



## Docusnap und Intune

*Automatisierung der Discovery-Windows.exe via Intune*

<b>TITEL</b>	Docusnap und Intune
<b>AUTOR</b>	Docusnap Consulting
<b>DATUM</b>	23.01.2026
<b>VERSION</b>	1.2   gültig ab 23.01.2026

Die Weitergabe, sowie Vervielfältigung dieser Unterlage, auch von Teilen, Verwertung und Mitteilung ihres Inhaltes ist nicht gestattet, soweit nicht ausdrücklich durch die Docusnap GmbH zugestanden. Zuwiderhandlung verpflichtet zu Schadenersatz. Alle Rechte vorbehalten.

This document contains proprietary information and may not be reproduced in any form or parts whatsoever, nor may be used by or its contents divulged to third parties without written permission of Docusnap GmbH. All rights reserved.

## INHALTSVERZEICHNIS

1.	Einleitung	4
2.	Voraussetzungen	4
3.	Azure Storage Account	5
4.	Container und Zugriff	7
4.1	Container für die Discovery-Windows.exe	7
4.2	Container für die Dateiablage	9
4.3	Shared Access Token	10
5.	Vorbereiten der Inventarisierung	11
5.1	Skript Parameter	11
5.2	Konfigurationsdatei	12
6.	Verteilung via Intune	14
6.1	Paket erstellen	14
6.2	Paket verteilen	15
6.3	Paket aktualisieren	17
7.	Download für Docusnap	18

## 1. Einleitung

Mit zunehmender Nutzung von Microsoft Azure sind Windows-Systeme nicht immer Teil des Active Directory. Selbstverständlich sind diese Systeme für eine vollständige Übersicht und Dokumentation relevant. Die automatische Inventarisierung genau dieser Systeme gestaltet sich oft schwierig.

In diesem HowTo wird beschrieben, wie man die Discovery-Windows.exe über Intune automatisieren kann. Dabei werden die Voraussetzungen und die Automatisierung der Ausführung beschrieben.

## 2. Voraussetzungen

Für die Implementierung der Lösung sind verschiedene Rahmenbedingungen notwendig. Dies betrifft die technischen Voraussetzungen sowie die Berechtigungen und Kenntnisse in Azure und Intune.

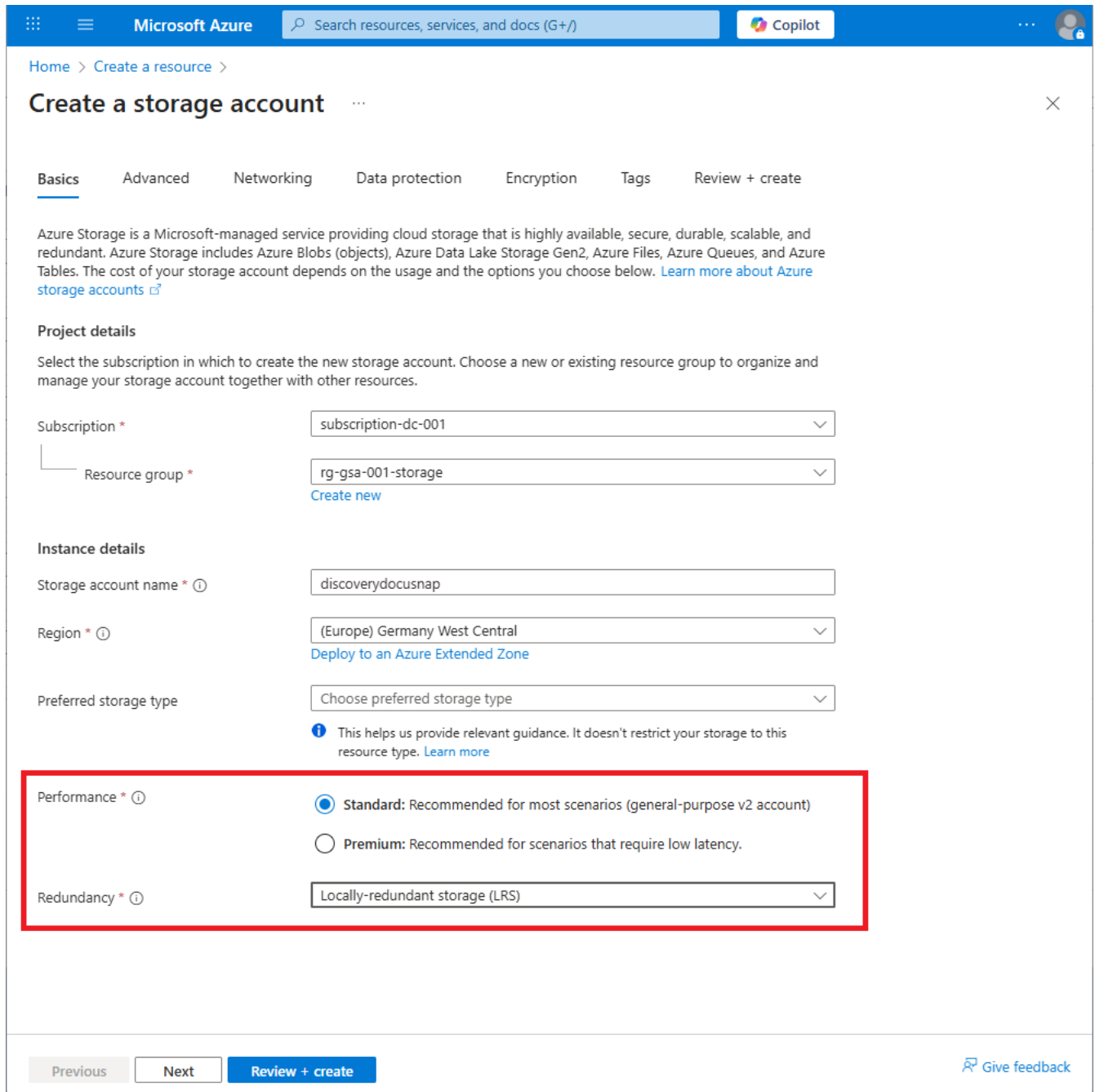
- Docusnap onPrem
- Entsprechende Berechtigungen in Intune und Azure
- Intune zur Automatisierung der Discovery-Windows.exe
- Azure Subscription
- Azure Storage Account
  - Bereitstellen der Discovery-Windows.exe
  - Ablage der Feedback Dateien
- Shared Access Token
- PowerShell Skript – Download Link in Kapitel [Vorbereiten der Inventarisierung](#)

**Achtung: Bei dieser Lösung entstehen Kosten in Ihrer Azure Subscription. Je nach Konfiguration und Verwendung können diese unterschiedlich hoch ausfallen. Um vorab diese kalkulieren zu können steht Ihnen der Microsoft Azure Preisrechner zur Verfügung.**

### 3. Azure Storage Account

Für die Bereitstellung wird ein Azure Storage Account benötigt. In diesem werden die Discovery-Windows.exe und die Ergebnisdateien der inventarisierten Systeme abgelegt. Es sollte ein eigener Storage Account erstellt, und kein bestehender verwendet werden.

Die Bezeichnung ist frei wählbar. Um Kosten zu sparen, sollte der Storage Account für die Discovery-Windows.exe das Performancelevel Standard und die Redundanzstufe LRS verwenden.



Microsoft Azure Search resources, services, and docs (G+/) Copilot

Home > Create a resource >

## Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

**Project details**

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \* subscription-dc-001

Resource group \* rg-gsa-001-storage [Create new](#)

**Instance details**

Storage account name \* ① discoverydocusnap

Region \* ① (Europe) Germany West Central [Deploy to an Azure Extended Zone](#)

Preferred storage type Choose preferred storage type

**Performance** \* ①

☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)

