

**ORACLE®**

**NOSQL DATABASE**

# Deploying Oracle NoSQL Database on Oracle Cloud Infrastructure

Quick Start White Paper | February 2017 | Version 1.0

ORACLE WHITE PAPER | FEBRUARY 2017

**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](mailto:oraclenosql-info_ww@oracle.com).



ORACLE®

---



## Table of Contents

Oracle NoSQL Database on Oracle Bare Metal Cloud Overview	1
Assumptions	2
Planning an Oracle NoSQL Database Deployment on BMCS	2
Oracle NoSQL Database Zone and Replication Architecture	2
Step 1: Create the Required BMCS Network Resources	3
Step 2: Create the Required BMCS Compute Instances	4
Step 3: Download Oracle NoSQL Database Software and Oracle JDK	5
Step 4: Install and Configure the Oracle NoSQL Database	6
Step 5: Verify the Oracle NoSQL Database Installation	7
Step 6: Install Oracle NoSQL Database Components on the Application Server	8
Step 7: Connect to the Oracle NoSQL Database from the Application Server	8
Optional: Adding Multiple NoSQL Data Stores to BMCS Compute Instances	9
Appendix A: Running Oracle NoSQL Database on BMCS VMs	9
Appendix B: Script Details	9
File System Setup	9
Network Configuration	10
NTP Installation	11
Further Reading	11

---

---

*“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 Bare Metal Cloud Overview

Many software engineering organizations face the challenge of building systems to handle extremely high throughput (tens of thousands writes/sec) while maintaining low latency (< 10 msec). The Oracle NoSQL Database running on the Oracle Bare Metal Cloud Service (BMCS) 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

The Oracle BMCS offers hourly metered bare metal instances. By removing the hypervisor, Oracle can deliver better performance at a lower cost than traditional IaaS providers. In addition to compute unencumbered by a hypervisor, Oracle BMCS offers instances with up to 28 TBs of locally attached NVMe storage. Each 28 TB instance (nine NVMe storage units) is capable of over three million 4 kB IOPs/sec.

Instances in the Oracle BMCS are attached using a 10 GB non-blocking network with no oversubscription. Each node has access to the full performance of the hardware. There are no “noisy neighbors” or hypervisors to share resources with. Instances in the same region are always less than 1 ms from each other.

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

---

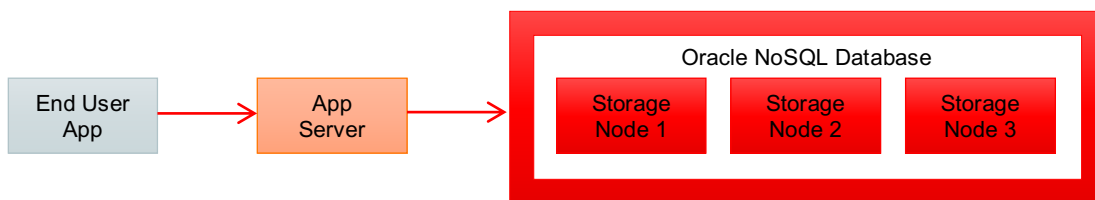
## Assumptions

To successfully use this white paper for your deployment, we assume the following:

- » You are familiar with the fundamentals of the Oracle BMCS. For more information, see the [Oracle BMCS documentation](#).
- » You have gone through the BMCS tutorials in the getting started topics. For more information, see [Launching Your First Linux Instance](#).
- » You have a basic understanding of Oracle NoSQL. For more information, see the [Oracle NoSQL Database Concepts Manual](#).

## Planning an Oracle NoSQL Database Deployment on BMCS

A minimal configuration for the Oracle NoSQL Database typically consists of a three-node server cluster, an end-user application, and a middle-tier application server between the end-user application and the Oracle NoSQL Database.



The middle tier can be either an application server or a web server. This is a resource that does not need much storage, but requires significant compute, memory and network resources. The backend database usually requires large amounts of persistent storage.

Because the resource types have different needs, you will choose from three instance shapes for the instances.

