Term Project

WORKING WITH CONTAINERS

DOCUMENT VERSION: 2.0

The primary objective of the term project is to help you gain exposure and experience with containers. Over the last decade, containerization has become a crucial component of both industry and academic development and deployment environments. If you become comfortable with both the concept and application of containers now, you will be in an advantageous position when entering the job market.

Take a Deep Breath! As you read through this document, you may see unfamiliar technologies being referenced (RESTful web server? Docker container? Kubernetes cluster?). As the semester progresses, we will release plentiful information about these topics, providing you with more than enough to complete your term project successfully. This supporting information will be in the form of lectures, Infospaces videos, recorded help sessions, documentation from us as well as pointers to official documentation online.

DUE DATE: This assignment will account for a maximum of 10 points towards your cumulative course grade. The term project has three deliverables.

TP-D1 due 9/30/2020.	[2 points]
TP-D2 due 10/28/2020	[3 points]
TP-D3 due 12/09/2020	[3 points]
Demo	[2 points]

1 Team

The term project is a team-based assignment, with teams numbering 3 students (in most cases). No more than 3 students are allowed per team. Similarly, attempting the term project alone is not allowed. You *must* find two other teammates, and only in rare circumstances (odd numbering of students, etc.) will teams of size 2 be allowed. Both distance and on-campus students will be treated as a single pool of students for team selection.

In order to find your teammates, you will be able to chat in the **TP Team Finding** channel within the **CompSci CS370 Fa20** Microsoft Teams team. Once you and two others have agreed to join a team, you will submit the team members per the Term Project Deliverable 1 instructions.

2 Deliverables

Throughout the semester, you will have **3** team deliverables to turn in. This is to make sure students are making progress on the project in a timely fashion, and no teams are procrastinating until the "last minute". The term project deliverable dates are as follows:

- **Term Project Deliverable 1**: Wednesday, September 30, by 5:00 PM Mountain Time
- **Term Project Deliverable 2**: Wednesday, October 28, by 5:00 PM Mountain Time
- **Term Project Deliverable 3**: Wednesday, December 9, by 5:00 PM Mountain Time

2.1 Term Project Deliverable 1

The goal of the first deliverable is to select your teammates, review the term project description with them, and come up with a creative project idea that meets the project specification and complexity level requirements.

For your deliverable submission, **one** of your team members will submit a PDF document to the **TP-D1** assignment on Canvas containing:

- A list of your team members in the format: <LastName>, <FirstName>, <eid>. Example:
  - Doe, Cameron, jdoe
  - Smith, John, smjohn
  - Anderson, Taylor, kingofguac
- Two paragraphs (4-8 sentences each):
  - The first of which describes the problem you are solving with your project design.
  - The second of which describes a description of how you think a containerized architecture could effectively address or solve the problem in the first paragraph.
- Adding headers or a title to your PDF is acceptable, so long as it contains the above.

Your preliminary idea is not final! It is just meant to get you started thinking about possibilities for your project, and how you could use containerization to solve them. You are more than welcome to change your idea after your preliminary suggestion; just email the CS370 GTA team your new idea.

Once the CS370 GTA team has received your team member list submission, and has approved your preliminary design and solution, we will assign you and your teammates a team number and add you to a Canvas group. Based on your team number, you will be assigned a virtual machine on one of the CS lab machines set aside for your Docker containers and Kubernetes cluster.

Note: It is crucial that you follow the instructions in this document. If the email title is not correct, or the format of the PDF is not correct, your submission will not be accepted, and you will be notified that you need to re-submit following the specified protocol.

2.2 Term Project Deliverable 2

For the second deliverable, **one** member of your team must submit a **.tar** file to Canvas assignment **TP-D2** containing your project in a *working state,* such that the following features are implemented:

- A RESTful web server, which is able to receive HTTP requests and log the request body to STDOUT.
- The ability to build/compile your project in a single command.
- The ability to run your web server in a single command.
- A README.md file containing professional markdown documentation on your project:
  - A "Description" section outlining the purpose of your project.
  - A "Usage" section containing instructions on how to build and run your project. Additionally, this section may contain options to provide at runtime, and their respective descriptions.
  - A "Docker" section, which contains instructions for building your Docker image, and more importantly, how to run the image as a container.
- A Dockerfile, which when built, builds a Docker image with your compiled project:
  - Running the Docker image should start your webserver.

