

實驗模組 2：Azure Kubernetes Service



預計時長：60 分鐘

模組 2 目錄

[練習：建立支援性資源](#)

[練習：建立 ACR 和範例工作負載映像](#)

[練習：使用 Azure CLI 建立 AKS叢集](#)

[練習：將工作負載部署到節點](#)

[練習：擴縮節點以滿足需求](#)

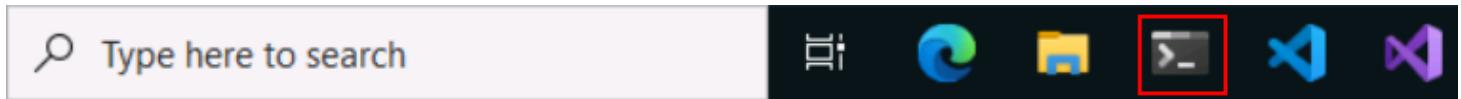
[練習：檢視容器洞察](#)

[練習：清理資源](#)

練習：建立支援性資源

任務 1 - 連線到 Azure 訂閱並安裝所需提供者

1. 打開 Windows Terminal（預設 PowerShell）。



Windows Terminal 允許你開啟多標籤命令終端。

2. 登入 Azure。

```
az login
```

3. 設定目前訂閱。

```
az account set --subscription "Azure Subscription XXX"
```

4. 註冊所需的提供者（如果在實驗 1 中未執行）。

```
az provider register --namespace Microsoft.Storage  
az provider register --namespace Microsoft.Compute  
az provider register --namespace Microsoft.Network  
az provider register --namespace Microsoft.Monitor  
az provider register --namespace Microsoft.ManagedIdentity  
az provider register --namespace Microsoft.OperationalInsights  
az provider register --namespace Microsoft.OperationsManagement  
az provider register --namespace Microsoft.KeyVault  
az provider register --namespace Microsoft.ContainerService  
az provider register --namespace Microsoft.Kubernetes
```

5. 打開瀏覽器造訪 portal.azure.com。

6. 搜尋並開啟 **訂閱**，選擇 **Azure Subscription XXX** 訂閱。

7. 向下捲動選擇 **資源提供者**。

The screenshot shows the Azure Pass - Sponsorship | Resource providers page. On the left, there's a sidebar with navigation links like Programmatic deployment, Resource groups, Resources, Preview features, Usage + quotas, Policies, Management certificates, My permissions, Resource providers (which is selected), Deployments, Properties, and Resource locks. At the top, there are buttons for Register, Unregister, Refresh, and Feedback. A search bar at the top left says 'Search (Ctrl+ /)'. A filter bar at the top right says 'Filter by name...'. The main area is a table with two columns: 'Provider' and 'Status'. The providers listed are Microsoft.Storage, Microsoft.Compute, Microsoft.Monitor, Microsoft.OperationalInsights, Microsoft.OperationsManagement, Microsoft.ManagedIdentity, Microsoft.KeyVault, Microsoft.Network, and Microsoft.ContainerService. The status for most providers is 'Registered', except for Microsoft.Compute, Microsoft.OperationalInsights, Microsoft.OperationsManagement, Microsoft.KeyVault, Microsoft.Network, and Microsoft.ContainerService, which are all in the 'Registering' state.

Provider	Status
Microsoft.Storage	Registered
Microsoft.Compute	Registering
Microsoft.Monitor	Registered
Microsoft.OperationalInsights	Registering
Microsoft.OperationsManagement	Registering
Microsoft.ManagedIdentity	Registered
Microsoft.KeyVault	Registering
Microsoft.Network	Registering
Microsoft.ContainerService	Registering

8. 等待註冊完成，期間可點選重新整理按鈕更新進度。全部註冊完成後繼續實驗。

任務 2 - 定義全域變數並建立資源群組

1. 選擇離你最近的區域，美國用 '**eastus**'，歐洲用 '**westeurope**'，東南亞用 '**southeastasia**'，其他請詢問講師。
2. 設定你的姓名縮寫。

```
$YOUR_INITIALS="abc"
```

3. 設定全域變數

```
$INITIALS=$(($YOUR_INITIALS)).ToLower()  
$RESOURCE_GROUP="azure-$(($INITIALS))-rg"  
$LOCATION="southeastasia"
```

4. 建立資源群組。

```
az group create --location $LOCATION --resource-group $RESOURCE_GROUP
```

5. 取得該區域可用的 2 核 VM 規格。

```
az vm list-sizes --location $LOCATION `--query "[?numberOfCores == ``2``].{Name:name}" -o table
```

6. 設定 VM SKU。

```
$VM_SKU="Standard_D2as_v5"
```

任務 3 - 建立託管識別

1. 定義識別名稱：

- **AKS Identity** - 用於 AKS 控制平面元件管理叢集資源，包括入口負載平衡器和 AKS 管理的公開 IP、叢集自動縮放、Azure 磁碟、檔案、Blob CSI 驅動程式
- **Kubelet Identity** - 用於與 Azure 容器登錄表 (ACR) 驗證。需要指派 **AcrPull** 角色。

```
$AKS_IDENTITY="aks-identity-$(INITIALS)"  
$KUBELET_IDENTITY="kubelet-identity-$(INITIALS)"
```

