

Book two

Introduction to Azure Kubernetes Service

Creation date
09/29/2020

Revision date
09/29/2020

Corporate Headquarters
Microsoft Corporation
1 Microsoft Way,
Redmond, WA 98052
www.microsoft.com

Submission/Revision History

Revision	Authors	Released date	Comments
1.0	Mauricio Zaragoza	09/29/2020	Initial release

Technical Review History

Examine	Examiner (s)	Date	Comments
1.0 Within group	Reggie Snead Bhavin Doshi	09/29/2020	Comments
2.0 Outside group		09/29/2020	Comments

Table of contents

Submission/Revision History2

Technical Review History2

Table of contents3

Introduction to Azure Kubernetes Service5

 Learning Objectives 5

Definitions.....5

Conventions5

Document updates6

Introduction8

Learning objectives8

Prerequisites8

.....9

What is Azure Kubernetes Service?..... 10

What is a container? 10

Why use a container? 10

What is container management? 11

What is Kubernetes? 12

What is Azure Kubernetes Service?..... 13

..... 14

How Azure Kubernetes Service works 15

Creating an AKS cluster..... 15

How workloads are developed and deployed to AKS	16
Azure Dev Spaces	16
Deployment Center	17
Azure Service Integration	17
.....	18
When to use Azure Kubernetes Service	19
Knowledge check	22
Responses	23
Summary	24
References	24

Introduction to Azure Kubernetes Service

Determine the types of business problems that can be solved using Azure Kubernetes Service. Describe the benefits of container orchestration with features like deployment management, automatic updates, and self-healing.

Learning Objectives

In this module, you will:

- Evaluate whether Azure Kubernetes Service is an appropriate Kubernetes orchestration platform for you
- Describe how the components of Azure Kubernetes Service work to support compute container orchestration



Definitions

Table 1: Terms and Acronyms used in this document

Term	Acronym	Definition	Comments

Conventions

Table 2: Key Terms and Symbols used in this document

Term	Definition
Note	Notes give you additional information that will help you obey the instructions written in the procedural steps.
	Red squares or rectangles shows you where to enter and / or to examine the necessary information (data, IP addresses, etc.).
Bold	Text written in bold in procedural steps indicate data that must be typed and / or menu selections that must be selected or clicked on with the mouse pointer.
	Red arrows indicate where to point and click your mouse pointer. Reference to these in screenshots and other parts of the document.

Document updates

This document contains information about Microsoft Azure and other Microsoft technologies. Due to the dynamics of the modern world, these technologies are evolving and changing all the time.

Microsoft is adding and deprecating features to Azure every day, increasing quotas, changing limits, releasing new products, changing, and improving architectures, etcetera.

In general, Microsoft is modifying the characteristics of the Azure services to accommodate customers' requirements and needs quite often. For this reason, the details about the features and products contained in this document might be enhanced or changed at some point in time.

Most of the concepts that are outlined in this guide are general concepts that are not expected to change drastically soon.

You need to be aware of the dynamics of Azure and use this document as reference. Please complement the information in this guide with the most current Azure documentation that is available online.

Values, limits, names, capabilities, features, costs, billing models, regions, etc. are valid at the time this document was published.

The document release date is in the **Revision** section, and a comprehensive list of online resources are published in the **References** section. These 2 portions need to be modified accordingly when Microsoft updates any of the functionalities described in this white paper.



Introduction

The use of containers to develop and deploy software has become popular over the last few years. Containers make it easy to package and deploy an application with all its services to any compute environment. When your application meets higher demand, you can easily scale out your services by deploying additional container instances. Containers are also less resource intensive than virtual machines. This efficiency allows you to make better use of compute resources that will save you money.

The standard container management runtime is focused on managing individual containers. However, there are times where you'll want to scale and have multiple containers working together. Scaling multiple containers becomes challenging as several factors need consideration when managing multiple containers. Suppose you need to handle load balancing, security, network connectivity, and deployment. To help make this process easier, it's common to use a container management platform such as Kubernetes.

Suppose you work at a fleet management company. Your company provides an asset tracking solution to customers worldwide. Your tracking solution is built and deployed as microservices. You're using containerized instances to quickly deploy into new customer regions and scale resources as needed to meet customer demands. You want to use a container orchestration platform that makes it simple to develop, deploy, and manage containerized applications.

Here, you'll see how Azure Kubernetes Service (AKS) makes it simple to manage a hosted Kubernetes environment in Azure. The goal is to help you decide if AKS is a good choice as a Kubernetes platform for your business.

