ORACLE

Oracle Autonomous JSON Database

Oracle Autonomous JSON Database offers developers a rich feature set including Document APIs and SQL, transactional consistency, advanced security and exceptional performance.

March, 2021, Version 21c
Copyright © 2021, Oracle and/or its affiliates
Public

Purpose statement

This document provides an overview of Oracle Autonomous JSON Database (AJD) and outlines how it can be a better document store than specialized NoSQL document stores in the market today. It is intended solely to help you evaluate the Autonomous JSON Database as a platform for your JSON storage and processing requirements.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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 in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Copyright © 2021, Oracle and/or its affiliates / Public

Table of contents

List of images

No table of figures entries found.

Copyright © 2021, Oracle and/or its affiliates / Public

Purpose

This document provides an overview of Oracle Autonomous JSON Database (AJD) and outlines how it can be a better document store than specialized NoSQL document stores in the market today. It is intended solely to help you evaluate the Autonomous JSON Database as a platform for your JSON storage and processing requirements.

- **Autonomous Database**. AJD is part of the Autonomous Database suite of services giving developers a consistent look-and-feel with interoperability across services. AJD is fully autonomous with 99.95% availability, zero-administration and fully elastic.
- **Document Store**. Developers use NoSQL-style document APIs (SODA) to store data in simple collections of schema-flexible JSON documents. Developers also have full SQL and PL/SQL access over the JSON data.
- **Much more than a Document Store**. AJD offers additional features compared to other NoSQL document stores including SQL and PL/SQL, transactions and ACID consistency, APEX low-code development.
- **Performance**. AJD utilizes the native JSON query processing and optimized binary JSON storage format of the Oracle Database platform. It runs on an Oracle Exadata engineered system to support exceptional performance at scale.
- **One-click upgrade to Autonomous Transaction Processing (ATP)**. AJD lets you "future proof" your application architecture with the unique ability to upgrade to ATP to adopt other data models including Relational, Graph, XML as your application requirements evolve.

"With Oracle Autonomous JSON Database service the administrative and operational aspect of JSON-based next gen Apps is taken care of, and the result is both a better developer experience and higher developer velocity."

Holger Mueller
VP and Principal Analyst, Constellation

Copyright © 2021, Oracle and/or its affiliates / Public

INTRODUCTION

Oracle Autonomous JSON Database (AJD) is a cloud document database service for developers building JSON-centric applications. This blog introduces the AJD service and gives an overview of the capabilities. Developers interface with collections of JSON documents using JSON based APIs called **SODA (Simple Oracle Document Access)**. These APIs enable developer agility with schemaless storage in JSON and native SODA language drivers to allow developers to implement in the language of their choice.

But unlike specialized NoSQL document stores, AJD also allows developers to use **SQL** over the same data for analytics or reporting. For developers, this opens up the whole relational ecosystem of tools and applications over their JSON data. AJD additionally includes many distinctive features not available in other NoSQL stores in the market including **transactional consistency**, **APEX** low code development among others.

AJD is part of the Autonomous Database suite of services based on the Oracle Converged database platform. AJD runs on an **Exadata** engineered system with JSON encoded in an Oracle optimized binary representation leveraging the full power of the Oracle Database infrastructure such as Parallel Query, Optimizer, RAC. In benchmarks, AJD significantly outperforms specialized NoSQL document stores without trading off transactional consistency.

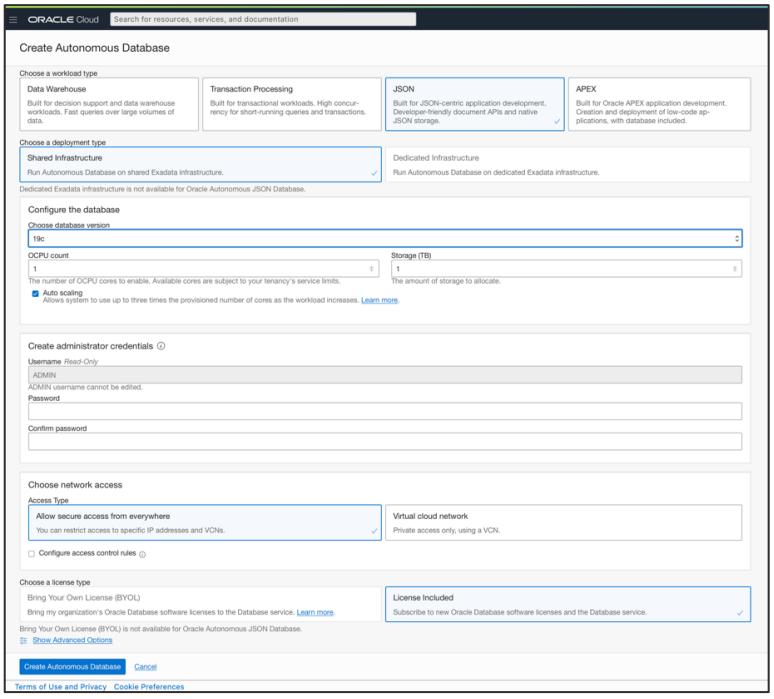
AJD is fully managed, serverless and elastic with pay-per-use automatic scaling. This makes it an ideal persistent store for modern enterprise microservices and applications that must constantly adapt to variability in demand. AJD offers developers a document store with a richer feature set and better performance than specialized NoSQL document stores, while also giving them access to the broad Autonomous Database platform.

