How it works

AWS KMS helps you centrally manage and securely store your keys. You can generate keys in AWS KMS or import them from your own key management infrastructure. You can submit data directly to AWS KMS to be encrypted, or decrypted. Other AWS services can use your keys on your behalf to protect data encryption keys that they use to protect your data.

Your keys never leave AWS KMS and you are always in control of your keys. You can set usage policies that determine which users can use your keys and what actions they can perform. All requests to use these keys are logged in AWS CloudTrail so that you can track who used which key, how and when.

Benefits and features

Fully managed

You can focus on your use of encryption while AWS handles durability, physical security, and policy enforcement to ensure that your keys are always available and safe.

Centralized key management

KMS presents a single control point to manage keys and define policies consistently across integrated AWS services and your own applications.

Integrated with AWS services

KMS is integrated with all major AWS services to simplify the use of encryption and to protect stored data through a common set of tools for controlling access.

Encryption for all your applications

KMS is integrated with the AWS Encryption SDK to help you build encryption and key management into your own applications.

Built-in auditing

KMS is integrated with AWS CloudTrail to provide a consolidated record of all key management activities and any attempt to use your keys.

No commitment

There is no commitment and no upfront charges. You only pay for the keys that you create and when you use them.

Secure

KMS uses FIPS 140-2 validated hardware security modules to generate and store your keys. Your keys can only be used inside these devices and can never leave them.

Compliance

The security and quality controls in AWS KMS have been certified by multiple compliance schemes to simplify your own compliance obligations.

Reliable

KMS uses highly durable storage and a resilient architecture to ensure that your keys are always available and are never lost.

General

Q: What is AWS Key Management Service (KMS)?

AWS KMS is a managed service that enables you to easily create and control the keys used for cryptographic operations. The service provides a highly available key generation, storage, management, and auditing solution for you to encrypt or digitally sign data within your own applications or control the encryption of data across AWS services.

Q: Why should I use AWS KMS?

If you are responsible for securing your data across AWS services, you should use it to centrally manage the encryption keys that control access to your data. If you are a developer who needs to encrypt data in your applications, you should use the AWS Encryption SDK with AWS KMS to easily generate, use and protect symmetric encryption keys in your code. If you are a developer who needs to digitally sign or verify data using asymmetric keys, you should use the service to create and manage the private keys you’ll need. If you’re looking for a scalable key management infrastructure to support your developers and their growing number of applications, you should use it to reduce your licensing costs and operational burden. If you’re responsible for proving data security for regulatory or compliance purposes, you should use it because it facilitates proving your data is consistently protected. It’s also in scope for a broad set of industry and regional compliance regimes.

Q: How do I get started with AWS KMS?

The easiest way is to get started using the service is to choose to encrypt your data within supported AWS services using AWS managed master keys that are automatically created in your account for each service. If you want full control over the management of your keys, including the ability to share access to keys across accounts or services, you can create your own customer master keys (CMKs) in AWS KMS. You can also use the CMKs that you create directly within your own applications. AWS KMS can be accessed from the KMS console that is grouped under Security, Identity and Compliance on the AWS Services home page of the AWS Console. AWS KMS APIs can also be accessed directly through the AWS KMS Command Line Interface or AWS SDK for programmatic access. AWS KMS APIs can also be used indirectly to encrypt data within your own applications by using the AWS Encryption SDK. Visit the Getting Started page to learn more.

Q: In what Regions is AWS KMS available?

Availability is listed on our global Products and Services by Region page.

Asymmetric keys are currently only available in Northern Virginia, Oregon, Sydney, Ireland, and Tokyo.

Q: What key management features are available in AWS KMS?

You can perform the following key management functions:

Create symmetric and asymmetric keys where the key material is only ever used within the service

Create symmetric keys where the key material is generated and used within a custom key store under your control\*

Import your own symmetric key material for use within the service

Create both symmetric and asymmetric data key pairs for local use within your applications

Define which IAM users and roles can manage keys

Define which IAM users and roles can use keys to encrypt and decrypt data

Choose to have keys that were generated by the service to be automatically rotated on an annual basis

Temporarily disable keys so they cannot be used by anyone

Re-enable disabled keys

Schedule the deletion of keys that you no longer use

Audit use of keys by inspecting logs in AWS CloudTrail

\* The use of custom key stores requires CloudHSM resources to be available in your account.

