aws EMP for Windows Server

AWS End-of-Support Migration Program (EMP) for Windows Server

Technical whitepaper version 2.1

Contents

1 Introduction 2
1.1 Objective 2
1.2 AWS EMP for Windows Server 2

2 EMP Compatibility Packager 3
2.1 Overview 3
2.2 Supported Operating Systems 3
2.3 Features 3

3 Application Compatibility Package 5
3.1 Package Overview 5
3.2 Moving N-Tier Server Applications to the Cloud 6

4 Features 7
4.1 Isolation 7
4.2 Redirection 7
4.3 Compatibility 7
4.4 Compatibility with Application Virtualization Solutions 8

5 Conclusion 9

1 Introduction

1.1 Objective

This whitepaper provides a technical overview of concepts and features associated with the AWS End-of-Support Migration Program (EMP) for Windows Server. Please contact your AWS representative or contact for specific solutions design or guidance for implementation.

1.2 AWS EMP for Windows Server

The AWS End-of-Support Migration Program (EMP) for Windows Server is a program that offers the technology and expert guidance to migrate your applications running on Windows Server 2003, 2008, and 2008 R2 to the latest, supported versions of Windows Server on AWS. Through EMP technology, critical applications are decoupled from the underlying OS so that they can be migrated to the supported version of Windows Server on AWS without code changes.

Many organizations run legacy applications on versions of Windows Server that are at or nearing end of support (EOS). You may get extended support for a limited amount of time, but this does not resolve the problem. After an EOS date, Microsoft offers only “critical” and “important” security updates for Windows Server. They stop sending operating system-wide security improvements, which leaves your systems open to security vulnerabilities.

To get regular security updates and to keep your system compliant after the limited extended support period expires, you must migrate to a later version of Windows Server, which may require refactoring of legacy applications. For applications with tight dependencies on out-of-support operating systems, missing installation media or code, or limited in-house expertise, it might appear that your only option is to continue using an unsupported version of Windows Server. However, continuing to use an unsupported version of Windows Server leads to security and compliance concerns. EMP is designed to mitigate the challenges of this scenario.

EMP subject matter experts (SMEs) from AWS Professional Services and AWS Partners utilize the EMP Compatibility Packager tool to create compatibility packages for legacy applications. These packages include a unique redirection and compatibility engine that makes it possible to migrate legacy applications from unsupported, non-compliant operating systems to Windows Server 2019/2016 running on AWS.

The end-to-end migration process followed by an EMP expert is as follows:

Phase 1: Application Assessment

Application requirements and testing criteria are presented to AWS SMEs to ensure robust packaging and migration of your legacy applications. Application requirements include application installation (if available), security, configuration, networking, firewall, and licensing settings.

Phase 2: Compatibility Packaging

AWS SMEs utilize the EMP Compatibility Packager, along with the information provided in Phase 1, to identify and package applications with libraries, files, and other dependencies. The resulting package resolves OS dependencies and enables the application to successfully run on new versions of Windows Server.

Phase 3: Migration Phase

AWS SMEs deploy the application package created in Phase 2 to an AWS environment running the latest versions of Windows Server. The SMEs apply operational best practices to accelerate the migration of workloads, which reduces risk and improves the outcome of the migration. When the application has been deployed into the AWS cloud, you can focus on optimization of applications, processes, operations, and costs.

2 EMP Compatibility Packager

2.1 Overview

The EMP Compatibility Packager facilitates the creation of packages for legacy Windows applications to run on any supported target operating system. It uses an install-and-capture snapshot process to create a package that includes all changes made during application installation. When the package has been created, you can start the application and runtime analysis to detect any additional configuration changes that occur when the application first runs. If required, the packaging process can be automated with scripts or managed by tools, such as AWS Systems Manager.

2.2 Supported Operating Systems

- Windows Server 2019 x64
- Windows Server 2016 x64
- Windows Server 2012 R2 x64
- Windows Server 2012 x64
- Windows Server 2008 R2 x64
- Windows Server 2008 x86
- Windows Server 2003 SP2 x86

2.3 Features

2.3.1 Install Capture

During packaging, a snapshot of the current state of the operating system is taken and a capture process is started to record all changes made during the application installation.

2.3.2 Runtime analysis

Applications typically change files, resources, and registry when they are started and configured for use. EMP Compatibility Packager enables SMEs to capture the runtime changes and include them in the application package.

2.3.3 Legacy OS Support

Packages are created on the operating system currently supported by the application (for example, Windows Server 2008), which packages the application as the supported operating system sees it. This packaging differs from the target operating system. This process supports scenarios where applications that include pre-install checks look for a specific operating system.

2.3.4 Cross Platform Support

A single package can be deployed across multiple supported operating systems.

2.3.5 Pre/Post Deployment Scripts

Custom scripts can be included in the package and used to deploy MSI (Microsoft Installer) files or to execute other tasks that must be performed as part of the package/application deployment process.

2.3.6 Fast Edit, Zero Recompilation

The time and frustration that packaging teams face when packaging, testing, and updating applications and configurations is reduced with fast editing, which requires no package recompilation. EMP packages can be easily updated to include application updates, or upgraded to include the latest EMP components.