Learning objectives

In this module, you will:

- Evaluate whether Azure Kubernetes Service is an appropriate Kubernetes orchestration platform for you
- Describe how the components of Azure Kubernetes Service work to support compute container orchestration

Prerequisites

- Basic understanding of microservices

Part 1

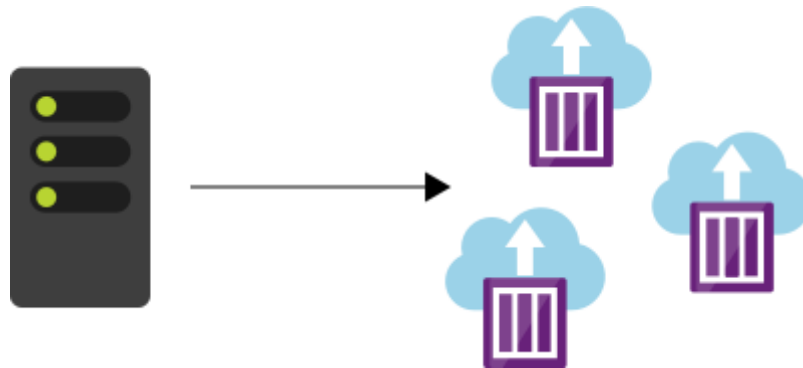
What is Azure
Kubernetes Service?

What is Azure Kubernetes Service?

Let's start with a few definitions and a quick tour through Azure Kubernetes Service (AKS). This overview should help you decide whether AKS might be a good fit for your containerization management strategy.

What is a container?

A *container* is an atomic unit of software that packages up code, dependencies, and configuration for a specific application. Containers allow us to split up monolithic applications into individual services that make up the solution. This rearchitecting of our application will enable us to deploy these separate services via containers.

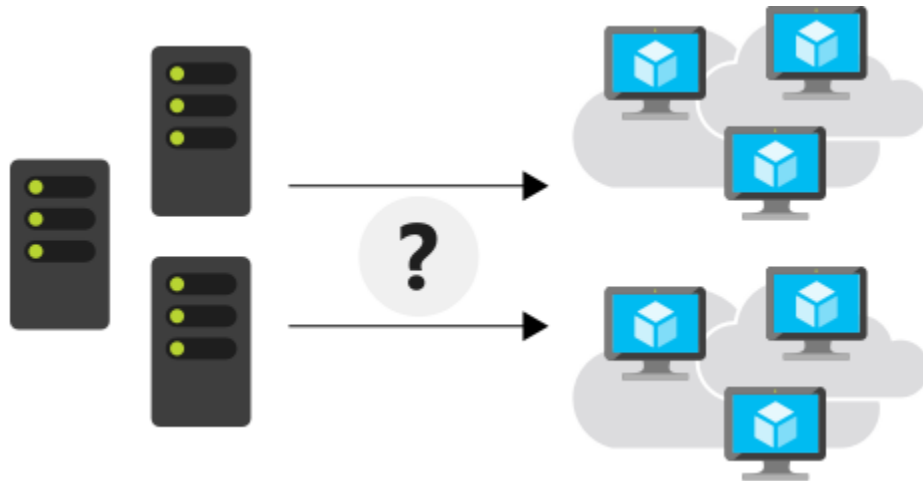


Why use a container?

Suppose your fleet management solution included three major applications:

- A fleet tracking website that includes maps and information about the assets being tracked.
- A data processing service that collects and processes information sent from tracked vehicles.
- An MSSQL database for storing tracking and user information captured from the website.

You realize that you have to scale out your solution to meet customer demand. One option is to deploy a new virtual machine (VM) for each application and then deploy the applications to the VMs. However, doing so will make you responsible for the management of each additional VM. For example, you'll have to make sure the correct operating system (OS) versions and dependencies for each application is installed and configured. You also must make sure you're installing and upgrading the correct versions of the applications. If there are errors, you have to make sure you can roll back the installation with the least amount of disruption to your solution.



