

Azure Virtual Datacenter: Lift and Shift Guide

Identify and plan the migration of applications and servers to the Microsoft Cloud

By David Read, Thuy Le, Mark Ozur, Tycen Hopkins, and Ed Price
Azure Customer Advisory Team (AzureCAT)

Contents

- Introduction 4
- Identifying potential lift and shift candidates..... 6
 - Step 1: Capture application inventory 7
 - Operating System (OS) versions and patch levels..... 7
 - Application frameworks and third-party libraries 7
 - Device/infrastructure integration 7
 - Operations integrations 7
 - Availability SLA 7
 - Usage and utilization metrics 7
 - Backup/archive, high availability, and disaster recovery 8
 - Step 2: Verify operating system and platform support 8
 - Step 3: Verify runtime frameworks and third-party library support..... 8
 - Step 4: Identify any potential infrastructure requirements or challenges..... 9
 - Network connectivity and isolation requirements 9
 - Specialized networking 9
 - Special device/hardware integration 9
 - Storage 9
 - Step 5: Validate other criteria and create a preliminary list of candidates 10
 - Operations..... 10
 - Legal and regulatory requirements..... 10
 - Costs 10
- Choosing the right deployment model 11
 - Use the correct Azure virtual machine sizes 11
 - Select the correct deployment patterns for your workload..... 11
 - Related services and features 13
- Next steps..... 13
- See also 14

List of figures

Figure 1. Multi-stage filtering process for potential lift and shift candidate applications.....	6
Figure 2. Comparing deployment patterns for application workloads	12

Authored by Thuy Le, David Read, Mark Ozur, and Tycen Hopkins from Microsoft Azure Customer Advisory Team (AzureCAT). Edited by RoAnn Corbisier and Ed Price. © 2018 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Introduction

Large organizations often maintain a variety of well-established applications, resources, and services that are used internally or by external customers. Traditionally, the hardware required to host these resources was provided by physical datacenters, both on-premises and co-located externally. This arrangement leaves an organization's IT department managing all the hardware, software, and network configuration tasks required to keep these datacenters up and running.

The introduction of public cloud computing platforms offers IT departments a variety of new options for developing, deploying, and maintaining applications and servers. To review all the Azure Virtual Datacenter resources and tools, see the [Azure Virtual Datacenter page on the Azure Architecture Center](#). Servers, storage devices, and networking hardware previously required physical assets and configuration, but they are now available as virtual devices and services that can be spun up or down on demand. Rather than requiring a large initial investment in physical hardware, cloud computing allows organizations to use only the computing, networking, and storage resources they need, when they need them. Cloud services also offer several scalability and availability options that are not available on physical hardware.

A standard part of IT planning includes reviewing and leveraging the options available on the cloud, to drive down the required maintenance effort and total cost of ownership for your infrastructure. That planning starts with an understanding of how to deploy or migrate applications or services into the cloud. [Microsoft Azure Migrate](#) offers a way to easily discover, assess, and migrate your on-premises virtual machines to Azure.

The Microsoft Azure cloud platform includes platform as a service (PaaS) and infrastructure as a service (IaaS) offerings. PaaS on Azure provides application hosting, storage, and database services without needing to worry about the underlying server resources. The Azure platform handles most of the hosting, networking, infrastructure, and configuration tasks required in a traditional datacenter. For example, you no longer need to patch your virtual machines or manage virtual machine operating system health. By offloading responsibilities, you greatly reduce the operations and support effort needed to stand up and maintain an application or service.

IaaS, on the other hand, allows you to create virtualized servers, storage, and network infrastructure, giving you a similar level of control over your cloud-based assets as you would have over the hardware in an on-premises datacenter. Applications or services hosted on IaaS require more operations and maintenance than those running on PaaS (although still usually far less than that required for on-premises physical hardware), but they offer more flexibility in how those applications or services are structured.

