

VSAN DATA AT REST ENCRYPTION

Table of Contents

[Introduction](#)

- [An introduction to Data at Rest Encryption](#)

[Overview](#)

- [Common Terminologies](#)
- [vSAN Data at Rest Encryption Specifics](#)
- [Native VMkernel Cryptographic Module](#)

[Architecture of vSAN Data at Rest Encryption](#)

- [Key Management Server](#)
- [vCenter Server](#)
- [vSAN Hosts](#)
- [vSAN Disk Groups](#)
- [Role Based Access Control](#)

[vSAN Data at Rest Encryption Workflows](#)

- [KMS Configuration](#)
- [Enable vSAN Encryption](#)
- [Disable vSAN Encryption](#)
- [Day 2 Encryption Operations](#)
- [Guidance when using "Erase disks before use"](#)
- [Replacing vCenter when vSAN Encryption is enabled](#)

[vSAN Encryption Troubleshooting](#)

- [KMS Server Accessibility](#)
- [KMS Profile Addressing](#)
- [Booting when vCenter is Unavailable](#)

[Native Key Provider for VMware vSAN](#)

- [Native Key Provider Operations](#)

[Additional References](#)

- [VMware Docs](#)

vSAN Data at Rest Encryption

Introduction

An introduction to Data at Rest Encryption

What is Data at Rest Encryption? The terms "Data at Rest Encryption" when used together, typically refer to data that is encrypted and stored, either in a transient or longer time frame, on some type of persistent media.

The purpose of data at rest encryption is essentially disallow access to the stored data without the appropriate key to unlock the data. In the event of media loss or theft, the data is secure without the presence of the unlocking key. Because of this, data at rest encryption is often employed in environments that require additional levels of security.

In a virtualized infrastructure environment, data at rest encryption can occur either inside a virtual machine or can be accommodated by the storage system.

Capacity Considerations

Virtualized environments that have many copies of common data, such as Virtual Desktop Environments, common server consolidation environments, or development and test environments often have the ability to have a significantly smaller capacity footprint when using deduplication and compression algorithms. This can be very advantageous to the overall cost of storage.

Enabling data at rest encryption in a virtualized environment has the potential of reducing this storage efficiency in some scenarios, while being transparent in others. This is entirely dependent on where the encryption process occurs.

- If encryption occurs within a virtual machine, at the host level, data that was once similar among many virtual machines, is no longer similar. This reduces the overall storage efficiency of deduplication algorithms. Data that might have once been easily compressible, will likely no longer be as compressible.
- If encryption occurs within the storage system, this could be also be the case or not. Encryption may occur in hardware or software, before or after data is written to persistent media. If encryption occurs before data services like deduplication and compression are implemented, the impact would be similar to in-guest encryption. If encryption occurs after data services are applied as data is being written to persistent media, overall consolidation ratios may not be impacted.

Compute Considerations

In today's enterprise storage arrays, data at rest encryption is often accomplished by the array controller. The array controller performs the task of encrypting data as it is written to persistent media rather than the array's storage processor. Enterprise arrays have the luxury of being able to use specific, albeit somewhat commodity, components. Specific selection of parts and their tolerances make the task of designing an enterprise storage array very controlled. Because of this specific instructions can be implemented to offload the task of data at rest encryption to these array controllers.

VMware vSAN was designed with the intention of being able to use any host on the vSphere HCL in conjunction with components that are certified on the vSAN Compatibility Guide. In essence, vSAN uses commodity parts that are readily available. Because these parts come in a wide variety of sizes, capabilities, and tolerances it is more efficient to handle data at rest encryption in software, without depending on the storage controller or storage devices. Instead, 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 through the use of AES-NI processor capabilities, vSAN can easily accomplish encryption with minimal additional overhead to vSphere hosts.

Data Loss/Theft Considerations

Depending on where virtual machine data is encrypted, there are different data loss/theft scenarios to consider.

When relying on storage array or local device encryption that is external to a virtual machine, that data is protected against the physical theft or loss of the device that contains the virtual machine's data. Loss can occur from maintenance operations such as degraded device replacement or intentional drive theft. This however does not protect from a rogue administrator 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.

Overview

Common Terminologies

Some common encryption terms and how they pertain to vSAN Encryption:

- **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.
 - Each disk in a vSAN disk group will have a DEK.
- **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.
 - Foo wrapped by Bar means the clear text of Foo was encrypted using Bar as the key, and the Bar 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.
- **Key cache** : A vSphere Host kernel module that caches the KEK from the KMS for use by vSAN Encryption and VM Encryption.
- **FIPS 140-2** : The Federal Information Processing Standard (**FIPS**) Publication **140-2**, is a U.S. Government standard for computer security that is used to approve cryptographic modules. The title is Security Requirements for Cryptographic Modules. It was initially published on 25 May, 2001. More information can be found on the [NIST site](#) .

vSAN Data at Rest Encryption Specifics

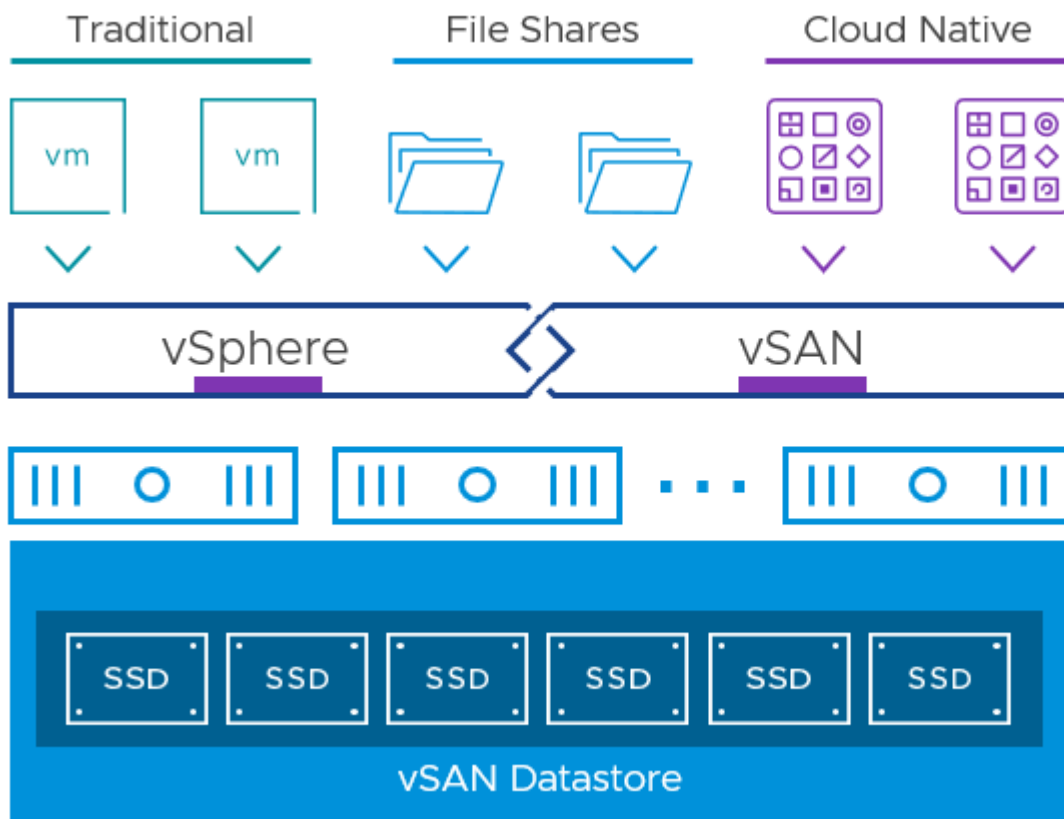
Since the release of vSAN in 2014, VMware has continued to add additional features and enhancements to vSAN. Data at rest encryption, is an important addition to the enterprise feature set of vSAN.

- Data Security
 - Data at rest is encrypted in both cache and capacity devices

- If cache or capacity devices are stolen or discarded, user data is securely encrypted and may not be retrieved
- New Non-Cryptographic Administrator role unable to perform encryption related tasks
- vSAN host core dumps are encrypted
- Operational Benefits
 - Encryption may be enabled or disabled for the whole vSAN cluster
 - Set and forget
 - No need to ensure encryption ruleset in Storage Policies
 - Performs rolling disk group reformat, similar to enabling deduplication and compression
 - Live ReKeying with no requirement to power off virtual machines
 - Minimal dependency on vCenter
 - Encryption continues to operate unaffected
 - Deduplication and Compression unaffected by enabling vSAN Encryption
 - Encryption occurs after deduplication and compression tasks
- Host/Architecture Requirements
 - vSAN Cluster Types
 - Hybrid and All-Flash vSAN configurations are supported with vSAN Encryption
 - vSAN Stretched Clusters are supported with vSAN Encryption
 - The vSAN Witness Host is not encrypted as
 - vSAN Licensing
 - vSAN Enterprise Edition licensing (per CPU or per Desktop) are required.

Native VMkernel Cryptographic Module

In December of 2017, VMware achieved FIPS 140-2 validation under the Cryptographic Module Validation Program (CMVP). The CMVP is a joint program between NIST and the Communications Security Establishment (CSE). FIPS 140-2 is a Cryptographic Modules Standards that governs security requirements in 11 areas relating to the design and implementation of a cryptographic module. vSphere 6.7 and vSAN 6.7 include the validated Cryptographic Module, using them for VM Encryption and vSAN Encryption features.



The VMware VMkernel Cryptographic Module has successfully satisfied all requirements of these 11 areas 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. The details of this validation, along with the tested configurations are available at: <https://csrc.nist.gov/Projects/Cryptographic-Module-Validation-Program/Certificate/3073>

The implementation and validation of the VMkernel Cryptographic Module shows VMware's commitment to providing industry leading virtualization, cloud, and mobile software that embrace commercial requirements, industry standards, and government certification programs.

Because the VMware VMkernel Cryptographic Module is part of the ESXi kernel, it can easily provide FIPS 140-2 approved cryptographic services to various VMware products and services.

Virtual machines encrypted with VM Encryption or vSAN Encryption work with all vSphere supported Guest Operating Systems and Virtual Hardware versions, and do not allow access to encryption keys by the Guest OS.

Architecture of vSAN Data at Rest Encryption

Key Management Server

Key management can be accomplished using a cryptographic appliance called a Key Management Server (KMS). KMS solutions 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.

The Domain of Trust

There are three parties participating in vSAN Encryption domain of trust

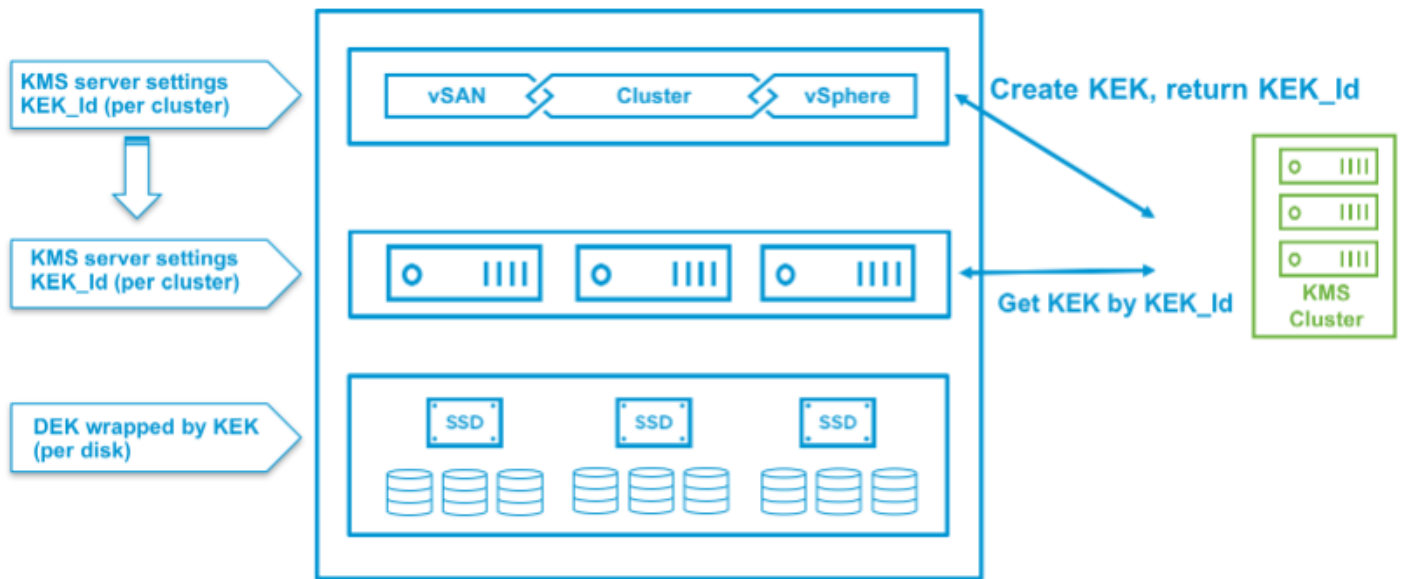
1. Key Management Server (KMS)
2. vCenter
3. vSphere Hosts with vSAN enabled (vSAN host)

Trust Establishment

VMware vCenter and vSphere hosts can only use a KMS after establishing a trust with the KMS. 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 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 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 is trusted after the KMS solution's certificate and private key are provided

Once the trust is established between the KMS and vCenter, a vSAN cluster (with vSAN Enterprise Licensing) may use vSAN Encryption. When vSAN Encryption is enabled on a cluster, the KMS cluster connection information and is pushed to the vSAN hosts. The vSAN hosts then provide a reference key, or key ID, to the KMS cluster. The KMS cluster then provides the Key Encryption Key (KEK) that is associated with the key id to the vSAN hosts. Disk keys (DEKs) are wrapped by the the KEK.



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.

- Enabling vSAN Encryption: Unique KEK is created for each vSAN cluster. DEKs are created when claiming vSAN cache and capacity devices. This is a rolling process.
- Disable 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.

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, KMS 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 KMS Server solution that provides a resilient and available KMS infrastructure is an important part of the vSAN Encryption design.

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. A current list of Key Management Server solutions that have been tested with vSAN Encryption and VM Encryption, can be found here:

<https://vmware.com/go/kmshcl>

KMS Infrastructure Placement

If using one or more KMS virtual appliances, they should not be deployed on an encrypted datastore.

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 unsupported scenario where A KMS appliance or cluster resides on the encrypted cluster it is providing keys for.

- 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.

In both of these cases, the KMS appliance/cluster is not available, because its storage is not available.

VMware **does not support** placing a KMS appliance or cluster on the encrypted datastore it is providing key management for.

*Note: The KMS name cannot be longer than 100 bytes in vSAN 6.6. [KB Article 52723](#) addresses this issue and provides a workaround.

