

Managing Oracle Database Encryption Keys in Oracle Cloud Infrastructure with Oracle Key Vault

Securing Oracle Advanced Security TDE Keys in Oracle Key Vault

ORACLE WHITE PAPER | MAY 2018





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Note: The white paper is subject to further revisions. Please verify that you have the latest version.

Revision History

The following revisions have been made to this white paper since its initial publication:

Date	Revision
May 25, 2018	<ul style="list-style-type: none"><li data-bbox="418 932 1281 989">• Added operational best practices for managing Oracle Key Vault in an Oracle Cloud Infrastructure environment<li data-bbox="418 1003 683 1031">• Deleted FAQs section

You can find the most recent versions of the Oracle Cloud Infrastructure white papers at <https://cloud.oracle.com/iaas/technical-resources>.



Table of Contents

Introduction	5
Oracle Key Vault Overview	6
Central Management of Oracle Wallets and Java KeyStores	6
Online TDE Master Key Management	7
Central Backup of Credential Files	8
Installing Oracle Key Vault in Oracle Cloud Infrastructure	8
Obtain the Oracle Key Vault Image and License	8
Install a BYOH KVM and Configure VCN Networking for the Oracle Key Vault VM	9
Install the Oracle Key Vault VM on the Bare Metal Instance	10
Configuring Oracle TDE Keys with Oracle Key Vault in Oracle Cloud Infrastructure	11
Enroll a Database Endpoint in Oracle Key Vault	12
Upload Oracle Wallet Keys to Oracle Key Vault	13
Migrate TDE Master from Oracle Wallet to Oracle Key Vault	13
Oracle Key Vault Best Practices	14
Prototype with Non-Critical Database Workloads	14
Secure Key Backups	15
Configure for High Availability	15
Enable Oracle Key Vault SSH Access	15
Use Oracle Key Vault Audit Logs	16
Use VCN Security Lists to Firewall Oracle Key Vault Instances	16
Manage Oracle Key Vault in the Oracle Cloud Infrastructure Environment	16
Configure Oracle Key Vault Active Data Guard	17
Configure Oracle Key Vault RAC	17
Configure Oracle Key Vault GoldenGate	17



Conclusion	17
Appendix	18
Enable SR-IOV on a Bare Metal instance	18
Enable VFs and Configure with MAC Address of Secondary VNIC	18
Create Network Interface Using the VLAN Tag of the Secondary VNIC	18
attach.xml file	19
rest.ini file	19
enroll_okv_endpoint file	19



Introduction


Oracle Database implements encryption of data at rest by using Transparent Data Encryption (TDE), which is available as part of Oracle Advanced Security. TDE transparently encrypts the database to higher-level applications that use the data, and the encryption can be implemented for the entire tablespace or specific columns in the tables.

TDE uses a two-tier encryption key architecture that consists of *database encryption keys* used for tablespace or column encryption, and a *TDE master key* used to wrap (encrypt) the database encryption keys. The wrapped database tablespace and column encryption keys are stored natively in the database, and the TDE master key is usually stored in an Oracle Wallet in the local filesystem or in the Automatic Storage Management (ASM) disk group for clustered access. Other recommended options for storing TDE master keys include a centralized key management platform such as Oracle Key Vault or a hardware security module (HSM).

The TDE master key is stored in an Oracle Wallet, in a PKCS#12 formatted file, and secured using a customer-provided password. When the database encryption keys are required to encrypt or decrypt the database, the customer opens the wallet by providing the correct password, and the wallet-based TDE master key is used to unwrap the database encryption keys, which are subsequently used by the database for encrypting or decrypting the database tables or columns. For lights-out operational requirements, the Oracle Wallet can also be provisioned without a password by using the auto-login option.

Unlike virtual machines (VMs), Oracle Cloud Infrastructure Compute bare metal instances are not managed by a higher-privileged Oracle Cloud Infrastructure-controlled hypervisor, and it is technically infeasible for any Oracle Cloud Infrastructure operators to access data stored in the memory or local disks of a bare metal instance. This enables customers to have complete control over any data stored on bare metal instances. Oracle Database offerings to Oracle Cloud Infrastructure customers leverage bare metal instances. Oracle Database options in Oracle Cloud Infrastructure are as follows:

- **Database instance:** Elastic and on-demand Oracle Databases on the bare metal instance with NVMe local flash storage. The following shapes of database instances are offered:
 - **HighIO:** 36 cores, 512 GB RAM, and 12.8 TB NVMe storage
 - **DenseIO:** 36 cores, 512 GB RAM, and 28.8 TB NVMe storage
 - **2 node RAC**
 - **Exadata:** Quarter rack, half rack, and full rack
- **Bring your own license (BYOL) database:** Customers can install an Oracle Database on their bare metal instance.



