Docusnap and Intune

Automation of the Discovery-Windows.exe via Intune

TITLE	Docusnap and Intune
AUTHOR	Docusnap Consulting
DATE	5/21/2026
VERSION	1.0 | valid from 05/20/2026

CONTENTS

1. Start 4
2. Requirements 4
3. Azure Storage Account 5
4. Container 7
4.1 Container for Discovery-Windows.exe 7
4.2 Container for the feedback files 9
4.3 Shared Access Token 10
5. Preparing the inventory 11
5.1 Script Parameter 11
5.2 Configuration file 12
6. Deployment via Intune 14
6.1 Create a package 14
6.2 Deploy package 15
6.3 Update package 17
7. Download for Docusnap 18

1. Start

With the increasing use of Microsoft Azure, Windows systems are not always part of Active Directory. Of course, these systems are relevant for a complete overview and documentation. Automatic inventorying of precisely these systems is often difficult.

This HowTo describes how to automate Discovery-Windows.exe via Intune. It describes the prerequisites and automation of execution.

2. Requirements

There are several prerequisites for implementing the solution. These include technical requirements as well as permissions and knowledge of Azure and Intune.

- Docusnap onPrem
- The appropriate permissions in Intune and Azure
- Intune for the automation of the Discovery Windows.exe
- Azure Subscription
- Azure Storage Account
    - Deploy the Discovery-Windows.exe
    - Storage of the feedback files
- Shared Access Token
- PowerShell Script – Download Link in chapter Preparing the inventory

Important: This solution incurs costs in your Azure subscription. These can vary depending on the configuration and use. The Microsoft Azure price calculator is available to help you calculate these in advance.

3. Azure Storage Account

An Azure Storage Account is required for provisioning. Discovery-Windows.exe and the result files of the inventoried systems are stored in this account. A separate storage account should be created rather than using an existing one.

The name can be chosen freely. To save costs, the storage account for the Discovery-Windows.exe should use the performance level Standard and the Locally-redundant storage (LRS).

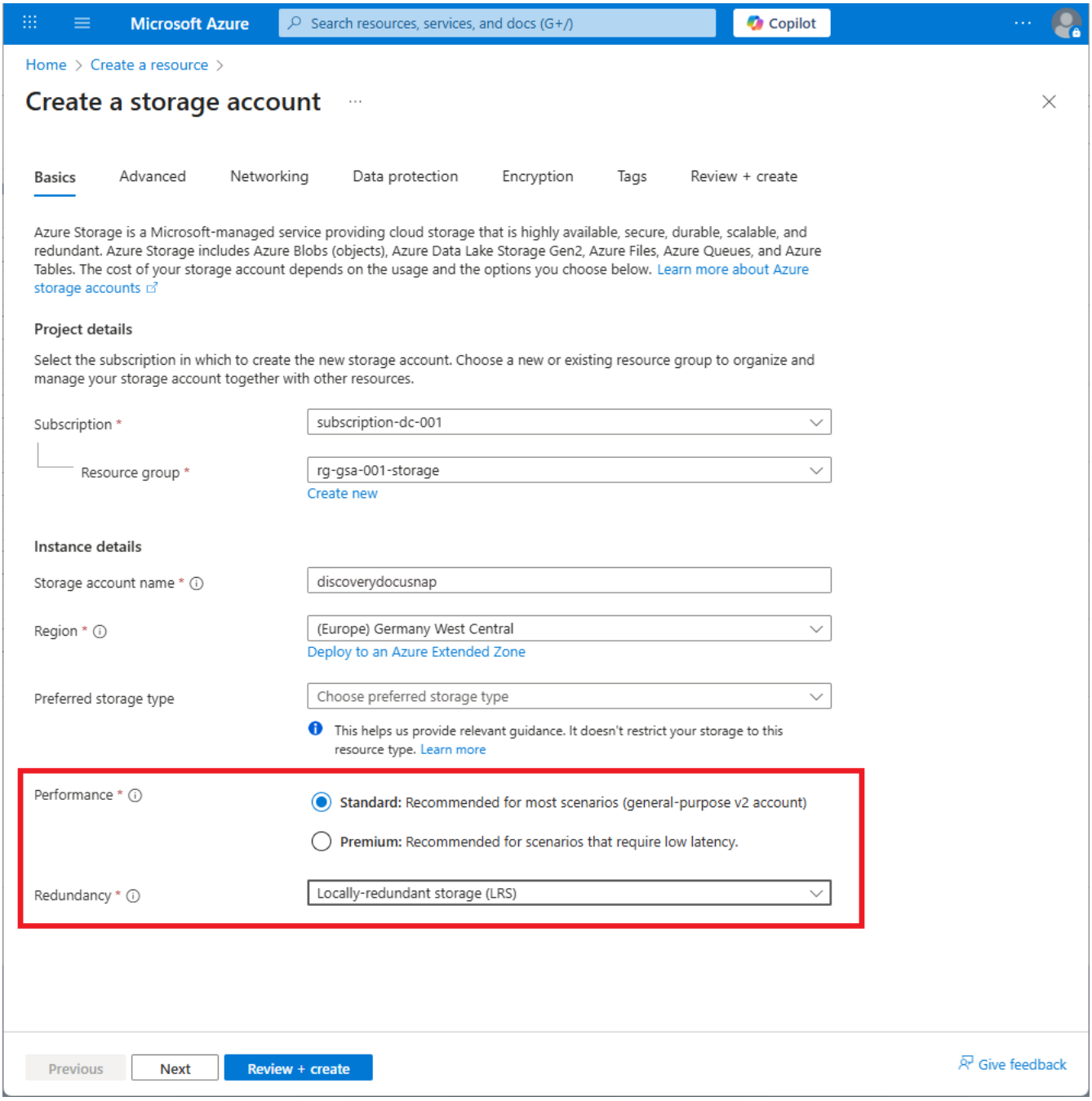

Figure 1 - Create a storage account - Basic settings

Anonymous access must be enabled in the advanced settings. This will allow clients to download the Discovery-Windows.exe and upload the XML files. The other settings remain at the default values.

