

# RDS的开源实现-Trove



### 个人介绍

- 姚 昕
- 恩墨学院Oracle高级讲师
- Cloudera认证讲师
- Oracle 10g、11g、12c OCM认证
- 曾在IBM任职亚太客户支持
- 中国移动(苏州)研发中心运营部
- 个人技术Blog: <a href="http://blog.csdn.net/kiwi\_kid">http://blog.csdn.net/kiwi\_kid</a>







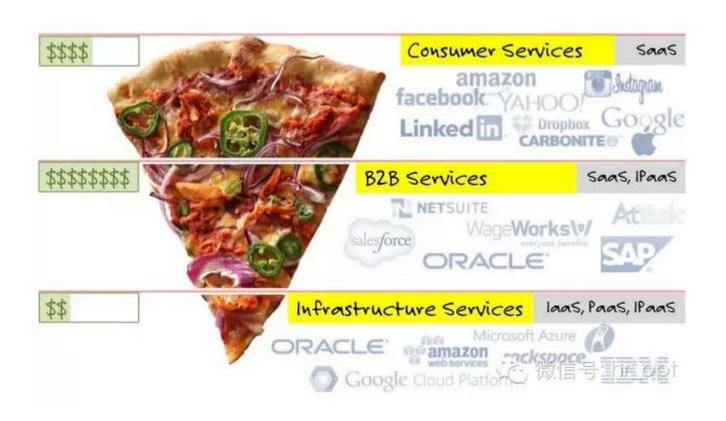
20世纪70、80、90年代出生的我们是非常幸运的。因为我们不仅经历了PC、互联网两次信息革命,体验到了信息革命给我们的生活工作带来了巨大影响与改变,如今又非常幸运地成为云计算时代的"弄潮儿",有机会深度参与第三次IT革命……

云计算之所以能被称为第三次IT 革命,主要因为云计算是近10年来少见的能够同时引起互联网、IT和电信业共同关注的新领域。在我看来,其主要原因在于云计算是一个融合了互联网商业模式、革命性的IT技术和电信服务运营的新生事物。它将资源更多地依赖于互联网的大环境实现"云"化,原来承担很多任务的用户端设备将逐渐获得"解放",变得更加简单和便捷……





#### 谁能举个通俗易懂的例子告诉我IAAS,SAAS,PAAS的区别?



### 序



本地部署 (On-Premises

在家自己做

基础设施即服务 (laaS)



买成品回家做

平台即服务 (PaaS)



则外卖 William gpt 软件即服务 (SaaS)



**夫披萨店吃** 





# SAAS

# **DBAAS**







amazon webservices



### 五、DBAAS



### • DBaaS如何改变企业IT?

# 大幅降低成本方面

- •像OpenStack Trove这种私有云DBaaS解决方案为企业提供了一种灵活的解决方案,这种解决方案可以安全地部署,同时遵循企业治理和数据安全体系。Trove提供了种类丰富的数据库技术,采用了易于使用、预先包装的配置。由于将使用各自数据库技术的最佳实践整合到这些配置当中,IT人员就不需要具备所有技术方面的专长。
- •标准化配置在降低支持成本的同时,还有助于确保数据完整性和安全性。DBaaS在IT基础设施里面运行,提供了帮助服务易于管理的众多功能,比如自动化备份、自动化升级和补丁管理以及自愈合。

# 加快配置减少等待

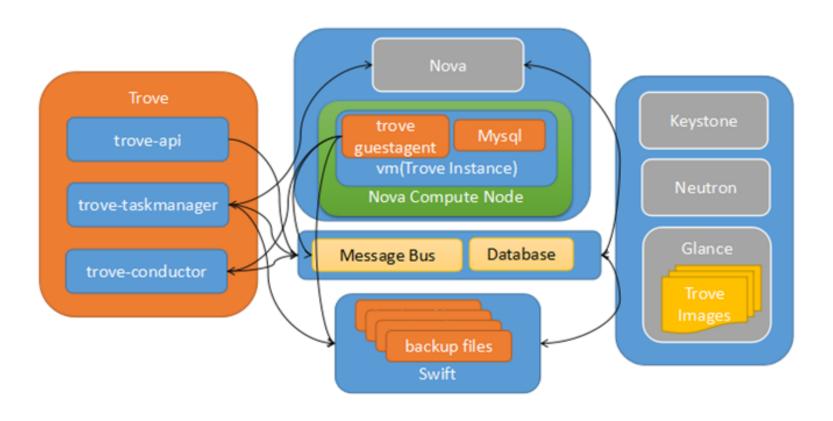
- •在企业里面的IT客户看来,DBaaS显著增强了灵活性。申请数据库服务器,然后等上数周、乃至数月来配置服务器,那种日子早就一去不复返了。现在只要轻松点几下按键,就能获得一台全面配置的数据库服务器!
- •此外,DBaaS提供的不仅仅是一台预先配置的数据库服务器;它还提供了全面管理的服务,让客户得以致力于最充分地利用数据库,同时依赖基础设施确保数据完整性、安全性和连续性。由于服务在私有云环境里面运行,可以保证符合数据保留和隐私等企业政策。

### 更容易享用 新的数据库 技术

- •在过去几年里,数据库领域已发生了显著的变化,NoSQL和NewSQL解决方案在可扩展性、可用性和灵活数据建模等方面带来了新的好处。在过去,开发人员被迫使用企业标准数据库,或者单独行事,自行管理任何新的数据库技术。
- •DBaaS数据库让IT客户得以选择适合其特定使用场合的数据库技术。MySQL在一种场合下也许是合适的解决方法,但是MongoDB在另一种场合下却可能是合适的解决方案。DBaaS解决方案让用户可以从数据库服务菜单当中选择合适的数据库技术。与此同时,可以向用户保证他们部署的配置已得到IT部门的审查,从而确保他们遵守行业最佳实践和企业政策。

