

Setup Automation with PowerShell

Forging the weapon of one man's war against manual
setup checklists

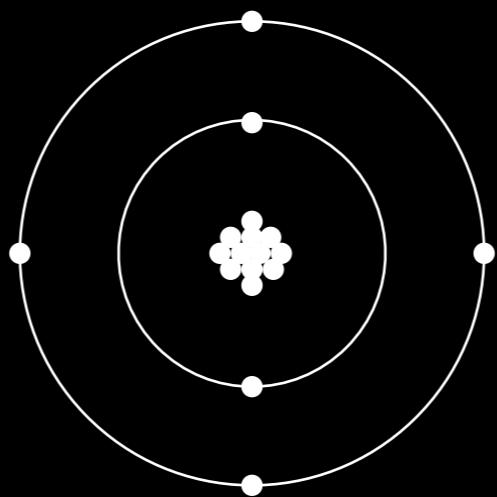
Aaron Jensen
“Software” Architect
WebMD Health Services
<http://splatteredbits.com>
<http://pshdo.com>

Thanks*

- WebMD Health Services and Chris Brooks
- Contributors: Scott Araujo and Charlie Shopsis
- Presentation Review Team: Scott Araujo, Dan Beutler, Shawn Goddard, Matt McLuckie, John Veneruso, Todd Whaley
- Open Source Bridge

* All names in reverse descending alphabetical order by last name

Carbon











*Ability to fly is symbolic and not guaranteed. Nothing will make you as awesome as Superman, unless you're Bruce Wayne.

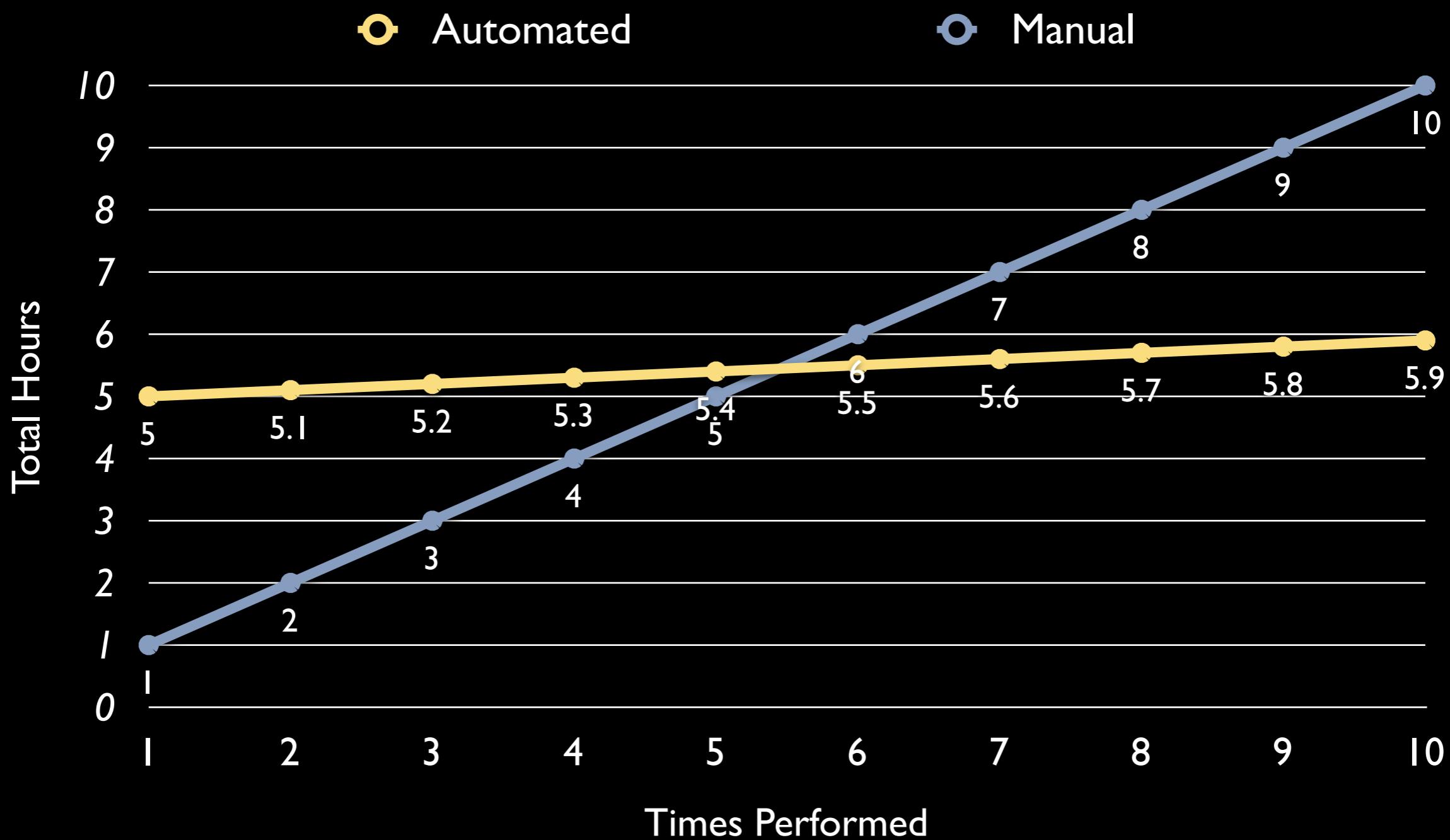
Why Carbon?



