

A word cloud in the shape of a map of India. The words are in various shades of gray, except for 'ATTACK' and 'DEFENSE' which are in red and blue respectively. The words include: ATTACK, DEFENSE, LABS, COURSES, PENTESTER ACADEMY, RED TEAM, HACKER, TOOL BOX, PATV, ACCESS POINT, WORLD-CLASS TRAINERS, TRAINING, and PENTESTING. The words are arranged to fill the outline of the map, with 'ATTACK' and 'DEFENSE' being the largest and most prominent.

Name	Lambda Enumeration
URL	https://attackdefense.com/challengedetails?cid=2281
Type	AWS Cloud Security : Lambda

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

Console Based Enumeration

Step 1: Click on the lab link to enumerate AWS credentials.

Access Credentials to your AWS lab Account

Login URL	https://276384657722.signin.aws.amazon.com/console
Region	Asia Pacific (Singapore) ap-southeast-1
Username	tlssEgDwcRKRYxxQCSKI
Password	LhNk02qCglhle9f
Access Key ID	AKIAUAWOPGE5GACZM5OT
Secret Access Key	5ICXYt13EJWAbawWvNYT8vHdoeqDCBLhn8qG2Plnz

Step 2: Sign in to AWS console.

Sign in as IAM user

Account ID (12 digits) or account alias

276384657722

IAM user name

tlssEgDwcRKRYxxQCSKI

Password

••••••••••••••••••••

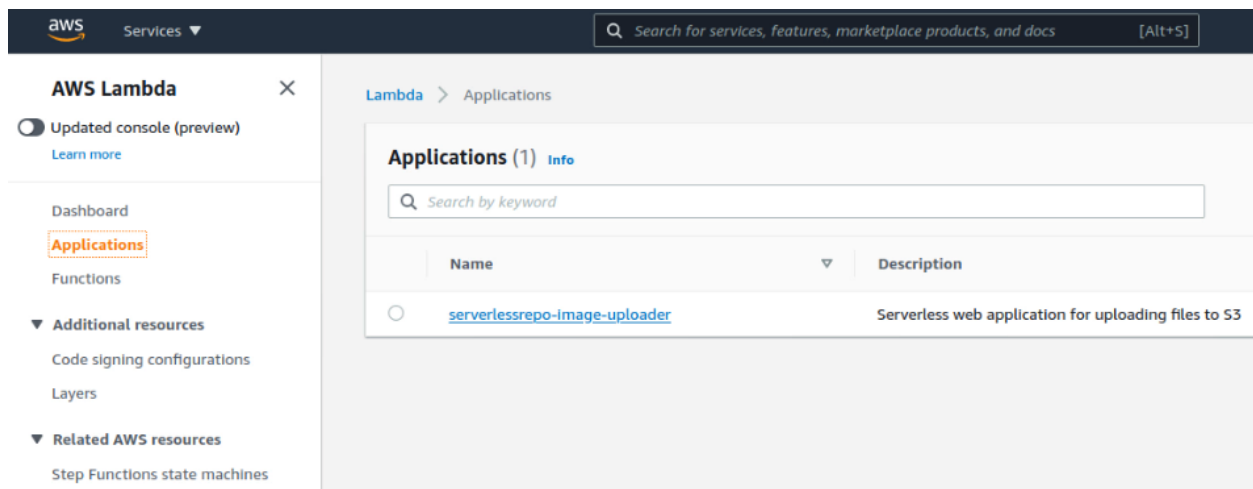
Sign in

[Sign in using root user email](#)

[Forgot password?](#)



Step 3: Navigate to Lambda application dashboard.



Step 4: Enumerate lambda applications. Visit API endpoint.

API endpoint

Endpoint
<https://cwlw44ht84.execute-api-southeast-1.amazonaws.com/Prod>

Resources (8)

Filter by tags and attributes or search by keyword

Logical ID	Physical ID	Type	Last modified
ServerlessRestApi	cwlw44ht84	ApiGateway RestApi	last year
uploader	serverlessrepo-image-uploader-uploader-RM72CSUT4KDA	Lambda Function	last year

<https://cwlw44ht84.execute-api-southeast-1.amazonaws.com/Prod>



Uploaded Files:

Check resources used by the application.

serverlessrepo-image-uploader-uploader-RM72CSUT4KDA Throttle

Configuration | Permissions | Monitoring

Designer

[Go back to application serverlessrepo-image-uploader](#)

