ORACLE
NOSQL DATABASE

Oracle NoSQL Database on the Oracle Cloud Infrastructure

Quick Start White Paper | Version 1.0

ORACLE WHITE PAPER | MARCH 2020

ORACLE

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.

Questions or comments on this paper? Please email: oraclenosql-info_ww@oracle.com.

Table of Contents

"Oracle's NoSQL offers value to customers looking at ACID transactions; geodistributed data; application security with authentication and session-level SSL encryption; and integration with Oracle Database, Oracle Wallet, and Hadoop."

"Oracle NoSQL is a key-value database that delivers good performance, scale, security, and high availability capabilities."

SOURCE: THE FORRESTER WAVE™: BIG DATA NOSQL, Q3 2016

Oracle NoSQL Database on Oracle OCI Overview

Many software engineering organizations are faced with the challenge of building systems to handle extremely-high throughput (10s of thousands writes/sec) while maintaining low latency (< 10 msec). The Oracle NoSQL Database running on the Oracle Cloud Infrastructure (OCI) easily handles these types of workloads in a secure, highly-available environment.

The Oracle NoSQL Database is a best-in-class NoSQL database that provides:

- High-performance distributed read/write capability using share-nothing architecture
- Linear scalability with transparent load rebalancing when new nodes are added
- Kerberos authentication, table-level authorization, and secure client/server and server/server communication
- Highly-configurable ACID transaction model
- Table model with SQL-like query capability
- Simple to use API and multi-programming language drivers.

The OCI offers hourly metered Bare Metal instances and Virtual machine(VM) instances; VM instances offer compute resources in many shapes, from a single OCPU to 24 OCPUs, catering to a variety of workloads and software architectures. All Oracle Cloud Infrastructure VM shapes support remote block storage. The dense I/O shapes also offer up to 25.6 TBs of local NVMe SSD storage for applications requiring low latency, millions of IOPS, and high local storage capacity.

The OCI VM instances run on the same type of servers as bare metal instances. Leverage the same cloud-optimized hardware, firmware, software stack, and networking infrastructure to deliver unrivaled performance and strong isolation.

This Quick Start white paper is designed as a reference guide for deploying the Oracle NoSQL Database on the OCI platform. The following sections describe the preliminary setup of the OCI environment and then how to run the NoSQL cluster install scripts.

Assumptions

Consumers of this document should –

- Be familiar with the fundamentals of the Compute Service in Oracle Cloud Infrastructure
- The OCI walkthrough is highly is highly recommended if this is the first time you have used the platform
- Have a basic understanding of Oracle NoSQL
  - Oracle NoSQL Database Concepts Manuals, select a release, and take a look at the Concepts manual

Planning an Oracle NoSQL Database Deployment on OCI

A minimal configuration for the Oracle NoSQL Database typically consists of a 3-node server cluster:

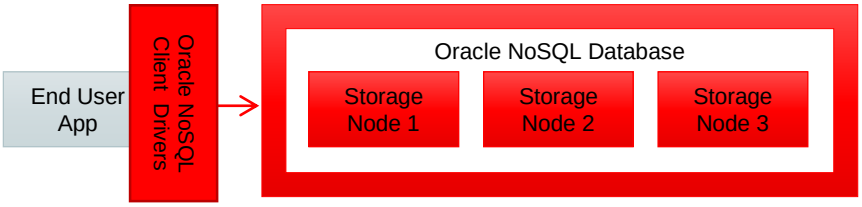

Figure-1 Oracle NoSQL Deployment

In this case, the backend database typically requires large amounts of persistent storage. For this paper, we are using the DenseIO virtual machine (VM) instances. The DenseIO VM instances provide high-performance with large local Non-Volatile Memory Express(NVMe) Solid State Drive storage and are suitable for high-performance database servers such as Oracle NoSQL Database.

Dense IO VMs are available in multiple configurations, or "shapes," and we are using the instances of the following shapes:

Oracle OCI Virtual Machine (VM) Shapes	
VM.DenseIO.2.8	6.4 TB of local NVMe SSD 8 OCPUs 120GB memory

For this quick start guide, we will walk through the steps of allocating 3 DenseIO VM NoSQL database server nodes. The database server nodes will be set up in 3 different Availability Domains (ADs) for enhanced reliability.

Outline of Steps to Get Oracle NoSQL Database Up and Running on Oracle OCI Virutual Machine

The following steps outline the process to get Oracle NoSQL Database up and running on Oracle OCI VM:

1. Set up OCI network and provision OCI instances
2. Acquire Oracle NoSQL Database License
3. Copy the Oracle NoSQL Database install scripts and Oracle NoSQL Database software to your local machine
4. Run the Oracle NoSQL Database cluster setup script and verify the install was successful
5. Run a simple test to verify basic operation
6. Verify the application server can communicate with the Oracle NoSQL Database.

The details for these steps are contained in the following sections.

