Open Source Databases

Latest additions to OCI Databases portfolio

Alexandre Fagundes, Cloud Architect
alexandre.af.fagundes@oracle.com

June 2024

Agenda

Why Open Source?

MySQL

PostgreSQL

NoSQL

OCI Cache with Redis

Open Search

Why is Oracle investing in Open Source Data Platforms?

OCI is committed to offer the widest choice to our customers

MANAGED CONTAINERS

CONTAINERS

INFRASTRUCTURE AS CODE

AUTONOMOUS LINUX

DATA SCIENCE

DATABASE

BIG DATA

ANALYTICS ENGINE

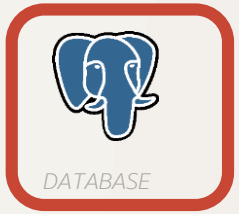
CACHING

MACHINE LEARNING FRAMEWORK

DATA COLLECTION

EVENT DELIVERY

STREAMING



DATABASE

PROGRAMMING LANGUAGE

FUNCTIONS

SEARCH

API GATEWAY

BLOCKCHAIN

Native integrations with the dev tools you're used to

DEV-OPS

AUTOMATION

INFRASTRUCTURE AS CODE

DEV-SEC-OPS

AUTOMATION

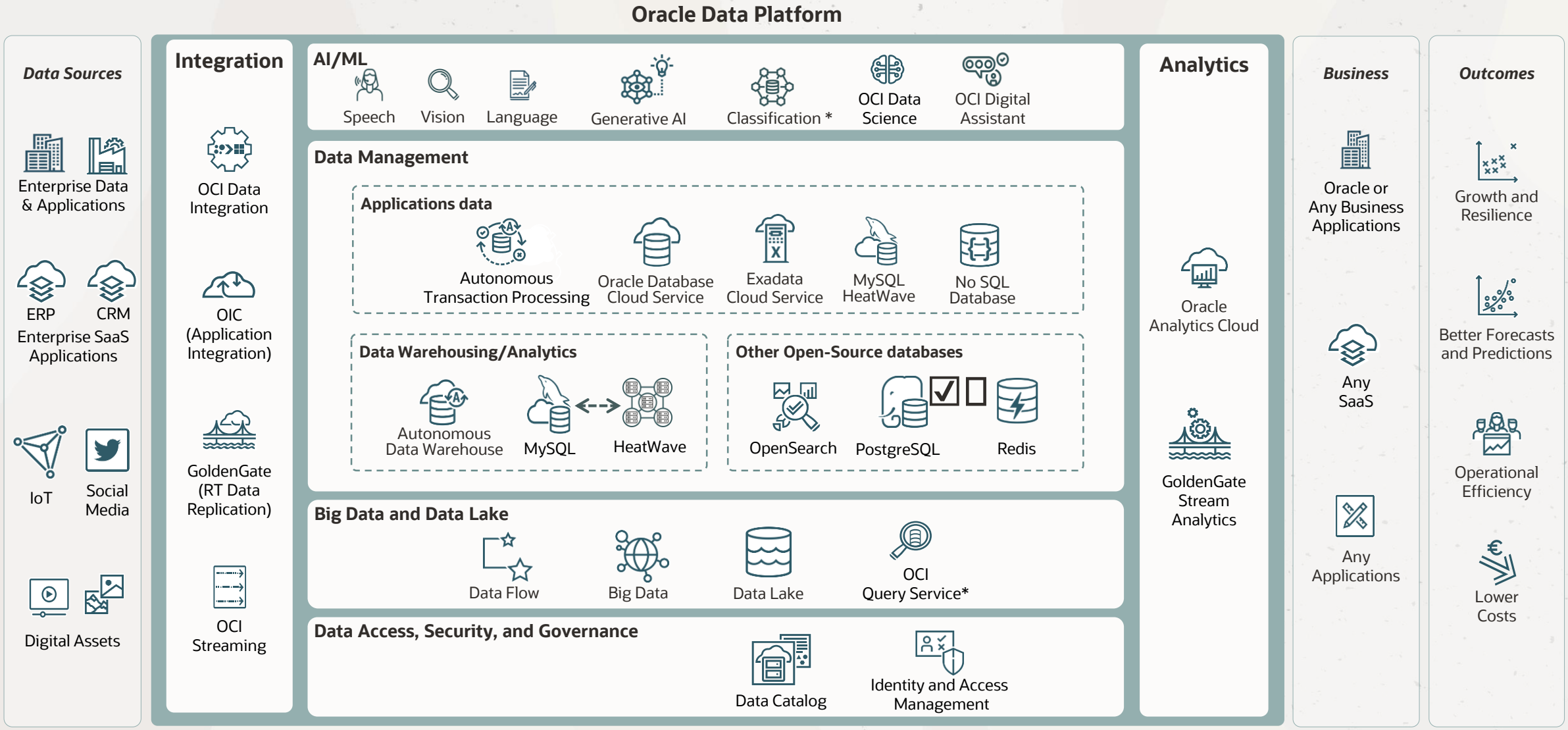
TEAM COLLABORATION

CONTAINER MANAGEMENT

PACKAGE MANAGEMENT

RISK MANAGEMENT

Oracle Data Platform: A complete suite of services



MySQL Heatwave

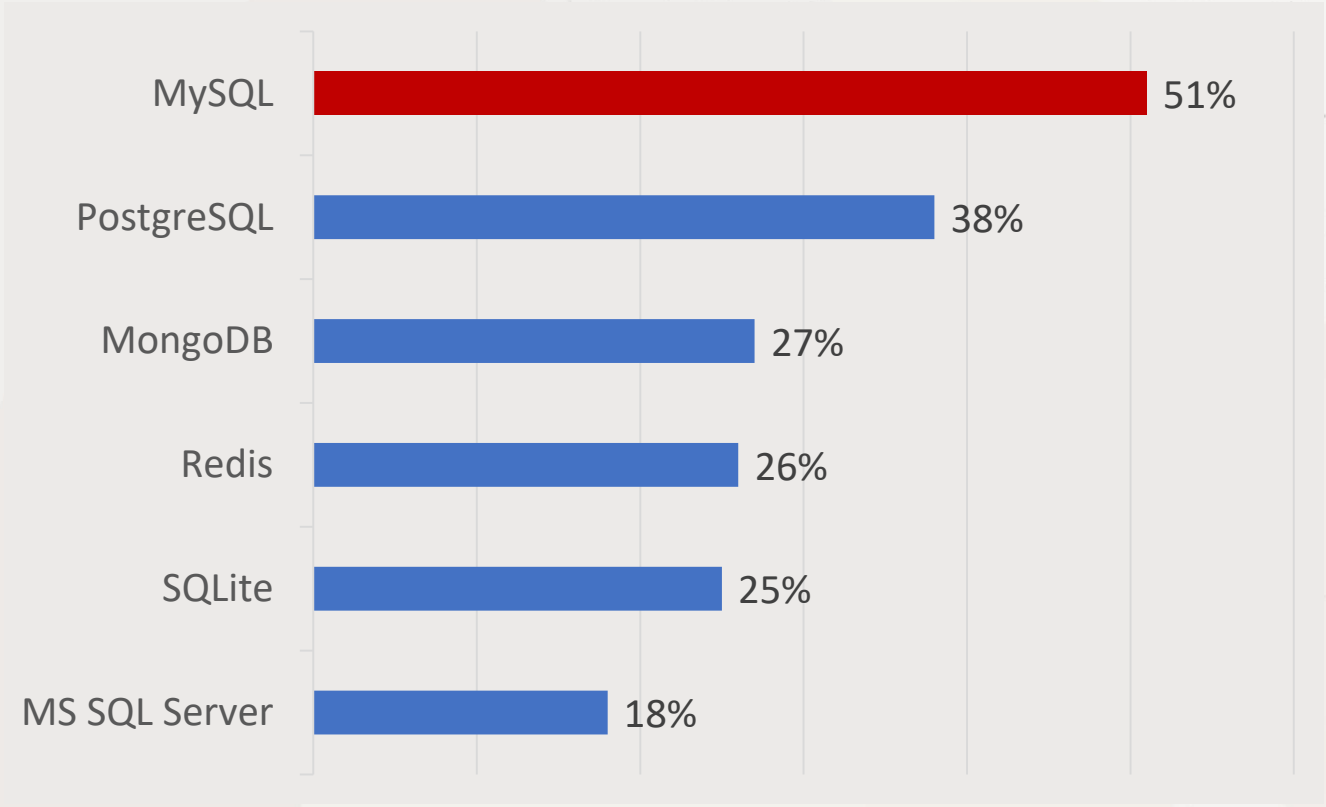
MySQL is the #1 Open Source Database

Rank Mar 2024	Rank Feb 2024	Rank Mar 2023	DBMS	Database Model	Mar 2024
1.	1.	1.	Oracle	Relational, Multi-model	1221.06
2.	2.	2.	MySQL	Relational, Multi-model	1101.50
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	845.81
4.	4.	4.	PostgreSQL	Relational, Multi-model	634.91
5.	5.	5.	MongoDB	Document, Multi-model	424.53

DB-ENGINES

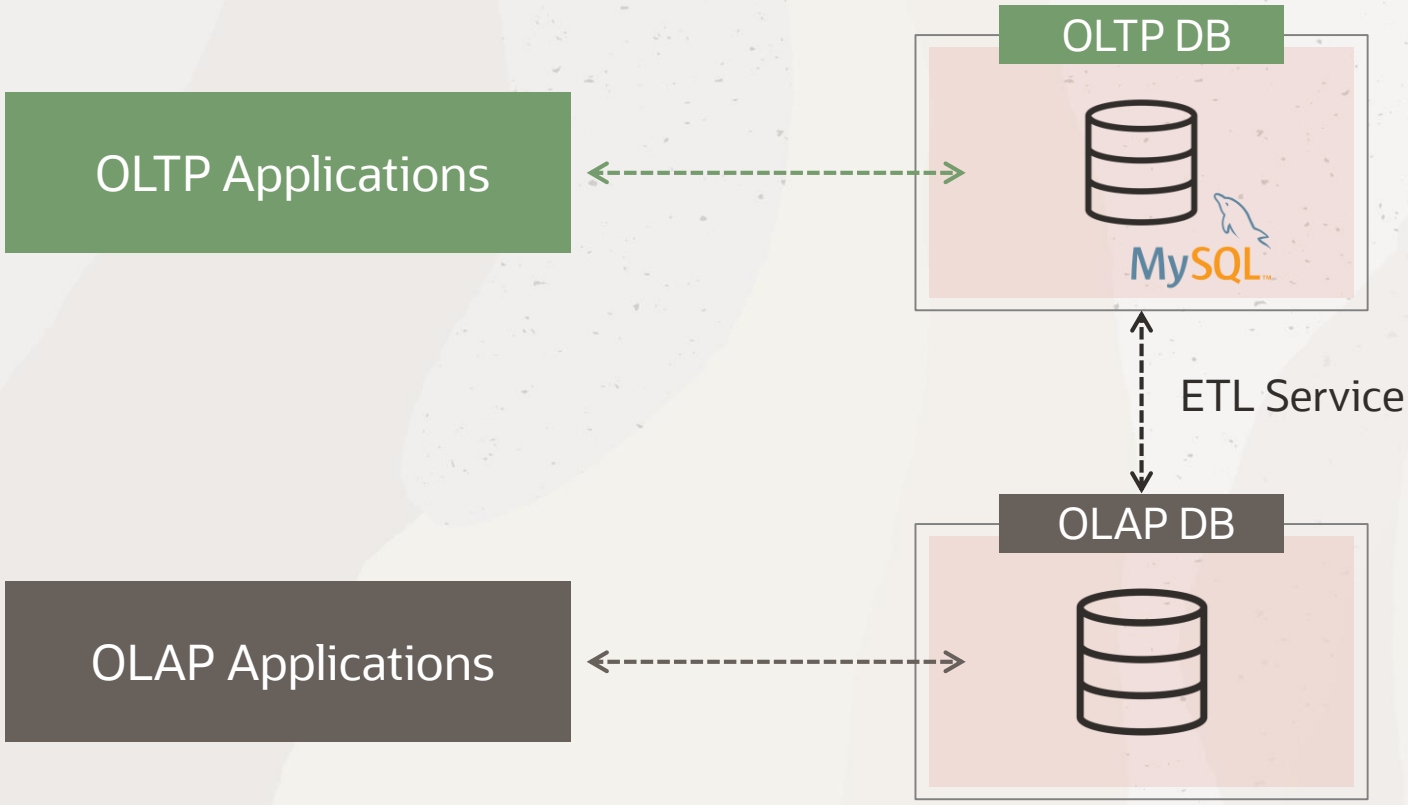
MySQL is the most popular database for developers

Which databases have you used in the last 12 months?



Jetbrains survey 2024

MySQL is optimized for OLTP, not designed for analytic processing



Separate analytics database

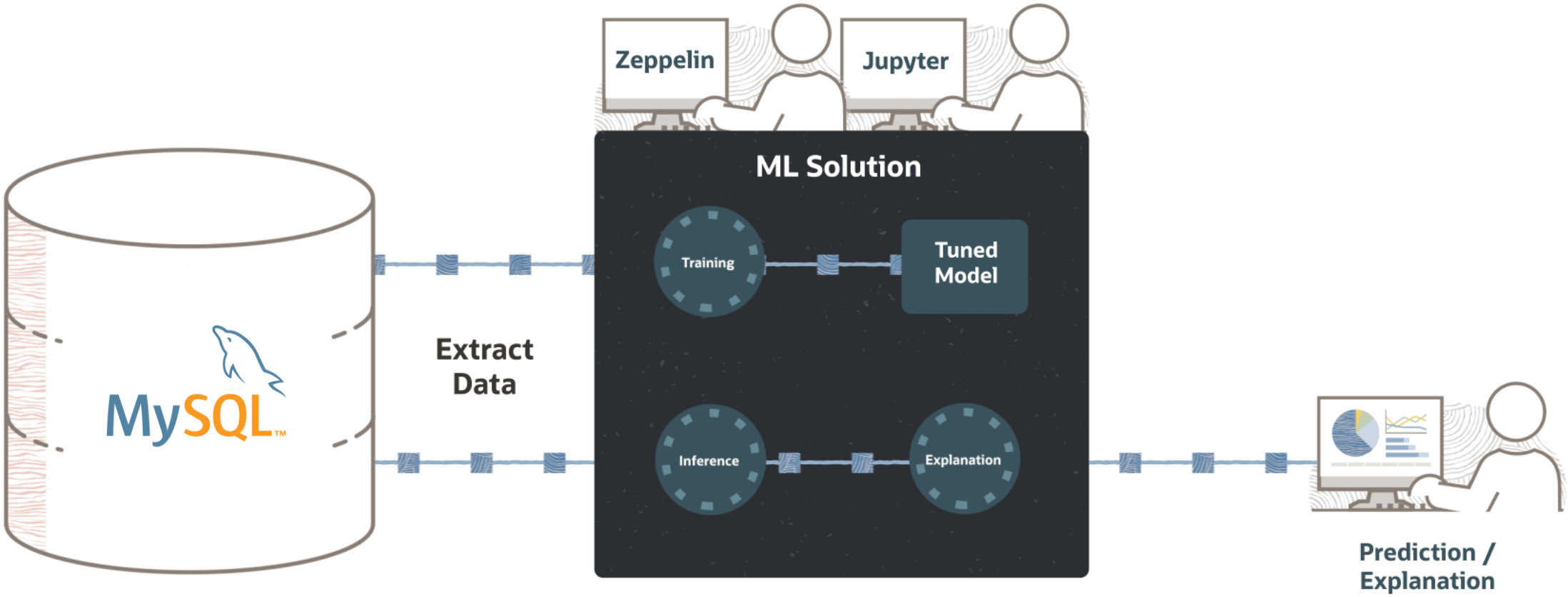
Complex ETL

No real-time analytics

Security & compliance risks

Increased costs

Need to ETL data to a separate ML solution for training and inference

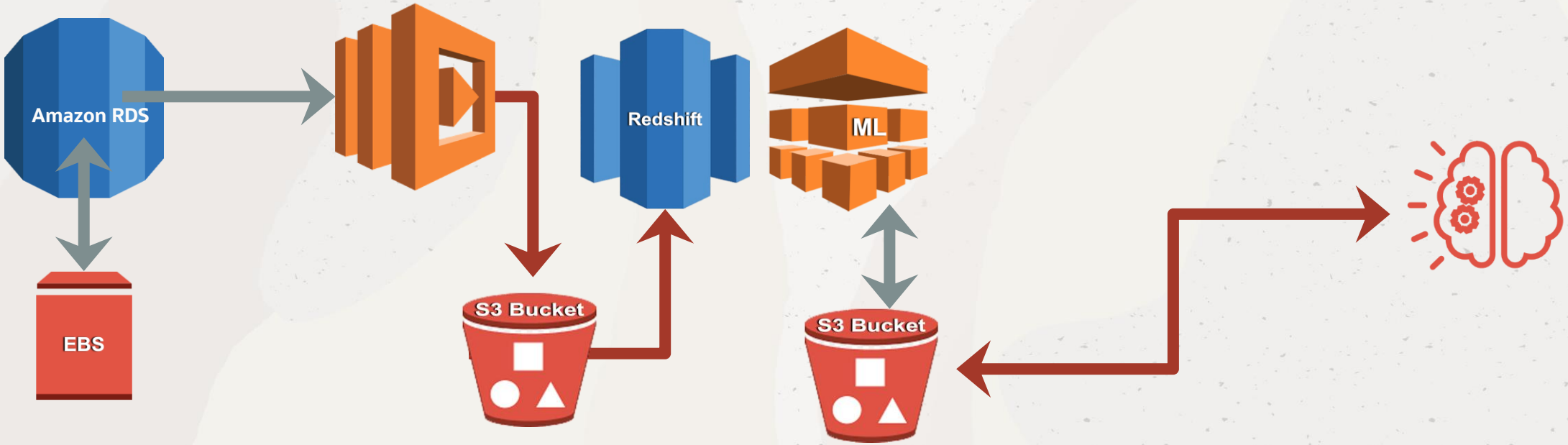


- Complex, time-consuming
- Increases costs and risks
- Need to learn new tools/languages

