e516: Engineering Cloud Computing

Indiana University

Gregor von Laszewski

2018-05-10

Contents

1 e516: Engineering Cloud Computing

- Lecture Notes: https://github.com/cloudmesh-community/book/blob/master/vonLaszewski-cloud.epub
- Piazza: https://piazza.com/iu/fall2018/516
- Indiana University
- Faculty: Dr. Gregor von Laszewski (laszewski@gmail.com)
- Credits: 3
- Hardware: You will need a computer to take this class, a phone, tablet, or chrome book is not sufficient.
- Prerequisite(s): Knowledge of a programming language, the ability to pick up other programming languages as needed, willingness to enhance your knowledge from online resources and additional literature. You will need access to a "modern" computer that allows using virtual machines and/or containers. If such a system is not available to you can also use cloud vms we provide and if you opt to do so one or more Raspberry's PI. All residential students will have access to a total of 200 Raspberry PIs. Online students can opt to purchase one or more based on our materials list that we will release throughout the semester. All students will have access to a cloud.
- This page is maintained and updated at e516: Engineering Cloud Computing
- Course Description URL: https://github.com/cloudmesh-community/book/blob/master/chapters/class/e516-engineering-cloud-computing.md
- Registrar information and Other related classes

This is an introductory class. In case you like to do research and more advanced topics, consider taking an independent study with Dr. von Laszewski.

1.1 Course Description

This course covers basic concepts on programming models and tools of cloud computing to support data intensive science applications. Students will get to know the latest research topics of cloud platforms, parallel algorithms, storage and high level language for proficiency with a complex ecosystem of tools that span many disciplines.

1.2 Course Objectives

The course has the following objectives:

- Provide a basic introduction to cloud computing
- Introduce the concept of cloud data centers
- Get familiar with cloud infrastructure as a Service such as OpenStack, Azure, or AWS
- Get familiar with cloud infrastructure such as Docker and Kubernetes
- Program cloud services
- Understand the differences between virtual machines and containers
- Develop sophisticated programming language independent REST services
- Learn advanced programming models for clouds such as Map/Reduce, Messaging, and GraphQl
- Exploration of Go for cloud computing
- Demonstrate knowledge of clouds while developing a significant project
- Explore state-of-the-art cloud technologies and services while providing a section and summary and commenting on its use for the cloud
- Learn how edge computing is enhancing cloud services and infrastructure
- Learn how to set up a cloud based on using commodity hardware

1.3 Learning Outcomes

- Be able to explain the concepts of the cloud computing paradigm including its paradigm shift, its characteristics, and the advantages. Contrast them with the challenges and disadvantages.
- Be able to identify infrastructure and programming models needed to support real world applications.
- Be able to implement a real world application or deploy a cloud and its services.
- Be able to conduct sophisticated performance analysis of cloud services.
- Be able to communicate the results through tutorials, manual, and reports.
- Be able to work in a team to develop collaboratively software or contribute collaboratively to develop sections explaining how to use clouds.

1.4 Syllabus

The topics are subject to change.

	Dates	Unit	Title	Description
	Week 1	1	Introduction	Gregor von Laszewski↩
				Class summary↩
				Definition of Cloud Computing↩
	Week 2	2	Tools	**Tools and Services**
				- Virtual Box↩
				- Vagrant↩
				- Github↩
				- Linux↩
	Week 3	3	Python	**Python**
				- Introduction↩
				- Installation↩
				- Interactive Python↩
				- Editors↩
				- Basic Language Features↩
				- Modules↩
				- Data Management↩
				- Matplotlib↩
				- Cloudmesh Commandshell CMD5↩
				- OpenCV↩
				- Secchi Disk↩
	Week 4	4		Data Center
	Week 5	5	Architectures	- NIST Big Data Reference Architecture↩
				- Cloud Architectures ↩
				- NIST Big Data Reference Architecture↩

	Dates	Unit	Title	Description
	Week 6	6	Virtualization	Virtualization, Qemu, KVM, Virtual machines↪
			Virtualization I	- Qemu↪
	Week 7	7	Infrastructure	Infrastructure as a Service↪
				- Azure ↪
				- AWS ↪
				- OpenStack ↪
			Chameleon Cloud	- Chameleon Cloud
				- Resources↪
				- Hardware↪
				- Charge↪
				- Quick start↪
				- KVM user guide↪
				- CLI↪
				- Horizon↪
				- Heat↪
				- Baremetal↪
				- FAQ↪
	Week 8	8	Virtualization II	Containers, Docker, Kubernetes
	Week 9	9	Programming	Python for Cloud Computing
				- Libcloud↪
				- Github as a Cloud Service↪
				- REST Services↪
				- Rest Services with OpenAPI↪
				- OpenAPI Spec↪
				- OpenAPI Codegen↪
	Week 10	10	Map/Reduce	Map/Reduce, Hadoop, Spark, and others

	Dates	Unit	Title	Description
	Week 11	11	Messaging	Messaging
	Week 12	12	Messaging	- MQTT↪
				- Graphql
	Week 13	13	Go	Go for the Cloud
	Week 14,15	14	Edge Computing	Edge Computing and the Cloud
	Week 16	15	Project Selection	Project selection for inclusion in the Proceedings

Students need only to do one project. The project is conducted thought the entire semester.

