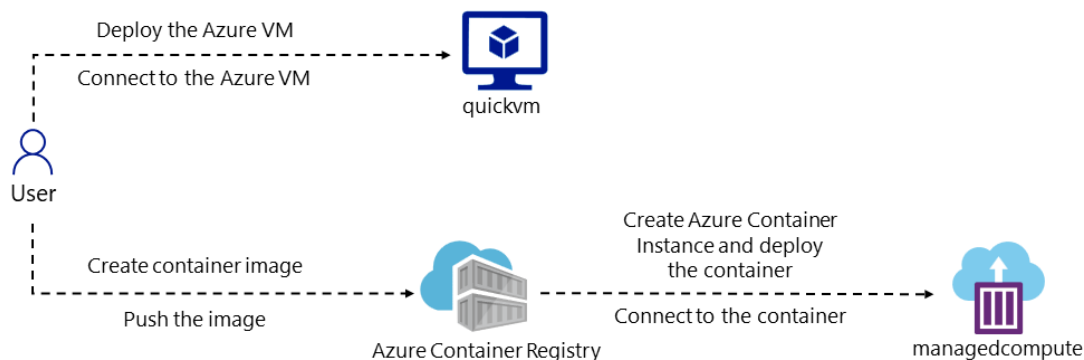


鑑於 Microsoft 雲端工具的動態特性，說明和步驟可能無法正確對應。  
如果有無法操作下去的情況，請留言讓我知道，會盡快更新。

## 架構圖



描述使用映像和容器部署計算工作負載的體系結構關係圖

## 練習 1：使用 Azure 命令列介面 (CLI) 建立 VM

### 任務 1：開啟 Azure 門戶

1. 在工作列上，選擇 Microsoft Edge 圖示。
2. 在開啟的瀏覽器視窗中，導航到 Azure 門戶 (<https://portal.azure.com>)，然後使用你將用於此實驗的帳戶登入。

**注意：**第一次登入 Azure 門戶時，你會看到一個門戶教程。如果想跳過該導覽，請選擇“開始使用”以開始使用門戶。

### 任務 2：建立資源組

1. 在 Azure 門戶的導航窗格中，使用“搜尋資源、服務和文件”文字框搜尋“資源組”，然後在結果列表中，選擇“資源組”。
2. 在“資源組”邊欄選項卡中，選擇“建立”。
3. 在“建立資源組”邊欄選項卡的“基本資訊”選項卡上，執行以下操作，然後選擇“檢視 + 建立”：

設定	操作
“訂閱”下拉列表	保留預設值
“資源組”文字框	輸入“ContainerCompute”
“區域”下拉列表	選擇“(US)美國東部”

以下螢幕截圖顯示了“建立資源組”邊欄選項卡上配置的設定。


The screenshot shows the 'Create a resource group' page in the Azure portal. The breadcrumb navigation at the top reads: 'All services > Create a resource > Marketplace > Resource group >'. The main heading is 'Create a resource group ...'. Below this, there are three tabs: 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab is currently selected. A descriptive paragraph explains that a resource group is a container for related resources. Under 'Project details', the 'Subscription' dropdown is set to 'Azure Pass - Sponsorship' and the 'Resource group' text box contains 'ContainerCompute' with a green checkmark. Under 'Resource details', the 'Region' dropdown is set to '(US) East US'. At the bottom, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Tags >'.

“建立資源組”邊欄選項卡

4. 在“檢視 + 建立”選項卡中，檢視在上述步驟中選擇的選項。
5. 選擇“建立”以使用指定的配置建立資源組。

**注意：** 等待建立任務完成後再繼續本實驗。

### 任務 3：開啟 Azure Cloud Shell

1. 在 Azure 門戶中，選擇“Cloud Shell”圖示  以開啟一個新的 PowerShell 會話。如果 Cloud Shell 預設設定為 PowerShell 會話，請選擇“PowerShell”，然後在下拉選單中選擇“Bash”。

