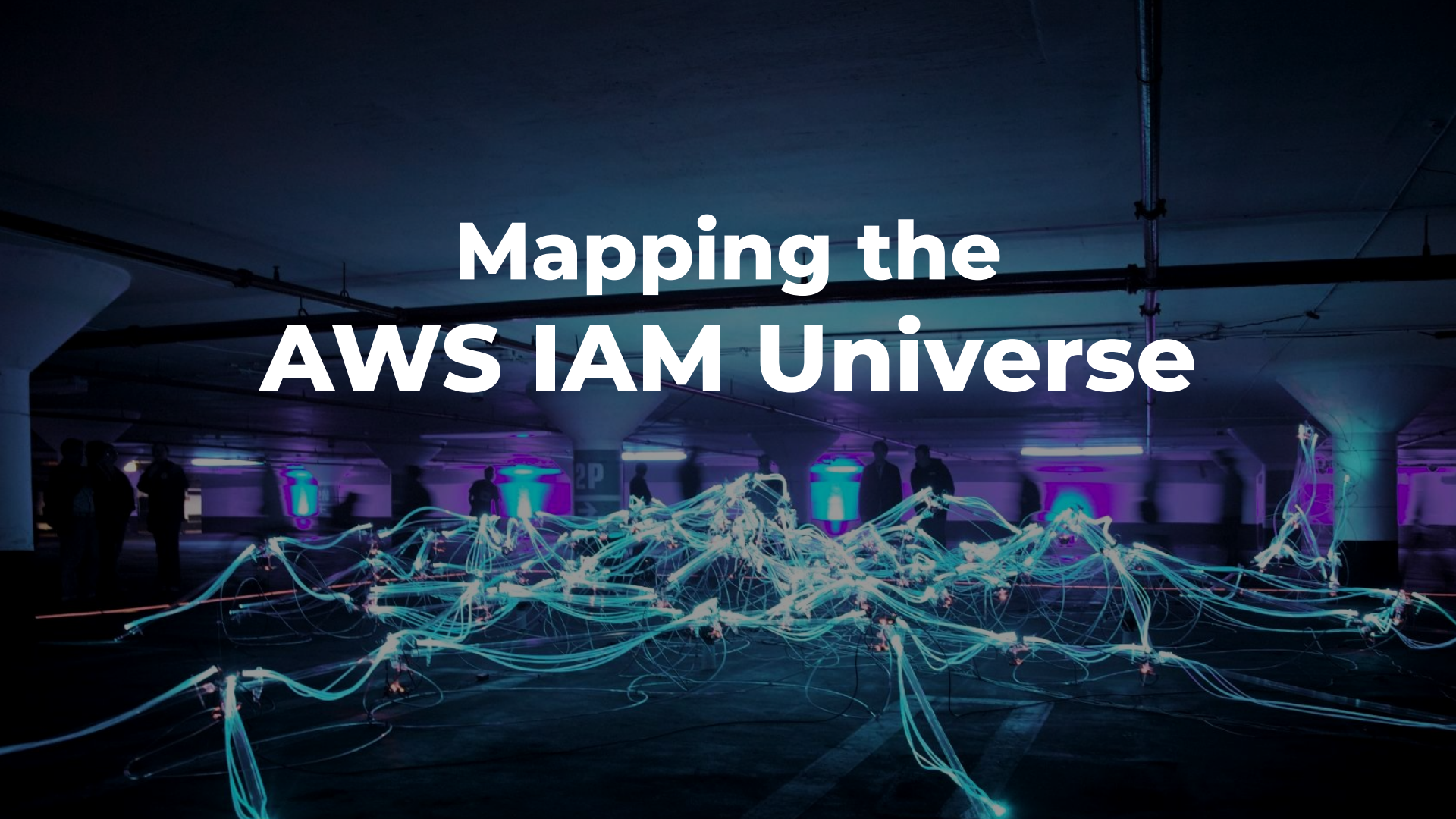
Mapping the AWS IAM Universe



Hello

Ian Mckay

Cloud Lead at Kablamo
AWS Community Hero and APN Ambassador

@iann0036

Agenda

Quick IAM Refresher

Least Privilege

Problems with the SAR

Fixing the problems

What’s Next

Quick IAM Refresher

Ian Mckay
@iann0036

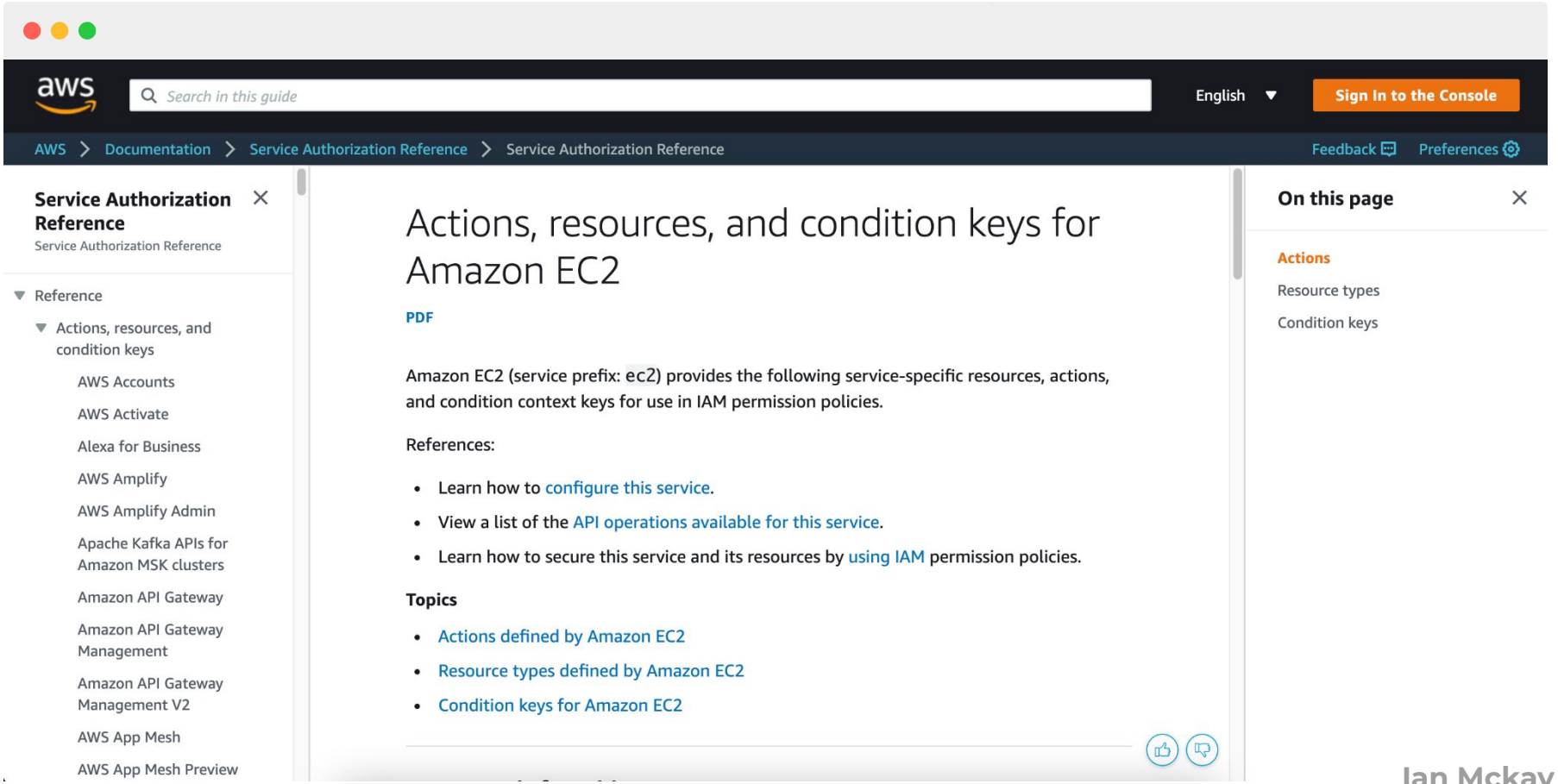
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TurnOnEC2InstancesInOneAZ",
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:AvailabilityZone": "us-east-1a"
                }
            }
        }
    ]
}
```

Ian Mckay
@iann0036

Ian Mckay
@iann0036

Ian Mckay
@iann0036

Ian Mckay
@iann0036



Ian Mckay
@iann0036

The Service Authorization Reference (SAR)

The AWS-provided source of truth for all AWS IAM permissions