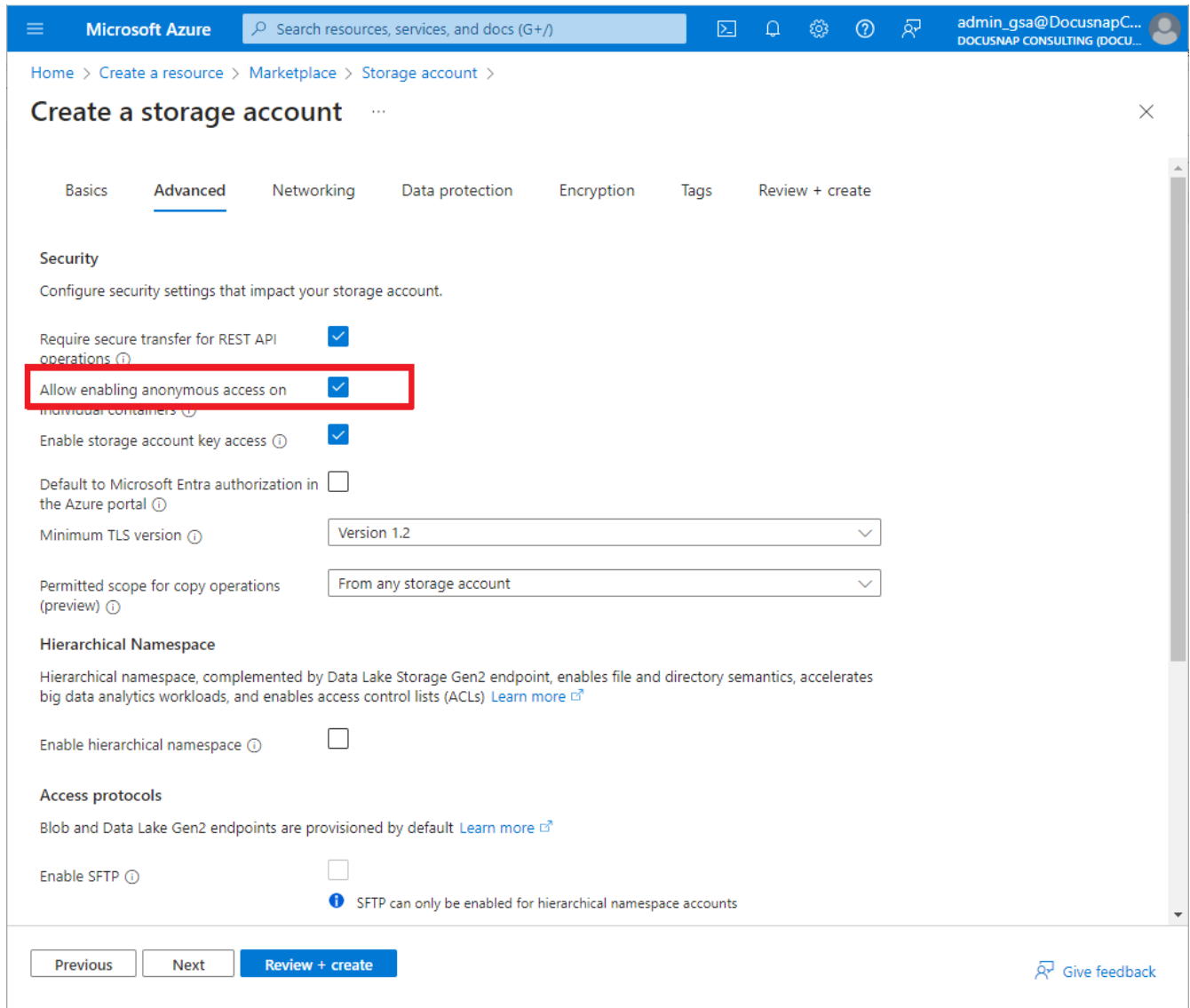
☐ **Premium:** Recommended for scenarios that require low latency.

**Redundancy** \* ① Locally-redundant storage (LRS)

Previous Next Review + create [Give feedback](#)

Abbildung 1 - Storage Account erstellen - Grundeinstellungen

In den erweiterten Einstellungen muss der anonyme Zugriff aktiviert werden. Dadurch können Clients die Discovery-Windows.exe herunterladen und die XML-Dateien hochladen. Die anderen Einstellungen bleiben auf den Standardwerten.



The screenshot shows the 'Create a storage account' wizard in the Microsoft Azure portal, specifically the 'Advanced' tab. The 'Security' section is active, and the checkbox for 'Allow enabling anonymous access on individual containers' is checked and highlighted with a red rectangle. Other settings like 'Require secure transfer for REST API operations' and 'Enable storage account key access' are also checked. The 'Minimum TLS version' is set to 'Version 1.2' and the 'Permitted scope for copy operations' is 'From any storage account'. The 'Hierarchical Namespace' section is currently disabled. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons.

**Microsoft Azure** Search resources, services, and docs (G+/)

Home > Create a resource > Marketplace > Storage account >

## Create a storage account

Basics **Advanced** Networking Data protection Encryption Tags Review + create

### Security

Configure security settings that impact your storage account.

Require secure transfer for REST API operations ☒

**Allow enabling anonymous access on individual containers** ☒

Enable storage account key access ☒

Default to Microsoft Entra authorization in the Azure portal ☐

Minimum TLS version

Permitted scope for copy operations (preview)

### Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) [Learn more](#)

Enable hierarchical namespace ☐

### Access protocols

Blob and Data Lake Gen2 endpoints are provisioned by default [Learn more](#)

Enable SFTP ☐

**SFTP can only be enabled for hierarchical namespace accounts**

[Previous](#) [Next](#) [Review + create](#)

[Give feedback](#)

Abbildung 2 - Storage Account erstellen - Erweitert

## 4. Container und Zugriff

Nach der Einrichtung des Storage Accounts werden zwei Container angelegt.

### 4.1 Container für die Discovery-Windows.exe

Der erste Container enthält die Datei Discovery-Windows.exe, die anonym heruntergeladen werden muss. Aus diesem Grund wird bei der Erstellung des Containers die anonyme Zugriffsebene auf Blob gesetzt.

Der Name ist frei wählbar. In unserem Beispiel wird der Name `script` gewählt. Im neu erstellten Container `script` wird die Discovery-Windows.exe hochgeladen.

**Achtung:** Der Name wird in der Configuration.xml des Skripts verwendet. Sollte dieser angepasst werden, muss das in der Configuration.xml ebenfalls berücksichtigt werden.

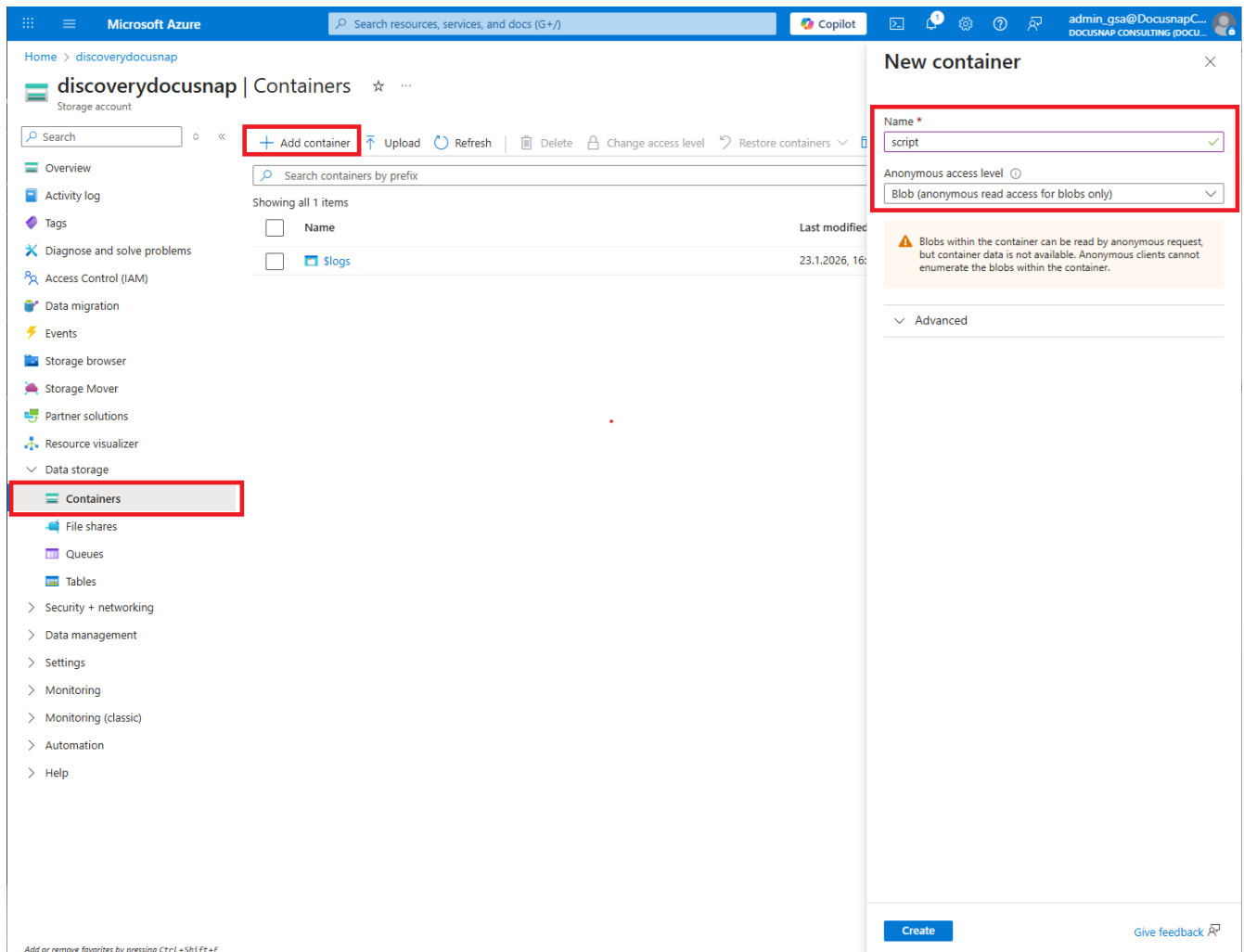


Abbildung 3 - Container für Discovery-Windows.exe

Nach einem Update von Docusnap steht eine neue Version der Discovery-Windows.exe zur Verfügung. Diese sollte zeitnah in den Container geladen werden. Die Clients laden die Discovery-Windows.exe bei jeder Inventarisierung neu aus dem Container. Dadurch ist auf den Clients keine Anpassung notwendig.