In Database and BYOL bare metal instances, TDE master keys are normally provisioned in Oracle Wallets on bare metal instances owned by customers, thereby allowing customers complete control over their TDE keys (that is, Oracle Cloud Infrastructure operators cannot access Oracle Wallets on bare metal instances). In this case, Oracle Cloud Infrastructure customers are responsible for individually managing (rotation, backup for disaster recovery, availability) the TDE master keys in Oracle Wallets on all the database instances they own. The resulting key management could impose non-trivial operational effort for some customers with large-scale database deployments.

Oracle Key Vault offers a solution for reducing the operational effort associated with TDE key management. Oracle Key Vault is a security-hardened software appliance used to store and manage TDE master keys of multiple Oracle databases and other security applications, including MySQL TDE, Solaris Crypto, and ASM Cluster File System (ACFS) encryption. Customers can install Oracle Key Vault on their bare metal instances and register all their Oracle databases in Oracle Cloud Infrastructure as endpoints, in order to manage all the TDE master keys in a centralized manner. By installing Oracle Key Vault on their bare metal instances, Oracle Cloud Infrastructure customers can continue to maintain control over their TDE master keys while relying on Oracle Key Vault functionality to ease the operational effort associated with key management.

This white paper provides instructions for installing and configuring Oracle Key Vault on customer-owned bare metal instances in their virtual cloud network (VCN), to manage their Oracle database TDE keys.

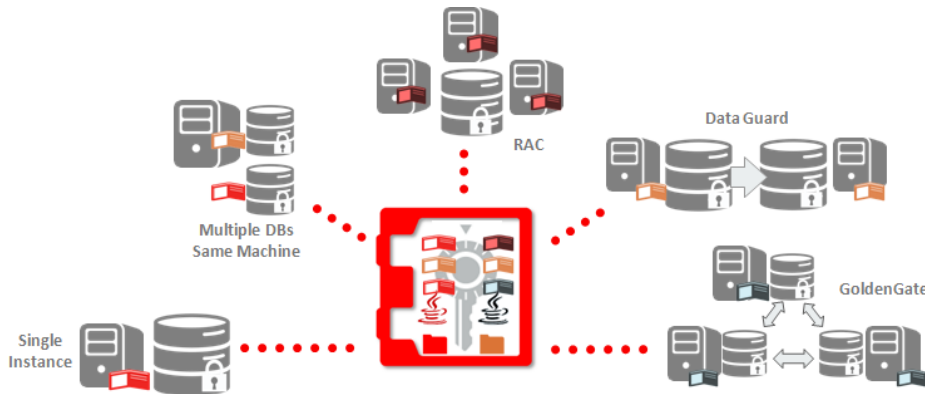
Oracle Key Vault Overview

Oracle Key Vault enables customers to quickly deploy encryption and other security solutions by centrally managing encryption keys, Oracle Wallets, Java KeyStores, and credential files. It is optimized for managing Oracle Advanced Security Transparent Data Encryption (TDE) master keys. The full-stack, security-hardened software appliance uses Oracle Linux and Oracle Database technology for security, availability, and scalability. Oracle Key Vault supports the OASIS Key Management Interoperability Protocol (KMIP) industry standard.

Central Management of Oracle Wallets and Java KeyStores

Oracle Wallets and Java KeyStores are often distributed across servers and server clusters manually. Oracle Key Vault itemizes and stores that contents of these files in a master repository while allowing server endpoints to continue operating disconnected from Oracle Key Vault using their local copies. Once archived, wallets and keystores can be recovered back to servers if their local copies are mistakenly deleted or their passwords are forgotten.