**注意：** “Cloud Shell”圖示使用大於符號 (>) 和下劃線字元 ( ) 表示。

**注意：**如果這是你第一次啟動 Cloud Shell，當系統提示你選擇“Bash”或“PowerShell”時，請選擇“Bash”。當出現訊息“未安裝儲存”時，請選擇你在本實驗中使用的訂閱，然後選擇“建立儲存”。

2. 在門戶的“Cloud Shell”命令提示符處，執行以下命令，以檢視 Azure CLI 工具的版本：

```
1 az --version
```

## 任務 4：使用 Azure CLI 命令

1. 執行以下命令以獲取 CLI 根級別的子組和命令列表：

```
1 az --help
```

2. 執行以下命令以獲取 Azure 虛擬機器子組和命令列表：

```
1 az vm --help
```

3. 執行以下命令，以檢視“建立虛擬機器”命令的引數和示例列表：

```
1 az vm create --help
```

4. 執行以下命令，以使用下列設定建立新的虛擬機器，請務必記錄系統要求在下方建立的密碼，稍後在實驗室中將需要使用此密碼來訪問虛擬機器：

- 資源組：ContainerCompute
- 名稱：quickvm
- 映像:Debian
- 管理員使用者名稱：student
- 管理員密碼：<CreateYourPassword>

**注意：**將以下命令中的 <CreateYourPassword> 替換為自己的密碼。

```
1 az vm create --resource-group ContainerCompute --name quickvm --image Debian --admin-username student --admin-
```

```
password <CreateYourPassword>
```

**注意：**等待 VM 建立完畢。該過程完成後，命令將返回一個 JavaScript 物件表示法 (JSON) 檔案，其中包含有關計算機的詳細資訊。

5. 執行以下命令，以獲取包含有關新建立 VM 的各種元資料的更詳細 JSON 檔案：

```
1 az vm show --resource-group ContainerCompute --name quickvm
```

6. 執行以下命令，以列出與 VM 關聯的所有 IP 地址：

```
1 az vm list-ip-addresses --resource-group ContainerCompute --name quickvm
```

7. 執行以下命令以篩選輸出，以便只返回第一個 IP 地址值：

```
1 az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].{ip:virtualMachine.network.publicIpAddresses[0].ipAddress}' --output tsv
```

8. 執行以下命令，以將上一個命令的結果儲存在名為 ipAddress 的新 Bash Shell 變數中：

```
1 ipAddress=$(az vm list-ip-addresses --resource-group ContainerCompute --name quickvm --query '[].{ip:virtualMachine.network.publicIpAddresses[0].ipAddress}' --output tsv)
```

9. 執行以下指令碼以呈現 Bash Shell 變數 ipAddress 的值：

```
1 echo $ipAddress
```

10. 執行以下命令，透過使用安全外殼 (SSH) 工具和儲存在 Bash Shell 變數 ipAddress 中的 IP 地址連線到之前在本實驗室中建立的 VM：

```
1 ssh student@$ipAddress
```

11. SSH 工具通知你無法驗證主機的真實性，然後詢問是否要繼續連線。輸入“yes”然後按 Enter，以繼續連線到 VM。
12. 然後，SSH 工具要求輸入密碼。輸入之前建立的密碼，然後按 Enter 鍵，使用 VM 進行身份驗證。
13. 使用 SSH 連線到 VM 後，執行以下命令，以獲取描述 Linux VM 的元資料：

```
1 uname -a
```
14. 使用**出口**命令結束 SSH 會話：

```
1 exit
```
15. 關閉門戶中的“Cloud Shell”窗格。

## 審閱

在本練習中，你使用了 Cloud Shell 作為自動指令碼的一部分建立 VM。

## 練習 2：建立 Docker 容器映像並將其部署到 Azure 容器登錄檔

### 任務 1：開啟 Cloud Shell 和編輯器

1. 在 Azure 門戶的導航窗格中，選擇“Cloud Shell”圖示，開啟一個新的 shell 例項。

**注意：**等待 Cloud Shell 完成連線到例項，再繼續進行本實驗室。

2. 在門戶中的“Cloud Shell”命令提示符中，執行以下命令，以從根目錄移動到 ~/clouddrive 目錄：

```
1 cd ~/clouddrive
```

3. 執行以下命令，以在 ~/clouddrive 目錄中建立名為“ipcheck”的新目錄：

```
1 mkdir ipcheck
```

4. 執行以下命令，以將活動目錄從 `~/clouddrive` 更改為 `~/clouddrive/ipcheck`：

```
1 cd ~/clouddrive/ipcheck
```

5. 執行以下命令，以在當前目錄中建立一個新的 .NET Core 控制檯應用程式：

```
1 dotnet new console --output . --name ipcheck --framework net6.0
```

6. 執行以下命令，以在名為 Dockerfile 的 `~/clouddrive/ipcheck` 目錄中建立一個新檔案：

```
1 touch Dockerfile
```

