

Lab Report 4 (Web APP) - Antonio Manuel Luque Molina

In this lab,

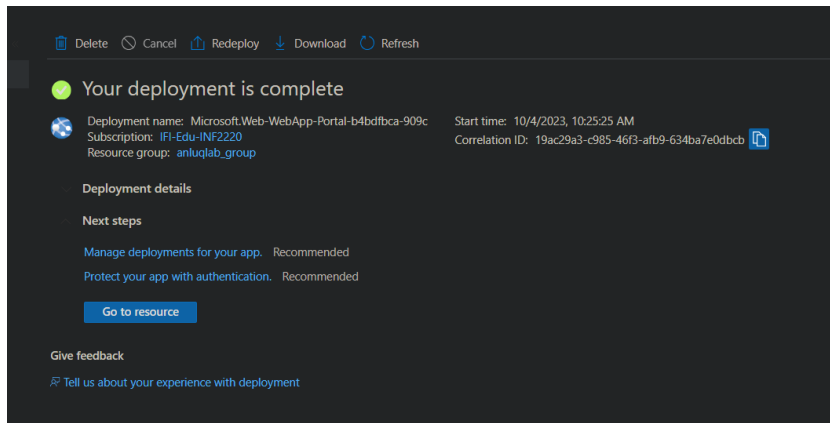
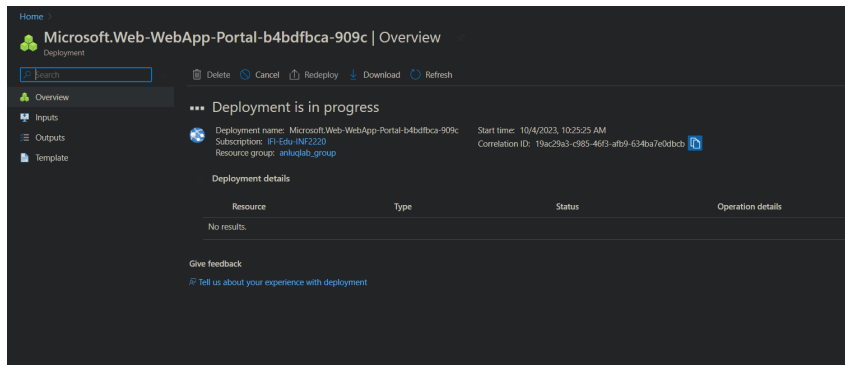
- We will create a web app in azure.
- Code in python to create an application.
- Deploy our code/application to the app service.

First of all We will go to Microsoft Azure and We will go to the marketplace and We will go to Create Web App:

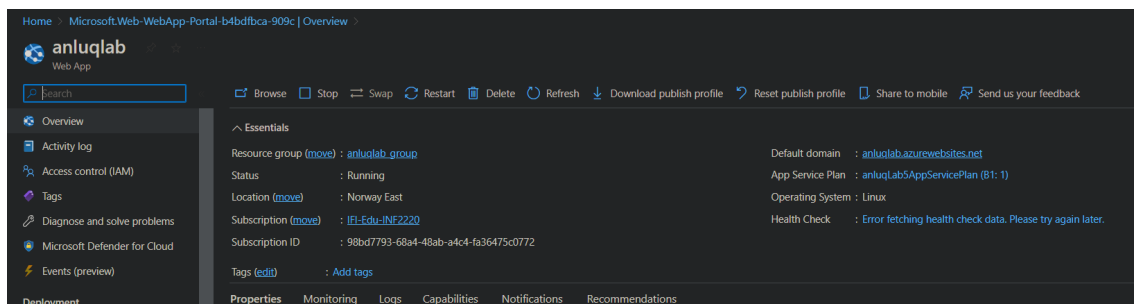
The screenshot shows the 'Create Web App' wizard in the Microsoft Azure portal, specifically the 'Basic' tab. The breadcrumb navigation at the top reads 'Home > Create a resource > Marketplace >'. The title 'Create Web App' is prominently displayed. Below the title, there are tabs for 'Basic', 'Deployment', 'Networking', 'Monitoring', 'Tags', and 'Review + create'. A brief description of App Service Web Apps is provided, followed by a 'Learn more' link. The 'Project Details' section instructs the user to select a subscription and a resource group. The 'Subscription' dropdown is set to 'IFI-Edu-INF2220', and the 'Resource Group' dropdown is set to '(New) anluqlab_group', with a 'Create new' link below it. The 'Instance Details' section asks if the user needs a database, with a link to 'Try the new Web + Database experience'. The 'Name' field is 'anluqlab', and a green checkmark indicates it's valid. The 'Publish' section has three radio buttons: 'Code' (selected), 'Docker Container', and 'Static Web App'. The 'Runtime stack' is set to 'Python 3.8', and the 'Operating System' is set to 'Linux' (selected) over 'Windows'.

