INFRASTRUCTURE TESTING WITH PESTER

- Brandon Olin
- @devblackops
- devblackops.io
- github.com/devblackops

WHAT IS SOFTWARE TESTING?

@devblackops

WHAT IS SOFTWARE TESTING?

- An investigation conducted to determine the quality of a software product or service under test

@devblackops

WHAT IS SOFTWARE TESTING?

- An investigation conducted to determine the quality of a software product or service under test
- Provides an objective, independent view of software quality

@devblackops

WHAT IS SOFTWARE TESTING?

- An investigation conducted to determine the quality of a software product or service under test
- Provides an objective, independent view of software quality
- Verifies that the software is fit for use

@devblackops

SO SOFTWARE TESTING...

Verifies one or more of the following:

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development
- Responds correctly to all kinds of inputs

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development
- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development
- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time
- It is sufficiently usable (fit for purpose)

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development
- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time
- It is sufficiently usable (fit for purpose)
- Can be run it its intended environment(s)

SO SOFTWARE TESTING...

Verifies one or more of the following:

- Requirements that guided its design and development
- Responds correctly to all kinds of inputs
- Performs its functions within an acceptable time
- It is sufficiently usable (fit for purpose)
- Can be run it its intended environment(s)
- Achieves the result its stakeholders desire

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?
- Is the Windows service 'xyz' running?

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?
- Is the Windows service 'xyz' running?
- Is the certificate expired or about to expire?

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?
- Is the Windows service 'xyz' running?
- Is the certificate expired or about to expire?
- Is the current state of the system what I expect?

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?
- Is the Windows service 'xyz' running?
- Is the certificate expired or about to expire?
- Is the current state of the system what I expect?
- Are these infrastructure properties correct?

WHAT IS INFRASTRUCTURE TESTING?

Validation of the operation of a system

- Does the website respond with 200 (OK)?
- Is the Windows service 'xyz' running?
- Is the certificate expired or about to expire?
- Is the current state of the system what I expect?
- Are these infrastructure properties correct?
- Is the system operating to customer expectations?

WHY TEST INFRASTRUCTURE?

WHY TEST INFRASTRUCTURE?

- To validate our **CURRENT** state matches our **EXPECTED** state

WHY TEST **INFRASTRUCTURE?**

- To validate our **CURRENT** state matches our **EXPECTED** state
- To **NOTIFY** us when they differ

WHY TEST INFRASTRUCTURE?

- To validate our **CURRENT** state matches our **EXPECTED** state
- To **NOTIFY** us when they differ
- To perform **SANITY CHECKS** before/after changes

WHY TEST INFRASTRUCTURE?

- To validate our **CURRENT** state matches our **EXPECTED** state
- To **NOTIFY** us when they differ
- To perform **SANITY CHECKS** before/after changes
- To provide **GUARDRAILS** for automation

EXAMPLES

BLUE/GREEN DEPLOYMENTS

@devblackops

VALIDATING SERVER SETTINGS

@devblackops

VERIFYING STATE BEFORE PROCEEDING

SO WHY DO WE TEST AGAIN?

SO WHY DO WE TEST AGAIN?

- So we can go **FAST**

SO WHY DO WE TEST AGAIN?

- So we can go **FAST**
- So we can be **SAFE**

SO WHY DO WE TEST AGAIN?

- So we can go **FAST**
- So we can be **SAFE**
- So we can bail when things are **UNSAFE**

TO PUT IT ANOTHER WAY…

“ CI/CD pipelines allow you to go 200mph. Tests are your seat belts, airbags, anti-lock brakes, traction control, collision avoidance, and autonomous driving systems so you can go 200mph without killing yourself or others.

- Me

@devblackops

CAN WE DO THIS WITH POWERSHELL?

YES! THAT'S WHY I'M HERE!

SOME POWERSHELL TESTING TOOLS

SOME POWERSHELL TESTING TOOLS

- Pester

SOME POWERSHELL TESTING TOOLS

- Pester
- Operation Validation Framework

SOME POWERSHELL TESTING TOOLS

- Pester
- Operation Validation Framework
- Assert

SOME POWERSHELL TESTING TOOLS

- Pester
- Operation Validation Framework
- Assert
- PSHealthZ

SOME POWERSHELL TESTING TOOLS

- Pester
- Operation Validation Framework
- Assert
- PSHealthZ
- Watchmen

SOME POWERSHELL
TESTING TOOLS

- Pester
- Operation Validation Framework
- Assert
- PSHealthZ
- Watchmen
- Others? (let me know)

PESTER

The ubiquitous test and mock framework for PowerShell

https://github.com/pester/Pester

PESTER

The ubiquitous test and mock framework for PowerShell

https://github.com/pester/Pester

- Describe
- Context
- It
- Should
- Mock
- etc...

PESTER

The ubiquitous test and mock framework for PowerShell

https://github.com/pester/Pester

- Describe
- Context
- It
- Should
- Mock
- etc...

Get-AnswerToUniverse.ps1

```
function Get-AnswerToUniverse {
    42
}
```

PESTER

The ubiquitous test and mock framework for PowerShell

https://github.com/pester/Pester

- Describe
- Context
- It
- Should
- Mock
- etc...

Get-AnswerToUniverse.ps1

```
function Get-AnswerToUniverse {
    42
}
```

Get-AnswerToUniverse.tests.ps1

```
. $PSScriptRoot/Get-AnswerToUniverse.ps1

Describe '[Universe]' {
    Context 'The answer to everything' {
        It 'The answer is [42]' {
            $answer = Get-AnswerToUniverse
            $answer | Should -Be 42
        }
    }
}
```

PESTER

The ubiquitous test and mock framework for PowerShell

https://github.com/pester/Pester

- Describe
- Context
- It
- Should
- Mock
- etc...

