vSAN Encryption Services

Understanding Encryption Offerings with vSAN using VMware Cloud Foundation 9.1

May 6, 2026

vmware by Broadcom

Table of Contents

Introduction

Data encryption is a common technique used in environments that requires additional levels of security. It consists of a process to ensure that data can only be consumed by systems that have appropriate levels of access. Approved systems must have and use the appropriate cryptographic keys to encrypt and decrypt the data. Systems that do not have the keys will not be able to consume the data in any meaningful way, as it will remain encrypted in accordance with the commonly used Advanced Encryption Standard (AES) from the National Institute of Standards and Technology, or NIST.

An introduction to vSAN Encryption Services

VMware vSAN in VCF offers two forms of cluster-based encryption services.

- **vSAN Data-at-Rest Encryption.** This vSAN feature securely encrypts all vSAN data as it lands on persistent storage devices in the hosts. The data is non decrypted until the process of a read operation requires the data to be read.
- **vSAN Data-in-Transit Encryption.** This vSAN feature securely encrypts all vSAN traffic in transit across hosts. In vSAN for VCF 9.1 and later, it supports the ability to encrypt all vSAN traffic from vSphere clusters mounting the datastore of a vSAN storage cluster. This can be enabled on the vSphere cluster.

These services can be turned on or off on a per-cluster basis, and used independently or together, and do not need or use self-encrypting drives. A list of frequently asked questions on vSAN Encryption Services can be found in the "Security" section of the vSAN FAQs.

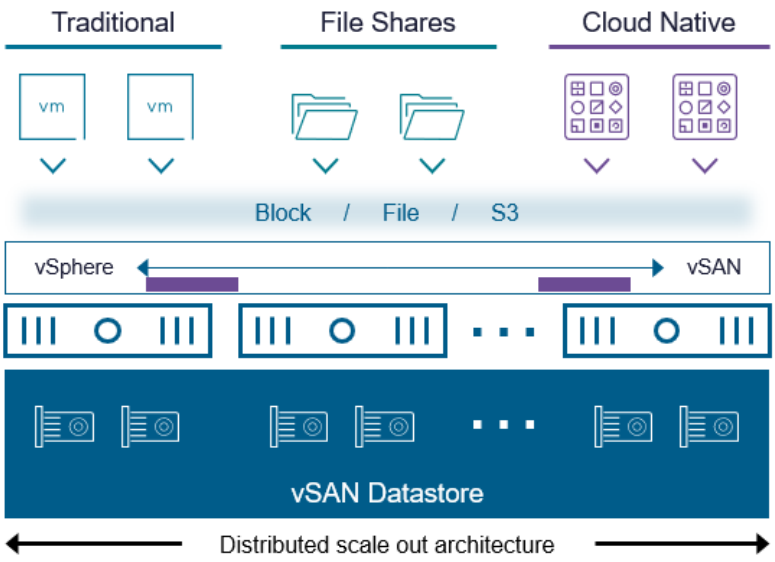
Note that vSphere VM Encryption (sometimes referred to as "VM Encrypt") is an independent feature of vSphere that can be used on all types of storage solutions, including vSAN. They do, however, share many commonalities. The advantages and disadvantages of using VM Encrypt on vSAN will be discussed later in this document.

Scope of Topics

The information provided in this document will assume the use of vSAN 9.1, and/or VMware Cloud Foundation (VCF) 9.1. VCF deployments may have additional requirements and support limitations that fall outside of the scope of this document. **For VCF environments, please refer to the Administration Guide for VMware Cloud Foundation for guidance as it relates to VCF.**

Hypervisor Integration

vSAN Encryption services in the Express Storage Architecture (ESA) (and the older vSAN OSA) use a native VMkernel Cryptographic module in vSphere to provide the highest levels of security and compliance. VMware achieves FIPS 140-3 validation under the Cryptographic Module Validation Program (CMVP). The CMVP is a joint program between NIST and the Communications Security Establishment (CSE). FIPS 140-3 is a Cryptographic Modules Standards that governs security requirements in 11 areas relating to the design and implementation of a cryptographic module. vSphere and vSAN use the validated Cryptographic Module for all encryption services. The hypervisor achieved FIPS 140-3 validation in September of 2024.



The VMware VMkernel Cryptographic Module has successfully satisfied all requirements and has gone through required algorithms and operational testing, rigorous review by CMVP and third party laboratory before being awarded certificate number 3073 by the CMVP. Since the VMware VMkernel Cryptographic Module is part of the ESXi kernel, it can easily provide FIPS approved cryptographic services to various VMware products and services. Virtual machines encrypted with vSphere's VM Encryption or vSAN Encryption Services work with all vSphere supported Guest Operating Systems and Virtual Hardware versions, and do not allow access to encryption keys by the Guest OS.

Implementation Considerations

Capacity Utilization

vSAN Data-at-Rest encryption is **designed to not interfere with any space efficiency techniques like data compression, deduplication and erasure coding**. In vSAN for VCF 9.1, global deduplication is fully supported when using vSAN Data-at-Rest encryption, and does not have any negative impact on deduplication rates. vSphere VM encryption, which is another technique to encrypt data, would interfere with these space efficiency techniques, and is not recommended as a method of securing data in a vSAN environment.

vSAN ESA implements encryption near the top of the vSAN stack, but after compression has occurred. This helps reduce the amount of data encrypted, reduces the amplification of the encryption processes. For more information, see the post: "Cluster Level Encryption with the vSAN Express Storage Architecture."

Recommendation: If you want to secure your environment using encryption, use vSAN encryption services (Data-at-Rest Encryption and Data-in-Transit Encryption) rather than vSphere's VM Encryption. This will allow you to take advantage of space efficiency techniques like vSAN Global Deduplication even while data is encrypted. VM Encryption would negate the benefits of deduplication, lower performance and introduce complexity into an environment. We also recommend not combining vSphere's VM Encryption with vSAN Encryption for the same reasons.

Compute and Performance Impacts

Enterprise storage arrays that offer encryption capabilities will perform this process using the CPU processing on the array controllers. Note that these encryption capabilities are generally limited to "at-rest" encryption. Three-tier architectures using storage arrays typically do not encrypt the storage fabric that transports the storage traffic.

vSAN can use any server with hardware that is certified on the Broadcom Compatibility Guide (BCG) for vSAN. To encrypt and decrypt data efficiently, vSAN uses the Advanced Encryption Standard-New Instructions (AES-NI) CPU offloading capabilities provided by current generation server processors. These advanced instruction offloading capabilities have been present in both Intel and AMD server processors for several years. By offloading encryption tasks using AES-NI processor

capabilities, vSAN can easily accomplish encryption with minimal additional overhead to vSphere hosts. To better understand the additional overhead on hosts and how it may or may not affect VMs, see the post: Performance when using vSAN Encryption Services. **The impact on overhead and performance will vary based on the workload and cluster configuration. vSAN ESA has demonstrated to be much more efficient at encryption than vSAN OSA.**

If you are concerned about the impact of performance of vSAN Encryption Services in your environment, we encourage you to test against real workloads. Synthetic tests may lead to skewed results. For more information, see the "vSAN Encryption Services Troubleshooting" section in this document.

Device Loss or Theft

Data encryption that occurs at a device level (e.g. self-encrypting drives) protects against the physical theft or loss of the device that contains the virtual machine's data. Loss can occur from intentional drive theft but does not protect from powering off a virtual machine, or cloning the virtual machine, and then downloading that virtual machine to a USB or other portable media device from an administrative console. This is because the data is only encrypted on the underlying storage device, not the storage construct that is presented (such as a block device/LUN or NFS file system).

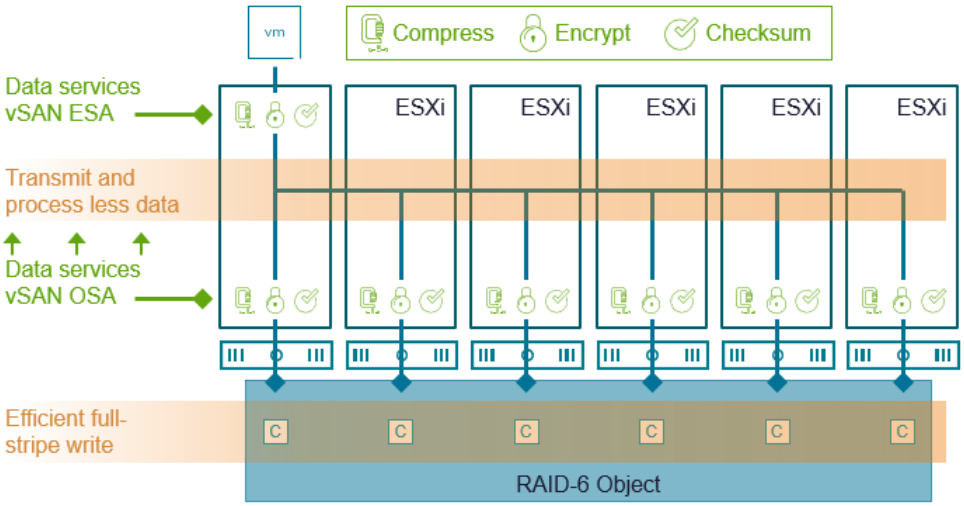
When using in-guest encryption solutions, or when using an alternative native VMware encryption solution like VM Encryption, the contents of the virtual machine are encrypted. Data is still secure in the device loss or theft scenario, in addition to protection from downloading a virtual machine to a USB or other portable media device from an administrative console.

While storage devices with "self-encrypting" capabilities exist on the Broadcom Compatibility Guide (BCG) for vSAN, it does not support the use of self-encrypting drives (SED) with that feature enabled. All encryption is required to be performed by vSphere/vSAN.

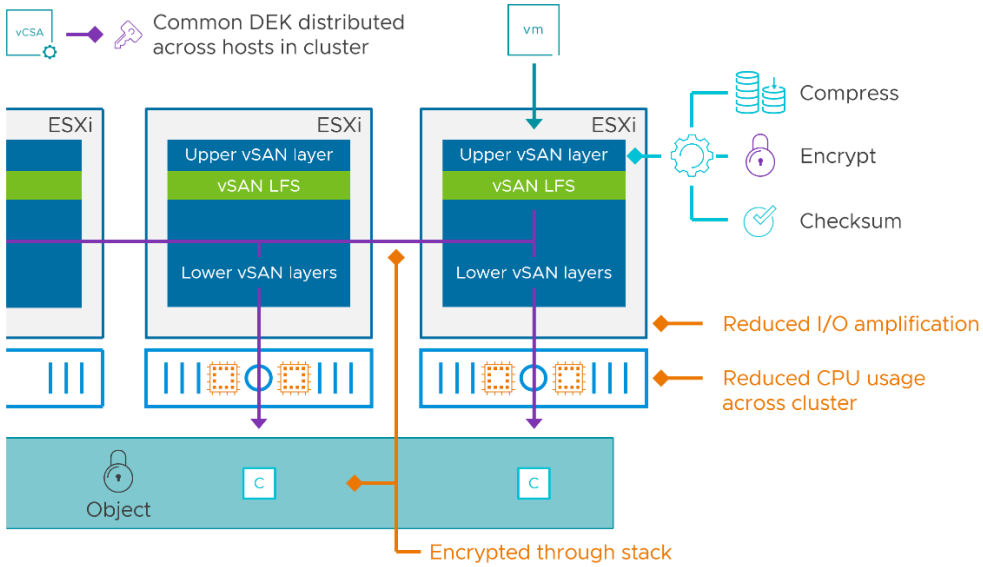
Encryption Services in vSAN ESA

The method vSAN uses to encrypt data-at-rest is extraordinarily efficient when compared to the implementation under the older, Original Storage Architecture. vSAN ESA's method to encrypt the data makes processing I/O more efficient, which minimizes overhead and improves performance. For more information on vSAN ESA, see the post: "An Introduction to the vSAN Express Storage Architecture." The release of vSAN in VCF 9.1 marks the 6th release of vSAN ESA, and as a result, this document will describe encryption services in vSAN ESA only.

In vSAN ESA, data encryption (and other services such as compression, and checksum processing) resides near the top of the storage stack. When a guest VM issues a write operation, it will encrypt this data shortly after it leaves the vSCSI layer and is ingested into the vSAN stack. Unlike vSAN OSA, this is performed once, and not only eliminates the need to encrypt the data on the other hosts holding the object, but also eliminates the decrypt re-encrypt processes found in the OSA. This reduces CPU and network resources across the cluster.



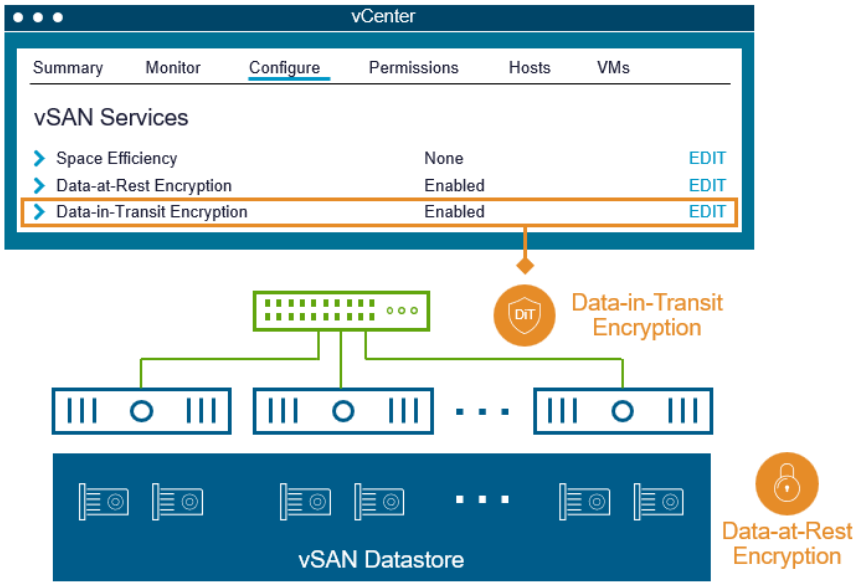
In vSAN ESA, encryption occurs in the upper layers of vSAN, as it receives incoming writes, but after compression occurs. This encryption step only occurs once, and since it occurs high in the stack, it means that all vSAN traffic transmitted in flight across hosts will also be encrypted.



Data-at-rest encryption is a cluster-based service. With vSAN in VCF 9.1, Data-at-Rest encryption can be used with vSAN Global Deduplication. This allows the ability to fully encrypt the data at rest while not interfering with the space efficiency capabilities of cluster-wide deduplication.

vSAN Data-at-Rest Encryption

vSAN's Data-at-Rest Encryption service provides encryption for all data objects as it resides on a vSAN datastore, without the need for any self-encrypting storage devices.



For vSAN ESA, Data-at-Rest Encryption is supported in both aggregated vSAN HCI clusters and disaggregated storage clusters.

Key Management using External KMS

Key management for vSAN Data-at-Rest encryption is achieved by using a Key Management Server (KMS). KMS solutions, which are either physical appliances or virtual machines, provide standards-compliant lifecycle management of encryption keys. Tasks such as key creation, activation, deactivation, and deletion of encryption keys are performed by Key Management Servers. The Key Management Interoperability Protocol (KMIP) can be used to communicate with a KMS by clients to use keys managed by the KMS. Note that the key provider chosen to be used with vSAN has no impact on performance. Key distribution is a control-plane task, and not a part of the data path.

