
AWS
re:Invent

SEC325

Event-driven data security using Amazon Macie

Neha Joshi

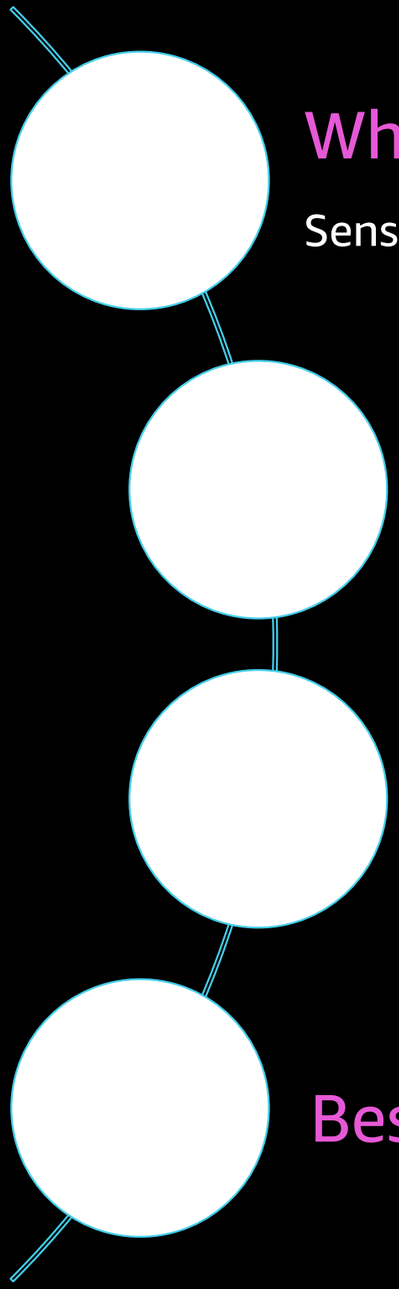
Sr. Technical Account Manager

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda



What's new?

Sensitive data discovery, data discovery jobs frequency

Event driven approach

Workflow, architecture, and configuration steps

Demo

Best practices

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What’s new?



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What's new about Amazon Macie

Price reduction

Enhancements

Amazon S3 Inventory

Redesigned console



Amazon Macie



“Protecting your customers should be your number one priority; without that you don’t have a business.”

Werner Vogels

VP and CTO
AWS

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sensitive data discovery



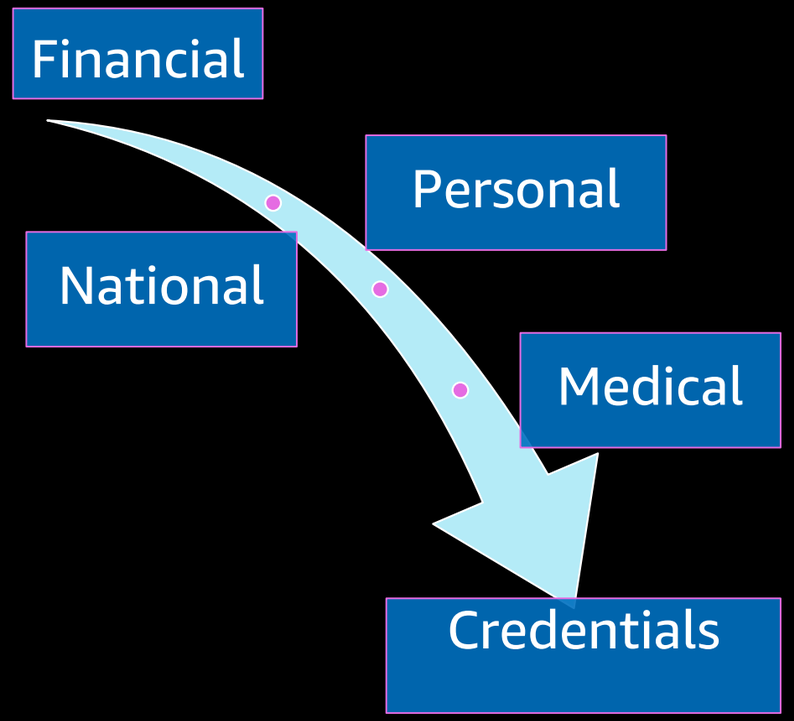
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data identifiers

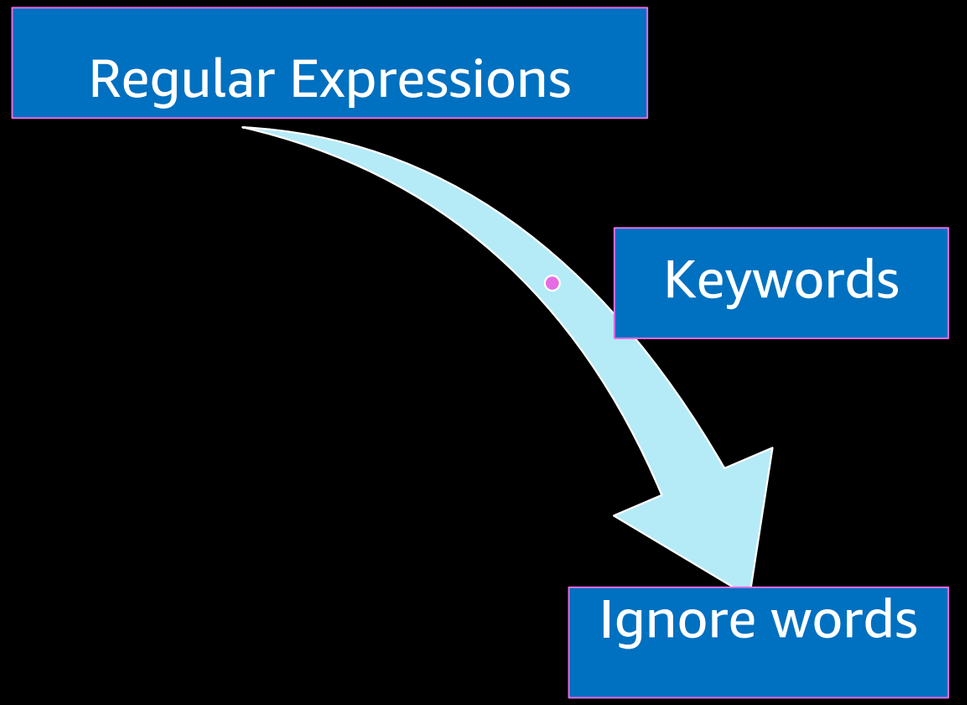
MANAGED

Maintains growing list of sensitive data types defined by data privacy regulations