Create the Required OCI Network Resources

For this part of the process you will need to go to Oracle Cloud Infrastructure console for e.g https://console.us-phoenix-1.oraclecloud.com to set up an account, if not already done, then navigate the OCI UI to do the following steps. Create a compartment using the steps outlined here. The Internet Gateway, Route Table, and Subnet configuration described below should be equivalent to the default settings when you create a new Virtual Cloud Network.

1. Create a new Virtual Cloud Network
    a. Name – Oracle_NoSQL_VCN
    b. CIDR block – 10.0.0.0/16
2. Create a new Internet Gateway
    a. Name - Oracle_NoSQL_IG
3. Create a new Route Table
    a. Name - Oracle_NoSQL_RT
    b. CIDR block – 0.0.0.0/0
    c. Target – Oracle_NoSQL_IG
4. Create Regional Subnet
    a. Name – Oracle_NoSQL_Subnet
    b. Subnet type - Regional
    c. CIDR block – 10.0.0.0/24
    d. Route table – Oracle_NoSQL_RT
    e. Subnet Access – Public or Private depending on if want to give access to the internet (rerfer to this guide)
5. Open Oracle NoSQL Ports
    a. Go to Networking > Virtual Cloud Networks for your compartment
    b. Click on your Virtual Cloud Network (Oracle_NoSQL_VCN)
    c. Click Security Lists then click Default Security List for Oracle_NoSQL_VCN
    d. Click Edit All Rules and add a Ingress rule with:
        - Source CIDR: 10.0.0.0/16
        - IP PROTOCOL: TCP
        - SOURCE PORT RANGE: All
        - DESTINATION PORT RANGE: 5000-5050

    Also, open the ICMP(Ping) and SSH ports by setting the security rules .

Create the Required OCI VM Compute Instances

Next, we create the VM compute instances. Please refer to this guide on instance creation

1. Create VM.DenseIO compute instance 1 to run Oracle NoSQL DB storage node 1
    a. Name – Oracle_NoSQL_DB_AD1_0
    b. Image – Oracle-Linux-7.7-2020.02.21-0
    c. Avalablity Domain – PHX-AD-1
    d. Instance Type – Virtual Machine
    e. Instance Shape - VM.DenseIO2.8
    f. Virtual Cloud Network Compartment - <Name of the compartment>
    g. Virtual Cloud Network – Oracle_NoSQL_VCN
    h. Subnet – Oracle_NoSQL_Subnet
    i. SSH Key – <public half of your key pair>
2. Create VM.DenseIO compute instance 2 to run Oracle NoSQL DB storage node 2
    a. Name – Oracle_NoSQL_DB_AD2_0
    b. Image – Oracle-Linux-7.7-2020.02.21-0
    c. Availablity Domain – PHX-AD-2
    d. Instance Type – Virtual Machine
    e. Instance Shape - VM.DenseIO2.8
    f. Virtual Cloud Network Compartment - <Name of the compartment>
    g. Virtual Cloud Network – Oracle_NoSQL_VCN
    h. Subnet – Oracle_NoSQL_Subnet
    i. SSH Key – <public half of your key pair>
3. Create VM.DenseIO compute instance 3 to run Oracle NoSQL DB storage node 3
    a. Name – Oracle_NoSQL_DB_AD3_0
    b. Image – Oracle-Linux-6.8-2017.01.09-0
    c. Availablity Domain – PHX-AD-3
    d. Instance Type – Virtual Machine
    e. Instance Shape - VM.DenseIO2.8
    f. Virtual Cloud Network Compartment - <Name of the compartment>
    g. Virtual Cloud Network – Oracle_NoSQL_VCN
    h. Subnet – Oracle_NoSQL_Subnet

i. SSH Key – <public half of your key pair>

For each instance note the public and private (RFC1918) IP addresses in the table below. These IPs can be found in the OCI UI on the Compute > Instances > Instance Details page.

Instance	Public IP	Private IP
Oracle_NoSQL_DB_AD1_0		
Oracle_NoSQL_DB_AD2_0		
Oracle_NoSQL_DB_AD3_0		

Oracle NoSQL Database Zone and Replication Architecture

The preceding compute instance setup has 3 OCI DenseIO VM instances (Storage Nodes), each having one 6.4TB NVMe drive. By default, the Oracle NoSQL Database uses a replication factor (RF) of 3, resulting in the following single-zone layout consisting of four shards with a Replication Factor of 3. The script creates a shard for every 30 GB of memory the system has, and since the DenseIO machine has a total of 120 GB memory, it would create (120/30= 4) shards.

