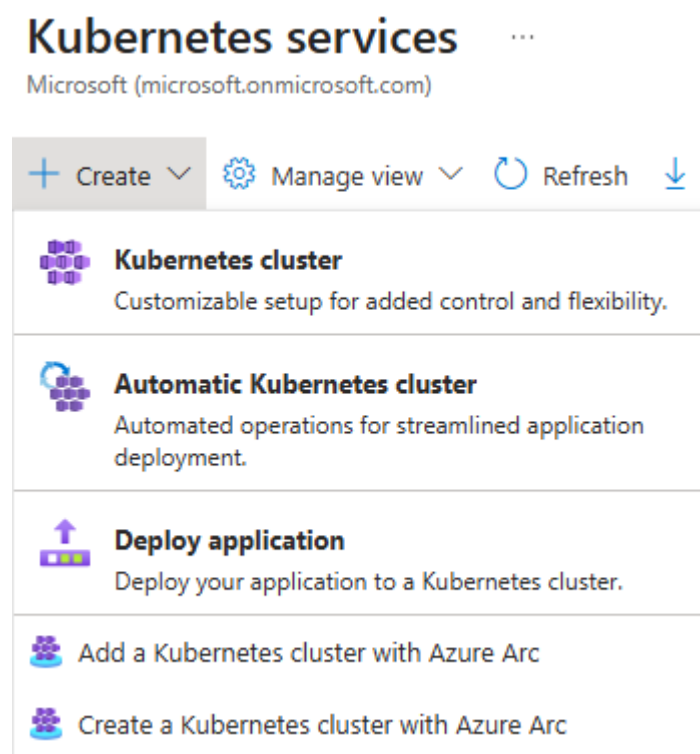


Créer un cluster automatique Azure Kubernetes Service (AKS)

Créer un cluster Kubernetes automatique

1. Pour créer un cluster AKS Automatic, recherchez **Kubernetes Services**, puis sélectionnez **Cluster Kubernetes Automatic** dans les options de la liste déroulante.



2. Dans l'onglet **Bases**, remplissez tous les champs obligatoires (Abonnement, Groupe de ressources, Nom du cluster Kubernetes et Région) requis pour commencer :

Create an Automatic Kubernetes Cluster ...

Basics Monitoring Advanced Tags Review + create


Azure Kubernetes Service (AKS) Automatic clusters are fully Azure-managed, with a simplified setup for more streamlined application deployment. This quicker, easier setup option includes sensible defaults that suit most production scenarios. Preview features must be registered to the selected subscription in order to create an Automatic Kubernetes cluster. [Learn more](#)



Automatic Kubernetes Cluster manages these elements for you:

- **Networking and Security** Azure CNI Overlay powered by Azure Cilium
- **Resource provisioning** Automated node provisioning and scaling
- **On-demand scaling** Optimal scaling tools like KEDA, HPA, and VPA
- **Kubernetes version upgrade** Automatic updates for enhanced stability


Project details


Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.


Subscription * 

 Resource group *  [Create new](#)

Cluster details

Kubernetes cluster name * 



Region * 

Automatic upgrade scheduler 

Start on: Fri Aug 29 2025 00:00 +00:00 (Coordinated Universal Time)
[Edit schedule](#)

Access control

Authentication and Authorization  Microsoft Entra ID authentication with Azure RBAC

 This RBAC configuration only applies to data actions on the Kubernetes API and not to actions on the Azure Resource Manager representation of the AKS cluster. [Learn more](#) 

3. Sous l'onglet **Surveillance**, choisissez vos configurations de supervision dans Azure Monitor, Prometheus managé, Grafana Dashboards, Container Network Observability (ACNS) et/ou configurez des alertes. Activez Managed Grafana (facultatif), ajoutez des balises (facultatif) et procédez à la création du cluster.

Create an Automatic Kubernetes Cluster ...

Basics Monitoring Advanced Tags Review + create

Azure Monitor

In addition to the CPU and memory metrics included in AKS by default, you can enable Container Insights for more comprehensive data on the overall performance and health of your cluster. Billing is based on data ingestion and retention settings.


[Learn more about container performance and health monitoring](#) 


[Learn more about Managed Grafana pricing](#) 


Container Insights

Enable Container Logs



 Azure monitor is recommended for production standard configuration.