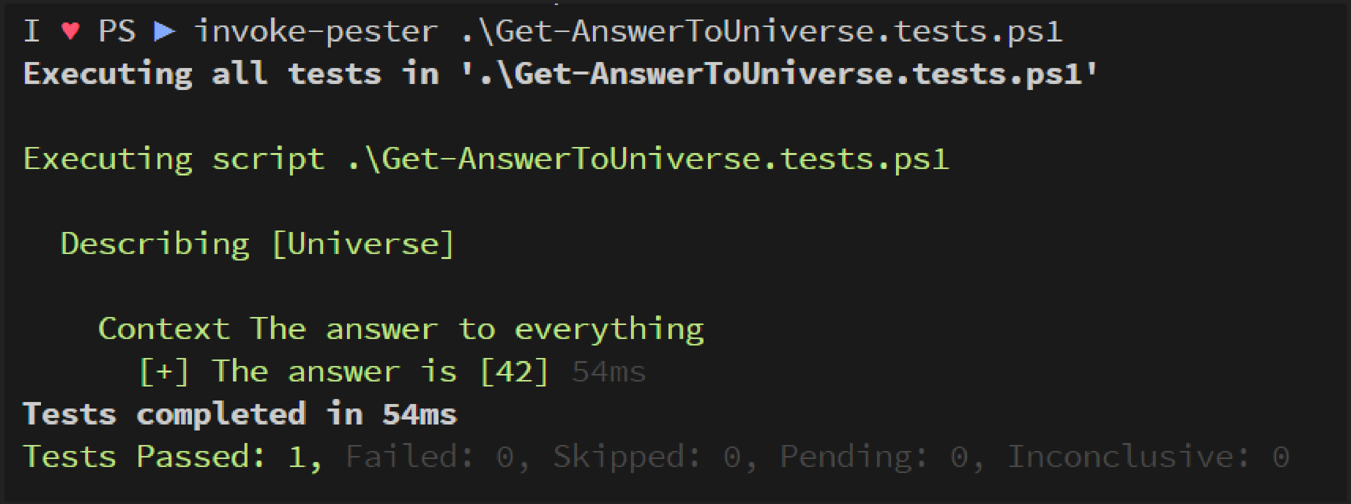
Get-AnswerToUniverse.ps1

```
function Get-AnswerToUniverse {
    42
}
```

Get-AnswerToUniverse.tests.ps1

```
. $PSScriptRoot/Get-AnswerToUniverse.ps1

Describe '[Universe]' {
    Context 'The answer to everything' {
        It 'The answer is [42]' {
            $answer = Get-AnswerToUniverse
            $answer | Should -Be 42
        }
    }
}
```

Result



```
I ♥ PS ▶ invoke-pester .\Get-AnswerToUniverse.tests.ps1
Executing all tests in '.\Get-AnswerToUniverse.tests.ps1'

Executing script .\Get-AnswerToUniverse.tests.ps1

  Describing [Universe]

    Context The answer to everything
      [+] The answer is [42] 54ms
Tests completed in 54ms
Tests Passed: 1, Failed: 0, Skipped: 0, Pending: 0, Inconclusive: 0
```

OPERATION VALIDATION FRAMEWORK

Framework for executing Pester tests contained in PS modules

https://github.com/PowerShell/Operation-Validation-Framework

OPERATION VALIDATION FRAMEWORK

Framework for executing Pester tests contained in PS modules

https://github.com/PowerShell/Operation-Validation-Framework

- Get-OperationValidation
- Invoke-OperationValidation

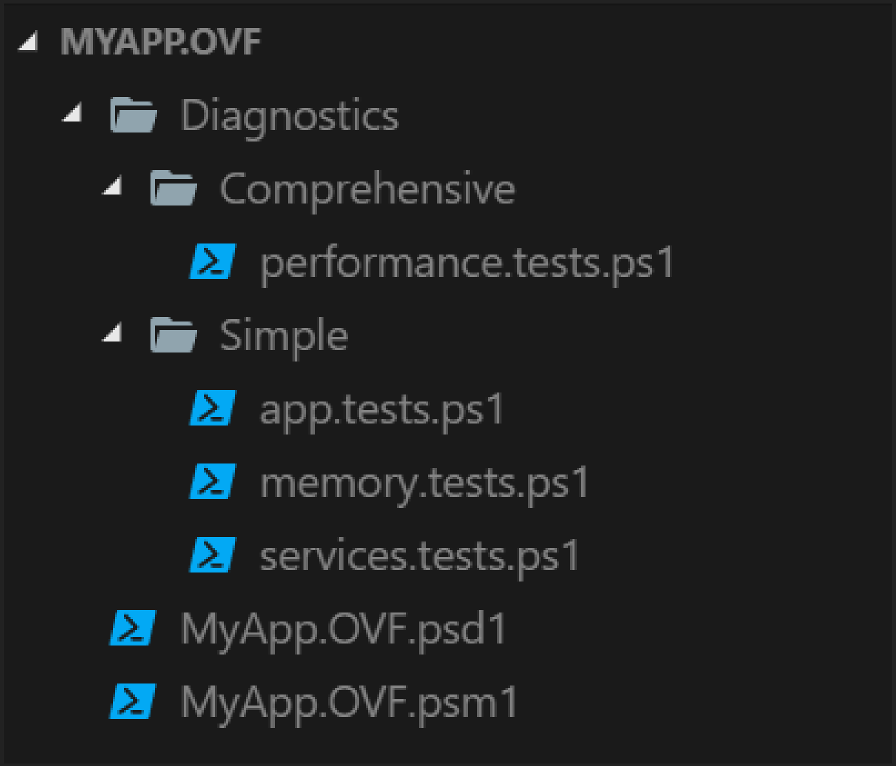
OPERATION VALIDATION FRAMEWORK

Framework for executing Pester tests contained in PS modules

https://github.com/PowerShell/Operation-Validation-Framework

- Get-OperationValidation
- Invoke-OperationValidation

Module Structure



```
MYAPP.OVF
  Diagnostics
    Comprehensive
      performance.tests.ps1
    Simple
      app.tests.ps1
      memory.tests.ps1
      services.tests.ps1
  MyApp.OVF.psd1
  MyApp.OVF.psm1
```

