ALA American Library Association

CLOUD SERVICES FOR DIGITAL REPOSITORIES

Jarrod Bogucki

Library Technology Reports
Expert Guides to Library Systems and Services

JULY 2021
Vol. 57 / No. 5
ISSN 0024-2586

Library Technology
REPORTS
Expert Guides to Library Systems and Services

Cloud Services for Digital Repositories

Jarrod Bogucki

ALA TechSource
American Library Association

Library Technology
REPORTS

ALA TechSource purchases fund advocacy, awareness, and accreditation programs for library professionals worldwide.

Volume 57, Number 5

Cloud Services for Digital Repositories
ISBN: 978-0-8389-4825-5
DOI: https://doi.org/10.5860/ltr.57n5

American Library Association

225 N. Michigan Ave., Suite 1300
Chicago, IL 60601-7616 USA
alatechsource.org
800-545-2433, ext. 4299
312-944-6780
312-280-5275 (fax)

Advertising Representative

Patrick Hogan
phogan@ala.org
312-280-3240

Editor

Patrick Hogan
phogan@ala.org
312-280-3240

Copy Editor

Judith Lauber

Production

ALA Production Services

Cover Design

Alejandra Diaz and ALA Production Services

Library Technology Reports (ISSN 0024-2586) is published eight times a year (January, March, April, June, July, September, October, and December) by American Library Association, 225 N. Michigan Ave., Suite 1300, Chicago, IL 60601-7616. It is managed by ALA TechSource, a unit of the publishing department of ALA. Periodical postage paid at Chicago, Illinois, and at additional mailing offices. POSTMASTER: Send address changes to Library Technology Reports, 225 N. Michigan Ave., Suite 1300, Chicago, IL 60601-7616.

ALA TechSource

About the Author

Jarrod Bogucki is the cloud and IT architect for the University of Wisconsin–Madison Law School. He is a librarian and cloud technology professional, and his work is focused on library technology, cloud systems administration, and data discovery. He is the lead programmer and systems architect for the University of Wisconsin Law School Digital Repository.

Abstract

A digital repository can provide a library or similar institution the capability to offer patrons a variety of media and rich cultural collections. Repositories can be robust, valuable resources, but for a library they can be large and potentially difficult to create and manage. Cloud resources offer a wide range of tools and services that can be used to build a repository of any kind and manage it in a sustainable, successful way. Subscription services, development tools, and virtual infrastructure can be used to leverage existing repository software or build a custom repository to exact specifications. Consider the capabilities and shortcomings of cloud resources when creating a digital repository.

Subscriptions

alatechsource.org/subscribe

Contents

Contents, continued

The Rationale for a Digital Repository in the Cloud

A library and a digital repository are concepts that, despite existing in two fundamentally different places, provide many of the same services and functions. If a repository is "a place, building, or receptacle where things are or may be stored,"[1] a digital repository serves the same purpose for digital things (or digital objects). Libraries and other institutions, such as archives, historical centers, and museums, contain collections of documents, art, and other materials and objects that are considered to have some significance. A digital repository may contain collections of the same significant objects, accessed on the internet and displayed as digital formats.

Building a Digital Repository

A digital repository can serve as a virtual space for gathering and sharing objects of interest and importance, where they can be searched, studied, and enjoyed at any place with access to the internet. The need for such spaces has become much more obvious as the world grapples with the repercussions of the COVID-19 pandemic; the restrictions, closures, and other lockdown measures implemented to keep people healthy made it difficult or impossible to experience many places and cultural events. Travel was restricted. Institutions such as libraries and museums were closed. Community gatherings, religious ceremonies, and other celebrations were moved online or cancelled outright. What's more, schools and other educational organizations were unable to meet in person as frequently or at all. All of this has made sharing and learning about the values of other cultures an unusual and sometimes difficult process.

Not only was access to culture more difficult due to the pandemic, the ability to work at libraries was drastically affected as well. The need for distance between people has forced the staff of many libraries to adopt distributed work practices and require staff to work partially or totally from home. As a result of fewer staff being on site to operate facilities and equipment, launching a digital repository by using traditional, on-premises hardware may be impractical or even impossible. The simple fact that some libraries have fewer open hours or are closed to the public entirely means that staff may be furloughed or eliminated from their positions, reducing the number of workers available to contribute to a repository project.

While the effects of the pandemic may be decreasing in parts of the world, there may be other occasions in the future where a library may have to pivot to adopt remote work practices. To address the twin needs of hosting cultural resource collections and working in remote and distributed environments, a library can consider creating a digital repository using cloud services and resources.

