Faster Application Development for MySQL

Mike Frank, MySQL Product Management Director | **Oracle**

Streamlining

Software development can be complicated

Developers strive to deliver high-quality products **faster and more efficiently**

Streamlining for Creating, Designing and Testing software requires

- Simplification
- Automation
- Optimization

In this talk we will present new techniques for ”Streamlining” AppDev with MySQL”

Streamlining development for MySQL

1 - Integration with VS Code for MySQL Devs and DBAs

SQL IDE for MySQL

MySQL Shell Extension for VS Code

Support SQL, JavaScript, Python

Admin APIs

Connections

MySQL HeatWave and OCI Integration

2 - Leverage Progressive Web Apps using REST

MySQL Restful Service within the MySQL Router

Automate using create from a schema

Integration to develop to RestAPIs within VS Code

3 - Supporting JavaScript dev inside the MySQL server

Write MySQL Stored Procedures

Develop, debug and and deploy within VS Code

4 - MySQL Kubernetes Operator

Automates

- Deployment
- Configuration
- Availability

Consistent environment for development, testing, and production

Allows developers to focus on code

Overview

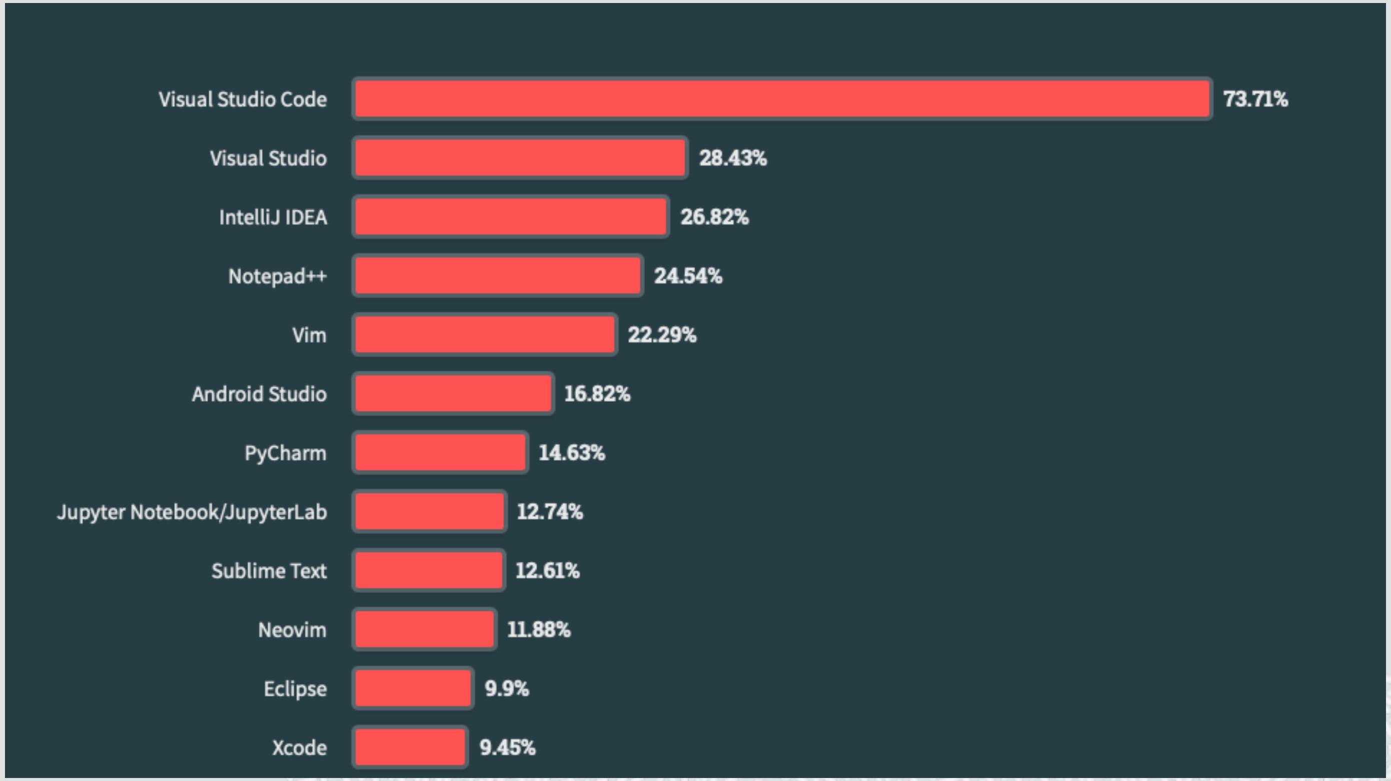
1 - Integration with VS Code for MySQL Devs and DBAs

VS Code IDE

- Extensions
- Multi-language
- Multi-platform
- Source Code Control Integration

With MySQL Shell Extension

- Schema Navigation
- SQL Editor
- SQL Worksheets
- Data/Results Grids
- OCI integration



2 - Leverage Progressive Web Apps using REST

Advantages

- PWAs are platform agnostic – same code across multiple platforms
- Leverages Web Dev Skills
- Faster Development
- Discoverable
- Security Standards supported
- Easy to update

“Progressive Web Apps are just web applications. Using progressive enhancement, new capabilities are enabled in modern browsers. Using service workers and a web app manifest, your web application becomes reliable and installable. If the new capabilities aren't available, users still get the core experience”

https://web.dev/articles/what-are-pwas

3 - Supporting JavaScript dev inside the MySQL server

Execute JavaScript Stored Programs and Stored Functions via GraalVM

Just like SQL Stored Programs, but now with
- Improved Developer Experience
- Security at its core
- State-of-the-art optimizations
- Designed for both Cloud Service and on-premise

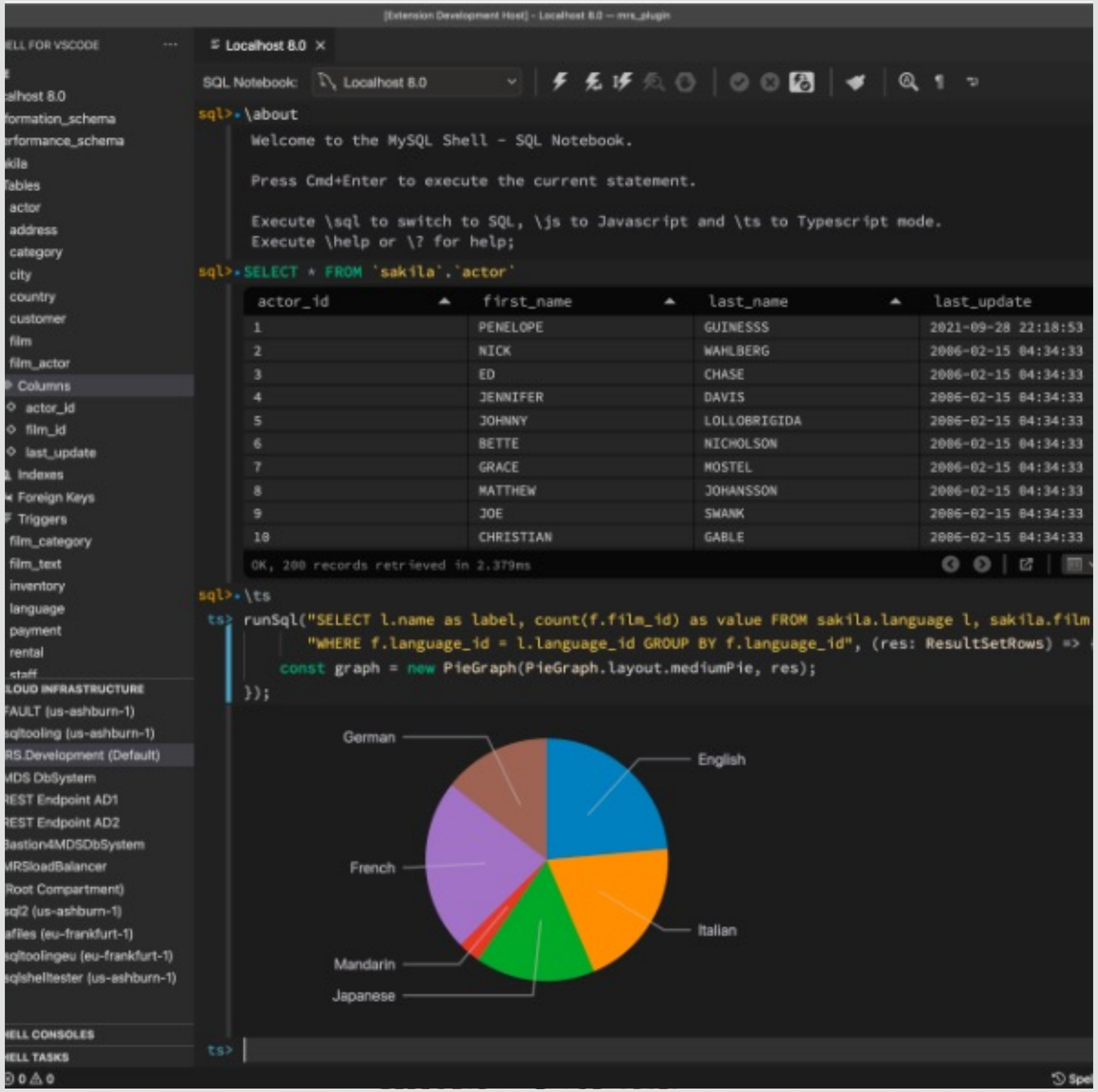
Available Now!
- MySQL Heatwave Database Service for OCI, AWS and Azure
- Preview of MySQL Enterprise Edition on Oracle Technology Network (OTN).

4 - MySQL Kubernetes Operator

Orgs have	Many machines
	Many workloads
Need to manage those efficiently	So developers can just focus on high value appdev
Need to hide complexity	Developers just want MySQL servers accessible via some host and port (IP/port)
Kubernetes solution	A portable, extensible platform for managing containerized workloads and services
	Facilitates both declarative configuration and automation

Details

1 - Integration with VS Code for MySQL Devs and DBAs



```
sql> \about
     Welcome to the MySQL Shell - SQL Notebook.

     Press Cmd+Enter to execute the current statement.

     Execute \sql to switch to SQL, \js to Javascript and \ts to Typescript mode.
     Execute \help or \? for help;
sql> SELECT * FROM `sakila`.`actor`
```

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESSS	2021-09-28 22:18:53
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33

```
OK, 200 records retrieved in 2.379ms
sql> \ts
ts> runSql("SELECT l.name as label, count(f.film_id) as value FROM sakila.language l, sakila.film
        "WHERE f.language_id = l.language_id GROUP BY f.language_id", (res: ResultSetRows) =>
      const graph = new PieGraph(PieGraph.layout.mediumPie, res);
    });
ts>
```

VS Code for MySQL Shell

Next generation UI/Dev platform

VS Code Extension - MySQL Shell

Successor to MySQL Workbench

IDE for MySQL DBAs and Developers

Key Features

- SQL Editor
- SQL Worksheets
- Schema Navigation
- Data/Results Grids
- JavaScript and TypeScript Editors
- Query, manipulate, and visualize your data.
- Integrates seamlessly into your development workflow.
- OCI MySQL Heatwave Service Integration

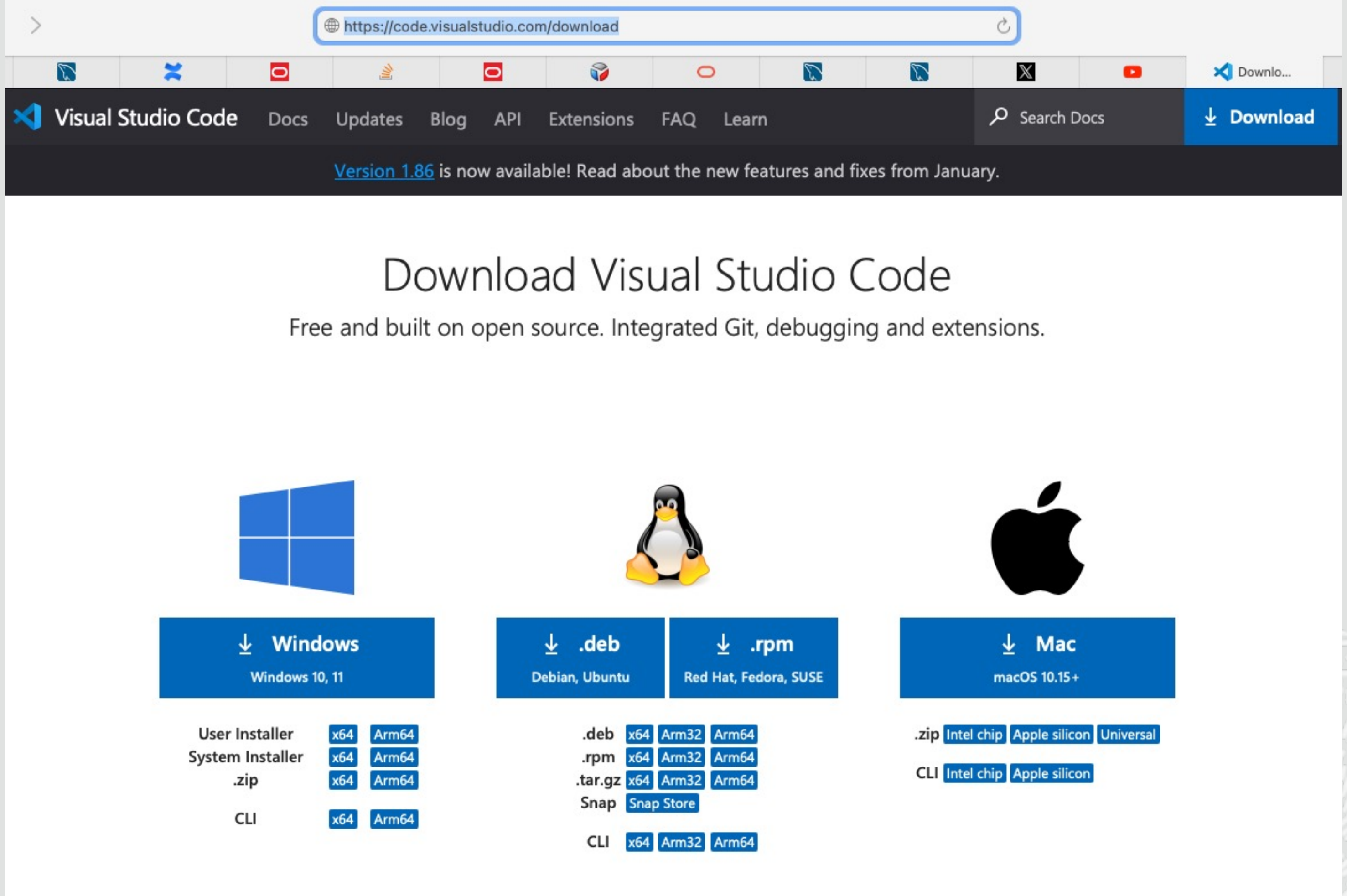
Install VS Code

Download

Run installer

Start VS Code

https://code.visualstudio.com/download



Version 1.86 is now available! Read about the new features and fixes from January.

Download Visual Studio Code

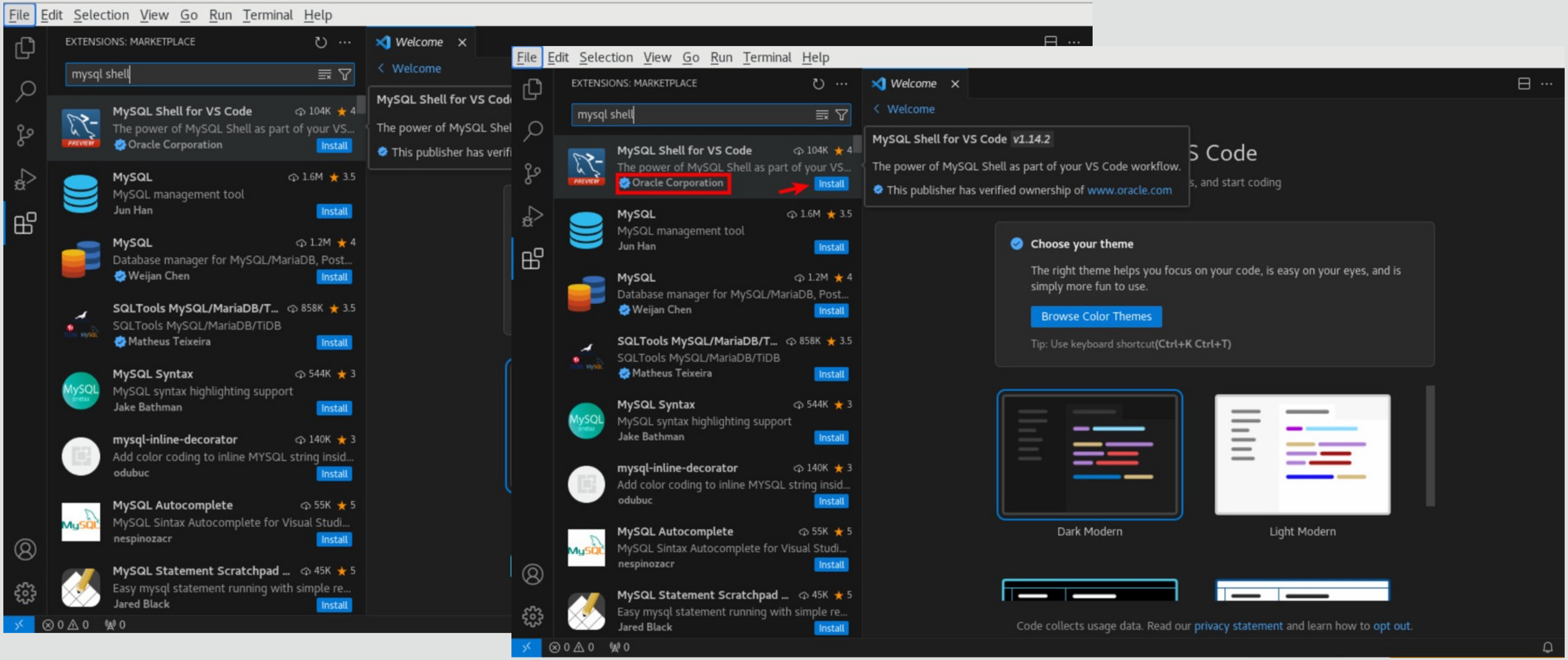
Free and built on open source. Integrated Git, debugging and extensions.

