



AWS Multi-Account Configuration Guide



info@okta.com
1-888-722-7871

Table of Contents

Overview..... 3

 How it Works 3

 User Access to AWS Accounts and Roles 3

 Managing User & Group Access to Accounts and Roles..... 5

 High-Level Design..... 7

 Set Up AWS for SAML 7

 Create a Management Layer of Groups in AD / LDAP 7

 Configure the AWS App in Okta for Group-Based Role Assignment..... 7

Set Up Instructions..... 8

 Prerequisites..... 8

 Step 1: Setting Up Your AWS Accounts & Roles for SAML SSO 8

 Step 2: Creating AWS Role Groups in AD / LDAP 8

 Step 3: Configuring AD / LDAP Management Groups to Map Users to AWS Accounts & Roles 9

 Step 4: Importing AWS Role Groups and Management Groups into Okta 11

 Step 5: Enabling Group Based Role Mapping in Okta 12

 Step 6: Assign All AWS Management Groups to the AWS App in Okta..... 13

Overview

It has become increasingly common for AWS customers have a large set of AWS accounts – some for development, some for testing, others for production, etc. In fact, it is not uncommon to have over 100 AWS accounts to manage all of these use cases.

In response, it is now possible in Okta to provide a secure and scalable way of granting single sign-on access across an unlimited number of AWS accounts and roles. Additionally, this model ensures that each group of users are only granted access to the appropriate AWS roles they need, offering fine-grained entitlement management.

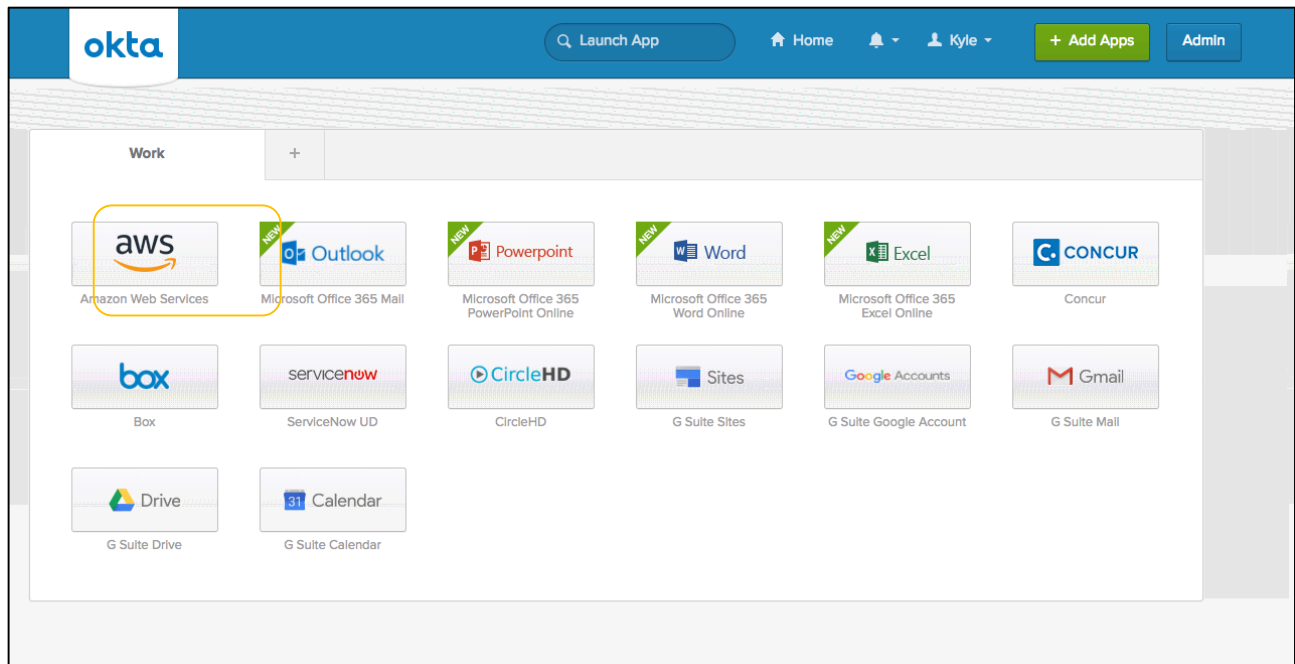
This is an Early Access feature. Contact Okta Support to enable it.

