Chapter 2

Implementing the Solution

Chapter 1, “Laying the Groundwork,” laid the foundation for a virtualization infrastructure that will deliver not only on today’s requirements but also allow for growth into the coming years.

This chapter focuses on the implementation of the solution. We go through the process of following the design blueprint and finish with checking our implementation for mistakes. In between, we talk about some of the considerations of being the implementer of the solution. The implementer of the solution might or might not also be the design engineer and we talk about both circumstances. In many cases, it is irrelevant as similar processes need to be followed either way.

We also talk about a number of real-life cases where things did not go as desired and talk about what could have—and should have—been done to avoid such issues. It is important to note that although some of this is directly relevant to a solution that has just been implemented using the processes described in Chapter 1, the information is also relevant to an infrastructure that might not have been designed with the understanding of the concepts and methodologies outlined in the previous chapter.

Following the Design Blueprint

The design blueprint refers to a set of documentation containing configurations, procedures, and processes that are being implemented. This documentation is a result of the design phase of the project with its roots in the functional requirements for the design. Although this section is not about the design, it is important to reiterate the importance of the functional requirements. Functional requirements are not something that will be handed over on a sheet of paper to you. Gathering them requires reviews with both stakeholders and technical staff.

As the implementer of a vSphere-based design, you might have been the designer as well. In this case, you will have a set of design documentation that you created and will be familiar with the functional requirements as defined. In other cases, you might be following the design blueprint that has been provided to you by another member of your organization. In either case, it is important to review the design deliverables before proceeding with the implementation.

Reviewing the Design Documentation

The design documentation should consist of the following at a minimum:

- Pre- and postimplementation performance comparison
- Site survey documentation
- Architecture documentation
- Design blueprint based on capacity planning results
- Design implementation procedures
- Design verification
- Installation and configuration procedures
- Application test procedures
- Operating procedures and guidelines
- Stakeholder review meeting notes

Some of these will end up being deliverables to stakeholders while some will only be part of a guided transition between the design and implementation persons or teams. They range in depth and breadth of being simple diagrams that may depict logical or physical viewpoints of the design to detailed information on port group configurations and settings. It is necessary to have both low- and high-level viewpoints of the design to ensure a successful implementation.

Stakeholder Review

A stakeholder review can happen several times throughout the design process. The initial importance of the stakeholder review is in gathering information that helps define the functional requirements of the design. Further reviews with stakeholders serve to verify the design is meeting these functional requirements while taking into consideration the constraints placed on the design and any associated assumptions. For example, many times we are dealing with timelines or budgets that constrain the design. This results in functional requirements that are not going to be met but need to be identified and discussed with stakeholders. These discussions with stakeholders are at a high level initially until a full technical review of the constraints and assumptions has occurred.

The result of these meetings and the process should be fully documented in the design documentation and presented as a deliverable. The main purpose of this is to clarify the design decisions made. This may be for present use or future use. Although a sign-off process should occur and everyone should understand the design decisions that were made, that does not always happen. As a result, it is necessary to have historical documentation on the justifications behind not meeting certain functional requirements. This need could occur any time from implementation to months later down the line.

It is also necessary to have a good understanding of these decisions by the implementing engineer when different from the design engineer. The following sections speak more about technical reviews, which help fill this gap. Verbal conversations, though, fail to capture the entirety of a situation and documented knowledge of the decisions is critical to accurately following an implementation plan. Ultimately, you must be able to communicate to stakeholders the changes that occurred. A failure to do so may result in a misunderstanding of the current solution being implemented and a lack of understanding of why a functional requirement was not met.

Functional Requirements

The primary end result of stakeholder reviews is the definition of the functional requirements for the project. A design seeks to meet its functional requirements while taking into consideration constraints to the design. Not all functional requirements need to be met, but the goal should be to do so.

Functional requirements are unique to an organization's project. With that said, several functional requirements are common for virtualization projects:

- Reduction of physical server real estate
- Reduction of power costs associated with many physical servers and associated costs of cooling those servers
- Reduction of total cost of ownership
- Enablement of High Availability
- Enablement of dynamically balanced workloads
- Enablement of a disaster recovery solution

Constraints

Items that limit your design are considered constraints. You should outline constraints in your design documentation and review them to make sure all constraints have been fully identified. Reviews with stakeholders are also a great time to discuss constraints as the design progresses. Proper designs identify how constraints violate other requirements of the design and the effect that such constraints will have on the design. It should be noted that existence of a constraint does not mean the constraint has to exist. Whether political or technical, an effort should be made to explore removing the constraint.

NOTE

It is important to note that during a virtualization project, items not part of the project often creep in. These are different than constraints and the functional requirements they constrain. These creeps of scope should be left for another project to ensure the focus is on meeting the functional requirements of the project as designed.

Again, documentation of constraints in the design is critical for both a historical understanding of the decisions made and as information to be utilized during the deployment. Thinking about constraints is often left to the design phase of the project and not always considered during or directly after the implementation. This is a mistake that can be avoided, though. Consider the following case.

Constraints Case Study

A solution has been designed for an environment where a major limitation has been identified in terms of available networking infrastructure to support the 12 new vSphere hosts. In conversations with the network administrator, it was revealed that there was a major constraint in terms of available gigabit network ports. Unfortunately, despite discussing the benefits that a server with six NICs would have, the additional switching that would need to be purchased to support the infrastructure was deemed unnecessary at this time for a virtualization initiative. With all the systems that will be decommissioned when they are converted from physical to virtual, there would be ports freed on the switching, but, unfortunately, these have all been identified as Fast Ethernet (10/100 Mb) ports. The servers are ordered with the six NICs as desired as a result of the increased benefits of redundancy that can be accomplished. Only four of the six NICs, however, will be hooked into the gigabit switching at the datacenter.

As a result of the networking constraints, the design lays out the management network spread across two Fast Ethernet switches. The remaining four ports are set up for virtual machine networking, vMotion, and IP storage networking. With six NICs hooked into a gigabit infrastructure, we normally would take management traffic and place it down the same set of networking as vMotion traffic because those two traffic types are the two that will play together nicely the best. With the constraint of Fast Ethernet networking, this is not possible. With the need to use IP-based storage to support NFS, the design will place IP-based storage on a single, dedicated NIC as will vMotion. For both of these, the other's active NIC will be used as its standby. The remaining two NICs for each server will be dedicated for virtual machine networking traffic. This might not be an ideal design, but it meets the functional requirements of the customer considering the constraints we have in networking.

As with any organization, things change and multiple projects tend to occur at the same time. Often, different teams or even individuals on the same team might fail to keep the other one in the loop. Perhaps both individuals are just so busy that they don't really have the time to discuss what is currently going on. In this case, the latter occurred.

It turned out another long-standing project had an impact on the network infrastructure that would be in place before the implementation was completed. The other project had to do with a turnkey vendor–provided solution that was now being upgraded. It consisted of more than ten physical servers, which, of course, in turn required a lot of network ports. Fast Ethernet ports were in abundance, but with the new version of the solution, gigabit networking for the switching was now required. They balked at the solution and chose to hold off on the upgrade until the vendor came down in pricing and, lo and behold, additional gigabit switching was added to their infrastructure.

Even though the individual working on the turnkey solution project was involved in the original design discussions and knew the vSphere design could really use the faster networking for all of its ports, months had passed and this constraint that was put into place had long been forgotten.

Unfortunately, by the time this came to light, production workloads were already placed on the infrastructure so individual vSphere hosts did have some down time. Thanks to the technologies within vSphere, the use of Maintenance mode and vMotion ensured that none of the virtual workloads had any down time. With careful planning, the design was changed and documented to provide a much more reliable and quicker networking access for each of the hosts all around.

This is a case where some due diligence leads to a big win. Thanks to following a process where we always verify our work throughout the process, even our constraints, we were able to resolve a less-desirable part of the design that no longer had to be the way it was.

Technology changes fast and, as a result, so do many things about an infrastructure. This case could have easily gone the other way and someone working on the other project could have taken up all those gigabit networking ports before the vSphere implementation took place. Always be cognizant that many things are going on in an organization's technology infrastructure at any given time. What you are working on will be affected by the work of others. Your actions will also have the same effect on others' work.

Constraints will be different for every organization and every project, but there are several common constraints for virtualization projects:

- Vendor preference
- Budget
- Organizational best practices and policies
- Government regulations
- Existing network infrastructure (that is, the lack of gigabit or 10-gigabit networking or limited networking capacity)
- Existing storage infrastructure (that is, the presence of only a certain type of storage with limited capabilities, limited performance, or limited space available)
- Existing host hardware

Technical Review

A technical review can occur several times throughout the design process. Technical reviews should be focused on meeting the functional requirements of the business while considering the constraints. Whether considered formal or not, you will find that you perform technical reviews following any stakeholder review. These may take form via simple verbal communication or may be a formal meeting. Regardless, any time a discussion about functional requirements and constraints occurs, a later discussion will take place to consider technical ramifications and how to proceed with the design and implementation. The process will occur several times, with updates being done to the design after each technical review iteration.

Technical reviews are a good time to review all assumptions before proceeding with an implementation. The goal should be to resolve as many of the assumptions as possible to eliminate the risk that assumptions themselves pose. Regardless of the formality of the meetings, they should be documented and distributed to all stakeholders. This documentation should be targeted. For example, the appropriate documentation will be much different for end users than for technical individuals involved with the project or senior management.

Assumptions

Assumptions in a vSphere design are no different than assumptions in any other part of our life. To make headway with a design, it is often necessary to make certain assumptions. You may assume that your installation media for Windows Server 2008 will be available on the network for installation, or you may assume that adequate network bandwidth will exist for a disaster recovery solution using storage-based replication. Regardless, it is important to have a list of assumptions for the design.

Failing to have the list of assumptions is one problem. Another is a failure to review these assumptions immediately before the implementation of the solution. In many instances, assumptions are items that may be put back on stakeholders, customers, or others for follow-up or completion. Rudimentary items like not having Windows installation media, licensing information, or network cables make it impossible to get past the initial steps of an implementation plan. These failures typically cause hours of lost time. Other items can cost you days or even weeks. For example, consider the following scenario.