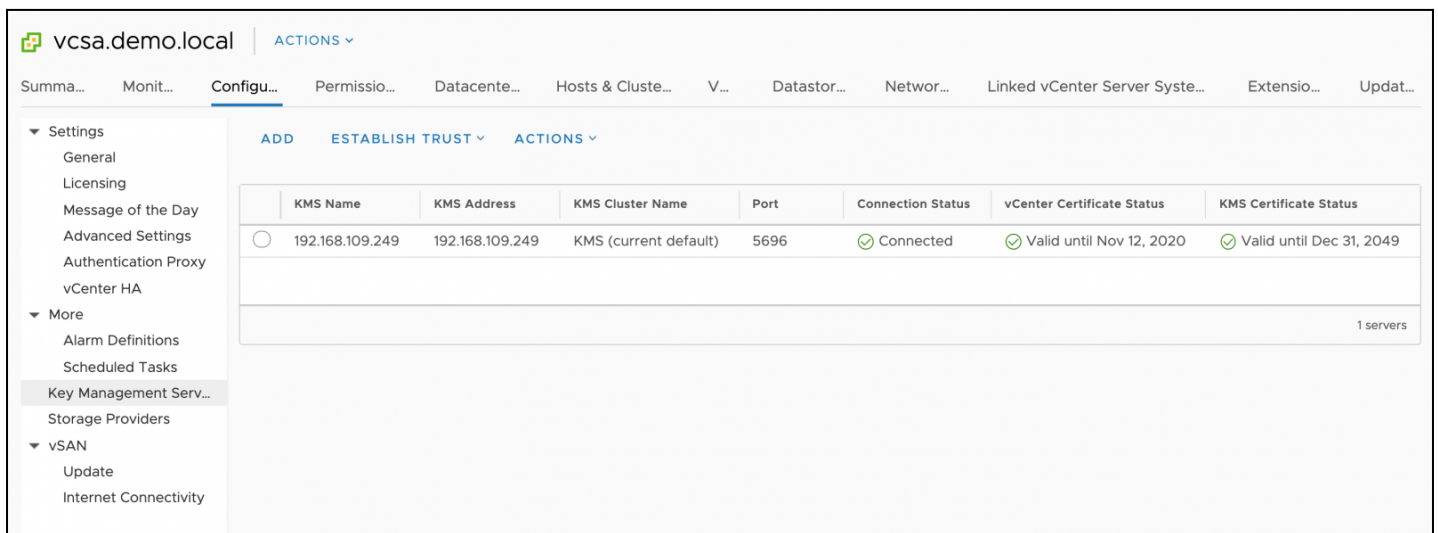
This has been resolved in vSAN 6.7.

vCenter Server

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.

KMS Configuration

In the previous section, key management services were described using a Key Management Server (KMS) or KMS Cluster. vCenter Server provides a central location for Key Management Server configuration that is available to be used by either vSAN Encryption or VM Encryption.



The screenshot shows the vCenter Server configuration page for Key Management Services. The page is titled 'vcsa.demo.local' and has a navigation menu on the left. The main content area shows a table with the following columns: KMS Name, KMS Address, KMS Cluster Name, Port, Connection Status, vCenter Certificate Status, and KMS Certificate Status. A single KMS entry is listed with a 'Connected' status and valid certificates.

	KMS Name	KMS Address	KMS Cluster Name	Port	Connection Status	vCenter Certificate Status	KMS Certificate Status
<input type="radio"/>	192.168.109.249	192.168.109.249	KMS (current default)	5696	Connected	Valid until Nov 12, 2020	Valid until Dec 31, 2049

1 servers

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 cluster configuration is often performed within the vSphere Web Client. vSAN is configured per vSphere cluster, and vSAN Encryption is a configuration option of a vSAN cluster. Provided a KMS has been configured, vSAN Encryption is easily enabled through the cluster management UI in the vSphere Web Client.

vSAN Services | vSANCluster ✕

Services

These settings require all disks to be reformatted. Moving large amount of stored data might be slow and temporarily decrease the performance of the cluster.

Deduplication and Compression (i)

Encryption (i)

Wipe residual data (i)

KMS cluster KMS ▼

Options

Allow Reduced Redundancy (i)

CANCEL
APPLY

When encryption is enabled a few options are available

- Wipe residual data (*formerly Erase disks before use*) - 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 during the enable/disable process. 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, similar to 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. The **no cryptography administrator** role was added with the introduction of VM Encryption in vSphere 6.5.

Users assigned to this role may not perform Cryptographic operations in vSphere 6.5 environments, including those with vSAN 6.6.

The screenshot shows the vSphere Roles configuration interface. On the left is a navigation pane with categories like Administration, Access Control, Licensing, Solutions, Deployment, Support, Certificates, and Single Sign On. The 'Roles' section under 'Access Control' is selected. The main area shows a list of roles with columns for Description, Usage, and Privileges. The role 'No cryptography administrator' is highlighted with a red box. The 'DESCRIPTION' tab is active, displaying the text 'Full access without Cryptographic operations privileges'.

Operations not associated with cryptographic tasks can be performed by uses assigned to this role.

Some of the normal vSAN operations that might be considered to be impacted by this role:

- Operations that are not allowed are
 - Manage KMS
 - Manage encryption policies
 - Manage keys
 - Register a VM
 - Register a Host
- Operations that are allowed
 - Adding an existing host to a vSAN Encrypted cluster
 - Moving a VM from a non-encrypted datastore to the encrypted vSAN datastore
 - Moving a VM from a non-encrypted vSAN cluster/datastore to an encrypted vSAN cluster/datastore

vSAN Hosts

Understanding how vSphere hosts are configured and behave in an encrypted vSAN cluster is very important. There are a few differences between standard vSphere hosts when vSAN 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 vSAN Disk Groups 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 KMS Cluster over the Management VMkernel interface to securely retrieve the KEK using the KEK_ID (which was pushed by vCenter when enabling vSAN encryption). 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.

The boot process for vSAN hosts participating in an encrypted vSAN cluster is as follows:

1. The ESXi host boots and once hostd starts, the system appears to be available in vCenter, but it isn't just yet.
2. vSAN tells hostd to disable all core dumps on the host
3. vSAN requests a unique key from the KMS for the purpose of encrypting the host core dumps
4. vSAN tells hostd to store the host key in the kernel key cache
5. vSAN passes the KEK_ID to the KMS
6. The KEK retrieved it placed into the kernel key cache, which is used to mount vSAN disks
7. vSAN mounts the encrypted disks.

This sequence of events differs from the normal boot process, as vSAN disks are mounted before hostd becomes available. Because of this process, when vSAN Encryption is enabled, it is possible to have a vSAN host "up" but still performing the process of mounting vSAN disks.

*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 by creating disk groups on the newly added host, or it may be requested from a reboot of the newly added host.

vSAN Disk Groups

Encryption occurs within each device that is part of a disk group. To better understand how vSAN Encryption works, it is important to understand the Disk Format Change process as well as the process of encrypting data at rest.

The Disk format change (DFC) process

Every time vSAN Encryption is enabled or disabled the each disk group in the vSAN cluster must go through a Disk Format Change (DFC). When enabling encryption, a new partition is added that holds a small amount of meta-data used by vSAN to manage operations on the encrypted cluster. This step essentially prepares the disk to encrypt any write that is directed to it.

A "generation-id" is created the 1st time encryption is enabled. Each time encryption is then disabled, reenabled, or a Deep ReKey is performed, the generation-id increments by a value of 1. The DFC process evaluates the generation-id for each device in a disk group to determine if a DFC needs to occur on that disk. This is especially beneficial in cases where the DFC process is interrupted or in cases where hosts were offline during a Deep ReKey.

Disk configuration before vSAN Encryption is enabled:



Disk configuration after vSAN Encryption is enabled:

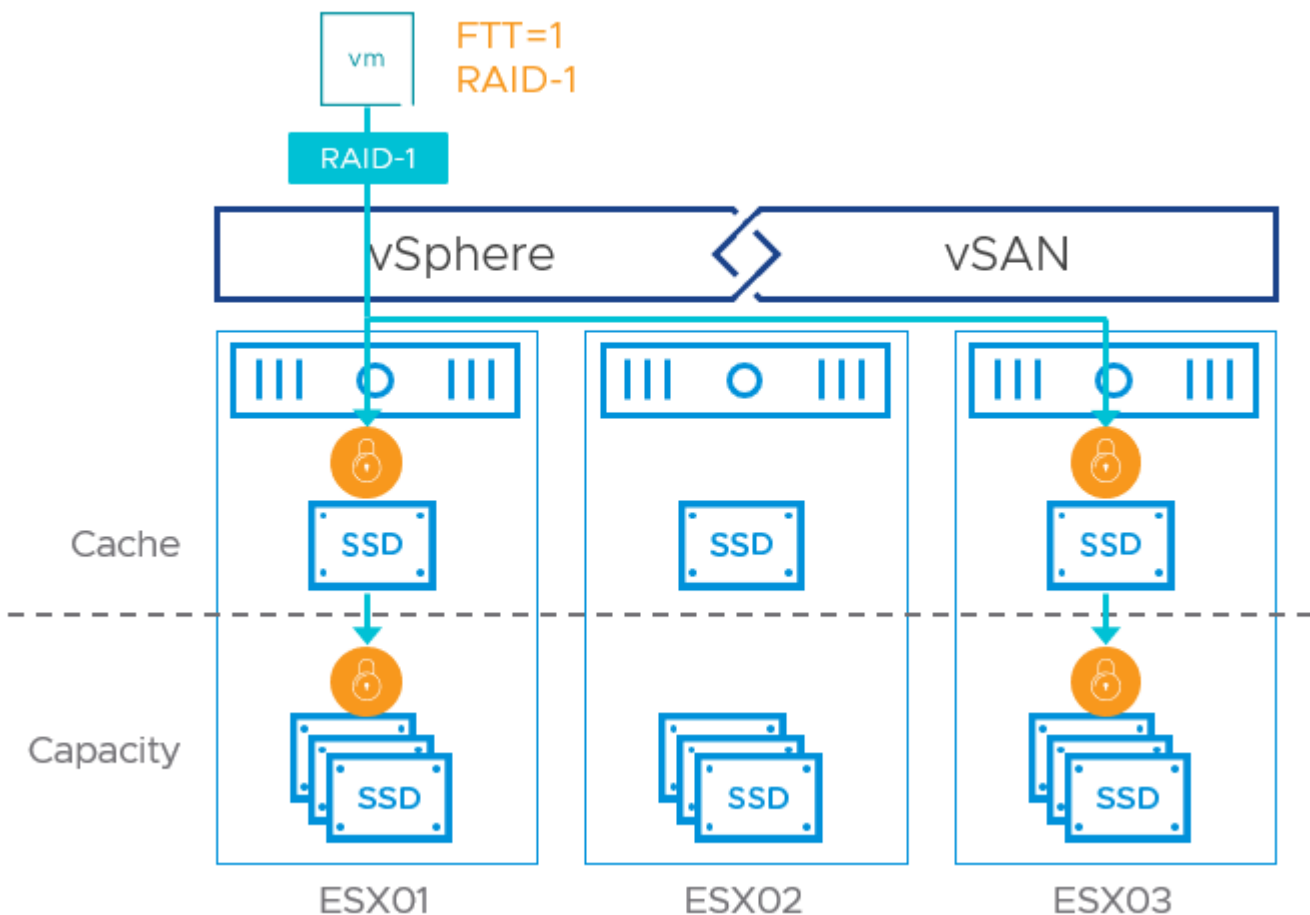


When Disk Format Change operations take place, things to consider are:

1. The DFC process is orchestrated as a rolling upgrade, one disk group at a time
2. If data is present on the disk, data is moved off of the disk before DFC is initiated, ensuring data is preserved
3. The optional feature “Erase disks before use” will soft erase the disk before writing new data.
 - o **Note** this step can be quite time consuming and should be used with care.
 - o It is not recommended to use this option unless the device is going through planned decommissioning or return to the manufacturer
 - o If the device is being permanently decommissioned or is going to leave the premises, it is recommended to combine “disk erasure” with other disk wipe utilities.

Writing data to an encrypted vSAN datastore

Encryption occurs as the last step on an I/O flow, for the highest level of protection and efficiency of deduplication and compression.



As data is written to the cache tier (write buffer)

1. Write I/O broken into 64K chunks
2. Checksum performed on 4K blocks
3. Encryption performed on 2K blocks
4. Lands in the write buffer

As data is destaged

1. Decryption is performed on 4K blocks
2. Deduplication is performed on 4K blocks
3. Compression is performed on 4K blocks
4. Encryption is performed on 2-4K blocks
5. Lands in the capacity tier

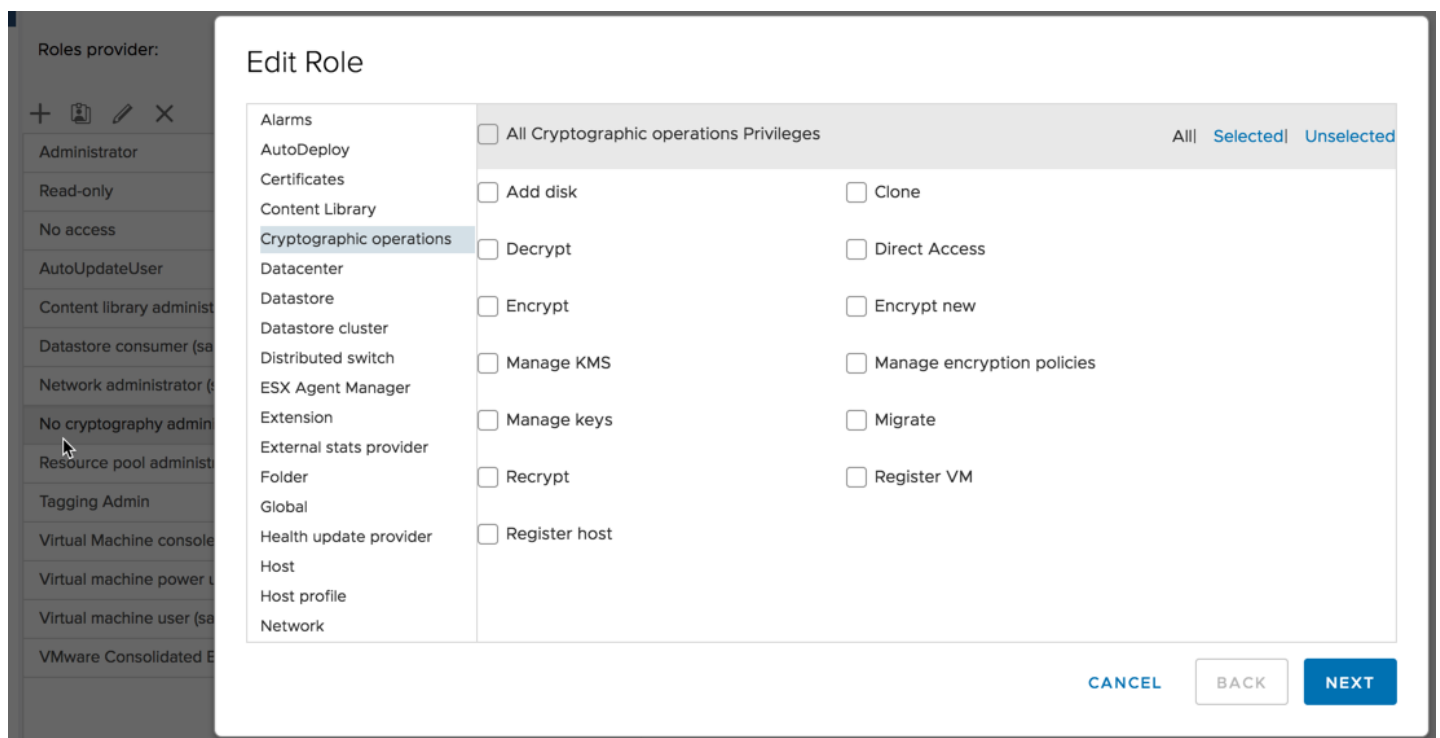
Data in flight is not encrypted, only data at rest.