This guide will explain how the Okta's AWS Multi-Account solution works and walks through set-up instructions to get started with the new feature.

How it Works

User Access to AWS Accounts and Roles

Once you have granted AWS access to certain individuals or groups, each user will begin by simply logging into the Okta End-User Dashboard. From here they can then select an **AWS** chicklet that appears once they have been assigned the app.



Once the AWS app is selected, an AWS account & role picker page will appear. This page will display all of the roles across all of the accounts that the specific user is granted access to. This will differ depending on the entitlements that users are granted – for instance, your DevOps administrator may see roles and accounts requiring more elevated permissions as compared to your Tier 1 Support agent.

Select a role:

▼ Account: PROD (96629466...)

☐ **EC2_FullAccess**
☐ **EC2_ReadOnly**

▼ Account: TEST (86593466...)

☐ **EC2_FullAccess**
☐ **EC2_ReadOnly**

Sign In









Behind the scenes, Okta is able to pass a list of roles and accounts the user is authorized for to AWS in real time based off the specific groups that the user belongs to. This makes administration extremely simple, by allowing admins to simply assign users to specific AD or LDAP groups that are authorized for a certain list of AWS accounts & roles. More details are explained below about the admin experience.

Managing User & Group Access to Accounts and Roles

In the initial release of this solution, administration of this feature is primarily supported in AD & LDAP. From here, administrators work with two different logical sets of AD / LDAP groups:

1 AWS Role Specific Groups

A group must exist in AD or LDAP for each specific account and role combination that you want to provide access to. You can think of these groups as *AWS Role Specific Groups*. The group name should follow a particular syntax as well (*more details in [setup instructions](#) on this topic*) .





AWS Role Groups 8 objects	
Name	Type
 aws#oktademo-eng#EC2_FullAccess#828314669240	Security Group - Global
 aws#oktademo-eng#EC2_ReadOnly#828314669240	Security Group - Global
 aws#oktademo-eng#S3_FullAccess#828314669240	Security Group - Global
 aws#oktademo-eng#S3_ReadOnly#828314669240	Security Group - Global
 aws#oktademo-prod#EC2_FullAccess#964913662442	Security Group - Global
 aws#oktademo-prod#EC2_ReadOnly#964913662442	Security Group - Global
 aws#oktademo-prod#S3_FullAccess#964913662442	Security Group - Global
 aws#oktademo-prod#S3_ReadOnly#964913662442	Security Group - Global

Any user who is a member of these role specific groups is essentially granted a single entitlement - access to one specific role in one specific AWS account. These groups can be created by a script, exported as a list from AWS, or created manually.