And it gets worse when using other databases…

The pain of using AWS services

Amazon Redshift with RDS and ML Modeling

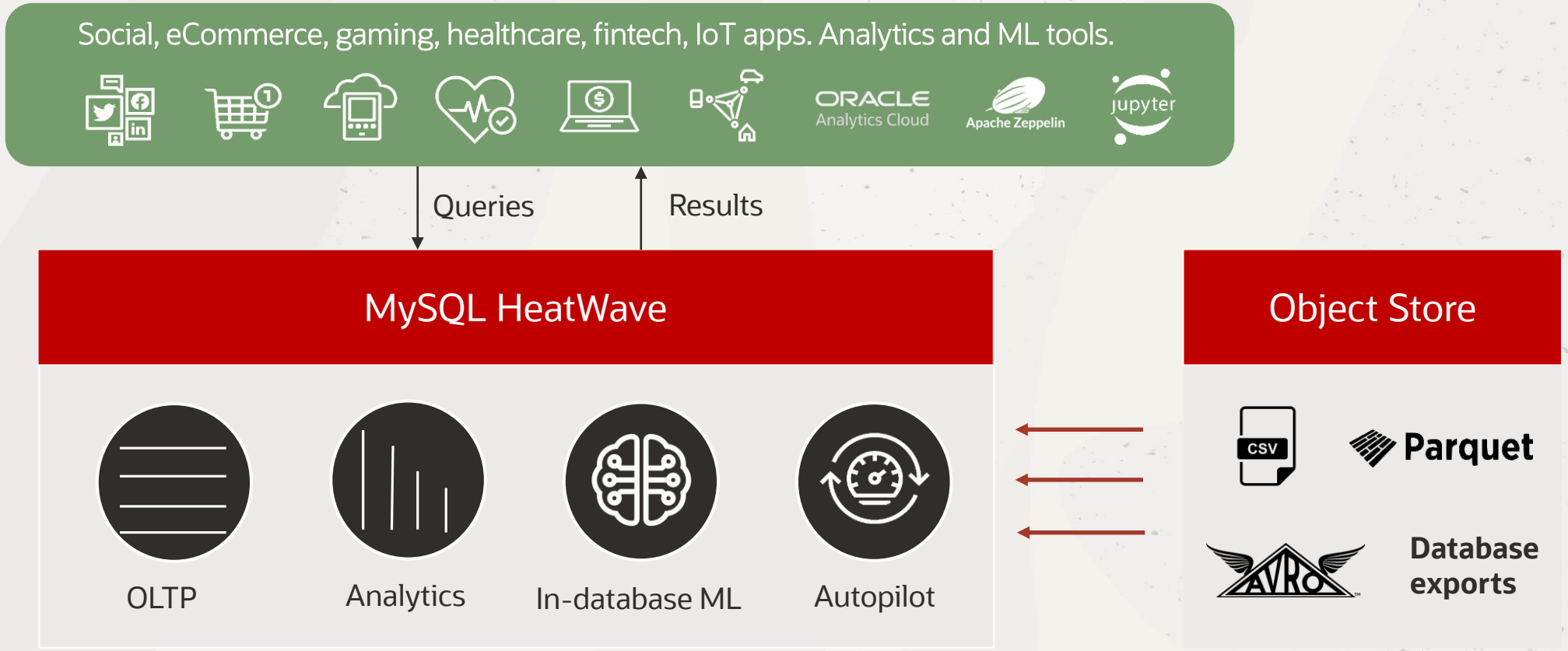


RDS to Redshift Pipeline
Utilizing Lambda, Glue, & S3

Redshift ML does NOT provide in-database ML; exports data to SageMaker via Amazon S3

MySQL HeatWave overview

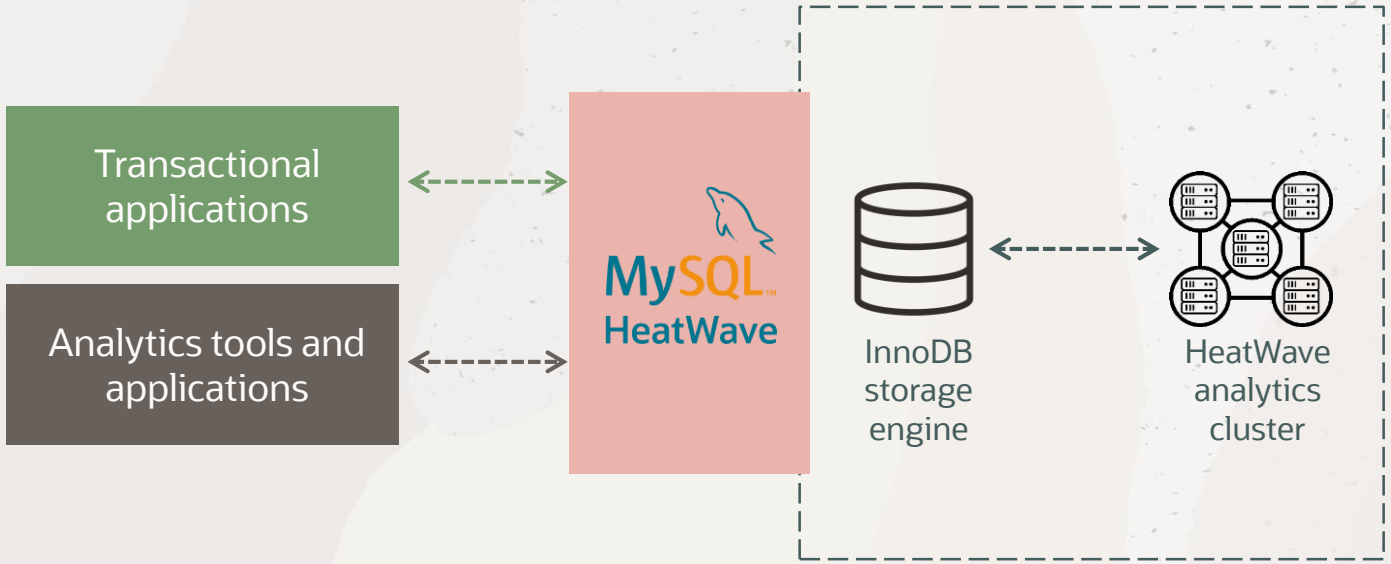
Transactions, real-time analytics across data warehouse and data lake, and machine learning in one database service



Data remains in object store, processing is done in HeatWave

For both non-MySQL and MySQL workloads

One database is better than two



- One service for OTLP & OLAP
- No ETL duplication
- Unmatched performance, at a fraction of the cost
- Real-time analytics
- Improved security
- Applications work without changes

MySQL

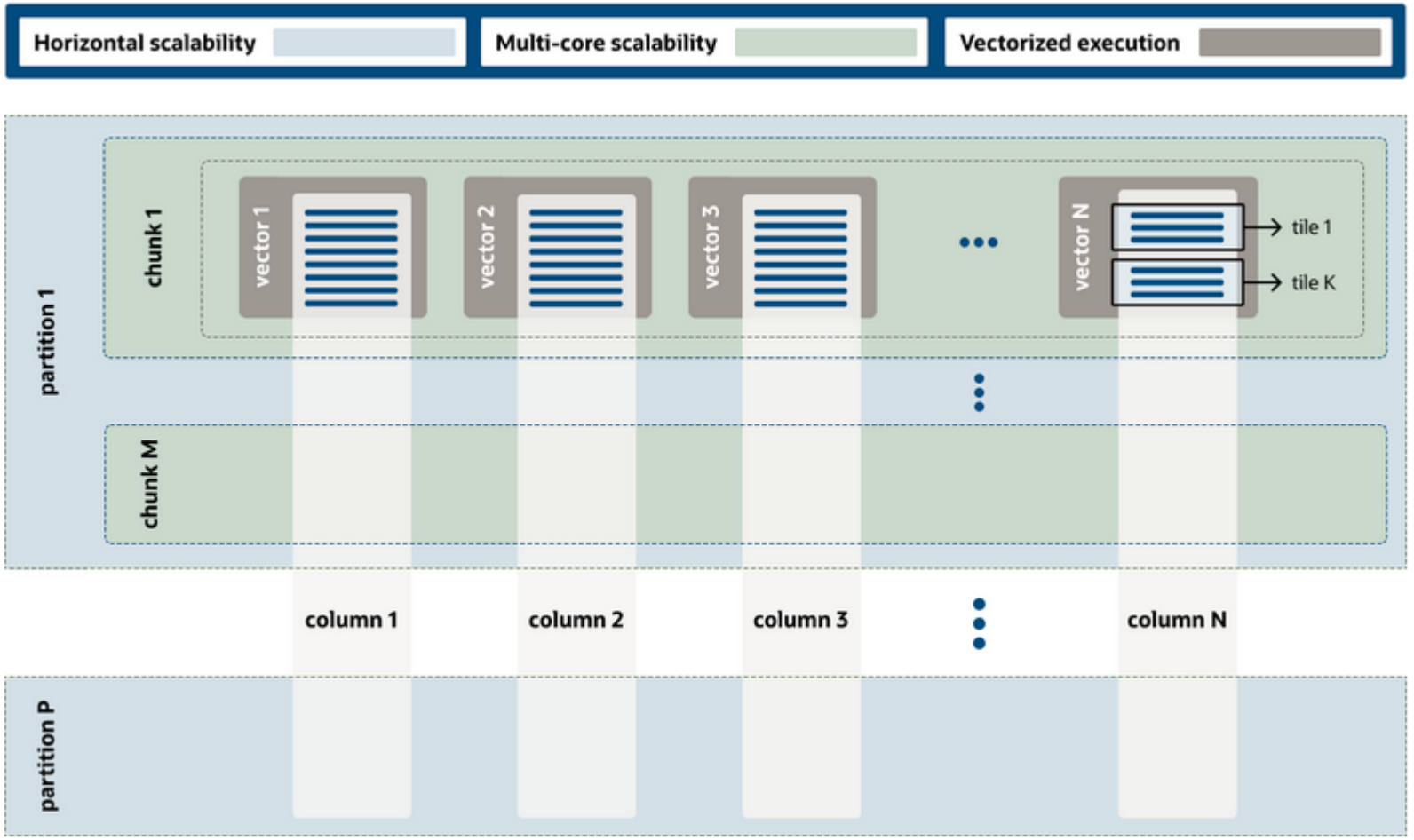
1>2 with MySQL HeatWave

MySQL HeatWave: fully managed database service

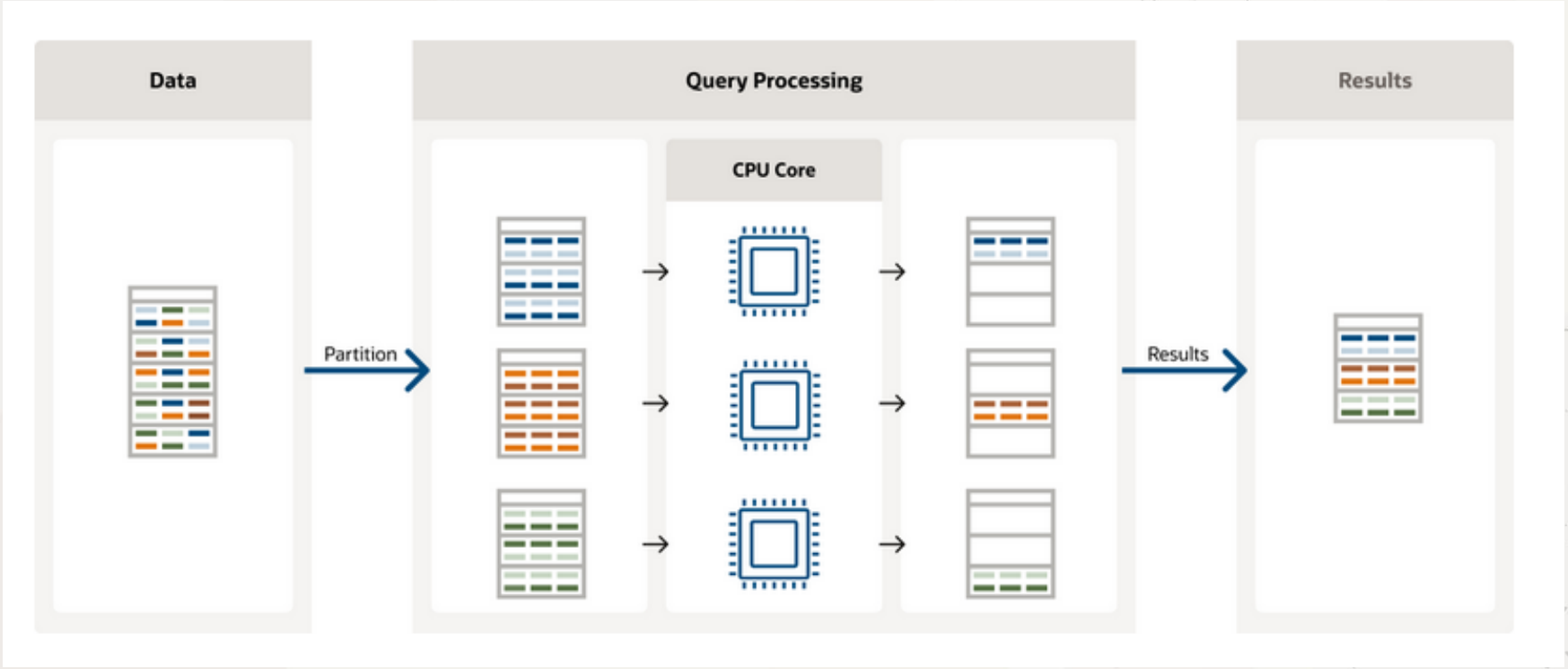
100% developed, managed, and supported by Oracle

	Automation	MySQL HeatWave
Database	High Availability	✓
	Read Replicas	✓
	Backup	✓
	Query Acceleration	✓
	MySQL AutoPilot	✓
	HeatWave AutoML	✓
	HeatWave Lakehouse	✓
	Security Patch & Upgrade	✓
	Provision & Configure	✓
OS	OS Security Patch & Upgrade	✓
	OS Installation	✓
Server	Hardware Provisioning & Maintenance	✓
Storage	Storage Provisioning & Maintenance	✓
Data Center	Rack & Space	✓
	Power, HVAC, Networking	✓

In-Memory hybrid columnar processing



Massively parallel architecture

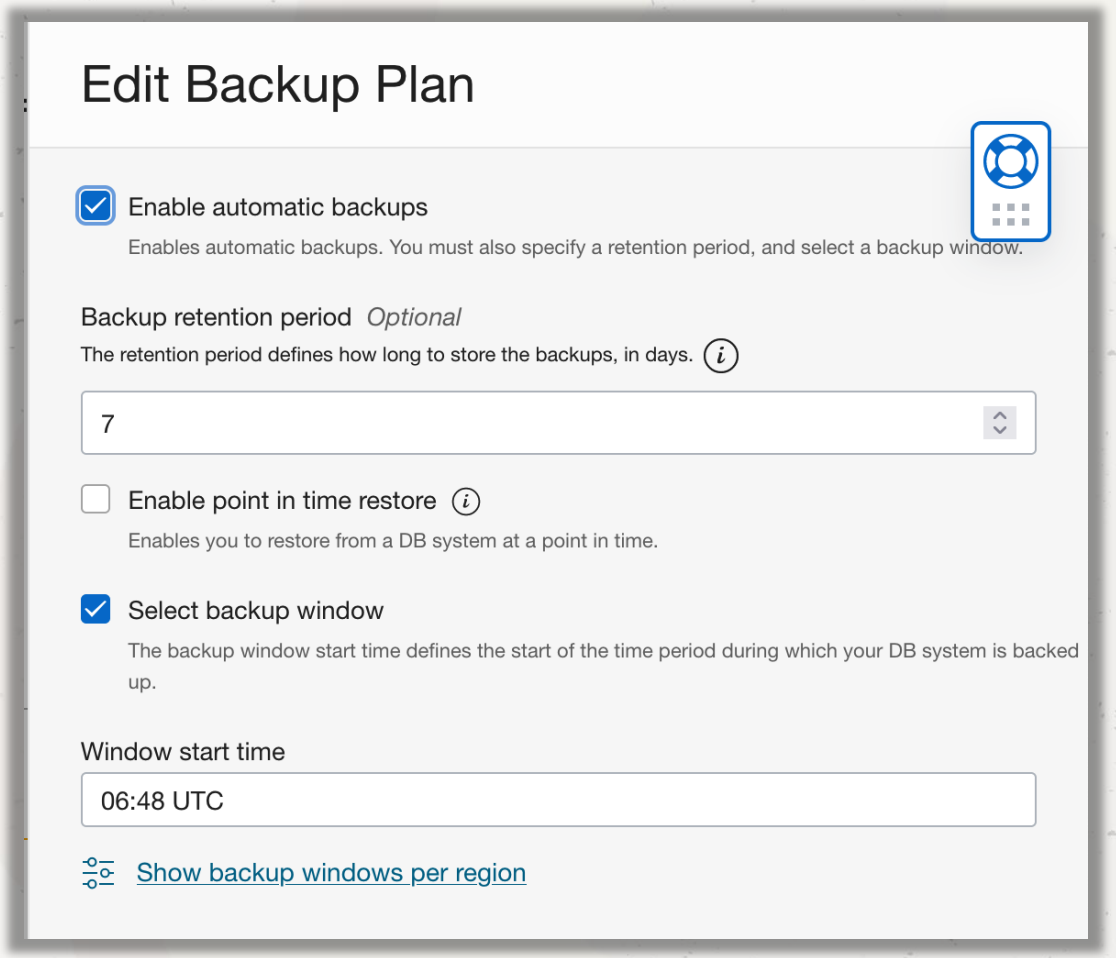


- High-fanout partitioning
- Machines & CPU cores can further process partitioned data in parallel
- Optimized for cache size and memory hierarchy of underlying hardware

