Cyberark AWS Console with STS Connection Component – Example Configuration

This is just a sample configuration and CyberArk does not provide any official recommendation on how to configure your AWS environment. In addition, the steps provided in this Knowledge Article are subject to change and for the latest steps please reference CyberArk's Official Documentation for this integration below:

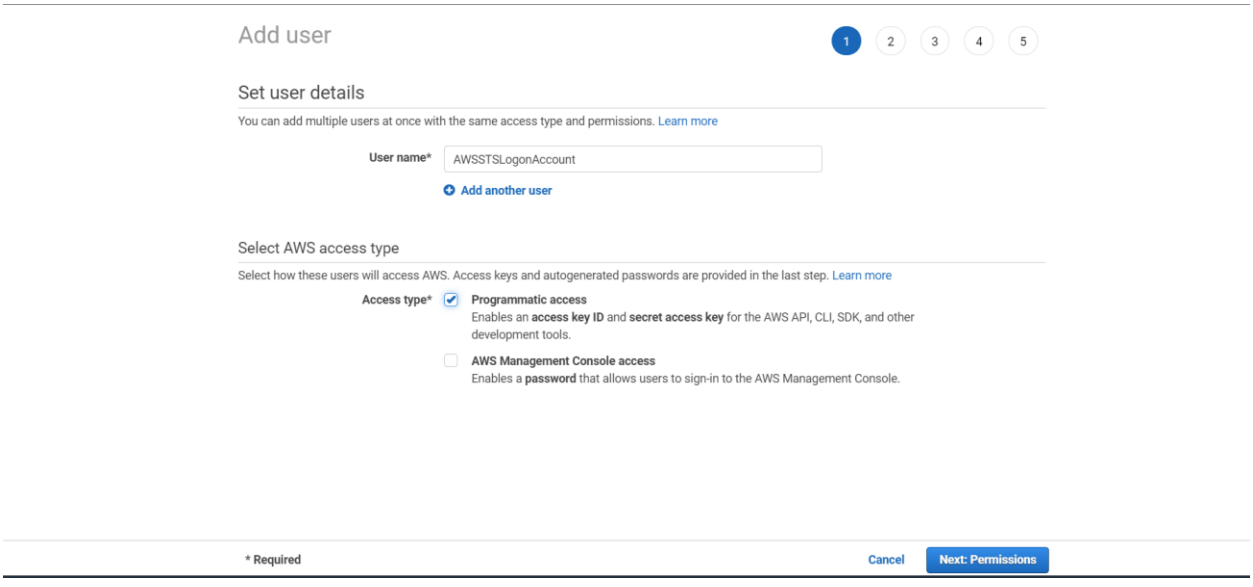
- AWS Cloud Services Management

In this short step by step guide, we will see how Cyberark STS Plugin can be used to provide secure AWS console access to different group of privilege users. The STS plugin is one of the best options customers should leverage when managing access to AWS with CyberArk due to following reasons:

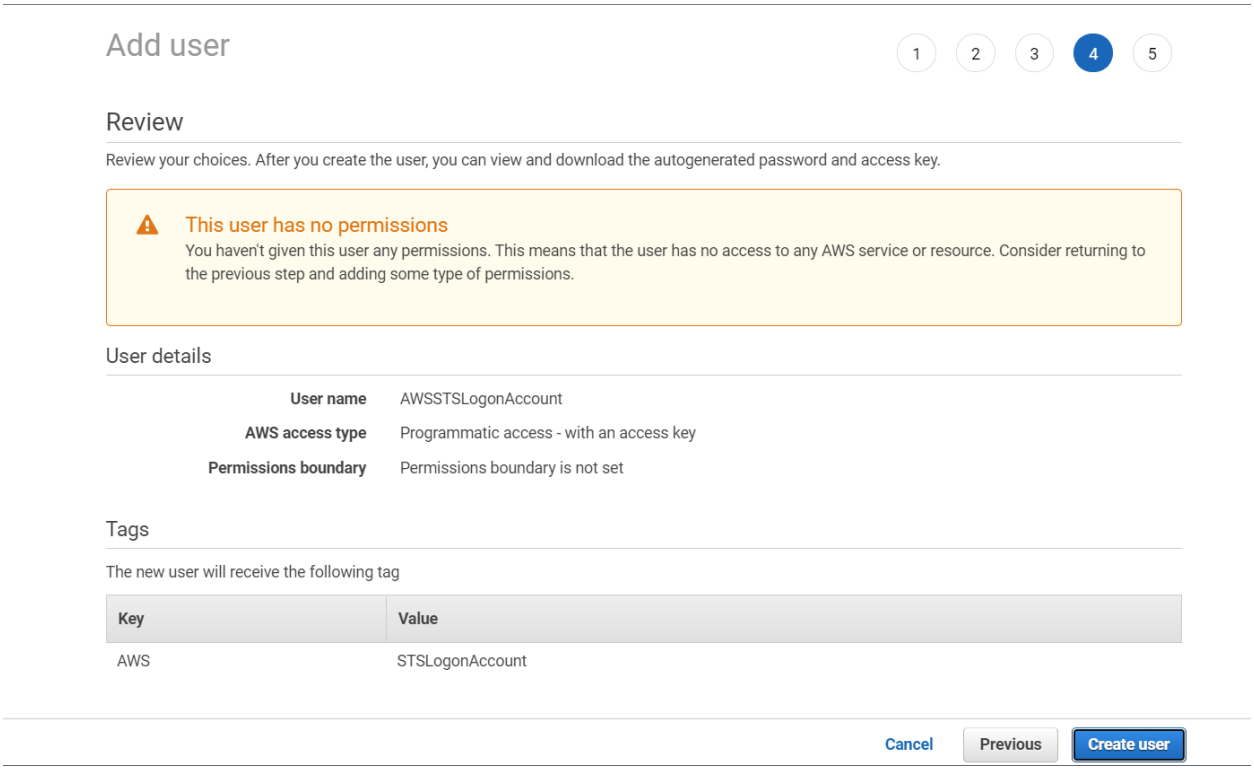
- No need to create multiple user identity on AWS IAM
- No credentials involved (Cyberark STS plugin involves the use of logon account)
- Provide JIT access to any of the given privilege users which will be tied to the granular roles or have specific set of permissions

We have divided the entire configuration into 4 main stages.

