UDLAP®

Understanding Externalised Settings on the Cloud

Course: LIS4112-1 Cloud Computing And Big Data Lab

Section 1

Term: Spring 2022

David Fernando Rosas Chávez - 163031

Understanding Externalised Settings on the Cloud

In this practice we deployed a web application written in python and powered by the flask framework. For this, multiple cloud services were used which were interacting with each other, among them were the following:

- Azure
- Github
- Ngrok
- Collab

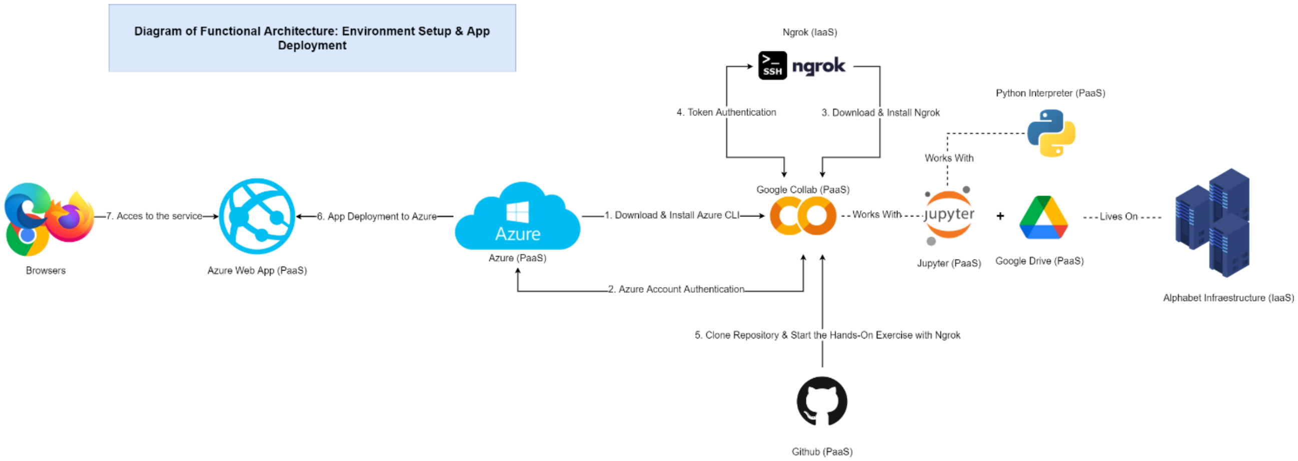
Diagram of Functional Architecture: Environment Setup & App Deployment
Ngrok (IaaS)
ngrok
SSH
4. Token Authentication
3. Download & Install Ngrok
Python Interpreter (PaaS)
Works With
Google Collab (PaaS)
Browsers
7. Acces to the service
Azure Web App (PaaS)
6. App Deployment to Azure
Azure
Azure (PaaS)
1. Download & Install Azure CLI
Works With
jupyter
Jupyter (PaaS)
+
Google Drive (PaaS)
Lives On
Alphabet Infraestructure (IaaS)
2. Azure Account Authentication
5. Clone Repository & Start the Hands-On Exercise with Ngrok
Github (PaaS)

Figure 1: Full Functional Architecture Diagram.

Azure Workflow

With respect to the Azure workflow, it was necessary to have an account created through the university's emailQ to enjoy some bonadades offered by the service, likewise, and within the google collab jupiter notebook, the Azure Cli application will be used to perform the creation of the app in the Azure service and its subsequent deployment, it should be noted that for all this it is required to perform an authentication through a token.

