



<https://www.defense.gov/Explore/News/Article/Article/1254719/project-maven-to-deploy-computer-algorithms-to-war-zone-by-years-end/>

# Keep up with AWS

In a security context

We're going to talk about how AWS IAM sucks  
How we can make dealing with it less stressful



About me

## Co-conspirators



**Alana Kirby**



**Aidan Steele**



**Rupert Bryant-Greene**

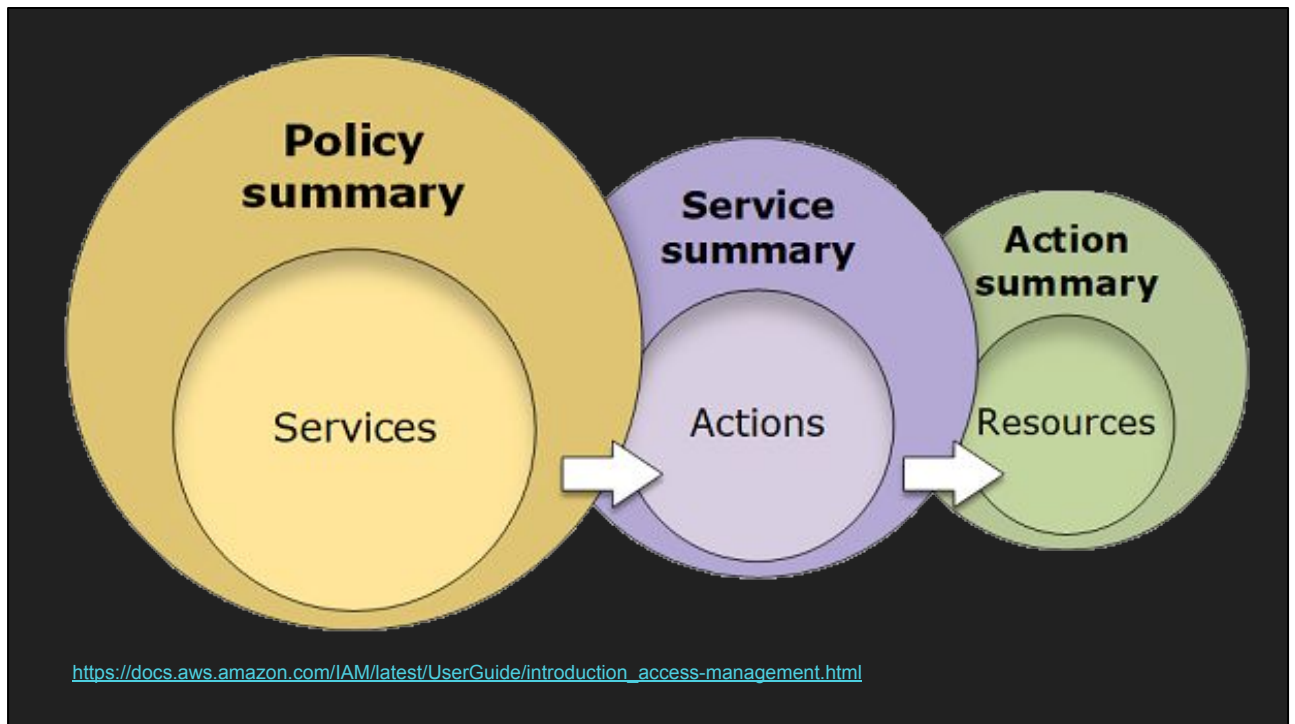
<https://github.com/alanakirby/aktion>

Original art by Alana, now there's four of us.



[https://commons.wikimedia.org/wiki/File:Amazon\\_Basics\\_battery\\_Papenburg\\_\(2019\).jpg](https://commons.wikimedia.org/wiki/File:Amazon_Basics_battery_Papenburg_(2019).jpg)

Let's start with some AWS basics  
IAM is ***the*** core AWS service  
It controls everything



AWS access controls are broken down into:  
Principal, Resource, Action  
Every time you do something it involves one of each of these





# Alice may eat apples



<https://blog.schlomo.schapiro.org/2017/06/understanding-iam-roles-in-amazon-aws.html>

Permissions are calculated by checking if the principal can perform an action on a resource.

