Module 4: Creating Managed Server Applications in Azure

Contents:

	Module overview
Lesson 1:	**Infrastructure-Backed Platform-as-a-Service (PaaS)**
Lesson 2:	**High-Performance Compute (HPC)**
Lesson 3:	**Migration**
Lab:	**Deploying Managed Server Workloads to Azure**
	Module review and takeaways

Module overview

This module describes services that use infrastructure but manage the infrastructure on behalf of the user instead of obfuscating the infrastructure resources. The module focuses on infrastructure-backed PaaS options such as Azure Service Fabric, Container Service, and App Service Environments. The module will explore how to deploy custom workloads to these services such as an HPC batch processing task.

Objectives

After completing this module, students will be able to:

- Describe the differences between App Service Environments, Service Fabric and Container Service.
- Use Azure Batch to manage an HPC workload.
- Migrate to an Infrastructure-backed PaaS service from another IaaS service or a legacy Cloud Service.

Lesson 1: Infrastructure-Backed Platform-as-a-Service (PaaS)

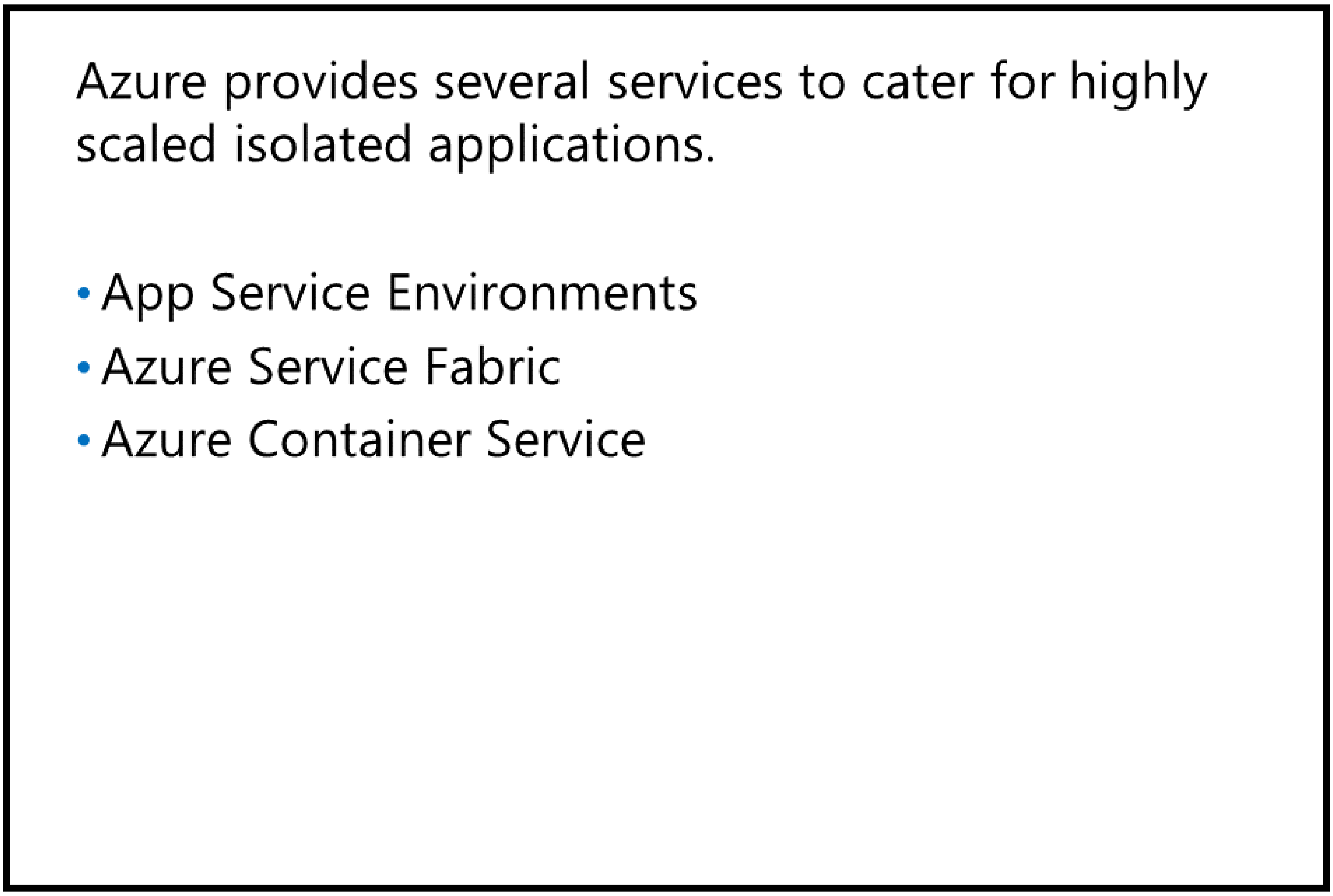
The need to deploy highly scalable, isolated applications with secure network access and high memory utilization is one that is growing with increased cloud adoption. This lesson will describe the options within Microsoft Azure to deploy App Service Environments, Azure Service Fabric and Azure Container Service applications. The growth of containerized applications and microservice based applications requires the use of these services to deploy open source solutions to deploy, manage and orchestrate at low cost and high flexibility.

Lesson objectives

After completing this lesson, you will be able to:

- Describe App Service Environments and distinguish between v1 and v2.
- Describe Azure Service Fabric and distinguish between Microservices and Containers
- Describe Azure Container Services and the variants, Docker, Mesosphere and Kubernetes
- Deploy an App Service Environment with a web app

Infrastructure Backed PaaS



Every Microsoft Azure PaaS service is already hosted on Azure IaaS. Azure provides several options for architecting and hosting your own PaaS applications at scale in the same secure and isolated infrastructure within the Azure Datacenters. To be able to provide highly scalable applications requiring isolation with secure network access and high memory utilization, there are three services available. App Service Environments provides a dedication scalable home for Azure Web Apps. Azure Service Fabric provides a cluster of VMs to host containers and microservices for those applications. Finally, there is the Azure Container Service this provides open source tools to orchestrate containers based on Docker Swarm, Mesosphere and Kubernetes clusters. The benefit of all these technologies is that Azure lets you concentrate on the application architecture, and code whilst it looks after the infrastructure and the orchestration of the containers. In this lesson, we will detail these services and their intended uses.

App Service Environments

- Dedicated environment for high scale, secure Apps.
  - Very high scale.
  - Isolation with secure network access.
  - High memory utilization.
- Single or Multi region
- Deployed to a Virtual Network
- An ASE is dedicated exclusively to a single subscription (Max 100 instances)

App Services are useful because they separate many of the hosting and management concerns for your web application and allow you to focus on your application's functionality and configuration. There are some scenarios, however, where you require a bit more control.

For example, your organization may require you to make sure that all of the virtual machines hosting your web applications do not allow any outbound requests. This is a typical scenario when implementing a web solution that must be PCI compliant. With App Services, you can't access or modify the configuration of the virtual machines hosting your applications. You do not have any mechanism to implement this requirement.

