New York University Tandon School of Engineering
Computer Science
Course Outline CS-GY 9223 DevOps
Fall 2020
Professor Eugene Callahan
Tuesday
10:00 AM – 11:30 AM Live Online Session

To contact professor: ejc369@nyu.edu
Phone: 646-997-3476
Office hours: by appointment

TA: Dennis Fenchenko
df1911@nyu.edu
Phone: 302-359-7702

Course Pre-requisites: Basic courses on programming.

Course Description: DevOps is the latest paradigm on how to develop software. It combines lean and agile development methods with the latest suit of operations tools to enable fast, worry-free, and regular shipping of code into production. The focus of the course will be on the DevOps way of working, but we will also survey the wide range of DevOps tools, and use some of them.

Course Objectives
1. Understand the DevOps way of organizing work.
2. Be familiar with the range of DevOps tools.
3. Understand how the tools support the way of work.
4. Become familiar with particular instances of several of the tools.

Course Structure
One lecture per week, and regular development work, reading, and videos.

Readings
Listed here:
http://www.thedevopscourse.com

Course requirements
Release work into production at least twice per week.

A *rough* grade breakdown, which will be adjusted as the semester proceeds:

Project: Approx. 55% of grade. **Regular**, incremental releasing of software is the largest component of the student's grade. Please note that you are graded on the project quality overall, as well as your individual contribution to the project.

Class participation: 5% of grade.

Tests during the semester, Approx. 20% of grade

Final Exam, Approx. 20% of grade

Course Project
Throughout the semester students will work on developing a project of their choice using the core principles and tools of DevOps. The main objective of the project is to develop a web application using Flask-RestX on the backend and (possibly) React on the frontend, which are capable of communicating with each other. The project includes a series of checkpoints, approximately every two weeks of the semester. At every checkpoint, the group of students will be required to demonstrate the MVP of their application which incorporates features learned in class. Some of the projects that students might choose to pursue are, but not limited to Social Media, E-Commerce, Weather, Messaging, Productivity, Betting, or Courses sign up application. Below you can find project breakdown into checkpoints.

Phase 1, due Sept 22
- Form a group of 3 (Exceptions can be made if you'd like to form a group of 4 or smaller);
- Set up a Slack channel for communications within your team;
- Set up a GitHub repository for your project;
- Set up a Kanban board for managing workflow within your team using GitHub "Projects" feature.
- Formulate your project proposal

Phase 2, due Sept 29
The purpose of this phase is to develop a template for your application and set up primary tools that you need. You will keep developing each part of your program throughout the semester.
- Setup a Flask-RESTX API server;
- You should aim to have, but not limited to, 8 post and get endpoints;

Phase 3, due Oct 6
- Create a Makefile allowing you to run tests locally, as well as push code into production.

Phase 4, due Oct 20
- Keep developing your application;
- Testing: **you should be writing tests as you write your code**. Testing includes unittest for Python code, Jest for testing React, as well as linters such as flake8 and eslint for Python and React respectively, which enforce proper code styling. Your submission this week must include automated tests!

Phase 5, due Nov 3
- Set up a CI/CD pipeline using Travis CI. Use it to push your code to PythonAnywhere.

Phase 6, due Nov 17

- Create a <u>Docker container</u> for your application. Deploy it to a cloud service. Create monitoring.

Phase 7, due Dec 8
- Final project hand in and presentations.

Moses Center Statement of Disability

If you are student with a disability who is requesting accommodations, please contact New York University's Moses Center for Students with Disabilities (CSD) at <u>212-998-4980</u> or <u>mosescsd@nyu.edu</u>. You must be registered with CSD to receive accommodations. Information about the Moses Center can be found at <u>www.nyu.edu/csd</u>. The Moses Center is located at 726 Broadway on the 2nd floor.

NYU School of Engineering Policies and Procedures on Academic Misconduct

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.

B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:

1. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
2. Fabrication: including but not limited to, falsifying experimental data and/or citations.
3. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.

4. Unauthorized collaboration: working together on work that was meant to be done individually.
5. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.
6. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.