Notes 2

**Cloud HSM**

AWS CloudHSM provides hardware security modules in the AWS Cloud. A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys.

When you use an HSM from AWS CloudHSM, you can perform a variety of cryptographic tasks:

* Generate, store, import, export, and manage cryptographic keys, including symmetric keys and asymmetric key pairs.
* Use symmetric and asymmetric algorithms to encrypt and decrypt data.
* Use cryptographic hash functions to compute message digests and hash-based message authentication codes (HMACs).
* Cryptographically sign data (including code signing) and verify signatures.
* Generate cryptographically secure random data.

If you want a managed service for creating and controlling your encryption keys but you don't want or need to operate your own HSM, consider using [AWS Key Management Service](https://aws.amazon.com/kms/).

To learn more about what you can do with AWS CloudHSM, see the following topics. When you are ready to get started with AWS CloudHSM, see [Getting started](https://docs.aws.amazon.com/cloudhsm/latest/userguide/getting-started.html).

**Use cases:**

* Offload the SSL/TLS processing for web servers
* Protect the private keys for an issuing certificate authority (CA)
* Enable transparent data encryption (TDE) for Oracle databases

**HSM Cluster**AWS CloudHSM provides hardware security modules (HSMs) in a *cluster*. A cluster is a collection of individual HSMs that AWS CloudHSM keeps in sync. You can think of a cluster as one logical HSM. When you perform a task or operation on one HSM in a cluster, the other HSMs in that cluster are automatically kept up to date.

You can create a cluster that has from 1 to 28 HSMs (the [default limit](https://docs.aws.amazon.com/cloudhsm/latest/userguide/limits.html) is 6 HSMs per AWS account per AWS Region). You can place the HSMs in different Availability Zones in an AWS Region. Adding more HSMs to a cluster provides higher performance. Spreading clusters across Availability Zones provides redundancy and high availability.

Making individual HSMs work together in a synchronized, redundant, highly available cluster can be difficult, but AWS CloudHSM does some of the undifferentiated heavy lifting for you. You can add and remove HSMs in a cluster and let AWS CloudHSM keep the HSMs connected and in sync for you.

**Cluster architecture**

When you create a cluster, you specify an Amazon Virtual Private Cloud (VPC) in your AWS account and one or more subnets in that VPC. We recommend that you create one subnet in each Availability Zone (AZ) in your chosen AWS Region. To learn how, see [Create a private subnet](https://docs.aws.amazon.com/cloudhsm/latest/userguide/create-subnets.html).

Each time you create an HSM, you specify the cluster and Availability Zone for the HSM. By putting the HSMs in different Availability Zones, you achieve redundancy and high availability in case one Availability Zone is unavailable.

When you create an HSM, AWS CloudHSM puts an elastic network interface (ENI) in the specified subnet in your AWS account. The elastic network interface is the interface for interacting with the HSM. The HSM resides in a separate VPC in an AWS account that is owned by AWS CloudHSM. The HSM and its corresponding network interface are in the same Availability Zone.

To interact with the HSMs in a cluster, you need the AWS CloudHSM client software. Typically you install the client on Amazon EC2 instances, known as client instances, that reside in the same VPC as the HSM ENIs, as shown in the following figure. That's not technically required though; you can install the client on any compatible computer, as long as it can connect to the HSM ENIs. The client communicates with the individual HSMs in your cluster through their ENIs.

The following figure represents an AWS CloudHSM cluster with three HSMs, each in a different Availability Zone in the VPC.


        Architecture of an AWS CloudHSM cluster with three HSMs.
      

**Cluster synchronization**

In an AWS CloudHSM cluster, AWS CloudHSM keeps the keys on the individual HSMs in sync. You don't need to do anything to synchronize the keys on your HSMs. To keep the users and policies on each HSM in sync, update the AWS CloudHSM client configuration file before you [manage HSM users](https://docs.aws.amazon.com/cloudhsm/latest/userguide/manage-hsm-users.html). For more information, see [Keep HSM users in snc](https://docs.aws.amazon.com/cloudhsm/latest/userguide/troubleshooting-keep-hsm-users-in-sync.html).