The Domain of Trust, and Trust Establishment

Three elements comprise a vSAN Encryption domain of trust. The **key provider** (either an external KMS, or vSphere NKP), the **vCenter Server**, and the **vSAN hosts** in the cluster with encryption enabled.

vCenter Server and vSphere hosts can only use a key provider after establishing a trust with the key provider. Setting up the domain of trust follows the standard Public Key Infrastructure (PKI) based management of digital certificates. A digital certificate must be provided to the KMS from the vCenter Server environment. Different implementations of KMS allow for different types of certificates to be used to establish the trust. These are:

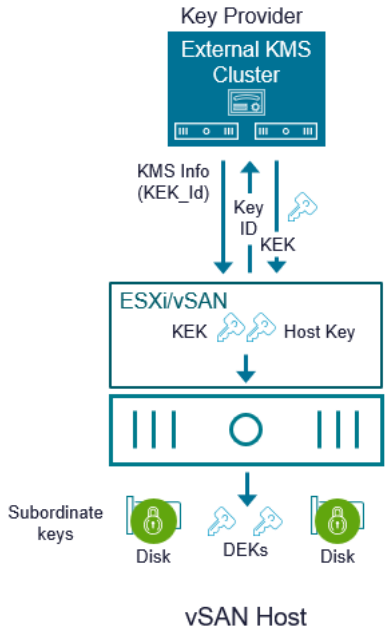
- **Root CA Certificate.** Trust is established for all certificates signed by the root certificate
- **Certificate.** The vCenter Server certificate is used to establish trust
- **New Certificate Signing Request.** vCenter Server generates a CSR, which is submitted to the KMS for signing. The generated certificate is then used to establish trust for the vCenter server.
- **Upload a certificate and private key.** vCenter Server is trusted after the KMS solution's certificate and private key are provided

Once the trust is established between the key provider and vCenter Server, a vSAN cluster may use vSAN Encryption Services. When vSAN Data-at-Rest Encryption is enabled on a cluster, the key provider connection information and is pushed to the vSAN hosts. The vSAN hosts then provide a reference key, or key ID, to the key provider. The key provider (an external KMS cluster or the VMware vSphere Native Key Provider, or NKP) then provides the Key Encryption Key (KEK) that is associated with the key id to the vSAN hosts. Disk Encryption Keys (DEKs) are wrapped by the KEK. vSAN uses a common DEK across the entire cluster. It is subordinate keys such as the Object Data Encryption Key (ODEK) Object

Metadata Encryption Key (OMEK) that apply discrete encryption keys to the specific objects, respectively. vSAN ESA also uses discrete Disk Metadata Disk Encryption Keys (DMEK) for each storage device, and is responsible for low level (LSOM) metadata

Since it occurs high in the vSAN stack (in comparison to the original storage architecture), the disk encryption key (DEK) is used across the cluster, instead of the discrete disks. **This allows each host in the cluster to decrypt the objects owned by other hosts. It is the other key types noted above that are responsible for discrete encryption.**

- vCenter Server will send the wrapped DEK to each host in the cluster, where the vSAN management daemon on each host will unwrap it by the KEK.
- vSAN Management daemon on each hosts will notify other parts of the stack (LFS, CLOM, etc.) via CMMDS which subordinate encryption keys are used.
- The unwrapped subordinate encryption keys will be inserted in the host key cache where our vSAN LFS can look it up.



Key Management tasks as they relate to vSAN Encryption

Keys are not automatically created for clients even though trusted communication has been established. vSphere administrators do not have direct access to the lifecycle of encryption keys, but actions performed through the vSAN UI impact the key lifecycle process.

- **Turn on Encryption.** Unique KEK is created for each vSAN cluster. DEKs are created when claiming vSAN cache and capacity devices. This is a rolling process.
- **Turn off vSAN Encryption.** DEKs and the KEK are removed from the vSAN cluster in a rolling process.
- **Shallow ReKey.** An existing KEK is recreated for the cluster a shallow rekey is being performed on.
- **Deep ReKey.** The existing KEK and DEKs are recreated. Like enabling or disabling vSAN Encryption, this is a rolling process.

Access to these capabilities can be restricted from vSphere Administrators by assigning user accounts to the **No cryptography administrator** role.

External KMS Availability

When designing any environment, services such as Domain Naming Service (DNS) or Network Time Protocol (NTP) are typically highly available. Just as DNS is critical for name resolution and NTP is critical for time synchronization, key provider services are critical to the availability of encrypted data. Some Key Management Server vendors provide highly available configuration capabilities, often in the form of configuring multiple KMS Servers into a KMS Cluster. Choosing a key provider solution that provides a resilient and available key provider infrastructure is an important part of the vSAN Encryption design.

An alternative or additional level of resilience can be achieved through key persistence on the hosts through a TPM device installed on each host in the cluster. More information on TPMs can be found in the Key Persistence section of this document.

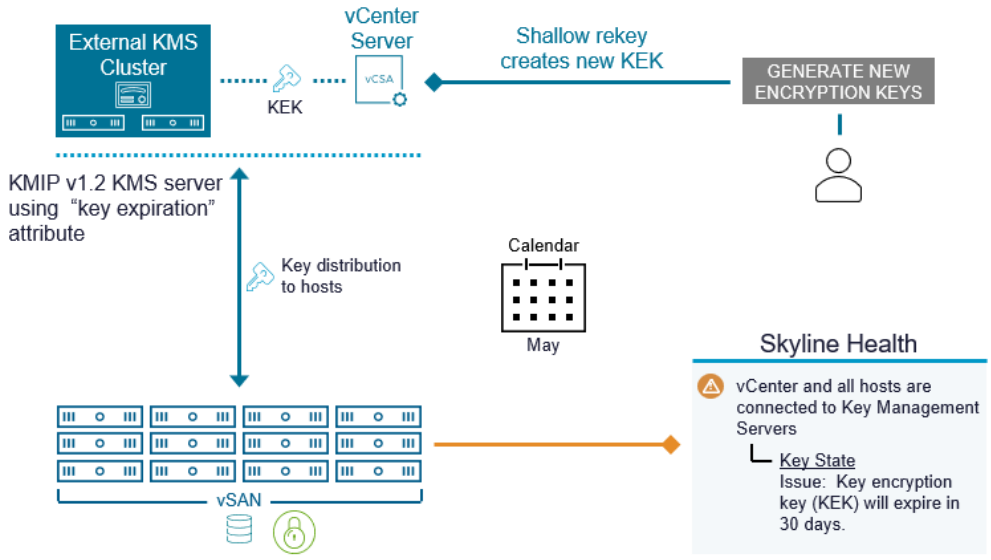
Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) on each host and provide the key to the host when the key provider is inaccessible.

External KMS Compatibility

While there are different KMIP protocol versions available today, VMware vSAN, and VM Encryption, support the KMS 1.1 protocol. Any KMIP 1.1 compatible Key Management Server solution that provides KMIP 1.1 is supported with vSAN and VM Encryption.

vSAN supports the use of TLS. To learn more on how to use a specific version of TLS, see the link: "Enable or Disable TLS Versions on ESXi hosts."

Some KMS solutions provide a scheduled "key expiration" capability to provide enhanced levels of key rotation security. This gives the ability for an administrator of a KMS to set a defined date that a KEK will expire at a given time. Prior to vSAN 8 U2, vSAN was unaware of this "key expiration" attribute provided by the KMS, and one could inadvertently have a disabled cluster because of the expired KEK. vSAN 8 U2 (ESA and OSA) and later provides visibility and awareness of the key expiration attribute and is integrated with vSAN Health. If the KMS uses this feature, a triggered health finding will provide the number of days remaining on the valid key, and a convenient way of performing a shallow rekey.



Infrastructure Placement of External KMS

If using one or more KMS virtual appliances, they should not be deployed on an encrypted datastore if the hosts do not have TPMs to persist the keys, and the appropriate configuration changes have been made. This is because placing a KMS appliance/cluster on top of the datastore it is providing keys for, creates a circular dependency. Consider the following scenario where A KMS appliance or cluster resides on the encrypted cluster it is providing keys for, with hosts that do not have TPMs for persistent key storage.

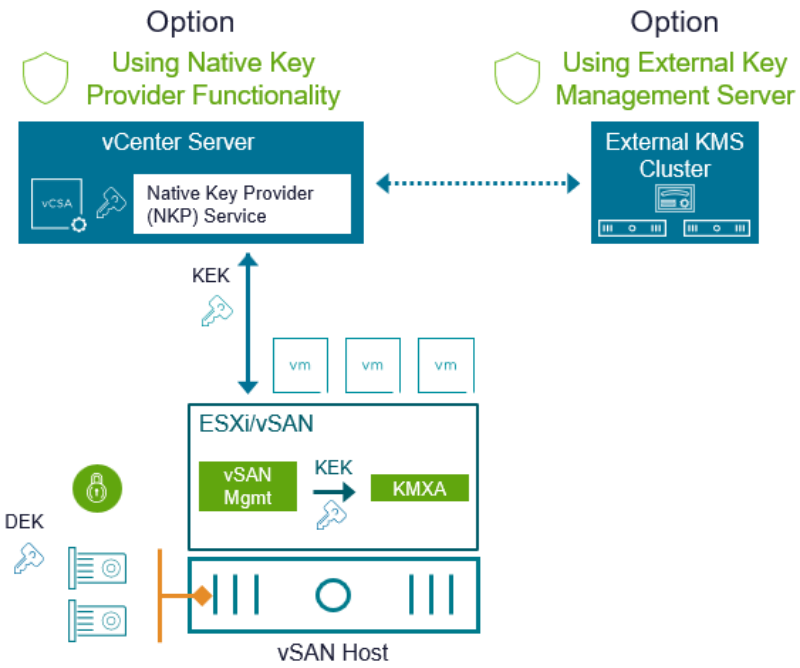
- If a single KMS appliance resides on the cluster, if the host it is running on fails/reboots, when the host comes online, it cannot mount the encrypted disks until the KMS returns to service, because the KMS is not available.
- If a KMS cluster resides on an encrypted cluster and all hosts suffer a power loss, when they are powered on, they will not be able to mount their disks, because the KMS cluster is not available.

Assuming the hosts in the examples above **are not** using TPMs, in both cases, the KMS appliance/cluster will not be available because its storage is not available.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) on each host and provide the key to the host when the key provider is inaccessible.

Key Management using Native Key Provider (NKP)

VCF includes the vSphere Native Key Provider (NKP), a built-in key provider that can be used with vSAN Data-at-Rest Encryption. When enabled, the NKP is provided by vCenter server, and enables many of the same key provider functionalities as a traditional Key Management Service (KMS). The NKP has proven to be an extremely useful and easy solution that gives customers of all sizes better access to encryption technologies. Additional information can be found on the homepage of the native key provider.



When using the vSphere NKP, many of the principles of key management are similar to using an external KMS solution. Additional configuration and operational guidance for using the vSphere NKP with vSAN is provided later in this document.

Improving Availability using Key Persistence

Key persistence allows for hosts to restart successfully even if the key provider is not available, which will improve the robustness of your environment during failure conditions. The capability of key persistence will depend on whether an external KMS or the vSphere Native Key Provider (NKP) is used, and whether Trusted Platform Modules (TPM) are installed in the vSphere hosts.

Key persistence when using external KMS solutions

As discussed earlier in this document, external Key Management Servers (KMS) typically provide durability of key distribution through KMS clusters (more than one KMS server). These environments may also benefit from key persistence, where

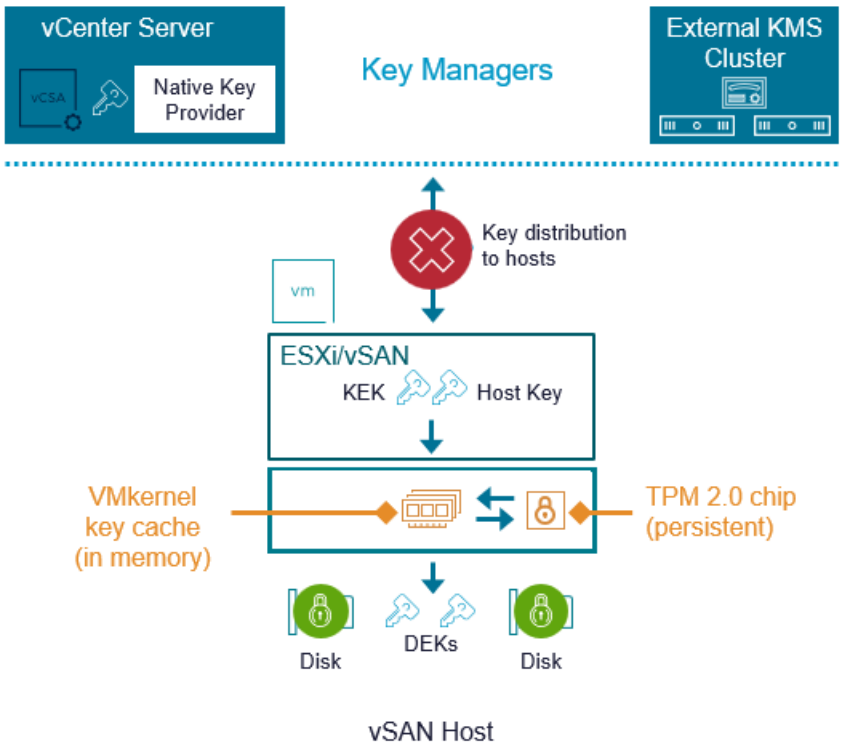
connectivity to a KMS is poor, or the KMS is not redundant. Activating key persistence in these configurations require additional steps. Doing so does introduce a tradeoff in resilience/durability versus security.

Key persistence when using the vSphere NKP

One significant advantage of using the vSphere NKP is the ability for hosts using the NKP to persist keys. How and where keys persist on the host depends on if the host has a TPM which cryptographically stores the keys on a special device on each host in the cluster.

- **Using NKP without TPMs on each host in the cluster.** This approach will store the Key Derivation Key (KDK) on the storage device used for the host configuration, or the boot bank of vSphere. It is encrypted but relies on the hypervisor to manage and decrypt the configuration.
- **Using NKP with TPMs on each host in the cluster.** This approach will cryptographically store the keys on each vSAN host to ensure availability of keys if the key provider is inaccessible. It requires the TPM to release the key after performing an integrity check and is much more secure under conditions of physical theft. This capability will work automatically for environments using the vSphere NKP.

The **addition of TPM chips on any new vSphere host should be a part of an organization's purchasing practices.** It is a simple and affordable way to make the distribution of encryption keys to physical hosts more robust.



vSAN Host

When using TPM 2.0 chips on all vSAN hosts in a cluster, any key issued that that is stored in the key cache, will also be persisted to the TPM chip immediately. **Upon rebooting the host, this key persistence feature would restore the key from the TPM chip to the key cache.** If the KEK, or the host key are not within the key cache for some reason, and cannot be fetched from the TPM, then they will be retrieved from the KMS. Keys will always be attempted to be fetched from the key cache first. Even upon host restart, the TPM will restore the keys immediately. If there are no keys locally, then they will be retrieved from the KMS.

Key persistence is enabled by default when using the vSphere NKP, but **when using an external KMS solution, will require enabling it through the following esxcli commands**.

```
esxcli system settings encryption set --mode=TPM
```

esxcli system security keypersistence enable

Note that many operational activities related to key management do not change with the use of cached keys on TPM chips in the vSAN hosts. Examples of operational activities that are unchanged from this enhancement include:

- Turn on/off encryption (stale/old KEK and old DEK automatically removed from persistent storage on TPM)
- Shallow rekey (stale/old KEK and old host key automatically removed from persistent storage on TPM)
- Deep rekey (stale/old KEK, old DEK, and old host key automatically removed from persistent storage on TPM)
- I/O with data-at-rest encryption.
- Remove disk (stale/old DEK automatically removed from persistent storage on TPM

Running a vCenter Server with the NKP enabled on the vSAN cluster it is encrypting is an acceptable practice so long as each host in the cluster uses TPMs installed and in use. This is the approach most customers use to ensure there is not a circular dependency.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) on each host and provide the key to the host when the key provider is inaccessible.

