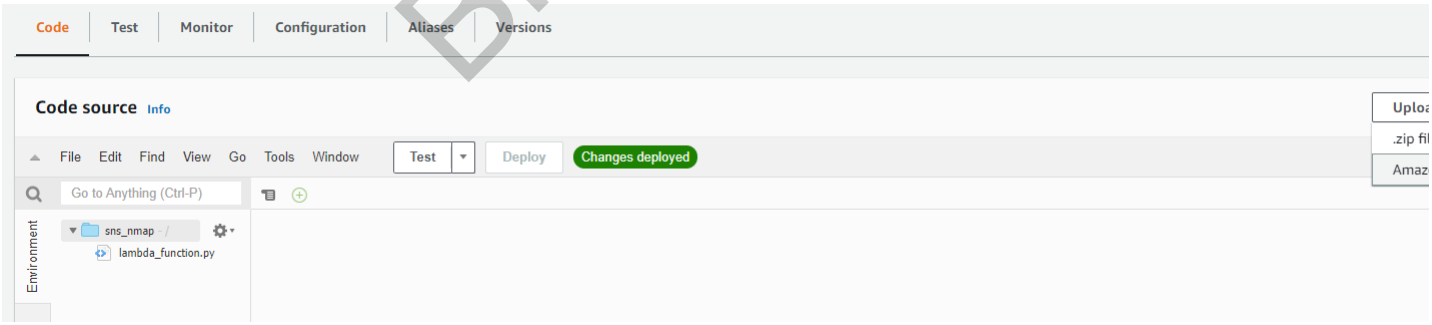


We should now see a screen similar to this...



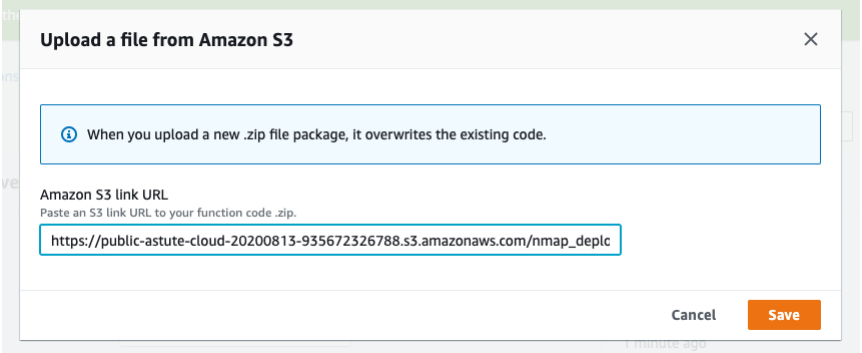
Click the "Upload from" button...

Then Click the "Amazon S3 location" link...



Input the following URL...

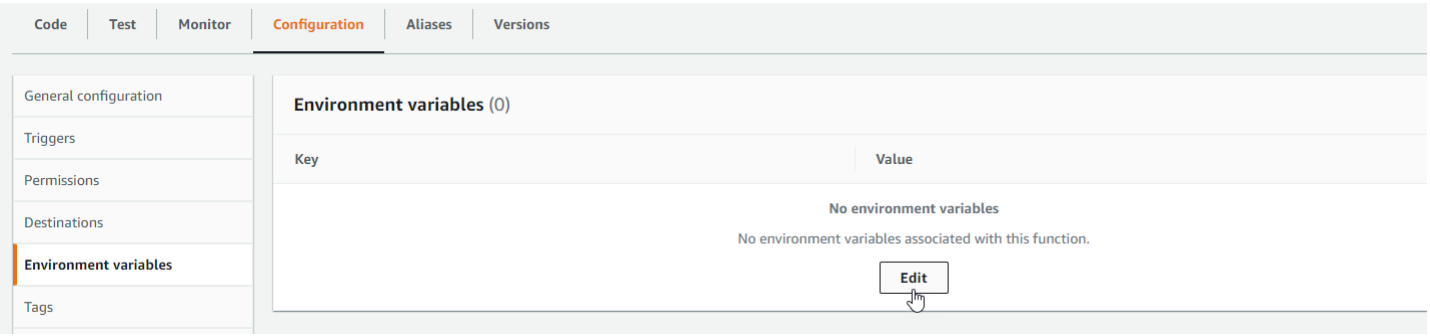
- Amazon S3 link URL: https://public-astute-cloud-20200813-935672326788.s3.amazonaws.com/nmap_deployment-v1_1_0.zip



Click the "Save" button.

