Deploying Microsoft Endpoint Protection for Windows Azure

Microsoft Corporation

Published: March 2012

Send suggestions and comments about this document to [eppazurefb@microsoft.com](mailto:eppazurefb@microsoft.com).

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2012 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Windows, Windows Server, System Center Operations Manager, and Active Directory are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

[1 Introduction 3](#_Toc318874588)

[2 Overview 3](#_Toc318874589)

[2.1 Functionality recap 3](#_Toc318874590)

[2.2 Providing feedback 4](#_Toc318874591)

[2.3 How it works 4](#_Toc318874592)

[3 Prerequisites 5](#_Toc318874593)

[3.1 Windows Azure SDK Version Support 6](#_Toc318874594)

[4 Deployment 6](#_Toc318874595)

[4.1 Install Microsoft Endpoint Protection for Windows Azure 6](#_Toc318874596)

[4.2 Enable your Windows Azure service for antimalware 7](#_Toc318874597)

[4.3 Optionally customize antimalware configuration options 8](#_Toc318874598)

[4.3.1 Updating configuration for deployed services 10](#_Toc318874599)

[4.4 Configure Azure Diagnostics to capture antimalware related information 10](#_Toc318874600)

[4.4.1 Antimalware events 10](#_Toc318874601)

[4.4.2 Antimalware logs 11](#_Toc318874602)

[4.5 Publish your service to Windows Azure 11](#_Toc318874603)

[5 Telemetry 12](#_Toc318874604)

[6 Updates 12](#_Toc318874605)

[6.1 Signature updates 12](#_Toc318874606)

# Introduction

This document explains how to use the Microsoft Endpoint Protection for Windows Azure Customer Technology Preview (CTP) to configure and deploy antimalware protection as part of your Windows Azure service.

Additional resources and information about Microsoft Endpoint Protection for Windows Azure can be found on the Web at the following address: <http://go.microsoft.com/fwlink/?LinkID=244362>

# Overview

Microsoft Endpoint Protection for Windows Azure is a software package that extends the Windows Azure Software Development Kit (“Windows Azure Tools for Microsoft Visual Studio”) by adding an antimalware plugin. The antimalware plugin provides a means of enabling antimalware protection in the underlying virtual machines (VMs) atop which your Windows Azure hosted service runs. Once you enable antimalware as part of your service definition, the antimalware components will be installed, configured, and started within each role instance VM when it is spun up by the Windows Azure Fabric Controller.

## Functionality recap

When you deploy the antimalware solution as part of your Windows Azure service, the following core functionality is enabled:

* **Real-time protection**monitors activity on the system to detect and block malware from executing.
* **Scheduled scanning**periodically performs targeted scanning to detect malware on the system, including actively running malicious programs.
* **Malware remediation**takes action on detected malware resources, such as deleting or quarantining malicious files and cleaning up malicious registry entries.
* **Signature updates**installs the latest protection signatures (aka “virus definitions”) to ensure protection is up-to-date.
* **Active protection**reports metadata about detected threats and suspicious resources to Microsoft to ensure rapid response to the evolving threat landscape, as well as enabling real-time signature delivery through the Dynamic Signature Service (DSS).

Microsoft’s antimalware endpoint solutions are designed to run quietly in the background without human intervention required. Even if malware is detected, the endpoint protection agent will automatically take action to remove the detected threat. Refer to the document “Monitoring Microsoft Endpoint Protection for Windows Azure” for information on monitoring for malware-related events or VMs that get into a “bad state.”