@devblackops

OPERATION VALIDATION FRAMEWORK

MyApp.OVF/Diagnostics/Simple/app.tests.ps1

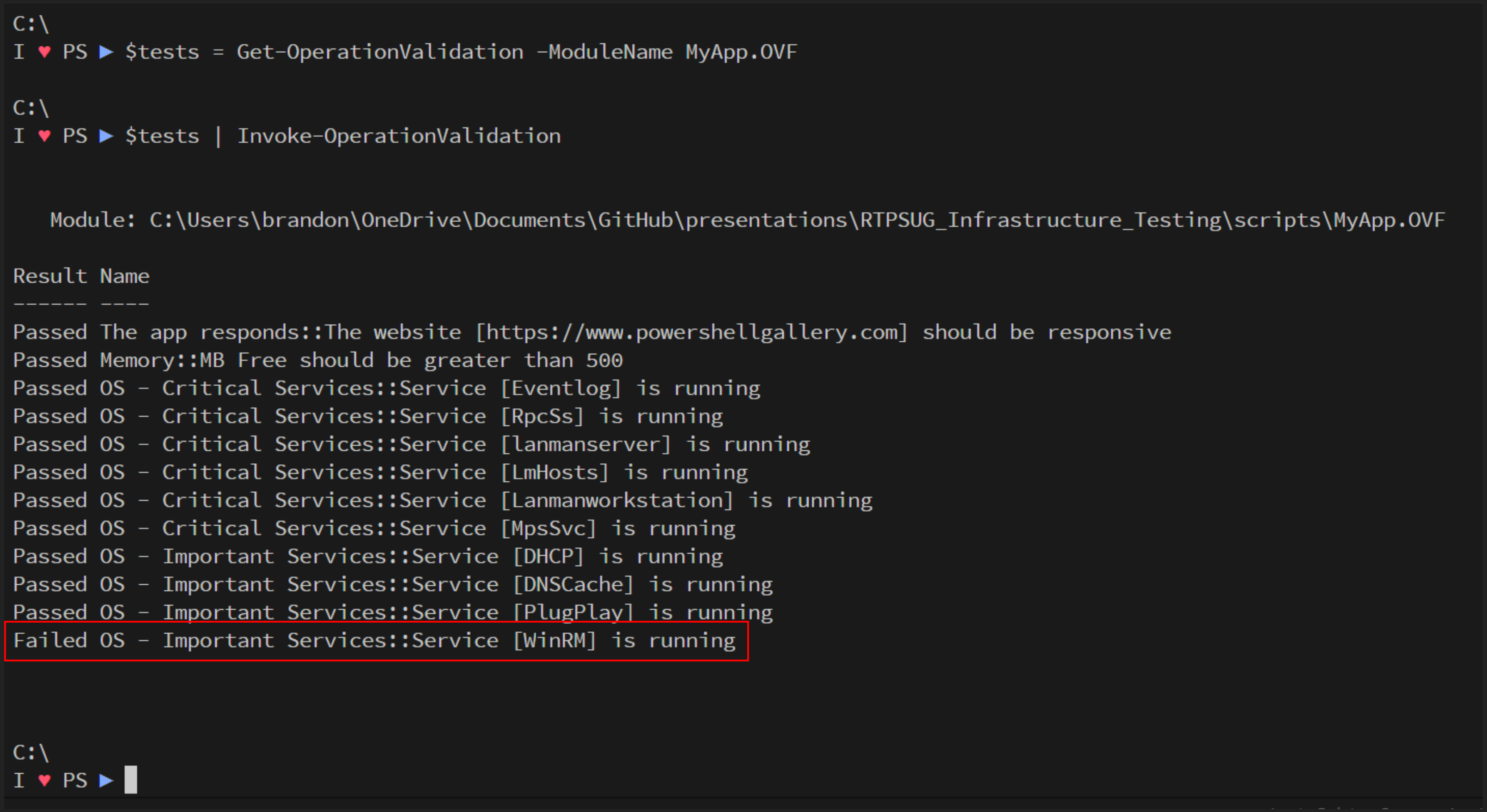
```
param(
    [string]$Url = 'https://www.powershellgallery.com',
    [string]$StatusCode = 200
)

Describe 'The app responds' -Tag 'App' {
    It "The website [$Url] should be responsive" {
        $response = Invoke-WebRequest -Uri $Url -UseBasicParsing
        $response.StatusCode | Should -Be $StatusCode
    }
}
```

OPERATION VALIDATION FRAMEWORK

MyApp.OVF/Diagnostics/Simple/services.tests.ps1

```
$critical  = @(
    'Eventlog', 'RpcSs', 'lanmanserver', 'LmHosts', 'Lanmanworkstation', 'MpsSvc'
)
$important = @('DHCP', 'DNSCache', 'PlugPlay' ,'WinRM')

describe 'OS - Critical Services' -Tag Critical {
    $critical.Foreach({
        it "Service [$_] is running" {
            (Get-Service $_).Status | Should -Be 'Running'
        }
    })
}

describe 'OS - Important Services' -Tag Important {
    $important.Foreach({
        it "Service [$_] is running" {
            (Get-Service $_).Status | Should -Be 'Running'
        }
    })
}
```

OPERATION VALIDATION FRAMEWORK



```
C:\
I ♥ PS ▶ $tests = Get-OperationValidation -ModuleName MyApp.OVF

C:\
I ♥ PS ▶ $tests | Invoke-OperationValidation


   Module: C:\Users\brandon\OneDrive\Documents\GitHub\presentations\RTPSUG_Infrastructure_Testing\scripts\MyApp.OVF

Result Name
------ ----
Passed The app responds::The website [https://www.powershellgallery.com] should be responsive
Passed Memory::MB Free should be greater than 500
Passed OS - Critical Services::Service [Eventlog] is running
Passed OS - Critical Services::Service [RpcSs] is running
Passed OS - Critical Services::Service [lanmanserver] is running
Passed OS - Critical Services::Service [LmHosts] is running
Passed OS - Critical Services::Service [Lanmanworkstation] is running
Passed OS - Critical Services::Service [MpsSvc] is running
Passed OS - Important Services::Service [DHCP] is running
Passed OS - Important Services::Service [DNSCache] is running
Passed OS - Important Services::Service [PlugPlay] is running
Failed OS - Important Services::Service [WinRM] is running



C:\
I ♥ PS ▶ █
```

