

CSCI4145/CSCI5409 Cloud Computing – Term Assignment

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 advancement 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 jump right in, figure services and tools out, and adapt to the changing technical "cloudscape".

The term assignment 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 working software in the cloud, which is why the term assignment is **an individual assignment to be completed by you, and you only**. 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. Every semester my students continue to amaze me with what you accomplish. **You've got this!**

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 you 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 code from other courses or personal projects), all code you write for this course must be original and entirely created by you 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, IoT devices that connect to a service-oriented architecture, etc. The choices are endless, and creativity is your friend to keep you engaged and passionate about learning these difficult services.

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 & Elastic Container Registry – Best way to run Docker containers!
 - **OR** Amazon Elastic Kubernetes Service (EKS) (do not combine with ECS)
- **AWS Lambda** – Functions that run without servers! Amazing! This is the future!
- **AWS Step Functions** – Build a serverless state machine!
- AWS IoT 1-Click – Trigger lambdas from IoT devices!

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
- AWS AppSync – GraphQL autoscaling and data synchronization
- AWS Athena – SQL querying of data stored in S3
- AWS IoT Analytics
- AWS Neptune – Fast, reliable graph database
- AWS Relational Database Service

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
- 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.), some students have encountered issues with this service, try it out first before committing.

General – Pick two (2):

- Use [Heroku](#) and AWS to build a multi-cloud system
- AWS Backup – Centralized backup across AWS services
- AWS Batch – Run batch jobs at any scale (must demonstrate large scale capability)

- Amazon Comprehend – Natural language processing with machine learning for deriving and understanding valuable insights from text within documents
- AWS Data Pipeline – Move data around between services.
- AWS DeepComposer – Machine learning enabled musical keyboard and ML training tool
- AWS DeepLens – Deep learning enabled video camera
- AWS DeepRacer – Race a car with reinforcement learning
- Amazon ElastiCache
- AWS Elastic Load Balancing
- **AWS EventBridge** – Build event driven apps
- Amazon Elastic Inference – GPU acceleration to EC2 and SageMaker instances
- Amazon Forecast – Forecast business outcomes using machine learning
- AWS Glue – Sanitize or alter data for machine learning processing
- AWS Glue DataBrew – Data cleaning and normalization
- AWS GuardDuty – Threat detection and action service.
- AWS IoT Core
- AWS IoT Greengrass
- Amazon Kendra – Enterprise level search powered by ML
- AWS Key Management Service – Store your cryptographic keys outside of your app
- AWS Kinesis – Capture data streams and do something with them
- Amazon Lex – Build chatbots with conversational AI
- AWS OpsWorks – Automate operations with Chef and Puppet
- Amazon Personalize – Recommender system (be careful on credit usage, backed by ML)
- Amazon Polly – Text to speech, you send text and get back an audiostream!
- Amazon Rekognition – Automated image and video analysis
- AWS Robomaker – ROBOTS!!!! That's all that has to be said here.
- AWS Secrets Manager – Secure secrets in your application (DB usernames / passwords, API keys, etc.)
- **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
- Amazon Sumerian – Build VR, AR and 3D Applications
- Amazon Textract – Automatically extract text, handwriting and data from scanned documents
- AWS Timestream – Fast time series database for storing and analyzing events
- Amazon Transcribe – Powerful speech recognition
- 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** – Set up a CI/CD infrastructure in gitlab that automatically builds docker containers, registers them in Amazon Elastic Container Registry (ECR), automatically deploys container to Amazon Elastic Container Service and then performs automated testing on the builds.
- **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. This will be checked.**

We are restricted to the services supported in the AWS Academy learner lab 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 - Services" document on Brightspace to see a full list of available services and any special considerations in their use in the lab environment.** Pay particularly close attention to the necessity to use the existing LabRole IAM role created for the lab environment which grants access from one service to another within AWS.