2. 建立使用者管理的識別

```
$AKS_IDENTITY_ID=$(az identity create `--name $AKS_IDENTITY `--resource-group $RESOURCE_GROUP `--query id -o tsv)  
  
$KUBELET_IDENTITY_ID=$(az identity create `--name $KUBELET_IDENTITY `--resource-group $RESOURCE_GROUP `--query id -o tsv)
```

3. 驗證設定

```
Write-Host "AKS Identity ID: $AKS_IDENTITY_ID"  
Write-Host "Kubelet Identity ID: $KUBELET_IDENTITY_ID"
```

任務 4 - 建立虛擬網路與子網並與 NSG 關聯

1. 定義 VNET。

可用主機 IP 範圍： **10.200.0.1 - 10.200.255.250**

可用主機： 65530

```
$AKS_VNET_NAME="aks-$(INITIALS)-vnet"  
$AKS_VNET_ADDRESS_PREFIX="10.200.0.0/16"
```

2. 定義節點子網。

可用主機 IP 範圍： **10.200.4.1 - 10.200.7.250**

可用主機： 1019

```
$NODES_SUBNET_NAME="aks-$(INITIALS)-nodes-subnet"  
$NODES_SUBNET_PREFIX="10.200.4.0/22"
```

3. 定義 Pod 子網。用於動態 IP 分配。

可用主機 IP 範圍： **10.200.128.1 - 10.200.255.250**

可用主機： 32760

```
$PODS_SUBNET_NAME="aks-$(INITIALS)-pods-subnet"  
$PODS_SUBNET_PREFIX="10.200.128.0/17"
```

4. 建立 Azure 虛擬網路與子網。節點子網會與 VNet 一併建立，因為所有 VNet 至少需要一個子網。 然後建立第二個子網。

```
az network vnet create --resource-group $RESOURCE_GROUP `  
  --name $AKS_VNET_NAME `  
  --address-prefix $AKS_VNET_ADDRESS_PREFIX `  
  --subnet-name $NODES_SUBNET_NAME `  
  --subnet-prefix $NODES_SUBNET_PREFIX  
  
az network vnet subnet create --resource-group $RESOURCE_GROUP `  
  --vnet-name $AKS_VNET_NAME `  
  --name $PODS_SUBNET_NAME `  
  --address-prefixes $PODS_SUBNET_PREFIX
```

5. 取得 vnet 與子網 ID

```

$AKS_VNET_ID=$(az network vnet show ` 
  --resource-group $RESOURCE_GROUP ` 
  --name $AKS_VNET_NAME ` 
  --query id -o tsv)

$NODES_SUBNET_ID=$(az network vnet subnet show ` 
  --resource-group $RESOURCE_GROUP ` 
  --vnet-name $AKS_VNET_NAME ` 
  --name $NODES_SUBNET_NAME --query id -o tsv)

$PODS_SUBNET_ID=$(az network vnet subnet show ` 
  --resource-group $RESOURCE_GROUP ` 
  --vnet-name $AKS_VNET_NAME ` 
  --name $PODS_SUBNET_NAME ` 
  --query id -o tsv)

```

6. 驗證設定

```

Write-Host "VNet ID: $AKS_VNET_ID"
Write-Host "Nodes Subnet ID: $NODES_SUBNET_ID"
Write-Host "Pods Subnet ID: $PODS_SUBNET_ID"

```

7. 建立網路安全群組 (NSG)

```

$NSG_NAME="aks-$(INITIALS)-nsg"
$NSG_ID=$(az network nsg create ` 
  --resource-group $RESOURCE_GROUP ` 
  --name $NSG_NAME ` 
  --query id -o tsv)

```

8. 將 NSG 關聯到子網

```
az network vnet subnet update `  
--resource-group $RESOURCE_GROUP `  
--name $NODES_SUBNET_NAME `  
--vnet-name $AKS_VNET_NAME `  
--network-security-group $NSG_NAME
```

```
az network vnet subnet update `  
--resource-group $RESOURCE_GROUP `  
--name $PODS_SUBNET_NAME `  
--vnet-name $AKS_VNET_NAME `  
--network-security-group $NSG_NAME
```

9. 驗證 NSG 關聯

```
az network vnet subnet show `  
--resource-group $RESOURCE_GROUP `  
--vnet-name $AKS_VNET_NAME `  
--name $NODES_SUBNET_NAME `  
--query networkSecurityGroup.id -o tsv
```

```
az network vnet subnet show `  
--resource-group $RESOURCE_GROUP `  
--vnet-name $AKS_VNET_NAME `  
--name $PODS_SUBNET_NAME `  
--query networkSecurityGroup.id -o tsv
```

