AWS IAM Key Rotation Runbook

Last updated: 24 June 2022
Version: 3.0

Laura Seletos, CISSP
Senior Security Consultant
AWS WWPS Professional Services
E: lseleto@amazon.com | **M:** +1.727.271.3205

Arvind Patel
Senior Customer Delivery Architect
AWS WWPS Professional Services
E: ptearvi@amazon.com | **M:** +1.703.828.7728

Table of Contents

Document Control

Author	Version	Date	Update Notes
Laura Seletos	1	04/06/2021	Initial document version
Laura Seletos	2	11/04/2021	Updated document for version 2 (new diagrams, troubleshooting, and unit testing). Version 2 was re-architected for scale, centralizing all main resources into a single, centralized account with assumed roles, allowing for cross-account access.
Arvind Patel	3	06/24/2022	This version includes the following changes 1. Option for running Lambda in VPC 2. Template for creating VPC Endpoints (required for Lambda in VPC) 3. Option for storing secret manager in central account 4. Option for replication region for credentials 5. SES email credentials are stored in SSM parameters and pulled at run time.

1.0 Introduction

This document is the runbook on how to deploy, configured, validate, and troubleshoot the Automated AWS IAM Key Rotation solution.

This runbook will walk you through the AWS CloudFormation template setup. This template will create a mechanism to scan daily, and automatically rotate your AWS IAM user Access Keys every 90 days and store the new Access Keys in a secret inside AWS Secrets Manager. An AWS SES notification will be sent to alert of the rotation. 10 days later, the old Access Keys will be disabled. And 10 days after that, deleted. This gives the user time to implement the new Access Keys in their applications.

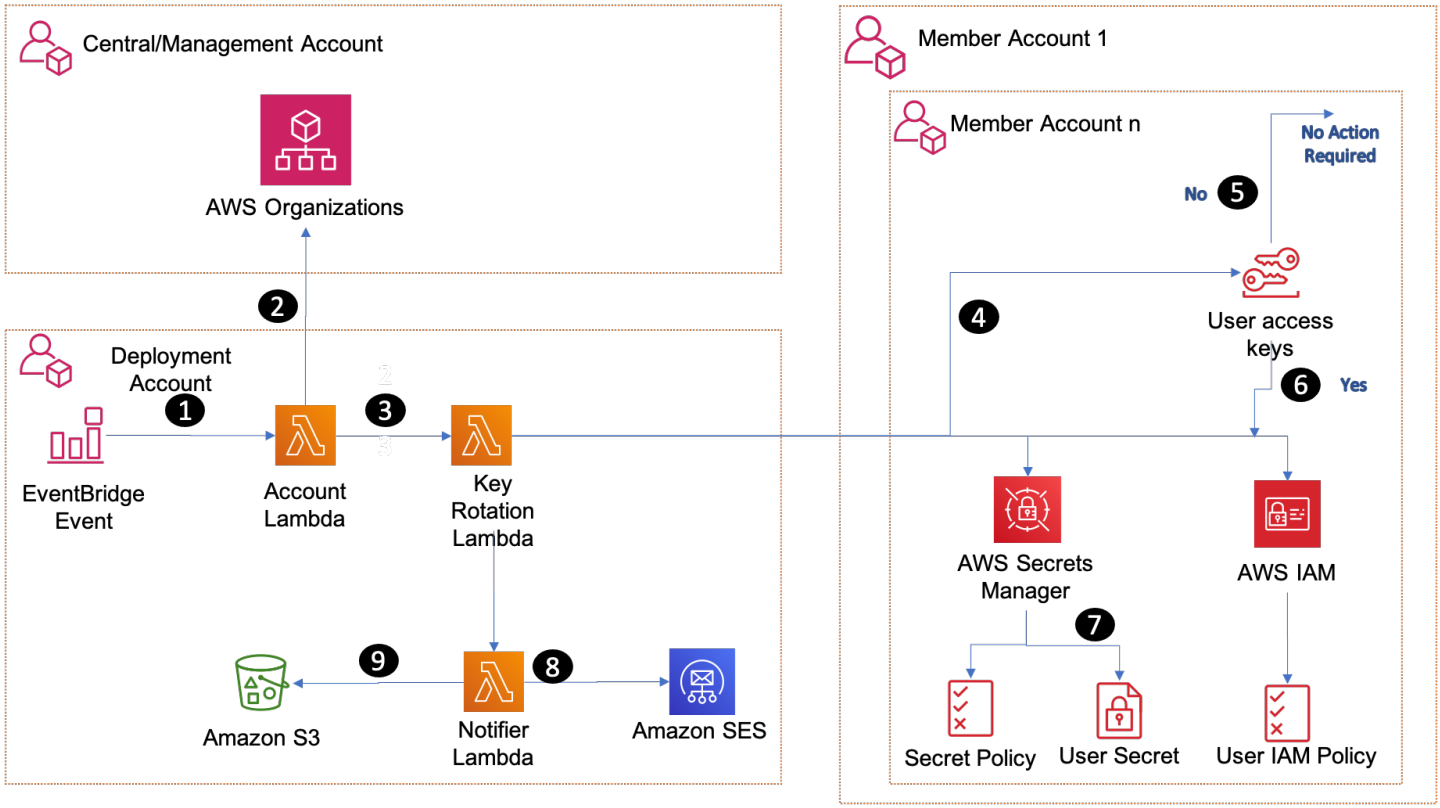
This document covers Config Rules for the following Security Audit Findings:

- 'Lack of Key Rotation (Active)'

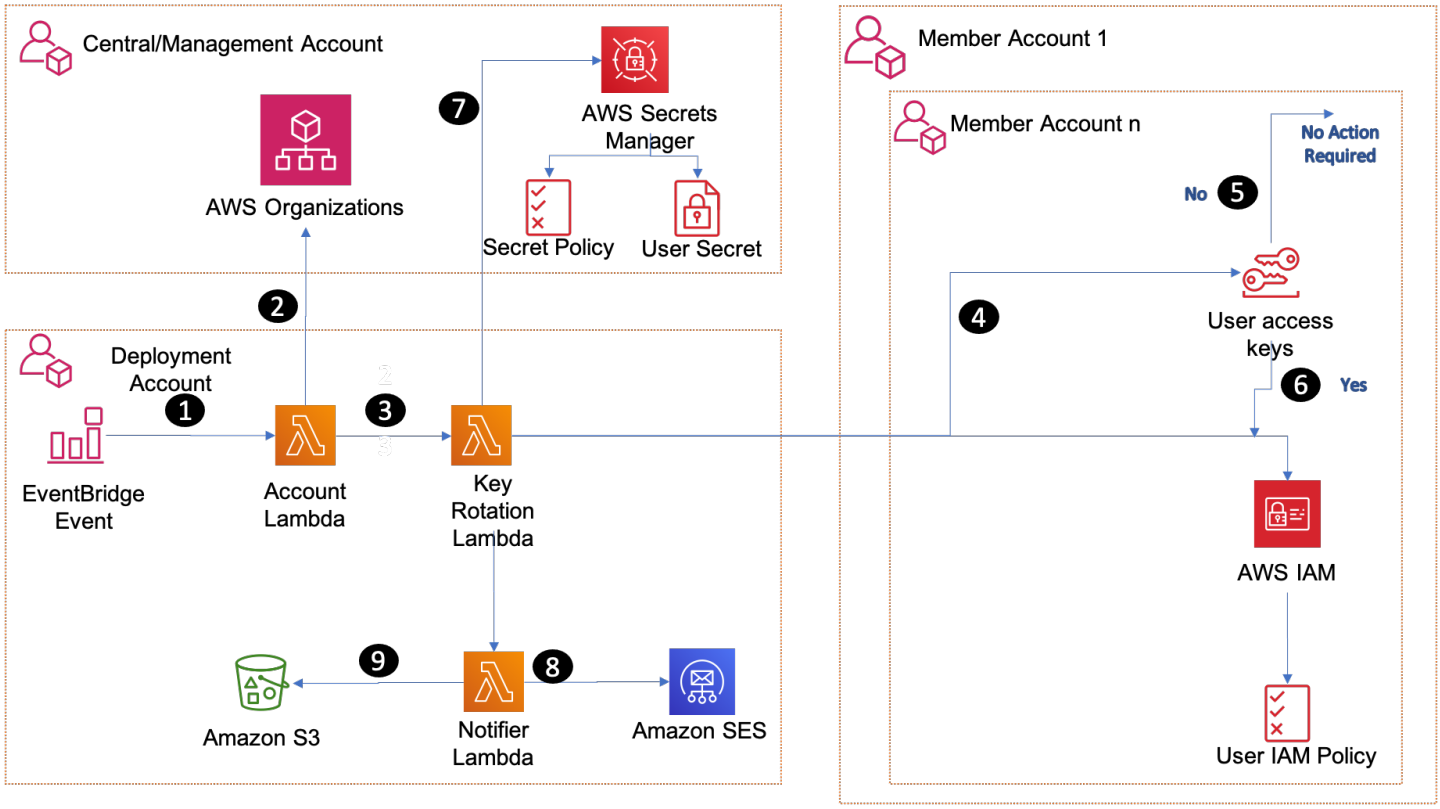
2.0 Architecture

This section covers a mechanism used to scan daily, and automatically rotate your AWS IAM user Access Keys every 90 days and store the new Access Keys in a secret inside AWS Secrets Manager. An AWS SNS notification will be sent to alert of the rotation. 10 days later, the old Access Keys will be disabled. And 10 days after that, deleted. This gives the user time to implement the new Access Keys in their applications.