Choosing the mix of PaaS and IaaS depends on the nature of the workloads you want to host on the cloud. New applications or services are not bound by legacy infrastructure and can use a greenfield (cloud-native) architecture, which allows you to take advantage of the PaaS and IaaS features from the beginning of the design process.

However, if you built applications or services for an on-premises datacenter, you need to carefully plan your migration to cloud services. You could invest a lot of development effort to refactor workloads to take full advantage of cloud service offerings. In addition, your internal development teams might not have the ability to modify third-party or legacy applications.

Your organization will want to minimize the amount of change required to get an application or

workload migrated to cloud hosting. In these scenarios, the lift and shift migration model might be the right choice. For enterprise customers, the bulk of your application portfolio should be supported by a lift and shift migration model. A follow-up paper will discuss handling the edge cases and situations that don't fit this model.

The lift and shift model involves moving your existing applications or services to Azure-based virtual machines, with an operating system and networking configuration as close to their current on-premises configuration as is possible on a cloud platform. A successful lift and shift migration takes advantage of the infrastructure benefits and management features of the cloud, while minimizing the both the migration cost and decreasing the time required to complete the migration.

You should consider whether PaaS can be part of your lift and shift migration (for example, simple web applications migrating to Azure web app hosting), and you should not ignore these services when moving through this process. However, for existing complex applications built around the on-premises or physical datacenter model, making use of PaaS can require an extensive retrofitting effort. This can potentially involve large development costs, which is what the lift and shift model tries to avoid.

Because large preexisting enterprise applications tend to require more modifications to support PaaS functionality, this document focuses on IaaS migrations, where you minimize the need for refactoring by matching your existing hosting and network infrastructure. However, when you choose between PaaS and IaaS technologies for migration, your decision might be based on migration agility and speed. Which option gets your application running on the cloud? Which choice allows you to migrate the most applications and services within a window of time?

Not all applications or services are a good fit for lift and shift migrations, and some that are a fit can be more difficult to migrate than others. Hardware requirements, licensing issues, and incompatibilities might make running software on IaaS resources problematic. In many ways, the research and planning steps required for a conventional datacenter migration also apply to a lift and shift process, and any existing information and requirements gathering processes in place for migrations can also be applied to lift and shift.

Identifying lift and shift candidates from your pool of applications requires a few steps. The first part of this document guides you through the information gathering process used to filter which applications will be easiest and most cost effective to quickly migrate to Azure IaaS.

For those applications that are a good fit, migrating to Azure offers several benefits in resource usage and scalability capabilities. The second part of this document discusses which deployment model can best handle your application's resource requirements and usage patterns, and how to optimize your costs based on those requirements.

This guide is a starting point when considering the migration of existing applications and services. The processes described below are meant to be iterative. By working to identify a first round of candidates for lift and shift, you will build an understanding of what's required to host and maintain applications in Azure, along with increasing the accuracy of cost estimates. This knowledge will make identifying subsequent candidates much easier.

Note that the Azure platform is continuously adding features and services, and costs can change (generally lower) as new capabilities come online. Although applications and services might not be candidates for lift and shift migrations now, they might be in the future, and any iterative review process should take platform changes into account.

Identifying potential lift and shift candidates

The process of identifying applications suitable for a lift and shift migration is essentially a multi-stage filtering process. Your team will look at each application or service you are considering as a candidate for moving to the cloud, and then generate a list of all the hosting resources used by the application, including both physical and virtual devices.

The list of resources is used to assess the viability of hosting the candidate on Azure. This validation occurs in several increasingly detailed steps. Any application or service that doesn't meet the criteria assessed for a particular step in this process is moved to a separate list for applications that require an alternative migration approach. Those that do meet the criteria remain in the lift and shift migration candidate list and move on to the next step.

