



Compte Rendu de Travaux Pratiques

Compte Rendu - Travaux Pratiques En Cloud & Virtualisation

Filière : Réseaux Informatiques & Télécommunications

Niveau : 4^{ème} Année

Sujet :

TP5 : Web app, Function app and Logic app

Réalisé par :

Zied KHARRAT
Nidhal JABNOUNI
Yassine BELARBI

Année Universitaire : 2024-25

TASK 01

1. We created an Azure Web App with the PHP runtime in the App Services section. This is used to host and run PHP-based web applications in a managed Azure environment.

The screenshot displays the Azure portal interface for an Azure Web App. At the top, there is a navigation bar with a 'Home' link and a search bar. Below the navigation bar, the web app's name 'mywebapp23' is prominently displayed, along with a 'Web App' label and a close button. A toolbar provides actions such as 'Browse', 'Stop', 'Swap', 'Restart', and 'Delete'. A left-hand sidebar lists various management options, with 'Overview' currently selected. The main content area, titled 'Essentials', provides key information about the web app, including its resource group, status, location, subscription, and default domain. At the bottom, a tabbed interface allows switching between 'Properties', 'Monitoring', 'Logs', 'Capabilities', and 'Notifications'.

Home >

mywebapp23 Web App

Search

Browse Stop Swap Restart Delete

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Events (preview)
- Recommended services (preview)
- Log stream
- Resource visualizer
- > Deployment

Essentials [JSON View](#)

Resource group (... : [rgwebapp](#))

Status : Running

Location ([move](#)) : Canada Central

Subscription (... : [Azure for Students](#))

Subscription ID : 59c49685-d27f-4288-85fc-5ee4a6614e34

Default domain : [mywebapp23-hhh2gdhedeapg0g5.canadace...](#)

App Service Plan : [ASP-rgwebapp-bb0a \(B1: 1\)](#)

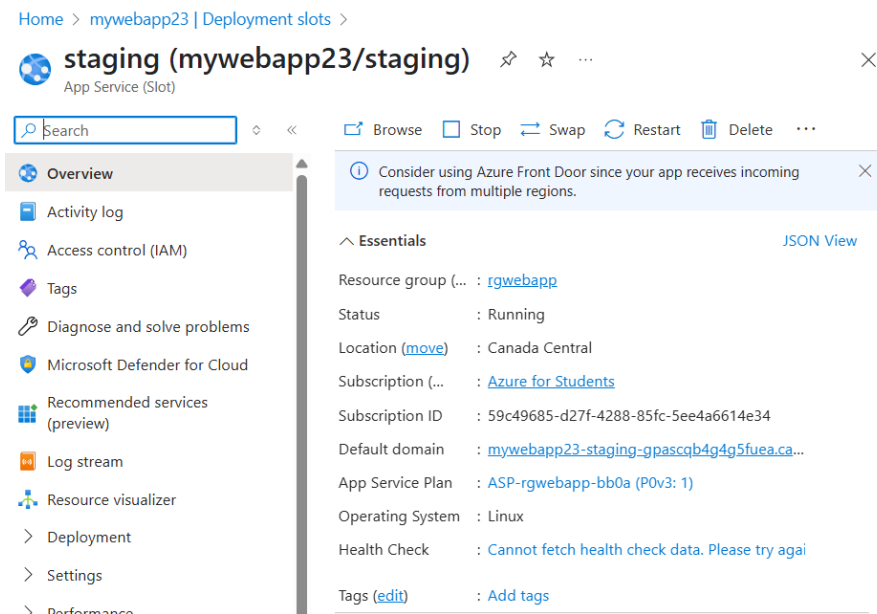
Operating System : Linux

Health Check : [Not Configured](#)