Note: In order to have your project in a working state, but still be able to develop on it as a team, you can use a GitHub or Bitbucket repository shared with your teammates! Then, you can turn in a tar'ed version of the **master** branch, while you continue to develop on other branches.

2.3 Term Project Deliverable 3

For the third and final deliverable, **one** team member must submit a PDF report to Canvas **TP-D3** assignment describing your project (the entire team should contribute to writing it, but only one submission is sufficient). The purpose of the term project report is to thoroughly evaluate your project outside of the development process, and be able to communicate the original problem, design decisions, obstacles encountered, and future possibilities to an outside audience.

The report should contain the following elements:

- Title page with a list of all team members and eids
- Introduction
- Problem Characterization
  - A technical description of your problem, as if your audience is your peers.
- Proposed Solution and Implementation Strategy
  - Methodology
  - A description of what libraries you relied on, versus what you implemented yourself.
- Conclusions
- Bibliography, in APA format.
- Word count of 2500-3000, *excluding the bibliography*.

A Note on the Use of Citations:
Citations have a specific purpose, they: (1) relate to work that has been published elsewhere, (2) substantiate your claims, and (3) could be used by readers to dig a little deeper. Remember to number your references and list them in your bibliography in the order they are referenced throughout the paper. If an article is in your bibliography, it must be cited in the main text. Citing at the right location indicates what your source is for a particular piece of information and demonstrates that you have read the article. Make sure that you cite all your references including Wikipedia and Online lecture notes that you may have perused. References that are not cited should *not* be in your bibliography.

Other Notes Regarding the Report:
In your discussions, be aware of when you digress and the importance of tying it to the overall narrative. If you go off on too many tangents, the paper will tend to be dull, unorganized, and ambiguous.
If you are looking at your topic from the software prism, it might be worthwhile to see if the software choices were constrained or motivated by the underlying hardware. An article is much more compelling if it lays out the trade-off space. Identify the confines within which things happen. If a system has achieved a certain feature what are the other things (both hardware and software) that make this feature possible. Try to address both the advantages and disadvantages of support for specific features.

3 Project Description and Requirements

3.1 Code Language

It is the team's responsibility to choose a language for their source code. We recommend that you choose something you're all semi-familiar with, and that builds easily within a Docker container without requiring too many dependencies. Some common choices are Java, Python, Golang, and C.

3.2 Project Components

Below are the required project components, in order of abstraction:

Microservice Level

- A RESTful API endpoint, allowing requests to be made externally and be handled within the service.
  - HTTP protocol is the students' choosing: JSON, XML, or any other protocols are acceptable.
  - Logging of the HTTP request body to STDOUT, per the first deliverable.
- A backend, which provides some service for incoming REST requests.
  - Examples might include a web server to serve web pages, a database, a storage service, a computational service, etc.

Docker Level

- A minimum of **2** Docker containers, and a maximum of **4.**
  - At least one Docker container which provides the RESTful API.
  - At least one Docker container which services requests as a backend for the other container running the RESTful API.
  - A communication strategy between the Docker containers.

Kubernetes Level

- A Kubernetes Pod, which contains the Docker containers.
- A Kubernetes Service, which allows requests to be routed from an external source to the appropriate container running the RESTful API.
- Ports used for API endpoints must be within the students' allocated range and *must not conflict with ports from other students*.

3.3 Term Project Demo

Congrats for making it this far in the semester! Now that you've finished all the development on your term project and report, it's time to show off your team's hard work and creativity. We will be hosting demos over Microsoft Teams, which will allow your team to demonstrate the functionality and features of your project to a GTA. Due to it being online, demo logistics will be announced towards the latter half of the semester.

3.4 Grading Rubric

The term project is worth **10 total points** towards your final grade, with the following breakdown:

- Deliverable 1: **[Total 2 points]**
  - Well-thought and creative preliminary problem and solution proposal [2 points]
- Deliverable 2: **[Total 3 points]**
  - Functional RESTful web server [1 point]
  - Functional Dockerfile [1 point]
  - Thorough, professional documentation [1 point]
- Deliverable 3: **[Total 3 points]**
- Demo: **[Total 2 points]**
  - Met Docker level requirements [1 point]
  - Met Kubernetes level requirements [1 point]

4 Late Policy

Click here for the class policy on submitting late assignments.