AWS Route 53 & IAM

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa

Content

- AWS Route 53
  - Hosted zones
  - NS , SOA, Alias, CNAME records
  - Routing policies
- AWS Certificate Manager
- IAM – Managing resource access
  - IAM policies
  - IAM roles
  - AWS STS (AssumeRole)
  - Trust relationship
  - Identity federation

Route 53

This is a highly available and scalable cloud Domain Name System (DNS) web service.

It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to applications by translating names, like www.example.com, into the numeric IP addresses, like 98.10.12.31.

An AWS service that allows management of website domains and DNS records.

Hosted zones

it represents a collection of records that can be managed together, belonging to a single parent domain name. All resource record sets within a hosted zone must have the hosted zone's domain name as a suffix.

For example, the amazon.com hosted zone may contain records named www.**amazon.com**, and www.aws.**amazon.com**, but not a record named www.amazon.ca.

One hosted zone costs $0.50 per month.

NS records

NS records point to the servers that help to translate domain names into the IP addresses that computers use to communicate with one another.

NS records are automatically created when you create a new hosted zone. Provide that to the domain name provider i.e., GoDaddy. Then you will have full control on your domain name in AWS and create the required records.

Record types

- **Alias** - A type of record that you can create with Amazon Route 53 to route traffic to AWS resources such as ALB, Amazon CloudFront distributions and Amazon S3 buckets.
- **CNAME** - It maps one domain name to another domain name. For example, RDS, ElastiCache.
- **SOA** - The record is created with hosted zone along with NS records. The SOA record stores important information about a domain when the domain was last updated, and how long the server should wait between refreshes.

Read more: Supported record types in AWS

Sub domain

A domain name that has one or more labels prepended to the registered domain name.

For example, The **example.com** domain can have sub domains:

- *accounting*.**example.com**
- *hr*.**example.com**
- *it*.**example.com** so on.

You can create a hosted zone for the sub domain and create, manage its sub domains of the subdomain. For example, **it.example.com**

- *team1.***it**.**example.com**
- *team2.***it**.**example.com**

Time To Live (TTL)

The amount of time, in seconds, that you want a DNS resolver to cache (store) the values for a record before submitting another request to Route 53 to get the current (new) values for that record.

If the DNS resolver receives another request for the same domain before the TTL expires, the resolver **returns the cached value**.

A longer TTL reduces your Route 53 charges, which are based in part on the number of DNS queries that Route 53 responds to.

Routing policies

- **Simple routing policy** – Route internet traffic to a single resource for your domain.
- **Geolocation routing policy** – Use when you want to route internet traffic to your resources based on the location of your users.
- **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
- **Failover routing policy** – Failover routing lets you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy.

Read more: AWS Route 53 routing policies

Amazon Certificate Manager (ACM)

With ACM, you can

- Create and renew SSL/TLS X.509 certificates for free
- Import third-party certificates into the ACM and use it in your AWS resources by referring its ARN (Amazon Resource Name).

ACM certificates can secure wildcard domains. ACM wildcard certificates can protect an unlimited number of subdomains.

ACM generates a CNAME record that you need to add in the corresponding hosted zone.

