

Azure SDK Create ResourceGroup (Application Registration)

1. Register the application in Microsoft Entra ID

We create the Service principal with the following command:

```
az ad sp create-for-rbac --name "myserviceprincipal" ^  
--role contributor ^  
--scopes /subscriptions/XXXXXXXXXXXXXXXXXXXXXXXXXXXX ^  
--sdk-auth
```

We navigate to Azure Microsoft Entra ID

The screenshot displays the Microsoft Azure portal. On the left is a navigation pane with various service links. In the center-right is the main content area showing the 'particular | Overview' page for Microsoft Entra ID. The 'App registrations' link in the right-hand menu is highlighted with a red box.

Navigation Pane (Left):

- Create a resource
- Home
- Dashboard
- All services
- FAVORITES**
- Function App
- SQL databases
- Azure Cosmos DB
- Virtual machines
- Load balancers
- Storage accounts
- Virtual networks
- Microsoft Entra ID**
- Monitor
- Advisor
- Microsoft Defender for Cloud
- Cost Management + Billing
- Help + support

Main Content Area (Right):

Home > **particular | Overview**
Microsoft Entra ID

Overview

- Preview features
- Diagnose and solve problems

Manage

- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Delegated admin partners
- Enterprise applications
- Devices
- App registrations**
- Identity Governance
- Application proxy
- Custom security attributes

First we get the Tenant ID

particular | Overview
Microsoft Entra ID

Overview | Monitoring | Properties | Recommendations | Tutorials

Search your tenant

Basic information

Name	particular	Users	1
Tenant ID	e099cebd-5eea-41a3-88db-bcb9a9c8a83e	Groups	0
Primary domain	luiscoenriquezhotmail.onmicrosoft.com	Applications	8
License	Microsoft Entra ID Free	Devices	0

Alerts

- Microsoft Entra Connect v1 Retirement**
All version 1.x builds of Microsoft Entra Connect (formerly AAD Connect) will soon stop working between October 2023 – March 2024. You must move to Cloud Sync or Microsoft Entra Connect v2.x.
- Azure AD is now Microsoft Entra ID**
Microsoft Entra ID is the new name for Azure Active Directory. No action is required from you.

We enter Microsoft Entra ID and we click on App registrations

particular | App registrations
Microsoft Entra ID

New registration | Endpoints | Troubleshooting | Refresh | Download | Preview features | Got feedback?

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

All applications | Owned applications | Deleted applications | Applications from personal account

Start typing a display name or application (client) ID to filter these r... [Add filters](#)

Display name	Application (client) ID	Created on	Certificates & secrets
FunctionApp1	352b0eb1-0fa6-413f-9223-0f992cd85...	11/19/2023	Current
mynewazureapplicationforweb	9477d6f7-a3e0-470d-a095-b397d37ff...	12/5/2023	-
myseviceprincipal	6002f92f-afd8-46d3-ae99-21212298b...	12/18/2023	Current
myseviceprincipaluis	86ef6f64-4b32-4191-8727-ca81cc8ed...	12/18/2023	Current
sample420231124142236	df4fae6c-7380-43f8-94d4-8a8904b1e...	11/24/2023	Current
sampleactivedirectoryauth	270dfd4f-6e05-4532-89fe-049c15f79c4f	11/28/2023	Current
seviceprincipaluis	101e45df-6282-4900-a27b-1ad1c87ca...	12/18/2023	Current
terraform	60c58c1a-116a-4869-912c-75c09309b...	4/19/2022	Expired

We click on the "myseviceprincipal", the service principal we created at the beginning of this section 2

We copy the **clientId** and (optionally) you can also get from here the **tenantId**, we paste both values in the application variables **clientId** and **tenantId**

Microsoft Azure

Search resources, services, and docs (G+/)

Home > particular | App registrations > myseviceprincipal

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

Essentials

Display name: myseviceprincipal

Application (client) ID: 6002f92f-afd8-46d3-ae99-21212298b618

Object ID: 3a306fe4-0843-4ea5-a932-ec53ce5e1ec3

Directory (tenant) ID: e099cebd-5eea-41a3-88db-bcb9a9cba83e

Supported account types: All Microsoft account users

Client credentials: 0 certificate, 1 secret

Redirect URIs: Add a Redirect URI

Application ID URI: Add an Application ID URI

Managed application in local directory: myseviceprincipal

We create a new secret clicking on Client credentials link

Microsoft Azure

Search resources, services, and docs (G+/)

Home > particular | App registrations > myseviceprincipal

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

Essentials

Display name: myseviceprincipal

Application (client) ID: 6002f92f-afd8-46d3-ae99-21212298b618

Object ID: 3a306fe4-0843-4ea5-a932-ec53ce5e1ec3

Directory (tenant) ID: e099cebd-5eea-41a3-88db-bcb9a9cba83e

Supported account types: All Microsoft account users

Client credentials: 0 certificate, 1 secret

Redirect URIs: Add a Redirect URI

Application ID URI: Add an Application ID URI

Managed application in local directory: myseviceprincipal

We press in the New client secret button

Microsoft Azure

Search resources, services, and docs (G+/)

Home > particular | App registrations > myseviceprincipal

myseviceprincipal | Certificates & secrets

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Got feedback?

