**Introduction to Azure Files**

Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard [Server Message Block (SMB) protocol](https://msdn.microsoft.com/library/windows/desktop/aa365233.aspx) (also known as Common Internet File System or CIFS). Azure File shares can be mounted concurrently by cloud or on-premises deployments of Windows, Linux, and macOS. Additionally, Azure File shares can be cached on Windows Servers with Azure File Sync (preview) for fast access near where the data is being used.

## Why Azure Files is useful

Azure File shares can be used to:

* **Replace or supplement on-premises file servers**:  
  Azure Files can be used to completely replace or supplement traditional on-premises file servers or NAS devices. Popular operating systems such as Windows, macOS, and Linux can directly mount Azure File shares wherever they are in the world. Azure File shares can also be replicated with Azure File Sync to Windows Servers, either on-premises or in the cloud, for performant and distributed caching of the data where it's being used.
* **"Lift and shift" applications**:  
  Azure Files makes it easy to "lift and shift" applications to the cloud that expect a file share to store file application or user data. Azure Files enables both the "classic" lift and shift scenario, where both the application and it's data are moved to Azure, and the "hybrid" lift and shift scenario, where the application data is moved to Azure Files, and the application continues to run on-premises.
* **Simplify cloud development**:  
  Azure Files can also be used in numerous ways to simplify new cloud development projects. For example:
  + **Shared application settings**:  
    A common pattern for distributed applications is to have configuration files in a centralized location where they can be accessed from many application instances. Application instances can load their configuration through the File REST API, and humans can access them as needed by mounting the SMB share locally.
  + **Diagnostic share**:  
    An Azure File share is a convenient place for cloud applications to their write logs, metrics, and crash dumps. Logs can be written by the application instances via the File REST API, and developers can access them by mounting the file share on their local machine. This enables great flexibility, as developers can embrace cloud development without having to abandon any existing tooling they know and love.
  + **Dev/Test/Debug**:  
    When developers or administrators are working on VMs in the cloud, they often need a set of tools or utilities. Copying such utilities and tools to each VM can be a time consuming exercise. By mounting an Azure File share locally on the VMs, a developer and administrator can quickly access their tools and utilities, no copying required.

## Key benefits

* **Shared access**. Azure File shares support the industry standard SMB protocol, meaning you can seamlessly replace your on-premises file shares with Azure File shares without worrying about application compatibility. Being able to share a file system across multiple machines, applications/instances is a significant advantage with Azure Files for applications that need shareability.
* **Fully managed**. Azure File shares can be created without the need to manage hardware or an OS. This means you don't have to deal with patching the server OS with critical security upgrades or replacing faulty hard disks.
* **Scripting and tooling**. PowerShell cmdlets and Azure CLI can be used to create, mount, and manage Azure File shares as part of the administration of Azure applications.You can create and manage Azure file shares using Azure portal and Azure Storage Explorer.
* **Resiliency**. Azure Files has been built from the ground up to be always available. Replacing on-premises file shares with Azure Files means you no longer have to wake up to deal with local power outages or network issues.
* **Familiar programmability**. Applications running in Azure can access data in the share via file [system I/O APIs](https://msdn.microsoft.com/library/system.io.file.aspx). Developers can therefore leverage their existing code and skills to migrate existing applications. In addition to System IO APIs, you can use [Azure Storage Client Libraries](https://msdn.microsoft.com/library/azure/dn261237.aspx) or the [Azure Storage REST API](https://docs.microsoft.com/en-us/rest/api/storageservices/file-service-rest-api).