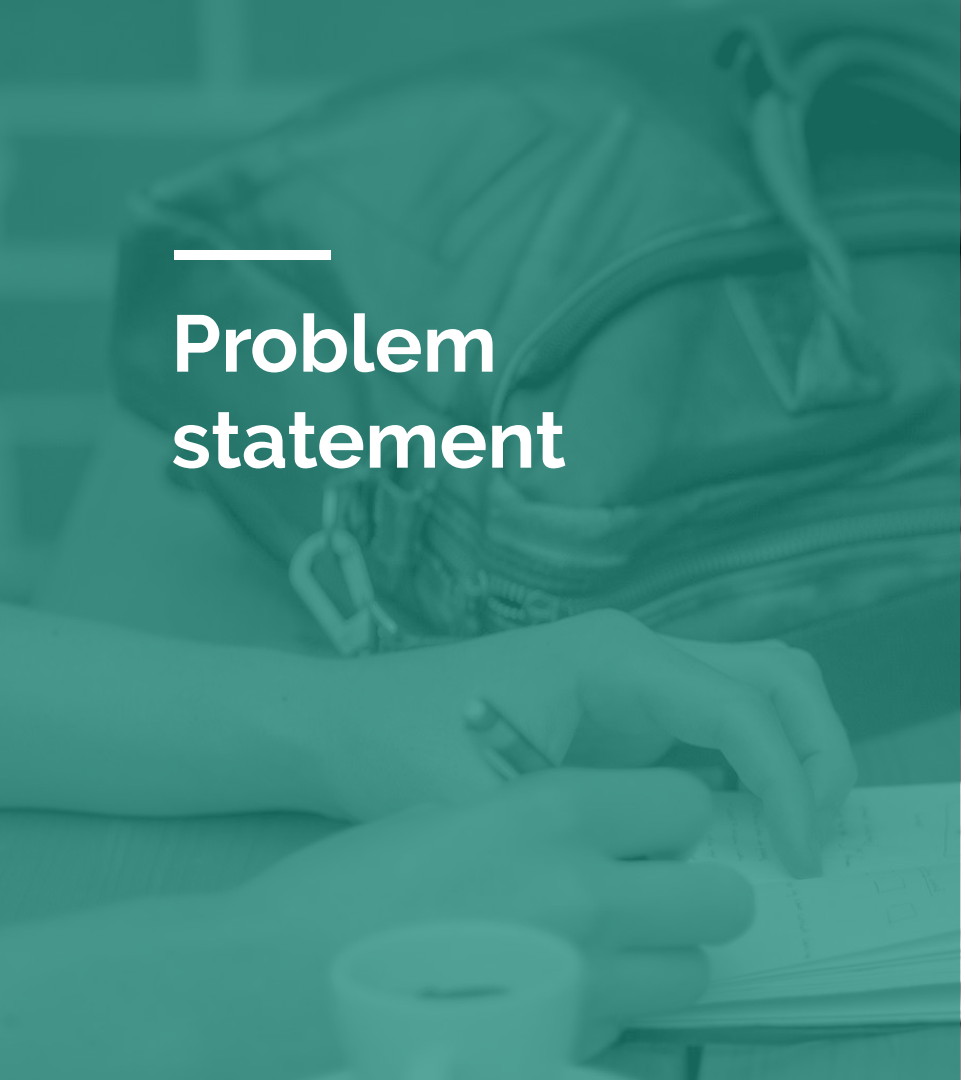
Email & Encryption using AWS

Inez Wibowo

Outline

The Problem



Problem statement

A small startup has limited resources and needs to focus on running the business

Reducing overhead costs associated with renting space, employees, and website system administration is typically a good approach.

What customers do today

An SF startup uses third party resources, like Gmail, and has no centralized secure solution

The startup might be losing business because clients are not comfortable with their information unprotected .

Use cases / user stories

- ➔ “As a Web Administrator, I would like to restrict permissions on who can email and encrypt documentation based on groups or roles.”
- ➔ “As a Developer, I would like to be able to implement new services.”
- ➔ “As an Executive, I would like to see a report of how many emails were sent, bounced, rejected, etc.”
- ➔ “As a User, I would like to know that my sensitive information is encrypted and secure.”

The Solution

Solution description

We are recommending AWS, Amazon Web Services.

It is a Cloud Computing service that can host a server instance in the cloud with Elastic Compute Cloud (EC2), email service with Simple Email Service (SES), allow for secure archival S3 (Simple Storage Solution), allows encryption with Key Management Services (KMS).