Backups

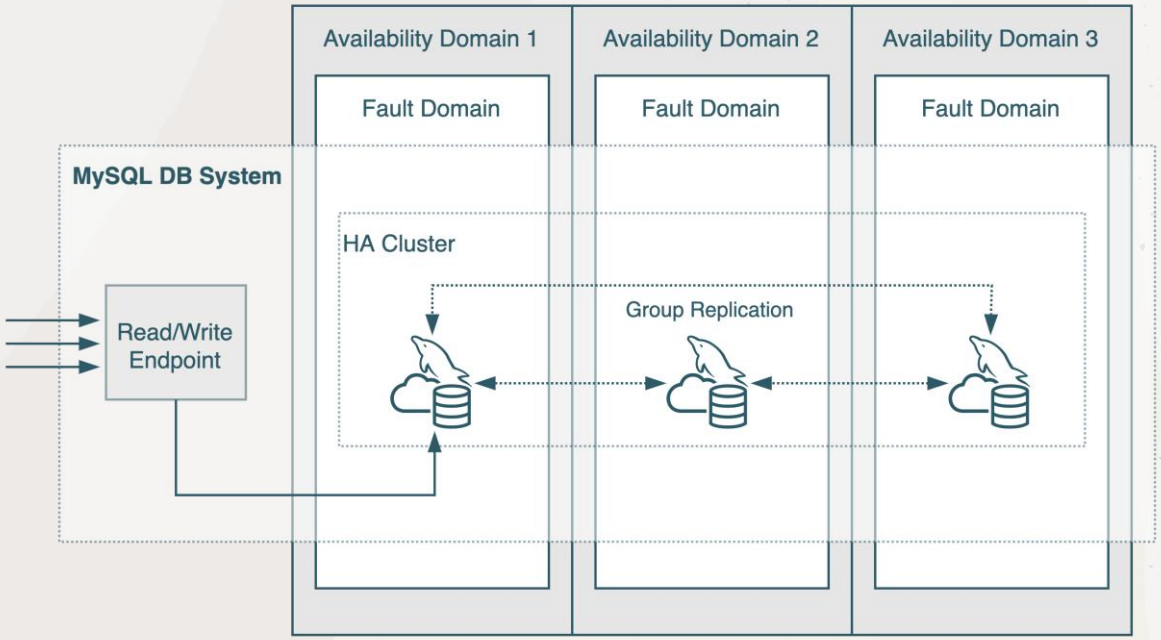
Manual or Automatic

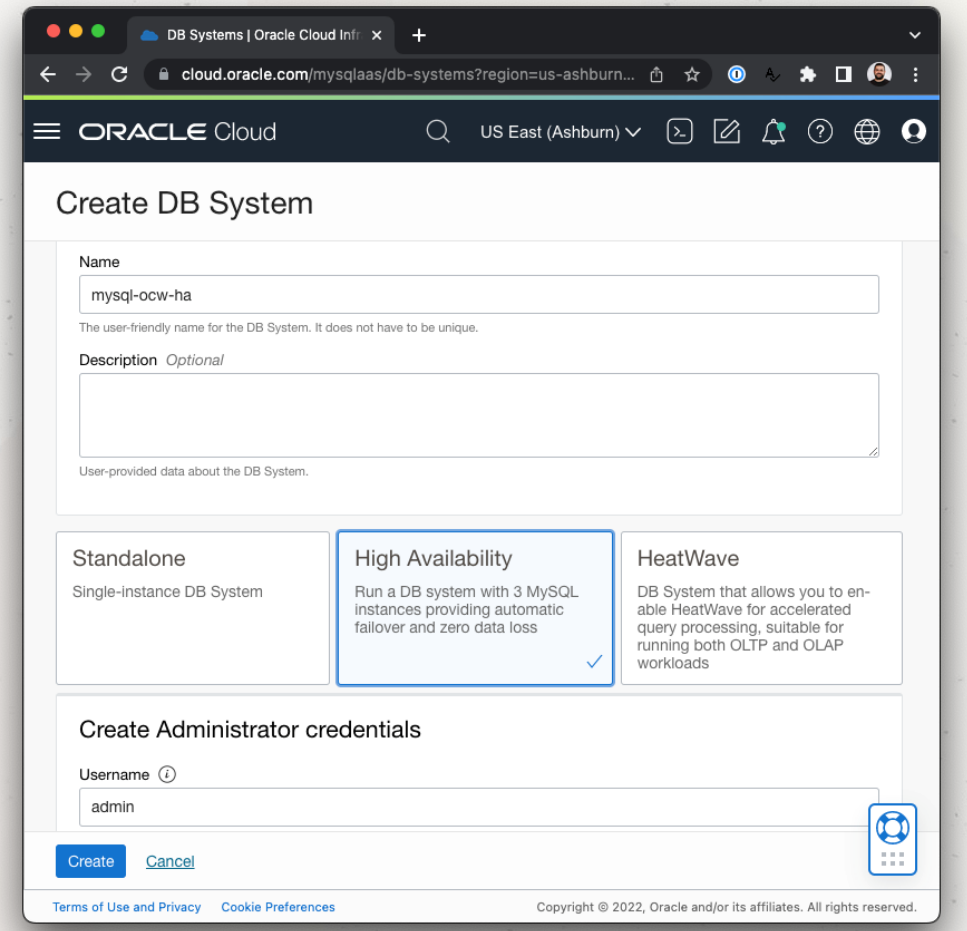
- Retention Period
- When to Backup
- Full or Incremental
- Point-in-Time Recovery (only non-HA DB Systems)



High Availability

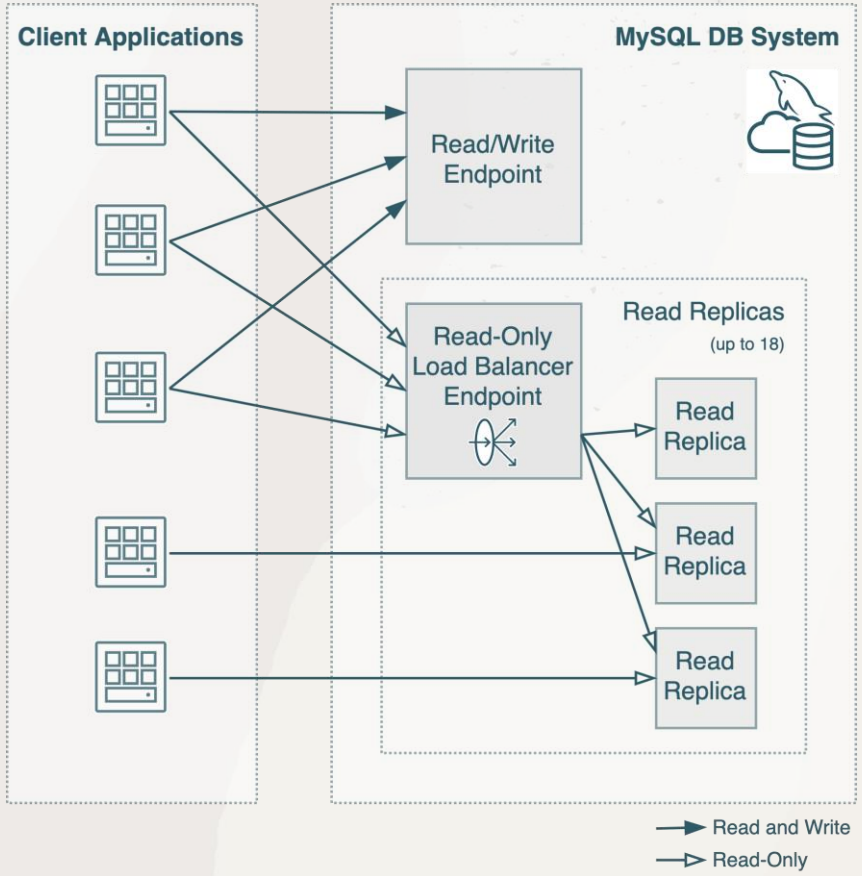
Single-click HA





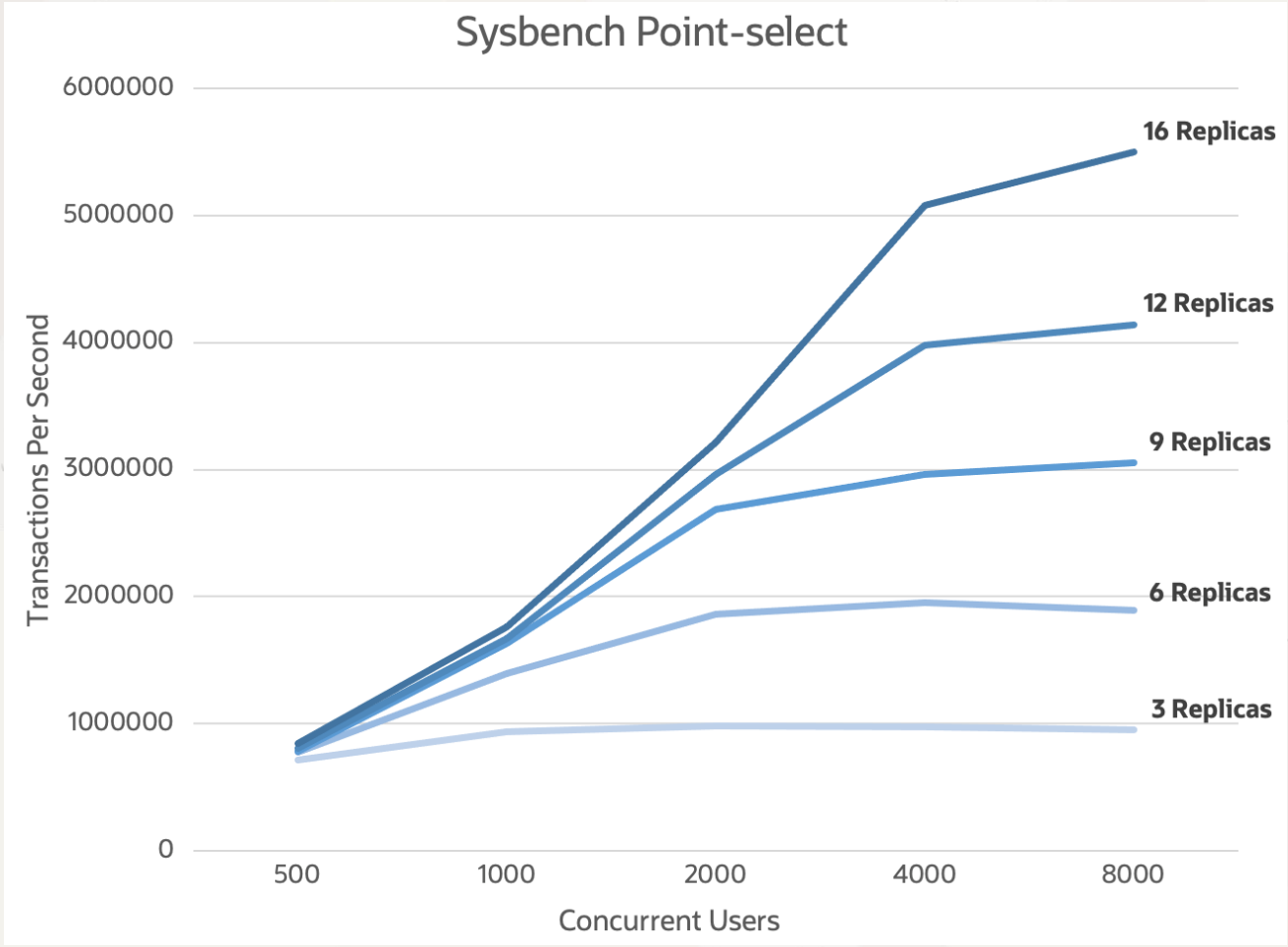
Read Replicas

Scale Read Workloads



- Increase capacity for read-intensive workloads
- Add and remove Read Replicas for horizontal elasticity
- Easy to deploy and maintain
- Built-in Load Balancer for the read-only endpoint
- HA Cluster is ready for switchover or failover

Read Scalability

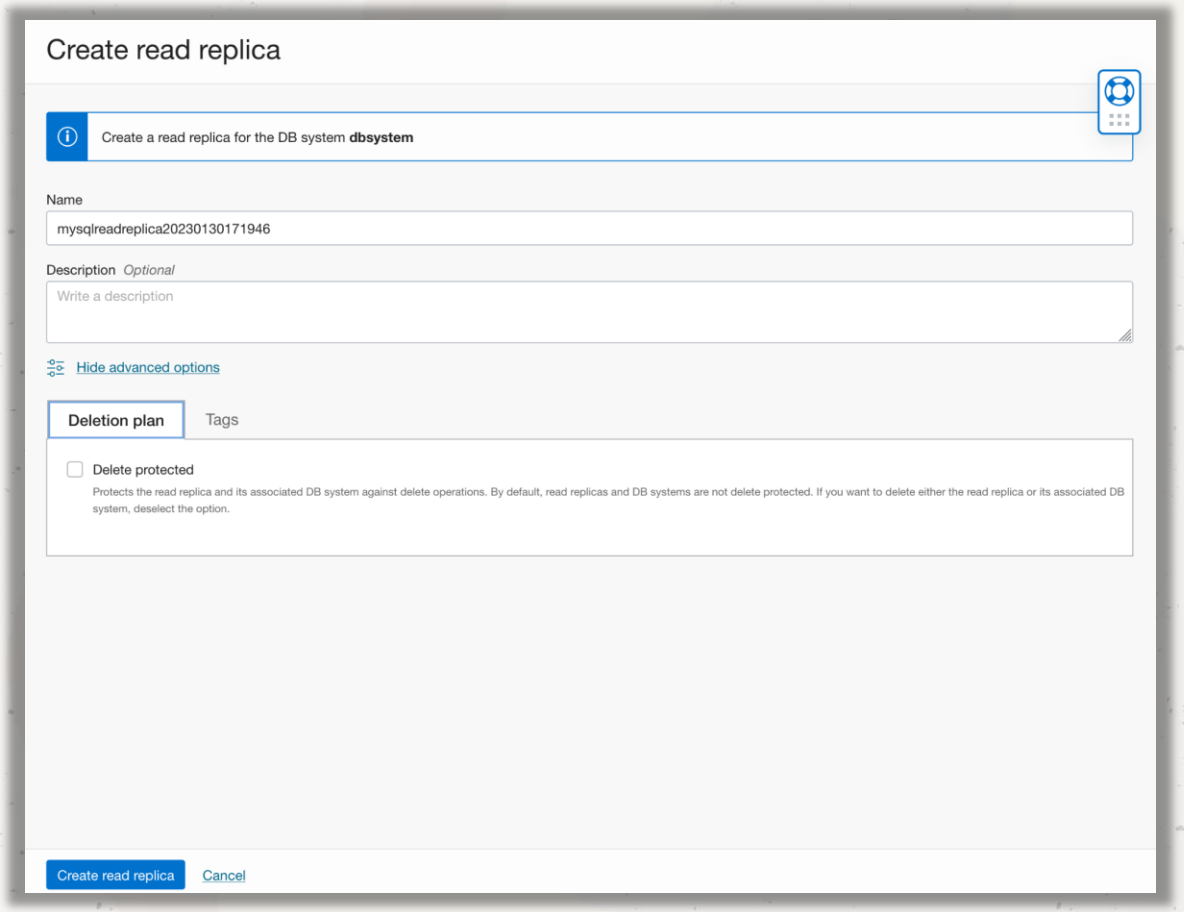


Read Replicas

Single-click Read Replicas

- A single click creates a Read Replica
  - Provision
  - Launch
  - Setup Replication
  - Monitor and Manage

- Read Replicas are associated with a DB System
  - RO endpoints in the DB System
  - Up to 18 max per DB System
  - Requires a shape of 4 OCPUs or larger
  - CLI, SDK and Terraform support

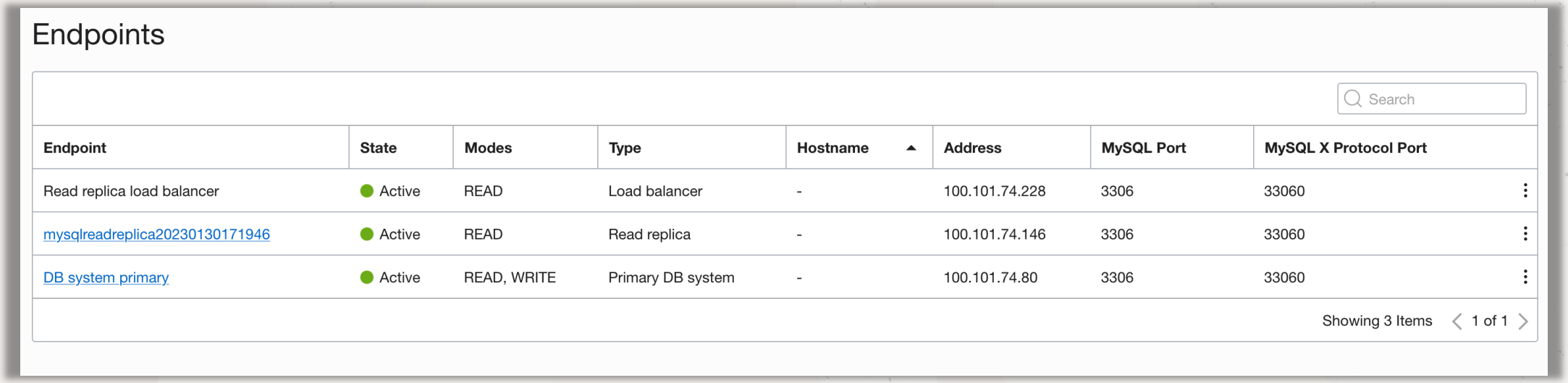


Load Balancer

Use Your Replicas Efficiently

When using Read Replicas a Load Balancer Endpoint is automatically provisioned in your DB System.

