Oracle NoSQL Database Cloud Service

Fast, Flexible, Fully Managed

WHITE PAPER / FEBRUARY, 2020

[PAGE TITLE] TABLE OF CONTENTS

INTRODUCTION

Oracle NoSQL Database Cloud Service is a fully managed NoSQL database cloud service for today’s most demanding applications that require low latency responses, flexible data models, and elastic scaling for dynamic workloads.

Developers focus on application development without dealing with the hassle of managing back-end servers, storage expansion, cluster deployments, software installation/patches/upgrades, backup, operating systems, and high availability.

Oracle NoSQL Database Cloud Service scales to meet dynamic application workloads and throughput requirements. Users create tables with throughput and storage requirements and Oracle NoSQL Database Cloud Service adapts to meet the table’s provisioned scale-out needs. Provisioned throughout is billed on an hourly basis and is based on provisioned amounts.

WHY FULLY MANAGED?

Oracle NoSQL Database Cloud Service is designed so that developers and organizations that deliver applications using this service can be assured that the underlying software is available, securable and repairable. Oracle NoSQL Database Cloud Service allows developers to create applications that scale as needed with zero administration.

Available

Oracle NoSQL Database Cloud Service is a service that is always available once provisioned. Once a table has been created, customers can connect to it and immediately begin to store and retrieve data. The customer determines the required service levels based on anticipated workloads, and Oracle takes care of the back-end infrastructure, ensuring reliability and high performance.

Securable

When security patches are available, Oracle NoSQL Database Cloud Service is updated quickly with the required patches. Oracle installs applicable security updates behind the scenes, so that the customer can concentrate on using their application. Oracle NoSQL Database Cloud Service remains active during any update, including security updates, which means that customers do not see any interruptions in service.

Repairable

Oracle NoSQL Database Cloud Service includes algorithms for detecting, routing around, and repairing failed nodes automatically. Through a combination of redundant replicas, topology aware routing, and fast ownership transfer algorithms, Oracle NoSQL Database Cloud Service can detect failed hardware and software components, immediately routing around the failure, while starting a background asynchronous process that replaces a failed component with a new component. This process happens transparently with no effect to the ongoing latency and throughput profile of running applications.

DESIGN

Cloud or On-Prem

Oracle NoSQL Database Cloud Service, together with Oracle NoSQL Database is the only fully managed, provisioned throughput NoSQL database system from a cloud provider that allows developers to create applications that run either in their data center or the Oracle Cloud Infrastructure. Developers are free to choose where their applications can store the data, either within a corporation's data center or in an Oracle Cloud Infrastructure region. With Oracle NoSQL Database Cloud Service, organizations can determine the best and most optimized location for their data. IT departments have total control over the location of their most critical asset… their data.

For customers already using Oracle NoSQL Database, it is straightforward to migrate over to Oracle NoSQL Database Cloud Service.   The cloud service uses the most current enterprise edition of Oracle NoSQL Database as its storage engine.

Zero Administration

Oracle NoSQL Database Cloud Service is a cloud service where the administration, maintenance and upgrades of the hardware and software are entirely the responsibility of Oracle. Developers, IT departments and users do not have to spend effort or time in monitoring the clusters health, updating software or be involved with any of the life cycle management of the cluster or software systems. Development time is accelerated and Oracle assures customers that the health of the cluster is constantly monitored.

On-Demand Provisioning of throughput and storage capacity

Oracle NoSQL Database Cloud Service is designed to respond to the changes in the workloads that organizations experience, whether seasonally or on a weekly or even a daily basis. Rather than guess at how many CPUs may be needed, Oracle NoSQL Database Cloud Service allows for the provisioning of the throughput desired as well as the table storage size. Throughput and storage provisioning is defined on a per table basis. Developers and service administrators determine the initial read and write throughput amounts needed to satisfy their requirements. These amounts are expressed as units, with a read unit defined as the throughput of up to one kilobyte (KB) of data per second for an eventually consistent read operation and a write unit defined as the ability to write 1 KB of data per second into the table

For example, if the developers of the application determine that 10,000 KB/sec of write performance is required and 50,000 KB/sec of read performance is required, they would provision 10 write units and 50 read units at table creation time. The "Easy, Elastic" section explains how these values can be dynamically modified. The developer or service administrator specifies the initial size of the table, expressed in gigabytes (GB). A cost estimator is located: here that can give the user an estimate of the cost of the provisioned read units, write units and storage amount for a given period.   Click on Data Management, and scroll until you see Oracle NoSQL Database Cloud.