User Installer x64 Arm64
System Installer x64 Arm64
.zip x64 Arm64
CLI x64 Arm64

.deb x64 Arm32 Arm64
.rpm x64 Arm32 Arm64
.tar.gz x64 Arm32 Arm64
Snap Snap Store
CLI x64 Arm32 Arm64

.zip Intel chip Apple silicon Universal
CLI Intel chip Apple silicon

Add the MySQL Shell Extension



mysql shell

MySQL Shell for VS Code 104K ★ 4
The power of MySQL Shell as part of your VS...
Oracle Corporation

MySQL 1.6M ★ 3.5
MySQL management tool
Jun Han

MySQL 1.2M ★ 4
Database manager for MySQL/MariaDB, Post...
Weijan Chen

SQLTools MySQL/MariaDB/T... 858K ★ 3.5
SQLTools MySQL/MariaDB/TiDB
Matheus Teixeira

MySQL Syntax 544K ★ 3
MySQL syntax highlighting support
Jake Bathman

mysql-inline-decorator 140K ★ 3
Add color coding to inline MYSQL string insid...
odubuc

MySQL Autocomplete 55K ★ 5
MySQL Sintax Autocomplete for Visual Studi...
nespinozacr

MySQL Statement Scratchpad ... 45K ★ 5
Easy mysql statement running with simple re...
Jared Black

MySQL Shell for VS Code v1.14.2
The power of MySQL Shell as part of your VS Code workflow.
This publisher has verified ownership of www.oracle.com

S Code
s, and start coding

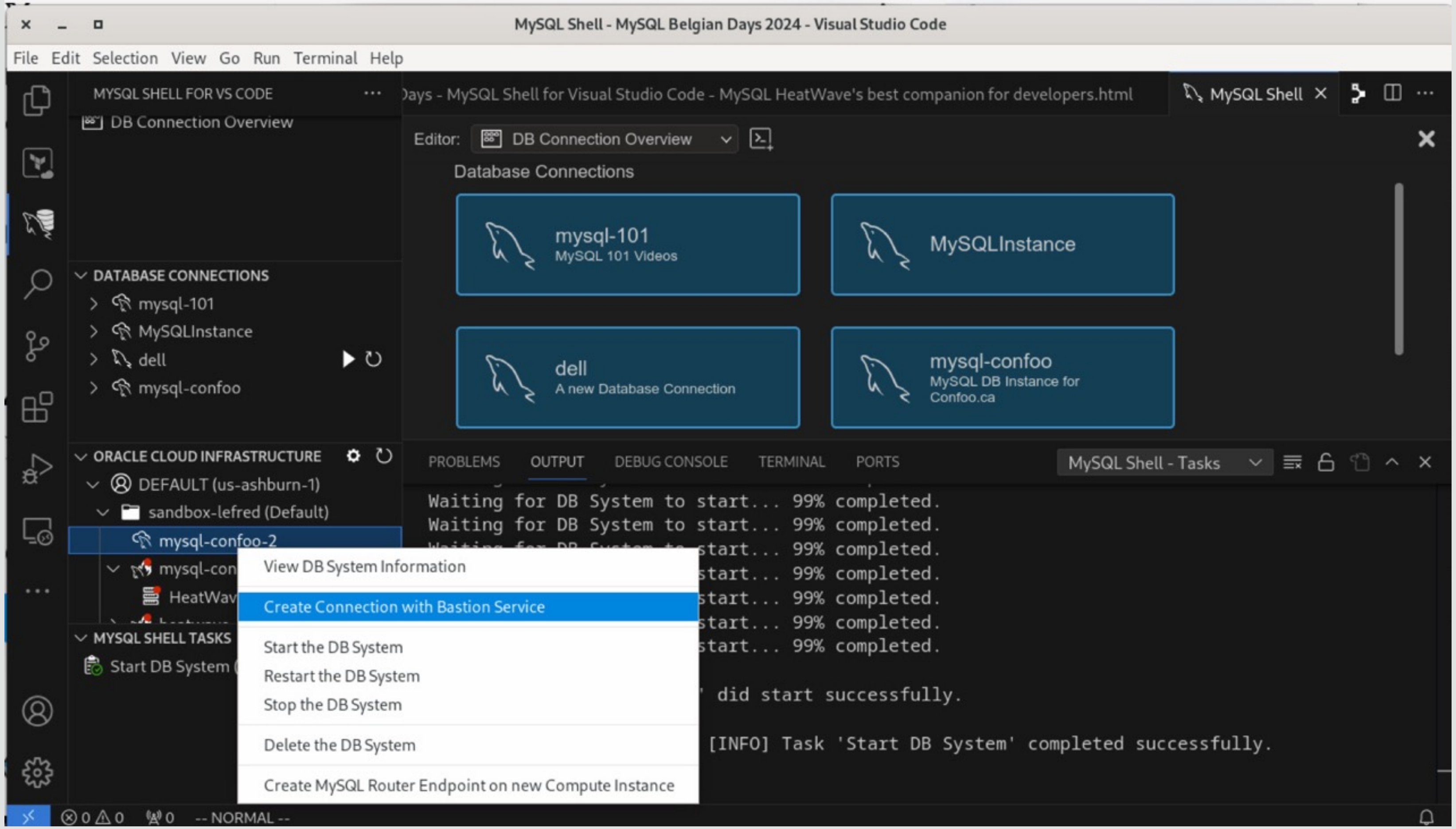
Choose your theme
The right theme helps you focus on your code, is easy on your eyes, and is simply more fun to use.
Browse Color Themes
Tip: Use keyboard shortcut(Ctrl+K Ctrl+T)

Dark Modern
Light Modern

Code collects usage data. Read our privacy statement and learn how to opt out.

Connect - OCI MySQL HeatWave Connection

As Developer, the easiest way to connect to your MySQL DB System if you don't have any VPN to your OCI tenancy, is to use the Bastion Service:

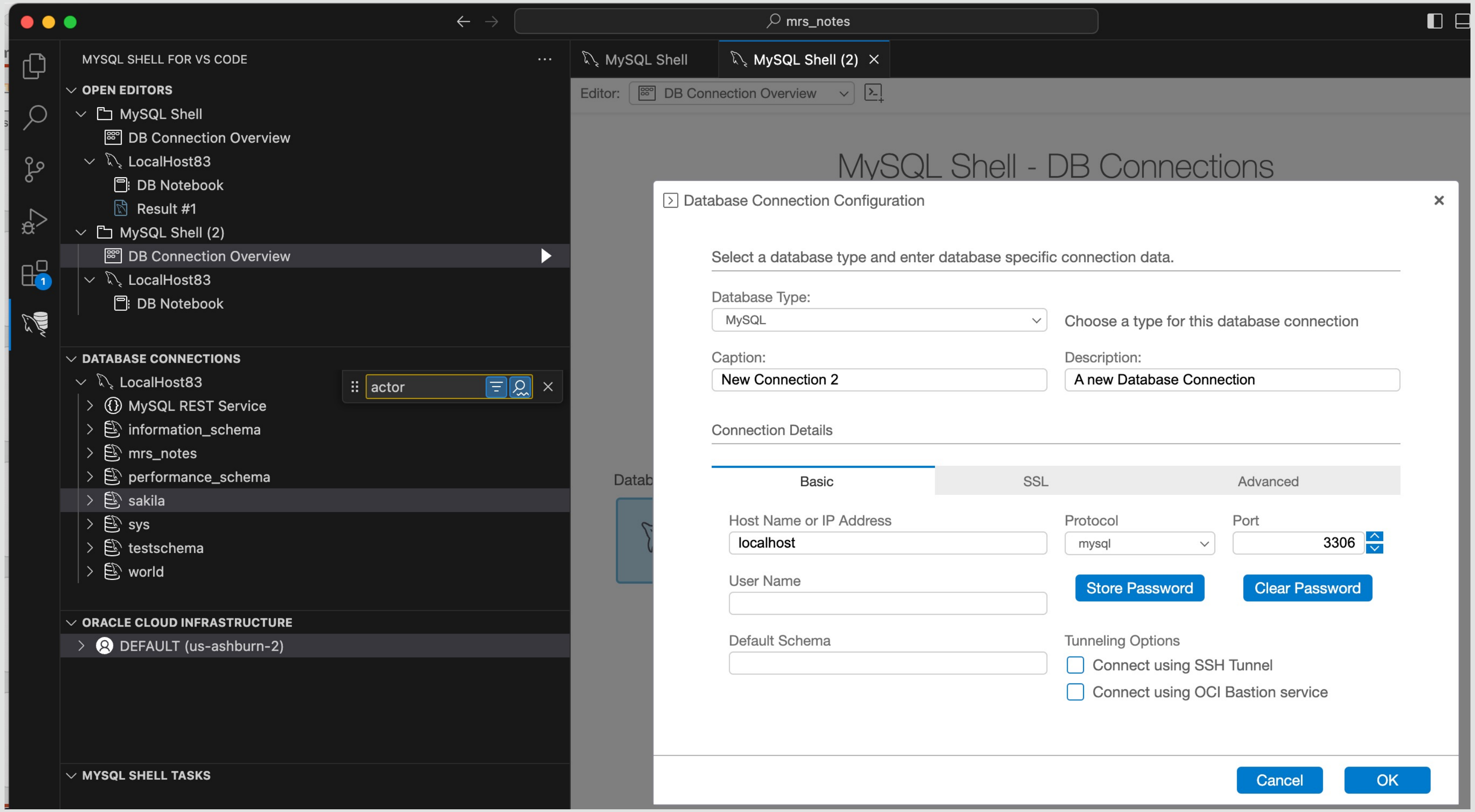


```
Waiting for DB System to start... 99% completed.
Waiting for DB System to start... 99% completed.
Waiting for DB System to start... 99% completed.
                          start... 99% completed.
                          start... 99% completed.
                          start... 99% completed.
                          start... 99% completed.
                          start... 99% completed.

                         ' did start successfully.

                         [INFO] Task 'Start DB System' completed successfully.
```

The right tools for AppDevs and DBAs

Connection Configuration Wizard

- Easily create, test, and save connections
- Store passwords securely to key vaults/rings
- Supports SSH tunneling and OCI Bastion service



Database Connection Configuration

Select a database type and enter database specific connection data.

Database Type: MySQL — Choose a type for this database connection

Caption: New Connection 2

Description: A new Database Connection

Connection Details

Basic | SSL | Advanced

Host Name or IP Address: localhost

Protocol: mysql

Port: 3306

User Name:

Store Password | Clear Password

Default Schema:

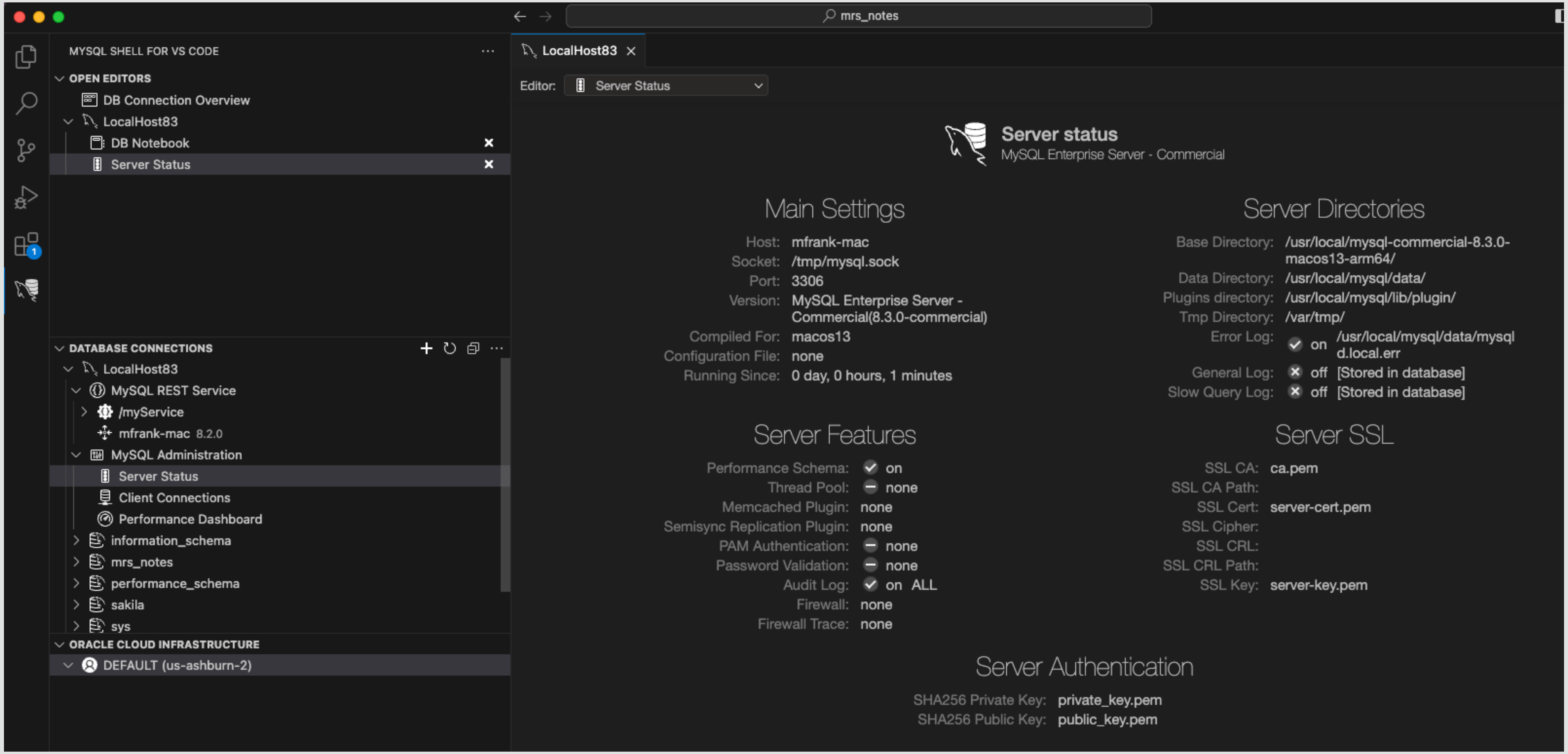
Tunneling Options

□ Connect using SSH Tunnel

□ Connect using OCI Bastion service

Cancel | OK

MySQL Server Status



Server status
MySQL Enterprise Server - Commercial

Main Settings

Host: mfrank-mac
Socket: /tmp/mysql.sock
Port: 3306
Version: MySQL Enterprise Server - Commercial(8.3.0-commercial)
Compiled For: macos13
Configuration File: none
Running Since: 0 day, 0 hours, 1 minutes

Server Directories

Base Directory: /usr/local/mysql-commercial-8.3.0-macos13-arm64/
Data Directory: /usr/local/mysql/data/
Plugins directory: /usr/local/mysql/lib/plugin/
Tmp Directory: /var/tmp/
Error Log: ✓ on /usr/local/mysql/data/mysqld.local.err
General Log: ✗ off [Stored in database]
Slow Query Log: ✗ off [Stored in database]

Server Features

Performance Schema: ✓ on
Thread Pool: ⊖ none
Memcached Plugin: none
Semisync Replication Plugin: none
PAM Authentication: ⊖ none
Password Validation: ⊖ none
Audit Log: ✓ on ALL
Firewall: none
Firewall Trace: none

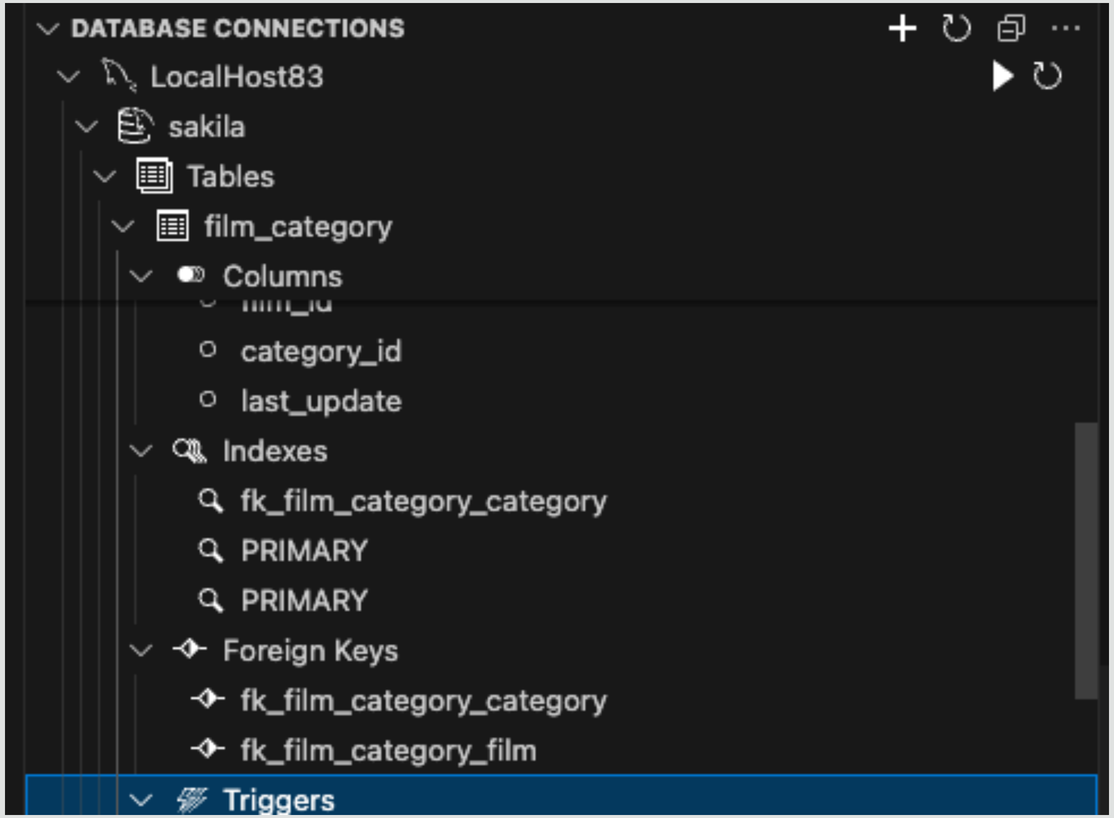
Server SSL

SSL CA: ca.pem
SSL CA Path:
SSL Cert: server-cert.pem
SSL Cipher:
SSL CRL:
SSL CRL Path:
SSL Key: server-key.pem

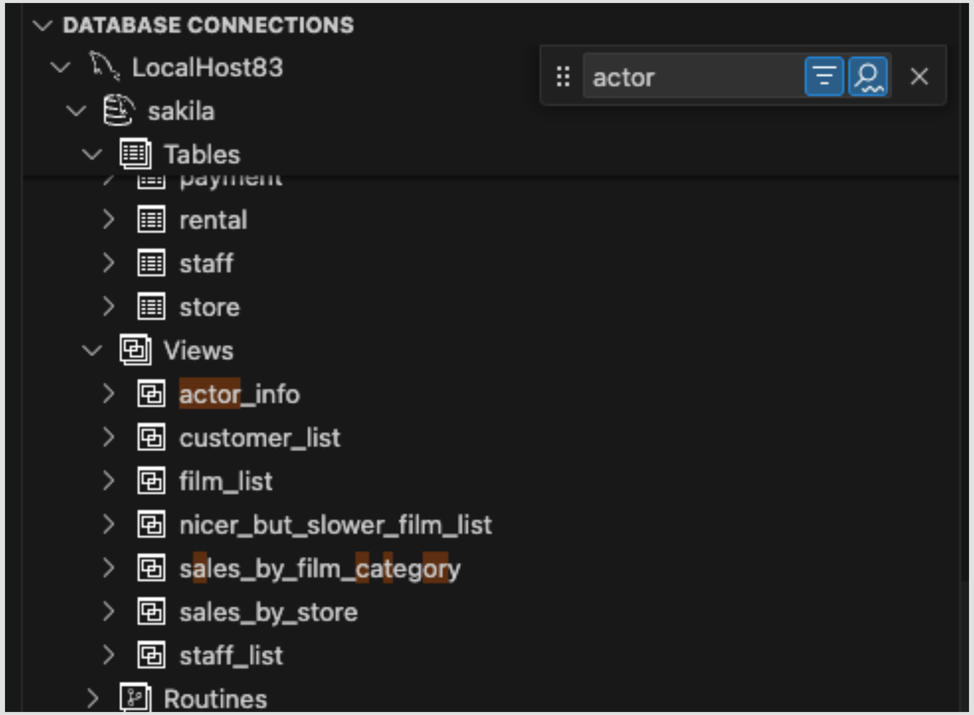
Server Authentication

SHA256 Private Key: private_key.pem
SHA256 Public Key: public_key.pem

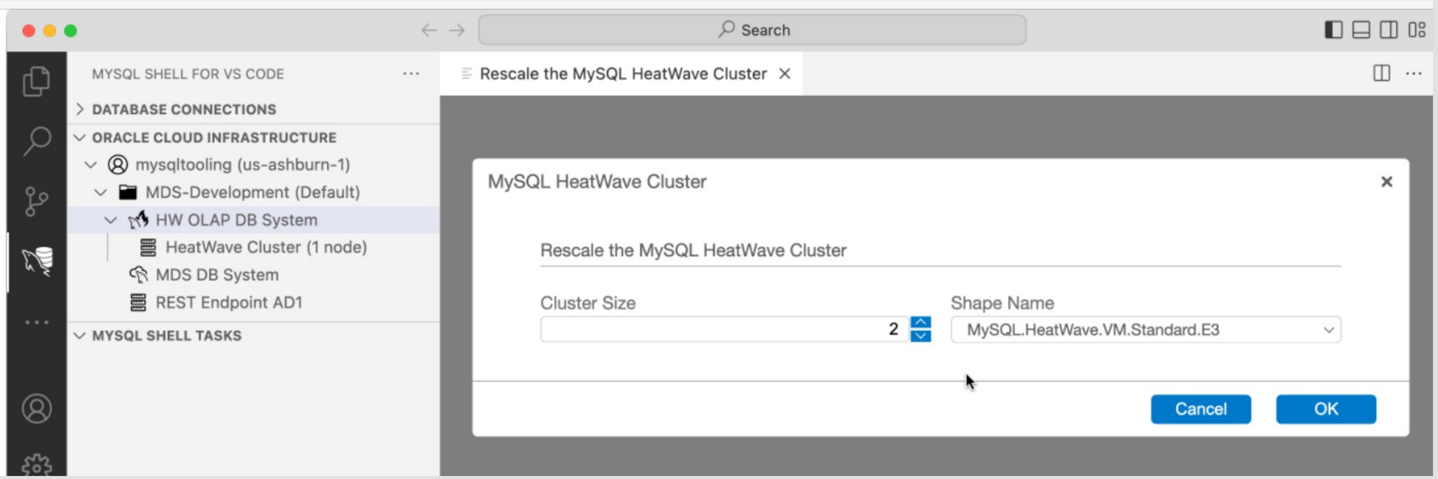
Database Object Browser



Quick Find



OCI HeatWave Integrated



Search

MYSQL SHELL FOR VS CODE

DATABASE CONNECTIONS

ORACLE CLOUD INFRASTRUCTURE

- mysqltooling (us-ashburn-1)
  - MDS-Development (Default)
    - HW OLAP DB System
      - HeatWave Cluster (1 node)
    - MDS DB System
    - REST Endpoint AD1