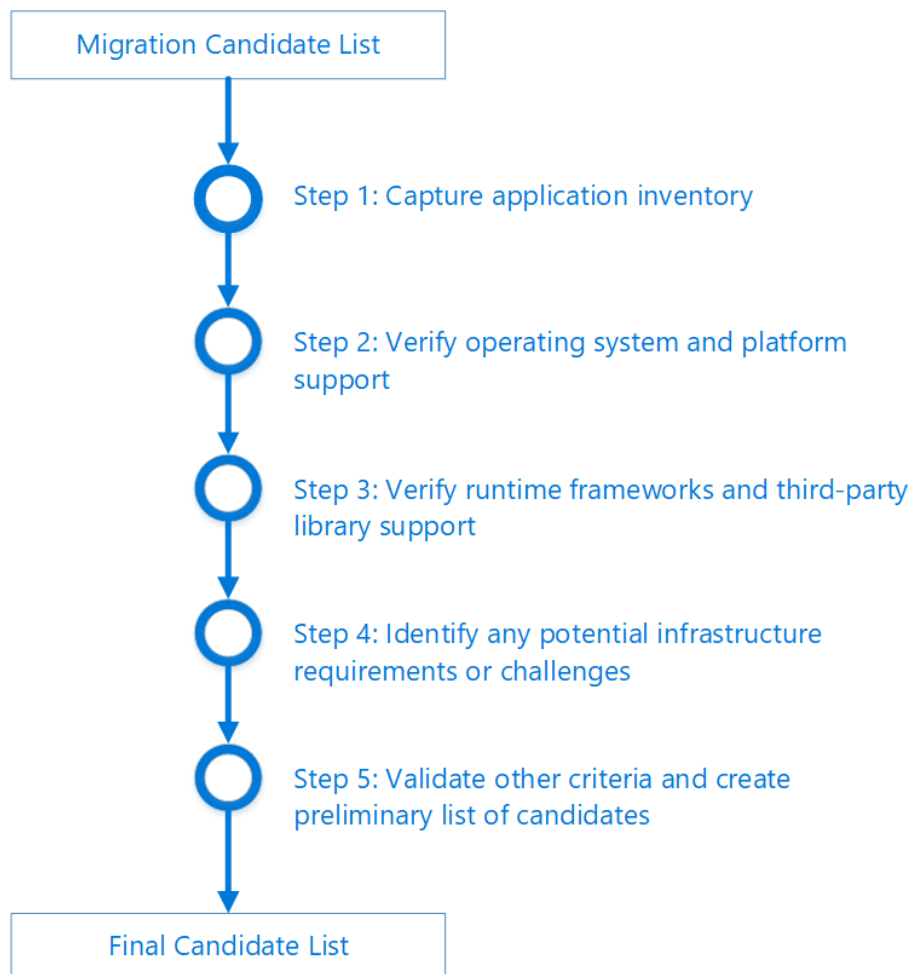


Figure 1. Multi-stage filtering process for potential lift and shift candidate applications

At the end of this process, you should have a list of candidate applications you can successfully migrate to Azure hosting with a minimal reengineering investment. At that point, you can begin to focus on how your application and service workloads can take advantage of cloud-specific deployment models to optimize your use of cloud resources.

Step 1: Capture application inventory

For each candidate application or service, have the teams responsible for hosting and maintaining it generate a full inventory of all dependencies. This inventory should not dive into extensive detail about each resource, but it should be thorough in listing all types of resources used by the candidate. It should focus on capturing key details about each resource type and what it does in relation to the application or service.

Although automated network detection tools and preexisting documentation will be a great help in creating these lists, it is important to manually review lists of datacenter resources, up to and including inspecting the physical datacenter. Systems that involve specialized hardware or legacy applications might have a lot of documentation that needs additional attention to make sure nothing is missed.

Note which servers are running on physical hardware vs. which ones are virtualized, as this can make a significant and measurable difference in the effort required to migrate servers. Virtualized servers have already gone through many of the steps required in a cloud migration, and tools are available to help migrate existing VMWare and Hyper-V virtual machines to Azure. Also note if any software or licensing issues that require resources to run on physical hardware.

In general, the inventory should capture the following information.

Operating System (OS) versions and patch levels

For any servers used by a candidate application or service, it is important to note the platform, operating system, version, and what patches have been applied.