Click the "Configuration" link

Click the "Environment variables" link

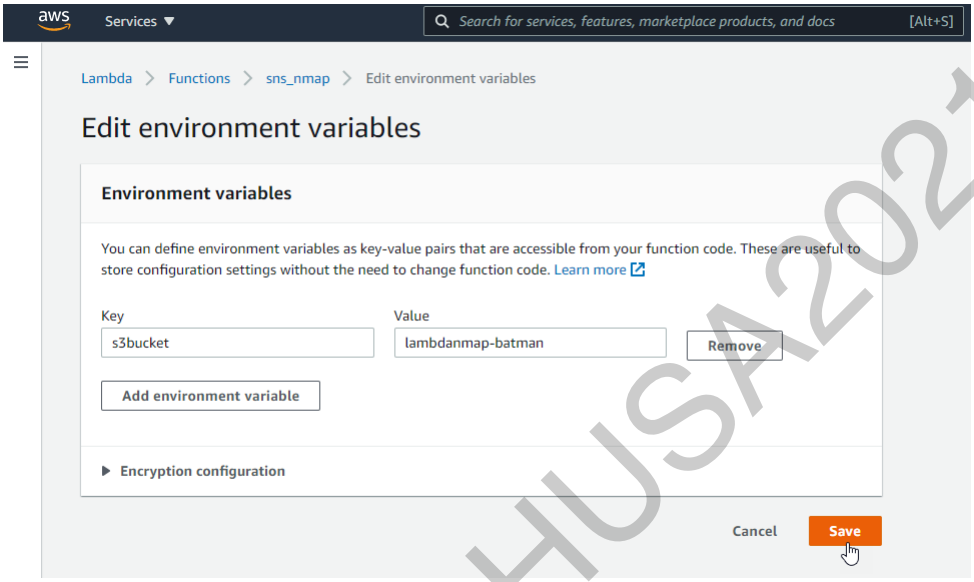


Create an environment variable named "s3bucket" with a value of the exact name of the S3 bucket you created earlier.

Click the "Edit" button

Click the "Add environment variable" button

- Key: s3bucket
- Value: lambdanmap-your_unique_string (e.g. lambdanmap-batman)

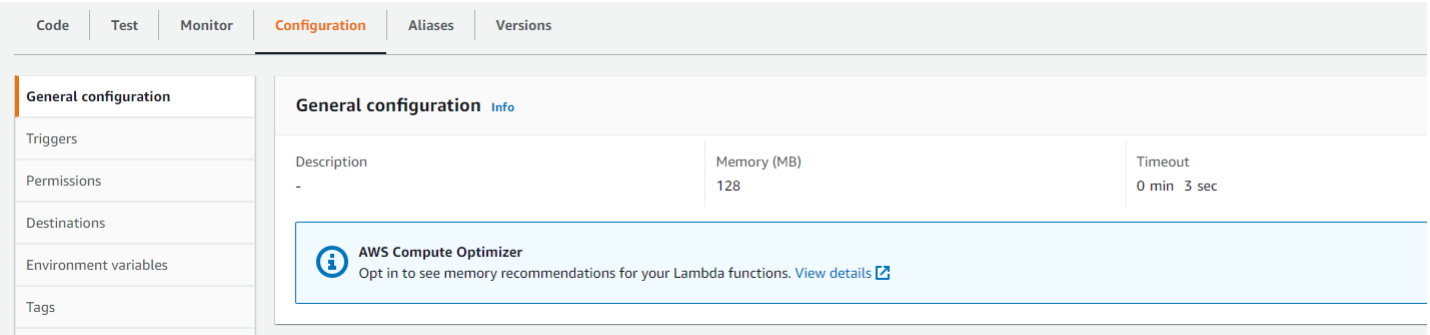


Click the "Save" button

Increase the memory allotment and maximum execution time...