It is cost effective, as it will be used on demand - therefore, the startup will be charged only as much as they use.

Amazon also provides a Service Level Agreement of 99.99% Monthly Uptime Percentage service commitment

Solution Exploration

- What is AWS
- Setting up AWS SES for a small business
- Setting up IAM users on the account
- Save items in the S3 bucket for archival and encrypt and decrypt it

What is AWS ?

Amazon Web Services (AWS) is a comprehensive, evolving cloud computing platform provided by Amazon. It provides a mix of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

Launched in 2006.

It provided more than 100 services including those for compute, databases, infrastructure management, application development and security.

We are using some of this services in class.

Setting up AWS SES

- Set up Amazon account and Choose AWS SES
- Launch an EC2 instance
- Download the .pem file
- Navigate to the download .pem file and SSH into it
    - Follow instructions provided when you click the Instance running and choose “Connect” as one of the actions
- Once you see ec2@... or ubuntu@... in your terminal , then you have successfully signed into the Amazon Web Services server instance

Install AWS SDK using NODE.JS

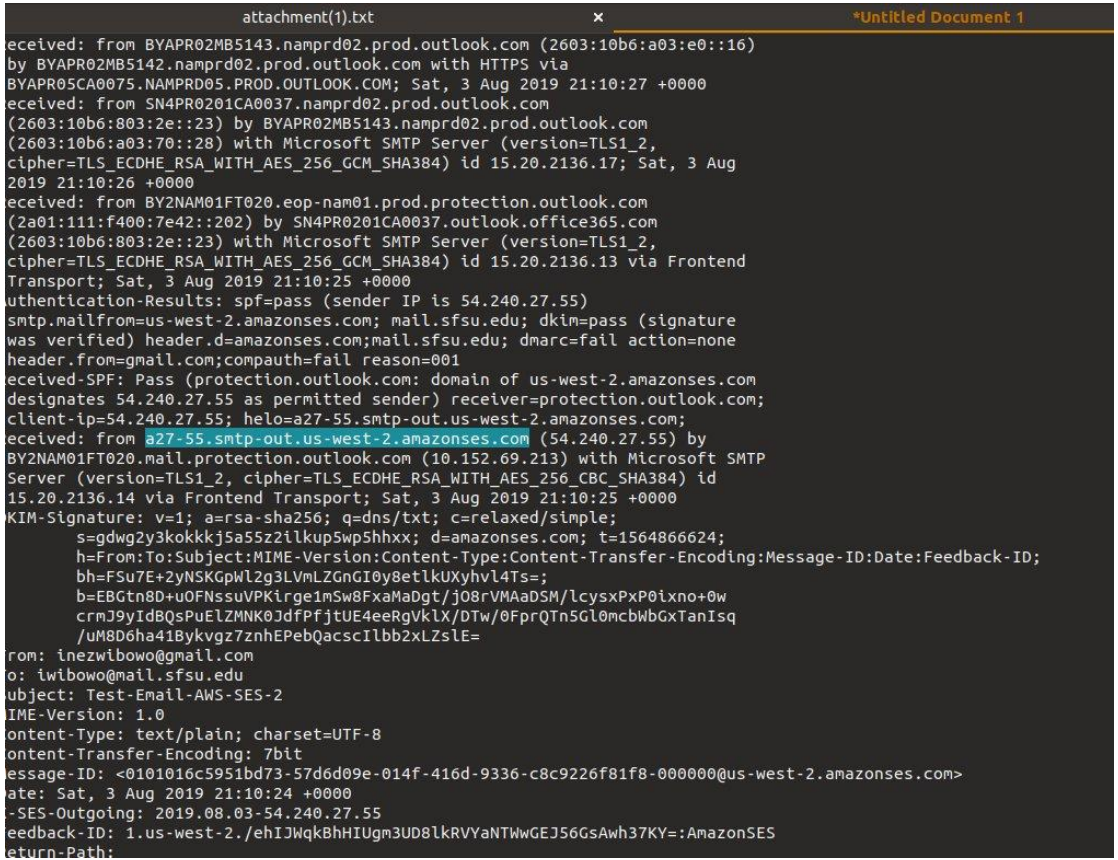
- You can set up the environment by following this tutorial courtesy of Andrew Puch's tutorial on github
- Clone https://github.com/andrewpuch/aws-ses-node-js-examples
- sudo su
- apt-get update
- apt-get upgrade -y
- apt-get dist-upgrade -y
- apt-get autoremove -y
- apt-get install nodejs npm git -y
- ln -s /usr/bin/nodejs /usr/bin/node
- git clone https://github.com/andrewpuch/aws-ses-node-js-examples.git
- cd aws-ses-node-js-examples
- npm install
- cp config-sample.json config.json
- edit app.js with your own email address
- edit config.js with the aws secret key and access id.
- https://console.aws.amazon.com/iam/home?#/security_credentials
- start service by typing the following command: node app.js

