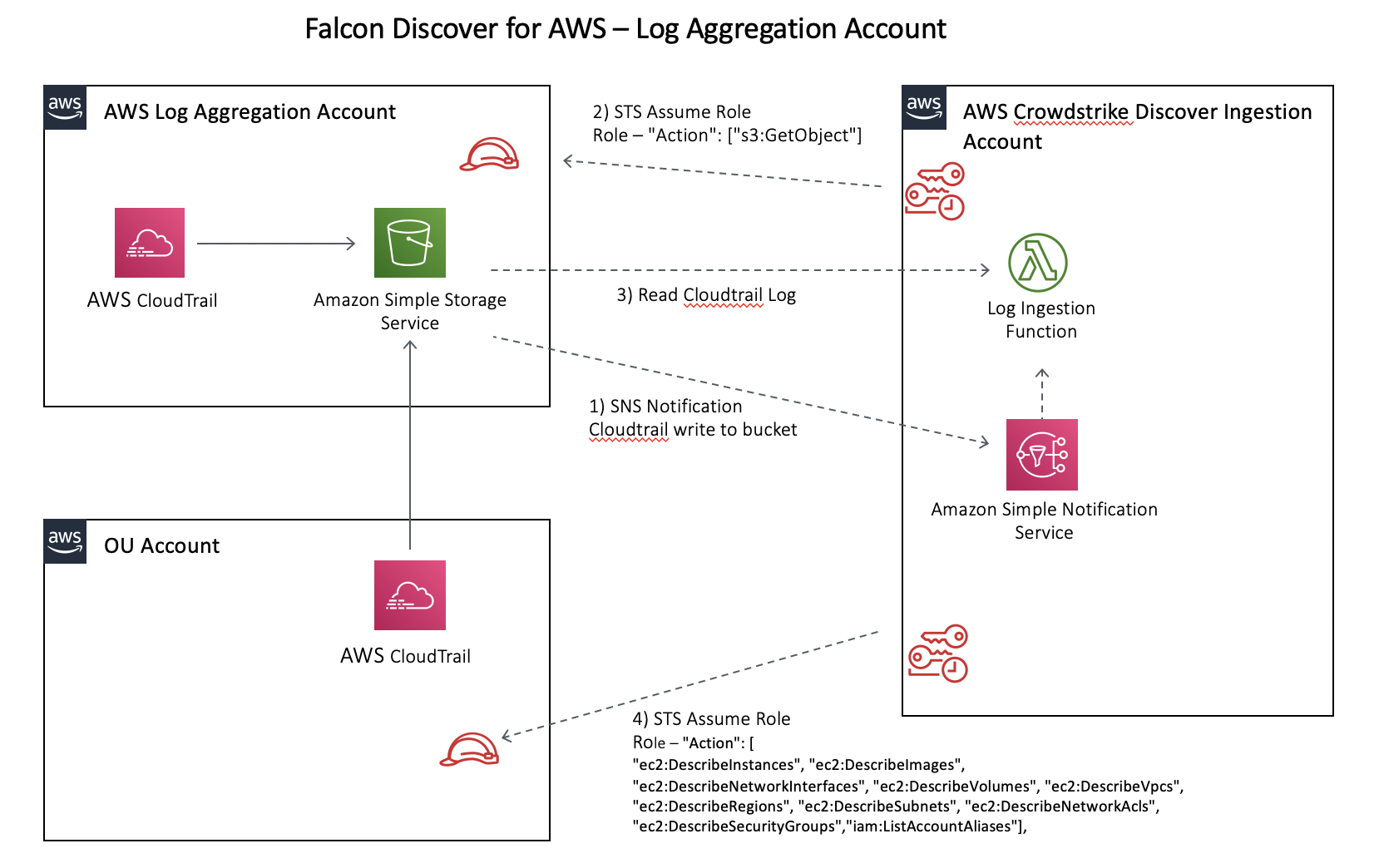
Setup Crowdstrike Discover for Cloud – Central Log Archive Manual Mode

# ***This guide describes the steps to setup Crowdstrike Discover for Cloud where you are using a central log archive account.***

# Prerequisites:

* Access to an AWS account that will be used as a log archive account.
* Administrative access that allows the creation of IAM roles, S3 Buckets, S3 bucket permissions.
* Subscription to Falcon Discover for Cloud & Containers OR the Falcon Cloud Workload Protection Bundle
* Subscription to Falcon Insight

# How Crowdstrike Discover Works



**Step 1: S3 bucket publishes and SNS notification to the Crowdstrike SNS topic**

When a new object is written to the S3 bucket the bucket will send an SNS notification to the Crowdstrike SNS topic. Crowdstrike hosts sns topics in all available US and EU regions. The topic has the format arn:aws:sns:<<region>>:292230061137:cs-cloudconnect-aws-cloudtrail.

The bucket should be configured so that it sends event notifications to a sns topic in the same region as the bucket is hosted.