這兩個查詢應返回 NSG 的資源 ID。

任務 5 - 建立 Log Analytics 工作區

1. 建立 Log Analytics 工作區。

```
$LOG_ANALYTICS_WORKSPACE_NAME="aks-$(INITIALS)-law"  
$LOG_ANALYTICS_WORKSPACE_RESOURCE_ID=$(az monitor log-analytics workspace create `  
--resource-group $RESOURCE_GROUP `  
--workspace-name $LOG_ANALYTICS_WORKSPACE_NAME `  
--query id -o tsv)  
Write-Host "Log Analytics Workspace Resource ID: $LOG_ANALYTICS_WORKSPACE_RESOURCE_ID"
```

模組 2 目錄

練習：建立 ACR 和範例工作負載映像

在本練習中，您將建立一個 Azure 容器登錄表（ACR），然後使用它構建一個範例工作負載映像。

任務 1 - 建立 Azure 容器登錄表

1. 定義 ACR 名稱。

注意：ACR 名稱必須是全球唯一的。

```
$SUFFIX=(Get-Date -Format "MMddyy")
$ACR_NAME="acr $($INITIALS)$($SUFFIX)"
Write-Host "ACR Name: $ACR_NAME"
```

請記住此名稱！在更新清單中的映像名稱時，您會需要它。

2. 建立 Azure 容器登錄表（ACR）

```
$ACR_ID=$(az acr create --resource-group $RESOURCE_GROUP ` 
  --name $ACR_NAME ` 
  --sku Standard ` 
  --workspace $LOG_ANALYTICS_WORKSPACE_RESOURCE_ID ` 
  --query id -o tsv)
```

3. 驗證設定

```
Write-Host "ACR ID: $ACR_ID"
```

任務 2 - 使用 ACR 構建映像

1. 登入 ACR 以便不使用 Docker（使用 `--expose-token` 選項）

```
az acr login --name $ACR_NAME --expose-token
```

2. 定義映像名稱與來源資料夾位置（注意替換成你自己的路徑）

```
$SAMPLE_IMAGE_NAME="mt3-chained-web:latest"
$SOURCE_FOLDER="D:/docs/aks/WorkshopPLUS - Kubernetes Technical Briefing with Labs/Labs/MathTri
```

3. 構建範例映像。由於映像會在 ACR 中建立，因此推送會自動完成。

```
az acr build --registry $ACR_NAME ` 
  --resource-group $RESOURCE_GROUP ` 
  --file $SOURCE_FOLDER/Dockerfile ` 
  --image $SAMPLE_IMAGE_NAME ` 
$SOURCE_FOLDER
```

4. 驗證映像是否在註冊表中

```
az acr repository list --name $ACR_NAME -o table
```

Result

mt3-chained-web

5. 取得完整的映像全名

```
Write-Host "Workload Image name: $($ACR_NAME).azurecr.io/$($SAMPLE_IMAGE_NAME)"
```

6. 編輯 **workload.yaml** 檔案（第 18 行），用您構建並推送的映像完整名稱替換目前的映像名稱。

[模組 2 目錄](#)

[模組列表](#)

練習：使用 Azure CLI 建立 AKS 叢集

在本練習中，您將使用 Azure CLI 建立 AKS 叢集。

任務 1 - 使用 system 節點池建立 AKS 叢集

1. 使用所有先前的設定與資源來建立 AKS 叢集。請查看下面指令的參數。

```
$AKS_NAME="aks-$(INITIALS)"  
Write-Host "AKS Cluster Name: $AKS_NAME"
```

注意： 請參閱 Microsoft 參考：https://docs.microsoft.com/en-us/cli/azure/aks?view=azure-cli-latest#az_aks_create

| 此步驟大約需要 5-10 分鐘完成。

```
az aks create --resource-group $RESOURCE_GROUP `  
  --name $AKS_NAME `  
  --generate-ssh-keys `  
  --enable-managed-identity `  
  --assign-identity $AKS_IDENTITY_ID `  
  --assign-kubelet-identity $KUBELET_IDENTITY_ID `  
  --attach-acr $ACR_ID `  
  --node-count 1 `  
  --enable-cluster-autoscaler `  
  --min-count 1 `  
  --max-count 3 `  
  --network-plugin azure `  
  --vnet-subnet-id $NODES_SUBNET_ID `  
  --pod-subnet-id $PODS_SUBNET_ID `  
  --node-vm-size $VM_SKU `  
  --nodepool-name system1 `  
  --enable-addons monitoring `  
  --workspace-resource-id $LOG_ANALYTICS_WORKSPACE_RESOURCE_ID `  
  --zones 1 2 `  
  --enable-ahub
```

2. 叢集準備就緒後（至少 5 分鐘後），將本機連線到叢集。

```
az aks get-credentials --name $AKS_NAME --resource-group $RESOURCE_GROUP
```

3. 列出叢集中的節點。起初應只有 1 個節點。

```
kubectl get nodes -o wide
```

| 請注意，節點的 IP 位址來自 **10.200.4.xx** 子網

NAME	STATUS	ROLES	AGE	VERSION
aks-system1-42326736-vmss000000	Ready	agent	108s	v1.20.9

