OUT OF THE BOX
DATABASE MONITORING

RENATS VALIAHMETOVS
TECHNICAL SUPPORT ENGINEER
ZABBIX
Out of the box database monitoring

CLASSIC ODBC MONITORING
WHAT IS ODBC

- Standard application programming interface (API) for accessing database management systems (DBMS).
- Drivers exist for most DBMSs:
  - Oracle
  - PostgreSQL
  - MySQL
  - Microsoft SQL Server
  - Sybase ASE
  - SAP HANA
  - DB2
- Because different technologies have different capabilities, most ODBC drivers do not implement all functionality defined in the ODBC standard
WHAT TO MONITOR

Monitor any possible database performance metrics and incidents using the ready-to-use Zabbix templates.

Typical areas to monitor:

- Database performance
- Database engine availability
- Configuration changes

See the full list of metrics in template descriptions.
HOW DOES IT WORK

No direct connection to the databases by Zabbix, queries are sent to ODBC Manager for processing, which loads the ODBC Driver to connect to database.

Timeout parameter is used as the ODBC login timeout

Zabbix does not limit the query execution time

Two configuration files are used for this process

- odbc.ini – holds a list of installed ODBC database drivers
- odbcinst.ini – holds definitions of data sources
WHERE TO START – CLASSIC WAY

- Install relevant ODBC driver
  
  ```
  # yum -y install unixODBC unixODBC-devel
  ```

- Modify ODBC driver configuration files

  **odbc.ini**
  
  ```
  [root@localhost ~]# cat /etc/odbc.ini
  [MySQL]
  Description=NewDatabase
  Driver=MariaDB
  Server=localhost
  User=root
  Password=VerySecurePassword
  Port=3306
  Database=DatabaseName
  ```

  **odbcinst.ini**
  
  ```
  [root@localhost ~]# cat /etc/odbcinst.ini
  [MySQL]
  Description=ODBC for MySQL
  Driver=/usr/lib/libmyodbc5.so
  Setup=/usr/lib/libodbcmy5.so
  Driver64=/usr/lib64/libmyodbc8a.so
  Setup64=/usr/lib64/libmyodbc8a.so
  FileUsage=1
  ```
WHERE TO START – CLASSIC WAY

Test the ODBC configuration using isql

[root@localhost ~]# isql MySQL

Connected!

sql-statement
help [tablename]
quit

SQL> select itemid from items where hostid=10084 limit 1;

+------------+
| itemid     |
+------------+
| 23327      |
+------------+

SQLRowCount returns 1
1 rows fetched
SQL>
WHERE TO START – CLASSIC WAY

- Create “Database monitor” item
  - Name: New Database Metric
  - Type: Database monitor
  - Key: `db.odbc.select[NewMetricName,MySQL]`

- Specify the query needed

- Test the item, enjoy the metrics!
# ODBC TEMPLATES

## Assign a template

<table>
<thead>
<tr>
<th>Name</th>
<th>Hosts</th>
<th>Applications</th>
<th>Items</th>
<th>Triggers</th>
<th>Graphs</th>
<th>Dashboards</th>
<th>Discovery</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQL by ODBC</td>
<td>Hosts</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Dashboards</td>
<td>Discovery</td>
<td>Web</td>
</tr>
<tr>
<td>MySQL by ODBC</td>
<td>Hosts 1</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Dashboards</td>
<td>Discovery</td>
<td>Web</td>
</tr>
<tr>
<td>Oracle by ODBC</td>
<td>Hosts 2</td>
<td>Applications</td>
<td>Items</td>
<td>Triggers</td>
<td>Graphs</td>
<td>Dashboards</td>
<td>Discovery</td>
<td>Web</td>
</tr>
</tbody>
</table>

## Execute the discovery rules

- **Databases discovery: MySQL: Size of database mysql**
  - SQL: `db.odbc.select(mysql_size,"{MYSQL DSN}"`)`
  - Duration: 5m 7d 123d
  - Status: Database monitor
  - Databases: MySQL
  - Status: Enabled

- **Databases discovery: MySQL: Size of database performance_schema**
  - SQL: `db.odbc.select(performance_schema_size,"{MYSQL DSN}"`)`
  - Duration: 5m 7d 123d
  - Status: Database monitor
  - Databases: MySQL
  - Status: Enabled

- **Databases discovery: MySQL: Size of database sys**
  - SQL: `db.odbc.select(sys_size,"{MYSQL DSN}"`)`
  - Duration: 5m 7d 123d
  - Status: Database monitor
  - Databases: MySQL
  - Status: Enabled