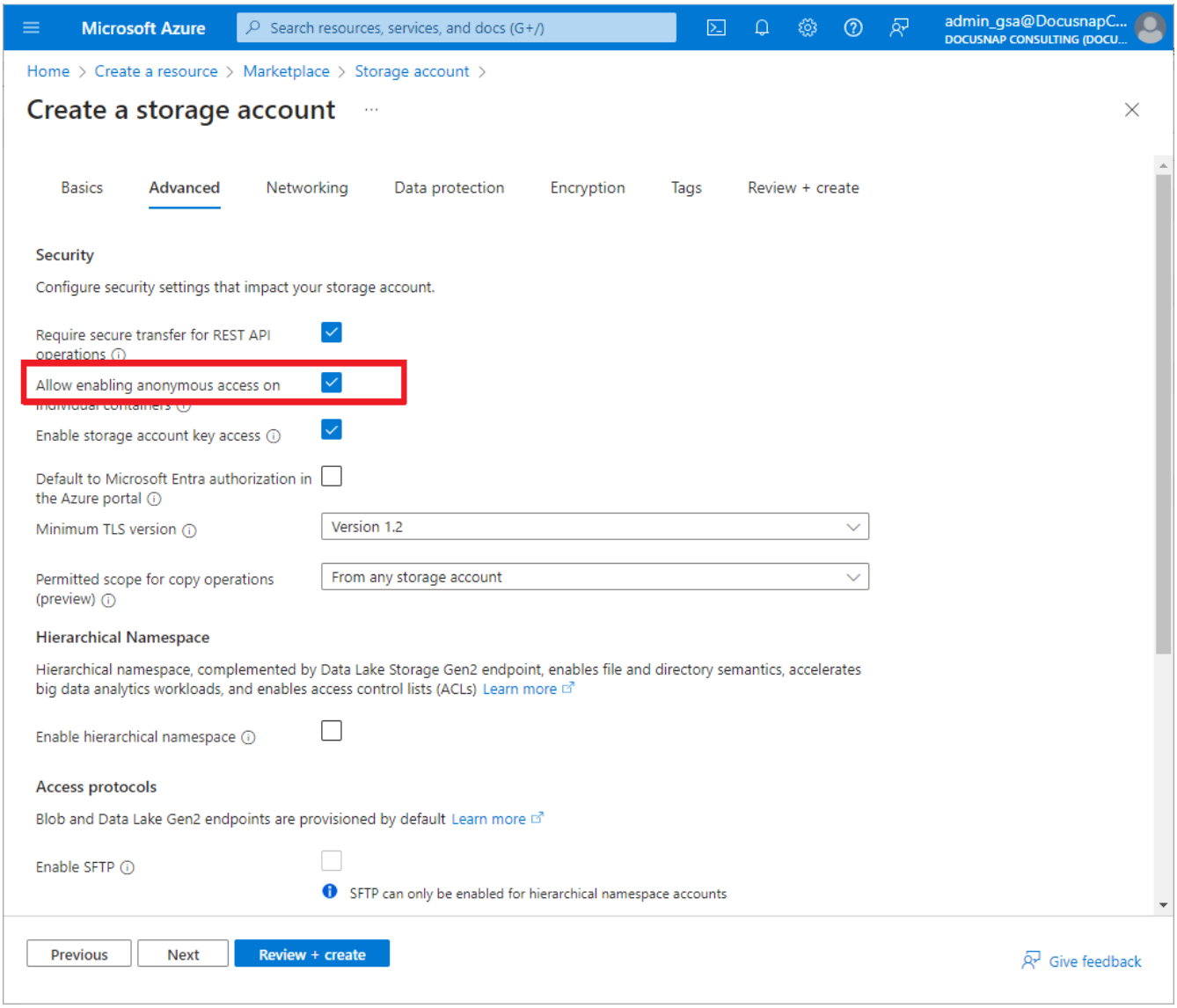

Figure 2 - Create storage account - Advanced settings

4. Container

After setting up the storage account, two containers are created.

4.1 Container for Discovery-Windows.exe

The first container contains the Discovery-Windows.exe file, which must be downloaded anonymously. For this reason, the anonymous access level is set to **Blob** when the container is created.

The name is freely selectable. In our example, the name **script** is chosen.

Important: The name is used in the Configuration.xml of the script. If this is changed, this must also be considered in the Configuration.xml.

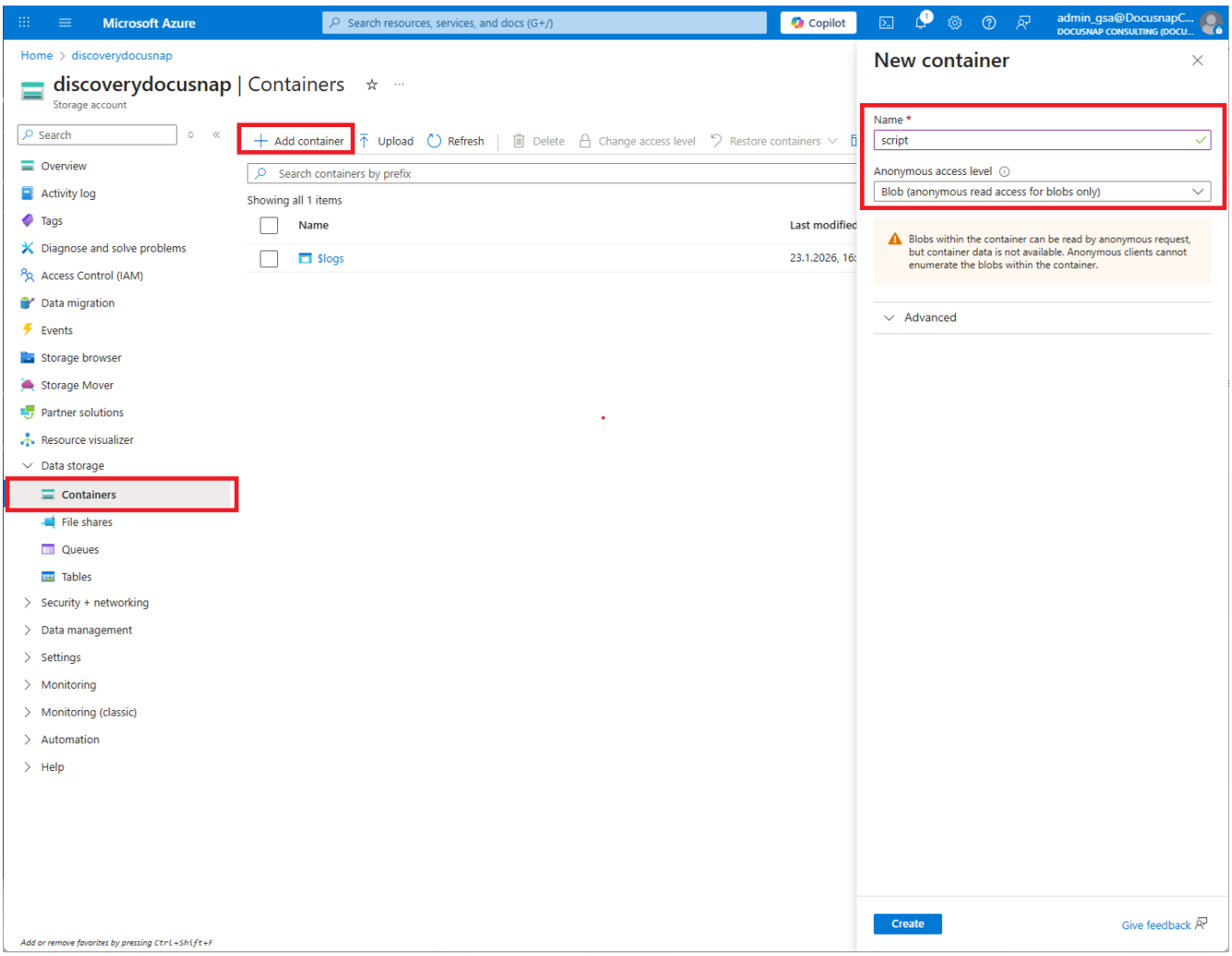

Figure 3 – Container for Discovery-Windows.exe

After an update of Docusnap, a new version of Discovery-Windows.exe is available. This should be uploaded to the container as soon as possible. The clients will download the Discovery-Windows.exe from the container with every inventory. This means that nothing needs to be changed on the clients.

