

Robertas Lazauskas

Nano Server and Containers in Windows Server 2016

Bachelor's thesis
Information Technology

2018



South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree	Time
Robertas Lazauskas	Bachelor of Engineering	April 2018
Thesis title		
Nano Server and Containers in Windows Server 2016		45 pages 1 page of appendices
Commissioned by		
Supervisor		
Matti Juutilainen		
Abstract		
<p>The purpose of this project was to review the new Windows Server 2016 features and test out Nano Server and Containers in the Windows Server 2016 environment. Only small part of people know what Containers and Nano Server are. Thus, this theses introduce this new features in more details to everyone.</p> <p>Windows Server 2016 has a lot of new impacts in various of fields. These impacts consist of compute, administration, identity and access, networking, storage, security and assurance and failover clustering. These impacts were analyzed, and based on my own decision I decided to test Nano Server and Containers in practice.</p> <p>The operating system and working environment were given by the supervisor. Server was created in a cluster. Based on the project theme, all configurations were made on Windows Server 2016. Container management was made using the Docker program providing containerization.</p> <p>The result of the thesis was a fully working Nano Server in Hyper-V that can do all the configurations. There is a web service that can be implemented, if needed. Later web services in a Container using nginx and docker-compose was made. Finally, fully running and working Nano Server in a container was implemented.</p>		
Keywords		
Windows Server 2016, new features, NanoServer, Containers		

CONTENTS

1	INTRODUCTION	4
2	THEORETICAL PART	5
2.1	General Review on Windows Server 2016	5
2.2	Short Review of Windows Server 2016 New Impacts	5
2.2.1	Compute	5
2.2.2	Administration	6
2.2.3	Identity and Access.....	7
2.2.4	Networking.....	8
2.2.5	Storage	9
2.2.6	Security and Assurance	11
2.2.7	Failover Clustering.....	13
3	NANO SERVER AND CONTAINERS.....	13
3.1	Nano Server.....	14
3.2	General Information about Containers.....	17
3.3	Windows Server Containers Versus Hyper-V Containers.....	20
4	PRACTICAL PART	24
4.1	Nano Server Installation	24
4.2	Basic Implementations with Containers	32
5	CONCLUSION.....	38
	REFERENCES	40

APPENDICES

Appendix 1. app.py

1 INTRODUCTION

The thesis subject I selected is about the new features in Windows Server 2016. Before starting with this topic, Windows Server 2016 was not tested on my own so it totally was a new thing for me. Anyway, I have some experience with the previous versions of Windows Server. In my university studies I have made a lot of labs to get acquainted with many of the Windows Server features. It gave me useful knowledge which can be used in this thesis.

Windows Server 2016 is the newest Microsoft release that has a lot of new impacts. The aim of the project is to introduce the reader with the new features and check the in advantages or disadvantages. All the Windows Server 2016 impacts can be divided to fields such as: compute, administration, identity and access, networking, storage, security and assurance and failover clustering. The main task is to review these new impacts and test some of them in more detail.

The aim of the practical part is to test Nano Server and Containers and to check how and what can be implemented by using them. It is also important to choose which of the containers software I am going to make. Additionally, the thesis describes how to install Nano server and how to run a container and if it is possible to combine Nano Server and Containers.

Lastly, I will summarise the theoretical and practical parts. It will maintain a short review of the study that was made about Windows Server 2016. In addition, there will be described achieved the thesis goals. It will be achieved by working with Windows Server 2016 and creating Nano Server and Containers in it.

2 THEORETICAL PART

This theoretical part introduces a general review of the newest Windows Server version release. Additionally, I will check some of the new features that improved Windows Server. The goal of this part is to make people familiar with Windows Server 2016.

2.1 General Review on Windows Server 2016

As all of us probably know, Windows servers are one of the most reliable and well-known operating systems in the world. Talking about the very beginning of the Microsoft project, it was released in 1993, and it has improved a lot since the first product. It had a user manager, the disk administrator, the performance monitor, the event viewer, backup applications and more. Now, in the newest version of Windows Servers we have so many features that writing them in is beyond the scope of this time.

Microsoft has developed the Windows Server 2016 operating system as a part of the Windows NT (version 10.0) family. It was developed contemporaneously with Windows 10. This new version was released in the late 2016 and is now continuously updated.

2.2 Short Review of Windows Server 2016 New Impacts

Windows Server 2016 came with a lot of new features and improvements. They have made new improvements on compute, identity and access, administration, networking, security and assurance, storage and failover clustering fields. The most important impacts and changes will be reviewed below.

2.2.1 Compute

Hyper-V has some improvements services. Alternate Credentials support in the terming solutions. Now it is possible to use a different set of crediantials in Hyper-V Manager, which is helpful when the user connects to another Windows Server

2016 server. Another new feature is to manage earlier versions version. It is possible to manage computers that run Hyper-V on Windows Servers 2016.

Updated management protocol improve the Remote Hyper-V hosts. It can now communicate with Hyper-V Manager using the WS-MAN (Web Services-Management) protocol which allows CredSSP, Kerberos or NTLM (NT LAN Manager) authentication. The WS-Man protocol now makes it easier to enable a host for remote management. While users are connected using CredSSP (Credential Security Support Provider), it is possible to do the live migration without any of administration in Active Directory. (Techgenix 2018.)

Windows containers are now available on Windows Server 2016. This feature permits the operating system level virtualization. This multiple isolated applications can be run on a single system.

One of the meaningful innovations is Nano Server. It is deployed as a separate installation option in Windows Server 2016. Nano Server is a lightweight Operating System that is adapted to be used as an OS layer for virtualized container instances.

2.2.2 Administration

As we all know Powershell is a task configuration and automation management framework from Microsoft which consist of a command-line shell and is associated with scripting language. Windows PowerShell 5.1 includes important new features, including support for new security features and devoliping with classes that improve its usability, extend its use, and allow user to control and manage Windows-based environments simpler and comprehensively. (Microsoft 2018.)

This new PowerShell version 5.1 is available in different editions which comes with platform compatibility and varying feature sets. This reference used strandly mentioned by Microsoft (2018).

- Core Edition: Built on .NET Core and provides compatibility with scripts and modules targeting versions of PowerShell running on reduced footprint editions of Windows such as Windows IoT and Nano Server.
- Desktop Edition: Built on .NET Framework and provides compatibility with modules and scripts targeting versions of PowerShell which is running on full footprint editions of Windows such as Windows Desktop and Server Core.

This new Windows Server version has a modern Package Management (OneGet) feature that can enable DevOps or professionals to automate software installation and discovery locally or remotely.

2.2.3 Identity and Access

This section has new features in identity improvement which has the ability to organizations to secure Active Directory (AD) environments. It also helps them to migrate it to the cloud-only deployments and hybrid deployments, where some applications and services are hosted in the cloud and others are hosted on premises.

Active Directory Certificate Services has new improvements. AD CS in new WinServer release TMK (Trusted Platform Module – microcontroller which is designed to secure the hardware through integrated cryptographic keys) was improved that now it is possible to use NDES (Network Device Enrollment Service) to get certificates despite that devices and key attestations are not joined to the domain.

Active Directory Domain Services gives the administrator to store and manage information about resources from network. Now AD DS has new improvements that helps the organisations to protect Active Directory environments and give better identity management understanding for personal devices and corporate.

According to Microsoft (2018) the AD DS improvements include the following:

- Privileged access management – . It maintain a new administrative access solution that is configured by using MIM (Microsoft Identity Manager). It helps to mitigate security concerns for Active Directory environments that are caused by credential theft techniques such spear phishing, pass-the-hash, and similar kind of attacks. It grants a new administrative access solution that is configured by using MIM.
- Extending cloud capabilities to Windows 10 devices through Azure Active Directory Join – Azure Active Directory Join increases identity experiences for business, enterprise and EDU customers – with improved capabilities for personal and corporate devices.
- Connecting domain-joined devices to Azure AD for Windows 10 experiences – with device management in Azure Active Directory (Azure AD), administrator can ensure that users are accessing your resources from devices that meet the standards for security and compliance.
- Enable Microsoft Passport for Work in the organization.
- Deprecation of File Replication Service (FRS) and Windows Server 2003 functional levels (DCs can support rolling a public key only user's NTLM secrets. DCs can support allowing network NTLM when a user is restricted to specific domain-joined devices. Kerberos clients after they successfully authenticates with the PKInit Freshness Extension will receive the fresh public key for security identifier (SID)).

Active Directory Federation Services is responsible to secure sharing of identity information. Now user can enable to configure AD FS to authenticate users stored in Lightweight Directory Access Protocol (LDAP) directories. (Microsoft 2018.)

2.2.4 Networking

As the evolution of software-defined datacenters (SDDCs) continues, traditional networking methods such as virtual Local Area Networks (VLANs) begin to become difficult to manage and maintain. The requirements to centrally manage and control the network landscape directly from software to dynamically create what is needed when it is needed is a key piece to this evolving concept.

Introduced in Windows Server 2012 R2, software-defined networking (SDN) has evolved further to become more Azure consistent. (Microsoft 2018.)

Software-Defined Networking is an approach use open protocols. Administrator now can set both route and mirror traffic to current or new virtual appliances. The distributed firewall and Network security groups can allow the user to dynamically split or secure workloads (similar to Azure).

System Center Virtual Machine Manager was created to manage and deploy the entire SDN (Software-Defined Networking) stack. Additionally, now it is possible to associate SDN policies to the containers. Host only need Docker which can manage Windows Server container networking.