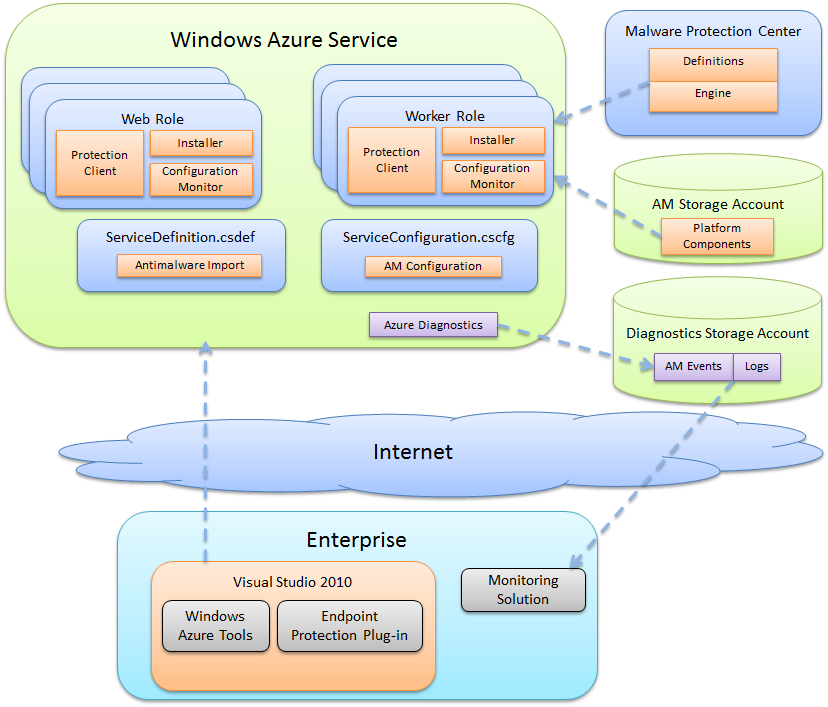
## Providing feedback

The goal of this technology preview version of Microsoft Endpoint Protection for Windows Azure is to give you a chance to evaluate this approach to providing antimalware protection to Windows Azure VMs and provide feedback. We want to hear from you! Please send any feedback to [eppazurefb@microsoft.com](mailto:eppazurefb@microsoft.com).

## How it works

Microsoft Endpoint Protection for Windows Azure includes SDK extensions to the Windows Azure Tools for Visual Studio which provides the means to configure your Windows Azure service to include endpoint protection in the specified roles. When you deploy your service, an endpoint protection installer startup task is included that runs as part of spinning up the virtual machine for a given instance. The startup task pulls down the full endpoint protection package platform components from Windows Azure Storage for the geographical region specified in the Service Configuration (.cscfg) file and installs it, applying the other configuration options specified.

Once up and running, the endpoint protection client downloads the latest protection engine and signatures from the Internet and loads them. At this point the virtual machine is up and running with antimalware protection enabled. Diagnostic information such as logs and antimalware events can be configured for persistence in Windows Azure storage for monitoring. The following diagram shows the “big pictures” of how all the pieces fit together:



# Prerequisites

Before you get started, you should already have a Windows Azure account configured and have an understanding of how to deploy your service in the Windows Azure environment. You will also need Microsoft Visual Studio 2010. If you have Visual Studio 2010, the Windows Azure Tools for Visual Studio, and have written and deployed Windows Azure services, you’re ready to go.

If not, do the following:

1. Sign up for a Windows Azure account  
   <http://windows.azure.com>
2. Install Visual Studio 2010  
   <http://www.microsoft.com/visualstudio>
3. Install Windows Azure Tools for Visual Studio   
   <http://msdn.microsoft.com/en-us/library/windowsazure/ff687127.aspx>

## Windows Azure SDK Version Support

Microsoft Endpoint Protection for Windows Azure has been tested with Windows Azure SDK 1.7 (included as part of installing Windows Azure Tools for Visual Studio). Versions of the Azure SDK previous to 1.7 are not supported. Versions of the Azure SDK greater than version 1.7 may work, but have not been tested, and are not officially supported. Therefore version 1.7 of the Windows Azure SDK is recommended.

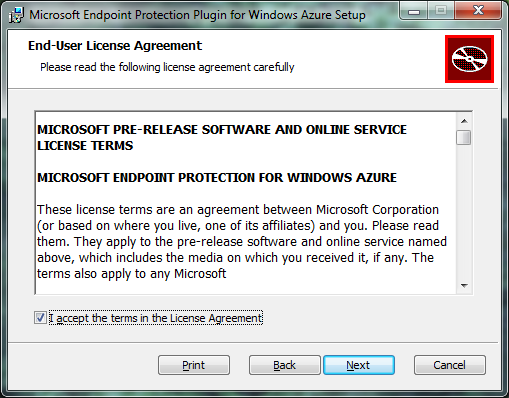
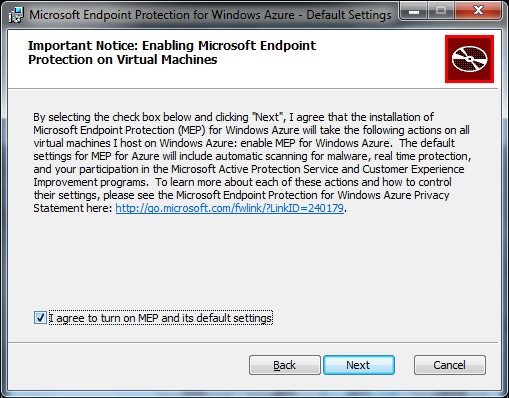
# Deployment

