Continuous Assurance

and my experience wrangling the cloud

Francesco Badraun

Cloud Security Consultant

ZX Security

Photo by John Adams on Unsplash

FranCHCon

More like

FranCHContinuous Assurance

Content

- **Overview**
- **Why**
- **Elements**
- My experience
  - **Implementation**
  - **In practice**
- **Challenges**
- **Summary**

My background

My background

Career

⏳ 5 years in Security

🆕 Consultant

🖍️ Security Specialist

🛠️ Security Engineer

My experience with Continuous Assurance

1 year

Enough to know it’s difficult...

My background

☕ Coffee geek 👨‍💻 Programming

(🐍 Python 🌐 Web dev)

🎹 Music production 🎮 Gaming

Overview of Continuous Assurance

The issue: misconfigured cloud infrastructure

Misconfiguration is the #1 cause of cloud data breaches

"Misconfigured clouds were a leading cause of breaches." [1] [2]

(19% of attack vectors)

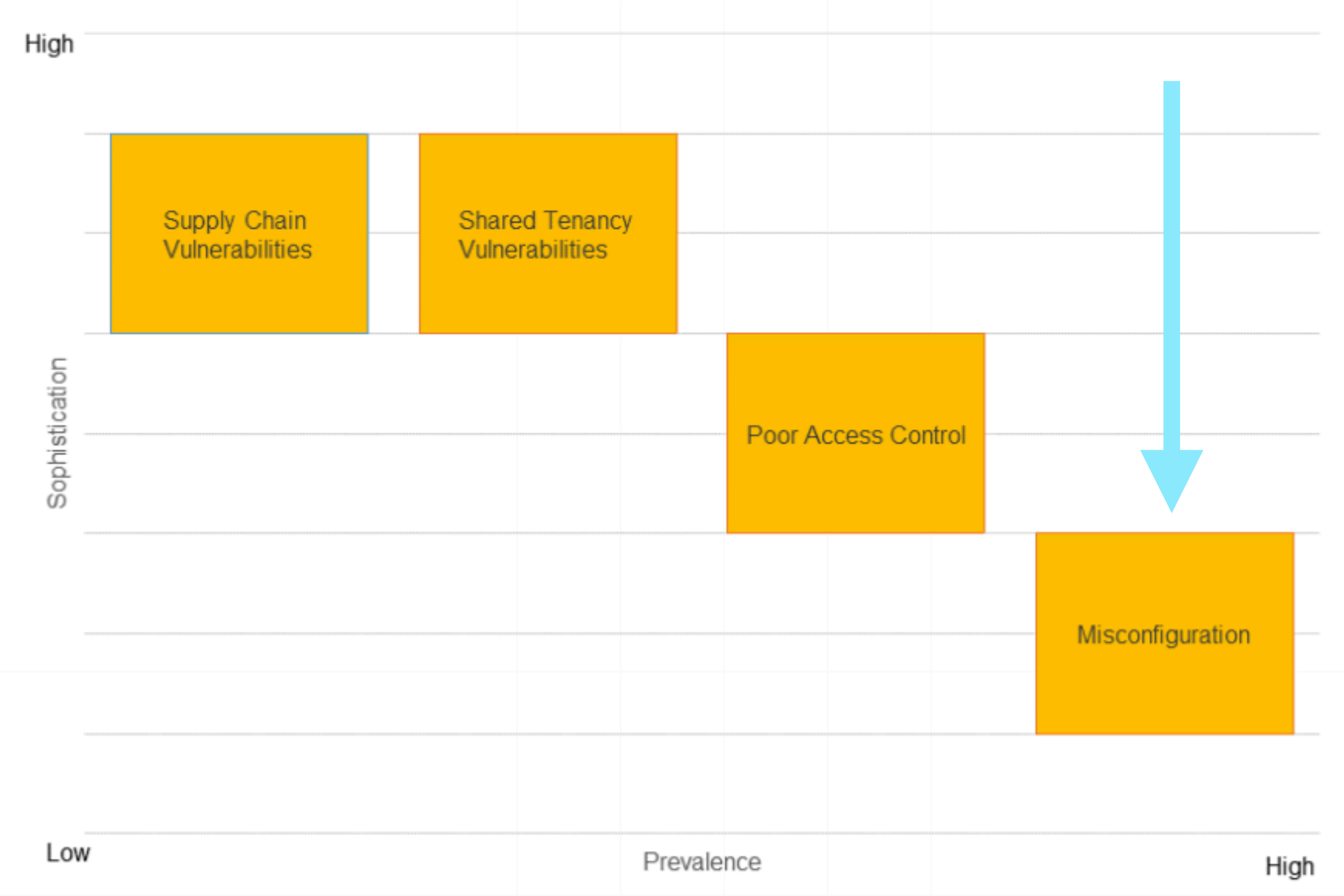
—IBM

- [1] IBM, Cost of a Data Breach Report (2020)
  - www.capita.com/sites/g/files/nginej291/files/2020-08/Ponemon-Global-Cost-of-Data-Breach-Study-2020.pdf
- [2] National Security Agency, Mitigating Cloud Vulnerabilities (2020)
  - media.defense.gov/2020/Jan/22/2002237484/-1/-1/0/CSI-MITIGATING-CLOUD-VULNERABILITIES_20200121.PDF

Not hard

Really common attack vector

Low attacker sophistication



Cloud Vulnerabilities – Prevalence versus Sophistication of Exploitation [2]

- [2] National Security Agency, Mitigating Cloud Vulnerabilities (2020)
  - media.defense.gov/2020/Jan/22/2002237484/-1/-1/0/CSI-MITIGATING-CLOUD-VULNERABILITIES_20200121.PDF

Purpose of Continuous Assurance

- **Automated remediation** of insecure infrastructure
- **Assurance** of security controls and compliance

Done through

- **Monitoring** resources and **enforcing** secure configurations

Data protection — example controls

S3 buckets

- Encryption
  - Enable **default encryption**
  - Bucket policy for secure transport (**TLS**)

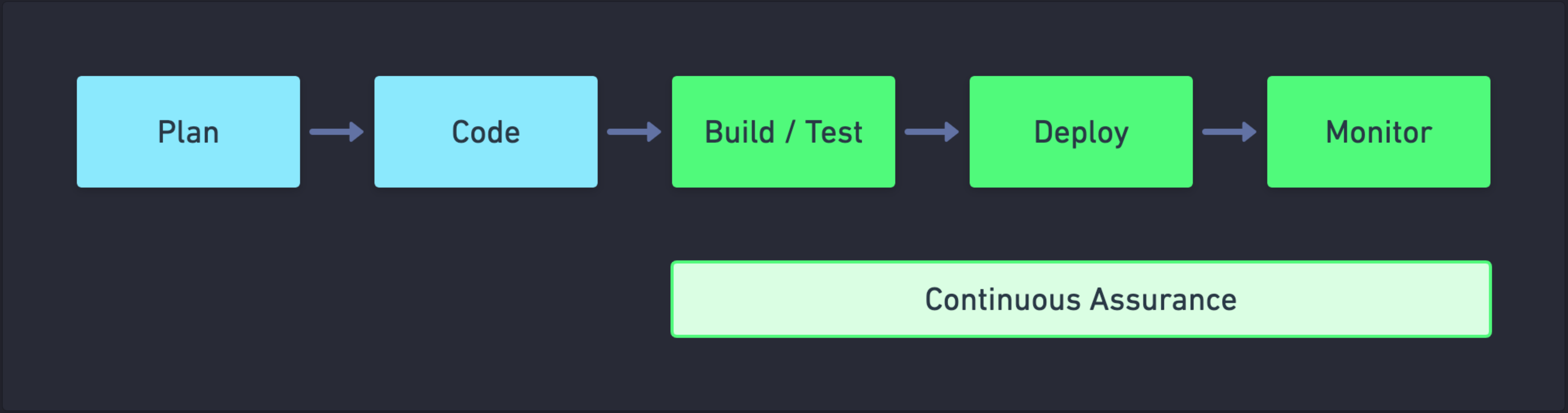
- Access control
  - Block **public access**
  - Bucket policy

- Backups
  - Enable **object versioning**
  - Cross-region **replication**

DevSecOps

Shift left

- Secure from the **start**
- Secure through **ongoing lifecycle**



Other names

- Automated Security Compliance
- Secure Infrastructure Configuration Enforcement
- **Cloud Wrangling**

Why Continuous Assurance?

Why

Benefits

Well-managed cloud

- Customer’s security **in** the cloud
- Understanding **inventory**

Scale

- Impractical to **manually review** everything
- Support **innovation at speed**

Auditing

Implementing controls

- ISO 27001, SOC 2, PCI DSS, GDPR, NZISM

Manual vs automated

- Snapshot in time vs **real time**
- Small sample vs **total compliance**
- Prove controls exist
    - Show documents
    - **Compliance as code**

Why not lint configuration templates?

