VIRTUALIZATION FOR PROCESSING AND ACCESSING DIGITAL ARCHIVES

Shelly Black
NC State University Libraries
USA
syblack@ncsu.edu
0000-0002-9046-4866

Brian Dietz
NC State University Libraries
USA
bjdietz@ncsu.edu
0000-0001-7190-2755

[Matthew] Farrell
Duke University Libraries
USA
matthew.j.farrell@duke.edu
0000-0003-1502-2651

Abstract – At a basic level, virtualization [1] is the use of a host computer or server's resources to run other computing environments. There are many ways in which virtualized computing environments may be deployed and interacted with, including using software to virtualize additional desktops on a local computer (e.g., VirtualBox, Hyper-V Manager, or VMWare) or accessing virtual command line interfaces hosted by a server or computer cluster [2],[3], and emulating old video game systems on contemporary hardware [4]. In this paper we discuss a cross-institutional collaboration on using containerization and desktop virtualization in digital curation at academic special collections libraries.

Keywords – Containerization, desktop virtualization, virtual machine, special collections, born-digital archives, virtual reading room

Conference Topics – We're All in this Together; Sustainability: Real and Imagined

I. INTRODUCTION

While there are many applications for virtual machines, we highlight two general affordances. First, a user accessing a virtual machine can make use of software and processes in a different operating system, such as Linux-only toolsets on a Windows host. Second, virtual machines allow users to use specific or unique computing environments remotely. Applying these specifically to born-digital special collections work, running virtualized environments allow staff and researchers to access consistent toolsets and configurations regardless of the host computer(s) in use. The authors' staff computing environments are composed of multiple, varied physical configurations, but through the use of virtual environments, each of our workstations can make use of consistently packaged, identical processing environments for technical services workflows. In terms of public services work, virtualized environments allow one to create controlled environments for researchers to access digital archival materials remotely.

II. CONTAINERIZATION FOR PROCESSING

Container technology allows one to package applications and dependencies into a Linux environment so that they can be tested and deployed and trusted to work consistently across computing platforms [5]. Compared to other types of virtualization, containers are usually defined to contain only the resources necessary to complete a specific set of tasks as opposed to an entire operating system. Separate containerized applications or "services" can be run together in an orchestrated way. For instance, an application may include separate containers for a web service, database, and SOLR index. Additionally, invoking a process in one container may call additional containers to perform additional automated or semi-automated processes. Docker and Podman are two popular platforms that support containerization.

For several years, North Carolina State University Libraries (NC State) has managed the majority of its born-digital processing tools using Homebrew for Mac [6], along with pip for installing Python packages. Duke University Libraries has managed Windows computers that run the BitCurator environment as a virtual machine and in a dual boot

iPRES 2023: The 19th International Conference on Digital Preservation, Champaign-Urbana, IL, US.

https://hdl.handle.net/2142/121116

configuration. Both organizations were motivated to find a more lightweight and flexible approach to managing applications, and one that might avoid the complications of updating working environments following new releases and installing tools on new machines.

In 2020, NC State started to examine container technology to address these issues [7]. Doing so would simplify installation and management of command line tools; better support cross-platform replication, functionality and user experience; and result in a shareable and replicable approach. Duke joined the process as a collaborator in 2021.

Early explorations began by defining a minimal viable product (MVP): a container one could use to perform virus scans, search for personally identifiable information (PII), and conduct file format characterization on files accessible via the host computer. NC State initially attempted to create an image using an official Docker build of Linux Homebrew [8], drawing on past experience working with Homebrew for Mac. When this proved infeasible, the next attempt was to build an image based on the official Docker build of Kali Linux, installing their “forensic metapackage” of applications [9]. This reached MVP, but it was ultimately decided that the extent of tools available in the metapackage resulted in a bloated image and container. Drawing on this success, we focused on using official Docker builds of Ubuntu and Fedora Linux [10], which resulted in the creation of a more tightly scoped image, i.e., one that excludes extraneous tools. Recent testing coincided with both organizations purchasing or assessing Apple computers with the Apple Silicon ARM chips, leading to the creation of containers based on the ARM Linux image. This period of iteration confirmed an early assumption: that adding to or otherwise updating a container is more efficient than performing similar maintenance across multiple standalone workstations.