4. 取得叢集中所有 Pod 的清單及其 IP

```
kubectl get pods -A -o wide
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP
kube-system	ama-logs-72zbf	2/2	Running	0	14m	10.200.128.12
kube-system	ama-logs-rs-6688bdfbb4-tj8wv	1/1	Running	0	15m	10.200.128.6
kube-system	azure-cns-gm4x6	1/1	Running	0	14m	10.200.4.4
kube-system	cloud-node-manager-kpq8h	1/1	Running	0	14m	10.200.4.4
kube-system	coredns-autoscaler-69b7556b86-77pz9	1/1	Running	0	15m	10.200.128.5
kube-system	coredns-fb6b9d95f-krtbb	1/1	Running	0	14m	10.200.128.14
kube-system	coredns-fb6b9d95f-t6lqg	1/1	Running	0	15m	10.200.128.20
kube-system	csi-azuredisk-node-kpwpr	3/3	Running	0	14m	10.200.4.4
kube-system	csi-azurefile-node-fpqf4	3/3	Running	0	14m	10.200.4.4

請注意，大部分的 pod 使用來自 **10.200.128.xx** 子網的 IP 位址，而其他不需要自身 IP 位址（因為它們永遠不會被存取）的 pod 則使用其節點的 IP 位址。透過將 **PodSpec** 的 **hostNetwork** 屬性設為 true 來達成。

5. 只取得 system 節點的清單。此時，清單應返回 1 個節點。

```
kubectl get nodes -l="kubernetes.azure.com/mode=system"
```

6. 列出叢集中的節點池。應回傳 1 個節點池。

```
az aks nodepool list --cluster-name $AKS_NAME --resource-group $RESOURCE_GROUP -o table
```

Name	OsType	KubernetesVersion	VmSize	Count	MaxPods	ProvisioningState	Mode
system1	Linux	1.20.9	Standard_DS2_v2	1	30	Succeeded	System

任務 5 - 建立 Linux 使用者節點池

1. 建立 Linux 使用者節點池。

```
az aks nodepool add --resource-group $RESOURCE_GROUP `  
  --cluster-name $AKS_NAME `  
  --os-type Linux `  
  --name linux1 `  
  --node-count 1 `  
  --enable-cluster-autoscaler `  
  --min-count 1 `  
  --max-count 3 `  
  --mode User `  
  --pod-subnet-id $PODS_SUBNET_ID `  
  --node-vm-size $VM_SKU
```

2. 列出叢集中的節點。現在應該有 2 個節點。

```
kubectl get nodes
```

3. 取得 system 節點的清單。system 節點仍然只有 1 個。

```
kubectl get nodes -l="kubernetes.azure.com	mode=system"
```

4. 列出叢集中的節點池。現在應該有 2 個。

```
az aks nodepool list --cluster-name $AKS_NAME --resource-group $RESOURCE_GROUP -o table
```

Name	OsType	KubernetesVersion	VmSize	Count	MaxPods	ProvisioningState	Mode
linux1	Linux	1.20.9	Standard_DS2_v2	1	30	Succeeded	User
system1	Linux	1.20.9	Standard_DS2_v2	1	30	Succeeded	System

任務 6 - 建立 Windows 使用者節點池

1. 建立 Windows 使用者節點池。

```
az aks nodepool add --resource-group $RESOURCE_GROUP \
--cluster-name $AKS_NAME \
--os-type Windows \
--name win1 \
--node-count 1 \
--mode User \
--pod-subnet-id $PODS_SUBNET_ID \
--node-vm-size $VM_SKU
```

2. 列出叢集中的節點。現在應該有 3 個節點。

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
aks-linux1-42326736-vmss000000	Ready	agent	10m	v1.20.9
aks-system1-42326736-vmss000000	Ready	agent	16m	v1.20.9
akswin1000000	Ready	agent	2m30s	v1.20.9

3. 取得 system 節點的清單。system 節點仍然只有 1 個。

```
kubectl get nodes -l="kubernetes.azure.com/mode=system"
```

4. 取得 Linux 節點的清單。應該有 2 個節點。

```
kubectl get nodes -l="kubernetes.io/os=linux"
```

5. 列出叢集中的節點池。現在應該有 3 個。

```
az aks nodepool list --cluster-name $AKS_NAME --resource-group $RESOURCE_GROUP -o table
```

Name	OsType	KubernetesVersion	VmSize	Count	MaxPods	ProvisioningState	Mode
linux1	Linux	1.20.9	Standard_DS2_v2	1	30	Succeeded	User
system1	Linux	1.20.9	Standard_DS2_v2	1	30	Succeeded	System
win1	Windows	1.20.9	Standard_DS2_v2	1	30	Succeeded	User

任務 7 - 調整實驗的自動縮放器

AKS 自動縮放器的預設設定是針對生產環境調整，但在展示與實作實驗中，縮放上下的回應時間較長。

1. 設定叢集範圍的自動縮放設定檔，以便所有節點池的自動縮放器能更快回應。

