ORACLE

Introducing MySQL HeatWave Migration Assistant

Guided, automated approach to cloud migration

Alexander Hitrov

Senior Principal Software Developer

MySQL & HeatWave Development

January 2026

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Alexander Hitrov

Senior Principal Software Developer

Key Topics We'll Explore

Key Benefits

Foundation

User Interface

Migration Type

Compatibility Checks

Key Benefits

Migrating Critical Databases to the Cloud is often viewed as complex

Why MySQL HeatWave Migration Assistant makes it easy

- **Step-by-step workflow**: Follow an intuitive, graphical wizard—no manual scripts or deep cloud expertise required
- **End-to-end automation**: The assistant can set up cloud accounts, compartments, networking, and secure connectivity, or let you select from existing resources for flexibility
- **Optimized for speed**: Parallel data transfer and import accelerate migrations—even for large workloads
- **Complete migration**: Handles schema, data, user accounts, and replication setups, with guided compatibility checks and recommendations
- **Minimal downtime**: Supports “hot” migrations with live replication, keeping databases in sync and enabling a smooth switch-over when you’re ready

How the Migration Works

1. **Prepare and Authenticate**: Log in to OCI (or create an account) and configure cloud resources with guided help
2. **Compatibility and Planning**: Built-in checks identify and address any issues in advance, while the Assistant prepares a tailored migration plan
3. **Parallel Data Movement**: Accelerated data transfer uses multiple streams for efficiency—ideal for both small and multi-terabyte datasets
4. **Live Replication for Hot Cutover**: For near-zero downtime, replication keeps your on-prem and OCI databases synchronized until the final cutover
5. **Ready for Use**: Once cutover is complete, your MySQL HeatWave instance is ready for OLTP, analytics, AI workloads, and more—with all permissions and schema intact

Foundation

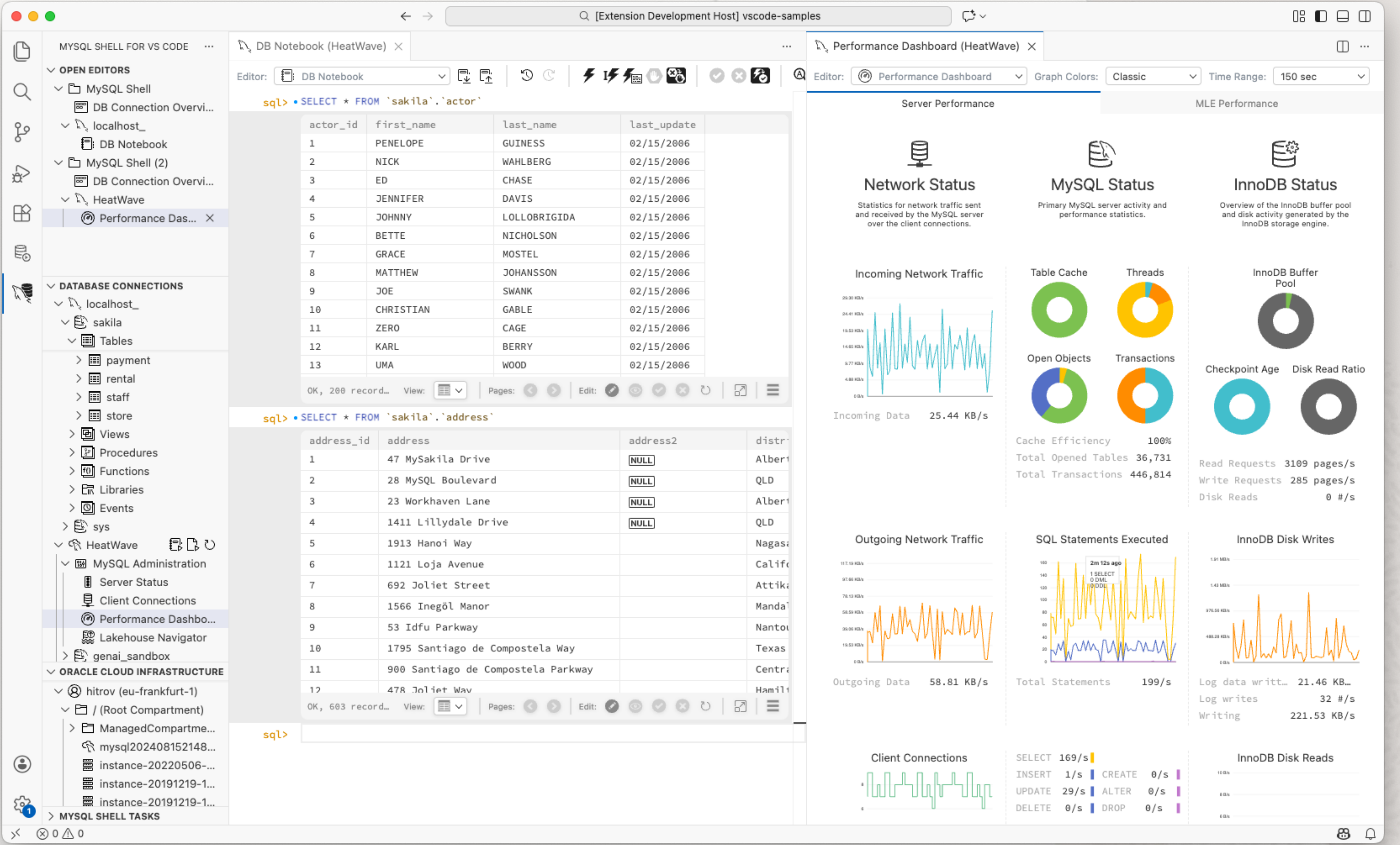
What is MySQL Shell?

Advanced command-line client and scripting environment for MySQL

Unlike the classic mysql client, it provides:

- **Multi-language support**: JavaScript, Python, and SQL modes
- **AdminAPI**: InnoDB Cluster, InnoDB ReplicaSet, and InnoDB ClusterSet management
- **Dump/Load utilities**: High-performance parallel dump and load with *util.dumpInstance()* and *util.loadDump()*
- **Upgrade Checker**: Pre-flight checks before major version upgrades via *util.checkForServerUpgrade()*
- **Plugin system**: Extensible architecture for custom functionality
- **GUI mode**: Web-based interface for visual database management

MySQL Shell for VS Code Extension



[Extension Development Host] vscode-samples

MYSQL SHELL FOR VS CODE

OPEN EDITORS
- MySQL Shell
  - DB Connection Overvi...
  - localhost_
    - DB Notebook
- MySQL Shell (2)
  - DB Connection Overvi...
  - HeatWave
    - Performance Das...

DATABASE CONNECTIONS
- localhost_
  - sakila
    - Tables
      - payment
      - rental
      - staff
      - store
    - Views
    - Procedures
    - Functions
    - Libraries
    - Events
  - sys
- HeatWave
  - MySQL Administration
    - Server Status
    - Client Connections
    - Performance Dashbo...
    - Lakehouse Navigator
  - genai_sandbox

ORACLE CLOUD INFRASTRUCTURE
- hitrov (eu-frankfurt-1)
  - / (Root Compartment)
    - ManagedCompartme...
    - mysql20240815214&...
    - instance-20220506-...
    - instance-20191219-1...
    - instance-20191219-1...

MYSQL SHELL TASKS

DB Notebook (HeatWave)

Editor: DB Notebook

```
sql> SELECT * FROM `sakila`.`actor`
```

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	02/15/2006
2	NICK	WAHLBERG	02/15/2006
3	ED	CHASE	02/15/2006
4	JENNIFER	DAVIS	02/15/2006
5	JOHNNY	LOLLOBRIGIDA	02/15/2006
6	BETTE	NICHOLSON	02/15/2006
7	GRACE	MOSTEL	02/15/2006
8	MATTHEW	JOHANSSON	02/15/2006
9	JOE	SWANK	02/15/2006
10	CHRISTIAN	GABLE	02/15/2006
11	ZERO	CAGE	02/15/2006
12	KARL	BERRY	02/15/2006
13	UMA	WOOD	02/15/2006

OK, 200 record… View: Pages: Edit:

```
sql> SELECT * FROM `sakila`.`address`
```

address_id	address	address2	distr
1	47 MySakila Drive	NULL	Albert
2	28 MySQL Boulevard	NULL	QLD
3	23 Workhaven Lane	NULL	Albert
4	1411 Lillydale Drive	NULL	QLD
5	1913 Hanoi Way		Nagasa
6	1121 Loja Avenue		Califo
7	692 Joliet Street		Attika
8	1566 Inegöl Manor		Manda
9	53 Idfu Parkway		Nantou
10	1795 Santiago de Compostela Way		Texas
11	900 Santiago de Compostela Parkway		Centra
12	478 Joliet Way		Hamilt

OK, 603 record… View: Pages: Edit:

```
sql>
```

Performance Dashboard (HeatWave)

Editor: Performance Dashboard | Graph Colors: Classic | Time Range: 150 sec

Server Performance | MLE Performance

Network Status

Statistics for network traffic sent and received by the MySQL server over the client connections.

MySQL Status

Primary MySQL server activity and performance statistics.

InnoDB Status

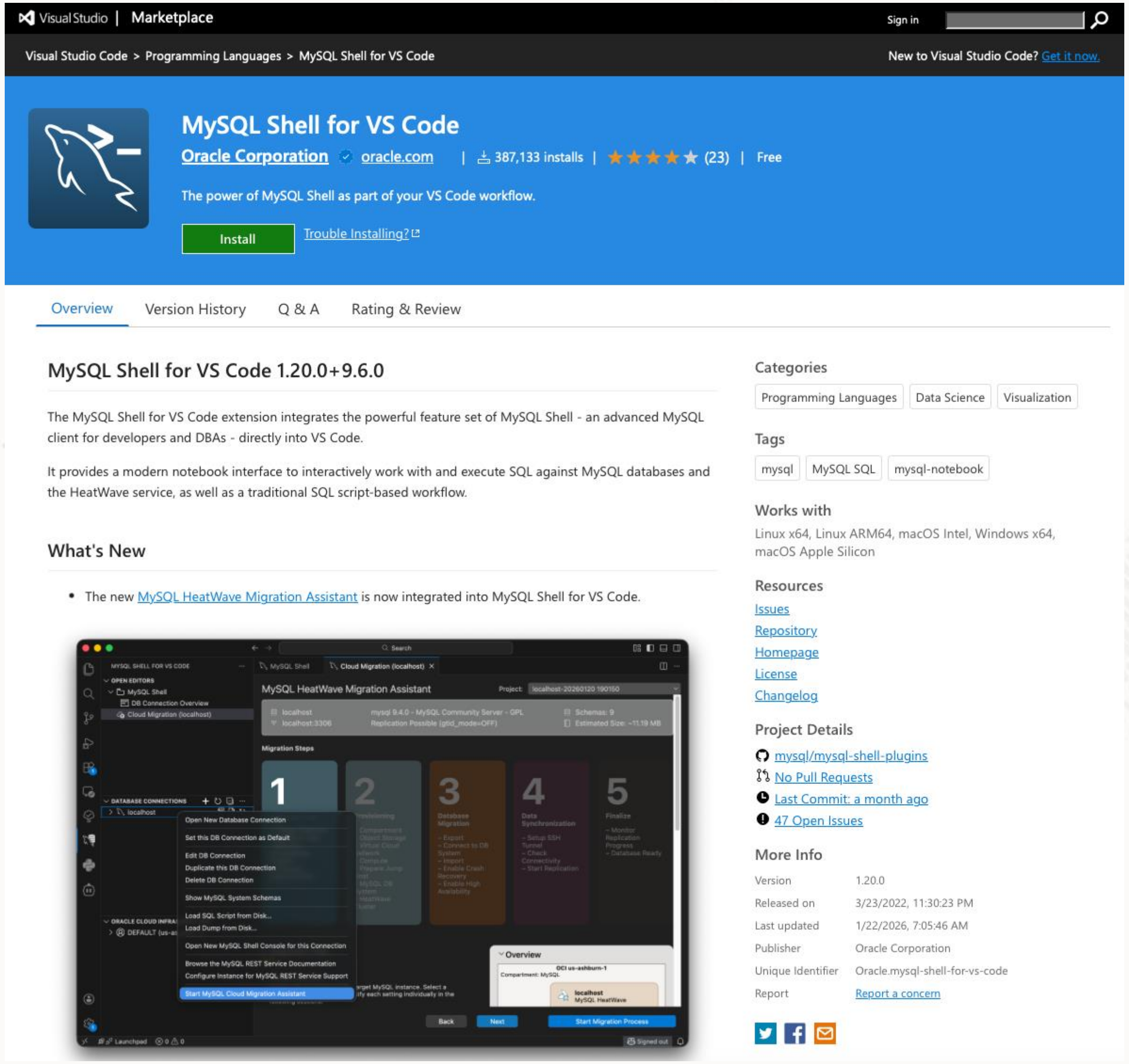
Overview of the InnoDB buffer pool and disk activity generated by the InnoDB storage engine.

```
Incoming Data          25.44 KB/s

Cache Efficiency             100%
Total Opened Tables        36,731
Total Transactions        446,814

Read Requests    3109 pages/s
Write Requests    285 pages/s
Disk Reads              0 #/s

Outgoing Data          58.81 KB/s

Total Statements            199/s

Log data writt...  21.46 KB...
Log writes             32 #/s
Writing           221.53 KB/s

SELECT  169/s
INSERT    1/s   CREATE   0/s
UPDATE   29/s   ALTER    0/s
DELETE    0/s   DROP     0/s
```

