Chapter 3
Key Enabling Technologies for Virtual Private Clouds

Jeffrey M. Nick, David Cohen, and Burton S. Kaliski Jr.

Abstract The concept of a virtual private cloud (VPC) has emerged recently as a way of managing information technology resources so that they appear to be operated for a single organization from a logical point of view, but may be built from underlying physical resources that belong to the organization, an external service provider, or a combination of both. Several technologies are essential to the effective implementation of a VPC. *Virtual data centers* provide the insulation that sets one organization's virtual resources apart from those of other organizations and from the underlying physical infrastructure. *Virtual applications collect* those resources into separately manageable units. *Policy-based deployment* and *policy compliance* offer a means of control and verification of the operation of the virtual applications across the virtual data centers. Finally, *service management integration* bridges across the underlying resources to give an overall, logical and actionable view. These key technologies enable cloud providers to offer organizations the cost and efficiency benefits of cloud computing as well as the operational autonomy and flexibility to which they have been accustomed.

3.1 Introduction

It is becoming relatively commonplace for organizations to outsource some or all of their IT operations to an external "cloud" service provider that offers specialized services over the Internet at competitive prices. This model promises improved total

J.M. Nick (✉)
Office of the CTO, EMC Corporation, Hopkinton, MA, USA
e-mail: jeff.nick@emc.com

D. Cohen
Cloud Infrastructure Group, EMC Corporation, Cambridge, MA, USA
e-mail: david.cohen@emc.com

B.S. Kaliski Jr.
Office of the CTO, EMC Corporation, Hopkinton, MA, USA
e-mail: burt.kaliski@emc.com

B. Furht, A. Escalante (eds.), *Handbook of Cloud Computing*,
DOI 10.1007/978-1-4419-6524-0_3,

cost of ownership (TCO) through the leverage of large-scale commodity resources that are dynamically allocated and shared across many customers. The problem with this model to date is that organizations have had to give up control of the IT resources and functions being outsourced. They may gain the cost efficiencies of services offered by the external provider, but they lose the autonomy and flexibility of managing the outsourced IT infrastructure in a manner consistent with the way they manage their internal IT operations.

The concept of a *virtual private cloud (VPC)* has emerged recently (Cohen, 2008; Wood, Shenoy, Gerber, Ramakrishnan, & Van der Merwe, 2009; Extend Your IT Infrastructure with Amazon Virtual Private Cloud, http://aws.amazon.com/vpc/) as answer to this apparent dilemma of cost vs. control. In a typical approach, a VPC connects an organization's information technology (IT) resources to a dynamically allocated subset of a cloud provider's resources via a virtual private network (VPN). Organizational IT controls are then applied to the collective resources to meet required service levels. As a result, in addition to improved TCO, the model promises organizations direct control of security, reliability and other attributes they have been accustomed to with conventional, internal data centers.

The VPC concept is both fundamental and transformational. First, it proposes a distinct abstraction of public resources combined with internal resources that provides equivalent functionality and assurance to a physical collection of resources operated for a single organization, wherein the public resources may be shared with many other organizations that are also simultaneously being provided their own VPCs. Second, the concept provides an actionable path for an organization to incorporate cloud computing into its IT infrastructure. Once the organization is managing its existing resources as a private cloud (i.e., with virtualization and standard interfaces for resource management), the organization can then seamlessly extend its management domain to encompass external resources hosted by a cloud provider and connected over a VPN.

From the point of view of the organization, the path to a VPC model is relatively straightforward. In principle, it should be no more complicated, say, than the introduction of VPNs or virtual machines into the organization's IT infrastructure, because the abstraction preserves existing interfaces and service levels, and isolates the new implementation details. However, as with introduction of any type of abstraction, the provider's point of view is where the complexities arise. Indeed, the real challenge of VPCs is not whether organizations will embrace them once they meet organizational IT requirements, but how to meet those requirements – especially operational autonomy and flexibility – without sacrificing the efficiency that motivated the interest in the cloud to begin with.

With the emergence of VPCs as a means to bring cloud computing to organizations, the next question to address is: *What are the key technologies cloud providers and organizations need to realize VPCs?*

3.2 Virtual Private Clouds

A *cloud*, following NIST's definition that has become a standard reference (Mell & Grance, 2009), is a pool of configurable computing resources (servers, networks, storage, etc.). Such a pool may be deployed in several ways, as further described in Mell and Grance (2009):

- A *private cloud* operated for a single organization;
- A *community cloud* shared by a group of organizations;
- A *public cloud* available to arbitrary organizations; or
- A *hybrid cloud* that combines two or more clouds.