In Windows Server 2016 we have new TCP performance improvements. TCP Fast Open (TFO) has been implemented and Initial Congestion Windows (ICW) has been increased (from 4 to 10). Now with this new TFO implementation, it takes less time to establish a TCP connection. InitialCongestion Window increasement allows larger objects to be transferred in the initial burst. Those two things can decidedly reduce the time required to transfer an Internet object between the cloud and client.

Network Controller can interact with the network and be interacted with through two different APIs specifically for each function. The Northbound API (implemented as a REST API) is used to interact with Network Controller and monitor the network as well as implement configuration changes. The Southbound API is used to interact with network devices and detect service configurations and basically understand the network. Using tools such as System Center Operations Manager and System Center Virtual Machine Manager, admin can manage and monitor the network directly from these consoles. (Microsoft 2018.)

2.2.5 Storage

In this section I will introduce the reader with the improvements and features that was made in field of Storage.

Lets review new capability Storage Spaces Direct. Here we have some good improvements that Windows Server 2016 can allow. Now it is easier to manage or deploy the software-defined storage systems. Besides that now it is possible to use of new disk devices classes (NVMe disk and SATA SSD devices) that were not allowed to use it in any of previously used clustered Storage spaces with shared disks. (Tomsitpro 2018.)

Storage Spaces Direct enables service providers and enterprises to use industry standard servers with local storage to build highly available and scalable software defined storage. Using servers with local storage decreases complexity, increases scalability, and enables use of storage devices that were not previously possible, such as SATA solid state disks to lower cost of flash storage, or NVMe solid state disks for better performance. (Microsoft 2018.)

Storage replica can enable block-level, synchronous replication between clusters or servers which can be used for disaster recovery or stretching of a failover cluster between sites. Synchronous replication provides mirroring of data in physical sites with crash-consistent volumes to ensure zero data loss at the file-system level. It also allows the site extension beyond metropolitan ranges with the possibility of data loss. (Microsoft 2018.)

According to Microsoft (2018) Storage Replication now enables to do the following:

- Providing a single vendor disaster recovery solution for planned and unplanned outages of mission critical workloads.
- Using SMB3 transport with proven reliability, scalability, and performance.
- Stretching Windows failover clusters to metropolitan distances.
- Using Microsoft software end to end for storage and clustering, such as Hyper-V, Storage Replica, Storage Spaces, Cluster, Scale-Out File Server, SMB3, Deduplication, and ReFS/NTFS.
- Helping reduce cost and complexity as follows:
 - Is hardware agnostic, with no requirement for a specific storage configuration like DAS or SAN.
 - Allows commodity storage and networking technologies.

Features ease of graphical management for individual nodes and clusters through Failover Cluster Manager. Includes comprehensive, large-scale scripting options through Windows PowerShell.

- Helping reduce downtime, and increase reliability and productivity intrinsic to Windows.
- Providing supportability, performance metrics, and diagnostic capabilities.

Next important storage service is Storage Quality of Service (QoS). QoS allows to create management policies using Hyper-V and CSV clusters and centrally monitor end-to-end storage performance. This feature automatically improves storage resource fairness between numerous of virtual machines. To ensure the fairness the virtual machines use the same file server cluster and allow policy-based minimum and maximum performance goals.

2.2.6 Security and Assurance

Windows Server 2016 came with Just Enough Administration. This feature called Just Enough Administration is a new security technology that enables delegated administration to anything that is possible with Windows PowerShell. This security technology includes a support for running under a network identity, connecting over PowerShell Direct, securely copying files to/from Just Enough Administration (JEA) endpoints. Additionally, configuring the PowerShell console to launch in a JEA context by default.

Now to protect obtained domain credentials we can invoke Credential Guard. This new helpful feature uses Windows Defender Credential Guard which uses virtualization-based security to protect user secrets and only the privileged system software can access them. To access those secrets with unauthorized access can lead to credential theft attack (Pass-the-Hass or Pass-the Ticket). The protection from these attacks is used on NTLM password hashes, Kerberos Ticket Granting Tickets and credentials stored by applications as domain credentials.

In remote security we have some improvements in Remote Credential Guard. Credential Guard includes support for Remote Desktop Protocol (RDP) sessions

so that the user credentials remain on the client side and are not exposed on the server side.

When a user attempts to remote desktop to a remote host, the Kerberos request is redirected back to the originating host for authentication. The credential simply does not exist on the remote host any more. If a remote host (i.e., an end user's computer or server) has malicious code running on it that can obtain credentials, remote credential guard will mitigate this because no credentials will be passed into the remote host. According to Microsoft (2018), there are some requirements for remote credential guard to operate:

- The user must be joined to the same Active Directory domain or a remote desktop server must be joined to a domain with a trust relationship to the client device's domain.
- They must use Kerberos authentication.
- They must be running at least Windows 10, version 1607 or Windows Server 2016.
- The Remote Desktop classic Windows app is required. The Remote Desktop Universal Windows Platform app doesn't support Remote Credential Guard. To turn on remote credential guard, user can configure this via a group policy and widely deploy this across user estate.

Device Guard (Code Integrity) has some changes. Device Guard provides user mode code integrity (UMCI) and kernel mode code integrity (KMCI) by creating policies where defines what code can run on the server. Using Windows defender device guard with Windows credential guard we have get an additional protection to our AD (Active Directory) domain users.

Windows Defender Antivirus is now included in Windows Server 2016. This feature called Windows Server Antimalware is installed and enabled by default. Windows Server Antimalware user interface is not installed by default, user can install it by using the Add Roles and Features Wizard in case he feel to have that interface in his server. Anywat, this security feature will protect the computer without the user interface. (Thomasmaurer 2018.)

Control Flow Guard was added in code generation application options. CFG is a highly-optimized security platform feature that place the restrictions on where the

function can execute code from. In other words, it was created to combat memory corruption vulnerabilities.

2.2.7 Failover Clustering

In this section we have new feature called Cluster Operating System Rolling Upgrade. This feature enables an administrator to upgrade the OS of the cluster nodes from WinServer 2012 R2 to Windows Server 2016 without any of interruption in the Hyper-V or the Scale-Out File Server workloads.

New feature Cloud Witness was introduced in Windows Server 2016. This is a new type of Failover Cluster quorum witness that keep leverages Microsoft Azure as the arbitration point. The Cloud Witness can participate in quorum calculations. Cloud witness is more useful for the organizations that have multiple sites in clusters or are running Hyper-V clusters in a remote office environment where we need a backup for any scenarios.

Last failover clustering new improvement is called Health Service. It improves the day-to-day monitoring, maintenance, and operations experience of cluster resources on a Storage Spaces Direct cluster. This automation checks physical disks state. It calculates the fault possibilities (which are defined in Media Failure, Lost Communication and Unresponsive).

3 NANO SERVER AND CONTAINERS

Probably the one of the biggest features that Microsoft introduced in Windows Server 2016 was Nano Server. This lightweight server does not have local logon and this software is increasingly being used for managing containers. Besides this purpose, it was developed for running cloud applications as well.

In this part I will review in more detail what Nano Server is and analyse what the advantages of this software are. Moreover, there will be general information about containers which are divided in Windows Server containers and Hyper-V containers. Finally, I will compare these two container types.

3.1 Nano Server

As I mentioned earlier, Nano Server is a lightweight software where Microsoft removed the graphical user interface (GUI) of the OS and features like 32-bit support and various MSI and Server Core default components. By eliminating these components, Windows Nano Server gains greater speed, security, stability and it reduces resource consumption.

This remotely administrated server operating system is similar to Server Core mode but Nano Server can only support 64-bit applications, agents and tools. Of course, it takes less disk space and it is significantly faster. Additionally, it requires less updates and restarts are more faster than Windows Server. Nano Server has such benefits as 93 percent lower VHD size, 80 percent less reboots and 92 percent fewer critical bulletins.

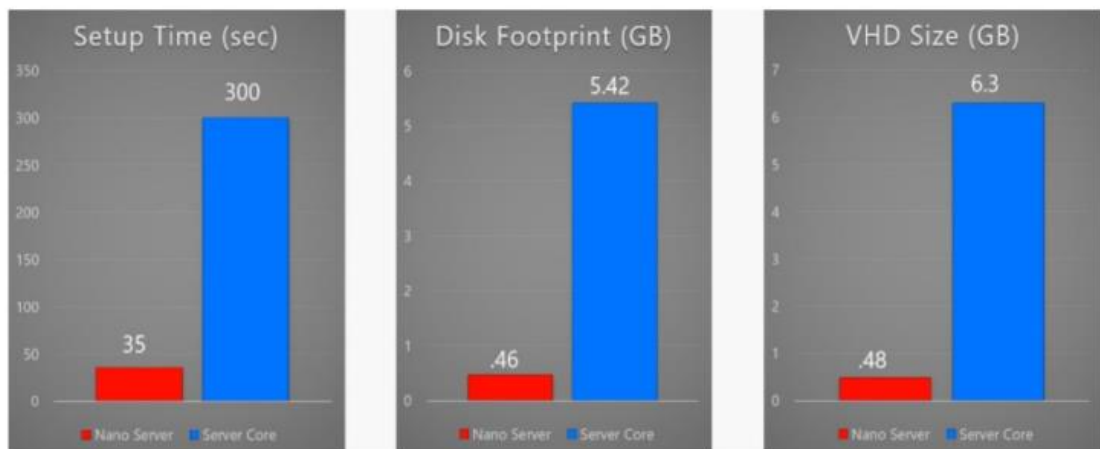


Figure 1. Deployment requirements Comparison between Nano Server and Core Server (Introducing Windows Server 2016 2018, 91)

Nano Server was improved since it was released, e.g. the Nano server version 1709 is about 80MB size, it was down from around 400MB. Now it is highly optimized for the .NET Core applications. Microsoft also updated it to run containers on Windows 10 IoT (Internet of Things) Core. This update allows containers to run on small devices.

