# ICT & Infra S3 Automation & Orchestration, week 12

|  |  |
| --- | --- |
| Class: |  |
| Student number: |  |
| Student name: |  |

## Introduction

This week, you will create a CI/CD pipeline in GitLab.

This is an **individual** assignment.

### Assignment 1. Create Continuous Integration (CI) for your project.

### Difficulty: ★★★★★

In Semester 2, you made a Flask app for your project. For this assignment, we ask you to create a Continuous Integration process, which can be verified by an automated build and automated tests (which you must create). The tests must test the actual application.

Beware: Creating a runner in EC2 can be expensive. Better create it on the local machine or ECS.

Provide screenshots (evidence) for your solution. Always explain your proof! As a professor, we expect at least:

* Provide CI process code defined in .gitlab-ci.yml.yml.yml
* Provide the tests you implemented in your project
* Provide multiple screenshots of successfully executed (passed) jobs (from git.fhict.nl -> CI.CD -> Jobs -> passed)

|  |
| --- |
| *Solution:* |

### Assignment 2. Implement Continuous Deployment, which will complete the CI/CD pipeline

### Difficulty: ★★★★☆

To make your app accessible online, you must deploy it to a web server. You can choose to deploy the app on EC2 (easy) or an ECS (challenging) environment. Complete CI/CD pipeline, so it can deploy your app if the automated build and automated tests succeed.

Provide screenshots (evidence) for your solution. Always explain your evidence! As a professor, we expect at least:

* Provide a complete process defined in .gitlab-ci.yml..yml
* Provide multiple screenshots of successfully executed (passed) jobs (from git.fhict.nl -> CI.CD -> Jobs -> passed)

|  |
| --- |
| *Solution:* |
|  |

Afbeelding met tekst, schermopname, software, Multimediasoftware

Door AI gegenereerde inhoud is mogelijk onjuist.Afbeelding met tekst, schermopname, software

Door AI gegenereerde inhoud is mogelijk onjuist.  
I created a new folder in my repository called .github/workflows, in which I put this code in main.yml

This whas my old code that I created to deploy my website to nothing it had a fake deploy but it shows you how the testing of the code would work so I will quickly go over the code to explain it to you when github receive a push it gets triggered then it gets the code from the libraby this is done by actions this is a github tool that’s is made by github it can be used for various things here is a link to the github [actions](https://github.com/marketplace?type=actions) here is a marketplace where you can find all driffent actions the checkout v3 is a pretty commen one. setup-python@v3 I found by watching this [tutorial,](https://youtu.be/HbZdOZTNO0o?si=svt2dUs7kF5NhSXJ) setup python speaks for itself, it sets up a Python library. Then I install dependencies that you can see in requirements.txt. Then I do a pytest. This is done inside GitHub. I then run the Python script using pytest. This script waits for a response from different URLs, and if the right return code is received, it will be successful.   
-v is there so that it shows which test fails or passes, then the fake deployment starts only if the test has passed again, then on Ubuntu, but it's fake, so nothing happens, it's just rerun a oke and the pipeline gives a success mark.

Sources used

<https://github.com/marketplace?type=actions>

<https://youtu.be/HbZdOZTNO0o?si=l3ygUNEB30dGwwKa>

<https://docs.github.com/en/actions/writing-workflows/workflow-syntax-for-github-actions>

Afbeelding met tekst, schermopname, Lettertype, nummer

Door AI gegenereerde inhoud is mogelijk onjuist.

Afbeelding met tekst, schermopname, software, Multimediasoftware

Door AI gegenereerde inhoud is mogelijk onjuist.

Here you can see my CI/CD pipeline working, and also the test being successful.

Github also has a workflow shop where you can find workflows for your specific needs I found one that deploys ECS services so I used that one and followed it step to deploy my website inside ECS here is the [link](https://github.com/marketplace/actions/amazon-ecs-deploy-task-definition-action-for-github-actions) to that workflow I did have to add somethings in AWS and my current workflow to make this work. I started with AWS, and I created an ECR repository called my-flask-router-app. This was needed because the workflow was storing the Docker image there. Then I did a basic fargate ECS setup that we learned in our previous lesson, or in my case, in the case study project. Called flask-router-cluster. Then created a service called. Flask-router-service. Then I put all these names in the right places inside the workflow that you can see below. If you follow this [link,](https://youtu.be/ZfCJvVWfY78) there is a YouTube video giving a small demo. I also had to add my access key to GitHub so that it could make changes inside AWS.  
 Afbeelding met tekst, software, Lettertype, schermopname

Door AI gegenereerde inhoud is mogelijk onjuist.  
Afbeelding met tekst, schermopname, menu, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

I also had to add a docker file and a task definition file that the workflow could upload to AWS I asked ai to create the docker and workflow for my and then gave it the workflow it gave these codes ass responses.

Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.  
Afbeelding met tekst, schermopname

Door AI gegenereerde inhoud is mogelijk onjuist.

So what's different here than from the first part? Firstly, the fake test got deleted, and I edited the found workflow to work in my workflow. So what I did was delete to old deploy, made a new one that can only start if the test is completed, and then it executes the filled-in workflow. I learned to find working solutions that were prefabed and I could use in my workflow, and for the beginning part, I learned how to set up a basic test and deployment workflow inside GitHub using GitHub Actions

sources I used for learning.

<https://github.com/marketplace?type=actions>

[and https://youtu.be/HbZdOZTNO0o?si=l3ygUNEB30dGwwKa](https://youtu.be/HbZdOZTNO0o?si=l3ygUNEB30dGwwKa)

<https://docs.github.com/en/actions/writing-workflows/workflow-syntax-for-github-actions>

<https://github.com/marketplace/actions/amazon-ecs-deploy-task-definition-action-for-github-actions>

repo link <https://github.com/joepemons/personal-project.git>, also below you can see the original Amazon ECS deploy workflow

Afbeelding met tekst, schermopname, software, ontwerp

Door AI gegenereerde inhoud is mogelijk onjuist.  
Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.  
Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.