

TP5 Report

WEB APP, FUNCTION APP AND LOGIC APP



By: Nadia FRIKHA and Rania MIDAOU, RT4-G2

Introduction:

As businesses continue to embrace digital transformation, cloud computing has emerged as a key enabler, providing scalable and cost-effective solutions for running applications and services.

Azure Web App, Azure Functions, and Azure Logic App are three critical cloud services offered by Microsoft Azure that allow businesses to build and deploy cloud-based solutions quickly and efficiently. By leveraging the power and flexibility of these services, we can help businesses streamline their operations and stay competitive in a rapidly evolving digital landscape.

Azure Web App :

Azure Web App is a fully managed platform for building, deploying, and scaling web apps. It provides a range of features to streamline the development process, including support for multiple programming languages and frameworks, automatic scaling, and built-in DevOps tools. With Azure Web App, developers can focus on writing code rather than managing infrastructure, making it an ideal choice for building and deploying web applications quickly and efficiently.

Azure Functions:

Azure Functions is a serverless compute service that enables developers to build event-driven applications. With Functions, developers can write code to respond to events such as changes in data, user actions, or messages from other services, without worrying about the underlying infrastructure. This makes Functions an ideal choice for building applications that require rapid scaling and high availability, as well as for integrating disparate services and systems.

Azure Logic App:

Azure Logic App is a cloud-based service that allows users to create and run workflows that integrate various applications and services. Logic Apps provide a visual interface for defining workflows, which can include actions such as data transformation, conditional logic, and integration with third-party services. With Logic Apps, users can automate business processes and workflows, making it easier to manage complex systems

and streamline operations. Logic Apps can integrate with a wide range of services, including Office 365, Dynamics 365, Salesforce, and many others.

Questions:

Task 1: Azure Web App:

1. We create a Web App named tp5webapp:

The screenshot shows the 'Create Web App' form in the Azure Portal. The form is titled 'Create Web App' and includes a search bar at the top. Below the search bar, there is a section for 'Subscription' and 'Resource Group'. The 'Subscription' dropdown is set to 'Azure pour les étudiants' and the 'Resource Group' dropdown is set to '(New) rgwebapp'. Below this, there is a section for 'Instance Details'. The 'Name' field is 'tp5webapp' and the 'Publish' dropdown is set to 'Code'. The 'Runtime stack' dropdown is set to 'PHP 8.2' and the 'Operating System' dropdown is set to 'Linux'. The 'Region' dropdown is set to 'East US'. Below the 'Instance Details' section, there is a 'Pricing plans' section. At the bottom of the form, there are three buttons: 'Review + create', '< Previous', and 'Next : Deployment >'. The 'Review + create' button is highlighted in blue.

2- We review the app link

The screenshot shows the 'tp5webapp' overview page in the Azure Portal. The page is titled 'tp5webapp' and includes a search bar at the top. Below the search bar, there is a section for 'Essentials'. The 'Essentials' section includes the following information: 'Resource group (move) : rgwebapp', 'Status : Running', 'Location (move) : East US', 'Subscription (move) : Azure pour les étudiants', and 'Subscription ID : 5dfcf723-ea7b-4a89-b344-86968954d351'. Below the 'Essentials' section, there is a 'Properties' section. The 'Properties' section includes the following information: 'Name : tp5webapp', 'Publishing model : Code', 'Runtime Stack : Php - 8.2', 'Default domain : tp5webapp.azurewebsites.net', and 'Custom domain : Add custom domain'. Below the 'Properties' section, there is a 'Deployment Center' section. The 'Deployment Center' section includes the following information: 'Deployment logs : View logs', 'Last deployment : No deployments found', and 'Deployment provider : None'. Below the 'Deployment Center' section, there is an 'Application Insights' section. The 'Application Insights' section includes the following information: 'Name : Enable Application Insights'. Below the 'Application Insights' section, there is a 'Networking' section. The 'Networking' section includes the following information: 'Virtual IP address : 20.119.16.20'. The 'Default domain' field in the 'Properties' section is highlighted with a red box.

We now update the SKU to standard and create a new slot (staging slot):

The screenshot shows the 'Spec Picker' interface in the Microsoft Azure portal. At the top, there's a search bar and a notification banner that says 'Updating App Service Plan' with the message 'Updating the plan ASP-rgwebapp-a8bf'. Below the search bar, there are three tabs: 'Dev / Test', 'Production', and 'Isolated'. The 'Production' tab is selected. Underneath, there are 'Recommended pricing tiers' displayed as purple boxes. The first tier, P1V2, is highlighted with a blue border. It lists: 210 total ACU, 3.5 GB memory, Dv2-Series compute equivalent, and 73.73 USD/Month (Estimated). Other tiers include P2V2, P3V2, P1V3, P2V3, and P3V3. Below the pricing tiers, there are sections for 'Included features' (Custom domains / SSL, Auto scale) and 'Included hardware' (Azure Compute Units (ACU), Memory). A 'See additional options' button is located below the pricing tiers.