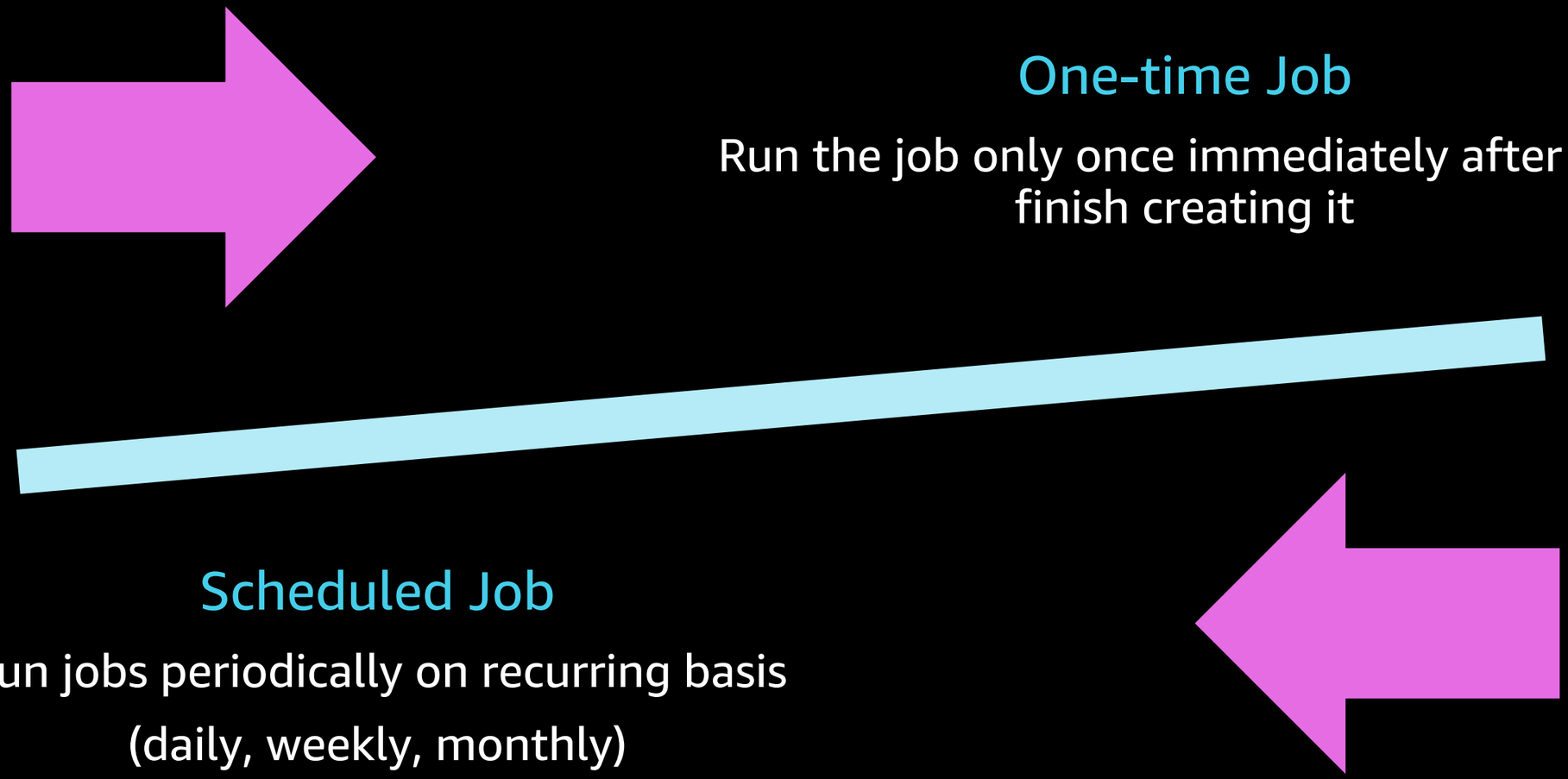


CUSTOM

Add custom-defined data types to discover proprietary or unique sensitive data for your business



Data discovery jobs frequency



One-time Job

Run the job only once immediately after you finish creating it

Scheduled Job

Run jobs periodically on recurring basis

(daily, weekly, monthly)



Not sure about frequency of new object(s) being added in target S3 bucket(s)?



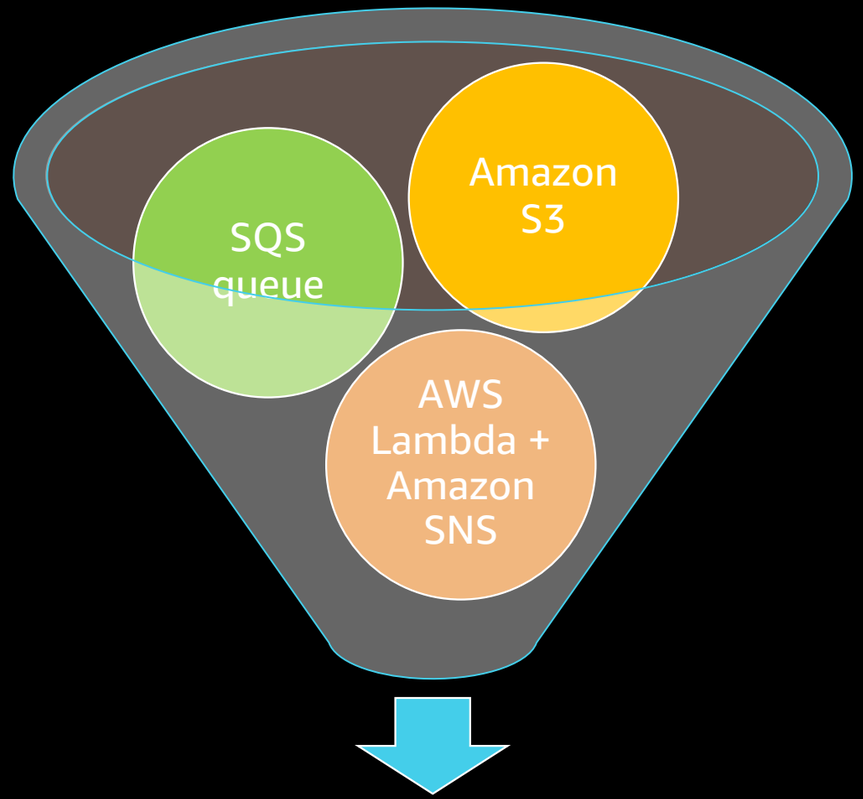
How do you ensure that you're optimizing the data discovery runs?

