# Autorotation of IAM Access Keys and Secret Keys with Lambda

# 

# AWS Access Keys are the credentials used by IAM User. They are used to sign programmatic requests to the AWS CLI or AWS API. Access Keys consist of two parts: an access key ID and a secret access key. Like a username and password, we have to provide both together to authenticate the requests. A security recommendation suggests that we should only use IAM roles instead of access keys but still in some cases we need to have the access keys handy. These access keys will provide a user complete access to a set of resources defined by the IAM User Policy. So, if the access keys are lost or went into wrong hands, your resources are at risk.

# (AK-Access key ID; SK – Secret Key ID)

# [IAM Best Practices](https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#remove-credentials) (as per AWS Security Compliance Management)

# Do Not Share Access Keys

# Rotate Credentials Regularly (Key must be renewed after proper interval)

# Remove Unnecessary Credentials (Inactive keys)

# Overview of this Automation

# 

# *Fig: Key Rotation*

# So, let’s automate this with a simple setup:

# Service required to achieve this task.

# Lambda function (Python & Boto3):

# Code functionality-

# 

# Scheduled CloudWatch trigger:

# CloudWatch Schedular will automatically triggered the lambda function as per our scheduling.

# SES:

# SES will automatically trigger mail to user without exposing AK and SK to any other person.

# Test Result Screenshots:

# For Quick test result I use key age as

# 1 day instead of 85 days; 2 day instead of 90 days; 3 days as 95 days.

# User whose access key is not in used:

# 

# User whose key age >90 days new Access key and Secret key:

# User whose key age >95 days deliver new Access key and Secret key and old keys deactivation notification:

# 