The screenshot shows a green checkmark icon followed by the text 'Updating App Service Plan'. Below this, it says 'The plan 'ASP-rgwebapp-a8bf' was updated successfully!'. There is a close button (X) in the top right corner of the message box.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a navigation pane with various options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Microsoft Defender for Cloud', 'Events (preview)', 'Deployment', 'Deployment slots', 'Deployment Center', 'Settings', 'Configuration', 'Authentication', 'Application Insights', 'Identity', 'Backups', and 'Custom domains'. The 'Deployment slots' option is selected. In the main area, there's a 'Deployment Slots' section with a table showing the current slot. The table has columns for 'NAME', 'STATUS', and 'APP SER'. The row shows 'tp5webapp' with a 'PRODUCTION' status and 'Running' status, and 'ASP-rgwebapp' as the app service. To the right, there's a 'Add a slot' dialog box. It has a 'Name' field with the value 'staging' and a 'Clone settings from' dropdown menu set to 'Do not clone settings'. There are 'Add' and 'Close' buttons at the bottom of the dialog.

NAME	STATUS	APP SER
tp5webapp	PRODUCTION	Running
		ASP-rgwebapp

Microsoft Azure

Home > tp5webapp | Deployment slots >

staging (tp5webapp/staging)

App Service (Slot)

Search

Overview

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

Deployment

- Deployment slots
- Deployment Center

Settings

- Configuration
- Authentication
- Application Insights
- Identity
- Backups
- Custom domains

Essentials

Resource group (move) : rgwebapp

Status : Running

Location (move) : East US

Subscription (move) : Azure pour les étudiants

Subscription ID : 5dfcf723-ea7b-4a89-b344-86968954d351

Tags (edit) : Click here to add tags

Default domain : tp5webapp-staging.azurewebsites.net

App Service Plan : ASP-rgwebapp-a8bf (P1V2: 0)

Operating System : Linux

Health Check : Not Configured

Properties

Monitoring

Logs

Capabilities

Notifications

Recommendations

Web app

Name : tp5webapp/staging

Publishing model : Code

Runtime Stack : Php - 8.2

Domains

Default domain : tp5webapp-staging.azurewebsites.net

Custom domain : Add custom domain

Deployment Center

Deployment logs : View logs

Last deployment : No deployments found

Deployment provider : None

Application Insights

Name : Enable Application Insights

Networking

Virtual IP address : 20.119.16.29

3- Configure Web app deployment settings:

Microsoft Azure

Home > tp5webapp | Deployment slots > staging (tp5webapp/staging)

staging (tp5webapp/staging) | Deployment Center

App Service (Slot)

Search

Overview

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

Deployment

- Deployment slots
- Deployment Center

Settings

- Configuration
- Authentication
- Application Insights
- Identity
- Backups
- Custom domains

Settings

Logs

FTPS credentials

Deploy and build code from your preferred source and build provider. [Learn more](#)

Source *

Local Git

Building with App Service Build Service.

Local Git

Local Git allows you to host a simple Git server for your app on your App Service plan. This can be used to quickly setup a CI/CD pipeline. [Learn more](#)

Repository

Your local git repository url will be generated upon completion.

Branch

master

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information. The breadcrumb trail indicates the path: Home > tp5webapp | Deployment slots > staging (tp5webapp/staging). The main heading is 'staging (tp5webapp/staging) | Deployment Center'. The left sidebar contains various navigation options under 'Deployment' and 'Settings'. The main content area shows the 'Settings' tab for the 'Local Git' source. The 'Git Clone Uri' field is highlighted with a red box, containing the URL: `https://tp5webapp-staging.scm.azurewebsites.net:443/tp5webapp.git`. Other settings visible include 'Build provider' (App Service Build Service), 'Runtime stack' (PHP), and 'Version' (PHP 8.2).

We specify user credentials:

The screenshot shows the Microsoft Azure portal interface, specifically the 'Application scope' and 'User scope' sections of the 'staging (tp5webapp/staging)' Deployment Center settings. The 'Git Clone Uri' is still visible at the top. The 'Application scope' section includes fields for 'FTPS Username' (tp5webapp_staging\tp5webapp_staging), 'Local Git Username' (\$tp5webapp_staging), and 'Password'. The 'User scope' section includes fields for 'Username' (tp5webapp), 'Password', and 'Confirm Password'. The 'Password' fields have a 'Reset' button next to them.