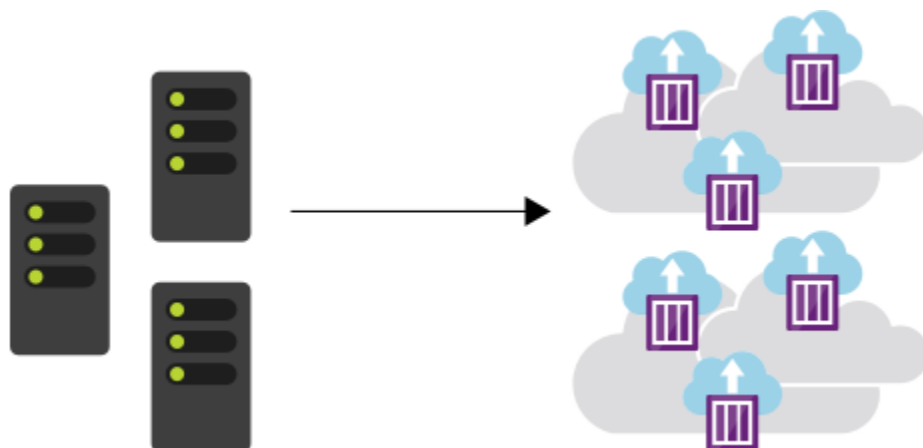
The above deployment is cumbersome, sometimes error-prone and doesn't easily scale single services. For example, you might find you can't easily scale only the caching service used in the web application. Containers help solve these types of problems.

The container concept gives us three major benefits:

1. **A container is immutable** - the unchanging nature of a container allows it to be deployed and run reliably with the same behavior from one compute environment to another. A container image tested in a QA environment is the same container image deployed to production.
2. **A container is lightweight** - you can think of a container as a VM image, but smaller. A VM image is normally installed on a physical host. The image contains both the OS and the application you want to run. In contrast, a container doesn't need an OS, only the application. The container always relies on the host installed OS for Kernel-specific services. Containers are less resource-intensive, and multiple containers can be installed on the same compute environment.
3. **Container startup is fast** - containers can start up in few seconds instead of minutes, like a VM.

The above benefits make containers a popular choice for developers and IT operations alike.

What is container management?



Even though you can think of containers as VMs, you have to keep in mind that they aren't. A container has a distinct life cycle. It's deployed, started, stopped, and destroyed as requested. This life cycle makes containers disposable and impacts how developers and IT operations should think about the management of large container deployments.

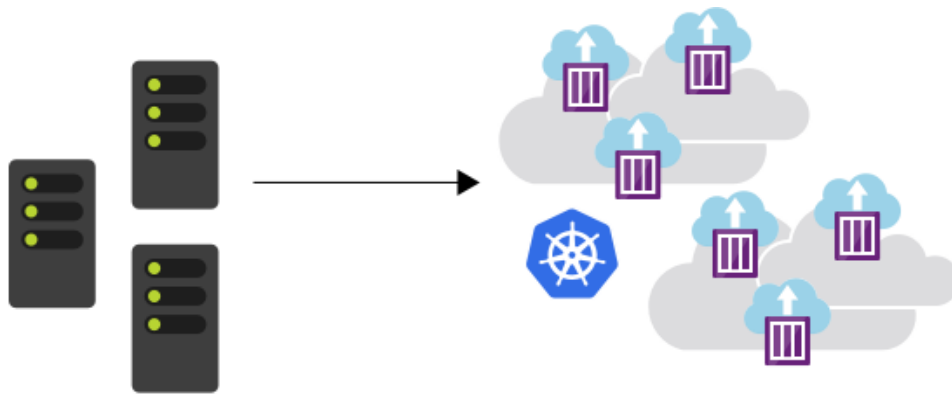
The process of deploying, updating, monitoring, and removing containers introduces many challenges.

Suppose you want to scale your fleet tracking website. You find that at specific times during the day, you need more instances of the site's caching service to manage performance. You can solve this problem by adding additional caching service containers.

Now let's suppose, you need to roll out a new version of your caching service. How do you make sure you update all the containers? How do you remove all the older versioned containers?

These types of questions justify a system to help you manage your container deployment.

What is Kubernetes?



Kubernetes is a portable, extensible open-source platform for automating deployment, scaling, and the management of containerized workloads. Kubernetes abstracts away complex container management and provides us with declarative configuration to orchestrate containers in different compute environments. This orchestration platform gives us the same ease of use and flexibility as with Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) offerings.

Kubernetes allows you to view your data center as one large computer. We don't worry about how and where we deploy our containers, only about deploying and scaling our applications as needed.

