Code.Hub

REGENERATION®

Project Assignment

ReGeneration Academy on CyberSecurity & DevOps Engineering

April 2023

Abstract

This is the description of the group project assignment for the ReGeneration Academy on CyberSecurity & DevOps Engineering. DevOps principles and concepts will be used to define the infrastructure of a typical cloud deployment. The project defines requirements for continuous pipelines with the use of containers and application of relevant continuous integration practices. The appropriate security measures and approaches must also be applied where applicable.

Introduction

DevsApps Corp. is a technical company based in Athens, Greece and provides services with CI/CD implementation. The company has contracted you and your team to create a DevOps pipeline for a given Web Application and its Database.

Sample application

The application has two basic screens, as it can be seen from Figure 1 and Figure 2

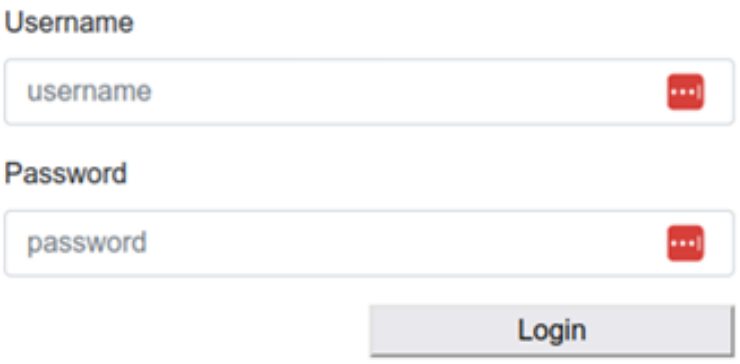

Figure 1. The login form of the demo application.

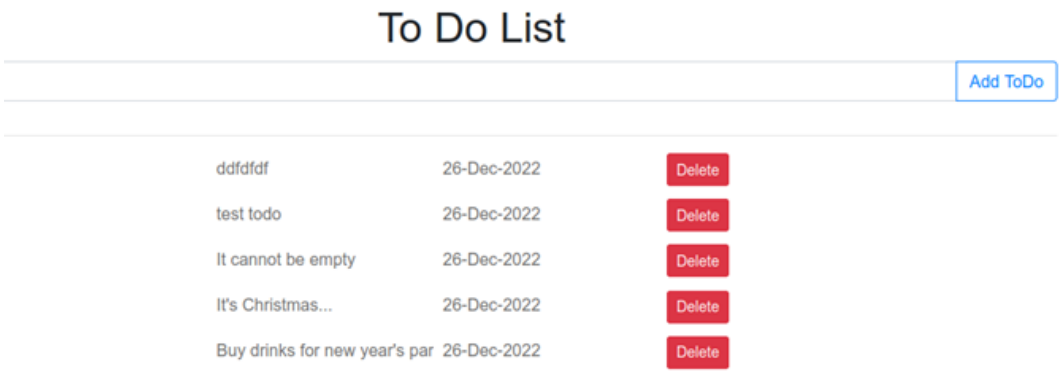

Figure 2. The main image of the demo application.

Accounts and Tools

Each team will need a GitHub account for the version control of the code and an Azure account for the deployment of the application on the cloud. The following tools are also needed: Terraform, Jenkins, Docker and Ansible.

The web application is implemented in Java and the build automation tool which is suggested to be used is Maven. For local development, you can use IntelliJ to run the project.

Project preparation

Perform the following DevOps operations:

Step 1: Create an Ubuntu VM with Terraform on Azure. This is going to be the CI Server. This step needs to only run once.

1. Install Jenkins on this VM and any tool required to complete the next steps.
2. All the following steps that are controlled by Jenkins are to be run on this VM as a pipeline.

Step 2: Setup the Java source code of the application within your GitHub account.

1. The GitHub repository of the web application is the following: https://github.com/codehub-learn/toDoAppWithLogin-Regeneration-CyberSecurity-DevOps
2. Fork this repository into a GitHub account of your own.