⊗ 0 ⚠ 0

Visual Studio Code Marketplace

https://marketplace.visualstudio.com/items?itemName=Oracle.mysql-shell-for-vs-code



Visual Studio | Marketplace

Sign in

Visual Studio Code > Programming Languages > MySQL Shell for VS Code

New to Visual Studio Code? Get it now.

MySQL Shell for VS Code

Oracle Corporation oracle.com | 387,133 installs | ★★★★★ (23) | Free

The power of MySQL Shell as part of your VS Code workflow.

Install

Trouble Installing?

Overview | Version History | Q & A | Rating & Review

MySQL Shell for VS Code 1.20.0+9.6.0

The MySQL Shell for VS Code extension integrates the powerful feature set of MySQL Shell - an advanced MySQL client for developers and DBAs - directly into VS Code.

It provides a modern notebook interface to interactively work with and execute SQL against MySQL databases and the HeatWave service, as well as a traditional SQL script-based workflow.

What's New

- The new MySQL HeatWave Migration Assistant is now integrated into MySQL Shell for VS Code.

MySQL HeatWave Migration Assistant
Migration Steps
1
2
3
4
5
Open New Database Connection
Set this DB Connection as Default
Edit DB Connection
Duplicate this DB Connection
Delete DB Connection
Show MySQL System Schemas
Load SQL Script from Disk...
Load Dump from Disk...
Open New MySQL Shell Console for this Connection
Browse the MySQL REST Service Documentation
Configure Instance for MySQL REST Service Support
Start MySQL Cloud Migration Assistant
Overview
Back
Next
Start Migration Process

Categories

Programming Languages | Data Science | Visualization

Tags

mysql | MySQL SQL | mysql-notebook

Works with

Linux x64, Linux ARM64, macOS Intel, Windows x64, macOS Apple Silicon

Resources

Issues
Repository
Homepage
License
Changelog

Project Details

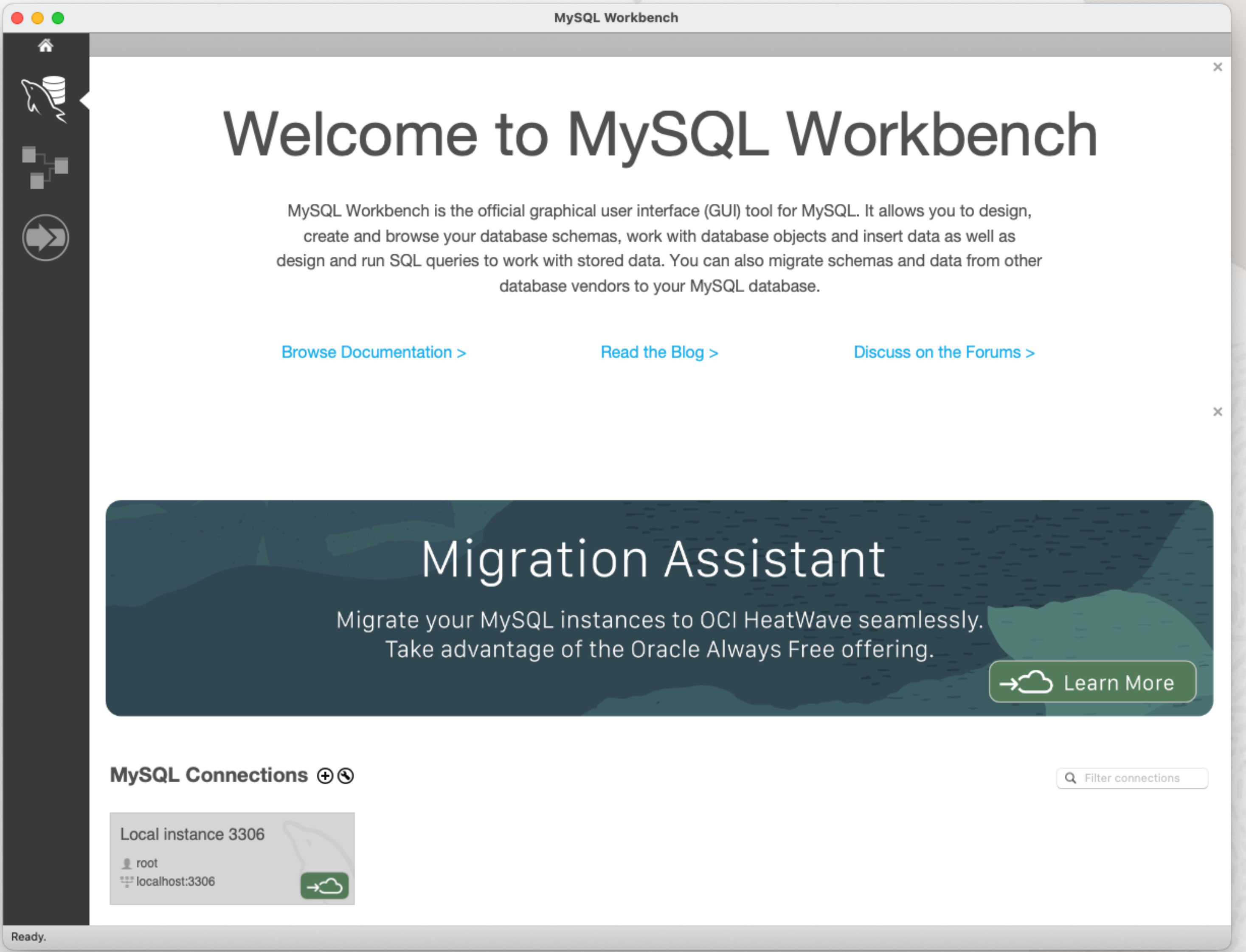
mysql/mysql-shell-plugins
No Pull Requests
Last Commit: a month ago
47 Open Issues

More Info

Version	1.20.0
Released on	3/23/2022, 11:30:23 PM
Last updated	1/22/2026, 7:05:46 AM
Publisher	Oracle Corporation
Unique Identifier	Oracle.mysql-shell-for-vs-code
Report	Report a concern

MySQL Workbench v8.0.45+

https://dev.mysql.com/downloads/workbench



MySQL Workbench

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation >

Read the Blog >

Discuss on the Forums >

Migration Assistant

Migrate your MySQL instances to OCI HeatWave seamlessly.
Take advantage of the Oracle Always Free offering.

Learn More

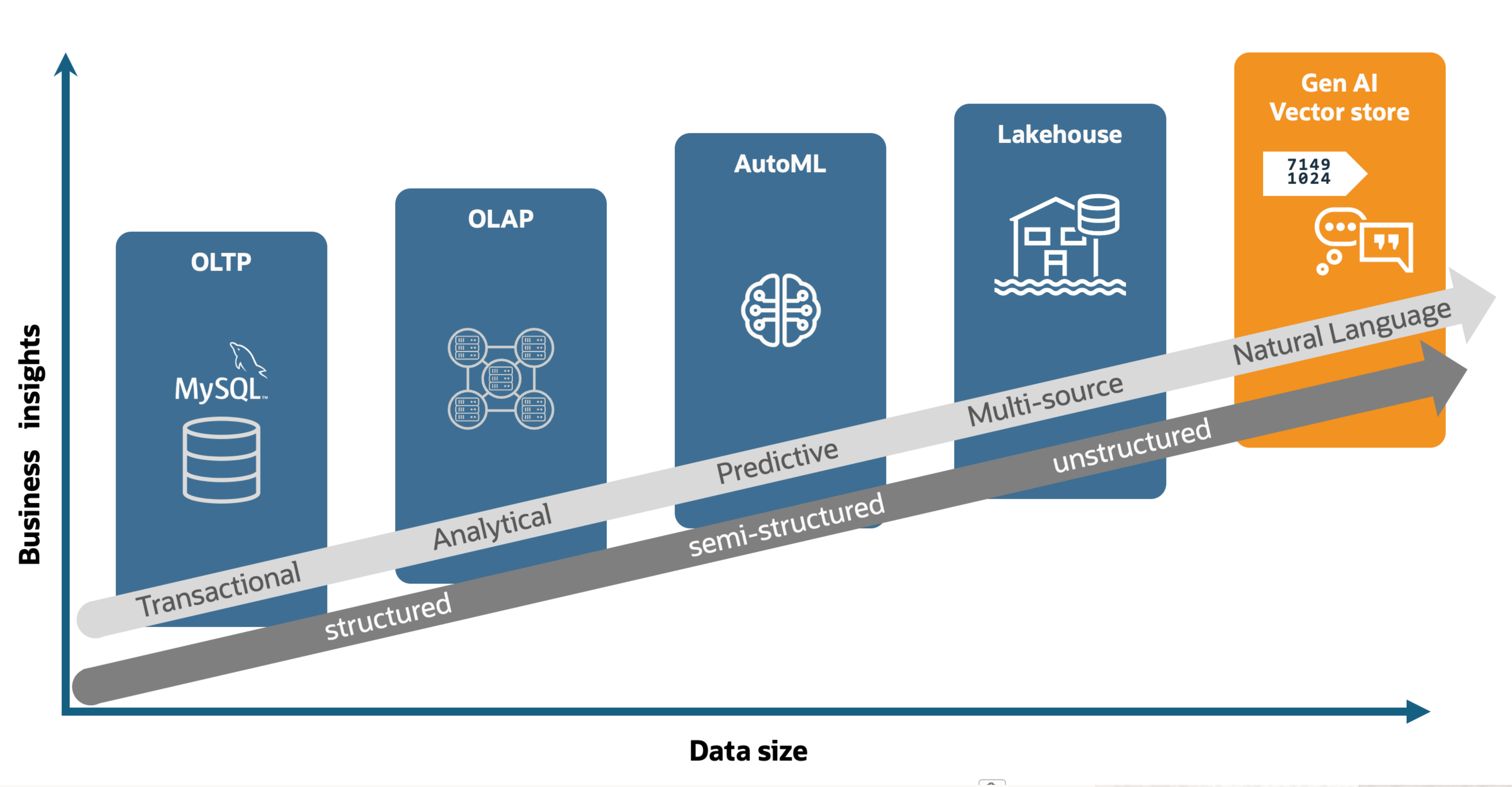
MySQL Connections

Filter connections

Local instance 3306
root
localhost:3306

Ready.

