# SageMaker Studio User Profile Access Control with IAM

Out of box, SageMaker Studio allows all user profiles within a Studio domain to access each other’s user profiles and associated notebooks. Customer wants to add IAM-based access control so that a particular user has access to a particular user profile. Traditional IAM federation provides user access via IAM roles into the AWS console at this customer.

## Permissions

The IAM role, assumed during the federation process, should contain these managed policies

|  |  |  |
| --- | --- | --- |
| Policy | Type | Notes |
| DataScientist | AWS Managed | Or your current baseline permissions |
| AmazonSageMakerFullAccess | AWS Managed |  |
| AmazonSageMakerFeatureStoreAccess | AWS Managed |  |
| SageMakerStudioByEnvVarPolicy | Customer Managed | Controls access based on tag on the user profile |

The policy SageMakerStudioByEnvVarPolicy provides the condition check that adds a DENY if the tag on the user profile does not match the identity. If this policy is not included, all users have access to all Studio profiles in their Studio domain.

### SageMakerStudioByEnvVarPolicy Policy

The SageMakerStudioByEnvVarPolicy policy adds an explicit deny to users that don’t have a tag matching their federated credentials.

{

    "Version": "2012-10-17",

    "Statement": [{

        "Sid": "DenyUnlessTag",

        "Effect": "Deny",

        "Action": "sagemaker:CreatePresignedDomainUrl",

        "Resource": "\*",

        "Condition": {

            "StringNotLike": {

                "sagemaker:ResourceTag/studiouserid": "${aws:userid}"

            }

        }

    }]

}

## Tags

As of March 2021, the AWS console does not surface tags for SageMaker user profiles. To add, change, or list tags requires use of the AWS SDK or CLI. This section provides examples.

Note, this content contains rudimentary shell scripts which have hard-coded variables. These examples are not intended for production use.

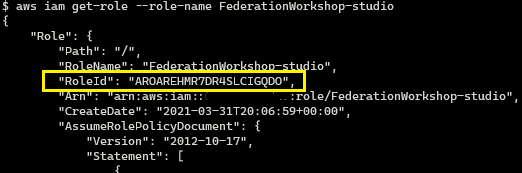
### Tag content

IAM includes the [aws:userid](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html) variable for all signed requests, including those from SAML-federated users.

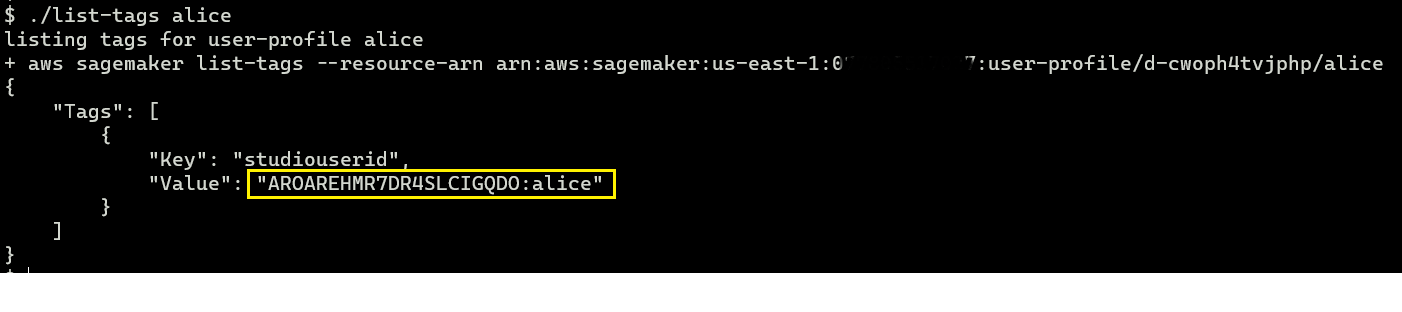
Aws:userid contains identity information formatted as **<RoleId >:<ActiveDirectoryUserName>**, where RoleId is the federation role. To enable access to a notebook/user profile, we need to tag that profile in this format.

To find the RoleId for a particular role, use on of these AWS CLI commands. Note, replace *federationRole* with the name of the role an AD credential federate into.

aws iam get-role --role-name <federationRole>



In the example case, above, the user federates into an IAM role named “FederationWorkshop-studio”. The user profile named alice should be tagged with AROAREHMR7DR4SLCIGQDO:alice to allow alice to access her profile.



### Tagging while creating profiles

To add a tag to a profile during profile creation:

aws sagemaker create-user-profile \

  --domain-id ${domainid} \

  --user-profile-name $user \

  --tags Key=studiouserid,Value="${user}" \

  --user-settings ExecutionRole=arn:aws:iam::$acct:role/$executionRole

|  |  |
| --- | --- |
| $domain-id | SageMaker Studio domain, similar to d-cwoph4tvjphp |
| $user | aws:userid for this user, similar to: AROAREHMR7DR4SLCIGQDO:alice |
| $acct | AWS account number |
| studiouserid | Tag name used by the IAM policy, do not change. |
| $executionRole | IAM role which is assumed by the user profile. Typically contains AmazonSageMakerFullAccess policy. |

Here is a sample script for creating profiles:

#!/bin/bash

if [ "$#" -ne 1 ]; then

   echo "usage: $0 role"

   exit

fi

user=$1

echo adding user $user

domainid=d-cwoph4tvjphp

set -x

aws sagemaker create-user-profile \

  --domain-id ${domainid} \

  --user-profile-name $user \

  --tags Key=studiouserid,Value="${user}" \

  --user-settings ExecutionRole=arn:aws:iam::111222333444:role/BayerSageMakerExecutionRole

### Tagging existing profiles

This script adds a tag to an existing profile:

#!/bin/bash

if [ "$#" -ne 1 ]; then

   echo "usage: $0 <user>"

   exit

fi

user=$1

# SageMaker Studio Domain ID

domain=d-cwoph4tvjphp

account=111222333444

region=us-east-1

# IAM Federation Role

federatedRole=FederationWorkshop-studio

RoleId=$(aws iam get-role --role-name $federatedRole | jq -r '.Role.RoleId')

# the tag which matches ${aws:userid}

tag=$RoleId:$user

# ARN which matches the user profile

arn=arn:aws:sagemaker:$region:$account:user-profile/$domain/$user

echo adding tag $tag to $user \($arn\)

aws sagemaker add-tags --resource-arn "$arn" --tags "[{ \"Key\": \"studiouserid\", \"Value\": \"$tag\"}]"

### List user profile tags

Tags are not yet surfaced on the console, so we will need to use the CLI, as follows:

#!/bin/bash

if [ "$#" -ne 1 ]; then

   echo "usage: $0 user"

   exit

fi

user=$1

domain=d-cwoph4tvjphp

region=us-east-1

echo listing tags for user-profile $user

aws sagemaker list-tags --resource-arn "arn:aws:sagemaker:$region:111222333444:user-profile/$domain/$user"

## Summary

You can control access to user profiles with a policy that adds an explicit deny by adding a tag to the SageMaker user profile and adding a policy to the user’s IAM console role.

## Ideas for further work:

1. Add policy for sharing notebooks across users
2. remove the ALLOW policy from AmazonSageMakerFullAccess and add it to SageMakerStudioByEnvVarPolicy
3. Add policy to prevent users from changing the tag on their user profile
4. harden sample scripts for production use