Application frameworks and third-party libraries

For any software that is a part of your candidate application or service, identify any application frameworks that are used, such as .NET or Java/JDK. Also make note of the framework version in use.

Similarly, take note of any third-party code or libraries used, along with the currently installed version.

Device/infrastructure integration

Identify any network switches, routers, specialized load balancers, firewalls, or other devices required to run your application or service. Note any specialized appliances or other hosting hardware, as well as any dedicated storage devices.

Operations integrations

List any automated agents or other processes providing monitoring, security, and notification capabilities for your applications, services, or network infrastructure. If there are manual processes involved in maintaining your system, note these as well.

Availability SLA

For each resource, document expected SLA and uptime requirements.

Usage and utilization metrics

Collect any usage and hardware utilization information for your server resources. Include information about both nominal and peak loads, and note any predictable patterns of usage, such

as high traffic during particular dates or at specific times of day.

Backup/archive, high availability, and disaster recovery

Document how resources are backed up, and note archiving and retention policies. Note the infrastructure in place to meet required availability SLAs, and list what disaster recovery policies and practices are in place, including recovery point objectives (RPO) and recovery time objectives (RTO).

Step 2: Verify operating system and platform support

Azure supports a wide range of guest OS images, including both Windows and Linux options. However, that support is not limitless. You'll need to check if the list of OS versions you recorded in your inventory are all supported.

Consult the list of [Azure endorsed Linux distributions](#) to see the officially supported Linux distributions that qualify for the standard Azure platform SLA. See [Information for Non-Endorsed Distributions](#) for more details on the minimum requirements to deploy a custom Linux guest OS.

For Windows servers, the Azure Marketplace provides supported preconfigured images for Windows Server 2008 R2 and later. Using custom guest OS images, you can deploy older versions that date back to Windows Server 2003. However, Windows versions past their end of support date require a Custom Support Agreement (CSA).

Some Windows Server roles and features, like DHCP server, Hyper-V, and BitLocker data drive encryption, are not supported on Azure guest virtual machines. See the full list of [Microsoft server software support for Microsoft Azure virtual machines](#) for details. Also note that while these roles and features might be not supported within Windows server virtual machines, almost all the functionality they represent can be supported using other Azure-based services or applications.

Consider whether you can migrate software that is hosted on legacy or unsupported operating systems, to newer Azure-supported versions, without requiring substantial changes or custom development. If not, then that application or service should be moved to the alternative migration approach list, while the remaining potential lift and shift candidates move to the next step.

Step 3: Verify runtime frameworks and third-party library support

Tied together with OS support, you will need to validate whether any application runtimes and related technologies, used by the candidate application or service, are supported on Azure. This includes making sure the currently used version of common frameworks, like .NET and Java, will work. In addition, any third-party libraries used also need to work properly on the target OS version.

You must be diligent on this issue, as upgrades to OS, frameworks, and libraries can introduce cascading dependency issues that can be very difficult to deal with in the migration process. As with OS support, if older frameworks and libraries cannot be upgraded to support the Azure environment without substantial changes to your application or service, then move that application or service to the alternative migration approach list before moving the remaining lift and shift candidates to the next step.

Step 4: Identify any potential infrastructure requirements or challenges

After you determine that a candidate application or service can operate as an Azure guest virtual machine, the next step is to identify any other technical concerns with hosting resources on Azure. This stage can require in-depth research to compare Azure capabilities with your current on-premises infrastructure, ensuring all the features and services required for the candidate are available on Azure.

If the infrastructure requirements of a candidate can't be met on Azure without unacceptable amounts of modification and cost, move that application or service to the alternative migration approach list.

Each organization, application architecture, and datacenter can have a unique infrastructure, so the list of potential issues to research will be unique to each of your candidate applications or servers. That said, many of the most common infrastructure challenges can be broken into the following broad categories.

Network connectivity and isolation requirements

How do resources within your candidate application or server communicate with each other and the outside world? Is the candidate meant for internal use only? Will users need to access it over the public network? What kind of routing and firewall capabilities do you need?