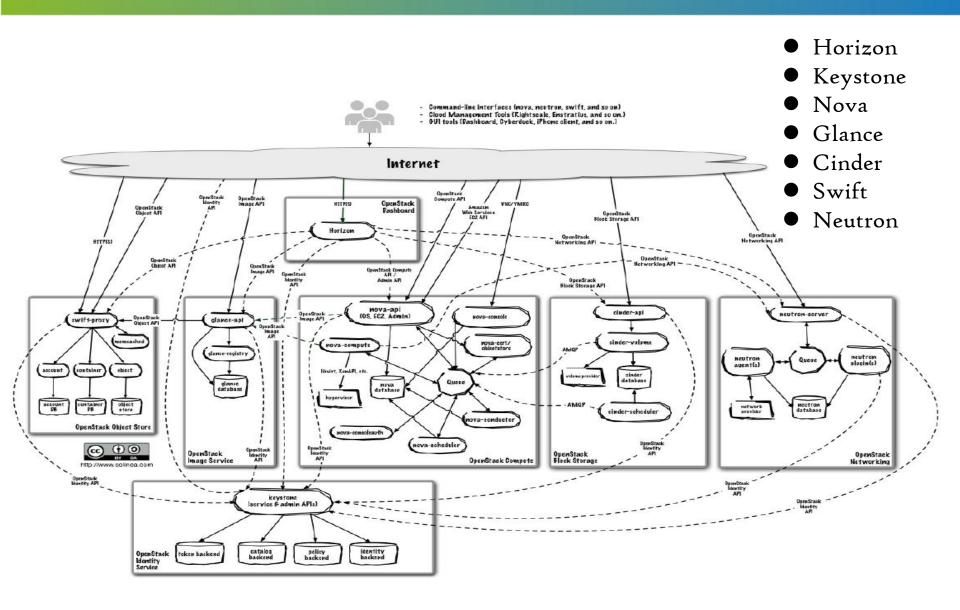
### 五、OpenStack Trove: 开源DBaaS





# OpenStack核心系统架构图





### Horizon Dashboard服务



在整个Openstack应用体系框架中,Horizon就是整个应用的入口。它提供了一个模块化的,基于web的图形化界面服务门户。用户可以通过浏览器使用这个Web图形化界面来访问、控制他们的计算、存储和网络资源,如启动实例、分配IP地址、设置访问控制等。



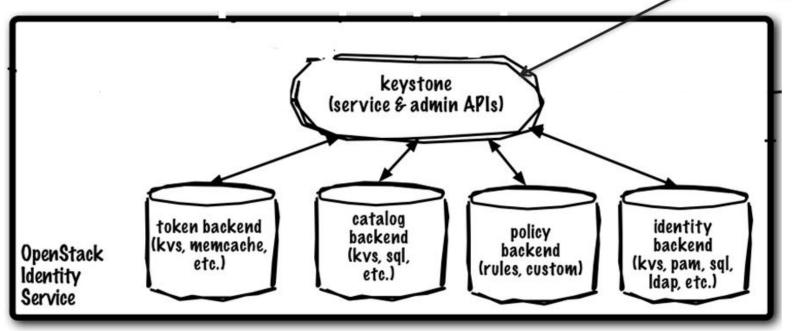
基于Django 框架实现

### Keystone身份验证服务





Keystone作为Openstack的核心模块,为Nova(计算),Glance(镜像),Swift(对象存储),Cinder(块存储),Neutron(网络)以及Horizon(Dashboard)提供认证服务



### Keystone基本概念介绍(一)



#### User

User即用户,他们代表可以通过keystone进行访问的人或程序。Users通过认证信息(credentials,如密码、 API Keys等)进行验证。

#### Tenant

Tenant即租户,它是各个服务中的一些可以访问的资源集合。例如,在Nova中一个tenant可以是一些机器,在Swift和Glance中一个tenant可以是一些镜像存储,在Neutron中一个tenant可以是一些网络资源。Users默认的总是绑定到某些tenant上。

#### Role

Role即角色,Roles代表一组用户可以访问的资源权限,例如Nova中的虚拟机、Glance中的镜像。Users可以被添加到任意一个全局的或租户的角色中。在全局的role中,用户的role权限作用于所有的租户,即可以对所有的租户执行role规定的权限;在租户内的role中,用户仅能在当前租户内执行role规定的权限。

#### Service

Service即服务,如Nova、Glance、Swift。根据前三个概念(User, Tenant和Role)一个服务可以确认当前用户是否具有访问其资源的权限。但是当一个user尝试着访问其租户内的service时,他必须知道这个service是否存在以及如何访问这个service,这里通常使用一些不同的名称表示不同的服务。

#### Endpoint

Endpoint,翻译为"端点",我们可以理解它是一个服务暴露出来的访问点,如果需要访问一个服务,则必须知道他的endpoint。因此,在keystone中包含一个endpoint模板,这个模板提供了所有存在的服务endpoints信息。一个endpoint template包含一个URLs列表,列表中的每个URL都对应一个服务实例的访问地址,并且具有public、private和admin这三种权限。public url可以被全局访问(如http://compute.example.com), private url只能被局域网访问(如http://compute.example.local), admin url被从常规的访问中分离。

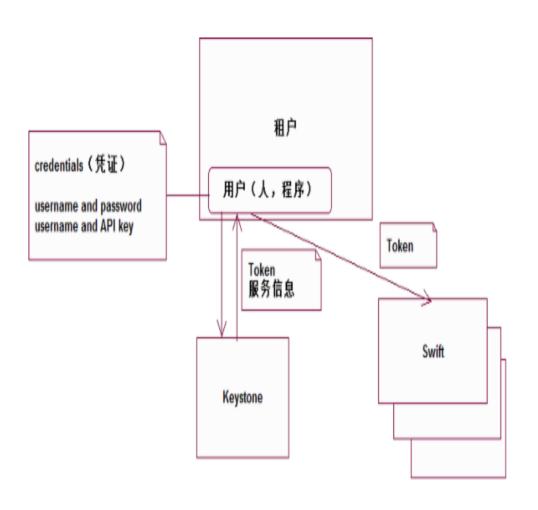
#### Token

Token是访问资源的钥匙。它是通过Keystone验证后的返回值,在之后的与其他服务交互中只需要携带Token值即可。每个Token都有一个有效期,Token只在有效期内是有效的

### Keystone基本概念介绍(二)



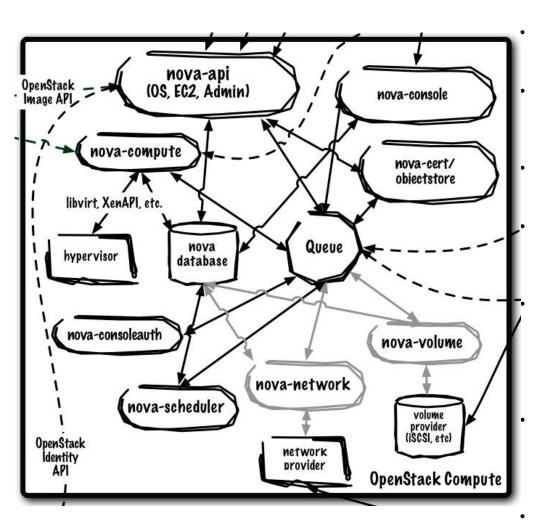
### 各种概念之间关系解释



- I、租户下,管理着一堆用户(人,或程序)。
- 2、每个用户都有自己的credentials (凭证)用户名+密码或者用户名 +API key,或其他凭证。
- 3、用户在访问其他资源(计算、存储)之前,需要用自己的credential 去请求keystone服务,获得验证信息(主要是Token信息)和服务信息(服务目录和它们的endpoint)。
- 4、用户拿着Token信息,就可以去访问特点的资源了。