- Managed by the service
- Materializes as a Read-Only endpoint
- Round robins traffic across Read Replicas
- Manages Read Replica backends automatically

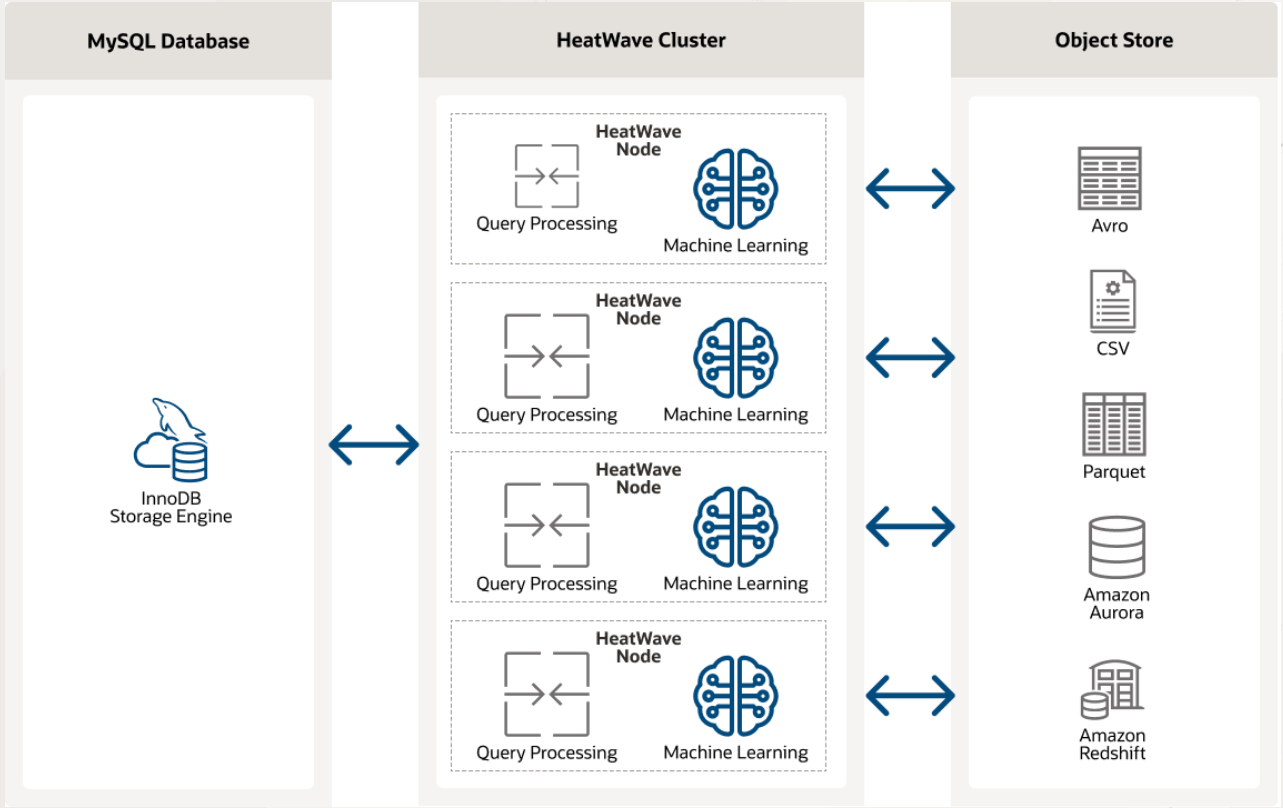


Massive amount of data stored in files

- Databases are systems of record
- Files are repository for other types of data (e.g IoT, web content, log files)
- Over 80% of the data we generate is in files
- 99.5% of collected data remains unused
  - Lack of time, resources, and expertise to process different data formats across different data sources

MySQL HeatWave Lakehouse

Query half a PB data in the object store—in a variety of file formats



- Query data in MySQL, in the object store, or across both—using standard SQL syntax
- Up to 500 TB of data—the HeatWave cluster scales to 512 nodes
- Querying the data in the object store is as fast as querying the database – **an industry first!**
- Scale out data processing in the object store, data is not copied to the MySQL Database: for both MySQL and non-MySQL workloads

Very simple to query files in the object store

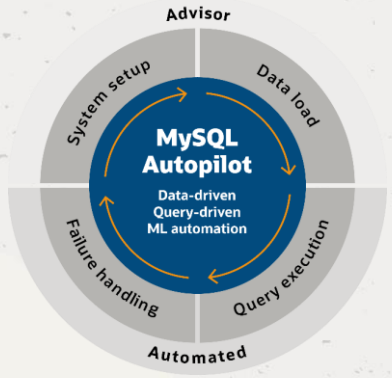
STANDARD SQL syntax generated by MySQL Autopilot, no human required

1. System Setup

- Run MySQL Autopilot on object store to determine cluster size and schema mapping
- Execute DDLs generated by Autopilot

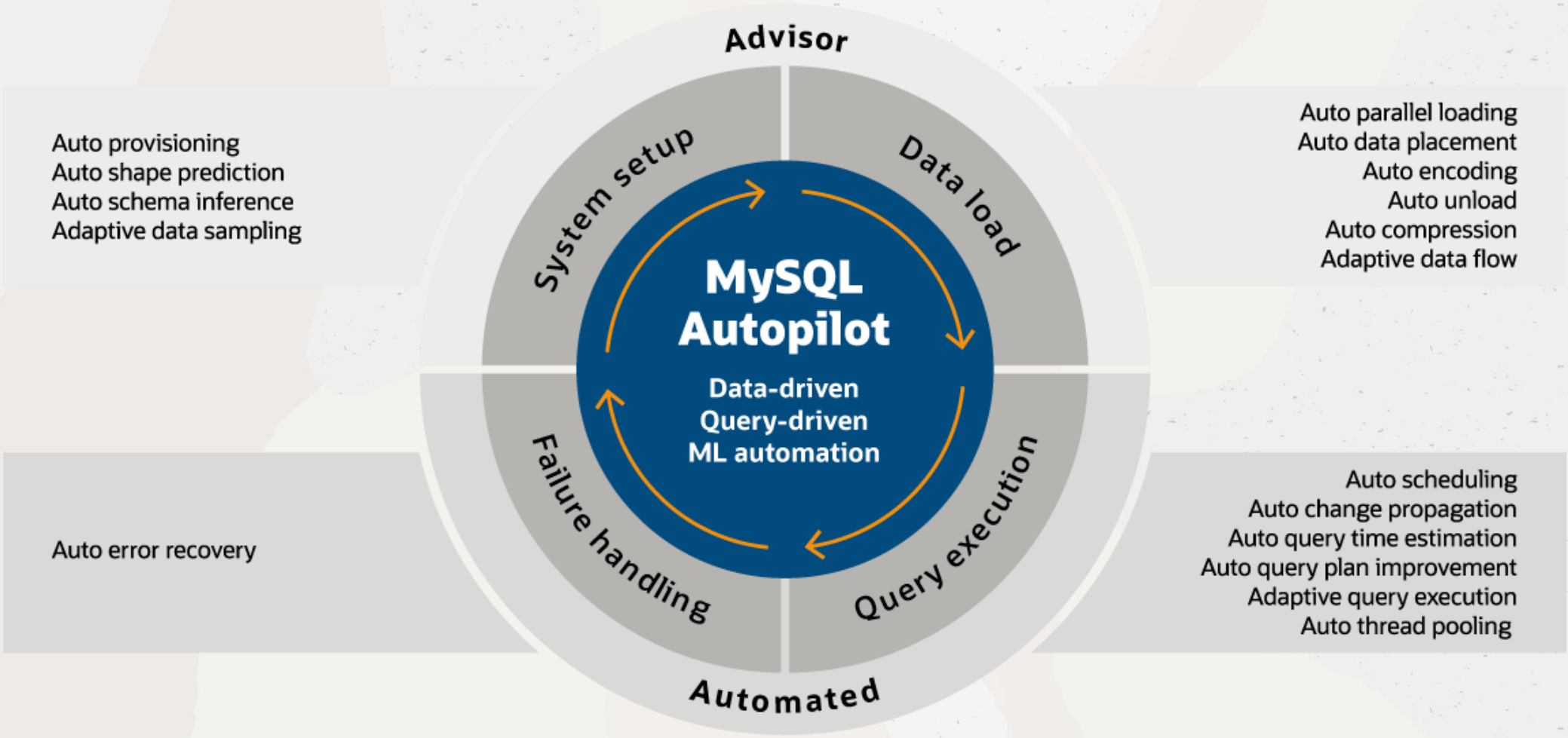
2. Run query across files and tables

- ```
  mysql> SELECT count(*) FROM Sensor, SALES WHERE Sensor.degrees > 30 AND Sensor.date =
  SALES.date;
  ```


  ```

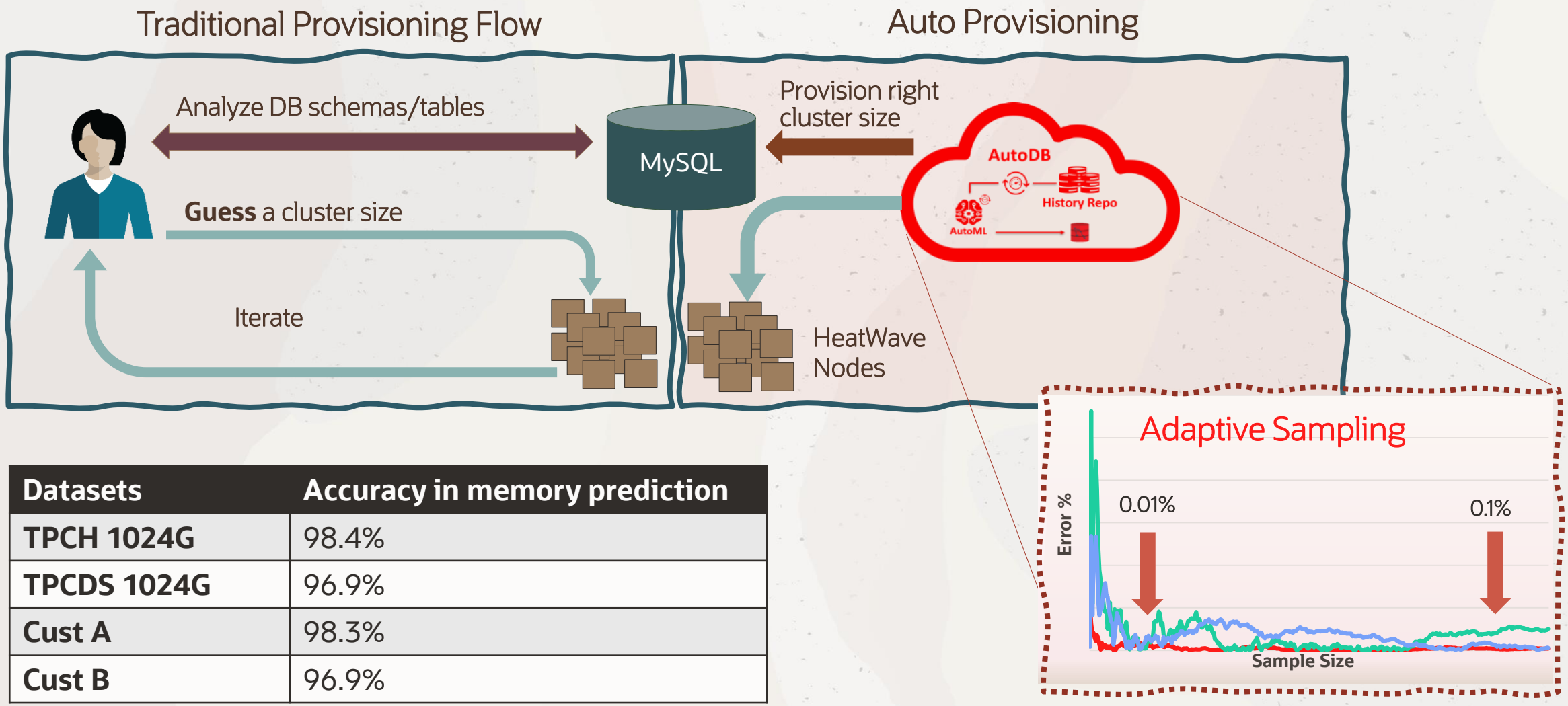
Machine learning-powered automation for MySQL HeatWave

High query performance at scale, higher OLTP throughput, and the best price performance



Auto Provisioning

Machine learning prediction of memory usage to estimate cluster size



Datasets	Accuracy in memory prediction
TPCH 1024G	98.4%
TPCDS 1024G	96.9%
Cust A	98.3%
Cust B	96.9%

Auto provisioning with MySQL HeatWave Lakehouse

How to determine the right cluster size required for processing data in object store?