A key aspect of the cloud service is the ability to increase or decrease the provisioned throughput based on the workloads. For example in figure 1, if the user has provisioned the throughput for either reading or writing at the level A, as the workload increases your application can increase the throughput to level B. You will be able to do this via your application. Then as the workload increases even further, you can increase your provisioned throughput to level C. This changing workload could be seen in a shopping site during peak buying seasons or for any application that has periodic workload changes. As your workload decreases, there is no need to use your Oracle cloud credits to maintain the high throughput at level C. Your application can recognize the decreased demands and scale back to level D, thus saving your company operational expenses. On the right-hand side of the figure, we show latency on the y-axis. While the provisioned throughput may be increasing or decreasing, the response time (latency) stays relatively the constant. Many optimizations have gone into the service to ensure a predictable, low latency response time. With internal measurements, we achieve single-digit millisecond latencies on reads.

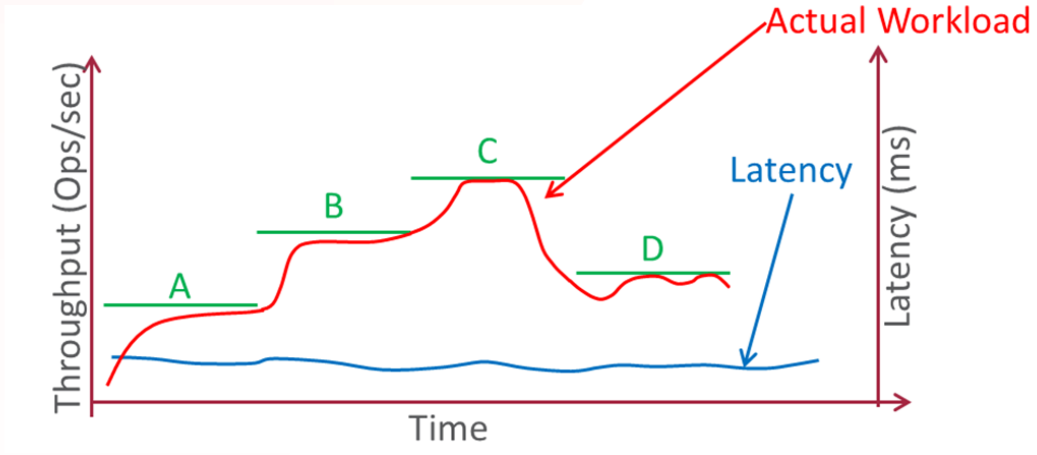

Figure 1 - Increasing and Decreasing provisioned throughput over time

Transactional updates of indices and data

Oracle NoSQL Database Cloud Service provides transactional guarantees for data and indices. When update or delete operations modify data in your table, you can be assured that those write operations will be immediately and consistently accessible via any secondary indexes on that table. Conversely, if any update or delete operations fail for that table, you can also be assured that your indexes have not been affected by these failed operations. Transactional maintenance of your secondary indexes means that your applications can be assured of the correctness of results for read operations on those indexes.

Flexible Data Models

Oracle NoSQL Database gives developers the flexibility to model their data that fits their applications in the most natural form.

A general direction that we see in NoSQL is the adoption of the document model. While each document-oriented database implementation differs on the details of this definition, in general, they all assume documents encapsulate and encode data (or information) in some standard format or encoding such as XML, YAML, and JSON.

JSON has become the accepted medium for interchange between numerous Internet services and unstructured data. The JSON-style document model enables customers to build services that are schema-less. Oracle NoSQL Database Cloud Service support for a native JSON data type defined by RFC 7159 enables efficient access to data in JSON documents.