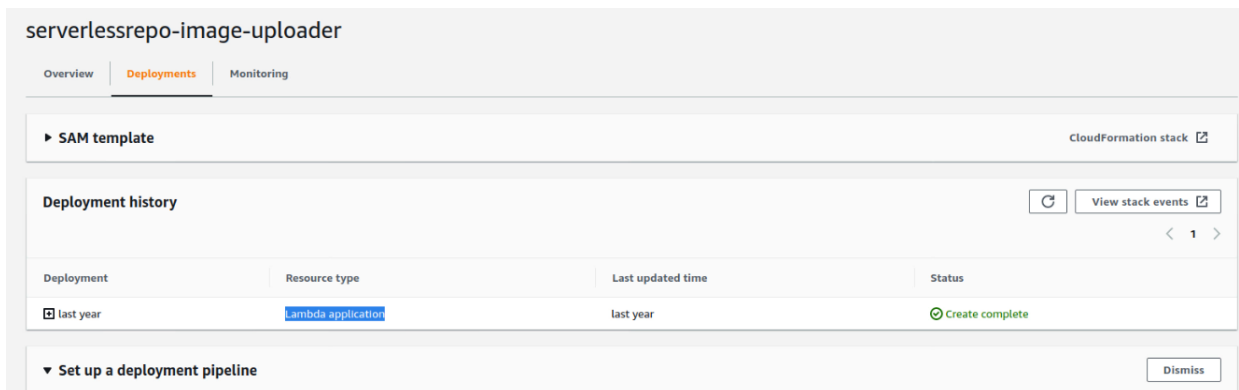
serverlessrepo-image-uploader-uploader-RM72CSUT4 KDA

Layers (0)

API Gateway (3)

[+ Add trigger](#)

Step 5: Check the deployment history.

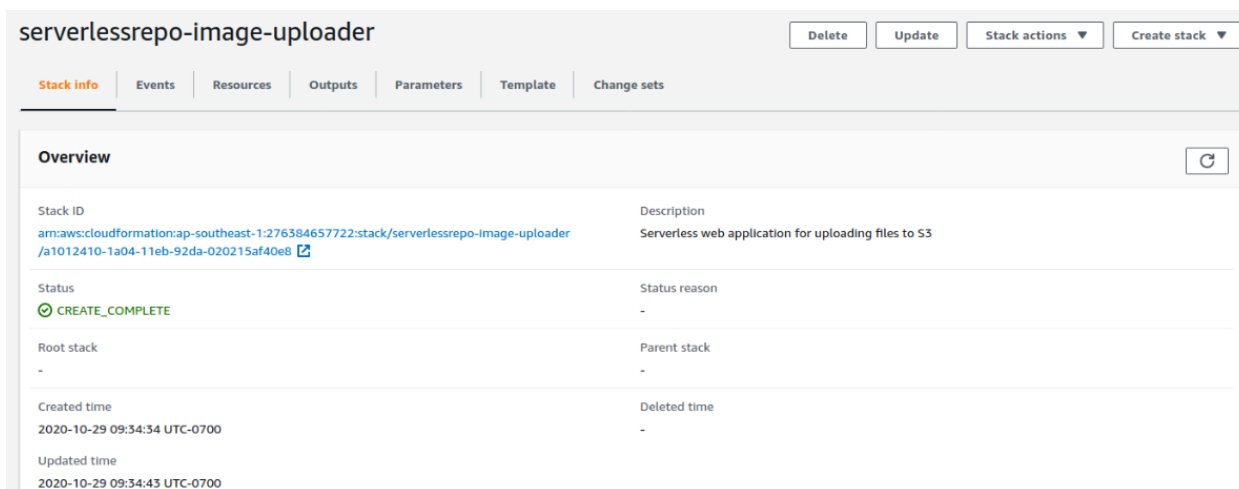


The screenshot shows the AWS SAM console for the stack 'serverlessrepo-image-uploader'. The 'Deployments' tab is selected. Under the 'Deployment history' section, a table lists the deployment details:

Deployment	Resource type	Last updated time	Status
last year	lambda application	last year	Create complete

Buttons for 'View stack events' and 'Dismiss' are visible.

Step 6: Check application deployment stack event logs.



The screenshot shows the AWS CloudFormation console for the stack 'serverlessrepo-image-uploader'. The 'Stack info' tab is selected, showing the 'Overview' section. The stack status is 'CREATE_COMPLETE'.