Different tools

- AWS **CloudFormation**, Terraform, Chef
- **GitHub**, AWS CodePipeline, Jenkins, AWS Console 😱
- Difficult to enforce

Drift

- Great for prevention, **useless for drift**
- Resources can **change after deployment**
- Templates only show snapshot, **no view of lifecycle**

Elements of Continuous Assurance

Elements

Monitoring / Remediation

Components of Continuous Assurance

Photo by Christian Wiediger on Unsplash

Monitoring

Resource inventory

- Real time record of **resources** and **compliance**

Track changes

- **Chronic** monitoring of **changes**

Detect non-compliance

- Find **incorrect configurations**
- **Unit tests**
- **Multiple data sources**

Remediation

On creation

- **Delete**

On modification

- Update to **desired configuration**

Notify owners

- Detected **issues** and **remediation actions**

Remediation bonus: Prevention

Service settings

- Enable **secure default settings**
- EC2: Encryption **by default**

Linting

- Helps **fix** issues **sooner**

My experience

Implementation

Context

- Disclaimer: we didn’t implement a full solution
- Cloud environment
  - **AWS**
  - Hundreds of accounts
  - Hundreds of **thousands of resources**
  - **Multiple services** used

My experience: Implementation

Chosen tools

Photo by Luca Bravo on Unsplash

Cloud Custodian (c7n)

Cloud Custodian℠

Monitoring

- Resource **inventory**
- Detect **non-compliance**

Remediation

- Compliance **as code**
- Configuration **enforcement**

Why not AWS Config?

Cloud Custodian

Cloud Custodian benefits

- **Granular** filters and actions
- Larger **service support**
- **Multi-cloud**
- Tooling out of the box
- Wrapper for AWS Config

Downsides

Numeronym

- **c7n**, k8s, a11y, i18n, a16z
- en.wikipedia.org/wiki/Numeronym

CloudTrail

- **Missing** actions

How it works

Cloud Custodian

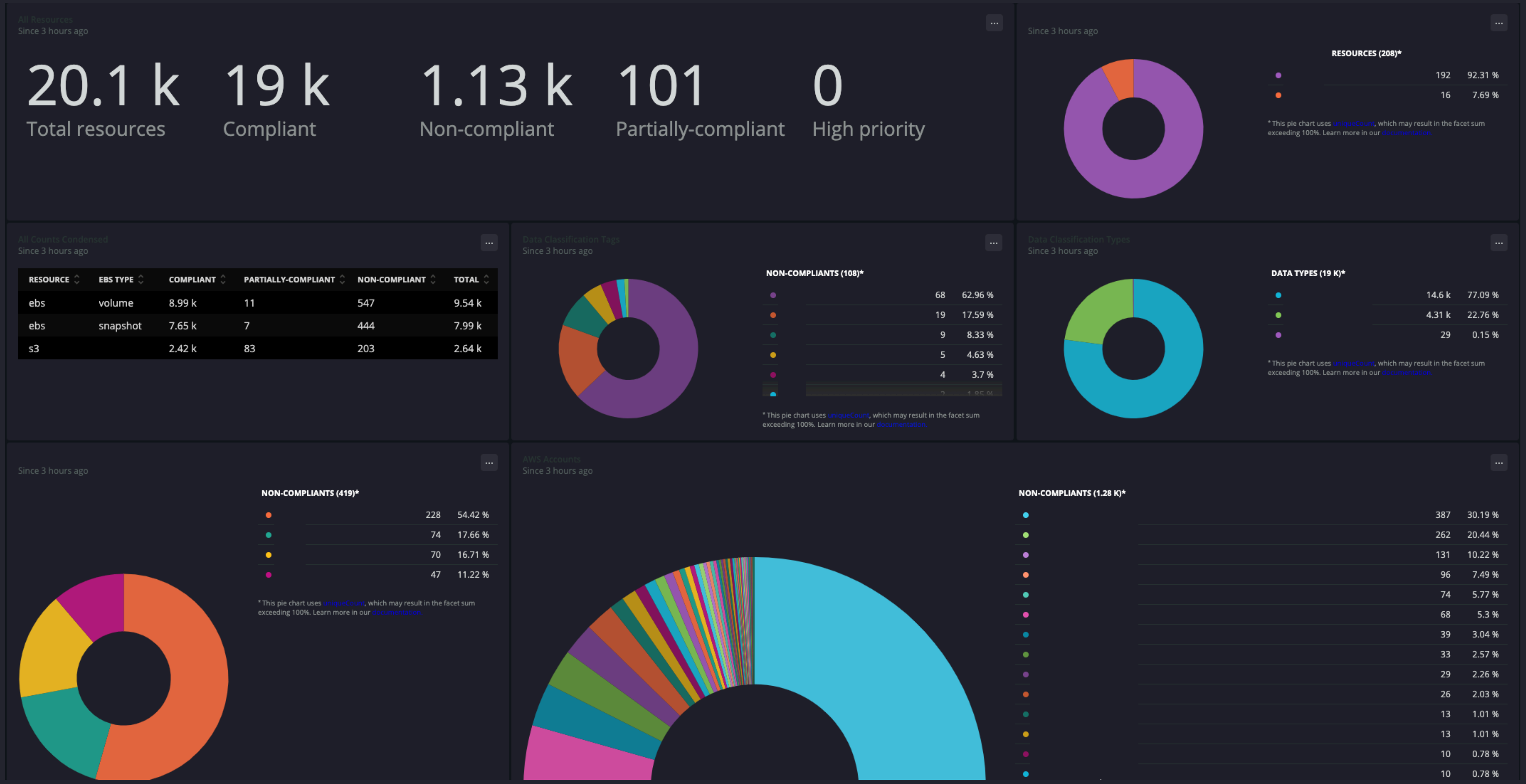
Policy components

- **Filters** (config)
- **Actions**

New Relic

Monitoring

- Resource inventory



All Resources
Since 3 hours ago

20.1 k
Total resources

19 k
Compliant

1.13 k
Non-compliant

101
Partially-compliant

0
High priority

Since 3 hours ago

RESOURCES (208)*

●	192	92.31 %
●	16	7.69 %

* This pie chart uses uniqueCount, which may result in the facet sum exceeding 100%. Learn more in our documentation.

All Counts Condensed
Since 3 hours ago

RESOURCE	EBS TYPE	COMPLIANT	PARTIALLY-COMPLIANT	NON-COMPLIANT	TOTAL
ebs	volume	8.99 k	11	547	9.54 k
ebs	snapshot	7.65 k	7	444	7.99 k
s3		2.42 k	83	203	2.64 k

Data Classification Tags
Since 3 hours ago

NON-COMPLIANTS (108)*

●	68	62.96 %
●	19	17.59 %
●	9	8.33 %
●	5	4.63 %
●	4	3.7 %
●	2	1.85 %

* This pie chart uses uniqueCount, which may result in the facet sum exceeding 100%. Learn more in our documentation.

Data Classification Types
Since 3 hours ago

DATA TYPES (19 K)*

●	14.6 k	77.09 %
●	4.31 k	22.76 %
●	29	0.15 %

* This pie chart uses uniqueCount, which may result in the facet sum exceeding 100%. Learn more in our documentation.

Since 3 hours ago

NON-COMPLIANTS (419)*

●	228	54.42 %
●	74	17.66 %
●	70	16.71 %
●	47	11.22 %

* This pie chart uses uniqueCount, which may result in the facet sum exceeding 100%. Learn more in our documentation.

AWS Accounts
Since 3 hours ago

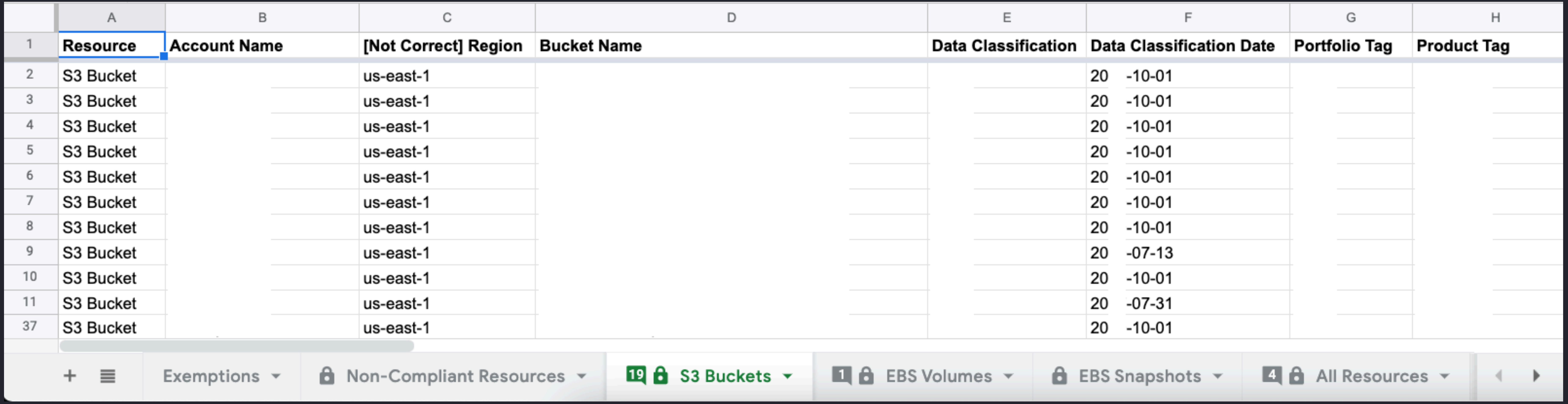
NON-COMPLIANTS (1.28 K)*

●	387	30.19 %
●	262	20.44 %
●	131	10.22 %
●	96	7.49 %
●	74	5.77 %
●	68	5.3 %
●	39	3.04 %
●	33	2.57 %
●	29	2.26 %
●	26	2.03 %
●	13	1.01 %
●	13	1.01 %
●	10	0.78 %
●	10	0.78 %