Talking about Nano Server security which is really improved comparing to other Windows servers. With reduced attack surface, no GUI or Internet Explorer to exploit it became one of the most secured installation options for Windows server. As Figure 2 scheme shows, Nano Server has only 12 ports opened, which limits attacks against the server.

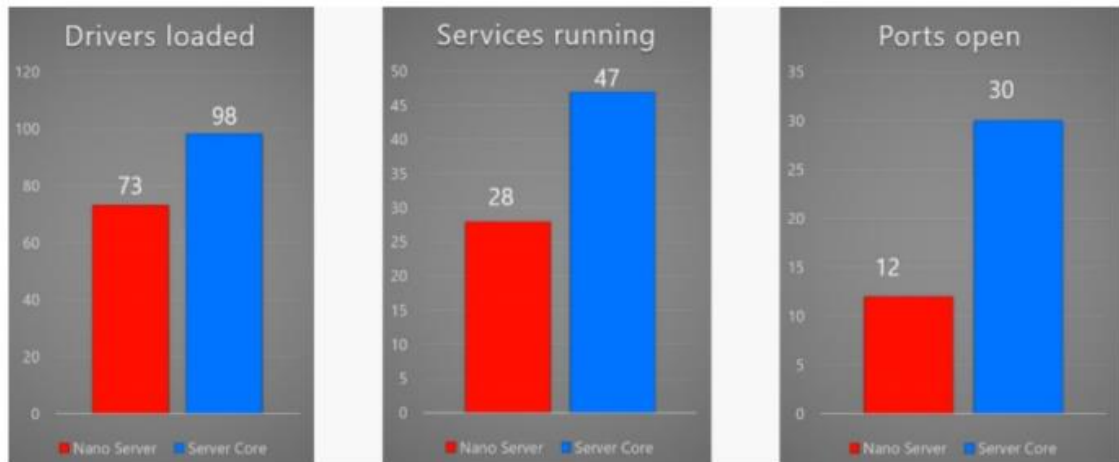


Figure 2. Default functionality comparison between Nano Server and Server Core (Introducing Windows Server 2016 2018, 90)

The tool to troubleshoot containers on Nano Server is Docker. This is an open-source project for automating the deployment of applications. Docker is also a company that develops and promotes this technology. The entire management of the OS is completed remotely via PowerShell and WMI (Windows Management Instrumentation). (Sixeyed 2018.)

There is a list of server roles that are currently supported on Windows Nano Server below in Table 1. The table was made according to Microsoft (2018).

Table 1. Nano Server roles / features

Role / Feature	Option
Storage	-Storage
Compute	-Compute
Clustering	-Clustering
Defender	-Defender
Reverse Forwarders	Added by default

Basic drivers for network adapters and storage controllers	-OEMDrivers
DNS Server role	-Package Microsoft-NanoServer-DNS-Package
PowerShell DSC (Desired State Configuration)	-Package Microsoft-NanoServer-DSC-Package
IIS (Internet Information Server)	-Package Microsoft-NanoServer-IIS-Package
Host support for Windows Containers	-Containers
System Center Virtual Machine Manager agent	-Package Microsoft-NanoServer-SCVMM-(Compute-)Package
System Center Operations Manager agent	Installed separately
Data Center Bridging (including DCBQoS)	-Package Microsoft-NanoServer-DCB-Package
Deploying on virtual machine	-Package Microsoft-NanoServer-Guest-Package
Deploying on a physical machine	-Package Microsoft-NanoServer-Host-Package
BitLocker, trusted platform module (TPM), volume encryption, platform identification, cryptography providers, and other functionality related to secure startup	-Package Microsoft-NanoServer-SecureStartup-Package
Hyper-V support for Shielded VMs	-Package Microsoft-NanoServer-ShieldedVM-Package
Simple Network Management Protocol (SNMP) agent	-Package Microsoft-NanoServer-SNMP-Agent-Package.cab
IPHelper service which provides tunnel connectivity using IPv6 transition technologies (6to4, ISATAP, Port Proxy, and Teredo), and IP-HTTPS	-Package Microsoft-NanoServer-IPHelper-Service-Package.cab

As we can see Nano Server made Windows Server 2016 look more interesting. It has high security level, which means that it is more protected from hackers. Nano Server promises that users will need only few reboots and because of lightweight, each reboot will not take too long. By removing GUI of the OS and features like 32-bit support and various MSI and Server Core default components the server gains big speed for the processes. Additionally, Nano Server is great software to manage containers. (Techtargeter 2018.)

Nano Server is truly headless – there is no way to use Remote Desktop to connect remotely. As a result, user must perform all Nano Server management remotely, either via Windows PowerShell, Windows Management Instrumentation (WMI), Windows Remote Shell (WinRS), Emergency Management Services (EMS), or remote GUI tools. (Tomsitpro 2018.)

In addition to the command-line remote management options discussed in the previous sections. User can use many existing remote GUI tools to remotely manage Nano Server. Because there is no local sign-in or Remote Desktop in Nano Server and there are tools that even in Server Core do not have remote GUI replacements. For example Task Manager, there are a set of web-based remote GUI called Server Management Tools available in Azure today. User can use these web-based remote GUI tools to manage Nano Server as well as Server Core and any of the other installation options. (Microsoft 2018.)

3.2 General Information about Containers

A container in its simplest form is exactly that – a container. It is an isolated environment in which user can run an application without fear of changes due to applications or configuration. Containers share key components (kernel, system drivers, and so on) that can reduce startup time and provide greater density than user can achieve with a VM. (Microsoft 2018.)

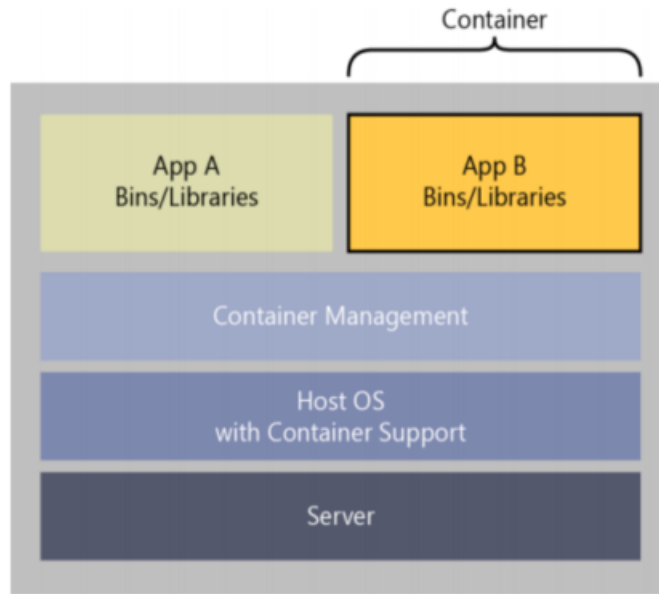


Figure 3. Abstract container layout (Microsoft 2018.)

Figure 2 shows how we can see that the host OS can host many containers and these containers can be isolated while sharing OS, such as the kernel (key components).

A huge plus about containers is that they are the applications itself. They might have various dependencies that exist only within the container. All the dependencies are required to run the application. This means that, if something wrong happens on Application A, it will not affect Application B. It is very useful in case we need to test out a new feature. To test the feature, we can install it on Application A and wait for feedback. In case there is no bad feedbacks, we can install it on Application B.

Because all binaries and dependencies are hosted within the container, the application running in the container is completely portable. Essentially, this means that developer can deploy a container to any host running the container manager software, and it will start and run without any modification. For example, a developer can begin developing his application and deploy it into a Hyper-V container using Windows 10 Anniversary Edition. When he is ready to roll it out in

production, it can be run on Windows Server 2016, including Nano Server, in a public, private or hybrid cloud. (Microsoft 2018.)

Containers are built on layers. The first layer is the base layer. This is the OS image on which all other layers will be built. This image is stored in an image repository so that user can refer to it when necessary. The next layer (and sometimes the final layer) is the application framework layer that can be shared between all of the applications. For example, if the base layer is Windows Server Core, the application framework layer could be .NET Framework and Internet Information Services (IIS). The second layer can also be stored as an image, which, when called, also describes its dependency on the base layer of Windows Server Core. Finally, the application layer is where the application itself is stored, with references to the application framework layer and, in turn, to the base layer. (Microsoft 2018.)

Containers have very big advantages. Firstly, containers can grant higher scale versus deploying an application to a VM. In the VM model user need at least three VMs – for development environment, production and testing. In a container model user need only one. That single VM, running a container manager, can run these 3 simulate on environments. So containers require us to have fewer VMs to run the environment and this allows us to achieve extremely higher scale in the cloud environments. (Argon Systems 2018.)

The second advantage, as I mentioned before, is for developers. The biggest pain for developers is when they are building applications revolves around moving the application from a development to test and then to production environments. Developers must spend a lot of time and effort checking the application's dependencies as it moves through the environments. However, when an application is deployed to a container, user can move the container between environments, because it is isolated and all of the binaries reside within the container itself. (Microsoft 2018.)

Morover, containers have some benefits. Firstly, they are lightweight. The operating system kernel is used by all containers once they are on a single machine. This means that they can start instantly and this makes more efficient use of RAM. They can share common files, making disk usage and image downloads much more efficient. The startup is much faster than a VM. It also has a small footprint and it scales well.

Secondly, containers are secured. All containers or container applications are isolated from each other and the underlying infrastructure while providing an added layer of protection for the application. Next, containers are infrastructure independent. The container are portable and can be run on Linux, Windows or cloud.

In addition, it is simple to rollback. It is easy to modify the deployment script and redeploy the container image. With VMs, administrator can rebuild the entire machine (or revert to the previous backup/snapshot), if needed.