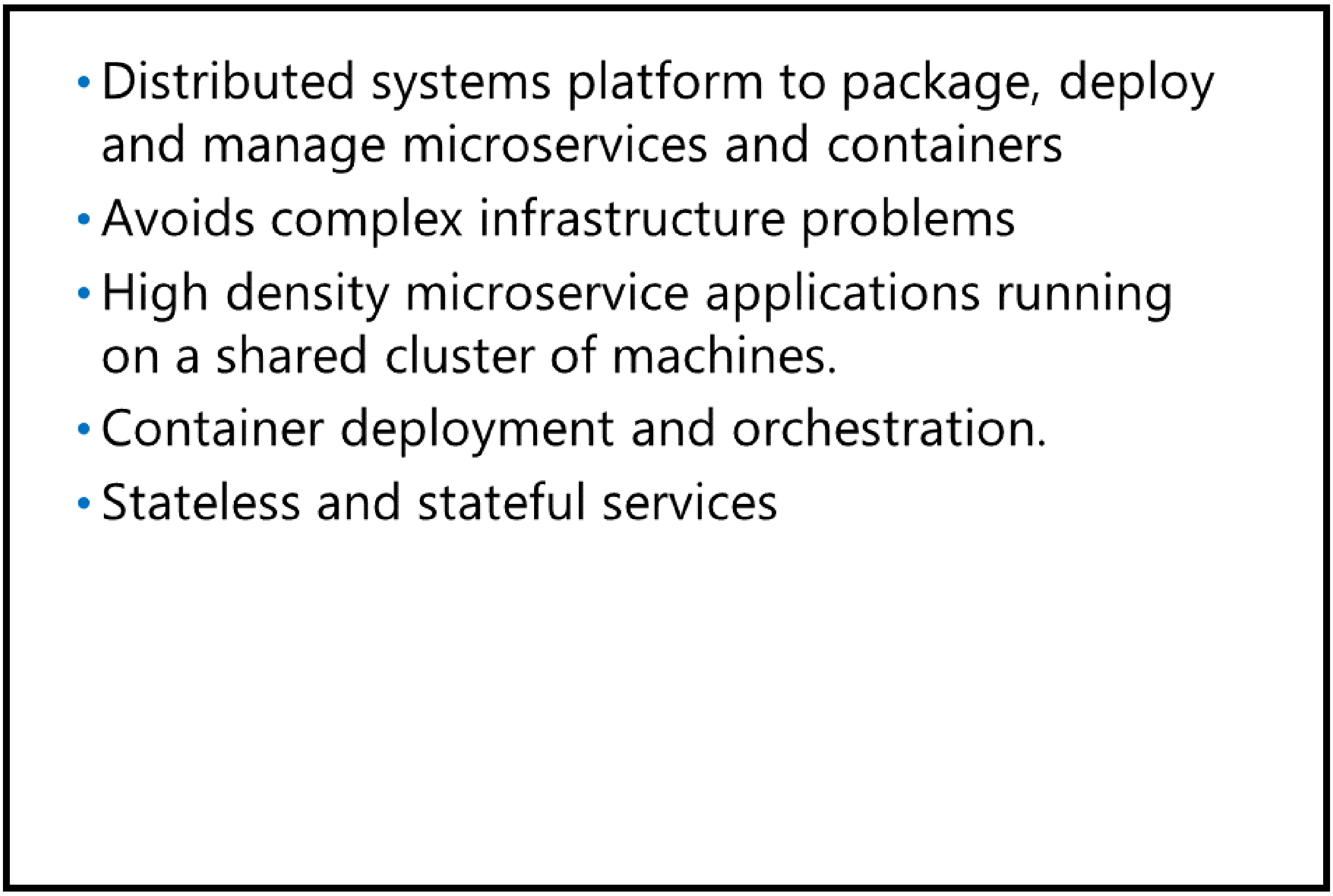
To implement scenarios where you require more control, you can use the App Service Environment (ASE) service in Azure. ASE allows you to configure network access and isolation for your applications. ASE also allows you to scale using pools of instances far beyond the limits of a regular App Service plan instance. Finally, ASE instances are dedicated to your application alone. You retain much of the convenience of using App Services such as automatic scaling, instance management, and load balancing when using ASE but you gain more control.

Networking in ASE

With ASE, you can configure network access using the same concepts and paradigms that you use in Virtual Machines. Environments are created within a subnet in an Azure Virtual Network. You can use Network Security Groups to restrict network communications to the subnet where the Environment resides. You can also use a protected connection to connect your Virtual Network to corporate resources so that your ASE instance can securely use the resources.

ASE is available in v1 and v2. The ASE v1 is useful since it can take advantage of both classic and Resource Manager Virtual Networks whilst v2 can only use Resource Manager resources. ASE v2 also automates most of the scaling, creation, and maintenance which v1 requires to be carried out manually.

Azure Service Fabric



Azure Service Fabric provides a distributed systems solution making it simple to package, deploy, and manage scalable and reliable containers and microservices. Developers can use Service Fabric to assist with the developing and managing cloud native applications.

The use of Service Fabric can avoid complex infrastructure problems and leave operations and development staff to concentrate on implementing mission-critical workloads that are easy to manage, scale and result in a reliable solution. Service Fabric is intended for container-based applications running on enterprise-class, cloud-scale environments.

Microservice based applications

Service Fabric creates a cluster of VMs that allow you to run high density, massively scalable applications composed of microservices. These applications can be stateful or stateless microservices running in containers which Azure Service Fabric will assist with provisioning, deploying, monitoring and managing in fast and efficient, automated fashion.

Service Fabric powers many Microsoft services today, including Azure SQL Database, Azure Cosmos DB, Cortana, Microsoft Power BI, Microsoft Intune, Azure Event Hubs, Azure IoT Hub, Dynamics 365, Skype for Business, and many core Azure services. Azure Service Fabric is ideal for hosting cloud-native applications at high scale often growing to hundreds of thousands of machines.

A microservice could be a user profile, a customer’s shopping cart, queues, caches, or any other component of an application. Service Fabric is a platform built to handle microservices by placing each one in a uniquely named container.

Service Fabric provides comprehensive runtime and lifecycle management capabilities to applications that are composed of these microservices. It hosts microservices inside containers that are deployed and activated across the Service Fabric cluster. The use of microservices adds the ability to increase the density of the containers used by a service.