Google Sheets

Monitoring

- **Resource inventory…?**

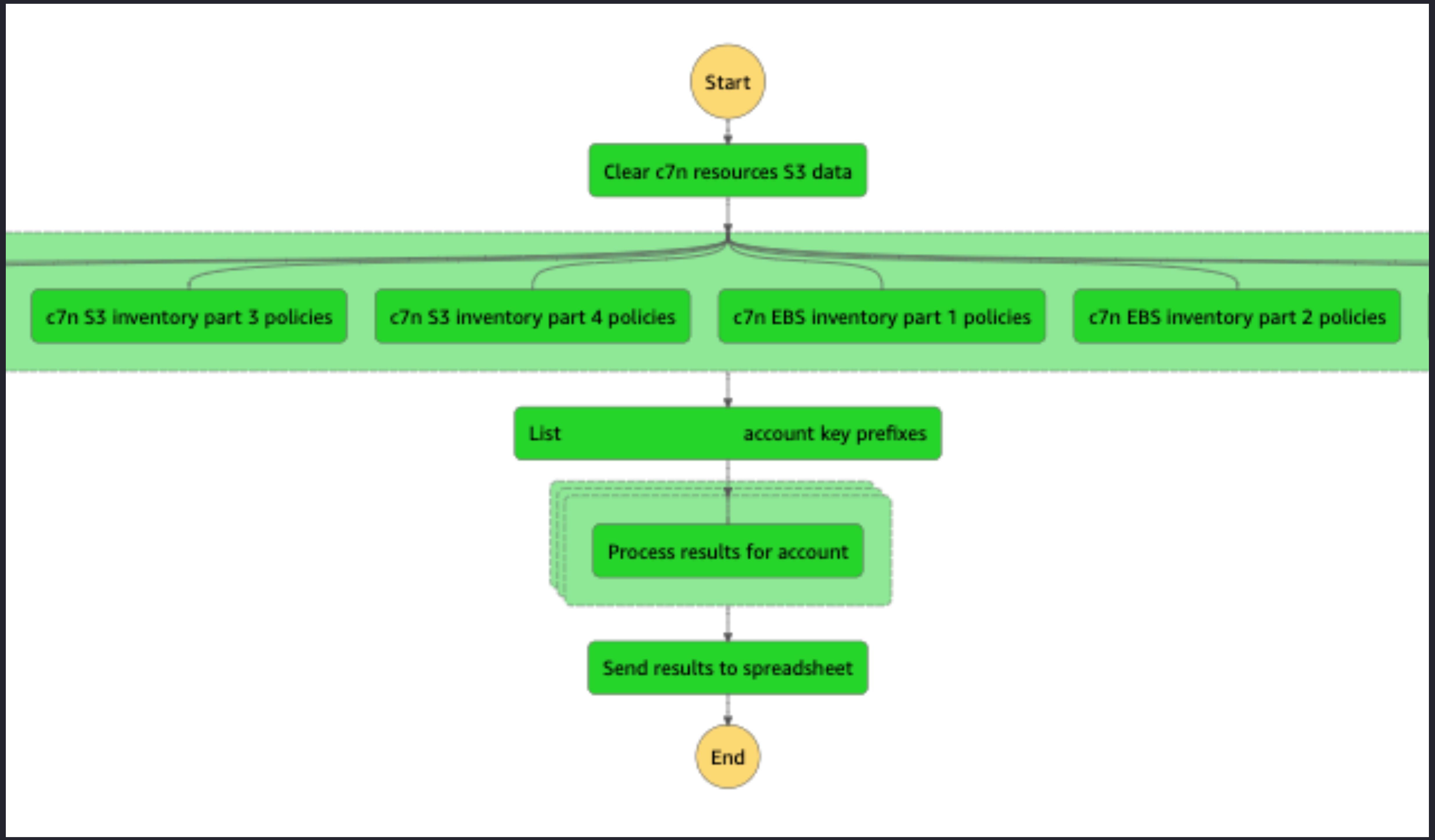


	A	B	C	D	E	F	G	H
1	Resource	Account Name	[Not Correct] Region	Bucket Name	Data Classification	Data Classification Date	Portfolio Tag	Product Tag
2	S3 Bucket		us-east-1			20 -10-01		
3	S3 Bucket		us-east-1			20 -10-01		
4	S3 Bucket		us-east-1			20 -10-01		
5	S3 Bucket		us-east-1			20 -10-01		
6	S3 Bucket		us-east-1			20 -10-01		
7	S3 Bucket		us-east-1			20 -10-01		
8	S3 Bucket		us-east-1			20 -10-01		
9	S3 Bucket		us-east-1			20 -07-13		
10	S3 Bucket		us-east-1			20 -10-01		
11	S3 Bucket		us-east-1			20 -07-31		
37	S3 Bucket		us-east-1			20 -10-01		

AWS Step Functions

Monitoring

- Resource inventory


Start
Clear c7n resources S3 data
c7n S3 inventory part 3 policies
c7n S3 inventory part 4 policies
c7n EBS inventory part 1 policies
c7n EBS inventory part 2 policies
List account key prefixes
Process results for account
Send results to spreadsheet
End