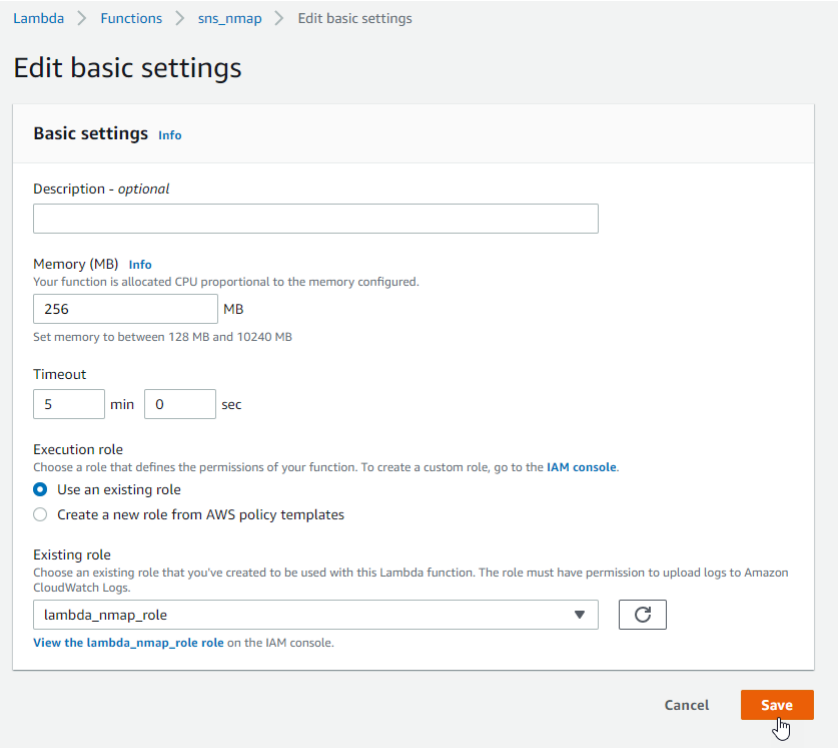
Click the "Configuration" link

Then, click the "General Configuration" link



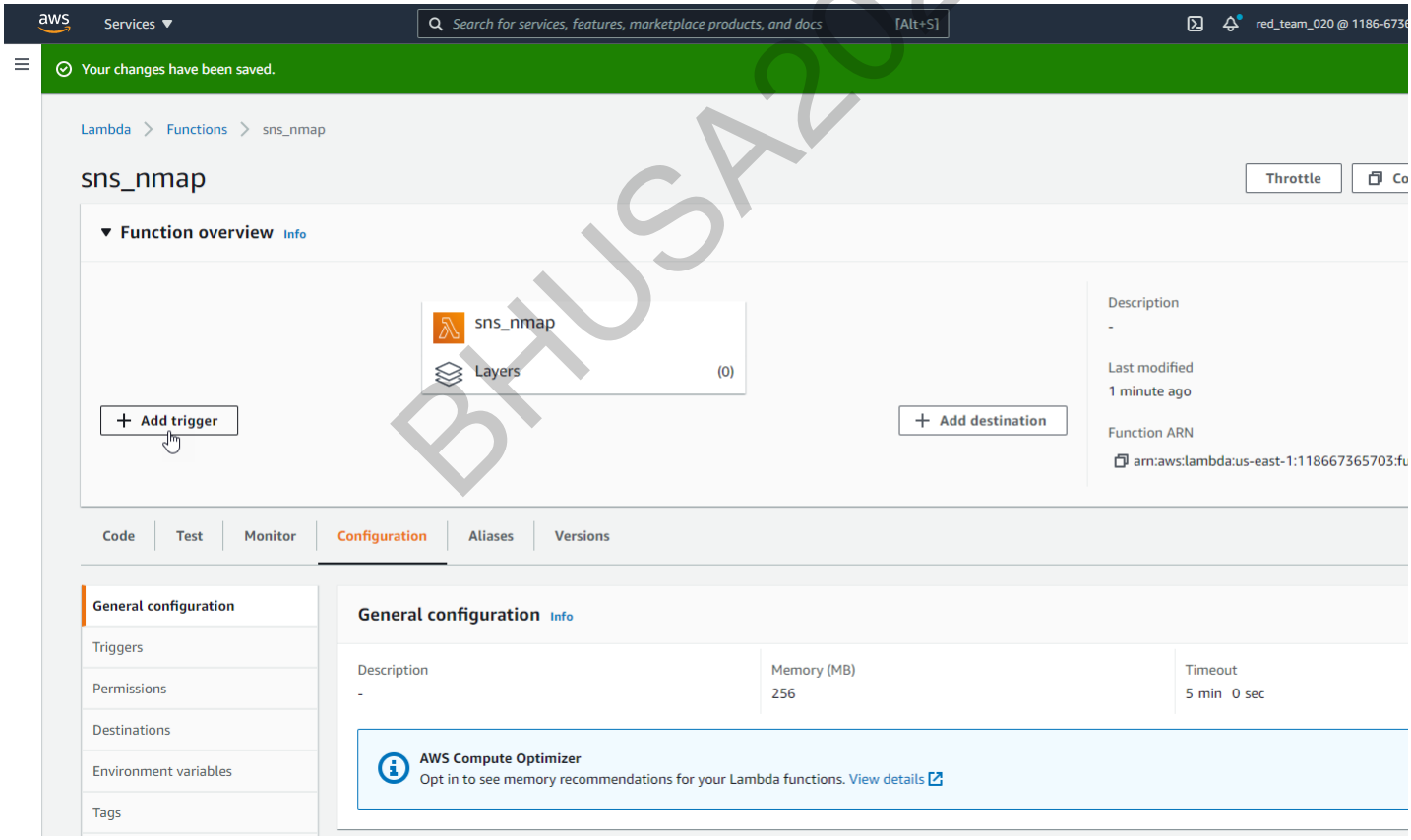
Click the "Edit" button...