Q: How does AWS KMS work?

You start using the service by requesting the creation of a CMK. You control the lifecycle of the CMK as well as who can use or manage it. The key material for a CMK is generated within hardware security modules (HSMs) managed by AWS KMS. Alternatively, you can import key material from your own key management infrastructure and associate it with a CMK. You can also have the key material generated and used in an AWS CloudHSM cluster as a part of the custom key store feature in AWS KMS.

Once you have created a CMK using any of the three supported options, you can submit data directly to the service AWS KMS to be signed, verified, encrypted, or decrypted using these CMK. You set usage policies on these keys that determine which users can perform which actions under which conditions.

AWS services and client-side toolkits that integrate with AWS KMS use a method known as envelope encryption to protect your data. Under this method, AWS KMS generates data keys which are used to encrypt data locally in the AWS service or your application. The data keys are themselves encrypted under a CMK you define. Data keys are not retained or managed by AWS KMS. AWS services encrypt your data and store an encrypted copy of the data key along with the encrypted data. When a service needs to decrypt your data, it requests AWS KMS to decrypt the data key using your CMK. If the user requesting data from the AWS service is authorized to decrypt under your CMK, the AWS service will receive the decrypted data key from AWS KMS. The AWS service then decrypts your data and returns it in plaintext. All requests to use your CMKs are logged in AWS CloudTrail so you can understand who used which key under what context and when they used it.

Q: Where is my data encrypted if I use AWS KMS?

There are typically three scenarios for how data is encrypted using AWS KMS. Firstly, you can use AWS KMS APIs directly to encrypt and decrypt data using your CMKs stored in the service. Secondly, you can choose to have AWS services encrypt your data using your CMKs stored in the service. In this case data is encrypted using data keys that are protected by your CMKs. Thirdly, you can use the AWS Encryption SDK that is integrated with AWS KMS to perform encryption within your own applications, whether they operate in AWS or not.

Q: Which AWS cloud services are integrated with AWS KMS?

AWS KMS is seamlessly integrated with most other AWS services to make encrypting data in those services as easy as checking a box. In some cases data is encrypted by default using keys that are stored in AWS KMS but owned and managed by the AWS service in question. In many cases the CMKs are owned and managed by you within your account. Some services give you the choice of managing the keys yourself or allowing the service to manage the keys on your behalf. See the list of AWS services currently integrated with AWS KMS. See the AWS KMS Developer’s Guide for more information on how integrated services use AWS KMS.

Q: Why use envelope encryption? Why not just send data to AWS KMS to encrypt directly?

While AWS KMS does support sending data up to 4 KB to be encrypted directly, envelope encryption can offer significant performance benefits. When you encrypt data directly with AWS KMS it must be transferred over the network. Envelope encryption reduces the network load since only the request and delivery of the much smaller data key go over the network. The data key is used locally in your application or encrypting AWS service, avoiding the need to send the entire block of data to AWS KMS and suffer network latency.

Q: What’s the difference between a CMK I create and CMKs created automatically for me by other AWS services?

You have the option of selecting a specific CMK to use when you want an AWS service to encrypt data on your behalf. These are known as customer managed CMKs and you have full control over them. You define the access control and usage policy for each key and you can grant permissions to other accounts and services to use them. If you don’t specify a CMK, the service in question will create an AWS managed CMK the first time you try to create an encrypted resource within that service. AWS will manage the policies associated with AWS managed CMKs on your behalf. You can track AWS managed keys in your account and all usage is logged in AWS CloudTrail, but you have no direct control over the keys themselves.

Q: Why should I create my own customer master keys?

