



ABAP SDK

Implementation guide for Managed Identities

<https://github.com/Microsoft/ABAP-SDK-for-Azure>

Author: Microsoft SAP Team

Version: 1.0

Contents

What is a Managed Identity?	3
Prerequisites.....	3
Managed identity types.....	3
System-assigned	3
User-assigned	3
How to create Managed Identities.....	3
Create System-assigned Managed Identity(SAMI)	3
Create User-assigned Managed Identity(UAMI)	3
How to enable Managed identity on an existing VM	5
Enable System-assigned Managed Identity	5
Enable User-assigned managed identity	6
How to assign Managed Identity to an Azure resource ?.....	6
Steps to use Managed Identity from SAP using ABAP SDK for Azure.....	9
Configuration.....	9
ZREST_CONFIG	9
ZREST_CONF_MISC	10
ZADF_CONFIG.....	12
ZADF_MI_CONFIG	13
DEMO Program.....	14
ABAP SDK Monitor	15

What is a Managed Identity?

Managed identity is an identity that allows authentication with Azure AD without having to manage any credentials. It eliminates the need of credentials management by providing an identity for the Azure resource in Azure AD to fetch access tokens and thus securing communication between services. Refer to [Managed identities for Azure resources | Microsoft Docs](#) for more insights.

It overcomes the below common challenges such as:

- Management of secrets and credentials.
- Forgetting to renew their credentials resulting in outages.
- Accidentally leaking credentials leading to data leaks.

Prerequisites

- Make sure you have installed ABAP SDK for Azure in your SAP system. Refer document 'ABAP SDK for Azure – GitHub' for more details, Visit <https://github.com/Microsoft/ABAP-SDK-for-Azure>

Managed identity types

System-assigned

It has a 1:1 relation with an Azure resource and shares the same life cycle i.e., when you delete the resource, we automatically clean up the identity.

User-assigned

This managed identity is a standalone Azure resource with its own life cycle.

As the lifecycle is not tied to an Azure resource, so once a resource is deleted, this Identity won't be deleted. You can [enable a user-assigned managed identity](#) and assign it to one or more Azure resources.

How to create Managed Identities

Create System-assigned Managed Identity(SAMI)

A System-assigned Managed Identity can be assigned to a single Azure service. Please refer to [enable SAMI on Virtual machine](#) section to enable this identity on VM. As System-assigned Managed Identity is directly enabled on Virtual machine, you can [add this SAMI identity to Azure service](#) by providing the Virtual machine name under Select Identity.

Create User-assigned Managed Identity(UAMI)

Enabling azure service with a user-assigned identity would require you to create the identity first and then add its resource identifier to your azure service identity management.

1. Sign in to the [Azure portal](#) to create the user-assigned managed identity.
2. In the search box, enter **Managed Identities**. Under **Services**, select **Managed Identities** and then select Create.

Home >

Managed Identities

Default Directory

[+ Create](#) [Manage view](#) [Refresh](#) [Export to CSV](#) [Open query](#) [Assign tags](#) [Delete](#) [Feedback](#)

Filter for any field... [Subscription == all](#) [Resource group == all](#) [Location == all](#) [Add filter](#)

Showing 1 to 1 of 1 records. [No groups](#)

☐ Name ↑↓ ☐ Type ↑↓ ☐ Resource group ↑↓ ☐ Location ↑↓

3. Enter values in the following boxes in the **Create User Assigned Managed Identity** pane:

Home > Managed Identities >

Create User Assigned Managed Identity

Basics Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource group * ⓘ [Create new](#)

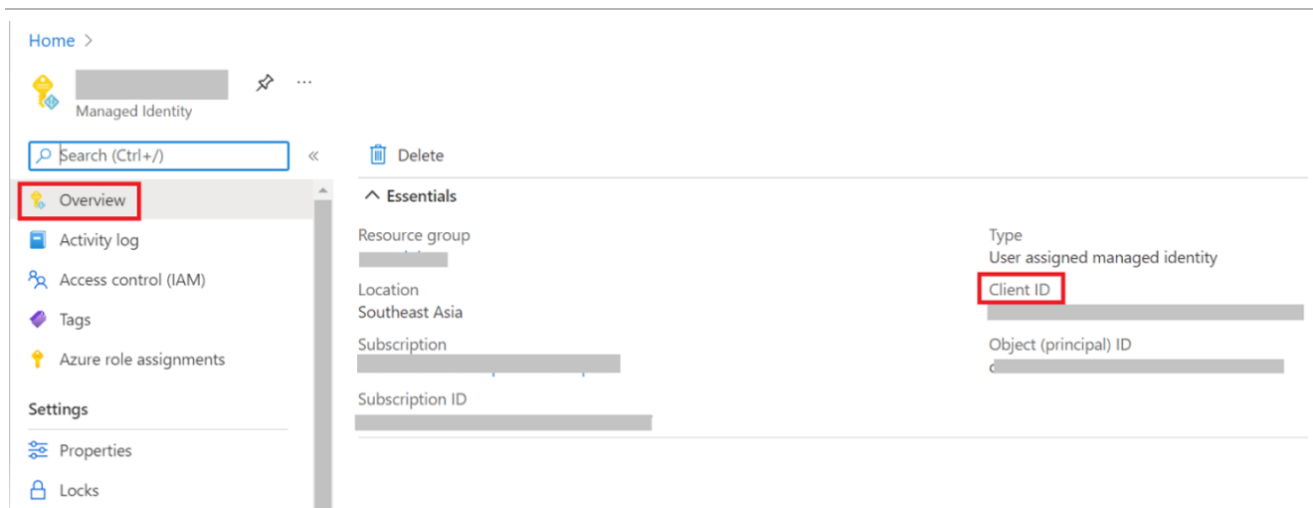
Instance details

Name of your User Assigned Managed Identity. Central US

Name * ⓘ

[Review + create](#) [< Previous](#) [Next : Tags >](#)

- **Subscription:** Choose the subscription to create the user-assigned managed identity under.
 - **Resource group:** Choose a resource group to create the user-assigned managed identity.
 - **Name:** Enter the name for your user-assigned managed identity.
4. Select **Review + create** to create the Managed identity.
 5. Note down the **Client Id of the UAMI** created. This would be required during UAMI entries configuration in SAP tables.