```
CALL sys.heatwave_load(@db_list, @options);
```

Output | Result #4 | Result #4.2 | Result #4.3 | **Result #4.4** | Result #4.5

CAPACITY ESTIMATION

Default encoding for string columns: VARLEN (unless specified in the schema)

Estimating memory footprint for 1 schema(s)

SCHEMA NAME	TOTAL OFFLOADABLE TABLES	ESTIMATED HEATWAVE NODE FOOTPRINT	ESTIMATED MYSQL NODE FOOTPRINT	TOTAL STRING COLUMNS	DICTIONARY ENCODED COLUMNS	VARLEN ENCODED COLUMNS	ESTIMATED LOAD TIME
`lakehouse`	6	75.49 TiB	10.50 MiB	142	0	142	1.40 h

Sufficient MySQL host memory available to load all tables.

HeatWave cluster memory might be insufficient to load all the tables.

The estimated load time assumes a cluster with sufficient size: 151 512-GB nodes

Please refer to the user manual for more details.

Auto Unload

The system recommends which tables can be unloaded based on workload history

HEATWAVE UNLOAD ADVISOR

SCHEMA NAME	TABLE NAME	REASON	MEMORY GAIN (HEATWAVE NODES)	MEMORY GAIN (HOST)
`tpch1`	`LINEITEM`	LAST QUERIED ON "2023-03-02 10:24:06"	732.00 MiB	0 bytes
`tpch1`	`ORDERS`	LAST QUERIED ON "2023-03-02 10:24:12"	164.00 MiB	0 bytes
`tpch1`	`PARTSUPP`	NEVER QUERIED	148.00 MiB	0 bytes
`tpch1`	`CUSTOMER`	NEVER QUERIED	28.00 MiB	0 bytes
`tpch1`	`PART`	NEVER QUERIED	28.00 MiB	0 bytes
`tpch1`	`NATION`	NEVER QUERIED	4.00 MiB	0 bytes
`tpch1`	`REGION`	NEVER QUERIED	4.00 MiB	0 bytes
`tpch1`	`SUPPLIER`	NEVER QUERIED	4.00 MiB	0 bytes

Recommends which tables can be unloaded

Provides explanation for the recommendation

Auto Shape Prediction

Determines the optimal instance type

Workload

MySQL Server

- InnoDB statistics
- Query Metrics
- *Traces/Logs (future)*

- MySQL Autopilot SQL interface
- Machine learning model
- Data aggregation and sampling

Predictions over time

- New statistics
- Intelligent sampling
- ML models inside the server
- Periodic prediction

Shape: MySQL 2.16

InnoDB — Heavy I/O — Core, Core, 16 GB Memory

Shape change

Shape: MySQL 4.32

InnoDB — Light I/O — Core, Core, Core, Core, 32 GB Memory

MySQL HeatWave is optimized for multiple clouds

Maximum flexibility and choice

Optimized for best price performance in each cloud

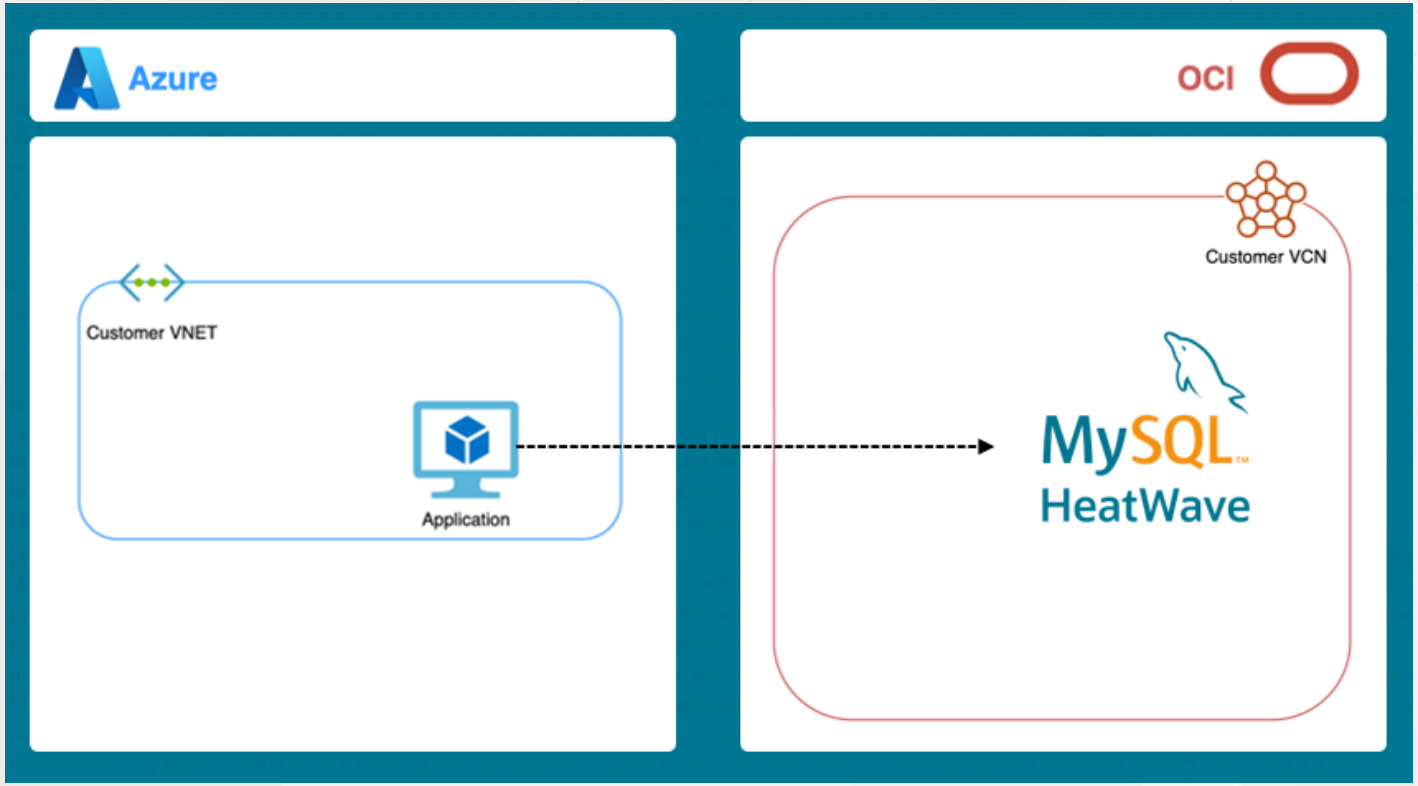
MySQL HeatWave on AWS

- MySQL HeatWave runs natively on AWS, optimized for AWS infrastructure
- Data doesn't leave AWS – saves egress cost, and avoids compliance approvals
- Lowest latency access to MySQL HeatWave
- Tight integration with the AWS ecosystem – S3, CloudWatch, PrivateLink
- Easier migration from other databases (e.g., Amazon Aurora, Redshift, Snowflake)

Combine 5 AWS services into ONE

MySQL HeatWave on Azure

Connecting to MySQL HeatWave on OCI from Azure VNET



- Familiar Azure-native user experience
- Automated identity, networking, and monitoring integration
- Private interconnect and networking with < 2 ms latency
- Use Microsoft Azure services with MySQL HeatWave
- Collaborative support

https://www.oracle.com/cloud/azure/oracle-database-for-azure

DB system Inbound and Outbound Replication

Cross-region replication

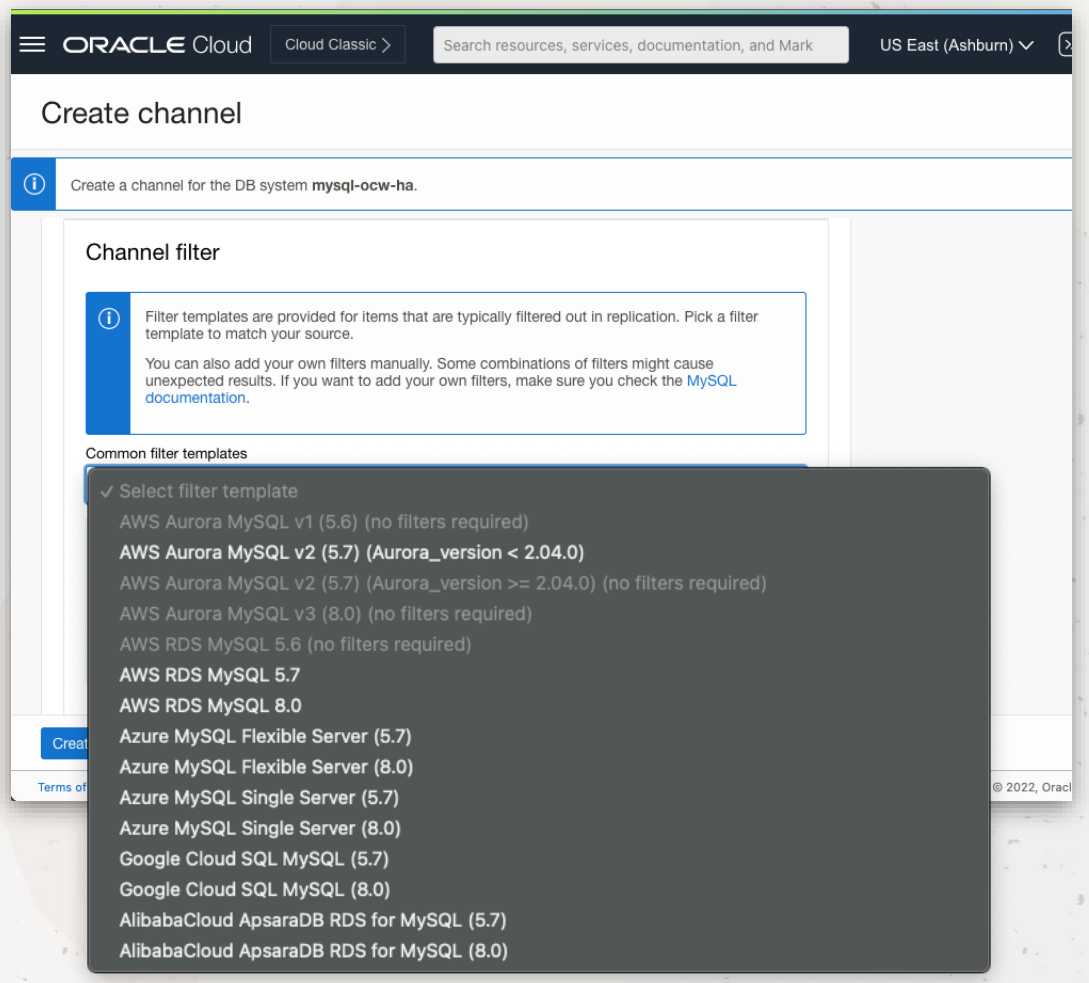
- DB System to DB System

Live Migrations

- Minimize downtime

Hybrid deployments

- On-premises and other cloud vendors
- Disaster Recovery
- Capacity bursting
- HeatWave for Analytics, ML, Lakehouse



PostgreSQL DB-Systems

PostgreSQL - one of the most advanced open source databases

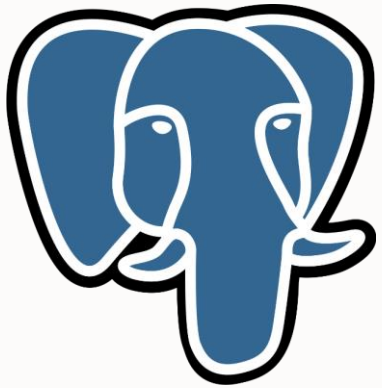
DB Engines ranking:

Rank			DBMS	Database Model	Score		
Nov 2023	Oct 2023	Nov 2022			Nov 2023	Oct 2023	Nov 2022
1.	1.	1.	Oracle	Relational, Multi-model	1277.03	+15.61	+35.34
2.	2.	2.	MySQL	Relational, Multi-model	1115.24	-18.07	-90.30
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	911.42	+14.54	-1.09
4.	4.	4.	PostgreSQL	Relational, Multi-model	636.86	-1.96	+13.70
5.	5.	5.	MongoDB	Document, Multi-model	428.55	-2.87	-49.35
6.	6.	6.	Redis	Key-value, Multi-model	160.02	-2.95	-22.03
7.	7.	7.	Elasticsearch	Search engine, Multi-model	139.62	+2.48	-10.70
8.	8.	8.	IBM Db2	Relational, Multi-model	136.00	+1.13	-13.56
9.	9.	↑ 10.	SQLite	Relational	124.58	-0.56	-10.05
10.	10.	↓ 9.	Microsoft Access	Relational	124.49	+0.18	-10.53

Bringing managed PostgreSQL to OCI

Fully managed and optimized for our customers

PostgreSQL: The world's most advanced open source database.



+

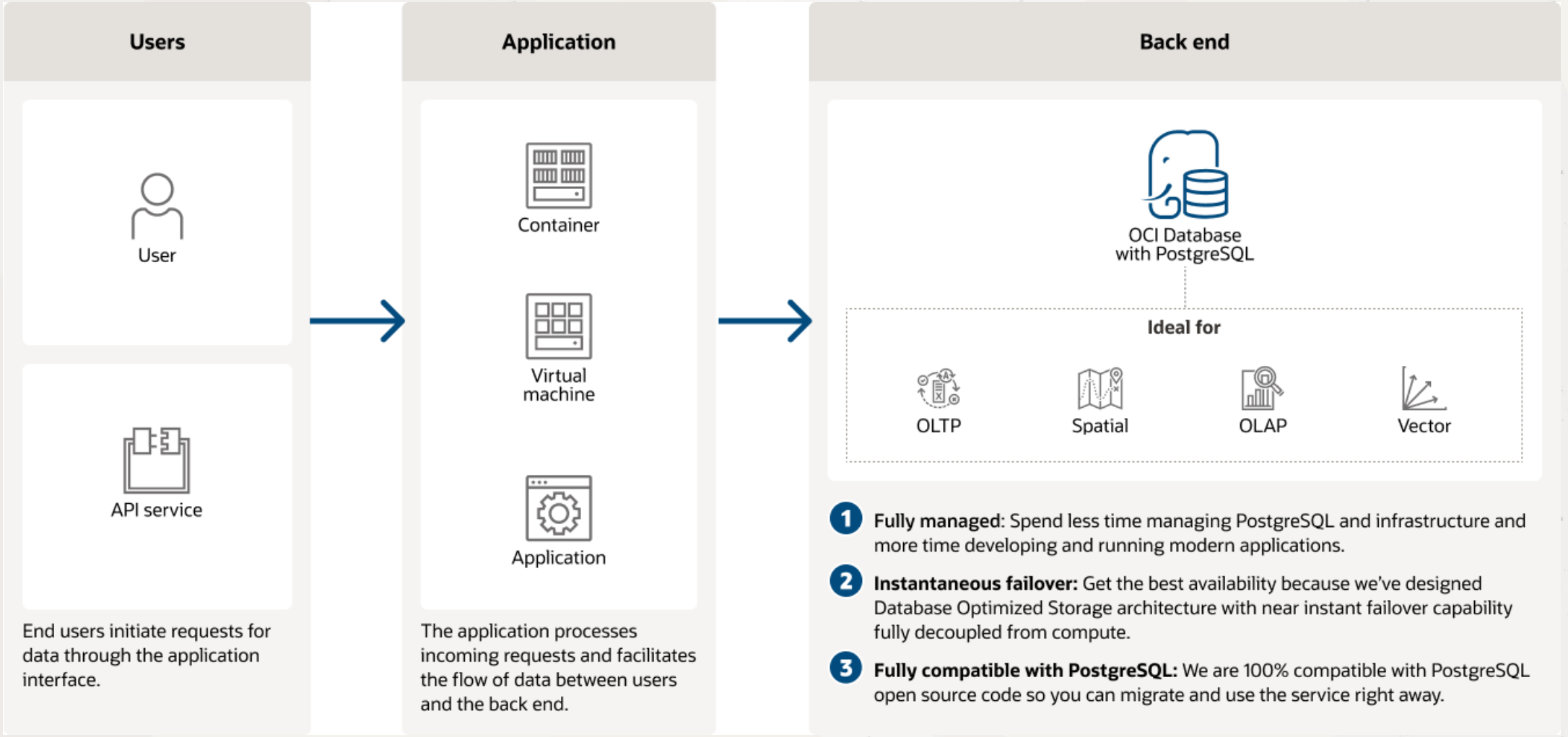
Optimized block storage: Elastic cloud storage that maximizes database availability.

Optimized Storage Features:

- Auto-scales with usage.
- Guaranteed performance
- Distributed across multiple Availability Domains for 0 RPO and fast recovery.
- Storge that grows and shrinks with use. Pay only for what you use.
- Available in single- or multi-node configurations.

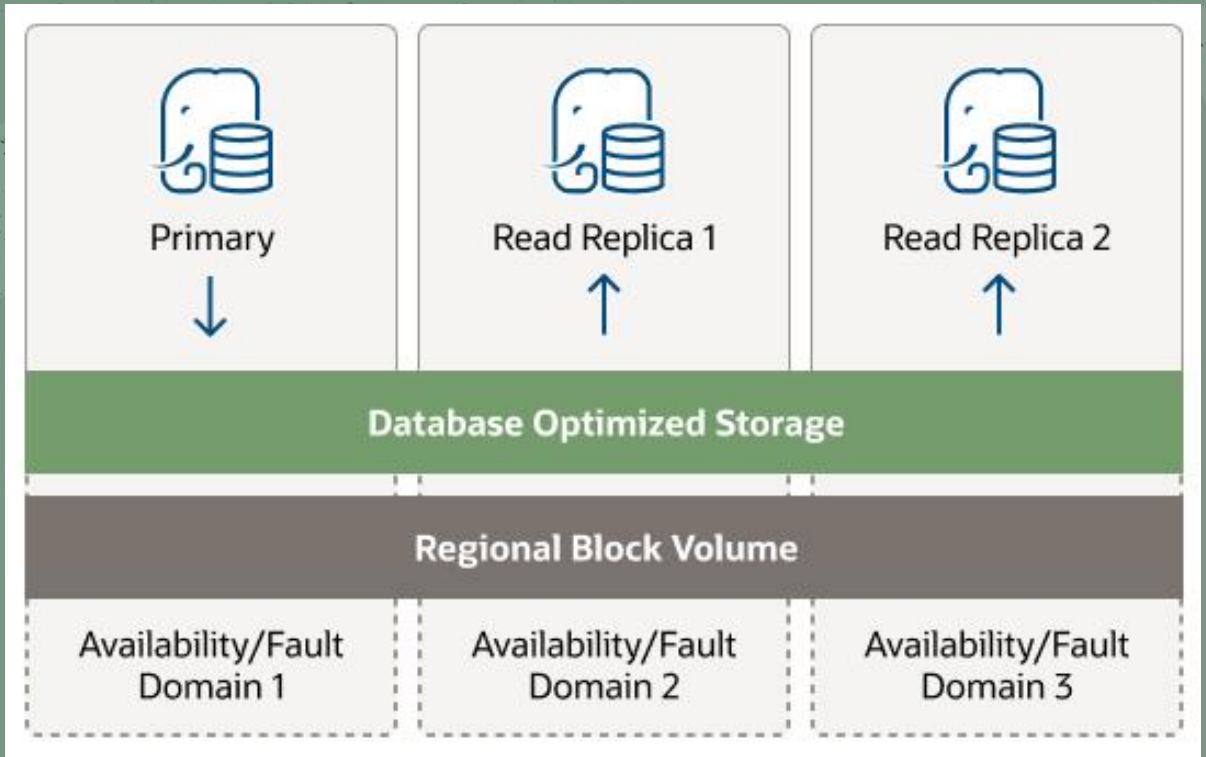
OCI Database with PostgreSQL

A fully managed service



OCI Database Optimized Storage

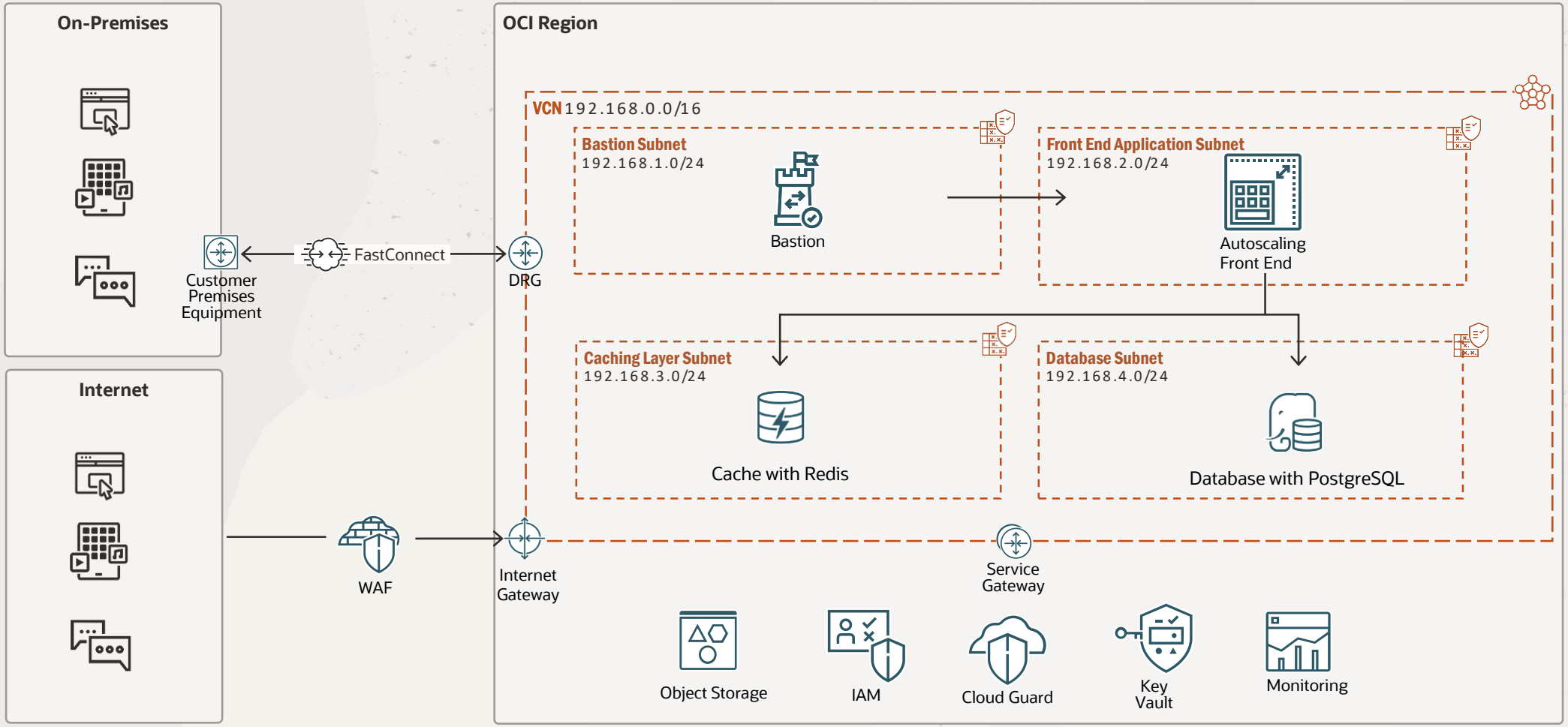
An Oracle Innovation



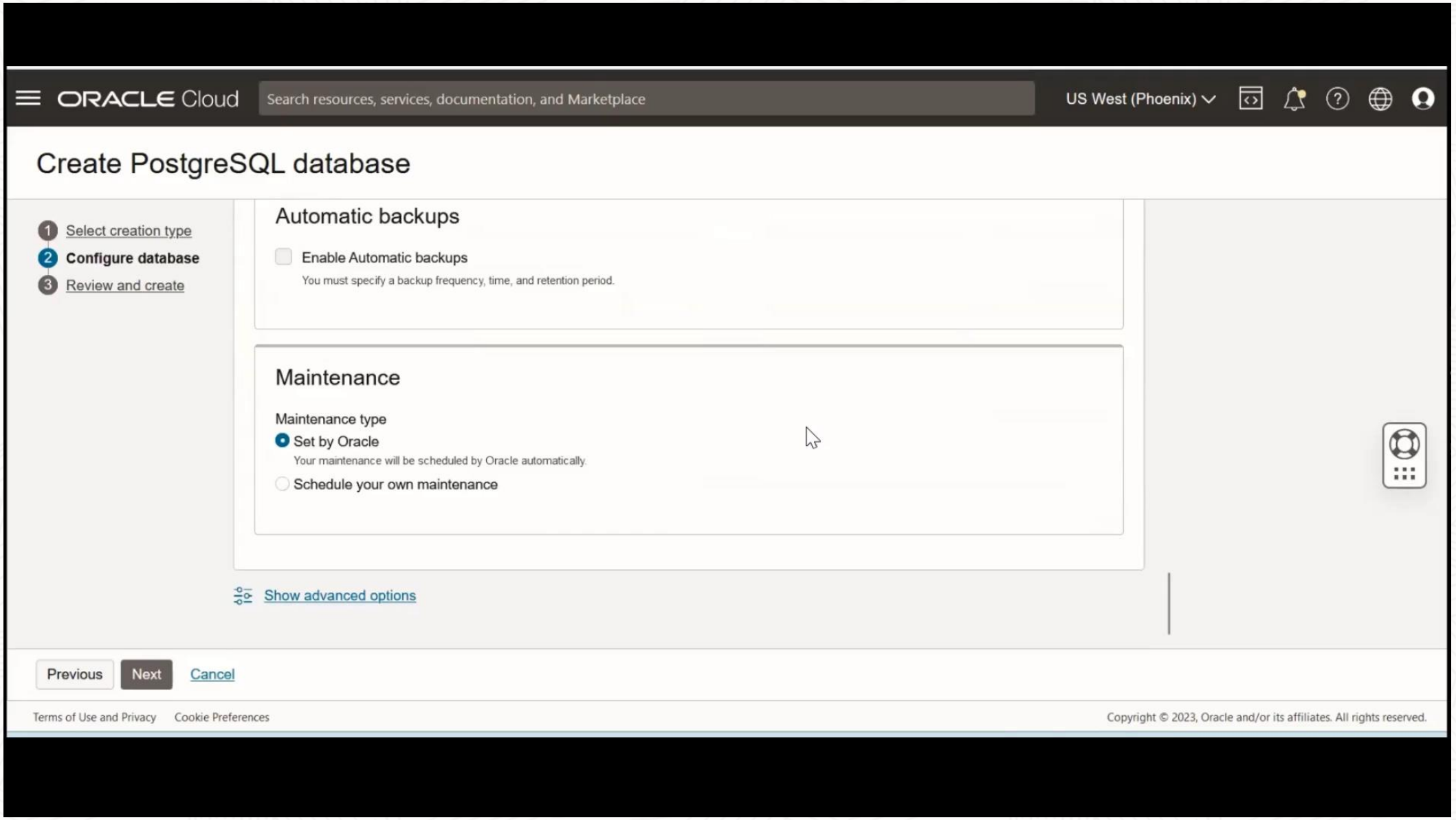
- Dynamic scaling with usage
- Pay only for what you use (scaling is zero downtime)
- Distributed across multiple Availability Domains for 0 RPO and fast recovery

OCI Database with PostgreSQL

Reference Architecture



OCI Database with Postgres



Features available now

Features	Availability
Operations	
Fully managed	✓
Storage Dynamic scaling	✓
Backups - manual and automatic	✓
Managed maintenance and patching	✓
Monitoring and metrics	✓
Notifications and alarms	✓
Terraform support	✓
Automatic minor upgrades	✓
HA / DR	
Multi availability-domain data replication	✓
RTO	< 2 min (multi node) < 20 min (single node)
RPO	0
Automatic/Manual failover	✓
Security	
End-to-end encryption	✓
Automatic security patching	✓
Development	
Version Support - Postgres 14	✓
Stored procedure languages	✓
Extensions – btree_gin; btree_gist; citext; cube; dict_int; fuzzystrmatch; hstore; intarray; isn; lo; ltree; pg_trgm; pgcrypto; plpgsql; pg_stat_statements; seg; tablefunc; tcn; tsm_system_rows; tsm_system_time; unaccent	✓
Limits: 32 TB storage, 1024 cores, 16 TiB RAM	✓

Over 20 Extension Supported and more on their way

See: docs.oracle.com/iaas/Content/postgresql/

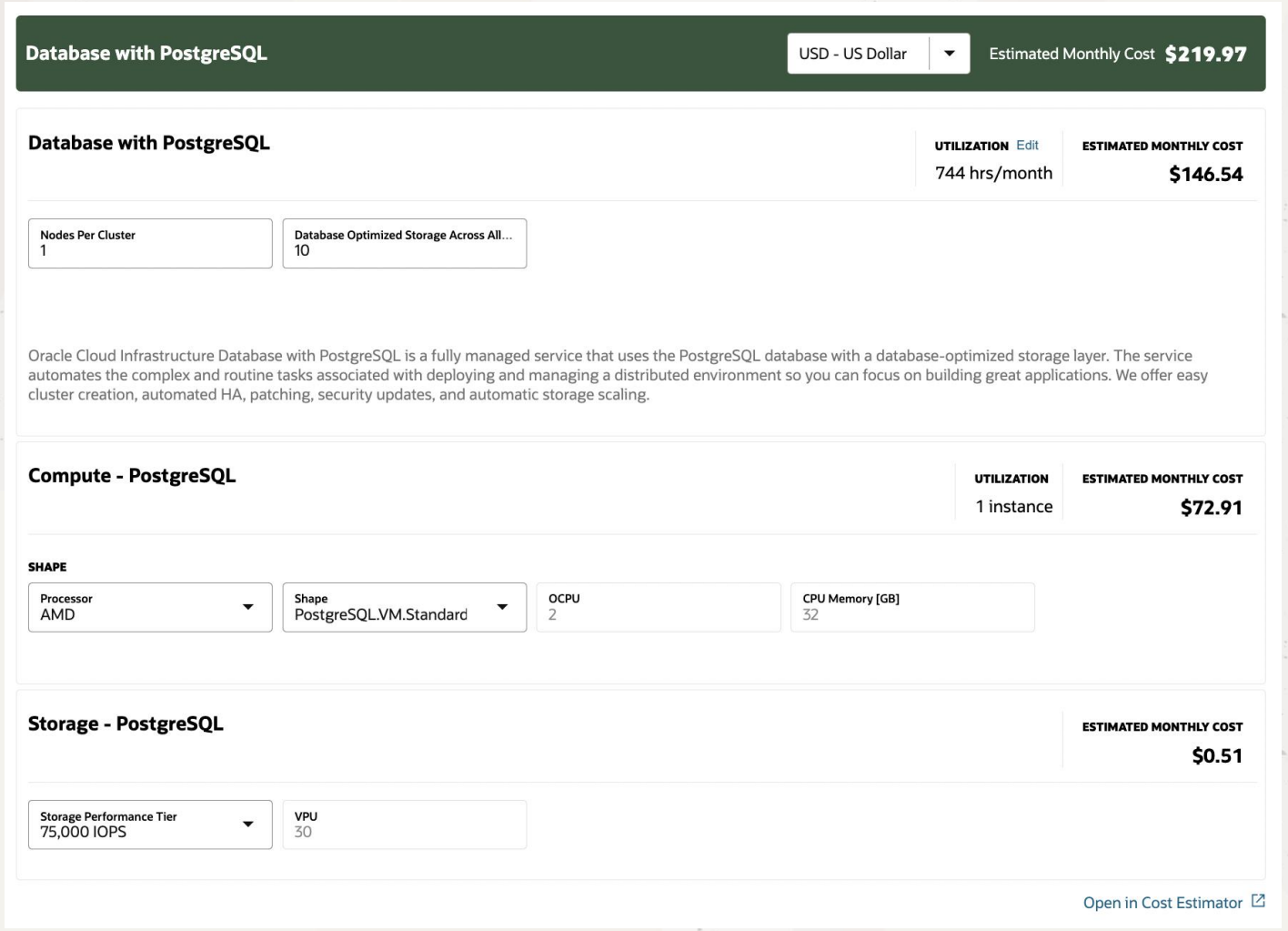
- btree_gin
- btree_gist
- citext
- cube
- dict_int
- fuzzystrmatch
- hstore
- intarray
- isn
- lo
- ltree
- pg_trgm
- pgcrypto
- plpgsql
- pg_stat_statements
- seg
- tablefunc
- tcn
- tsm_system_rows
- tsm_system_time
- unaccent

Features in the roadmap

Category	Features
Versions	v13, v15, v12, v16
Operation	Autonomous configuration
	Custom IP endpoints
HA / DR	Point-in-time recovery
	Cross-region replication
Extensions	PotgresGIS TimescaleDB
	Other extensions: uuid-ossp, dblink, file_fdw_program, pg_buffercache, pg_partman, pg_trgm, pgcrypto, pglogical, Plpgsql, pg_catalog, plpython2u, postgres_fdw, **pgvector**
Security	BYOK
Migration	Manual/Migration service
Dev	Connection Pooling

Among the best price/performance ratio on the market

See: oracle.com/cloud/postgresql/pricing/



OCI Database with PostgreSQL

Fully managed

Unique price/perf ratio

Versatile and Extensible database

Easy-to-Use service

Meets you where you are

NoSQL

Oracle NoSQL Database Cloud Service

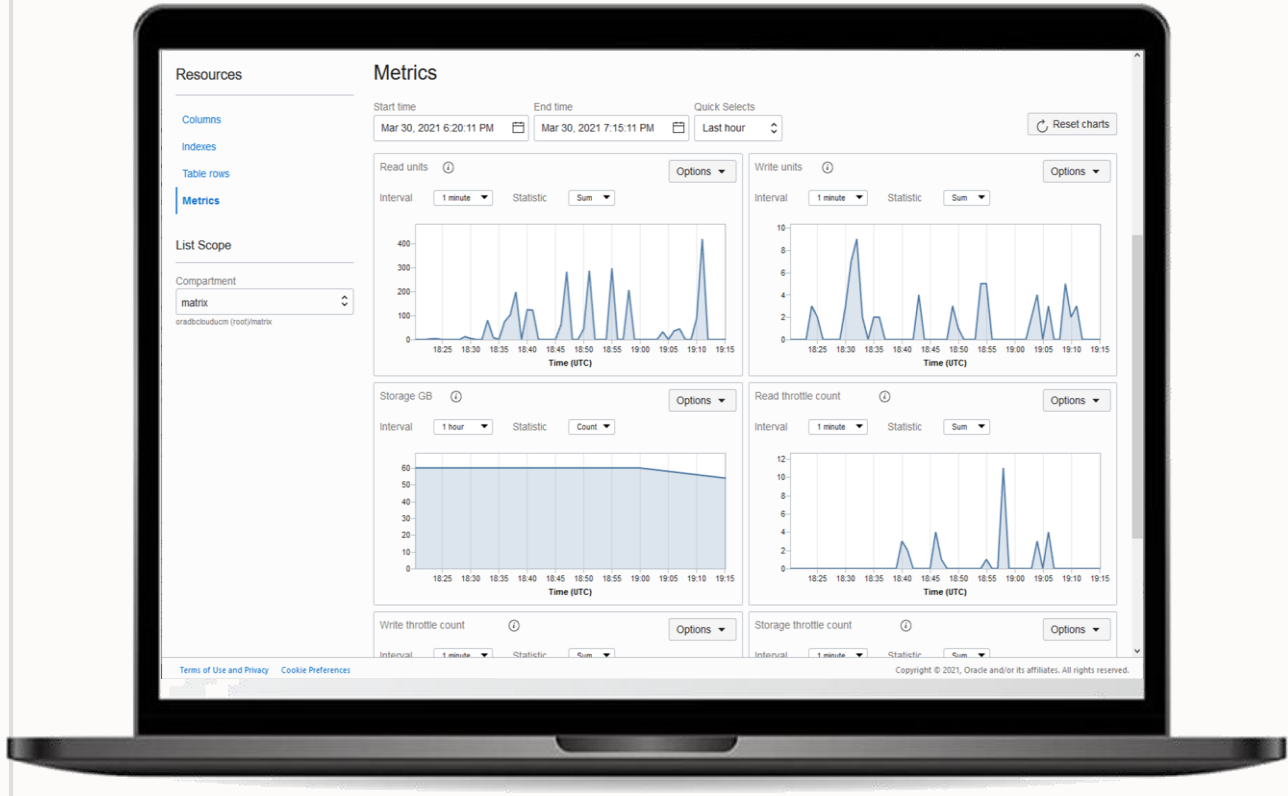
Built for extreme, dynamic workloads of today's modern applications

Fast, Flexible NoSQL Database Service at any scale

- **Fully managed**, **serverless** NoSQL database **table service**
- **Single digit millisecond** and **predictable** latency at any scale
- **Linear throughput scaling** for **extreme** workloads
- Multi-model support (document, fixed schema, key/value)
- **Built-in high availability** for business continuity
- Fully **ACID compliant** and **adjustable** read consistency
- Serverless computing through Oracle Functions
- Available in 30+ OCI commercial regions worldwide (July2024)

Differentiated Use Cases

- **Request level granularity** for extreme workloads, and handles **spikes** and **drops optimally**
- Fast, constant, high-volume workloads requiring **predictable low latency** for **highly responsive applications**
- Designed for business applications requiring **scale**, **performance**, and **high availability** with flexible consistency

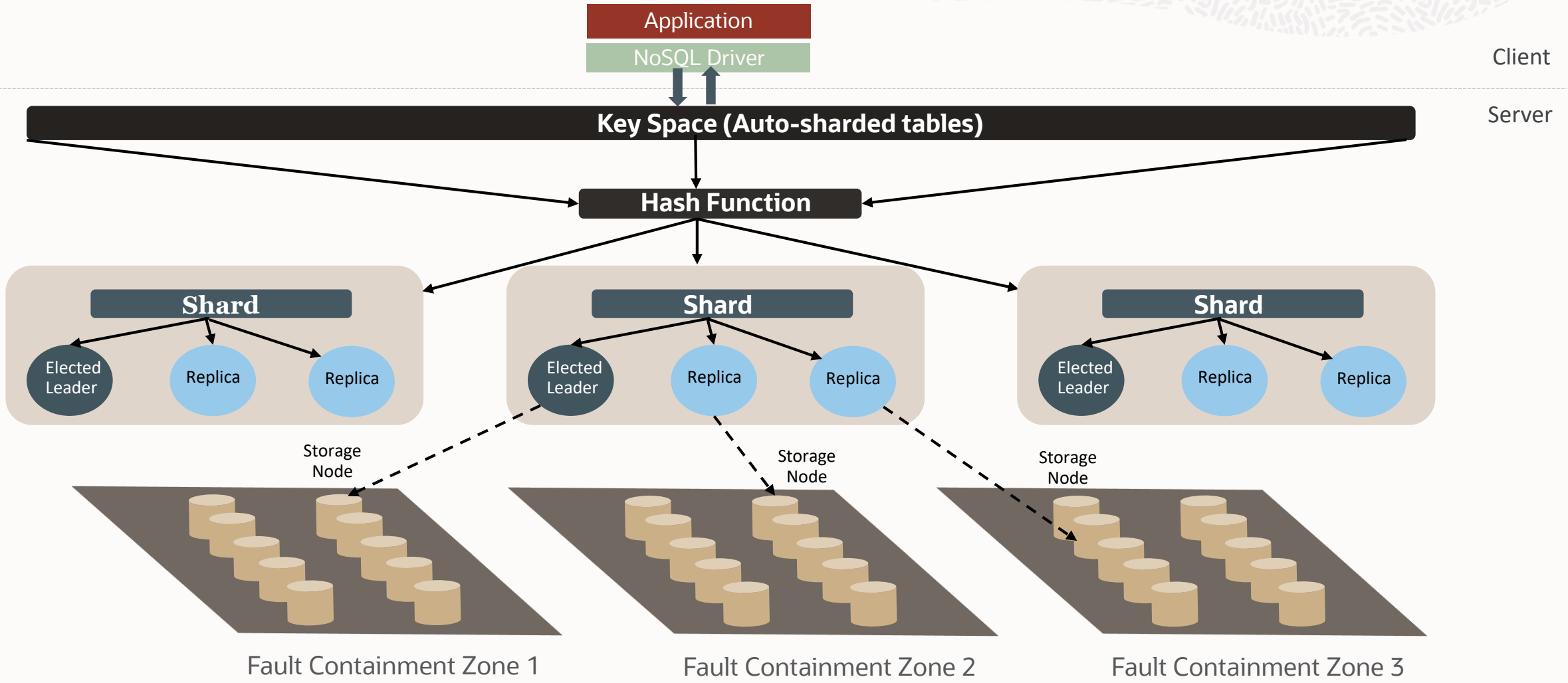


NoSQL Database –Use Cases

Mobile applications	**Internet of things**	**360 degree customer view**	**User profile management**	**Catalog data**
				
Content management	**Online advertising**	**Real time Big data**	**Social network**	**Gaming**
				

Oracle NoSQL Database Architecture Overview

A distributed, shared nothing key/value data store architected for HA



Oracle NoSQL Database Cloud Service - Metrics

Throughput provisioning

1 Write Unit

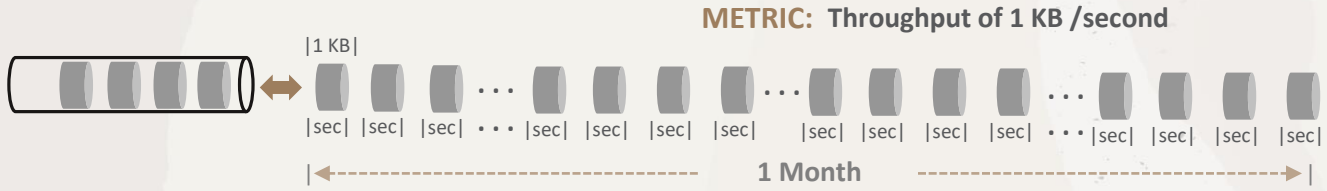
- The throughput of up to 1 kilobyte (KB) of data per second for a write operation over a one-month period
- Approximately 2.67 million writes per month

1 Read Unit

The throughput of up to 1 kilobyte (KB) of data per second for an eventually consistent read operation

Approximately 2.67 million eventually consistent reads per month

2 Read units are needed for an absolute consistent read



Period of a month
=3600 KB/Hr*744 Hr
=2.67 million (writes/reads) KBs

Oracle NoSQL Database Cloud Service – Capacity

Provisioned capacity vs. on-demand capacity

Provisioned Capacity

- Must determine read/write units in advance
- Adjustments done via API or console
- Increasing unlimited
- Decreases limited to 4 per day
- Pay for what you provision
- Deep understanding of workload needed

On-Demand Capacity

Automatic scaling

No rate limiting in your application

No workload characterization

Simple to use

Pay for what consumed

Oracle NoSQL Database Cloud Service – Provisioned Capacity

Provisioned throughput

Provision reads/sec, writes/sec, GB storage at table creation time

- Dynamically increase
- Dynamically decrease