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 use requirements.

No Cryptography Administrator Role

vSphere 6.5 introduced the No Cryptography Administrator role along with the introduction of VM Encryption. This 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 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 disable 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 at Rest Encryption Workflows

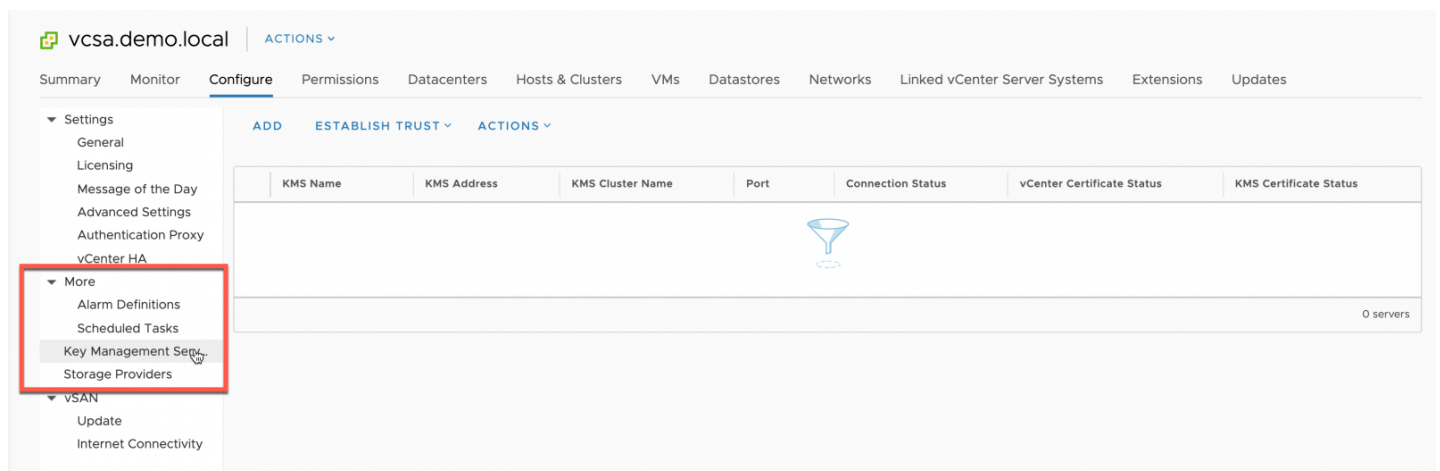
KMS Configuration

The process of configuring a KMS server is not difficult. The process can be broken down to these steps:

- Add the KMS
- Have the KMS trust vCenter

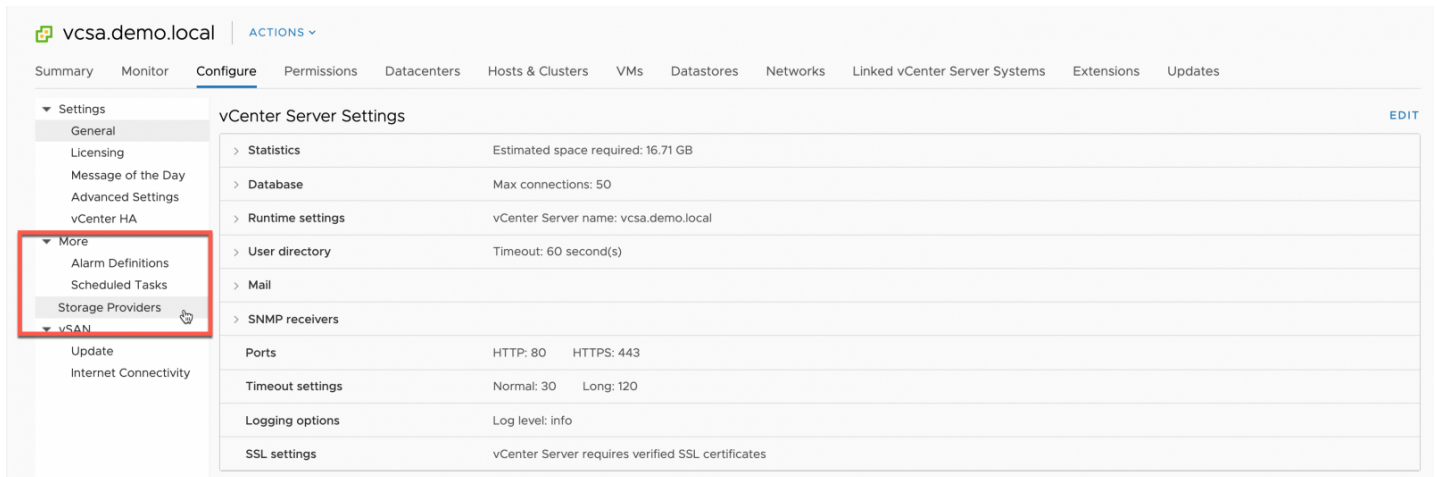
Adding the KMS

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



The screenshot shows the vCenter web interface for 'vcsa.demo.local'. The 'Configure' tab is selected in the top navigation bar. The left-hand 'Navigator' panel shows a tree view with 'Settings' expanded, and 'More' sub-items. 'Key Management Servers' is highlighted with a red box. The main content area shows a table with columns: KMS Name, KMS Address, KMS Cluster Name, Port, Connection Status, vCenter Certificate Status, and KMS Certificate Status. The table is currently empty, displaying a funnel icon and '0 servers' at the bottom right.

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



The screenshot shows the vCenter Server Settings page. The left sidebar has a red box around the 'More' section, with 'Storage Providers' highlighted. The main content area shows the following settings:

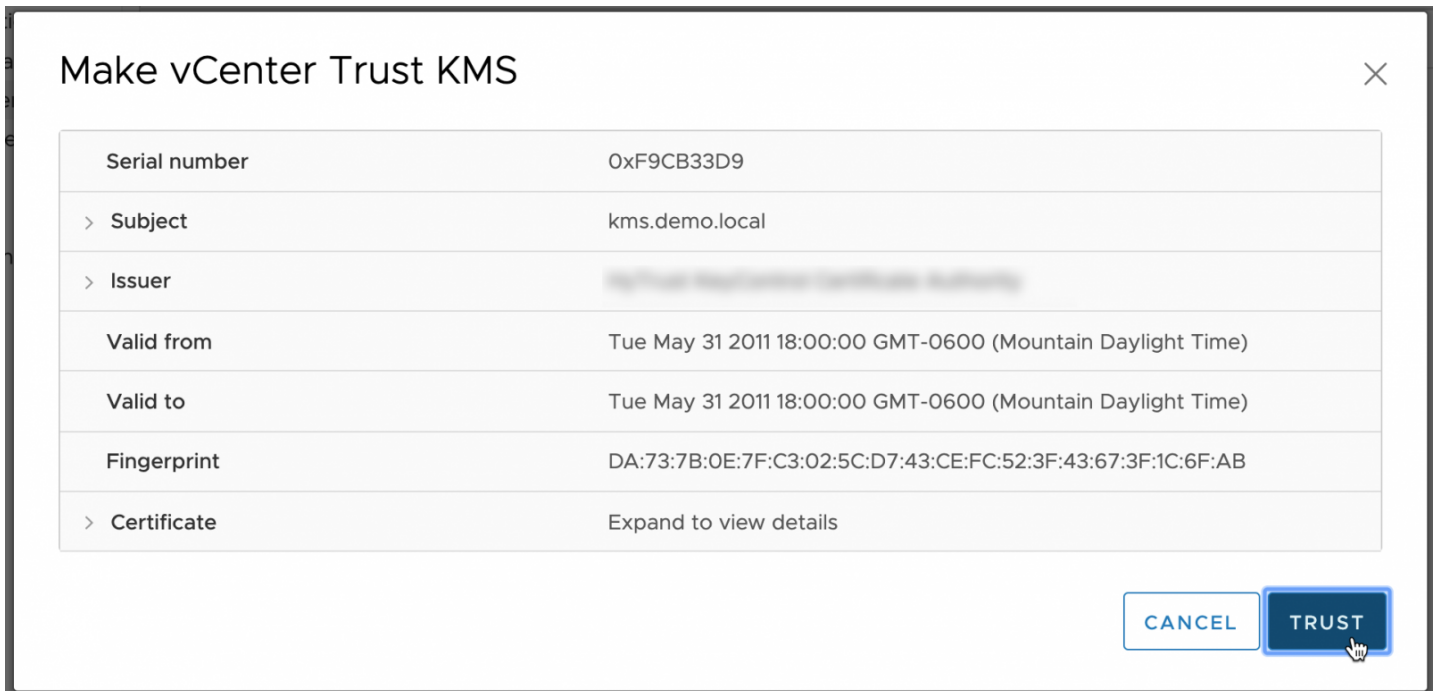
vCenter Server Settings		EDIT
> Statistics	Estimated space required: 16.71 GB	
> Database	Max connections: 50	
> Runtime settings	vCenter Server name: vcsa.demo.local	
> User directory	Timeout: 60 second(s)	
> Mail		
> SNMP receivers		
Ports	HTTP: 80 HTTPS: 443	
Timeout settings	Normal: 30 Long: 120	
Logging options	Log level: info	
SSL settings	vCenter Server requires verified SSL certificates	

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.

Add KMS ×

KMS cluster	Create new cluster ▼
New cluster name	KMS
	<input checked="" type="checkbox"/> Make this the default cluster
Server name	192.168.109.249
Server address	192.168.109.249
Server port	5696
Proxy address	Optional
Proxy port	Optional
User name	Optional
Password	Optional

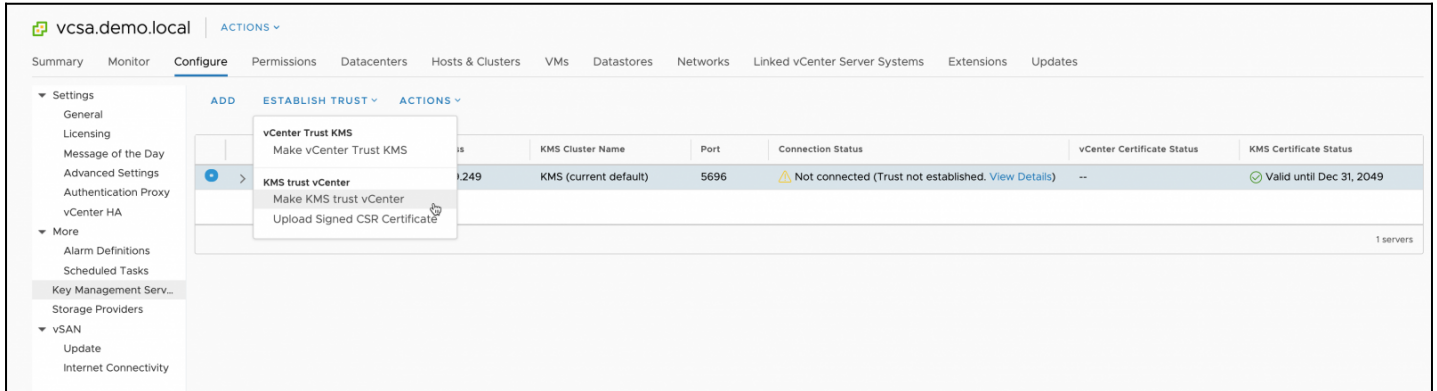
When asked to Trust the KMS Server's certificate, select **Trust**.



At this point, vCenter 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**

Make KMS trust vCenter

- 1 Choose a method
- 2 Establish Trust

Choose a method

Choose a method to make the KMS trust the vCenter based on the KMS vendor's requirements. Once the trust is established, all replicas in the same KMS cluster will also trust the vCenter.

- vCenter Root CA Certificate
Download the vCenter root certificate and upload it to the KMS. All certificates signed by this root certificate will be trusted by the KMS.
- vCenter Certificate
Download the vCenter certificate and upload it to the KMS.
- KMS certificate and private key
Upload the KMS certificate and private key to vCenter.
- New Certificate Signing Request (CSR)
Submit the vCenter-generated CSR to the KMS then upload the new KMS-signed certificate to vCenter.

[CANCEL](#) [NEXT](#)

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

Make KMS trust vCenter

- 1 Choose a method
- 2 Upload KMS Credentials

Upload KMS Credentials ✕

Upload the KMS certificate and private key to vCenter to establish the trust.

KMS Certificate UPLOAD A FILE

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIFAPnLM9swDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMC
VVMxFTATBgNVBAoTDEh5VHJ1c3QgSW5lLjExMC8GA1UEAxMoSHIucnVzdCBLZXID
b250cm9sIENlcnRpZmlyYXRlIEF1dGhvcml0eTAeFw0xOTExMTIyMjM4MjFaFw0y
MDExMTIyMjM4MjFaMDUxOzAxBG9NBAYTAIVTMRUwEwYDQKQKwEwleVRydXNOIElu
Yy4xZDZANBgNVBAMTBnZtd2FyZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANFrSuVE14Uxe2ybG4/tKiVwuul6DRQJjxjEztfH08mMLguwp2dpqOac/yNm
4IPVzYWmsJiaa78K8avMDD66rvi6vnxODt+T86VL AdSwXfGidePlaeTfTsnMkTMa
```

KMS Private Key UPLOAD A FILE

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIFAPnLM9swDQYJKoZIhvcNAQELBQAwVzELMAkGA1UEBhMC
VVMxFTATBgNVBAoTDEh5VHJ1c3QgSW5lLjExMC8GA1UEAxMoSHIucnVzdCBLZXID
b250cm9sIENlcnRpZmlyYXRlIEF1dGhvcml0eTAeFw0xOTExMTIyMjM4MjFaFw0y
MDExMTIyMjM4MjFaMDUxOzAxBG9NBAYTAIVTMRUwEwYDQKQKwEwleVRydXNOIElu
Yy4xZDZANBgNVBAMTBnZtd2FyZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBANFrSuVE14Uxe2ybG4/tKiVwuul6DRQJjxjEztfH08mMLguwp2dpqOac/yNm
4IPVzYWmsJjga78K8ayMDD66ryl6vnxODt+T86VLAdSwXfGjqePlaeTfTsnMkTMa
```

CANCEL
BACK
ESTABLISH TRUST

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

vcsa.demo.local
ACTIONS

- Summary
- Monitor
- Configure
- Permissions
- Datcenters
- Hosts & Clusters
- VMs
- Datstores
- Networks
- Linked vCenter Server Systems
- Extensions
- Updates

ADD
ESTABLISH TRUST
ACTIONS

KMS Name	KMS Address	KMS Cluster Name	Port	Connection Status	vCenter Certificate Status	KMS Certificate Status
192.168.109.249	192.168.109.249	KMS (current default)	5696	Connected	Valid until Nov 12, 2020	Valid until Dec 31, 2049

1 servers

Enable vSAN Encryption

Enabling vSAN Encryption is not a difficult task. However, before attempting to enable vSAN Encryption, a few items need to be considered.

