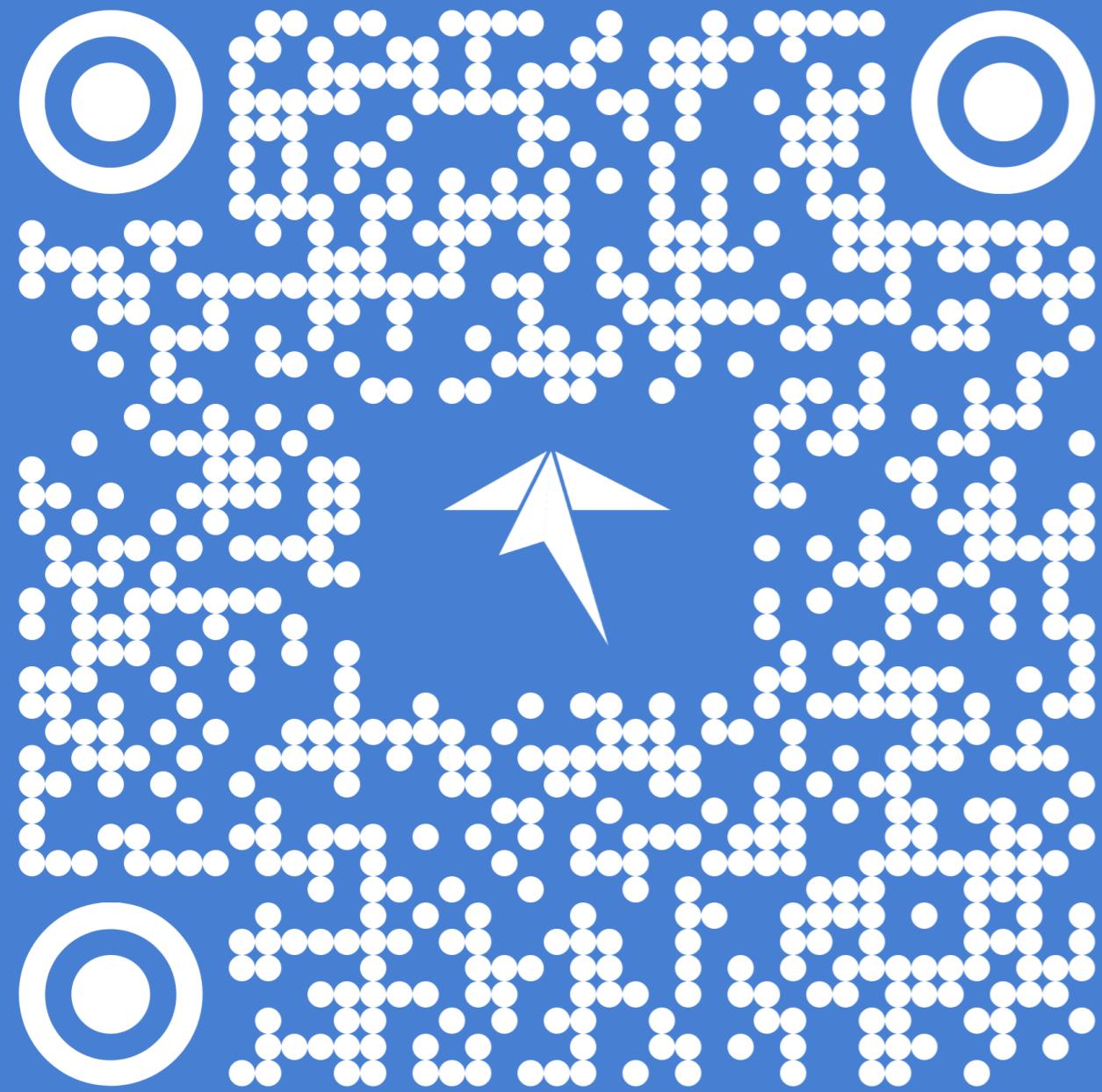


Exploring boto3 Events With Mitmproxy

Michael Twomey





fourTheorem

I'll be talking
about solving a
problem

AWS S3



boto3



HTTP



TLS



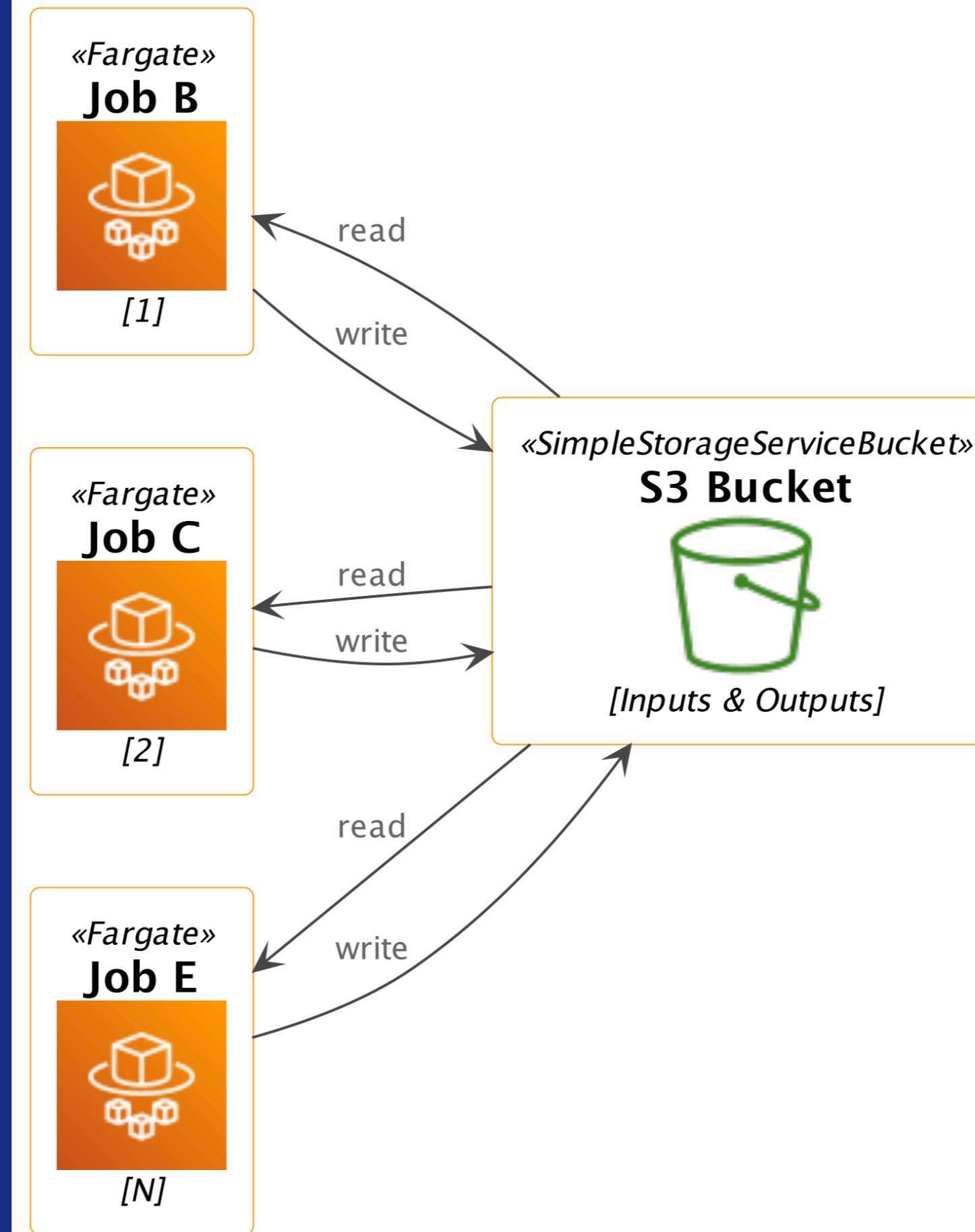
Python



mitmproxy

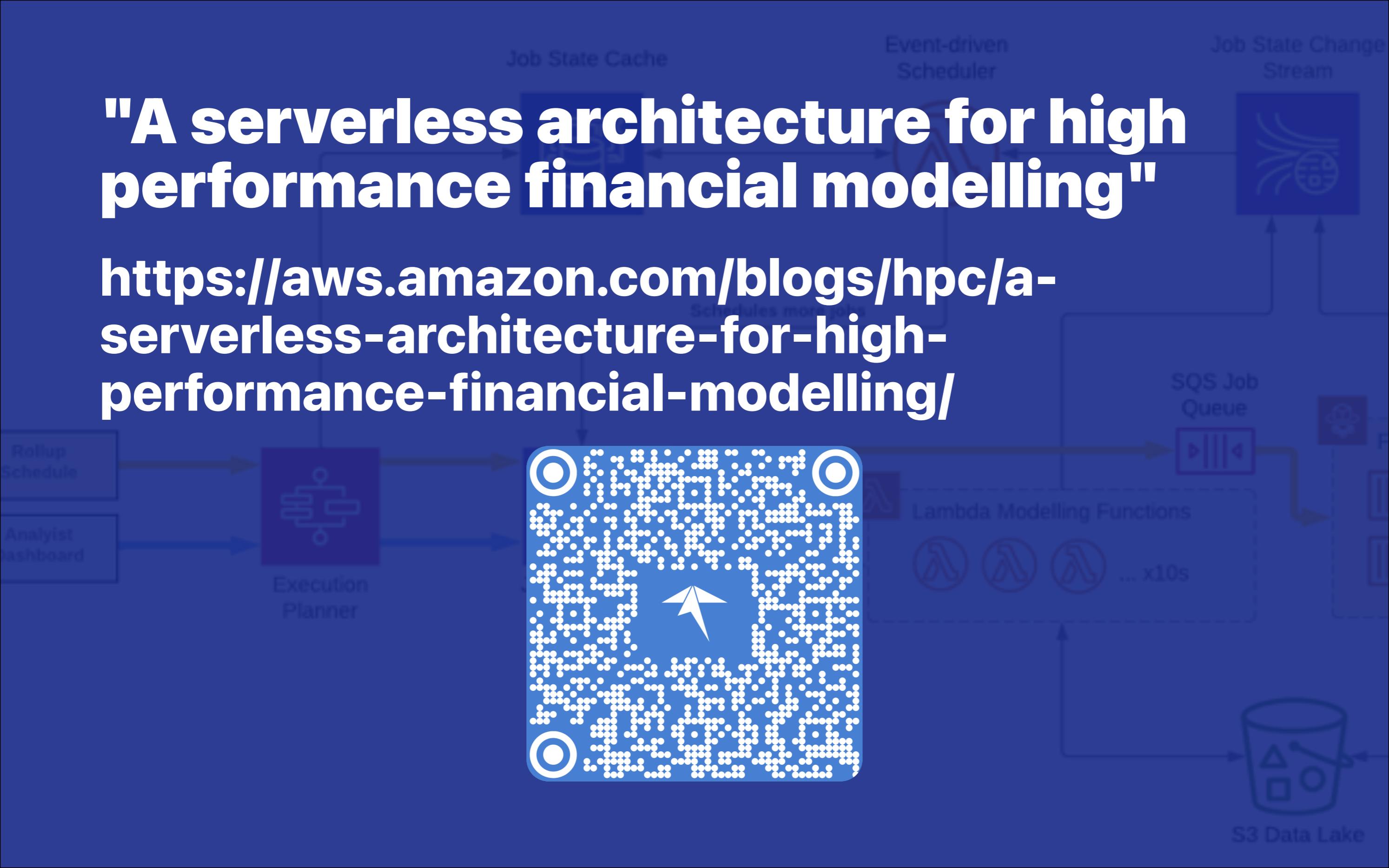
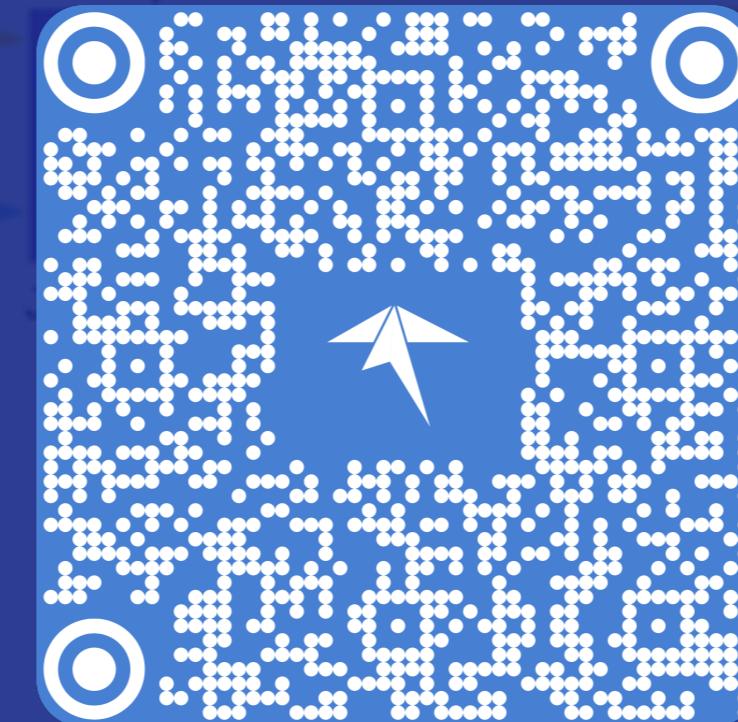


The Setup



"A serverless architecture for high performance financial modelling"

<https://aws.amazon.com/blogs/hpc/a-serverless-architecture-for-high-performance-financial-modelling/>



The Problem:

We got one of
the dreaded
questions

*"Why is my compute
job so slow?"*

A clue in the logs

```
Traceback (most recent call last):
```

```
  File "...examples/s3_get_object.py", line 12, in <module>
    response = s3.get_object(Bucket=bucket, Key=key)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "...botocore/client.py", line 535, in _api_call
    return self._make_api_call(operation_name, kwargs)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "...botocore/client.py", line 980, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.exceptions.ClientError: An error occurred (429) when
calling the GetObject operation (reached max retries: 4): Too
Many Requests
```

*"Are we triggering a lot of S3
request retries?"*

**Request
retries?**

**Rate
limits?**

**Isn't the
cloud as
infinite as
your wallet?**

S3 PUT
object

3,500
requests
per
second

boto3 retries requests automatically

```
import boto3

s3 = boto2.client("s3")
# Implicitly retries, sleeping each
time:
response = s3.get_object(...)
```

**Default: 5
attempts
(4 retries)**

**Up to 30
seconds
sleeping**

Theory

Even though we mostly eventually succeed, we take a really long time doing stuff because of all the retries and sleeps

**How Can We Be
Sure it's
Retrying?**

Logging?

```
logging.basicConfig(level=logging.DEBUG)
```

```
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-parameter-build.s3.GetObject: calling handler <function sse_md5 at 0x110d3f920>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-parameter-build.s3.GetObject: calling handler <function validate_bucket_name at 0x110d3f880>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-parameter-build.s3.GetObject: calling handler <function remove_bucket_from_url_paths_from_model at 0x110d5d9e0>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-parameter-build.s3.GetObject: calling handler <bound method S3RegionRedirectorv2.annotate_request_context of
<botocore.utils.S3RegionRedirectorv2 object at 0x114319fd0>>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-parameter-build.s3.GetObject: calling handler <function generate_idempotent_uuid at 0x110d3f6a0>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-endpoint-resolution.s3: calling handler <function customize_endpoint_resolver_builtins at 0x110d5dbc0>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-endpoint-resolution.s3: calling handler <bound method S3RegionRedirectorv2.redirect_from_cache of
<botocore.utils.S3RegionRedirectorv2 object at 0x114319fd0>>
2023-11-07 12:05:04 DEBUG botocore.regions Calling endpoint provider with parameters: {'Bucket': 'micktwomey-scratch-example', 'Region': 'eu-west-1', 'UseFIPS': False,
'UseDualStack': False, 'ForcePathStyle': False, 'Accelerate': False, 'UseGlobalEndpoint': False, 'DisableMultiRegionAccessPoints': False, 'UseArnRegion': True}
2023-11-07 12:05:04 DEBUG botocore.regions Endpoint provider result: https://micktwomey-scratch-example.s3.eu-west-1.amazonaws.com
2023-11-07 12:05:04 DEBUG botocore.regions Selecting from endpoint provider's list of auth schemes: "sigv4". User selected auth scheme is: "None"
2023-11-07 12:05:04 DEBUG botocore.regions Selected auth type "v4" as "v4" with signing context params: {'region': 'eu-west-1', 'signing_name': 's3', 'disableDoubleEncoding':