**Oracle BMCS Instance Shapes**

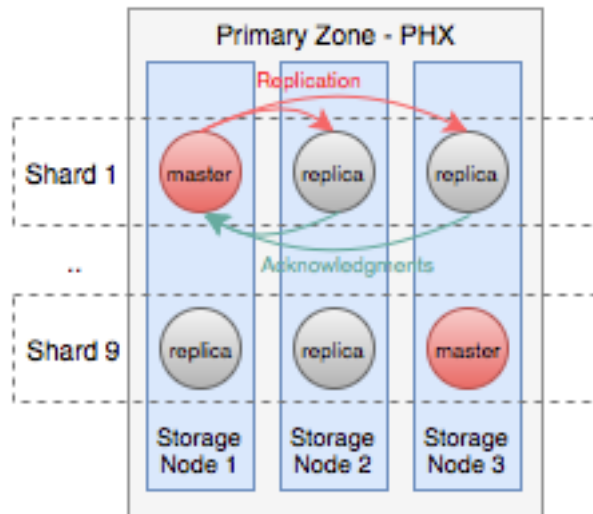
<b>BM.StandardIO1.36</b>	<b>BM.HighIO1.36</b>	<b>BM.DenseIO1.36</b>
32 GB local disk storage 36 cores 256 GB memory	12.8 TB local NVMe storage 36 cores 512 GB memory	28.8 TB local NVMe storage 36 cores 512 GB memory

Standard instances are suitable for middle-tier services (for example, web servers). High and dense shapes have local direct attached storage and are recommended for high-performance database servers.

In this white paper, we'll detail the steps of allocating a single standard shape middle-tier application server and three DenseIO NoSQL database server nodes. The database server nodes will be set up in three different availability domains for enhanced reliability.

## Oracle NoSQL Database Zone and Replication Architecture

This white paper details the set-up of a compute instance with three BMC DenseIO instances (storage nodes). Each instance has nine 3 TB NVMe drives. By default, the Oracle NoSQL Database uses a replication factor (RF) of three resulting in the following single-zone layout consisting of nine shards.



In this case there is a one-to-one mapping between shards and the number of NVMe drives on each machine.

To get Oracle NoSQL Database up and running on Oracle Bare Metal Cloud Services, perform the tasks listed in the following sections.

## Step 1: Create the Required BMCS Network Resources

Use the Oracle Bare Metal Cloud Services Console at <https://console.us-phoenix-1.oraclecloud.com> to set up an account. Once you have an account, log in to the Console and perform the following tasks.

1. Click Networking > Virtual Cloud Networks and create a new Virtual Cloud Network (VCN) with the following values:
  - o Name: Oracle\_NoSQL\_VCN
  - o CIDR block: 10.0.0.0/16
2. Click the VCN you created, click Internet Gateways, and create a new internet gateway with the name, Oracle\_NoSQL\_IG.
3. In your VCN, click Route Tables and create a route table with the following values:
  - o Name: Oracle\_NoSQL\_RT
  - o CIDR block: 0.0.0.0/0
  - o Target: Oracle\_NoSQL\_IG
4. In your VCN, click Subnets and create a new subnet, Subnet 1, with the following values:
  - o Name: AD1\_ Oracle\_NoSQL\_private
  - o AD: PHX-AD-1
  - o CIDR block: 10.0.0.0/24
  - o Route table: Oracle\_NoSQL\_RT
5. Create Subnet 2 with the following values:
  - o Name: AD2\_ Oracle\_NoSQL\_private
  - o AD: PHX-AD-2

- CIDR block: 10.0.1.0/24
  - Route table: Oracle\_NoSQL\_RT
6. Create Subnet 3 with the following values:
- Name: AD3\_Oracle\_NoSQL\_private
  - AD: PHX-AD-3
  - CIDR block: 10.0.2.0/24
  - Route table: Oracle\_NoSQL\_RT
7. Open Oracle NoSQL ports by going to **Networking > Virtual Cloud Networks** for your compartment.
- a. Click your Virtual Cloud Network (**Oracle\_NoSQL\_VCN**).
  - b. Click **Security Lists**.
  - c. Click Default Security List for Oracle\_NoSQL\_VCN.
  - d. Click **Edit All Rules** and add a rule with:
    - Source CIDR: 10.0.0.0/16
    - IP PROTOCOL: TCP
    - SOURCE PORT RANGE: All
    - DESTINATION PORT RANGE: 5000-5050

## Step 2: Create the Required BMCS Compute Instances

Use the Console to create four instances.