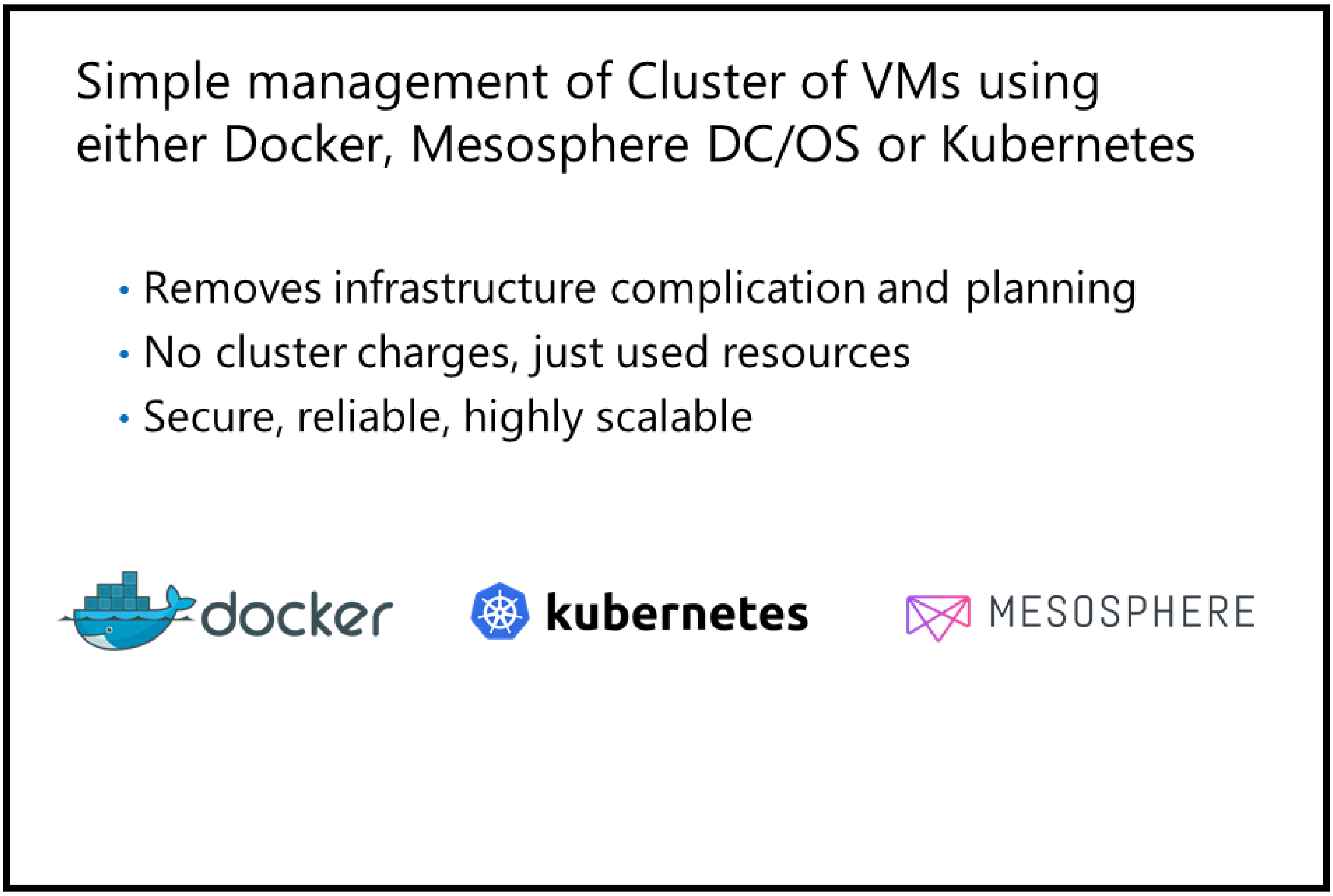
As an example, a single cluster for Azure SQL Database contains hundreds of machines running tens of thousands of containers hosting a total of hundreds of thousands of databases. Each Azure SQL database is a Service Fabric state-full microservice.

Application lifecycle management

Azure Service Fabric supports the application lifecycle and CI/CD of cloud applications including containers.

Service Fabric empowers operators to use simple workflows to provision, deploy, patch, and monitor applications. This dramatically reduces the workload for the operators. Service Fabric is integrated with CI/CD tools such as Visual Studio Team Services, Jenkins, and Octopus Deploy.

Azure Container Service



Azure Container Service (ACS) provides several alternative ways to simplify the creation, configuration, and management of a cluster of virtual machines that are configured to run your applications in a container. Using open-source orchestration and scheduling tools provides a large community of experience and expertise to assist in deploying and managing your containerized applications.

ACS provides three distinct container orchestration technologies:

- Docker Swarm
- Mesosphere DC/OS
- Kubernetes

ACS uses Docker images to provide container portability. The use of the orchestration technologies enables the ability to scale applications to tens of thousands of containers. ACS makes no charges for the service or orchestration since all the tools provided are open source software. Charges are made for resource utilization only. ACS is available for Standard A, D, DS, G, GS, F, and FS series Linux virtual machines. You are only charged for these VMs, Storage and networking resources actually used.

The latest service available is Kubernetes clusters. This is the AKS service. AKS passes the cluster management functions to Azure and removes it from the operator or administrator. Azure manages health, maintenance, and monitoring. AKS also provides easy scaling, automatic version upgrades, and self-healing master nodes.

Since Azure provides all the usual management functions, there is no SSH access to the AKS cluster.

Access is provided to the Kubernetes API endpoints. This allows any software capable of communicating with Kubernetes endpoints to be used. Examples could include kubectl, helm, or draft. Whilst Azure handles the infrastructure management, Kubernetes itself handles your containerized applications.

The list of Kubernetes management features includes:

- Automatic binpacking
- Self-healing
- Horizontal scaling
- Service discovery and load balancing
- Automated rollouts and rollbacks
- Secret and configuration management
- Storage orchestration
- Batch execution

Lesson 2: High-Performance Compute (HPC)

Azure provides solutions to allow for massively scalable parallel processing of memory-intensive workloads, such as 3D rendering and tasks that process, transform, and analyze large volumes of data. Azure Batch provides this service for short-term massively scalable infrastructure. For other cloud-based and even hybrid solutions, Azure provides access to both Windows and Linux HPC clusters. This lesson will describe these solutions and their uses.

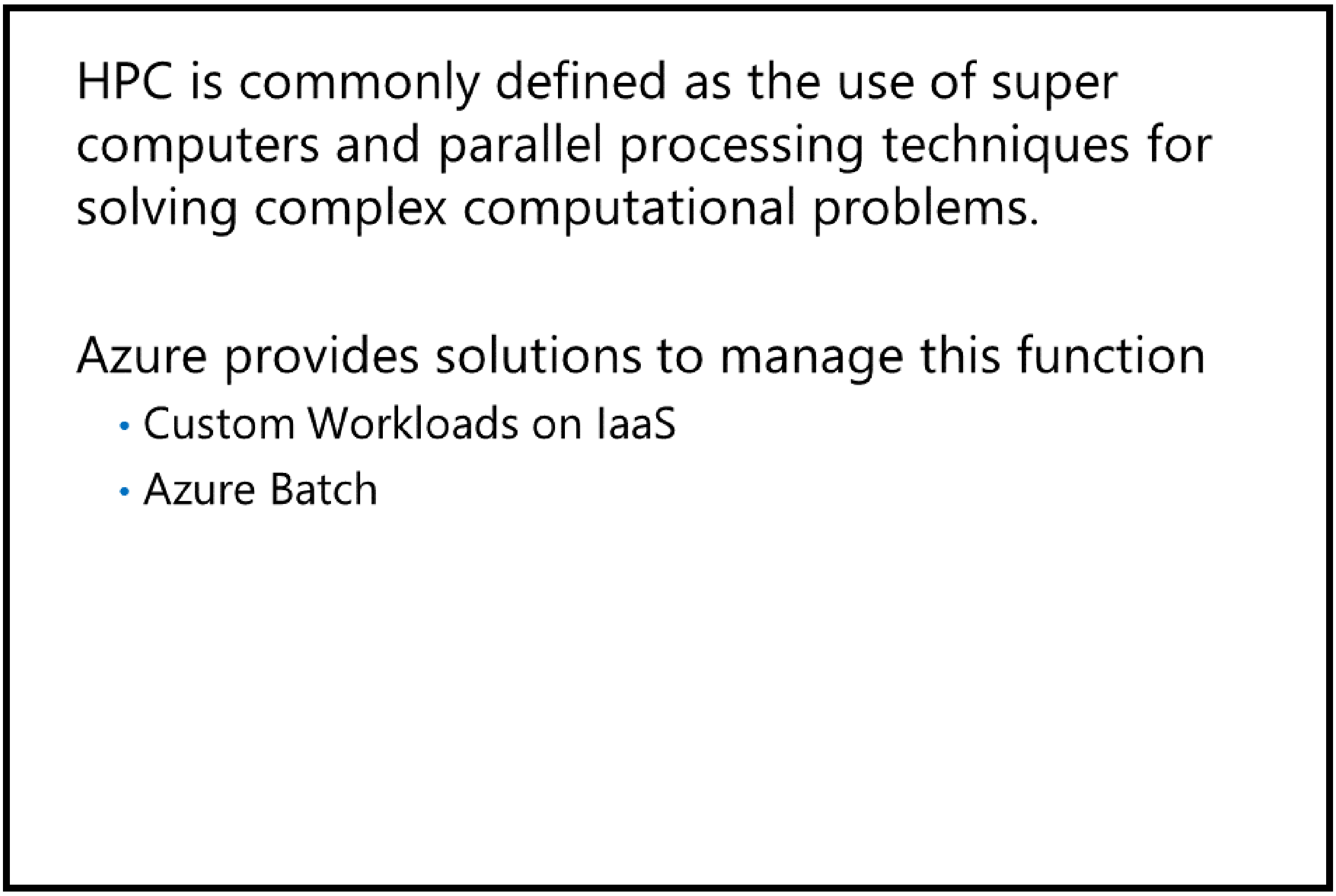
Lesson objectives

After completing this lesson, you will be able to:

- Describe Azure HPC Pack features.
- Describe the Azure Batch service.
- Understand the use of the HPC pack in Cloud only and hybrid scenarios.

- Deploy an Azure HPC pack for Windows.

High Performance Computing (HPC)



Introducing High-Performance Computing (HPC)

Traditionally, complex processing was something saved for universities and major research firms. A combination of cost, complexity, and accessibility served to keep many from pursuing potential gains for their organization from processing large and complex simulations or models. Cloud platforms have democratized hardware so much that massive computing tasks are within reach of hobbyist developers and small to medium-sized enterprises.

High-Performance Computing (HPC) typically describes the aggregation of complex processes across many different machines thereby maximizing the computing power of all of the machines. Through HPC in the cloud, one could create enough compute instances to create a model or perform a calculation and then destroy the instances immediately afterward. Advancements in the HPC field have led to improvements in the way that machines can share memory or communicate with each other in a low latency manner.

Remote Direct Memory Access

Remote Direct Memory Access, or RDMA, is a technology that provides a low-latency network connection between processing running on two servers, or virtual machines in Azure. This technology is essential for engineering simulations, and other compute applications that are too large to fit in the memory of a single machine. From a developer perspective, RDMA is implemented in a way to make it seem that the machines are "sharing memory." RDMA is efficient because it copies data from the network adapter directly to memory and avoids wasting CPU cycles.

The A8 and A9 VM size in Azure use the InfiniBand network to provide RDMA virtualized through Hyper-V with near “bare metal” performance of less than 3-microsecond latency and greater than 3.5 Gbps bandwidth.

HPC Pack using Azure Virtual Machines

HPC Pack is Microsoft’s HPC cluster and job management solution for Windows. The HPC Pack can be installed on a server that functions as the "head node," and the server can be used to manage compute nodes in an HPC cluster.

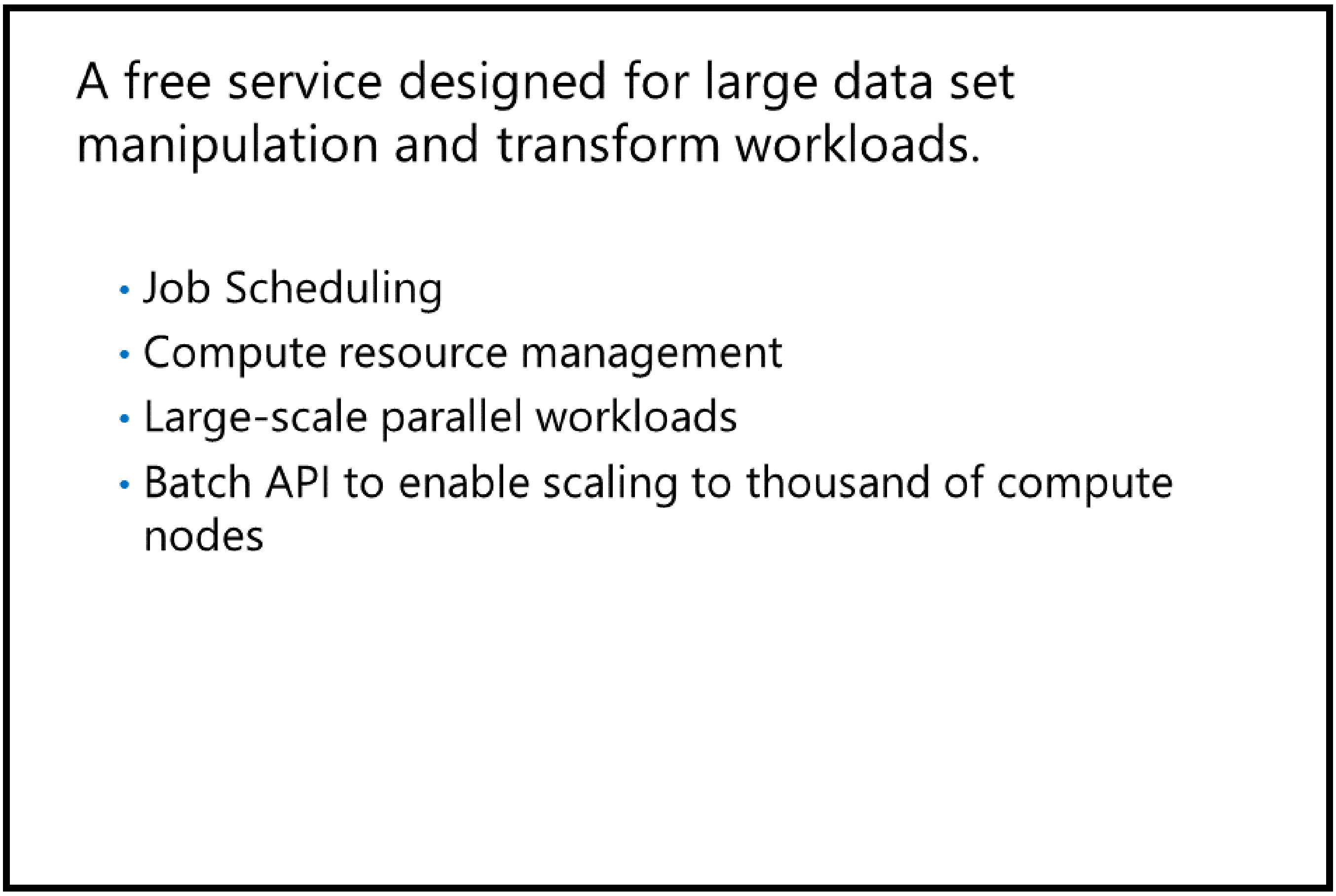
HPC Pack is not required for you to use the A8, A9, A10, and A11 instances with Windows Server, but it is a recommended tool to create Windows HPC clusters in Azure. In the case of A8 and A9 instances, HPC Pack is the most efficient way to run Windows MPI applications that access the RDMA network in Azure. HPC Pack includes a runtime environment for the Microsoft implementation of the Message Passing Interface for Windows.

HPC Pack can also be used in hybrid scenarios where you want to "burst to Azure" with A8 or A9 instances to obtain more processing power.

RDMA on Linux Virtual Machines in Azure

Starting with HPC Pack 2012 R2 Update 2, HPC Pack supports several Linux distributions to run on compute nodes deployed in Azure VMs, managed by a Windows Server head node. With the latest release of HPC Pack, you can deploy a Linux-based cluster that can run MPI applications that access the RDMA network in Azure. Using HPC Pack, you can create a cluster of virtual machines using either Windows or Linux that use the Intel Message Passing Library (MPI) to spread the workload of simulations and computations among compute nodes of many virtual machines.