Once you have Visual Studio 2010 and the Windows Azure Tools installed, you’re ready to get antimalware protection up and running in your Azure VMs. To do so, follow these steps:

1. Install Microsoft Endpoint Protection for Windows Azure
2. Enable your Windows Azure service for antimalware
3. Optionally customize antimalware configuration options
4. Configure Azure Diagnostics to capture antimalware related information
5. Publish your service to Windows Azure

## Install Microsoft Endpoint Protection for Windows Azure

Run the Microsoft Endpoint Protection for Windows Azure setup package. The package can be downloaded from the Web at <http://go.microsoft.com/fwlink/?LinkID=244362>.

Follow the steps in the setup wizard to install the endpoint protection components. The required files are installed in the Windows Azure SDK plugins folder. For example:

"%Program Files%\Microsoft SDKs\Windows Azure\.NET SDK\2012-06\bin\plugins\Antimalware

Once the components are installed, you’re ready to enable antimalware in your Windows Azure roles.

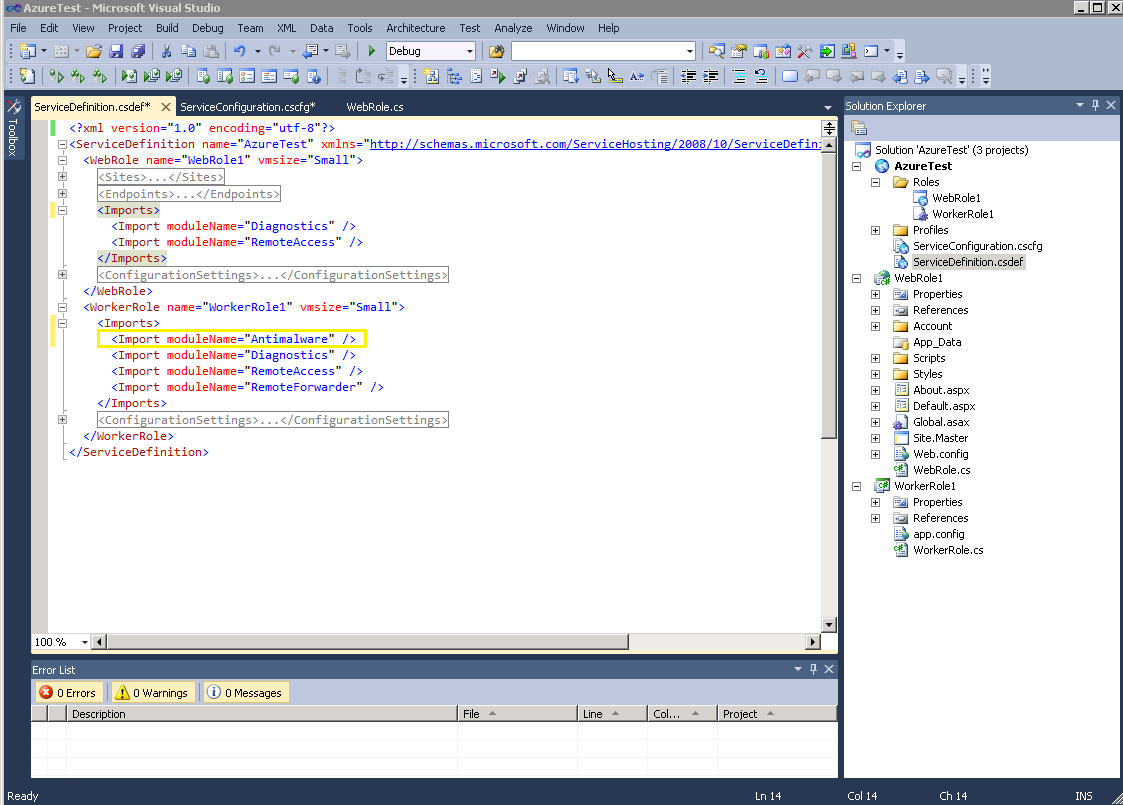
## Enable your Windows Azure service for antimalware

To enable your service to include endpoint protection in your role VMs, simply add the “Antimalware” plugin when defining the role.

