Security in the Private Cloud

John R. Vacca

Infrastructure as a Service

Publication details

Mario Santana
Published online on: 01 Sep 2016

How to cite :- Mario Santana. 01 Sep 2016, Infrastructure as a Service from: Security in the Private Cloud CRC Press
Accessed on: 11 Dec 2018

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
In this chapter, I will review the major components of a private cloud infrastructure (PCI) to help us think about the security of that architecture. Whether a cloud environment performs business-critical tasks or supports peripheral activities or houses important data, we must understand the mechanics that make a cloud work before we can secure it.

We’ll begin our discussion by touching on a few high-level conceptual issues. Then we’ll actually break down a cloud infrastructure into its various component pieces. These components include the compute nodes that actually encapsulate the computational capability of a cloud infrastructure and make it fractionally available to several users for sharing; the

## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Contextual Considerations</td>
<td>38</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Greatest Common Denominator</td>
<td>38</td>
</tr>
<tr>
<td>3.1.2</td>
<td>User Communities</td>
<td>39</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Shared Impact</td>
<td>39</td>
</tr>
<tr>
<td>3.2</td>
<td>Components of a Private Cloud Infrastructure</td>
<td>40</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Compute</td>
<td>40</td>
</tr>
<tr>
<td>3.2.1.1</td>
<td>Hypervisors</td>
<td>40</td>
</tr>
<tr>
<td>3.2.1.2</td>
<td>Containers</td>
<td>41</td>
</tr>
<tr>
<td>3.2.1.3</td>
<td>Bare Metal</td>
<td>42</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Network</td>
<td>42</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>Underlying Approaches</td>
<td>42</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>Security Implications</td>
<td>43</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Storage</td>
<td>43</td>
</tr>
<tr>
<td>3.2.3.1</td>
<td>Underlying Approaches</td>
<td>43</td>
</tr>
<tr>
<td>3.2.3.2</td>
<td>Security Implications</td>
<td>44</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Management</td>
<td>44</td>
</tr>
<tr>
<td>3.2.4.1</td>
<td>Underlying Approaches</td>
<td>45</td>
</tr>
<tr>
<td>3.2.4.2</td>
<td>Security Implications</td>
<td>45</td>
</tr>
<tr>
<td>3.3</td>
<td>Summary</td>
<td>46</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>46</td>
</tr>
</tbody>
</table>
network connectivity that allows several compute nodes to communicate with each other and with the outside world while maintaining network layer isolation between unrelated users of the cloud infrastructure; the storage services that provide functionality for applications in the cloud environment to maintain state, usually in the form of virtualized hard drives, that can be used and manipulated by compute nodes and potentially shared over the network capability; and the management layer that lets us manage all the rest. As we look at these component pieces of an infrastructure-as-a-service (IaaS) cloud infrastructure, we’ll take a look at some underlying approaches used by various cloud technologies to deliver each component, and we’ll review some of the key security considerations that pertain to each component. Let’s get started!

3.1 CONTEXTUAL CONSIDERATIONS

Cloud infrastructure is appealing for many reasons. However, fundamentally, the value proposition is very simple: We use it to optimize resource utilization by sharing those resources among multiple users. As we consider security from the perspective of a user of a cloud infrastructure, it will be valuable to consider the bigger picture of how the operation of the cloud infrastructure as a whole can potentially impact us as a single user of that cloud. We’ll review three important concepts [1].

3.1.1 Greatest Common Denominator

One important concept is that of “greatest common denominator.” In operating a cloud environment in support of multiple users, the cloud operator must meet the security requirements of all users of the environment. These various users will generally have various security requirements as well. In a simplistic understanding of this concept, it’s clear the some users will have more rigorous security requirements than others. This understanding actually holds true much of the time, especially to the extent that some users must meet a more stringent set of compliance requirements than others. For example, users running their payment processing or point of sale (POS) backends in the cloud will need to meet PCI and probably additional requirements, compared to a training tool that only hosts nonsensitive corporate training content. In this situation, the concept of greatest common denominator dictates that the cloud environment must meet the needs of the user with the most stringent set of requirements, because this will also meet the needs of all users with less stringent requirements.

