Private Clouds: Selecting the Right Hardware for a Scalable Virtual Infrastructure

Greg Shields

sponsored by

Realtime publishers

Copyright Statement

Chapter 4: Building the Business Case for Scalable Virtualization Using Private Clouds

At first glance, this book might appear to be all about hardware. Much of its first three chapters are dedicated to an explanation of hardware resources. I talk at length about hardware supply and demand inside the virtual environment. Indeed the book's name itself references *selecting the right hardware* to help you create a scalable virtual infrastructure.

But that conversation on virtual hardware is only one facet of the story. I mentioned in Chapter 2 that a private cloud is at its core little more than a virtualization technology, some really good management tools, the right set of hardware, and *business process integration*. It is this business process integration that elevates virtualization from simple technology to powerful business enabler. Correctly constructed, a private cloud will create a flexible and infinitely malleable infrastructure for hosting business services. Correctly managed, that infrastructure will always be capable of meeting the computing needs of an ever-changing business.

These are indeed some lofty, feel-good buzzwords, but I find that I honestly believe their promise as they're applied to virtualization. After years in IT, I've heard vendors using statements like these many times to describe something new that will "completely change how you do IT." I'm sure you have too. Yet unlike many other promises, the elevation of simple virtualization to private cloud computing appears to be one of the rare situations where real business value quickly arrives once you make the jump.

That said, you can't get business enablement solely through racks of aluminum and silicon. Nor can you get it from virtualization software alone. You'll need the toolsets that convert these items into something that usefully transacts business processes. It also needs to be affordable. As you know, every IT purchase is scrutinized through the lens of business requirements (or, at least we hope that's the case). No business invests in a virtual infrastructure without being shown a return. No business matures that virtual infrastructure to a private cloud without seeing even greater return.

It's my goal in this chapter to help you build that business case. I'll do so by showing you how a converged infrastructure directly and positively impacts your business processes. That infrastructure's combination of asserted resource supply alongside templatized resource demand will allow you to apply benefit-side dollar figures to what might otherwise be only costs.

Modularization Completes the Mission of Virtualization: "Converging the Infrastructure"

Virtualization has long been hailed as "the first technology in years whose benefit can actually be measured by the corporate accountants." These are in fact words I wrote during virtualization's early years. Yet in reality, it took a period of time for them to ring completely true. In the beginning, virtualization's costs were substantial while its actual business impacts were in fact minimal. Left unchecked, many investments created cost, and sometimes massive cost, as they promised to reduce them.

Much of that initial misalignment between investment and benefit existed as a function of virtualization's heavy technology focus. IT organizations saw virtualization as an assist to their processes, keeping servers running where they might otherwise not. Rapidly deploying servers (and, later, desktops) meant IT could meet business demands; however, many of those computers weren't built with performance and capacity in mind. Overprovisioning of services was common. Tools to right-size demand to supply simply didn't exist.

Much has changed since then. As you'll discover in a minute, the tools now exist to make smarter decisions in service deployment. Further, modularization itself completes that original mission of virtualization by finally and completely converging the infrastructure into a greater whole.

Modularization creates a fluid pool of resources upon which IT services can be delivered, but more importantly, it enables the rational quantification of that pool. This is a good thing because *once you get a number you can trust, you can then apply business metrics to that number.* Quantifying resource supply and demand is crucial to this recognition and to unlocking the benefit-side calculations the accountants desire.

Until very recently, however, getting that trustable number hasn't been an easy task. Management toolsets from virtualization's first-party vendors got us part of the way. If you take a look through the statistics gathered through VMware's vCenter Server console or Hyper-V's VMM (see Figure 4.1), you can indeed see numbers that quantify processing, memory, networking, and storage usage.

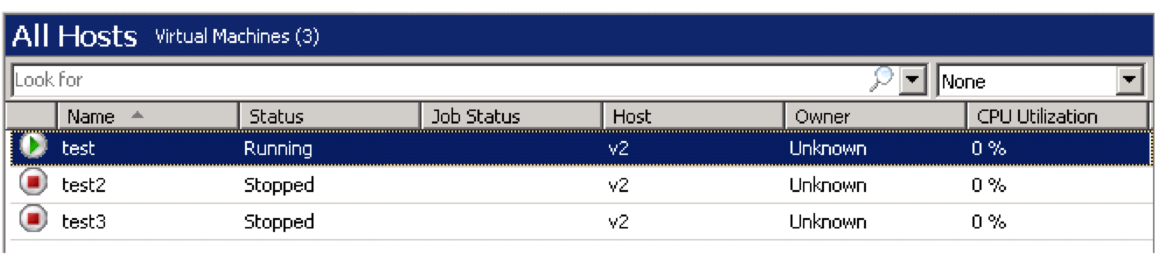


Figure 4.1: CPU utilization in System Center Virtual Machine Manager.

At issue is the actual usefulness of these metrics, as well as how they are gathered. Capacity information gathered through these tools is sourced from a layer above the virtualization stack. From that positioning, you can indeed learn about a virtual machine's resource consumption, but limited is your ability to see its relation to other virtual machines in the environment. In short, with first-party tools, it is exceptionally difficult to see the big picture.