True}
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-call.s3.GetObject: calling handler <function add_expect_header at 0x110d3fc40>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-call.s3.GetObject: calling handler <function add_recursion_detection_header at 0x110d3df80>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-call.s3.GetObject: calling handler <function inject_api_version_header_if_needed at 0x110d5d1c0>
2023-11-07 12:05:04 DEBUG botocore.endpoint Making request for OperationModel(name=GetObject) with params: {'url_path': '/test/prefix1/ham/spam/foo.txt', 'query_string': {},
'method': 'GET', 'headers': {'User-Agent': 'Boto3/1.28.79 md/Botocore#1.31.79 ua/2.0 os/macos#23.0.0 md/arch#x86_64 lang/python#3.11.5 md/pyimpl#CPython cfg/retry-mode#legacy
Botocore/1.31.79'}, 'body': b'', 'auth_path': '/micktwomey-scratch-example/test/prefix1/ham/spam/foo.txt', 'url': 'https://micktwomey-scratch-example.s3.eu-west-
1.amazonaws.com/test/prefix1/ham/spam/foo.txt', 'context': {'client_region': 'eu-west-1', 'client_config': <botocore.config.Config object at 0x114319550>, 'has_streaming_input':
False, 'auth_type': 'v4', 's3_redirect': {'redirected': False, 'bucket': 'micktwomey-scratch-example', 'params': {'Bucket': 'micktwomey-scratch-example', 'Key':
'test/prefix1/ham/spam/foo.txt'}}, 'signing': {'region': 'eu-west-1', 'signing_name': 's3', 'disableDoubleEncoding': True}}}
2023-11-07 12:05:04 DEBUG botocore.hooks Event request-created.s3.GetObject: calling handler <bound method RequestSigner.handler of <botocore.signers.RequestSigner object at
0x114319510>>
2023-11-07 12:05:04 DEBUG botocore.hooks Event choose-signer.s3.GetObject: calling handler <bound method ClientCreator._default_s3_presign_to_sigv2 of
<botocore.client.ClientCreator object at 0x110e138d0>>
2023-11-07 12:05:04 DEBUG botocore.hooks Event choose-signer.s3.GetObject: calling handler <function set_operation_specific_signer at 0x110d3f560>
2023-11-07 12:05:04 DEBUG botocore.hooks Event before-sign.s3.GetObject: calling handler <function remove_arn_from_signing_path at 0x110d5db20>
2023-11-07 12:05:04 DEBUG botocore.auth Calculating signature using v4 auth.
2023-11-07 12:05:04 DEBUG botocore.auth CanonicalRequest:
GET
/test/prefix1/ham/spam/foo.txt

host:micktwomey-scratch-example.s3.eu-west-1.amazonaws.com
x-amz-content-sha256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
x-amz-date:20231107T120504Z
x-amz-security-token:...

host;x-amz-content-sha256;x-amz-date;x-amz-security-token
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
2023-11-07 12:05:04 DEBUG botocore.auth StringToSign:
AWS4-HMAC-SHA256
20231107T120504Z
20231107/eu-west-1/s3/aws4_request
4b4ea6ab346ff7949f47a10a3bf46a9169749520d75a30696b0fff2c077eb7a2
2023-11-07 12:05:04 DEBUG botocore.auth Signature:
30dfcec555fb14f449a78972ecd2063c76b5c93a70f5ef5a6a16e52aed405b79
2023-11-07 12:05:04 DEBUG botocore.hooks Event request-created.s3.GetObject: calling handler <function add_retry_headers at 0x110d5d940>
2023-11-07 12:05:04 DEBUG botocore.endpoint Sending http request: <AWSPreparedRequest stream_output=True, method=GET, url=https://micktwomey-scratch-example.s3.eu-west-
1.amazonaws.com/test/prefix1/ham/spam/foo.txt, headers={'User-Agent': b'Boto3/1.28.79 md/Botocore#1.31.79 ua/2.0 os/macos#23.0.0 md/arch#x86_64 lang/python#3.11.5
md/pyimpl#CPython cfg/retry-mode#legacy Botocore/1.31.79', 'X-Amz-Date': b'20231107T120504Z', 'X-Amz-Security-Token': b'...', 'X-Amz-Content-SHA256':
b'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855', 'Authorization': b'AWS4-HMAC-SHA256 Credential=ASIAVYOFBVCGIENLWHC/20231107/eu-west-1/s3/aws4_request,
SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token, Signature=30dfcec555fb14f449a78972ecd2063c76b5c93a70f5ef5a6a16e52aed405b79', 'amz-sdk-invocation-id':
```