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) Client secrets (1) Federated credentials (0)

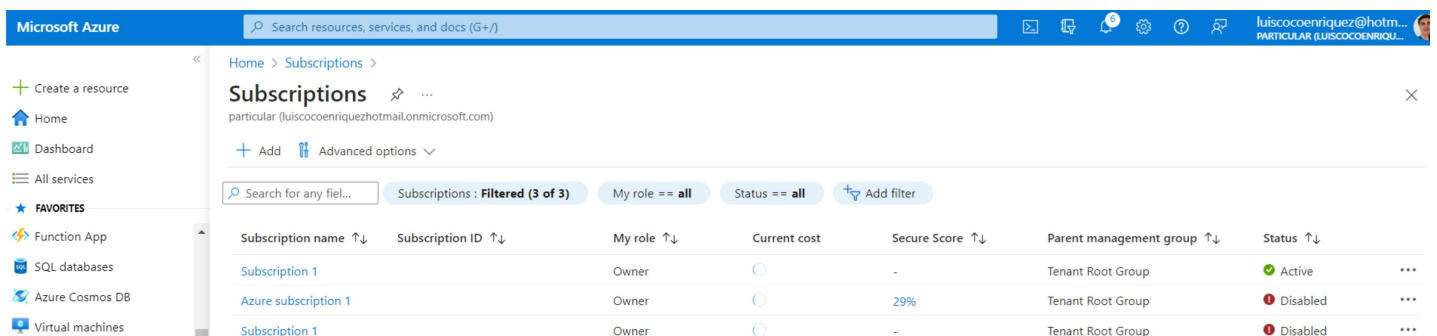
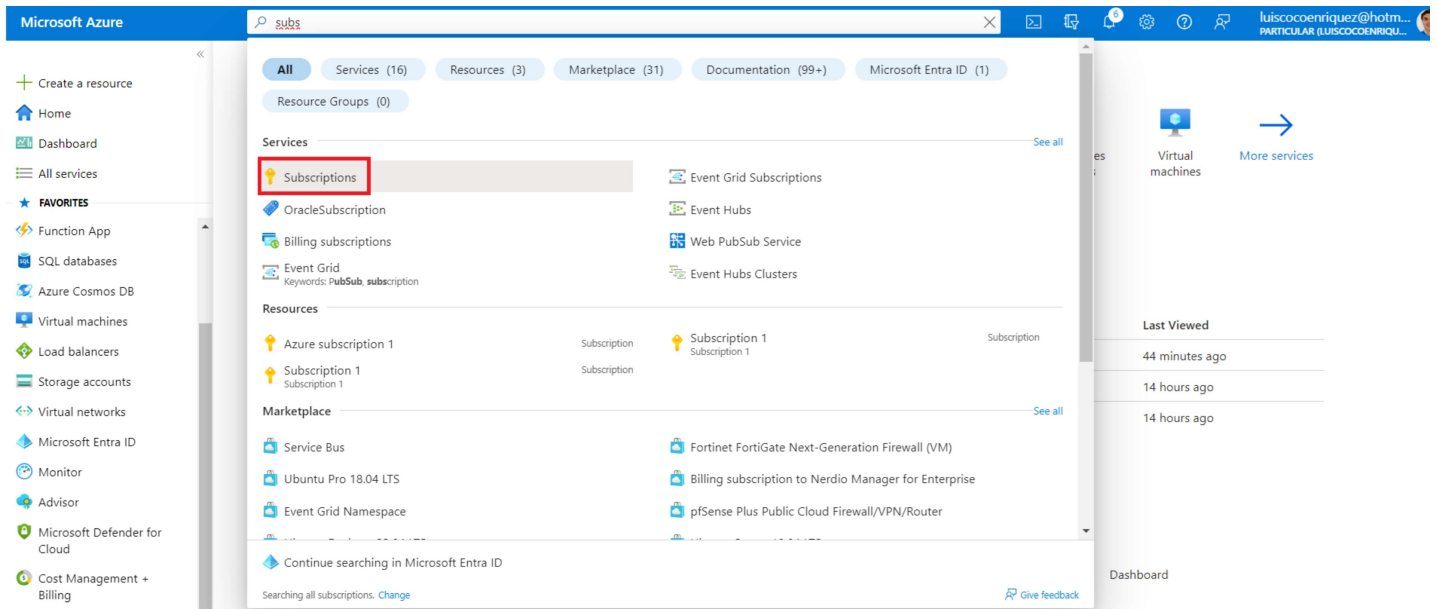
A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	New client secret	Expires	Value	Secret ID
rbac		12/18/2024	Z-f*****	fbf00cb0-4b9e-4381-9120-81fd1b716c4a

We copy the Secret value to paste in the application clientSecret variable

We navigate in Azure Portal to the Subscriptions service and we copy the subscription subscriptionId



2. Input the application source code

Open VSCode and create a new C# application with this command:

```
dotnet new console --framework net8.0
```

Add the dependencies with the commands:

```
dotnet add package Microsoft.Azure.Management.ResourceManager --version 3.17.4-preview
```

```
dotnet add package Microsoft.Rest.Azure.Authentication
```

Input the C# source code for creating a new Azure ResourceGroup with Azure SDK for .NET

```
using Microsoft.Azure.Management.ResourceManager;
using Microsoft.Azure.Management.ResourceManager.Models;
using Microsoft.Rest.Azure.Authentication;

namespace AzureResourceGroupCreation
{

```

```
class Program
{
    static void Main(string[] args)
    {
        // Replace these variables with your own values
        string clientId = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        string clientSecret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        string tenantId = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        string subscriptionId = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        string resourceGroupName = "myresourcegroupluiscocoenriquez";
        string location = "westeurope"; // e.g., "westus"

        // Authenticate with Azure
        var serviceClientCredentials = ApplicationTokenProvider.LoginSilentAsync(tenantId,

        // Initialize the resource management client
        var resourceManagementClient = new ResourceManagementClient(serviceClientCredentia
        {
            SubscriptionId = subscriptionId
        });

        // Create the resource group
        var resourceGroup = new ResourceGroup { Location = location };
        resourceGroup = resourceManagementClient.ResourceGroups.CreateOrUpdate(resourceGro

        System.Console.WriteLine("Resource group created: " + resourceGroup.Name);
    }
}
```