1. Create BM.DenseIO compute instance 1 to run Oracle NoSQL DB storage node 1:
    - Name: Oracle\_NoSQL\_DB\_AD1\_0
    - Image: Oracle-Linux-6.8-2017.01.09-0
    - Shape: BM.DenseIO1.512
    - AD: PHX-AD-1
    - Cloud Network: Oracle\_NoSQL\_VCN
    - Subnet: AD1\_Oracle\_NoSQL\_private
    - SSH Key: <public half of your key pair>
  2. Create BM.DenseIO compute instance 2 to run Oracle NoSQL DB storage node 2:
    - Name: Oracle\_NoSQL\_DB\_AD2\_0
    - Image: Oracle-Linux-6.8-2017.01.09-0
    - Shape: BM.DenseIO1.512
    - AD: PHX-AD-2
    - Cloud Network: Oracle\_NoSQL\_VCN
    - Subnet: AD2\_Oracle\_NoSQL\_private
    - SSH Key: <public half of your key pair>
  3. Create BM.DenseIO compute instance 3 to run Oracle NoSQL DB storage node 3:
    - Name: Oracle\_NoSQL\_DB\_AD3\_0
    - Image: Oracle-Linux-6.8-2017.01.09-0
    - Shape: BM.DenseIO1.512
-

- o **AD:** PHX-AD-3
- o **Cloud Network:** Oracle\_NoSQL\_VCN
- o **Subnet:** AD3\_Oracle\_NoSQL\_private
- o **SSH Key:** <public half of your key pair>

4. Create one BM.Standard compute instance to run the application server that will interface with your Oracle NoSQL Database:

- o **Name:** Oracle\_NoSQL\_AS\_AD1\_0
- o **Image:** Oracle-Linux-6.8-2017.01.09-0
- o **Shape:** BM.Standard01.512
- o **AD:** PHX-AD-1
- o **Cloud Network:** Oracle\_NoSQL\_VCN
- o **Subnet:** AD1\_Oracle\_NoSQL\_private
- o **SSH Key:** <public half of your key pair>

Note the public and private (RFC1918) IP addresses for each instance in the following table. You can find these addresses in the **Compute > Instances > Instance Details** page of the Console.

Instance	Public IP	Private IP
Oracle_NoSQL_DB_AD1_0		
Oracle_NoSQL_DB_AD2_0		
Oracle_NoSQL_DB_AD3_0		
Oracle_NoSQL_AS_AD1_0		

*Note: Communication between Oracle NoSQL Database nodes and application servers should be across the private IP address of the instances, not the public IPs. Using the public IP adds latency to the connection and limits the bandwidth. Using the RFC1918 private IP guarantees access to the full network bandwidth and the lowest possible latency. You can use the public IPs can be used for external access, for example, to SSH to the machines or as parameters to the Oracle NoSQL Database install scripts.*

### Step 3. Download Oracle NoSQL Database Software and Oracle JDK

1. Download Oracle NoSQL Database Enterprise Edition, Basic Edition, or Community Edition, depending on which software license you own.
  - **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).
  - **Basic Edition:** If you have an Oracle Database Enterprise Edition license, you can use the Basic Edition at no additional cost, as long as you are in compliance with the terms of the Oracle Database Enterprise Edition license agreement. You can download the Basic Edition from the Oracle Software Download Cloud (edelivery.oracle.com).
  - **Community Edition:** You can download this version from the Oracle Technology Network (OTN) downloads page at: <http://www.oracle.com/technetwork/database/database->



[technologies/nosqldb/downloads/index.html](https://technologies.nosqldb/downloads/index.html). You can also download the Oracle NoSQL Database Community Edition to your local machine using the following command.

```
$ wget http://download.oracle.com/otn-pub/otn_software/nosql-database/kv-ce-4.3.11.tar.gz
```

2. Download the latest 64-bit JDK for RPM-based Linux Platforms. Oracle NoSQL Database requires Java version 1.8 or greater. For example, go to [Java SE Development Kit 8 Downloads](#) and click the link to download `jdk-8u121-linux-x64.rpm` to download the file to your local machine.