However, the mass of requirements gets more complicated when there are users with different requirements, where one set of requirements isn’t necessarily more stringent than another. An example of this situation is when one user runs a POS backend in the cloud environment, whereas another user hosts medical information. One user may need to meet PCI requirements, whereas the other user must comply with Health Insurance Portability and Accountability Act (HIPAA) standards. Both of these standards are stringent in their own ways, but it’s not always possible to say that one standard is more stringent than the other. Therefore, there is no single set of requirements that will meet the needs of both users. Instead, in this situation, we must meet the superset of all requirements demanded by all users, understanding that there is no user that needs all of these requirements.
3.1.2 User Communities

Managing a cloud environment in accordance with the greatest common denominator principle can be a daunting task, because meeting multiple overlapping sets of complex security requirements and managing the audits and certifications around those requirements require a great deal of overhead in terms of planning and governance. One way to reduce the governance overhead in this kind of circumstance is to isolate different groups of tenants or applications into various “communities,” according to their various security needs. The key to this approach is to find users with similar security requirements and group them together. Therefore, any users or applications that deal with credit card or payment information are grouped together into, say, a PCI group. Likewise, all users that deal with medical information are grouped together in a HIPAA group, all users that deal with corporate finance information are grouped together into a Sarbanes–Oxley Act (SOX) group, and so on. We can even have a community of users or applications with minimal security requirements, perhaps called the “out in the cold” group.

Once the various communities have been identified, and if they’ve been well sorted into groups with similar sets of security requirements, the task of managing in accordance with the principle of greatest common denominator is greatly simplified. There is still a need to manage diverse security requirements, but these requirements can be compartmentalized, reducing their impact on each other, the complexity of the overall operational, governance, and management overhead, and ultimately reducing the cost of providing a cloud environment that meets all the users’ needs. In turn, the users enjoy a simpler relationship with their cloud environment, needing only to understand the ways in which the environment meets their security needs without the distraction and confusion of superfluous requirements that don’t apply to their situation. And, of course, if a user’s or application’s needs evolve slightly, the security configuration of their existing community could be evolved to meet the new requirement. Alternatively, if the new requirements represent a more drastic change, the user or application could be migrated to a different community that more closely matches their new set of needs.

3.1.3 Shared Impact

One more overarching concept that is important to keep in mind is the idea of “shared impacts.” There are obvious shared impacts, of course, related to attacks on or failures of the shared infrastructure. Clearly, if an attacker targets any of the users on a shared cloud environment with any kind of successful resource exhaustion attack, then all the other users of that shared cloud environment are likely to feel the effects of resource exhaustion. The best cloud infrastructures are configured in ways that minimize these effects—for example, by minimizing the types and amounts of shared resources that can be exhausted by any one user of the environment.

However, there are other shared impacts that are less obvious, though no less critical. For example, consider the effects of a court-issued subpoena to seize the servers of one user in a multitenant cloud environment. Even if the subpoena is only concerned with the data related to one user or application, law enforcement may judge that the subpoena calls for physical hardware rather than virtual. In that scenario, law enforcement may shut down
and seize hardware that supports multiple users—not just the target user!—and therefore cause an outage that impacts all users of the environment.

Now, this particular scenario was once very common. The best and largest cloud operators have relationships with local law enforcement agencies, and the local offices of regional and national law enforcement agencies, specifically to mitigate the shared impact of evidence seizure activities. Moreover, law enforcement is increasingly educated about the high and unnecessary impact of brute-force seizures, and thus, this specific issue is much less prevalent now than it was 5 years ago. However, it’s a great example of the kinds of nonobvious shared impacts that can make one user’s actions and interactions have a significant effect on other cloud users that have nothing to do with the action or interaction that caused the impact.

3.2 COMPONENTS OF A PRIVATE CLOUD INFRASTRUCTURE

Cloud infrastructures are built to manage various kinds of resources and distribute those resources efficiently among workloads and applications. In the following sections, we’ll take a look at what kinds of resources are managed and distributed by a cloud infrastructure [2].

3.2.1 Compute

The compute component of a cloud infrastructure is the workhorse of that infrastructure (see Figure 3.1). In the context of an IaaS cloud, a workload typically we’ll call a compute instance. The physical hardware that provides compute resources we’ll call compute nodes. One compute node may run multiple compute instances. The compute instances actually encapsulate the computational capability, in the form of CPU processing time and RAM memory working space, of the compute nodes in the cloud infrastructure and make it fractionally available to several users for sharing [3]. There are three underlying approaches to implementing compute instances: hypervisor, container, and bare metal.

3.2.1.1 Hypervisors

In practice, most cloud infrastructure designs leverage virtualization to implement the compute component. That’s because virtualization defines hardware as software—a virtualized compute instance is essentially a file, like a program or a configuration file or the file I’m editing as I write this chapter. It is much easier to copy a file to a new location than to move a physical server. It is much easier to reconfigure a server by editing a file than by opening a physical piece of hardware, and add or remove CPUs, RAM memory, storage, and so on. To that end, there are two main kinds of virtualization in use for cloud infrastructures today.