Oracle Key Vault streamlines the sharing of wallets across database clusters such as Oracle RAC, Oracle Active Data Guard, and Oracle GoldenGate. Secure sharing of wallets also facilitates movement of encrypted data by using Oracle Data Pump and Oracle Transportable Tablespaces. Oracle Key Vault can be used with Oracle Wallets from all supported releases of Oracle Middleware and Oracle Database.

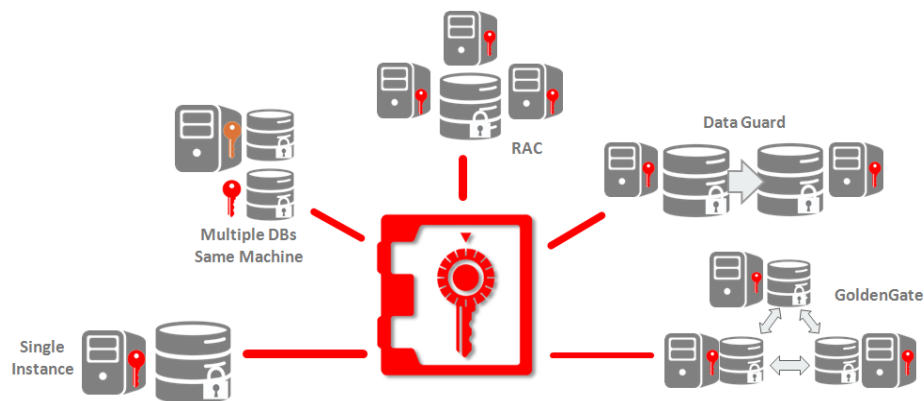


Oracle Key Vault Wallet Management Scenario

Online TDE Master Key Management

For Oracle databases that use TDE, Oracle Key Vault centrally manages TDE master keys over a direct network connection as an alternative to using local wallet files. This connection eliminates the operational challenges of wallet file management, such as periodic password rotation, backing up wallet files, and recovery from forgotten passwords. This method also provides physical separation between the encryption key and the encrypted data, which is often mentioned in regulatory compliance. The master keys stored in Oracle Key Vault can be made available for decrypting tablespace keys or table keys across databases according to endpoint access control settings.

This method of sharing keys without local wallet copies is useful when TDE is running on database clusters such as Oracle RAC, Oracle Active Data Guard, and Oracle GoldenGate. Existing master keys used for encrypted data in Oracle databases can be easily migrated from Oracle Wallet to Oracle Key Vault as part of the initial setup. Direct network connections between TDE and Oracle Key Vault are supported for Oracle Database 11gR2 and Oracle Database 12c without requiring database patching.



Oracle Key Vault Online TDE Master Key Scenario

Central Backup of Credential Files

Credential files that contain SSH keys, Kerberos keytab files, and similar credential files are also widely distributed without appropriate protective mechanisms. Oracle Key Vault backs up credential files for long-term retention and recovery. Oracle Key Vault easily recovers these files when needed, audits access to them, and shares them across trusted endpoints.

Additionally, Oracle Key Vault centralizes key management for MySQL Transparent Data Encryption, Solaris Crypto, and ASM Cluster File System (ACFS) file encryption solutions.


Installing Oracle Key Vault in Oracle Cloud Infrastructure

Oracle Key Vault is designed to be installed as a software appliance on a physical host in an on-premises network. Because of these deployment features, the current version of Oracle Key Vault cannot be installed as-is on a bare metal instance; instead, it is installed as a VM on a customer-owned bare metal instance. The customer installs a hypervisor on the bare metal instance, before installing the Oracle Key Vault VM. In this bring-your-own-hypervisor (BYOH) model, the customer manages the hypervisor as administrators, allowing them complete control over the bare metal instance and the Oracle Key Vault VM running on it.

This section provides information about obtaining the Oracle Key Vault image and license, installing the hypervisor, and installing the Oracle Key Vault VM on a bare metal instance. These instructions use the KVM hypervisor.

Obtain the Oracle Key Vault Image and License

Follow the [download instructions](#) to download an Oracle Key Vault ISO image for installation. See the [documentation](#) for installation and necessary administration tasks.



Oracle Key Vault is a separately licensed product within the Oracle Database Security product portfolio. Procure the necessary license for all production and non-production (test and development) environments.