@devblackops

ASSERT

A set of advanced assertions for Pester to simplify test authoring

https://github.com/nohwnd/Assert

ASSERT

A set of advanced assertions for Pester to simplify test authoring

https://github.com/nohwnd/Assert

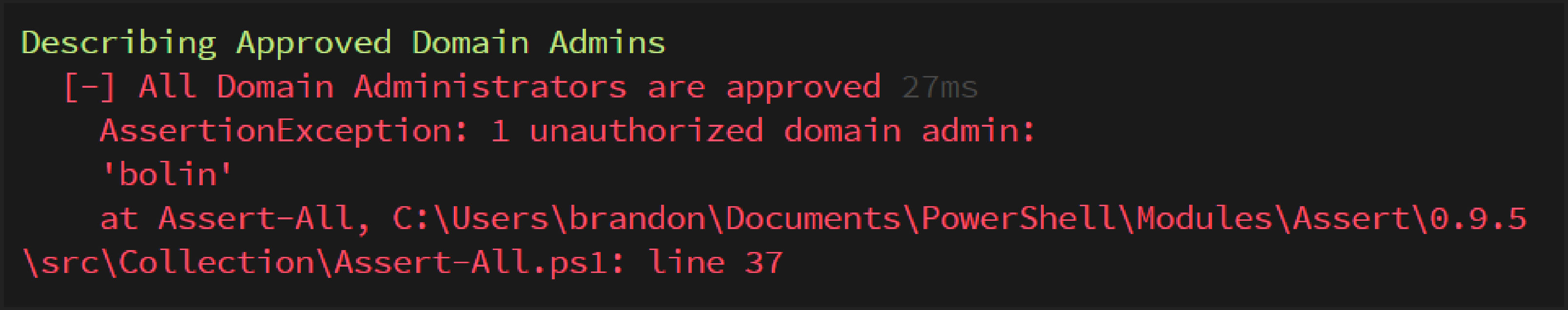
- Assert-Contain
- Assert-Equivalent
- Assert-Like
- Assert-Same
- Assert-StringEqual
- Assert-Type
- etc...

ASSERT

```powershell
function Get-ApprovedDomainAdmins {
    @('mwilliams', 'jthompson', 'kreyes')
}

function Get-ActualDomainAdmins () {
    @('mwilliams', 'jthompson', 'kreyes', 'bolin')
}

Describe 'Approved Domain Admins' {
    It 'All Domain Administrators are approved' {
        $failMsg  = "<actualFilteredCount> unauthorized domain admin: `n'<actualFiltered>'"
        $actual   = Get-ActualDomainAdmins
        $approved = Get-ApprovedDomainAdmins
        $actual | Assert-All {$_ -in $approved} -CustomMessage $failMsg
    }
}
```

@devblackops

ASSERT



```
Describing Approved Domain Admins
  [-] All Domain Administrators are approved 27ms
    AssertionException: 1 unauthorized domain admin:
    'bolin'
    at Assert-All, C:\Users\brandon\Documents\PowerShell\Modules\Assert\0.9.5
\src\Collection\Assert-All.ps1: line 37
```

@devblackops

PSHEALTHZ

HTTP(s) listener that executes OVF tests over REST API

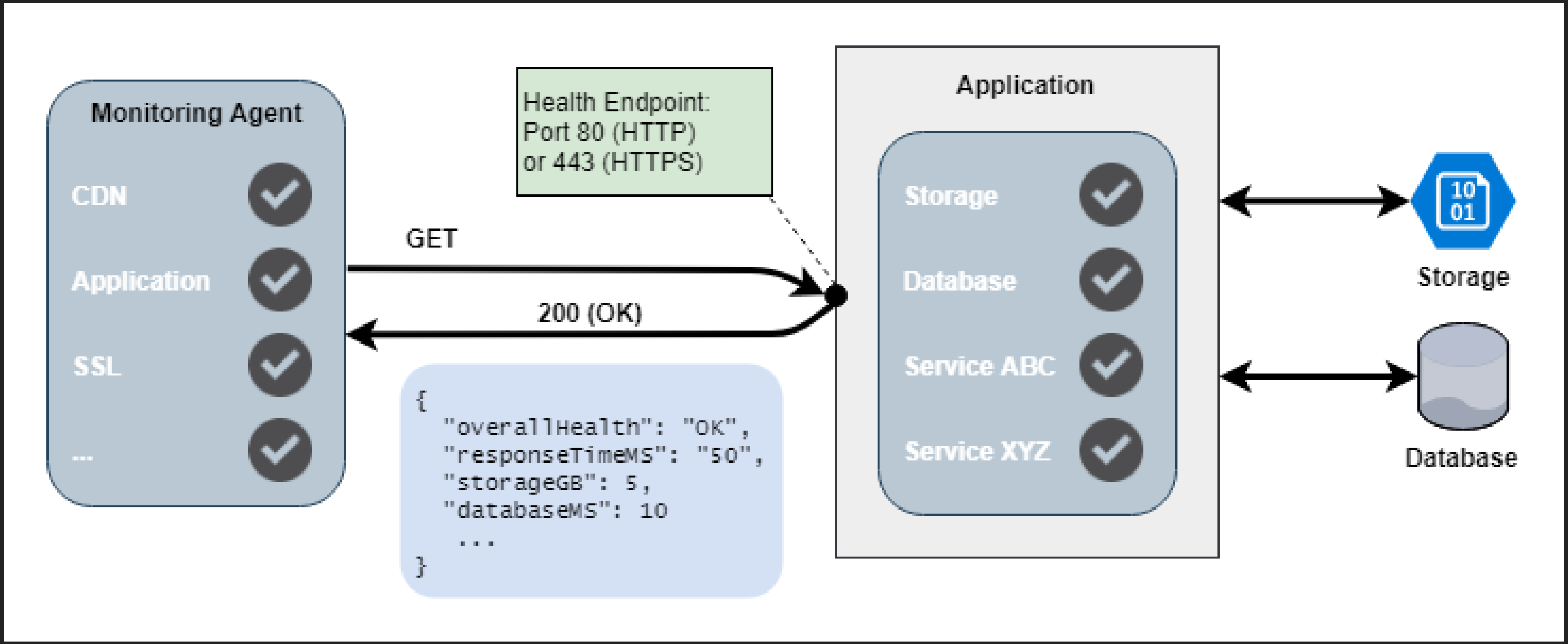
https://github.com/devblackops/pshealthz