Needed for a complete understanding is an additional bottom-up approach to resource monitoring. That bottom-up approach, from the perspective of the hardware itself, gives administrators a much greater understanding about aggregate resource supply and demand.

Figure 4.2 shows an example aggregate view one could potentially gather using this bottom-up approach. By converging the infrastructure beneath a unified and hardware-centric management solution—one that includes visibility into every hardware and virtualization component—it becomes possible to more quantitatively identify resource supply.

Metric	baseline
Scenarios Overview	
Headroom	★★★★★
Cpu (cores)	31
Memory (GB)	45.00
Number of Physical Servers including VM hosts	20
Number of VM Hosts	0
Number of VMs	0
Number of DRS Clusters	0
Number of VM Hosts in DRS Clusters	0
Number of VMs in DRS Clusters	0
Number of HP-UX instances	0
Number of Windows instances	20
Number of Linux instances	0
Power and Energy	
Average W (Watts)	3,728.89
Average kWh/month (kWh)	2,684.80
kWh/month HVAC (kWh)	1,610.88
Total kWh (kWh)	4,295.68
Energy cost/month ($)	343.65
Absolute Utilization	
Average CPU (cores)	4.84
Average Memory (GB)	30.88
Average Power (Watts)	3,728.89

Figure 4.2: A virtual environment baseline.

The image in Figure 4.2 represents a scenario that has been created for 20 physical servers. The capacity of those physical servers and the Windows virtual machines residing on top is documented in terms of processing and memory consumption as well as power consumption. Not shown in this scenario is storage and networking demand data; however, such data can be gathered using similar instrumentation.

Data such as this is much better gathered using monitoring components that are built directly into each module and gathered through a management solution that is hardware aware. Taking this data a step further, a hardware-centric management solution presents the ability to compare scenarios across different hardware options. Figure 4.3 shows an extension of the scenario created in Figure 4.2. In this example, average CPU, memory, and power are compared across three different hardware options. This kind of data is essential in helping administrators decide where best to position virtual workloads for the best return on investment.

Absolute Utilization			
Average CPU (cores)	5.05	6.01	4.84
Average Memory (GB)	53.28	49.82	30.88
Average Power (Watts)	1,627.38	866.94	3,728.89

Figure 4.3: A resource comparison across three hardware options.

That same bottom-up approach assists in other ways as well. It can further enhance the level of intelligence in aggregating hardware into resource pools. Simply put, a hardware-centric management solution can know more about the hardware that makes up the private cloud. With its constituent information readily available, a hardware-centric management solution can be smarter in constructing pools out of existing resources.

Such intelligence is seen in Figure 4.4, showing how a resource pool might be created out of server blades in a processing enclosure. There, instrumentation within individual enclosures exposes greater detail about the hardware in the environment—enclosure names, number of bays and locations of individual servers, quantity of memory per individual server, and so on. With this information, pools can be created based on requirements such as segregation, security, and fulfillment of business rules that have nothing to do with technology.

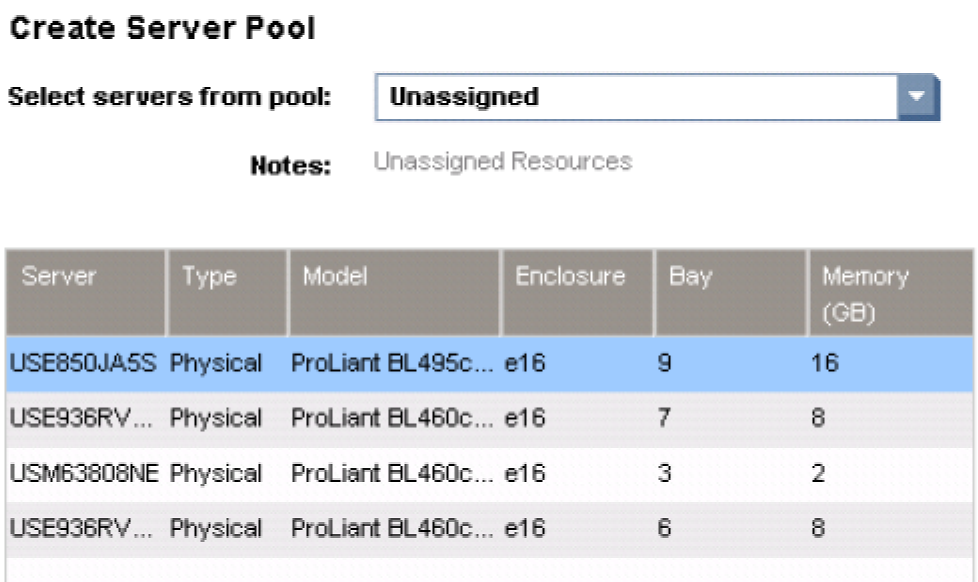


Figure 4.4: Resource pools are created out of individual modules.

Contrast this business-centric data with the technology-centric data you're limited to seeing inside the first-party tools. Such tools are typically limited to recognizing virtual hosts as individual machines and not necessarily as modules that make up a larger infrastructure. As the following sidebar explains, module awareness grows increasingly important as the environment grows in size.

