Deploying Oracle Audit Vault and Database Firewall in Oracle Cloud Infrastructure

Database Security and Compliance Auditing
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.
# Table of Contents

Oracle Cloud Infrastructure Database Security 5
  Database Options 5
  Supported Security Features 6
  Database Auditing 6

Oracle Database Auditing and Oracle AVDF Overview 7
  Oracle Database Audit Capabilities 7
  Audit Vault and Database Firewall Security Appliance 9

Installing Oracle AVDF on an Oracle Cloud Infrastructure BM Instance 10
  Obtain the Oracle AVDF Image and License 11
  Install a BYOH KVM for Running the Oracle AVDF VM 11
  Install the Oracle AVDF VM on the BM instance 12

Configuring a DBaaS Instance with Oracle AVDF 15
  Install the Audit Vault Agent on a DBaaS Instance 15
  Registering the DBaaS Instance in the Audit Vault Server 16
  Start the Audit Vault Agent on the DBaaS Instance 17
  Configure the DBaaS Instance Database as an Oracle AVDF Secured Target 18
  Configure the Audit Trail in Oracle AVDF 18
  Provision Database Audit Policies 19
  Monitor Database Activity in Oracle AVDF 21

Oracle Database Auditing and Oracle AVDF Best Practices 21
  Use VCN Security Lists to Firewall the Oracle AVDF VM 21
  Know the Performance Impact of Auditing on the DBaaS Instance 22
  Know Your Audit Vault Server Storage Requirements 22
Archive Audit Records to Oracle Cloud Infrastructure Object Storage 22
Configure for High Availability 22
Enable SSH Access to the Audit Vault Server 23

Conclusion 23
FAQ 23
Appendix 24

Enable SR-IOV on a BM instance 24
Enable VFs and Configure with MAC Address of Secondary VNIC 24
Create Network Interface Using the VLAN Tag of the Secondary VNIC 24
attach.xml file 25
Oracle Cloud Infrastructure Database Security

Oracle Cloud Infrastructure offers a wide variety of high-performance Oracle Databases in the cloud. Security is an important consideration in the cloud, and Oracle Cloud Infrastructure databases provide many security features by default. Oracle Cloud Infrastructure offers Oracle Databases on Compute bare metal (BM) instances, which are not managed by a higher-privileged Oracle Cloud Infrastructure-controlled hypervisor; this provides maximum security isolation to customers. Oracle Cloud Infrastructure also offers virtual machine (VM) based databases, which adopt the latest security practices for virtualization security.

Database Options

The Oracle Cloud Infrastructure Database service (DBaaS) provides the following database options:

- **Bare metal (BM) database instance**: Elastic and on-demand Oracle Databases on the BM instance with NVMe local flash storage, in 1-node and 2-node RAC configuration. The 2-node RAC uses shared storage with automatic failover. The different shapes for the BM database instances are as follows:
  - 1-node HighIO: 36 cores, 512 GB RAM, and 12.8 TB NVMe storage
  - 1-node DenseIO: 36 cores, 512 GB RAM, and 28.8 TB NVMe storage
  - 2-node RAC: Two BM instances with 36 cores on each node (72 cores in total), 512 GB memory, and 24 TB (composed of 20 x 1.2TB SSD drives) direct attached shared storage

- **Virtual machine (VM) database instance**: Oracle Databases on 1-node and 2-node VMs, where each VM node is configurable with 1, 2, 4, 8, or 16 cores. The 2-node VMs are in RAC configuration.

- **Exadata database**: Full, half, or quarter rack Exadata X6 systems composed of compute and storage nodes connected with a low-latency InfiniBand network, running optimized Oracle Database software.

- **Customer-managed database**: Customers can install an Oracle Database on a BM instance and manage it. Although the majority of customers will use the preceding DBaaS instances, this option is provided for completeness.