Infrastructure Placement of vSphere NKP

The placement of a vSphere NKP in a vSAN powered environment is much more flexible than when using an external KMS solution. vSAN clusters have always fully supported the ability to run vCenter server on top of the cluster it manages. **vSAN clusters encrypted with vSAN Data-at-Rest Encryption also support the ability to run vCenter server on top of the cluster it manages, even when the vCenter server is providing NKP services.** This type of configuration is supported because the NKP stores the Key Derivation Key (KDK) locally, and does not need to talk to the key provider to unlock the storage devices upon a host reboot. This makes the use of an NKP flexible, and robust. While it is possible to do so without the use of TPMs installed on each physical host in the cluster, **we highly recommend the use of TPMs to make the topology more secure and durable.**

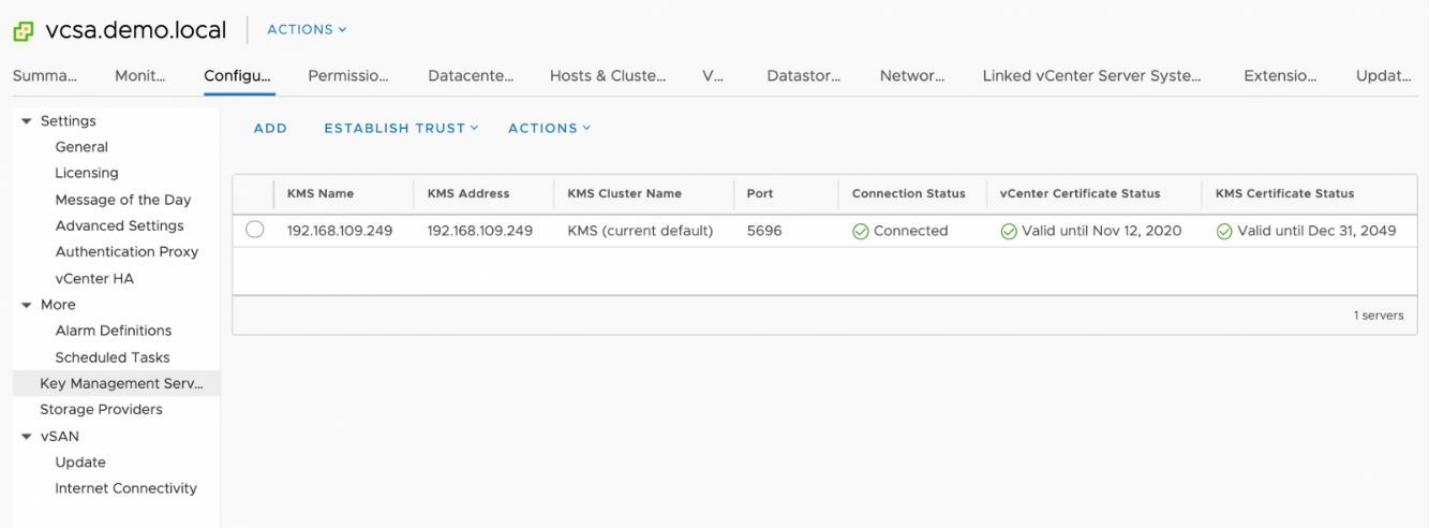
Recommendation: Always backup the NKP keys to an external location along with performing routine backups of the vCenter Server, and understand the respective recovery procedures. This will make any scenario of recovery of a vCenter Server that is providing NKP services to the cluster it resides on much easier.

vCenter Server

An environment's vCenter Server is the primary management application for vSphere. It is a familiar platform that acts as a control plane for the configuration and management of the different parts of vSphere and additional VMware solutions. vCenter provides a rich set of APIs used by other VMware solutions, 3rd party solutions, and often custom code. VMware solutions use these APIs to provide a uniform and consistent framework for better interoperability and management.

External KMS Configuration

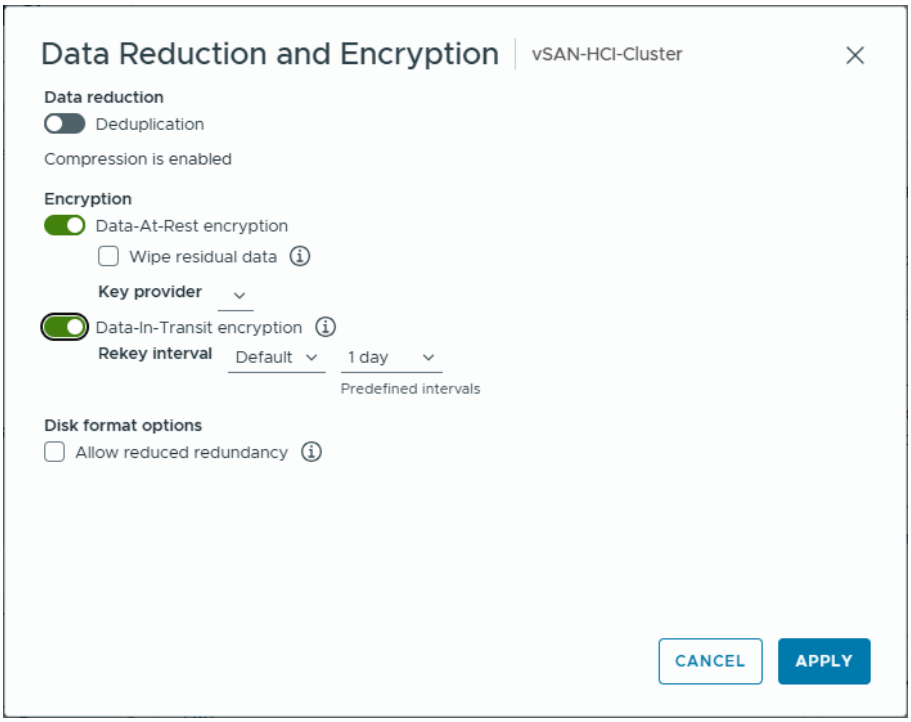
vCenter Server provides a central location for KMS configuration that is available to be used by either vSAN Encryption or VM Encryption. It is also the primary interface when deploying and using the vSphere NKP.



Certificates used to establish the trust with the KMS are persisted into the VMware Endpoint Certificate Store (VECS). These certificates are shared by both vSAN Encryption and VM Encryption. To ensure proper trust between the hosts and the KMS, certificates and the KEK_ID are pushed to vSphere hosts for vSAN Encryption. Using the KEK_ID and KMS configuration, hosts can directly communicate with the KMS cluster without the dependency of vCenter being available.

vSAN Encryption Configuration

vSAN is configured on a per-cluster basis, and vSAN Encryption is a configuration option of a vSAN cluster. If a key provider (external KMS or NKP) has been configured, vSAN Encryption is easily enabled through the cluster management UI in the vSphere Client.



vSAN encryption services for vSAN in VCF 9.1

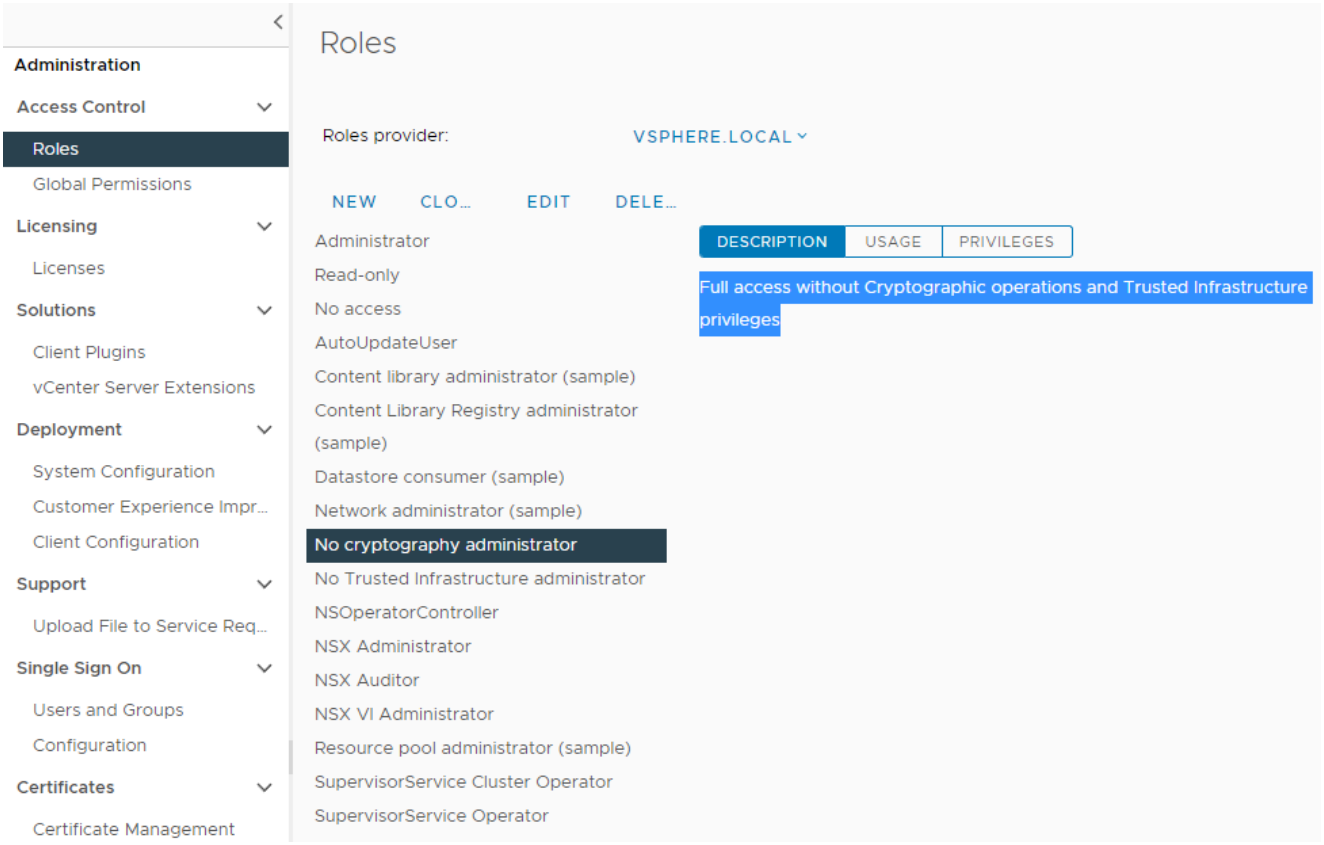
- **Wipe residual data.** Useful for disks that already have data on them. This wipes any data from the disk before encryption occurs.

- **Allow Reduced Redundancy**. vSAN will reduce the protection level when the service is turned off or on. This is typically only used when a vSAN cluster is at the maximum number of hosts or fault domains required to meet a protection policy.

Securing Administrator Access to Cryptographic Operations

Access to the cryptographic properties and actions of a vSAN cluster, like encrypted virtual machine properties/actions for VM Encryption, are limited to users or groups assigned with the Administrators role. It is often necessary to create custom roles or provide "Administrator-like" access to vCenter, vSphere hosts, and the vSAN cluster itself.

Users assigned to this role may not perform Cryptographic operations in vSphere and vSAN environments.



Operations not associated with cryptographic tasks can be performed by users assigned to this role. The summary below describes the basic capabilities of this role.

Operations not Allowed	Operations Allowed
Manage KMS	Add a host to an encrypted vSAN cluster.
Manage encryption policies	Move VM from a non-encrypted datastore to encrypted datastore
Manage keys	Move VM from a non-encrypted vSAN cluster/datastore to an encrypted vSAN cluster/datastore
Register a VM	
Register a host	

vSAN Hosts

The following described noteworthy considerations about the hosts in a vSAN cluster when vSAN Data-at-Rest Encryption is enabled.

Configuration

When vSAN Encryption is enabled, several items are configured/pushed to the vSphere host. Items such as the KMS Cluster information, the Key Encryption Key ID (KEK_ID), and a host key that is unique for the cluster.

KMS information that is pushed to each host in the vSAN cluster

- Cluster ID/name of the KMS cluster
- KMS Port - This is typically 5696
- Address of the cluster
- Proxy Address/Port if used
- A host key used core dumps.

vSAN Encryption specific information that is pushed to/created for the cluster

- The Key Encryption Key - Used to retrieve the KEK from the KMS on boot up so storage devices claimed by vSAN can be mounted so data may be read from or be written to them.
- vSAN Host Key ID - Key used to encrypt/decrypt host core dumps.

Depending on the state of vSAN encryption, some other values are set

- Whether the host is going through the process of enabling or disabling encryption
- Whether the current KMS Server is the current, or previous KMS Cluster during a KMS Cluster changing process

Host behavior at boot up

When vSAN Encryption is enabled, to participate in data operations requiring data encryption/decryption, a host must have access to the KEK. Hosts connect directly to the key provider over the Management VMkernel interface to securely retrieve the KEK using the KEK_ID (which is pushed by vCenter Server when enabling vSAN encryption). If the hosts do not use TPMs, the KEK is not persistently stored, but rather stored in secure location in host memory in the key cache kernel module. This kernel module caches keys for allowed processes and is used by both vSAN Encryption and VM Encryption.

Because the KEK is not persistent, each time a host boots, it must use the KEK_ID and KMS settings to connect to the KMS Cluster and retrieve the KEK. The KEK is then placed in the key cache kernel module for use by vSAN Encryption. With KEK_ID and KMS settings being persistently stored on each host, there is no requirement to communicate with vCenter server to retrieve the KEK. This is advantageous in the situation where vCenter Server may be offline from a failure, reboot, or network isolation.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) on each host and provide the key to the host when the key provider is inaccessible.

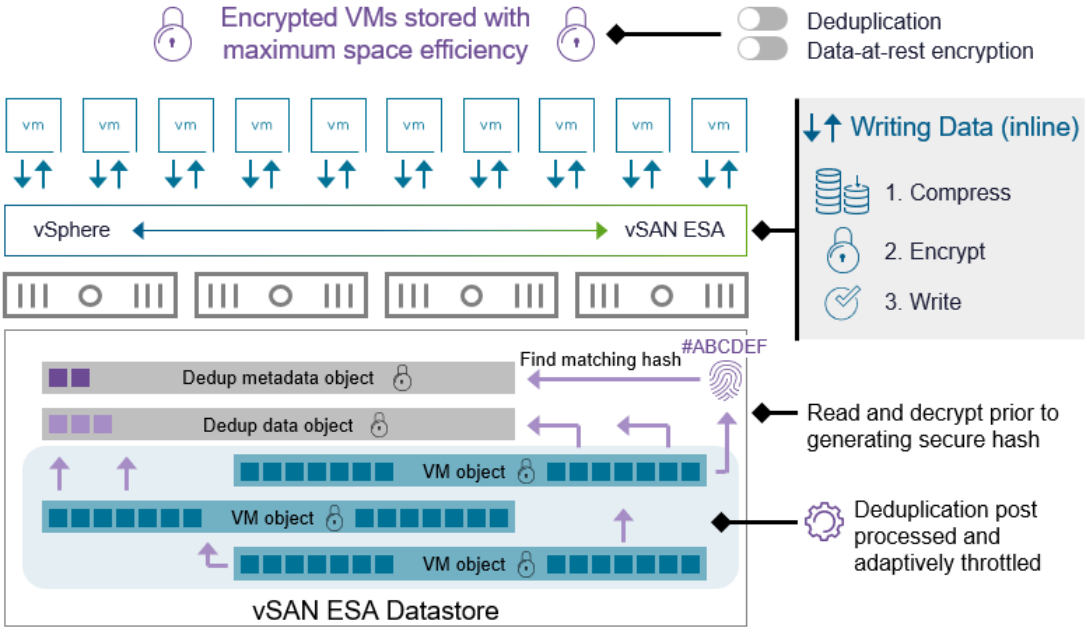
If a running host is added to an encrypted vSAN cluster, it will not immediately have access to the existing vSAN datastore. A KEK request must be performed. This can occur in the cases of a reboot of the newly added host.

Note that for stretched cluster and 2-node topologies, the vSAN witness host appliance is not encrypted. Why? **It is more secure to not encrypt the witness node.** If the witness node is encrypted, it must store all the credentials to get the secret key from the Key Management Server (KMS). These credentials become another attack surface that we have to protect. However, since witness runs in a virtual environment, it is easier to be attacked than regular hosts which run in physical environments. Not encrypting witness node reduced attack surface and makes the system more secure. What can be leaked on the witness node includes number and size of each vSAN object, their log sequence number, and policy. None of these are sensitive user data.

When using vSAN Data-at-Rest Encryption, it will only encrypt the data at rest. To encrypt the data in flight, enable the vSAN Data-in-Transit service on the cluster.

Write Operations

In vSAN for VCF 9.1, Data-at-Rest encryption occurs at a slightly different location than previous versions of ESA. The initial encryption process begins at the DOM owner instead of the DOM client. This allows for better decoupling of encryption responsibilities whether it be aggregated vSAN HCI clusters, or disaggregated vSAN storage clusters. This also sets the stage for fast and efficient global deduplication post-processing activities in vSAN.