Install a BYOH KVM and Configure VCN Networking for the Oracle Key Vault VM

For BYOH, the essential feature is the VCN's secondary VNIC. Secondary VNIC allows additional VNICs to attach to a bare metal instance, assign a VCN-routable IP address to the VNIC, and attach it to a VM running on the BYOH bare metal instance. For more information about secondary VNICs, see the [Networking service documentation](#).

This section summarizes the high-level steps for BYOH KVM installation for completeness. For detailed instructions, see the corresponding [Installing and Configuring KVM on Bare Metal Instances with Multi-VNIC](#) white paper. The high-level steps are as follows:

1. Launch a bare metal instance with an Oracle Linux 7.x image.
2. Log in to the bare metal instance with your SSH key, to test connectivity. If you cannot connect, check the VCN security lists and instance firewall rules. We recommend installing the VNC server on the bare metal instance, in order to be able to connect to the instance by using a VNC client. Instructions for configuring a VNC server on Oracle Linux are available at https://docs.oracle.com/cd/E52668_01/E54669/html/ol7-vnc-config.html.
3. In the Oracle Cloud Infrastructure Console, create a minimum 256-GB block storage volume and attach it to the bare metal instance. Mount a filesystem on the attached volume, and copy the Oracle Key Vault ISO into the mounted filesystem. Depending on the number of files to be stored, we recommend using a 1-TB block volume.
4. Using the Console or the API, attach a secondary VNIC, and note the IP address, MAC address, and VLAN tag of the secondary interface. Note that this is the secondary IP address that you will assign to the Oracle Key Vault VM so it can be network reachable from other VCN hosts, including the Oracle Cloud Infrastructure Database instance.

5. Install KVM hypervisor on the bare metal instance:

```
sudo yum install qemu-kvm qemu-img virt-manager libvirt libvirt-python  
libvirt-client virt-install virt-viewer bridge-utils
```

6. Enable SR-IOV and restart the bare metal instance. See the Appendix for details.
7. After the bare metal instance starts, enable the SR-IOV virtual functions (VFs) in the OS. Select a VF and configure it with the MAC address of the secondary VNIC that you created previously. See the Appendix for details.

8. Create a network interface by using the VLAN tag of the secondary VNIC. The interface is bridged with the VF that you configured in the previous step. See the Appendix for details.
9. Run `pifconfig` on the bare metal instance to show the network device created.

Install the Oracle Key Vault VM on the Bare Metal Instance

1. Create a 500G virtual disk by using `qemu-img`. This virtual disk will be used by the Oracle Key Vault VM.

```
qemu-img create -f raw <path_to_disk_image> 500G
```

2. Install Oracle Key Vault VM by using `virt-install`:

```
sudo virt-install --arch=x86_64 --name=<OKV_VM_name> --ram 16000 --cpu Haswell-noTSX --vcpus=4 --hvm --video qxl --nonetwork --os-type linux --noautoconsole --boot hd,cdrom -disk <path_to_OKV_ISO>,device=cdrom,bus=ide -disk <path_to_OKV_VM_disk_image>,format=raw,bus=scsi --graphics vnc,port=<VNC_port>,listen=0.0.0.0,password=<VNC_password>
```

The preceding command also creates a VNC connection to the Oracle Key Vault VM console to see the boot logs.

3. Create an SSH tunnel on the localhost, and use a VNC client to connect to the Oracle Key Vault VM console. This is especially useful when the installation has errors.


```
ssh -i <bare_metal_SSH_key> -L <VNC_port>:localhost:<VNC_port> opc@<bare_metal_host_IP>
```

<bare_metal_SSH_key> is the SSH key for connecting to bare metal instance,
<VNC_port> is the port number specified in `virt-install` in step 2, and
<bare_metal_host_IP> is the IP address of the bare metal instance.

On Mac, you can use the native VNC client (Screen Sharing) to connect to Oracle Key Vault VM console by using the `vnc://opc@localhost:<VNC_port>` and <VNC_password> configured in step 2.

4. Attach the VNIC network interface (created in the last section) using `virsh`. The correct VNIC MAC address and network device name must be filed in the `attach.xml` file (see the Appendix for details of the file). After attaching the VNIC network interface, destroy and restart the Oracle Key Vault VM.

```
sudo virsh attach-device <VM_name> ./attach.xml -config  
  
sudo virsh destroy <VM_name>  
  
sudo virsh start <VM_name>
```



When the VM starts installing, it should detect the VNIC network device attached to the VM. The VM installation takes about 30 minutes. More information about the Oracle Key Vault installation is available at

https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_install.htm#OKVAG10641.