To date, NC State and Duke have written Dockerfiles that contain instructions for building an Ubuntu-based AMD64 Linux image and Fedora-based AMD64 and ARM Linux images [10]. The container used at each institution during processing is derived from these images. At both NC State and Duke, the container environment includes command line applications for searching for sensitive data and duplicates, virus and malware detection, and file characterization, as well as general Linux file utilities. With these toolsets, the containers support the same range of files and content types as is currently supported in systems such as the BitCurator environment. However, some steps in our workflows will continue to be done on the host. Disks can be shared as volumes, and their files can be packaged from within the container. Yet containers cannot access devices, unless used on a host Linux computer. Disk imaging and optical disc audio ripping must be completed on host Mac and Windows workstations. Our containers are currently deployed on these three host operating systems, and we are using containers in production or expect to be by fall 2023.

III. Desktop Virtualization for Access

NC State uses desktop virtualization to provision a remote virtual reading room service. A special collections reading room is traditionally a mediated environment where researchers can use materials. In some cases, there are copyright, privacy, or other donor-imposed access restrictions [11]. This applies to physical and digital materials. In the past, researchers at NC State who requested born-digital or digitized materials had to use an air gapped [12] laptop in the reading room. Specifically, WiFi and USB device access were disabled, so that researchers could not transfer the materials to themselves. Desktop virtualization allows NC State to replicate this secure environment for accessing digital materials online, eliminating the need for travel, and allowing multiple researchers to use it simultaneously.

Some institutions use digital asset management systems (DAMS) which function as virtual reading rooms [11, pp. 162-163]. These are appropriate for materials that can be openly shared, and for file formats that can be rendered in a browser or downloaded for viewing locally. However, maintaining a DAMS can be labor intensive. NC State’s virtual reading room relies on existing infrastructure provided by the university’s Virtual Computing Lab (VCL) [2]. This on-demand, virtualized computing service allows classes and researchers to connect to a remote server using Remote Desktop Protocol [13] software and access custom software environments.

In 2020 NC State began working with VCL on our server reservation. We created an Ubuntu Linux image that contains software and networking configurations from which the server can reboot. We installed open source software for viewing text documents, images, videos, and other common file formats, as well as a module to redirect sound. Security configurations include firewall rules blocking HTTPS traffic, a disabled SSH client, and disabled drive and clipboard redirection. Thus, researchers cannot copy, download, or email materials to themselves. Linux permissions are also applied, so that the researcher can only view the files they requested. We cannot prevent them from taking screenshots. However, when they request to use the virtual reading room, they agree that materials are non-circulating and any pictures taken are for research purposes only. Another safeguard is that the virtual reading room can only be accessed with NC State credentials or by external researchers who create accounts with VCL. Administrative access to the server is controlled by an access group, to which staff were added through the VCL website.

The virtual reading room is currently an active service, having been used by five researchers in the 2022-2023 academic year. All researchers have succeeded in accessing and viewing their desired materials, with one exception, where the researcher could not connect for unidentified reasons. Feedback provided by researchers has been encouraging. We also receive regular inquiries from other institutions on how to implement this service, and Duke is interested in exploring or adapting NC State's approach for use with its patrons.

IV. Lessons Learned and Future Work

Once we have more production experience with containers, there are additional areas of exploration to consider. This includes best practices in building images and efficiencies in maintaining them. While we currently use one container for all processes, we may further explore whether and when to split our containers into separate, coordinated, specialized services, such as those for processing email archive files or used in post-processing work. We are also eager to explore the extent to which containers might support certain automated workflows.

Testing is also anticipated for the virtual reading room. It is most likely that researchers would request text documents, images, videos, or other common file formats. However, future use cases may include providing access to less common file formats, such as those used in computer-aided design, or an emulator to run legacy software. Overall, the user experience for researchers can be improved. This includes video streaming quality when using Microsoft Remote Desktop for Mac. Using assistive technology with the virtual reading room also needs to be tested. Additionally, ongoing maintenance involves ensuring that the virtual reading room uses a currently supported version of Ubuntu. It was originally installed with Ubuntu 18.04 LTS, which having just reached its end of life [14], required an upgrade. Because the security configurations were scripted, setting up a Ubuntu 22.04 LTS server as the virtual reading room required little effort.

Setting up, maintaining, and using both containers and virtualized desktop environments requires some degree of technical knowledge. As we deploy containers into full production, we will be gathering feedback from full-time and student staff, particularly to better understand gaps in technical skills. To use the environments, they need a basic working knowledge of a shell and the Linux file system. More technical knowledge is required to administer containers and customized virtual desktops, including a general understanding of virtual computing. Specific knowledge is required for building, deploying, updating, and managing these environments. That said, the authors are self-taught and do not have formal backgrounds in systems administration or IT desktop support.