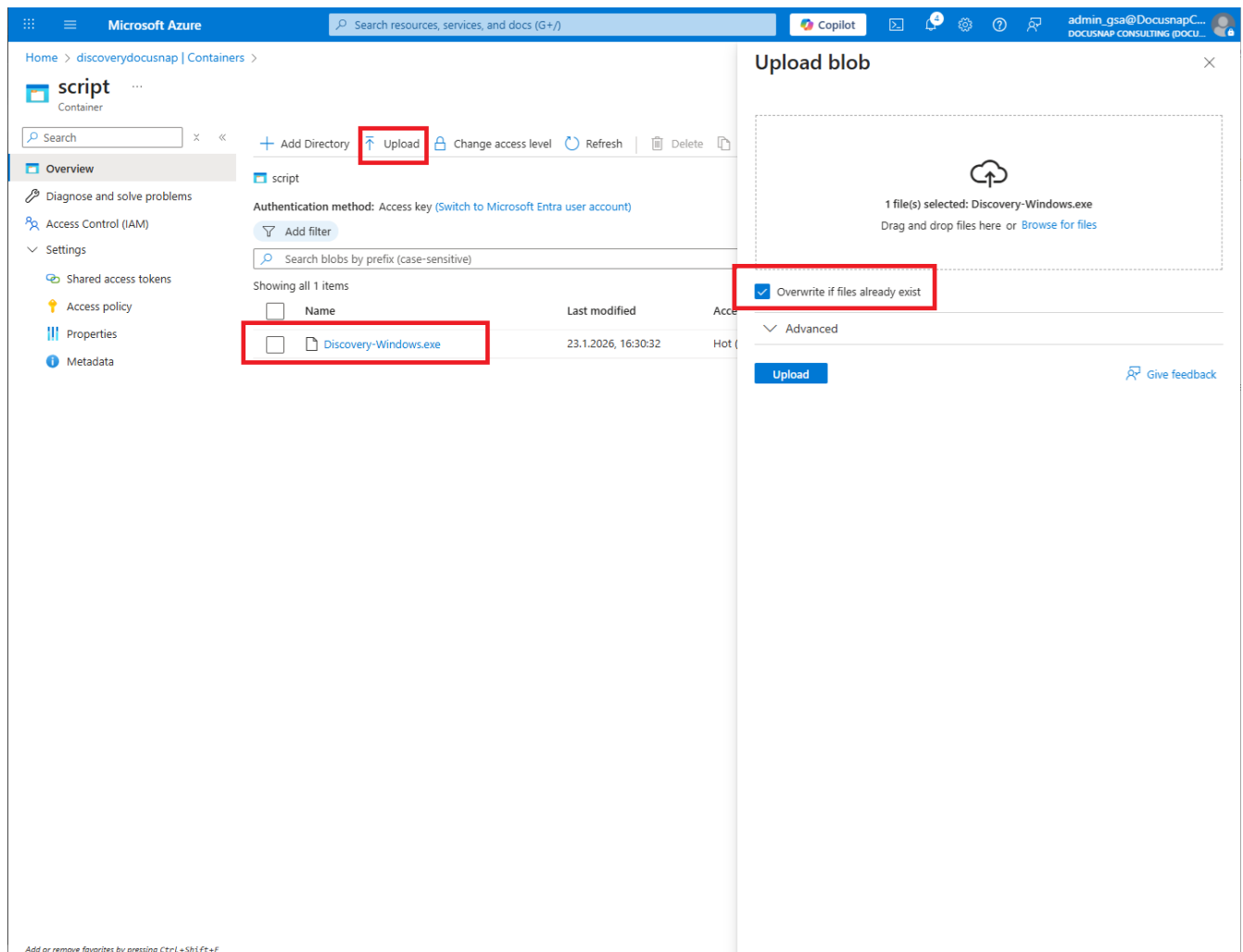


Abbildung 4 - Upload / Aktualisieren der Discovery-Windows.exe



## 4.2 Container für die Dateiablage

Der zweite Container dient als Upload-Verzeichnis für die Feedback-Dateien der Inventarisierung.

Die anonyme Zugriffsebene ist hier auf **Private** gesetzt. Als Name wurde in unserem Beispiel **exports** verwendet. Auch hier kann der Name frei gewählt werden.

**Achtung:** Der Name wird in der Configuration.xml des Skripts verwendet. Sollte dieser angepasst werden, muss das in der Configuration.xml ebenfalls berücksichtigt werden.

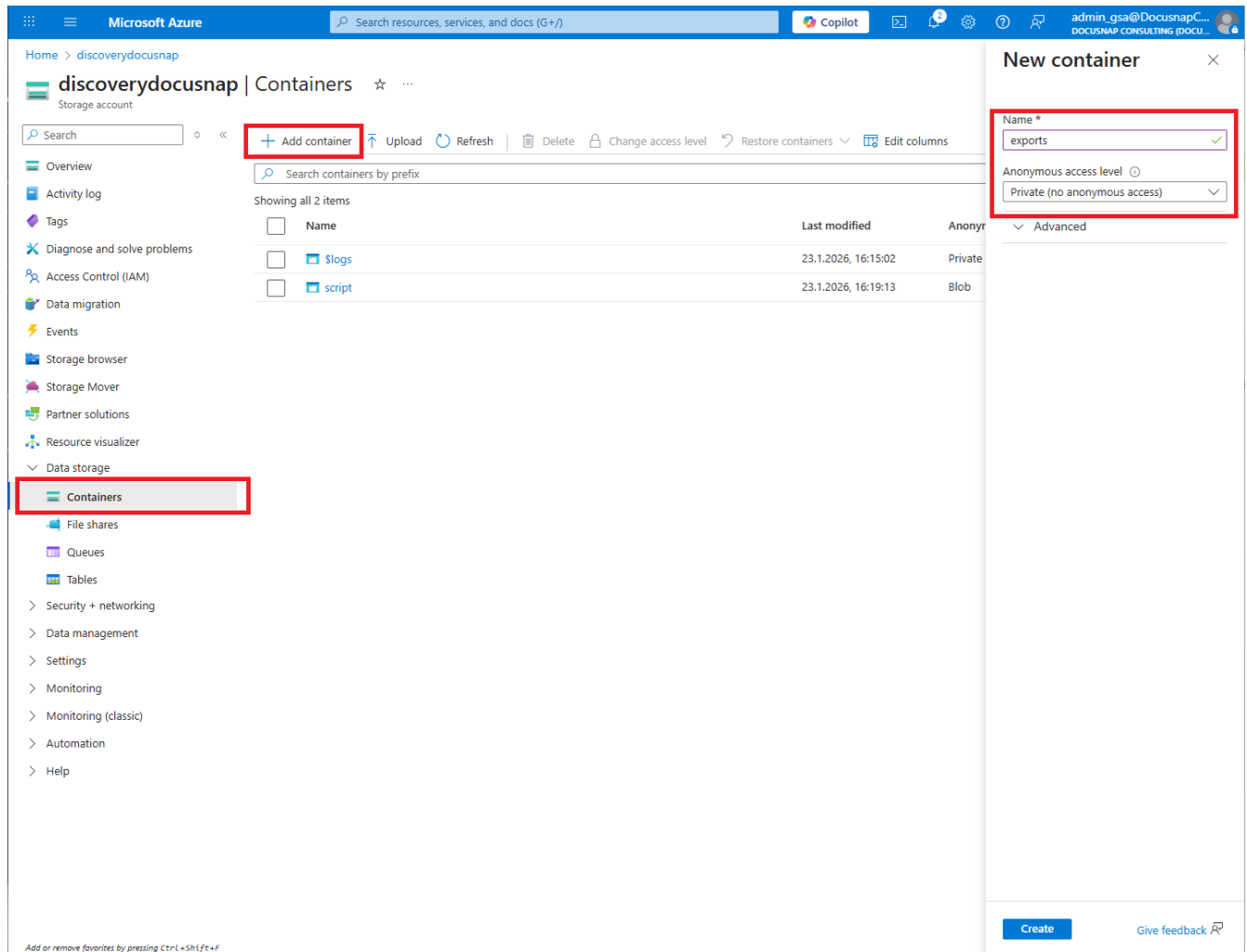


Abbildung 5 - Container für Feedback Dateien

### 4.3 Shared Access Token

Damit Clients die exportierten Daten ohne Authentifizierung im Container ablegen können, wird ein Shared Access Token benötigt. Dieser muss erstellt werden. Dabei ist darauf zu achten, dass der Token nur das Schreiben, nicht aber das Lesen von Dateien erlaubt. Bei Verlust des Tokens an Dritte ist somit sichergestellt, dass diese zwar Daten in den Container schreiben, aber keine Inhalte auslesen können.

