Hosting your Web Application on Azure

Azure Machine Learning provides a powerful platform for running and deploying large language models (LLMs). This guide will walk you through the steps to deploy your LLM and create an endpoint for your LLM in Azure Machine Learning.

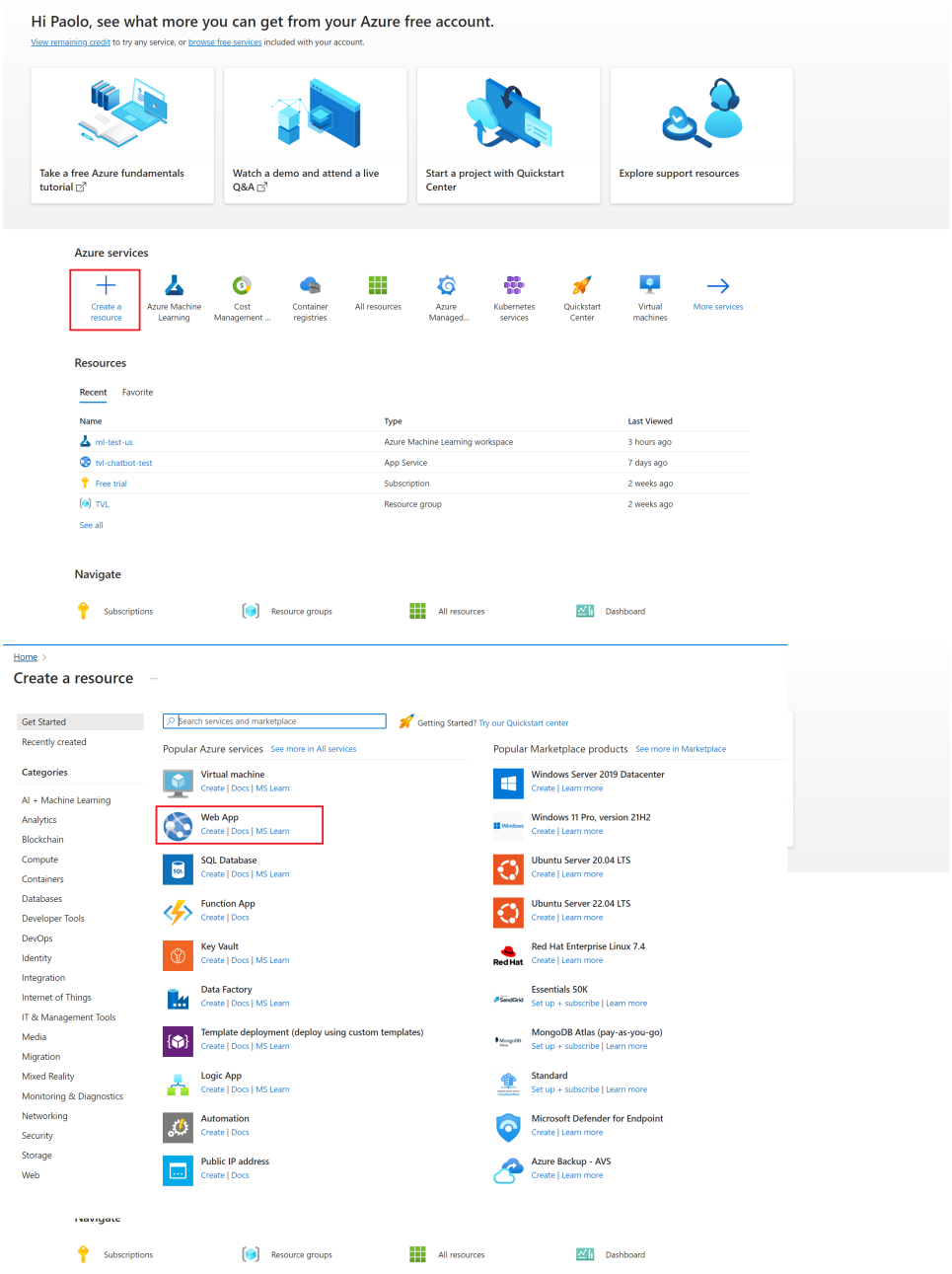
Prerequisites

- An active Azure account - Setting Up a free Azure Account - NLP Community - Confluence (iabg.de)
- An API-accessible LLM, either from closed-source providers (GPT, Claude, Mistral Large etc.) OR a self-deployed LLM - Deploying Large Language Model (LLM) on Azure

Steps to Host your Web Application

- **Sign in to Azure Portal:**
    - Open your web browser and navigate to the Azure Portal: https://portal.azure.com.
    - Sign in with your Azure account credentials