During the installation, you are prompted for the following information:


- **Oracle Key Vault installation passphrase:** This passphrase is used for initial login to the Oracle Key Vault console.
 - **Oracle Key Vault network configuration:** This includes the Oracle Key Vault VM IP address, gateway IP address, and netmask. Provide the IP address of the attached secondary VNIC as the IP address of the Oracle Key Vault VM (OKV_VM_IP). Provide 10.0.0.1 for the gateway IP address, and 255.255.255.0 as the netmask
5. After installation is complete, open a web browser on the host bare metal instance, and type `https://OKV_VM_IP`, where `OKV_VM_IP` is the IP address of the Oracle Key Vault VM.

The browser opens the Oracle Key Vault console.

6. Use the installation passphrase to log in.
7. When prompted, set the username and password for the Key Administrator, System Administrator, and Audit Manager. Also when prompted, set the Recovery password (used for key recovery from secure backups), Root password (root privilege on the VM), and Support password (for SSH access to the VM). More information about the users to create in the Oracle Key Vault console is available at https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_install.htm#OKVAG10740.
8. Ensure that REST services are enabled on the Oracle Key Vault. In the Oracle Key Vault console, under **System**, select the **Restful Services** check box, and then save the configuration. You might want to disable the REST services after all the endpoints are enrolled and provisioned.

Configuring Oracle TDE Keys with Oracle Key Vault in Oracle Cloud Infrastructure

This section provides instructions on enrolling an Oracle database as an Oracle Key Vault endpoint, and migrating the TDE master key from Oracle Wallet to Oracle Key Vault. You perform these tasks by using the Oracle Key Vault RESTful utility. The RESTful utility is a KMIP client running on database instances, enrolling the databases as endpoints to the Oracle Key Vault



KMIP server. If you want to enroll multiple database instances, you can use the Oracle Key Vault RESTful utility programmatically to automate the enrollment process.

Enroll a Database Endpoint in Oracle Key Vault


1. Log in to the Oracle Key Vault console from the database instance by using `https://HOST_BARE_METAL_IP`. The IP table rules in the previous section forward port 443 on the host bare metal instance to the Oracle Key Vault VM.
2. In the console, click **RESTful Service Utility**, and download the utility to the instance. Assume the utility (`okvrestservices.jar`) is downloaded to `/home/oracle/Downloads` where `/home/oracle` is the home directory of the database instance.
3. In the `Downloads` directory, create the following two files (for representative examples of these files, see the Appendix):
 - `rest.ini` configuration file
 - `enroll_okv_endpoints` script file
4. In the `home` directory of the database instance, create an `okvhome` directory (`/home/oracle/okvhome`).
5. Use the following command to enroll the database as an endpoint and create an Oracle Key Vault virtual wallet in which to store the TDE master keys:

```
java -jar okvrestservices.jar --config rest.ini --script enroll_okv_endpoint
```

The command prompts for the Oracle Key Vault admin password.

Assume that `CUSTOMER_DB` is the name of the Oracle database endpoint and `CUSTOMER_DB_WALLET` is the Oracle Key Vault virtual wallet that will hold the database TDE master key. On successful completion of the command, the `CUSTOMER_DB` endpoint should be listed in the Oracle Key Vault console on the **Endpoints** tab. You should also see a virtual wallet called `CUSTOMER_DB_WALLET` on the **Keys & Wallets** tab. At this point, the `CUSTOMER_DB_WALLET` should be empty. The command also creates the `CUSTOMER_DB` subdirectory in `/home/oracle/okvhome` directory with various utilities (including `okvutil`, which you will use in the next section).

6. Run the `/home/oracle/okvhome/CUSTOMER_DB/bin/root.sh` file as root (or `sudo`). This copies the Oracle Key Vault PKCS#11 driver to a specified location in the Oracle database filesystem (`/opt/oracle/extapi/64/hsm/oracle/1.0.0`). The Oracle database uses functions in this driver to interact with Oracle Key Vault.



