

Infrastructure Testing with Pester

INTRODUCTION TO TESTING



Adam Bertram

MICROSOFT CLOUD AND DATACENTER MANAGEMENT MVP

@adbertram www.adamtheautomator.com





Your environment changes frequently

You automate (script) infrastructure changes

You don't completely trust the scripts that you run

You need to be 100% certain your infrastructure is how you expect



What This Course is About



Not just about testing
PowerShell code



Not Pester 101
<http://bit.ly/pluralsight-pester>



Building code to
ensure your
infrastructure is how
you think it is



Knowledge Prerequisites



Knowing what to test



PowerShell

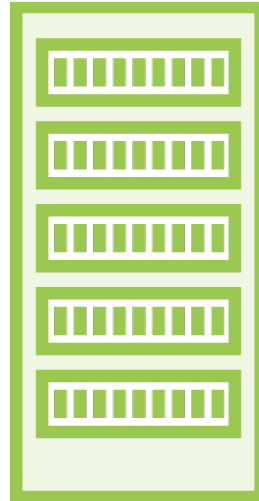


Pester

Hardware Prerequisites



A computer



Hyper-V Host



Windows Server 2012
R2 ISO

Installing and Using Pester



Demo



Downloading Pester

Installing Pester

Checking Pester version



Demo



What makes up a Pester test

How to invoke tests



The Test Hierarchy: Testing in Layers



The Testing "Layer" Stack



Unit Tests



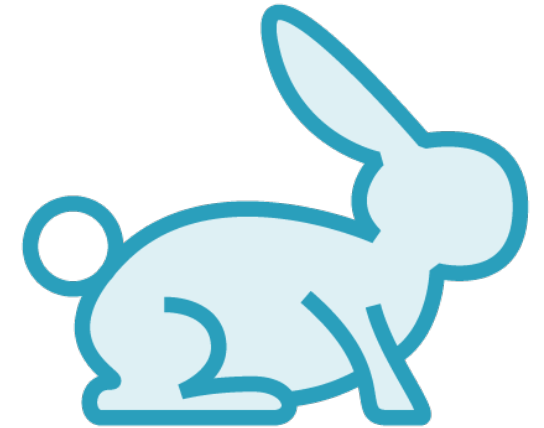
Test code
execution **ONLY**



Does not touch
the environment



Specific to the
language



Fast!

Testing an Active Directory User Creation: Unit Test

```
New-AdUser -Name adammmm -Department Accounting
```



Unit Test

```
New-AdUser -Name adam -Department Accounting FAIL
```

Execution



Demo



Brief explanation of what a Pester test consists of

Build a unit test for a simple function

Learn a little about mocking



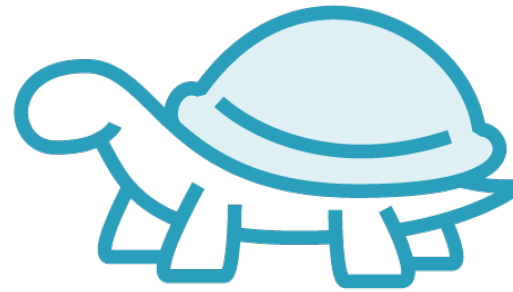
Integration Tests



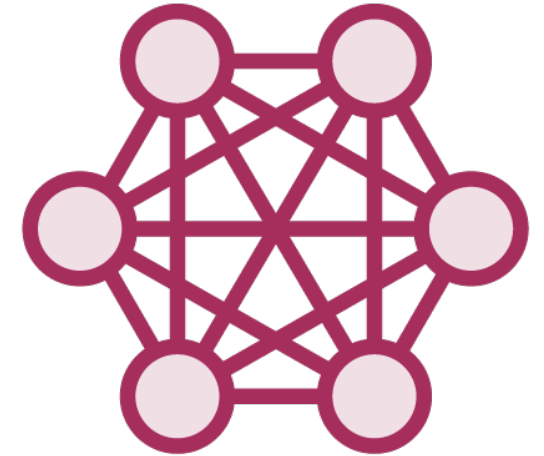
Tests
environmental
changes



Not dependent
on code



Much slower
than unit tests



Has
environmental
dependencies

Testing an Active Directory User Creation: Integration Test

```
New-AdUser -Name adammmm -Department Accounting
```