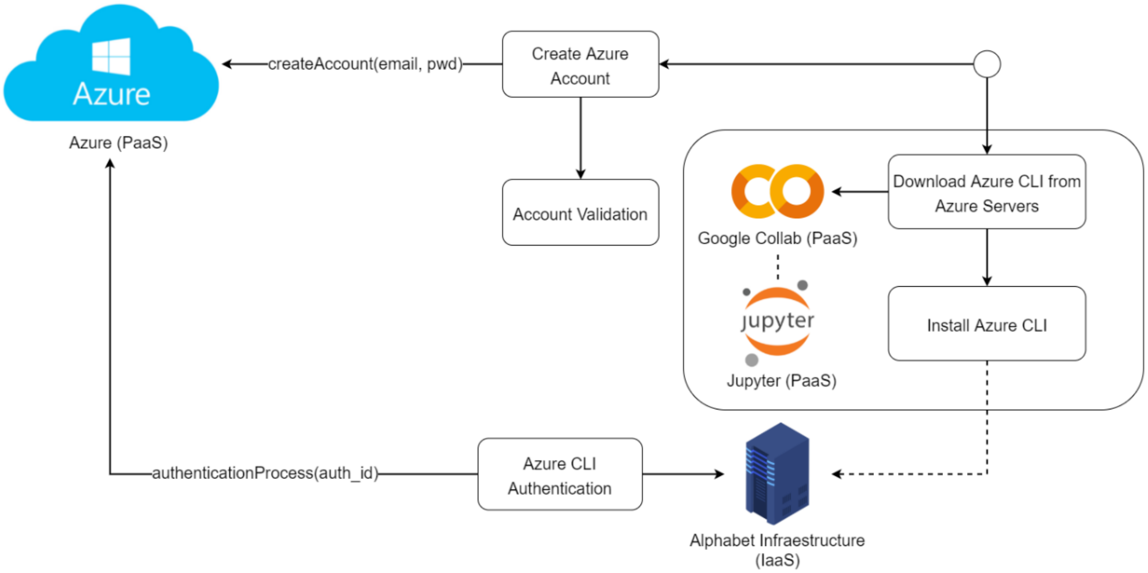

Figure 2: Azure Workflow Diagram.

Ngrok Workflow

For the ngrok workflow, a process very similar to the previous one had to be carried out, an account had to be created, a token had to be generated, the binary had to be downloaded and an authentication had to be performed to use the services of this platform.

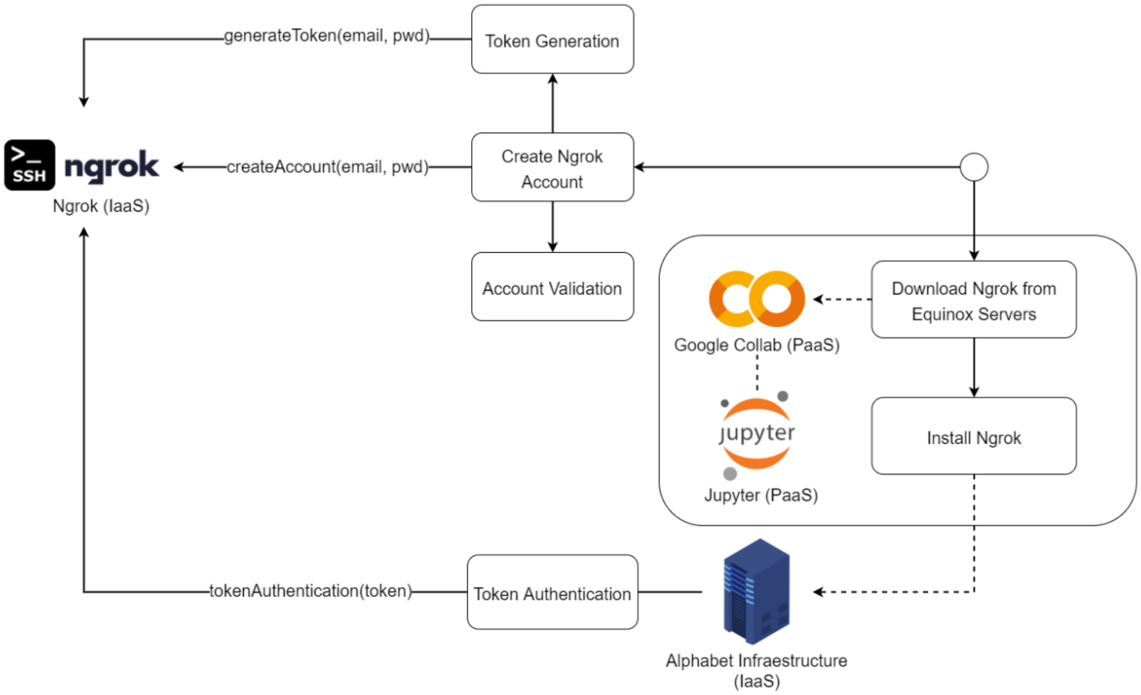

Figure 3: Ngrok Workflow Diagram

Ngrok Deployment

To perform a simulated deployment with ngrok, it was required to install some additional dependencies to those of the github sample project, as well as the creation of a small script to automate the whole deployment.