In Oracle databases 12.1.0.2 and 12.2.0.1, an Oracle Key Vault wallet is created for each container database (CDB), and all the TDE master keys of all the multi-tenant pluggable databases (PDBs) under the CDB are stored in the same wallet. This is irrespective of the number of PDBs running. This could change in a future version where separate Oracle Key Vault wallets can be created for PDBs.

Upload Oracle Wallet Keys to Oracle Key Vault

Use the following command to upload the Oracle Wallet contents to the Oracle Key Vault. In the example, the Oracle Wallet is located at `/etc/oracle/wallets/orcl` on the Oracle database instance (`orcl` is the database SID name). Replace it with your database SID.

```
/home/oracle/okvhome/CUSTOMER_DB/bin/okvutil upload -t WALLET -g CUSTOMER_DB_WALLET -l /etc/oracle/wallets/orcl
```

The command prompts for the Oracle Wallet password.

On successful completion of this command, you should see various key and certificate identifiers on the **Keys & Wallets** tab of the Oracle Key Vault console.

You can also use the following command to list the IDs of keys and certificates for the `CUSTOMER_DB` endpoint in Oracle Key Vault. The results should correspond with the contents listed under `CUSTOMER_DB` in the Oracle Key Vault console.

```
/home/oracle/okvhome/CUSTOMER_DB/bin/okvutil list
```

Migrate TDE Master from Oracle Wallet to Oracle Key Vault

1. Close the Oracle Wallet on the database instance by using the following command in a `sqlplus` terminal. `WALLET_PASSWD` is the password of the Oracle Wallet on the database instance.

```
administer key management set keystore close identified by "WALLET_PASSWD";
```

2. Alter the `$ORACLE_HOME/network/admin/sqlnet.ora` configuration file from `(METHOD=FILE)` to `(METHOD=HSM)`.
3. Use the following command to verify that the changes are in effect. Log in to the database by using `sqlplus / as sysdba` to issue the command in a `sqlplus` terminal. Both `FILE` and `HSM` should be `CLOSED`.

```
select wrl_type,status from v$encryption_wallet;
```

4. Migrate the TDE master key from Oracle Wallet to Oracle Key Vault by using the following command in a `sqlplus` terminal. In the command, "null" indicates a null password for the database endpoint. (If you use a non-null value, whenever the database accesses Oracle Key Vault for the TDE master key, it prompts for a password, which could pose issues for lights-out operation.)

```
administer key management set encryption key identified by "null" migrate
using "WALLET_PASSWD" with backup;
```

The Migrated TDE master key identifier should be visible on the **Keys & Wallets** tab in the Oracle Key Vault console.

5. Subsequently, you can rotate the TDE master key in the Oracle Key Vault by using the following command in a `sqlplus` terminal:

```
administer key management set encryption key identified by "null";
```

Tip: To migrate the TDE master key back to Oracle Wallet, change (METHOD=HSM) to (METHOD=FILE) in the `$ORACLE_HOME/network/admin/sqlnet.ora` configuration file, and issue the following command at a `sqlplus` terminal:

```
administer key management set encryption key identified by "null" reverse
migrate using "WALLET_PASSWD" with backup;
```

Oracle Key Vault Best Practices

Use the following Oracle Key Vault best practices to enhance security and operations.

Prototype with Non-Critical Database Workloads

Losing TDE master keys results in the inability to access encrypted data (encrypted tablespaces and columns) in the Oracle database. Therefore, when deploying *any* key management solutions including Oracle Key Vault, we *highly recommend* initially prototyping the key management solution by using appropriate, noncritical development or test database workloads. This plan enables you to gain sufficient understanding of all aspects of the key management solution while formulating a key management architecture suitable for handling mission-critical production workloads.

Secure Key Backups

To support disaster recovery (DR), you are *required* to make periodic secure backups of stored Oracle Key Vault TDE master keys. Losing the TDE master keys prevents access to the encrypted database data. Following are options for creating backups of Oracle Key Vault TDE master:

- **Automated Block Volume backups:** Create periodic snapshots of the block storage volume that stores the Oracle Key Vault VM. You can configure the backups and frequency in the Oracle Cloud Infrastructure Console. Note that the Oracle Key Vault appliance natively encrypts keys stored at rest, and an Oracle Cloud Infrastructure block storage volume is encrypted at rest by the Oracle Cloud Infrastructure control plane.
- **Automated Oracle Key Vault secure backups:** Configure periodic, automated Oracle Key Vault key backups to the host bare metal instance (because only the host bare metal instance is reachable from the Oracle Key Vault VM). Additionally, we recommend copying the Oracle Key Vault key backup files to a customer-owned bucket in Oracle Cloud Infrastructure Object Storage, in order to free space on the host bare metal instance. For DR, the Oracle Key Vault is restored from these backups by using the Recovery password (do not lose the Oracle Key Vault Recovery password).