MYSQL SHELL TASKS

Rescale the MySQL HeatWave Cluster

MySQL HeatWave Cluster

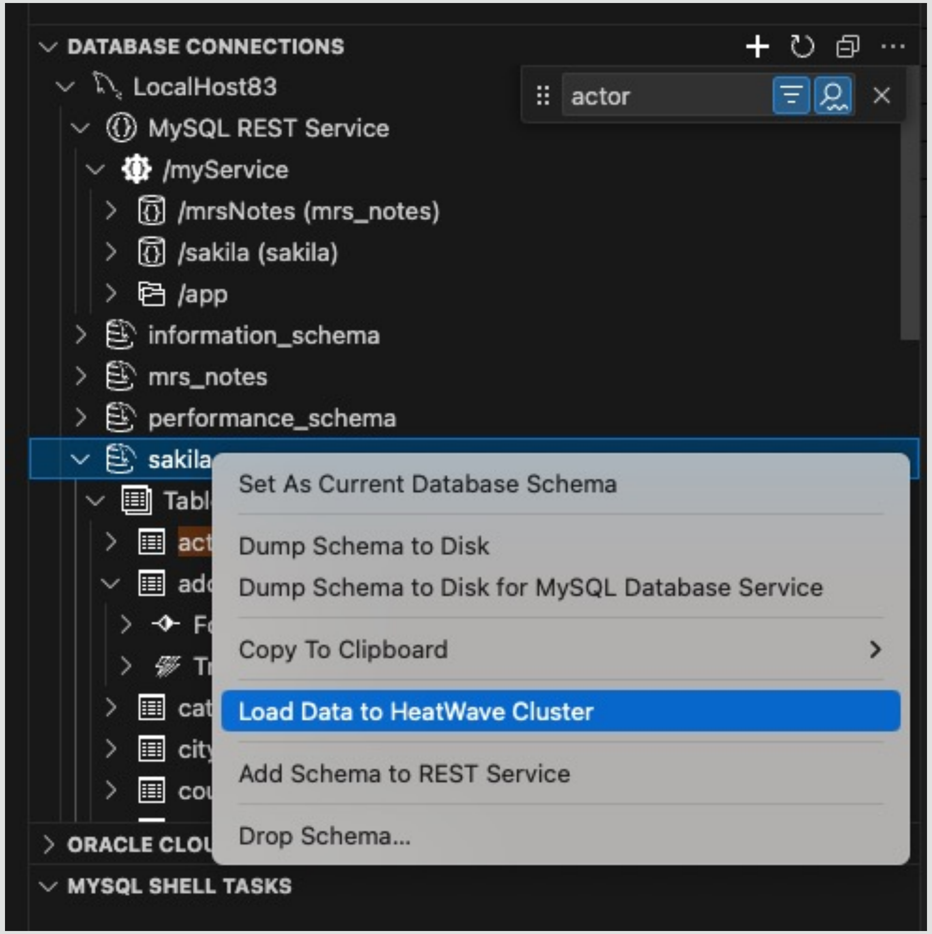
Rescale the MySQL HeatWave Cluster

Cluster Size: 2

Shape Name: MySQL.HeatWave.VM.Standard.E3

Cancel OK

Migrate to HeatWave



DATABASE CONNECTIONS

actor

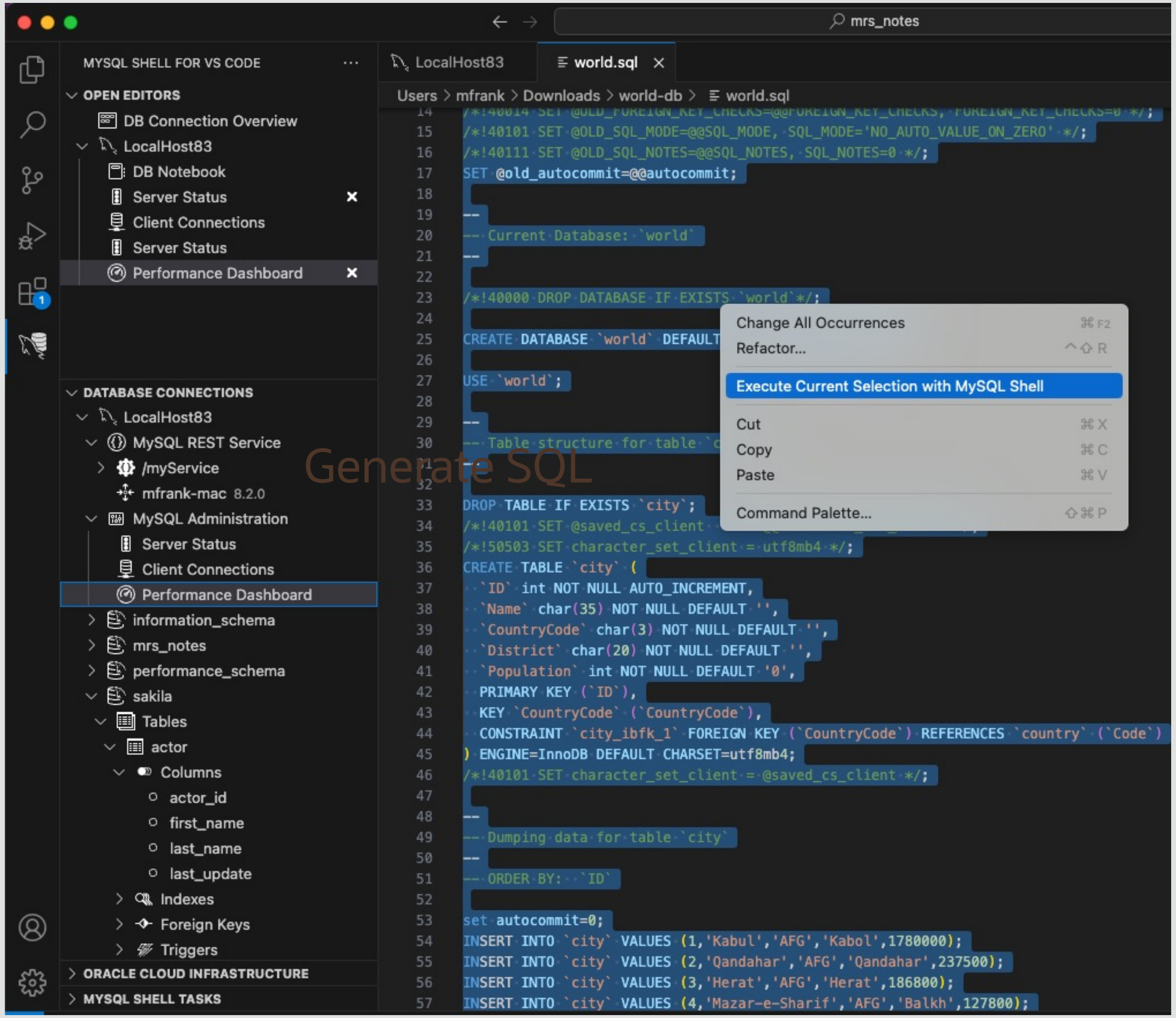
- LocalHost83
  - MySQL REST Service
    - /myService
      - /mrsNotes (mrs_notes)
      - /sakila (sakila)
      - /app
  - information_schema
  - mrs_notes
  - performance_schema
  - sakila
    - Tabl
      - act
      - adc
        - Fc
        - Tr
      - cat
      - city
      - cou

ORACLE CLOU

MYSQL SHELL TASKS

Set As Current Database Schema
Dump Schema to Disk
Dump Schema to Disk for MySQL Database Service
Copy To Clipboard
Load Data to HeatWave Cluster
Add Schema to REST Service
Drop Schema...

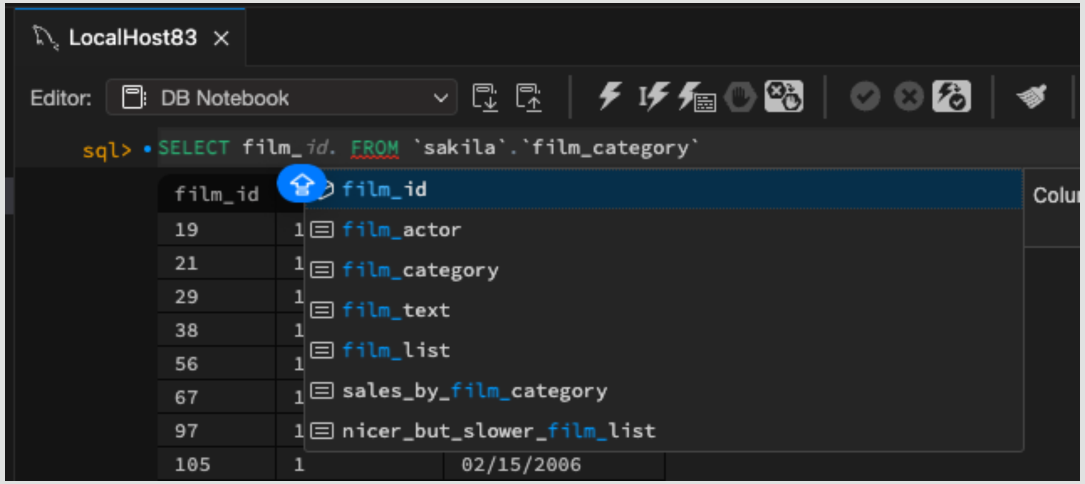
Run scripts



```
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
SET @old_autocommit=@@autocommit;

--
-- Current Database: `world`
--

/*!40000 DROP DATABASE IF EXISTS `world`*/;

CREATE DATABASE `world` DEFAULT

USE `world`;

--
-- Table structure for table `c
--

DROP TABLE IF EXISTS `city`;
/*!40101 SET @saved_cs_client
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `city` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `Name` char(35) NOT NULL DEFAULT '',
  `CountryCode` char(3) NOT NULL DEFAULT '',
  `District` char(20) NOT NULL DEFAULT '',
  `Population` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`ID`),
  KEY `CountryCode` (`CountryCode`),
  CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `city`
--
-- ORDER BY:  `ID`

set autocommit=0;
INSERT INTO `city` VALUES (1,'Kabul','AFG','Kabol',1780000);
INSERT INTO `city` VALUES (2,'Qandahar','AFG','Qandahar',237500);
INSERT INTO `city` VALUES (3,'Herat','AFG','Herat',186800);
INSERT INTO `city` VALUES (4,'Mazar-e-Sharif','AFG','Balkh',127800);
```

Generate SQL

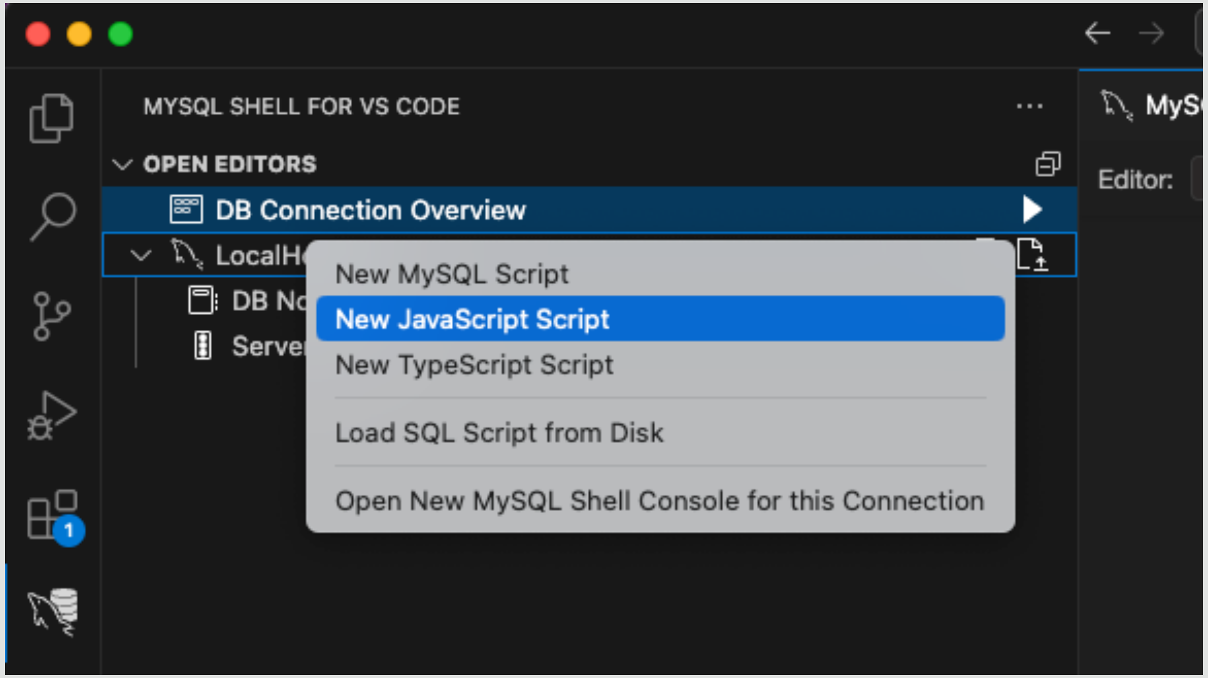
Intellisense



```
sql> SELECT film_id. FROM `sakila`.`film_category`
```

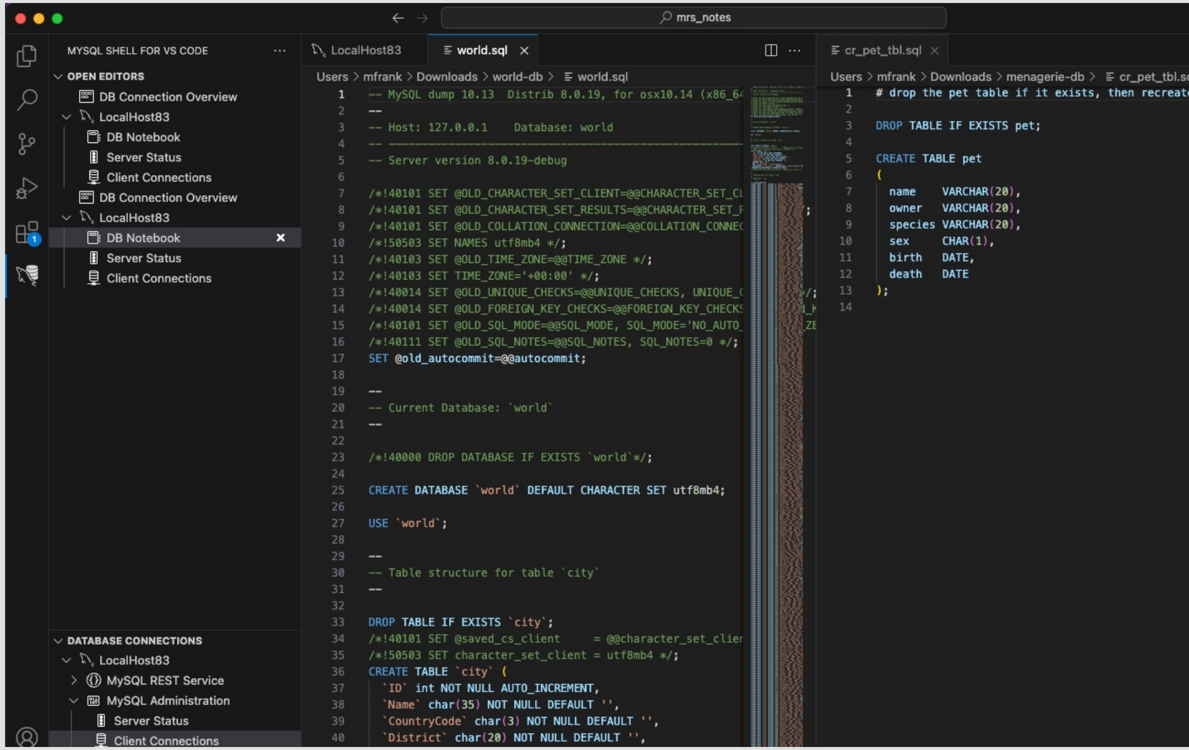
film_id		
19	1	
21	1	
29	1	
38	1	
56	1	
67	1	
97	1	
105	1	02/15/2006

Script
- SQL
- JavaScript
- Typescript
- MySQL Shell



MySQL Notebooks

- Allows devs to integrate code, text, equations, and v
- In a single document known as a “notebook.”