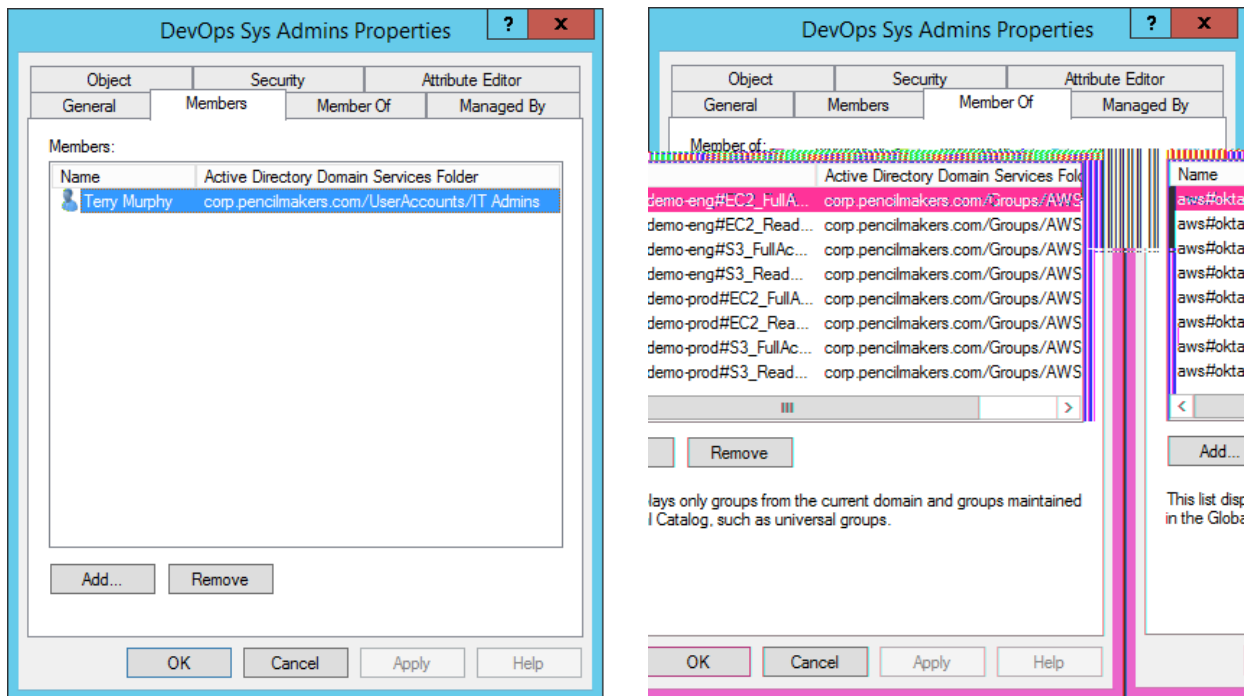
2 Management Groups

As you might imagine, it does not scale to manage user access by assigning each user to specific AWS Role Groups. To simplify administration, we recommend you also create a number of groups for all of the distinct user-sets in your organization that require different sets of AWS entitlements.

These groups may already exist in your AD/LDAP hierarchy in the form of different department specific groups, but can also be created solely for AWS if preferred.

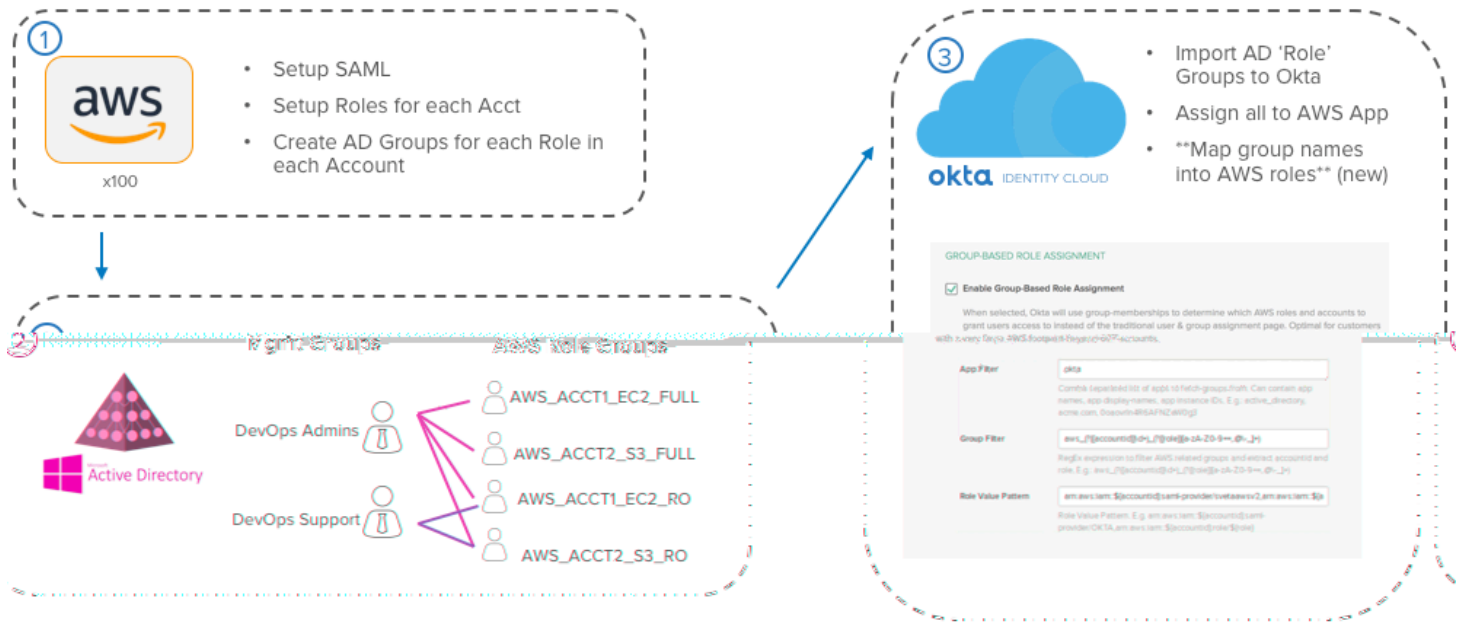
AWS Mgmt Groups 4 objects	
Name	Type
 DevOps Sys Admins	Security Group - Global
 DevOps Tier 1	Security Group - Global
 EC2 Admins	Security Group - Global
 S3 Admins	Security Group - Global

