Association for Information Systems
AIS Electronic Library (AISeL)

BLED 2016 Proceedings | BLED Proceedings

2016

Teaching Software Engineering in the Cloud: Applying Cloud Computing Services in E-Business Education

Markus Grüne
Frankfurt University of Applied Sciences, Germany, magruene@fb3.fra-uas.de

29th Bled eConference

Digital Economy

June 19 - 22, 2016; Bled, Slovenia

Teaching Software Engineering in the Cloud: *Applying Cloud Computing Services in E-Business Education*

Prof. Dr. Markus Grüne

Frankfurt University of Applied Sciences, Germany

magruene@fb3.fra-uas.de

Abstract

Teaching Business Information Systems topics at academic institutions can profit a lot from applying the theoretical knowledge in hands-on exercises in- and outside the class room employing standard software. Most software vendors provide academic licenses for their products which can be utilized by academic institution. The installation and maintenance of software products especially at Universities of Applied Sciences is usually not feasible due to staffing, knowledge, and funding restrictions. Cloud services lower these "barriers of entry". This paper provides a show case which demonstrates how cloud services can be combined in a course setting teaching "E-Business" in an undergraduate course at Frankfurt University of Applied Sciences. During exercises in a PC lab and on their own devices, the usage of Cloud services allowed students to gain practical experience in working with an industry standard collaborative code versioning system (git).

Keywords: E-Learning, Software Engineering, Code Management, Cloud Computing Services

1 Introduction

Teaching Software Engineering, e.g., in undergraduate courses at academia, regularly requires the installation of information systems, integrated development environments, databases and other technological infrastructure components to allow students to gain hands-on experience and build up expertise for the job market (Zhu, 2015). The work spent on self-hosting information systems and IT products is usually wasted. This article describes a way for teaching Software Engineering principles on Cloud services and contrasts this to the "traditional way" of self-hosting products for teaching. Unfortunately, Cloud services are not used frequently in teaching. We hope to provide a relevant show case in this article which can be extended to future courses at Frankfurt University of Applied Sciences and beyond. It extends known solutions (Cloud Services) to a general problem in IS teaching (technology management for teaching). In terms of Design Sciences Research, the research focuses on improvement and exaptation (Gregor & Hevner, 2013). Further research is required to generalize the observations.

2 Related Work and traditional IS Teaching Approach

Although Cloud Services have matured over the recent decades, only few papers have been published addressing the differences in efforts when using Cloud Services instead of self-hosted Information Systems. Cloud Services are being used by many universities for teaching IT topics such as cyber security analysis skills (Weiss et al., 2015), big data analysis (Rabkin, Reiss, Katz, & Patterson, 2013), or programming in Java (Hollingsworth & Powell, 2010). The article tries to show case the benefits of Cloud Services in teaching IS.

2.1 Traditional Teaching Approach – Self-hosting

Many software products that can be used in teaching, e.g., relational database management systems (DBMS) or business process management systems (BPMS), are available under academic licensing and can be freely installed and used in the university's hosting environment. Teachers and instructors using these products in the curriculum have to invest a considerable amount of time for administration of the software and hardware (self-administration) and for their own training as well as to invest in training of administrative IT staff to keep pace with technological changes. Although this time is not wasted, it would be more beneficial, if it were used for working with the students.

The following list shows sample aspects / areas that need to be addressed, reflecting the author's experience when self-administering and installing a software product:

- *Ordering of hardware*– Often processes for ordering the hardware are unknown to the instructor. The IT department may be split across different organizational units, e.g., one team at departmental level plus a central IT which executes the final purchasing requests.
- *Configuring the server machines* and *installing the operating system* – Configuring the server machines and the operating system requires specialized knowledge on hardware and the operating system itself. Hardware can be anything from a server running at the teacher's individual desk up to blades running in data centre racks. For large, computation-intensive architectures, it is usually required to prepare server images that can be easily added to a grid of servers.
- *Installing the software products and running regular patches* – The installation of the product often requires setting up of individual user roles. Regular patches are necessary to keep the software up-to-date and for closing security gaps. Procedures for removal of the software and all associated sensitive data need to be defined.
- *Integration of the product into the academic institution's network* –Institutions should keep the "training infrastructure" on nodes that are separated from the core campus

network to avoid adverse impact on the daily work of the staff and IT-based operations of the university.