Assumptions Case Study

A design is needed to virtualize 150 existing physical servers. These servers are located throughout various branch offices and the business wants to centralize them at its main datacenter, which currently hosts four other existing physical servers. The business is asking for virtualization as a means to reduce its server footprint and save on hardware and power costs. Additionally, the business is requiring the capability to tolerate hardware failures and limited growth within its environment.

Without going into the specifics based on capacity planning results and the required average and peak workloads, you find you are able to virtualize nearly all the physical systems onto 11 rackmount servers. This takes into account both the required average and peak workloads as well as two additional hosts that are provided for growth and redundancy. In the end, it is also determined that two of the four existing physical servers at the main datacenter must remain physical due to internal business policies and vendor supportability.

It is late into the first day of the implementation and you are just finishing up the cabling of the eleventh server. After all that cabling, you think to yourself, “boy, it would have been nice if they bought blades.” You begin powering on the servers one by one and as you begin to hear the loud blast of fans from the sixth server, you then hear the opposite all around you. The UPS tripped due to a load it couldn’t handle. The sound of silence in a datacenter is something you don’t often hear, but it is something you will have to hear about if you are involved.

Maybe as the implementer it wasn’t your fault. You look at the design documentation after the dust settles and see the assumption listed, “Customer has existing UPS that can handle the load of the existing servers. Customer will verify with vendor peak load and acquire additional UPS if needed."

Although it might be true the customer failed to verify the UPS, the implementer has the responsibility to verify these items before implementation. This is the case even if an assumption wasn't listed in the design documentation. In this case, a good place to start would have been to check the model of the UPS. Further verification could have been accomplished by physically checking the UPS's load. Issues with implementing the vSphere solution didn't just hinder the new infrastructure in this case, but brought down the existing physical infrastructure as well.

Assumptions have to be made; you cannot possibly check everything. For some things, however, assumptions should never be made. This is especially true as you consider increasing consolidation ratios in your vSphere deployments. Consider reviewing all assumptions during an assurance review so that all assumptions are understood and signed off on. As the design or implementation engineer, you can trust that the assumption holds true, but you must also verify that it holds true.

Design Deviations

It has been established that at times it might be necessary to deviate from the original intention of the design. Many times, this is the result of a change in functional requirements. Other times, assumptions that were made about the existing environment might have ended up being invalid. Either way, the net result is the same. Something has changed within the environment and it must be properly documented. Furthermore, this change will have an effect on the rest of the implementation and must be properly considered.

When Functional Requirements Change

Functional requirements may continually change during the design phase of a vSphere deployment. These changes can be easily dealt with by properly adjusting and reevaluating the design with any new constraints that the new functional requirements may pose. If the functional requirements begin to change during an implementation, it might be time to formally reengage stakeholders to validate whether the new functional requirements need to be integrated now or postponed until later. There are certainly cases for both depending on the functional requirements that have changed.

Continuing the Implementation Case Study

You are rolling along with your implementation and have about half of the physical servers already converted to virtual machines. In just several months, you'll be finished with the project.

Significant time has already been spent up until this point in creating a design that considers a multitude of factors. Functional requirements have been balanced with constraints to come up with a polished design and implementation plan. Any deviations from this can pose serious risk to the successful implementation of the design. You now are approached about virtualizing even more of the infrastructure than originally intended.

When the solution was designed, it took into account just a portion of the business. With the business operating under several divisions, it has been hard to come to a consensus on a road map for the datacenter and, specifically, virtualization technologies. One division decided it would look in to building its own infrastructure and building a proof of concept for Hyper-V. Therefore, a plan was put in place to move forward based on the assumption that certain servers would be virtualized.

A capacity planner was performed and the information generated was collected and analyzed. This resulted in a design that fully met the functional requirements of the business. A problem arises, though. The proof of concept the other division started didn't match its needs. Knowing the vSphere infrastructure would be set up and ready for more virtual machines also makes the other division think again about deploying its own separate physical servers in the future. Furthermore, with recent initiatives to reduce costs, initiatives have come down to begin consolidating systems to reduce the amount of excess server capacity that is wasted as a result of administrators operating in silos.

The business now wants to virtualize all of the systems together. There is a problem, though, in terms of capacity. The original design was to virtualize 100 servers plus room for growth over the next three years. The new requirements add another 100 servers to the infrastructure, making it impossible to meet the new infrastructure being deployed as designed.

Fortunately, your boss understands that you'll need much more in your infrastructure than you currently have in terms of hosts, CPU, memory, and storage. He says there is room in the budget and you should be able to double the hosts with matching CPU and memory configurations and expand the storage array to meet the capacity of the systems being newly introduced.

After talking with your boss, you give the go-ahead to move along with the implementation despite the new requirements. You are not worried, though, because you now will have double the resources available.

Thanks to the help of IT admins in some of the other divisions, you quickly move through implementing and virtualizing the existing physical server workloads. As you get closer to the end of the project, you begin to receive many complaints about the performance. You hear from people that things were fine several months ago but have progressively gotten worse and worse.

A few small details are uncovered. When the first 100 systems were configured, it was discovered that memory was drastically overprovisioned as was the number of processors. Adjustments were made and then carried over to the second 100 systems that rightsized the virtual machines to match what their expected peak usage would be. It turns out the second 100 systems needed more than this. Several more physical vSphere hosts would be necessary to account for this mistake.

Memory and CPU contention is reduced, but some poor performance is still noticed. The culprit now turns out to be the storage. It turns out the second batch of systems not only used more memory and CPU, but several systems were heavily disk intensive and the infrastructure was now starved for available IOPS.

This is just one example of many we have seen where changing requirements during an implementation can cause unexpected results. In all fairness, a lot of these situations end up being the result of political issues. People who don't understand the technology are sometimes the same people in charge of making technology decisions or responsible for the budgets for technology. In reality, what started as a change in requirements should have resulted in a redesign. A Capacity Planner for the second 100 servers would have been an ideal start.

With that horror story, let's consider the case for halting the implementation.

Halting the Implementation Case Study

You have acknowledged the risk that changing the design may pose at this point, but have realized there are some things you desperately need this implementation to do. You don't have weeks or months to wait and need to begin thinking about this now.

Some individuals in this story might have sworn that the need to do this was right now. We live in a world where technology enables us to move quickly. As a result, expectations can often follow that assume such agility. Although there is often a false idea of urgency around things, there are certainly cases where things might be more immediate. Let's consider the case for halting the implementation.

You are rolling along in your implementation and are just a few weeks away from migrating your physical infrastructure over. During the design, you decided not to incorporate your Microsoft Cluster Service systems. This was a conscious choice after considering the constraints this would place on your design. In particular, the lack of support for vMotion was of most concern because this would make operating and maintaining the infrastructure more complicated.

These are the only physical systems that are to remain after the project is completed. Over a two-week period, you have noticed two drive failures, one in each of the clusters. You find the systems are out of warranty but spare drives are easily ordered and affordable.

A week later, two more fail, both of which are on the same system. However, it isn't a problem because this is a Microsoft Cluster, so everything failed over. You spend a few hours troubleshooting and are able to recover from the failure pretty easily. You certainly have a problem on your hands, though, because the hardware is out of warranty and clearly older than was originally realized. There is no room in the budget to replace the hardware anytime soon and you must react quickly.

You decide it is necessary to stop where you are and not move any of the other physical servers over so that you have enough available capacity for the Microsoft Clusters. Fortunately, several of the servers to be completed are still newer hardware that was purchased only six months prior. These systems are also not business critical and don't necessarily need to remain highly available. You check capacity and see that you have plenty available for the Microsoft Clusters. You carry over these redesign efforts to your storage where you make room for the new LUNs that will need to be created using Raw Device Mapping (RDM).

You migrate the clusters over and everything is up and running. Although you are not exceeding your capacity, you have only a little room for growth. It was probably a good thing you held off from moving any of the other workloads over and they will have to wait until the next budget cycle when you can purchase the necessary hardware.

Although everyone would agree it would have been a much worse idea to take the risk of virtualizing the workloads and exceeding capacity, there are some more details.

After you were committed to the project, a few things were uncovered that enabled you to not only virtualize the rest of the physical workloads, but also to do so without the need to purchase any additional servers. When looking at the requirements of the Microsoft Clustered Application, you found that the recovery time objective (RTO) needed hours and not minutes or seconds as MSCS is appropriate for. VMware's High Availability would easily solve this requirement. Further fault tolerance would be considered in the future if a smaller RTO was desired.

This discovery led to the first key reduction in infrastructure requirement. You were now able to reduce the virtual server imprint from these applications from four down to two. This led to some more available resources, a reduction in your Windows licensing, and the lift of some of the restrictions MSCS clusters placed on the environment. This was immediately a big win.

Although this freed up some resources, it was clear this was not going to be enough to virtualize all the remaining workloads. A funny thing happened when you looked at the configurations of some of the physical servers in play. Several of them were identical in model and configuration minus some memory to the deployed vSphere hosts. Ordering up some additional memory easily resolved that issue.

With the ordering of a few additional vSphere licenses, the infrastructure was now ready and you began assisting in moving the remaining workloads off and redeploying those matching servers into the vSphere cluster.

When Assumptions Prove Incorrect

We discussed earlier in this chapter the importance of validating certain assumptions. Those assumptions that could have a large impact on an infrastructure or pose significant delays to an implementation should always be verified. We also acknowledged that it is impossible to validate every assumption. For example, it can be very difficult to verify Windows installation media is easily available, or you might not have access to check whether adequate network ports exist to install a vSphere host as originally designed.

When you have acknowledged that an assumption you made was not correct, you then must take proper action going forward to reduce any further effect on the implementation. This starts by documenting the change. Unlike a change in functional requirements that can occur, an incorrect assumption might be smaller in scale and might need to be corrected on the fly. For example, you might have assumed you have adequate network infrastructure to support the two new vSphere servers based on information provided to you.

In the previous case study, when gigabit networking became available during the project, you ended up in a positive situation. In this situation, you are now onsite and ready to deploy the solution, but you now have a change in the technological infrastructure that will be supporting the solution and you must properly document the situation and move forward as appropriate.

In an ideal world, you should have done a site survey and confirmed specific switches, blades, and ports the vSphere hosts would be plugged in to. Due to certain circumstances, this might not always be possible and situations like this will occur as a result.

