Oracle® NoSQL Database

Quick Start to KVLite

Release 24.3
E87605-32
November 2024

Oracle NoSQL Database Quick Start to KVLite, Release 24.3

E87605-32

Contents

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

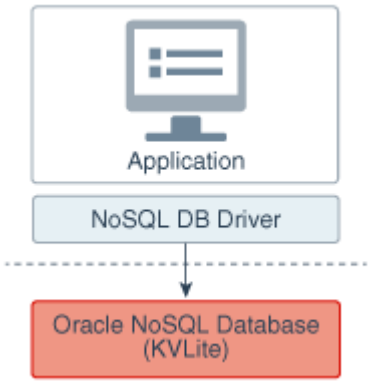
Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Introduction

The Oracle NoSQL Database is a scalable, distributed NoSQL database, designed to provide highly reliable, flexible and available data management across a configurable set of storage nodes. It consists of two parts - a NoSQL DB Driver and a collection of storage nodes called the data store. The NoSQL DB Driver is an intelligent driver that transparently handles all the core operations of Oracle NoSQL Database, and the data store consists of storage nodes.

KVLite is a simplified version of the Oracle NoSQL Database. It provides a single storage node, single shard store, that is not replicated. It runs in a single process without requiring any administrative interface. You configure, start, and stop KVLite using a command line interface.



Note: KVLite is intended for use by application developers who want to develop and unit test their Oracle NoSQL Database applications. It can be used as a development platform for developers to get familiar with Oracle NoSQL APIs, and test different ways of interacting with these APIs. KVLite runs on a single system. It is not intended for production deployment, or for performance measurements.

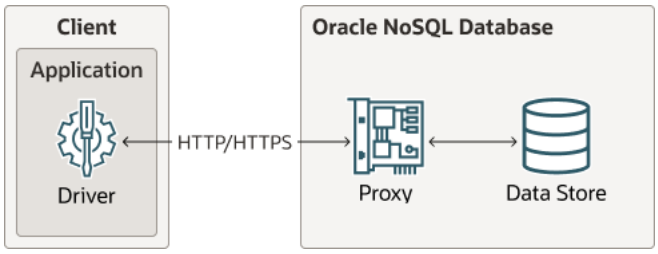
Also, KVLite is secure by default. If you want to run KVLite in non-secure mode, you will have to explicitly provide parameters to disable security while installing KVLite as demonstrated in this guide.

This Quick Start guide demonstrates how to perform the following tasks:

1. About the Oracle NoSQL Database Proxy
2. Install KVLite
3. Start KVLite
4. Verify your Installation
5. Stop and Restart KVLite

About the Oracle NoSQL Database Proxy

The Oracle NoSQL Database Proxy is a middle-tier component that lets the Oracle NoSQL Database SDK communicate with the Oracle NoSQL Database (kvlite configuration).



The Oracle NoSQL Database drivers are available in various programming languages that are used in the client application. The Oracle NoSQL Database Proxy is a server that accepts requests from the client application and processes them using the Oracle NoSQL Database. The JAR file (httpproxy.jar) for the Oracle NoSQL Database Proxy is included in the Enterprise Edition distribution and the Community Edition distribution of Oracle NoSQL Database that you downloaded. You can download the JAR for the Oracle NoSQL Database Proxy from the Oracle Technology Network.

After you start KVLite, you must run the following command to start up the proxy.

- For a non-secure kvlite:

```
java -jar lib/httpproxy.jar \
-storeName <kvstore_name> \
-helperHosts <kvstore_helper_host> \
[-hostname <proxy_host>] \
[-httpPort <proxy_http_port>]
```

- For a secure kvlite:

```
java -jar lib/httpproxy.jar \
-storeName <kvstore_name> \
-helperHosts <kvstore_helper_host> \
[-hostname <proxy_host>] \
[-httpsPort <proxy_https_port>] \
-storeSecurityFile proxy/proxy.login \
-sslCertificate certificate.pem \
-sslPrivateKey key-pkcs8.pem \
-sslPrivateKeyPass <privatekey_password> \
[-verbose true]
```

For more information, see Oracle NoSQL Database Proxy.

Install KVLite

KVLite is bundled with the Oracle NoSQL Database software. To install KVLite, perform the following:

1. Download the tar.gz or .zip file (depending on your operating system) from Oracle Technology Network.
2. In a Linux operating system, use `gunzip` and `tar` commands to extract the `.tar.gz` package (or use `unzip` command if you downloaded the `.zip` package). Oracle NoSQL Database version 24.1.11 Community Edition is used in this example. The actual package names and directory names will change, depending upon the release version you are using, and whether you are using Community Edition (CE) or Enterprise Edition (EE).

