

Course Introduction

Section 1

PowerShell Core

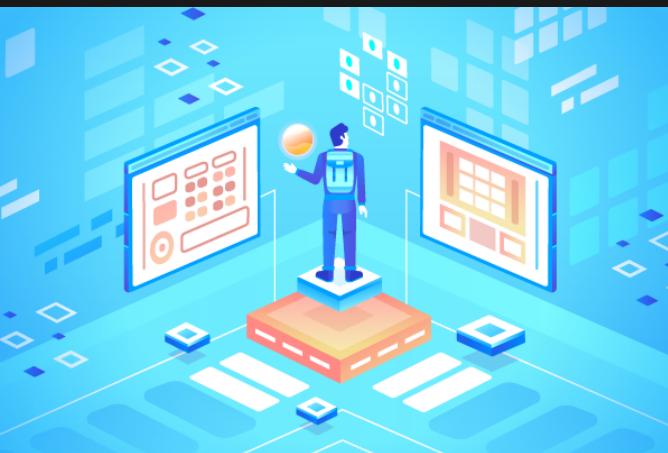
Section 2

**Create and Manage
Azure Resources**

Section 3

Conclusion

Section 4



[Introduction](#)

Course Introduction

Welcome to **Linux Academy**, and our Cross-Platform PowerShell in Azure course.

Version 6 of PowerShell has gone cross-platform! That means you can run PowerShell on both Windows and Linux. Commonly referred to as PowerShell Core, version 6 is built on top of .NET Core Runtime 2.0 and is completely open source.

In this course, you will become familiar with running PowerShell Core on Linux. In the first part of the course, we'll go through commands and scripting, and then move on to using the Az module and provisioning resources in Azure.

The prerequisites for this course include:

[Azure PowerShell Essentials](#)

[Microsoft Azure Fundamentals](#)

About the Training Architect



Chad Crowell

Training Architect



@chadmcrowell



[linkedin.com/in/chadmcrowell](#)



[The Power of PowerShell](#)[Commands and Scripting](#)[Azure PowerShell Az Module](#)[Different Shells for Different Spells](#)

PowerShell Is ...

A **shell** and a **scripting language**

```
Get-Command -Verb Get -Module PSReadLine,PowerShellGet -Syntax
```

Cmdlet Parameter

Switch

script.ps1 **more**

```
$i = 1
while ($i -le 5)
{
    ``$i = $i"
    $i = $i + 1
}
```

script.ps1 **more**

```
$var = 1
if ($var -eq 1)
{
    "yes it does equal one"
}
```

[Next](#)



PowerShell Is ...

A **shell** and a **scripting language**

A shell is a user interface for accepting text-based commands in order to access the operating system's services.

Cmdlet

Parameter

Switch

script.ps1

more

```
$i = 1
while ($i -le 5)
{
    ``$i = $i"
    $i = $i + 1
}
```

script.ps1

more

```
$var = 1
if ($var -eq 1)
{
    "yes it does equal one"
}
```

Next



The Power of PowerShell

Commands and Scripting

Azure PowerShell Az Module

Different Shells for Different Spells

PowerShell Is ...

A **shell** and a **scripting language**

A scripting language is a programming language for a special runtime environment that PSReadLine, PowerShellGet -Syntax automates the execution of tasks.

Scripting languages are often interpreted (rather than compiled).

Switch

script.ps1

more

```
$i = 1
while ($i -le 5)
{
    ``$i = $i"
    $i = $i + 1
}
```

script.ps1

more

```
$var = 1
if ($var -eq 1)
{
    "yes it does equal one"
}
```

Next



Linux Academy



Commands

Cmdlets

`Get-ChildItem` `Get-Help` `Get-Command`
`Set-Location` `Get-Alias` `Get-Process`

...more

Pipeline

`Get-ChildItem | Where-Object Length -gt 1kb`
`Get-Process | Export-Csv procs.csv`
`Dir | Out-File listdirectory.txt`

...more

PS Remoting

```
$session = New-PSSession -HostName $hostname -UserName $user  
Enter-PSSession -Session $session
```

...more

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

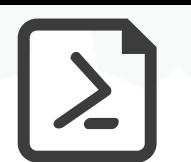
Your user ID will be \$UserID with a temporary password of \$TemporaryPassword and our website is \$Website. We can be reached at \$PhoneNumber.