The notification is passed to an SQS queue where it is read and the name of the log archive is extracted. The object name is specified in

Figure -1 Example Notification

*{*

*"Type":"Notification",*

*"MessageId":"5caad601-303a-5424-8360-81563a2dd550",*

*"TopicArn":"arn:aws:sns:us-west-1:* *106xxxxxxx53:discovernotification",*

*"Subject":"Amazon S3 Notification",*

*"Message":"{*

*\"Records\":[*

*{*

*\"eventVersion\":\"2.1\",*

*\"eventSource\":\"aws:s3\",*

*\"awsRegion\":\"us-west-1\",*

*\"eventTime\":\"2020-06-07T15:29:24.373Z\",*

*\"eventName\":\"ObjectCreated:Put\",*

*\"userIdentity\":{*

*\"principalId\":\"AWS:AROAIF4JELG3VJGB7GNKM:regionalDeliverySession\"*

*},*

*\"requestParameters\":{*

*\"sourceIPAddress\":\"34.255.117.103\"*

*},*

*\"responseElements\":{*

*\"x-amz-request-id\":\"8B31967DB0022DB8\",*

*\"x-amz-id-2\":\"BBP1gx85O\/IPBAt46K2nrDP5Se+e0rFZvoBgW\/zpVk0bmdjgN0hqdEyvukVyPKUG28Txu5SrdVmydazKoPpCP4nuXzKJv3vt\"*

*},*

*\"s3\":{*

*\"s3SchemaVersion\":\"1.0\",*

*\"configurationId\":\"YTIxZGZmMDUtYzhiZi00NmQ4LThkN2YtODE2MzBmNzAwMjhi\",*

*\"bucket\":{*

*\"name\":\"discovertrailbucket1\",*

*\"ownerIdentity\":{*

*\"principalId\":\"A3B7CH7GNBQZO9\"*

*},*

*\"arn\":\"arn:aws:s3:::discovertrailbucket1\"*

*},*

*\****"object\":{***

***\"key\":\"AWSLogs\/106xxxxxxx53\/CloudTrail\/eu-west-1\/2020\/06\/07\/106xxxxxxx53\_CloudTrail\_eu-west-1\_20200607T1525Z\_M4JIQrtGMS4b1qWd.json.gz\",***

*\"size\":3261,*

*\"eTag\":\"0e3ec1b908e5c5524cee6372b31abd0a\",*

*\"sequencer\":\"005EDD07D60CBC821F\"*

*}*

*}*

*}*

*]*

*}",*

*"Timestamp":"2020-06-07T15:29:27.734Z",*

*"SignatureVersion":"1",*

*"Signature":"hMazOWuY\/wHZFO6n6Ph9eJ6YeywGZ1SuWjuE5q7yxJf1ZiV9JOW36BfNAUu67U20RJEpKy+2SgO12\/EkEMJVqh8OIuMWtDyjDaccDUzma+uFcMK3fBvwnESakGmKh8cOTZ054VJ7LQfTf\/XL\/kDxr26lMI7xcjoPbIcrlbzEUQJ+2pehaDJEFzDkSW7nqCWnWU8voA7cQV9p3jws8+McrqWqv9vb58jVGBE7C6e6BPoVZ9+rjwQUznZS\/qZ9G4i5kMpZrJMcXymT8A8SIrtaK9AAd8VBpM7IYipE5B\/2IumqfpbfHtbbNqisIlTLFJMFsj1kC\/l+nM0Vl67y+\/fzcA==",*

*"SigningCertURL":"https:\/\/sns.us-west-1.amazonaws.com\/SimpleNotificationService-a86cb10b4e1f29c941702d737128f7b6.pem",*

*"UnsubscribeURL":"https:\/\/sns.us-west-1.amazonaws.com\/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-west-1:106xxxxxxx53:discovernotification:f30a6728-72c7-486d-9364-97d75ce32817"*