Module Awareness Gains Importance as the Environment Scales

A simple virtual environment that's comprised of 5 or 10 servers is relatively easy to administer. Any administrator should be able to recognize which virtual host corresponds to which piece of physical hardware. However, this mapping between logical management element and physical device becomes much more complex as the environment scales upward. With hundreds or more hosts, figuring out which in-the-management-tool icon corresponds to which physical hardware becomes administratively challenging.

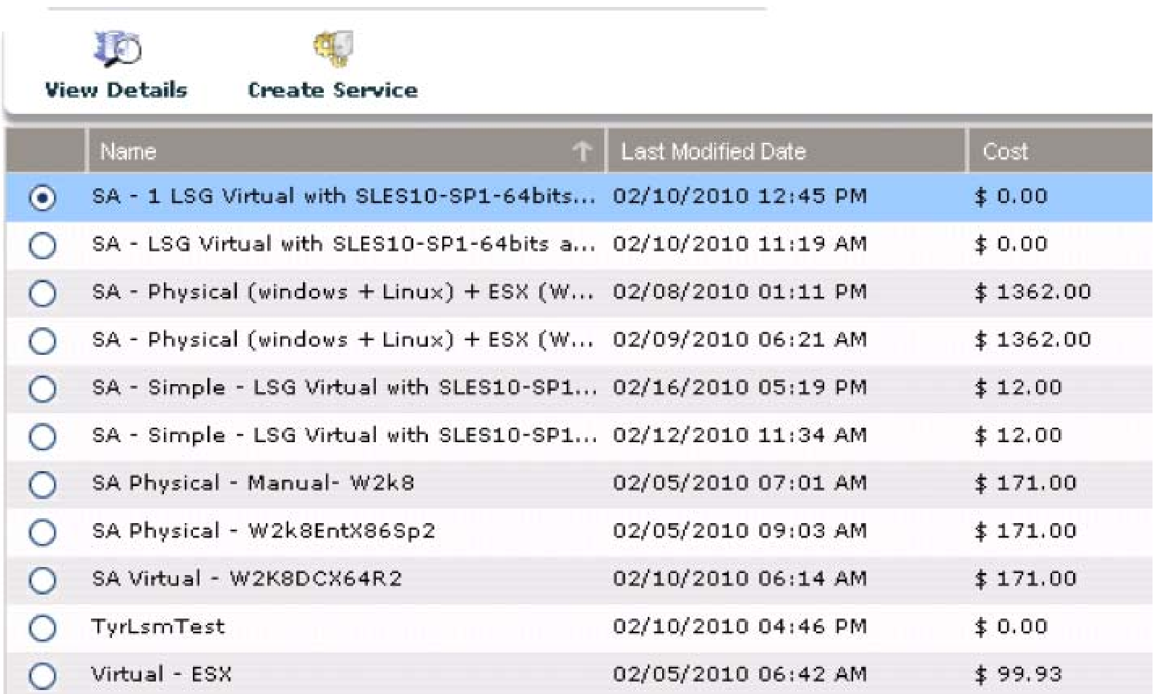
Further, environments that scale introduce other issues as well, particularly as they age. Virtual machines in one enclosure may not move between virtual hosts or enclosures, either due to technology limitations or rules of regulatory compliance. Some virtual workloads may need to be physically separated from each other to ensure security between them. Even just finding the right virtual host to replace in the case of a failure can be an exercise without knowing where to look.

As environments scale, administrators require smarter toolsets that include greater hardware awareness. Those toolsets will not be provided by virtualization vendors. Rather, virtualization vendors will provide mechanisms to wrap the functionality of their toolsets into the enterprise management suites of hardware vendors.

Templatizing Makes Virtualization's Investment Recognizable

Even with hardware-centric management, we're only halfway to a business case. The previous statements discuss formalizing the supply side of virtual resources, but they do nothing to outline how a private cloud's resource demand becomes measurable by the business. I said back in the first chapter, "Modular hardware from top-tier vendors should arrive with an assertion regarding how many resources that hardware will contribute to the private cloud." But what I haven't yet said is that those same top-tier vendors must also assert what quantity of resources different workloads should likely require.

You get that information by using pre-generated and pre-engineered server *templates* inside your hardware-centric management suite. Figure 4.5 shows a view of what those templates could look like, along with some very interesting information that you wouldn't normally get through a first-party management solution.



	Name	Last Modified Date	Cost
◉	SA - 1 LSG Virtual with SLES10-SP1-64bits...	02/10/2010 12:45 PM	$ 0.00
○	SA - LSG Virtual with SLES10-SP1-64bits a...	02/10/2010 11:19 AM	$ 0.00
○	SA - Physical (windows + Linux) + ESX (W...	02/08/2010 01:11 PM	$ 1362.00
○	SA - Physical (windows + Linux) + ESX (W...	02/09/2010 06:21 AM	$ 1362.00
○	SA - Simple - LSG Virtual with SLES10-SP1...	02/16/2010 05:19 PM	$ 12.00
○	SA - Simple - LSG Virtual with SLES10-SP1...	02/12/2010 11:34 AM	$ 12.00
○	SA Physical - Manual- W2k8	02/05/2010 07:01 AM	$ 171.00
○	SA Physical - W2k8EntX86Sp2	02/05/2010 09:03 AM	$ 171.00
○	SA Virtual - W2K8DCX64R2	02/10/2010 06:14 AM	$ 171.00
○	TyrLsmTest	02/10/2010 04:46 PM	$ 0.00
○	Virtual - ESX	02/05/2010 06:42 AM	$ 99.93

Figure 4.5: A set of server templates.

