EMC VSPEX END-USER COMPUTING
Citrix XenDesktop and VMware vSphere with EMC ScaleIO

EMC VSPEX

Abstract
This Proven Infrastructure Guide describes how to design an EMC® VSPEX® end-user computing solution for Citrix XenDesktop using EMC ScaleIO® and VMware vSphere to provide the storage and virtualization platforms.

July 2015
Main features .................................................................................................................. 39

Chapter 3  Sizing the Solution 40
Overview ......................................................................................................................... 41
Reference workload ....................................................................................................... 41
Scaling out the environment .......................................................................................... 42
VSPEX building blocks ................................................................................................. 43
  Building block approach ............................................................................................... 43
  Validated building blocks ............................................................................................ 44
Planning for high availability ........................................................................................ 45
Sizing guidelines ............................................................................................................. 45
  Introduction to the Customer Sizing Worksheet .......................................................... 45
  Using the Customer Sizing Worksheet ......................................................................... 46
  Customizing the building block .................................................................................. 49
  Calculating the building block requirement ............................................................... 60
Fine-tuning hardware resources ................................................................................... 62
Summary ......................................................................................................................... 63

Chapter 4  Solution Implementation 64
Overview ......................................................................................................................... 65
Setting up the network .................................................................................................... 66
  Configuring the infrastructure network ....................................................................... 66
  Configuring the VLANs ............................................................................................... 67
  Completing the network cabling .................................................................................. 67
Installing and configuring the vSphere hosts ............................................................... 68
  Installing vSphere ....................................................................................................... 68
  Configuring vSphere networking ................................................................................. 68
  Connecting the VMware datastores .......................................................................... 69
Installing and configuring the SQL Server database ..................................................... 70
Deploying VMware vCenter Server .............................................................................. 71
Preparing and configuring the storage ......................................................................... 72
  Preparing the ScaleIO environment ............................................................................ 72
  Registering the ScaleIO plug-in ................................................................................ 73
  Uploading the OVA template ...................................................................................... 74
  Accessing the ScaleIO plug-in ................................................................................... 75
  Installing SDC on ESXi ............................................................................................... 75
Deploying ScaleIO ......................................................................................................... 76
Creating volumes .......................................................................................................... 84
Creating datastores ....................................................................................................... 86
Installing the GUI ........................................................................................................... 86
Installing and configuring XenDesktop Delivery Controllers .................................................. 86
Installing the server-side components of XenDesktop ......................................................... 87
Configuring a site .................................................................................................................. 87
Adding a second controller ................................................................................................. 87
Installing Citrix Studio ......................................................................................................... 87
Preparing the master virtual machine ............................................................................... 87
Provisioning the virtual desktops ....................................................................................... 88
Installing and configuring Citrix Provisioning Services (PVS) ........................................... 88
Configuring a PVS server farm .......................................................................................... 89
Adding a second PVS server .............................................................................................. 89
Creating a PVS store .......................................................................................................... 89
Configuring inbound communication ............................................................................... 90
Configuring a bootstrap file .............................................................................................. 90
Configuring boot options 66 and 67 on DHCP server ....................................................... 91
Preparing the master virtual machine ............................................................................. 91
Provisioning the virtual desktops ....................................................................................... 91

Chapter 5  Solution Verification 92
Overview ............................................................................................................................... 93
Verifying the installation with the post-installation checklist .............................................. 94
Deploying and testing a single virtual desktop .................................................................. 94
Verifying component redundancy .................................................................................... 94

Chapter 6  Reference Documentation 95
EMC documentation .......................................................................................................... 96
Other documentation .......................................................................................................... 96

Appendix A  Customer Sizing Worksheet 97
Customer Sizing Worksheet for end-user computing ...................................................... 98
Printing the worksheet ....................................................................................................... 99

Appendix B  Configuration Worksheet 100
Customer Configuration Worksheet .................................................................................. 101
Printing the worksheet ....................................................................................................... 101
Figure 39. Configure networks .................................................................82
Figure 40. Create a new data network .........................................................83
Figure 41. Create volume .........................................................................84
Figure 42. Create volume .........................................................................85
Figure 43. Configure Bootstrap dialog box ......................................................90
Figure 44. Printable customer sizing worksheet ..............................................99
Figure 45. Opening attachments in a PDF file .............................................101

Tables
Table 1. Terminology ..............................................................................13
Table 2. Solution components .................................................................16
Table 3. Server hardware .......................................................................25
Table 4. Recommended 10 Gb switched Ethernet network layer ..................27
Table 5. System requirements for ScaleIOVM ............................................36
Table 6. VSPLEX end-user computing: Design process ...............................41
Table 7. Reference virtual desktop characteristics ......................................42
Table 8. MCS-provisioned non-PvD desktops ...............................................44
Table 9. PVS-provisioned non-PvD desktops ...............................................44
Table 10. MCS-provisioned PVD desktops ..................................................44
Table 11. PVS-provisioned Pvd desktops .....................................................45
Table 12. MCS and PVS-provisioned HSD desktops ....................................45
Table 13. Customer Sizing Worksheet example .........................................46
Table 14. Reference virtual desktop resources ..........................................47
Table 15. CPU desktop resources .............................................................49
Table 16. Maximum number of virtual desktops per node ...........................50
Table 17. Number of nodes to support 150 non-PvD virtual desktops ..........50
Table 18. Number of nodes to support 150 PVD virtual desktops ................51
Table 19. Number of nodes to support 150 HSD virtual desktops ..............51
Table 20. Baseline building block sizing calculation: Example 1 ..................52
Table 21. User-customized building block node configuration: Example 1 ....52
Table 22. User-customized building block sizing calculation: Example 2 .......53
Table 23. User-customized building block with added storage disk: Example 3..54
Table 24. Baseline building block sizing calculation: Example 4 ..................55
Table 25. User-customized building block node configuration: Example 5 ......55
Table 26. User-customized building block sizing calculation: Example 5 ........55
Table 27. User-customized building block with added storage disk: Example 6 ..56
Table 28. Baseline building block sizing calculation: Example 7 ...................57
Table 29. User-customized building block node configuration: Example 8 .......58
Contents

Table 30. User-customized building block sizing calculation: Example 8 ............58
Table 31. User-customized building block with added storage disk: Example 9 .59
Table 32. Node scaling example ........................................................................60
Table 33. Server resource component totals ....................................................63
Table 34. Implementation process overview ......................................................65
Table 35. Tasks for switch and network configuration ........................................66
Table 36. Tasks for server installation ...............................................................68
Table 37. Tasks for SQL Server database setup ................................................70
Table 38. Tasks for vCenter configuration ........................................................71
Table 39. Tasks for storage configuration ........................................................72
Table 40. Tasks for XenDesktop controller setup .............................................86
Table 41. Tasks for XenDesktop controller setup .............................................88
Table 42. Tasks for testing the installation .......................................................93
Table 43. Customer Sizing Worksheet ...............................................................98
Table 44. ScaleIO information ........................................................................101
Table 45. Common server information ..........................................................101
Table 46. vSphere Server information .............................................................102
Table 47. Network infrastructure information ................................................102
Table 48. VLAN information ..........................................................................102
Table 49. Service accounts ............................................................................103
This chapter presents the following topics:

- **Introduction** ................................................................. 10
- **Purpose of this guide** ..................................................... 11
- **Business value** ............................................................... 12
- **Audience** ......................................................................... 13
- **Terminology** ..................................................................... 13
Introduction

EMC® has joined forces with the industry-leading providers of IT infrastructure to create a complete virtualization solution that accelerates the deployment of the private cloud and Citrix XenDesktop virtual desktops. EMC® VSPEX® enables customers to accelerate their IT transformation with faster deployment, greater simplicity and choice, higher efficiency, and lower risk, compared to the challenges and complexity of building an IT infrastructure themselves.

VSPEX validation by EMC ensures predictable performance and enables customers to select technology that uses their existing or newly acquired IT infrastructure while eliminating planning, sizing, and configuration burdens. VSPEX provides a virtual infrastructure for customers who want the simplicity of truly converged infrastructures with more choice in individual stack components.

VSPEX Proven Infrastructures, as shown in Figure 1, are modular, virtualized infrastructures validated by EMC and delivered by EMC VSPEX partners. They include virtualization, server, network, and storage layers. Partners can select the virtualization, server, and network technologies that best fit a customer's environment, while the server local disks with elastic EMC ScaleIO® software provide the storage.
Purpose of this guide

The EMC VSPEX end-user computing architecture provides customers with a modern system capable of hosting a large number of virtual desktops at a consistent performance level. This VSPEX end-user computing solution for Citrix XenDesktop runs on a VMware vSphere virtualization layer backed by EMC ScaleIO, which provides the storage. In this solution, the desktop virtualization infrastructure components are layered on a VSPEX Private Cloud for VMware vSphere Proven Infrastructure, while the desktops are hosted on dedicated resources.

The compute and network components, which are defined by VSPEX partners, are designed to be redundant and sufficiently powerful to handle the processing and data needs of a large virtual machine environment.
Chapter 1: Executive Summary

This validated VSPEX end-user-computing solution supports up to 500 virtual desktops. The validated configurations are based on a reference desktop workload and form the basis for creating cost-effective, custom solutions for individual customers.

An end-user computing or virtual desktop infrastructure (VDI) is a complex system offering. This Proven Infrastructure Guide describes how to design an end-user computing solution for Citrix XenDesktop according to best practices. It also describes how to size the solution to fit the customer’s needs using the EMC VSPEX Sizing Tool or the Customer Sizing Worksheet, provided in Appendix A.

Business value

Employees are more mobile than ever, and they expect access to business-critical data and applications from any location and any device. They want the flexibility to bring their own devices to work. IT departments are increasingly investigating and supporting these bring your own device (BYOD) initiatives. However, these initiatives increase complexity and bring additional security challenges. Deploying virtual desktops is one way to manage the additional layers of complexity and to safeguard sensitive information.

Implementing large-scale virtual desktop environments presents many challenges. Administrators must rapidly roll out persistent or non-persistent desktops for all users—task workers, knowledge workers, and power users—while offering an outstanding user experience that outperforms physical desktops.

In addition to providing the required performance, a virtual desktop solution must be simple to deploy, manage, and scale, and offer substantial cost savings over physical desktops. Storage is also a critical component of an effective virtual desktop solution.

EMC VSPEX Proven Infrastructures are designed to help you address the most serious of IT challenges by creating solutions that are simple, efficient, and flexible. In this solution, they enable you to take advantage of the many possibilities that VSPEX with ScaleIO offers.

This VSPEX end-user computing solution for Citrix XenDesktop provides:

- End-to-end virtualization, using the capabilities of the unified infrastructure components
- Efficient virtualization of up to 500 virtual desktops for varied customer use cases
- Reliable, flexible, and scalable reference architectures
Audience

This guide is intended for internal EMC personnel and qualified EMC VSPEX Partners. The guide assumes that VSPEX partners who intend to deploy this VSPEX Proven Infrastructure for Citrix XenDesktop have the necessary training and background to install and configure an end-user computing solution based on Citrix XenDesktop using vSphere as the hypervisor, ScaleIO storage, and associated infrastructure.

You should also be familiar with the customer’s infrastructure and database security policies.

This guide directs you to external references where applicable. EMC recommends that partners implementing this solution are familiar with these documents. For details, refer to Chapter 6: Reference Documentation.

Terminology

Table 1 lists terminology used in this guide.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-user computing</td>
<td>Computing that decouples the desktop from the physical machine. In an end-user computing environment, the desktop operating system (OS) and applications reside inside a virtual machine running on a host computer, with data residing on shared storage. Users access their virtual desktop from any computer or mobile device over a private network or Internet connection.</td>
</tr>
<tr>
<td>Reference architecture</td>
<td>A validated architecture that supports this VSPEX end-user-computing solution at a particular point of scale.</td>
</tr>
<tr>
<td>Reference workload</td>
<td>For VSPEX end-user computing solutions, a single virtual desktop—the reference virtual desktop—with the workload characteristics indicated in Table 7 on page 42. By comparing the customer’s actual usage to this reference workload, you can extrapolate which reference architecture to select as the basis for the customer’s VSPEX deployment. Refer to Reference workload for details.</td>
</tr>
</tbody>
</table>
This chapter presents the following topics:

- **Overview** .................................................................................................................. 15
- **Solution architecture** ................................................................................................. 15
- **Key components** ........................................................................................................ 16
- **Desktop virtualization broker** ..................................................................................... 17
- **Virtualization layer** .................................................................................................... 20
- **Compute layer** ............................................................................................................ 24
- **Network layer** ............................................................................................................. 27
- **Storage layer** ............................................................................................................... 30
- **Citrix ShareFile StorageZones solution** ..................................................................... 39
Overview

This chapter provides an overview of the EMC VSPEX end-user computing solution for Citrix XenDesktop on VMware vSphere and the key technologies used in the solution. EMC has designed and proven this solution to provide the desktop virtualization, server, network, and storage resources to support your customers’ virtual desktop deployment.

The desktop virtualization infrastructure components of the solution are designed to be layered on a VSPEX Private Cloud for VMware vSphere Proven Infrastructure. However, the reference architectures do not include configuration details for the underlying infrastructure. Refer to the EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide for information on configuring the required infrastructure components.

Solution architecture

High-level architecture

The EMC VSPEX end-user computing solution for Citrix XenDesktop provides a complete system architecture that supports up to 500 virtual desktops.

The solution uses server local disks with EMC ScaleIO software and VMware vSphere to provide the storage and virtualization platforms for a Citrix XenDesktop environment of Microsoft Windows 7 virtual desktops provisioned by Citrix Provisioning Services (PVS) or Machine Creation Services (MCS).

The desktop virtualization infrastructure components are layered on a VSPEX Private Cloud for VMware vSphere Proven Infrastructure, backed by the elastic EMC ScaleIO software, which provides the storage. The infrastructure services for the solution, as shown in Figure 2, can be provided by existing infrastructure at the customer site, by the VSPEX Private Cloud, or by deploying them as dedicated resources as part of the solution.

Planning and designing the storage infrastructure for a Citrix XenDesktop environment is critical because the shared storage must be able to absorb large bursts of I/O that occur during the day. These bursts can lead to periods of erratic and unpredictable virtual desktop performance. Users can adapt to slow performance, but unpredictable performance frustrates them and reduces efficiency.

To provide predictable performance for end-user computing solutions, the storage system must be able to handle the peak I/O load from the clients while keeping response time to a minimum. In this solution, we\(^1\) used EMC ScaleIO software to build a storage system with high performance and scalability using the servers’ local disks.

\(^1\) In this document, “we” refers to the EMC Solutions engineering team that validated the solution.
Figure 2 shows the logical architecture of this solution.

Figure 2. Logical architecture

Key components

Table 2 provides an overview of the key technologies used in this solution.

Table 2. Solution components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| Desktop virtualization broker    | Manages the provisioning, allocation, maintenance, and removal of the virtual desktop images that are provided to its users. This critical software enables on-demand creation of desktop images, allow maintenance to the image without affecting user productivity, and prevent the environment from growing in an unconstrained way.
|                                  | The desktop broker in this solution is Citrix XenDesktop.                                                                                                                                                     |
| Virtualization layer             | Allows the physical implementation of resources to be decoupled from the applications that use them. In other words, the application’s view of the resources available is no longer directly tied to the hardware. This enables many key features in the end-user computing concept.
|                                  | This solution uses VMware vSphere for the virtualization layer.                                                                                                                                                |
### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute layer</td>
<td>Provides memory and processing resources for the virtualization layer software as well as for the applications running in the infrastructure. The VSPEX program defines the minimum amount of compute layer resources required, but allows the customer to implement the requirements using any server hardware that meets these requirements.</td>
</tr>
<tr>
<td>Network layer</td>
<td>Connects users to the resources they need and connects the storage layer to the compute layer. The VSPEX program defines the minimum number of network ports required for the solution and provides general guidance on network architecture, but allows the customer to implement the requirements using any network hardware that meets these requirements.</td>
</tr>
<tr>
<td>Storage layer</td>
<td>Functions as a critical resource for the implementation of the end-user computing environment. The storage layer must be able to absorb large bursts of activity as they occur without unduly affecting the user experience. This solution uses EMC ScaleIO software with servers' local disks to handle this workload efficiently.</td>
</tr>
<tr>
<td>Citrix ShareFile StorageZones solution</td>
<td>Provides cloud-based file-sharing and storage services. Citrix ShareFile Storage Zones is an optional component of this solution.</td>
</tr>
</tbody>
</table>

### Desktop virtualization broker

#### Overview

Desktop virtualization encapsulates and hosts desktop services on centralized computing resources at remote data centers. This enables end users to connect to their virtual desktops from different types of devices across a network connection. Devices can include desktops, laptops, thin clients, zero clients, smartphones, and tablets.