Oracle NoSQL Database Cloud Service offers flexible options for working with JSON and the choice of which option to use depends on the structure of JSON in your application:

Option 1: JSON documents with different structures can be managed using native JSON data type

- Pro: Documents don’t need to conform to the same structure.
- Con: Each document is self-describing, which means there could be some storage overhead.

Option 2: JSON documents with identical structure can be automatically mapped to a table

- Pro: Oracle NoSQL Database Cloud Service can interpret JSON, create indices, and evolve schema. Schema information is efficiently stored once.
- Con: All documents must have same structure.

Additionally, Oracle NoSQL Database Cloud Service supports columnar and key-value data models. Next there are several application code examples of doing basic operations against the service.

- Below is an example of how to create a table in the application.

```
===to create a table in Oracle NoSQL Database Cloud Service ================


           TableRequest tableRequest = new TableRequest()

                .setStatement("create table if not exists users(id integer, " +

                              "name string, primary key(id))")

                .setTableLimits(new TableLimits(50, 50, 50));

           TableResult tres = handle.tableRequest(tableRequest);

           System.out.println("After Create Table request: " + tres);

           /* Wait for the table to become active */

           tres = TableResult.waitForCompletion(handle,

                                                30000, /* wait 30 sec */

                                                1000); /* delay ms for poll */
```

```
==== end: to create a table in Oracle NoSQL Database Cloud Service =========
```

- Below is an example of how to store a row in the table, which was created above.

```
======================to insert row(s) in a table==================


            /* construct a simple row */

       MapValue value = new MapValue().put("id", 1).put("name","myname");

       PutRequest putRequest = new PutRequest()

           .setValue(value)

           .setTableName("users");


       PutResult putRes = handle.put(putRequest);

======================= end: to insert row(s) in a table========
```

- Below is an example of how to store JSON data in the table. In this example, we are assuming a table users2 exists. This table has two columns, an id column and a name column. We take the JSON record and insert it into the name column.

```
=======================to insert JSON data ========================


    /* PUT row using JSON */

    /* construct JSON row */

private final static String JSON_DATA = "{" +
    "  \"userdata\": ["                    +
    "    {"                                +
    "      \"id\": \"1\","                 +
    "      \"name\": \"Julie Sherman\","   +
    "      \"gender\" : \"female\","       +
    "      \"Age\" : \"25\""               +
    "    }"                                +
    "  ]"                                  +
    "}";



  putRequest = new PutRequest()
      .setValueFromJson(JSON_DATA, null) // no options
      .setTableName("users2");

   putRes = handle.put(putRequest);


 ========================= end: to insert JSON data =====================
```

HIGHLIGHTED FEATURES

ACID Compliant Transactions

ACID (Atomicity, Consistency, Isolation, Durability) transactions are fully supported for the data stored in Oracle NoSQL Database Cloud Service. If needed, consistency can be relaxed in favor of latency.

JSON data support

Applications can query JSON data using familiar SQL queries. This powerful feature gives developers the ability to use SQL to query schema-less JSON data. Oracle NoSQL Database Cloud Service offers the flexibility of rich queries on schema-less data along-side more structured queries. Oracle NoSQL Database Cloud Service is a true multi-model database.

Partial JSON Updates

Oracle NoSQL Database Cloud Service provides an ability to atomically update (change, add, remove) parts of a JSON document. This update happens on the server side eliminating the need for read-modify-write cycle, which could consume throughput capacity. By using this feature, the application saves on throughput resources and improves performance.

Time-To-Live

Allows for data to be stored for a specified period and then deleted automatically. Data can be aged out of the database which is a critical requirement for applications like sensor data capture in an Internet of Things (IoT) service.

SQL Queries

Oracle NoSQL Database Cloud Service supports rich SQL query over fixed and schema-less models, supporting the following query features:

- Projections of fixed schema columns or fragments of JSON documents via JSON path expressions.
- JSON constructors to easily convert fixed schema data into JSON documents.
- Support for rich expressions in predicates, including full support for AND and OR expressions.
- Support for filter expressions for slicing through arrays.