The Oracle Cloud Infrastructure DBaaS instances support Oracle Database Releases 11.2, 12.1, and 12.2, along with popular manageability features such as backup and restore, patching, and Oracle Data Guard. Oracle Database customers with an Unlimited License Agreement or Non-Unlimited License Agreement can use their license with Oracle Cloud Infrastructure DBaaS instances in the cloud.
Supported Security Features

To provide enhanced security for mission-critical customer databases, Oracle Cloud Infrastructure DBaaS instances support a variety of security features by default:

- **Database encryption by default:** Oracle Cloud Infrastructure databases have Transparent Data Encryption (TDE) enabled by default. TDE encrypts the entire tablespace or specific table columns, with database encryption keys stored natively in the database. A TDE master key is used to wrap (encrypt) the database encryption keys, and the TDE master key is stored in an Oracle Wallet in the local filesystem or Automatic Storage Management (ASM) disk group for clustered access. Options exist to manage the TDE master keys by using Oracle Key Vault or by storing the TDE master key in an hardware security module (HSM).

- **Defense-in-depth security isolation:** Oracle Cloud Infrastructure database instances are launched in a customer virtual cloud network (VCN), which provides network security isolation to the instances. For example, a database instance can be launched in a VCN private subnet, which prevents all external access to the database. In addition, BM instances provide more security isolation for customer database data and the storage of TDE master keys. A database agent, running on the Oracle Cloud Infrastructure database instance, is responsible for provisioning and management of the instance, but no Oracle Cloud Infrastructure operators can access the BM instance.

- **Network access control:** Using VCN security lists, network firewalls can be configured to allow access to Oracle Cloud Infrastructure database instances only from customer-configured IP addresses.

- **IAM authentication and authorization:** Using Oracle Cloud Infrastructure IAM, an Oracle Cloud Infrastructure database can be instantiated in an IAM compartment, and authorized users and groups can have authenticated access to the database instance.

- **Automated secure backups:** Oracle Cloud Infrastructure databases have automated and encrypted backups of database data, and the backups are stored in the customer’s Object Storage bucket.

- **API audit logs:** All the API calls made to provision, shut down, and configure customer Oracle Cloud Infrastructure databases are provided to the customer for security and compliance auditing.

Database Auditing

For security and compliance, database auditing is crucial. Database auditing records all database access in an audit log by including information such as which database object was accessed or modified, what database account performed the action, and when the action was taken. Database audit logs are used by security teams to create security alerts to trigger notifications when certain
security-critical events occur, or to perform root cause analysis as part of forensics for investigating potential security incidents (such as intrusions, data integrity attacks, and data loss). The ability to monitor database access by employees and privileged users for detecting insider attacks is becoming increasingly important for customers.

In addition to security requirements, compliance regulations such as the Health Insurance Portability and Accountability Act (HIPAA), the new European Union General Data Protection Regulation (EU GDPR), and the Sarbanes-Oxley Act (SOX) require enterprises to provide detailed reports showing database access, down to the row level.

Oracle Audit Vault and Database Firewall (Oracle AVDF) provides a comprehensive security solution for auditing and monitoring databases (Oracle and others), along with other relevant host-level logs (such as OS and network). This white paper focuses on using the Oracle Audit Vault Server for auditing access to Oracle Cloud Infrastructure DBaaS instances.

**Note:** In this paper, Oracle AVDF refers only to the Audit Vault Server functionality. Database Firewall is out of scope for this white paper and will be handled separately.

On-premises Oracle AVDF deployments can collect audit data from both cloud and on-premises databases. The objective of this white paper is to provide instructions for customers who want to install and configure Oracle AVDF on an OCM instance in their VCN to audit and monitor their Oracle Cloud Infrastructure DBaaS instances.

### Oracle Database Auditing and Oracle AVDF Overview

Oracle AVDF provides a security solution for monitoring and alerting on database access events. Audit Vault ingests various types of logs, including audit trails from Oracle and non-Oracle databases, OS logs, network logs, and application logs, providing a unified security audit and monitoring solution. However, in order to use Oracle AVDF, an essential prerequisite is to enable audit trail in the Oracle Database. This section provides an overview of Oracle Database auditing and Oracle AVDF.

### Oracle Database Audit Capabilities

Oracle Database has a comprehensive set of logging capabilities for recording various types of database access events in the audit trail. The audit trail is written to the SYS.AUD$ and SYS.FGA_LOG$ tables in the SYSTEM tablespace.
The Oracle Database audit log options are as follows:

- **Mandatory auditing**: Certain events, such as database startup, database shutdown, and SYSDBA and SYSOPER logins, are recorded automatically. This auditing is mandatory and cannot be turned off. For database startup and shutdown, the audit log includes data, timestamp, user, and user terminal information.