In this solution, we used Citrix XenDesktop to provision, manage, broker, and monitor the desktop virtualization environment.

#### Services

**Citrix XenDesktop 7.6**

XenDesktop is the desktop virtualization solution from Citrix that enables virtual desktops to run on the vSphere virtualization environment. Citrix XenDesktop 7.6 integrates Citrix XenApp application delivery technologies and XenDesktop desktop virtualization technologies into a single architecture and management experience. This new architecture unifies both management and delivery components to enable a scalable, simple, efficient, and manageable solution for delivering Windows applications and desktops as secure mobile services to users anywhere on any device.

Figure 3 shows the XenDesktop 7.6 architecture components.
The XenDesktop 7.6 architecture includes the following components:

- **Citrix Director**—Director is a web-based tool that enables IT support and help desk teams to monitor an environment, troubleshoot issues before they become system-critical, and perform support tasks for end users.

- **Citrix Receiver**—Installed on user devices, Citrix Receiver provides users with quick, secure, self-service access to documents, applications, and desktops from any of the user’s devices including smartphones, tablets, and PCs. Receiver provides on-demand access to Windows, web, and software-as-a-service (SaaS) applications.

- **Citrix StoreFront**—StoreFront provides authentication and resource delivery services for Citrix Receiver. It enables centralized control of resources and provides users with on-demand, self-service access to their desktops and applications.

- **Citrix Studio**—Studio is the management console that enables you to configure and manage your deployment, eliminating the need for separate consoles for managing delivery of applications and desktops. Studio provides various wizards to guide you through the process of setting up your environment, creating your workloads to host applications and desktops, and assigning applications and desktops to users.

- **Delivery Controller**—Installed on servers in the data center, Delivery Controller consists of services that communicate with the hypervisor to distribute applications and desktops, authenticate and manage user access, and broker connections between users and their virtual desktops and applications. Delivery Controller manages the state of the desktops and starting and stopping them based on demand and administrative configuration. In some editions, the controller enables you to install profile management to manage user personalization settings in virtualized or physical Windows environments.
• **License Server**—License Server assigns user or device licenses to the XenDesktop environment. Install the License Server along with other Citrix XenDesktop components or on a separate virtual/physical machine.

• **Virtual Delivery Agent (VDA)**—Installed on server or workstation operating systems, the VDA enables connections for desktops and applications. For remote PC access, install the VDA on the office PC.

• **Database**—Database stores all the XenDesktop site configuration and session information. Microsoft SQL Server is required as a database server.

• **Server OS machines**—The server OS machines are virtual machines or physical machines, based on the Windows Server operating system, that are used for delivering applications or hosted shared desktops (HSDs) to users.

• **Desktop OS machines**—The desktop OS machines are virtual or physical machines, based on the Windows Desktop operating system that deliver personalized desktops to users or applications.

**Machine Creation Services**

Machine Creation Services (MCS) is a provisioning tool integrated with the XenDesktop management interface, Citrix Studio, enabling you to provision, manage, and decommission desktops from a centralized point of management.

With MCS, you can manage several machine types within a catalog in Citrix Studio. Desktop customization is persistent for machines that use the Personal vDisk (PvDisk or PvD) feature, while non-Personal vDisk machines are appropriate if desktop changes are to be discarded when the user logs off.

**Citrix Provisioning Services**

Citrix Provisioning Services (PVS) takes a different approach from traditional desktop imaging solutions by fundamentally changing the relationship between hardware and the software that runs on it. By streaming a single shared disk image (vDisk) instead of copying images to individual machines, PVS enables organizations to reduce the number of disk images that they manage. As the number of machines continues to grow, PVS provides the efficiency of centralized management with the benefits of distributed processing.

Because machines stream disk data dynamically in real time from a single shared image, machine image consistency is ensured. In addition, large pools of machines can completely change their configuration, applications, and even OS during a reboot operation.

**Citrix Personal vDisk**

The Citrix Personal vDisk (PvDisk or PvD) feature enables users to preserve customization settings and user-installed applications in a pooled desktop by redirecting the changes from the user’s pooled virtual machine to a separate Personal vDisk. During runtime, the content of the Personal vDisk is blended with the content from the base virtual machine to provide a unified experience to the end user. The Personal vDisk data is preserved during reboot and refresh operations.
Citrix Profile Management

Citrix Profile Management preserves user profiles and dynamically synchronizes them with a remote profile repository. Profile Management downloads a user’s remote profile dynamically when the user logs in to XenDesktop, and applies personal settings to desktops and applications regardless of the user's login location or client device.

The combination of Profile Management and pooled desktops provides the experience of a dedicated desktop while potentially minimizing the amount of storage required in an organization.

Virtualization layer

Overview

VMware vSphere

VMware vSphere is the leading virtualization platform in the industry. It provides flexibility and cost savings by enabling the consolidation of large, inefficient server farms into nimble, reliable infrastructures. The core VMware vSphere components are the VMware vSphere hypervisor and VMware vCenter Server for system management.

This solution uses VMware vSphere Desktop Edition, which is intended for customers who want to purchase vSphere licenses for desktop virtualization only. vSphere Desktop provides the full range of features and functionalities of the vSphere Enterprise Plus edition, enabling customers to achieve scalability, high availability, and optimal performance for all of their desktop workloads. vSphere Desktop also comes with unlimited vRAM entitlement.

VMware vCenter Server

VMware vCenter Server is a centralized platform for managing vSphere environments. It provides administrators with a single interface for all aspects of monitoring, managing, and maintaining the virtual infrastructure and can be accessed from multiple devices.

vCenter Server also manages advanced features such as vSphere High Availability (HA), vSphere Distributed Resource Scheduler (DRS), vSphere vMotion, and vSphere Update Manager.

VMware vSphere High Availability

VMware vSphere High Availability (HA) provides uniform, cost-effective failover protection against hardware and OS outages:

- If the virtual machine OS has an error, you can automatically restart the virtual machine on the same hardware.
- If the physical hardware has an error, you can automatically restart the impacted virtual machines on other servers in the cluster.
VMware vSphere provides several advanced features that help optimize performance and overall use of resources. This section describes the key features for memory management and considerations for using them with a VSPEX solution.

Memory virtualization techniques allow the vSphere hypervisor to abstract physical host resources, such as memory, to provide resource isolation across multiple virtual machines while avoiding resource exhaustion. In cases where advanced processors (such as Intel processors with EPT support) are deployed, memory abstraction takes place within the CPU. Otherwise, it occurs within the hypervisor itself using a feature known as shadow page tables.

vSphere provides the following memory management techniques:

- **Memory over-commitment**
  Memory over-commitment occurs when more memory is allocated to virtual machines than is physically present in a VMware vSphere host. Using sophisticated techniques such as ballooning and transparent page sharing, vSphere is able to handle memory over-commitment without any performance degradation. However, if more memory is being actively used than is present on the server, vSphere might resort to swapping portions of a virtual machine's memory.

- **Non-Uniform Memory Access (NUMA)**
  vSphere uses a NUMA load-balancer to assign a home node to a virtual machine. Memory access is local and provides the best performance possible because memory for the virtual machine is allocated from the home node. Applications that do not directly support NUMA also benefit from this feature.

- **Transparent page sharing**
  Virtual machines running similar operating systems and applications typically have identical sets of memory content. Page sharing allows the hypervisor to reclaim the redundant copies and return them to the host’s free memory pool for reuse.

- **Memory compression**
  vSphere uses memory compression to store pages that would otherwise be swapped out to disk through host swapping in a compression cache located in the main memory.

- **Memory ballooning**
  Memory ballooning relieves host resource exhaustion by allocating free pages from the virtual machine to the host for reuse with little to no impact on the application’s performance.

- **Hypervisor swapping**
  Hypervisor swapping causes the host to force arbitrary virtual machine pages out to disk.

For further information, refer to the VMware white paper *Understanding Memory Resource Management in VMware vSphere 5.0*. 
Memory configuration guidelines

Carefully consider proper sizing when configuring the server memory. This section provides guidelines for allocating memory to virtual machines considering vSphere overhead and the virtual machine memory settings.

vSphere memory overhead

Memory space overhead associated with virtualizing memory resources has two components:

- System overhead for the VMkernel
- Additional overhead for each virtual machine

The overhead for the VMkernel is fixed, whereas the amount of additional memory for each virtual machine depends on the number of virtual CPUs and the amount of memory configured for the guest OS.

Virtual machine memory settings

Figure 4 shows the memory settings parameters in a virtual machine, including:

- Configured memory—Physical memory allocated to the virtual machine at the time it is created
- Reserved memory—Memory that is guaranteed to the virtual machine
- Touched memory—Memory that is active or in use by the virtual machine
- Swappable—Memory that can be de-allocated from the virtual machine if the host is under memory pressure from other virtual machines using ballooning, compression, or swapping

EMC recommends that you follow these best practices for virtual machine memory settings:

- Do not disable the default memory-reclamation techniques. These lightweight processes provide flexibility with minimal impact to workloads.
- Carefully consider sizing the memory allocation for the virtual machines.
Over-allocation wastes resources, while under-allocation causes performance impacts that can affect other virtual machines’ sharing resources. Over-committing can lead to resource exhaustion if the hypervisor cannot procure memory resources. In severe cases, when hypervisor swapping occurs, virtual machine performance might be adversely affected.

Having performance baselines of your virtual machine workloads assists in this process.

**Allocating memory to virtual machines**

Server capacity is required for two purposes:

- To support the required infrastructure services such as authentication, DNS, and database
  
  For further details on the hosting requirements for these infrastructure services, refer to *EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide* listed in Chapter 6.

- To support the virtualized desktop infrastructure
  
  Each virtual desktop is assigned 2 GB of memory, as defined in Table 7 on page 42. We validate the solution with statically assigned memory and no over-commitment of memory resources. If memory over-commitment is used in a real-world environment, regularly monitor the system memory utilization and associated page file I/O activity to ensure that a memory shortfall does not cause unexpected results.

**High availability and failover**

EMC recommends configuring high availability in the virtualization layer and automatically allowing the hypervisor to restart virtual machines that fail. Figure 5 illustrates the hypervisor layer responding to a failure in the compute layer.

![Figure 5. High availability at the virtualization layer](image)

By implementing high availability at the virtualization layer, the infrastructure will attempt to keep as many services running as possible, even in the event of a hardware failure.
Chapter 2: Solution Architecture Overview

Compute layer

Overview

VSPEX solutions are designed to run on a wide variety of server platforms. EMC ScaleIO can run on any server. VSPEX defines the minimum CPU, memory, and disk resources required, and the customer can use any server platform that meets or exceeds the minimum requirements.

The ScaleIO environment is designed to work with a minimum of three physical server nodes. The physical server node provides all the hardware resources for the system. With the EMC ScaleIO software, all the compute and storage resources on the physical servers are converged in the single layer architecture, aggregating capacity and performance and simplifying management.

Configuration guidelines

When choosing a server platform you must consider both the technical requirements of the environment, as well as the supportability of the platform, existing relationships with the server provider, advanced performance, and management features, and many other factors. For example:

- From a virtualization perspective, if you understand the system's workload, you can use features such as memory ballooning and transparent page sharing to reduce the aggregate memory requirement.

- If the virtual machine pool does not have a high level of peak or concurrent usage, you can reduce the number of vCPUs. Conversely, if the applications being deployed are highly computational, you might need to increase the number of CPUs and the amount of memory.

At a minimum, the server infrastructure must meet the following requirements:

- Sufficient CPU cores, memory, and disks to support the required number and types of virtual machines
- Sufficient network connections to enable redundant connectivity to the system switches
- Sufficient excess capacity to enable the environment to withstand a server failure and failover

Server best practices

Consider the following best practices for the server layer:

- **Use identical server units.**
  
  Use identical or at least compatible servers. VSPEX implements high-availability technologies at the hypervisor level that might require similar instruction sets on the underlying physical hardware. By implementing VSPEX on identical server units, you can minimize compatibility problems in this area.

- **Use recent processor technologies.**
  
  For new deployments, use recent releases of common processor technologies. These will perform as well as, or better than, the systems we used to validate the solution.
• **Implement high availability to accommodate single server failures.**

Implement the high-availability features available in the virtualization layer to ensure that the compute layer has sufficient resources to accommodate at least single server failures. This will also allow you to implement minimal-downtime during upgrades.

**Note:** When implementing hypervisor layer high availability, the largest virtual machine you can create is constrained by the smallest physical server in the environment.

• **Monitor resource utilization and adapt as needed.**

In any running system, monitor the resource utilization and adapt as needed. For example, the reference MCS-provisioned PvD or non-PvD virtual desktop and required hardware resources assume that there are no more than six virtual CPUs for each physical processor core (6:1 ratio). There are also no more than four virtual CPUs for each physical processor core (4:1 ratio) for HSD desktops and all types of PVS provisioned desktops.

In most cases, this provides enough resources for the hosted virtual desktops; however, this ratio might not be appropriate in all cases. EMC recommends monitoring CPU utilization at the hypervisor layer to determine if more resources are required and then adding as needed.

**Validated server hardware**

Table 3 identifies the server hardware and the configurations that we validated for this solution.

**Table 3. Server hardware**

<table>
<thead>
<tr>
<th>Servers for virtual desktops</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCPU</td>
<td>• MCS</td>
</tr>
<tr>
<td></td>
<td>• PvD and non-PvD desktop: 1 vCPU (6 PvD and non-PvD desktops per Core)</td>
</tr>
<tr>
<td></td>
<td>• HSD: 0.25 vCPU (4 HSDs per core)</td>
</tr>
<tr>
<td></td>
<td>• PVS</td>
</tr>
<tr>
<td></td>
<td>• PvD and non-PvD desktop: 1 vCPU (4 PvD and non-PvD desktops per core)</td>
</tr>
<tr>
<td></td>
<td>• HSD: 0.25 vCPU (4 HSDs per core)</td>
</tr>
<tr>
<td>Memory</td>
<td>• Non-PvD and PvD: 2 GB RAM per desktop:</td>
</tr>
<tr>
<td></td>
<td>• 2 GB RAM reservation per vSphere host</td>
</tr>
<tr>
<td></td>
<td>• 11 GB RAM for ScaleIO (3 GB for SVM + 8 GB for RAM cache)</td>
</tr>
<tr>
<td></td>
<td>• HSD: 0.67 GB RAM per desktop:</td>
</tr>
<tr>
<td></td>
<td>• 2 GB RAM reservation per vSphere host</td>
</tr>
<tr>
<td></td>
<td>• 11 GB RAM for ScaleIO (3 GB for SVM + 8 GB for RAM cache)</td>
</tr>
</tbody>
</table>
## Servers for virtual desktops

<table>
<thead>
<tr>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
</tr>
<tr>
<td>• 2 x 1 GbE NICs per server for management network</td>
</tr>
<tr>
<td>• 2 x 10 GbE NICs per server for data network</td>
</tr>
</tbody>
</table>

### Notes:
- The 6:1 vCPU (4:1 for HSD) to physical core ratio applies to the MCS-provisioned reference workload and the 4:1 vCPU to physical core ratio applies to all PVS-provisioned reference workloads defined in this guide.
- The infrastructure requires one additional server to support VMware vSphere HA in addition to the minimum requirement in Table 3.

### High availability and failover

While the solution offers flexibility in choosing your compute layer server, using enterprise class servers designed for data centers is best. This type of server has redundant power supplies, as shown in Figure 6. You should connect these power supplies to separate power distribution units (PDUs) in accordance with your server vendor’s best practices.

![Figure 6. Redundant power supplies](image)

We also recommend that you configure high availability in the virtualization layer. This means that you must configure the compute layer with enough resources to ensure that the total number of available resources meets the needs of the environment, even with a server failure.
Network layer

Overview
The infrastructure network requires redundant network links for each vSphere host. This configuration provides both redundancy and additional network bandwidth. Redundant network links are required regardless of whether the network infrastructure already exists or you are deploying it with other solution components.

Configuration guidelines
This section provides guidelines for setting up a redundant, highly available network configuration. The guidelines include network redundancy, link aggregation, and traffic isolation.