Secondary Indices

Oracle NoSQL Database Cloud Service has rich support for secondary indexing including:

- Deep JSON indexing– Create a secondary index deep into a nested JSON document using a JSON path expression.
- Singleton or composite – Create a secondary index on a single attribute or multiple attributes.
- Scalar or non-scalar data types – Create a secondary index on an array, timestamp, or any other scalar or non-scalar data type.

Data Security

Data is encrypted on disk (at rest) using Advanced Encryption Standard (AES 256). It is also encrypted on the wire (transferring between the application and Oracle NoSQL Database Cloud Service) using HTTPS.

EASY, ELASTIC

ACQUIRING ORACLE NOSQL DATABASE CLOUD SERVICE

It is extremely easy to acquire Oracle NoSQL Database Cloud Service and to start becoming productive with your applications. Step by step instructions can be found in our docs.

CONNECTING TO ORACLE NOSQL DATABASE CLOUD SERVICE

Simply write the application and connect to Oracle NoSQL Database Cloud Service. Below is how your application would connect to Oracle NoSQL Database Cloud Service.

```
===== to create a connection to Oracle NoSQL Database Cloud Service ====

/* Use the SignatureProvider to supply your credentials to NoSQL
 * Database. By default, the SignatureProvider will read your OCI
 * configuration file from the default location, ~/.oci/config. See
 * SignatureProvider for additional options for reading
 * configurations in other ways.
 */

SignatureProvider sp = new SignatureProvider(
 tenantId,      // a string, OCID
 userId,        // a string, OCID
 fingerprint ,  // a string
 privateKey,    // a string, content of private key
 passPhrase     // optional, char[]
);

/* Create a handle to access the cloud service in the eu-zurich-1
 * region.
 */

NoSQLHandleConfig config = new
NoSQLHandleConfig(Region.EU_ZURICH_1);
config.setAuthorizationProvider(sp);
NoSQLHandle handle = NoSQLHandleFactory.createNoSQLHandle(config);

/* At this point, your handle is set up to perform data operations.
 */

===end to create a connection to Oracle NoSQL Database Cloud Service ===
```

ELASTIC

One of the main features of Oracle NoSQL Database Cloud Service is the ability to scale as workloads increase or decrease. A simple DDL interface is available to modify the number of read units, write units or GB of storage in an online fashion while your application is running. The following code fragment shows how to create and provision a table for the first time.

=== to create an Oracle NoSQL Database Cloud Service table =================

```
TableRequest tableRequest = new TableRequest()
```

```
        .setStatement(‘‘create table if not exists foo (a integer)’’)

        .setTableLimits(new TableLimits(200, 100, 500))

        .setTimeout(1000);

TableResult res = conn.tableRequest(tableRequest);
```

======end to create an Oracle NoSQL Database Cloud Service table =================

In this example, the number of Read Units here is set to 200 and the number of Write Units to 100. The storage for this table is initially set to 500 GB.

Eventually, the workloads may increase due to increased customer demand or business needs. The following code fragment changes the Read Units to 400 and the Write Units to 200 and leaving the storage capacity at 500 GB. Of course, the storage could have been altered as well in the same statement.

==================== Adjust table, increase limits =================

```
TableRequest tableRequest = new TableRequest()

        .setTableName(‘‘foo’’)

        .setTableLimits(new TableLimits(400, 200, 500))

        .setTimeout(1000);

TableResult res = conn.tableRequest(tableRequest);
```

===============end Adjust table, increase limits =================

Perhaps later, the performance requirement for the application dropped and the Read Units and/or the Write Units can be reduced resulting in a lower expenditure of credits. The code fragment would look like the following. Note that the same API call is made whether increasing or decreasing the provisioned amounts.

==================Adjust table, decrease limits =================

```
TableRequest tableRequest = new TableRequest()

        .setTableName(‘‘foo’’)

        .setTableLimits(new TableLimits(100, 100, 500))
        .setTimeout(1000);
TableResult res = conn.tableRequest(tableRequest);
```

===============end Adjust table, decrease limits =================

DEPLOYMENT CHOICES

Oracle offers a range of deployment options of its NoSQL APIs, libraries and storage technologies. Whether deploying on an “On-Premises” cluster or utilizing the Oracle NoSQL Database Cloud Service, developers and end-users get the latest in NoSQL technology.

