3.2 - GitHub Actions: Azure

Introduction

A final thing we can explore in GitHub Actions is performing commands on Azure environments. Azure has some tools on its own that can be used for this, but configuring it in GitHub Actions gives us some flexibility in connecting it in tools we are used to.

Objective

Setting up GitHub Actions and Azure will allow us to work with the Azure CLI in an automated environment.

We will set up the Azure CLI to connect to our Azure Subscription in the GitHub Repository and the Actions.

Knowledge

- Azure CLI commands
- GitHub Actions YAML files

Skills

- Working with Azure CLI
- Setting up GitHub Actions for automated workflows

Necessities

- A terminal
- A Git Repository on GitHub
- An IDE such as Visual Studio Code

Configure Azure Machine Learning Access

This chapter will explore the way to set up a GitHub Actions pipeline to connect to our Azure Machine Learning workspace using the Python SDK 2.0 and the Azure CLI.

Setting up Azure

The first thing we'll be doing is configure something on our Azure portal. Like we said, our pipelines will be running automatically, which also means that we have no manual way to log in to Azure.

In order to do that, Azure uses **Service Principals (SP)** which we can create on our Azure Portal, and through the Azure CLI.

This Service Principle will be used to authorise our CI/CD pipeline to work with our Azure account, and create or use Azure resources and services on our behalf. Think of the Service Principle as a separate account that we introduce in Azure. It has it's separate rights and access methods.

We have a few steps to perform in the Azure Portal to do this. We will explain the steps for the Azure CLI as well, that way there is a choice of what to use.

🚨 In an Azure for Students subscription, there is no way to use the Azure Portal to create this App Registration. Use the Azure CLI for the Azure for Students setup.

Azure Portal

- Create a new App Registration and call it `MLOps-SP-NS`. Where `NS` stands for the initials of **N**athan **S**egers. If there is an error during creation, the App Registration might already be in use, then create a new name.
- Go to the Resource Group used in the previous assignment and navigate to IAM (Access Control).
- Add a new Role Assignment.
- Choose the Contributor role. This allows your MLOps pipeline to access any resource into your account. **NOTE: If you want to customise the access, you can search for the different resources that you should allow. But as this is just a demonstration, we will not dive deeper into all the different resource roles.**
- Add the `MLOps-SP-NS` App Registration as a member. This is called a Service Principal.

As you are not using your own Azure account when you use the Service Principal, this does not compromise your passwords and is thus very secure. But don't forget you've given all the rights to create and delete Azure resources into your resource group.

Azure CLI

If you do not want to perform these actions in the Azure Portal, and just want to execute one command in the Azure CLI, you could follow these instructions.

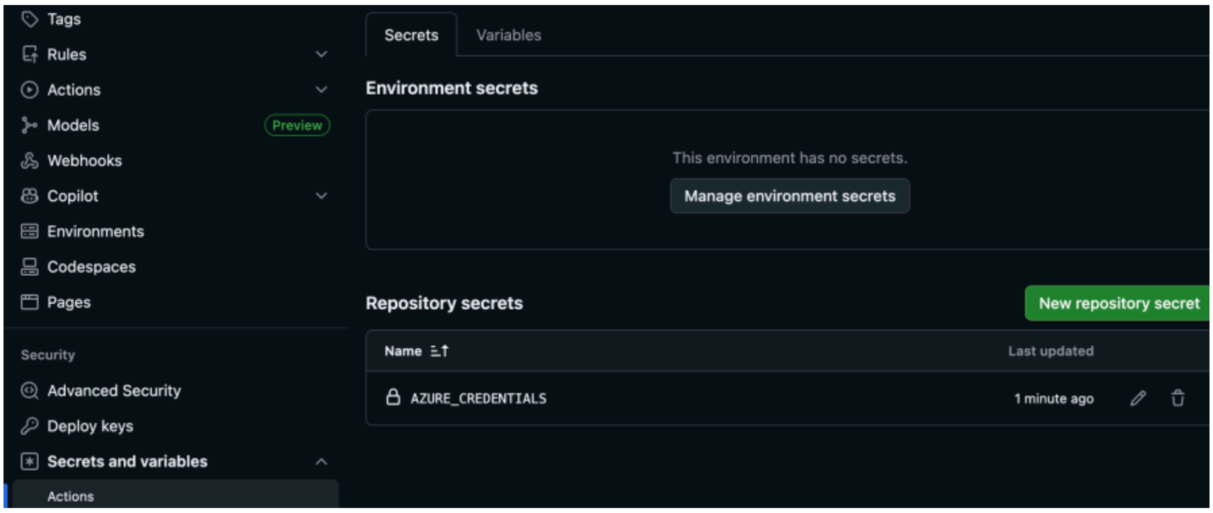
- Open up a terminal which has the Azure CLI connected using `az login`
- Enter the following command which will give you a result like below.
  `az ad sp create-for-rbac -n "MLOps-SP-NS" --role Contributor --scopes /subscriptions/<subID>/resourceGroups/<resourceGroupName> --json-auth`
  Fill in the resourceGroup name, which should be something like `mlops` like we used in the last assignment.
  The SubscriptionID can be found on the Azure Machine Learning Studio Portal.
  'MLOps-SP-**NS**' has `NS` which stands for my initials. If you get an error during creation, the App Registration might already be in use, then you should create a new name. `SP` stands for Service Principal.

