

A person is walking away from the camera on a suspension bridge. The bridge has wooden planks and metal mesh railings. The background shows a dense forest of evergreen trees, and in the distance, there are rugged mountains with patches of snow. The sun is low in the sky, creating a bright glow and lens flare effects. A semi-transparent dark blue rectangle is overlaid on the center of the image, containing the title and author information.

AKS AZURE.

Pierre-Mickael CHANCRIN

Livre blanc . 09/2020

Sommaire.

AUTEUR. 2

PRÉSENTATION DE AKS

1.	La continuité de service avec AKS	6
2.	Mise en place d'un cluster AKS	12
3.	Fonctionnalités AKS	20
4.	Les pools de nodes dans AKS	21
5.	Affinité et anti-affinité	22
6.	Node Affinity	24
7.	Node Selector	25
8.	Gestion du réseau des nodes	27
9.	Intégrer Kubernetes avec Continuous Integration (CI)	29
10.	Intégrer Kubernetes avec Active Directory	38
11.	Azure Container Registry	41
12.	L'autoscaling des Pods	44
13.	L'autoscaling des nodes	48
14.	Présentation de Azure Dev Dpaces	52
15.	La sécurité dans AKS	54
16.	AKS avec Security Center	58
17.	Azure Monitor pour AKS	62
18.	Troubleshooting	68

CONCLUSION

CERTIFICATIONS MICROSOFT & KUBERNETES

DÉCOUVREZ TOUS LES LIVRES BLANCS.	75
A PROPOS DE IPPON TECHNOLOGIES.	76
NOS SOLUTIONS.	77
LICENCE.	78

L'auteur.



Pierre-Mickael CHANCRIN

Consultant Cloud chez Ippon Technologies, passionné depuis toujours par les nouvelles technologies autour de la virtualisation et du cloud. J'interviens en tant qu'Expert & Architecte pour Analyser, recommander et valider des solutions techniques. Je suis certifié sur les technologies VMWare, Microsoft & AWS afin de valoriser mes connaissances acquises au travers de mes différentes expériences professionnelles.

Retrouvez ses dernières publications sur le site [Ippon Technologies](#) & ses articles sur le [blog Ippon](#).

A person with a backpack is walking away from the camera on a suspension bridge. The bridge spans a deep valley filled with dense evergreen trees. In the distance, rugged mountains are partially covered in snow under a cloudy sky. The scene is captured in a cinematic style with a blue-tinted overlay.

PRÉSENTATION DE AKS.

De la virtualisation à la containerisation dans Azure, Microsoft lance sa première offre de containerisation Azure Container Service (ACS) en 2016, celle-ci était une offre PAAS, permettant de déployer un cluster Kubernetes.

Ce service est facilement exploitable et permet ainsi de se concentrer sur le développement de ces applications et non sur la mise en place de Kubernetes.

Du fait de la montée en puissance de Kubernetes, Microsoft a remplacé l'acronyme ACS par AKS (Azure Kubernetes Services) en 2017 et en a fait ainsi un service managé. En 2019, Microsoft arrête le support de ACS pour pousser les clients à migrer vers AKS.

Il faut savoir que Brendan Burns, co-créateur de Kubernetes, est maintenant chez Microsoft et y dirige les projets liés aux containers.

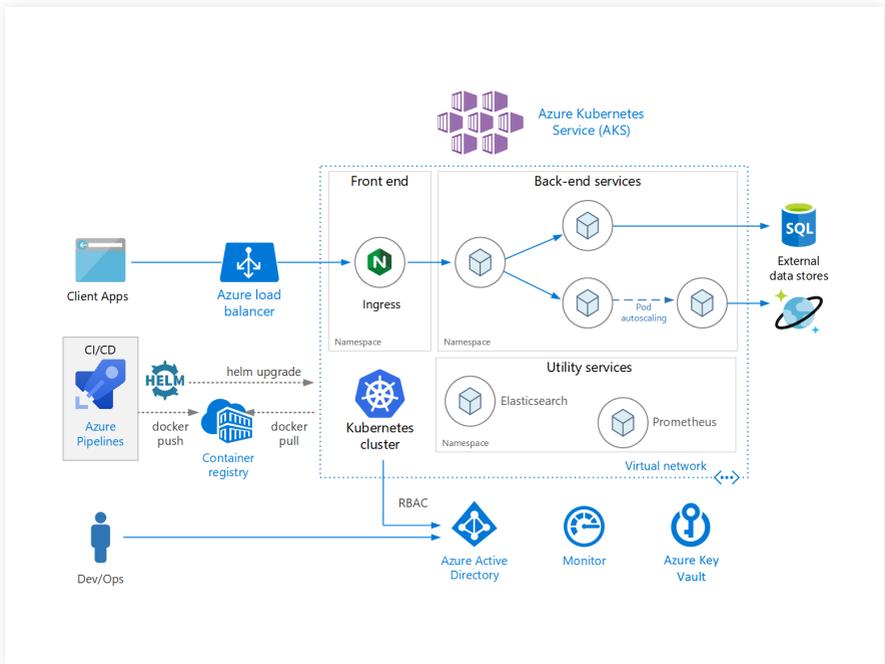
De plus, Microsoft a rejoint la CNCF (Cloud Native Computing Foundation) en 2018 et a également fait l'acquisition de Deis, une entreprise à la base de plusieurs outils Kubernetes open source.

Dans le cadre de son offre de Service Managé, Azure gère pour nous les nodes Master. Nous devons uniquement nous soucier des nodes Worker concernant notre application.

[Les ressources Azure AKS sont disponibles.](#)

Ci-dessous le schéma de principe du fonctionnement de Kubernetes dans Azure.

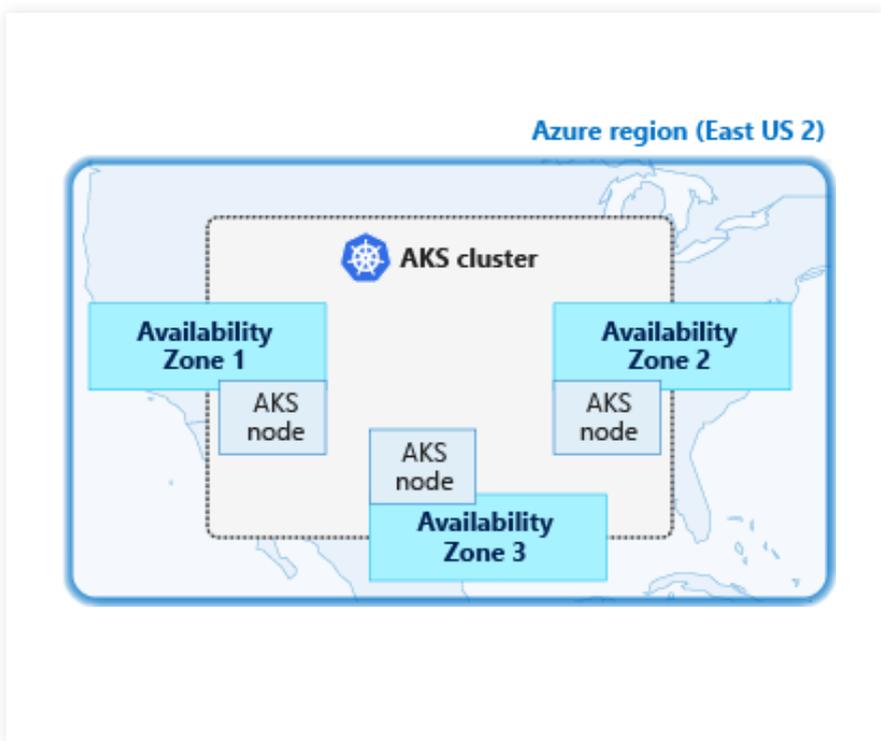
Cette architecture peut servir de référence :

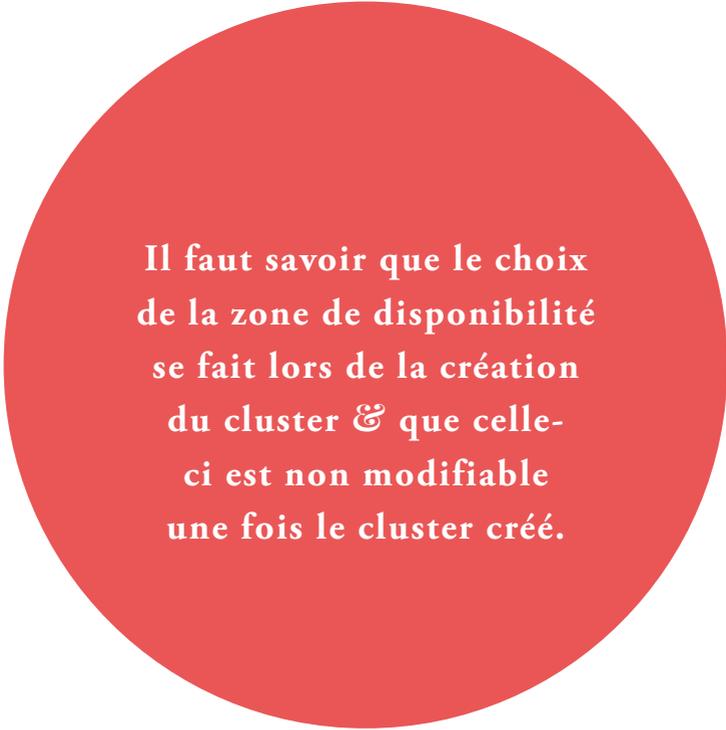


1. LA CONTINUITÉ DE SERVICE AVEC AKS

AKS nous permet de faire de la résilience en cas de défaillance d'un service. Pour cela, il faut penser à répartir les nodes dans les différentes zones de disponibilité au sein de notre région.

Pour connaître les services disponibles sur une région, je vous invite à cliquer sur ce [lien](#).





**Il faut savoir que le choix
de la zone de disponibilité
se fait lors de la création
du cluster & que celle-
ci est non modifiable
une fois le cluster créé.**

› La sauvegarde des données

Azure recommande de sauvegarder les données de ces nodes si cela est nécessaire (bases de données par exemple) à l'aide d'un outil tel que Velero ou Azure Site Recovery et de s'assurer de l'intégrité de ces sauvegardes.

Velero permet de sauvegarder les volumes persistants des ressources de cluster et de leur configuration.

La sauvegarde via Velero nécessite un Blob storage au sein d'un storage account pour stocker les backups.

Voici comment configurer la sauvegarde d'un environnement avec Velero. Tout d'abord, on crée le storage account et le blob storage pour stocker les backups.

Voici le script pour définir les variables :

```
# Prepare variables
TENANT_ID=...
SUBSCRIPTION_ID=...
SOURCE_AKS_RESOURCE_GROUP=MC_...
TARGET_AKS_RESOURCE_GROUP=MC_... # (optional, if you
want to migrate)
BACKUP_RESOURCE_GROUP=backups
BACKUP_STORAGE_ACCOUNT_NAME=velero$(uuidgen | cut -d '-'
-f5 | tr '[A-Z]' '[a-z]')
```

Voici le script pour définir la création du Storage Account :

```
# Create Azure Storage Account
az storage account create \
  --name $BACKUP_STORAGE_ACCOUNT_NAME \
  --resource-group $RESOURCE_GROUP \
  --sku Standard_GRS \
  --encryption-services blob \
  --https-only true \
  --kind BlobStorage \
  --access-tier Hot
```

Voici le script pour définir la création du Blob storage (équivalent à un bucket S3) :

```
# Create Blob Container
az storage container create \
  --name velero \
  --public-access off \
  --account-name $BACKUP_STORAGE_ACCOUNT_NAME
```

Il faut ensuite créer un SPN pour permettre l'accès aux ressources par Velero :

```
# Create a Service Principal for RBAC
AZURE_CLIENT_SECRET=`az ad sp create-for-rbac \
  --name "velero" \
  --role "Contributor" \
  --query 'password' \
  -o tsv \
  --scopes /subscriptions/$SUBSCRIPTION_ID/
resourceGroups/$BACKUP_RESOURCE_GROUP /
subscriptions/$SUBSCRIPTION_ID/resourceGroups/$SOURCE_
AKS_RESOURCE_GROUP /subscriptions/$SUBSCRIPTION_ID/
resourceGroups/$TARGET_AKS_RESOURCE_GROUP`
AZURE_CLIENT_ID=`az ad sp list --display-name "velero"
--query '[0].appId' -o tsv`
```

On installe ensuite Velero, pour cela, on renseigne les informations de la souscription :

```
cat << EOF > ./credentials-velero
AZURE_SUBSCRIPTION_ID=${AKS_SUBSCRIPTION_ID}
AZURE_TENANT_ID=${TENANT_ID}
AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
AZURE_RESOURCE_GROUP=${SOURCE_AKS_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

On lance ensuite l'installation de Velero :