MySQL HeatWave



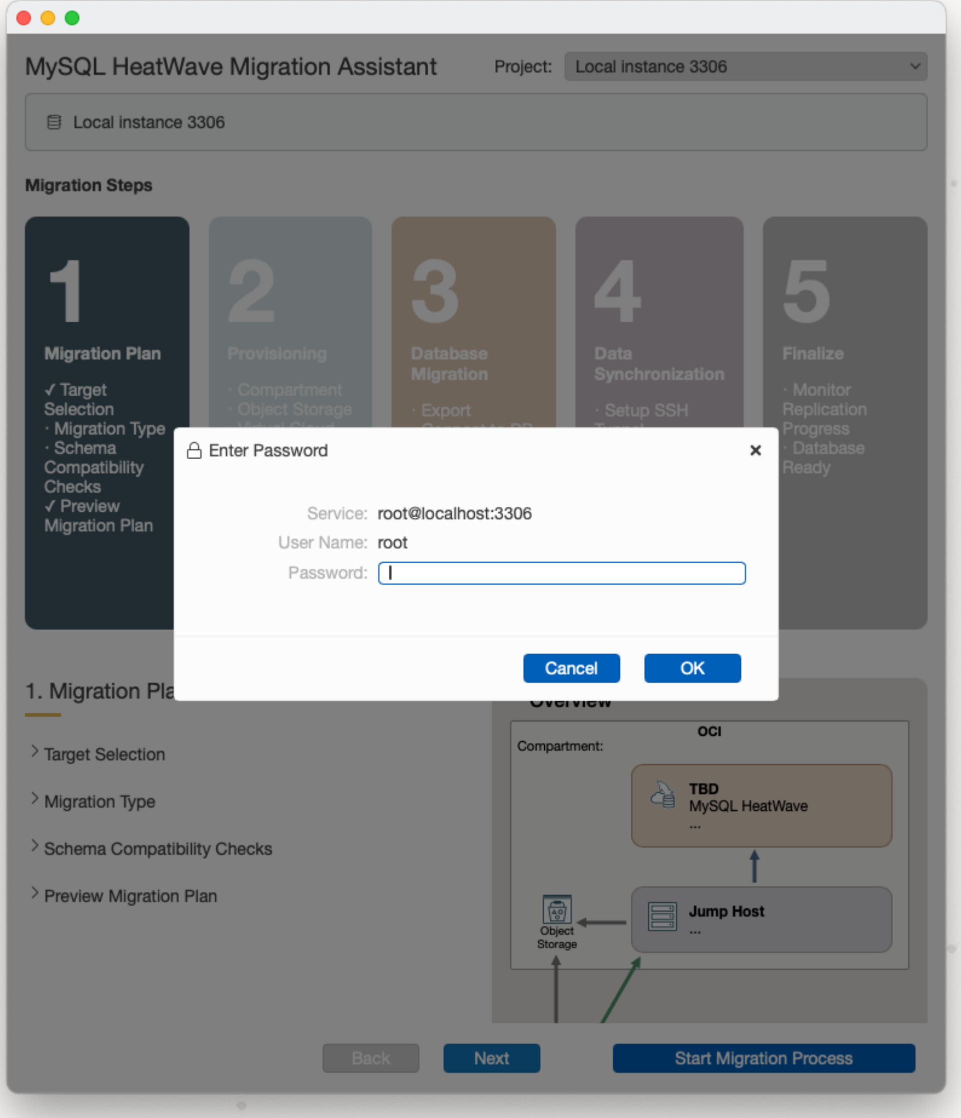
MySQL Shell Utilities

The Migration Assistant leverages several MySQL Shell utilities under the hood

	Purpose
util.dumpInstance()	Exports entire database instance to OCI Object Storage using parallel threads. Supports compatibility transformations and streaming via Pre-Authenticated Requests (PAR)
util.dumpInstance() --dry-run --ocimds	Performs schema analysis without exporting data. Identifies MDS compatibility issues and reports required transformations
util.loadDump()	Imports dump into target DB System with parallel loading, progress tracking, and resume capability via progress files
util.checkForServerUpgrade()	Analyzes source database for upgrade path issues when migrating from older MySQL versions (5.6, 5.7, 8.0, 8.4) to latest HeatWave
util.checkForServerUpgrade() list	Retrieves available upgrade checks to selectively run relevant ones

User Interface

UI Quick Glance



MySQL HeatWave Migration Assistant

Project: Local instance 3306

Local instance 3306

Migration Steps

1 Migration Plan
✓ Target Selection
· Migration Type
· Schema Compatibility Checks
✓ Preview Migration Plan

2 Provisioning
· Compartment
· Object Storage

3 Database Migration
· Export

4 Data Synchronization
· Setup SSH

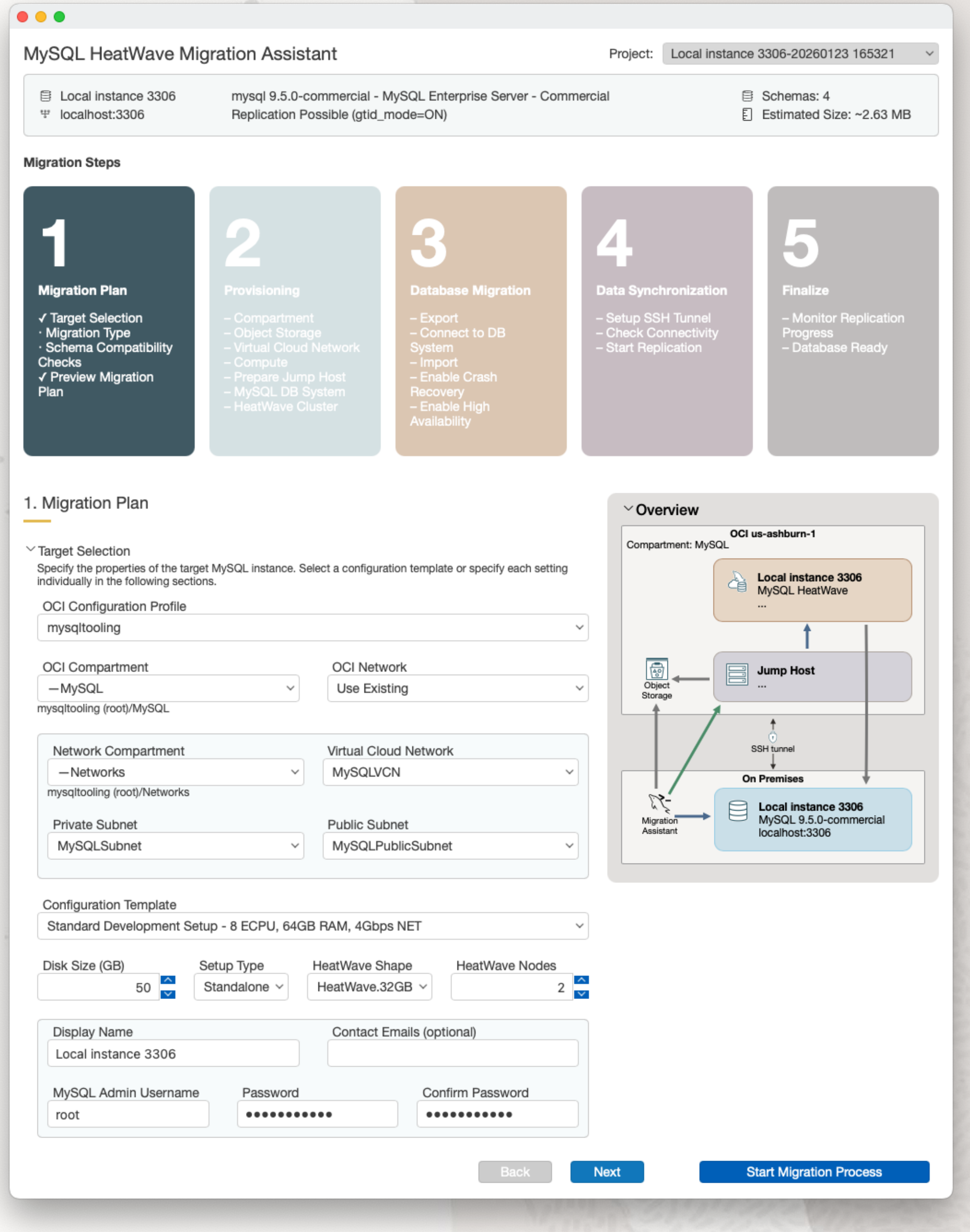
5 Finalize
· Monitor Replication Progress
· Database Ready

Enter Password

Service: root@localhost:3306
User Name: root
Password:

1. Migration Pla

› Target Selection
› Migration Type
› Schema Compatibility Checks
› Preview Migration Plan



MySQL HeatWave Migration Assistant

Project: Local instance 3306-20260123 165321

Local instance 3306
localhost:3306

mysql 9.5.0-commercial - MySQL Enterprise Server - Commercial
Replication Possible (gtid_mode=ON)

Schemas: 4
Estimated Size: ~2.63 MB

Migration Steps

1 Migration Plan
✓ Target Selection
· Migration Type
· Schema Compatibility Checks
✓ Preview Migration Plan

2 Provisioning
– Compartment
– Object Storage
– Virtual Cloud Network
– Compute
– Prepare Jump Host
– MySQL DB System
– HeatWave Cluster

3 Database Migration
– Export
– Connect to DB System
– Import
– Enable Crash Recovery
– Enable High Availability

4 Data Synchronization
– Setup SSH Tunnel
– Check Connectivity
– Start Replication

5 Finalize
– Monitor Replication Progress
– Database Ready

1. Migration Plan

Target Selection

Specify the properties of the target MySQL instance. Select a configuration template or specify each setting individually in the following sections.

