CSxxx or SExxx – Cloud Native Software Engineering
Dr. Brian Mitchell – bsm23@drexel.edu – Office: 1134

Syllabus

Couse Objectives
Introduction to various aspects of software design, development and architecture used to create modern cloud computing software products. Focus will be placed on covering software engineering concepts, techniques and technologies used to build and deploy applications that run at scale on cloud infrastructure. Key topics include cloud native software engineering concepts such as working with API-driven infrastructure, software design/architecture considerations for cloud native applications, and various modern technology stacks used in creating cloud software products.

This course does not require previous background in cloud computing but does assume the student has some background in software design, software architecture and software development.

Couse Lecture Format
This course will balance both theory and practical application. Most lectures will be structured with 2/3 of the content on cloud native software engineering concepts, with the remaining 1/3 being hands on tutorials on specific cloud native technologies.

Topics (Roughly by Week)
1. Introduction to cloud computing
    a. Cloud computing models – IaaS, PaaS, FaaS
    b. Fully managed cloud services
    c. Cloud infrastructure concepts – Regions, Availability Zones
    d. Cloud native application characteristics – Scalability, Elasticity, Resiliency, Automation
    e. The big 3 cloud providers – AWS, Azure, GCP
    f. **Tutorial**: Introduction to the Go programming language
2. Software Architectures for Cloud Native Application
    a. Event-based architectures
    b. Cloud Architecture Patterns for resiliency and scale
    c. **Tutorial**: Go programming continued, API frameworks for Golang
3. Designing Cloud Native Applications
    a. Domain Driven Design Composition
    b. Reference Architectures for Cloud Computing Products
    c. **Tutorial**: API Construction and Analysis – Request/Reply (REST) & Event Based APIs
4. API Construction and Deployment
    a. Best practices for API construction

    b. API Framework components – Routes, Middleware, Asynchronous actions
    c. Deep dive into creating Go-based APIs with Golang-Gin. Design, Deployment, Testing
    d. API Ecosystem – API Catalogs, API Gateways, API Security w/ oAuth
    e. **In Class Workshop**: Guided tour creating and developing two collaborating APIs
5. Containerization
    a. Docker Architecture Overview
    b. Container Architecture Overview
    c. Practices for Building, Deploying, and managing containers
    d. **Tutorial**: Docker
6. Container Orchestration
    a. Why containers must be orchestrated
    b. Container integration concepts
    c. Container architecture concepts
    d. Container networking
    e. **Tutorial**: Container Orchestration – Docker Compose
7. Kubernetes – Industrial Strength Container Orchestration
    a. Kubernetes Overview
    b. Kubernetes Architecture
    c. **Tutorial**: Kubernetes Part 1
8. Kubernetes – Industrial Strength Container Orchestration
    a. Kubernetes Architecture Continued
    b. Kubernetes offerings in the cloud
    c. Managing Kubernetes Deployments – e.g., Helm Charts
    d. **Tutorial:** Kubernetes Part 2
9. Function as a Services
    a. FaaS concepts
    b. FaaS tradeoffs vs Containers
    c. **Tutorial**: OpenFaaS/ KNative
10. Additional Considerations for Cloud Native Software Engineering
    a. Security Policy Management - IAM
    b. Data Considerations for Cloud Native Applications
    c. Walkin topics for discussion
    d. **In Class Demo**: Cloud Database Setup

Expected Learning

By the end of this course the student should be expected to have a solid grasp on:

- Basic knowledge of programming languages used in modern cloud platforms – this class will focus on Go – aka GoLang
- Software architectures, frameworks and patterns used to construct resilient and scalable microservices and modern API based software products
- Working with Popular cloud native services such as Docker, Kubernetes, and FaaS offerings
- Cloud infrastructure architecture – regions, availability zones, edge locations

Textbook

There is no required textbook for this course. Materials will be provided by the instructor in Blackboard, and students will be expected to review online materials (websites, Blogs, YouTube videos) provided by the instructor

Assignments

This course will balance hands on application with more traditional homework assignments. A course project will be assigned with multiple deliverables that follow the course lectures. In addition, there will be approximately 5-6 written assignments involving reviewing material provided by the instructor (research papers, videos, etc) and answering questions based on specific prompts.

Grading Breakdown

50% homework assignments
50% projects
Note that this class will not have a midterm or a final

Time Commitment and Additional Help Resources

The time commitment required to successfully complete this course will vary significantly depending on students experience with software development toolchains, programming languages and build tools. If you have any questions or doubts about your readiness to take this course please setup time with the instructor to discuss.

Grades

A+ (98-100); A (93-97); A- (90-92)
B+ (87-89); B (83-86); B- (80-82)
C+ (77-79); C (73-76); C- (70-72)
D (60-69)
F (< 60)

University Policies:

This course follows university, college, and department policies, including but not limited to:

- Academic Integrity, Plagiarism, Dishonesty and Cheating Policy: http://www.drexel.edu/provost/policies/academic_dishonesty.asp
- Student Life Honesty Policy from Judicial Affairs: http://www.drexel.edu/provost/policies/academic-integrity
- Students with Disability Statement: http://drexel.edu/oed/disabilityResources/students/
- Course Add/Drop Policy: http://www.drexel.edu/provost/policies/course-add-drop
- Course Withdrawal Policy: http://drexel.edu/provost/policies/course-withdrawal

- Department Academic Integrity Policy: http://drexel.edu/cci/resources/current-students/undergraduate/policies/cs-academic-integrity/
- Drexel Student Learning Priorities: http://drexel.edu/provost/assessment/outcomes/dslp/
- Office of Disability Resources: http://www.drexel.edu/ods/student_reg.html

Students requesting accommodations due to a disability at Drexel University need to request a current Accommodations Verification Letter (AVL) in the ClockWork database before accommodations can be made. These requests are received by Disability Resources (DR), who then issues the AVL to the appropriate contacts. For additional information, visit the DR website at drexel.edu/oed/disabilityResources/overview/, or contact DR for more information by phone at 215.895.1401, or by email at disability@drexel.edu.

Couse Change Policy

The instructor may, at their discretion, change any part of the course during the term, including assignments, grade brakdowns, due-dates, and the schedule. Such changes will be communicated to students via the course web site Announcements page. This page should be checked regularly and frequently for such changes and announcements. Other announcements, although rare, may include class cancellations and other urgent announcements will be communicated via the course discord channel.