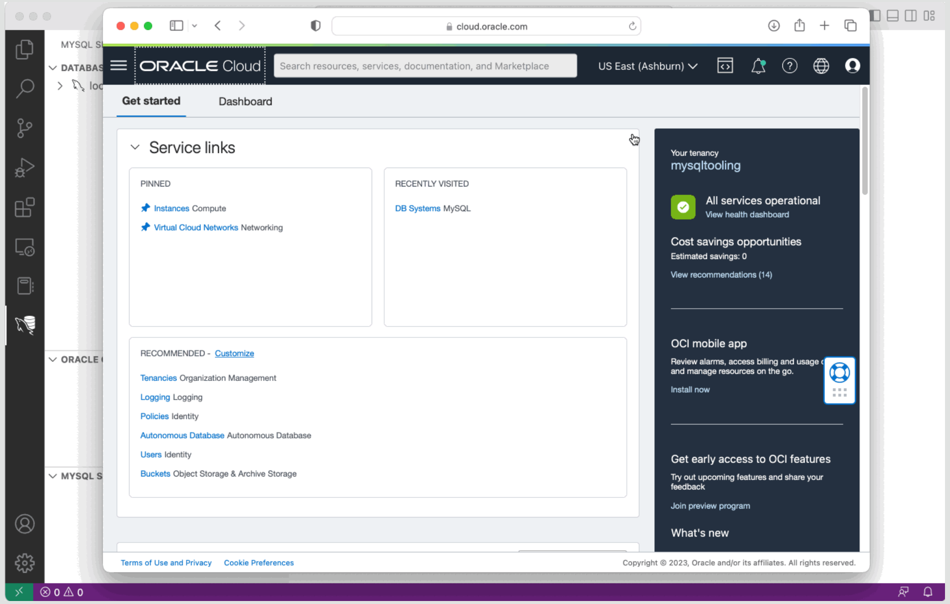
```sql
-- MySQL dump 10.13  Distrib 8.0.19, for osx10.14 (x86_64
--
-- Host: 127.0.0.1    Database: world
-- ------------------------------------------------------
-- Server version 8.0.19-debug

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CL
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_R
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNEC
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_C
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
SET @old_autocommit=@@autocommit;

--
-- Current Database: `world`
--

/*!40000 DROP DATABASE IF EXISTS `world`*/;

CREATE DATABASE `world` DEFAULT CHARACTER SET utf8mb4;

USE `world`;

--
-- Table structure for table `city`
--

DROP TABLE IF EXISTS `city`;
/*!40101 SET @saved_cs_client     = @@character_set_clien
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `city` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `Name` char(35) NOT NULL DEFAULT '',
  `CountryCode` char(3) NOT NULL DEFAULT '',
  `District` char(20) NOT NULL DEFAULT '',
```

```sql
# drop the pet table if it exists, then recreat

DROP TABLE IF EXISTS pet;

CREATE TABLE pet
(
  name    VARCHAR(20),
  owner   VARCHAR(20),
  species VARCHAR(20),
  sex     CHAR(1),
  birth   DATE,
  death   DATE
);

```

VS Code Notebook API allowed the MySQL Visual extension to
- Open files as notebooks,
- Execute notebook code cells
- Render notebook outputs in a variety of rich and interactive formats

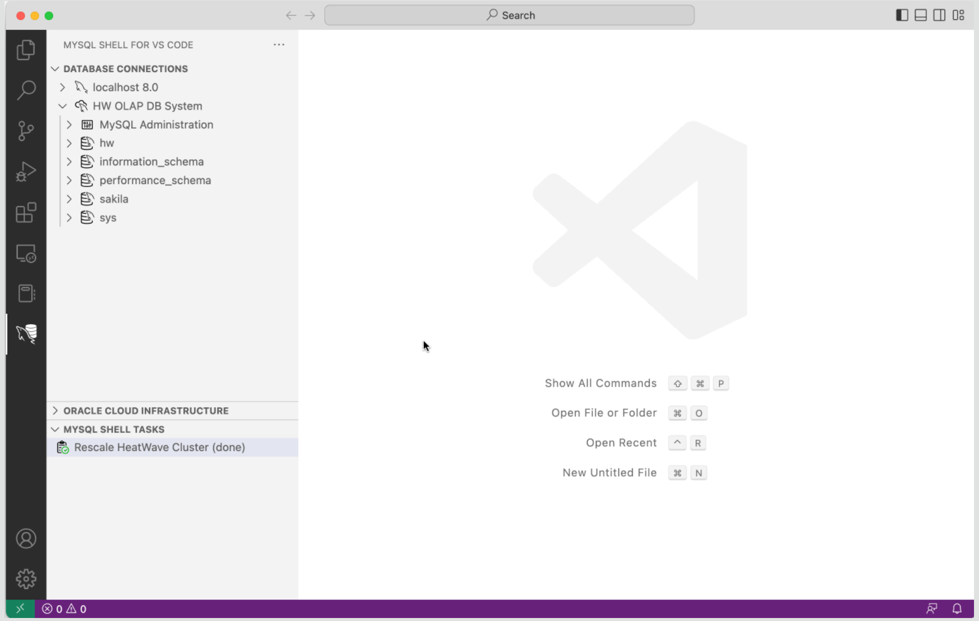
https://code.visualstudio.com/blogs/2021/11/08/custom-notebooks

Creating a DB Connection

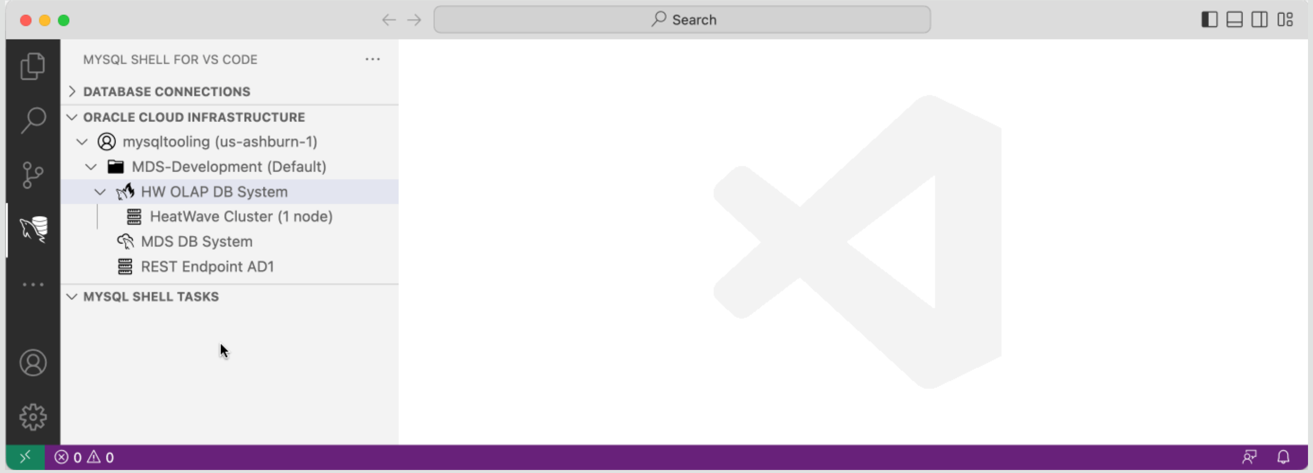


cloud.oracle.com
ORACLE Cloud
Search resources, services, documentation, and Marketplace
US East (Ashburn)
Get started
Dashboard
Service links
PINNED
Instances Compute
Virtual Cloud Networks Networking
RECENTLY VISITED
DB Systems MySQL
RECOMMENDED - Customize
Tenancies Organization Management
Logging Logging
Policies Identity
Autonomous Database Autonomous Database
Users Identity
Buckets Object Storage & Archive Storage
Your tenancy
mysqltooling
All services operational
View health dashboard
Cost savings opportunities
Estimated savings: 0
View recommendations (14)
OCI mobile app
Review alarms, access billing and usage and manage resources on the go.
Install now
Get early access to OCI features
Try out upcoming features and share your feedback
Join preview program
What's new
Terms of Use and Privacy
Cookie Preferences
Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

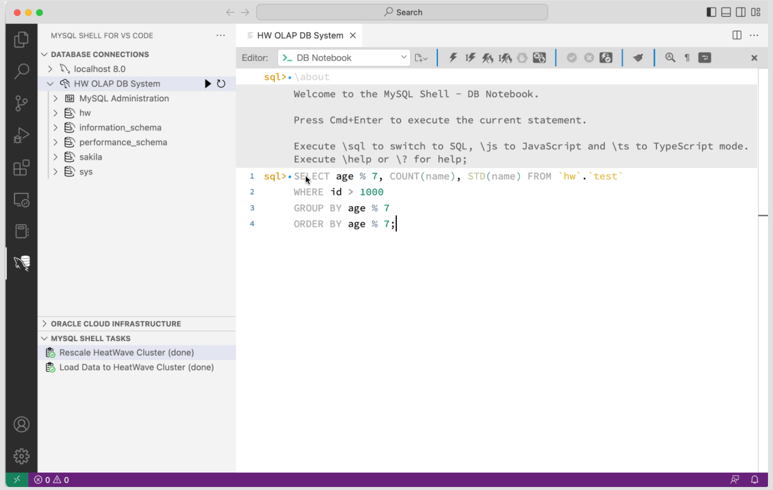
Rescale HA



Load Data



Trace Execution Errors



```
sql> \about
     Welcome to the MySQL Shell - DB Notebook.

     Press Cmd+Enter to execute the current statement.

     Execute \sql to switch to SQL, \js to JavaScript and \ts to TypeScript mode.
     Execute \help or \? for help;
sql> SELECT age % 7, COUNT(name), STD(name) FROM `hw`.`test`
     WHERE id > 1000
     GROUP BY age % 7
     ORDER BY age % 7;
```

2 - Leverage Progressive Web Apps using REST

Rapidly develop PWAs using RESTful Web Services

Restful Web Services

- Automates creation of REST for tables, views, and procedures
- {JSON} responses
- Developer support (GUI, CLI, API)

MySQL Shell for VS Code

- GUI frontend for MySQL Rest Service management
- Interactive documentation
- CLI and scripting support

Built in User Management

- Support for popular OAuth2 services
- User Role, Group & Hierarchy Management
- User Management UI

High Level Architecture

Fast and secure HTTPS access to your MySQL data.

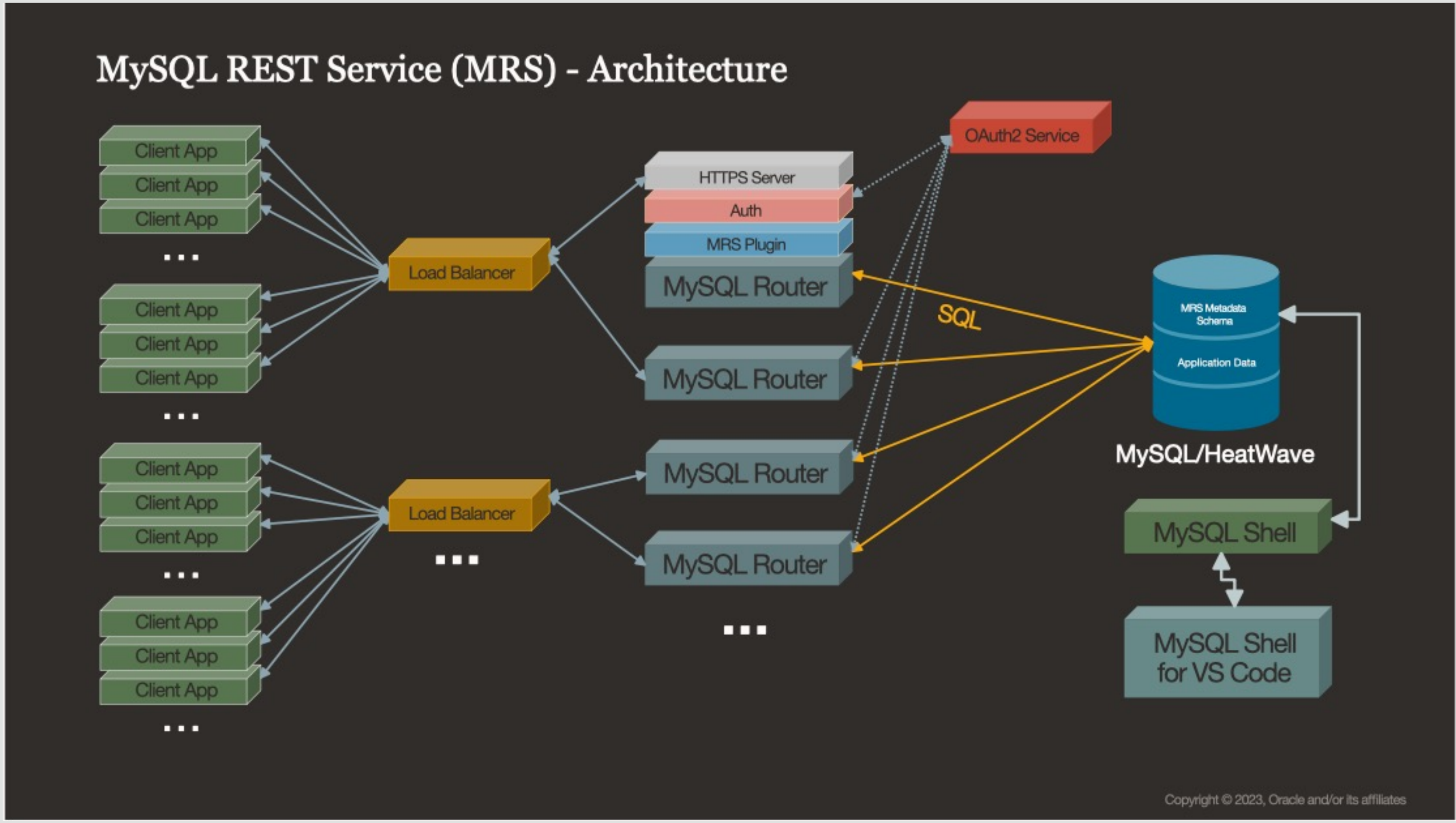
Implemented as a MySQL Router feature

Simple to configuration

Full UI integration with VS Code for MySQL

Ideally suited for Progressive Web Apps

Highly available and scalable



MySQL Restful Service Overview

Works with all popular MySQL Deployment Models

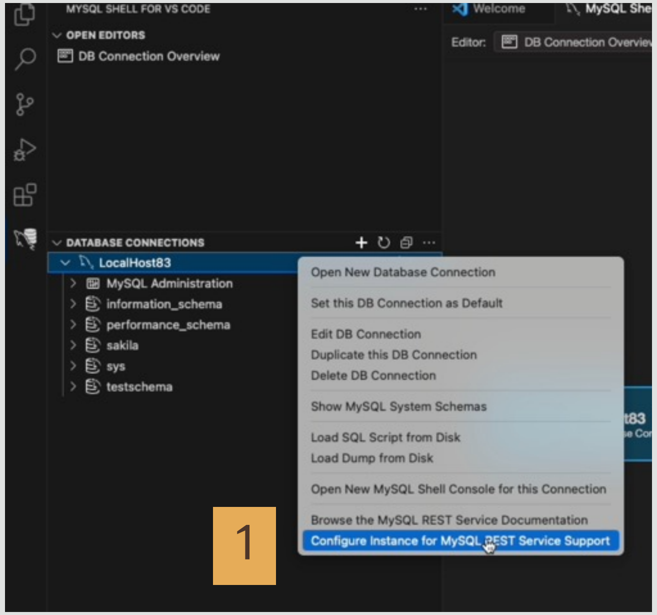
- Cloud - MySQL HeatWave
- On premises - MySQL InnoDB Cluster Set, Replica Set, …

MySQL Router

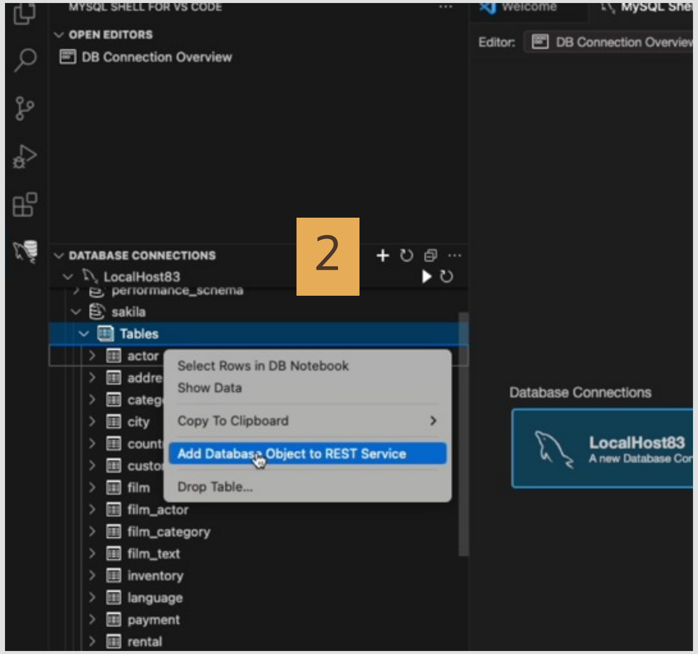
- Serves JSON data via RESTful Web Services

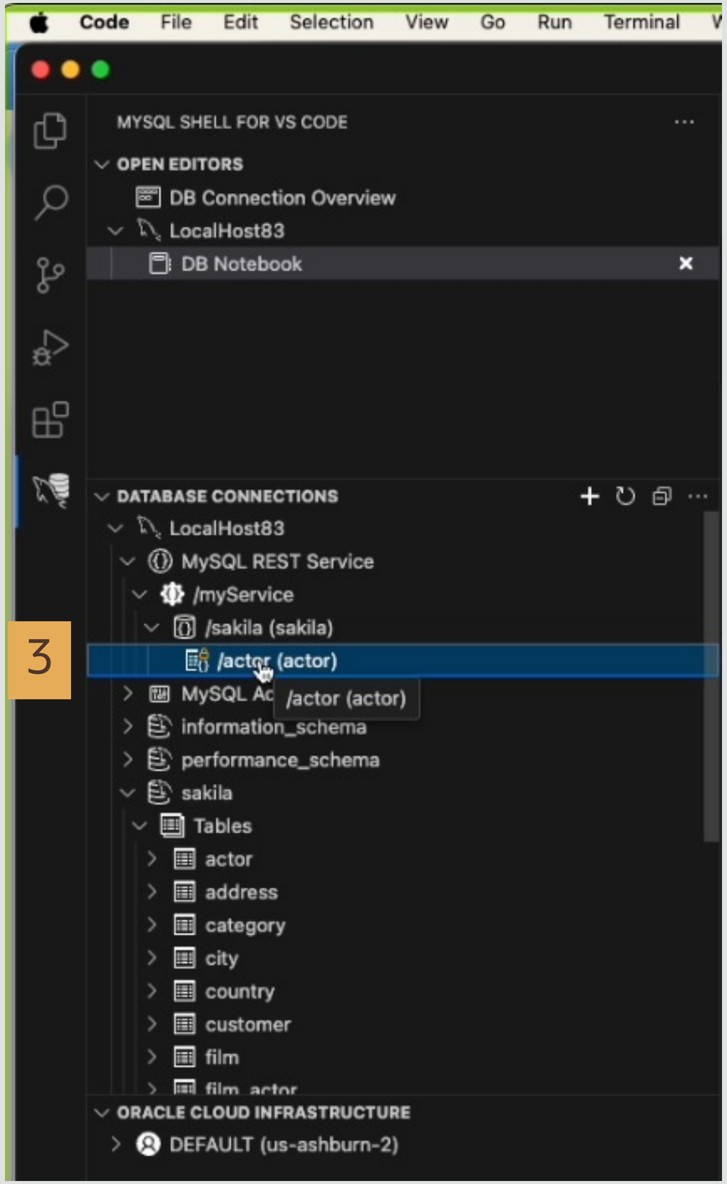
Steps

1. Configure MySQL Instance for Rest
2. Add schema to Rest
3. Add database objects to the service
   - Auto REST for tables, views and procedures
4. Manage Rest Objects