The configuration of a package is defined in a series of XML files, which includes:

- Configurations for Registry keys and values (AppRegistry.xml)
- File, Registry, and Network redirections (Redirections.xml)
- File Type Associations (FileAssociations.xml)
- Shortcuts (Shortcuts.xml)
- Executable programs (Programs.xml)
- Environment Variables (EnvironmentVariables.xml)

3 Application Compatibility Package

3.1 Package Overview

Compatibility packages for applications include everything an application requires to run on a modern operating system, including all of the application files, runtimes, components, deployment tools, and the embedded redirection and compatibility engine. The EMP Compatibility Packager creates a package on the existing operating system running the legacy application (for example, Windows Server 2003 or 2008) so that it can determine the dependencies on the out-of-support operating system and set the required redirections. The redirection and compatibility engine is designed to run in user mode. When the package is deployed to the target machine, the file type associations are registered and short cuts are created to run the application.

Compatibility packages rely on three principles:

- Packaging is performed on the operating system supported by the application.
- Runtime analysis is performed to detect resources added by the application after installation.
- The redirection and compatibility engine is included in the package.

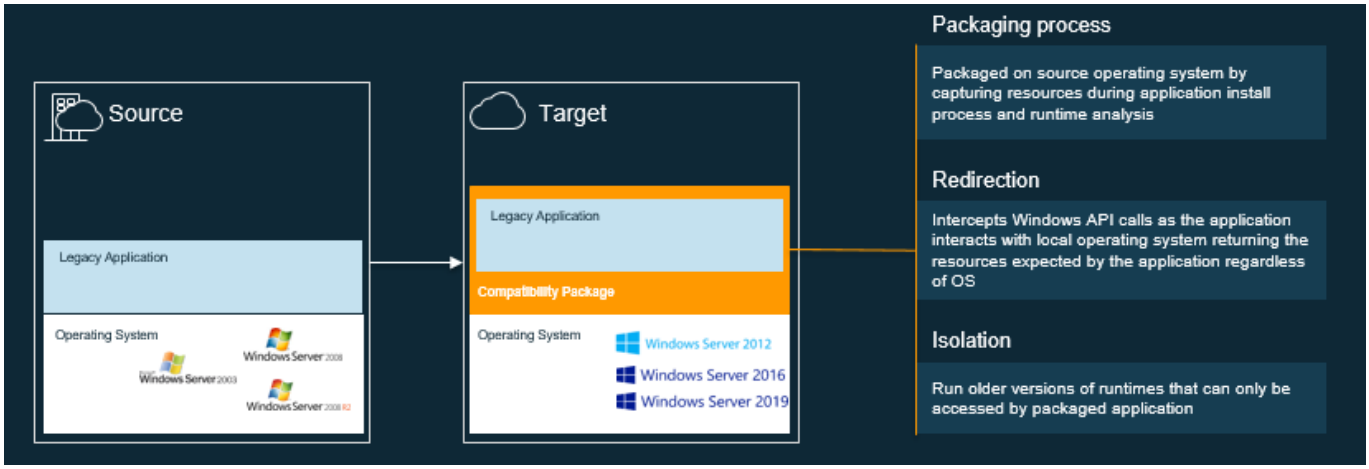

Figure 1 EMP compatibility packages

During installation, the EMP Compatibility Packager captures the resources from which the application installer/MSI reads and to which it writes. When the install capture process is complete, the Compatibility Packager performs runtime analysis to detect configuration changes that occur during the first launch of the application, for example, changes to configuration files.

The resulting package includes everything that the application needs to run on a modern operating system, including application files, runtimes, components, and deployment tools. The package does not include the legacy operating system, which means you never run any part of the legacy Windows Server version on the new Windows Server to which the application is upgraded.

The package additionally includes a redirection engine that intercepts the API calls that the application makes to the underlying Windows Server OS, and redirects them to the files and registry within the created package. Therefore, the application always requests resources from within the package rather than from the new Windows Server OS, so that the application runs successfully on the new OS.

This lightweight strategy means that the target application can see a combination of the redirected and local file system and registry with minimal impact on the performance of the application or local machine. The approximate RAM overhead per application is 3 to 5 MB, with no measurable CPU impact after the application has started.

Common redirections supported by the EMP compatibility engine include:

1. The application uses a fixed port: the EMP engine redirects to an appropriate and available port on the new version of the operating system.
2. The application accesses data stored in a fixed location that is not available on the new OS version: the EMP engine redirects these requests to the appropriate location on the new version of the operating system.
3. The application is hard coded to c:WinNT: the EMP engine redirects the application to C:\Windows.
4. The application requires Java 1.3: the EMP engine packages and isolates the application with Java 1.3 so that it does not make calls to newer Java runtimes in the new version of the operating system.

