Windows Server 2003 End of Support Application Migration

Options for modernizing your corporate infrastructure

Jason Masterman
Distinguished Engineer, Neudesic

Table of Contents

Table of Figures

Abstract

This whitepaper is intended for IT professionals, developers and architects who need to address the migration of applications off Windows Server 2003 as it reaches its end of support in July of 2015. Many companies are using this milestone as an opportunity to not only move from Windows Server 2003 but to also virtualize their workloads with Amazon Web Services and reduce costs in the process.

Introduction

Microsoft Windows Server 2003 **reaches the end of extended support on July 14, 2015**[1]. What many don't realize is that mainstream support ended over four years ago. Businesses will have enjoyed a five-year runway leading to the final support date later this year. Unfortunately, 75% of organizations surveyed have not migrated, nor do they have a plan to deal with this issue[3]. According to analysts there are approximately 12-13 million Windows Server 2003 servers still in production[4].

Clients ask, "What does it really mean when a product runs out of support?" Contrary to what some may fear, end of support does not render the operating system inoperable or cause a catastrophic stop come July 14, 2015. Rather, end of support means[5]:

No security updates - 37 critical updates were released in 2013 for Windows Server 2003 R2 under extended support. No updates will be developed or released after end of support.

No regulatory compliance - Lack of compliance with various standards and regulations can be devastating. In some industries, this may result in fines being levied on organizations.

No application support - Many applications will cease to be supported once the operating system they are running on is no longer supported. This includes all Microsoft applications.

Given the frequent discovery of new attack vectors and the constant threat of attacks by hackers, the lack of updates released by Microsoft should become a major consideration. Companies within regulated industries may face very expensive alternatives as they attempt to manage these servers in the coming years.

Clients who plan to continue running Windows Server 2003 past the end of support window can purchase a Microsoft custom support contract. However, it is important to understand a few key things in regard to these contracts in order to plan accordingly. First, these custom support contracts cannot be purchased if the client does not already have a Premier Support contract in place. Second, these support contracts start at several hundred thousand dollars per year and may be much, much higher depending on complexity.

[1] http://support.microsoft.com/lifecycle/search/default.aspx?alpha=Windows+Server+2003+R2&wa=wsignin1.0

[2] AppZero - "State of Readiness for Windows Server 2003 End of Support" Survey

[3] http://www.appzero.com/sites/default/files/white-papers/docs/AppZero_State_of_Readiness_for_WS2003_End_of_Support_Survey_Results._0.pdf

[4] http://www.directive.com/blog/item/the-time-to-upgrade-windows-server-2003-is-now.html

[5] http://blogs.technet.com/b/uktechnet/archive/2014/06/25/are-you-ready-to-migrate-windows-server-2003-end-of-life-is-coming-on-the-july-14th-2015.aspx

Additionally, in order to purchase a support contract the client must have a plan in place that illustrates how they will be moving off the platform (the support contract is not a permanent solution; it simply allows businesses to postpone the move). Finally, the servers running Windows Server 2003 and requiring support are most likely running either custom applications or third-party applications. Many third-party vendors have policies stating that they will not support their applications on unsupported operating systems. This makes the purchased custom support contract with Microsoft of no value to the application needing support.

Given the large number of servers running Windows Server 2003, and the limited time left to mitigate the issue, companies must quickly determine how they will manage the risk. With so little time left, many organizations will undoubtedly be running some number of unsupported servers post-July 14, 2015. The goals for these companies should be the following:

- Prioritize all applications and servers, with emphasis on those that pose the greatest risk to the business should they fail and be unsupported.
- Determine a series of options, best practices and plans to migrate the highest risk applications first. No single plan will work for all scenarios; take advantage of all options.
- Start implementing the plan as quickly as possible to minimize the amount of exposure and risk posed to the business.

The remaining sections will focus on the goals above and how to achieve them. Additional focus on cloud options will also be covered as this was a significant area of interest and focus for clients surveyed. According to a recent survey, 35% of clients will be migrating these applications to the cloud and an additional 40% were unsure but may choose cloud as the option. With 75% of clients potentially migrating Windows Server 2003 servers to the cloud, it is important to help clients understand the process, options and cost-saving opportunities in doing so.

Windows Server 2003 Mitigation and Migration Options

There are several Windows Server 2003 mitigation options available for clients. For example, one option is to continue running the old servers but to isolate them from other servers or internet traffic. For completeness, let's take a brief look at mitigation options and then turn our focus to the technical options for migration. Many mitigating options require no technical action and the primary focus for the remaining sections is to help with specific tools, best practices, cost considerations and approaches to migrating.