We will create a new Resource group for this and We will choose the following Instance Details and We will choose the B1 App Service Plan:

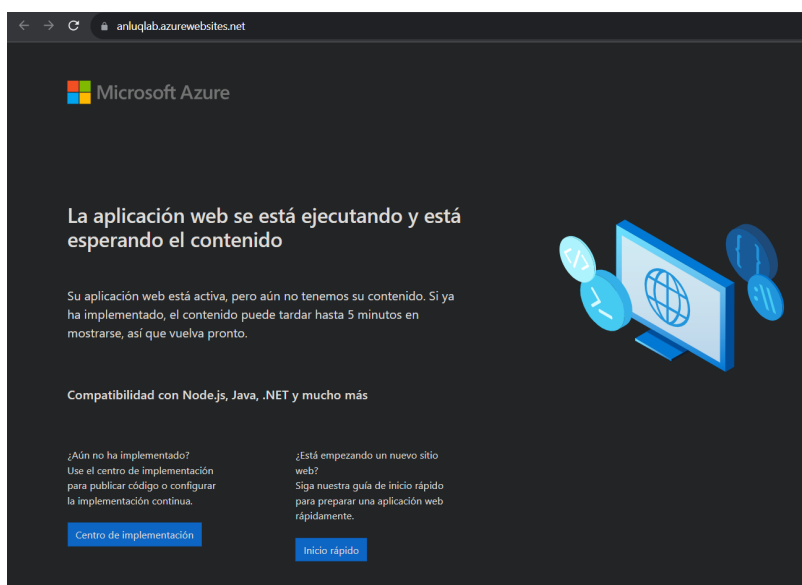
The screenshot shows the 'Pricing plans' tab of the 'Create Web App' wizard. The 'Region' dropdown is set to 'Norway East'. Below it, a message states: 'Not finding your App Service Plan? Try a different region or select your App Service Environment.' The 'Pricing plans' section explains that the App Service plan pricing tier determines location, features, cost, and compute resources, with a 'Learn more' link. The 'Linux Plan (Norway East)' dropdown is set to '(New) anluqlab5AppServicePlan', with a 'Create new' link below it. The 'Pricing plan' dropdown is set to 'Basic B1 (100 total ACU, 1.75 GB memory, 1 vCPU)', with an 'Explore pricing plans' link below it. The 'Zone redundancy' section explains that an App Service plan can be deployed as a zone redundant service. It has two radio buttons: 'Enabled' (unselected) and 'Disabled' (selected). The 'Disabled' option is described as: 'Your App Service Plan and the apps in it will not be zone redundant. The minimum App Service plan instance count will be one.' At the bottom, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Deployment >'.



