



2018

PowerShell Core and Win32-OpenSSH



Joey Aiello, Program Manager
@joeyaiello on  

PowerShell Core 6.0 GA

- Manage your heterogenous environments in the hybrid cloud
 - "Anything, anywhere"
- PowerShell Core 6.0 released on January 10
- Thank you to our contributors!
 - ~50% of pull requests came from outside Microsoft
 - Track our metrics at <https://aka.ms/PSGitHubBI>
- Big features
 - Cross-platform: Windows, macOS, Linux
 - Side-by-side and portable
 - SSH-based PowerShell Remoting
 - [And more!](#)



Coverage for PowerShell Core

- The success of PowerShell Core depends on you being able to retain your skillset
- Working with Azure, Windows, and Office to increase coverage
- [AzureRM.NetCore](#) making progress on porting Azure PowerShell
- [WindowsPowerShellCompatibilityPack](#)
 - Transparent implicit remoting to local Windows PowerShell
 - Will supplant [WindowsPSModulePath](#)
 - Manage legacy aliases
- What are the modules you want/need?



Why are we optimistic?

- Clear transition path from Windows PowerShell -> PowerShell Core
 - PowerShell Standard for .NET
 - CDXML and PowerShell work everywhere
- The goal is porting and compatibility, not reimplementation
- Many of the “hard problems” are reoccurring
 - E.g. 2FA with Azure PowerShell
- Value add of PowerShell Core is easy to communicate
 - Cross-platform
 - Performant
 - Portable



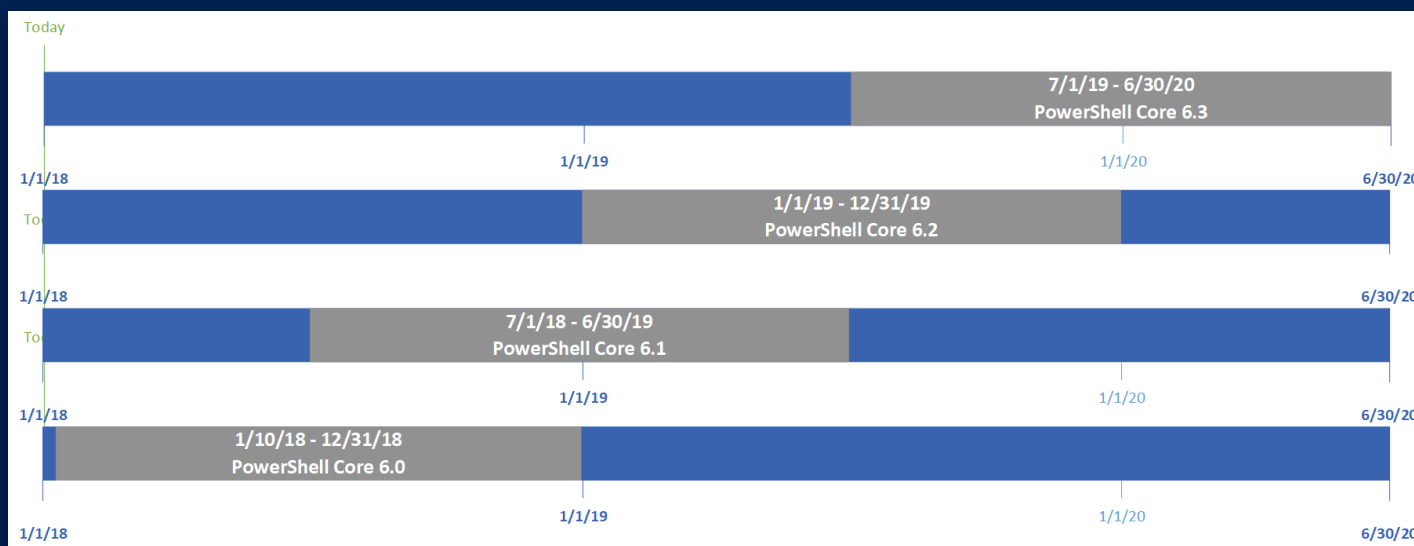
Remoting on PowerShell Core

- Three ways to remote:
 - PowerShell Remoting over WSMAN/WinRM
 - PowerShell Remoting over SSH
 - Plaintext SSH remoting
- Limitations
 - WSMAN PSRP server is experimental on non-Windows platforms
 - WSMAN PSRP client doesn't support Kerberos on non-Windows platforms
- Our hypothesis
 - SSH is the future and everyone should use it
 - WSMAN/WinRM is still very important and we should continue to support it



