Developing in the Cloud

Aftab Iqbal Michael Hausenblas
Stefan Decker

DERI Technical Report 2012-01-09
January 2012

DERI Galway
IDA Business Park
Lower Dangan
Galway, Ireland
http://www.deri.ie/

DERI Technical Report
DERI Technical Report 2012-01-09, January 2012

Developing in the Cloud

Aftab Iqbal[1] Michael Hausenblas[1] Stefan Decker[1]

Abstract. With cloud computing as an infrastructure model maturing, first efforts around cloud-based software development are emerging. In this model, a cloud-based environment enables developers to edit, test and deploy Web applications through a Web browser. We consider this tool stack to be disruptive in its nature, changing the way software development is being carried out and opening up new opportunities for service providers as well as (teams of) developers. In this report we review the state of the art of cloud-based software development environments—in particular browser-based development environments—and discuss their features as well as related challenges and opportunities.

Keywords: cloud computing, agile software development, collaborative software development, integrated development environment.

[1]DERI, National University of Ireland, Galway, Ireland. firstname.lastname@deri.org

Acknowledgements: This work has been carried out in the Linked Data Research Centre (LiDRC) and has been supported by Science Foundation Ireland under Lion 2 and the European Commission's FP7 Support Action LOD-Around-The-Clock (LATC), project no. 256975, Intelligent Information Management (ICT-2009.4.3).

Contents

1 Introduction

Nowadays, Web applications are not launched in the classical sense anymore—they are continuously released; with every refresh in the browser one has potentially a new version of the application at hand (Fig. 1). On the Web, we continuously integrate and constantly iterate, take, for example, the roll-out of new features in Google's Office suite, Twitter's Web front-end or the changes to the Facebook UI.

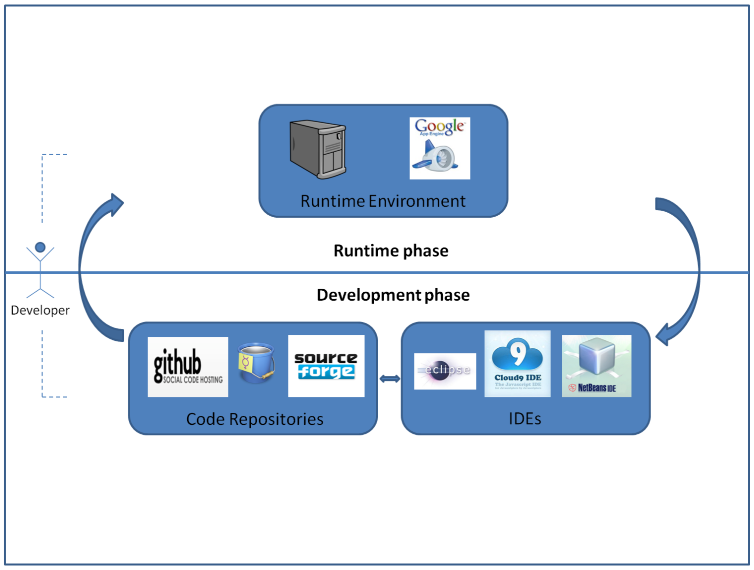


Figure 1: Continuous Integration and Deployment of Software Appplications

Software developers use project management tools, development frameworks, databases, etc., in their day-to-day software development tasks that help overcoming most obstacles, upfront. Some of these tools are primarily desktop-based (for example, the Eclipse IDE) while others are mostly Web-based (like Github). Making all these tools work together can be a pain, because each tool serves a specific purpose and there is a lack of interoperability among the tools, especially concerning data import and export.

Managing activities within a project using a variety of software tools make the software development process over the last years look quite different than it used to be. Developers are geographically distributed, they are agile and like to collaborate on different projects with other developers across the world. Code forges are a good example of global software development. It has gained a lot of attraction from the public and the software engineering community over the past decade. The success of these forges is highly dependent on the infrastructure provided to the developers and users in order to collaborate with each other [ST09] which helps increasing transparency, feedback, trust and tracking of dispersed software developers. They provide support for many software tools such as discussion forums, mailing lists, bug tracking systems, versioning systems, etc. Providing collaborative tools support in code forges promotes distributed software development. Imagine a group of developers would like to start a new project, they are not required to setup the necessary Web-based software tools internally because hosting or creating a project on a code forge automatically enables most of these software tools to the developers in order to collaborate with each other on the project.