### 核心模块NOVA





计算管理(codenamed "Nova") 基于用户需求为VM提供计算资源管理. 基于Python语言编写。

Nova-API:对外统一提供标准化接口.接受和响应最终用户Compute API的请求,同时还实现与Openstack其他各逻辑模块的通讯与服务提供

Nova-Scheduler:从队列上得到一个虚拟机实例请求并且 决定它应该在哪里运行(使用多种过滤器或算法调度)

Queue: 提供了一个守护进程之间传递消息的中央枢纽。 消息队列系统作用还可以实现与Openstack其他各逻辑 模块之间的通信建立连接枢纽

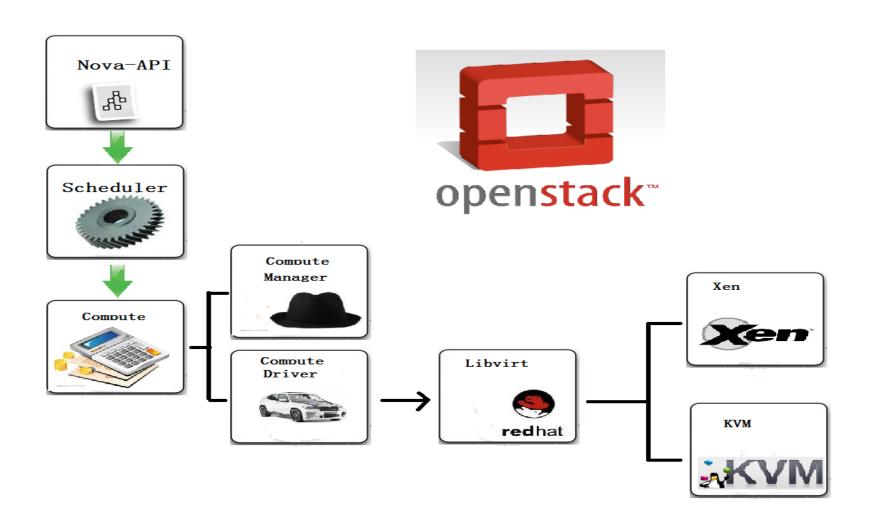
Nova-Database:存储云基础设施的编译时和运行时的状态,从理论上讲,OpenStack Nova可以支持任何SQL-Alchemy支持的数据库,但是目前被广泛使用的数据库有sqlite3(只适用于测试和开发工作),MySQL和PostgreSQL。

Nova-Compute:主要是一个人工守护进程,它可以通过虚拟机管理程序的API(XenAPI for XenServer/XCP,libvirt for KVM or QEMU, VMwareAPI for VMware等)来创建和终止虚拟机实例。支持多种虚拟化平台

Nova还提供控制台的服务,让最终用户通过代理服务器访问他们的虚拟实例的控制台。这涉及到多个守护进程(nova-console, nova-novncproxy、nova-xvpnvncproxy和nova-consoleauth)

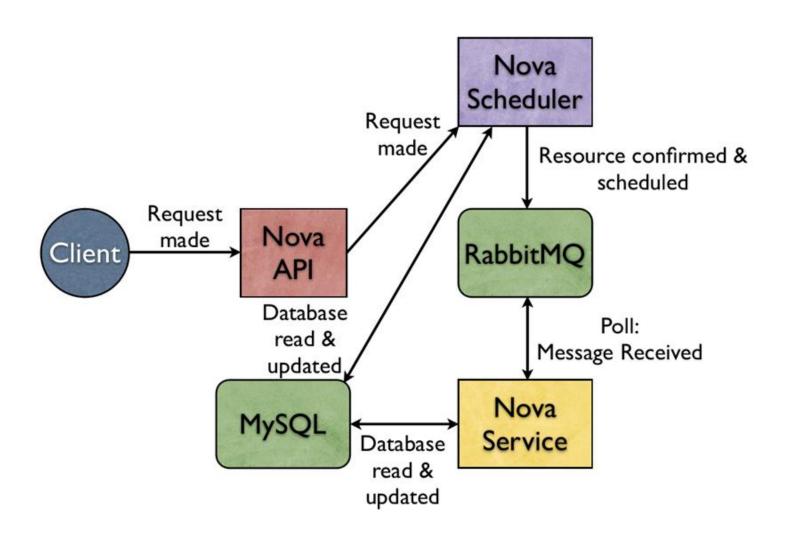
## NOVA架构





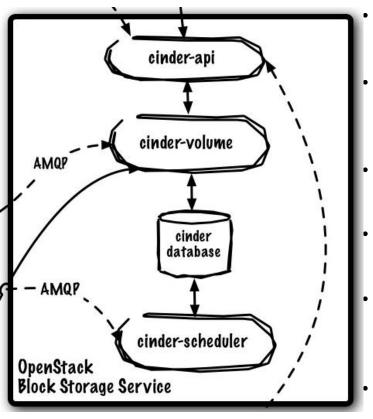
### NOVA处理过程





### 块存储Cinder





块存储管理模块(codenamed "Cinder") 提供到虚拟机的永久性块存储卷.类似AWS的EBS块存储服务

- 多个卷可以被挂载到单一虚拟机实例,同时卷可以在虚拟机实例间移动,单个卷在同一时刻只能被挂载到一个虚拟机实例
- 块存储系统管理块设备到虚拟机的创建,挂载以 及卸载.
- 块设备卷完全与Openstack Compute集成,并支持 云用户在Dashboard中管理数据自己的存储.
  - 除了支持简单的Linux服务器本地存储之外,还支持众多的存储平台,包括 Ceph, NetApp, Nexenta, SolidFire, Zadara.
  - 快照管理提供了强大的在块存储上实现数据备份 的功能可以用来作为引导卷使用
- 块存储适合性能敏感性业务场景,例如数据库存储 大规模可扩展的文件系统或服务器需要访问到块 级裸设备存储.



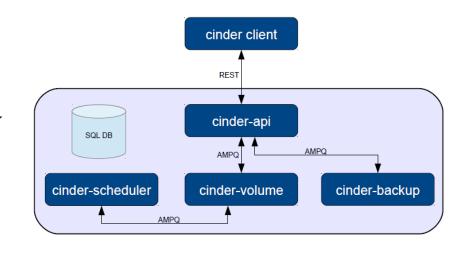
### Cinder架构解释



- ●API Service: 负责接受和处理 Rest 请求,并将请求放入 RabbitMQ对列。
- Scheduler Service: 处理任务队列的任务,并根据预定策略选择合适的 Volume Service 节点来执行任务。目前版本的 Cinder 仅仅提供了一个 Simple Scheduler, 该调度器选择卷数量最少的一个活跃节点来创建卷。
- ●Volume Service: 该服务运行在存储节点上,管理存储空间。每个存储节点都有一个Volume Service,若干个这样的存储节点联合起来可以构成一个存储资源池。为了支持不同类型和型号的存储,均通过Drivers的形式为Cinder的 Volume Service 提供相应的Cinder-Volume。

