2-2016

DH Box: A Virtual Computer Lab in the Cloud

Stephen Zweibel
Graduate Center, City University of New York

DH BOX: A VIRTUAL COMPUTER LAB IN THE CLOUD

by

STEPHEN ZWEIBEL

A capstone research project submitted to the Graduate Faculty in Liberal Studies in partial fulfillment of the requirements for the degree of Master of Liberal Studies, The City University of New York
2016

DH BOX: A VIRTUAL COMPUTER LAB IN THE CLOUD

by

STEPHEN ZWEIBEL

This manuscript has been read and accepted for the Graduate Faculty in Liberal Studies in satisfaction of the capstone project requirement for the degree of M.A.

	Matt Gold_____
_____	_____
Date	**Advisor**
	Matt Gold_____
_____	_____
Date	**Executive Officer**

THE CITY UNIVERSITY OF NEW YORK

Abstract

DH BOX: A VIRTUAL COMPUTER LAB IN THE CLOUD

by

STEPHEN ZWEIBEL

Adviser: Professor Matthew Gold

Education in the use and manipulation of Digital Humanities (DH) tools is rife with challenges, stemming from issues of knowledge (ie, how to install, configure, and work with DH tools on a variety of devices) and resources (access to hardware and software). Particularly in an academic environment, humanities scholars may not only lack access to institutional devices, but their drive to experiment with new programs may run counter to the security concerns of their information technology department personnel. DH Box, originally developed as a project for an introduction to DH course, was conceived as a solution to these challenges: a toolset that would offer simpler setup and execution of DH instruction, to create an easy-to-deploy learning environment for DH students. The project has cycled through a number of iterations, from local installs to a cloud-based solution, and in its current form has been tested in several classroom and workshop scenarios. This testing has confirmed its utility with respect to the challenges of learning DH tools and has informed its continued development. With growing awareness and further development, it is anticipated that DH Box will see still wider adoption.

Table of Contents

Introduction & Description of Problem

DH Box was originally conceived as a final project proposal for the Digital Humanities Praxis Seminar led by Professors Matthew K. Gold and Steve Brier in the fall of 2013. The idea arose out of my experiences holding workshops on technical topics (such as Python, JavaScript, CSS) for a largely non-technical audience. I noticed that it was often necessary to spend a lot of time before and during workshops in installation, configuration, and troubleshooting participants' setups.

In my work as a librarian at Hunter College, I taught a number of classes for fellow CUNY librarians and faculty introducing technical topics and techniques. As these classes were held at different campuses across CUNY, I encountered a variety of IT practices and restrictions, and nowhere was it readily possible to use institutional computers to do the kind of work that was necessary for the subject matter I was teaching. For example, in order to write a script in Python, one must have access to the command line and the ability to install programs, and such activities are routinely prohibited by academic IT departments.

These situations all required some sort of workaround, which often meant asking attendees to bring and use their own laptops. This solution is far from ideal: participants may lack access to a computer of their own, or may simply forget to bring it and be unable to do the work. In an institution of higher learning, especially when there are computers available, I think it should be possible for students and faculty to gain hands-on experience grappling with a technical problem. I found the IT equipment and policies at the schools I visited were confounding this goal.

At most campuses, computers that are available to students or situated in computer labs are completely locked down, barring users from downloading or installing software, running an executable file, or even saving files except in a single, designated "student" folder. In my

experience, the only available operating system is typically Windows; at some campuses, there might be a few Mac machines, but all were equally locked down.

Academic IT departments take a defensive stance that is in many ways understandable. They have a need to prevent users from accidentally or maliciously installing viruses or other malware; from putting offensive or disturbing images on the desktop background; or from otherwise manipulating a computer's settings such that it becomes unusable for the next person. IT departments construct blanket policies against such misuse of equipment because it is much easier and less labor-intensive than it would be to evaluate, monitor, and support individual users' requests for more space or access. Having a policy without exceptions saves IT departments from the costs of replacing or repairing equipment and from the added time spent in consultation with users who want to do more than average with their machines.

Academic IT departments espouse values of safety, risk management, and clarity of responsibility; and they therefore create environments suitable for a relatively passive use of computers. They allow users to find resources online and read or print and write about them. They may provide for the analysis of data, using whatever software they have decided to install (for example, data suites like IBM's SPSS). This is because IT must also deal directly with software vendors and consider the kind of licensing and support these companies offer.

As long as the kinds of files that can be used on an institutional computer are carefully circumscribed, the security of the resource is upheld; but the device is rendered into a kind of sophisticated television, a passive vehicle for receiving information, rather than creating or manipulating it. Users may be able to compose an essay, but they are very limited in their ability to take advantage of computing power. In contrast, many students and scholars are interested in experimentation and play using digital, and especially computational, techniques. As many

people who do not have a background in computer science are discovering, computational techniques have the power to enrich their research and enable the exploration of new questions.