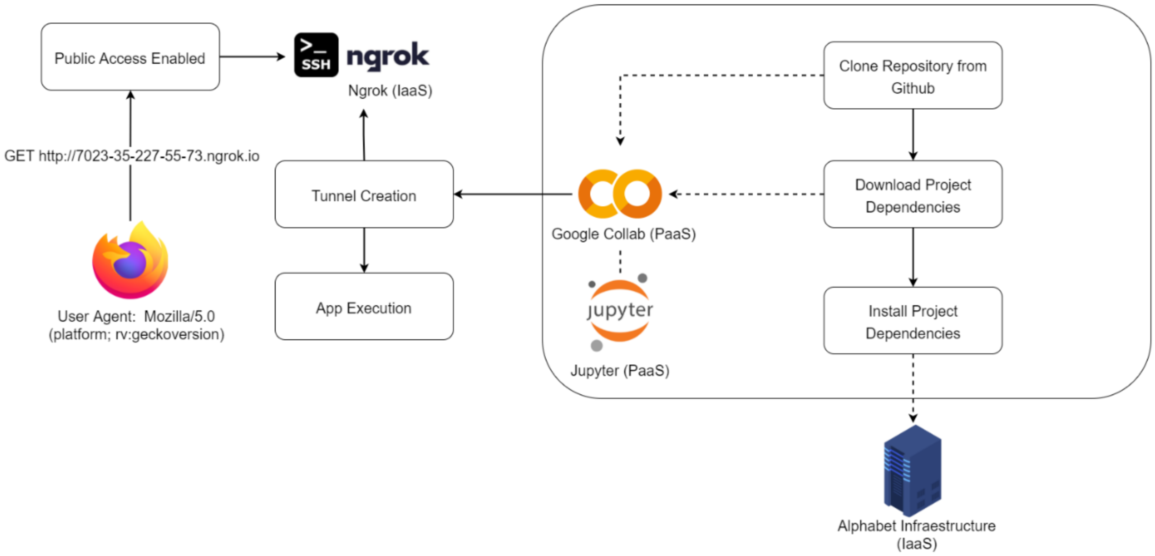
Public Access Enabled
SSH ngrok
Ngrok (IaaS)
GET http://7023-35-227-55-73.ngrok.io
User Agent: Mozilla/5.0
(platform; rv:geckoversion)
Tunnel Creation
App Execution
Google Collab (PaaS)
jupyter
Jupyter (PaaS)
Clone Repository from Github
Download Project Dependencies
Install Project Dependencies
Alphabet Infraestructure (IaaS)

Figure 4: Ngrok Deployment Diagram.

Azure Deployment

Finally, in order to perform a deployment in the Azure service, some functionalities of the Azure CLI application were performed, as well as the configuration of a custom domain.

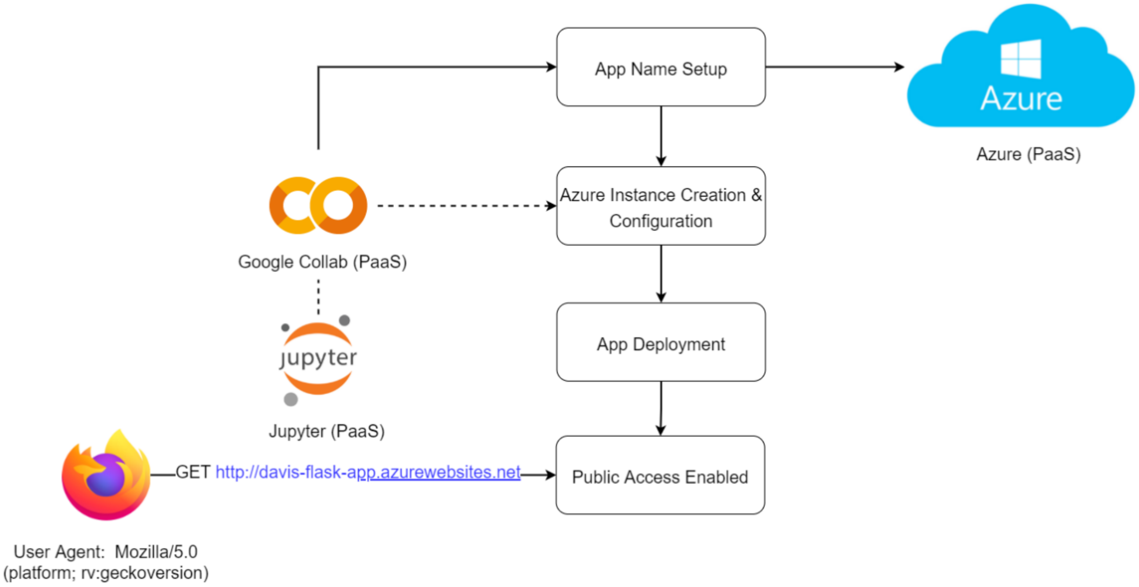
App Name Setup
Azure
Azure (PaaS)
Azure Instance Creation & Configuration
Google Collab (PaaS)
App Deployment
jupyter
Jupyter (PaaS)
GET http://davis-flask-app.azurewebsites.net
Public Access Enabled
User Agent: Mozilla/5.0 (platform; rv:geckoversion)

Figure 5: Azure Deployment Diagram.