Events?

```
s3 = boto3.client("s3")
s3.meta.events.register(
    "needs-retry.*", my_function
)
```

```
"provide-client-params.s3.ListObjects"
"provide-client-params.s3.*"
"provide-client-params.*"
"**"
```

```
import boto3

def print_event(event_name, **kwargs):
    print(event_name)

print("\nS3:")
s3 = boto3.client("s3")
s3.meta.events.register("*", print_event)
s3.list_buckets()

print("\nEC2:")
ec2 = boto3.client("ec2")
ec2.meta.events.register("*", print_event)
ec2.describe_instances()
```

S3:

```
provide-client-params.s3.ListBuckets
before-parameter-build.s3.ListBuckets
before-call.s3.ListBuckets
request-created.s3.ListBuckets
choose-signer.s3.ListBuckets
before-sign.s3.ListBuckets
before-send.s3.ListBuckets
response-received.s3.ListBuckets
needs-retry.s3.ListBuckets
after-call.s3.ListBuckets
```

EC2:

```
provide-client-params.ec2.DescribeInstances
before-parameter-build.ec2.DescribeInstances
before-call.ec2.DescribeInstances
request-created.ec2.DescribeInstances
choose-signer.ec2.DescribeInstances
before-sign.ec2.DescribeInstances
before-send.ec2.DescribeInstances
response-received.ec2.DescribeInstances
needs-retry.ec2.DescribeInstances
after-call.ec2.DescribeInstances
```

```
import boto3

def print_event(event_name, **kwargs):
    print(event_name, kwargs)

s3 = boto3.client("s3")
s3.meta.events.register("*", print_event)
```

```
request-created.s3.ListBuckets
{
    'request':
<botocore.awsrequest.AWSRequest>,
    'operation_name': 'ListBuckets'
}
```

```
# example of how your hook is called
my_hook(
    "request-created.s3.ListBuckets",
    request=...,
    operation_name=...
)
```



```
needs-retry.s3.ListBuckets
{
    'response': (
        <botocore.awsrequest.AWSResponse>,
        {
            'ResponseMetadata': {
                'RequestId': 'QZV9EWHJMR4T8VQ9',
                ...
            }
            'endpoint': s3(https://s3.eu-west-1.amazonaws.com),
            'operation': OperationModel(name=ListBuckets),
            'attempts': 1,
            'caught_exception': None,
            'request_dict': {
                'url_path': '/',
                'query_string': '',
                'method': 'GET',
                ...
            }
        }
    )
}
```

```
needs-retry.s3.ListBuckets  
...  
'attempts': 1,  
'caught_exception': None,  
...  
  
# example of how your hook is called  
my_hook(  
    "needs-retry.s3.ListBuckets",  
    attempts=1,  
    caught_exception=None,  
    ...  
)
```

**We have an event to
watch but we don't
know what failure
looks like**

**No Documentation
of needs-
retry.s3.ListBuckets
Event** 😭

**How do we
figure out what
the event looks
like for real?**

1.
**Triggering
the rate
limit for
real** 💰

**4. Hacking
the OS** 📱

**2. Hacking
the library**



**3. Hacking
Python** 🐍

**5. Hacking
the
network** 🌐



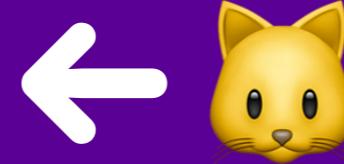
HTTP: The Bits We Care About

Client Request →



```
GET /cat.jpeg HTTP/1.1
Host: example.s3.eu-west-
2.amazonaws.com
...
```

Server Response



```
HTTP/1.1 200 OK 🤝👀
Content-Type: image/jpeg
```

...

Hacking HTTP

Bet: AWS uses
"429 Too Many
Requests" for
rate limits

Bet: boto3 looks
at the HTTP
response code

Bet: boto3
ignores
everything
except that 429

From Success 😭



**ERROR
FREE
REQUESTS**



**BROKEN
RATE LIMITS**

imgflip.com

HTTP/1.1 200 OK ➡
Content-Length: 34202
Content-Type: application/json
...
{
...
}

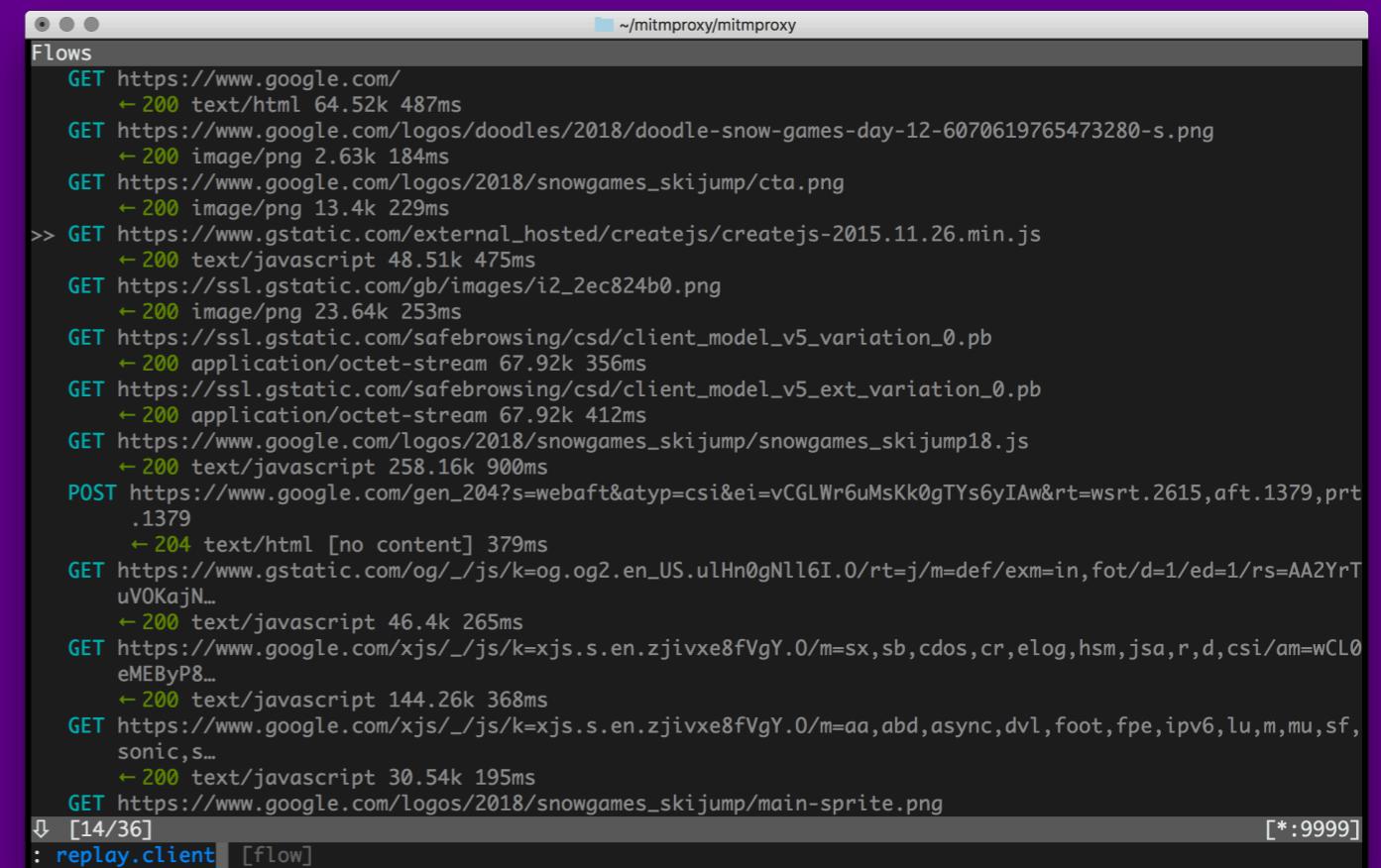
To Failure! 😬

HTTP/1.1 429 Rate limit exceeded ➡
Content-Length: 34202
Content-Type: application/json;
...
{
...
}

How: mitmproxy

<https://mitmproxy.org>

mitmproxy is a free
and open source
interactive HTTPS
proxy.



The screenshot shows the mitmproxy application window titled 'Flows'. It displays a list of network requests and responses. The requests include GETs for Google's homepage, various doodles, and other resources like 'createjs-2015.11.26.min.js' and 'safebrowsing files'. There are also POST requests for Google's search engine. The interface includes a status bar at the bottom with '[14/36]', ': replay.client', and '[flow]'. The title bar indicates the path is '~/mitmproxy/mitmproxy'.

```
Flows
GET https://www.google.com/
    ← 200 text/html 64.52k 487ms
GET https://www.google.com/logos/doodles/2018/doodle-snow-games-day-12-6070619765473280-s.png
    ← 200 image/png 2.63k 184ms
GET https://www.google.com/logos/2018/snowgames_skijump/cta.png
    ← 200 image/png 13.4k 229ms
>> GET https://www.gstatic.com/external_hosted/createjs/createjs-2015.11.26.min.js
    ← 200 text/javascript 48.51k 475ms
GET https://ssl.gstatic.com/gb/images/i2_2ec824b0.png
    ← 200 image/png 23.64k 253ms
GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_variation_0.pb
    ← 200 application/octet-stream 67.92k 356ms
GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_ext_variation_0.pb
    ← 200 application/octet-stream 67.92k 412ms
GET https://www.google.com/logos/2018/snowgames_skijump/snowgames_skijump18.js
    ← 200 text/javascript 258.16k 900ms
POST https://www.google.com/gen_204?s=webaft&atyp=csi&ei=vCGLWr6uMsKk0gTYs6yIAw&rt=wsrt.2615,aft.1379,prt
    .1379
    ← 204 text/html [no content] 379ms
GET https://www.gstatic.com/og/_/js/k=og.og2.en_US.ulHn0gNl16I.0/rt=j/m=def/exm=in,fot/d=1/ed=1/rs=AA2YrT
    uVOKajN...
    ← 200 text/javascript 46.4k 265ms
GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8fVgY.0/m=sx,sb,cdos,cr,elog,hsm,jsa,r,d,csi/am=wCL0
    eMEByP8...
    ← 200 text/javascript 144.26k 368ms
GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8fVgY.0/m=aa,abd,async,dvl,foot,fpe,ipv6,lu,m,mu,sf,
    sonic,s...
    ← 200 text/javascript 30.54k 195ms
GET https://www.google.com/logos/2018/snowgames_skijump/main-sprite.png
    ↓ [14/36]
    : replay.client [flow]
[*:9999]
```