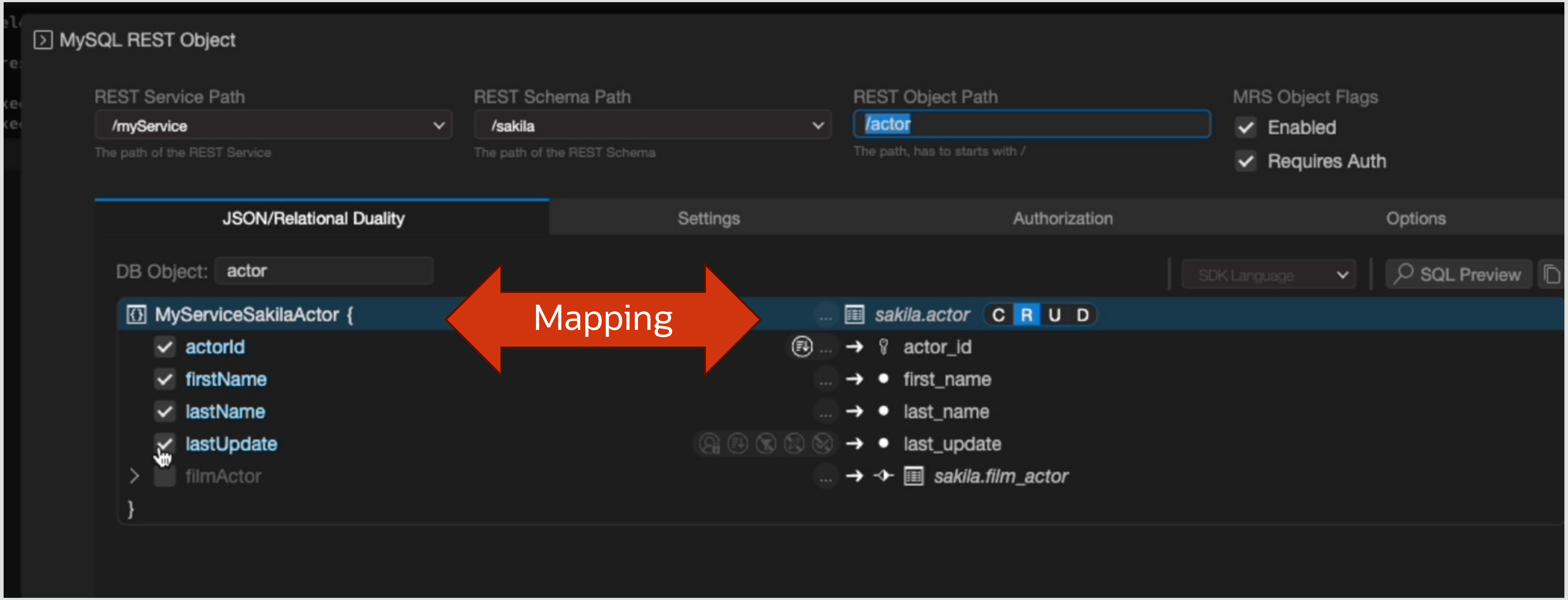


MYSQL SHELL FOR VS CODE
OPEN EDITORS
DB Connection Overview
DATABASE CONNECTIONS
LocalHost83
MySQL Administration
information_schema
performance_schema
sakila
sys
testschema
Open New Database Connection
Set this DB Connection as Default
Edit DB Connection
Duplicate this DB Connection
Delete DB Connection
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
1





Mapping – JSON/Relational Duality Via Rest Service



MySQL REST Object

REST Service Path: /myService
The path of the REST Service

REST Schema Path: /sakila
The path of the REST Schema

REST Object Path: /actor
The path, has to starts with /

MRS Object Flags
☑ Enabled
☑ Requires Auth

DB Object: actor

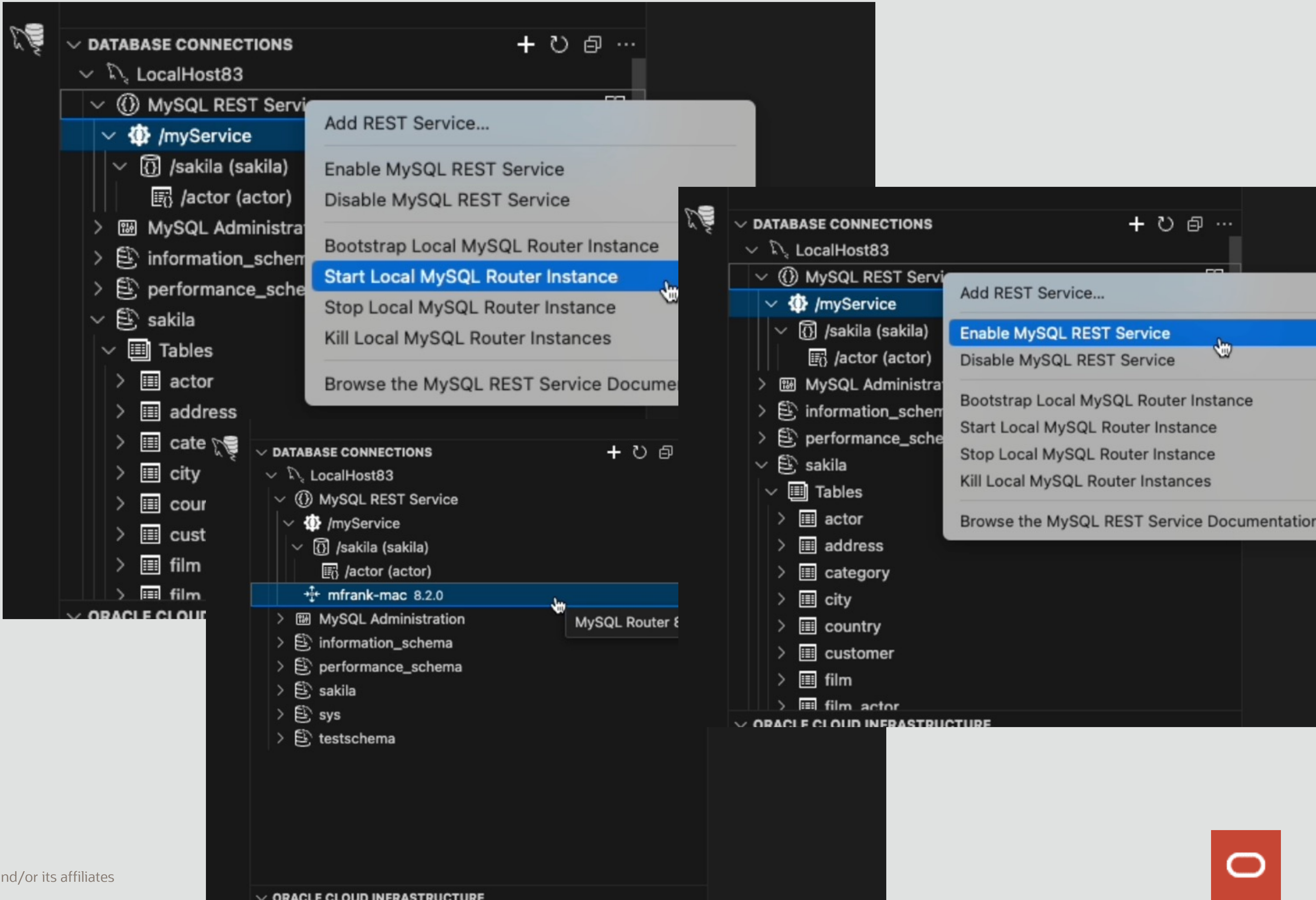
```
MyServiceSakilaActor {                 sakila.actor  C R U D
  ☑ actorId                          → actor_id
  ☑ firstName                        → first_name
  ☑ lastName                         → last_name
  ☑ lastUpdate                       → last_update
> □ filmActor                        → sakila.film_actor
}
```

Mapping

Manage

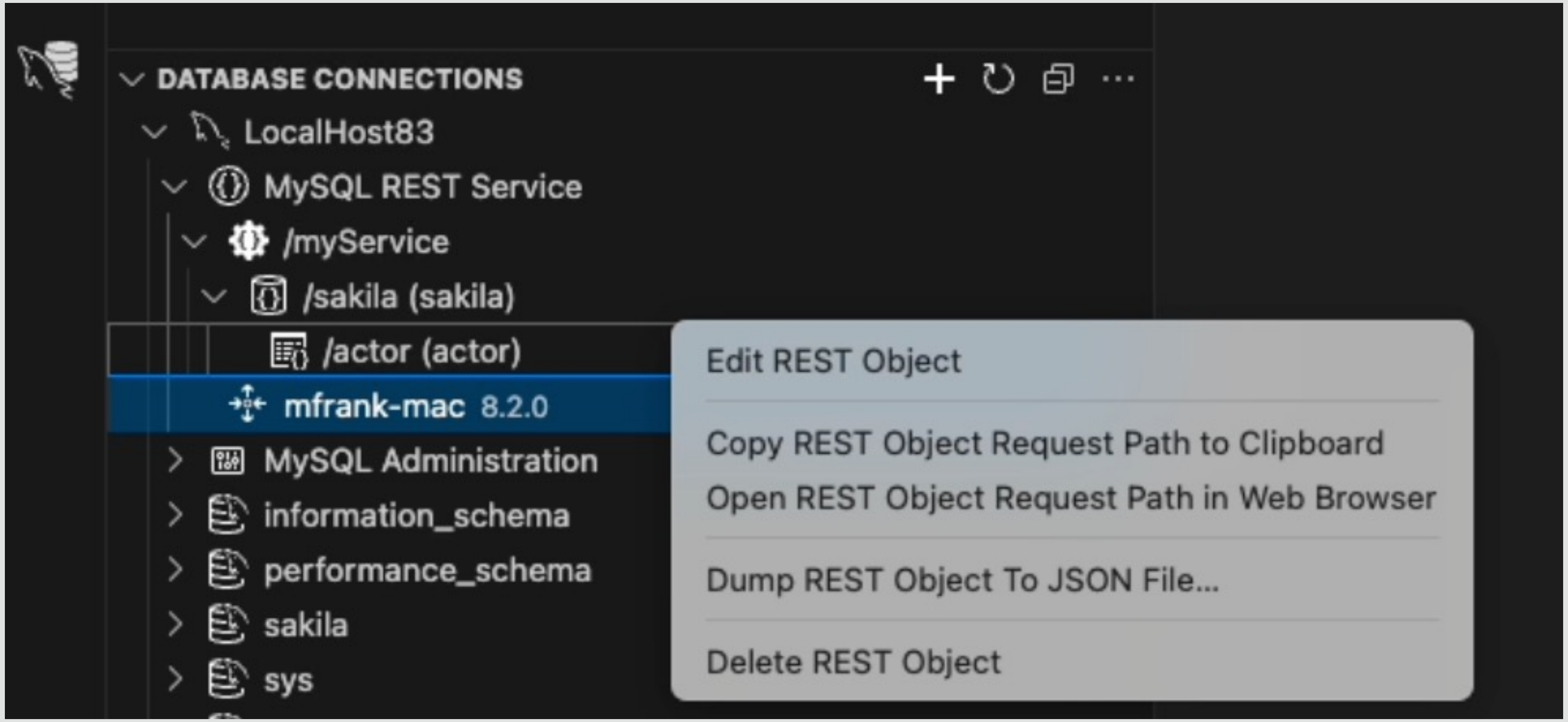
- Start/Stop
- Enable/Disable
- Router Service Management
- Bootstrap
- Configure
- Start
- Stop
- Kill

Integrated documentation

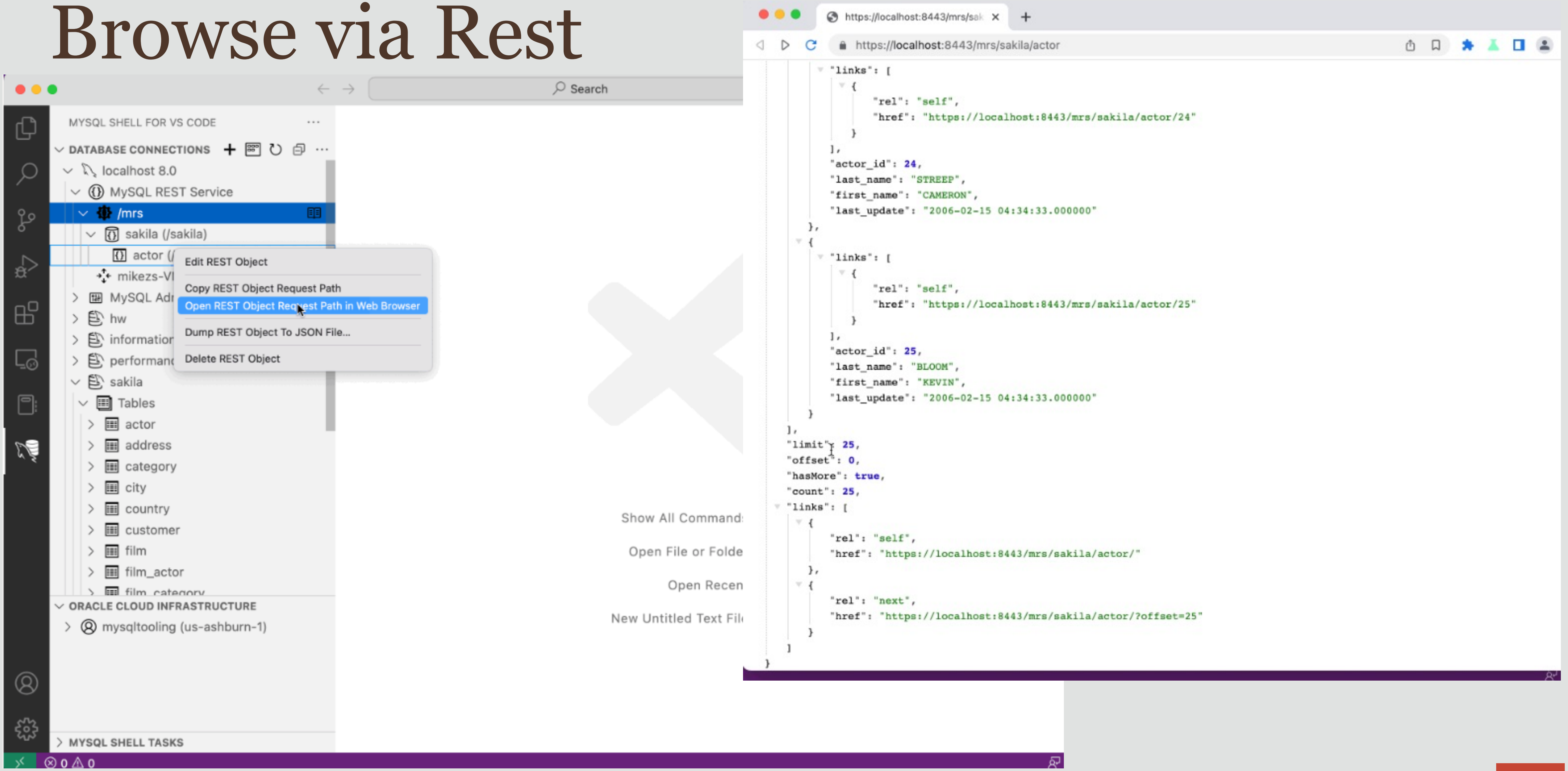


Rest Shortcuts

- Copy
- Open
- Dump
- Remove



Browse via Rest



```
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8443/mrs/sakila/actor/24"
        }
      ],
      "actor_id": 24,
      "last_name": "STREEP",
      "first_name": "CAMERON",
      "last_update": "2006-02-15 04:34:33.000000"
    },
    {
      "links": [
        {
          "rel": "self",
          "href": "https://localhost:8443/mrs/sakila/actor/25"
        }
      ],
      "actor_id": 25,
      "last_name": "BLOOM",
      "first_name": "KEVIN",
      "last_update": "2006-02-15 04:34:33.000000"
    }
  ],
  "limit": 25,
  "offset": 0,
  "hasMore": true,
  "count": 25,
  "links": [
    {
      "rel": "self",
      "href": "https://localhost:8443/mrs/sakila/actor/"
    },
    {
      "rel": "next",
      "href": "https://localhost:8443/mrs/sakila/actor/?offset=25"
    }
  ]
}
```

TypeScript integration

TypeScript

- JavaScript With Syntax For Types
- VS code for MySQL can report errors when the types don't match

Integration allows interactive execution of TypeScript code inside a DB Notebook.

- Makes working with the MySQL REST Service easier
- Is available within the DB Notebooks

RESTful development specific

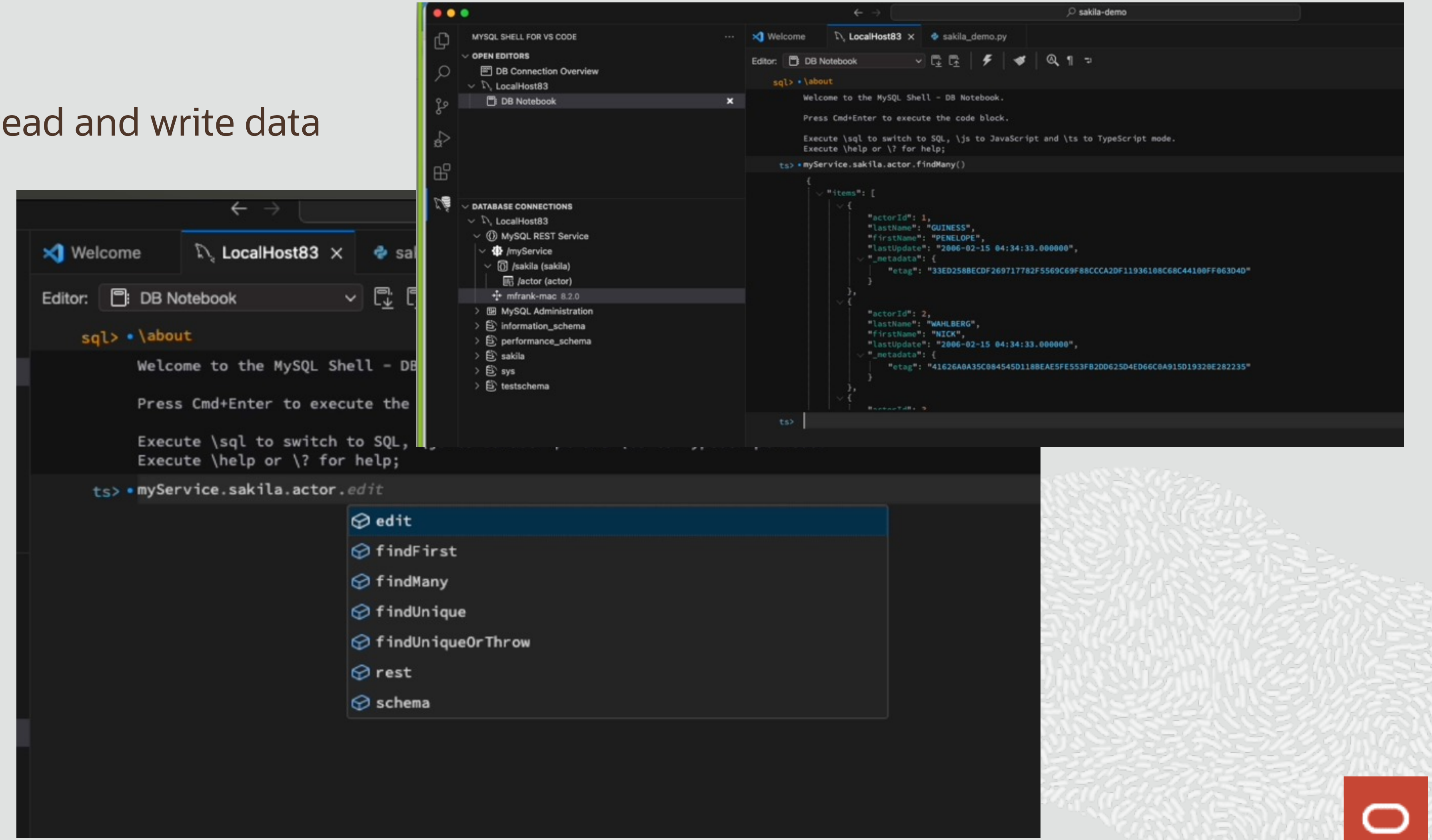
- TypeScript SDK is updated in real time as REST service involves
- Allows instant prototyping of REST queries using the Client API inside VS Code