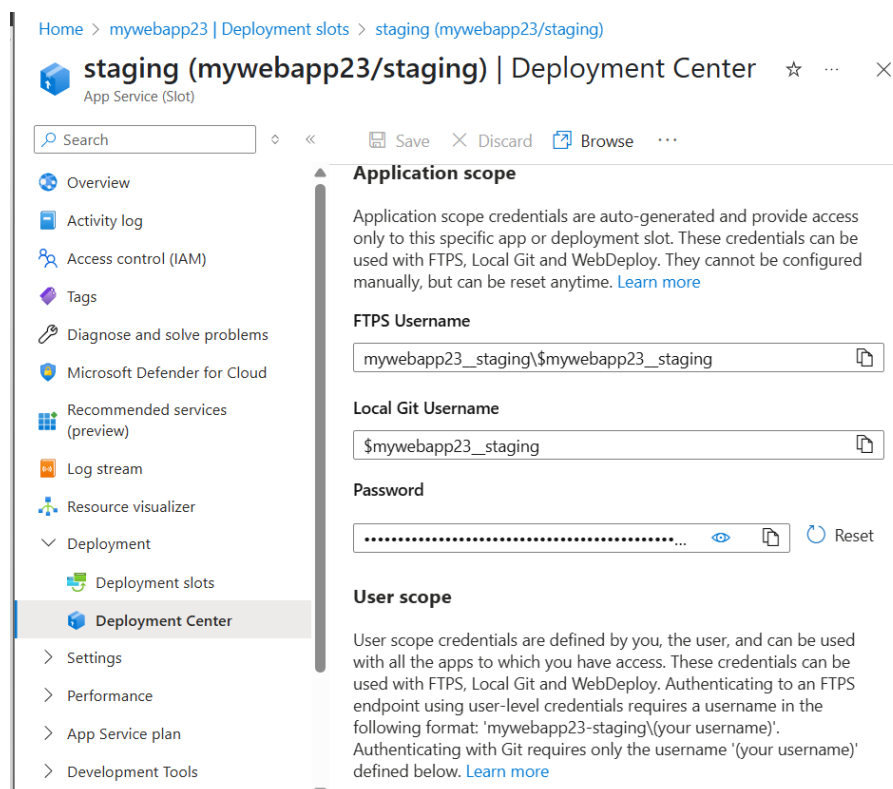
Tags ([edit](#)) : [Add tags](#)

Properties Monitoring Logs Capabilities Notifications

2. We created a staging deployment slot without cloning settings. This is used to deploy and test new versions of the app in isolation before moving them to production.



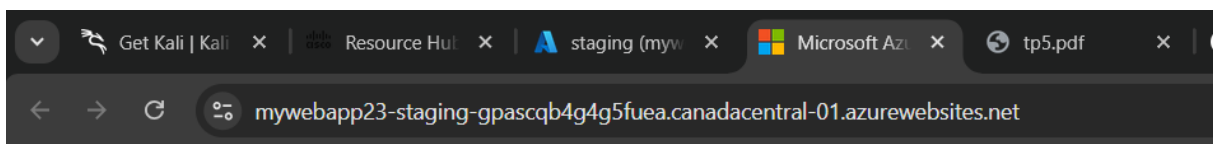
3. We configured deployment settings on the staging slot by enabling Local Git as the source and setting credentials. This allows us to push code directly from a local Git repository to the staging environment.



4. We cloned a sample PHP app, added a Git remote pointing to the staging slot, and pushed the code using Git. This deploys the Hello World web app to the staging environment, verifying deployment configuration.

```
Switch to Bash Restart Manage files New session Editor Web preview Settings Help
-t, --[no-]track <branch>
    branch(es) to track
-m, --[no-]master <branch>
    master branch
--[no-]mirror[=(push|fetch)]
    set up remote as a mirror to push to or fetch from


PS /home/jabnouninidhal/php-docs-hello-world> git remote add mywebapp23_staging https://mywebapp23-staging-gpascqb4g4g5fuea.scm.canadacentral-01.azurewebsites.net:443/mywebapp23.git
PS /home/jabnouninidhal/php-docs-hello-world> git push mywebapp23_staging master
Username for 'https://mywebapp23-staging-gpascqb4g4g5fuea.scm.canadacentral-01.azurewebsites.net:443': mywebapp23_staging
Password for 'https://mywebapp23_staging@mywebapp23-staging-gpascqb4g4g5fuea.scm.canadacentral-01.azurewebsites.net:443':
fatal: Authentication failed for 'https://mywebapp23-staging-gpascqb4g4g5fuea.scm.canadacentral-01.azurewebsites.net:443/mywebapp23.git/'
```