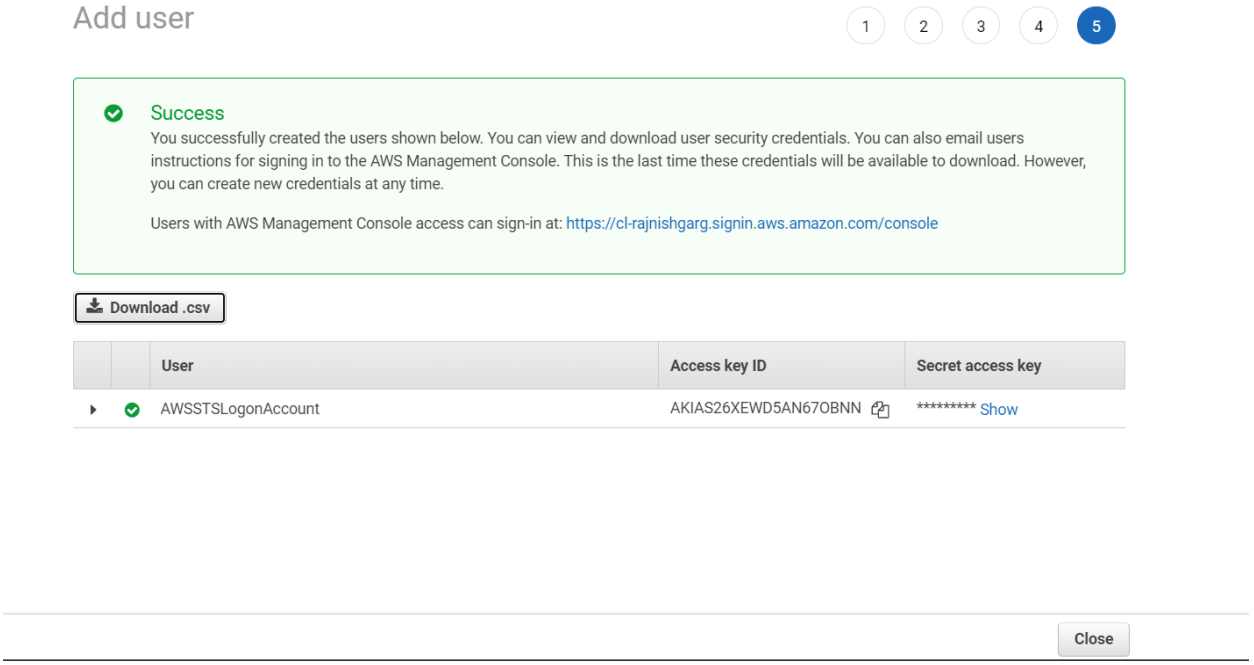
1. **Configure the Pre-Requisites for the CyberArk STS Integration noted in our CyberArk Documentation:**
    a. Prerequisites
    b. Install Internet Explorer
    c. Configure Applocker
2. **Creation and On-Boarding of the STS logon account**
    a. Log into the AWS Console, go to AWS IAM Service, and then click on to add new user



    b. Proceed without assigning any policy or permissions.



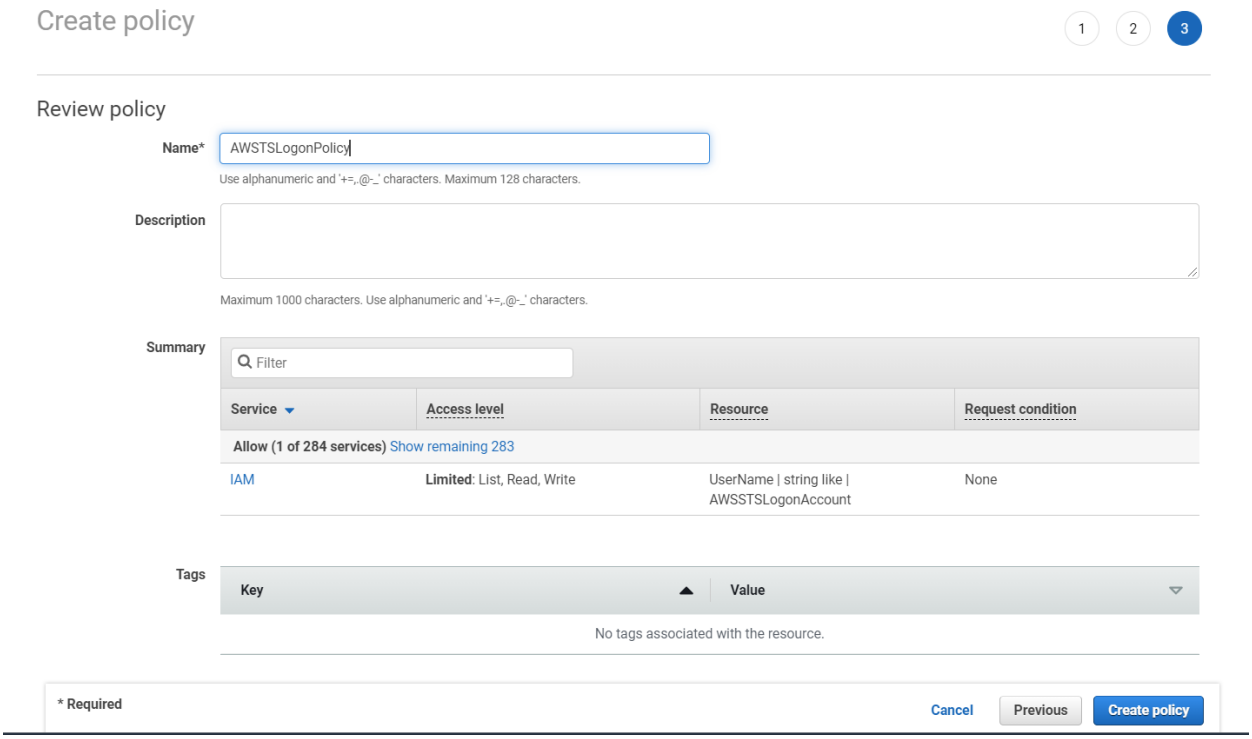
c. Click on create user and note the values of the following fields:
   I. User (You will specify this field)
   II. Access key ID
   III. Secret Access Key