ScaleIO network
The ScaleIO network creates a Redundant Array of Independent Nodes (RAIN) topology between the server nodes. In practice, this means that the system distributes data so that the loss of a single node will not affect data availability. This, in turn, requires that the ScaleIO nodes send data to other nodes to maintain consistency. A high-speed, low-latency IP network is required for this to work correctly.

We created the test environment with redundant 10 Gb Ethernet (GbE) networks. During testing, at small points of scale, the network was not heavily used. For that reason, at small points of scale, you can implement the solution using 1 GbE networking. We recommend a 10 GbE IP network designed for high availability, as shown in Table 4.

Table 4. Recommended 10 Gb switched Ethernet network layer

<table>
<thead>
<tr>
<th>Nodes</th>
<th>10 Gb switched Ethernet</th>
<th>1 Gb switched Ethernet</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Recommended</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Possible</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Not recommended</td>
</tr>
</tbody>
</table>

Network redundancy
The infrastructure network requires redundant network links for each vSphere host, the storage array, the switch interconnect ports, and the switch uplink ports. This configuration provides both redundancy and the required network bandwidth. Figure 7 provides an example of a highly available network topology.
Figure 7. Highly-available network design example

**Link aggregation**

Link aggregation enables multiple active Ethernet connections to appear as a single link with a single MAC address and potentially multiple IP addresses.

We configured the Link Aggregation Control Protocol (LACP) on the vSphere hosts to combine multiple Ethernet ports into a single virtual device. If a link is lost in the Ethernet port, it fails over to another port. We distributed all network traffic across the active links.

**Traffic isolation**

This solution uses virtual local area networks (VLANs) to segregate network traffic of various types to improve throughput, manageability, application separation, high availability, and security.

VLANs segregate network traffic to enable traffic of different types to move over isolated networks. In some cases, physical isolation might be required for regulatory or policy compliance reasons; in many cases, logical isolation using VLANs is sufficient.

This solution calls for a minimum of three VLANs:

- Client access
- Storage
- Management

---

2 A link aggregation resembles an Ethernet channel but uses the LACP IEEE 802.3ad standard. This standard supports link aggregations with two or more ports. All ports in the aggregation must have the same speed and be full duplex.
Figure 8 shows the design of these VLANs.

![Figure 8](image)

**Figure 8. Required networks**

The client access network is for system users, or clients, to communicate with the infrastructure. The storage network is used for communication between the compute layer and the storage layer. The management network provides administrators with dedicated access to the management connections on the storage array, network switches, and hosts. The management network is also used for vMotion traffic.

EMC recommends using at least three networks with one management network and two independent data networks. In this solution, we use two 1 GbE networks for the management network (for redundancy) and two independent 10 GbE networks for the ScaleIO data network.

Each vSphere host has multiple connections to guard against link failures, as shown in Figure 9. Spread these connections across multiple Ethernet switches as protection against component failure in the network.

![Figure 9](image)

**Figure 9. Network layer high availability**

Having no single points of failure in the network layer ensures that the compute layer can access storage and communicate with users even if a component fails.
Chapter 2: Solution Architecture Overview

Storage layer

Overview

Since data and application access is essential to a positive end-user experience, storage is the core of end-user computing. Initially, end-user computing deployments can start small but grow quickly, so scale-out storage support for VDI is important, especially for growing organizations. Successful end-user computing implementations require continuous maintenance and management of complex administrative tasks to keep the dynamic user community productive and happy.

EMC ScaleIO

ScaleIO is a software-only solution that uses hosts’ existing local disks and LAN to realize a virtual SAN that has all the benefits of external storage at a fraction of the cost and the complexity. ScaleIO turns local internal storage into shared block storage that is comparable to, or better than, the more expensive external shared block storage. The lightweight ScaleIO software components are installed in the application hosts (vSphere hosts for this solution) and inter-communicate using a standard LAN to handle the application I/O requests sent to ScaleIO block volumes. An extremely efficient decentralized block I/O flow, combined with a distributed, sliced volume layout, results in a massively parallel I/O system that can scale to thousands of nodes.

ScaleIO is designed and implemented with enterprise-grade resilience as an essential attribute. Furthermore, the software provides efficiently distributed auto-correction processes that resolve media and node failures without requiring administrator involvement. Dynamic and elastic, ScaleIO enables administrators to quickly add or remove nodes and capacity. The software immediately responds to the changes, rebalancing the storage distribution and achieving a layout that optimally suits the new configuration.

Software components

The ScaleIO Data Client (SDC) is a lightweight device driver in each host for which applications or file systems require access to the ScaleIO virtual SAN block devices. The SDC exposes block devices representing the ScaleIO volumes that are currently mapped to that host.

The ScaleIO Data Server (SDS) is also a lightweight software component within each host that contributes local storage to the central ScaleIO virtual SAN.

Convergence of storage and compute

ScaleIO unites the storage and application layers. The hosts that run applications can also be used to realize shared storage, yielding a wall-to-wall, single layer of hosts. Because the same hosts run applications and provide storage for the virtual SAN, a SDC and SDS are typically both installed in each of the participating hosts.

Carefully designed and implemented to consume the minimum required computing resources, the ScaleIO software components have a negligible impact on the applications running in the hosts.
Pure block storage implementation

ScaleIO implements a pure block storage layout. Its entire architecture and data path are optimized for block storage access needs. For example, when an application submits a read I/O request to its SDC, the SDC instantly identifies which SDS is responsible for the specified volume address and then interacts directly with the relevant SDS. The SDS reads the data (by issuing a single read I/O request to its local storage or by fetching the data from the cache in a cache-hit scenario), and returns the result to the SDC. The SDC provides the read data to the application.

This flow is simple, consuming as few resources as necessary. The data moves over the network only once, and a single I/O request is sent to the SDS storage. The write I/O flow is similarly simple and efficient. Unlike some block storage systems that run on top of a file system or object storage that runs on top of a local file system, ScaleIO offers optimal I/O efficiency.

Massively parallel, scale-out I/O architecture

ScaleIO can scale to a large number of nodes, thus breaking the traditional scalability barrier of block storage. Because the SDCs propagate the I/O requests directly to the pertinent SDSs, there is no central point through which the requests move—and thus avoiding a potential bottleneck. This decentralized data flow is crucial to the linearly scalable performance of ScaleIO. Therefore, a large ScaleIO configuration results in a massively parallel system. The more servers or disks the system has, the greater the number of parallel channels available for I/O traffic and the higher the aggregated I/O bandwidth and IOPS will be.

Mix-and-match nodes

The vast majority of traditional scale-out systems are based on a “symmetric brick” architecture. Unfortunately, data centers cannot be standardized on exactly the same bricks for a prolonged period because hardware configurations and capabilities change over time. Therefore, such symmetric scale-out architectures are bound to run in small islands. ScaleIO was designed from the ground up to support a mixture of new and old nodes with dissimilar configurations.

Hardware agnostic

ScaleIO works with existing underlying hardware resources on any platform. Besides its compatibility with various types of disks, networks, and hosts, it can take advantage of the write buffer of existing local RAID controller cards—and can also run in servers that do not have a local RAID controller card.

For the local storage of an SDS, you can use internal disks, directly attached external disks, virtual disks exposed by an internal RAID controller, partitions within such disks, and more. Partitions can be useful to combine system boot partitions with ScaleIO capacity on the same raw disks. If the system already has a large, mostly unused partition, ScaleIO does not require repartitioning of the disk, as the SDS can actually use a file within that partition as its storage space.

Volume mapping and volume sharing

You can map the volumes that ScaleIO exposes to the application clients to one or more clients running in different hosts. You can change the mapping dynamically if necessary. In other words, ScaleIO volumes can be used by applications that expect
shared-everything block access and by applications that expect shared-nothing or shared-nothing-with-failover access.

**Clustered, striped volume layout**

A ScaleIO volume is a block device that is exposed to one or more hosts. It is the equivalent of a logical unit in the SCSI world. ScaleIO breaks each volume into a large number of data chunks, which are scattered across the SDS cluster’s nodes and disks in a fully balanced manner. This layout practically eliminates hot spots across the cluster and allows you to scale the overall system I/O performance by adding nodes or disks. Furthermore, this layout enables a single application that is accessing a single volume to use the full IOPS of all the cluster’s disks. This flexible, dynamic allocation of shared performance resources is one of the major advantages of converged scale-out storage.

**Management and monitoring**

ScaleIO provides several tools to manage and monitor the system, including:

- A command line interface (CLI)
- An active GUI
- A representational state transfer (REST) management application program interface (API) commands.

The CLI gives administrators direct platform access to perform backend configuration actions and obtain monitoring information.

The active GUI, shown in Figure 10, provides system dashboards for capacity, throughput, bandwidth statistics, access to system alerts, and the ability to provision backend devices. The REST management API allows users to execute the same management and monitoring commands available with the CLI using a next-generation, cloud-based interface.

![Figure 10. ScaleIO active GUI](image)
Interoperability

ScaleIO is integrated with VMware and OpenStack to provide greater flexibility in deploying ScaleIO with existing environments. The VMware plug-in facilitates the provisioning of a ScaleIO system in ESX and runs from within the vSphere web interface. Administrators can accomplish basic and advanced administrative tasks quickly and easily in vSphere with the ScaleIO plug-in.

The OpenStack integration ("Cinder" support) allows you to use commodity hardware with ScaleIO, providing a software-defined block volume solution in an OpenStack environment.

Additionally, ScaleIO software can be packaged with EMC ViPR® to provide block data services for commodity and EMC ECS™ hardware platforms.

Enterprise features

Whether you are a service provider delivering hosted infrastructure as a service (IaaS) or your IT department delivers IaaS to functional units within your organization, ScaleIO offers a set of features that gives you complete control over performance, capacity, and data location. For both private cloud data centers and service providers, these features enhance system control and manageability, ensuring that quality of service is met. With ScaleIO, you can limit the amount of performance—IOPS or bandwidth—that selected customers can consume. The limiter allows you to impose and regulate resource distribution to prevent application “hogging” scenarios. You can apply data masking to provide added security for sensitive customer data. ScaleIO offers instantaneous, writeable snapshots for data backups.

For improved read performance, DRAM caching enables you to improve read-access by using SDS server RAM. Fault sets—a group of SDSs that are likely to go down together—can be defined to ensure data mirroring occurs outside the group, improving business continuity. You can create volumes with thin provisioning, providing on-demand storage as well as faster setup and startup times.

Finally, tight integrations with other EMC products are available. You can use ScaleIO with EMC XtremCache™ for flash cache auto tiering to further accelerate application performance.

Figure 11 shows the ScaleIO enterprise features.
Figure 11. ScaleIO enterprise features

**ScaleIO 1.32**

ScaleIO 1.32 includes the following new features and functionality:

- Release of the ScaleIO “free and frictionless” download, a free ScaleIO download for non-production environments with no time, function, or capacity limits
- Support for VMware ESX 6.0 (VMware certified)
- Support for SLES 12
- Installation Manager enhancements
- Additional flexibility during the configuration process
- Enhanced background scanning and data remediation
- Larger SDS support

The solution includes layouts for the server local disks used in validation testing. Each layout balances the available storage capacity with the performance capability of the drives. There are several layers to consider when designing the storage layouts. Specifically, the ScaleIO system has a collection of disks that are assigned to a storage pool. From that pool, you can create volumes to be provisioned as datastores to the VMware vSphere Cluster. Each layer has a specific configuration that is defined for the solution.

It is generally acceptable to replace a drive with another of the same performance characteristic and more capacity or with types that have higher performance characteristics and the same capacity.
Where there is a need to deviate from the proposed number and type of drives specified, ensure that the target layout delivers the same or greater resources to the system.

**Validated storage configuration**

ScaleIO exposes its volumes as iSCSI targets to connect to vSphere as the storage for virtual desktops. Chapter 3 describes how to size the solutions to determine the number of servers and SAS drives needed to support the required virtual desktops the customer requires.

The optional storage for the infrastructure is not included in the storage configurations, but these components can be layered on a VSPEX Private Cloud solution for VMware vSphere. Refer to the *EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide* for the storage sizing.

To create optional storage for user data and user profiles, use existing CIFS shares or create new shares from supported storage arrays. We recommend that you use EMC VNX or VNXe unified storage to create the CIFS shares for user data and user profiles.

**vSphere storage virtualization**

This section provides guidelines for setting up the storage layer to provide high availability and the expected performance level.

VMware vSphere provides host-level storage virtualization. It virtualizes the physical storage and presents the virtualized storage to the virtual machine.

A virtual machine stores its OS and all other files related to the virtual machine activities in a virtual disk. The virtual disk can be one file or multiple files. VMware uses a virtual SCSI controller to present the virtual disk to the guest OS running inside the virtual machine.

The virtual disk resides in either a VMware Virtual Machine File system (VMFS) datastore or an NFS datastore. An additional option, raw device mapping (RDM), allows the virtual infrastructure to connect a physical device directly to a virtual machine. Since ScaleIO is used to build a virtual SAN, this solution does not use an NFS datastore.

Figure 12 shows the various VMware virtual disk types in a ScaleIO solution, including:

- **VMFS**—A cluster file system that provides storage virtualization optimized for virtual machines. You can deploy VMFS over any SCSI-based local or network storage.
- **Raw device mapping** —A mapping file that acts as a proxy for a raw physical storage device. It uses the iSCSI protocol and allows a virtual machine direct access to a volume on the physical storage.
ScaleIO virtual machine

ScaleIO virtual machine (ScaleIOVM) must be deployed in a VMware environment to install the ScaleIO software components, including MDM, SDS, and SDC. Use ScaleIOVM OVA to deploy ScaleIOVM on each VMware vSphere host. Table 5 shows the system requirements for ScaleIOVM.

Table 5. System requirements for ScaleIOVM

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>2 vCPU</td>
</tr>
<tr>
<td>Memory</td>
<td>12 GB</td>
</tr>
<tr>
<td>Disk space</td>
<td>8 GB</td>
</tr>
<tr>
<td>Connectivity</td>
<td>10 GbE</td>
</tr>
</tbody>
</table>

Redundancy scheme and rebuild process

ScaleIO uses a mirroring scheme to protect data against disk and node failures. The ScaleIO architecture supports a distributed two-copy redundancy scheme. When an SDS node or SDS disk fails, applications can continue to access ScaleIO volumes; their data is still available through the remaining mirrors. ScaleIO immediately starts a seamless rebuild process creating another mirror for the data chunks that were lost in the failure. During the rebuild process, those data chunks are copied to free areas across the SDS cluster, so it is not necessary to add capacity to the system.

All the surviving SDS cluster nodes together carry out the rebuild process by using the cluster’s aggregated disk and network bandwidth. As a result, the process is dramatically faster—resulting in a shorter exposure time and less application-performance degradation. When the rebuild process completes, all the data is fully mirrored and healthy again.

If a failed node rejoins the cluster before the rebuild process completes, ScaleIO dynamically uses the rejoined node’s data to further minimize the exposure time and the use of resources. This capability is particularly important for overcoming short outages efficiently.

High availability and failover
Elasticity and rebalancing

Unlike many other systems, a ScaleIO cluster is extremely elastic. Administrators can quickly add and remove capacity and nodes “on the fly” during I/O operations. When a cluster is expanded with new capacity (such as when new SDSs or new disks are added to existing SDSs), ScaleIO immediately responds to the event and rebalances the storage by seamlessly migrating data chunks from the existing SDSs to the new SDSs or disks. Such a migration does not affect the applications, which continue to access the data stored in the migrating chunks.

As shown in Figure 13, by the end of the rebalancing process, all the ScaleIO volumes span across all the SDSs and disks, including the newly added ones, in an optimally balanced manner. Adding SDSs or disks not only increases the available capacity but also increases the performance of the applications as they access their volumes.

Figure 13. Automatic rebalancing when adding disks

When an administrator decreases capacity (for example, by removing SDSs or removing disks from SDSs), ScaleIO performs a seamless migration that rebalances the data across the remaining SDSs and disks in the cluster, as shown in Figure 14.

Figure 14. Automatic rebalancing when removing disks

In all types of rebalancing, ScaleIO migrates the least amount of data possible. Furthermore, ScaleIO is flexible enough to accept new requests to add or remove capacity while still rebalancing previous capacity additions and removals.

