


How to deploy Azure Functions

4 minutes

In this unit, we'll start by opening a project file called "**Devkit Translator**" in Visual Studio Code. We'll provision Azure services to create an Azure function then deploy the Azure function in the cloud by using Azure IoT Workbench extension.

The components in this unit are:

- **Azure IoT workbench:** an extension supported by Visual Studio Code. It simplifies the end-to-end process of the development of the IoT Devkit. It will also install some dependency extensions like **Azure Functions** and **Azure IoT Toolkit**.
- **Azure provisioning service:** allows you to create an automated process to connect to the cloud.
- **Devkit Translator:** a translator project sample that is available on GitHub: <https://github.com/Azure-Samples/mxchip-iot-devkit-translator> . The project contains two main parts - Functions and Device folders.
- **Functions folder:** has two code files – DevKitTranslatorFunction.cs, SpeechTranslation.cs.
- **DevKitTranslatorFunction.cs:** a function code file that lives in the Functions folder in the project sample. The function name in the code file is **devkit_translator**. The function calls the task from **SpeechTranslation.cs** that recognizes the speech, uploads the audio file to the Azure AI Speech service, and then translates the speech from the source language to the target language.

Steps we follow are:

1. Start Visual Studio Code.
2. Open Devkit Translator sample
3. Provision Azure Services (IoT Hub, IoT Hub Device and Azure Function)
4. Update the Azure function with the device name, Speech Service key, and location.
5. Deploy Azure function to the cloud

By the end of this unit, your Azure function app will be deployed to the cloud. The function app will publish a URL. This URL includes a key that is required, by default, to access your function over HTTP. You'll get the function URL from Azure.

Next unit: How to deploy device code to the IoT device

[Continue >](#)