#### 1. **Create an Azure App Service in the Azure Portal**

1. **Sign in to Azure Portal**:
   * Navigate to [Azure Portal](https://portal.azure.com/) and log in with your Microsoft Azure account.
2. **Create a New Resource**:
   * In the left sidebar, click on **Create a resource**.
   * Search for **App Service** and select **App Service**.
3. **Configure App Service**:
   * **Subscription**: Select your Azure subscription.
   * **Resource Group**: Either choose an existing resource group or create a new one. (A resource group is a container for resources).
   * **Name**: Enter a name for your App Service (e.g., AlphabetCheckerApi).
   * **Publish**: Choose **Code** (for deploying the application code).
   * **Runtime Stack**: Select .NET 6 or the version you are using (e.g., **.NET 7**).
   * **Region**: Select the region closest to your users for better performance.
   * **App Service Plan**: Choose a pricing tier based on your needs (e.g., **F1 Free Tier** or **B1 Basic Tier** for smaller apps).
4. **Review and Create**:
   * Review your configuration and click **Create** to provision your Azure App Service.
   * Wait for the deployment to complete.

#### 2. **Publish the Application Locally (Prepare for Deployment)**

Now, you'll prepare your application for deployment by publishing it as a **release** build.

1. **Publish the Application**:
   * Open your project in Visual Studio.
   * Right-click on the project in **Solution Explorer** and select **Publish**.
2. **Choose Publish Target**:
   * Select **Azure** > **Azure App Service (Windows or Linux)** depending on your choice of environment.
   * Click **Next**.
3. **Select Existing App Service or Create New**:
   * **Select an Existing App Service**: Choose the one you created earlier in Azure.
   * Alternatively, you can click **Create New** if you want to create a new App Service instance directly from Visual Studio (this option is similar to what you did through the Azure Portal).
   * Click **Finish** once you’ve selected the App Service.
4. **Publish the Application**:
   * After selecting the target, Visual Studio will automatically deploy your app to Azure App Service.
   * Click **Publish** and wait for the process to finish. Visual Studio will build and deploy the app to Azure.

#### 3. **Configure the Application (Optional)**

Once the app is deployed to Azure, you might need to configure certain settings such as environment variables, database connections, or logging.

1. **Open App Service Configuration**:
   * In the **Azure Portal**, go to your **App Service** (the one you created earlier).
   * In the left-hand menu, scroll down to **Configuration** under the **Settings** section.
2. **Set Application Settings**:
   * You can add key-value pairs for settings like ASPNETCORE\_ENVIRONMENT, API keys, or database connection strings.
   * For example, if you have any environment-specific configurations (like a development or production environment), add them here.
   * Click **Save** after making changes.

#### 4. **Verify the Deployment**

After the deployment process is complete, you can verify that your application is working as expected.

1. **Find the URL of the App**:
   * In the **Azure Portal**, go to your App Service.
   * Under **Overview**, you will see a URL for your app (e.g., https://yourappname.azurewebsites.net).
2. **Access the API**:
   * Open the URL in a web browser or use an API testing tool like **Postman** or **cURL** to test your API endpoints.

Example:

curl https://yourappname.azurewebsites.net/api/alphabetchecker/check

This should return the appropriate response, e.g., a JSON result indicating whether the string contains all the alphabets.

### 5. **Set Up Continuous Deployment (Optional)**

If you want to set up continuous deployment (CD) from **GitHub** or **Azure DevOps**, you can do so easily in Azure App Service.

#### **Using GitHub for Continuous Deployment**:

1. **Set up GitHub Integration**:
   * In the **Azure Portal**, go to your App Service.
   * Under **Deployment** in the left-hand menu, click **Deployment Center**.
   * Choose **GitHub** as the source, and then follow the steps to authenticate with your GitHub account and select the repository and branch to deploy from.
2. **Configure Deployment**:
   * Choose **GitHub Actions** or **Azure Pipelines** for continuous integration.
   * Azure will automatically create a pipeline for you (in GitHub Actions, for example) and deploy whenever you push changes to the selected GitHub repository.

#### **Using Azure DevOps for Continuous Deployment**:

1. **Set up Azure DevOps Integration**:
   * Go to the **Deployment Center** in Azure App Service.
   * Choose **Azure DevOps** as the source and follow the prompts to link your Azure DevOps repository.
2. **Create Pipeline**:
   * If you don’t already have a pipeline, Azure will create one for you. This pipeline will deploy your app automatically each time changes are pushed to the repository.