Log Analytics workspace * 

DefaultWorkspace-9b8218f9-902a-4d20-a65c-e98acec5362f-CUS 


[Create new](#)

Cost Preset * 

Logs and events 

1 min collection frequency
No namespaces
Syslogs Disabled

Managed Prometheus

Managed Prometheus provides a highly available, scalable, and secure metrics platform to monitor your containerized workloads. [Learn more](#) 

Enable Prometheus metrics



Azure Monitor workspace *

defaultazuremonitorworkspace-centralus (defaultresourcegroup-centr... 

[Create new](#)

Grafana Dashboards

Grafana dashboards are included at no additional cost, with an option to enable a fully managed instance for Prometheus data in Azure Monitor.

[Learn more about Azure Monitor dashboards with Grafana](#) 

[Learn more about Managed Grafana pricing](#) 


Dashboards with Grafana (preview)



Enable Grafana



Container Network Observability (ACNS)


Collect pod metrics and flow logs using Advanced Container Networking Services (ACNS). Please note that generation and storage of flow logs require custom resource configuration after ACNS is enabled. [Learn more](#) 


[Learn more about ACNS pricing](#) 

Enable Container Network Observability with ACNS



Alerts

To Monitor events on your cluster, enable alerts recommended for your monitoring configuration above (you can select and customize the rules) [Learn more about recommended alert rules](#) 

Enable recommended alert rules 



Alert rules

Alert me if

- CPU Usage Percentage is greater than 95%
- Memory Working Set Percentage is greater than 100%

Notify me by


4. Sous l'onglet **Avancé**, mettez à jour vos paramètres de mise en réseau (facultatif), d'identité managée (facultatif), de sécurité et d'espaces de noms managés (facultatif) et passez à la création du cluster.

[Home](#) > [Kubernetes services](#) >

Create an Automatic Kubernetes Cluster ...


Basics Monitoring **Advanced** Tags Review + create

Private access

Enable a private cluster to restrict worker node to API access, enhancing your Kubernetes workload's security and isolation. [Learn more](#) 

Enable private cluster ⓘ ☐

Azure Virtual Networking

Bringing your own Azure virtual network will provision the AKS cluster nodes into a subnet from a virtual network provided by you. [Learn more](#) 

Bring your own Azure virtual network ⓘ ☐

Managed identity


Managed identity System assigned

Container Network Security (ACNS)

Prepare your cluster for security features such as advanced network policies, encryption, etc. with Advanced Container Networking Services (ACNS). Please note that network policy-related capabilities require explicitly creating network policy resources; no policies are applied by default. [Learn more](#) 
[Learn more about ACNS pricing](#) 

Enable Container Network Security with ACNS ☒

Managed Kubernetes Namespaces (preview)

Managed Kubernetes Namespaces enable you to configure resource quotas, enforce network policies, and control access within your cluster. To create a Managed Kubernetes Namespace, you must have write permissions on the cluster and the cluster must be integrated with Azure RBAC for access management. [Learn more](#) 

 Add Managed Kubernetes Namespace  Delete

<input type="checkbox"/>	Name	CPU Requests	CPU Limits	Memory Requests	Memory Limits
--------------------------	------	--------------	------------	-----------------	---------------

No items found

ⓘ This subscription does not have the flags: ManagedNamespacePreview registered. Preview features must be registered in order to create a Managed Kubernetes Namespace. To continue, register the required flags to the subscription.
[Register preview features](#)

Se connecter au cluster

Ajouter le rôle **Azure Kubernetes Service Cluster User** à votre compte :

- Ouvrez le Portail Azure et naviguez vers votre cluster myAKSAutomaticCluster.
- Cliquez sur Contrôle d'accès (IAM) dans le menu de gauche.
- Cliquez sur + Ajouter > Ajouter une attribution de rôle.
- Recherchez et sélectionnez le rôle **Azure Kubernetes Service Cluster User**.
- Sous Membres, cliquez sur + Sélectionner des membres et recherchez votre utilisateur connecté.
- Cliquez sur Vérifier + affecter.

Ouvrir un terminal powershell et exécuter la commande suivante :

```
az login --username <votre_compte_azure>
```

Configurez kubectl pour se connecter à votre cluster Kubernetes à l'aide de la commande [az aks get-credentials](#). Cette commande télécharge les informations d'identification et configure l'interface CLI Kubernetes pour les utiliser, exécutez la commande suivante :

```
az aks get-credentials --resource-group myResourceGroup --name myAKSAutomaticCluster
```

Pour vérifier la connexion à votre cluster, exécutez la commande [kubectl get](#). Cette commande renvoie la liste des nœuds de cluster.