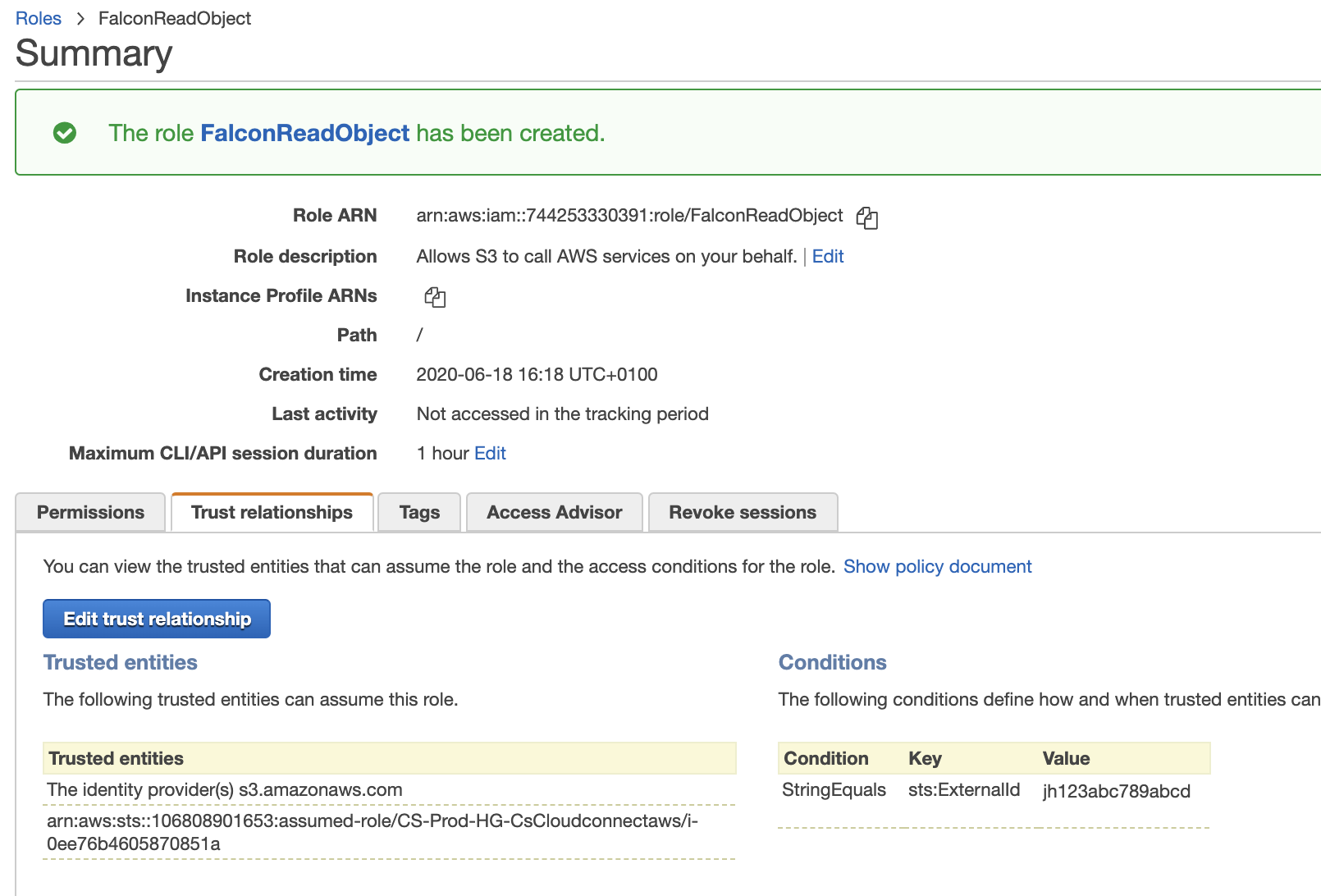
*}*

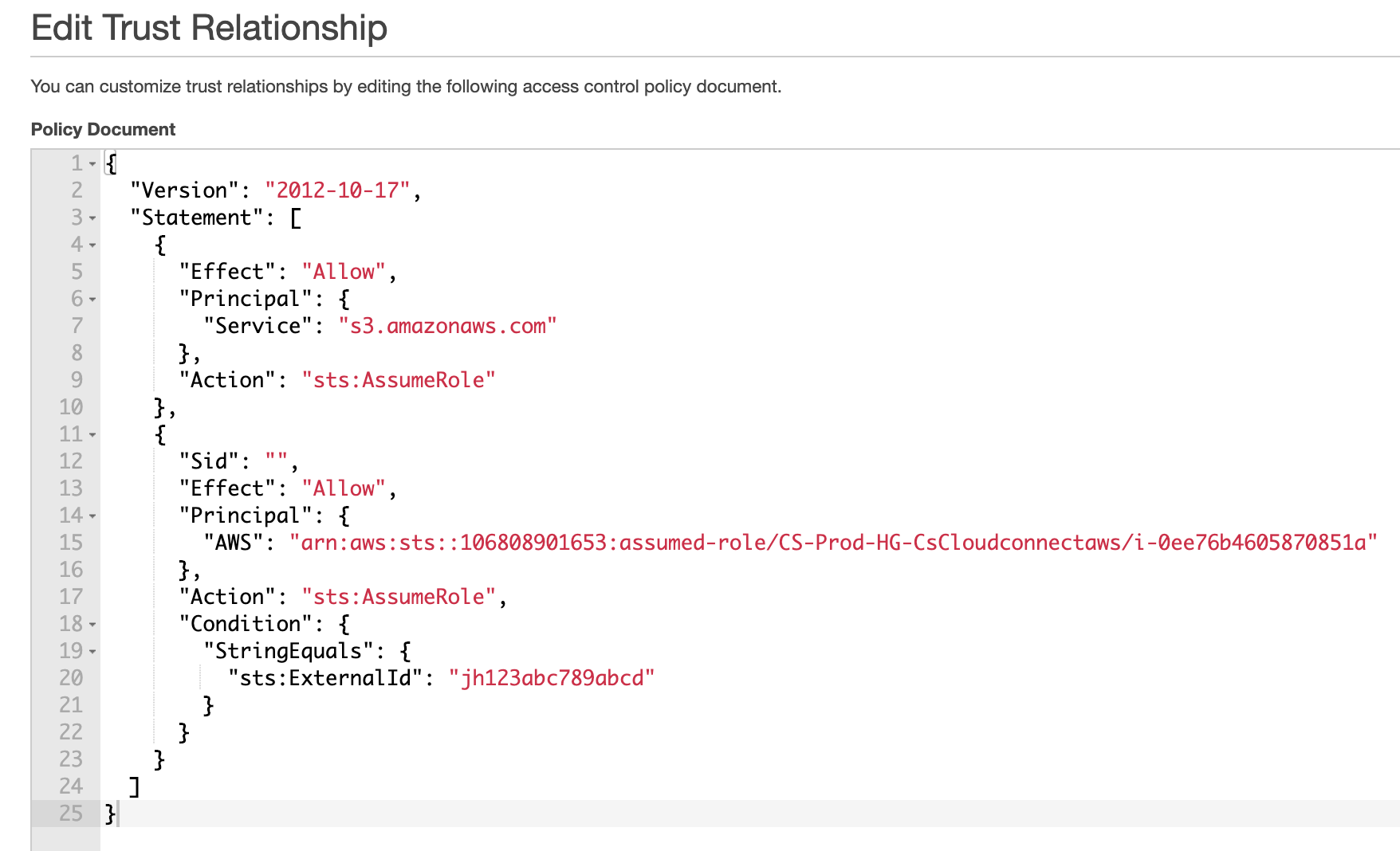
**Step 2: Crowdstrike assumes a role from the customer account that permits access.**

On receipt of the SNS notification Crowdstrike will make an api call to the customer account and will attempt to assume an IAM role to allow us to read the object from the S3 bucket. The customer should have previously granted permissions for Crowdstrike account to assume the role. The bucket policy should specify the Crowdstrike account and externalid. For more information on the authentication mechanism <https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-user_externalid.html>