- **Standard auditing**: Using the AUDIT statement, a user can audit various types of access to schema objects. Depending on the AUDIT_TRAIL parameter value, the audit logs are written to the SYS.AUD$ table or to the OS syslogs. The AUDIT statement can be modified with qualifiers such as BY ACCESS (insert an audit record for each audited statement), BY SESSION (insert one audit record for the session including the audited statement), WHENEVER SUCCESSFUL (insert audit record when user action succeeds), and WHENEVER NOT SUCCESSFUL (insert audit record when user action fails). The following types of events are audited:
  
  - **SQL statement auditing**: Audit DML and DDL statements related to schema objects or database structures, but not named objects. For example, to audit all successfully executed statements by user BOB, we can use AUDIT ALL STATEMENTS BY bob BY ACCESS WHENEVER SUCCESSFUL. If we want to constrain auditing to specific table actions by bob, we can use AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE BY bob BY ACCESS.

  - **Privilege auditing**: Audit the use of system privileges. This auditing is triggered only when a user action requires a system privilege. For example, suppose user BOB is granted the SELECT ANY TABLE privilege, and this privilege is being audited. If user BOB selects a table in his schema (BOB.EMPLOYEES), it is not audited because the SELECT ANY TABLE privilege is not used. However, if BOB selects a table from another schema (for example, HR.EMPLOYEES), an audit record is generated. A set of privileges (for example, CREATE USER, DROP USER, and ALTER USER) are audited by default.

  - **Schema object auditing**: Audit the actions by all users on audited schema objects such as tables, views, stored procedures, functions, and packages. For example, AUDIT SELECT on BOB.EMPLOYEES creates audit records for the SELECT operation by any user on the BOB.EMPLOYEES object.

  - **Network auditing**: Audit errors in the network layer by using AUDIT NETWORK.

- **Fine-grained auditing (FGA)**: Fine-grained auditing enables you to create policies that define specific conditions that must happen for the audit to occur. This type of auditing enables you to monitor data access based on content. It provides granular auditing of queries, and INSERT, UPDATE, and DELETE operations.
In Oracle 12c, all the audit trails (SYS.AUD$, SYS.FGA_LOG$, DVSYS.AUDIT_TRAIL$, and so on) have been unified into a single view, SYS.UNIFIED_AUDIT_TRAIL. With this, audit tools such as Oracle AVDF can analyze an entire set of audit data in one location, rather than having to gather the data into one location. A new schema AUDSYS is used for storing the unified audit data. The following figure shows the high-level operation of Oracle 12c unified audit:

Unified Audit in Oracle 12c

For better separation of duties, two new database roles are available for auditing: AUDIT_ADMIN for managing database audit management, and AUDIT_VIEWER for viewing audit trails only. More information about Oracle 12c unified audit is available at http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/12c/r1/security/sec_uni_audit/sec_uni_audit.html.

Audit Vault and Database Firewall Security Appliance

Oracle AVDF leverages agents running on the database instances, which send the database audit trail records to an Audit Vault Server. The Audit Vault Server is a security appliance, designed to run on a physical host or a VM, and a centralized resource for log aggregation, monitoring, and alerting. The Audit Vault Server management is done using a web console. Oracle AVDF has two primary users: administrator, who can configure databases and audit policies; and auditor, who can manage Oracle AVDF and view audit records.
The following figure shows the Audit Vault Agent, running on the database instance and sending all the audit data to an Audit Vault Server. The Audit Vault Server can aggregate audit data from multiple database instances.

Audit Vault Agent on Database Instance and Audit Vault Server

Based on the audit data aggregated, Audit Vault Server generates a variety of audit reports including activity and compliance reports. Activity reports enumerate events such as failed logins, database schema changes, and SQL statements. Out-of-the-box audit assessment reports are available for compliance regulations such as PCI-DSS, HIPAA, and SOX. Authorized security auditors can access all these reports through the Oracle AVDF web console.

