**Lab 13: Azure Principals and App Registration**

**Introduction:**

A service principal is a security identity that is used by applications, hosted services, and automated tools to access specific Azure resources. By using service principals, you can manage permissions and access control for non-human users with greater precision. This lab will guide you through creating a service principal via the Azure CLI and registering an application through the Azure portal.

**Objectives:**

* Create a service principal using the Azure CLI.
* Retrieve and understand the service principal credentials.
* Register an application using the Azure portal.
* Understand how to configure application registration settings.

**Steps:**

**Part 1: Creating a Service Principal Using Azure CLI**

**Step 1:** Open Azure Cloud Shell

* **Explanation**: Cloud Shell provides a command-line environment to manage Azure resources. Using Cloud Shell pre-authenticates your session.
* **How to Do It**: Click on the Cloud Shell icon at the top of the Azure portal or visit [shell.azure.com](https://shell.azure.com/) to access it directly.

**Step 2:** Authenticate with Your Subscription

* **Explanation**: This step ensures that the Azure CLI commands execute under your subscription.
* **How to Do It**: Type **az login** if prompted and follow the instructions to log in.

**Step 3:** Create the Service Principal

* **Explanation**: The service principal is created to enable applications to access Azure resources securely.
* **How to Do It**:
  + Replace **<subscription id>** with your actual subscription ID.
  + Run the command:

css

Copy code

az ad sp create-for-rbac --sdk-auth --name azuremldatascientist --role Contributor --scopes /subscriptions/<subscription id>

* + This command creates a service principal named "azuremldatascientist" with the "Contributor" role.

**Step 4:** Save the Output

* **Explanation**: The output contains sensitive information such as **clientId**, **clientSecret**, and **tenantId**, which will be used to authenticate your application.
* **How to Do It**: Copy and securely store the generated output.

**Step 5:** Retrieve Service Principal Information

* **Explanation**: You can get more information about the service principal.
* **How to Do It**: Run the command:

css

Copy code

az ad sp show --id <clientId from previous result>

* + Replace **<clientId from previous result>** with the actual clientId from the previous step.

**Part 2: Registering an Application in Azure Portal**

**Step 1:** Access Microsoft Entra ID (formerly Azure Active Directory)

* **Explanation**: App registrations are managed through Microsoft Entra ID.
* **How to Do It**: In the Azure portal, search for "Microsoft Entra ID" or "Azure Active Directory" in the search bar and click on it.

**Step 2:** Access App Registrations

* **Explanation**: This section manages applications that can integrate with Azure services.
* **How to Do It**: Click on "App registrations" in the left menu.

**Step 3:** Start a New Registration

* **Explanation**: This step initiates the application registration process.
* **How to Do It**: Click on the "New registration" button at the top.

**Step 4:** Fill in the Application Details

* **Explanation**: Provide the application with a name and define what accounts can access it.
* **How to Do It**:
  + Enter a name for the application.
  + Choose the supported account types. The default option works for most scenarios.
  + Click "Register" to create the application.

**Step 5:** Review and Configure Application Registration

* **Explanation**: After registration, you can configure options such as client secrets and roles.
* **How to Do It**: The application registration page will open upon clicking "Register."
  + Here, you can explore options to configure roles, permissions, secrets, certificates, and other settings.

**Summary:**

In this lab, you successfully created a service principal via the Azure CLI and registered an application through the Azure portal. Understanding service principals and app registrations helps in securely managing access for applications and services to Azure resources.