---

*Make sure that both the NoSQL .tar.gz file and the JDK .rpm files are in the same directory as your NoSQL cluster install scripts so that the installation scripts can copy them to the BMCS compute nodes.*

---

3. Download the installation scripts from the [Oracle NoSQL Database repository](#) on GitHub. The NoSQL installation directory on your local machine includes the following files:
  - `ons_cluster_install.sh`
  - `ons_node_install.sh`
  - `ons_server_install.sh`

Your local installation directory also includes some variation of the following:

- `jdk-8u121-linux-x64.rpm`
- `kv-ce-4.3.11.tar.gz`

The installation script automatically picks up the jdk and kv files based on their prefix/suffix patterns. For detailed information about what the installation scripts do, see [Appendix B: Script Details](#).

## Step 4: Install and Configure the Oracle NoSQL Database

The installation scripts allow you to install and configure the Oracle NoSQL Database from your local machine. You can run the scripts from a Bash or OS X shell. These scripts do the following:

- Install the NVMe file system on BMCS nodes
- Configure the BMCS network and firewall ports for NoSQL storage nodes
- Install operating system utilities and services needed for the Oracle NoSQL database, including Oracle JDK and NTP service
- Install the Oracle NoSQL Database
- Configure and start up the Oracle NoSQL Database

To install the Oracle NoSQL Database cluster, perform the following tasks:

1. Make sure you're at the machine with the public key that you created in [Step 2: Create the Required BMCS Compute Instances](#).
2. Run the following command:

```
./ons_cluster_install.sh --zone <zoneid> --store <dbname> <ipaddr>
```

where:

- **zone**: name of the zone where the cluster will be installed (for example, PHX)
- **store**: name of the database to be installed on the cluster (for example, ONSDB)

- **ipaddrs:** the public IP addresses of the instances you configured in [Step 2: Create the Required BMCS Compute Instances](#)
- **capacity:** number of replication nodes a storage node supports
- **partitions:** number of partitions per shard
- **username:** admin user for the cluster when security is enabled
- **passphrase:** admin user's password

If any of the `ons_cluster_install.sh` parameters are missing, the install program displays a prompt .  
For example:

```
./ons_cluster_install.sh --zone PHX --store kvstore 129.146.1.1
129.146.2.2 129.146.3.3
Enter a passphrase to create a secure database.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Enter username (admin):
Enter capacity (9): 3
Enter partitions (90): 30
```

The cluster installation script copies the local Oracle JDK .rpm file and Oracle NoSQL Database .tar.gz file to each node in the cluster. Then it copies the `ons_node_install.sh` script to each node, runs the node install scripts in parallel, and waits for them to complete. After the installations are complete, the admin node (first node in the list) is given a deployment plan that it uses to deploy the database to all storage nodes in the cluster.

## Step 5: Verify the Oracle NoSQL Database Installation

Once the installation is complete, perform the following tasks:

1. Use the **opc** account to SSH into one of the storage nodes in the Oracle NoSQL Database cluster.

```
ssh opc@<database IP>
```

2. Verify that the install was successful by running the following command :

```
$ java -jar $KVHOME/lib/kvstore.jar ping -host `hostname` -port 5000
```

If successful, you will see a return like the following example:

```
Pinging components of store ONSDB based upon topology sequence #336
270 partitions and 3 storage nodes
Time: 2017-01-19 17:55:09 UTC   Version: 12.1.4.3.11
Shard Status: healthy:9 writable-degraded:0 read-only:0 offline:0
Admin Status: healthy
Zone [name=PHX id=zn1 type=PRIMARY allowArbiters=false]   RN Status:
online:24 offline:0 maxDelayMillis:1 maxCatchupTimeSecs:0
Storage Node [sn1] on Oracle_NoSQL_DB_AD1_0:5000   Zone: [name=PHX id=zn1
type=PRIMARY
```

```
Storage Node [sn2] on Oracle_NoSQL_DB_AD2_0:5000    Zone: [name=PHX id=zn1
type=PRIMARY
Storage Node [sn3] on Oracle_NoSQL_DB_AD3_0:5000    Zone: [name=PHX id=zn1
type=PRIMARY
```

