

Oracle NoSQL Database Cheat Sheet

Oracle NoSQL Database is a distributed, shared-nothing, non-relational database that provides large-scale storage and access to key/value, JSON, and tabular data. It can deliver predictable, low latencies to simple queries at any scale and is designed from the ground up for high availability. For complete documentation, see [Oracle NoSQL Database Documentation](#).

QUICK START TO KVLITE

KVLite is bundled with the [Oracle NoSQL Database software](#).

NOTE: You need a VM instance to run these commands. Download the .zip file and extract the package.

Go to the directory where you unpacked the content.

Secure mode:

```
java -Xmx64m -Xms64m -jar lib/kvstore.jar
kvlite [stagedirsizegb N]
```

Non-secure mode:

```
java -jar lib/kvstore.jar kvlite
[-stagedirsizegb N] -secure-config disable
```

See all the supported parameters in [KVLite documentation](#).

QUICK START TO KVLITE IN A CONTAINER

Pull the image directly from the GitHub Container Registry.

```
docker pull ghcr.io/oracle/nosql:latest-ce
```

```
docker tag ghcr.io/oracle/nosql:latest-ce
oracle/nosql:ce
```

Run Oracle NoSQL Database in a Container

```
docker run -d --name=kvlite --hostname=kvlite
--env KV_PROXY_PORT=8080 -p 8080:8080
oracle/nosql:ce
```

Check the Status of the Container

```
docker ps
```

Validate your deployment

```
docker exec -ti kvlite java -jar
lib/kvstore.jar ping -host kvlite -port 5000
```

INSTALL DATA STORE

Download the [Oracle NoSQL Database software](#) and extract the package.

```
unzip kv-ee-<latest release>.zip
```

Set the appropriate values for \$KVHOME and \$KVROOT.

```
export $KVHOME=$HOME/nosql/kv-<latest release>
export $KVROOT=$KVHOME/kvroot
```

Verify the installation

```
java -Xmx64m -Xms64m -jar
$KVHOME/lib/kvclient.jar
```

Verify that the Oracle NoSQL Database processes are running

```
jps -m
```

CONFIGURE DATA STORE

Create the initial bootconfig configuration file on a Storage Node.

Using password file:

```
java -jar $KVHOME/lib/kvstore.jar
makebootconfig -root kvroot -host $KVHOST -
harange 5005,5050 -port 5000 -store-security
configure -pwdmgr pwdfile -capacity 1 -
stagedir <storage path> -stagedirsize
<n>_GB
```

Using wallet:

```
java -jar $KVHOME/lib/kvstore.jar
makebootconfig -root kvroot -host $KVHOST -
harange 5005,5050 -port 5000 -store-security
configure -pwdmgr wallet -capacity 1 -
stagedir <storage path> -stagedirsize
<n>_GB
```

NOTE: For non-secure data store, include store-security parameter as none.

Start the Oracle NoSQL Database Storage Node Agent (SNA)

```
nohup java -Xmx64m -Xms64m \
-jar $KVHOME/lib/kvstore.jar start -root
$KVROOT &
```

Check if the SNA is running on the Storage Node

```
ssh <Storage node>
java -Xmx64m -Xms64m -jar
$KVHOME/lib/kvstore.jar ping -host $KVHOST -
port 5000 -security
$KVROOT/security/client.security
```

For non-secure data store, remove -security parameter

Stop a Storage Node

```
java -Xmx64m -Xms64m -jar
$KVHOME/lib/kvstore.jar stop -root $KVROOT
```

Clear KVROOT directory

```
rm -rf $KVROOT
```

See [Configuring a single region data store](#).

ADMIN CLI COMMANDS

Start the [Admin Command Line Interface \(CLI\)](#) utility

```
java -jar $KVHOME/lib/kvstore.jar runadmin -
host $KVHOST -port 5000 -security
$KVROOT/security/client.security
```

Name your data store

```
configure -name <store name>
```

Create a Zone

```
plan deploy-zone -name <zone name>
-rf <replication factor> [-type [primary |
secondary]] [-arbiters | -no-arbiters ]
[-json ] [-master-affinity | -no-master-
affinity] [-plan-name <name>] [-wait] [-
noexecute] [-force]
```

View Store topology

```
show topology
```

Deploy a Storage Node (<sn>)

```
plan deploy-sn -zn <zone name> -host <sn> -
port 5000 -wait
```

Create admin process on the Storage Node

```
plan deploy-admin -sn <sn> -wait
```