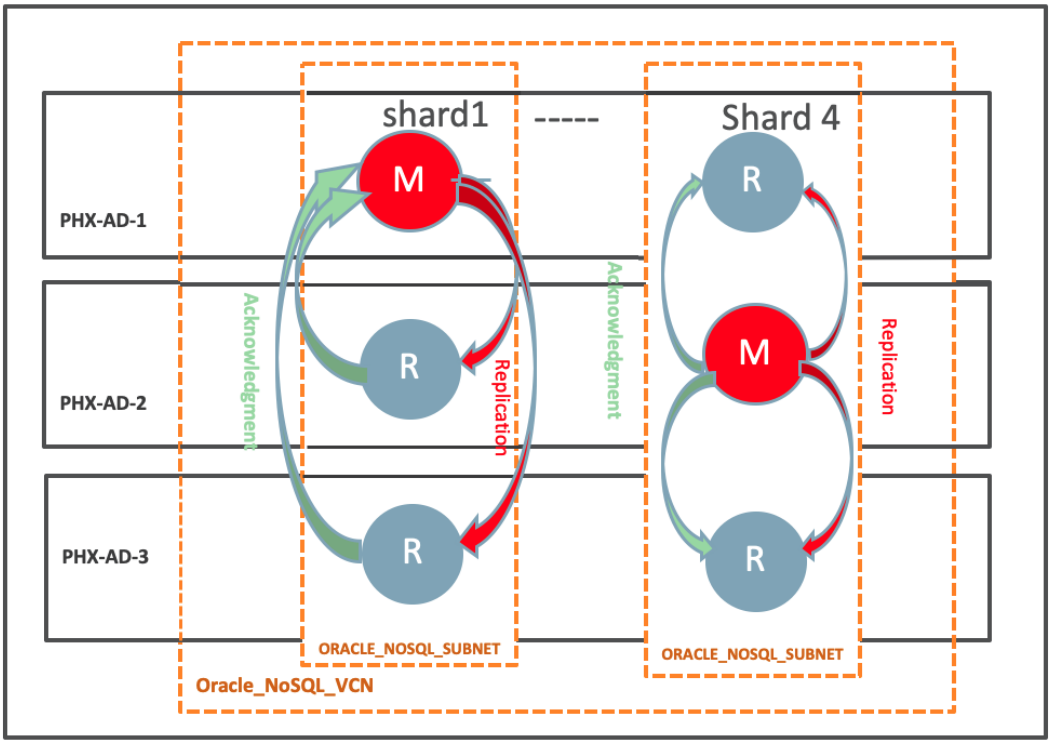

Figure-2 Oracle NoSQL Database with 4 shards and a replication factor of 3 spread across different ADs in PHX region

Installing the Oracle NoSQL Database on OCI

The following subsections describe the steps needed to install the Oracle NoSQL database on OCI.

Download Oracle NoSQL Database Software and Oracle JDK

Depending on which software license you own, you can download either Oracle NoSQL Database Enterprise Edition, Basic Edition or Community Edition as follows:

1) **Enterprise Edition** – Requires a commercial license from Oracle. This version can be downloaded for commercial use from the Oracle Software Download Cloud (edelivery.oracle.com).

2) **Community Edition** – This version can be downloaded from the Oracle Technology Network (OTN) downloads page at:

   http://www.oracle.com/technetwork/database/database-technologies/nosqldb/downloads/index.html

As an example, the Oracle NoSQL Database Community Edition can be downloaded to your local machine using the following command :

```
$ wget http://download.oracle.com/otn-pub/otn_software/nosql-database/kv-ce-18.1.27.tar.gz
```

Later, during the install process described below, this tar file will be uploaded to the OCI compute nodes. Keep the NoSQL .tar.gz file ready so that the install script can copy them to the OCI compute nodes.

Script to Install and Configure the Oracle NoSQL Database

The install script allows you to install and configure the Oracle NoSQL Database from your local machine. The scripts can be run from a BASH or OS X shell. The install script can be downloaded from:

https://github.com/oracle/nosql-examples/tree/master/cluster_setup

The script will do the following tasks:

- Ask for the path to the downloaded Oracle NoSQL kv tar.gz/.zip file
- Ask for all hostnames (or IP addresses) of target hosts, and ssh username/options
- Ask for the installation directory for NoSQL binaries/libraries
- Ask for the name of the store to be configured
- Ask for the data and log directory locations to store data/logs on each target host
- Ask for the starting port number for cluster communications
- Ask for the desired cluster data replication factor (if installing on more than one target host)
- Check ssh connections from your setup host to all target hosts
- Gather information from all target hosts (memory, cpu, etc.)
- Install java 8 if not already installed.
- Ask for (optional) security information (for secure store setup)
- Optionally check network connectivity on several ports between all target hosts
- Show installation parameters and ask for confirmation
- Copy .tar.gz/.zip files to all target hosts in parallel
- Set up each target host, one by one, to run NoSQL Storage Nodes

- Create helper scripts in KVHOME/scripts on the setup hosts for various operations
- Deploy cluster topology and start all cluster Replication Nodes (this can take a while)
- Setup HTTP Proxy.
- Run a simple test to verify essential operation
- Optionally run a longer extended test to get fundamental performance indicators
- Display store parameters for access to the new running store

Prerequisites for the cluster setup scripts

Before you can run the cluster_setup.sh script, you'll need to meet the following minimum requirements:

1. A setup host, from which you run the script, such as your laptop. The setup host must be capable of running bash in a unix-like environment (linux, macOS, cygwin, etc.)

