A Deep Dive into Trove

SCALE 13x, Los Angeles, CA

Kenneth Rugg
February 22, 2015
Who Am I?

- Ken Rugg, Founder & CEO of Tesora
- Tesora: the Trove Company
  - #1 contributor to Trove project
  - Tesora DBaaS Platform
    - packaged, supported version of Trove
  - Nearly all work is upstream first
  - Database & distribution certifications
- Database Virtualization Engine (DVE)
  - Open source, transparent sharding for MySQL

Diverse and growing community
- 128 contributors from 28 companies
- 1723 commits, and 231829 lines of code
# Transformation of Cloud Data Management

<table>
<thead>
<tr>
<th><strong>Traditional IT</strong></th>
<th><strong>Cloud</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning by DBA’s</td>
<td>Self-service provisioning</td>
</tr>
<tr>
<td>Database management by specialists</td>
<td>Developers manage their own databases</td>
</tr>
<tr>
<td>Waterfall development</td>
<td>Agile development</td>
</tr>
<tr>
<td>Few large machines / bare metal</td>
<td>Many small machines / virtualization</td>
</tr>
<tr>
<td>Oracle enterprise licenses</td>
<td>Many data management technologies</td>
</tr>
<tr>
<td>Captive audience</td>
<td>Competition with AWS</td>
</tr>
</tbody>
</table>
What is Trove?

- Database as a Service for OpenStack
- API’s for both development and operations
- Self service database provisioning
- Full database lifecycle management
- Multi-database support
- Both Relational and NoSQL
What is Trove? RDS for OpenStack
...and DynamoDB ...and RedShift

<table>
<thead>
<tr>
<th>Amazon AWS</th>
<th>OpenStack</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynamoDB</td>
<td>NoSQL</td>
</tr>
<tr>
<td>RDS</td>
<td>Trove</td>
</tr>
<tr>
<td>RedShift</td>
<td>RDBMS</td>
</tr>
<tr>
<td>EC2</td>
<td>DW</td>
</tr>
<tr>
<td>S3</td>
<td>Nova</td>
</tr>
<tr>
<td>EBS</td>
<td>Swift</td>
</tr>
<tr>
<td></td>
<td>Cinder</td>
</tr>
</tbody>
</table>

...and API’s for operations
What is Trove? - The Trove Mission

OpenStack Trove mission statement:

“To provide **scalable** and **reliable** Cloud Database as a Service provisioning functionality for both relational and non-relational database engines, and to continue to improve its fully-featured and extensible open source framework.”
## Tuning
- Automatically tune my.cnf
  - Buffer Pool Size
  - Log file size
  - max_connections
- Sane defaults
  - InnoDB only
  - Disable load data infile
  - Disable select into outfile
- API to programmatically set configuration groups

## Management
- Create database / schema
- Create users
- Grant permissions to a Schema
- Enable root user
- Resize flavor
- Resize volume
- Full and incremental backups
- Create replica

## Security
- Security groups
- Turn off SSH
- Remove anonymous user
- Remove non-localhost users
- Remove local file access
- Mangle root user password
- Apply security patches automatically
Trove In Production Today

- **Private Cloud: eBay**
  - Began mid 2013
  - Multiple Databases
    - MySQL, MongoDB, Redis
    - Cassandra, Couchbase
  - Multi-region + HA
  - Working on Clustering

- **Public Cloud**
  - HP Cloud Relational Database
    - Launched May 2012
  - Rackspace Cloud Databases
    - Launched August 2012

### Key Use Cases
- Development & test
- Web application hosting
- On-demand analytics

### Critical Capabilities
- Self-service provisioning & management
- Fleet wide configuration
- Multi-datastore architecture
What is Trove? - OpenStack Architecture

Trove

OpenStack

Tesora Database as Service/Trove

Red Hat Enterprise Linux OpenStack Platform

TROVE-API

NOVA

GUEST AGENT

SQL/NOSQL

MESSAGE BUS

CINDER

CINDER VOLUME

TROVE-TASKMANAGER

SWIFT

DB BACKUP

GLANCE

GLANCE IMAGES

TROVE-CONDUCTOR

GUEST IMAGE

NEUTRON

KEYSTONE
What is Trove? Multi-Datastore Architecture

Datastore agnostic code in Trove Controller & Dashboard

Datastore specific code isolated to Guest Agents

Trove Dashboard (Horizon)

Trove Controller

Guest Agent

MySQL

cassandra

PERCONA

mongoDB

MariaDB

Couchbase

Guest Agent

Guest Agent

Guest Agent

Guest Agent

Guest Agent

Guest Agent

PostgreSQL

redis
Trope Architecture: Code Modularity

trove/guestagent/
[...]  
backup/  
common/  
datastore/  
[...]  
strategies/  
[...]  

trove/guestagent/strategies/backup  
base.py  
base.pyc  
couchbase_impl.py  
couchbase_impl.pyc  
__init__.py  
__init__.pyc  
mysql_impl.py  
mysql_impl.pyc  
pagresql_impl.py  

trove/guestagent/strategies  
backup/  
[...]  
replication/  
restore/  
storage/  

grep class guestagent/strategies/backup/mysql_impl.py  
class MySQLDump(base.BackupRunner):  
class InnoBackupEx(base.BackupRunner):  
class InnoBackupExIncremental(InnoBackupEx):
Using OpenStack Trove
Getting started with OpenStack Trove

