CSCI4145/CSCI5409 Cloud Computing – Group Project

Overview

The technology of cloud computing changes more quickly than most other technology in software development, in fact this is part of why companies have flocked to putting their software in the cloud. They can ride this wave of technological advances to keep their software on the cutting edge. The services and technologies you learn during this course will be deprecated, changed, or expanded upon within only a few years. For this reason, the most critical skill for you to develop in this course is your ability to dive into a cloud provider's service offerings, to learn what they have available, and how the services you are interested in work. To work in the field of cloud computing you must be confident that you can dive in and work with any of the services a cloud provider offers.

The group project is the primary driver for experiential learning in this course. We can only truly say you understand cloud computing once you've built and deployed a working software system in the cloud. To accomplish this goal will require tenacity, self-directed learning, experimentation and critical analysis. You are 4th year students or graduate students; this is what you've been training for. **You got this!**

Group Size

You will work in groups of 1 – 3 students. I encourage most of you to work together, but sometimes students just can't stand the thought of working with other students and you've already learned the lessons of being good group members in other courses.

There will be a spreadsheet named Groups.xslx in the Files tab of Microsoft Teams. Students must form their own groups and enter their group information in the spreadsheet. Unless students request to be in a 1-person group, we will automatically assign students who have not formed their own groups to a random group of 3 students on the group formation deadline. Graduate students in CSCI5409 may only group with other enrolled students in CSCI5409.

There are a limited number of 1 person groups we can support due to the need for TA advice, supervision and marking. We will play this by ear rather than setting a fixed amount. Obviously if everyone wanted to do the projects on their own I would have to draconically form random groups.

The instructor reserves the right to at any time:

- **Move students from one group to another**
- **Merge groups**
- **Dissolve groups**

I will alter groups if I see problems that affect a student's ability to learn, or if I see problems in terms of the distribution of effort within the group.

Requirements

There are many career paths in cloud computing. From traditional app developers, to data scientists, machine learning, IoT and even people just interested in supporting cloud app development (DevOps). For this reason, rather than describing a project for you to implement I am leaving the nature of your project largely up to your group to decide. There will be a menu with different categories of technologies, services or methodologies you must select from, however what you build with these technologies is entirely up to you. Use this opportunity to gain experience with something you might want to work with further when you enter industry or to build some cool thing you can further develop and even bring to market.

No matter what services you choose from the menu, you will also need to develop the software that runs on the services you select. **You are not allowed to use existing code (not even your own from other courses or personal projects), all code you write for this course must be original and entirely created by your group members for this course.** The software does not have to be a traditional web page. It could be an API, a website, a mobile app that talks to an API, etc. The choices are endless.

Each category below will require you to select a number of items from the list of choices to include in the design and implementation of your project. Pick the things you are most interested in learning about or doing. The instructor has no preferences here, all choices are equally valid.

Categories Menu

Compute – Pick two (2):

- AWS EC2 – Host a web app in a virtual machine
- AWS Elastic Beanstalk – Automatically run, load balance and scale web apps via virtual machines
- Docker & AWS Elastic Beanstalk – Run a web app in a container with the Docker platform
- AWS Elastic Container Service – Best way to run Docker containers **(Note: Not supported by AWS Academy, you'd need to use your starter account or professional account)**
- AWS Lambda – Functions that run without servers! Amazing! This is the future!
- AWS Step Functions – Build a serverless state machine!

Storage – Pick one (1):

- AWS S3 – Simple file storage
- AWS DynamoDB – NoSQL database
- AWS Aurora (be careful, this will eat your credits up fast!) – Managed database

Network – Pick one (1):

- AWS Virtual Private Cloud (VPC) – A private network protecting internal services speaking to each other over a virtual private network
- AWS API Gateway – Secure and route API requests to lambdas or container APIs (**Note: not available via your Academy account, you would need to use your starter or professional account to use this service)**
- Amazon CloudFront – Content delivery network for high-speed content delivery
- Amazon EventBridge – Build event driven architectures by decoupling event sources and HTTP endpoints (lambdas, microservices, etc.)