Data driven application development

Query builder support to read and write data

Loved by developers

- Intuitive
- Automated
- Type safe
- Autocomplete
- View results

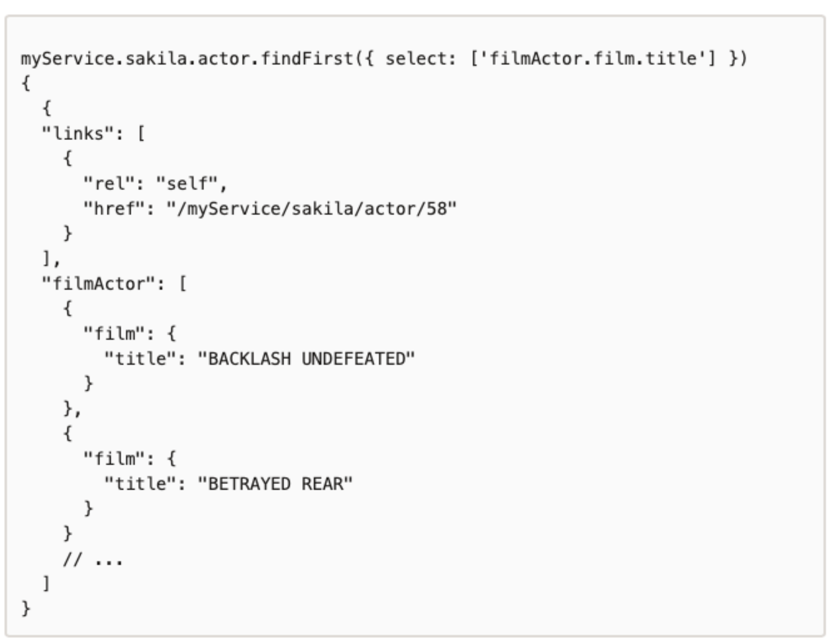


```
sql> \about
     Welcome to the MySQL Shell - DB Notebook.

     Press Cmd+Enter to execute the code block.

     Execute \sql to switch to SQL, \js to JavaScript and \ts to TypeScript mode.
     Execute \help or \? for help;
ts> myService.sakila.actor.findMany()
     {
       "items": [
         {
           "actorId": 1,
           "lastName": "GUINESS",
           "firstName": "PENELOPE",
           "lastUpdate": "2006-02-15 04:34:33.000000",
           "_metadata": {
             "etag": "33ED258BECDF269717782F5569C69F88CCCA2DF11936108C68C44100FF063D4D"
           }
         },
         {
           "actorId": 2,
           "lastName": "WAHLBERG",
           "firstName": "NICK",
           "lastUpdate": "2006-02-15 04:34:33.000000",
           "_metadata": {
             "etag": "41626A0A35C084545D118BEAE5FE553FB2DD625D4ED66C0A915D19320E282235"
           }
         },
         {
ts>
```

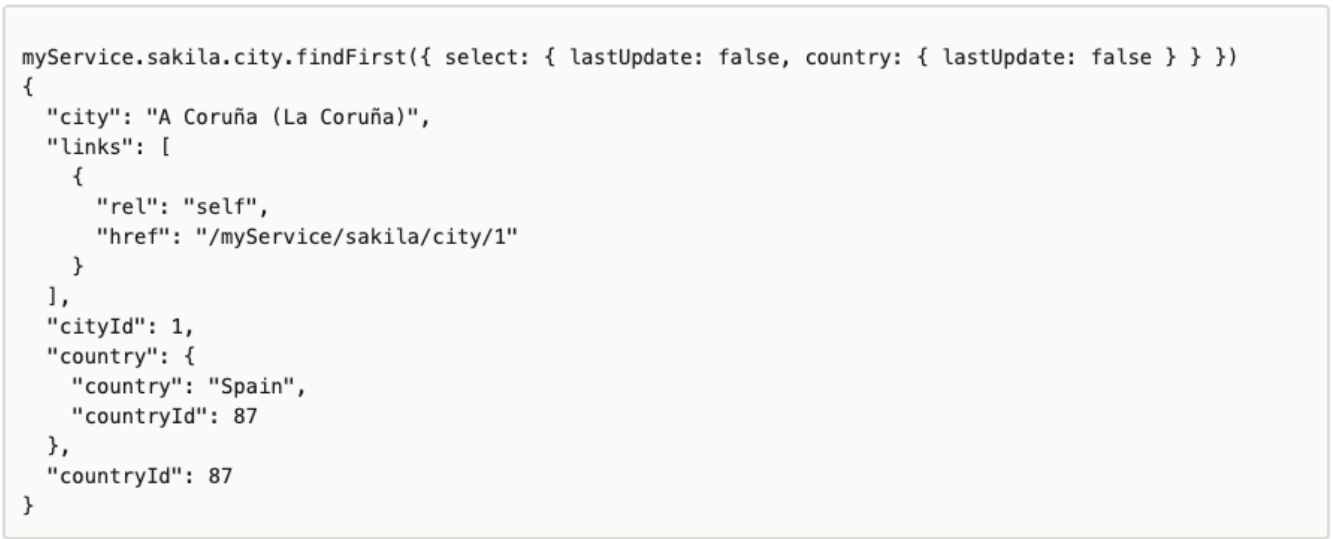
```
sql> \about
     Welcome to the MySQL Shell - DB

     Press Cmd+Enter to execute the

     Execute \sql to switch to SQL,
     Execute \help or \? for help;
ts> myService.sakila.actor.edit
                               edit
                               findFirst
                               findMany
                               findUnique
                               findUniqueOrThrow
                               rest
                               schema
```

REST APIs for

- `create`
- `createMany`
- `findFirst`
- `findUnique`
- `findUniqueOrThrow`
- `findMany`
- `delete`
- `deleteMany`
- `update`
- `updateMany`



```
myService.sakila.actor.findFirst({ select: ['filmActor.film.title'] })
{
  {
  "links": [
    {
      "rel": "self",
      "href": "/myService/sakila/actor/58"
    }
  ],
  "filmActor": [
    {
      "film": {
        "title": "BACKLASH UNDEFEATED"
      }
    },
    {
      "film": {
        "title": "BETRAYED REAR"
      }
    }
    // ...
  ]
}
```



```
myService.sakila.city.findFirst({ select: { lastUpdate: false, country: { lastUpdate: false } } })
{
  "city": "A Coruña (La Coruña)",
  "links": [
    {
      "rel": "self",
      "href": "/myService/sakila/city/1"
    }
  ],
  "cityId": 1,
  "country": {
    "country": "Spain",
    "countryId": 87
  },
  "countryId": 87
}
```

https://dev.mysql.com/doc/dev/mysql-rest-service/latest/sdk.html

Learn by example

Includes PWA demo app

Showcases features

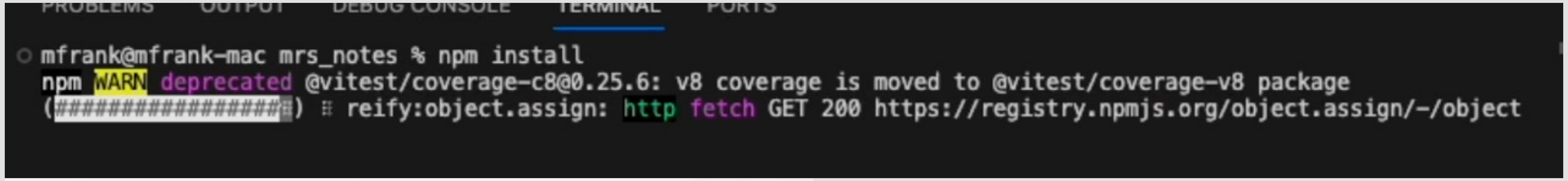
- Of the MySQL REST Service

Deployed directly

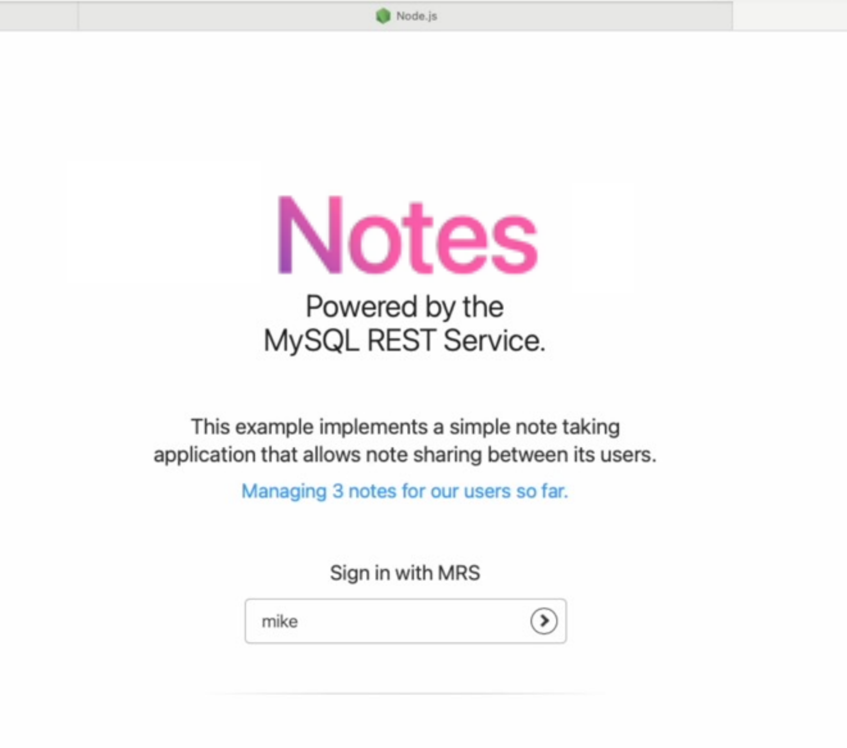
- From with VS Code

Upload and serve the application

- Using MySQL Routers



```
○ mfrank@mfrank-mac mrs_notes % npm install
  npm WARN deprecated @vitest/coverage-c8@0.25.6: v8 coverage is moved to @vitest/coverage-v8 package
  (##################) ⠸ reify:object.assign: http fetch GET 200 https://registry.npmjs.org/object.assign/-/object
```



Notes

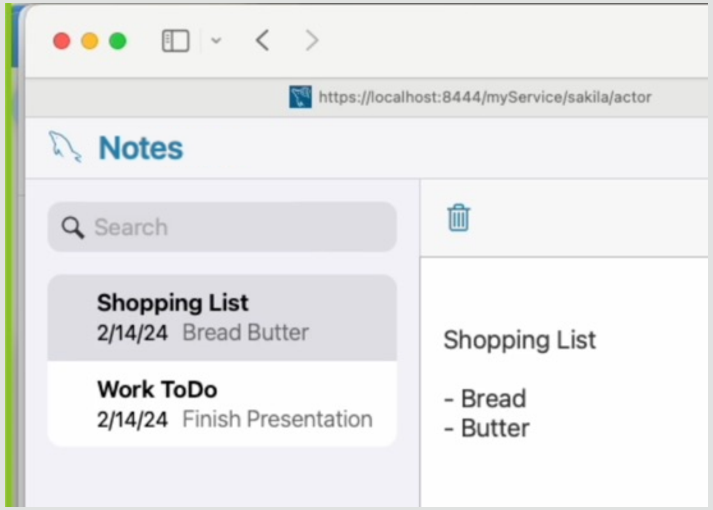
Powered by the MySQL REST Service.

This example implements a simple note taking application that allows note sharing between its users.

Managing 3 notes for our users so far.

Sign in with MRS

mike



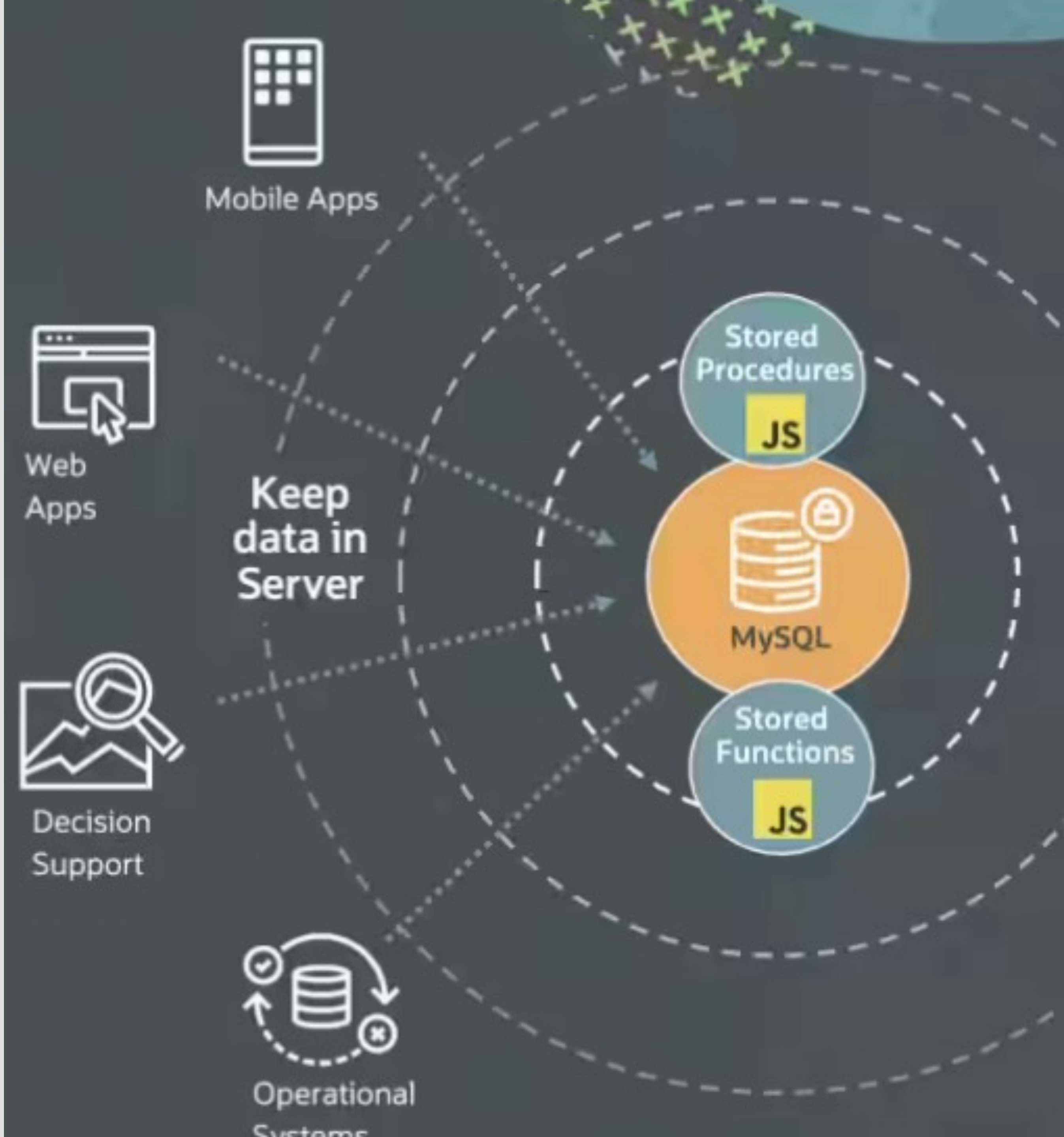
Shopping List
2/14/24 Bread Butter

Work ToDo
2/14/24 Finish Presentation

Shopping List

- Bread
- Butter

3 - Supporting JavaScript dev inside the MySQL server



JavaScript applications with MySQL

JavaScript Applications are popular

- Powerful for light weight front-end and server-side applications

Handles data-intensive use cases

- Data Validation
- JSON & String processing / Formatting
- Data Cleansing / Transformation
- Minimize data movement between server and clients

Streamline using Procedural programs inside Database

Handle	Data-intensive app functionality inside the database server
Minimize	Data movement
Reduce	Cost
Improve	Security
Simplify	ETL (Extract, Transform, and Load) to simpler ELT (Extract, Load, Transform) data pipelines - the modern data warehouse approach

MySQL Stored Programs - SQL vs JavaScript

	SQL Stored Procedures		JavaScriipt Stored Programs	
Expressiveness	✗	Hard to use, lacks basic constructs like containers (arrays, maps)	✓	Highly expressive and robust
Efficiency	✗	Challenging to optimize due to interpreted code	✓	Many JS code analysis tools. JavaScript apps are fast and optimized by GraalVM
Ecosystem	✗	Insufficient: Lacks support from IDEs, debuggers, testing frameworks, …	✓	Massive ecosystem of tools for developers of JavaScript applications
Availability of developers	✗	Few experienced programmers Especially with MySQL Ecosystem	✓	13.8 M Developers The most popular developer language
Reusable 3rd Party libraries	✗	Few, mostly code examples	✓	Thousands

JavaScript

Ubiquitous

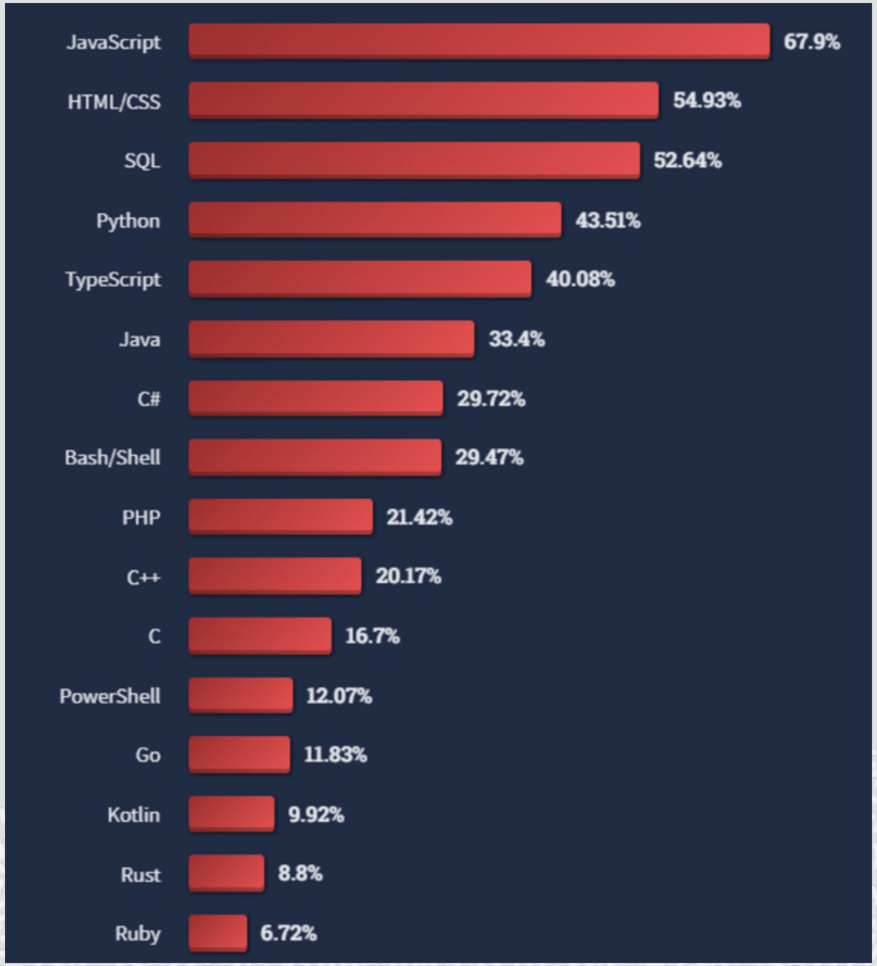
- One of the most used language by developers*
- > 98% of all web pages use JavaScript**

Multiple Runtimes

- Support in all major web browsers
- Massively used server-side runtimes
- Node.js
- Deno

Development Eco-system

- Npm contains > 2 million free to use JavaScript packages***
- > 10 million users use the npm package manager



* Stack Overflow 2024 survey
** https://w3techs.com/technologies/details/cp-javascript
*** https://www.npmjs.com/

Defining JavaScript stored programs

Simple Syntax

- LANGUAGE clause now allows JavaScript
- String quoting mechanism to enclose non-SQL language
  - AS$$...$$
  - AS $JavaScript$ ... $JavaScript$

Function Environment

- No function redefinition in JavaScript required
- SQL argument identifiers directly available in JavaScript

Auto Type-Conversion

Transparent MySQL ↔ JavaScript type conversion

Supports all variations of INT, FLOATS, DATETIME, VARCHAR (utf8mb4)

```
CREATE FUNCTION gcd_js (a INT, b INT)
RETURNS INT LANGUAGE JAVASCRIPT AS $$

    let [x, y] = [Math.abs(a), Math.abs(b)];

    while (y) [x, y] = [y, x % y];

    return x;

$$
```

JavaScript inside SQL

SELECT

- Use anywhere where SQL stored functions can be used
- Expressions, Projection, WHERE clause, GROUP-BY, JOIN, ORDER BY, HAVING etc.