```
az aks update --resource-group $RESOURCE_GROUP `  
--name $AKS_NAME `  
--cluster-autoscaler-profile `  
scale-down-delay-after-add=1m `  
scale-down-unready-time=1m `  
scale-down-unneeded-time=1m `  
skip-nodes-with-system-pods=true
```

注意： 有關可用選項的完整說明，請參閱此連結：[AKS 叢集自動縮放器](#)

模組 2 目錄

模組列表

練習：將工作負載部署到節點

在本練習中，您將把不同的 Pod 部署到叢集中的各種節點池中。

任務 1 - 部署一個簡單的工作負載，不指定節點

1. 將目前資料夾變更為 **Module2**

```
cd C:\k8s\labs\Module2
```

2. 部署一個具有 6 個複本且沒有 Node Selector 的工作負載。

```
kubectl apply -f workload.yaml
```

3. 取得所有 Pod 的完整清單並檢視結果。

```
kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
ATES						
workload-5f795755d8-8v8lb	1/1	Running	0	37s	10.240.0.28	aks-system1-42326736-vmss000000
workload-5f795755d8-gs7r8	0/1	ImagePullBackOff	0	37s	10.240.0.84	akswin1000000
workload-5f795755d8-ph4t6	0/1	ErrImagePull	0	37s	10.240.0.92	akswin1000000
workload-5f795755d8-q6zxw	1/1	Running	0	37s	10.240.0.52	aks-linux1-42326736-vmss000000
workload-5f795755d8-r4szw	1/1	Running	0	37s	10.240.0.43	aks-linux1-42326736-vmss000000
workload-5f795755d8-srn79	0/1	ImagePullBackOff	0	37s	10.240.0.71	akswin1000000

注意： 請注意，一些 Pod 在 Linux 節點上建立，而其他 Pod 在 Windows 節點上被排程，導致無法執行。指定的 Linux 映像與執行 Windows Server 2019 的 Windows 節點不相容。

4. 描述任何失敗的 Pod 以查看實際錯誤。

```
kubectl describe pod <failed pod name>
```

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	7m3s	default-scheduler	Successfully assigned default/workload-5f795755d8-bb4ct to akswin000001
Normal	SandboxChanged	6m59s	kubelet	Pod sandbox changed, it will be killed and re-created.
Warning	Failed	6m15s (x3 over 7m)	kubelet	Failed to pull image "nginx:1.18": rpc error: code = Unknown desc = no matching manifest for windows/amd64 10.0.17763 in the manifest list entries
Normal	Pulling	5m23s (x4 over 7m)	kubelet	Pulling image "nginx:1.18"
Warning	Failed	114s (x22 over 6m55s)	kubelet	Error: ImagePullBackOff

注意： 如果沒有任何指引，Kubernetes 排程器會嘗試平均分配工作負載到所有具有可用資源的節點。它不會檢查 Deployment 的內容以確認其映像是否與所選節點相容。

5. 更新 **workload.yaml** 檔案，新增以下 nodeSelector：

```
nodeSelector:  
  kubernetes.io/os: linux
```

```
spec:  
  nodeSelector:  
    kubernetes.io/os: linux  
  containers:  
    - name: nginx  
      image: nginx:1.18
```

6. 再次套用部署並列出 Pod。

```
kubectl apply -f workload.yaml  
kubectl get pods -o wide
```

注意： 在刪除不良 Pod 之前，它們可能會顯示為 **Terminated** 狀態，因此需等待幾秒鐘。

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
workload-86dd8dfdd6-4tnzg	1/1	Running	0	13s	10.240.0.65	aks-linux1-42326736-vmss000000
workload-86dd8dfdd6-87k4c	1/1	Running	0	11s	10.240.0.7	aks-system1-42326736-vmss000000
workload-86dd8dfdd6-s9wxm	1/1	Running	0	11s	10.240.0.45	aks-linux1-42326736-vmss000000
workload-86dd8dfdd6-vb7z9	1/1	Running	0	11s	10.240.0.46	aks-linux1-42326736-vmss000000
workload-86dd8dfdd6-w7dr6	1/1	Running	0	13s	10.240.0.11	aks-system1-42326736-vmss000000
workload-86dd8dfdd6-zm8sb	1/1	Running	0	13s	10.240.0.64	aks-linux1-42326736-vmss000000

請注意，現在所有 Pod 都在執行，但它們分散在 system 節點與 user 節點上。建立使用者節點的原因是將工作負載與系統工具分離。

7. 更新 **workload.yaml** 檔案，在 nodeSelector 中新增額外標籤 "kubernetes.azure.com/mode: user"。最終的 nodeSelector 段落應如下所示：

```
nodeSelector:  
  kubernetes.io/os: linux  
  kubernetes.azure.com/mode: user
```