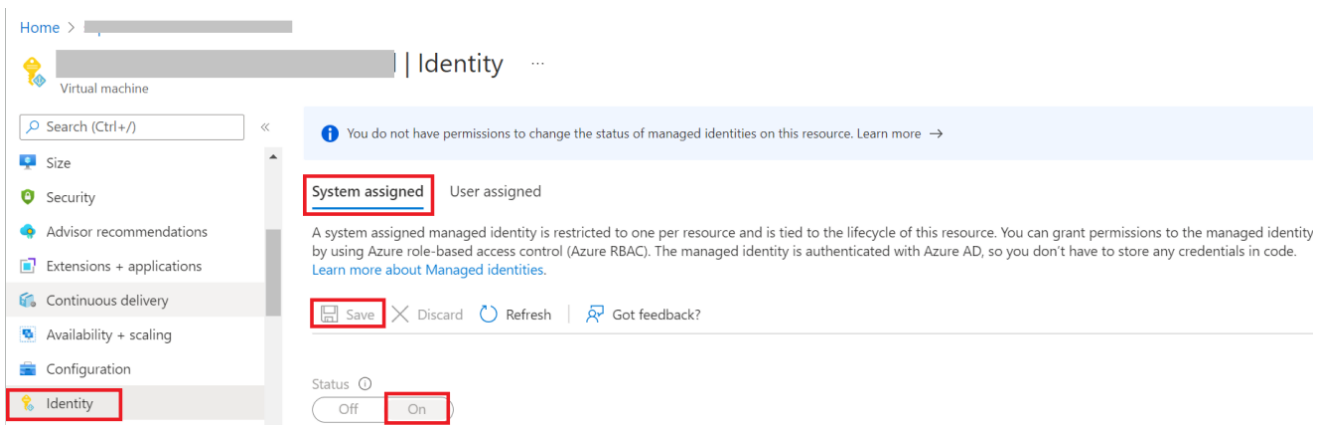


How to enable Managed identity on an existing VM

Enable System-assigned Managed Identity

To enable system-assigned managed identity on a VM that was originally provisioned without it, follow below steps:

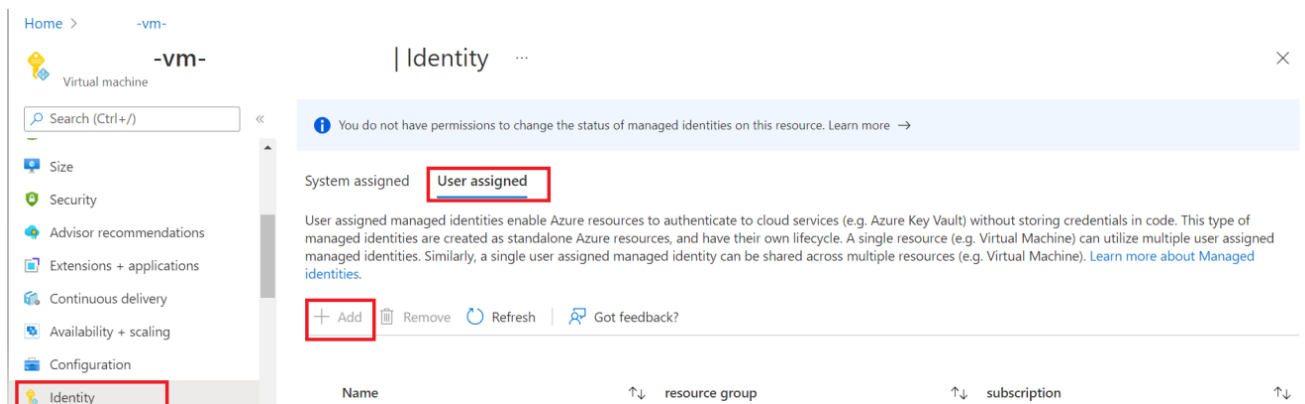
1. Sign in to the [Azure portal](#).
2. Navigate to the desired Virtual Machine and select **Identity**.
3. Under **System assigned**, select **Status** as **On** and then click **Save**:



Enable User-assigned managed identity

To enable user-assigned managed identity on a VM that was originally provisioned without it, follow below steps:

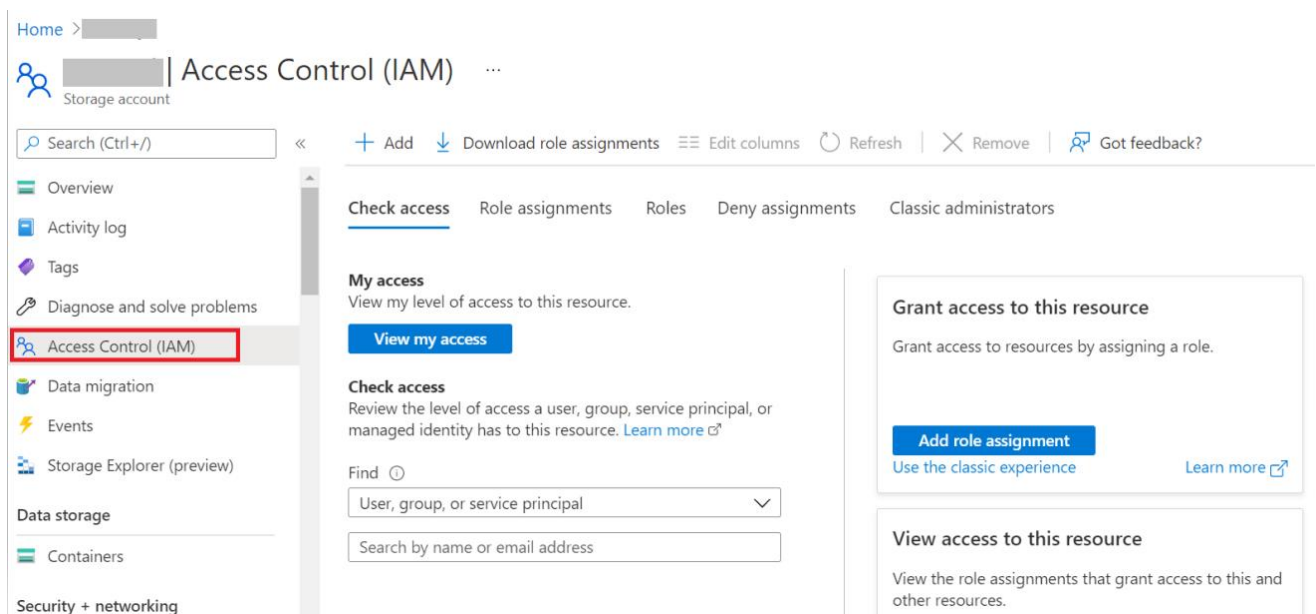
1. Sign in to the [Azure portal](#).
2. Navigate to the desired VM and click **Identity** in the left pane. Select **User assigned** and then **+Add**



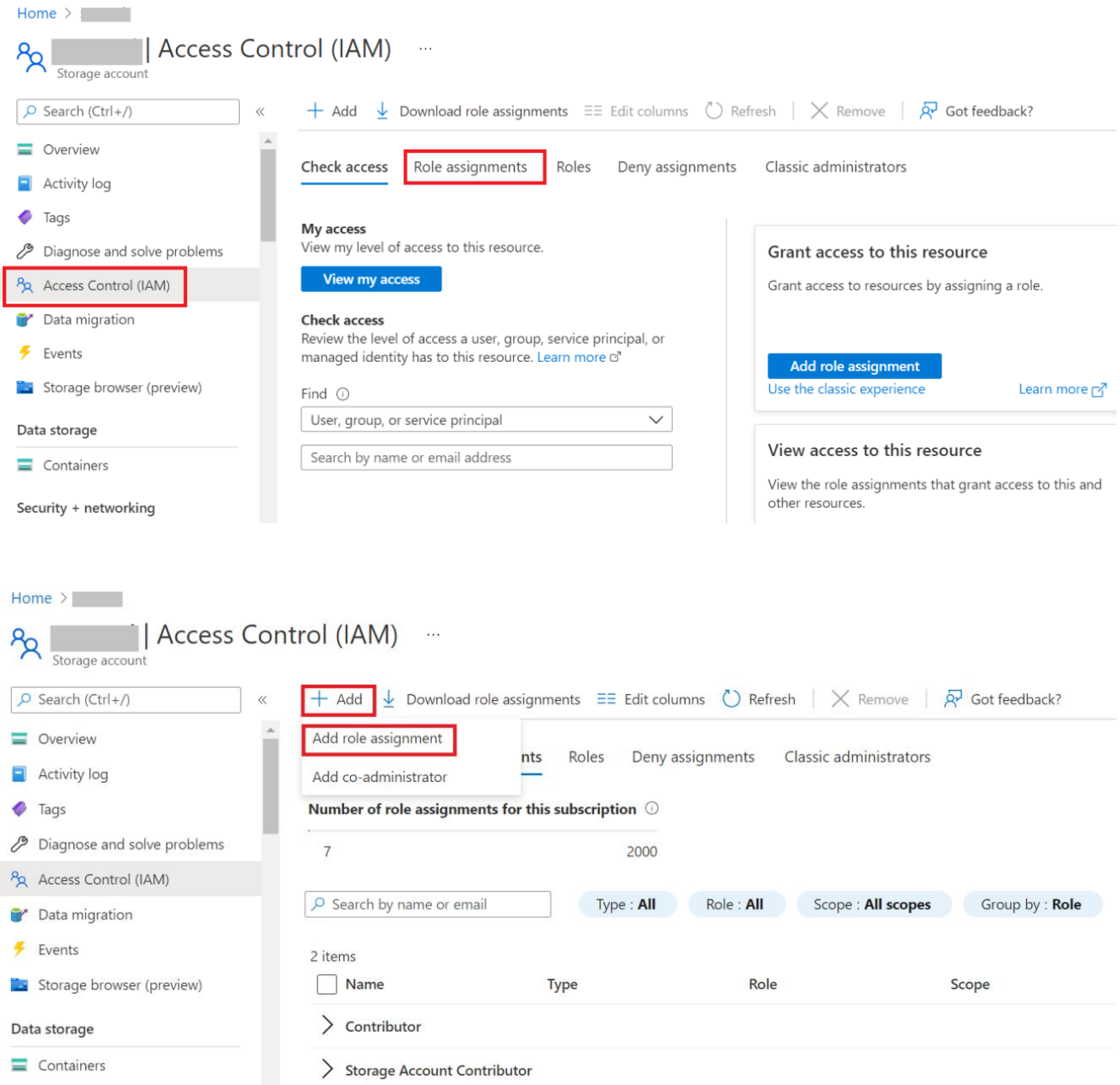
3. Click the user-assigned identity you want to add to the VM and then click **Add**.

How to assign Managed Identity to an Azure resource ?

