



Secure Your Data Workshop

MySQL Enterprise Security

Dale Dasker

Manager, MySQL Solutions Engineering
dale.dasker@oracle.com

March 14, 2024

Catherine Schrimsher

Principal MySQL Solutions Engineer
catherine.schrimsher@oracle.com

Eric Yanta

Principal MySQL Solutions Engineer
eric.yanta@oracle.com



Agenda

Achieve Compliance with MySQL Enterprise Security Features

- Workshop Overview
- Setup and Installation of:
 - Enterprise Audit
 - Enterprise Transparent Data Encryption
 - Enterprise Data Masking
 - Enterprise Firewall

Workshop Overview

- **Goals:**
 1. Create a OCI Compute server for hosting MySQL Enterprise Edition
 2. Install MySQL Enterprise Edition
 3. Overview & Setup Enterprise Audit, Enterprise Transparent Data Encryption, Enterprise Data Masking and Enterprise Firewall.
- **Not intended for:**
 - In-depth tutorial on Oracle Cloud Infrastructure
 - MySQL Training Class
- **Lab:**
 - https://bit.ly/MySQL_Workshop_Security

Why?

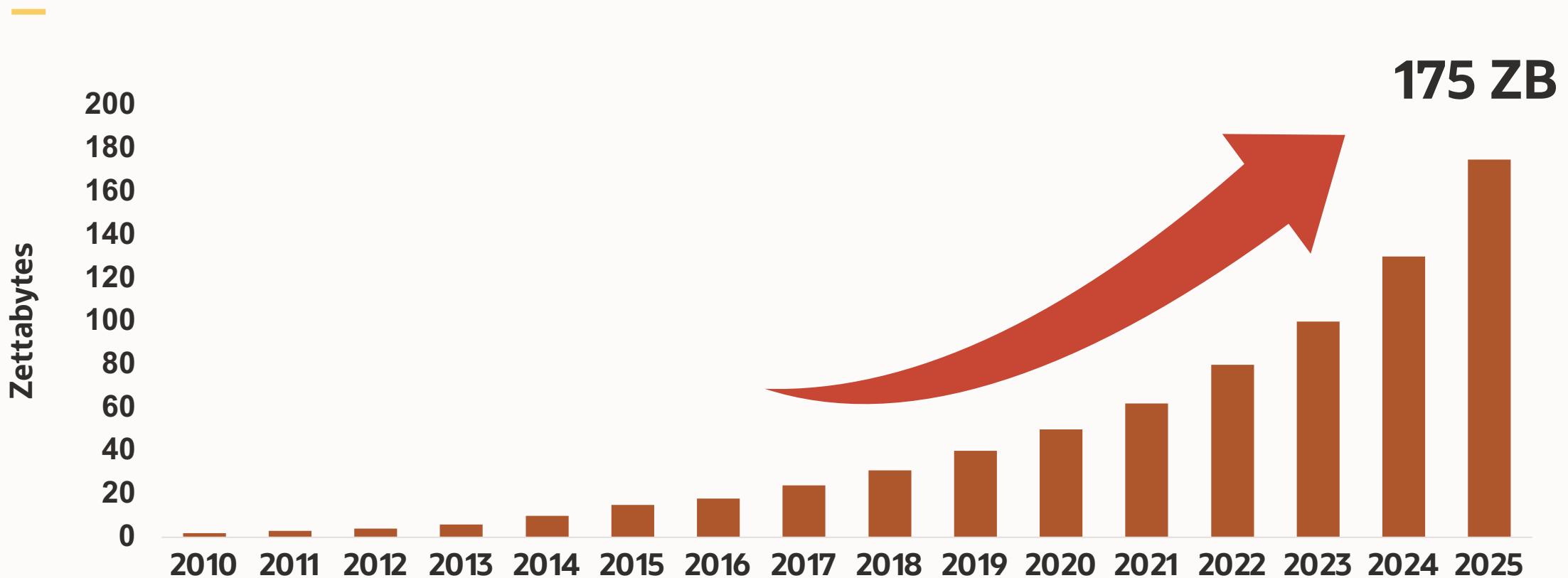


Data: Your Most Valuable Asset

The image is a circular graphic containing a dense cloud of text labels related to personally identifiable information (PII). The labels are color-coded and include:

- Name
- Surname
- Fingerprint
- Nationality
- Street
- Payment Card Information
- Credit Card Number
- Gender
- County
- NINO
- Citizenship
- Passport Number
- Race
- Postal Code
- Employee Identification Number
- Linked Personally Identifiable Information
- Salary
- Location
- Physical Characteristics
- Height
- Next of Kin
- Disability
- IMB
- Date of Birth
- Religion
- CURP
- Bonus
- MAC Address
- Country
- Financial Information
- Weight
- Cookie
- IP Address
- Stock
- Marital Status
- Personal Identification Numbers
- Visa Number
- Password
- Employment Data
- Phone Number
- City
- Name
- Email Address
- Health Insurance Number/Patient ID
- Patient Identification Number

Global Datasphere



Data Breaches – keep increasing...

Number of breaches in December 2023: **1,351**

Number of breached records in December 2023:
2,241,916,765

Unprotected Real Estate Wealth Network had more than 1.5 billion records stolen

Popular parental control app Kid Security had more than 300 million records exposed

<https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-2023>

What?