DMLs, DDLs, VIEWs

- Support inside DMLs (INSERT, UPDATE, DELETE, …)
- DDLs including CREATE TABLE AS SELECT
- Support inside VIEWs

Interoperability

- Invoke JavaScript & SQL functions and Programs inside existing SQL stored functions or procedures
- Chain JavaScript & SQL stored functions together using input / output arguments

```
SELECT col1, col2, gcd_js(col1,col2)
FROM my_table
WHERE gcd_js(col1, col2) > 1
ORDER BY gcd_js(col1, col2);
CREATE TABLE gcd_table
AS SELECT gcd_js(col1,col2) FROM
my_table;

CREATE TABLE gcd_table
AS SELECT gcd_js(col1,col2) FROM
my_table;
```

SQL inside JavaScript

Statement Types

- Simple SQL statements
- Prepared statements with bind parameters

Data Access API

- Execute SQL inside JavaScript using XDevAPI
- Seamless MySQL ↔ JavaScript type conversion for query results

Session State

- Continue transactions inside JavaScript
- Access all session state inside JavaScript such as session variables & temporary tables

```
CREATE PROCEDURE gen_random_age (IN row_count INT) LANGUAGE
JAVASCRIPT AS $$
let insertStatement = session.prepare( "INSERT INTO
my_table(age) VALUES ( ? )"); for (let j = 0; j < row_count;
j++) {
let random_age = Math.trunc(Math.random() * 100);
insertStatement.bind(random_age).execute();
}
$$
```

```
CREATE PROCEDURE average_age (OUT avg_age FLOAT) LANGUAGE
JAVASCRIPT AS $$
    let age_sum = 0, count = 0;
    let selectStatement = session.sql(
      "SELECT age FROM my_table");
    let result = selectStatement.execute(), row = null;
    while(row = result.fetchOne()) {
        age_sum += row[0]; count++;
}
avg_age = age_sum / count;

$$
```

Debugging simplified

Standard Streams

- Access language standard output and error streams inside MySQL

Error Handling

- Translates unhandled JavaScript exceptions into MySQL errors
- Allow access to JavaScript stack traces in case of unhandled runtime error
- Translates MySQL errors and warnings into JavaScript exceptions while executing SQL statements inside JavaScript

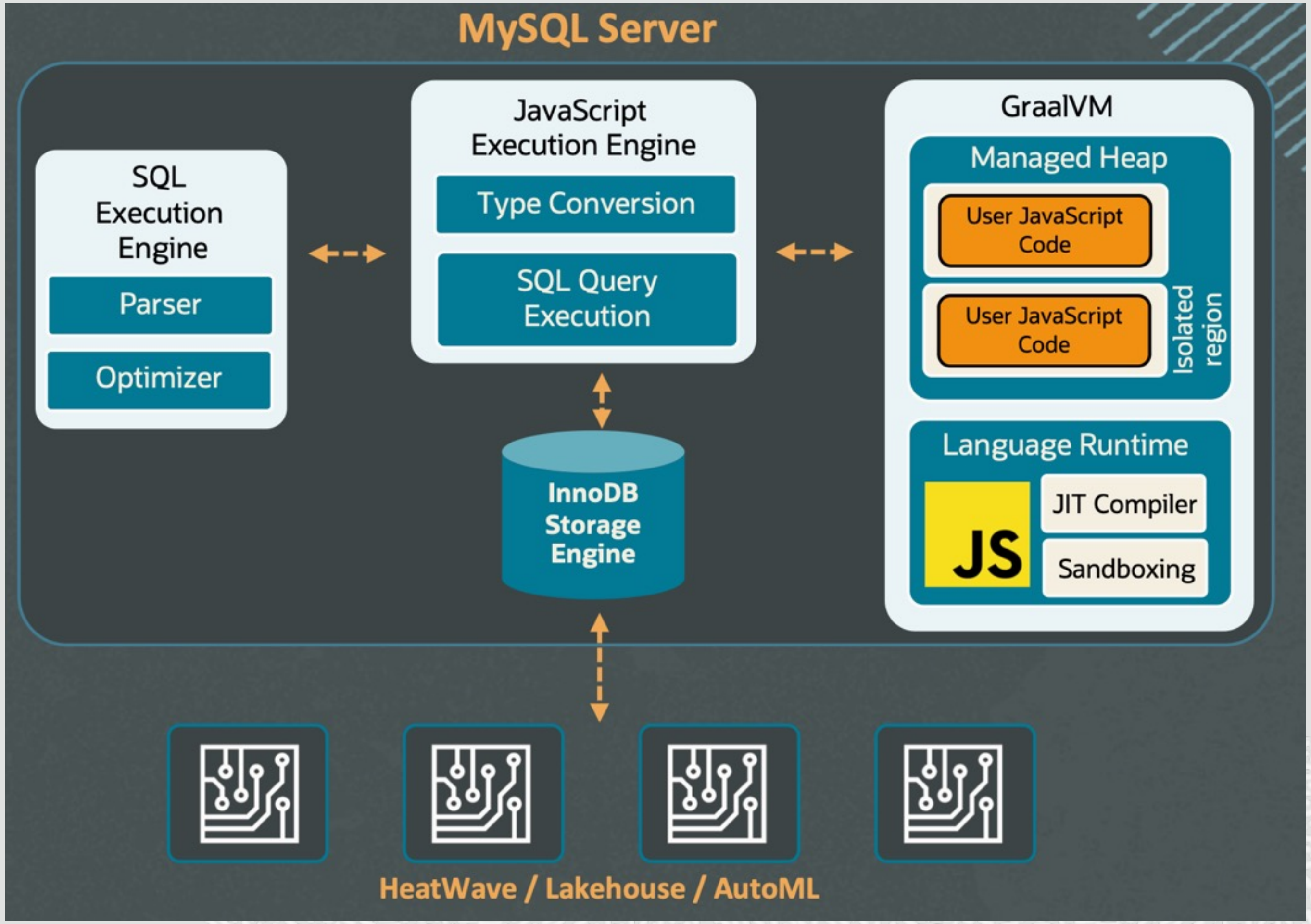
```
CREATE PROCEDURE division (IN a INT, IN b INT, OUT
result DOUBLE) LANGUAGE JAVASCRIPT AS $$
    function validate(num) {
      console.log("validating input value: ", num);
      if (num === 0) throw ("Division by Zero!");
    }
    validate(b);
    result = a / b;
$$
```

```
CALL division( 5, 0, @res);
ERROR 6000 (HY000): JavaScript> Division by Zero!

SELECT mle_session_state("stdout");
validating input value: 0

SELECT mle_session_state("stack_trace"); <js>
validate(division:9:187-214)
<js> division(division:11:222-232)
<js> :anonymous(division:15:256-265)
```

JavaScript inside MySQL Server

Works seamlessly with:

- InnoDB
- HA / Replication
- HeatWave Analytics
- HeatWave AutoML
- HeatWave AutoPilot
- HeatWave Lakehouse



JavaScript Stored Programs: Key Benefits

01

Express complex logic in database using JavaScript

02

Push data-intensive application logic inside the database

03

Reduce data movement cost

04

Includes GraalVM Enterprise Edition optimizations at no additional cost

05

Integrates with MySQL HeatWave

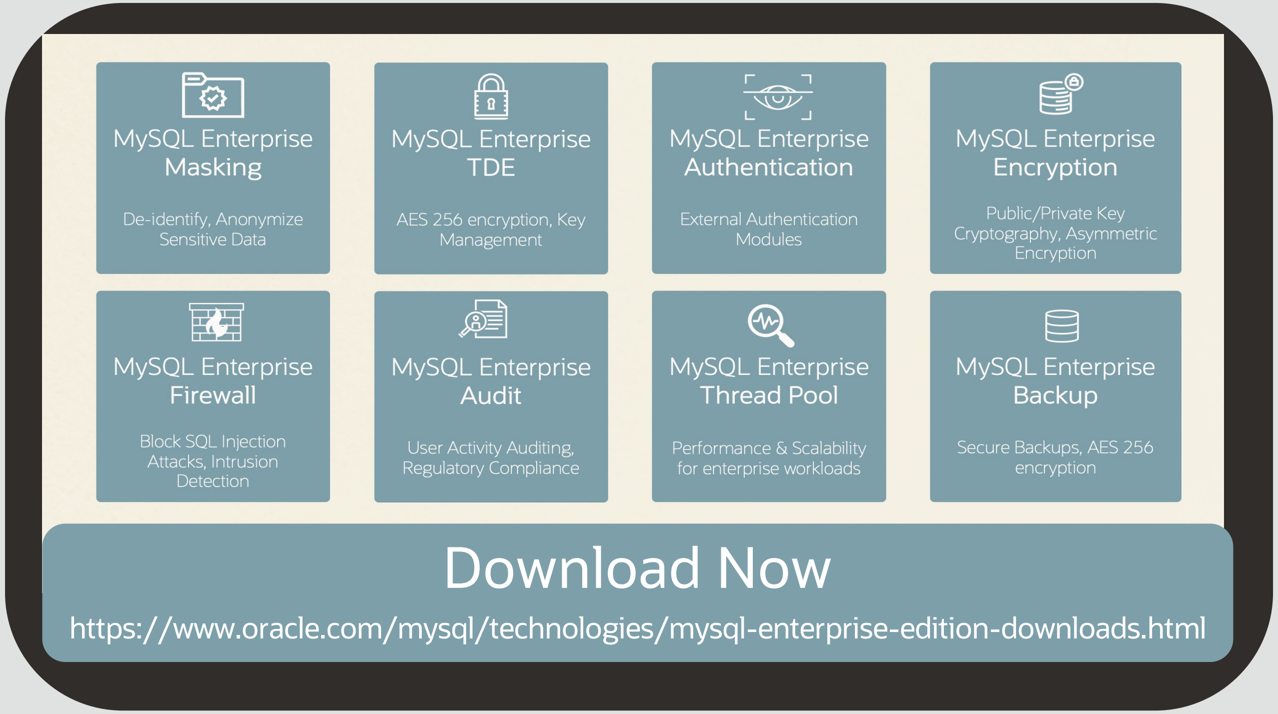
MySQL for Developers License

FREE DOWNLOAD OF MYSQL ENTERPRISE EDITION FROM OTN

Full Access to MySQL Enterprise Edition

- Enterprise Server
- Backup
- Router
- Shell
- Connectors
- JavaScript

Learn, Develop, Prototype



MySQL Enterprise Masking

De-identify, Anonymize Sensitive Data

MySQL Enterprise TDE

AES 256 encryption, Key Management

MySQL Enterprise Authentication

External Authentication Modules

MySQL Enterprise Encryption

Public/Private Key Cryptography, Asymmetric Encryption

MySQL Enterprise Firewall

Block SQL Injection Attacks, Intrusion Detection

MySQL Enterprise Audit

User Activity Auditing, Regulatory Compliance

MySQL Enterprise Thread Pool

Performance & Scalability for enterprise workloads

MySQL Enterprise Backup

Secure Backups, AES 256 encryption

Download Now

https://www.oracle.com/mysql/technologies/mysql-enterprise-edition-downloads.html

CLOUD NATIVE
COMPUTING FOUNDATION

4 - MySQL Kubernetes Operator

- Provides a consistent environment across development, testing, and production
- Allows developers to focus on code changes

Kubernetes

CLOUD NATIVE
COMPUTING FOUNDATION

A portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

Kubernetes Operator

Method of automatically deploying and managing a service.

Goals of an operator:

- Deployment
- Configuration
- Self-healing
- Backup & Restore
- Observability
- Using Kubernetes custom resources

Both Kubernetes Operator
& MySQL InnoDB Cluster
share a common goal to make it easier to deploy, automate and manage a service.

https://www.oracle.com/news/announcement/oracle-expands-support-for-open-source-community-2023-11-07/

Streamlining with Kubernetes

MySQL maintains container images

- Modern container based development model
- Kubernetes MySQL operator seamlessly streamlines
  - Containerized to Containerized
    - Dev to QA
    - QA to Production

MySQL Operator

- Automated deployment and management
  - Server
  - Router
  - HA/DR
- Self-healing
- Backup & Restore to/from
  - Object Stores
- Scaleup/Scaledown

- Rolling upgrades
  - Minimizes downtime
- Configuration Management
- Database Cloning
- Private container registries
- CNCF cert-manager support
- Enterprise Edition

https://dev.mysql.com/doc/mysql-operator/en/

MySQL Operator for Kubernetes Architecture

Uses Controllers

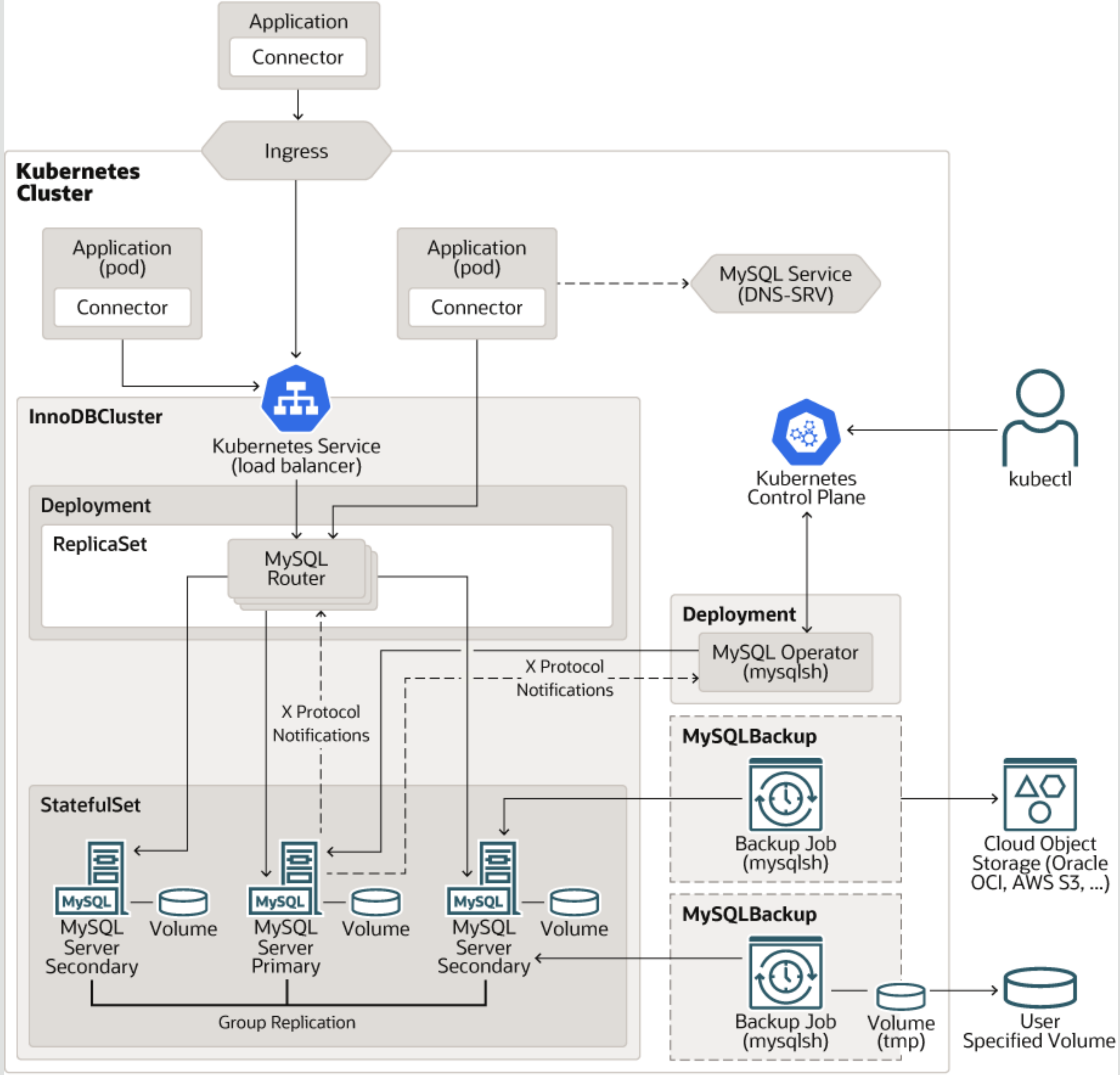
- Manage the life-cycle of containerized workloads
- Workload run as Pods

A MySQL Operator is

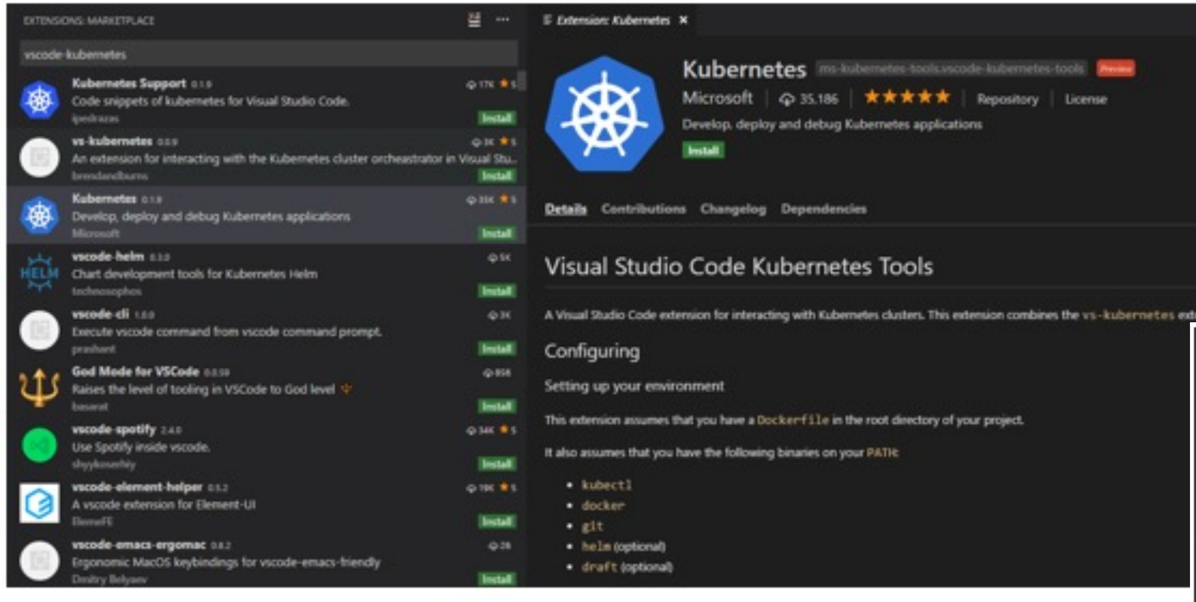
- Software running inside the Kubernetes cluster
- Interacts with the Kubernetes API to observe resources and services to assist with the life-cycle management.