Creating your own CMK gives you more control than you have with AWS managed CMKs. When you create a symmetric customer managed CMK, you can choose to use key material generated by AWS KMS, generated within an AWS CloudHSM cluster (custom key store), or import your own key material. You can define an alias and description for the key and opt-in to have the key automatically rotated once per year if it was generated by AWS KMS. You also define all the permissions on the key to control who can use or manage the key. With asymmetric customer managed CMKs, there are a couple of caveats to management: the key material can only be generated within AWS KMS HSMs and there is no option for automatic key rotation.

Q: Can I bring my own keys to AWS KMS?

Yes. You can import a copy of your key from your own key management infrastructure to AWS KMS and use it with any integrated AWS service or from within your own applications. You cannot import asymmetric CMKs into AWS KMS.

Q: When would I use an imported key?

You can use an imported key to get greater control over the creation, lifecycle management, and durability of your key in AWS KMS. Imported keys are designed to help you meet your compliance requirements which may include the ability to generate or maintain a secure copy of the key in your infrastructure, and the ability to immediately delete the imported copy of the key from AWS infrastructure.

Q: What type of keys can I import?

You can import 256-bit symmetric keys.

Q: How is the key that I import into AWS KMS protected in transit?

During the import process, your key must be wrapped by an AWS KMS-provided public key using one of two RSA PKCS#1 schemes. This ensures that your encrypted key can only be decrypted by AWS KMS.

Q: What’s the difference between a key I import and a key I generate in AWS KMS?

There are two main differences:

You are responsible for maintaining a copy of your imported keys in your key management infrastructure so that you can re-import them at any time. AWS, however, ensures the availability, security, and durability of keys generated by AWS KMS on your behalf until you schedule the keys for deletion.

You may set an expiration period for an imported key. AWS KMS will automatically delete the key material after the expiration period. You may also delete imported key material on demand. In both cases the key material itself is deleted but the CMK reference in AWS KMS and associated metadata are retained so that the key material can be re-imported in the future. Keys generated by AWS KMS do not have an expiration time and cannot be deleted immediately; there is a mandatory 7 to 30 day wait period. All customer managed CMKs, irrespective of whether the key material was imported, can be manually disabled or scheduled for deletion. In this case the CMK itself is deleted, not just the underlying key material.

Q: Can I rotate my keys?

Yes. You can choose to have AWS KMS automatically rotate CMKs every year, provided that those keys were generated within AWS KMS HSMs. Automatic key rotation is not supported for imported keys, asymmetric keys, or keys generated in an AWS CloudHSM cluster using the AWS KMS custom key store feature. If you choose to import keys to AWS KMS or asymmetric keys or use a custom key store, you can manually rotate them by creating a new CMK and mapping an existing key alias from the old CMK to the new CMK.

Q: Do I have to re-encrypt my data after keys in AWS KMS are rotated?

If you choose to have AWS KMS automatically rotate keys, you don’t have to re-encrypt your data. AWS KMS automatically keeps previous versions of keys to use for decryption of data encrypted under an old version of a key. All new encryption requests against a key in AWS KMS are encrypted under the newest version of the key.

If you manually rotate your imported or custom key store keys, you may have to re-encrypt your data depending on whether you decide to keep old versions of keys available.

Q: Can I delete a key from AWS KMS?

Yes. You can schedule a customer master key and associated metadata that you created in AWS KMS for deletion, with a configurable waiting period from 7 to 30 days. This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it. The default waiting period is 30 days. You can cancel key deletion during the waiting period. The key cannot be used if it is scheduled for deletion until you cancel the deletion during the waiting period. The key gets deleted at the end of the configurable waiting period if you don’t cancel the deletion. Once a key is deleted, you can no longer use it. All data protected under a deleted master key is inaccessible.

For customer master keys with imported key material, you can delete the key material without deleting the customer master key id or metadata in two ways. First, you can delete your imported key material on demand without a waiting period. Second, at the time of importing the key material into the customer master key, you may define an expiration time for how long AWS can use your imported key material before it is deleted. You can re-import your key material into the customer master key if you need to use it again.

Q: What should I do if my imported key material has expired or I accidentally deleted it?

You can re-import your copy of the key material with a valid expiration period to AWS KMS under the original customer master key so it can be used.

Q: Can I be alerted that I need to re-import the key?