OVERVIEW

GETTING STARTED

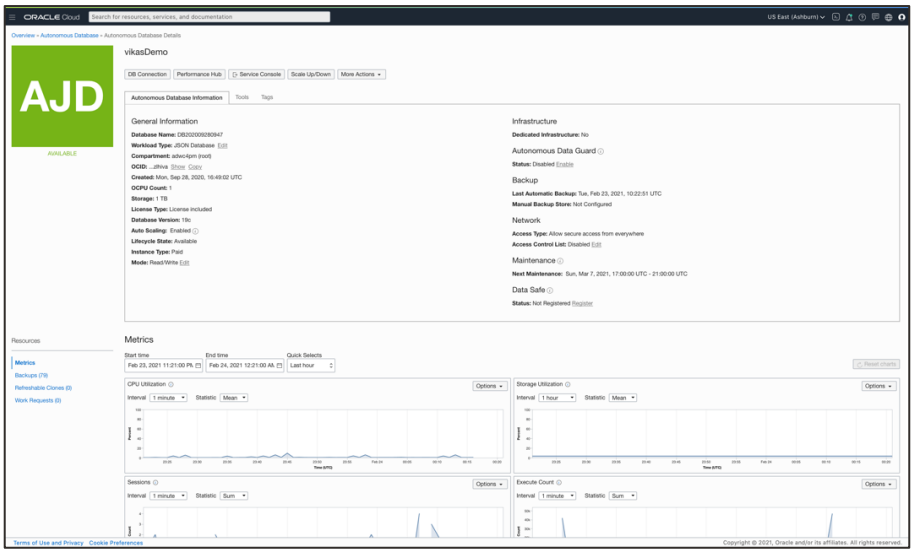
Getting Started with Autonomous JSON Database for instructions on getting started with AJD. Developers can sign up for paid AJD directly or start with a free version of the Autonomous Database at https://www.oracle.com/cloud/free/. The Free service is capped at 1 OCPU with 20 GB of storage and can be easily promoted to a paid AJD instance. AJD is competitively priced and the fully-licensed price is that same as the BYOL (Bring your own license) price for other services, which can be up to 75% lower than the full priced services.

After signing up, creating a new AJD instance is easy and fast.

