January 2023

# Customizable Cloud Clusters

Diwakar Gupta

Alex Matute

# Customizable Cloud Clusters

ABSTRACT

In cloud computing, sole tenant nodes dedicate hardware to a single customer, enabling the customer to tightly optimize utilization and performance to match their load profile. However, with sole tenancy, the burdens of capacity management, risk management, task scheduling, triggering live-migration, etc., fall on the customer. This disclosure describes techniques to enable customers to run their workloads in standard, multitenant cloud computing environments while still being able to customize the experience to levels comparable to those offered via dedicated hardware or sole tenant hosts. The high level of customization in a multitenant environment is enabled even as the customer uses the familiar API set of the public cloud. Benefits of single tenancy such as load profile optimized overcommit, custom placement, autoscaling, etc., accrue without the burdens of capacity management, task scheduling, risk management, etc. as the workloads enjoy the better statistical multiplexing offered by the standard, multitenant public cloud.

KEYWORDS

- Virtual private cloud (VPC)
- Sole tenancy
- Overcommit
- Live migration
- Managed instance group (MIG)
- Instance template
- Compact placement
- Input/output operations (IOPs)

## BACKGROUND

Cloud computing services offer various products that enable sophisticated customers to optimize the utilization and performance of their cloud resources, often grouped as clusters. For example:

- Sole tenant nodes (dedicated hosts or dedicated instances) dedicate hardware to a single customer that is free to run multiple virtual machines (VM) on that hardware, effectively forming a virtual private cloud and enabling the customer to tailor their overcommits to their load statistics. Overcommitting is a technique that enables a physical host machine to take on a greater number of virtual machines than suggested by the availability of its physical computational resources. Overcommitting works due to statistical multiplexing; different VMs experience load and resource demand spikes at different times.
- Compact placement enables customers to place nodes and workloads in close physical proximity to reduce network latency.
- A managed instance or autoscaling group enables customers to manage bulk instances by treating a group of virtual machines as a single entity based on an instance template.

While enabling customers to achieve high performance and utilization, current cloud computing setup causes the burden of cloud management to fall on the customer. For example, in single tenancy mode, the customer can bring their own hypervisor, tailor overcommits to their load profile, or otherwise customize the machine to suit specific business needs; however, the overheads of capacity management (expansion or contraction), task scheduling, bin packing, pricing, etc. fall on the customer. Also, the customer absorbs the risk of malfunction on the single tenant node, e.g., the burdens of polling for malfunction, requesting for and acquiring a new single tenant node upon malfunction, and promptly live-migrating to the new node fall on the customer.

Customers have to stitch their virtual private cloud together to achieve their performance and utilization goals which is an involved task when their preferences differ from the underlying configuration provided by the cloud service provider. Customers are also unable to leverage the cloud management facilities native to the cloud service provider e.g., automatic capacity management, live-migration, statistical multiplexing across a diversity of customer workloads, etc., e.g., as available with standard multitenancy.

## DESCRIPTION

This disclosure describes techniques that enable customers to work in standard multitenant environments while still being able to customize their experience to levels comparable to those offered via dedicated hardware or sole tenant hosts. The high level of customization in a multitenant environment occurs even as the customer uses the same, familiar API set as the regular public cloud. Benefits of single tenancy such as load profile optimized overcommit, compact placement, autoscaling, etc., accrue without the burdens of capacity management, task scheduling, risk management, etc. while enjoying the better statistical multiplexing offered by the standard, multitenant public cloud. The described techniques also enable cloud customers to obtain a satisfactory level of performance from their virtual machines (VMs) without having their own VMs competing with each other for resources as is the case with sole tenancy. Prior cloud computing facilities only provide compact placement and managed instance groups, whereas the described techniques enable these along with the additional ability to customize overcommit on a per VM basis.