The full definition of a private cloud given in Mell and Grance (2009) is

> *Private cloud*. The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.

The definition suggests three key questions about a cloud deployment:

1. Who uses the cloud infrastructure?
2. Who runs the infrastructure?
3. Where is the infrastructure?

The distinction among private, community, public, and hybrid clouds is based primarily on the answer to the first question. The second and third questions are implementation options that may apply to more than one deployment model. In particular, a cloud provider may run and/or host the infrastructure in all four cases.

Although NIST's definition does not state so explicitly, there is an implication that the cloud infrastructure refers to physical resources. In other words, the computing resources in a private cloud are physically dedicated to the organization; they are used only (i.e., "solely") by that organization for a relatively long period of time. In contrast, the computing resources in a public or community cloud are potentially used by multiple organizations over even a short period of time.

The physical orientation of the definition motivates the concept of a *virtual private cloud*, which, following the usual paradigm, gives an appearance of physical separation, i.e., extending (Mell & Grance, 2009):

> *Virtual private cloud (VPC)*. The cloud infrastructure appears as though it is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise — or some combination of these options.

In other words, a VPC offers the *function* of a private cloud though not necessarily its *form*. The VPC's underlying, physical computing resources may be operated for many organizations at the same time. Nevertheless, the virtual resources presented to a given organization – the servers, networks, storage, etc. – will satisfy the same requirements as if they were physically dedicated.

The possibility that the underlying physical resources may be run and/or hosted by a *combination* of the organization and a third party is an important aspect of the definition, as was first articulated by R. Cohen in a May 2008 blog posting (Cohen, 2008) that introduced the VPC concept:

> A VPC is a method for partitioning a public computing utility such as EC2 into quarantined virtual infrastructure. A VPC may encapsulate multiple local and remote resources to appear as a single homogeneous computing environment bridging the ability to securely utilize remote resources as part of [a] seamless global compute infrastructure.

Subsequent work has focused on a specific implementation profile where the VPC encompasses just the resources from the public cloud. Wood et al. in a June 2009 paper (Wood, Shenoy, Gerber, Ramakrishnan, & Van der Merwe, 2009) write:

> A VPC is a combination of cloud computing resources with a VPN infrastructure to give users the abstraction of a private set of cloud resources that are transparently and securely connected to their own infrastructure.

Likewise, Amazon describes its virtual private cloud in a January 2010 white paper (Extend Your IT Infrastructure with Amazon Virtual Private Cloud, http://aws.amazon.com/vpc/) as "an isolated portion of the AWS cloud," again connected to internal resources via a VPN.

In both Wood et al. and Amazon, a VPC has the appearance of a private cloud, so meets the more general definition stated above. However, the implementation profile imposes the limitation that the physical resources underlying the VPC are hosted and run by a cloud provider. In other words, the answer to the second and third questions above is "external." Although internal resources, e.g., the "enterprise site" of Wood et al., are connected to the VPC over the VPN, they are not part of the VPC proper.

This article maintains R. Cohen's broader definition because the cloud for which an organization will be responsible, ultimately, will encompass most or all of its resources, not just the external portions. The primary VPC implementation profile considered here is therefore one in which the underlying resources are drawn from a public cloud and an internal, private cloud – or, in other words, from a hybrid cloud that combines the two (see Fig. 3.1) (Note 1). How those resources are managed in order to meet organizational IT requirements is the focus of the sections that follow.

Note

1. The implementation profile is most relevant to medium to large enterprises that already have substantial internal IT investments and are likely to maintain some of those resources while incorporating IT services from an external cloud provider. For enterprises that outsource all IT to a cloud provider, the implementation profile would include only the external resources. The key enabling technologies for VPCs are relevant in either case.

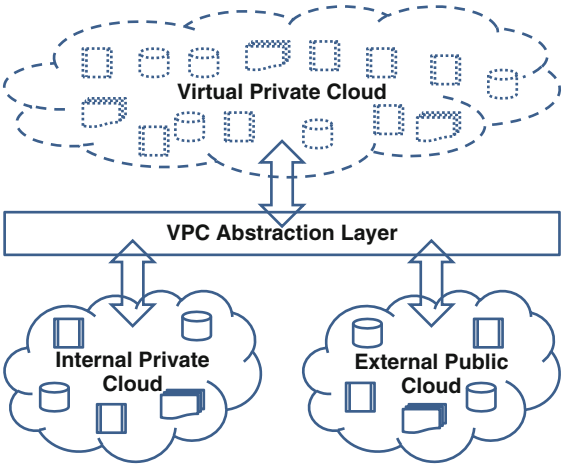


Fig. 3.1 Primary virtual private cloud (VPC) implementation profile: VPC is built from the hybrid of an external public cloud and an internal private cloud

3.3 Virtual Data Centers and Applications

An organization's objective for its use of IT in general is to realize certain information-based business processes while conforming with applicable laws and regulations, and optimizing the cost/benefit tradeoff. Whether implemented with cloud computing or with conventional IT, the organization's high-level IT objectives are the same. The promise of cloud computing is that over time, organizations may be able to meet those objectives with an ever improving cost/benefit tradeoff.