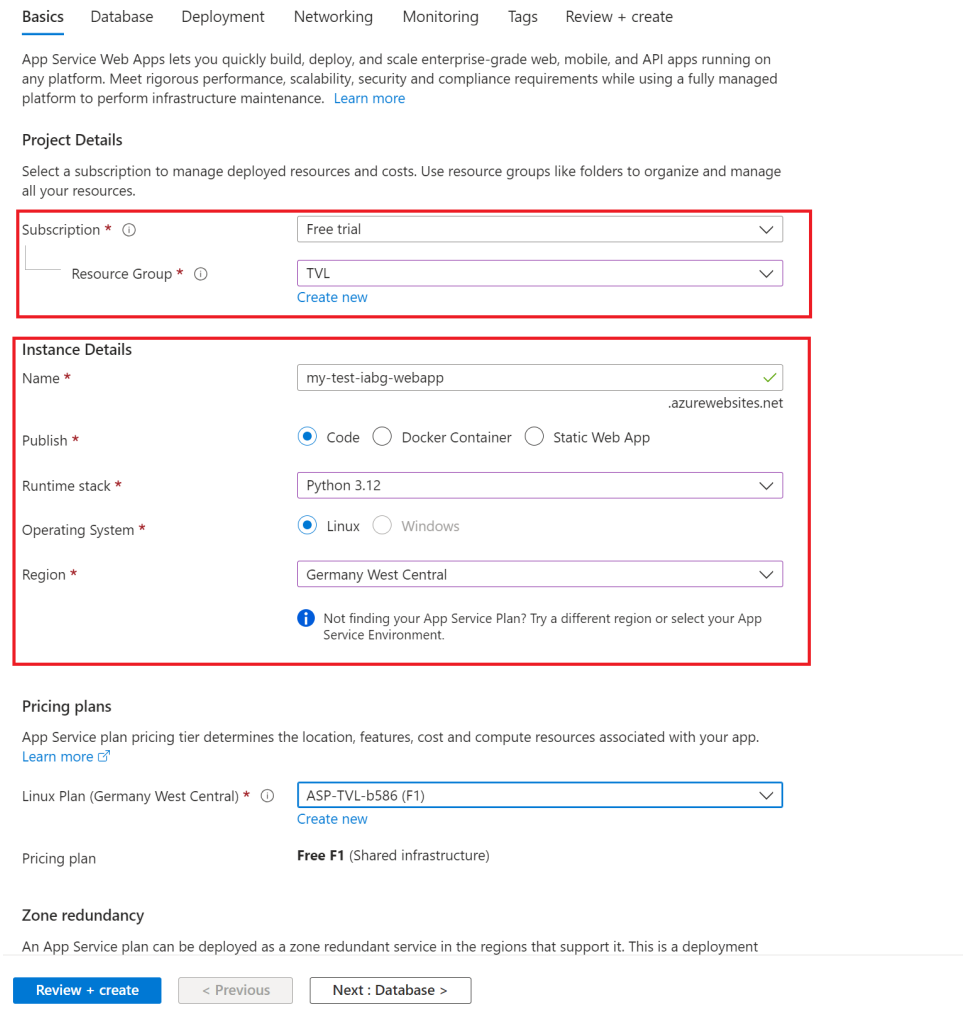
- **From the home page, select "Create a resource" and then "Web App":**



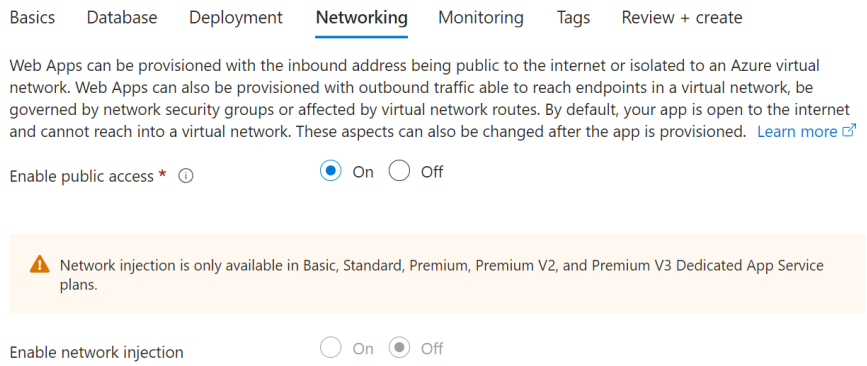
- **Fill in the initial configuration for your Web App:**

For this tutorial, we are deploying a Python web-based application directly from a code repository. Other options (e.g. Docker deployment, other programming languages) can also be specified here.

Under "Basics", select your free trial subscription and a resource group (if you have created one). Specify a name for your web app (which will also be used as the final URL), select "Code" as publish mode, your prefered Python version and Region (e.g. "Germany West Central").