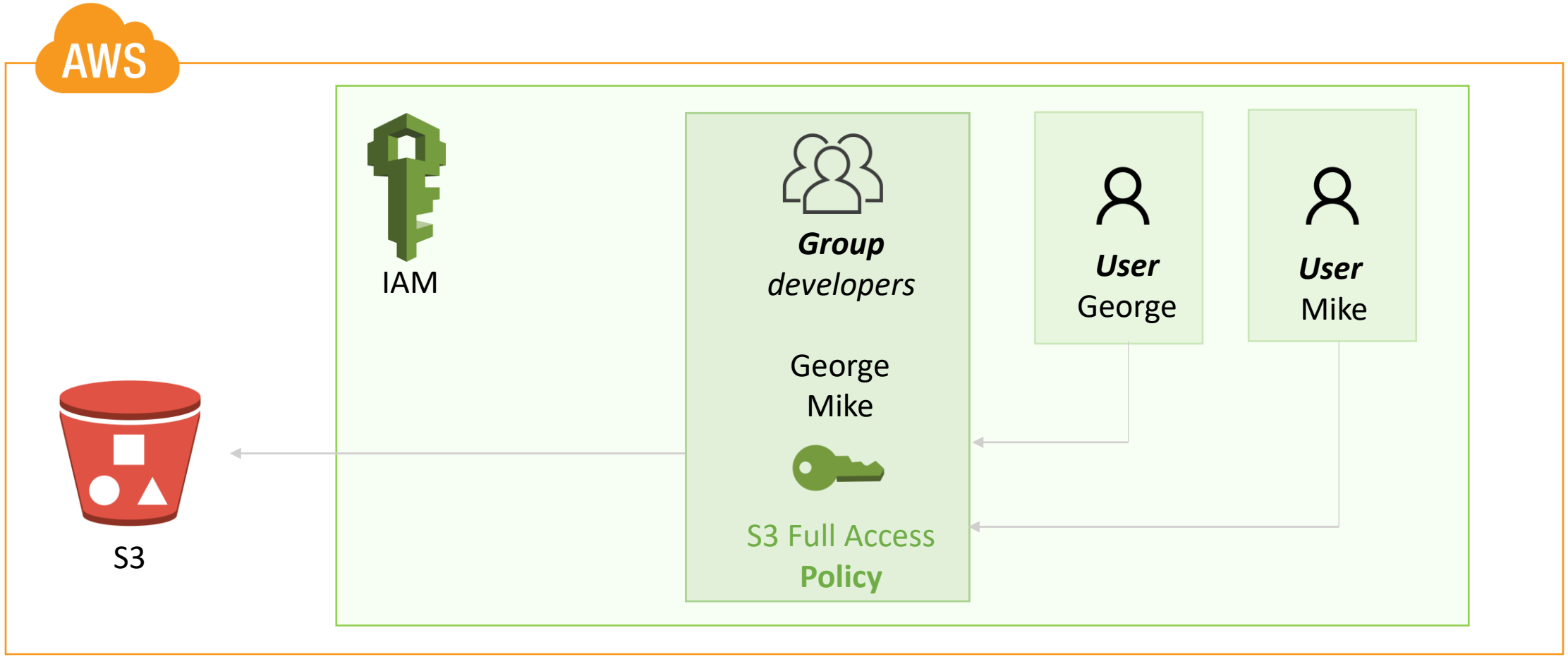
Identity & Access Management (IAM)

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who (user or role) is authenticated (signed in) and authorized (has permissions) to use resources.

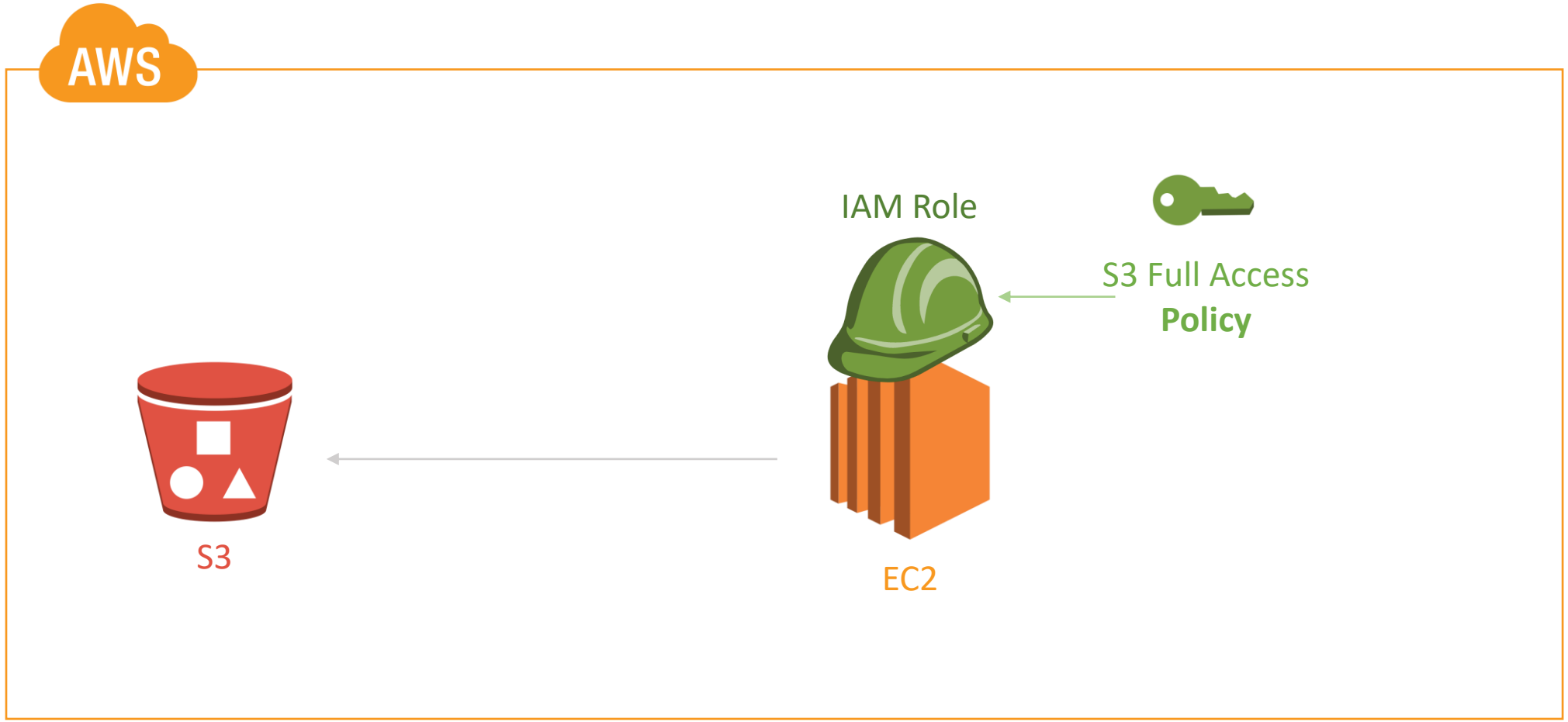
IAM is used to manage:

1. IAM Access **Policies** - We attach IAM Policies to Users, Groups, and Roles
2. **Users & Groups** - You developers
3. **Roles** - Resources in AWS

AWS IAM Group



AWS IAM Role



IAM Policies

- **IAM Policies** are **permissions** that you can assigned to any User, Group, and Roles.
- We don't attach a IAM Policy to a Service, instead we would need to use a **Role**.
- There are **AWS managed** policies and **user managed** policies.
- **Identity-based** policies – Attach policies to IAM identities (users, or roles)
- **Resource-based** policies – Attach inline policies to resources (S3). Has a **principal** tag.

Read More about: IAM Policies

IAM JSON policy elements: Condition

The Condition element (or Condition block) lets you specify conditions for when a policy is in effect.
In the Condition element, you build expressions in which you use condition operators (equal, less than, etc.) to match the condition keys and values in the policy against keys and values in the request context.
Learn more: [IAM JSON policy elements: Condition]

```
"Condition" : { "{condition-operator}" : { "{condition-key}" : "{condition-value}" }}
```

```
"Condition" : { "StringEqualsIgnoreCase" : { "aws:username" : "johndoe" }}
```

```
{
    "Statement":[{
      "Effect":"effect",
      "Action":"action",
      "Resource":"arn",
      "Condition":{
        "condition":{
          "key":"value"
          }
        }
      }
    ]
  }
```

The **Effect** element can be Allow or Deny

The **Action** element is the specific API action for which you are granting or denying permission

The **Resource** element specifies the resource that's affected by the action

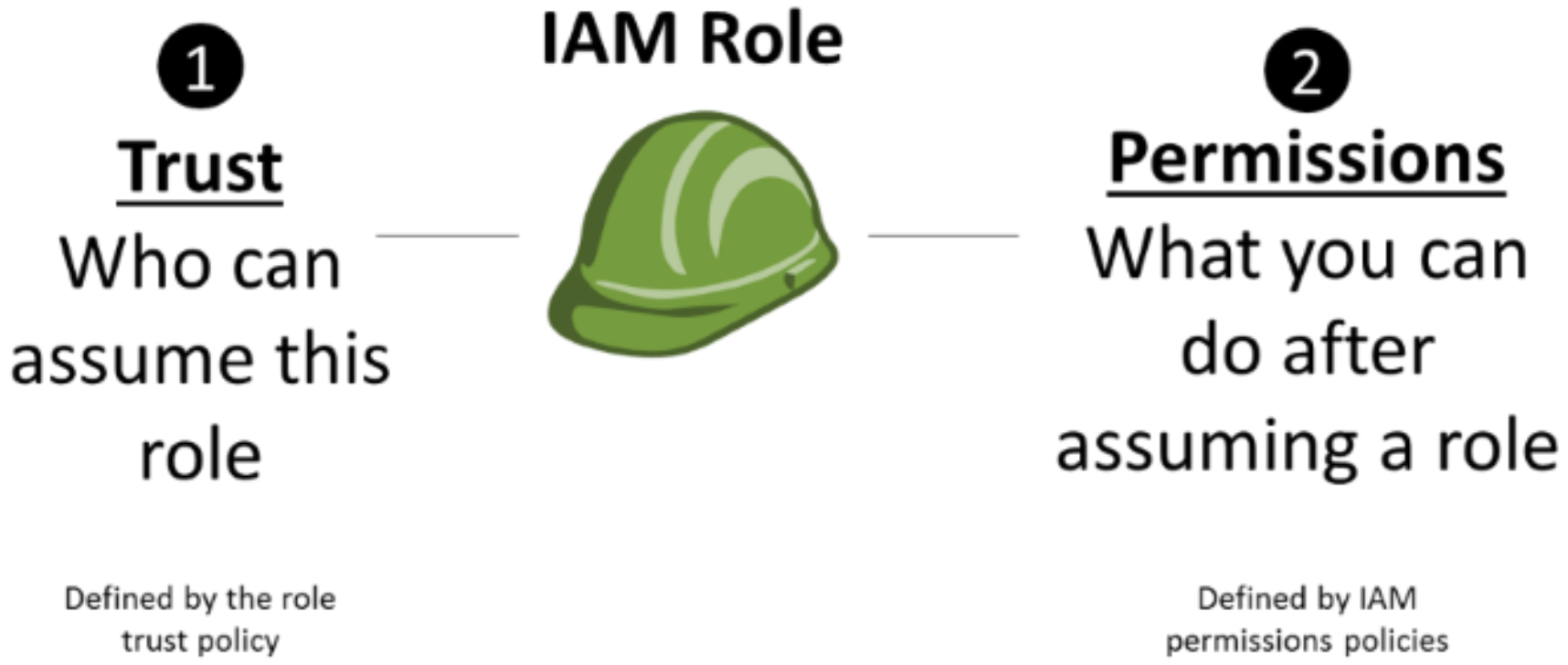
The **Condition** element is optional and can be used to control when your policy is in effect