Your web app is running and waiting for your content

Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.



 Built with PHP

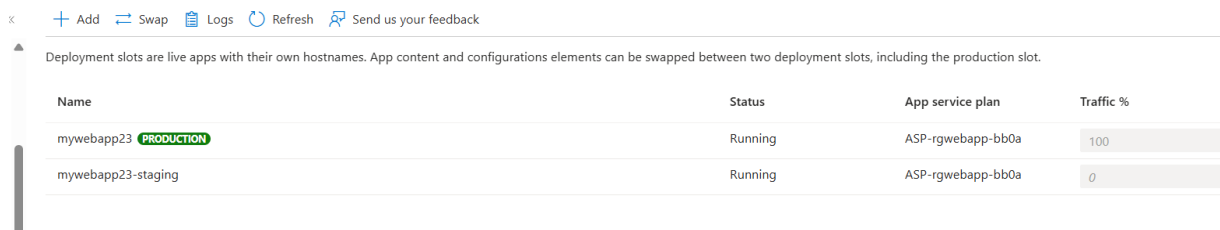
Haven't deployed yet?
Use the deployment center to
publish code or set up continuous
deployment.

[Deployment center](#)

Starting a new web site?
Follow our Quickstart guide to get
a web app ready quickly.

[Quickstart](#)

5. We swapped the staging and production slots. This allows us to promote the tested version of the app to production with zero downtime and easy rollback if needed.

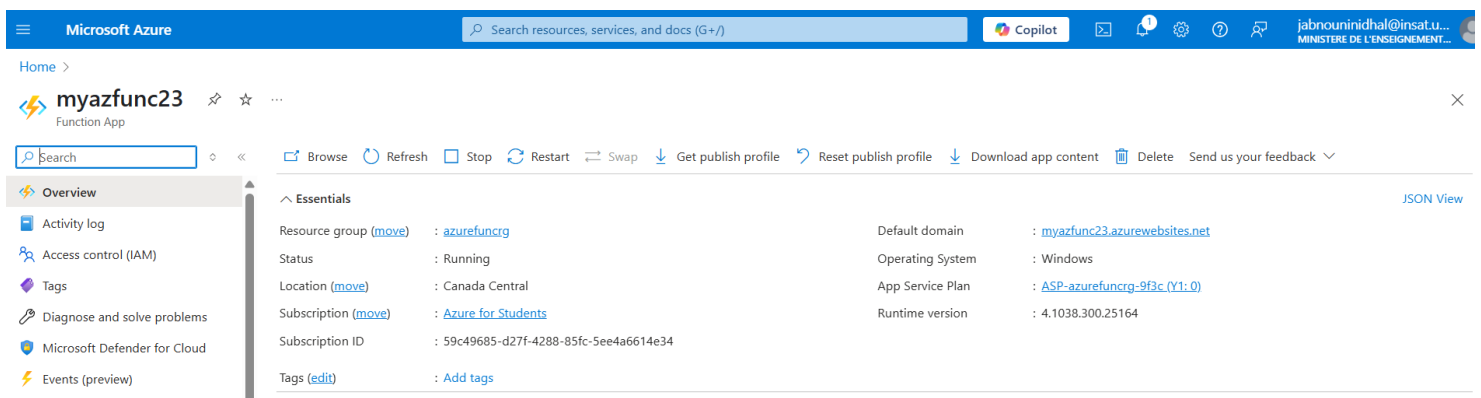


Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot.			
Name	Status	App service plan	Traffic %
mywebapp23 PRODUCTION	Running	ASP-rgwebapp-bb0a	100
mywebapp23-staging	Running	ASP-rgwebapp-bb0a	0

7. We cleaned up the resources by deleting the resource group. This is used to avoid unnecessary charges and maintain a tidy Azure environment.