1. Run `mitmproxy` (or `mitmweb` for a fancier web interface)
2. Set the HTTP proxy settings to `mitmproxy`'s (defaults to `http://localhost:8080`)
3. Run your program
4. Watch in `mitmproxy`

HTTP GET Example

```
import httpx

httpx.get("http://httpbin.org/get")
```

```
export http_proxy=localhost:8080

python http_get.py
```

```
mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy ⌘2
Flows
>>17:16:49 HTTP GET httpbin.org /get 200 application/json 303b 229ms

[*:8080]
[1/1]
Flow: ⌂ Select D Duplicate r Replay e Export d Delete m Mark b Save body
Proxy: ? Help q Quit E Events O Options i Intercept f Filter w Save flows z Clear list
```

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-07 17:16:49 GET http://httpbin.org/get
← 200 OK application/json 303b 229ms

Request	Response	Detail
Host: httpbin.org		
Accept: */*		
Accept-Encoding: gzip, deflate, br		
Connection: keep-alive		
User-Agent: python-httplib/0.25.1		
No request content		[m:auto]

[1/1] [*:8080]

Flow: e Edit D Duplicate r Replay e Export d Delete b Save body ↲ Next flow p Prev flow
Proxy: ? Help q Back E Events o Options i Intercept f Filter w Save flows z Clear list



mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⊜⌘2

Flow Details

2023-11-07 17:16:49 GET http://httpbin.org/get
← 200 OK application/json 303b 229ms

Request

Response

Detail

Date: Tue, 07 Nov 2023 17:16:49 GMT
Content-Type: application/json
Content-Length: 303
Connection: keep-alive
Server: gunicorn/19.9.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
ls:

JSON

[m:auto]

```
{  
    "args": {},  
    "headers": {  
        "Accept": "*/*",  
        "Accept-Encoding": "gzip, deflate, br",  
        "Host": "httpbin.org",  
        "User-Agent": "python-httplib/0.25.1",  
        "X-Amzn-Trace-Id": "Root=1-654a7101-66e698be1113a5f9629b4805"  
    },  
    "origin": "46.7.6.55",  
    "url": "http://httpbin.org/get"  
}
```

↓ [1/1]

[*:8080]

Flow:	e Edit	D Duplicate	r Replay	e Export	d Delete	b Save body	↶ Next flow	p Prev flow
Proxy:	? Help	q Back	E Events	o Options	i Intercept	f Filter	w Save flows	z Clear list



mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⊜⌘2

Flow Details

2023-11-07 17:16:49 GET http://httpbin.org/get
← 200 OK application/json 303b 229ms

Request

Response

Detail

Server Connection:

Address httpbin.org:80
Resolved Address 3.219.235.85:80
HTTP Version HTTP/1.1

Client Connection:

Address [::1]:58065
HTTP Version HTTP/1.1

Timing:

Client conn. established	2023-11-07 17:16:49.165
First request byte	2023-11-07 17:16:49.166
Request complete	2023-11-07 17:16:49.168
Server conn. initiated	2023-11-07 17:16:49.169
Server conn. TCP handshake	2023-11-07 17:16:49.280
First response byte	2023-11-07 17:16:49.393
Response complete	2023-11-07 17:16:49.395
Client conn. closed	2023-11-07 17:16:49.397
Server conn. closed	2023-11-07 17:16:49.398

[1/1]

[*:8080]

Flow:	e Edit	D Duplicate	r Replay	e Export	d Delete	b Save body	↶ Next flow	p Prev flow
Proxy:	? Help	q Back	E Events	o Options	i Intercept	f Filter	w Save flows	z Clear list

S3 Example

```
import boto3

s3 = boto3.client("s3")
s3.get_object(
    Bucket="micktwomey-scratch-example",
    Key="test/prefix1/ham/spam/foo.txt",
)
```

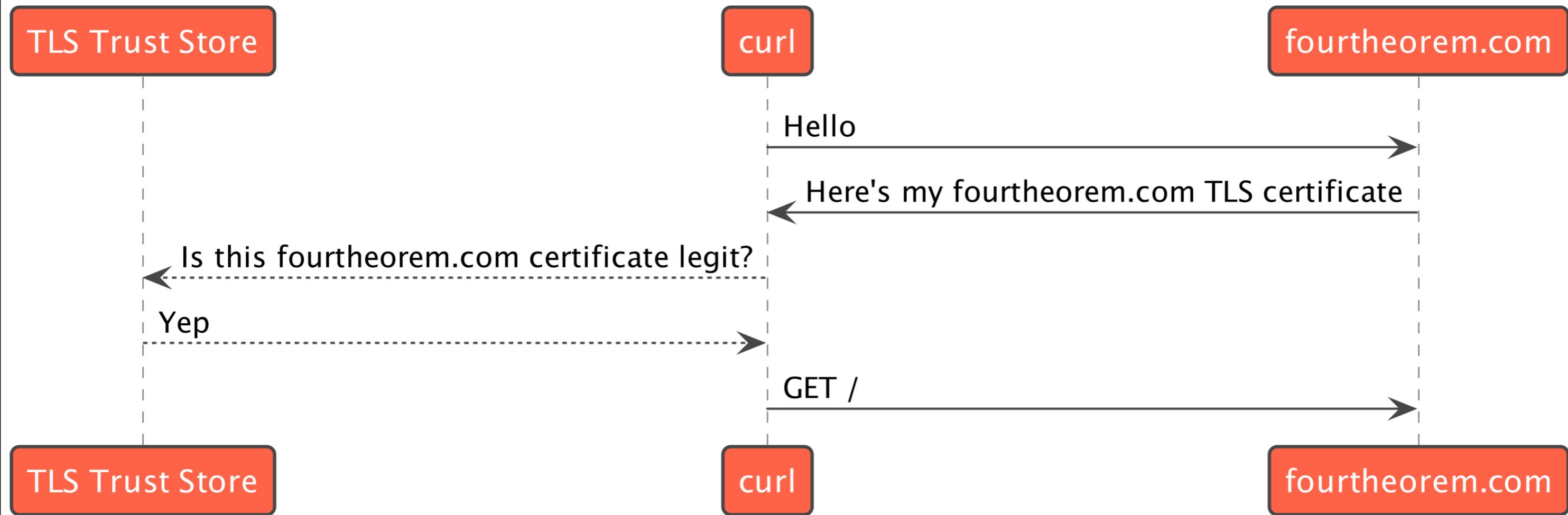
```
export http_proxy=localhost:8080
export https_proxy=localhost:8080
```

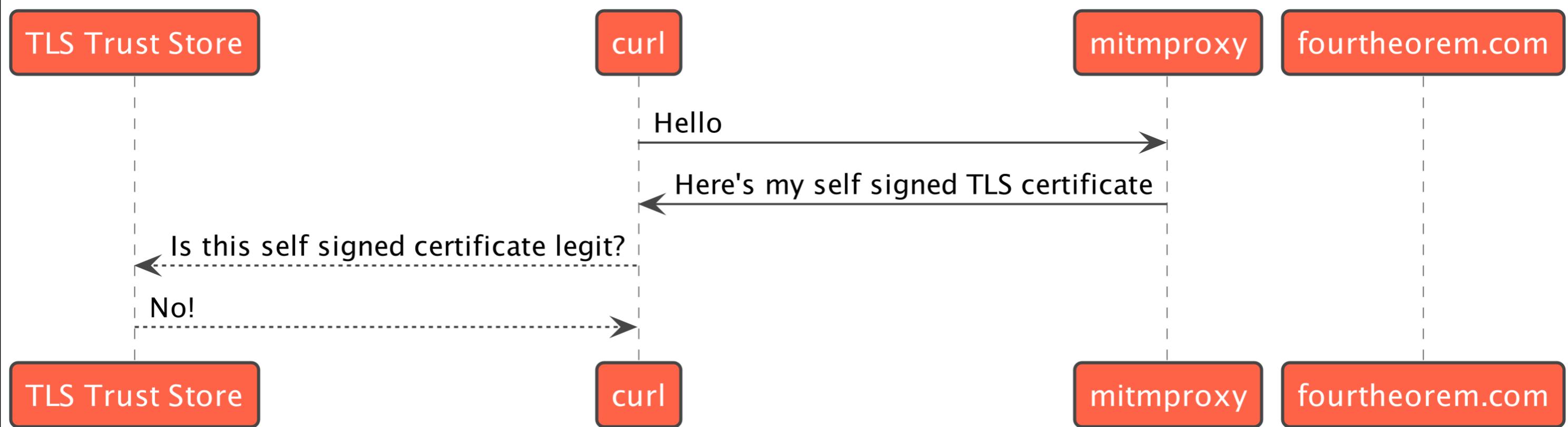
```
python s3_get_object.py
```

Wut?

```
...
File "/Users/mick/src/github.com/micktowomey/exploring-boto3-
events-with-mitmproxy/.venv/lib/python3.11/site-
packages/botocore/endpoint.py", line 377, in _send
    return self.http_session.send(request)
                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/Users/mick/src/github.com/micktowomey/exploring-boto3-
events-with-mitmproxy/.venv/lib/python3.11/site-
packages/botocore/httpsession.py", line 491, in send
    raise SSLError(endpoint_url=request.url, error=e)
botocore.exceptions.SSLError: SSL validation failed for
https://micktowomey-scratch-example.s3.eu-west-
1.amazonaws.com/test/prefix1/ham/spam/foo.txt [SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to
get local issuer certificate (_ssl.c:1006)
```

**TLS is doing its
job**





(ab)using self signed certs for the win!