When you S3:ListBuckets, **you** are **listing** the **buckets**.

There are two places where policies are applied



# Alice may eat apples



<https://blog.schlomo.schapiro.org/2017/06/understanding-iam-roles-in-amazon-aws.html>

Both Principals and Resources can have policies applied to them to control access. Sometimes there are Roles, service control policies etc. but these essentially extend either the Principal or Resource's policies.

"Olly, you're over simplifying things" - IAM nerds in the room

As Dave says...



“[the] mental gymnastics  
of post-modern IAM  
policy theory”

It gets more complex

I'm not going to do this with you today,  
but if I don't mention it other IAM nerds will at me



If you hang out with IAM nerds, you'll have seen this referenced  
We don't have time to go down that rabbit hole



Permissions boundaries are also just more policies  
Organisation Service Control Policies and Session Policy also  
Caveat that you might have permission even if not all of these are set



# Blacklisting

# Whitelisting

Whitelists vs Blacklists

Whitelists are great but often impractical

Blacklists aren't strict enough

Everyone sits somewhere in the middle



**Aidan W Steele**

@\_\_steele



If you grant `s3:PutObject*` in an IAM policy, you're likely unintentionally granting `s3:PutObjectLockConfiguration` - a bucket-level API.

7:46 PM · Jan 5, 2020 · [Twitter Web App](#)

Whitelisting is the ideal scenario for Security people who don't care or want to get work done, get frustrated and cut corners  
Most people use it because it includes `PutObject` and `PutObjectTag`  
Who knows how many `PutObject*` actions there are?

s3:PutObject

s3:PutObjectAcl

s3:PutObjectLegalHold

s3:PutObjectRetention

s3:PutObjectTagging

s3:PutObjectVersionAcl

s3:PutObjectVersionTagging

s3:PutObjectLockConfiguration

There are 8!

What is a LegalHold?

Prevents object from being deleted or overridden

Don't worry s3:BypassGovernanceRetention exists



It can't be changing that fast...



	31 Oct 2019	31 Dec 2019	20 Feb 2020
Services	191	216 ▲ 25	217 ▲ 1
Actions	6858	7801 ▲ 943	7884 ▲ 83
Managed Policies	566	621 ▲ 55	631 ▲ 10



re:Invent

Granted, it was re:invent season

Do you know for sure, which of the  
217 Services,  
7801 actions and  
631 managed policies  
your org uses and how that compares to the access granted?

I want to show you something that can help, but first an example

```

"PolicyId": "ANPAIJLU43R6AGRBK76DM",
"PolicyName": "AWSMobileHub_FullAccess",
"PolicyVersion": {
  "CreateDate": "2018-02-05T23:44:29Z",
  "CreateDate": "2019-12-19T23:15:52Z",
  "Document": {
    "Statement": [
      {
        "Action": [
          "apigateway:GET",
          "apigateway:GetResources",
          "apigateway:GetRestApis",
          "apigateway:POST",
          "apigateway:TestInvokeMethod",
          "cloudfront:GetDistribution",
          "devicefarm:CreateProject",
          "devicefarm:GetProject",

```

26 +37,7 @@

```

"lex:GetIntents",
"lex:GetSlotType",
"lex:GetSlotTypes",
"mobilehub:CreateProject",
"mobilehub>DeleteProject",
"mobilehub>DeleteProjectSnapshot",
"mobilehub:DescribeBundle",
"mobilehub:ExportBundle",
"mobilehub:ExportProject",
"mobilehub:GenerateProjectParameters",
"mobilehub:GetProject",
"mobilehub:GetProjectSnapshot",
"mobilehub:ImportProject",
"mobilehub:ListAvailableConnectors",
"mobilehub:ListAvailableFeatures",
"mobilehub:ListAvailableRegions",
"mobilehub:ListBundles",
"mobilehub:ListProjectSnapshots",
"mobilehub:ListProjects",
"mobilehub:SynchronizeProject",
"mobilehub:UpdateProject",
"mobilehub:ValidateProject",
"mobilehub:VerifyServiceRole",
"mobilehub:*",
"sns:ListTopics"