It is highly recommended to complete a site survey beforehand. When possible, multiple site surveys are ideal. Two different problems tend to spawn from doing site surveys too early or too late. From our experience, not doing a site survey well in advance fails to expose critical assumptions that cause a failure in the implementation. Nothing is worse than being expected to implement on a given day and having to hold off because the required pieces are not in place. On the other hand, leaving too much time in between can pose issues as well. For example, at times network connectivity can be scarce in certain locations and unused ports have a way of finding themselves used over time, even when reserved for other usage.

Automating Implementation Tasks

Automation allows you to accomplish implementations faster and more accurately. It might not eliminate, but will greatly reduce the amount of repetitive tasks that are part of a vSphere deployment. This might be reason enough to automate certain portions of your implementation. The greatest benefit, however, is in the standardization that automation can provide.

Several methods can accomplish this. The following sections discuss a few technologies that enable such automation. Each of these methods is briefly discussed and the following sections provide some useful community resources that take advantage of automation. The sections also talk about how these technologies can even be used with each other to deliver a robust and powerful solution for automating vSphere deployments.

PowerCLI

PowerCLI is a Windows PowerShell snap-in that provides a command-line method to automate many aspects of a vSphere deployment. It is easy to learn the basics and execute your first PowerCLI script. Beyond that, it is also a useful tool for querying and generating reports about an existing vSphere infrastructure. Although Microsoft really hit the nail on the head with the PowerShell language, the real power of the technology for vSphere environments is in the many scripts and learning resources that have been shared among the VMware community.

Throughout this book, several PowerShell scripts are provided that will aid in implementing, managing, and operating your environment. Learning PowerShell is beyond the scope of this book, but if you are looking to become familiar with PowerCLI, you can refer to Appendix A at the back of this book, which provides a list of excellent resources on this topic.

Host Profiles

Host Profiles are a feature of vCenter that allows profiles to be created and applied to vSphere hosts. Host Profiles ease configuration management of your vSphere hosts and provide the ability to monitor and check against configuration drift. This feature is part of the Enterprise Plus Edition only.

During an implementation, Host Profiles are great for automating the application of configuration to your hosts. After that, we rarely see them used for continuous verification and alerting of noncompliance. This isn't due to a lack of capability but rather a lack of knowledge of the product. Many people don't realize that a task can be scheduled to check compliance of a profile individual times or on a recurring basis. Additionally, alerts can be generated to confirm the host is compliant, noncompliant, or that the compliance check has occurred.

By default, profiles are checked once a day via a scheduled task that is created for each profile when it is created. Like alerts , there isn't any email notification by default and this must be configured. You may configure email notification to confirm the task itself has completed by editing the scheduled task. You probably already get enough email every day though and want to know specifically when things are problematic. You can get this level of granularity through the use of alarms that are configured for email notification.

You may configure notification for Host Profile application, compliance, and noncompliance at the vCenter, datacenter, cluster, or host level. First, select the appropriate level of hierarchy you want to configure. Next, create a new alarm and go to

> Alarm Settings, Alarm Type: Hosts, Monitor for Specific Events, Triggers: Host Profile Applied, Host Compliant with Profile, Host Noncompliant with Profile

You may configure notification to ensure cluster compliance is being checked as well. This can be done at the vCenter, datacenter, or cluster level. First, select the appropriate level of hierarchy you want to configure. Next, create a new alarm and go to

> Alarm Settings, Alarm Type: Clusters, Monitor for Specific Events, Triggers: Check Cluster for Compliance

DEPLOYMENT TIP

When you first install vSphere, you have a 60-day grace period before applying your license. If you aren't using the Enterprise Plus Edition, this is a great time to try some of the features, such as Auto Deploy and the distributed virtual switch. In the case of Host Profiles, this is a great time to not only try the feature, but also to save some time during the implementation. When you are finished configuring the first host, simply create a Host Profile and then attach it to the rest of your cluster. After it has applied, it will be configured just like the other hosts.

Auto Deploy Server

vSphere 5 has added the capability to automatically provision hosts through the use of an Auto Deploy Server. Auto Deploy is a new feature in vSphere 5 that auto provisions hosts in the infrastructure. The Auto Deploy Server in many ways turns the vSphere host's hardware into a commodity. It does so by deploying a stateless image that runs from memory. You may choose to run Update Manager to do patches that don't require reboot only. Remember that these hosts are stateless and will load the unmodified boot image at load time, effectively wiping out any changes that were made that aren't part of a Host Profile. Instead, you would simply update the host image using Image Builder.

The Auto Deploy Server requires Enterprise Plus licensing; however, there is an offset in costs that will result from eliminating local storage from the servers.

Auto Deploy also requires some additional infrastructure components that might or might not already exist. As depicted in Figure 2.1, these include DHCP, PXE, TFTP, and PowerCLI. The process for setting up Auto Deploy is documented by VMware in the *vSphere 5 Evaluation Guide*, *Volume Four*, the link for which is provided in Appendix A. A common place of error is during the DHCP scope configuration.

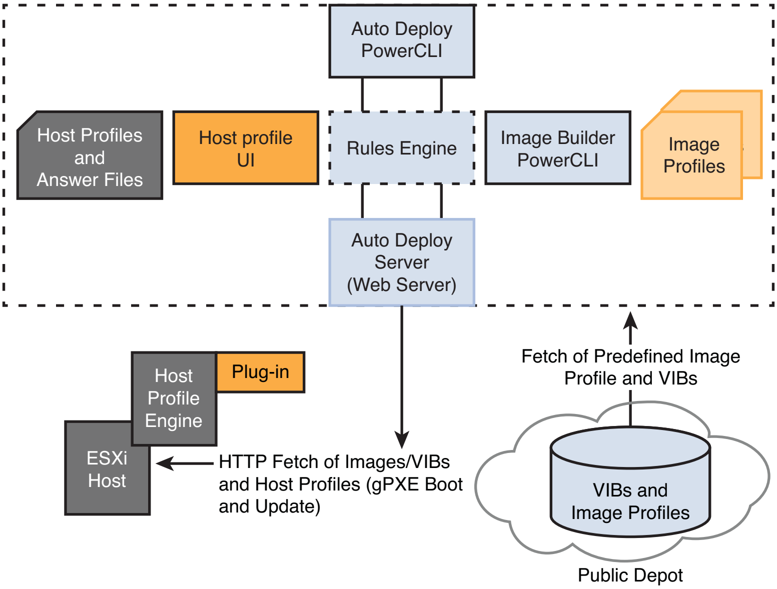
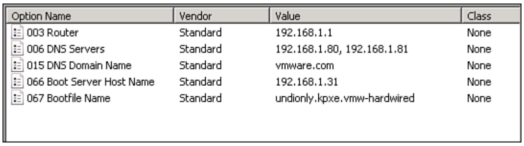

Auto Deploy PowerCLI
Host Profiles and Answer Files
Host profile UI
Rules Engine
Image Builder PowerCLI
Image Profiles
Auto Deploy Server (Web Server)
Host Profile Engine
Plug-in
ESXi Host
HTTP Fetch of Images/VIBs and Host Profiles (gPXE Boot and Update)
Fetch of Predefined Image Profile and VIBs
VIBs and Image Profiles
Public Depot

Figure 2.1 Auto Deploy Components

For a Windows DHCP scope, you must configure both options 66 and 67, as noted in Figure 2.2. Option 66 will specify the TFTP server while option 67 will be the bootfile name, which should be undionly.kpxe.vmw-hardwired.



Option Name	Vendor	Value	Class
003 Router	Standard	192.168.1.1	None
006 DNS Servers	Standard	192.168.1.80, 192.168.1.81	None
015 DNS Domain Name	Standard	vmware.com	None
066 Boot Server Host Name	Standard	192.168.1.31	None
067 Bootfile Name	Standard	undionly.kpxe.vmw-hardwired	None

Figure 2.2 Configuring Windows DHCP Scope for Auto Deploy

For a Cisco DHCP scope, there can be some difficulty in making the configurations, especially if you are less familiar with Cisco networking. Not to worry as there are only two lines you need to configure for the scope. *Next-server* refers to the TFTP server you have configured, while *bootfile* refers to the bootfile name, which should be undionly.kpxe.vmw-hardwired.

```
Router(config)# ip dhcp pool My-Pool
Router(dhcp-config)# next-server 192.168.1.44
Router(dhcp-config)# bootfile undionly.kpxe.vmw-hardwired
```

Auto Deploy was originally released by VMware Labs as a fling before being incorporated into the most recent release of vCenter. In that release, Auto Deploy was an appliance that incorporated a TFTP server. With the release of Auto Deploy for vCenter 5, the TFTP server is not included and you must provide your own.

Solarwinds has a great TFTP server that will fit your needs with one caveat. By default, like many free TFTP solutions, it will not start automatically with Windows. However, this is not a problem because some of these applications, including Solarwinds, enable you to set the service to start automatically as a Windows service.

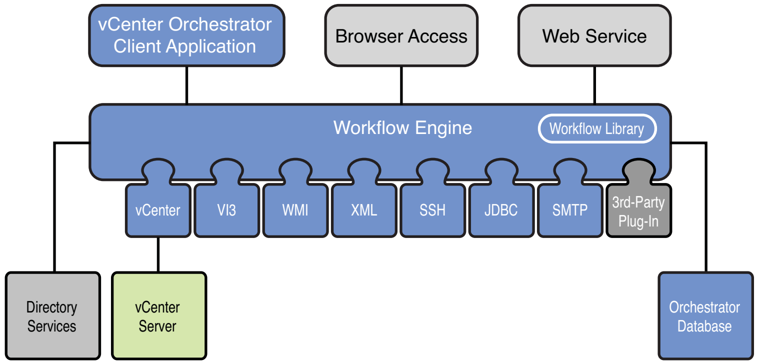
Auto Deploy has a good use case when it comes to rectifying hardware-based host failures in a vSphere cluster. This includes environments where host failover capacity is not purchased or the environments that want extra hardware ready and available in the case of several hardware failures.