AWS SES endpoints

Amazon Simple Email Service (Amazon SES)

Region Name	Region	API (HTTPS) Endpoint	SMTP Endpoint	Email Sending or Receiving
US East (N. Virginia)	us-east-1	email.us-east-1.amazonaws.com	email-smtp.us-east-1.amazonaws.com	Email sending
US West (Oregon)	us-west-2	email.us-west-2.amazonaws.com	email-smtp.us-west-2.amazonaws.com	Email sending
EU (Ireland)	eu-west-1	email.eu-west-1.amazonaws.com	email-smtp.eu-west-1.amazonaws.com	Email sending
US East (N. Virginia)	us-east-1	N/A	inbound-smtp.us-east-1.amazonaws.com	Email receiving
US West (Oregon)	us-west-2	N/A	inbound-smtp.us-west-2.amazonaws.com	Email receiving
EU (Ireland)	eu-west-1	N/A	inbound-smtp.eu-west-1.amazonaws.com	Email receiving

Example of email received from AWS server



attachment(1).txt × *Untitled Document 1

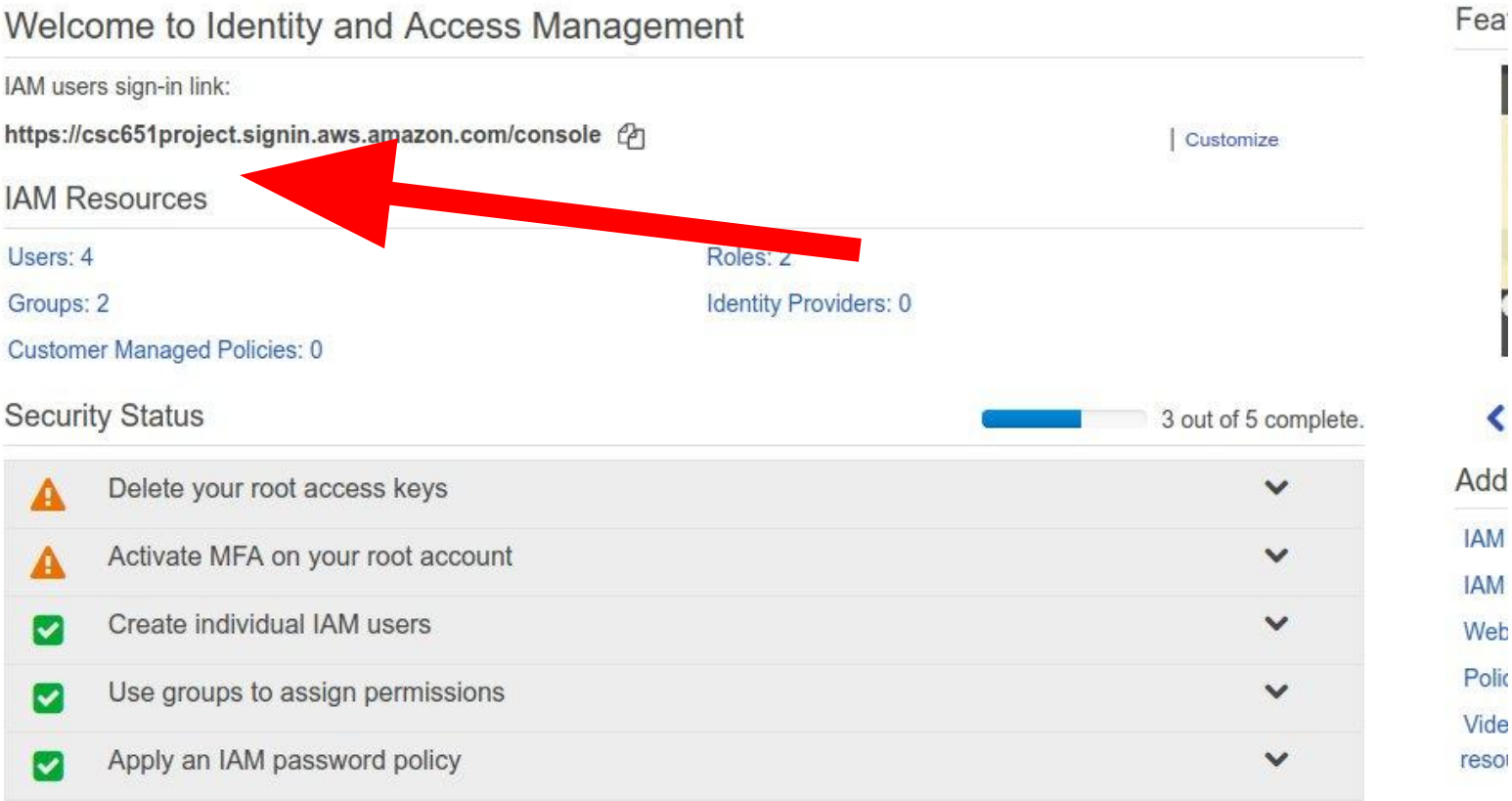
```
eceived: from BYAPR02MB5143.namprd02.prod.outlook.com (2603:10b6:a03:e0::16)
by BYAPR02MB5142.namprd02.prod.outlook.com with HTTPS via
BYAPR05CA0075.NAMPRD05.PROD.OUTLOOK.COM; Sat, 3 Aug 2019 21:10:27 +0000
eceived: from SN4PR0201CA0037.namprd02.prod.outlook.com
(2603:10b6:803:2e::23) by BYAPR02MB5143.namprd02.prod.outlook.com
(2603:10b6:a03:70::28) with Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.2136.17; Sat, 3 Aug
2019 21:10:26 +0000
eceived: from BY2NAM01FT020.eop-nam01.prod.protection.outlook.com
(2a01:111:f400:7e42::202) by SN4PR0201CA0037.outlook.office365.com
(2603:10b6:803:2e::23) with Microsoft SMTP Server (version=TLS1_2,
cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.2136.13 via Frontend
Transport; Sat, 3 Aug 2019 21:10:25 +0000