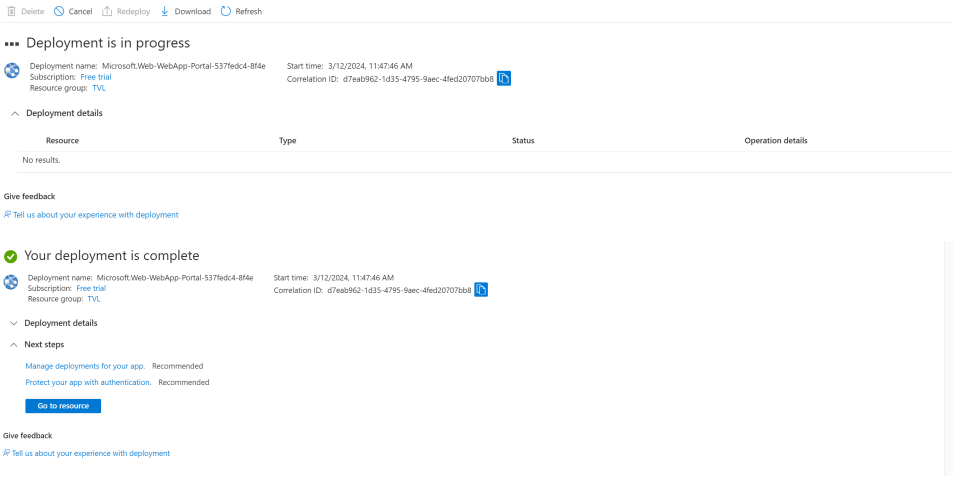


You can also customize other aspects of your applications in the other tabs. For example, if you want your app to be publicly accessible, ensure that "Enable public access" is turned on under "Networking":



In general, the default options for the advanced tabs are sufficient.
When you are done, click on "Review + create" in the lower part of the screen.

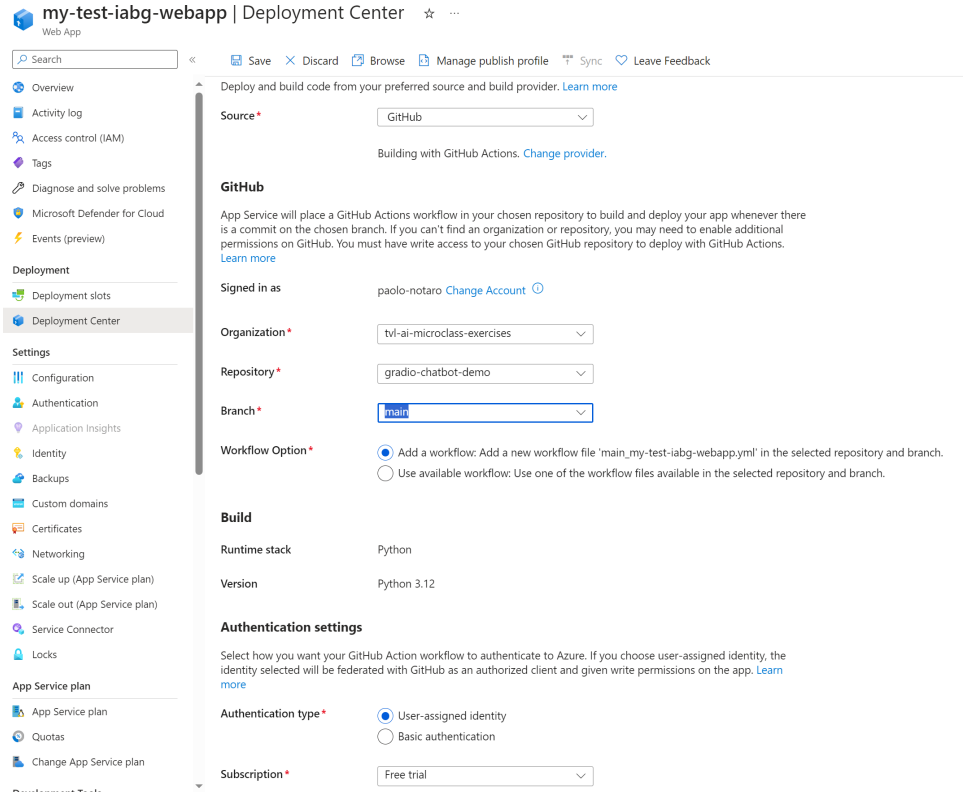
- **Wait for the App to be deployed**



when deployment is complete, click on "Go to resource". This page gives you an overview on everything related to your current WebApp deployment (usage, logs, monitoring, accessibility, etc..)