4- Deploy code to staging:

```
PS /home/raniamidaoui/php-docs-hello-world> git clone https://github.com/Azure-Samples/php-docs-hello-world
Cloning into 'php-docs-hello-world'...
remote: Enumerating objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Receiving objects: 100% (26/26), 5.64 KiB | 5.64 MiB/s, done.
Resolving deltas: 100% (6/6), done.
PS /home/raniamidaoui/php-docs-hello-world> Set-Location -Path $HOME/php-docs-hello-world/
```

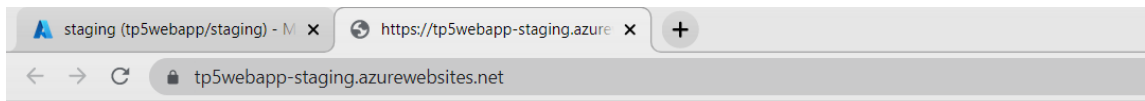
```
PS /home/raniamidaoui/php-docs-hello-world> git remote add tp5webapp123 https://tp5webapp-staging.scm.azurewebsites.net:443/tp5webapp.git
fatal: not a git repository (or any parent up to mount point /home)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).
PS /home/raniamidaoui/php-docs-hello-world> git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/raniamidaoui/php-docs-hello-world/.git/
PS /home/raniamidaoui/php-docs-hello-world> git remote add tp5webapp123 https://tp5webapp-staging.scm.azurewebsites.net:443/tp5webapp.git
PS /home/raniamidaoui/php-docs-hello-world> ls
```

```
PS /home/raniamidaoui/php-docs-hello-world> git push tp5webapp123 master
```

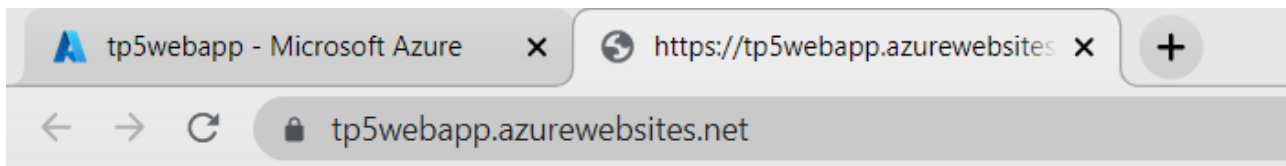
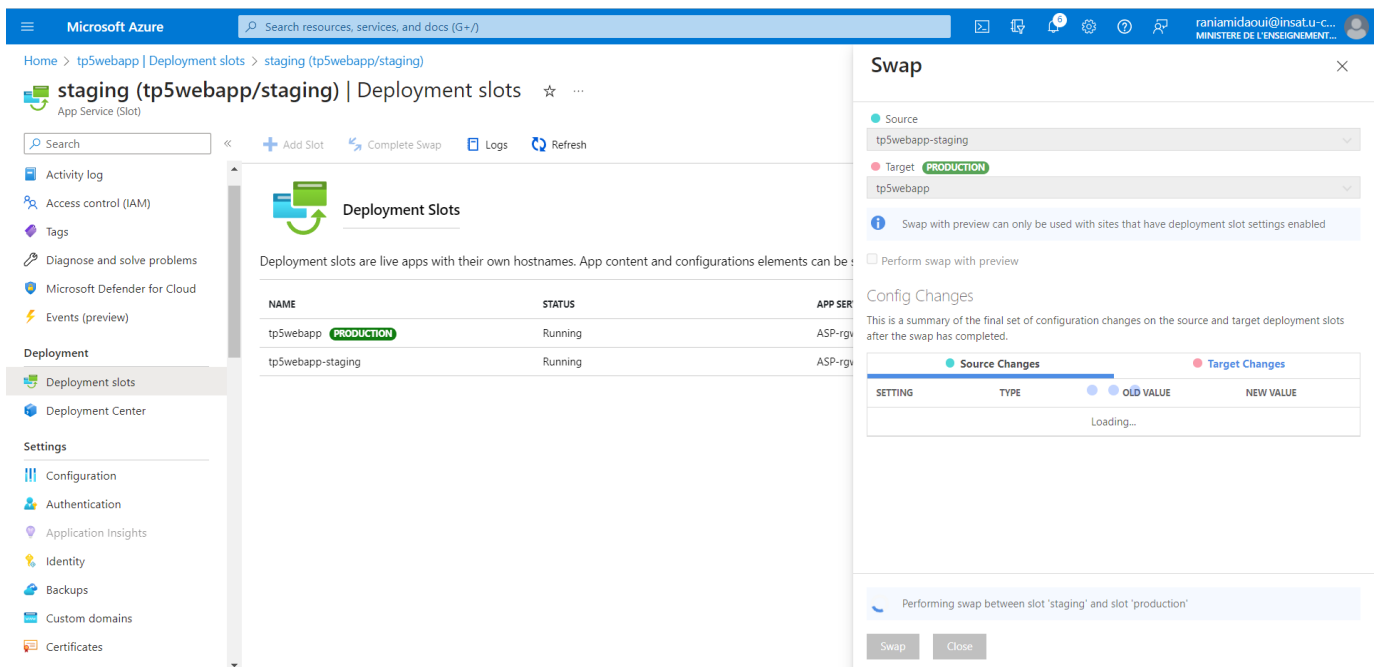
```
Username for 'https://tp5webapp-staging.scm.azurewebsites.net': tp5webapp123
Password for 'https://tp5webapp123@tp5webapp-staging.scm.azurewebsites.net':
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 2 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (26/26), 5.64 KiB | 5.64 MiB/s, done.
Total 26 (delta 6), reused 26 (delta 6), pack-reused 0
remote: Deploy Async
remote: Updating branch 'master'.
remote: Updating submodules.
remote: Preparing deployment for commit id 'df425ea6ef'.
remote: PreDeployment: context.CleanOutputPath False
remote: PreDeployment: context.OutputPath /home/site/wwwroot
remote: Repository path is /home/site/repository
remote: Running oryx build...
remote: Operation performed by Microsoft Oryx, https://github.com/Microsoft/Oryx
remote: You can report issues at https://github.com/Microsoft/Oryx/issues
remote:
```

```
remote: Verifying checksum...
remote: Extracting contents...
remote: performing sha512 checksum for: php-composer...
remote: Done in 0 sec(s).
remote:
remote: PHP executable: /tmp/oryx/platforms/php/8.2.5/bin/php
remote: No 'composer.json' file found; not running 'composer install'.
remote: Preparing output...
remote:
remote: Copying files to destination directory '/home/site/wwwroot'...
remote: Done in 0 sec(s).
remote:
remote: Removing existing manifest file
remote: Creating a manifest file...
remote: Manifest file created.
remote: Copying .ostype to manifest output directory.
remote:
remote: Done in 2 sec(s).
remote: Running post deployment command(s)...
remote:
remote: Generating summary of Oryx build
remote: Parsing the build logs
remote: Found 0 issue(s)
remote:
remote: Build Summary :
remote: =====
remote: Errors (0)
remote: Warnings (0)
remote:
remote: Triggering recycle (preview mode disabled).
remote: Deployment successful. deployer = deploymentPath =
remote: Deployment Logs : 'https://tp5webapp-staging.scm.azurewebsites.net/newui/jsonviewer?view_url=/api/deployments/df425ea6ef61f981c71537ec89d1d821a2de975c/log'
To https://tp5webapp-staging.scm.azurewebsites.net/tp5webapp.git
* [new branch]      master -> master
```