Property	Value
Stack ID	arn:aws:cloudformation:ap-southeast-1:276384657722:stack/serverlessrepo-image-uploader/a1012410-1a04-11eb-92da-020215af40e8
Description	Serverless web application for uploading files to S3
Status	CREATE_COMPLETE
Status reason	-
Root stack	-
Parent stack	-
Created time	2020-10-29 09:34:34 UTC-0700
Deleted time	-
Updated time	2020-10-29 09:34:43 UTC-0700

Step 7: Click on the layers tab on the navigation panel on the left to enumerate lambda layers.

aws Services Search for services, features, marketplace products, and docs [Alt+S]

AWS Lambda X

Updated console (preview) Learn more

Dashboard Applications Functions

Additional resources Code signing configurations Layers

Related AWS resources Step Functions state machines

Lambda > Layers

Layers (5)

Filter by tags and attributes or search by keyword

Name	Version	Version ARN
FileUploaderLayer	5	arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:5
boto3-jinja	1	arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1
cryptography	1	arn:aws:lambda:ap-southeast-1:276384657722:layer:cryptography:1
php-runtime	4	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:4
php-vendor	3	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-vendor:3

Step 8: Click on the layer name and check compatible runtimes and versions.

php-runtime

Version details

Version	Description	Created
4		14 hours ago

Compatible runtimes

None

All versions

Version	Version ARN
4	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:4
3	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:3
2	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:2
1	arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:1

boto3-jinja

Version details

Version	Description	Created
1		last year

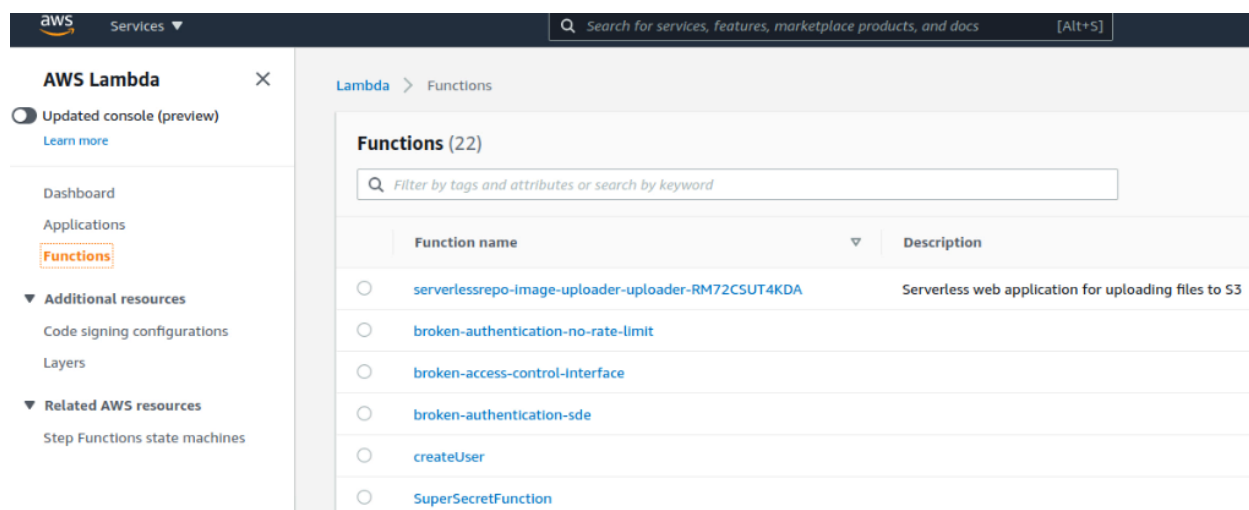
Compatible runtimes

python3.8

All versions

Version	Version ARN
1	arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1

Step 9: Click on the functions tab on the navigation panel on the left to enumerate lambda functions.



The screenshot shows the AWS Lambda console interface. On the left is a navigation panel with the 'Functions' tab selected. The main area displays a list of 22 functions. The first few functions visible are:

Function name	Description
serverlessrepo-image-uploader-uploader-RM72CSUT4KDA	Serverless web application for uploading files to S3
broken-authentication-no-rate-limit	
broken-access-control-interface	
broken-authentication-sde	
createUser	
SuperSecretFunction	

Step 10: Click on the function name and enumerate function's source code, api endpoints and layers.

dom-xss


Throttle


Configuration


Permissions

Monitoring

▼ Designer

 dom-xss

 Layers (1)

 API Gateway

+ Add trigger

dom-xss

Function code Info

File Edit Find View Go Tools Window

Test

Deploy

Changes deployed

Go to Anything (Ctrl-P)