Step 3: Create a JAR file from the Java source code.

1. Configure Jenkins to generate a JAR file (using Maven) for the Java project on manual demand.

Step 4: Build a Docker image from the JAR file

1. Configure Jenkins to generate a Docker image from the JAR. The Dockerfile has been provided to you and is located within the root directory of the GitHub repository.
2. Upload the docker image on Docker Hub, as a public image.

Step 5: Use Terraform to create a virtual machine on Azure through Jenkins. This is going to be one of the node machines.

1. Configure Jenkins to trigger Terraform to build a VM on Azure where the docker containers should be hosted. The virtual machine should be publicly accessible on port 8080.
2. Configure Jenkins to deploy new infrastructure changes if the Terraform code is updated.

Step 6: Configure Jenkins to trigger Ansible to execute the following configurations to the node:

1. Install Docker, download the appropriate images (Web Application & MySQL Database) and start the containers.
2. Create a Docker network to connect the MySQL container with the Application container.
3. When running, ensure that MySQL starts first so that the Application can see it when it starts. The MySQL server version should be 5.
4. For the Web Application, the following environment variables can be set:
    a. **DB_HOST**, default value is localhost,
    b. **DB_PORT**, default value is 3306,
    c. **DB_USER**, default value is root,
    d. **DB_USER_PASSWORD**, default value is root.
5. The application process listens to port 8080, therefore the internal container port should be set to 8080.
6. If everything is running, check from a browser that the application has been deployed on Azure and is accessible. To login into the application, use the following credentials:
    a. Username: **admin,**
    b. Password: **admin**

Step 7: Setup Jenkins so that when a change takes place in your Java code on GitHub, then the new version of the application is automatically deployed on Azure and is reflected in the production instance. In order to test this functionality, add a comment within any of the Java files. You can add comments by typing "***//***" followed by your comment, i.e. "***// This is a comment.***".

Extras

This section includes any additional interesting functionality you may add to the overall project. While the steps under the project preparation section must be completed by all teams, the requirements described within this section are optional, and you can do them only if you have sufficient time. You can choose any of the extra functionality in any order to implement.

- Enable the MySQL database to have at least TLS version 1.2.
- Setup the application to be accessed through a domain name instead of an IP. You are not expected to buy a domain but to enable the respective Azure functionality.
- Setup the application to be accessed through HTTPS and not through HTTP. When the user attempts to access through HTTP, the request should be redirected through the HTTPS port. You are not expected to buy a certificate. You can generate your own certificate and use that instead (at least for the current version of the application).
- Setup an Azure Key Vault service which generates/stores the database username and password. Before it is run, the application should be "fed" the credentials from the vault.
- The database in the proposed pipeline is used as a container installation. Replace the container with a MySQL Service provided by Azure. The service may also be in the same resource group as the application.
- The virtual machines provisioned through Terraform should have password authentication disabled and should only be accessible through an SSH key.

Deliverables

A. Perform the operations described in the project preparation and extras section and document all the steps with screenshots. Submit this deliverable along with any comments you may have.
B. State your recommendations to enhance the security of the above pipeline. Furthermore, discuss the security options for when the data of the deployed application is in motion and when it is at rest. The expected file of this deliverable should be called "security_considerations" and could either be a text file or a Word document. Also consider adding these considerations to the next step which is the presentation.
C. Create a PowerPoint presentation summarising your work, giving highlights of the purpose of the project, the decisions made and their justifications, as well as points of future improvement or expansion. This presentation will be used during the final session of the academy.

File submission

All submitted material must be in English.

To submit your team files, you need to do the following:

1. Prepare all the deliverables and upload them in your team's channel.

2. Include a text file named ‘github_link.txt’ which contains the link to your GitHub project. The file may contain more than one GitHub link, depending on how you have set up the overall pipeline and project.
3. Ensure that the GitHub repository link is accessible by external people.

The deadline for the submission is **27th April**.