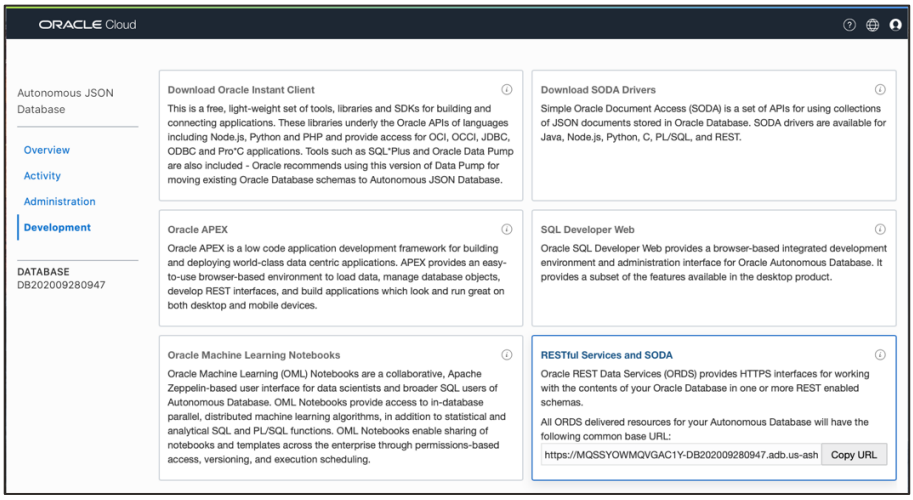


Copyright © 2021, Oracle and/or its affiliates / Public

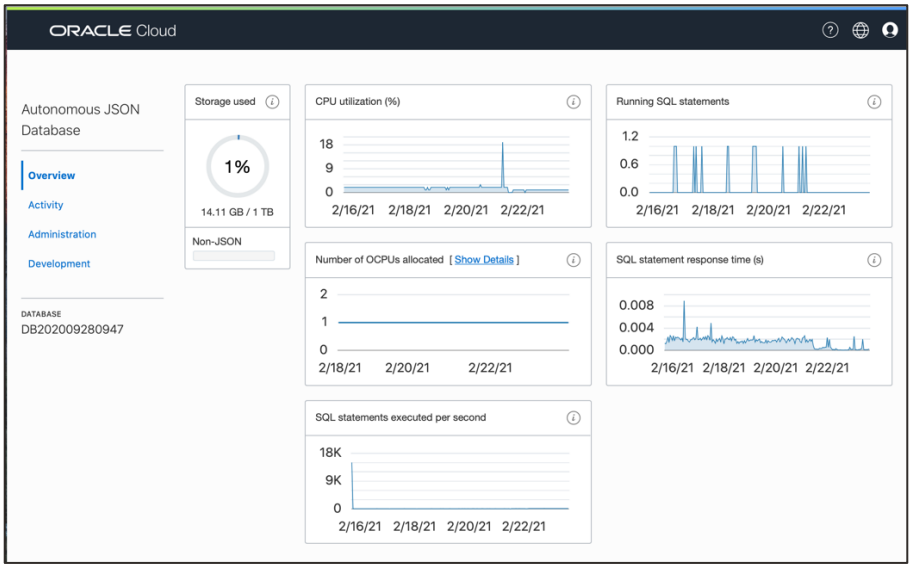
It takes a few seconds to provision a new AJD instance.



With AJD provisioned, developers have access to tools and libraries for interacting with the service.



A Service Health Dashboard can be used to look at the system health and metrics. Users can subscribe to notifications and record of incidents from the Health dashboard at https://ocistatus.oraclecloud.com.

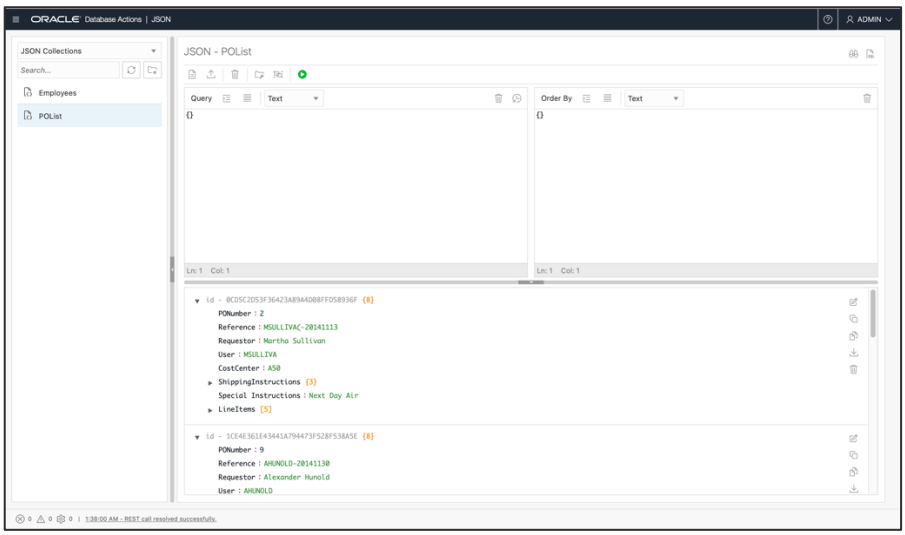


Copyright © 2021, Oracle and/or its affiliates / Public

JSON DOCUMENT STORE

AJD lets developers create and operate on collections of JSON documents using the SODA API (Simple Oracle Document Access). SODA includes native language drivers for Java, Node.js, Python, OCI, PLSQL. Autonomous Database includes a REST server which allows access to all collections using SODA REST. SODA can also be executed from the Oracle SQL Developer Web that is preconfigured and can be accessed from the console, and also from the command line using SQLcl.

The JSON workshop in Oracle SQL Developer Web can be used to browse through and query collections and documents, as well as to create and load new documents.

