OpenStack Trove
Open Source DBaaS for the Cloud

Bartosz Żurkowski
Samsung R&D Institute Poland
Outline

Database as a Service introduction
Solutions available on the market
OpenStack Trove overview
Trove in a bigger picture
Towards open source contribution
Who are we?

- Samsung R&D Institute Poland (SRPOL)
- Building private cloud platform for 5G networks requirements based on OpenStack
- Team developing and maintaining DBaaS component - OpenStack Trove
- Active open source community members
  - Trove PTL (Project Technical Lead)
  - 3 core contributors
Operating databases is hard

<table>
<thead>
<tr>
<th>DBA responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application optimization</td>
</tr>
<tr>
<td>DB performance tuning</td>
</tr>
<tr>
<td>High availability</td>
</tr>
<tr>
<td>Monitoring</td>
</tr>
<tr>
<td>Scaling</td>
</tr>
<tr>
<td>Periodic backups</td>
</tr>
<tr>
<td>DB software upgrades</td>
</tr>
<tr>
<td>Hardening</td>
</tr>
<tr>
<td>DB software setup and config</td>
</tr>
<tr>
<td>Virtual resource provisioning</td>
</tr>
</tbody>
</table>
...in particular, when there are many instances
...instances of different types
... since we live in an era of polyglot persistence

Recommendations
- neo4j
  - Rapidly traverse links between entities

Reporting
- MariaDB
  - Tabular nature of data

User sessions
- redis
  - Rapid R/W access, no need to be durable

Products
- MongoDB
  - Products make natural document aggregate

Analytics
- cassandra
  - Large-scale analytics on large cluster

Financial data
- MariaDB
  - Tabular nature of data, transactional updates
What is Database as a Service?

This is what differentiates your business

This is what slows you down

DBA responsibilities

DBaaS responsibilities

Application optimization

DB performance tuning

High availability

Monitoring

Scaling

Periodic backups

DB software upgrades

Hardening

DB software setup and config

Virtual resource provisioning
DBaaS solutions available on the market

**AWS**
- AWS Aurora
  - Cloud native SQL database
- AWS RDS
  - MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server
- AWS DynamoDB
  - Key-value and document structs
- AWS ElastiCache
  - Redis, memcached
- AWS Neptun
  - Graph database

**Azure**
- Azure SQL Database
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Cosmos DB
  - Key-value, graph, column-family, and document structs
- Redis Cache

**Google Cloud**
- Google Cloud SQL
  - Mysql, PostgreSQL
- Google Cloud Bigtable
  - Wide column store
- Google Cloud Spanner
- Google Cloud Memorystore
  - Redis
Open source DBaaS

TROVE

an OpenStack Community Project
Introducing OpenStack Trove

- Database as a Service for OpenStack
- Build entirely on OpenStack
- Started around 2012
- Entered "The Big Tent" in 2014
- Full database lifecycle management
- 11 datastores
- Unified user interface
- Original project sponsor: Hewlett-Packard
- Major contributors over time: Rackspace, Tesora, HP, Red Hat, Mirantis, EasyStack, SUSE, eBay, China Mobile, IBM
- Current PTL: Dariusz Król, Samsung SRPOL
## Datastore capability matrix

<table>
<thead>
<tr>
<th>Datastore</th>
<th>Type</th>
<th>Provisioning</th>
<th>Schema management</th>
<th>Backup and restore</th>
<th>Resizing</th>
<th>Replication</th>
<th>Clustering</th>
<th>Cluster backups</th>
<th>Cluster resizing</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL</td>
<td>Relational</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MariaDB</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Percona</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>DB2</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Redis</td>
<td>Key-value</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cassandra</td>
<td>Column</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Vertica</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>MongoDB</td>
<td>Document</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CouchDB</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Couchbase</td>
<td>Multi-model</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>–</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
Features overview

- Instance provisioning
- Configuration
- Backup and restore
- Replication
- Clustering
• Create a database instance

$ trove create db-inst-1 m1.xlarge
  --datastore mariadb
  --datastore_version 10.1.32
  --size 2000
  --nic 'net-id=int-net'
  --databases db1 db2 db3
  --users usr1:pass1 usr2:pass2
  --availability-zone zone-1
  --configuration mariadb-cg
Features overview: Instance provisioning