3.3 Windows Server Containers Versus Hyper-V Containers

Windows Server 2016 provides us with two types of containers:

- Windows Server Containers
- Hyper-V Containers

Windows containers work the same as Linux containers. Each containerized applications belong to user-mode, isolated container on a shared host operating system. Containers can share the same libraries and while the application has a dependency on a certain Operating System version and a base OS image may be downloaded, there it must match the host OS version because they share the same common kernel and OS.

According to ITProToday (2018), There are two challenges that may cause a problem in certain environments:

1. Not enough isolation since the isolation is at user-mode meaning a shared kernel. In a single tenant environment where applications can be trusted this is not a problem but in a multi-tenant environment a bad tenant may try to use the shared kernel to attack other containers.
2. There is a dependency on the host OS version and even patch level which may cause problems if a patch is deployed to the host which then breaks the application.

Hyper-V containers can help to prevent those problems. Hyper-V containers use the main image which is defined for the application. It automatically uses that base image and creates a Hyper-V VirtualMachine.

Hyper-V containers are still using Windows containers within the VM. The only difference is the Windows container is now running inside a Hyper-V VM which provides kernel isolation and separation of the host patch/version level from that used by the application. The application is containerized using Windows containers and then at deployment time user can pick the level of isolation required by choosing a Windows or Hyper-V container. (ITProToday 2018.) In Figure 4 we can see how does Windows Containers and Hyper-V Containers looks like.

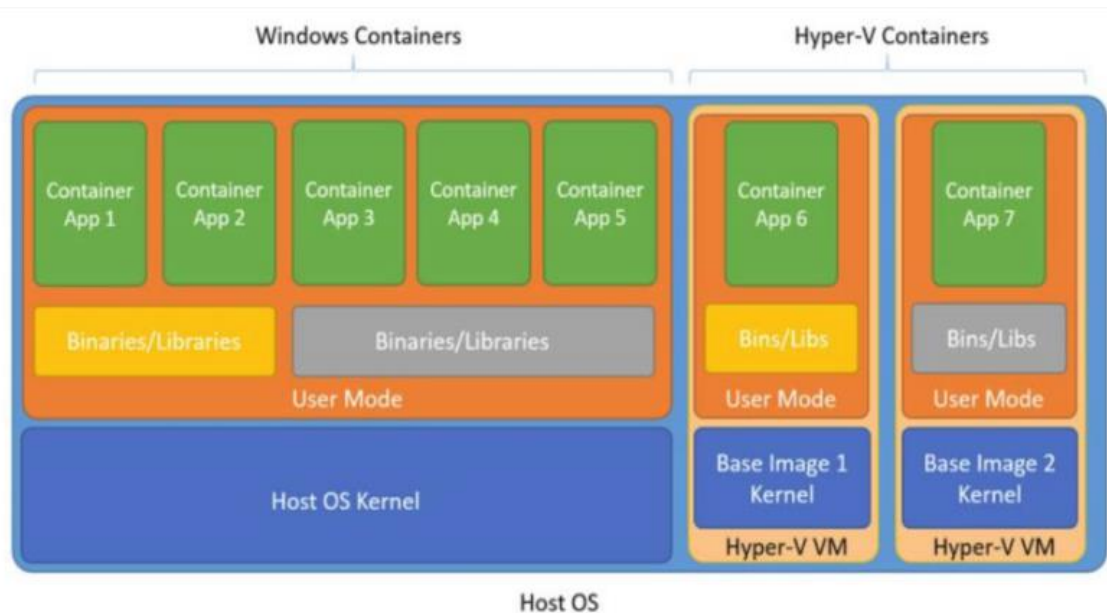


Figure 4. Windows Containers Versus Hyper-V Containers (ITProToday 2018.)

There we can notice that Hyper-V containers can use a common base image and it will not require a manual management of the VMs. The Virtual Machines are created and deleted automatically. Additionally, it is possible to use nested virtualization once Windows Server 2016 can support that. This feature means that even if our container host is a Hyper-V Virtual Machine, we still are able to use Hyper-V container on container host as it enables us to create VMs in VM. (Tomsitpro 2018.)

With the introduction of Windows Server containers and Hyper-V containers, Docker becomes even more useful because user can use it to manage Docker containers on Windows as well as the traditional Linux environment. Additionally, there we can access to all of the images that are available through Docker, so we can download and deploy.

The Docker runtime engine works as an abstraction on top of Windows Server containers and Hyper-V containers. Docker provides all of the necessary tooling to develop and operate its engine on top of Windows containers, be it Hyper-V containers or Windows Server containers. This will afford the same flexibility of developing an application in one container and being able to truly run it anywhere.(Microsoft 2018.)

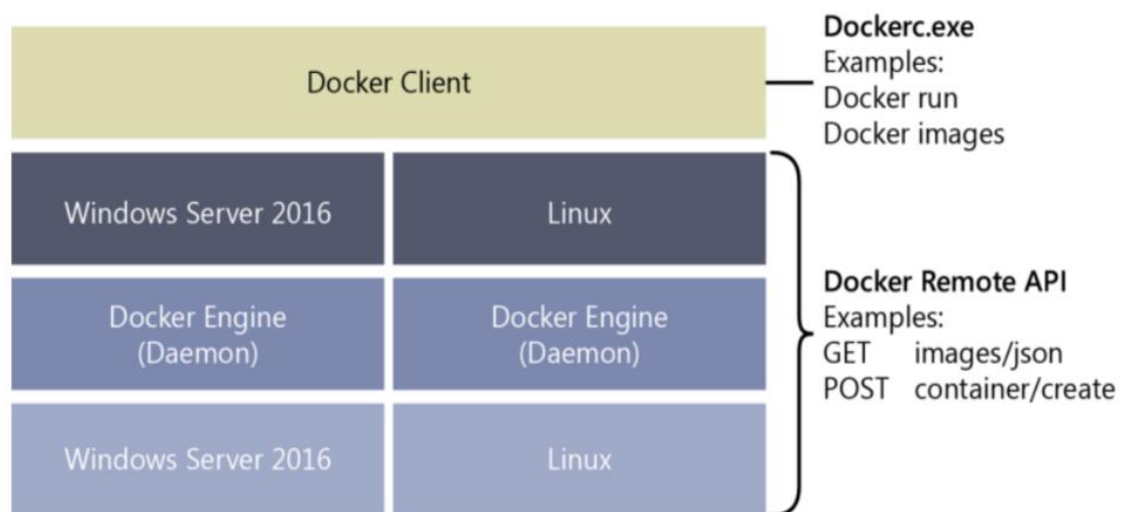


Figure 5. The Docker engine on Windows and Linux (Introducing Windows Server 2016 2018, 101)

The Docker engine runs at the same level in either a Windows Server container or Linux container environment, and it can run with Windows Server or Linux above the Docker engine. The Docker client will connect to any Docker engine and provide a consistent management experience for the end user.

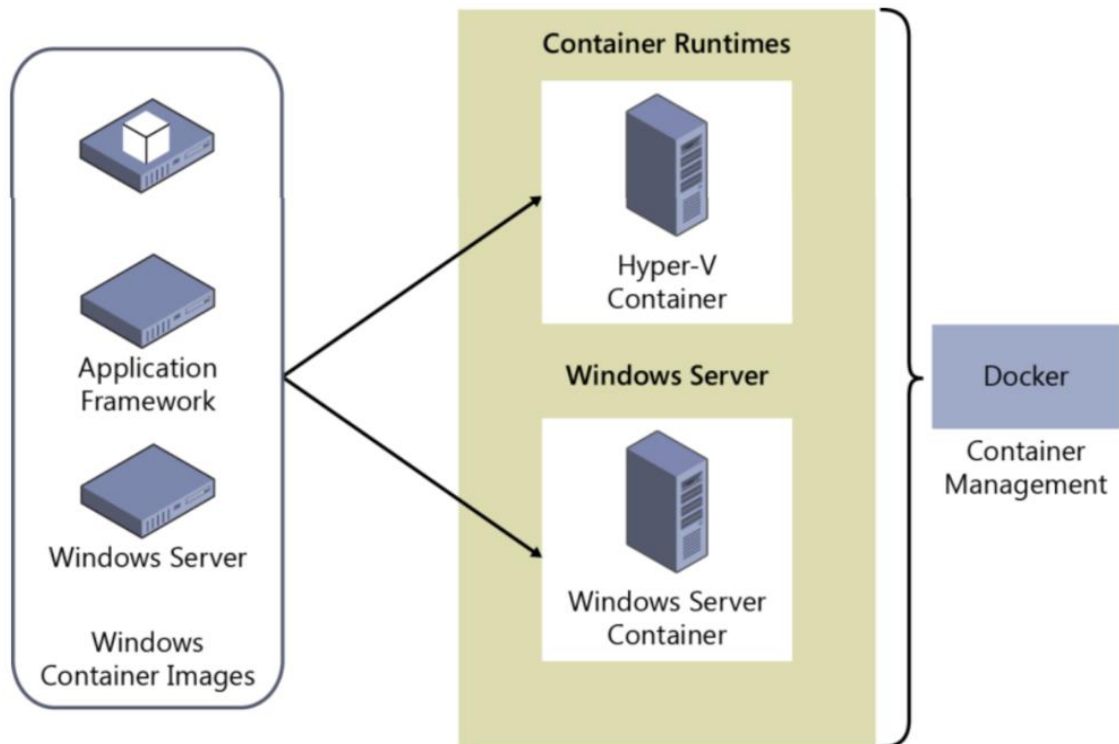


Figure 6. Containers can write once and deploy anywhere (Introducing Windows Server 2016 2018, 102)

As we can see in Figure 6, containers can be deployed anywhere – to a Hyper-V container or Windows Server container. All the management is done with the Docker.