Mitigation Options

Much has been written on the topic of Windows Server 2003 end-of-support mitigation options. To summarize, these are the five most prevalent mitigation options available:

1. Purchase a custom support agreement from Microsoft while you continue to migrate away from the platform.

2. Isolate the servers from other servers in your data center (e.g., separate subnets) and set up additional monitoring to watch for potential issues and prevent attacks from reaching other servers.

3. Sunset existing applications and simply retire these servers and applications.

4. Leave the servers running, taking no immediate action. When something happens causing the application to fail or require attention, implement a pre-defined plan at that point in time.

5. Where possible, find replacement applications, roll out replacement applications and then retire the older system.

Each of the above options has significant differences in approach, cost and risk. While performing the necessary inventory, assessment and plan for each of the applications in question, organizations will likely find that a mix of the above mitigation options will be viable as they define prioritization and risk.

Migration Options

There are several options available in terms of migration. Migrations can be performed using either a manual approach or tools to help automate the approach. Additionally, the destination can be one of a variety of locations. These options include the following manual approaches:

Windows Server Installation Upgrade - Uninstall the application from the existing Windows Server 2003, upgrade the OS to a more recent version of Windows Server, install and configure the application on new OS.

- New server migration - Spin up new server infrastructure and install the old application on a more recent version of Windows Server.

- Cloud migration - Leverage Amazon Web Services to spin up an instance on a more recent version of Windows Server and install the application there.

- Physical to virtual - Migrate P2V (physical to virtual) with the assumption that WS2003 may be currently running on physical hardware but virtually after the upgrade. Keep in mind this virtual platform may be in a private data center or in the cloud.

- Virtual to virtual - Migrate V2V (virtual to virtual) by moving your existing virtual Windows Server 2003 to a newer virtual platform. Again, keep in mind that this could be in a private data center or in the cloud.

In each approach above, it will be very important to test the application in detail once it is running on the newer version of Windows Server. This topic will be addressed in more detail in the section on testing later in this paper. However, it is important to consider the numerous security enhancements and changes to the operating system that can impact how well an older application will run with default settings on a new version of Windows Server.

Hardware Considerations

There are some important considerations with Windows Server 2003 in regards to hardware platform. The x64 architecture, which is commonplace today, is not supported by Windows Server 2003. Instead the primary 64-bit option was previously the Intel Itanium architecture which is still sold, but Microsoft phased out support for it with Windows Server 2008. When people speak of 64-bit support today, they are referring to the x64 architecture. According to Microsoft, this "is a backwards-compatible extension with x86.
It provides a legacy 32-bit mode, which is identical to x86, and a new 64-bit mode. The term 'x64' includes both AMD 64 and Intel64. The instruction sets are close to identical."[6] This means that some Windows Server 2003 machines may be running on Intel Itanium 64-bit processors which have a different instruction set than the x64 bit instructions. Though this is a consideration, most companies continued with the 32-bit versions of the platform and ran 32-bit versions of Windows Server 2003.

Windows Server 2008 supported all of the above[7], and many organizations that had existing 32-bit machines moved to Windows Server 2008 32-bit, while others purchasing new hardware typically opted for the x64 platform.

But Windows Server 2012 only supports x64. This means that those looking to upgrade Windows Server 2003 most likely will be moving from x32 to x64-based machines, regardless of whether those machines are physical boxes in the client's data center, virtual platforms, or cloud-based infrastructure.

Overview of Automated Approach

Migrating applications using the manual approaches listed previously will have many significant challenges which we'll discuss in detail. However, there are also a few fully-automated approaches available. Interestingly, there is a huge movement in the open source and Linux world working to solve this problem. Its goal is to make applications easily portable across underlying operating systems by packaging them in a standard way, much like containers are moved in the shipping industry. If you have not heard of Docker[8], it is well worth looking into in order to understand how others are working to solve the problem of application portability. Our focus is on Windows machines, and in this arena one company stands out: AppZero[9]. With AppZero it is possible to:

- Automate the packaging of an existing application for deployment to any server regardless of location, whether on premises or in the cloud
- Extract existing applications, configurations and dependencies even without the original installation program
- Move the application as an application image, called a VAA or Virtual Application Appliance, not as a machine image or virtual machine

The AppZero technology can automate many application migrations that would otherwise only be possible through manual and error-prone steps. We will look at this process in detail in the following sections.