Done, everything works:



5- We swap the staging slots:



Hello World!

6- We configure and test autoscaling of the Azure web app:

The screenshot shows the Azure portal interface for configuring autoscaling on a web app named 'tp5webapp'. The left sidebar shows the 'Scale out (App Service plan)' option selected. The main area displays the 'Default' auto-created scale condition. The 'Scale rule' panel on the right shows the configuration for a metric-based scale rule.

Scale rule configuration:

- Metric source:** Current resource (ASP-rgwebapp-a8bf)
- Resource type:** App Service plans
- Resource:** ASP-rgwebapp-a8bf
- Criteria:**
 - Metric namespace:** Standard metrics
 - Metric name:** CPU Percentage
 - Dimension Name:** Instance
 - Operator:** =
 - Dimension Values:** All values
- Action:** Increase count by 1
- Cool down (minutes):** 5

The screenshot shows the Azure portal interface for configuring autoscaling on a web app named 'tp5webapp'. The left sidebar shows the 'Scale out (App Service plan)' option selected. The main area displays the 'Custom autoscale' configuration. The 'Scale rule' panel on the right shows the configuration for a metric-based scale rule.

Custom autoscale configuration:

- Autoscale setting name:** ASP-rgwebapp-a8bf-Autoscale-193
- Resource group:** rgwebapp
- Instance count:** 1

Scale rule configuration:

- Metric source:** Current resource (ASP-rgwebapp-a8bf)
- Resource type:** App Service plans
- Resource:** ASP-rgwebapp-a8bf
- Criteria:**
 - Metric namespace:** Standard metrics
 - Metric name:** CPU Percentage
 - Dimension Name:** Instance
 - Operator:** =
 - Dimension Values:** All values
- Action:** Increase count by 1
- Cool down (minutes):** 5

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

Home > tp5webapp

tp5webapp | Scale out (App Service plan) ☆ ...