Automation

Do it once manually, shame on me. Do it twice manually, I've got no business writing software or doing anything with computers really, so fire me so I can find a job mowing lawns or something.

Automation Increases Bandwidth



Automation Increases Quality



In the beginning...



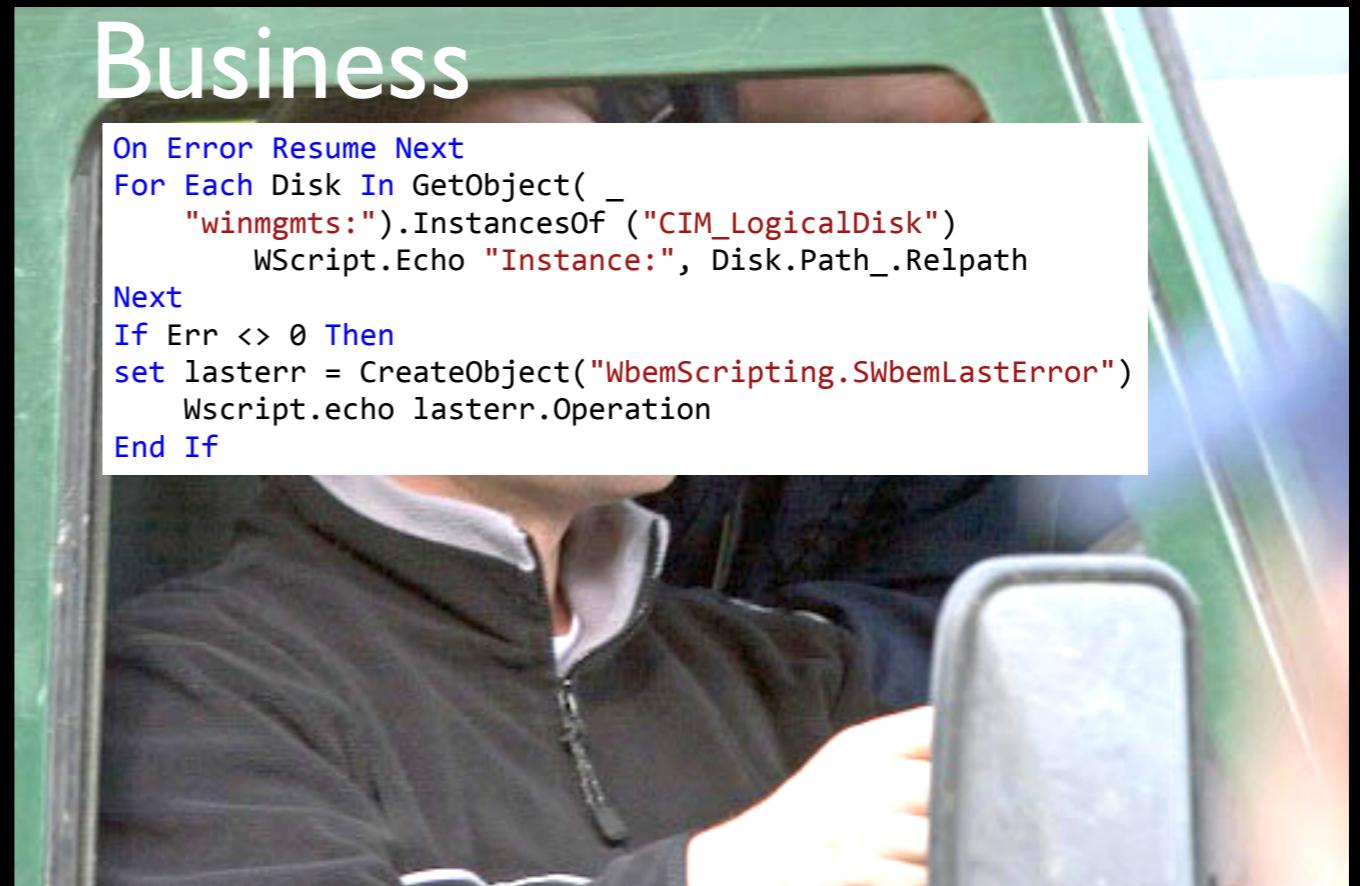
UNIX

Windows



Business

```
On Error Resume Next
For Each Disk In GetObject(
    "winmgmts:").InstancesOf ("CIM_LogicalDisk")
    WScript.Echo "Instance:", Disk.Path_.Relpath
Next
If Err <> 0 Then
    set lasterr = CreateObject("WbemScripting.SWbemLastError")
    Wscript.echo lasterr.Operation
End If
```



Windows

Server/.NET

```
<target name="build">
  <exec program="${MSBuildPath}">
    <arg line='${SolutionFile}' />
    <arg line="/property:Configuration=${SolutionConfiguration}" />
    <arg value="/target:Rebuild" />
    <arg value="/verbosity:normal" />
    <arg value="/nologo" />
    <arg line='/logger:"C:\Program Files\CruiseControl.NET\server\ThoughtWorks.CruiseControl.MsBuild.dll"' />
  </exec>
</target>
```

```
<Target Name="AfterBuild" Condition="$(Configuration) == 'Release' ">
  <CreateItem Condition="'%(Extension)'=='.dll'" Include="@{ReferenceCopyLocalPaths}">
    <Output TaskParameter="Include" ItemName="IlmergeAssemblies" />
  </CreateItem>
  <Exec Command="$(ProjectDir)..\\..\\..\\tools\\IlMerge\\Ilmerge.exe" /targetplatform:"v4,C:\\Windows\\Micro\\Framework64\\v4.0.30319" /ndebug /out:@(MainAssembly) "&quot;@{IntermediateAssembly}&quot; @{IlmergeAssemblies->'&q(FullPath)&quot;,' ')" />