In the figure, you can see templates for Windows 2008 servers alongside those for ESX and SLES servers. Each template is created out of a set of known characteristics that any server will require: processing, memory, networking, storage, provisioning instructions, and so on.

But here's where things get terrifically interesting: *Notice the dollar amounts next to each template.* Recall that one of the tenets of our private cloud environment is an alignment of computing workloads with business rules. We have at this point quantified our supply of resources through the modular approach. We know how much those resources cost because they're sold as purchasable units.

Figure 4.5's dollar amounts provide the other half of the equation. In it, the level of resources needed by any template are known and quantified. I know, for example, that a Windows Server with Microsoft SharePoint requires a specific quantity of processing, memory, networking, and storage. Because I know how much those resources cost, *I can now assign a dollar value that recognizes the cost in deploying that workload*.

Figure 4.5 shows a simplistic view of that information. In it you can see, for example, that a very basic Windows 2008 Enterprise Edition SP2 server might cost $171 to deploy. Those dollars are representative of the resources required to deploy the server, and comes out of the total cost invested into the virtual environment. A more expensive server might cost $1362 or more. In effect, these measurements represent the actual costs "that can be measured by the corporate accountants." That valuation becomes the basis of your business case.

Acquiring and Using Templates

Obviously creating those templates requires a lot of work in researching best practices. Important to recognize is that those same engineers who create your private cloud's modular hardware are uniquely qualified to also create the templates.

Here's how that template creation process might work: You already know that resource amounts will be very different depending on the need. That said, known workloads such as Microsoft Exchange atop Windows Server, Apache atop SLES, Microsoft SharePoint atop Windows Server, and so on tend to have reasonably well-defined resource requirements. Best practices exist, even if they're not intimately understood by your internal IT teams. A smart hardware vendor is uniquely capable of making the investment in consolidating best practices into sets of templates. Once created by the vendor, all your engineers need to do is download them and add them into the interface.

Once these templates are added into the solution's console, provisioning virtual workloads requires little more than invoking a template. Figure 4.6 shows a logical representation of how this might work. In it you can see that an administrator selects an Exchange template from a library of those downloaded from the vendor. Other templates might be available for Citrix XenApp, SAP, or Red Hat services; anything that has already been created and made available. Contained within that template are the necessary characteristics needed by the workload.

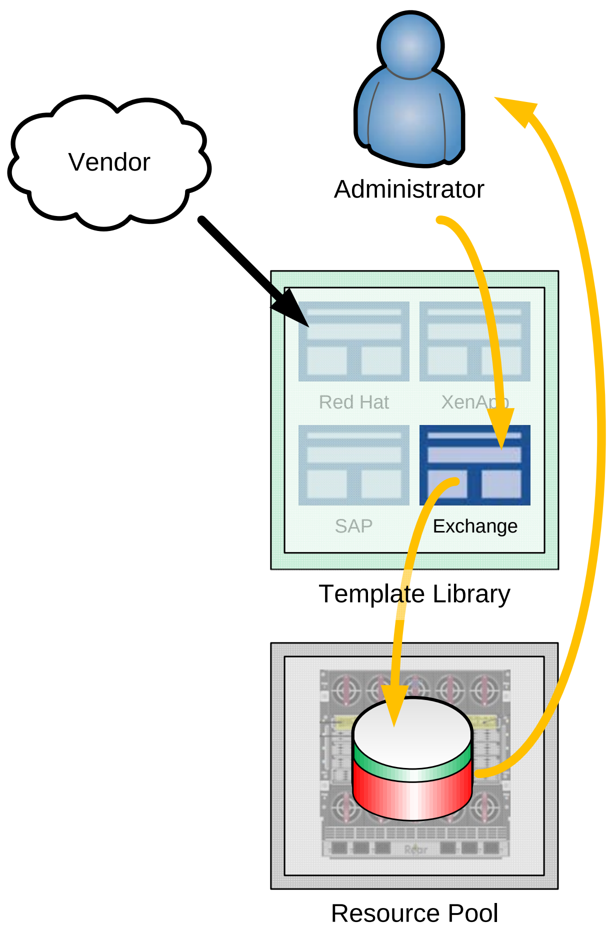


Figure 4.6: Logical template constructs provision actual resources.

Also contained within the template are instructions for functionally deploying the server and its applications into an available resource pool. Figure 4.6 shows how that template reconfigures the resource pool and eventually delivers control of the completed service back to the administrator.

Templatization Enables Smarter Reactions

This notion of provisioning templates isn't new. Templatization is an activity that's been part of virtualization since its beginnings. Because a virtual environment can create new servers and desktops through a simple file copy, most IT teams have already created their own suite of virtual machine "templates" as sources from which production computers start their lives.