IAM permissions are usually, but not always, mapped directly to the name of an API method

- ***Mapped Directly (~93%):*** EC2.StartInstances -> ec2:StartInstances
- ***Not Mapped Directly (~7%):*** S3.ListBuckets -> s3:ListAllMyBuckets

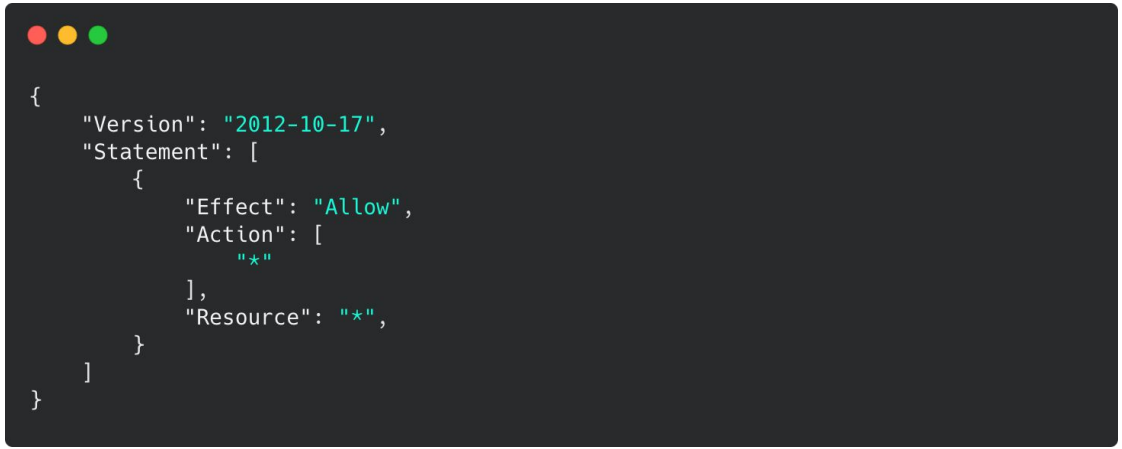
Actions	Description	Access Level	Resource Types	Condition Keys	Dependent Actions
MyAction	Description of the permission	List || Tagging || Read || Write || Permissions management	rtype1* rtype2	service:Prop	service:OtherAction

Least Privilege

Ian Mckay
@iann0036

Least Privilege

Is this least privilege?



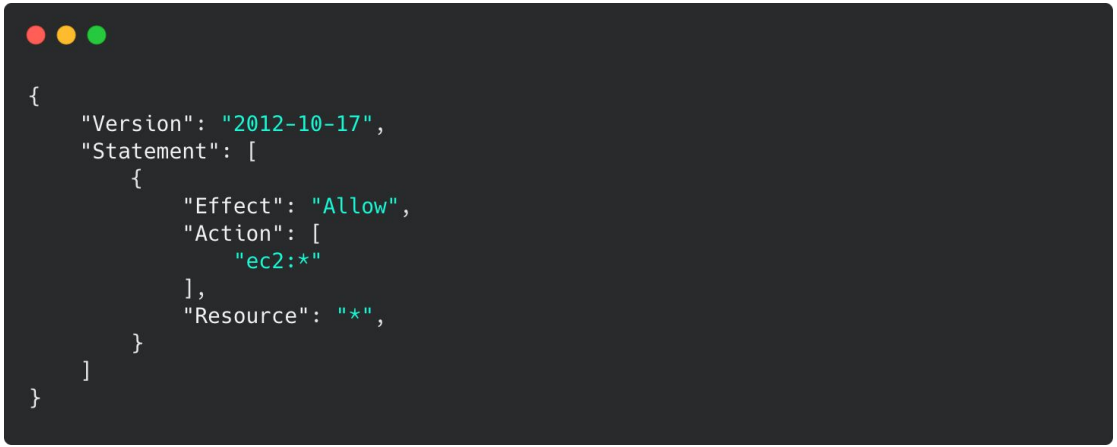
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "*"
            ],
            "Resource": "*",
        }
    ]
}
```

Ian Mckay
@iann0036

Least Privilege

Is this least privilege?

What if we limit by service?