Deployment architecture

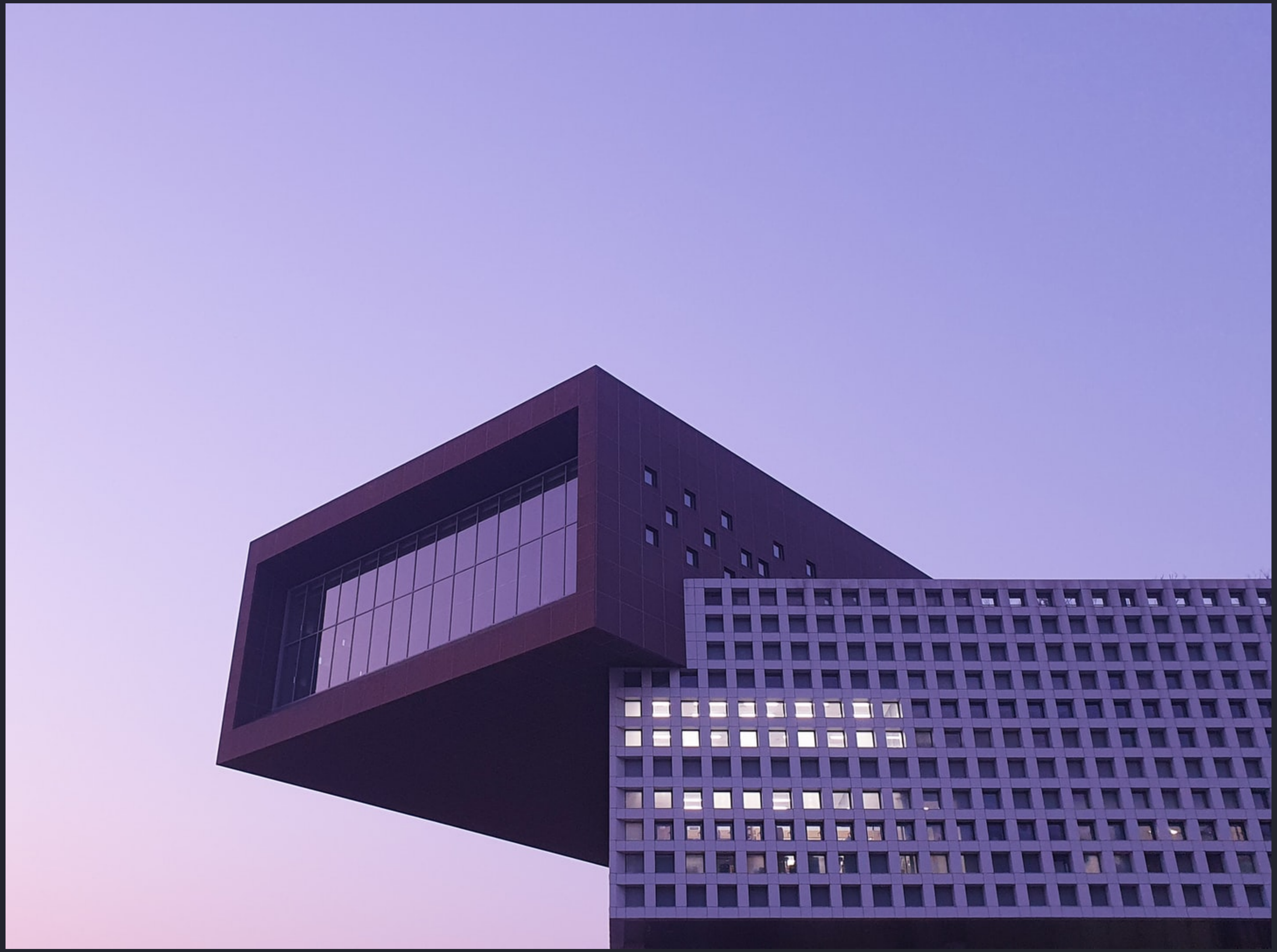
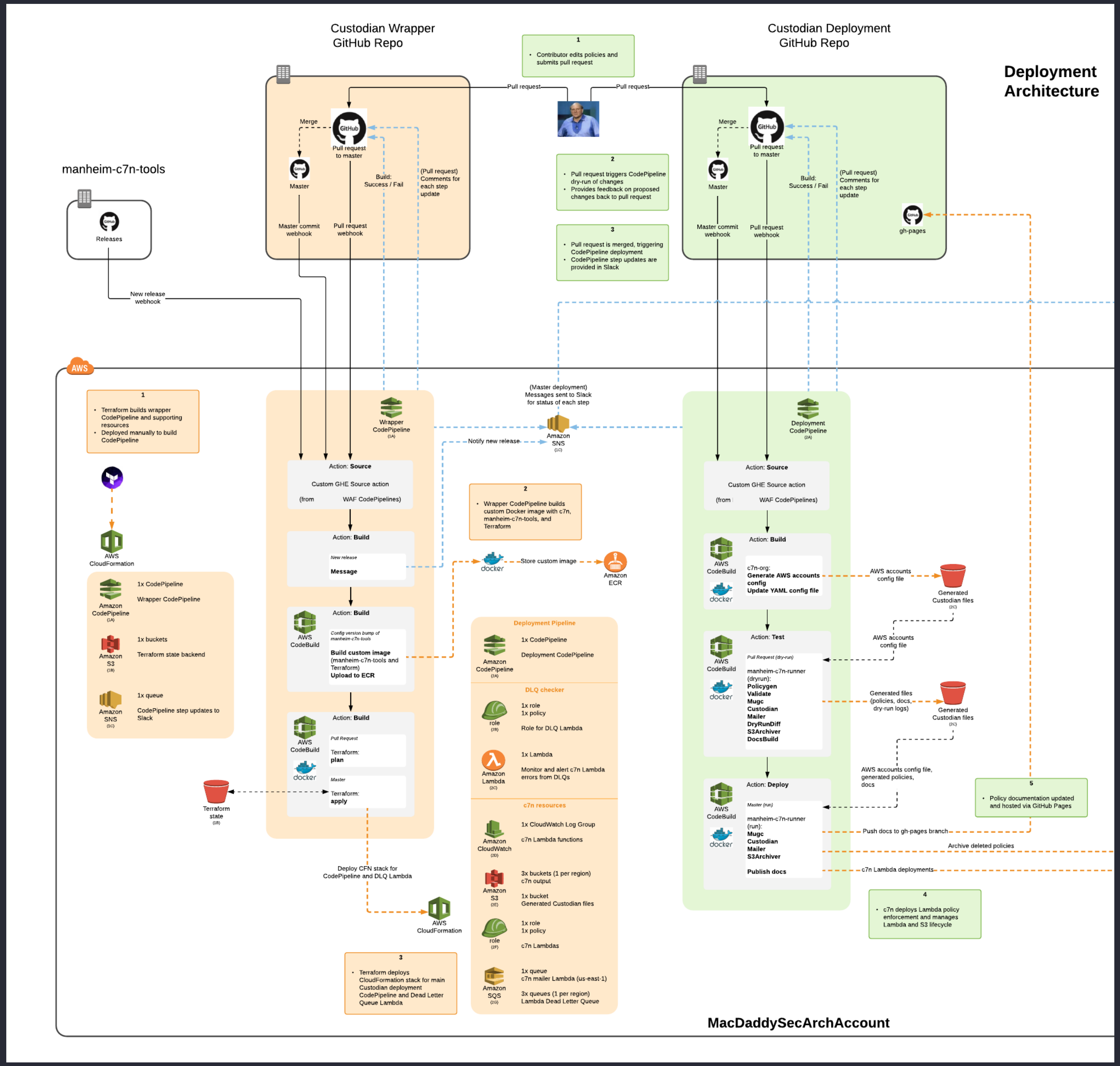


Photo by Jeong Yejune on Unsplash

Expectation vs. reality



1
- Contributor edits policies and submits pull request

2
- Pull request triggers CodePipeline dry-run of changes
- Provides feedback on proposed changes back to pull request

3
- Pull request is merged, triggering CodePipeline deployment
- CodePipeline step updates are provided in Slack

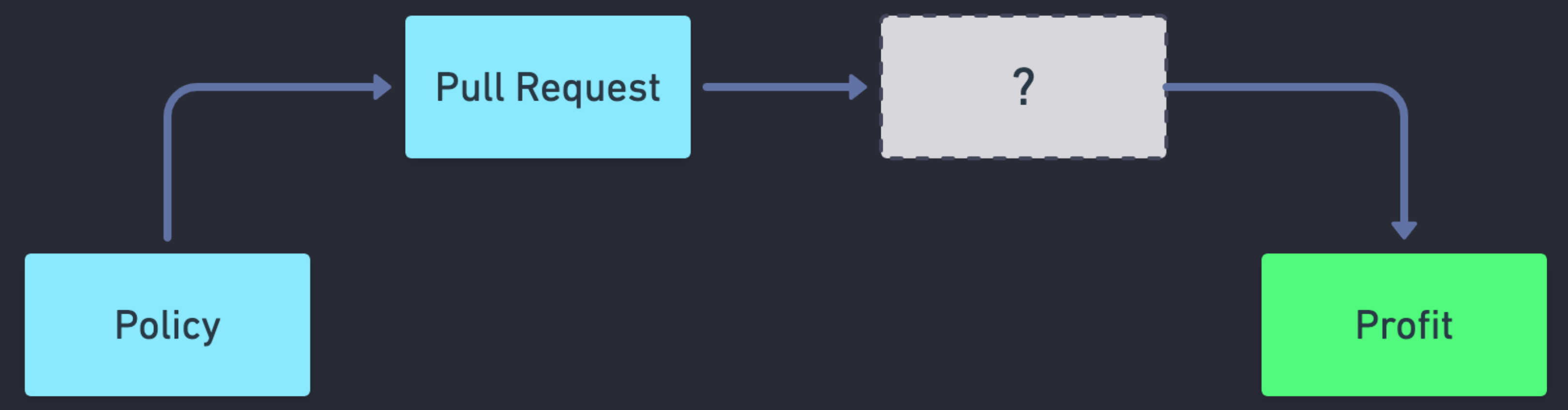
1
- Terraform builds wrapper CodePipeline and supporting resources
- Deployed manually to build CodePipeline

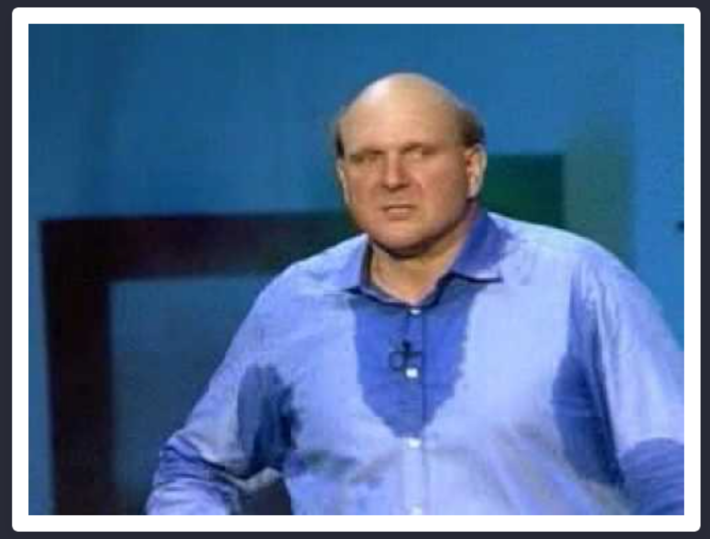
2
- Wrapper CodePipeline builds custom Docker image with c7n, manheim-c7n-tools, and Terraform

3
- Terraform deploys CloudFormation stack for main Custodian deployment CodePipeline and Dead Letter Queue Lambda

