Cloud Operator Interviews: Quota Management

September 2016

Danielle Mundle
Piet Kruithof
Shamail Tahir
Carol Barrett

openstack UX

Powered by SurveyMonkey

What are quotas?

Quota definition according to OpenStack

“Quotas are operational limits.”

*“To prevent system capacities from being exhausted without notification, you can set up quotas. Quotas are operational limits. For example, the number of gigabytes allowed for each tenant can be controlled so that cloud resources are optimized. Quotas can be enforced at both the tenant (or project) and the tenant-user level.” **

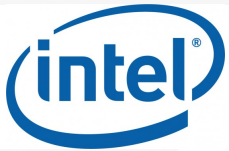
- Implemented in Cinder, Neutron, and Nova
- Come preset with a default value upon install
- OpenStack setup influences quota pain points

* http://docs.openstack.org/admin-guide/dashboard-set-quotas.html

Study Design

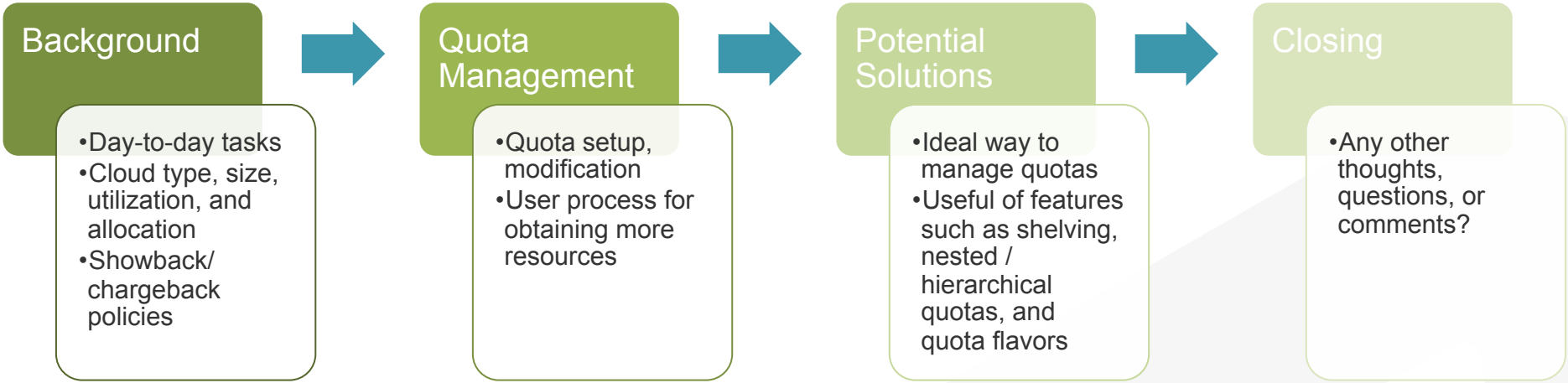
Goal, method, and participants

- **Goal:** Understand the difficulties of quota management and scope solutions based on operator feedback
- **Method:** Semi-structured individual interviews lasting 45 minutes each
  - Topics included cloud environment and utilization, quota setup and modification, pain points of quota management, and potential solutions for streamlining quota-related tasks and considerations.
- **Participants:** Nine (9) operators spanning a mix of organizations and cloud configuration
  - Organizations represented included CERN, PNNL, GoDaddy, WorkDay, Intel, SWITCH, academia (universities), and government clouds



Study Detail

Method Detail: Flow



Persona*: Rey the Cloud Operator

Rey is involved in installing, operating, using, and updating the OpenStack cloud services. Rey ensures that the cloud is up and running and must fix any issues as soon as possible. Collaborating with unskilled IT personnel is very challenging for Rey.

* http://docs.openstack.org/contributor-guide/ux-ui-guidelines/ux-personas.html

Executive Summary

Cloud Operator Interviews: Quota Management

- While participants communicated different needs and ideas for improving quota management, two main themes emerged – automation of processes and increased user control
  - As quota management involves tedious tasks, automation frees operators to handle more complex issues
  - When users provide details surrounding their quota request, it streamlines the approval process

> *"Quotas are inconsistent. They overlap and differ across projects in the way the exceptions are set. In particular, it would be nice if Nova and Neutron referenced the same values in a shared database."*

> *"Quotas are a very boring task. We manage things through Horizon, which requires many steps. I need to assign myself as an admin to see a user's resources and calculate an increase from there."*

- Potential solutions include a "one-stop quota management shop" and dashboard for operators considering user needs from a business perspective
  - Dashboard for quota across projects to understand high-level user needs at a glance

Main Findings

Automation in Quota Management

"All personal projects are all configured the same and are 'free' to people with appropriate need. They get allocated automatically. If you have an idea, you shouldn't have to wait to get started."

