

使用 PowerShell 可以编写命令并立即执行它们。 这称为“交互模式”。

回想一下， 客户关系管理 (CRM) 示例中的总体目标是创建三个包含虚拟机的测试环境。 可使用资源组来确保已将 VM 组织到单独的环境中：一个用于单元测试、一个用于集成测试、一个用于验收测试。 只需要创建一次资源组，这意味着使用 PowerShell 的交互模式是一个不错的选择。

在 PowerShell 中键入命令时，PowerShell 会将该命令与随后用于执行所请求的操作的 *cmdlet* 进行匹配。让我们了解一些可用的常见命令，然后了解如何安装 PowerShell 的 Azure 支持。

## 什么是 PowerShell cmdlet？

PowerShell 命令称为“cmdlet”（读成“command-let”）。 cmdlet 是操纵单个功能的命令。 术语“cmdlet”意在表示“小命令”。 按照惯例，cmdlet 作者应使 cmdlet 保持简单且用途单一。

基本 PowerShell 产品附带可用于会话和后台作业等功能的 cmdlet。 将模块添加到 PowerShell 安装以获取操纵其他功能的 cmdlet。 例如，提供有第三方模块，可用于 ftp、管理操作系统、访问文件系统等。


Cmdlet 遵循动词-名词命名约定；例如，Get-Process、Format-Table 和 Start-Service。 动词选择也有一个惯例：“get”用于检索数据、“set”用于插入或更新数据、“format”用于格式化数据、“out”用于将输出定向到目标等。

建议 cmdlet 创建者为每个 cmdlet 随附一个帮助文件。 **Get-Help** cmdlet 显示任何 cmdlet 的帮助文件。 例如，可以使用以下语句来获取有关 `Get-ChildItem` cmdlet 的帮助：

PowerShell	
<code>Get-Help Get-ChildItem -detailed</code>	

## 什么是 PowerShell 模块？

Cmdlet 在模块中提供。 PowerShell 模块是一个 DLL，其中包含用于处理每个可用 cmdlet 的代码。 可以通过加载包含 cmdlet 的模块，将这些 cmdlet 载入 PowerShell。 可以使用 `Get-Module` 命令获取已加载的模块列表：

PowerShell	
<code>Get-Module</code>	

此命令的输出如下所示：


输出

复制

ModuleType	Version	Name	ExportedCommands
-----	-----	----	-----
Manifest	3.1.0.0	Microsoft.PowerShell.Management	{Add-Computer, Add-Content, Checkpoint-Computer, Clear-Con...
Manifest	3.1.0.0	Microsoft.PowerShell.Utility	{Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Binary	1.0.0.1	PackageManagement	{Find-Package, Find-PackageProvider, Get-Package, Get-Pack...
Script	1.0.0.1	PowerShellGet	{Find-Command, Find-DscResource, Find-Module, Find-RoleCap...
Script	2.0.0	PSReadline	{Get-PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PS...

## 什么是 Az 模块？

“Az”是 Azure PowerShell 模块的正式名称，其中包含用于处理 Azure 功能的 cmdlet。 它包含数百个 cmdlet，可以控制几乎每个 Azure 资源的所有方面。 可用于资源组、存储、虚拟机、Azure Active Directory、容器、机器学习等。 此模块是 [GitHub](#) 中提供的开源组件。

 **备注**


你可能见过或使用过使用 `-AzureRM` 格式的 Azure PowerShell 命令。 2018 年 10 月，我们宣布将“AzureRM”模块替换为“Az”模块。 此新模块有几项功能，值得注意的是缩短了 `-Az` cmdlet 名词的前缀，而非 `-AzureRM` cmdlet 名词的前缀。“Az”模块附带与“AzureRM”模块的后向兼容性，因此 `-AzureRM` cmdlet 格式有效，但应转换为“Az”模块并使用前面的 `-Az` 命令。

