**index**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Content** | **Page.no** |
| **1** | **Authenticate Python apps to Azure services by using the Azure SDK for Python Article:**  Recommended app authentication approach   * Token-Based Authentication: * Azure SDK for Python: | **2** |
| **2** | Managed Identity:  Managed identity Types:   * + 1. System-assigned Managed Identity     2. User-assigned Managed Identity | **2** |
| **3** | DefaultAzureCredential:   * + 1. Environment Variables     2. Managed Identity     3. Visual Studio Code (VS Code) Credentials     4. Azure CLI Credentials | **3** |
| **4** | Authenticate Python Apps to Azure SDK:   * + 1. Install the Azure SDK     2. Import Required Libraries and Create Client Objects     3. Use DefaultAzureCredential for Authentication | **4-5** |

**Authenticate Python apps to Azure services by using the Azure SDK for Python Article**

Recommended app authentication approach

* + **Token-Based Authentication:**

With token-based authentication, your app requests an access token from Azure Active Directory (Azure AD) when it needs to access an Azure resource. This token acts as a temporary "key" to access the resource.

* + **Azure SDK for Python:**

Azure provides software development kits (SDKs) for different programming languages, including Python. The Azure SDK for Python includes classes that make it easy for your Python apps to work with token-based authentication.

**Managed Identity**

When you host an app in Azure using services like Azure App Service, Azure Virtual Machines, or Azure Container Instances, the recommended approach to authenticate an app to Azure resources is with [managed identity](https://learn.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview).

**Managed Identity types:**

**System-assigned Managed Identity**: Automatically tied to an Azure resource. Enables easy authentication for that resource. Deleted when the resource is deleted.

**User-assigned Managed Identity**: Created separately as a standalone resource. Can be shared among multiple Azure resources for consistent identity and permissions.

**DefaultAzureCredential:**

* DefaultAzureCredential is a class provided by the Azure SDKs that simplifies the process of authenticating your application or code with Azure resources.
* DefaultAzureCredential tries multiple authentication methods in a predefined order until it successfully obtains the necessary credentials to authenticate with Azure resources.

The DefaultAzureCredential class automatically tries the following authentication methods:

* 1. **Environment Variables:**

It checks for environment variables like “AZURE\_CLIENT\_ID, AZURE\_TENANT\_ID, AZURE\_CLIENT\_SECRET, and AZURE\_USERNAME” to authenticate with Azure.

* 1. **Managed Identity:**

If the code is running on an Azure resource with a system-assigned managed identity or a user-assigned managed identity, it will use the managed identity to authenticate.

* 1. **Visual Studio Code (VS Code) Credentials:**

If you are running your application in a local development environment with Visual Studio Code and the Azure Account extension, it can use the credentials stored in VS Code.

* 1. **Azure CLI Credentials:**

f the Azure CLI is installed on your machine and you have signed in with az login, the credentials from the CLI are used.

**Authenticate Python Apps to Azure SDK:**

1. **Install the Azure SDK**

you need to install the Azure SDK for Python, which provides the necessary libraries and tools to work with Azure services.

**Pip Install <package-Manager>**

1. **Import Required Libraries and Create Client Objects**
   1. you need to import the required libraries from the Azure SDK for Python. Depending on the specific Azure service you want to access, you'll import the relevant modules.
   2. After importing the necessary libraries, you'll create client objects. These client objects represent connections to Azure services and allow you to interact with them programmatically.
2. **Use DefaultAzureCredential for Authentication**

The DefaultAzureCredential class provided by the Azure SDK for Python is a versatile authentication solution that streamlines the process of authenticating Python applications to Azure services. Rather than relying on explicit credentials like connection strings, this class automatically handles authentication using multiple methods, adapting to the application's runtime environment seamlessly.

EXAMPLE:

from azure.identity import DefaultAzureCredential

from azure.mgmt.resource import ResourceManagementClient

# Replace with your Azure subscription ID

subscription\_id = 'your-subscription-id'

# Authenticate using your Azure credentials

credential = DefaultAzureCredential()

# Create the resource management client

resource\_client = ResourceManagementClient(credential, subscription\_id)

# Fetch a list of all resource groups

resource\_groups = resource\_client.resource\_groups.list()

# Iterate through the resource groups

for resource\_group in resource\_groups:

    # Get the name of the resource group

    resource\_group\_name = resource\_group.name

    print(f'Tags for resource group {resource\_group\_name}:')

    # Fetch the tags for the resource group

    tags = resource\_group.tags

    # Print the tags

    print(f'Tags for resource group {resource\_group\_name}:')

    for key, value in tags.items():

        print(f'Tag: {key}, Value: {value}')

    print()