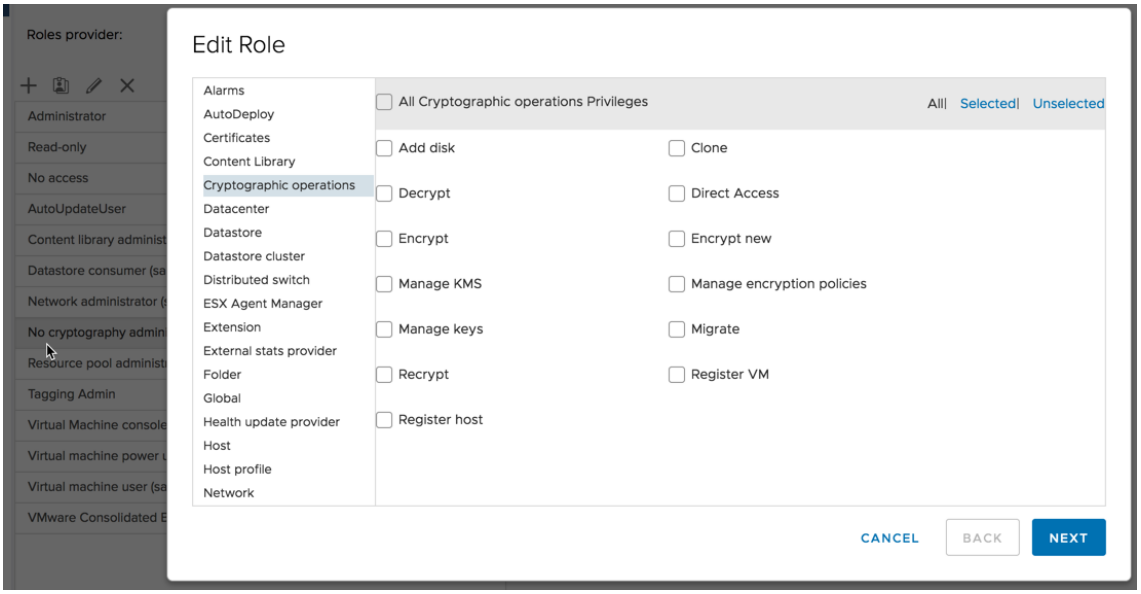
Encryption occurs as data is written to disk. When deduplication post processing occurs at a later time (usually a time when the cluster is less active) each block scanned will be decrypted prior to generating a hash that is used to compare against any other potential duplicates. This process occurs in memory, and allows vSAN to deduplicate encrypted data without any compromise in space efficiency

Role Based Access Control

Securing workloads does not end with the use of encryption technologies. Access granted to data and its management must also be properly secured. Effective access to these workloads must align with the responsibilities associated with their management, configuration, reporting, and user requirements.

No Cryptography Administrator Role

The "No Cryptography" role is very similar to the normal administrator with many of the same privileges. Operations such as power on or off a virtual machine, boot, shutdown, vMotion, as well as normal vSAN management may be performed. However, this role is not allowed to perform any cryptographic operations.



The permissions in the illustration show that users assigned to the No Cryptography Administrator role do not have any permissions to perform any operations that require any cryptographic operations.

No Cryptography Administrator and VM Encryption

Users assigned to the No Cryptography Administrator role are **not granted** the following privileges:

- Ability to encrypt or decrypt virtual machines with VM Encryption
- Direct console access to virtual machines that are encrypted with VM Encryption
- The ability to download virtual machines that are encrypted with VM Encryption. This will prevent the user from downloading a virtual machine to a USB or other offline media.
- The ability to add hosts to vCenter. This limitation exists, because the process of adding a host to vCenter grants the host access to the cryptographic keystore.

No Cryptography Administrator and vSAN Encryption

Users assigned to the No Cryptography Administrator role are **not granted** the following privileges:

- The ability to enable or turn off vSAN Encryption
- The ability to generate new encryption keys (Shallow or Deep Rekey)
- The ability to add hosts to vCenter.

Users assigned to the No Cryptography Administrator role are granted the following privileges:

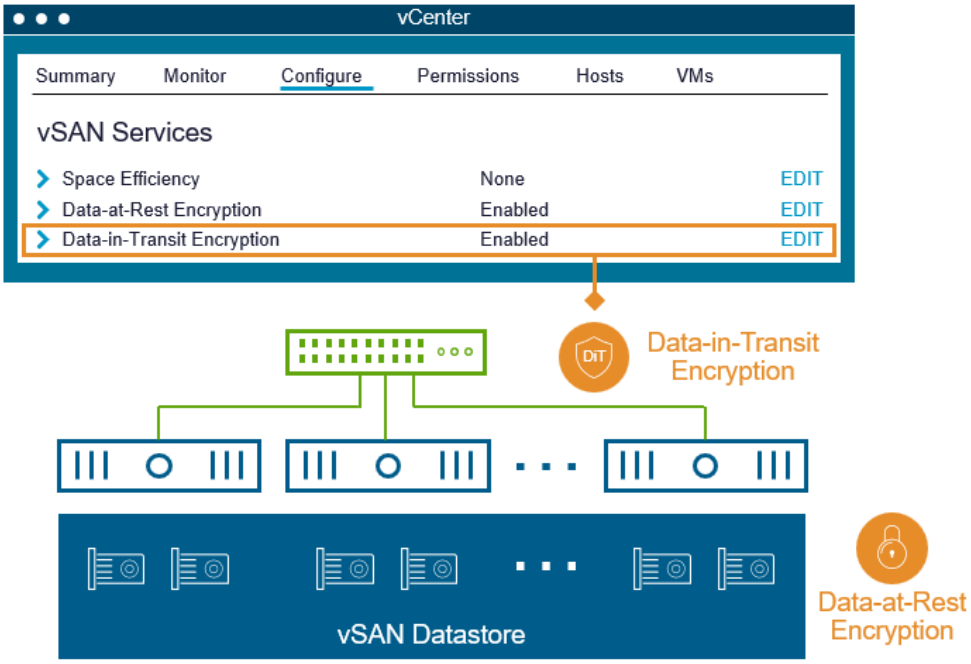
- Direct console access to virtual machines that reside on a vSAN Cluster with vSAN Encryption enabled
- The ability to download virtual machines that reside on a vSAN Cluster with vSAN Encryption enabled.
- The ability to add hosts to a vSAN Cluster*.

* In a situation where a host needs to be added to a vSAN Cluster, a user with Cryptographic rights would have to add the host to vCenter. Once added to vCenter a Non-Cryptographic Administrator could then add the host to an encrypted vSAN Cluster.

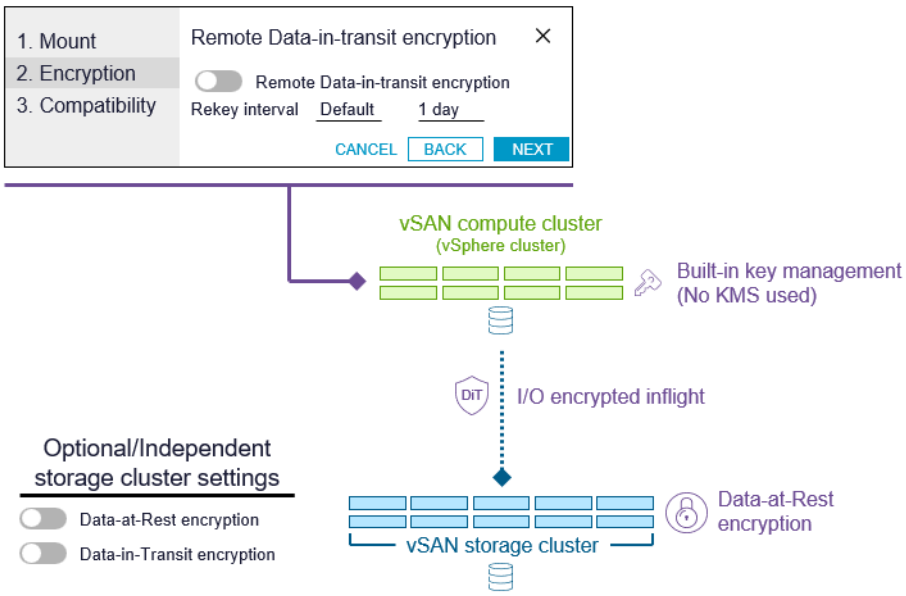
vSAN Data-in-Transit Encryption

Complementing vSAN Data-at-rest encryption, vSAN also provides a cluster-based feature for encrypting vSAN data in transit. The vSAN "Data-in-Transit Encryption" securely encrypts all vSAN traffic in transit across hosts. It uses the same

FIPS 140-3 validated Cryptographic modules as used with Data-at-rest encryption, as well as vSphere's VM Encrypt, and does so in an automated manner that does not require a KMS server for key management. The Data-in-Transit Encryption option can be enabled independently from the vSAN data-at-rest encryption, but enabling both will provide a complete, end-to-end encryption solution on a per-cluster basis.



vSAN in VCF 9.1 introduces support for vSAN Data-in-Transit encryption for both aggregated vSAN HCI clusters and disaggregated vSAN storage clusters. This allows for end-to-end encryption for all storage-related vSAN traffic.



If you do have a requirement to encrypt all storage traffic, and you are using a disaggregated deployment model using vSAN storage clusters, then you will want to enable vSAN Data-in-Transit encryption on the vSphere clusters mounting the remote datastore (available in vSAN for VCF 9.1 and later). This ensures that the storage traffic between the client cluster and the server cluster is encrypted.

Unlike traditional three-tier architectures, where the storage fabric may be physically isolated from all other types of traffic, vSAN may use the same uplinks to serve multiple needs. vSAN Data-in-Transit Encryption addresses this architectural difference and provides a capability that is typically not found in other architectures: over-the-wire encryption.

Since there are no management tasks related to key management for vSAN Data-in-Transit Encryption, most of the guidance in this document is focused on vSAN Data-at-Rest Encryption.

When using Data-at-Rest encryption in vSAN ESA, it does imply that the vSAN data will be encrypted in-transit (across the wire) since it occurs high in the storage stack. While this is true, it may not meet some strict security requirements that state each packet transmitted must have its own unique hash. To ensure the highest levels of security, Data-in-Transit encryption remains as an available toggle in the cluster data services section. If this is enabled with Data-at-Rest Encryption in the ESA, it will encrypt each network packet uniquely so that so that identical data is not transmitted over the network. For more information, see the blog post: "Cluster Level Encryption with the vSAN Express Storage Architecture." Some may find that using Data-at-Rest Encryption is sufficient enough to meet their "in-transit" encryption requirements. But if not, use vSAN Data-in-Transit encryption as well.

Data-in-Transit encryption is supported for disaggregated environments using vSAN storage clusters in VCF 9.1. For customers interested in providing end-to-end encrypted storage using vSAN storage clusters, it could conceivably be configured in one of two ways.

- **Option 1: Enable Data-at-Rest encryption & Data-in-Transit encryption on the vSAN storage cluster, and enable Data-in-Transit encryption on the client cluster mounting the datastore.** This will offer end-to-end encryption, and guarantees every packet transmitted within the storage network has its own unique hash. This may have some impact on storage performance for the backing storage cluster.
- **Option 2: Enable only Data-at-Rest encryption on the vSAN storage cluster, and enable Data-in-Transit encryption only on the client cluster mounting the datastore.** This will offer end-to-end encryption. When ESA encrypts data for Data-at-Rest encryption, it performs encryption high in the stack (DOM), but it will not necessarily use a unique hash for identically replicated traffic (a mirrored write for example). The data transmitted within the storage cluster would in fact be encrypted, just not at the level to guarantee unique hashes across the network for mirrored writes. Unless a customer is under the strictest regulatory requirements, this configuration may be good enough, and would eliminate the potential performance penalty on the east-west traffic within the storage cluster.

Key Management

All key management tasks are handled automatically by the vSAN hosts participating in the cluster, which means that there is no need for a key provider (external KMS or vSphere NKP) if vSAN Data-in-Transit is enabled. If both encryption services are enabled, Data-at-Rest Encryption will use its own built-in key provider (e.g. external KMS or vSphere NKP), while the Data-in-Transit encryption will continue to manage the keys for Data-in-Transit encryption automatically.

When Data-in-Transit Encryption is enabled, all hosts that join a vSAN cluster are authenticated with dynamically generated symmetric keys. Upon the removal of a host from a cluster, any existing authentication is removed.

vSAN Encryption Services Operations

The following will detail many of the common operations related to vSAN clusters that use vSAN encryption services. Other operational recommendations for vSAN Encryption Services can be found in the "Data Services" section of the vSAN Operations Guide.

Enable vSAN Encryption

When using the vSAN ESA, enabling vSAN encryption services is as easy as clicking on one toggle in vCenter Server. Before attempting to enable vSAN Encryption, a few items need to be considered.

- **Is AES-NI supported and enabled in each the bios of each host in the vSAN cluster?** The encryption process takes advantage of Advanced Encryption Standard New Instructions (AES-NI) in many of today's CPUs. These

additional instruction sets supported by AES-NI enabled processors, perform much of the work of the encryption and decryption, removing the need to perform these tasks in software alone. Some server configurations have AES-NI enabled by default, while others do not. Consult the system manufacturer's documentation to determine whether AES-NI is supported and how to verify it is enabled.

- **Will Reduced Redundancy be required?** Because the process of enabling or disabling encryption requires a disk format change, there must be enough nodes or fault domains for the data being moved off of a storage device to reside elsewhere in the vSAN cluster. 3-Node vSAN configurations will require Reduced Redundancy because there is no additional eligible host for the data to reside. This may not be the case in a 3-fault domain configuration, depending on how many nodes are in each fault domain. 2-Node vSAN configurations are treated identically to 3 node vSAN configurations when performing operations requiring a disk format change. Reduced Redundancy will be required in 2 Node vSAN configurations as well. Configurations that have a sufficient number of fault domains may also require Reduced Redundancy in situations where there is not enough free capacity to migrate data being evacuated from a storage device.

Enabling or disabling encryption introduces a rolling reformat of the storage devices that are claimed by vSAN. This rolling reformat does not place the host into maintenance mode and simply reformats one storage device at a time in vSAN ESA. **While the hosts are not placed into maintenance mode during the enabling or disabling of vSAN Data-at-Rest encryption, this rolling reformat may generate a substantial amount of data movement across the network during the transition.** Depending on the configuration, capacity usage, hardware, and if it is ESA vs OSA, there may be some performance impacts to workloads. This is why **we encourage setting the encryption services as desired prior to placing the cluster into production.** vSAN ESA will typically show less impact during this transition than OSA.

In initial versions of vSAN ESA, encryption must be enabled at the time of the cluster build up. With more recent versions of vSAN, encryption on ESA clusters can be enabled after the cluster is initially deployed.

If there are any availability issues with vSAN hosts during the time of enabling or disabling encryption services, the failure handling will work as described in the vSAN Availability Technologies document.