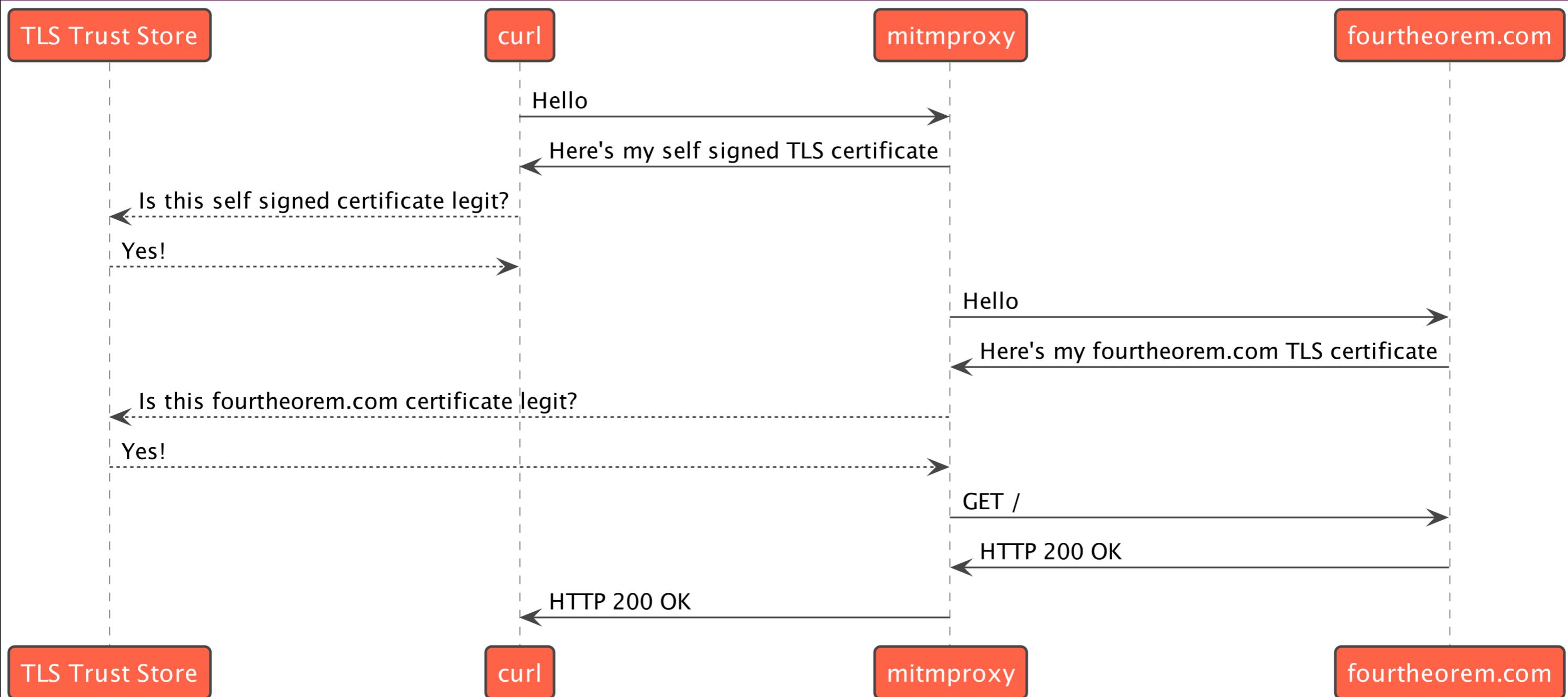
```
$ ls -l ~/.mitmproxy/
total 48
-rw-r--r-- 1 mick staff 1172 Sep  4 19:26 mitmproxy-ca-cert.cer
-rw-r--r-- 1 mick staff 1035 Sep  4 19:26 mitmproxy-ca-cert.p12
-rw-r--r-- 1 mick staff 1172 Sep  4 19:26 mitmproxy-ca-cert.pem
-rw----- 1 mick staff 2411 Sep  4 19:26 mitmproxy-ca.p12
-rw----- 1 mick staff 2847 Sep  4 19:26 mitmproxy-ca.pem
-rw-r--r-- 1 mick staff  770 Sep  4 19:26 mitmproxy-dhparam.pem
```

⚠ Danger! ⚠ Here be Dragons!



Full guide:

<https://docs.mitproxy.org/stable/concepts-certificates/>



AWS_CA_BUNDLE

```
https_proxy=localhost:8080 \
AWS_CA_BUNDLE=$HOME/.mitmproxy/mitmproxy-ca-
cert.pem \
python examples/s3_get_object.py
```



mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy



Flows

>>16:40:41 HTTPS GETamazonaws.com /test/prefix1/ham/spam/foo.txt 200 text/plain 6b 110ms

[1/1]

[:8080]

Flow:	Select	D Duplicate	r Replay	e Export	d Delete	m Mark	b Save body	
Proxy:	? Help	q Quit	E Events	O Options	i Intercept	f Filter	w Save flows	z Clear list

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy		⌘⌘2					
Flow Details							
2023-11-07 16:40:41 GET https://micktwomey-scratch-example.s3.eu-west-1.amazonaws.com/test/prefix1/ham/spam/foo.txt ← 200 OK text/plain 6b 110ms							
Request	Response	Detail					
Host: micktwomey-scratch-example.s3.eu-west-1.amazonaws.com							
Accept-Encoding: identity							
User-Agent: Boto3/1.28.79 md/Botocore#1.31.79 ua/2.0 os/macos#23.0.0 md/arch#x86_64 lang/python#3.11.5 md/pyimpl#CPython cfg/retry-mode#legacy Botocore/1.31.79							
X-Amz-Date: 20231107T164041Z							
X-Amz-Security-Token: IQoJb3JpZ2luX2VjEHEaCwV1LXd1c3QtMSJHMEUCIhgUJK0rWtpx6ZTh4i8wIPlLYQ+F1ZkLd1QThDlqv07qAiEAtfy+umHCf3 AS3NEjtTM3LCvR0Iu57UB1LsIaRYE9ds4qhgMIqv//////////ARAAAGgwz0TYw0DcxMDk3NzYiDE0axb/VgH2C6orMeyraAod/ RSue4N2xklc7eKhSgg8uUs09Q5A60sG58QIQ+90kfizE0YYdPHf3kkIbPphkMEQbXQ/QRml0md1Ic7MSlpVl59ufCfwUkZfp6B w3voojvPrB1uZjGpw7Ksd01vAb59JkvmNN+6NoPpYk0Xid2I9vx7NlR6idIbjmlZmNjazpzSTYHaYVJi5hPhirBpt86QsdZwt 80yI1yLaTH04FffESEbZxFTCKxxm4oCF0J9j0L6+sIQh5wDIKYA62Q+We0r6Drmrxt6K09NTldQ04ZK8E7IRVm6xrY2oZM+o2Z y9/PXvaPqZgM+hFX00yhHeR3RqrKHUTVp1eepapSDLiI4o5T0SJDQ94NT9XPJ7lNNxZZG8J/fJR0s3a3TT62vWfj8IyaxUwIBJ heNsrwqJuyDNTFg4zP67sSHcI+zYGm66AU5Sj+vcuwpzHUF9yALQ3zLfA4mVYvUleNcwh9GpqgY6pgE/fFkcuChENFFaa6JiA 7lIlocsM0nq/TVl1N/Szmzg9kHboII6Y8qrVrf66g0PSVmDl+VSDg2+mu0BEr4TGoKfxfhhb7Abor4k7RSvB4MXZdESKj977Pz 1s68CPj3tvppf19yzUhdqXv9eGU8baLrwRHKTwly94gpc5kDODvSPTihBR6cUuCJbDM0oX2Nvdup6L3SmCuUBgc01BkN28MjIW Yt8GCw							
X-Amz-Content-SHA256: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855							
Authorization: AWS4-HMAC-SHA256 Credential=ASIAVY0FBVCIMVZDM3VP/20231107/eu-west-1/s3/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token, Signature=7b391ecc3402462e0c64766c5f07a38b35258c150ba54b2fb2ab8d6cccd4549f							
amz-sdk-invocation-id: c1cb0eeef-010b-4fad-af70-077be6636cc9							
amz-sdk-request: attempt=1							
No request content		[m:auto]					
[1/1]		[*:8080]					
Flow: e Edit	D Duplicate	r Replay	e Export	d Delete	b Save body	Next flow	p Prev flow
Proxy: ? Help	q Back	E Events	o Options	i Intercept	f Filter	w Save flows	z Clear list



mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⌘2

Flow Details

2023-11-07 16:40:41 GET https://micktwomey-scratch-example.s3.eu-west-1.amazonaws.com/test/prefix1/ham/spam/foo.txt
← 200 OK text/plain 6b 110ms

Request	Response	Detail
x-amz-id-2:	ZtbuzYalwmED0MWZcP9I0un7uGin2a0Ukmeo0NeDy13UfxwC/WRFq8FMyI4KgVFsxnyHlDlh5Ec=	
x-amz-request-id:	G1B3PKNVJSJ5MKQF	
Date:	Tue, 07 Nov 2023 16:40:42 GMT	
Last-Modified:	Fri, 20 Oct 2023 17:06:43 GMT	
ETag:	"b1946ac92492d2347c6235b4d2611184"	
x-amz-server-side-encryption:	AES256	
Accept-Ranges:	bytes	
Content-Type:	text/plain	
Server:	AmazonS3	
Content-Length:	6	

Raw

[m:auto]

hello

[1/1]

[*:8080]

Flow: e Edit D Duplicate r Replay e Export d Delete b Save body ↲ Next flow p Prev flow
Proxy: ? Help q Back E Events o Options i Intercept f Filter w Save flows z Clear list



Flow Details

2023-11-07 16:40:41 GET https://micktwomey-scratch-example.s3.eu-west-1.amazonaws.com/test/prefix1/ham/spam/foo.txt
← 200 OK text/plain 6b 110ms

Request

Response

Detail

Server Connection:

Address micktwomey-scratch-example.s3.eu-west-1.amazonaws.com:443
Resolved Address 52.218.109.0:443
HTTP Version HTTP/1.1
ALPN http/1.1

Server Certificate:

Type RSA, 2048 bits
SHA256 digest 63 4c 10 56 3c 85 c7 ae c4 b4 72 43 69 9f 8d 2e dd 80 18 92 2a d7 06 cb 7a d2 17 8a 7b a3 c0 29
Valid from 2023-10-10 00:00:00+00:00
Valid to 2024-09-12 23:59:59+00:00
Serial 3497643272326306858947754322096787020
Subject CN *.s3-eu-west-1.amazonaws.com
Issuer C US
O Amazon
CN Amazon RSA 2048 M01
Alt names s3-eu-west-1.amazonaws.com, *.s3-eu-west-1.amazonaws.com, s3.eu-west-1.amazonaws.com,
*.s3.eu-west-1.amazonaws.com, s3.dualstack.eu-west-1.amazonaws.com,
*.s3.dualstack.eu-west-1.amazonaws.com, *.s3.amazonaws.com, *.s3-control.eu-west-1.amazonaws.com,
s3-control.eu-west-1.amazonaws.com, *.s3-control.dualstack.eu-west-1.amazonaws.com,
s3-control.dualstack.eu-west-1.amazonaws.com, *.s3-accesspoint.eu-west-1.amazonaws.com,
*.s3-accesspoint.dualstack.eu-west-1.amazonaws.com, *.s3-deprecated.eu-west-1.amazonaws.com,
s3-deprecated.eu-west-1.amazonaws.com, s3-external-3.amazonaws.com, *.s3-external-3.amazonaws.com