However, this view might be slightly misleading as there are a few aspects to keep in mind:

- Kubernetes isn't a full PaaS offering. It operates at the container level and offers only a common set of PaaS features.
- Kubernetes isn't monolithic. It's not a single application that is installed. Aspects such as deployment, scaling, load balancing, logging, and monitoring are all optional. You're responsible for finding the best solution that fits your needs to address these aspects.
- Kubernetes doesn't limit the types of applications that can run. If your application can run in a container, it can run on Kubernetes. Your developers need to understand concepts such as microservices architecture, to make optimal use of container solutions.
- Kubernetes doesn't provide middleware, data-processing frameworks, databases, caches, nor cluster storage systems. All these items are run as containers or as part of another service offering.

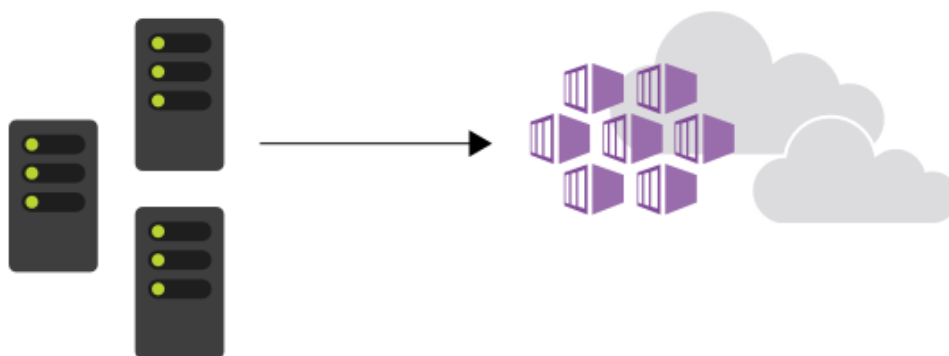
- A Kubernetes deployment is configured as a cluster. A cluster consists of at least one master machine and one or more workers machines. For production deployments, the preferred master configuration is a multi-master high availability deployment with three to five replicated masters. These machines can be physical hardware or VMs. These worker machines are called nodes or agent nodes.

With all the benefits you receive with Kubernetes, keep in mind that you're responsible for maintaining your Kubernetes cluster. For example, you need to manage OS upgrades and the Kubernetes installation and upgrades. You also manage the hardware configuration of the host machines, such as networking, memory, and storage.

⚠ Note

Kubernetes is sometimes abbreviated to **K8s**. The 8 represents the eight characters between the K and the s of the word K[ubernete]s.

What is Azure Kubernetes Service?