The [Azure Virtual Datacenter](#) provides a comprehensive approach to maintaining your overall governance, security, and network design. Can the existing [Azure Virtual Networks](#) and [network virtual appliances](#) support your needed capabilities?

Also, does the candidate application or service have any dependencies on resources that will need to remain hosted on-premises, or at another datacenter or cloud location? Will on-premises applications, services, or users need to use the candidate from within your local on-premises network? Can Azure connectivity solutions, such as [ExpressRoute](#) or VPN connections, support these requirements?

Specialized networking

Does your existing infrastructure depend on specialized networking capabilities (such as multicast, Relocatable VIPs, Custom overlay networks, and so on)? Are these capabilities supported in Azure Virtual Networks?

Special device/hardware integration

Are there any specialized hardware devices that the candidate application or service needs to utilize (such as a Telephone/PBX or physical security device)? Are there equivalent virtual devices or appliances to provide this capability on Azure?

Storage

What are your overall storage requirements? Here are some common questions:

- How much total storage capacity do you need?
- Do you need to share disks between virtual machines?
- Do your virtual machines require high amounts (for example, greater than 50 TB) of attached storage?

- What kind of latency is acceptable for attached storage? Is ultra-low latency a requirement?

Can Azure Storage, Managed Disks, and VHD files used in attached storage, support these requirements?

Also take into account the organizational and legal requirements for archiving, and long-term storage retention for the candidate application or service. Can the policies be supported in Azure?

Step 5: Validate other criteria and create a preliminary list of candidates

At this point, you should have a high level of confidence that your remaining list of applications and services, on a technical level, are good lift and shift candidates. However, from an operations perspective, less explicitly technical criteria can be just as important in deciding if a migration is justified.

Operations

If a candidate is successfully migrated to Azure, how much will your operations processes need to change? Here are some common concerns:

- How will your backup and recovery mechanisms need to change when resources are hosted on Azure? Will your disaster recover planning need to be modified?
- Will existing monitoring tools work in the new environment, and, if not, how much work will be required to identify and implement an alternative approach?
- Will your existing high-availability architecture translate to Azure?
- Will existing DevOps tools and processes need to be modified?

Legal and regulatory requirements

Are there legal or regulatory compliance concerns related to your application or service, such as privacy laws or data retention requirements? As with any migration plan, you will likely need to consult your organization's legal compliance team to thoroughly vet the lift and shift process for your candidate applications and services.

Costs

Ultimately, a migration is a business decision, and if the cost of moving to the cloud can't be justified, there's no point moving forward. The initial migration involves one-time costs related to setting up your Azure environment and migrating applications and services. However, aside from these migration costs, you must [understand the ongoing operations costs](#) required to host your application or service.

You should now have a good idea about the number and type of resources you will need on Azure to migrate your candidate application or service, and these should allow you to build cost estimates for the migration. To understand ongoing costs, review and verify any assumptions you have about operations and the related support tasks. Azure has extensive capabilities to scale your application resource needs. As part of this review process, consider the deployment models discussed in the following sections to optimize your Azure resource usage.

With this information, you can generate quality cost estimates for hosting and maintaining your candidate applications and services.

Choosing the right deployment model

In a physical datacenter, the initial costs (notwithstanding depreciation models) of purchasing and configuring hardware, can be very large in comparison to ongoing operations expenses. Your peak loads might be much higher than most operations, and some workloads may only be needed occasionally. However, you still need to purchase and operate all the server hardware necessary to handle these higher loads.

On a public cloud platform, there is no hardware to purchase and no upfront costs to create a new virtual machine. You pay for the compute, storage, and networking resources you actually use. Virtual machines that aren't needed for a specific time can be shut down to save compute and networking costs, and you can start them back up when they are needed. You can spin up additional virtual machines during high-traffic or special events like Black Friday or month-end processing. You can resize individual virtual machines to provide additional capacity, and when the load diminishes, you can spin them down again. [Azure Cost Management](#) can help you optimize what you spend on the cloud, while maximizing cloud potential.