To sum up, containers are lightweight and useful for everyone, especially for developers. Anyway, it can cause some problems: 1. Bad tenant may try to use the shared kernel and attack other containers. 2. There is a dependency on the host OS version. Hyper-V is the best solution to solve these problems. Hyper-V containers use the main image which is defined for the application.

4 PRACTICAL PART

In this part I will create the Nano server and do some implementations with containers using Nano server. There will be the information on how to install and configure the Nano server. Additionally, I will check what is possible to manage while we are connected locally.

For my practical part I have a Windows Server 2016. This server was assigned to ITLAB organization (domain). All the specifications are listed below in Table 2.

Table 2. Windows Server 2016 software and hardware specification list

Edition	Processor	RAM	System Type
Windows Server 2016 Datacenter	Intel® Xeon® CPU E5-2620 v3 @ 2.40GHz	8 GB	64-bit OS

As we can see, the server is powerful. It matches the requirements of the practical part's tasks.

4.1 Nano Server Installation

To begin, we have to download the Windows Server 2016 ISO file which is easily downloadable from the official Microsoft web page. After we downloaded the ISO file, we have to mount it to any of available drives and here we have Nano Server with all the packages and the Nano Server image generator which are the scripts that we will use in order to configure the Nano Server.

Moreover, we create a folder and name it Nano, there we paste the script that is in the Nano Server image generator folder (in ISO). After we complete that, we need to open the Windows PowerShell as administrator. Here we will import the module of the Nano Server which is *NanoServerImageGenerator.psm1*. The command is introduced in Figure 7.

```
PS C:\Users\Administrator> Import-Module C:\Nano\NanoServerImageGenerator.psm1
```

Figure 7. Nano server image generator

Now we have the module and we have to continue with Nano Server creation which will be done with a script. Figure 8 shows the command.

```
PS C:\Users\Administrator> New-NanoServerImage -Edition Standard -MediaPath E:\ -BasePath C:\Nano -TargetPath C:\Nano\robertas-nano01.vhdx -DeploymentType Guest -ComputerName Nano-01 -storage -Package Microsoft-NanoServer-IIS-Package

cmdlet New-NanoServerImage at command pipeline position 1
Supply values for the following parameters:
AdministratorPassword: *****
Done. The log is at: C:\Nano\Logs\2018-04-03_21-53-22-07
PS C:\Users\Administrator>
```

Figure 8. Creating new Nano Server image

Edition – Here we set the Nano Server edition. Here we choose the *Standard* edition.

MediaPath – Here we specify where the Windows Server 2016 media is currently mounted. Here we write that its *E:*.

BasePath – It will create a temporary location for the Nano Server to store its files before configuring the final vhdx file that we are going to use. Here we write *C:\Nano*.

TargetPath – It will configure the vhdx file for the Nano Server that we will later attach it for our Hyper-V. Here we specify it as *C:\Nano\robertas-nano01.vhdx* where we set '*robertas-nano01.vhdx*' as the name of our vhdx file.

DeploymentType – Here we write *Guest*, which means that the Nano Server will be used as a Virtual Machine. This will add the additional driver to the server so it will allow the Nano Server to run as a Virtual Machine.

ComputerName - Here we have to write the computer name as I chose to name it *Nano-01*.

Storage – This option is needed to install the storage packages.

Package – Here we want to install IIS packages. With this package we can test our server and see if its going to be a good OS application server.

To proceed with the Nano Server Image installation we are forced to write the administrator password. This password will be used to login locally to the Nano Server.

The installation process takes a few moments, it configures the virtual disk. If we go to the *C:\Wano* folder we can see some temporary folders that have been created after the vhdx. This is shown in Figure 9.

Name	Date modified	Type	Size
Logs	3.4.2018 21.53	File folder	
Packages	3.4.2018 21.53	File folder	
Convert-WindowsImage	16.7.2016 5.30	Windows PowerS...	160 KB
NanoServer.wim	16.7.2016 20.12	WIM File	168 709 KB
NanoServerImageGenerator	26.5.2016 0.42	Windows PowerS...	1 KB
NanoServerImageGenerator	15.7.2016 0.18	Windows PowerS...	99 KB
robertas-nano01	3.4.2018 21.55	Hard Disk Image F...	550 912 KB

Figure 9. Nano folder view

As now we have a Virtual Machine, we can deploy it on Hyper-V. To accomplish that we have to install Hyper-V. We have to add a new role which will be Hyper-V. After the installation it is possible to create a new Virtual Machine.

It is done by using Hyper-V Manager. We have to press on *New > Virtual Machine*. Firstly, we have to add the name of the server which is going to be *'NanoServer-01'*. After that we enable *'Store the virtual machine in a different location'* and choose the folder where we moved the vhdx file.

In the generation section we choose *Generation 2* because it provides a support for newer virtualization features and this generation has better reviews for deploying the Nano Server.

Here we will not need a lot of RAM, so we leave 1024 MB which is a default value. In the network section we choose the network that we have or can create a new network adapter, if needed.

In the section *'Connect Virtual Hard Disk'* we mark *'Use an existing virtual hard disk'*. Here we have to attach the vhdx file that was created before. To complete that we just press *Finish*.

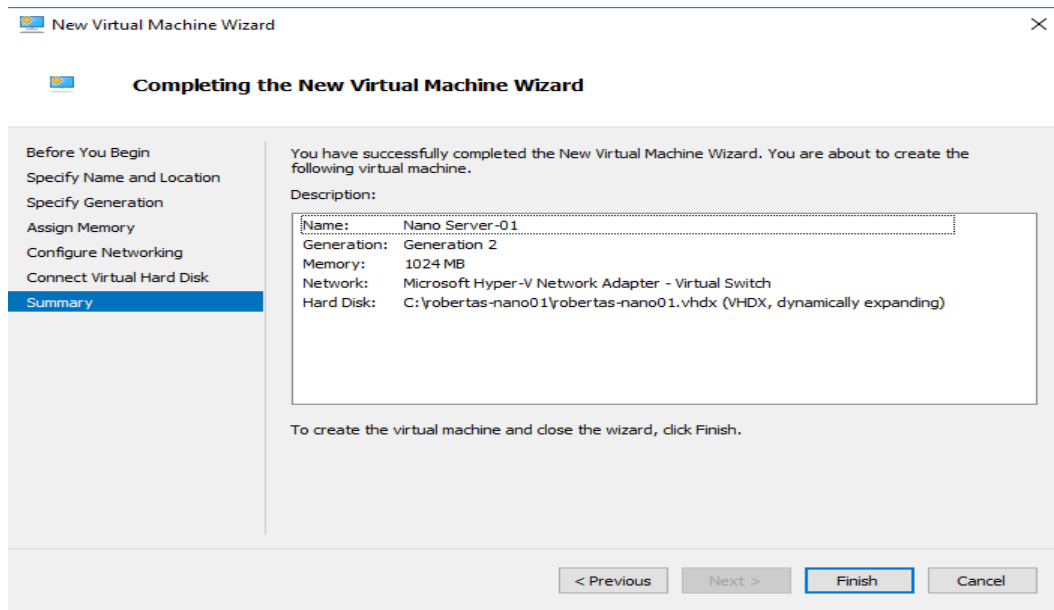


Figure 10. Nano Server deployment on Hyper-V

Finally, the creation only takes a few minutes. After the process is completed, we can connect to the server and power it on. It takes only few moments to load, here we have a view that reminds a bit of Command Prompt .

The view for login locally is shown in Figure 11. Here we can type User name, Password and Domain. Domain is needed only when we have signed our Nano server to the domain. In that case we can use the domain account to connect to the Nano Server.



Figure 11. Nano Server login locally

After we have successfully logged in, we can see the main server information – computer name, user name, workgroup, OS, local date and time. All the other configurations should be done connected remotely.

Additionally, we have networking configurations, where we can enable DHCP or set our static IP address. Of course, we need a static IP address for the server. To manage that, we have to go to the networking configurations and press F11 to configure IPv4 settings.

By default, we have DHCP enabled, to disable it we have to press F4 which will toggle the DHCP to disabled. After we have disabled the DHCP, we can enter our preferable IP address, subnet mask and default gateway. We use TAB to go through all of these fields. Finally, we press ENTER two times to save the new configurations.

```
IP Configuration
=====
Ethernet
Microsoft Hyper-V Network Adapter
00-15-5D-02-8C-00
-----

DHCP          [ Disabled ]
IP Address    172.16.2.141_____
Subnet Mask   255.255.248.0_____
Default Gateway 172.16.0.5_____

-----
ESC: Cancel | ENTER: Save
```

Figure 12. Nano Server IP configuration

Now we have set the IP address to the Nano Server. Before going any further, we can enable File and Printer inbound firewall rules that will set the Nano Server

available on the network. After we enabled these rules, our server network is fully configured. We can test it out by searching Administrator Share.

On Windows Server 2016 we can open Nano Server administrator share. In File Explorer we type [\\172.16.2.141\c\\$](file://172.16.2.141/c$). Basically, we can see here all the Nano Server files. This is shown in Figure 13.