Installing Oracle AVDF on an Oracle Cloud Infrastructure BM Instance

Oracle AVDF is a security appliance designed for installation on an on-premises physical host with network reachability to all databases to be monitored. Because of these deployment features, the current version of Oracle AVDF cannot be installed as-is on a BM or VM instance. The two options for deploying Oracle AVDF in Oracle Cloud Infrastructure are as follows:

- Using an Oracle AVDF boot image customized for installation on an Oracle Cloud Infrastructure instance
- Running Oracle AVDF as a VM on a customer bring-your-own hypervisor (BYOH) on an Oracle Cloud Infrastructure BM instance

In terms of Oracle AVDF functionality and robustness, the BYOH solution is equivalent to Oracle AVDF running natively on a BM instance (albeit with the overhead of managing a hypervisor). This white paper documents the BYOH option.
In a BYOH model, customers may install and manage a hypervisor on their Oracle Cloud Infrastructure BM instance and run Oracle AVDF as a VM on the hypervisor. Here, the customer is the hypervisor administrator, and has complete control over the BM instance and the Oracle AVDF VM running on it. In this white paper, the KVM hypervisor is used for the customer BYOH. KVM is a robust, secure, and high-performance hypervisor, available in several Linux distributions. It is deployed in many production environments, including multiple public clouds.

This section provides information about obtaining an Oracle AVDF image and license, installing the hypervisor, and installing the Oracle AVDF VM on an Oracle Cloud Infrastructure BM instance.

Obtain the Oracle AVDF Image and License

Follow the [download instructions](#) to download an Oracle AVDF ISO image for installation (use the latest version). See the [documentation](#) for installation and administration tasks.

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

Install a BYOH KVM for Running the Oracle AVDF VM

For BYOH, the essential feature is the VCN’s secondary VNIC. Secondary VNIC allows additional VNICs to attach to a BM instance, assign a VCN-routable IP address to the VNIC, and attach it to a VM running on the BYOH BM 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 BM instance with an Oracle Linux 7.x image.

2. Log in to the BM 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 BM instance, in order to be able to connect to the BM 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](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 BM instance. Mount a filesystem on the attached volume, and copy the Oracle AVDF ISO into the mounted filesystem. Depending on the number of audit records 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 AVDF VM so it can be network reachable from other VCN hosts, including the Oracle Cloud Infrastructure DBaaS instance.

5. Install KVM hypervisor on the BM instance:

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

6. Enable SR-IOV and restart the BM instance. See the Appendix for details.

7. After the BM 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 BM instance to show the network device created.

### Install the Oracle AVDF VM on the BM instance

Now you have a BM instance running a KVM hypervisor, and a secondary VNIC with a VCN-routable IP address to be assigned to the Oracle AVDF VM. Follow these steps to install the Oracle AVDF VM.

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

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

2. Install the Oracle AVDF VM by using `virt-install`:

   ```bash
   sudo virt-install --arch=x86_64 --name=<AVDF_VM_name> --ram 16000 --cpu Haswell-noTSX --vcpus=4 --hvm --video qxl --nonetwork --os-type linux --noautoconsole --boot hd,cdrom --disk <path_to_AVDF_ISO>,device=cdrom,bus=ide --disk <path_to_AVDF_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 AVDF VM console to see the boot logs.
3. Create a SSH tunnel on the localhost, and use a VNC client to connect to the Oracle AVDF VM console. This is especially useful when the installation has errors.

```
ssh -i <BM_SSH_key> -L <VNC_port>:localhost:<VNC_port> opc@<BM_host_IP>
```

`<BM_SSH_key>` is the SSH key for connecting to BM instance, `<VNC_port>` is the port number specified in `virt-install` in step 2, and `<BM_host_IP>` is the IP address of the BM instance.

On Mac, you can use the native VNC client (Screen Sharing) to connect to Oracle AVDF 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 AVDF VM.

```
sudo virsh attach-device <AVDF_VM_name> ./attach.xml -config
sudo virsh destroy <AVDF_VM_name>
sudo virsh start <AVDF_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 AVDF installation is available at [https://docs.oracle.com/cd/E37100_01/doc.1211/e27778/install.htm#SIGIG177](https://docs.oracle.com/cd/E37100_01/doc.1211/e27778/install.htm#SIGIG177).

