

Cloud Computing Engineering

e516

Gregor von Laszewski

Editor

laszewski@gmail.com

<https://github.com/cyberaide/bookmanager>

August 23, 2019 - 11:14 AM

Created by Cloudmesh & Cyberaide Bookmanager, <https://github.com/cyberaide/bookmanager>

CLOUD COMPUTING ENGINEERING

Gregor von Laszewski

(c) Gregor von Laszewski, 2018, 2019

CLOUD COMPUTING ENGINEERING

1 PREFACE

[1.1 Disclaimer](#) 

[1.1.1 Acknowledgment](#)

[1.1.2 Extensions](#)

2 SYLLABUS

[2.1 Syllabus Engineering Cloud Computing](#) 

[2.1.1 Instructor](#)

[2.1.2 Audience](#)

[2.1.3 Course summary](#)

[2.1.4 References](#)

[2.1.5 Tools](#)

[2.1.6 Course Schedule](#)

[2.1.7 Attendance](#)

[2.1.8 Assignments](#)

[2.1.9 Project Examples](#)

[2.1.10 Statement on Academic Misconduct](#)

3 OVERVIEW

[3.1 Communication](#) 

[3.1.1 Piazza](#)

[3.1.2 Class Resources](#)

[3.1.3 Online Meeting](#)

[3.1.4 Assignments](#)

[3.1.5 Post Your Bio](#)

[3.1.6 Evolving Document](#)

[3.1.7 Books](#)

[3.1.8 You Get Credit for Improving the Books](#)

4 APPENDIX

[4.1 ePub Readers](#) 

[4.2 Corrections](#) 

[4.3 Contributors](#) 

[4.4 Creating the ePubs from source](#) 

[4.4.1 Docker](#)

[4.4.1.1 Using OSX](#)

[4.4.1.2 Using the Docker Image](#)

[4.4.2 Using the Native System](#)

[4.4.3 Using Vagrant](#)

[4.4.4 Creating a Book](#)

[4.4.5 Publishing the Book to GitHub](#)

[4.4.6 Creating Unpublished Drafts](#)

[4.4.7 Creating a New Book](#)

[4.4.8 Managing Images](#)

[4.5 Notation](#) 

[4.5.1 Figures](#)

[4.5.2 Hyperlinks in the document](#)

[4.5.3 Equations](#)

[4.5.4 Tables](#)

[4.6 Updates](#) 

[5 REFERENCES](#)

1 PREFACE

Fri Aug 23 11:14:08 EDT 2019 

1.1 DISCLAIMER

This book has been generated with [Cyberaide Bookmanager](#).

Bookmanager is a tool to create a publication from a number of sources on the internet. It is especially useful to create customized books, lecture notes, or handouts. Content is best integrated in markdown format as it is very fast to produce the output.

Bookmanager has been developed based on our experience over the last 3 years with a more sophisticated approach. Bookmanager takes the lessons from this approach and distributes a tool that can easily be used by others.

The following shields provide some information about it. Feel free to click on them.

     

1.1.1 Acknowledgment

If you use bookmanager to produce a document you must include the following acknowledgement.

“This document was produced with Cyberaide Bookmanager developed by Gregor von Laszewski available at <https://pypi.python.org/pypi/cyberaide-bookmanager>. It is in the responsibility of the user to make sure an author acknowledgement section is included in your document. Copyright verification of content included in a book is responsibility of the book editor.”

The bibtex entry is

```
@Misc{www-cyberaide-bookmanager,  
author = {Gregor von Laszewski},
```

```
title =    {{Cyberaide Book Manager}},  
howpublished = {pypi},  
month =    apr,  
year =     2019,  
url={https://pypi.org/project/cyberaide-bookmanager/}  
}
```

1.1.2 Extensions

We are happy to discuss with you bugs, issues and ideas for enhancements.
Please use the convenient github issues at

- <https://github.com/cyberaide/bookmanager/issues>

Please do not file with us issues that relate to an editors book. They will provide you with their own mechanism on how to correct their content.