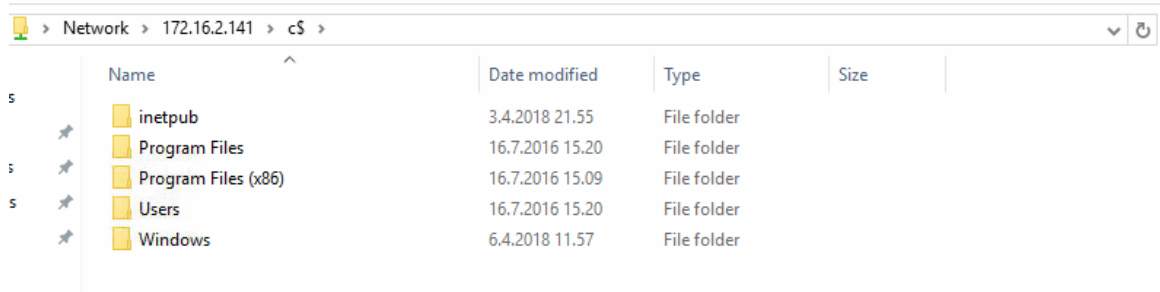


Figure 13. Nano Server shared files.

Here we can manage the files that are on Nano server. We can delete or copy some files to this folder locally. However, I will use this availability later and now move on to review the remote connection.

Remote connection to the server can be done using Windows PowerShell. We run it as an administrator. In the first step, we should add the exception to the domain controller in order for it to allow the remote connection to the Nano Server. Issue a command: `Set-Item WSMAN:\Localhost\Client\TrustedHosts "172.16.2.141"`. The last step is to confirm the modifications by entering Y.

In the next step we can specify the new variable by typing `$ip="172.16.2.141"`. The second command to enter the PowerShell session: `Enter-PSSession -ComputerName $ip -Credential $ip\Administrator`. Here we also have to specify the computer name which is our variable. The credential is our ip and administrator because we use a local account to connect to the Nano Server. This is shown in Figure 14.

```

PS C:\Users\Administrator> Set-Item WSMAN:\localhost\Client\TrustedHosts "172.16.2.141"
winRM Security Configuration.
This command modifies the TrustedHosts list for the WinRM client. The computers in the TrustedHosts list might not be
authenticated. The client might send credential information to these computers. Are you sure that you want to modify
this list?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PS C:\Users\Administrator> $ip="172.16.2.141"
PS C:\Users\Administrator> Enter-PSSession -ComputerName $ip -Credential $ip\Administrator
[172.16.2.141]: PS C:\Users\Administrator\Documents>

```

Figure 14. PowerShell session to connect to the Nano Server

The first thing that I will test is if the Nano Server is working fine and is fully operational. The first step is to create html file and name it default.html where we type the following:

```

<h1><center>Welcome to Windows Nano Server! </h1>
<h1>server is online</center></h1>

```

The next step is to use File Explorer and type [\\172.16.2.141\c\\$\inetpub\wwwroot](http://172.16.2.141/c$/inetpub/wwwroot). Here we place the file we just created. Finally, in Internet Explorer we type our Nano Server IP address <http://172.16.2.141/>. There should be the *default.html* code visible. This confirms that the server works fine and we can implement something more. Figure 15 shows what it looks like.

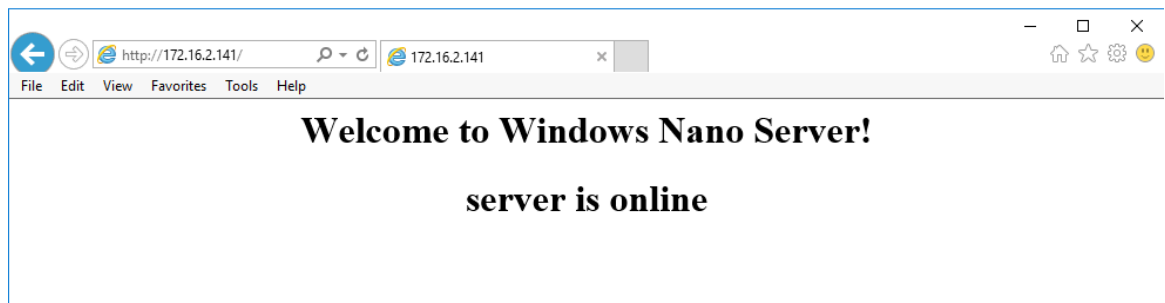


Figure 15. Nano Server web page

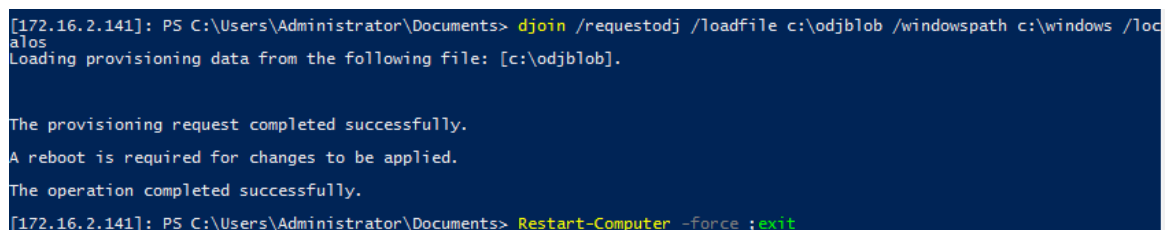
By entering this to our browser, it is possible to see if our Nano Server is online. In case the server is offline, it will not load the page. Additionally, we can implement something more with this tool.

The next step is to add Nano server to the domain. There are two options to configure that and join the domain. The first option is by going into the script and adding the domain there. The second option is by doing an offline domain join

using the `djoin` option. This is going to create a blob file that will be used to present the Nano Server on a later state.

In order to create the offline domain join blob file it has to be done on the domain controller and issue a command: `djoin.exe /provision /domain itlab.xamk.fi /machine Nano-01 /savefile C:\odjblob`. After the command has been issued, we can find our `djoin.exe` file on C:\ disk. Another essential point is to copy this `odjblob` file and move it to the Nano server folder. We can do that by entering [\\172.16.2.141\c\\$](#) shares folder where we can execute it locally.

Moreover, we have to connect to the Nano Server remotely and execute this `odjblob` file. After the remote connection is started, we can issue the command `djoin /requestodj /loadfile c:\odjblob /windowspath c:\windows /localos`. We should receive a message which will confirm that is completed successfully. This is shown in Figure 16.



```
[172.16.2.141]: PS C:\Users\Administrator\Documents> djoin /requestodj /loadfile c:\odjblob /windowspath c:\windows /localos
Loading provisioning data from the following file: [c:\odjblob].

The provisioning request completed successfully.
A reboot is required for changes to be applied.
The operation completed successfully.
[172.16.2.141]: PS C:\Users\Administrator\Documents> Restart-Computer -force ;exit
```

Figure 17. `djoin` on Nano Server remote connection

To finish the domain joining we have to restart the Nano Server. We can do that in the PowerShell remote connection by typing: `Restart-Computer -force; exit`. As a result, we can now connect to our Nano Server using domain credentials. In our local logon we can see that we have connected not with workgroup but *itlab.xamk.fi* domain credentials.

To summarise, Nano Server is simple to install. It only requires to have a Windows Server 2016 ISO file. In this file we can find Nano Server folder with an image generator which makes it more easier to install. In addition, the server works fine, all reboots takes only few moments. Locally we can configure the network and inbound-outbound firewall configurations. To fully manage it, we

have to use remote connection, which is possible to do in PowerShell by creating a new remote connection to the server.

4.2 Basic Implementations with Containers

In this section I will test some main features that are possible to do with the containers. Here we will start with installation and finish with deploying an app using composer. It was implemented on Windows Server 2016.

At first, we have to download the Docker which is one of the best application is to manage with containers. In the browser search field we type <https://docs.docker.com/docker-for-windows/install/> (2018.) and choose which of channel you are willing to download. There are two options – Stable or Edge. Edge is useful when we are trying to have experimental features faster but it can contain some bugs and instability. Stable is the best option, if we want a reliable platform to work with. Here I am willing to test some basic things with containers so the Stable channel is the best option for me.

After we have downloaded our preferable docker version, we can install it. In the end it gives us a suggestion to login. It is totally free to register on Docker, anyway, Docker does not require the user to login.

To start with managing the containers, open PowerShell. We issue the commands that will confirm us that we have Docker on our server. Enter *docker --version*. It will ensure that here we have a supported Docker version. Besides, we can issue this command without '--'. Without these symbols it will show more details about our Docker.

```
PS C:\Users\Administrator> docker --version
Docker version 18.03.0-ce, build 0520e24
PS C:\Users\Administrator>
PS C:\Users\Administrator> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9bb5a5d4561a: Pull complete
Digest: sha256:f5233545e43561214ca4891fd1157e1c3c563316ed8e237750d59bde73361e77
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Figure 18. Starting with Docker

Later, we can test if the Docker installation works fine by running the simple Docker image. The command is: `docker run hello-world`. It will download the image for `hello-world` and will deploy that. Figure 18 shows how it looks.

After we have tested Docker by issuing `hello-world`, we can check how Dockerized applications work. We will try something more complex there and will implement a webserver. Here we have to issue the `docker run --interactive --tty ubuntu bash` command. It will create an Ubuntu OS image and it will run on an interactive terminal inside the spawned container.

The previous steps will move us into the container. At the beginning we are on the root # prompt. The command `hostname` can show hostname of the container. Exit is the command to exit the container prompt. It is shown in Figure 19.

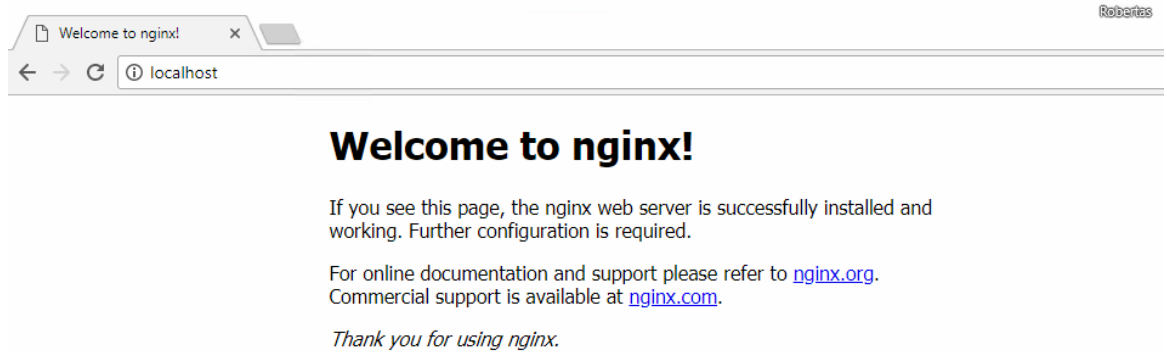
```
PS C:\Users\Administrator> docker run --interactive --tty ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d3938036b19c: Pull complete
a9b30c108bda: Pull complete
67de21feec18: Pull complete
817da545be2b: Pull complete
d967c497ce23: Pull complete
Digest: sha256:9ee3b83bcaa383e5e3b657f042f4034c92cdd50c03f73166c145c9ceaea9ba7c
Status: Downloaded newer image for ubuntu:latest
root@967b032ad300:/# hostname
967b032ad300
root@967b032ad300:/# exit
exit
PS C:\Users\Administrator> docker container ls --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
967b032ad300	ubuntu	"bash"	About a minute ago	Exited (0)	About a minute ago
aad22ade8bbf	xenodochial_bardeen	"/hello"	About an hour ago	Exited (0)	About an hour ago
	hello-world				
	eager_turing				