TASK 02

1. We created an Azure Function App with the .NET runtime on a consumption plan. This is used to run small pieces of code on demand without managing infrastructure.



Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home >

myazfunc23
Function App

Search

Overview

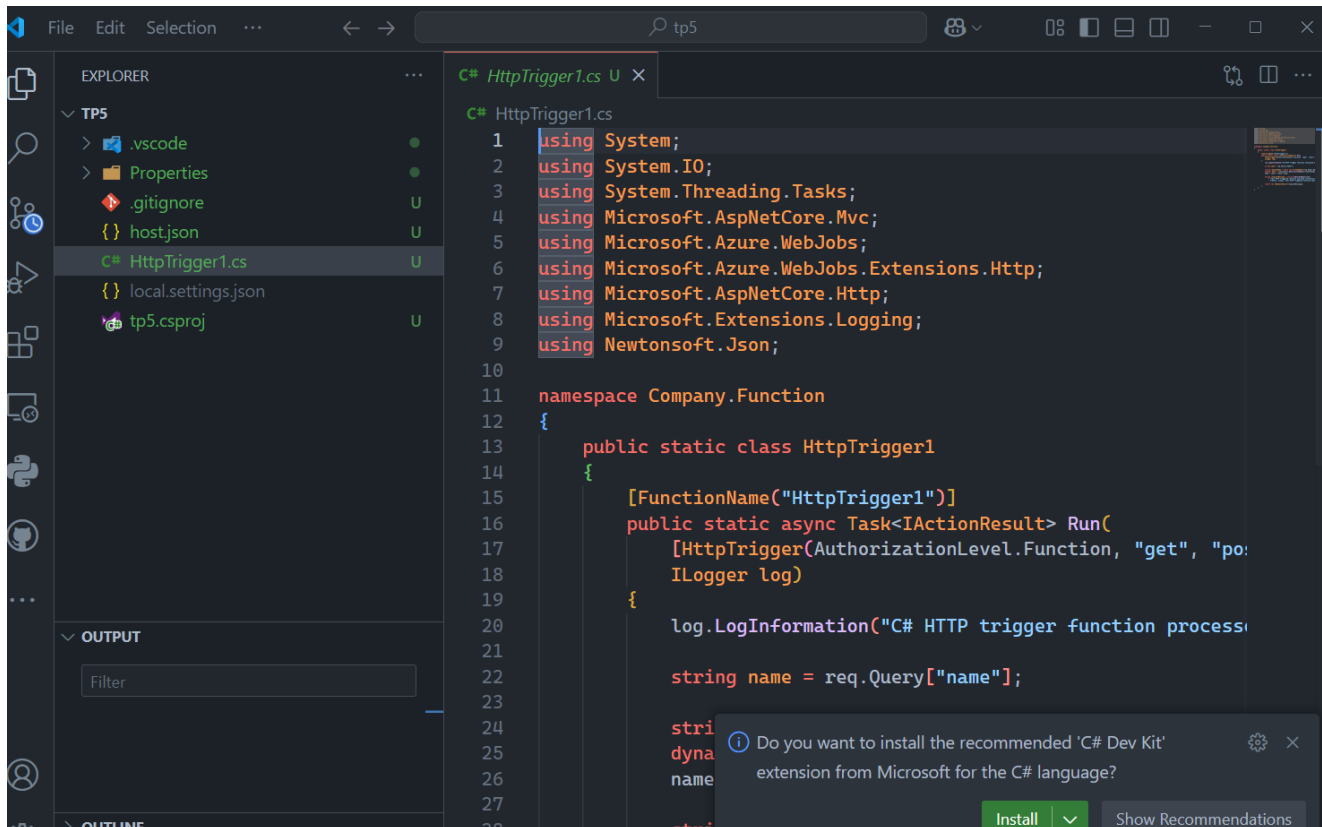
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Microsoft Defender for Cloud
- Events (preview)