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*"
            ],
            "Resource": "*",
        }
    ]
}
```

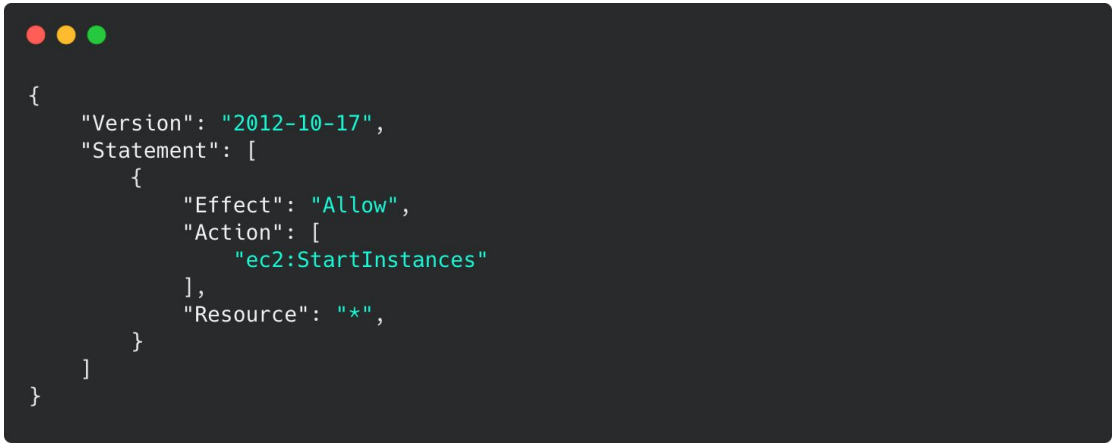
Ian Mckay
@iann0036

Least Privilege

Is this least privilege?

What if we limit by service?

What if we limit by action?



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "*",
        }
    ]
}
```

Ian Mckay
@iann0036

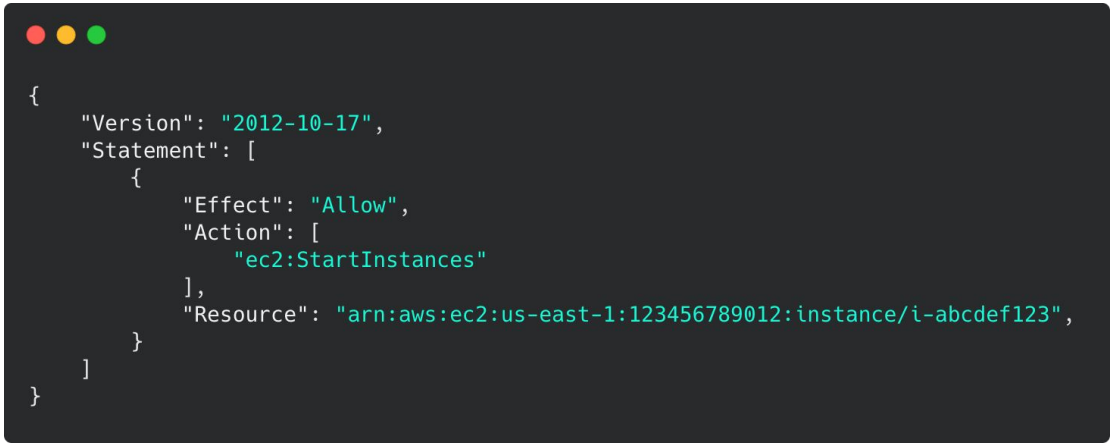
Least Privilege

Is this least privilege?

What if we limit by service?

What if we limit by action?

What if we limit by resource?



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-abcdef123",
        }
    ]
}
```

Ian Mckay
@iann0036

Least Privilege

Is this least privilege?

What if we limit by service?

What if we limit by action?

What if we limit by resource?

What if we include all the conditions?

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-abcdef123",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Owner": "${aws:username}",
                    "ec2:InstanceType": "t3.medium",
                    "ec2:Region": "us-east-1",
                    "ec2:Tenancy": "default",
                    "ec2:AvailabilityZone": "us-east-1a"
                },
                "Bool": {
                    "ec2:EbsOptimized": "false"
                }
            }
        }
    ]
}
```