2. One or more target host machines to where the NoSQL database engine will reside. These can be in OCI. They must be running Linux. It's best if they all have the same hardware resources (cpu, ram, disk).

3. Ssh access, with no passphrase, to the target host machines from the setup host. This may require some setup in ~/.ssh/config, or alternatively, ssh options can be given in the cluster_setup.sh script.

4. Java 8 or greater installed on the target host(s). The minimum version is 8. Any installed java version 8 or above works fine. For OCI, the script will take care of setting java.

5. One or more file system paths which specify the location(s) on the target hosts to be used to store NoSQL data. NVMe will result in the best performance, but any mounted drive (network or otherwise) will suffice. The amount of free space required varies greatly with the size and number of records you plan to store. In general, a minimum of 10GB free space is recommended.

6. (optional): Separate director(ies) for NoSQL logs, similar to #5.

7. A tar.gz or zip Oracle NoSQL kv installation file downloaded to the setup host (for example, kv-ee-19.5.13.tar.gz). Downloads are available from https://www.oracle.com/database/technologies/nosql-database-server-downloads.html

The Oracle NoSQL Database cluster install script lets a user set up a small cluster (1-10 machines) quickly, for use in proof-of-concepts, small on-premise installations, and cluster installations in cloud environments (OCI, AWS, Azure). It's intended to be run from a setup host that is not in the set of cluster hosts, typically a user laptop or other separate machine. The setup host should have a unix-like environment where bash can run. MacOS is fine.

You do not have to edit this script or pass any command-line parameters to it. It will prompt you for various inputs as it runs. All inputs will be saved, and rerunning the script will use the values from the previous run, so you don't have to retype everything if you run the script multiple times. You can change inputs when you rerun it.

Run the script as below. The script would prompt for host and cluster configuration spread across various input screen

```
$ ./cluster_setup.sh
```

Input Screen #1 Listing pre-requistes and if it should use cache values from previous runs

```
This script is intended to help set up a small Oracle NoSQL cluster for
testing and basic usage. The following requirements must already be met:

 1) ssh access to host machine(s)
 2) a downloaded Oracle NoSQL release .tar.gz or .zip file
 If not running in Oracle OCI environment, the following are also required:
    • Disk/nvme data drives set up on host machine(s)
    • Java 8+ installed on host machine(s)

Downloads can be obtained from
https://www.oracle.com/database/technologies/nosql-database-server-
downloads.html

Hit <enter> to continue:

Use cached values from last program run? (y/n) [y]: n
```

If you have successfully run the script, you can use the cached values from the last program run by typing '`y`' here. When you do so, the previously cached values are imported. Values from the previous run show in brackets ([]) at the end of each prompt.

Users can hit <`enter`> to use previous values or provide new input values.

Input Screen #2 Path of the local Oracle NoSQL Database release tar file

```
Enter path to the locally downloaded Oracle NoSQL release
tar.gz or zip file(ex: kv-ee-19.5.13.tar.gz)

tar.ga/.zip file: ~/Downloads/kv-19.5.19.zip
kv-19.5.19
```

Input Screen #3 Hostnames or IP addresses and SSH details of the OCI compute instances.

```
Enter hostnames or IP addresses of the hosts you want to run Oracle
NoSQL server processes on.
You need to have ssh access to these machines from where this
program is running. Separate hostnames/IPs with spaces.

hosts/IPs: <IP of Host1> <IP of Host2> <IP of Host3>

If ssh access to these hosts requires a specific username, enter
that here. If not, just hit <enter>.
ssh username: <username>

If ssh access to these hosts requires any other ssh options, enter the
ssh options, all on one line, here. For no options, just hit <enter>.


The most common use of ssh options is to supply an alternate keyfile
path, with "-i <path_to_key_file>", but any valid ssh options can
be entered here.


ssh options: -i/.ssh/ida_rsa
```

Note: IP Addresses of the Compute Instances and the ssh username are masked for security reasons.

Input Screen #4 Oracle NoSQL Database Server installation path

In the above step, the users can hit<enter> and use the default path [/opt/oracle/nosql] or provide a custom server installation path.

```
Oracle NoSQL programs installation:

Enter the desired install directory absolute path for Oracle NoSQL
programs on each host. Oracle suggests using /opt/oracle/nosql for
this value. JAR files, scripts, etc. will be located here. You will
be prompted in subsequent steps for the paths for NoSQL data and logs.

The directory will be created if it does not already exist.

Install path [/opt/oracle/nosql]:
```

Input Screen #5 Oracle NoSQL Database Store Name

```
Enter a name to use for the store. This should be a short, simple
name with only alphanumeric characters. It will be used throughout
the setup and later by connecting clients.

Store name [DefaultStore]:myStore
```

At this stage the script will do some basic sanity check about the target OCI hosts.