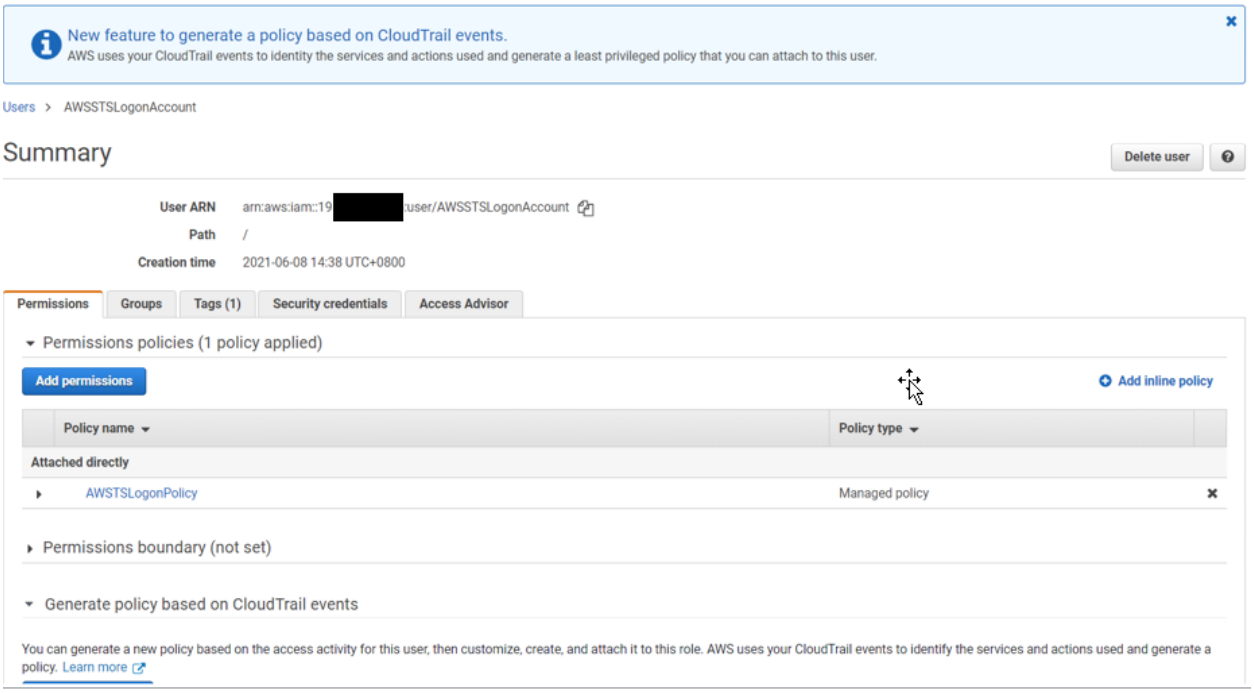
d. We will create new policy where we will assign enough permission to the AWSSTSLogonAccount to verify and rotate the access keys as per the Organization security policy.
e. Go to IAM policies and click create Policy. We are restricting the following permissions only to our “AWSSTSLogonAccount” only via the “Resource:” section below. In addition, we are providing only the following 4 permissions what we require for the sample configuration:
    I. iam:DeleteAccessKey
    II. iam:GetAccessKeyLastUsed
    III. iam:CreateAccessKey
    IV. iam:ListAccessKeys



f. Assign a policy name. In this configuration it is named “AWSTSLogonPolicy”