Typically the role will be restricted so that only a specified Crowdstrike role in the Crowdstrike account can assume the role. The arn of the Crowdstrike role that should be granted permissions is

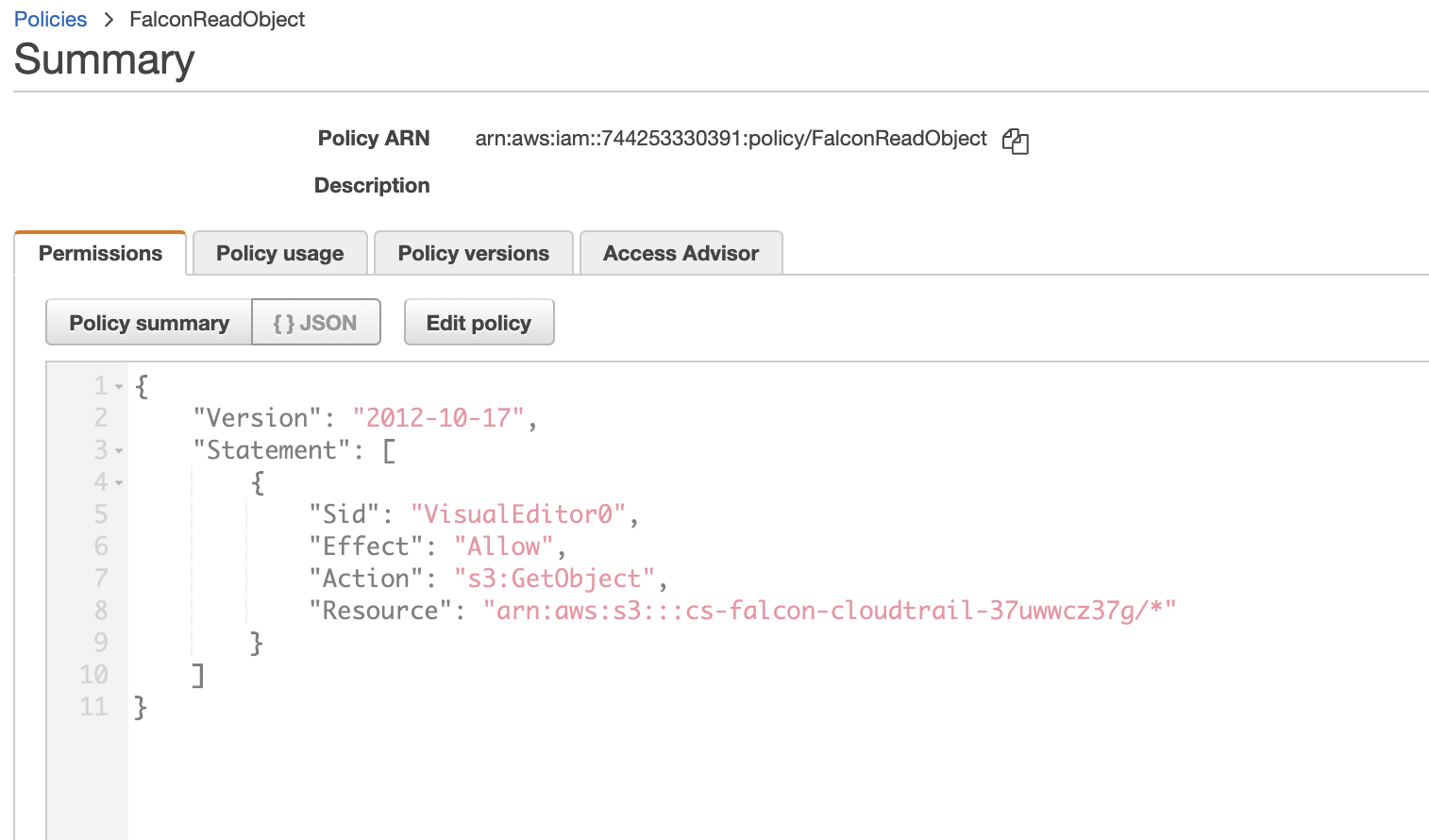
arn:aws:sts::106808901653:assumed-role/CS-Prod-HG-CsCloudconnectaws/i-0ee76b4605870851a