1. Sign in to the [Azure portal](#) under which you have configured the managed identity.
2. Navigate to the desired resource on which you want to modify access control. For e.g., to give an Azure virtual machine access to a storage account, switch to the Storage container menu.
3. In the left pane, click on the **Access control (IAM)**.



- Go to the **Role Assignments** option and click **Add** to open the Add role assignment page.



The screenshot shows the Azure Access Control (IAM) interface for a storage account. The left sidebar contains navigation options: Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM) (highlighted with a red box), Data migration, Events, Storage browser (preview), Data storage, Containers, and Security + networking. The main content area has tabs: Check access, Role assignments (highlighted with a red box), Roles, Deny assignments, and Classic administrators. Below the tabs, there's a 'My access' section with a 'View my access' button. A 'Check access' section includes a search bar with the text 'User, group, or service principal' and a 'Search by name or email address' input field. On the right, there's a 'Grant access to this resource' section with an 'Add role assignment' button (highlighted with a red box) and a 'View access to this resource' section. Below the main content area, there's a table showing the number of role assignments for this subscription, with 7 assignments out of 2000. A search bar 'Search by name or email' is present, along with filters for Type (All), Role (All), Scope (All scopes), and Group by (Role). The table lists 2 items: Contributor and Storage Account Contributor.

- Goto your Storage account role assignment Page as shown below. Assign the necessary roles .e.g., storage Blob Data contributor for Azure Blob storage service.

Home > >

Add role assignment ...

 Got feedback?

Storage Account Contributor	Lets you manage storage accounts, including accessing storage account keys which pro...	BuiltInRole	Storage	View
Storage Account Key Operator Ser...	Storage Account Key Operators are allowed to list and regenerate keys on Storage Acco...	BuiltInRole	Storage	View
Storage Blob Data Contributor	Allows for read, write and delete access to Azure Storage blob containers and data	BuiltInRole	Storage	View
Storage Blob Data Owner	Allows for full access to Azure Storage blob containers and data, including assigning PO...	BuiltInRole	Storage	View
Storage Blob Data Reader	Allows for read access to Azure Storage blob containers and data	BuiltInRole	Storage	View
Storage Blob Delegator	Allows for generation of a user delegation key which can be used to sign SAS tokens	BuiltInRole	Storage	View
Storage File Data SMB Share Cont...	Allows for read, write, and delete access in Azure Storage file shares over SMB	BuiltInRole	Storage	View
Storage File Data SMB Share Eleva...	Allows for read, write, delete and modify NTFS permission access in Azure Storage file sh...	BuiltInRole	Storage	View
Storage File Data SMB Share Read...	Allows for read access to Azure File Share over SMB	BuiltInRole	Storage	View

Review + assign

Previous

Next

6. Select **Managed identity** and click **Select members** under Members Tab.

Home > >

Add role assignment ...

 Got feedback?

[Role](#) **[Members](#)** [Conditions \(optional\)](#) [Review + assign](#)
Selected role Storage Blob Data Contributor
Assign access to
☐ User, group, or service principal
☒ **Managed identity**
Members [+ Select members](#)

Name

Object ID

Type

No members selected

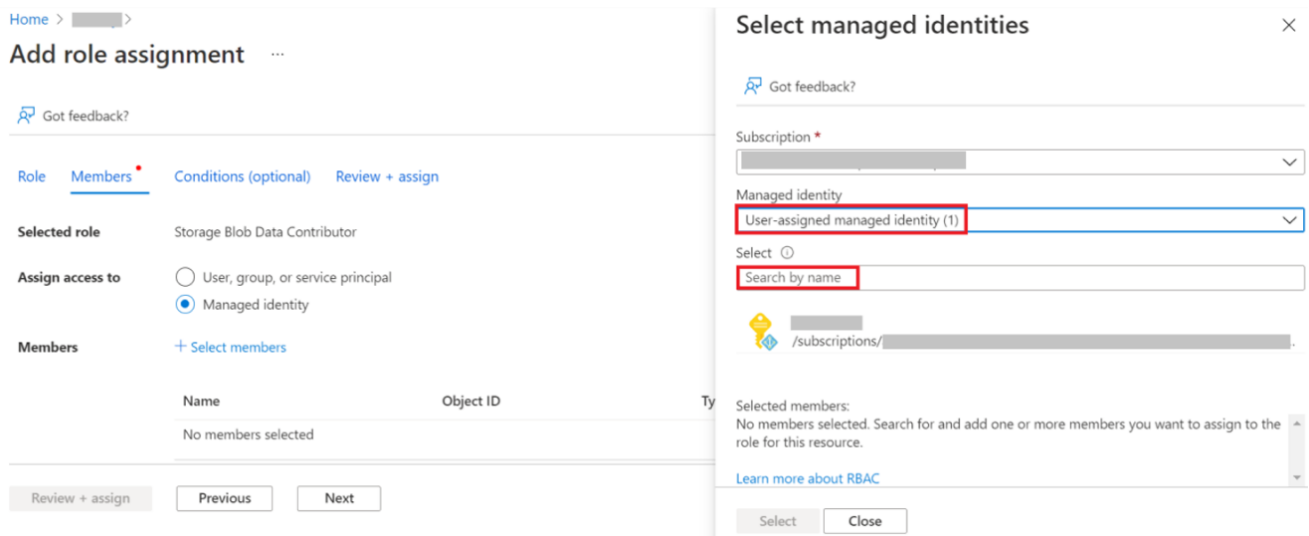
Review + assign

Previous

Next

7. Grant access to the Managed Identity to connect to Azure Blob storage as mentioned below:
 Under select '**Managed identities**' choose subscription where MI belongs to, and Managed Identity as shown below.
 You can select type as [user-assigned managed identity](#) or [system-assigned managed identity](#).

*Note: In case of SAMI , choose system assigned Managed identity as MI type and select SAMI corresponding to the virtual machine.



