### **Laboratory 4 Work Report**

### **Objective: Familiarize with the process of creating CI/CD pipelines**

### **Tasks:**

1. Study the Google Cloud services available for creating CI/CD pipelines.
2. Implement a CI/CD pipeline on the Google Cloud platform to deploy our application on Cloud Run.

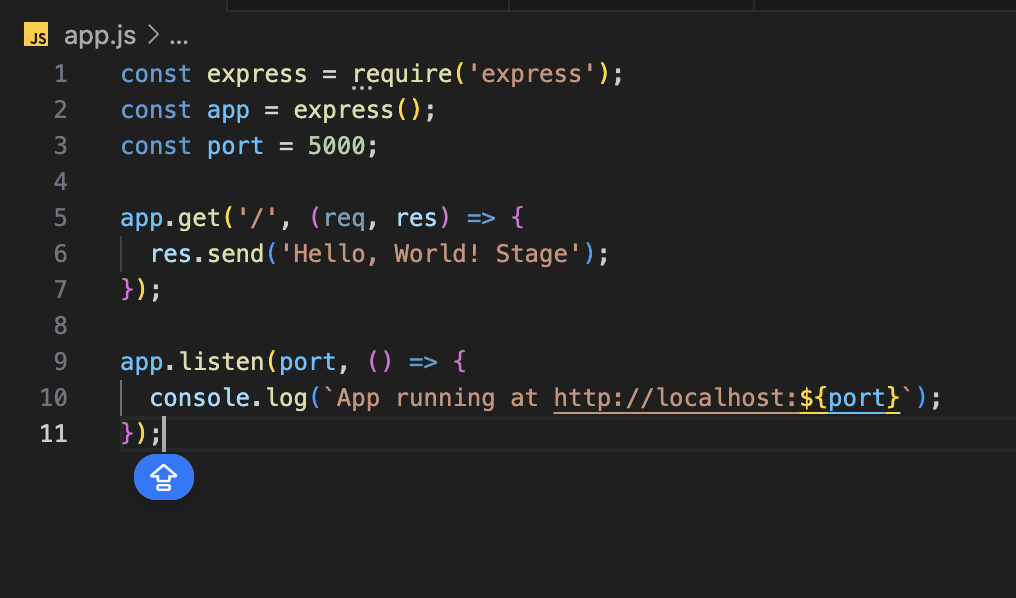
### **Description:**

In this lab work, we will use several Google Cloud services to implement a CI/CD process. We will use Google Cloud Build to create and deploy Docker images, Google Cloud Run to host and run containerized applications, and Google Artifact Registry to store Docker images. Below is a detailed description of the entire process.

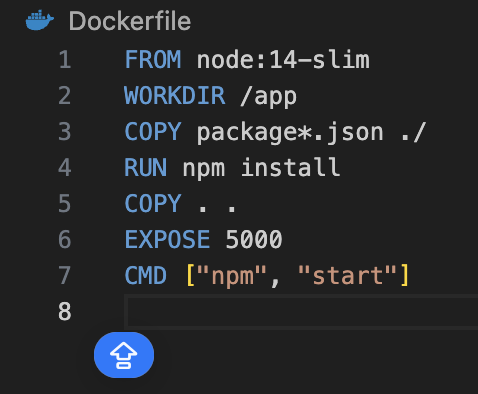
### **Step 1: Creating and Configuring the Application**

First, we'll create a simple web application using Node.js and Express. The application will respond to requests with the message "Hello, World! Stage".

##### **Application Source Code (app.js):**



##### **Dockerfile:**

To containerize our application, we'll create a Dockerfile:

### **Step 2: Creating Repositories in Artifact Registry**

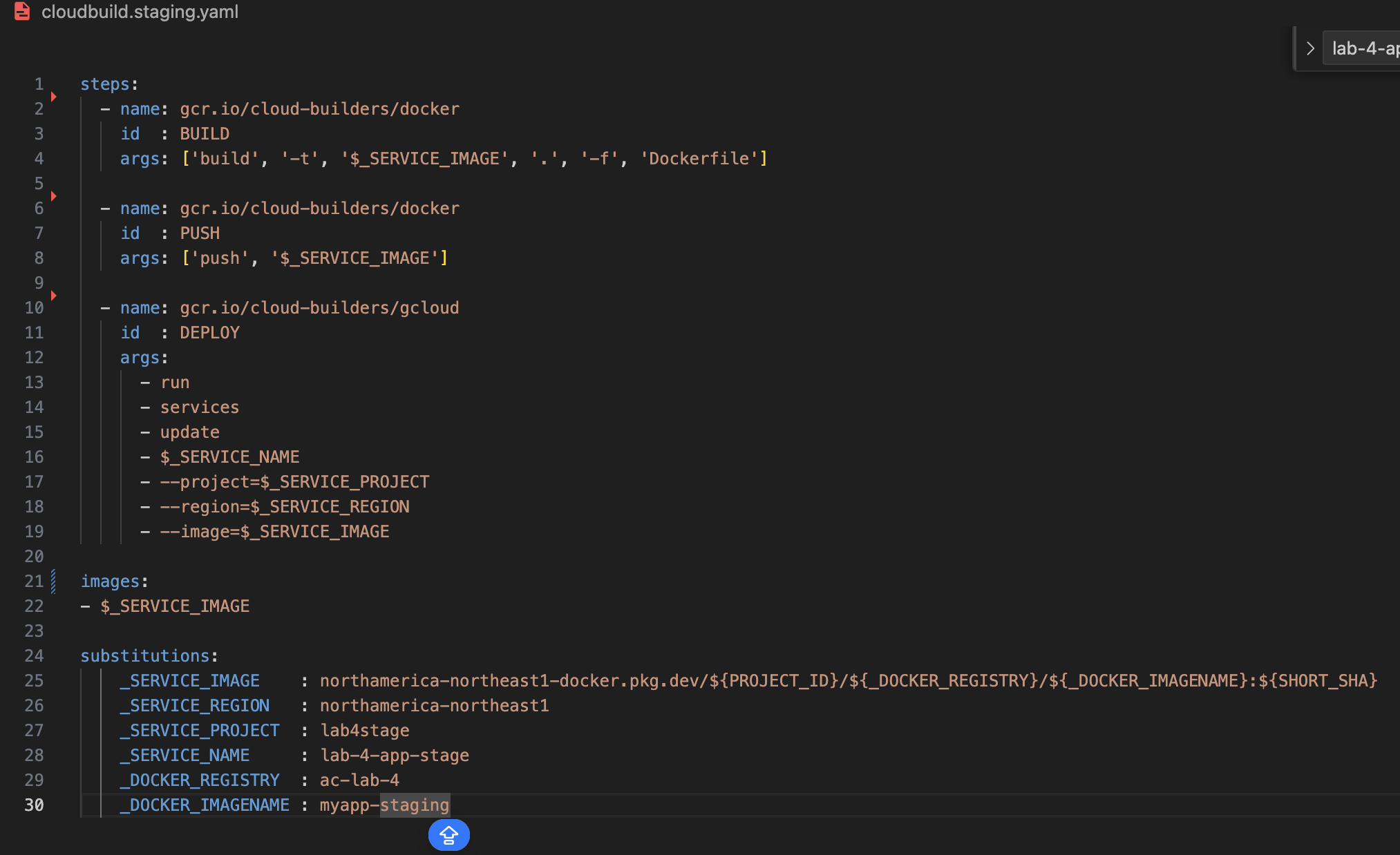
To store Docker images, we'll create repositories in Google Artifact Registry. One repository will be used for the production environment, and the other for the staging environment.

* Repository for production: lab-4-app-prod
* Repository for staging: lab-4-app-stage

### **Step 3: Configuring CI/CD Pipelines**

Now we'll set up automation for the build, test, and deployment processes of our application using Google Cloud Build. We'll create two configuration files for Cloud Build: cloudbuild.prod.yaml and cloudbuild.staging.yaml.

##### **Configuration File cloudbuild.prod.yaml:**



##### **Configuration File cloudbuild.staging.yaml:**

### **Step 4: Setting Up Automatic Pipeline Triggers in Google Cloud Console**

To automate the CI/CD pipelines, follow these steps:

1. Go to the [Google Cloud Console](https://console.cloud.google.com/).
2. Navigate to Cloud Build.
3. Click on Triggers in the left menu.
4. Click Create Trigger.
5. Fill in the fields as follows:
   * Name: Deploy to Prod
   * Event: Choose "Push to a branch"
   * Source: Select your repository
   * Branch: main (or another branch you use for production)
   * Configuration: Cloud Build configuration file
   * Build configuration: Select "cloudbuild.prod.yaml"
6. Repeat steps 3-5 to create a trigger for the staging environment:
   * Name: Deploy to Staging
   * Event: Choose "Push to a branch"
   * Source: Select your repository
   * Branch: staging (or another branch you use for staging)
   * Configuration: Cloud Build configuration file
   * Build configuration: Select "cloudbuild.staging.yaml"

### **Results**

Our applications are successfully deployed and available at the following URLs:

* Production:<https://lab-4-app-prod-rgdakfwcfq-nn.a.run.app/>
* Staging:<https://lab-4-app-stage-nyzpvjvvxq-nn.a.run.app/>

### **Conclusion**

During this laboratory work, we studied the Google Cloud services available for creating CI/CD pipelines. We successfully implemented a CI/CD process for our application, including automatic building, uploading, and deploying Docker images to Cloud Run. This process significantly simplifies and automates the management and deployment of applications in the cloud.

### **Links**

* [GitHub Repository](https://github.com/Evgheni-Cernev/ac-lab-4-app)
* Production:<https://lab-4-app-prod-rgdakfwcfq-nn.a.run.app/>
* Staging:<https://lab-4-app-stage-nyzpvjvvxq-nn.a.run.app/>