OCI Configuration Profile: mysqltooling

OCI Compartment: —MySQL
mysqltooling (root)/MySQL

OCI Network: Use Existing

Network Compartment: —Networks
mysqltooling (root)/Networks

Virtual Cloud Network: MySQLVCN

Private Subnet: MySQLSubnet

Public Subnet: MySQLPublicSubnet

Configuration Template: Standard Development Setup - 8 ECPU, 64GB RAM, 4Gbps NET

Disk Size (GB): 50

Setup Type: Standalone

HeatWave Shape: HeatWave.32GB

HeatWave Nodes: 2

Display Name: Local instance 3306

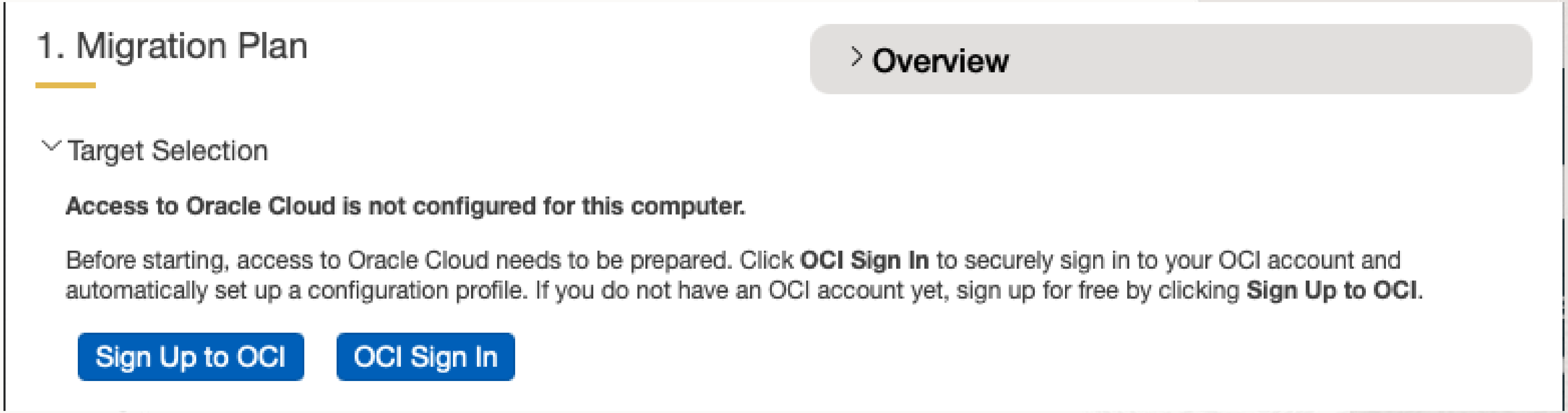
Contact Emails (optional):

MySQL Admin Username: root

Password: ●●●●●●●●●●●

Confirm Password: ●●●●●●●●●●●

Log in to OCI (or create an account)



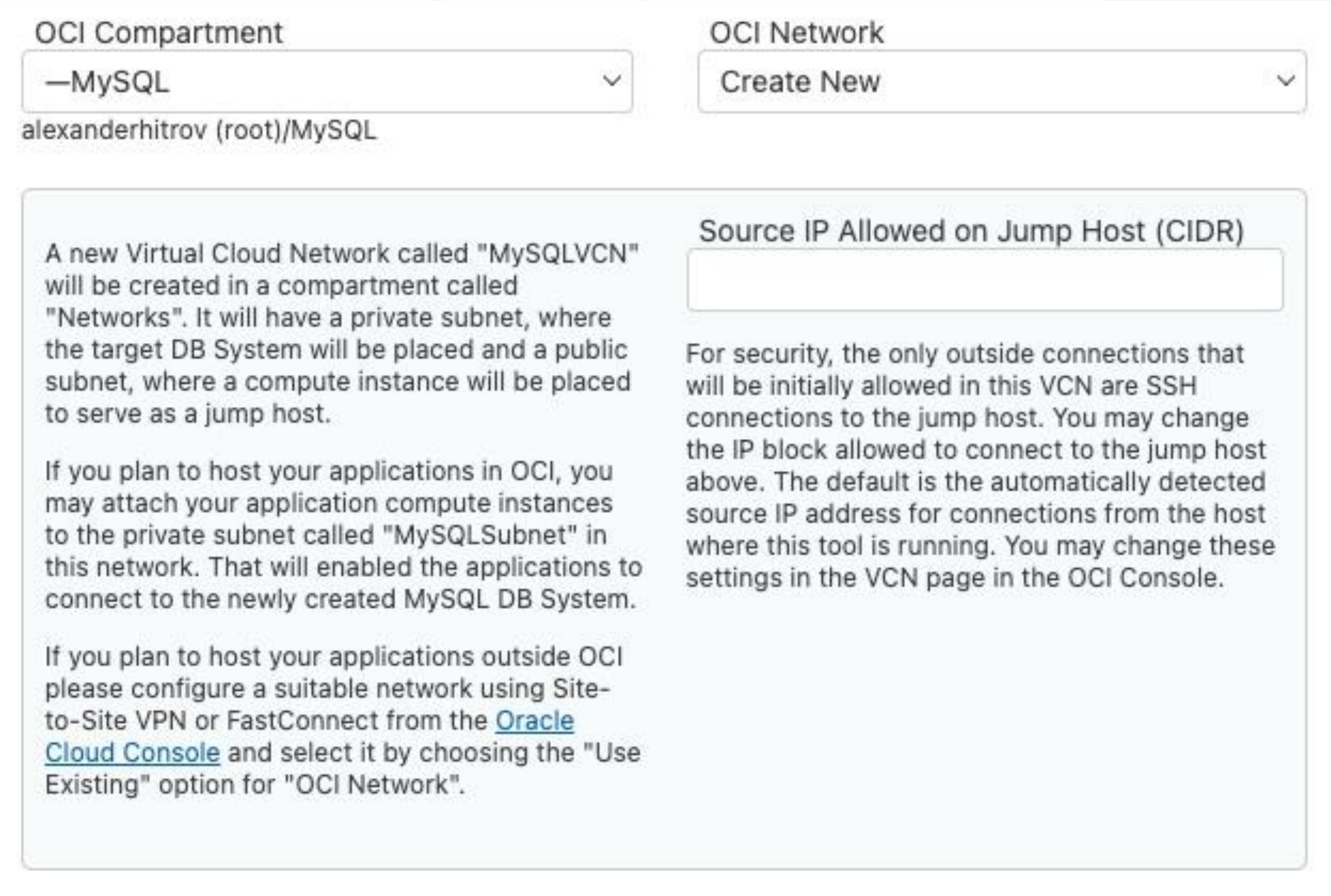
1. Migration Plan

⌄ Target Selection

Access to Oracle Cloud is not configured for this computer.

Before starting, access to Oracle Cloud needs to be prepared. Click **OCI Sign In** to securely sign in to your OCI account and automatically set up a configuration profile. If you do not have an OCI account yet, sign up for free by clicking **Sign Up to OCI**.

Automatic Network Provisioning



OCI Compartment

—MySQL

alexanderhitrov (root)/MySQL

OCI Network

Create New

A new Virtual Cloud Network called "MySQLVCN" will be created in a compartment called "Networks". It will have a private subnet, where the target DB System will be placed and a public subnet, where a compute instance will be placed to serve as a jump host.

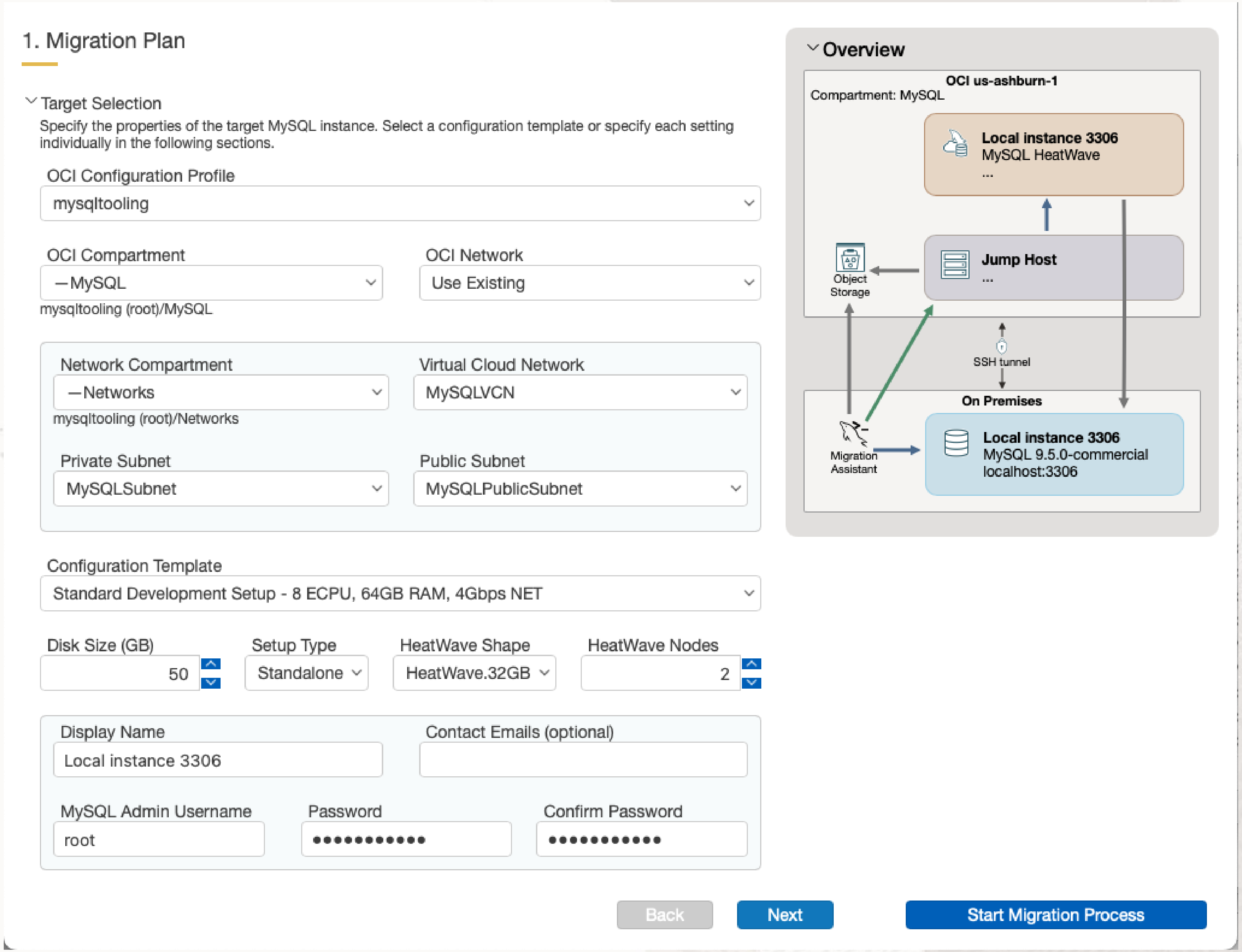
If you plan to host your applications in OCI, you may attach your application compute instances to the private subnet called "MySQLSubnet" in this network. That will enabled the applications to connect to the newly created MySQL DB System.

If you plan to host your applications outside OCI please configure a suitable network using Site-to-Site VPN or FastConnect from the Oracle Cloud Console and select it by choosing the "Use Existing" option for "OCI Network".

Source IP Allowed on Jump Host (CIDR)

For security, the only outside connections that will be initially allowed in this VCN are SSH connections to the jump host. You may change the IP block allowed to connect to the jump host above. The default is the automatically detected source IP address for connections from the host where this tool is running. You may change these settings in the VCN page in the OCI Console.

Target Selection



1. Migration Plan

Target Selection

Specify the properties of the target MySQL instance. Select a configuration template or specify each setting individually in the following sections.