3.3.1 Virtual Data Centers

In conventional IT, *data centers* provide a convenient way of organizing resources into locally connected pools. The locality provides an opportunity for common physical oversight and improved network performance among resources within the data center. In effect, a data center can be viewed as a local *container* of IT resources that can be managed together from a resource, security, and/or information perspective.

Virtualization, as a general paradigm, *insulates* resources and functionality from the underlying physical implementation, with the consequent advantage that virtual resources can be dynamically allocated to an organization without concern (by the organization) for the underlying physical implications. Moreover, the virtual resources can readily be migrated from one physical environment to another – for instance, between the organization's data centers and data centers operated by a cloud provider.

From this perspective, virtual resources "in the cloud" are, in principle, location- and container-independent. However, for the same reasons as in conventional IT,

containers and location-type attributes may play an important role in practice, because organizations will continue to call for the performance advantages that locality brings, and it will be convenient to manage resources as sets. Accordingly, just as data centers are the basic, high-level unit of management in conventional IT, it is reasonable to expect that *virtual data centers* – the first key enabling technology for VPCs – will be the basic, high-level unit of resource management (Notes 1, 2):

> *Virtual data center (VDC).* A pool of virtual resources that appear in terms of performance to be locally connected, and can be managed as a set.

For practical reasons, a VDC will typically be implemented based on a single, underlying physical data center; the apparent local connectivity would otherwise be difficult to achieve (although there are recent high-performance network technologies that do span physical data centers). The limitation is only in one direction, of course: A given physical data center can host more than one VDC. Furthermore, a data center operated by a public cloud provider may offer VDCs to multiple organizations, or henceforth, *tenants*, so the underlying computing environment is *multi-tenant*.

In addition to local connectivity, the placement of resources in a particular location may offer geographical advantages such as proximity to certain users or energy resources, or diversity of applicable laws and regulations. The placement of resources across multiple, independent locations can also help improve resilience. Such geographical aspects may be "virtualized" by policy-based management (see Section 3.4 below). The VDC (and/or its resources) would be selected so that they achieve the desired properties, with the actual location left to the implementation (although certain properties may only be achievable in a specific geography).

In addition, VDCs, like physical data centers, may vary in terms of the capabilities they offer, such as:

1. The types of virtual resources that are supported;
2. The cost, performance, security, and other attributes of those resources (and of the VDC in general), including the types of energy used; and
3. Specific resources that are already present and may be inconvenient to obtain elsewhere, such as large data sets or specialized computing functionality.

Rather than presenting the appearance of a physical data center as it actually is, the VDC abstraction can show a data center *as it ideally should be*. As a result, a VDC offers the opportunity for simplified, unified management from the point of view of the organization using it.

Given the VDC as a basic unit of management, the primary VPC implementation profile may be further refined as one in which the virtual resources are organized into the combination of

- One or more private VDCs hosted by a cloud provider; and
- One or more internal, private VDCs hosted by the organization

The cloud provider's VDCs would be based on scalable partitions of the cloud provider's public data centers; the internal VDCs could be simply the virtualization of the organization's existing data centers, or perhaps again scalable partitions. In either case, the modifier *private*, is essential: In order for the resulting VPC to appear as though it is operated solely for an organization, the component VDCs must be so as well.

Building a VPC thus decomposes into the problem of building a private VDC, or, expanding the definition, a pool of virtual resources that appears to be locally connected and to be operated for a single organization. The specific translation between a private VDC and underlying physical resources is, of course, a matter of implementation, but several technologies clearly will play a key role, including, obviously, virtualization and resource management, as well as, possibly, encryption of some form (Note 3).

With this first enabling technology in place, an organization using a VPC will have at its disposal some number of private VDCs or containers into which it may deploy resources, as well as the possibility of obtaining additional VDCs if needed. How those containers are used is the subject of the next enabling technology.