Im Container **exports** wird ein Shared Access Token mit den Berechtigungen **Hinzufügen, Erstellen und Schreiben** erstellt.

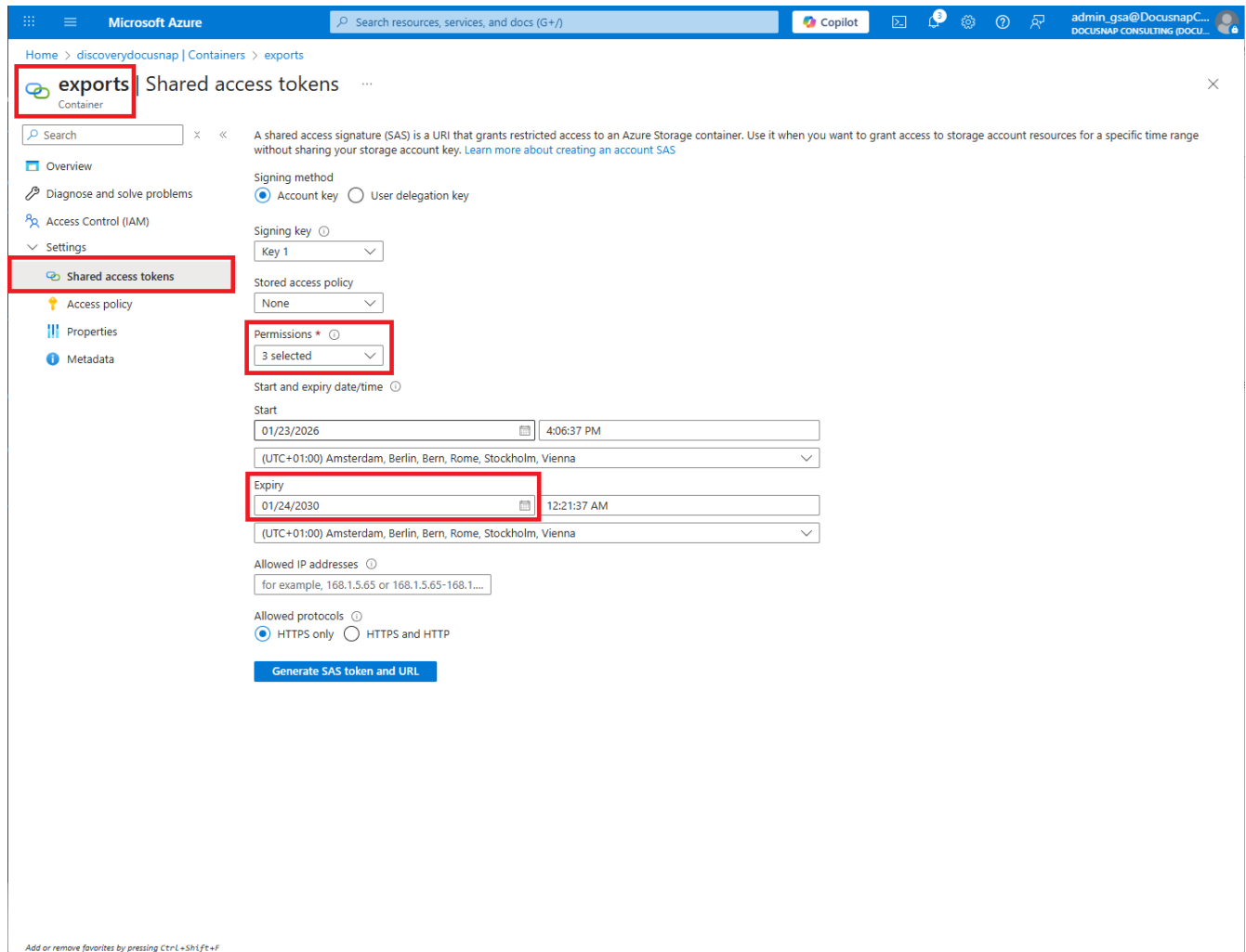


Abbildung 6 - Erstellen & Berechtigen des Shared Access Tokens

Das Shared Access Token hat ein Ablaufdatum. Da der Token fest im Skript hinterlegt ist, muss bei dessen Ablauf das Skript angepasst, neu paketiert und verteilt werden. Aus diesem Grund wird eine lange Gültigkeitsdauer empfohlen.

Mit Klick auf **SAS-Token und -URL generieren** wird das SAS Blob Token und die SAS Blob URL angezeigt. Nachdem das Token generiert wurde, muss die Blob-SAS-URL kopiert und **zwischengespeichert** werden.

**Die Blob-SAS-URL wird für das Ausführungsskript benötigt. Geht die Blob-SAS-URL verloren, muss ein neuer SAS-Token erstellt werden.**

## 5. Vorbereiten der Inventarisierung

Das Skript erstellt auf jedem ausgerollten System das Verzeichnis C:\Discovery-Windows und setzt die Berechtigungen für Administratoren sowie für das Systemkonto. Existiert das Verzeichnis bereits, werden die bestehenden Berechtigungen überschrieben.

Für die Automatisierung wird das PowerShell Skript `Discovery-Windows-Azure` verwendet. Dieses hat mehrere Funktionen, die via Parameter aufgerufen werden. Die Einstellungen sowie die Informationen zum SAS-Token werden in einer `Configuration.xml` hinterlegt.

Das PowerShell Skript und die `Configuration.xml` kann von unserem [Download Portal](#) heruntergeladen werden.

### 5.1 Skript Parameter

Es stehen für den Aufruf folgende Parameter zur Verfügung.

#### Install

kopiert dieses Skript sowie die Konfigurationsdatei an einen lokalen Ablageort und konfiguriert eine Windows-Aufgabe für eine regelmäßigen Ausführung des Skripts mit der Parameter `-Run`

#### Run

führt die `Discovery-Windows.exe` zur aus und lädt die Ergebnisdatei auf den konfigurierten Azure Storage-Account hoch. Die Konfigurationsdatei wird unter dem Namen `Configuration.xml` im selben Verzeichnis wie die Skript-Datei erwartet.

#### Collect

lädt alle Feedback-Dateien vom Azure Storage Account herunter und stellt diese in einem zentralen Verzeichnis zur Verfügung. Die Feedback-Dateien werden im Azure Storage gelöscht.

#### Uninstall

deinstalliert sowohl das Skript als auch die angelegte Windows-Aufgabe.

#### ConfigName

(optional) Wird eine Konfigurationsdatei verwendet, welche vom Standardnamen `Configuration.xml` abweicht, kann dieser Name mit dem Parameter `-ConfigName` angegeben werden. Die Datei muss sich im selben Verzeichnis wie auch die Skript-Datei befinden.

#### Verbose

zeigt zusätzliche Informationen zu den Einzelschritten des Skripts an. Parameter dient zur Problemanalyse.

## 5.2 Konfigurationsdatei

Zuerst wird das Skript für die Inventarisierung konfiguriert. Die Konfiguration des Download Skripts für den Docusnap Server bzw. einen Docusnap Discovery Service wird im Kapitel [Download für Docusnap](#) erläutert.

Für die Configuration.xml des Skripts müssen die Parameter **DiscoveryExe** und **ResultUpload** hinterlegt werden.

### <RunParameter>

Angabe der Parameter für die Discovery-Windows.exe. Im Standard sind die Parameter -A 0, -W und -G hinterlegt. Es werden somit keine Archiv Versionen der Feedback Dateien gespeichert. Ebenfalls wird die Discovery-Windows.exe im Silent Modus ausgeführt und keine Netzwerkverbindungen erfasst. Die Parameter können via Parameter **Discovery-Windows.exe -H** angezeigt werden.

### <DiscoveryExe>

Kann in den Eigenschaften der Discovery-Windows.exe im entsprechenden Azure Container kopiert werden. Die Eigenschaften werden über das Kontextmenü der Discovery-Windows.exe aufgerufen. Die URL setzt sich aus dem Namen des Storage Accounts und des Containers zusammen.

