

October 23, 2016

Windows NanoServer – Welcome to the Mini-OS

Earlier this month at the Ignite conference, Microsoft rolled out its latest and greatest version of its Server Operating System, Server 2016. There were a lot of really innovative and really cool features that came along with the release, and one of them was the introduction of NanoServer. This bite-sized code promises not only to change the way we deliver the Windows operating system itself, but also change the way we think about how we create, deploy, and manage applications that run on it.

First, a bit of history to frame it all up. NanoServer was originally created by Microsoft to perform two functions for their Azure cloud. One was to have a very small footprint of the Windows operating system to run applications that were built for the cloud. These applications were later termed “Born-in-the-cloud” applications. The second was to accommodate their venture into the next generation of DevOps and beyond with the Microsoft Container technology. Linux containers were already in wide use in Azure and developers and their operations staff wanted them for Windows as well. They already had the Server Core offering, but it still wasn’t fast enough, provide high density factors using very little resources, as well as a very small attack surface overall. They also took a slightly different approach on the code itself, deciding to refactor the existing Windows code rather than spend an enormous amount of time creating something new or rewriting what they had.

What was released, was a slimmed down, modular, secure, and very fast version of the Microsoft operating system. The latest release of NanoServer is estimated to be around 20 times the size of Server Core, with almost a 95 percent reduction in overall virtual machine file size. The attack surface has been reduced to requiring less than an estimated 20 percent of the total reboots required by the full OS per year and 90 percent fewer security bulletins. The cold boot time of NanoServer can be as little as 5 seconds in a virtual machine, and 30 seconds on bare metal hardware minus the BIOS and drivers. And the density factor for NanoServer running as a Hyper-V host is off the charts with estimates that it could host up to 1,000 NanoServer guest machines within one terabyte of RAM. That is just not feasible with any other flavor of the OS.

What about the management of NanoServer? Since the kernel is very small and secure, there is no GUI built-in to the code. When NanoServer boots, you are presented with what is called the “Recovery Console”. This text-based console allows you to change just a few very distinct and critical things in the OS. They include changing the firewall rules, enabling WinRM, and IPv4 and IPv6 networking. Outside of that console, you do have an option for a text based “Out-of-Band” serial console called the Emergency Management Service. But, for most day to day management, of which you will have to do very little with NanoServer, you will most likely use PowerShell or any of the Windows Administrative tools that are available to manage on any Windows Server OS machines. These include tools like Event Viewer, Services, Server Manager, and the like. One amazing new offering from Microsoft in Azure is called the [Remote Server Management Tools](#). This allows you do some really great things with any Windows Server, including NanoServer, from the cloud. You can run a RegEdit session, collect performance data, and launch a remote PowerShell session right from the cloud. How cool is that!

Let's talk about how to create a NanoServer image. As mentioned, NanoServer can run as a Hyper-V host as well as a Hyper-V guest. It can also be run right from bare metal with the appropriate drivers loaded. Now, NanoServer is not a downloadable ISO as the other flavors are. You must create a NanoServer image from the full Windows Server 2016 bits. There are a few ways to create your image, and the [TechNet documentation here](#) shows you how, step-by-step. It talks about how to do it with PowerShell, using the "NanoServerImageGenerator" module supplied in the Server 2016 ISO. But, for those that are still not PowerShell savvy, there is also now a GUI that will guide you every step of the way, and it is called the [NanoServer Image Builder](#). Either way, after just a few minutes, you will have an image that is ready to be deployed and have you up and running with everything you need.

Now, NanoServer is meant to run a very limited amount of applications and services. When you create your NanoServer image, you can specify these to be included in the final image by adding what are called Packages. Packages can contain things such as DNS, IIS, DHCP, and more. You could also add those packages at any time after the image is built by running a simple PowerShell command. You can also add drivers, change firewall rules, and much more. The list of official Microsoft packages is small for now, but it will grow over time. You can also install applications and services that are not packages, such as the [OpenSSH Server](#), [MySQL](#), the [Systemal Suite](#) (and a [GitHub script](#) to auto install it), simply by copying the files into the NanoServer and running the commands necessary to install and run them. There are many, many others that are available, just head over to [GitHub](#) and search!

And in case you thought I forgot about running Microsoft containers and NanoServer, in my next writing installment, I'll be covering that in-depth and it will include all you can do with these awesome new innovations from Microsoft.