   Also, make sure you meet the following requirements to run KVLite:

   - Install Java version 17 in your machine.
   - Maintain a minimum disk space of 5GB.

   ```
   $ gunzip kv-ce-24.1.11.tar.gz
   $ tar xvf kv-ce-24.1.11.tar
   ```

Start KVLite

Perform the following steps to start a KVLite instance:

You could start KVLite in secure mode (the default option) or non-secure mode. If the KVLite is started in a secure mode, you should also configure a secure proxy and start it. Similarly if the KVLite is started in a non-secure mode, a non-secure proxy needs to be configured and started.

The following environmental variables have to be set before invoking the script to start KVLite as these variables are used in the scripts.

Table Environment variables

Name of the variable	Description	Sample values
KV_HOST	Identifies a host name associated with the node on which the script to configure kvlite is run.	localhost or the name of the machine
KV_PROXY_PORT	The TCP/IP port on which proxy should be contacted.	Non-Secure proxy: Use 80 if root privilege is there, else 8080 Secure proxy: Use 443 if root privilege is there, else 8443
KV_HARANGE	A range of free ports that the Replication Nodes and Admins use to communicate among themselves. These ports should be sequential.	5010-5020
KV_SERVICERANGE	A range of ports that may be used for communication among administrative services running on a Storage Node and its managed services.	5021-5049
KV_PORT	The TCP/IP port on which Oracle NoSQL Database should be contacted. Sometimes referred to as the registry port.	5000
KV_ADMIN_PORT	The TCP/IP port on which the admin service should be started.	5999

Start KVLite in secure mode:

- Download and extract the zipped file (start_securekvlite.zip) into the directory where you extracted kvlite. This zipped file contains the scripts needed for starting KVLite in secure mode.
- Invoke the script (`start_kvlite.sh`) to start KVLite in secure mode.

```
$/bin/bash start_kvlite.sh
```

You get an output as shown below:

```
Waiting for kvstore to start...
Waiting for kvstore to start...
Generated password for user admin:*********
```

```
User login file: kvroot/security/user.security
Created new kvlite store with args:
-root kvroot -store kvstore -host localhost -port 5000 -admin-web-port
5999 -secure-config enable
```

- Configure proxy for the secure kvlite: You need to do the following to configure a secure proxy:
  - Create a user (`proxy_user`) as the proxy needs an identity to connect to the secure data store.
  - Create a new password file to store the credentials needed to login as the proxy user.
  - Create a login file `proxy.login` for the proxy user.
  - Create self-signed certificates that can be used to securely connect to the Oracle NoSQL Database Proxy.

  Invoke the script (`setup-http-proxy-sec.sh`) to configure proxy in secure mode.

  ```
  $ /bin/bash setup-http-proxy-sec.sh
  ```

  You get an output as shown below:

  ```
  Creating password
  Creating USER proxy_user
  Oct 01, 2024 6:48:23 AM org.jline.utils.Log logr
  WARNING: Unable to create a system terminal, creating a dumb terminal
  (enable debug logging for more information)
  sql-> Statement completed successfully
  sql-> Creating proxy secfiles
  Created
  Secret created
  Creating certificate
  Generating a RSA private key
  ..................................................................................
  ..................................................................................
  ..................................+++
  ..........................................+++
  writing new private key to 'kvroot/proxy/key.pem'
  -----
  Certificate was added to keystore
  ```

- Use the script (`start_proxy.sh`) to start the proxy for a secure data store:

  ```
  /bin/bash start_proxy.sh
  ```

  You get an output as shown below:

  ```
  Starting Proxy
  Proxy creating SSL channel
  Proxy started:
  async=false
  helperHosts=localhost:5000
  httpPort=0
  httpsPort=8443
  idleReadTimeout=0
  ```

```
kvConsistency=NONE_REQUIRED
kvDurability=COMMIT_NO_SYNC
kvRequestTimeout=-1
monitorStatsEnabled=false
numAcceptThreads=3
numRequestThreads=32
proxyType=KVPROXY
sslCertificate=kvroot/proxy/certificate.pem
sslPrivateKey=kvroot/proxy/key-pkcs8.pem
sslPrivateKeyPass=iTO6aUCnh9XdsgkxFig=
sslProtocols=TLSv1.2,TLSv1.1,TLSv1
storeName=kvstore
storeSecurityFile=kvroot/proxy/proxy.login
verbose=true
proxyVersion=null
kvclientVersion=24.1.11
```

Start KVLite in non-secure mode:

- Download and extract the zipped file (start_nonsecurekvlite.zip) into the directory where you extracted kvlite. This zipped file contains the scripts needed for starting KVLite in non-secure mode.
- Invoke the script (`start_nonsecure_kvlite.sh`) to start KVLite in non-secure mode.

  ```
  $/bin/bash start_nonsecure_kvlite.sh
  ```

  You get an output as shown below:

  ```
  Created new kvlite store with args: -root kvroot -store kvstore -host
  localhost
  -port 5000 -admin-web-port 5999 - secure-config disable
  ```

- Use the script (`start_nonsecure_proxy.sh`) to start the proxy for a non-secure data store:

  ```
  $/bin/bash start_nonsecure_proxy.sh
  ```

  You get an output as shown below:

  ```
  Starting Proxy
  Proxy started:
  async=false
  helperHosts=localhost:5000
  httpPort=8080
  httpsPort=0
  idleReadTimeout=0
  kvConsistency=NONE_REQUIRED
  kvDurability=COMMIT_NO_SYNC
  kvRequestTimeout=-1
  monitorStatsEnabled=false
  numAcceptThreads=3
  numRequestThreads=32
  proxyType=KVPROXY
  sslProtocols=TLSv1.2,TLSv1.1,TLSv1
  storeName=kvstore
  ```

```
verbose=true
proxyVersion=null
kvclientVersion=24.1.11
```

Verify your Installation

You can verify your installation and ensure that KVLite is running.

- Start a new shell and run the following command:

  ```
  $ jps -m
  ```

  Your list of processes running will include the kvlite(kvstore.jar) and the proxy that you configured and started. You get an output similar to the one shown below.

  ```
  3523439 Jps -m
  3500313 httpproxy.jar -helperHosts localhost:5000 -storeName kvstore -
  httpPort 8080 -verbose true
  3499946 kvstore.jar kvlite -secure-config disable -root kvroot -host
  localhost -port 5000 -admin-web-port 5999 -harange 5010,5020 -servicerange
  5021,5049 -storagedirsizegb 10
  ```

  You can also ping your KVLite instance to see if the KVLite is configured and started successfully.

  For secure KVLite:

  ```
  java -Xmx64m -Xms64m -jar lib/kvstore.jar ping -host localhost \
  -port 5000 -security kvroot/security/user.security
  ```

  For non-secure KVLlite:

  ```
  java -Xmx64m -Xms64m -jar lib/kvstore.jar ping -host localhost -port 5000
  ```

  You get an output similar to the one shown below:

  ```
  Pinging components of store kvstore based upon topology sequence #14
  10 partitions and 1 storage nodes
  Time: 2024-10-21 05:23:45 UTC   Version: 24.1.11
  Shard Status: healthy: 1 writable-degraded: 0 read-only: 0 offline: 0
  total: 1
  Admin Status: healthy
  Zone [name=KVLite id=zn1 type=PRIMARY allowArbiters=false
  masterAffinity=false]
  RN Status: online: 1 read-only: 0 offline: 0
  Storage Node [sn1] on localhost: 5000
  Zone: [name=KVLite id=zn1 type=PRIMARY allowArbiters=false
  masterAffinity=false]
  Status: RUNNING   Ver: 24.1.11 2024-10-21 05:24:19 UTC
  Build id: e0c93c1f1395 Edition: Enterprise    isMasterBalanced: true
  serviceStartTime: 2024-10-21 05:24:21 UTC
  Admin [admin1] Status: RUNNING,MASTER  serviceStartTime: 2024-10-21
  05:24:23 UTC
  stateChangeTime: 2024-10-21 05:25:04 UTC   availableStorageSize: 2 GB
  Rep Node [rg1-rn1] Status: RUNNING,MASTER sequenceNumber: 85 haPort: 5011
  availableStorageSize: 9 GB storageType: HD serviceStartTime: 2024-10-21
  ```

```
05:24:05 UTC
stateChangeTime: 2024-10-21 05:24:07 UTC
```

Stop and Restart KVLite

To stop and restart KVLite, perform the following steps:

To stop KVLite, use `Ctrl C (^C)` from within the shell where KVLite is running.

To restart the process, run the KVLite utility without any command line options. Do this even if you provided non-standard options when you first started KVLite. This is because KVLite remembers information such as the port value and the store name in between run times. You cannot change these values by using the command line options.

```
$ java -Xmx64m -Xms64m -jar KVHOME/lib/kvstore.jar kvlite
```

If you want to start over with different options than you initially specified, delete the `KVROOT` directory (`./kvroot`), and then rerun the KVLite utility with the options you need. Refer to Start KVLite.

Note: If you decide to start over, all your previous data will be lost.