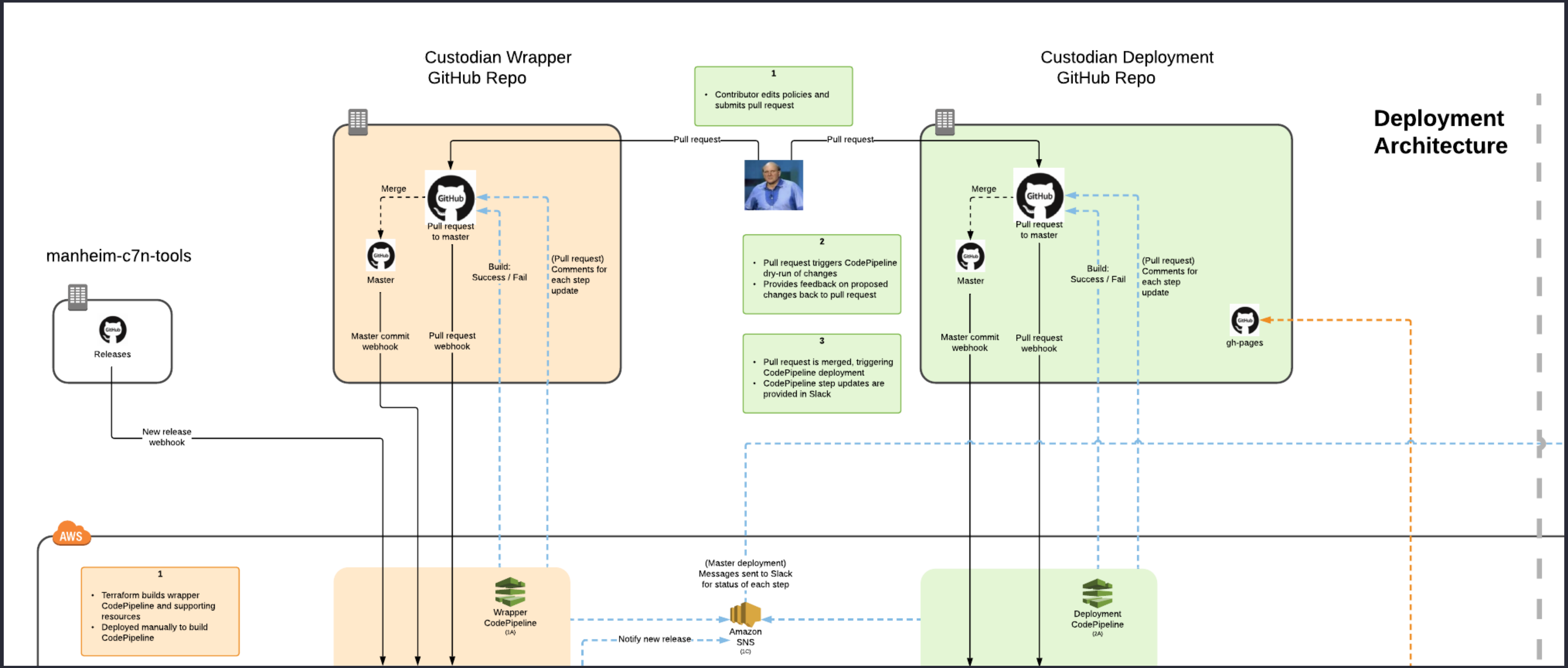
4
- c7n deploys Lambda policy enforcement and manages Lambda and S3 lifecycle

5
- Policy documentation updated and hosted via GitHub Pages





What we wanted



Deployment Architecture

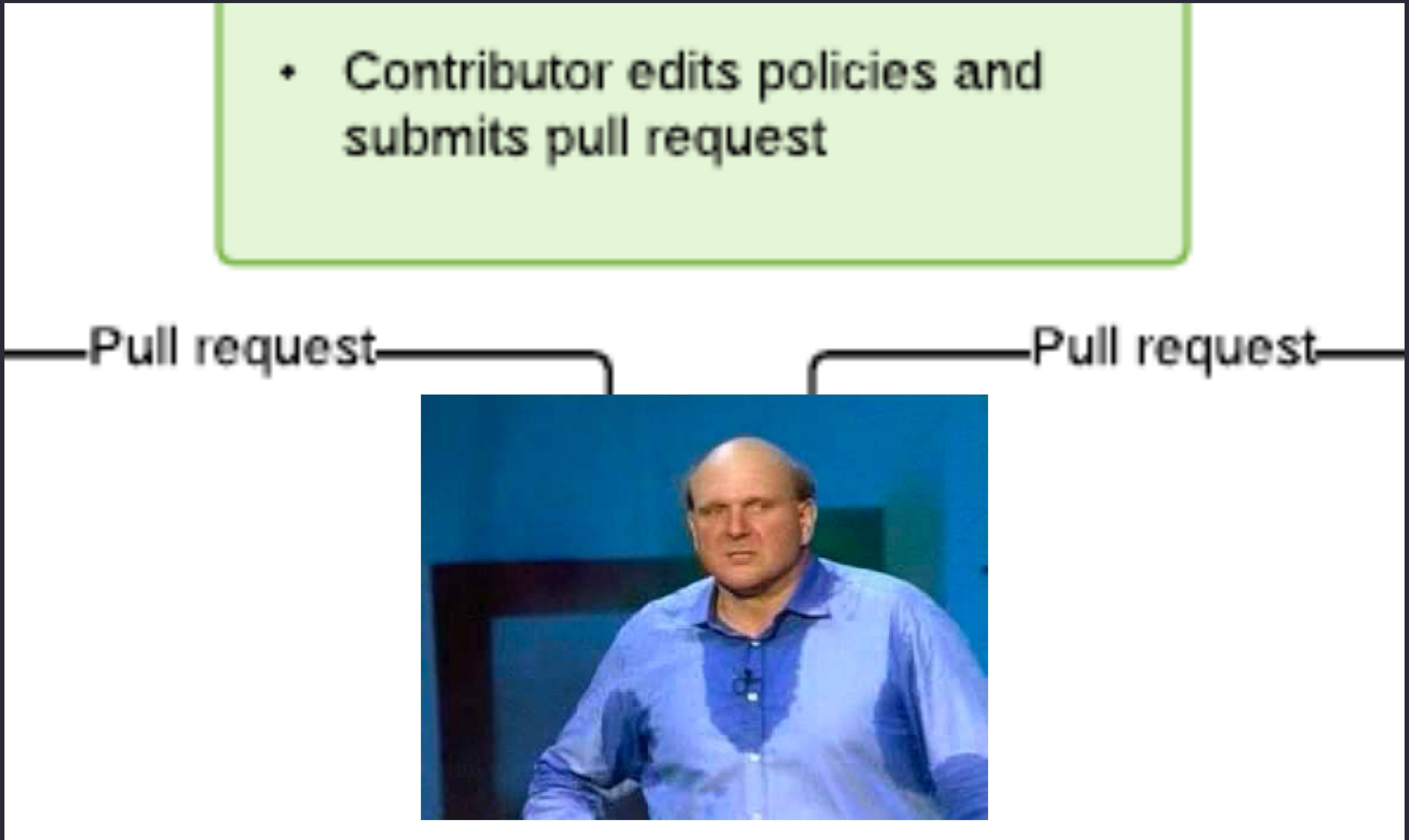
1
- Contributor edits policies and submits pull request

2
- Pull request triggers CodePipeline dry-run of changes
- Provides feedback on proposed changes back to pull request

3
- Pull request is merged, triggering CodePipeline deployment
- CodePipeline step updates are provided in Slack

