

Pourquoi créer une StorageClass supplémentaire dans AKS ?

Bien qu'Azure Kubernetes Service (AKS) fournisse plusieurs StorageClass prêtes à l'emploi pour couvrir la majorité des besoins courants, il existe de nombreuses situations où définir une StorageClass personnalisée est pertinent, voire nécessaire.

Raisons principales de créer une StorageClass personnalisée

- **Paramétrage avancé des performances**
 - Choix précis du type de disque (Standard, Premium, Ultra SSD, ZRS/LRS, etc.).
 - Définition de la taille minimale, du nombre d'IOPS, du débit, du mode de cache ou du bursting pour répondre à des exigences applicatives spécifiques (par exemple : bases de données haute performance) ^[1] ^[2].
- **Gestion du cycle de vie et des politiques**
 - Définir une politique de rétention différente (Retain au lieu de Delete) pour conserver les volumes après suppression des PVC, utile pour la sauvegarde ou la récupération de données ^[3].
 - Autoriser ou interdire l'expansion dynamique des volumes (allowVolumeExpansion).
- **Conformité et sécurité**
 - Utiliser un chiffrement spécifique (ex : Disk Encryption Set), ou imposer des tags Azure pour la gestion des coûts et la traçabilité ^[1] ^[4].
 - Restreindre l'accès réseau ou choisir un compte de stockage particulier pour répondre à des contraintes de sécurité.
- **Optimisation des coûts**
 - Créer une StorageClass qui utilise un SKU moins coûteux que les classes standards (par exemple, forcer l'utilisation de LRS au lieu de ZRS pour réduire la facture) ^[1].
- **Cas d'usage avancés**
 - Besoin d'un protocole ou d'une intégration spécifique (NFS sur Azure Blob, Blobfuse, NetApp Files, etc.) non couverts par les classes standards ^[4].
 - Déploiement multi-tenant : chaque équipe ou application peut avoir sa propre StorageClass avec des paramètres adaptés à ses besoins.

Exemples concrets

Besoin métier ou technique	Solution via StorageClass personnalisée
Disque Premium avec rétention des volumes	<code>reclaimPolicy: Retain</code> et <code>skuName: Premium_LRS</code>
Stockage partagé NFS sur Azure Blob	Protocole NFS et tags spécifiques dans StorageClass

Besoin métier ou technique	Solution via StorageClass personnalisée
Chiffrement avancé avec clé cliente	Paramètre <code>diskEncryptionSetID</code> dans la StorageClass
Optimisation du coût avec stockage LRS	<code>skuName: Standard_LRS</code> au lieu de ZRS par défaut
Volume élastique (expansion à chaud)	<code>allowVolumeExpansion: true</code>

À retenir

- Les StorageClass standards d'AKS couvrent les usages génériques.
- Une StorageClass personnalisée permet d'adapter le stockage aux besoins précis de performance, sécurité, coût ou conformité de chaque application ou équipe.
- Cette flexibilité est essentielle pour les environnements de production exigeants ou réglementés ^[1] ^[3] ^[4].

Créer une StorageClass supplémentaire, c'est donc répondre à un besoin spécifique non couvert par les classes prédéfinies du cluster.

*~

1. <https://learn.microsoft.com/en-us/azure/aks/azure-csi-disk-storage-provision>
2. <https://github.com/paolosalvatori/burstable-managed-csi-premium>
3. <https://www.netapp.com/learn/azure-cvo-blg-azure-kubernetes-service-configuring-persistent-volumes-in-aks/>
4. <https://learn.microsoft.com/en-us/azure/aks/azure-csi-blob-storage-provision?tabs=secret>