Notes

1. Cloud computing can be realized without the data center metaphor, for instance in settings where local connectivity is not important, such as highly distributed or peer-to-peer applications. The focus here is on the typical enterprise use cases, which are data-center based.
2. Virtualization, in principle, gives an appearance of privacy in the sense that if all tenants interact only through the VDC abstraction, then, by definition, they cannot access one another's resources (assuming of course that in the physical counterpart whose appearance is being presented, they cannot do so). Thus, virtualization of servers, networks, storage, etc., alone is arguably sufficient to build a VDC (as far as appearances; resource management is also needed to handle scheduling, etc.).

   There are two main problems with this position. First, there may be paths outside the abstraction by which parties may interact with the underlying resources. At the very least, the infrastructure operator will have such access, both physical and administrative. Second, there may be paths *within* the abstraction that inadvertently enable such interaction, whether due to errors or to side channels that are not completely concealed. This introduces the possibility of malevolent applications or *mal-apps* that target other tenants sharing the same public computing environment. The *cloud cartography* and *cross-VM information leakage* techniques of Ristenpart, Tromer, Shacham, and Savage (2009) are recent examples.

It is worth noting that comparable vulnerabilities are already being dealt with by conventional data centers through a range of security controls, from encryption to behavioral analysis. The difference in cloud computing is not as much the nature of the vulnerabilities, as the number of potential adversaries "in the camp." A base and adaptive set of security controls will be essential for the abstraction, robustly, to maintain its assurances, while applications implement additional controls above the abstraction just as they would if running directly on physical infrastructure. A good example of such an additional control is the VPN (which is visible to applications in the private VDC model).

Trusted computing (Mitchell, 2005) may also play a role in private VDCs by providing a *root of trust* with respect to which tenants may verify the integrity of their resources. The integration of trusted computing and virtualization is explored more fully in projects such as Terra by Garfinkel, Pfaff, Chow, Rosenblum, and Boneh (2003) and Daonity (now continued as Daoli) (Chen, Chen, Mao, & Yan, 2007).

3.3.2 Virtual Applications

Information-based processes in conventional IT are realized by various *applications* involving interactions among collections of resources. The resources supporting a given application may run in a single data center or across multiple data centers depending on application requirements.

Continuing the analogy with conventional IT, one may expect that *virtual applications* – the second key enabling technology – will be the basic, high-level unit of resource *deployment*:

> *Virtual application*. A collection of interconnected virtual resources deployed in one or more virtual data centers that implement a particular IT service.

A virtual application consists not only of the virtual machines that implement the application's software functionality, but also of the other virtual resources needed to realize the application such as virtual networks and virtual storage. In this sense, a virtual application extends the concept of a *virtual appliance* (Sapuntzakis & Lam, 2003), which includes the complete software stack (virtual machines and operating system, with network interfaces) implementing a single service, to encompass the set of services supporting the application.

Just as a VDC can show a data center in a more ideal form, a virtual application can present the appearance of an application as it ideally should be. Today, security, resource management, and information management are typically enforced by the operating system and application stack, which makes them complex and expensive to implement and maintain. With the simplified, unified management provided by the virtual application abstraction and encapsulation of application components in virtual machine containers, the virtual application container becomes a new control

point for consistent application management. Instead of orchestrating each resource individually, an organization can operate on the full set in concert, achieving the equivalent of "one click" provisioning, power on, snapshot, backup, and so on.

The primary VPC implementation profile may be now refined again as one in which virtual applications consisting of virtual resources run across one or more VDCs (see Fig. 3.2) (Note 1).

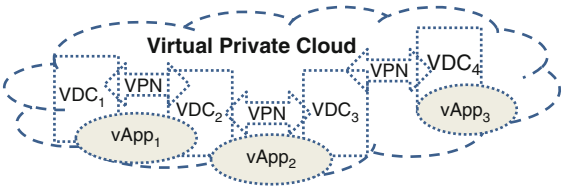


Fig. 3.2 Virtual applications run across one or more private virtual data centers (VDCs), connected by virtual private networks (VPNs)

The Open Virtualization Format (OVF, 2009) recently standardized by the Data Management Task Force offers a convenient way to specify collections of virtual machines. Through metadata, the interconnections between those machines and their dependencies on other resources such as networks and storage may also be expressed, supporting full virtual applications as defined here. In addition to several commercial products, the specification is also supported in the Open-OVF open source project (open-ovf.sourceforge.net).

An organization using a VPC with the first two enabling technologies now introduced will be able to use its private VDCs to deploy virtual applications. The next enabling technologies address the contract between those applications and the VPC that enables the automatic assembly of components to meet organizational IT objectives while maintaining flexibility for optimization.