Data Protection Laws & Regulations



EU General Data Protection Regulation (GDPR)

- GDPR is a European Union “EU”-wide framework
 - Protection of personal data of EU-based individuals
- Published May 2016, Enforced May 2018
- Fines for GDPR violations are
 - The **greater of 20,000,000 Euros or 4% of annual revenue** (R150, A83)
- Data must be processed with controls that provide “*appropriate security and confidentiality*”
 - Data privacy as a fundamental right
 - Defines Data protection responsibilities, baselines, principles
 - Provides Enforcement Powers
- Exact security controls are not specified in the GDPR
 - **What? But not How?**

Regulatory Compliance

- **Regulations**

- PCI – DSS: Payment Card Data
- HIPAA: Privacy of Health Data
- Sarbanes Oxley, GLBA, The USA Patriot Act:
 - Financial Data, NPI "personally identifiable financial information"
- FERPA – Student Data
- EU General Data Protection Directive: Protection of Personal Data (GDPR)
- Data Protection Act (UK): Protection of Personal Data

- **Requirements**

- Continuous Monitoring (Users, Schema, Backups, etc)
- Data Protection (**Encryption**, Privilege Management, etc.)
- Data Retention (Backups, User Activity, etc.)
- Data **Auditing** (User activity, etc.)



How?



Data Protection & Regulatory Compliance

MySQL Enterprise Edition



Manage
Privileged Users



Protect
Dev & Test Data



Encrypt
Your Data



Detect
Database Activity



MySQL Enterprise Authentication



MySQL Enterprise Masking
MySQL Enterprise Backup

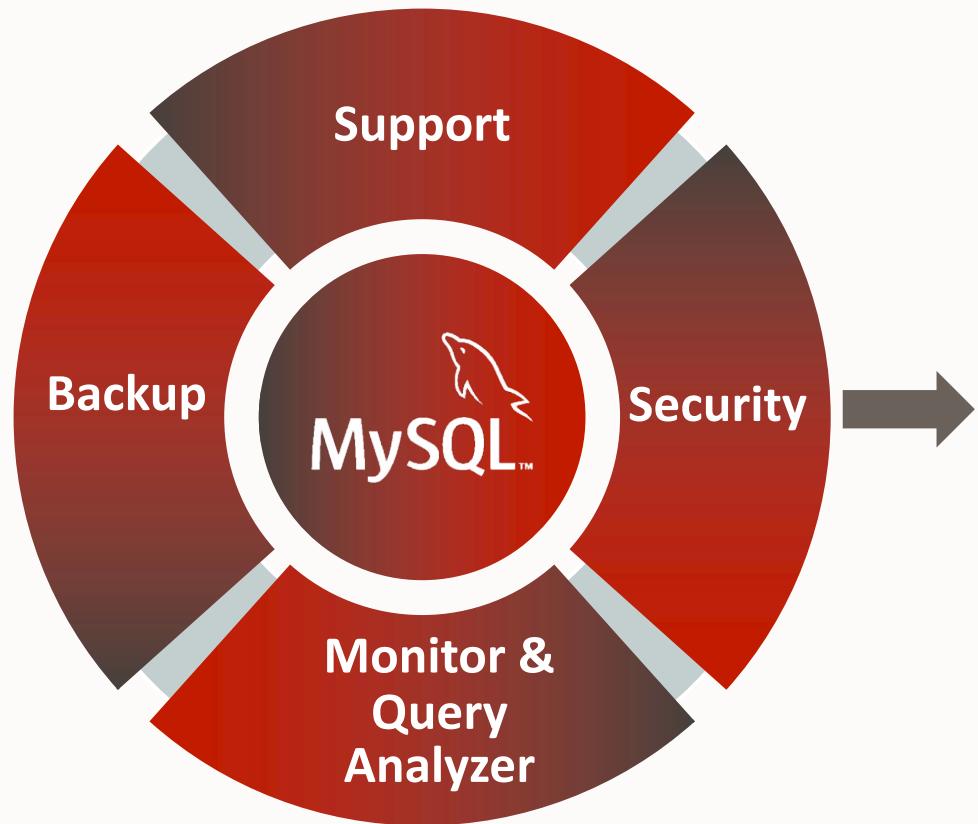


MySQL Enterprise TDE
MySQL Enterprise Encryption



MySQL Enterprise Audit
MySQL Enterprise Firewall
MySQL Enterprise Monitor

MySQL Enterprise Edition



- ✓ MySQL Enterprise **Audit**
- ✓ MySQL Enterprise **Transparent Data Encryption**
- ✓ MySQL Enterprise **Data Masking**
- ✓ MySQL Enterprise **Firewall**