These management groups become the administration layer where you assign users (as group *Members*) and map these users to specific entitlements through AWS Role Groups (as *Members Of*)



Once these groups have been created in Active Directory or LDAP, all administration should take place with the Management Groups. Add / Remove users to these groups to grant access to your listed AWS accounts & roles, and update the specific entitlements by adding or removing AWS Role Groups in the *Member Of* group property.

High-Level Design



Set Up AWS for SAML

To begin, each of your AWS accounts must be configured for SAML access. This entails adding Okta as a trusted IDP to your AWS account and then creating a trust relationship for each of your roles that permits access via the new IDP. These are the same steps that one would follow to provide SAML SSO into any single AWS account, but must be performed across all of your accounts. For advanced organizations, this can be automated with Cloud Formation or AWS API scripts for simple SAML setup in each Account.

Create a Management Layer of Groups in AD / LDAP

Once SAML has been configured, you must now create [AWS Role Groups](#) in AD/LDAP for each role & account you want users to be able to access through Okta. This can be completed via a script between AWS and AD/LDAP, by exporting a CSV to AD and scripting against the CSV on the AD side, or by manual effort.

Next, you can create a link between these AWS Role specific Groups and other AD /LDAP groups by assigning [Management Groups](#) as *Members Of* the AWS Role Groups you want to grant them access to. Once complete, assign users to these Management groups to allow access to all of the AWS roles and accounts that the Management Group is a member of.

Configure the AWS App in Okta for Group-Based Role Assignment

Finally, in Okta, import both the AD/LDAP Management Groups & Role Groups via Okta's AD or LDAP Agent. Next, assign your management groups to the AWS application you set up in Step 1 – this assigns the proper users to the AWS app. Lastly, set up Group Based Role Assignment to translate the names of each of your AWS Role Groups into a format that AWS can consume to list the proper roles on the Role Picker Page for your users.

Set Up Instructions

These steps assume you understand the [intended experience](#) and [high-level design](#) of this feature. If unsure, please review the sections above.

Step 1: Setting Up Your AWS Accounts & Roles for SAML SSO

First we will setup all of your AWS accounts for SAML access with Okta.