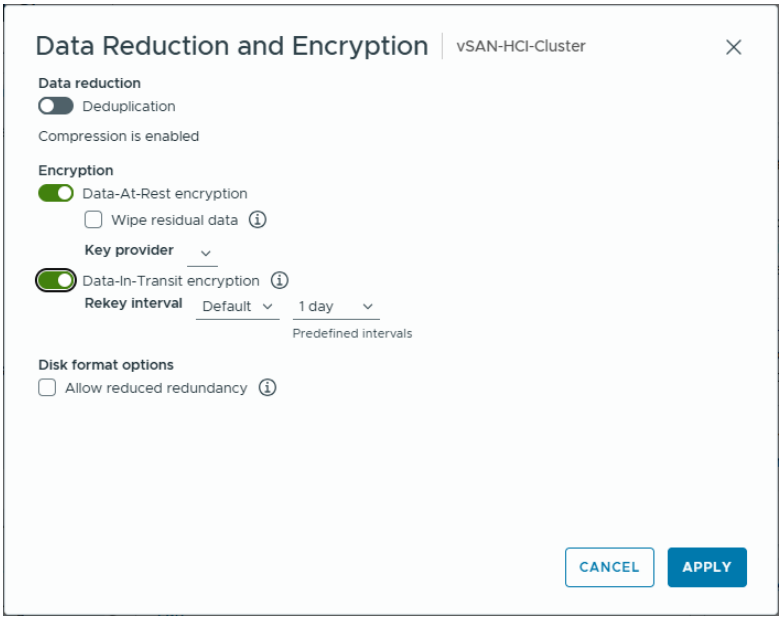
Permissions required to enable vSAN Encryption

To enable vSAN encryption, users or groups first must have the proper access. A user or group must have the following permissions:

- Host > Inventory > Modify Cluster
- Cryptographic Operations > Manage encryption policies
- Cryptographic Operations > Manage KMS
- Cryptographic Operations > Manage keys

Enabling vSAN Encryption

Select **Cluster > Configure > Services > Edit** to enable vSAN Encryption



If the user or group does not have the proper permissions, the Encryption option will not be presented. Referring to the suggestions previously covered, consider the options:

- Deduplication and Compression
  - If there is no long-term desire to change the state, then do not change this setting.
  - If planning to change this setting in the near term, it may be better to make this change at the same time as changing encryption.
- Encryption - Select to enable
  - This will force a disk format change from the current state.
  - KMS connectivity and cluster key ID information is pushed to the vSAN hosts to enable the ability to communicate directly with the KMS server.
  - A host key is created for the cluster
- Wipe residual data (formerly Erase disks before use)
  - This will wipe any existing data from a disk as the encryption process occurs
  - When wiping any existing data from a disk the encryption enablement process has a significantly longer duration
  - Any time additional disks are added to a host, the devices will be wiped before being added to the host's storage pool
- Allow Reduced Redundancy - Will be required if the conditions previously covered regarding 2 Node, 3 Node, or there is minimal available capacity preventing an evacuation.

Select **Ok**.

The vSAN Encryption process will begin.

Turning off vSAN Encryption

When using the vSAN ESA, turning off vSAN encryption services is as easy as clicking on one toggle in vCenter Server. Before attempting to turn off vSAN Encryption, a few items need to be considered.

- **Will Reduced Redundancy be required?** Because the process of enabling or disabling encryption requires a disk format change, there must be enough nodes or fault domains for the data being moved off of a disk to reside elsewhere in the vSAN cluster. 3-Node vSAN configurations will require Reduced Redundancy because there is no additional node for the data on disks being evacuated to move to. This may not be the case in a 3 fault domain

configuration, depending on how many nodes are in each fault domain. 2-Node vSAN configurations are treated identically to 3 node vSAN configurations when performing operations requiring an disk format change. Reduced Redundancy will be required in 2 Node vSAN configurations as well. Configurations that have enough fault domains may also require Reduced Redundancy in situations where there is not enough free capacity to migrate data being evacuated from a disk.

Permissions required to change the status of vSAN Encryption Services

To change the status of vSAN encryption services, user or group first must have the proper access. A user or group must have the following permissions:

- Host > Inventory > Modify Cluster
- Cryptographic Operations > Manage encryption policies
- Cryptographic Operations > Manage KMS
- Cryptographic Operations > Manage keys

If the user or group does not have the proper permissions, the Encryption option will not be presented. Referring to the suggestions previously covered, consider the options:

- Deduplication and Compression
  - If there is no long-term desire to change the state, then do not change this setting.
  - If planning to change this setting in the near term, it may be better to make this change at the same time as changing encryption.
- Encryption - Deselect to turn off
  - This will force a disk format change from the current state.
  - Encryption will be turned off for the cluster
- Allow Reduced Redundancy - Will be required if the conditions previously covered regarding 2 Node, 3 Node, or there is minimal available capacity preventing a disk evacuation.

Select **Ok**.

The process of turning off vSAN Encryption will begin.

External KMS Operations

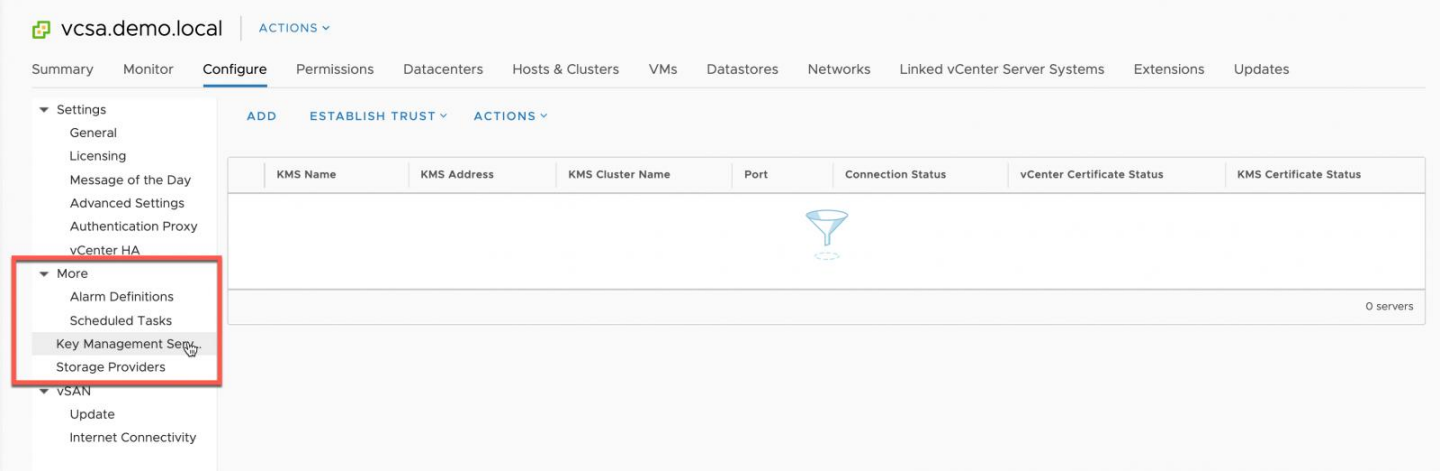
When using vSAN Data-at-Rest Encryption, a KMS must be used to manage the keys used to encrypt and decrypt the data. The KMS can either be an external third party KMS solution, or the vSphere Native Key Provider (NKP). vSAN Data-in-Transit encryption will always manage its own keys, and thus does not need or use a key provider, even if it is used in combination with the Data-in-Transit Encryption service.

The process of configuring a KMS server is relatively simple. The process can be broken down to these steps:

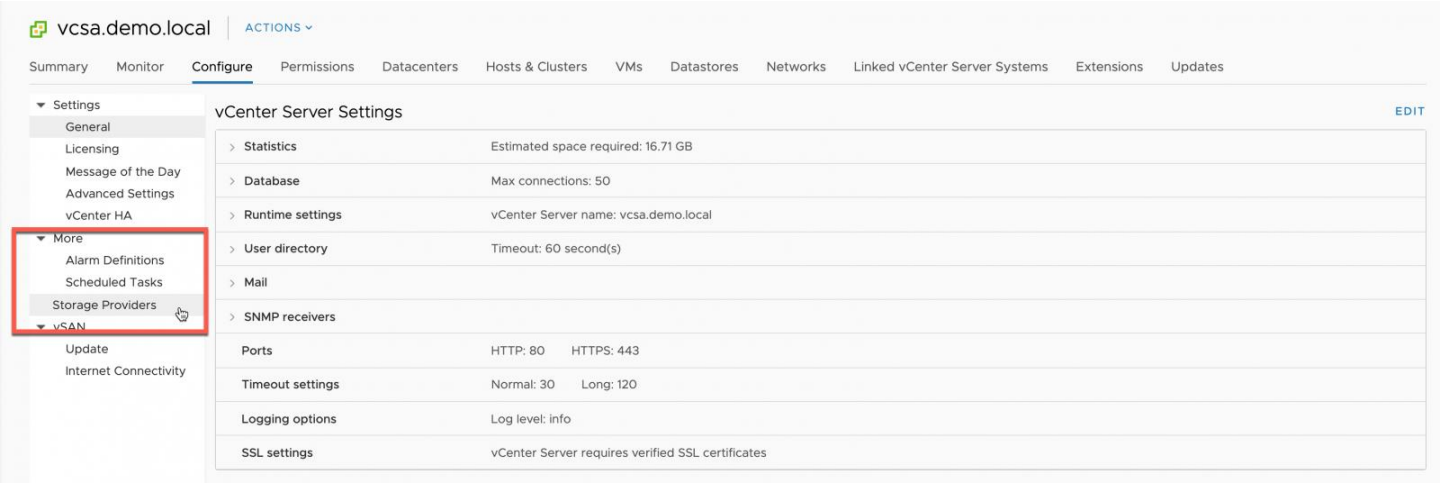
- Add the KMS
- Have the KMS trust vCenter Server.

Adding the KMS

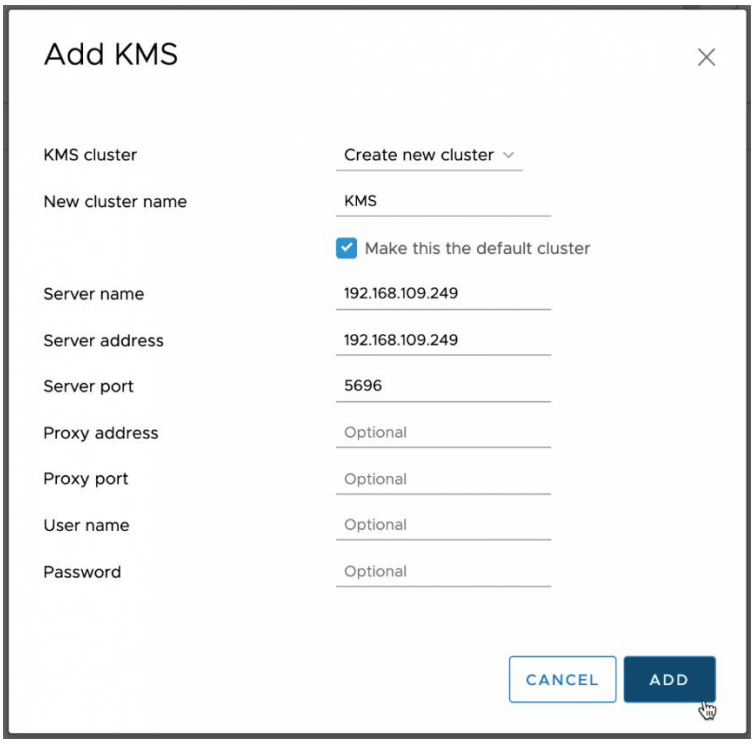
To add the KMS, select vCenter in the **Navigator** panel, choose **Configure**, then select **Key Management Servers**.



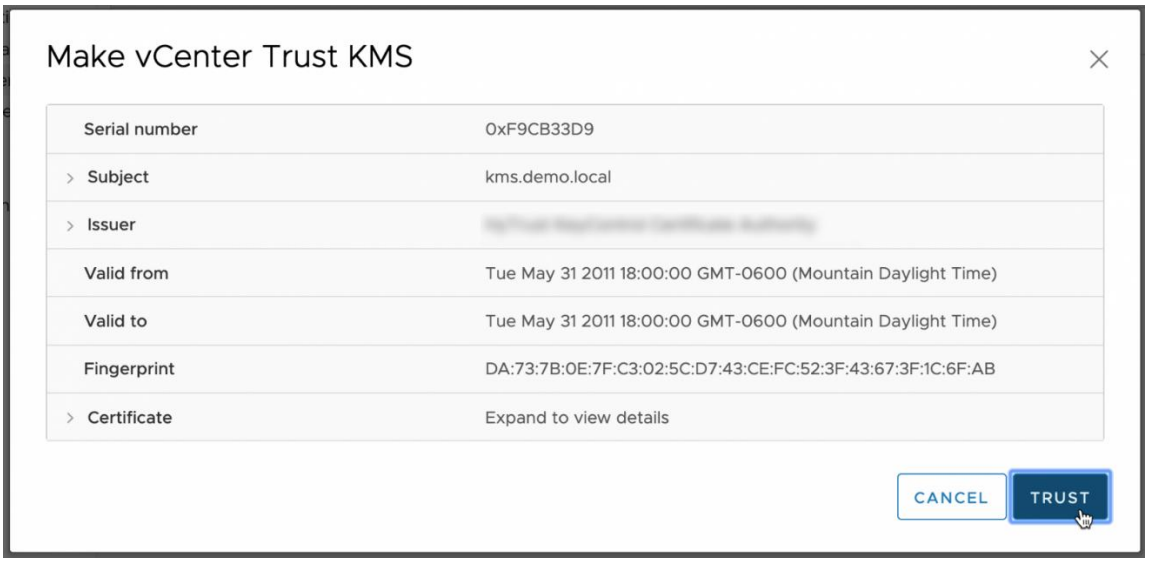
If the **Key Management Servers** option isn't visible, the account logged in does not have Manage KMS privileges in vCenter Server.



To add a KMS, select Add KMS and add the KMS cluster properties. Depending on the network configuration, a proxy and credentials may be required. The KMS Server port will normally be 5696, but an alternate port may be used.



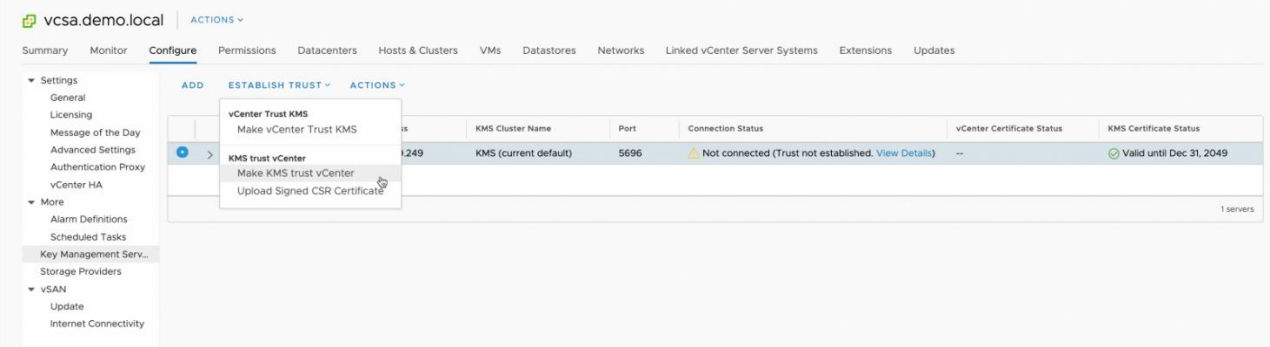
When asked to Trust the KMS Server's certificate, select Trust.



At this point, vCenter Server trusts the KMS server.

Have the KMS trust vCenter

For the KMS server to properly communicate with the vCenter Server, and ultimately vSAN hosts, the trust has to be bidirectional.