Client Connection:

Address [::1]:56923
HTTP Version HTTP/1.1

[1/1]

[*:8080]

Flow: e Edit

D Duplicate

r Replay

e Export

d Delete

b Save body

u Next flow

p Prev flow

Proxy: ? Help

q Back

E Events

o Options

i Intercept

f Filter

w Save flows

z Clear list



mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

`⌘2

Flow Details

2023-11-07 16:40:41 GET https://micktwomey-scratch-example.s3.eu-west-1.amazonaws.com/test/prefix1/ham/spam/foo.txt
← 200 OK text/plain 6b 110ms

Request	Response	Detail
*.s3.eu-west-1.amazonaws.com, s3.dualstack.eu-west-1.amazonaws.com, .s3.dualstack.eu-west-1.amazonaws.com, *.s3.amazonaws.com, *.s3-control.eu-west-1.amazonaws.com, s3-control.eu-west-1.amazonaws.com, *.s3-control.dualstack.eu-west-1.amazonaws.com, s3-control.dualstack.eu-west-1.amazonaws.com, *.s3-accesspoint.eu-west-1.amazonaws.com, *.s3-accesspoint.dualstack.eu-west-1.amazonaws.com, *.s3-deprecated.eu-west-1.amazonaws.com, s3-deprecated.eu-west-1.amazonaws.com, s3-external-3.amazonaws.com, *.s3-external-3.amazonaws.com		

Client Connection:

Address [::1]:56923
HTTP Version HTTP/1.1
TLS Version TLSv1.3
Server Name Indication micktwomey-scratch-example.s3.eu-west-1.amazonaws.com
Cipher Name TLS_AES_256_GCM_SHA384
ALPN http/1.1

Timing:

Client conn. established 2023-11-07 16:40:41.285
Server conn. initiated 2023-11-07 16:40:41.288
Server conn. TCP handshake 2023-11-07 16:40:41.354
Server conn. TLS handshake 2023-11-07 16:40:41.419
Client conn. TLS handshake 2023-11-07 16:40:41.429
First request byte 2023-11-07 16:40:41.430
Request complete 2023-11-07 16:40:41.432
First response byte 2023-11-07 16:40:41.538
Response complete 2023-11-07 16:40:41.540
Client conn. closed 2023-11-07 16:40:41.560
Server conn. closed 2023-11-07 16:40:41.560

↓ [1/1]

[*:8080]

Flow: e Edit D Duplicate r Replay e Export d Delete b Save body ↲ Next flow p Prev flow
Proxy: ? Help q Back E Events o Options i Intercept f Filter w Save flows z Clear list

More than a HTTP debugger

<https://docs.mitmproxy.org/stable/mitmproxytutorial-interceptrequests/>

<https://docs.mitmproxy.org/stable/mitmproxytutorial-modifyrequests/>

1. Hit `i` to create an intercept
2. `~d s3.eu-west-1.amazonaws.com & ~S`
`~d` match on domain, `~S` match on server response
3. Run the command
4. In the UI go into the response and hit `e`
5. Change the response code to `429`
6. Hit `a` to allow the request to continue
7. Watch what happens in the command

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

✉⌘1

Flows

↓ [0/0]

[*:8080]

Flow: L Load flows n Create new

Proxy: ? Help q Quit

E Events

O Options

i Intercept

f Filter

Layout

ctrl → Switch F Follow new

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flows

```
↓ [0/0] [*:8080]
: set intercept '~d s3.eu-west-1.amazonaws.com & ~s'[]
```

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⤵⌘1

Flows

↳ [0/0] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow: L Load flows n Create new

Proxy: ? Help q Quit E Events O Options i Intercept f Filter L Layout ctrl → Switch F Follow new

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⋮⌘1

Flows

>>21:25:46 HTTPS GETamazonaws.com /

200 application/xml 398b 658ms

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow: ⌂ Select D Duplicate r Replay e Export d Delete m Mark a Resume b Save body
Proxy: ? Help q Quit E Events O Options i Intercept f Filter w Save flows z Clear list - Layout

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
← 200 OK application/xml 398b 658ms

Request Response intercepted Detail

Host: s3.eu-west-1.amazonaws.com
Accept-Encoding: identity
User-Agent: aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls
X-Amz-Date: 20231109T212545Z
X-Amz-Security-Token: IQoJb3JpZ2luX2VjEKb//////////wEaC WV1LXd lc3QtMSJHMEUCIDSp svSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQf kx4y4/BihWkuXLbNkoXPHm6ylTso3dYqhgMI3//////////ARAAGgwz0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG+6W/ORG0dXH77kuelLZJIhol59mqFne04aNT29+ImilKhY3TawQ0mTH4FI6clfur48vQkb7yQBK2LMQYCUBSbRb8Hny14ziWugW7i3YkJKVckm9ZC1k4tw5z7n3WgT6C58c9fsNF320NtdV VHzyU2WMAdygJXAexdAZ17/fn0gs9o6x+tTTASfqxUmpwaBT LcyAclEx4bv7P7Pk h1DJCxV0JAPweyb+vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1kVgtgygESB0r+iw/yq2xmdufneDIZyAND7Yls3crqilBXdc44CcRGspyDa9Y0PsZputtLXRjFwzXMJ2YigXmc6fxIrW6uJlzch/gespoPLMXI Qzyl6ihSGaT4+WfH9BccFdjpsYYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/gw1Jy1qgY6pgFvt+kIagxlgbS60a9ii iN2yNmNgXiEJ0MWiZWFVSZehYYkiCUSBnw9PmuGOGozpsrku3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02MGhoTp3paspl2C+o+iUcr5IWFFepbZxUlPpmrCrZjFeuspnn8XqZGLXlL2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9sV23h/cqrDhwI
X-Amz-Content-SHA256: e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855
Authorization: AWS4-HMAC-SHA256 Credential=ASIAVY0FBVCIBZEC6NVH/20231109/eu-west-1/s3/aws4_request,
SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token,
Signature=a8eec836da4c874e6106816bab940e4d4923ac8d6744716d56a548efb3406e2b

No request content

[m:auto]

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow:	e Edit	D Duplicate	r Replay	e Export	d Delete	a Resume	b Save body	Next flow	p Prev flow
Proxy:	? Help	q Back	E Events	o Options	i Intercept	f Filter	w Save flows	z Clear list	- Layout

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
← 200 OK application/xml 398b 658ms

Request	Response intercepted	Detail
Host: s3.eu-west-1.amazonaws.com		
Accept-Encoding: identity		
User-Agent: aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls		
X-Amz-Date: 20231109T212545Z		
X-Amz-Security-Token: IQoJb3JpZ2luX2VjEKb//////////wEaC fkx4y4/BihWkuXLbNkoXPHm6ylTso3dYq +6W/ORG0dXH77kuelLZJIhol59mqFne04 ZC1k4tw5z7n3WgT6C58c9fsNF320NtdVV +vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1k 2YigXmc6fxIrW6uJlzch/gespoPLMXIQz gw1Jy1qgY6pgFvt+kIagxlgbS60a9iiii MGhoTp3paspl2C+o+iUcr5IWFFepbZxUl V23h/cqrDhwI	<p>Part</p> <ul style="list-style-type: none">1) cookies2) urlencoded form3) multipart form4) path5) method6) query7) reason8) request-headers9) response-headersa) request-bodyb) response-bodyc) status_coded) set-cookiese) url	svSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQ z0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG FI6clfur48vQkb7yQBK2LMQYCUBsBb8Hny14ziWugW7i3YkJKVckm9 n0gs9o6x+tTTASfqxUmpwaBTLCyAclEx4bv7P7Pk1DJCxVOJAPweyb neDIZyAND7Yls3crqilBXdc44CcRGFgspyDa9Y0PsZputtLXRjFwzXMJ YYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/ YYkiCUSBnw9PmuGOGozpsrku3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02 L2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9s
X-Amz-Content-SHA256: e3b0c44298fc1c149afbf4c8996fb9242	7852b855	
Authorization: AWS4-HMAC-SHA256 Credential=ASIAV SignedHeaders=host;x-amz-content- Signature=a8eec836da4c874e6106816	9/eu-west-1/s3/aws4_request, -security-token, 6d56a548efb3406e2b	[m:auto]
No request content		

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
 ← 200 OK application/xml 398b 658ms

	Request	Response intercepted	Detail
Host:	s3.eu-west-1.amazonaws.com		
Accept-Encoding:	identity		
User-Agent:	aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls		
X-Amz-Date:	20231109T212545Z		
X-Amz-Security-Token:	IQoJb3JpZ2luX2VjEKb//////////wEaC fkx4y4/BihWkuXLbNkoXPHm6ylTso3dYq +6W/ORG0dXH77kuelLZJIhol59mqFne04 ZC1k4tw5z7n3WgT6C58c9fsNF320NtdVV +vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1k 2YigXmc6fxIrW6uJlzch/gespoPLMXIQz gw1Jy1qgY6pgFvt+kIagxlgbS60a9iiii MGhoTp3paspl2C+o+iUcr5IWFFepbZxUl V23h/cqrDhwI	Part 1) cookies 2) urlencoded form 3) multipart form 4) path 5) method 6) query 7) reason 8) request-headers 9) response-headers a) request-body b) response-body c) status_code d) set-cookies e) url	svSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQ z0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG FI6clfur48vQkb7yQBK2LMQYCUBsBbRb8Hny14ziWugW7i3YkJKVckm9 n0gs9o6x+tTTASfqxUmpwaBTLCyAclEx4bv7P7Pk1DJCxVOJAPweyb neDIZyAND7Yls3crqilBXdc44CcRGFgspyDa9Y0PsZputtLXRjFwzXMJ YYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/ YYkiCUSBnw9PmuGOGozpsrku3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02 L2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9s 7852b855 9/eu-west-1/s3/aws4_request, -security-token, 6d56a548efb3406e2b
X-Amz-Content-SHA256:	e3b0c44298fc1c149afbf4c8996fb9242		
Authorization:	AWS4-HMAC-SHA256 Credential=ASIAV SignedHeaders=host;x-amz-content- Signature=a8eec836da4c874e6106816		
No request content			[m:auto]