MySQL Enterprise Audit

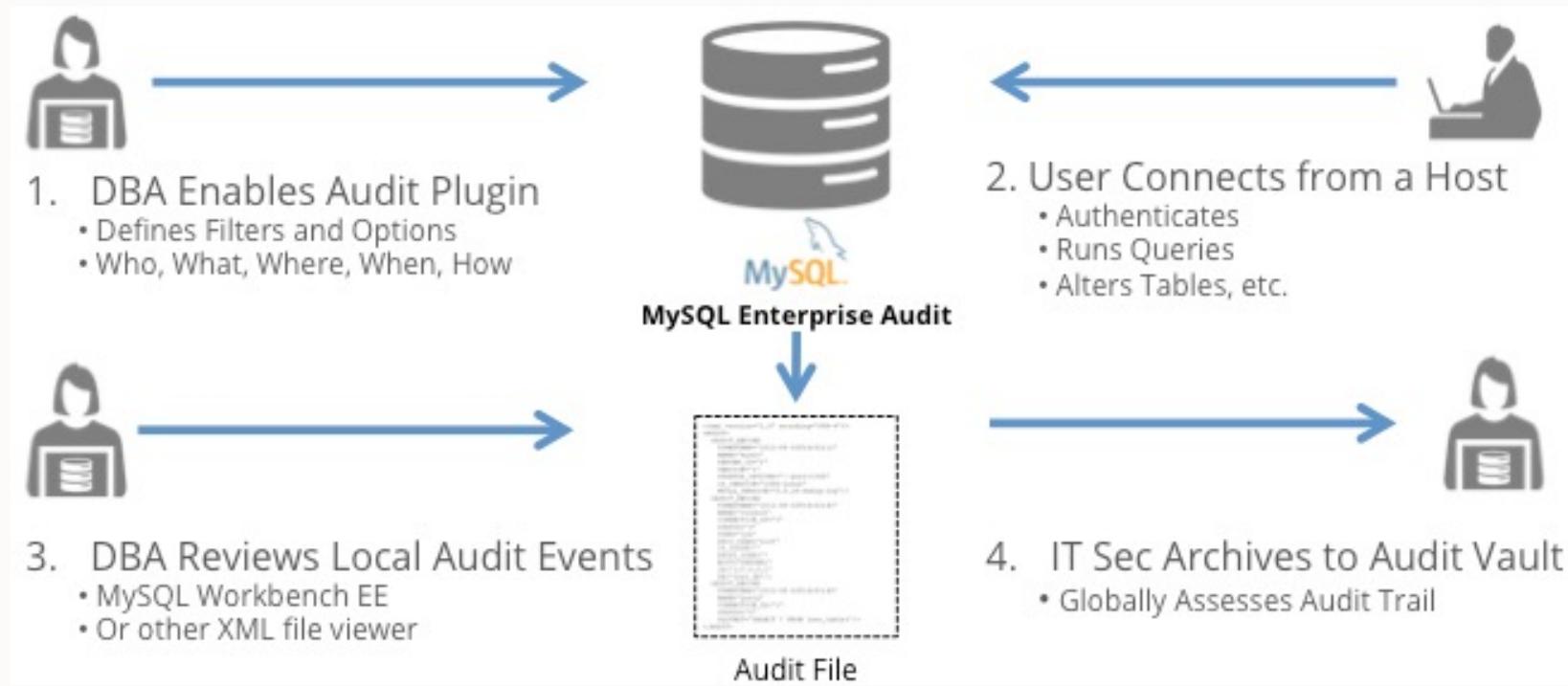
MySQL Enterprise Audit

- Out-of-the-box logging of connections, logins, and queries.
- Simple to fine grained policies for filtering and log rotation.
- Dynamically enabled & disabled.
- Various options for the Audit Logs:
 - XML-based audit stream
 - New 5.7.21+
 - JSON
 - Compression
 - Encryption
 - Remote Read Only SQL statement access
- Send data to a remote server / audit data vault
 - Oracle Audit Vault, Splunk, etc.

Adds regulatory compliance to
MySQL applications
(HIPAA, Sarbanes-Oxley, GDPR, etc.)

MySQL Enterprise Audit

Work Flow



Complete Audit Data

Complete event details

- Who
- What
- When
- Where
- How
- Status
- DB version
- OS version
- Options
- and more...

```
<?xml version="1.0" encoding="UTF-8"?>
<AUDIT>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:12"
    NAME="Audit"
    SERVER_ID="1"
    VERSION="1"
    STARTUP_OPTIONS="--port=3306"
    OS_VERSION="i686-Linux"
    MYSQL_VERSION="5.5.28-debug-log"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:41"
    NAME="Connect"
    CONNECTION_ID="1"
    STATUS="0"
    USER="joe"
    PRIV_USER="root"
    OS_LOGIN=""
    PROXY_USER=""
    HOST="SERVER1"
    IP="127.0.0.1"
    DB="joes_db"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:53:45"
    NAME="Query"
    CONNECTION_ID="1"
    STATUS="0"
    SQLTEXT="SELECT * FROM joes_table;"/>
</AUDIT>
```

Audit Log File Formats

Log File Format

XML - audit_log_format=NEW

```
<?xml version="1.0" encoding="utf-8"?>
<AUDIT>
<AUDIT_RECORD>
  <TIMESTAMP>2019-10-03T14:06:33 UTC</TIMESTAMP>
  <RECORD_ID>1_2019-10-03T14:06:33</RECORD_ID>
  <NAME>Audit</NAME>
  <SERVER_ID>1</SERVER_ID>
  <VERSION>1</VERSION>
  <STARTUP_OPTIONS>/usr/local/mysql/bin/mysqld --
socket=/usr/local/mysql/mysql.sock --port=3306</STARTUP_OPTIONS>
  <OS_VERSION>i686-Linux</OS_VERSION>
  <MYSQL_VERSION>5.7.21-log</MYSQL_VERSION>
</AUDIT_RECORD>
```

JSON – audit_log_format=JSON

```
{ "timestamp": "2019-10-03 14:21:56",
  "id": 0,
  "class": "audit",
  "event": "startup",
  "connection_id": 0,
  "startup_data": { "server_id": 1,
                    "os_version": "i686-Linux",
                    "mysql_version": "5.7.21-log",
                    "args": ["/usr/local/mysql/bin/mysqld",
                            "--loose-audit-log-format=JSON",
                            "--log-error=log.err",
                            "--pid-file=mysql.pid",
                            "--port=3306" ] } }
```