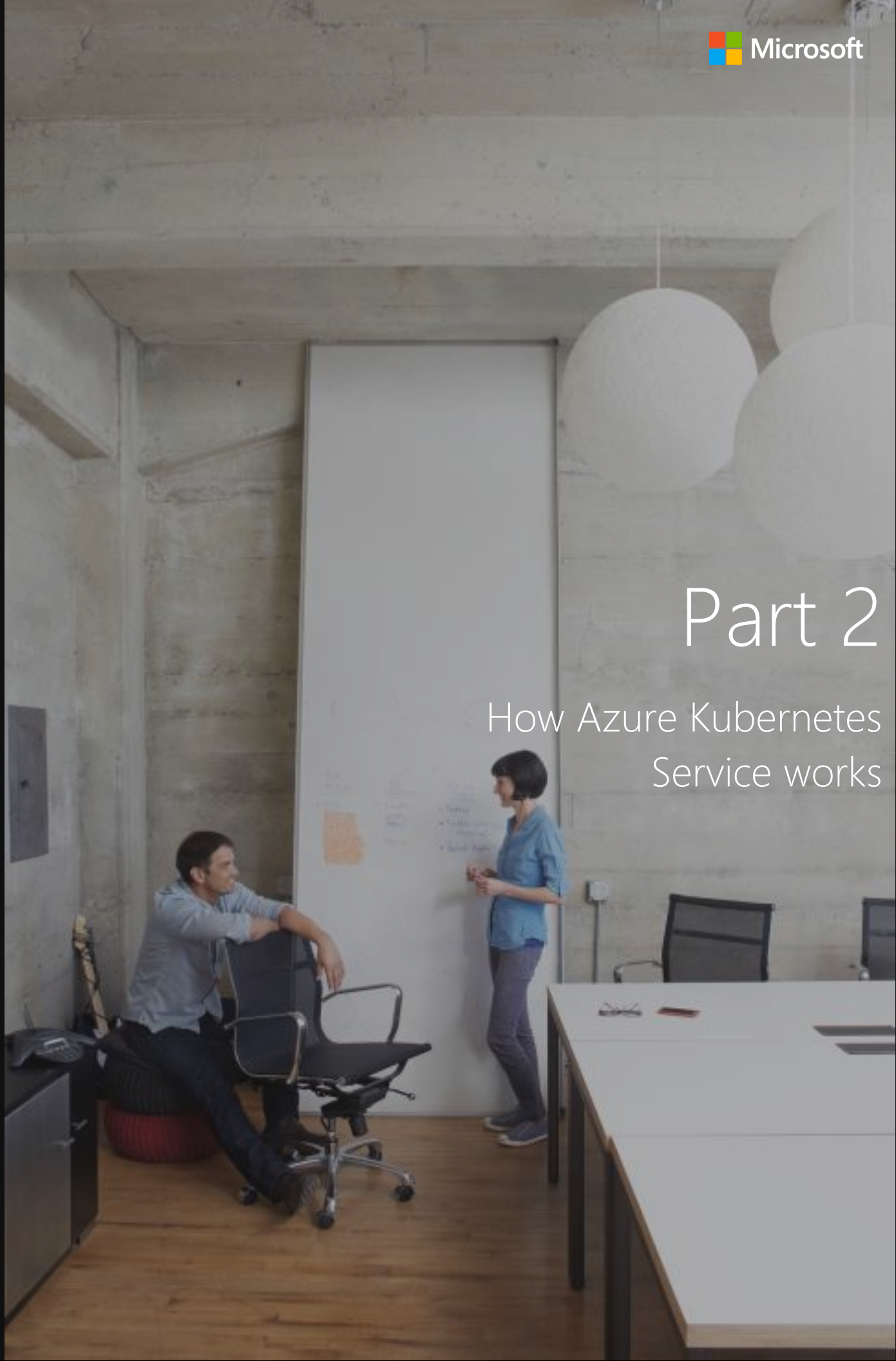
Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment and makes it simple to deploy and manage containerized applications in Azure. Your AKS environment is enabled with features such as automated updates, self-healing, and easy scaling. The Kubernetes cluster master is managed by Azure and is free. You manage the agent nodes in the cluster and only pay for the VMs on which your nodes run.

You can either create your cluster in the Azure portal or use the Azure CLI. When you create the cluster, you can use Resource Manager templates to automate cluster creation. With these templates, you specify features such as advanced networking, Azure Active Directory (AD) integration, and monitoring. This information is then used to automate the cluster deployment on your behalf.

With AKS, we get the benefits of open-source Kubernetes without the complexity or operational overhead compared to running our own custom Kubernetes cluster.

Part 2

How Azure Kubernetes
Service works



How Azure Kubernetes Service works

Now that you know the basics of Azure Kubernetes Service, let's see what information you need to set up a simple AKS cluster. This information should help you decide how much additional configuration AKS may require when integrating with existing development and deployment processes.

Creating an AKS cluster

At its core, an AKS cluster is a cloud hosted Kubernetes cluster. Unlike a custom Kubernetes installation, AKS streamlines the installation process and takes care of most of the underlying cluster management tasks.

You have two options when you create an AKS cluster. You either use the Azure portal or Azure CLI. Both options require you to configure basic information about the cluster. For example:

- The Kubernetes cluster name
- The version of Kubernetes to install
- A DNS prefix to make the master node publicly accessible
- The initial node pool size

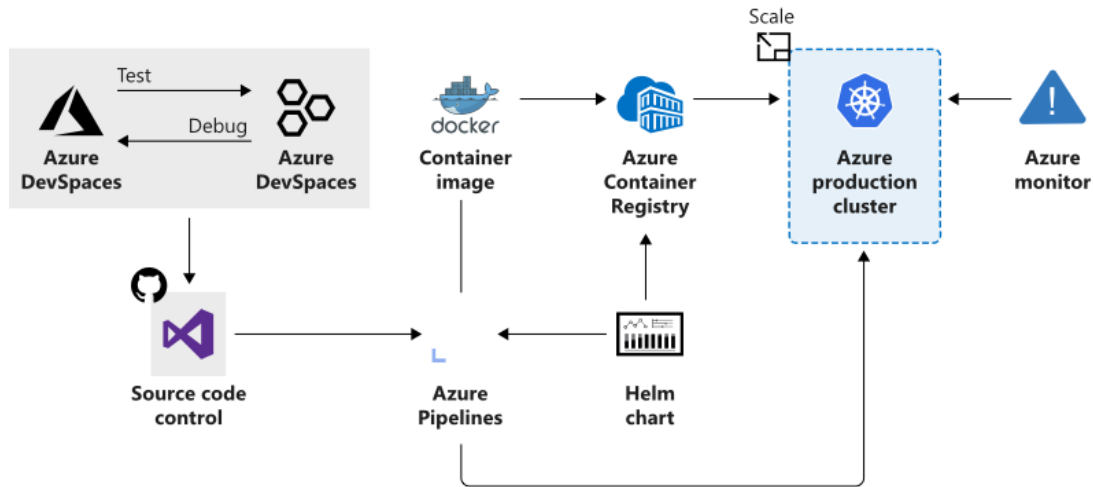
The initial node pool size defaults to two nodes, however it's recommended that at least three nodes are used for a production environment.