PSHEALTHZ

HTTP(s) listener that executes OVF tests over REST API

https://github.com/devblackops/pshealthz

Implementation of Health Endpoint Monitoring pattern



```
{
  "overallHealth": "OK",
  "responseTimeMS": "50",
  "storageGB": 5,
  "databaseMS": 10
   ...
}
```

PSHEALTHZ

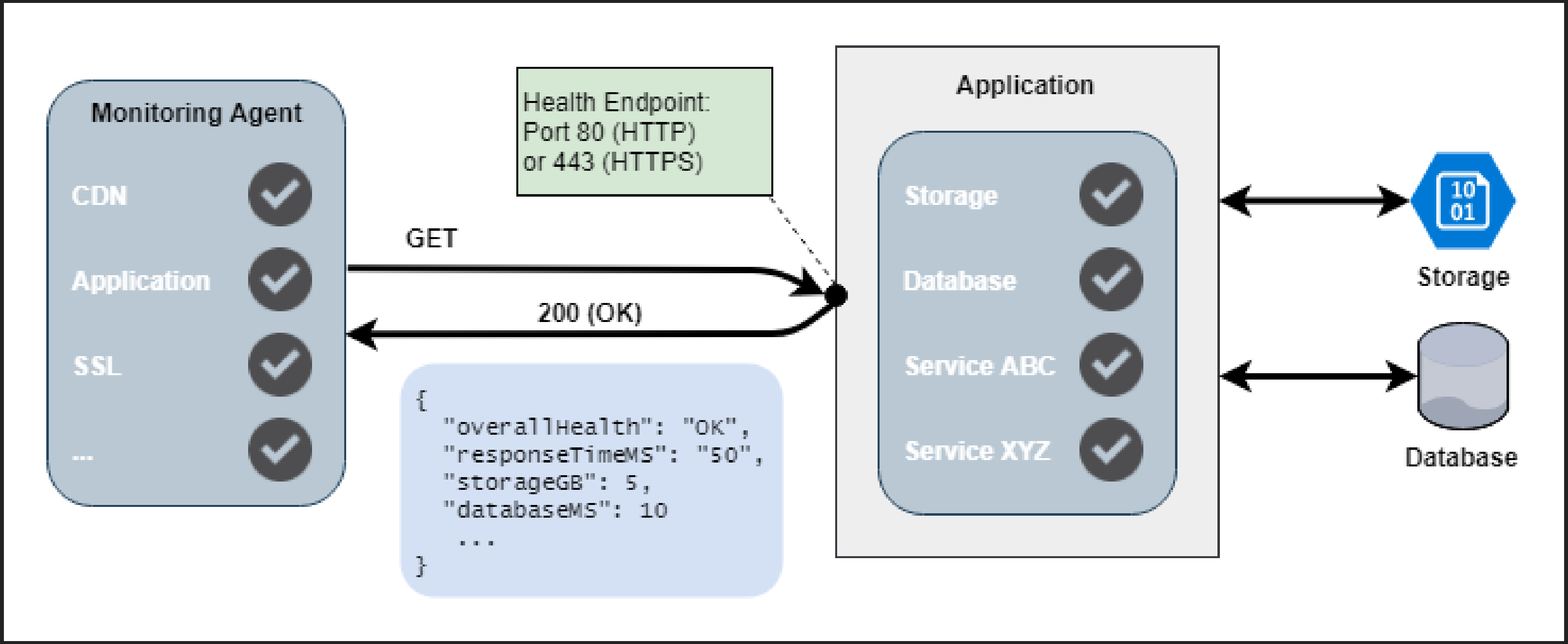
HTTP(s) listener that executes OVF tests over REST API

https://github.com/devblackops/pshealthz

Functions:

- Get-HealthZListener
- New-HealthZToken
- Start-HealthZListener
- Stop-HealthZListener

Implementation of Health Endpoint Monitoring pattern



```
{
  "overallHealth": "OK",
  "responseTimeMS": "50",
  "storageGB": 5,
  "databaseMS": 10
   ...
}
```

PSHEALTHZ

@devblackops

PSHEALTHZ

Create a new listener

```
$listener = Start-HealthZListener -Port 8080 -Path 'health' -PassThru
```

PSHEALTHZ

Create a new listener

```
$listener = Start-HealthZListener -Port 8080 -Path 'health' -PassThru
```

Get a listener

```
PS> Get-HealthZListener

Id                     : 5
State                  : Running
Uri                    : http://*:8080/health/
Port                   : 8080
Path                   : health/
Auth                   : Anonymous
SSL                    : False
CertificateThumbprint  :
Token                  :
Log                    : C:\Users\brandon\AppData\Local\Temp\PSHealthZ\
                         03e95a3a-617c-4794-a096-f135ddca284a.log
InstanceId             : 03e95a3a-617c-4794-a096-f135ddca284a
```

@devblackops

PSHEALTHZ

Get OVF tests

```
PS> $tests = Invoke-RestMethod -Uri 'http://localhost:8080/health'
PS> $tests


success         : True
time            : 2019-01-02 04:11:43Z
timeElapsedMS   : 423.1821
message         : PSHealthZ responds but does not execute tests without being told to. Add query
                  parameter '?test=<testname>' and/or '?module=<modulename>' to execute
                  specific tests. Available tests are listed in the 'availableTests' property of
                  this response. Use '?test=*' or '?module=*' to execute all available tests
                  regardless of module. Specific tests can be executed by filtering with the
                  'test' and 'module' query parameters.
availableTests : {@{name=Logical Disks; module=OVF.Windows.Server}, @{name=Memory;
                  module=OVF.Windows.Server}, @{name=Network Adapters;
                  module=OVF.Windows.Server}, @{name=Operating System; module=OVF.Windows.Server}}
testResults     : {}
failedTests     : {}
```

@devblackops

PSHEALTHZ

PSHEALTHZ

List available tests

```
PS> $tests.availableTests



name              module
----              ------
Logical Disks     OVF.Windows.Server
Memory            OVF.Windows.Server
Network Adapters  OVF.Windows.Server
Operating System  OVF.Windows.Server
```

PSHEALTHZ

List available tests

```
PS> $tests.availableTests


name              module
----              ------
Logical Disks     OVF.Windows.Server
Memory            OVF.Windows.Server
Network Adapters  OVF.Windows.Server
Operating System  OVF.Windows.Server
```