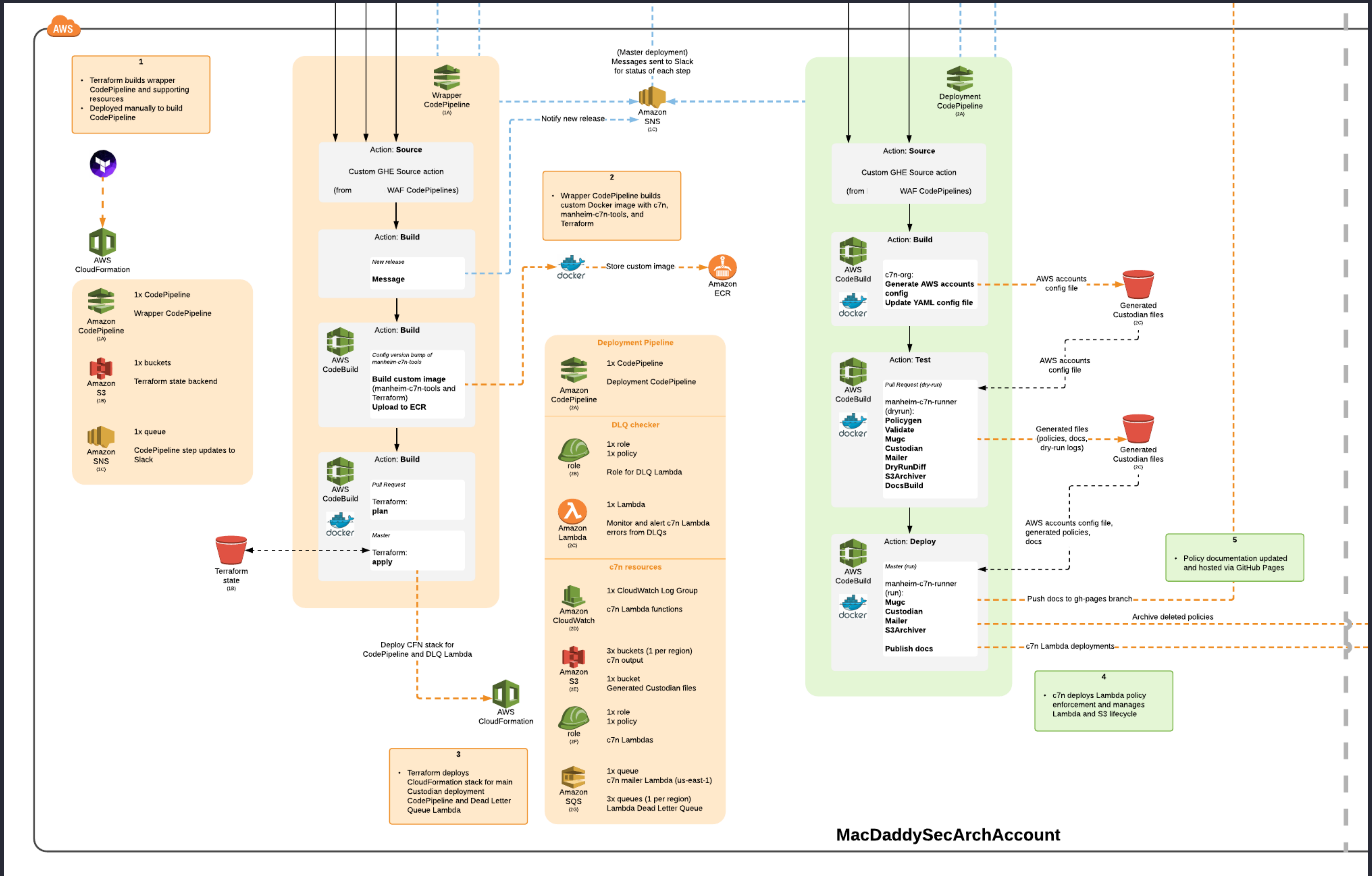
1
- Terraform builds wrapper CodePipeline and supporting resources
- Deployed manually to build CodePipeline

What we wanted



- Contributor edits policies and submits pull request

What we wanted



1

- Terraform builds wrapper CodePipeline and supporting resources
- Deployed manually to build CodePipeline

Amazon CodePipeline (1A)	1x CodePipeline Wrapper CodePipeline
Amazon S3 (1B)	1x buckets Terraform state backend
Amazon SNS (1C)	1x queue CodePipeline step updates to Slack

2

- Wrapper CodePipeline builds custom Docker image with c7n, manheim-c7n-tools, and Terraform

Deployment Pipeline

Amazon CodePipeline (2A)	1x CodePipeline Deployment CodePipeline

DLQ checker

role (2B)	1x role 1x policy Role for DLQ Lambda
Amazon Lambda (2C)	1x Lambda Monitor and alert c7n Lambda errors from DLQs

c7n resources

Amazon CloudWatch (2D)	1x CloudWatch Log Group c7n Lambda functions
Amazon S3 (2E)	3x buckets (1 per region) c7n output 1x bucket Generated Custodian files
role (2F)	1x role 1x policy c7n Lambdas
Amazon SQS (2G)	1x queue c7n mailer Lambda (us-east-1) 3x queues (1 per region) Lambda Dead Letter Queue

3

- Terraform deploys CloudFormation stack for main Custodian deployment CodePipeline and Dead Letter Queue Lambda

4

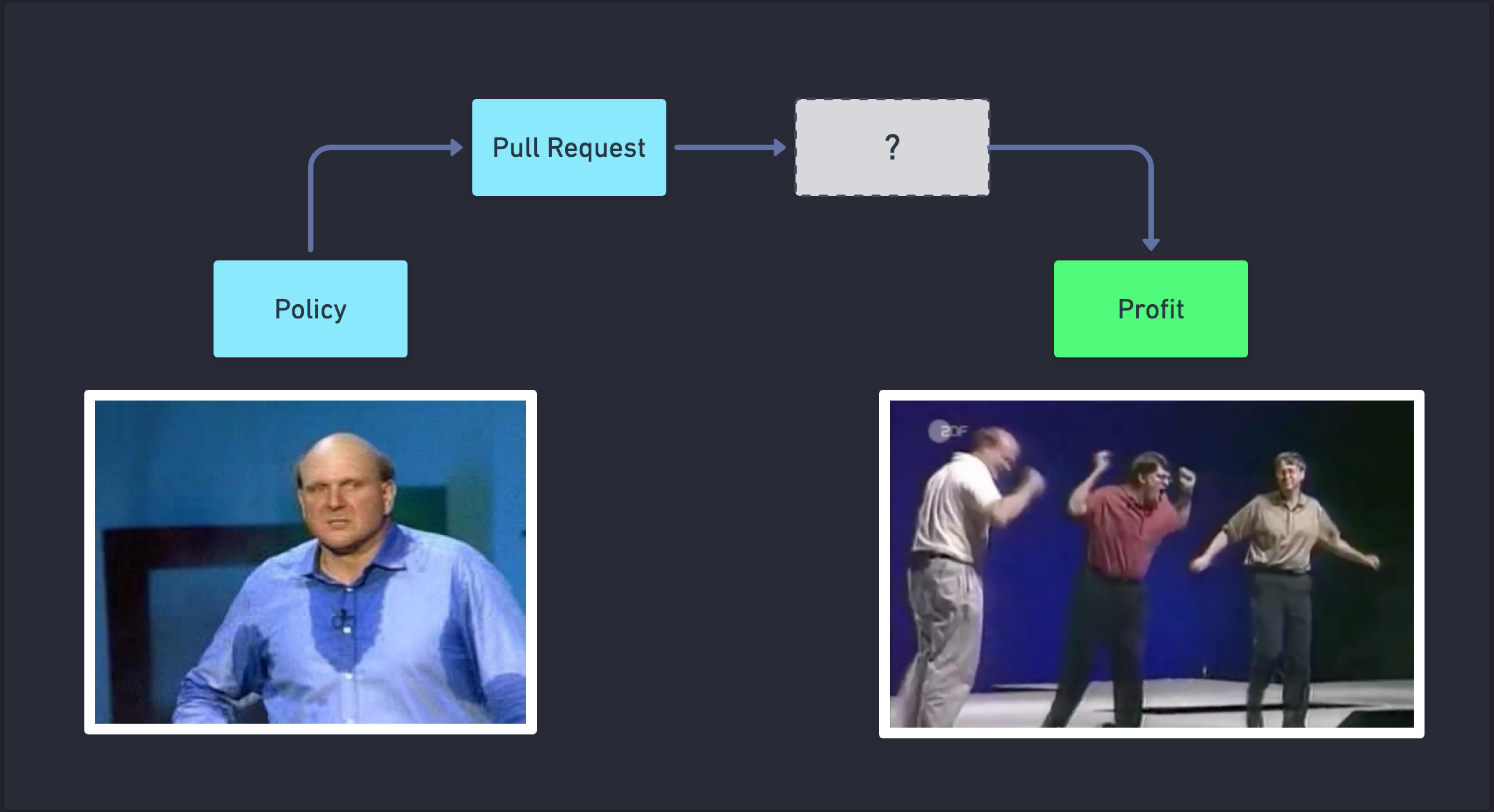
- c7n deploys Lambda policy enforcement and manages Lambda and S3 lifecycle

5

- Policy documentation updated and hosted via GitHub Pages

What we got

Handing over management of deployed policies



Policy

Pull Request

?