## 安装 Az 模块

名为“PowerShell 库”的全局存储库中提供了 Az 模块。 可以通过 `Install-Module` 命令在本地计算机上安装该模块。 需要使用权限提升的 PowerShell shell 从 PowerShell 库安装模块。


若要安装最新的 Azure PowerShell 模块，请运行以下命令：

- 打开“开始”菜单，键入 Windows PowerShell。
- 右键单击“Windows PowerShell”图标并选择“以管理员身份运行”。
- 在“用户帐户控制”对话框中，选择“是”。
- 键入以下命令，然后按 Enter：


PowerShell	
<code>Install-Module -Name Az -AllowClobber -SkipPublisherCheck</code>	

默认情况下，这会为所有用户安装模块（由 `scope` 参数控制）。

该命令依赖于使用 NuGet 来检索组件，具体取决于已安装的 NuGet 版本。系统可能会提示下载并安装最新版本的 NuGet。

输出	
<p>NuGet provider is required to continue PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet provider must be available in 'C:\Program Files (x86)\PackageManagement\ProviderAssemblies' or 'C:\Users\&lt;username&gt;\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider by running</p> <p>'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install and import the NuGet provider now?</p>	

默认情况下，PowerShell 库未配置为 PowerShellGet 的受信任存储库。 首次使用 PSGallery 时会看到以下提示：

输出	
<p>You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from 'PSGallery'?</p>	

## 脚本执行失败

根据安全配置，`Import-Module` 可能会失败并返回如下所示的消息。

输出	复制
<pre>import-module : File C:\Program Files (x86)\WindowsPowerShell\Modules\az\0.7.0\Az.psm1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170. At line:1 char:1 + import-module Az + ~~~~~ + CategoryInfo          : SecurityError: (:) [Import-Module], PSSecurityException + FullyQualifiedErrorId : UnauthorizedAccess,Microsoft.PowerShell.Commands.ImportModuleCommand</pre>	

这通常表示执行策略“受限制”，即，无法执行从外部源下载的模块 - 包括 PowerShell 库。可以通过执行命令 `Get-ExecutionPolicy` 来检查是否存在这种情况。如果该命令返回“Restricted”，请执行以下操作：

- 打开权限提升的 PowerShell 命令提示符。
- 使用 `Set-ExecutionPolicy` cmdlet 将策略更改为“RemoteSigned”：

PowerShell	复制
<pre>Set-ExecutionPolicy RemoteSigned</pre>	

系统会提示你授予权限：

输出	复制
<pre>The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy? [Y] Yes  [A] Yes to All  [N] No   [L] No to All  [S] Suspend  [?] Help (default is "N"): Y</pre>	

然后，应可使用 `Import-Module` 来加载 cmdlet。

## 更新模块

如果收到警告或错误消息，指示已安装 Azure PowerShell 模块的某一版本，可以发出以下命令更新到最新版本：

PowerShell	复制
<pre>Update-Module -Name Az</pre>	

如同 `Install-Module` 命令，如果系统提示是否信任该模块，则回答“是”或“全部选是”。如果遇到问题，还可以使用 `Update-Module` 命令来重新安装模块。

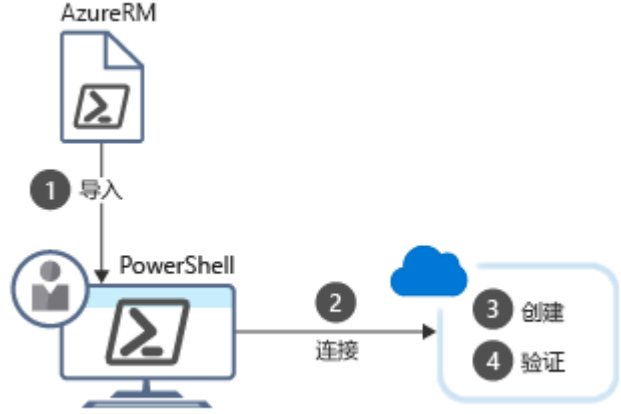
## 示例：如何使用 Azure PowerShell 创建资源组

加载 Azure 模块后，可以开始使用 Azure。让我们执行一个常见任务 - 创建资源组。我们知道，使用资源组可以统一管理相关的资源。创建新资源组是启动新 Azure 解决方案时要执行的首要任务之一。