1. 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 CPU's.
- 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.

2. What is the state of Deduplication & Compression?

- The process of enabling or disabling vSAN Encryption requires a disk group format change. This is performed in a rolling process through the cluster.
- Changing the state of deduplication & compression also requires a disk group format change, and is also performed in a rolling process through the cluster.
- Enabling/disabling encryption and deduplication & compression independently would require multiple disk group format changes.
- It may be more advantageous to perform encryption and deduplication & compression setting changes simultaneously. This allows for both setting changes to occur during one disk group format change in a rolling process through the cluster.

3. Will Reduced Redundancy be required?

- Because the process of enabling or disabling encryption requires a disk group format change, there must be enough nodes or fault domains for the data being moved off of a disk group 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 disk groups 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 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 disk group.

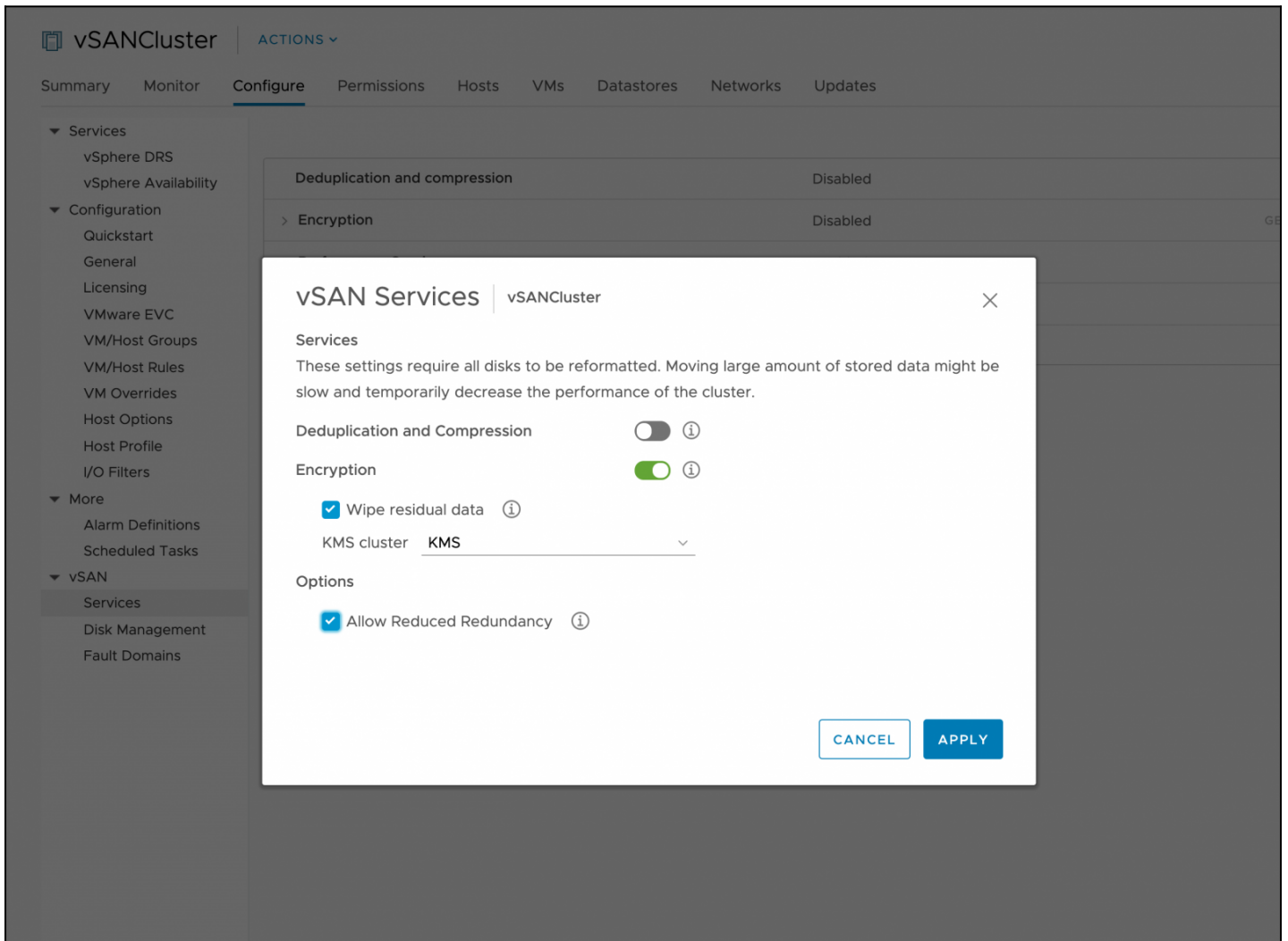
Permissions required to enable vSAN Encryption

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

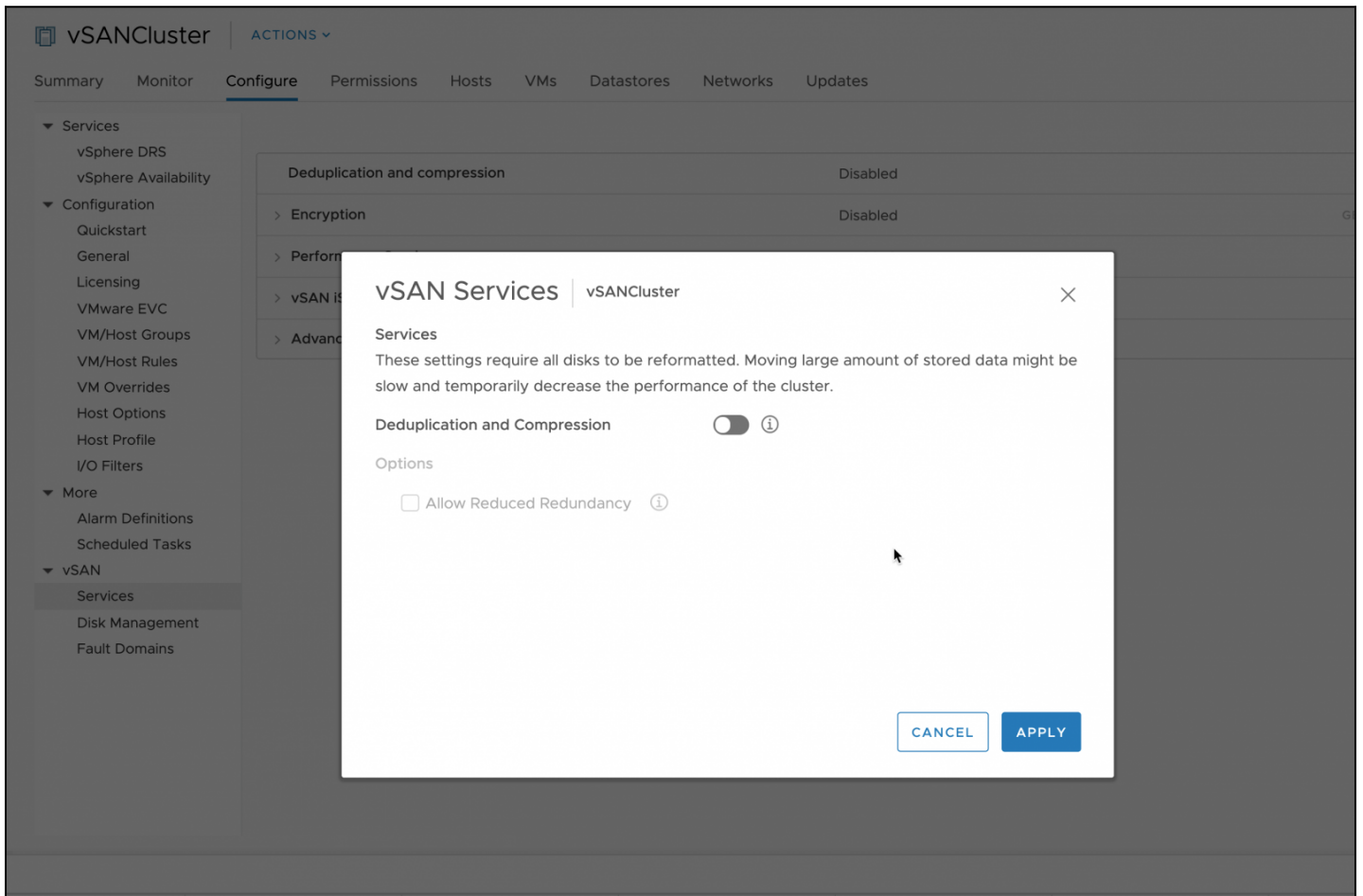
- Host > Inventory > Modify Cluster
- Cryptographic Operations > Manage encryption policies
- Cryptographic Operations > Manage KMS
- Cryptographic Operations > Manage keys

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 disk group or a disk group is created drives will be wiped before being added to the disk group
- 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 group evacuation.

Select **Ok**.

The vSAN Encryption process will begin.

Disable vSAN Encryption

Disabling vSAN Encryption is also not a difficult task. However, before attempting to disable vSAN Encryption, a few items need to be considered.

1. What is the state of Deduplication & Compression?

- The process of enabling or disabling vSAN Encryption requires a disk group format change. This is performed in a rolling process through the cluster.
- Changing the state of deduplication & compression also requires a disk group format change, and is also performed in a rolling process through the cluster.
- Enabling/disabling encryption and deduplication & compression independently would require multiple disk group format changes.
- It may be more advantageous to perform encryption and deduplication & compression setting changes simultaneously. This allows for both setting changes to occur during one disk group format change in a rolling process through the cluster.

2. Will Reduced Redundancy be required?

- Because the process of enabling or disabling encryption requires a disk group format change, there must be enough nodes or fault domains for the data being moved off of a disk group 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 disk groups 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 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 disk group.

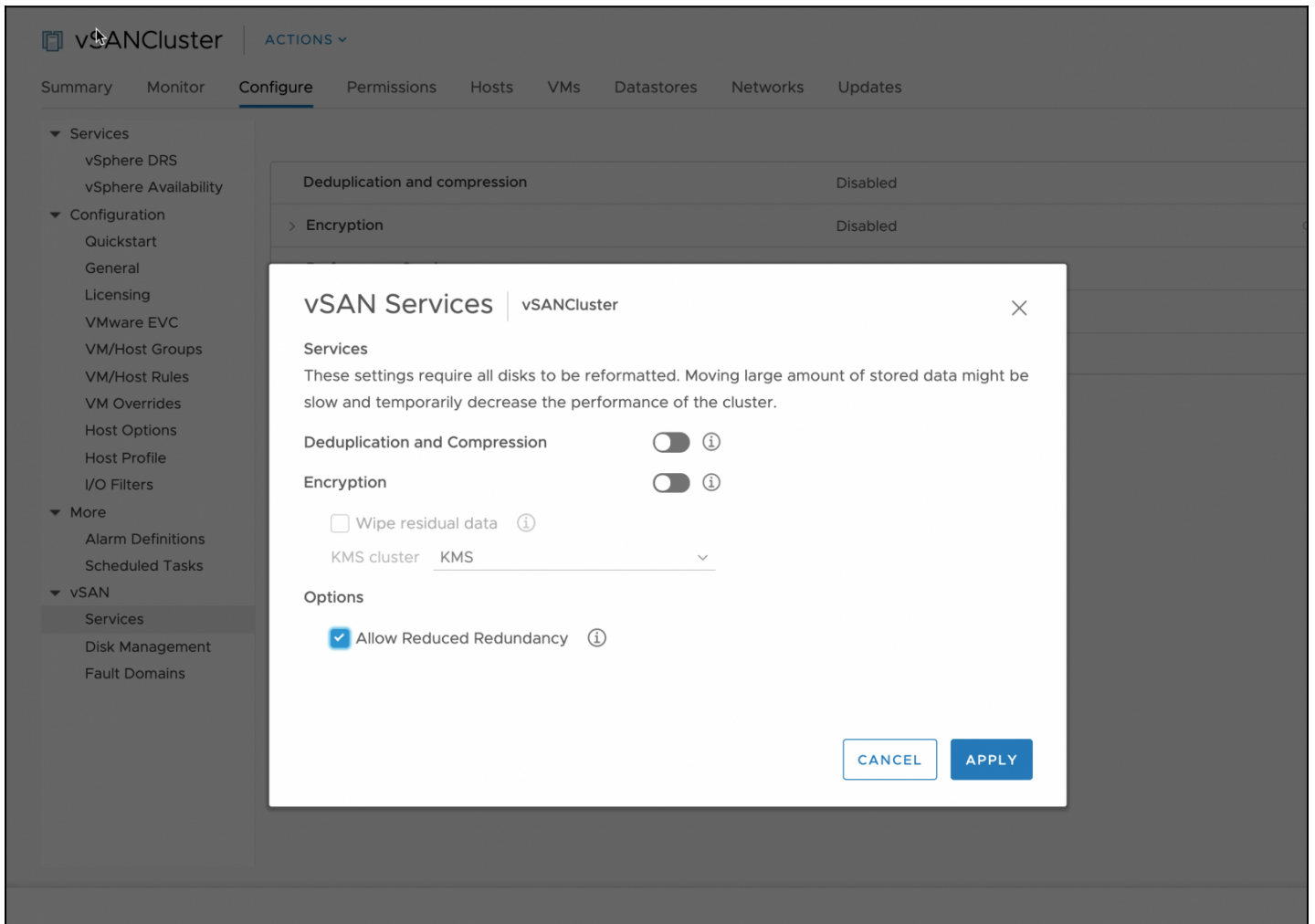
Permissions required to enable vSAN Encryption