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

        {
            "Effect": "Allow",
            "Action": [
                "rds:RebootDBInstance",
                "rds:StartDBInstance",
                "rds:StopDBInstance"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalTag/Department": "DBAdmins",
                    "rds:db-tag/Environment": "Production"
                }
            }
        }
    ]
}
```

IAM role

An IAM role is an IAM entity that **defines** a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, **trusted entities assume roles**, such as IAM users, applications, or AWS services such as EC2.



AWS Security Token Service (AWS STS)

STS is a web service that enables you to request **temporary** credentials for IAM role, users, or federated users. When assume a role, it calls AWS STS under the hood.

The temporary credentials consist of:

1. **Access key ID** - Access keys are long-term credentials for an IAM user. Like username.
2. **Secret access key** - Like password.
3. **Session token** - Validates temporary credentials.
4. **Duration** - defines how long the temporary credentials lasts. Most cases 12 hours. 15-min is min.

STS - AssumeRole

When "you" are code running on an EC2 instance, and the instance has an instance role, the EC2 infrastructure actually calls assume-role on behalf of the instance, and you can fetch the temporary credentials.

RoleArn is required when assuming a role.

```
https://sts.amazonaws.com/
?Version=2011-06-15
&Action=AssumeRole
&RoleSessionName=testAR
&RoleArn=arn:aws:iam::123456789012:role/demo
&PolicyArns.member.1.arn=arn:aws:iam::123456789012:policy/demopolicy1
```

```xml
<AssumeRoleResponse xmlns="https://sts.amazonaws.com/doc/2011-06-15/">
  <AssumeRoleResult>
  <SourceIdentity>Alice</SourceIdentity>
    <AssumedRoleUser>
      <Arn>arn:aws:sts::123456789012:assumed-role/demo/TestAR</Arn>
      <AssumedRoleId>ARO123EXAMPLE123:TestAR</AssumedRoleId>
    </AssumedRoleUser>
    <Credentials>
      <AccessKeyId>ASIAIOSFODNN7EXAMPLE</AccessKeyId>
      <SecretAccessKey>wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY</SecretAccessKey>
      <SessionToken>
       AQoDYXdzEPT/////////wEXAMPLEtc764bNrC9SAPBSM22wDOk4x4HIZ8j4FZTwdQW
       LWsKWHGBuFqwAeMicRXmxfpSPfIeoIYRqTflfKD8YUuwthAx7mSEI/qkPpKPi/kMcGd
       QrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburEDy0KPkyQDYwT7WZ0wq5VSXDvp75YU
       9HFvlRd8Tx6q6fE8YQcHNVXAkiY9q6d+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
       +scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==
      </SessionToken>
      <Expiration>2019-11-09T13:34:41Z</Expiration>
    </Credentials>
    <PackedPolicySize>6</PackedPolicySize>
  </AssumeRoleResult>
  <ResponseMetadata>
    <RequestId>c6104cbe-af31-11e0-8154-cbc7ccf896c7</RequestId>
  </ResponseMetadata>
</AssumeRoleResponse>
```

Making AWS STS call with AWS SDK

In this example, it is getting **temporary** credentials by assuming a role. When assume a role, it calls AWS STS under the hood. Then using the temporary credentials to make a call to AWS S3.