Oracle offers a complete range of services to support your data warehouse, from on-premises to private cloud to public cloud.	**On-Premises**	**BYOL in Oracle Cloud Infrastructure**	**Oracle NoSQL Database Cloud Service**
Optimized HW	Possibly	Yes	Yes
Licensing Model	Purchased (Perpetual or Term)	Purchased (Perpetual or Term) + OCI IaaS	Metered (throughput and storage)
Database Automation	No	No	Yes
Location	On-Prem	Oracle Cloud Infrastructure	Oracle Cloud Infrastructure
Fully (SW, HW) Managed by Oracle	No	No	Yes

Dealing with Provisioned Limits

Oracle NoSQL Database Cloud Service allows the developer to set limits on the throughput used. When the provisioned throughput is reached for either writing or reading, the requests will be throttled. The write/read requests are transparently retried by the Oracle NoSQL Database Cloud Service driver until successful. No interaction is required from the application. However, the ability to handle exceptions in the application is available, should the developer wish to deal with the exceptions directly. When throttling happens, Oracle NoSQL Database Cloud Service will throw an exception ThrottlingException back to the application. At this stage the application can decide to wait before retrying the request. Operations resulting in this exception can be retried immediately, but it is recommended that callers use a delay before retrying to minimize the chance that a retry will also be throttled. In general, applications should attempt to avoid throttling exceptions by rate limiting the application themselves or by provisioning higher throughput.

Oracle Cloud Components

OCI - The Oracle Cloud Infrastructure combines the elasticity and utility of the public cloud with the granular control, security, and predictability of on-premises infrastructure to deliver high-performance, high availability and cost-effective infrastructure services. The storage component consists of SSDs over NVMe which delivers extremely high levels of performance.

Universal Credits - Oracle NoSQL Database Cloud Service is integrated with Oracle Universal Credits. Oracle Universal Credits are the industry's most flexible buying and consumption model for cloud services. With Universal Credits, customers have one simple contract that provides unlimited access to all current and future Oracle PaaS and IaaS services, spanning Oracle Cloud and Oracle Cloud at Customer. Customers gain on-demand access to all services plus the benefit of the lower cost of pre-paid services. Additionally, they have the flexibility to upgrade, expand or move services across data centers based on their requirements. With Universal Credits, customers gain the ability to switch the PaaS or IaaS services they are using without having to notify Oracle. Customers also benefit from using new services with their existing set of cloud credits when made available.

ORACLE NOSQL DATABASE CLOUD SERVICE USE CASES

Oracle NoSQL Database Cloud Service can solve a broad range of application problems frequently prevalent in modern applications. With support for predictable, low latency access to key/value data, with its rich JSON query experience, Oracle NoSQL Database Cloud Service is an excellent choice for applications requiring single-digit, millisecond access in numerous scenarios. This is especially relevant to data with extensive request velocities and frequently fluctuating seasonal demands. Several of the following important Oracle NoSQL Database Cloud Service characteristics make it an ideal platform for use cases such as user experience personalization, real-time fraud detection, and IoT data management:

- Low latency access to simple queries – NoSQL database access does not involve complex queries or aggregations. Instead, Oracle NoSQL Database Cloud Service focuses on delivering predictable, single-digit millisecond latencies to very simple queries.
- Data auto-sharding – Data is automatically partitioned across a shared-nothing database cluster that is replicated across availability domains.
- On-demand scaling up or down – The entire back end cluster is autonomously managed for high availability and scalability. Developers increase or decrease provisioned throughput to match the demands of application.
- Flexible reserved throughput pricing – Your requested database throughput is reserved and you can change it dynamically at any time.

Personalization at Scale

To enable a rich and responsive personalized user experience, modern applications require extremely fast access to many fragments of database data. In fact, it is common for especially rich customized user interfaces that must read tens to hundreds of data fragments during page rendering. To serve each user with predictable page load times (typically less than two seconds), each database access must complete predictably in a few milliseconds, no matter how many users your application is serving concurrently.