- Memory: 256 MB
- Timeout: 5 min 0 sec



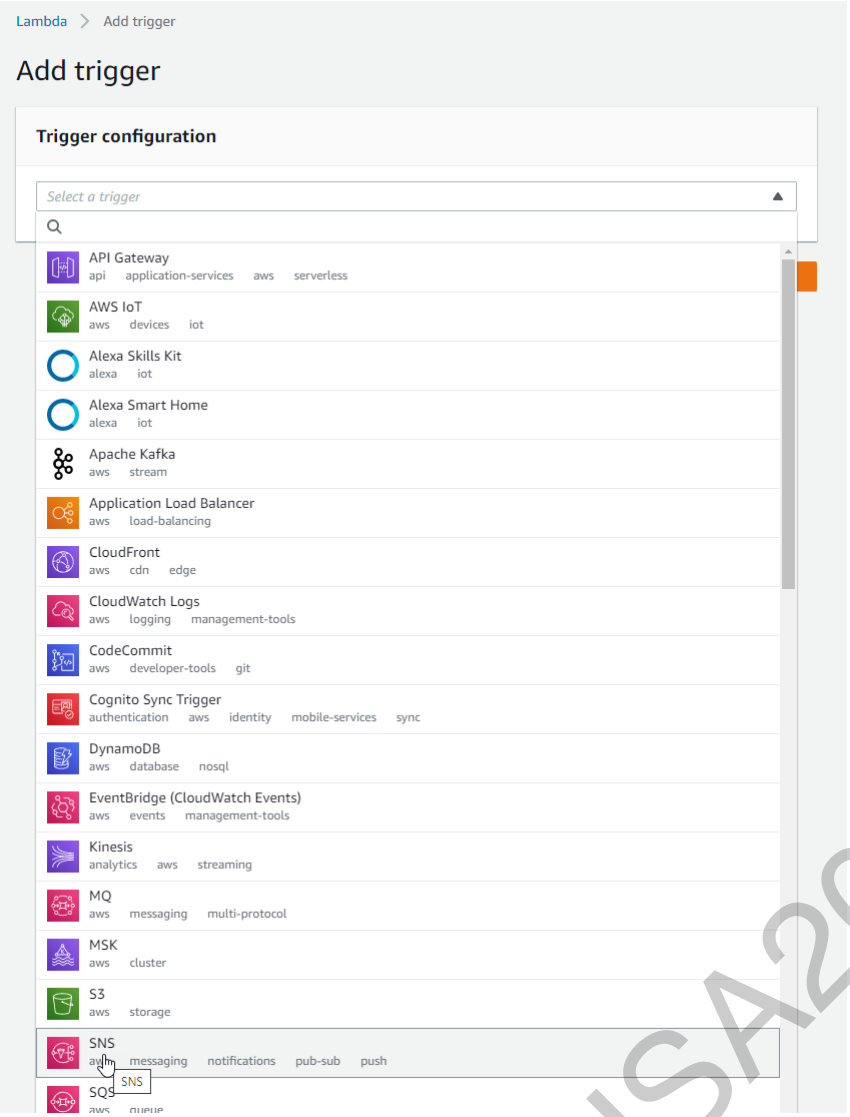
Click the "Save" button

We should now see a screen similar to the following...