Audit Log File Security



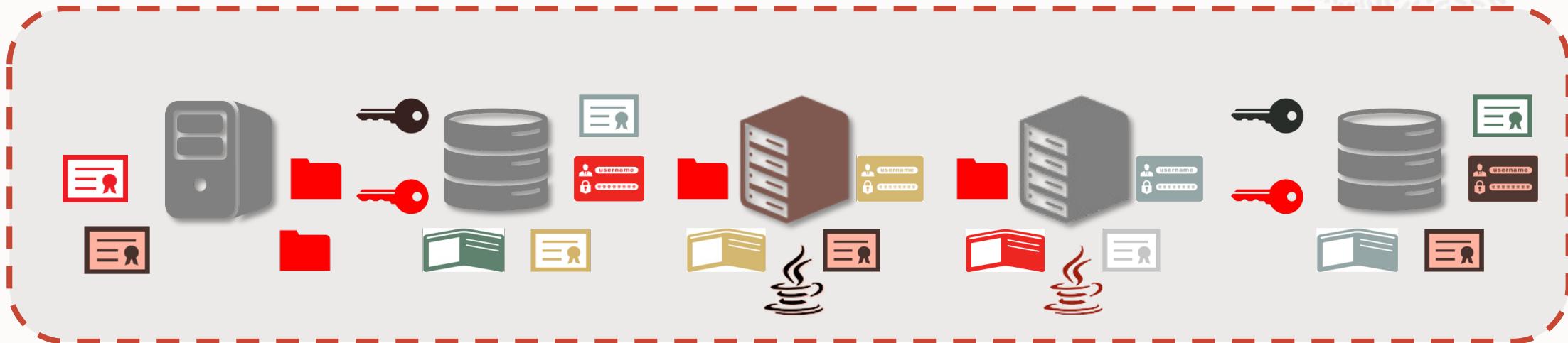
Compression

- Based upon gzip
- audit_log_compression=NONE|GZIP
- Adds .gz suffix to log files

Encryption

- Based upon AES-256-CBC
- audit_log_encryption=NONE|AES
- Uses **MySQL keyring plugin**
- Adds *.pwd_id.enc* suffix to log files

The Challenges of Key Management



Management

- Proliferation of encryption wallets and keys
- Authorized sharing of keys
- Key availability, retention, and recovery
- Custody of keys and key storage files

Regulations

- Physical separation of keys from encrypted data
- Periodic key rotations
- Monitoring and auditing of keys
- Long-term retention of keys and encrypted data

MySQL Key Ring



Get/Put MySQL Keys
On MySQL KeyRing

OKV or KMIP
Compliance Key Vault



Keys on the keyring are only accessible to internal components
Internal Code or Internal plugins

Key Rings are not persisted – in memory and protected in memory

ACLs - who key is for – for example InnoDB Tablespaces

Audit Filtering

Allows DBAs to “custom” design audit process

- Use very fine grained rules
 - Reduce audit log file size
 - Reduce File System IO and Storage / Increases performance (less items logged).
 - Increases audit log post processing efficiency – less data to process for immediate answers.
 - Defined using JSON
- Coarse grained rules
 - When you need to watch everything
 - Obsolete. Recommended is to use new audit log filtering.

Audit Log Filters

Expanded “Event” model

- Allows for very fine grained auditing

Simple but powerful

- Uses JSON to define filters

Event class	Event subclass
GENERAL	STATUS
CONNECTION	CONNECT
CONNECTION	CHANGE_USER
CONNECTION	DISCONNECT
TABLE_ACCESS	READ
TABLE_ACCESS	INSERT
TABLE_ACCESS	UPDATE
TABLE_ACCESS	DELETE
MESSAGE	INTERNAL
MESSAGE	USER

Comparison Audit to General Log

Connection

Audit Log output:

```
{  
  "account": {  
    "host": "",  
    "user": "root"  
  },  
  "class": "general",  
  "connection_id": 64,  
  "event": "status",  
  "general_data": {  
    "command": "Query",  
    "query": "select USER()",  
    "sql_command": "select",  
    "status": 0  
  },  
  "id": 2,  
  "login": {  
    "ip": "10.20.1.1",  
    "os": "",  
    "proxy": "",  
    "user": "root"  
  },  
  "timestamp": "2019-12-19 00:43:02"  
}
```

General Query Log output:

```
2019-12-19T00:43:02.532984Z 64 Connect root@10.20.1.1 on using  
SSL/TLS  
2019-12-19T00:43:02.533608Z 64 Query select @@version_comment  
limit 1  
2019-12-19T00:43:02.551259Z 64 Query select USER()  
2019-12-19T00:43:15.373949Z 60 Quit
```

- *Not as detailed*
- *No means for filtering content*
- *Can be easily disabled*
- *No log management*

Audit - Filtering Connections

Connection Event Fields

Name	Type	Description
status	INT	Status of the event: 0: OK, otherwise error state
user.str	STRING	Connecting user string
connection_type	INT	TCP/IP, socket, named pipe, SSL, shared memory
... (many more)		

Log all connection events:

- Successful and failed connection attempts
- Disconnects
- User change during session (change_user command)

Filters can be SIMPLE

```
(root@localhost)[mysql]SET @f = '{ "filter": { "class": { "name": "connection" } } }';  
Query OK, 0 rows affected (0.00 sec)
```

```
(root@localhost)[mysql]SELECT audit_log_filter_set_filter('log_conn_events', @f);  
+-----+  
| audit_log_filter_set_filter('log_conn_events', @f) |  
+-----+  
| OK |  
+-----+  
1 row in set (0.01 sec)
```

```
(root@localhost)[mysql]SELECT * FROM mysql.audit_log_filter;  
+-----+-----+  
| NAME | FILTER |  
+-----+-----+  
| log_conn_events | {"filter": {"class": {"name": "connection"}}} |  
+-----+-----+  
1 row in set (0.00 sec)
```

Filters can be Specific

Log failed SSL connection attempts:

```
{ "filter": {  
    "class": {  
        "name": "connection",  
        "event": {  
            "name": "connect",  
            "log": {  
                "and": [  
                    { "not": { "field": { "name": "status",  
                                         "value": 0 } } },  
                    { "field": { "name": "connection_type",  
                               "value": "::ssl" } } ] } } } }
```