As business grows, acquiring ever more users, database latency requests must remain predictable. A common practice is to use customer browsing behavior, historical purchases, and other demographic data to present product recommendations. These recommendations are then stored in Oracle NoSQL

Database Cloud Service, typically as JSON data, for fast retrieval and rendering of UI elements. Hierarchical product catalog data, recent purchase history, and product recommendations are examples of the kinds of data required to customize great user experience. With its predictable single-digit latencies, rich JSON data support, and automatic scale out, Oracle NoSQL Database Cloud Service is the ideal platform for enabling a personalized user experience.

Real-Time Fraud Detection

Electronic card payments have grown at an average yearly rate of 8.6%, to a total of 37.3 billion transactions per year. Accompanying this increase in volume is a corresponding increase in potential fraudulent transactions.

As fraud detection models get more sophisticated, organizations have moved fraud scoring closer to the point of sale (POS), to prevent fraud proactively, rather than responding reactively. Adding fraud scoring into the POS transaction path requires a data store capable of responding with predictable low latency to a high volume of lookup and update requests. Furthermore, as transaction rates fluctuate with seasonal peaks in demand, the data store must adapt accordingly: transparently scaling to meet peak demands and contracting during low demand periods to save costs. Oracle NoSQL Database Cloud Service provides a perfect platform for implementing your real-time fraud detection solutions. How? By its support for key/value access, predictable low latency at scale, and the capability to scale up or down automatically on demand.

Internet of Things (IoT)

What is the IoT space? In simple terms, the IoT is a system of interconnected devices passing information from one to another without human intervention. Typically, this capability requires edge processing, network connectivity, data storage, and back-end analytics processing.

Equally important, but often not publicized, is how and where back-end data is stored. A viable IoT back end must handle blindingly fast write operations, while simultaneously supporting data analytics demands for read operations. Oracle NoSQL Database Cloud Service, with its support of time series data, deterministic automatic data expiry, and the ability to scale to handle extreme insert velocity, while concurrently supporting analytic data read demands, is an excellent choice for the data storage of IoT style workloads.

In addition to high-velocity data ingestion, the back end must support the data formats of popular IoT transmission protocols. While also supporting fixed schema and key/value data formats, Oracle NoSQL Database Cloud Service offers native support for JSON objects, one of the most common data formats used by IoT data protocols. Using the Oracle NoSQL Database Cloud Service JSON data type, IoT applications can store and query high volumes of data, with minimal extra processing to transform the data payload. With Oracle NoSQL Database Cloud Service rich support for SQL over JSON data, and deep support for JSON secondary indexes, the IoT development task for back-end storage and analytics preparation becomes much simpler and the time for implementation and deployment for your IoT solutions shrink accordingly.

EXAMPLES BASED ON WORKLOAD REQUIREMENTS

Various applications, depending on the workloads and types of data that need to be stored will require differing amounts of read units and write units. Several factors impact how capacity needs to be provisioned. The factors include:

- Record size – as the size increases, the number of request units increase.

- Data Consistency – If absolute consistency is required by the application, more read units will be required.
- Secondary Indexes – updating the secondary indexes will consume write capacity units.
- Data models – the choice of the data model employed affects the number of read units or write units provisioned.

Please see the Capacity Planning white paper.

SUMMARY

Oracle NoSQL Database Cloud Service is an ideal system for developers that require the ability to change the provisioned throughput quickly based on workloads. The API for Oracle NoSQL Database Cloud Service is simple and easy to use. A variety of data types can be used in a table. The predictable latencies appeals to any developer that is creating an application where users require fast interaction. As a fully managed service, developers and organizations can concentrate on creating innovative applications without the hassle of maintaining a server, storage, and network infrastructure.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries

TELE	+ 1.650.506.7000	+ 1.800.ORACLE1
FAX	+ 1.650.506.7200	

oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com/nosql facebook.com/oracle twitter.com/oraclenosql

Integrated Cloud Applications & Platform Services

Copyright © 2020, 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. 0220

Oracle NoSQL Database Cloud Service V13
February 2020
Author: MB

Oracle is committed to developing practices and products that help protect the environment