```
assumeRoleResult = AssumeRole(role-arn);
tempCredentials = new SessionAWSCredentials(
   assumeRoleResult.AccessKeyId,
   assumeRoleResult.SecretAccessKey,
   assumeRoleResult.SessionToken);
s3Request = CreateAmazonS3Client(tempCredentials);
```

Trust relationships

With IAM roles, you can establish trust relationships between AWS services , accounts, and identity federation. It is a policy **who** can assume that role to get temporary credentials to do actions in the permission policy.

Permission policy defines **what** actions the user or role can do. For example, get or put an object from S3.

Following json trust policy user1 and user2 can assume the role.

```
"Principal": {
    "AWS": [
        "arn:aws:iam::AWS-account-ID:user/user1",
        "arn:aws:iam::AWS-account-ID:user/user2"
    ]
}
```

Trust relationships in AWS console

In this example, support.amazonaws.com can assume the AWSServiceRoleForSupport. There is no conditions.

Roles > AWSServiceRoleForSupport

Summary

This service-linked role cannot be deleted in IAM. Learn more

Role ARN	arn:aws:ian ole/aws-service-role/support.amazonaws.com/AWSServiceRoleForSupport
Role description	Enables resource access for AWS to provide billing, administrative and support services | Edit
Instance Profile ARNs	
Path	/aws-service-role/support.amazonaws.com/
Creation time	2021-05-23 21:53 EST
Last activity	Not accessed in the tracking period

Permissions | Trust relationships | Tags | Access Advisor

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