The Need for Cultural Collections

For institutions that have not already decided to build a digital repository, it may be helpful to consider the value in doing so. It can be a large and difficult undertaking, but the completed result can create an accessible and lasting gateway to a wealth of important resources that can benefit not only scholarship, but also personal health and well-being. Culture itself is a wide-reaching concept that encompasses many aspects of every group and society worldwide. Many of its numerous definitions include the beliefs, practices, and other features common among a group of people.[2] These aspects of culture are seen as meaningful to the societies in which they develop and likely have measurable benefits as well. A paper prepared for the Ministry of Tourism, "Culture and Sport of the

Province of Ontario," discussed the practical benefits of engaging the various aspects of culture: "Culture enhances our quality of life and increases overall well-being for both individuals and communities."[3]

By capturing the physical representations of culture as digital objects, an institution is offering a way of preserving and connecting to cultures in the same way that libraries, archives, and museums have done for centuries. Ancient and isolated cultures can be studied and shared to prevent them from being forgotten and can be viewed in context next to related research and resources available from educational and cultural institutions around the world. Additionally, the cultures of disadvantaged, suppressed, or ancient and extinct peoples can be made visible and shared, with the intent of bringing to light their many contributions that have been stolen, changed, or ignored. By using photography, rare-book scanning equipment, and other specialized digitization techniques, digital objects can be created of fragile and rare artifacts without destroying them or removing them from their rightful owners. And unlike its physical counterparts, a digital collection can potentially be accessed from any location and can be made discoverable through searching and browsing techniques that are otherwise impossible.

Building a Repository with Cloud Infrastructure

Digital repositories (or cultural repositories, digital archives, etc.) extend data preservation and discovery out from the physical world and into its virtual counterpart. This can be an exciting prospect for many collections; the depth and variety of material housed in cultural heritage projects can lend itself to a wide variety of media formats, interactive applications, and interconnected discovery tools, not to mention the near limitless scale to which these collections can grow. Such a possibility can present a library with an exciting creative opportunity but also a potentially daunting project. Yet the recent proliferation of what is being called "cloud technology" offers libraries the tools to best demonstrate the importance of their cultural and scholastic collections. Cloud technology can provide an institution access to existing software to quickly deploy and easily operate a digital repository, and it can allow another institution to create a bespoke platform to craft a custom-made repository to meet precise needs and specifications. It is a powerful, flexible set of resources with which any library, university, cultural center, or other institution can find the best solution for sharing a cultural heritage collection.

The Need for Cloud Technology

The rapid adoption of cloud architecture in recent years speaks to its utility, but the jargon used to describe it may be confusing. Simply put, cloud technology is a collection of remotely hosted computer resources (i.e., resources in the cloud) available anywhere with internet access. In most cases, this means that large buildings called data centers contain all of the servers, storage space, and other hardware required to provide vast numbers of users with the means to do almost any computing task that was once possible only by having the necessary equipment on site. By using web browsers and specific software applications, users are provided with the ability to create servers and databases, manage network traffic, run custom code, and recreate the functionality of most common computer hardware in a virtual environment. Moreover, this technology provides users with pre-built, production-ready solutions to many common (and in many cases specific) IT challenges, including those presented by creating a digital repository.

Projects of every size can be completed using these tools, and there are solutions targeted toward customers with every level of IT literacy, from professional system administrators and programmers to librarians with great ideas for collections but little to no technical savvy. The scope of what is offered by the companies that manage these services (cloud service providers) is vast, perhaps to some even overwhelming. But it is because of this broad selection that an institution can develop its own digital repository, crafted to suit the needs of any collection. For all practical implementations of a digital repository, there is nothing that cannot be achieved with the tools offered by cloud technology providers.

The Value of Reduced Physical Spaces

It is the remote nature of cloud architecture that enables its far-reaching availability; a library no longer needs a server room to host websites or storage space, and cloud-based projects can be administered by a distanced, distributed workforce. This can be beneficial for several reasons, not the least of which is that by using cloud services, a project like a repository can be constructed under circumstances that require remote work or prolonged social distancing. This can allow for continuity of a repository project when staff cannot enter a library or office; even IT professionals no longer need to be present on site to make sure a server is successfully running the repository software, as this can all be managed remotely. The benefits continue during times of full office capacity as well, as the cloud's remote capabilities inherently provide the flexibility to house staff in distributed locations for any number of reasons, be it limited space or network bandwidth on site, limited access to locally available workforce or specific expertise, or an institutional push for work/life balance by providing work-from-home options to employees. Using cloud resources can potentially require significantly less power usage on site to run large computer equipment, and it can offer stability for those institutions where continuous power to run computers is not always possible. Whatever the reason or wherever the location, a repository can be developed, deployed, and maintained with complete access to the necessary tools and resources.