Software-only—but as resilient as a hardware array

Traditional storage systems typically combine system software with commodity hardware—which is comparable to application servers’ hardware—to provide enterprise-grade resilience. With its contemporary architecture, ScaleIO provides similar enterprise-grade, no-compromise resilience by running the storage software directly on the application servers. Designed for extensive fault tolerance and high availability, ScaleIO handles all types of failures, including failures of media, connectivity, and nodes, software interruptions, and more. No single point of failure
can interrupt the ScaleIO I/O service. In many cases, ScaleIO can overcome multiple points of failure as well.

**Managing clusters of nodes**

Many storage cluster designs use tightly coupled techniques that might be adequate for a small number of nodes but begin to break when the cluster is larger than a few dozen nodes. The loosely coupled clustering management schemes of ScaleIO provide exceptionally reliable, yet lightweight, failure and failover handling in both small and large clusters.

Most clustering environments assume exclusive ownership of the cluster nodes and might even physically fence or shut down malfunctioning nodes. ScaleIO uses application hosts. The ScaleIO clustering algorithms are designed to work efficiently and reliably without interfering with the applications with which ScaleIO coexists. ScaleIO never disconnects or invokes Intelligent Platform Management Interface shutdowns of malfunctioning nodes because they might still be running healthy applications.

**Protection domains**

Figure 15 shows a large ScaleIO storage pool can be divided into multiple protection domains, each of which contains a set of SDSs. ScaleIO volumes are assigned to specific protection domains. Protection domains are useful for mitigating the risk of a dual point of failure in a two-copy scheme or a triple point of failure in a three-copy scheme.

![Protection domains](image)

**Figure 15. Protection domains**

For example, if two SDSs in different protection domains fail simultaneously, all data remains available. Just as incumbent storage systems can overcome a large number of simultaneous disk failures as long as they do not occur within the same shelf, ScaleIO can overcome a large number of simultaneous disk or node failures as long as they do not occur within the same protection domain.
Chapter 2: Solution Architecture Overview

Citrix ShareFile StorageZones solution

Overview

Citrix ShareFile is a cloud-based file sharing and storage service for enterprise-class storage and security. ShareFile enables users to securely share documents with other users. ShareFile users include employees and users who are outside the enterprise directory and are referred to as clients.

ShareFile StorageZones enables businesses to share files across the organization while meeting compliance and regulatory concerns. StorageZones enables customers to keep their data on storage systems that are on-premises. It facilitates sharing of large files with full encryption and provides the ability to synchronize files with multiple devices.

By keeping data on the premises and closer to users than data residing on the public cloud, StorageZones can provide improved performance and security.

Main features

The main features available to ShareFile StorageZones users are:

- Use of StorageZones with or instead of ShareFile-managed cloud storage
- Ability to configure Citrix CloudGateway Enterprise to integrate ShareFile services with Citrix Receiver for user authentication and user provisioning
- Automated reconciliation between the ShareFile cloud and an organization’s StorageZones deployment
- Automated antivirus scans of uploaded files
- File recovery from Storage Center backup Storage Center is the server component of StorageZones. StorageZones enables you to browse the file records for a particular date and time and tag any files and folders to be restored from Storage Center backup.
- With added infrastructure components, the EMC VSPEX end-user computing solution for Citrix XenDesktop supports ShareFile StorageZones with Storage Center.
This chapter presents the following topics:

- **Overview** .......................................................... 41
- **Reference workload** ........................................... 41
- **Scaling out the environment** .............................. 42
- **VSPEX building blocks** ....................................... 43
- **Planning for high availability** ......................... 45
- **Sizing guidelines** ............................................... 45
Overview

This chapter describes how to design a VSPEX end-user computing solution for Citrix XenDesktop and size it to fit the customer’s needs. It introduces the concepts of a reference workload, building blocks, and validated end-user computing maximums, and describes how to use these building blocks to design your solution. Table 6 outlines the high-level steps you need to complete when sizing the solution.

Table 6. VSPEX end-user computing: Design process

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use the Customer Sizing Worksheet in Appendix A to collect the customer requirements for the end-user computing environment.</td>
</tr>
</tbody>
</table>
| 2    | Use the EMC VSPEX Sizing Tool to determine the recommended VSPEX reference architecture for your end-user computing solution, based on the customer requirements collected in step 1.  

*Note:* If the Sizing Tool is not available, you can manually size the end-user computing solution using the guidelines in this chapter.

Reference workload

VSPEX defines a reference workload to represent a unit of measure for quantifying the resources in the solution reference architectures. By comparing the customer’s actual usage to this reference workload, you can extrapolate which reference architecture to select as the basis for the customer’s VSPEX deployment.

For this solution, we used Login VSI 3.7 to run a user load against the desktops. Login VSI provides the guidance to gauge the maximum number of users a desktop environment can support. The Login VSI workload can be as light, medium, heavy, or custom. The reference workload for this solution used the Login VSI medium workload.

For VSPEX end-user computing solutions, the reference workload is defined as a single virtual desktop—the reference virtual desktop—with the workload characteristics indicated in Table 7.

The equivalent number of reference virtual desktops for a particular resource requirement is determined by translating the resource requirement to the number of reference virtual desktops needed to meet that requirement.
### Table 7. Reference virtual desktop characteristics

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
</table>
| Virtual desktop type                               | • Desktop OS (PvD and non-PvD desktop types): Microsoft Windows 7 Enterprise Edition (32-bit)  
  • Server OS (HSD): Windows Server 2008 R2          |
| Virtual processors per virtual desktop             | • MCS                                                                |
|                                                    |  ▪ PvD and non-PvD desktop: 1 vCPU (6 PvD and non-PvD desktops per core) |
|                                                    |  ▪ HSD: 0.25 vCPU (4 HSDs per core)                                   |
|                                                    | • PVS                                                                 |
|                                                    |  ▪ PvD and non-PvD desktop: 1 vCPU                                   |
|                                                    |  ▪ HSD: 0.25 vCPU (4 HSDs per core)                                   |
| RAM per virtual desktop                            | • PvD and non-PvD desktop: 2 GB                                      |
|                                                    |  • HSD: 0.67 GB                                                      |
| Average IOPS per virtual desktop at steady state   | 8                                                                    |
| using Login VSI 3.7 medium workload                |                                                                       |

This desktop definition is based on user data that resides on shared storage. The I/O profile is defined by using a test framework that runs all desktops concurrently with a steady load generated by the constant use of office-based applications such as browsers and office productivity software.

### Scaling out the environment

ScaleIO is designed to scale up from a minimum of three nodes. This solution validates various configurations of up to eight nodes with up to 10 spindles each. Unlike most traditional storage systems, as the number of servers grows, so does capacity, throughputs, and IOPS. Performance scalability directly aligns with the growth of the deployment. When needed, additional storage and compute resources (that is, additional servers and disk drives) can be added modularly. Storage and compute resources grow together, so the balance between them is maintained. You can also add storage resources if sufficient compute resources are available. Figure 16 shows two ways to add resources to increase the number of MCS-provisioned virtual desktops in an environment.
Sizing the system to meet the virtual server application requirement can be a complicated process. When applications generate an I/O operation, server components, such as server CPU, server dynamic random access memory (DRAM) cache, and disks, serve that I/O. Customers must consider various factors when planning and scaling their storage system to balance capacity, performance, and cost for their applications.

VSPEX uses a building block approach using ScaleIO to linearly scale out and reduce complexity. A building block is a server with predefined CPU, memory, and disk spindles that can support a specific number of virtual desktops. Each building block combines the CPU, memory, and disk spindles as one ScaleIO node to support the needs of the end-user computing environment. Both SDS and SDC are installed on each building block node to assign the server local disk to the ScaleIO storage pool and expose ScaleIO shared block volumes to run the virtual desktops.

Figure 16. Two examples scaling out resources for virtual desktops

Note: We used the same server configuration of each node for sizing in this solution, but ScaleIO supports nodes with different configurations.

VSPEX building blocks

Building block approach
Chapter 3: Sizing the Solution

The building blocks for this solution combined with the node required by the VSPEX private cloud supports the solution’s infrastructure services. For more information about the VSPEX private cloud node sizing, refer to the *EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide* listed in Chapter 6.

**Validated building blocks**

The configuration of a building block includes the physical CPU core number, memory size, and number of disk spindles per server.

Table 8, Table 10, and Table 12 show sample baseline server configurations for one validated node and provide a flexible solution for VSPEX sizing. With this configuration, up to 40 virtual desktops are supported by one building block with both MCS and PVS provisioning methods. *Customizing the building block* provides information about customizing this baseline configuration. Building block configurations are shown for the following desktop types.

**Non-PvD**

The non-PvD desktops are created by selecting the *Desktop OS with Random desktop experience*. A user is randomly connected to one of the available desktops and all changes made by the user are lost upon reboot. Table 8 shows a building block node configuration with MCS-provisioned non-PvD desktops.

**Table 8. MCS-provisioned non-PvD desktops**

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>7</td>
<td>96</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 9 shows a building block node configuration with PVS-provisioned non-PvD desktops.

**Table 9. PVS-provisioned non-PvD desktops**

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>10</td>
<td>96</td>
<td>6</td>
</tr>
</tbody>
</table>

**PvD**

The PvD desktops are created by selecting the *Desktop OS with personal vDisk desktop experience*. A user is assigned to a particular desktop and all the changes made by the user are stored on personal vDisk and saved between reboots. Table 10 shows a building block node configuration with MCS-provisioned PvD desktops.

**Table 10. MCS-provisioned PvD desktops**

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>7</td>
<td>96</td>
<td>8</td>
</tr>
</tbody>
</table>
Table 11 shows a building block node configuration with PVS-provisioned PvD desktops.

### Table 11. PVS-provisioned PvD desktops

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>10</td>
<td>96</td>
<td>8</td>
</tr>
</tbody>
</table>

**HSD**

The HSD desktops are created by selecting the server OS. Multiple users are connected to the same server. However, each user will get a desktop interface. The resources on the server are shared by all the users logged on to that server. All the changes made by the users are lost on logging off. Table 12 shows a building block node configuration with MCS and PVS-provisioned HSD desktops.

### Table 12. MCS and PVS-provisioned HSD desktops

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>10</td>
<td>48</td>
<td>4</td>
</tr>
</tbody>
</table>

**Note:** When sizing the node requirement, reserve at least one node for high availability.

### Planning for high availability

Because of the scale-out multi-node architecture of ScaleIO, you should consider the possibility of losing a system node. ScaleIO is designed to keep data copies on multiple nodes to protect against just such an occurrence. Any node loss affects the virtual machines running on that node, but you need to ensure that it does not affect the other users of the ScaleIO environment.

EMC recommends that you plan for one node more than the workload dictates to ensure that you can support the environment during a service interruption or system maintenance. In the Sizing guideline section, we reserve one extra node to ensure high availability.

### Sizing guidelines

**Introduction to the Customer Sizing Worksheet**

To select the appropriate reference architecture for a customer environment, determine the resource requirements of the environment and then translate these requirements to an equivalent number of reference virtual desktops with the characteristics defined in Table 7. This section describes how to use the Customer Sizing Worksheet to simplify the sizing calculations and factors that contribute to your architecture deployment decision.
The Customer Sizing Worksheet helps you to assess the customer environment and calculate its sizing requirements.

Table 13 shows a completed worksheet for a sample customer environment. Appendix A provides a blank Customer Sizing Worksheet that you can print out and use to help size a customer solution.

### Table 13. Customer Sizing Worksheet example

<table>
<thead>
<tr>
<th>User type</th>
<th>Users</th>
<th>vCPUs</th>
<th>RAM</th>
<th>IOPS</th>
<th>Equivalent reference virtual desktops</th>
<th>Total reference desktops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heavy users</td>
<td></td>
<td>2</td>
<td>8 GB</td>
<td>12</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>Moderate users</td>
<td></td>
<td>2</td>
<td>4 GB</td>
<td>8</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>120</td>
</tr>
<tr>
<td>Typical users</td>
<td></td>
<td>1</td>
<td>2 GB</td>
<td>8</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>400</td>
</tr>
</tbody>
</table>

To complete the Customer Sizing Worksheet, follow these steps:

1. Identify the user types planned for migration into the VSPEX end-user computing environment and the number of users of each type.
2. For each user type, determine the compute resource requirements in terms of vCPUs, memory (GB), storage performance (IOPS), and storage capacity.
3. For each resource type and user type, determine the equivalent reference virtual desktops requirements—that is, the number of reference virtual desktops required to meet the specified resource requirements.
4. Determine the total number of reference desktops needed from the resource pool for the customer environment.
Determining the resource requirements

Consider the following factors when determining resource requirements.

**CPU**

The reference virtual desktop outlined in Table 7 assumes that most desktop applications are optimized for a single CPU. If one type of user requires a desktop with multiple virtual CPUs, modify the proposed virtual desktop count to account for the additional resources. For example, if you virtualize 100 desktops, but 20 users require two CPUs instead of one, your pool must provide resources for 120 virtual desktops.

**Memory**

Memory plays a key role in ensuring application functionality and performance. Each group of desktops will have different targets for the amount of available memory that is considered acceptable. Like the CPU calculation, if a group of users requires additional memory resources, simply adjust the number of planned desktops to accommodate the additional resource requirements.

For example, if there are 100 desktops to be virtualized, but each one needs 4 GB of memory instead of the 2 GB that the reference virtual desktop provides, plan for 200 reference virtual desktops.

**IOPS**

The storage performance requirements for desktops are usually the least understood aspect of performance. The reference virtual desktop uses a workload generated by an industry-recognized tool to run a wide variety of office productivity applications that should be representative of the majority of virtual desktop implementations.

**Storage capacity**

The storage capacity requirement for a desktop can vary widely depending on the type of provisioning, the types of applications in use, and specific customer policies. The virtual desktops in this solution rely on additional shared storage for user profile data and user documents. This requirement is an optional component that can be met by adding specific storage hardware or by using existing file shares in the environment.

Determining the equivalent reference virtual desktops

Once all of the resources are defined, determine the number of equivalent reference virtual desktops by using the relationships listed in Table 14. Round all values up to the closest whole number.

<table>
<thead>
<tr>
<th>Desktop type</th>
<th>Resource</th>
<th>Value for reference virtual desktop</th>
<th>Relationship between requirements and equivalent reference virtual desktops</th>
</tr>
</thead>
<tbody>
<tr>
<td>PvD and non-PvD desktop types</td>
<td>vCPU</td>
<td>1</td>
<td>Equivalent reference virtual desktops = Resource requirements</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>2</td>
<td>Equivalent reference virtual desktops = (Resource requirements) / 2</td>
</tr>
</tbody>
</table>
Chapter 3: Sizing the Solution

<table>
<thead>
<tr>
<th>Desktop type</th>
<th>Resource</th>
<th>Value for reference virtual desktop</th>
<th>Relationship between requirements and equivalent reference virtual desktops</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IOPS</td>
<td>8</td>
<td>Equivalent reference virtual desktops = (Resource requirements)/8</td>
</tr>
<tr>
<td>HSD desktop type</td>
<td>vCPU</td>
<td>0.25</td>
<td>Equivalent reference virtual desktops = (Resource requirements)/0.25</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>0.67</td>
<td>Equivalent reference virtual desktops = (Resource requirements)/0.67</td>
</tr>
<tr>
<td></td>
<td>IOPS</td>
<td>8</td>
<td>Equivalent reference virtual desktops = (Resource requirements)/8</td>
</tr>
</tbody>
</table>

For example, the heavy user type in Table 13 on page 46 requires two virtual CPUs, twelve IOPS, and eight GB of memory for each desktop in a desktop OS environment. This translates to two reference virtual desktops of CPU, four reference virtual desktops of memory, and two reference virtual desktops of IOPS. As shown in Figure 17, the example requires four virtual machines.

Figure 17. Required resources from the reference virtual machine pool

The number of reference virtual desktops required for each user type equals the maximum required for an individual resource. For example, the number of equivalent reference virtual desktops for the heavy user type in Table 13 is four, as this number meets the all resource requirements for IOPS, vCPU, and memory.

To calculate the total number of reference desktops for a user type, multiply the number of equivalent reference virtual desktops for that user type by the number of users.

Determining the total reference virtual desktops

After the worksheet is completed for each user type that the user wants to migrate into the virtual infrastructure, calculate the total number of reference virtual desktops required in the resource pool by totaling the reference virtual desktops for all user types. In the example in Table 13, the total is 200 virtual desktops.
The node configurations shown in Table 8, Table 10, and Table 12 define the CPU, memory, and disk configuration for one server. However, ScaleIO can run on any server. This VSPEX solution also provides more options for the building block node configuration. The user can redefine our building block with different configurations, but this also changes the number of virtual desktops that the building block can support.