1. In Visual Studio 2010, open the service definition file for your service (ServiceDefinition.csdef).
2. For each role defined in the service definition (e.g. your worker roles and web roles), update the <imports> section to import the “Antimalware” plugin by adding the following line:

<Import moduleName="Antimalware" />

The following image shows an example of adding antimalware for the worker role “WorkerRole1” but not for the project’s Web role.

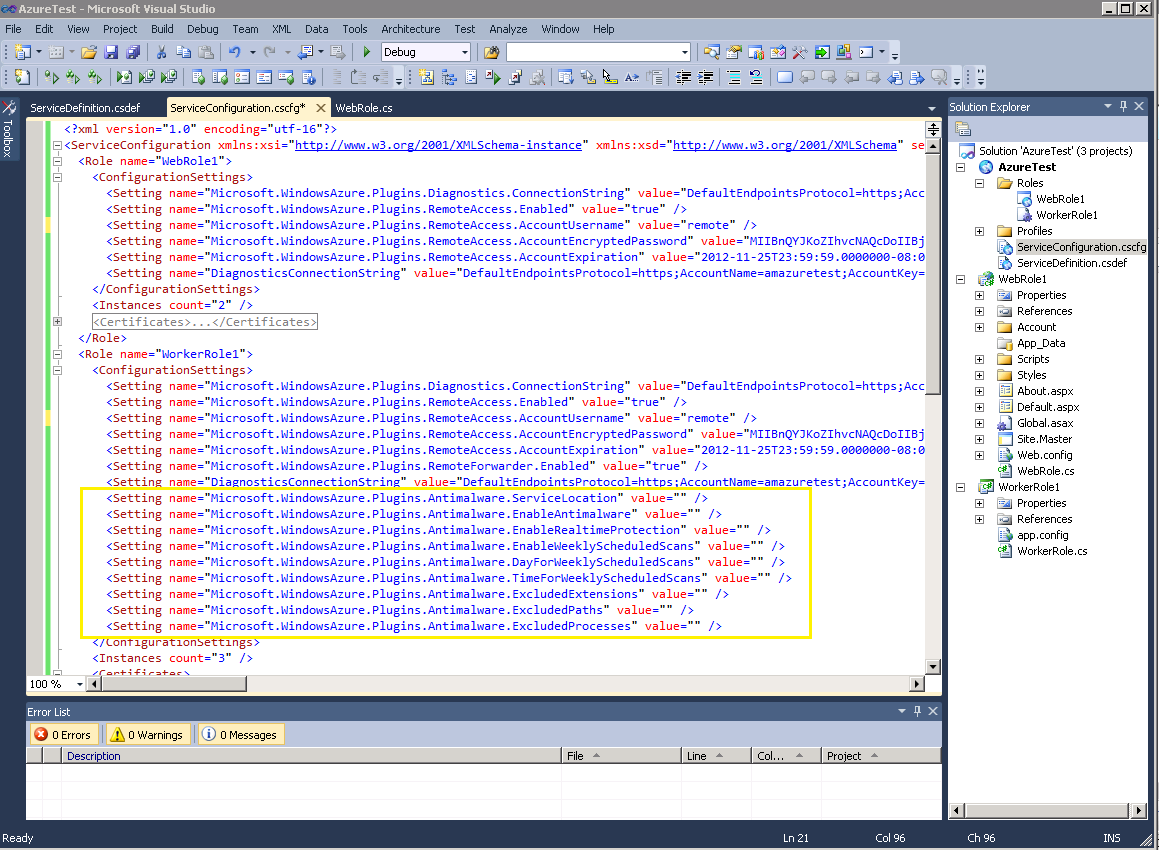


1. Save the service definition file.

In this example, the worker role instances for the project will now include endpoint protection running in each virtual machine. However the web role instances will not include antimalware protection, because the antimalware import was only specified for the worker role. The next time the service is deployed to Windows Azure, the endpoint protection startup task will run in the worker role instances and install the full endpoint protection client from Windows Azure Storage, which will then install the protection engine and signatures from the Internet. At this point the virtual machine will have active protection up and running.