2.1.1 Option 1: Store the credentials in the AWS Secrets Manager of the member account



2.1.2 Option 2: Store the credentials in the AWS Secrets Manager of the central account



***Note**: The Lambda Function in the 'Main Deployment Account' assumes a local role, in the individual AWS Account(s), that allows it to facilitate localized IAM key rotation actions (i.e. violation detection, rotation, secret creation, etc.).*

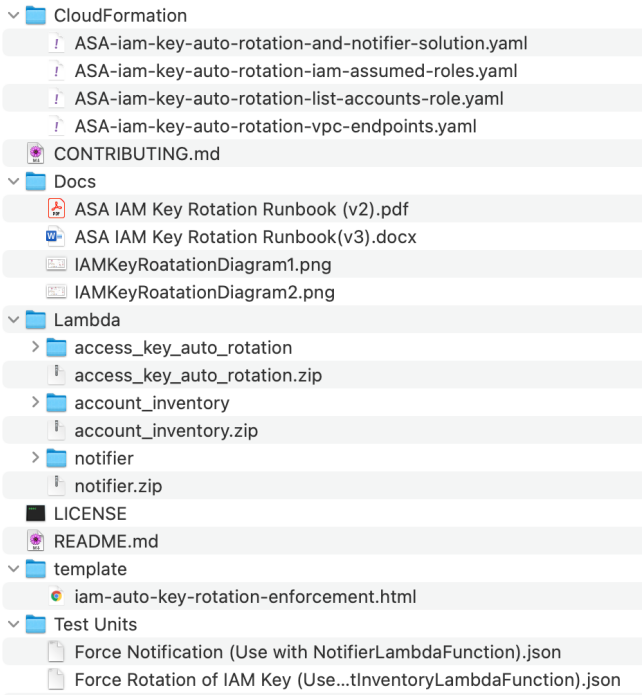
1. Once every 24 hours, the CloudWatch Event will trigger the 'ASA-Account-Inventory' Lambda Function.
2. The 'ASA-Account-Inventory' Lambda Function will list all AWS Account within AWS Organizations, capturing Account ID and Account Email.
3. For each Account ID, the Lambda 'ASA-IAM-Access-Key-Rotation-Function' executes and assumes a role in the target account, scanning every IAM user in the account's Access Keys, checking for creation date.
    - The following are supported key actions:
        - UNUSED_EXPIRED_KEY = 'Expired key has never been used.'
        - EXPIRED_ACTIVE_KEY = 'Active key has expired.'
        - FORCED_ROTATION = 'Forced active key rotation.'
        - EXPIRED_ACTIVE_KEY_CONFLICT_LRU = 'Expired active key with conflict, least recently used.'
        - EXPIRED_INACTIVE_KEY_CONFLICT = 'Expired key with conflict, already inactive.'
        - FORCED_ROTATION_CONFLICT_LRU = 'Forced active key rotation with conflict, least recently used.'
        - FORCED_INACTIVE_KEY_CONFLICT = 'Forced rotation with conflict, already inactive.'
        - INSTALL_GRACE_PERIOD_END = 'Installation grace period has ended.'

- RECOVER_GRACE_PERIOD_END = 'Recovery grace period has ended.'
- KEY_PENDING_ROTATION = 'Key will be rotated soon.'
- KEY_PENDING_DEACTIVATION = 'Key will be deactivated soon, please install new key.'
- KEY_PENDING_DELETION = 'Key will be permanently deleted soon, please validate new key.'
- KEY_PENDING_EXPIRATION_CONFLICT = 'Key will expire soon, cannot be rotated due to presence of other key.'
- KEY_PENDING_DELETION_CONFLICT = 'Key will be permanently deleted soon, due to conflict.'
- UNUSED_KEY_PENDING_DELETION = 'Key will be permanently deleted soon, key is about to expire and has never been used.'

4. If there are IAM users in the IAM group 'IAMKeyRotationExemptionGroup', those users will not be evaluated.
5. If there are Access Keys, outside of the exemption IAM group, newer than 90 days old, or no Access Keys exist, the function exits.
6. If there are Access Keys, outside of the exemption IAM group, that need rotation, the function will create a new Access Key pair and either create a new Secret named after the user (*in the event it's the first time for rotation*), or update the Secret with the new Access Key pair. The secret will be stored in the AWS Secrets Manager either in Central account or respective account based on the option selected while deploying the solution.
7. It will then attach an IAM policy to the user allowing access to the secret (*if it's the first time, if not, it will be ignored*).
8. It will attach a resource policy to the secret, allowing only the specific user access (if it's the first time, if not, it will be ignored).
9. Upon any creation, deactivation, deletion actions on an IAM access key, the 'ASA-IAM-Access-Key-Rotation-Function' Lambda will trigger the 'ASA-Notifier' Lambda Function.
10. The 'ASA-Notifier' Lambda Function will reach out to the S3 Bucket to pull the customizable email template and facilitate sending an email, via Amazon Simple Email Service (SES), to the email associate to the AWS Organization's Account ID.

3.0 Required Files

Project files included in the zip:



4.0 Deployment

This CloudFormation Template will deploy all remediation artifacts discussed in this runbook. Please follow the steps in the sequence.

4.1 Upload Project Files to S3 Bucket

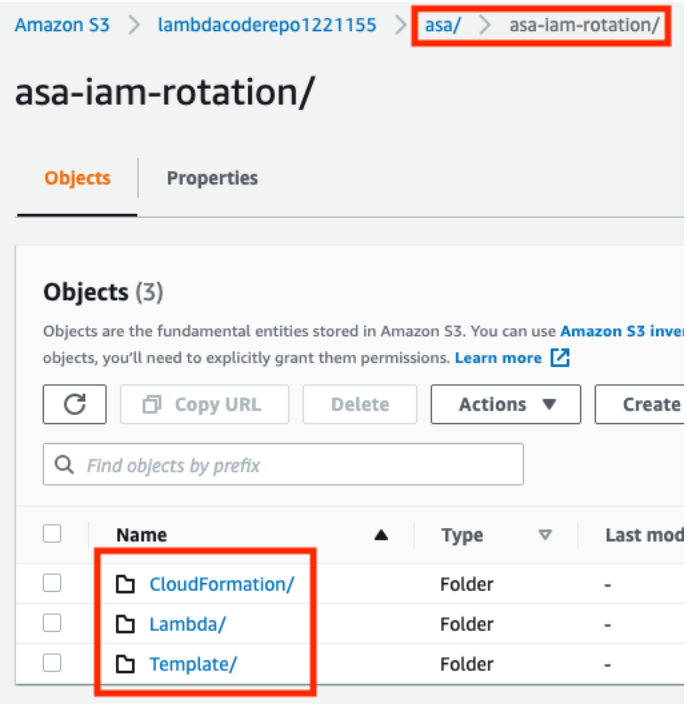
Step 1: Unzip the project files.

Step 2:. Log into the **AWS Management Console**, and select **S3** from the **Services** menu.

- 

Step 3: Drag & Drop the ASA folder into your S3 Bucket.

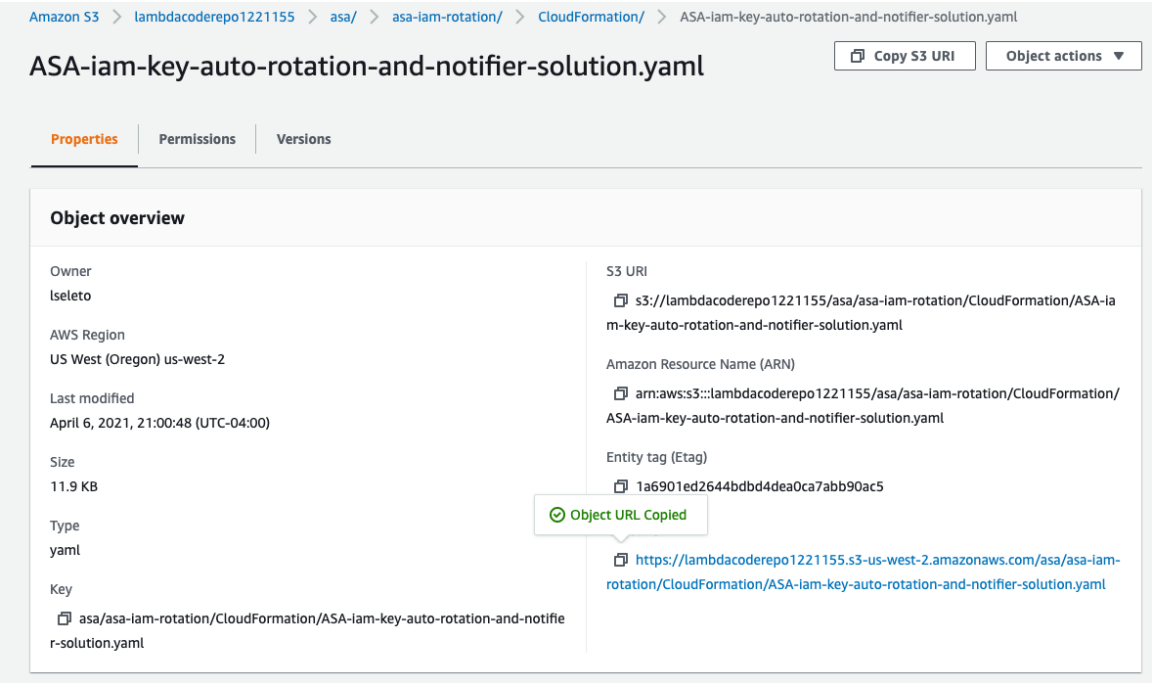
- **IMPORTANT:** Make sure all files are in the 'asa/asa-iam-rotation' folder structure.
- The 3 main folders from you need from the zip file are:
    - 'CloudFormation/', 'Template/', and 'Lambda/' as shown below:



IMPORTANT NOTE: *Ensure the account where you are deploying CloudFormation Stacks and StackSets from has access to this S3 Bucket.*

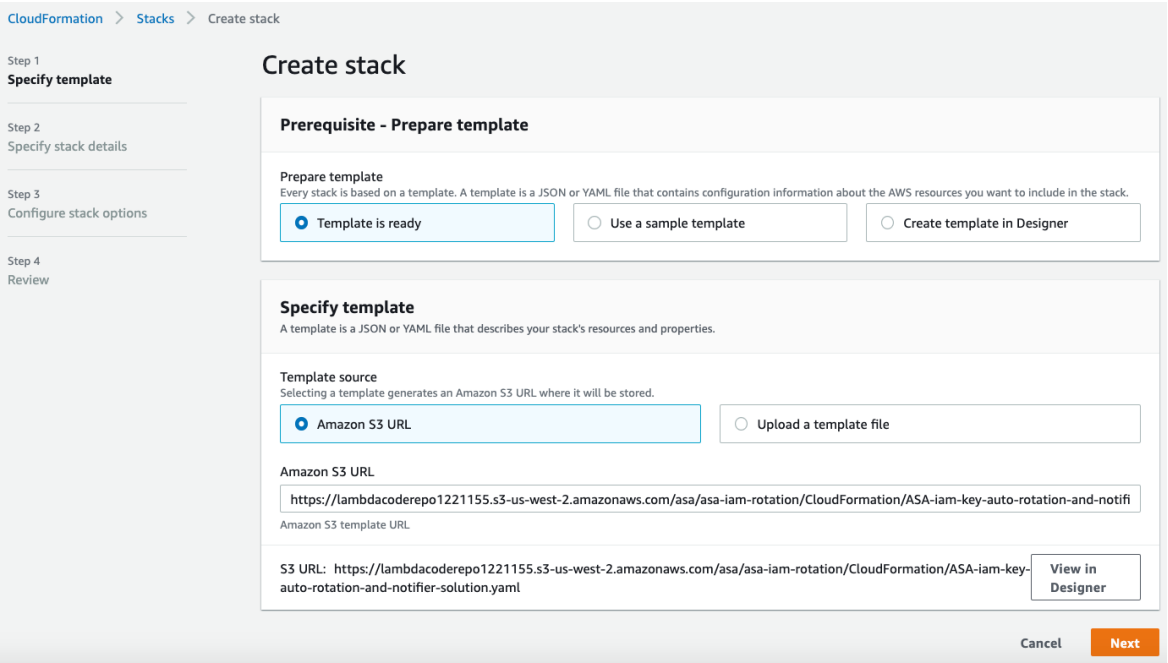
4.2 Deploy the main IAM Key Rotation Solution as a CloudFormation Stack

Step 1: Copy the Object URL of the "*ASA-iam-key-auto-rotation-and-notifier-solution.yaml*" template you uploaded to the S3 Bucket.



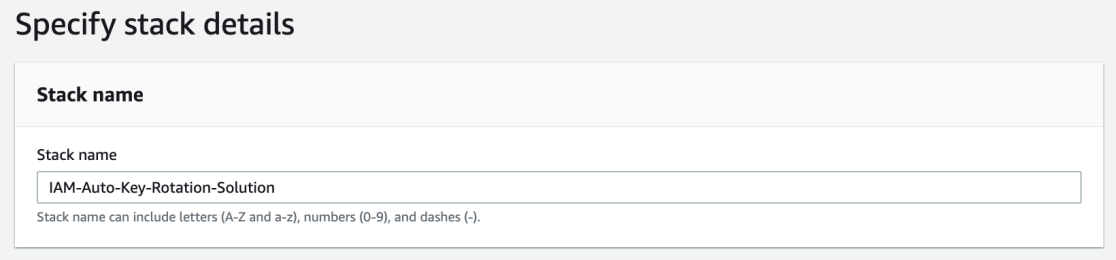
Step 2: Go to CloudFormation service and select Stacks. Paste the copied Object URL into the 'Template Source' and click 'Next'.

Note: Make sure the Account ID you are deploying this stack from matches the 'Primary AWS Account ID' from '4.2 Deploy IAM Assumed Roles CloudFormation Template as a StackSet'.