- **Connect a repository to your Web App**

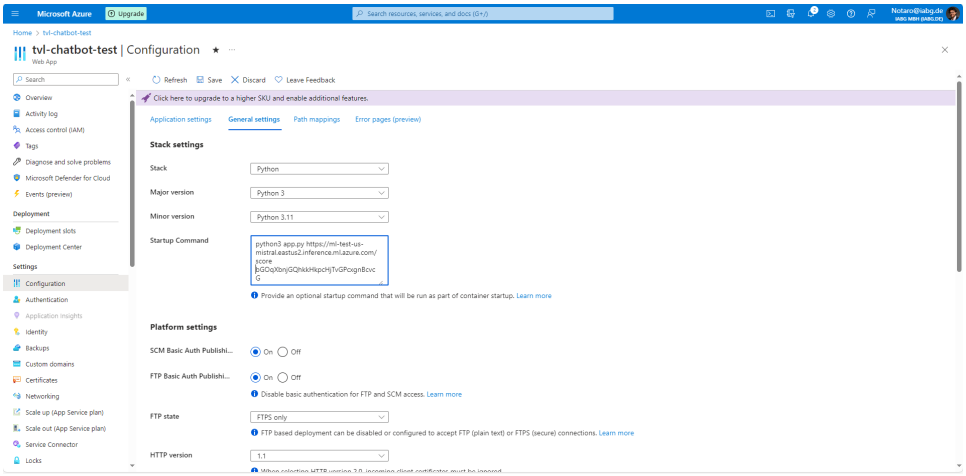
  Under "Deployment Center" in the left pane, you can specify the source of your app code. Fill in the Source provider (in our example "GitHub"), repository name, branch and other details.
  This code repository will be copied in the app container. You can access it later on via SSH (see Debugging).



- **Set up an entrypoint for your Web App**

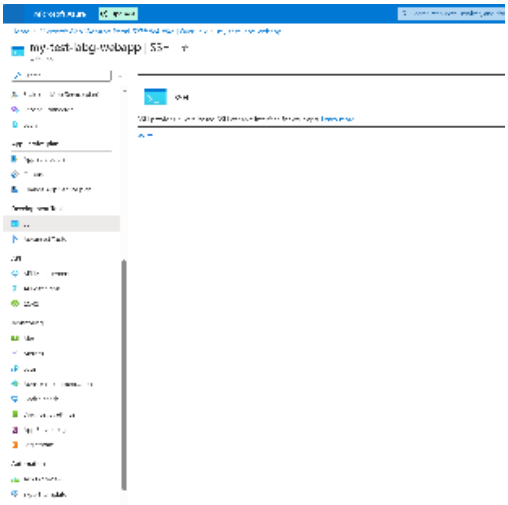
  Under "Settings" in the left pane, select the "General Settings" tab. Then, specify your entrypoint options.

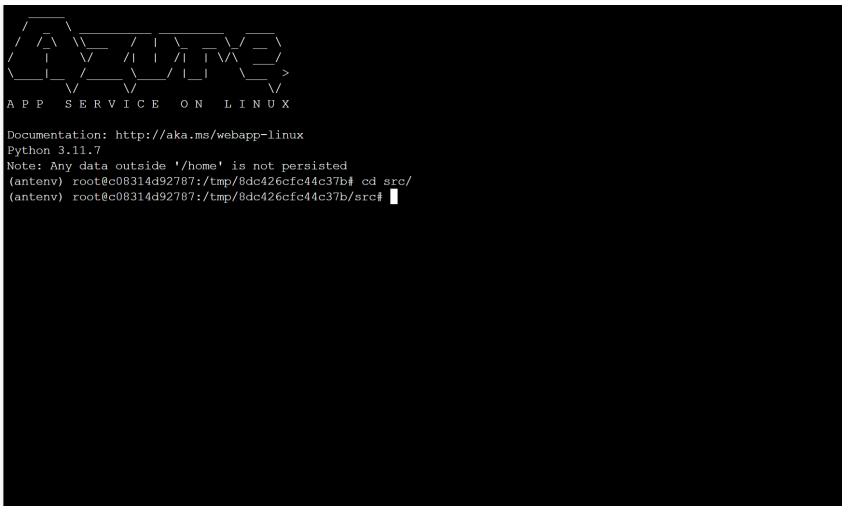
In the example below, we specified a Python start-up script with CLI arguments (LLM API endpoint URL, LLM API key).



Debugging your Web App

You can use the SSH connection option from the left pane to connect inside your app container. There you have all your application code and computing resources. You can also set up a debugger via SSH tunnel (e.g. in VSCode) or try running tests directly.




A P P S E R V I C E O N L I N U X
Documentation: http://aka.ms/webapp-linux
Python 3.11.7
Note: Any data outside '/home' is not persisted
(antenv) root@c08314d92787:/tmp/8dc426cfc44c37b# cd src/
(antenv) root@c08314d92787:/tmp/8dc426cfc44c37b/src#