[6] http://msdn.microsoft.com/en-us/library/windows/hardware/ff561499(v=vs.85).aspx

[7] http://social.technet.microsoft.com/wiki/contents/articles/3556.differences-between-the-editions-of-windows-server-2008.aspx

[8] https://www.docker.com/

[9] http://www.appzero.com/

Migration Approaches in Detail

Now that we have an idea of the various approaches and a good understanding of the hardware considerations, we can begin looking at each approach more closely. We'll also discuss some common challenges typical with each approach and ways to optimize any of the approaches taken.

NOTE: Before you migrate, proper backup procedures must be in place and a recent backup of the OS and application must be performed. If this is not the case, or if you are unsure, do not proceed until a solid backup has been created and verified for restoration.

Windows Server Installation Upgrade

At first glance, a server software upgrade (as opposed to migration) often appears to be a convenient option. A typical upgrade with most software is performed by obtaining the new version and running the installation/upgrade process. However, it is important to first take a close look at the supported paths from Microsoft, since some paths are not supported.[10] In particular, there is no direct path supported from Windows Server 2003 to Windows Server 2012. There is direct support for certain situations to Windows Server 2008.

A snapshot of the supported direct upgrade paths from the Microsoft TechNet site shows the following:

From Windows Server 2003 (SP2, R2)	Upgrade to Windows Server 2008 R2
Datacenter	Datacenter
Enterprise	Enterprise, or Datacenter
Standard	Standard, or Enterprise

Table 1: Upgrade Paths from Window Server 2003

The following are excellent guides from Microsoft that you should review prior to migration:

- "Guide for Upgrading to Windows Server 2008 R2"; http://technet.microsoft.com/en-us/library/ff972408(v=ws.10).aspx

- "Known Issues When Upgrading to Windows Server 2008 R2"; http://technet.microsoft.com/en-us/library/ff972310(v=ws.10).aspx

When the upgrade must be done on existing hardware that is running the old OS, the best results are usually achieved by performing a clean installation. With a clean installation the new version is installed over the old version, and deletes the old version. Given this scenario, any applications currently running on the old version are no longer available in the new version and must be reinstalled and configured. The act of reinstalling old software may be easy for some organizations but very difficult for others. At any rate, there is no simple rollback option, and the application is out of commission until it is reinstalled.

A Common Challenge

One of the most common challenges facing any of the manual migration approaches above is the non-existence of application installation media and configuration details. For many clients, these systems have been running for over a decade and vendors have gone out of business, the developers who wrote the applications may have left the company, or useful information about the application is simply not available. This conundrum makes this particular upgrade approach a near impossibility or too risky to attempt on hardware that is already in place. A better approach would be to leave this server in place, install a new server and migrate the application to the new server, as discussed in the following sections.

[10] http://technet.microsoft.com/en-us/windowsserver/dn527667.aspx

New Server Migration

With so many new features available in Windows Server 2012 and the significant hardware transition from 32-bit to 64-bit that most clients are facing, a common choice among many clients will be to choose new server hardware as the destination for migrating applications from Windows Server 2003. This gives clients many options for the location of the new server. In particular, this new server could even be infrastructure in the cloud. However, in this section we will focus on an actual physical server.

For clients choosing new physical servers running a new version of Windows Server, whether it's 2008 or 2012, the process really isn't that much different, but risk is lowered as the existing machine can continue running while the new machine is set up to perfection. In this scenario, time is usually the critical factor, as hardware purchases typically take time to approve. There are also lead times for delivery, physical configuration in the data center, etc. This time aspect may further complicate the schedule for migrating the many applications within the organization. Once the hardware is available, the remaining tasks, centering on application setup and configuration, are fairly straightforward.

New Server Configuration

New server configuration focuses on leveraging hardware that is either on premises or in the data center. The goal of this approach is to build out a new machine with a more recent release of Windows Server, such as 2008 or 2012, and then install the application onto the new server machine.

First, install and configure the new operating system, either Windows Server 2008 or Windows Server 2012, making certain to set appropriate features and roles for the server. Next, install and configure the application that is migrating to the new server. The typical migration challenges are relevant here, so take great care in copying application files, configuration information, documentation, etc.

Once the application is installed on the new OS, the key step is testing and verifying that everything works as it did previously. If the application runs on the new installation, a big portion of the battle has been won. Verifying that the application behaves as expected across all scenarios is a critical step in ensuring that the application truly meets expectations. If this application doesn't already have a set of integration tests, determine whether there is sufficient time and resources to set some up before the migration. If the application has a medium or high business impact, this could be critical.

