CSCI4145/5409 Adv. Topic in Cloud Computing – 2024 Summer

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 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, cloud architect, cloud engineer, cloud systems administrator, 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. If you select services that aren't listed here, please check with the instructor to determine which category they belong to before you proceed. Additionally, ensure that the services you choose are supported by the AWS Academy Learner Lab.

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 (RDS)

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):

- 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
- Infrastructure as code: build a CI/CD system to automatically build and deploy and provision your infrastructure via AWS CloudFormation

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 Supported 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.

We require all students to provision their infrastructure with AWS CloudFormation. All required cloud resources must be provisioned in a single step without human intervention.

Students are expected to learn CloudFormation on their own. As a 4th-year undergraduate or graduate student, you should be capable of independently learning Infrastructure as Code (IaC). **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

The term assignment represents a combined 25% of your final grade.

Final Written Report – 40%

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.
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. We require you to draw the architecture diagram using the free tool https://app.diagrams.net/. You should be able to learn how to use this tool by yourself.
- 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 Demonstration of CloudFormation (20%)

Given that it may not be efficient to demonstrate how your CloudFormation scripts provision your application and infrastructure during the one-on-one meeting, we request that you submit a video, which should be 5-10 minutes in duration, to showcase the complete provisioning of your application through CloudFormation. You can cut the part of the video where resources are being provisioned and then explain the services you used and a general flow of the services in your project.

One-on-One Video Meeting – 40%

The marker will schedule a one-on-one videoconference meeting with each student, lasting at about 10 minutes. At the beginning of the video, please ensure that your student ID card is visible in front of the camera and keep the camera active throughout the entire meeting. During this meeting, your goal is to showcase your working application and demonstrate the successful implementation of your project. This is where you validate that your project functions as described. We won't be assessing the code quality of individual services you've implemented; rather, our focus is on the final working product. You will be expected to present the back-end implementation, AWS configuration, and the full functionality of your system in its live, working state. This is your opportunity to proudly present your creation. The marker will ask questions for clarification regarding your accomplishments.