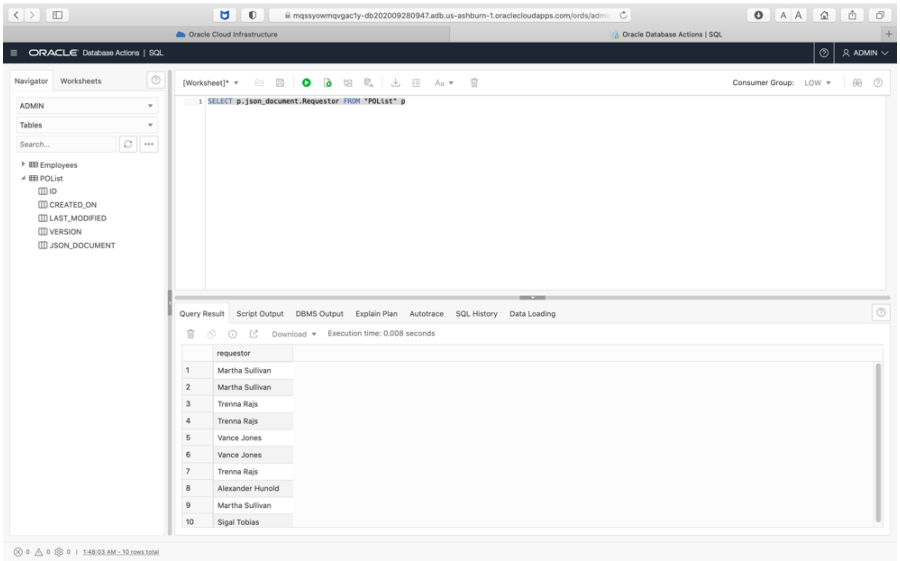


AJD comes with the DBMS_CLOUD package installed to facilitate loading data from Oracle Object Store. The DBMS_CLOUD.COPY_COLLECTION procedure can can bulk-load files containing JSON documents into a collection.

SODA supports a rich set of capabilities over the collections. This includes

- CRUD for collections and documents
- Query by Example based on a JSON query specification
- Indexing including Schema Discovery (data guide)
- Full Text search
- Spatial searches

Collections are backed by Oracle Database Tables which can be accessed using Oracle SQL/JSON. Users can switch between SQL Developer and the JSON workshop to query over the same data at any time.



Copyright © 2021, Oracle and/or its affiliates / Public

AUTONOMOUS DATABASE

AJD is part of Oracle's Autonomous Database services which require no configuration or administration. The database is configured automatically for JSON workloads. Features like Parallel Query are enabled with the degree of parallelism determined automatically, optimizer statistics are gathered automatically. Database protection is automatically configured with standard and supplemental database backups. The Database can detect physical corruptions and perform recovery on its own. Users can also use the Database Console for point-in-time recovery.

Autonomous Database customers are billed based on compute and storage used. Autonomous Database supports auto-scaling where the CPUs and storage are automatically scaled to match the workload. The scaling is transparent to the application and requires no downtime. Users also have the option of scaling the service manually by adjusting the compute (OCPUs) and storage independently. The PGA, SGA memory and I/O bandwidths are derived based on the OCPU and storage assigned.

ORACLE CLOUD INFRASTRUCTURE

Autonomous Database is based on the Oracle Cloud Infrastructure (OCI) which is a next-generation cloud infrastructure for cloud-native applications. It includes a comprehensive menu of services for the application development lifecycle such as containers, identity management. These can be used to build applications backed by the Autonomous Database. A reference example for an end-to-end application on the Oracle Cloud is the mushop application. The code for the application is on github. The carts service runs on JSON collections and the code is available on github as well.

Copyright © 2021, Oracle and/or its affiliates / Public

A BETTER DOCUMENT STORE THAN SPECIALIZED DOCUMENT STORES

Oracle AJD delivers the features that users expect of a document store service including schemaless representation, document APIs, auto-scaling, high availability as described in the previous section. However, AJD also adds many more capabilities to the document store to elevate the developer experience and eliminate administrative overheads

FULL SQL/PLSQL ACCESS

One of the most powerful features of AJD is that it gives developers **full SQL and PL/SQL access** over their collection data. The full breadth of Oracle's SQL/JSON support conforming to the SQL 2016 standard is available to developers. This includes SQL operators for querying and generating JSON. Oracle PL/SQL with native APIs to manipulate JSON can also be used over the JSON data and can be used to define triggers and stored procedures.

Unlike specialized NoSQL document stores, there is no need to move the data out to a SQL based store and deal with maintaining pipelines. AJD is based on the Oracle converged database that gives developers a duality of access mechanisms over the same collection data. Developers are free to choose the most intuitive mechanism depending on the requirement, such as SODA for application development and SQL for analytics. Moreover, with SQL operators like JSON_TABLE to project out relational views over JSON, the rich ecosystem of relational tools and applications can be brought to bear on the JSON data.

TRANSACTIONS AND ACID CONSISTENCY

NoSQL document stores are typically architected to trade off transactional consistency for other system objectives such as availability, performance. Transactional consistency can be layered on but requires two-phase commits across nodes and can be expensive and complex. Sacrificing transactional consistency is a high price to pay for developers because it pushes the complexity of maintaining consistency into the application code and makes it much harder for developers to reason about the correctness of their applications.

AJD on the other hand uses Oracle's database infrastructure to **give developers the availability and performance they need without having to trade off transactional consistency**. AJD uses Oracle's transactions to guarantee Atomicity, Consistency, Isolation and Durability (ACID). Oracle Read Consistency is enabled by default that leverages multi-version concurrency control to allow reads to be interleaved with updates.

SODA developers have access to the underlying JDBC, OCI and python connections that they can use to wrap SODA operations within begin and end transaction commands. As a result, transactions with SODA can span multiple documents and multiple collections.