Audit – Filtering Tables

Table Event Fields

Name	Type	Description
connection_id	STRING	Unique connection id.
sql_command_id	UINT	SQL statement type (SELECT, INSERT...)
query	STRING	Query string accessing the table
table_database	STRING	Database (schema) name
table_name	STRING	Table name
... (many more)		

Rules can be Specific related to Tables

All deletions, insertions, updates on bank_database.accounts

```
{ "filter":{  
    "class": {  
        "name": "table_access",  
        "event": {  
            "name": [ "delete", "insert", "update" ],  
            "log": {  
                "and": [ { "field": { "name": "table_database.str",  
                    "value": "bank_database" } },  
                        { "field": { "name": "table_name.str",  
                    "value": "accounts" } } ] } } } }
```

MySQL Enterprise **Transparent Data Encryption**

MySQL Enterprise Security Transparent Data Encryption

At Rest Encryption Covers

- InnoDB Tables and Tablespace
 - File Per Table Tablespace or General (Multi-Table) Tablespace
- MySQL System Tablespace
 - Data Dictionary Tables
- Binlog Encryption
- MySQL Enterprise Audit Logs
- MySQL Enterprise Backup Files
- Note: DBAs can optionally force Table Encryption
 - i.e. Users can only create encrypted tables



MySQL Enterprise Security **Transparent Data Encryption**

- **Data at Rest Encryption**

- [System | General | Data Dictionary] Tablespaces, Undo/Redo & Binary/Relay logs, Storage, OS File system
- Policy to **enforce table encryption**
- Strong Encryption – AES 256

- **Transparent to applications and users**

- No application code, schema or data type changes

- **Transparent to DBAs**

- Keys are hidden from DBAs, no configuration changes

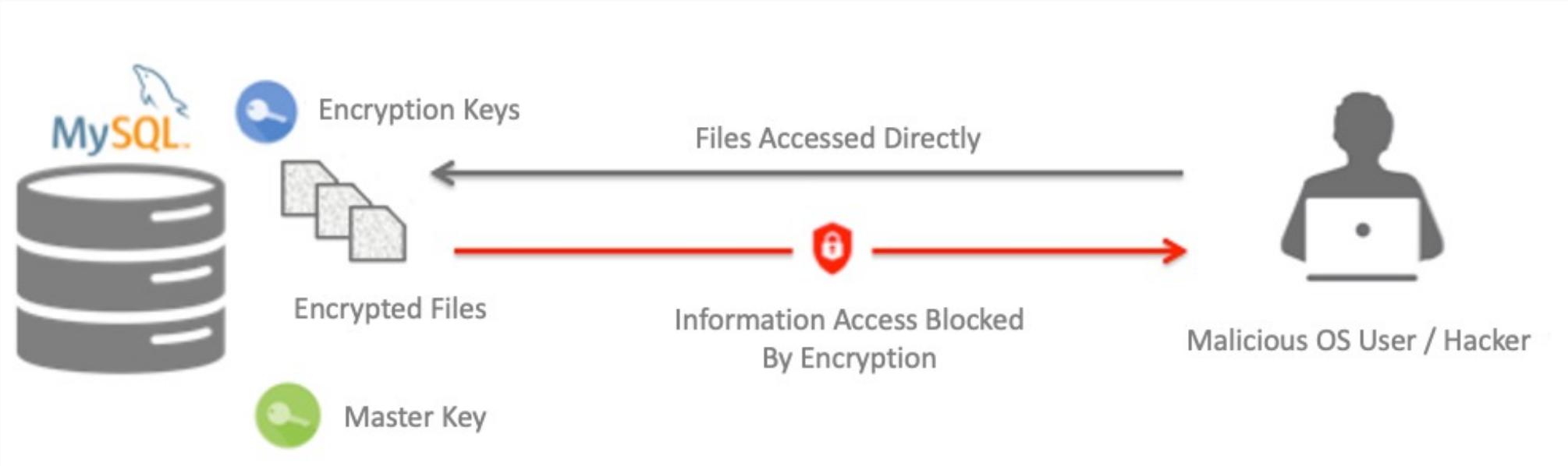
- **Requires Key Management**

- Protection, rotation, storage, recovery



MySQL Enterprise Transparent Data Encryption (TDE)

Protects against Attacks on Database Files



Encrypted Tablespace Files, Undo/Redo logs,
Shared Tablespaces, and Binary & Relay logs

Keyring plugin - used to retrieve keys from Key
Stores over Standardized **KMIP** protocol

MySQL Enterprise Data Masking

MySQL Enterprise Edition: Masking and De-Identification

De-identify, Anonymize Sensitive Data

Data De-Identification helps database customers improve security

Accelerates compliance for

- Government – GDPR, CHHS
- Financial - PCI
- Healthcare – HIPAA, Clinic Trials Data

Reduce IT costs by simplifying sanitizing production data

- Transforming sensitive data for use in analytics, testing, development, and more

Employee Table

ID	Last	First	SSN
1111	Smith	John	555-12-5555
1112	Templeton	Richard	444-12-4444

Random Data Generation

ID	Last	First	SSN
2874	Smith	John	XXX-XX-5555
3281	Templeton	Richard	XXX-XX-4444

Masked View

MySQL Enterprise Edition: Masking and De-Identification

Data Masking

String data masking

- Mask a substring within a string : ArthXXXXnt
- Mask substrings at the beginning and at the end : XXthurDeXX

SSN masking : XXXX-XX-1234

Payment Card masking

- Strict: XXXXXXXXXXXXXXXXX7395
- Relaxed: 493812XXXXXXXXXXXX7395

Dictionary based masking

- gen_blacklist("007", "00designations", "Cover_identity") => Universal Exports

The screenshot shows three separate MySQL queries in the SQL editor and their corresponding results in the Result Grid.