Via Kubernetes controllers the operator configures MySQL

- MySQL Servers
- MySQL Replication (using Group Replication)
- MySQL Router



Install the Kubernetes Extension



Kubernetes Support 0.1.9 — Code snippets of kubernetes for Visual Studio Code. — ipedrazas

vs-kubernetes 0.0.9 — An extension for interacting with the Kubernetes cluster orcheastrator in Visual Stu... — brendandburns

Kubernetes 0.1.9 — Develop, deploy and debug Kubernetes applications — Microsoft

vscode-helm 0.3.0 — Chart development tools for Kubernetes Helm — technosophos

vscode-cli 1.0.0 — Execute vscode command from vscode command prompt. — prashant

God Mode for VSCode 0.0.59 — Raises the level of tooling in VSCode to God level — basarat

vscode-spotify 2.4.0 — Use Spotify inside vscode. — shyykoserhiy

vscode-element-helper 0.5.2 — A vscode extension for Element-UI — ElemeFE

vscode-emacs-ergomac 0.8.2 — Ergonomic MacOS keybindings for vscode-emacs-friendly — Dmitry Belyaev

Kubernetes ms-kubernetes-tools.vscode-kubernetes-tools Preview

Microsoft | 35,186 | ★★★★★ | Repository | License

Develop, deploy and debug Kubernetes applications

Visual Studio Code Kubernetes Tools

A Visual Studio Code extension for interacting with Kubernetes clusters. This extension combines the `vs-kubernetes` extension by @brendandburns and the `vs-helm` extension by @technosophos.

Configuring

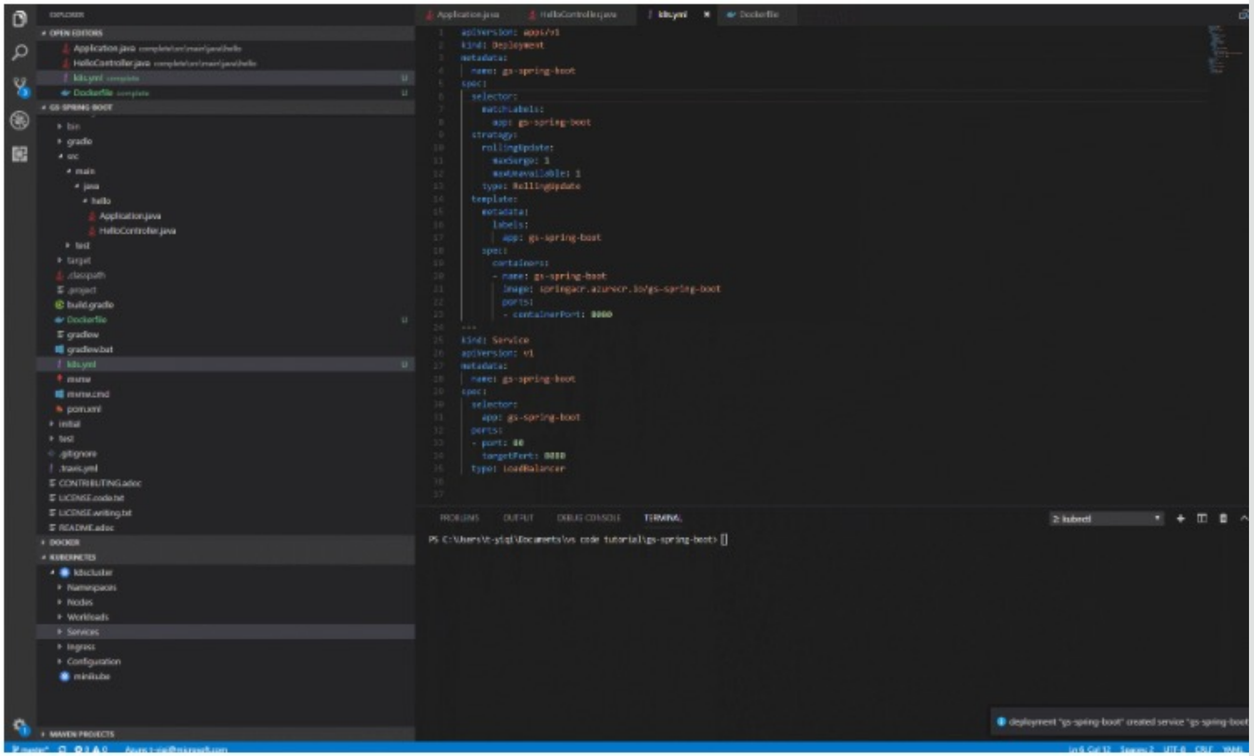
Setting up your environment

This extension assumes that you have a `Dockerfile` in the root directory of your project.

It also assumes that you have the following binaries on your `PATH`:

- `kubectl`
- `docker`
- `git`
- `helm` (optional)
- `draft` (optional)

Edit your manifest file
Open Command Pallette
Run Kubernetes Create



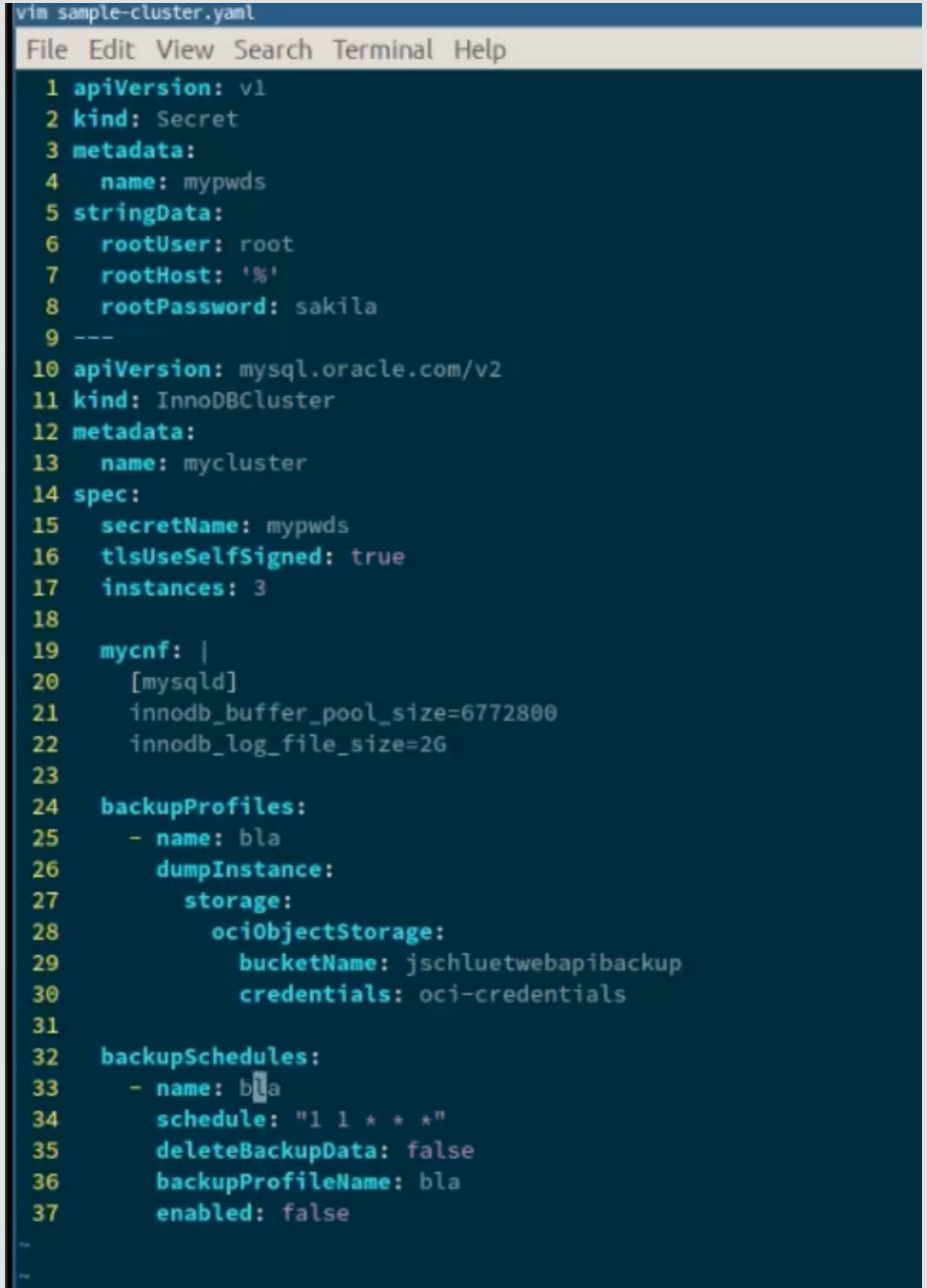
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: gs-spring-boot
spec:
  selector:
    matchLabels:
      app: gs-spring-boot
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: gs-spring-boot
    spec:
      containers:
      - name: gs-spring-boot
        image: springacr.azurecr.io/gs-spring-boot
        ports:
        - containerPort: 8080
---
kind: Service
apiVersion: v1
metadata:
  name: gs-spring-boot
spec:
  selector:
    app: gs-spring-boot
  ports:
  - port: 80
    targetPort: 8080
  type: LoadBalancer
```

```
PS C:\Users\t-yiqi\Documents\vs code tutorial\gs-spring-boot>
```

deployment "gs-spring-boot" created service "gs-spring-boot"

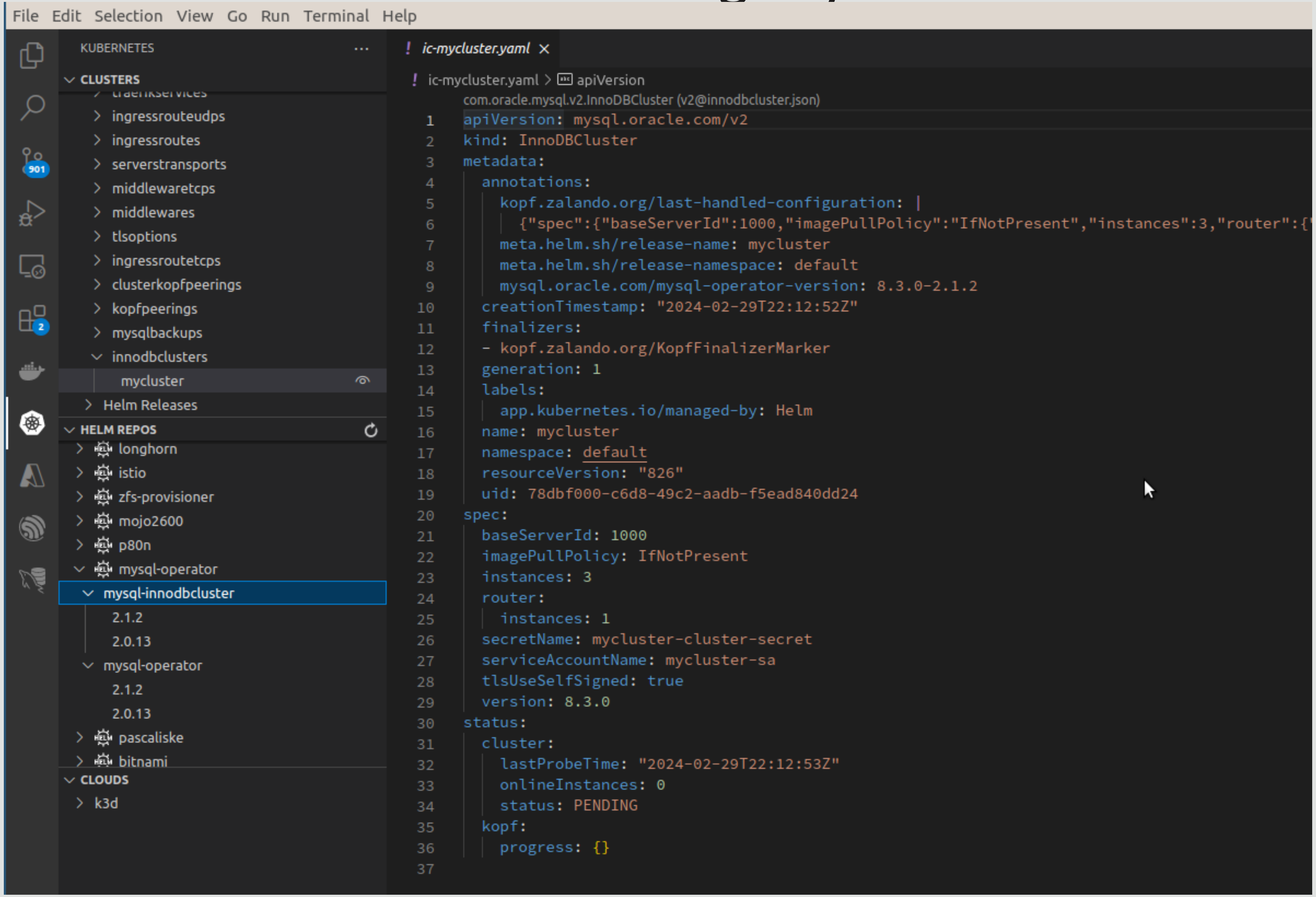
https://marketplace.visualstudio.com/items?itemName=ms-kubernetes-tools.vscode-kubernetes-tools

Define using YAML



```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mypwds
stringData:
  rootUser: root
  rootHost: '%'
  rootPassword: sakila
---
apiVersion: mysql.oracle.com/v2
kind: InnoDBCluster
metadata:
  name: mycluster
spec:
  secretName: mypwds
  tlsUseSelfSigned: true
  instances: 3

  mycnf: |
    [mysqld]
    innodb_buffer_pool_size=6772800
    innodb_log_file_size=2G

  backupProfiles:
    - name: bla
      dumpInstance:
        storage:
          ociObjectStorage:
            bucketName: jschluetwebapibackup
            credentials: oci-credentials

  backupSchedules:
    - name: bla
      schedule: "1 1 * * *"
      deleteBackupData: false
      backupProfileName: bla
      enabled: false
~
~
```

VS Code k8 extension running MySQL Innodb Cluster

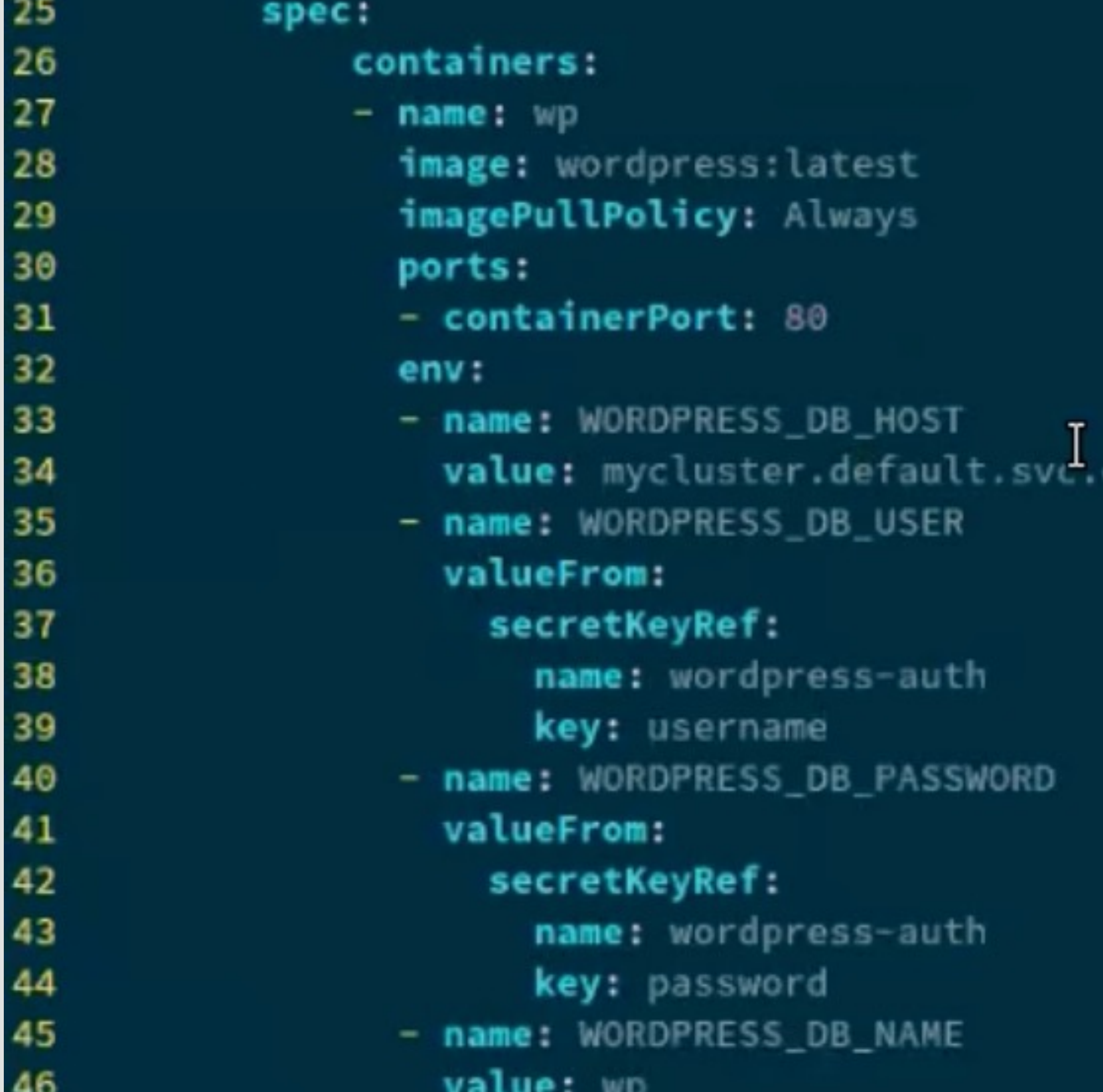


```yaml
apiVersion: mysql.oracle.com/v2
kind: InnoDBCluster
metadata:
  annotations:
    kopf.zalando.org/last-handled-configuration: |
      {"spec":{"baseServerId":1000,"imagePullPolicy":"IfNotPresent","instances":3,"router":{
    meta.helm.sh/release-name: mycluster
    meta.helm.sh/release-namespace: default
    mysql.oracle.com/mysql-operator-version: 8.3.0-2.1.2
  creationTimestamp: "2024-02-29T22:12:52Z"
  finalizers:
  - kopf.zalando.org/KopfFinalizerMarker
  generation: 1
  labels:
    app.kubernetes.io/managed-by: Helm
  name: mycluster
  namespace: default
  resourceVersion: "826"
  uid: 78dbf000-c6d8-49c2-aadb-f5ead840dd24
spec:
  baseServerId: 1000
  imagePullPolicy: IfNotPresent
  instances: 3
  router:
    instances: 1
  secretName: mycluster-cluster-secret
  serviceAccountName: mycluster-sa
  tlsUseSelfSigned: true
  version: 8.3.0
status:
  cluster:
    lastProbeTime: "2024-02-29T22:12:53Z"
    onlineInstances: 0
    status: PENDING
  kopf:
    progress: {}
```

Create complete Environment Stack with Kubernetes

For example Adding WordPress to MySQL

Defining the appdev ecosystem



```
    spec:
      containers:
      - name: wp
        image: wordpress:latest
        imagePullPolicy: Always
        ports:
        - containerPort: 80
        env:
        - name: WORDPRESS_DB_HOST
          value: mycluster.default.svc.
        - name: WORDPRESS_DB_USER
          valueFrom:
            secretKeyRef:
              name: wordpress-auth
              key: username
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: wordpress-auth
              key: password
        - name: WORDPRESS_DB_NAME
          value: wp
```

Conclusion

Equipped with these next generation tools

Developers can

Increase efficiency and productivity

Automate tasks

Focused App Dev

Integrated documentation

Improve quality

Automated checks

Testing environment

Simplify architecture

Reduce Time to Market

Shortened development time

Quickly iterate

Thank You!

Q&A

ORACLE