3.2 Moving N-Tier Server Applications to the Cloud

On-premises N-tier server applications running on Windows Server 2003 and Windows Server 2008 present challenges to cloud migration, and to datacenter consolidation and migration projects. Typical migration blockers include:

- The application is unable to run on Server 2019, 2016, or 2012 R2.
- There is no install media or code.
- The application has infrastructure dependencies on server names, IP addresses, and more.

EMP compatibility packages solve these challenges by resolving application dependencies on machine names, IP addresses, and ports that were created when the application was first installed on an out-of-support version of Windows Server. Network redirections are applied to packages so that the application can be migrated to a new server. EMP compatibility packages support Windows services common in Oracle Application Server, PeopleSoft, Tomcat, IBM WebSphere, and SQL Server 2005.

Running Windows Server applications in EMP packages does not change the deployment architecture or behavior of the application. When considering the use of compatibility packages for cloud or datacenter implementations, architects must consider the following factors.

- The number of tiers remains the same. For example, 3-tier applications remain as 3-tier applications.
- Scalability and redundancy rules remain the same. Note that fewer servers may be required due to the increased performance of Server 2016, the machine, and the instance.
- High availability and disaster recovery capabilities remain the same. If the application provides these capabilities, they will be available when the application runs in the compatibility package, and the same architectural rules will apply.

4 Features

4.1 Isolation

4.1.1 Application and Runtime Isolation

Applications with conflicting requirements or outdated run times can safely run by being isolated from other applications on the server. Isolation enables EMP compatibility packages to include runtimes that conflict with other runtimes on the server, so that they are used only by the packaged application. Examples of conflicting requirements might include use of Java 1.3, .Net 2.0, and msxml.dll.

4.1.2 Network Isolation

To help reduce application vulnerabilities, EMP packages can be configured to complement your existing firewall services and to provide isolation at the TCP/IP layer by hooking Winsock APIs.

4.2 Redirection

4.2.1 File and Registry Redirection

By packaging applications on the operating system version supported by the application, the EMP Compatibility Packager can overcome install and runtime challenges by creating redirections for the registry and file paths. This includes redirections for hard-coded locations, or locations such as C:\WinNT that might not exist on the modern operating system.

4.2.2 Network Redirections (IP and/or Port)

The Redirection and Compatibility Engine hooks Winsock APIs, and applies redirection rules to IP addresses and ports. Network redirections can overcome server name or network address/port settings that are hard-coded in the server application.

4.3 Compatibility

4.3.1 Application to Operating System Compatibility

EMP software runs legacy applications within a compatibility package on modern, up-to-date operating systems. The EMP compatibility package does not contain any parts of the legacy operating system. Instead, it intercepts the operating system requests and redirects and resolves them against the up-to-date host operating environment.

The EMP compatibility package permits the legacy application to run alongside other applications and/or other versions of the same application. For example, multiple versions of Microsoft Office can run simultaneously on the target operating system, or two incompatible 32-bit applications can run together, isolated from each other by the EMP compatibility package.

Additional compatibility features include:

- The ability to report a different OS version back to the application.
- Multi-instance out-of-process COM virtualization for applications and Office 2007/2003.
- The ability to suppress UAC prompts when applications request, but do not require it.
- Process flagging so they don’t run under WOW64.

4.3.2 Client-Side Integration

By default, applications behave as though they are installed natively so they can access other applications, printers, and devices. Isolation should be enabled if this is not the desired behavior.

Historically, user-mode application virtualization solutions have been tightly sandboxed and unable to communicate beyond their virtual environment to integrate with local devices. With EMP, applications don't have to be packaged together or run in the same compatibility package to work together. This is because the user mode redirection engine in the package supports integration from the packaged application to any local device or application, by hooking into locally installed processes (either all processes or named processes).

4.3.3 Services and Drivers

EMP Compatibility packages support Windows Services out of the box. Drivers aren't captured in an EMP package but can be extracted and deployed locally using the EMP deployment script feature, as long as they are compatible with the target operating system.

4.3.4 Technical Specifications

Refer to the EMP documentation for the complete requirements to install the EMP Compatibility Packager and run packages. The following list is a summarized version of the complete requirements:

- The Redirection and Compatibility Engine is written in C++ and does not require installation of C runtimes.
- The EMP package and the compatibility packager require .NET Framework 4.0 Client Profile (or higher) to be installed.
- Limited to 4 MB client binaries per package.
- Limited to approximately 2 to 5 MB RAM per package.
- Limited to approximately 1% CPU per package.

4.4 Compatibility with Application Virtualization Solutions

Applications running in EMP packages are compatible with other application virtualization solutions deployed on the server or desktop. EMP compatibility packages can communicate with applications packaged using other technologies, unless the package or third-party application virtualization package has been configured for isolation.

5 Conclusion

AWS EMP for Windows Server enables application migration from legacy Windows Server systems to supported Windows operating systems. The EMP process includes an assessment of your application and testing requirements, packaging of the application and its dependencies on the legacy operating system based on your requirements, and migration to an AWS environment running a supported Windows Server version. When migrated to the new server environment, the application will run on the new OS as it would on the legacy operating system. The new package redirects application requests and does not include installation of any part of the legacy operating system.