## Optionally customize antimalware configuration options

When you enable a role for antimalware protection, the configuration settings for the antimalware plugin are automatically added to your service configuration file (ServiceConfiguration.cscfg). The configuration settings have been pre-optimized for running in the Windows Azure environment. You do not need to change any of these settings. However, you can customize these settings if required for your particular deployment.



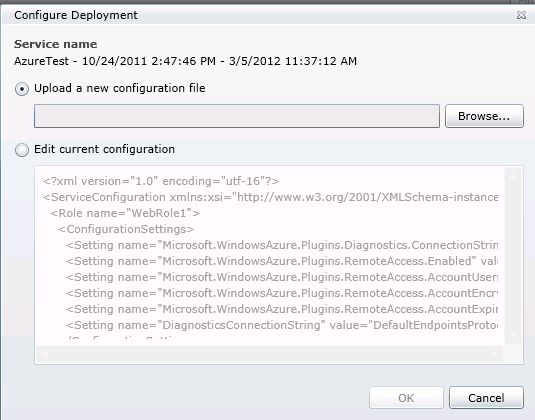
**Default antimalware configuration added to service configuration**

The following table summarizes the settings available to configure as part of the service configuration:

|  |  |  |  |
| --- | --- | --- | --- |
| Setting | Options | Default | Description |
| ServiceLocation | East Asia  North Central US  North Europe  South Central US  Southeast Asia  West Europe | South Central US | Specifies the region to download the antimalware client components from. Using the same region as your service can improve deployment time and decrease costs |
| EnableAntimalware | true  false | true | Enables or disables endpoint protection |
| EnableRealtimeProtection | true  false | true | Enables or disables real-time scanning, such as on-access disk scanning |
| EnableWeeklyScheduledScans | true  false | true | Enables or disables a periodic “quick scan” for active malware on the system |
| DayForWeeklyScheduledScans | 0 - 7 | 7 | 0 – scan daily  1 – Sunday  2 – Monday  …  7 – Saturday |
| TimeForWeeklyScheduledScans | 0 - 1440 | 120 | Hour at which to begin the scheduled scan. Measured in 60 minute increments corresponding to the desired hour.  60 – 1:00 AM  120 – 2:00 AM  …  1380 – 11:00 PM |
| ExcludedExtensions | extension1|extension2 | None | Pipe-delimited list of file extensions to exclude from scanning. Example: gif|log|txt  would exclude files with the .gif, .log, or .txt extension from being scanned. |
| ExcludedPaths | path1|path2 | None | Pipe-delimited list of paths to files or folders to exclude from scanning. Example: e:\approot\worker.dll|e:\approot\temp would exclude the file worker.dll in the e:\approot folder and anything under the folder e:\approot\temp from being scanned. |
| ExcludedProcesses | process1|process2 | None | Pipe-delimited list of process exclusions. Any file opened by an excluded process will not be scanned (the process itself will still be scanned – to exclude the process itself, use the ExcludedPaths configuration). Example: D:\Program Files\MyApp.exe would exclude any files opened by MyApp.exe from being scanned. |

### Updating configuration for deployed services

Windows Azure provides the ability to update a service configuration “on the fly” to a service that is already running in Windows Azure. For example on the Windows Azure Portal you can select the “Configure” option to upload a new configuration file or manually edit configuration settings for an existing deployment.



Microsoft Endpoint Protection for Windows Azure supports applying changes to a deployed service. If you change the antimalware settings in the service configuration file, you can deploy the new configuration to your running service and the antimalware related settings will update automatically.

Microsoft Endpoint Protection for Windows Azure must have already been deployed as part of service deployment. You cannot deploy or remove endpoint protection through a configuration update.