General – Pick two (2):

- Use Heroku and AWS to build a multi-cloud system
- Amazon Cognito – Add user sign up, sign in and access control to your app with support for social identity providers (Facebook, google, etc.)
- Amazon Comprehend – Natural language processing with machine learning for deriving and understanding valuable insights from text within documents
- AWS SNS – Send text messages, emails or mobile push notifications
- AWS SQS – Build a producer/consumer type system where one services produces output to a queue that another service consumes
- AWS Secrets Manager – Secure secrets in your application (DB usernames / passwords, API keys, etc.)
- AWS Key Management Service – Store your cryptographic keys outside of your app
- AWS Kinesis – Capture data streams and do something with them
- AWS Robomaker – ROBOTS!!!! That's all that has to be said here.
- AWS Glue – Sanitize or alter data for machine learning processing
- AWS Elastic Load Balancing
- Amazon Lex – Build chatbots with conversational AI
- Amazon Polly – Text to speech, you send text and get back an audiostream!
- Amazon Rekognition – Automated image and video analysis
- Amazon Textract – Automatically extract text, handwriting and data from scanned documents
- Amazon Translate – Fast, high quality text translation
- Difficult - Brave the wilds of the FCS OpenStack infrastructure to build a hybrid system that combines public cloud and private cloud
- **(4145 Students Only)** Difficult - Infrastructure as code: build a CI/CD system to automatically build and deploy and provision your infrastructure via AWS CloudFormation
- Difficult - AWS Machine learning – Anything you do here will be both impressive and difficult. Caution: You cannot use your work from other machine learning courses.

We are restricted to the services supported in the AWS Academy sandbox. Not all AWS services are supported, and sometimes they don't work as well in the sandbox as they do in the real

AWS infrastructure (for example IAM is extremely limited in AWS Academy unfortunately). Please refer to the "AWS Academy Learner Lab - Foundational Services" document on Brightspace to see a full list of available services and any special considerations in their use in the lab environment. If you see something available in AWS Academy that you want to use that isn't listed here, then just ask! We'll tell you whether you can use it and which category it applies to.

CSCI5409 Students Extra Requirement

Because you are graduate students your expectations are higher than undergrad students. **This semester we will be requiring all graduate students to provision their infrastructure with AWS CloudFormation.** We will not study CloudFormation in the tutorials until the very last tutorial in the semester, therefore it is critical that each member of your group learn and apply this tool very early in your project. Start right from the beginning with CloudFormation as your method for provisioning your infrastructure in the cloud, avoid using the console as much as possible.

Critical Analysis & Response Tasks

After we cover each of the major sections of cloud computing topics in our lectures, your group will be posed a set of questions related to the material. Your group will need to meet and work together to create a written response to these questions. These questions will force you to critically analyze the system you are developing and respond with how you will address the topic in your projects. Consider these tasks as group assignments that help you progress towards the implementation of your project.

1 - Project Proposal, Deployment & Delivery Model

Your group proposes **what** you will build, and **how** you will build it. You will write a document that describes the application you intend to build and how it will work. You will also have to explain how you will deploy your app. Will it be entirely on AWS or use a multi-cloud or hybrid cloud approach?

2 – Mechanisms

Your group explains which services your project will use, and why you are using those services over other potential choices or approaches.

3 – Architecture

Your group will explain the cloud architecture you are using, or, if your system doesn't fit an architecture taught in the course you will explain why that is, and whether your choices are wise or potentially flawed.

4 – Security & Business Considerations

Your group will analyze and describe your project's approach to security, particularly its approach to securing data. You will also analyze the cost metrics for operating your system. You will calculate the up-front, on-going and additional costs to build this system in the real world.

You will also explain alternative approaches that might have saved you money, or maybe provide justification for a more expensive solution.

Project Deliverables & Grade Distribution

The group project represents a combined 50% of your grade.

Critical Analysis & Response Tasks – 20%

- Project proposal, deployment & delivery model – 5%
- Mechanisms – 5%
- Architecture – 5%
- Security & business considerations – 5%

Final Written Report – 10%

This document will combine your responses from the critical analysis tasks, as well as expand on the final result of your project. You will describe the architecture of your final system and describe each group member's role in the implementation of the project. This report identifies and proves how you met your menu item requirements.

Video Presentation – 20%

Your group will record a video that demonstrates all of the functionality of your application. This is where you prove that what you built actually works. We aren't going to grade the quality of the code you write for the individual services you implement, what we care about is the final working product. Your video will demonstrate the back-end implementation, the AWS configuration (and any other cloud services, e.g. OpenStack), and every part of the functionality it provides. This is your chance to show it off and show us the cool thing your group made. 5409 students must demonstrate the full provisioning of their application via CloudFormation.

Group Communication

To ensure all group members have a reliable and accessible mechanism to communicate with other group members, and to help us in the resolution of group conflicts, all students are **required** to use Microsoft Teams for group communication. Dalhousie's language of instruction is English, and therefore **all communications must be in English**. Each group will have its own private channel within the course Team channel to discuss their project, hold online meetings and to communicate with the instructor and your TA.

Group Work & Conflicts

You are working on a group project and are assessed as a group. Each of you shares an equal responsibility for contribution to the project. We do not want an imbalanced level of effort on the projects. Unlike other projects, you cannot focus on only one aspect of the project (e.g., documentation, testing, reports, etc.). **Each of you must participate in all aspects of the project, especially its implementation.** If it is determined at the end of the course that you did not contribute equally to your group, you may be subject to a grade decrement (see below).