```
JAVA code sample:
TableRequest tableRequest = new TableRequest()
        .setStatement("create table if not exists foo (id integer,
value JSON)")
        .setTableLimits(new TableLimits(2000, 100, 500))
        .setTimeout(1000);
TableResult res = NoSQLHandle.tableRequest(tableRequest);
```

Modify the table lowering the read units to 1000

```
tableRequest.setTableLimits(new TableLimits(1000, 100, 500));
```

Note: Every TableRequest is a DDL call to the NoSQL store and may be performed 4 times within a minute.

Oracle NoSQL Database Cloud Service – On-Demand Capacity

Auto-scaling throughput

Python Code Sample

```
request=TableRequest().set_statement(statement).set_table_limits(
       TableLimits(0,0,500,TableLimits.CAPACITY_MODE.ON_DEMAND))

handle.do_table_request(request, 50000, 3000)
```

Data Models

Key-Value, Schemaless JSON, Fixed Schema

```
create table if not exists myTable(id long generated always as identity,
                                   resNum string,
                                   value JSON,
                                   primary key((shard(id)))'
```

```
{
  "lastName" : "Jones",
  "firstName" : "Philly-Joe",
  "bagInfo" : [ {
      "tagNum" : "17657806285185",
      "lastSeenAt" : "LHR",
      "flightLegs" : [ {
          "flightNo" : "BM254",
          "flightDate" : "2019-02-28T22:00:00",
          "fltRouteSrc" : "SYD",
          "fltRouteDest" : "LHR"
      } ] }
]
```

Key-Value

