WHIZLABS

0 K

Home / AWS / Guided Lab / Encryption and Decryption Using KMS

Encryption and Decryption Using KMS

Level: **Intermediate**

Amazon EC2 AWS Key Management Service Amazon Web Services IAM

Required Points	10
Lab Duration	01:00:00
Average Start time	Less than a minute

Start Lab →

Need help?

- How to use Hands on Lab
- Troubleshooting Lab
- FAQs

Submit Feedback | Share

Lab Overview

Cloud Administrator

Security, Compute

Lab Details

1. This lab walks you through the steps to encrypt and decrypt and re-encrypt the data

2. Duration: **01:00:00 Hrs**

3. AWS Region: **US East (N. Virginia)**

Introduction

What is KMS ?

Definition: KMS stands for Key Management Service and it's a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data.

- AWS KMS integrated with the other AWS services including EBS, S3, Redshift, Elastic Transcoder, Amazon relational database and others to make it simple to encrypt your data with encryption keys that you manage.

Case Study

- AWS KMS or Key Management Service gives us a fully managed key management infrastructure. When we need to protect our data at rest, when we need to encrypt that data, and perform server side encryption, one of the most important aspects to that process is choosing where the keys come from.
- We can leverage many of the services in AWS that perform server side encryption, and can pull their encryption keys from KMS. So it is really important that we have a reliable key management infrastructure, because if we lose the keys, we lose the data.
- KMS performs three critical things. One, it stores customer master keys, and these customer master keys can be used to perform encryption directly on up to 4,096 bytes or we could also have KMS generate unique symmetric data keys. Those data keys could be either 128 bits or 256 bits, and then KMS would use the customer master key to encrypt the data key.
- Now, all the encryption is performed and all of these data keys are generated by hardware security modules. In fact, these hardware security modules are validated against FIPS 140-2, so they are trusted to be very secure. Ultimately, to put it simply, KMS you can think of as a multi-tenant API in front of hardware security modules. Now, we as customers don't have any direct access to the hardware security modules. We interact with the KMS API, which in turn interacts with the hardware security modules. Only AWS has access to the hardware.
- We can also count on KMS logging key use to CloudTrail, so if we do need to satisfy compliance or if we do need to inspect what keys were being used by what credentials, then we can look to CloudTrail.
- Another aspect of the security of KMS is the fact that the customer master key, the cryptographic material that makes the customer master key, never leaves KMS. We always refer to CMK by ID, and that cryptographic material is always held inside the

hardware security module. KMS also integrates with a number of other Amazon services. Just about any service that can perform server-side encryption within AWS can integrate with KMS, so that it can retrieve its data keys from KMS. KMS is also certified against a number of very stringent controls including SOC1, SOC2, SOC3, and PCI DDS level 1 among others. So taking a look here, if we wanted to leverage KMS to perform encryption,

- For example, if I have an application and I need my application to encrypt data. Then I would have a customer master key within KMS. And in fact, every account already includes the customer master key within KMS. Now, you can create additional customer master keys, but there is always a default one. And so, if we have an application that needs to perform encryption, what our application would do is to make a call to KMS.
- If what we are encrypting is larger than 4,096 bytes then we would ask KMS for a data key, and now we of course would get back a plain text data key. Not only do we want a plain text data key, but we would also want an encrypted data key. Keep in mind that the only thing that KMS stores is the customer master key. KMS will generate master keys, but it will not store them. It is up to us to store the data key. So we can have it generate a data key, we can then use the customer master key to encrypt the data key. So KMS will then give us both the plain text, and the encrypted data key. Our application can then use that plain text data key to encrypt our plain text data, and then we end up with encrypted data. It is then our responsibility to store both the encrypted data, and the encrypted data key. It is very important to keep in mind, because if we lose the key, we lose the data. This process of using one key to encrypt another key is called envelope encryption.

Lab Tasks

1. Sign in to AWS Management Console
2. Create a group for KMS users and attach a policy to the group.
3. Create 2 users for managing the KMS.
4. Creating a KMS Key
5. Launch an EC2 instance.
6. SSH into EC2 Instance
7. Perform KMS Encryption and Decryption.
8. Validation of the Lab

Launching Lab Environment

1. To launch the lab environment, Click on the **Start Lab** button.
2. Please wait until the cloud environment is provisioned. It will take less than a minute to provision.
3. Once the Lab is started, you will be provided with **IAM user name**, **Password**, **Access Key**, and **Secret Access Key**.

Note : You can only start one lab at any given time

About Us Subscription Instructions and Guidelines FAQ’s Contact Us