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

- **Oracle AVDF installation passphrase**: This passphrase is used for initial login to the Oracle AVDF web console.
- **Oracle AVDF network configuration**: This includes Oracle AVDF VM IP address, gateway IP address, and netmask. Provide the secondary IP address (from the attached VNIC) as the IP address of the Oracle AVDF VM (AVDF_VM_IP). Provide 10.0.0.1 for the gateway IP address, and 255.255.255.0 as netmask.

5. After the installation is complete, open a web browser on the host BM instance, and type `https://AVDF_VM_IP`, where `AVDF_VM_IP` is the IP address assigned to the Oracle AVDF VM.

The browser opens the Oracle AVDF console.

6. Use the installation passphrase to log in.
7. When prompted, set the username and password for the Administrator and Audit Manager. Also when prompted, set the Root password (root privilege on the VM) and Support password (for SSH access to the VM). Also, it is important to configure the time on the Audit Vault Server and DBaaS instances by using NTP to keep time synchronized. Unsynchronized time between the Audit Vault Server and DBaaS instances negatively affects the collection of database audit trails.

To open the graphical Oracle AVDF web console on the BM instance, you need an X windows connection to the BM instance through a VNC connection to the BM instance, or SSH with X windows enabled. The following figure shows the Oracle AVDF administrator console of the Oracle AVDF VM (with IP address 10.0.0.12) running on a BYOH BM instance.

8. Configure the VCN security lists to ensure that Audit Vault Server is reachable from the DBaaS instances to be audited. Check the network connectivity between the Audit Vault Server and DBaaS instances by using ICMP ping. If all the previous steps were executed correctly, the Audit Vault Server and DBaaS instances should be able to reach each other.
Configuring a DBaaS Instance with Oracle AVDF

This section provides instructions for deploying the Audit Vault Agent on DBaaS instances and collecting database audit trails in the Audit Vault Server. These instructions are intended to get you started with using Oracle AVDF for auditing DBaaS instances; they are not meant to be comprehensive. For details about configuration options, see the comprehensive Oracle AVDF documentation.

Install the Audit Vault Agent on a DBaaS Instance

1. Log in to the Oracle AVDF console as administrator.

2. On the Hosts tab, click Agent and download the agent.jar file.

3. Copy the agent.jar file to the DBaaS instance.

4. Set $AVDF_AGENT_HOME as the directory where the Audit Vault Agent will be installed. The following command creates the $AVDF_AGENT_HOME directory and installs the agent:

   ```
   java -jar agent.jar -d $AVDF_AGENT_HOME
   ```

   The following figure shows the $AVDF_AGENT_HOME (/home/oracle/avdf_agent) on a DBaaS instance (dbtest), and various files in the subdirectories. /bin/agentctl is the script for enabling the agent.
Registering the DBaaS Instance in the Audit Vault Server

1. Verify that the database audit trail is enabled on the DBaaS instance by running the following command:
   
   ```
   show parameter audit
   ```
   
   The `AUDIT_TRAIL` parameter should have the value `DB`. If `AUDIT_TRAIL` is set to `NONE`, perform the following steps:
   
   A. Connect to the database as SYS.
   
   B. Use the following command to enable the audit trail:
      
      ```
      ALTER SYSTEM SET AUDIT_TRAIL=DB
      ```
   
   C. Shut down and restart the database to activate the audit trail.
   
2. Because Oracle AVDF requires privileges to collect audit data from the database and manage audit policies, you must create a user (`audituser`) with the appropriate privileges. Oracle AVDF provides a PL/SQL script (`oracle_user_setup.sql`) to configure `audituser` with the appropriate privileges. The script is available at
   
   `$AVDF_AGENT_HOME/av/plugins/com.oracle.av.plugin.oracle/config`.
   
   Run the following commands at the SQL prompt on the DBaaS instance to create the `audituser` user with the necessary privileges:
   
   ```
   CREATE USER audituser IDENTIFIED BY <password>
   ```
   
   `<password>` is a strong password for `audituser`.
   