Different here are how your teams' sources are created. In the traditional approach, IT creates source machines from individual components in what can almost be a metaphor for the white box hardware construction discussed in earlier chapters. Essentially, your teams create their templates based on best guesses and the limited information they have available.

Different in the vendor-driven approach is that templates are instead generated based on per-workload best practices. Similar to pre-engineered modular hardware, server templates can be created with greater intelligence. That intelligence can be delivered based on the research investment by the manufacturer and not your internal teams. Not to mention saving you time and money.

Further, every template starts its life in a similar fashion, making it documentable for configuration control purposes. Figure 4.7 shows an XML file that contains sample instructions used in constructing an Exchange 2010 server. You can see that this template has been specifically designed for creating an Exchange service for 5000 users using 1GB mailboxes. This template doesn't stop there. It in fact creates two Exchange servers that are connected into a Database Availability Group, which protects two copies of the mailbox data. Hub Transport and Client Access Server roles are additionally created through the deployment of this template.

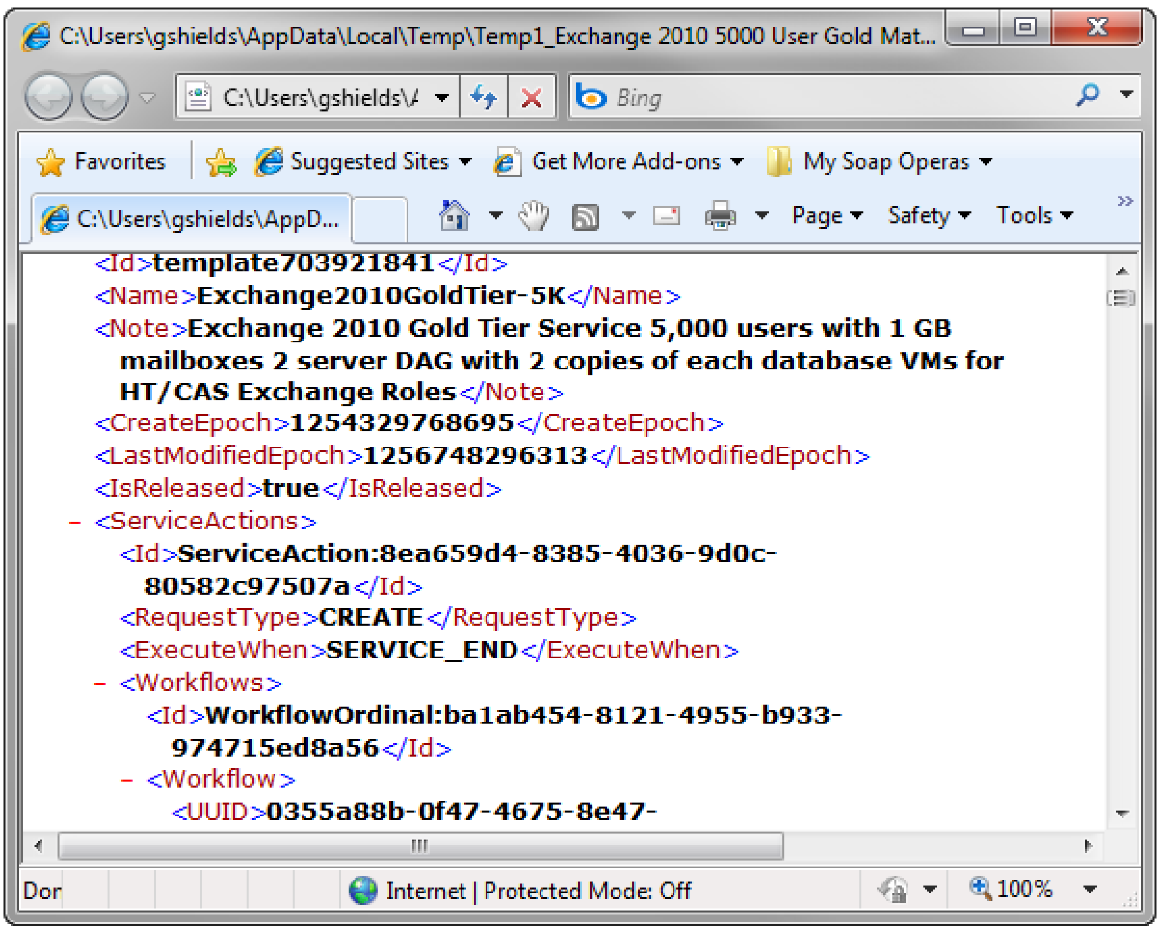


Figure 4.7: A configuration-controlled deployment template.

Constructing all of these pieces yourself will require research and investment. Using an existing template means quickly spinning up a new service as it is needed.

Enhancing Management Control: Physical with the Virtual

Now this is obviously a great solution if that's exactly the type of service that you need. It's also a lot of activity that can and will occur through the invocation of a single XML file. But every deployment is generally just a little bit different. Your 5000 user Exchange organization might want slightly larger or smaller mailboxes or additional protected copies of mailbox data. These aren't necessarily virtual machine configurations; they may be Exchange configurations. In either case, they are an important part of deploying Exchange via a template.