Auto Deploy delivers pieces based on deployment rules that are defined by patterns. Simply defining patterns based on the host hardware type of some older, but still fully supported, hardware would allow another host to be fully set up with patches like the rest of your hosts. Additionally, once online, Host Profiles will finish the setup and standardize the host as you have chosen.

Of course, you also need to make sure the host has the same processor type/level as the existing cluster or enable the Enhanced vMotion capability. Sure, this capability to bring another host online exists without the use of Auto Deploy, but it is important to remember that not all infrastructures being built will have someone available to go through all the steps in the process to bring a host online and integrate it into the vSphere infrastructure. Even those that do might not have had any hands-on time with vSphere in some time. Being set up for Auto Deploy in this case means you've taken the time beforehand to ensure everything will be ready to go when it is needed most. Times of crisis are times when a lot is learned, but that doesn't mean they always have to be.

vCenter Orchestrator

vCenter Orchestrator is perhaps one of the least understood features of vCenter. We also have found it to be one of the least used features as a result. vCenter Orchestrator provides workflows that aid in automating actions. One of the core benefits of the product is it provides a library of workflows to choose from so that you are not reinventing the wheel for every automation need you might have. Products such as vCloud Director, for example, are heavily reliant on the workflows put in place for setting up a vCloud Director infrastructure. Additionally, custom workflows can be created from existing or new additions to the library. Figure 2.3 shows these components and note that it also shows vCenter Orchestrator's reliance on its own dedicated database server.



From Official VMware Visio Stencil Set

Figure 2.3 vCenter Orchestrator Components

One thing that comes as a shock to many is that vCenter Orchestrator is installed by default with vCenter and after a brief configuration is ready for your workflows. It is, however, only available with vCenter Standard Edition or above.

Detailed information on the use of vCenter Orchestrator is outside the scope of this book. Cody Bunch, however, has published a complete work on its use in his book *Automating vSphere with VMware vCenter Orchestrator*, published by VMware Press.

Verifying Implementation

The hardware has been deployed, vSphere has been installed, and you are ready to start eliminating all of those physical servers. Before going any further and placing production workloads in the infrastructure, the implementation should be verified. When talking about verifying the implementation, note that there are actually two different types of verification.

First, you need to verify the implementation for functionality. This means testing for a desired outcome. Here, you confirm items are functional under a number of scenarios outlined in your verification plan. Second, you need to verify configurations of your implementation. Again, you analyze the implementation, but this time you look to ensure your configurations match the intended design.

There are a number of scenarios in which a vSphere environment may function correctly in the current state yet not match the intended configurations. Note that when you test functionality, you are only testing against the present state. Configurations outlined in the design, however, may take into account anticipated changes to the infrastructure. Assessing both functionality and configuration helps mitigate these issues.

Testing Functionality

As previously mentioned, functionality testing will be performed to ensure features of the design function as intended. When discussing functionality, it means you are focusing on whether an item works. Furthermore, the feature needs to function at certain levels and meet set expectations of performance. A detailed list of the functionality to be tested should be included with the design documentation. Furthermore, an outlined plan of how to test each function should be produced when possible. This test plan should be updated with new releases of vSphere as subfeatures of items—like High Availability and Distributed Resource Scheduler—tend to change, and other new ones are introduced, such as Storage DRS with the release of vSphere 5.0. Whereas DRS balances virtual machines across hosts based on CPU and memory utilization of the hosts, Storage DRS balances virtual machines based not only on space usage, but also on I/O latency. Now let's talk about some of the specific functionality that should be tested during a typical vSphere implementation.

High Availability

VMware's High Availability feature, or HA, will restart virtual machines on another host in the event of a failure to a vSphere host. Additionally, a subfeature of HA known as *VM monitoring* or *VM HA* will restart a specific virtual machine if an operating system crashes. This section goes through High Availability at a high level; however, I highly recommend you check out Duncan Epping's *HA Deepdive* (see Appendix A) for a deeper understanding of configuration options.

For HA, then, you have two main features you need to verify, the ability to restart all VMs on a host and the ability to restart a specific VM that has failed.

Host Failure

To determine your functionality testing plan for HA Host failures, you need to make sure you understand the intended result of your configurations. HA is configured with a host isolation response that dictates the expected action for virtual machines if an HA failure has been detected. The isolation response will be one of the following. Note the description beside each of the available options.

- **Shut Down**—Safe shutdown of the guest after 5 minutes. This was the default in vSphere 4.1.
- **Power Off**—Hard shutdown of the guest immediately.
- **Leave Powered On**—Virtual machines remain powered on. This is the default in vSphere 5.

Now that you know the possible isolation responses, which one do you configure? In general, our recommendation is to use Leave Powered On for most scenarios. This option keeps virtual machines running if the host still has access to the storage. This is the best option for both IP-based and Fibre Channel–based storage where a host isolation wouldn't also cause an isolation from the storage. Furthermore, if the storage also becomes isolated, this option leads to a power off.

If it is possible hosts will still have access to storage during isolation and a restart is required, then choosing the Shut Down option is recommended. This provides a safe shutdown of the operating system and eliminates chances of corruption occurring to operating systems that are not cleanly dismounted.

If it is likely hosts will not have access to storage during an isolation event, then Shutdown is not going to be an option because the virtual machine cannot be accessed. This might be the case when using IP-based storage and when a network isolation would result in that connectivity being severed. When the requirement is to quickly restart virtual machines when an isolation event occurs, then the Power Off option is recommended.

Knowing the anticipated isolation response is only part of what you need to understand. You also need to know which virtual machines you are expecting to power back on and in what order. This is especially important in environments that are close to or 100% virtualized as application dependencies such as Active Directory, DNS, and SQL will prevent other machines from properly starting their own services that are dependent upon them.

Again, place close attention to noticing which virtual machines are expected to be powered back on. The design will dictate this because it might be expected that only a given set of virtual machines will be powered back on as a trade-off for reduced failover capacity. A failure to properly lay out the virtual machines that will power back on along with their order could lead to critical virtual machines being restarted later or not at all due to a violation of HA admission control settings. Consider designing a power down and restart plan as part of operational procedures. This process should include not only the order of restart of virtual machines but also application dependencies. As changes occur with applications and services over time, also consider HA settings and the effect of a host failure on applications.

Several methods exist to simulate a host failure for testing purposes:

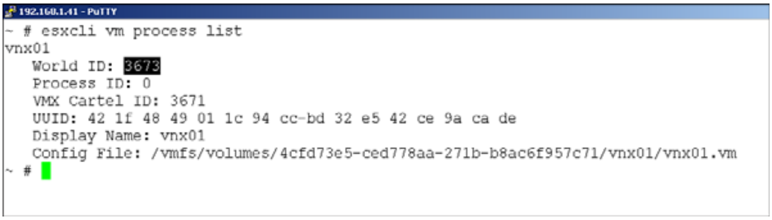
- Remove all networking.
- Remove all power.
- For blade servers, remove the blade.
- Force a kernel panic, the Purple Screen of Death (PSOD).

Another method that also accomplishes the simulation of a host failure for purposes of testing is forcing a kernel panic, which creates a dump file. To force a kernel panic:

1. Connect to the vSphere host via SSH or the ESXi console.
2. Type **vsish**.
3. Type **set /reliability/crashMe/Panic**.

A VM that has blue screened or is otherwise unresponsive can be restarted using VM HA. To test the functionality of a virtual machine, restarting will be necessary to have a way to force a failure. This can be accomplished through a number of utilities that trigger a blue screen. You can also initiate the failure of a specific virtual machine from a vSphere host by doing the following:

1. Connect to a host using SSH.
2. As shown in Figure 2.4, determine the World ID of the virtual machine(s) to be crashed by entering the command **esxcli vm process list** and locating the value for World ID.



```
~ # esxcli vm process list
vnx01
   World ID: 3673
   Process ID: 0
   VMX Cartel ID: 3671
   UUID: 42 1f 48 49 01 1c 94 cc-bd 32 e5 42 ce 9a ca de
   Display Name: vnx01
   Config File: /vmfs/volumes/4cfd73e5-ced778aa-271b-b8ac6f957c71/vnx01/vnx01.vm
~ #
```

Figure 2.4 Forcing a Virtual Machine Failure

3. Enter the command **/sbin/vmdumper** *wid* **nmi** where *wid* represents the World ID of the virtual machine to which you would like to send the NMI.

Distributed Resource Scheduler

VMware's Distributed Resource Scheduler (DRS) dynamically balances your resources among vSphere hosts. It accomplishes this via the use of resource pools and allows several levels of automation to take action based on recommendations provided. Recommendations that are provided range from priority 1 to 5, with 1 being the most conservative and 5 being the most aggressive. The automation levels available are manual, partially automated, and fully automated. As discussed earlier in Chapter 1, the recommended setting is fully automated unless a constraint exists that deems otherwise.

Several subfeatures of DRS exist. These include affinity and anti-affinity rules, DRS groups, Distributed Power Management (DPM), and automation levels for individual VMs.

DRS groups allow the creation of two different kinds of groups, host groups and virtual machine groups. When testing the functionality of DRS, the main objective is to ensure DRS is functioning by moving virtual machines during times of contention. This is accomplished when vMotion operations are initiated.

To accomplish this, you need to simulate a high load on a host, which can be easily accomplished by spiking the utilization of CPU on individual virtual machines. You could also seek to utilize memory on virtual machines; however, it will typically take many more virtual machines to cause contention with memory than it will to cause contention with the processor. Many tools exist to accomplish these actions, and two that we recommend are as follows:

- **CPUbusy**—There are many variations of this script. The software will spike the CPU on a virtual machine while running and cause near 100% utilization.
- **Heavyload**—This software performs CPU, memory, and disk load to aid in testing performance.

One thing that might come to light here as well is issues with virtual machines or hosts that are not configured correctly to enable vMotion. Running a separate test ahead of time to check for vMotion compatibility is a good idea and you learn more about this later in the chapter.

Networking

When testing networking functionality, you must verify functionality across a number of areas. With the use of IP-based storage, this means you need to verify storage functionality in addition to the networking components they consist of.

For a vSwitch, two types of network port groups can be created. A virtual machine port group can contain only virtual machines, and only virtual machines can be in a virtual machine port group. A VMkernel port group, on the other hand, can be used for one or several types of traffic:

- Management
- Virtual machine
- vMotion
- FT
- Storage networking (either iSCSI or NFS)

Each of these different types of networks requires unique plans for testing functionality because they differ in functionality themselves. Let's start our functionality verification at the management level.