**Step 3: Crowdstrike reads the log file from the bucket.**

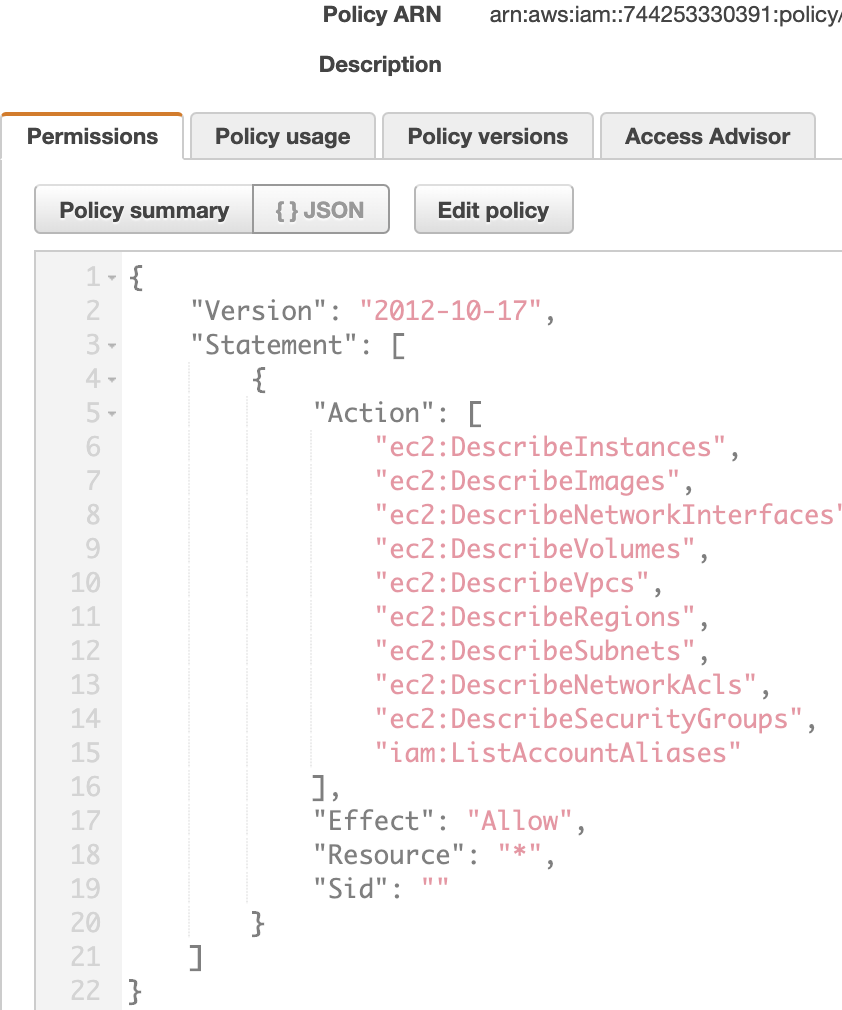
The log file is read from the S3 bucket. The role we have assumed in the customer account the should permit s3:getObject access to all objects in the bucket.



**Step 4: Crowdstrike assumes a role from the customer account that permits access.**

If an event of interest is found in the log file, Crowdstrike will attempt to assume a role in the ***account where the event has occurred*** and discover more about the AWS object that has been created.

A role must exist in the account that has the following permissions



# Setup – Manual Mode Using existing trails and S3 bucket

## Setup of the Central Logging account

If the customer has used control tower or a similar mechanism to create the central logging account and S3 bucket we will need to create the following resources in the central logging account.

* IAM role that we will assume to read the S3 logs
* S3 bucket event SNS notification
* S3 bucket policy that allows s3:getobject.

CloudFormation and terraform templates to perform these operations can be found here <https://github.com/jhseceng/discover-onboarding>.

Note: This repository is still being updated. Regular updates will be posted here.

Once these resources have been created

Send a request to POST /cloud-connect-aws/entities/accounts/v1

The following parameters are required

"cloudtrail\_bucket\_owner\_id": "<Central Log Account ID>",

"cloudtrail\_bucket\_region": "<Central Log Account S3 Bucket Region>",

"external\_id": "<Customer created unique string>",

"iam\_role\_arn": "<ARN of the IAM role created that provides s3:getobject premissions>",

"id": ""<Customer created unique string>",

Example request

curl -X POST \

 https://api.crowdstrike.com/cloud-connect-aws/entities/accounts/v1?mode=manual \

 -H 'Authorization: Bearer ' \

 -H 'Content-Type: application/json' \

 -H 'Postman-Token: 85ecdec7-0b37-4446-89a6-71c337bbac9e' \

 -H 'cache-control: no-cache' \

 -d '{

 "resources": [

   {

"cloudtrail\_bucket\_owner\_id": 111111111111

"cloudtrail\_bucket\_region": “us-east-1",

“id”: “111111111111”,

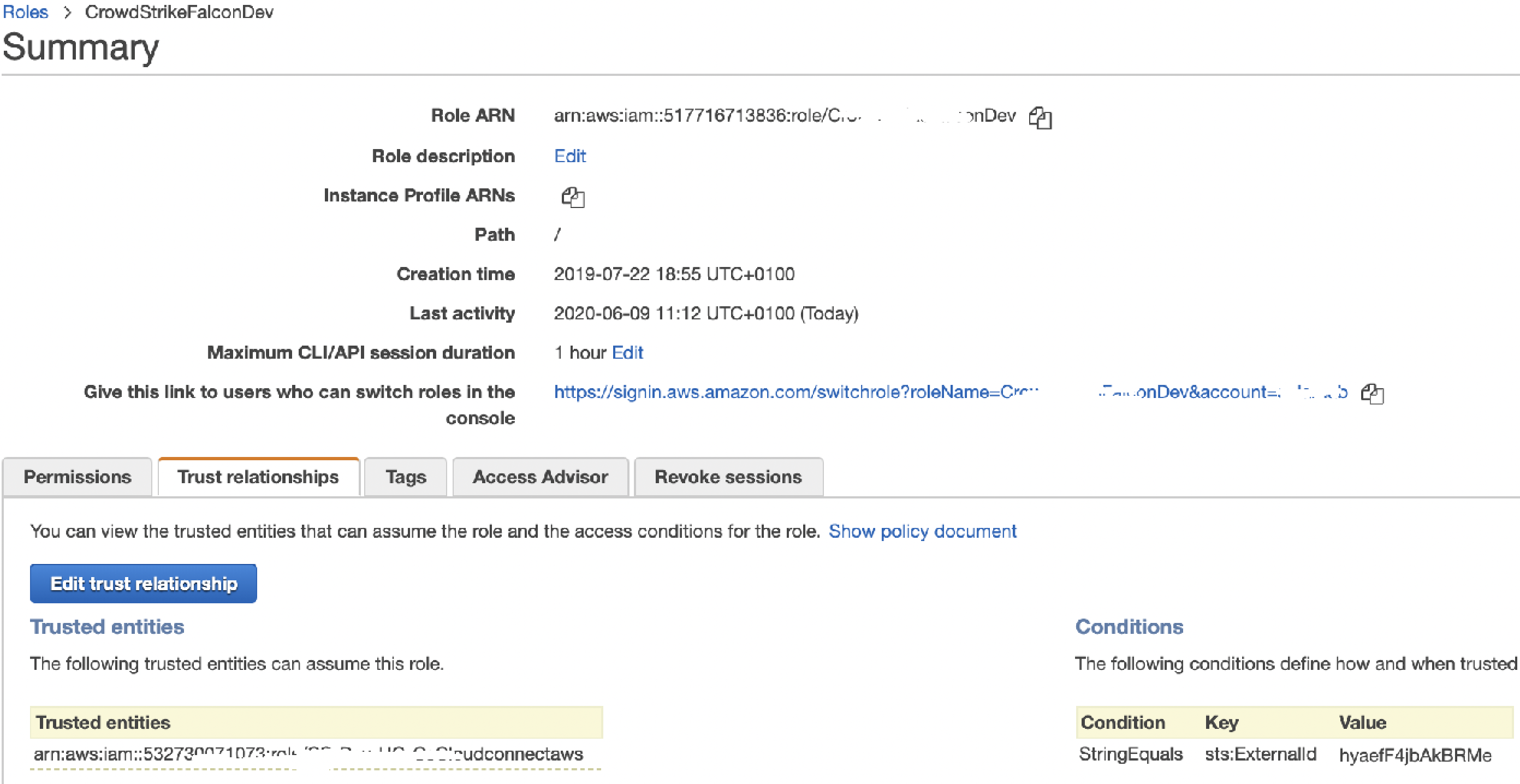
“iam\_role\_arn": "arn:aws:iam::111111111111:role/CrowdStrikeFalcon"