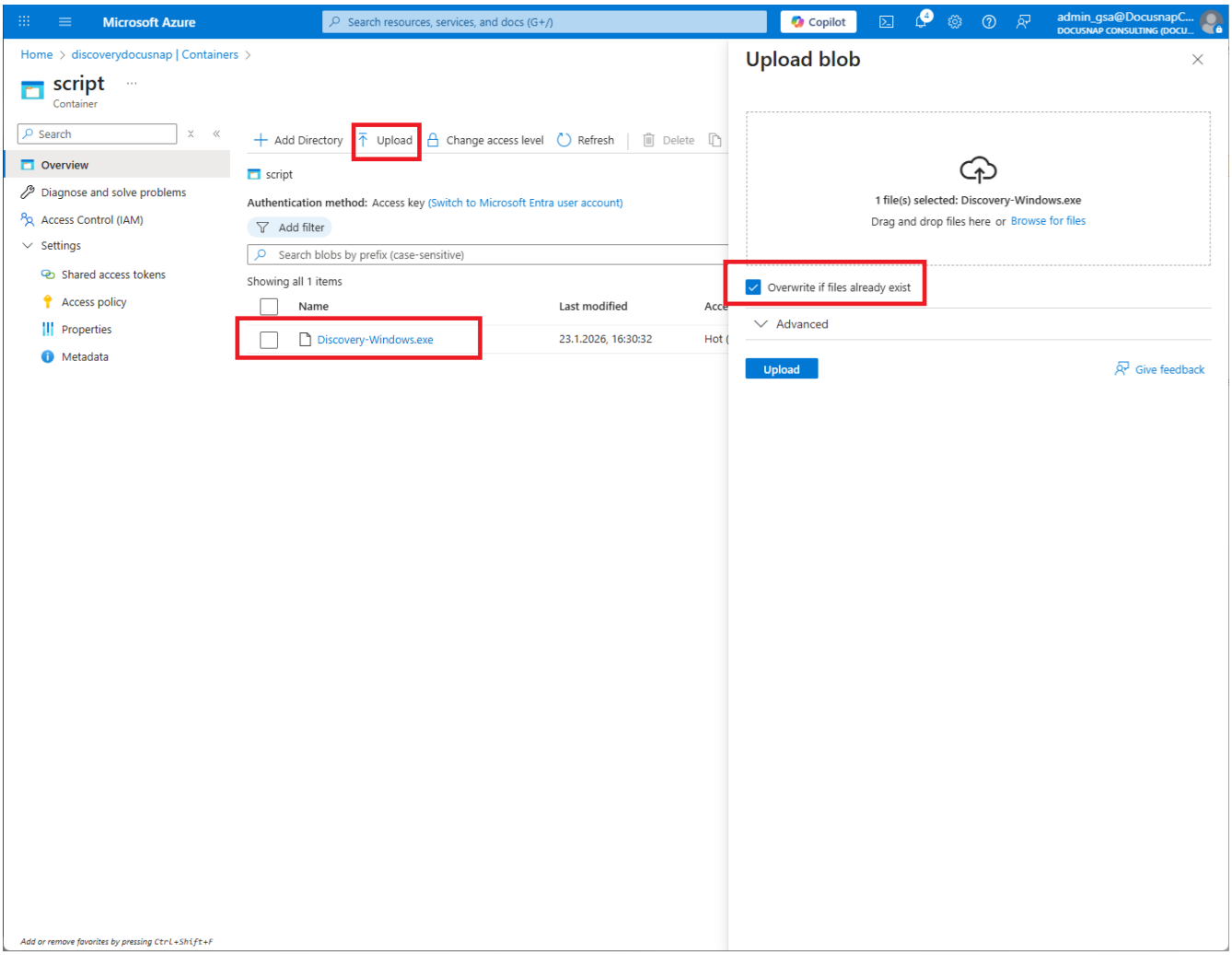

Figure 4 – Uploading and updating the Discovery-Windows.exe

4.2 Container for the feedback files

The second container is the upload directory for the feedback files of the inventory.

The anonymous access level is set to **Private**. The name used in our example is **exports**. The name can be chosen freely.

Important: The name is used in the Configuration.xml of the script. If this is changed, this must also be considered in the Configuration.xml.

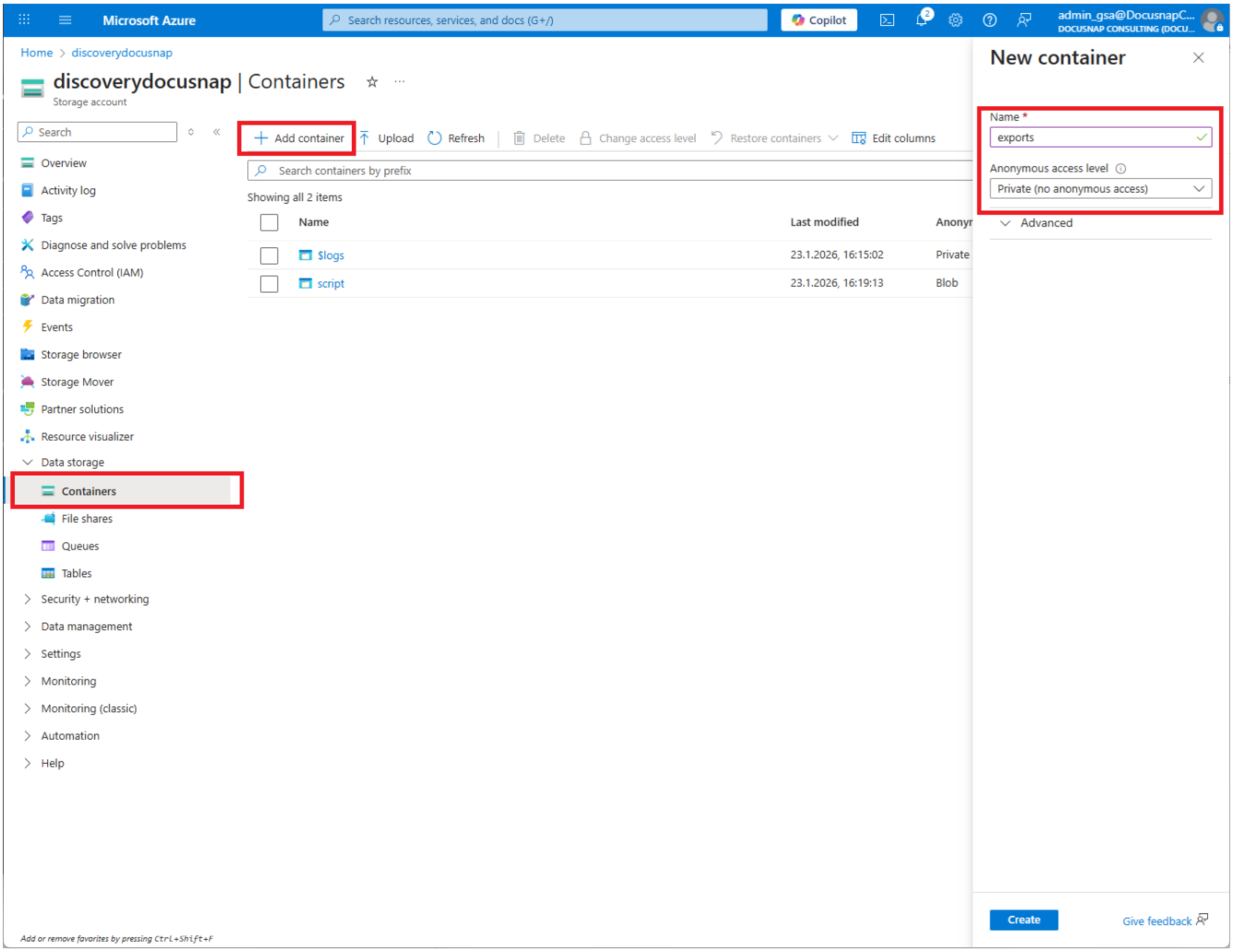

Figure 5 - Container for Feedback files

4.3 Shared Access Token

A shared access token is required to allow clients to store the exported data in the container without authentication. This token needs to be created. It is important to note that the token only allows files to be written, not read. If the token is lost to a third party, they will be able to write to the container, but not read any content.