```
velero install \  
  --provider azure \  
    --plugins velero/velero-plugin-for-microsoft-azure:v1.0.0 \  
  --bucket velero \  
  --secret-file ./credentials-velero \  
    --backup-location-config resourceGroup=$BACKUP_RESOURCE_GROUP,storageAccount=$BACKUP_STORAGE_ACCOUNT_NAME \  
    --snapshot-location-config apiTimeout=5m,resourceGroup=$BACKUP_RESOURCE_GROUP \  
  --wait
```

Nous pouvons ensuite lancer notre premier backup :

```
velero backup create firstbackup
```

Puis tester une restauration :

```
velero restore create firstrestore --from-backup firstbackup
```

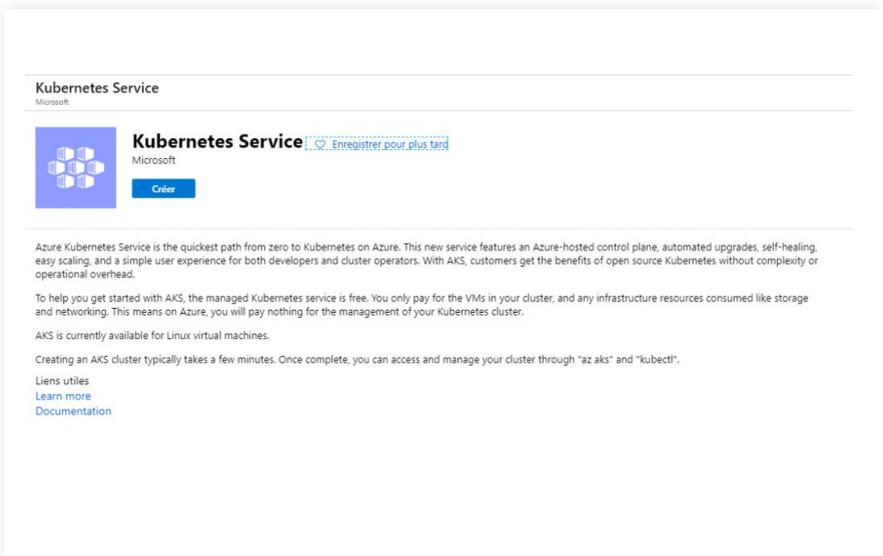
Si tout est bien correct, nous pouvons planifier la sauvegard :

```
velero create schedule NAME --schedule="@every 24h"
```

2. MISE EN PLACE D'UN CLUSTER AKS

Pour créer notre cluster, nous avons deux possibilités, soit via Cloud Shell, soit depuis [le portail Azure](#) en cliquant sur “Kubernetes Service”.

Azure Cloud Shell est un interpréteur de commandes interactif accessible depuis un navigateur internet pour gérer vos ressources Azure.



Il faut ensuite renseigner les différents champs demandés :

- groupe de ressource,
- nom du cluster,
- région,
- préfixe DNS :

Créer un cluster Kubernetes

De base Échelle Authentification Supervision Supervision Balises Vérifier + créer

Azure Kubernetes Service (AKS) gère votre environnement Kubernetes hébergé, facilitant et accélérant ainsi le déploiement et la gestion des applications en conteneur sans nécessairement disposer d'expertise en matière d'orchestration de conteneurs. Cela élimine également le fardeau des opérations et de la maintenance en provisionnant, en mettant à niveau et en mettant à l'échelle les ressources à la demande, sans mettre vos applications hors connexion. [En savoir plus sur Azure Kubernetes Service](#)

Détails du projet

Sélectionner un abonnement pour gérer les ressources déployées et les coûts associés. Utilisez des groupes de ressources comme des dossiers pour organiser et gérer toutes vos ressources.

Abonnement *

└─ Groupe de ressources * [Créer nouveau](#)

Détails du cluster

Nom du cluster Kubernetes * ✓

Région * ✓

Version de Kubernetes * ✓

Préfixe des noms DNS * ✓

Pool de nœuds principal

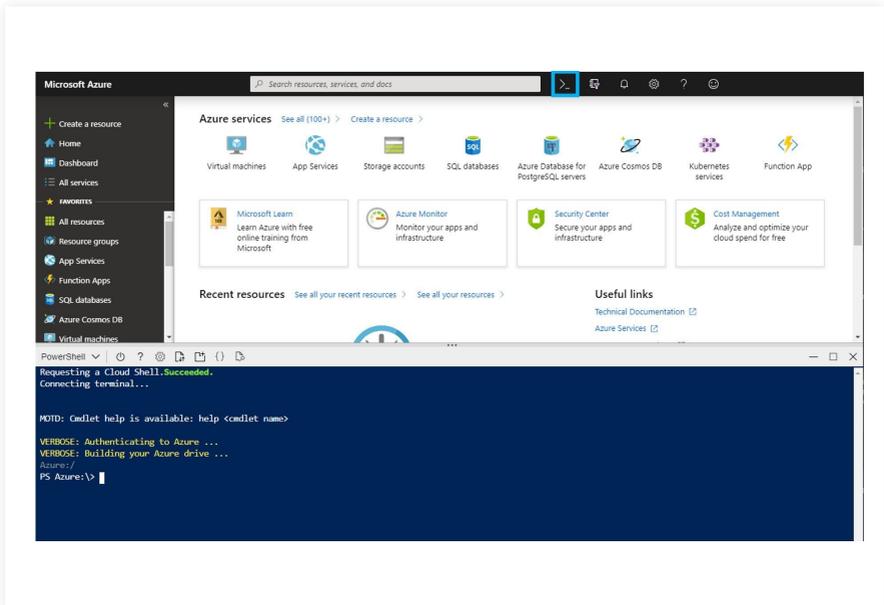
Le nombre et la taille des nœuds dans le pool de nœuds principal de votre cluster. Pour les charges de production, au moins 3 nœuds sont recommandés pour assurer la résilience. Pour les charges de développement ou de test, il suffit d'un seul nœud. Après avoir créé un cluster, vous ne pourrez pas modifier la taille des nœuds. En revanche, vous pourrez modifier son nombre de nœuds. Si vous souhaitez obtenir davantage de pools de nœuds, vous devez activer la fonctionnalité « X » dans l'onglet « Mettre à l'échelle », ce qui vous permettra d'ajouter davantage de pools de nœuds après la création du cluster. [En savoir plus sur les pools de nœuds dans Azure Kubernetes Service](#)

Taille de nœud * **Standard B2s**
2 processeurs virtuels, mémoire de 4 Gio
[Modifier la taille](#)

Nombre de nœuds * **1**

[Vérifier + créer](#) < Précédent **Suivant : Échelle >**

Vous pouvez aussi créer votre cluster via Cloud Shell en cliquant sur le bouton Powershell dans le bandeau du haut de votre portail Azure.



Voici la commande cloud shell pour créer notre cluster AKS :

```
az aks create \  
  --resource-group {Resource group name} \  
  --name {Cluster name} \  
  --node-count 2 \  
  --generate-ssh-keys \  
  --attach-acr {ACR name}
```

Ce qui nous donne la commande suivante :

```
az aks create --resource-group aksAvailabilityZone  
--name aks --generate-ssh-keys --vm-set-type  
VirtualMachineScaleSets --load-balancer-sku standard  
--node-count 3 --zones 1 2 3
```

Installer la CLI sur votre poste en ouvrant PowerShell et en exécutant la commande suivante :

```
az aks install-cli  
Connectez-vous ensuite sur votre cluster  
az aks get-credentials --resource-group {Resource group  
name} --name {Cluster name}
```

Ce qui nous donne la commande suivante :

```
az aks get-credentials --resource-group aksAvailabilityZone  
--name aks
```

La commande suivante permet de lister vos nodes :

```
$ kubectl get nodes
```

Voici le résultat attendu :

NAME	STATUS	ROLES	AGE	VERSION
Aks-nodepool-test	Ready	agent	50m	v1.14

La commande suivante permet d'avoir un détail de vos nodes :

```
kubectl describe nodes | select-string -pattern  
'^Name:', 'zone='
```

Celle-ci permet de monter le nombre de vos nodes à 5 nodes :

```
az aks scale --resource-group aksAvailabilityZone --name  
myAKSCluster --node-count 5
```

Celle-là permet de voir comment sont répartis les nodes au sein de votre cluster :

```
kubectl describe nodes | select-string -pattern  
'^Name:', 'zone='
```

Cette autre commande permet de déployer un pod reverse proxy nginx sur un node dans chacune de vos zones de disponibilité :

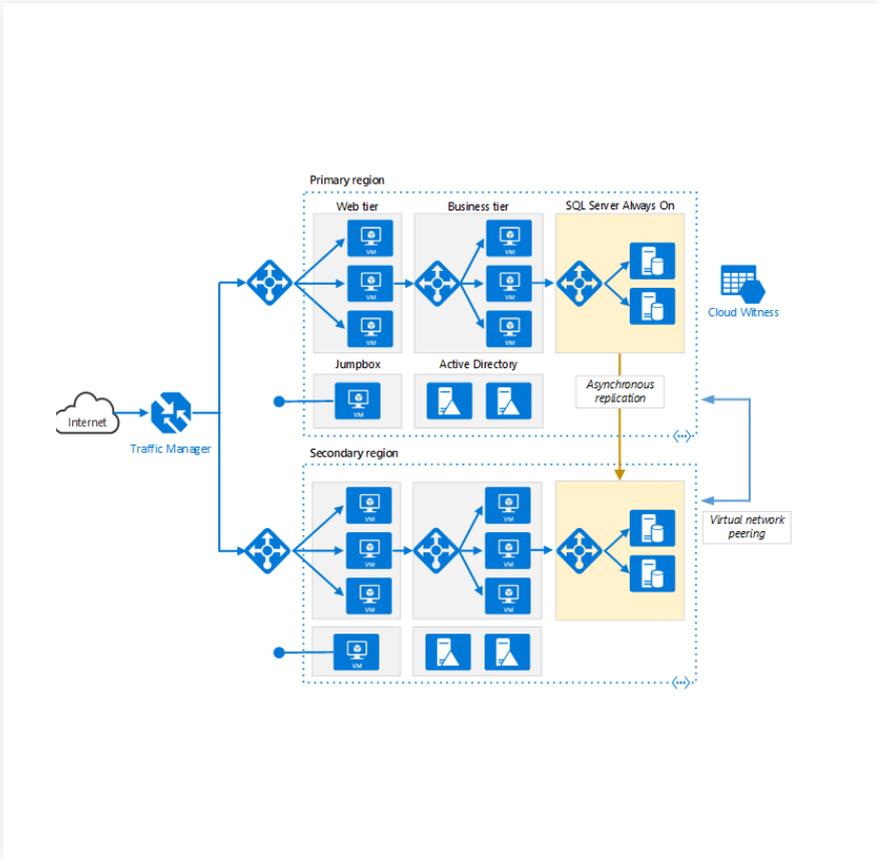
```
kubectl run nginx --image=nginx --replicas=3
```

La commande suivante permet d'avoir un descriptif de vos pod :

```
kubectl describe pod | select-string -pattern  
'^Name:', '^Node:'
```

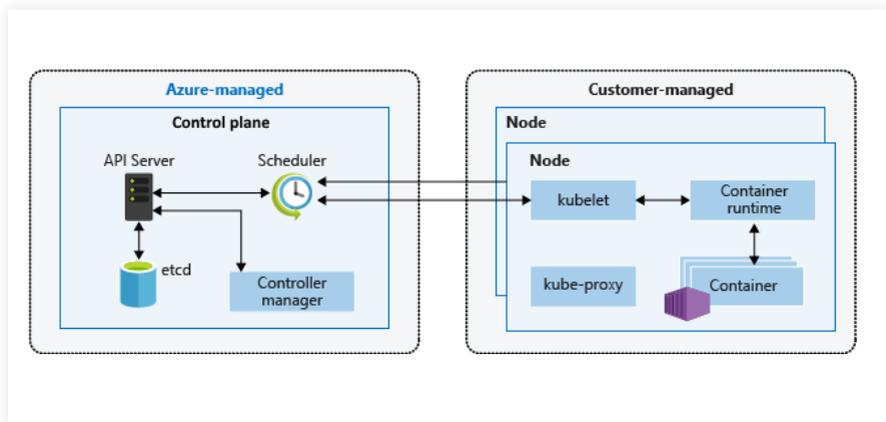


La solution Azure Traffic Manager permet de diriger les flux sur vos clusters se trouvant dans différentes régions suivant la latence, la zone géographique par exemple.



3. FONCTIONNALITÉS AKS

Kubernetes permet d'automatiser le déploiement et la gestion d'applications en micro-service (principalement avec Docker) tout en proposant de la scalabilité.



Comme dit précédemment, Azure gère les nodes master. Les rôles de ces derniers sont les suivants :

- Master : Gère la disponibilité des nodes.
- Etcid : La base de donnée de configuration.
- Kubelet : Il est responsable de l'état du node.
- Kube-proxy : Il est responsable de la partie réseau.
- Pod : C'est une enveloppe virtuelle pouvant abriter un ou plusieurs containers.
- Node : Machine virtuelle permettant l'exécution de pods (Containers).

4. LES POOLS DE NODES DANS AKS

Par défaut, AKS regroupe les nodes dans des pools de nodes. Un pool contient des vms identiques.

La commande permettant d'ajouter un pool de nodes à un cluster :