Exécutez la commande suivante :

```
kubectl get nodes
```

L'exemple de sortie suivant montre comment vous êtes invité à vous connecter.

Sortie

To sign in, use a web browser to open the page <https://microsoft.com/devicelogin> and enter the code AAAAAAAAAA to authenticate.

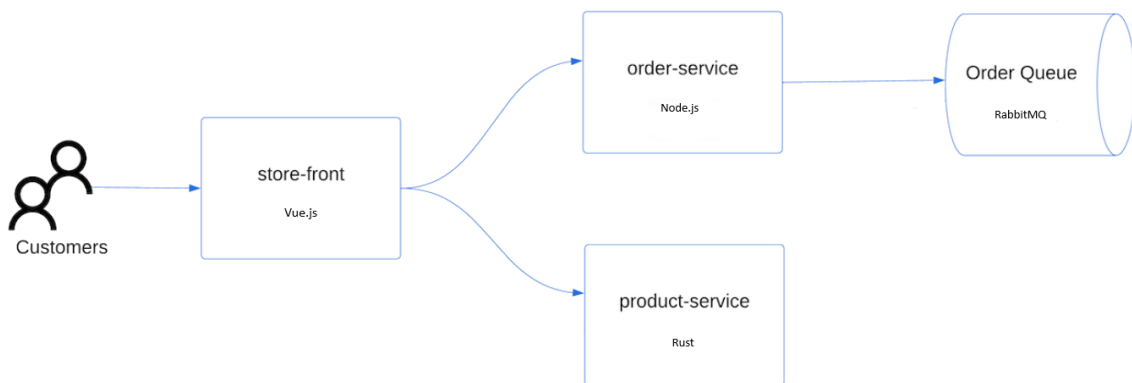
Une fois connecté, l'exemple de sortie suivant montre les pools de nœuds système managés. Assurez-vous que l'état du nœud est *Prêt*.

Sortie

NAME	STATUS	ROLES	AGE	VERSION
aks-nodepool1-13213685-vmss000000	Ready	agent	2m26s	v1.28.5
aks-nodepool1-13213685-vmss000001	Ready	agent	2m26s	v1.28.5
aks-nodepool1-13213685-vmss000002	Ready	agent	2m26s	v1.28.5

Déployer l'application

Pour déployer l'application, vous utilisez un fichier manifeste pour créer tous les objets nécessaires à l'exécution de l'[application AKS Store](#). Un [fichier manifeste Kubernetes](#) définit un état souhaité d'un cluster, notamment les images conteneur à exécuter. Le manifeste inclut les déploiements et services Kubernetes suivants :



- **Vitrine** : application web permettant aux clients d'afficher les produits et de passer des commandes.
- **Service de produit** : affiche les informations sur le produit.
- **Service de commande** : passe des commandes.
- **Rabbit MQ** : file d'attente de messages pour une file d'attente de commandes.

Remarque

Nous ne recommandons pas l'exécution de conteneurs avec état, comme Rabbit MQ, sans stockage persistant pour la production. Ils sont utilisés ici pour des raisons de simplicité, mais nous vous recommandons d'utiliser des services managés, comme Azure CosmosDB ou Azure Service Bus.

1. Créez un espace de noms aks-store-demo pour y déployer les ressources Kubernetes.

Exécuter la commande suivante:

```
kubectl create ns aks-store-demo
```

2. Déployez l'application à l'aide de la commande [kubectl apply](#) dans l'espace de noms aks-store-demo. Le fichier YAML définissant le déploiement se trouve sur [GitHub](#).

Exécuter la commande suivante:

```
kubectl apply -n aks-store-demo -f https://raw.githubusercontent.com/Azure-Samples/aks-store-demo/main/aks-store-ingress-quickstart.yaml
```

L'exemple de sortie suivant montre les déploiements et les services :

Sortie

```
statefulset.apps/rabbitmq created
configmap/rabbitmq-enabled-plugins created
service/rabbitmq created
deployment.apps/order-service created
service/order-service created
deployment.apps/product-service created
service/product-service created
deployment.apps/store-front created
service/store-front created
ingress/store-front created
```

Tester l'application

Quand l'application s'exécute, un service Kubernetes expose le front-end de l'application sur Internet. L'exécution de ce processus peut prendre plusieurs minutes.

1. Vérifiez l'état des pods déployés à l'aide de la commande [kubectl get pods](#). Vérifiez que tous les pods sont Running avant de continuer. S'il s'agit de la première charge de travail que vous déployez, l'[approvisionnement automatique de nœuds](#) peut prendre quelques minutes pour créer un pool de nœuds visant à exécuter les pods.