   ```
   CONNECT SYS / AS SYSDBA
   ```
   
   ```
   @oracle_user_setup.sql audituser SETUP
   ```
   
3. Log in to the Oracle AVDF console as administrator.
   
4. On the Hosts tab, click Register.
   
5. Enter the DBaaS instance name (`DB_NAME`) in the Host Name field, and the IP address (`DB_IP`) in the Host IP field.
   
6. Click Save.

   A unique activation key is generated. Copy the activation key, which you will use in the next section while installing the Audit Vault Agent. Note that the `DB_NAME` has no relation to the `ORACLE_SID` of the database running on the DBaaS instance, and could be any meaningful string.
Start the Audit Vault Agent on the DBaaS Instance

1. On the DBaaS instance, go to the $AVDF_AGENT_HOME directory, and run the following command:
   
   ```bash
   ./agentctl start -k
   ```

2. When prompted for the agent activation key, provide the activation key that you copied from the Oracle AVDF console. The activation key is available in the Hosts tab when you log in to the console as administrator.

3. To verify that the Audit Vault Agent was successfully activated and is running, check the Agent Status on the Hosts tab in Oracle AVDF console (after logging in as administrator). The status should say Running with a green dot.

   In addition, the DB_NAME and DB_IP values should be listed in the Host Name and Host IP columns. The following figure shows the status of the agent, the DBaaS host name (dbtest), and the IP address (10.0.0.13).
Configure the DBaaS Instance Database as an Oracle AVDF Secured Target

1. On the DBaaS instance, go to the $ORACLE_HOME/network/admin/tnsnames.ora file, and copy the value of the SERVICE_NAME parameter.

2. Log in to the Oracle AVDF console as administrator.

3. On the Secured Targets tab, click Register.

4. Enter the following values in the fields and then click Save:
   - **New Secured Target Name**: ORACLE_SID (SID of the DBaaS instance database to be audited)
   - **Secured Target Type**: Oracle Database
   - **Host Name/IP Address**: DB_IP
   - **Protocol**: TCP
   - **Port**: 1521
   - **Service Name**: SERVICE_NAME parameter value from step 1
   - **User Name**: audituser
   - **Password**: <audituser_password> (password created for audituser in a previous step)

**Note:** It is important to synchronize the time on the Audit Vault Server and secured targets (DBaaS instances, in this case) by using the NTP server. Unsynchronized time negatively affects audit trail collection.

Configure the Audit Trail in Oracle AVDF

1. Log in to the Oracle AVDF console as administrator.

2. On the Secured Targets tab, click Audit Trails under Monitoring in the left-hand pane, and then click Add.

3. Enter the following values in the fields and then click Save:
   - **Audit Trail Type**: TABLE
   - **Collection Host**: DB_NAME
   - **Secured Target**: ORACLE_SID (provided during secured target configuration)
   - **Trail Location**: sys.aud$
Provision Database Audit Policies

You can provision audit policies in the database from the Audit Vault Server. Provisioning new policies and modifying existing ones requires auditor user privileges.

1. Log in to the Oracle AVDF console as auditor.

2. On the Policy tab, select the secured target for which you want to create policies.

   The console shows all the audit policies. The following figure shows audit policies for the dbtest secured target.

   ![Audit Policies Figure]

   Note: The following steps show an example of adding a statement audit policy, but the steps are generic and can be used for any audit policy.

3. To add an audit policy for statements, click Statement in the Audit Type column, and then click Create. Define the audit policy.
The following figure shows an example audit policy AUDIT ALL STATEMENTS BY DBA_DESC BY ACCESS for the secured target dbtest:

4. Go to the **Secured Target** pane, select **Statement** and then click **Provision**.

5. Select the **Provision** option, and provide the **Secured Target Database User Name** (audituser) and **Password** (<audituser_password>). Then click **Provision**.
Monitor Database Activity in Oracle AVDF

Continuing from the previous example, all the SQL statements by user DBA_DEBRA on the dbtest secured target running on the DBaaS instance are audited by the Audit Vault Server. You can view all the activity in the Oracle AVDF console.

1. Log in to the Oracle AVDF console as auditor.

2. On the Report tab, click All Activity.

The following figure shows all statements issued by the DBA_DEBRA user on the dbtest secured target. Timestamp, status, user, SQL command, and other information are shown for each activity.

Oracle Database Auditing and Oracle AVDF Best Practices

Use the following best practices for Oracle Database security and compliance auditing with Oracle AVDF.