Perhaps the most common kind of virtualization is a hypervisor. A hypervisor creates virtual machines, complete with virtual CPUs, memory, network cards, peripheral busses, disks, even a complete virtual basic input/output system (BIOS). It isolates compute instances by keeping their kernel space separate, while allowing to share a single physical CPU, RAM, and storage space—this is the chief security concern when considering hypervisor implementations. With a hypervisor, each compute instance is running an entirely distinct and isolated operating system.
3.2.1.2 Containers

In contrast to the hypervisor approach, a container virtualizes the operating system. The same operating system kernel is shared among multiple applications or workloads, which represent the compute instance in the container model. Each virtualized environment in this approach, what we generically call a compute instance, is called a container in the context of this approach. The container functionality is actually part of the host operating system that runs on the host compute node. This approach is much more lightweight than the hypervisor approach.

A container isolates applications by keeping their user space separate, while allowing them to share a single kernel space. It’s the job of the container software to ensure that one container is not accessible in any way by a different container. The maintenance of that strict isolation, a thorough reworking of every operating system facility that might expose one container to another, is the chief isolation challenge of a container implementation, making it more complex to isolate workloads in a container model compared to a hypervisor.
3.2.1.3 Bare Metal

A “bare metal” compute component manages workloads without virtualization. Each workload is run on a separate piece of physical hardware. In the bare metal approach, the distinction between compute node and compute instance is eliminated—they are one and the same. The benefit of including bare metal compute in a cloud architecture is the ability to monitor and report on the bare metal node in ways that are compatible with the monitoring and reporting of the rest of the cloud infrastructure.

And the reason why bare metal compute nodes are valuable is because there are certain workloads and applications that do not lend themselves to efficient virtualization. Performance-intensive applications—especially applications that require fast, low-latency disk I/O performance, such as large and high-performance databases—are the prime example of applications that don’t lend themselves to effective virtualization. For this reason, many cloud infrastructure support bare metal nodes where these kinds of applications can run, while still allowing the management of the overall infrastructure using a cloud-oriented management model.

3.2.2 Network

The network component of a cloud infrastructure allows for connectivity among the compute, storage, and other elements of that infrastructure, as well as with the broader environment outside that infrastructure. At a minimum, the network component connects the network facilities of the compute component to the edge of the cloud environment and manages the kind of access that the compute instances have between each other and to the broader environment. Beyond that minimum required capability, there are several techniques to manage the network topology in a cloud infrastructure: virtual switching, management of physical network equipment, and software-defined networking.

3.2.2.1 Underlying Approaches

Virtual switching enables a compute node to define virtual networks that only exist inside the hypervisor or container. This allows the creation of a virtual subnet by connecting the virtual network interfaces of several virtual machines to this virtual switch. This virtual subnet is totally isolated from the physical network, but the virtual workloads and applications on that virtual subnet can communicate as if they were physical machines connected to the same physical network switch.

More advanced versions of this technology allow for “ports” on the virtual switch to exist on separate physical compute nodes, so that workloads on different physical compute nodes can connect to the same virtual switch. This makes it possible to configure the network so that compute instances can operate as if they were on the same switch, regardless of which physical compute node is running that instance, and regardless of whether that workload is migrated to different compute nodes over time.

To support these advanced operations, the network component of a cloud infrastructure must often manage physical network equipment—nonvirtualized switches, firewalls, and so on—to create the network topology that has been configured. By coopting physical network equipment, the network component can manage communications between...
compute instances across separate compute nodes when those compute nodes are connected to separate physical switches and even when those compute nodes are in different data centers across the world.

A “software-defined network” technology takes these concepts even further. In software-defined networking, the network topology definition is abstracted and completely virtualized. This allows for switch ports, virtual local area networks (VLANs), firewall rules, and other advanced network configurations of a cloud infrastructure to be managed by the network component through management interfaces or configuration files. The network component then configures the details of each physical and virtual network device in the infrastructure, to implement the topology as defined in the abstraction. The main benefits of a software-defined network are centralized network management of diverse devices; more flexible configuration options; and seamless disaster recovery, load balancing, and other typical cloud operations.

3.2.2.2 Security Implications
The most security-critical responsibility of the network component is isolation. The network component must keep separate compute instances from communicating to each other in unauthorized ways. For example, the network component must prevent malicious workloads from accessing unauthorized VLANs and subnets that may be accessible by authorized workloads sharing the same physical hardware. The virtualization of the network component also makes it possible to implement a wide variety of monitoring schemes for network traffic in the cloud infrastructure.

3.2.3 Storage
The storage component of a cloud infrastructure provides data storage services to that infrastructure. At a minimum, the storage component stores cloud management information, such as virtual machine and virtual network definitions, and provides working space to applications and workloads running in the cloud environment. Beyond that minimum required capability, there are several techniques to provide workload migration, automated backups, integrated version control, and optimized application-specific storage mechanisms.