uthentication-Results: spf=pass (sender IP is 54.240.27.55)
smtp.mailfrom=us-west-2.amazonses.com; mail.sfsu.edu; dkim=pass (signature
was verified) header.d=amazonses.com;mail.sfsu.edu; dmarc=fail action=none
header.from=gmail.com;compauth=fail reason=001
eceived-SPF: Pass (protection.outlook.com: domain of us-west-2.amazonses.com
designates 54.240.27.55 as permitted sender) receiver=protection.outlook.com;
client-ip=54.240.27.55; helo=a27-55.smtp-out.us-west-2.amazonses.com;
eceived: from a27-55.smtp-out.us-west-2.amazonses.com (54.240.27.55) by
BY2NAM01FT020.mail.protection.outlook.com (10.152.69.213) with Microsoft SMTP
Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384) id
15.20.2136.14 via Frontend Transport; Sat, 3 Aug 2019 21:10:25 +0000
KIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
       s=gdwg2y3kokkkj5a55z2ilkup5wp5hhxx; d=amazonses.com; t=1564866624;
       h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Message-ID:Date:Feedback-ID;
       bh=FSu7E+2yNSKGpWl2g3LVmLZGnGI0y8etlkUXyhvl4Ts=;
       b=EBGtn8D+uOFNssuVPKirge1mSw8FxaMaDgt/jO8rVMAaDSM/lcysxPxP0ixno+0w
       crmJ9yIdBQsPuElZMNK0JdfPfjtUE4eeRgVklX/DTw/0FprQTn5Gl0mcbWbGxTanIsq
       /uM8D6ha41Bykvgz7znhEPebQacscIlbb2xLZslE=
rom: inezwibowo@gmail.com
o: iwibowo@mail.sfsu.edu
ubject: Test-Email-AWS-SES-2
IME-Version: 1.0
ontent-Type: text/plain; charset=UTF-8
ontent-Transfer-Encoding: 7bit
essage-ID: <0101016c5951bd73-57d6d09e-014f-416d-9336-c8c9226f81f8-000000@us-west-2.amazonses.com>
ate: Sat, 3 Aug 2019 21:10:24 +0000
-SES-Outgoing: 2019.08.03-54.240.27.55
eedback-ID: 1.us-west-2./ehIJWqkBhHIUgm3UD8lkRVYaNTWwGEJ56GsAwh37KY=:AmazonSES
eturn-Path:
```