- 1. Spawn VM
- 2. Allocate storage volume
- 3. Attach volume to VM
- 4. Format and mount filesystem
- 5. Coordinate database service setup
- 6. Apply initial configuration
- 7. Start database service
- 8. Secure database (e.g. remove anonymous user)
- 9. Create initial databases and users
- 10. Check service status
- 11. Report database status to the control plane
• **List available configuration parameters**

```
$ trove configuration-parameter-list 10.1.32 --datastore mariadb
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Min Size</th>
<th>Max Size</th>
<th>Restart Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_increment_increment</td>
<td>integer</td>
<td>1</td>
<td>65535</td>
<td>False</td>
</tr>
<tr>
<td>auto_increment_offset</td>
<td>integer</td>
<td>1</td>
<td>65535</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>innodb_buffer_pool_size</td>
<td>integer</td>
<td>5242880</td>
<td>18446744073709551615</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>max_connections</td>
<td>integer</td>
<td>1</td>
<td>100000</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

• **Create a configuration group**

```
$ trove configuration-create mariadb-cg
  '麼"max_connections": 1000, "wait_timeout": 50000}'
  --datastore mariadb
```
Features overview: Configuration management

- Attach configuration to an instance
  
  ```
  $ trove configuration-attach
db-inst-1 mariadb-cg
  $ trove configuration-attach
db-inst-2 mariadb-cg
  ```

- Update configuration
  
  ```
  $ trove configuration-update mariadb-cg
  "{"max_connections": 2000,
  "wait_timeout": 100000}"
  ```

- Detach configuration
  
  ```
  $ trove configuration-detach db-inst-1
  ```
Features overview: Backup and restore

• Create backup of an instance
  
  $ trove backup-create db-inst-1 bkp-1
  $ trove backup-create db-inst-2 bkp-2
  $ trove backup-create db-inst-3 bkp-3

• Restore backup into a new instance
  
  $ trove create db-inst-4 m1.small
  --datastore mariadb
  --backup bkp-1
  ...

• Schedule periodic backup
  
  $ trove schedule-create
  db-inst-1 "0 0 2 ? * *" daily-bkp
Features overview: Replication

- Create a replica set
  
  $ trove create rp-set ...

- Add replicas to replica set
  
  $ trove create rp-set
  --replica_of rp-set --replica_count 2

- Promote replica to be the new replica source of its set
  
  $ trove promote-to-replica-source rp-set-1

- Eject replica source
  
  $ trove eject-replica-source rp-set-1

- Detach replica from its replica source
  
  $ trove detach-replica rp-set-2
Features overview: Replication

- Create a replica set
  
  $ trove create rp-set ...

- Add replicas to replica set
  
  $ trove create rp-set
  
  --replica_of rp-set --replica_count 2

- Promote replica to be the new replica source of its set
  
  $ trove promote-to-replica-source rp-set-1

- Eject replica source
  
  $ trove eject-replica-source rp-set-1

- Detach replica from its replica source
  
  $ trove detach-replica rp-set-2
Features overview: Replication

- Create a replica set
  ```
  $ trove create rp-set ...
  ```

- Add replicas to replica set
  ```
  $ trove create rp-set
  --replica_of rp-set --replica_count 2
  ```

- Promote replica to be the new replica source of its set
  ```
  $ trove promote-to-replica-source rp-set-1
  ```

- Eject replica source
  ```
  $ trove eject-replica-source rp-set-1
  ```

- Detach replica from its replica source
  ```
  $ trove detach-replica rp-set-2
  ```
Features overview: Replication

- Create a replica set
  
  ```
  $ trove create rp-set ...
  ```

- Add replicas to replica set
  
  ```
  $ trove create rp-set
      --replica_of rp-set --replica_count 2
  ```

- Promote replica to be the new replica source of its set
  
  ```
  $ trove promote-to-replica-source rp-set-1
  ```

- Eject replica source
  
  ```
  $ trove eject-replica-source rp-set-1
  ```

- Detach replica from its replica source
  
  ```
  $ trove detach-replica rp-set-2
  ```
Features overview: Clustering

• Create a cluster
  
  $ trove cluster-create db-cluster-1 mariadb 10.1.26
  --instance "flavor=m1.small,volume=1000,nic='net-id'=int-net"
  --instance "flavor=m1.small,volume=1000,nic='net-id'=int-net"
  --instance "flavor=m1.small,volume=1000,nic='net-id'=int-net"
  --locality anti-affinity

• Add nodes to a cluster
  
  $ trove cluster-grow db-cluster-1
  --instance "flavor=m1.small,volume=1000,nic='net-id'=int-net"