We will determine your group contribution based on the following metrics:

- Attendance in group meetings
- Reports you write indicating how you've distributed work
- Oral and written reports from your fellow group members
- Observation by the instructor or TAs

We will assume you are a good member of your group and that your group members have no complaints against you until we hear otherwise from your fellow group members or from our own personal observations.

We will use the following levels of grade decrements; this penalty applies to the full weight of your group project on your final grade:

Underperformed vs. rest of the group: -15%
Significantly underperformed vs. rest of the group: -30%
Performed less than half the work of rest of the group: -50%
Contributed almost nothing: -90%
No group work/contact: -100%

Suggested Project Timeline

This timeline is my best guess at the best path to success for your group working on your cloud-based application. The "what you should work on" I describe below is very general and may need to change depending on what your group is building. **It does not account for the specific programming requirements unique to your project.** You will need to adjust this plan depending on the complexity you're striving for.

Date	Course Topic	What you should work on / project deliverables
January 18 - 20	Intro, key terminology, goals & benefits, risks & challenges	Your groups are formed this week. Introduce yourselves and start talking to each other about what kinds of projects you might be interested in taking on. Use a Doodle poll to figure out a regular meeting time. At first, once per week should be OK as long as you are productive, plan out your tasks and work well once apart. Later, or if you are not seeing progress, you should meet at least a couple of times per week. Pay attention to introductory lectures to learn what cloud computing can do, use this to come up with ideas for your project.
January 21 - 27	Cloud-enabling Technology	Start thinking about software you use on a daily basis: what kind of data does it store, what kinds of components might that software be made of, and which of these would be in the cloud? Your group

		should be throwing ideas around like crazy until you find one that sticks. When you have an idea, if you don't know where to start in terms of meeting the project requirements now is the time to work with your TA and get advice!
January 28 – February 4	Deployment & delivery models, project discussion	Pay attention to these lectures, you will need to choose a deployment and delivery model for the project you build, and you must be able to justify your choices. In the labs you've learned a lot about Docker and containers. Now is the time to decide which computer technologies you will use for your project. Containers or virtual machines on EC2. Your group should have a solid idea of what kind of cloud-based app you want to build, and now you are searching for the tools and right combination of AWS / other services to make it happen.
February 5 - 13	DevOps & managing releases, guest lecture	Dive into services on AWS Academy. You may not have learned all of the details of the services, or how to use them, but you should understand what they all do and what role they play in cloud computing. If you are stuck seek my advice! Your group must work intensely together this week to finalize your project proposal. **Your project proposal and deployment & delivery model critical analysis and response report is due February 13th at 23:59 Atlantic time.**
February 20 – 27	Winter study break	Use this week to learn any technology your project and group members need you to use that you may not have experience in, use AWS Academy to your full advantage. Make sure by the end of this week you're ready to be an equal contributor to your project. Your group needs to prepare your critical analysis and response on the mechanisms your group plans to use to implement your projects. **Your mechanisms report is due February 27th at 23:59 Atlantic time.**
February 28 – March 4	Fundamental cloud architectures	It's time to get to work! Get coding and provisioning your project infrastructure. Equally divide the work of provisioning IT resources and writing the code you need to run your app, do not assign all of any one task to one person. Use a "whole team" approach where every group member understands the architecture and implementation of the whole project so that you can assist each other in your work.

March 7 – 11	Advanced cloud architectures	At this point you've learned various architectural approaches to designing software for the cloud. Because of timing you've already begun work on your projects, maybe your app doesn't match one of these architectures but that's ok. In your critical analysis and response task you will be given the opportunity to compare what your architectural choices are to the ones you've learned in class, there's no wrong answer as long as your app works, but at least you'll have a chance to analyze how your architecture differs and the pros and cons of your choices. **Your architecture critical analysis and response report is due March 20th at 23:59 Atlantic time.**
March 14 – 18	Security	Continue working hard! You've got this! Aim to reach **33% functional completeness** by this date.
March 21 – 25	Business considerations	You'll need to prepare your critical analysis and response report on security and business considerations this week. This will be a very hectic and busy time of the semester, with many deliverables in other courses as well I would imagine. Plan your time wisely! Tell your friends/family/guildmates/significant others that you can't talk to them until April. ;) **Your security and business considerations analysis and response report is due March 29th at 23:59 Atlantic time.** This won't take you very long to complete, so also **aim to reach 60% functional completeness** by this date.
March 26 – April 6 (~2 weeks)	Case studies and discussion	This is crazy go time! Go, go, go, get that project 100% complete! **Your final report and video presentation is due on April 6th at 23:59 Atlantic time.**