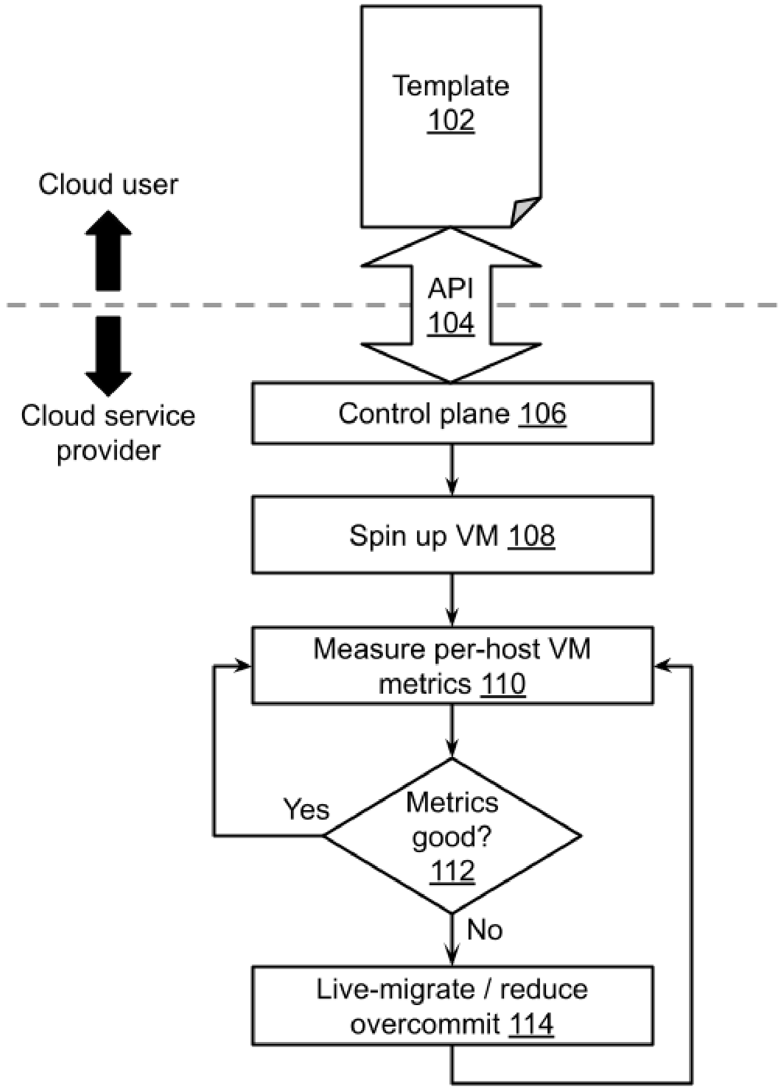


**Fig. 1: Customizable cloud clusters**

Fig. 1 illustrates customizable cloud clusters. Customers can provide their configuration requirements via a template (102) that includes a level of overcommit they want their VMs to experience, placement constructs (spread, compact); autoscaling configuration that controls how the group can expand/shrink (while meeting VM placement requirements); etc. Customers can launch both on-demand or preemptible VMs using this template by passing it in during VM creation. The custom template can include overcommit settings; autoscaling settings on when to tune overcommit; placement information like spread/compact to help distribute VMs; etc.

Customers can express customize their VMs in bulk via this mechanism with the same, familiar tools they use with non-customized VMs. As such, these are levels of customization not typically offered in multitenant clouds.

The custom template can be absorbed by an API (104) exposed in the control plane (106) of the cloud service provider that takes the custom template as input and models it as a new type of object that can be observed, manipulated, and/or deleted by the customer. Once these objects exist, the control plane enables the cloud service provider to reference the objects as part of the instance creation APIs such that the additional customization settings can be applied to the newly created instances.

Upon receiving the custom template, the cloud service provider, through a placement module, spins up VMs (108) on the hosts that are appropriate for the customer request. These can be standard, multitenant cloud hosts, thus providing customers a fully elastic cloud computing experience. Customers can launch any kind of VM with their templates, including on-demand or preemptible, and the cloud service provider automatically places the demanded instances.

Based on the provided configuration, VMs can be live-migrated to meet or exceed overcommit targets. If VM metrics (110) on a per-host basis reach levels that are deemed slow or bad (112), the cloud service provider can automatically reduce overcommit and/or live-migrate VMs (114). Each VM is a full VM, capable of using persistent disks (PD), virtual private cloud (VPC), etc., and providing the complete public cloud experience. Additionally, customers can control the level of input/output operations per second (IOPS) in PD by controlling how many cores can be dedicated to PD processing.

The cloud service provider applies overcommit settings for VMs belonging to the customer, either by using a pool of hosts that already have the correct settings or by creating per-

VM settings on a host that might run other VMs from other customers with different settings. As is common for multitenant environments, the cloud service provider continues performing usual cloud management functions, e.g., tuning overcommits as needed, live-migrating VMs transparently during maintenance, etc., while ensuring compliance with the overall template service level agreements/service level objectives (SLAs/SLOs).

The virtual machine scheduler, which can be powered by a job scheduler, is modified suitably to enable varying levels of overcommit within the same host. Alternatively, pools of hosts can be maintained with differing overcommit options. Virtual machines on a single host directly or directly share resources such as networking throughput and memory bandwidth. If, as described herein, VMs on a single host have different overcommit expectations, isolation between these VMs can be achieved by changing the virtualization layer to adjudicate fair splitting of shared host resources amongst the VMs by accounting for their respective overcommit configuration. Schedule placement logic is updated to optimize host utilization and, as far as possible, to forestall stranding valuable capacity.

In this manner, customers get a highly configurable cloud deployment tunable for their precise applications, including their preferences regarding overcommit, placement, and autoscaling, while still getting the benefits of the public cloud. The described techniques apply to direct VMs and can be exposed via compute cluster management services where customers run containers that run in VMs. Customers can provide preferences on groups of containers to overcommit (not), on resize resource reservations, etc. The compute cluster management service can use the support provided by the underlying cloud service provider to meet those preferences.

## CONCLUSION

In cloud computing, sole tenant nodes dedicate hardware to a single customer, enabling the customer to tightly optimize utilization and performance to match their load profile. However, with sole tenancy, the burdens of capacity management, risk management, task scheduling, triggering live-migration, etc., fall on the customer. This disclosure describes techniques to enable customers to run their workloads in standard, multitenant cloud computing environments while still being able to customize the experience to levels comparable to those offered via dedicated hardware or sole tenant hosts. The high level of customization in a multitenant environment is enabled even as the customer uses the familiar API set of the public cloud. Benefits of single tenancy such as load profile optimized overcommit, custom placement, autoscaling, etc., accrue without the burdens of capacity management, task scheduling, risk management, etc. as the workloads enjoy the better statistical multiplexing offered by the standard, multitenant public cloud.