Setting up IAM user

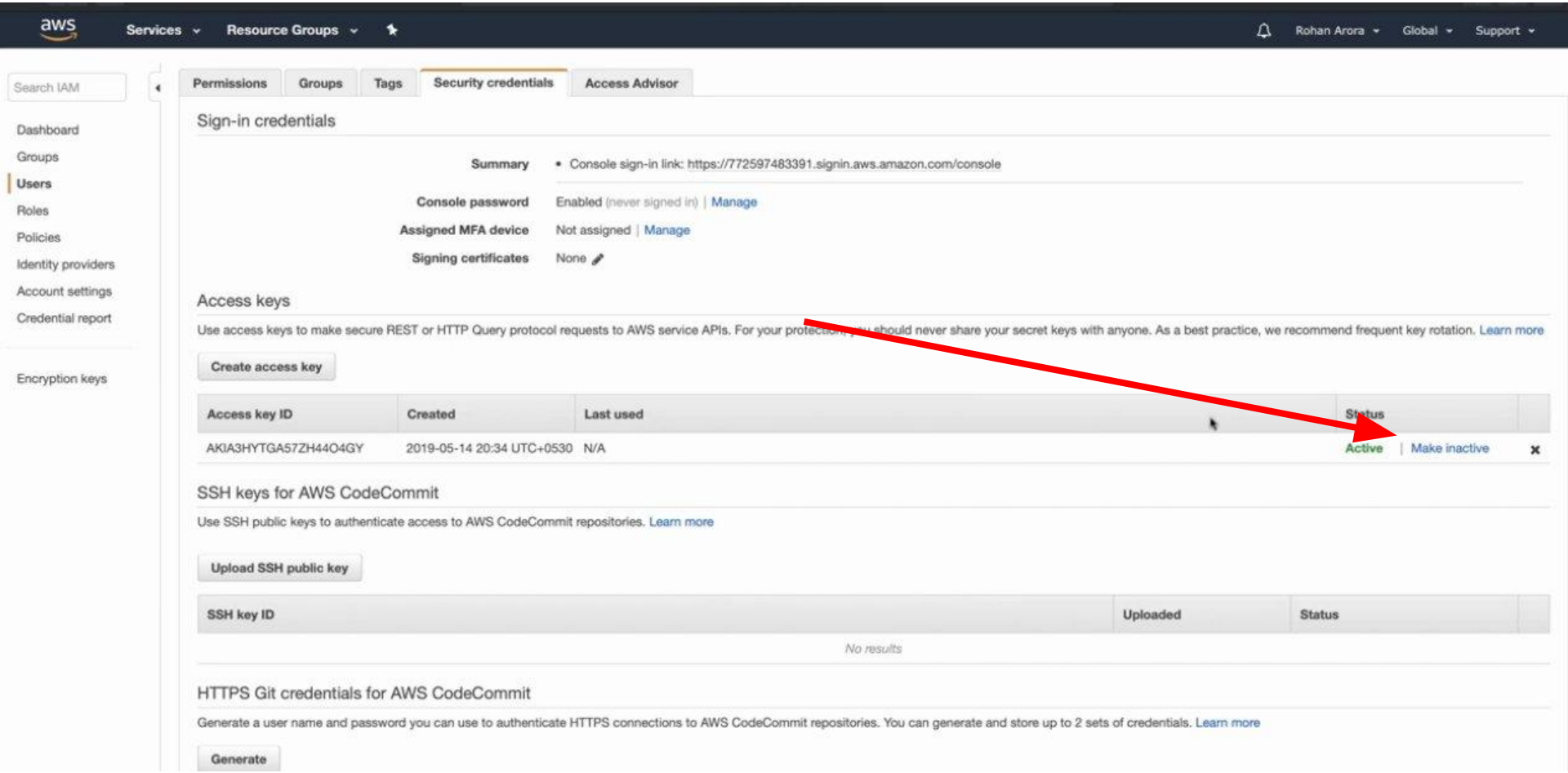
Creating user permissions

1. Set up a Root account.
   - Responsible for administrative purposes: defining authentication.
   - Does not perform launching, terminating activities, implementation, etc.
2. Set up users.
   - On one page we can set up several users.
   - Administrator to retain the csv file at first set up that contains access key and secret key.
   - If someone forgets their access key, then the user has to be marked inactive and another user can be created.

```
User name,Password,Access key ID,Secret access key,Console login link
kat,,AKIATAOJUO6YJ2RRVO3W,cUYa6lD/2nRxTz3YHds1kLfax3UmsPBcnZI/gsHu,https://207118170032.signin.aws.amazon.com/console
nayan,,AKIATAOJUO6YGFOX6WSH,jV39DpKOQYdQd7gPMQZdkRL/0+XeZiGE+Z8hLIBO,https://207118170032.signin.aws.amazon.com/console
```