7. 執行以下命令，以在當前目錄的上下文中開啟嵌入式圖形編輯器：

```
1 code .
```

## 任務 2：建立並測試 .NET 應用程式

1. 在圖形編輯器中，在“檔案”窗格上，選擇“Program.cs”檔案，以在編輯器中開啟該檔案。
2. 刪除“Program.cs”檔案的全部內容。
3. 將以下程式碼複製並貼上到“Program.cs”檔案中：

```
1 public class Program
2 {
3     public static void Main(string[] args)
4     {
5         // Check if network is available
6         if (System.Net.NetworkInformation.NetworkInterface
7             .GetIsNetworkAvailable())
8         {
9             System.Console.WriteLine("Current IP Address:");
```

```

10         // Get host entry for current hostname
11         string hostname = System.Net.Dns.GetHostName();
12         System.Net.IPEndPoint host = System.Net.Dns.GetHostEntry(hostname);
13
14         // Iterate over each IP address and render
15         // their values
16         foreach(System.Net.IPAddress address in host.AddressList)
17         {
18             System.Console.WriteLine($"{address}");
19         }
20     else
21     {
22         System.Console.WriteLine("No Network Connection");
23     }
24 }
25 }

```

4. 使用圖形編輯器中的選單或 Ctrl+S 鍵盤快捷鍵儲存 **Program.cs** 檔案。不要關閉圖形編輯器。
5. 返回到命令提示符處，執行以下命令以執行應用程式：

```
1 dotnet run
```

6. 檢視執行的結果。應該為 Cloud Shell 例項列出至少一個 IP 地址。
7. 在圖形編輯器中，在編輯器的“檔案”窗格上，選擇“Dockerfile”檔案，以在編輯器中開啟該檔案。
8. 將以下程式碼複製並貼上到“Dockerfile”檔案中：

```

1 # Start using the .NET 6 SDK container image
2 FROM mcr.microsoft.com/dotnet/sdk:6.0-alpine AS build
3
4 # Change current working directory
5 WORKDIR /app

```

```
6
7 # Copy existing files from host machine
8 COPY . ./
9
10 # Publish application to the "out" folder
11 RUN dotnet publish --configuration Release --output out
12
13 # Start container by running application DLL
14 ENTRYPOINT ["dotnet", "out/ipcheck.dll"]
```

9. 使用圖形編輯器中的選單或 Ctrl+S 鍵盤快捷鍵儲存 **Dockerfile** 檔案。
10. 讓 Cloud Shell 保持開啟狀態，下一個任務中會用到。

### 任務 3：建立 Azure 容器登錄檔資源

1. 在門戶中的 Cloud Shell 命令提示符下，執行以下命令，以為容器登錄檔資源建立具有唯一值的變數：

```
1 registryName=conregistry$RANDOM
```

2. 在門戶中的 Cloud Shell 命令提示符處，執行以下命令，以驗證在上一步中建立的名稱是否可用：

```
1 az acr check-name --name $registryName
```

如果結果顯示名稱可用，請繼續執行下一步。如果名稱不可用，請在上一步中重新執行該命令，然後再次驗證可用性。

3. 在門戶中的 Cloud Shell 命令提示符處，執行以下命令，以建立容器登錄檔資源：

```
1 az acr create --resource-group ContainerCompute --name
  $registryName --sku Basic
```

**注意：**等待建立任務完成，再繼續操作本實驗室。

### 任務 4：儲存容器登錄檔元資料



1. 在門戶的 Cloud Shell 命令提示符中，執行以下命令，以檢視訂閱中的所有容器登錄檔列表：

```
1 az acr list
```

2. 執行以下命令，確保將登錄檔的名稱視為輸出。如果看不到除“[]”以外的輸出，請等待一分鐘，然後嘗試再次執行該命令。

```
1 az acr list --query "max_by([], &creationDate).name" --output tsv
```

3. 執行以下命令：

```
1 acrName=$(az acr list --query "max_by([], &creationDate).name" --output tsv)
```

4. 執行以下命令：

```
1 echo $acrName
```

## 任務 5：將 Docker 容器映像部署到容器登錄檔

1. 執行以下命令，以將活動目錄從 ~/ 更改為 ~/clouddrive/ipcheck：

```
1 cd ~/clouddrive/ipcheck
```

2. 執行以下命令以獲取當前目錄的內容：

```
1 dir
```

3. 執行以下命令，以將原始碼上傳到容器登錄檔並將容器映像構建為容器登錄檔任務：

```
1 az acr build --registry $acrName --image ipcheck:latest .
```