Execute a test

```
PS> $results = Invoke-RestMethod -Uri 'http://localhost:8080/health?module=ovf.windows.server'
```

PSHEALTHZ

Evaluating results

```
PS> $results

success         : False
time            : 2019-01-02 04:21:42Z
timeElapsedMS   : 1370.2033
message         :
availableTests  : {@{name=Logical Disks; module=OVF.Windows.Server},
                   @{name=Memory; module=OVF.Windows.Server},
                   @{name=Network Adapters; module=OVF.Windows.Server},
                   @{name=Operating System; module=OVF.Windows.Server}}
testResults     : {...}
failedTests     : {...}
```

@devblackops

PSHEALTHZ

See failed tests

```
PS> $results.failedTests

test     : Service property 'status' for 'WinRM' should be running
module   : C:\Program Files\PowerShell\Modules\OVF.Windows.Server\1.0.2
passed   : False
result   : Failed
describe : Operating System
context  : Availability
file     : C:\Program Files\PowerShell\Modules\OVF.Windows.Server\1.0.2\Diagnostics\Simple\
           Services.tests.ps1
message  : Expected 'running', but got Stopped.
duration : 00:00:00.0263150

...
```

WATCHMEN

Infrastructure test runner and notification system using OVF and Pester

https://github.com/devblackops/watchmen

WATCHMEN

Infrastructure test runner and notification system using OVF and Pester

https://github.com/devblackops/watchmen

Functions:

- Get-WatchmenTest
- Invoke-WatchmenTest

WATCHMEN

Infrastructure test runner and notification system using OVF and Pester

https://github.com/devblackops/watchmen

Functions:

- Get-WatchmenTest
- Invoke-WatchmenTest

Notifiers:

- Email
- EventLog
- InfluxDB
- LogFile
- PowerShell
- Slack
- Syslog

WATCHMEN

Watchmen file

```
WatchmenTest 'MyApp.OVF' {
    notifies {
        logfile 'c:\temp\watchmen.log'
        slack @{
            Token      = '<SLACK-TOKEN>'
            Channel    = '#Watchmen'
            AuthorName = $env:COMPUTERNAME
            PreText    = 'Everything is on :fire:'
            IconEmoji  = ':fire:'
        }
    }
}
```

Execute Watchmen tests

```
PS> $tests = Get-WatchmenTest -Path .\watchmen.ps1
PS> $tests | Invoke-WatchmenTest
```

WATCHMEN

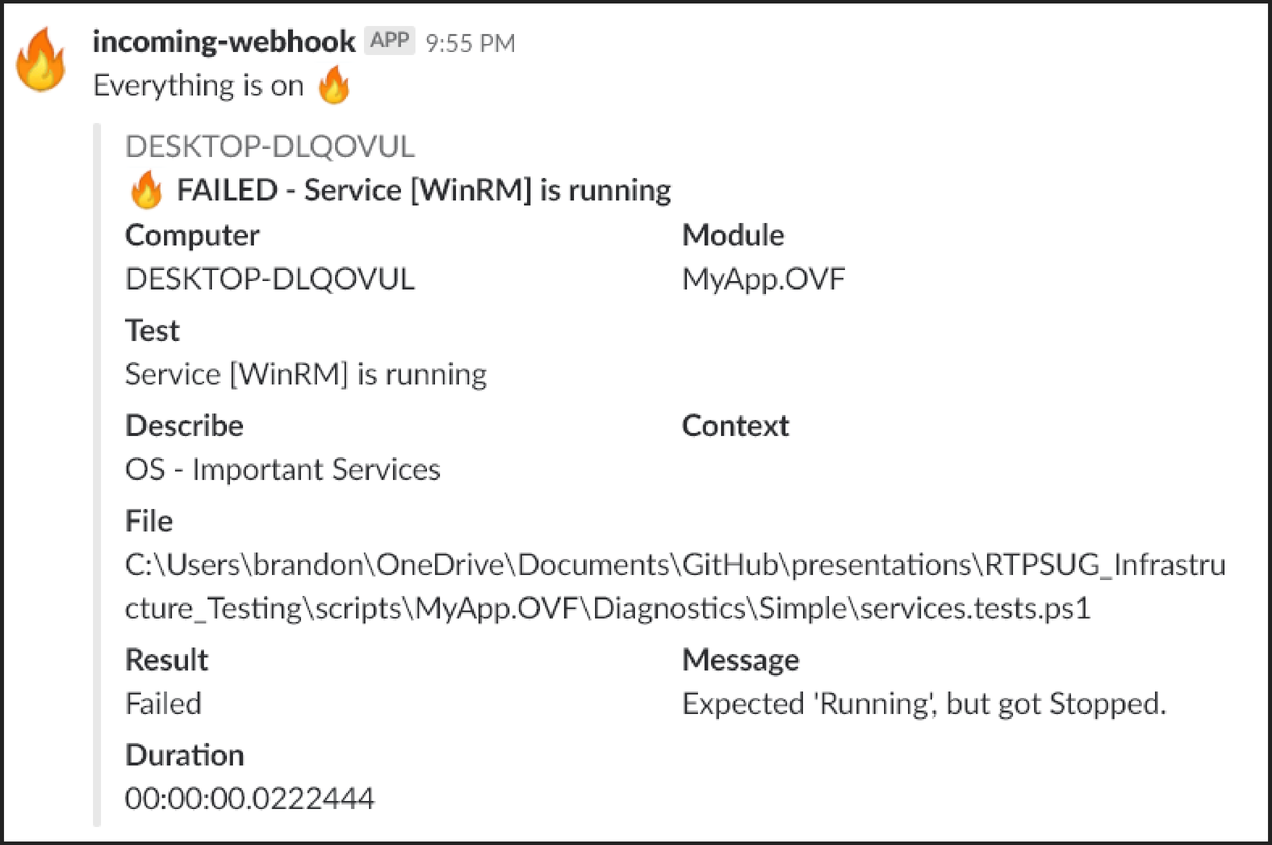
Log file result

```
PS> Get-Content C:\temp\watchmen.log
[2019-01-02 05:55:05Z] - ERROR - OS - Important Services ->  -> Service [WinRM] is running -> FAILED
```

@devblackops

WATCHMEN

Slack result



incoming-webhook APP 9:55 PM
Everything is on 🔥

DESKTOP-DLQOVUL
🔥 **FAILED - Service [WinRM] is running**

Computer
DESKTOP-DLQOVUL

Module
MyApp.OVF

Test
Service [WinRM] is running

Describe
OS - Important Services

Context

File
C:\Users\brandon\OneDrive\Documents\GitHub\presentations\RTPSUG_Infrastructure_Testing\scripts\MyApp.OVF\Diagnostics\Simple\services.tests.ps1

Result
Failed

Message
Expected 'Running', but got Stopped.