ⓘ Note

The master node in your cluster is free. You only pay for node VMs, storage and networking resources consumed in your cluster.

Unless specified, the Azure service creation workflow creates a Kubernetes cluster using default configuration for scaling, authentication, networking and monitoring. Creating an AKS cluster typically takes a few minutes. Once complete, you can change any of the default AKS cluster properties. Access and management of your cluster is done through the Azure portal or from the command line.

How workloads are developed and deployed to AKS



AKS supports the Docker image format that means that you can use any development environment to create a workload, package the workload as a container and deploy the container as a Kubernetes pod.

Here you use the standard Kubernetes command-line tools or the Azure CLI to manage your deployments. The support for the standard Kubernetes tools ensures that you don't need to change your current workflow to support an existing Kubernetes migration to AKS.

AKS also supports all the popular development and management tools such as Helm, Draft, Kubernetes extension for Visual Studio Code and Visual Studio Kubernetes Tools.

Azure Dev Spaces

Setting up a local Kubernetes cluster on a developer machine can be complex and most solutions offers a single node configuration. It's also common to mock or replicate dependencies between developer teams when working on microservices projects.

Azure Dev Spaces helps your development teams be more productive on Kubernetes and allows you to:

- Minimize the local dev machine setup for each team member as developers can work directly in AKS
- Rapidly iterate and debug code directly in Kubernetes using Visual Studio or Visual Studio Code
- Generate Docker and Kubernetes configuration-as-code assets to use from development through to production
- Develop your code in isolation, and do integrated testing with other components without replicating or mocking up dependencies

⚠ Important

Azure Dev Spaces is supported only by AKS clusters in specific regions.

Deployment Center

Deployment center simplifies setting up a DevOps pipeline for your application. You can use this configured DevOps pipeline to set up a continuous integration (CI) and continuous delivery (CD) pipeline to your AKS Kubernetes cluster.

With Azure DevOps Projects you can:

- Automatically create Azure resources, such as an AKS cluster
- Create an Azure Application Insights resource for monitoring an AKS cluster
- Enable Azure Monitor for containers to monitor performance for the container workloads on an AKS cluster

You can add richer DevOps capabilities by extending the default configured DevOps pipeline. For example, you can add approvals before deploying, provision additional Azure resources, run scripts or upgrade workloads.