Solution:
Event-driven approach



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

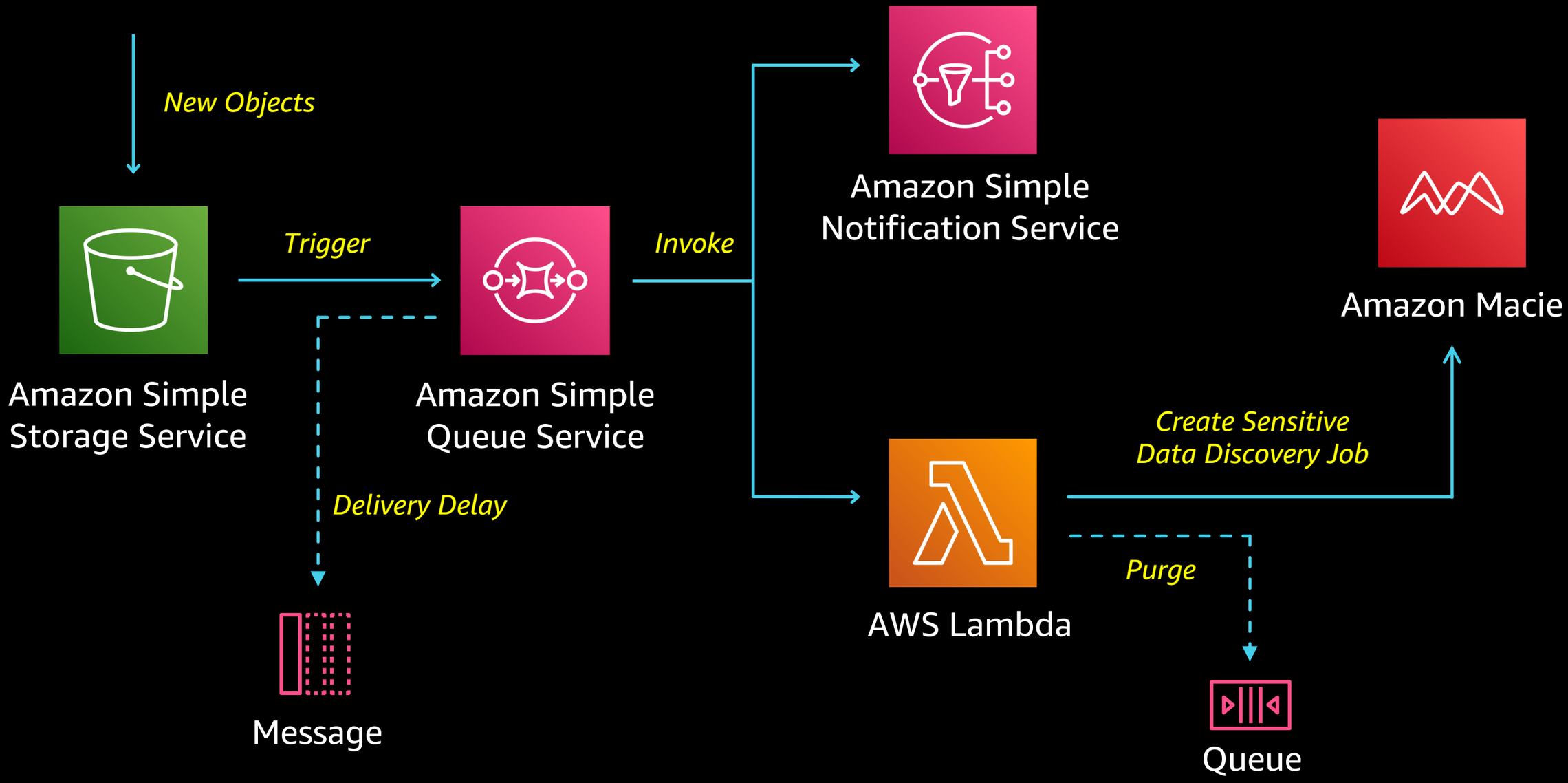
Design workflow



Amazon Macie

1. New object(s) added to selected S3 bucket(s)
2. Trigger Amazon SQS
3. Invoke:
   - AWS Lambda Function
   - SNS notification
4. Data Discovery Job in Macie

Event-driven architecture



Configuration steps

PRE-REQUISITES

1

SNS

Notifications

2

SQS Queue

Create Data Discovery Job

CONFIGURATION

3

Lambda Function

Messages Queue

4

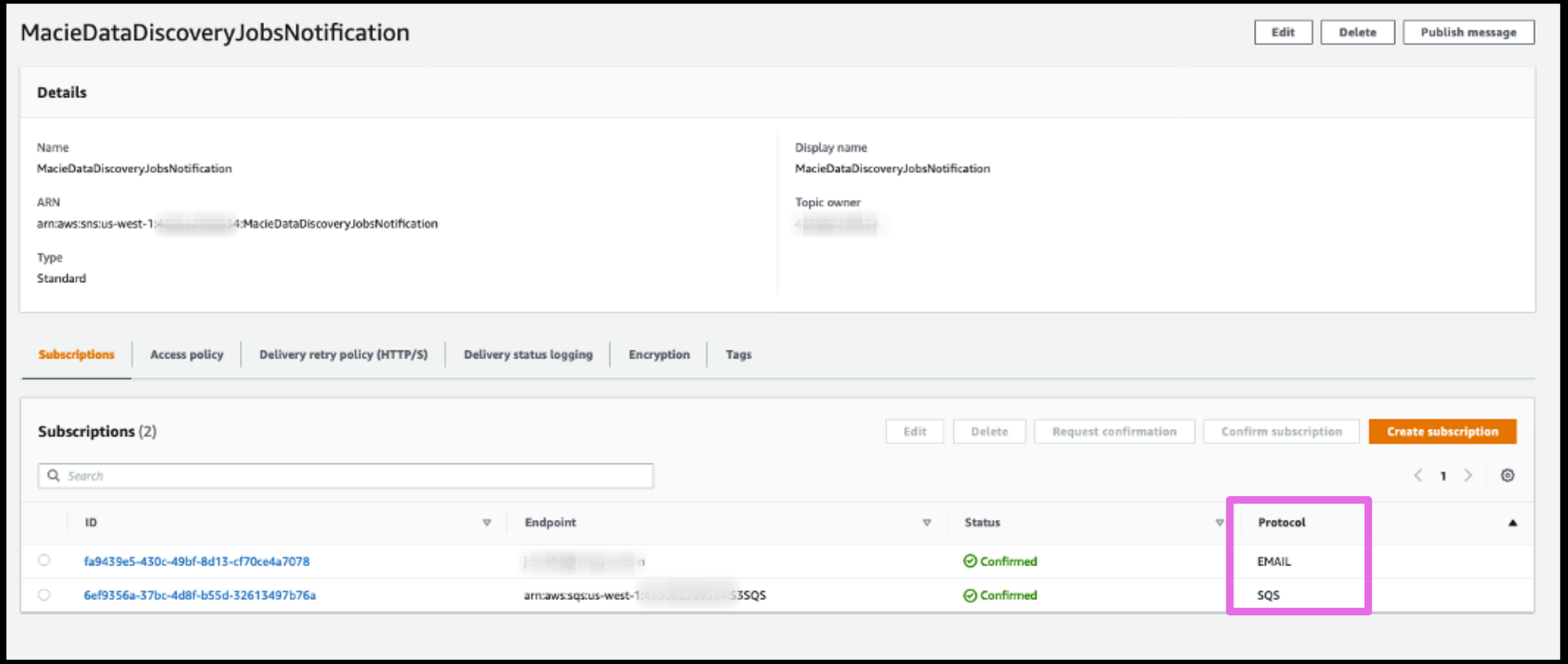
S3 Events

S3 Event Notifications

Configuration steps

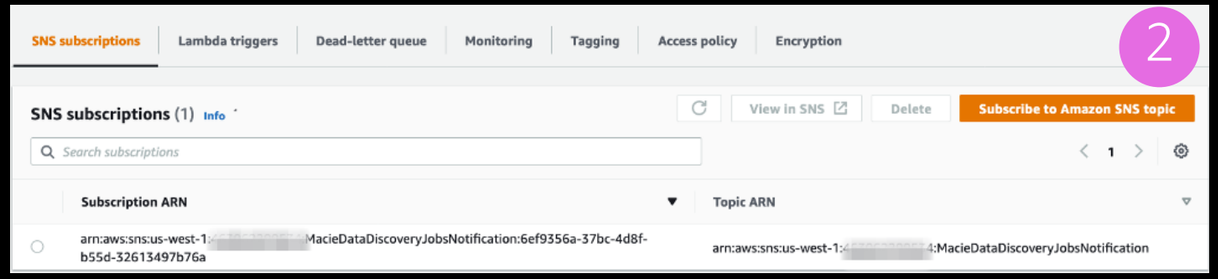
AMAZON SNS:

Create and subscribe to an SNS topic for notifications

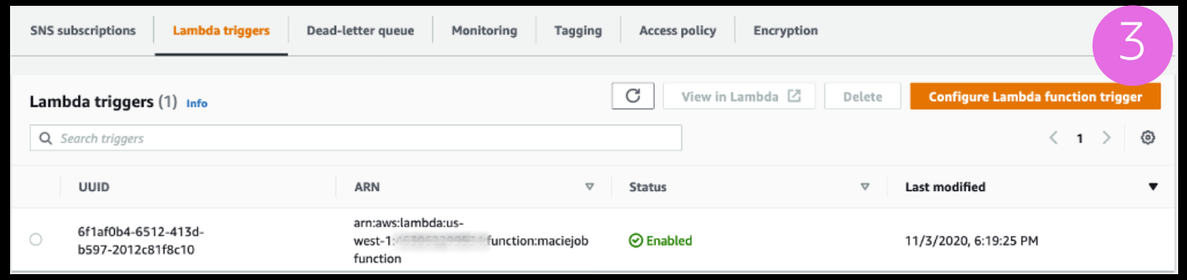


Configuration steps

SQS QUEUE:



SNS Subscription



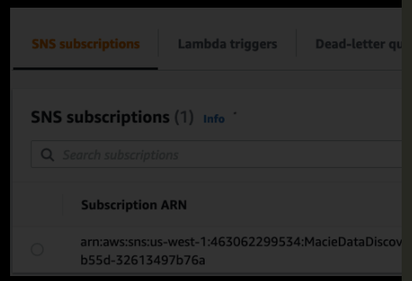
Lambda Trigger

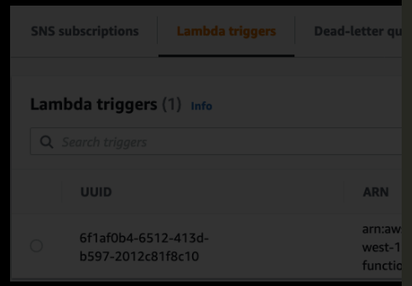


SQS Access Policy

Configuration steps

SQS QUEUE:



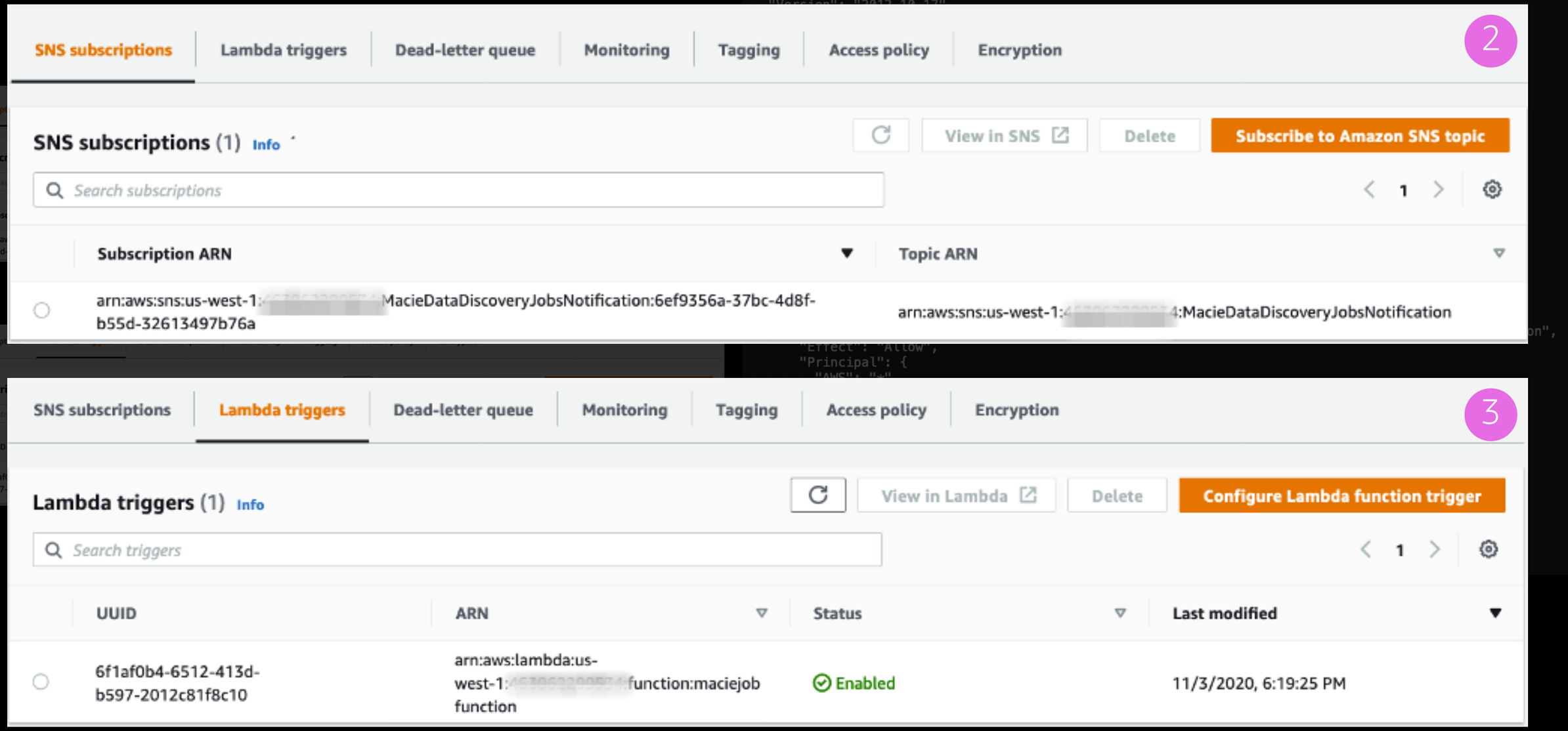


1

```
{
  "Version": "2012-10-17",
  "Id": "example-ID",
  "Statement": [
    {
      "Sid": "example-statement-ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:us-west-1:[redacted]:S3SQS",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "[redacted]"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:[redacted]"
        }
      }
    },
    {
      "Sid": "topic-subscription-arn:aws:sns:us-west-1:[redacted]MacieDataDiscoveryJobsNotification",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:us-west-1:[redacted]:S3SQS",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:sns:us-west-1:[redacted]:MacieDataDiscoveryJobsNotification"
        }
      }
    }
  ]
}
```

Configuration steps

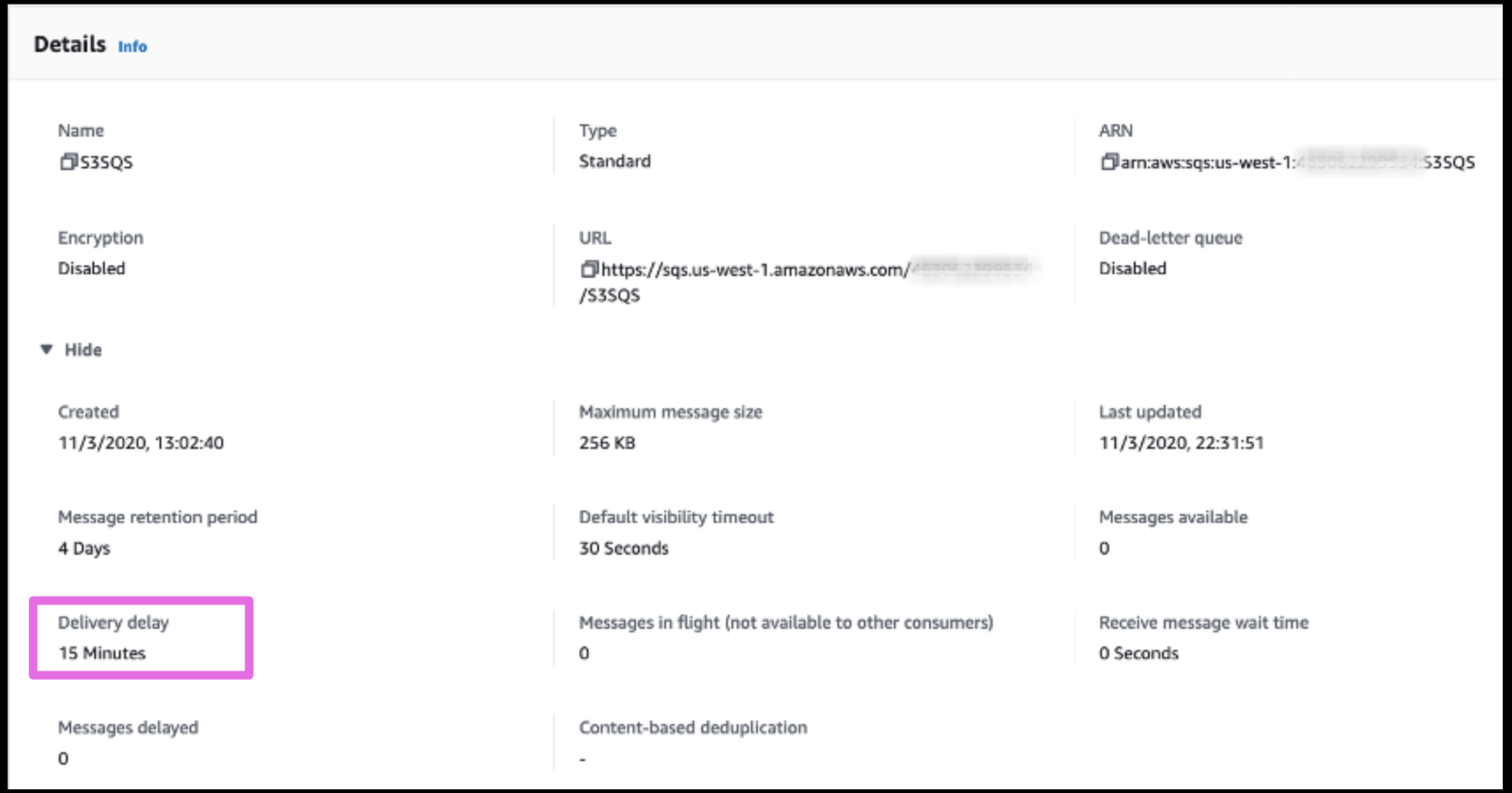
SQS QUEUE:



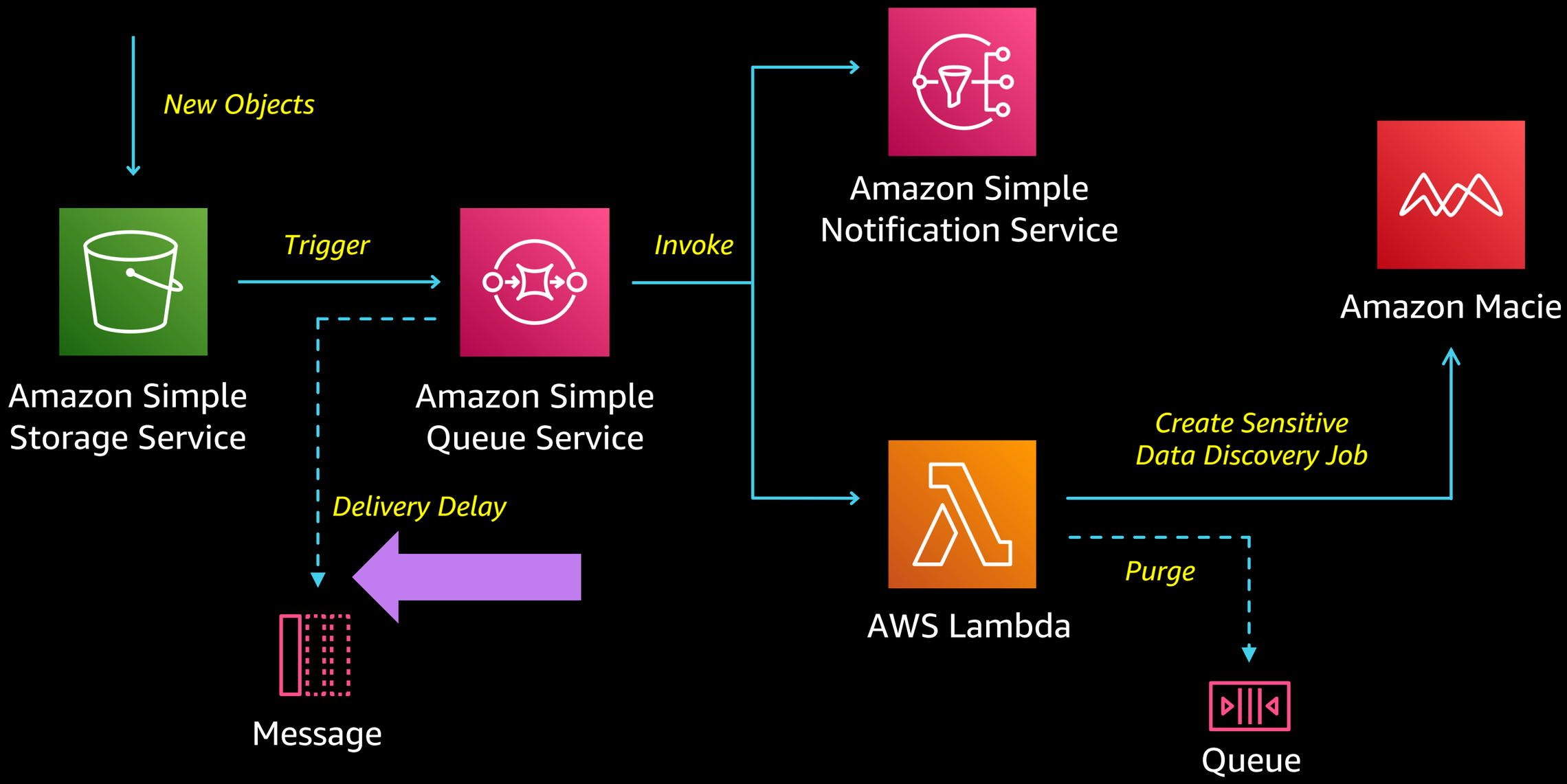
Configuration steps

SQS QUEUE:

SQS Queue Details



Event-driven architecture



Configuration steps

LAMBDA FUNCTION:

1

```
import boto3
import string
import random
import json
import datetime
from datetime import datetime, timedelta, timezone
import time


def lambda_handler(event, context):

    client1= boto3.client('macie2', 'us-west-1')

    N = 4
    res = ''.join(random.choices(string.ascii_uppercase + string.digits, k = N))
    jobname = 'customjob_test_'+ str(res)
    current_datetime = datetime.now()
    old_datetime = current_datetime - timedelta(minutes = 15)
    old_datetime = old_datetime.isoformat()
    old_datetime = old_datetime [:-3]
    old_datetime = old_datetime + 'Z'
```

2

```
response1 = client1.create_classification_job(
customDataIdentifierIds=[
    '[redacted]',
],
description='custom identifier job',
initialRun=True,
jobType='ONE_TIME',
name='string',
s3JobDefinition={
    'bucketDefinitions': [
        {
            'accountId': '[redacted]',
            'buckets': [
                '[redacted]',
            ]
        },
    ],
    'scoping': {
        'excludes': {
            'and': [
                {
                    'simpleScopeTerm': {
                        'comparator': 'LT',
                        'key': 'OBJECT_LAST_MODIFIED_DATE',
                        'values': [
                            old_datetime,
                        ]
                    },
                },
            ]
        },
    }
},
)
print(response1)
```

3

```
client2 = boto3.client('sqs', 'us-west-1')
response2 = client2.purge_queue(
    QueueUrl='[redacted]')

print(response2)
```