Tutorial Comparation: Azure vs Our Class

The main differences between the tutorial offered by Azure and the one seen in this lab are the following:

Work Environment

For this case, although an azure account is required, the working environment is somewhat different since it is assumed that the trainee will be working in a local environment, whose Python version must be 3.9 or higher and also a virtual environment will be used (in order to avoid working directly with the Python interpreter, libraries and scripts installed on the computer).

If this is compared with what was seen in class, we will realize that the work environment was located in a service provided by google, powered by jupyter and linux, in my opinion it did not require the use of a python virtual environment because the jupyter notebook is highly configurable.

First Execution

The first execution of the application occurs inside the equipment, so access to it can only be done locally, This contrasts with the first execution seen in class since it is done through a tunnel generated by ngrok, which allows direct internet access to the application executed in the jupyter notebook.

Web App Creation

With regard to the creation of the application when following the Azure tutorial, it is very variant, since it recommends different ways to do it (from the Azure portal, VS code and Azure CLI), comparing it with what was done in class, it is similar to the fact of using Azure CLI since it was required to perform the installation along with the subsequent authentication of the service.

Deployment

Finally, the deployment has a range of possibilities similar to the previous step, here one of the most interesting examples is the use of github since it allows to have the CD/CI methodology, something that was not seen as such in the class, however, some previous confirmations must be made so that this can work correctly.

Now, returning to the laboratory practice, the deployment occurs quite easily thanks to the use of Azure CLI, with a couple of commands, our application will be working properly.

Glosary

Ngrok

Ngrok is an IaaS-type service that allows us to expose to the Internet a dynamically generated URL, which points to a web service running in a local environment.

Flask

Flask is a minimalistic web development framework powered by the Python programming language that allows the creation of web applications quickly and with a minimum number of lines of code.

CLI

The command line interface (CLI) is a text interface, which is accessed by commands in prompts, instead of using the mouse through the graphical user interface (GUI). CLI is also referred to as the Windows command prompt, or terminal/command screen on macOS.

Notebook

A notebook interface (also called a 'computational notebook') is a virtual notebook environment used for literary programming. It combines the functionality of word processing software with both the shell and kernel of that notebook's programming language.

Goolge Colaboratory

Google Colaboratory is a tool that allows you to run Python scripts through Google's servers. This allows you to run cells of code as if it were a Jupyter Notebook.

Azure

Microsoft Azure is a cloud computing service created by Microsoft to build, test, deploy and manage applications and services using its data centres.

SaaS

It is a software distribution model where the software and the data it handles are hosted on servers of an information and communication technology company, accessed via the Internet from a client.

PaaS

It is a set of cloud-based services that enable business users and developers to create applications quickly and cost-effectively.

IaaS

Infrastructure-as-a-Service (IaaS) refers to online services that provide high-level APIs used to index low-level infrastructure details such as physical computing resources, location, data partitioning, scaling, security, backup etc.

Git

Git is a mature and actively maintained open source project originally developed by Linus Torvalds, the famous creator of the Linux operating system kernel, in 2005. A staggering number of software projects rely on Git for version control, including commercial and open source projects.

Github

Github is a portal created to host the code of any developer's applications. It is mainly used for the creation of source code for computer programs. The software that runs GitHub was written in Ruby on Rails.

Proxy

A proxy, or proxy server, in a computer network, is a server -programme or device-, which acts as an intermediary in the resource requests made by a client to another server.

Reverse Proxy

In computer networks, a reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers. These resources are returned to the client as if they originated from the Web server itself.

Tunneling

Tunnelling is a communication protocol that allows the movement of data from one network to another. It involves specific steps that allow communications from the private network to be sent over a public network, this process is called encapsulation.

Virtual Environment

A virtual environment is a Python environment in which the Python interpreter, libraries and scripts installed in it are isolated from those installed in other virtual environments, and (by default) any libraries installed in a Python "system", i.e. one that is installed as part of your operating system.

References

Microsoft. (n/d). Quickstart: Deploy a Python (Django or Flask) web app to Azure App Service. Retrieved March 14, 2022 from: https://docs.microsoft.com/en-us/azure/app-service/quickstart-python?tabs=bash&pivots=python-framework-flask