The following list provides links to some relevant documents to aid in testing and fixing common application issues. Given the breadth of information available, we won't rehash all of the content here. Notice that the focus is primarily on migrating to Windows Server 2008, not Windows Server 2012. However, much of the Windows Server 2008 content is still relevant to Windows Server 2012. The guidance for upgrading Windows Server 2003 is focused on supported upgrade paths—there is little guidance to show which issues to expect in terms of application considerations when skipping a version.

- "Application Considerations When Upgrading to Windows Server 2008 R2"; http://technet.microsoft.com/en-us/library/ff972309.aspx
- "Migrate Roles and Features to Windows Server"; - http://technet.microsoft.com/en-us/library/jj134039.aspx
- "Install, Use, and Remove Windows Server Migration Tools"; - http://technet.microsoft.com/library/jj134202
- "Windows and Windows Server compatibility cookbook: Windows 8, Windows 8.1, and Windows Server 2012"; http://www.microsoft.com/en-us/download/confirmation.aspx?id=27416

Cloud Migration

The previously-mentioned migration options have focused solely on hardware data center approaches. This section focuses on the opportunity to leverage cloud-based resources for supporting Windows Server 2003 migrations to the Amazon Web Services cloud. We will look at several factors, including the ever-important security questions, costs analysis, and scalability.

The Security Question

There are several very important security questions to address with clients in regard to moving server workloads to the cloud. It is only natural that organizations have a high level of concern about how their data is going to be protected, and will want to know which processes, policies, certifications, and proof are available to better understand how seriously AWS approaches this.

A good place to start is the security landing page at AWS (http://aws.amazon.com/security/). There you can learn how AWS follows a "shared-responsibility" model for security. In other words, AWS has secured the underlying infrastructure and you must secure anything you put on the infrastructure (e.g., client-side data encryption). The page lists several layers of built-in security features.

- "The AWS cloud infrastructure has been architected to be one of the most flexible and secure cloud computing environments available today. It provides an extremely scalable, highly reliable platform that enables clients to deploy applications and data quickly and securely."[11]

Depending on your industry, you may also need to review the AWS compliance page[12], which lists up-to-date compliance certifications and industry-standard assurances. You are encouraged to request a compliance report certification to better understand the level of compliance provided by AWS. This report can be requested at http://aws.amazon.com/compliance/contact/.

Here is a list of some of the controls and certifications in place today, as described on the AWS website:

- "AWS Service Organization Control (SOC) Reports are independent third-party examination reports that demonstrate how AWS achieves key compliance controls and objectives. The purpose of these reports is to help you and your auditors understand the AWS controls established to support operations and compliance."[13]

- "AWS Payment Card Industry (PCI) DSS Level 1 Data Security Standard (DSS) is an information security standard defined by the Payment Card Industry Security Standards Council. PCI certification is required for organizations (merchants and service providers) that process credit card payments. The certification is designed to prevent credit card fraud through increased controls around data and its exposure to compromise."14

- "AWS ISO 27001 is a security management standard that specifies security management best practices and comprehensive security controls following the ISO 27002 best practice guidance. This is a widely-recognized international security standard in which our clients showed significant interest."15

- "AWS FedRAMP, The Federal Risk and Authorization Management Program (FedRAMP) is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. FedRAMP is mandatory for Federal Agency cloud deployments and service models at the low and moderate risk impact levels."[16]

[11] http://aws.amazon.com/security/

[12] http://aws.amazon.com/compliance/

[13] http://aws.amazon.com/compliance/soc-faqs/

[14] http://aws.amazon.com/compliance/pci-dss-level-1-faqs/

[15] http://aws.amazon.com/compliance/iso-27001-faqs/

[16] http://aws.amazon.com/compliance/fedramp-faqs/

- "AWS DoD CSM, The Department of Defense (DoD) Cloud Security Model (CSM) provides a formalized assessment and authorization process for cloud service providers (CSPs) to gain a DoD Provisional Authorization, which can subsequently be leveraged by DoD clients. A Provisional Authorization under the CSM provides a reusable certification that attests to our compliance with DoD standards, reducing the time necessary for a DoD mission owner to assess and authorize one of their systems for operation on AWS."[17]

The more you learn about how seriously AWS works to ensure client security is a top priority, the more confident you will become that any concerns or questions around this topic can be quickly answered. Many clients find that the controls and measures in place at AWS far exceed what most organizations could practically afford implementing in their own private data centers.

Planning for Migration to Cloud

Clients always ask, "How much does it cost?" and as any good consultant will always answer, "It depends!" Before we go into how to determine the answer to the cost question, an even more important question any client should ask is, "How do I properly prepare and plan for migrating to the cloud?" In this section we will provide you with the tools necessary to properly plan for migrating to the cloud.