Oracle Key Vault secure backup instructions are available at https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_ha_backup.htm#OKVAG10746.

Configure for High Availability

High availability (HA) comprises using primary and standby Oracle Key Vault servers where online backups of keys are made from primary to secondary. We recommend having primary and secondary Oracle Key Vault VMs on two separate BYOH bare metal hosts.

Enable Oracle Key Vault SSH Access

SSH access is useful for troubleshooting and performing operational activities with the Oracle Key Vault VM, and we recommend enabling SSH access to the VM. In the Oracle Key Vault console, go into the System settings and enable SSH access to the Oracle Key Vault VM from the host bare metal instance. After this step, you can log in to the Oracle Key Vault VM from the host bare metal instance by using `ssh support@OKV_VM_IP`.

Instructions for enabling SSH access are available at http://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_appliance.htm#OKVAG10885.



Use Oracle Key Vault Audit Logs

Oracle Key Vault records all actions related to stored TDE master keys in an audit trail. Each row of the audit trail shows who (Oracle Key Vault user) performed what action (Operation) on which object (TDE Master Key), and the result of the action. These Oracle Key Vault audit trails can be exported by the Oracle Key Vault Audit Manager as CSV files, for auditing.

Instructions for exporting Oracle Key Vault audit trails are available at https://docs.oracle.com/cd/E65319_01/OKVAG/okv_appliance.htm#OKVAG10870.

Use VCN Security Lists to Firewall Oracle Key Vault Instances

You can use VCN security lists on the host bare metal instance to allow network connections to the Oracle Key Vault VM only from authorized database instances in the VCN. The security lists allow TCP connections on port 5696 (Oracle Key Vault KMIP server port) of the host bare metal instance only from IP addresses that correspond to the database instances configured as Oracle Key Vault endpoints. If remote access is required for the Oracle Key Vault web console, you also need to allow access on port 443.

Manage Oracle Key Vault in the Oracle Cloud Infrastructure Environment

Oracle Key Vault instances are completely managed by the customers who are using them to store the TDE master keys of their Oracle Cloud Infrastructure databases. Customers are responsible for the installation and configuration of the Oracle Key Vault instances on bare metal instances in their VCN, and for their management, including adding database instances as Oracle Key Vault endpoints. Customers are also responsible for setting up automated backups of the stored Oracle Key Vault keys, and restoring them for DR. Oracle Cloud Infrastructure is not involved in any management of the Oracle Key Vault VMs.

For customer-managed Oracle databases in Oracle Cloud Infrastructure, Oracle Key Vault works in Oracle Cloud Infrastructure as it would on-premises. In the case of Oracle Cloud Infrastructure Database instances, certain features such as RMAN and Data Guard are automated and they assume that the TDE key is present in the Oracle Wallet on the database instance. If you need to use Oracle Key Vault with Oracle Cloud Infrastructure Database instances, work with the Oracle Cloud Infrastructure team.



Configure Oracle Key Vault Active Data Guard

Oracle Key Vault removes the manual steps of copying keys between primary and standby databases in an Active Data Guard configuration. This is done by registering a shared Oracle Key Vault wallet between the primary and standby.

For instructions, go to

https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_scenarios.htm#OKVAG10849.

Configure Oracle Key Vault RAC

Define an Oracle Key Vault virtual wallet, and configure it as a shared default wallet between all the RAC nodes.

For instructions, go to

https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_scenarios.htm#OKVAG10704.

Configure Oracle Key Vault GoldenGate

When the source Oracle database is configured as an Oracle Key Vault endpoint, the GoldenGate password is stored in the same Oracle Key Vault virtual wallet as the TDE master keys of the source database. The target database is configured as another Oracle Key Vault endpoint.

For instructions, go to

https://docs.oracle.com/cd/E50341_01/doc.1210/e41361/okv_scenarios.htm#OKVAG10845.