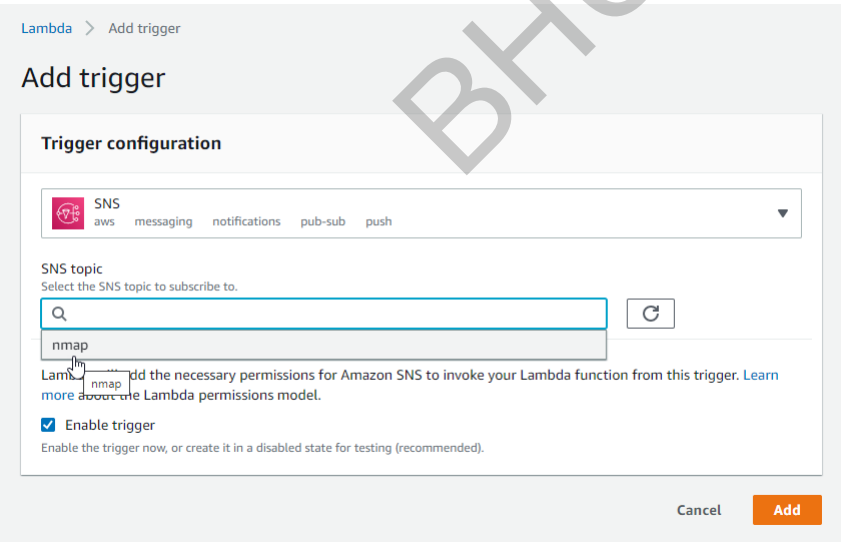


Click the "+ Add trigger" button

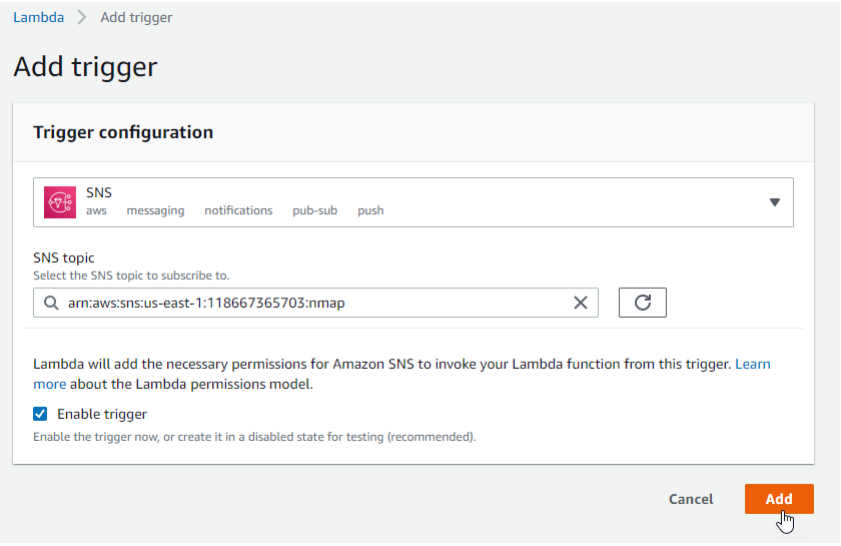
And Select "SNS"...



Select the "SNS topic" of "nmap"...

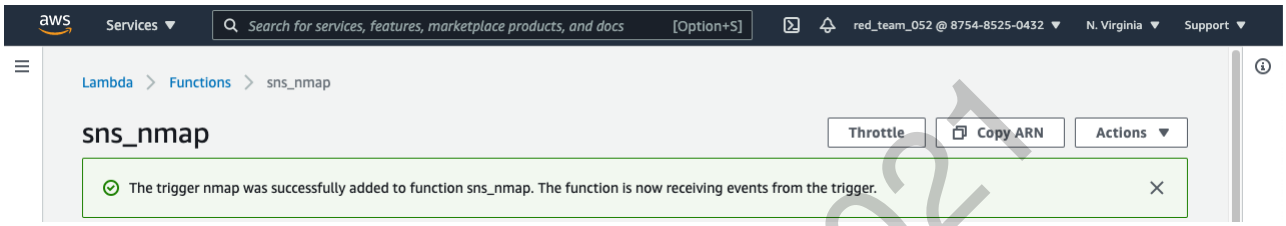


And then it should auto-populate the associated ARN for the selected SNS topic...