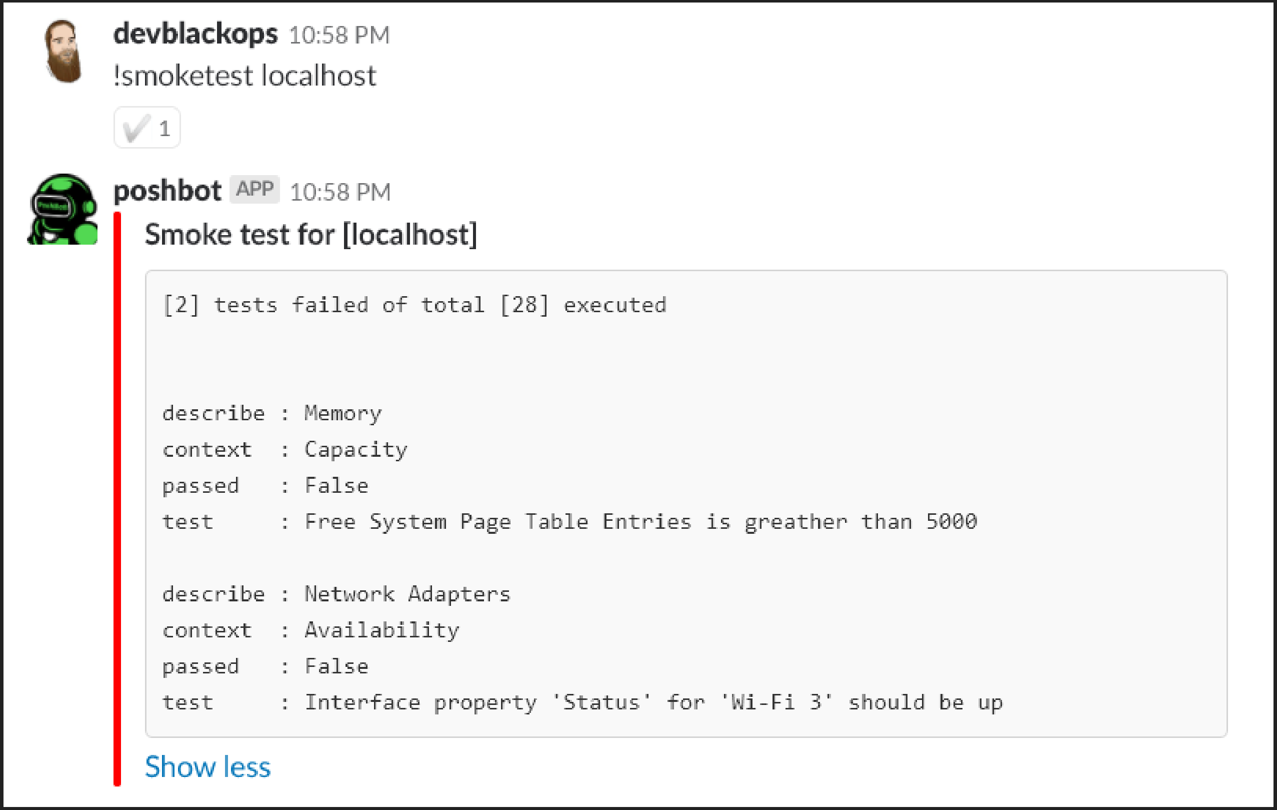
Duration
00:00:00.0222444

@devblackops

EXAMPLE USE CASES

RUN SMOKE TESTS FROM SLACK

Use PoshBot plugin to call PSHealthZ endpoint to execute OVF tests



devblackops 10:58 PM

!smoketest localhost

✓ 1

poshbot APP 10:58 PM

Smoke test for [localhost]

```
[2] tests failed of total [28] executed



describe : Memory
context  : Capacity
passed   : False
test     : Free System Page Table Entries is greather than 5000

describe : Network Adapters
context  : Availability
passed   : False
test     : Interface property 'Status' for 'Wi-Fi 3' should be up
```