- *Defining a security concept including firewalls, identity and access management for administrators and student users* – Universities in Germany are subject to laws and policies which enforce privacy and security of the students' personal data. The storage of private user data in the university's network can be seen as a major driver to not use Cloud-based solutions.
- *Penetration testing in case the software product will be Web-facing*
- *Operating the new product* including backup of data, reacting to downtimes. This includes maintenance and migration of data to new versions of the software.
- Finally u*sing the product within the course context* Students will be measured e.g. by automated grading versus manual grading of assignments.

2.2 Taking IT and Teaching into the Cloud

The overheads for self-hosting the IT products required for teaching in the academic institution's network as highlighted in section 1 are often prohibitive for the instructors. The usage of Cloud Computing Services (CCS) being available for numerous software products can help to reduce these overheads. According to (Bengel, Baun, Kunze, & Stucky, 2015), the "basic concepts as well as [its] general objectives" of cloud computing are "undisputed: Cloud computing uses virtualizatioM8rxBrn and the modern Web to dynamically provide resources of various kinds as services which are provisioned electronically. These services should be available in a reliable and scalable way so that multiple customers can use them".

2.2.1 Cloud Computing

Cloud Computing Services (CSS) are based on *utility computing* where users pay for the amount of services used and on *grid computing* which allows consumers to share hardware resources and add resources to the grid when required (Marinescu, 2013). CCS can be set up in private environments (*private clouds*) or in public environments for use by many institutions and individuals (*public clouds*), or combining both (*hybrid clouds*) (NIST, 2011). CCS are meeting international standards for security such as the standards proposed by the Cloud Security Alliance and promise to be as reliable as self-hosted software installations. Cloud computing services reach from simple server and file hosting solutions up to complex IT products such as Enterprise Resource Planning (ERP) systems. Large software vendors consider cloud computing knowledge as a valuable asset for students when searching for a job and have established programs for academic institutions (e.g. (IBM, 2016), (AWS, 2016), (Microsoft, 2016)).

2.2.2 Scenarios for Cloud Computing in Teaching

The scenarios for usage of CCS in teaching can range from using Cloud-hosted infrastructure (IaaS, mainly in terms of operating system) up to Software-as-a-Service (SaaS) solutions in which the product is used directly over the cloud for teaching (Pardeshi, 2014). In the second scenario, all installation, maintenance and operation of the software product lie in the responsibility of the cloud provider. In the course setting which is described below, public cloud services of type SaaS and IaaS were used in combination.

2.2.3 Using cloud-based Solutions in Teaching E-Business

The module "E-Business" is lectured each summer term at Frankfurt University of Applied Sciences (UAS) for students of "International Business Information Systems" (IBIS). It consists of a lecture which highlights major concepts of E-Business and two exercise tracks. In the first track underlying economic principles of E-Business (market types, production functions, etc.) are explained (VanHoose, 2014), whereas the second exercise track focuses on the application of software development practices for engineering E-Business applications (so-called "*programming exercises*").

During the programming exercises students are asked to collaboratively build a small software application in HTML5 and JavaScript after an introduction to software delivery lifecycle models (waterfall, agile, continuous integration) and collaborative development concepts (code versioning systems, git, branching). Approximately 100 students were registered for the programming exercises.

3 Setting up a Cloud-based Course Environment

Setting up a course environment required the combination of multiple CCS from different providers. The CCS listed in Table 1 were used and integrated to provide a consistent user experience to the students in the course.

CCS services selected based on their prospective ease of use. Especially the user administration within Bitbucket helped to keep the setup of the Integrated Development Environment (IDE) to a minimum as Bitbucket user credentials could be re-used for log in. Only for the creation of the working environment within the IDE students had to copy a URL and key in the Bitbucket password for cloning the central repository.

The Elastic Cloud Virtual Server setup was straight forward, only requiring the installation of the Jenkins server and the Apache Web Server. Code deployment for the students on the server was automatically triggered by Jenkins based on a Bitbucket hook that checked whether a new commit version was available in the master branch of the central repository.

The following paragraph describes a typical scenario for code development and deployment in the programming exercise.

3.1 Using cloud services for programming exercises

Figure 1 summarizes the standard scenario which was used for the development and deployment of HTML5 code in the course setting. For the setup of Cloud 9 for personal use, students opened their "dashboard" in Cloud 9 and cloned the central repository from Bitbucket (not shown).