    <Delete Files="@{ReferenceCopyLocalPaths->'$(OutDir)%$(DestinationSubDirectory)%$(Filename)$(Extension)'")" />
  </Target>
```

NAnt/MSBuild



PowerShell

```
<#
.SYNOPSIS
Packages and publishes Carbon packages.
#>

# Copyright 2012 Aaron Jensen
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

[CmdletBinding()]
param(
)

Set-StrictMode -Version Latest
$ErrorActionPreference = 'Stop'
$PSScriptRoot = Split-Path -Parent $MyInvocation.MyCommand.Definition

$licenseFileName = 'LICENSE.txt'

$releaseNotesFileName = 'RELEASE NOTES.txt'
$releaseNotesPath = Join-Path $PSScriptRoot $releaseNotesFileName -Resolve
$releaseNotes = Get-Content $releaseNotesPath

if( $releaseNotes[0] -notmatch "^#\s+(\d+\.\d+\.\d+)\s*" )
{
    Write-Error "Missing version from release notes file. The first line must contain the version about to be released."
    exit
}
$version = [Version]$matches[1]
$releaseNotes[0] = "# $version ($((Get-Date).ToString("d MMMM yyyy")))"
$releaseNotes | Out-File -FilePath $releaseNotesPath -Encoding OEM

Copy-Item (Join-Path $PSScriptRoot LICENSE.txt -Resolve) (Join-Path $PSScriptRoot Carbon\ -Resolve)

$carbonZipFileName = "Carbon-$version.zip"
$zipAppPath = Join-Path $PSScriptRoot Tools\7-Zip\7za.exe -Resolve

Push-Location $PSScriptRoot
tar
```

Layers of Abstraction

Carbon

PowerShell

.NET

appcmd.exe
msiexec.exe
net.exe
netsh.exe
sc.exe
servermanagercmd.exe
ocsetup.exe
...