Input Screen #7 Format Disk and Mount NVMe drives

An OCI compute instance is delivered as a raw machine with Oracle Linux installed. It is necessary to do some initial hardware configuration, including formatting and mounting drives. The script takes care of that on the OCI instances automatically, as shown below:

```
Checking if all target hosts are running in Oracle OCI...yes

It appears that the target hosts have raw, unmounted NVMe drives.


Would you like this program to attempt to format and mount them
for use in the Oracle NoSQL cluster?
Format and mount NVMe drives on all target hosts? (y/n): y


Setting up NVMe drives on <IP of Host1>...
  formatting and mounting /dev/nvme0n1...
Setting up NVMe drives on <IP of Host2>...
  formatting and mounting /dev/nvme0n1...
Setting up NVMe drives on <IP of Host3>...
  formatting and mounting /dev/nvme0n1...

NVMe drives mounted successfully.

The following paths will be used for Oracle NoSQL data:

 /nosql/nvme0n1

Hit <enter> to continue:
```

Input Screen #8 Oracle NoSQL Database Input Ports

```
Enter the desired port number to use as a base for communications. Oracle
NoSQL will use a range of ports starting at this address (exact range will
be determined in later steps). Oracle recommends a default port number of
5000.
Enter starting port number [5000]: 5000
```

The script would automatically determine the port range.

Note: Oracle NoSQL Database uses a range of ports for its communication. The main starting port is used for client-server communication, and a range of additional ports are used for various server-server (and some client-server) features.

The actual range of ports needed depends on the capacity of each of the target hosts (machines with many NVMe drives and lots of RAM, for example, will use more ports) and whether security is enabled or not. A large installation with high capacity hosts may use 100 or more ports. Most installations typically use less than 30. The script determines the port range needed and can optionally test those port ranges across the target hosts.

If you are running Oracle OCI instances, this script can set up minimal firewall rules for you, if you select to do so. In all other environments (Azure or AWS), it does not make any attempt to change firewall rules or open ports in any way. Oracle recommends you work with your sysadmins to determine the proper steps/commands/processes to open ports in your environment.

Input Screen #10 Cluster Replication Factor

```
Enter the cluster replication factor (1, 2, or 3). A replication
factor of 1 is dangerous since no copies of the data will exist
and you will lose data if a host dies.

Replication factor of 3 is recommended for best data retention.
Replication factor [3]: 3
```

Replication Factor is the number of nodes belonging to a shard. It is recommended that you supply the number that matches the number of storage nodes. In our case, it is 3 since we have three storage nodes.

At this stage, the script will probe the compute instances for :

- Any existing Oracle NoSQL installation
- Internal(Private) IP address
- Hostname-to-IP Mapping
- Check SSH connection from your setup host to OCI compute instances
- Verify Java 8 or greater is installed. If not, it would install it on OCI automatically.
- Increases file/process limits for the OCI users to match the requirements
- Gather information from all target hosts (memory, cpu, etc.)

You should output like following for each of the storage nodes :

```
Collecting / verifying data on <IP of Host1>...
Checking for existing Oracle NoSQL installation...
getting internal IP address...
checking hostname-to-ip mapping...
Checking java installation...
Collecting system resources...
Increasing file/process limits for user "opc"...
```

Input Screen #11 Security details

```
This program will set up a secure store. If you do not want a secure
store, enter "n" at the prompt below.

Set up secure store (y/n) [y]: y
```

If you are setting a secure store, then the script prompts the input screen to set up the user passwords. Users can either generate the password automatically or manually supplying that to the script. If auto-generate is chosen, then the script shows the generated password. Copy the generated passwords and store them at some secure place for future access.

```
Secure setup will create two DB users: "admin" and "nosql".

"admin" is used for administrative tasks (topology, maintenance, etc.)

"nosql" is used for client-side data operations (insert, query, etc.)

Would you like this program to auto-generate passwords for the above
users? Enter "n" here to enter passwords manually.

auto-generate user passwords? (y/n) [y]: y

Passwords generated. Copy these to a safe location for later use.

admin password: G4)pJ2:lF2%o
nosql password: R0.bV7@eR6/q
```

After this, the script performs an optional network port connectivity between the OCI compute instances. It does that by installing the “netcat” utility to verify that. Checking the network connectivity is an optional step and users can choose to skip it by hitting <enter>

Input Screen #12 Network connectivity check(Optional)