8. 套用部署並列出 Pod。

```
kubectl apply -f workload.yaml  
kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
workload-bbc8765b7-5gnts	1/1	Running	0	23s	10.240.0.40	aks-linux1-42326736-vmss000000
workload-bbc8765b7-5mzbz	1/1	Running	0	23s	10.240.0.55	aks-linux1-42326736-vmss000000
workload-bbc8765b7-n8j9r	1/1	Running	0	20s	10.240.0.36	aks-linux1-42326736-vmss000000
workload-bbc8765b7-pzz59	1/1	Running	0	21s	10.240.0.43	aks-linux1-42326736-vmss000000
workload-bbc8765b7-rsrc5	1/1	Running	0	23s	10.240.0.42	aks-linux1-42326736-vmss000000
workload-bbc8765b7-xnsvb	1/1	Running	0	21s	10.240.0.49	aks-linux1-42326736-vmss000000

完美！

[模組 2 目錄](#)

[模組列表](#)

練習：擴縮節點以滿足需求

在本練習中，您將觀察 AKS 自動縮放器如何依據增加的需求調整節點數量，然後在負載減少時將其縮回。

任務 1 - 驗證目前節點的名稱與數量

1. 取得 Linux 使用者節點的清單。

```
kubectl get nodes -l="kubernetes.azure.com/mode=user,kubernetes.io/os=linux"
```

任務 2 - 增加複本數以觸發自動縮放器擴充

- 增加複本數，以強制自動縮放器在 Linux 使用者節點池中建立更多節點。

```
kubectl scale --replicas=40 deploy/workload
```

- 取得 Pod 清單。

```
kubectl get pods -o wide
```

請注意，一些 Pod 處於 **Pending** 狀態。

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
ATES						
workload-bbc8765b7-4fptg	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-4tttk	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-5852w	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-5gnts	1/1	Running	0	2m3s	10.240.0.40	aks-linux1-42326736-vmss000000
workload-bbc8765b7-5mzbz	1/1	Running	0	2m3s	10.240.0.55	aks-linux1-42326736-vmss000000
workload-bbc8765b7-6d4cv	1/1	Running	0	4s	10.240.0.37	aks-linux1-42326736-vmss000000
workload-bbc8765b7-6pdxj	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-6qt9p	1/1	Running	0	4s	10.240.0.58	aks-linux1-42326736-vmss000000
workload-bbc8765b7-75bsh	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-9pjxj	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-b2qkr	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-b6r2f	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-c7ph6	1/1	Running	0	4s	10.240.0.41	aks-linux1-42326736-vmss000000
workload-bbc8765b7-dk748	0/1	Pending	0	4s	<none>	<none>
workload-bbc8765b7-fv9hd	1/1	Running	0	4s	10.240.0.46	aks-linux1-42326736-vmss000000
workload-bbc8765b7-hrl4h	0/1	Pending	0	4s	<none>	<none>

- 描述一個待處理的 Pod

```
kubectl describe pod <pending pod name>
```

請注意，事件部分描述了問題與解決方案。

Events:				
Type	Reason	Age	From	Message
Warning	FailedScheduling	74s	default-scheduler	0/3 nodes are available: 1 Insufficient cpu, 2 node(s) didn't match Pod's node affinity.
Warning	FailedScheduling	74s	default-scheduler	0/3 nodes are available: 1 Insufficient cpu, 2 node(s) didn't match Pod's node affinity.
Normal	TriggeredScaleUp	66s	cluster-autoscaler	pod triggered scale-up: [{aks-linux1-42326736-vmss 1->3 (max: 3)}]

- 開始監視節點

```
kubectl get nodes -l="kubernetes.azure.com/mode=user,kubernetes.io/os=linux" -w
```

幾分鐘後，您會看到節點數量增加。

NAME	STATUS	ROLES	AGE	VERSION
aks-linux1-42326736-vmss000000	Ready	agent	35m	v1.20.9
aks-linux1-42326736-vmss000001	Ready	agent	2m	v1.20.9
aks-linux1-42326736-vmss000002	Ready	agent	119s	v1.20.9

5. 當新節點處於 **Ready** 狀態時，按 Ctrl-C 中斷監視並返回主控台。

6. 取得 Pod 清單。

```
kubectl get pods -o wide
```

請注意，所有待處理的 Pod 都已執行，且它們都位於同一節點池中，排程在新節點上。

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
workload-bbc8765b7-4fptg	1/1	Running	0	5m50s	10.240.0.106	aks-linux1-42326736-vmss000001
workload-bbc8765b7-4ttnk	1/1	Running	0	5m50s	10.240.0.137	aks-linux1-42326736-vmss000002
workload-bbc8765b7-5852w	1/1	Running	0	5m50s	10.240.0.110	aks-linux1-42326736-vmss000001
workload-bbc8765b7-5gnts	1/1	Running	0	7m49s	10.240.0.40	aks-linux1-42326736-vmss000000
workload-bbc8765b7-5mzbz	1/1	Running	0	7m49s	10.240.0.55	aks-linux1-42326736-vmss000000
workload-bbc8765b7-6d4cv	1/1	Running	0	5m50s	10.240.0.37	aks-linux1-42326736-vmss000000
workload-bbc8765b7-6pdxj	1/1	Running	0	5m50s	10.240.0.155	aks-linux1-42326736-vmss000002
workload-bbc8765b7-6qt9p	1/1	Running	0	5m50s	10.240.0.58	aks-linux1-42326736-vmss000000
workload-bbc8765b7-75bsh	1/1	Running	0	5m50s	10.240.0.126	aks-linux1-42326736-vmss000001

任務 3 - 減少工作負載以觸發自動縮放器縮減

1. 刪除工作負載。