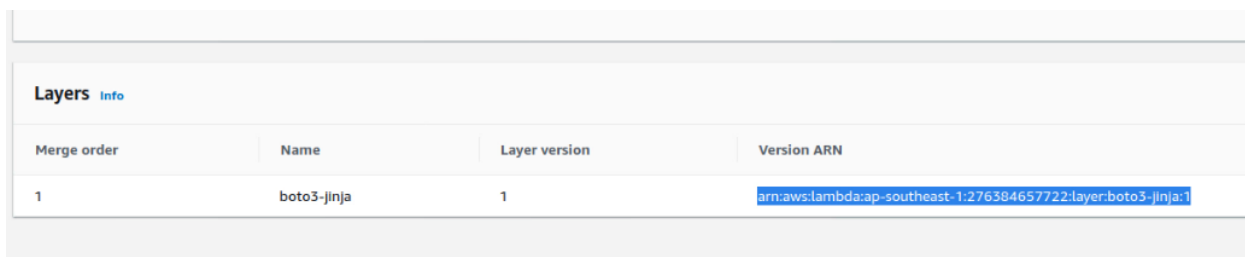
lambda_function x

Environment

dom-xss /
templates
lambda_function.py

```
1 import os
2 import sys
3 import json
4 import boto3
5 import traceback
6 from jinja2 import Environment, FileSystemLoader
7 def lambda_handler(event, context):
8
9     env = Environment(loader=FileSystemLoader(os.path.join(os.path.dirname(__file__), "templates")))
10
11     client = boto3.client('dynamodb')
12     users=client.scan(Table='Users', Select = 'ALL_ATTRIBUTES', ScanFilter = {"username":
13 response_users=[]
14 for user in users.get('Items', []):
15     response_user={}
16     response_user["username"]=user["username"]
17     response_user["first_name"]=user["first_name"]
18     response_user["last_name"]=user["last_name"]
19     response_user["age"]=str(user["age"])
20     response_users.append(response_user)
21
22
23 template = env.get_template("index.html")
24
25 return {
26     "statusCode": 200,
```

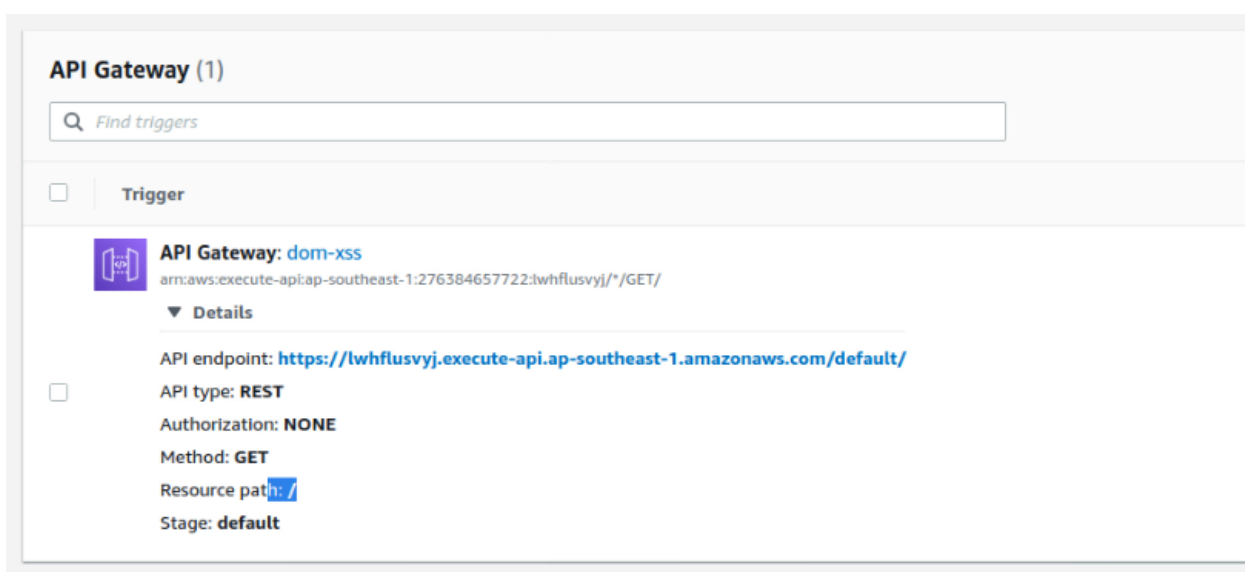
Check source code.



The screenshot shows the 'Layers' section of the AWS Lambda console. It features a table with columns for Merge order, Name, Layer version, and Version ARN. A single layer named 'boto3-jinja' is listed with merge order 1 and version 1. The Version ARN is 'arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1'.