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

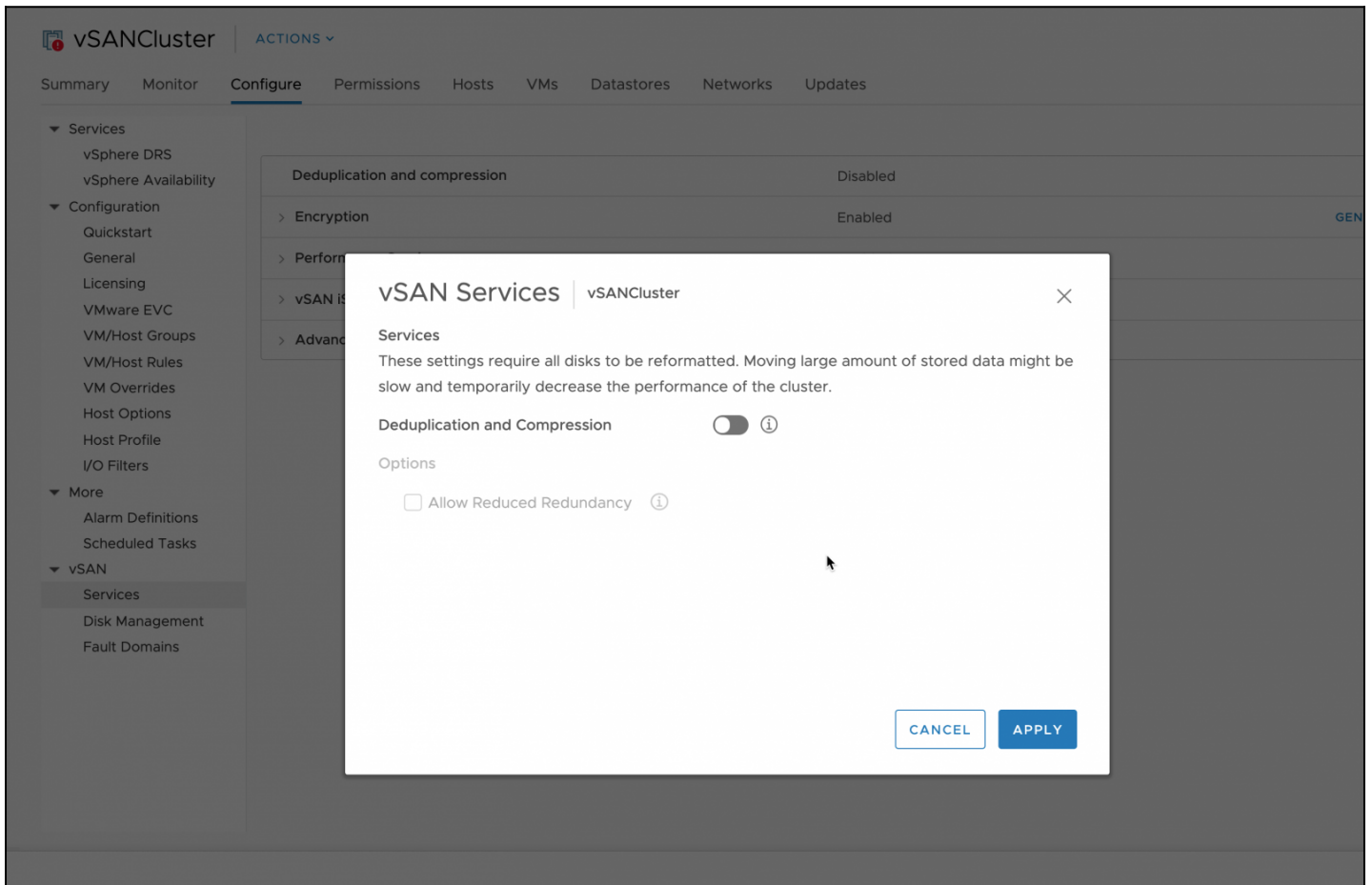
- Host > Inventory > Modify Cluster
- Cryptographic Operations > Manage encryption policies
- Cryptographic Operations > Manage KMS
- Cryptographic Operations > Manage keys

Disabling vSAN Encryption

Select **Cluster** > **Configure** > **General** > **Edit** to disable vSAN



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 disable
 - This will force a disk format change from the current state.
 - Encryption will be disabled 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 group evacuation.

Select **Ok**.

The process of disabling vSAN Encryption will begin.

Day 2 Encryption Operations

Some common workflows that are performed on encrypted vSAN clusters include:

- Shallow Rekey via UI
- Deep Rekey via UI
- Shallow/Deep Rekey via API/PowerCLI
- Changing the KMS Server
- Addition of a non-encrypted host to an encrypted vSAN cluster

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**

The screenshot shows the vSAN configuration interface. The left sidebar lists various services, with 'vSAN' expanded to show 'Services'. The main panel displays a table of services with their status and an 'EDIT' button. The 'Encryption' service is highlighted, and the 'GENERATE NEW ENCRYPTION KEYS' button is visible. A dialog box titled 'Generate New Encryption Keys' is open, showing the following text:

Generate New Encryption Keys

All encryption keys on the key management server cluster are regenerated.

Also re-encrypt all data on the storage using the new keys ⓘ

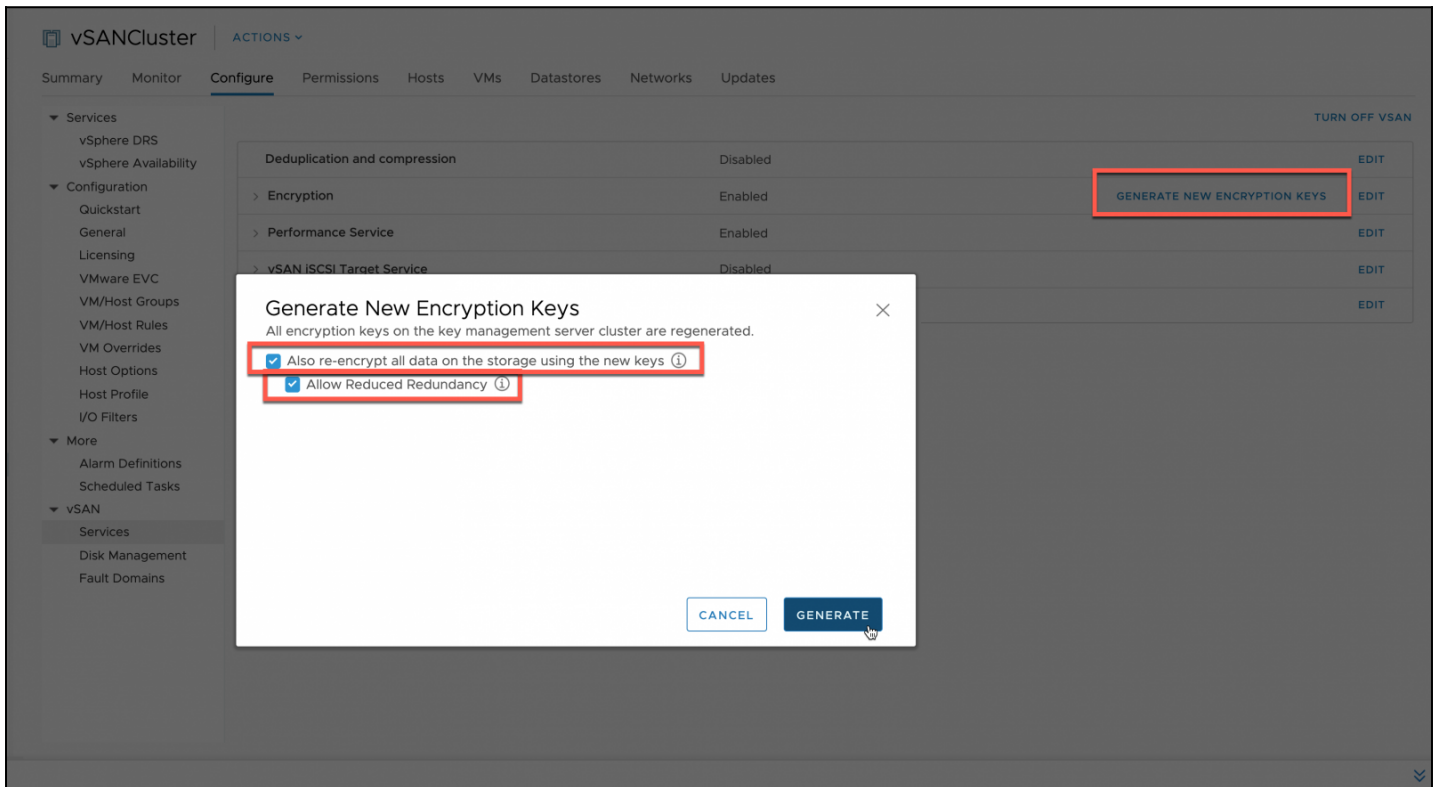
Allow Reduced Redundancy ⓘ

Buttons: CANCEL, GENERATE

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.

Deep Rekey via UI

A Deep Rekey is also a simple process accomplished by the vSphere Client. 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 disk group in the same fashion as the process of enabling encryption.



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.

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      vSAN ISCSI Target Service      State      % Complete Start Time      Finish Time
----      -
Regenerate new keys for encry... Running    0          07:21:51 PM
```

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

Shallow Rekey:

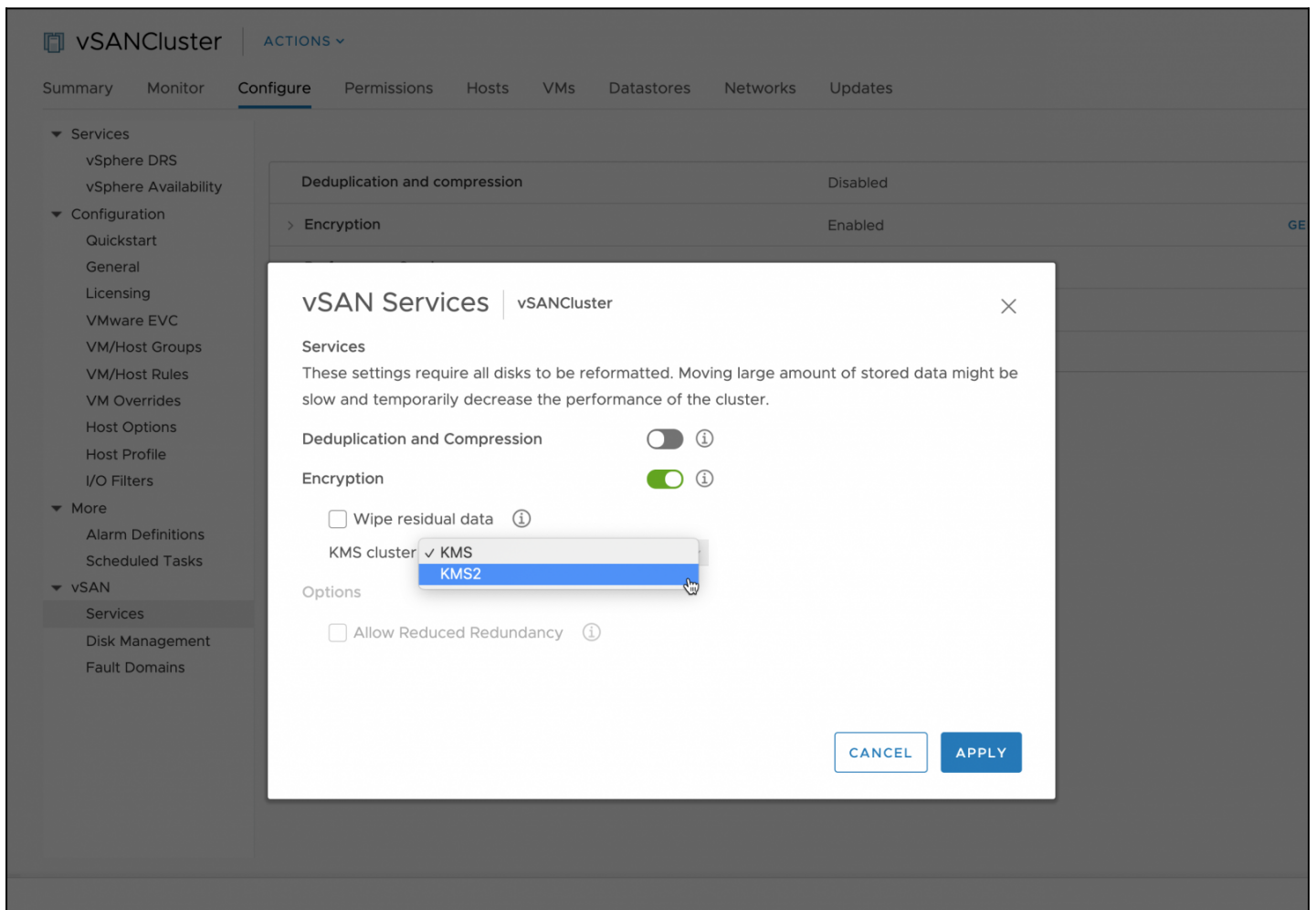
```
Start-VsanEncryption -Cluster "ClusterName" -ShallowRekey -Confirm:$False
(Confirm false to proceed without prompting)
```

Deep Rekey:

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

Changing the 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.

Change the KMS with PowerCLI:

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

Adding a new host to the encrypted cluster

Adding a new host to an existing vSAN Cluster is very easy.

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

- 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.
- Data may be written to the encrypted disk group(s) on the new host.

Guidance when using "Erase disks before use"

A few questions routinely come up when using "Wipe residual data" (*formerly Erase disks before use*) with vSAN Encryption.

vSAN Services | vSANCluster ✕

Services
These settings require all disks to be reformatted. Moving large amount of stored data might be slow and temporarily decrease the performance of the cluster.

Deduplication and Compression ⓘ

Encryption ⓘ

Wipe residual data ⓘ

KMS cluster KMS ▼

Options

Allow Reduced Redundancy ⓘ

What occurs when "Wipe residual data" is used?

When a vSAN cluster has this setting checked, random data is written to each device in a disk group before it is encrypted. Why random data? In one case, if only ones or zeros were written to the device, deduplication and compression could potentially interfere with the effectiveness of the wipe.

This process can be very time-consuming. Different items come into consideration when determining the amount of time necessary to wipe the storage devices. Some of these include the media type, specifically its performance characteristics and its connection protocol (such as NVMe, SAS, NL-SAS, SATA), as well as the capacity, and the number of devices in each host in a cluster.

This wipe process aligns with the [NIST 800-88 Revision 1](#) "Clear" definition:

Clear applies logical techniques to sanitize data in all user-addressable storage locations for protection against simple non-invasive data recovery techniques; typically applied through the standard Read and Write commands to the storage device, such as by rewriting with a new value or using a menu option to reset the device to the factory state (where rewriting is not supported).

Further inspection of [NIST 800-88 Revision 1](#), specifically *Appendix A*, addresses different requirements for different media

types when it comes to a Clear operation. The following table is a summary relevant to vSAN device types:

Media Type	Media Types	Guidance for Clear Operations
ATA Hard Disk Drives	PATA, SATA, eSATA, etc	Overwrite media by using organizationally approved and validated overwriting technologies/methods/tools. The Clear pattern should be at least a single write pass with a fixed data value, such as all zeros. Multiple write passes or more complex values may optionally be used.
SCSI Hard Disk Drives	Parallel SCSI, SAS, FC, UAS, and SCSI Express	Overwrite media by using organizationally approved and validated overwriting technologies/methods/tools. The Clear procedure should consist of at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may optionally be used.
ATA Solid State Drives (SSDs)	PATA, SATA, eSATA, etc	<ol style="list-style-type: none"> Overwrite media by using organizationally approved and tested overwriting technologies/methods/tools. The Clear procedure should consist of at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used. <i>Note:</i> It is important to note that overwrite on flash-based media may significantly reduce the effective lifetime of the media and it may not sanitize the data in unmapped physical media (i.e., the old data may still remain on the media). Use the ATA Security feature set's SECURITY ERASE UNIT command, if supported.
SCSI Solid State Drives (SSSDs)	Parallel SCSI, SAS, FC, UAS, and SCSI Express	<p>Overwrite media by using organizationally approved and tested overwriting technologies/methods/tools. The Clear procedure should consist of at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.</p> <p><i>Note:</i> It is important to note that overwrite on flash-based media may significantly reduce the effective lifetime of the media and it may not sanitize the data in unmapped physical media (i.e., the old data may still remain on the media).</p>
NVM Express SSDs	NVMe devices	Overwrite media by using organizationally approved and tested overwriting technologies/methods/tools. The Clear procedure should consist of at least one pass of writes with a fixed data value, such as all zeros. Multiple passes or more complex values may alternatively be used.

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.

When should “Wipe residual data” be used?

Choosing when to perform a wipe operation is as important to know as the wiping process itself. Should “Erase disks before use” regularly at some interval? Or is it just when first enabling vSAN Encryption on a vSAN cluster?

Some valid use cases for this wiping before encryption include:

- Enabling encryption if a cluster already has data on it
- Adding hosts or disks that already had non-encrypted clear text data on them to an encrypted vSAN cluster

It is important to consider that *only new data is encrypted* when vSAN Encryption is enabled. This wiping process ensures there is no residual data on a storage device used by vSAN.

What happens if “Wipe residual data” isn’t used on an existing cluster or added host?

The process of enabling vSAN Encryption only encrypts new data.

Whether it is an existing cluster, or simply an existing host being added to a vSAN cluster, *any residual data could potentially still be recovered.*

Recommendations

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 when vSAN Encryption is enabled

In order to recover from this and similar scenarios, it is 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.

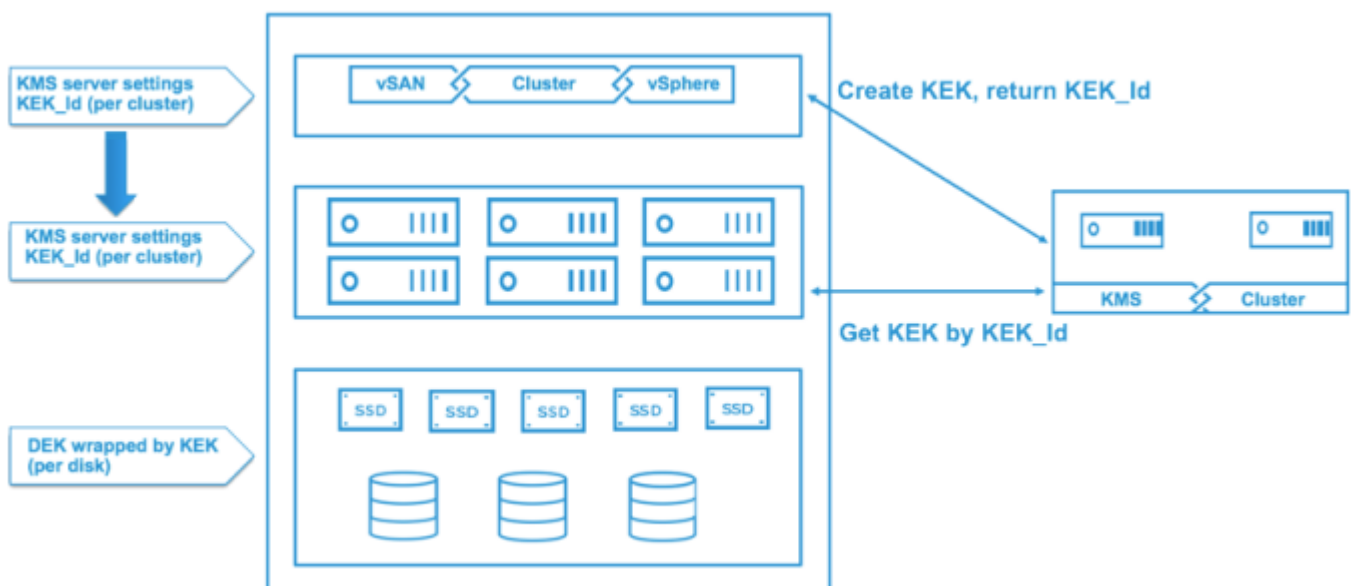
It is imperative that the same KMS cluster ID remains in order for the recovery feature to work.

Although the old vCenter is gone, 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.

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, how do we recreate these items in a newly deployed vCenter?

In this diagram we can see how the keys are distributed to vCenter, hosts, etc. The KMS server settings are passed to hosts from vCenter by the KEK_id.



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[0001]/old =
/vsan/kmipServer/child[0001]/port =
/vsan/kmipServer/child[0001]/address =
/vsan/kmipServer/child[0001]/kmskey =
/vsan/kmipServer/child[0001]/userName =
/vsan/kmipServer/child[0001]/name =
/vsan/kmipServer/child[0001]/kmipClusterId =
```