A shared access token with the authorizations **Add**, **Create** and **Write** is created in the container **exports**.

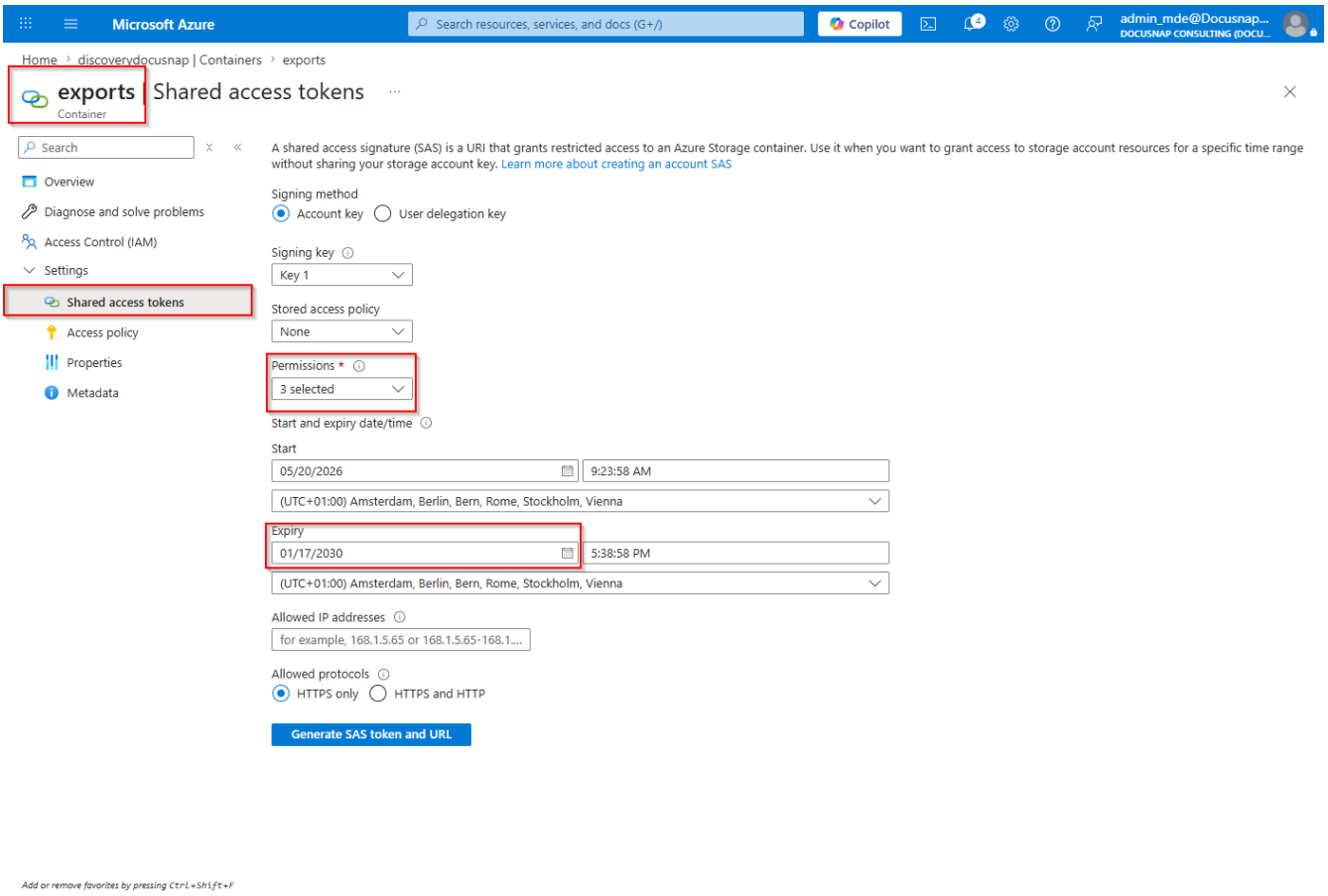

Figure 6 - Creating and authorizing the Shared Access Token

The shared access token has an expiration date. When a token expires, the script must be modified, repackaged, and redeployed because the token is hard coded into the script. For this reason, it is recommended that you set a long-term validity.

When you click on **Generate SAS token and URL**, the blob SAS token and the blob SAS URL are displayed. Once the token has been generated, the Blob SAS URL must be copied and **saved temporarily**.

The blob SAS URL is required for the execution script. If the blob SAS URL is lost, a new SAS token must be created.

5. Preparing the inventory

The script creates the C:\Discovery-Windows directory on each system that has been rolled out and sets the permissions for administrators and the system account. If the directory already exists, the existing permissions are overwritten.

The Discovery-Windows-Azure PowerShell script is used for automation. It has several functions that are invoked via parameters. Settings and information about the SAS token are stored in a Configuration.xml file.

Both the PowerShell script and the Configuration.xml file can be downloaded from our download portal.

5.1 Script Parameter

The following parameters are available

Install
copies this script and the configuration file to a local location and configures a Windows task to run the script regularly with the parameter -Run

Run
executes the Discovery-Windows.exe and uploads the result file to the configured Azure Storage account. The configuration file is expected under the name Configuration.xml in the same directory as the script file.

Collect
downloads all feedback files from the Azure Storage account and makes them available in a central directory. The feedback files are deleted from Azure Storage.

Uninstall
uninstalls the script as well as the created Windows task.

ConfigName
(optional) If a configuration file is used that differs from the default name Configuration.xml, this name can be specified with the -ConfigName parameter. The file must be located in the same directory as the script file.

Verbose
displays additional information on the individual steps of the script. Parameter is used to analyse problems.

5.2 Configuration file

First, the script for the inventory is configured. The configuration of the download script for the Docusnap Server or a Docusnap Discovery Service is explained in the chapter Download for Docusnap.

The parameters **DiscoveryExe** and **ResultUpload** must be stored for the Configuration.xml of the script.

<RunParameter>

Specification of the parameters for the Discovery-Windows.exe. By default, the parameters -A 0, -W and -G are defined. This means that no archive versions of the feedback files are stored. Discovery-Windows.exe is also executed in silent mode and no network connections are recorded. The parameters can be displayed via parameter Discovery-Windows.exe -H.

<DiscoveryExe>

Can be copied in the properties of the Discovery-Windows.exe in the corresponding Azure container. The properties are called up via the context menu of the Discovery-Windows.exe. The URL is composed of the name of the storage account and the container.

```
https://<Name des Storage Accounts>.blob.core.windows.net/<Name des Containers>/Discovery-Windows.exe
```