```
record = {'resNum' : 'JS8PKQ',
          'value' : {
              'lastName' : 'Jones',
              'firstName' : 'Philly-Joe',
              'bagInfo' : […]}
         }

req = PutRequest().set_table_name('myTable').
          set_value(record)
handle.put(request)
```

Schema-less JSON

```
statement = ( 'select
                m.value.bagInfo.lastSeenAt
              from
                myTable m
              where
                m.bagInfo.tagNum = '17657806285185' )

request = QueryRequest().set_statement(statement)
result = handle.query(request)
```

Fixed Schema

```
statement = ( 'select
                m.value
              from
                myTable m
              where
                m.resNum = 'JS8PKQ' )

request = QueryRequest().set_statement(statement)
result = handle.query(request)
```

Data Models & SQL Interoperability

Seamless SQL interoperability with schemaless and fixed schema

Schemaless

Data Definition

```
create table profile(cookieID STRING,
        content JSON, primary key(cookieID))
```

Question

Find all visitors to my site in November who are males between 24 and 30 years of age

Query

```
select cookieID from profile
where
   cast(content.lastVisit as TIMESTAMP) >=
        cast("2019-11-01" as TIMESTAMP) and
   cast(content.lastVisit as TIMESTAMP) <=
         cast("2019-11-30" as TIMESTAMP) and
   content.demographic.gender = 'M' and
   content.demographic.age >= 24 and
   content.demographic.age <= 30
```

Identical SQL

Fixed Schema

Data Definition

```
create table profile(cookieID STRING,
     content RECORD(lastVisit TIMESTAMP,
               demographic RECORD(age INTEGER,
                          gender ENUM(M, F))),
     primary key(cookieID))
```

Question

Find all visitors to my site in November who are males between 24 and 30 years of age

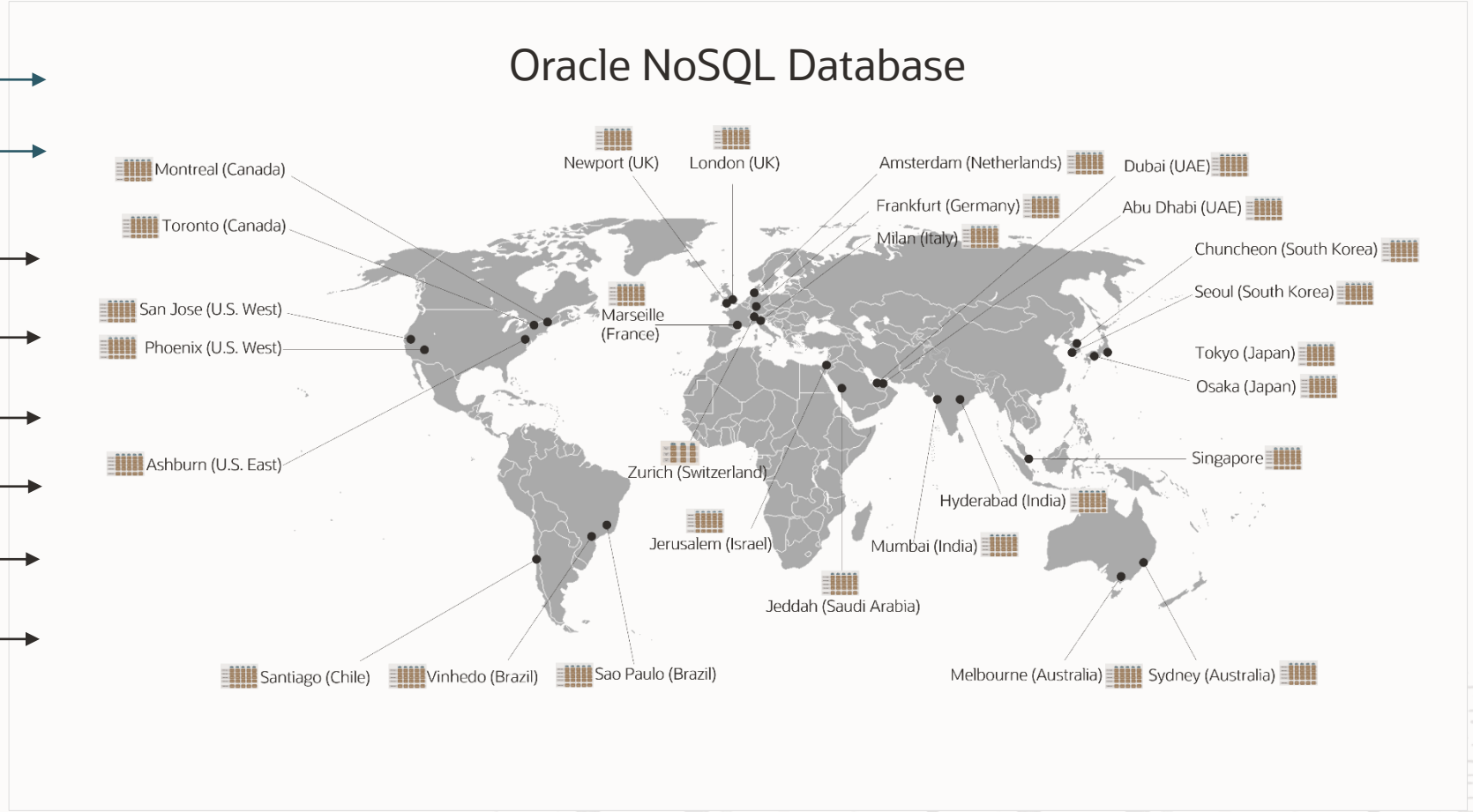
Query