That's why this conversation should lead you to a follow-on question. That question relates to c*ustomization*. Or, more specifically, how can you converge the construction of the virtual environment with those special configurations required inside the actual "physical" server? It is here where a truly converged infrastructure joins its virtual activities with the on-system configurations required inside each server.

A solution that works for your business must be able to bridge both the physical world and the virtual world. You require this for a number of different reasons; for example, not every server will get virtualized and not every configuration is one that happens at the hypervisor layer. A workable solution will provide a workbench for evolving templates—as well as other configurations—to exactly what your business needs.

Those customization activities can be wrapped up into a hardware-centric management solution's console, such as the example map you see in Figure 4.8. There, a deployment activity can be augmented with additional logic that creates services, adds servers, and modifies disks and network connections. Particularly in the case of complex services such as those that spread across multiple computers, disks, and networks, constructing services with the aid of graphical tools means ensuring that all the pieces are always correctly connected.

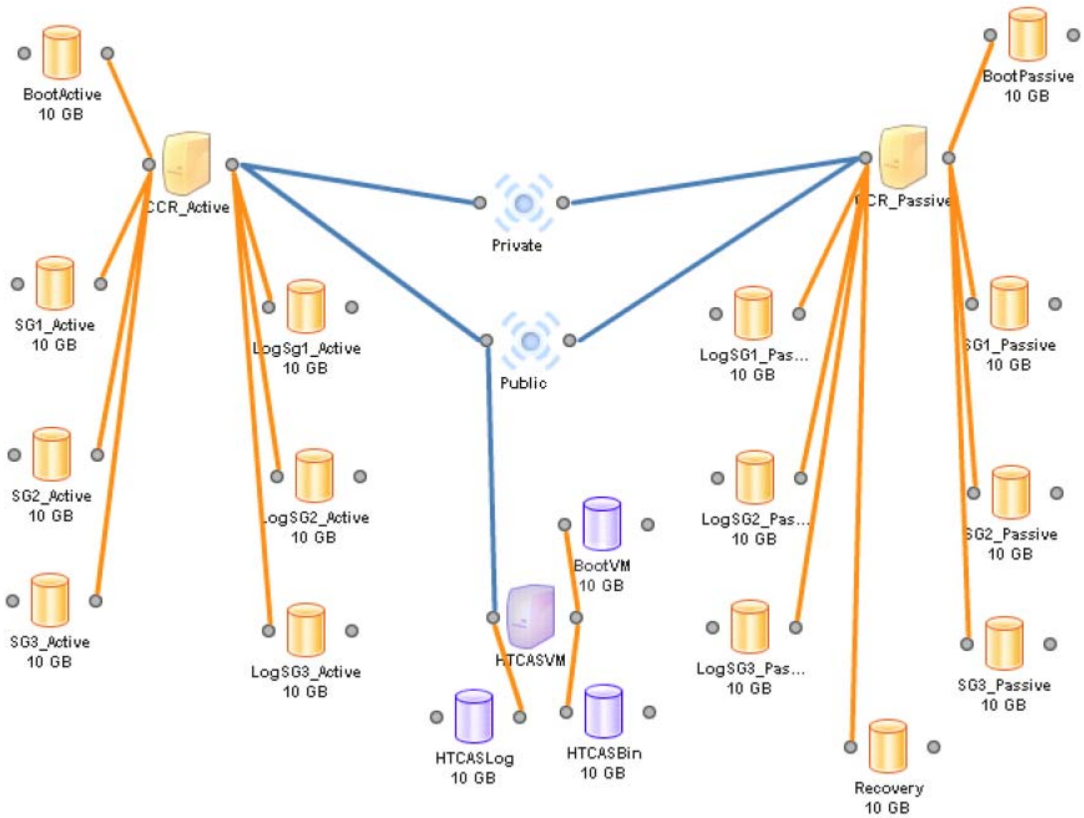


Figure 4.8: Constructing templates through graphical means.

Manager of Managers

It is also important to recognize that a workable solution can't do everything all by itself. In fact, a best-in-class solution shouldn't attempt to accomplish every task all on its own. A much superior solution, one that has the ability to scale with your needs, will be one that instead works with your existing management services (see Figure 4.9).