Thanks!
Linux Academy Support Team
"@

\$WelcomeLetter

Next





Commands

Cmdlets

[Get-ChildItem](#) [Get-Help](#) [Get-Command](#)`Get-ChildItem -Path /tmp -Name ...more`

Pipeline

`Get-ChildItem | Where-Object Length -gt 1kb``Get-Process | Export-Csv procs.csv``Dir | Out-File listdirectory.txt ...more`

PS Remoting

`$session = New-PSSession -HostName $hostname -UserName $user``Enter-PSSession -Session $session`

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

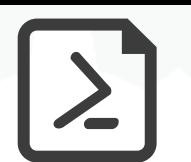
```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

Your user ID will be \$UserID with a temporary password of \$TemporaryPassword and our website is \$Website. We can be reached at \$PhoneNumber.

Thanks!
Linux Academy Support Team
"@

```
$WelcomeLetter
```

[...more](#)[Next](#)



Commands

Cmdlets

Get-ChildItem

Get-Help

Get-Command

Get-Help Format-Table

...more

Pipeline

Get-ChildItem | Where-Object Length -gt 1kb

Get-Process | Export-Csv procs.csv

Dir | Out-File listdirectory.txt ...more

PS Remoting

\$session = New-PSSession -HostName \$hostname -UserName \$user

Enter-PSSession -Session \$session

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

```
Your user ID will be $UserID with a temporary password of  
$TemporaryPassword  
and our website is $Website. We can be reached at $PhoneNumber.
```

```
Thanks!  
Linux Academy Support Team  
"@
```

\$WelcomeLetter

...more

Next



The Power of PowerShell

Commands and Scripting

Azure PowerShell Az Module

Different Shells for Different Spells



Commands

Cmdlets

Get-ChildItem

Get-Help

Get-Command

Get-Command -ParameterName *Auth* -ParameterType AuthenticationMechanism

Pipeline

Get-ChildItem | Where-Object Length -gt 1kb

Get-Process | Export-Csv procs.csv

Dir | Out-File listdirectory.txt ...more

PS Remoting

\$session = New-PSSession -HostName \$hostname -UserName \$user

Enter-PSSession -Session \$session

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

```
Your user ID will be $UserID with a temporary password of  
$TemporaryPassword  
and our website is $Website. We can be reached at $PhoneNumber.
```

```
Thanks!  
Linux Academy Support Team  
"@
```

\$WelcomeLetter

...more

Next



The Power of PowerShell

Commands and Scripting

Azure PowerShell Az Module

Different Shells for Different Spells



Commands

Cmdlets

Set-Location /tmp

Set-Location

Get-Alias

Get-Process

...more

Pipeline

Get-ChildItem | Where-Object Length -gt 1kb

Get-Process | Export-Csv procs.csv

Dir | Out-File listdirectory.txt ...more

PS Remoting

\$session = New-PSSession -HostName \$hostname -UserName \$user

Enter-PSSession -Session \$session

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

Your user ID will be \$UserID with a temporary password of \$TemporaryPassword and our website is \$Website. We can be reached at \$PhoneNumber.

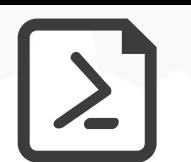
Thanks!
Linux Academy Support Team
"@

\$WelcomeLetter

...more

Next





Commands

Cmdlets

```
Get-Alias -Name gp*, sp* -Exclude *ps
```

[Set-Location](#)[Get-Alias](#)[Get-Process](#)[...more](#)

Pipeline

```
Get-ChildItem | Where-Object Length -gt 1kb
```

```
Get-Process | Export-Csv procs.csv
```

```
Dir | Out-File listdirectory.txt      ...more
```

PS Remoting

```
$session = New-PSSession -HostName $hostname -UserName $user
```

```
Enter-PSSession -Session $session
```

Scripting

```
$FirstName='Chad'  
$LastName='Crowell'  
$UserID=$FirstName+'.'+$LastName  
$PhoneNumber='555-555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

Your user ID will be \$UserID with a temporary password of \$TemporaryPassword and our website is \$Website. We can be reached at \$PhoneNumber.