Click the "Add" button to add the trigger.

We should now see a banner at the top of the screen that looks similar to the following...



Let's scan the IP address associated with the website "opback.com" using this new lambda function.

```
root@ip-10-0-1-188:/shared# ping -c3 opback.com
PING opback.com (13.82.46.24) 56(84) bytes of data.

--- opback.com ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
```

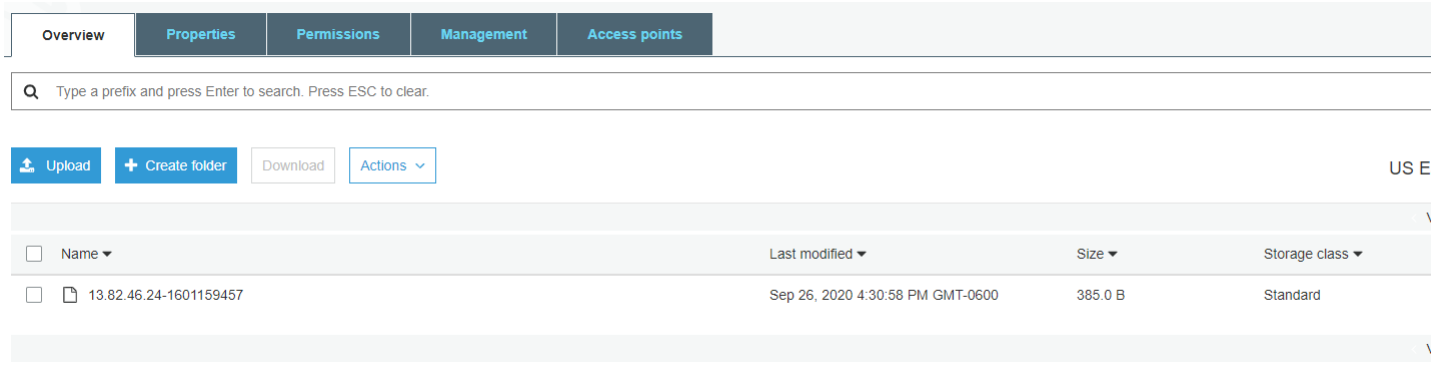
NOTE: The current IP address of the opback.com domain may be different for you (e.g. 13.82.46.24).

Now use the AWS CLI on your machine to push a message to SNS, triggering the lambda functions scan (replacing the string "ip.ip.ip.ip" with the IP address of the opback.com website)

```
root@ip-10-0-1-215:/shared# cd /shared

root@ip-10-0-1-188:/shared# aws sns list-topics --region us-east-1
{
    "Topics": [
        {
            "TopicArn": "arn:aws:sns:us-east-1:480927147553:nmap"
        }
    ]
}
root@ip-10-0-1-215:/shared# aws sns publish --topic-arn arn:aws:sns:us-east-1:480927147553:nmap --region us-east-1 --message ip.ip.ip.ip
{
    "MessageId": "1842feba-86e4-5e96-8058-ad614c244a86"
}
```

After the time it takes for lambda to invoke and run the scan, approximately 5 minutes, you will find the results written to your specified S3 bucket (e.g. "lambdanmap-batman").



The contents of the file should be similar to the following:

```
host;hostname;hostname_type;protocol;port;name;state;product;extrainfo;reason;version;
13.82.46.24;;;tcp;22;ssh;open;OpenSSH;"Ubuntu Linux; protocol 2.0";syn-ack;7.2p2 Ubunt
13.82.46.24;;;tcp;80;http;open;Apache httpd;(Ubuntu);syn-ack;2.4.18;10;cpe:/a:apache:h
13.82.46.24;;;tcp;443;https;closed;;;conn-refused;;3;
```

Thousands of these functions can be invoked simultaneously, significantly speeding up nmap, DNS discovery, directory busting, and similar tasks.

References

Check out the following references for more information:

- Run a Serverless "Hello, World!" - https://aws.amazon.com/getting-started/tutorials/run-serverless-code/
- Create a Simple Lambda Function - https://docs.aws.amazon.com/lambda/latest/dg/get-started-create-function.html
- Getting Started - AWS Lambda - https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html