Merge order	Name	Layer version	Version ARN
1	boto3-jinja	1	arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1

Check layers.




The screenshot shows the 'API Gateway (1)' section of the AWS console. It includes a search bar labeled 'Find triggers'. Below it, a checkbox is next to the word 'Trigger'. A list item for 'API Gateway: dom-xss' is shown with its ARN. A 'Details' section is expanded, displaying the API endpoint, type (REST), authorization (NONE), method (GET), resource path (/), and stage (default).

API Gateway (1)

Find triggers

☐ Trigger

 **API Gateway: dom-xss**
arn:aws:execute-api:ap-southeast-1:276384657722:whflusvyj/* /GET/

▼ Details

☐ API endpoint: <https://whflusvyj.execute-api.ap-southeast-1.amazonaws.com/default/>

API type: **REST**

Authorization: **NONE**

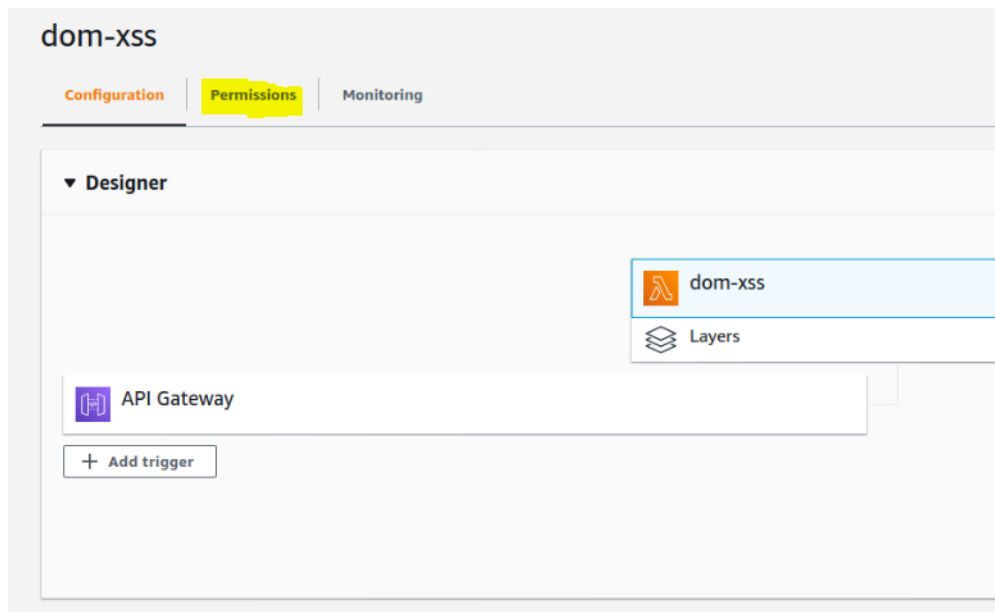
Method: **GET**

Resource path: **/**

Stage: **default**

Check API gateways.

Step 11: Click on the permissions tab to enumerate permissions for function.



Step 12: Check role and resource permission policy document.

Execution role

Role name
[dom-xss-role-navgos1x](#)

Resource summary

AWS Key Management Service
2 actions, 1 resource

To view the resources and actions that your function has permission to access, choose a service.

By action | **By resource**

Resource	Actions
All resources	Allow: kms:DescribeKey Allow: kms:ListAliases

Resource-based policy [Info](#)

```
1 {  
2   "Version": "2012-10-17",  
3   "Id": "default",  
4   "Statement": [  
5     {  
6       "Sid": "0d8e5e0d-b3b2-4652-8f3e-833c4dc6a97a",  
7       "Effect": "Allow",  
8       "Principal": {  
9         "Service": "apigateway.amazonaws.com"  
10      },  
11      "Action": "lambda:InvokeFunction",  
12      "Resource": "arn:aws:lambda:ap-southeast-1:276384657722:function:dom-xss",  
13      "Condition": {  
14        "ArnLike": {  
15          "AWS:SourceArn": "arn:aws:execute-api:ap-southeast-1:276384657722:lwhtflusvyj/*/GET/"  
16        }  
17      }  
18    }  
19  ]  
20 }
```

CLI Based Enumeration

Step 1: Click on the lab link to enumerate AWS credentials.

Access Credentials to your AWS lab Account

Login URL	https://276384657722.signin.aws.amazon.com/console
Region	Asia Pacific (Singapore) ap-southeast-1
Username	YqbrNVqnTEQEccFVnovf
Password	vtJ9O0r0Mjgxdqkj
Access Key ID	AKIAUAWOPGE5JBAUFKDZ
Secret Access Key	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

Step 2: Configure AWS CLI with AWS access keys.