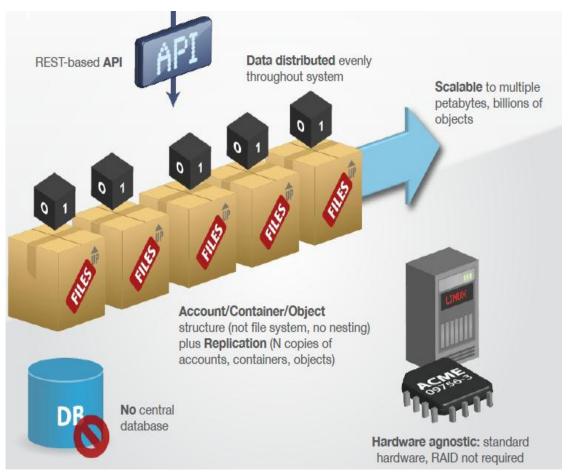
#### Cinder目前支持的存储类型

- 本地存储: LVM, Sheepdog
- 网络存储: NFS, RBD(Ceph)
- HP:3PAR (iSCSI/FC),LeftHand (iSCSI)
- IBM: Storwize family/SVC (iSCSI/FC), XIV GPFS, zVM
- Netapp: NetApp(iSCSI/NFS)
- EMC: VMAX/VNX (iSCSI), Isilon(iSCSI)
- Solidfire: Solidfire cluster(iSCSI)



### Swift 对象存储

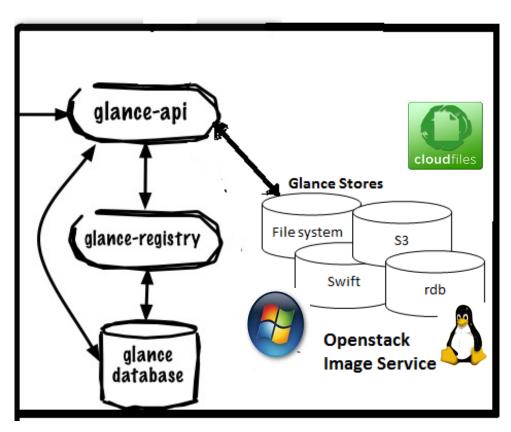




- · 分布式对象存储系统,类似于 AWS的S3
- 通过采用基于标准化服务器的 集群架构提供冗余,可扩展的 对象存储
- 具有良好的扩展性,可以实现 PB级别数据的存储
- 支持存储对象写入的多份拷贝, 并且支持当拷贝丢失后的自我 修复功能.确保数据的一致性
- 提供每GB高性价比的极佳的 可用性和数据耐久性
- 支持原生的 OpenStack™ API
   以及S3 compatible API

### Glance 镜像服务





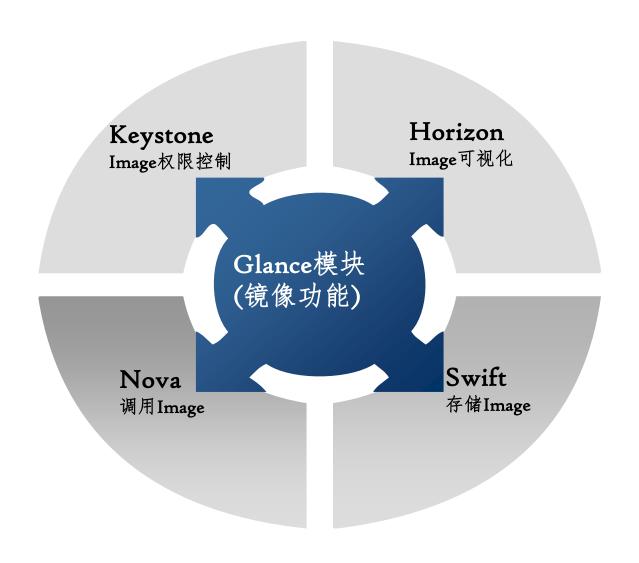
Glance是OpenStack镜像服务,用来注册、登陆和检索虚拟机镜像。Glance服务提供了一个REST API,使你能够查询虚拟机镜像元数据和检索的实际镜像。通过镜像服务提供的虚拟机镜像可以存储在不同的位置,从简单的文件系统对象存储到类似OpeenStack对象存储系统。

#### 镜像服务组件:

- Glance-API:接收最终用户或Noav对镜像的请求,检索和存储镜像的相关API调用。
- Glance-registry: 存储,处理和检索有关镜像的元数据,元数据大小、类型等等。
- Database:存储镜像元数据,可以支持多种数据库,现在使用比较广泛的是mysql和sqlite.

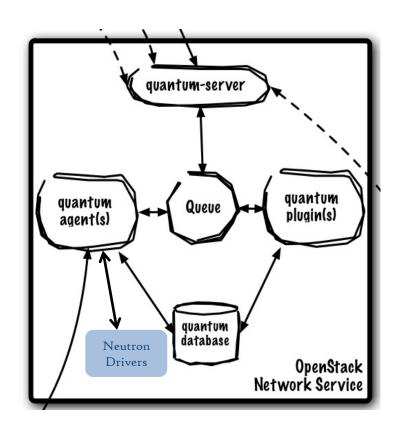
## Glance与其他组件的关系





### Neutron网络服务





- 网络服务 (codenamed "Quantum/Neutron") 提供在被管理设备之间的网络连接服务.
- 允许用户自己创建自己的网络并attach端口使用.
- 通过开发的Plugins支持SDN和OpenFlow
- 用户自定义子网地址,私有网络/公有网络以及 Floating IP分配规则

### Neutron 关键概念



### ● 网络Network

- ▶ 一个L2二层网络单元
- ▶ 租户可通过Quantum/Neutron API 创建自己的网络

### • 子网Subnet

- ▶ 一段IPV4/IPV6地址段
- ▶ 为Instance提供私网或公网地址

#### ● 路由器Router

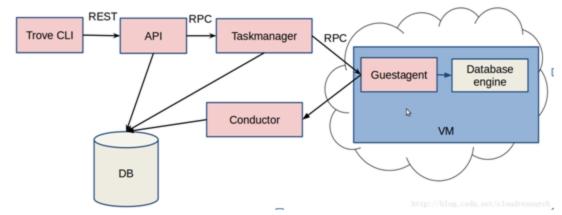
- > 三层路由器
- ▶ 为租户的Instance提供路由功能

#### ● 端口Port

- ▶ 虚拟交换机上的端口
- ▶ 管理Instance的网卡

### Trove架构说明





trove-api

提供REST风格的API,支持json和xml,程序源代码入口在trove/cmd/api.py。API server与两个组件通信,所有复杂的异步任务它都交给taskmanager去完成,有些信息它会直接从trove的后端DB中获得。

根据其配置文件api-paste.ini,可以看到起最终APP的入口是: trove/common/api.py,这就是整个trove除了 version和extension外所有资源对象操作的总入口。在API类中明确定义了url和处理的controller的路由关系,非常的清晰;