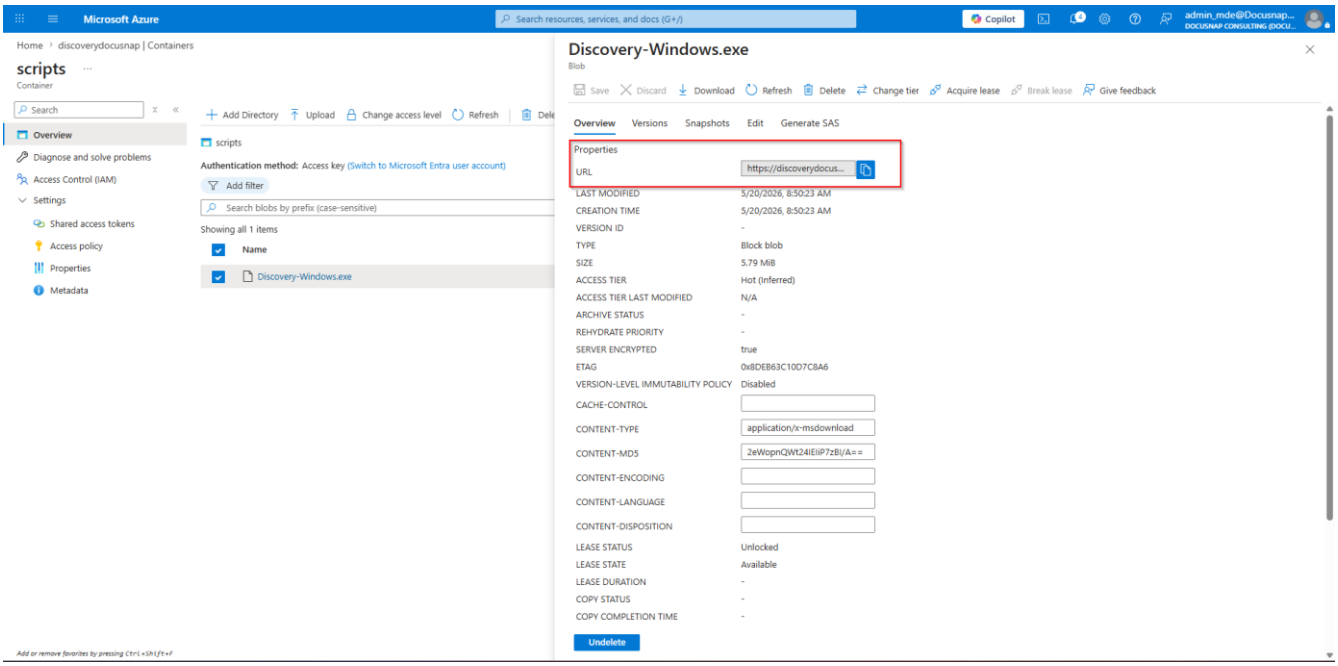

Figure 7 - Discovery-Windows.exe URL

<ResultUpload>

Blob-SAS-URL from the chapter Shared Access Token.

Optionally, you can customise the parameters of Discovery-Windows.exe and the schedule interval of the scheduled task. The latter specifies how often Discovery-Windows.exe is run on the systems.

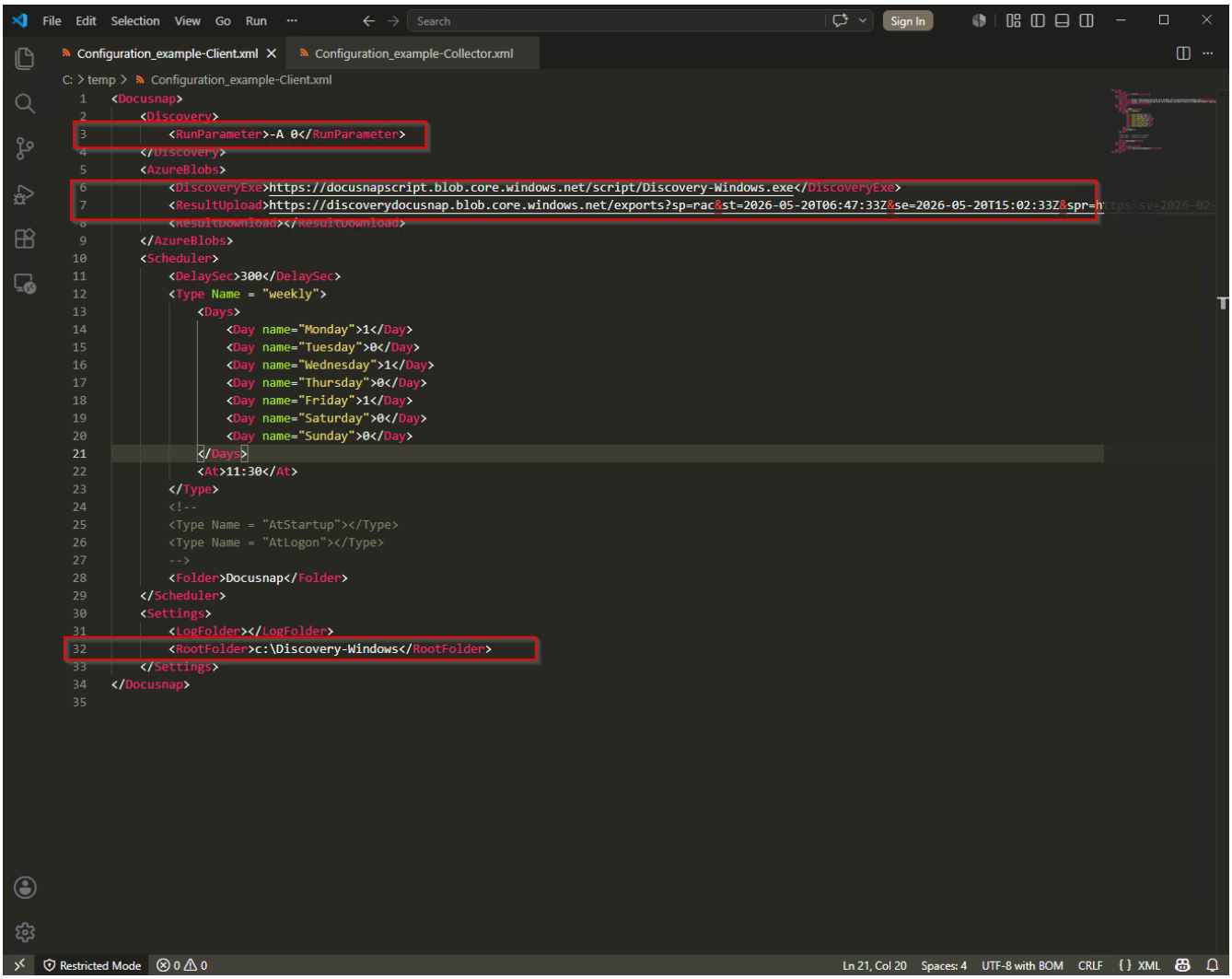

Figure 8 - Parameter Configuration.xml

6. Deployment via Intune

The following describes how to deploy the scripts by using Intune. Alternatively, you can deploy the PowerShell scripts manually or by using other software.

6.1 Create a package

For deployment via Intune, the installation, uninstallation, and execution scripts must be packed together. The download script is not required in the package. For packaging, IntuneWinAppUtil.exe is used.

https://github.com/microsoft/Microsoft-Win32-Content-Prep-Tool/blob/master/IntuneWinAppUtil.exe

The script, the the IntuneWinAppUtil.exe and the Configuration.xml are saved in the same directory. PowerShell is then launched in this directory as an administrator. The following PowerShell command is then run in this directory.

```
$CurrentDirectory = Get-Location
.\IntuneWinAppUtil.exe -c $CurrentDirectory -s Discovery-Windows-Azure.ps1 -o $CurrentDirectory
```

The -c parameter ensures that all files in this folder are compressed into the .intunewin file. When the package is then deployed to the target systems via Intune, the Configuration.xml is also available there.