Yes. Once you import your key to a customer master key, you will receive an Amazon CloudWatch Metric every few minutes that counts down the time to expiration of the imported key. You will also receive an Amazon CloudWatch Event once the imported key under your customer master key expires. You can build logic that acts on these metrics or events and automatically re-imports the key with a new expiration period to avoid an availability risk.

Q: Can I use AWS KMS to help manage encryption of data outside of AWS cloud services?

Yes. AWS KMS is supported in AWS SDKs, AWS Encryption SDK, the Amazon DynamoDB Client-side Encryption, and the Amazon S3 Encryption Client to facilitate encryption of data within your own applications wherever they run. Visit the AWS Crypto Tools and Developing on AWS website for more information.

Q: Is there a limit to the number of keys I can create in AWS KMS?

You can create up to 10000 CMKs per account per region. As both enabled and disabled CMKs count towards the limit, we recommend deleting disabled keys that you no longer use. AWS managed CMKs created on your behalf for use within supported AWS services do not count against this limit. There is no limit to the number of data keys that can be derived using a CMK and used in your application or by AWS services to encrypt data on your behalf. You may request a limit increase for CMKs by visiting the AWS Support Center.

Q: What types of symmetric key types and algorithms are supported?

AWS KMS supports 256-bit keys when creating a CMK. Generated data keys returned to the caller can be 256-bit, 128-bit, or an arbitrary value up to 1024-bytes. When AWS KMS uses a 256-bit CMK on your behalf, the AES algorithm in Galois Counter Mode (AES-GCM) is used.

Q: What kind of asymmetric key types are supported?

AWS KMS supports the following asymmetric key types - RSA 2048, RSA 3072, RSA 4096, ECC NIST P-256, ECC NIST P-384, ECC NIST-521, and ECC SECG P-256k1.

Q: What kinds of asymmetric encryption algorithms are supported?

AWS KMS supports the RSAES\_OAEP\_SHA\_1 and RSAES\_OAEP\_SHA\_256 encryption algorithms with RSA 2048, RSA 3072, and RSA 4096 key types. Encryption algorithms cannot be used with the elliptic curve key types (ECC NIST P-256, ECC NIST P-384, ECC NIST-521, and ECC SECG P-256k1).

Q: What kinds of asymmetric signing algorithms are supported?

When using RSA key types, AWS KMS supports the RSASSA\_PSS\_SHA\_256, RSASSA\_PSS\_SHA\_384, RSASSA\_PSS\_SHA\_512, RSASSA\_PKCS1\_V1\_5\_SHA\_256, RSASSA\_PKCS1\_V1\_5\_SHA\_384, and RSASSA\_PKCS1\_V1\_5\_SHA\_512 signing algorithms.

When using elliptic curve key types, AWS KMS supports the ECDSA\_SHA\_256, ECDSA\_SHA\_384, and ECDSA\_SHA\_512 signing algorithms.

Q: Can symmetric CMKs be exported out of the service in plain text?

No. A symmetric CMK or the private portion of an asymmetric CMK cannot be exported in plain text from the HSMs. The public portion of an asymmetric CMK can be exported from the console or by calling the “GetPublicKey” API.

Q: Can data keys and data key pairs be exported out of the HSMs in plain text?

Yes. The symmetric data keys can be exported using either the “GenerateDataKey” API or the “GenerateDataKeyWithoutPlaintext” API. And the private and public portion of asymmetric data key pairs can both be exported out of AWS KMS using either the “GenerateDataKeyPair” API or the “GenerateDataKeypairWithoutPlaintext” API.

Q: How are data keys and data key pairs protected for storage outside the service?

The symmetric data key or the private portion of the asymmetric data key is encrypted under the symmetric CMK you define when you request AWS KMS to generate the data key.

Q: How do I use the public portion of an asymmetric CMK?

The public portion of the asymmetric key material is generated in AWS KMS and can be used for digital signature verification by calling the “Verify” API, or for public key encryption by calling the “Encrypt” API. The public key can also be used outside of AWS KMS for verification or encryption. You can call the GetPublicKey API to retrieve the public portion of the asymmetric CMK.