Note

1. The interplay between VDCs and virtual applications is an important aspect of meeting organizational IT requirements with a VPC, which do depend in some cases on (possibly relative) location, as noted in Section 3.3.1. Thus, in addition to the primary role of virtual applications in enabling portability *between* clouds, virtual applications may also be viewed as a way to enable effective deployment *within* a cloud by describing the desired relationships among virtual resources.

3.4 Policy-Based Management

Over time, a VPC will be populated with resources supporting virtual applications running at various VDCs. Those resources will be deployed and assembled with the ultimate intent of meeting the organizational IT requirements. This is the essence of the "contract," formal or otherwise, between an organization and the VPC.

The role of such a contract can be viewed in two parts: putting its terms into practice, and checking that the practice is correct.

3.4.1 Policy-Based Deployment

Consider an organization that wants to deploy an e-commerce application with certain objectives for security, performance, and business continuity. In a conventional data center, that application might be implemented as the combination of resources starting with a web server and a database. A firewall would be added to meet the security objectives, and a load-balancer to assign transactions to additional web servers as needed to meet the performance objectives. To address the business continuity objectives, a second instance of these components might be placed in another data center, coordinated with the first through a business continuity manager.

Suppose further that the organization also wants to deploy a customer relationship management (CRM) application with similar service objectives. That application might likewise be implemented as a combination of web servers, databases, firewalls, load-balancers, and so on, across two data centers.

Now, consider what happens when the organization decides to deploy these applications in a VPC (under some "contract," per the comments above). Following the model in Section 3.3.2, each application would be deployed as a collection of virtual resources. The VPC would thus be hosting the combination of the sets of resources for the two applications: two sets of virtual web servers, two virtual databases, two firewalls, two load-balancers, etc., and this collection would be repeated across two VDCs.

Deploying an application in a VPC as just described has some advantages, such as dynamic allocation of resources and economies of scale. However, such a process is really no more than a migration of components from one environment to another, or what could also be called a *literal virtualization*. Infrastructure sprawl in the data center translates directly into *virtual sprawl*, with as many components to manage as before, just consolidated onto fewer servers, and arranged in a more flexible way.

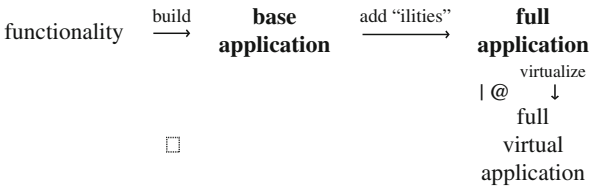
Cloud computing environments can improve this situation by organizing components and capabilities into searchable lists of virtual applications and resources that can readily be deployed. By selecting services from an offering catalog and inventory, rather than imposing entirely unique choices, an organization can take advantage of optimizations that the cloud provider may already have put in place. The load-balancer would be a good example. Instead of each application contributing its own load-balancer, the VPC would offer one itself for use by multiple applications.

Once an application designer knows that load-balancing will be available, he or she no longer needs to specify a virtual application as a combination of, say, two web servers and a load-balancer. Instead, the virtual application may be expressed as the combination of a single web server (and other functional components) and a *policy* that the VPC should create additional instances of the web server and balance transactions among them to maintain a specified performance level. This policy has the further benefit that the application may automatically be scaled *beyond* the two instances originally specified in a literal counterpart to the data center version.

Load-balancing is one instance of a general pattern: Applications are designed as a combination of *functionality* and *qualities* relating to *service-level agreements*

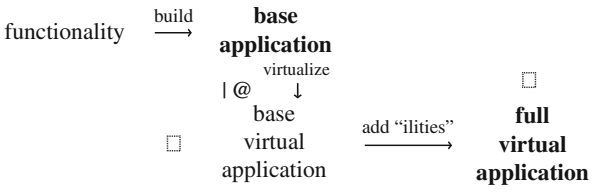
(SLAs). These qualities, sometimes also called, "*ilities*" (from the common suffix of scalability, availability, etc.), generally are implemented with quite similar components across different applications, like load-balancers, firewalls, business continuity managers, and so on. They are therefore natural candidates for services supporting multiple applications in a VPC.