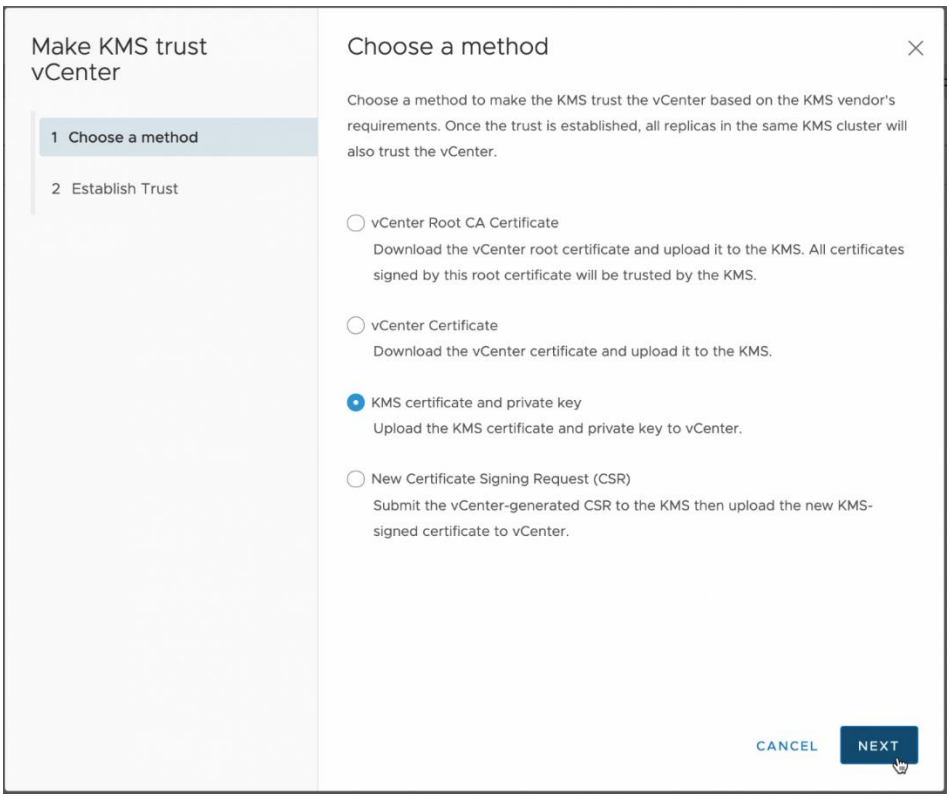
To establish the trust with the KMS, select **Establish trust with KMS**

One of four options will be available:

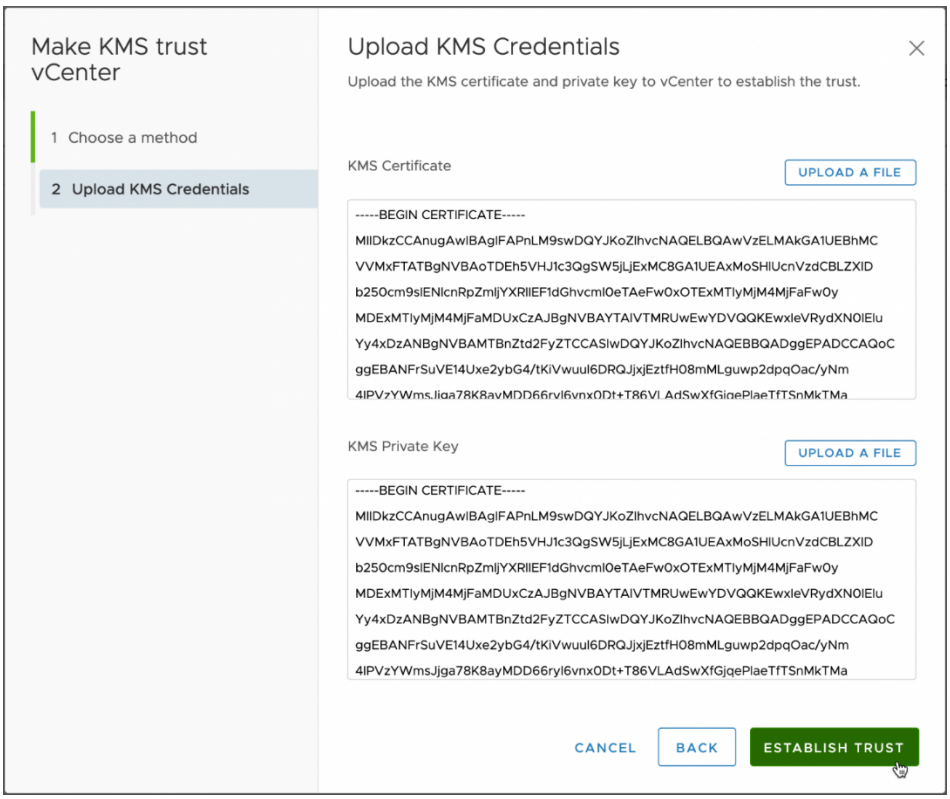
- Root CA Certificate
- Certificate
- New Certificate Signing Request
- Upload certificate and private key

Consult the vendor-specific documentation for the proper trust establishment process for the chosen KMS provider.

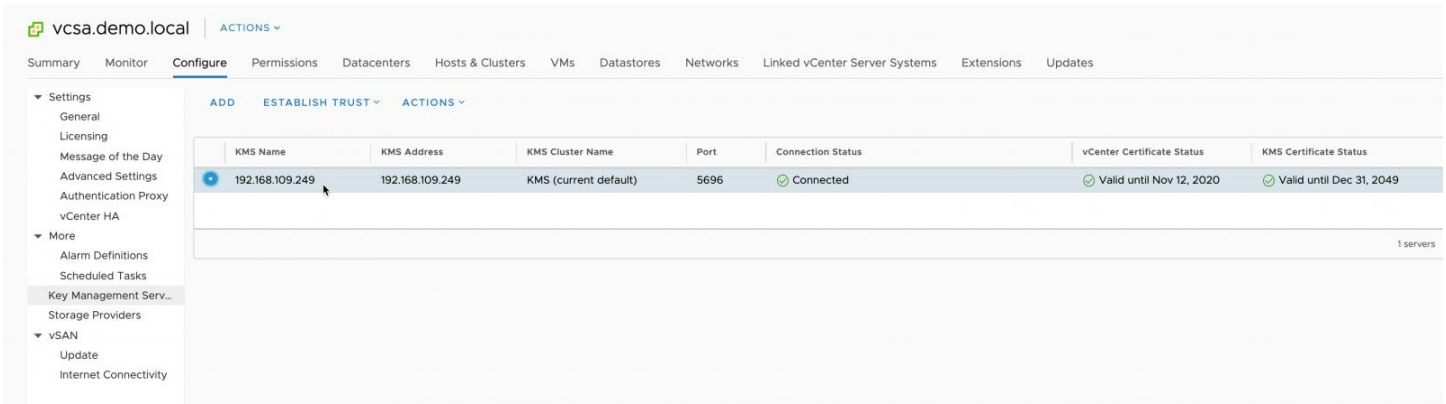
As an example of establishing trust using a certificate and private key, select **KMS certificate and private key**



When prompted, either upload the certificate file and private key file, or past the contents and select **Ok**.

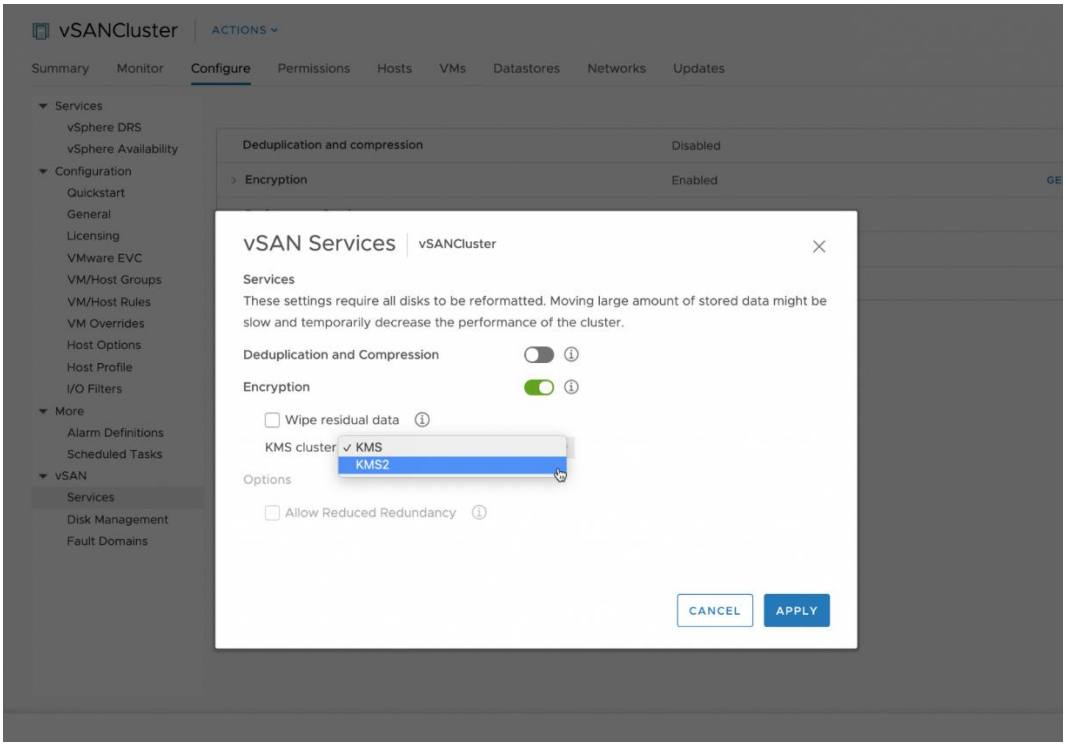


When the certificate and private key have been uploaded, the trust will be established.



Changing KMS Server

The process of changing KMS Servers is essentially a Shallow Rekey operation.



1. The initial KMS configuration is in place
2. The administrator selects an alternate KMS Cluster
3. The new KMS configuration is pushed to the vSAN hosts
4. A new host key is generated
5. vSAN performs a Shallow Rekey

To also perform a Deep Rekey, this should be accomplished after the initial Shallow Rekey has taken place.

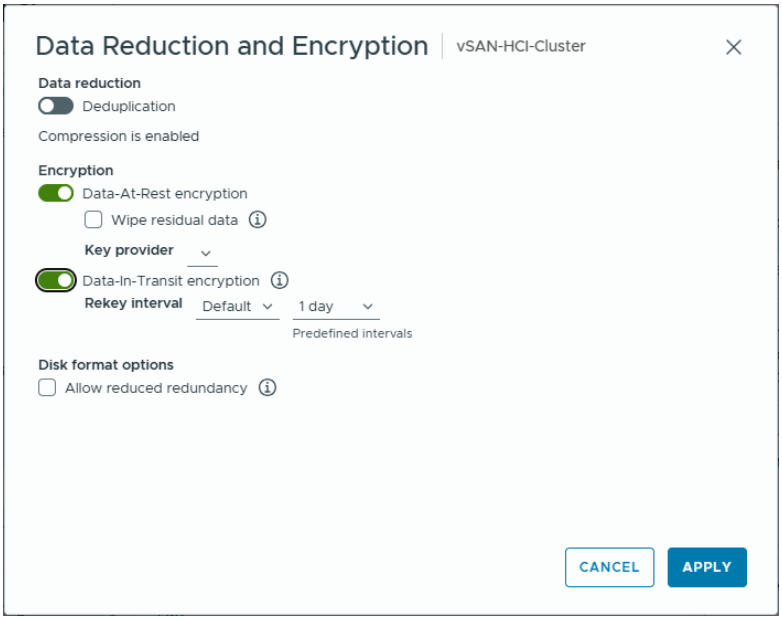
Changing the KMS Server via PowerCLI

The process of changing KMS Servers is essentially a Shallow Rekey operation but designating a new KMS Server. The Start-VsanEncryptionConfiguration cmdlet can be used to change the KMS.

Start-VsanEncryptionConfiguration -Cluster "ClusterName" -KMS "KMS Profile Name" -Confirm:$False (Confirm false to proceed without prompting)

Secure Device Wiping Options

The "Wipe residual data" (formerly Erase disks before use) is an option available in the vSAN Services dialog box.



When a vSAN cluster has this setting checked, random data is written to each device before it is encrypted. Using random data helps ensure a fully secured wipe, as only ones or only zeros may not have the desired effect when using deduplication and compression.

This process can be very time-consuming. The performance characteristics of the media type, bus protocol, capacity, and number of devices will all impact the time that it takes to complete this operation. This wipe process aligns with the NIST 800-88 Revision 1 "Clear" definition:

The "**Wipe residual data**" option meets the requirements of each of these. Though all zeros are not being written, random data (a more complex value) is written instead.

A secure wipe feature is available that will allow for a securely erased device by NIST standards. Below is an example of PowerCLI commands used to achieve this result.

Log into the server

PS C:\windows\system32> Connect-VIServer '[REDACTED]' -UserName 'administrator@vsphere.local' -Pass [REDACTED]

Get host

PS C:\windows\system32> $h = Get-VMHost

Query wipe status

PS C:\windows\system32> $s = VsanWipeDiskStatus -VMHost $h[0] -CanonicalName ("mpx.vmhba0:C0:T3:L0)

Start wipe disk

PS C:\windows\system32> Start-WipeDisk -VMHost $h[0] -CanonicalName "mpx.vmhba0:C0:T1:L0"

Confirm

Are you sure you want to perform this action?

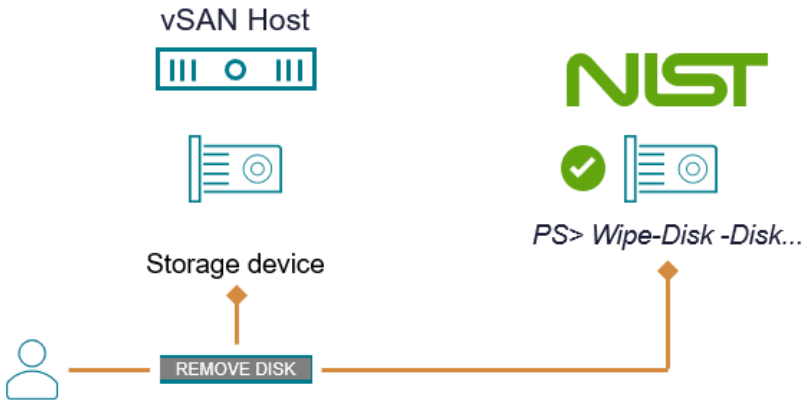
Performing this operation "Start wipe vSAN disks" on target "mpx.vmhba0:C0:T1:L0".

[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): Y

Stop wipe disk

PS: C:\windows\system32> $ret = Stop-WipeDisk -VMHost $h[0] -CanonicalName (“mpx.vmhba0:C0:T3:L0”)

PS: C:\windows\system32> $ret.Success



vSAN data-at-rest encryption encrypts all the vSAN object data that reside on the storage devices across a cluster. If storage devices have been repurposed from other servers, that unrelated residual data may have not necessarily been securely wiped, and in theory could be recovered for nefarious means. This wiping process ensures there is no residual data on a storage device used by vSAN.

Recommendations for “Wipe residual data” when using vSAN Encryption are:

- Select “Wipe residual data”
  - When enabling vSAN Encryption for existing vSAN clusters that have vSAN objects on them
  - When adding a host that has data on local devices to an encrypted vSAN cluster
  - When performing a rekey operation to invoke a deep rekey (requesting a new KEK and new unique DEKs created for each vSAN storage device)
- Deselect “Wipe residual data”
  - When enabling vSAN Encryption for a new vSAN cluster that has not previously had data on the vSAN devices
  - When adding a host that has not had data on local devices that is being added to an encrypted vSAN cluster
  - When performing a rekey operation to invoke a shallow rekey (only requesting a new KEK)

Replacing vCenter Server when vSAN Encryption is Enabled

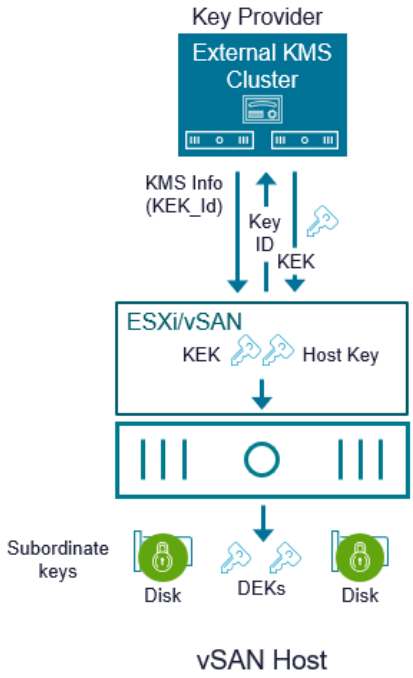
To recover from these types of scenarios, it will be necessary to create a new cluster with the same exact configuration that was originally in use by vSAN Encryption. The same KMS must be used, as well as have the same KMS Cluster ID. It is imperative that the same KMS cluster ID remains for the recovery feature to work.