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

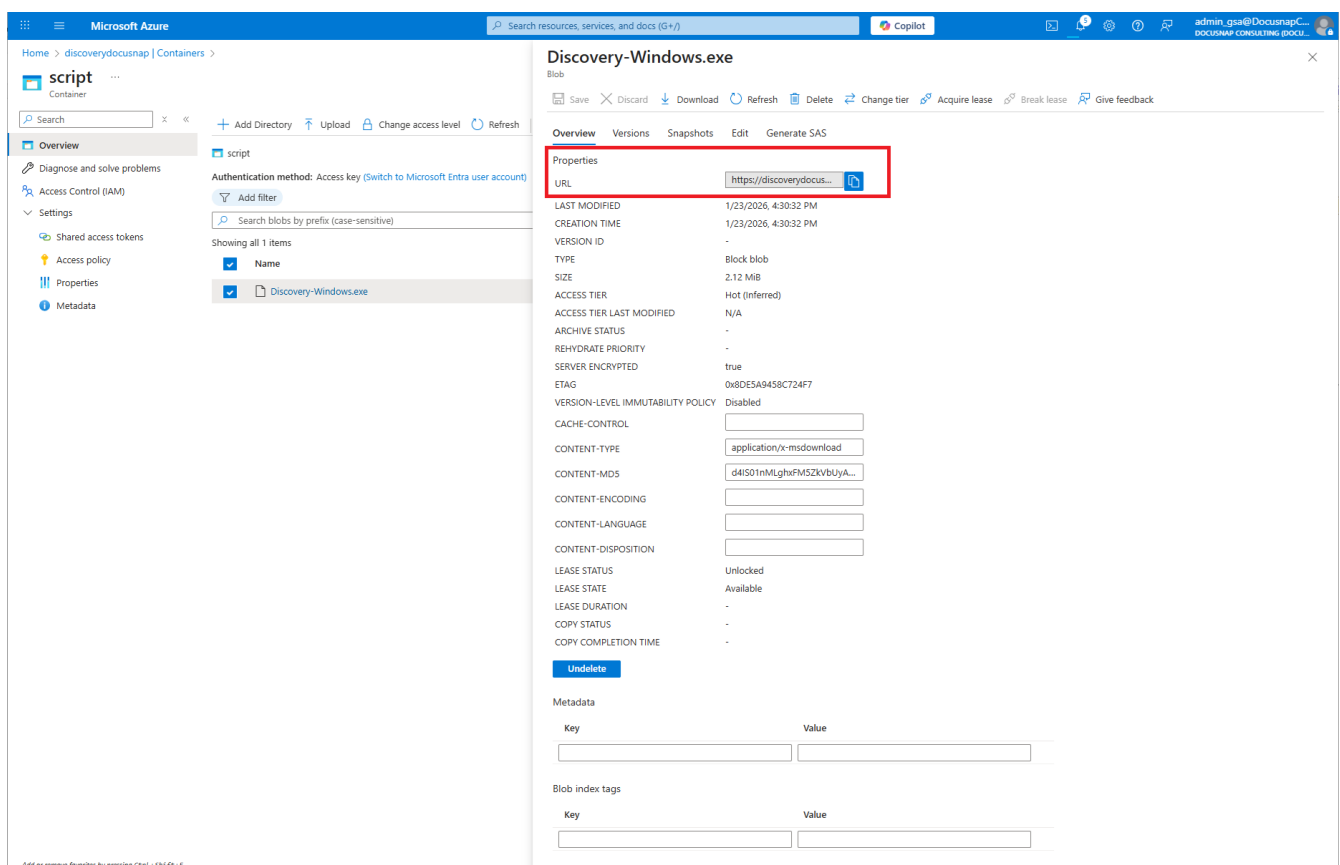


Abbildung 7 - Discovery-Windows URL

<ResultUpload>

Blob-SAS-URL aus dem Kapitel [Shared Access Token](#).

Optional können die Parameter der Discovery-Windows.exe sowie das Zeitintervall des geplanten Tasks angepasst werden. Letzteres gibt an, wie oft die Discovery-Windows.exe auf den Systemen ausgeführt wird.

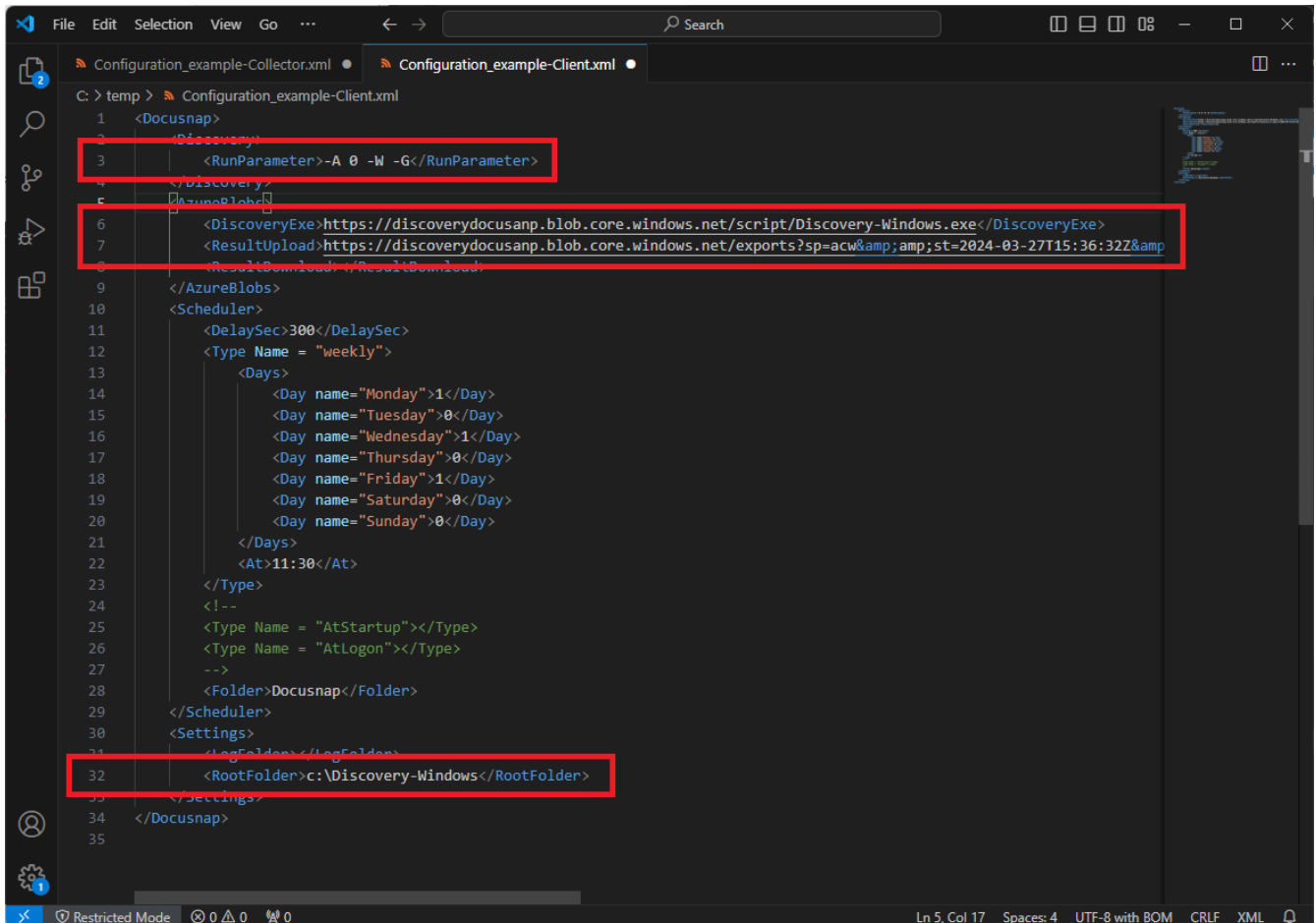


Abbildung 8 - Parameter Configuration.xml

## 6. Verteilung via Intune

Im Folgenden wird die Verteilung des Skripts über Intune beschrieben. Alternativ kann diese auch über andere Softwarelösungen verteilt werden.

### 6.1 Paket erstellen

Für die Verteilung über Intune muss das Skript inklusive der nötigen Parameter zusammengepackt werden. Für die Paketierung wird `IntuneWinAppUtil.exe` verwendet.