The Discovery-Windows-Azure.intunewin package is located in the output folder

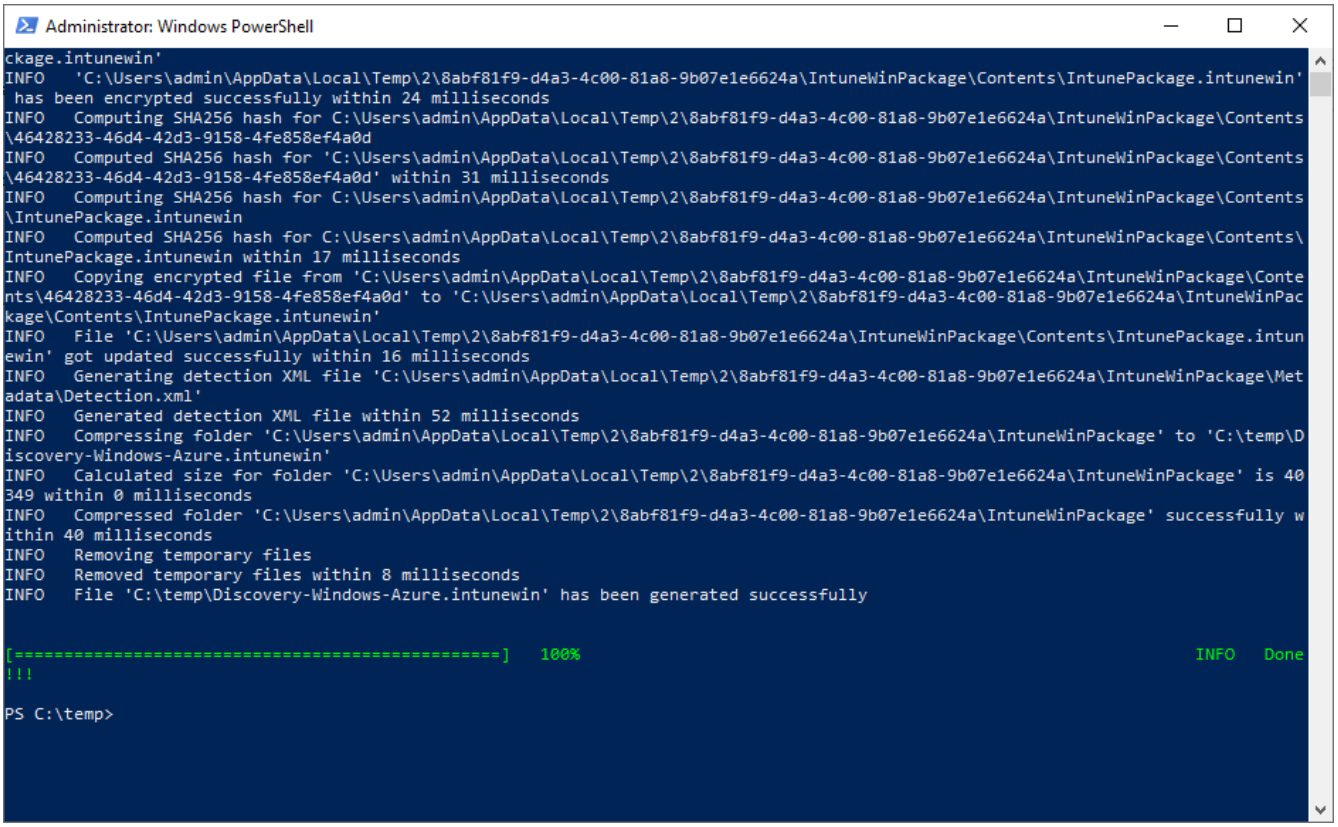

Figure 9 - Running IntuneWinAppUtil.exe

6.2 Deploy package

To deploy the package, a new Win32 application is created in Intune. The previously created Intune package Discovery-Windows-Azure.intunewin must be uploaded in the dialog.

A name and a publisher must be specified for the app information, e.g. Discovery-Windows and Docusnap. Description etc. are optional and freely selectable.

The following installation parameters must be selected:

Install Command:

```
powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -File Discovery-Windows-Azure.ps1 -Install
```

Uninstall Command:

```
powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -File Discovery-Windows-Azure.ps1 -Uninstall
```

Install Behavior:

```
System
```

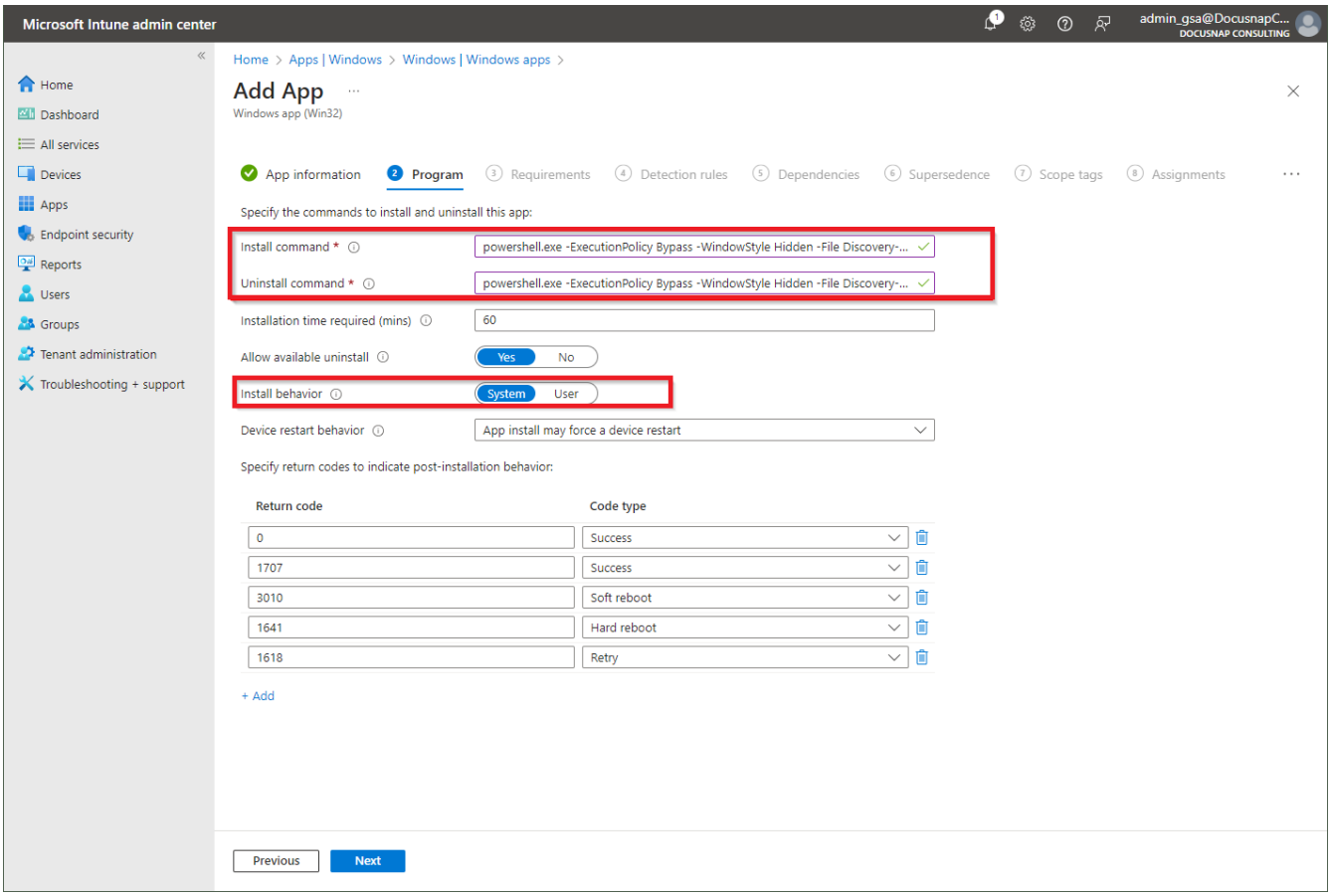

Figure 10 - Intune program options

The Detection Rules check if Run-DocusnapScript.ps1 has been stored in the protected directory. For this purpose, the detection rule is configured as shown in the screenshot.