↳ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
← 200 OK application/xml 398b 658ms

	Request	Response intercepted	Detail
Host:	s3.eu-west-1.amazonaws.com		
Accept-Encoding:	identity		
User-Agent:	aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls		
X-Amz-Date:	20231109T212545Z		
X-Amz-Security-Token:	IQoJb3JpZ2luX2VjEKb//////////wEaCWV1LXd1c3QtMSJHMEUCIDSpsvSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQfkx4y4/BihWkuXLbNkoXPHm6ylTso3dYqhgMI3//////////ARAAGgwz0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG+6W/ORG0dXH77kuelLZJIhol59mqFne04aNT29+ImilKhY3TawQ0mTH4FI6clfur48v0kb7yQBK2LMQYCUBSbRb8Hny14ziWugW7i3YkJKVckm9ZC1k4tw5z7n3WgT6C58c9fsNF320NtdVHzyU2WMAdygJXAexdAZ17/fn0gs9o6x+tTTASfqxUmpwaBTLcyAclEx4bv7P7Pk1DJCxV0JAPweyb+vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1kVgtgygESB0r+iw/yq2xmdufneDIZyAND7Yls3crqilBXdc44CcRGFgspyDa9Y0PsZputtLXRjFwzXMJ2YigXmc6fxIrW6uJlzch/gespoPLMXI0zyl6ihSGaT4+WfH9BccFdjpsYYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/gw1Jy1qgY6pgFvt+kIagxlgbS60a9iiN2yNmNgXiEJ0MWiZWFVSZehYYkiCUSBnw9PmuGOGozsrsru3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02MGhoTp3paspl2C+o+iUcr5IWFFepbZxUlPpmrCrZjFeuspnn8XqZGLXlL2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9sV23h/cqrDhwI		
X-Amz-Content-SHA256:	e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855		
Authorization:	AWS4-HMAC-SHA256 Credential=ASIAVY0FBVCIBZEC6NVH/20231109/eu-west-1/s3/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token, Signature=a8eec836da4c874e6106816bab940e4d4923ac8d6744716d56a548efb3406e2b		

No request content

[m:auto]

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]
: flow.set @focus status_code 429

[*:8080]

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
← 429 OK application/xml 398b 658ms

	Request	Response intercepted	Detail
Host:	s3.eu-west-1.amazonaws.com		
Accept-Encoding:	identity		
User-Agent:	aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls		
X-Amz-Date:	20231109T212545Z		
X-Amz-Security-Token:	IQoJb3JpZ2luX2VjEKh//////////wEaCWV1LXdIc3QtMSJHMEUCIDSpsvSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQfkx4y4/BihWkuXLbNkoXPHm6ylTso3dYqhgMI3//////////ARAAGgwz0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG+6W/ORG0dXH77kuelLZJIhol59mqFne04aNT29+ImilKhY3TawQQmTH4FI6clfur48vQkb7yQBK2LMQYCUBSbRb8Hny14ziWugW7i3YkJKVckm9ZC1k4tw5z7n3WgT6C58c9fsNF320NtdVHzyU2WMAdygJXAexdAZ17/fn0gs9o6x+tTTASfqxUmpwaBTLcyAclEx4bv7P7Pk1DJCxVOJAPweyb+vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1kVgtgygESB0r+iw/yq2xmdufneDIZyAND7Yls3crqilBXdc44CcRGspyDa9Y0PsZputtLXRjFwzXMJ2YigXmc6fxIrW6uJlzch/gespoPLMXIQzyl6ihSGaT4+WfH9BccFdjpsYYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/gw1Jy1qgY6pgFvt+kIagxlgbS60a9iiN2yNmNgXiEJ0MWiZWFVSZehYYkiCUSBnw9PmuGOGozsrsru3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02MGhoTp3paspl2C+o+iUcr5IWFFepbZxUlPpmrCrZjFeuspnn8XqZGLXlL2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9sV23h/cqrDhwI		
X-Amz-Content-SHA256:	e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855		
Authorization:	AWS4-HMAC-SHA256 Credential=ASIAVY0FBVCIBZEC6NVH/20231109/eu-west-1/s3/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token, Signature=a8eec836da4c874e6106816bab940e4d4923ac8d6744716d56a548efb3406e2b		

No request content

[m:auto]

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]
Alert: [21:26:30.367] Set status_code on 1 flows.

[*:8080]

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-09 21:25:46 GET https://s3.eu-west-1.amazonaws.com/
← 429 OK application/xml 398b 658ms

	Request	Response	Detail
Host:	s3.eu-west-1.amazonaws.com		
Accept-Encoding:	identity		
User-Agent:	aws-cli/2.9.12 Python/3.11.3 Darwin/22.6.0 source/x86_64 prompt/off command/s3.ls		
X-Amz-Date:	20231109T212545Z		
X-Amz-Security-Token:	IQoJb3JpZ2luX2VjEKb//////////wEaCWV1LXd1c3QtMSJHMEUCIDSpsvSA2REqh+Dd3w3IF36x4ZSLpyWImV0+FVP4mhAhAiEA6RvvBvvbRxQfkx4y4/BihWkuXLbNkoXPHm6ylTso3dYqhgMI3//////////ARAAGgwz0TYw0DcxMDk3NzYiDEx0PLx1X4LEAVDpyCraAuQgmaLo/Gze2vLynG+6W/ORG0dXH77kuelLZJIhol59mqFne04aNT29+ImilKhY3TawQ0mTH4FI6clfur48vQkb7yQBK2LMQYCUBSbRb8Hny14ziWugW7i3YkJKVckm9ZC1k4tw5z7n3WgT6C58c9fsNF320NtdVHzyU2WMAdygJXAexdAZ17/fn0gs9o6x+tTTASfqxUmpwaBTLCyAclEx4bv7P7Pk1DJCxVOJAPweyb+vdrEQWp2vARE/SW+mrfPTQZVLQQpwZ1kVgtgygESB0r+iw/yq2xmdufneDIZyAND7Yls3crqilBXdc44CcRGFgspyDa9Y0PsZputtLXRjFwzXMJ2YigXmc6fxIrW6uJlzch/gespoPLMXI0zyl6ihSGaT4+WfH9BccFdjpsYYoCCeNmYaZo5SmajqjLbUwvh6Vz0j0778Ilgn61UfiGnkbUgcTaf/gw1Jy1qgY6pgFvt+kIagxlgbS60a9iiN2yNmNgXiEJ0MWiZWFVSZehYYkiCUSBnw9PmuGOGozsrsru3i7Tdxgw1sjSaTHj5BX8/q7J/+WNM02MGhoTp3paspl2C+o+iUcr5IWFFepbZxUlPpmrCrZjFeuspnn8XqZGLXlL2gnFC7tdpYVDC/CoYdCTTuteYn15Cd54VhZo0DbndUyl0/kpIn9q9sV23h/cqrDhwI		
X-Amz-Content-SHA256:	e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855		
Authorization:	AWS4-HMAC-SHA256 Credential=ASIAVY0FBVCIBZEC6NVH/20231109/eu-west-1/s3/aws4_request, SignedHeaders=host;x-amz-content-sha256;x-amz-date;x-amz-security-token, Signature=a8eec836da4c874e6106816bab940e4d4923ac8d6744716d56a548efb3406e2b		

No request content

[m:auto]

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow:	e Edit	D Duplicate	r Replay	e Export	d Delete	V Restore	b Save body	Next flow	p Prev flow
Proxy:	? Help	q Back	E Events	O Options	i Intercept	f Filter	w Save flows	z Clear list	- Layout

mitmproxy ~s/g/m/exploring-boto3-events-with-mitmproxy

⋮⌘1

Flows

>>21:25:46 HTTPS GETamazonaws.com / 429 application/xml 398b 658ms

↓ [1/1] [i:~d s3.eu-west-1.amazonaws.com & ~s]

[*:8080]

Flow: ⌂ Select D Duplicate r Replay e Export d Delete m Mark V Restore b Save body
Proxy: ? Help q Quit E Events O Options i Intercept f Filter w Save flows z Clear list - Layout

```
> https_proxy=localhost:8080 AWS_CA_BUNDLE=$HOME/.mitmproxy/mitmproxy-ca-cert.pem aws s3 ls  
An error occurred () when calling the ListBuckets operation:  
x 254 ↵ 54s a scratch.AWSAdministratorAccess/eu-west-1 21:26:39  
↳
```

I'm Lazy

```
import logging
from mitmproxy import flowfilter
from mitmproxy import http

class RateLimitExceededFilter:
    filter: flowfilter.TFilter

    def __init__(self):
        self.filter = flowfilter.parse("~d s3.eu-west-1.amazonaws.com &
~s")

    def response(self, flow: http.HTTPFlow) -> None:
        if flowfilter.match(self.filter, flow):
            flow.response.status_code = 429
            flow.response.reason = "Too Many Requests"

addons = [RateLimitExceededFilter()]
```

```
mitmproxy -s examples/mitm_interception.py
```



mitmproxy -s example ~/s/g/m/exploring-boto3-events-with-mitmproxy



Flows

↓ [0/0] [scripts:1]

[*:8080]

Flow: L Load flows n Create new

Proxy: ? Help q Quit

E Events

O Options

i Intercept

f Filter

Layout

ctrl → Switch F Follow new

```
> https_proxy=localhost:8080 AWS_CA_BUNDLE=$HOME/.mitmproxy/mitmproxy-ca-cert.pem aws s3 ls  
An error occurred () when calling the ListBuckets operation:  
x 254 | a scratch.AWSAdministratorAccess/eu-west-1 | 20:50:00  
> []
```

mitmproxy -s example ~/s/g/m/exploring-boto3-events-with-mitmproxy

⌘⌘1

Flows

>>20:32:40 HTTPS GETamazonaws.com / 429 application/xml 398b 137ms

↳ [1/1] [scripts:1]

[*:8080]

Flow: ⌂ Select D Duplicate r Replay e Export d Delete m Mark b Save body
Proxy: ? Help q Quit E Events O Options i Intercept f Filter w Save flows z Clear list - Layout

mitmproxy -s example ~/s/g/m/exploring-boto3-events-with-mitmproxy

Flow Details

2023-11-09 20:32:40 GET https://s3.eu-west-1.amazonaws.com/
← 429 Too Many Requests application/xml 398b 137ms

	Request	Response	Detail
x-amz-id-2:	+wB3WIjx8bmius4XI+5wE7U9xIx0gaw1KL48FfsiQLY3S89IylFB3dp3Hi9iWPGKVnn0NKHDR90=		
x-amz-request-id:	A1P49R1GY171MVZ2		
Date:	Thu, 09 Nov 2023 20:32:42 GMT		
Content-Type:	application/xml		
Transfer-Encoding:	chunked		
Server:	AmazonS3		