To calculate the number of virtual desktops that the new building block can support, we have to consider the following components.

**CPU capability**

CPU requirements depend on the provisioning method and desktop types. Table 15 shows the CPU requirement for each desktop provisioning method and desktop type.

<table>
<thead>
<tr>
<th>Desktop provisioning method</th>
<th>Desktop type</th>
<th>Number of desktops supported per CPU core</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS</td>
<td>Non-PvD</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>PvD</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>HSD</td>
<td>4</td>
</tr>
<tr>
<td>PVS</td>
<td>Non-PvD</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>PvD</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>HSD</td>
<td>4</td>
</tr>
</tbody>
</table>

For example, a server node with 16 physical cores can support up to 96 MCS-provisioned PvD or non-PvD type virtual desktops, but only 64 PVS-provisioned PvD or non-PvD type virtual desktops.

**Memory capability**

To size the memory for a server node, consider the ScaleIO virtual machine and hypervisor use. The ScaleIO virtual machine consumes 3 GB of RAM and reserves 2 GB RAM for the hypervisor. We do not recommend overcommitting memory in this environment.

*Note:* ScaleIO 1.3 introduces the new RAM cache feature, which uses the SDS server RAM. By default, the size of the ScaleIO virtual machine is set to 3 GB of RAM, 128 MB of which are used as SDS server RAM cache. Add to the RAM cache value on top of the 3 GB of RAM required on the ScaleIO virtual machine if more RAM cache is needed. The validated solution used 8 GB RAM cache on the ScaleIO virtual machine.
Chapter 3: Sizing the Solution

**Disk capacity**

ScaleIO uses a Redundant Array of Independent Nodes (RAIIN) topology to ensure data availability. In general, the capacity available is a function of the capacity per node (formatted capacity) and the number of nodes available.

Assuming \( N \) nodes and \( C \) GB of capacity per server, the storage available, \( S \), is:

\[
S = \frac{(N - 1) \times C}{2}
\]

This formula accommodates two copies of data providing the ability to survive a single node failure.

For example, if a customer has three nodes (\( N = 3 \)) and each server has six disks where each disk has 810 GB usable capacity (\( C = 6 \) disks \( \times \) 810 GB), then the available space is:

\[
S = \frac{(3 - 1) \times (6 \times 810)}{2}
\]

\[
S = 4860 \text{ GB}
\]

When using PvD, the customer should consider the additional capacity requirement needed to support the PvD on top of the virtual desktop disks.

**IOPS**

The primary method for adding IOPS capability to a node without considering cache technologies is to increase either the number of disk units or the speed of those units. Table 16 shows the number of virtual desktops supported with different desktop types per node for MCS and PVS provisioning methods.

**Table 16. Maximum number of virtual desktops per node**

<table>
<thead>
<tr>
<th>Reference virtual desktops</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Non-PvD</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>60</td>
<td>9</td>
</tr>
</tbody>
</table>

**Example 1:** If a company needs 150 non-PvD virtual desktops, Table 17 shows a few choices for the number of nodes needed to support 150 non-PvD virtual desktops.

**Table 17. Number of nodes to support 150 non-PvD virtual desktops**

<table>
<thead>
<tr>
<th>10K SAS drives per node</th>
<th>Node number</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8+1</td>
</tr>
<tr>
<td>6</td>
<td>4+1</td>
</tr>
<tr>
<td>9</td>
<td>3+1</td>
</tr>
</tbody>
</table>
Example 2: If a company needs 150 PvD virtual desktops, Table 18 shows a few choices for the number of nodes needed to support 150 PvD virtual desktops.

Table 18. Number of nodes to support 150 PvD virtual desktops

<table>
<thead>
<tr>
<th>10K SAS drives per node</th>
<th>Node number</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8+1</td>
</tr>
<tr>
<td>8</td>
<td>4+1</td>
</tr>
<tr>
<td>12</td>
<td>3+1</td>
</tr>
</tbody>
</table>

Example 3: If a company needs 150 HSD virtual desktops, Table 19 shows a few choices for the number of nodes needed to support 150 HSD virtual desktops.

Table 19. Number of nodes to support 150 HSD virtual desktops

<table>
<thead>
<tr>
<th>10K SAS drives per node</th>
<th>Node number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8+1</td>
</tr>
<tr>
<td>4</td>
<td>4+1</td>
</tr>
<tr>
<td>6</td>
<td>3+1</td>
</tr>
</tbody>
</table>

Note: The values in Table 16, Table 17, Table 18, and Table 19 assume that the CPU and memory resources of each node are sufficient.

Determining the maximum number of MCS-provisioned non-PvD virtual desktops per building block

With the entire configuration defined for the building block node, we calculate the number of virtual desktops that each component can support to determine the maximum number of desktops the building block node can support. In these examples, non-PvD (Desktop OS with random desktop experience) is used to calculate the maximum number of desktops with MCS provisioning.

The following three examples display sizing calculations for determining the maximum number of virtual desktops for a building block:

- Example 1: Baseline building configuration
- Example 2: User-customized building configuration
- Example 3: User-customized building configuration with storage disk addition
Chapter 3: Sizing the Solution

Example 1: Baseline building block configuration

As shown in Table 20, we defined the baseline building block with seven physical CPU cores, 96 GB memory, and six 10K 600 GB SAS drives.

Table 20. Baseline building block sizing calculation: Example 1

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>VMs supported</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores: 7</td>
<td>42</td>
<td>7 cores * 6 desktops per core = 42 virtual machines</td>
</tr>
<tr>
<td>RAM: 96 GB</td>
<td>41</td>
<td>(96 GB total RAM – 2 GB (hypervisor reserved) – 3 GB (ScaleIO VM) – 8 GB (reserved for ScaleIO RAM cache)) / 2 = 41.5</td>
</tr>
<tr>
<td>Storage performance</td>
<td>40</td>
<td>See Table 16</td>
</tr>
</tbody>
</table>

Figure 18 shows how to determine the maximum number of MCS-provisioned virtual desktops that the baseline building block configuration can support.

Figure 18. Determine the maximum number of virtual desktops: Example 1

For example, if the customer uses three baseline building blocks to build a ScaleIO system, the system should support 80 virtual desktops (2 x 40, with one building block reserved for high availability).

Example 2: User-customized building configuration

The customer can customize a larger building block configuration, as shown in Table 21.

Table 21. User-customized building block node configuration: Example 1

<table>
<thead>
<tr>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>192</td>
<td>9</td>
</tr>
</tbody>
</table>
Table 22 shows the calculations for larger building block sizing.

**Table 22. User-customized building block sizing calculation: Example 2**

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>VMs supported</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores: 16</td>
<td>96</td>
<td>16 cores * 6 desktops per core = 96 virtual machines</td>
</tr>
<tr>
<td>RAM: 192 GB</td>
<td>89</td>
<td>(192 GB total RAM – 2 GB (hypervisor reserved) – 3 GB (ScaleIO VM) – 8 GB (reserved for ScaleIO RAM cache)) / 2 = 89.5</td>
</tr>
<tr>
<td>Storage performance</td>
<td>60</td>
<td>See Table 16</td>
</tr>
</tbody>
</table>

**Note:** When sizing the node requirement, reserve at least one node for high availability.

Figure 19 shows how to determine the maximum number of MCS-provisioned virtual desktops that a customer-redefined building block configuration can support.

![Figure 19](image)

**Figure 19. Determine the maximum number of virtual desktops: Example 2**

For example, if the customer uses three user-customized building blocks mentioned in Table 21 to build a ScaleIO system, the system should support 120 virtual desktops (2 x 60, with one building block node reserved for high availability).

**Example 3: User-customized building configuration with storage disk addition**

The customized building block configuration in example 2 has sufficient CPU (16 CPU cores support 96 virtual desktops) and memory (192 cores support 89 virtual desktops) resources to support additional virtual desktops. However, in this case, the storage disks reach the maximum IOPS requirement.

With ScaleIO, a user can add storage disks to the existing environment and the workload is rebalanced automatically to all the disks. If three more disks are added to the user-customized building block, 80 virtual desktops can be supported with twelve SAS drives, as shown in 0. This increases the number of virtual desktops supported to 80 without adding CPU or memory resources.
Table 23. User-customized building block with added storage disk: Example 3

<table>
<thead>
<tr>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>192</td>
<td>9 + 3</td>
</tr>
</tbody>
</table>

**Note:** When sizing the node requirement, reserve at least one node for high availability.

Figure 20 shows how to determine the maximum number of MCS-provisioned virtual desktops for a customer-redefined building block configuration with additional disks.

Figure 20. Determine the maximum number of virtual desktops: Example 3

For example, if the customer uses three user-customized building blocks mentioned in 0 to build a ScaleIO system, the system should support 160 virtual desktops (2 x 80, with one building block node reserved for high availability).

**Determining the maximum number of MCS-provisioned PvD virtual desktop building blocks**

With the entire configuration defined for the building block node, we calculate the number of virtual desktops that each component can support to determine the maximum number of desktops the building block node can support. In these examples, PvD (Desktop OS with PvDisk desktop experience) is used to calculate the maximum number of desktops with MCS provisioning.

The following three examples display sizing calculations for determining the maximum number of virtual desktops for a building block:

- Example 4: Baseline building configuration
- Example 5: User-customized building configuration
- Example 6: User-customized building configuration with storage disk addition
Example 4: Baseline building configuration

As shown in Table 24, we defined the baseline building block with seven physical CPU cores, 96 GB memory, and eight 10K 600 GB SAS drives.

Table 24. Baseline building block sizing calculation: Example 4

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>VMs supported</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores: 7</td>
<td>42</td>
<td>7 cores * 6 desktops per core = 42 virtual machines</td>
</tr>
<tr>
<td>RAM: 96 GB</td>
<td>41</td>
<td>(96 GB total RAM – 2 GB (hypervisor reserved) – 3 GB (ScaleIO VM) – 8 GB (reserved for ScaleIO RAM cache)) / 2 = 41.5</td>
</tr>
<tr>
<td>Storage performance</td>
<td>40</td>
<td>See Table 16</td>
</tr>
</tbody>
</table>

Figure 21 shows how to determine the maximum number MCS-provisioned virtual desktops that the baseline building block configuration can support.

Figure 21. Determine the maximum number of virtual desktops: Example 4

For example, if the customer uses three baseline building blocks to build a ScaleIO system, the system should support 80 virtual desktops (2 x 40, with one building block reserved for high availability).

Example 5: User-customized building configuration

The customer can customize a larger building block configuration, as shown in Table 25.

Table 25. User-customized building block node configuration: Example 5

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 CPU cores</td>
<td>192</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 26 shows the sizing calculations for larger building blocks.
Note: When sizing the node requirements, reserve at least one node for high availability.

Figure 22 shows how to determine the maximum number of MCS-provisioned virtual desktops that a customer-redefined building block configuration can support.

For example, if the customer uses the three user-customized building blocks mentioned in Table 25 to build a ScaleIO system, the system should support 120 virtual desktops (2 x 60, with one building block node reserved for high availability).

**Example 6: User-customized building configuration with storage disk addition**

The customized building block configuration in Example 5 has sufficient CPU (16 CPU cores support 96 virtual desktops) and memory (192 cores support 89 virtual desktops) resources to support higher virtual desktops. However, the storage disks reached its maximum IOPS requirement.

With ScaleIO, a user can simply add storage disks to the existing environment and the workload is rebalanced automatically to all the disks. If four more disks are added to the user customized building block, 80 virtual desktops can be supported with twelve SAS drives, as shown in Table 27. This increases the number of virtual desktops supported to 80 without adding CPU or memory resources.

**Table 27. User-customized building block with added storage disk: Example 6**

<table>
<thead>
<tr>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>192</td>
<td>12 + 4</td>
</tr>
</tbody>
</table>

Note: When sizing the node requirement, reserve at least one node for high availability.
Figure 23 shows how to determine the maximum number of MCS-provisioned virtual desktops that a customer redefined building block configuration with additional disks can support.

For example, if the customer uses the three user-customized building blocks shown in Table 27 to build a ScaleIO system, the system should support 160 virtual desktops (2 x 80, with one building block node reserved for high availability).

**Determining the maximum number of MCS- or PVS-provisioned HSD virtual desktop building blocks**

With the entire configuration defined for the building block node, we calculate the number of virtual desktops that each component can support to determine the maximum number of desktops the building block node can support. In these examples, HSD (Server OS) is used to calculate the maximum number of desktops for MCS or PVS provisioning.

The following three examples display sizing calculation for determining the maximum number of virtual desktops for a building block:

- **Example 7**: Baseline building configuration
- **Example 8**: User-customized building configuration
- **Example 9**: User-customized building configuration with storage disk addition

**Example 7: Baseline building configuration**

As shown in Table 28, we defined the baseline building block with 10 physical CPU cores, 48 GB memory, and four 10K 600 GB SAS drives.

**Table 28. Baseline building block sizing calculation: Example 7**

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>VMs supported</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores: 10</td>
<td>40</td>
<td>10 cores * 4 desktops per core = 40 virtual machines</td>
</tr>
<tr>
<td>RAM: 48 GB</td>
<td>52</td>
<td>(48 GB total RAM – 2 GB (hypervisor reserved) – 3 GB (ScaleIO virtual machine) -8 GB (reserved for ScaleIO RAM cache)) / 0.67 = 52.2</td>
</tr>
<tr>
<td>Storage performance</td>
<td>40</td>
<td>See Table 16</td>
</tr>
</tbody>
</table>
Therefore, the final number that the baseline building block node can support is 40—the minimum number for the CPU, memory, and SAS drives, according to the calculation results.

Figure 24 shows how to determine the maximum number of MCS- or PVS-provisioned virtual desktops that the baseline building block configuration can support.

![Figure 24. Determine the maximum number of virtual desktops: Example 7](image)

For example, if the customer uses three baseline building blocks to build a ScaleIO system, the system should support 80 virtual desktops (2 x 40, with one building block reserved for high availability).

**Example 8: User-customized building block configuration**

The customer can customize a larger building block configuration, as shown in Table 29. The final number that this building block node can support is 60—the minimum number for the CPU, memory, and SAS drives, according to the calculation results.

<table>
<thead>
<tr>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>96</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 29. **User-customized building block node configuration: Example 8**

Table 30 shows the calculations for sizing larger building blocks.

<table>
<thead>
<tr>
<th>Physical attribute</th>
<th>VMs supported</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU cores: 24</td>
<td>96</td>
<td>24 cores * 4 desktops per core = 96 virtual machines</td>
</tr>
<tr>
<td>RAM: 96 GB</td>
<td>123</td>
<td>(96 GB total RAM – 2 GB (hypervisor reserved) – 3 GB (ScaleIO virtual machine) - 8 GB (reserved for ScaleIO RAM cache)) / 2 = 123.8</td>
</tr>
<tr>
<td>Storage performance</td>
<td>60</td>
<td>See Table 16</td>
</tr>
</tbody>
</table>

**Note:** When sizing the node requirement, reserve at least one node for high availability.
Figure 25 shows how to determine the maximum number MCS- or PVS-provisioned virtual desktops that a customer redefined building block configuration can support.

**Example 8**: If the customer uses three user-customized building blocks to build a ScaleIO system, the system should support 120 virtual desktops (2 x 60, with one building block node reserved for high availability).

**Example 9**: User-customized building configuration with storage disk addition

The customized building block configuration in Example 8 has sufficient CPU (24 CPU cores support 96 virtual desktops) and memory (96 cores support 123 virtual desktops) resources to support higher virtual desktops. However, the storage disks reached their maximum IOPS requirement.

With ScaleIO, a user can simply add storage disks to the existing environment and the workload rebalances automatically to all the disks. If two more disks are added to the user-customized building block, 80 virtual desktops can be supported with eight SAS drives, as shown in Table 31. This increases the number of virtual desktops supported to 80 without adding CPU or memory resources.

**Table 31. User-customized building block with added storage disk: Example 9**

<table>
<thead>
<tr>
<th>Physical CPU cores</th>
<th>Memory (GB)</th>
<th>10K SAS drives</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>96</td>
<td>6 + 2</td>
</tr>
</tbody>
</table>

**Note**: When sizing the node requirement, reserve at least one node for high availability.
Figure 26 shows how to determine the maximum number of MCS- or PVS-provisioned virtual desktops that a customer redefined building block configuration with additional disks can support.