Use VCN Security Lists to Firewall the Oracle AVDF VM

You can use VCN security lists to allow network connections to the Oracle AVDF VM only from authorized database instances in the VCN.
Know the Performance Impact of Auditing on the DBaaS Instance

Audit Vault Agents run on the DBaaS instance to read Oracle Database audit trails and copy records to the Audit Vault Server. Oracle AVDF employs the following collectors:

- DBAUD (to read from database audit tables)
- OSAUD (to read from OS files)
- REDO (to read REDO logs)

The more events collected by the agents, the greater the load on the DBaaS instance. In general, collecting 100 audit records per second imposes about 2-3 percent CPU overhead for DBAUD and OSAUD, and about 6 percent overhead for REDO. This collector overhead does not include the performance overhead imposed by standard auditing and FGA.

The more events audited per second, the greater the performance overhead. So, we recommend caution in deciding which events to audit.

Know Your Audit Vault Server Storage Requirements

Audit records consume space on the Audit Vault Server. On average, one million audit records require about 900 MB of disk space. Depending on the number of audit records generated per day and their retention period, it is necessary to allocate enough disk space (both block volume and Oracle AVDF virtual disk sizes). Failure to provision an appropriate-sized disk could lead to an Oracle AVDF VM crash and loss of all collected audit records.

Archive Audit Records to Oracle Cloud Infrastructure Object Storage

We recommend setting up periodic archiving of Oracle AVDF audit records to your Oracle Cloud Infrastructure Object Storage bucket. Oracle AVDF allows archiving using SCP (secure copy) to an IP-addressable host (among other options such as NFS). We recommend archiving Oracle AVDF audit records on the BYOH BM host, and transferring them by using a script from the BM host to your Object Storage bucket. Refer to the Oracle AVDF documentation for more information about setting up periodic archiving.

Configure for High Availability

You can configure a pair of Audit Vault Servers, one as primary and one as secondary. Audit records in the primary are automatically synchronized to the secondary. In the scenario described in this paper, we recommend that primary and secondary Oracle AVDF VMs be installed on two separate BYOH BM instances for maximum availability. Refer to the Oracle AVDF documentation for a high-availability (HA) setup.
Enable SSH Access to the Audit Vault Server

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

Conclusion

This white paper presents a solution for deploying Oracle Audit Vault and Database Firewall on Oracle Cloud Infrastructure for auditing Oracle Cloud Infrastructure DBaaS instances for improved security and compliance. Along with Oracle Cloud Infrastructure DBaaS API audit logs, the Oracle AVDF auditing provides comprehensive audit logging and monitoring capability for DBaaS instances. This customer-managed solution requires customers to deploy a BYOH on their BM instance in order to run the Oracle AVDF security appliance.

FAQ

Why can’t the Oracle AVDF appliance be run directly on an Oracle Cloud Infrastructure instance?

Oracle AVDF is built as a security appliance that includes auditing software packaged with an Oracle Linux operating system. Installing Oracle AVDF requires wiping the boot disk and installing the Oracle AVDF ISO. To install Oracle AVDF directly on Oracle Cloud Infrastructure instances would require customizing the Oracle AVDF ISO to boot on Oracle Cloud Infrastructure instances. At present, we do not have an Oracle AVDF ISO image available.

Is this a managed Oracle AVDF service?

This white paper enumerates a customer-managed solution to meet a customer’s critical security and compliance requirements by using a proven product used on-premises by enterprises. The customer is responsible for deploying and managing the Oracle AVDF appliance in their VCN. Please contact Oracle Cloud Infrastructure to let us know your interest in a managed Oracle AVDF service.
Appendix

Enable SR-IOV on a BM 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 BM 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 VFDEVICE_NAME down
   ip link set VFDEVICE_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

```xml
<?xml version="1.0"?>
<interface type='direct'>
    <mac address='\$VNIC_MAC'/>  
    <source dev='\$VLAN_DEVICE_NAME' mode='passthrough'/>  
    <model type='e1000'/>  
</interface>
```
Deploying Oracle Audit Vault and Database Firewall in Oracle Cloud Infrastructure

April 2018

Authors: Nachiketh Potlapally, George Csaba
Contributing Author: Michael Mesoros