Besides hosting of the project's source code and its related artefacts on the Web, the typical developer still carries out the actual development on her own machine using desktop-based environments. Imagine a developer who would like to work on multiple projects using different programming and database environments. She is required to setup and configure the necessary desktop-based software tools such as an Integrated Development Environment (IDE) before starting to contribute to a particular project. A developer who joins the team is also required to install and configure all the necessary software tools before being a productive member to the project. Last but not least, the transition to a new developer machine requires again installation and configuration of IDEs or development frameworks, which can turn out to be time-consuming.

2 Cloud-based Software Development Stack

With the emergence of cloud computing infrastructure, small and large corporations alike have started to move towards hosting their data, software applications, operational communication networks, etc. on the large-scale server farms. Time and cost benefits are the leading motivating factors and measures for any business accessing tools and services via the cloud [Wil]. This whole process is rapid, uncomplicated and does require only little end-user knowledge of the physical location and configuration of the system that delivers the services.

The most popular cloud computing services fit into one of the following categories: *Infrastructure, Platform* or *Software*. Software as a Service (SaaS)[1] are offered by many service providers, including online project management applications, customer relation management, online office applications and many more. Following the successful adoption of cloud services by the enterprises, different cloud service providers started to offer data access, software, hardware and data storage services. Among them, few efforts takes "as a service" to the software development field where we get *Development as a service (DaaS)*. DaaS is a suite of tools that allows to use traditional development practices for creating on-demand applications. DaaS expands the cloud computing development process to encompass external tools such as integrated development environments, source control systems and batch scripts to facilitate development and deployment[2] with the aim to facilitate and manage software tools and infrastructure for an enterprise, especially with development teams geographically distributed. By leveraging a cloud platform, an enterprise can start using software tools instantly, cost effectively and without managing any development infrastructure [Wil]. An enterprise can utilize cloud development infrastructure to [AFG$^+$10]:

1. Scale on demand.
2. Spend more resources on time-to-market by relieving resources from infrastructure management responsibilities.
3. Achieve higher cost efficiencies.
4. Configure development infrastructure in minutes rather than spending whole day on an installed infrastructure.

Besides hosting development infrastructure services, such as source control, in the cloud, service providers have also started to offer browser-based development environment, allowing to edit, test,

[1]http://en.wikipedia.org/wiki/Software_as_a_service

[2]http://wiki.developerforce.com/page/An_Introduction_to_Metadata_and_Development_as_a_Service

debug and deploy applications using the browser without the need of installing or configuring any tool on a local machine. It enables developers to harness the cloud computing power: just as Platform-as-a-Service[3] (PaaS) enables an enterprise to run applications in the hosted platforms, DaaS provides a new software development stack[4] giving developers the power to write and deploy software applications in the cloud as depicted in Fig. 2:

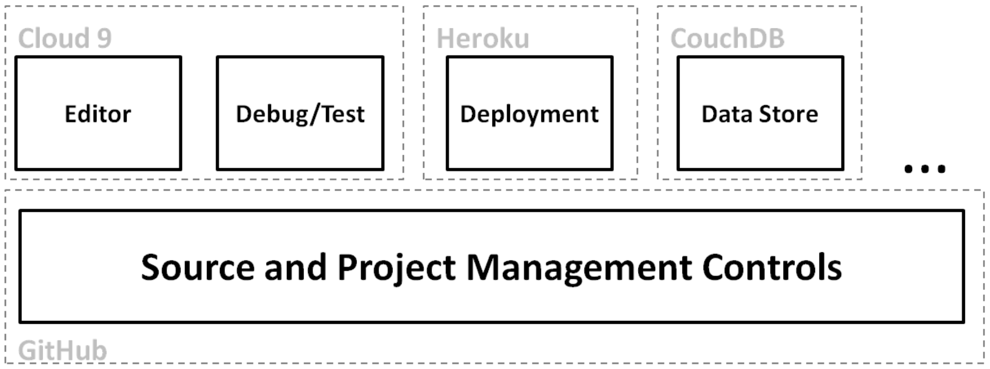


Figure 2: Cloud-based Software Development Stack.

The cloud-based software development stack typically comprises the following components (exemplary coverage overlaid in Fig. 2):

1. **Editor**—A browser-based code editor, allowing to edit, debug and test source code, for example, Cloud9 IDE [cloa].

2. **Deployment**—Deployment of an end-user application in the cloud through a PaaS provider, such as Heroku [Her].

3. **Data Store**—A data storage system (RDBMS or NoSQL) used by an application to store and query application data, for example CouchDB [cou].

