



WHITEPAPER

Microsoft's DevOps
Toolchain

DEVOPS AND MICROSOFT

Most people associate DevOps with open source platforms and applications and with good reason. In the forward for the book *Continuous Delivery with Windows and .NET*, Dave Farley, who literally wrote the book on continuous delivery, writes, "I think it fair to say that some of the initial innovation in the Continuous Delivery space came from the Open Stack community." But Microsoft has been pushing itself as a viable option for continuous workflows, offering its Azure cloud platform and its Visual Studio Online products as alternatives to Linux-based solutions.

Farley goes on in that same forward: "It has never been as simple as Open Stack has the best tools and Windows plays catch-up... over time, both worlds have developed their own collections of technology to support Continuous Delivery and its associated practices." In addition to announcements that Microsoft is going to support Docker containers on Azure, they are also working on their own containerization option called Drawbridge. Software like Puppet also supports Windows management, although a Windows server cannot act as the Puppet Master, and there is also Microsoft's own configuration automation solution, SCCM. DevOps in the Microsoft space is real.

In 2014, Microsoft open sourced the .NET development stack, making it cross platform capable and more accessible to various communities. Just this month, they announced their flagship database application, Microsoft SQL Server, would be available on Linux for the first time. It's clear Microsoft realizes how important DevOps and related practices are becoming to IT departments and that technologies that don't fit into these improved workflows will be left behind.

MICROSOFT AND POWERSHELL

One of the most important advances Microsoft has made in this regard is the continuing expansion and integration of PowerShell into administration tasks. PowerShell gives people the same kind of control and automation capability *nix systems have had for some time, and with the option to get PowerShell code out of the GUI when performing tasks in Microsoft products, traditional sysadmins have a path to learn this language and how to incorporate it into their operations.

But even with all of this progress, can Microsoft tools perform as well as their open source counterparts?

	Common Tools	Microsoft
Containerization	Docker/Vagrant	Drawbridge
Cloud	AWS/Google	Azure
Development	Various	.NET (Visual Studio Code)
Config Automation	Puppet/Chef	SCCM and PowerShell
Versioning	Git	Team Foundation Server
Collaboration	Slack/Asana/Google Apps	Visual Studio Team Services
Database	MySQL, MongoDB	Microsoft SQL Server
Web Server	Apache, nginx	IIS

THE TOOLS

The tools are there. How they are used depends on the implementation and the environment. Because most DevOps-inclined people come down on one side or the other of the Microsoft vs. everything else debate and many IT shops are already tied into one technology or the other, both solutions are seeing use in businesses today with similar results. But the very fact that one has to “make a case” for Microsoft in the DevOps sphere puts them at a disadvantage, especially competing against major open source options with large community bases and proven performance.

Many of these third party tools work (to some degree) with Microsoft, as you might expect, given that many businesses run a mixed-platform environment. However, Microsoft nonetheless seeks to provide an entire DevOps toolchain that would make end-to-end continuous deployment possible solely with Microsoft products. In practice, however, it's far more likely that multiple technologies will be used and leveraged together (or against one another, depending on how well they are implemented) to accomplish the business goals of the company. This leads to picking and choosing Microsoft products based on each one's specific functionality and the advantages and disadvantages it has, rather than just buying into the entire “Microsoft way.”

Microsoft does seem to walk the walk, however, implementing DevOps and Agile practices in its own software delivery pipeline and publishing case studies of how their own teams have used DevOps culture to improve their output and reliability. Many people say “DevOps isn't about tools” and this is one case where Microsoft and anti-Microsoft people can agree: DevOps begins with a culture change, a commitment to doing business differently. Providing opportunities for businesses with large Microsoft deployments to utilize DevOps methodologies will help keep the overall culture change in IT rolling forward and not leave Windows sysadmins and .NET developers in traditional--what are becoming legacy-- environments.

CONCLUSION

There's no best (or single) solution to deployment pipelines or DevOps culture. The technology changes with the field and innovation drives best practices. Microsoft should not be counted out of the DevOps space or relegated to a second-tier when it comes to tools, but neither does it outmatch the third party software doing the same thing. Businesses who are already established with a Linux-based deployment process have little incentive to move to Microsoft, but they may have cause to include some Microsoft technology in addition, such as utilizing both Azure and other cloud vendors for resiliency, managing existing Windows systems with SCCM, or using Microsoft's new SQL Server on Linux as an alternative for MySQL.

Companies looking to enter the DevOps sphere and move to a continuous delivery model will have to first understand their current environment, what its strengths and weaknesses are, which platforms or applications drive the business, and can then make informed decisions on what tools will best serve them. Microsoft isn't necessarily the first option people think of when they consider DevOps, but it is an option, especially for companies who already rely heavily on Microsoft technologies. Moving forward, expect the gap between Microsoft and other tools to close further, as they continue pressing their business in this direction. But the dividing line between pro- and anti-Microsoft people and ideologies isn't likely to disappear any time soon, so prepare for more innovation from open source projects as well.