Trusted entities

The following trusted entities can assume this role.

Trusted entities
The identity provider(s) support.amazonaws.com

Conditions

The following conditions define how and when trusted entities can assume the role.

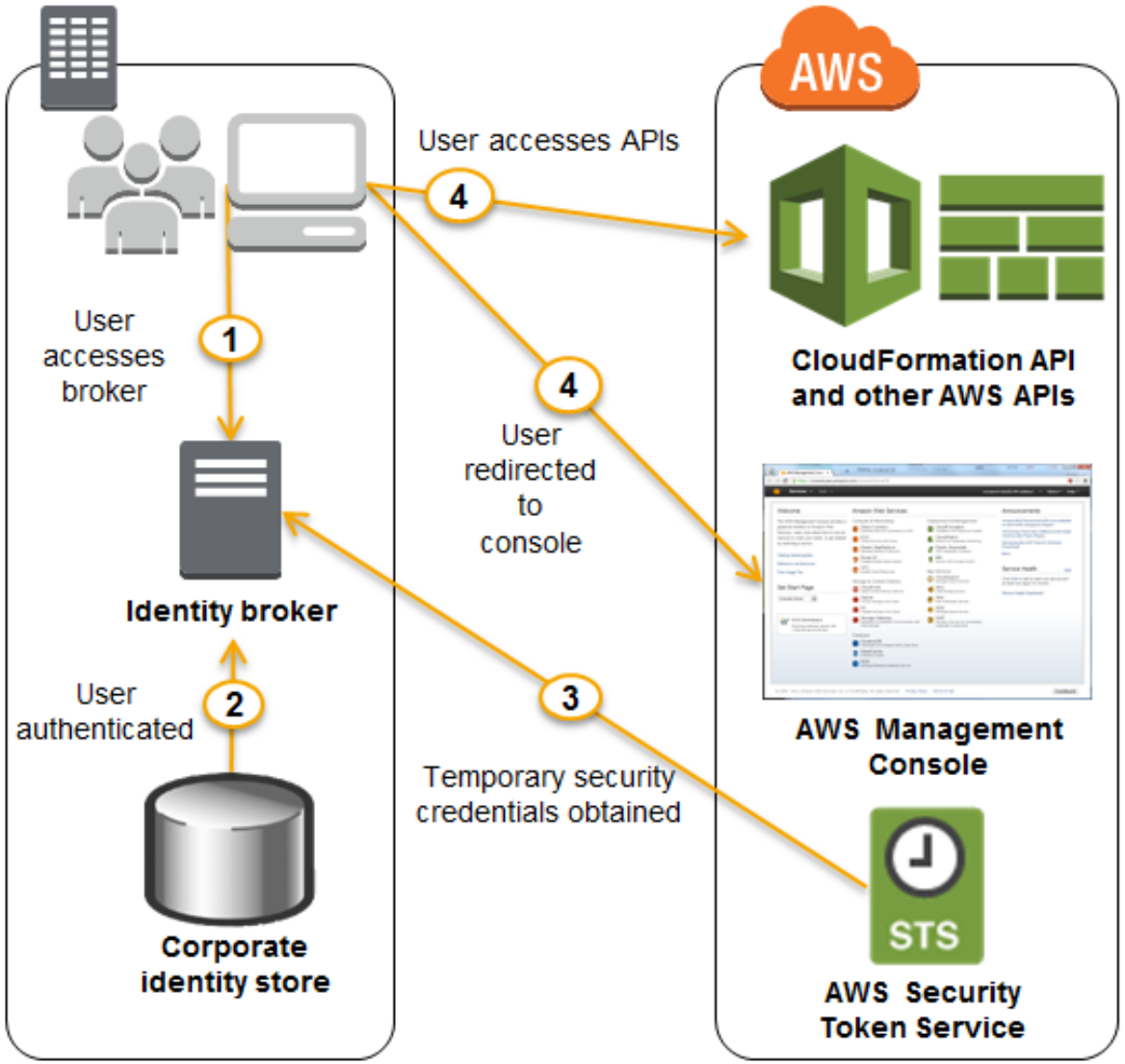
There are no conditions associated with this role.

Identity federation

Identity federation grants external identities secure access to resources in your AWS account. These external identities can come from your corporate identity provider such as Microsoft Active Directory or from a web identity provider such as Amazon Cognito, Facebook, Google.

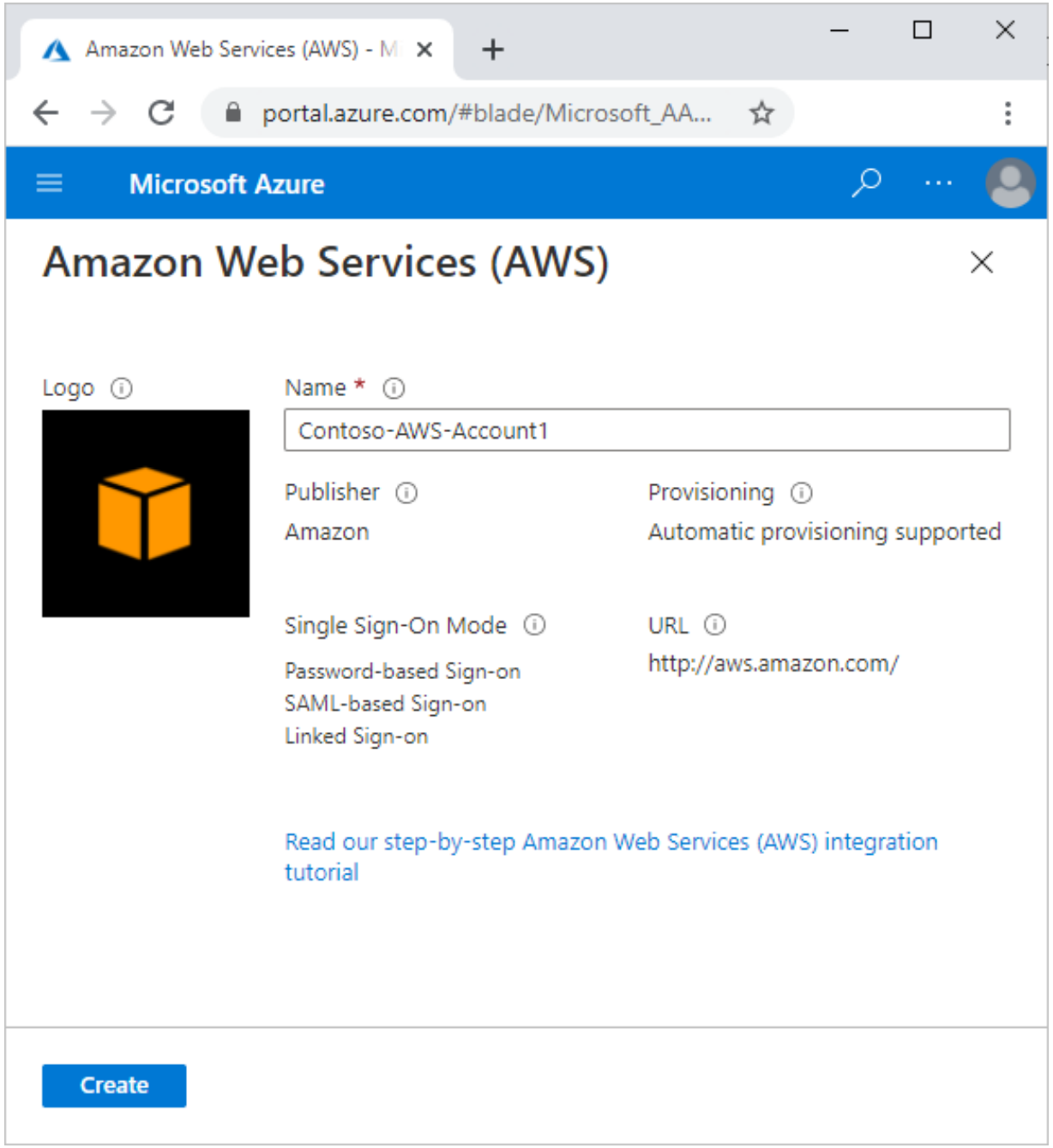