The lift and shift migration model moves your applications and services to the cloud with as little modification as possible. However, you will save a lot on your ongoing costs if you properly understand your resource utilization and pick the best deployment model to use when you migrate resources.

Use the correct Azure virtual machine sizes

The first step to optimize your Azure resource usage is to understand the capabilities of your existing hardware. Use that information to select the best Azure virtual machine size to handle that load. Larger virtual machines cost more to run, so make sure you use the right size for your workload.

Consider the age and utilization levels of your current hosting platform. It is possible that a Compute core on Azure is materially faster, requiring fewer cores than your current deployments. Make good use of any utilization data you have on your existing hosts or virtual machines to identify performance requirements. Note the nominal and peak usage on each server and map these to an appropriate virtual machine size:

- [Sizes for Windows virtual machines in Azure](#)
- [Sizes for Linux virtual machines in Azure](#)

Virtual machines can easily be resized, so after deployment to Azure, continue to monitor and tune your usage assumptions to make sure you're using the correct virtual machine sizes, as your loads change over time.

Select the correct deployment patterns for your workload

The simplest way to set up an IaaS application or service in Azure is to create a fixed number of always-on virtual machines, in sizes capable of handling your peak usage requirement. This will handle your needs, but it imposes the costs of operating at peak capacity at all times.

One of the key advantages of IaaS on the Azure platform is the ability to scale your virtual machines as your needs change. You can reduce costs by adding or removing additional virtual

machines instances, or by increasing the capabilities of existing virtual machines.