```
az aks nodepool add --resource-group rgAks --cluster-name myAKSCluster --name nodepool2 --node-count 1 --node-vm-size Standard_D4s_v3 --no-wait
```

La commande de suppression d'un pool de node :

```
az aks nodepool delete -g rgAks --cluster-name myAKSCluster --name nodepool2 --no-wait
```

5. AFFINITÉ ET ANTI-AFFINITÉ

Les deux notions importantes de Kubernetes : les teintes (taints) et les tolérances (tolerations). Celles-ci permettent de contrôler sur quels nodes vont s'exécuter les pods :

- Quand une teinte est appliquée à un node, seuls les pods spécifiques peuvent être planifiés sur le node.
- Ensuite, une tolérance est appliquée à un pod pour lui permettre de tolérer la teinte d'un node.

Les effets sont les suivants :

- **NoSchedule** : les pods ne seront pas déployés sur le node.
- **PreferNoSchedule** : Kubernetes déploiera le pod sur ce node uniquement si besoin
- **NoExecute** : si un pod est en cours d'exécution sur un node qu'il ne tolère pas, il sera supprimé.

La commande pour attribuer une teinte à un node est la suivante :

```
kubectl taint nodes nodename key=value:effect
```

Ce qui nous donne :

```
kubectl taint node aks-nodepool1 sku=gpu:NoSchedule
```

Ci-dessous un exemple de fichier yaml d'un pod avec une tolérance pour la "taint" « *noschedule* » :

```
apiVersion: v1
kind: Pod
spec:
  tolerations:
  - key: "key"
    operator: "Equal"
    value: "value"
    effect: "NoSchedule"
```

Si on ne souhaite pas que le node execute d'autres pods :

```
kubectl taint nodes es-node elasticsearch=false:NoExecute
```

On peut déployer un pod en utilisant `kubectl apply -f gpu-toleration.yaml`. Kubernetes planifie ensuite le pod sur l'un des nodes avec la teinte appliquée.

Lorsque qu'une teinte est appliquée, kube-scheduler va essayer de placer le pod avec la teinte sur le node grâce à la tolérance. Mais si le node n'est pas disponible pour un problème de ressource, par exemple, il sera planifié sur un autre node. Pour palier à ce problème, il faut utiliser les Nodes Affinity et les Nodes Selector.

Source : <https://docs.microsoft.com/fr-fr/azure/aks/operator-best-practices-advanced-scheduler>

6. NODE AFFINITY

Pour forcer un pod à tourner sur un node spécifique, il faut utiliser les Nodes Affinity comme ci-dessous via un système de clé/valeur :

```
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
        - matchExpressions:
            - key: kubernetes.io/hostname
              operator: In
              values:
                - es-node
```

Le champ Affinity permet de spécifier des affinités :

- requiredDuringSchedulingIgnoredDuringExecution
- preferredDuringSchedulingIgnoredDuringExecution

Il est aussi possible aussi d'utiliser ces opérateurs : ***In, NotIn, Exists, DoesNotExist.***

Source : <https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#affinity-and-anti-affinity>

7. NODE SELECTOR

Le sélecteur de node est donc un moyen simple d'affecter des pods à un node donné. Il se base sur un système de label, c'est à dire une isolation logique permettant de contrôler l'accès aux ressources au sein d'un cluster.

Ci-dessous, l'affectation d'un label :

```
kubectl label nodes <your-node-name> hardware:highmem
```

Ce qui donne:

```
kubectl label node aks-nodepool1 hardware:highmem
```

On vérifie ensuite les labels présents :

```
kubectl get nodes --show-labels
Le fichier yaml d'un pod en spécifiant le node selector :
kind: Pod
apiVersion: v1
metadata:
  name: tf-mnist
spec:
  containers:
  - name: tf-mnist
    image: microsoft/samples-tf-mnist-demo:gpu
    resources:

      requests:
        cpu: 0.5
        memory: 2Gi
      limits:
        cpu: 4.0
        memory: 16Gi
    nodeSelector:
      hardware: highmem
```

8. GESTION DU RÉSEAU DES NODES

Lors de la création d'un cluster AKS avec Azure CLI, il est également possible de créer un cluster AKS en activant un réseau Azure CNI. Pour information, Azure CNI est un plugin open source qui intègre les pods Kubernetes à un réseau virtuel Azure (également connu sous le nom de VNet) fournissant des performances réseau au même niveau que les machines virtuelles.

Ci-dessous les commande pour créer un cluster AKS en activant un réseau Azure CNI :

Tout d'abord, récupérez l'ID de la ressource du sous-réseau auquel le cluster AKS sera joint.

```
$ az network vnet subnet list \  
  --resource-group myVnet \  
  --vnet-name myVnet \  
  --query "[0].id" --output tsv  
  
/subscriptions/<guid>/resourceGroups/myVnet/providers/  
Microsoft.Network/virtualNetworks/myVnet/subnets/  
default
```

Utilisez la commande [*az aks create*](#) avec l'argument `--network-plugin azure` pour créer un cluster avec une mise en réseau avancée.

Remplacez la valeur `--vnet-subnet-id` par l'ID du sous-réseau recueilli à l'étape précédente :

```
az aks create \  
  --resource-group myResourceGroup \  
  --name myAKSCluster \  
  --network-plugin azure \  
  --vnet-subnet-id <subnet-id> \  
  --docker-bridge-address 172.17.0.1/16 \  
  --dns-service-ip 10.2.0.10 \  
  --service-cidr 10.2.0.0/24 \  
  --generate-ssh-keys
```

Il est possible de faire cela via le portail Azure :

The screenshot shows the 'Create Kubernetes cluster' page in the Azure portal, specifically the 'Networking' tab. The page title is 'Create Kubernetes cluster' and the breadcrumb is 'Home > New > Marketplace > Everything > Azure Kubernetes Service (preview) > Create Kubernetes cluster'. The 'Networking' tab is active, and the 'Advanced' radio button is selected and highlighted with a red box. The 'Basic' radio button is unselected. Below the radio buttons, there are several configuration fields: 'Virtual network' (aks-vnet-16033614), 'Subnet' (aks-subnet), 'Kubernetes service address range' (10.0.0.0/16), 'Kubernetes DNS service IP address' (10.0.0.10), and 'Docker Bridge address' (172.17.0.1/16). The 'Docker Bridge address' field has a green checkmark next to it. The 'Http application routing' section has 'No' selected. The 'Learn more about networking in Azure Kubernetes Service' link is visible.

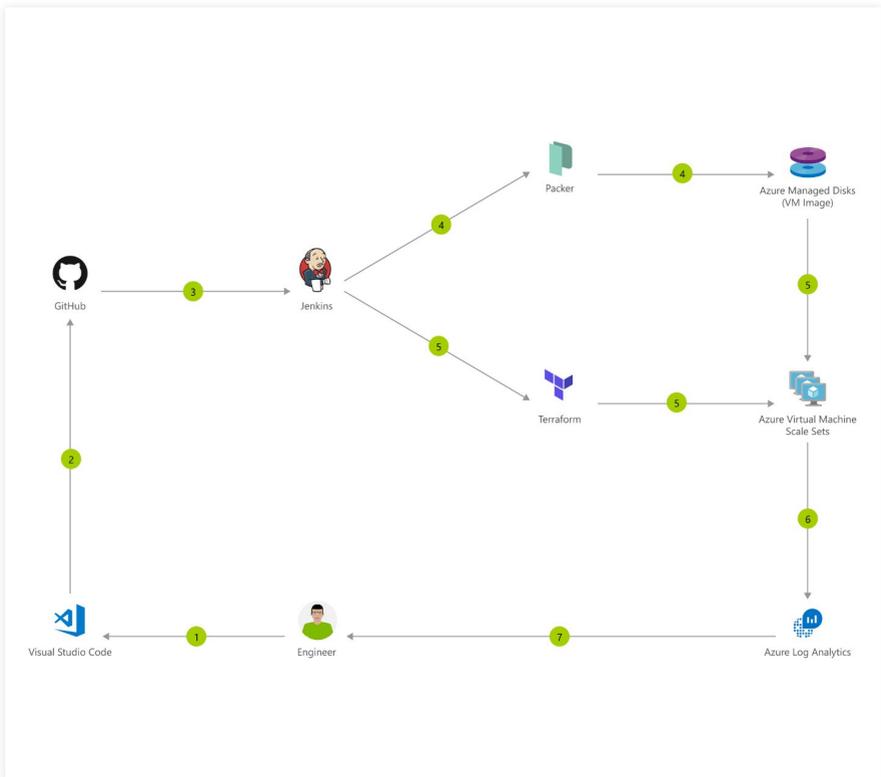
Source : <https://docs.microsoft.com/en-us/azure/aks/configure-azure-cni>

9. INTÉGRER KUBERNETES AVEC CONTINUOUS INTEGRATION (CI)

Pour effectuer du déploiement continu, on peut utiliser Azure Devops, Gitlab, Github ou Jenkins par exemple.

Un pipeline Azure Kubernetes contient généralement les étapes permettant de récupérer le code, de créer l'image Docker, de pousser ensuite cette image vers un référentiel puis de publier ensuite les artefacts.

<https://azure.microsoft.com/fr-fr/resources/templates/jenkins-cicd-container/>



Créer d'abord le groupe de ressources pour le déploiement :

```
az group create --name <resource-group-name> --location  
<resource-group-location>
```

Déployer dans votre groupe de ressources votre cluster Kubernetes :