OCI Configuration Profile: mysqltooling

OCI Compartment: —MySQL

mysqltooling (root)/MySQL

OCI Network: Use Existing

Network Compartment: —Networks

mysqltooling (root)/Networks

Virtual Cloud Network: MySQLVCN

Private Subnet: MySQLSubnet

Public Subnet: MySQLPublicSubnet

Configuration Template: Standard Development Setup - 8 ECPU, 64GB RAM, 4Gbps NET

Disk Size (GB): 50

Setup Type: Standalone

HeatWave Shape: HeatWave.32GB

HeatWave Nodes: 2

Display Name: Local instance 3306

Contact Emails (optional):

MySQL Admin Username: root

Password: ●●●●●●●●●●●

Confirm Password: ●●●●●●●●●●●

Pre-Configured Shape Templates

The assistant offers several pre-configured templates for common use cases

	DB System	Compute	Notes
Always Free	MySQL.Free (2 ECPU, 8GB)	VM.Standard.A1.Flex (2 OCPU, 12GB)	DBS is limited to 1 per tenancy
Small Development	MySQL.2 (2 ECPU, 16GB)	VM.Standard.E5.Flex (2 OCPU, 16GB)	Good for testing
Standard Development	MySQL.8 (8 ECPU, 64GB)	VM.Standard.E5.Flex (4 OCPU, 32GB)	Default selection
Large Development	MySQL.64 (64 ECPU, 512GB)	VM.Standard.E5.Flex (12 OCPU, 96GB)	Production-scale testing
Production Workloads (Small, Standard, Large)	Same as dev	Same as dev	Enables backups, HA, storage auto-expansion
Custom	User-configured	User-configured	Disables backups, HA, storage auto-expansion

Plan Steps

	Description
OCI Profile	Select OCI configuration profile, authenticate to cloud
Source Database	Configure connection to source MySQL, validate connectivity, detect server type
Migration Type	Choose Cold (offline) or Hot (online with replication) migration
Schema Compatibility Checks	Run compatibility analysis, resolve issues
Target Options	Configure DB System shape, storage, HA, HeatWave cluster
Preview Migration Plan	Review final configuration before execution

Provisioning

	Description
Compartment	Create or verify OCI compartment
Object Storage	Create bucket for temporary dump storage
Virtual Cloud Network	Provision VCN with public/private subnets
Compute	Launch compute instance for jump host
Prepare Jump Host	Install MySQL Shell and migration helper on compute
MySQL DB System	Launch target HeatWave DB System
HeatWave Cluster	Add HeatWave cluster if configured

Migration

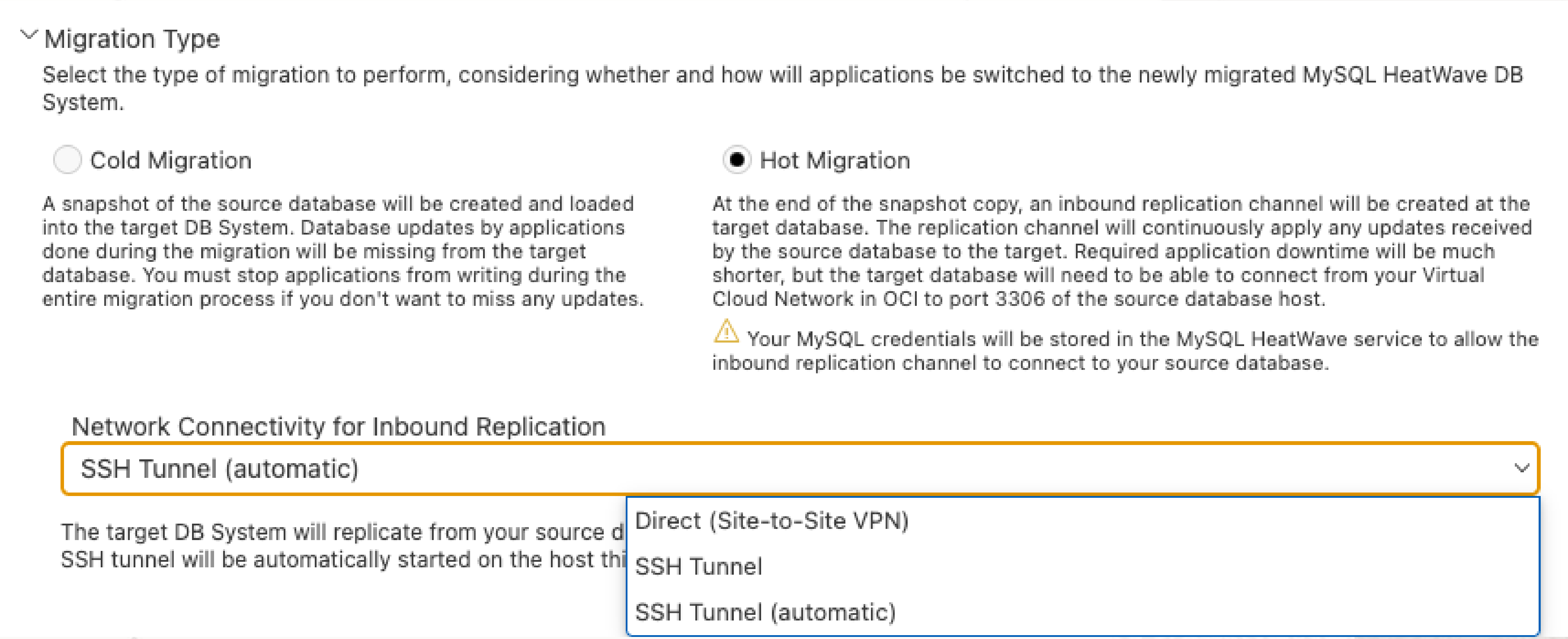
	Description
Export	Dump source database to Object Storage
Connect to DB System	Verify jump host can reach target DB System
Import	Load dump into target via jump host
Enable Crash Recovery	Re-enable InnoDB crash recovery (disabled during load)
Enable High Availability	Enable HA if configured

Sync

	Description
Setup SSH Tunnel	Establish tunnel for replication (hot migration only)
Check Connectivity	Verify network path from OCI to source
Start Replication	Create inbound replication channel
Monitor Replication Progress	Track replication lag until cutover

Migration Type

Migration Type



Select the type of migration to perform, considering whether and how will applications be switched to the newly migrated MySQL HeatWave DB System.

○ Cold Migration

A snapshot of the source database will be created and loaded into the target DB System. Database updates by applications done during the migration will be missing from the target database. You must stop applications from writing during the entire migration process if you don't want to miss any updates.

● Hot Migration

At the end of the snapshot copy, an inbound replication channel will be created at the target database. The replication channel will continuously apply any updates received by the source database to the target. Required application downtime will be much shorter, but the target database will need to be able to connect from your Virtual Cloud Network in OCI to port 3306 of the source database host.

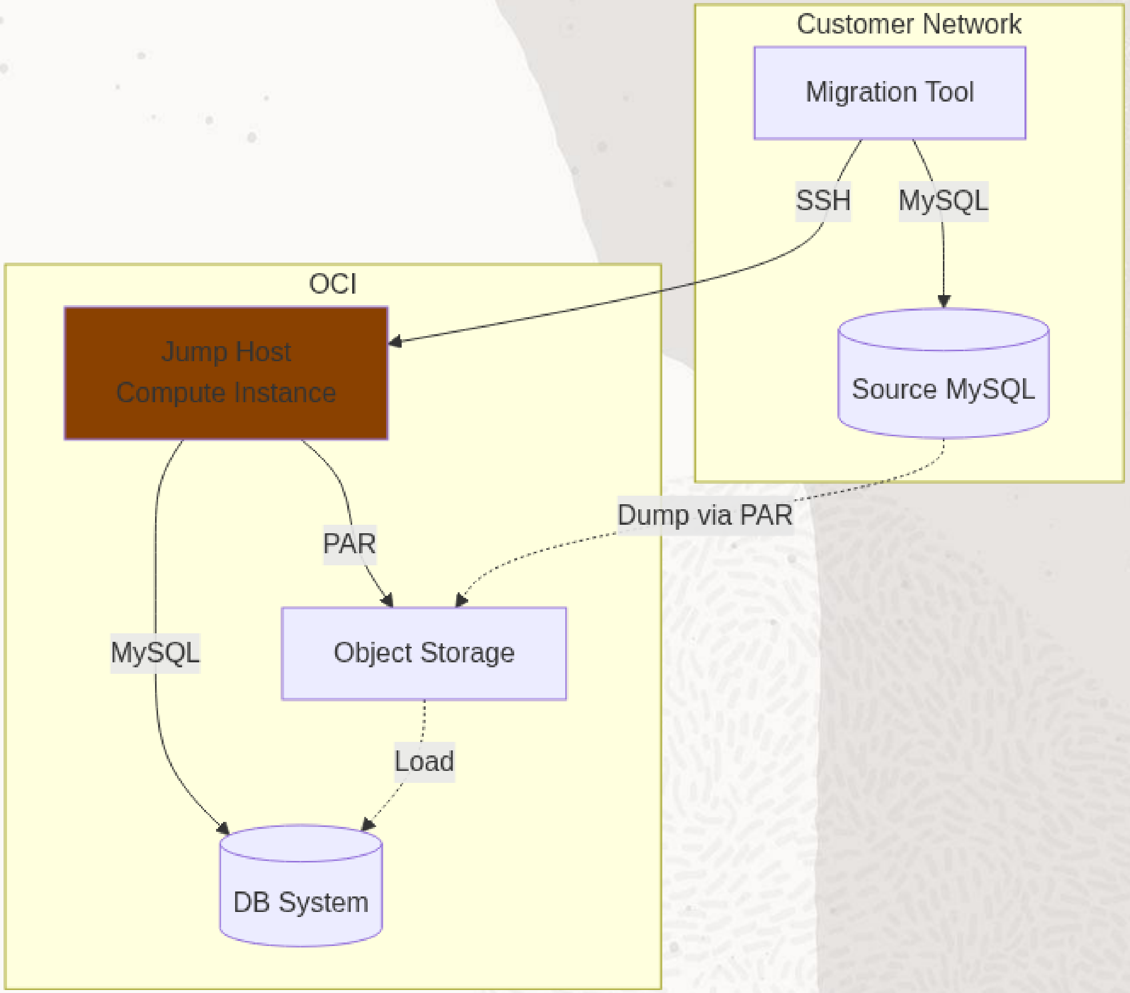
⚠ Your MySQL credentials will be stored in the MySQL HeatWave service to allow the inbound replication channel to connect to your source database.