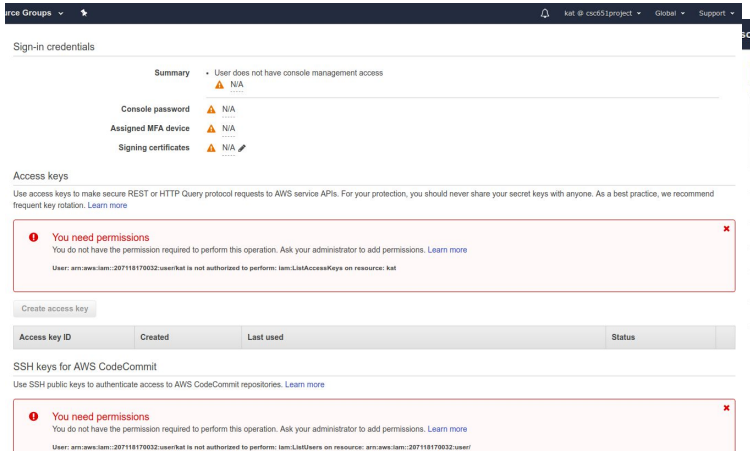
Root Account Dashboard


aws
Services
Resource Groups
Rohan Arora
Global
Support
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Permissions
Groups
Tags
Security credentials
Access Advisor
Sign-in credentials
Summary
Console sign-in link: https://772597483391.signin.aws.amazon.com/console
Console password
Enabled (never signed in) | Manage
Assigned MFA device
Not assigned | Manage
Signing certificates
None
Access keys
Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more
Create access key
Access key ID
Created
Last used
Status
AKIA3HYTGA57ZH44O4GY
2019-05-14 20:34 UTC+0530
N/A
Active
Make inactive
SSH keys for AWS CodeCommit
Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more
Upload SSH public key
SSH key ID
Uploaded
Status
No results
HTTPS Git credentials for AWS CodeCommit
Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can generate and store up to 2 sets of credentials. Learn more
Generate

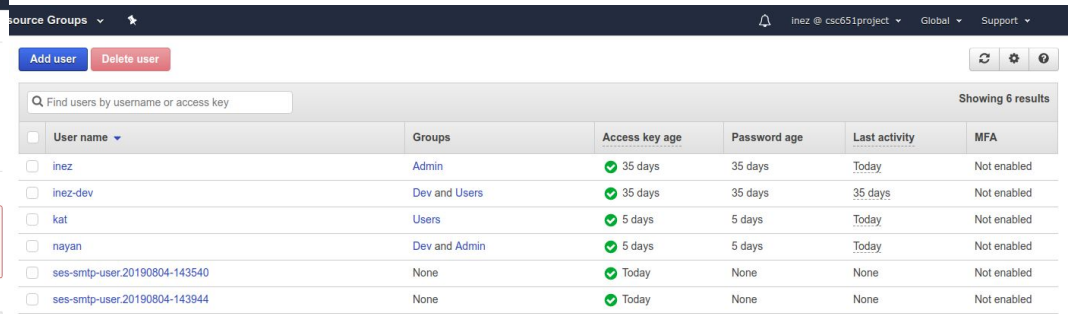
3. Set up permissions by groups (a set of permissions) and/or roles (a set of groups) as best practice.

   a. If no permissions is set up, then the user will not have access to any activity
   b. There are preloaded group policies in AWS to make it easier to get set up. Here is an example of an admin, and a user group policy.

