

ECE 465 - Cloud Computing

Spring 2021

The Cooper Union

General Course Information

Instructor: Prof. Rob Marano
Email: rob@cooper.edu & robmarano@gmail.com
Semester of the course: Fall 2020
Dates of the course: 21 January – 13 May 2021

Course Description

You will dive deep, hands-on, to study critical, foundational technology components that enable cloud computing, and the engineering advancements that have led to today's thriving cloud computing ecosystem. Students understand, design, build and test representational software units that implement different distributed computing components, e.g., parallel logic, software-defined networking, computers, storage, databases, and security. Topics include multi-threaded programming; functional programming (map-reduce). Hadoop: a programmer's perspective; streams processing to collect data; AI/ML inference model training, validation, and execution; & modern development and DevOps of serverless applications. This course prepares the student with foundational knowledge and experience to prepare for a variety of cloud computing professional certifications.

3 credits. 3 hours per week (45 total hours).

Course Prerequisites

ECE 251 and ECE 264 serve as the minimum prerequisites.

Course Structure/Method

Lecture/Lab. This class meets remotely via Microsoft Teams on 1/21; 1/28; 2/4; 2/11; 2/18; 2/25; 3/4; 3/11; 3/18; 3/25; 4/1; 4/15; 4/22; 4/29; 5/13, for a total of 15 sessions. **Note** that we do not meet on 4/8 nor on 5/6. The class meets from 6:00 to 9:00 pm on all days,

and it starts 10 minutes after the hour, and ends 10 minutes before the hour. Before class, please get ready and prepare your thoughts and development environment.

Course Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Establish baseline experience and knowledge of command-line scripting to configure a leading cloud computing environment.
2. Demonstrate design and implementation of multi-threaded and multi-node networked processing software using Java or Python.
3. Understand and demonstrate configuration and execution as well as operations of cloud computing, storage, database, networking, and development of modern serverless applications.
4. Collaborative programming using GitHub for source code and project documentation, in addition to Kanban agile management.

Communication Policy

The best way to contact me is via email then by Slack. I will do my best to respond within 24 hours. Communication and participation in class is not only encouraged, but required.

Course Expectations

Class Preparation

Each session will consist of two components: discussion and lab. Discussion consists of a mix of lectures, programming examples, and question-driven group analysis of one or more large programming problems. Lab will consist of either group or individual work on exercises or projects. Questions arising during lab may be used to fuel additional discussion as time permits.

Attendance

Success as a student begins with attendance. Class time is not only for learning new skills, but also for practicing what you have learned. Some assignments and demos may be completed in class, but practice and study are required outside of class. Students are expected to attend classes regularly, arrive on time, and participate.

I take attendance during every session, and it forms part of your grade. Students are encouraged to e-mail me when they are absent. Students are responsible for all academic work missed as a result of absences. It is at my discretion to work with students outside of class time in order to make-up any missed work.

Materials

Software

All software will be open source.

Assessment Strategy and Grading Policy

All assignments must be completed by the end of this course in order to receive a passing grade. Your final project is approximately 60% and attendance is approximately 20% of your overall grade.

Title	Pts	Due Date
HW 1a - 1-node, threaded dist algo app	4	1/29 11:59pm ET
HW 1b - Next rev 1-node, threaded dist algo app	6	2/5 11:59pm ET
HW 2a - n-node, threaded dist algo app	4	2/12 11:59pm ET
HW 2b - Next rev n-node, threaded dist algo app	6	2/19 11:59pm ET
HW 3a - AWS setup & config/run n-node, threaded app	4	2/26 11:59pm ET
HW 3b - Deploy n-node app onto Elastic Beanstalk	4	3/5 11:59pm ET
HW 4 - Design/implement secure Wordpress site on EC2, Aurora	10	3/25 11:59pm ET
Final project: design/implement secure serverless app & UI	30	4/14 11:59pm ET
Class attendance	30	2 pts/class

Final Projects MVPs

The course assessment expects you to design, implement, and leverage DevOps in a single final project. Get started early, but pass your ideas past the professor. The following constraints govern the grading mechanics of your final project.

1. Collaborate on teams of 2 people, not less, not more.
2. Source code to be maintained in a GitHub repository per team. Invite your professor as a collaborator.
3. Design and document in GitHub Wiki the software architecture and source code of your team's MVP.
4. Breakdown the MVP design in manageable sets of tasks, and track high-level via GitHub Project.
5. Document the design of each subsystem for your MVP in its appropriate GitHub Wiki section.
6. Demonstrate the MVP as part of your team's final presentation on the last day of class.

Course Outline

Session 1: 01/21/2021, 6pm-9pm

Description: Course Overview; History of IT computing & distributed computing along with HPC; von Neumann architecture & his stored-program concept - recall computer architecture basics; distributed computing architectures including serverless; Moore's Law and now that free lunch is over for software engineers; cloud computing models; cloud deployment models; "sprechen Sie Sprachen der Wolke" - do you speak languages of the cloud?; let's dig into command-line Bash, mapreduce, JSON on jq, and beyond; Java threading and distributed algorithms... Whoa!

Assignments: Choose one distributed algorithm and then implement MVP1 a Java single-node, multi-threaded (not multi-process with IPC) program.

Check out <https://github.com/robmarano/threadedJavaJourney> for Java threading program structure ideas; research producer-consumer programming models. Think Big, like blockchain or distributed image processing or even your own network filesystem, or as simple as the Dining Philosophers problem. **Due on 1/29/2021 11:59pm ET**

Session 2: 01/28/2021, 6pm-9pm

Description: Discuss other cloud-ready languages, other distributed algorithms and Big-O notation; hack more threaded Java on your computer.