Network Connectivity for Inbound Replication: SSH Tunnel (automatic)

The target DB System will replicate from your source d
SSH tunnel will be automatically started on the host thi

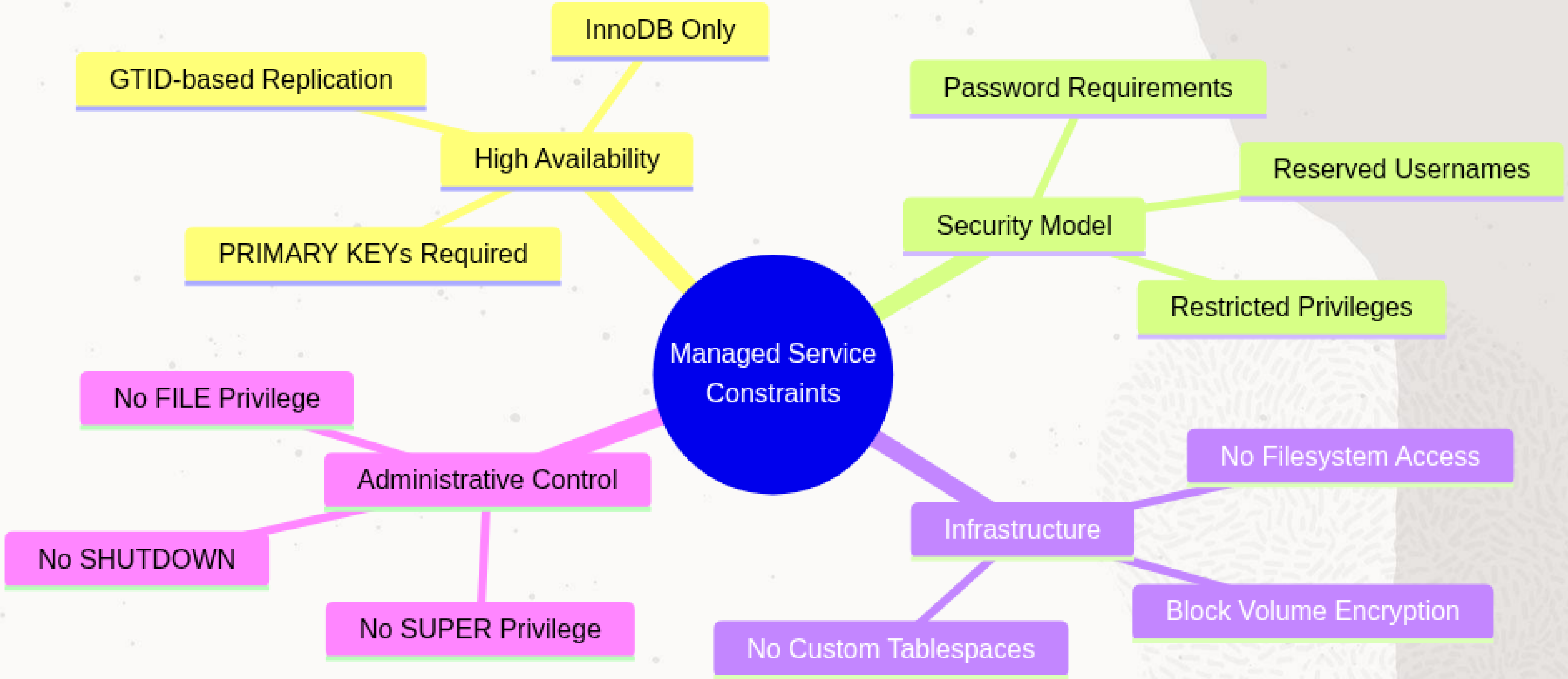
The Jump Host Pattern

- Bridges the network boundary: Compute instance in OCI that can reach the private DB System (which has no public IP)
- Runs MySQL Shell load: *util.loadDump()* executes on the jump host with low-latency access to both Object Storage and DB System
- SSH tunnel endpoint: For hot migrations, relays replication traffic from OCI back to the source MySQL
- Remote helper (Python script) is deployed to the jump host and acts as a bridge between the migration tool and MySQL Shell operations running inside OCI.



Compatibility Checks

Why Cloud Compatibility Checks Matter



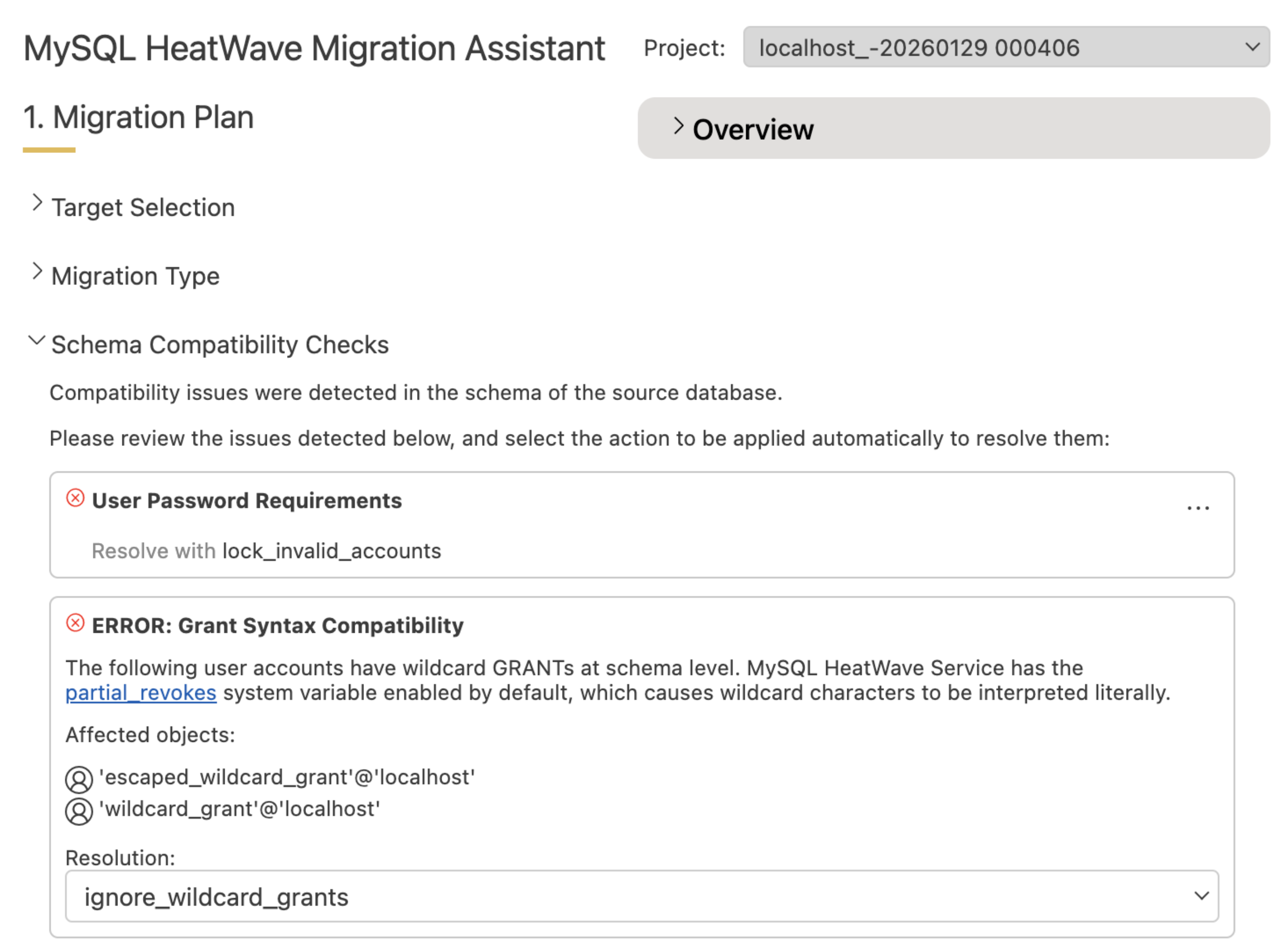
Constraint Categories [1]

	Why It Exists	Resulting Compatibility Checks
High Availability	HeatWave uses Group Replication for HA. InnoDB with proper PKs is required for conflict detection and row-based replication	missing_pk, unsupported_engine
Security Isolation	Multi-tenant cloud: users cannot access filesystem, cannot shut down the instance, cannot use accounts reserved for Oracle management	restricted_grants, reserved_usernames, no_password
Block-Level Encryption	OCI encrypts storage at the block volume level. Per-table encryption options are redundant and unsupported	schema/encryption, table/encryption
No Filesystem Access	Users cannot specify *DATA DIRECTORY* or *INDEX DIRECTORY*. All data lives in managed storage	data_or_index_directory

Constraint Categories [2]

	Why It Exists	Resulting Compatibility Checks
DEFINER Restrictions	Objects cannot have *DEFINER* set to accounts that don't exist or are reserved. Prevents privilege escalation	restricted_definer, invalid_definer, strip_definers
Case Sensitivity	HeatWave enforces *lower_case_table_names=1*. Views referencing tables with inconsistent casing will break	mismatched_reference
Partial Revokes	partial_revokes=ON by default. Wildcard grants like db_% are interpreted literally, not as patterns	wildcard_grant, escaped_wildcard_grant
Auth Plugin Support	Only certain authentication plugins are supported. Legacy plugins like *mysql_native_password* are deprecated or disabled by default	deprecated_auth_plugin, unsupported_auth_plugin

Compatibility Checks UI



MySQL HeatWave Migration Assistant

Project: localhost_-20260129 000406

1. Migration Plan

Schema Compatibility Checks

Compatibility issues were detected in the schema of the source database.

Please review the issues detected below, and select the action to be applied automatically to resolve them:

⊗ **User Password Requirements**

Resolve with lock_invalid_accounts

⊗ **ERROR: Grant Syntax Compatibility**

The following user accounts have wildcard GRANTs at schema level. MySQL HeatWave Service has the partial_revokes system variable enabled by default, which causes wildcard characters to be interpreted literally.

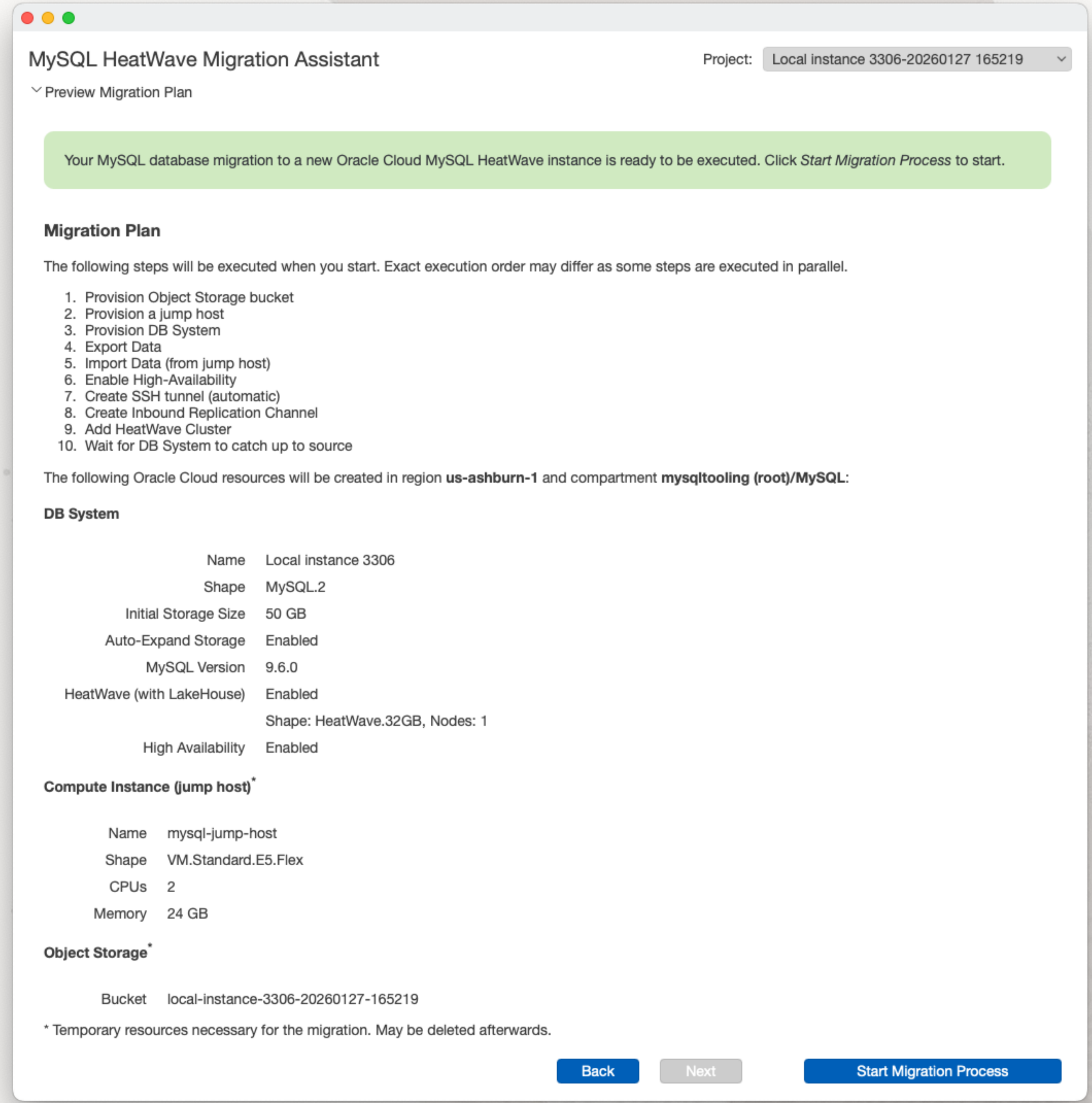
Affected objects:

'escaped_wildcard_grant'@'localhost'
'wildcard_grant'@'localhost'

Resolution: ignore_wildcard_grants

Time to Migrate

In parallel and fast as possible



MySQL HeatWave Migration Assistant
Project: Local instance 3306-20260127 165219
Preview Migration Plan

Your MySQL database migration to a new Oracle Cloud MySQL HeatWave instance is ready to be executed. Click *Start Migration Process* to start.

