# What is Azure Container Storage?

# Article

# 07/24/2024

# 5 contributors

# Feedback

# In this article

# Supported storage types

# Regional availability

# What's new in Azure Container Storage

# Why Azure Container Storage is useful

# Key benefits

# Glossary

# Next steps

# Show less

# Azure Container Storage is a cloud-based volume management, deployment, and orchestration service built natively for containers. It integrates with Kubernetes, allowing you to dynamically and automatically provision persistent volumes to store data for stateful applications running on Kubernetes clusters.

# Important

# Azure Container Storage isn't available in all Azure regions. See regional availability. Azure Container Storage is now generally available (GA) beginning with version 1.1.0. The GA version is recommended for production workloads. If you previously installed the Azure Container Storage preview and don't have auto-upgrade enabled, be sure to update to the GA version.

# To get started using Azure Container Storage, see Use Azure Container Storage with Azure Kubernetes Service or watch the video.

# This video provides an introduction to Azure Container Storage, an end-to-end storage management and orchestration service for stateful applications. Learn how to optimize the performance of stateful workloads on Azure Kubernetes Service (AKS) to effectively scale across storage services while providing a cost-effective, container-native experience.

# Supported storage types

# Azure Container Storage utilizes existing Azure Storage offerings for actual data storage and offers a volume orchestration and management solution purposely built for containers. You can choose any of the supported backing storage options to create a storage pool for your persistent volumes.

# Azure Container Storage offers persistent volume support with ReadWriteOnce access mode to Linux-based Azure Kubernetes Service (AKS) clusters. Supported backing storage options include block storage offerings only: Azure Disks, Ephemeral Disks (local NVMe or temp SSD), and Azure Elastic SAN (Preview). The following table summarizes the supported storage types, recommended workloads, and provisioning models.

# Expand table

# Storage type Description Workloads Offerings Provisioning model

# Azure Elastic SAN (Preview) Provision on demand, fully managed resource General purpose databases, streaming and messaging services, CD/CI environments, and other tier 1/tier 2 workloads. Azure Elastic SAN Provisioned on demand per created volume and volume snapshot. Multiple clusters can access a single SAN concurrently, however persistent volumes can only be attached by one consumer at a time.

# Azure Disks Granular control of storage SKUs and configurations Azure Disks are a good fit for tier 1 and general purpose databases such as MySQL, MongoDB, and PostgreSQL. Premium SSD, Premium SSD v2, Standard SSD, Ultra Disk Provisioned per target container storage pool size and maximum volume size.

# Ephemeral Disk Utilizes local storage resources on AKS nodes (NVMe or temp SSD) Ephemeral disk is extremely latency sensitive (low sub-ms latency), so it's best for applications with no data durability requirement or with built-in data replication support such as Cassandra. NVMe is available on storage optimized VM SKUs Deployed as part of the VMs hosting an AKS cluster. AKS discovers the available ephemeral storage on AKS nodes and acquires them for volume deployment.

# Feature support for different storage types

# Feature support depends on which backing storage option you select. The following table lists key features of Azure Container Storage and indicates which storage options support them.

# Expand table

# Feature Local NVMe Local SSD Azure Disks Azure Elastic SAN (Preview)

# Storage pool expansion/resize Supported Supported Supported Not supported

# Replication Supported Not supported Natively supported Preview

# Resource consumption Supported Supported Supported Preview

# SSE/CMK Not supported Not supported Supported Preview

# Expose ZRS option N/A N/A Supported Preview

# Persistent volumes Supported1 Supported1 Supported Preview

# Ephemeral volumes Supported Supported Supported Preview

# Snapshots Supported Supported Supported Not supported

# 1 For local NVMe and local SSD, Azure Container Storage uses generic ephemeral volumes by default, in which the data isn't persistent. However, you can update your Azure Container Storage installation to support the creation of persistent volumes from ephemeral disk storage pools.