2 SYLLABUS

2.1 SYLLABUS ENGINEERING CLOUD COMPUTING

Learning Objectives

- Get a quick overview what the class is about
-

2.1.1 Instructor

- Gregor von Laszewski laszewski@gmail.com
- Office hours: By appointment

2.1.2 Audience

- We recommend you know one programming language.
- Knowledge of python is of advantage but not required. Python is easy.

2.1.3 Course summary

This class will introduce you to state-of-the-art cloud computing concepts and engineering approaches. This will include virtual machines, containers and Map/Reduce. The course will have a Lab in which you can practically explore these concepts. You will for example have the option to create a cloud as part of this course and explore cloud computing tools and frameworks.

2.1.4 References

The course does not have required readings. We will provide the following references as pointers to what we will discuss:

2.1.5 Tools

You will be required to have a computer to log into the cloud. We will give you access to an OpenStack cloud. Access to Azure, AWS, Google and others can be achieved through their free tier,

2.1.6 Course Schedule

Week	References
1	Course Introduction
2	Cloud Data Centers
3	Python for Cloud Computing, Start of Project Selection
4	Cloud Architectures
5	Virtualization I - OpenStack
6	Virtualization II - AWS, Azure, Google
7	Multi Cloud Environment
8	Cloud Technology Presentation - Project Review
9	Containers - Docker, Kubernetes
10	Map Reduce
11	Messaging
12	REST
13	GO
14	Project Work
15	Projects Due
16	Project Improvements

For each of the topics you will find one or more relevant chapters or sections in our online book.

2.1.7 Attendance

Attendance accounts for 10% of your final grade. If you need to skip class for any reason, you need to notify the instructor and TAs

2.1.8 Assignments

This course will not have exams. Instead, we have the following graded assignment categories:

- Lab Assignments (pass/fail) are assignments that will be conducted on a weekly basis. They will help you making sure you not only understand the material theoretically, but try them out.
- Cloud Technology Review and Examples (Graded): (This can be substituted for more programming in your project). This is a document about a Cloud technology that is not yet included in our handbook to introduce an interested party to it. It should not contain advertisements but be a rational description of the technology with examples that you have tried yourself. You will have to give a non plagiarized presentation and document about it.
- Project Assignments (Graded): The most important part of the class for which you will be working throughout the semester. Up to three students can work in a project. In case of group projects, the project deliverables are increased.

The project has three submissions that are spread throughout the semester. Each submission builds on the previous one and modifies previous documents into a consistent project report and documentation for your project.

- Project Outline

A description of what your project is about and how it relates to cloud computing and address:

- What is the problem you try to address?
- What are you doing to address this problem?
- How are you addressing this problem?
- What is the architecture that addresses the problem that you will implement?

- Code and Documentation Review

- You will be asked to have a meeting with the TA's and/or instructor to showcase your code and have at least one review prior to your final

submission.

- This will usually take place through the Lab hours on regular basis.
 - A first project discussion must have been done at least once at midterm time.
- Final Project Submission
 - All code and documentation must be checked into GitHub well before the semester is over. This allows us to give you feedback for improvements.

Please note that the syllabus is subject to change. Students in this class often come from a wide variety of backgrounds and experiences. As such, the instructor reserves the right to change the content of the course to accommodate the needs and expectations of the students.

2.1.9 Project Examples

- <https://cloudmesh.github.io/cloudmesh-manual/projects/index.html>

2.1.10 Statement on Academic Misconduct

Students will be expected to uphold and maintain academic and professional honesty and integrity as outlined in the Code of Student Rights, Responsibilities, and Conduct. Cases of academic misconduct will be handled according to the student disciplinary procedures described in IU's policies.

3 OVERVIEW

3.1 COMMUNICATION

3.1.1 Piazza

This class uses piazza for communication. It is your responsibility to enroll in the Piazza for the class. A link to our Piazza is provided in CANVAS. Piazza works just like a forum in which you can ask questions or post notes. In case of a question STudent sand teachers can formulate answers. A thread system in the question can be used to gather the answer.

- <https://piazza.com/iu/fall2019/e516fall19>

You will get a grade for participation in the class discussions on Piazza. Please note that we do not recommend you go from CANVAS to piazza as CANVAS has a bad layout and in fact does not allow you to see the Piazza Hyperlinks. Instead go to Piazza directly.