需要执行四个步骤：

- 导入 Azure cmdlet。
- 连接到 Azure 订阅。
- 创建资源组。
- 验证创建是否成功（参见下文）。

下图显示了这些步骤的概述。



每个步骤对应于不同的 cmdlet。

## 导入 Azure cmdlet

在启动时，PowerShell 默认只加载核心 cmdlet。这意味着将不会加载需用于 Azure 的 cmdlet。加载所需 cmdlet 的最可靠方法是在启动 PowerShell 会话之前手动导入它们。

使用 `Import-Module` cmdlet 加载模块。此 cmdlet 具有许多参数，可处理各种情况。例如，它可以加载多个模块、某一特定模块版本、部分模块等。

例如，可以在权限提升的 PowerShell 会话中使用以下命令加载 Az 的所有 cmdlet：

PowerShell	复制
<pre>Import-Module Az</pre>	

### 提示

如果经常使用 Azure PowerShell，有两种方法可以自动执行模块加载过程。可以向 PowerShell 配置文件添加一个条目，以便在启动时导入 Azure 模块，或使用最新版本的 PowerShell，它会在使用 cmdlet 时自动加载包含模块。

## 连接

使用 Azure PowerShell 的本地安装时，在执行 Azure 命令之前需要进行身份验证。Connect-AzAccount cmdlet 会提示输入 Azure 凭证，然后连接到 Azure 订阅。它具有许多可选参数，但如果只需要交互式提示，则不需要任何参数：

PowerShell

Connect-AzAccount

复制

需要针对启动的个新 PowerShell 会话重复上述步骤，因为此模块不属于核心集。

## 使用订阅

如果你是 Azure 的新手，也许只有一个订阅。 但如果你使用 Azure 有一段时间，可能已创建了多个 Azure 订阅。 可将 Azure PowerShell 配置为针对特定的订阅执行命令。

每次只能在一个订阅中操作。 使用 `Get-AzContext` cmdlet 确定哪个订阅处于活动状态。 如果该订阅不是正确的订阅，可以更改它。

- 使用 `Get-AzSubscription` 命令获取帐户中所有订阅名称的列表。
- 通过传递要选择的订阅名称来更改订阅。

PowerShell

Select-AzSubscription -Subscription "Visual Studio Enterprise"

复制

## 获取所有资源组的列表

可以检索活动订阅中所有资源组的列表：

PowerShell

Get-AzResourceGroup

复制

为使视图显得更简洁，可以使用竖线“|”将 `Get-AzResourceGroup` 的输出发送到 `Format-Table` cmdlet。

PowerShell

Get-AzResourceGroup | Format-Table

复制

此命令的输出如下所示：

输出						复制
ResourceGroupName	Location	ProvisioningState	Tags	TagsTable	ResourceId	
-----	-----	-----	----	-----	-----	
cloud-shell-storage-southcentralus	southcentralus	Succeeded			/subscriptions/xxxxxxxx-d3ce-4172...	
ExerciseResources	eastus	Succeeded			/subscriptions/xxxxxxxx-d3ce-4172...	