Figure 19. Container prompt

The command `docker container ls --all` will show the list of containers. We issue that with '`--all`', because no containers are running now after we exited the container. It shows the information that contains the container ID, image name, command, created (time), status and ports.

Next, I will create and run a webserver. This dockerized nginx server will be named `webserver`. We open PowerShell and type `docker run --detach --publish 80:80 --name webserver nginx`. Here we specified the default HTTP port and name. The option `--detach` sets the container in a detached mode exit, which means that it will exit the process when the root process used to run the container exits.



```

PS C:\Users\Administrator> docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
f10a687862d1      nginx              "nginx -g 'daemon of..." About a minute ago  Up About a minute  0.0.0.0:80->80
/tcp              webserver

```

Figure 20. Webserver and running containers

We redirect the browser at <http://localhost> to see that our nginx webserver is online and ready. Here we do not have to write :80, because we have specified the default HTTP port in the command before. The webserver is successfully working as we can see in Figure 20. We issue the command `docker container ls` to check the running containers. Here the webserver is listed. The command `docker container stop webserver` will stop the running nginx container. The nginx container is assigned the name `webserver`. In addition, we issue `docker container rm webserver` to remove the container, if needed.

The third thing that we will test is Docker Compose. There I will run a web application on Docker Compose. This website contain with a hit counter in Redis.

To begin here, we have to install Compose. In PowerShell we issue `[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12`. Compose requires TLS1.2 (Transport Layer Security). Now we can issue `Invoke-WebRequest "https://github.com/docker/compose/releases/download/1.21.0/docker-compose-Windows-x86_64.exe" -UseBasicParsing -OutFile $Env:ProgramFiles\docker\docker-compose.exe`, which installs Compose version 1.21.0.

As we have compose ready, it is time to setup application dependencies. Now we create the directory which will maintain the project files *mkdir compose* and go to the folder *cd compose*. Here we create a file and name it *app.py*. There we place the webserver code which will count entries to the page (Appendix 1). The port for Redis is 6379.

Next, we create a *requirements.txt* file where we write: *flask redis*, where Flask is a micro web framework and Redis is a data structure store. After this text file is created, we have to create a *Dockerfile* with command *New-Item Dockerfile -type file*. Following Docker (2018) guidelines we write here as following:

```
FROM python:3.4-alpine
ADD . /code
WORKDIR /code
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

This code makes the Docker to build an image, adding the current directory into the path */code* in the image sets the working directory to */code*, installs the Python dependencies and sets the default command for the container to Python *app.py*. (Docker 2018.)

```
PS C:\Users\Administrator\compose> New-Item Dockerfile -type file

Directory: C:\Users\Administrator\compose

Mode                LastWriteTime         Length Name
----                -
-a----             15.4.2018          22.26          0 Dockerfile

PS C:\Users\Administrator\compose> docker-compose up
Building web
Step 1/5 : FROM python:3.4-alpine
3.4-alpine: Pulling from library/python
81033e7c1d6a: Pull complete
9b61101706a6: Pull complete
415e2a07c89b: Pull complete
f22df7a3f000: Pull complete
af78bda78f1f: Pull complete
Digest: sha256:989b6044c434ffadf4dbc116719d73e7e31f5ac0f75f59b7591aeb766c874e26
Status: Downloaded newer image for python:3.4-alpine
```

Figure 21. Build and run the app with Compose

While we are in the project directory, we start up the application by issuing the command *docker-compose up*. It will build an image for the code and starts the services. We open a browser and type *localhost:5000* to see that the application is running. This application will count how many time the page has been seen.



Figure 22. View of the webserver Compose

This counter will work until we press *ctrl+c* in our PowerShell to stop the Compose. Anyway, we can run it on the background by typing *docker-compose up*. And we type *docker-compose stop* once we want to stop this compose.

The last thing I want to test with containers, is to run the Nano Server in a container. For this process we will have to switch the Docker host to Windows containers mode. It can be done by clicking on Docker icon and choosing 'Switch to Windows containers'. It will take a minute to switch. When we are switched to that mode, we can download Nano Server of the hub. We enter the command *docker pull Microsoft/nanoserver*. It will automatically download and extract the image. (Argon Systems 2018.)

```

PS C:\Users\Administrator> docker pull microsoft/nanoserver
Using default tag: latest
latest: Pulling from microsoft/nanoserver
bce2fbc256ea: Pulling fs layer
83eec61707e8: Pulling fs layer
image operating system "windows" cannot be used on this platform
PS C:\Users\Administrator> docker pull microsoft/nanoserver
Using default tag: latest
latest: Pulling from microsoft/nanoserver
bce2fbc256ea: Pull complete
83eec61707e8: Pull complete
Digest: sha256:59558bd57c0d14a4df5b827a676fb061abacefebfa2089038f018cf9eea17ecb
Status: Downloaded newer image for microsoft/nanoserver:latest
PS C:\Users\Administrator>
  
```

Figure 23. Create the Microsoft Nano Server Container

After the installation is completed, we can run the container. To run the Nano Server container we enter the command: *docker run -i -t Microsoft/nanoserver*. The options *-i* and *-t* will allocate a tty for the container process. It will initiate the container and we will be inside the container. (Argon Systems 2018.)

As Figure 24 indicates, we can see that the hostname is the container ID. We can check the IP configurations. We can also test the connectivity by pinging to the container from the host server.

```

Administrator: Windows PowerShell
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\>ipconfig

Windows IP Configuration

Ethernet adapter vEthernet (Container NIC 5c3f8ae9):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::a0b4:43ef:1c5c:4383%38
    IPv4 Address. . . . . : 172.23.224.33
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 172.23.224.1

C:\>hostname
9d01919ea3a7
C:\>

Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>hostname
Win-Robertas

C:\Users\Administrator>ping 172.23.224.33

Pinging 172.23.224.33 with 32 bytes of data:
Reply from 172.23.224.33: bytes=32 time<1ms TTL=128
Reply from 172.23.224.33: bytes=32 time<1ms TTL=128
Reply from 172.23.224.33: bytes=32 time<1ms TTL=128
Reply from 172.23.224.33: bytes=32 time<1ms TTL=128