Web App

Search

Deployment Center

Settings

- Configuration
- Authentication
- Application Insights
- Identity
- Backups
- Custom domains
- Certificates
- Networking
- Scale up (App Service plan)
- Scale out (App Service plan)
- Service Connector
- Locks

App Service plan

- App Service plan
- Quotas

Resource group: rgwebapp

Instance count: 1

Default * Auto created default scale condition

Delete warning: The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode: ☒ Scale based on a metric ☐ Scale to a specific instance count

Rules: It is recommended to have at least one scale in rule. To create new rules, click [Add a rule](#)

Scale out

When	ASP-rgwebapp-a8bf	(Maximum) CpuPercentage > 10	Increase count by 1
+ Add a rule			

Instance limits

Minimum *	Maximum *	Default *
1	3	1

Schedule: This scale condition is executed when none of the other scale condition(s) match

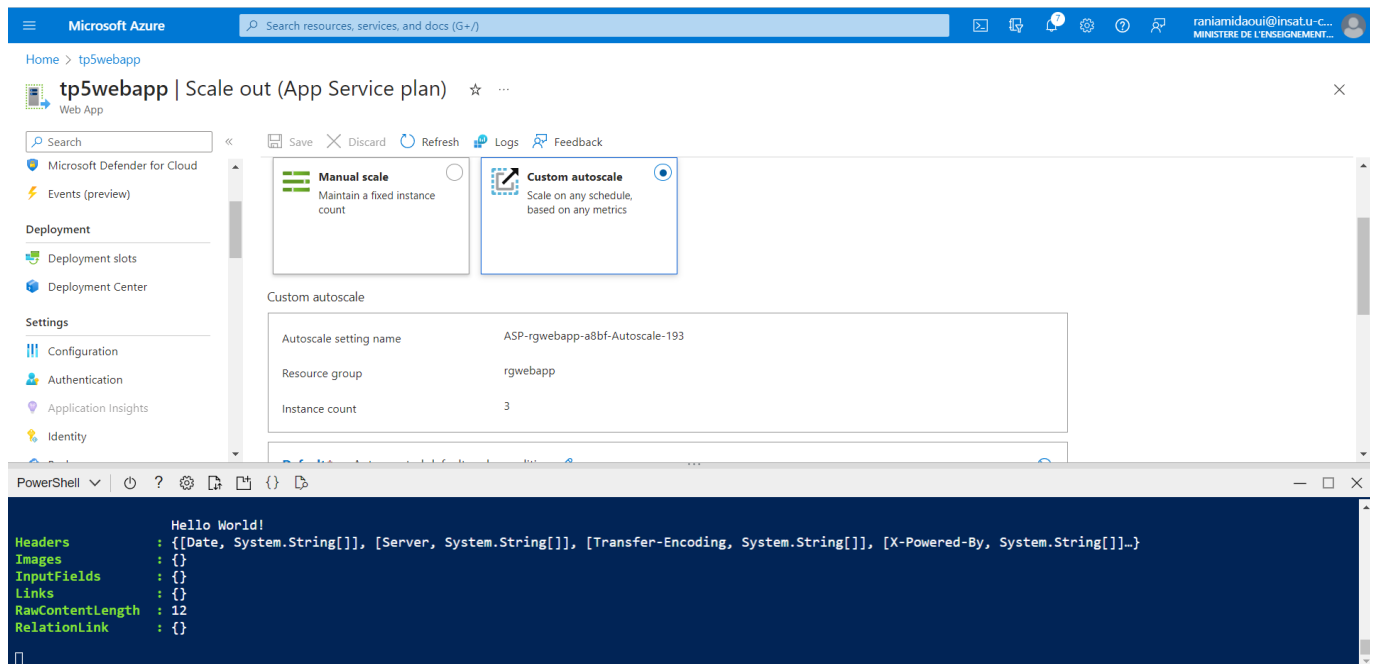
+ Add a scale condition

```
PS /home/raniamidaoui> $rgName = 'rgwebapp'
PS /home/raniamidaoui> $webapp = Get-AzWebApp -ResourceGroupName $rgName
PS /home/raniamidaoui>
```

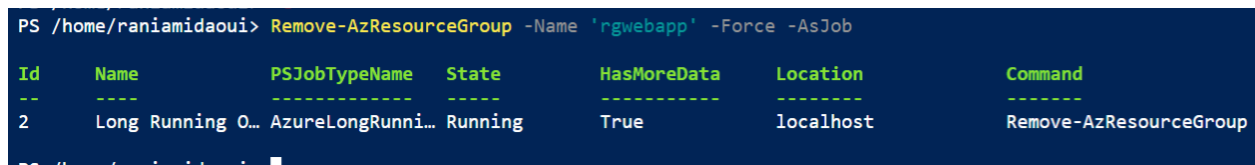
```
PS /home/raniamidaoui> while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }

StatusCode      : 200
StatusDescription : OK
Content         : Hello World!
RawContent      : HTTP/1.1 200 OK
                  Date: Sun, 07 May 2023 20:05:16 GMT
                  Server: nginx/1.22.1
                  Transfer-Encoding: chunked
                  X-Powered-By: PHP/8.2.5
                  Content-Type: text/html; charset=utf-8

                  Hello World!
Headers         : {[Date, System.String[]], [Server, System.String[]], [Transfer-Encoding, System.String[]], [X-Powered-By, System.String[]]...}
Images          : {}
InputFields     : {}
Links           : {}
RawContentLength : 12
RelationLink    : {}
```