Configuration steps

LAMBDA FUNCTION:

```python
import boto3
import string
import random
import json
import datetime
from datetime import datetime, timedelta, timezone
import time


def lambda_handler(event, context):

    client1= boto3.client('macie2', 'us-west-1')

    N = 4
    res = ''.join(random.choices(string.ascii_uppercase + string.digits, k = N))
    jobname = 'customjob_test_'+ str(res)
    current_datetime = datetime.now()
    old_datetime = current_datetime - timedelta(minutes = 15)
    old_datetime = old_datetime.isoformat()
    old_datetime = old_datetime [:-3]
    old_datetime = old_datetime + 'Z'
```

Configuration steps

LAMBDA FUNCTION:

```
import boto3
import string
import random
import json
import datetime
from datetime import datetime, timedelta, timezone
import time


def lambda_handler(event, context):

    client1= boto3.client('macie2', 'us-west-1')

    N = 4
    res = ''.join(random.choices(string.ascii_uppercase + string
    jobname = 'customjob_test_'+ str(res)
    current_datetime = datetime.now()
    old_datetime = current_datetime - timedelta(minutes = 15)
    old_datetime = old_datetime.isoformat()
    old_datetime = old_datetime [:-3]
    old_datetime = old_datetime + 'Z'
```

```
client2 = boto3.client('sqs', 'us-west-1')
response2 = client2.purge_queue(
    QueueUrl='https://sqs.us-west-1.amazonaws.com/463062299

print(response2)
```

```
response1 = client1.create_classification_job(
customDataIdentifierIds=[
    '[redacted]',
],
description='custom identifier job',
initialRun=True,
jobType='ONE_TIME',
name='string',
s3JobDefinition={
    'bucketDefinitions': [
        {
            'accountId': '[redacted]',
            'buckets': [
                '[redacted]',
            ]
        },
    ],
    'scoping': {
        'excludes': {
            'and': [
                {
                    'simpleScopeTerm': {
                        'comparator': 'LT',
                        'key': 'OBJECT_LAST_MODIFIED_DATE',
                        'values': [
                            old_datetime,
                        ]
                    },
                },
            ]
        },
    }
},
)
print(response1)
```

Configuration steps

LAMBDA FUNCTION:

```
import boto3
import string
import random
import json
import datetime
from datetime import datetime, timedelta, timezone
import time

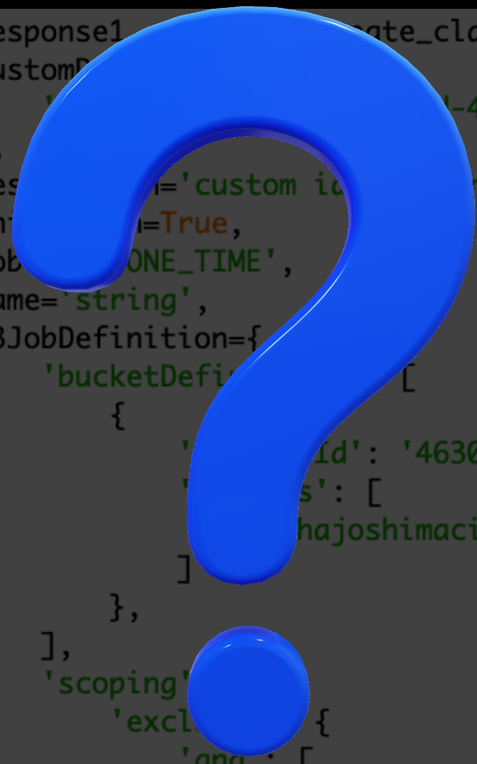

def lambda_handler(event, context):

    client1= boto3.client('macie2', 'us-west-1')

    N = 4
    res = ''.join(random.choices(string.ascii_uppercase + string.digits, k = N))
    job
    cu
    old
    old
    old
    old
```

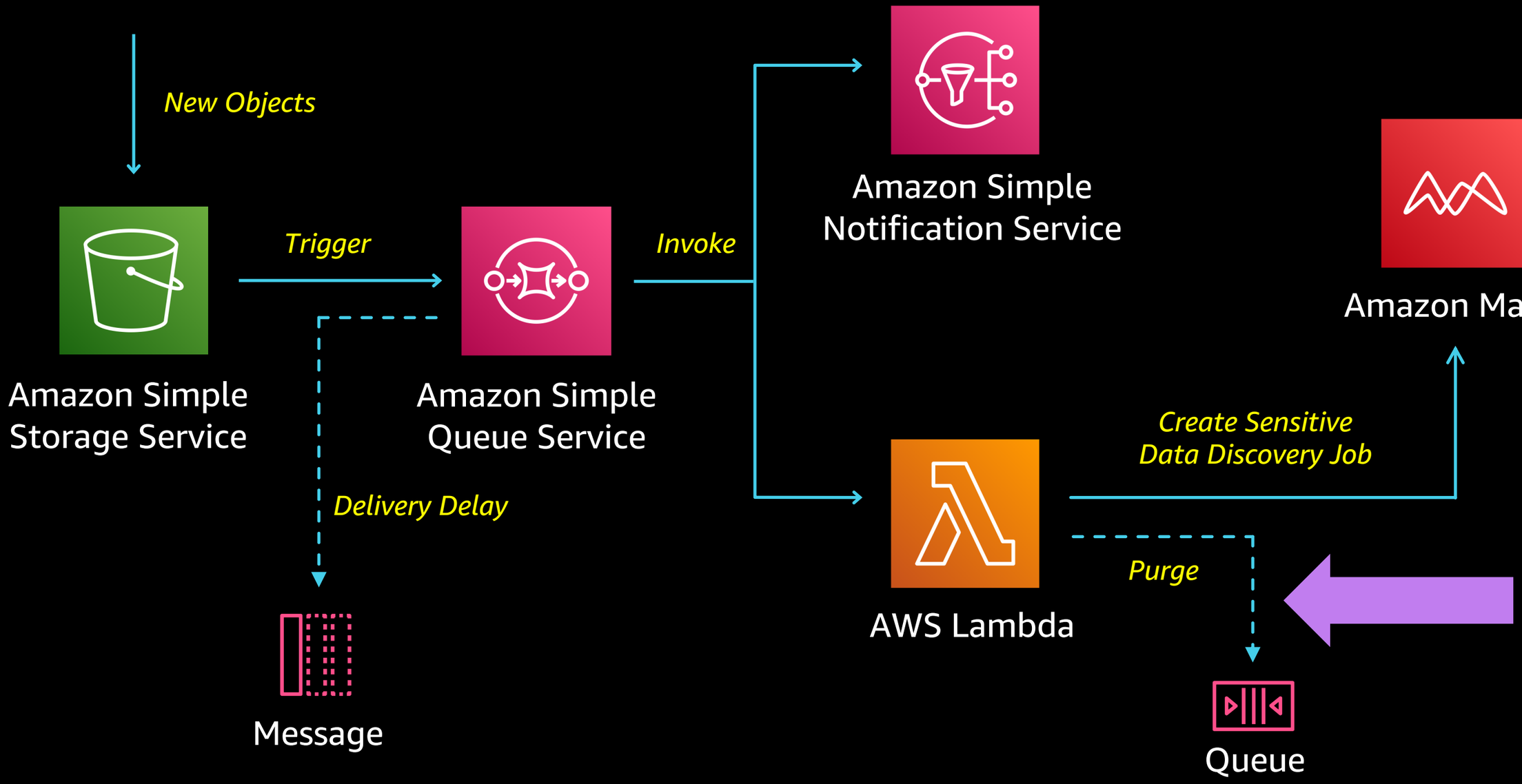
```
client2 = boto3.client('sqs', 'us-west-1')
response2 = client2.purge_queue(
    QueueUrl='[redacted]')

print(response2)
```