Steps to use Managed Identity from SAP using ABAP SDK for Azure

Configuration

ABAP SDK has following main configuration tables which needs to be maintained.

- **ZREST_CONFIG** : Master Table for Interface ID Maintenance. You must define a new Interface name and keep the destination as blank.
- **ZREST_CONF_MISC** : This table stores miscellaneous information for interfaces such as information on alerts and re- processing of failed messages automatically.
- **ZADF_CONFIG** : This is an Interface extension table which stores data that is more specific to Azure Services like Interface type, SAS keys, base URI, call type and processing Method.
- **ZADF_MI_CONFIG** : This table stores the information relevant to establishment of connection with Azure service such as Managed Identity Client Id and the resource.

Note : 2 interface IDs need to be maintained in these tables, i.e., 1 interface ID for fetching Managed Identity token and another Interface ID(e.g., Azure Blob Interface Id) for accessing the Azure service using this token.

Maintaining these config tables for MI access token is as shown below:

ZREST_CONFIG

1. MI Token Configuration:

Create a new Interface ID such as 'Z_MI_DEMO' and keep the RFC destination as blank.

Interface : MI Interface ID

Destination – Blank

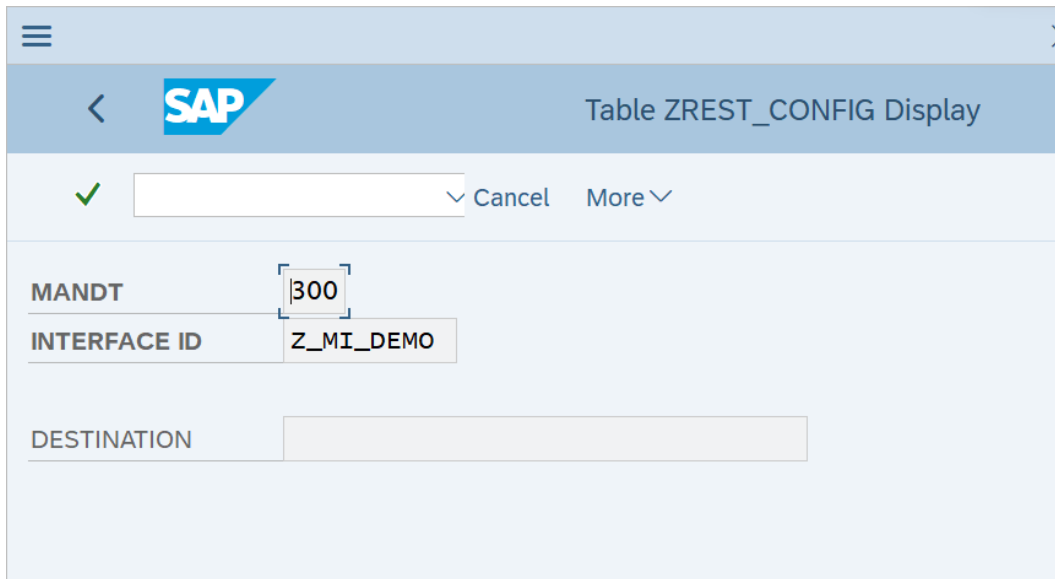