This ideal of computational exploration and experimentation comes to a hard stop against the security-oriented goals of university IT departments. I found my requests for a loosening of restrictions met with not just resistance but incomprehension, as a museum curator might respond to a request to graffiti the gallery walls. My requests ran directly contrary to IT department goals. I was asking them to risk their hardware and software for the purpose of open-ended play and learning.

At the same time, it was becoming clear to me in the DH Praxis class that humanists were struggling to integrate certain technologies into their scholarship. DH techniques, such as topic modeling, data mining, web scraping, and digital archiving, promise a fresh perspective on many of the traditional objects of humanities research. In the last five years there has been a great deal of interest in computational approaches to the humanities, not least due to the success of these tools in providing a new angle of inquiry. But for many students and professors in the humanities, there is a gap between learning about the existence of DH tools and techniques and actually applying them to research.

To use DH tools such as IPython, RStudio, and NLTK, scholars must first set up a development environment, a process that often results in compatibility issues that can be resolved only through specialized knowledge or through trial and error. Such issues are often compounded by version-specific errors or the configuration requirements of an individual project. While learning to set up a DH computing environment can be a valuable learning experience in itself, group workshops and classes can be easily derailed by setup problems. The time required to configure each machine in a computer lab is often prohibitive, in addition to the

difficulties presented by local networking and security considerations. The configuration process can also deter those seeking to learn these tools on their own, since a curious student must invest significant time on highly technical preparation before the engaging work of exploration can begin.

Even veteran DH scholars can find it frustrating to prepare an environment on a new machine or ensure that new software is compatible with an existing setup. Access to technical resources (including hardware and support personnel) is a related concern, particularly for students and faculty in the humanities, where there may be limited access to computer laboratories and/or the technical support services they need.

Recognizing these barriers to entry for DH, especially for those lacking technical skills or background experience, I wanted to develop a more portable, approachable, and reproducible way to get interested people up and running with DH tools.

Development & Iteration of DH Box

Preliminary work on DH Box began in Spring 2014 as part of the DH Praxis Seminar, in collaboration with a group of other students from the class. The team consisted of myself, Cailean Cooney (outreach coordinator), Harlan Kellaway (HTML), and Gioia Stevens (project manager). Together, we strove to create a working prototype and a supporting website by May 2014. Early work was completed using Google Drive as a central document repository and GitHub to host our code. To manage tasks, we initially chose Asana, but soon decided to use GitHub's Issues utility.

Gioia Stevens created issues for each task we needed to accomplish and grouped them into milestones with due dates tracked to meet our launch deadline. We discussed the level of granularity for tasks and the question of tracking mainly larger milestones vs. smaller "sprints"

with shorter due dates. We ultimately chose to go with a more general, “big picture” plan, which worked well given our small team, and our near-daily email communication. We started making this work visible with our project blog, and then developed a Twitter account and a website. Additionally, we gave a number of presentations on DH Box to introduce it to potential user communities.

Part of the inspiration for DH Box came from the growing availability of affordable, highly portable microcomputers such as the Raspberry Pi. I originally imagined DH Box as a customized Linux environment for DH learners that would be useable on any computer that could run Linux; but I was especially excited by the idea of instantiating this environment and suite of DH tools onto the Raspberry Pi, a pocket-sized device that costs $35. In essence, DH Box would be a set of scripts that installed common DH applications (like Omeka, MALLET, NLTK) onto the user’s system. DH Box could also be pre-installed on the light and portable Raspberry Pi, making it especially useful in environments where technical resources are scarce.

But as DH Box developed, the platform shifted. Increasingly, it made sense to move away from solutions that depended on the idiosyncrasies of each individual user’s system, toward a framework that made use of cloud computing. The group met with Dennis Tenen, Assistant Professor of English and Comparative Literature at Columbia University, who had worked on a similar project. He pointed us toward cloud technology and advocated for a change in infrastructure from local installs on user hardware to a web application that would be accessible remotely. It became clear that the cloud installation was much more feasible and would allow us to avoid having to troubleshoot various hardware, operating systems, and the details of different edge cases. With the remote, cloud-based infrastructure, we could decide on

and target just one computing environment. We began to look toward hosting instances of a virtual computer that any user could launch from an internet-connected device.

By the end of the 2014 spring semester, we had developed DH Box into a cloud-based Linux computer, and configured it with a number of tools for analysis and visualization. Users could access and work with the DH Box suite of tools (including R Studio, a statistical analysis tool; Omeka, a tool for creating digital exhibits; or MALLET, a topic modeling tool) directly from a web browser, obviating the need to install or configure anything new.