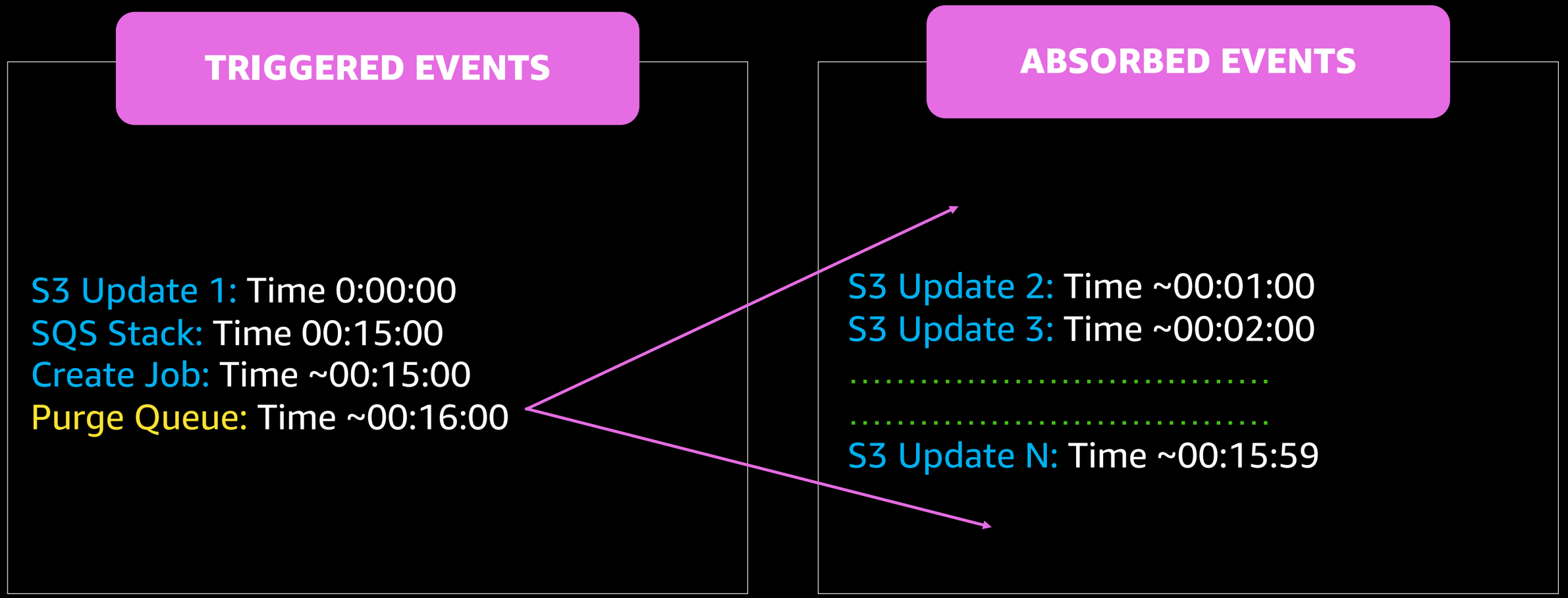


```
response1
custom
    '                    -4d52371d4cf2',
],
des      ='custom id      job',
in       =True,
job      ONE_TIME',
name='string',
s3JobDefinition={
    'bucketDefi          [
        {
            '        id': '463062299534',
            '        s': [
                     hajoshimacietestbucket-us-west-1',
            ]
        },
    ],
    'scoping
        'excl        {
            'ana' : [
                {
                                                    DATE',
},
)
print(response1)
```