Profit

My experience

In practice

My experience: In practice

Enforcing encryption at rest

Project background

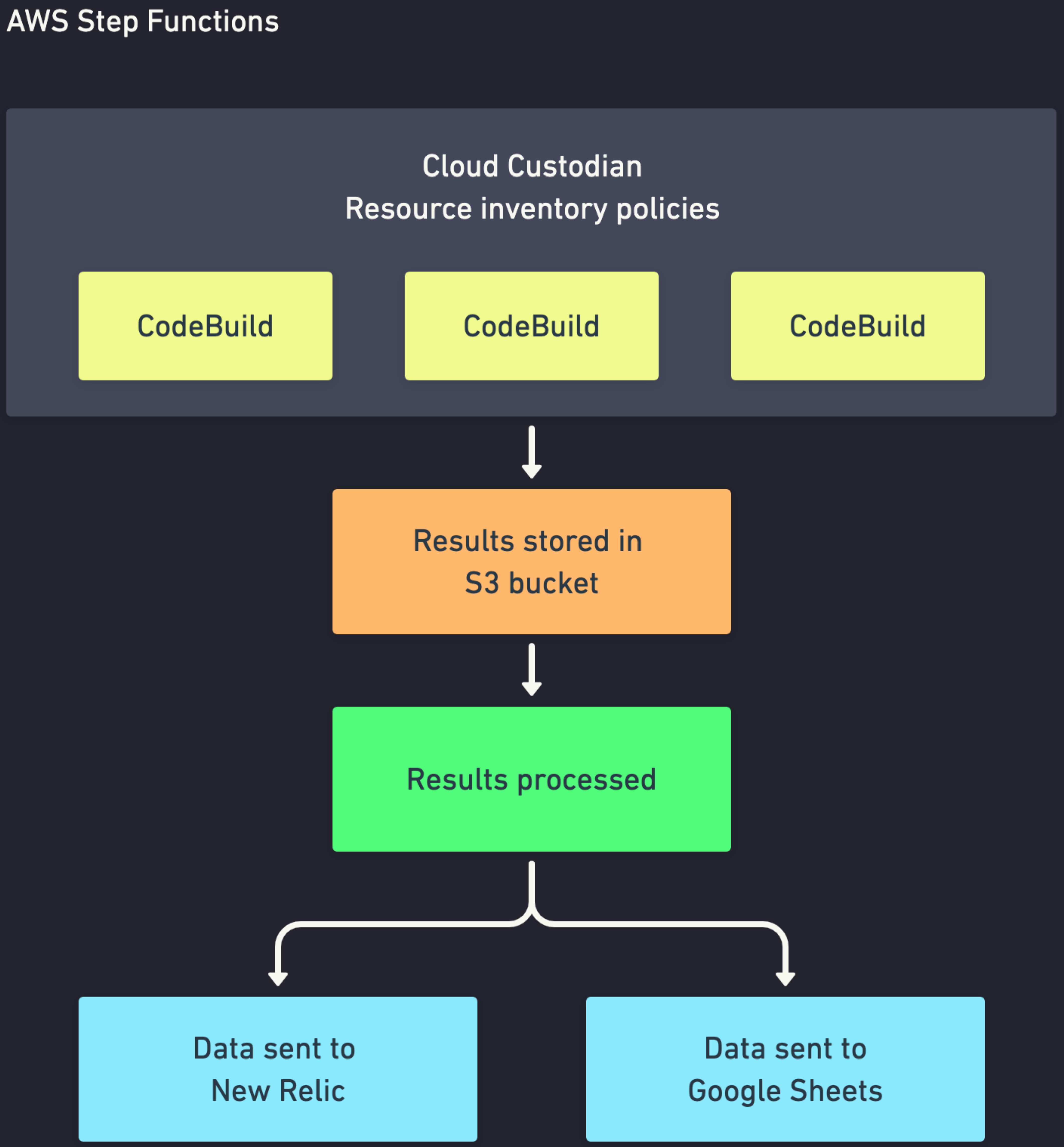
Ensure data is encrypted at rest

- Not enforced
- **S3** and **EBS** (EC2)
- **~30,000** resources

Solution

Resource inventory

Pipeline to **scan** resources, **process** data, and **report**



AWS Step Functions

Cloud Custodian
Resource inventory policies

CodeBuild

CodeBuild

CodeBuild

Results stored in S3 bucket

Results processed

Data sent to New Relic

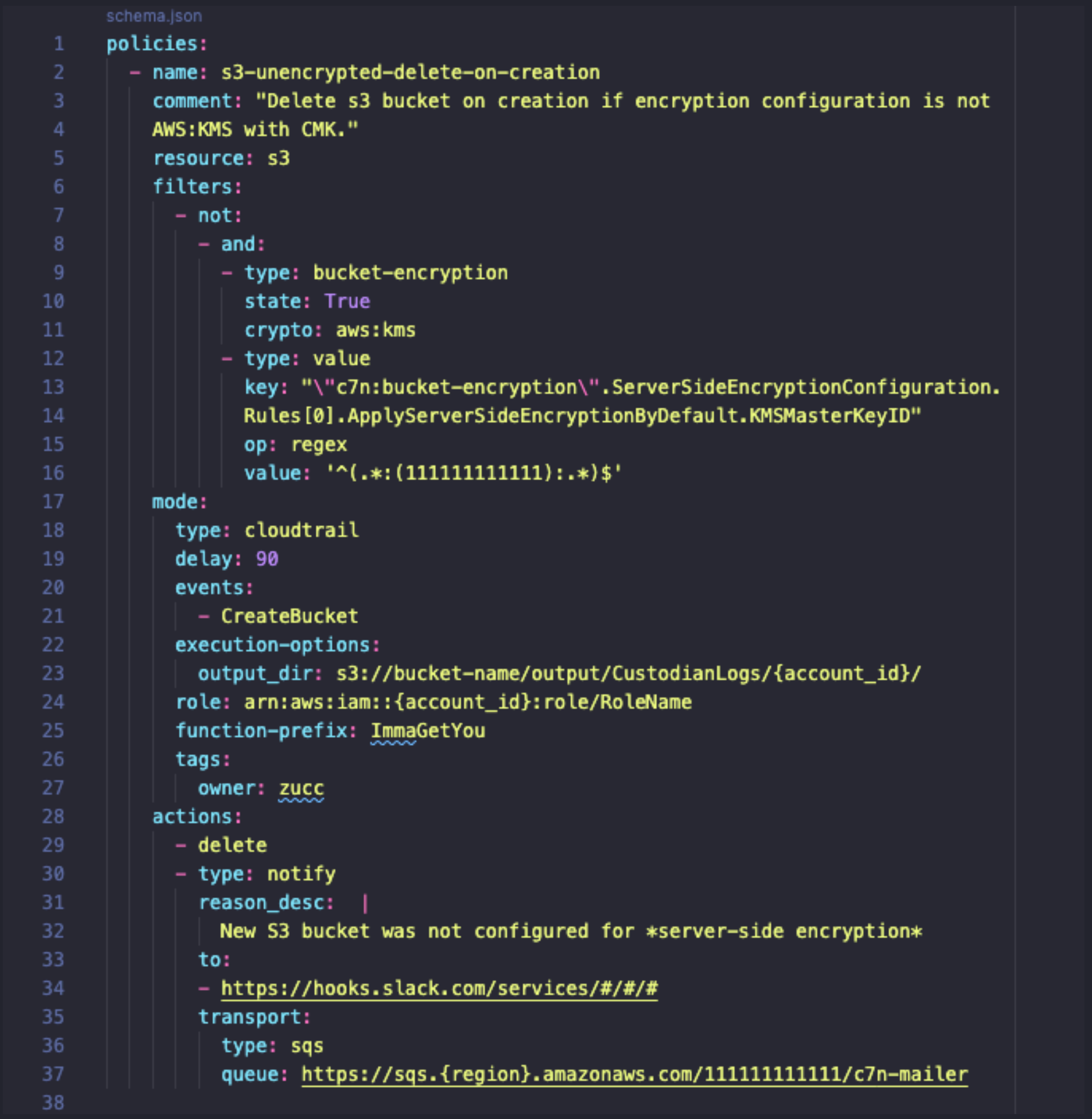
Data sent to Google Sheets

Remediation

S3

On creation

- Policy to **delete new** S3 buckets **without desired configuration**



schema.json

```yaml
policies:
  - name: s3-unencrypted-delete-on-creation
    comment: "Delete s3 bucket on creation if encryption configuration is not
    AWS:KMS with CMK."
    resource: s3
    filters:
      - not:
        - and:
          - type: bucket-encryption
            state: True
            crypto: aws:kms
          - type: value
            key: "\"c7n:bucket-encryption\".ServerSideEncryptionConfiguration.
            Rules[0].ApplyServerSideEncryptionByDefault.KMSMasterKeyID"
            op: regex
            value: '^(.*:(111111111111):.*)$'
    mode:
      type: cloudtrail
      delay: 90
      events:
        - CreateBucket
      execution-options:
        output_dir: s3://bucket-name/output/CustodianLogs/{account_id}/
      role: arn:aws:iam::{account_id}:role/RoleName
      function-prefix: ImmaGetYou
      tags:
        owner: zucc
    actions:
      - delete
      - type: notify
        reason_desc: |
          New S3 bucket was not configured for *server-side encryption*
        to:
        - https://hooks.slack.com/services/#/#/#
        transport:
          type: sqs
          queue: https://sqs.{region}.amazonaws.com/111111111111/c7n-mailer
```