# Regional availability

# Azure Container Storage is only available for a subset of Azure regions:

# (Africa) South Africa North

# (Asia Pacific) Australia East

# (Asia Pacific) East Asia

# (Asia Pacific) Japan East

# (Asia Pacific) Korea Central

# (Asia Pacific) Southeast Asia

# (Asia Pacific) Central India

# (Europe) France Central

# (Europe) Germany West Central

# (Europe) North Europe

# (Europe) West Europe

# (Europe) UK South

# (Europe) Sweden Central

# (Europe) Switzerland North

# (Middle East) UAE North

# (North America) East US

# (North America) East US 2

# (North America) West US

# (North America) West US 2

# (North America) West US 3

# (North America) Central US

# (North America) North Central US

# (North America) South Central US

# (North America) West Central US

# (North America) Canada Central

# (North America) Canada East

# (South America) Brazil South

# What's new in Azure Container Storage

# Based on feedback from customers, we've included the following capabilities with the latest updates:

# Improve stateful application availability by using multi-zone storage pools and ZRS disks.

# Enable server-side encryption with customer-managed keys (Azure Disks only).

# Scale up by dynamically expanding volumes and storage pools backed by Azure Disks and Ephemeral Disk without downtime.

# Clone persistent volumes within a storage pool.

# Optimize applications with Azure Linux Container Host.

# Increase resiliency for applications using local NVMe volumes with replication.

# For more information on these features, email the Azure Container Storage team at azcontainerstorage@microsoft.com.

# Why Azure Container Storage is useful

# Until now, providing cloud storage for containers required using individual container storage interface (CSI) drivers to use storage services intended for IaaS-centric workloads and make them work for containers. This creates operational overhead and increases the risk of issues with application availability, scalability, performance, usability, and cost.

# Azure Container Storage is derived from OpenEBS, an open-source solution that provides container storage capabilities for Kubernetes. By offering a managed volume orchestration solution via microservice-based storage controllers in a Kubernetes environment, Azure Container Storage enables true container-native storage.

# You can use Azure Container Storage to:

# Accelerate VM-to-container initiatives: Azure Container Storage surfaces the full spectrum of Azure block storage offerings that were previously only available for VMs and makes them available for containers. This includes ephemeral disk that provides extremely low latency for workloads like Cassandra, as well as Azure Elastic SAN (Preview) that provides native iSCSI and shared provisioned targets.

# Simplify volume management with Kubernetes: By providing volume orchestration via the Kubernetes control plane, Azure Container Storage makes it easy to deploy and manage volumes within Kubernetes - without the need to move back and forth between different control planes.

# Reduce total cost of ownership (TCO): Improve cost efficiency by increasing the scale of persistent volumes supported per pod or node. Reduce the storage resources needed for provisioning by dynamically sharing storage resources. Note that scale up support for the storage pool itself isn't supported.

# Key benefits

# Rapid scale out of stateful pods: Azure Container Storage mounts persistent volumes over network block storage protocols (NVMe-oF or iSCSI), offering fast attach and detach of persistent volumes. You can start small and deploy resources as needed while making sure your applications aren't starved or disrupted, either during initialization or in production. Application resiliency is improved with pod respawns across the cluster, requiring rapid movement of persistent volumes. Leveraging remote network protocols, Azure Container Storage tightly couples with the pod lifecycle to support highly resilient, high-scale stateful applications on AKS.

# Improved performance for stateful workloads: Azure Container Storage enables superior read performance and provides near-disk write performance by using NVMe-oF over RDMA. This allows customers to cost-effectively meet performance requirements for various container workloads including tier 1 I/O intensive, general purpose, throughput sensitive, and dev/test. Accelerate the attach/detach time of persistent volumes and minimize pod failover time.

# Kubernetes-native volume orchestration: Create storage pools and persistent volumes, capture snapshots, and manage the entire lifecycle of volumes using kubectl commands without switching between toolsets for different control plane operations.

# Glossary

# It's helpful to understand some key terms relating to Azure Container Storage and Kubernetes:

# Containerization

# Packing application code with only the operating system and required dependencies to create a single executable.

# Kubernetes

# Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

# Cluster

# A Kubernetes cluster is a set of compute nodes (VMs) that run containerized applications. Each node is managed by the control plane and contains the services necessary to run pods.

# Pod

# A pod is a group of one or more containers, with shared storage and network resources, and a specification for how to run the containers. A pod is the smallest deployable unit in a Kubernetes cluster.

# Azure Kubernetes Service (AKS)

# Azure Kubernetes Service is a hosted Kubernetes service that simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational overhead to Azure. Azure handles critical tasks, like health monitoring and maintenance.

# Storage pool

# The Azure Container Storage stack attempts to unify the object model across cluster owned resources and platform abstractions. To accomplish the unified representation, the available storage capacity is aggregated into a storage pool object. The storage capacity within a storage pool is considered homogeneous. An AKS cluster can have multiple storage pools. Storage pools also serve as the authentication and provisioning boundary. They provide a logical construct for operators to manage the storage infrastructure while simplifying volume creation and management for application developers.

# Storage class

# A Kubernetes storage class defines how a unit of storage is dynamically created with a persistent volume. For more information, see Kubernetes Storage Classes.

# Volume

# A Kubernetes volume is a directory containing data accessible to containers in a given pod. Volumes can be persistent or ephemeral. Volumes are thinly provisioned within a storage pool and share the performance characteristics (IOPS, bandwidth, and capacity) of the storage pool.

# Persistent volume

# Persistent volumes are like disks in a VM. They represent a raw block device that you can use to mount any file system. Application developers create persistent volumes alongside their application or pod definitions, and the volumes are often tied to the lifecycle of the stateful application. For more information, see Persistent Volumes.

# Persistent volume claim (PVC)

# A persistent volume claim is used to automatically provision storage based on a storage class.