“external\_id”: “sdfdsfl234xx3”

   },

]

Note: The external\_id must match the external ID that the customer has configured in the IAM role that has been setup for S3 access. The external ID can be verified in the AWS console.



## Setup Additional accounts

Additional accounts will require both a cloudtrail configured to forward to the s3 bucket in the log archive account and an IAM role that allows Crowdstrike to gather information about the resources.

Note : If the customer has used control tower or a similar mechanism to create accounts it is likely that cloudtrail has already been configured and we then only need to create the IAM role.

Example IAM policy that should be associated with the role.

"iamPolicyDescribeAccess": {  
 "Type": "AWS::IAM::Policy",  
 "Properties": {  
 "PolicyName": "DescribeAPICalls",  
 "PolicyDocument": {  
 "Statement": [  
 {  
 "Action": [  
 "ec2:DescribeInstances",  
 "ec2:DescribeImages",  
 "ec2:DescribeNetworkInterfaces",  
 "ec2:DescribeVolumes",  
 "ec2:DescribeVpcs",  
 "ec2:DescribeRegions",  
 "ec2:DescribeSubnets",  
 "ec2:DescribeNetworkAcls",  
 "ec2:DescribeSecurityGroups",  
 "iam:ListAccountAliases"  
 ],  
 "Effect": "Allow",  
 "Resource": "\*",  
 "Sid": ""  
 }  
 ],  
 "Version": "2012-10-17"  
 },  
 "Roles": [ {  
 "Ref": "iamRole"  
 }]  
 },  
 "DependsOn": "iamRole"  
},

The IAM Role can be created from the resources here <https://github.com/jhseceng/discover-onboarding/tree/master/templates/terraform/additional-account-existing-trail>

Note: It is important to reference the S3 Bucket in the central logging account, both the cloudtrail\_bucket\_owner\_id andcloudtrail\_bucket\_owner\_id and the local account external\_id and iam\_role\_arn must also be speficied.

Once the account, trail and IAM role have been created the account can be registered with Crowdstrike.

Send a request to POST /cloud-connect-aws/entities/accounts/v1

The following parameters are required

"cloudtrail\_bucket\_owner\_id": "<**Central Log Account ID**>",

" cloudtrail\_bucket\_owner\_id ": "<**Central Log Account S3 Bucket Region**>",

"external\_id": "<**Customer created unique string in the LOCAL account>",**

"iam\_role\_arn": "<**ARN of the IAM role in the LOCAL account**>",

"id": ""<Customer created unique string>",

Example request

curl -X POST https://api.crowdstrike.com/cloud-connect-aws/entities/accounts/v1?mode=manual \

-H 'Authorization: Bearer '

-H 'Content-Type: application/json'

-H 'Postman-Token: 85ecdec7-0b37-4446-89a6-71c337bbac9e'

-H 'cache-control: no-cache' \

 -d '{

 "resources": [

   {

"cloudtrail\_bucket\_owner\_id": 111111111111

"cloudtrail\_bucket\_region": “us-east-1",

“id”: “222222222222”,

“iam\_role\_arn": "arn:aws:iam:: 222222222222:role/CrowdStrikeFalcon"

“external\_id”: “abcdefghij”

   },

]

## Updating the Crowdstrike API

It is possible to update the “external\_id” or other attribute via a “PATCH” request.

