

Technologies Cloud et Sécurité - AZURE - Partie II

Anis Hanniz

30 novembre 2024

Table des matières

1	Services de Conteneurs Azure	3
1.1	Comparaison ACI vs ACA vs AKS	3
2	Architecture Conteneurs et Kubernetes	3
2.1	Concepts Fondamentaux	3
2.2	Ingress Controller	3
2.3	Azure Monitor vs Azure Analytics Workspace	4
3	Backup et Disaster Recovery	4
3.1	Solutions Kubernetes	4
4	Guide Pratique AKS	4
4.1	Configuration Initiale et Création du Cluster	4
5	Gestion des Node Pools	5
5.1	Déploiement d'Applications	6

1 Services de Conteneurs Azure

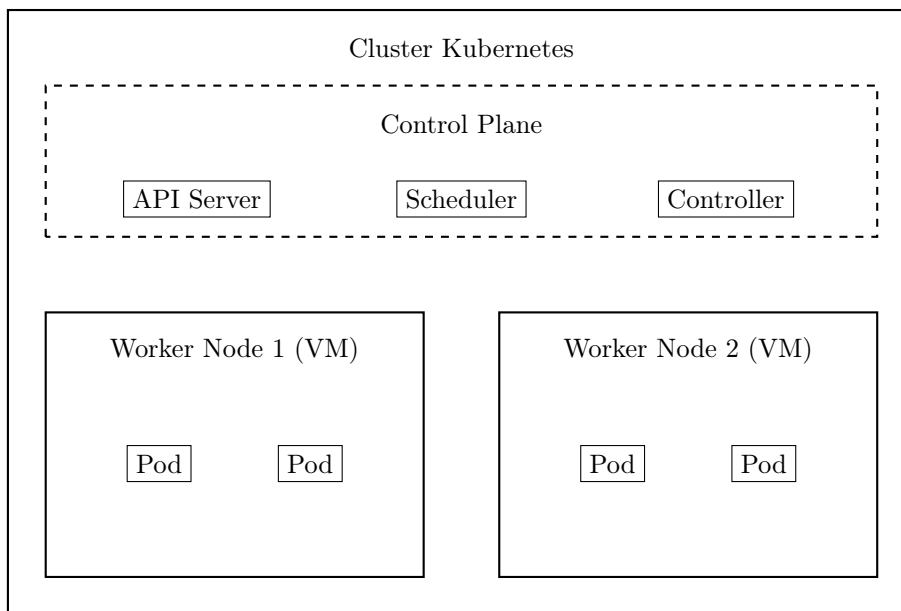
1.1 Comparaison ACI vs ACA vs AKS

Caractéristique	ACI	ACA	AKS
Type	Serverless	Serverless + Orchestration légère	Orchestration complète
Usage	Dev/Test, Tâches ponctuelles	Applications modérées, Microservices	Production complexe, Grande échelle
Scaling	Manuel	Auto-scaling basé sur métriques	Auto-scaling avancé, HPA
Complexité	Simple	Moyenne	Élevée
Coût	Pay-as-you-go	Pay-as-you-go + Services managés	Nodes + Services managés
Staging	Non	Oui (Révisions)	Oui (Namespaces)

2 Architecture Conteneurs et Kubernetes

2.1 Concepts Fondamentaux

- **Conteneurs** : Unités d'application isolées
- **Pods** : Plus petite unité déployable, peut contenir plusieurs conteneurs
- **Node** : Machine virtuelle hébergeant des pods
- **TANZU** : Solution VMware pour Kubernetes enterprise



2.2 Ingress Controller

- Gestion avancée du routage
- Configuration des redirections URL
- SSL/TLS termination
- Load balancing applicatif

2.3 Azure Monitor vs Azure Analytics Workspace

Azure Monitor	Log Analytics Workspace
Collecte temps réel	Analyse approfondie
Métriques de base	Requêtes KQL complexes
Alertes simples	Corrélation de logs
Visualisation basique	Insights avancés
Focus performance	Focus troubleshooting

3 Backup et Disaster Recovery

3.1 Solutions Kubernetes

- **Veeam Kasten :**
 - Backup natif K8s
 - Restore granulaire
 - Support multi-cloud
 - Automation policies
- **Portworx PX-Backup :**
 - Backup applications stateful
 - Intégration CI/CD
 - Disaster recovery
 - Réplication cross-cluster

4 Guide Pratique AKS

4.1 Configuration Initiale et Création du Cluster

```
1 # Configuration des variables d'environnement
2 export RESOURCE_GROUP=rg-contoso-video
3 export CLUSTER_NAME=aks-contoso-video
4 export LOCATION=eastus
5
6 # Création du groupe de ressources
7 az group create --name=$RESOURCE_GROUP --location=$LOCATION
8
9 # Création du cluster AKS avec support Windows
10 az aks create \
11     --resource-group $RESOURCE_GROUP \
12     --name $CLUSTER_NAME \
13     --node-count 2 \
14     --generate-ssh-keys \
15     --node-vm-size Standard_B2s \
16     --network-plugin azure \
17     --windows-admin-username localadmin
```

5 Gestion des Node Pools

```
1 # Ajout d'un pool de noeuds Windows
2 az aks nodepool add \
3     --resource-group $RESOURCE_GROUP \
4     --cluster-name $CLUSTER_NAME \
5     --name uspool \
6     --node-count 2 \
7     --node-vm-size Standard_B2s \
8     --os-type Windows
9
10 # Récupération des informations d'identification
11 az aks get-credentials \
12     --name $CLUSTER_NAME \
13     --resource-group $RESOURCE_GROUP
14
15 # Vérification des noeuds
16 kubectl get nodes
17
18 # Affichage détaillé des noeuds avec OS
19 kubectl get nodes -o custom-columns=NAME:.metadata.name,\
20 TYPE:.metadata.labels.kubernetes\.azure\.com/mode,\
21 OS:.status.nodeInfo.osImage
22
23 # Liste des clusters AKS
24 az aks list --query "[].{Name:name,ResourceGroup:resourceGroup}" -o table
```

5.1 Déploiement d'Applications

```
1 # deployment.yaml
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: contoso-website
6 spec:
7   selector:
8     matchLabels:
9       app: contoso-website
10  template:
11    metadata:
12      labels:
13        app: contoso-website
14    spec:
15      nodeSelector:
16        kubernetes.io/os: linux
17      containers:
18      - image: mcr.microsoft.com/mslearn/samples/contoso-website
19        name: contoso-website
20        resources:
21          requests:
22            cpu: 100m
23            memory: 128Mi
24          limits:
25            cpu: 250m
26            memory: 256Mi
27        ports:
28        - containerPort: 80
29          name: http
```

```
1 # Application du déploiement
2 kubectl apply -f ./deployment.yaml
3
4 # Nettoyage du contexte
5 kubectl config delete-context aks-contoso-video
```