- 1 Begin by creating a new AWS app in Okta and select **SAML** from the Single Sign-On tab.
- 2 Open the in-product guide, and perform steps 1 and 2 under the “Connect Okta to a [Single AWS Instance](#)” portion of the guide:
 - a. [\(Single Instance\) Step 1: Configure Okta as your Identity Provider in your AWS account](#)
 - b. [\(Single Instance\) Step 2: Add Okta Identity Provider as a Trusted Source in your AWS Roles](#)
- 3 Do this for all of your AWS accounts and roles that you want to grant users access to – and **ensure that all of your accounts have been set up with the same exact SAML metadata and have been named the same exact name.** Any account with a different SAML provider name or metadata document will not be accessible.

Step 2: Creating AWS Role Groups in AD / LDAP

Once all AWS accounts have been configured for SAML, groups must be created in AD for each AWS role in each account that you want users to have access to. This can be accomplished in a few different ways:

- **Option 1: Script between AWS and AD / LDAP that creates AD groups for each role in each account**
This offers the greatest possibility of automation, but requires coordination between your AWS management teams and AD / LDAP management teams for the script to be configured. In the future, Okta hopes to provide sample scripts to help simplify the setup, but no such scripts will be provided in the initial release of this solution.
- **Option 2: CSV Export from AWS**
If a scripting approach between AWS and AD / LDAP is not a possibility, a lighter weight approach may be to simply export a list of role names for each of your AWS accounts in a CSV that you provide to you AD / LDAP administration teams. From there, they can manage the creation of AWS Role groups

however they see fit without any sort of dependencies or direct integration with your AWS accounts themselves.

- **Option 3: Manual Creation**

Lastly, it is always possible to create AWS Role Groups in AD / LDAP manually. This model is the simplest, however, it will require upkeep as well as ample set up time to create groups in AD / LDAP for each of the roles in each of your accounts.

Regardless, of how you choose to create these AWS Role Specific Groups in your directory, we recommend the following procedure:

- 1 Create a new OU somewhere in your directory so that you can isolate all of your AWS Role Specific groups. This is not required, but recommended in order to make group management simple for your administrators. Potential OU names could be “AWS Role Groups”, “AWS Entitlements”, etc.
- 2 Create AD security groups for each role following a standard syntax. For simplicity, Okta recommends the following syntax.

aws#<account alias>#<role name>#<account #>

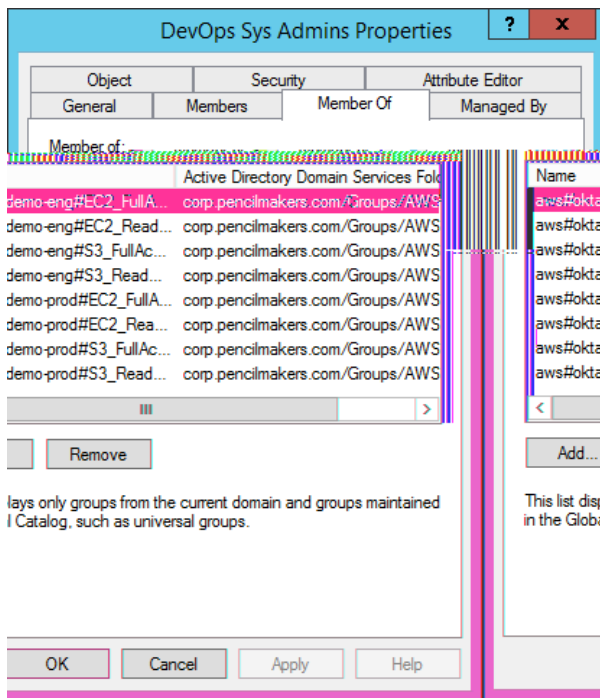
example: **aws#northamerica-production#Tier1_Support#828416469395**

if you prefer to use your own group syntax, then please make sure to include **account alias**, **role name**, and **account #** with **recognizable delimiters** in between each. This will also require you to be able to create a custom regex expression in later steps and therefore should only be done if you are comfortable with these advanced topics.