3.1.2 Class Resources

All class material will be posted at

- <https://piazza.com/iu/fall2019/e516fall19/resources>

3.1.3 Online Meeting

If you are an online student:

A poll has been posted that you must fill out before Friday of the first week of the semester for possible meeting times.

- <https://piazza.com/class/jzkfveoqwri3e4?cid=8>

The link to the meeting information is at

- <https://piazza.com/class/jzkfveoqwri3e4?cid=7>

3.1.4 Assignments

All Assignments are links in the piazza resource page as a simple link to CANVAS in the resources page:

- <https://piazza.com/iu/fall2019/e516fall19/resources>

3.1.5 Post Your Bio

To assure that you have access to Piazza and can post to it, please, post a formal Bio.

3.1.6 Evolving Document

This document is improved throughout the semester with weekly activities. We recommend that you regular download a new version. Be aware that some browsers or ePub readers may cache the previous version. Thus make sure to check out the newest version.

3.1.7 Books

This class does not use books from publishers as cloud computing is constantly changing and by the time you place the purchase order, the book may already be outdated. Instead we have prepared online books that are constantly evolving.

We like to illustrate on a simple example on what happened to us in the past. We prepared with great detail information about Azure. However throughout the semester the way on how to interact with it was changed dramatically. Now if you would have had a book, the entire chapter about Azure would have been outdated. However, within less than a week, we were able to replace the content so everyone in class can benefit from our updates very quickly. Not only that, because our material is online you will likely find an updated version in future even after your class is completed. No printed text book can provide this service.

We have the following books in ePub and PDF available. The ePub version is our preferred version:

- [List of Books](#)

The weekly Lecture notes are contained at

- [e516: Cloud Computing Engineering](#)

This page will bring you to the following books that we use as part of e516. It will also clearly specify which assignments you need to do as part of the weekly Lab activities.

- [Cloud Computing](#)
- [Linux](#)
- [Python](#)
- [Markdown](#)

3.1.8 You Get Credit for Improving the Books

Please note that improving the document in GitHub via pull requests will get you credits. This is an easy opportunity for you to get an excellent grade.

4 APPENDIX

4.1 EPUB READERS

This document is distributed in ePub format. Every OS has a suitable ePub reader to view the document. Such readers can also be integrated into a Web browser so that when you click on an ePub it is automatically opened in your browser. As we use eBooks the document can be scaled based on the user's preferences. If you ever see a content that does not fit on a page we recommend you zoom out to make sure you can see the entire content.

We have made good experiences with the following readers:

- **macOSX:** [Books](#), which is a built-in eBook reader
- **Windows 10:** [Microsoft edge](#), but it must be the newest version, as older versions have bugs. Alternatively use [calibre](#)
- **Linux:** [calibre](#)

If you have an iPad or Tablet with enough memory, you may also be able to use them.

4.2 CORRECTIONS

The material collected in this document is managed in

- <https://github.com/cloudmesh-community/book/chapters>

In case you see an error or like to make a contribution of your own section or chapter, you can do so in GitHub via pull requests.

The easiest way to fix an error is to read the ePub and click on the cloud symbol in a heading where you see the error. This will bring you to an editable document in GitHub. You can directly fix the error in the web browser and create there a pull request. Naturally, you need to be signed into GitHub before you can edit and create a pull request.

As a result contributors and authors will be integrated automatically next time we compile the material. Thus even if you corrected a single spelling error, you will be acknowledged.