Federated users (external identities) are users you manage outside of AWS in your corporate directory, but to whom you grant access to your AWS account using **temporary security credentials**.

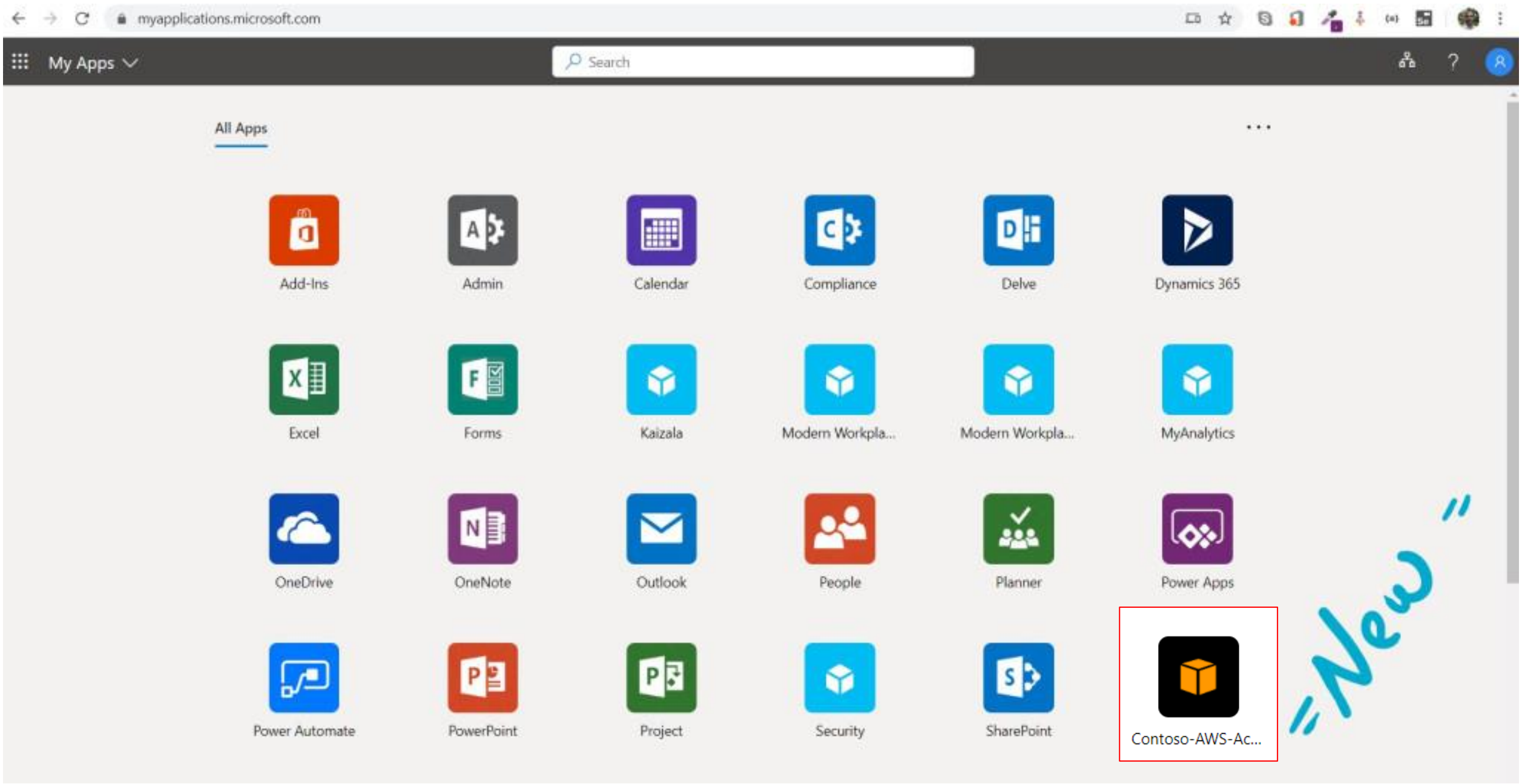
Federated users and temporary security credentials STS

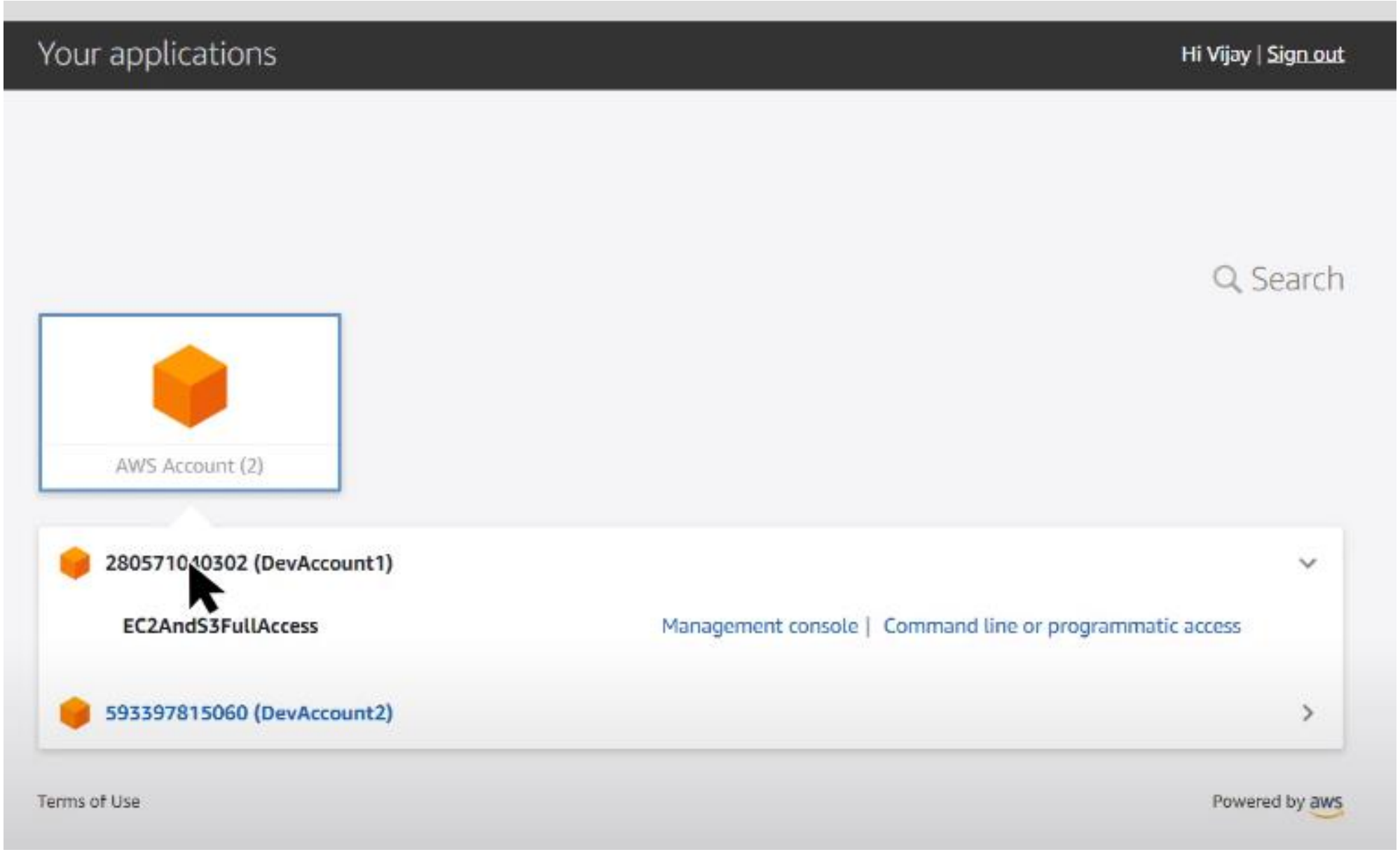


A new Azure AD enterprise application is being created for AWS.
With this setup, we can access AWS by signing into our Microsoft account.

Read more: [Azure AD for AWS]




myapplications.microsoft.com
My Apps
Search
All Apps
Add-Ins
Admin
Calendar
Compliance
Delve
Dynamics 365
Excel
Forms
Kaizala
Modern Workpla...
Modern Workpla...
MyAnalytics
OneDrive
OneNote
Outlook
People
Planner
Power Apps
Power Automate
PowerPoint
Project
Security
SharePoint
Contoso-AWS-Ac...
"New"



Learn more: [AWS SSO]