Although the old vCenter Server is no longer available, the hosts still have the information and keys from the KMS cluster, if we connect to the same KMS cluster with the same cluster ID, the hosts will be able to retrieve the key (assuming the key still exists and was not deleted). The KMS credentials will be re-applied to all hosts so that hosts can connect to KMS to get the keys.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) on each host and provide the key to the host when the key provider is inaccessible.

vCenter is lost and KMS information isn't documented

If vCenter is lost, and in cases where the KMS are KMS Cluster ID are not documented, these items can be recreated with a newly deployed vCenter? In the diagram below we see how the keys are distributed to vCenter, hosts, etc. The KMS server settings are passed to hosts from vCenter by the KEK_id.



vSAN Host

In order to obtain the KMIP Cluster ID, we need to look for it under the esx.conf file for the hosts. You can use cat, vi, or grep (easier) to look at the conf file. You want to look for kmipClusterId, name(alias), etc. Make sure the KMS cluster on the new vCenter configured exactly as it was previously.

cat /etc/vmware/esx.conf

or something easier…

grep "/vsan/kmipServer/" /etc/vmware/esx.conf

```
/vsan/kmipServer/child[0000]/old =
/vsan/kmipServer/child[0000]/port =
/vsan/kmipServer/child[0000]/address =
/vsan/kmipServer/child[0000]/kmskey =
/vsan/kmipServer/child[0000]/userName =
/vsan/kmipServer/child[0000]/name =
/vsan/kmipServer/child[0000]/kmipClusterId =
```

After the KMS cluster has been added to new vCenter as it was configured in the previous vCenter Server, there is no need for reboots. During reconfiguration the new credentials will be sent to all hosts and such hosts should reload keys for all disks in a few minutes.

Shallow Rekey via UI

A Shallow Rekey is performed to change the KEK associated with a vSAN cluster. This is a simple process that can be accomplished from the vSphere Client.

Select the vSAN cluster, **Configure**, then **Services**, and select **Generate New Encryption Keys**

This process will create new a new KEK for the cluster and push it to the hosts. Each device's DEK will then be re-wrapped with the new KEK+DEK combination.

When using the vSAN ESA, a shallow rekey procedure is as follows:

- vCenter Server creates a new KEK, and replaces the existing KEK with a new one by generating the KEK and wrapping the DEK with it on one host.
- vCenter Server will then persist all relevant info to the cluster state (KEK_ID, KMS info, etc.)
- vCenter Server will then update encryption configuration on each host in the cluster
- The updated wrapped DEK is stored in the config store, instead of disk metadata.

Deep Rekey via UI

A Deep Rekey is also a simple process accomplished by vCenter Server. The process is the same as performing a Shallow ReKey, with the **Also re-encrypt all data on the storage using new keys option**. This will trigger a Disk Format Change (DFC). A DFC will evacuate data from each device in the same fashion as the process of enabling encryption. For more information, see the post: "Key Rotation Options for vSAN ESA in VMware Cloud Foundation 5.1 and vSAN 8 U2."

Notice that when a Deep Rekey is selected, the **Allow Reduced Redundancy** option is enabled. This should be considered when performing a Deep Rekey as considered when enabling or disabling encryption, depending on the number of available hosts (or fault domains) and available capacity.

A deep rekey operation for vSAN ESA is only available in vSAN 8 U2 and newer.

Shallow/Deep Rekey via API/PowerCLI

Rekeying is also available using PowerCLI or other vSAN Management API methods.

```
PS /Users/jase/PowerCLI> Start-VsanEncryptionConfiguration -Cluster "vSANCluster" -ShallowRekey -Confirm:$False

Name                           State      % Complete Start Time   Finish Time
----                           -----      ---------- ----------   -----------
Regenerate new keys for encry… Running             0 07:21:51 PM

PS /Users/jase/PowerCLI>
```

The PowerCLI cmdlet Start-VsanEncryptionConfiguration can perform a Shallow or Deep Rekey. The syntax is as follows:

Shallow Rekey:

Start-VsanEncryptionConfiguration -Cluster "ClusterName" -ShallowRekey -Confirm:$False (Confirm false to proceed without prompting)

Deep Rekey:

Start-VsanEncryptionConfiguration -Cluster "ClusterName" -DeepRekey -AllowReducedRedundancy (if desired) -Confirm:$False (Confirm false to proceed without prompting)

Add Non-Encrypted Host to vSAN Cluster

Adding a new host to an existing vSAN Cluster:

1. Add the host to the encrypted vSAN Cluster - Compute only nodes or nodes contributing storage to vSAN
    a. The hostKey will be installed on the host
    b. Configure a VMkernel interface with vSAN Traffic that will allow connectivity with the other hosts in the vSAN cluster
    c. The host will be able to access the encrypted vSAN datastore
2. Add one or more storage devices to hosts in a vSAN cluster

a. When disks are claimed, the KEK will be requested from the KMS cluster, disks will be added after a Disk Format Change occurs. This occurs without rebooting the host.
b. Data may be written as encrypted to the storage devices on the host.

vSAN Encryption Services Troubleshooting

Performance Degradations

Depending on the circumstances, the use of vSAN Encryption Services can have an impact on the potential performance capabilities of the cluster. This most often shows up with performing synthetic testing to determine the top-line performance capabilities of the cluster. For vSAN ESA, performance degradations tend to show up more when using vSAN Data-in-Transit encryption, as opposed to vSAN Data-at-Rest Encryption. This is because it must encrypt every network packet transmitting vSAN data across the hosts in the cluster, which may lower IOPS and throughput, and increase latency. Since vSAN ESA Data-at-Rest Encryption encrypts the data near the top of the stack, data that is distributed across the network is in fact already encrypted. It just cannot guarantee that a unique hash is created for every block temporarily stored in vSAN's log structured file system. There are two ways to address this matter.

- **Evaluate the performance impact of encryption on production workloads, not using synthetic testing.** It is synthetic testing that can skew the perceived impact on performance, as synthetic tests are typically committing every CPU cycle possible for the purpose of writing data, which is not reflective of a real workload.
- **Determine if vSAN Data-at-Rest Encryption may be good enough to meet your security requirements in your topology**. Of the two data services, vSAN Data-in-Transit typically has the most significant performance impact. Since vSAN ESA encrypts data across the network even when using just Data-at-Rest encryption, you may find that for vSAN HCI clusters, using just Data-at-Rest Encryption may meet your security requirements to encrypt storage traffic across the network.

If you do have a requirement to encrypt all storage traffic, and you are using a disaggregated deployment model using vSAN storage clusters, then you will want to enable vSAN Data-in-Transit encryption on the vSphere clusters mounting the remote datastore (available in vSAN for VCF 9.1 and later). This ensures that the storage traffic between the client cluster and the server cluster is encrypted.

For more information, see the post: "Performance when using vSAN Encryption Services." The post pre-dates the introduction of vSAN ESA, which encrypts data much more efficiently than the original storage architecture. However, the concepts discussed in the post may generally apply to both architectures.

External KMS Server Accessibility

The availability of the key provider (whether it be an external KMS, or using the vSphere Native Key Provider in vSphere) plays an important role in the proper distribution and management of keys.

If the KMS resides on the datastore it is providing key management for, and the hosts do not have TPMs to persist the keys, storage devices claimed by vSAN will not be mounted if the keys are unavailable. Hosts in a vSAN cluster that has vSAN Encryption enabled and do not have TPMs to persist the keys, will directly contact the KMS they are assigned to upon boot up to unlock/mount storage devices.

Consider the following scenario:

1. KMS resides on a vSAN cluster that has vSAN Encryption enabled.
2. Hosts that have KMS disks for a virtualized KMS appliance lose power. The KMS is then not accessible.
3. Those hosts are rebooted, and attempt to connect to the (now unavailable) KMS appliance.
4. The previously failed vSAN hosts will boot, but will not unlock or mount the storage devices.
5. The KMS appliance's disks are still not available and will not be.

While it remains advisable to house the KMS appliances in a location (perhaps a management cluster) other than the vSAN datastore it is providing keys for, using TPMs in the hosts will ensure that keys will persist in the event that there is an issue with the KMS on the cluster, or anywhere else in the infrastructure.

Sample PowerCLI code exists that can be used to check and see if a KMS appliance is residing on the vSAN Cluster it is providing key management for located here: https://code.vmware.com/samples/3773/

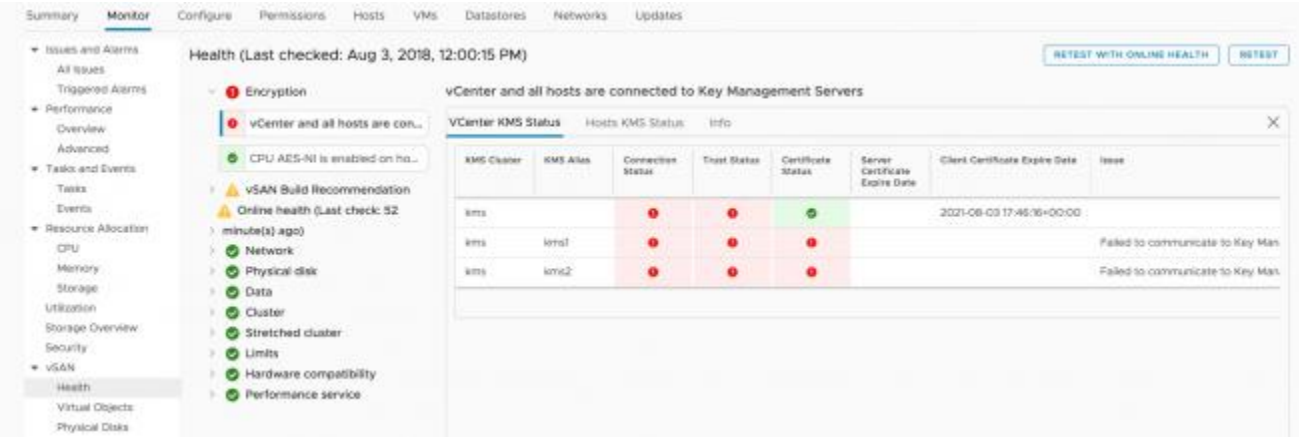
Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) and ensure keys can be used under these types of failure conditions.

External KMS Profile Addressing

When using vSAN Encryption, one of the vSAN Health Check tests will show the health of the connection between the vSAN Hosts and the KMS Cluster as well as vCenter and the KMS Cluster. A recent scenario came up where the vSAN Health Check indicated that the vSAN Hosts could properly communicate with the KMS Cluster, but the vCenter server had intermittent connectivity to the KMS Cluster.

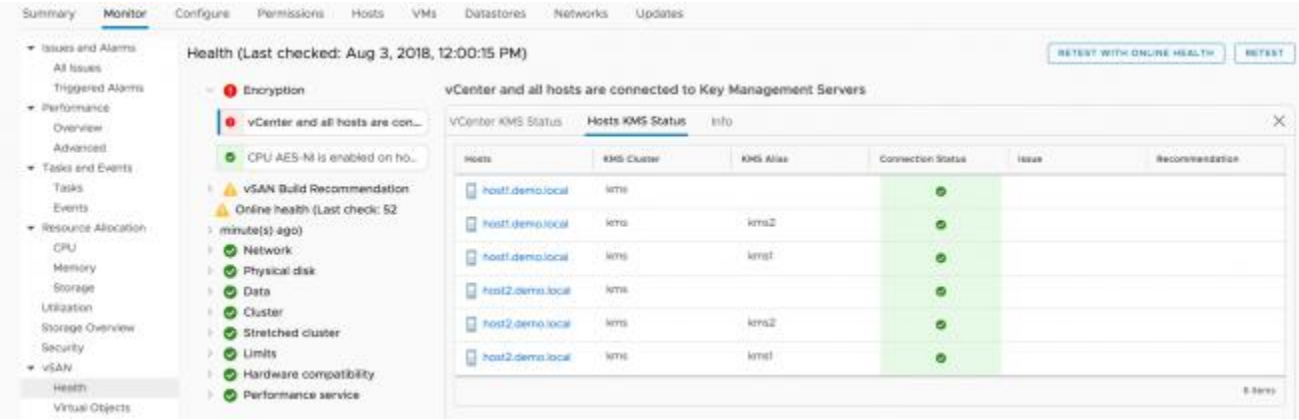
Troubleshooting indicated that there were no blocked ports between the vCenter Server and the KMS Cluster as well as they were able to properly ping each other. vSAN Hosts could properly ping the KMS Cluster as well, and no ports were blocked.

Here is the vSAN Health Check's reported error for **the vCenter KMS Status**.



Notice that the certificate status is valid, but the connection and trust statuses are not.

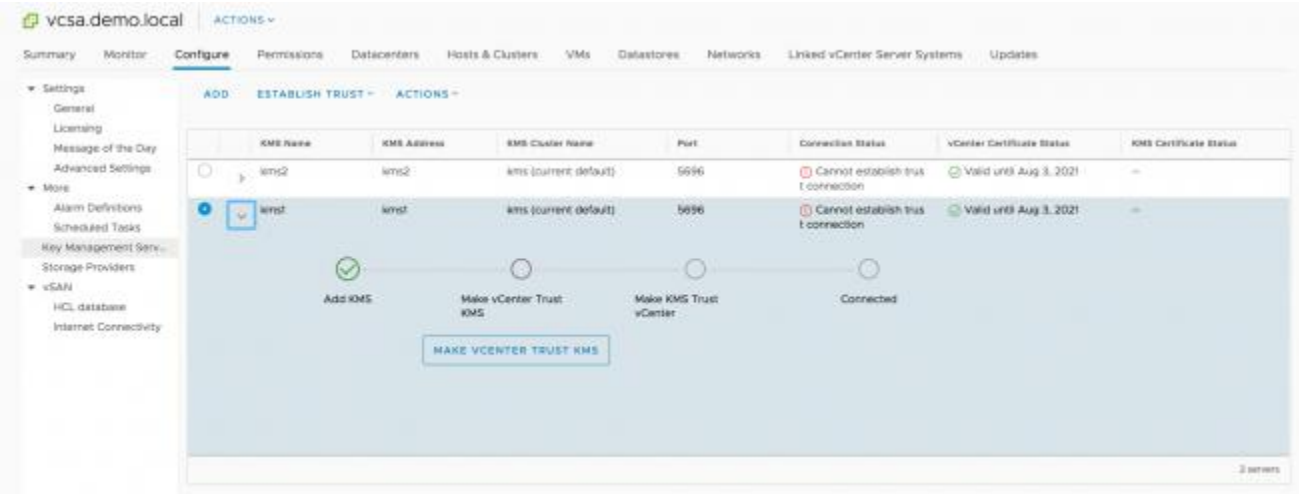
Looking at the **Host KMS Status** it can be seen that the hosts are properly communicating with the KMS Server.



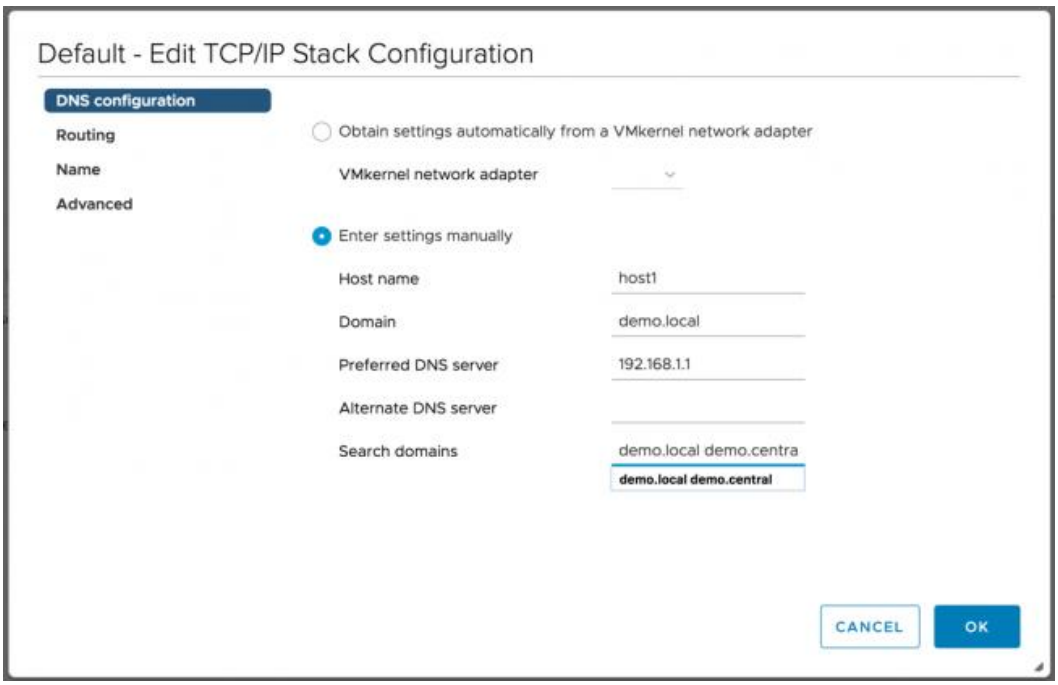
The process of enabling vSAN Encryption includes the following steps:

1. A KMS Connection Profile is created in vCenter and the trust is established.
2. vSAN Encryption is enabled in the Configuration>Data Services menu in the vSAN UI.
3. The KMS Connection Profile is pushed to each of the ESXi hosts, they use the kekId and hostkeyId in this profile to retrieve the KEK and HostKey for the vSAN Cluster.