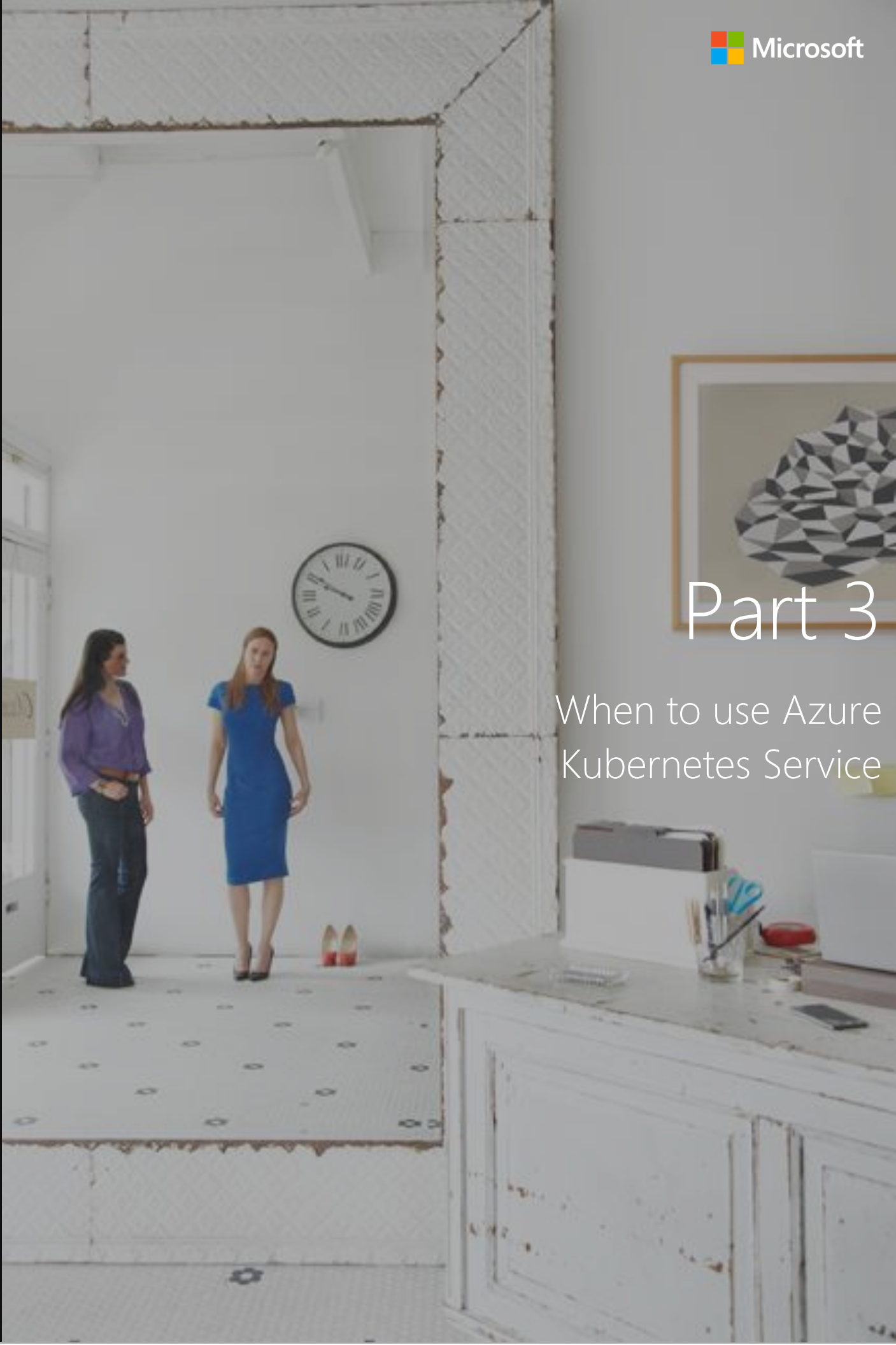
Azure Service Integration

AKS allows us to integrate any Azure service offering and use it as part of an AKS cluster solution.

For example, remember that Kubernetes doesn't provide middleware and storage systems. Suppose you need to add a processing queue to the fleet management data processing service. You can easily integrate Storage queues using Azure Storage to extend the capacity of the data processing service.

Part 3

When to use Azure
Kubernetes Service



When to use Azure Kubernetes Service

Here, we'll discuss how you can decide whether Azure Kubernetes Service (AKS) is the right choice for you.

You'll either approach your decision from a green fields or a lift-and-shift project point of view. A green fields project will allow you to evaluate AKS based on default features. A lift-and-shift project will force you to look at which features are best suited to support your migration.

We saw earlier that there are several features that enhance the AKS Kubernetes offering. Each of these features can be a compelling factor in your decision to use AKS.

Identity and security management	Do you already use existing Azure resources and make use of Azure AD? You can configure an AKS cluster to integrate with Azure AD and reuse existing identities and group membership.
Integrated logging and monitoring	AKS includes Azure Monitor for containers to provide performance visibility of the cluster. With a custom Kubernetes installation, you normally decided on a monitoring solution that requires installation and configuration.
Auto Cluster node and pod scaling	Deciding when to scale up or down in large containerization environment is tricky. AKS supports two auto cluster scaling options. You can use either the horizontal pod autoscaler or the cluster autoscaler to scale the cluster. The horizontal pod autoscaler watches the resource demand of pods and will increase pods to match demand. The cluster autoscaler component watches for pods that can't be scheduled because of node constraints. It will automatically scale cluster nodes to deploy scheduled pods.
Cluster node Upgrades	Do you want to reduce the number of cluster management tasks? AKS manages Kubernetes software upgrades and the process of cordoning off nodes and draining them to minimize disruption to running applications. Once done, these nodes are upgraded one by one.
GPU enabled Nodes	Do you have compute-intensive or graphic-intensive workloads? AKS supports GPU enabled node pools.
Storage volume Support	Is your application stateful, and does it require persisted storage? AKS supports both static and dynamic storage volumes. Pods can attach and reattach to these storage volumes as they're created or rescheduled on different nodes.
Virtual network Support	Do you need pod to pod network communication or access to on-premise networks from your AKS cluster? An AKS cluster can be deployed into an existing virtual network with ease.
Ingress with	Do you need to make your deployed applications publicly available?