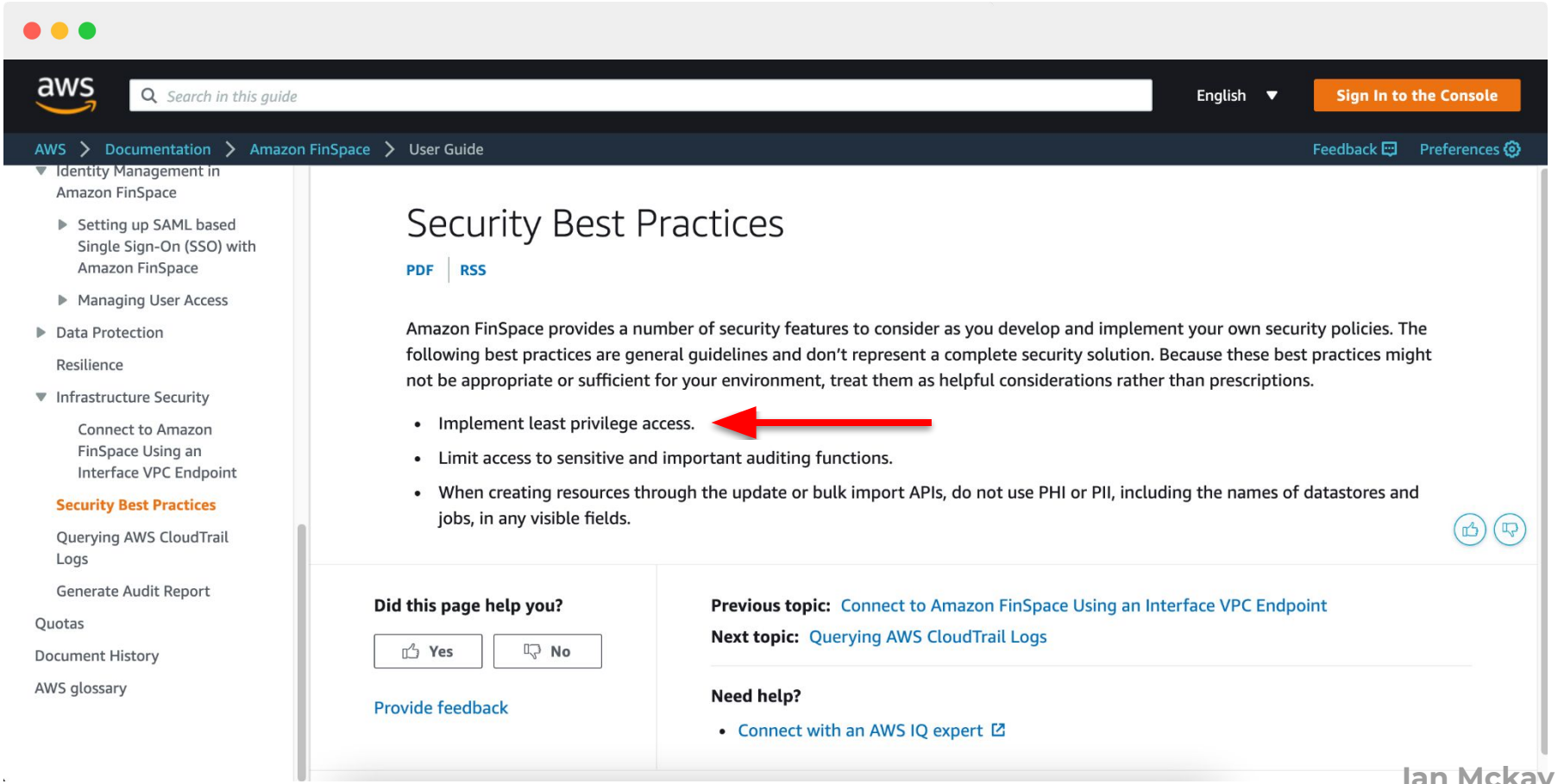
Ian Mckay
@iann0036

Least Privilege

Tools for achieving least privilege using AWS CloudTrail

- Trailscraper - Florian Sellmayr
- CloudTracker - Scott Piper (@0xdabbad00) & Duo Labs
- IAM Access Analyzer

Generally can scope down to specific actions, but often not to the resource, as the information is sometimes unavailable in a CloudTrail log entry

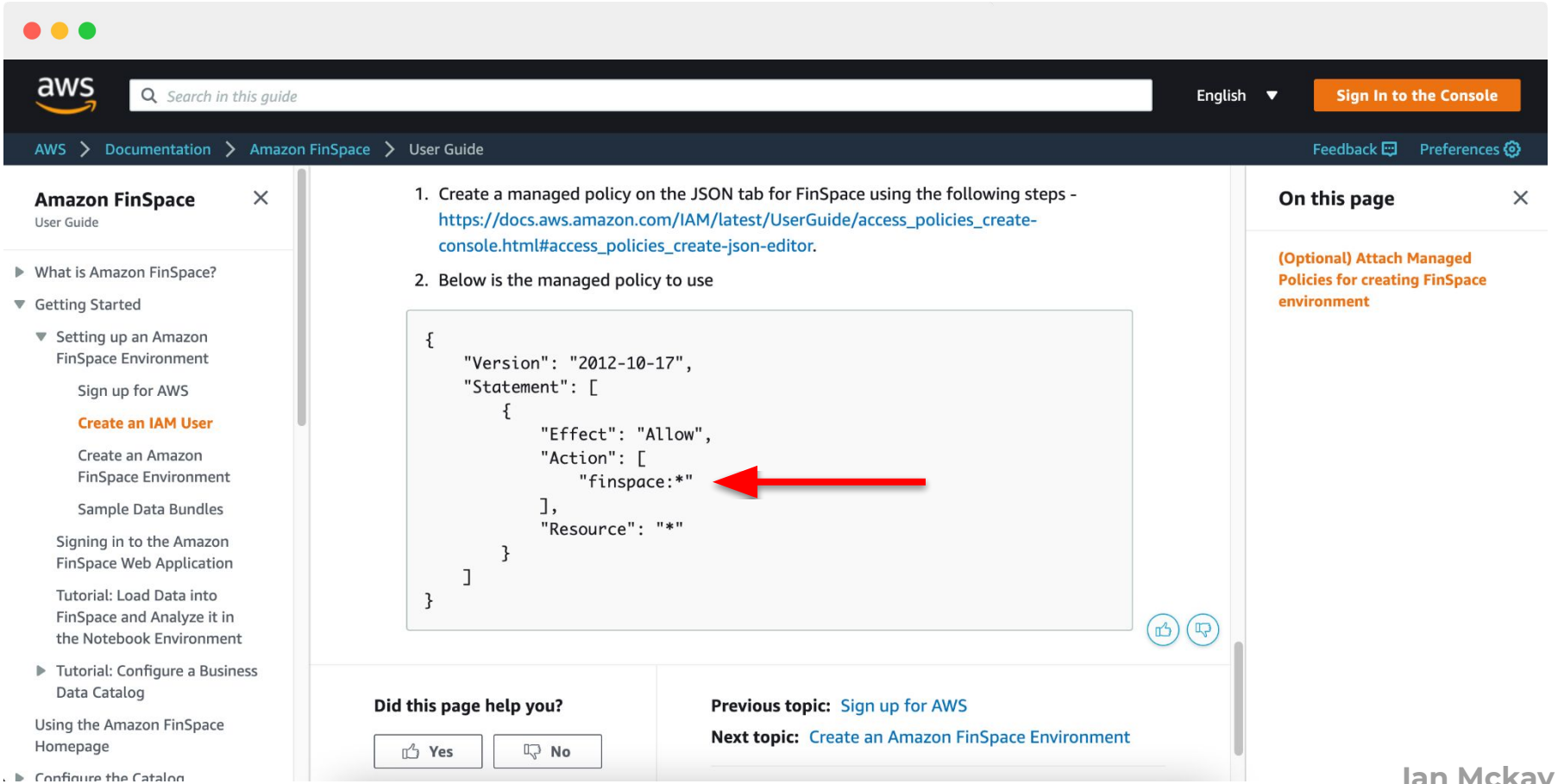


6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter Administrators.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box next to **AdministratorAccess** policy. Then choose **Create group**.