After the KMS cluster has been added to new vCenter as it was configured in the previous vCenter, 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.

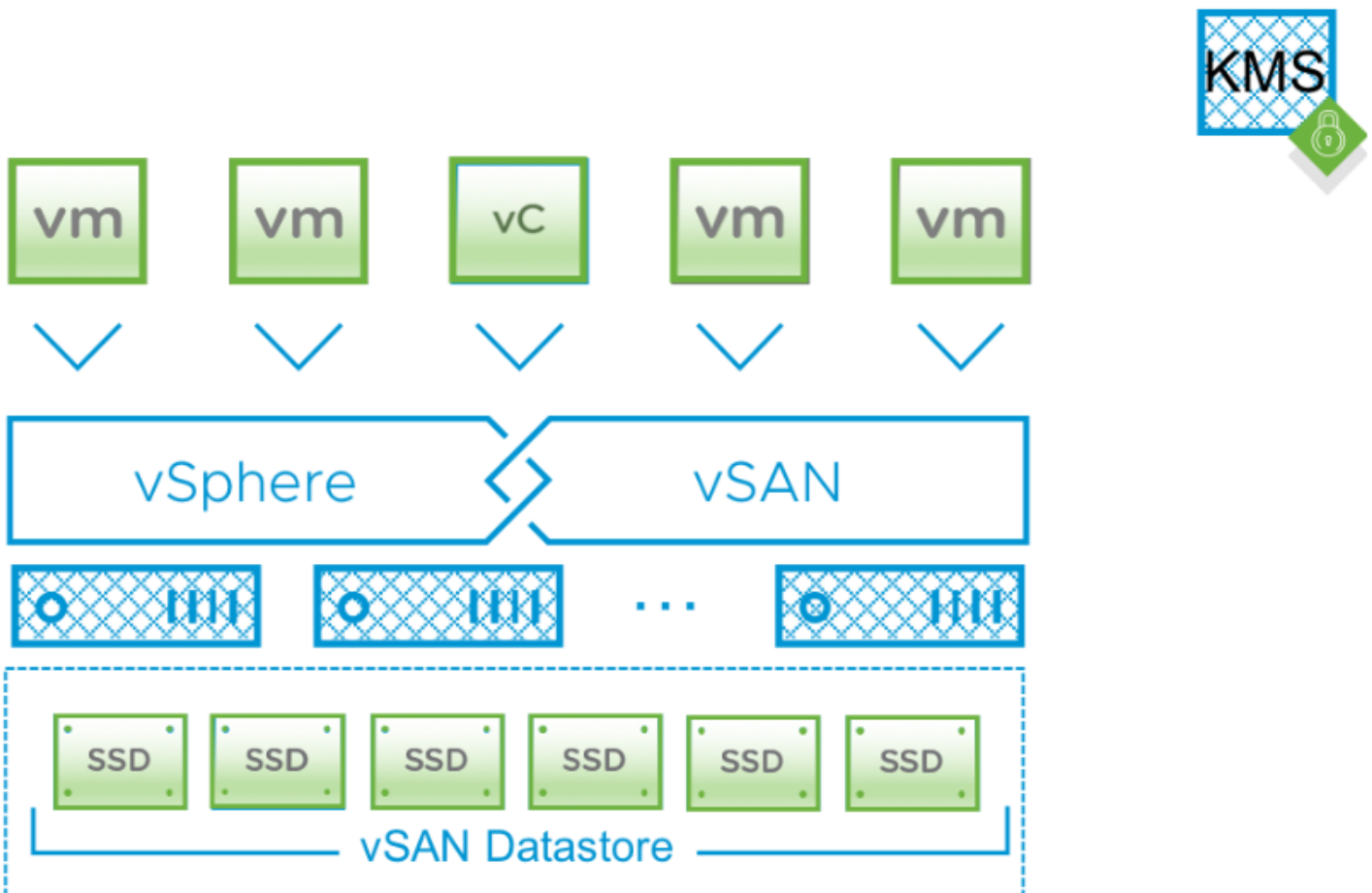
vSAN Encryption Troubleshooting

KMS Server Accessibility

vSAN 6.6 introduced data at rest encryption as a new feature that provides another choice (in addition to VM Encryption introduced in vSphere 6.5) for customers to secure data in vSphere. Despite the fact that these technologies work a bit differently (per datastore for vSAN Encryption or per VM for VM Encryption) these technologies still use a common Cryptographic Library to perform their work. Key Management is also common among these two technologies.

Additional relevant content can be found at: [Key Manager Concepts and Topology Basics for VM and vSAN Encryption](#)

Specific to vSAN though, it is important to keep the KMS external to the vSAN datastore it is providing key management for.



If the KMS resides on the datastore it is providing key management for, a circular dependency can occur.

Hosts in a vSAN cluster that has vSAN Encryption enabled will directly contact the KMS they are assigned to upon boot up to unlock/mount disk groups.

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 disk groups.
5. The KMS appliance's disks are still not available and will not be.

It is important to remember that a KMS appliance should not be stored on the vSAN datastore that it is providing keys for. This is not a supported configuration.

We have some sample PowerCLI code 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/>

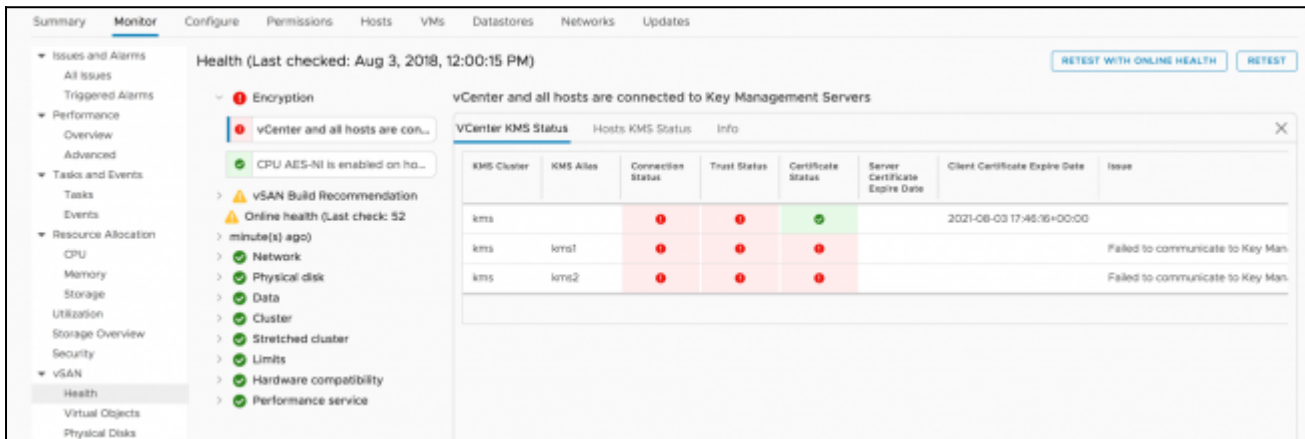
KMS Profile Addressing

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.

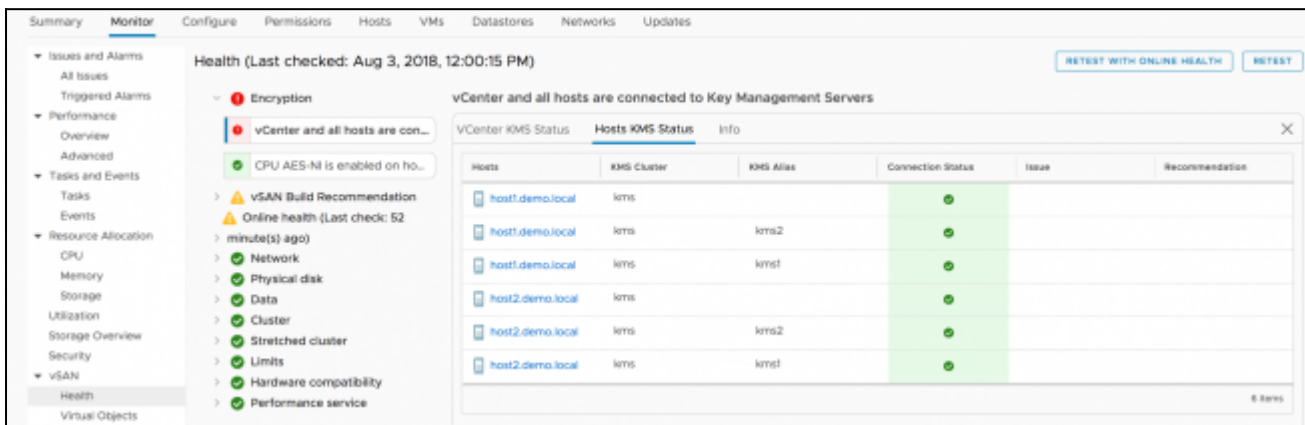
Troubleshooting indicated that there were no blocked ports between the vCenter Server and the KMS Cluster as well as they were able to properly ping each other. vSAN Hosts could properly ping the KMS Cluster as well, and no ports were blocked.

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.

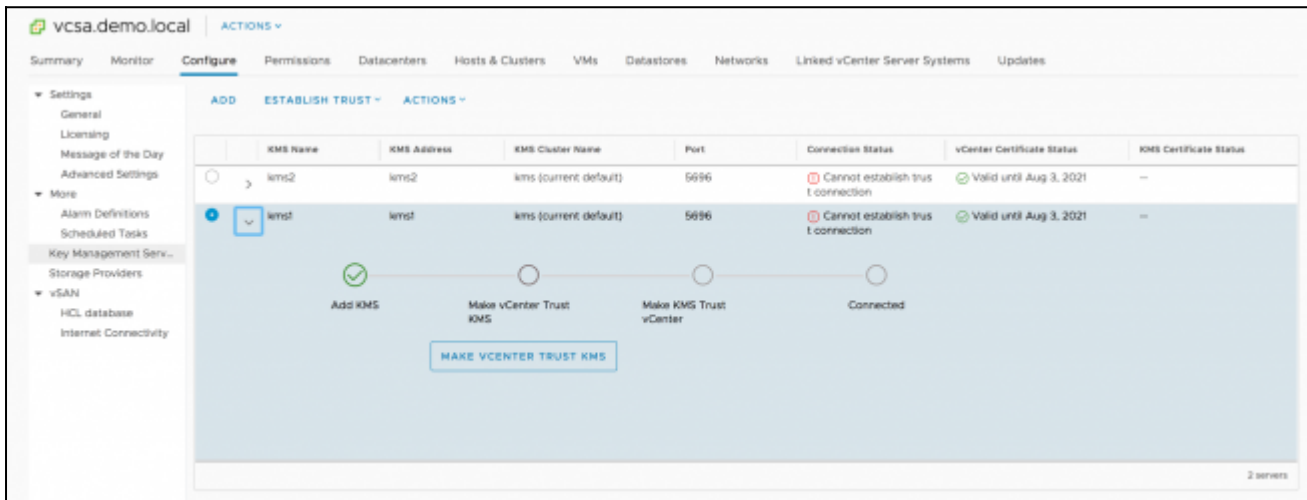
The connection has to be correct in vCenter Server before it can be correct/pushed to vSAN Hosts. Something must have changed in the environment to cause this issue.

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.