## 创建资源组

我们知道，在 Azure 中创建资源时，始终要将这些资源放入资源组，以便于管理。 启动新应用程序时，资源组通常是要创建的首要对象之一。

可以使用 `New-AzResourceGroup` cmdlet 创建资源组。 必须指定名称和位置。 名称在订阅中必须唯一。 该位置决定资源组元数据的存储位置（出于符合性原因，这可能很重要）。 使用“美国西部”、“北欧”或“印度西部”等字符串指定位置。 与大多数 Azure cmdlet 一样，`New-AzResourceGroup` 具有许多可选参数；但核心语法是：

PowerShell

New-AzResourceGroup -Name <name> -Location <location>

复制

### 备注

请记住，我们将在可自动创建资源组的 Azure 沙盒中操作。 如果在自己的订阅中操作，将使用上述命令。

## 验证资源

`Get-AzResource` 列出 Azure 资源。 这对于验证资源组是否成功创建非常有用。

PowerShell

Get-AzResource

复制

与 `Get-AzResourceGroup` 命令一样，可以通过 `Format-Table` cmdlet 来精简视图。 下面我们将使用简写形式 `ft`：

PowerShell

Get-AzResource | ft

复制

此外，还可以根据特定的资源组进行筛选，以便仅列出与该组关联的资源：

PowerShell

Get-AzResource -ResourceGroupName ExerciseResources

复制

## 创建 Azure 虚拟机

使用 PowerShell 可以完成的另一个常见任务是创建 VM。

Azure PowerShell 提供 `New-AzVm` cmdlet 用于创建虚拟机。 该 cmdlet 具有许多参数，可以处理大量 VM 配置设置。 大多数参数都有合理的默认值，因此我们只需要指定五个参数：

- ResourceGroupName**：用来放置新 VM 的资源组。
- 名称**：Azure 中 VM 的名称。
- 位置**：预配 VM 的地理位置。
- 凭证**：包含 VM 管理员帐户的用户名和密码的对象。 我们将使用 `Get-Credential` cmdlet。 此 cmdlet 将提示输入用户名和密码，并将其打包成凭证对象。
- 映像**：VM 使用的操作系统映像。 这通常是某个 Linux 分发版或 Windows Server。

PowerShell

复制

```
New-AzVm
  -ResourceGroupName <resource group name>
  -Name <machine name>
  -Credential <credentials object>
  -Location <location>
  -Image <image name>
```

可按如上所示直接向 cmdlet 提供这些参数。或者，可以使用其他 cmdlet（例如 Set-AzVMOperatingSystem、Set-AzVMSourceImage、Add-AzVMNetworkInterface 和 Set-AzVMOSDisk）来配置虚拟机。

下面是将 Get-Credential cmdlet 和 -Credential 参数串接在一起的示例：

PowerShell

复制

```
New-AzVM -Name MyVm -ResourceGroupName ExerciseResources -Credential (Get-Credential) ...
```

AzVM 后缀特定于 PowerShell 中基于 VM 的命令。可以使用其他多个命令：

命令	说明
Remove-AzVM	删除 Azure VM。
Start-AzVM	启动已停止的 VM。
Stop-AzVM	停止正在运行的 VM。
Restart-AzVM	重启 VM。
Update-AzVM	更新 VM 的配置。

## 示例：获取 VM 的信息

可以使用 Get-AzVM -Status 命令列出订阅中的 VM。在此命令中还可以指定具有 -Name 属性的 VM。下面，我们将它分配到 PowerShell 变量：

PowerShell

复制

```
$vm = Get-AzVM -Name MyVM -ResourceGroupName ExerciseResources
```

有趣的是，可与此对象进行交互。例如，可以提取该对象并进行更改，然后使用 Update-AzVM 命令将更改推回到 Azure：

PowerShell

复制

```
$ResourceGroupName = "ExerciseResources"
$vm = Get-AzVM -Name MyVM -ResourceGroupName $ResourceGroupName
$vm.HardwareProfile.VmSize = "Standard_DS3_v2"

Update-AzVM -ResourceGroupName $ResourceGroupName -VM $vm
```

PowerShell 的交互模式适用于一次性任务。在本示例中，我们将在项目的生存期内使用相同的资源组，这意味着以交互方式创建它是合理的。针对此任务，交互模式通常比编写脚本并执行一次该脚本更快且更容易。

## 下一单元: 练习 - 在 Azure PowerShell 中使用脚本创建 Azure 资源

继续 >