## Check the latest data

- **MySQL: Buffer pool efficiency**
  - Duration: 1m 7d 123d
  - Status: Calculated
  - Value: 2021-03-12 02:31:34
  - Change: 0.00173 %
  - Graph

- **MySQL: Buffer pool utilization**
  - Duration: 1m 7d 123d
  - Status: Calculated
  - Value: 2021-03-12 02:31:35
  - Change: 87.4756 %
  - Graph

- **MySQL: Bytes received**
  - Duration: 7d 123d
  - Status: Dependent
  - Value: 2021-03-12 02:31:31
  - Change: 2.22 KBps
  - Graph
WHERE TO FIND TEMPLATES

✅ Fresh installation of Zabbix
✅ https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/templates
✅ https://www.zabbix.com/integrations
✅ https://share.zabbix.com/
Out of the box database monitoring

SYNTHETIC MYSQL MONITORING
Create new UserParameters

UserParameter=mysql.ping[*], mysqladmin -h"$1" -P"$2" ping
UserParameter=mysql.get_status_variables[*], mysql -h"$1" -P"$2" -sNX -e "show global status"
UserParameter=mysql.version[*], mysqladmin -s -h"$1" -P"$2" version
UserParameter=mysql.db.discovery[*], mysql -h"$1" -P"$2" -sN -e "show databases"

Make sure the commands work

Create .my.cnf under /var/lib/zabbix with the following content

[client]
user='zbx_monitor'
password=''<password>'

Confirm that user has necessary permissions to access DB

Use zabbix_get utility to test the UserParameter keys

If everything is working, assign MySQL by Zabbix agent template
Out of the box database monitoring

DB MONITORING WITH ZABBIX AGENT 2
ZABBIX AGENT 2

Zabbix agent 2 – new and improved version of Zabbix agent

Features:

✔ Custom intervals with active checks
✔ Written in Go
✔ Optimized
✔ Plugins
✔ Supports older configuration files

Installation:

```bash
# yum -y install zabbix-agent2
```
WHY ZABBIX AGENT 2

✔️ No need to install ODBC drivers because plugins do all the work.

✔️ Plugins are also written in GO language

✔️ Out of the box DB monitoring plugins:
  • MySQL
  • Oracle
  • PostgreSQL

✔️ Templates for agent 2

<table>
<thead>
<tr>
<th>Name</th>
<th>Hosts</th>
<th>Applications</th>
<th>Items</th>
<th>Triggers</th>
<th>Graphs</th>
<th>Dashboards</th>
<th>Discovery</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL by Zabbix agent 2</td>
<td>Hosts</td>
<td>Applications 2</td>
<td>41</td>
<td>11</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>Web</td>
</tr>
<tr>
<td>Oracle by Zabbix Agent 2</td>
<td>Hosts</td>
<td>Applications 2</td>
<td>77</td>
<td>16</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>Web</td>
</tr>
<tr>
<td>PostgreSQL Agent 2</td>
<td>Hosts</td>
<td>Applications 2</td>
<td>60</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Web</td>
</tr>
</tbody>
</table>
CONFIGURATION

✔ Two ways to configure:

- keys' parameters in frontend

  ![Type: Zabbix agent]

  ![Key: mysql.ping[tcp://127.0.0.1,user,password]]

- named sessions in zabbix_agentd2.conf (case sensitive)


✔ Create an item or apply a template

✔ Check available MySQL related item keys documentation page
Out of the box database monitoring

LLD FOR DB MONITORING
WHY LLD

- Automatically creates items, triggers, and graphs for different entities on a host.
- Parses data, received in Zabbix-specific JSON format
- Different sources can be used:
  - Built in discovery keys
  - Dependent on a built-in item key
  - Dependent on a custom script/custom UserParameter
HOW TO CONFIGURE CUSTOM LLD

- Decide on the payload delivery method (script, sender, UserParameter).
- Make sure payload is in the following format:

  ```json
  ["{#DATABASE}":"information_schema"],
  "{#DATABASE}":"mysql"],
  "{#DATABASE}":"performance_schema"],
  "{#DATABASE}":"sys"],
  "{#DATABASE}":"zabbix"
  ```

- Create LLD rule with type according to delivery method.
- Test the rule, if possible.
- Create filters or overrides, if required.
- Create prototypes.
Modify LLD rules of the official templates

- Modify/Create new entities
- Clone the templates
- Refer to templated discovery rule configuration
QUESTIONS?

RENATS VALIAHMETOVS
TECHNICAL SUPPORT ENGINEER
THANK YOU!

RENATS VALIAHMETOVS
TECHNICAL SUPPORT ENGINEER
ZABBIX