```

This is an AWS Managed Policy  
That they can't even be bothered keeping up to date

```
"mobilehub:ValidateProject",  
"mobilehub:VerifyServiceRole",  
"mobilehub:*",  
"sns:ListTopics"
```

Policy version numbers can't be used to control which version of an AWS managed policy is being used, only by AWS to roll back just like you can yourself. The only thing it allows you to do is (manually) track when AWS changes the policy.

# Demo Time

[see backup slides later]



Freudian Llama

📅 January 25, 2020 - February 20, 2020 ▾

Search

Advanced

Today

Yesterday

Last 7 Days

Last 30 Days

This Month

Last Month

Custom Range

&lt;

Jan 2020

Su

Mo

Tu

We

Th

Fr

Sa

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

Feb 2020

Su

Mo

Tu

We

Th

Fr

Sa

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

1

2

3

4

5

6

7

01/25/2020 - 02/20/2020

Cancel

Apply

<https://action.io/>

Select date range

IAM Actions Tracker

GitHub

Fork me on GitHub

February 14, 2020 - February 20, 2020

Search

Advanced

### Policies changed

Click an item to see the diff

13	AWSIoTDeviceTesterForFreeRTOSFullAccess	...
11	AmazonChimeUserManagement	...
12	CloudFrontFullAccess	...
12	CloudFrontReadOnlyAccess	...

### Services changed

Click an item to see the diff

7	chime	...
3	cloud9	...
2	iot-device-tester	...
2	rekognition	...
1	securityhub	...
3	shield	...

Shows you what's changed



## Services changed

Click an item to see the diff

7

chime

...

3

cloud9

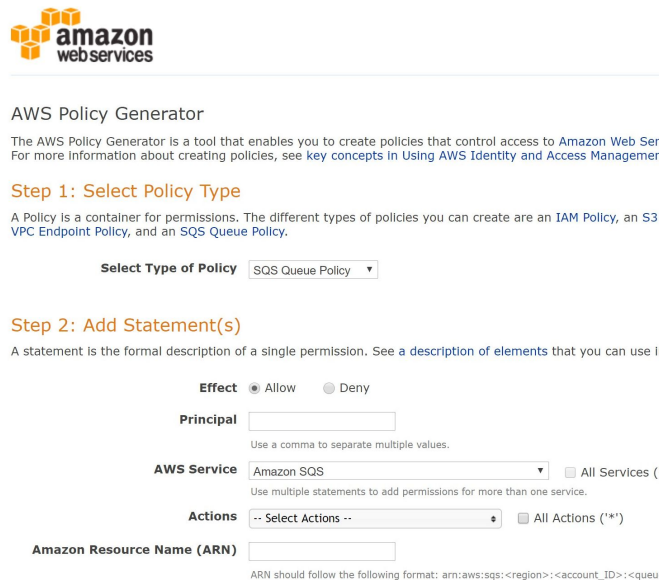
...

cloud9:ListTagsForResource

cloud9:TagResource

cloud9:UntagResource

Tada! Cloud9 added Tagging support



The screenshot shows the AWS Policy Generator web interface. At the top is the Amazon Web Services logo. Below it, the title "AWS Policy Generator" is displayed. A brief description states: "The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#)." The interface is divided into two main steps.   
**Step 1: Select Policy Type**  
This section explains that a policy is a container for permissions and lists three types: IAM Policy, S3 VPC Endpoint Policy, and SQS Queue Policy. A dropdown menu labeled "Select Type of Policy" currently shows "SQS Queue Policy".   
**Step 2: Add Statement(s)**  
This section explains that a statement is a formal description of a single permission and refers to a [description of elements](#). It contains several input fields:   
- **Effect**: Radio buttons for "Allow" (selected) and "Deny".   
- **Principal**: A text input field with a note "Use a comma to separate multiple values."   
- **AWS Service**: A dropdown menu showing "Amazon SQS" and a checkbox for "All Services (\*)". A note below says "Use multiple statements to add permissions for more than one service."   
- **Actions**: A dropdown menu showing "-- Select Actions --" and a checkbox for "All Actions (\*)".   
- **Amazon Resource Name (ARN)**: A text input field with a note: "ARN should follow the following format: arn:aws:sqs:<region>:<account\_ID>:<queue\_name>".