Windows/Win32 APIs

Carbon

Active Directory

MSMQ

Certificates

Performance Counters

Computer

PowerShell

Cryptography

Security

.NET

Shares

File System

Users and Groups

Hosts File

Windows Features

IIS

Windows Firewall

INI Files

Windows Services

Mercurial

Carbon

ACTIVE DIRECTORY

- Get-ADDomainController
- Get-ADUser
- Format-ADSpecialCharacters

CERTIFICATES

- Get-Certificate
- Get-CertificateStore
- Get-SslCertificateBinding
- Get-SslCertificateBindings
- Install-Certificate
- Remove-Certificate
- Remove-SslCertificateBinding
- Set-SslCertificateBinding
- Test-SslCertificateBinding

COMPUTER

- Disable-IEEnhancedSecurityConfiguration
- Enable-IEActivationPermissions
- Invoke-WindowsInstaller
- Remove-EnvironmentVariable
- Set-EnvironmentVariable
- Test-OSIs32Bit
- Test-OSIs64Bit

CRYPTOGRAPHY

- Protect-String
- Unprotect-String

.NET

- Set-DotNetAppSetting
- Set-DotNetConnectionString

FILE SYSTEM

- Get-FullPath
- Get-PathCanonicalCase
- Get-PathRelativeTo
- New-Junction
- New-TempDir

- Remove-Junction
- Test-PathIsJunction

HOSTS FILE

- Get-PathToHostsFile
- Reset-HostsFile
- Set-HostsEntry

IIS

- Add-IisDefaultDocument
- Get-IisHttpRedirect
- Get-IisVersion
- Get-IisWebsite
- Install-IisApplication
- Install-IisAppPool
- Install-IisVirtualDirectory
- Install-IisWebsite
- Invoke-AppCmd
- Remove-IisWebsite
- Set-IisAnonymousAuthentication
- Set-IisBasicAuthentication
- Set-IisDirectoryBrowsing
- Set-IisHttpRedirect
- Set-IisSslFlags
- Set-IisWebsiteSslCertificate
- Set-IisWindowsAuthentication
- Test-IisAppPoolExists
- Unlock-IisBasicAuthentication

INI FILES

- Set-IniEntry
- Split-Ini

MERCURIAL

- Set-HgDefaultPushUrl

MSMQ

- Get-MsmqMessageQueue
- Get-MsmqMessageQueuePath
- Grant-MsmqMessageQueuePermissions

- Install-Msmq
- Install-MsmqMessageQueue
- Remove-MsmqMessageQueue
- Test-MsmqMessageQueue

PERFORMANCE COUNTERS

- Get-PerformanceCounters
- Install-PerformanceCounter
- Test-PerformanceCounter
- Test-PerformanceCounterCategory
- Uninstall-PerformanceCounter

POWERSHELL

- Complete-Jobs
- Get-PowerShellPath
- Invoke-PowerShell
- Test-PowerShellIis32Bit
- Test-PowerShellIis64Bit

POWERSHELL REMOTING

- Add-TrustedHosts
- Get-TrustedHosts
- Set-TrustedHosts

SECURITY

- Assert-AdminPrivileges
- Convert-SecureStringToString
- Grant-Permissions
- New-Credential
- Test-AdminPrivileges

SERVICES

- Grant-ServiceControlPermission
- Install-Service
- Invoke-SubInAcl
- Remove-Service
- Restart-RemoteService

SHARES

- Install-Share

USERS AND GROUPS

- Add-GroupMembers
- Get-WmiLocalUserAccount
- Install-Group
- Install-User
- Remove-User
- Test-User

WINDOWS FEATURES

- Install-WindowsFeatureIis
- Install-WindowsFeatureMsmq
- Install-WindowsFeatures
- Test-WindowsFeature
- Uninstall-WindowsFeature