Figure 26. Determine the maximum number of virtual desktops: Example 9

For example, if the customer uses the three user-customized building blocks shown in Table 31 to build a ScaleIO system, the system should support 160 virtual desktops (2 x 80, with one building block node reserved for high availability).

The VSPEX ScaleIO end-user computing building block defines discrete server node sizes—for example, nodes defined in Table 8, Table 10, and Table 12 support 40 reference virtual desktops with MCS provisioning. The total number of reference virtual desktops from the completed worksheet indicates which reference architecture would be adequate for the customer requirements.

The example in Table 13 shows that the customer requires the ability to support 200 virtual desktops from the pool. Therefore, the six building blocks (5+1, reserving one building block for high availability) defined in Table 12 provide sufficient resources for the current requirement, and still provide room for growth.

Table 32 shows an example of scaling for the baseline configuration nodes.

<table>
<thead>
<tr>
<th>Node number</th>
<th>Maximum number of virtual desktops on baseline configuration</th>
<th>Maximum number of virtual desktops on user-customized building block configuration</th>
<th>Maximum number of virtual desktops on user-customized building block configuration with storage disk addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2+1</td>
<td>80</td>
<td>120</td>
<td>160</td>
</tr>
<tr>
<td>3+1</td>
<td>120</td>
<td>180</td>
<td>240</td>
</tr>
<tr>
<td>4+1</td>
<td>160</td>
<td>240</td>
<td>320</td>
</tr>
<tr>
<td>5+1</td>
<td>200</td>
<td>300</td>
<td>400</td>
</tr>
<tr>
<td>6+1</td>
<td>240</td>
<td>360</td>
<td>480</td>
</tr>
<tr>
<td>7+1</td>
<td>280</td>
<td>420</td>
<td>560</td>
</tr>
</tbody>
</table>

Calculating the building block requirement
Customers have several choices when configuring nodes using building blocks. For example, with the requirement of 250 virtual desktops:

- 7+1 building blocks are needed when using the baseline configuration shown in
• **Example 1: Baseline building** block configuration.
• 5+1 building blocks are needed when using the configuration shown in **Example 2: User-customized building configuration**.
• 4+1 building blocks are needed when using the configuration shown in **Example 2: User-customized building configuration**.

**Note:** The extra building block (+1) in these examples provide for tolerance of a node failure.

In addition to the validated desktop numbers, consider the following factors when deciding which reference architecture to deploy:

• **Concurrency**—The reference workload used to validate this solution assumes that all desktop users will be active at all times. We tested the reference architecture with up to 500 desktops with all desktop starting up at the same time, and all generating workload in parallel. If the customer expects to have 400 users, but only 50 percent of them will be logged on at any given time due to time zone differences or alternate shifts, the 200 active users out of the total 400 users can be supported by the 200-desktop architecture.

• **Heavier desktop workloads**—The reference workload is considered a typical office worker load. However, you might have users with a more active profile.

  If a company has 90 users and, due to custom corporate applications, each user generates 16 IOPS as compared to the 8 IOPS used in the reference workload, you need about 180 reference desktops. In this example, the four building blocks (3+1, reserving one building block for high availability) configuration is insufficient because it has been rated to 120 reference desktops. Consider using the six building blocks (5+1, reserving one building block for high availability) solution.

**Fine-tuning hardware resources**

In most cases, the Customer Sizing Worksheet suggests a reference architecture adequate for the customer’s needs. In other cases, you might want to further customize the hardware resources. A complete description of the system architecture is beyond the scope of this document.

**Storage resources**

In some applications, there is a need to separate certain storage workloads from other workloads. The node configuration for the reference architectures assigned all the virtual desktops to a single resource pool. To achieve workload separation, deploy additional disk drives for each group that needs workload isolation and add them to a dedicated pool.
You should not reduce the number of disks per node to support isolation or to reduce the capability of the pool without additional guidance beyond this Proven Infrastructure Guide. We designed the node configuration to balance many different factors, including high availability, performance, and data protection. Changing the components of the node can have a significant and difficult-to-predict impact on other areas of the system.

**Compute resources**

For the server resources in the solution, it is possible to customize the hardware resources more effectively. To do this, first total the resource requirements for the server components, as shown in Table 33.

**Table 33. Server resource component totals**

<table>
<thead>
<tr>
<th>User types</th>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Number of users</th>
<th>Total CPU resources</th>
<th>Total memory resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heavy users</td>
<td>Resource requirements</td>
<td>2</td>
<td>6</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>Moderate users</td>
<td>Resource requirements</td>
<td>2</td>
<td>4</td>
<td>40</td>
<td>80</td>
</tr>
<tr>
<td>Typical users</td>
<td>Resource requirements</td>
<td>1</td>
<td>2</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The example in Table 33 requires 190 virtual CPUs and 440 GB of memory. The reference architectures assume six desktops per physical processor core for MCS-provisioned PvD or non-PvD desktops and no memory over-provisioning. This translates to 32 physical processor cores and 440 GB of memory.

In contrast, the six building block nodes (5+1, reserve one building block for high availability) defined in Table 12 provide 35 physical cores and 480 GB memory for a desktop operating system with a non-PvD configuration. This means the six-building-block node solution can be implemented effectively with fewer server resources.

**Note:** Keep high availability requirements in mind when customizing the resource pool hardware.

**Summary**

EMC considers these requirements the minimum set of resources to handle the workloads based on the stated definition of a reference virtual desktop. In any customer implementation, the system load varies over time as users interact with the system. If the customer's virtual desktops differ significantly from those defined in the reference definition, you may need to add more resources of that type to the system.
This chapter presents the following topics:

**Overview** .............................................................................................................................................65

**Setting up the network** ...........................................................................................................................66

**Installing and configuring the vSphere hosts** .........................................................................................68

**Installing and configuring the SQL Server database** .............................................................................70

**Deploying VMware vCenter Server** ........................................................................................................71

**Preparing and configuring the storage** ...................................................................................................72

**Installing and configuring XenDesktop Delivery Controllers** ...............................................................86

**Installing and configuring Citrix Provisioning Services (PVS)** .............................................................88
Overview

This chapter describes how to implement the reference architectures of the end-user computing solution. If you already have a VSPEX Proven Infrastructure environment, you can skip the sections for the implementation steps already completed. Otherwise, refer to the *EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide* listed in Chapter 6 for information on configuring the required infrastructure components.

**Note:** This solution requires specific infrastructure services that maybe already available within the existing infrastructure, by a VSPEX Private Cloud solution, or by deploying them as dedicated resources as part of this solution.

Table 34 lists the main stages in the solution implementation process, with links to the relevant sections in the chapter.

**Table 34. Implementation process overview**

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configure the switches and networks and connect to the customer network.</td>
<td>Setting up the network</td>
</tr>
<tr>
<td>2</td>
<td>Install and configure the vSphere hosts and infrastructure servers.</td>
<td>Installing and configuring the vSphere hosts</td>
</tr>
<tr>
<td>3</td>
<td>Set up SQL Server (used by vCenter and Citrix XenDesktop).</td>
<td>Installing and configuring the SQL Server database</td>
</tr>
<tr>
<td>4</td>
<td>Configure the vCenter Server.</td>
<td>Deploying VMware vCenter Server</td>
</tr>
<tr>
<td>5</td>
<td>Configure the ScaleIO environment.</td>
<td>Preparing and configuring the storage</td>
</tr>
<tr>
<td>6</td>
<td>Set up XenDesktop Controller.</td>
<td>Installing and configuring XenDesktop Delivery Controllers</td>
</tr>
<tr>
<td>7</td>
<td>Provision virtual desktops.</td>
<td>Provisioning the virtual desktops</td>
</tr>
</tbody>
</table>
Setting up the network

This section describes the requirements for preparing the network infrastructure required to support this solution. Table 35 summarizes the tasks to be completed, with references for further information.

Table 35. Tasks for switch and network configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the infrastructure network</td>
<td>Configure the vSphere host infrastructure networking.</td>
<td>Configuring the infrastructure network</td>
</tr>
<tr>
<td>Configure the VLANs</td>
<td>Configure private and public VLANs as required.</td>
<td>Configuring the VLANs Vendor’s switch configuration guide</td>
</tr>
<tr>
<td>Complete the network cabling</td>
<td>Connect the switch interconnect ports, and vSphere server ports.</td>
<td>Completing the network cabling</td>
</tr>
</tbody>
</table>

The infrastructure network requires redundant network links for each vSphere host, switch interconnect ports, and switch uplink ports. This configuration provides both redundancy and the ability to expand network bandwidth.

This configuration is required regardless of whether the network infrastructure for the solution already exists or is being deployed with other components of the solution.

Figure 27 shows a sample redundant Ethernet infrastructure for this solution. It illustrates the use of redundant switches and links to ensure that no single point of failure exists in network connectivity.
EMC recommends using at least three networks: one management network, and two independent data networks. In this solution we use two 1 GbE networks for management (for redundancy) and two independent 10 GbE networks for data.

**Configuring the VLANs**

Ensure that there are adequate switch ports for the storage array and vSphere hosts. EMC recommends that you configure the vSphere hosts with a minimum of three VLANs:

- **Client access network**: Virtual machine networking and Common Internet File System (CIFS) traffic (customer-facing networks, which can be separated if needed)
- **Storage network**: ScaleIO data networking (private network)
- **Management network**: vSphere management and VMware vMotion (private network)

**Completing the network cabling**

Ensure that all solution servers, switch interconnects, and switch uplinks have redundant connections and are plugged into separate switching infrastructures. Ensure that there is a complete connection to the existing customer network.

**Note**: At this point, the new equipment is connected to the existing customer network.
Installing and configuring the vSphere hosts

This section provides information for installing and configuring the vSphere hosts and infrastructure servers required to support the architecture. Table 36 describes the tasks required to install the server.

Table 36. Tasks for server installation

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install vSphere</td>
<td>Install the vSphere hypervisor on the physical servers deployed for the solution.</td>
<td>vSphere Installation and Setup Guide</td>
</tr>
<tr>
<td>Configure vSphere networking</td>
<td>Configure vSphere networking including network interface card (NIC) trunking, VMkernel ports, and virtual machine port groups.</td>
<td>vSphere Networking</td>
</tr>
<tr>
<td>Connect VMware datastores</td>
<td>Connect the VMware datastores to the vSphere hosts deployed for the solution.</td>
<td>vSphere Storage Guide</td>
</tr>
</tbody>
</table>

Installing vSphere

When powering up the servers being used for vSphere, confirm or enable the hardware-assisted central processing unit (CPU) virtualization setting and the hardware-assisted memory management unit (MMU) virtualization setting in the server BIOS.

Start up the vSphere installation media and install the hypervisor on each of the servers. vSphere hostnames, IP addresses, and a root password are required for installation. The completed Customer Configuration Worksheet provides the appropriate values.

Configuring vSphere networking

The vSphere Networking guide describes vSphere networking configuration, including load balancing, link aggregation, and failover options. Select the appropriate load-balancing option based on what is supported by the network infrastructure. Refer to the list of documents in Chapter 6 for more information.

Network interface cards

During the installation of vSphere, a standard virtual switch (vSwitch) is created. By default, vSphere selects only one physical NIC as a vSwitch uplink. To maintain redundancy and bandwidth requirements, configure an additional NIC, either by using the vSphere console or by connecting to the vSphere host from the vSphere client.

Each vSphere server should have multiple interface cards for each virtual network to ensure redundancy and provide for network load balancing, link aggregation, and network adapter failover.
VMkernel ports
Create VMkernel ports as required, based on the infrastructure configuration:

- VMkernel port for vMotion
- Virtual desktop port groups (used by the virtual desktops to communicate on the network)

The *vSphere Networking* guide describes how to configure these settings. Refer to the list of documents in *Chapter 6* for more information.

**Note**: The vSwitch and VMkernel ports created here are used for user management and for virtual desktop communication. The ScaleIO data networking will be created by the ScaleIO installation wizard.

---

**Connecting the VMware datastores**

Connect the datastores configured in *Preparing and configuring the storage* to the appropriate vSphere servers. These include the datastores configured for the following:

- Virtual desktop storage
- Infrastructure virtual machine storage (if required)
- SQL Server storage (if required)

The *vSphere Storage Guide* provides instructions on how to connect the VMware datastores to the vSphere host.
Installing and configuring the SQL Server database

Table 37 describes the tasks required for setting up and configuring a Microsoft SQL Server database. When the tasks are complete, the SQL Server is set up on a virtual machine, with all databases required by vCenter, Citrix XenDesktop, and Citrix PVS configured for use.

Table 37. Tasks for SQL Server database setup

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a virtual machine for Microsoft SQL</td>
<td>Create a virtual machine to host SQL Server on one of the vSphere servers designated for infrastructure virtual machines, and use the datastore designated for the shared infrastructure. Verify that the virtual server meets the hardware and software requirements.</td>
<td><em>vSphere Virtual Machine Administration</em></td>
</tr>
<tr>
<td>Install Microsoft SQL Server</td>
<td>Install Microsoft SQL Server on the virtual machine.</td>
<td><em>SQL Server Installation (SQL Server 2012)</em></td>
</tr>
<tr>
<td>Configure the database for VMware vCenter</td>
<td>Create the database required for vCenter Server on the appropriate datastore.</td>
<td><em>Preparing vCenter Server Databases</em></td>
</tr>
<tr>
<td>Configure VMware vCenter database</td>
<td>Configure the database server with appropriate permissions for vCenter.</td>
<td><em>Preparing vCenter Server Databases</em></td>
</tr>
</tbody>
</table>

*Note:* We recommend that you put the OS volume for the SQL Server virtual machine into the VSPEX private cloud pool. The recommended values for CPU and memory are 2 vCPU and 6 GB, respectively.
Deploying VMware vCenter Server

Table 38 describes the tasks required to configure VMware vCenter Server for the solution.

**Note:** EMC recommends that you put the OS volume for the vCenter Server virtual machine into the VSPEX private cloud pool. The recommended values for CPU and memory are 4 vCPU and 8 GB, respectively.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create the vCenter host virtual machine</td>
<td>Create a virtual machine for vCenter Server.</td>
<td>vSphere Virtual Machine Administration</td>
</tr>
<tr>
<td>Install the vCenter guest OS</td>
<td>Install Windows Server 2012 Standard Edition on the vCenter host virtual machine.</td>
<td>VMware vSphere Documentation</td>
</tr>
<tr>
<td>Update the virtual machine</td>
<td>Install VMware Tools, enable hardware acceleration, and allow remote console access.</td>
<td>vSphere Virtual Machine Administration</td>
</tr>
<tr>
<td>Create vCenter Open Database Connectivity (ODBC) connections</td>
<td>Create the 64-bit vCenter and 32-bit vCenter Update Manager ODBC connections.</td>
<td>vSphere Installation and Setup, Installing and Administering VMware vSphere Update Manager</td>
</tr>
<tr>
<td>Install vCenter Server</td>
<td>Install the vCenter Server software.</td>
<td>vSphere Installation and Setup</td>
</tr>
<tr>
<td>Install Web Client</td>
<td>Install the Web Client software.</td>
<td></td>
</tr>
<tr>
<td>Install PowerCLI</td>
<td>Install the PowerCLI software on the vCenter Server.</td>
<td></td>
</tr>
<tr>
<td>Create a virtual data center</td>
<td>Create a virtual data center.</td>
<td>vCenter Server and Host Management</td>
</tr>
<tr>
<td>Apply vSphere license keys</td>
<td>Type the vSphere license keys in the vCenter licensing menu.</td>
<td>vSphere Installation and Setup</td>
</tr>
<tr>
<td>Add vSphere Hosts</td>
<td>Connect the vCenter server to the vSphere hosts.</td>
<td>vCenter Server and Host Management</td>
</tr>
<tr>
<td>Configure vSphere clustering</td>
<td>Create a vSphere cluster and move the vSphere hosts into it.</td>
<td>vSphere Resource Management</td>
</tr>
</tbody>
</table>
Preparing and configuring the storage

Table 39 describes the tasks required to configure the storage for the solution.