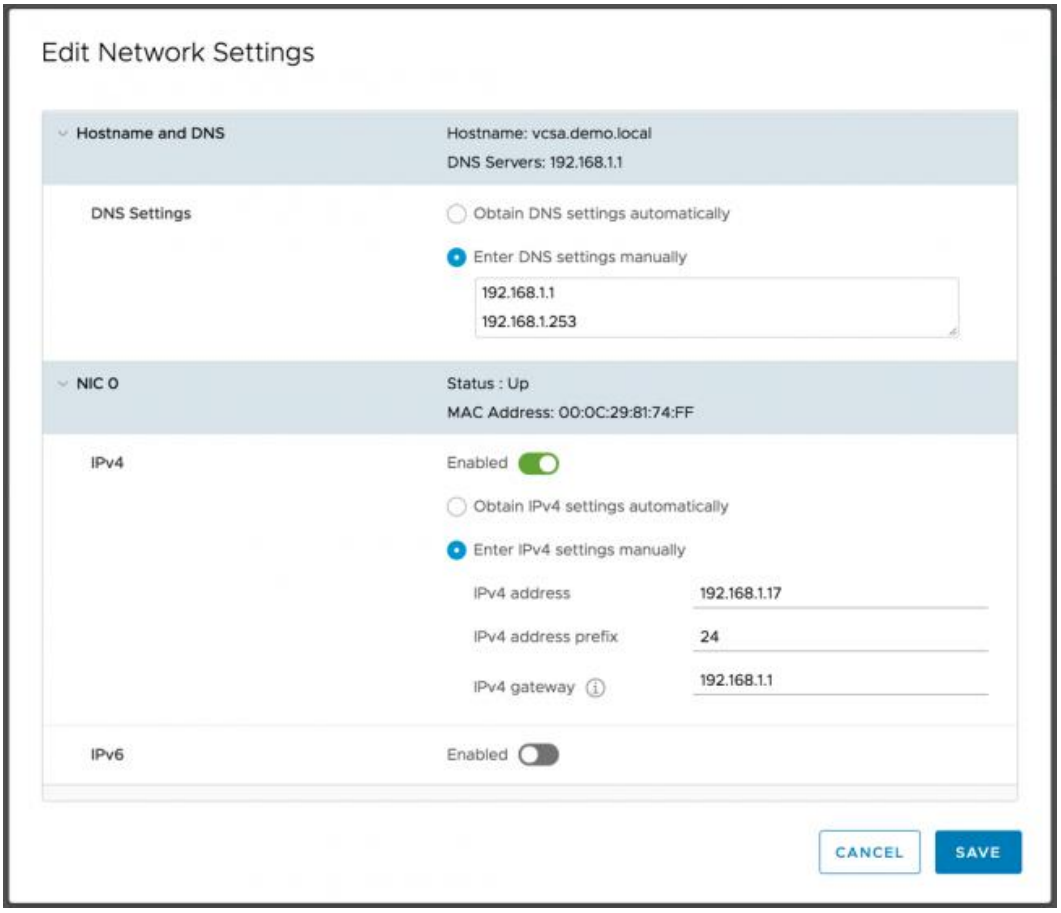
The connection has to be correct in vCenter Server before it can be correct/pushed to vSAN Hosts. Something must have changed in the environment to cause this issue. Further investigation indicated that the connectivity to the KMS Cluster was intermittent. Sometimes **the vCenter KMS Status** reported **green** and other times reported red. So maybe nothing changed. Careful review of the **vCenter KMS Status** and **Host KMS Status** health checks, the **KMS Alias** is a "short name." Maybe there is an issue where the short name is intermittently resolved from DNS… But the vSAN Hosts were not showing any intermittent connectivity, only the VCSA. The **Key Management Servers** configuration Profile in the vCenter's settings shows that the trust cannot be established. The **KMS Address** is the same value as the **KMS Alias** in the vSAN Health Check.



When using a short name, the default TCP/IP stack of a vSAN host uses designated search domains in the name resolution process. In the case of this cluster, **demo.local** and **demo.central** can be used in short name resolution.

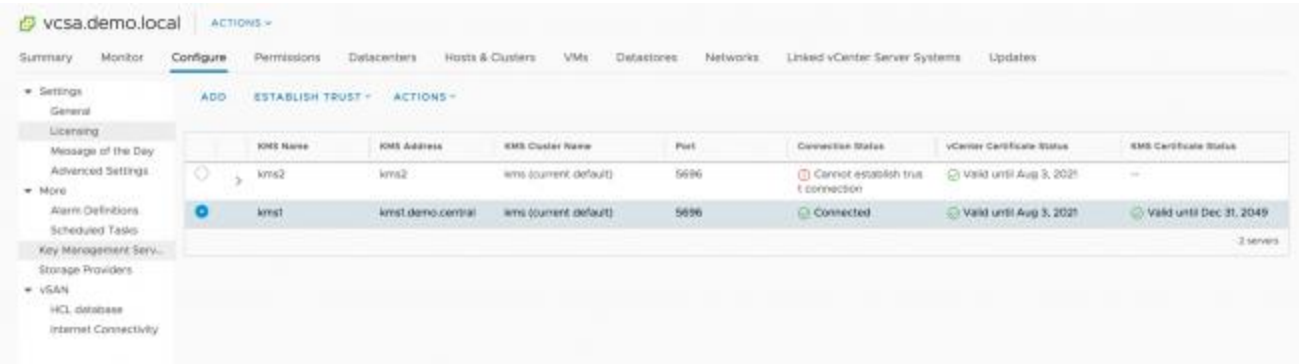


The VCSA, on the other hand, does not have any search domains:

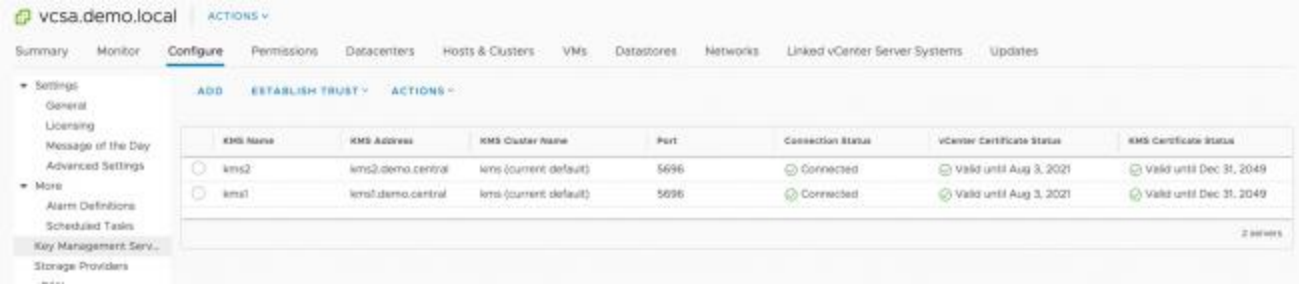


Without search domains to assist with the short name, vCenter Server would rely on the DNS server for name resolution.

The suggestion was made to change the **KMS Address** value for each KMS Cluster node to either an IP address or the Fully Qualified Domain Name (FQDN). Changing one of the two KMS entries showed some success.



Adjusting the **KMS Address** for the alternate KMS Cluster node cleared the issue up entirely.



In the case that this was brought up, an alternate vCenter Server had no issues connecting to the KMS Cluster, but an IP address was used instead of a short name. Without digging into DNS configurations of the environment, setting the Fully Qualified Domain Name (FQDN) resolved the issue.

In summary, when configuring the **Key Management Server** connection profile for a **KMS Cluster**, ensure that the **KMS Address** is one that vCenter and vSAN hosts can correctly resolve. Using a Fully Qualified Domain Name or IP address can prevent "short name" related issues.

Booting when vCenter Server is Unavailable

The Host Key and KEK are not stored on vSAN hosts, but rather stored in the key cache after being requested by the vSAN host when vSAN Encryption is enabled. When a vSAN host reboots, these keys are discarded. When a vSAN host reboots, because the Host Key and KEK are not present, they must be requested directly from the Key Management Server if the hosts do not use a TPM, which will persist the keys safely on the host. The Key Management Server profile, Host Key Id, and KEK Id information stored in **/etc/vmware/esx.conf** is used to request the Host Key and KEK.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) and ensure keys can be used under these types of failure conditions.

The example below demonstrates the values that can be found in **/etc/vmware/esx.conf** for vSAN Encryption:

```
/vsan/autoClaimStorage = "false"
/vsan/hostDecommisionState = "decom-state-none"
/vsan/kmipClusterId = "KMS"
/vsan/kmipServer/child[0000]/kmskey = "KMS/kms.demo.local"
/vsan/kmipServer/child[0000]/address = "192.168.1.29"
/vsan/kmipServer/child[0000]/kmipClusterId = "KMS"
/vsan/kmipServer/child[0000]/name = "kms.demo.local"
/vsan/kmipServer/child[0000]/port = "5696"
/vsan/kmipServer/child[0000]/old = "false"
/vsan/encryptionChanging = "false"
/vsan/hostKeyId = "7981401c-845d-11e8-b994-005056b058cb"
/vsan/kekId = "4a627731-8535-11e8-b994-005056b058cb"
/vsan/encryptionEnabled = "true"
```

KMS Profile Information

Host Key Id

KEK Id

When vSAN Encryption is enabled, or when a deep rekey operation is invoked, the vSAN host creates a unique DEK (XTS-AES-256) for each device, and it is encrypted with the KEK. A shallow rekey operation swaps out the KEK and rewraps each DEK. When a host with vSAN Encryption enabled attempts to mount a vSAN storage device, the DEK is unwrapped using the KEK, allowing vSAN to mount and then use the vSAN storage device.

The Boot Process

An encrypted vSAN cluster will demonstrate the following behavior at boot up given the set of conditions below:

- Entire vSAN encrypted vSAN cluster is offline.

- vCenter Server residing on offline vSAN cluster is also unavailable
- Hosts in offline vSAN cluster do not use TPM devices
- External KMS is online

In the scenario above, once the hosts are powered up, the disks are not immediately mounted, and thus the VMs, including vCenter server are offline.

1. The host boot process will read the values in **/etc/vmware/esx.conf** and request the KEK and Host Key from the KMS using the KEK Id and the Host Key Id respectively, directly from the KMS. This is because the previous keys were retained in the non-persistent key cache, and the hosts do not have a TPM to persist the keys.
2. The KEK and Host Key are placed in memory in the key cache. If the hosts used TPM devices, they would be cryptographically stored on the TPM devices in each host of the cluster. At this point the KEK is then used to mount the vSAN storage devices. The VMs, including vCenter Server can be powered on.

As long as vSAN hosts using vSAN Encryption have connectivity to their configured KMS or have them cached on a TPM, they have no issue booting, even when vCenter is offline. The boot process is not dependent on vCenter to unlock and mount storage devices claimed by vSAN

If the key provider (KMS) were also unavailable in this scenario, then the storage devices would not be mounted and the VMs would be unavailable. Using TPMs in each host helps avoid this scenario and improve the robustness of an encrypted cluster substantially.

Recommendation: Broadcom recommends the use of Trusted Platform Modules (TPM) on each server. This will allow for the keys distributed to the hosts to be securely stored on persistent media (the TPM) and ensure keys can be used under these types of failure conditions.

Summary

vSAN Encryption services can provide an effective strategy at securing your data to meet the requirements of your organization. Data can be encrypted at-rest, or in-transit, all exclusively through the capabilities of the hypervisor.

Additional Resources

vSAN technical blogs. Stay up to date on the most recently published technical information about vSAN. These posts are created by the vSAN Technical Marketing team.

VMware Resource Center. The location for design guides, operations guides and other technical white papers on vSAN. These assets are created by the vSAN Technical Marketing and Product Enablement teams.

Official vSAN documentation. The location for all “how to” documentation on vSAN.

VMware vSphere Security and Compliance Links. The latest information on security guidance and recommendations for vSphere.

About the Author

Pete Koehler is a Product Marketing Engineer in the VCF division at Broadcom. With a primary focus on vSAN, Pete covers topics such as design and sizing, operations, performance, troubleshooting, and integration with other products and platforms.

Appendix A: Common Terminology

Below are common encryption terms throughout this document, and how they pertain to vSAN Encryption Services:

- **Key Provider:** The entity providing keys. May be referring to either an external third party Key Management Server (KMS), or the vSphere Native Key Provider (NKP)
- **KMIP :** Key Management Interoperability Protocol.
  - A standard protocol that clients talk to KMS.
  - The KMIP 1.1 protocol is required for use with vSAN Encryption
- **KMS** : Key Management Server.
  - Several third-party vendors provide KMS solutions that are compatible with vSAN Encryption.
  - The current list of supported KMS solutions can be found at https://www.vmware.com/resources/compatibility/search.php?deviceCategory=kms
- **KMS Cluster** : A cluster of KMS servers.
  - The servers in the cluster maintain replication (mostly synchronous replication) so every key operation that renders a modification will be reflected by other server nodes immediately.
  - KMS cluster resiliency and availability is paramount to consider when implementing any encryption solution.
- **KEK** : Key Encryption Key.
  - This is the key stored in KMS. This is a per-tenant key, resulting in each vSAN cluster having one KEK.
  - Key Encryption Keys are AES-256
- **DEK :** Data Encryption Key.
  - This is the key used in the I/O path to encrypt/decrypt data.
  - Data Encryption Keys are XTS-AES-256 keys.
  - For vSAN ESA, a common DEK will be used across the cluster, wrapped by the KEK
- **OMEK:** Object Metadata Encryption Key
  - This key is responsible for object metadata, primarily generated by vSAN's LFS.
  - Each object's metadata is encrypted using its own OMEK, not shared by other objects
- **ODEK:** Object Data Encryption Key
  - This key is responsible for the actual object data
  - Each object is encrypted using its own ODEK, not shared by other objects.
- DMEK: Disk Metadata Encryption Key
  - This key is responsible for low level (LSOM) metadata
  - Each storage device has it's own DMEK.
- **Host Key** : This is similar to KEK, but is used to encrypt vSAN host core dumps, not data.
  - All hosts in a vSAN cluster use the same HostKey.
  - By providing a Host Key, customers can safely send encrypted core dumps to VMware Global Support without disclosing DEKs.
  - This assists in maintaining the integrity of customer data, while assisting VMware Global Support with problem resolution.
  - vSAN Host Keys are AES-256
- **Wrapped** : Wrapped is synonymous with encrypted.
  - "X" wrapped by "Y" means the clear text of "X" was encrypted using "Y" as the key, and the "Y" is needed to unwrap the wrapped key.
  - With vSAN Encryption, after the DEK is wrapped using the KEK, it is stored on persistent media.
- Rekey : change the key used in encryption.
  - **Shallow rekey :** change the KEK only. The DEK is wrapped with a new KEK. This is usually very fast.
  - **Deep rekey** : change the DEK for each device and re-encrypt all data using each device's new DEK. This will be very slow because all data needs to be rewritten.

- **TPM.** A Trusted Platform Module (TPM) is a device that sits inside of a host, that can cryptographically store keys that are issued to the host from the key provider. This module is an affordable (typically around $50 USD) way to ensure that host keys can be retrieved when there is an issue with communication to the key provider. **Whether an environment is using the vSphere Native Key Provider (NKP) or a dedicated KMS cluster, using a TPM in each host is highly recommended.**
- **Key cache** : A vSphere Host kernel module that caches the KEK from the KMS for use by vSAN Encryption and VM Encryption.
- **FIPS 140-x** : The Federal Information Processing Standard (FIPS) Publication 140-x, is a U.S. Government standard for computer security that is used to approve cryptographic modules. The title is Security Requirements for Cryptographic Modules. More information can be found on the NIST site .

vmware® by Broadcom

Copyright ©2025 Broadcom. All rights reserved.
The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.