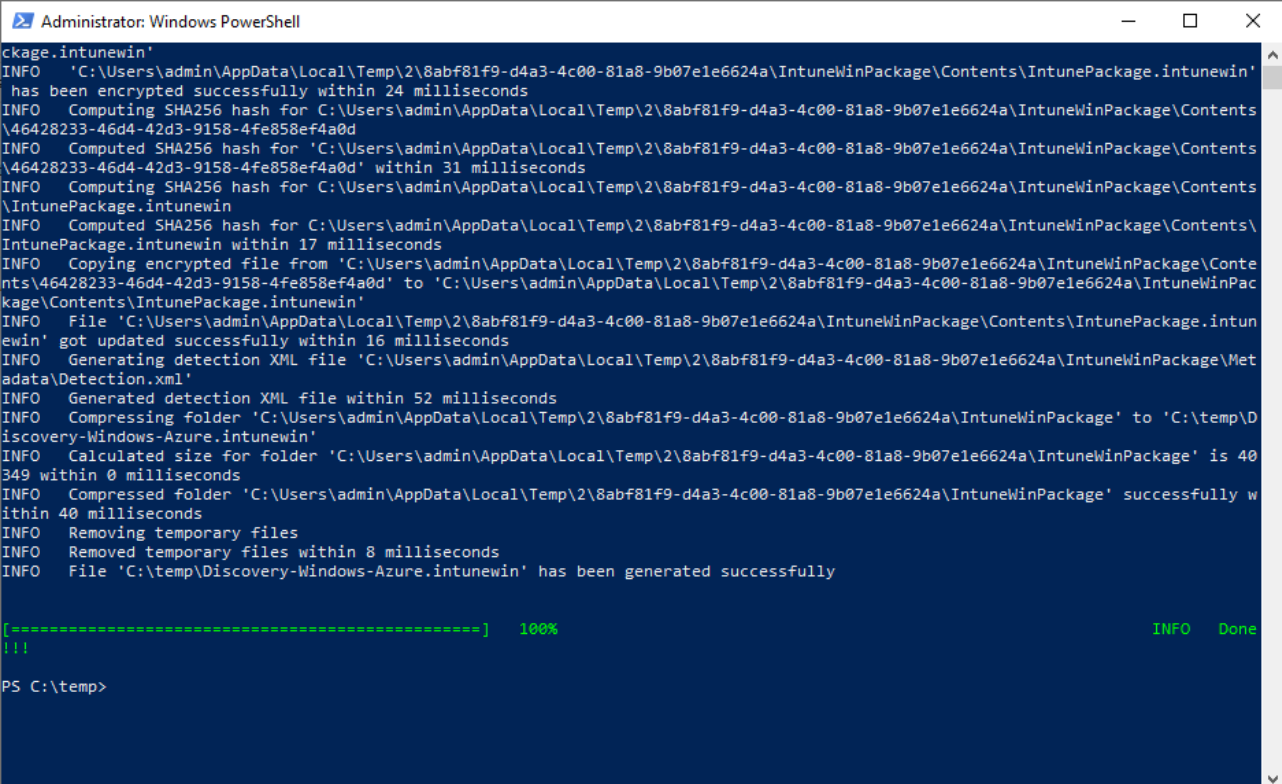
<https://github.com/microsoft/Microsoft-Win32-Content-Prep-Tool/blob/master/IntuneWinAppUtil.exe>

Das Skript, die `IntuneWinAppUtil.exe` und die `Configuration.xml` werden im selben Verzeichnis gespeichert. Anschließend wird die PowerShell in diesem Verzeichnis als Administrator gestartet. In diesem Verzeichnis wird dann der folgende PowerShell Befehl ausgeführt.

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

Der Parameter `-c` sorgt dafür, dass alle Dateien in diesem Ordner in die `.intunewin`-Datei komprimiert werden. Wird das Paket anschließend via Intune auf die Zielsysteme verteilt, ist die `Configuration.xml` auch dort vorhanden.

Im Ausgabeordner befindet sich das Paket `Discovery-Windows-Azure.intunewin`



```
Administrator: Windows PowerShell
ckage.intunewin'
INFO 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents\IntunePackage.intunewin'
has been encrypted successfully within 24 milliseconds
INFO Computing SHA256 hash for C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents
\46428233-46d4-42d3-9158-4fe858ef4a0d
INFO Computed SHA256 hash for 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents
\46428233-46d4-42d3-9158-4fe858ef4a0d' within 31 milliseconds
INFO Computing SHA256 hash for C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents
\IntunePackage.intunewin
INFO Computed SHA256 hash for C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents
\IntunePackage.intunewin within 17 milliseconds
INFO Copying encrypted file from 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Conte
nts\46428233-46d4-42d3-9158-4fe858ef4a0d' to 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPac
kage\Contents\IntunePackage.intunewin'
INFO File 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Contents\IntunePackage.intun
ewin' got updated successfully within 16 milliseconds
INFO Generating detection XML file 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage\Met
adata\Detection.xml'
INFO Generated detection XML file within 52 milliseconds
INFO Compressing folder 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage' to 'C:\temp\D
iscovery-Windows-Azure.intunewin'
INFO Calculated size for folder 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage' is 40
349 within 0 milliseconds
INFO Compressed folder 'C:\Users\admin\AppData\Local\Temp\2\8abf81f9-d4a3-4c00-81a8-9b07e1e6624a\IntuneWinPackage' successfully w
ithin 40 milliseconds
INFO Removing temporary files
INFO Removed temporary files within 8 milliseconds
INFO File 'C:\temp\Discovery-Windows-Azure.intunewin' has been generated successfully

[=====] 100% INFO Done
!!!
PS C:\temp>
```

Abbildung 9 - IntuneWinAppUtil.exe Aufruf

## 6.2 Paket verteilen

Um das Paket zu verteilen, wird in Intune eine neue Win32-Anwendung erstellt. Dazu wird die zuvor erstellte Intune Paketdatei **Discovery-Windows-Azure.intunewin** im Dialog hochgeladen.

Bei den App-Informationen muss ein Name und ein Herausgeber angegeben werden, z.B. Discovery-Windows und Docusnap. Beschreibung etc. sind optional und frei wählbar.

Folgende Installationsparameter müssen ausgewählt werden

### Install Command:

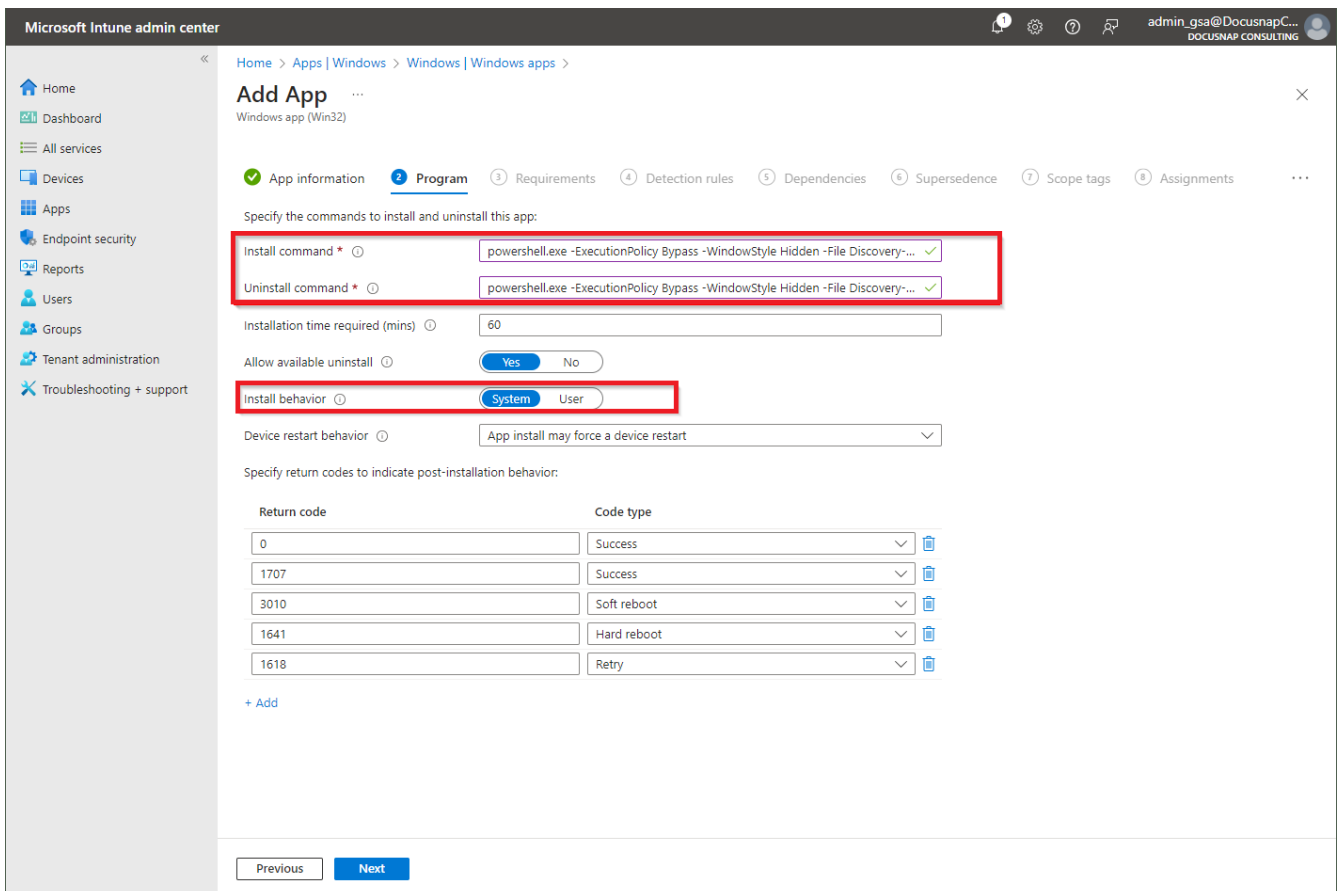
```
powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -File Discovery-Windows-Azure.ps1 -install
```

### Uninstall Command:

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

### Install Behavior:

System



Microsoft Intune admin center

Home > Apps | Windows > Windows | Windows apps >

### Add App

Windows app (Win32)

App information **Program** Requirements Detection rules Dependencies Supersedeance Scope tags Assignments

Specify the commands to install and uninstall this app:

Install command \*

Uninstall command \*

Installation time required (mins)

Allow available uninstall ☒ Yes ☐ No

Install behavior **System** ☐ User

Device restart behavior

Specify return codes to indicate post-installation behavior:

Return code	Code type
<input type="text" value="0"/>	<input type="text" value="Success"/>
<input type="text" value="1707"/>	<input type="text" value="Success"/>
<input type="text" value="3010"/>	<input type="text" value="Soft reboot"/>
<input type="text" value="1641"/>	<input type="text" value="Hard reboot"/>
<input type="text" value="1618"/>	<input type="text" value="Retry"/>

[+ Add](#)

Abbildung 10 - Intune Programm Optionen

Unter Erkennungsregeln wird geprüft, ob Discovery-Windows-Azure.ps1 im geschützten Verzeichnis abgelegt wurde. Dazu wird die Erkennungsregel wie im Screenshot gezeigt konfiguriert.