Table 39. Tasks for storage configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare the ScaleIO environment</td>
<td>Configure each ESXi host as required.</td>
<td><em>vSphere Networking</em></td>
</tr>
<tr>
<td>Register the ScaleIO plug-in</td>
<td>Register the ScaleIO plug-in in vSphere Web Client.</td>
<td><em>EMC ScaleIO User Guide</em></td>
</tr>
<tr>
<td>Upload the OVA template</td>
<td>Upload the OVA template to the ESXi host.</td>
<td></td>
</tr>
<tr>
<td>Access the plug-in</td>
<td>Use vSphere Web Client to access the ScaleIO plug-in.</td>
<td></td>
</tr>
<tr>
<td>Install SDC on ESXi</td>
<td>Install SDC directly on the ESXi server.</td>
<td></td>
</tr>
<tr>
<td>Deploy ScaleIO</td>
<td>Deploy the ScaleIO system from vSphere Web Client.</td>
<td></td>
</tr>
<tr>
<td>Create volumes</td>
<td>Create volumes with required capacity from the ScaleIO system and map the volumes to the ESXi hosts.</td>
<td></td>
</tr>
<tr>
<td>Create datastores</td>
<td>Scan the ScaleIO LUN from ESXi hosts and create datastores.</td>
<td><em>vSphere Storage Guide</em></td>
</tr>
<tr>
<td>Install the GUI</td>
<td>Install the ScaleIO GUI to manage the system.</td>
<td><em>EMC ScaleIO User Guide</em></td>
</tr>
</tbody>
</table>

You can deploy ScaleIO components in two ways in the VMware environment:

- The ScaleIO components—Meta Data Manager (MDM), ScaleIO Data Server (SDS), and ScaleIO Data Client (SDC)—and an iSCSI target, are installed on dedicated ScaleIO virtual machines (SVMs).

  The SDS adds the ESXi physical devices to the ScaleIO to be used for storage, thus enabling the creation of volumes. Using iSCSI targets, the volumes are exposed to ESXi, via an iSCSI adapter.

  ScaleIO volumes must be mapped both to the SDC and to iSCSI initiators. This ensures that only authorized ESXi hosts can see the targets. Enabling multipathing, either automatically or manually, enhances reliability. The ScaleIO vSphere VMware deployment wizard enables you to complete these activities in a simple, efficient manner, over all the machines in a vCenter.

- The MDM and SDS ScaleIO components are installed on a dedicated SVM.

  The SDC is installed directly on the ESXi server. This eliminates the need for iSCSI. This is the recommended method of deployment, and this option can be implemented on ESXi version 5.5 or higher.
Before deploying ScaleIO, complete the following prerequisite steps:

1. Configure the management network and Virtual Machine Port Group on all ESXi hosts that are part of the ScaleIO system.
2. Ensure that the devices you are adding to SDS are free of partitions.
3. Create a datastore from one of the local devices for all the ESXi hosts. This datastore is needed when deploying SVMs.

Registering the ScaleIO plug-in

The ScaleIO plug-in is registered on the vCenter Server to enable vSphere Web Client to install and manage the ScaleIO system. The plug-in is provided as a Zip file that can be downloaded by the vSphere Web Client servers in your environment. You can download the Zip file directly from the EMC Online Support site or from a file server if the web servers do not have internet access. Perform the following steps to register the ScaleIO plug-in:

1. If you are uploading the Zip file to a HTTP server, complete the following steps:
   a. On the computer where vSphere Web Client is installed, locate the webclient.properties file.
      - Windows 2003:
        %ALLUSERPROFILE%Application Data\VMware\vSphere Web Client
      - Windows 2008:
        %ALLUSERSPROFILE%\VMware\vSphere Web Client
      - Windows 2012:
        C:\ProgramData\VMware\vSphere Web Client
      - Linux:
        /var/lib/vmware/vsphere-client
   b. Add the following line to the file: allowHttp=true
   c. Restart the VMware vSphere Web Client service.
2. Using PowerCLI for VMware set it to Run as administrator, run Set-ExecutionPolicy RemoteSigned
3. Close PowerCLI, reopen it, and set it to Run as administrator.
4. Extract the following file: EMC-ScaleIO-vSphere-plugin-installer-1.32.XXX.X.zip
5. Change to the directory where the extracted files in step 4 are placed using the cd command. Run the ScaleIOPPluginSetup-1.32.XXX.X.ps1 script in interactive mode, and provide the required information.
   a. Type the vCenter name or IP address, user name, and password.
Chapter 4: Solution Implementation

b. Select **Option 1** to register the ScaleIO plug-in.

c. Select **Standard** for Select Registration Mode.

**Note:** You can use the **Advanced** option from Select Registration Mode to install the plug-in using a ScaleIO Gateway from a previous installation or using your own web service. In either case, you must place this version’s plugin.zip file (EMC-ScaleIO-vSphere-web-plugin-1.32.XXX.X.zip) in your resources folder before running the installation. To use a previous ScaleIO Gateway, the resource folder is ScaleIO Gateway installation folder\webapps\root\resources.

6. Log out and log back in to vSphere Web Client to load the ScaleIO plug-in.

### Uploading the OVA template

ScaleIO uses a PowerShell script to upload the open virtualization archive (OVA) template to vCenter Server. To upload the OVA template, perform the following steps:

1. Save **ScaleIOVM_1.32.xxx.0.ova** on the local computer.
2. Run **PowerCLI** and navigate to the location of the extracted file, **EMC-ScaleIO-vSphere-web-plugin-package-1.32.XXX.X.zip**.
3. Run the **ScaleIOPluginSetup-1.32.XXX.X.ps1** script.

   a. Type the vCenter name or IP address, user name, and password.
   b. Select **Option 3** to create the SVM template.

The CLI wizard requires the following parameters:

- Data center name
- Path to the OVA template
- Datastore names

For faster deployment in large-scale environments, you can upload the OVA template to as many as eight datastores. To do so, type the datastore names, and when you are finished, leave the next line blank.

The following example shows how to enter two datastores:

```
datastores[0]: datastore1
datastores[1]: datastore2
datastores[2]:
```

The upload procedure can take several minutes. When the upload is complete, the following message appears: **Your new EMC ScaleIO Templates are ready to use.**
After you register the ScaleIO plug-in on the vCenter Server, the EMC ScaleIO icon appears in the vSphere Web Client home tab, as shown in Figure 28. Click the icon to view the EMC ScaleIO screen.

![Figure 28. EMC ScaleIO plug-in in vSphere Web Client](image)

ScaleIO 1.32 provides the option to install SDC directly to the ESXi server. This option is available for ESXi version 5.5 and above. Complete the following steps to install SDC on the ESXi host:

1. From the **EMC ScaleIO screen**, under Basic tasks, click **Install SDC on ESX**.
2. Select the ESX (ESXi) hosts to be installed on SDC.
3. Enter the root password, as shown in Figure 29.
Chapter 4: Solution Implementation

Deploying ScaleIO

ScaleIO provides a wizard to deploy the application using vSphere Web Client. Complete the following steps to deploy ScaleIO:

1. From the **EMC ScaleIO screen**, under Basic tasks, click **Deploy ScaleIO Environment**.
2. Review and approve the license terms and click **Next**.

   **Note:** The deployment wizard assumes that you are using the provided ScaleIO OVA template to create the ScaleIO virtual machines.

3. In the **Select Installation** screen, select **Create a new ScaleIO system** and click **Next**.
4. In the **Create New System** screen, enter the following information, and then click **Next**:
   - **System Name**—Enter a unique name for this system.
   - **Admin Password**—Enter a password for the ScaleIO admin user. The password must meet the following criteria:
     - Be between 6 and 31 characters
Chapter 4: Solution Implementation

- Include at least three of the following groups: [a-z], [A-Z], [0-9], special characters (!@#$...)

- Include no white spaces.

5. In the Add ESXi Hosts to Cluster screen, shown in Figure 30, select the vCenter on which to deploy the ScaleIO system. Select the ESXi hosts to add to the ScaleIO system and click Next.

![Figure 30. Add the ESXi host to the cluster](image)

**Note:** You must select a minimum of three ESXi hosts to configure ScaleIO.

6. In the Select management components screen, shown in Figure 31, match the ScaleIO management components to the ESXi hosts. Click Next.

![Figure 31. Select management components](image)

7. In the Configure call home screen, select Configure Call Home, enter the email settings, and select a minimum severity level for call home events.

8. Enter the details to configure the DNS servers. Click Next.

9. In the Configure Protection Domains screen, enter the Protection Domain (PD) name and RAM read cache size per SDS. Click Add to create a PD.
10. Click **Next**.

A default Storage Pool (SP) is automatically created under the PD in the **Configure Storage Pools** screen, as shown in Figure 32. You can use this default SP or create a new SP by clicking **Add**.

**Create a new Storage Pool in ScaleIO system (optional):**

- **Storage Pool name:**
- **Assign the new Storage Pool to the Protection Domain:**
  - Domain1

**New and existing Protection Domains and Storage Pools:**

11. Click **Next**.

The **Create Fault Sets** screen appears. Optionally, you can create the fault sets first and then click **Next**.

12. In the **Add SDSs** screen, as shown in Figure 33, select one of the following values for each ESXi host/SVM and then click **Next**:

- If the SVM is an SDS, select a PD (required) and fault set (optional).
- If the SDS has flash devices, select **Optimize for Flash** to optimize ScaleIO efficiency for the flash devices.
13. **Under Assign ESX host devices to ScaleIO SDS components**, as shown in Figure 34, complete the following steps:

   a. Click **Select devices** and select storage devices to add a single SDS.

   b. Click **Replicate selection** and select devices for other SDSs by replicating the selections made in the **Select devices** screen. This is useful if the ESXi hosts have identical attached devices.

   c. Under the **Information** tab, select an ESXi host under the cluster and click **Select devices**.

---

**Figure 33. Add SDS**

**Figure 34. Assign ESXi host devices to ScaleIO SDS components**
14. Select Add Device and select a storage pool, as shown in Figure 35.

![Figure 35. Select devices for SDS](image)

Refer to Chapter 3 to calculate the number of disks for each ESXi host to add to the ScaleIO system.

In almost all cases, raw device mapping (RDM) is the preferred method to add physical devices. Use the Virtual Machine Disk (VMDK) method only if the:

- physical device does not support RDM.
- device already has a datastore and is not being fully utilized.

The excess capacity that is not being used will be added to the ScaleIO device.

**Note:** In this case, one device contains a datastore from which to deploy the SVM. Use VMDK for this device only and use RDM for all the other devices.

15. Repeat Step 14 to add devices for each ESXi host. Click **Next**.

16. In the Add SDCs screen, as shown in Figure 36, select one of the following values for each ESXi host/SVM and then click **Next**:

   a. If installing SDC to the SVM, set the **SDC mode** to **SVM**. If installing SDC directly to the ESX server, set the **SDC mode** to **ESX** and specify the ESXi server root password.

   b. Select whether to enable or disable the LUN comparison for ESXi hosts.

   **Note:** Before selecting this setting, consult your environment administrator.
17. In the **Configure ScaleIO Gateway** screen, as shown in Figure 37, set the following values and then click **Next**:

- ESXi host for the ScaleIO gateway virtual machine
- Admin password for the gateway
- Lightweight Installation Agent (LIA) password

18. In the **Select OVA Template** screen, shown in Figure 38, complete the following steps and then click **Next**:
a. Select the template to use to create the SVMs. **EMC ScaleIO SVM Template** is the default template. If you uploaded a template to multiple datastores, select them all for faster deployment.

b. Enter a new password for all SVMs that you will create.

**Figure 38. Select OVA template**

19. In the **Configure networks** screen, shown in Figure 39, select either a single network or separate networks for management and data transfer.

**Figure 39. Configure networks**

**Note:** The selected network must be able to communicate with all of the system nodes. In some cases, while the wizard does verify that the network names match, this does not guarantee communication, as the VLAN IDs may have been manually altered.

EMC recommends using separate networks for security and increased efficiency. In this solution, we use two data networks for high availability.
The management network, used to connect and manage the SVMs, is usually connected to the client management network, a 1 GbE network. The data network is internal, enabling communication between the ScaleIO components, and is generally a 10 GbE network.

20. Select a management network label and then configure the data network by clicking **Create new network**, as shown in Figure 40.

![Create New Data Network](image)

21. In the **Create New Data Network** screen, specify the following information:
   - **Network name**—Type the name of the VMware network.
   - **VMkernel name**—Type the name of the VMkernel.
   - **VLAN ID**—Type the network ID.
   - For each listed ESXi host, select a **Data NIC**, a **VMkernel IP**, and a **VMkernel Subnet Mask**.

22. Click **OK**.

The data network is created. The wizard automatically configures the following information for the data network:
   - vSwitch
   - VMkernel port
   - Virtual Machine Port Group
   - iSCSI Software adapter
   - VMkernel Port Binding

23. Repeat step 21 and step 22 to configure the second data network. Click **Next**.

**Note:** For best results, use the plug-in to create the data networks, as shown in the preceding steps, rather than creating them manually.
24. In the Configure SVM network screen, enter the IP address, subnet mask and default gateway for each SVM. You can select the datastore to host the SVM yourself, or set to it to automatic so that the system selects a datastore. Click Next.

**Note:** Because you are configuring two data networks, you need three IP addresses for each SVM: one for management and the other two for data transfer. You must separate these networks in three different subnets.

25. In the Review Summary screen, review the configuration and click Finish to begin deployment.

26. Click Refresh in the browser to view the deployment progress on the ScaleIO screen. During the deployment process you can view progress, stop the deployment, and view the logs.

27. Click Finish when the deployment is complete.

### Creating volumes

This section describes how to use the plug-in to create volumes in the VMware environment. You can map volumes to SDCs in the same step. To create volumes from devices in a storage pool, perform the following steps:

1. From the Storage Pools screen, click Actions > Create volume, as shown in Figure 41.

   ![Storage Pools Screen](image)

   **Figure 41. **Create volume

2. In the Create Volume dialog box, shown in Figure 42, type the following information:

   - **Volume name**—Type a name for the new volume.
   - **Number of volumes to create**
   - **Volume size (GB)**—Type the size of the volume.

   **Note:** Use the maximum capacity of the storage pool when using the volume for provisioning the full-cloned virtual desktops.

   - **Volume provisioning**—Select Thick.
- **Use RAM Read Cache**—Accept the default setting.
- **Obfuscation**—Accept the default setting.

![Create Volume](image)

3. Complete the following steps to map the volume to SDCs:
   a. Select **Map volume to SDCs/ESXi hosts**.
   b. In the **Select SDCs/ESXi hosts** area, select the clusters or SDCs to which this volume should be mapped.
   c. Select **Manually configure LUN identifier** and specify the LUN identify number to manually configure the LUN identifier.
   d. Type the identifier ID and click **OK**.
   e. Type the password for the ScaleIO admin user.

4.Repeat the preceding steps to create the required number of volumes.
Rescan the datastore to discover the ScaleIO LUNs on the appropriate ESXi hosts. Create datastores for these LUNs.

The *vSphere Storage Guide* provides instructions on how to create the VMware datastores on the ESXi host.

**Installing the GUI**

This section describes how to install the ScaleIO GUI. You can install the GUI on a Windows or Linux workstation.

To install the GUI, run one of the following commands:

- **For Windows:** `EMC-ScaleIO-gui-1.32.0.xxx.msi`
- **For RHEL:** `rpm -U scaleio-gui-1.32.0-xxx.noarch.rpm`
- **For Debian:** `sudo dpkg -i scaleio-gui-1.32.0-xxx.deb`

**Installing and configuring XenDesktop Delivery Controllers**

This section describes how to set up and configure XenDesktop Delivery Controllers.

For a new XenDesktop installation, Citrix recommends that you complete the tasks in Table 40 in the order shown.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
</table>
| Creating virtual machines for XenDesktop Delivery Controllers | Create two virtual machines in the vSphere client. These virtual machines are used as XenDesktop Delivery Controllers. | *vSphere Virtual Machine Administration*
| Installing the guest operating system for the XenDesktop Delivery Controllers | Install the Windows Server 2012 R2 or Windows Server 2012 guest operating system on the virtual machines. | |
| Installing the XenDesktop server-side components | Install the required XenDesktop server components on the first Delivery Controller. | *Citrix website*
| Installing Citrix Studio | Install Citrix Studio to manage XenDesktop deployment remotely. | |
| Configuring a site | Configure a site in Citrix Studio. | |
| Adding a second XenDesktop Delivery Controller | Install an additional Delivery Controller for high availability. | |
| Preparing a master virtual machine | Create a master virtual machine as the base image for the virtual desktops. | |
| Provisioning the virtual desktops | Provision the virtual desktops using MCS. | |
Installing the server-side components of XenDesktop

Install the following XenDesktop server-side components on the first Delivery Controller:

- **Delivery Controller**: Distributes applications and desktops, manages user access, and optimizes connections
- **Citrix Studio**: Creates, configures and manages infrastructure components, applications, and desktops
- **Citrix Director**: Monitors performance and troubleshoots problems
- **License server**: Manages product licenses
- **Citrix StoreFront**: Provides authentication and resource delivery services for Citrix Receiver

**Note**: Citrix supports the installation of XenDesktop components only through the procedures described in Citrix documentation.