Oracle NoSQL server requires network connectivity among the replication nodes. To accomplish this, firewall rules typically need to be added. In an OCI environment, this script can set up these rules for you. To do this, you must run the optional network connectivity check:

```
This program can optionally verify network port connectivity between
hosts. Doing so will use the "netcat" utility ("nc") to verify
connections.

If netcat is not installed, this program will attempt to
install it by calling "sudo yum install -y nc".

Would you like this program to try verifying network connectivity?
(y/n) [y]: y
Checking for "nc" on <Ip address of Host>...


Attempting to install "nc" on <Ip address of Host>...

Loaded plugins: langpacks, ulninfo
```

```
Resolving Dependencies
--> Running transaction check
---> Package nmap-ncat.x86_64 2:6.40-19.el7 will be installed
--> Finished Dependency Resolution


Dependencies Resolved
================================================================================
 Package          Arch          Version                 Repository         Size
================================================================================
Installing:
 nmap-ncat        x86_64        2:6.40-19.el7           ol7_latest        206 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 206 k
Installed size: 423 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1
  Verifying  : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1

Installed:
  nmap-ncat.x86_64 2:6.40-19.el7

Complete!
Checking for "nc" on 152.67.4.170...

Attempting to install "nc" on 152.67.4.170...

Loaded plugins: langpacks, ulninfo
Resolving Dependencies
--> Running transaction check
---> Package nmap-ncat.x86_64 2:6.40-19.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package          Arch          Version                 Repository         Size
================================================================================
Installing:
 nmap-ncat        x86_64        2:6.40-19.el7           ol7_latest        206 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 206 k
Installed size: 423 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1
  Verifying  : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1

Installed:
  nmap-ncat.x86_64 2:6.40-19.el7
```

```
Complete!
Checking for "nc" on 140.238.247.110...

Attempting to install "nc" on 140.238.247.110...

Loaded plugins: langpacks, ulninfo
Resolving Dependencies
--> Running transaction check
---> Package nmap-ncat.x86_64 2:6.40-19.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package          Arch          Version                 Repository         Size
================================================================================
Installing:
 nmap-ncat        x86_64        2:6.40-19.el7           ol7_latest        206 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 206 k
Installed size: 423 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1
  Verifying  : 2:nmap-ncat-6.40-19.el7.x86_64                               1/1

Installed:
  nmap-ncat.x86_64 2:6.40-19.el7

Complete!

Copying network validation scripts to hosts in parallel...

stop_listeners.sh.65300                                     100%  108     3.4KB/s
00:00
stop_listeners.sh.65300                                     100%  108     3.9KB/s
00:00
stop_listeners.sh.65300                                     100%  108     4.1KB/s
00:00
start_listeners.sh.65300                                    100%  600    23.1KB/s
00:00
start_listeners.sh.65300                                    100%  600    22.2KB/s
00:00
start_listeners.sh.65300                                    100%  600    22.6KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    57.5KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    56.7KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    57.1KB/s
00:00

Starting listeners on all hosts in parallel...
Running validation scripts on all hosts in parallel...
Ncat: No route to host.
Ncat: No route to host.
connectivity failed from <Ip address of Host> to <IP address of Host> on port 5000
Ncat: No route to host.
…..
```

……

Network connectivity tests failed.

```
Since the target hosts are running in an OCI environment, this
program can attempt to set up simple firewall rules to fix the
network connectivity issues.


NOTE: The firewall settings that this program can set up may violate
      your company security policies and potentially leave your
      instances vulnerable to security issues. If you are unsure
      about this, do not have this program modify firewall settings.
```

Have this program modify firewall settings? (y/n) [n]: y

```
Setting up firewall rules for ports 5000-5028 on <Ipaddess of Host1>...
success
success

Setting up firewall rules for ports 5000-5028 on <Ipaddress of Host2>...
success
success

Setting up firewall rules for ports 5000-5028 on <Ipaddress of Host3>
success
success

Copying network validation scripts to hosts in parallel...

stop_listeners.sh.65300                                     100%  108     4.2KB/s
00:00
stop_listeners.sh.65300                                     100%  108     4.2KB/s
00:00
stop_listeners.sh.65300                                     100%  108     3.8KB/s
00:00
start_listeners.sh.65300                                    100%  600    23.4KB/s
00:00
start_listeners.sh.65300                                    100%  600    22.8KB/s
00:00
start_listeners.sh.65300                                    100%  600    22.3KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    58.8KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    63.0KB/s
00:00
verify_connectivity.sh.65300                                100% 1631    60.2KB/s
00:00

Starting listeners on all hosts in parallel...
Running validation scripts on all hosts in parallel...
Stopping listeners and removing scripts all hosts in parallel...
```

Network connectivity tests passed

Confirmation Screen

```
Parameters that will be used for cluster setup:

  hosts:      <Instance1 IP Address> <Instance2 IP Address> <Instance IP Address>
  ipaddrs:    <Private IP for host1> <Private IP for host2><Private IP for host3>
  installdir: /opt/oracle/nosql
  mainport:   5000
  port_range: 5000-5028
  datapaths:  /opt/oracle/nosql/data
 logpaths:   /opt/oracle/nosql/logs
  security:   1
  repfactor:  3

Continue? (y/n) [y]Y
```

Once the node install script completes you should see the following mounted volumes on each of the compute nodes

The script copies the Oracle NoSQL Database binaries to each of the compute host in parallel and starts setting up the cluster based on the user inputs. This may take serveral minutes. Do not kill this program while it is running.