• Remove nodes from a cluster
  
  $ trove shrink-cluster db-cluster-1 db-cluster-1-member-4 db-cluster-1-member-4

• Upgrade cluster to a new datastore version
  
  $ trove cluster-upgrade db-cluster-1 mariadb 10.1.32
Is it production ready?

Trove usage

- Production: 5%
- Testing: 8%
- Interested: 17%

Projects used in Production Deployments

n=357
Latest release (Rocky) stats:
- Commits: 113
- Contributors: 31 (10 active)
- Reviewers: 46 (9 active)
- Core team: 10
Large projects are built by companies

Trove

Ocata

Trove

Rocky

Table:

<table>
<thead>
<tr>
<th>#</th>
<th>Company</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tesora</td>
<td>114</td>
</tr>
<tr>
<td>2</td>
<td>AT&amp;T</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>EasyStack</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>NEC</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>SUSE</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Red Hat</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Fujitsu</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>Independent</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>99cloud</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>ZTE Corporation</td>
<td>2</td>
</tr>
</tbody>
</table>

Table:

<table>
<thead>
<tr>
<th>#</th>
<th>Company</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Awcloud</td>
<td>41</td>
</tr>
<tr>
<td>2</td>
<td>China Mobile</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>Samsung</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Red Hat</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>99cloud</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>China Telecom</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Fujitsu</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Independent</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>NEC</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>EasyStack</td>
<td>2</td>
</tr>
</tbody>
</table>
Trove in a bigger picture

[Diagram showing trove components and their interactions]

SOSCON 2018
1. Trove adjusts cluster topology according to user request

2. ... and sends updates to service registry (e.g. Consul or etcd)

3. Proxy instances poll registry updates from service registry

4. ... and reconfigure load balancing strategy
Trove in a bigger picture: Autonomic control loop

- **ANALYZE**
  - Root cause
  - Alerts, metrics

- **PLAN**
  - Action plan

- **MONITOR**
  - Metrics

- **EXECUTE**
  - Actions

- **SENSORS**
  - Collect metrics

- **EFFECTORS**
  - Execute actions

- **MANAGED RESOURCE**
1. Database nodes report metrics to the monitoring service

2. Monitoring service analyzes collected metrics and emits alerts based on static thresholds or ML predictions

3. Alerts and resource entities consumed by RCA service allow to determine root cause of detected issues and produce action plan (e.g. scaling or healing)

4. Workflow manager executes action plan by triggering APIs of different infrastructure services

5. Infrastructure services, e.g. database service, actuate plan on database nodes
Towards open source contribution

**First steps**
- Submitting company contrib. approval
- Researching community development workflow

**New roles**
- Becoming core contrib.
- Becoming PTL (Project Technical Lead)

**Upstream first**
- Adopting upstream-first development

**Inner circle**
- Internal development (features, bug fixes)
- No cooperation with the community

**Gaining trust**
- Reviews
- First commits
- Participation in IRC meetings

**New activities**
- Participating events
- Organizing meetings
- Planning release goals
- Attracting contributors
Final thoughts

• Open source projects give us great opportunity to innovate

• Open source commitment brings benefits to the company
  – Reduces maintenance and update cost
  – Respect and outside reputation
  – High-quality feedback from 3rd-parties
  – Developers morale

• Open source is not a one-way street
  – It is our duty to contribute back
  – At least give feedback or report bugs

• Large projects need strong companies standing behind
  – They create the roadmap in the long run
  – They bring major contributors and users into the project
Want to join one of the most prominent open source communities in the history?

Contact me:
Bartosz Żurkowski b.zurkowski@samsung.com

I will be glad to help You get started :)
Thank You