#### **Taskmanage**

负责配置数据库虚拟机实例,管理数据库虚拟机实例的生命周期和执行这对数据库虚拟机实例的各种操作. 监听消息队列服务器的RPC service。需要特别注意: trove-taskmanager是一个有状态的服务,它负责组织 复杂的业务流程。如果在有状态的处理流程当中,taskmanager挂了,则任务就被认为失败。

#### trove-guestagent

运行于数据库虚拟机实例的内部,负责管理和实际执行对数据库管理程序的任务。guestagent在消息总线上监听RPC消息,执行要求的操作。程序源代码入口在trove/cmd/guest.py。每一种数据库都有与之对应的不同的guestagent,显而易见,redis的guestagent和mysql的guestagent的行为不可能是一样的。

#### trove-conductor

负责从guestagents收集状态信息然后将其写入Trove的后端数据库,其与guestagent交互基于RPC实现。这个组件的引入就是为了实现这样的目的,即db instance内的gusestagent不应直接与trove的后端数据库相连。程序源代码入口在trove/cmd/conductor.py。

# Trove接口调用关系 (一)



| 功能名称   | RDS组件             | 作用          | 接口   | 调用对象      |
|--------|-------------------|-------------|--|-----------|
|        | trove-api         | 获取flavor对象  | trove.instance.models.instance.create  | Openstack |
|        |                   | 创建安全组       | trove.taskmanager.models.FreshInstanceTasks.c<br>reate_instance                  | Neutron   |
| 创建实例   | trove-taskmanager | 创建云硬盘       | trove.taskmanager.models.FreshInstanceTasks<br>create_server_volume_individually | Openstack |
|        |                   | 创建云主机       | trove.taskmanager.models.FreshInstanceTasks<br>create_server_volume_individually | Openstack |
|        | trove-api         | 获取flavor对象  | trove.instance.models.instance.create  | Openstack |
|        | trove-taskmanager | 创建安全组       | trove.taskmanager.models.FreshInstanceTasks.c<br>reate_instance                  | Neutron   |
| 创建只读实例 |                   | 创建云硬盘       | trove.taskmanager.models.FreshInstanceTasks<br>create_server_volume_individually | Openstack |
|        |                   | 创建云主机       | trove.taskmanager.models.FreshInstanceTasks<br>create_server_volume_individually | Openstack |
|        |                   | 删除创建的快照     | trove.taskmanager.managercreate_replication_slave                                | Swift     |
|        | trove-guestagent  | 创建主机快照(备份)  | trove.guestagent.datastore.mysql.manager.get_re<br>plication_snapshot            | Swift     |
|        |                   | 从主机快照中恢复数据库 | trove.guestagent.datastore.mysql.managerperf<br>orm_restore                      | Swift     |

# Trove接口调用关系(二)



|                  |                   | 删除云主机  | trove.taskmanager.models.BuiltInstanceTasksde<br>lete_resources                    | Openstack |
|------------------|-------------------|--|--|-----------|
|                  |                   | 检查云主机状态  | trove.taskmanager.models.BuiltInstanceTasksde<br>lete_resources.server_is_finished | Openstack |
| 终止实例             | trove-taskmanager | 删除云硬盘  | trove.taskmanager.models.BuiltInstanceTasksde<br>lete_resources                    | Openstack |
|                  |                   | 删除安全组  | trove.instance.models.BaseInstance.delete_async                                    | Neutron   |
|                  |                   | 删除实例关联的备份  | trove.instance.models.BaseInstance.delete_async                                    | Swift     |
| 实例列表             | trove-api         | 获取云主机列表  | trove.instance.models.instances.load   | Openstack |
| 实例详情概况 trove-api | 获取云主机详情           | trove.instance.models.load_simple_instance_serve<br>r_status | Openstack  |           |
|                  | 获取实例网络详情          | trove.instance.models.load_network_info                      | Neutron  |           |
| 创建备份             | trove-guestagent  | 保存备份文件到容器  | trove.guestagent.backup.backupagent.stream_backup_to_storage                       | Openstack |
| 删除备份             | trove-taskmanager | 删除容器中的备份文件   | trove.taskmanager.models.delete_files_from_swift                                   | Openstack |
|                  | trove-api         | 获取flavor对象   | trove.instance.models.instance.create  | Openstack |
|                  |                   | 创建安全组  | trove.taskmanager.models.FreshInstanceTasks.cre<br>ate_instance                    | Neutron   |
| 备份恢复             | trove-taskmanager | 创建云硬盘  | trove.taskmanager.models.FreshInstanceTaskscr<br>eate_server_volume_individually   | Openstack |
|                  |                   | 创建云主机  | trove.taskmanager.models.FreshInstanceTaskscr<br>eate_server_volume_individually   | Openstack |
|                  | trove-guestagent  | 保存备份文件到容器  | trove.guestagent.backup.backupagent.execute_rest ore                               | Swift     |

### Trove的内部RESTful交互逻辑(一)



### 1、申请keystone的Token认证:

```
curl -s -d '{"auth": {"tenantName": "admin", "passwordCredentials": \
> {"username": "admin", "password": "882f520bd67212bf9670"}}}'\
> -H 'Content-Type: application/json' http://192.168.117.5:5000/v2.0/tokens \
> | python -m json.tool
```

#### 2、keystone反馈服务的接口信息:

```
{
    "endpoints": [
    {
        "adminURL": "http://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa551
        85a122f3b",
        "id": "566ac5c0da564e88974e77504275a1a5",
        "internalURL": "http://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa
        55185a122f3b",
        "publicURL": "http://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55
        185a122f3b",
        "region": "RegionOne"
        }
        ],
        "endpoints_links": [],
        "name": "trove",
        "type": "database"
        },
        [...]
```

```
"token": {
    "audit_ids": [
    "OCg2NHfARzStxMCmYTc05A"
],
    "expires": "2015-03-18T20:34:09Z",
    "id": "4290e97568744445852699633789f92b",
    "issued_at": "2015-03-18T19:34:09.785685",
    "tenant": {
    "description": null,
    "enabled": true,
    "id": "979bd3efad6f42448ffa55185a122f3b",
    "name": "admin",
    "parent_id": null
    }
},
```

### Trove的内部RESTful交互逻辑(二)



3、根据接口信息连接相应服务:

```
curl -H "X-Auth-Token: 4290e97568744445852699633789f92b"
http://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/datastores
```