```
az group deployment create --resource-group <my-resource-  
group> --template-uri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/jenkins-  
cicd-container/azuredeploy.json
```

1

Les développeurs itèrent, testent et déboguent rapidement leur application.

2

Le code est fusionné dans GitHub et les builds. Les tests automatisés sont exécutés par Azure Pipelines.

3

L'image de conteneur est stockée dans Azure Container Registry.

4

Les clusters Kubernetes sont créé via Terraform.

5

Les opérateurs créés leurs stratégies de déploiements de leur cluster AKS.

6

Le pipeline de mise en production exécute automatiquement une stratégie de déploiement prédéfinie à chaque modification du code.

7

L'audit et l'application de la stratégie sont ajoutés au pipeline CI/CD à l'aide d'Azure Policy.

8

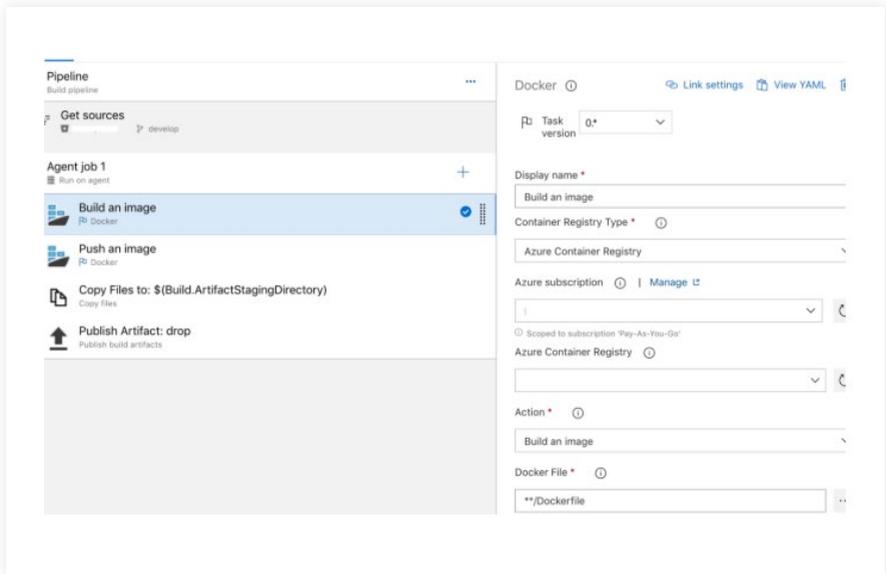
La surveillance de l'intégrité des conteneurs et l'analyse des journaux en temps réel sont effectuées via Azure Monitor.

9

Application Insights (Fonctionnalité de Azure Monitor) est utilisé pour résoudre les problèmes.

Ci-dessous le déploiement avec Azure Devops :

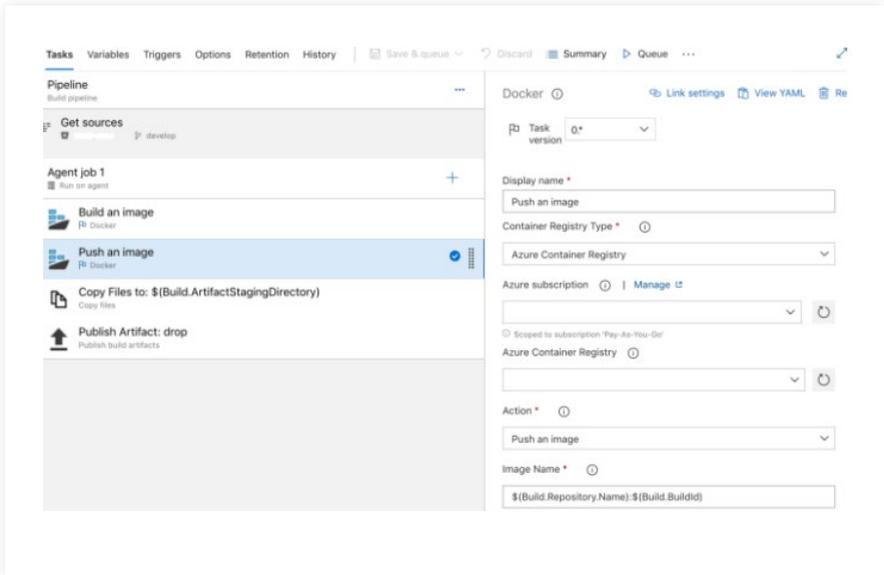
- **Récupérer les sources** depuis votre repo Azure Devops.
- **Définir ensuite quel type d'agent** doit exécuter le code.
- **Construire ensuite l'image** en cliquant sur « Build an Image » depuis la registry Docker dans la souscription Azure souhaitée ou depuis une registry public.



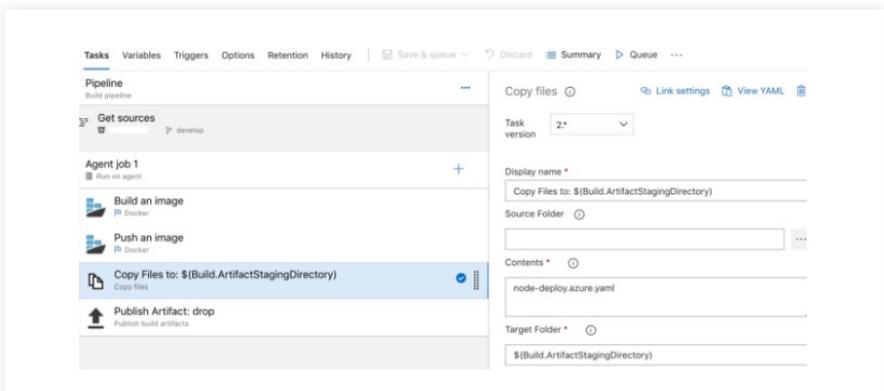
- **Choisir ensuite « Push an image »**, puis définir le nom de l'image en utilisant le pattern suivant :

`$(Build.Repository.Name):$(Build.BuildId)`

Cela permet de mettre le nom du référentiel source suivi de l'ID de la build, cela va nous permettre de récupérer le nom de l'image dans le déploiement.

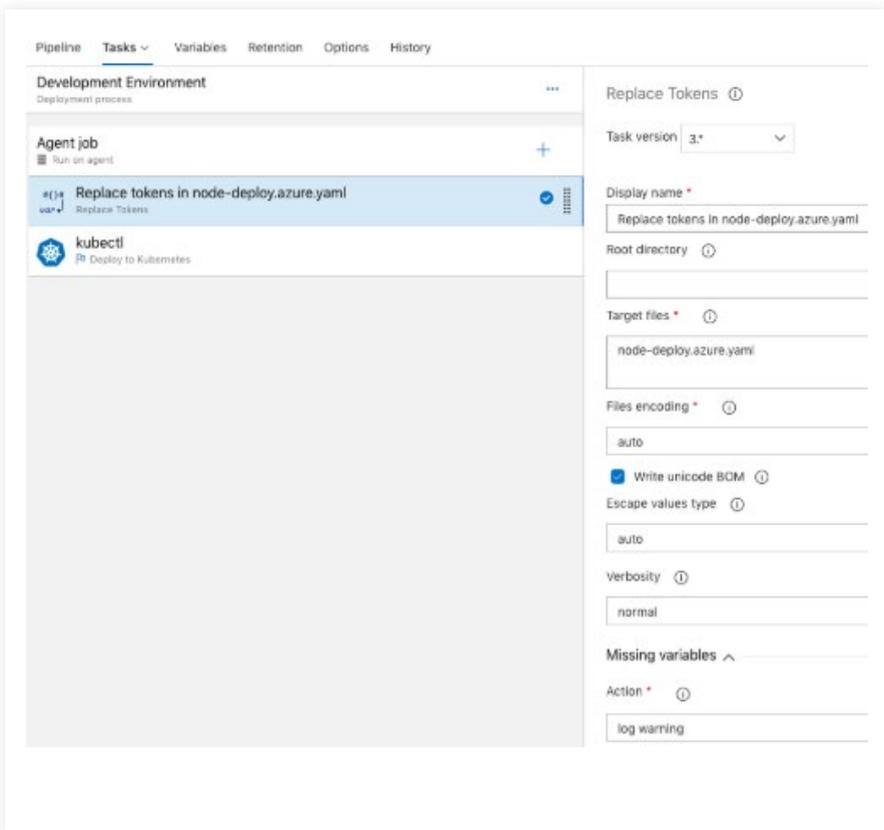


- **Sélectionner ensuite l'étape « Copy File »** pour transférer les artefacts vers un répertoire de staging.



- ***Il ne reste plus qu'à publier les artefacts de la build.***
Pour cela, nous allons créer un pipeline afin de pousser la bonne build dans le bon environnement

Sélectionnons le fichier de déploiement de l'artefact en utilisant les champs « Root Directory » et « Target Directory ».



Ensuite, vous pouvez utiliser la tâche Kubectl, cela lance la commande `kubectl Apply` et exécute votre déploiement.

Il nous reste maintenant à créer notre fichier de déploiement dont voici un exemple :

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  selector:
    app: my-service
  ports:
  - name: http
    port: 8000
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-service
  labels:
    version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-service
  template:
    metadata:
```

Ce déploiement crée un service http sur un port défini (port 8000), le déploiement utilisera le dernier build du repository.

Pour que Kubernetes puisse accéder au repository, nous allons configurer sa prise en charge ici :

```
az aks update -n myAKSCluster -g myResourceGroup --attach-acr <acrName>
```

Pour activer une extension azure, il faut utiliser cette commande, dans notre cas, aks-preview:

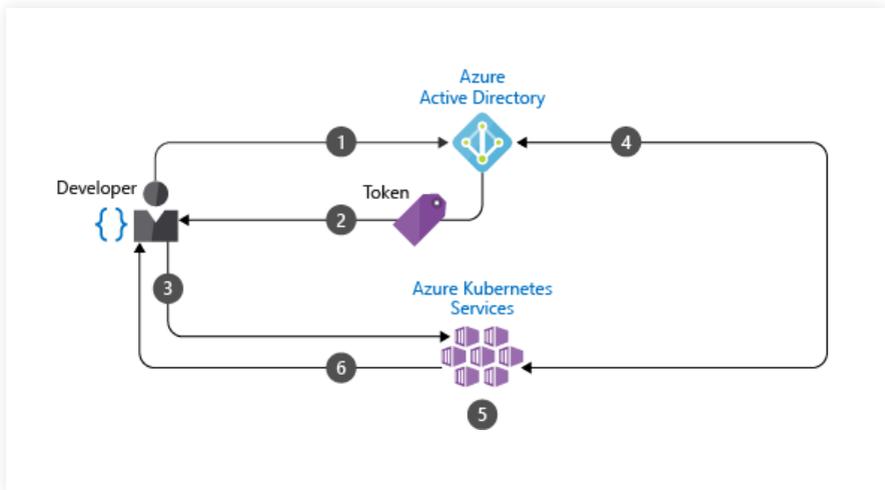
```
az extension add --name aks-preview
```

10. INTÉGRER KUBERNETES AVEC ACTIVE DIRECTORY

L'utilisation d'Azure AD permet de centraliser la gestion des identités. Utilisez les Rôles pour affecter aux utilisateurs ou aux groupes les autorisations nécessaires.

Kubernetes ne propose pas de solution de gestion des identités pour contrôler les utilisateurs pouvant interagir avec certaines ressources. Pour cela, vous devez intégrer votre cluster à une solution d'identité existante.

L'intégration d'Azure AD et le contrôle d'accès aux ressources sont représentés dans le diagramme suivant :



1

**Le développeur s'authentifie
auprès d'Azure AD.**

2

**Le point de terminaison d'émission de
jeton Azure AD émet le jeton d'accès.**

3

**Le développeur effectue une action
à l'aide du jeton Azure AD, par
exemple `kubectl create pod`.**

4

**Kubernetes valide le jeton auprès d'Azure
Active Directory et récupère (fetch) les
appartenances aux groupes du développeur.**

5

**Les stratégies relatives au cluster et au
contrôle d'accès en fonction du rôle
(RBAC) Kubernetes sont appliquées.**



La demande du développeur réussit ou non selon la validation précédente de l'appartenance au groupe Azure AD / les stratégies relatives au cluster & au RBAC Kubernetes. Le contrôle d'accès s'effectue en fonction des rôles (RBAC).

11. AZURE CONTAINER REGISTRY

Un registry Docker permet d'héberger des images Docker, nous allons voir comment mettre en place cela depuis Azure Cloud Shell.

Créer un groupe de ressources:

```
az group create --name rg_registry --location eastus
```

Créer une registry dans le groupe de ressource :

```
az acr create --resource-group rg_registry --name  
containeregistrypiermick --sku Basic --admin-enabled  
true
```

Se connecter ensuite sur la registry :

```
docker login --username containeregistrypiermick  
--password <passwd> containeregistrypiermick.azurecr.io
```

Lister les images : `docker images`.

Taguer ensuite l'image pour la registry :

```
docker tag azurepierzack containregistrypierzack.azurecr.io/azurepierzack:latest
```

Puis push ensuite cette image dans la registry :

```
docker push containregistrypierzack.azurecr.io/azurepierzack:latest
```

Créer ensuite l'image depuis la registry :

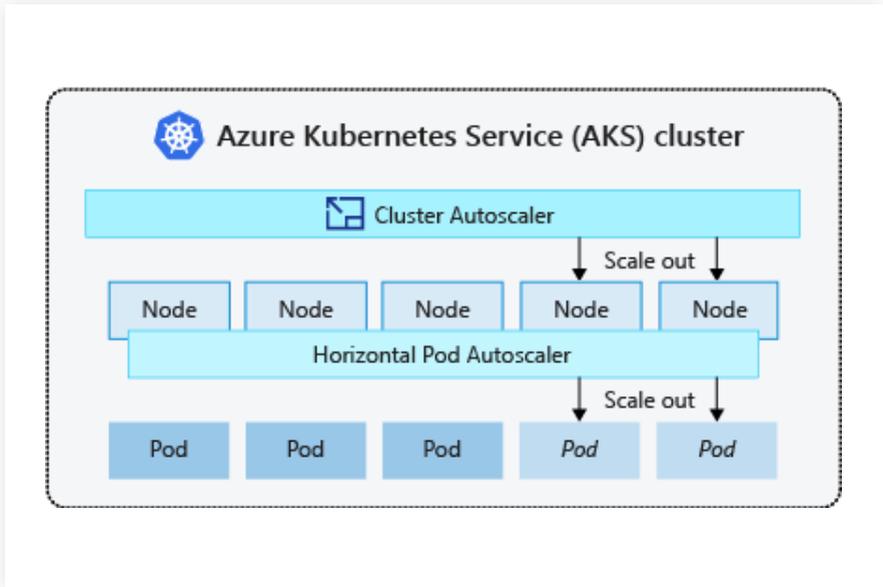
```
az container create --resource-group rg_registry --name azurepierzack --image containregistrypierzack.azurecr.io/azurepierzack:latest --dns-name-label azurepierzack --ports 80
```

Cette commande me permet d'accéder au container :

```
az container show --resource-group rg_registry --name  
instancelinuxwebsite --query "{FQDN:ipAddress.  
fqdn,ProvisioningState:provisioningState}" --out table
```

Connectez-vous ensuite sur votre navigateur avec l'url récupéré.