Cloud service used	Usage scenario	Advantages and comments
Cloud9 (www.c9.io)	An integrated development environment (IDE) that can be used for developing software applications in major programming languages	Installation of an IDE could be avoided, both on the participants' computers as well as on the computers in the PC pool. Integrates nicely with github.org and bitbucket.org. Bitbucket accounts can be used for log in. User interface shows a code editor, a bash console and a preview browser for test running code. JavaScript / HTML 5 and major JavaScript frameworks are supported. Students can use one personal "workspace" for free.
Atlassian Bitbucket (bitbucket.org)	A central service for hosting code repositories. Code repositories can either be git or mercurial repositories.	Unlimited use under an academic license. Visualization of the check ins, branching and versions of the centrally hosted repositories. Credentials could be used also to automatically log in into Cloud 9. Some training material from Atlassian could be used for teaching git. Apart from code management, Bitbucket was used for the management of support tickets for the course. Instead of git, also mercurial could be used.
Amazon Elastic Cloud (aws.amazon.com)	Server hosting (Apache Tomcat) for the HTML5 applications that were built by the students. Many more services are available at AWS.	Elastic Cloud (EC2) is part of the Amazon Web Services (AWS) technology products portfolio. EC2 Linux server was used for hosting a web server and a Jenkins server for continuous deployment. For demonstration purposes, both servers were hosted on the same virtual machine. AWS supports academic initiatives by offering grants and subsidized trainings. The usage of AWS was very basic, but will be extended in future.

Table 1: Cloud services used in the course

Afterwards they had to follow the Gitflow workflow for the management of code changes (https://de.atlassian.com/git/tutorials/comparing-workflows/ gitflow-workflow). In Gitflow, all changes and enhancements to a software product are based on feature branches that are derived from the develop branch. Besides the develop branch, the master branch is used to keep the latest stable version of a software product that was released. Usually, changes that were merged from feature branches into the develop branch will be included in the master / release branch once the develop branch is merged into it. The students created feature branches before they coded. After finishing the coding, every student had to raise a pull request against the teacher who then confirmed that the student's changes could be merged into the develop branch

For publishing the code changes to the Web Server two steps were required by the teacher:

1. The pull requests had to be confirmed before the students were permitted to merge their features into the develop branch and
2. All code changes had to be merged into the master branch.

Jenkins took notice of any new versions of the master branch and deployed the changes to the Apache Web Server without any further manual interaction (continuous integration).

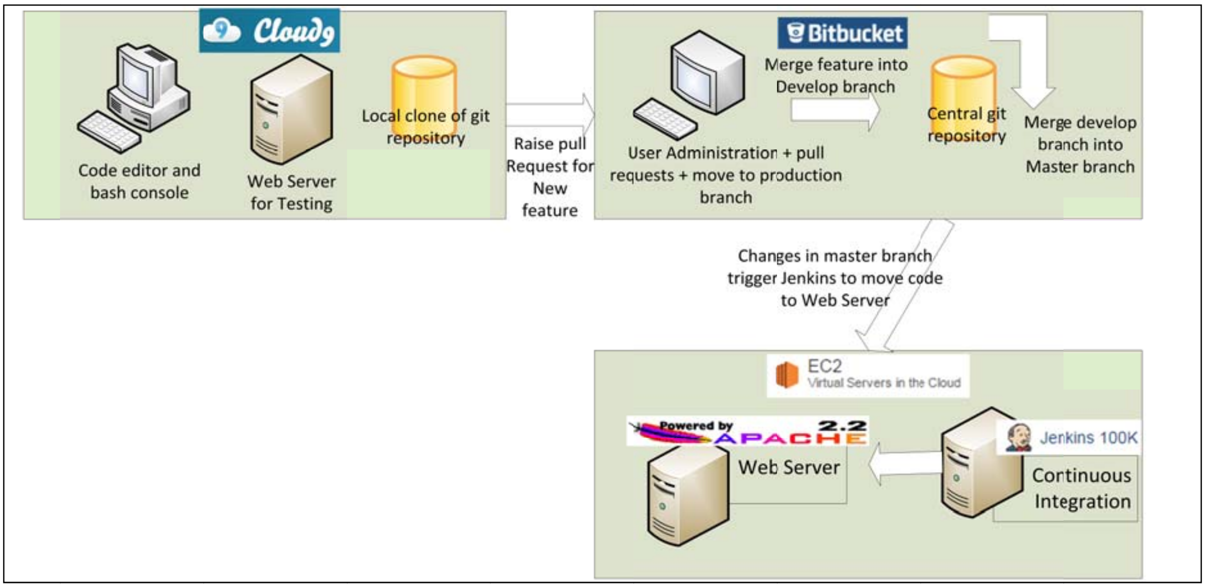

Figure 1: Orchestration of loud services to construct the learning environment

4 Results and Evaluation

The following paragraphs highlight benefits and limitations of cloud services in teaching. Further application scenarios and benefits for cloud computing in academic institutions can be found, e.g., in (Mokhtar, Ali, Al-Sharafi, & Aborujilah, 2013).