Q: What is the size limit for data sent to AWS KMS for asymmetric operations?

The size limit is 4KB. If you want to digitally sign data larger than 4KB, you have the option to create a message digest of the data and send it to AWS KMS. The digital signature is created over the digest of the data and returned. You specify whether you are sending the full message or a message digest as a parameter in the Sign API request. Any data submitted to the Encrypt, Decrypt, or Re-Encrypt APIs that require use of asymmetric operations must also be less than 4KB.

Q: How can I distinguish between asymmetric or symmetric CMKs I have created?

In the console, each key will have a new field called “Key Type”. It will have either the value “Asymmetric Key” or “Symmetric Key” to indicate the type of key. The “DescribeKey” API will return a “KeyUsage” field that will specify if the key can be used to sign or encrypt.

Q: Is automatic rotation of asymmetric CMKs supported?

No. Automatic key rotation is not supported for asymmetric CMKs. You can manually rotate them by creating a new CMK and mapping an existing key alias from the old CMK to the new CMK.

Q: Can a single asymmetric CMK be used for both encryption and signing?

No. When creating a CMK, you must specify whether the key can be used for decrypt or sign operations. An RSA key type can be used for signing or encryption operations, but not both. Elliptic curve key types can only be used for signing operations.

Q: Are there service limits related to asymmetric keys?

Yes. The request per second rate limits are different for different key types and algorithms. Please refer to the AWS KMS limits page for details.

Q: Do asymmetric keys work with AWS KMS custom key stores or the Import Key feature?

No. You cannot use the custom key store functionality with asymmetric keys nor can you import asymmetric keys into AWS KMS.

Q: Can I use asymmetric CMKs for digital signing applications that require digital certificates?

Not directly. AWS KMS doesn’t store or associate digital certificates with asymmetric CMKs it creates. You could choose to have a certificate authority such as ACM PCA issue a certificate for the public portion of your asymmetric CMK. This will allow the entities that are consuming your public key to verify that the public key indeed belongs to you.

Q: For what use scenarios should I use ACM Private Certificate Authority vs. AWS KMS?

The primary reason to use the ACM Private Certificate Authority (CA) service is to provide a public key infrastructure (PKI) for the purpose of identifying entities and securing network connections. PKI provides processes and mechanisms, primarily using X.509 certificates, to put structure around public key cryptographic operations. Certificates provide an association between an identity and a public key. The certification process in which a certificate authority issues a certificate allows the trusted certificate authority to assert the identity of another entity by signing a certificate. PKI provides identity, distributed trust, key lifecycle management, and certificate status vended through revocation. These functions add important processes and infrastructure to the underlying asymmetric cryptographic keys and algorithms provided by AWS KMS.

ACM Private CA allows you to issue certificates to identify web and application servers, service meshes, VPN users, internal API endpoints, and IoT devices. Certificates let you establish the identity of these resources and create encrypted TLS/SSL communications channels. If you are considering using asymmetric keys for TLS termination on web or application servers, Elastic Load Balancers, API Gateway endpoints, EC2 instances or containers, you should consider using ACM Private CA for issuing certificates and providing a PKI infrastructure.

In contrast, AWS KMS lets you generate, manage, and use asymmetric keys for digital signing and/or encryption operations that don’t require certificates. While certificates can enable verification of sender and recipient identity between untrusted parties, the kind of raw asymmetric operations offered by AWS KMS are typically useful when you have other mechanisms to prove identity or don’t need to prove it at all to get the security benefit you desire.

Q: Can I use my applications’ cryptographic API providers such as OpenSSL, JCE, Bouncy Castle, or CNG with AWS KMS?

There is no native integration offered by AWS KMS for any other cryptographic API providers. You must use AWS KMS APIs directly or through the AWS SDK to integrate signing and encryption capabilities into your applications.

Q: Does AWS KMS offer a Service Level Agreement (SLA)?

Yes. The AWS KMS SLA provides for a service credit if a customer's monthly uptime percentage is below our service commitment in any billing cycle.

Custom Key Store

Q: What is a custom key store?