Query 1: This query uses the `mask_inner` function to mask specific substrings in the `users` table. The result grid shows the original data alongside the masked versions.

first_name	last_name	marital_status	email	ssn	phone
Ca****	Ow***	Married	y*****@example.com	XXX-XX-9083	1-555*****
Ad*****	Pa****	Married	u*****@example.com	XXX-XX-5668	1-555*****
Li**	Su*****	Married	o*****@example.com	XXX-XX-2918	1-555*****
Ja*****	Ro****	Single	q*****@example.com	XXX-XX-3303	1-555*****
Le***	Ma****	Married	o*****@example.com	XXX-XX-9266	1-555*****
An**	Do*****	Married	e*****@example.com	XXX-XX-6535	1-555*****
Ca****	Ad***	Married	h*****@example.com	XXX-XX-9213	1-555*****
Pa***	Ro***	Married	o*****@example.com	XXX-XX-8641	1-555*****
Lu**	Do****	Single	n*****@example.com	XXX-XX-9358	1-555*****
Ad***	Di***	Married	i*****		
Ti***	St****	Married	c*****		
Th***	Ch*****	Single	d*****		
Br***	Pe***	Single	t*****		
Ma***	Da***	Single	a*****		

Query 2: This query uses the `mask_ssn` function to mask the SSN column in the `users` table. The result grid shows the original data alongside the masked SSN values.

first_name	last_name	ssn
Carina	Ow***	XXX-XX-9083
Adelaide	Pa****	XXX-XX-5668
Liv	Su*****	XXX-XX-2918
Jasmine	Ro****	XXX-XX-3303
Lenniv	Ma****	XXX-XX-9266
Anna	Do*****	XXX-XX-6535
Carlos	Ad***	XXX-XX-9213
Paioe	Ro***	XXX-XX-8641
Luke	Do*****	XXX-XX-9358
Aoata	Di***	XXX-XX-9986

Query 3: This query uses the `mask_payment_card` function to mask the payment card number column in the `users` table. The result grid shows the original data alongside the masked payment card numbers.

first_name	last_name	ssn	payment_card
Carina	Ow***	XXX-XX-9083	XXXXXXXXXXXX4337
Adelaide	Pa****	XXX-XX-5668	XXXXXXXXXXXX0385
Liv	Su*****	XXX-XX-2918	XXXXXXXXXXXX5736
Jasmine	Ro****	XXX-XX-3303	XXXXXXXXXXXX3624
Lenniv	Ma****	XXX-XX-9266	XXXXXXXXXXXX4020
Anna	Do*****	XXX-XX-6535	XXXXXXXXXXXX5282
Carlos	Ad***	XXX-XX-9213	XXXXXXXXXXXX7037
Paioe	Ro***	XXX-XX-8641	XXXXXXXXXXXX3919
Luke	Do*****	XXX-XX-9358	XXXXXXXXXXXX4060
Aoata	Di***	XXX-XX-9986	XXXXXXXXXXXX5986

MySQL Enterprise Firewall

MySQL Enterprise Firewall



Real-time Database Intrusion Detection

Real Time Protection

- Queries analyzed and matched against Allow List

Blocks SQL Injection Attacks

- Positive Security Model

Block Suspicious Traffic

- Out of Policy Transactions detected & blocked

Learns Allow List

- Automated creation of approved list of SQL command patterns on a per user basis

Transparent

- No changes to application required

Enterprise Firewall		Configured: 8 of 8
Item	Info	
[+] [] [] Account Has Overly Permissive White List	[?]	
[+] [] [] Account Sending Excessive Percentage of Blocked Queries	[?]	
[+] [] [] Account Without Firewall Protection	[?]	
[+] [] [] Excessive Number of Queries Blocked By Firewall	[?]	
[+] [] [] Firewall Max Query Size Too Small	[?]	
[+] [] [] Firewall Not Enabled	[?]	
[+] [] [] Firewall Not Installed	[?]	
[+] [] [] Firewall Trace Has Been Enabled	[?]	

MySQL Enterprise Firewall monitoring

MySQL Enterprise Firewall

Block SQL Injection Attacks

- Allow: SQL Statements that match Whitelist
- Block: SQL statements that are not on Whitelist

Intrusion Detection System

- Detect: SQL statements that are not on Whitelist
 - SQL Statements execute and alert administrators