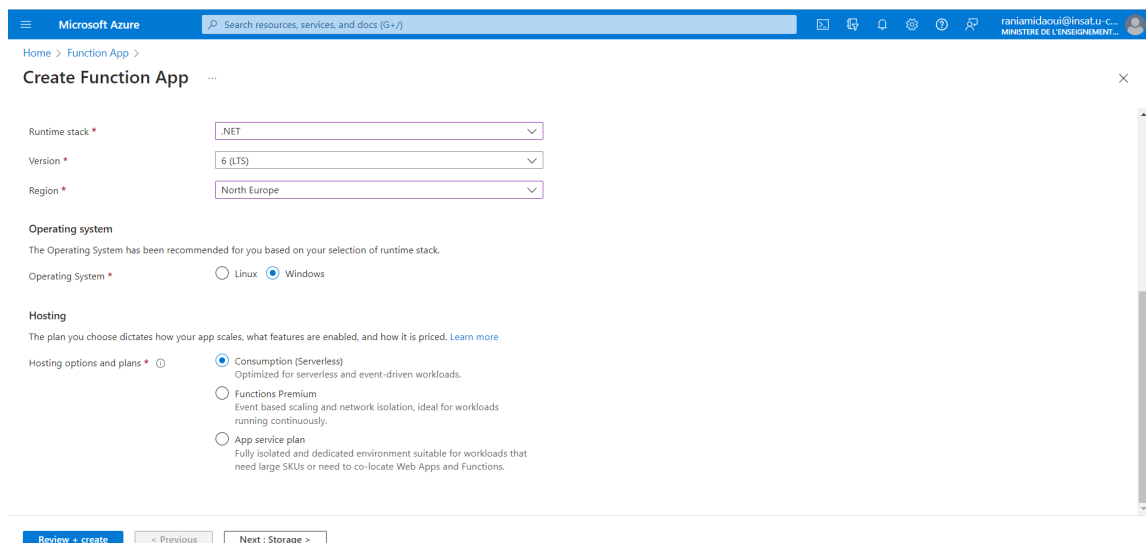


7- We remove the resource group:



Task 2: Azure Functions and Azure Logic App:

1- We create a Function App:



2- We create a function with HTTP trigger:

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation pane is visible with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Functions, App keys, App files, Proxies, Deployment, Deployment slots, Deployment Center, Settings, and Configuration. The main area displays the 'Create function' dialog. The 'Development environment' is set to 'Develop in portal'. The 'Select a template' section shows a list of templates, with 'HTTP trigger' selected. The 'Create' button is highlighted.

Template	Description
HTTP trigger	Fonction exécutée quand elle reçoit une requête HTTP (réponse basée sur les données dans la chaîne du corps ou de la requête)
Timer trigger	Fonction exécutée selon une planification définie
Azure Queue Storage trigger	Fonction exécutée quand un message est ajouté à une file d'attente Stockage Azure spécifiée
Azure Service Bus Queue trigger	Fonction exécutée quand un message est ajouté à une file d'attente Service Bus spécifiée
Azure Service Bus Topic trigger	Fonction exécutée quand un message est ajouté à la rubrique Service Bus spécifiée
Azure Blob Storage trigger	Fonction exécutée quand un objet blob est ajouté à un conteneur spécifié
Azure Event Hub trigger	Fonction exécutée quand un hub d'événements reçoit un nouvel événement

The screenshot shows the Microsoft Azure portal interface. The main area displays the 'HttpTrigger1 | Code + Test' view. The code editor shows the C# code for the HTTP trigger function. The 'Logs' section at the bottom shows the function's execution logs.

```
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<ActionResult> 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;
```

Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, please go to 'Monitor' from the Function menu.

The screenshot shows the 'HttpTrigger1' function integration page in the Microsoft Azure portal. The page title is 'HttpTrigger1 | Integration'. The left sidebar shows the 'Integration' tab selected. The main content area shows a diagram of the function's integration. The diagram consists of four boxes: 'Trigger' (HTTP req), 'Inputs' (No inputs defined), 'Function' (HttpTrigger1), and 'Outputs' (HTTP (\$return)). Arrows indicate the flow from the Trigger to the Function, and from the Function to the Outputs. The 'Inputs' box is currently empty, and the 'Outputs' box has a '+ Add output' link.

3- We create a logic app that uses the function app:

The screenshot shows the 'Create Logic App' page in the Microsoft Azure portal. The page title is 'Create Logic App'. The left sidebar shows the 'Logic apps' tab selected. The main content area shows the 'Create Logic App' form. The form has several sections: 'Subscription' (Azure pour les étudiants), 'Resource Group' (functionapprg), 'Instance Details' (Logic App name: tp5logicapp, Region: North Europe, Enable log analytics: No), and 'Plan' (Standard: Best for enterprise-level, serverless applications, with event-based scaling and networking isolation; Consumption: Best for entry-level. Pay only as much as your workflow runs; Looking for the classic consumption create experience? Click here). The 'Plan' section is currently selected. The 'Zone redundancy (preview)' section is also visible, with a note about availability zones. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Tags >'.