Participants described different levels of automation between their quota management strategies; however, all consistently desired further automation.

- A few participants manage quotas entirely manually, and saw benefit in having automated tools for certain quota allocation and modification tasks
  - Automatic setup of new projects that have a default resource allocation
- Others who already leveraged automation for allocation/modification felt cross-project quota management could be more streamlined
  - Inconsistency between projects was the biggest driver of this need – most used individual scripts for each quota, although they ideally wanted to have one solution
  - Use of the OpenStackClient could potentially help solve this issue

User Control in Quota Management

"I'd like it all to be self-service, even quota modification. Using tickets creates a delay that doesn't have to be there. Tickets are still a manual process, and it doesn't have to be that way."

Delegating tasks to users helped offset some of the tedious tasks that participants undergo.

- Although generally not seen as optimal solutions, ticketing systems were commonly implemented for user-generated quota requests
  - This reduces back-and-forth communication, as users include detailed information (such as project details, intended use, and other resource justification) on the ticket
  - Allocation decisions are easier for operators to make when more information is provided
- Participants also relied on users to monitor their own resources to know when their project is close to capacity
  - Those that monitor resources found this laborious when done manually, and desired an algorithm or heuristic to aid them with this task

Most Difficult Aspect of Quota Management

"The biggest problem is inconsistency. There are overlapping quotas in some projects and then none in Glance."

Overall, business-related decisions (such as considering user requests) and dealing with quota inconsistencies were considered the hardest things about quota management.

- Balancing over-allocation risk with customer satisfaction was generally challenging
  - Declining quota modification requests, balancing quotas by region, and avoiding customer "bill shock"
  - Those that lacked a showback process often felt that their growth was uncontrolled
- Manual processes, such as keeping Nova quotas in sync, start as small issues but compound into a larger management problem
  - Mismatches in the system are easy to fix but annoying to having to keep addressing
  - The ability to fine-tune on a more granular level and receive more feedback were considered ways to mitigate this problem

Additional Findings

Environment Details

Participants indicated the following when describing their cloud environment:

- All participants managed at least one production cloud, typically operating around 80% capacity
    - Many described a concern for future lack of resources if loose allocation habits continue – some participants mentioned they have cut back default quota to compensate
- The number of projects varied widely, with some as few as 62 to as high as 3600
- Active users ranged anywhere from 100 to 2500+
    - The ratio of users to projects changed depending on
        - Whether users were automatically given “personal” projects (= more projects than users)
        - Whether “work” projects were accessed by all project members instead of one admin (= more users than projects)
        - Whether users are part of more than one project (= more projects than users)

Chargeback and Showback

"Once a month we aggregate project and resource information for the business unit owner. They see what their quota is and what they are using. Actual utilization inside each VM would be nice, though."

Participants felt the main benefit of showback and chargeback is to manage "power users."

- Less than half of participants had a showback system in place, and most others mentioned they would like to provide some kind of usage feedback to their users.
  - Participants with diminishing resource availability often leverage showback to keep heavy cloud users' consumption in check
  - Others participants said they "just try to be fair" in allocations or give feedback in less formalized ways
- Fewer participants' environments afforded the use of chargeback, but of those that did, strategies differed
  - One approach used a "bucket" metaphor: buying a known quantity upfront to avoid management
  - Another approach tracked resources: sending feedback to users to visualize their used resources
  - When reaching billing decisions, one participant recounted the struggle between the business team's desire to charge by quota and the engineering team's suggestion to charge by actual resource use

Nesting / Hierarchy of Tenants or Projects

"We have nested projects in Keystone. This feature is important to users because experimenters want to manage their own quota. Sub-projects get 'spike-y' and it's easy for them to adjust priorities."

Although only one participant leveraged nested quotas, most others saw potential benefit.

- Most thought nested quotas would be too complicated to implement and/or only benefit a smaller subset of users, making the effort impractical in their current setup
  - Nested quotas are not possible to implement in older versions of OpenStack
  - Potential use cases involved: Maintaining a relation between two separate projects (such as a test and production version of an app), allocating quota to a department that divides resources between owned projects, and allocating quota to a short-term project (such as students working for a semester)
- Larger clouds were considered the ideal candidate for nested quotas
  - Further subdividing large projects increases admin control of resource delegation
  - One participant reported using nested quotas in Keystone, and ideally wanting the feature supported in Nova

Allocating Quotas

"We have a template where users provide information on a new request ticket. This request gets peer reviewed and approved by management, and then Jenkins automatically pushes the resources."

The process of allocating quota was seen as a balancing act between making it easy to get started on a project, but keeping serious resource use in check.

