CPSY 300 Operating Systems

Docker Project: Part 1 out of 4 Parts

Weight: 12.5% Marks: /100

Student Name:

Student ID:

Date:

This page was intentionally left blank.

Table of Contents

[Docker Project 2](#_Toc177721137)

[Introduction 2](#_Toc177721138)

[Equipment and Materials 2](#_Toc177721139)

[Objectives 2](#_Toc177721140)

[Deploying a RESTful API with Database Integration, Managed as a Process Using Docker 3](#_Toc177721141)

[Overview 3](#_Toc177721142)

[Mandatory steps (Part 1 in your PDF) 3](#_Toc177721143)

[Mandatory steps (Part 2 in your PDF) 4](#_Toc177721144)

[Final Submission Requirements 5](#_Toc177721145)

[Marking Criteria 6](#_Toc177721146)

# Docker Project

## Introduction

In this project, you will build a web application using any programming language of your choice (e.g., Python, Node.js, Ruby, Java, C#, etc.). You’ll connect it to a database (e.g., MySQL, PostgreSQL, MongoDB, etc.) and deploy it using Docker on your local machine. Along the way, you'll also learn about key operating system concepts like process management, CPU scheduling, memory management, and file systems. The whole project is divided into four parts. Hereafter is Part 1 out of 4 Parts.

## Equipment and Materials

* Framework for the language of your choice (Python, .NET Framework, Java, etc.)
* [Docker](https://www.docker.com/).
* Database (MySQL, PostgreSQL, MongoDB, etc.)

## Objectives

Upon completing this project, you will have achieved the following objectives:

* Docker: Learn how to package and run apps in isolated containers.
* Programming language: Build a web app using any language you choose.
  + RESTful services: Understand how to create web apps and small services.
  + Database: Build a web app that works with a database.
  + Cloud integration: See how traditional OS concepts work in modern cloud environments (Part 2, etc.).
  + Docker skills: Get hands-on experience with Docker, a popular tool for easy app deployment.
  + Programming languages and frameworks: Learn important skills in programming and frameworks, needed to build big applications.
  + Database management: Understand how apps interact with databases to create reliable software.
  + DevOps practices: Learn basic DevOps skills like containerization and app deployment, essential in today’s tech industry.
  + Operating system knowledge: Understand how the OS manages processes (Part 1), CPU scheduling (Part 2), memory (Part 3), and file system (Part 4) to improve app performance in containers and cloud setups.

## Project Part 1 (of 4)

## Deploying a RESTful API with Database Integration, Managed as a Process Using Docker

### Overview

This project-part-assignment guides and offers a detailed walkthrough for developing a RESTful API connecting with a database, deploying it with Docker, and managing Docker containers as processes—specifically through creation, termination, and signaling—within a Windows 10, Windows 11 or Windows 10 VM, built in the first lab. To ensure thoroughness and clarity, you include all relevant screenshots and details in your submission. Such screenshots must be high resolution, **must include the current date**, and must include what is asked to be eligible for marks. Compile your final submission into a single PDF file.

### Mandatory steps (Part 1 in your PDF)

* **Code content includes the following code snippets (20 marks)**:
  + Provide the code for the application including endpoints and database connection as well as handling exceptions for common errors.

Add your screenshots here

* + Include the Dockerfile configurations used to create the Docker image.

Add your screenshots here

* + List the dependencies required for the application (e.g., requirements.txt).

Add your screenshots here

* **Compile and execute code (10 marks). Steps**:
  + Install all dependencies locally and take a screenshot(s) of the successful installation.

Add your screenshots here

* + Run the application locally to verify that it works as expected and take screenshot(s).

Add your screenshots here

* [**Deploy code on docker**](https://www.docker.com/products/cli/) **(10 marks). Steps**:
  + Create a Docker image using the Dockerfile (docker build -t …) and take screenshot(s) of the successful image creation.

Add your screenshots here

* + Run the Docker container with port mapping from the image created (docker run -d -p …) and take screenshot(s).