4.1 Benefits of using Cloud Service in Teaching

4.1.1 Independence of Devices and easy Re-installations

One of the main benefits for students working in the cloud is that they are not required to install any software on their personal devices. The only restriction is the availability of a current Web browser. The Cloud9 IDE was available with very small downtimes. After closing the browser session of the IDE, the users were able to log in exactly in the same environment setting as they left it. As all code was kept in a central repository, the users were able to re-install a cloud IDE workspace by cloning the central repository within less than 2 minutes. This was helpful as some student users misconfigured their frontends and it would have taken more time to undo the users' misconfigurations than simply setting up the IDE workspaces from scratch. At the time of writing, the Cloud 9 IDE does not support IE 10. All other browsers, irrespective of operating system, could be used. Students accessed their IDE from multiple devices (tablets, laptop computers, and the computers at the university's PC lab). Users also did not have to care about backups. As a positive side effect, the students did not raise any OS specific support cases regarding the installation of software, missing drivers, hardware issues, etc.

4.1.2 Independence from local IT Teams and Hardware Provisioning

The local IT team was not involved in the maintenance of the PC pool or for installing software for the specific purpose of the course. For future courses, the IT team will be involved for the creation of higher level services by integrating the IDE and the source code repository with Moodle, the E-Learning System at UAS. The independence from technical infrastructure also means that all organizational processes for the provisioning of hardware were irrelevant and therefore, the administrative overhead could be reduced to a bearable minimum. The goal of the course could be reached with minimal effort avoiding the overheads of a "traditional installation" that were described in section 1.

4.1.3 Public private partnerships

Cloud computing service providers like Amazon do have an interest in spreading the knowledge on their cloud products by engaging with academic institutions as this helps to train students in technologies that are used in industry and gain direct access to trained resources. The engagement varies from the provisioning of "grants" for infrastructure, subsidized training courses for teaching staff up to the contribution of courseware and whole training courses that can be offered also to students.

4.1.4 Benefits from the Architecture used in the Course

The environment set up for the course allowed for teaching current software engineering practices. It helped to demonstrate to the students how to work with source code and to learn agile development practices. For a full "simulation" of industry practices which would include staging, quality assurance and automated deployments, the architecture would need to be enhanced by adding applications.

4.1.5 Didactical Concept

First and foremost, CCS enable the students to play with technological concepts in a very lean way. Installation of software is not required which lowers the barriers of entry for many students focusing more on managerial areas within their studies of Business Information Systems than on technological aspects. Secondly, students could directly gain practical knowledge by learning collaborative coding in a team. The knowledge gained is important to understand the software delivery lifecycles in large IT organizations.

4.2 Limitations of using the Cloud in Teaching

4.2.1 Compliance Aspects

Using cloud services means that students had to create one account with Atlassian to use Bitbucket. In future courses, especially for course settings where the usage of cloud services is mandatory, compliance to the university's rules and regulations regarding private data needs to be confirmed. Compliance also needs to be checked when integrating cloud services with Moodle, the university's learning management and courseware system.

4.2.2 Budgeting and Cost Management

Extending the usage of cloud services means that the instructor needs to overlook the resulting costs. Before planning a cloud-based course, it is important to ensure that the budgets provided by the cloud service providers are not exceeded by the students' usage of resources to avoid disputes. (Gonzalez, Border, & Oh, 2013) describe sample cost of cloud services in a course on principles of system administration.

4.2.3 Administrative aspects and extension of the concepts

There are many areas for improvement. Firstly, the administration of users should be automated, e.g., by scripting. Also, scripts are required to monitor whether students provided the expected artifacts. Secondly, users currently work on the same code base and directory structure. It would be better, if every student worked within her own environment without the ability to see code in other directories. Docker might be a solution here to set up server environments based on scripting. It is also not necessarily required to set up a complete server for "playing with" HTML5. Thirdly, cloud services make sense as they scale up and down. For the course this means that all environments should be frozen and taken down once the course is complete. Freezing would allow keeping the final code state for documentation purposes.