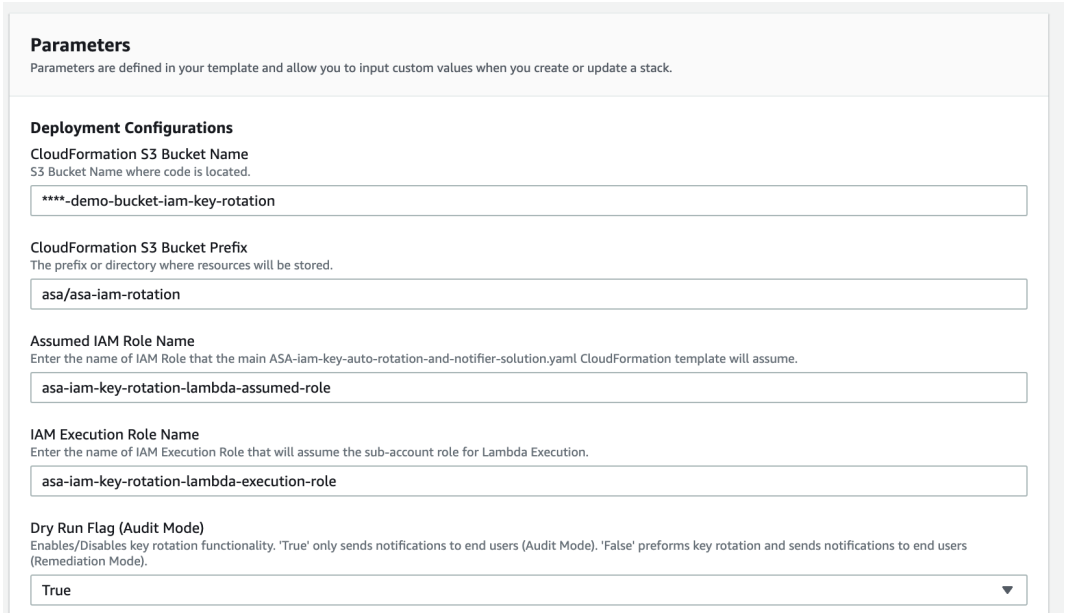


Step 3: Specify stack details

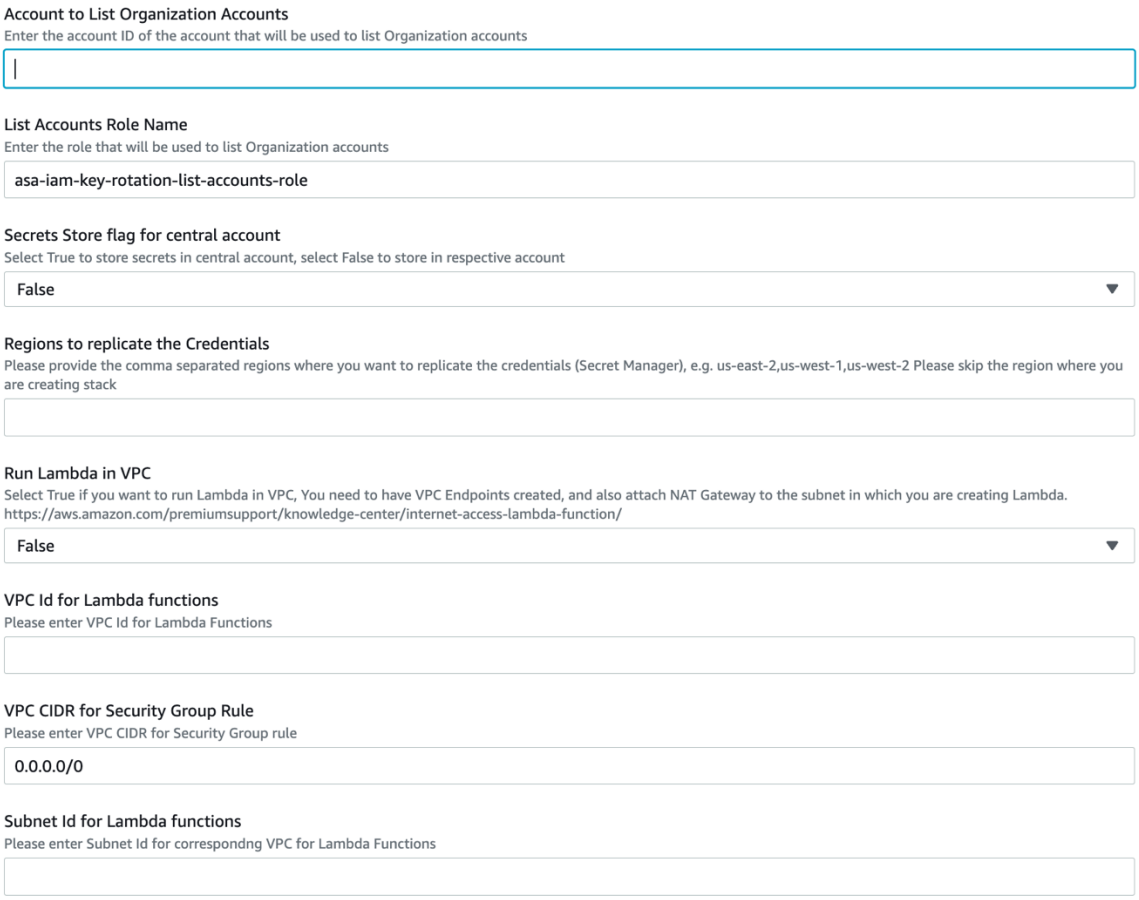
- Fill in the Stack name (ex: 'IAM-Auto-Key-Rotation-Solution').



- Enter the S3 Bucket name you uploaded all your files to in step '4.1 Upload Project Files to S3 Bucket' into 'CloudFormation S3 Bucket Name'.
- 'CloudFormation S3 Bucket Prefix' should match the folder structure of what you uploaded in the S3 Bucket (*ex: 'asa/asa-iam-rotation'*).
- You can leave 'Assumed IAM Role Name' and 'IAM Execution Role Name' as default or you can customize them.
- The 'Dry Run Flag' will allow you to simulate the AWS IAM Rotation Solution without actually rotating end user's keys. This is a great way to notify users or test the solution.
    - *Note: You can easily toggle between 'True' or 'False' values by updating the already deployed CloudFormation stack.*



- Enter the account ID of the account that will be used to list Organization accounts
- Enter the role that will be used to list Organization accounts
- Select the flag that will decide which account secrets will be stored. Select True to store secrets in central account, select False to store in respective account
- Please provide the comma separated regions where you want to replicate the credentials (Secret Manager), e.g. us-east-2,us-west-1,us-west-2 Please skip the region where you are creating stack.
- Select the flag that decides whether to run Lambda in VPC or standalone. Select True if you want to run Lambda in VPC, You need to have VPC Endpoints created, and also attach NAT Gateway to the subnet in which you are creating Lambda. https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/
- Please enter VPC Id for Lambda Functions , you can leave this field if you selected Run Lambda in VPC to be False
- Please enter VPC CIDR for Security Group rule, you can leave this field if you selected Run Lambda in VPC to be False
- Please enter Subnet Id for corresponding VPC for Lambda Functions, you can leave this field if you selected Run Lambda in VPC to be False



- Enter your team email distro under ‘Admin Email Address’.
    - *Note: This will be used in the ‘sent from’ section of email notifications to end users.*
- Enter your ‘AWS Organization ID’.
- You can leave the fields for ‘Email Template File Names’ default or customize them.
- Enter the SSM param name for SMTP User name, , this is required for Notifier Lambda to send an email
- Enter the SSM param name for SMTP Password, this is required for Notifier Lambda to send an email

Configure ASA Notifier Module

Admin Email Address
Email address that will be used in the 'sent from' section of the email.

Resource Owner Tag
(Optional) Tag key used to indicate the owner of an IAM user resource.

AWS Organization ID
Enter your AWS Organization ID, this will be used to restricted execution permissions to only approved AWS Accounts within your AWS Organization.

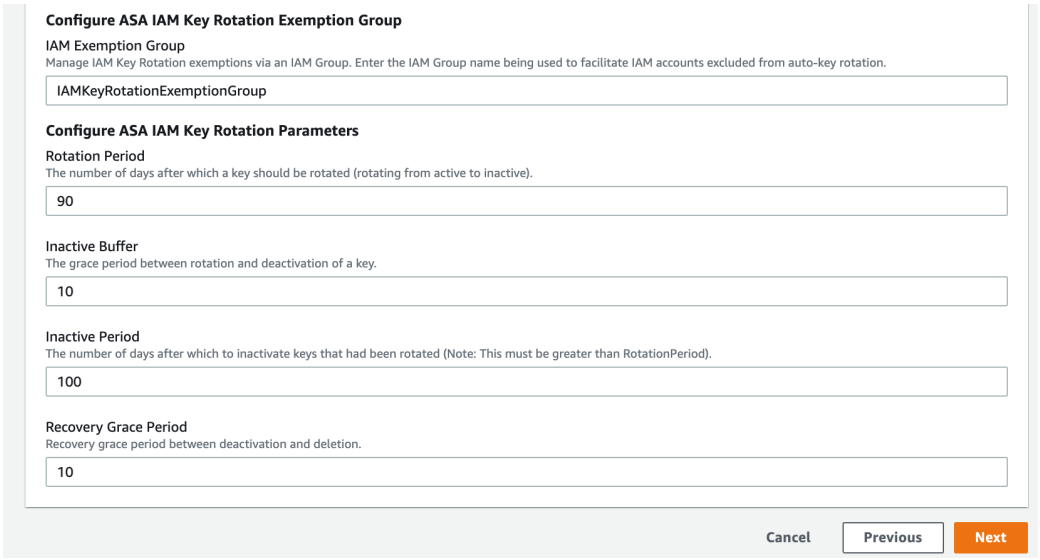
Email Template File Name [Audit Mode]
Enter the file name of the email html template to be sent out by the Notifier Module for Audit Mode. Note: Must be located in the 'S3 Bucket Prefix/Template/template_name.html' folder
iam-auto-key-rotation-enforcement.html

Email Template File Name [Enforce Mode]
Enter the file name of the email html template to be sent out by the Notifier Module for Enforce Mode. Note: Must be located in the 'S3 Bucket Prefix/Template/template_name.html' folder
iam-auto-key-rotation-enforcement.html

SMTP User SSM Parameter Name in case running Lambda in VPC
Enter the SSM param name for SMTP User, disregard this parameter if you are not running Lambda in VPC
/iam-key-rotation/smtp/user

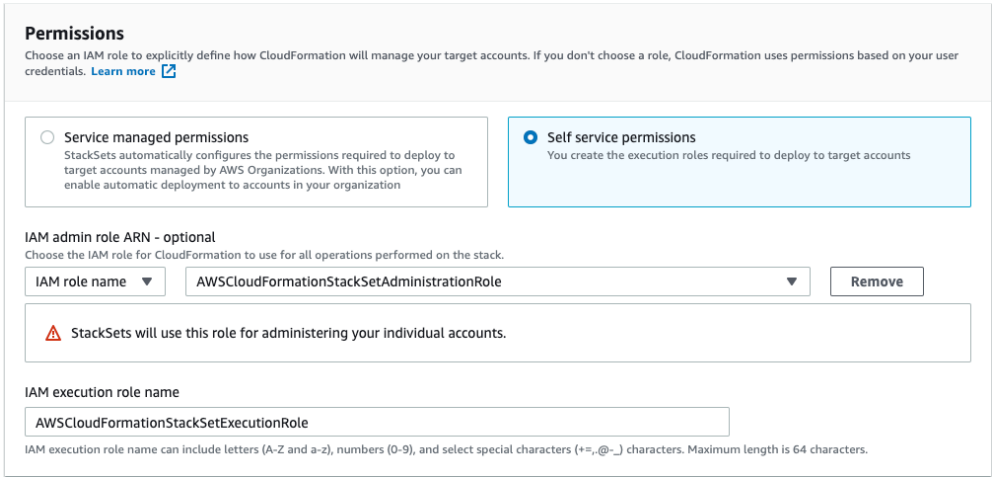
SMTP Password SSM Parameter Name in case running Lambda in VPC
Enter the SSM param name for SMTP User, disregard this parameter if you are not running Lambda in VPC
/iam-key-rotation/smtp/password

- You can leave 'IAMKeyRotationExemptionGroup' as default or you can customize it.
- You can leave the 'Configure ASA IAM Key Rotation Parameters' section as default or you can customize it.
- Once the stack details are filled out, click the 'Next' button.