#### 4、反馈最后服务结果:

```
{"datastores": [{"default version": "79c5e43b-d7d0-4137-af20-84d75c485c56", "versions": [{"name": "5.5", "links": [{"href":
"https://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/datastores/versions/79c5e43b-d7d0-4137-af20-
84d75c485c56", "rel": "self" }, {"href": "https://192.168.117.5:8779/datastores/versions/79c5e43b-d7d0-4137-af20-84d75c485c56",
"rel": "bookmark"}], "image": "bc014926-31df-4bff-8d4e-a6547d56af02", "active": 1, "packages": "mysql-server-5.5", "id":
"79c5e43b-d7d0-4137-af20-84d75c485c56"},{"name": "inactive version", "links": [{"href":
"https://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/datastores/versions/fd8cf5a2-c456-4ba9-9ad2-
d2c4390dccfb", "rel": "self"}, {"href": "https://192.168.117.5:8779/datastores/versions/fd8cf5a2-c456-4ba9-
9ad2-d2c4390dccfb", "rel": "bookmark"}], "image": "bc014926-31df-4bff-8d4e-a6547d56af02",
{"default version": "2417d4bb-79ae-48cd-96bf-854d381466ff", "versions": [{"name": "5.5", "links": [{"href":
"https://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/datastores/versions/2417d4bb-79ae-48cd-96bf-
854d381466ff", "rel": "self"}, {"href": "https://192.168.117.5:8779/datastores/versions/2417d4bb-79ae-48cd-96bf-854d381466ff",
"rel": "bookmark"}], "image": "80137e59-f2d6-4570-874c-4e9576624950", "active": 1, "packages": "percona-server-server-5.5",
"id": "2417d4bb-79ae-48cd-96bf-854d381466ff"}], "id": "c523235f-f977-45a5-b5bd-8a3a1e486e3e", "links": [{"href":
"https://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/datastores/c523235f-f977-45a5-b5bd-8a3a1e486e3e",
"rel": "self"}, {"href": "https://192.168.117.5:8779/datastores/c523235f-f977-45a5-b5bd-8a3a1e486e3e", "rel": "bookmark"}],
"name": "percona"}]}
```

### Trove Cient的交互逻辑(一)



| ubuntu@trove-book- |      |           | +                 |        |          |          |
|--------------------|------|-----------|-------------------|--------|----------|----------|
| ID                 | Name | Datastore | Datastore Version | Status | Flavor I | )   Size |
| [instance uuid]    |      |           |                   |        | 2        |          |

```
ubuntu@trove-book-vm:~$ trove --os-username admin \
```

- > --os-password 882f520bd67212bf9670 \
- > --os-auth-url http://192.168.117.5:5000/v2.0 \
- > --os-tenant-name admin list

```
OS_TENANT_NAME=admin
OS_USERNAME=admin
ADMIN_PASSWORD=882f520bd67212bf9670
OS_PASSWORD=882f520bd67212bf9670
OS_AUTH_URL=http://192.168.117.5:5000/v2.0
```

### Trove Cient的交互逻辑(二)



```
ubuntu@trove-book-vm:~$ trove --debug list
DEBUG (session:195) REQ: curl -g -i -X -H "Accept:application/json" -H "User-Agent: python-keystoneclient"
DEBUG (retry:155) Converted retries value: 0 -> Retry(total=0, connect=None, read=None, redirect=0)
INFO (connectionpool:203) Starting new HTTP connection (1): 192.168.117.5
DEBUG (connectionpool:383) "GET /v2.0 HTTP/1.1" 200 339
DEBUG (session:223) RESP: [200] content-length: 339 vary: X-Auth-Token server: Apache/2.4.7
(Ubuntu) date: Wed, 18 Mar 2015 22:25:08 GMT content-type: application/json x-openstackrequest-id: req-66e34191-6c96-4139-b8e5-
aa9ec139ed51
RESP BODY: {"version": {"status": "stable", "updated": "2014-04-17T00:00:00Z", "mediatypes": [{"base": "application/json", "type":
"application/vnd.openstack.identityv2.0+json"}], "id": "v2.0", "links": [{"href": "http://192.168.117.5:5000/v2.0/",
"rel": "self"}, {"href": "http://docs.openstack.org/", "type": "text/html", "rel": "describedby"}]}}
DEBUG (v2:76) Making authentication request to http://192.168.117.5:5000/v2.0/tokens
DEBUG (retry:155) Converted retries value: 0 -> Retry(total=0, connect=None, read=None, redirect=0)
DEBUG (connectionpool:383) "POST /v2.0/tokens HTTP/1.1" 200 4575
DEBUG (iso8601:184) Parsed 2015-03-18T23:25:08Z into {'tz sign': None, 'second fraction':
None, 'hour': u'23', 'daydash': u'18', 'tz hour': None, 'month': None, 'timezone': u'Z',
'second': u'08', 'tz minute': None, 'year': u'2015', 'separator': u'T', 'monthdash':
u'03', 'day': None, 'minute': u'25'} with default timezone <iso8601.iso8601.Utc object at0x7fb7247648d0>[...]
DEBUG (session:195) REQ: curl -g -i -X GET http://192.168.117.5:8779/v1.0/979bd3efad6f42448ffa55185a122f3b/instances -H "User-
Agent: python-keystoneclient" -H "Accept: application/
json" -H "X-Auth-Token: {SHA1}7a8693593cc9fa0d612b064041515fc8a31c4d7a"
DEBUG (retry:155) Converted retries value: 0 -> Retry(total=0, connect=None, read=None,
redirect=0)
INFO (connectionpool:203) Starting new HTTP connection (1): 192.168.117.5
DEBUG (connectionpool:383) "GET /v1.0/979bd3efad6f42448ffa55185a122f3b/instances HTTP/1.1"
200 17
DEBUG (session:223) RESP: [200] date: Wed, 18 Mar 2015 22:25:09 GMT content-length: 17
content-type: application/ison
RESP BODY: {"instances": []}
```

### Trove基础操作——创建MySQL实例



```
ubuntu@trove-book-vm:~$ trove datastore-list
                                     l Name
 170c8cce-15c0-497c-a148-d9120cc0630f | mysql
 c523235f-f977-45a5-b5bd-8a3a1e486e3e | percona
ubuntu@trove-book-vm:~$ trove datastore-version-list 170c8cce-15c0-497c-a148-d9120cc0630f
79c5e43b-d7d0-4137-af20-84d75c485c56 | 5.5
ubuntu@trove-book-vm:~$ trove datastore-version-list c523235f-f977-45a5-b5bd-8a3a1e486e3e
| ID
2417d4bb-79ae-48cd-96bf-854d381466ff | 5.5
ubuntu@trove-book-vm:~$ trove create m1 2 --size 3 --datastore mysql --datastore version 5.5
 Property
                | Value
 created
                 2015-03-18T18:14:43
 datastore
                   mysql
 datastore_version |
                     5.5
 flavor
 id
                     d92c7a01-dc16-48d4-80e0-cb57d8a5040a
 name
                     m1
 status
                     BUILD
 updated
                     2015-03-18T18:14:43
 volume
```

### Trove基础操作——创建数据库



```
ubuntu@trove-booktrove create m6 2 --size 2 --datastore mysgl --databases \
> chapter_3_db_1 chapter_3_db_2 --users user1:password1 user2:password2
                    | Value
 Property
 created
                     2015-03-18T18:19:55
 datastore
                     mysql
 datastore version
                     5.5
 flavor
 id
                     64f32200-9da1-44af-b6c6-5e5b01ec0398
  name
 password
                     xQ6wyJCUZzhjVrkNeRtQpCYeh2XcQEfbY8Cf
 status
                     BUILD
 updated
                     2015-03-18T18:19:55
  volume
```

Once the instance comes online you can list the databases.