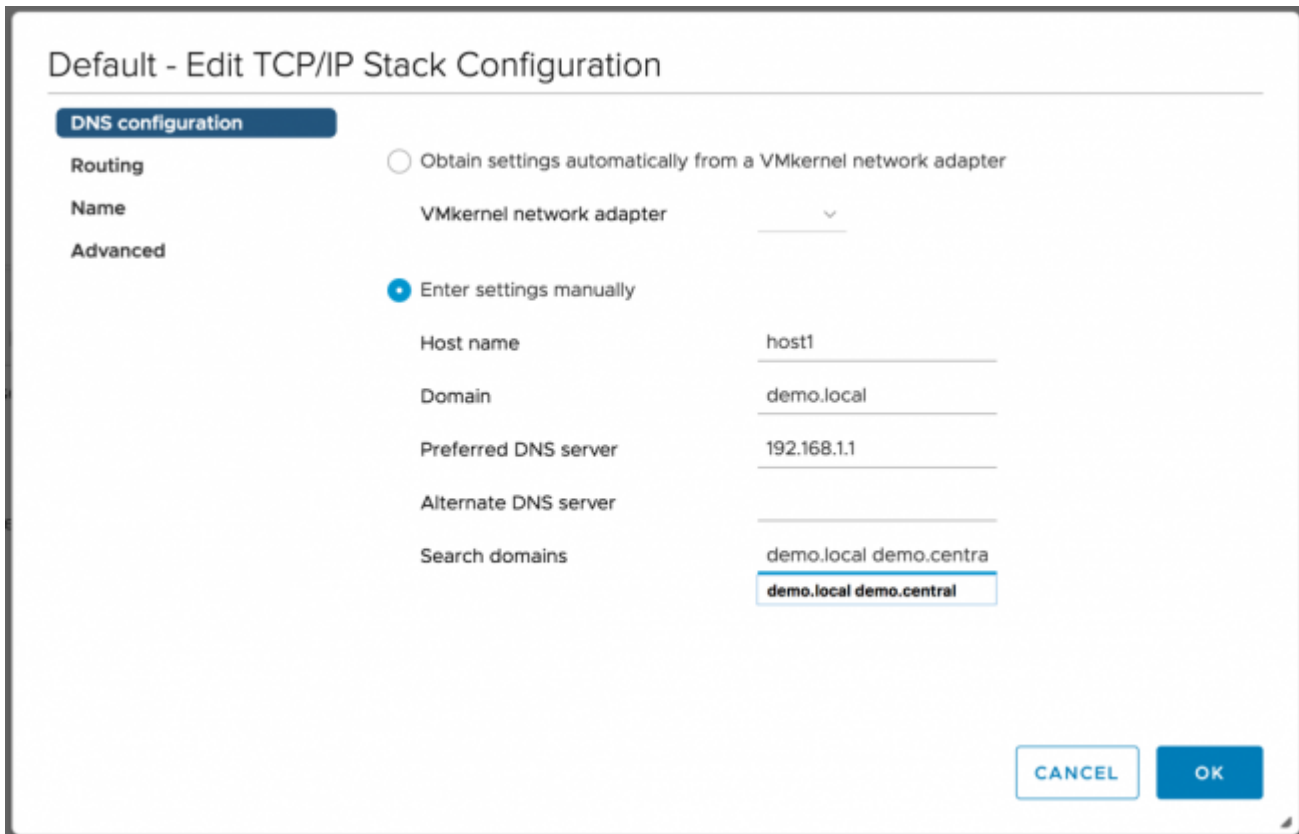
Careful review of the **vCenter KMS Status** and **Host KMS Status** health checks, the **KMS Alias** is a “short name”.

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.

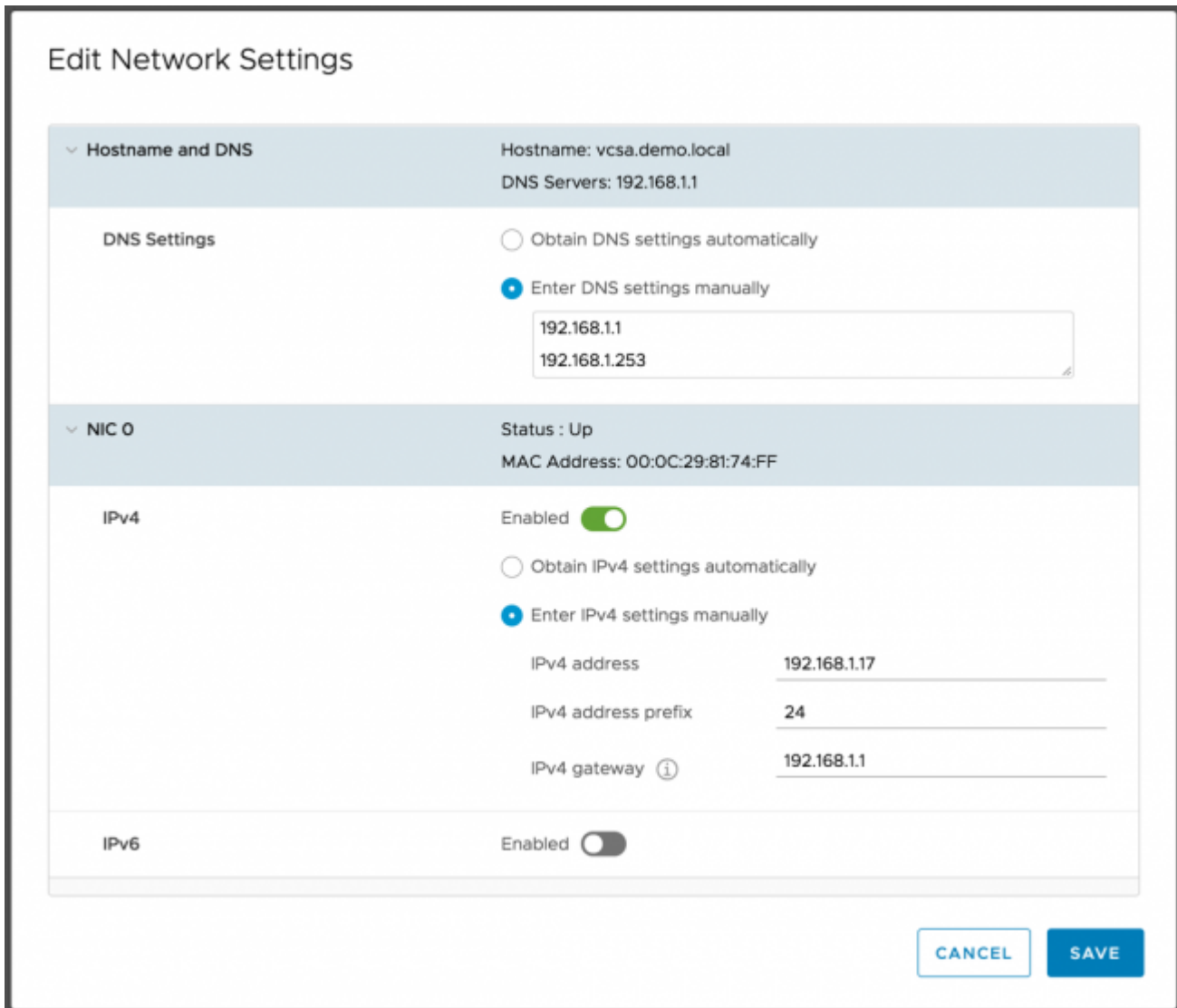
The **Key Management Servers** configuration Profile in the vCenter’s settings shows that the trust cannot be established. The **KMS Address** is the same value as the **KMS Alias** in the vSAN Health Check.



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.

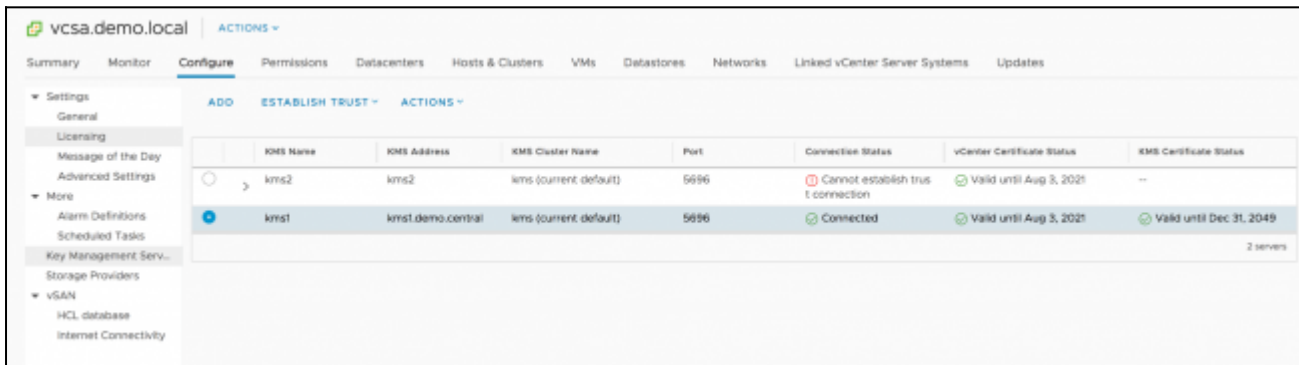


The VCSA, on the other hand, does not have any search domains:



Without search domains to assist with the short name, vCenter 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.

KMS Name	KMS Address	KMS Cluster Name	Port	Connection Status	vCenter Certificate Status	KMS Certificate Status
<input type="radio"/> kms2	kms2.demo.central	kms (current default)	5696	Connected	Valid until Aug 3, 2021	Valid until Dec 31, 2049
<input type="radio"/> kms1	kms1.demo.central	kms (current default)	5696	Connected	Valid until Aug 3, 2021	Valid until Dec 31, 2049

In the case that this was brought up, an alternate vCenter 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 short, 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 is Unavailable

Since the introduction of vSAN Encryption in vSAN 6.6 some of the routine questions that get asked include:

- Do vSAN hosts get encryption key information from vCenter?
- What if vCenter is offline, is vSAN encryption impacted?
- What if vCenter is removed/replaced/not available, is vSAN impacted?

The answers to these are pretty straightforward and easy to answer. To better understand the answers to those questions, it is important to understand the requirements and workflow of setting up vSAN Encryption.

Key Management Solution (KMS)

vSAN Encryption requires a KMS that is compliant with the KMIP 1.1 protocol. As of today, 9 different vendors offering 21 different solutions have been certified for use with vSAN Encryption. Those can be found [here](#). Other KMS'es that are compliant with the KMIP 1.1 protocol are supported though.

The KMS must have connectivity with the vCenter Server and each of the vSAN hosts that will have encryption enabled.

vCenter

vCenter Server must be a version that is supported for use with the vSAN version. vSAN Encryption is available in vSAN 6.6 and vSAN 6.7. Refer to the [VMware Product Interoperability Matrices](#) to ensure that the vCenter and vSAN versions are compatible.

The creation of a Key Management Server profile in vCenter Server creates connection information used by vSAN hosts. As part of the profile creation process, a trust is established between the vCenter Server and the Key Management Solution. The Key Management Server profile allows vSAN Encryption the ability to use keys from the Key Management Solution.

When vSAN Encryption is enabled, vCenter requests two keys from the Key Management Solution for use by the encrypted vSAN Cluster. The Key Encryption Key (KEK) is used to encrypt each Data Encryption Key (DEK) on each vSAN storage device, and the Host Key is used by each host to encrypt core dumps.

vSAN Hosts

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. 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.

Here are some of those values found in `/etc/vmware/esx.conf` for vSAN Encryption:

```

/vmkdevmgr/logical/pcim02000b000/alias = "vmnic1"
/vmkdevmgr/logical/pcim010003000/alias = "vmnic0"
/vmkdevmgr/logical/pcim010013000/alias = "vmhba0"
/vsan/enabled = "true"
/vsan/hostDecommissionVersion = "0"
/vsan/faultDomainName = ""
/vsan/autoClaimStorage = "false"
/vsan/hostDecommissionState = "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"
/vsan/network/child[0000]/ifaceUuid = "29e6445b-10be-3f02-c57e-005056b0a877"
/vsan/network/child[0000]/vmknics = "vmk0"
    
```

KMS Profile information

Host Key Id

KEK Id

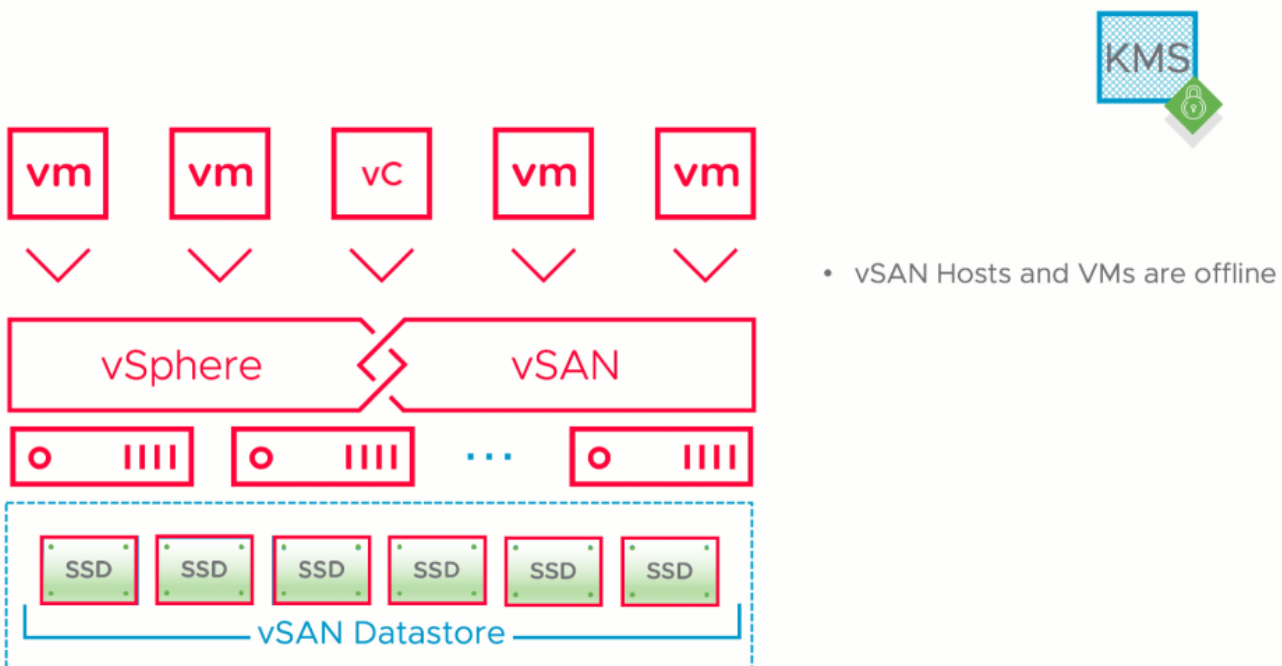
When vSAN Encryption is enabled, or when a deep rekey operation is invoked, the vSAN host creates a unique DEK (XTS-AES-256) for each device, and it is encrypted with the KEK. A shallow rekey operation swaps out the KEK and rewraps each DEK.