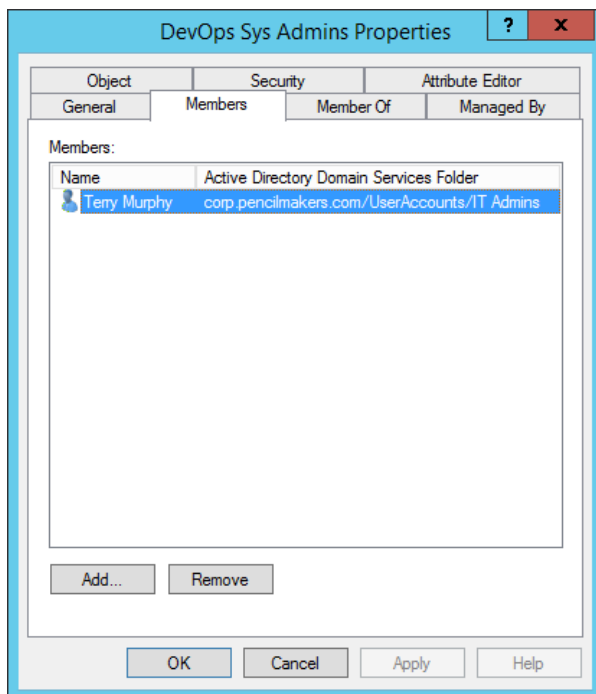
Step 3: Configuring AD / LDAP Management Groups to Map Users to AWS Accounts & Roles

Next, another set of AD / LDAP groups will be created or used to establish a link between sets of users, and the specific AWS accounts and roles they should have access to.

- 1 If you do not already have groups in AD that you want to use to manage the AWS entitlements that different users should have access to, then
 - a. Create another OU in your directory for “AWS Management Groups”. Alternatively, you can place these groups wherever you prefer in your directory – a different OU is recommended to simply aid in ease of administration.
 - b. Create groups for each different user population that requires a different set of AWS roles and accounts. Name these however you see fit – for instance, “Tier 1 AWS Support”, “Database Admins”, “AWS Super Admins”, etc.
- 2 Once you have management groups you would like to use, make each of these groups a **member of** all of the AWS Role Groups that this group should have access to. This establishes a link between the management groups and the entitlements in all of your AWS accounts that group users should have access to. You can add, remove, modify, and audit AWS entitlements from this page for each of your management groups.



- 3 Next, you can begin assigning users directly to the group by making users members of these groups. Similarly, you can add, remove, modify, and audit user membership of each group from this page as well.



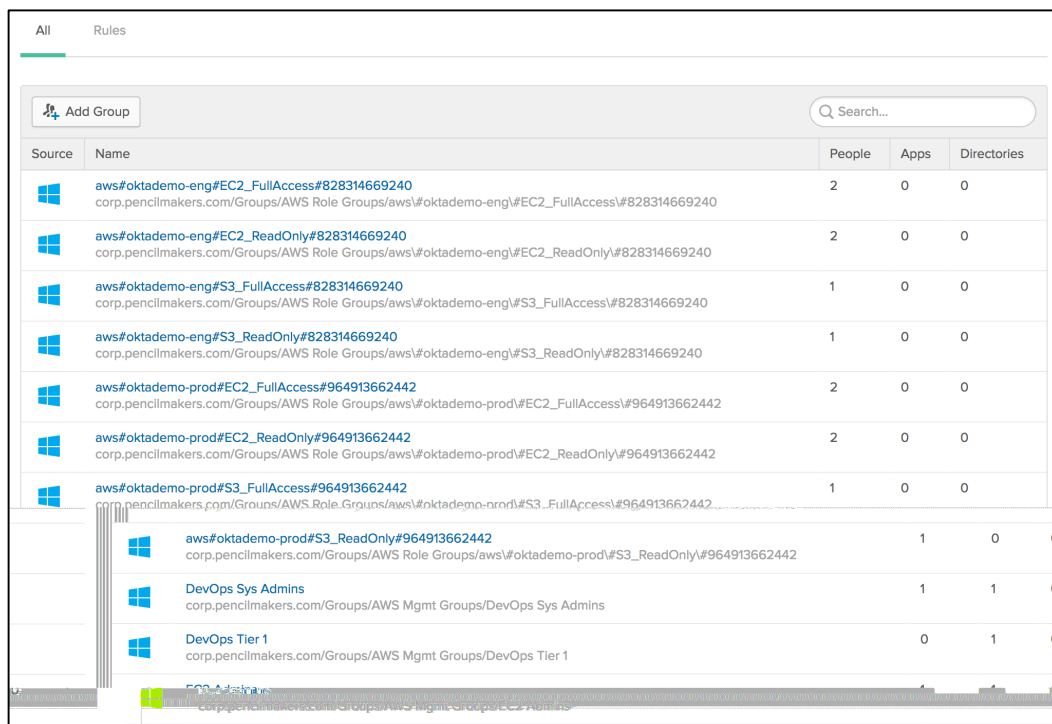
These management groups become the central control point for you to manage & audit user access to different sets of AWS entitlements.