The AWS KMS custom key store feature combines the controls provided by AWS CloudHSM with the integration and ease of use of AWS KMS. You can configure your own CloudHSM cluster and authorize AWS KMS to use it as a dedicated key store for your keys rather than the default AWS KMS key store. When you create keys in AWS KMS you can chose to generate the key material in your CloudHSM cluster. CMKs that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all AWS KMS operations that use those keys are only performed in your HSMs. In all other respects CMKs stored in your custom key store are consistent with other AWS KMS CMKs.

Additional guidance for deciding if using a custom key store it is right for you can be found in this blog.

Q: Why would I need to use a custom key store?

Since you control your AWS CloudHSM cluster, you have the option to manage the lifecycle of your CMKs independently of AWS KMS. There are four reasons why you might find a custom key store useful. Firstly, you might have keys that are explicitly required to be protected in a single tenant HSM or in an HSM over which you have direct control. Secondly, you might have keys that are required to be stored in an HSM that has been validated to FIPS 140-2 level 3 overall (the HSMs used in the standard AWS KMS key store are either validated or in the process of being validated to level 2 with level 3 in multiple categories). Thirdly, you might need the ability to immediately remove key material from AWS KMS and to prove you have done so by independent means. Finally, you might have a requirement to be able to audit all use of your keys independently of AWS KMS or AWS CloudTrail.

Q: Do custom key stores affect how keys are managed?

There are two differences when managing keys in a custom key store compared to the default AWS KMS key store. You cannot import key material into your custom key store and you cannot have AWS KMS automatically rotate keys. In all other respects, including the type of keys that can be generated, the way that keys use aliases and how policies are defined, keys that are stored in a custom key store are managed in the same way as any other AWS KMS customer managed CMK.

Q: Can I use a custom key store to store an AWS managed customer master key?

No, only customer managed CMKs can be stored and managed in an AWS KMS custom key store. AWS managed CMKs that are created on your behalf by other AWS services to encrypt your data are always generated and stored in the AWS KMS default key store.

Q: Do custom key stores affect how keys are used?

No, API requests to AWS KMS to use a CMK to encrypt and decrypt data are handled in the same way. Authentication and authorization processes operate independently of where the key is stored. All activity using a key in a custom key store is also logged to AWS CloudTrail in the same way. However, the actual cryptographic operations happen exclusively in either the custom key store or the default AWS KMS key store.

Q: How can I audit the use of keys in a custom key store?

In addition to the activity that is logged to AWS CloudTrail by AWS KMS the use of a custom key store provides three further auditing mechanisms. First, AWS CloudHSM also logs all API activity to CloudTrail, for example to create clusters and to add or remove HSMs. Second, each cluster also captures its own local logs to record user and key management activity. Third, each CloudHSM instance copies the local user and key management activity logs to AWS CloudWatch.

Q: What impact does using a custom key store have on availability of keys?

The use of an AWS KMS custom key store makes you responsible for ensuring that your keys are available for use by AWS KMS. Errors in configuration of CloudHSM and accidental deletion of key material within an AWS CloudHSM cluster could impact availability. The number of HSMs you use and your choice of availability zones (AZs) can also affect the resilience of your cluster. As in any key management system, it is important to understand how the availability of keys can impact the recovery of your encrypted data.

Q: What are the performance limitations associated with a custom key store?

The rate at which keys stored in an AWS KMS custom key store can be used via AWS KMS API calls are lower than for keys stored in the default AWS KMS key store. See the AWS KMS Developer Guide for the current performance limits.

Q: What are the costs associated with using a custom key store?

AWS KMS prices are unaffected by the use of a custom key store. However, each custom key store does require that your AWS CloudHSM cluster contains at least two HSMs. These HSMs are charged at the standard AWS CloudHSM prices. There are no additional charges for using a custom key store.

Q: What additional skills and resources are required to configure a custom key stores?

AWS KMS users that wish to use a custom key store will need to set up an AWS CloudHSM cluster, add HSMs, manage HSMs users and potentially restore HSMs from backup. These are security sensitive tasks and you should ensure that you have the appropriate resources and organizational controls in place.