Essentials

Resource group (move)	: azurefuncrg	Default domain	: myazfunc23.azurewebsites.net
Status	: Running	Operating System	: Windows
Location (move)	: Canada Central	App Service Plan	: ASP-azurefuncrg-9f3c (Y1: 0)
Subscription (move)	: Azure for Students	Runtime version	: 4.1038.300.25164
Subscription ID	: 59c49685-d27f-4288-85fc-5ee4a6614e34		
Tags (edit)	: Add tags		

JSON View

2. We created a new function with an HTTP trigger and configured its integration. This allows external services to trigger the function via HTTP requests. We did this by leveraging VSCode's integration with Azure, then deployed the function to our previously created function app.



```
1 using System;
2 using System.IO;
3 using System.Threading.Tasks;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.Azure.WebJobs;
6 using Microsoft.Azure.WebJobs.Extensions.Http;
7 using Microsoft.AspNetCore.Http;
8 using Microsoft.Extensions.Logging;
9 using Newtonsoft.Json;
10
11 namespace Company.Function
12 {
13     public static class HttpTrigger1
14     {
15         [FunctionName("HttpTrigger1")]
16         public static async Task<IActionResult> Run(
17             [HttpTrigger(AuthorizationLevel.Function, "get", "post")]
18             ILogger log)
19         {
20             log.LogInformation("C# HTTP trigger function processed a request.");
21
22             string name = req.Query["name"];
23
24             string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
25             dynamic data = JsonConvert.DeserializeObject(requestBody);
26             name = name ?? data?.name;
27
28             return name != null ? Ok($"Get request for {name}") : NotFound();
29         }
30     }
31 }
```

