**Practical no-2 Create a function app in the Azure portal**

This module requires a sandbox to complete. You have used 3 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Activate sandbox

Top of Form

Bottom of Form

You are now ready to start implementing the temperature service. In the previous unit, you determined that a serverless solution would best fit your needs. Let's start by creating a function app to hold our Azure Function.

## What is a function app?

Functions are hosted in an execution context called a **function app**. You define function apps to logically group and structure your functions and a compute resource in Azure. In our escalator example, you would create a function app to host the escalator drive gear temperature service. There are a few decisions that need to be made to create the function app; you need to choose a service plan and select a compatible storage account.

### Choose a service plan

Function apps may use one of two types of service plans:

* Consumption plan
* Azure App Service plan

When using the Azure serverless application platform, choose the **Consumption plan**. This plan provides automatic scaling and bills you only when your functions are running. The Consumption plan comes with a configurable timeout period for executing a function. By default, it's five (5) minutes, but may be configured to have a timeout as long as 10 minutes.

The **Azure App Service plan** enables you to avoid timeout periods by having your function run continuously on a VM that you define. When using an App Service plan, you are responsible for managing the app resources the function runs on, so this is technically not a serverless plan. However, it may be a better choice if your functions are used continuously, or if your functions require more processing power or longer execution time than the Consumption plan can provide.

### Storage account requirements

When you create a function app, it must be linked to a storage account. You can select an existing account or create a new one. The function app uses this storage account for internal operations, such as logging function executions and managing execution triggers. On the **Consumption plan**, this is also where the function code and configuration file are stored.

## Create a function app

Let's create a function app in the Azure portal.

1. Sign in to the [Azure portal](https://portal.azure.com/learn.docs.microsoft.com) using the same account you used to activate the sandbox.
2. Under **Azure services**, select **Create a resource**.

The **Create a resource** pane appears.

1. In the menu, select **Compute**, and then select **Function App** in the Popular products list. The **Create Function App** pane appears.
2. On the **Basics** tab, enter the following values for each setting.

| **Setting** | **Value** |
| --- | --- |
| **Project Details** |  |
| Subscription | Concierge Subscription |
| Resource Group | From the dropdown list, select **[sandbox resource group name]** |
| **Instance Details** |  |
| Function App name | Enter a globally unique app name, which becomes part of the base URL of your service. For  example, you can name it **escalator-functions-xxx**, where you can replace xxx with your  initials and a number. Valid characters are a-z, 0-9 and - |
| Publish | Code |
| Runtime stack | Node.js (which is the language we use to implement the function examples in this exercise). |
| Version | Accept default |
| Region | Select a geographical location close to you. In a production system, you would want to select  a location near your customers or consumers of the function. |

1. Select **Review + create**, and then select **Create**. Deployment will take a few minutes. You'll receive a notification when deployment is completed.

## Verify your Azure function app

1. When deployment completes, select **Go to resource**. The Function App pane for your escalator function appears.
2. In the **Essentials** section, select the **URL** link to open it in a browser. A default Azure web page appears with a message that your Functions app is up and running.