Achtung. Sollte der <Rootfolder> in Kapitel [Konfigurationsdatei](#) angepasst worden sein, muss das in den Erkennungsregeln berücksichtigt werden.

**Pfad:**

C:\Discovery-Windows

**Datei oder Ordner:**

Discovery-Windows-Azure.ps1

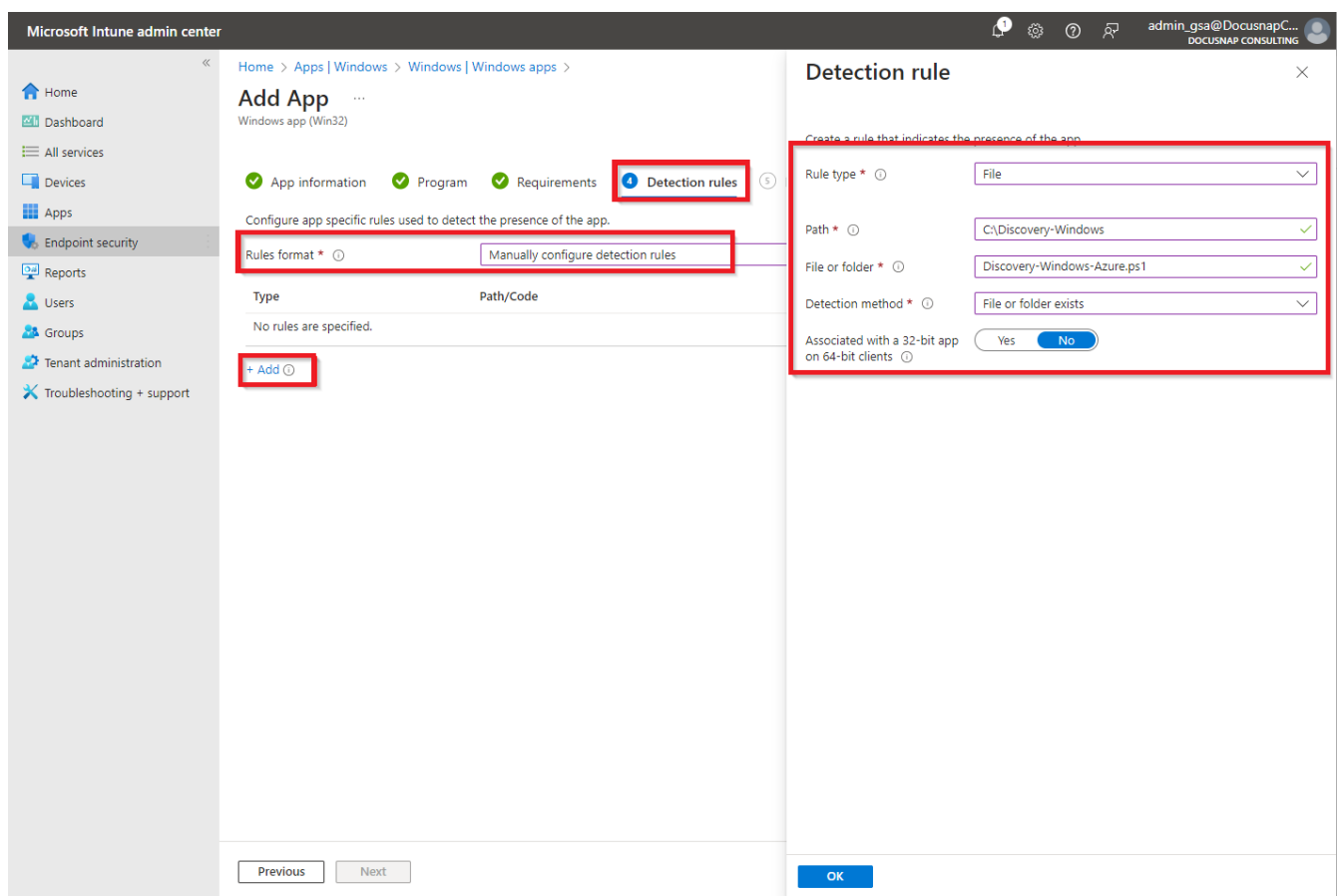


Abbildung 11 - Intune Erkennungsregeln

Ansonsten wird die Anwendung nach eigenem Ermessen mit Informationen gefüllt und den gewünschten Gruppen zugeordnet.



## 6.3 Paket aktualisieren

Um eine ältere Version des Skripts auf jedem System vollständig aktualisieren zu können, muss in Intune ein neues Paket erstellt werden. Das Paket darf denselben Namen haben. In den App-Informationen empfiehlt es sich, eine neue Versionsnummer zu hinterlegen.

Um die veraltete Version vollständig zu entfernen, muss im Schritt **Ablösung** das alte Paket ausgewählt und die Option **Vorherige Version deinstallieren** aktiviert werden.

Die übrige Konfiguration bleibt unverändert.

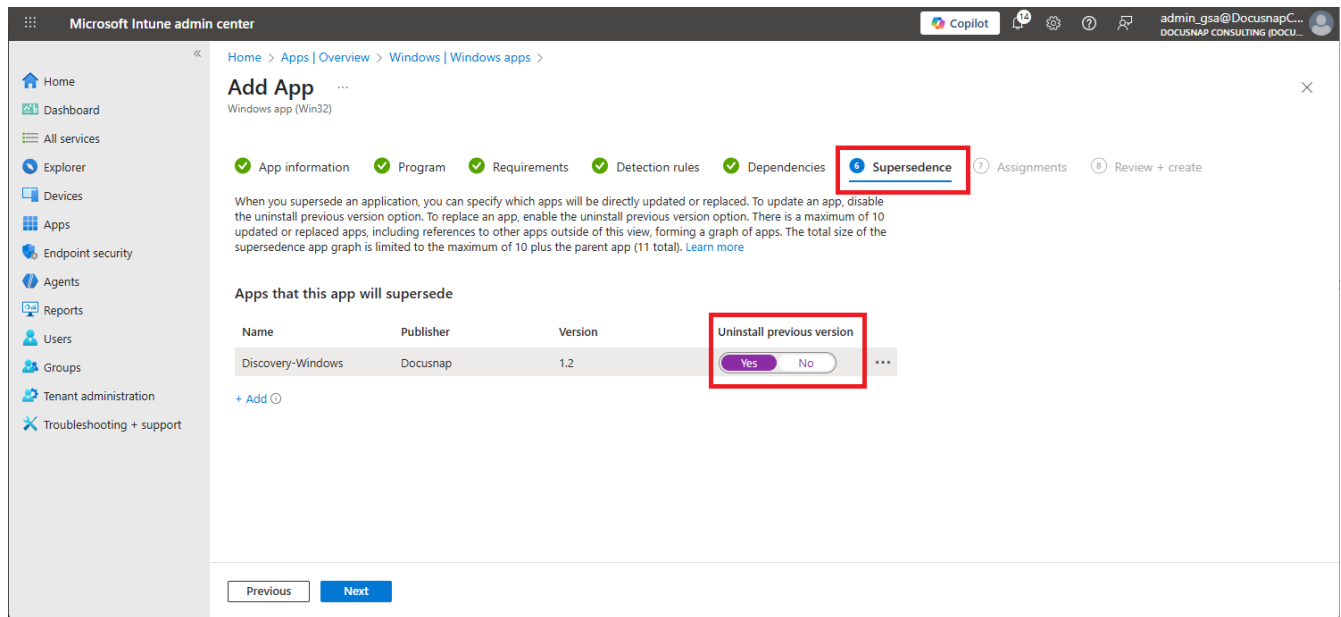


Abbildung 12 - Ablösung eines alten Skripts

## 7. Download für Docusnap

Damit die Ergebnisdateien der Discovery-Windows.exe auf dem Docusnap Server verfügbar sind, wird ein Download-Skript benötigt. Dieses lädt die Ergebnisdateien aus dem Azure Container herunter und stellt sie lokal auf dem Docusnap system zur Verfügung. Für die Erstellung des geplanten Tasks kann wieder das Skript Discovery-Windows-Azure.ps1 verwendet werden. Dazu wird eine neue Konfigurationsdatei verwendet. Für die ConfigurationDownload.xml des Skripts muss der Parameter ResultDownload hinterlegt werden.