A simple formula illustrates both the pattern and the problem. A typical application is constructed first by building a *base application* that meets some functional requirements, then adding "ilities" that address the non-functional ones. The resulting full application is then virtualized and deployed in the VPC. This pattern may be summarized as follows:



Given only the full virtual application, the VPC will likely have a problem recognizing the "ilities," and therefore, managing them or optimizing their delivery, as much as it is difficult to optimize object code without the original source. However, given some of that original source, the VPC will have much more opportunity to add value. Consequently, the preferred model for application deployment in a VPC is for the computing environment to *add "ilities" as part of deployment.*

The pattern now becomes the following:



The "ilities" may be added by configuring the base virtual application or by deploying additional services. Following the VDC model in Section 3.3.2, policy may also be realized according to the placement of virtual resources into specific VDCs, e.g., where local connectivity, proximity to certain users, independence, etc. are required.

The paradigm may be summarized as the third key enabling technology, *policy-based deployment*:

> *Policy-based deployment*. The assembly of application components in a computing environment according to predefined policy objectives.

Although policy-based deployment can also be applied in other types of clouds (as well as in data centers), the technology is particularly important for VPCs because of their primary use case: organizations that need to meet well-defined IT objectives.

Shifting the introduction of some policy features from development to deployment doesn't necessarily make deployment easier, and in fact may make it harder, as Matthews, Garfinkel, Hoff, and Wheeler (2009) observe, due to number of stakeholders and administrators potentially involved. *Automation* is essential to simplifying deployment, and a well-defined language for expressing policy is essential to automation. The separation of "ilities" from functionality is therefore necessary but not sufficient. In addition, the "ilities" must be expressed as machine-readable rules that the computing environment can implement. In Matthews et al. (2009), such rules take the form of a *Virtual Machine Contract*, defined as:

> A Virtual Machine Contract is a complex declarative specification of a simple question, should this virtual machine be allowed to operate, and if so, is it currently operating within acceptable parameters? (Matthews et al., 2009)

A specification like OVF can be employed to carry contracts and other policy information so that they travel along with the virtual machines, and, more generally, virtual applications, to which the conditions apply.

With automated policy-based deployment in place, an organization is able to specify to the VPC its expectations as far as security, performance and other SLAs, and the VPC can then, automatically, optimize its operations toward those objectives. The military expression, "You get what you inspect, not what you expect," motivates the challenge addressed by the next enabling technology: How to verify that the terms of the contract with the VPC are actually met.

3.4.2 Policy Compliance

In whatever computing environment an organization chooses to deploy an application, the organization will need some evidence that the application is running as intended. This evidence serves both the organization's own assurances and those of auditors or customers. Even if no malice is involved, the environment's optimizations may only approximate the intended result.

Policy objectives are particularly difficult to achieve in a multi-application setting because of the potential for resource contention. A physical computing resource, for instance, may reliably meet the computational performance objectives for the one application it supports, but when that resource interacts with another resource, the presence of network traffic from other applications may make the communication performance unpredictable. Network reservation schemes and similar approaches for storage play an important role in meeting SLAs for this reason. There may also be opportunities for different applications, by design, to operate in a complementary fashion that reduces the contention.

A multi-tenant computing environment such as a public cloud hosting VPCs for multiple organizations introduces further complexities. As with any Internet service, tenants are affected by one another's behavior, which can be unpredictable. Because there is no direct opportunity for negotiation among different tenants with respect to the underlying computing environment (in principle, they cannot even detect one another), any contention must be resolved by the cloud provider itself.

The objectives of different tenants are not necessarily aligned with one another, so in addition to the basic resource contention, there may also be contention among optimization strategies. The potential for interference is another strong motivation for placing the policy services within the computing environment rather than individual applications.

Finally, the tenants' objectives are not necessarily aligned with those of the public cloud provider. Although serving customers will presumably be the first priority of a successful cloud provider, staying in business is another, and there is a definite motivation for implementing further optimizations that cut costs for the provider without necessarily increasing benefit for any of the tenants (Note 1).

Given the difficulty of meeting policy requirements perfectly across multiple applications and tenants, it becomes particularly important for the VPC to provide, and the tenant to receive, some information about the extent to which those requirements are met, or not, at various points in time. This leads to the fourth key enabling technology, *policy compliance.*

> *Policy compliance*. Verification that an application or other IT resource is operating according to predefined policy objectives.

Per the separation of "ilities" from based functionality discussed in Section 3.4.1, it is reasonable to expect that policy compliance itself will eventually be considered as just another service to be added to the application when deployed in the VPC (Note 2). Such a capability goes hand in hand with policy-based deployment: It will be much easier for a VPC to gather appropriate evidence from resources it has already assembled with policy objectives in mind, than to have to discover the objectives, the resources, and the evidence after the fact.