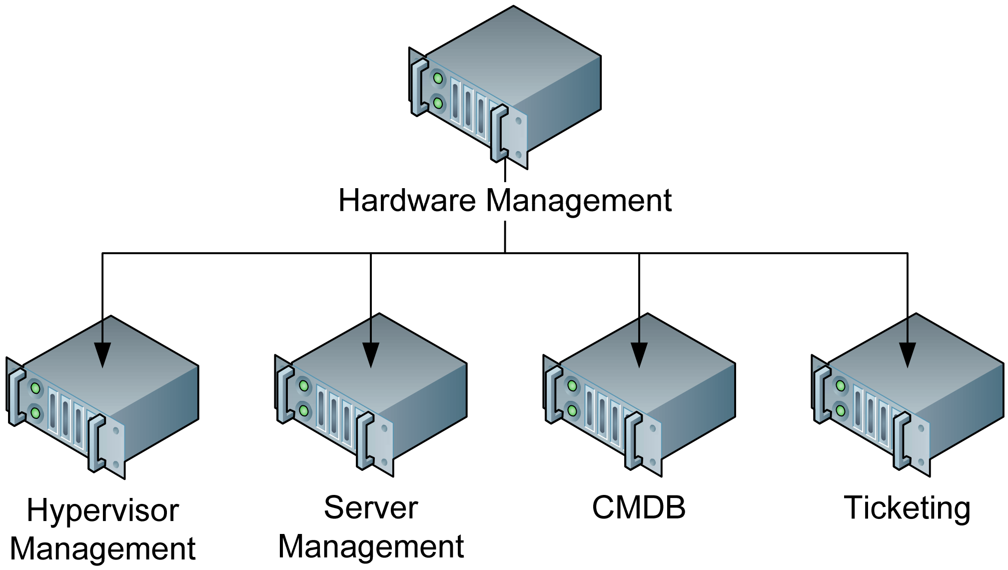


Figure 4.9: Hardware management ties into existing services.

A hardware management solution can serve as a kind of "manager of managers" for the services that already exist in your environment. For example, those services can be hypervisor management for enacting actions on virtual machines, server management for making changes to server configurations, CMDB for discovering, documenting, and governing change control across systems, and even ticketing systems for managing workflows across teams of individuals. A well-designed hardware management solution will interface with these already-existing services to unify management underneath a single pane of glass without requiring a reconfiguration of existing management service.

Division of Responsibility

Any private cloud environment also comes with a cast of actors, one that grows in size with the environment itself. Governing those actors then becomes an important function of a private cloud's management solution. Using templates makes this division of responsibility very simple because different individuals have different roles to play in the environment.

Just a few of those roles are explained in Figure 4.10's workflow diagram. In that diagram, you can see how three different role types can be strung together to complete the workflow of provisioning a new service. In that example, the responsibility of an IT Architect is in defining and constructing the library of service templates. An IT Architect may be the individual responsible for downloading and customizing vendor templates and revising them for your individual company needs.

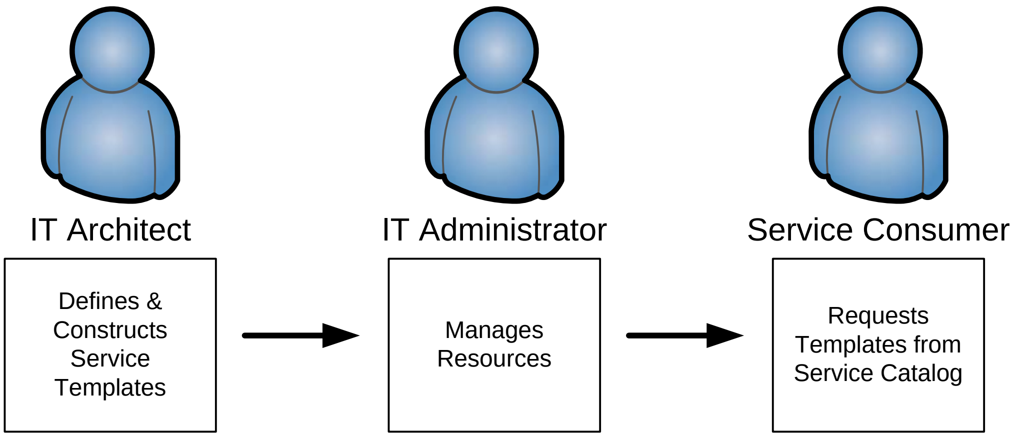


Figure 4.10: Different roles for different individuals.

That individual may also be responsible for applying cost information to templates, using the management solution's built-in calculations to apply financial logic where appropriate. Segregating financial calculations into the IT Architect role ensures that cost information remains in the hands of a protected few individuals.

Once templates are created, customized, and fully defined for the environment, they can be made available for Service Consumers (at the right in Figure 4.10). Private cloud computing often comes equipped with self-service tools that enable service owners to request the services they need. Leveraging templates enables IT to deliver services to those owners based on investment.

As this book has already discussed in its earlier chapters, templatizing also provides predictability in terms of resource use. Managing those resources is a job for IT administrators. This team's job is to ensure the upkeep of the private cloud components, along with ensuring resources remain available for current and future requests. Figure 4.11 shows an example of the types of charts an IT administrator might use to manage aggregate resources across the entire private cloud. There, CPU, memory, and network I/O utilization are measured across the entire environment.