Bash

```
kubectl get pods -n aks-store-demo
```

2. Recherchez une adresse IP publique pour l'application de vitrine. Surveillez la progression avec la commande [kubectl get service](#) et l'argument `--watch`.

Bash

```
kubectl get ingress store-front -n aks-store-demo --watch
```

La sortie **ADDRESS** pour le service store-front n'affiche rien initialement :

Sortie

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
store-front	webapprouting.kubernetes.azure.com	*		80	12m

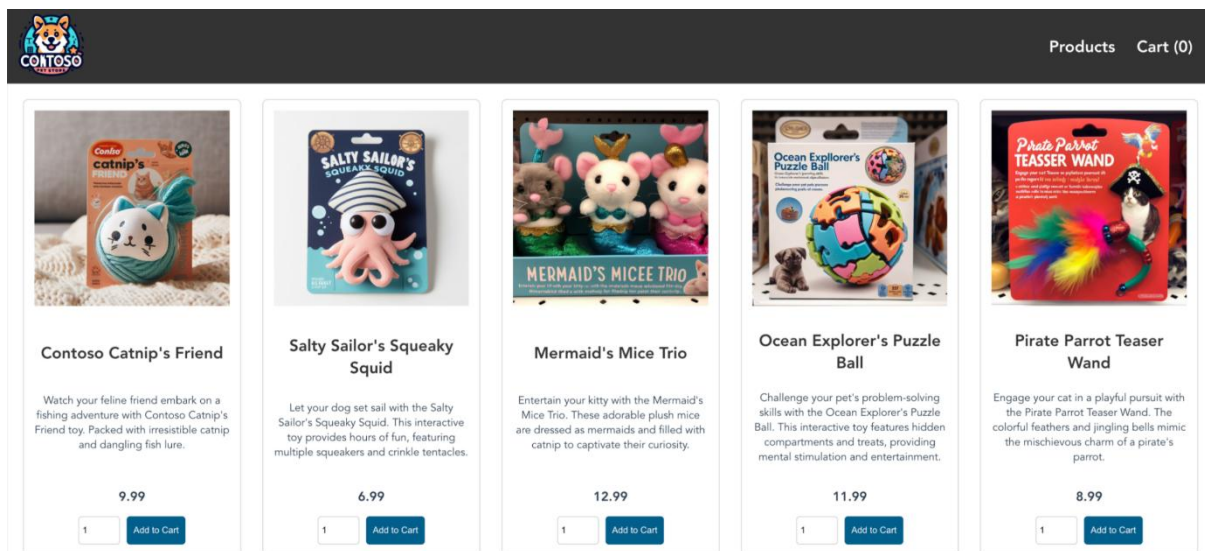
- Une fois qu'**ADDRESS** passe de rien à une adresse IP publique réelle, utilisez CTRL-C pour arrêter le processus de surveillance kubectl.

L'exemple de sortie suivant montre une adresse IP publique valide affectée au service :

Sortie

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
store-front	webapprouting.kubernetes.azure.com	*	4.255.22.196	80	12m

- Ouvrez un navigateur web à l'adresse IP externe de votre entrée pour voir l'application Azure Store en action.



Supprimer le cluster

Pour éviter les frais Azure, si vous ne prévoyez pas de suivre le [Tutoriel AKS](#), nettoyez vos ressources inutiles. Exécutez la commande [az group delete](#) pour supprimer le groupe de ressources, le service conteneur ainsi que toutes les ressources associées.

Azure CLI

```
az group delete --name myResourceGroup --yes --no-wait
```

Remarque

Le cluster AKS a été créé avec une identité managée affectée par le système, qui est l'option d'identité par défaut utilisée dans ce guide de démarrage rapide. La plateforme gère cette identité pour vous éviter de devoir la supprimer manuellement.

Étapes suivantes

Dans ce guide de démarrage rapide, vous avez déployé un cluster Kubernetes avec [AKS Automatic](#), puis vous y avez déployé une application multiconteneur simple. Cet exemple d'application est fourni à des fins de version de démonstration uniquement et ne représente

pas toutes les meilleures pratiques pour les applications Kubernetes. Pour obtenir des conseils sur la création de solutions complètes avec AKS pour la production, consultez [Conseils pour les solutions AKS](#).

Pour en savoir plus sur AKS Automatic, passez à la présentation.