### <ResultDownload>

Es wird ein neues [Shared Access Token](#) für den Container exports erstellt. Folgende Berechtigungen werden benötigt. Lesezugriff, Löschen und Liste. Ansonsten bleibt der Prozess gleich.

Optional kann das Zeitintervall des geplanten Tasks angepasst werden. Dieses gibt an, wie oft die Dateien aus dem Azure Blob Storage heruntergeladen werden. Für das Download-Skript wurde der Basispfad auf C:\Discovery-Docusnap geändert.

Beim erstmaligen Ausführen der geplanten Aufgabe wird automatisch das Verzeichnis C:\Discovery-Docusnap mit den beiden Unterverzeichnissen Download und Import angelegt. Im Verzeichnis Import werden die Dateien abgelegt, die in Docusnap importiert werden können. Dafür ist ein Dateiimport Auftrag in Docusnap nötig. Die Automatisierung ist im [HowTo Inventarisierung - Discovery-Windows - Skriptbasierte Windows-Inventarisierung](#) beschrieben.

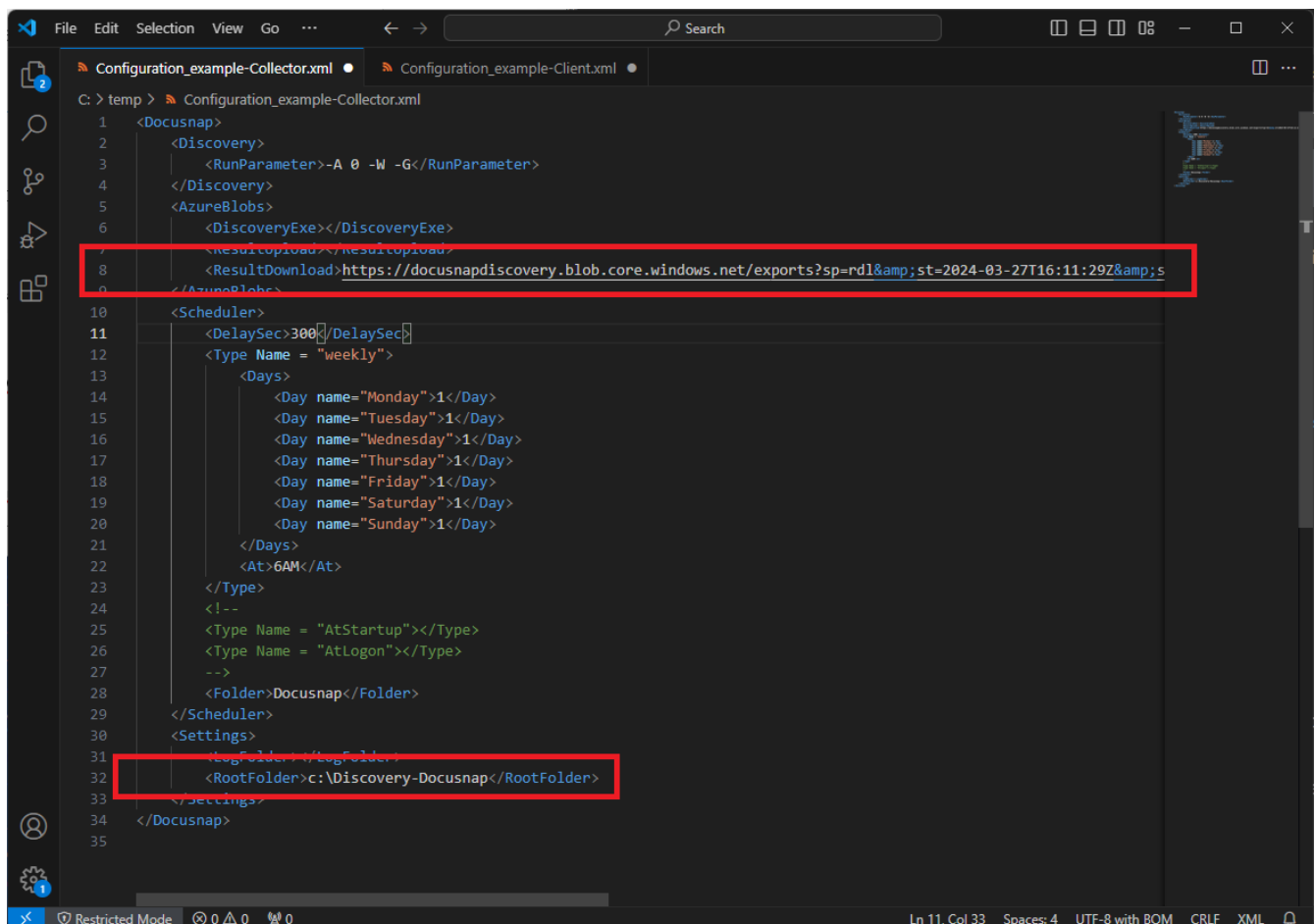


Abbildung 13 - Parameter DownloadConfiguration.xml

Um die geplante Aufgabe zu erstellen, wird die PowerShell in dem Verzeichnis gestartet, indem sich die Discovery-Windows-Azure.ps1 mit der ConfigurationDownload.xml befindet.

**Achten Sie darauf, dass Sie die PowerShell im Administratormodus gestartet ist, und dieser Schritt auf dem Docusnap System durchgeführt wird, das den automatischen Import durchführt.**

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

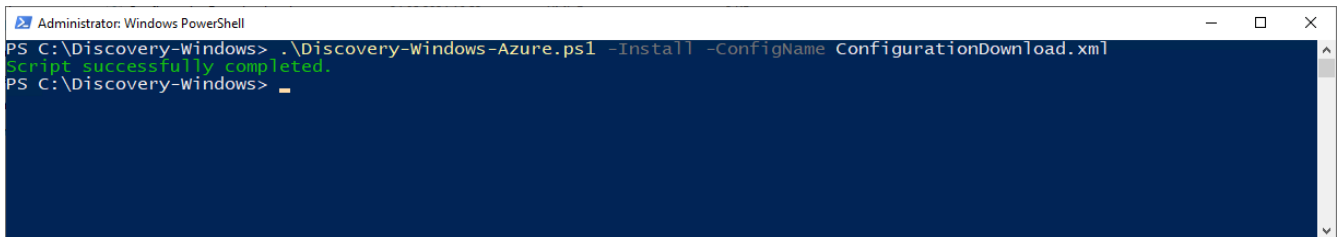


Abbildung 14 – Download automatisieren

Beim Download werden die Ergebnisdateien automatisch aus dem Azure Blob Storage gelöscht. Dadurch wird sichergestellt, dass alte Ergebnisdateien keinen Einfluss auf die Kosten des Azure Blob Storage haben.

## ABBILDUNGSVERZEICHNIS

ABBILDUNG 1 - STORAGE ACCOUNT ERSTELLEN - GRUNDEINSTELLUNGEN .....	5
ABBILDUNG 2 - STORAGE ACCOUNT ERSTELLEN - ERWEITERT .....	6
ABBILDUNG 3 - CONTAINER FÜR DISCOVERY-WINDOWS.EXE .....	7
ABBILDUNG 4 - UPLOAD / AKTUALISIEREN DER DISCOVERY-WINDOWS.EXE .....	8
ABBILDUNG 5 - CONTAINER FÜR FEEDBACK DATEIEN .....	9
ABBILDUNG 6 - ERSTELLEN & BERECHTIGEN DES SHARED ACCESS TOKENS .....	10
ABBILDUNG 7 - DISCOVERY-WINDOWS URL .....	12
ABBILDUNG 8 - PARAMETER CONFIGURATION.XML .....	13
ABBILDUNG 9 - INTUNEWINAPPUTIL.EXE AUFRUF .....	14
ABBILDUNG 10 - INTUNE PROGRAMM OPTIONEN .....	15
ABBILDUNG 11 - INTUNE ERKENNUNGSREGELN .....	16
ABBILDUNG 12 - ABLÖSUNG EINES ALTEN SKRIPTS .....	17
ABBILDUNG 13 - PARAMETER DOWNLOADCONFIGURATION.XML .....	18
ABBILDUNG 14 – DOWNLOAD AUTOMATISIEREN .....	19

## VERSIONSHISTORIE

---

Datum	Beschreibung
24.05.2024	Version 1.0 – HowTo wurde erstellt
27.02.2025	Version 1.1 – HowTo – Downloadlink für Skript integriert
23.01.2026	Version 1.2 – HowTo – Bugfix im Skript und Kapitel 6.3 – Paket aktualisieren hinzugefügt