The most important aspects of planning for any migration to the cloud is to understand the benefits and options that factor into improving application reliability, availability, performance, security and overall costs. It is important to look at the whole picture when migrating an application to the cloud and not just focus on a typical forklift approach. Some clients start their cloud experience by trying to take an existing application running on a particular piece of hardware and simply lifting it up and setting it back down on a like-for-like server in the cloud. This is often called the forklift approach as the application is being lifted up and moved, much like a forklift would lift a pallet of goods and move them to a new location.

Typical Sizing Mistakes

In some cases the forklift approach is adequate, but generally speaking it has some significant drawbacks. The first major drawback is that clients will simply look at their existing hardware configuration and just replicate that in the cloud. For example, suppose the current server has 32GB of RAM, 4 CPUs with a certain specification and 200GB of storage. Let's stop and consider that existing piece of hardware and how it was given the sizing and configuration it currently has. In most on-premises scenarios, applications are typically sized to meet peak demand while also allowing headroom for growth.

The peak load of a particular application running on that server may only be hit for an hour or less per day. Even more commonly, the server might not be used at all during the evening hours when employees have gone home. In short, this server might only be used eight hours per day. In most companies, this server would be turned on and never turned off. This is drastically different from how cloud services can be configured—in fact, rethinking this configuration will bear heavily on how costs are calculated.

Expanding on the forklift analogy, if a company brings over the application and runs it on the same size server and runs that server all day, every day, they are paying for a significant amount of services that they are not actually using. This can be avoided by selecting the appropriate Amazon Elastic Compute Cloud (Amazon EC2) instance size for your Windows Server, and turning it off when not in use.

Scaling Up and Down

In the planning phase it is important to determine exactly how a server is used and what resources are really needed. The goal is to reduce the infrastructure being paid for in the cloud and minimize the amount of time services are running to only the period needed. Additionally, if possible, the server resources should scale up when needed and scale back down when not needed. This could mean, "If I don't need that huge server for more than one hour of peak usage a day then perhaps I can spin up one or more smaller instances as needed and spin them back down when not needed."

Determining Resource Usage and Cost

The next important question is, "How does one determine the resource needs of a particular application and server?" There are some standard metrics that most operating systems can make available if one knows how to look for them. In the case of Windows Server 2003, the System Monitor tool can be used to watch the built-in performance counters. This tool can be configured to log important metrics such as CPU, memory, disk, network and other factors to help determine peak periods during the day and/or month. A good guide on the topic is available on the Microsoft Support site: http://support.microsoft.com/kb/248345.

[17] http://aws.amazon.com/compliance/dod-csm-faqs/

In addition to System Monitor, there are other third-party applications that can help, not only in determining resource usage of existing environments but in mapping those resource needs to specific instance types in Amazon EC2. One company, Cloudamize[18], provides the ability to monitor existing workloads in your on-premises environment, and then leverage this information to make recommendations for running these applications on AWS. This can help clients find the right blend of resources and reduce infrastructure costs while running in the cloud. In the case of new clients, a small component is installed on existing application servers. This tool collects data over a period of time, preferably a month's worth of data, and then produces a report to suggest proper sizing and mapping onto the AWS platform. For clients who may already have workloads running in the cloud, the tool can also be used to continue monitoring those workloads to further improve the configuration and resource usage to reduce costs on the AWS platform.

The AWS Simple Monthly Calculator[19] can also be used to help plan for and estimate the costs associated with running on AWS. This tool, coupled with the output of a Cloudamize report, can really assist organizations in determining proper sizing and estimated costs when migrating workloads to the cloud.

Connectivity

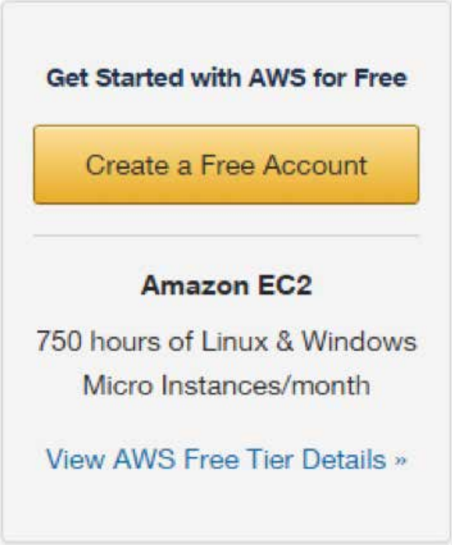
How does one access resources running on the AWS cloud platform? Amazon Machine Images (AMIs) running in the cloud can be accessed in a variety of ways depending on the application type, security concerns, and needs of the application. One of the more common ways to access resources in the cloud is to simply open a VPN connection to the resource and then utilize the machine via a direct connection. However, there are other options for web applications and applications that connect to other machines on AWS, or that still access data living on premises.