Q: Can I import keys into a custom key store?

No, the ability to import your own key material into an AWS KMS custom key store is not supported. Keys that are stored in a custom key store can only be generated in the HSMs that form your AWS CloudHSM cluster.

Q: Can I migrate keys between the default AWS KMS keys store and a custom key store?

No, the ability to migrate keys between the different types of AWS KMS key store is not currently supported. All keys must be created in the key store in which they will be used, except in situations where you import you own key material into the default AWS KMS key store.

Q: Can I rotate keys stored in a custom key store?

The ability to automatically rotate key material in an AWS KMS custom key store is not supported. Key rotation must be performed manually by creating new keys and re-mapping AWS KMS key aliases used by your application code to use the new keys for future encryption operations.

Q: Can I use my AWS CloudHSM cluster for other applications?

Yes, AWS KMS does not require exclusive access to your AWS CloudHSM cluster. If you already have a cluster you can use it as a custom key store and continue to use it for your other applications. However, if your cluster is supporting high, non-AWS KMS, workloads you may experience reduced throughput for operations using CMKs in your custom key store. Similarly, a high AWS KMS request rate to your custom key store could impact your other applications.

Q: How can I learn more about AWS CloudHSM?

Visit the AWS CloudHSM web site for an overview of the service and for more details on configuring and using the service read the AWS CloudHSM User Guide.

Billing

Q: How will I be charged and billed for my use of AWS KMS?

With AWS KMS, you pay only for what you use, there is no minimum fee. There are no set-up fees or commitments to begin using the service. At the end of the month, your credit card will automatically be charged for that month’s usage.

You are charged for all CMKs you create and for API requests made to the service each month above a free tier.

For current pricing information, please visit the AWS KMS pricing page.

Q: Is there a free tier?

Yes. With the AWS Free Usage Tier you can get started with AWS KMS for free\* in all regions. AWS managed CMKs that are created on your behalf by AWS services are free to store in your account. There is a free tier for usage that provides a free number of requests to the service each month. For current information on pricing, including the free tier, please visit the AWS KMS pricing page.

\*API requests involving asymmetric CMKs and API requests to the GenerateDataKeyPair and GenerateDataKeyPairWithoutPlaintext APIs are excluded from the free tier.

Q: Do your prices include taxes?

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS services is subject to Japanese Consumption Tax. You can learn more here.

Security

Q: Who can use and manage my keys in AWS KMS?

AWS KMS enforces usage and management policies that you define. You choose to allow AWS Identity and Access Management (IAM) users and roles from your account or other accounts to use and manage your keys.

Q: How does AWS secure the CMKs that I create?

AWS KMS is designed so that no one, including AWS employees, can retrieve your plaintext CMKs from the service. The service uses hardware security modules (HSMs) that have been validated under FIPS 140-2, or are in the process of being validated, to protect the confidentiality and integrity of your keys regardless of whether you use AWS KMS or AWS CloudHSM to create your keys or you import them into the service yourself. Your plaintext CMKs never leave the HSMs, are never written to disk and are only ever used in the volatile memory of the HSMs for the time needed to perform your requested cryptographic operation. AWS KMS keys are never transmitted outside of the AWS regions in which they were created. Updates to software on the service hosts and to the AWS KMS HSM firmware is controlled by multi-party access control that is audited and reviewed by an independent group within Amazon as well as a NIST-certified lab in compliance with FIPS 140-2.

More details about these security controls can be found in the AWS KMS Cryptographic Details whitepaper. You can also review the FIPS 140-2 certificate for AWS KMS HSM along with the associated Security Policy to get more details about how AWS KMS HSM meets the security requirements of FIPS 140-2. In addition, you can download a copy of the Service Organization Controls (SOC) report from AWS Artifact to learn more about security controls used by the service to protect your CMKs.

Q: How do I migrate my existing CMKs to use FIPS 140-2 validated HSMs?

All CMKs regardless of their creation date or origin are automatically protected using HSMs that have been validated under FIPS 140-2, or are in the process of being validated. No action is required on your part to use the FIPS 140-2 validated HSMs.