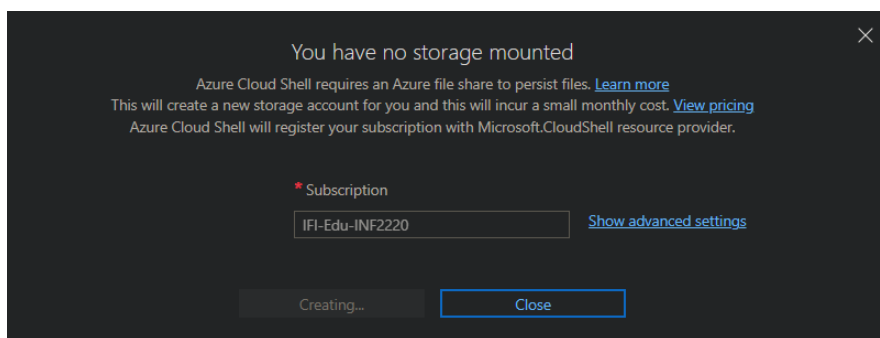
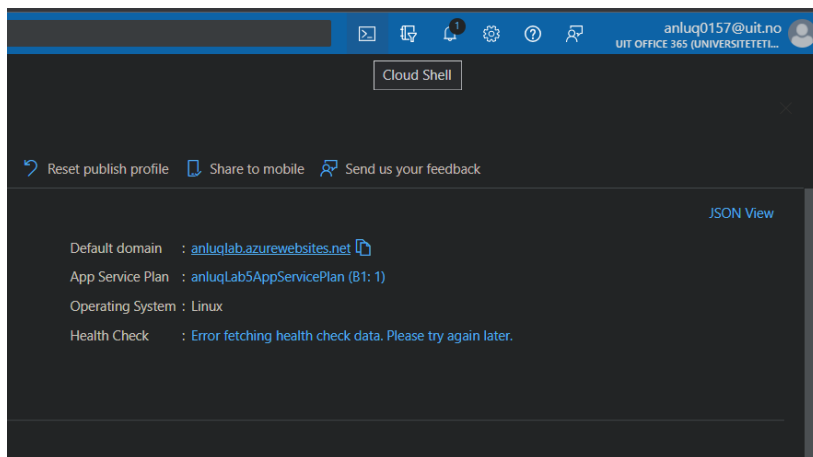
Once our deployment is complete We will go to our Web App:



We can go to the link of the website to see if it is running:

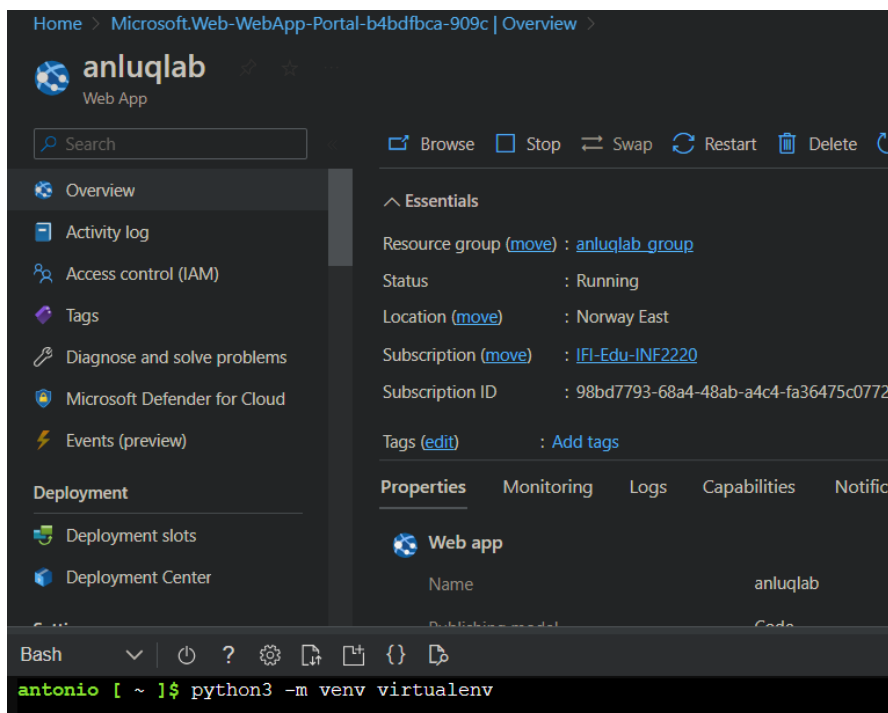


Now We are going to add more information to the website using python:



Here We are creating a storage mounted since We do not have one.

Once created we can start adding python code using the bash from Azure Cloud Shell:



With this We are creating a virtual environment for python.

```
antonio [ ~ ]$ source virtualenv/bin/activate
(virtualenv) antonio [ ~ ]$
```

With this We are activating the created virtual environment.