4. **Source and Project Management Controls**—Managing and sharing software repositories through code forges, such as GitHub[5].

From the software development perspective, the ability to write, build and deploy a software application using only a browser is a promising approach. This shift of writing code directly in browser-based IDEs is getting more and more attention recently with some trailblaizing example such as Ace [ace], Cloud9, Orion [ori], SourceKit [sou], BrainEngine [bra] and CodeMirror [coda] making impressive advancements[6].

2.1 Browser-based IDE examples

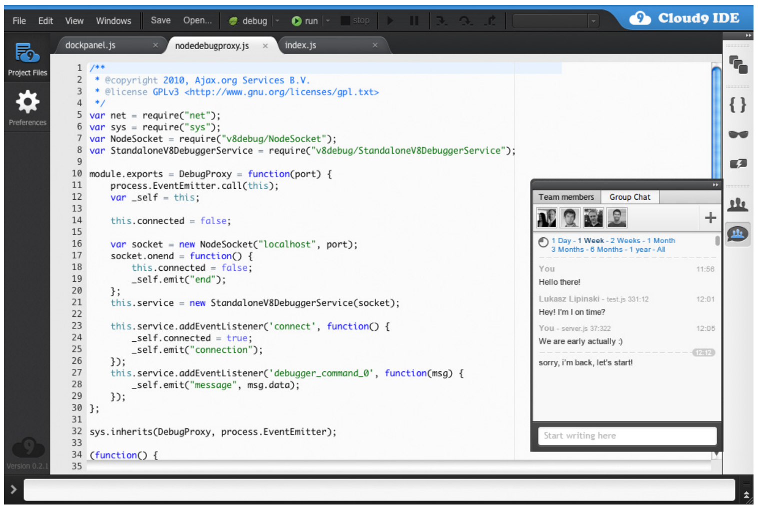
Service providers offer single components or a combination of components as shown in the cloud-based software development stack (Fig. 2). In the beginning, browser-based IDEs were considered

[3]`http://en.wikipedia.org/wiki/Platform_as_a_service`
[4]Kudos to Mike Amundsen for coining the term "cloud-stack" programming in his recent blog post available via: `http://www.amundsen.com/blog/archives/1116`
[5]`https://github.com/`
[6]`http://blogs.developerforce.com/developer-relations/2011/03/browser-based-ides.html`

nothing more than a text editor with syntax highlighting but these online IDEs are more than that, nowadays: they offer a hosted development environment where developers can develop their Web applications in JavaScript, Java, PHP, Python, Ruby, etc. Besides allowing developers to write programs in different programming languages, browser-based IDEs also supports agile and collaborative software development processes. In the following, we will focus on this category; first we describe features of some exemplary components and then we provide a comparison of a selection of such components.

2.1.1 Cloud9 IDE

Cloud9, the leading cloud-based IDE for Web and mobile applications, is a development-as-a-service platform for JavasScript coders and other developers. It all started with a vision of an online JavaScript development platform where all the code is open source and anyone can use it anytime and anywhere. It is built entirely on standard HTML, JavaScript and CSS. It supports syntax highlighting for most popular languages as well as it parses and analyze the code in the background and points out any errors in the code if exists. It lets developer access, edit, debug, test, deploy and share projects. Cloud9 also supports integration with GitHub to support versioning, maintenance and sharing source code online. Further, it facilitates collaboration by allowing developers to chat and collaborate with other fellow developers without leaving the browser-based IDE (see Fig. 3).



```
/**
 * @copyright 2010, Ajax.org Services B.V.
 * @license GPLv3 <http://www.gnu.org/licenses/gpl.txt>
 */
var net = require("net");
var sys = require("sys");
var NodeSocket = require("v8debug/NodeSocket");
var StandaloneV8DebuggerService = require("v8debug/StandaloneV8DebuggerService");

module.exports = DebugProxy = function(port) {
    process.EventEmitter.call(this);
    var _self = this;

    this.connected = false;

    var socket = new NodeSocket("localhost", port);
    socket.onend = function() {
        this.connected = false;
        _self.emit("end");
    };
    this.service = new StandaloneV8DebuggerService(socket);

    this.service.addEventListener('connect', function() {
        _self.connected = true;
        _self.emit("connection");
    });
    this.service.addEventListener('debugger_command_0', function(msg) {
        _self.emit("message", msg.data);
    });
};

sys.inherits(DebugProxy, process.EventEmitter);

(function() {
```

1 Day - 1 Week - 2 Weeks - 1 Month
3 Months - 6 Months - 1 year - All

You 11:56
Hello there!