## Step 6: Install Oracle NoSQL Database Components on the Application Server

An application server is a middle-tier server that sits between a client application and the Oracle NoSQL Database. This server needs Java components from the Oracle NoSQL Database to communicate with the Oracle NoSQL Database storage nodes. To install these components, run the following command on your local machine:

```
$ ./ons_server_install.sh --server <ipaddr> --dbnodes <ipaddrs>
```

where:

- `server` is the server address where NoSQL server software will be installed. In the example configuration, this value is the IP address of the **Oracle\_NoSQL\_AS\_AD1\_0** server you created in the section, [Step 2: Create the Required BMCS Compute Instances](#).
- `dbnodes` are the database public IP addresses from the Oracle NoSQL Database cluster (that is, the IP addresses for **Oracle\_NoSQL\_DB\_\*** you created [Step 2: Create the Required BMCS Compute Instances](#)).

## Step 7: Connect to the Oracle NoSQL Database from the Application Server

After the application server is installed, connect to the Oracle NoSQL Database from the application server:

1. SSH into the application using the **opc** account.
2. Run the following example test to verify that the application server can communicate with the Oracle NoSQL Database.

```
$ ssh opc@<appserver IP>
$ printf "1\n2\n3">/tmp/t.dat
$ java -Xmx256m -Xms256m -jar $KVHOME/lib/sql.jar -helper-hosts
Oracle_NoSQL_DB_AD1_0:5000 -store ONSDB
sql-> create table t(i INTEGER, PRIMARY KEY (i));
Statement completed successfully
sql-> import -table t -file /tmp/t.dat CSV
Loaded 3 rows to t.
sql-> select i from t order by i;
+---+
| i |
+---+
| 1 |
| 2 |
| 3 |
+---+
```

## Optional: Adding Multiple NoSQL Data Stores to BMCS Compute Instances

By default, the Oracle NoSQL Database installation scripts create a single data store on a set of BMCS compute instances. In some cases, this might not be the best use of these compute resources. For example, during an initial development phase you might want separate data stores for development, test, and production that run on the same BMCS hardware. You can implement these custom configurations by running the `ons_cluster_compile.sh` script. The following example installs three databases (dev, test, and prod) on a single cluster.

```
./ons_cluster_compile.sh --zone PHX --store dev, test,prod -capacity 3 -  
partitions 30 -P "" 129.146.1.1 129.146.2.2 129.146.3.3
```

## Appendix A: Running Oracle NoSQL Database on BMCS VMs

It's possible to install Oracle NoSQL Database on BMCS virtual machines (VMs). This appendix details how to create a three-node Oracle NoSQL Database with each node having three block volumes attached.

1. Create the required block volumes.
  - a. Open the Console and navigate to **Storage>Block Volumes**.
  - b. Create nine block volumes, three in each availability domain. Each block volume should have 256 GB of storage.
    - Oracle\_NoSQL\_BV\_AD1\_0, Oracle\_NoSQL\_BV\_AD1\_1, Oracle\_NoSQL\_BV\_AD1\_2
    - Oracle\_NoSQL\_BV\_AD2\_0, Oracle\_NoSQL\_BV\_AD2\_1, Oracle\_NoSQL\_BV\_AD2\_2
    - Oracle\_NoSQL\_BV\_AD3\_0, Oracle\_NoSQL\_BV\_AD3\_1, Oracle\_NoSQL\_BV\_AD3\_2
2. Create the compute instances.
  - a. In the Console, navigate to **Compute>Instances**.
  - b. Create three virtual machines. Follow the instructions in [Step 2: Create the Required BMCS Compute Instances](#) except use **VM.Standard1.4** shapes instead of **BM.DenseIO** shapes.
3. Attach the block volumes to the compute instances. In the Console, click each virtual compute instance and add all three block volumes for the associated availability domain.
4. Run the `ons_cluster_install.sh` script, as in the following example:

```
./ons_cluster_install.sh --zone PHX --store ONSDB 129.146.1.1 129.146.2.2  
129.146.3.3
```

## Appendix B: Script Details

The following sections provide some code snippets from the `ons_node_install.sh` script. These snippets are meant to give insight into the BMCS system configuration.