12. L'AUTOSCALING DES PODS



La montée en charge des pods dans Kubernetes nous permet d'ajouter ou de diminuer le nombre de Pods suivant l'utilisation de notre application. Cela permet ainsi de répondre à la variation de la demande.

La commande `kubectl scale` nous permet de modifier instantanément le nombre d'instances dont on souhaite disposer.

Dans l'exemple ci-dessous : 5.

```
kubectl scale --replicas=5 deployment/myApp
```

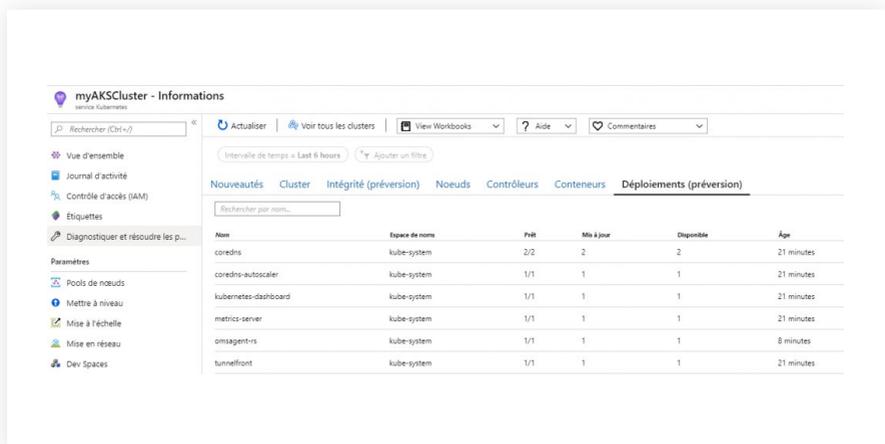
Pour provisionner un cluster AKS avec le monitoring activé, il faut utiliser cette commande :

```
az aks enable-addons -a monitoring -n MyExistingManagedCluster -g MyExistingManagedClusterRG
```

Le résultat sera le suivant :

```
provisioningState : Succeeded
```

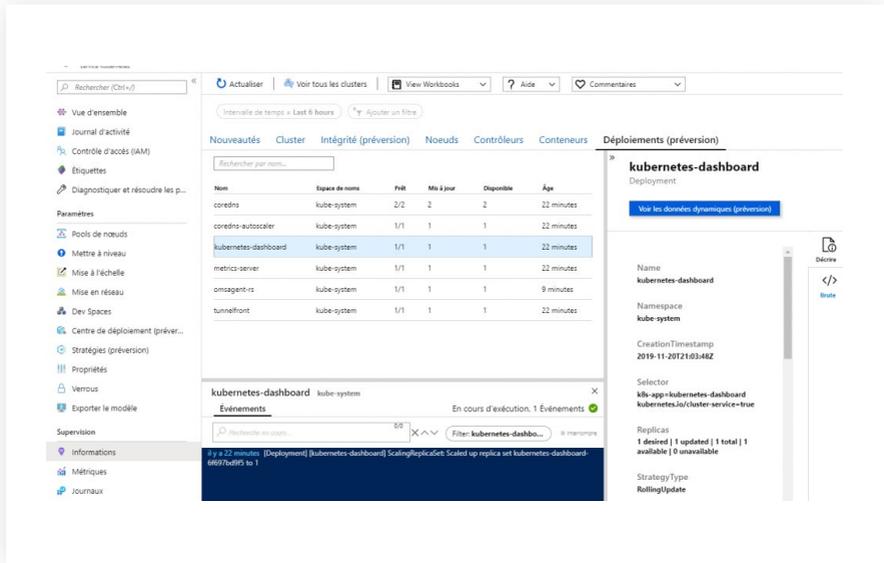
Pour monitorer, on va dans Azure Monitor :



The screenshot shows the Azure portal interface for monitoring an AKS cluster. The page title is 'myAKScluster - Informations'. The left sidebar contains navigation options like 'Vue d'ensemble', 'Journal d'activité', 'Contrôle d'accès (IAM)', 'Étiquettes', 'Diagnostic et résoudre les p...', 'Paramètres', 'Pools de nœuds', 'Mettre à niveau', 'Mise à l'échelle', 'Mise en réseau', and 'Dev Spaces'. The main content area shows the 'Déploiements (préversion)' tab selected, with a table listing the following deployments:

Nom	Espace de noms	Prêt	Mis à jour	Disponible	Âge
coredns	kube-system	2/2	2	2	21 minutes
coredns-autoscaler	kube-system	1/1	1	1	21 minutes
kubernetes-dashboard	kube-system	1/1	1	1	21 minutes
metrics-server	kube-system	1/1	1	1	21 minutes
omsagent-rs	kube-system	1/1	1	1	8 minutes
tunnelfront	kube-system	1/1	1	1	21 minutes

Il est aussi possible de visualiser les logs des Nodes et des Pods depuis le portail Azure :



Il est possible de définir des limites et des valeurs garanties aux ressources Kubernetes dans Azure.

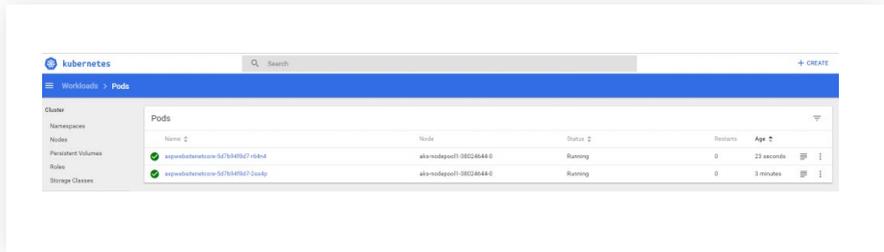
Requests étant la garantie des ressources disponibles pour le Pod, comme par exemple : 100m de CPU et 15Mi de RAM.

Limits étant les ressources maximales qu'il peut utiliser, s'il les dépasse, il sera détruit comme par exemple : 500m de CPU et 512Mi de RAM.

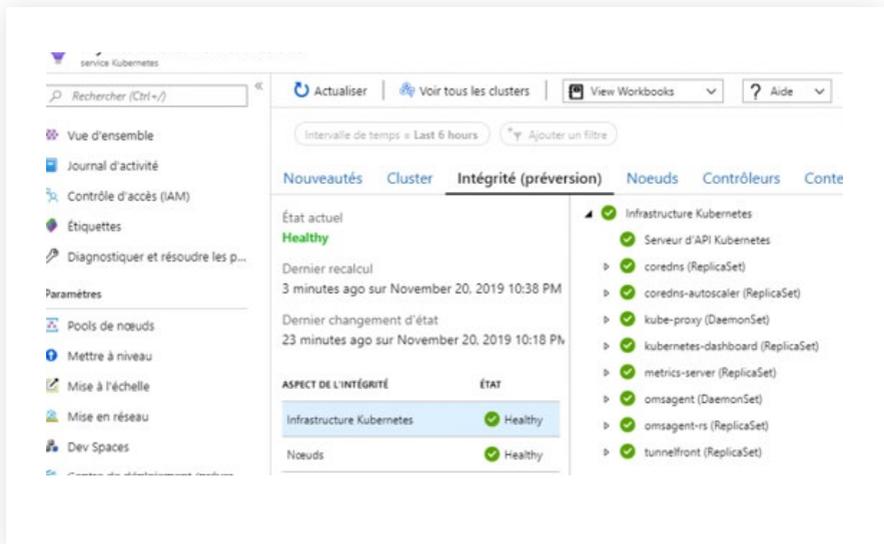
Dans l'exemple ci-dessous, L'upscale se fera si le pourcentage CPU consommé dépasse les 10% de ce qui est alloué.

```
kubectl autoscale deployment aspwebsitenetcore --max 10 --min 2 --cpu-percent 10
```

Les appels vers le site web seront ensuite automatiquement répartis vers ces deux pods via le loadbalancer du service.



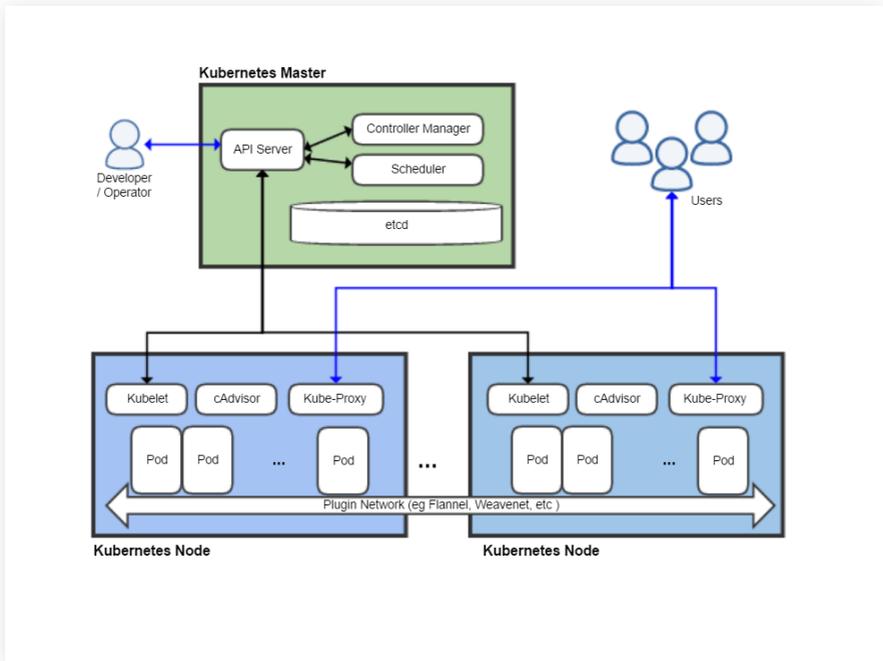
La commande `get hpa`, permet de visualiser les conteneurs par rapport à la limite fixée. Sur le portail Azure, l'onglet intégrité renseigne les informations sur l'état de santé du cluster.



On peut supprimer l'autoscaling du cluster AKS grâce via la commande `delete hpa [nom_hpa]`.

13. L'AUTOSCALING DES NODES

Après le scaling des pods, nous allons parler du scaling des nodes, possible grâce à l'autoscaler de cluster.



- › Création du cluster

```
az group create --name aks --location eastus
az aks create --resource-group aks --name myAKS --node-count 1 --generate-ssh-keys
```

Voici la commande permettant de scaler un pool en spécifiant le nom du pool :

```
az aks scale --resource-group aks --name myAKS --node-count 3 --nodepool-name #NodePoolName#
```

Voici la commande pour avoir le descriptif d'un pool :

```
az aks show --resource-group aks --name myAKS --query agentPoolProfiles
```

La commande pour lister les nodes d'un pool :

```
az aks nodepool list --resource-group aks --cluster-name myAKS
```

Voici la commande si vous n'avez qu'un seul pool pour scaler le nombre de node à 3 :

```
az aks scale --resource-group aks--name myAKS --node-count 3
```

Pour mettre en place un autoscale automatique, on provisionne un cluster AKS avec la prise en charge de l'autoscaling des nodes, pour cela, il faut spécifier une taille minimale et maximale pour le pool de node.

Le scaling des nodes est ensuite effectué automatiquement si les ressources du cluster sont insuffisantes dans la limite du scale défini dans le cluster.

Source : <https://docs.microsoft.com/fr-fr/azure/aks/cluster-autoscaler>

```
az aks create --resource-group aks --name myAKSWithAutoscale  
--node-count 1 --vm-set-type VirtualMachineScaleSets  
--enable-cluster-autoscaler --min-count 1 --max-count 3
```

Ce cluster scale le nombre de node entre 1 et 3.