Infrastructure

The infrastructure of DH Box consists of a sign-up website; a back-end that "listens" for requests, takes data, and initiates the launch of a DH-Box script; the launch script itself, including customized applications; and a simple user menu.

1) Sign-up website

Users can sign up for a new DH Box instance via a website (www.dhbox.org) created using Jekyll, which includes a web form and fields for the entry of a username, password, and email address. We are working on adding options for more customization at this step, allowing users, for instance, to choose the software that will be installed on each DH Box, and a selection of datasets to work with. Once users have submitted their basic sign-up data, the back-end server is engaged.

2) Back-end database

The server "listening" for requests from the sign-up website is a standard web application programming interface (API), written in Python and using Flask, a library for creating web applications. It initiates and runs the DH Box installation script, inserting the data collected from the DH Box sign-up page.

3) Launch script

The DH Box launch script is as a Dockerfile, a configuration file that instructs Docker how to build a container. Docker automates the installation and configuration of software on virtual servers, as well as other elements of systems administration. The launch script applies the user's specifications to each of the tools installed on the DH Box and uses the same username and password to secure each individual application.

The script launches a new DH Box instance, that is, a new virtual computer, installs the requested applications, and makes the computer available on the web at a new IP address.

4) Customized applications

All of the DH Box applications are available at the specified IP address, and are customized by the install script. All applications are password-protected. Applications with a web component can be accessed via a link to that address; libraries, command-line tools, etc., must be accessed via the web command-line application. Desktop applications may be made accessible in the future, probably through a visual secure-shell (SSH).

5) User menu

Users can access all the tools available on their DH Box through a central web page, from which each tool can be launched with the click of a button. We are developing a feature that will allow users to upload and download files to work with from this page.

Advantages of DH Box

DH Box streamlines the process of setting up a digital working environment. Because DH Box is accessed via the browser, it functions equally well across a broad range of devices, both mobile and desktop. The cloud-based nature of the DH Box platform removes barriers to entry that stand between students and the exciting methodologies provided by DH tools. At the

same time, veteran DH researchers benefit from the ease and portability of a cloud-based digital working environment.

DH Box cuts down on class time devoted to preparing technology, shifting the focus away from maintenance and configuration and toward learning and exploration. Further, because DH Box is compatible with a variety of devices and operating systems, students can access DH tools from their own favored devices rather than from an assigned lab machine, reducing dependence on institutional resources that are often limited.

DH Box users simply log in to their account from any device, at which point they are presented with a graphical user interface listing the currently available apps, libraries, and extensions. When a user clicks on one of these utilities, she is granted immediate access to the tool, with all the same functionality as a version installed, configured, and run on her local machine.

Use Cases (Co-written with Patrick Smyth)

DH Box has been built for use in the classroom, though it is useful for advanced researchers as well. The following use cases show how DH Box might be incorporated into both research and pedagogy:

- Laura, a student in a seminar on nineteenth-century British literature, wishes to determine whether Dickens used active-voice constructions more often in his early career. Laura is new to the digital humanities, and isn't sure if natural-language processing with NLTK or a statistical approach using RStudio would be the best way to answer her research question. Laura logs into DH Box, clicks the icon for NLTK, and imports a Dickens corpus through a command line accessible via her browser. Now she

can experiment with NLTK's native language processing functions without a significant time investment in installation and setup.

- Dr. Perez is planning a module on Baroque painting in his Art History survey. As part of this module, he wishes to show his students how to perform image analysis with SciPy and wants his students to be able to follow along with his on-screen demonstrations during class. He would also like his students to be able to access SciPy outside of class for use in assigned projects. Rather than refer students to IT or restrict their time with SciPy to the lab, Dr. Perez has them sign up for DH Box before class and log in during the first few minutes of the session. Students can continue the work they have begun in class as homework by logging in to DH Box at a later time.

- Dr. Tan, a lecturer on book history, wishes to introduce her students to Omeka, a content management system which her class will use to assemble a series of items from their institution's library into an online exhibit. Rather than set up her own remote server or use Omeka.net, which is limited by tiered payment plans and accompanying theme/plugin restrictions, Dr. Tan logs into DH Box to access her Omeka administrative console. After Dr. Tan's demonstration, her students begin creating their own Omeka project in class using their own DH Box accounts, which they can access later to complete their assigned projects from home.

The first real use case of DH Box took place in Professor Jeff Allred's undergraduate English class at Hunter College. Through presentations on DH Box made at Hunter College's ACERT seminar, the project came to the attention of Professor Allred, who expressed interest and agreed to participate in an alpha test of DH Box in his classroom.