Thanks!
Linux Academy Support Team
"@

```
$WelcomeLetter
```

[...more](#)[Next](#)

The Power of PowerShell

Commands and Scripting

Azure PowerShell Az Module

Different Shells for Different Spells



Commands

Cmdlets

```
Get-Process | Where-Object {$_.WorkingSet -gt 20000000}
```

[Set-Location](#)[Get-Alias](#)[Get-Process](#)[...more](#)

Pipeline

```
Get-ChildItem | Where-Object Length -gt 1kb
```

```
Get-Process | Export-Csv procs.csv
```

```
Dir | Out-File listdirectory.txt ...more
```

PS Remoting

```
$session = New-PSSession -HostName $hostname -UserName $user
```

```
Enter-PSSession -Session $session
```

Scripting

```
$FirstName='Chad'  
$LastName='Hall'  
$Email=$FirstName+'.'+$LastName+'@'+$PhoneNumber  
$PhoneNumber='555-5555'  
$Website='https://linuxacademy.com'  
$TemporaryPassword=GET-RANDOM
```

```
$WelcomeLetter=@"  
Hello and welcome to Linux Academy, $FirstName !
```

```
Your user ID will be $UserID with a temporary password of  
$TemporaryPassword  
and our website is $Website. We can be reached at $PhoneNumber.
```

```
Thanks!  
Linux Academy Support Team  
"@
```

```
$WelcomeLetter
```

[...more](#)[Next](#)

[The Power of PowerShell](#)[Commands and Scripting](#)[Azure PowerShell Az Module](#)[Different Shells for Different Spells](#)

The Az Module

```
Install-Module -Name Az -AllowClobber -Scope AllUsers  
  
Connect-AzAccount  
  
Get-Command -Verb Get -Noun AzVM* -Module Az.Compute  
  
Get-InstalledModule -Name AzureRM -AllVersions  
  
Uninstall-AzureRm
```

* It is highly recommended to migrate to Az module.

[Azure-RM vs. Az](#)

Find Common Commands

```
Get-Command -Verb Get -Noun AzResource -Module Az.Resources  
  
Get-Command -Verb Get -Noun AzStorage -Module Az.Storage  
  
Get-Command -Verb Get -Noun AzKey -Module Az.KeyVault  
  
Get-Command -Verb Get -Noun AzWeb -Module Az.Websites  
  
Get-Command -Verb Get -Noun AzSql -Module Az.Sql
```

[Create a VM](#)[Create a Storage Account](#)[Create a Key Vault](#)[Create a Container](#)[Create a Web App](#)[Create a Service Principal](#)[Create a New Remote Session](#)[Create a SQL DB](#)[Cloud Shell](#)

In-browser shell managed by Microsoft with built-in tools and language support in Bash and PowerShell

[Next](#)

The Power of PowerShell

Commands and Scripting

Azure PowerShell Az Module

Different Shells for Different Spells



PowerShell vs. Bash

Bash

- Command line interpreter
- Scripting environment
- Treats output as a string

PowerShell

- Command line interpreter
- Scripting environment
- Framework
- Programming language
- Treats output as an object

PowerShell and Bash

- Alias to Linux commands
- Pipelining enabled
- Use of strings, arrays, and logic flow
- Use of functions

```
Get-Alias | more
```

```
Get-PSProvider
```

```
Get-PSDrive
```

```
Get-ChildItem | Where-Object Length -gt 1kb
```

```
$items = (Get-ChildItem).Count
```

```
$array = "Chad", "Crowell"
```

```
for ($i = 0; $i -le 5; $i++)  
{  
    "`$i = $i"  
}
```

```
function Write-HelloWorld()  
{  
    'Hello World'  
}
```





Log in to Your Azure Account

1

Interactive Sign-in

Connect-AzAccount:

This cmdlet presents a token string to authenticate and connect to Azure

3

Managed Identity

Connect-AzAccount -Identity:

App-only access token to access other resources. Only available for Azure resources.

