Synnefo - Feature # 829

Status:	Assigned	**Priority:**	Low
Author:	Vangelis Koukis	**Category:**	Cyclades API
Created:	07/15/2011	**Assignee:**	Giorgos Gousios
Updated:	07/15/2011	**Due date:**	
Subject:	Garbage collection for VirtualMachine and Image instances		

Description

Whenever a user does DELETE /servers/id, the corresponding instance of the VirtualMachine model doesn't get deleted, rather it gets marked with deleted=True.
This happens, among other things, to support differential GETs with the ?changes-since parameter.

A management command must be written, to be run periodically and actually delete VirtualMachine and Image instances which have been deleted more than e.g., 30' ago, and are no longer referenced by any other object.

History

#1 - 07/15/2011 10:06 am - Giorgos Gousios

I don't think that deleting data is a good idea, for various reasons, 2 of which might be the following

-What happens if we want to trace back some strange behavior?
-What happens if the user wants to see his wallet history?

#2 - 07/15/2011 10:22 am - Vangelis Koukis

These are all valid questions.
If there is no garbage collection mechanism, does that mean the DB gets to contain information on **every** VM, **ever** created in the system, **for ever**? This does not scale. There has to be a configurable data retention period, after which terminated VMs will be collected.

- Tracing strange behavior will have to happen using logs, if this period has expired. The retention period of logs is completely unrelated and may be much longer.
- The wallet history, especially searchable, detailed information on debit and credit operations will be able to go up to a certain point back in the past. Even banking institutions only allow the user to view detailed transaction lists for the past one or two months.
- The retention policy for instances of the Debit and Credit or similar models in the DB may be entirely different altogether. The user may be able to see a charge for VM 12345, without the VM instance with id=12345 actually being there any more.

#3 - 07/15/2011 10:51 am - Giorgos Gousios

I think we are optimising prematurely here.

> *If there is no garbage collection mechanism, does that mean the DB gets to contain information on **every** VM, **ever** created in the system, **for ever**? This does not scale.*

How can you be sure about that? Let's say you have 30000 users, each one creating 3 VMs every day for 3 years. How many rows do you get in db_virtualmachine? 32.400.000. Of which, only the last few days' worth of VMs (100000?) will need to be kept hot in memory-based indices. Most serious DBMSs should laugh in the face of such numbers.

> *There has to be a configurable data retention period, after which terminated VMs will be collected.*

True, but after we face scalability issues. At the moment, our problem is to get users to use the service. Speaking of data retention, are there any laws affecting our service on that front?

> *- Tracing strange behavior will have to happen using logs, if this period has expired. The retention period of logs is completely unrelated and may be much longer.*

So you effectively exchange high quality, structured, interconnected data, with low quality unconnected data...

> *- The wallet history, especially searchable, detailed information on debit and credit operations will be able to go up to a certain point back in the past. Even banking institutions only allow the user to view detailed transaction lists for the past one or two months.*

6 months, but banks do face scaling problems with the millions of transactions they run per day (+they keep copies of all data in offline DBs). We don't.

> *- The retention policy for instances of the Debit and Credit or similar models in the DB may be entirely different altogether. The user may be able to see a charge for VM 12345, without the VM instance with id=12345 actually being there any more.*

This is not true in a correctly structured db schema. If we delete a VM, referential integrity mechanisms should delete all related rows.

#4 - 07/15/2011 11:03 am - Vangelis Koukis

- Target version deleted (v0.7)

I agree on almost all of the above points.
It is a matter of sizing, and it's true we don't really have any data on this now.

Let's leave this ticket open, and deal with it when there seems to be a real need for such mechanism.

The assumption that the user will create in the order of 3 VMs/day holds true with the current setup, and may only change when VM creation becomes really cheap (cloneable snapshots). Then, in theory at least, the user may request VMs in the order of 1000s per day, programmatically, e.g. for HPC.