Conclusion

This paper presents an Oracle Key Vault solution for managing Oracle database TDE master keys in Oracle Cloud Infrastructure. The Oracle Key Vault appliance runs on a customer-owned bare metal instance and allows you to completely control all the keys while leveraging Oracle Key Vault functionality to ease the operational effort associated with managing TDE master keys of multiple Oracle databases in Oracle Cloud Infrastructure. Depending on your operational requirements, you have the choice of using a combination of Oracle Wallet and Oracle Key Vault to manage the Oracle database encryption keys in Oracle Cloud Infrastructure.

Appendix

Enable SR-IOV on a Bare Metal instance

1. In the `/etc/default/grub` file, add `intel_iommu=on` on the `GRUB_CMDLINE_LINUX` line.

2. Generate new grub configuration file:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

3. Reboot the bare metal server.

Enable VFs and Configure with MAC Address of Secondary VNIC

1. Enable virtual functions (VFs) and set the `vepa` bridging mode. On Oracle Linux, `ens2f0` is the physical interface.

```
echo "16" > /sys/class/net/ens2f0/device/sriov_numvfs  
  
bridge link set dev ens2f0 hwmode vepa
```

2. List the available VFs. Note the VF number (`VF_NUM`) of an available VF.

```
ip link show ens2f0
```

3. Configure the VF with MAC address of VNIC (`VNIC_MAC`):

```
ip link set ens2f0 vf VF_NUM mac VNIC_MAC spoofchk off
```

Create Network Interface Using the VLAN Tag of the Secondary VNIC

1. Get the VF network device name (`VF_DEVICE_NAME`).

For VF numbered `VF_NUM`, select the `(VF_NUM+1)` line number in the output of the following command. For example, if `VF_NUM` is equal to 1, then pick the second line of the output. The port, slot, and function number are listed in hexadecimal format, as the first field of the line. For example, `13:10:2` denotes port number 19, slot number 16, and function number 2, and the `VF_DEVICE_NAME` is `enp19s16f2`.

```
lspci -nn | grep -i virtual
```

2. Bring up the VF network device:

```
ip link set VF_DEVICE_NAME down  
  
ip link set VF_DEVICE_NAME up
```

3. Assign the VF network device to the VNIC VLAN:

```
ip link add link VF_DEVICE_NAME name VLAN_DEVICE_NAME type vlan id
VNIC_VLAN_TAG

ip link set VLAN_DEVICE_NAME up
```

attach.xml file

```
<interface type='direct'>
  <mac address='<VNIC_MAC>' />
  <source dev='<VLAN_DEVICE_NAME>' mode='passthrough' />
  <model type='e1000' />
</interface>
```

rest.ini file

The contents of the `rest.ini` file are as follows:

```
server=OKV_VM_IP
usr=<OKV_ADMIN_USER>
log_level=ALL
```

<OKV_ADMIN_USER> is the username used for logging in to the Oracle Key Vault console.

enroll_okv_endpoint file

```
create_wallet -wallet_name CUSTOMER_DB_WALLET
create_endpoint -ep_name CUSTOMER_DB -ep_platform LINUX64 -ep_type ORACLE_DB
set_default_wallet -ep_name CUSTOMER_DB -wallet_name CUSTOMER_DB_WALLET
provision -autologin -ep_name CUSTOMER_DB -dir /home/oracle/okvhome
```

In the file, `CUSTOMER_DB` is the name of the Oracle database endpoint, and `CUSTOMER_DB_WALLET` is the Oracle Key Vault virtual wallet with the TDE master keys of the enrolled database. For each Oracle database enrolled as an endpoint, unique names should be provided in the script. For each database to be enrolled as an endpoint, we recommend using the following formats for the database endpoint and the wallet names:

- `ORACLE_DB_$$SID`, where `$$SID` is the Oracle database SID
- `ORACLE_DB_$$SID_WALLET`, where `$$SID` is the Oracle database SID




Oracle Corporation, World Headquarters


500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

 oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0518

Managing Oracle Database Encryption Keys in Oracle Cloud Infrastructure with Oracle Key Vault
May 2018

Authors: Nachiketh Pottapally, Saikat Saha



Oracle is committed to developing practices and products that help protect the environment