When you add a new HSM to a cluster, AWS CloudHSM makes a backup of all keys, users, and policies on an existing HSM. It then restores that backup onto the new HSM. This keeps the two HSMs in sync.

If the HSMs in a cluster fall out of synchronization, AWS CloudHSM automatically resynchronizes them. To enable this, AWS CloudHSM uses the credentials of the [appliance user](https://docs.aws.amazon.com/cloudhsm/latest/userguide/manage-hsm-users.html#understanding-users). This user exists on all HSMs provided by AWS CloudHSM and has limited permissions. It can get a hash of objects on the HSM and can extract and insert masked (encrypted) objects. AWS cannot view or modify your users or keys and cannot perform any cryptographic operations using those keys.

**Cluster high availability and load balancing**

When you create an AWS CloudHSM cluster with more than one HSM, you automatically get load balancing. Load balancing means that the [AWS CloudHSM client](https://docs.aws.amazon.com/cloudhsm/latest/userguide/client-tools-and-libraries.html) distributes cryptographic operations across all HSMs in the cluster based on each HSM's capacity for additional processing.

When you create the HSMs in different AWS Availability Zones, you automatically get high availability. High availability means that you get higher reliability because no individual HSM is a single point of failure. We recommend that you have a minimum of two HSMs in each cluster, with each HSM in different Availability Zones within an AWS Region.

For example, the following figure shows an Oracle database application that is distributed to two different Availability Zones. The database instances store their master keys in a cluster that includes an HSM in each Availability Zone. AWS CloudHSM automatically synchronizes the keys to both HSMs so that they are immediately accessible and redundant.


        An application and AWS CloudHSM cluster distributed to two Availability Zones for high
          availability.
      

**AWS CloudHSM cluster backups**


      AWS CloudHSM cluster backups encrypted in a service-controlled Amazon S3 bucket.
    

AWS CloudHSM makes periodic backups of the users, keys, and policies in the cluster. The service stores backups in a service-controlled Amazon Simple Storage Service (Amazon S3) bucket in the same region as your cluster. Backups are secure, durable, and updated on a predictable schedule.

**Security of backups**

When AWS CloudHSM makes a backup from the HSM, the HSM encrypts all of its data before sending it to AWS CloudHSM. The data never leaves the HSM in plaintext form.

To encrypt its data, the HSM uses a unique, ephemeral encryption key known as the ephemeral backup key (EBK). The EBK is an AES 256-bit encryption key generated inside the HSM when AWS CloudHSM makes a backup. The HSM generates the EBK, then uses it to encrypt the HSM's data with a FIPS-approved AES key wrapping method that complies with [NIST special publication 800-38F](http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38F.pdf). Then the HSM gives the encrypted data to AWS CloudHSM. The encrypted data includes an encrypted copy of the EBK.

To encrypt the EBK, the HSM uses another encryption key known as the persistent backup key (PBK). The PBK is also an AES 256-bit encryption key. To generate the PBK, the HSM uses a FIPS-approved key derivation function (KDF) in counter mode that complies with [NIST special publication 800-108](http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf). The inputs to this KDF include the following:

* A manufacturer key backup key (MKBK), permanently embedded in the HSM hardware by the hardware manufacturer.
* An AWS key backup key (AKBK), securely installed in the HSM when it's initially configured by AWS CloudHSM.

The encryption processes are summarized in the following figure. The backup encryption key represents the persistent backup key (PBK) and the ephemeral backup key (EBK).


        A summary of the encryption keys that are used to encrypt AWS CloudHSM backups.
      

AWS CloudHSM can restore backups onto only AWS-owned HSMs made by the same manufacturer. Because each backup contains all users, keys, and configuration from the original HSM, the restored HSM contains the same protections and access controls as the original. The restored data overwrites all other data that might have been on the HSM prior to restoration.

A backup consists of only encrypted data. Before the service stores a backup in Amazon S3, the service encrypts the backup again using AWS Key Management Service (AWS KMS).

**Durability of backups**

AWS CloudHSM stores cluster backups in an Amazon S3 bucket that the service controls. Backups have a 99.999999999% durability level, the same as any object stored in Amazon S3.