Ping statistics for 172.23.224.33:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>

```

Figure 24. Inside the Nano Server container

Thus, it is possible to do the configurations while we are in the container or connect to it remotely. Of course, it is possible to run PowerShell inside the container. We should go to *C:\Windows\System32\WindowsPowerShell\v1.0* and type *powershell* which will open PowerShell. (Argon Systems 2018.)

To summarise, containers are handy and simple to use. There is a lot of information on the internet which can help to run Dockers. In a container we can run webservers, Nano servers or other services. Container is useful when we are trying to move the application to the development or production environment. It is isolated and all the binaries reside within the container itself. It is providing us with higher scale versus deploying an application to a VM.

5 CONCLUSION

The purpose of this thesis was to review new impacts of Windows Server. There are Hyper-V improvements that allow a different set of credentials. Remote Hyper-V hosts can now communicate with Hyper-V using the WS-MAN protocol. There are Windows containers that are isolated applications and can run on a single system. In the computing field, there is an innovation – Nano Server. It is a lightweight server that is adapted to be used as an OS layer for virtualized container instances.

Windows PowerShell 5.1 includes important new features. It now supports new security features and devlopinf with classes that can improve its usability. In this new Windows Server version we can find a modern OneGet feature that can enable professionals or DevOps to automate software installation, discovery locally or remotely.

SDN now can set a route and mirror traffic to current or new virtual appliances. It is possible to associate SDN policies to the containers. The network controller can communicate with the network and be interacted with through two different APIs specifically for each function. TCP Fast Open has been implemented. With this new feature, it takes less time to establish a TCP connection.

The storage field has been improved. Now it enables using storage devices that were not previously possible, e.g. SATA SSD to lower the cost of flash storage, or NVMe SSD for better performance. Storage replica can be enabled in block-level synchronous replication between clusters or servers. It is used for disaster recovery or stretching a failover cluster between sites. Storage Quality of Services can allow user to create management policies using Hyper-V and SV clusters.

Now we have Just Enough Administration which is a security technology and it enables delegated administration to anything that is possible with Windows PowerShell. To protect the obtained domain credentials we can invoke Credential Guard which uses a virtualization-based security. Only privileged system

software can access that. Now Credential Guard can support Remote Desktop Protocol sessions. User credentials remain only on the client side.

Failover Clustering has some new improvements. The Cloud Witness can be used for the organizations that have multiple sites in clusters or are running Hyper-V clusters in a remote office environment. It is useful to do a backup for any of disaster scenarios. The new feature Health Service improves the day-to-day monitoring, maintenance, and operations experience of cluster resources on a Storage Spaces Direct cluster.

I have reviewed and tested the Nano Server. Nano server is a lightweight software which is basically managed through the remote connection. In Windows Server 2016 I have created the Nano Server, described all the installation process, made some configurations and remote connection.

Containers is an isolate environment. In container administrator can run an applications without fear of changes. Containers grants a higher scale versus deploying an application to a VM. It is a good approach for developers. I have tested containers and tried to install Web Servers. First I chose nginx to test and later tested docker-compose. Finally, the last test was to run a Nano Server in a container. It takes only few steps to complete that. This makes containers handy and simple to use.

This thesis can be more developed by doing more difficult testings. It is also possible to make a research on how to manage containers on Nano Server. The review of how to transfer containers and to simulate environments can be tested.

REFERENCES

Berg James, 2016. Running #NanoServer in a #Container on Windows Server 2016 with #Docker in Powershell #DevOps. Available at:

<https://argonsys.com/learn-microsoft-cloud/library/running-nanoserver-in-a-container-on-windows-server-2016-with-docker-in-powershell-devops/> [Accessed 16 April 2018].

Buzdar Karim, 2017. How to Join a Nano Server with Domain. Available at:

<http://www.itprotoday.com/windows-8/how-join-nano-server-domain> [Accessed 12 April 2018].

Cady Adam, 2017. Linux Servers vs. Microsoft Windows Servers. Available at:

<https://www.singlehop.com/blog/linux-servers-vs-microsoft-windows-servers/>. [Accessed 17 March 2018].

Cattanach David, 2017. How Does Premium Assurance Differ From CSA?

Available at: <http://imageframe.co.uk/tag/microsoft/>. [Accessed 18 March 2018].

Chemistruck Dan, 2016. Step by Step Guide to Building Your Own Nano Server with Windows Server 2016 Technical Preview 4. Available at:

<https://www.agileit.com/news/step-by-step-guide-to-building-your-own-nano-server-with-windows-server-2016-technical-preview-4/>. [Accessed 19 March 2018].

Docker, 2018. Get started with Docker Compose. WWW document. Available at:

<https://docs.docker.com/compose/gettingstarted/#step-2-create-a-dockerfile> [Accessed 15 April 2018].

Elton Stoneman, 2017. How to Dockerize Windows Applications: The 5 Steps.

Available at: <https://blog.sixeyed.com/how-to-dockerize-windows-applications/> [Accessed 25 March 2018].

Finnegan Matthew, 2016. What's new in Windows Server 2016: Containers, Nano Server and licensing changes. Available at:

<https://www.computerworlduk.com/applications/windows-server-2016-what-expect-in-microsofts-latest-windows-server-operating-system-3623167/>.

[Accessed 17 March 2018].

Flores John, Poggemeyer Liza, John Tobin, 2017. Forest and Domain Functional Levels. Available at: [https://docs.microsoft.com/en-us/windows-server/identity/ad-](https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/active-directory-functional-levels/)

[ds/active-directory-functional-levels/](https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/active-directory-functional-levels/). [Accessed 21 March 2018].

Gerend Jason, Andrew Hansen, Liza Poggemeyer, wmgries, Garrett Watumull, 2017. What's new in Storage in Windows Server. Available at:

<https://docs.microsoft.com/en-us/windows-server/storage/whats-new-in-storage/>.

[Accessed 21 March 2018].

Jaimeo, Justinha, Poggemeyer Liza, 2017. What's New in Windows Server 2016. Available at: [https://docs.microsoft.com/en-us/windows-server/get-started/whats-](https://docs.microsoft.com/en-us/windows-server/get-started/whats-new-in-windows-server-2016)

[new-in-windows-server-2016](https://docs.microsoft.com/en-us/windows-server/get-started/whats-new-in-windows-server-2016) [Accessed 17 March 2018].

Jaimeo, Justinha, Poggemeyer Liza, 2017. Install Nano Server. Available at:

[https://docs.microsoft.com/en-us/windows-server/get-started/getting-started-with-](https://docs.microsoft.com/en-us/windows-server/get-started/getting-started-with-nano-server)
[nano-server](https://docs.microsoft.com/en-us/windows-server/get-started/getting-started-with-nano-server) [Accessed 18 March 2018].

Jaimeo, Poggemeyer Liza, Nysten Chris, Plett Corey, 2017. Deploy Nano Server.

Available at: [https://docs.microsoft.com/en-us/windows-server/get-started/deploy-](https://docs.microsoft.com/en-us/windows-server/get-started/deploy-nano-server/)
[nano-server/](https://docs.microsoft.com/en-us/windows-server/get-started/deploy-nano-server/). [Accessed 19 March 2018].

Jones Mike, Cesar De la Torre, Wenzel Maira, Anderson Rick, Latham Luke, 2017. What is Docker? [https://docs.microsoft.com/en-](https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/container-docker-)

[us/dotnet/standard/microservices-architecture/container-docker-](https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/container-docker-introduction/docker-defined)
[introduction/docker-defined](https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/container-docker-introduction/docker-defined) [Accessed 18 March 2018].

Lich Brian, Justinha, 2017. Protect derived domain credentials with Windows Defender Credential Guard. <https://docs.microsoft.com/en-us/windows/access-protection/credential-guard/credential-guard/>. [Accessed 21 March 2018].

Maurer Thomas, 2016. How to Disable and configure windows defender on Windows Server 2016 Using PowerShell. Available at: <https://www.thomasmaurer.ch/2016/07/how-to-disable-and-configure-windows-defender-on-windows-server-2016-using-powershell/>. [Accessed 20 March 2018].

McCabe John, 2016. Containers. *Introducing Windows Server 2016*, 97–102.

Microsoft, 2016. Nano Server. *Introducing Windows Server 2016*, 89–96.

Otey Michael, 2015. Top Ten: What You Need to Know about Microsoft Nano Server. Available at: <http://www.itprotoday.com/windows-8/top-ten-what-you-need-know-about-microsoft-nano-server/>. [Accessed 20 March 2018].

Posey Brien, 2016. What's new in Windows Server 2016: Containers, Nano Server and licensing changes. Available at: <http://searchservirtualization.techtarget.com/tip/Windows-Server-2016-Hyper-V-Manager-comes-with-a-few-improvements/>. [Accessed 17 March 2018].

Posey Brien, 2017. Differences between Windows Server Containers, Hyper-V Containers and VMs. Available at: <http://searchservirtualization.techtarget.com/tip/Differences-between-Windows-Server-Containers-Hyper-V-Containers-and-VMs/>. [Accessed 20 March 2018].

Rouse Margaret, 2015. Microsoft Nano Server. Available at: <http://searchwindowsserver.techtarget.com/definition/Microsoft-Nano-Server> [Accessed 17 March 2018].

Roussey Benjamin, 2017. Top new hyper-V features in Windows server 2016. Available at: <http://techgenix.com/hyper-v-features-windows-server-2016/>. [Accessed 18 March 2018].

Rubens Paul, 2017. What are containers and why do you need them? Available at: <https://www.cio.com/article/2924995/software/what-are-containers-and-why-do-you-need-them.html> [Accessed 20 March 2018].

Savill John, 2015. The differences between Windows Containers and Hyper-V Containers in Windows Server 2016. Available at: <http://www.itprotoday.com/windows-8/differences-between-windows-containers-and-hyper-v-containers-windows-server-2016/>. [Accessed 20 March 2018].

Stroud Forrest, 2017. Nano Server. Available at: <https://www.webopedia.com/TERM/N/nano-server.html/>. [Accessed 17 March 2018].

Tang Ryen Kia Zhi, 2017. Nano Server: Getting Started in Container with Docker. Available at: https://social.technet.microsoft.com/wiki/contents/articles/38652-nano-server-getting-started-in-container-with-docker.aspx#Preparation_on_Windows [Accessed 17 March 2018].

Vilcinskas Markus, Tillman Mike, 2017. Introduction to device management in Azure Active Directory. Available at: <https://docs.microsoft.com/en-us/azure/active-directory/device-management-introduction/>. [Accessed 21 March 2018].

Vilcinskas Markus, Moreau Erik, sdabbiru, Az Balaji, Tillman Mike, 2018. How to configure hybrid Azure Active Directory joined devices Introduction to device management in Azure Active Directory. Available at: <https://docs.microsoft.com/en-us/azure/active-directory/device-management-hybrid-azuread-joined-devices-setup/>. [Accessed 21 March 2018].

Vilcinskas Markus, Tillman Mike, el-Melhaoui Karim, Mathers Bill, femila, 2018. Enable Microsoft Windows Hello for Business in your organization. Available at: <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-azureadjoin-passport-deployment/>. [Accesed 24 March 2018].

Warner Tim, 2016. 10 Best New Features in Windows Server 2016. Available at: <http://www.tomsitpro.com/articles/best-windows-server-2016-features,2-1061.html>. [Accesed 28 March 2018].

Wrock Matt, 2016. Installing and running a Chef client on Windows Nano Server. Available at: <https://matt-wrock.squarespace.com/?offset=1462120748617/>. [Accesed 19 March 2018].

App.py code

```
import time
import redis
from flask import Flask
app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)
def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)
@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)
if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)
```

(Docker 2018.)