If the setup goes through successfully, then you should see the output along the lines of the following:

```
Pinging newly formed cluster...
Pinging components of store myStore based upon topology sequence #276
256 partitions and 3 storage nodes
Time: 2020-03-23 03:29:27 UTC   Version: 19.5.19
Shard Status: healthy:4 writable-degraded:0 read-only:0 offline:0 total:4
Admin Status: healthy
Zone [name=myStoreZone id=zn1 type=PRIMARY allowArbiters=false masterAffinity=false]   RN
Status: online:12 read-only:0 offline:0 maxDelayMillis:1 maxCatchupTimeSecs:0
Storage Node [sn1] on Oracle_NoSQL_DB_AD1_0:5000    Zone: [name=myStoreZone id=zn1
type=PRIMARY allowArbiters=false masterAffinity=false]    Status: RUNNING   Ver: 19.5.19
2020-01-27 07:16:20 UTC  Build id: 6783109c3c07 Edition: Enterprise
      Admin [admin1]          Status: RUNNING,MASTER
      Rep Node [rg1-rn1]      Status: RUNNING,REPLICA sequenceNumber:165 haPort:5002
available storage size:30 GB delayMillis:0 catchupTimeSecs:0
      Rep Node [rg2-rn1]      Status: RUNNING,REPLICA sequenceNumber:165 haPort:5003
available storage size:30 GB delayMillis:0 catchupTimeSecs:0
      Rep Node [rg3-rn1]      Status: RUNNING,MASTER sequenceNumber:164 haPort:5004
available storage size:30 GB
      Rep Node [rg4-rn1]      Status: RUNNING,MASTER sequenceNumber:163 haPort:5005
available storage size:30 GB
Storage Node [sn2] on Oracle_NoSQL_DB_AD2_0:5000    Zone: [name=myStoreZone id=zn1
type=PRIMARY allowArbiters=false masterAffinity=false]    Status: RUNNING   Ver: 19.5.19
2020-01-27 07:16:20 UTC  Build id: 6783109c3c07 Edition: Enterprise
      Admin [admin2]          Status: RUNNING,REPLICA
      Rep Node [rg1-rn2]      Status: RUNNING,REPLICA sequenceNumber:165 haPort:5002
available storage size:30 GB delayMillis:1 catchupTimeSecs:0
      Rep Node [rg2-rn2]      Status: RUNNING,MASTER sequenceNumber:165 haPort:5003
available storage size:30 GB
      Rep Node [rg3-rn2]      Status: RUNNING,REPLICA sequenceNumber:164 haPort:5004
available storage size:30 GB delayMillis:0 catchupTimeSecs:0
      Rep Node [rg4-rn2]      Status: RUNNING,REPLICA sequenceNumber:163 haPort:5005
available storage size:30 GB delayMillis:1 catchupTimeSecs:0
Storage Node [sn3] on Oracle_NoSQL_DB_AD3_0:5000    Zone: [name=myStoreZone id=zn1
type=PRIMARY allowArbiters=false masterAffinity=false]    Status: RUNNING   Ver: 19.5.19
2020-01-27 07:16:20 UTC  Build id: 6783109c3c07 Edition: Enterprise
      Admin [admin3]          Status: RUNNING,REPLICA
      Rep Node [rg1-rn3]      Status: RUNNING,MASTER sequenceNumber:165 haPort:5002
available storage size:30 GB
      Rep Node [rg2-rn3]      Status: RUNNING,REPLICA sequenceNumber:165 haPort:5003
available storage size:30 GB delayMillis:0 catchupTimeSecs:0
      Rep Node [rg3-rn3]      Status: RUNNING,REPLICA sequenceNumber:164 haPort:5004
available storage size:30 GB delayMillis:0 catchupTimeSecs:0
```

```
      Rep Node [rg4-rn3]      Status: RUNNING,REPLICA sequenceNumber:163 haPort:5005
available storage size:30 GB delayMillis:1 catchupTimeSecs:0

Cluster startup succeeded
```

The script also runs a basic test to insert a key/value pair and retrieve the same to verify things are working as expected. You should see the following ouput:

```
Running simple tests to verify cluster...

Statement completed successfully
{"NumRowsInserted":1}

1 row returned
{"id":12345,"data":"{\"test\":\"data\"}"}
1 row returned
Statement completed successfully
```

The script also sets up the HTTP proxy to be used with various language drivers.

```
Oracle recommends configuring a NoSQL httpproxy for use with various
language drivers (Python, Go, Node.js, etc). This program can set up
the proxy for you.

Would you like to set up an httpproxy? (y/n) [y]: y

Enter the port number desired to run Oracle NoSQL httpproxy on. This
port will be used by various language drivers to connect to the store.

httpproxy port: 8080
```