The projects in this presentation started at a single institution before expanding to a peer organization [15], but wider distribution has been a consideration since the earliest stages. Our containerized processing environments can easily be distributed via Git as Dockerfiles, and can be reused, amended, and otherwise modified from the base versions to fit the use cases of other institutions. Similarly, the shell script to configure the virtual reading room can be shared and applied to a virtualized desktop hosted by other institutions or cloud computing services. We believe virtualization can increase the availability of processing environments and digital special collections for staff and researchers, respectively.

1. REFERENCES

[1] IBM. "What is Virtualization?" https://www.ibm.com/topics/virtualization (accessed Mar. 1, 2023).

[2] NC State University. “Virtual Computing Lab.” https://vcl.ncsu.edu/ (accessed Mar. 1, 2023).

[3] Duke University. “Virtual Computing Manager.” https://vcm.duke.edu/ (accessed Mar. 1, 2023).

[4] Rhizome. “Rhizome to Restore and Present Theresa Duncan CD-ROMs.” https://rhizome.org/editorial/2014/nov/18/announcing-theresa-duncan/ (accessed Mar. 1, 2023).

[5] A. Gaitonde. “Introduction to Containers: Basics of Containerization.” https://medium.com/geekculture/introduction-to-containers-basics-of-containerization-bb60503df931 (accessed Mar. 7, 2023).

[6] Homebrew. https://brew.sh/ (accessed Mar. 7, 2023); NCSU-Libraries. “bd-brewfile.” https://github.com/NCSU-Libraries/bd-brewfile (accessed Mar. 7, 2023).

[7] B. Dietz. (17 Nov. 2021). Lightweight Distribution of Tools. Presented at 2021 BitCurator Users Forum. [Online]. Available: https://docs.google.com/presentation/d/1-y2tVJc6TOsV4Ahb-gAJaAAJKxje-obHIb7ZnQW6P4I/edit#slide=id.gf344f557f2_2_50 (accessed Jun. 22, 2023).

[8] Docker Hub, “Homebrew.” https://hub.docker.com/u/homebrew (accessed Jun. 26, 2023).

[9] Kali Linux, “Containers.” https://www.kali.org/get-kali/#kali-containers (accessed Jun. 22, 2023); Docker Hub, “Kali Linux.” https://hub.docker.com/u/kalilinux (accessed Jun. 26, 2023); Kali-Meta, “kali-tools-forensics.” https://www.kali.org/tools/kali-meta/#kali-tools-forensics (accessed Jun. 26, 2023).

[10] Docker Hub, “Ubuntu,” https://hub.docker.com/_/ubuntu. (accessed Jun. 26, 2023); Docker Hub, “Fedora.” https://hub.docker.com/_/fedora (accessed Jun. 26, 2023).

[11] E. Arroyo-Ramírez, et al., “Speeding Towards Remote Access: Developing Shared Recommendations for Virtual Reading Rooms,” in *The Lighting the Way Handbook: Case Studies, Guidelines, and Emergent Futures for Archival Discovery and Delivery.* Stanford, CA: Stanford University Libraries, 2021, pp. 141-167. [Online]. Available: https://doi.org/10.25740/gg453cv6438 (accessed Mar. 7, 2023).

[12] National Institute of Standards and Technology, “air gap - Glossary.” https://csrc.nist.gov/glossary/term/air_gap (accessed Jun. 27, 2023)

[13] Microsoft, “Understanding the Remote Desktop Protocol (RDP),” https://learn.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol (accessed Jun. 27, 2023).

[14] L. Sandecki. (14 Mar. 2023). “Time to prepare for Ubuntu 18.04 LTS End of Standard Support on 31 May 2023.” https://ubuntu.com//blog/18-04-end-of-standard-support (accessed Jun. 30, 2023).

[15] S. Black, B. Dietz, and Farrell. (12 Jul. 2022). Virtual Computing for Digital Special Collections. Presented at 2022 Triangle Research Libraries Network (TRLN) Annual Meeting. [Online]. Available: https://docs.google.com/presentation/d/1Lxka4dweKIBON3ctx7zzbrfsO0yVGMP7WKAPW-59PKk (accessed Jun. 28, 2023)