### Trove基础操作——重启/删除MySQL实例



ubuntu@trove-book-vm:~\$ trove show 64f32200-9da1-44af-b6c6-5e5b01ec0398

| Property  | Value  |
|---|--|
| created datastore datastore_version flavor id ip name status updated volume volume_used | 2015-03-18T18:19:55<br>  mysql<br>  5.5<br>  2<br>  64f32200-9da1-44af-b6c6-5e5b01ec0398<br>  10.0.0.3<br>  m6<br>  ACTIVE<br>  2015-03-18T22:00:15<br>  2<br>  0.17 |
| +   | <del></del>  |

ubuntu@trove-book-vm:~\$ trove delete 64f32200-9da1-44af-b6c6-5e5b01ec0398

ubuntu@trove-book-vm:~\$ trove snow b4t32200-yda1-44at-bbcb-5e5b01ec0398

| Property  | Value   |
|---|---|
| created datastore datastore_version flavor id ip name status updated volume | 2015-03-18T18:19:55<br>mysql<br>5.5<br>2<br>64f32200-9da1-44af-b6c6-5e5b01ec0398  <br>10.0.0.3<br>m6<br>REBOOT<br>2015-03-18T21:59:56 |
| +   | ++  |

### Trove基础操作——用户管理

user1 | % | chapter\_3\_db\_1, chapter\_3\_db\_2
user2 | % | chapter\_3\_db\_1, chapter\_3\_db\_2



```
ubuntu@trove-book-vm:~$ trove user-create 64f32200-9da1-44af-b6c6-5e5b01ec0398 \
> user3 password3 --databases chapter 3 db 1
ubuntu@trove-book-vm:~$ trove user-list 64f32200-9da1-44af-b6c6-5e5b01ec0398
 Name | Host | Databases
 user1 | % | chapter 3 db 1, chapter 3 db 2
 user2 | % | chapter_3_db_1, chapter_3_db_2
 user3 | % | chapter_3_db_1
ubuntu@trove-book-vm:~$ trove user-delete 64f32200-9da1-44af-b6c6-5e5b01ec0398 user3
ubuntu@trove-book-vm:~$ trove user-list 64f32200-9da1-44af-b6c6-5e5b01ec0398
 Name | Host | Databases
```

### Trove高级功能——备份



```
ubuntu@trove-book:~$ trove backup-create ed77ec23-6444-427d-bc0c-56a8b202974e \
> backup-1-incremental \
> --parent ff6398a6-e15f-4063-8c66-58f3a8528794 --description "make an incremental backup"
                Value
  Property
  created
                2015-04-19T22:07:16
  datastore
                {u'version': u'5.6', u'type': u'mysql',
                u'version_id': u'a5ec21dc-3a5b-41c8-97fe-4fc3ed007023'}
               make an incremental backup
 description
  id
               496589a6-3f82-4ccc-a509-91fbfc2a3091
  instance id | ed77ec23-6444-427d-bc0c-56a8b202974e
  locationRef |
               None
                backup-1-incremental
  name
                ff6398a6-e15f-4063-8c66-58f3a8528794
  parent id
  size
                None
                NEW
  status
  updated
                2015-04-19T22:07:16
```

# Trove高级功能——恢复



ubuntu@trove-book:~\$ trove create from-backup 2 --size 2 --backup 496589a6-3f82-4ccc-a509-91fbfc2a3091

| +  | ++   |
|--|--|
| Property   | Value  |
| created datastore datastore_version flavor id name status updated volume | 2015-04-19T22:29:34<br>mysql<br>5.6<br>2<br>f62e48fa-33f7-4bb4-bf2d-c8ef8ee4cb42<br>from-backup<br>BUILD<br>2015-04-19T22:29:34<br>2 |
| T  | r  |

### Trove高级功能——主从复制(一)



### 1、查看实例状态:

ubuntu@trove-book-vm:/opt/stack/trove\$ trove show 41037bd2-9a91-4d5c-b291-612ad833a6d5

| 4   | ·  |
|---|--|
| Property  | Value  |
| created datastore datastore_version flavor id ip name status updated volume volume_used | 2015-03-24T19:33:07<br>  mysql<br>  5.5<br>  2<br>  41037bd2-9a91-4d5c-b291-612ad833a6d5<br>  172.24.4.4<br>  m2<br>  ACTIVE<br>  2015-03-24T19:33:12<br>  2<br>  0.17 |
| +   | <del> </del>   |

### 2、创建复制实例:

ubuntu@trove-book-vm:/opt/stack/trove\$ trove create m2-mirror 2 --size 2 \
> --replica\_of 41037bd2-9a91-4d5c-b291-612ad833a6d5

| Property  |  |
|---|--|
| created datastore datastore_version flavor id name replica_of status updated volume | 2015-03-24T19:49:50<br>  mysql<br>  5.5<br>  2<br>  091b1a9b-8b0c-44fc-814e-ade4fca4e9c6<br>  m2-mirror<br>  41037bd2-9a91-4d5c-b291-612ad833a6d5<br>  BUILD<br>  2015-03-24T19:49:50<br>  2 |

# Trove高级功能——主从复制(二)



### 3、查看从库状态:

ubuntu@trove-book-vm:/opt/stack/trove\$ trove show 091b1a9b-8b0c-44fc-814e-ade4fca4e9c6

| 4  | L   |
|--|---|
| Property   | Value   |
| created datastore datastore_version flavor id ip name replica_of status updated volume volume_used | 2015-03-24T19:49:50<br>mysql<br>5.5<br>2<br>091b1a9b-8b0c-44fc-814e-ade4fca4e9c6<br>172.24.4.5<br>m2-mirror<br>41037bd2-9a91-4d5c-b291-612ad833a6d5<br>ACTIVE<br>2015-03-24T19:50:23<br>2<br>0.18 |
| T  |   |

mysql> show master status;

| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
|------------------|----------|--------------|------------------|
| mysql-bin.000001 | 332      |              | İ                |

1 row in set (0.00 sec)