Azure Batch



Microsoft Azure Batch is a fully-managed cloud service that provides job scheduling and compute resource management for developers in organizations, independent software vendors, and cloud service providers. Both new and existing high-performance computing (HPC) applications running on workstations and clusters today can be readily enabled to run in Azure at scale, and with no on-premises infrastructure required. Common application workloads include image and video rendering, media transcoding, engineering simulations, Monte Carlo simulations, and software test execution, among others; all highly parallel, computationally intensive workloads that can be broken into individual tasks for execution. With Azure Batch, you can scale from a few VMs, up to tens of thousands of VMs, and run the most massive, most resource-intensive workloads.

Azure Batch

Azure Batch is a service that manages Virtual Machines for large-scale parallel and high-performance computing (HPC) applications. Batch is a Platform as a Service (PaaS) offering that manages the VMs necessary for your compute jobs for you instead of forcing you to manage an HPC cluster or job schedule. Batch provides auto-scaling functionality and job scheduling functionality in addition to managing compute nodes.

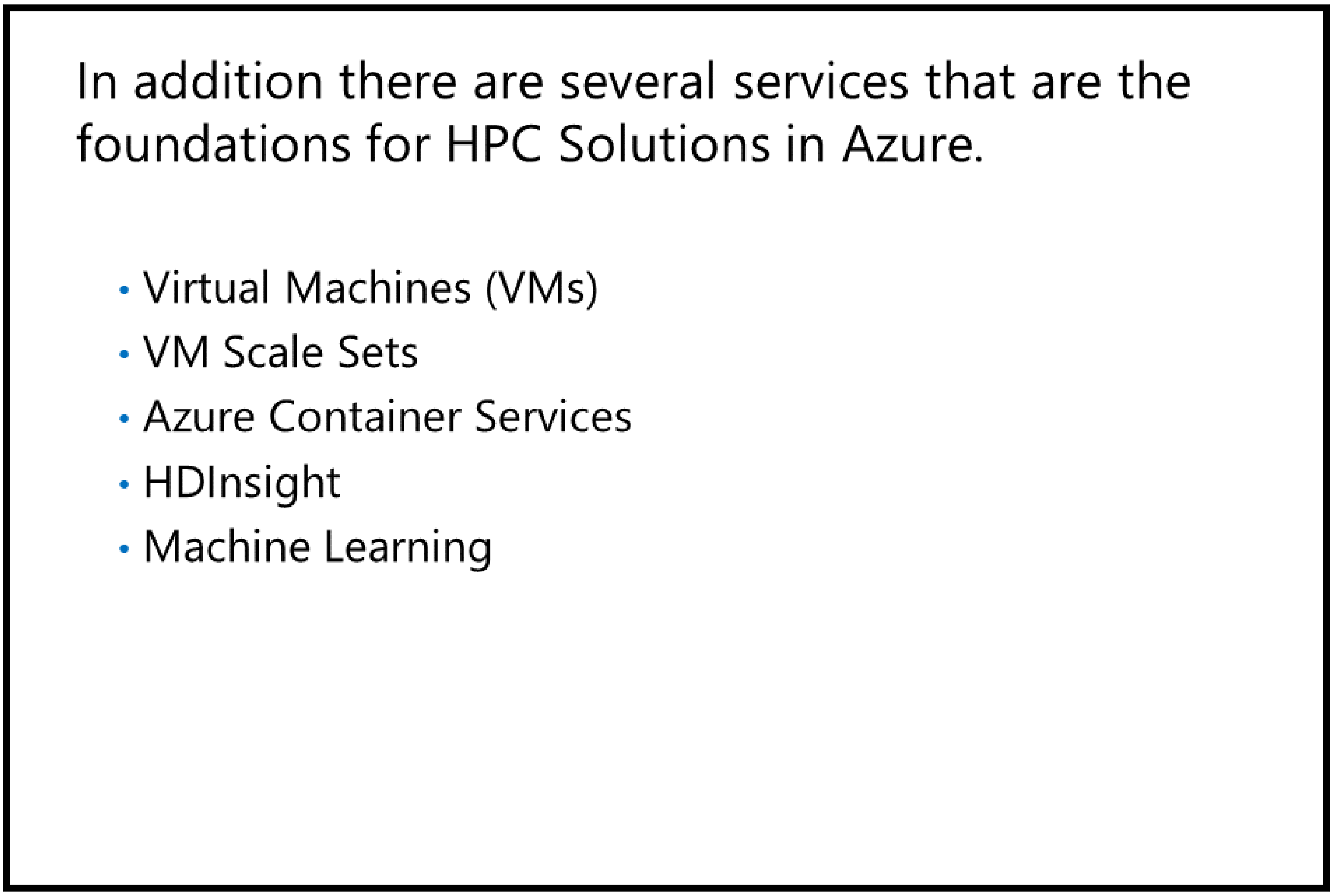
Batch computing is a familiar pattern for organizations that process, transform, and analyze large amounts of data, either on a schedule or on-demand. It includes end-of-cycle processing such as a bank’s daily risk reporting or a payroll that must be done on schedule. It also includes large-scale business, science, and engineering applications that typically need the tools and resources of a compute cluster or grid. Applications include traditional HPC applications such as fluid dynamics simulations as well as specialized workloads in fields ranging from digital content creation to financial services to life sciences research. Batch works well with intrinsically parallel (sometimes called "embarrassingly parallel") applications or workloads, which lend themselves to running as parallel tasks on multiple computers.

Scaling out Parallel Workloads

The Batch API can handle scale out of an intrinsically parallel workload such as image rendering on a pool of up to thousands of compute cores. Instead of having to set up a compute cluster or write code to queue and schedule your jobs and move the necessary input and output data, you automate the scheduling of large compute jobs and scale a pool of compute VMs up and down to run them. You can write client apps or front-ends to run jobs and tasks on demand, on a schedule, or as part of a larger workflow managed by tools such as Azure Data Factory.

You can also use the Batch Apps API to wrap an existing application, so it runs as a service on a pool of compute nodes that Batch manages in the background. The application might be one that runs today on client workstations or a compute cluster. You can develop the service to let users offload peak work to the cloud, or run their work entirely in the cloud. The Batch Apps framework handles the movement of input and output files, the splitting of jobs into tasks, job and task processing, and data persistence.

Stateless Component Workloads



In addition to the formal HPC services above, Azure provides non-managed solutions and related services to provide a degree of HPC using IaaS.

These include:

- Virtual Machines
- VM scale Sets
- Azure Container Services
- HDInsight
- Machine Learning

And plenty more. The inbuilt massively scalable Azure fabric provides a perfect platform to deploy parallel compute tasks on several services. The use of Azure Resource manager to automate deployment and auto scale based on various criteria enables the developer and architect to be creative with their application solutions.

As an example, deploying VMSS to manage an HPC workload by varying the number of VMs based on the queue length would combine auto-scale based on a custom metric with HPC Pack and templated infrastructure deployment. You do not have to rely on the pre defined HPC services.