```
kubectl delete deploy/workload
```

2. 監視節點

```
kubectl get nodes -l="kubernetes.azure.com/mode=user,kubernetes.io/os=linux" -w
```

注意： 幾分鐘後，兩個節點會進入 **NotReady** 狀態，然後消失。並非所有新建立的節點都會成為「被刪除」對象。縮減節點會根據內部縮減策略選擇節點。

NAME	STATUS	ROLES	AGE	VERSION
aks-linux1-42326736-vmss000000	NotReady	agent	42m	v1.20.9
aks-linux1-42326736-vmss000001	NotReady	agent	8m22s	v1.20.9
aks-linux1-42326736-vmss000002	Ready	agent	8m21s	v1.20.9

注意： 若 Pod 無法被刪除，例如下列情況，叢集自動縮放器可能無法縮減節點：

- Pod 是直接建立的，而不是由控制器物件（例如 Deployment 或 ReplicaSet）管理。
- Pod 中斷預算（PDB）限制過於嚴格，不允許 Pod 數量低於某個閾值。
- Pod 使用節點選擇器或反親和性，導致其他節點在排程時無法滿足這些要求。

3. 手動取得節點清單，重複以下命令，直到只剩 1 個節點。

```
kubectl get nodes -l="kubernetes.azure.com/mode=user,kubernetes.io/os=linux"
```

NAME	STATUS	ROLES	AGE	VERSION
aks-linux1-42326736-vmss000002	Ready	Agent	10m	v1.20.9

[模組 2 目錄](#)

[模組列表](#)

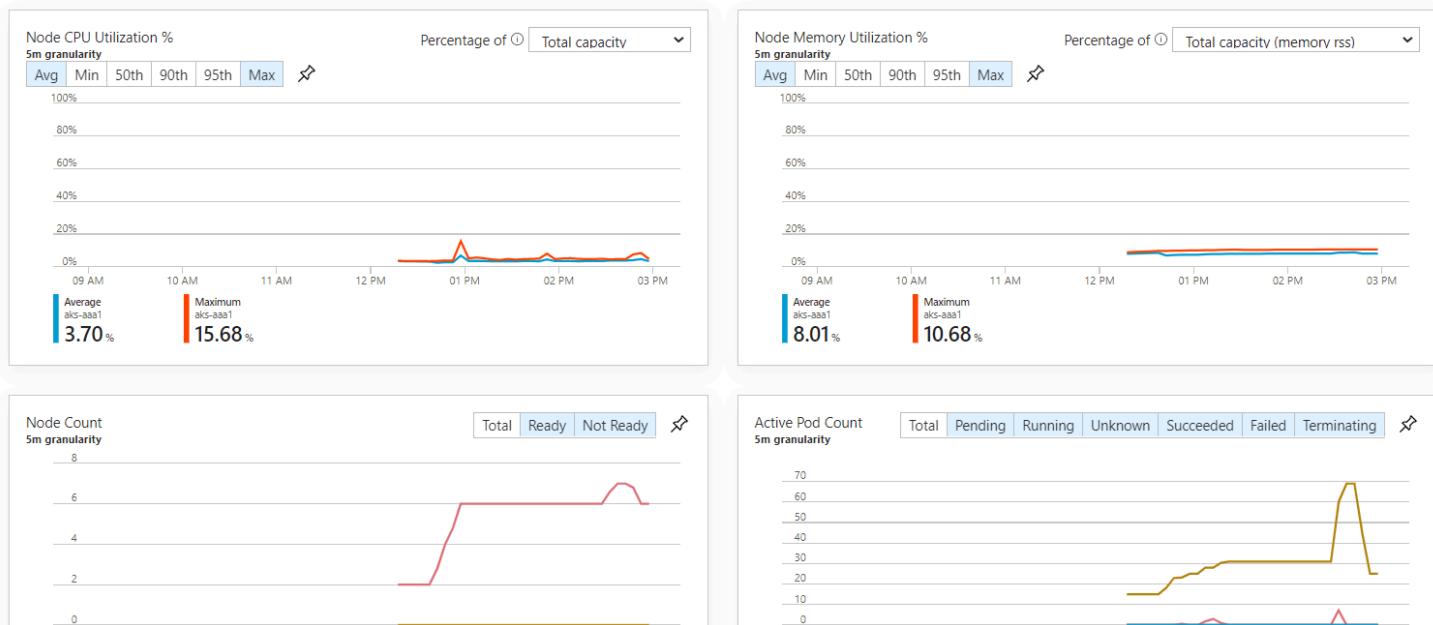
練習：檢視容器洞察

現在您已使用叢集一段時間，容器洞察中應有一些指標與日誌可供檢視。

任務 1 - 檢視容器洞察

1. 在瀏覽器中打開 Azure 入口網站。
2. 搜尋 "Kubernetes"。點選您的叢集。
3. 向下捲動到面板上的 **Monitoring -> Insights**。
4. 在各個分頁之間瀏覽。

叢集指標總覽：



節點清單：

Search by name...		Metric: CPU Usage (millicores) (computed from Capacity)	Min	Avg	50th	90th	95th	Max	8 items
Name	Status	95th % ↓	95th	Containers	UpTime	Controller	Trend 95th % (1 bar = 15m)		
▶ akswin1000000	Ok	7%	143 mc	2	3 hours	-			
▶ akswin1000001	Ok	7%	131 mc	3	3 hours	-			
▶ aks-linuxagent1-36262365-vmss000000	Ok	4%	86 mc	25	2 hours	-			
▶ aks-systempool-36262365-vmss000001	Ok	4%	82 mc	9	3 hours	-			
▶ aks-linuxagent1-36262365-vmss000001	Ok	4%	74 mc	25	2 hours	-			
▶ aks-linuxagent1-36262365-vmss000002	Ok	4%	72 mc	12	29 mins	-			
▶ aks-systempool-36262365-vmss000000	Ok	3%	69 mc	11	3 hours	-			

容器：

Search by name...		Metric: CPU Usage (millicores)	Min	Avg	50th	90th	95th	Max	87 items
Name	Status	95th % ↓	95th	Pod	Node	Restarts	UpTime	Trend 95th % (1 bar = 15m)	
omsagent-win	Ok	8%	17 mc	omsagent-win-8dtvw	akswin1000000	0	2 hours		
omsagent-win	Ok	7%	14 mc	omsagent-win-mb5zk	akswin1000001	0	2 hours		

5. 向下捲動到面板上的 Logs。

6. 將此查詢複製到編輯器中，然後點選 Run