FLEXIBLE DENORMALIZATION

Relational databases represent data in a fully normalized representation. Specialized NoSQL document stores steer developers towards the other extreme of a denormalized representation stored in JSON objects. However, applications usually have some data which is easily denormalized into a single collection, and some which requires creating separate collections to avoid redundancy. The latter is effectively normalization of the data, yet NoSQL document stores offer little or weak support for referential integrity and leave it to the application to manage.

AJD on the other hand, leverages Oracle's advanced database infrastructure to let developers **find the right balance between normalizing and denormalizing their data**. Developers can choose to model their data as denormalized in JSON collections or as separate collections with references. They have the ability to create constraints across collections to enforce referential integrity. They can use transactions to ensure atomic updates of the collections. They can also perform joins across collections using SQL. AJD thus empowers developers to pick the best data model for their needs and provides support for the level of denormalization they need.

Copyright © 2021, Oracle and/or its affiliates / Public

RICH MULTI-MODEL SUPPORT

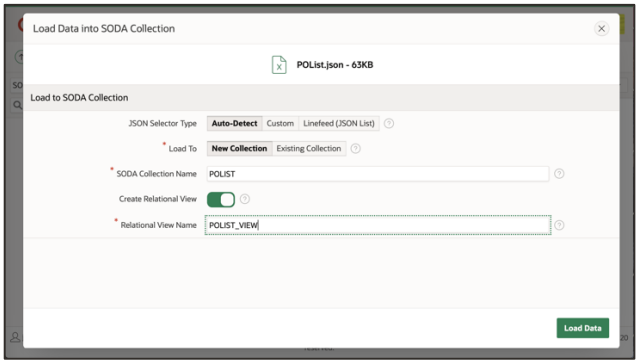
AJD leverages Oracle's Converged Database platform to enable **Full Text search** over the JSON data based on Oracle Text. Oracle Text is architected to be JSON document aware so that searches can be restricted to specific fragments of JSON documents and can also be used for value and range searches so that separate BTree indexes do not need to be maintained in addition. Oracle Text supports a broad set of features including keyword searches, mixed thematic queries, multilingual analysis, classification and clustering. AJD thus includes a complete Text search solution with no need to move data to an external search engine as with other NoSQL document stores.

AJD can also be promoted to ATP (as described later) which is a full multi-model database that includes support for Relational, JSON, Graph, XML, among others. Developers have **immense flexibility in combining data models** and querying them in-place with SQL. This is unlike NoSQL document stores which specialize in a single data model and require data to be moved out to other data stores such as Oracle for relational processing, Neo4J for Graph. Hybrid models combining Relational and JSON are particularly popular with developers and Oracle's Database platform sets the standard with the most comprehensive interoperability between Relational and JSON.

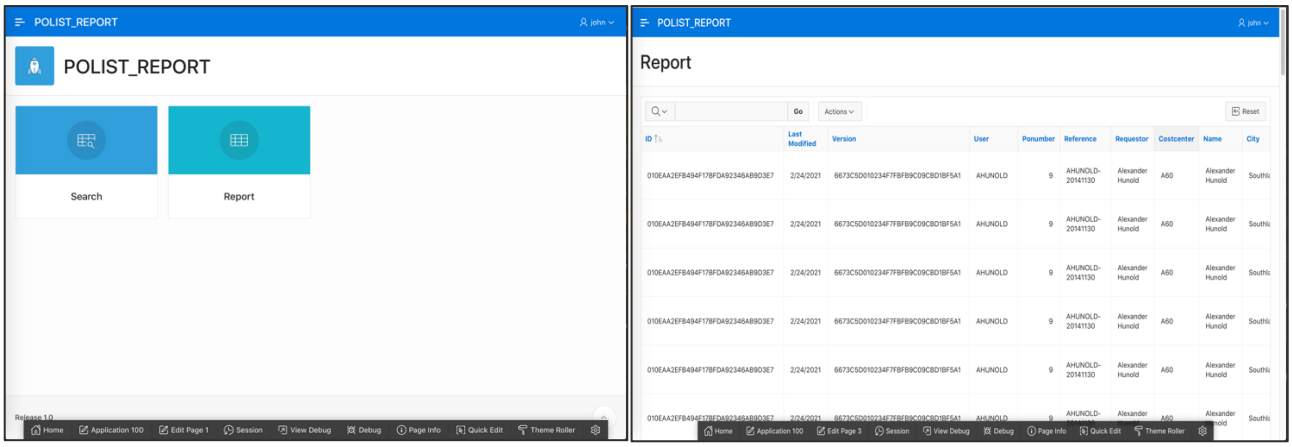
APEX LOW CODE DEVELOPMENT

AJD includes Oracle APEX which is the industry leading low code platform for enterprise apps. Oracle APEX is integrated with collections in AJD to provide **a unique low-code development environment in a document store**. It enables developers to build complex and responsive dashboards over their JSON data in a fraction of the time it would take on other document stores.

The APEX support is based on relational views constructed over the JSON data. Oracle APEX in AJD lets you ingest JSON data into a collection. APEX uses the data guide to discover the structure of the ingested JSON and takes care of constructing relational views automatically.