An example request to update the “external\_id” in an account

curl -X PATCH https://api.crowdstrike.com/cloud-connect-aws/entities/accounts/v1?mode=manual \

-H 'Authorization: Bearer '

-H 'Content-Type: application/json'

-H 'Postman-Token: 85ecdec7-0b37-4446-89a6-71c337bbac9e'

-H 'cache-control: no-cache' \

 -d '{

 "resources": [

   {

"cloudtrail\_bucket\_owner\_id": 111111111111

"cloudtrail\_bucket\_region": “us-east-1",

“id”: “222222222222”,

“iam\_role\_arn": "arn:aws:iam:: 222222222222:role/CrowdStrikeFalcon"

“external\_id”: “abcdefghij”

   },

]

# Troubleshooting

## Check the s3 Bucket Event Policy

Verify that the bucket is setup for notifications

*% aws s3api get-bucket-notification-configuration --bucket cs-falcon-cloudtrail-xxxxxxxx*

*{*

*"TopicConfigurations": [*

*{*

*"Id": "xxxxxxxxxxx",*

*"TopicArn": "arn:aws:sns:eu-west-1:292230061137:cs-cloudconnect-aws-cloudtrail",*

*"Events": [*

*"s3:ObjectCreated:Put"*

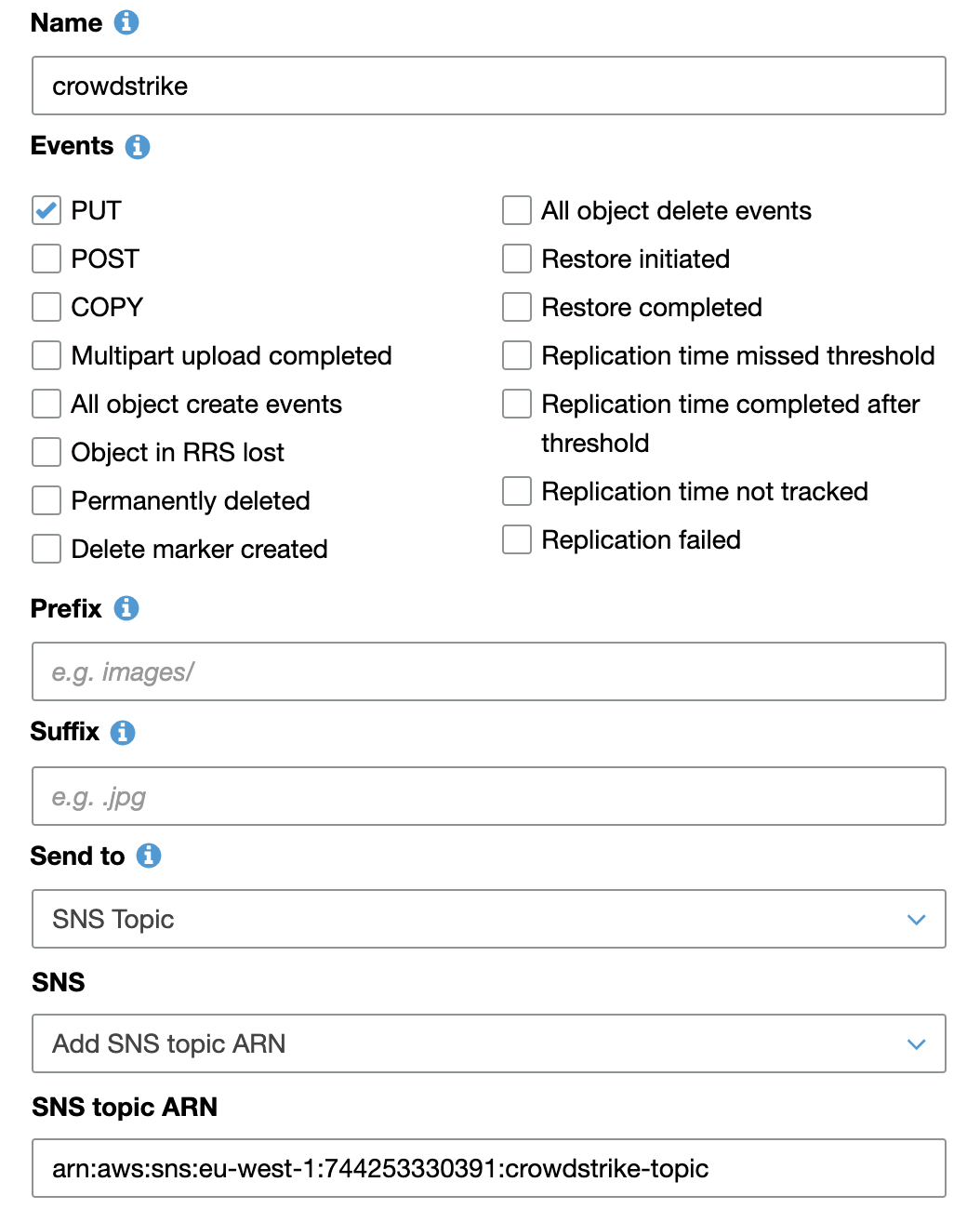
*]*

*}*

*]*

*}*

Via the console *Bucket –> Properties -> Events*



## Check the s3 Bucket Policy

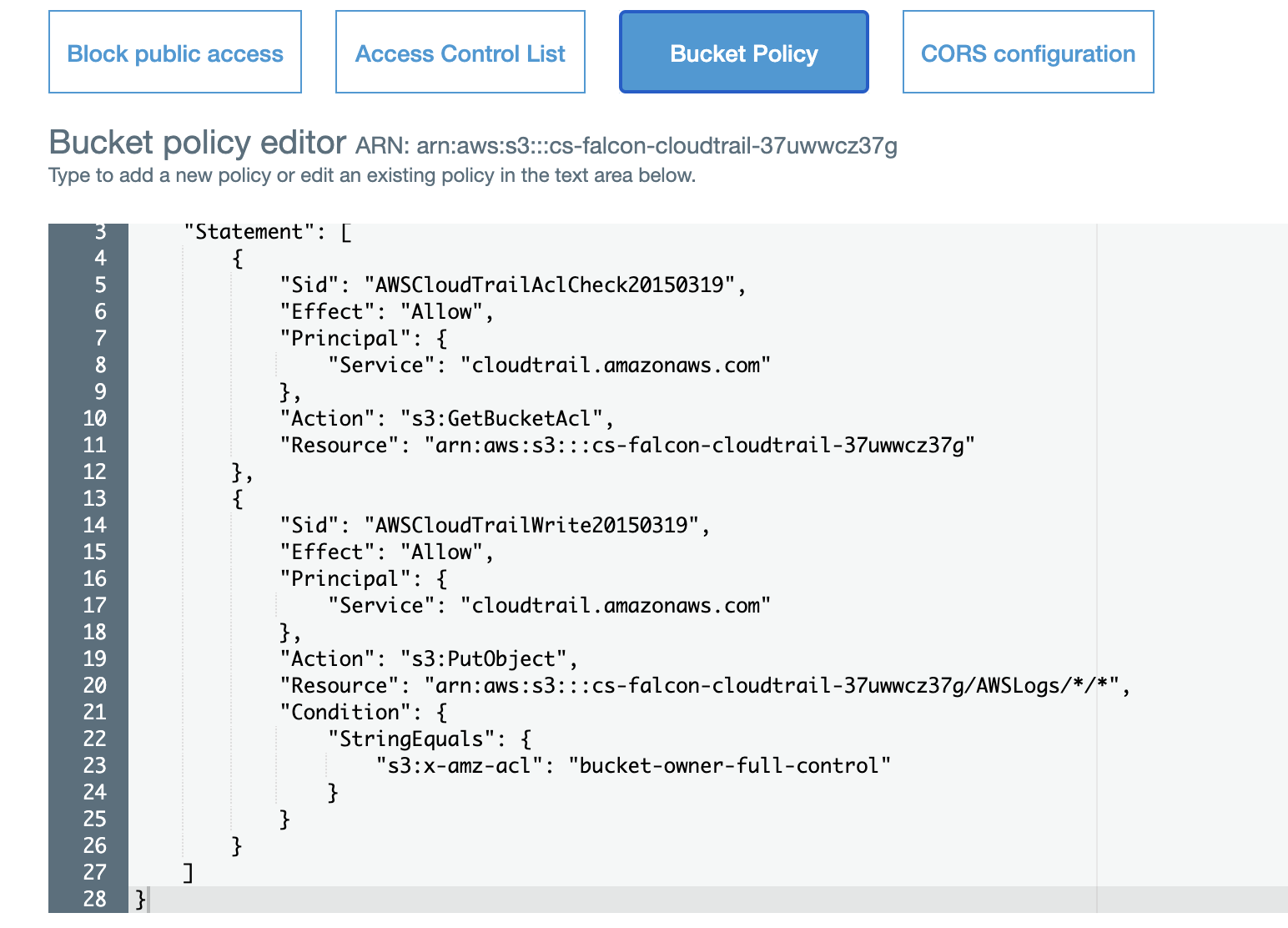
By default, an S3 object is owned by the AWS account that uploaded it. This is true even when the bucket is owned by another account. To get access to the object, the object owner must explicitly grant you (the bucket owner) access.

<https://aws.amazon.com/premiumsupport/knowledge-center/s3-bucket-owner-access/>

The object owner can grant the bucket owner full control of the object by updating the access control list (ACL) of the object. The object owner can update the ACL either during a put or copy operation, or after the object is added to the bucket. In the case of cloudtrail logs add a bucket policy that grants users access to put objects in your bucket only when they grant the bucket owner full control of the object.