4.3 CONTRIBUTORS

Contributors are sorted by the first letter of their combined Firstname and Lastname and if not available by their github ID. Please, note that the authors are identified through git logs in addition to some contributors added by hand. The git repository from which this document is derived contains more than the documents included in this document. Thus not everyone in this list may have directly contributed to this document. However if you find someone missing that has contributed (they may not have used this particular git) please let us know. We will add you. The contributors that we are aware of include:

Anand Sriramulu, Ankita Rajendra Alshi, Anthony Duer, Arnav, Averill Cate, Jr, Bertolt Sobolik, Bo Feng, Brad Pope, Dave DeMeulenaere, De'Angelo Rutledge, Eliyah Ben Zayin, Eric Bower, Fugang Wang, Geoffrey C. Fox, Gerald Manipon, Gregor von Laszewski, Hyungro Lee, Ian Sims, IzoldaIU, Javier Diaz, Jeevan Reddy Rachepalli, Jonathan Branam, Juliette Zerick, Keith Hickman, Keli Fine, Mallik Challa, Mani Kagita, Miao Jiang, Mihir Shanishchara, Min Chen, Murali Cheruvu, Orly Esteban, Pulasthi Supun, Pulasthi Supun Wickramasinghe, Pukit Maloo, Qianqian Tang, Ravinder Lambadi, Richa Rastogi, Ritesh Tandon, Saber Sheybani, Sachith Withana, Sandeep Kumar Khandelwal, Silvia Karim, Swarnima H. Sowani, Tharak Vangalapati, Tim Whitson, Tyler Balson, Vafa Andalibi, Vibhatha Abeykoon, Vineet Barshikar, Yu Luo, ahilgenkamp, aralshi, bfeng, brandonfischer99, btpope, garbeandy, harshadpitkar, himanshu3jul, hrbahramian, isims1, janumudvari, joshish-iu, juaco77, karankotz, keithhickman08, mallik3006, manjunathsvan, qianqian tang, rajni-cs, rirasto, shilpasinhg21, swsachith, trawat87, tvangalapati, varunjoshi01, vineetb-gh, xianghang mi, zhengyili4321

4.4 CREATING THE EPUBS FROM SOURCE

In case you wish to create the ePUB from source, we have included this section.

However the easiest way is to use our docker container as described in [Section 4.4.1](#).

4.4.1 Docker

We recommend the docker creation method for

- Ubuntu
- Windows 10
- macOSX

4.4.1.1 Using OSX

The easiest way to create a system that can compile the book on macOS, is to use a docker container. To do so you will need to first install docker on macOS while following the simple instructions at

- <https://docs.docker.com/docker-for-mac/install/>

Once you have docker installed, you can follow the instructions in [Section 4.4.1](#).

4.4.1.2 Using the Docker Image

In case you have docker installed on your computer you can create ePubs with our docker image. To create that image by hand, we have included a simple makefile. Alternatively you can use our image from dockerhub if you like, it is based on ubuntu and uses our [Dockerfile](#).

First, you need to download the repository:

```
$ git clone https://github.com/cloudmesh-community/book.git  
cd book
```

To open an interactive shell into the image you say

```
$ make shell
```

Now you can skip to [Section 4.4.4](#) and compile the book just as documented

there.

Please note that we have not integrated pandoc-mermaid and pandoc-index at this time in our docker image. If you like to contribute them, please try it and make a pull request once you got them to work.

In case you want to create or recreate the image from our [Dockerfile](#) (which is likely not necessary, you can use the command

```
$ make image
```

4.4.2 Using the Native System

In case you like to use your native environment (which is typically faster than the container) you need to make sure you have an up to date environment.

Please note, that you must have at least Pandoc version 2.5 installed as earlier versions will not work. We recommend that you use Python version 3.7.4 to run the scripts needed to assamble the document. However eralier version of Python 3 may also work, but are not tested. You can check the versions with

```
$ pandoc --version  
$ python --version
```

4.4.3 Using Vagrant

In case you have installed vagrant on your computer which is available for macOS, Linux, and Windows 10, you can use our vagrant file to start up a virtual machine that has all software installed to create the ePub.

First, you need to download the repository:

```
$ git clone https://github.com/cloudmesh-community/book.git  
$ cd book
```

Next you have to create the virtual machine with

```
$ vagrant up
```

You can loginto the VM with

```
$ vagrant ssh
```

The book folder will be mounted in the VM and you can follow the instructions in [Section 4.4.1](#).

4.4.4 Creating a Book

Once you have decided for one of the methods, you can create a book.