WINDOWS FIREWALL

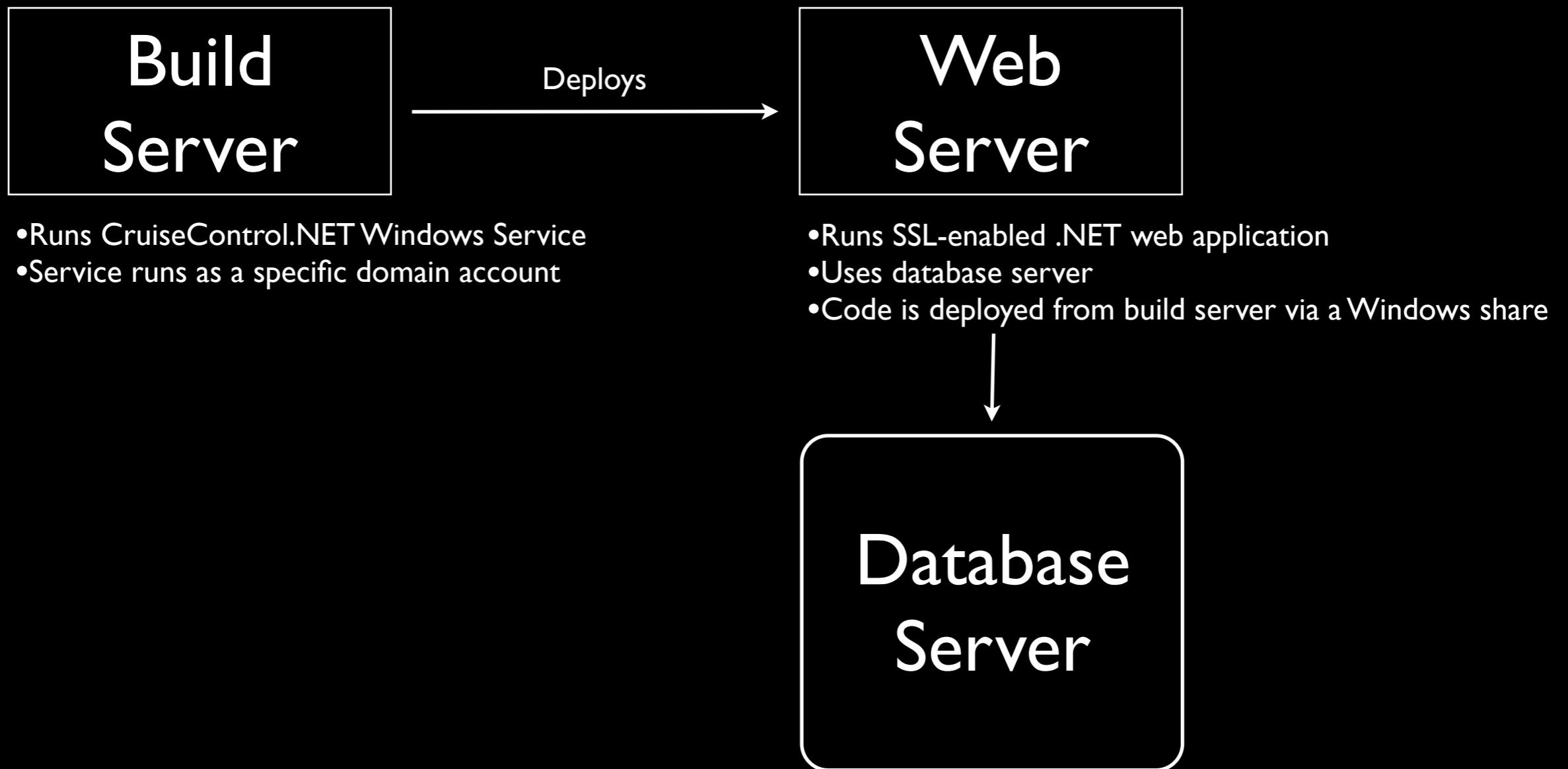
- Assert-FirewallConfigurable
- Disable-FirewallStatefulFtp
- Enable-FirewallStatefulFtp
- Get-FirewallRules
- Test-FirewallStatefulFtp

TYPES

- Diagnostics.Process.ParentProcessID
- ServiceProcess.ServiceController.StartMode
- ServiceProcess.ServiceController.UserName
- IO.DirectoryInfo.IsJunction
- IO.DirectoryInfo.TargetPath

Why Carbon?

faux Environment



Code

<meta />

- Automated tests
- Continuous Integration
- Documentation
- Semantic Versioning – semver.org

Semantic Versioning

Major.Minor.Patch, e.g. 1.8.3

Get Involved @ Hacker Lounge

More Information

<http://get-carbon.org>

twitter
@pshdo.com
http

email [aaron@splatteredbits.com]
http

Q&A