Create a Service Principal

This can be used for the **service principal method of logging in** or for the **managed identity method**.

2

Service Principal

Connect-AzAccount -ServicePrincipal:

For non-interactive Azure accounts. **Password-based** or **certificate-based** authentication managed with Azure AD.

3

Multi-Tenant or Multi-Cloud

Connect-AzAccount -Tenant or -Environment:

If you are a cloud solution provider, the tenant flag must be used. Government and other regulated environments must use the environment flag.

What Is a Managed Identity?

Next



A Service Principal Is...

An identity that can be used with applications, hosted services, and automated tools in order to access Azure resources.

Allows you to have more granular control over what that service principal can do. You can assign a role to a service principal with **Access control (IAM)**.

* It is recommended to use service principals with automated tools rather than allowing user access identity.

- [cube] Overview
- [log] Activity log
- [user] Access control (IAM)
- [tag] Tags
- [lightning] Events

```
PS /> $sp = New-AzADServicePrincipal -DisplayName my-spl
PS /> $BSTR = [System.Runtime.InteropServices.Marshal]::
SecureStringToBSTR($sp.secret)
PS /> $UnsecureSecret = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
PS /> $UnsecureSecret
c982f555-f9ad-6fff-yy01-66nov3579b29
```

Service Principal Password

Next



Run As a Job

PowerShell Jobs let you run cmdlets in the background or do multiple tasks on Azure at once, from inside a single PowerShell session.

```
PS /> $job = New-AzVM -Name myNewVM -AsJob -ImageName UbuntuLTS
```

```
PS /> Get-Job
```

```
PS /> $vm = Receive-Job $job
```

About Jobs

Azure Contexts with PowerShell Jobs

Your Azure credentials must be shared with the PowerShell session in order to run the command as a job. If you have multiple subscriptions, you can set a context to ensure your commands are applied to the correct subscription.

```
PS /> Get-AzContext -ListAvailable
```

```
PS /> Get-AzSubscription -SubscriptionName 'mysub' |  
Set-AzContext -Name 'newContextName'
```

Next



[Authentication Methods](#)[Service Principals](#)[Using RunAs to Background Tasks](#)[Deployment Modes](#)

Deploying Azure Resources with PowerShell

Complete Mode

A resource is deleted if that resource is not specified in the template.

If the template contains a condition statement, and the condition evaluates to false for a resource, the resource **will not be deleted** if you use the **API version '2019-05-10'**. The resource **will be deleted** if you use the **API version later than '2019-05-10'**.

If you deploy more than one resource group in the same template, the resources in the secondary resource group will not be deleted.

[Learn More](#)

Incremental Mode

Resources in a template can be added. Everything else in the environment is left unchanged, even if not in the template.

Changes to individual properties of a resource are left unchanged. If you make changes to the resource from within the template, the entire resource is redeployed. Properties of the resource that are not specified in the template are set to the default value. The default value is set by the resource provider.

The template always contains the final state of the resource.

[Learn More](#)

Both Modes:

All resources must be specified in the template. If the resource already exists and the settings are unchanged, no action is taken. If the property values for a resource are updated, the resource is updated (not deleted). If the location or type of the resource is updated, the deployment will fail with an error.

[You Did It!](#) [What's Next?](#)

You Did It!

Congratulations on completing the course!

Hopefully you know understand how PowerShell is an excellent tool for automating and scripting resources in Azure and beyond!

Make sure to rate this course, and if there are any lessons that stood out, please give a thumbs up!

If you liked it, please share this course with a friend!

Click the links below this video to become acquainted with our other PowerShell courses.