Lukasz Lipinski - test.js 331:12 12:01
Hey! I'm I on time?

You - server.js 37:322 12:05
We are early actually :)

12:12
sorry, i'm back, let's start!

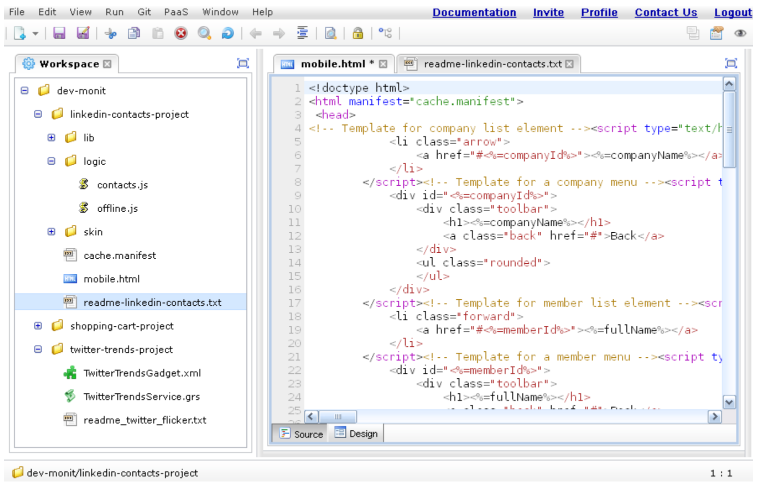
Figure 3: Cloud9 IDE taken from [cloa].

2.1.2 eXo Cloud IDE

The eXo Cloud IDE [clob] supports Java application development and the first to support Java Spring applications[7]. It offers a multi-tenant hosted development environment where teams/developers can collaboratively develop Java, PHP, HTML/JavaScript and Ruby Web applications. It also supports syntax highlighting for many languages along with other features such as Code Auto-complete, Code outline etc. (see Fig. 4). A strong feature of eXo Cloud IDE is that it facilitates

[7]http://www.prnewswire.com/news-releases/exo-cloud-ide-gives-developers-an-on-ramp-to-vmware-cloud-foundry-paas-12
html

developers to easily migrate from development to staging and deployment by selecting from a list of different Platform-as-a-Service (PaaS) environments which includes CloudBees [Cloc], Heroku, RedHat Open Shift [Ope] and CloudFoundry [Ope]. Like a traditional desktop development IDE with a version control system, it also allows to connect to a distributed version control systems such as Git[8].



```
<!doctype html>
<html manifest="cache.manifest">
 <head>
<!-- Template for company list element --><script type="text/h
            <li class="arrow">
                <a href="#<%=companyId%>"><%=companyName%></a>
            </li>
        </script><!-- Template for a company menu --><script t
            <div id="<%=companyId%>">
                <div class="toolbar">
                    <h1><%=companyName%></h1>
                    <a class="back" href="#">Back</a>
                </div>
                <ul class="rounded">
                </ul>
            </div>
        </script><!-- Template for member list element --><scr
            <li class="forward">
                <a href="#<%=memberId%>"><%=fullName%></a>
            </li>
        </script><!-- Template for a member menu --><script ty
            <div id="<%=memberId%>">
                <div class="toolbar">
                    <h1><%=fullName%></h1>
```

Figure 4: eXo Cloud-IDE browser.

Although we have described two powerful browser-based IDEs (see Table 1 for a list of different browser-based IDEs), there are also many PaaS providers available to date which offers variety of software services to run or deploy software applications.