Notes

1. *OED Online*, s.v. "repository," accessed October 27, 2020.
2. *OED Online*, s.v. "culture," accessed October 27, 2020; *Merriam-Webster.com Dictionary*, s.v. "culture," https://www.merriam-webster.com/dictionary/culture.
3. American Sociological Association, "Culture," accessed October 27, 2020, https://www.asanet.org/topics/culture; Government of Ontario, "The Importance of Culture," in "Environmental Scan of the Culture Sector: Ontario Culture Strategy Background Document," April 2016, para. 2, https://www.ontario.ca/document/environmental-scan-culture-sector-ontario-culture-strategy-background-document/importance-culture.

Getting Started with Cloud Services

The phrase *cloud computing* is spreading from IT terminology into business language, consumer electronic marketing, and many areas of academia. It is used to describe many sites, products, and services that exist on the web, and this is not in error; the distributed networks of computer hardware that support cloud computing are capable of a vast range of functionality at a mass scale, all of which can be used in completely different ways by people all over the world, simultaneously. With such breadth of capability, it may appear difficult to apply a precise definition to the term, although the National Institute of Standards and Technology does provide a definition:

> **What Cloud Services Provide**
>
> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[1]

While this definition has perhaps been challenged and expanded upon since its creation, it does describe many of the traits pertaining to a large set of online products and services sold as cloud computing resources. These products are available online and on demand, are configurable, and are analogous to their locally hosted and physical counterparts. This specific definition continues to describe cloud computing in several ways, and while analyzing this definition is not an aim of this report, it may help to consider some of its elements to better understand the cloud services an institution may wish to utilize when building a digital repository.

Software as a Service (SaaS)

A basic function of the web is for users to connect to distant servers they do not own or operate, and server owners have long used professional computing power to run software or perform tasks that users cannot or will not perform on their own computers. Remotely hosted services such as website hosting, social media platforms, and web-based e-mail have long been available to users worldwide, served from data centers miles away from where they are consumed. This type of use is called software as a service (SaaS), web-based software, or many other different names, and it is one of the main categories of cloud computing. It is also the most accessible aspect of cloud computing for many users as it often requires little technical skill to use, making it particularly useful those who lack the skill or interest to utilize its more technical aspects. Without any knowledge of programming or systems administration, a fully functional repository can be deployed and operated by a single individual, utilizing up-to-date software and running on professionally maintained equipment in a secured location. This type of online software can be subscribed to through general cloud service providers or through private companies and nonprofit organizations that offer a specific piece or pieces of software to their customers to fill a specialized need. Such a cloud service may be of particular interest to institutions that wish to create a repository but have little or no expertise in IT; fully functional repository software is available to use immediately for any institution willing to pay a subscription fee. All that is required is some configuration and available content to provide a finished, though perhaps generic and lacking in features, repository to patrons.

Platform as a Service (PaaS)

For institutions that wish to utilize SaaS computing but have access to some programming expertise, it is possible to extend existing repository software or to create completely new software for a repository. This can allow for a greater level of customization than what may be possible using the out-of-the-box configuration options the software may offer by default. It may also provide an institution the opportunity to precisely realize its vision of what a repository should be. These customizations may include specialized functionality and interoperability that connects with data and applications already present in an institution's digital ecosystem, or they may include complete control of site branding and styling. Whatever the reason, a programming environment is necessary to perform such tasks. And while an organization may have the funds to hire a programmer, it may not be able to support such an environment on its local infrastructure. This may lead it to another category of cloud computing called PaaS, or platform as a service. This category entails using cloud-based tools to develop, deploy, and manage software applications and can be a helpful option for those unable to dedicate the time and effort to administering the systems required to facilitate professional application development. Many development tools may be included in PaaS, ranging from basic to advanced. Systems that manage code repositories, continuous integration, application testing, and software updates can be implemented and used without dedicating time, money, and staff power to running the computers needed to make these systems possible.