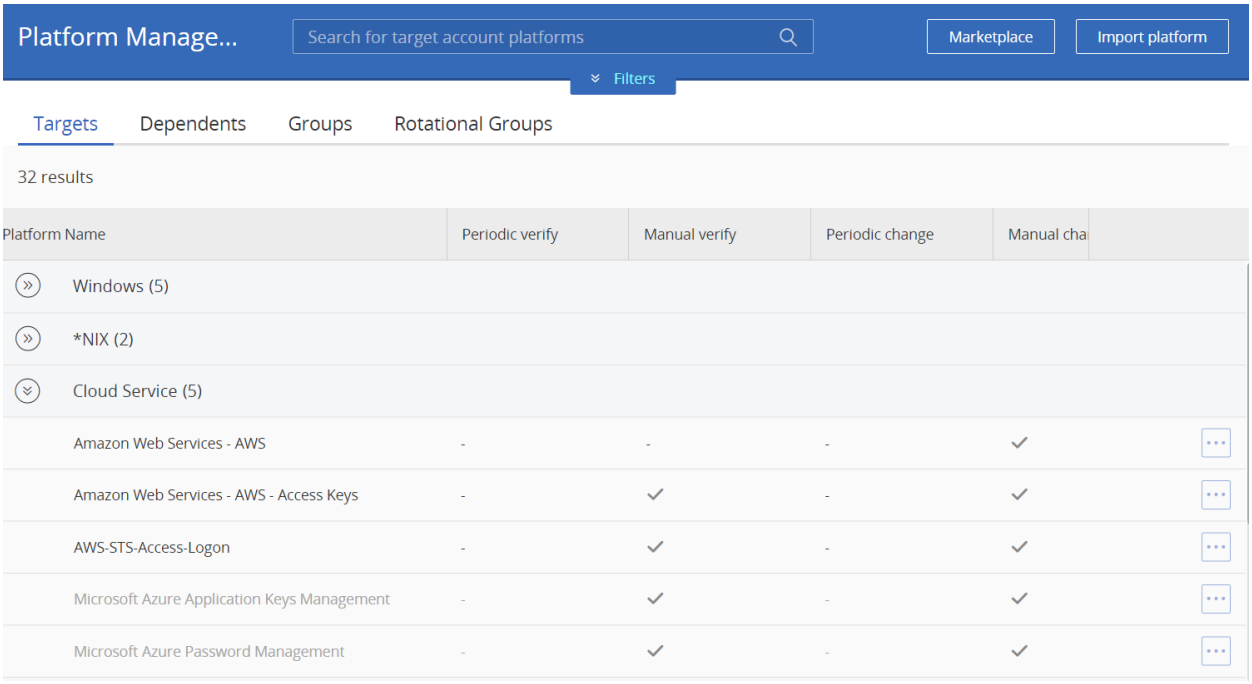
g. Assign the policy created in previous step to “AWSSTSLogonAccount”



h. All steps have been completed on the AWS side and now we will on-board the “AWSSTSLogonAccount” into Cyberark. We shall be on-boarding this account manually, but it can be on-boarded using automated functionalities of CyberArk.

i. Login into Cyberark PVWA using the appropriate credentials who is authorized to on-board new accounts

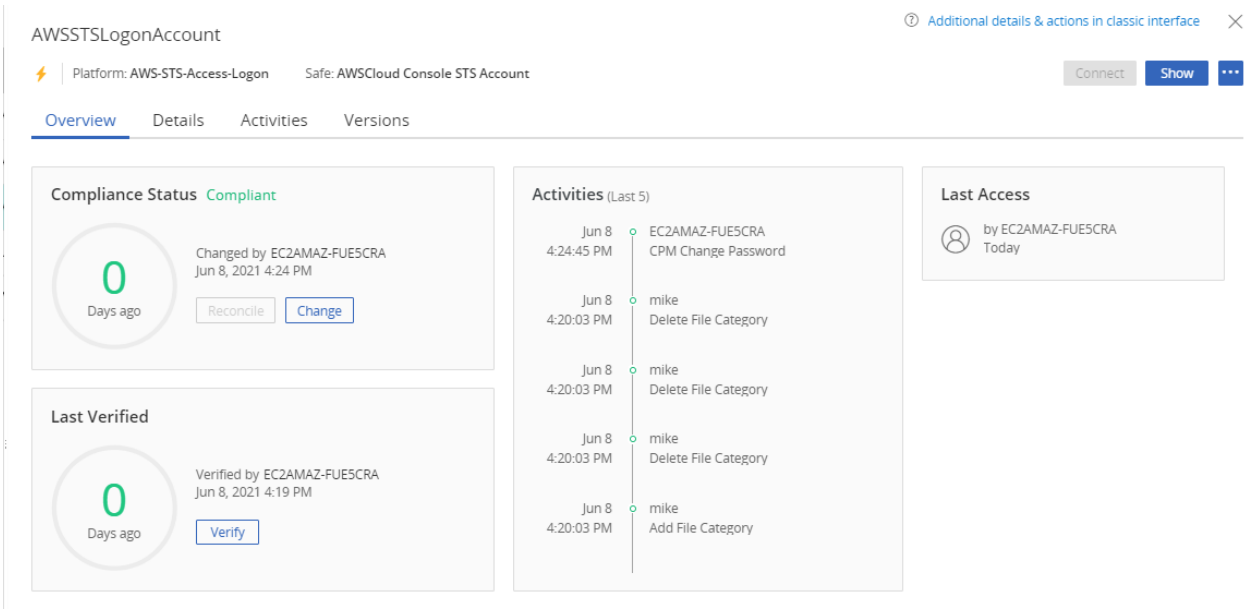
j. Go to Platform Management and duplicate the “Amazon Web Services-AWS-Access keys” platform under the Cloud Service section.
k. In this example, the duplicate of the AWS-Access Keys platform was named “AWS-STS-Access-Logon Account”. If the duplicated platform is not enabled, please enable the duplicated platform.