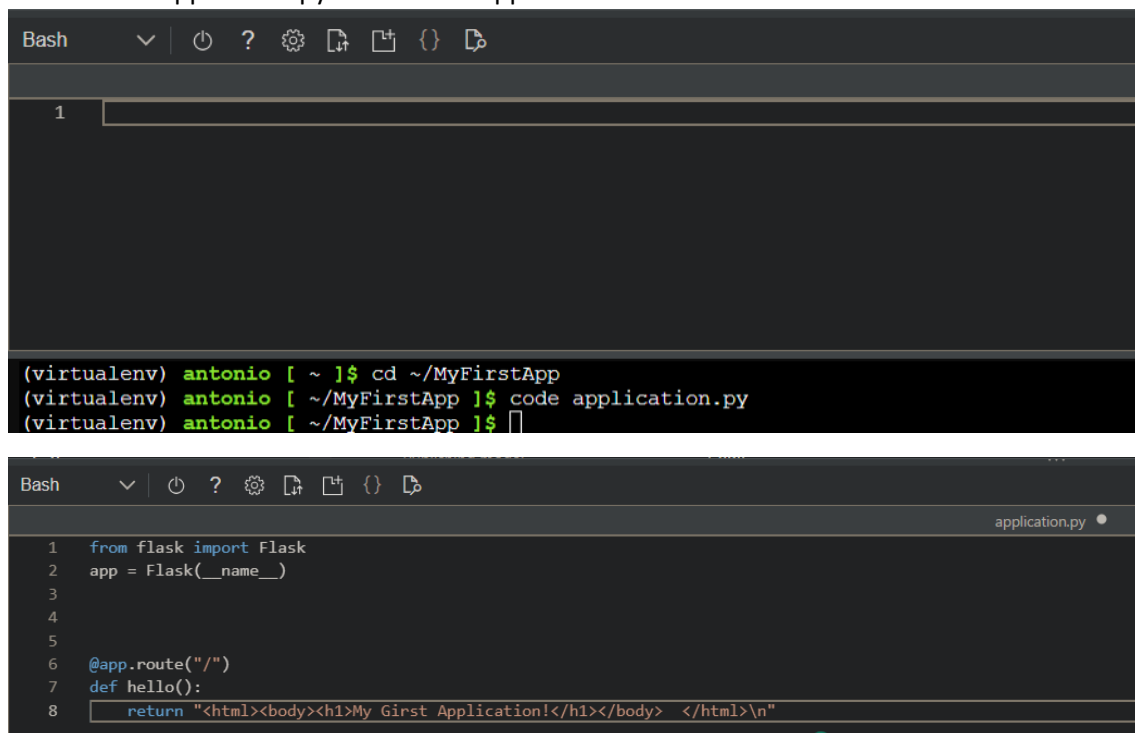
We will need a web application framework. For this, we will use flask. And We will install this with:

```
(virtualenv) antonio [ ~ ]$ pip install flask
Collecting flask
  Downloading flask-3.0.0-py3-none-any.whl (99 kB)
    99.7/99.7 KB 1.2 MB/s eta 0:00:00
```

Now, We will create a new directory for our web application and We will change our location to the created directory:

```
(virtualenv) antonio [ ~ ]$ mkdir ~/MyFirstApp
(virtualenv) antonio [ ~ ]$ cd ~/MyFirstApp
(virtualenv) antonio [ ~/MyFirstApp ]$
```

Once there We will type code **application.py** and this will open the interactive editor to create and edit the application.py for our web app.



The screenshot shows a code editor with a dark theme. The top toolbar includes icons for Bash, a dropdown menu, a power button, a question mark, a settings gear, a file icon, a folder icon, a code icon, and a document icon. The editor area shows a file named `application.py` with the following code:

```
1 from flask import Flask
2 app = Flask(__name__)
3
4
5
6 @app.route("/")
7 def hello():
8     return "<html><body><h1>My Girst Application!</h1></body> </html>\n"
```

We need to save the list of application requirement in the requirements.txt file. For this We will type **pip freeze > requirements.txt**

```
(virtualenv) antonio [ ~/MyFirstApp ]$ pip freeze > requirements.txt
```

Now We can run the created application with the following commands:

```
(virtualenv) antonio [ ~/MyFirstApp ]$ export FLASK_APP=application.py
(virtualenv) antonio [ ~/MyFirstApp ]$ flask run
* Serving Flask app 'application.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Now We will open other shell window:

```
Microsoft Azure
Search resources, se

Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Storage fileshare subscription 98bd7793-68a4-48ab-a4c4-fa36475c0772 is
re, unregistered subscriptions will have restricted access to CloudShell

antonio [ ~ ]$ curl http://127.0.0.1:5000/
<html><body><h1>My Girst Application!</h1></body> </html>
antonio [ ~ ]$
```

```
antonio [ ~ ]$ source virtualenv/bin/activate
(virtualenv) antonio [ ~ ]$ ls
clouddrive MyFirstApp virtualenv
(virtualenv) antonio [ ~ ]$ cd MyFirstApp/
(virtualenv) antonio [ ~/MyFirstApp ]$
```

Before deploying our application, we need to gather some information about our web app. We need to set shell variables that contains the web app's name, resource group name, plan name, sku, and location For this, We will run these following commands:

```
(virtualenv) antonio [ ~/MyFirstApp ]$ APPNAME=$(az webapp list --query [0].name --output tsv -g anluqlab_group)
(virtualenv) antonio [ ~/MyFirstApp ]$ APPRG=$(az webapp list --query [0].resourceGroup --output tsv -g anluqlab_group)
(virtualenv) antonio [ ~/MyFirstApp ]$ APPPLAN=$(az appservice plan list --query [0].name --output tsv -g anluqlab_group)
(virtualenv) antonio [ ~/MyFirstApp ]$ APPSKU=$(az appservice plan list --query [0].sku.name --output tsv -g anluqlab_group)
(virtualenv) antonio [ ~/MyFirstApp ]$ APPLOCATION=$(az appservice plan list --query [0].location --output tsv -g anluqlab_group)
(virtualenv) antonio [ ~/MyFirstApp ]$
```

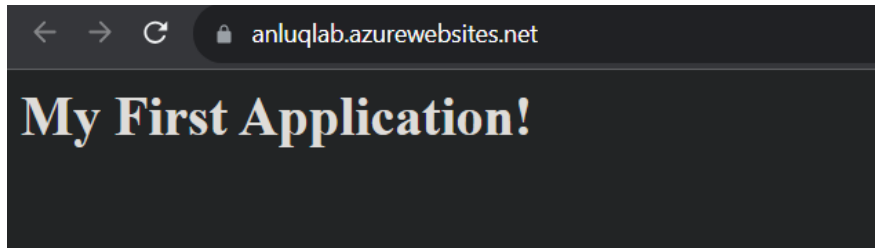
Now, We will check that all the variables above are set to the expected values from the azure portal:

```
(virtualenv) antonio [ ~/MyFirstApp ]$ echo "APPNAME=$APPNAME, APPRG=$APPRG, APPPLAN=$APPPLAN, APPSKU=$APPSKU, APPLOCATION=$APPLICATION"
APPNAME=anluqlab, APPRG=anluqlab_group, APPPLAN=anluqlab5AppServicePlan, APPSKU=B1, APPLOCATION=Norway East
(virtualenv) antonio [ ~/MyFirstApp ]$
```

Now we will deploy our python application:

```
Bash
(virtualenv) antonio [ ~/MyFirstApp ]$ az webapp up --name $APPNAME --resource-group $APPRG --plan $APPPLAN --sku $APPSKU --location "$APPLICATION"
Webapp 'anluqlab' already exists. The command will deploy contents to the existing app.
Creating AppServicePlan 'anluqlab5AppServicePlan' or Updating if already exists
Updating runtime version from PYTHON|3.8 to PYTHON|3.10
Waiting for runtime version to propagate ...
Creating zip with contents of dir /home/antonio/MyFirstApp ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
You can launch the app at http://anluqlab.azurewebsites.net
Setting 'az webapp up' default arguments for current directory. Manage defaults with 'az configure --scope local'
--resource-group/-g default: anluqlab_group
--sku default: B1
--plan/-p default: anluqlab5AppServicePlan
--location/-l default: norwayeast
--name/-n default: anluqlab
{
  "URL": "http://anluqlab.azurewebsites.net",
  "appserviceplan": "anluqlab5AppServicePlan",
  "location": "norwayeast",
  "name": "anluqlab",
  "os": "Linux",
  "resourcegroup": "anluqlab_group",
  "runtime_version": "python|3.10",
  "runtime_version_detected": "-",
  "sku": "BASIC",
  "src_path": "/home//antonio/MyFirstApp"
}
(virtualenv) antonio [ ~/MyFirstApp ]$
```

We will see a link to your application before the JSON block If We click on it We will get a new tab with our application:



Changing our app code adding an h2:

```
Bash  application.py
1  from flask import Flask
2  app = Flask(__name__)
3
4
5
6  @app.route("/")
7  def hello():
8      return "<html><body><h1>My First Application!</h1>\n<h2>Retesting my application adding an h2</h2></body></html>\n"
9
10
```

```
(virtualenv) antonio [ ~/MyFirstApp ]$ code application.py
(virtualenv) antonio [ ~/MyFirstApp ]$
```

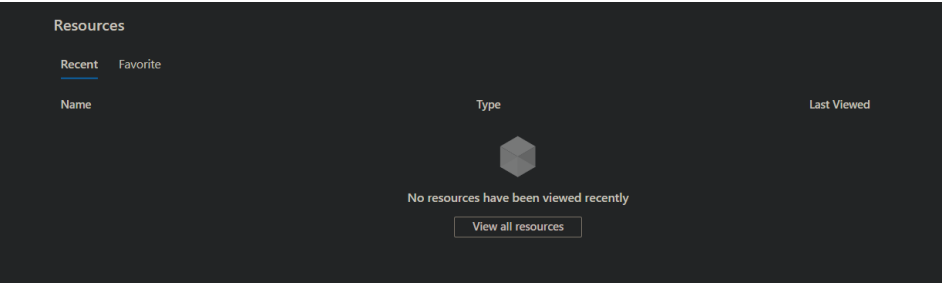
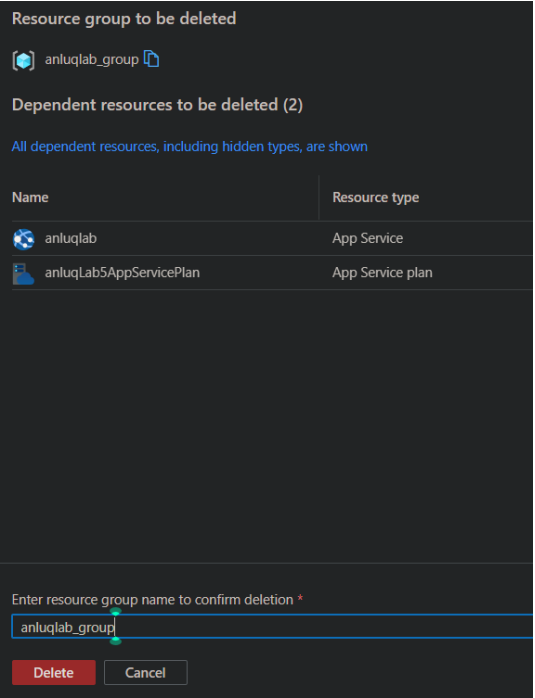
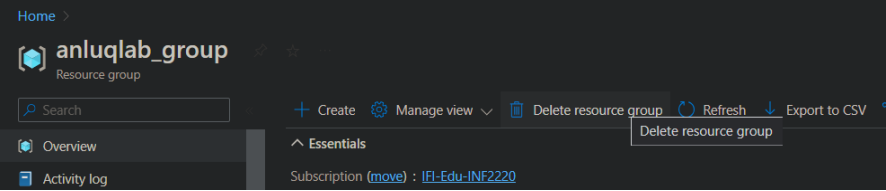
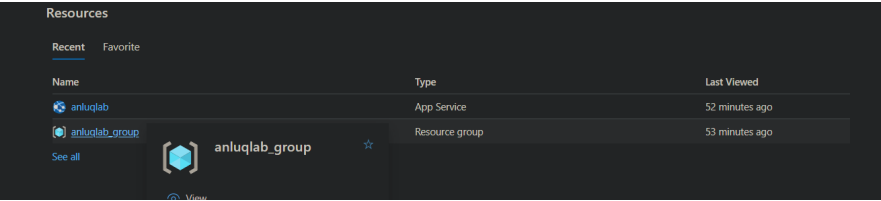
Now, We redploy the app using the earlier command