Table ZREST_CONFIG Display

✓ Cancel More ▾

MANDT 300

INTERFACE ID Z_MI_DEMO

DESTINATION

2. Blob Configuration is as follows:

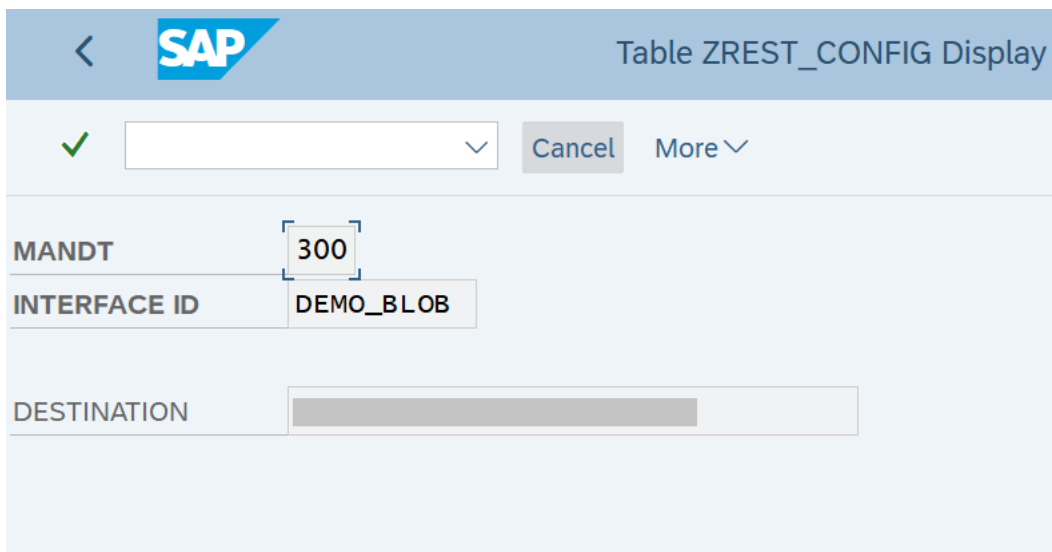


Table ZREST_CONFIG Display

✓ Cancel More ▾

MANDT 300

INTERFACE ID DEMO_BLOB

DESTINATION

ZREST_CONF_MISC

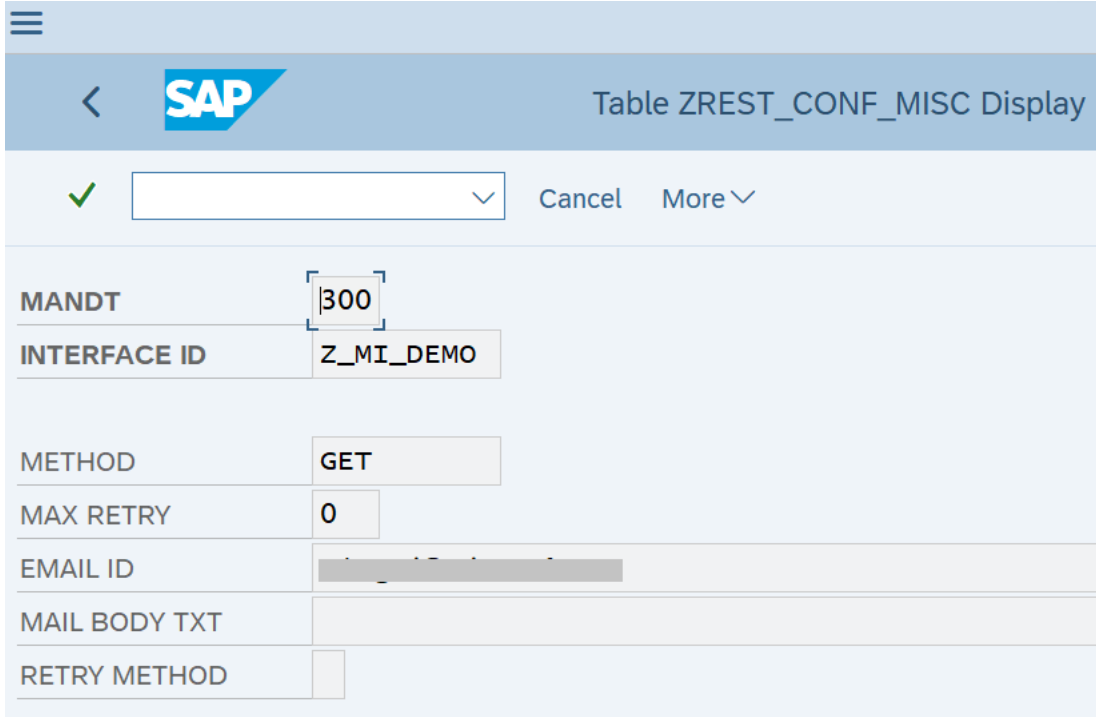
1. MI Token Configuration:

Create an entry in table 'ZREST_CONF_MISC' for the interface Id 'Z_MI_DEMO' created.

Table configuration:

- METHOD – 'GET' to fetch the token
- MAX_RETRY : Number of retry in case of service failure.
- EMAIL_ID : Email id to which alerts would be send.

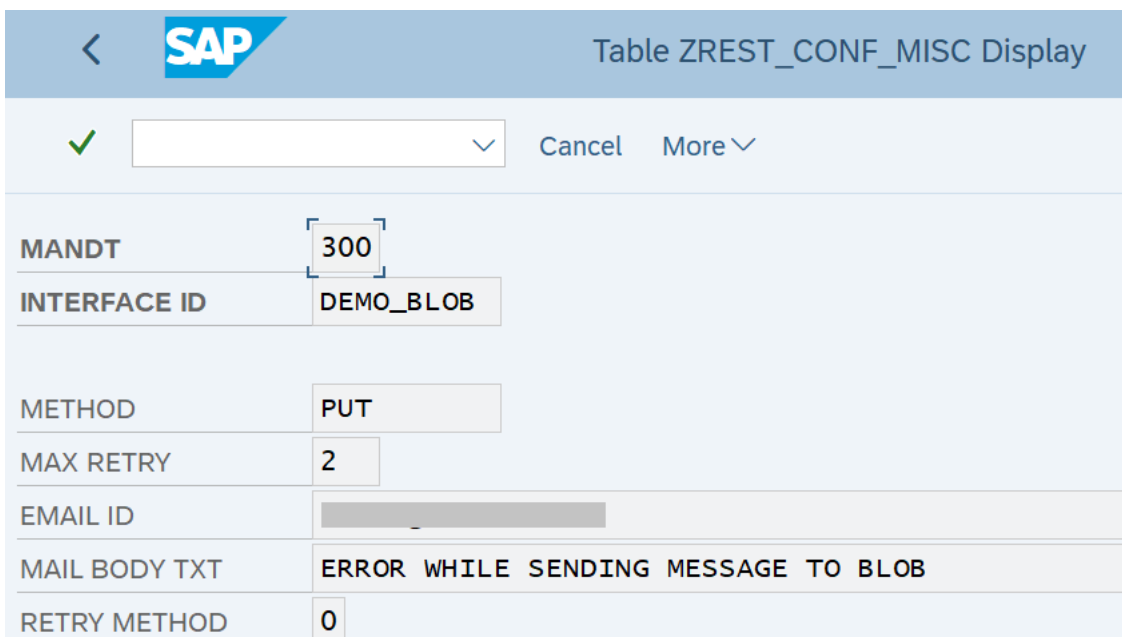
- MAIL_BODY_TXT : Text Id to be maintained for the mail content to be sent.
- RETRY_METHOD : Blank.
- Change Interface ID



The screenshot shows the SAP Table ZREST_CONF_MISC Display interface. At the top, there is a header bar with the SAP logo and the table name. Below the header, there is a search bar with a green checkmark and a dropdown arrow, followed by 'Cancel' and 'More' buttons. The main area contains a form with the following fields:

MANDT	300
INTERFACE ID	Z_MI_DEMO
METHOD	GET
MAX RETRY	0
EMAIL ID	
MAIL BODY TXT	
RETRY METHOD	

2. Blob Configuration is as follows:



The screenshot shows the SAP Table ZREST_CONF_MISC Display interface for Blob Configuration. The fields are as follows:

MANDT	300
INTERFACE ID	DEMO_BLOB
METHOD	PUT
MAX RETRY	2
EMAIL ID	
MAIL BODY TXT	ERROR WHILE SENDING MESSAGE TO BLOB
RETRY METHOD	0

ZADF_CONFIG

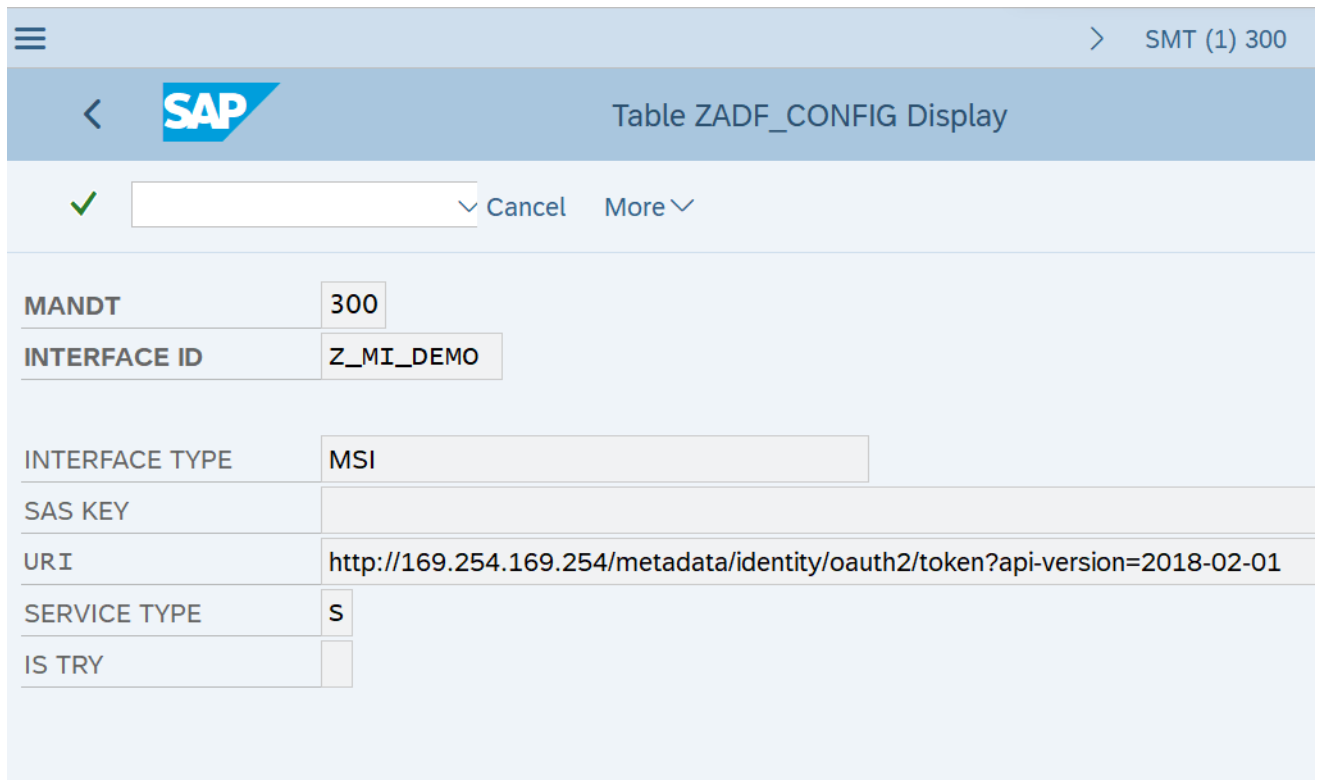
1. MI Token Configuration:

Create an entry in table 'ZADF_CONFIG' for the interface Id 'Z_MI_DEMO' created above.

Table configuration:

- INTERFACE_TYPE - 'Azure Managed Identities'.
- SAS_KEY : Blank.
- URI - 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01' to get the token from the Instance Metadata Service.
- SERVICE_TYPE - Synchronous(S).
- IS_TRY : Keep reprocessing flag blank.

Note: This field can be utilized in our future release to control the reprocessing based on value of X. Presently it should be enabled as blank.



The screenshot shows the SAP SMT (1) 300 Table ZADF_CONFIG Display. The interface includes a header bar with the SAP logo and a title bar. Below the header, there is a search bar with a green checkmark, a dropdown menu, and buttons for 'Cancel' and 'More'. The main area displays the configuration for the table ZADF_CONFIG. The configuration is organized into a table with the following fields and values:

MANDT	300
INTERFACE ID	Z_MI_DEMO
INTERFACE TYPE	MSI
SAS KEY	
URI	http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01
SERVICE TYPE	S
IS TRY	