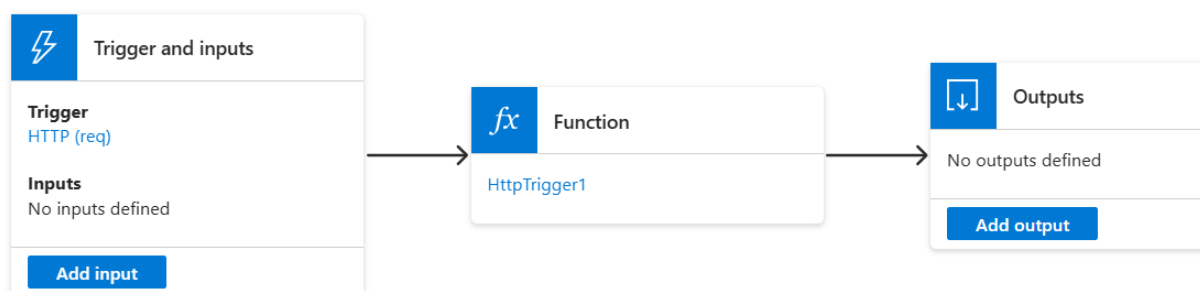
Home > myfunc23 > HttpTrigger1

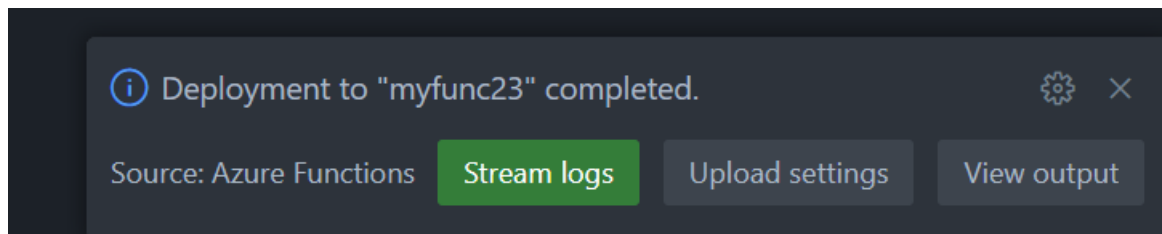
HttpTrigger1 | Integration

myfunc23

Code + Test **Integration** Function Keys Invocations Logs Metrics

Refresh Send us your feedback





The screenshot shows the Azure portal interface for a Function App named "myfunc23". The left sidebar contains a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Recommended services (preview), Log stream, Resource visualizer, Functions, App keys, App files, Proxies, and Deployment. The main content area is titled "Overview" and includes a search bar and a list of actions: Browse, Refresh, Stop, Restart, Swap, Get publish profile, Reset publish profile, Download app content, Delete, and Send us your feedback. Below this, there are tabs for Essentials, Functions, Metrics, Properties, and Notifications (0). The Essentials tab is active, displaying various properties of the Function App in a key-value format, such as Resource group (azurfuncrg), Status (Running), Location (Canada Central), Subscription (Azure for Students), Subscription ID, Default domain (myfunc23.azurewebsites.net), Operating System (Windows), App Service Plan (ASP-azurfuncrg-b8c9 (V1.0)), and Runtime version (4.638.200.25164). At the bottom, there is a table for Functions with columns for Name, Trigger, Status, and Monitor. One function, "HttpTrigger1", is listed with an HTTP trigger and an "Enabled" status.

3. We created a Logic App in the same region and resource group, and used the Code View to define an HTTP request trigger. This sets up a workflow that starts on receiving an HTTP POST request.

The screenshot displays the "Logic app code view" for a Logic app named "mylogicapp". The interface includes a search bar, a "Save" button, and a "Discard" button. A left sidebar shows a navigation menu with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Development Tools, Logic app designer, Logic app code view (selected), Logic app templates, Run history, Versions, API connections, Quick start guides, Settings, and Settings. The main area shows the JSON definition of the Logic app workflow. The JSON is as follows:

```
1
2
3 {
4   "definition": {
5     "$schema":
6     "https://schema.management.azure.com/providers/Microsoft.Logic/schemas/2016-06-01/workflowdefinition.json#",
7     "actions": {},
8     "contentVersion": "1.0.0.0",
9     "outputs": {},
10    "parameters": {},
11    "triggers": {
12      "manual": {
13        "inputs": {
14          "schema": {
15            "properties": {
16              "name": {
17                "type": "string"
18              }
19            },
20            "type": "object"
21          },
22          "kind": "Http",
23          "type": "Request"
24        }
25      }
26    },
27    "parameters": {}
28  }
29 }
```

Home > mylogicapp ...

Run details Resubmit Cancel run Refresh Info File a bug

Workflow diagram: manual trigger → myfunc23-HttpTrigger1 function.

myfunc23-HttpTrigger1 Parameters:

- Request Body: {"name": "name"}
- Advanced parameters: Showing 0 of 4

Run details summary:

- manual: 0s ✓
- myfunc23-HttpTrigger1: 1s ✓

13. We cleaned up the resources by deleting the Function App and Logic App. This frees up cloud resources and avoids incurring additional costs.

Delete a resource group ×

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

rgwebapp

Enter resource group name to confirm deletion *

rgwebapp

Delete Cancel

Conclusion:

In this lab, we explored the deployment and scaling capabilities of Azure Web Apps, including staging slots and autoscaling rules. We also implemented a Function App and integrated it with a Logic App to automate workflows via HTTP triggers. These exercises demonstrated key concepts in app hosting, deployment, and serverless computing. By completing these tasks, we gained hands-on experience with building scalable and event-driven solutions on Azure.