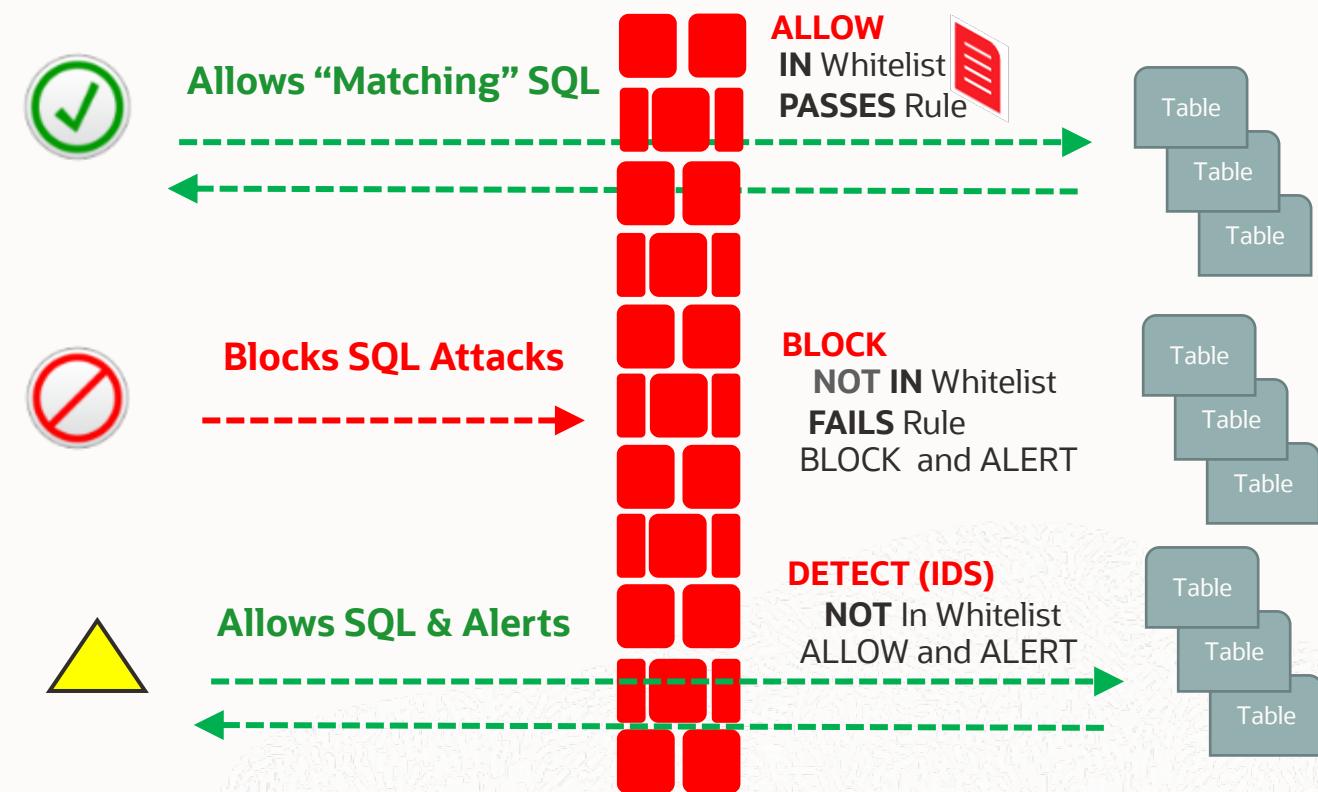


MySQL Enterprise Firewall: Operating Modes

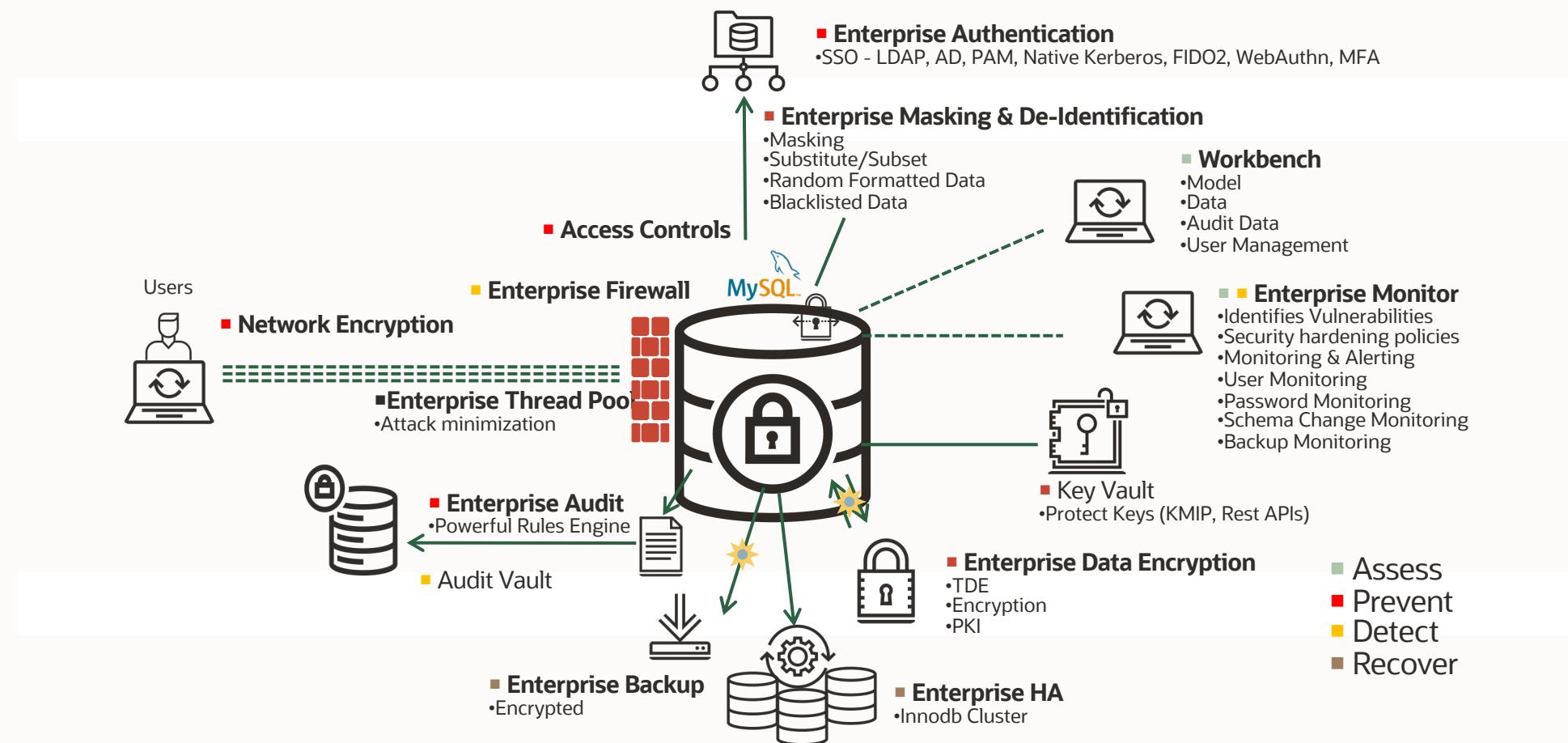
- 1 **ALLOW** – Execute SQL
 - SQL Matches Whitelist
 - SQL Passes Rule