<https://aws.amazon.com/premiumsupport/knowledge-center/s3-require-object-ownership/>

The bucket should have an ACL policy associated with it



## Check the s3 Bucket ACL

From the aws cli tool

aws s3api get-bucket-acl --bucket <bucket-name>

{

"Owner": {

"DisplayName": "bd\_integ",

"ID": "7d137abc41d6d3455663518126c28df027f25c1709f269239010f71bff0e9839"

},

"Grants": [

{

"Grantee": {

"Type": "CanonicalUser",

"DisplayName": "bd\_integ",

"ID": "7d137abc41d6d3455663518126c28df027f25c1709f269239010f71bff0e9839"

},

"Permission": "FULL\_CONTROL"

}

]

}

From the output the bucket *“Owner”* is *“bd\_integ”* and has *“FULL\_CONTROL”* over the bucket

## Check s3 Object Permissions

From the aws cli tool

aws s3api get-object-acl --bucket <bucket-name> --key <log-file-key>

*{*

*"Owner": {*

*"DisplayName": "aws\_cloudtrail\_us-east-1",*

*"ID": "65e03a8a45a3aaaa881ada217702e0ad77152fe0196a8a14c44aa3bc4f11d2ae"*

*},*

*"Grants": [*

*{*

*"Grantee": {*

*"Type": "CanonicalUser",*

*"DisplayName": "aws\_cloudtrail\_us-east-1",*

*"ID": "65e03a8a45a3aaaa881ada217702e0ad77152fe0196a8a14c44aa3bc4f11d2ae"*

*},*

*"Permission": "FULL\_CONTROL"*

*},*

*{*

*"Grantee": {*

*"Type": "CanonicalUser",*

*"DisplayName": "bd\_integ",*

*"ID": "7d137abc41d6d3455663518126c28df027f25c1709f269239010f71bff0e9839"*

*},*

*"Permission": "FULL\_CONTROL"*

*}*

*]*

*}*

The log “Owner” is “*aws\_cloudtrail\_us-east-1” and the “Grantee” “bd\_integ”* who has ownership of the bucket has been granted *“FULL\_CONTROL*