Lesson 3: Migration

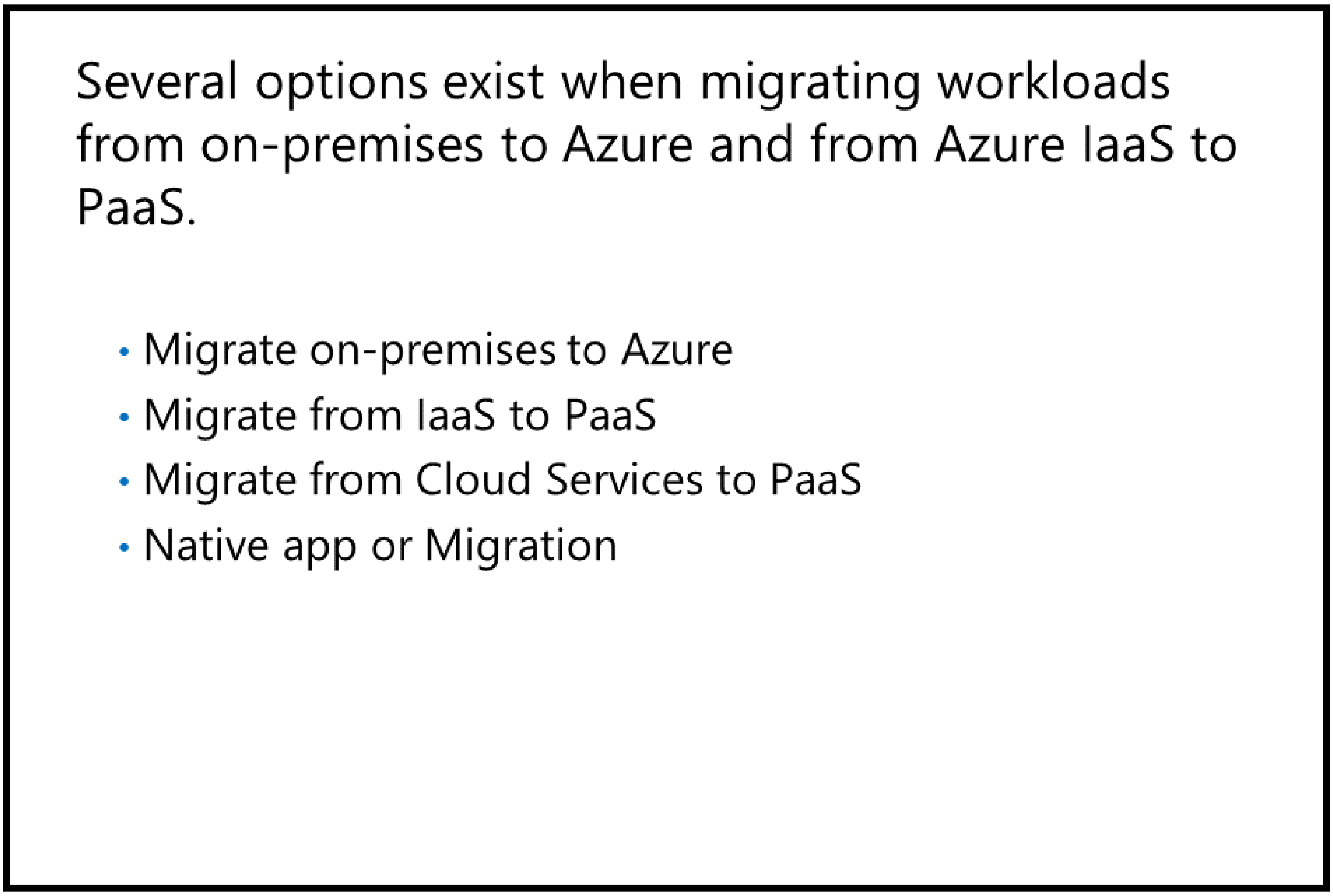
Migrating from on-premises workloads into Azure can be achieved in several ways that depend upon the individual workloads and the level of rewriting that the user is prepared or able to achieve. This lesson covers the Azure Migrate preview service and strategies for migrating from on-premises to Azure IaaS, and PaaS solutions. The ability to convert cloud service roles into Azure Service Fabric microservices is also discussed.

Lesson objectives

After completing this lesson, you will be able to:

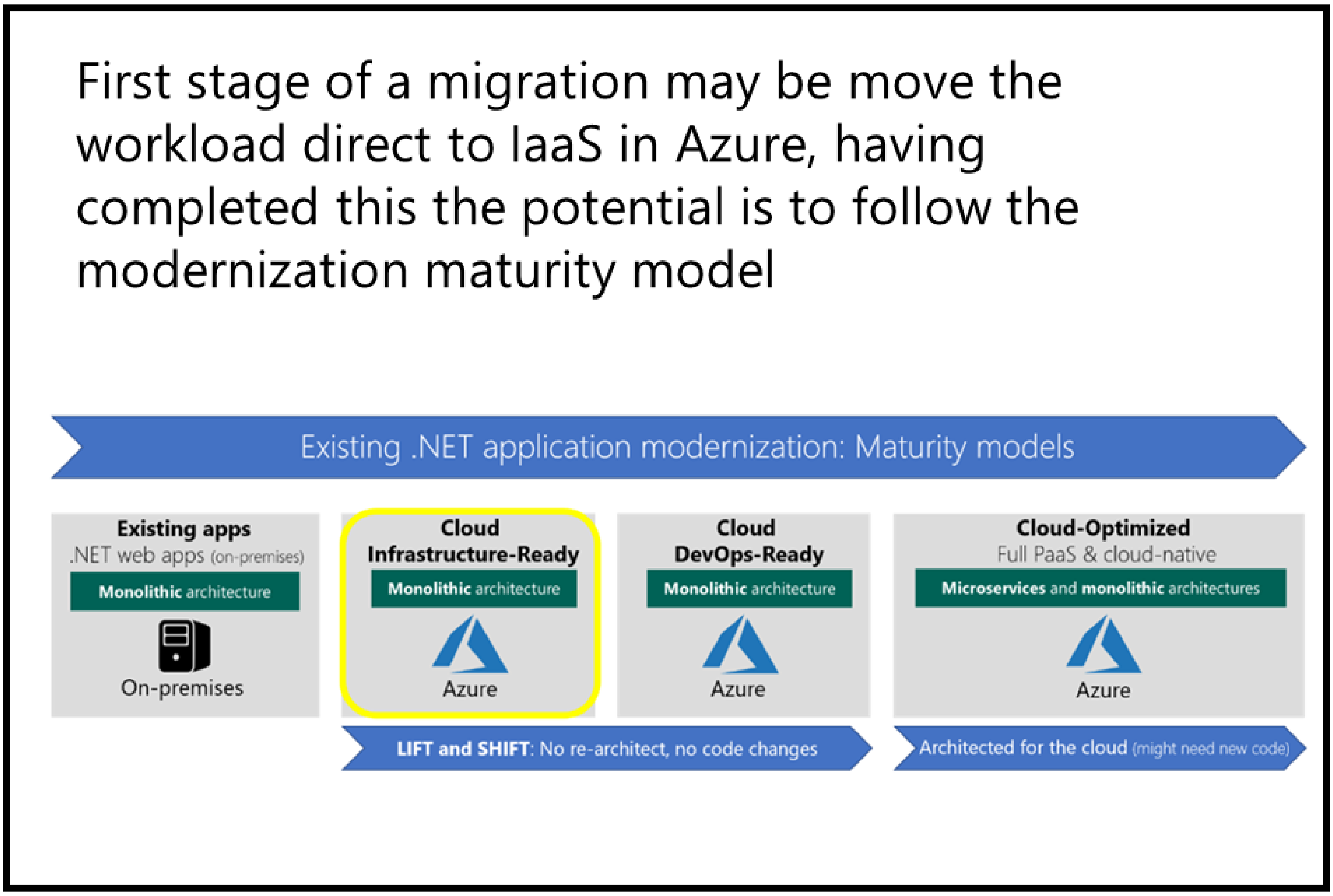
- Describe Azure Migrate.
- Understand the process to migrate from IaaS to PaaS.
- Describe the best way to migrate apps to Azure.
- Use Azure Migrate to plan a single VM migration.