As far as the evidence itself, for precisely the purpose of evaluating performance, many IT resources are instrumented with activity logs that record transactions and other events. For instance, a physical network router may keep track of the source, destination, size, timestamp and other metadata of the packets it transfers (or is unable to transfer); a physical storage array may record similar information about the blocks it reads and write. With appropriate interfaces, the virtual environment can leverage these features to gather evidence of policy compliance. For example, *I/O tagging* embeds virtual application identifiers as metadata in physical requests, with the benefit that the identifiers are then automatically included in activity logs for later analysis by the virtual environment with minimal impact on performance (Note 3).

The collection of system logs from physical computing, networking, and storage resources, containing information about virtual applications and resources and their activities, provides an information set from which policy compliance evidence may be derived. This information set, keyed by the virtual application identifiers and related quantities, enables *distributed application context and correlation* – in effect, a virtual view of the activity of the virtual application, across the VPC.

Constructing such a view, especially from heterogeneous unstructured system logs that were designed only for local visibility, and management interfaces that were intended only for local control, depends on a fifth and final enabling technology, one that responds to the question: How to bring all this information together intelligently?

Notes

1. A related situation where a cloud storage provider may lose some portion of tenants' data as a result of its own performance optimizations (or actual malice) is explored in Juels and Kaliski (2007) and Bowers, Juels, and Oprea (2009), which also propose mechanisms for detecting and recovering from such loss before it reaches an irreversible stage. The detection mechanism may be viewed as an example of policy compliance for stored data.
2. If an application includes built-in policy compliance and the components involved are portable, then the compliance will continue to function in the VPC. Such verification provides a helpful checkpoint of the service levels achieved within a given computing environment. However, as more applications with built-in policy compliance are deployed, the VPC will see a sprawl of application-specific compliance components. This is another motivation for building policy compliance into the VPC.
3. I/O tagging offers the additional benefit of enabling virtualization-aware physical resource managers to enforce policies based on the virtual identifiers. This is a promising area for further exploration, particularly for methods to resolve the contention, as observed above, among policies for different applications and tenants.

3.5 Service-Management Integration

Virtual data centers, the first of the enabling technologies, may be viewed as providing *insulation* that sets one organization's virtual resources apart from those of other organizations, and from the underlying physical resources. Virtual applications, the second, collect those resources into separately manageable units. Policy-based deployment and policy compliance, the third and fourth, offer a means of *control* and verification of the operation of the virtual applications across the VDCs. All four rest on a fifth technology: a more basic foundation that bridges across underlying boundaries, one oriented toward seamless *integration*.

Recall the original implementation profile for the VPC, per Section 3.2: a hybrid of an internal, private cloud and a public cloud. Following Section 3.3, the VPC provides the appearance of some number of VDCs, some drawn from the internal cloud, some from the public cloud. Throughout Section 3.4, this VPC is essentially viewed as seamless, which it is in appearance (the exposure of multiple VDCs is an architectural feature). Thus, Section 3.4 can speak of deploying an application into the VPC, collecting evidence from the VPC, and so on, without regard to the fact that the deployment and collection ultimately involve interactions with physical resources, and more significantly, that these physical resources are in multiple data centers operated by at least two different entities.

The fundamental challenge for satisfaction of policy-based management in a VPC is *how* to enable such seamless interaction between resource, service and policy management components: across data center infrastructure boundaries, and then across federated service provider boundaries.

Such bridges are not easy to build, because the various management interfaces – like the local logs in Section 3.4.2 – were designed for separate purposes. At the physical layer, they may use different names for the same entity or function, employ incompatible authentication and access control systems, and express the same conditions in different ways. The information the organization and the VPC need is available, but is not immediately useful without some translation. Moreover, that translation is not simply a matter of converting between formats, but, in effect, virtualizing the interfaces to the management metadata across the borders of the underlying management component.

The fifth and final key enabling technology, *service-management integration*, addresses this last challenge:

> *Service-management integration*. The translation of heterogeneous management information from separate domains into an overall, logical and actionable view.

Service-management integration is a special case of the broader technology of *information integration*, which is concerned, similarly, with translating of federating general information from multiple domains. The special case of VPCs is concerned in particular with federating three things: (1) the underlying *infrastructure* into one virtual computing environment, (2) *identities* interacting with the resources in the environment, and (3) *information* about the resources.

By its nature, service-management integration for VPCs is amenable to an *eventing* paradigm where the basic unit of information is an event published by one entity in the system, and consumed by another. This paradigm is a good match for a policy compliance manager that is interested in the content of multiple physical logs, as that evidence accumulates. It also provides a deployment manager with a current view of the underlying resources as they continually change. Further, the architectural separation between publisher and subscriber lends itself to the physical separation and distribution of participating elements across data center and cloud federation boundaries.