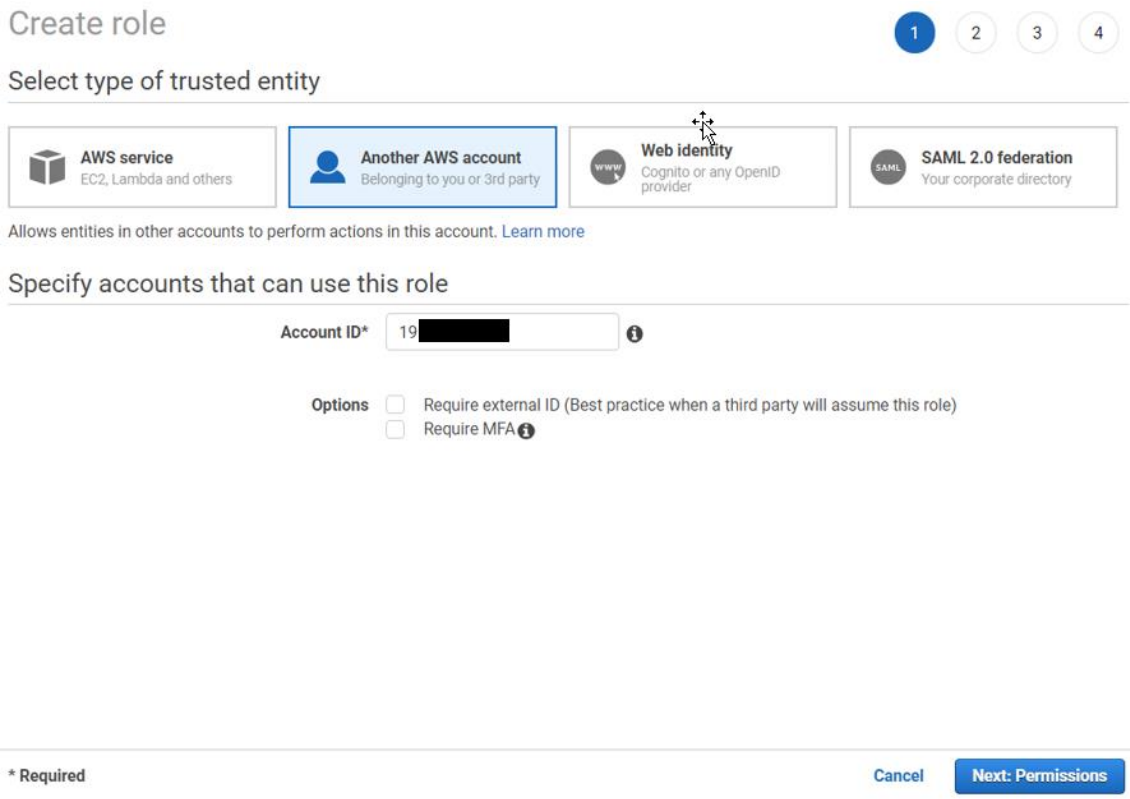
l. Create a new safe to store the “AWSSTSLogonAccount.” In this example, we created a safe called “AWSCloud Console STS account” and assigned the appropriate members.
m. Onboard the “AWSSTSLogonAccount” to the “AWSCloud Console STS account” safe and enter parameters as shown below.

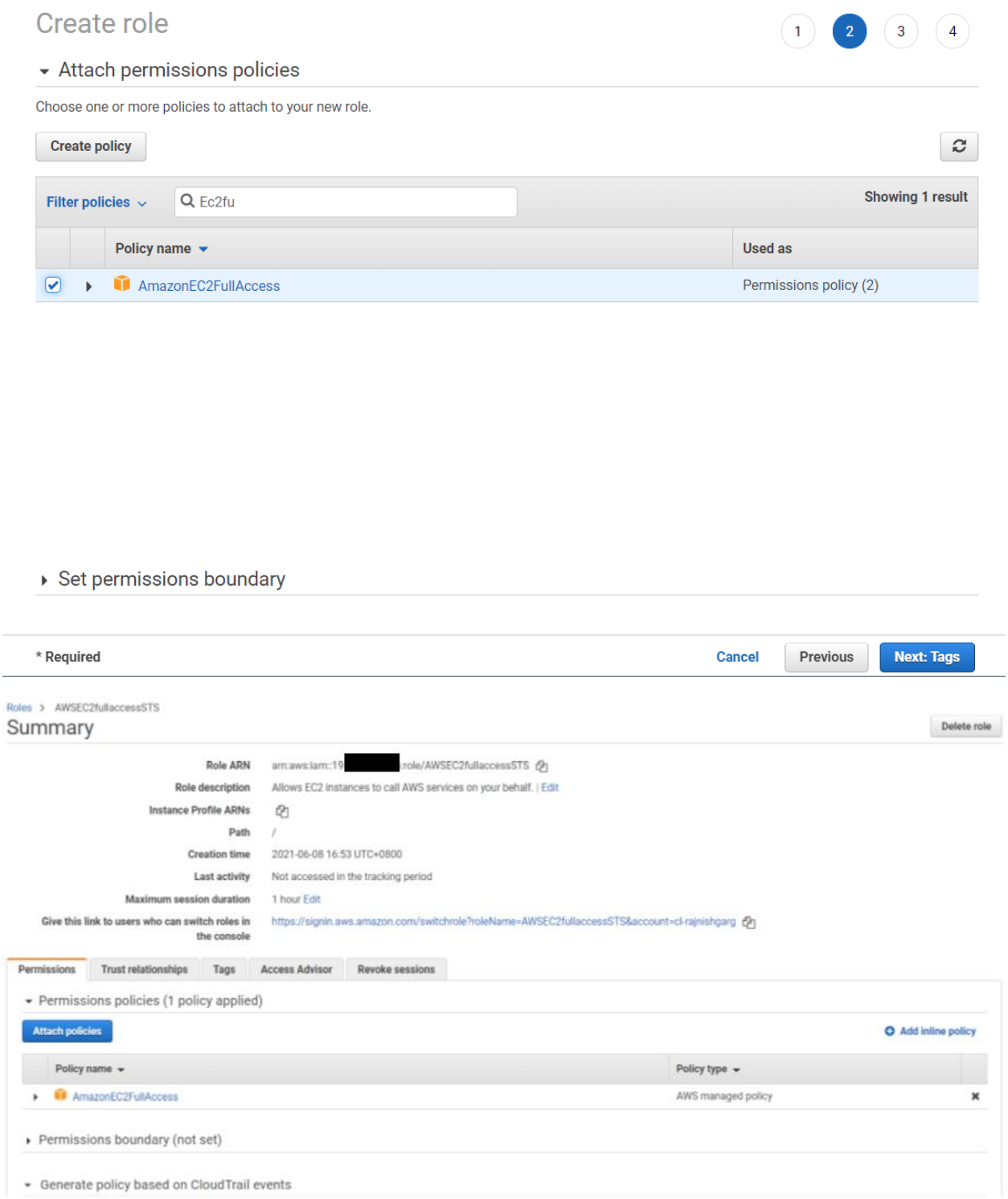
S.No.		
1.	System Type	Cloud
2.	Select Platform	AWS-STS-Access-Logon
3.	Select Safe	AWS Cloud Console STS Account
4.	IAM Username	AWSSTSLogonAccount
5.	AWS Access Key Secret	XXXX (You have noted down this while configuring new user on AWS)
6.	AWS Access Key ID	XXXX (You have noted down this while configuring new user on AWS)
7.	AWS Account Number	Key in your AWS Account number
8.	AWS Account Alias	AWS STS Logon Account (You can choose anything)