Add your screenshots here

* **Connect to DB Docker container (10 marks). Steps**:
  + Run a database container if not included in the main Docker image as a separate container and take screenshot(s) of the database container running.

Add your screenshots here

* + Connect to the database using [CLI tools](https://www.docker.com/products/cli/) or GUI applications and take screenshot(s) of the successful connection to the database.

Add your screenshots here

* **Select records from DB Docker container (10 marks). Steps**:
  + Access the database using the Docker CLI (docker exec -it db …) and take screenshot(s).

Add your screenshots here

* + Run a query to retrieve records (SELECT \* FROM … ) and take screenshot(s) showing student records.

Add your screenshots here

* [**Postman**](https://www.postman.com/downloads/) **screenshots (20 marks). Steps**:
  + POST request for creating a student record and take screenshot(s) in each step:
    - Set up a POST request in Postman to create a student record.
    - URL: <http://localhost:8080/student>
    - Method: POST
    - Body: Provide JSON payload for the request, sample input JSON:

{

"studentID": "Your student ID",

"studentName": "Your full name",

"course": "OS",

"presentDate": "current date"

}

* + Expected response: *HTTP status code 201 created, message student created successfully*. Take a screenshot showing that expectation plus the current date, your student ID, and your name, provided in JSON.

Add your screenshots here

* POST request for testing and handling the existing records and take a screenshot(s) such that use the same endpoint with identical data to test the handling of existing records.
  + Expected response: HTTP status code 409 conflict, message student already exists. Take a screenshot showing that.

Add your screenshots here

### Mandatory steps (Part 2 in your PDF)

* **Assign a name to the container**, start the Docker container in detached mode and take a screenshot (4 marks).

Add your screenshots here

* **Monitoring Docker containers (4 marks)**:
  + Using [Docker CLI](https://www.docker.com/products/cli/):
    - List running containers (docker ps) and take screenshot(s) of the output.

Add your screenshots here

* + - View logs and take screenshot(s) of the logging output.

Add your screenshots here

* + Using Windows Task Manager:
    - Open Task Manager (CTRL+ Shift + ESC) and then monitor the Docker processes under the Processes tab and take screenshot(s) showing Docker process.

Add your screenshots here

* **Terminate Docker container (4 marks)**:
  + Using Docker CLI:
    - Stop the container and take screenshot(s) confirming that the container has stopped.

Add your screenshots here

* + Using Windows Task Manager:
    - Find the Docker process, end it and take screenshot(s) showing Docker process termination.

Add your screenshots here

* **Signaling Docker container (4 marks). Graceful shutdown**:
  + Stop the container with a Docker command and take screenshot(s) showing graceful shutdown.

Add your screenshots here

* + Repeat the above step to force shutdown in that time and take screenshot(s)showing force shutdown.

Add your screenshots here

* **Quick monitoring and troubleshooting (4 marks)**:
  + Monitor resource usage using a Docker command and take screenshot(s) showing resource usage statistics.

Add your screenshots here

* + Inspect container state using Docker command (docker inspect …) and take screenshot(s) of inspection output.

Add your screenshots here

* + Check and view logs for issues using the necessary Docker command and take screenshot(s) of output.

Add your screenshots here

## Final Submission Requirements

## You should include the following items in a single ZIP folder:

* **PDF Document:**
  + Compile all screenshots and **all code snippets** into one comprehensive PDF file.
  + Ensure that all screenshots are high-resolution, with clear and readable text that meets the criteria for eligible marks.
* **Video Explanation:**
  + Record a short video explaining each screenshot. Ensure your face is visible in the video for a personal touch.
* **References:**
  + List all references used in this assignment.

Add your screenshots here

## Marking Criteria

|  |  |
| --- | --- |
| Item | Marks |
| Code content (App and Dockerfile) | /20 |
| Compile and execute code | /10 |
| Deploy code on Docker | /10 |
| Connect to DB Docker container | /10 |
| Select records from the DB in the container | /10 |
| Postman screenshots | /10 |
| Managing Docker containers on Windows 10 | /20 |
| A short video | /5 |
| References | /5 |
| Total Marks | **/100** |