Verifying Management Networking Functionality

Of all the traffic types that can be configured for a VMkernel port, management traffic is the only one that is required to be configured for at least one of the VMkernel ports on a host. A single VMkernel port might not be ideal and you might choose to configure a second VMkernel port that is configured with a different set of outbound NICs backed by separate network infrastructure from the first VMkernel port. You might also decide like many that a second VMkernel port entails a more complex configuration and implementation and that the risk of a VMkernel port failing is fairly low. In this case, a single VMkernel port for management backed by a set of multiple NICs that are part of a separate network infrastructure might meet your requirements.Regardless of how your management network is configured, you need to make sure it is functional at all times. The best way to do this is to verify redundancy, the process of which is described in the following steps:

1. Disconnect one of the two management networks. If your implementation consists of two VMkernel ports, remove the network connectivity to one of those ports. If your implementation consists of a single VMkernel port backed by multiple physical network adapters, remove network connectivity to one of those adapters.
2. Verify management connectivity still exists to the host. This can be done by pinging the host or connecting with the vSphere client.
3. Reconnect the management network that was disconnected.
4. Disconnect the other management network.
5. Verify management connectivity to the host still exists.
6. Reconnect the management network that was disconnected.

If this test fails, it is likely you have not configured the management port group correctly or one of the management networks is not configured properly end to end.

You need to ensure that you have configured the vmnics for the management port group in a manner that allows for proper failover. If you mistakenly configured a vmnic as an unused adapter, then this test will always fail. We recommend an active/active setup for the management port group. Earlier we discussed a recommendation of two NICs dedicated to management traffic; however, we understand this might not always be the case. You might have fewer physical NICs to use and as a result, you might then have shared NICs for management with vMotion traffic. In this case, you need to check and ensure the port group is configured in an active/standby fashion, with the active NIC for management being the standby NIC for vMotion and vice versa.

Verifying Virtual Machine Networking Functionality

Creating a virtual infrastructure is without purpose if virtual machines cannot run uninhibited, so checking the functionality of the virtual machine networking is critical to any project. When you look at virtual machine networking, you are going to be concerned with not only whether it works, but also whether it is performing to your expectations. Additionally, you need to verify redundancy and the ability to failback after a failover if required. Security is also important, and you must ensure you are isolating virtual machines as required by the requirement of the design.

Verifying Virtual Machine Networking Isolation

If any of the virtual machines are to be isolated from other virtual machines, it is necessary to verify these machines cannot talk to each other. For some environments, this might be few or no virtual machines that are isolated from others. In other environments, there might be many machines isolated from each other using Private Virtual Local Area

Networks (PVLAN) or vShield Zones. Verify network connectivity does not exist for any or all services that are supposed to be blocked. This can be done by attempting to communicate to other hosts or networks that are designed to be isolated.

Furthermore, any machines that need to talk to each other should have their communication with each other verified. This will be a majority of your virtual machines.

Verifying Virtual Machine Networking Redundancy and Failback

Verifying virtual machine networking redundancy is similar to the verification of management networking redundancy you ran through earlier. You should do the following for each virtual machine network that exists:

1. Define several virtual machines to test with. Issue a continuous ping to each system and verify connectivity currently exists.
2. Disconnect one of virtual machine network vmnics.
3. Verify connectivity still exists to the virtual machines.
4. Reconnect the vmnic that was disconnected.
5. Disconnect another vmnic.
6. Verify connectivity still exists to the virtual machines.
7. Reconnect the vmnic that was disconnected.
8. Repeat for each vmnic that backs the virtual machine network port group.

If this test fails, it is likely you have not configured the virtual machine port group correctly or one of the network links is not configured properly end to end. Additionally, you might not have properly defined the active and/or standby vmnics properly.

Verifying Virtual Machine Networking Performance

You will test virtual machine networking performance from the perspective of the guest operating system. To do this, you use a tool such as iPerf, which allows performing tests end to end from multiple servers or workstations.

Of course for this testing, the expected results depend on what type of network connectivity exists end to end. Even with 10-gigabit networking, if the source and destination have to be routed through a 1-gigabit router, you cannot expect anything higher than the smallest link in terms of throughput. On the flip side, two virtual machines that exist on the same host will have the greatest possible throughput with each other.

From a single virtual machine, check the throughput as follows:

- With another virtual machine on the same host
- With another virtual machine on a different host
- With another machine on a different subnet

From multiple virtual machines, check the throughput as follows:

- With other virtual machines on the same host
- With other virtual machines on different hosts
- With other virtual machines on different subnets

Checking individual virtual machine networking can be tedious and there is another way to ensure that networking is not being saturated from the vmnic level. From each host, you can use esxtop and look at the networking performance for each vmnic.

From the vSphere host, either logged in remotely or at the console, you can load esxtop by entering **esxtop** at the Command Line Interface (CLI). Additionally, you can use resxtop when using the remote CLI or the vSphere Management Appliance (vMA). Once loaded, you can enter the network view by typing **n**. You might notice a high number of packets of data being transmitted; however, the biggest indicator of network saturation tends to be dropped packets. Anything over zero is an indicator of issues for either dropped packets transmitted (%DRPTX) or dropped packets received (%DRPRX).

A full lesson on esxtop is beyond the scope of this book; however, refer to Appendix A for a great esxtop learning resource.

Verifying vMotion Networking Functionality

vMotion networking differs from the other types of networking required in that it has high requirements for throughput while having occasional or rare use in some environments. As a result, it is best to dedicate network adapters strictly for vMotion to not only give it the throughput it needs, but also to ensure it does not contend with other critical virtual machine or IP storage traffic.

Verifying vMotion Works

Several things can cause a vMotion attempt to fail even if the networking is correctly configured. You might have a CD-ROM drive attached to an ISO file that the destination does not have or a network configured or named differently on the destination host. It can be tedious to check these all out ahead of time manually, but you can use PowerShell and some of the many great resources already out there to ensure all of your virtual machines are capable of moving to any host.

Please refer to Appendix A for a scripted resource to test vMotion.

This is critical to perform now because a failure to verify this functionality could lead to issues down the line. For example, if you have virtual machines that can't vMotion, that means DRS will not be able to balance your workloads appropriately. Furthermore, if you are trying to do maintenance and a running virtual machine can't be moved off the host, then it is going to be impossible to perform maintenance on that host without virtual machine downtime or resolution of the configuration problems.

You might have a cluster with mixed CPU types and will not be able to vMotion machines across the hosts without further configuration. In this case, configure Enhance vMotion Capability (EVC) to mask the host CPUs to allow for vMotion compatibility.

Verifying vMotion Network Performance

After you have verified you can vMotion any virtual machine from and to any host in the cluster, you need to make sure that vMotion operations perform to a level of performance that is deemed acceptable for your environment. Again like the testing of storage performance, this depends on your configuration, although in this case variables are much more confined. You may be using only 2-gigabit NICs without Multi-NIC vMotion or you may be using up to four 10-gigabit NICs with Multi-NIC vMotion enabled. As such, the results will vary greatly; however, following this plan allows for proper verification of vMotion network performance:

- Define a baseline. Know what to expect when just one virtual machine vMotion is performed at a time.
- Perform an individual vMotion to and from each host in your cluster.
- Perform simultaneous vMotion operations to and from various hosts in your cluster.
- Disconnect one or several NIC cables to evaluate performance during a failure.
- Use esxtop to monitor network traffic and ensure contention does not exist.

As discussed earlier, vMotion traffic is now capable of saturating a 10-gigabit link. It is likely that if you are deploying 10-gigabit CNAs, you don't have more than four of them in each server. Consider the use of Network I/O Control (NIOC) to help eliminate contention when using 10-gigabit links that are shared by traffic types other than vMotion.

When vMotion operations fail, these are the most common reasons we typically see:

- The VMkernel port did not have the vMotion check box checked.
- The CD-ROM had ISO on local storage that was not visible to the destination host.
- The virtual machine itself was located on local storage.
- CPU affinity was set on the virtual machine.
- There are inconsistent settings on the vSwitch or port group between source/destination hosts.
- There is an active connection to an internal-only vSwitch.

When vMotion operations fail to perform as expected, the most common reason is a lack of dedicated networking for vMotion. When it comes to security, we also often see the vMotion network being routed. We recommend keeping this network physically isolated or through the use of nonrouted Virtual Local Area Networks (VLAN). Also, it is important to note that vMotion operations may be taking longer than normal if you are storing VM swap files locally or on a datastore not available to all hosts. This causes vMotion operations to take longer as the swap file needs to be copied during the vMotion process.

Verifying Fault Tolerance Networking Functionality

Fault tolerance provides continuous protection for virtual machines in the event of a vSphere host failure. Whereas High Availability will restart virtual machines on another host in the event of a failure, fault tolerance ensures the virtual machine will be continuously available via a secondary copy of the virtual machine that is kept in sync with the primary.

Every operation that is performed on the primary virtual machine is repeated on the secondary virtual machine. So when you go into the machine and remove all the boot files for the operating system and reboot, you will have twice the amount of boot failures. With that said, it is not a means of providing backups and not a replacement for clustering where a single virtual machine failure cannot be tolerated.

Fault tolerance is an item that is easy to configure but requires careful thought in its design as it has implications on the design as noted:

- Requires high bandwidth network for Fault Tolerance Logging
- One vCPU maximum
- Memory reservation equal to configured memory

- DRS disabled for the protected virtual machine
- Protected machine's virtual disks must be eager zeroed thick

NOTE

A thick disk can be created as either lazy zeroed thick or eager zeroed thick. A disk that is created as lazy zeroed has all its space allocated at creation time but the blocks are not zeroed out. Eager zeroed thick allocates and zeroes out all the blocks at creation time. This leads to longer virtual machine creation times as result.

- Hosts must be processor capable and supported for FT
- Requires Enterprise or Enterprise Plus licensing

NOTE

Refer to VMware Knowledge Base (KB) article 1013428 for a complete list of considerations and requirements when using fault tolerance.