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

Home > tp5logicapp >

Logic Apps Designer

Save Discard Run Trigger Designer Code view Parameters Templates Connectors Help Info Try Preview Designer

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

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

Home > tp5logicapp >

Logic Apps Designer

Save Discard Run Trigger Designer Code view Parameters Templates Connectors Help Info Try Preview Designer

*** Saving logic app...
Saving logic app tp5logicapp...

When a HTTP request is received

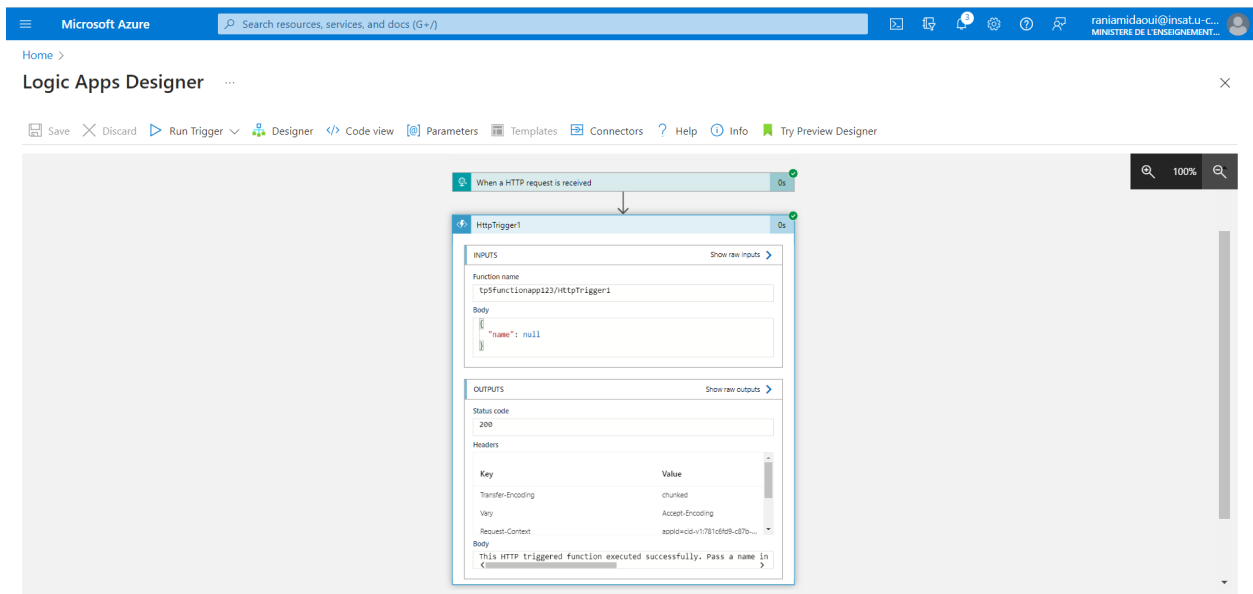
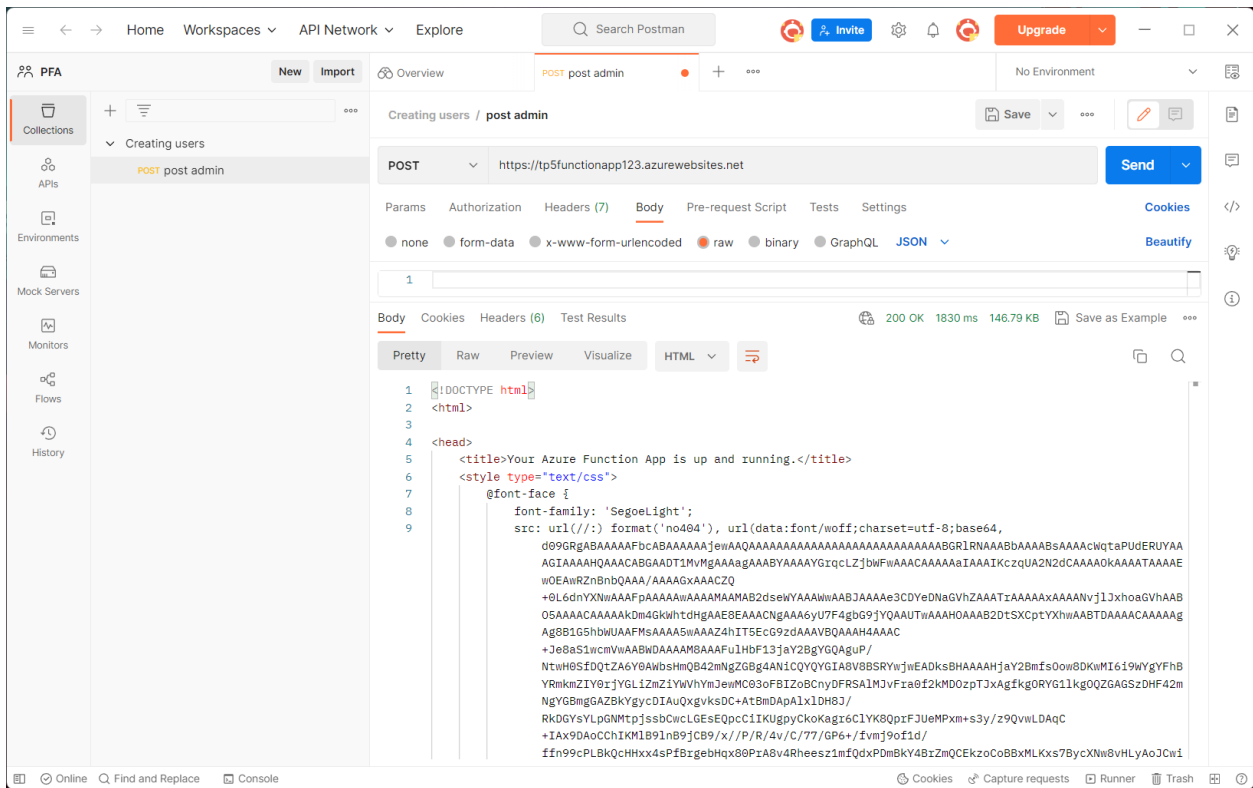
HttpTrigger1