n. Once the “AWSSTSLogonAccount” has been added, we will perform a password verification and change to verify CPM functionality with the account. Once the CPM testing is completed, then we will move to the next step.

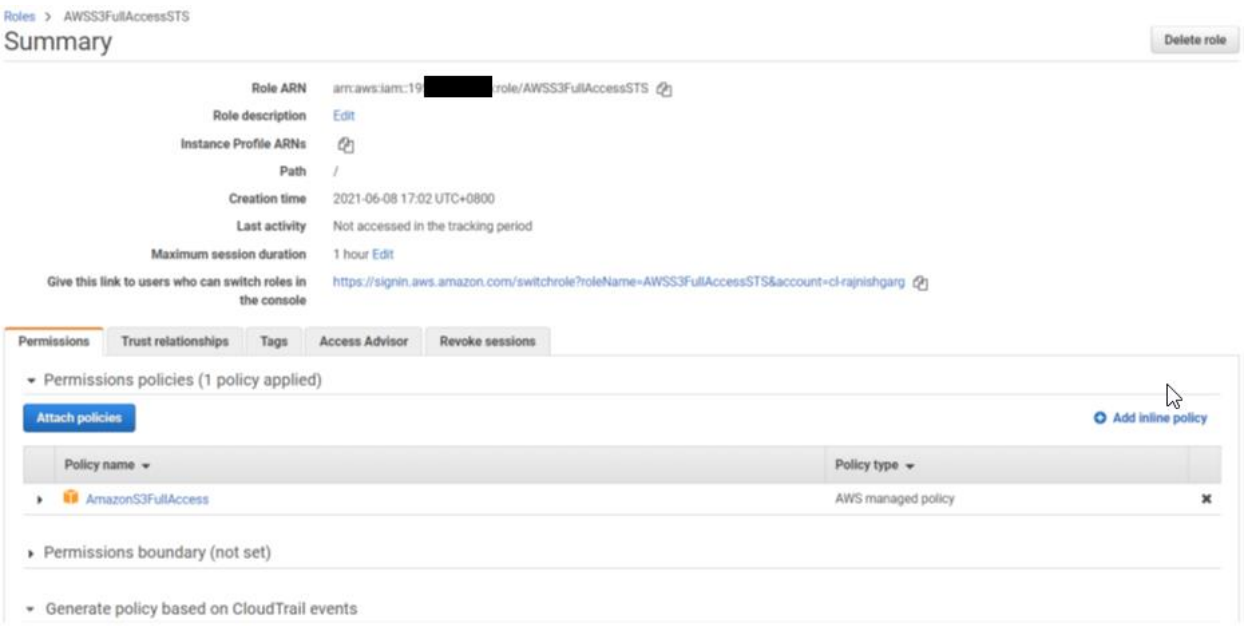


3. **Creation of roles in AWS and assigning of those roles within CyberArk**
    a. Create 2 different roles for the following AWS permissions:
        I. AmazonEC2FullAccess
        II. AmazonS3FullAccess
    b. Here are the steps on how to setup the “AWSEC2fullaccessSTS” role where any entity within your account can use this role.