Command: aws configure

```

X $ root@Kali ~$ aws configure
AWS Access Key ID [*****M5OT]: AKIAUAWOPGE5JBAUFDKZ
AWS Secret Access Key [*****Plnz]: 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
Default region name [ap-southeast-1]:
Default output format [None]:
$ root@Kali ~$

```

Step 3: List lambda layers.

Command: aws lambda list-layers

```

{
  "LayerName": "boto3-jinja",
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja",
  "LatestMatchingVersion": {
    "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1",
    "Version": 1,
    "CreateDate": "2020-12-17T13:20:25.446+0000",
    "CompatibleRuntimes": [
      "python3.8"
    ]
  }
},
{
  "LayerName": "cryptography",
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:cryptography",
  "LatestMatchingVersion": {
    "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:cryptography:1",
    "Version": 1,
    "CreateDate": "2020-12-19T17:19:55.514+0000",
    "CompatibleRuntimes": [
      "python3.8"
    ]
  }
},
{
  "LayerName": "php-runtime",
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime",
  "LatestMatchingVersion": {
    "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:4",
    "Version": 4,
    "CreateDate": "2021-02-26T20:35:10.271+0000",
    "CompatibleRuntimes": [
      "php"
    ]
  }
}

```

Step 4: Check lambda layers versions and compatible runtimes.

Commands:

```

aws lambda list-layer-versions --layer-name boto3-jinja
aws lambda list-layer-versions --layer-name php-runtime

```

```

root@Kali ~# aws lambda list-layer-versions --layer-name boto3-jinja
{
  "LayerVersions": [
    {
      "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:boto3-jinja:1",
      "Version": 1,
      "CreateDate": "2020-12-17T13:20:25.446+0000",
      "CompatibleRuntimes": [
        "python3.8"
      ]
    }
  ]
}
root@Kali ~#

```

```

root@Kali ~# aws lambda list-layer-versions --layer-name php-runtime
{
  "LayerVersions": [
    {
      "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:4",
      "Version": 4,
      "CreateDate": "2021-02-26T20:35:10.271+0000",
      "CompatibleRuntimes": []
    },
    {
      "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:3",
      "Version": 3,
      "CreateDate": "2021-02-26T19:53:51.485+0000",
      "CompatibleRuntimes": [
        "provided"
      ]
    },
    {
      "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:2",
      "Version": 2,
      "CreateDate": "2021-02-26T19:49:32.299+0000",
      "CompatibleRuntimes": [
        "provided"
      ]
    }
  ]
}

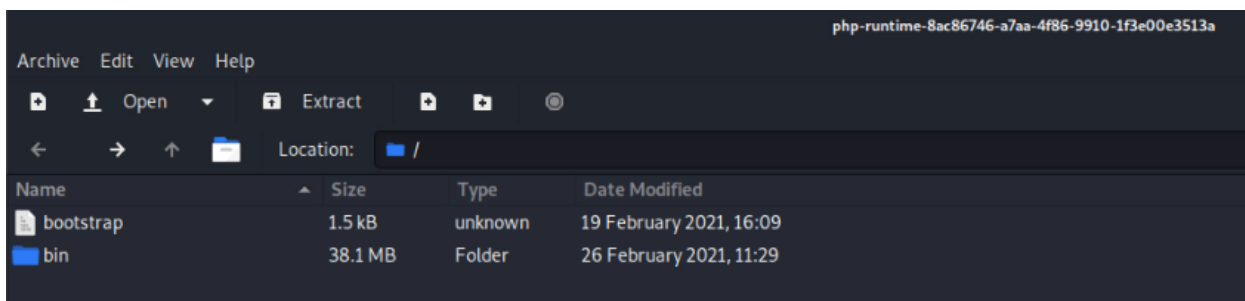
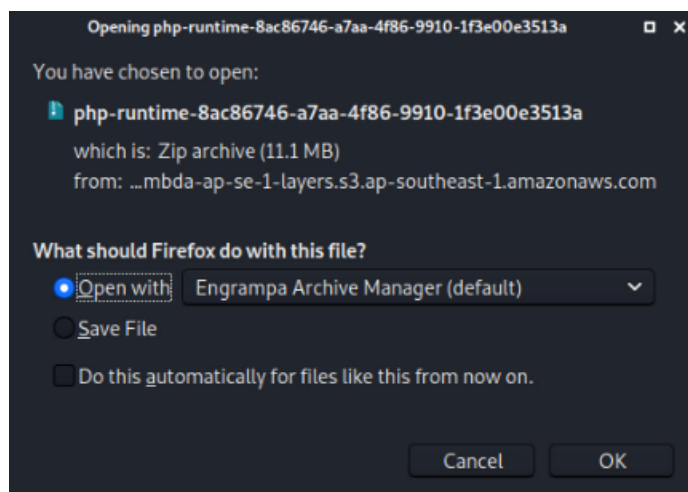
```

Step 5: Get layer version information.