- Example chapters are indicated with ↪
- Dates may change as the semester evolves
- The project is a long term assignment (and are ideally worked on weekly by residential students). It is the major part of the course grade.
- Sections and chapters prepare you for documenting a technical aspect related to cloud computing. It is a preparation for a document that explains how to execute your project in a reproducible manner to others.
- Additional lectures will be added that allow easy management of the project. These lectures can be taken any time when needed.

1.5 Assessment

This course is focusing on the principal “Learning by Doing” which is assessed by simple graded and non-graded activities. The assessment may include comprehension of the material taught, programming assignments, participation in online discussion forums, or the contribution of additional material to the class showcasing your comprehension.

The comprehension is also measured by the development of a chapter and sections in markdown that can be distributed and replicated to other students. This is done in preparation for the project that must include a simple deployment and runtime instruction set.

The main deliverable of the class is a project. The project is assessed through the following artifacts:

1. Deployment and install instructions,

2. Project report (typically 2-3 pages per month, tutorial and chapters can be reused if possible),
3. Working project code that can be installed and executed in reproducible manner by a third party
4. Code developed by the project team distributed in github.com
5. Project progress notes checked into github throughout the semester. Each week the project progress is reported and will be integrated into the final grade.
6. Discussions or progress reports with the instructors can be conducted on online and residential class hours.

The grade distribution is as follows

- 10% Comprehension Activities
- 10% Sections
- 10% Chapter
- 70% Project

As the project is the main deliverable of the course it is obvious that those starting a week before the deadline will not succeed in this class. The project will take a significant amount of time and fosters the principal of “Learning by Doing” at all stages throughout the semester.

The class will not have a regular midterm, but it is expected that you have worked on your project and can provide a snapshot of the progress outlining the goals of the project and how you will achieve these goals till the end of the semester.

The final Project is due 3 weeks before the semester ends. Issues with your project ought to have been discussed before this deadline with the TA’s The TAs will in the next 3 weeks go over the projects and evaluate mayor and minor issues that you may be able to fix without penalty. Larger changes will receive a grade penalty. The last fix (upon approval) possible will be 2 weeks before semester end without penalty. It is recommended that you are available till the last day of class.

1.6 Assignments

	Dates	Unit	Title	Description
	due Week 8	A1	Sections	Contribute significant sections. Do not develop redundant or duplicated content.
	due Week 8	A2	Chapter	Contribute a significant chapter that may use your section to the class documentation. Do not develop redundant or duplicated content.
	Due Week 13	A3	Project Type A	Build a cloud out of Raspberry PIs

Dates	Unit	Title	Description
			Kubernetes, Hadoop, SLURM + OpenAPI Service,
		Project Type B	Build a Significant OpenAPI REST Service
		Project Type C	Build an Edge Service Interfacing with a Cloud
		Project Type D	Contribute to the new Cloudmesh code
		Project Type E	Your own Project Type A, B, C, D [upon approval)

1.7 Calendar

Assignment #	Event		Date
	Full Term		16 Weeks
	Begins		Week 1
1, 2	**Bio, Notebook**	assigned	Week 1
1, 2	**Bio, Notebook**	due	Week 2
3	**Sections**	assigned	Week 3
4	**Chapter**	assigned	Week 3
5	**Project**	selection or proposal	Week 4
5	**Project**	Draft	Week 8
3	**Section**	due	Week 10
4	**Chapter**	due	Week 10
5	**Project**	due	Week 12
5	**Project** (no penalty)	improvements	Week 13
5	**Project** (with penalty)	improvements	Week 14
	Final Deliverables due		Week 14
	Grading		Week 15, 16
	Ends		Week 16

- TA's must be available till all grades have been submitted.
- Bio: a formal 3 paragraph Bio
- Notebook: a markdown in which you record your progress of this class in bullet form
- All times are in EST
- Dependent on class progress Comprehension Assignments may be added

1.8 Incomplete

Please see the university regulations for getting an incomplete. However, as this class uses state-of-the-art technology that changes frequently, you must expect that an incomplete may result in significant additional work on your behalf as your project may need significant updates on infrastructure, technology, or even programming models used. It is best to complete the course within one semester.