• As a Trove user
  – OpenStack distribution that includes Trove (such as HP Helion Dev Platform)
  – Tesora DBaaS platform, a Trove packaging tailored for the enterprise

• As a Trove developer
  $ git clone http://github.com/openstack/trove-integration
  $ cd trove-integration/scripts
  $ ./redstack install
  $ ./redstack kick-start mysql

• On top of DevStack
  – Add to localrc:
    • ENABLED_SERVICES+=,trove,tr-api,tr-tmgr,tr-cond
  – Swift should also be enabled for Backup and Restore.
    • ENABLED_SERVICES+=,s-proxy,s-object,s-container,s-account
    • SWIFT_HASH=<swift-hash-here>
Provisioning a database instance with Trove

$ trove create <instance-name> <flavor-id> --size <volume-size>

- Support for flavors
- Support for volumes using Cinder
- Optional parameters to create
  - Image per ‘datastore type’ and ‘version’
  - Support AZs using --availability_zone
  - Support for Neutron using --nic
Managing a database with Trove

• Resize flavor

• Resize volume

• Datastore specific extensions:
  – Create Database / Schema
  – Create Users
  – Grant Users Permissions on Databases

• Enable a Root User
Backup and Restore

$ trove backup-create
   <backup-name> <instance-id>

• Optional params:
  – Description through --description
  – Incremental backups using --parent

• To Restore backup use create:

  $ trove create
     <instance-name> <flavor-id>
     --size <volume-size>
     --backup <backup-id>

• Fully managed
• Triggered and tracked via API
• Streamed to Swift (OpenStack Object Storage)
• Incremental & full backups
• Multiple formats per datastore supported via strategies:
  – XtraBackup (Percona)
  – mysqldump

A Deep Dive into Trove - SCALE 13x
OpenStack Trove: Completed in Juno

- Async MySQL replication (master-slave)
- Clusters for MongoDB
- Neutron Support
- Support for PostgreSQL
- Config-groups enhancements
  - Configuration groups per datastore / version
  - Config-groups for MongoDB
- Backups for Cassandra and Couchbase
- Additional Tempest Tests
New in Juno: Replication

$ trove create <instance-name> <flavor-id>
  --size <volume-size>
  --replica_of <instance-id>

- Support for async MySQL replication (MySQL slave instances)

- Manual detach using
  $ trove update <instance-id> --detach-replica-source
New in Juno: Clusters

$ trove cluster-create <datastore> <ds-version>

• Optional parameters to cluster-create
  – --instance <flavor_id=flavor_id, volume=volume>
  – Specify multiple times to create multiple instances for your cluster

• Initial support in Juno added for MongoDB Clusters
  – Sets up mongo config server, and mongo query routers
  – Transparent and driven by configuration options
  – Support adding shards to existing cluster for horizontal scale out.
Planned for Kilo

• Building out clusters
  – Semi Synchronous MySQL clusters (Galera)
• Async Replication v2
  – GTID based replication
  – Manual failover support
• Associate flavors with datastores
• Access datastore logs via API
• Removing deprecated oslo-incubator code
• Upgrade testing through grenade
Where can I get OpenStack Trove?

• Get the source
  – OpenStack Trove Project
    • [https://git.openstack.org/cgit/openstack/trove](https://git.openstack.org/cgit/openstack/trove)
    • [http://github.com/openstack/trove.git](http://github.com/openstack/trove.git)
  – Trove python client binding and command line client
    • [https://git.openstack.org/cgit/openstack/python-troveclient](https://git.openstack.org/cgit/openstack/python-troveclient)
    • [http://github.com/openstack/python-troveclient.git](http://github.com/openstack/python-troveclient.git)
  – Trove design specifications for blueprints
    • [https://git.openstack.org/cgit/openstack/trove-specs](https://git.openstack.org/cgit/openstack/trove-specs)
    • [http://github.com/openstack/trove-specs](http://github.com/openstack/trove-specs)
  – Trove scripts for installation and testing, and elements for building guest images
    • [https://git.openstack.org/cgit/openstack/trove-integration](https://git.openstack.org/cgit/openstack/trove-integration)
    • [http://github.com/openstack/trove-integration.git](http://github.com/openstack/trove-integration.git)

• Installable packages and guest images from Tesora
  – Community Edition:
  – Enterprise Edition trial:
More about Trove

• Trove Wiki
  – https://wiki.openstack.org/wiki/Trove
• Trove Source
  – https://git.openstack.org/cgit/openstack/trove
• On IRC
  – #openstack-trove
• Trove Day
  – Tesora.com/troveday
  – Slideshare.net/Tesoracorp
• LinkedIn
  – OpenStack Trove Group

Contact information

Contact Tesora
  info@tesora.com
  www.tesora.com
  @tesoracorp

Contact Ken
  krugg@tesora.com
  www.tesora.com
  @kenrugg
Thank You!