Request Body: {"name": name x }

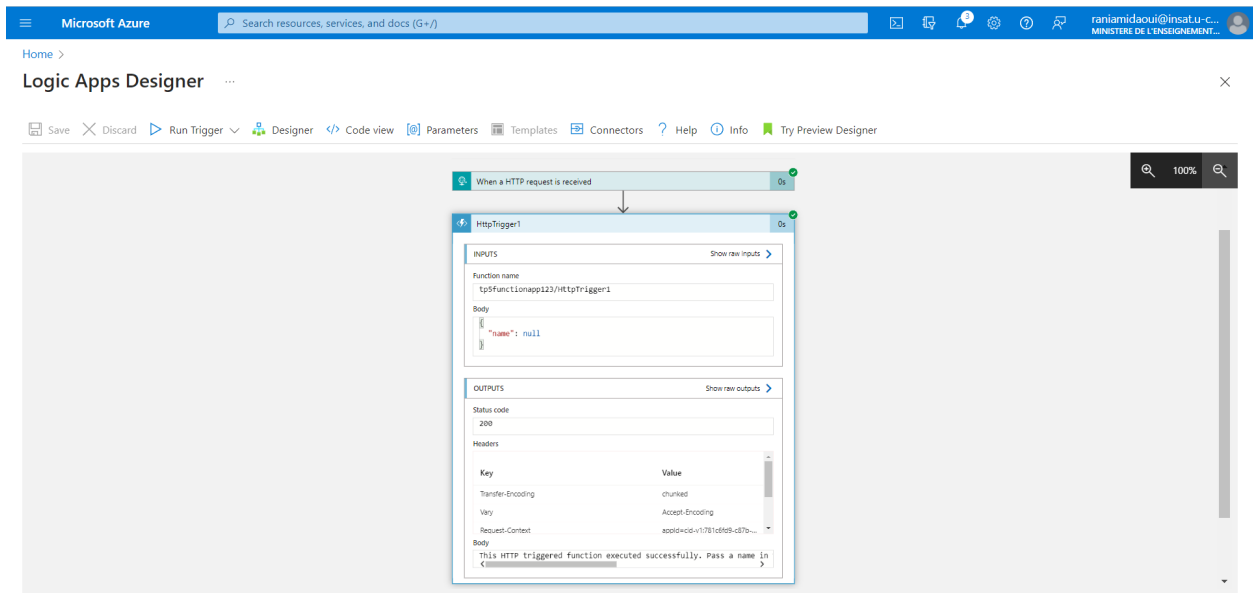
Add new parameter

+ New step

100%



4- Delete the resource group:



Conclusion:

In conclusion, Azure Web App, Azure Functions, and Azure Logic App are powerful cloud services offered by Microsoft Azure that enable businesses to develop and deploy cloud-based solutions with ease. These services provide a range of features to streamline the development process and automate workflows, making it easier to manage complex systems and stay competitive in a rapidly evolving digital landscape.

By leveraging the capabilities of these services, businesses can build robust and responsive web applications that can handle diverse workflows and tasks, and integrate with a wide range of services and systems.