> **Note**
> You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities

Ian Mckay
@iann0036



Ian Mckay
@iann0036

iamlive

Generates an IAM policy from AWS calls

Works with the AWS CLI, AWS SDKs, most tools utilizing the SDKs (e.g. Terraform)

Started with client-side monitoring (CSM) mode

- Research from Scott Piper (@0xdabbad00) https://summitroute.com/blog/2020/05/25/client_side_monitoring/

```
POST https://ec2.amazonaws.com/
?Action=StartInstances&InstanceId.1=i-1234567890abcdef0
```

via UDP port 31000

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:*:instance/*"
        }
    ]
}
```

Ian Mckay
@iann0036

iamlive

Added an embedded HTTP(s) proxy approach to gather properties within the API call, to construct the full resource ARN

Works with the 5 primary AWS call protocols

AWS Account ID determined using Access Key ID

- Research from Aidan Steele (@__steele)
  https://awsteele.com/blog/2020/09/26/aws-access-key-format.html

```
POST https://ec2.amazonaws.com/
?Action=StartInstances&InstanceId.1=i-1234567890abcdef0
```

via HTTP(s) proxy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
        }
    ]
}
```

Ian Mckay
@iann0036

Problems with the SAR

Ian Mckay
@iann0036

Problems with the SAR - Fundamental

Undocumented actions

- 378 / 9712 undocumented as of March 2021 (~4%)
- Probable list constructed using AWS methods that didn't have an associated IAM permission
- Confirmed by calling the methods (fuzzing) using the API and a zero permissions user, then gathering ARN templates by interpreting error responses

Missing / poor resource type or conditions information

- *
- arn:${Partition}:medialive:${Region}:${Account}:channel:*
- arn:${Partition}:cloud9:${Region}:${Account}:environment:${ResourceId} (97 occurrences)

Problems with the SAR - Fundamental

Missing dependant action information (in particular iam:PassRole)

- The SAR only documents ~60 of the >300 iam:PassRole dependencies found
- iam:PassRole research by Noam Dahan (@NoamDahan)
  https://ermetic.com/whats-new/blog/auditing-passrole-a-problematic-privilege-escalation-permission/

Dependant action “chains”

- 30 chains that are 3 deep
- 2 chains that are 4 deep

`mq:CreateBroker` depends on
`ec2:CreateVpcEndpoint` which depends on
`route53:AssociateVPCWithHostedZone` which depends on
`ec2:DescribeVpcs`

Problems with the SAR - Inconsistency

Spaces in resource names

- Lex/LexV2: "bot version", "bot alias", "intent version", "slottype version"
- Lambda: "code signing config", "function alias", "function version"

Incorrect Resource Type References

- quicksight:CreateDataSet (and similar) requires “datasource”, but should be “dataset”
- machinelearning:GetEvaluation requires “datasource”, but should be “evaluation”

Incorrect ${Region} / ${Account} casing

- arn:${Partition}:ecs:${region}:${Account}:task-set/${ClusterName}/${ServiceName}/${TaskSetId}
- arn:${Partition}:transfer:${region}:${account}:user/${serverId}/${username}
- arn:${Partition}:chatbot::${account}:${resourceType}/${resourceName}

Ian Mckay
@iann0036

Problems with the SAR - Inconsistency

Use of ${AccountId} instead of ${Account} variable

- arn:${Partition}:chime::${AccountId}:...
- arn:${Partition}:datasync:${Region}:${AccountId}:...
- arn:${Partition}:gamelift:${Region}:${AccountId}:...
- arn:${Partition}:lookoutequipment:${Region}:${AccountId}:...
- arn:${Partition}:panorama:${Region}:${AccountId}:...

Incorrect ARN format

- arn:${Partition}:redshift:${Region}:${Account}:snapshotschedule:${ParameterGroupName}
- arn:${Partition}:ses:${Region}:${Account}:dedicated-ip-pool/${CustomVerificationEmailTemplateName}
- arn:${Partition}:ses:${Region}:${Account}:deliverability-test-report/${CustomVerificationEmailTemplateName}

Ian Mckay
@iann0036

Problems with the SAR - Weird ARNs

arn:${Partition}:organizations::${MasterAccountId}:handshake/o-${OrganizationId}/${HandshakeType}/h-${HandshakeId}	***(Leading variable value prefix)***
arn:${Partition}:mgh:${Region}:${Account}:progressUpdateStream/${Stream}/migrationTask/${Task}	***(Camel case)***
arn:${Partition}:iotthingsgraph:${Region}:${Account}:Workflow/${NamespacePath}	***(Upper case)***
arn:${Partition}:greengrass:${Region}:${Account}:/greengrass/things/${ThingName}/runtimeconfig	***(Leading slash)***
arn:${Partition}:opsworks:${Region}:${Account}:stack/${StackId}/	***(Trailing slash)***

Ian Mckay
@iann0036

Problems with the SAR - Weird ARNs

arn:${Partition}:mq:${Region}:${Account}:broker:${broker-id}	***(Dash in variable)***
arn:${Partition}:ssm:${Region}:${Account}:automation-definition/${AutomationDefinitionName:VersionId}	***(Colon within variable)***
arn:${Partition}:events:${Region}:${Account}:rule/[${EventBusName}/]${RuleName}	***(Optional variable)***
arn:${Partition}:${Vendor}:${Region}:*:${ResourceType}:${RecoveryPointId}	***(Barely populated ARN)***

Ian Mckay
@iann0036

Fixing the problems

iam-dataset - A new source of truth

https://github.com/iann0036/iam-dataset

Utilizes the work of other open source contributors

- Parliament - Scott Piper (@0xdabbad00) & Duo Labs (@duo_labs)
- MAMIP - Victor Grenu (@zoph)

The API Method to IAM Privilege mapping (map.json)

Easier crowdsourcing of IAM inaccuracies

The Mapping

Primary data source for iamlive

~10,000 API mappings

~100,000 lines of JSON

~3 months effort to build from scratch

Built with a custom mapping tool

```
"AutoScaling.PutLifecycleHook": [
    {
        "action": "autoscaling:PutLifecycleHook",
        "resource_mappings": {
            "GroupId": {
                "template": "*"
            },
            "GroupFriendlyName": {
                "template": "${AutoScalingGroupName}"
            }
        }
    },
    {
        "action": "iam:PassRole",
        "arn_override": {
            "template": "${RoleARN}"
        }
    }
]
```

Ian Mckay
@iann0036

Pinpoint.GetJourneyDateRangeKpi

[docs] Skip To [*]

Action: mobiletargeting:GetJourneyDateRangeKpi

Restype: apps*
ARN style: arn:${Partition}:mobiletargeting:${Region}:${Account}:apps/${AppId}
fullarn:
AppId: ${ApplicationId}

Restype: journeys*
ARN style:
arn:${Partition}:mobiletargeting:${Region}:${Account}:apps/${AppId}/journeys/${JourneyId}
fullarn:
AppId: ${ApplicationId}
JourneyId: ${JourneyId}

Submit Mapping [`] Skip Mapping [\]

httprequesturi: /v1/apps/{application-id}/journeys/{journey-id}/kpis/daterange/{kpi-na