2.1.3 Jelastic

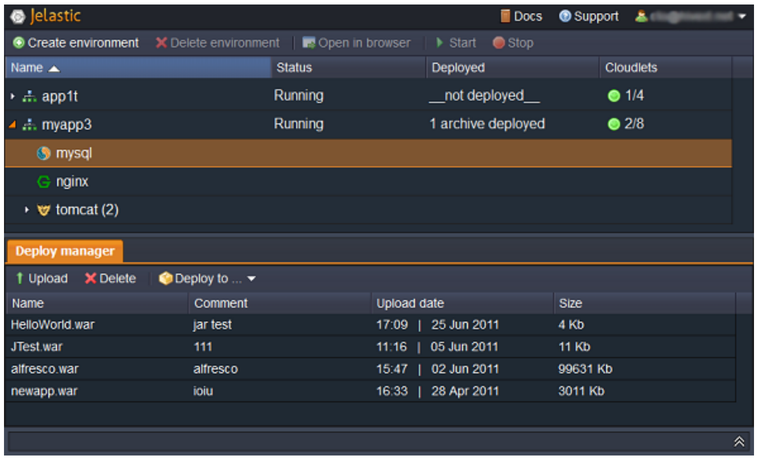
Jelastic [jela] is a next generation Platform-as-a-Service for Web applications. Using Jelastic, application provider can select the software stack, an initial instance size without installing or configuring anything as shown in Fig. 5. Jelastic is designed to support deployment of Java-based Web applications in the cloud and hence only allows the application provider to select among a list of relational databases and Java supportive application servers.

With Jelastic, application deployment is only a matter of uploading the application package (e.g. *.war file) to the selected environment. It also keeps track of the history of deployments so that the application provider can deploy an older version of the application, if necessary.

With browser-based IDEs like Cloud9, eXo Cloud etc., developing and deploying applications becomes much more streamlined and enables cloud platforms easily accessible to the developers. These Web-based programming environments can (nearly) replace our desktop IDEs and code editors. They supports most of the programming languages available to date as well as allows developers to start connecting to each other into the cloud and build up teams in a way that they couldn't be able to do before, hence giving them more freedom to organize their team development.

[8]`http://git-scm.com/`

Apart from team building and programming language supports, it offers convenience, pace – the ability to ramp up a project in the order of minutes rather than days or weeks which is a huge benefit.



Name	Status	Deployed	Cloudlets
app1t	Running	__not deployed__	1/4
myapp3	Running	1 archive deployed	2/8
mysql			
nginx			
tomcat (2)			

Name	Comment	Upload date	Size
HelloWorld.war	jar test	17:09 | 25 Jun 2011	4 Kb
JTest.war	111	11:16 | 05 Jun 2011	11 Kb
alfresco.war	alfresco	15:47 | 02 Jun 2011	99631 Kb
newapp.war	ioiu	16:33 | 28 Apr 2011	3011 Kb

Figure 5: Jelastic Environment Manager taken from [jelb].

2.2 Comparison of browser-based IDEs

In this section we compare browser-based IDEs as examples of cloud-based software development components (Table 1): we compare them in terms of programming language support, integration with source control and concerning project collaboration features as well as concerning deployment options. Developing code directly in the cloud should also enable developers to deploy the application into the cloud from their browser-based IDE, we will see which browser-based IDEs currently supports application deployment in the cloud. More often than not, a software project is developed by a team of developers who are geographically distributed; we hence outline which browser-based IDEs supports the social collaboration aspect.

Product	**Language**	**Source Control Integration**	**Project Collaboration**	**Deployment/Data Store**
CodeRun [codb]	C#/PHP/JavaScript/HTML	No	Yes	Yes
Cloud9	Ruby/PHP/JavaScript/HTML	Yes	Yes	Yes
eXo Cloud	Java/PHP/Ruby/JavaScript/HTML	Yes	Yes	Yes
Bespin [bes]	JavaScript/HTML/CSS	No	Yes	No
Kodingen [kod]	PHP/Perl/Python	Yes	Yes	No
codeanywhere [codc]	PHP/CSS/JavaScript/HTML	No	No	No
ECCO [ecc]	PHP/Java/JavaScript/HTML	No	No	No
WonderFL [won]	Action Script3	No	Yes	No

Table 1: Comparison between browser-based IDE(s).

As shown in Table 1, all browser-based IDEs supports Web standard scripting languages although few of them also support object oriented languages such as Java. Version tracking on source code is an important asset in software development but not all browser-based IDEs currently support it. Some browser-based IDEs also supports project collaboration by allowing developer to share his/her workspace with other fellow developers in order to collaborate with each other. Only few browser-based IDEs support all the features under consideration.

In the next section, we will discuss challenges concerning browser-based IDEs along with potential features which these IDEs could take into consideration to gain a competitive advantage.