Integration Test

```
$user = (Get-AdUser -Identity adam -Properties Department).Where({  
    $_.Department -eq Accounting  
})
```

```
$user -ne $null
```

FAIL



Demo



Build an integration test for a function

Show how integration tests compare to unit tests

Demonstrate the tear down needed for integration tests



Acceptance Testing



Tests the "effect" of code



Tests the user experience

Testing an Active Directory User Creation: Acceptance Test

```
New-AdUser -Name adammmm -Department Accounting
```

PS1

Acceptance Test



Adam

Authentication Request



Authentication Denied



Active Directory

FAIL

PS1



Demo



Build an acceptance test for a function

Show an acceptance test's relation to an integration test

Validate that the user that was created actually can authenticate



Infrastructure Testing



An informal definition



A mix-up of
integration and
acceptance testing



Tests the environment;
not code

Infrastructure Testing Example

Create an Active Directory user you *think* is called "adam" somehow.

Infrastructure Test

```
$user = (Get-AdUser -Identity adam -Properties Department).Where({  
    $_.Department -eq Accounting  
})  
  
$user -ne $null
```

FAIL



Adam

Authentication Request



Authentication Denied



Active Directory

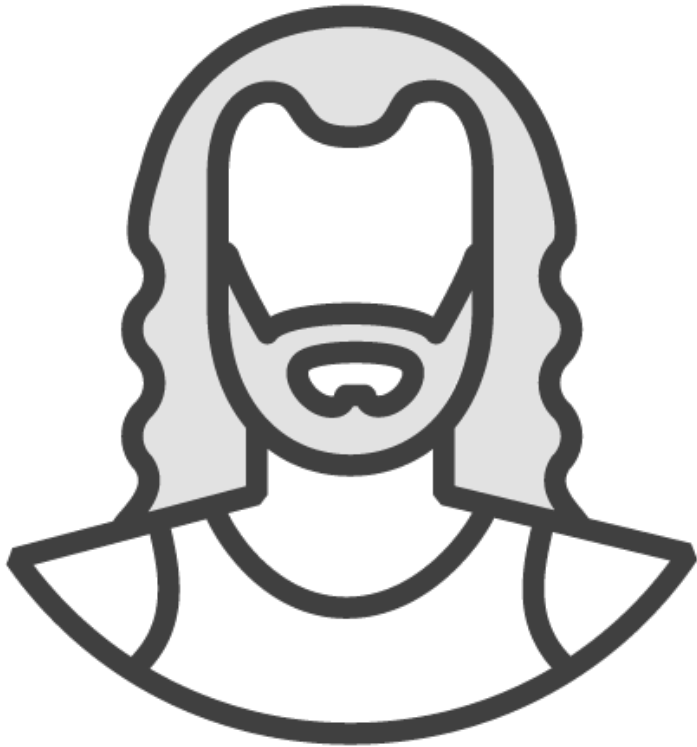
PS1



Course Scenario



Meet George



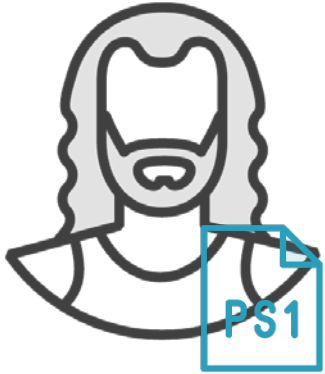
System administrator / IT professional

PowerShell scripter extraordinaire

Wears lots of hats

Automation junkie but still fights unnecessary fires

Get-TestExPowercat - Powershell Script



Create
Hyper-V VM



Promote to
domain
controller



Read CSV of
AD objects



- Organizational Units
- Groups
- Users

Course introduction

**Unit, integration, acceptance and
infrastructure testing walkthrough**

Pester introduction

Course scenario