Support Lifecycle

- PowerShell Core supported under Modern Lifecycle Policy
 - 6 months to move to latest minor version
- Visit <https://aka.ms/pslifecycle> for up-to-date support info
- Don't think too hard about the version number
- Fix-forward is good for module developers



Limitations of PowerShell Core

- Some modules are incompatible with .NET Core
- A few “built-in” cmdlets are missing from PowerShell Core
 - WMI v1 cmdlets, PerfCounter, EventLog, LocalAccounts
 - On non-Windows platforms, these modules are missing:
 - CimCmdlets
 - Microsoft.PowerShell.Diagnostics
 - Microsoft.WSMan.Management
 - PSDiagnostics
- Removed snap-ins and workflows

What about Windows PowerShell?

- Still fully supported and serviced
- Will not be “replaced” by PowerShell Core within Windows
 - Remaining a stable platform for existing workloads
- No new feature innovation planned
 - Includes PSRP over SSH
- Label to track backporting is [Consider-WindowsPowerShell51](#)

PowerShell Standard

- “Universal” API between Windows PowerShell and PowerShell Core
- Version number snaps to minimum compatible version

PowerShell Standard	Windows PowerShell	PowerShell Core
3.0	3.0 – 5.1	6.x
5.1	5.1	6.x

- Keep your eyes peeled for a blog!

PowerShell Standard with Posh-SSH

PowerShell Core 6.1

- GA release planned for late June, early July
- <https://github.com/PowerShell/PowerShell-RFC>
- Check out Steve Lee's talk on PowerShell Core 6.1
 - Thursday, April 19 at 14:55

Room for improvement

- Committee transparency/openness
- RFC responsiveness
- Breaking changes
- Reference documentation

Call to Action

- Use PowerShell Core on non-Windows platforms
- [File issues](#)! Contribute [code](#) and [documentation](#)!
- Join the [PowerShell Community Call](#) (3rd Thursday of every month)
- Port your PowerShell modules to [PowerShell Standard](#)
 - File issues
 - Upload them to the [PowerShell Gallery](#)
- Give feedback to product teams that own PowerShell modules



OpenSSH

- OpenSSH is a suite of tools:
 - public / private key management
 - remote authentication / administration
 - secure file transfers
- Thanks to the OpenBSD Foundation and community for creating and supporting OpenSSH and LibreSSL!
- Includes:
 - LibreSSL (libcrypto.dll)
 - scp.exe
 - sftp.exe
 - sftp-server.exe
 - ssh.exe
 - ssh-add.exe
 - ssh-agent.exe (service)
 - sshd.exe (service)
 - ssh-keygen.exe
 - ssh-keyscan.exe
 - ssh-shellhost.exe

Win32-OpenSSH

- Native Windows port of OpenSSH and LibreSSL
- Client and server each shipped as a beta “feature-on-demand” in Fall Creators Update
 - Client will soon be on by default
 - Server is still optional
 - Both will be production ready
- No plans yet to officially support GitHub builds on downlevel platforms
 - What are your requirements here?

What's next for Win32-OpenSSH?

- Get upstream with OpenSSH Portable
- Investigate
 - filling out longer tail of [unsupported features](#)
 - installer/servicing model for downlevel platforms
- New features?
 - What do folks need on Windows?

PSRP over SSH with 2FA

Survey @
<https://aka.ms/PSSurvey>

Questions?

Next Steps

- Now: 15 min break
- Grab a coffee
- Stay here to enjoy next presentation
- Change track and switch to another room
- Ask me questions or meet me in a breakout session room afterwards