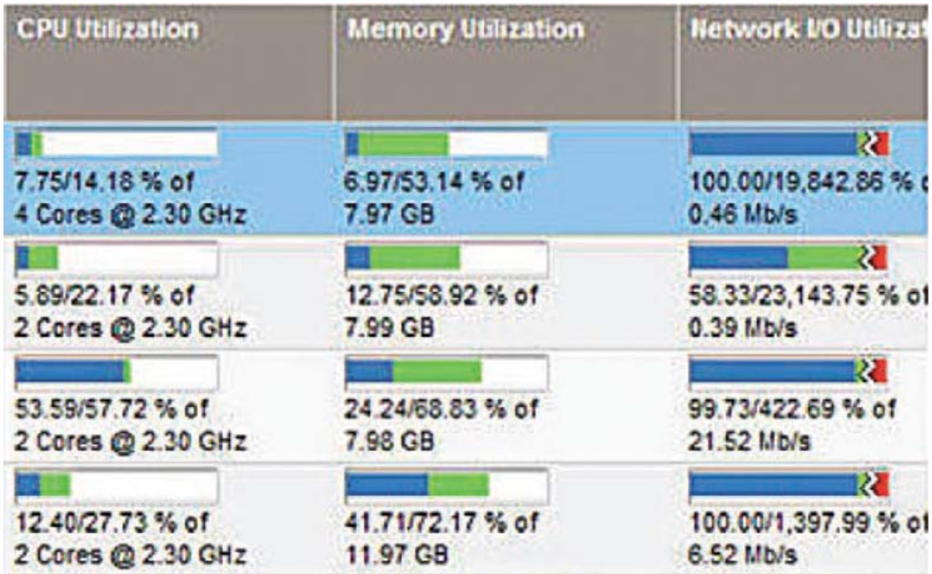


Figure 4.11: Managing aggregate resource use.

Cost Predictability

Keeping resources well-managed is obviously important for performance and capacity management. But environments that can tie resource supply and demand to financial figures also gain an additional benefit: *cost predictability*. Knowing the cost of resources in relation to their demand from services makes budgeting and planning for expansion both extremely quantifiable activities.

That's an extremely attractive capability because it has long been one that IT hasn't been capable of delivering on in the past. Combining private cloud computing with modular hardware, IT for the first time can better react (or, in some cases *actually react*) to budgetary demands—as opposed to its old line of thinking: "Give me the budget and you'll get the service when we get around to upgrading our infrastructure."

Leveraging this combination along with a smart set of management tools, IT can better make long-term use of capital. It can better align its expansion activities to actual resource needs as well as quantify the resource needs its customers anticipate in the short term.

Sprawl Prevention through Service Leasing

The capacity for rapid service delivery also brings about one very new way in which IT services can be delivered to customers. That new way involves the concept of *service leasing*. In the traditional IT model where services and servers were inexorably linked to each other, any creation of a new service typically meant the long-term to infinite-term management of that service after creation.

A good friend and former manager of mine used to repeat a saying when it came to such new service requests: "Anything temporary is permanent." That saying has held true even unto today, when services that were intended for a single use and/or a short lifespan remain operational and within IT's management budget long past their time of usefulness.

Service leasing embodies the concept that services by nature can always be made temporary. At their point of provisioning (see Figure 4.12) service templates can be configured with a Service Lease timeout value. Such a timeout value automatically puts a timer on the length of the service, ensuring that future decisions about the service's usefulness will be considered at some point down the road.

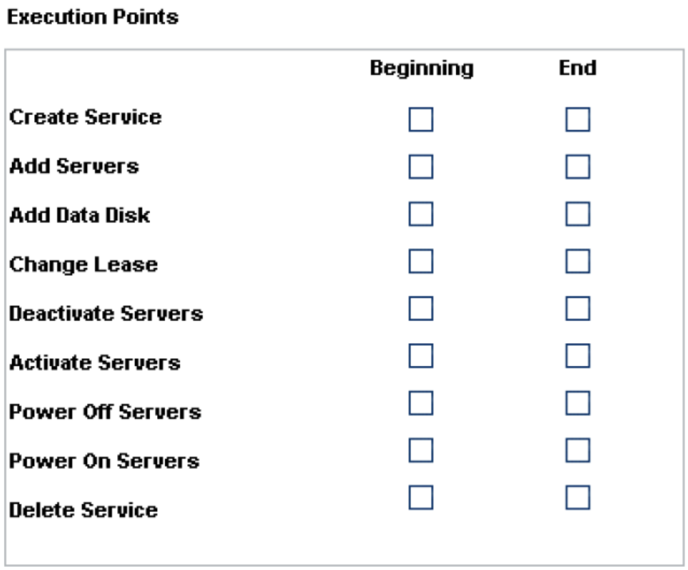


Figure 4.12: Creating automations in templates.

Service leasing also protects the virtual environment against one of virtualization's biggest growing pains: *sprawl*. Virtual sprawl, as you've probably already experienced, happens as easy-to-build virtual machines begin to grow in number past the point of successful management. In essence, when a thing becomes easy to do, we do it. Virtual sprawl represents a major negative impact on the cost-to-benefit calculations of a virtual environment; thus, eliminating it where possible is absolutely necessary to protecting your investment. A smart management solution will provide one or more mechanisms to attach timeout values to services to protect your private cloud from virtual sprawl.

Building Your Private Cloud Business Case

In the end, this book isn't necessarily all about hardware. It is, however, all about evolving past the white box tactics that our industry has fallen into for a second time. Those white box tactics first manifest in the ways virtualization's hardware can be constructed. Once built, those same white box tactics can take root in the ways a private cloud is managed. Elevating your management philosophies towards a more predictable infrastructure requires selecting the right hardware for a scalable virtual infrastructure. It also requires selecting the right management for scaling that hardware correctly.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.