On peut faire *évoluer* une configuration existante via la commande ***-update-cluster-autoscaler*** :

```
az aks update -resource-group aks -name myAKSWithAutoscale  
-update-cluster-autoscaler -min-count 1 -max-count 5
```

Voici la commande permettant de supprimer un autoscale :

```
az aks update --resource-group aks --name myAKSWithAutoscale  
--disable-cluster-autoscaler
```

La mise à l'échelle automatique de clusters écrit également l'état d'intégrité sur un élément ConfigMap nommé cluster-autoscaler-status. voici la commande pour récupérer ces journaux :

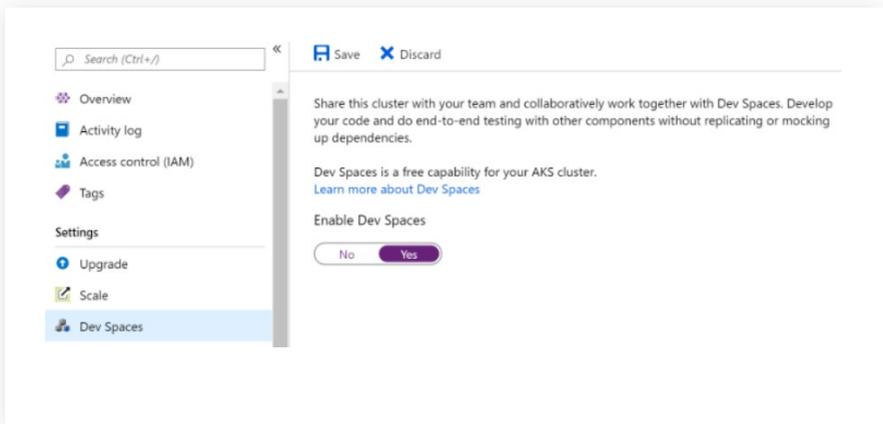
```
kubectl get configmap -n kube-system cluster-autoscaler-  
status -o yaml
```

Si notre cluster n'a pas été configuré avec un autoscale, il est possible de l'activer via la commande suivante :

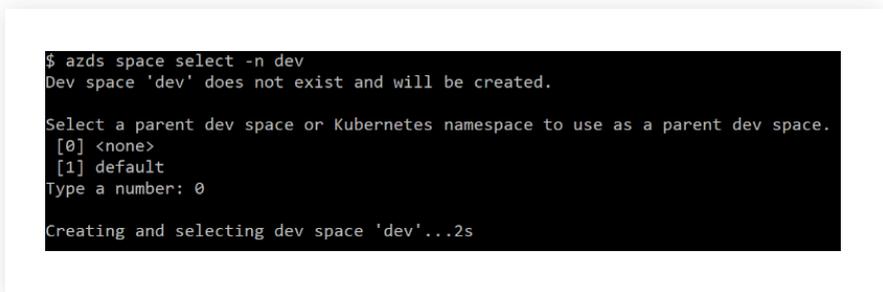
```
az aks nodepool update --resource-group aks --cluster-  
name myAKS --name nodepool1 --enable-cluster-autoscaler  
--min-count 1 --max-count 5
```

14. PRÉSENTATION DE AZURE DEV DPACES

En parallèle Microsoft a mis en place un nouveau service nommé Azure Dev Spaces permettant à toute votre équipe de développement de partager un cluster AKS, au lieu de requérir des environnements distincts pour chaque développeur et leur permettre ainsi de pouvoir tester leur application de bout en bout. Cette option est à activer depuis le portail Azure.



Il faut ensuite spécifier le namespace ou l'espace de nom dans lequel s'exécute Dev Spaces comme ci-dessous :



Pour créer un espace de développement enfant appelé newfeature dans l'exemple, on donne la référence parent créée précédemment DEV en utilisant le choix 2 :

```
$ azds space select -n newfeature
Dev space 'newfeature' does not exist and will be created.

Select a parent dev space or Kubernetes namespace to use as a parent dev space.
 [0] <none>
 [1] default
 [2] dev
Type a number: 2

Creating and selecting dev space 'dev/newfeature'...3s
```

Ce nouvel environnement aura sa propre URL d'accès.

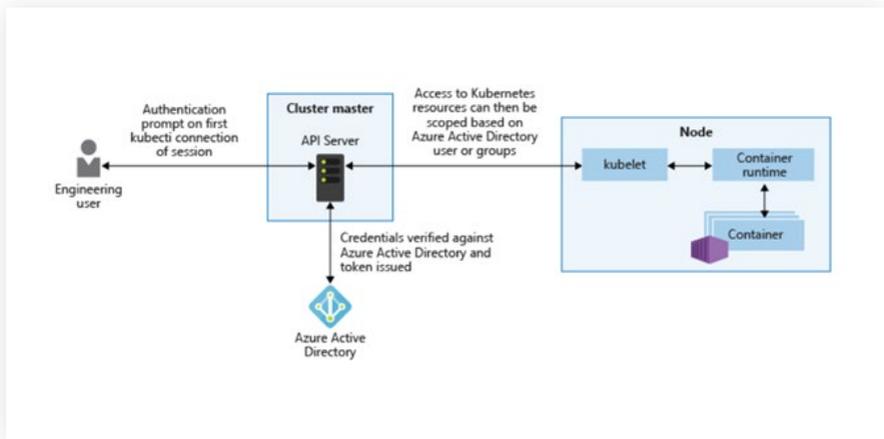
15. LA SÉCURITÉ DANS AKS

La sécurisation de l'accès au serveur d'API Kubernetes est une tâche importante pour protéger votre cluster. Pour cela, il faut utiliser le principe de moindre privilège pour le compte de service.

Comme c'est un service managé la sécurité des nodes Master est géré par Microsoft. La sécurité des nodes AKS est de notre ressort, ce sont des machines virtuelles on doit en assurer la gestion et la maintenance.

Les secrets sont stockés dans la base de données etcd qui utilise le format clés/valeurs. AKS chiffre la base de données etcd au repos et Microsoft gère les clés de chiffrement.

Quand un cluster AKS est créé ou fait l'objet d'un scale-up, les nodes sont déployés automatiquement avec les dernières configurations et mises à jour de sécurité du système d'exploitation.

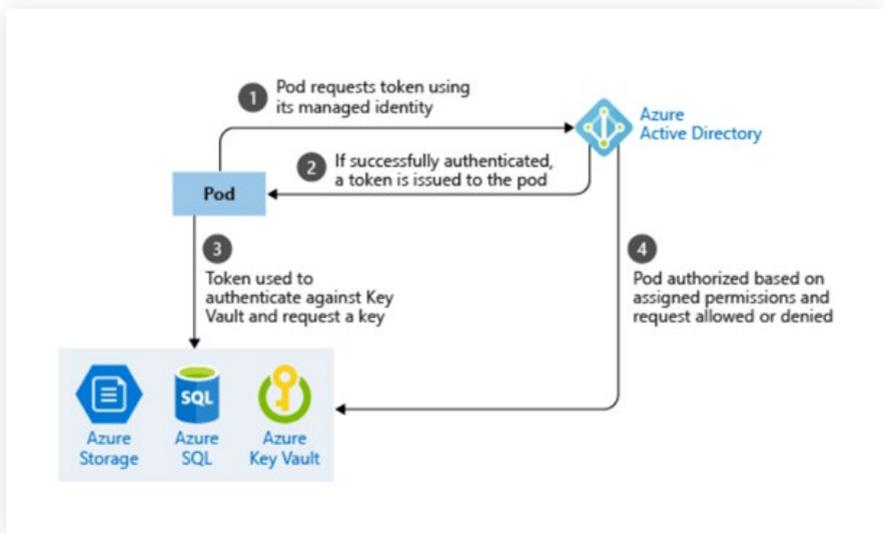


A savoir que pour les nodes Windows Server, Windows Update n'exécute pas et n'applique pas automatiquement les dernières mises à jour.

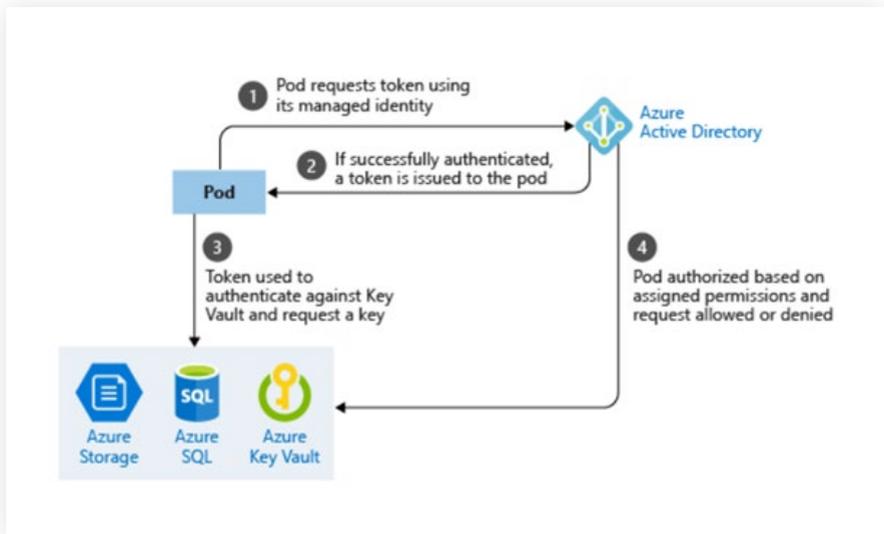
Les nodes sont déployés sur un sous-réseau de réseau virtuel privé, sans aucune adresse IP publique affectée. Pour des raisons de gestion et de résolution des problèmes, SSH est activé par défaut. Cet accès SSH n'est disponible qu'au moyen de l'adresse IP interne.

Pour filtrer le flux du trafic dans les réseaux virtuels, Azure utilise des network policies. Ces règles définissent les plages d'adresses IP source et de destination, les ports et les protocoles qui se voient autoriser ou refuser l'accès aux ressources. Des règles par défaut sont créées pour autoriser le trafic TLS vers le serveur d'API Kubernetes.

Si vos services ou applications n'ont pas d'identités managées, vous pouvez toujours utiliser des informations d'identification ou des clés pour vous authentifier. Pour cela, vous pouvez utiliser Azure KeyVault.



Ci-dessous le workflow utilisé pour récupérer des informations d'identification à partir d'Azure Key Vault à l'aide d'identités de pod managées. Key Vault vous permet de stocker et effectue la rotation des secrets, tels que les informations d'identification, les clés de compte de stockage ou les certificats :



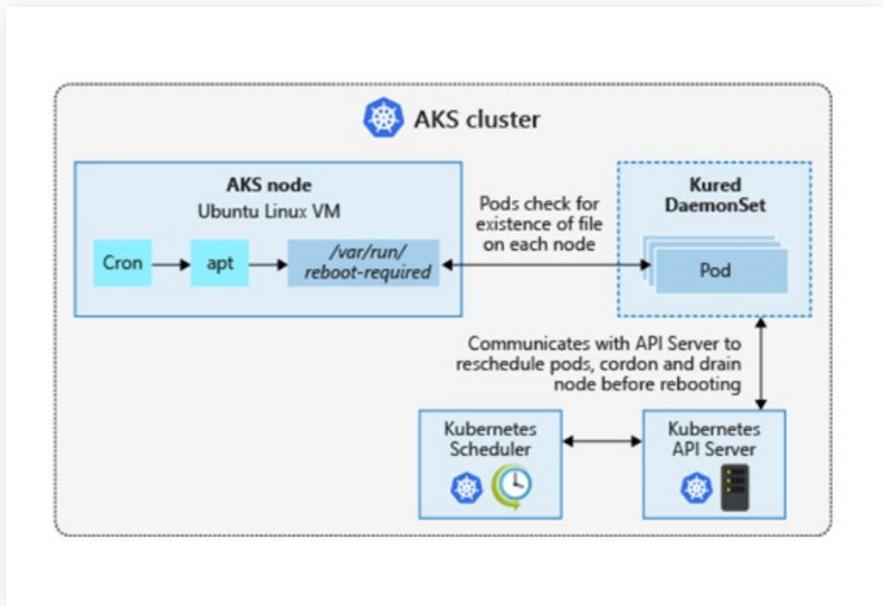
La commande suivante permet de consulter les mises à jour kubernetes disponibles pour votre cluster :

```
az aks get-upgrades --resource-group myResourceGroup --name myAKSCluster
```

Vous pouvez ensuite mettre à niveau votre cluster AKS à l'aide de la commande :

```
az aks upgrade --resource-group myResourceGroup --name myAKSCluster --kubernetes-version KUBERNETES_VERSION
```

AKS télécharge et installe automatiquement les correctifs de sécurité sur chacun des nodes Linux, mais ne les redémarre pas automatiquement. Utilisez kured pour surveiller les redémarrages en attente, Kured est un projet open source de Weaveworks.