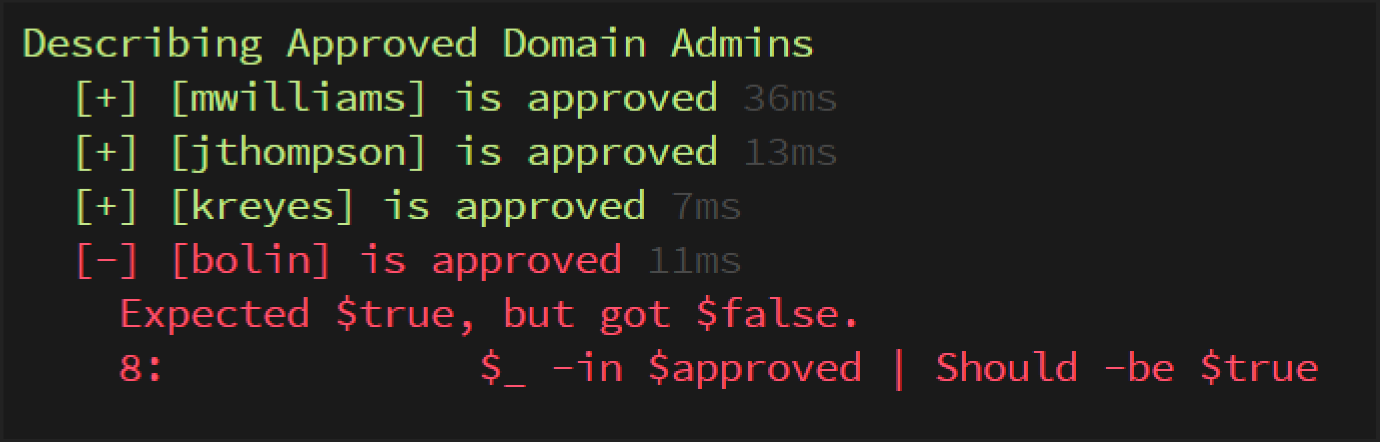
Show less

@devblackops

COMPLIANCE AUDITS

Use Pester to test Domain Admin membership against approved list

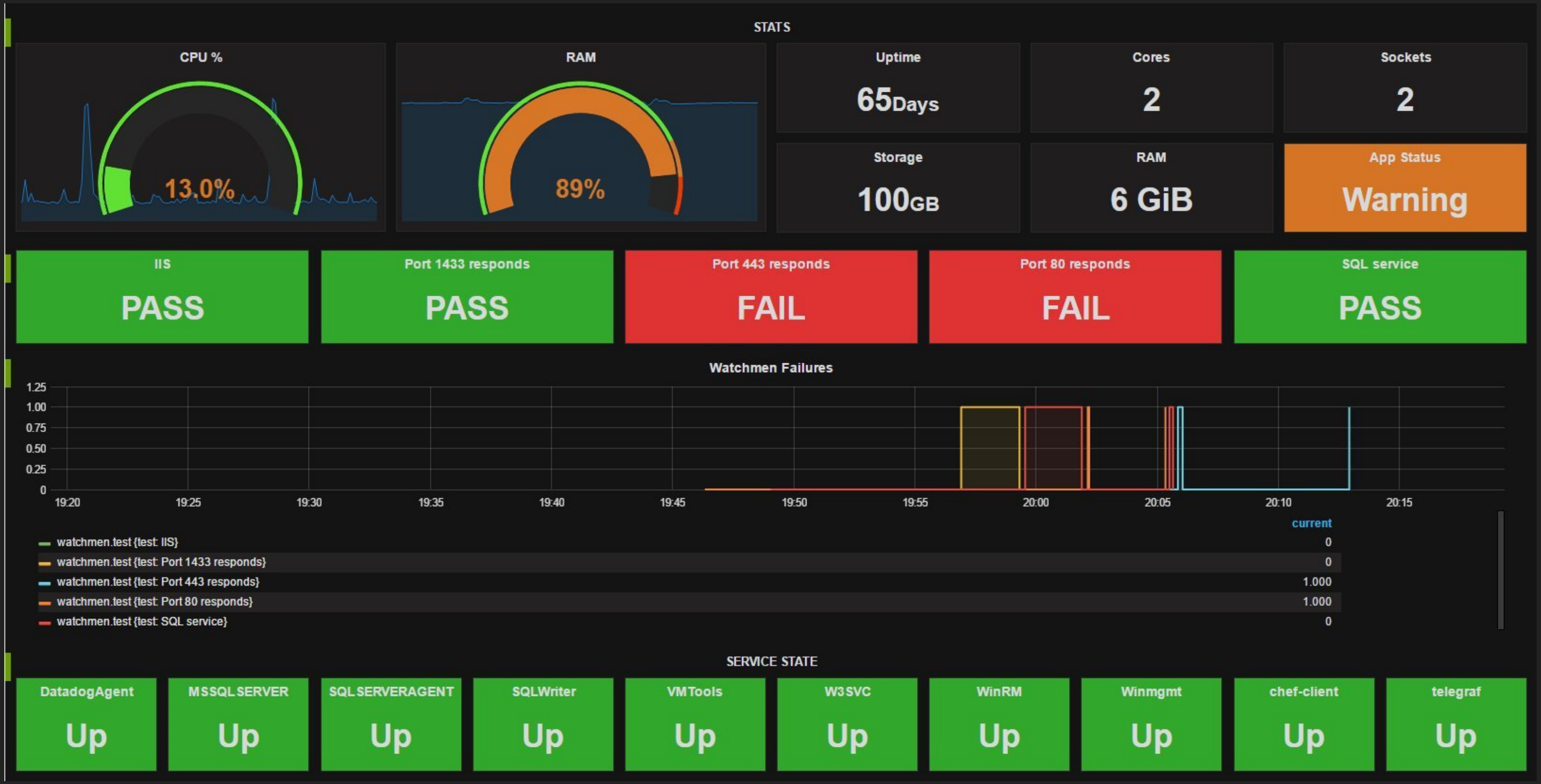
```
Describe 'Approved Domain Admins' {

    $actual   = Get-ActualDomainAdmins
    $approved = Get-ApprovedDomainAdmins

    $actual | ForEach-Object {
        It "[$_] is approved" {
            $_ -in $approved | Should -be $true
        }
    }
}
```



```
Describing Approved Domain Admins
  [+] [mwilliams] is approved 36ms
  [+] [jthompson] is approved 13ms
  [+] [kreyes] is approved 7ms
  [-] [bolin] is approved 11ms
    Expected $true, but got $false.
    8:           $_ -in $approved | Should -be $true
```

VISUALIZE TESTS AS TIME SERIES METRICS

Combine Pester test results with
system performance for dashboarding



	current
watchmen.test {test: IIS}	0
watchmen.test {test: Port 1433 responds}	0
watchmen.test {test: Port 443 responds}	1.000
watchmen.test {test: Port 80 responds}	1.000
watchmen.test {test: SQL service}	0

GOOD PRACTICES

GOOD PRACTICES

- Focus tests on what the user sees

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate
- Tests should be in version control

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate
- Tests should be in version control
- Use Configuration Management to deploy tests

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate
- Tests should be in version control
- Use Configuration Management to deploy tests
- Enable application owners to author tests

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate
- Tests should be in version control
- Use Configuration Management to deploy tests
- Enable application owners to author tests
- Automate deployment of tests into the environment

GOOD PRACTICES

- Focus tests on what the user sees
- Start small and iterate
- Tests should be in version control
- Use Configuration Management to deploy tests
- Enable application owners to author tests
- Automate deployment of tests into the environment
- **Make it easy to do the right thing**

THANK YOU!

- Brandon Olin
- @devblackops
- devblackops.io
- github.com/devblackops