The intermediation between publisher and consumer can be achieved through a *messaging system*. As a dedicated communication layer for events, such a system provides a federated information delivery "backplane" that bridges multiple management domains (e.g., internal data centers, cloud provider data centers) into a single service-oriented architecture, translating back and forth among the security and management languages of the various domains. Events published in one domain can be consumed in another according to various subscription rules or filters; the policy compliance manager for a particular tenant, for instance, will only be interested in (and should only know about) events related to that tenant's virtual applications.

The messaging system can implement its translations through a set of adapters, informed by an understanding of the connections among the identities and events in the different domains. The system's learning of those connections can occur automatically, or it may require manual intervention, and in some cases it may need to be augmented with a significant amount of computation, for instance to search for correlated events in the different domains. In a cloud computing environment, the resources for such computation will not be hard to find. (How to balance between the use of resources to make the overall environment more efficient, versus allocating them directly to tenants, is another good question for further exploration.)

3.6 Conclusions

This article started with the simple premise that cloud computing is becoming more important to organizations, yet, as with any new paradigm, faces certain challenges.

One of the challenges is to define a type of cloud computing most appropriate for adoption. A virtual private cloud built with IT resources from both the organization's own internal data centers and a cloud provider's public data centers has been offered as a preferred implementation profile. To ensure privacy, i.e., the appearance that the cloud is operated solely for the organization, certain additional protections are also needed.

Another challenge is to make good use of the collective resources. A literal type of virtualization where applications are basically ported from a data center to the VPC would realize some of benefits, but the greater potential comes from enabling the VPC itself to optimize the assembly of applications. The starting point for that advance is the separation of functionality from policy within the specification of a virtual application so that policy requirements can be met in a common and therefore optimized way by the VPC. Commonality of policy management also enables the VPC to verify that policies are met.

Finally, information infrastructure rests as the foundation of realizing a VPC. Indeed, virtualization is all about turning resources into information. The better the VPC can engage with that information, rising from the shadows of the private data center past and the public cloud present, the more effectively organizations can move into the promise of the virtual private cloud future.

References

Bowers, K. D., Juels, A., & Oprea, A. (November 2009). HAIL: A high-availability and integrity layer for cloud storage. *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS), ACM, Chicago, IL, USA*, 187–198.

Cohen, R. (May 2008). *Virtual private cloud*, *Elastic vapor: Life in the cloud.* Retrieved January 2010, from http://www.elasticvapor.com/2008/05/virtual-private-cloud-vpc.html.

Chen, H., Chen, J., Mao, W., & Yan, F. (June 2007). Daonity – Grid security from two levels of virtualization. *Elsevier Journal of Information Security Technical Report, 12*(3), 123–138.

Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., & Boneh, D. (December 2003). Terra: A virtual machine-based platform for trusted computing. *ACM SIGOPS Operating Systems Review, 37*(5), 193–206.

Juels, A., & Kaliski, B. S., Jr. (October 2007). PORs: Proofs of retrievability for large files. *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS), ACM, Alexandria, VA, USA*, 584–597.

Matthews, J., Garfinkel, T., Hoff, C., & Wheeler, J. (June 2009). Virtual machine contracts for datacenter and cloud computing environments. *Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds, ACM, Barcelona,* 25–30.

Mell, P., & Grance, T. (2009). *The NIST definition of cloud computing, version 15, NIST*. Retrieved January 2010, from http://csrc.nist.gov/groups/SNS/cloud-computing/.

Mitchell, C. (Ed.). (2005). *Trusted computing*. London: IET.

OVF (January 2010). *Open virtualization format specification, DMTF Document DSP0243, Version 1.0.0*, Retrieved January 2010, from http://www.dmtf.org/.

Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (November 2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS), ACM, Chicago, IL,* 199–212.

Sapuntzakis, C., & Lam, M. S. (May 2003), Virtual appliances in the collective: A road to hassle-free computing. *Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems, USENIX, Lihue, Hawaii*, 55–60.

Wood, T., Shenoy, P., Gerber, A., Ramakrishnan, K. K., & Van der Merwe, J. (June 2009). The case for enterprise-ready virtual private clouds. *Proceedings of HotCloud '09 Workshop on Hot Topics in Cloud Computing*, *San Diego, CA, USA*, Retrieved January 2010, from http://www.usenix.org/event/hotcloud09/tech/.