Attention. If the <Rootfolder> has been adjusted in the configuration file chapter, this must be considered in the detection rules.

Path:

```
C:\Discovery-Windows
```

File or folder:

```
Discovery-Windows-Azure.ps1
```

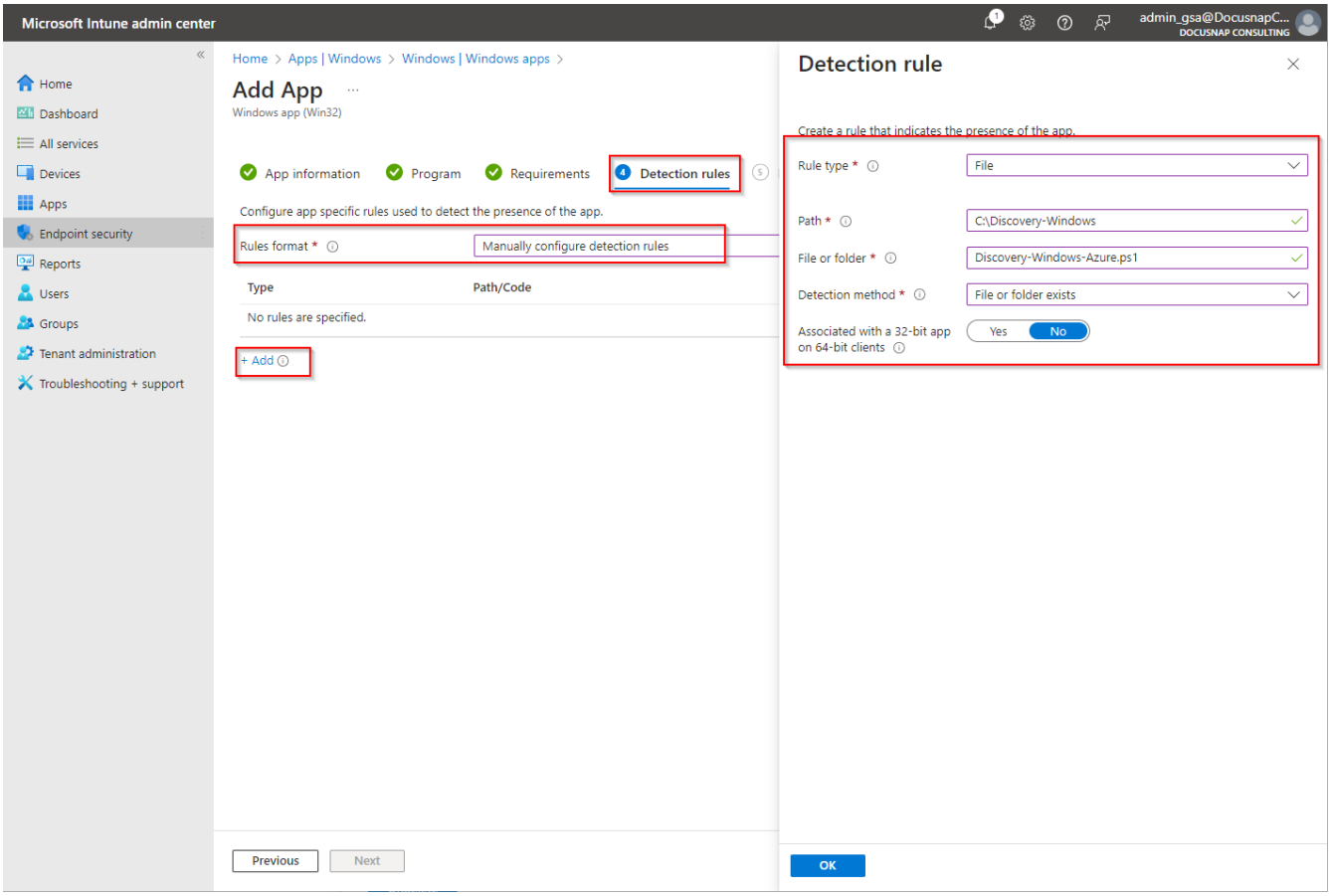

Figure 11 - Intune Detection rule

Otherwise, the application is provided with information at its own discretion and finally assigned to the desired groups.

6.3 Update package

The creation of a new package in Intune is necessary to completely update an older version of the script on each system. This package can have the same name. It is recommended that a new version number be entered in the app information.

To completely remove the outdated version, select the old package in the **supersedence** step and enable the Uninstall previous version option.

The rest of the configuration settings can remain unchanged.

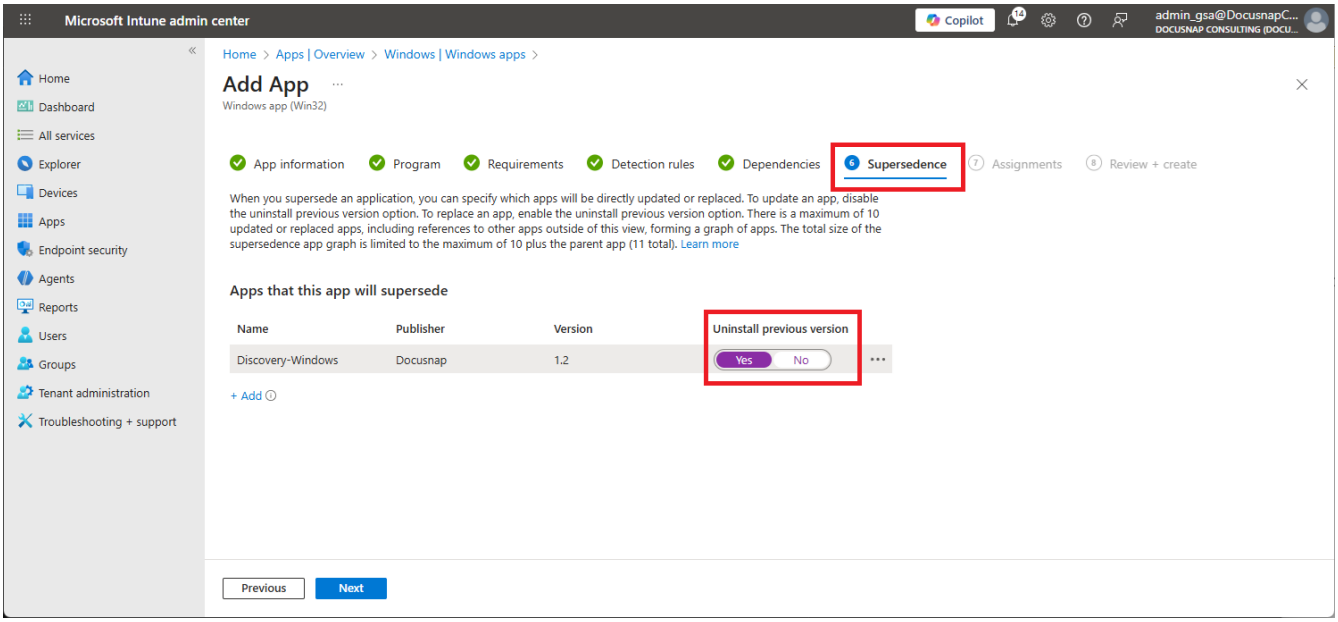

Figure 12 - Replacement of an old script

7. Download for Docusnap

A download script is required to provide the result files of the Discovery-Windows.exe on the Docusnap server. This downloads the result files from the Azure container and makes them available locally on the Docusnap system. The Discovery-Windows-Azure.ps1 script can be used for the creation of the scheduled task. A new configuration file is used for this. The **ResultDownload** parameter must be stored for the ConfigurationDownload.xml of the script.