To create a book, you have to first check out the book source from github with if you have not yet done so (for example if you were to use the docker container method):

```
git clone git@github.com:cloudmesh-community/book.git
```

Books are organized in directories. We currently have created the following directories

```
./book/cloud/  
./book/big-data-applications/  
./book/pi  
./book/writing  
./book/222  
./book/516
```

To compile a book go to the directory and make it. Let us assume you like to create the cloud book for cloud

```
$ git clone https://github.com/cloudmesh-community/book.git  
$ cd cloud  
$ make new
```

To view it you say

```
$ make view
```

After you have done modifications, you need to do one of two things. In case you add new images you need to use

```
$ make new
```

otherwise you can just use

```
$ make
```

The structure of the books is maintained in the yaml file `chapters.yaml`. You can add this chapter to the yaml file, but discuss this first with Gregor. In case you add a new chapter, you have to say

```
$ make clean  
$ make update  
$ make  
$ make view
```

4.4.5 Publishing the Book to GitHub



This task is only to be done by Gregor von Laszewski. You will not have to do this step.

To publish the book say

```
$ make publish
```

4.4.6 Creating Unpublished Drafts

Developers of the manual can modify the `Makefile` and locate the variable `DRAFT=` to add additional sections and chapters they work on, but should not yet been distributed with the main publication. Simply add them to the list and say

```
$ make draft  
$ make view
```

to create the draft sections only and view them.

To conveniently call them in a lazy fashion in a terminal you could use the following two aliases.

```
alias m='make; make view'  
alias d='make draft; make view'
```

This allows you to typ `m` for the main volume and `d` for the draft. Please note that all artifacts are written into the dest folder.

4.4.7 Creating a New Book

Let us assume you like to create a new book. The easiest way to start is to copy from an existing book. However, make sure not to copy old files in dest. Let us assume you like to call the book gregor and you copy from the 222 directory.

You have to do the following

```
$ cd 222  
$ make clean
```

```
$ cd ..  
$ cp -r 222 gregor
```

Now edit the file chapters.yaml and copy the section with `BOOK_222=` to `BOOK_gregor=`. Make modifications to the outline as you see fit.

Now you can create the book with

```
$ cd gregor  
$ make update  
$ make new
```

4.4.8 Managing Images

In case you have added images to the book, they must be on the same level as your contribution, but in a directory called images. E.g.

```
./chapters/cloud/mydocument.md  
./chapters/cloud/images/myimage.md
```

In the document the image is than refered to as

```
![My image caption](images/myimage.md){#fig:cloud-myimage}
```

The label `#fig:cloud-myimage` must be unique in all of the documents. While adding the directory cloud before the image name this is the case in our example.

4.5 NOTATION

The material here uses the following notation. This is especially helpful, if you contribute content, so we keep the content consistent.

if you like to see the details on how to create them in the markdown documents, you will have to look at the file source while clicking on the cloud in the heading of the Notation section ([Section 4.5](#)). This will bring you to the markdown tex, but you will still have to look at the [raw content](#) to see the details.

• Or  ![Github](images/github.png)

If you click on the • or  in a heading, you can go directly to the > document in github that contains the next content. This is > convenient to fix errors or make additions to the content. The cloud

will be automatically added upon inclusion of a new markdown file that includes in its first line a section header.

\$

Content in bash is marked with verbatim text and a dollar sign

`$ This is a bash text`

[1]

References are indicated with a number and are included in the > reference chapter [1]. Use it in markdown with > `[@las14cloudmeshmultiple]`. References must be added to the `references.bib` file in BibTex format.

O or ✘

Chapters marked with this emoji are not yet complete or have some issue that we know about. These chapters need to be fixed. If you like to help us fixing this section, please let us know. Use it in markdown with `:o2:` or if you like to use the image with `![No](images/no.png)`.



[REST 36:02](#)

Example for a video with the `![Video](images/video.png)` emoji. Use it in markdown with `![Video](images/video.png) REST 36:02](https://youtu.be/xjFuA6q5N_U)`



[Slides 10](#)

Example for slides with the `![Presentation](images/presentation.png)` emoji. These slides may or may not include audio.



[Slides 10](#)

Slides without any audio. They may be faster to download. Use it in markdown with `![Presentation](images/presentation.png) Slides 10](TBD)`.