When verifying the functionality of fault tolerance, you should verify several items:

- Protecting a virtual machine
- Simulating a host failure to ensure continuous protection
- Reconfiguring protection to another host

Protecting a Virtual Machine

A failure in protecting a virtual machine can indicate a few different things. For starters, it might be as simple as configuring the Fault Tolerance Logging check box for the VMkernel port. Without a Fault Tolerance Logging Network turned on, fault tolerance for a virtual machine will fail. It might be more complex than that. Remember that fault tolerance has several limitations that exclude its use in several situations. If a virtual machine has more than one CPU, for example, the configuration will fail.

Testing the protection of a virtual machine should be a fairly easy process, but if it fails, it can be a fairly complex resolution because it will likely be the result of a miscommunication or misunderstanding during the design process. It may be the VM has a need for more than one processor. In this case, perhaps the machine might not need both CPUs configured and everything will be fine. I've seen instances, though, where fault tolerance could not be used because of an oversight during the planning of the project. In several of these cases, fault tolerance was one of the bigger, if not the biggest, functional requirements to be met. Here is one of those cases.

Case Study

You are done with the implementation and are now getting ready to verify functionality and configuration of the infrastructure over the next several days. You have verified whether High Availability, Distributed Resource Scheduling, and management networking redundancy are all properly functioning. You are ready to verify fault tolerance functionality at this point as a formality. You know it is going to work because you've configured all the hosts with dedicated Fault Tolerance Logging Networks. All the virtual machines are single-processor machines. You've even configured the disks to be eager zeroed thick virtual disks just to speed up the process.

You turn on fault tolerance on the virtual machine and to your surprise you see an error. The error states, "The Virtual Machine is running in a monitor mode that is incompatible with Fault Tolerance."

Something is not right, and after verifying your settings and trying again on several different virtual machines, it is still failing. You begin to dig deeper and notice the vSphere hosts have a lower-level CPU than was ordered and this CPU is not supported for fault tolerance. You call up the vendor to tell them the wrong CPU was sent for the hosts and the vendor tells you that was the configuration that was ordered. This can't be right—the plan was to order a configuration that included a fault-tolerant–capable CPU. After digging deeper, you find that the decision was made afterward by management to order the same model server with slightly lower specifications for CPU because of cost savings.

The packing list was checked but who would have thought that the wrong CPU could have been shipped, let alone the wrong one ordered? In the end, the project took more than a month longer to complete because of the delay. Companies with tighter budgets might not have had the flexibility to bring the servers up to spec. In this case, if fault tolerance was a design requirement, it was one that was not going to be met by the current design.

Simulating a Host Failure

If you are fortunate enough to have successfully turned on fault tolerance for one or several virtual machines, it is time to simulate a host failure. Simulating a host failure will be done to ensure the operating system and application-level continuity exists during and after the failure.

Several ways exist to force a host failure:

- Remove all networking.
- Remove all power.
- For blade servers, remove the blade.
- Force a kernel panic, the PSOD.

Verification of this test requires monitoring the operating system and application as the failover occurs as well as afterward. Simply monitoring a ping to the host and accessing the application's Web browser interface might not be enough for some environments, so planning ahead and involving the proper stakeholders is vital to testing functionality for fault tolerance and any other area for that matter.

Reprotecting a Virtual Machine

When finished, you need to reprotect the virtual machine. The process is similar to protecting the virtual machine, only this time the secondary virtual machine should be placed on a different vSphere host. This ensures the entire cluster is correctly configured and capable of supporting virtual machines configured for fault tolerance.

Verifying IP Storage Networking Functionality

Today, most virtual infrastructures deployed have some form of IP storage, whether it is NFS or iSCSI. This storage may be the primary storage for the infrastructure or it may be implemented only to store test/development or templates. Regardless, there is a need to verify whether the storage networking capabilities are functional.

This testing ends up encompassing a lot of layers, and troubleshooting issues can take the work of individuals on several teams. For example, consider the following:

- The virtual machines are running on hardware that is using gigabit or higher physical NICs to connect to physical network switches that are in some way connected to a storage device.
- This storage device in turn will have a certain storage configuration.
- It may have few or many gigabit or higher physical NICs.
- It may have disks of various speeds that are configured as various RAID types with varying amounts of LUNs on such RAID groups.
- The LUNs themselves may have varying amounts of virtual machines running on them all with distinct IOPS requirements.

In fact, it is even more complicated than mentioned. The storage will also have some varying type of cache. We also must consider the layer between the virtual machine and the physical NICs on the vSphere host. The port group in which the VMkernel port exists will have its own set of configurations that will contribute to the performance and availability of the storage to virtual machines. This certainly isn't the place to skimp on costs by reusing old Cat 5 cabling.

In terms of testing functionality, we will be looking at testing two key areas:

- Storage networking availability
- Storage networking performance

Storage Networking Availability

Similar to your testing of management networking availability earlier, you need to ensure storage networking is highly available in the event of a failover as well. You accomplish this similarly by providing redundancy in networking from the VMkernel layer all the way through to the physical network card (vmnic) and physical switch layer. Additionally, though, you might also need to configure storage failover policies.

iSCSI networking allows for the configuration of multiple VMkernel ports, and as such the storage will have configurable options for multipathing. NFS, on the other hand, is a session-based protocol that allows only a single network path from one vSphere host to one NFS server. This means that for NFS, there is no way to create multiple paths. Instead, you should create multiple VMkernel ports and multiple NFS servers on separate subnets for each NFS datastore or sets of NFS datastores.

> **NOTE**
>
> It is important to note that storage networking should be on physically separate and redundant switches. Separating out ports on an existing switch is not recommended because the resources are shared, potentially causing contention for bandwidth. Additionally, this makes it more likely that a maintenance activity to the network impacts the storage and ultimately the virtualization infrastructure.

Storage Failover NFS

Our test plan for NFS storage failover is going to be similar to that of management networking failover in that our main objective will be to verify the redundancy as there will not be storage path failover like that of iSCSI. The failover for NFS like that of management networking will be via redundant active or standby network connections from port groups that are backed by multiple vmnics. Ideally, these vmnics should be distributed to separate physical network switches. Even though NFS will only ever use one active connection, the standby network connections and secondary switches are critical if a vmnic or physical switch fails. To verify redundancy, follow this procedure:

1. Disconnect one of the vmnics that backs the VMkernel port for the NFS network being tested.
2. Verify whether NFS connectivity to the host still exists. This can be done on the host using a **vmkping** command as documented in VMware KB article 1003728.
3. Reconnect the NFS network that was disconnected.
4. Disconnect another of the vmnics that backs the VMkernel port for the NFS network being tested.
5. Verify whether NFS connectivity still exists to the host.
6. Reconnect the NFS network that was disconnected.

NOTE

To sidestep for a minute, let's talk about the **vmkping** command and why we don't issue just the **ping** command. The **vmkping** command will source the request using the server's VMkernel port. If we used the **ping** command, we would be testing connectivity from the server's management network to the NFS server's IP address. We do not recommend it, but the NFS network may be routed. If this were the case, then this would result in a successful ping test, even though we haven't actually checked whether the server can truly access the NFS server via its VMkernel port.

Storage Failover iSCSI

The use of iSCSI provides some unique options for multipathing and load balancing not available to NFS. For redundancy, we recommend configuring iSCSI port binding using multiple VMkernel ports backed by separate vmnics. When used with the Round Robin path selection policy, this provides the greatest benefits in terms of redundancy and load balancing. Additionally, it is highly recommended that separate physical network switches dedicated for iSCSI traffic be used.

When configuring iSCSI port binding, you must consider the following:

- You must configure multiple VMkernel ports.
- Each VMkernel port should be backed by a unique active NIC and only one active NIC.

- Vendor requirements may dictate your configuration. For example, port binding with the EMC Clariion requires VMkernel ports in different subnets.

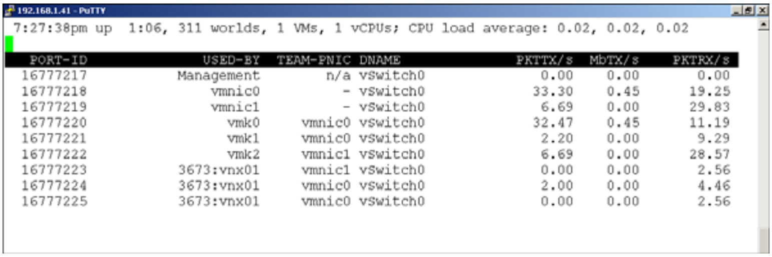
Although this used to be a process that could be done only via the command line, vSphere 5 has now introduced the ability to bind VMkernel ports through the graphical user interface (GUI). For more detailed information on configuring iSCSI port binding, refer to the *vSphere Storage Guide* (see Appendix A).

To verify redundancy of iSCSI port binding, you may do the following:

1. Disconnect one of the vmnics that backs one of the VMkernel ports for the iSCSI network being tested.
2. Verify whether connectivity to the host still exists.
3. Reconnect the network that was disconnected.
4. Disconnect another of the vmnics that backs the VMkernel port for the iSCSI network being tested.
5. Verify whether connectivity to the host still exists.
6. Reconnect the network that was disconnected.

To verify iSCSI multipathing is functioning, complete the following steps:

1. Launch esxtop.
2. Press the **n** key to enter the networking view.
3. Perform a storage vMotion involving a datastore on your iSCSI network.
4. Monitor the Packet Transmits (PKTTX/s) for activity. You will be looking for activity on all the VMkernel ports that were bound for the initiator, as highlighted in Figure 2.5.



```
 7:27:38pm up  1:06, 311 worlds, 1 VMs, 1 vCPUs; CPU load average: 0.02, 0.02, 0.02

   PORT-ID              USED-BY  TEAM-PNIC DNAME              PKTTX/s  MbTX/s    PKTRX/s
  16777217           Management        n/a vSwitch0              0.00    0.00       0.00
  16777218              vmnic0           - vSwitch0             33.30    0.45      19.25
  16777219              vmnic1           - vSwitch0              6.69    0.00      29.83
  16777220                vmk0      vmnic0 vSwitch0             32.47    0.45      11.19
  16777221                vmk1      vmnic0 vSwitch0              2.20    0.00       9.29
  16777222                vmk2      vmnic1 vSwitch0              6.69    0.00      28.57
  16777223           3673:vnx01     vmnic1 vSwitch0              0.00    0.00       2.56
  16777224           3673:vnx01     vmnic0 vSwitch0              2.00    0.00       4.46
  16777225           3673:vnx01     vmnic0 vSwitch0              0.00    0.00       2.56
```

Figure 2.5 Verifying iSCSI Multipathing

On the vSphere side, you have now verified the network failover redundancy, but let's take a moment to talk about some things you haven't verified by doing so. You haven't verified whether the network switches themselves are ready for a failover event. You also have not verified whether there is true redundancy on the storage side. You might know that four physical network connections go into the storage split among multiple switches, but at this point, you have not verified the storage will function in the event of failure to one or more of those network connections. Networking connectivity to storage is often itself virtualized, so there is a lot to consider when testing and troubleshooting performance and connectivity issues with IP-based storage.