3 Challenges and Opportunities

In traditional in-house software development, we deal with different software repositories which surround a particular project. These software repositories are necessary to maintain and execute a project in an structured way. As we have discussed the emerging trends of cloud IDEs to support software development (cf. section 2) and the different browser-based IDEs which are available to date (cf. Table 1), we foresee the development of software projects in the cloud. However, these browser-based IDEs do have their limitations: most focus on Web application development using only HTML, JavaScript etc., others support Web application development using Java etc. Further, quite often support for accessing other software repositories is lacking, which is quite often necessary to manage large projects. In the following we discuss challenges or areas of improvement which can foster the adaption of browser-based IDEs:

1. **Dealing with the agile software development life-cycle.** In agile software development, immediate feedback is essential and it is generated through frequent commits, builds, testing and continuous integration. As the cloud-based development infrastructure offers support for these software tools as a single integrated platform (e.g. CollabNet TeamForge[9]), it is worth investigating if the online browser-based IDE adds more efficiency, support and transparency than developing the code in a desktop-based environment.

2. **Automation of tasks.** Cloud-based infrastructure makes it straight-forward to export and share data from software tools (i.e., coding, testing, build logs, integration tests etc.). Real time capturing of data from these tools will enable organizations as well as developers to measure performance, tracking activities of a developer and monitoring the overall progress on the project.

3. **Integration with code forges.** Hundreds of thousands of projects are hosted on different code forges. Only few browser-based IDEs do support integration with code forges (e.g., Cloud9 supports GitHub integration). Integration with code forges will enable developers to easily work on software projects which are already hosted on these code forges, using browser-based IDE. As code forges hosts a variety of project management tools (i.e., bug tracking systems, mailing lists, discussion forums), it will further assist developers to reuse the existing project management tools hosted on these code forges. For example, a bug could be fixed by modifying the source code in a browser-based IDE, compiling, building, testing and later pushing the changes back to the code forge along with changing the status of a bug on the code forge.

4. **Bridging the "off-line gap".** Browser-based IDEs should allow developers to import or export projects directly from their desktop IDEs to the cloud. A developer to should be able to code on his local machine and a single push button would allow him to push the changes to

[9]`http://www.collab.net/products/ctf/`

the cloud so that other team members can access or review the code changes using browser-based IDE. In the same direction the question has to be answered how (local or remote) backup are supported.

4 Conclusion

In this report we have established a cloud-based development stack and reviewed and discussed browser-based IDEs that can bring a significant change in the way software development is carried out. With the existing browser-based IDEs, *small applications* can be easily written and deploy, however we believe that they are not ready yet for prime time to handle large software projects. We have identified a number of challenges in the cloud-based development stack, especially concerning the browser-based IDEs, which might hamper their take-up.

References

[ace] `http://ajaxorg.github.com/ace/`. Last accessed on: 01/05/2012.

[AFG+10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53:50–58, 2010.

[bes] `http://mozillalabs.com/blog/2009/02/introducing-bespin/`. Last accessed on: 01/05/2012.

[bra] `http://www.brainengine.net/`. Last accessed on: 01/05/2012.

[cloa] `http://cloud9ide.com/`. Last accessed on: 01/05/2012.

[clob] `http://cloud-ide.com/`. Last accessed on: 01/05/2012.

[Cloc] `http://www.cloudbees.com/`. Last accessed on: 01/05/2012.

[coda] `http://codemirror.net/`. Last accessed on: 01/05/2012.

[codb] `http://www.coderun.com/studio/`. Last accessed on: 01/05/2012.

[codc] `https://codeanywhere.net/`. Last accessed on: 01/05/2012.

[cou] `http://couchdb.apache.org/`. Last accessed on: 01/05/2012.

[ecc] `http://ecco.sourceforge.net/`. Last accessed on: 01/05/2012.

[Her] `http://www.heroku.com/`. Last accessed on: 01/05/2012.

[jela] `http://jelastic.com/`. Last accessed on: 01/05/2012.

[jelb] `http://jelastic.com/docs/setting-up-environment`. Last accessed on: 01/05/2012.

[kod] `https://kodingen.com/`. Last accessed on: 01/05/2012.

[Ope] https://openshift.redhat.com/app/. Last accessed on: 01/05/2012.

[ori] http://wiki.eclipse.org/Orion. Last accessed on: 01/05/2012.

[sou] https://chrome.google.com/webstore/detail/iieeldjdihkpoapgipfkeoddjckopgjg. Last accessed on: 01/05/2012.

[ST09] Bianca Shibuya and Tetsuo Tamai. Understanding the process of participating in open source communities. In *Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, FLOSS '09, pages 1–6, Washington, DC, USA, 2009. IEEE Computer Society.

[Wil] Wang Willie. Reinforcing Agile Software Development in the Cloud. White paper.

[won] http://wonderfl.net/. Last accessed on: 01/05/2012.