When a host with vSAN Encryption enabled attempts to mount a vSAN Disk Group, the DEK is unwrapped using the KEK, allowing vSAN to mount and then use the vSAN Disk Group.

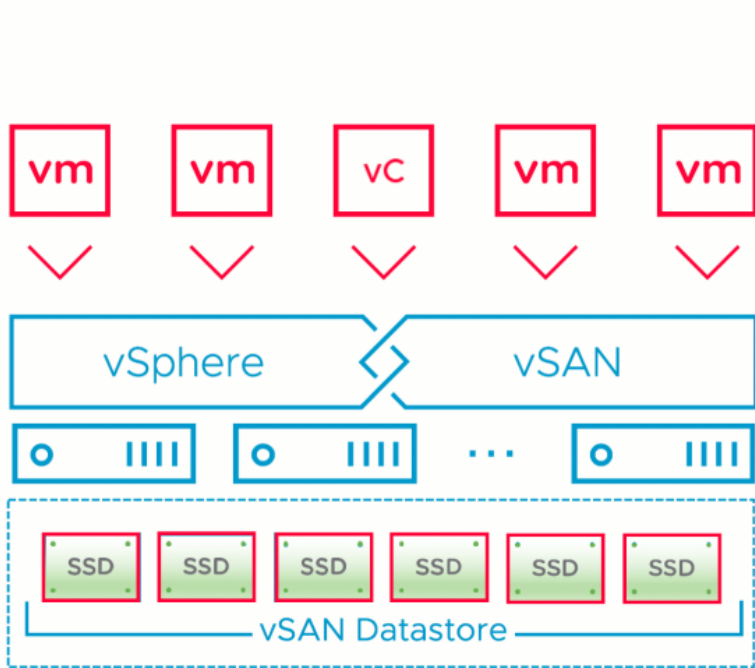
The Boot Process

So what happens if a vSAN Cluster is completely offline? How does the boot process work, and how are VMs brought back online when vSAN Encryption is in place?

Notice in this illustration that hosts are powered off as well as all of the virtual machines, including vCenter. The KMS Server is **online** and external to the vSAN Cluster.

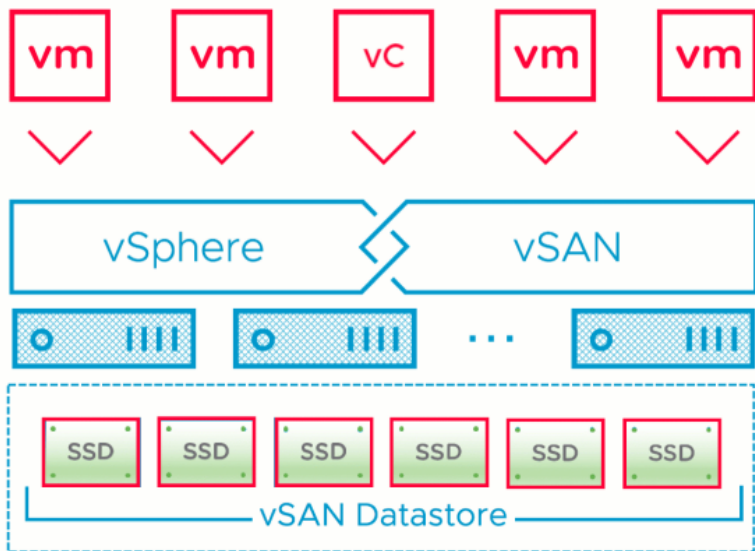


vSAN hosts are then booted. Disks are not immediately mounted, and VMs are offline (including vCenter).



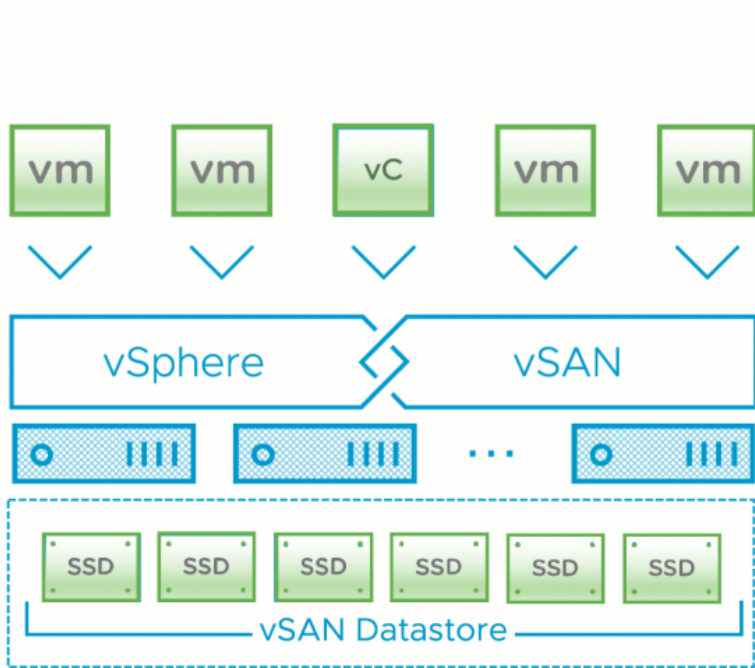
- vSAN Hosts and VMs are offline
- vSAN Hosts are booted

After a vSAN host boots, it will read the values in `/etc/vmware/esx.conf` and request the KEK and Host Key from the KMS using the KEK Id and Host Key Id respectively, directly from the KMS.



- vSAN Hosts and VMs are offline
- vSAN Hosts are booted
- vSAN Hosts retrieve KEK information
 - Directly from KMS

The KEK and Host Key are placed in memory in the key cache. These keys are not persistently stored on the vSAN hosts. The KEK is then used to mount the vSAN Disk Groups and VMs may be powered on.



- vSAN Hosts and VMs are offline
- vSAN Hosts are booted
- vSAN Hosts retrieve KEK information
 - Directly from KMS
- vSAN Hosts mount disk groups
 - Using KEK from KMS
- VMs can be powered on

If the KMS were residing on top of vSAN, a circular dependency would occur. This is because the KMS is needed to provide the KEK to the vSAN hosts. This is covered in the **Understanding vSAN Encryption - KMS Server Accessibility** post here: <https://blogs.vmware.com/virtualblocks/2018/05/17/vsan-enc-kms-accessibility/>

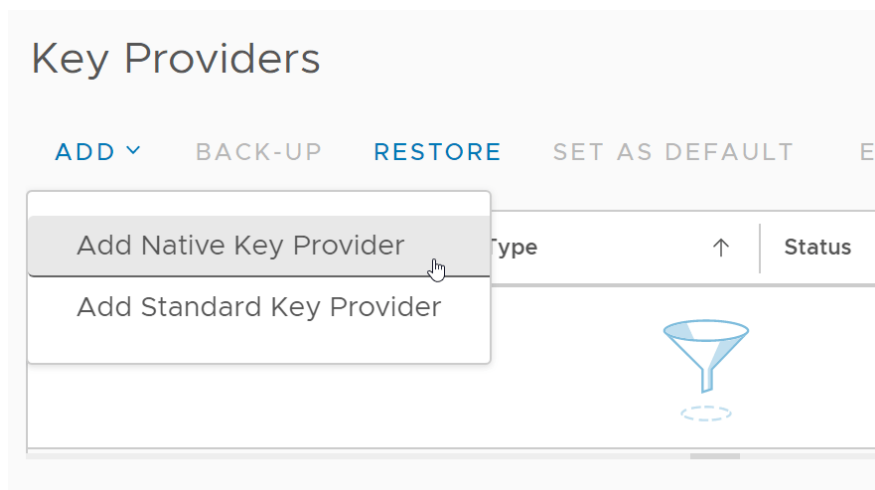
As long as vSAN hosts using vSAN Encryption have connectivity to their configured KMS, they have no issue booting, even when vCenter is offline. The boot process is not dependent on vCenter to unlock and mount vSAN Disk Groups.

Native Key Provider for VMware vSAN

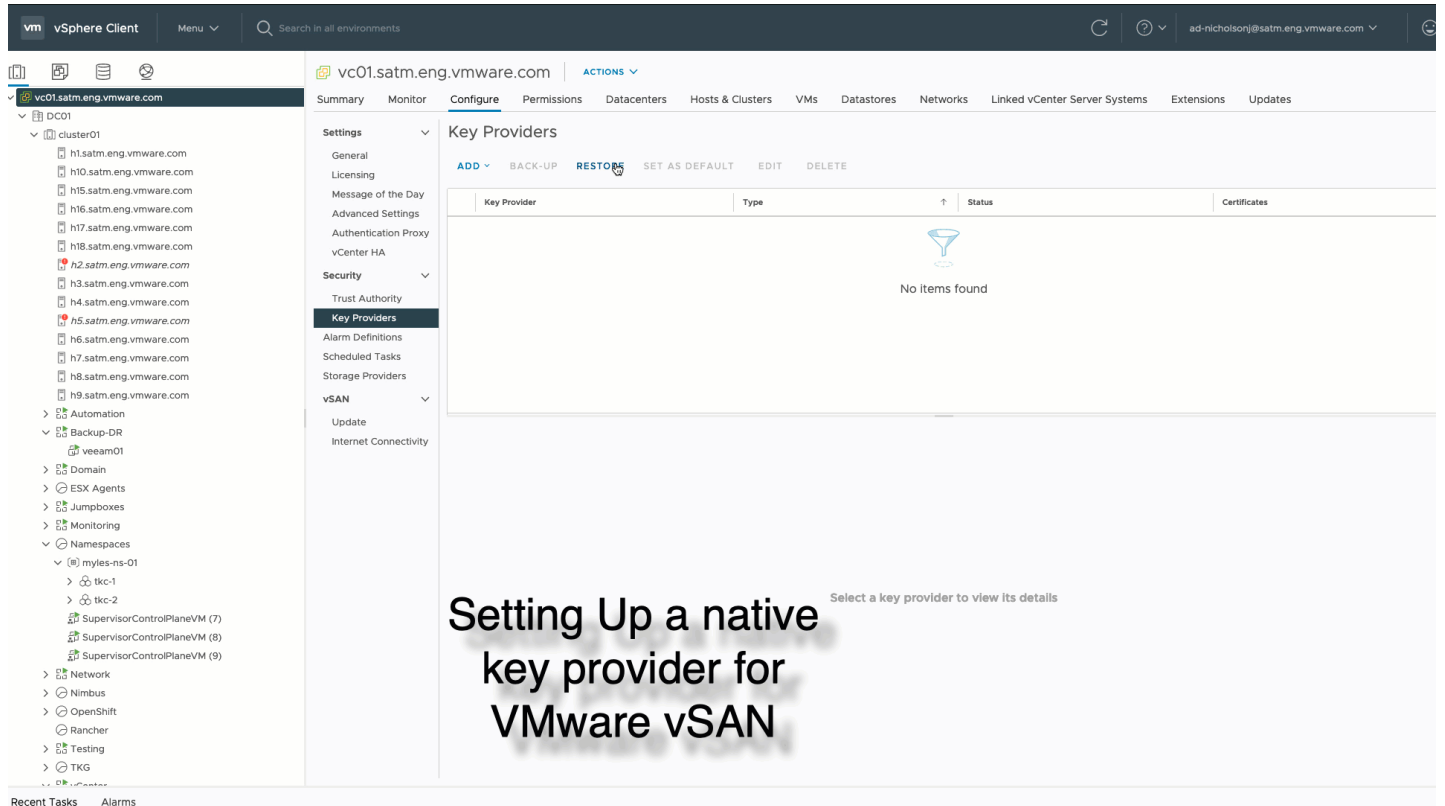
vSphere 7 Update 2 introduced the vSphere Native Key Provider, a mechanism to enable vTPM, VM Encryption, and vSAN Encryption that exists completely within vSphere itself. It is driven by vCenter Server and clustered ESXi hosts and, to vSphere, enables nearly the same functionality as with a traditional Key Management Service (KMS). With this, customers of all sizes have better access to encryption technologies. Additional information can be found at the [native key provider homepage](#).

Native Key Provider Operations

Creating a Native Key Provider



A new native key provider can be configured in a few simple clicks.



Configuring vSAN to use a configured Native Key Provider

vSAN Services | vSAN-Cluster
×

Space efficiency None ▼ ⓘ

Data-At-Rest encryption ⓘ

Wipe residual data ⓘ

Key provider DMZ-KeyProvider ▼

Allow reduced redundancy ⓘ

Data-In-Transit encryption ⓘ

Data-In-Transit encryption is not supported if the cluster mounts or exports a remote datastore.

Rekey interval DEFAULT ▼ 1 day ▼

CANCEL
APPLY

Native Key Provider Design Considerations

- Native Key Provider is not a full featured KMS. It can not be leveraged for other KIMP needs.
- vSAN Encryption requires vSAN Enterprise, or Enterprise Plus licensing.

Backing up native key provider keys

A manual backup of the key provider should be performed from within the vCenter UI. Do note that when you configure the Native Key Provider and want to back it up, you need to access the vSphere UI via the fully qualified domain name. vSphere Native Key Provider is backed up as part of the vCenter Server file-based backup. However, you must back up the vSphere Native Key Provider at least once before you can use it. When you create a vSphere Native Key Provider, it is not backed up.

Back up Native Key Provider | DMZ-KeyProvider



Protect Native Key Provider data with password (recommended)

Password

.....



COPY PASSWORD

Verify password

.....



I have saved the password in a secure place.

Make sure this password is securely saved, as it will be required to restore the Native Key Provider configuration in case of disaster. Without this password access to resources such as encrypted VMs and VMs with virtual TPM devices will be lost.

CANCEL

BACK UP KEY PROVIDER

Enforce usage of vSAN Encryption

Additional SPBM storage rules can specify that a virtual machine or disk being created will be placed on a Data-At-Rest encrypted cluster.

Create VM Storage Policy

- 1 Name and description
- 2 Policy structure
- 3 vSAN
- 4 Storage compatibility
- 5 Review and finish

X

vSAN

Availability
Storage rules
Advanced Policy Rules
Tags

Encryption services (i)

Data-At-Rest encryption

No encryption

No preference

Space efficiency (i)

Deduplication and compression

Compression only

No space efficiency

No preference

Storage tier (i)

All flash

Hybrid

No preference

CANCEL BACK NEXT

Additional References

VMware Docs

- VMware VM Encryption and vSAN Encryption FAQ
<https://vmware.com/go/encryptionfaq>
- Using Encryption on a vSAN Cluster
<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.virtualsan.doc/GUID-F3B2714F-3406-48E7-AC2D-3>

677355C94D3.html

- How vSAN Encryption Works
<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.virtualsan.doc/GUID-37F9636A-7481-4486-AAA9-E0A1A49343A1.html>
- Enabling Encryption on an existing vSAN Cluster
<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.virtualsan.doc/GUID-E7CA36B7-D7EB-423A-ADD1-7E410E36F5A7.html>
- vSAN Encryption and Core Dumps
<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.virtualsan.doc/GUID-6701FDE9-D1BA-4455-BD9F-3519646D408C.html>
- Design Considerations for vSAN Encryption
<https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.virtualsan.doc/GUID-1DFF5BC8-E9B5-4460-BEBD-3BB76E46B35F.html>



**VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax
650-427-5001 www.vmware.com**

Copyright © 2021 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.