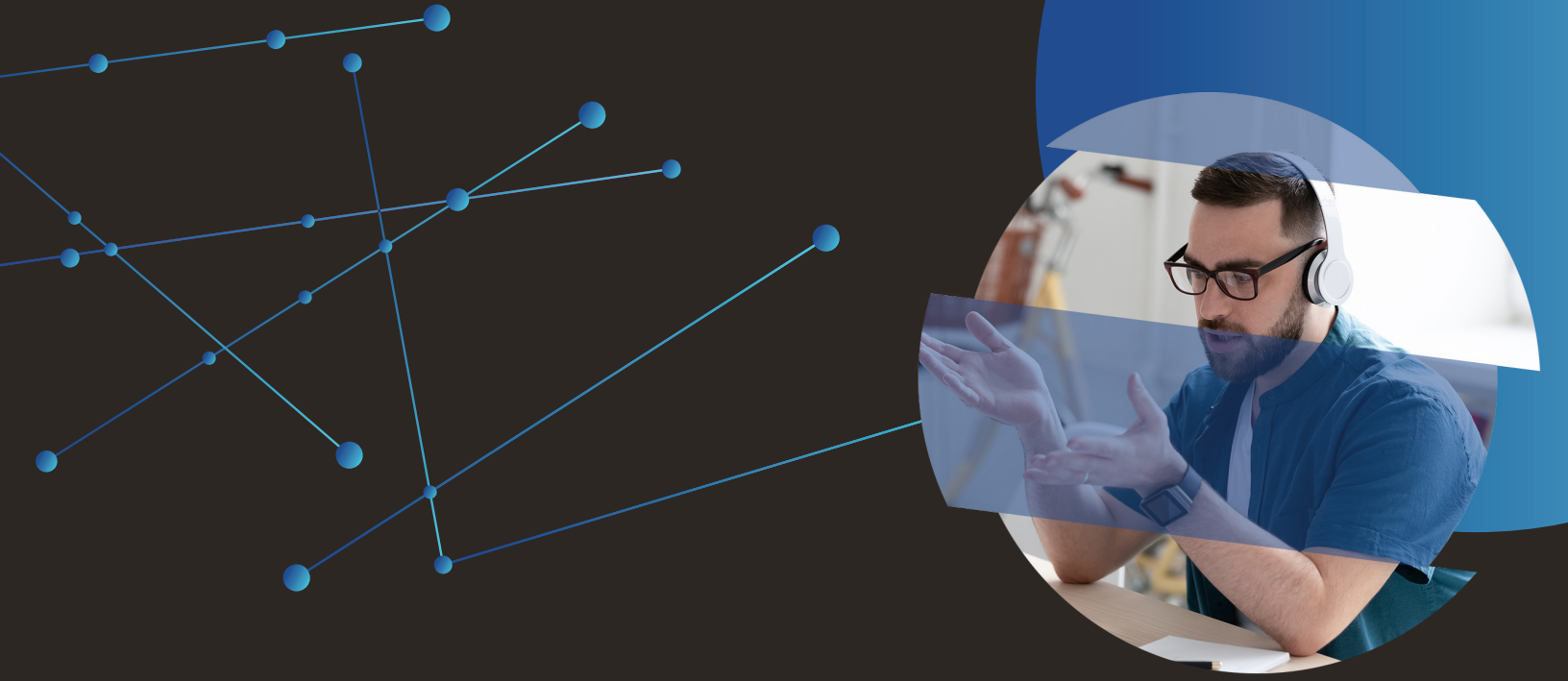

Oracle's Automation Database

Is it the Right Fit for You?

Pythian

love your data

Oracle's Automation Database

Is it the Right Fit for You?

Many organizations are considering how the latest generation of managed database services can add value. Oracle is heavily promoting its new Autonomous Database suite of services while competitors such as Google are trying to compete with similar but different cloud-based and fully-managed offerings.

What is Oracle Autonomous Database?

The Oracle Autonomous Database (ADB) isn't a product or service. Rather, it's an umbrella term that refers to four (at the time of writing) underlying services:

- the Autonomous Data Warehouse (ADW)
- the Autonomous Transaction Processing database (ATP)
- the Autonomous APEX Database, and Autonomous JSON Database

The Autonomous Data Warehouse was introduced first in 2017 and to add complexity to the discussion, is often (or was originally) referred to as Autonomous Data Warehouse Cloud (ADWC). For clarity, from this point forward, we'll refer to the ADB offering as the combination of ATP, ADW, Autonomous APEX, and Autonomous JSON.

The next logical question is: what's the difference between the various flavors? The answer is pretty simple: they're all built on common underlying technology and components (more on that to come) but are tuned and optimized differently for different workloads. The ADW has some Oracle database settings tuned for data warehouse workloads. And significantly, the underlying data is stored in the columnar format. Columnar data stores allow for effective reads of large amounts of data at the column level. They're less effective for transient data and row updates or queries that process many or all table columns.

Not surprisingly, the ATP uses Oracle database settings that are tuned for Online Transaction Processing (OLTP) type workloads that might involve some small tables, reasonably frequent deletes, and updates. It uses the more traditional row storage format.