Step 4: Importing AWS Role Groups and Management Groups into Okta

Next, both AWS role groups and management groups need to be imported into Okta and configured for use in the AWS app you configured in Step 1.

Importing these groups is typically done via the Okta AD or LDAP Agent. Instructions on installing the Okta AD / LDAP Agent can be found in product by navigating to **Directory > Directory Integrations**.

Upon completion, you should be able to see both your AWS Role groups and Management groups from the **Groups** page in the Okta Admin Console



The screenshot shows the 'Groups' page in the Okta Admin Console. At the top, there are tabs for 'All' and 'Rules', with 'All' selected. Below the tabs is a search bar and an 'Add Group' button. The main content is a table with columns: 'Source', 'Name', 'People', 'Apps', and 'Directories'. The table lists several groups, including AWS Role Groups (e.g., 'aws#oktademo-eng#EC2_FullAccess#828314669240') and Management Groups (e.g., 'DevOps Sys Admins', 'DevOps Tier 1'). Each group has a corresponding icon (Windows logo for role groups, and a generic icon for management groups) and a detailed name in the 'Name' column. The 'People', 'Apps', and 'Directories' columns show the count of associated entities for each group.

Source	Name	People	Apps	Directories
aws#oktademo-eng#EC2_FullAccess#828314669240	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-eng\#EC2_FullAccess\#828314669240	2	0	0
aws#oktademo-eng#EC2_ReadOnly#828314669240	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-eng\#EC2_ReadOnly\#828314669240	2	0	0
aws#oktademo-eng#S3_FullAccess#828314669240	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-eng\#S3_FullAccess\#828314669240	1	0	0
aws#oktademo-eng#S3_ReadOnly#828314669240	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-eng\#S3_ReadOnly\#828314669240	1	0	0
aws#oktademo-prod#EC2_FullAccess#964913662442	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-prod\#EC2_FullAccess\#964913662442	2	0	0
aws#oktademo-prod#EC2_ReadOnly#964913662442	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-prod\#EC2_ReadOnly\#964913662442	2	0	0
aws#oktademo-prod#S3_FullAccess#964913662442	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-prod\#S3_FullAccess\#964913662442	1	0	0
aws#oktademo-prod#S3_ReadOnly#964913662442	corp.pencilmakers.com/Groups/AWS Role Groups/aws\#oktademo-prod\#S3_ReadOnly\#964913662442	1	0	0
DevOps Sys Admins	corp.pencilmakers.com/Groups/AWS Mgmt Groups/DevOps Sys Admins	1	1	0
DevOps Tier 1	corp.pencilmakers.com/Groups/AWS Mgmt Groups/DevOps Tier 1	0	1	0
EC2 Admins	corp.pencilmakers.com/Groups/AWS Mgmt Groups/EC2 Admins	1	1	0

Once the groups have been imported into Okta, the AWS application you set up in Step 1 must be configured to translate AWS Role group membership into entitlements that AWS can understand syntactically.