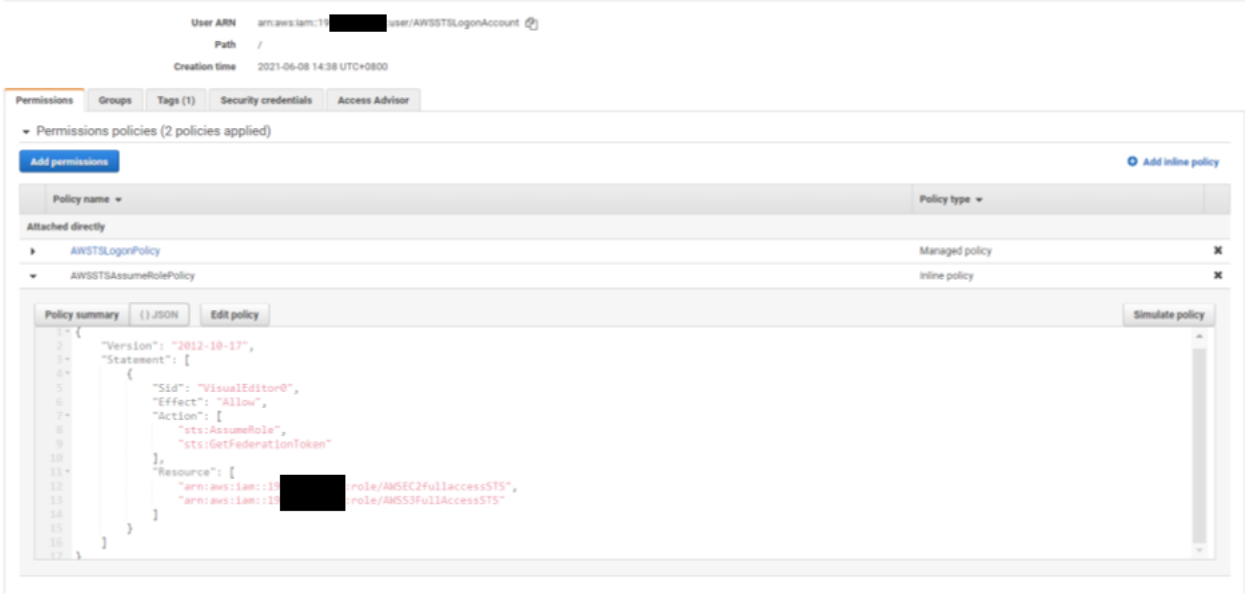




c. The screenshot below shows how the “AWSS3FullAccessSTS” role should be configured:



d. Once the roles have been created, then we will configure the policy applied to our "AWSSTSLogonAccount" to be able to assume these 2 roles.We will ensure the permissions to assume the new roles will be granted to our "AWSSTSLogonAccount" via the "Resource" section below and will grant the following permissions to the user to assume the roles:
    I. sts:AssumeRole
    II. sts:GetFederationToken



e. The configuration has been completed on AWS side has now been completed. Now it's the time to setup the appropriate configuration within CyberArk.
f. Log into the CyberArk PVWA again and make sure that "Amazon Web Services-AWS" platform is enabled under the Cloud Service section.

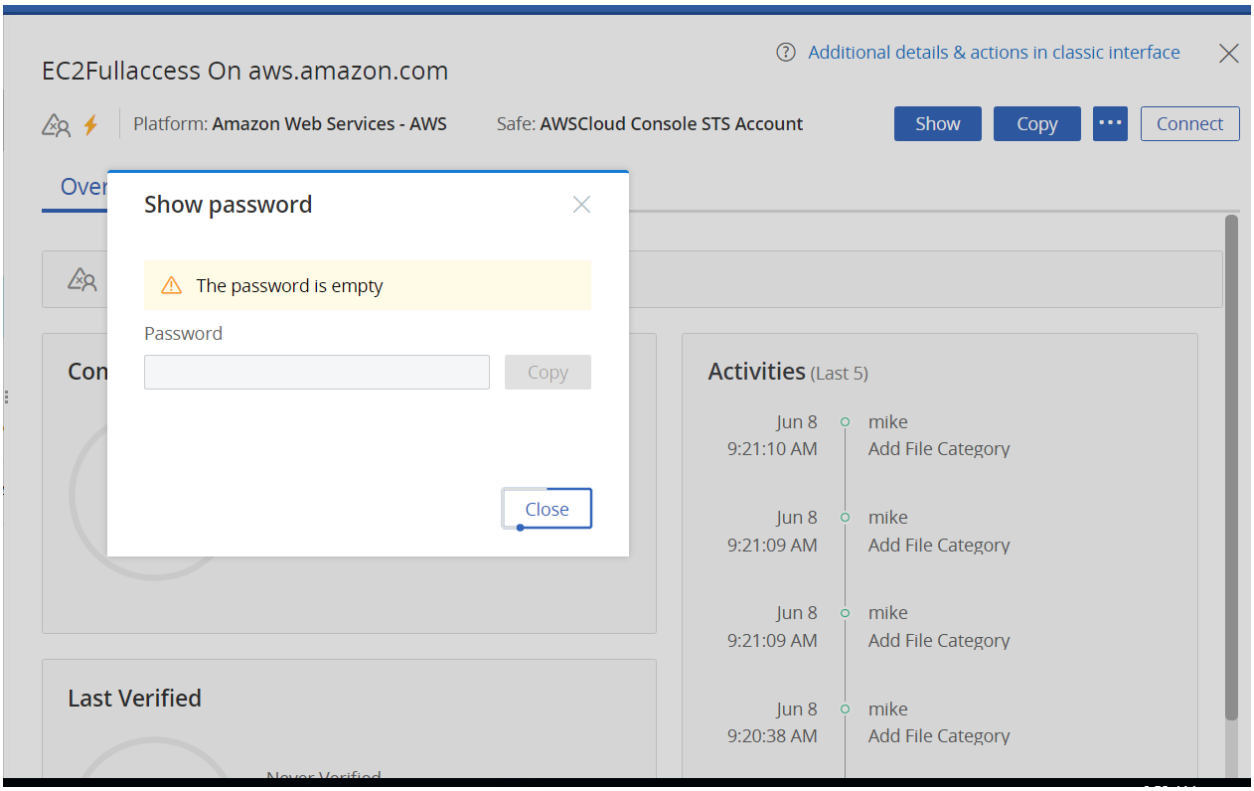
g. CyberArk Best Practice recommends creating a new safe to have better role-based access control for storing the AWS Role Accounts. However, in the configuration below we shall leverage the same safe we used before to store AWS STS Logon Account.
h. Add the "AWSEC2fullaccessSTS" role account and then enter the parameters as mentioned below.