Command: `aws lambda get-layer-version --layer-name php-runtime --version-number 2`


```
root@kali ~# aws lambda get-layer-version --layer-name php-runtime --version-number 2
{
  "Content": {
    "Location": "https://awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/php-run
z-Security-Token=IQoJb3JpZ2luX2VjEEQaDmFwLXNvdXRoZWZdC0xIkgwRgIhA0I1byjAu00MzktCR7diQvi3M8LDqmwQyXTs%2FyOr%2FZ0AiE
x3ZyqbAwxtrC1oioj08LZnIcJOXSHGy63iSfyB21SsQUoa8Yy6bKBSIR0U39I9dGGvLcoXWkIKCnC6qglvUGUnLQVku64Z%2BNcEuSRUeDd5k8wTedQ
YGoTATY5cM26bgSgIMaAWGkur3NvhxEV45pCFimJLhKjEAm9peHcxLVhzuq9Tg6esgQRqJ0LJdNWwKctr2Yjxao8mF4IrhXsWqzA6HLMX5r3Ip7SVD
ryoWVwKiX1keNhai11sNs0tlbphw49TSIdDgd3AH%2BLMcSXNeArTiAwwUcorNTwKA7XmQadPNemSjavK87XkHYdqxt9%2F0892ZC7iNFNoFC0KXIkX
i6z%2FV0ocnzGuDEcjHeFvmtXU%2Fp%2Fd%2FJ33u1LRfKfRUBnnADLfnjGG7DDc0i9AhsWVmZKU%2FgFtcejXm%2Fci5Q9MC6LHuKALgEr2X2AMGmM1
RZCHrDntDuwuQvsZv05q3vccfV7V9CVvUiN%2B2FKAC3f0V8JCPxuytvSYt8sZV3k0Sw5hi2jXSNAZPFy%2F5dVYyHBnhDTAmg%3D%3D6X-Amz-Algor
6006X-Amz-Credential=ASIAUJQ407LPT4EVXA2G%2F20210227%2Fap-southeast-1%2Fs3%2Faws4_request6X-Amz-Signature=64c67ce61f
"CodeSha256": "H81tStX6Y85e0NdEOxAysqHzxb+LEo1qRf2tYHjAcFI=",
    "CodeSize": 11587139
  },
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime",
  "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:2",
  "Description": "",
  "CreateDate": "2021-02-26T19:49:32.299+0000",
  "Version": 2,
  "CompatibleRuntimes": [
    "provided"
  ]
}
```

Step 6: Open layer location link to download layer package.



Step 7: List lambda functions.

Command: aws lambda list-functions

```
{
  "FunctionName": "dynamodb-nosqli-demo",
  "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:dynamodb-nosqli-demo",
  "Runtime": "python3.8",
  "Role": "arn:aws:iam::276384657722:role/service-role/dynamodb-nosqli-demo-role-yjj31w7u",
  "Handler": "lambda_function.lambda_handler",
  "CodeSize": 344359,
  "Description": "",
  "Timeout": 3,
  "MemorySize": 128,
  "LastModified": "2020-12-12T21:09:38.906+0000",
  "CodeSha256": "FEpF9Zo01dm5g6btq2MrJK08CIWnTrU5NhEj38LHzK8=",
  "Version": "$LATEST",
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "RevisionId": "5256010f-2aab-424d-b98e-de4a62bc00ea",
  "PackageType": "Zip"
},
{
  "FunctionName": "ipcalc-interface",
  "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:ipcalc-interface",
  "Runtime": "python3.8",
```

Step 8: Get details for function.