Fully testing the redundancy of your storage or network will vary based on your configuration and vendor; however, we suggest at a minimum you power off redundant physical switches and fail back and forth among them. Additionally, we suggest removing storage controller cabling and inducing failover and failback of LUNs presented to storage processors.

Storage Redundancy and Failback

We previously discussed IP storage functionality testing, and it is now time to focus on Fibre Channel networking functionality specifically. You may have a 2-, 4-, or 8Gbps Fibre Channel infrastructure and you will have a varying amount of storage processors, disks, raid types, LUNs, and virtual machines running on those storage devices. You can reference the previous section on IP storage performance testing, which is the same in terms of the execution; however, the expectations vary, so keep the considerations of your configuration in mind when you lay out your performance testing.

When looking at storage redundancy, your testing will mimic that of the iSCSI redundancy. Similar to iSCSI redundancy testing, you also need to test the redundancy of the storage and networking itself by failing over LUNs to the other storage processor where applicable. Although iSCSI is dependent upon network switching and the proper setup of traditional IP networking, Fibre Channel storage is dependent upon proper zoning of hosts and masking of LUNs to work effectively. This must be done correctly for all storage processors for failover to work correctly. Your environment will ideally have hosts with multiple HBAs connected to two separate fabrics, which in turn are connected to storage with multiple storage processors. This might not always be possible, so your testing can vary. To verify Fibre Channel storage redundancy from vSphere, follow these steps:

1. Disconnect one of the HBAs.
2. Verify whether connectivity to the host still exists.
3. Reconnect the HBA that was disconnected.

4. Disconnect another HBA.
5. Verify whether connectivity to the host still exists.
6. Reconnect the HBA that was disconnected.

You also need to test the failover by performing the following:

1. From your storage device, failover LUNs to another storage processor.
2. Disconnect paths from your storage device.
3. Disconnect one of the fabrics, where multiple fabrics exist.
4. Disconnect paths from your vSphere hosts.

You might also want to remove power to a storage processor or from redundant switching to test what will occur during a failure of either of these components.

PowerPath/VE

This is a good time to briefly mention another option for managing paths that will provide for the greatest benefits in terms of throughput and failover. PowerPath/VE provides dynamic load balancing and failover for virtual environments. It also works to continually ensure paths are optimized. An in-depth discussion of PowerPath is beyond the scope of this book; however, you can find more information on PowerPath and the best practices for using it in your vSphere environment in Appendix A.

Storage Networking Performance

Developing a test plan for storage performance involves a firm understanding of the individual infrastructure that has been rolled out. We expect very different results for the performance of NFS storage running over gigabit connectivity that is backed by 10K SATA drives than NFS storage running over 10-gigabit connectivity that is backed by 15K SATA drives. Regardless, a standard methodology can be followed that allows the testing of the storage that you have deployed. You just need to make sure you have a baseline developed beforehand with your environment's expectations. In addition, you should be seeking to define a baseline going forward for storage performance expectations.

You should look at a few different areas of storage performance. For starters, you need to look at the overall performance of the storage for virtual machine operation. Second, you need to look at performance during actions such as rebalancing of datastores using storage vMotion.

When looking at storage performance for virtual machine operation, you can accomplish this at one of three layers:

- At the Virtual Machine layer
- At the Host layer
- At the Storage layer

You can measure performance by looking at the performance statistics of each individual virtual machine. You can do this by looking at each virtual machine and gauging performance by looking at the performance of guest operating systems. You can use tools such as Perfmon in Windows or top in Linux to see what the performance is from the perspective of the guest. You also can use tools like IOMeter to stress test the storage and see what your disk throughput is.

You also can measure performance at the Host layer. This can be done by using esxtop to check disk metrics for each host. If you are not familiar with esxtop, check out Appendix A for further resources. You can additionally check out VMware KB article 1008205, which describes the use of esxtop for troubleshooting performance issues (also see Appendix A for a link to said article).

Finally, you can measure performance at the Storage layer. This varies from vendor to vendor, but your storage vendor should have several interfaces and tools that allow historical and real-time monitoring of the storage performance.

In addition to looking at storage networking performance for normal operations, you need to test during one and several Storage vMotion operations. Again, the testing can take place in one or all of the layers. When testing, you are looking to define a baseline for future performance expectations.

Quality Assurance Assessment

Quality assurance can mean different things to different people, so let's start by defining what *we* mean by quality assurance verification for vSphere environments.

During the quality assurance assessment, you are looking to verify your implementation matches its intended configuration. This configuration is detailed in the design documentation but may have been overlooked or incorrectly input during the implementation. If there have been no changes in configuration due to a change in requirements, the configuration must match exactly. Checking functionality does not reveal all errors in configuration. Items may be fully functional, but when not configured correctly, could cause issues later on or provide less-than-optimal performance.

Automation should be used where possible to avoid inconsistencies and human error. Whether automation has been used or not, configuration errors are possible. Similar to the test for functionality, a quality assurance checklist should be created and maintained for current versions of vSphere.

The following sections describe some of the configuration items you should verify and some tools for helping the process.

VMware vSphere Health Check Delivery

The VMware vSphere health check is a paid engagement delivered by either VMware or an authorized solution provider of VMware. During this engagement, data is collected using the Health Check tool and analyzed to provide a report. This report is delivered to the customer along with guidance on resolving issues and concerns that have been found. When it comes to verifying configuration and compliance with generally accepted best practices and norms, the money spent toward a Health Check is well worth it. Areas are found where the configuration is not optimal, which assists in finding areas for remediation. Items are also uncovered that might not be configured as you originally intended or might be inconsistent between hosts in a cluster. Health Checks are typically done for an existing environment that has been running for some time, but immediately after an implementation is a great time for them as well.

As a VMware authorized solution provider, we can attest firsthand to the benefits of the Health Check delivery.

You are guaranteed to have someone who is a VMware Certified Professional (VCP) or above.

Individuals performing and delivering the Health Check engagements have performed these engagements across many different organizations and, as a result, have a level of experience that has exposed them to the common misconfigurations found in vSphere deployments.

We have performed many health checks, and there are clear trends in some of the items that are commonly misconfigured. Throughout this book, we provide you with many of these common pitfalls along with guidance on remediating your environment.

VMware vSphere Health Check Script

If you prefer to run a health check yourself, a great community resource is available that allows for a daily email to be sent with a nicely configured report. This provides for a nice and simple view of your environment and shows off not just configuration items, but performance information as well. More resources on this script and implementing it can be found in Appendix A.

Verifying Configurations

Many configurations could be made into an infrastructure, and it would be impossible to discuss them all. There are, however, several common misconfigurations we encounter that occur either during the implementation phase or are changed afterward. In some cases, these may be items that are the defaults and should be changed and in others are simply misconfigured.

vCenter Specific Configurations

The following settings are focused on configuration items on the vCenter server:

- **HA Admission Control Policy not configured as intended**—If a mistake is made in configuring the HA Admission Control Policy, an issue almost never occurs immediately. However, two things can happen later. First, the cluster might not allow the powering on of additional virtual machines to a quantity that was previously expected. Second, in situations with mixed configuration hosts, a certain host failure could lead to a situation where not all the virtual machines can continue to run safely or in an optimal fashion.
- **DRS not configured as intended**—A failure to enable DRS or configure it correctly is common. Not enabling leads to no rebalancing of your hosts during resource contention. Misconfiguring may lead to either too many migrations or very few migrations occurring.
- **vCenter server configured below minimum specifications**—This is more than just how you have configured the vCenter server but also what other pieces you are choosing to install on that server. In some smaller environments, SQL may be installed on the same box along with Update Manager. If you accidentally configured the vCenter server with only one CPU and only 2GB of RAM, you will soon realize the need to increase these configurations.
- **Database server not configured correctly**—The database server configuration is an area that is often overlooked. I rarely run in to a VMware administrator who was or is a database administrator, so this should come as no surprise. It is not uncommon to see any of the following issues with the database server:
  - Disks backing the server are not sufficient for the database.
  - The database is on the same volume as the operating system, Update Manager database, or other software that can lead to contention and volumes filling up.
  - Database cleanup tasks are not configured correctly and regular maintenance fails to occur. This leads to poor performing databases and the potential to fill up volumes and halt the database.

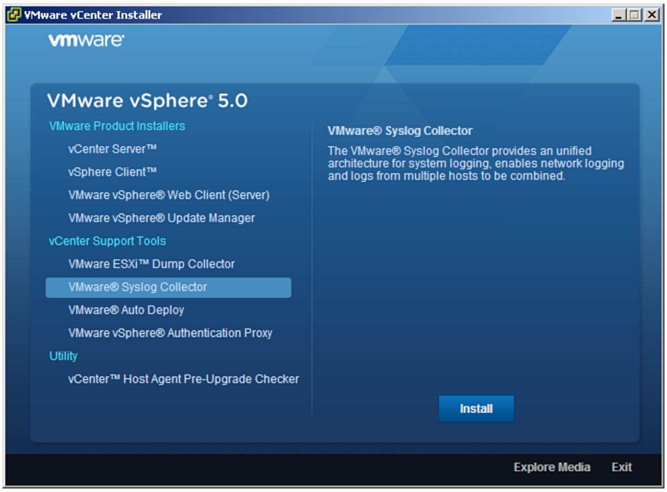
- **DNS servers not specified correctly**—If you input the DNS servers wrong, you might not immediately notice any issues. However, once you try to use Update Manager, you will soon realize the DNS servers are not correctly configured.
- **ESXi added to vCenter by IP address**—We've seen many times where hosts are added to vCenter by IP address. Hosts should be added to DNS and then added by hostname to avoid any potential issues with High Availability or Update Manager.
- **Incorrectly configured resource reservations and limits**—This can be resource pools, specific reservations, or limits. It can be in the form of CPU and memory, or network and storage. Regardless, a failure to properly execute the configuration of any resource reservations or limits can lead to drastic resource issues in your infrastructure.
- **Not configuring syslog**—We often find that a syslog server is either incorrectly configured or not configured at all. In an environment without a syslog server, there are other alternatives. With ESXi being stateless and not retaining the logs upon a reboot, it is critical to ship the logs somewhere in the event of an issue. It is not often that I hear of a vSphere host crash; however, it is not uncommon to speak to someone who is having issues and rebooted his or her host as part of troubleshooting. Important historical information about the issue will not be present if the logs are not shipped somewhere else.

