Here’s a **step-by-step walkthrough** to deploy the **SimpleApp.zip** (or any .NET Core ZIP package) to **Azure App Service** **using only the Azure Portal**—no CLI required.

https://mywebapp91234.azurewebsites.net /api/products

**✅ 1️⃣ Prepare Your Files**

1. Make sure you have the **SimpleApp.zip** file ready (from earlier).
2. This ZIP should contain the **project files** (Program.cs, Controllers, etc.), **not** a pre-built bin/Release output.
   * App Service will build it if needed.

**✅ 2️⃣ Create an Azure SQL Database (Portal)**

1. Sign in to **Azure Portal**.
2. **Create a resource → Databases → SQL Database**.
3. **Basics** tab:
   * Database name: mydb
   * Server: **Create new** → give a unique name (e.g., myserver123).
   * Admin login: myadmin
   * Password: (strong password)
   * Compute + storage: choose **Basic** or **Free** for testing.
4. **Review + Create → Create**.
5. After deployment, open the new **SQL server** → **Networking** → enable  
   ✅ **Allow Azure services and resources to access this server**.

**✅ 3️⃣ Create the App Service**

1. In the portal, **Create a resource → Web App**.
2. **Basics** tab:
   * Subscription & Resource Group: select or create.
   * Name: mywebapp123 (must be unique).
   * Publish: **Code**.
   * Runtime stack: **.NET 8 (LTS)**.
   * Operating System: **Linux** or **Windows** (either works; .NET 8 works well on Linux).
   * Region: same as your SQL database for best performance.
3. **Pricing plan**: select **Free (F1)** or **Basic (B1)**.
4. Click **Review + Create → Create**.

**✅ 4️⃣ Configure the Connection String**

1. After the Web App is created, go to **App Service → Settings → Configuration**.
2. Click **Connection strings → New connection string**.
   * **Name:** DefaultConnection
   * **Value:**
   * Server=tcp:<your-server-name>.database.windows.net,1433;
   * Initial Catalog=mydb;
   * User ID=myadmin;
   * Password=<YourPassword>;
   * Encrypt=True;
   * Connection Timeout=30;
   * **Type:** SQLAzure
3. Click **OK**, then **Save** (top toolbar).
4. Accept the prompt to **restart the app**.

**✅ 5️⃣ Deploy the ZIP Package**

There are two easy portal methods:

**Option A: Deployment Center (Recommended)**

1. Go to your **Web App → Deployment → Deployment Center**.
2. In **Source**, choose **Zip deploy** or **External Git / Local Git** → select **Zip Deploy**.
3. Click **Browse**, select your **SimpleApp.zip**, and click **Save/Deploy**.
4. Azure uploads and deploys the app automatically.

**Option B: Advanced Tools (Kudu)**

1. Go to **Web App → Development Tools → Advanced Tools (Kudu)** → **Go**.
2. This opens a new tab (https://<appname>.scm.azurewebsites.net).
3. In Kudu:
   * Click **Debug Console → CMD**.
   * Navigate to site/wwwroot.
   * Drag and drop your **SimpleApp.zip** into this folder.
   * Right-click the ZIP → **Extract**.

**✅ 6️⃣ Verify Deployment**

1. In the portal, open your **Web App** → **Overview**.
2. Click the **Browse** button (top toolbar) or go to:
3. https://mywebapp123.azurewebsites.net/api/products
4. You should see an empty JSON array [] (or data if you’ve seeded the DB).

**⚡ Optional: Apply EF Migrations**

If your app uses Entity Framework migrations:

* You can run them **locally** pointing to the Azure SQL connection string, or
* Use a tool like **Azure Cloud Shell** with dotnet ef database update.

**✅ Quick Recap**

| **Step** | **Portal Location** | **Action** |
| --- | --- | --- |
| 1 | **SQL Database** | Create DB & allow Azure access |
| 2 | **Web App** | Create App Service |
| 3 | **Configuration** | Add SQL connection string |
| 4 | **Deployment Center** | Upload ZIP & deploy |
| 5 | **Overview** | Browse the live site |

That’s it! Your **SimpleApp** will now run on **Azure App Service** and connect securely to your **Azure SQL Database** using the connection string you configured.

**✅ 4️⃣ Create the Table**

1. In the **Query editor**, paste this SQL script:
2. CREATE TABLE Products (
3. Id INT IDENTITY(1,1) PRIMARY KEY,
4. Name NVARCHAR(100) NOT NULL,
5. Price DECIMAL(18,2) NOT NULL
6. );
   * Id → auto-increment primary key
   * Name → product name (up to 100 characters)
   * Price → decimal value with 2 decimal places
7. Click **Run** (the green ▶️ button).

You should see a message:

Commands completed successfully.

**✅ 5️⃣ (Optional) Insert Sample Data**

To test the API and connection:

INSERT INTO Products (Name, Price)

VALUES

('Laptop', 1200.00),

('Phone', 699.99),

('Headphones', 49.99);

Click **Run** again.

**✅ 6️⃣ Verify the Data**

Check your table contents:

SELECT \* FROM Products;

Click **Run** → You should see your rows in the Results pane.

**✅ 7️⃣ Test From Your App**

* Once your **App Service** is deployed and configured with the correct **connection string**,  
  navigate to:
* https://<yourwebappname>.azurewebsites.net/api/products
* You should see the rows you inserted (in JSON).