(cont.) HTTP application routing support	(cont.) The HTTP application routing add-on makes it easy to access AKS cluster deployed applications.
Docker image support	Do you already use Docker images for your containers? AKS by default supports the Docker file image format.
Private container registry	Do you need a private container registry? AKS integrates with Azure Container Registry (ACR). You aren't limited to ACR though, you can use other container repositories, public, or private.

All the above features are configurable either when you create the cluster or following deployment.

Knowledge check

Knowledge check
and summary

Knowledge check

1. Suppose you work for a company that builds a Massively Multiplayer Online (MMO) game. You decide to move all your services to Azure Kubernetes service. Which of the following components will contribute to your monthly Azure charge?

- a. The Master node
- b. Per deployed Pod
- c. Per node VM

2. Suppose you work for an event management company. Your events website needs persisted storage. Which feature of AKS will you use?

- a. Use Azure Container Registry
- b. Storage volume support
- c. Enable the horizontal pod autoscaler

3. Suppose you work for a company that monitors weather conditions. Your team of developers needs to do end-to-end testing with other components in your AKS cluster. The team wants to do testing without replicating or mocking up dependencies. Which service should they use?

- a. Deployment Center
- b. Draft
- c. Azure Dev Spaces

Responses

1. Suppose you work for a company that builds a Massively Multiplayer Online (MMO) game. You decide to move all your services to Azure Kubernetes service. Which of the following components will contribute to your monthly Azure charge?

- a. The Master node
- b. Per deployed Pod
- c. **Per node VM**
You only pay for the virtual machines instances, storage, and networking resources consumed by your Kubernetes cluster.

2. Suppose you work for an event management company. Your events website needs persisted storage. Which feature of AKS will you use?

- a. Use Azure Container Registry
- b. **Storage volume support**
AKS supports both static and dynamic storage volumes for containers that need persisted storage.
- c. Enable the horizontal pod autoscaler

3. Suppose you work for a company that monitors weather conditions. Your team of developers needs to do end-to-end testing with other components in your AKS cluster. The team wants to do testing without replicating or mocking up dependencies. Which service should they use?

- a. Deployment Center
- b. Draft
- c. **Azure Dev Spaces**
Azure Dev Spaces allows for the development of code in isolation, and do end-to-end testing with other components without replicating or mocking up dependencies

Summary

Our goal was to help you evaluate whether AKS would be a good choice as a Kubernetes orchestration platform for your business. We looked at several features that enhance the AKS Kubernetes offering. We saw how these features can help you decide if AKS is a good fit for new projects, or convince you to move from another Kubernetes solution, to Azure.

We also saw how AKS makes use of familiar concepts such as:

- The Docker file format for creating containers
- Popular development and management tools such as Helm, Draft, Kubernetes extension for Visual Studio Code and Visual Studio Kubernetes Tools
- Integration with Azure DevOps Projects to simplify setting up a DevOps pipeline for your application

You were looking for a container orchestration platform to quickly deploy your fleet management solution into new customer regions. AKS allows you to manage your hosted Kubernetes environment and makes it simple to develop, deploy, and manage containerized applications in Azure.

You don't need to manage any of the clusters infrastructures and you only pay for what you use. It streamlines the installation process and takes care of most of the underlying cluster management tasks.

Finally, AKS is part of Azure and can be integrated with other Azure services to extend your product or exchange data with other services.

References

- ✓ **Introduction to Azure Kubernetes Service**
<https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-kubernetes-service/>



Cloud Solution Architects

Customer Success

NorthEast Region

August 2020

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. Microsoft makes no warranties, express or implied, in this document.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2012 Microsoft Corporation. All rights reserved.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Microsoft, list Microsoft trademarks used in your white paper alphabetically are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.