<ResultDownload>

A new shared access token is created for the container exports. The following authorizations are required. Read access, delete and list. Otherwise the process remains the same.

A new Shared Access Token is created for the container exports. The following permissions are required. **Read**, **delete**, and **list**. Otherwise, the process is the same.

When the scheduled task is executed for the first time, the directory C:\Discovery-Docusnap is automatically created with the two subdirectories **Download** and **Import**. The files that can be imported into Docusnap are stored in the Import directory. This requires a **file import** job in Docusnap. The automation is described in the HowTo Inventory - Discovery Windows - script-based Windows Inventory.

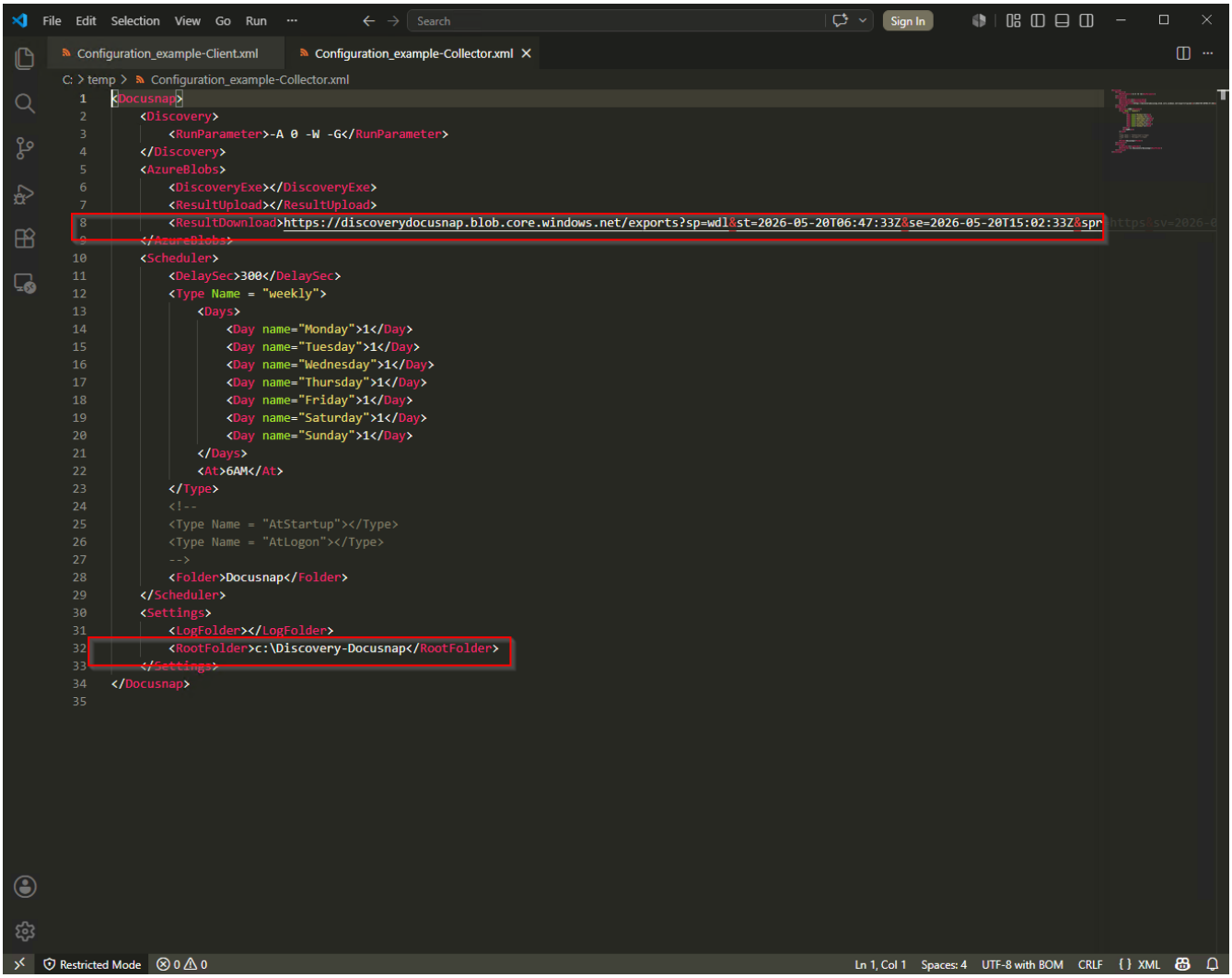

Figure 13 - Parameter DownloadConfiguration.xml

To create the scheduled task, the PowerShell is started in the directory where the Discovery-Windows-Azure.ps1 with the ConfigurationDownload.xml is located.

Make sure that you start the PowerShell in administrator mode and that this is performed on the Docusnap system that executes the automatic import.

```
.\Discovery-Windows-Azure.ps1 -Install -ConfigName ConfigurationDownload.xml
```

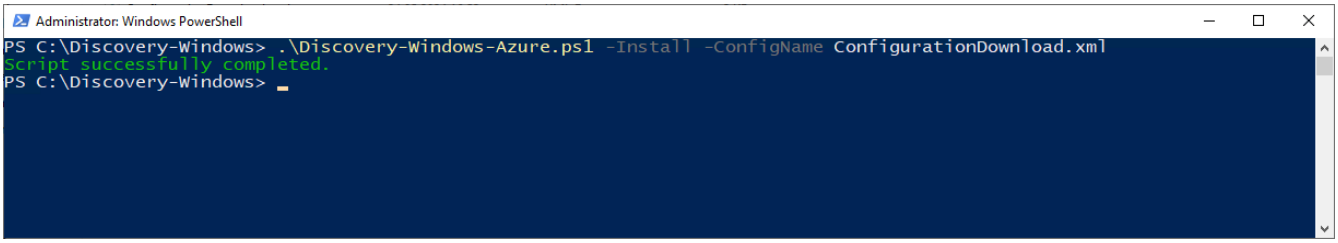

Figure 14 - Automate download

The result files are automatically deleted from the Azure Blob Storage during the download. This ensures that old result files have no impact on the costs of Azure Blob Storage.

LIST OF FIGURES

FIGURE 1 - CREATE A STORAGE ACCOUNT - BASIC SETTINGS 5
FIGURE 2 - CREATE STORAGE ACCOUNT - ADVANCED SETTINGS 6
FIGURE 3 – CONTAINER FOR DISCOVERY-WINDOWS.EXE 7
FIGURE 4 – UPLOADING AND UPDATING THE DISCOVERY-WINDOWS.EXE 8
FIGURE 5 - CONTAINER FOR FEEDBACK FILES 9
FIGURE 6 - CREATING AND AUTHORIZING THE SHARED ACCESS TOKEN 10
FIGURE 7 - DISCOVERY-WINDOWS.EXE URL 12
FIGURE 8 - PARAMETER CONFIGURATION.XML 13
FIGURE 9 - RUNNING INTUNEWINAPPUTIL.EXE 14
FIGURE 10 - INTUNE PROGRAM OPTIONS 15
FIGURE 11 - INTUNE DETECTION RULE 16
FIGURE 12 - REPLACEMENT OF AN OLD SCRIPT 17
FIGURE 13 - PARAMETER DOWNLOADCONFIGURATION.XML 18
FIGURE 14 - AUTOMATE DOWNLOAD 19

Version history

Date	Description
05/08/2026	Version 1.0 created