- Many participants allocate a small default amount of resources to users in the form of a "personal" project to get started with
  - Some of these projects are allocated automatically with Keystone provisioning, others are created manually after the operator receives a ticket request
- "Work" or "group" projects were typically larger and underwent a longer review process
  - Admins/project leads communicate resource need to operators via email, in-house app, or ticketing system
  - A bot or API typically handles larger project creation if not done manually through the CLI

Modifying Quotas

"Modification requests come in only a few times per month. Typically they're from users who already have a high quota, and they want more."

In comparison to other quota activities, requests for modifying quotas happen less frequently.

- Although infrequent, modifying quotas seemed more complicated than creating a new project
  - Due to cross-project considerations, some participants mentioned creating scripts and other custom solutions to ensure proper quota handling across all projects
  - Using the CLI was the most common modification solution, but was often paired with scripts
- A few participants mentioned more unique methods for quota modification
  - Two mentioned that users manage their project in Horizon
    - In one case, users perform tasks such as viewing request history or placing a new resource request, and a django allocation app provisions quota automatically using a bot
    - In the other, Horizon takes the user to the help desk to complete their request
  - One other leverages a Nova scheduler ratio to modify quotas by dispatching compute requests

Shelving

"When you shelve resources, you free up resources to another project. But the problem is that you're not physically released. I set up a test cloud with Mitaka and I'm using shelving a lot."

Participants' feelings toward the practice of "shelving" unused resources were mixed.

- About half the participants were interested in or actively trying to learn about shelving
  - Using an earlier version of OpenStack was the most frequent reason for a lack of implementation – many were using Kilo or Liberty and thought it might be a feature for Mitaka
  - Those that supported shelving noted it's utility for suspending VMs and potential for saving on resources
- Others failed to see significant incentive for users to use shelving, or simply believed the solution would not fit in the context of their environment
  - Users were perceived to have less motivation to shelve their resources when not in use
    - Returning resources may not have a direct benefit to users
  - IP addresses were an issue for some participants
    - The fixed number of IP addresses would get exhausted as the shelved resources still use them

Resource Management

"After 90 days, the system will retain any resources actively used plus 20% more – the rest will get taken away. This 'resource reclamation' is a known concept to users so there is no warning. If they find they need more, we'll just add it back later."

While almost half of the participants mentioned using a process to reclaim unused resources, others often struggled with managing resources effectively.

- Those that reclaim resources described a script which checks for inactive VMs, and often paired with a warning to the user, deletes them after a certain timeframe
  - Typically the period of inactivity was around 30-90 days; some offered a chance to extend inactive use
  - Although a few mentioned such a "policing system" was unfavorable with users, others explained it was an accepted part of the system
- Others that did not have a formal management strategy in place described other ways to manage resources
  - Some over-allocated so users rarely hit a ceiling, or relied on "good Samaritans" to return unused VMs
  - Others who didn't monitor relied on the quota itself to protect against cloud overuse ("runaway scripts")

Cross-Project Quota Management

"It's terrible. If someone requests floating IP in Nova, you need to change it in both Nova and Neutron. The UI finds the overlap, but if you use the API there is a mismatch."

Unless they used only one or two projects that involve quotas, participants found cross-project quota management one of the hardest aspects overall.

- Mismatch of quota between Nova and other projects was problematic, particularly for those that used the CLI or individual project scripts to make quota modifications
  - A discrepancy between the quota displayed to the user and the quota available on the system backend was a common issue
  - Some mentioned that it is not too difficult to solve this problem, but should ultimately not be an issue
- Aside from inconsistency between projects, a lack of quota in Glance (used for image services) also emerged as a pain point
  - Participants ideally would like to set the total capacity and the number of snapshots, know the number of metadata on an image, and set per-user/per-project values in Glance

Potential Solutions

Potential Solutions

"The delimiter library looks promising. It may solve our sync problems and help us implement quotas."

Solutions believed to bring about the most benefit to operators include:

1. Streamlining quota management via a centralized tool kit
    - This may improve inconsistent semantics through use of APIs, the delimiter library, and the OpenStack Client. It also can help automate tasks that would otherwise be manual

2. Delegating control to domain ops
    - Delegating quota management to domain ops allows them to allocate via embedded projects

3. Offering quota flavors, and implementing a UI for this feature for cloud ops
    - Consistency across quota flavors will add extra benefit on top of the tool kit's consistency within the code

Other Potential Solutions

"I would like quota on host aggregates, or flavors. Quota is just on instances, and that assumes that one instance is the same as another, and they're not."

Additional participant ideas for quota management improvement:

- Flexibility to use "shelved" ephemeral VMs to do a burst of extra data crunching
  - These "floating" VMs would be returned to their original project when needed
- Provide quota on host aggregates
  - Quota remains global at scale, even across a large OpenStack cloud that is otherwise partitioned

Thanks!

openstack UX

Powered by SurveyMonkey