Oracle APEX uses these views to automatically create interactive reports as well as allowing users to customize these and produce sophisticated and responsive dashboards with minimal effort.



Copyright © 2021, Oracle and/or its affiliates / Public

ADVANCED SECURITY

As part of the Oracle Autonomous Database, AJD is secure by default. All data is stored encrypted with Transparent Data Encryption and secured with Data Vault which provides an added layer of security. AJD also includes Advanced security features of the database such as VPD, data masking that can be used with JSON data in collections.

With the Oracle Data Safe service, administrators have the option of having a unified control for the security of data across Oracle Databases that can include an AJD instance.

SUMMARY

The comparison with MongoDB as of Aug 2020 (from AJD blog) illustrates AJD's advantages as a document store.

	Autonomous JSON Database	MongoDB Atlas
Max Document Size	32 MB	16 MB
Max nested depth of documents	1024 levels	100 levels
Indexes per collection	unlimited	64
Compound index fields	unlimited	32
Full document index	JSON Search Index	x
Server-side functions	Functions, procedures, triggers	Not recommended*
Multi-document transactions	Always ACID	ACID only upon request via explicit API calls
Transaction duration	unlimited	60 seconds default
Transaction size	unlimited	maximum of 1000 documents*
Aggregation data size	unlimited	100 MB RAM + explicit allowDiskUse param
Serverless auto-scaling	✓	x
SQL access over JSON documents	✓	x
Comprehensive security (e.g. Virtual Private Database, Data Redaction, Custom Database Roles)	✓	x
Low Price	$2.74 / hour	$3.95 / hour

* recommendations as per MongoDB documentation: link1, link2

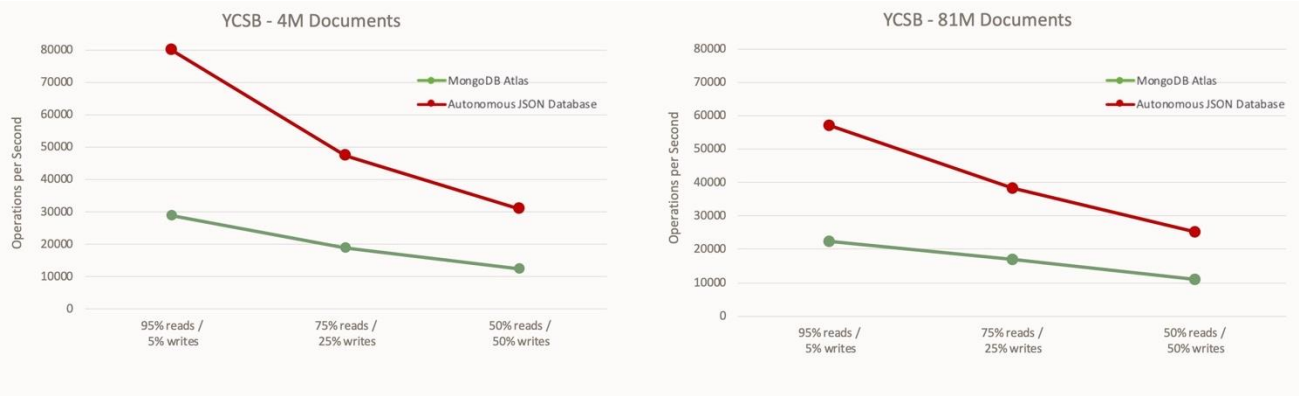
Copyright © 2021, Oracle and/or its affiliates / Public

PERFORMANCE

Oracle Autonomous JSON Database is based on the Oracle Database platform running on an engineered system, with deep integration of JSON and finely tuned for JSON workloads. This enables AJD to deliver industry leading performance and single digit ms response times for customers.

BENCHMARK

Oracle AJD was benchmarked in-house at Oracle in August 2020, using the YCSB benchmark which is a popular benchmark to evaluate NoSQL stores. When compared with benchmark numbers for MongoDB Atlas as published by MongoDB, Oracle AJD delivered **2x** throughput of MongoDB Atlas across different workload types and collection sizes.