A set of learning objectives with the emoji.



A section is release when it is marked with this emoji in the syllabus. Use it in markdown with .



Indicates opportunities for contributions. Use it in markdown with .



Indicates sections that are worked on by contributors. Use it in markdown with .



Sections marked by the contributor with this emoji when they are ready to be reviewed.



Sections that need modifications are indicated with this emoji .



A warning that we need to look at in more detail .



Notes are indicated with a bulb .

Other emojis

Other emojis can be found at <https://gist.github.com/rxaviers/7360908>. However, note that emojis may not be viewable in other formats or on all platforms. We know that some emojis do not show in calibre, but they do show in macOS iBooks and MS Edge

This is the list of emojis that can be converted to PDF. So if you like a PDF, please limit your emojis to

:cloud: ☁ :o2: O :relaxed: ☺ :sunny: ☼ :baseball: ⚾ :spades: ♠ :hearts: ♥ :clubs: ♣ :diamonds: ♦
:hotsprings: 🌊 :warning: ⚠ :parking: P :a: A :b: B :recycle: 🔍 :copyright: © :registered: ® :tm: ™
:bangbang: !! :interrobang: !? :scissors: ✂ :phone: ☎

4.5.1 Figures

Figures have a caption and can be referred to in the ePub simple with a number. We show such a reference pointer while referring to [Figure 1](#).



Figure 1: Figure example

Figures must be written in the md as

```
![Figure example](images/code.png){#fig:code-example width=1in}
```

Note that the text must be in one line and must not be broken up even if it is longer than 80 characters. You can refer to them with `@fig:code-example`. Please note in order for numbering to work figure references must include the `#fig:` followed by a unique identifier. Please note that identifiers must be really unique and that identifiers such as `#fig:cloud` or similar simple identifiers are a poor choice and will likely not work. To check, please list all lines with an identifier such as.

```
$ grep -R "#fig:" chapters
```

and see if your identifier is truly unique.

4.5.2 Hyperlinks in the document

To create hyperlinks in the document other than images, we need to use proper markdown syntax in the source. This is achieved with a reference for example in sections headers. Let us discuss the reference header for this section, e.g. Notation. We have augmented the section header as follows:

```
# Notation {#sec:notation}
```

Now we can use the reference in the text as follows:

```
In @sec:notation we explain ...
```

It will be rendered as: In [Section 4.5](#) we explain ...

4.5.3 Equations

Equations can be written as

```
$$a^2+b^2=c^2$$ {#eq:pythagoras}
```

and used in text:

$$a^2 + b^2 = c^2 \quad (1)$$

It will render as: As we see in [Equation 1](#).

The equation number is optional. Inline equations just use one dollar sign and do not need an equation number:

```
This is the Pythagoras theorem: $a^2+b^2=c^2$
```

Which renders as:

This is the Pythagoras theorem: $a^2 + b^2 = c^2$.

4.5.4 Tables

Tables can be placed in text as follows:

```
: Sample Data Table {#tbl:sample-table}
```

```
x y z
```

```
--- --- ---  
1   2   3  
4   5   42
```

As usual make sure the label is unique. When compiling it will result in an error if labels are not unique. Additionally there are several md table generators available on the internet and make creating table more efficient.

4.6 UPDATES

As all documents are managed in github, the list of updates is documented in the commit history at

- <https://github.com/cloudmesh-community/book/commits/master>

In case you do a lecture withus we recommend that you download a new version oof the ePub every week. This way you are typically staying up to date. You can check the commit history and identify if the version of the ePub is older than the committed version, if so we recommend that you download a new version.



We typically will not make announcements to the class as the GitHub commit history is sufficient and you are responsible to monitor it as part of your class activities.

5 REFERENCES



- [1] G. von Laszewski, F. Wang, H. Lee, H. Chen, and G. C. Fox, “Accessing Multiple Clouds with Cloudmesh,” in *Proceedings of the 2014 acm international workshop on software-defined ecosystems*, 2014, p. 8 [Online]. Available: <https://github.com/cyberaide/paper-cloudmesh/raw/master/vonLaszewski-cloudmesh.pdf>