Validating Infrastructure Changes in Pester

Adam Bertram

MICROSOFT CLOUD AND DATACENTER MANAGEMENT MVP

@adbertram   www.adamtheautomator.com

Overview

Using External Sources of Test Input

Building the Test Framework

Building Tests for VM Creations

Building Tests for Automated OS Deployment

Test Inputs and Data Sources

Expected value + actual value = Pester test

Pester test compares values

Test varies widely

Hard to manipulate data in many tests

It is possible to extract values to external source

```
{ "ConfigurationItems": [ {

    "VirtualMachine": []

  },  { "OperatingSystem": [] }

}
```

Coordinating Configuration Items

JSON is a structured format that's easy to read and consume

Demo

Create expected values

Create the configuration item

Building the Test Framework

Demo

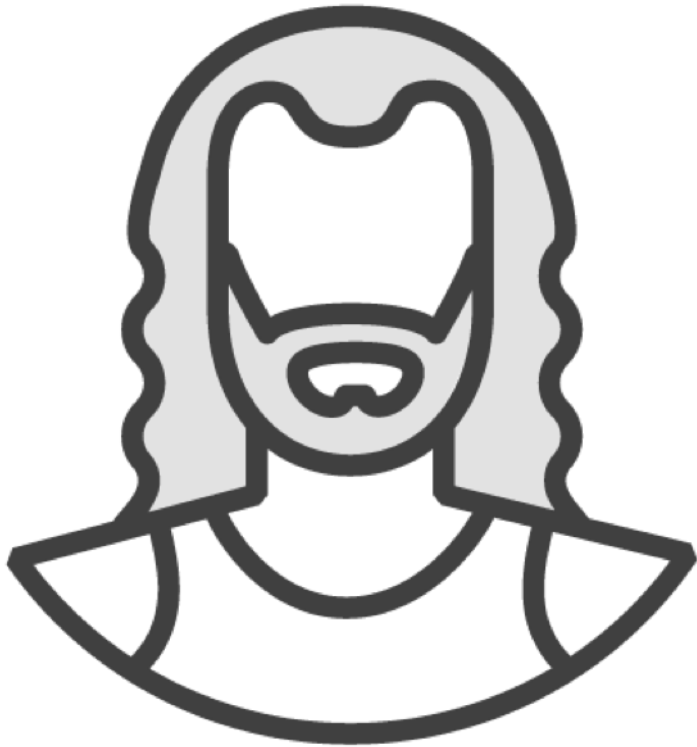**Build a Pester test framework to use in individual tests**

Demo

Reading JSON into Pester tests through PowerShell

George's Progress – VM Creation

Check-in on George's progress

VM Creation

Demo

George automates VM creation

Building Tests for VM Creations

Define the Values in the Test

Virtual Machine

- Name

- Memory

- Generation

- SwitchName

- Path

VHD

- Path

- IsDynamic

- AttachedToVMName

- Size

Demo

Defining configuration items for VM creation

Demo

Run the VM infrastructure tests

Build Pester tests to compare values

George's Progress – OS Deployment

Check-in on George's progress

Automate OS Deployment

Minimal code in automating Windows install

Use XML files

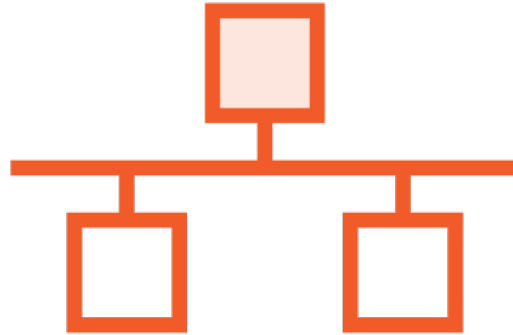**Locate XML files correctly**

Demo

Automate Windows OS deployment

Building Tests for Automated OS Deployments

Define Important Elements

OS Install Settings

NIC Settings

Firewall

Disk Configuration

Demo

Extract important values from AutoUnattend.xml

Bring values to configuration items JSON file

Demo

Build Pester tests to test AutoUnattend settings

Summary

Built Pester test template

Automate VM Creation and Windows Install

Explored George's code

Built first Pester tests