  Our recommendation is to install a syslog server. They are very handy for not just your virtual infrastructure but for all of your networking devices as well. Furthermore, vCenter now ships with a syslog server that can be installed from the vCenter media, as shown in Figure 2.6. Additionally, the vSphere Management Assistant (vMA) also allows for syslog collection and has for some time now.

  If this is not feasible, you can also change the log directory and redirect the logs to another datastore. If you point this to a centralized datastore location, you will have all of your log files in one location. Alternatively, if you want to use some of that unused local datastore space to store your logs, they will persist even after a reboot. To do this, you need to go to each host and do the following:

  1. On the Configuration tab, choose Advanced Settings under the Software section.
  2. Browse to Syslog in the left pane.
  3. Change Syslog.global.logdir to a path on the host's local datastore.
  4. You must properly format the datastore name by typing it between the two brackets:

     ```
     [vmfs01-raid1-data] /logs/vspherehost01
     ```


VMware vCenter Installer
vmware
VMware vSphere 5.0
VMware Product Installers
vCenter Server™
vSphere Client™
VMware vSphere® Web Client (Server)
VMware vSphere® Update Manager
vCenter Support Tools
VMware ESXi™ Dump Collector
VMware® Syslog Collector
VMware® Auto Deploy
VMware vSphere® Authentication Proxy
Utility
vCenter™ Host Agent Pre-Upgrade Checker
Install
Explore Media
Exit

VMware® Syslog Collector

The VMware® Syslog Collector provides an unified architecture for system logging, enables network logging and logs from multiple hosts to be combined.

Figure 2.6 VMware Syslog Collector

Host-Specific Configurations

Now that we have discussed some vCenter configuration issues, let's move on to the hosts themselves. The following settings or configurations are focused on vSphere hosts themselves, including not only the vSphere software but also the underlying hardware and connectivity at the host level:

- **Host hardware not configured correctly or optimally**—For starters, it is recommended to install your vendor's specific customized ISO if available. A failure to do so can lead to issues such as the following:
  - Information missing under Hardware Status due to lack of CIM provider
  - Less-than-optimal hardware performance

  Additionally, these other items often are not correctly configured:
  - Up-to-date host BIOS and hardware firmware. Without disregarding VMware's Hardware Compatibility List, you need to ensure that the host BIOS and hardware firmware are up to date across not only the server hardware but also the networking and storage environment. This tool can be accessed online by going to http://vmware.com/go/hcl.
  - If your hardware is NUMA capable, you must disable node interleaving in the BIOS.

    - If your hardware supports hyperthreading, enable it.
    - PCI devices not placed consistently across hosts in a cluster.
- **vSphere not configured correctly on host**—These are several common configuration items we find that should be corrected:
    - Different versions of vSphere installed on hosts in the same cluster. This often does not cause any issues; however, it is not recommended.
    - Failure to stop technical support mode or SSH after usage.
    - Failure to properly configure Network Time Protocol (NTP). This is often set up; however, many forget to make sure it is set to start up automatically.
    - Inconsistency in configurations host-to-host in the cluster. This can lead to confusion when administering as well as issues with vMotion and other critical features of vSphere.
    - Inconsistent Path Selection Plugin (PSP) and primary paths between hosts in a cluster.

Virtual Machine–Specific Configurations

The following settings or configurations are focused on virtual machine–specific configuration items:

- VMs have ISOs attached on nonshared storage or they are located on nonshared storage.
- VMs have snapshots taken during initial rollout that have not been removed.
- VMs that have been migrated from a physical server using P2V converter or another tool have hardware that is no longer needed, such as communication ports or floppy drives.
- Although not specifically a configuration item, it is important to remember to make sure snapshots are removed after an initial rollout if they were created. A failure to do so could lead to issues later on. PowerShell is a great tool to check for the existence of snapshots in your cluster.
- Virtual machine has leftover software for hardware-specific functionality after P2V conversion. A server that used to be physical will have lots of software that is no longer needed. This eats up resources, so removing this software is highly recommended.

- Virtual machine was not properly resized after P2V conversion. Rarely, a virtual machine should be configured exactly identical to its existing configuration. When this occurs, it is usually to ensure there are no issues as a result of changes to these items on top of migrating to a virtual server. During the migration process, you have the option to change not only CPU or memory, but also contract or expand drives in the process. For CPU and memory, start small and scale up unless the operating system or application specifically requires it. For disk space, ensure you have enough for the present plus room to grow, but don't carry over a 1TB operating system drive just because the physical server had it.

Storage Configurations

Proper storage configuration is going to be heavily reliant on the best practices and configurations recommended by your storage provider. There are, however, many areas in which configuration errors commonly occur, outside of vendor-specific configurations:

- **Storage firmware and updates**—Your storage vendor will have bugs and fixes just like any other hardware and operating system. As a result, you need to pay close attention when implementing that the storage is at a firmware level that is currently supported. Additionally, you need to make sure your host's HBAs are also up to date and in line with your storage vendor's recommendations.
- **Failure to properly configure multipathing**—A failure to properly configure multipathing leads to unexpected results if a storage path goes down. If you are also deploying load balancing, your storage might not be sending traffic down multiple paths at the same time.
- **Storage vendor recommendations**—Pay close attention to any recommendations made by your storage vendor. For example, you need to make any recommended changes to the queue depth settings on the HBAs. Additionally, depending on the type of storage, you need to ensure the recommended path selection policy is selected.
- **Failure to properly size datastores**—If you create many smaller datastores, you will likely have much more wasted space than if you had several larger datastores. However, the larger the datastore, the more virtual machines that will be running and the greater the chance that you will experience contention for disk resources. The primary performance issue we see in virtualization environments is storage performance as a result of the lack of disk spindle count.
- **Failure to redirect swap files**—If you are using VMware's Site Recovery Manager (SRM), the solution might have been designed around storing the swap files on separate datastores that would not be replicated. Failing to properly implement this

leads to a large amount of data that will be replicated. If the links between the two sites are not sufficient, this can cause some issues and a failure to meet recovery point objectives.

Network Configurations

Proper network configuration ensures your infrastructure's networking performs well during normal operation and continues to function properly during a failure. The following are key areas we often find incorrectly configured:

- **Failure to configure portfast**—A failure to configure portfast on the switch can lead to issues if a spanning tree loop is detected during convergence. To configure portfast on a Cisco switch, you must enter Configuration mode and enter the command **spanning-tree portfast** for an access port or **spanning-tree portfast trunk** for a trunk port. For Nexus switches, enter **spanning-tree portfast type edge trunk**. For more information on portfast, refer to the VMware KB article 1003804 (a link is provided in Appendix A).

- **Failure to adjust default security settings**—By default, the security for a vSwitch is not set to what is recommended. Both options for MAC address changes and forged transmits are set to Accept. For security reasons, unless these are needed, they should be set to Reject for all vSwitches.

- **Failure to configure auto-negotiation for vmnics**—Although everything may work fine with networking configured as Gb/full or 10Gb/full, there are known issues by doing so:

  - In certain cases, performance is degraded when auto-negotiation does not take place. Some non-Cisco devices support the use of half-duplex Gigabit.

  - If flow control is desired, you cannot statically configure your network adapters; they must be negotiated.

  For more information on why auto-negotiation is recommended, refer to VMware KB article 1004089 (a link is provided in Appendix A).

- **Lack of networking redundancy**—In some cases, I've seen the order of network adapter port numbers differ from that of the vmnics. In other cases, it turned out they were hooked up in the wrong location. Either way, the result was a lack of redundancy against failure that was originally planned for. This can happen when virtual machine port groups are set up with certain active or standby NICs in a manner that provides redundancy across separate physical switches and via separate network interface cards. What can result are both NICs from a particular port group traversing the same switch.

This type of configuration issue might be noticed immediately if some of the ports are configured as access ports. Hooking into the wrong access port causes you to quickly notice a lack of network connectivity for one of your services, but if they are all trunk ports with the same VLANs being trunked, then there won't be any connectivity or functionality issues.

Implementing the Solution Summary

This chapter focused on the implementation phase of virtualization projects. This phase of the project requires taking a design and translating it into a working infrastructure. A design blueprint is a critical prerequisite to this phase and provides for the greatest success of the implementation in terms of the quality. Even with a well-outlined design in terms of a design blueprint, issues do occur.

It is, then, those decisions about how to tackle those issues that are of utmost importance.

The responsibility of the implementer is not simply translating a design blueprint into a fully implemented virtualization solution. As the implementer, you should review the work of the design laid out and ensure all the pieces of the puzzle are still there and as they were when the design was scoped out. Don't assume anything.

Furthermore, as the implementer, you should question decisions made in the design where necessary. Although it might not be likely best practices are going to be violated, it is very possible solutions will be designed that fail to consider pieces of information that were not known at the time of the design.

In some environments, the infrastructure might never be touched again for months at a time. This is the beauty of VMware and its feature set, including HA and DRS. A host could lose power and virtual machines may begin to contend with each other. If everything is designed and implemented to plan properly, though, there is not a need to manually administer anything. Always implement with this type of self-sufficiency in mind and you will continue laying a solid foundation for the organization's infrastructure.