S.No.	Parameters	Settings
1.	System Type	Cloud
2.	Select Platform	Amazon Web Services-AWS
3.	Select Safe	AWS Cloud Console STS Account
4.	IAM Username	EC2Fullaccess (Can be anything)
5.	AWS Role	arn:aws:iam::19XXXX:role/AWSEC2fullaccessSTS (This is how it should look like)
6.	AWS Account Number	Key in your AWS Account number
7.	AWS Account Alias	EC2 FullAccess (You can choose anything)
8.	Associate Logon Account	AWS STS Logon Account (Created in Step 1)

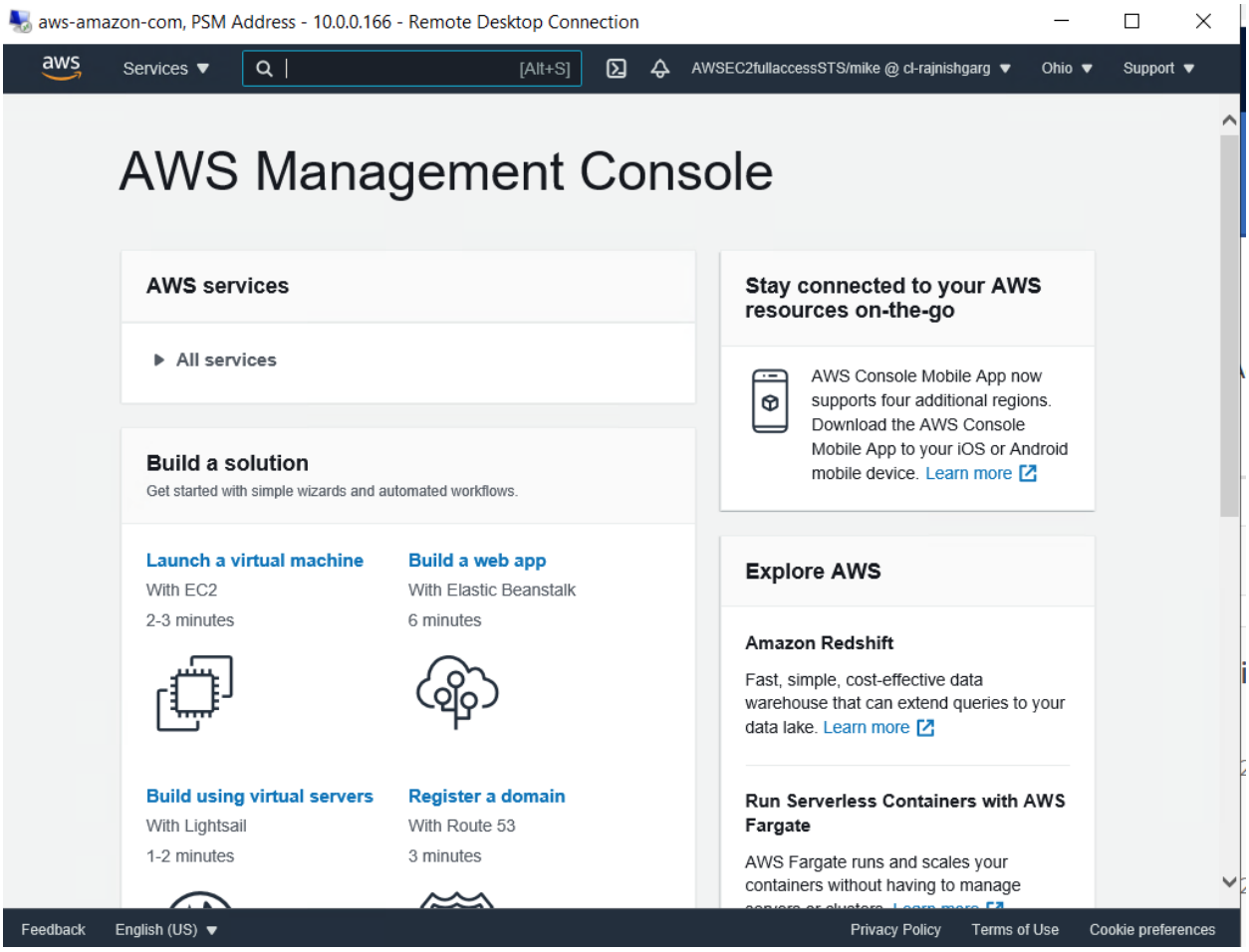
i. Similarly add the "AWSS3fullaccessSTS" role account and then enter the parameters as mentioned below.

S.No.	Parameters	Settings
1.	System Type	Cloud
2.	Select Platform	Amazon Web Services-AWS
3.	Select Safe	AWS Cloud Console STS Account
4.	IAM Username	S3Fullaccess (Can be anything)
5.	AWS Role	arn:aws:iam::19XXXX:role/AWSS3fullaccessSTS (This is how it should look like)
6.	AWS Account Number	Key in your AWS Account number
7.	AWS Account Alias	S3 FullAccess (You can choose anything)
8.	Associate Logon Account	AWS STS Logon Account (Created in Step 1)

4. **Authorized users with access to these credentials in CyberArk can click on the connect button for the given configured EC2FullAccess or S3FullAccess role account.**
   a. You will see that there is no password configured because it is not required for the STS access with the role accounts



b. Once you click on connect button, then an RDP file will be downloaded to your workstation. The user will be able to access to the AWS Console via Cyberark jumphost when they open the RDP file. Below is an example of how the session will appear form within the RDP file:


aws-amazon-com, PSM Address - 10.0.0.166 - Remote Desktop Connection
aws
Services
[Alt+S]
AWSEC2fullaccessSTS/mike @ cl-rajnishgarg
Ohio
Support
AWS Management Console
AWS services
All services
Build a solution
Get started with simple wizards and automated workflows.
Launch a virtual machine
With EC2
2-3 minutes
Build a web app
With Elastic Beanstalk
6 minutes
Build using virtual servers
With Lightsail
1-2 minutes
Register a domain
With Route 53
3 minutes
Stay connected to your AWS resources on-the-go
AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more
Explore AWS
Amazon Redshift
Fast, simple, cost-effective data warehouse that can extend queries to your data lake. Learn more
Run Serverless Containers with AWS Fargate
AWS Fargate runs and scales your containers without having to manage
Feedback
English (US)
Privacy Policy
Terms of Use
Cookie preferences