Professor Allred wanted his class to collectively produce an Omeka exhibit website about novels set in New York City in the era of Henry James. His class' use of DH Box and the documentation he provided gave us essential user data, and we have been able to make several upgrades based on his feedback, especially with regard to workflow and appearance (the placement of buttons) and documentation. His experience also highlighted a need for simple import and export of files to and from DH Box, a feature we are in the process of developing.

Other use cases of DH Box include several workshops at the CUNY Graduate Center, led by the GC's Digital Fellows, on topics like the use of the command line and an introduction to databases. Overall, DH Box has met the needs of workshop presenters for these kinds of topics; future developments will focus on increasing its flexibility in demonstrating other Linux-compatible tools.

Challenges

While these use examples emphasize the flexibility and accessibility of DH Box's server-side architecture, there are a number of drawbacks associated with building a cloud-based platform. Some of these difficulties include issues with offline accessibility, information siloing, and data security. Many proponents of cloud computing, for instance, argue that as broadband connections are now commonplace, offline access to key tools is no longer a necessity.

However, in many countries and at a variety of resource-poor institutions, high-speed internet access continues to be unreliable. For this reason, the DH Box team intends to prepare a distribution of the platform that can be installed using flash drives or other readily available removable media, allowing DH Box to be deployed in more unconventional teaching environments. The team has also moved to prevent data siloing by allowing users to export their

own instances of the platform. This means that users can download not only the data they generate, but also a working copy of their entire DH Box setup.

Finally, because security is a concern for cloud-based services that provide considerable flexibility to their users, the DH Box team plans to implement anonymized monitoring processes that will alert administrators to potential misuse without compromising user privacy. In these ways, the DH Box team hopes to address some of the challenges that accompany cloud-based computing.

Funding and Future of DH Box

We received a Faculty Innovations in Teaching with Technology (FITT) Grant award to incorporate DH Box in the curriculum of Professor Allred's English class at Hunter College during the Fall 2014 semester. DH Box has also been used to introduce students in the Digital Praxis Seminar at the CUNY Graduate Center to a variety of DH tools. DH Box also won an Educational Research Grant award from Amazon Web Services that has helped fund server needs in 2014 and 2015.

In Spring 2015, the DH Box team won a Digital Humanities Start-up Grant from the NEH's Office of Digital Humanities, which has provided support for work to expand outreach, documentation, and access to more tools. Project development goals include:

· Incorporate new tools, including Text Encoding Initiative (TEI) software, and Gephi

· Facilitate easier uploading and downloading to and from DH Box

· Add more pre-installed datasets and corpora, such as Project Gutenberg Books and nineteenth-century newspaper archives

· Provide more extensive documentation

· Further tailor DH Box for use in the classroom

· Stress test security measures and simultaneous user capacity

· Gather user feedback to ensure the long-term growth and sustainability of the project

Beyond the technical work of building and refining DH Box, we are also using this funding to make DH Box available to a wider audience of teachers, students, and scholars. And we are working to build a community of educators, librarians, and developers around increasing the utility of DH Box for academia.

It is gratifying to see DH Box in action, addressing the challenges that motivated its development. For students and scholars starting to experiment with DH techniques and for educators hoping to incorporate DH tools into their curricula, DH Box has removed barriers and eased technical difficulties. The process of developing and refining DH Box, from local installs to cloud instantiation, through the real-world user experience that has helped us continue to improve it, has likewise been a learning experience. As DH Box is being used in more classrooms and workshops around CUNY and elsewhere, including a key role in the 2016 Digital Humanities Summer Institute, we are putting personnel and infrastructure in place to facilitate further developments. The overarching goal for DH Box will continue to be to enable digital humanities scholarship.

Bibliography

Anderson, Sheila. "What are Research Infrastructures?." *International Journal of Humanities and Arts Computing* 7.1-2 (2013): 4-23.

Anderson, Sheila, and Tobias Blanke. “Taking the Long View: From E-science Humanities to Humanities Digital Ecosystems”. *Historical Social Research / Historische Sozialforschung* 37.3 (141) (2012): 147–164.

Blanke, Tobias, Conny Kristel, and Laurent Romary. "Crowds for Clouds: Recent Trends in Humanities Research Infrastructures." *arXiv preprint arXiv:1601.00533* (2015).

Selwyn, Neil. *Digital Technology and the Contemporary University: Degrees of Digitization*. New York: Routledge, 2014.

Thaller, Manfred. “Controversies Around the Digital Humanities: An Agenda”. *Historical Social Research / Historische Sozialforschung* 37.3 (141) (2012): 7–23.

Van Zundert, Joris. "If you build it, will we come? Large scale digital infrastructures as a dead end for digital humanities." *Historical Social Research/Historische Sozialforschung* (2012): 165-186.