AWS Direct Connect[20] provides the ability to set up a private direct connection from an organization's existing location or data center directly to AWS. This option is of huge benefit for multiple reasons. From an application perspective, the benefit is that AWS resources appear as if they are simply part of the existing corporate network—without changing a single line of code. From a user perspective there is no difference when accessing the application. Another advantage is the ability to further reduce network costs and increase bandwidth while also simplifying how services are accessed in the cloud.

Actual Application Migration

Let's pull our focus back to the issue at hand, moving the application to a server running on AWS.

Standing up a Windows Server 2008 or Windows Server 2012 instance to host an existing Windows Server 2003 application in AWS is only a few mouse clicks away. All that is needed is an AWS account. To set this up, visit http://aws.amazon.com/ and select the Create a Free Account button shown in Figure 1.



Get Started with AWS for Free

Amazon EC2

750 hours of Linux & Windows Micro Instances/month

Figure 1: AWS Account Setup

Once an account is set up, sign into the account and go to the AWS Management Console to create a new instance in the region desired. The example below shows the N. Virginia region selected in the upper-right corner.

[18] http://www.cloudamize.com/company/

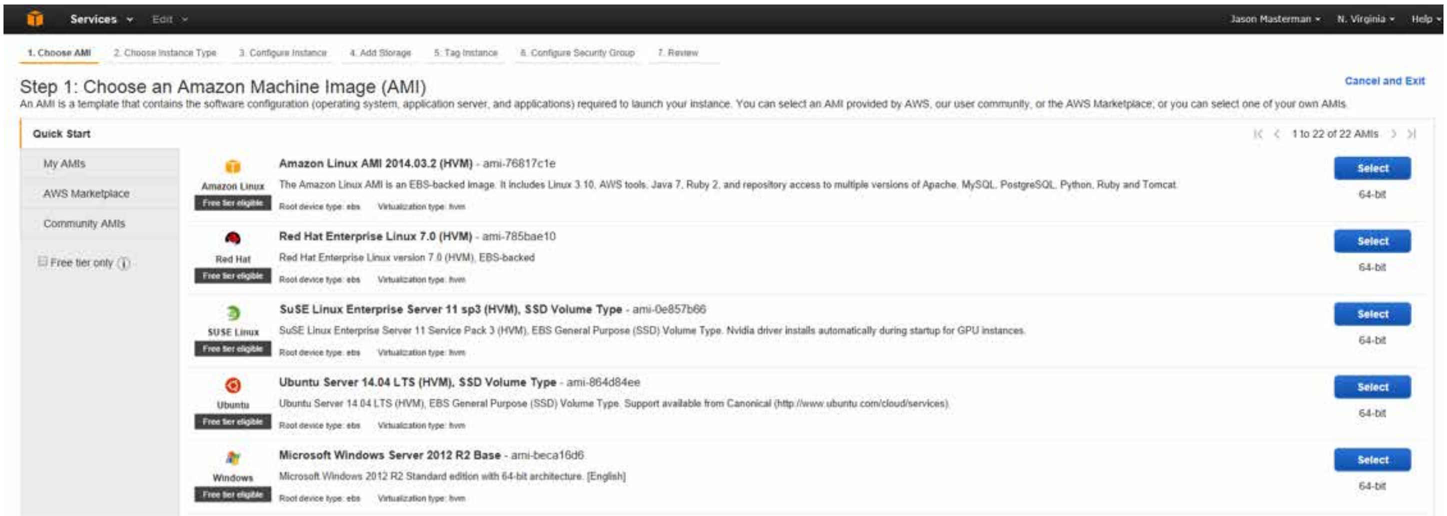
[19] http://calculator.s3.amazonaws.com/index.html

[20] http://aws.amazon.com/directconnect/

After you sign in to the AWS Management Console, you can launch a new Amazon EC2 instance by choosing EC2 in the Compute section. For step-by-step instructions, see the following guide:

"Getting Started with Amazon EC2 Windows Instances"; http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2Win_GetStarted.html