Create a Storage Node Pool

```
pool create -name <pool name>
pool join -name <pool name> -sn <sn>
```

Calculate number of partitions

```
(Total number of disks hosted by the Storage
Nodes / Replication Factor) * 10
```

Create Topology name

```
topology create -name <topology name> -pool
<pool name> -partitions <partition count>
```

Create and Deploy Replication Nodes

```
plan deploy-topology -name <topology name> -
wait
```

Check plan status

```
show plans
```

Verify data store

```
verify configuration
```

```
ping <sn>
```

Create an admin user

```
execute 'CREATE USER root IDENTIFIED BY
\'password\' ADMIN'
```

Create password file to store credentials for admin user (root)

Run from an Oracle NoSQL Database instance.

```
java -Xmx64m -Xms64m -jar
$KVHOME/lib/kvstore.jar securityconfig pwdfile
create -file $KVROOT/security/login.passwd
java -Xmx64m -Xms64m -jar
$KVHOME/lib/kvstore.jar securityconfig pwdfile
secret -file $KVROOT/security/login.passwd -
set -alias root
<Supply the password for admin user and re-
enter the password to confirm>
```

Create the wallet to store credentials for the admin user (root)

```
java -Xmx64m -Xms64m -jar
KVHOME/lib/kvstore.jar securityconfig wallet
create -dir $KVROOT/security/login.wallet
java -Xmx64m -Xms64m -jar
KVHOME/lib/kvstore.jar securityconfig wallet
secret -dir $KVROOT/security/login.wallet -set
-alias root
<Supply the secret value for admin user and
re-enter the value to confirm>
```

START SQL SHELL

Run from an Oracle NoSQL Database instance.

```
java -jar $KVHOME/lib/sql.jar -helper-hosts
node01:5000 -store kvstore
```

See all the supported parameters in [Running the SQL Shell](#).

USER MANAGEMENT

Run the commands from SQL shell.

Create user

```
CREATE USER user_name (IDENTIFIED EXTERNALLY |
IDENTIFIED BY password [PASSWORD EXPIRE |
PASSWORD LIFETIME duration_time_unit])
[ACCOUNT LOCK|UNLOCK] [ADMIN]
```

Manage user-defined roles

```
CREATE ROLE role_name
DROP ROLE role_name
SHOW ROLES
```

Oracle NoSQL Database Cheat Sheet

Grant privileges to a user-defined role

```
GRANT { grant_roles | grant_system_privileges |
grant_object_privileges }
grant_roles ::= role [, role]... TO { USER
user | ROLE role }
grant_system_privileges ::= (system_privilege
| ALL PRIVILEGES)[,{system_privilege | ALL
PRIVILEGES}]...TO role
grant_object_privileges ::=(object_privileges|
ALL [PRIVILEGES])[,{object_privileges| ALL
[PRIVILEGES]}]...ON object TO role
```

Examples:

```
CREATE ROLE manager
GRANT WRITE_ANY TO manager
CREATE ROLE employee
GRANT READ_ANY TO employee
GRANT employee TO ROLE manager
GRANT manager TO USER Kate
GRANT READ_TABLE on T1 TO Kate
```

ORACLE NOSQL DATABASE PROXY

Run from an Oracle NoSQL Database instance.

Secure mode:

```
java -jar $KVHOME/lib/httpproxy.jar -storeName
<store name> -helperHosts $KVHOST [-hostname
<proxy_host>] [-httpsPort <proxy_https_port>]
-storeSecurityFile
$KVROOT/security/proxy.login -sslCertificate
certificate.pem -sslPrivateKey key-pkcs8.pem -
sslPrivateKeyPass <privatekey_password> [-
verbose true]
```

Non-secure mode:

```
java -jar $KVHOME/lib/httpproxy.jar -storeName
<store name> -helperHosts $KVHOST [-hostname
<proxy_host>] [-httpPort <proxy_http_port>]
```

See all the supported parameters in [Configuring the Proxy](#).

ORACLE NOSQL DATABASE MIGRATOR

Download the Oracle NoSQL Database Migrator utility from the [Oracle NoSQL Downloads](#) page.

Identify the Source and Sink. See [Supported Sources and Sinks](#).

Prepare the configuration file (JSON format) with the identified Source and Sink details. See [Source](#)

Configuration Templates and Sink Configuration Templates.

Run the Migrator utility from the command prompt to perform migration by passing the configuration file:

```
./runMigrator --config <complete path to the
JSON config file>
```

Run the Migrator utility from the command prompt to generate the configuration file interactively and then perform migration:

```
./runMigrator
```