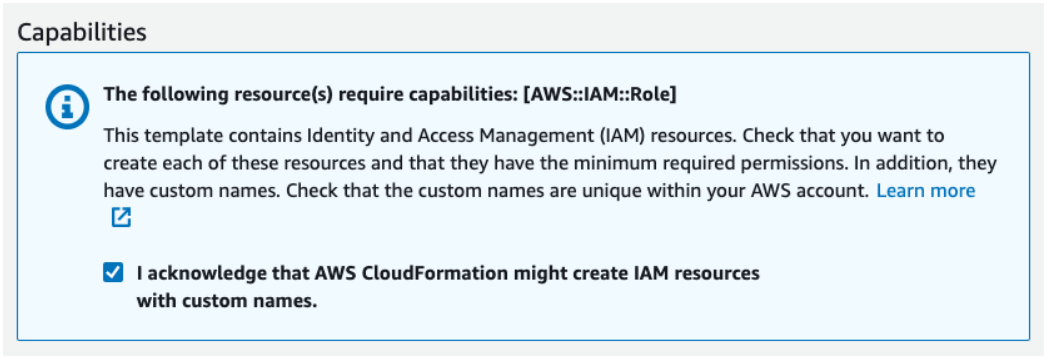


Step 4: For Permissions, select 'Service-managed permissions', and then 'Next'. Under 'Set deployment options', you can select either 'Deploy to organization' or 'Deploy to organizational units (OUs)'.

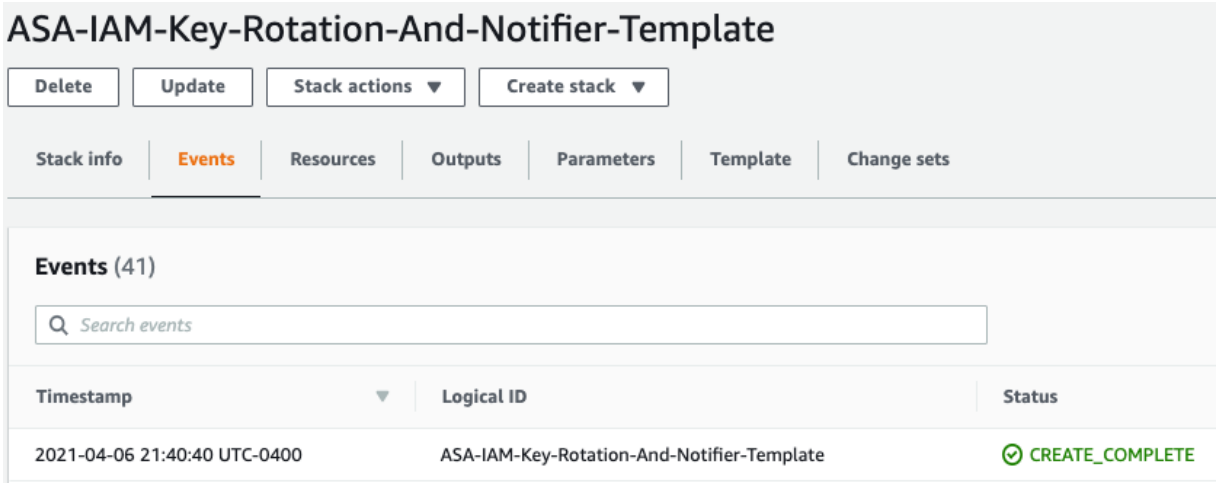
- You may leave 'Automatic deployment' and 'Account removal behavior' as defaults. Under ‘Specify regions’ select a region (*since IAM is a global service, the stack will be deployed within 1 region but the IAM role will be global for that account*). You can also leave ‘Deployment options’ default.



Step 5: On the final screen, make sure to check off the ‘I acknowledge that AWS CloudFormation might create IAM resources with custom names’ Option under ‘Capabilities’. Then click ‘Submit’.



Step 6: After launching the Stack, you will have to wait for it to deploy all of the resources. You can track progress via the ‘Events’ tab.



Step 7: Ensure the sender email is either verified within Amazon Simple Email Service (SES) or your account is removed from sandbox.

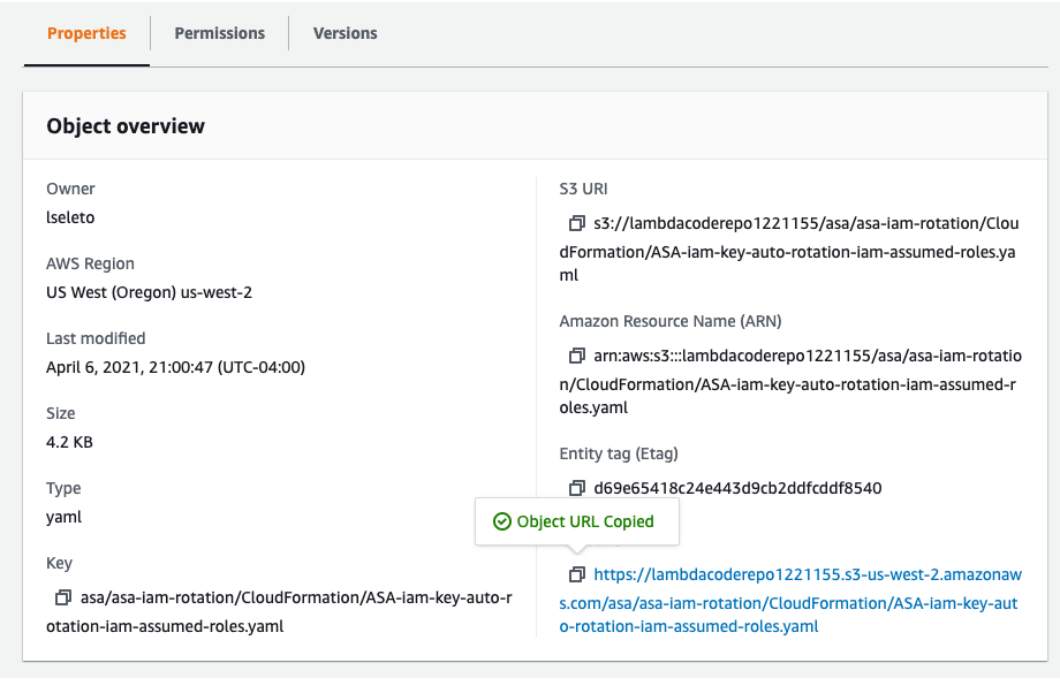
- See Section “6.2 MessageRejected: Email address is not verified.” for tutorials on how to correctly enable these configurations.

4.3 Deploy IAM Assumed Roles CloudFormation Template as a StackSet

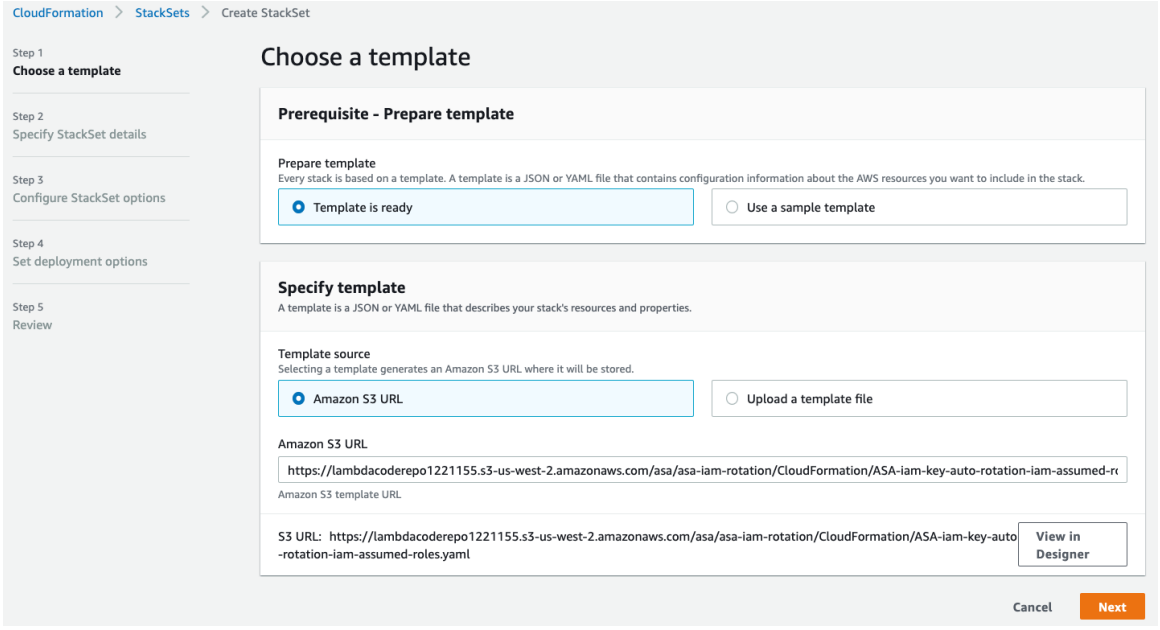
> ***IMPORTANT NOTE:*** *The ‘list_accounts’ API operation can only be called from the organization's management account or by a member account that is a delegated administrator for an AWS service.*
>
> *Reference:*
>
> *https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/organizations.html#Organizations.Client.list_accounts*

Step 1: Still in the AWS console, choose **CloudFormation** from the **Services** menu.