When launching an instance, you can choose from a list of available AMIs for Windows Server. The following illustration shows the Windows Server 2012R2 base images available. There are also AMIs for Windows Server 2008, instances with SQL Server already installed, and other configuration options. What is important to see is how easy it is to obtain a running instance of Windows Server.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

☐ Free tier only

Amazon Linux AMI 2014.03.2 (HVM) - ami-76817c1e
The Amazon Linux AMI is an EBS-backed image. It includes Linux 3.10, AWS tools, Java 7, Ruby 2, and repository access to multiple versions of Apache, MySQL, PostgreSQL, Python, Ruby and Tomcat.
Root device type: ebs   Virtualization type: hvm

Red Hat Enterprise Linux 7.0 (HVM) - ami-785bae10
Red Hat Enterprise Linux version 7.0 (HVM), EBS-backed
Root device type: ebs   Virtualization type: hvm

SuSE Linux Enterprise Server 11 sp3 (HVM), SSD Volume Type - ami-0e857b66
SuSE Linux Enterprise Server 11 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Nvidia driver installs automatically during startup for GPU instances.
Root device type: ebs   Virtualization type: hvm

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-864d84ee
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs   Virtualization type: hvm

Microsoft Windows Server 2012 R2 Base - ami-beca16d6
Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]
Root device type: ebs   Virtualization type: hvm

Figure 3: Amazon Machine Image Selection

Installing or Migrating the Application

Once the new Windows Server instance is running on AWS the next step is to migrate the application to run on the new instance. We will discuss two approaches to getting a Windows Server 2003 application moved to the new Windows Server 2012 instance.

Application Installation

The first approach is no different than our earlier example of installing the application from existing installation media, configuration files and documentation. This assumes that the information is readily available. For some organizations this will be a very easy process and once the application is installed and configured, it is a matter of testing to ensure everything is working as expected.

Application Packaging

The real challenge to migrating applications involves applications for which there is no installation media or documentation, or the vendor is out of business, or the developers are long gone. These applications can be much more difficult to migrate to a new server. Determining which files comprise the actual application, which registry settings, dependent assemblies, and DLLs are needed, and where the configuration files are can be very challenging. In these scenarios, we suggest that organizations look at a technology provided by AppZero and leverage their tools for packaging up the existing application and moving it to the new server running on AWS.

The following diagram shows the AppZero process:

AppZero Does for Applications What VMs Do for Machines

AppZero extracts *existing* enterprise applications and packages them into VAAs that can be provisioned and run *natively* on any OS, machine or cloud, fast (including managed clouds), rapidly.