**注意：**等待生成任務完成後再繼續本實驗室。

4. 關閉門戶中的“Cloud Shell”窗格。

## 任務 6：在容器登錄檔中驗證容器映像

1. 在 Azure 門戶的“導航”窗格中，選擇“資源組”連結。
2. 在“資源組”邊欄選項卡中，選擇你之前在本實驗室中建立的“ContainerCompute”資源組。
3. 在“ContainerCompute”邊欄選項卡中，選擇你之前在本實驗室中建立的容器登錄檔。
4. 在“容器登錄檔”邊欄選項卡中，在“服務”部分中選擇“儲存庫”連結。
5. 在“儲存庫”部分中，選擇“ipcheck”容器映像儲存庫，然後選擇“最新”標記。
6. 審閱具有“最新”標記的容器映像版本的元資料。

**注意：**你也可以選擇“執行 ID”連結以檢視有關生成任務的元資料。

## 審閱

在本練習中，你建立了一個 .NET 控制檯應用程式來顯示計算機的當前 IP 地址。然後，你將 **Dockerfile** 檔案新增到應用程式，以便將其轉換為 Docker 容器映像。最後，你將容器映像部署到容器登錄檔。

## 練習 3：部署 Azure 容器例項

### 任務 1：在容器登錄檔中啟用管理員使用者

1. 在 Azure 門戶的“導航”窗格中，選擇“資源組”連結。
2. 在“資源組”邊欄選項卡中，選擇你之前在本實驗室中建立的“ContainerCompute”資源組。
3. 在“ContainerCompute”邊欄選項卡中，選擇你之前在本實驗室中建立的容器登錄檔，然後選擇“更新”。
4. 在“更新容器登錄檔”邊欄選項卡中，在“管理員使用者”部分中選擇“啟用”。
5. 選擇“儲存”，然後關閉“更新容器登錄檔”邊欄選項卡。

### 任務 2：自動將容器映像部署到 Azure 容器例項

1. 在“容器登錄檔”邊欄選項卡中，在“服務”部分中選擇“儲存庫”連結。
2. 在“儲存庫”部分，選擇“ipcheck”容器映像儲存庫。
3. 在“儲存庫”邊欄選項卡中，選擇與“最新”標記條目關聯的省略號選單，然後選擇“執行例項”。
4. 在“建立容器例項”邊欄選項卡中，執行以下操作，然後選擇“建立”：

設定	操作
“容器名稱”文字框	輸入“managedcompute”
“容器映像”文字框	保留預設值
“OS 型別”部分	選擇“Linux”
“訂閱”文字框	保留預設值
“資源組”下拉列表	選擇“ContainerCompute”
“位置”下拉列表	選擇“美國東部”
“核心數”下拉列表	選擇“2”
“記憶體 (GB)”文字框	輸入“4”
“公共 IP 地址”部分	請選擇“否”

以下螢幕截圖顯示了“建立容器例項”邊欄選項卡上配置的設定。



## Create container instance ...

Container name *	<input type="text" value="managedcompute"/>
Container image	<input type="text" value="conregistry1570.azurecr.io/ipcheck:latest"/>
OS type	<input checked="" type="radio"/> Linux <input type="radio"/> Windows
Subscription *	<input type="text" value="Azure Pass - Sponsorship"/>
Resource group *	<input type="text" value="ContainerCompute"/> <a href="#">Create new</a>
Location *	<input type="text" value="East US"/>
Number of cores	<input type="text" value="2"/>
Memory (GB) *	<input type="text" value="4"/>
Public IP address	<input type="radio"/> Yes <input checked="" type="radio"/> No

Create

“建立容器例項”邊欄選項卡

**注意：** 等待容器例項建立完成，再繼續操作本實驗室。

### 任務 3：手動將容器映像部署到容器例項

1. 在 Azure 門戶的“導航”窗格中，選擇“建立資源”連結。
2. 在“建立資源”邊欄選項卡的“搜尋服務和市場”文字框中，輸入“容器例項”，然後按 Enter 鍵。
3. 在“市場”搜尋結果邊欄選項卡中，選擇“容器例項”結果。
4. 在“容器例項”邊欄選項卡中，選擇“建立”。
5. 在“建立容器例項”邊欄選項卡的“基本資訊”選項卡上，執行以下操作，然後選擇“檢視 + 建立”：

設定	操作
“訂閱”下拉列表	保留預設值
“資源組”下拉列表	選擇“ContainerCompute”