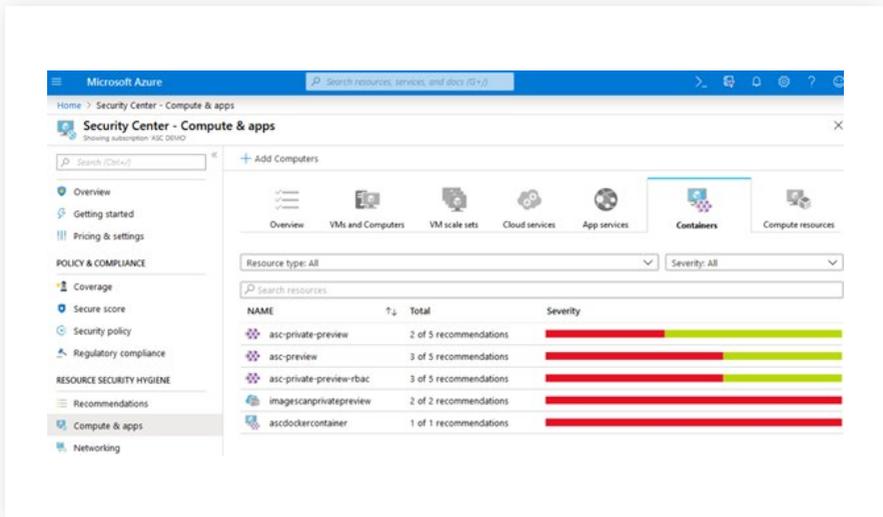
16. AKS AVEC SECURITY CENTER

Kubernetes est en train de devenir la nouvelle norme en matière de déploiement et de gestion de logiciels dans le cloud.

Azure Security Center est la solution Azure native pour la sécurité des conteneurs, elle constitue aussi un point de contrôle unique de la sécurité de vos charges de travail cloud, machines virtuelles, serveurs et conteneurs.

Ci-dessous les principaux aspects de la sécurité des conteneurs :

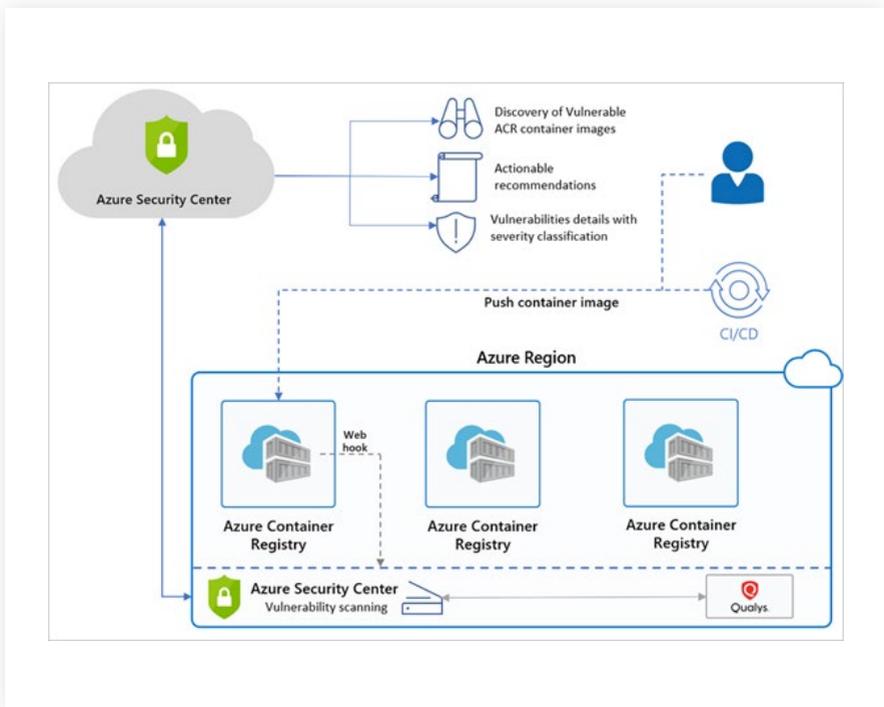
- Gestion des vulnérabilités
- Sécurisation renforcée de l'environnement des conteneurs
- Protection du Runtime



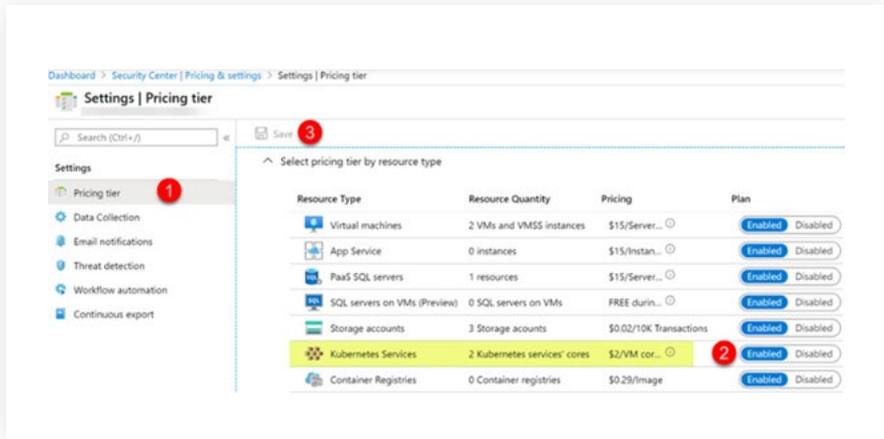
Si des problèmes sont détectés par Qualys ou Security Center, vous aurez une notification dans le tableau de bord. Security Center inclut la totalité des règles définies dans le CIS Docker Benchmark qui est le document de référence pour l'audit et la sécurité des environnements kubernetes.

Source : https://docs.docker.com/compliance/cis/docker_ce/

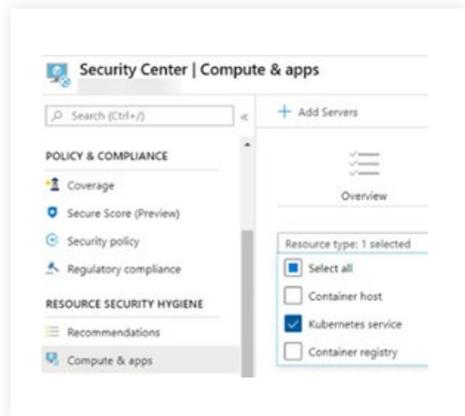
Azure Security Center vous envoie une alerte si vos conteneurs ne satisfont pas à tous les contrôles. Quand il détecte des configurations incorrectes, Security Center génère des recommandations de sécurité. De ce fait, Security Center assure une détection des menaces en temps réel pour vos environnements conteneurisés et génère des alertes en cas d'activités suspectes.



Pour activer Security Center pour AKS, depuis le portail Azure, aller dans **Security Center** dans le bandeau de gauche. Sélectionner ensuite **Pricing Tiers** puis activer **Kubernetes Services** et valider



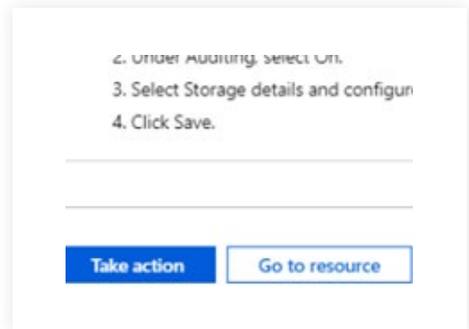
par **Save**. Une fois la protection activé, Security Center est prêt à sécuriser votre infrastructure. Les recommandations sont visibles dans **Compute & apps**, onglet **Container** puis faire un filtre sur **Kubernetes service**.



Voici un exemple de problème remonté dans Azure Security Center :

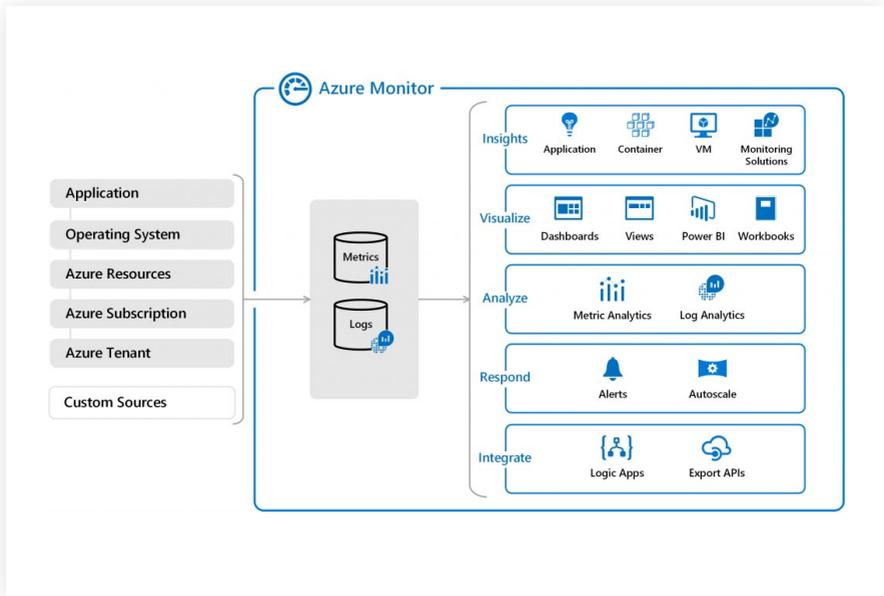


Pour corriger des problèmes, Security Center vous fait des recommandations qui peuvent être appliqués directement via le bouton ***Take Action*** :



17. AZURE MONITOR POUR AKS

Azure Monitor va être utile pour superviser notre cluster AKS au niveau mémoire et CPU pour les nodes et pods ainsi que visualiser les logs du cluster et des conteneurs.



L'activation d'Azure monitor se fait lors de la création du cluster dans l'onglet Supervision.

Créer un cluster Kubernetes

[De base](#) [Mise à l'échelle](#) [Authentification](#) [Mise en réseau](#) [Supervision](#) [Étiquettes](#)

Azure Kubernetes Service vous fournit des métriques d'utilisation de processeur et de mémoire pour et vous pouvez activer des fonctionnalités de supervision de conteneurs et obtenir des insights sur les pe de votre cluster Kubernetes. Vous êtes facturé en fonction de la quantité de données ingérées et de vo: conservation des données.

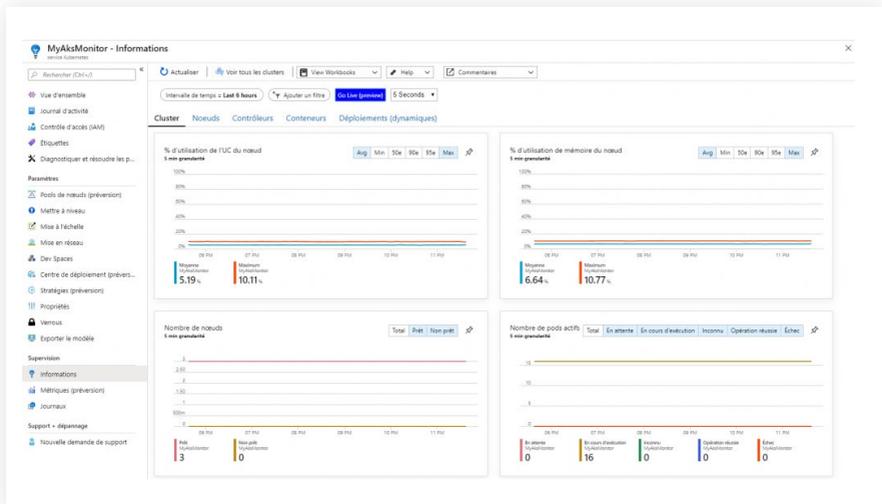
[En savoir plus sur les performances et la supervision de l'intégrité des conteneurs](#)
[En savoir plus sur les tarifs](#)

Azure Monitor

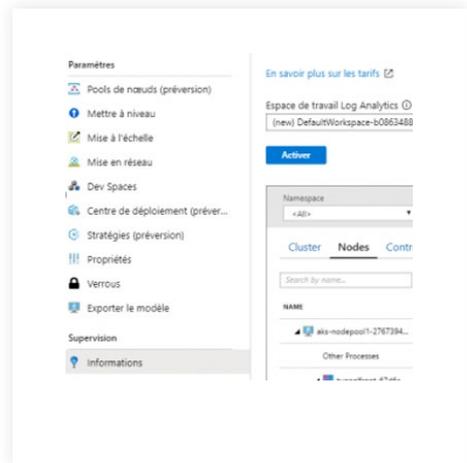
Activer la supervision de conteneurs

Non Oui

Dans le menu Informations du cluster AKS, on peut retrouver les métriques :



Vous pouvez bien sûr activer cette option après avoir créé votre cluster en cliquant sur **Activer** du menu Information.