Migration Plan

The following steps will be executed when you start. Exact execution order may differ as some steps are executed in parallel.

1. Provision Object Storage bucket
2. Provision a jump host
3. Provision DB System
4. Export Data
5. Import Data (from jump host)
6. Enable High-Availability
7. Create SSH tunnel (automatic)
8. Create Inbound Replication Channel
9. Add HeatWave Cluster
10. Wait for DB System to catch up to source

The following Oracle Cloud resources will be created in region **us-ashburn-1** and compartment **mysqltooling (root)/MySQL**:

DB System

Name	Local instance 3306
Shape	MySQL.2
Initial Storage Size	50 GB
Auto-Expand Storage	Enabled
MySQL Version	9.6.0
HeatWave (with LakeHouse)	Enabled Shape: HeatWave.32GB, Nodes: 1
High Availability	Enabled

Compute Instance (jump host)*

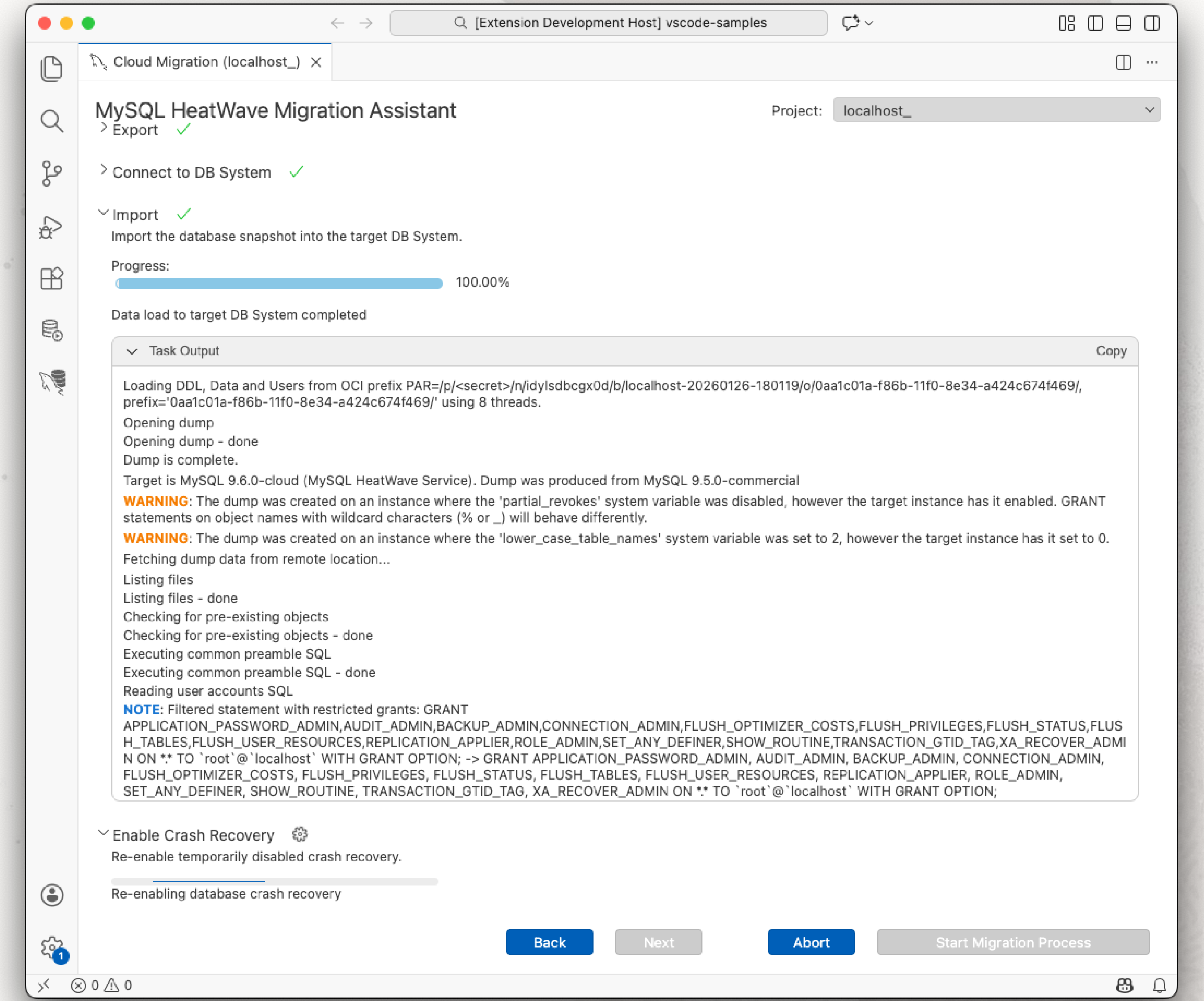
Name	mysql-jump-host
Shape	VM.Standard.E5.Flex
CPUs	2
Memory	24 GB

Object Storage*

Bucket	local-instance-3306-20260127-165219

* Temporary resources necessary for the migration. May be deleted afterwards.

Import logs


[Extension Development Host] vscode-samples
Cloud Migration (localhost_)
Back
Next
Abort
Start Migration Process
0 0

MySQL HeatWave Migration Assistant

Project: localhost_

Export ✓

Connect to DB System ✓

Import ✓

Import the database snapshot into the target DB System.

Progress: 100.00%

Data load to target DB System completed

Task Output — Copy

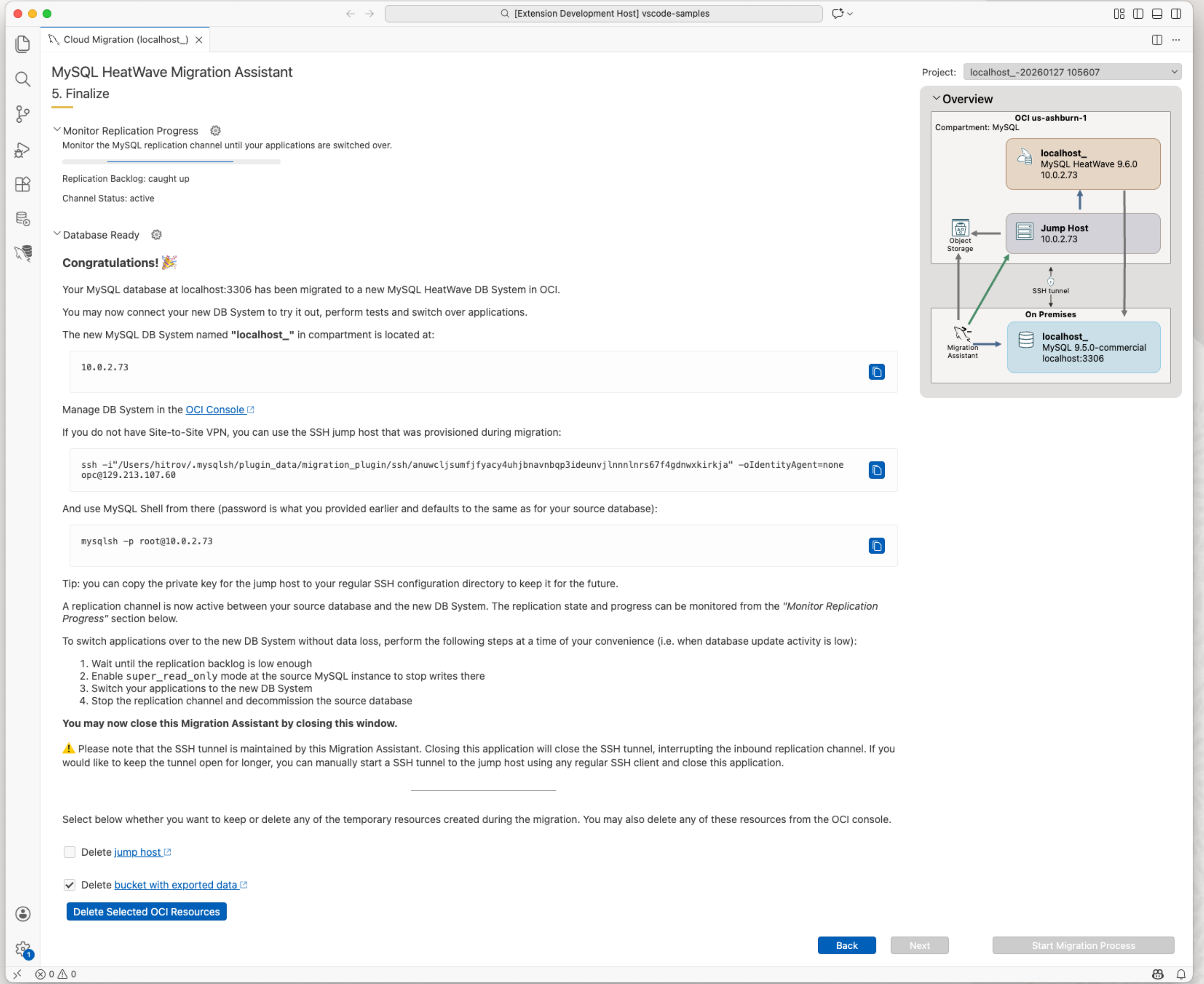
```
Loading DDL, Data and Users from OCI prefix PAR=/p/<secret>/n/idylsdbcgx0d/b/localhost-20260126-180119/o/0aa1c01a-f86b-11f0-8e34-a424c674f469/, prefix='0aa1c01a-f86b-11f0-8e34-a424c674f469/' using 8 threads.
Opening dump
Opening dump - done
Dump is complete.
Target is MySQL 9.6.0-cloud (MySQL HeatWave Service). Dump was produced from MySQL 9.5.0-commercial
WARNING: The dump was created on an instance where the 'partial_revokes' system variable was disabled, however the target instance has it enabled. GRANT statements on object names with wildcard characters (% or _) will behave differently.
WARNING: The dump was created on an instance where the 'lower_case_table_names' system variable was set to 2, however the target instance has it set to 0.
Fetching dump data from remote location...
Listing files
Listing files - done
Checking for pre-existing objects
Checking for pre-existing objects - done
Executing common preamble SQL
Executing common preamble SQL - done
Reading user accounts SQL
NOTE: Filtered statement with restricted grants: GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ADMIN,BACKUP_ADMIN,CONNECTION_ADMIN,FLUSH_OPTIMIZER_COSTS,FLUSH_PRIVILEGES,FLUSH_STATUS,FLUSH_TABLES,FLUSH_USER_RESOURCES,REPLICATION_APPLIER,ROLE_ADMIN,SET_ANY_DEFINER,SHOW_ROUTINE,TRANSACTION_GTID_TAG,XA_RECOVER_ADMIN ON *.* TO `root`@`localhost` WITH GRANT OPTION; -> GRANT APPLICATION_PASSWORD_ADMIN, AUDIT_ADMIN, BACKUP_ADMIN, CONNECTION_ADMIN, FLUSH_OPTIMIZER_COSTS, FLUSH_PRIVILEGES, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES, REPLICATION_APPLIER, ROLE_ADMIN, SET_ANY_DEFINER, SHOW_ROUTINE, TRANSACTION_GTID_TAG, XA_RECOVER_ADMIN ON *.* TO `root`@`localhost` WITH GRANT OPTION;
```

Enable Crash Recovery

Re-enable temporarily disabled crash recovery.