### File System Setup

The BMCS is delivered as a raw, bare-metal machine with Oracle Linux installed. It is necessary to do some initial hardware configuration, including formatting and mounting drives. The node install script runs the following file-system installation procedure on each node in the cluster.

```
#!/bin/bash  
  
>/tmp/fstab
```

```

for nvme in `sudo fdisk -l | grep nvme | sort | cut -f 2 -d ' ' | sed
's://g`; do
    echo -e "n\np\n1\n\n\nw" | sudo fdisk -u -c $nvme
    sudo mkfs.ext4 $nvme
    mount_dir=`echo $nvme | sed 's/dev/ons/g'`
    sudo mkdir -p $mount_dir
    sudo mount $nvme $mount_dir
    sudo chown opc:opc $mount_dir
    printf "$nvme\t$mount_dir\ttext4\tdefaults\t0\t0\n" >> /tmp/fstab
done

sudo su -c "cat /tmp/fstab >> /etc/fstab"

exit 0

```

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

```

$ df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	38G	2.4G	34G	7%	/
tmpfs	252G	0	252G	0%	/dev/shm
/dev/sda1	543M	280K	543M	1%	/boot/efi
/dev/nvme0n1	2.9T	72M	2.8T	1%	/ons/nvme0n1
/dev/nvme1n1	2.9T	72M	2.8T	1%	/ons/nvme1n1
/dev/nvme2n1	2.9T	72M	2.8T	1%	/ons/nvme2n1
/dev/nvme3n1	2.9T	72M	2.8T	1%	/ons/nvme3n1
/dev/nvme4n1	2.9T	72M	2.8T	1%	/ons/nvme4n1
/dev/nvme5n1	2.9T	72M	2.8T	1%	/ons/nvme5n1
/dev/nvme6n1	2.9T	72M	2.8T	1%	/ons/nvme6n1
/dev/nvme7n1	2.9T	72M	2.8T	1%	/ons/nvme7n1
/dev/nvme8n1	2.9T	72M	2.8T	1%	/ons/nvme8n1

## Network Configuration

The Oracle NoSQL database needs to open ports for client/server and server/server communication. These port numbers are typically in the range 5000+ and will fit in the range 5000–5050 for a 9-shard cluster. Each 9-NVMe node requires 20 ports: one admin port, one client port, and two ports per shard for inter-server communication. The node install script opens up ports as follows.

» For Oracle-Linux release versions less than 7.0 it uses the following:

```

$ sudo /sbin/iptables -D FORWARD -j REJECT --reject-with icmp-host-prohibited

```

```
$ sudo /sbin/iptables -D INPUT -j REJECT --reject-with icmp-host-prohibited
```

» For Oracle-Linux release versions 7.0 and greater it uses the following:

```
$ sudo firewall-cmd --permanent --zone=public --add-rich-rule='rule
family="ipv4" source address="10.0.0.0/27" port protocol="tcp" port="5000-
5050" accept'
```

## NTP Installation

NTP is installed to keep the clocks sync'd on the NoSQL database nodes. This is required for NoSQL replication.

Commands to install NTP and add ntpd to the boot configuration:

```
$ sudo yum -y install ntp
$ sudo service ntpd start
$ sudo chkconfig ntpd on
```

## Further Reading

For more information about the Oracle NoSQL Database, see the following:

- » [Oracle NoSQL Database Overview](#)
- » [Oracle NoSQL Database Documentation](#)







**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

CONNECT WITH US

-  [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)
-  [facebook.com/oracle](http://facebook.com/oracle)
-  [twitter.com/oracle](http://twitter.com/oracle)
-  [oracle.com](http://oracle.com)

**Integrated Cloud Applications & Platform Services**

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

Deploying Oracle NoSQL on the Oracle Bare Metal Cloud Service  
January 2017  
Author: Rick George ([rick.george@oracle.com](mailto:rick.george@oracle.com))  
Contributing Authors: Craig Carl ([craig.carl@oracle.com](mailto:craig.carl@oracle.com))