```

KubePodInventory
| where TimeGenerated > ago(1h)
| join kind=inner (
    ContainerLogV2
) on $left.ContainerID == $right.ContainerID
| project TimeGenerated, Computer, PodName=Name, ContainerName=split(ContainerName, "/",1)[0], LogEntry

```

The screenshot shows the Azure Log Analytics workspace interface. At the top, there is a toolbar with buttons for 'Run', 'Save', 'Share', 'New alert rule', 'Export', 'Pin to dashboard', and 'Format query'. Below the toolbar, the query results are displayed in a table format.

Completed

Time range: Set in query | Display time (UTC+00:00) | Group columns

TimeGenerated [UTC]	Computer	PodName	ContainerName	LogEntry
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000001	workload-bbc8765b7-sdwz8	nginx	10-listen-on-ipv6-by-default.sh: Ge
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000001	workload-bbc8765b7-sdwz8	nginx	10-listen-on-ipv6-by-default.sh: Er
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000001	workload-bbc8765b7-sdwz8	nginx	/docker-entrypoint.sh: Launching ,
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000001	workload-bbc8765b7-sdwz8	nginx	/docker-entrypoint.sh: Configurati
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	/docker-entrypoint.sh: /docke
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	/docker-entrypoint.sh: Looking fo
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	/docker-entrypoint.sh: Launching ,
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	10-listen-on-ipv6-by-default.sh: Ge
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	10-listen-on-ipv6-by-default.sh: Er
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	/docker-entrypoint.sh: Launching ,
8/4/2021, 8:14:10.000 PM	aks-linuxagent1-36262365-vmss000000	workload-bbc8765b7-4b2qj	nginx	/docker-entrypoint.sh: Configurati

您將看到容器日誌如何被傳送到 Log Analytics 工作區進行分析。

容器洞察的範例 Kusto 查詢

以下連結列出了一些常見的容器洞察資料 Kusto 查詢：

<https://docs.microsoft.com/en-us/azure/azure-monitor/containers/container-insights-log-query>

[模組 2 目錄](#)

[模組列表](#)

練習：清理資源

任務 - 刪除叢集及其資源 - Azure CLI

完成叢集操作後，可以刪除叢集。您可以在此處找到完整的說明，以便重新建立它。

建議您多次重複此實驗，變更某些設定，以熟悉 AKS 叢集的操作。

1. 刪除叢集比建立叢集簡單得多。

```
az aks delete --resource-group $RESOURCE_GROUP --name $AKS_NAME
```

2. 如果您打算稍後重新建立叢集，則可以跳過刪除這些資源。
3. 刪除 Log Analytics 工作區。

```
az monitor log-analytics workspace delete --resource-group $RESOURCE_GROUP --workspace-name $LOG_WORKSPACE_NAME
```

4. 刪除虛擬網路

```
az network vnet delete --resource-group $RESOURCE_GROUP --name $AKS_VNET
```

提示：您可以逐一刪除上述資源，也可以刪除整個資源群組，這將刪除其中的所有內容。只有在您打算重新建立所有支援資源時才執行此操作。

5. 刪除整個資源群組

```
az group delete --resource-group $RESOURCE_GROUP
```

[模組列表](#)