4. The following are views from IAM page for different users, Group User, and Group Admin



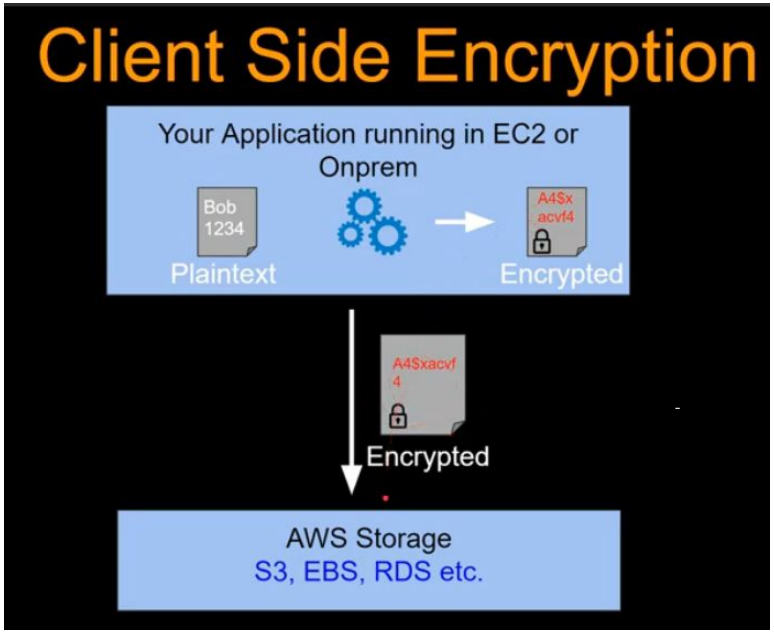
USER - READ ONLY

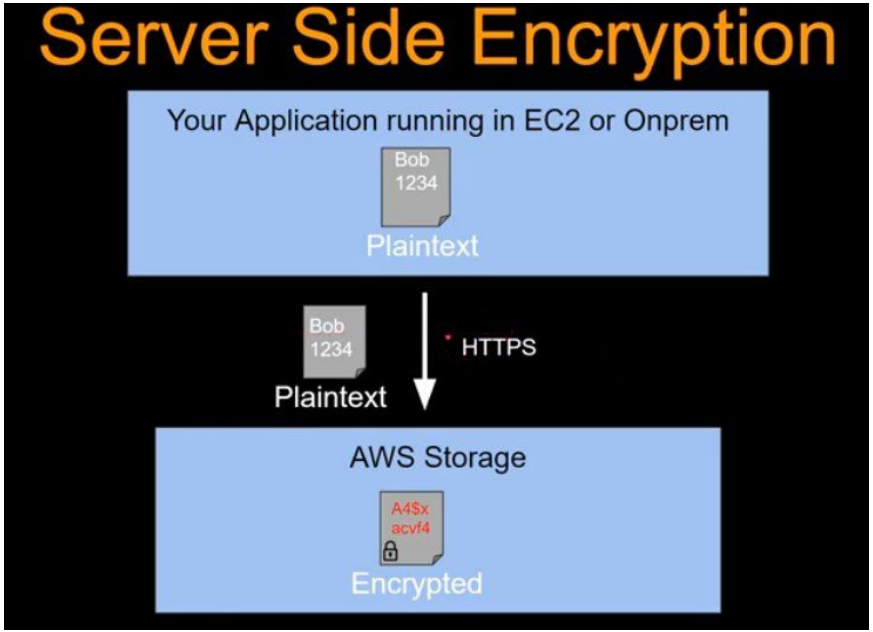


ADMIN -READ,WRITE

Setting up encryption on S3

AWS Encryption Models





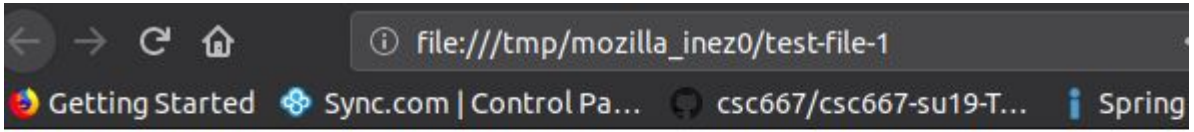
Upload to S3. Encrypt & decrypt with KMS

1. Open an S3 bucket with the region of your choice. If we are going to use it with AWS SES, then we need to choose one of the 3 regions available: Oregon, N. Virginia, Ireland.
2. Then with S3 bucket we can use various keys
    a. AWS managed key: You can use AWS-256, or AWS-KMS server side,
    b. Custom key: You can create customer master key (CMK) in the AWS Management Console or by using the CreateKey operation. You can list keys using AWS Command Line Interface: aws kms list-keys
    c. S3 and All KMS keys are not global. They are region specific.
3. We can use AWS Key Management Service to generate a key
4. Back in the S3 bucket we can upload and choose the AWS KMS key that we generated.
    a. We can use S3 console for AWS-256 and AWS KMS
    b. For AWS Custom Managed Keys (CMK) we will need to use the AWS SDK, using the AWS CLI