Prevention

EBS

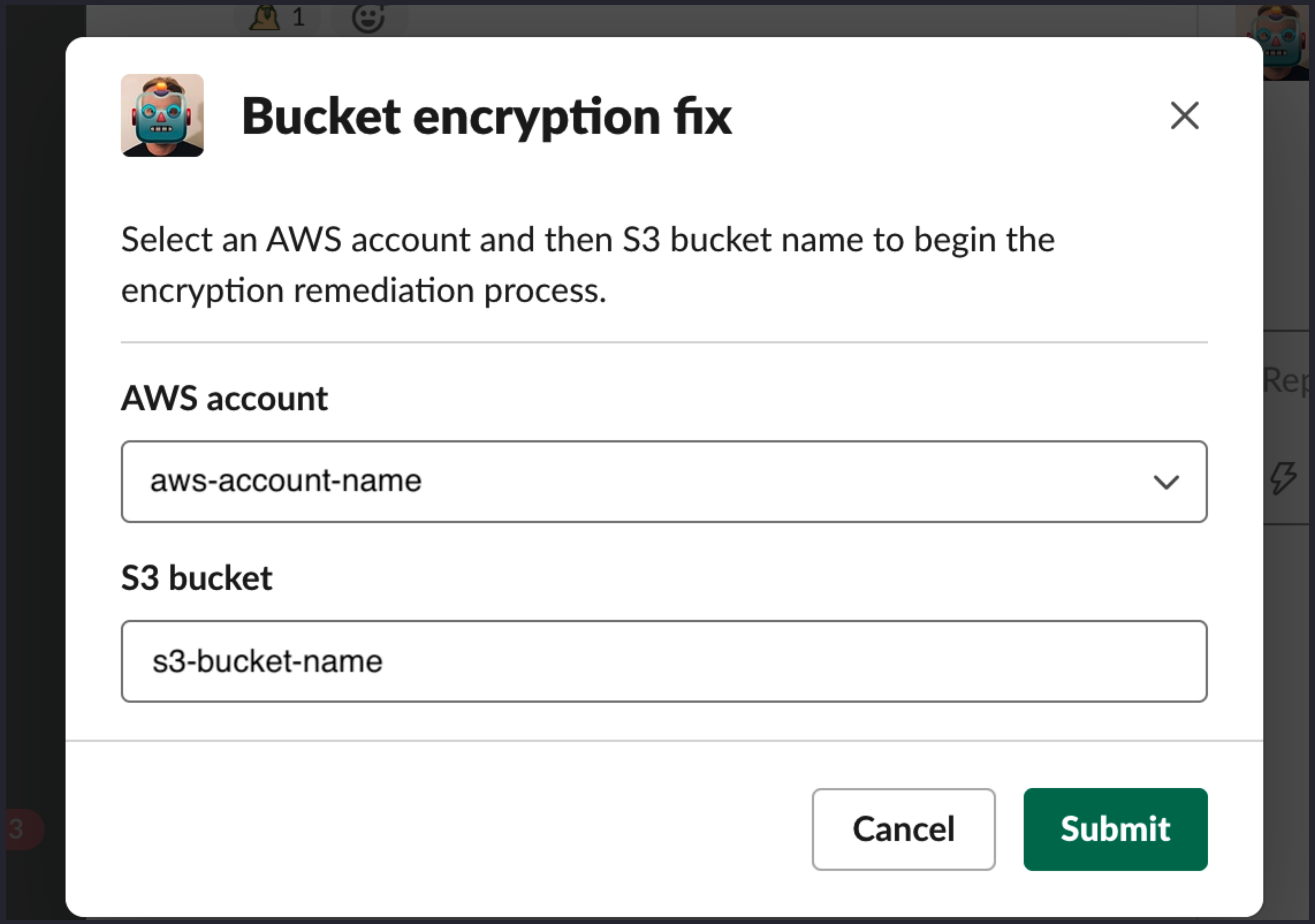
EC2 account setting

- Enables **default encryption** for **new** EBS volumes and snapshots
- On-demand policy to manually run

Prevention

S3

Batch S3 object remediation Slack bot



Bucket encryption fix

Select an AWS account and then S3 bucket name to begin the encryption remediation process.

AWS account

aws-account-name

S3 bucket

s3-bucket-name

Challenges

Challenges

Tools

Cloud Custodian

Bugs, docs, troubleshooting

- Plenty of **troubleshooting**
- Resource configuration caching

No concurrency

- All policies run **synchronously**

Resource inventory

- **No inventory capability**
- Hacked together inventory policies to build whole picture
    - **~30 policies**, for 3 resources
- Go with another tool like AWS config

Challenges

Tools

Google Sheets

- **40,000+ rows**
- **Slow** to use
- Sometimes API updates **timed out**

New Relic

- **Limited queries**, especially with large data

Ownership issues

Lots of unowned / unmaintained resources

- **Temporary teams** or pods, resources left behind
- **Hardest** part of the project
- Had to solve this before encryption

Challenges

Political

- Difficult to get **buy-in** from senior management
- Difficult to get **engagement** from developers and engineers

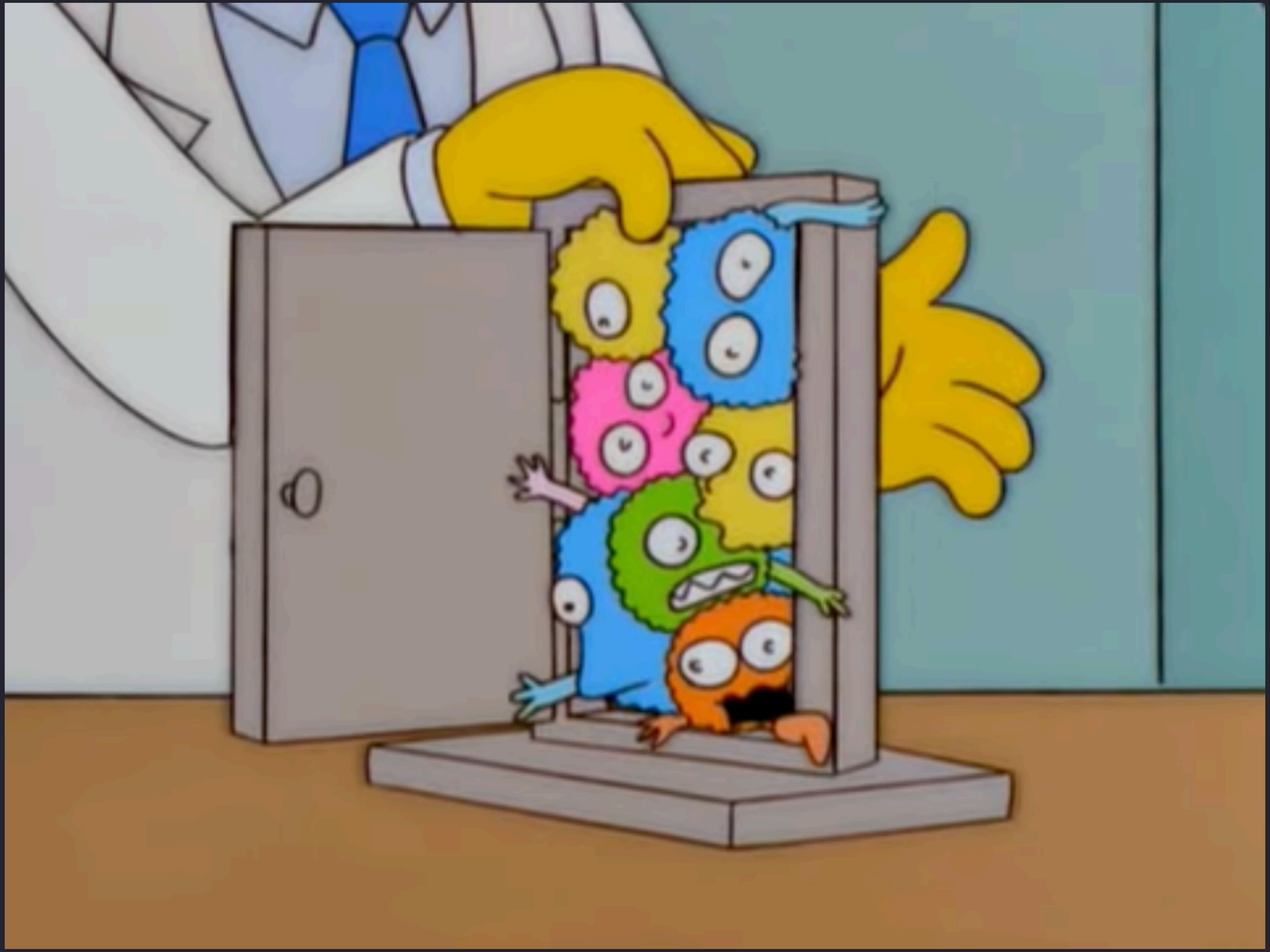
Challenges

The ol' Swedish handoff

- Got handed the project with **6 months** to complete
- **Rushed** solution to roll out massive organisational changes

Priorities

- Everything's a priority, so **nothing** is



Summary

Summary

Continuous Assurance

- Important for all
- Solution to insecure infrastructure
- Well-managed cloud
- Easier auditing
- Scale
- Choose your tools wisely
- Enterprise issues

Want to chat / feedback?

linkedin.com/in/francesco-badraun

Summary

ZX Security — Hiring

ZX Security is currently hiring

We're on the lookout for Security Consultants (Junior – Lead) across:

- ZXPenTest
- ZXCloud
- Cyber Strategy & Risk

Reach out to anyone at ZX for more info.