Step 2: In the left-hand pane, choose **StackSets**. (If you’ve never created a CloudFormation stack before, choose **Get Started**.)
Step 3: Click on **Create StackSet**.
Step 4: Copy the Object URL of the “*ASA-iam-key-auto-rotation-iam-assumed-roles.yaml*” template you uploaded to the S3 Bucket.

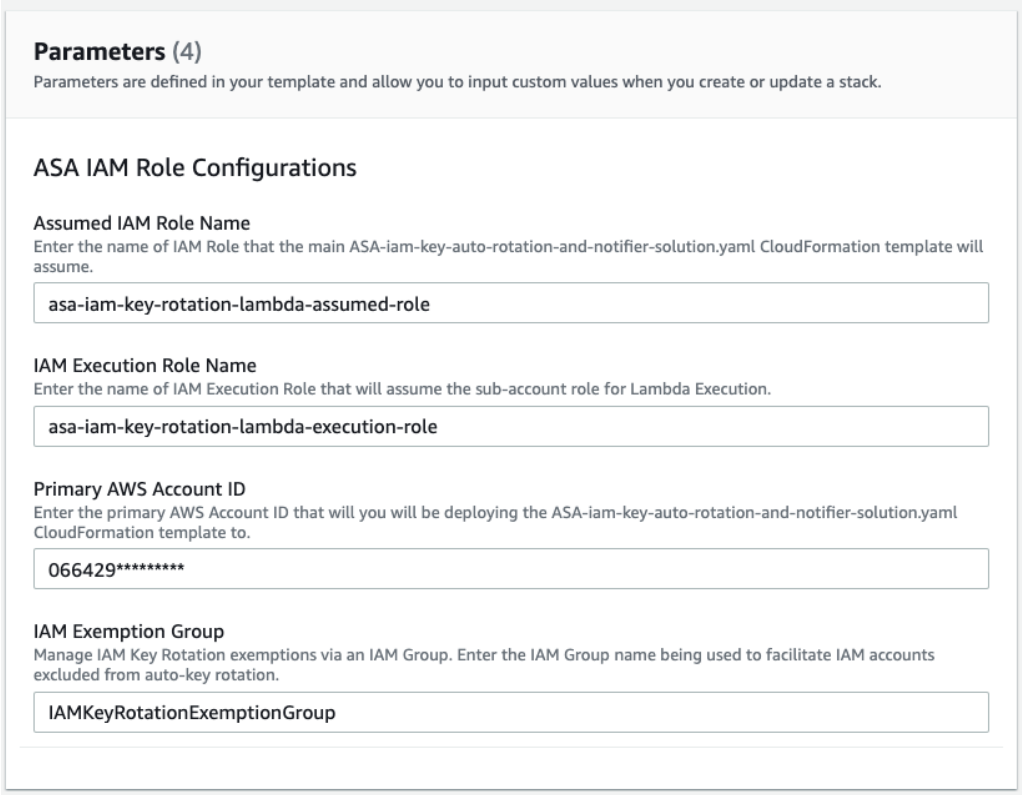


Step 2: Go to CloudFormation service and select StackSets. Paste the copied Object URL into the 'Template Source' and click 'Next'.



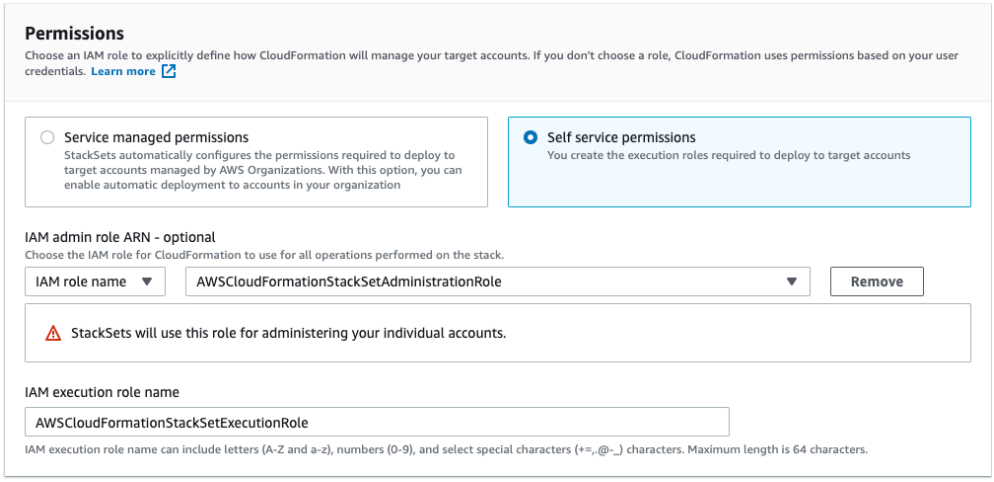
Step 3: Fill in the StackSet name (ex: 'IAM-Auto-Key-Rotation-Assumed-Roles').

- You can leave '*Assumed IAM Role Name*' and '*IAM Execution Role Name*' and '*IAMKeyRotationExemptionGroup*' as default or you can customize them.
- You will need to enter the 'Primary AWS Account ID' and 'AWS Organization ID'.
    - This Account ID is where you will be deploying the ASA-iam-key-auto-rotation-and-notifier-solution.yaml CloudFormation template to.
    - The Organization ID is to help further lock down the deployed IAM assumed roles.
- Click the 'Next' button.



Step 4: Select 'Service-managed permissions', and then 'Next'. Under 'Set deployment options', you can select either 'Deploy to organization' or 'Deploy to organizational units (OUs)'.

- You may leave 'Automatic deployment' and 'Account removal behavior' as defaults. Under 'Specify regions' select a region (*since IAM is a global service, the stack will be deployed within 1 region but the IAM role will be global for that account*). You can also leave 'Deployment options' default.

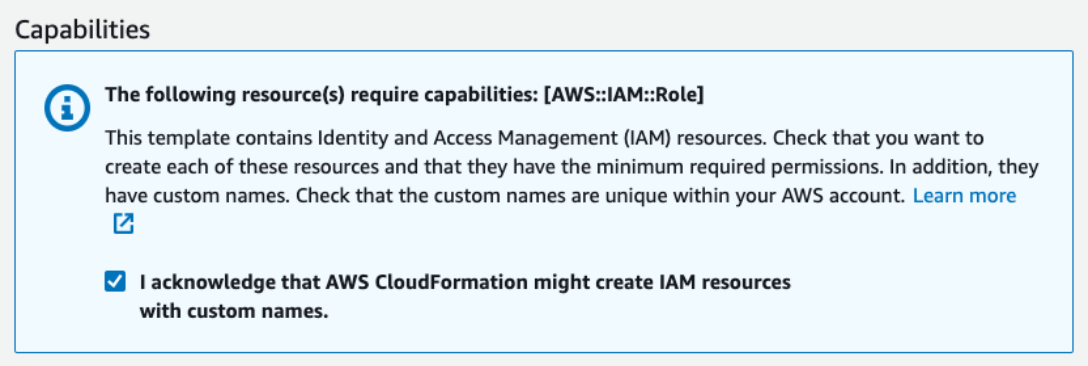


IMPORTANT NOTE: *If you selected more than 1 region per account you will get an error message similar to:*

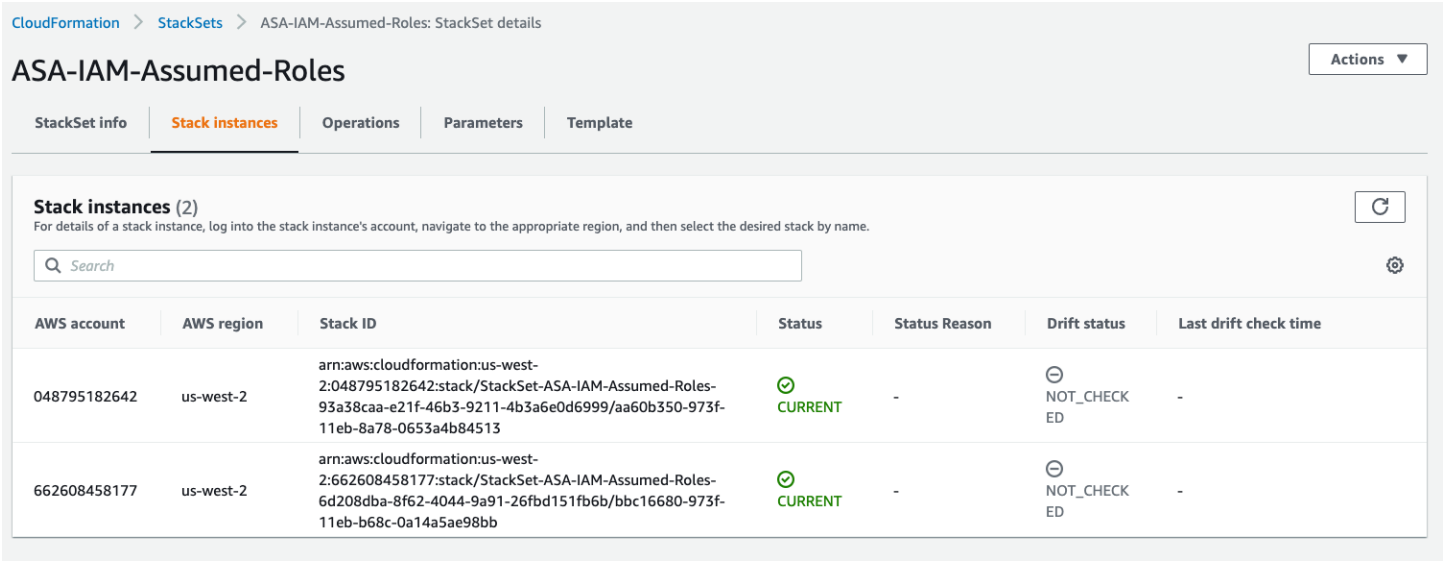
ResourceLogicalId:ASAIAMAssumedRole, ResourceType:AWS::IAM::Role, ResourceStatusReason:asa-iam-key-rotation-lambda-assumed-role already exists.