Event-driven architecture



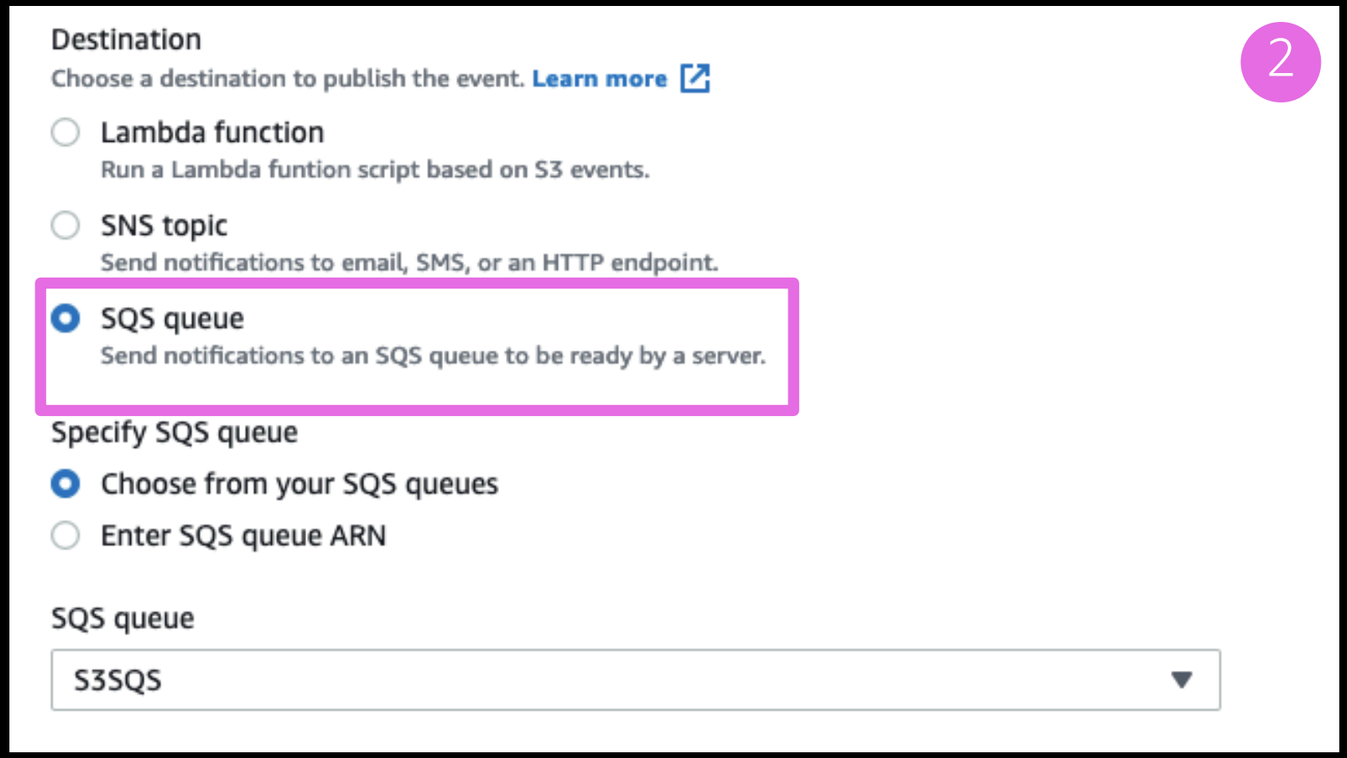
Events sequence

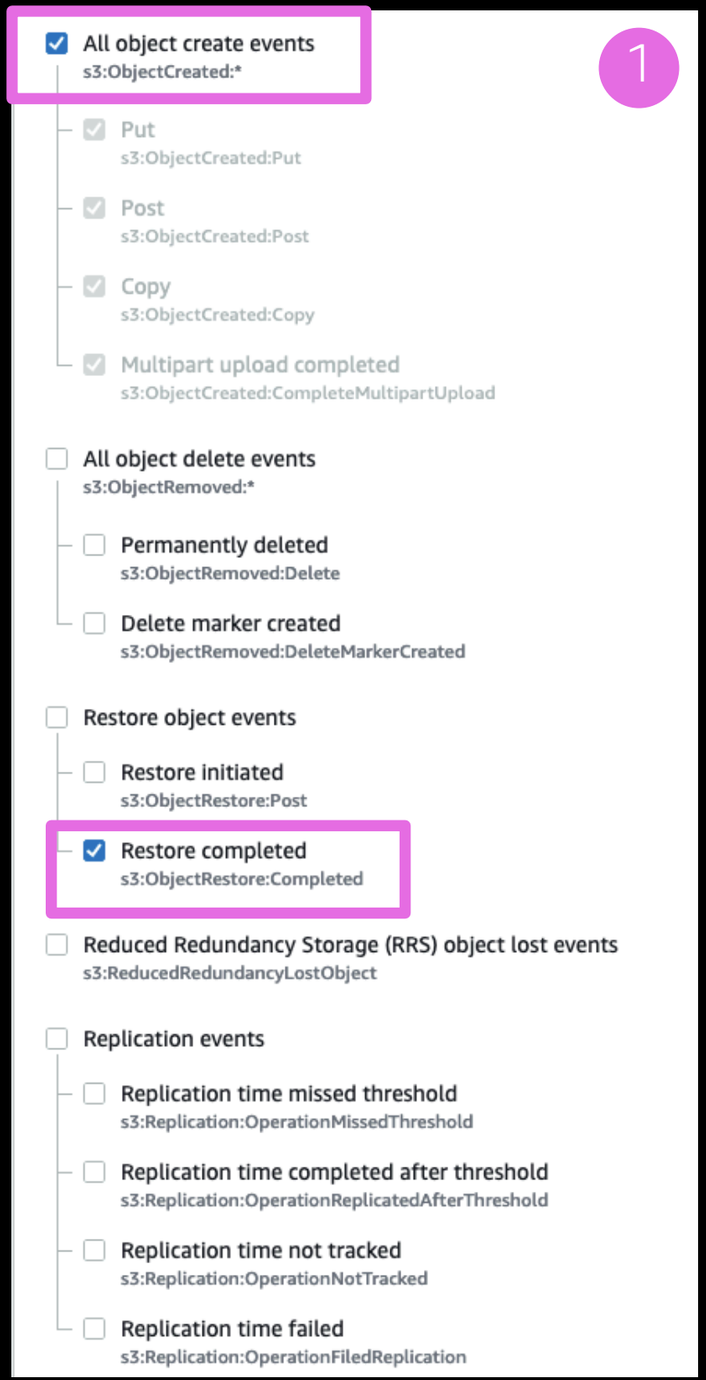


Configuration steps

S3 EVENTS:

Publish Amazon S3 events to SQS queue



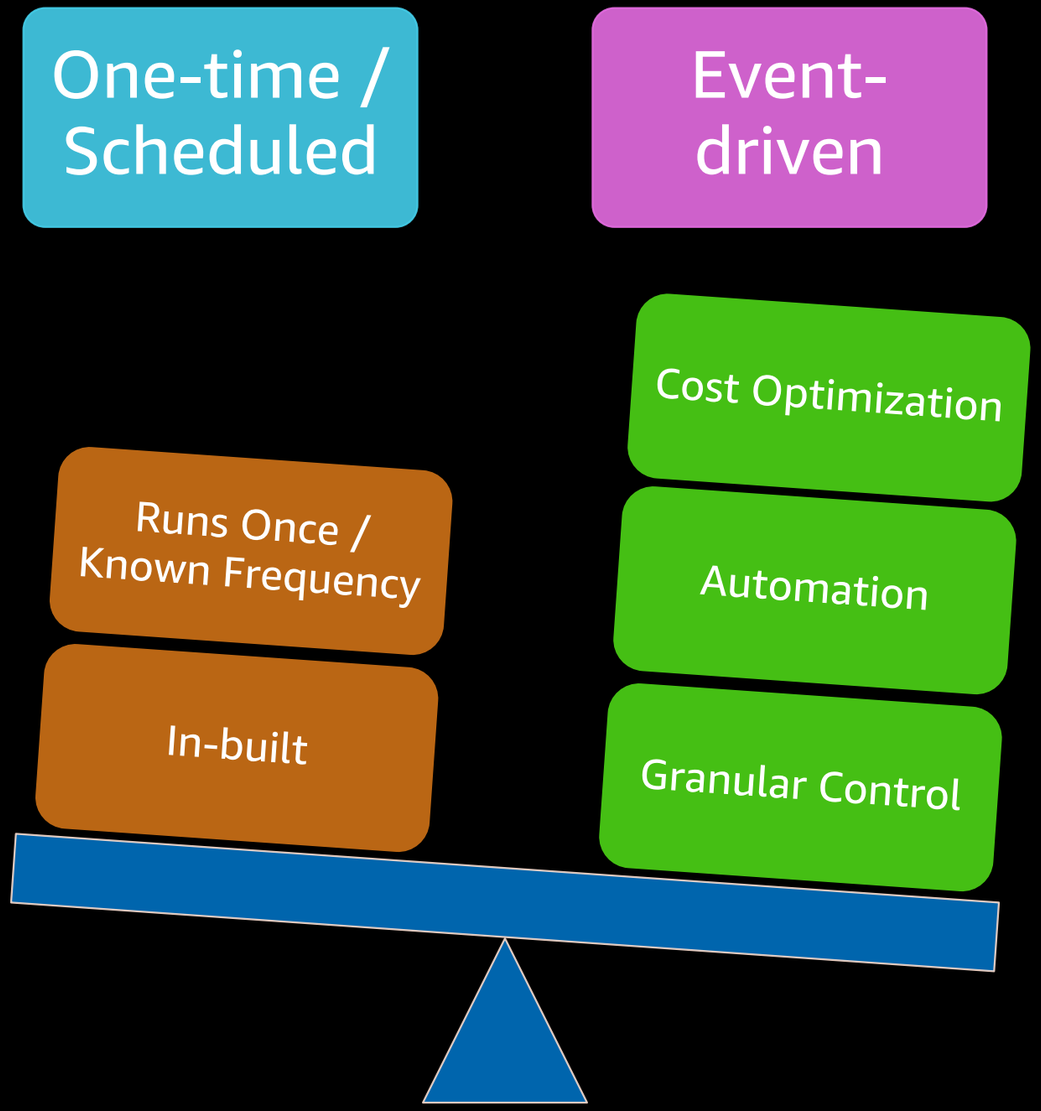


Benefits



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

One-time/Scheduled and Event-driven



Best practices



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started? Know best practices



Export detailed data discovery result

Getting started? Know best practices



Encrypted objects in S3? Understand what data Macie can and can**not** analyze

Client-side, SSE-C, SSE-S3, AWS-KMS, SSE-KMS

Getting started? Know best practices



Scope Data Discovery Jobs to align with compliance needs cost-effectively

Frequency, Include/Exclude object criteria

Getting started? Know best practices



Leverage Cost Estimation Feature

Getting started? Know best practices



Automate: Act on your Findings

Getting started? Know best practices



Meeting security requirements, assessing your data privacy and security, identifying sensitive data in data migration

Getting started? Know best practices



Monitor Sensitive Data Discovery Quota

Getting started? Know best practices



Monitor API calls using AWS CloudTrail

Key takeaways



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sensitive data discovery in a nutshell

KNOW THE PATTERNS AND ANTI-PATTERNS!

One-time	Scheduled	Event-driven
Suitable for historical data analysis	Suitable if new data is added at a fixed frequency	Suitable if you cannot predict frequency of new data
Include / Exclude objects criteria	Optionally classify new objects only	Automated approach and cost optimization
Not suitable for ongoing data analysis	Not suitable if critical data is added randomly	Not suitable if non-critical data is added too frequently

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.