- 2 **BLOCK** – Block the request
 - Not in Whitelist
 - SQL FAILs Rule
 - In Block Mode

- 3 **DETCT** – Execute SQL & Alert
 - Not in Whitelist
 - SQL FAILs Rule
 - In Alert Mode



MySQL Enterprise Security Architecture



Resources

MySQL Secure Deployment Guide

- <https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/>

60+ blogs to dive into specific topics and features

- https://blogs.oracle.com/mysql/search.html?contentType=Blog-Post&default=security*
- <https://dev.mysql.com/blog-archive/?cat=Security>

Whitepapers

- <https://www.mysql.com/why-mysql/white-papers/#en-22-40>

On Demand Webinars

- <https://www.mysql.com/news-and-events/on-demand-webinars/>

Forums

- <https://forums.mysql.com/>

MySQL Summit 2024

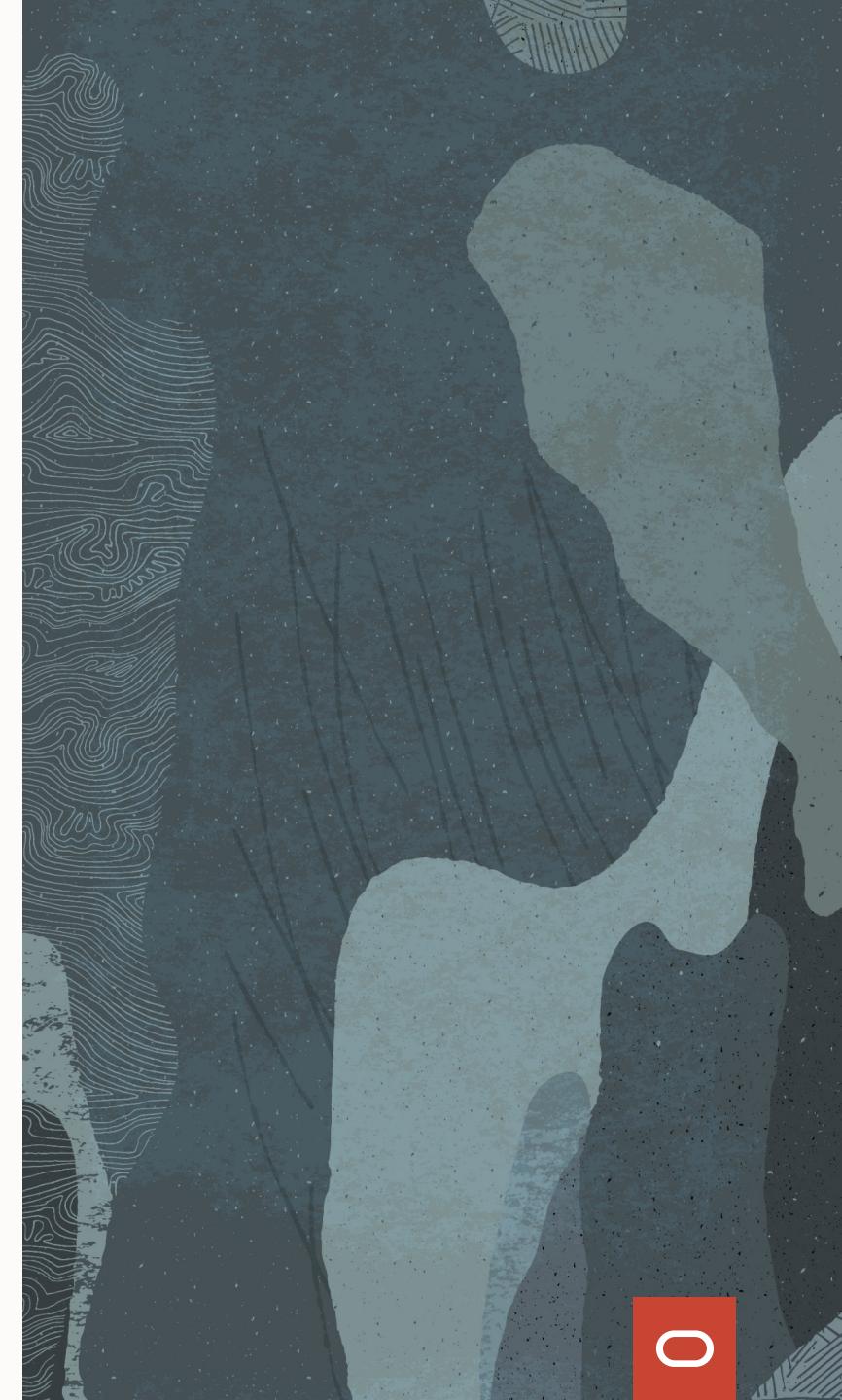
Wednesday, May 1st

Oracle Conference Center, Redwood Shores, California

- » **Generative AI and Vector Store**
- » **Machine Learning**
- » **Lakehouse and Analytics**
- » **Performance Tuning Tips and Tricks**
- » **High Availability and Disaster Recover**
- » **And many more popular topics**

Register for this free event

<https://www.oracle.com/events/mysql-summit/redwood-shores/>



Thank you!

Contact Us:

dale.dasker@oracle.com

eric.yanta@oracle.com

catherine.schrimsher@oracle.com