> This is due to IAM being a global service, once it's deployed in 1 region it will be there for all regions.

Step 5: On the final screen, make sure to check off the 'I acknowledge that AWS CloudFormation might create IAM resources with custom names' Option under 'Capabilities'. Then click 'Submit'.



Step 6: After launching the StackSet, you will have to wait for it to deploy the IAM role to all sub-accounts. You can track progress via the 'Stack instances' tab.

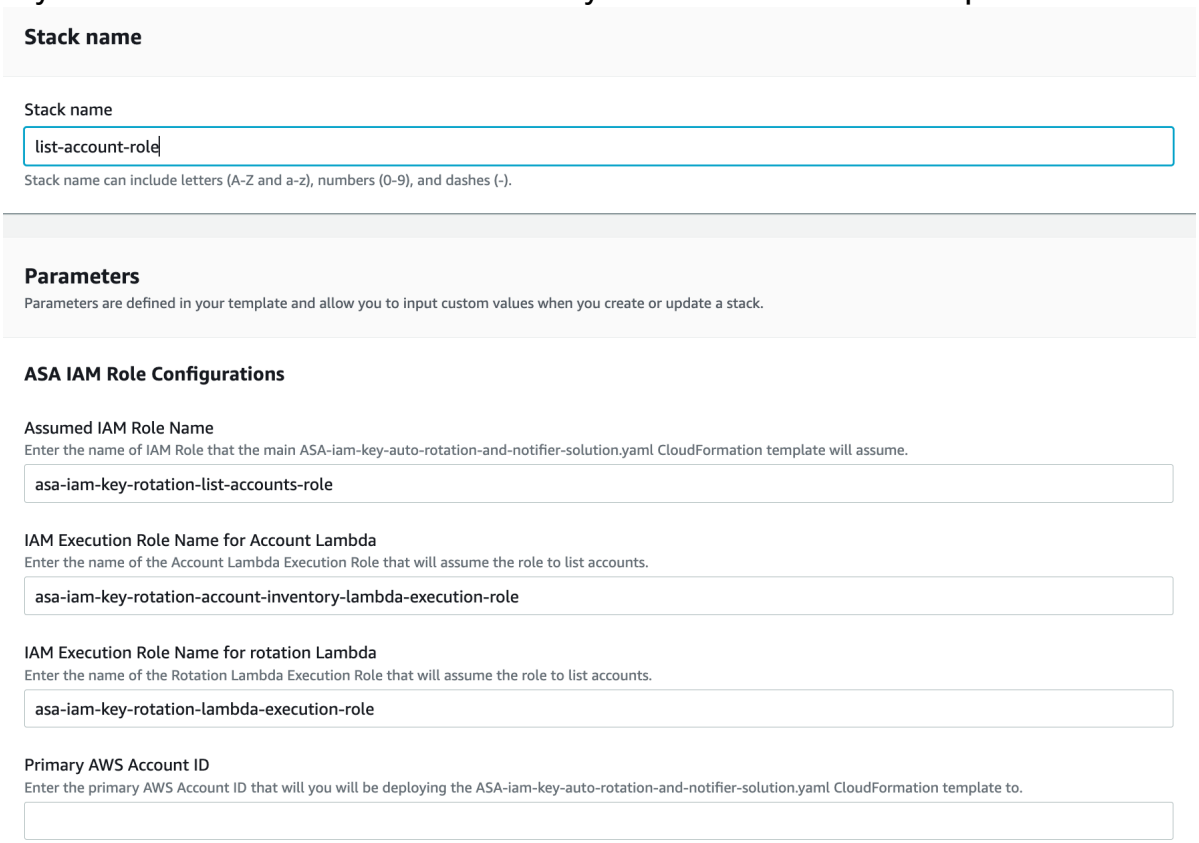


4.4 Deploy the List Account Role in the Central/Management Account.

Step 1: We need to create the IAM role in the central/management account so that this role can be assumed by account Lambda to list the accounts under the AWS organization. Go to AWS Console, CloudFormation, click on Create stack and enter details as below

- Enter the name of IAM Role that the main ASA-iam-key-auto-rotation-and-notifier-solution.yaml CloudFormation template will assume.
- Enter the name of the Account Lambda Execution Role that will assume the role to list accounts.

- Enter the name of the Rotation Lambda Execution Role that will assume the role to list accounts.
- Enter the primary/deployment AWS Account ID that will you will be deploying the ASA-iam-key-auto-rotation-and-notifier-solution.yaml CloudFormation template to.

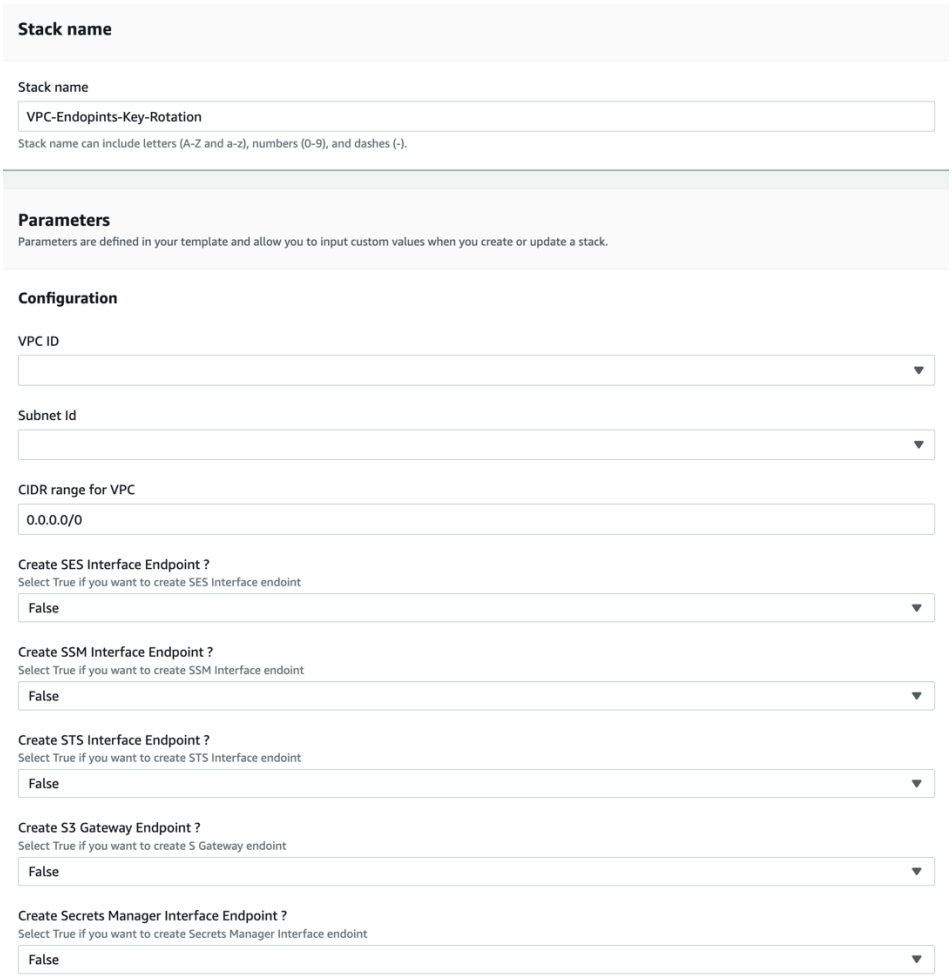


- Click Next → Next → Acknowledge the resource creation and click on “Create Stack”

4.5 Deploy the VPC Endpoint template if you are running Lambda in VPC

Step 4: We need to create VPC endpoints if Lambdas are going to be run in VPC. Go to AWS Console, CloudFormation, click on Create stack and enter details as below

- Select VPC ID
- Select Subnet ID
- SES Interface endpoint: Select True if you want to create one, you can skip this if it already exists
- SSM Interface endpoint: Select True if you want to create one, you can skip this if it already exists
- STS Interface endpoint: Select True if you want to create one, you can skip this if it already exists
- S3 Gateway endpoint: Select True if you want to create one, you can skip this if it already exists
- Secrets Manager Interface endpoint: Select True if you want to create one, you can skip this if it already exists