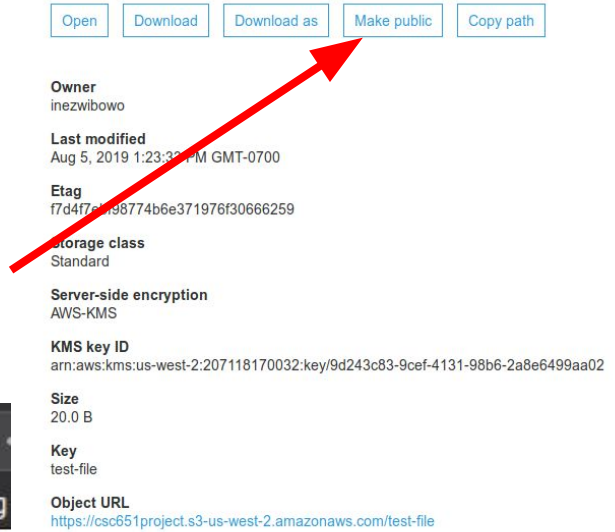
Encrypt and decrypt on S3 Console

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <Error>
    <Code>AccessDenied</Code>
    <Message>Access Denied</Message>
    <RequestId>F74C487E8B4E40D9</RequestId>
  - <HostId>
      ibuRNSln+ByoNus3FaYORkvxJoK0hmYdCWzc+DBAgmrgqtWP72rNRX/6xOSGaIrWSyX3q9t6Ssw=
    </HostId>
  </Error>
```



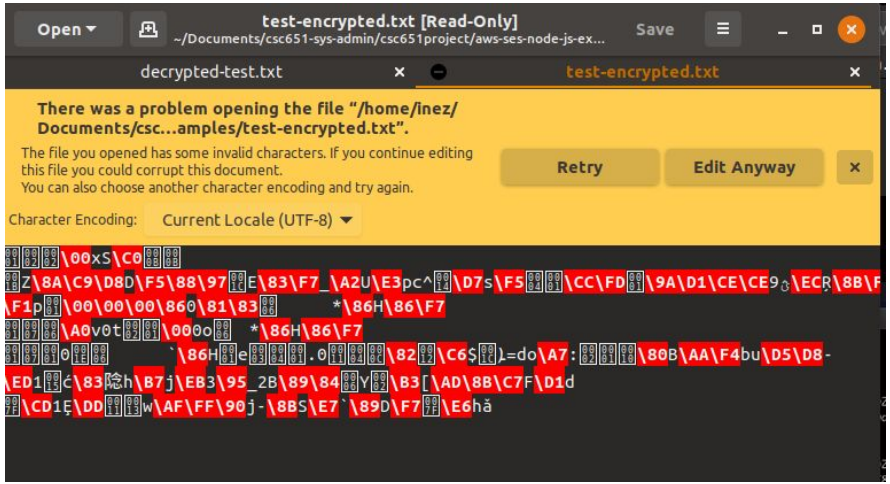
this is a test file



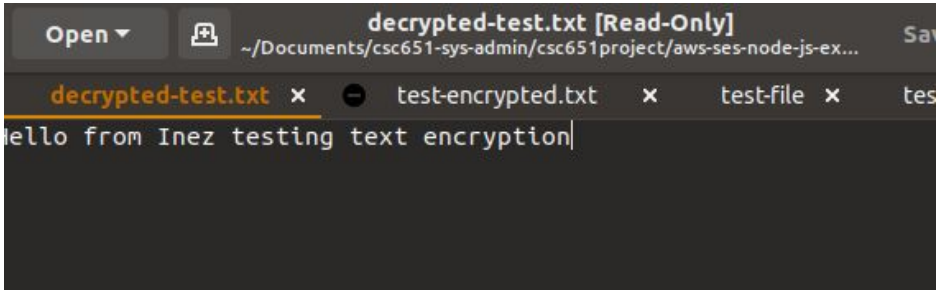
Encrypt and decrypt on AWS CLI

To encrypt
aws kms encrypt --key-id 62f8daa7-845b-466f-b4ce-de9391c503ce --plaintext "Hello from Inez testing text encryption" --output text --query CiphertextBlob | base64 --decode > test-encrypted.txt

To decrypt
aws kms decrypt --ciphertext-blob fileb://test-encrypted.txt --output text --query Plaintext | base64 --decode > decrypted-test.txt



Encrypted Text



Decrypted Text

References

AWS SES NodeJS Examples

AWS Security Basics - AWS KMS, Client/Server Side Encryption, CMK, Data Key, Real World Use | Demo

AWS Tutorial | AWS Security | IAM Masterclass | 14th May '19

AWS Official Documentation

Questions?
Thank you!