Pour activer Azure Monitor avec Azure CLI, il faut utiliser la commande suivante :

```
az aks create --resource-group rg_aks_monitor_cli --name myAKSCluster --enable-addons monitoring
```

Pour ajouter le monitoring à un AKS existant, il faut utiliser la commande suivante :

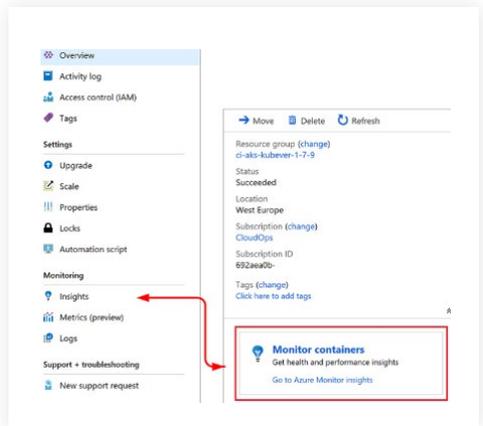
```
az aks create --resource-group rg_aks_monitor_cli --name myAKSClusterWithoutMonitor
az aks enable-addons -a monitoring -n MyAKSClusterWithoutMonitor -g rg_aks_monitor_cli
```

Pour vérifier qu'un Azure Monitor est bien configuré et actif pour votre AKS, utilisez cette commande :

```
kubectl get deploy --all-namespaces puis contrôlez la présence de omsagent-rs
```

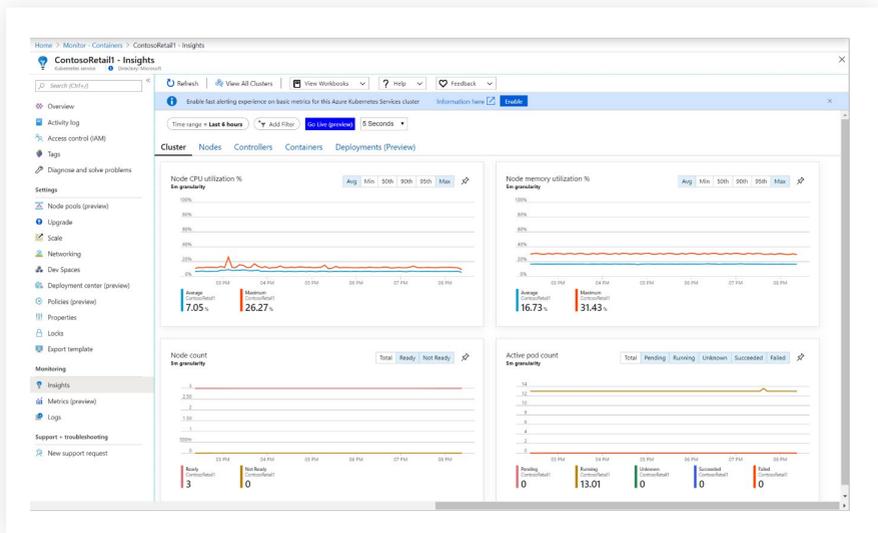
```
kube-system metrics-server 1 1 1 1 44d
kube-system omsagent-rs 1 1 1 1 14m
kube-system tunnelfront 1 1 1 1 44d
```

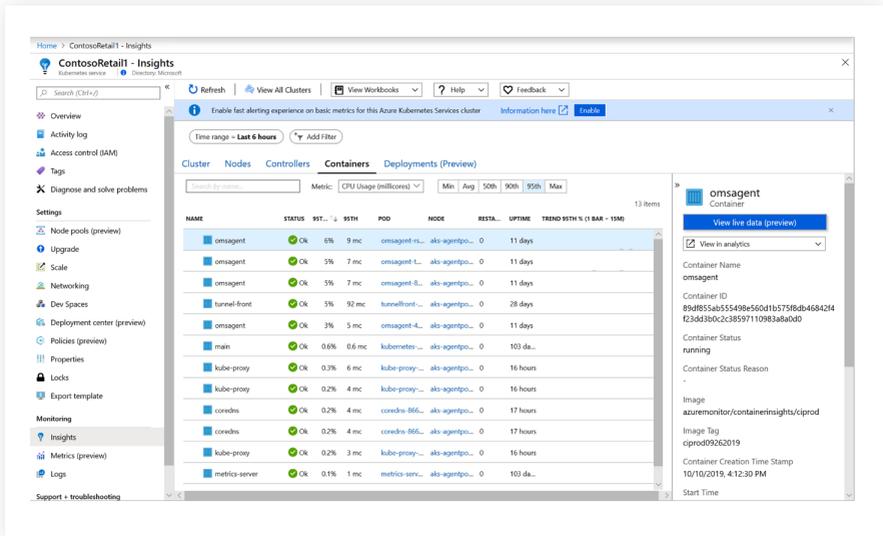
Depuis le portail Azure, sélectionner **Insights** puis **Monitor Container**.



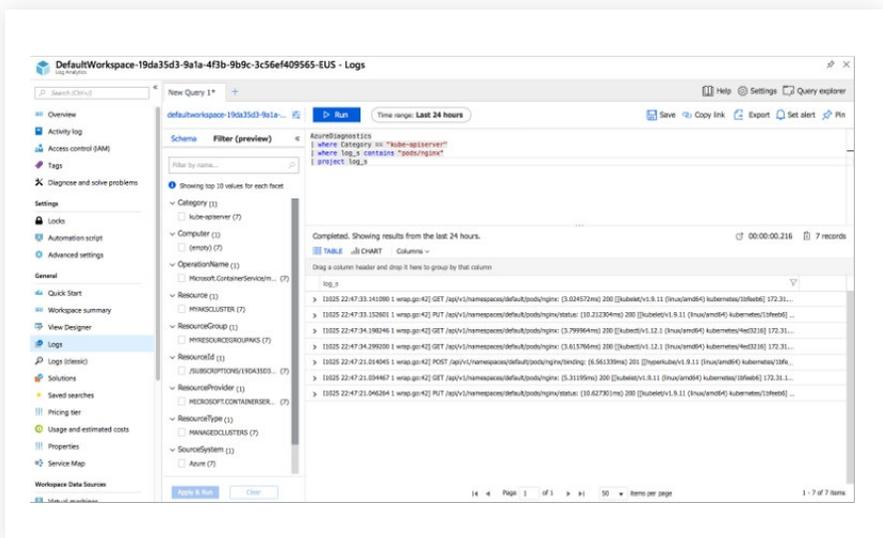
La vue Cluster nous permet de voir les graphs de supervision de notre environnement AKS :

- % d'utilisation CPU
- % d'utilisation mémoire
- Nombre de nodes actifs
- Nombre de pods actifs





La vue Journaux nous permet de faire une recherche de logs :



18. TROUBLESHOOTING

Le site de Microsoft nous donne pas mal d'infos sur la résolution de problèmes pouvant être rencontrés dans AKS.

<https://docs.microsoft.com/fr-fr/azure/aks/troubleshooting>



CONCLUSION.

Le principal avantage de Kubernetes est la possibilité de déployer un environnement automatisé, via terraform par exemple, en production sans l'asservissement à un fournisseur de Cloud.

C'est en effet une des raisons pour laquelle les entreprises migrent vers le cloud. La solution Kubernetes rassure sur le fait qu'il y ait une possibilité de faire machine arrière et de redéployer on-premises l'infrastructure déployée dans le Cloud ou donne la possibilité de redéployer cette même infrastructure chez un autre fournisseur dans le cloud tout en limitant les modifications.

Il est par exemple possible d'utiliser Kubernetes dans un Cloud hybride et donc d'utiliser plusieurs plateformes.

L'utilisation de Kubernetes dans le Cloud Azure permet de se concentrer sur la gestion et la création de nos applications sans se soucier de la partie mise en place, configuration et supervision des noeuds d'infrastructure (master node) de notre cluster Kubernetes.

De ce fait, la containerisation est devenue incontournable comme l'a été avant la virtualisation pour l'optimisation des ressources serveurs et le Cloud pour la rapidité de déploiement et la flexibilité induite par les services managés, entre autres.

Sources :

<https://docs.microsoft.com/fr-fr/azure/developer/terraform/create-k8s-cluster-with-tf-and-aks>

https://www.terraform.io/docs/providers/azurerm/r/kubernetes_cluster.html

<https://www.hashicorp.com/blog/kubernetes-cluster-with-aks-and-terraform/>



**Cette solution permet aussi
de surveiller,
à tout moment, l'état des
conteneurs fonctionnant sur
des machines virtuelles,
& de compenser directement
une panne.**



CERTIFICATIONS MICROSOFT & KUBERNETES

Certification Microsoft :

Cours Microsoft

- › [*AZ-203-Developing Solutions for Microsoft Azure*](#)
- › [*AZ-204-Developing Solutions for Microsoft Azure*](#)

Cours Whizlabs

- › [*Microsoft Azure Exam AZ-203 Practice Tests & Online Course*](#)
- › [*Microsoft Azure Exam AZ-204 Certification*](#)



Certification Kubernetes :

Cours Whizlabs

- › [*Certified Kubernetes Administrator \(CKA\)*](#)

Cours UDEMY

- › [*Certified Kubernetes Administrator \(CKA\) Practice Exam Tests*](#)
- › [*Introduction à Kubernetes*](#)
- › [*Kubernetes : Les bases indispensables*](#)



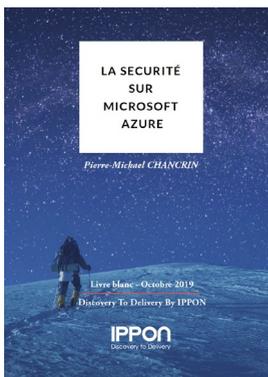
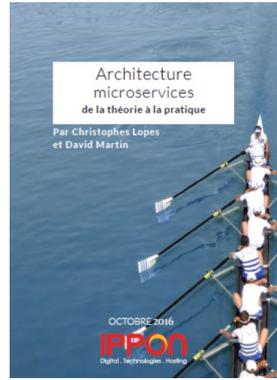
labs en ligne :

Ces labs permettent de tester les fonctionnalités de Kubernetes sans avoir installer et configurer d'environnement sur votre poste comme Minikube par exemple

- › [*Katakoda*](#)
- › [*Play with Kubernetes*](#)



Découvrir tous les livres blancs.



A propos d'Ippon Technologies.

Créé en 2002, Ippon Technologies est un cabinet de conseil qui accélère les produits innovants et la stratégie digitale de ses clients.

Ippon Technologies accompagne des entreprises de secteurs et de tailles variés (startups, grands groupes, acteurs publics) dans le développement et la transformation de leur système d'information avec des applications performantes et des solutions robustes.

Notre proposition de valeur à 360° permet de répondre à l'ensemble des besoins en innovation technologique.

Implanté sur 4 continents, Ippon Technologies assure un rayonnement et un partage d'expérience international.

Ippon Accelerator

Votre projet de la page blanche
au cloud en quelques semaines



Discovery to delivery.

Innover & optimiser votre TIME TO MARKET



Licence.



Ce document vous est fourni sous licence Creative Commons Attribution Share Alike. Le texte ci-dessous est un résumé (et non pas un substitut) de la licence.

Plus d'informations sur www.creativecommons.org/licenses.

Vous êtes autorisé à :

- › Partager — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats.
- › L'offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.
- › Selon les conditions suivantes :

Attributions :

- › Vous devez créditer l'oeuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'oeuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'offrant vous soutient ou soutient la façon dont vous avez utilisé son oeuvre.
- › Pas d'utilisation commerciale — vous n'êtes pas autorisé à faire un usage commercial de cette oeuvre, tout ou partie du matériel la composant.
- › Pas de modifications — dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'oeuvre originale, vous n'êtes pas autorisé à distribuer ou mettre à disposition l'oeuvre modifiée.
- › Pas de restrictions complémentaires — vous n'êtes pas autorisé à appliquer des conditions légales ou des mesures techniques qui restreindraient légalement autrui à utiliser l'oeuvre dans les conditions décrites par la licence.

Notes :

- › Vous n'êtes pas dans l'obligation de respecter la licence pour les éléments ou matériel appartenant au domaine public ou dans le cas où l'utilisation que vous souhaitez faire est couverte par une exception.
- › Aucune garantie n'est donnée. Il se peut que la licence ne vous donne pas toutes les permissions nécessaires pour votre utilisation. Par exemple, certains droits comme les droits moraux, le droit des données personnelles et le droit à l'image sont susceptibles de limiter votre utilisation.



IPPON

Discovery to Delivery

fr.ippon.tech

blog.ippon.fr

medium.com/ippon

contact@ippon.fr

+33 1 46 12 48 48

@ipponentech



AUSTRALIE

Melbourne



USA

New York, NY

Washington, DC

Richmond, VA



FRANCE

Paris

Bordeaux

Nantes

Lyon

Toulouse

Lille



RUSSIE

Moscou