XML
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
 <Owner>
 <ID>1305744cb586daf054976969e08336547e38cf32939d2bacd7f31a896db7b67e</ID>
 <DisplayName>aws-scratch-account</DisplayName>
 </Owner>
 <Buckets>
 <Bucket>
 <Name>micktwomey-scratch-example</Name>
 <CreationDate>2022-09-04T15:13:22.000Z</CreationDate>
 </Bucket>
 </Buckets>
</ListAllMyBucketsResult>

[1/1] [scripts:1]

[*:8080]

Flow: e Edit D Duplicate r Replay e Export d Delete b Save body u Next flow p Prev flow
Proxy: ? Help q Back E Events o Options i Intercept f Filter w Save flows z Clear list - Layout

mitmproxy -s example ~/s/g/m/exploring-boto3-events-with-mitmproxy

Flows

>>21:30:03	HTTPS	GETamazonaws.com /		429	application/xml	398b	325ms
21:30:04	HTTPS	GETamazonaws.com /		429	application/xml	398b	42ms
21:30:05	HTTPS	GETamazonaws.com /		429	application/xml	398b	53ms
21:30:09	HTTPS	GETamazonaws.com /		429	application/xml	398b	61ms
21:30:12	HTTPS	GETamazonaws.com /		429	application/xml	398b	37ms
21:50:53	HTTPS	GETamazonaws.com /		429	application/xml	398b	135ms
21:50:54	HTTPS	GETamazonaws.com /		429	application/xml	398b	193ms
21:50:56	HTTPS	GETamazonaws.com /		429	application/xml	398b	71ms
21:50:59	HTTPS	GETamazonaws.com /		429	application/xml	398b	121ms
21:51:05	HTTPS	GETamazonaws.com /		429	application/xml	398b	358ms
21:51:23	HTTPS	GETamazonaws.com /		429	application/xml	398b	146ms
21:51:24	HTTPS	GETamazonaws.com /		429	application/xml	398b	197ms
21:51:25	HTTPS	GETamazonaws.com /		429	application/xml	398b	35ms
21:51:26	HTTPS	GETamazonaws.com /		429	application/xml	398b	74ms
21:51:32	HTTPS	GETamazonaws.com /		429	application/xml	398b	197ms
21:51:53	HTTPS	GETamazonaws.com /		429	application/xml	398b	121ms
21:51:54	HTTPS	GETamazonaws.com /		429	application/xml	398b	55ms
21:51:56	HTTPS	GETamazonaws.com /		429	application/xml	398b	124ms
21:51:57	HTTPS	GETamazonaws.com /		429	application/xml	398b	83ms
21:52:03	HTTPS	GETamazonaws.com /		err	...sed connection		
21:52:29	HTTPS	GETamazonaws.com /		429	application/xml	398b	248ms
21:52:30	HTTPS	GETamazonaws.com /		429	application/xml	398b	42ms
21:52:31	HTTPS	GETamazonaws.com /		429	application/xml	398b	37ms
21:52:34	HTTPS	GETamazonaws.com /		429	application/xml	398b	210ms
21:52:36	HTTPS	GETamazonaws.com /		429	application/xml	398b	37ms

[1/25] [scripts:1]

[*:8080]

Flow:	✉ Select	D Duplicate	r Replay	e Export	d Delete	m Mark	b Save body		
Proxy:	? Help	q Quit	E Events	O Options	i Intercept	f Filter	w Save flows	z Clear list	- Layout

```
import boto3

def print_event(event_name: str, attempts: int, response,
**kwargs):
    print(
        event_name,
        attempts,
        response[1] ["ResponseMetadata"] ["HTTPStatusCode"],
    )

s3 = boto3.client("s3")
s3.meta.events.register("needs-retry.s3.ListBuckets",
print_event)
s3.list_buckets()
```

mitmproxy -s example ~/s/g/m/exploring-boto3-events-with-mitmproxy

Flows

>>21:30:03	HTTPS	GETamazonaws.com /		429	application/xml	398b	325ms
21:30:04	HTTPS	GETamazonaws.com /		429	application/xml	398b	42ms
21:30:05	HTTPS	GETamazonaws.com /		429	application/xml	398b	53ms
21:30:09	HTTPS	GETamazonaws.com /		429	application/xml	398b	61ms
21:30:12	HTTPS	GETamazonaws.com /		429	application/xml	398b	37ms

[1/5] [scripts:1]

[*:8080]

Flow:	Select	D Duplicate	r Replay	e Export	d Delete	m Mark	b Save body		
Proxy:	? Help	q Quit	E Events	O Options	i Intercept	f Filter	w Save flows	z Clear list	- Layout

```
~/.s/g/m/exploring-boto3-events-with-mitmproxy
```

```
⌘%2
```

```
> AWS_CA_BUNDLE=$HOME/.mitmproxy/mitmproxy-ca.pem https_proxy=localhost:8080 python examples/intercepting.py  
needs-retry.s3.ListBuckets 1 429  
needs-retry.s3.ListBuckets 2 429  
needs-retry.s3.ListBuckets 3 429  
needs-retry.s3.ListBuckets 4 429  
needs-retry.s3.ListBuckets 5 429  
An error occurred () when calling the ListBuckets operation (reached max retries: 4):
```

```
... ⟲ 8s ✨ exploring-boto3-events-with-mitmproxy ↵ scratch.AWSAdministratorAccess/eu-west-1 * impure 23:10:17  
> [ ]
```

Metrics!

```
import boto3
from rich import print

def increment_metric(name):
    print(f"{name}|increment|count=1")

def handle_retry(event_name: str, attempts: int, **_):
    print("retry event?")
    if attempts > 1:
        increment_metric(event_name)
    else:
        print("nope!")

s3 = boto3.client("s3")
s3.meta.events.register("needs-retry.s3.*", handle_retry)
s3.list_buckets()
print("All done!")
```

```
~/.s/g/m/exploring-boto3-events-with-mitmproxy
```

```
⌘⌘2
```

```
> AWS_CA_BUNDLE=$HOME/.mitmproxy/mitmproxy-ca.pem https_proxy=localhost:8080 python examples/retry_metrics.py
retry event?
nope!
retry event?
needs-retry.s3.ListBuckets|increment|count=1
retry event?
needs-retry.s3.ListBuckets|increment|count=1
retry event?
needs-retry.s3.ListBuckets|increment|count=1
retry event?
needs-retry.s3.ListBuckets|increment|count=1
All done!
```

```
... ⟲ 5s  ♡ exploring-boto3-events-with-mitmproxy  ↵ scratch.AWSAdministratorAccess/eu-west-1  * impure  23:16:26
> []
```

```
1 fields @timestamp, event_name
2 | filter ispresent(event_name)
3 | filter event_name == 'needs-retry.s3.PutObject'
4 | filter attempts > 1
5 | sort by @timestamp asc
6 | stats count() by bin(1m)
7 # | limit 200
8
```

Run query

Save

Actions ▼

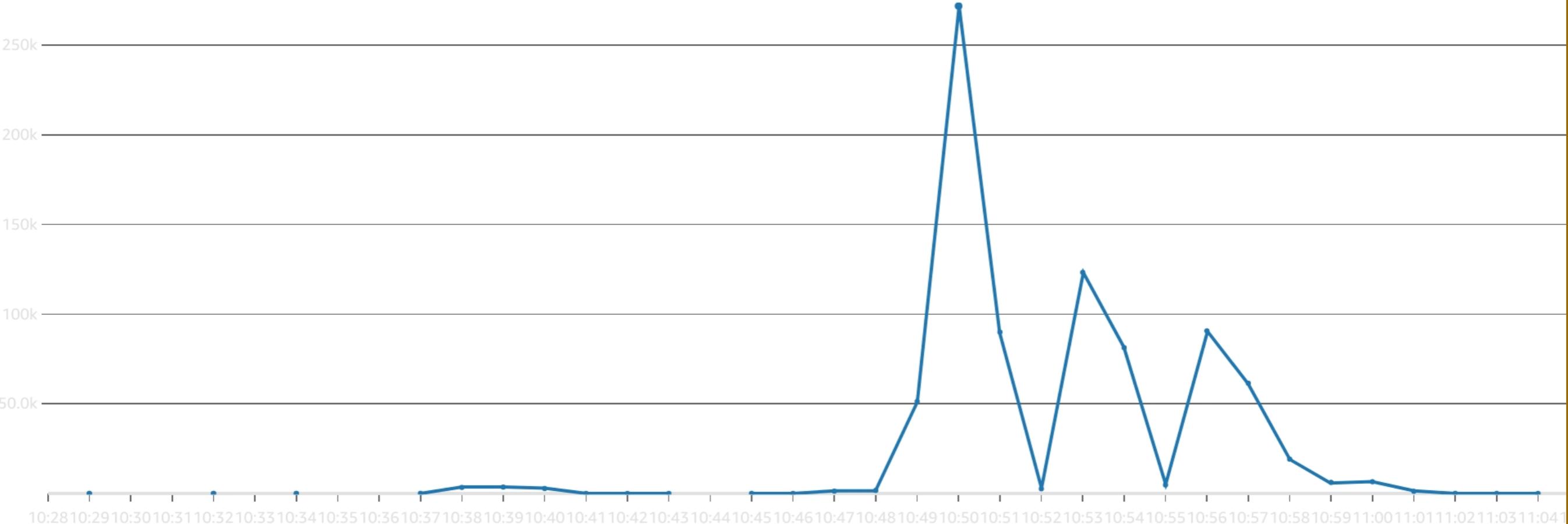
Histor

Queries are allowed to run for up to 15 minutes.

Logs

Visualization

Line ▼



Finally!

1. Got some jobs taking a long time
2. Guessed it's retries causing this
3. Want to use events to monitor these but don't have docs
4. Want to somehow modify responses to simulate retries and see what happens
5. Can intercept requests using mitmproxy
6. Now know the shape of the data we get in 429 responses
7. Can emit metrics and graph them

What We Covered

How
request
retries
behave

AWS API
limits

boto3's
event
system

mitmproxy That
nothing is
ever
simple!

Thank You! 



mastodon.ie/@micktwomey

 fourtheorem.com

[**github.com/micktwomey/
exploring-boto3-events-
with-mitmproxy**](https://github.com/micktwomey/exploring-boto3-events-with-mitmproxy)