Q: Which AWS regions have FIPS 140-2 validated HSMs?

FIPS 140-2 validated HSMs are available in all AWS regions where AWS KMS is offered.

Q: What is the difference between the FIPS 140-2 validated endpoints and the FIPS 140-2 validated HSMs in AWS KMS?

AWS KMS is a two-tier service. The API endpoints receive client requests over an HTTPS connection using only TLS ciphersuites that support perfect forward secrecy. These API endpoints authenticate and authorize the request before passing the request for a cryptographic operation to the AWS KMS HSMs or your AWS CloudHSM cluster if you’re using the KMS custom key store feature.

Q: How do I make API requests to AWS KMS using the FIPS 140-2 validated endpoints?

You configure your applications to connect to the unique regional FIPS 140-2 validated HTTPS endpoints. AWS KMS FIPS 140-2 validated HTTPS endpoints are powered by the OpenSSL FIPS Object Module. You can review the security policy of the OpenSSL module at https://www.openssl.org/docs/fips/SecurityPolicy-2.0.13.pdf. FIPS 140-2 validated API endpoints are available in all commercial regions where AWS KMS is available.

Q: Can I use AWS KMS to help me comply with the encryption and key management requirements in the Payment Card Industry Data Security Standard (PCI DSS 3.2.1)?

Yes. AWS KMS has been validated as having the functionality and security controls to help you meet the encryption and key management requirements (primarily referenced in sections 3.5 and 3.6) of the PCI DSS 3.2.1.

For more details on PCI DSS compliant services in AWS, you can read the PCI DSS FAQs.

Q: How does AWS KMS secure the data keys I export and use in my application?

You can request that AWS KMS generate data keys and return them for use in your own application. The data keys are encrypted under a master key you define in AWS KMS so that you can safely store the encrypted data key along with your encrypted data. Your encrypted data key (and therefore your source data) can only be decrypted by users with permissions to use the original master key to decrypt your encrypted data key.

Q: Can I export a CMK and use it in my own applications?

No. CMKs are created and used only within the service to help ensure their security, enable your policies to be consistently enforced, and provide a centralized log of their use.

Q: What geographic region are my keys stored in?

Keys generated by AWS KMS are only stored and used in the region in which they were created. They cannot be transferred to another region. For example; keys created in the EU-Central (Frankfurt) region are only stored and used within the EU-Central (Frankfurt) region.

Q: How can I tell who used or changed the configuration of my keys in AWS KMS?

Logs in AWS CloudTrail will show all AWS KMS API requests, including both management requests (e.g. create, rotate, disable, policy edits) and cryptographic requests (e.g. encrypt/decrypt). Turn on AWS CloudTrail in your account to view these logs.

Q: How does AWS KMS compare to AWS CloudHSM?

AWS CloudHSM provides you with a FIPS 140-2 Level 3 overall validated single-tenant HSM cluster in your Amazon Virtual Private Cloud (VPC) to store and use your keys. You have exclusive control over how your keys are used via an authentication mechanism independent from AWS. You interact with keys in your AWS CloudHSM cluster similar to the way you interact with your applications running in Amazon EC2. You can use AWS CloudHSM to support a variety of use cases, such as Digital Rights Management (DRM), Public Key Infrastructure (PKI), document signing, and cryptographic functions using PKCS#11, Java JCE, or Microsoft CNG interfaces.

AWS KMS allows you to create and control the encryption keys used by your applications and supported AWS services in multiple regions around the world from a single console. The service uses an FIPS HSM that has been validated under FIPS 140-2, or are in the process of being validated, to protect the security of your keys. Centralized management of all your keys in AWS KMS lets you enforce who can use your keys under which conditions, when they get rotated, and who can manage them. AWS KMS integration with AWS CloudTrail gives you the ability to audit the use of your keys to support your regulatory and compliance activities. You interact with AWS KMS from your applications using the AWS SDK if you want to call the service APIs directly, via other AWS services that are integrated with AWS KMS or by using the AWS Encryption SDK if you want to perform client-side encryption.