Assignments: Next revision of your single-node, multi-threaded program. **Due on 2/5/2021 11:59pm ET**

Session 3: 02/04/2021, 6pm-9pm

Description: Reviewing threading topics; introduction to AWS Console and your account; AWS Identity and Access Management through the lens of the AWS Console and the AWS CLI.

Assignments: Re-design your single-node, multi-threaded Java to run on many networked nodes in a either a primary-secondary mode or peer-to-peer mode. You will design your own network protocol for handling inter-process communication between the nodes. **Due on 2/12/2021 11:59pm ET**

Session 4: 02/11/2021, 6pm-9pm

Description: Software-defined networking; AWS VPC and CloudFormation, as well as CDK; EC2 - all through the AWS Console and with AWS CLI.

Assignments: Next revision of your multi-node, multi-threaded program. **Due on 2/19/2021 11:59pm ET.** Brainstorm ideas for your complex, multi-stage final project with your teammate.

Session 5: 02/18/2021, 6pm-9pm

Description: More exercises and live coding on AWS VPC, EC2, CDK; ephemeral EC2 storage; EBS storage

Assignments: Migrate your multi-node, multi-threaded program to AWS EC2 on its own secure VPC. **Due on 2/26/2021 11:59pm ET.**

Session 6: 02/25/2021, 6pm-9pm

Description: Abstracting EC2 away and introducing a managed code deployment and execution service called Elastic Beanstalk; Simple Storage Service (S3).

Assignments: Re-design and deploy your multi-node, multi-threaded Java app on to Elastic Beanstalk. **Due on 3/5/2021 11:59pm ET.**

Session 7: 03/04/2021, 6pm-9pm

Description: Abstracting our compute and storage, what are the various, purpose-built cloud-based databases? Do you have "maintain" them? Abstracting out support and maintenance... Covering relational, NoSQL, graph, and time-series databases.

Assignments: Research the most common application today - informational websites built with content management systems (CMS). Focus on WordPress with a relational database backend. Migrate a typical WordPress site to AWS so that it is secure with minimal support and maintenance. **Due on 3/26/2021 11:59pm ET.**

Session 8: 03/11/2021, 6pm-9pm

Description: Wait, let's abstract compute from a virtual machine to a function! How do you manage code, new and updated, in an automated fashion. Wait is that DevOps too? How about centralizing my distributed application logs and analyze the continuous reams of logs to find problems? Abstracting further from traditional IT systems and processes to streamlined, everything-is-software-defined, so that you can focus just on the problem to be solved with logic - your differentiation in the marketplace or thesis uniqueness! ;-) Oh, more on cloud storage.

Assignments: Work on your Wordpress migration to serverless and your final project.

Session 9: 03/18/2021, 6pm-9pm

Description: More on Lambda, but now with all that work on containerized software developed on Docker, well reall from Unix first, how do they move to the cloud?

Assignments: Work on your Wordpress migration to serverless and your final project.

Session 10: 03/25/2021, 6pm-9pm

Description: Continuing with software containers; AWS ECS, EKS.

Assignments: Work on your Wordpress migration to serverless and your final project.

Session 11: 04/01/2021, 6pm-9pm

Description: Putting it all together to create cloud applications that scale to millions and millions of users and only pay for what you use. Mind blown! In years past, companies paid UP-FRONT for equipment that would stand idle but needed for surge and growth capacities. That equipment was already old in functionality 18-24 months after purchase, but they would amortize it over 5 to 10 years. Not good for continual, rapid innovation... Let's see what we need to build serverless applications and client-side MVC systems, exploiting the advantages of the internet and cheaper, more powerful edge devices.

Assignments: Work on your final project incorporating these serverless appdev components into your final project designs.

Session 12: 04/15/2021, 6pm-9pm

Description: What about handling Twitter, Facebook, TikTok-scale or even bigger like IoT devices and their feeds of new data arriving constantly? Text, images, videos? You'll need a highly scalable streaming system. So what about cloud messaging and ingestion. Do I ever throw data out? Hmm... How big is your cloud?

Assignments: Work on your final project incorporating these serverless streaming components into your final project designs.

Session 13: 04/22/2021, 6pm-9pm

Description: So how do all those streaming IoT devices ensure their telemetry data to the cloud? How is it processed? It has to be fast and scalable. Let's dig into using cloud computing to enable such processing systems. Sometimes decisions need to be made as close to the IoT device at the edge of the network due to latency. Btw, these systems don't only have to be limited to IoT telemetry, like temperature readings. Why not audio over the wire? Video? Let's go!

Assignments: Work on your final project incorporating these serverless streaming components into your final project designs.

Session 14: 04/29/2021, 6pm-9pm

Description: One of the best examples of using distributed computing (in the cloud) includes machine learning (ML) and artificial intelligence (AI), that is, teaching a computing system to think similar to a human to make decisions to take actions based on problems that don't necessarily have a closed form solution and may have increasing numbers of inputs to consider. We'll cover Amazon's Jupyter notebook service, i.e., SageMaker, to learn how to build and execute models in the cloud and at the (IoT) edge.

Assignments: One more week: Work on your final project incorporating these serverless streaming components into your final project designs.

Session 15a: 05/13/2021, 6pm-9pm

Description: Final project presentations from each team of two. All materials including code are to be stored on your git repository.

Assignments: Final repo for source code and presentations due 5/14/2021 at 11:59:59pm. Feel free to record your video presentation again if you want me to renew your grade from class presentation. Make video available on YouTube. I encourage this highly!

Session 15b: 05/14/2021, 11:59:59pm

Description: Final, working source code due on your team's main branch.

See <https://github.com/github/renaming>.

Assignments: Officially hand-in your two-person team project.