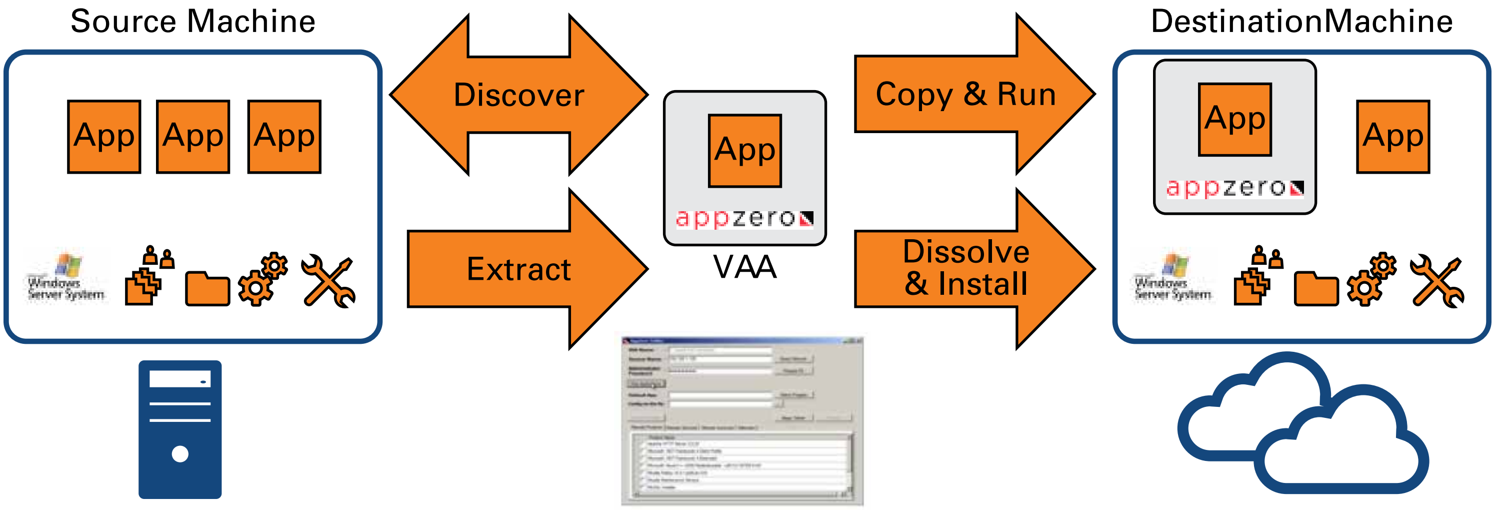


Figure 4: AppZero Application Extract, Package and Deploy Migration

Let's take a deeper dive into the steps portrayed in the diagram.   Note that AppZero does not require tools or files to be installed onto the source machine. The only files installed are on the destination machine. There are only a few high-level steps in the process:

- Application components are identified via discovery on the source server.
- Application files, components, and dependencies are extracted.
- Application components are encapsulated into a VAA.
- The application is copied and deployed to the target machine, but the VAA is maintained.
- While the application is running on the target machine, if a file is needed that is not part of the VAA, a connection is maintained to the source machine, and the requested file is extracted and brought into the VAA.
- Once the application has been tested and is running without further file requests, the VAA package is dissolved, leaving the application installed on the target machine just as if it were natively installed.
- The application now runs natively on the target machine.
- If needed, the application could again be packaged back into a VAA and moved to another server.

Note that the connection can be maintained if desired, and this gives AppZero the ability to continue monitoring an application in the early stages of migration testing. While the application is running in the VAA, any files or missing pieces can be detected and pulled over to the destination machine until all dependencies are found and the application

is confirmed to be running without issues. This is very powerful technology and provides a convenient way to move an application from one machine to another.

Applications that will not work with AppZero:

- Microsoft Exchange Server (any version)
- Microsoft SharePoint Server (any version)
- 16-bit applications
- Microsoft Internet Information Services (IIS), any version other than 2000 (IIS5)
- Lightweight Directory Access Protocol (LDAP) and Active Directory
- System level tools such as debuggers

AppZero Service Provider (SP) Edition

AppZero SP edition is a service that tightly integrates the migration automation lifecycle within a service provider cloud. This service is currently available on AWS. The service is also provided at no cost to clients[21].

Physical to Virtual (P2V) and Virtual to Virtual (V2V)

Clients running on physical servers today who wish to migrate to virtual server technology, either running in a data center or in the cloud, have several tools at

their disposal. Unfortunately these won't help with the migration from Windows Server 2003. Given the primary goal of moving away from Windows Server 2003, simply moving the system off a physical machine to a virtual machine (P2V) won't mitigate the underlying problem. For this reason we recommend using one of the approaches discussed previously.

For clients running existing Windows Server 2003 servers as virtual machines, the challenge is the same as P2V scenarios. Moving the virtual machine alone won't solve the problem. There are tools that can take existing VMs and import them into Amazon EC2 instances on the AWS cloud, and for some clients this may be a good first step to cloud- based infrastructure. You may be able to yield a real migration and mitigation plan by combining a VM's import into AWS with one of the approaches we discussed in previously for migrating the application to a new instance running an updated version of Windows Server.

[21] https://aws.appzero.com/

Additional Considerations

There are a couple of additional items to consider when looking at a Windows Server 2003 mitigation and migration plan. One consideration is the opportunity for application consolidation during migration. It may be prudent to consolidate multiple applications onto a single new instance of Windows Server 2008 or 2012. This can offer additional benefits in terms of reducing the number of machines, licenses, and costs associated with running and maintaining multiple boxes. This consolidation should not be performed without additional testing to ensure there is no negative interaction with the applications as they share resources on a single machine. However, with the many improvements in terms of process isolation over the years, Windows Server has proved to be a solid platform for running multiple applications on a single machine. Companies are encouraged to assess their application migrations while also looking to consolidate where possible. And keep in mind that in the cloud, higher CPU utilization is sometimes a good thing. Instead of paying for idle CPU resources, you can automatically scale up and down as needed to ensure that you always pay for servers that are just busy enough to provide the responsiveness that your users expect.

Conclusion

Windows Server 2003 end of support brings the opportunity for organizations to look at many options for migrating applications onto newer versions of Windows Server. Client mitigation plans must have a complete strategy for mitigating the risks, and taking advantage of the options and cost saving opportunities. Clients looking to leverage the cloud have additional opportunities to scale migrated applications up and down on demand and consolidate them while also running those applications with improved and robust security standards.

neudesic™

Make your move to the cloud

303.248.8321