- | App Filter | Group Filter | Role Value Pattern |
|---|---|--|
| <p>active_directory</p> <p>Comma separated list of apps to fetch groups from. Can contain app names, app display names, app instance IDs. E.g.: active_directory, acme.com, 00aovrln4R6AFNZeW0g</p> | <p>^aws\#\S+\#(?:[role])\w\-\+)?(?:acc</p> <p>Regex expression to filter AWS related groups and role. E.g.: aws_(?([accountid])d+</p> | <p>arn:aws:iam::\$(accountid):saml-provider/RelInvent.Oktapreview,arn:aws:iam::\$(accountid):saml-provider/OKTA,arn:aws:iam::\$(accountid):role/\$[role]</p> <p>Role Value Pattern. E.g. arn:aws:iam::00aovrln4R6AFNZeW0g:saml-provider/OKTA,arn:aws:iam::00aovrln4R6AFNZeW0g:role/RelInvent.Oktapreview</p> |

- $$^aws\#\S+\#(?{\{role\}}[\w-]+)\#(?{\{accountid\}}\d+)\$$$

If you didn't use the default recommended AWS role group syntax, then you must create a regex expression that properly filters your AWS role groups, and captures the AWS role name and AWS Account ID within two distinct Regex groups named **{{role}}** and **{{accountid}}** respectively.

¥ ;,5\$O2'&5\$C2((51C->.?'4.&,7'-\$H&'>-&')*!'%1,&'\$27'\$001#2-'=O'0\$(-#%&7'<.->.2'>-&'?B2-\$
B1#%')*!'%1,&'5%1#(?D'\$27'-%\$2?,\$-&'?'.-2-1'>-&'(%1(&'?B2-\$K')*!'%&#.%&'?'.2':H-\$L?'
\$??&%-.12'-1'\$,1<'#?&%?'-1';.&<'>-&.%'\$001#2-'\$27'%1,&'?<>&2'>-&B'?'.52'.2E

l>.'?4.&?'>1#7'\$,<\$B?'41,,1<'>.'??'(&0.4.0'?B2-\$KY'

#.,#)*'1#"\$MH#22-5,/14!"#\$%&3%.-A14!.009PQR(S.-A14!.(T#\$!U#.,#)*'1#"\$MH#22-5,/14!.-&!NMH.-&!!

(
N&(\$0&009PQR(S.-A14!.(T#\$!U>'>-&'2\$A&'14'>-&!)!+U'(%1;.7&%'->-\$'B1#'?&-'#('2'\$,,14'
B1#%')*!'\$001#2-'?'.2'!-&('FE'I>&'%&'?-'14'>-&'?-%.25'?>1#7'21-'@&C\$#&9&7B'R'(\$?-&E

#(5<\$]E\$!99)0-\$!'\$!"#\$%2-205>5-(\$31,&<9\$(\$(?5\$!"#\$!<<\$)-\$!\$2
U\$?-,BD'21<'>-\$'>-&')*!'\$((>\$?'@&2'(%1(&%,B'0124.5#%&7'-1'A\$(')*!'%1,&'5%1#(?-1'&2-.,&A&2-'D'?A(
\$??'.52'\$,,14'B1#%')*!'!+\$2\$5&A&2-'6%1#(?-1'>-&'\$((,.0\$-.12'.2':H-\$E'I>.'?<.,'\$#-1A\$-.0\$,B'\$??'.52'\$,,14':
\$((%1(%.\$-&'#?&%?'-1'>-&')*!'\$((D'\$27'>-&'2?-'#0-.12?'B1#01A(,&-&7'0!&,'&2?#%&'>-\$'>-&B'12,B'?&&'>-&'\$((%1(%.\$-&'&2-.,&A&2-'>-&B'?>1#7'>\$;&'\$00&??'-1E



!&-#('?'21<'01A(,&-&i'_&%4B'>-\$'>-&'#?&%?'0\$2'\$00&??'>-&')*!'\$((4%1A'>-&.%'/#?&27\$?>@1\$%7'\$27'
?.5212'.'?&\$A,&??