2. Blob Configuration is as follows:

<
SAP
Table ZADF_CONFIG Display

✓

v

Cancel
More v

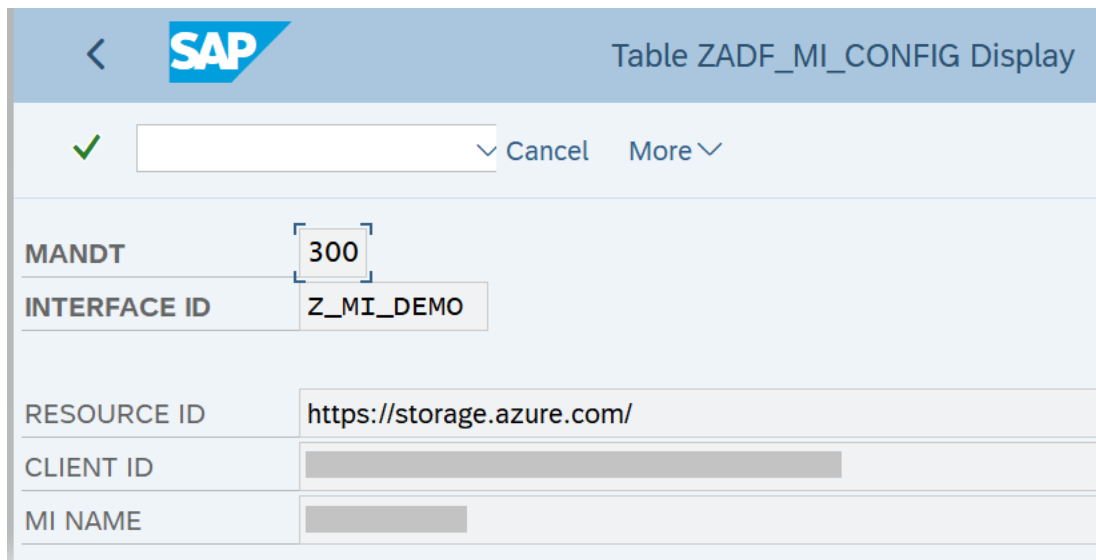
MANDT	300
INTERFACE ID	DEMO_BLOB
INTERFACE TYPE	BLOB
SAS KEY	*****
URI	
SERVICE TYPE	
IS TRY	

ZADF_MI_CONFIG

Create an entry in table 'ZADF_MI_CONFIG' for the above interface Id 'Z_MI_DEMO'.

Table Configuration:

- **Interface** : MI Interface ID
- **Client ID** : Provide client Id of the user-assigned Managed Identity created in the VM. Refer section for the [client id](#).
- **Resource Id** : Maintain resource url for which needs to be accessed e.g., <https://storage.azure.com/> to connect to Azure storage service.
- **MI Name** : Provide service line specific name representing the UAMI configuration.



The screenshot shows the SAP 'Table ZADF_MI_CONFIG Display' interface. At the top, there is a navigation bar with a back arrow, the SAP logo, and the table name. Below this is a status bar with a green checkmark, a search input field, and buttons for 'Cancel' and 'More'. The main area contains a table with the following fields:

MANDT	300
INTERFACE ID	Z_MI_DEMO
RESOURCE ID	https://storage.azure.com/
CLIENT ID	
MI NAME	

DEMO Program

Please refer to DEMO program '**ZADF_DEMO_AZURE_MI**' in SMT system for fetching access token and establishing connection with Azure Blob service. Code level usage overview is as below:

- Call factory method of the 'ZCL_ADF_SERVICE_FACTORY' to create an instance of ZCL_ADF_SERVICE_AAD.
- Now, leverage method '**GET_AAD_TOKEN_MSI**' of this 'ZCL_ADF_SERVICE_AAD' instance to generate MI token for authentication.
- Usage of the method 'ADD_EXPIRY_TIME' of class 'ZCL_ADF_SERVICE_BLOB' is no longer needed for Azure Blob service.
- Pass the header attribute as '**Authorization**' with the above MI token to the call the Azure service.

Report ZADF_DEMO_AZURE_MI Active

```


58 | *-----*
59 |
60 | TRY.
61 |     CALL METHOD zcl_adf_service_factory=>create
62 |     EXPORTING
63 |         iv_interface_id      = 'MI_TEST' " Generate the token
64 |         iv_business_idenfier = 'MI_TEST_TOKEN'
65 |     RECEIVING
66 |         ro_service           = lo_oeref.
67 | CATCH zcx_adf_service .
68 | CATCH zcx_interace_config_missing .
69 | CATCH zcx_http_client_failed .
70 | ENDTRY.
71 |
72 | lo_ref_aad ?= lo_oeref.
73 |
74 | IF lo_ref_aad IS BOUND.
75 |
76 | TRY.
77 |     CALL METHOD lo_ref_aad->get_aad_token_msi
78 |     IMPORTING
79 |         ev_aad_token = lv_aad_token
80 |         ev_response  = lv_response.
81 |
82 | CATCH zcx_adf_service .
83 | CATCH zcx_interace_config_missing.
84 | CATCH zcx_http_client_failed .
85 | ENDTRY.



```

ABAP SDK Monitor













































You can leverage the Interface Monitor (Transaction ZREST_UTIL) provided to monitor history of all the messages posted to Azure Services. You can view the statuses of the messages as well in this monitor in case of scheduled background jobs.





Go to transaction ZREST_UTIL and provide your Interface ID in the selection screen and execute to view all the corresponding messages.

<  Screen

✓ Cancel More ▾   Exit

17 of Records: 17

Exception	Correlation Id	Op	Status	Http	Execution d.	Execution	Retry Time	Host	URI	Interface Id	Calling Program
	000D3AE6703F1EEC96B6F10539201C0A	GET	OK	200	12/10/2021	04:49:02	00:00:00	169.254.169.254	/metadata/identity/oauth2/token_Z_MI_SCM	ZCL_ADF_SERVICE	
	000D3AE6703F1EEC96B6EF6AE99AFC0A	GET	OK	200	12/10/2021	04:48:42	00:00:00	169.254.169.254	/metadata/identity/oauth2/token_Z_MI_SCM	ZCL_ADF_SERVICE	
	000D3AE6703F1EEC96B6C456F429DC0A	GET	OK	200	12/10/2021	04:38:22	00:00:00	169.254.169.254	/metadata/identity/oauth2/token_Z_MI_SCM	ZCL_ADF_SERVICE	
	000D3AE6703F1EEC96B6A4E07ABDDC0A	GET	OK	200	12/10/2021	04:30:31	00:00:00	169.254.169.254	/metadata/identity/oauth2/token_Z_MI_SCM	ZCL_ADF_SERVICE	

In this monitor, you can view the status of the HTTPs message and its headers, response, payload and so on.