Configuring a site

Start Citrix Studio and configure a site as follows:

1. License the site and specify which edition of XenDesktop to use.
2. Set up the site database using a designated login credential for SQL Server.
3. Provide information about your virtual infrastructure, including the vCenter SDK path that the controller uses to establish a connection to the VMware infrastructure.

Adding a second controller

After you have configured a site, you can add a second Delivery Controller to provide high availability. The XenDesktop server-side components required for the second controller are:

- Delivery Controller
- Citrix Studio
- Citrix Director
- Citrix StoreFront

Do not install the license-server component on the second controller because it is centrally managed on the first controller.

Installing Citrix Studio

Install Citrix Studio on the appropriate administrator consoles to manage your XenDesktop deployment remotely.

Preparing the master virtual machine

Complete the following steps to prepare the master virtual machine:

1. Install the Windows 7 guest OS.
2. Install the required integration tools, such as VMware Tools.
3. Optimize the OS settings to prevent unnecessary background services from generating inessential I/O operations that adversely affect the overall performance of the storage array. Refer to the white paper [Citrix Windows 7 Optimization Guide for Desktop Virtualization](#) for details.
4. Install the Virtual Delivery Agent.
5. Install the third-party tools or applications, such as Microsoft Office, relevant to your environment.

Provisioning the virtual desktops

Complete the following steps in Citrix Studio to deploy MCS-based virtual desktops:
1. Create a machine catalog using the master virtual machine as the base image.
   MCS allows you to create a machine catalog that contains various types of desktops. We tested the following desktop types for this solution:
   - **Windows Desktop OS:**
     - **Random:** Users connect to a new (random) desktop each time they log on.
     - **Personal vDisk:** Users connect to the same (static) desktop each time they log on. Changes are saved on a separate Personal vDisk.
   - **Windows Server OS:** Provides hosted shared desktops deploying standardized machines.
2. Add the machines created in the catalog to a delivery group so that the virtual desktops are available to the end users.

Installing and configuring Citrix Provisioning Services (PVS)

This section provides information about how to set up and configure Citrix PVS. For a new installation of PVS, Citrix recommends that you complete the tasks in Table 41 in the order shown.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating virtual machines for PVS servers</td>
<td>Create two virtual machines in the vSphere Client. These virtual machines are used as PVS servers.</td>
<td>vSphere Virtual Machine Administration</td>
</tr>
<tr>
<td>Installing the guest operating system for the PVS servers.</td>
<td>Install the Windows Server 2012 R2 or Windows Server 2012 guest OS for the PVS servers.</td>
<td></td>
</tr>
<tr>
<td>Installing the PVS server-side components</td>
<td>Install the PVS server components and console on the PVS server.</td>
<td>Citrix website</td>
</tr>
<tr>
<td>Configuring a PVS server farm</td>
<td>Run the Provisioning Services Configuration Wizard to create a PVS server farm.</td>
<td></td>
</tr>
<tr>
<td>Adding a second PVS server</td>
<td>Install the PVS server components and console on the second server and join it to the existing server farm.</td>
<td></td>
</tr>
<tr>
<td>Creating a PVS store</td>
<td>Specify the store path where the vDisks will reside.</td>
<td></td>
</tr>
</tbody>
</table>
Task | Description | Reference
--- | --- | ---
Configuring inbound communication | Adjust the total number of threads to be used to communicate with each virtual desktop. |  
Configuring a bootstrap file | Update the bootstrap image to use both PVS servers to provide streaming services. |  
Configuring boot options 66 and 67 on the DHCP server | Specify the TFTP server IP and the name of the bootstrap image used for the Preboot eXecution Environment (PXE) boot. |  
Preparing a master virtual machine | Create a master virtual machine as the base image for the virtual desktops. |  
Provisioning the virtual desktops | Provision the virtual desktops using PVS. |  

### Configuring a PVS server farm

After installing the PVS server components on the PVS server, start the Provisioning Services Configuration Wizard and configure a new server farm using the following steps:

1. Specify the DHCP service to be run on another computer.
2. Specify the PXE service to be run on this computer.
3. Select **Create farm** to create a new PVS server farm using a designated SQL Server database instance. When creating a new server farm, you need to create a site. Provide an appropriate name for the new site and target device collection.
4. Select the license server that is running on the XenDesktop controller.
5. Select **Use the Provisioning Services TFTP service**.

### Adding a second PVS server

After configuring the PVS server farm, you can add a second PVS server to provide high availability. Install the PVS server components and console on the second PVS server and run the Provisioning Services Configuration Wizard to join the second server to the existing server farm.

### Creating a PVS store

A PVS store is a logical container for vDisks. PVS supports the use of a CIFS share as the storage target of a PVS store. Complete the following steps to create a PVS store:

1. Set the default store path to the local PVS drive or universal naming convention (UNC) path of a CIFS share that is hosted on the shared storage, such as EMC VNX storage.
2. Right-click a store in the **Provisioning Services** console.
3. Select **Properties** and **Validate** to confirm that all PVS servers in the server farm can access the CIFS share.
Chapter 4: Solution Implementation

**Configuring inbound communication**

Each PVS server maintains a range of User Datagram Protocol (UDP) ports to manage all inbound communications from virtual desktops. Ideally, there should be one thread dedicated to each desktop session. The total number of threads supported by a PVS server is calculated as:

\[
\text{Total threads} = (\text{Number of UDP ports} \times \text{Threads per port} \times \text{Number of network adapters})
\]

Adjust the thread count to match the number of deployed virtual desktops.

**Configuring a bootstrap file**

To update the bootstrap file required for the virtual desktops to PXE boot, complete the following steps:

1. In the Provisioning Services console, select **Farm > Sites > Site-name > Servers**.
2. Right-click a server and select **Configure Bootstrap**. The **Configure Bootstrap** dialog box appears, as shown in Figure 43.

![Configure Bootstrap dialog box](image)

3. Update the bootstrap image to reflect the IP addresses used for all PVS servers that provide streaming services in a round-robin fashion. Select **Read Servers from Database** to obtain a list of PVS servers automatically or select **Add** to manually add the server information. If the PVS server information is incorrect (IP address, port, subnet mask, or default gateway), highlight the PVS server entry and select **Edit** to make the change.

4. After modifying the configuration, click **OK** to update the ARDBP32.BIN bootstrap file, which is located at **C:\ProgramData\Citrix\Provisioning Services\Tftpboot**.

5. Navigate to the folder and examine the timestamp of the bootstrap file to ensure that it is updated on the intended PVS server.
To PXE boot the virtual desktops successfully from the bootstrap image supplied by the PVS servers, set the boot options 66 and 67 on the Microsoft DHCP server.

Complete the following steps to configure the boot options on the DHCP server:

1. From the DHCP management interface of the DHCP server, right-click **Scope Options** and select **Configure Options**.
2. Select **066 Boot Server Host Name**. In **String value**, type the IP address of the PVS server configured as the TFTP server.
3. Select **067 Bootfile Name**. In **String value**, type **ARDBP32.BIN**. The ARDBP32.BIN bootstrap image is loaded on a virtual desktop before the vDisk image is streamed from the PVS servers.

Complete the following steps to prepare the master virtual machine:

1. Install the Windows 7 guest OS.
2. Install the required integration tools, such as VMware Tools.
3. Optimize the OS settings to prevent unnecessary background services from generating inessential I/O operations that adversely affect the overall performance of the storage array. Refer to the white paper *Citrix Windows 7 Optimization Guide for Desktop Virtualization* for details.
4. Install the Virtual Delivery Agent.
5. Install the third-party tools or applications, such as Microsoft Office, relevant to your environment.
6. Install the PVS target device software on the master virtual machine.
7. Modify the BIOS of the master virtual machine so that the network adapter is at the top of the boot order to ensure PXE boot of the PVS bootstrap image.

Complete the following steps to deploy the PVS-based virtual desktops:

1. Run the PVS imaging wizard to clone the master image onto a vDisk.
2. When the cloning is complete, shut down the master virtual machine and modify the following vDisk properties:
   - **Access mode**: Standard Image
   - **Cache type**: Cache on device hard drive
3. Prepare a virtual machine template to be used by the XenDesktop Setup Wizard in the next step.
4. Run the XenDesktop Setup Wizard in the PVS console to create a machine catalog that contains the specified number of virtual desktops.
5. Add the virtual desktops created in the catalog to a delivery group so that the virtual desktops are available to the end users.
This chapter presents the following topics:

- **Overview** ................................................................. 93
- **Verifying the installation with the post-installation checklist** ................................................................. 94
- **Deploying and testing a single virtual desktop** ................................................................. 94
- **Verifying component redundancy** ................................................................. 94
### Overview

After you configure the solution, complete the tasks in Table 42 to verify the configuration and functionality of the solution’s components and to ensure that the configuration supports the core availability requirements.

**Table 42. Tasks for testing the installation**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify the installation using the post installation checklist</td>
<td>Verify that adequate virtual ports exist on each vSphere host virtual switch.</td>
<td>• <em>vSphere Networking</em></td>
</tr>
<tr>
<td></td>
<td>Verify that each vSphere host has access to the required datastores and VLANs.</td>
<td>• <em>vSphere Storage Guide</em></td>
</tr>
<tr>
<td></td>
<td>Verify that the vMotion interfaces are configured correctly on all vSphere hosts.</td>
<td>• <em>vSphere Networking</em></td>
</tr>
<tr>
<td>Deploy and test a single virtual desktop</td>
<td>Deploy a single virtual machine from the vSphere interface by using the customization specification.</td>
<td>• <em>vCenter Server and Host Management</em></td>
</tr>
<tr>
<td></td>
<td>• <em>vSphere Virtual Machine Management</em></td>
<td></td>
</tr>
<tr>
<td>Verify the redundancy of the solution components</td>
<td>Verify the data protection of ScaleIO system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Verify the redundancy of switches.</td>
<td>• Vendor documentation</td>
</tr>
<tr>
<td></td>
<td>Verify the virtual machine vMotion.</td>
<td>• <em>vCenter Server and Host Management</em></td>
</tr>
<tr>
<td>Provision the remaining virtual desktops</td>
<td>Provision desktops using MCS or PVS.</td>
<td>• <em>Installing and configuring XenDesktop Delivery Controllers</em></td>
</tr>
<tr>
<td></td>
<td>• <em>Installing and configuring Citrix Provisioning Services (PVS)</em></td>
<td></td>
</tr>
</tbody>
</table>
Verifying the installation with the post-installation checklist

The following configuration tasks are critical to the functionality of the solution. Verify the configuration before deploying the system in production. On each vSphere server used as part of this solution, verify that:

- The vSwitches hosting the client VLANs are configured with sufficient ports to accommodate the maximum number of virtual machines a host can accommodate.
- All the required virtual machine port groups are configured and each server has access to the required VMware datastores.
- The interface is configured correctly for vMotion. Refer to the VMware vSphere Networking guide for details.

Refer to the list of documents in Chapter 6 for more information.

Deploying and testing a single virtual desktop

Deploy a single virtual machine to verify the operation of the solution. Ensure that the virtual machine has been joined to the applicable domain, has access to the expected networks, and that it is possible to log in.

Verifying component redundancy

To ensure that the all solution components maintain availability requirements, test the following scenarios related to maintenance or hardware failures:

- Power off one ScaleIO node and ensure that data access to the ScaleIO LUNs is still possible and that the data rebuild process runs properly.
- Disable each of the redundant switches in turn and verify that the vSphere host virtual machine remains available and functions properly.
- On a vSphere host that contains at least one virtual machine, enable maintenance mode and verify that the virtual machine can successfully migrate to an alternate host.
Chapter 6  Reference Documentation

This chapter presents the following topics:

EMC documentation ........................................................................................................... 96

Other documentation ......................................................................................................... 96
Chapter 6: Reference Documentation

EMC documentation

The following documents, available on EMC Online Support or EMC.com, provide additional and relevant information. If you do not have access to a document, contact your EMC representative.

- *EMC ScaleIO User Guide*
- *EMC VSPEX Private Cloud: VMware vSphere and EMC ScaleIO Proven Infrastructure Guide*

Other documentation

VMware The following documents, available on the VMware website, provide additional and relevant information:

- *Preparing vCenter Server Databases*
- *Understanding Memory Resource Management in VMware vSphere 5.0*
- *vCenter Server and Host Management*
- *vSphere Installation and Setup Guide*
- *vSphere Networking*
- *vSphere Resource Management*
- *vSphere Storage Guide*
- *vSphere Virtual Machine Administration*

Citrix Refer to the Citrix website for Citrix XenDesktop documentation, including *Citrix Windows 7 Optimization Guide for Desktop Virtualization.*

Microsoft Refer to the following topics on the Microsoft TechNet and Microsoft MSDN websites:

- Installing Windows Server 2008 R2
- SQL Server Installation (SQL Server 2012)
Appendix A  Customer Sizing Worksheet

This appendix presents the following topic:

Customer Sizing Worksheet for end-user computing ........................................98
Printing the worksheet ...........................................................................................99
Customer Sizing Worksheet for end-user computing

Before selecting a reference architecture on which to base a customer solution, use the Customer Sizing Worksheet to gather information about the customer's business requirements and to calculate the required resources.

Table 43 shows a blank worksheet. To enable you to print with ease, a standalone copy of the worksheet is attached to this guide in Microsoft Office Word format.

Table 43. Customer Sizing Worksheet

<table>
<thead>
<tr>
<th>User type</th>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>IOPS</th>
<th>Equivalent reference virtual desktops</th>
<th>No. of users</th>
<th>Total reference desktops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equivalent reference virtual desktops</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total
Printing the worksheet

To view and print the worksheet:

1. In Adobe Reader, open the Attachments panel as follows:
   - Select View > Show/Hide > Navigation Panes > Attachments
     or
   - Click the Attachments icon, as shown in Figure 44.

2. Under Attachments, double-click the attached file to open and print the worksheet.
This appendix presents the following topic:

**Customer Configuration Worksheet** ................................................................. 101

**Printing the worksheet** .................................................................................. 101
Customer Configuration Worksheet

Before configuring the solution, you need to gather some customer-specific configuration information such as IP addresses, hostnames, and so on. You can use the tables in this appendix as a worksheet to record the information. You can also print the worksheet and give it to the customer for future reference.

Printing the worksheet

A standalone copy of the worksheet is attached to this document in Microsoft Office Word format. To view and print the worksheet:

1. In Adobe Reader, open the Attachments panel, as follows:
   - Select View > Show/Hide > Navigation Panes > Attachments.
   or
   - Click the Attachments icon, as shown in Figure 45.

2. Under Attachments, double-click the attached file to open and print the worksheet.

Use the following tables to record the server, vSphere, network infrastructure, VLAN, and service account information for your configuration.

Table 44. ScaleIO information

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScaleIO management IP address</td>
<td></td>
</tr>
<tr>
<td>Storage pool name</td>
<td></td>
</tr>
<tr>
<td>ScaleIO volume name</td>
<td></td>
</tr>
<tr>
<td>ScaleIO network IP address</td>
<td></td>
</tr>
</tbody>
</table>

Table 45. Common server information

<table>
<thead>
<tr>
<th>Server name</th>
<th>Purpose</th>
<th>Primary IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS Primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS Secondary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 46. vSphere Server information

<table>
<thead>
<tr>
<th>Server name</th>
<th>Purpose</th>
<th>Primary IP</th>
<th>Private net (storage) addresses</th>
<th>VMkernel IP</th>
<th>vMotion IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>vSphere host 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vSphere host 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 47. Network infrastructure information

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
<th>IP</th>
<th>Subnet mask</th>
<th>Default gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet switch 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet switch 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 48. VLAN information

<table>
<thead>
<tr>
<th>Name</th>
<th>Network purpose</th>
<th>VLAN ID</th>
<th>Allowed subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Client access network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Storage network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Management network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account</td>
<td>Purpose</td>
<td>Password (optional, secure appropriately)</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Windows Server administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Root</td>
<td>vSphere root</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMware vCenter administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Citrix XenDesktop administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL Server administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Citrix Provisioning Services administrator</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>