3.2.3.1 Underlying Approaches
The fundamental capabilities of the storage component involve managing the storage required by the cloud infrastructure management functions. The details of this depend on the specific technology and approach of the compute component. For compute nodes running a hypervisor, the storage component must provide storage for the hypervisor, which then virtualizes that storage for the virtual machines under its control. For compute nodes running containers, the storage component can be nothing more than the file system drivers built into the operating system kernel shared among the host compute node and the compute instances running in the containers on that host. For compute nodes running on bare metals, the compute instance being indistinguishable from the host compute node, the workload typically has direct access to the host’s storage hardware—indeed, this
is the primary reason for running bare metal compute nodes in the first place, because this allows the workload to operate without the performance overhead of virtualized storage I/O for applications such as high-performance databases.

Beyond the minimum requirement to run a workload in a cloud environment, the storage component as implemented by most modern cloud infrastructure technologies can provide various additional capabilities for advanced operational capabilities, or simply to improve management convenience. For example, most hypervisor implementations support so-called snapshots. These snapshots capture the state of a virtual machine at a moment in time, allowing the user to revert to that state at any time in the future. A modern storage component can implement automated backups by working with the hypervisors to create a snapshot at regular intervals and storing them according to the configured backup scheme. These backups capture everything about the state of the machine at the time of the backup, so that if the backup ever needs to be restored, the workload can begin working right where it left off. Closely related to this is integrated version control, for example, to enable multiple versions of an application to be run at any time in support of development or support efforts.

Another way that the storage component can collaborate with the hypervisor or container is to allow for workload migration among host compute nodes. The storage component enables this by providing some mechanism for different compute nodes to access the storage related to a given workload. This may be accomplished through network file systems (NFSs), shared storage fabrics such as internet small computer system interface (iSCSI) or fiber channel storage area network, or some copy-on-demand mechanism. Although it is managed, it allows the compute component to shuffle workloads among the various compute nodes.

There is a fine line between advanced storage component functionality and functionality more properly attributed to the database component. For example, we’ll discuss key-value stores in the database section, but these may be implemented by the same component that provides more traditional storage mechanisms. Cloud infrastructure technology is developing quickly and gray areas like this abound.

### 3.2.3.2 Security Implications

As with other components of a cloud architecture, perhaps the most critical security-related consideration involves the enforcement of complete isolation between compute instances, workloads, and applications. In the case of storage, the compute component plays an important role in enforcing that isolation; however, the storage component must also ensure that compute instances can access only authorized storage areas, for example, by configuring robust NFS permissions and iSCSI authentication. Malicious code on one compute instance should not be able to access another workload’s data by manipulating the storage infrastructure.

### 3.2.4 Management

The management component of a cloud infrastructure provides the means by which users and administrators can configure and operate all the managed aspects of that infrastructure.
Ideally, all other components of the infrastructure can be managed via the management component. The functionality of the management component may be exposed via graphical user interface for manual operation or via application program interface for automated operation and integration with other tools and technologies. The introspection functionality of a cloud architecture exposes various aspects of the internal operating state of the various components that make up that architecture. This allows application running outside the component to view and possibly modify the running state of the cloud component in question.

**3.2.4.1 Underlying Approaches**

The management component must provide a mechanism for managing the cloud infrastructure components under its control. This includes the following:

- Defining specific workloads or applications as virtual machines, containers, or bare metal nodes, and providing a means to provision CPU and memory resources to these workloads
- Defining network access mechanisms as appropriate for the network components being managed and allocating these network resources to defined workloads or applications
- Defining storage and database units as appropriate for the underlying storage component being managed and making them available to workloads on compute nodes
- Coordinating migration of all these resources for various use cases, including load balancing and outage resiliency

The management component may also manage the introspection capabilities of the cloud infrastructure. These capabilities expose the information about the state of the various components in the cloud infrastructure and enable operations that manipulate or record that state. These capabilities support tools, for example, that can make forensically sound recordings of raw disk blocks used by a virtual machine, or record network traffic generated by that virtual machine. For some use cases that only leverage introspection sporadically, such as digital forensics and incident response, it may be feasible to manage the introspective mechanisms manually on a case-by-case basis. However, for security use cases that depend heavily on routine introspection, such as antivirus and intrusion detection system (IDS), the management component provides critical support for scaling and automating introspective activities.

**3.2.4.2 Security Implications**

Securing the management component is a critical part of securing a cloud infrastructure deployment. It is also arguably the most complex part of a cloud infrastructure to properly secure. That is both because of the complexity of the operations performed by the management component and because of the many interface points the management component must support.
3.3 SUMMARY

PCIs can be extremely complex, yet it’s important to understand it intimately in order to secure private clouds properly. It’s impossible to cover the topic thoroughly in one chapter, but this information will get you started and help you determine what knowledge gaps to fill in more detail.

REFERENCES