Migration



The challenge of migrating workloads and applications to a cloud-based or hybrid platform or even from converting from cloud-based IaaS to a PaaS solution is one that causes much concern when adopting a public cloud strategy. There are several options for beginning the ‘journey to the cloud.’ For moving workloads, the ‘lift and shift’ approach provides all the benefits of pay as you go computing without the potential headaches of rewriting application code or re-architecting the application to fit a specific cloud application pattern.

On-Premises Lift and Shift

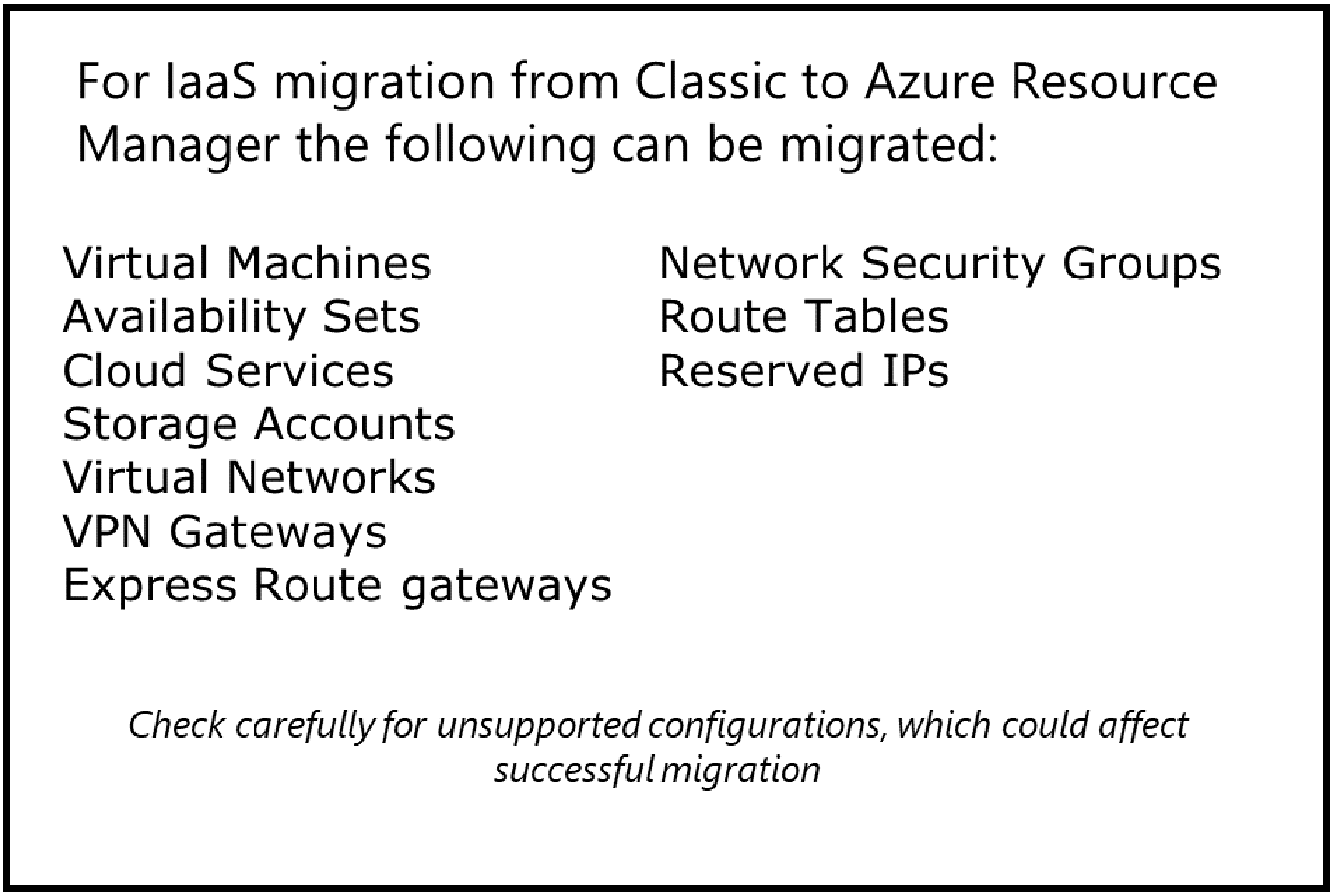


A new service from Microsoft, the Azure Migrate service provides a discovery and assessment tool. Azure Migrate assesses suitability for migration and ensures that sizing is correct for the performance of the VM. An estimate of the cost of running your VM in Azure is also available; this service is designed for those considering a lift and shift migration or beginning to plan their migration to the cloud.

Migrate provides the ability to visualize dependencies of a specific VM or for all VMs in a group. The service currently only provides assessment for VMWare VMs but soon will also include Hyper-V VMs.

The lift and shift approach is an ideal method to begin the migration of workloads to modernization. Commencing with lift and shift, moving through the phases shown below to cloud-optimized taking advantage of the PaaS services and cloud native applications. The latter often requires the applications to be rewritten.

Migration from Classic IaaS

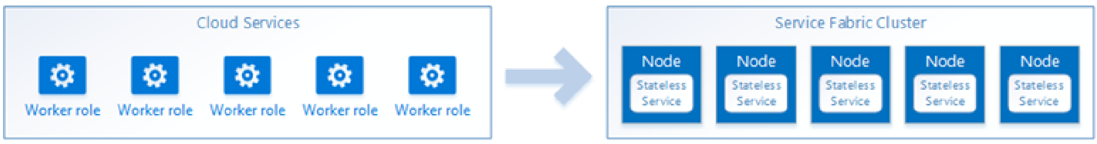


If there are resources in an Azure subscription based on the classic or Azure Service Manager model, it is now possible to migrate them to an Azure Resource Manager (ARM) deployment. The following resources can be migrated to ARM from ASM:

- Virtual Machines
- Availability Sets
- Cloud Services
- Storage Accounts
- Virtual Networks
- VPN Gateways
- Express Route gateways
- Network Security Groups
- Route Tables
- Reserved IPs

Migration from Cloud Services

Migrating Web and Worker roles in a cloud service to Azure Fabric stateless services is the simplest method of migration to Service Fabric.



When migrating cloud services to a PaaS solution, it is necessary to consider the difference between VMs, workloads, and applications in each model.

A cloud service deploys applications as VMs; code is connected to a VM instance which might be a Web role or a worker role. To scale the application, more VMs are deployed.

The deployment package contains the web role and worker role definition and specifies the instance count for each role; an instance is a VM hosting that role.