5 Discussion

This short paper looked at Cloud Services for teaching an undergraduate course in software development practices. The usage of Cloud Services does not completely free the teacher from tasks such as user management and integrating services into a consistent experience / environment for the students and it incurs costs that need to be closely watched (Gonzalez et al., 2013). But it helps a lot in reducing the workload when trying out new software products as hardware provisioning is no longer required and a large number of software products that are required for Business Information System teaching are made available as a packaged version by cloud service providers. For Germany, data security aspects need to be considered, e.g., when maintaining personal student information in a Cloud. Regarding pedagogy, special attention should be paid to the varying knowledge on programming that the students show. In future courses, it makes sense to distinguish between different knowledge levels and embed a strategy for addressing these levels in a didactical concept. This article starts the observations. Additional research is required to establish a sound basis for evaluating specific Cloud Services for teaching and abstract the observations. Also, additional research is required to check on the "willingness" of teachers to employ Cloud Services. To broaden the view on Cloud Computing Services, a seminar was held in winter term 2015 with 25 students who conducted research on various cloud topics (legal aspects, architectures, business models, etc.)

6 Outlook / Conclusion

The application of cloud services has shown new perspectives and makes appetite for more. Cloud providers such as Amazon and Google continuously increase their portfolio. Cloud services can not only be used in teaching, but also in research and enhancing general services offered by data centers at universities. Although, migration to cloud-based solutions requires critical evaluation, the integration of cloud-based services into the IT portfolio would allow IT staff to focus more on service delivery than maintenance of infrastructure.

References

- **Books**

Christian Baun, Marcel Kunze, Jens Nimis & Stefan Tai. (2011). Cloud Computing. Berlin: Springer.

David D. VanHoose. (2014). E-commerce economics. Critical concepts in economics. London: Routledge.

Dan C. Marinescu. (2013). Cloud Computing – Theory and practice. Boston: Morgan Kaufmann / Elsevier.

- **Proceedings from conferences**

Carlos Gonzalez, Charles Border, and Tae Oh. (2013). Teaching in amazon EC2. In Proceedings of the 13th annual ACM SIGITE conference on information technology education, 11 – 13-Oct-2012 (149). New York: ACM. DOI:10.1145/2512276.2512322

Joel Hollingsworth and David J. Powell. (2010). Teaching web programming using the Google Cloud. In Proceedings of the 48th Annual ACM Southeast Regional Conference, 15 - 17-Apr-2010 (1). New York: ACM. DOI:10.1145/1900008.1900110

Shamsul Anuar Mokhtar, Siti Haryani Shaikh Ali, Abdulkarem Al-Sharafi, and Abdulaziz Aborujilah. (2013). Cloud computing in academic institutions. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, 17 – 19-Jan-2013 (1-7). New York: ACM, 1–7. DOI:10.1145/2448556.2448558

Richard S. Weiss, Stefan Boesen, James F. Sullivan, Michael E. Locasto, Jens Mache, and Erik Nilsen. (2015). Teaching Cybersecurity Analysis Skills in the Cloud. In Proceedings of the

46th ACM Technical Symposium on Computer Science Education, 04 - 07-Mar-2015 (332-337). New York: ACM. DOI:10.1145/2676723.2677290

Weiying Zhu. (2015). Hands-On Network Programming Projects in the Cloud. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 04 - 07-Mar-2015 (326-331). New York: ACM. DOI:10.1145/2676723.2677257

- **Journal Articles**

Ariel Rabkin, Charles Reiss, Randy Katz, and David Patterson. (2013). Using clouds for MapReduce measurement assignments. Trans. Comput. Educ. 13 (1), 1–18. DOI: 10.1145/2414446.2414448

Shirley Gregor and Alan R. Hevner. (2013): Positioning and presenting Design Science Research for maximum Impact. MIS Quarterly 37 (2), 337-355.

Vaishali H. Pardeshi. (2014). Cloud Computing for Higher Education Institutes: Architecture, Strategy and Recommendations for Effective Adaptation. Procedia Economics and Finance. 11, 589–599. DOI:10.1016/S2212-5671(14)00224-X

- **Web pages**

Amazon. (2016). Amazon Web Services (AWS) Educate Program. Retrieved 03-Mar-2016, from http://aws.amazon.com/ de/education/awseducate/

IBM. (2016). IBM Academic Initiative. Retrieved 03-Mar-2016, from https://developer .ibm.com/academic/

Microsoft. (2016). Microsoft Azure in teaching. Retrieved 03-Mar-2016, from http://azure.microsoft.com/en-us/community/education/

- **Standards**

NIST - National Institute of Standards and Technology. (2011). NIST Cloud Computing Reference Architecture. NIST SP 500-292. http://www.nist.gov/customcf/ get_pdf.cfm?pub_id=909505.