## Configure Azure Diagnostics to capture antimalware related information

In order to monitor the health of your antimalware deployment, you’ll need to configure Windows Azure to pull the antimalware useful events and logs into Windows Azure storage. From there, any number of Windows Azure monitoring solutions can be used for antimalware monitoring and alerting.

Refer to the MSDN documentation for Windows Azure Diagnostics for general information:

<http://msdn.microsoft.com/en-us/library/windowsazure/gg433048.aspx>

### Antimalware events

For antimalware events, you need to configure diagnostics to pull events from the System event log where the source is “Microsoft Antimalware.” In general you’ll want to pull Error and Warning events. See the “Monitoring Microsoft Endpoint Protection for Windows Azure” document for more information on which events to monitor.

Here’s an example of the code you might add to the entry point for your service:

//add antimalware diagnostics

var config = DiagnosticMonitor.GetDefaultInitialConfiguration();

//exclude informational and verbose event log entries

config.WindowsEventLog.DataSources.Add("System!\*[System[Provider[@Name='Microsoft Antimalware'] and (Level=1 or Level=2 or Level=3)]]");

//write to persisted storage every 1 minute

config.WindowsEventLog.ScheduledTransferPeriod = System.TimeSpan.FromMinutes(1.0);

DiagnosticMonitor.Start("Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString", config);

In this case, any antimalware errors or warnings from the System event log will be written to Windows Azure Storage every 1 minute. Use an interval that makes sense for your monitoring requirements.

You can view the raw events by looking at the WADWindowsEventLogsTable table in the storage account you configured to use with Windows Azure Diagnostics. This can be useful to validate that antimalware event collection is working. For example start with including informational events (Level=4) to validate your configuration end-to-end, then turn them off via configuration update.

### Antimalware logs

In addition, the following locations include logs that may be useful if you are encountering problems getting the antimalware components up and running in your Windows Azure VMs:

* %programdata%\Microsoft Endpoint Protection  
  includes logs for the startup task that is deployed with your Windows Azure service
* %programdata%\Microsoft\Microsoft Security Client  
  includes logs for the installation of the endpoint protection platform components

You can configure Windows Azure diagnostics to pull these logs as well by implementing custom logging to move these logs to blob storage. See the following documentation on MSDN for more information:

<http://msdn.microsoft.com/en-us/library/windowsazure/hh411528.aspx>

## Publish your service to Windows Azure

The final step once you have everything configured in Visual Studio 2010 is to publish your service to Windows Azure. The roles that have antimalware configured will include the additional startup task to install and start the endpoint protection client as part of service deployment. As with any Windows Azure service, you can package your service for deployment through the Windows Azure portal, or you can publish from within Visual Studio 2010. Either option will work with Microsoft Endpoint Protection for Windows Azure.

Once you publish your service, any roles with antimalware enabled will have the Microsoft Endpoint Protection client running within the VM.

# Telemetry

Microsoft Endpoint Protection for Windows Azure includes the Microsoft Active Protection Service (MAPS), which reports telemetry on events such as detected malware threats and suspicious files to the Microsoft Malware Protection Center (MMPC). This data is used to improve the protection capabilities of the client. Microsoft Endpoint Protection for Windows Azure requires MAPS.

# Updates

Microsoft Endpoint Protection for Windows Azure uses the latest version of the endpoint protection client, which is installed at deployment time. As part of the startup task that runs in the virtual machine, the latest protection client components are downloaded and installed from Windows Azure Storage. In-place upgrades to newer clients are not supported (in other words, while a particular VM instance is running, the endpoint protection client will not update to a newer version). The term “endpoint protection client” refers to the platform components such as the antimalware service executables, filter drivers, and so forth. It does not include the protection engine and signatures, which are released on a more frequent basis.

While a VM will not update to a newer protection client while it is running, keep in mind that the next time a role instance is created by the Windows Azure Fabric Controller, it will always get the latest version of the endpoint protection client.

## Signature updates

Signatures are released by the Microsoft Malware Protection Center on a frequent basis, usually multiple times throughout the day. Microsoft Endpoint Protection downloads and installs the latest signatures once per day. New versions of the protection engine are installed as part of the normal signature update process. Whenever a new protection engine is published, the next signature update will include the new engine version.