Additionally, the script also performs an (optional) extended running performance test. Since it is optional, the users have the choice to skip the test by hitting <enter>. In this test, the script creates a table `testData` and performs

GET and PUT Operations on that table and compute the 99th and 99.9th percentile latency as following:

```
Running extended test on <IP address of the ...
Creating store handle...
Created store successfully
Creating table "testData"...
Created table "testData" successfully
Performing get/put operations...
10% complete
20% complete
30% complete
40% complete
50% complete
60% complete
70% complete
80% complete
90% complete
Dropping table "testData"...
Dropped table "testData" successfully

Put operations:
  total: 9844
  operations per second: 864.30
  median latency: 1057us
```

```
  99th percentile latency: 3117us
  99.9th percentile latency: 4970us

Get operations:
  total: 9822
  operations per second: 2032.52
  median latency: 430us
  99th percentile latency: 1592us
  99.9th percentile latency: 3949us


Extended test ran successfull
```

Connecting to the Oracle NoSQL Database from the Application

Once the cluster setup is complete you should be able to log in (using account 'opc') and verify that it can communicate with the Oracle NoSQL Database using the following parameters:

```
helperHosts=<privateIP of instance1>:5000,<privateIP of instance2>:5000,<privateIP of
isntance3>:5000
storeName=myStore

Start/stop scripts for NoSQL kvstore are located on host(s) at:

   /opt/oracle/nosql/kvstore/scripts/start_kvstore.sh
   /opt/oracle/nosql/kvstore/scripts/stop_kvstore.sh

httpproxy(s) successfully started. Use the following parameters when
connecting to the proxy(s) using various language drivers:

   endpoint=http://<ip>:8080
   endpoint=http://<ip>:8080
   endpoint=http://<ip>:8080

Start/stop scripts for NoSQL httpproxy are located on host(s) at:

   /opt/oracle/nosql/proxy/scripts/start_proxy.sh
   /opt/oracle/nosql/proxy/scripts/stop_proxy.sh

Extended test script and TestClient.java example program are
located at:

   /opt/oracle/nosql/kvstore/examples/TestClient
```

Appendix A - Running Oracle NoSQL Database on Other clouds

Amazon EC2:

- Spin up one or more virtual machines – **Red Hat Linux**. NOTE: You will need something more powerful than a micro, otherwise requests might timeout
- Make sure that the port range 5000-5100 is open
    - Add a rule to the security group opening these ports
- Install java 8 on each VM (if a version 8 or greater is not already installed)
    - many EC2 instances have java7 installed. To get java 8, do:
    - sudo yum remove java
    - sudo yum install java-1.8.0-openjdk
- Download a release distribution of Oracle NoSQL Database to your setup host
    - https://www.oracle.com/database/technologies/nosql-database-server-downloads.html

- Download cluster_setup.sh and make it executable
    - curl 'https://raw.githubusercontent.com/oracle/nosql-examples/master/cluster_setup/cluster_setup.sh' > cluster_setup.sh
    - chmod +x ./cluster_setup.sh
- Run cluster_setup.sh. This will guide you through the setup of a NoSQL Cluster on your EC2 instances similar to what we saw earlier on the OCI instances
    - ./cluster_setup.sh

Users need to manually configure the firewall rules for EC2

Microsoft Azure:

- Spin up one or more virtual machines – **Oracle Linux 7.x**
- Make sure that the port range 5000-5100 is open.
    - Refer to your local sysadmins for guidance on this. An example of how to do this in some systems may be:
    - sudo iptables -I INPUT -p tcp -m tcp --dport 5000:5100 -j ACCEPT
- Install Java 8 on each the VMs (if a version 8 or greater is not already installed)
    - sudo yum install java
- Download a release distribution of Oracle NoSQL Database to your setup host
    - https://www.oracle.com/database/technologies/nosql-database-server-downloads.html
- Download cluster_setup.sh and make it executable
    - curl 'https://raw.githubusercontent.com/oracle/nosql-examples/master/cluster_setup/cluster_setup.sh' > cluster_setup.sh
    - chmod +x ./cluster_setup.sh
- Run cluster_setup.sh. This will guide you through the setup of a NoSQL Cluster on your Azure instances
    - ./cluster_setup.sh

Users need to manually configure the firewall rules for Azure.

Further Information about the Oracle NoSQL Database

Check out the following links for more information about the Oracle NoSQL Database

» http://www.oracle.com/technetwork/database/database-technologies/nosqldb/overview/index.html

» http://www.oracle.com/technetwork/database/database-technologies/nosqldb/documentation/index.html

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Oracle NoSQL on the Oracle Cloud Infrastructure
April 2020
Author: Anand Chandak anand.chandak@oracle.com) and Michael Brey (michael.brey@oracle.com)
Script Owner: John Connelly(john.connelly@oracle.com)