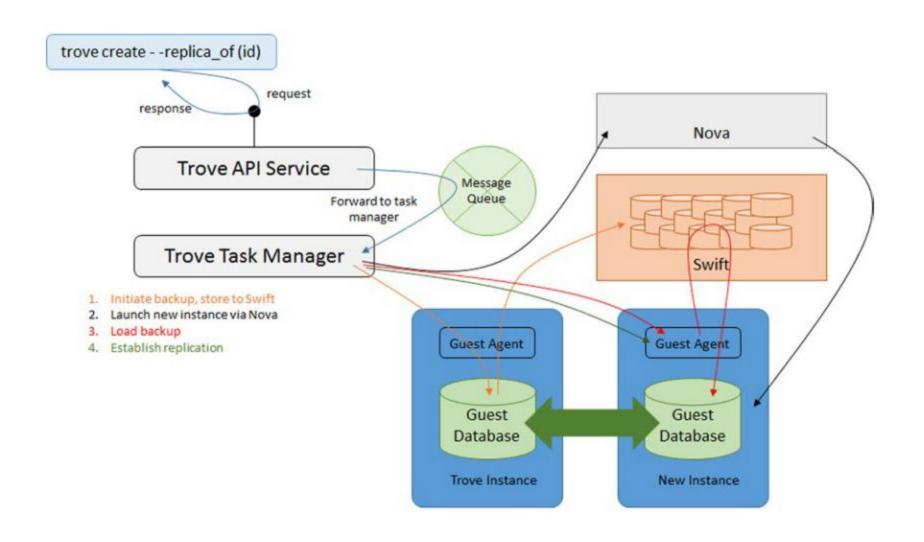
mysql> show slave hosts;

|            | Host | Port | Master_id |
|------------|------|------|-----------|
| 1236306658 | ĺ    | 3306 | 506806179 |
| +          | +    |      | ++        |

1 row in set (0.00 sec)

## Trove高级功能——主从复制实现





### Trove高级功能——Failover



```
ubuntu@trove-book:~$ trove show ed77ec23-6444-427d-bc0c-56a8b202974e
 Property
                    | 2015-04-09T11:44:42
 created
 datastore
                    | mysql
| datastore version | 5.6
| flavor
| id
                    | ed77ec23-6444-427d-bc0c-56a8b202974e
                    1 172.24.4.3
| ip
| name
                    1 m1
 replicas
                    | 22ac3792-0dd4-4b55-b4b4-b36a8a221ba8,
                    | 7ac6d15f-75b8-42d7-a3d8-9338743cc9b7,
                    | c9643f1c-8393-46cd-acb5-6fe85db16f0a
| status
                    I ACTIVE
 updated
                    | 2015-04-09T11:44:56
| volume
                    1 2
                    1 0.11
 volume used
ubuntu@trove-book:~$ trove promote-to-replica-source m14
ubuntu@trove-book:~$ trove show m1
                                                             ubuntu@trove-book:~$ trove show m14
 Property
                    | Value
                                                            | | Property
| id
                    | ed77ec23-6444-427d-bc0c-56a8b202974e | | created
                                                                                  | 2015-04-19T23:57:33
| ip
                    1 172.24.4.3
                                                            | | datastore
                                                                                | mysql
| name
                    | m1
                                                            | | datastore version | 5.6
| replica of
                    | 22ac3792-0dd4-4b55-b4b4-b36a8a221ba8 | | flavor
| status
                    | ACTIVE
                                                           | | id
                                                                                  | 22ac3792-0dd4-4b55-b4b4-b36a8a221ba8
| updated
                   | 2015-04-20T00:50:15
                                                           | | ip
                                                                                  1 172.24.4.7
| volume
                    1 2
                                                            | | name
                                                                                  I m14
 volume used
                                                           | | replicas
                                                                                  | 7ac6d15f-75b8-42d7-a3d8-9338743cc9b7,
                                                                                  | c9643f1c-8393-46cd-acb5-6fe85db16f0a,
                                                                                  | ed77ec23-6444-427d-bc0c-56a8b202974e
                                                              | status
                                                                                  | ACTIVE
                                                                                  | 2015-04-20T00:50:15
                                                              | updated
                                                               volume
                                                                                  1 2
                                                               volume used
                                                                                  1 0.12
```

# Trove高级功能——实例的弹性扩容



### 1、存储扩容

| .buntu@trove-book:~\$ | trove show m2                        | u   | abuntu@trove-book:~\$ |                                      |
|-----------------------|--------------------------------------|-----|-----------------------|--------------------------------------|
| Property              | Value                                | 1 1 | Property              | Value                                |
|                       | 2015-04-24T11:20:36                  |     | created               | 2015-04-24T11:20:36                  |
| datastore             | mysql                                | 1 1 | datastore             | mysql                                |
| datastore_version     | 5.6                                  |     | datastore_version     | 5.6                                  |
| flavor                | 2                                    | 1 1 | flavor                | 2                                    |
| id                    | 53796f4d-b0d8-42f1-b9c6-82260a78150f | 1 1 | id                    | 53796f4d-b0d8-42f1-b9c6-82260a78150f |
| ip                    | 10.0.0.2                             |     | ip                    | 10.0.0.2                             |
| name                  | m2                                   |     | name                  | m2                                   |
| status                | ACTIVE                               |     | status                | ACTIVE                               |
| updated               | 2015-04-24T12:22:50                  |     | updated               | 2015-04-24T12:29:37                  |
| volume                | 2                                    |     | volume                | 4                                    |
| volume used           | 0.11                                 | 1 1 | volume used           | 0.11                                 |

### 2、性能扩容

| ubuntu@trove-book:~\$ trove show m2 |                                      | ubuntu@trove-book:~\$ trove resize-instance m2 3 ubuntu@trove-book:~\$ trove show m2 |                                     |
|-------------------------------------|--------------------------------------|--|-------------------------------------|
| Property                            | Value                                | Property   | +<br>  Value                        |
| created                             | 2015-04-24T11:20:36                  | + +<br>     created  | 2015-04-24T11:20:36                 |
| datastore                           | mysql                                | datastore  | mysql                               |
| datastore version                   | 5.6                                  | datastore version  | 5.6                                 |
| flavor                              | 2                                    | flavor   | 3                                   |
| id                                  | 53796f4d-b0d8-42f1-b9c6-82260a78150f |  | 53796f4d-b0d8-42f1-b9c6-82260a78150 |
| ip                                  | 10.0.0.2                             |  | 10.0.0.2                            |
| name                                | m2                                   | name   | m2                                  |
| status                              | ACTIVE                               | status   | ACTIVE                              |
| updated                             | 2015-04-24T12:29:37                  | updated  | 2015-04-24T12:40:09                 |
| volume                              | 4                                    | volume   | 4                                   |
| volume_used                         | 0.11                                 | volume_used  | 0.11                                |

### Trove的安装测试



1、Openstack的安装(自动化部署)

多节点环境:

Fuel for Openstack: <a href="https://wiki.openstack.org/wiki/Fuel">https://wiki.openstack.org/wiki/Fuel</a>

Packstack: <a href="https://www.rdoproject.org/install/quickstart/">https://www.rdoproject.org/install/quickstart/</a>

单节点环境:

Devstack: <a href="http://docs.openstack.org/developer/devstack/">http://docs.openstack.org/developer/devstack/</a>

2、Trove组件的安装(自动化部署)

trove-integration: <a href="https://github.com/openstack/trove-integration">https://github.com/openstack/trove-integration</a>

3、推荐测试方式:

Devstack+手动部署Trove