Command: aws lambda get-function --function-name xxe-handler

```
root@Kali ~# aws lambda get-function --function-name xxe-handler
{
  "Configuration": {
    "FunctionName": "xxe-handler",
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:xxe-handler",
    "Runtime": "provided.al2",
    "Role": "arn:aws:iam::276384657722:role/service-role/xxe-handler-role-efqzpgms",
    "Handler": "hello.handler",
    "CodeSize": 3221,
    "Description": "",
    "Timeout": 900,
    "MemorySize": 128,
    "LastModified": "2021-02-27T03:41:31.588+0000",
    "CodeSha256": "utEpNOS7YHtYisTSlLB/AkhJaVTWcqRWz+BkLtDeCsE=",
    "Version": "$LATEST",
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "RevisionId": "fea13fab-41fb-42b3-96fb-edefe80a3ddf",
```

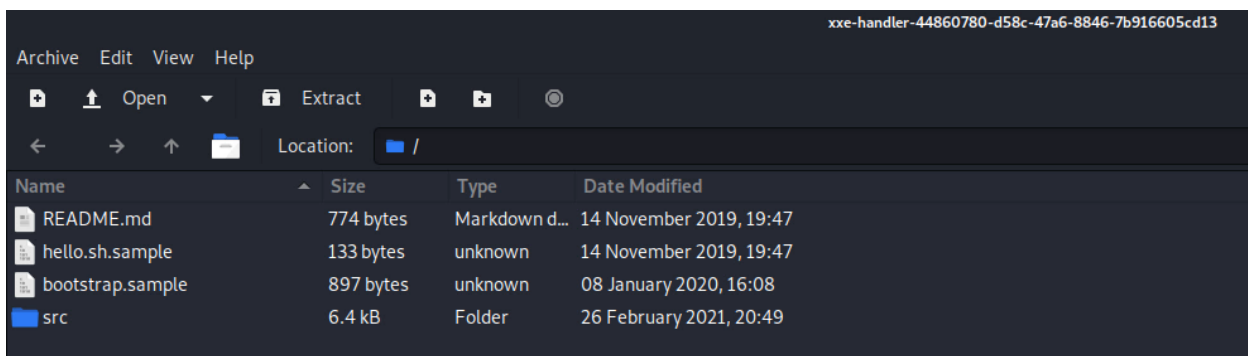
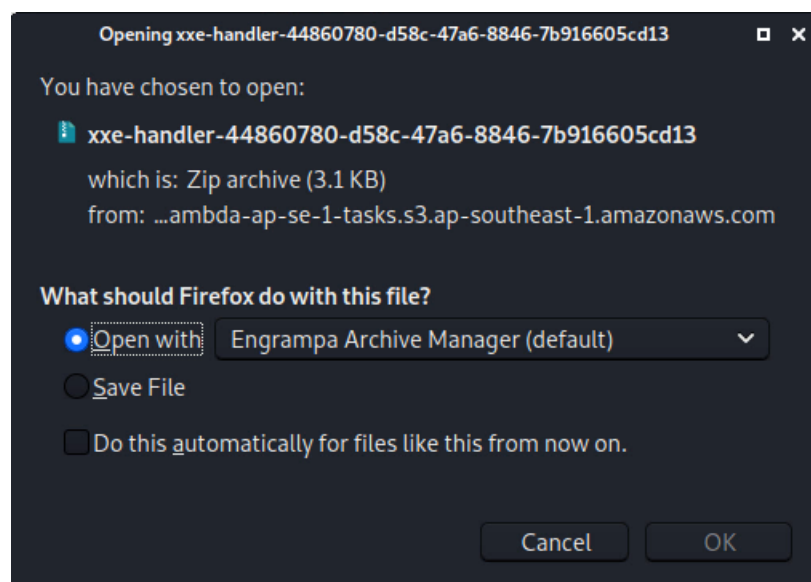


```

"Layers": [
  {
    "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-runtime:4",
    "CodeSize": 11587139
  },
  {
    "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:php-vendor:3",
    "CodeSize": 557421
  }
],
"State": "Active",
"LastUpdateStatus": "Successful",
"PackageType": "Zip"
},
"Code": {
  "RepositoryType": "S3",
  "Location": "https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722-FunctionCode-2020-11-14-19-47-5aJXoKpsDarvt6EkjeECZ4KjeIwNenBysRaDu31dSMgxZ%2BkW25G2YvFKuoRjSZvrbtT%2BfP851e6WFzmXYTZAecTBw37rnSx"
}

```

Step 9: Open function location link to download function source packages.



Step 10: Check function event source mappings.

Command: `aws lambda list-event-source-mappings --function-name dom-xss`

```
➤ root@Kali ~ ➤ aws lambda list-event-source-mappings --function-name dom-xss
{
  "EventSourceMappings": []
}
➤ root@Kali ~ ➤
```

References:

1. AWS CLI (<https://docs.aws.amazon.com/cli/latest/reference/>)