Similarly, the Autonomous APEX service is price optimized for Oracle Application Express (APEX) applications (though APEX is still usable in the other ADB services as well), and the Autonomous JSON is optimized for JSON document storage.

Underlying Components of the Oracle Autonomous Database

A key aspect of ADB that's often overlooked is that it's fundamentally built on known technologies and components. Specifically, Oracle Exadata hardware leverages software features such as Pluggable Database (PDB) with Lockdown Profiles to provide tenant isolation and restrict the customer's ability to see or reach outside their PDB. And subsequently, Oracle introduced a dedicated version for customers requiring full isolation of the infrastructure.

A key aspect of ADB that's often overlooked is that it's fundamentally built on known technologies and components. Specifically, Oracle Exadata hardware leverages software features such as Pluggable Database (PDB) with Lockdown Profiles to provide tenant isolation and restrict the customer's ability to see or reach outside their PDB.

One can think of the ADB philosophy this way: over recent years (or decades) Oracle has built an elaborate set of hardware and software components for optimal running of Oracle databases. Exadata is at the core of their highly performant and scalable hardware. On top of that, they have horizontal scaling and redundancy with Real Application Clusters (RAC) and high availability via Data Guard. Exadata also adds storage redundancy and scalability connected with a high-speed Infiniband network and Exadata-specific software, allowing this hardware footprint to be used to its maximum potential while ensuring that a "build-your-own" similar system isn't truly an equivalent. Inside the database, they have many "somewhat" automated tuning and query stabilization features. And true multitenant capabilities through the Container Database (CDB) and Pluggable Database (PDB) architecture with resource sharing and governing software to reduce the risk of nosy neighbors hijacking the entire system.

Essentially, what Oracle did was package all of these (their most advanced technologies) together, manage them for you (the automation part), add a modest amount of new features, and make you, the customer, a consumer of all of this. The customer gets a PDB in an Oracle container database, running on RAC (after a certain number of allocated CPUs), locked down for isolation and security, on an Exadata system that Oracle procured and maintained in their own data center, and with Oracle managing the entire stack. That essentially is ADB. Some are tuned for data warehouse workloads for ADW, others for OLTP workloads for ATP and others for APEX or JSON-specific use cases.

It deviates a little from this with some of the Oracle 19c features they're adding. For example, autonomous indexing is a new feature initially released to ADB first and not available to customers with their own similar but non-ADB systems. New features may be ADB exclusive or released to ADB prior to non-ADB systems.

Benefits of the Oracle Autonomous Database

The ADB service takes care of a lot of key database management services such as:

- monitoring
- addressing underlying component errors
- automatic backups and simplified recoveries
- (some aspects of) tuning
- query stabilization
- table indexing

However, it's still providing a fully-fledged Oracle database. Usually, this is a PDB in a multi-tenant architecture (although a dedicated compute option for ATP was added). Still, even at that, the PDB is a powerful Oracle database that gives users access to many of the features of what is arguably the most powerful and feature-rich RDBMS product in the market. And the ability to monitor query performance in detail using the same set of tools that they're used to, plus some new ADB-specific cloud console-based options.

ADB doesn't automatically build applications; the autonomous part refers to the management only. But applications can still exploit everything from the concept of simple tables to advanced features such as server-side code (procedures, functions, triggers, types), complex table and index types, and much more.

While a lot of the general maintenance tasks such as backups, monitoring, checking log files, patching, and quite possibly tuning are offloaded to Oracle, there is still a lot that the DBA needs to do.

Security isolation and governance

The different deployment options on the OCI Exadata infrastructure allows to achieve the desired level of isolation from other Oracle cloud users:

- Shared Infrastructure: a simple and elastic alternative, all managed by Oracle in OCI
- Dedicated Infrastructure: all the hardware resources available only for your tenant in OCI, as you can have in a private cloud
- Cloud@Customer: the hardware is in your data center with the network configured to OCI and management under the same web interface

Limitations of the Oracle Autonomous Database

As would be expected with a managed service, there are several features and functionality that can't be used or are limited. This would include things like:

- The ability to change encryption or disable TDE.
- No control over tablespaces.
- No ability to control database instance memory sizes and configurations.
- Limited control over database initialization parameters.
- Inability to use some Oracle features.

In addition, a few commands, such as 'CREATE DATABASE LINK or ALTER TABLESPACE, are restricted by the use of PDB Lockdown Profiles.

However, the service is evolving rapidly and new capabilities are being added regularly.

Do we still need a DBA in an autonomous world?

Short answer: "YES". While a lot of the general maintenance tasks such as backups, monitoring, checking log files, patching, and quite possibly tuning are offloaded to Oracle, there is still a lot that the DBA needs to do. Your DBAs are still needed to:

- Move your data in and out of ADB.
- Create long-term retention backups/exports as may be required for regulatory or business needs.

- Perform specialized restoration or duplication activities.
- Model and design application schemas.
- Design, implement, and enforce database security (controls).
- Tune and help developers write well-structured queries.
- Manage resource scale and monitor spend.