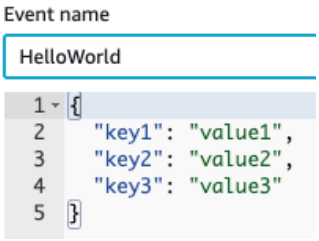
- Click Next → Next → Acknowledge the resource creation and click on “Create Stack”

5.0 Validating Deployment & Manual Tests

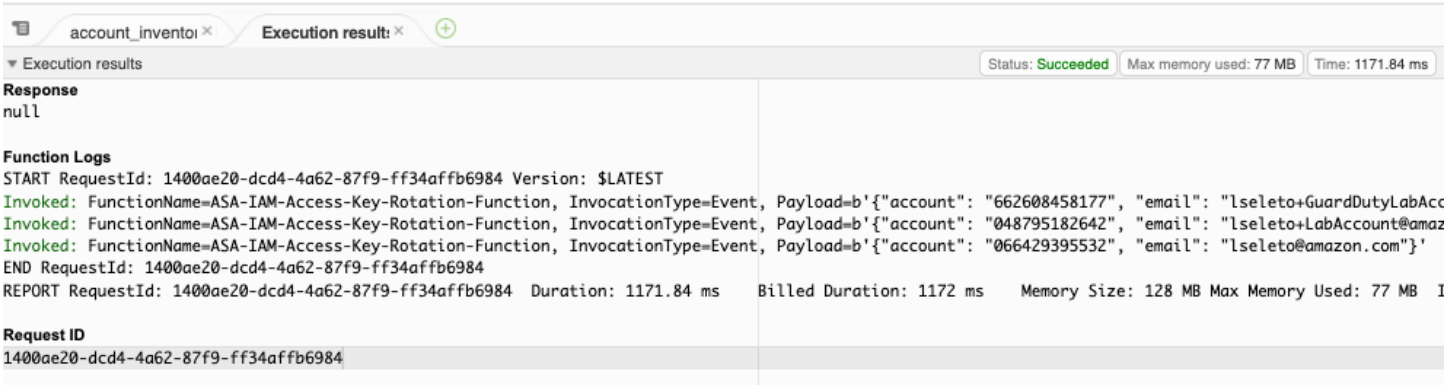
5.1 Manually Test: Daily Schedule via ASA-Account-Inventory Lambda Function

You can either wait for the daily CloudWatch cron job or access the 'ASA-Account-Inventory' Lambda Function directly.

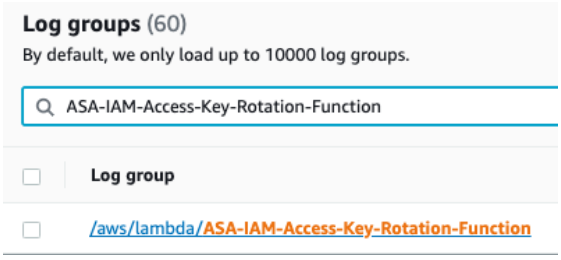
If you want to kick it off manually, just create a default 'HelloWorld' test event (*content doesn't matter since it is cron job triggered*).



Then click 'Test', you should see Invoked outputs where the 'ASA-Account-Inventory' Function invokes the 'ASA-IAM-Access-Key-Rotation-Function' Lambda.



This can also be monitored via CloudWatch log groups.



5.2 Manually Test: ASA-IAM-Access-Key-Rotation-Function Lambda Function

Note: This section's code can be found under the 'Test Units' folder included with the solution.

Example json event message getting sent to the 'ASA-IAM-Access-Key-Rotation-Function' Lambda Function from the 'ASA-Account-Inventory' Lambda Function.

```
{
        "account": "<AccountID>",
        "email": "<AccountEmailHere>"
}
```

To Force a key rotation, include 'ForceRotate' in the json body:

```
{
        "ForceRotate": "<IAM Username>" To be rotated"
        "account": "<AccountID>",
        "email": "<AccountEmailHere>",
        "name": "<Account Name>"
}
```

5.3 Manually Test: ASA-Notifier Lambda Function

Note: This section's code can be found under the 'Test Units' folder included with the solution.

Example json event message getting sent to the 'ASA-Notifier' Lambda Function from the 'ASA-IAM-Access-Key-Rotation-Function' Lambda Function.

```
{
        "email": "PLACE EMAIL HERE",
        "invokedby": "arn:PARTITION:lambda:REGION:ACCOUNT:function:ASA-IAM-Access-Key-Rotation-Function",
        "subject": "[IMPORTANT] Active AWS IAM Access Key was Rotated to Inactive due to Key Age Security Violation.",
        "email_template": "iam-auto-key-rotation-enforcement.html",
  "template_values":
    {
        "account_id": "PLACE ACCOUNT ID HERE",
        "timestamp": "2021-11-04T22:48:39.640450+00:00",
        "actions": ["ACTION: ROTATE key username-here:key-arn-here.  Forced active key rotation."],
        "rotation_period": 90,
        "installation_grace_period": 7,
        "recovery_grace_period": 10,
        "partition_name": "AWS Standard"
    }
}
```

6.0 Troubleshooting

6.1 ClientError: An error occurred (AccessDenied) when calling the AssumeRole operation

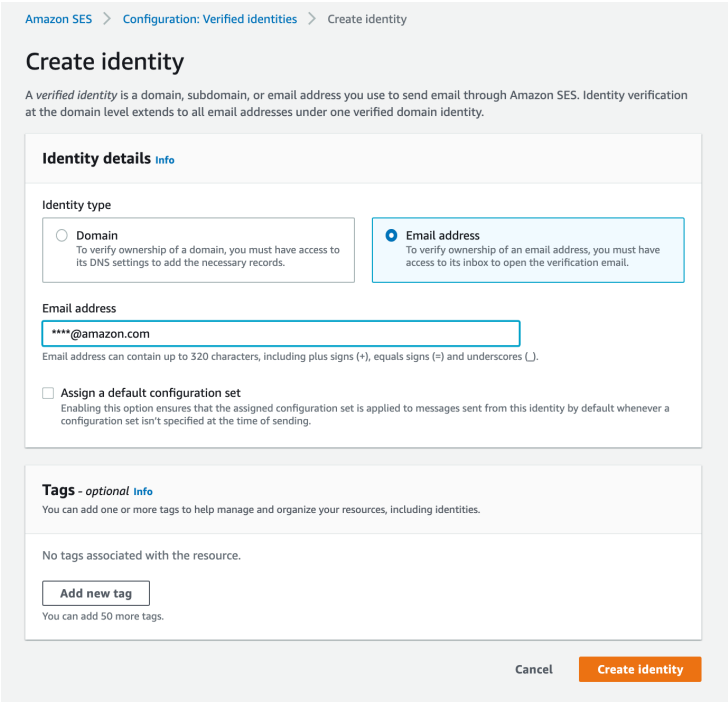
- If you see this error message in Lambda or CloudWatch logs, it means that the Assumed Role StackSet was not successfully deployed to that account.
- Review the account in question and redeploy the CloudFormation Template described in '4.2 Deploy IAM Assumed Roles CloudFormation Template as a StackSet' to it.
- If deployed via organizations, the root org account will not be included.

Full Error Message:

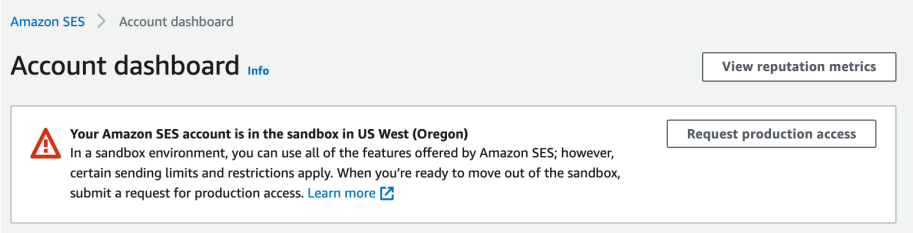
> [ERROR] ClientError: An error occurred (AccessDenied) when calling the AssumeRole operation:
>
> User: arn:aws:sts::066429******:assumed-role/ASA-IAM-Key-Rotation-And-RotationLambdaFunction/ASA-IAM-Access-Key-Rotation-Function is not authorized to perform: sts:AssumeRole on resource: arn:aws:iam::06642*******:role/asa-iam-key-rotation-lambda-assumed-role

6.2 MessageRejected: Email address is not verified.

- If you see this error message in Lambda or CloudWatch logs, it means the Amazon Simple Email Service (SES) is in sandbox mode and the sender email is not verified.
- To verify an SES sender identity follow this tutorial:
  - https://docs.aws.amazon.com/ses/latest/dg/creating-identities.html



- To remove SES from sandbox mode, follow this tutorial:
    - https://docs.aws.amazon.com/ses/latest/DeveloperGuide/request-production-access.html



Full Error Message:

> "errorMessage": "An error occurred (MessageRejected) when calling the SendEmail operation: Email address is not verified. The following identities failed the check in region US-WEST-2: *****@****.com"