<https://awspolicygen.s3.amazonaws.com/policygen.html>

## How it works

- Cron
- Scrapy Scrape Policy Generator and SDKs
- Commit
- Caching stuffs
- Static + serverless
- Netlify??



## Recent Announcements APP 10:02 AM

### AWS CodeBuild Adds Support for Amazon EFS

You can now use [Amazon Elastic Filesystem \(EFS\)](#) in [AWS CodeBuild](#) build jobs. This can be achieved by specifying the EFS file system Id in your CodeBuild [Project](#). (edited)





## GitHub APP 10:05 AM

 **github-actions[bot]**


**1 new commit** pushed to master

2ff715d7 - New actions

 glassechidna/trackiam

 **github-pages[bot]**


Successfully deployed 2ff715d to github-pages

 glassechidna/trackiam

We're pretty good at keeping up

2,6 +42,7 @@

```
{
  "Arn": "arn:aws:iam::aws:policy/AWSCodeBuildAdminAccess",
  "CreateDate": "2016-12-01T19:04:44Z",
  "DefaultVersionId": "v8",
  "DefaultVersionId": "v9",
  "IsAttachable": true,
  "Path": "/",
  "PolicyId": "ANPAJQJG10IE3CD2TQXDS",
  "PolicyName": "AWSCodeBuildAdminAccess",
  "PolicyVersion": {
    "CreateDate": "2019-11-05T22:12:30Z",
    "CreateDate": "2020-02-06T20:26:30Z",
    "Document": {
      "Statement": [
        {
          "Action": [
            "ec2:DescribeVpcs",
            "ecr:DescribeRepositories",
            "ecr:ListImages",
            "elasticfilesystem:DescribeFileSystems",
            "events:DeleteRule",
            "events:DescribeRule",
            "events:DisableRule",
```

 Aidan bot  
authored 2/7/2020 @ 10:05 AM

1 modified

↑ 2

Path

polices/AWSCodeBuildAdminAccess

Do you check all the managed policies you give your developers access to?

Check out Aktion.io or follow @\_\_steele on Twitter



**Aidan W Steele** @\_\_steele · 52m

Looks like ElastiCache is coming to an outpost near you

```
■ policies/ElastiCacheServiceRolePolicy.json
.. @@ -1,13 +1,13 @@
1 {
2   "Arn": "arn:aws:iam::aws:policy/aws-service-role/E
3   "CreateDate": "2017-12-07T17:50:04Z",
4   "DefaultVersionId": "v2",
5   "DefaultVersionId": "v3",
6   "IsAttachable": true,
7   "Path": "/aws-service-role/",
8   "PolicyId": "ANPAIML5LIBUZBVCSF7PI",
```

Sometimes we get the scoop on things before they're official  
And by "we" I mean Aidan

# Where to next?

TLS transparency logs  
Track what actions go in CloudTrail  
RSS / WebHooks / Twitterbot / Slackbot  
Linking to relevant material  
Classifying changes  
See GitHub for details

Mainly making consumption easier

Please check out GitHub / Contribute

<https://aktion.io/>

- There is so, so much to keep up with
- Even AWS can't keep up
- You're not alone

<https://ewert.co.nz/slides>

@OllyTheNinja

If you are scared you should be, but know that everyone else is too  
Because it's hard and even AWS has trouble staying up to date  
One more tool in your belt  
Enjoyed being here, great community, please don't be a stranger in person or online