So there's still a lot that the DBA is needed for, even in the ADB world. Some of the above may be minimal for commercial, packaged applications and more applicable with custom or in-house developed applications.

Why wouldn't we want to move into ADB?

ADB is an attractive offering for a lot of customers and specific use cases, but as with anything, there are a number of exceptions. Some reasons where ADB may not be a good fit:

- Applications with tight restrictions on the Oracle database release or version.
- Applications that require specialized (known as "one-off") patches to fix specific problems.
- Databases requiring very specific initialization parameters or tablespace settings.
- Applications that rely on some more advanced Oracle features or options which may not be included on the ADB platform.
- Workflows that involve logical replication of data (a subset of the entire
- database) out of the source database using tools such as Oracle GoldenGate.
- Businesses with a high expectation of performance consistency.

To expand on that last point: the mantra of "faster is better" isn't always true. In the database world, sometimes "consistent" is more important than "fastest". To explain: say there is a regular process that typically runs for six hours every week. One might think that there would be cause for concern if the process runs abnormally long. Certainly an exceptionally long run (or even just "longer than usual" runtime) is concerning, but even more concern might be raised when it's the other way around and the job runs for noticeably shorter than expected.

The business or customers (internal or external) expect that duration and get used to it. If the ADB autonomously tunes that process or re-indexes the underlying tables for performance, is that actually a good thing? Particularly since it happens transparently and without notification (aka "autonomously"). If, in one particular week,a process, with the expectation of a six-hour batch run, suddenly runs in just 30 minutes, it may cause undo panic or concern. Business users might think "something must have gone wrong" and technical teams might start hunting for an error, when one doesn't exist.

Business users might think "something must have gone wrong" and technical teams might start hunting for an error, when one doesn't exist.

Conclusion

The Oracle ADB suite of managed database services is an attractive offering for many, but not all Oracle Database customers. It's important to remember that this is currently an Oracle offering – in OCI or in your own datacenter using Exadata Cloud@Customer. ADB isn't a feature that you can install in your own hardware as you used to do with the database on-premises. Oracle feels that it can only provide "autonomous management" if they fully control all of the underlying components—which is already done in its own cloud or in the pre-configured Exadata hardware.

What's great about ADB is that it allows customers to leverage much of Oracle's most advanced hardware and software with minimal effort - just a few clicks to get up and running. But the Oracle DBA is still needed for many tasks, although maybe with this new paradigm, the name "Database Administrator" needs to evolve. Many groups are discussing what the most appropriate new title is and a consistent new term (for example, Data Platform Engineer) is yet to emerge. Regardless of title, a technical expert is still required as if the schema and queries are poorly designed, secured, and implemented. ADB isn't going to magically fix that.

Finally, ADB is constantly evolving. New features such as CPU autoscaling and dedicated infrastructure that weren't originally included were added, and others such as automatic indexing are available with the newest Oracle Database release.

To most, the evolution of features and improvements in performance is considered a good thing, and ADB offers that in a fully managed stack. But others may value consistency. The bottom line is, like with many things, understanding what you're getting into is paramount for success.

With over 20 years of experience managing the world's most complex Oracle systems, Pythian's certified experts can help you maximize your investment in Oracle technologies. Pythian's certified Oracle experts configure your architecture to suit your unique processes and continuously maximize the performance of your environment with our renowned 24×7×365 support. By combining the platform with the people who know it best, we can ensure constant system optimization, network monitoring, and performance that you can depend on.

Contact us **to speak with our Pythian experts.**

Simon Pane
Principal Consultant
Oracle ACE Pro
Google Cloud Champion Innovator
Co-Author

Simon is an accomplished Principal Consultant, who has developed a multitude of complex solutions for Pythian clients. He leverages his understanding of the industry and technologies such as Oracle, SQL Server, Linux, Oracle Cloud, AWS and more, to propose timely solutions that best suit the needs of clients. As a technology enthusiast, Simon is a highly sought after speaker at many user groups and international conferences.

Nelson Calero
Principal Consultant
Oracle ACE Director
Oracle Certified Expert

Based out of Montevideo, Uruguay, Nelson enjoys a beautiful ocean view working from home as an Oracle Database Consultant. In addition to being the Founder and President of the Uruguayan Oracle User Group (UYOUG), Nelson has contributed to the community by frequently speaking at industry events, and working as an Instructor at Oracle University. He loves facing new challenges at work, and confronts them by pushing boundaries to create new solutions. When he isn't working, Nelson enjoys photography, spending time with his family and travelling.

About Pythian

Founded in 1997, Pythian is a data and analytics services company that helps organizations transform how they compete and win by helping them turn data into valuable insights, predictions, and products. From cloud automation to machine learning, Pythian designs, implements, and supports customized solutions to the toughest data challenges.

Contact us

+1-866-798-4426 | info@pythian.com | Linkedin | Twitter

Offices

Ottawa, Canada
Minneapolis, USA
New York City, USA
London, England
Hyderabad, India
Bangalore, India

Pythian

love your data