Re-enabling database crash recovery

Database Ready



MySQL HeatWave Migration Assistant

5. Finalize

Monitor Replication Progress

Monitor the MySQL replication channel until your applications are switched over.

Replication Backlog: caught up

Channel Status: active

Database Ready

Congratulations! 🎉

Your MySQL database at localhost:3306 has been migrated to a new MySQL HeatWave DB System in OCI.

You may now connect your new DB System to try it out, perform tests and switch over applications.

The new MySQL DB System named **"localhost_"** in compartment is located at:

```
10.0.2.73
```

Manage DB System in the OCI Console

If you do not have Site-to-Site VPN, you can use the SSH jump host that was provisioned during migration:

```
ssh -i"/Users/hitrov/.mysqlsh/plugin_data/migration_plugin/ssh/anuwcljsumfjfyacy4uhjbnavnbqp3ideunvjlnnnlnrs67f4gdnwxkirkja" -oIdentityAgent=none
opc@129.213.107.60
```

And use MySQL Shell from there (password is what you provided earlier and defaults to the same as for your source database):

```
mysqlsh -p root@10.0.2.73
```

Tip: you can copy the private key for the jump host to your regular SSH configuration directory to keep it for the future.

A replication channel is now active between your source database and the new DB System. The replication state and progress can be monitored from the *"Monitor Replication Progress"* section below.

To switch applications over to the new DB System without data loss, perform the following steps at a time of your convenience (i.e. when database update activity is low):

1. Wait until the replication backlog is low enough
2. Enable `super_read_only` mode at the source MySQL instance to stop writes there
3. Switch your applications to the new DB System
4. Stop the replication channel and decommission the source database

You may now close this Migration Assistant by closing this window.

⚠ Please note that the SSH tunnel is maintained by this Migration Assistant. Closing this application will close the SSH tunnel, interrupting the inbound replication channel. If you would like to keep the tunnel open for longer, you can manually start a SSH tunnel to the jump host using any regular SSH client and close this application.

Select below whether you want to keep or delete any of the temporary resources created during the migration. You may also delete any of these resources from the OCI console.

□ Delete jump host

☑ Delete bucket with exported data

Delete Selected OCI Resources

Demo

For a deeper dive – check out this video

MySQL HeatWave Migration Assistant

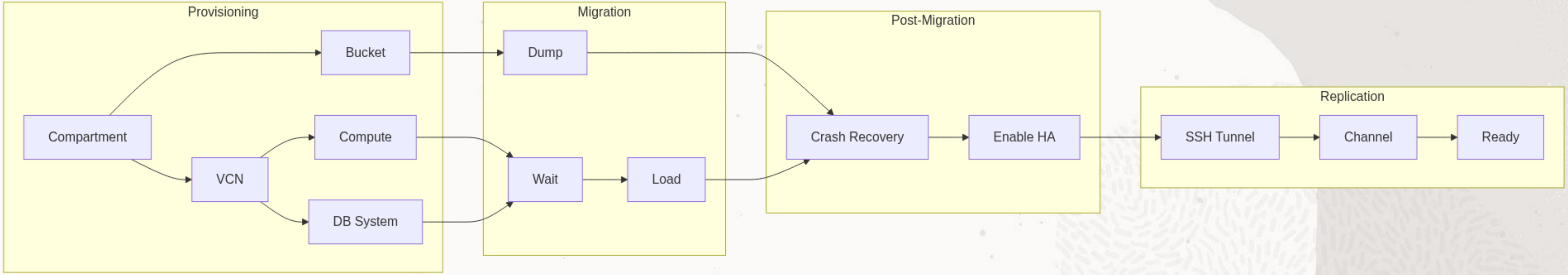
Seamless end-to-end migration from on-premise to OCI

Advanced

How Certain Things Work Under the Hood

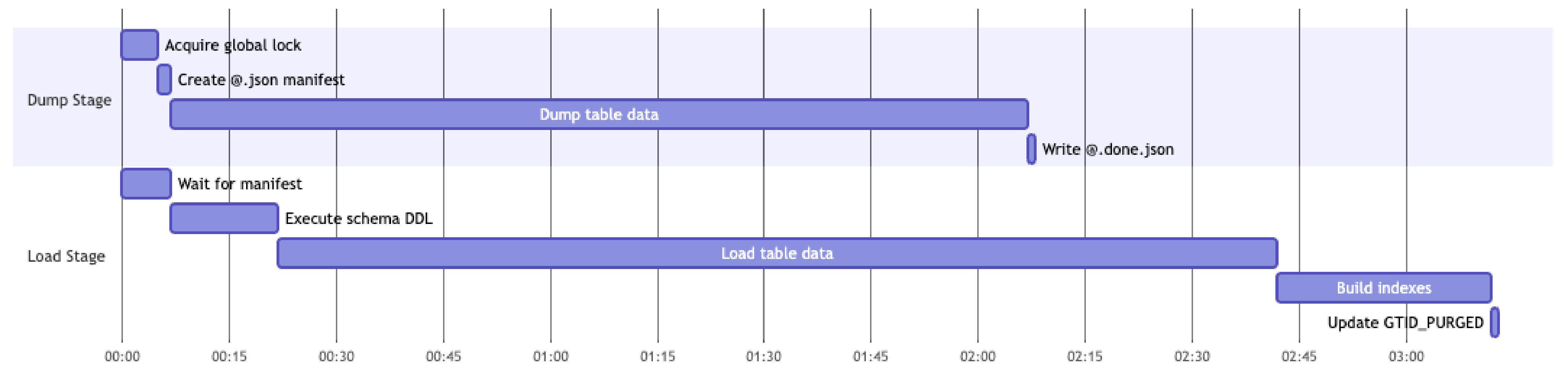
DAG - Directed Acyclic Graph - Task Scheduler

The execution phase uses a directed acyclic graph (DAG) where each stage knows its dependencies. Stages run in parallel when possible - for example, bucket creation and VCN provisioning can happen simultaneously since they both only depend on compartment creation.



The dependency graph is built at orchestrator initialization. Each stage declares its dependencies, and the scheduler automatically parallelizes where possible. The dump and load stages have a special relationship—load starts when dump begins producing data, not when dump completes

Streaming: Dump and Load



Compatibility Flags (Auto-Resolution Options) [1]

	What It Fixes
create_invisible_pks	Tables without primary keys get an invisible auto-increment PK added. Required for HeatWave/HA
force_innodb	Non-InnoDB tables (MyISAM, MEMORY, etc.) are converted to InnoDB during import
strip_definers	Removes DEFINER clauses from views, routines, triggers, events. Objects run as invoker
strip_tablespaces	Removes tablespace assignments. All tables use default InnoDB tablespace in HeatWave
strip_restricted_grants	Removes privileges not allowed in HeatWave (SUPER, FILE, SHUTDOWN, etc.)
strip_invalid_grants	Removes grants on non-existent objects

Compatibility Flags (Auto-Resolution Options) [2]

	What It Fixes
skip_invalid_accounts	Skips user accounts that cannot be migrated (system accounts, special plugins)
lock_invalid_accounts	Migrates accounts with unsupported auth plugins but locks them
ignore_missing_pks	Proceeds without adding PKs (not recommended for HA/HeatWave)
ignore_wildcard_grants	Keeps wildcard grants unchanged despite partial_revokes behavior difference
unescape_wildcard_grants	Converts escaped wildcards in grants to literal interpretation
force_non_standard_fks	Allows foreign keys across different storage engines

Binlog Retention Race

During migration, there's a race between dump completion and binlog purging at the source. The dump captures a GTID position, and after load completes, replication must fetch transactions from that position forward. If those binlogs have been purged, replication setup fails.

The assistant checks *binlog_expire_logs_seconds* during planning and warns if retention is less than 24 hours. For large databases where migration may take longer, users should increase retention before starting.

MySQL 8.0 EOL – April 2026

Key Benefits of upgrading to MySQL HeatWave 8.4 LTS

Improved Performance: MySQL 8.4 introduces optimizations within the InnoDB storage engine, enhancing query execution and resource efficiency.

Advanced Security Features: Security is significantly reinforced in 8.4.

Optimized Replication: Multi-threaded replication has been upgraded, ensures faster replica synchronization, enhancing performance and reliability in high-availability configurations.

Streamlined Architecture: Server related code optimizations have resulted in reduced system complexity, enhancing maintainability and operational stability.

Improve OLTP & mixed workload performance: In 8.4, HeatWave accelerates more workloads which includes SQL functions/operators support such as Window functions, analytic functions (HLL, CUBE, Qualify, table sample, grouping sets) and JSON functions.

HeatWave Lakehouse: load and query hundreds of terabytes of data from in Object Storage—in CSV, Avro, ND-JSON and Parquet formats.

HeatWave AutoML: In addition to support classification and regression tasks, HeatWave AutoML has added support for anomaly detection, forecasting and recommender system tasks

See [FAQ7437 MySQL Heatwave 8.0 End Of Life (EOL) FAQ]() for more details

MySQL 8.0 EOL – April 2026

Plan your upgrade ASAP. Recommend to upgrade to 8.4 LTS or 9.7 LTS

Check for any changes (What is New in MySQL 8.4 since MySQL 8.0) some breaking changes examples:

1. `default_authentication_plugin` is removed in 8.4
    1. New defaults is `caching_sha2_password`
    2. `mysql_native_password` is disabled by default; upgrade may temporarily enable it for existing users, but you should migrate users to `caching_sha2_password`
2. New reserved words: `MANUAL`, `PARALLEL`, `QUALIFY`, `TABLESAMPLE`
   Upgrade fails if you have `AUTO_INCREMENT` on `FLOAT/DOUBLE` columns (Oracle Documentation)

Run MySQL Shell upgrade checker

Run the MySQL Shell upgrade checker against your instance (catches a lot of "this will break" issues): `util.checkForServerUpgrade()` supports checks for upgrading 8.0 → 8.4. (MySQL Developer Zone)

Always test in your dev/test environment before upgrade your production environment

1. Setup a test environment with 8.4 version and run an end-to-end testing with your application
2. Check the version of the MySQL connector used in your application. Legacy connectors that support up to 8.0 may fail to connect

Always create a full backup before upgrade

2 options to upgrade (KB166277 Major version Upgrade MySQL HeatWave for more details)

1. in-place upgrade
2. Blue-green upgrade

See FAQ7437 MySQL Heatwave 8.0 End Of Life (EOL) FAQ for more details

Thank you

Grab the source code - and contribute if you'd like

https://github.com/mysql/mysql-shell-plugins

Head on over to http://mysqlcommunity.slack.com to join us!

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.

ORACLE