ApplicationId [A]: {"location":"uri","locationName":"application-id"}

EndTime [A]: {"shape":"S2w","location":"querystring","locationName":"end-time"}

JourneyId [A]: {"location":"uri","locationName":"journey-id"}

KpiName [A]: {"location":"uri","locationName":"kpi-name"}

NextToken [A]: {"location":"querystring","locationName":"next-token"}

PageSize [A]: {"location":"querystring","locationName":"page-size"}

StartTime [A]: {"shape":"S2w","location":"querystring","locationName":"start-time"}

Output


```
{
    "info": "This file is sourced from https://github.com/iann0036/iam-dataset",
    "sdk_permissionless_actions": [
        "DynamoDB.DescribeEndpoints",
        "STS.GetCallerIdentity"
    ],
```


Ian Mckay
@iann0036

permissions

permissions.cloud

The https://permissions.cloud/ website exposes the IAM Dataset information in a clean, easy-to-read format

Created in order to provide an alternate, community-driven source of truth for AWS identity

Ian Mckay
@iann0036

permissions

Search...

GENERAL

Dashboard

Reference Usage

Managed Policies

REFERENCE

Alexa for Business

AWS IAM Access Analyzer

AWS Accounts

AWS Certificate Manager

AWS Certificate Manager Private Certificate Authority

AWS Activate

Amazon Managed Workflows for Apache Airflow

AWS Amplify

AWS Amplify Admin

Amazon API Gateway Management

Amazon API Gateway Management V2

GENERAL / DASHBOARD

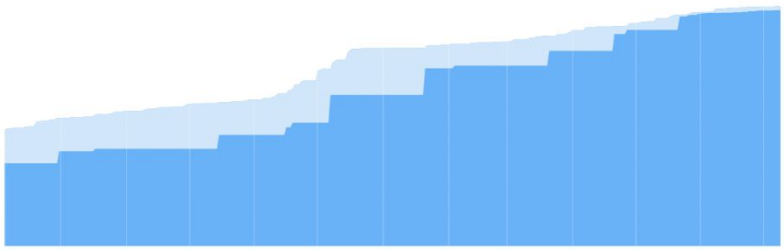
Dashboard

Global AWS Counts

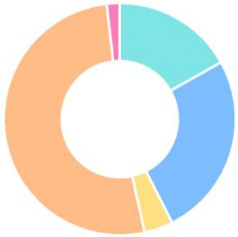
IAM PERMISSIONS API METHODS

10,359
IAM PERMISSIONS

10,436
API METHODS



Permissions by Access Level



LIST
1,748 17%

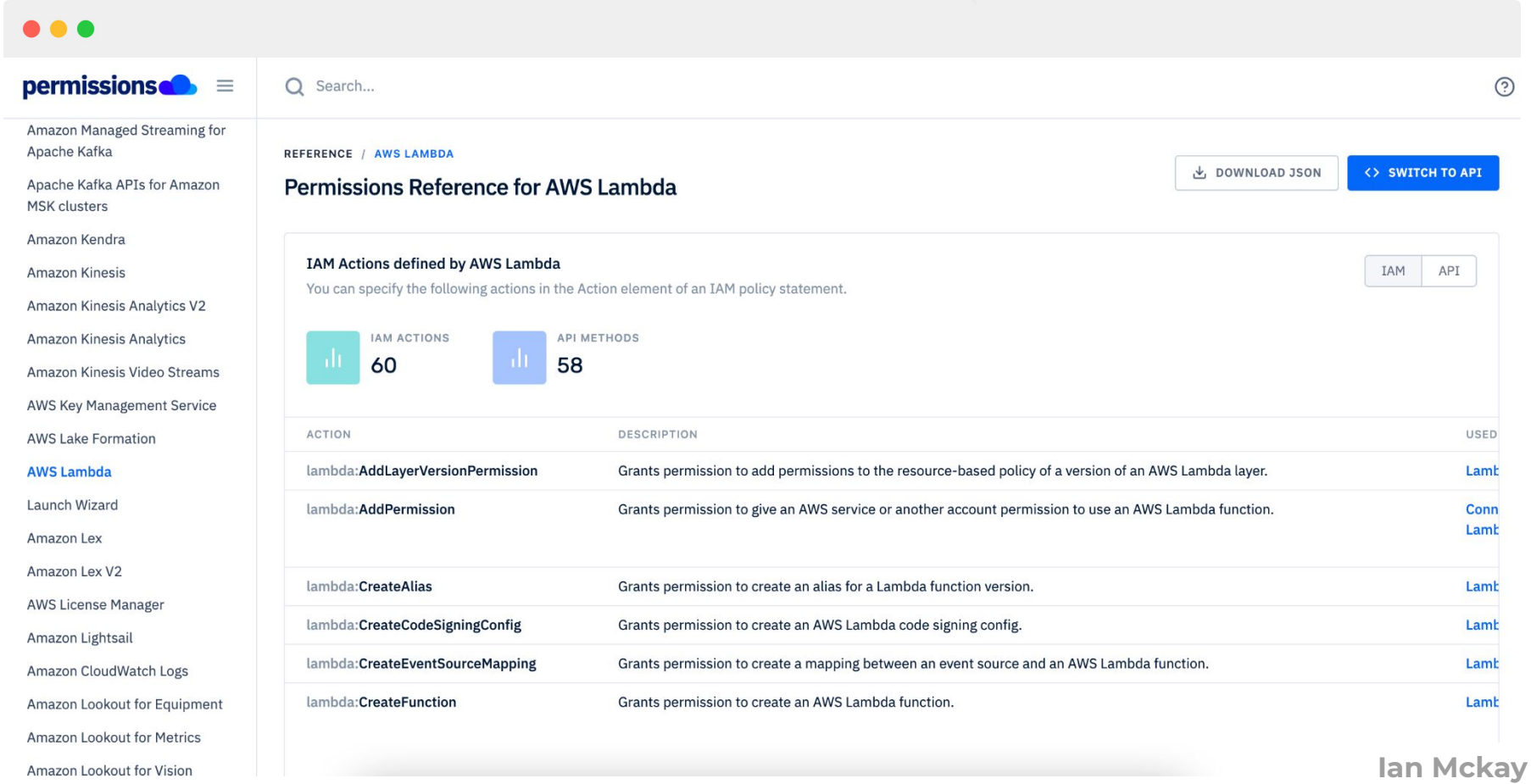
READ
2,660 26%

TAGGING
416 4%

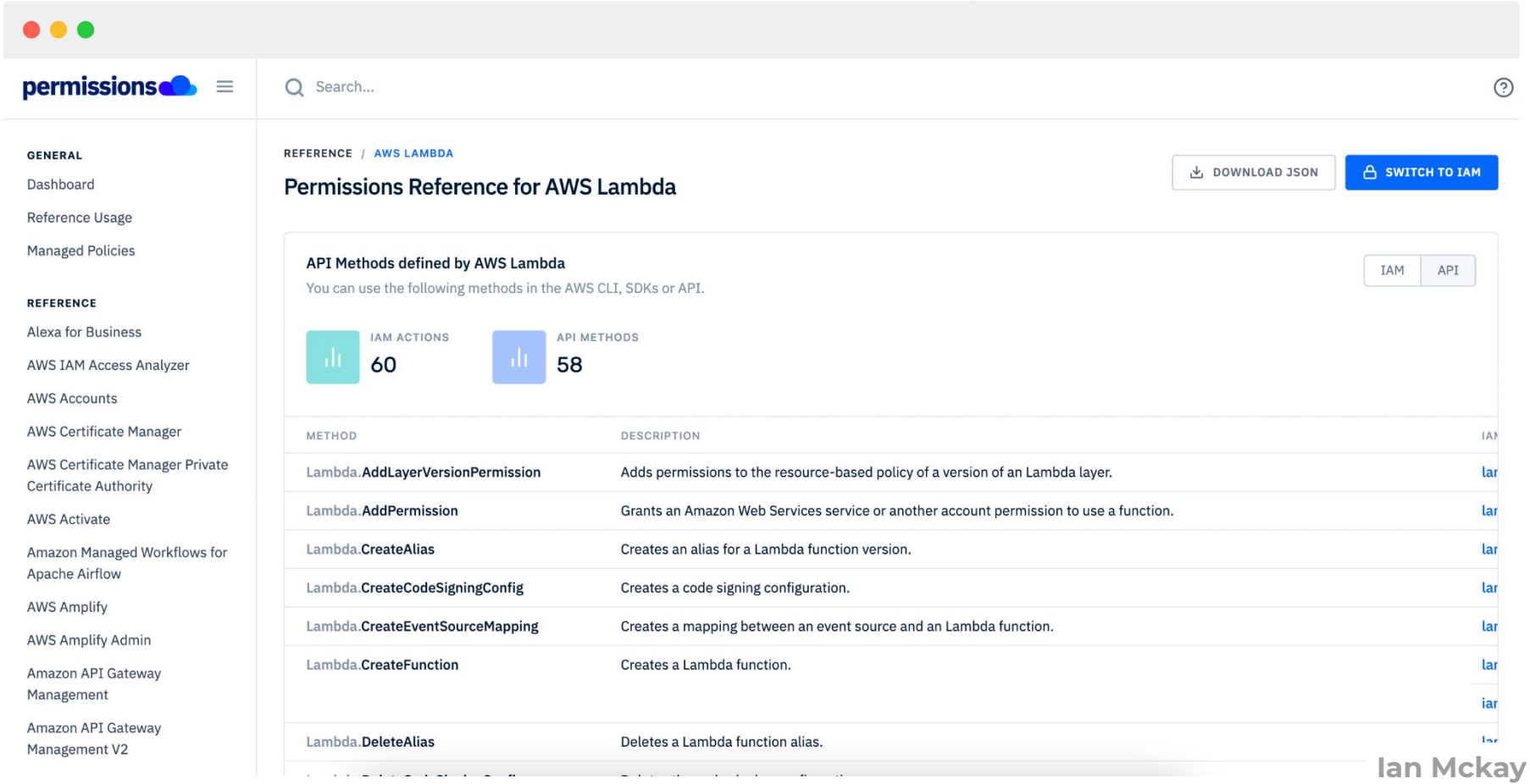
WRITE
5,349 52%

PERMISSIONS MANAGEMENT
186 2%

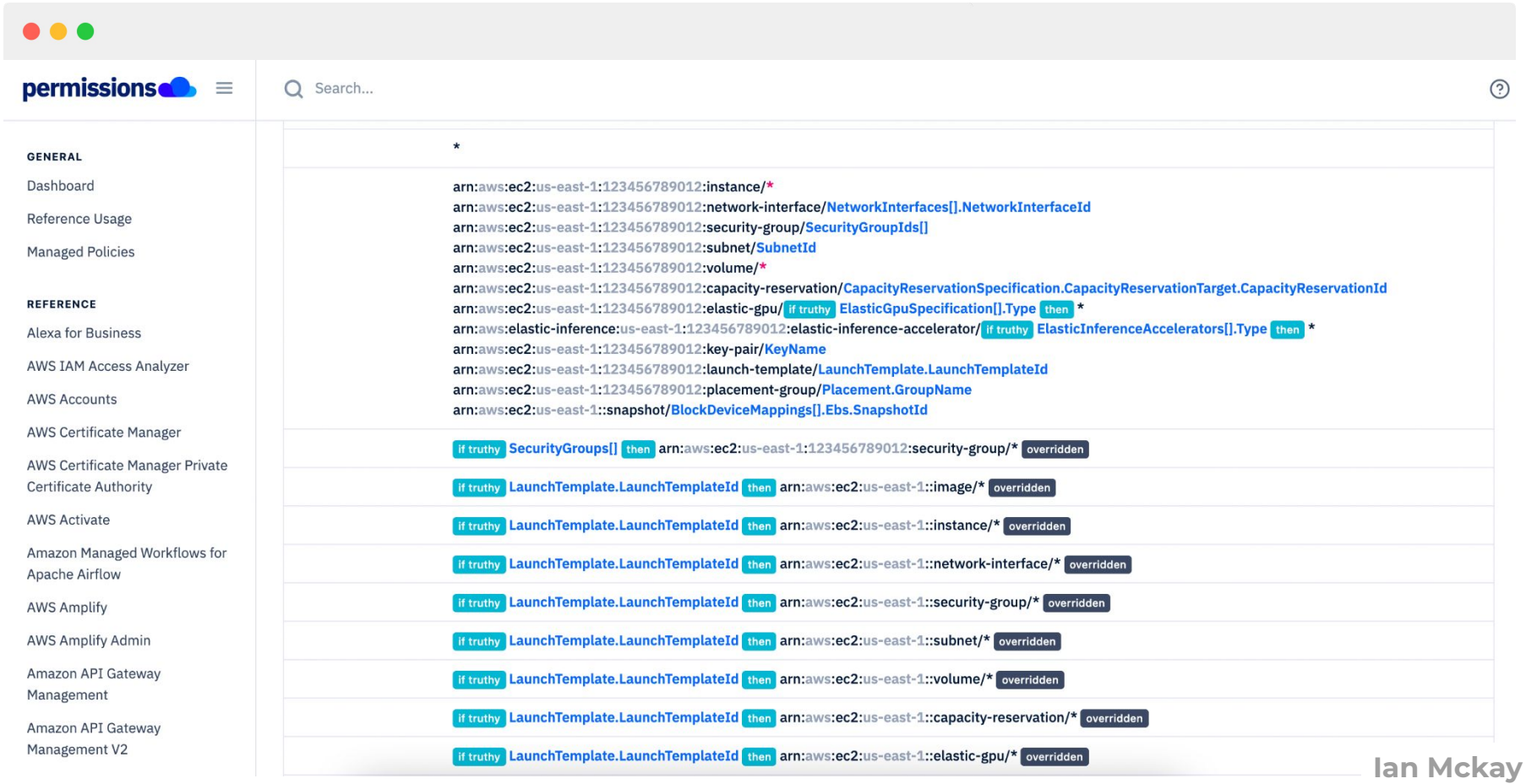
Ian Mckay
@iann0036



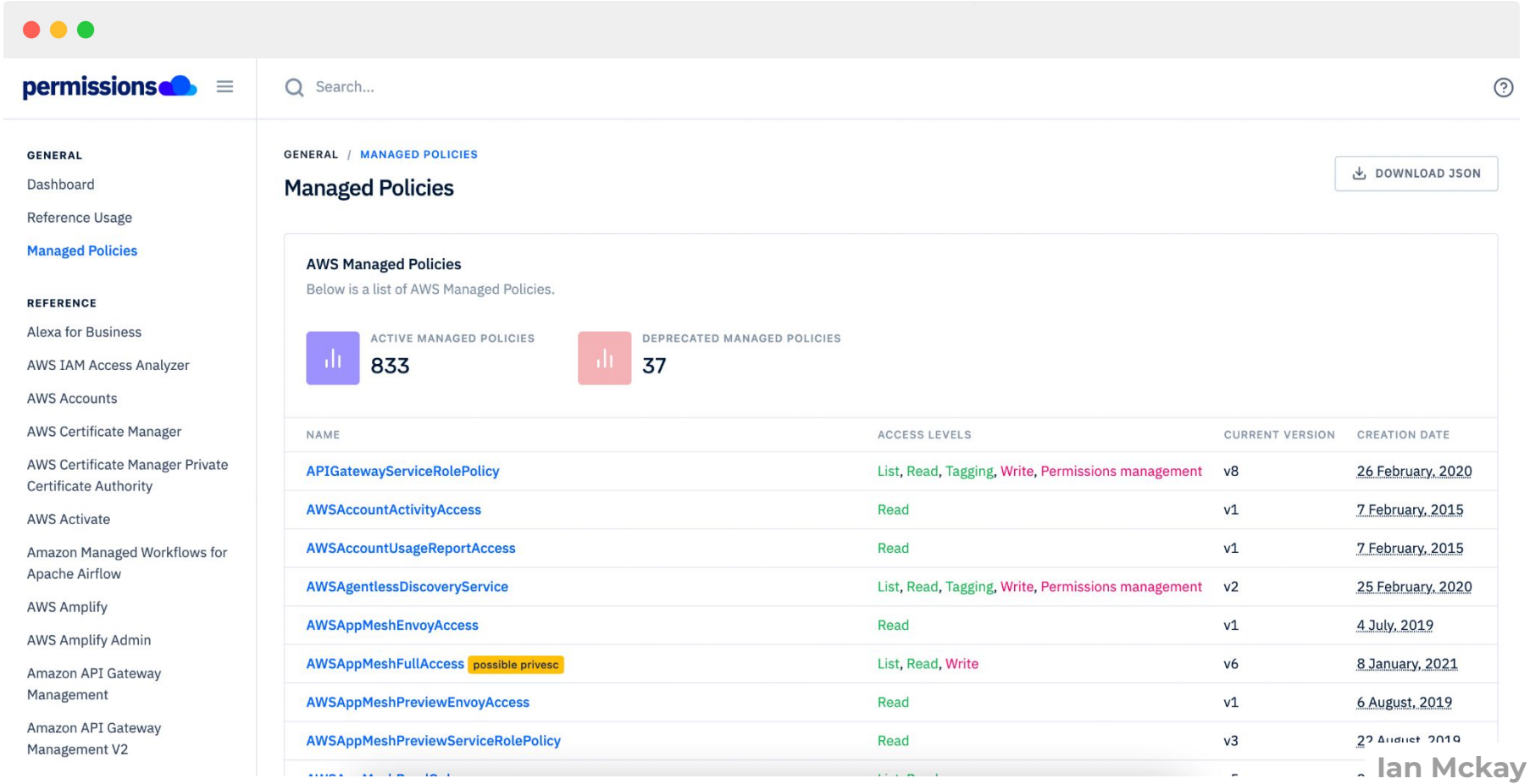
Ian Mckay
@iann0036



Ian Mckay
@iann0036



Ian Mckay
@iann0036



Ian Mckay
@iann0036

What’s Next

Ian Mckay
@iann0036

iamfast

AWS IAM policy generation from application code

First iteration targeting support for JavaScript, Python, Java & Go

```
> iamfast-js tests/test1.js

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dynamodb:PutItem",
            "Resource": [
                "arn:aws:dynamodb:us-east-1:123456789012:table/CUSTOMER_LIST"
            ]
        }
    ]
}
```

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'us-east-1'});

// Create the DynamoDB service object
var ddb = new AWS.DynamoDB({apiVersion: '2012-08-10'});

var params = {
  TableName: 'CUSTOMER_LIST',
  Item: {
    'CUSTOMER_ID' : {N: '001'},
    'CUSTOMER_NAME' : {S: 'Richard Roe'}
  }
};

// Call DynamoDB to add the item to the table
ddb.putItem(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

Thank you

https://permissions.cloud/

https://github.com/iann0036/iamlive

https://github.com/iann0036/iam-dataset

https://github.com/iann0036/iamfast-js

https://bit.ly/fwdcs21-mckay

@iann0036