As a result we get this:

```
{
  "clientId": "******",
  "clientSecret": "******",
  "subscriptionId": "******",
  "tenantId": "******",
  "activeDirectoryEndpointUrl": "https://login.microsoftonline.com",
  "resourceManagerEndpointUrl": "https://management.azure.com/",
  "activeDirectoryGraphResourceId": "https://graph.windows.net/",
  "sqlManagementEndpointUrl": "https://management.core.windows.net:84
43/",
  "galleryEndpointUrl": "https://gallery.azure.com/",
  "managementEndpointUrl": "https://management.core.windows.net/"
}
```

Adding the Service Principle to GitHub

To use this Service Principal in GitHub, we need to provide a secret in GitHub, with this as a content. Paste this content in your GitHub Repository settings in the **Security > Secrets and variables > actions** part. Name it `AZURE_CREDENTIALS` and paste the content from the previous result.



Environment secrets

This environment has no secrets.

Repository secrets

Name	Last updated
AZURE_CREDENTIALS	1 minute ago

Using the Azure CLI in the GitHub Actions

In order to use the Azure CLI with our Machine Learning SDK, we need to make sure the SDK gets installed in **every** job step.

In this example step, we can try to get all the computes we currently have. This is just a demonstration that we can remove later.

```
    - name: Azure test - Get Compute
      uses: Azure/CLI@v2.1.0
      with:
        azcliversion: 2.64.0
        inlineScript: |
          az extension add --name ml
          az configure --defaults group=$GROUP workspace=$WORKSPACE
location=$LOCATION
          az ml compute list
```

❓ **QUESTION 1 - Check latest versions**
What is currently the latest version of the `azcli` and the `Azure/CLI` Action? Where did you find it?

💻 **ANSWER 1 - Check latest versions**

- **azcli (Azure CLI – lokaal geïnstalleerde tool)**
- **Huidige versie:** `2.78.0`
- **Bron:**
  - Microsoft Learn – officiële install-pagina
  - Officiële *release notes*
- **Laatste release:** 14 oktober 2025

Azure/CLI (GitHub Action voor Azure CLI)

- **Huidige versie:** `v2.2.0`
- **Laatste release:** 22 september 2025
- **Bron:**
  - Officiële azure/cli Action repository → *Releases* sectie
  - Vermelding: **“GitHub Action for Azure CLI v2.2.0 — Latest”**

What did you learn?

Fill in something that you learned during this lesson

Ik heb geleerd hoe je een Service Principal gebruikt om veilig toegang te geven tot Azure resources vanuit CI/CD

Give three interesting exam questions

1. Welke informatie bevat een Azure Service Principal en waarvoor wordt deze gebruikt?
2. Hoe verbind je een GitHub Actions workflow veilig met Azure?
3. Waarom gebruik je het liefst een aparte Service Principal voor CI/CD i.p.v. je eigen gebruikersaccount?

Handing in this assignment

You can hand this in by duplicating this document on Notion, print this document as a `.pdf` and submit that document on Leho.

Also hand in the written Source Code in a `.zip` file please.

Checkboxes:

- [x] ~~I have duplicated this file~~
- [x] ~~I have filled in all the answers~~
- [x] ~~I have added something that I learned~~
- [x] ~~I have added three interesting exam questions~~
- [x] ~~I have zipped my project and uploaded it to Leho~~
- [x] ~~I turned off all the Azure services I don't need anymore, to save some costs.~~