```
select cookieID from profile
where
   cast(content.lastVisit as TIMESTAMP) >=
         cast("2019-11-01" as TIMESTAMP) and
   cast(content.lastVisit as TIMESTAMP) <=
          cast("2019-11-30" as TIMESTAMP) and
   content.demographic.gender = 'M' and
   content.demographic.age >= 24 and
   content.demographic.age <= 30
```

Oracle NoSQL Database - SDKs

Different Programing Interfaces for Your Applications

- Terraform
- OCI CLI
- JAVA
- PYTHON
- NODE.JS
- GO
- C#
- SPRING



OCI Search with OpenSearch

Use Case: Leveraging open source databases

OCI Search with OpenSearch

Fully Managed

Spend less time on maintenance tasks

- Managed maintenance activities include patching, updating, backups, and no-downtime resizing

Optimized price/ performance ratio

Best price-performance guarantee

- With OCI flexible shapes you can build any combination of CPU and RAM to optimize your performance
- Competitive pricing with a very low management fee.

Simplified migration

Migrating from an existing OpenSearch deployment

- Whether self-managed or hosted, migration is a straightforward process.
- OCI experts support you

OCI Search with OpenSearch

OpenSearch 2.8 enables Neural Search

- Ingest, search, visualize and analyze data in near real-time.
- OpenSearch 2.8 Release:
  i. Includes plugins: ML Commons, k-NN and Neural Search
  ii. Updated security with LDAP, SAML, OpenID support
  iii. Improved search with cross cluster search and searchable snapshots
  iv. Improved alerting and notifications

Powerful Insights

- Enhanced neural search capabilities
- Visualize data with OpenSearch Dashboards
- Access dozens of visualization tools out of the box

Fully Managed

- Automated cluster creation
- Patching provides latest security updates
- Use automated backups or Snapshot API to Object Storage

Automated HA

- Deploy 2 or more nodes and OCI automatically utilizes multiple Availability or Fault Domains

Unparalleled Simplicity

- Choose exact amount of CPU, memory and storage required by your specific use case

Extensible

- Scale nodes vertically or horizontally to right size your specific workloads

ORACLE
Cloud Infrastructure
+
OpenSearch

OCI Cache with Redis

Delivering ultra-fast data

- Customers use Redis to deliver ~1 millisecond, low latency data to their end users.
- Caching key values in memory:
  i. increases application responsiveness and reduces latency for an exceptional user experience
  ii. reduces the network load on databases calls, saving on network traffic
  iii. stores read data in memory for quick access

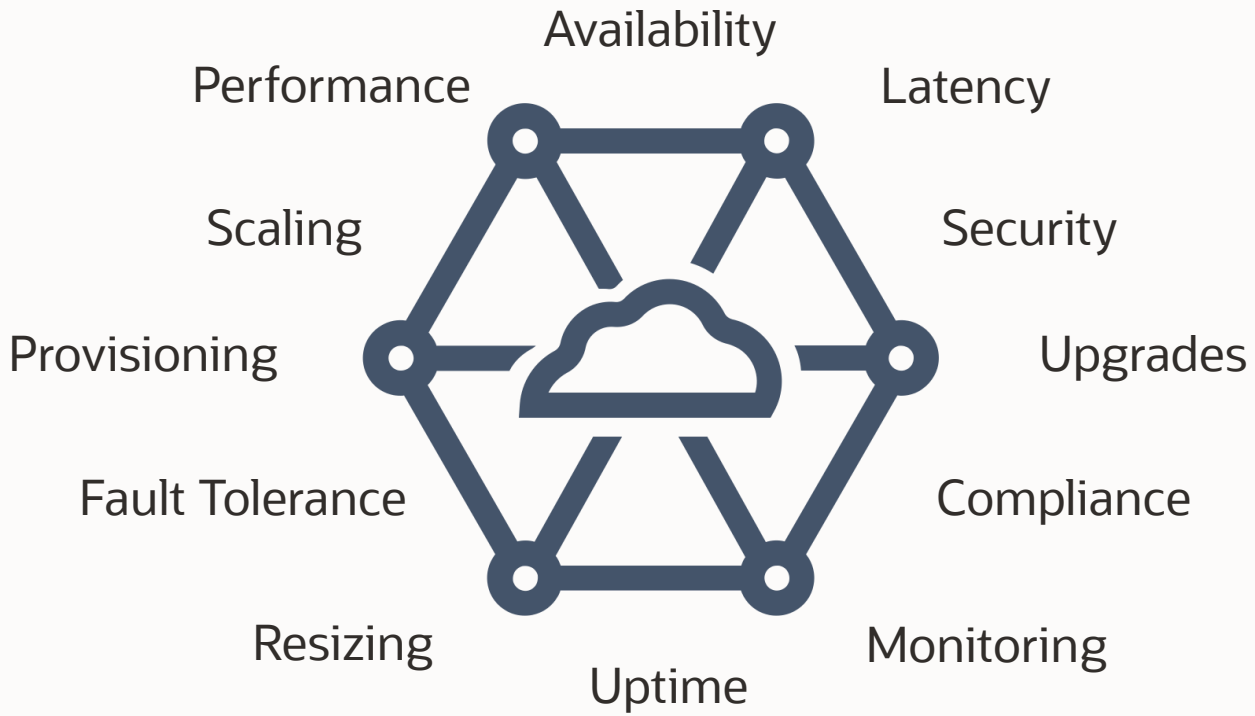
Gaming

Ecommerce

Social Media

IoT

Managing a distributed cache is complex…



…but doesn't have to be

Introducing OCI Caching Service with Redis

OCI Caching Service with Redis is a fully managed distributed caching service that uses open-source **Redis 7.0.5.**

The service automates the complex and routine tasks associated with deploying and managing a distributed environment so you can focus on building great applications.

We offer easy cluster creation, automated HA, patching, security updates, and resizing.

Key features

Fully Managed

- Automated cluster creation
- OS-level patching
- Monitoring
- Failover
- SLA 99.95%

Unparalleled Simplicity

- Specify the amount of memory and number of nodes and we take care of everything else
- Choose exact amount of memory for your workload. No need to overprovision

Scalable

- Scale your cache on the fly with no downtime
- Scale from 2GB up to 500GB per node and 5 nodes per cluster to meet your caching needs

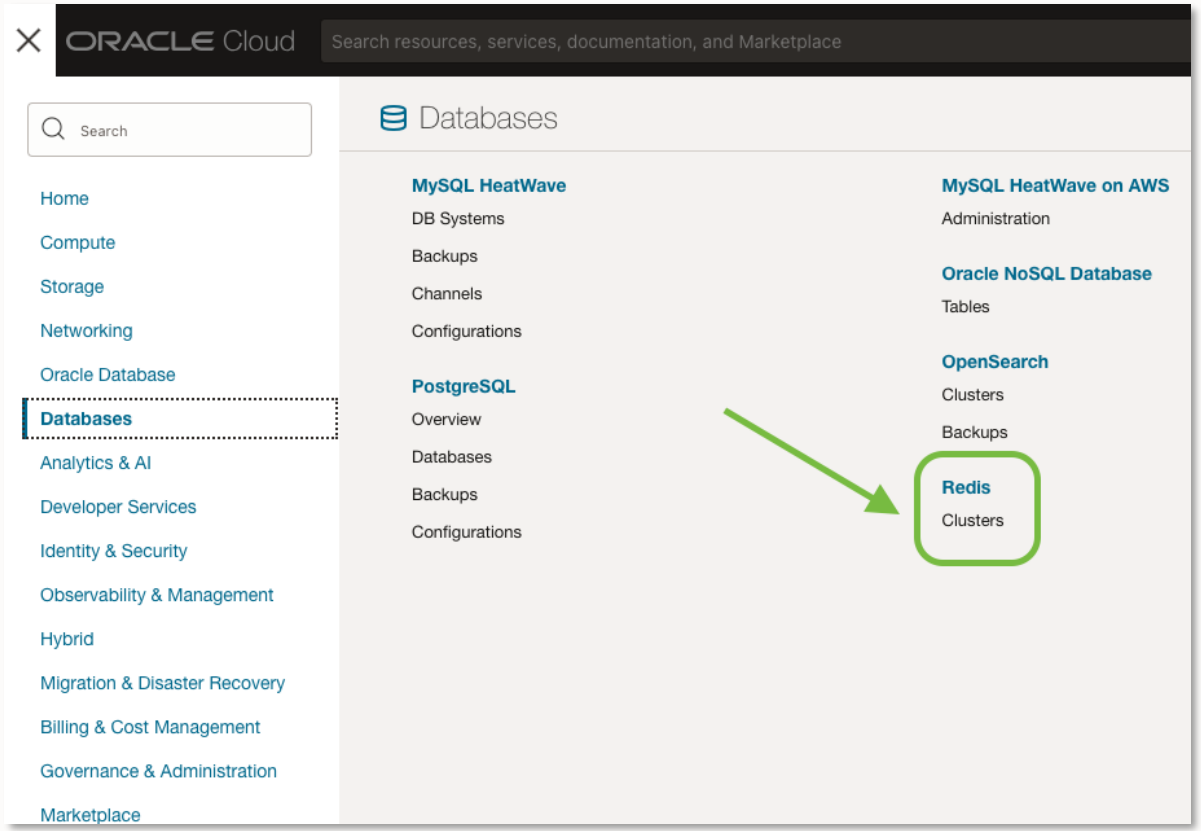
Powerful Insights

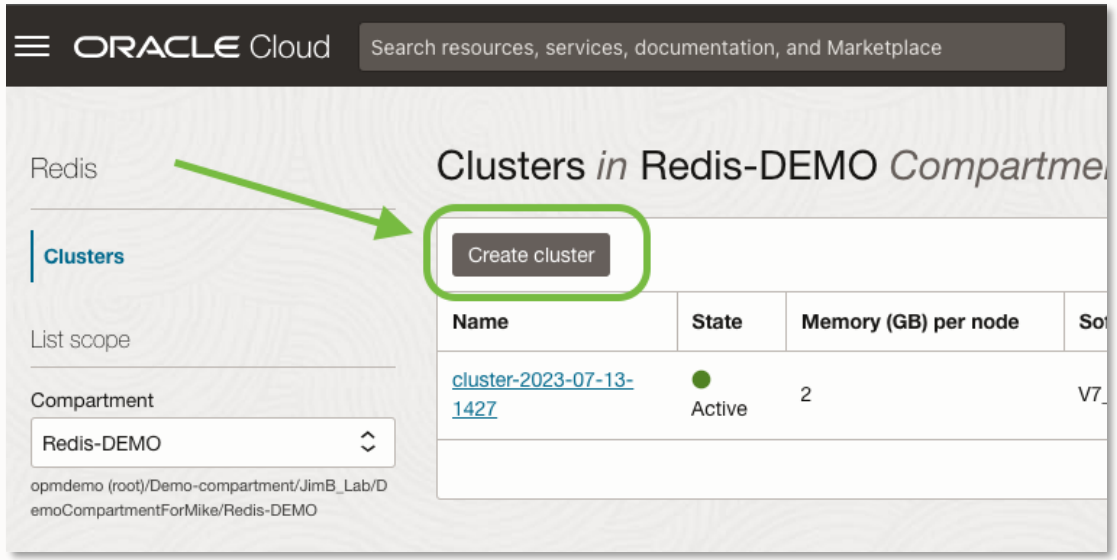
- Out of box metrics to monitor cluster health and performance
- Access CPU and Memory utilization
- See transmitted and received network bytes

Automated HA

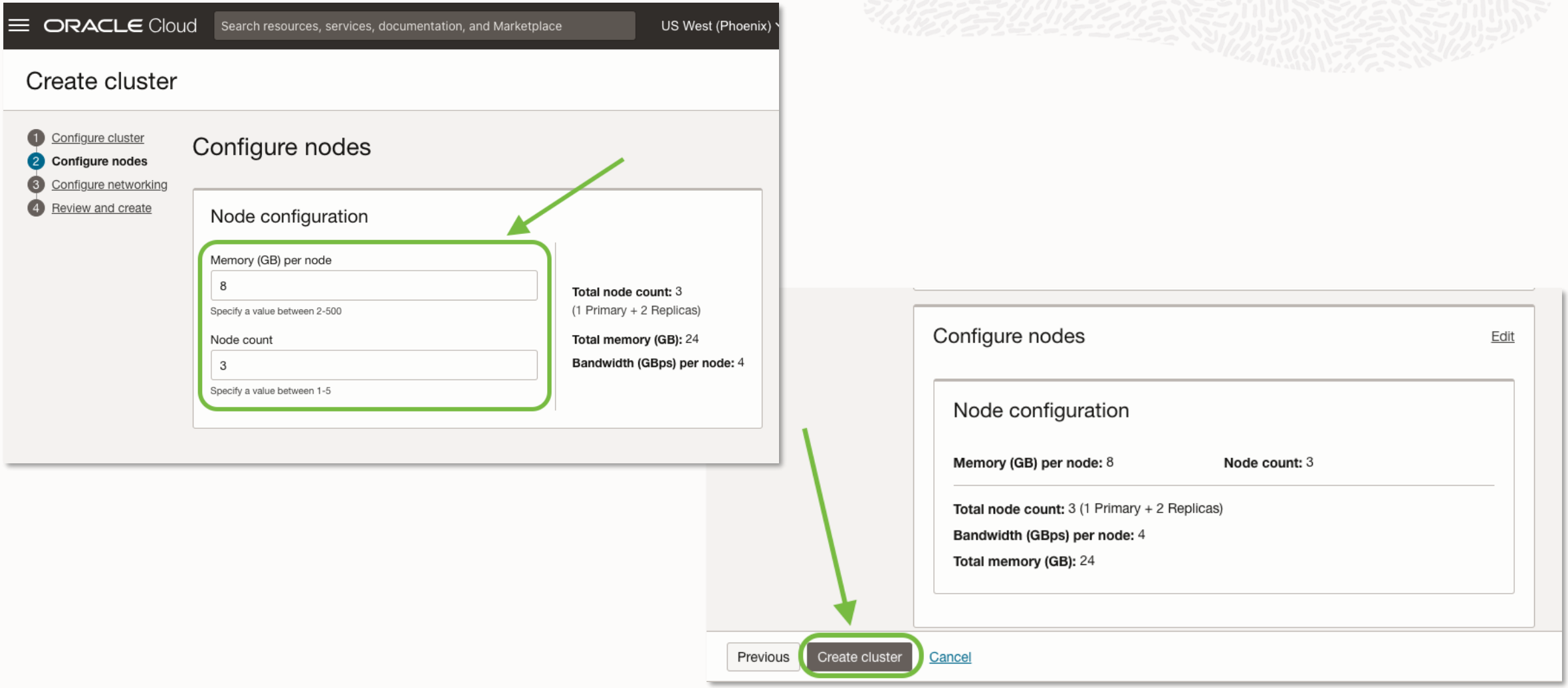
- Deploy 2 or more nodes and OCI automatically spreads them across multiple Availability or Fault Domains

User Experience – Create a cluster

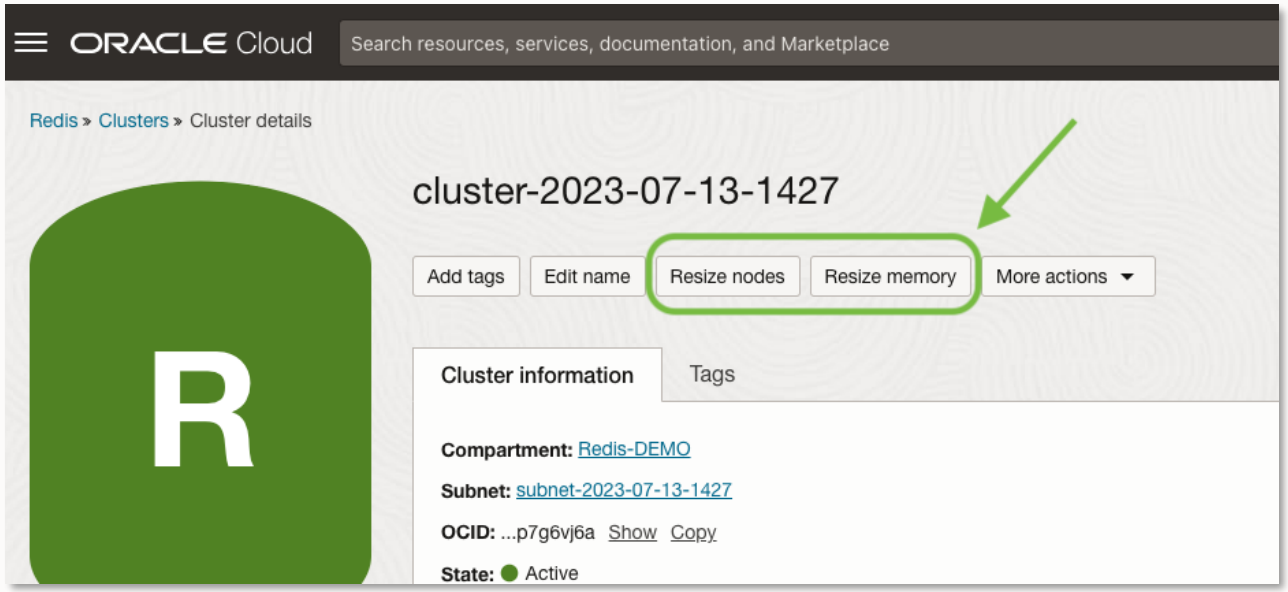


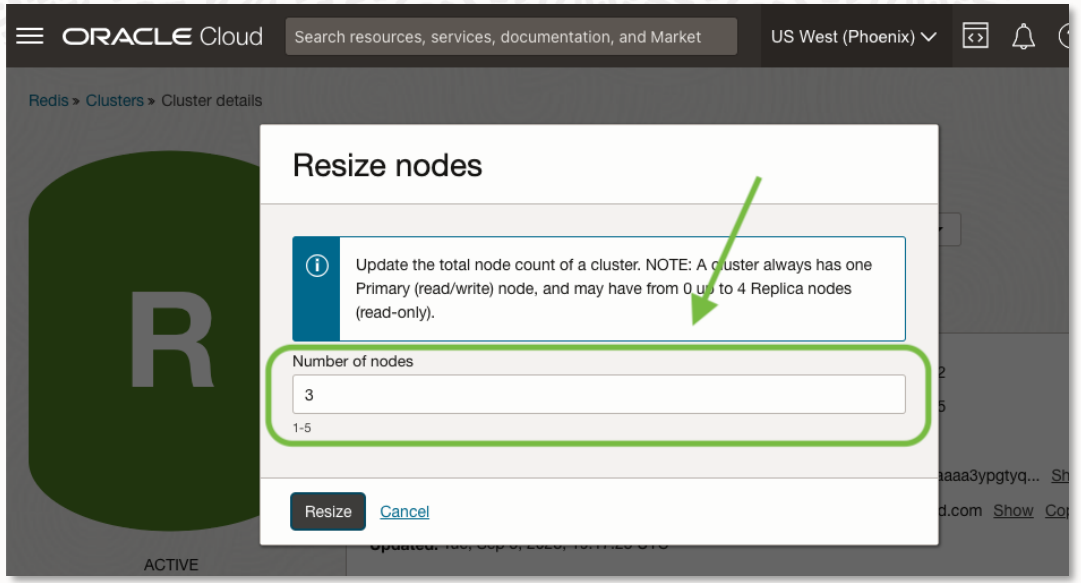


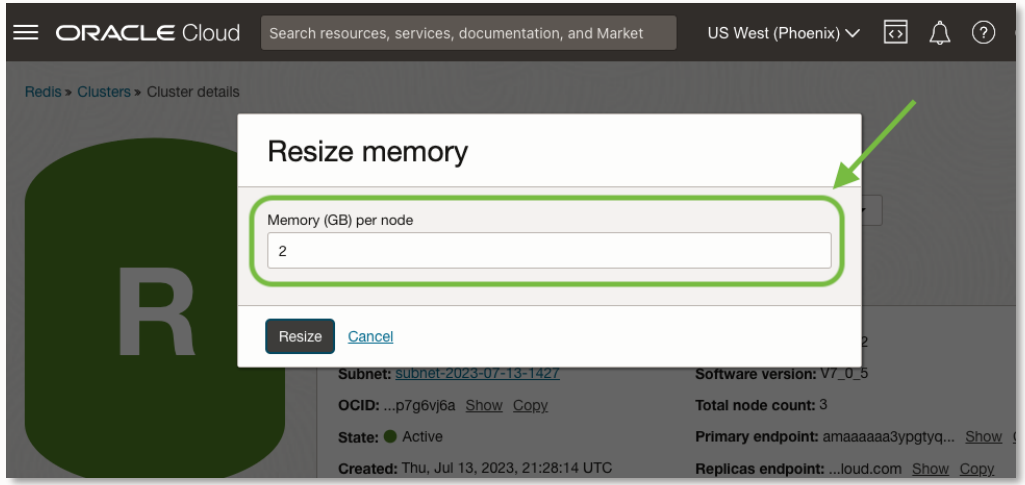
User Experience – Create a cluster



User Experience – Resize nodes and memory







GA Features

Feature	Description
Multi-node cluster deployment	Customer can deploy a cluster with a single primary node and multiple read replica nodes (CRUD operations).
Multi-AD HA cluster deployment	Replica nodes in additional ADs with partition tolerance.
Automatic failure detection	When one or more nodes in a cluster goes down, new nodes automatically join the cluster to compensate.
Automatic failover during an AD outage	For multi-AD scenarios, when an AD is lost from the cluster, a read replica is promoted to the new primary and the DNS is updated to point to the new primary.
Encryption: in-transit	Support for TLS encryption of communication between nodes, as well as encryption for client communication.
Resize: reduce cluster size	Remove nodes on demand.
Resize: larger cluster capacity	Add nodes to increase cluster total in-memory storage capacity.
Performance: one-millisecond read/write operations	One-millisecond read/write operations and with optimal configurations
Patching: automatic host OS patching	Hosts are patched without customer action.
Interface with cluster via Redis API, console, SDK, and Terraform	Baseline parity with Redis API. Service run the latest stable version of Redis and supports standard Redis API protocols. Also, access Redis through the console, SDK, and Terraform.

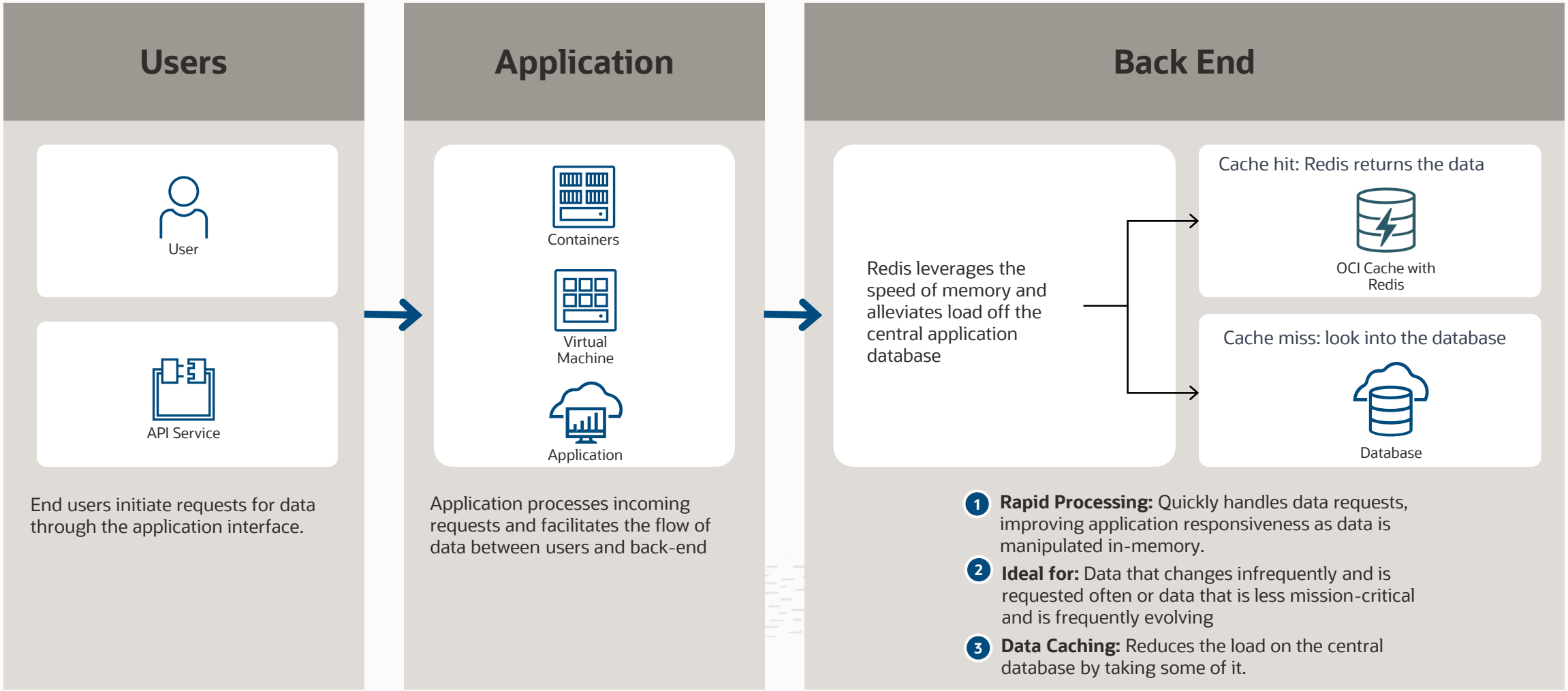
GA Features

Feature	Description
Instance health metrics	Integrated with OCI Monitoring service.
Compliance Baseline	Service complies with all OCI baseline Compliance programs: FIPS, PCI, HIPAA, GDPR, SOC, and ISO.

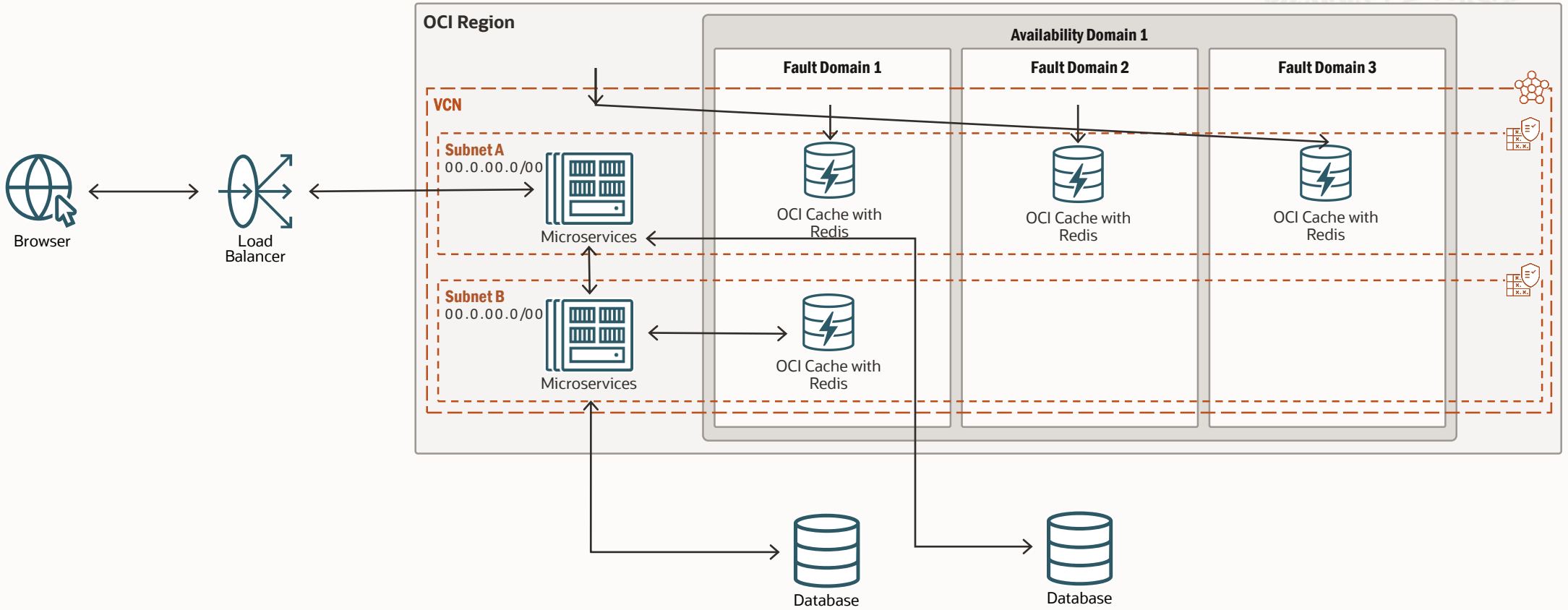
Competitive Table

Features	OCI Redis	AWS ElastiCache	Azure Cache for Redis
Fully Managed	Yes	Yes	Yes
Flexible Memory Shapes	Yes	No	No
Automatic Failover	Yes	Yes	Yes
Manual Failover	No	Yes	Yes
Read Replicas	Yes	Yes	Yes
Sharding	No	Yes	Yes
Metrics	Yes	Yes	Yes
Notifications and Alarms	Yes	Yes	Yes
Backups-Manual and Automatic	No	Yes	Yes
Version Support	Yes, Redis 7.0.5 only	Yes, multiple Redis versions 2.6.x -7x	Yes, Redis 4 and 6
Version Upgrade	No	Yes	Yes
Maintenance window schedule	No	Yes	Yes
Cross Region Replication	No	Yes	Yes
Autoscaling	No	Yes	No
Compliance programs	Yes	Yes	Yes
Lua Scripting Support	No	Yes	No

How it works



Reference Architecture



Use Case: Boosting MySQL HeatWave with OCI Cache with Redis

Users

End users initiate requests for data through the application interface.

Application

Application processes incoming requests and facilitates the flow of data between users and back-end

Back End

Redis leverages the speed of memory and alleviates load off the central application database

Cache hit: Redis returns the data

Cache miss: look into the database

1. **Rapid Processing:** Quickly handles data requests, improving application responsiveness as data is manipulated in-memory.
2. **Ideal for:** Data that changes infrequently and is requested often or data that is less mission-critical and is frequently evolving
3. **Data Caching:** Reduces the load on the central database by taking some of it.

Takeways

OCI's data platform has the lowest TCO

			Oracle (OCI)	Amazon (AWS)	Google (GCP)
BIG DATA	**Spark-based Data Engineering**	(100 TB/Month)	**$204**	3.5X	2.5X
	No-Code ETL	(TB/Month)	**$236**	5.5X	4.75X
	Hadoop Cluster	(96vCPU, 336 GB RAM, 6 TB, Monthly)	**$2089**	2X	2X
DATA MANAGEMENT	**PostgreSQL Database**	(16 vCPU, 128 GB RAM, 500 GB, Monthly)	**$909**	+77%	+71%
	MySQL Database	(16 vCPU, 64 GB RAM, 500 GB, Monthly)	**$345**	3x	3x
	OpenSearch	(32 vCPU, 128 GB RAM, 5 TB, Monthly)	**$796**	3x	5x
	Redis Cache	(32 GB, Monthly)	**$355**	+60%	+127%

Green = Lowest cost
Based on published pricing as of August, 2024

Thank You!

Alexandre Fagundes
alexandre.af.fagundes@oracle.com
Cloud Architect, Oracle Latin America


ORACLE