If you see something available in AWS Academy that you want to use that isn't listed here, then just ask the instructor! 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 learn about CloudFormation in the infrastructure as code lecture, however do not wait for this! **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.** Not only will this meet your requirement, but it will protect you in the event of data loss on AWS Academy.

Project Deliverables & Grade Distribution

For CSCI4145 students the term project is 30% of your final grade.

For CSCI5409 students the term project is 20% of your final grade.

This total is split 50/50 between the final written report and video presentation described below.

Final Written Report

It is our recommendation that you build this report throughout the semester, slowly adding to it as you learn the material it must describe and make the choices for your own system.

Think of it as a journal of your design and implementation of this system.

The following is a **brief** description of what the report will include. For the full description of required sections in the report, and to see the marking rubric, please view the assignment on Brightspace.

In this document you will describe the following:

- What you built and what it is supposed to do.
- How you met your menu item requirements: you will list the services you selected, and provide a comparison of alternative services, explaining why you chose the services in your system over the alternatives.
- A description of the deployment model you chose for your system, with reasoning for why you chose this deployment model.
- A description of the delivery model you chose for your system, with reasoning for why you chose this delivery model.
- The architecture of your final system. If your system does not match an architecture taught in the course you will describe why that is, and whether your choices are wise or potentially flawed.
- An analysis of your project's approach to security, particularly its approach to securing data through all stages of the system (in transit, at rest).
- An analysis of 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 alternatively provide justification for a more expensive solution.

Video Presentation

The following is a brief description of what the video will include. For the full description of required video content, and to see the marking rubric, please view the assignment on Brightspace.

We need you to show us your working application and that you really did what you said you would. This is where you prove that what you built 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. You 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 in the working live version of your system. 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.**

Suggested Project Timeline

This timeline is my best guess at the best path to success for 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 you are building, currently it is adjusted strongly to the side of students that will require a lot of self-directed learning to get started with AWS. **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. The more difficult your project the more you should begin programming earlier.

Date	Course Topic	What you should work on / project deliverables
Week 1 & 2	Intro, key terminology, goals & benefits, risks & challenges	Start exploring, figure out what you want to build. Lookup each of the services in the list above and get creative. Plan, and start to follow your plan.
Week 3	Cloud-enabling Technology	You should know what you want to make by now and begin working on it. You haven't learned a lot of cloud technologies yet, but you could for example build any front ends or GUI components it will require.
Week 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 finalize which compute technologies you will use for your project. Containers? Virtual machines on EC2? You should know 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. Deep dive into potential services on AWS Academy. You may not have learned all 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! After this week your in-progress final report should have the deployment model and delivery model sections filled out.
Week 5	Cloud Architectures	This week we will study some common system architectures used in Cloud Computing. You will need to describe your architecture and compare and contrast to these architectures in your report. After this week your in-progress final report should have all the service section filled out and finalized.
Week 6 / Break / Week 7	DevOps & Infrastructure as Code	Your choices should all be made by now. It's time to get to work! Get coding and provisioning your project infrastructure. Learn and practice! Tenacity is required! Remember 5409 students you must deploy your term project with infrastructure as code and

		CI/CD so pay close attention here, make sure you are staying on track.
Week 8	Kubernetes	Use this week to learn some of the technologies your project will 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 begin full programming production.
Week 9	Security	Continue working hard! You've got this! Aim to reach 33% functional completeness by this date. Hey! You just learned about security in the cloud, fill out that section of your in-progress final report.
Week 10	Business considerations	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 the semester is done! Aim to reach 60% functional completeness by this date. Hey! You just learned business considerations for building software in the cloud, fill out that section of your in-progress final report! Holy cow it's done! Now you can focus 100% on completing the implementation.
Weeks 11 & 12	Microservices, case studies and discussion	This is crazy go time! Go, go, go, get that project 100% complete! Make sure you check when your final report and video presentation are due. We will not accept late submissions!