Characterizing Application Workloads

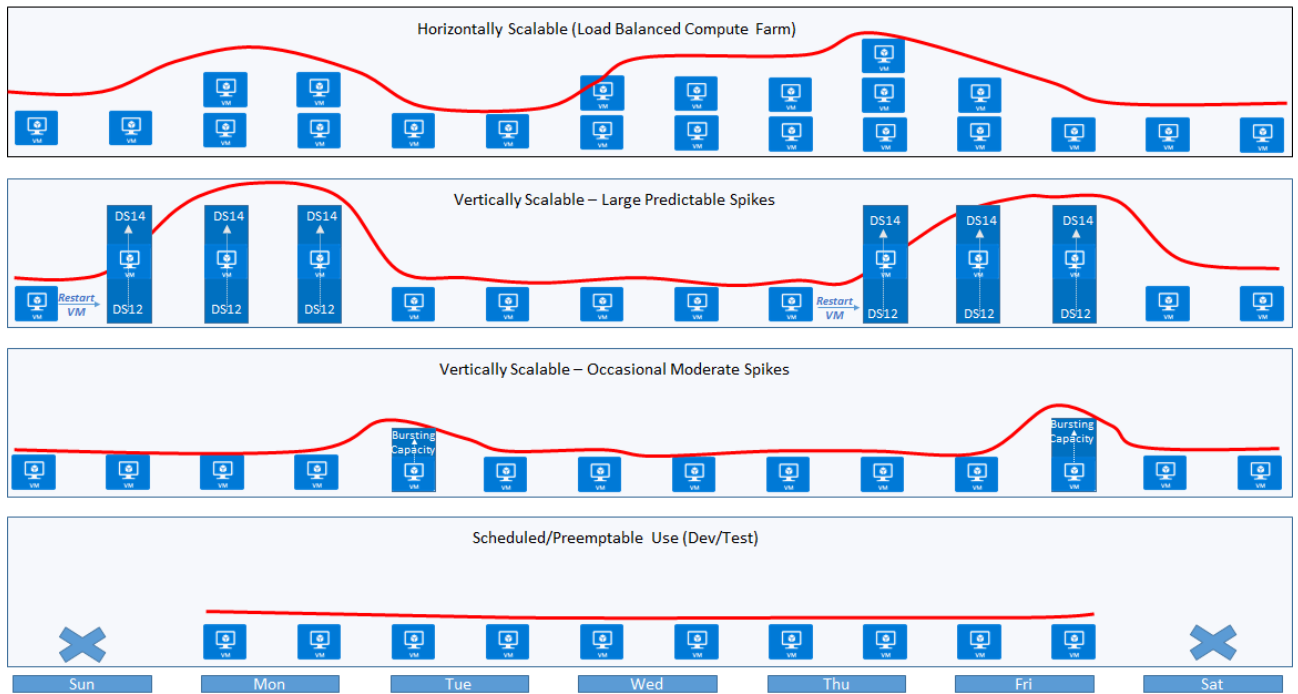


Figure 2. Comparing deployment patterns for application workloads

For each application or service that you are planning to migrate to Azure, review the following deployment patterns to see if you can use them to optimize your resource utilization without requiring any significant changes to the applications themselves.

Horizontal scale out/in

Horizontal scaling involves spinning up additional virtual machines to handle increased demand on your application or service. When the demand declines, these additional virtual machines are spun back down to reduce resource usage.

This pattern is well suited for stateless applications, such as the web tier of an n-tier application, calculation services, or other processes that handle independent transactions. Applications that already make use of a load balancer are usually good candidates for this pattern.

Vertical scale up/down

Vertical scaling involves increasing the size of individual virtual machines to handle increased demands. When the demand decreases, the virtual machine size is decreased. If workload increases are predictable (for example, based on a specific day each month), vertical scaling can be scheduled to increase the virtual machine size during that period of time. Scaling can also be set to happen dynamically, based on resource usage.

This pattern is suited for applications like monolithic database servers, stateful services, or applications that need to track events across transactions.

If you scale between sizes, you do not need to restart the guest virtual machine. Any applications or services that make use of this pattern, need to be able to tolerate this brief outage.

Scheduled on/off

For any workloads that do not need to be running 24/7, the relevant virtual machines can be turned off when not needed. This can be done manually or managed using rules to schedule when a virtual machine is started or stopped.

This is a particularly useful pattern for development and test resources that might only need to be used occasionally, or for business applications that do not need to be on at all times, such as reporting services.

Workload consolidation using containers

Some applications may be good candidates for containerization. Container platforms, like Docker or Kubernetes, allow you to consolidate your code into an easy-to-deploy package that operates in an isolated software environment within a larger host OS. This can provide a more efficient way to run smaller applications that do not need the resources of a dedicated virtual machine.

Related services and features

- [Azure Automation](#)
- [Virtual Machine Scale Sets](#)
- [Azure Service Fabric](#)
- [Azure App Service Environments](#)
- [Azure Container Service](#)
- [Azure DevTest Labs](#)

Next steps

Many of the risks and concerns involved in a lift and shift migration are identical to those experienced by IT teams contemplating any datacenter migration. You will need to understand the full complexity of any systems you intend to move, and you must accommodate the existing functionality in the new environment. Consider testing your approach to migration using [Azure Migrate](#).

The process described in this document explains what's involved in migrating workloads to the cloud, and what kind of cost model this migration will entail. However, you might need to deploy a few proof-of-concept applications to Azure before you feel confident estimating these costs.

Consider taking three or four representative migration candidates through this process to fully understand what's required, and to further refine your cost model when planning for future migrations. The more applications and services you move through this process, the higher confidence you'll have in the costs and technical feasibility of future migrations. You can use the provided [example Excel spreadsheet](#) to help track the high-level criteria for each application candidate you are considering.

See also

Additional Virtual Datacenter Resources

- [Azure Virtual Datacenter on the Azure Architecture Center](#)
- [Azure Virtual Datacenter eBook](#)
- [Azure Virtual Datacenter: A Network Perspective](#)
- [Azure Virtual Datacenter: Presentation Deck](#)

Additional Lift and Shift Resources

- [Azure Migrate](#)
- [Azure Cost Management](#)
- [Lift and shift SQL Server Integration Services workloads to the cloud](#)
- [Lift, shift, and modernize using containers on Azure Service Fabric](#)