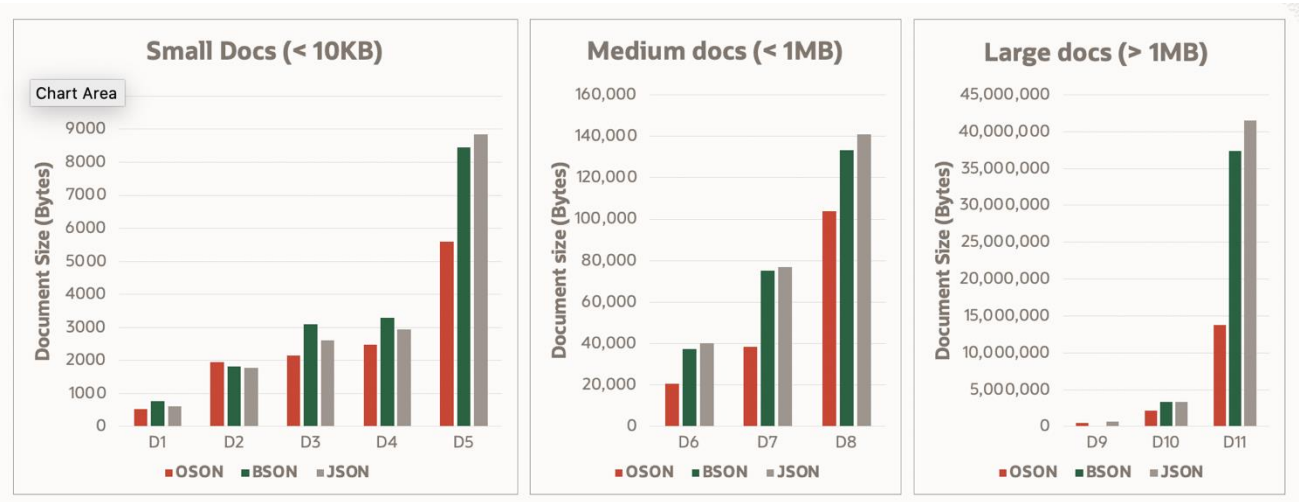


The benchmark was run on Oracle AJD with 8 OCPUs which is equivalent to the M60 cluster used by MongoDB for its published numbers at https://www.mongodb.com/atlas-vs-amazon-documentdb/performance on 8/12/2020.

The Oracle AJD YCSB benchmark kit is available at https://github.com/oracle/json-in-db/tree/master/YCSB.

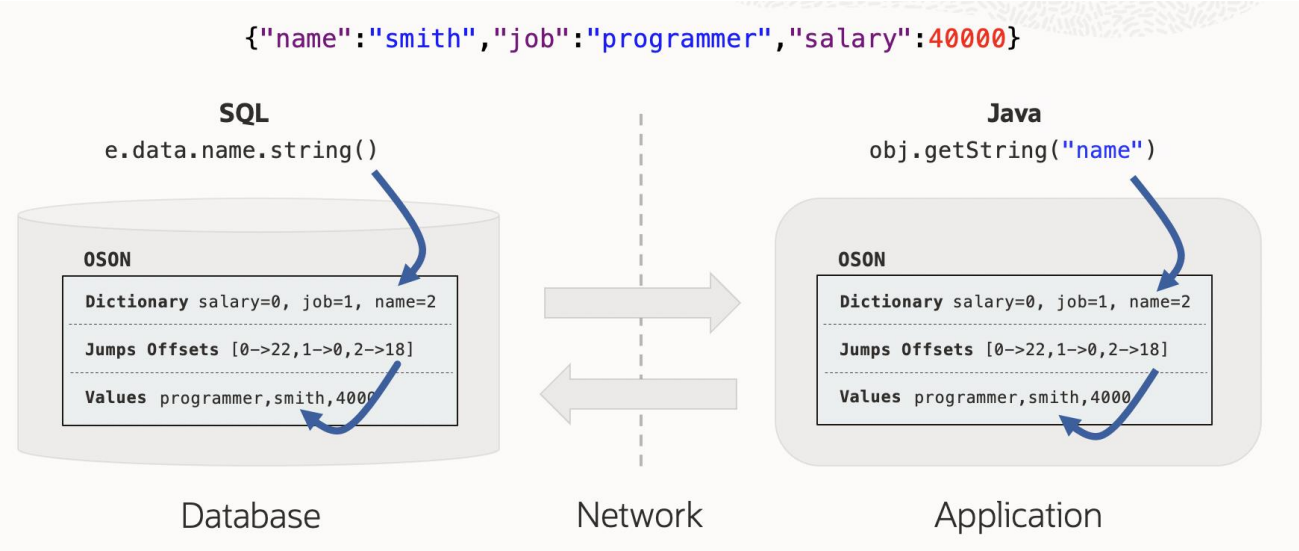
ORACLE BINARY REPRESENTATION FOR JSON

Oracle AJD uses an optimized binary representation called OSON that is designed for fast query and piecewise updates. OSON uses dictionary encoding to create a **highly compressed** representation that makes it efficient to store, retrieve and transmit OSON. As shown below, OSON is up to 2.7x more compressed than BSON and up to 3x more compressed than JSON.



The OSON representation also includes a tree structure which enables **efficient random access** for fields within documents. The dictionary encoding and the tree structure allow paths to be evaluated efficiently without string comparisons. In the figure below, a field can be extracted by jumping to offset associated with the field in the tree structure. This leads to very fast response times in extraction of fields and values from JSON documents.

Copyright © 2021, Oracle and/or its affiliates / Public



Oracle's support for OSON is built into both the database server as well as the client libraries including OCI and JDBC drivers. As a result, OSON is supported **end-to-end across the client and the server** and can be transmitted without any intermediate Text parsing or serializations.