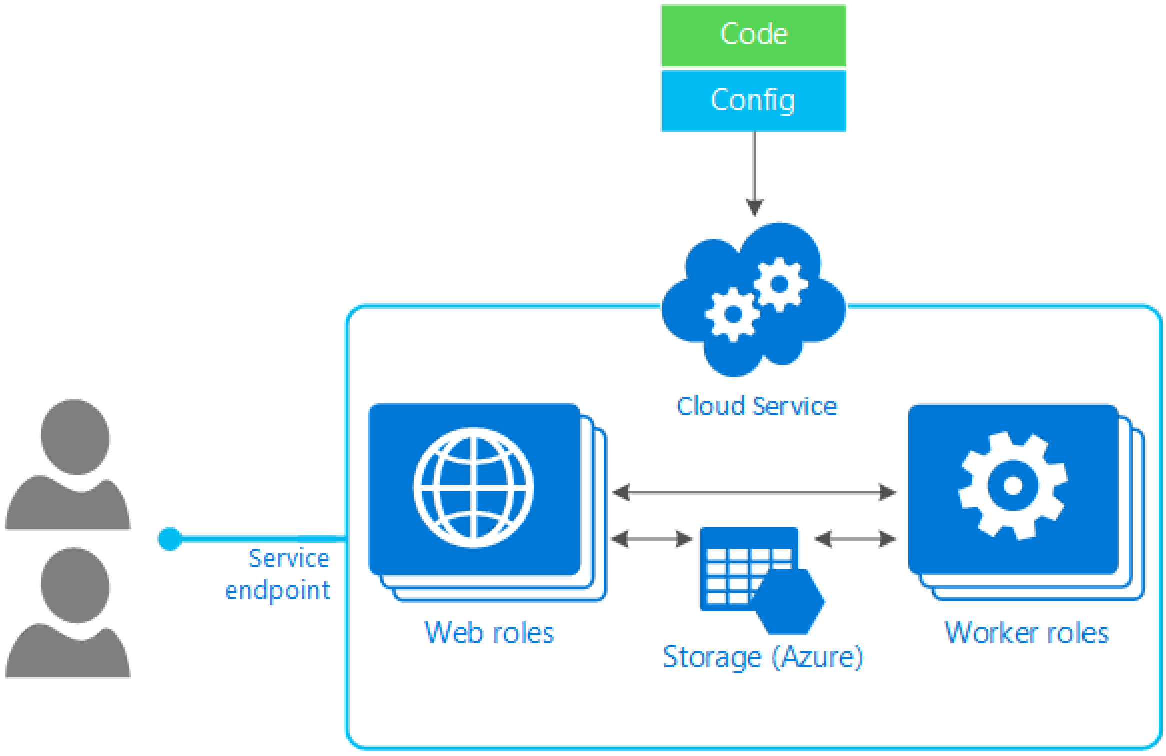


FIGURE 4.1: DEPLOYMENT PACKAGE

Migrating a cloud service to Service Fabric switches to deploying applications to VMs that are already running Service Fabric either on Windows or Linux. The applications or services that are deployed are entirely unrelated to the VM infrastructure. The service fabric application platform hides the VM layer from the application. This also allows multiple applications to be deployed to a service fabric cluster.

The architecture of applications running on the two services is fundamentally different.

A cloud service application will typically have external dependencies which manage the data and state of an application and the method of communicating between web and worker roles.

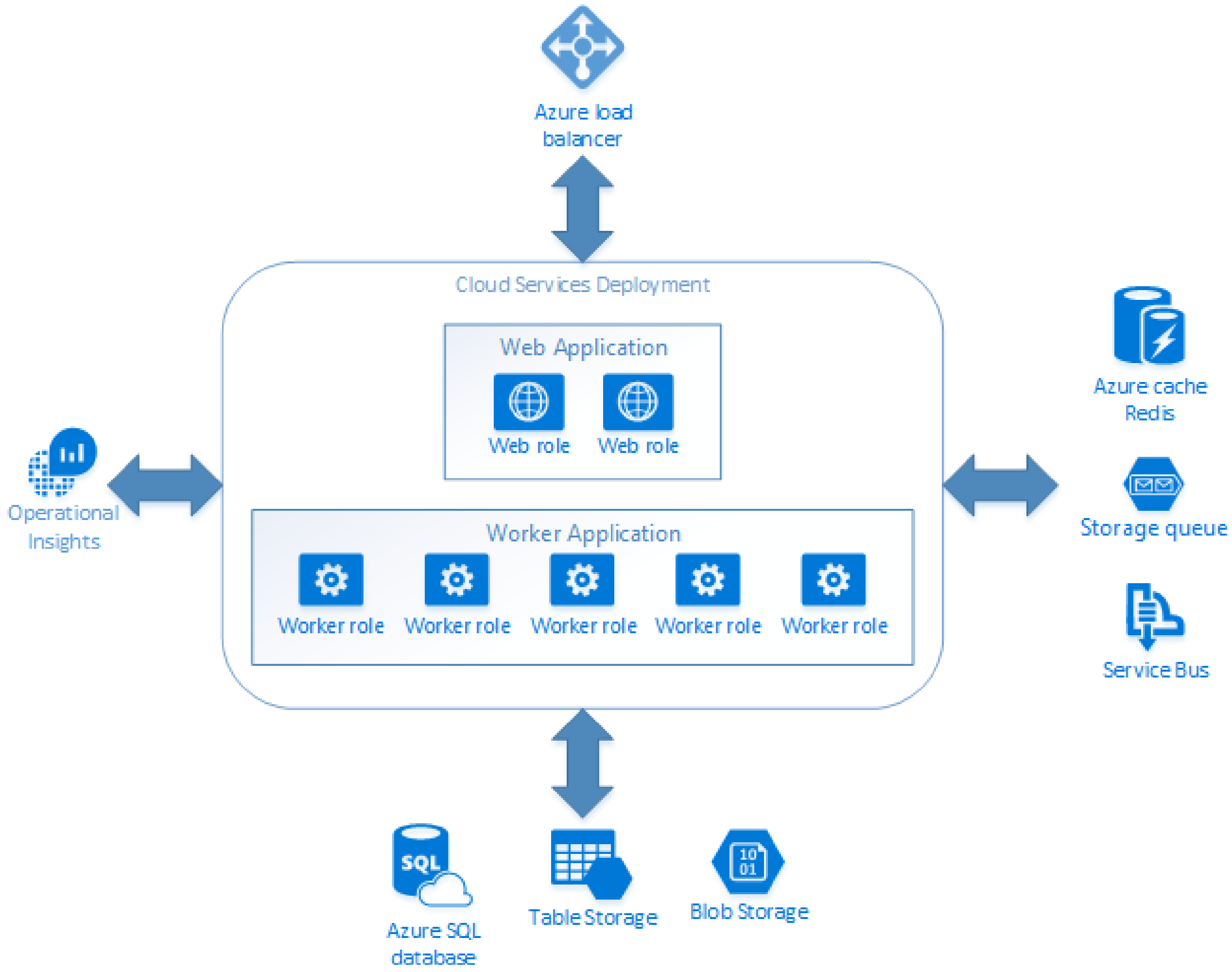


FIGURE 4.2: CLOUD SERVICE SOLUTION

A Service fabric application can also rely on the same external service dependencies. The quickest and easiest way to migrate a Cloud Service application to service fabric is to merely convert the Web roles and worker roles to stateless services whilst keeping the architecture the same.

If the aim is to remove the external dependencies and take full advantage of the ability to unify deployment, management and upgrade models, then state-full services would be required which will mean full code and application rewrites.

Migration both to the cloud in general as a lift and shift and to modern cloud-native applications is a complicated and time-consuming project.

Lab: Deploying Managed Server Workloads to Azure

Scenario

The latest client has internally developed their new web application using Docker. The developers on the team have created a Docker container image as their build output and wished to host this image in Azure.

Objectives

- Deploy an Azure Container Service cluster.
- Deploy an image to the cluster.

Lab setup

Estimated Time: 60 minutes

Virtual machine: **20535A-SEA-ARCH**

User name: **Admin**

Password: **Pa55w.rd**

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Create Azure Container Service Cluster

Exercise 2: Docker Image

Exercise 3: Cleanup Subscription

Review Question(s)

Module review and takeaways

Review Question(s)