```
(virtualenv) antonio [ ~/MyFirstApp ]$ code application.py
(virtualenv) antonio [ ~/MyFirstApp ]$ az webapp up --name $APPNAME --resource-group $APPRG --plan $APPPLAN --sku $APPSKU --location "$APPLICATION"
Webapp 'anuqlab' already exists. The command will deploy contents to the existing app.
Creating AppServicePlan 'anuqlab5AppServicePlan' or Updating if already exists
Creating zip with contents of dir /home/antonio/MyFirstApp ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
You can launch the app at http://anuqlab.azurewebsites.net
Setting 'az webapp up' default arguments for current directory. Manage defaults with 'az configure --scope local'
--resource-group/-g default: anuqlab_group
--sku default: B1
--plan/-p default: anuqlab5AppServicePlan
--location/-l default: norwayeast
--name/-n default: anuqlab
{
  "URL": "http://anuqlab.azurewebsites.net",
  "appserviceplan": "anuqlab5AppServicePlan",
  "location": "norwayeast",
  "name": "anuqlab",
  "os": "Linux",
  "resourcegroup": "anuqlab_group",
  "runtime_version": "python|3.10",
  "runtime_version_detected": "-",
  "sku": "BASIC",
  "src_path": "/home/antonio/MyFirstApp"
}
(virtualenv) antonio [ ~/MyFirstApp ]$
```

Refreshing our web app We can see the changes:

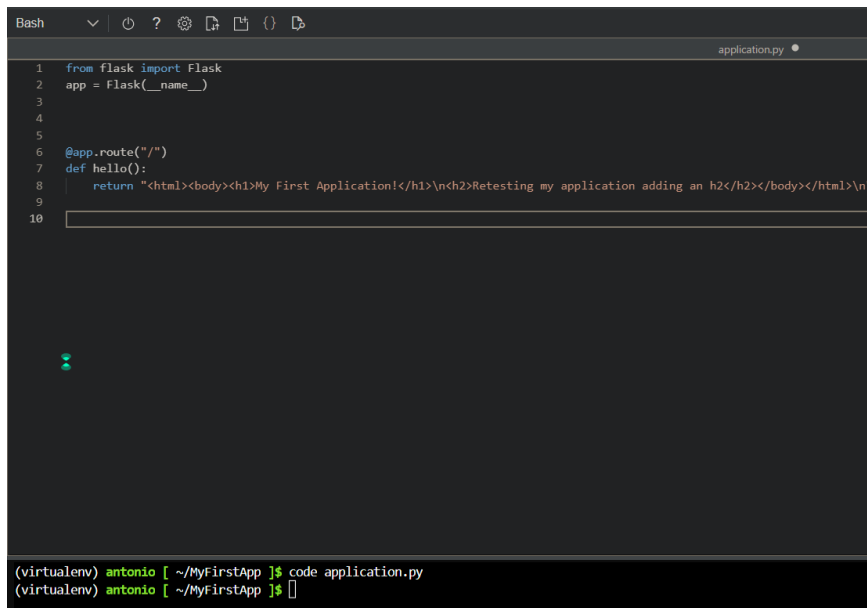


Now it is the time for delete all the resources:



I deleted all the resources in the colloquium class before doing the task of creating a new innerlink of the URL, i.e., display some content in this link: `$previouslink$/test/`

So, I am going to explain how I would have done that:



```
1 from flask import Flask
2 app = Flask(__name__)
3
4
5
6 @app.route("/")
7 def hello():
8     return "<html><body><h1>My First Application!</h1>\n<h2>Retesting my application adding an h2</h2></body></html>\n"
9
10
```

(virtualenv) antonio [~/MyFirstApp]\$ code application.py
(virtualenv) antonio [~/MyFirstApp]\$

Here I would add other `@app.route("/test")`, other `def hello():` and `return` `"<html><body><h1>My Second link!</h1></body></html>\n"`

After that I would redploy the app using the command: `'az webapp up --name $APPNAME --resource-group $APPRG --plan $APPPLAN --sku $APPSKU --location "$APPLOCATION"'`.

After this, Refreshing our web app We could see the changes.

That was all. Thank you very much for reading!