設定	操作
“容器名稱”文字框	輸入“manualcompute”
“區域”下拉列表	選擇“(US)美國東部”
“映像源”部分	選擇“Azure 容器登錄檔”
“登錄檔”下拉列表	選擇你之前在本實驗室中建立的“Azure 容器登錄檔”資源
“映像”下拉列表	選擇“ipcheck”
“映像標記”下拉列表	選擇“最新”

以下螢幕截圖顯示了“建立容器例項”邊欄選項卡上配置的設定。

All services > Create a resource > Marketplace > Container Instances >

## Create container instance ...

Basics Networking Advanced Tags Review + create

Azure Container Instances (ACI) allows you to quickly and easily run containers on Azure without managing servers or having to learn new tools. ACI offers per-second billing to minimize the cost of running containers on the cloud.  
[Learn more about Azure Container Instances](#)

**Project details**

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

Subscription \* ⓘ Azure Pass - Sponsorship ▼

Resource group \* ⓘ ContainerCompute ▼  
[Create new](#)

**Container details**

Container name \* ⓘ manualcompute ✓

Region \* ⓘ (US) East US ▼

Image source \* ⓘ ☐ Quickstart images  
☒ Azure Container Registry  
☐ Docker Hub or other registry

Registry \* ⓘ RegNamecdk ▼

Image \* ⓘ ipcheck ▼

Image tag \* ⓘ latest ▼

OS type Linux

Size \* ⓘ 1 vcpu, 1.5 GiB memory, 0 gpus  
[Change size](#)

[Review + create](#) < Previous Next : Networking >

“建立容器例項”邊欄選項卡

- 在“檢視 + 建立”選項卡中，檢視選定的選項。

7. 選擇“建立”以使用指定配置建立容器例項。

**注意：**等待容器例項建立完成，再繼續操作本實驗室。

## 任務 4：驗證容器例項是否成功執行

1. 在 Azure 門戶的導航窗格中，選擇“資源組”連結。
2. 在“資源組”邊欄選項卡中，選擇你之前在本實驗室中建立的“ContainerCompute”資源組。
3. 在“ContainerCompute”邊欄選項卡中，選擇你之前在本實驗室中建立的 manualcompute 容器例項。
4. 在“容器例項”邊欄選項卡中，在“設定”部分中選擇“容器”連結。
5. 在“容器”部分中，檢視“事件”列表。
6. 選擇“日誌”選項卡，然後檢視容器例項中的簡訊日誌。

**注意：**你也可以選擇檢視 managedcompute 容器例項中的“事件”和“日誌”。

注意：此時 manualcompute 和 managedcompute 可能不包含任何事件。


**注意：**應用程式完成執行後，容器因為已完成工作而終止。對於手動建立的容器例項，你表示可以接受成功退出，因此該容器只執行一次。自動建立的例項沒有提供此選項，並且假定該容器應始終處於執行狀態，因此該容器會反覆重啟。

## 審閱

在本練習中，使用多種方法將容器映像部署到 Azure 容器例項。透過使用手動方法，你可以進一步自定義部署，並將基於任務的應用程式作為容器執行的一部分執行。

## 練習 4：清理訂閱

### 任務 1：開啟 Azure Cloud Shell 並列出資源組

1. 在 Azure 門戶中，選擇“Cloud Shell”圖示  以開啟一個新的 Bash 會話。如果 Cloud Shell 預設設定為 PowerShell 會話，請選擇“PowerShell”，然後在下拉選單中選擇“Bash”。

**注意：**如果這是你第一次啟動 Cloud Shell，當系統提示你選擇“Bash”或“PowerShell”時，請選擇“PowerShell”。當出現訊息“未安裝儲存”時，請選擇你在本實驗中使用的訂閱，然後選擇“建立儲存”。

## 任務 2：刪除資源組

1. 在“Cloud Shell”窗格中，執行以下命令以刪除“ContainerCompute”資源組：

```
1 az group delete --name ContainerCompute --no-wait --yes
```

**注意：**該命令以非同步方式執行（由 --nowait 引數確定），因此，儘管可立即在同一個 Bash 會話中執行另一個 Azure CLI 命令，但實際上要花幾分鐘才能刪除資源組。

1. 關閉門戶中的“Cloud Shell”窗格。

## 任務 3：關閉活動應用程式

- 關閉當前正在執行的 Microsoft Edge 應用程式。

## 審閱

在本練習中，你透過刪除本實驗室中使用的資源組清理了訂閱。