Oracle's binary representation serves as a foundation for fast JSON processing in the database and combined with other optimizations and indexing techniques leads to very fast response times for JSON operations. For a deeper dive on OSON refer to the Oracle Database blog or Oracle's VLDB 2020 paper.

EXADATA

Oracle AJD is a part of the Autonomous Database-Shared which runs on the Oracle Exadata Database machine. The Oracle Exadata Database Machine is engineered to deliver dramatically better performance, cost effectiveness, and availability for Oracle databases. Exadata features a modern cloud-enabled architecture with scale-out high-performance database servers, scale-out intelligent storage servers with state-of-the-art PCI flash, leading-edge storage cache using persistent memory, and cloud scale RDMA over Converged Ethernet (RoCE) internal fabric that connects all servers and storage. Unique algorithms and protocols in Exadata implement database intelligence in storage, compute, and networking to deliver higher performance and capacity at lower costs than other platforms, for all types of modern database workloads including Online Transaction Processing (OLTP), Data Warehousing (DW), In-Memory Analytics, Internet of Things (IoT), as well as efficient consolidation of mixed workloads.

AJD is based on Oracle's Database JSON infrastructure architected to maximally leverage Exadata. These optimizations include Oracle binary JSON processing libraries built into Exadata storage servers so that JSON queries can use smart scans whereby JSON predicates are pushed down into the storage servers. In addition, the size of JSON in-lined into LOBs is increased from the default 4KB to 8KB to maximize feasibility of smart scans. The binary JSON retrieved from smart scans is designed so that it traverses all layers of Exadata without conversions and is highly compressed to make the most efficient use of Exadata's caches. All of these optimizations significantly speed up JSON processing on Exadata. AJD is unique relative to other document stores in the depth of integration of the software with the underlying Exadata hardware.

Copyright © 2021, Oracle and/or its affiliates / Public

ONE-CLICK PROMOTION TO ATP

AUTONOMOUS DATABASE

Autonomous JSON Database is part of Oracle's Autonomous Database suite of services for different workloads. Oracle's Autonomous Database combines the flexibility of the cloud with machine learning to deliver a fully automated data management platform. The Autonomous Database is self-driving, self-securing and self-repairing.

Oracle Autonomous Database workloads include

- Transaction Processing (ATP)
- Data Warehouse (ADW)
- JSON (AJD)
- Low code development (APEX)

Oracle Autonomous Database gives users a consistent look and feel across services, as well as unified tools and APIs. The Autonomous Database architecture based on Oracle's converged database platform avoids the fragmentation in APIs and interfaces seen with services in other cloud that create services based on disparate products.

AJD PROMOTION TO ATP

The Autonomous Database architecture enables all JSON features of AJD to be made available as part of Autonomous Transaction Processing (ATP) and Autonomous Data Warehouse (ADW) services with the same set of APIs and interfaces. ATP in particular expands the feature set available to the application to add full multi-model capabilities including Relational, Spatial, XML among others.

The full multi-model capabilities are also available in AJD to try out, up to the 20GB limit for non-JSON data. Users have the option of promoting their AJD service to full ATP to take full advantage of the multi-model capabilities in ATP as application requirements evolve. The promotion can be done at any time with a single click.

CONCLUSION

Oracle's Autonomous JSON Database offers a unique platform for JSON-centric development. JSON based APIs and native language drivers empower developers to build applications quickly with no schema design and in the language of their choice. The service is fully elastic with pay per use and is an ideal persistent store for microservices or applications which often have to deal with variable loads. The service is built on the Exadata engineered system and offers best-of-class performance with a specialized binary JSON representation optimized end-to-end from the client to the server.

Developers additionally also have full SQL and PL/SQL access over the same JSON data for SQL-based development. AJD includes Oracle APEX which is a low-code development environment to build responsive dashboards and reports over the JSON data within a few minutes. At the same time, AJD eliminates some of the trade-offs traditionally associated with using NoSQL document stores. AJD offers transactional consistency, multi-model support, and referential integrity that gives developers considerable latitude in the design of their application data models.

Oracle's Autonomous JSON database is part of the Autonomous Database suite of services for different workloads including Transaction Processing (ATP), Data Warehouse (ADW), Low code development (APEX). All of the services leverage the powerful Oracle converged database platform to give users a consistent look-and-feel across services, uniform tools and interfaces, as well as interoperability between the services. Unlike NoSQL stores which are specialized for a single data model, the Autonomous JSON Database offers developers unparalleled flexibility with the ability to start out with any storage model and promote to ATP to adopt other data models if application requirements change.

Copyright © 2021, Oracle and/or its affiliates / Public

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 facebook.com/oracle twitter.com/oracle

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.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

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. 0120

Copyright © 2021, Oracle and/or its affiliates / Public