Infrastructure as a Service (IaaS)

For many institutions, a repository is one of many digital projects and services that are provided to patrons. Websites, e-commerce shops, discovery platforms, and online publications all exist digitally and require infrastructure somewhere to support their existence. Institutions large enough to offer a wide range of products have traditionally relied on in-house computing infrastructure for their creation and ongoing support. This infrastructure, coupled with on-site systems administration expertise, has allowed for the fine-tuning and careful control of these supporting computer processes, which in turn has provided speed, reliability, and availability to the users of these products. These infrastructure systems exist in closets, rooms, and sometimes dedicated buildings; use significant energy; and require special considerations for security and redundancy to ensure their continued success. Processing power, disk drives, tape backups, specialized servers—before cloud computing, all of this had to be physically present to be used for projects. Yet infrastructure as a service (IaaS) can bring much of this capability to the cloud, replacing the massive size and expense of owning this hardware with an ongoing subscription service. IaaS can replicate and potentially improve upon every piece of hardware that exists locally and may offer institutions access to (virtual) hardware that they may not have previously been able to acquire. For a repository, this provides a blank slate upon which to create, a vast and comprehensive selection of tools, and the potential to scale beyond what was previously restricted to the number of servers that could fit in a room. In other words, IaaS is raw infrastructure upon which any digital repository can be built.

As technology changes, so does the definition of cloud technology. While these categories are by no means an exhaustive description of all that cloud computing has to offer, they do describe what can be used to build a digital repository.

Specific Cloud Tools That Are Available

The number of types of cloud tools that can be used to build a repository is very large and constantly changing. Because of this, it is impossible to list everything that might be used, but it may be useful to learn more about some of the more common tools and services that may be applicable to repository projects.

Storage

Perhaps the most common and easily understood implementation of cloud services is what is known as cloud storage. Cloud storage can be easy to use for end users, and it provides a large number of benefits and is widely available across many platforms; many services selling cloud storage are commercially available and natively integrated with phones, tablets, and computers. These services offer a range of features and pricing models but at their core provide the same service: they allow users to upload a digital object to a remote storage space and download the same object at a later time. They have the practical application of giving a user access to more storage space than their devices can physically provide. Another benefit is specific to its cloud nature: the services can be used on multiple devices and are available anywhere the service can be accessed. Pricing for cloud storage varies from provider to provider, as does access speed, file versioning and management capabilities, and other built-in productivity tools like image manipulation and word processing integration. The interfaces for these services differ as well and can be a major factor in selecting the appropriate cloud storage solution for a repository.

When using cloud storage with a repository, it may be useful to find a service that offers means of accessing files beyond a graphical user interface (or GUI). A GUI is the typical interface, be it web-based or through a dedicated application, where a user can manually move files into a storage space and retrieve them later. This can be useful for small collections of files but would not be a practical solution for storing repository data; without an alternative form of access, cloud storage may not integrate with repository processes and features. For cloud storage to integrate with a repository, it needs a different type of interface, such as an application programming interface (API), an accessible directory (like that of a file system), a command line interface (CLI), or a programming language–specific development kit (SDK). With these integrations being used by a repository, files can be placed directly into cloud storage without any additional steps. And with repository objects placed into cloud storage, there will not be the same limitations of space on physical drives. Additionally, cloud storage offers inherent protection against on-premises disasters or power failures; the digital objects are stored at a remote location and will be accessible to the repository and, subsequently, the end user.

Software Subscriptions

As mentioned earlier, there is a selection of repository software available for immediate use, requiring little to no technological expertise. Not only repository software, but many other online applications that can be used in conjunction with a repository: search engines, media players, image manipulation software, and a variety of storage solutions can be added to a repository to extend its basic functionality. A cloud service provider may have an online store or marketplace that offers a large selection of applications, each available for subscription. Other SaaS offerings are provided directly by their creators, available on specific websites with instructions and support options. Sometimes software is provided in various tiers of service, which may offer different amounts of bandwidth, access to different features, and increased customization options. Relative to other cloud tools, SaaS tools can be easy to implement and use; the provider is responsible for operating the back-end infrastructure, installing updates, and dealing with security issues. Some providers will handle data migration and visual customization as well, leaving only the operation of the software to the customer. The subscription model does have the potential drawback of requiring ongoing payments for uninterrupted use of the software, but for many institutions this may be the most practical approach.

Servers

Traditional IT infrastructure has relied on servers to host projects such as repositories. Servers are computers specifically configured to run programs to be accessed remotely, such as a repository to be accessed over the web. They can be as powerful or as lightweight as needed, provided the up-front costs and space requirements can be satisfied. In a way they act as a blank canvas upon which to work; software, programming languages, databases, and other resources can be added and customized in almost unlimited ways to realize any vision. Cloud servers provide the same functions as their physical counterparts and can be accessed in the same ways. The main difference is their location, with the cloud version existing in a data center that can be accessed from any office, home office, hotel, or coffee shop. Additionally, cloud servers offer a flexibility not possible with on-premises servers, as they can be changed easily and quickly. When building a repository it is important to select the appropriate server for the project; some repository software runs only on certain types of server architecture, and depending on your existing architecture and institutional requirements, you may be limited in the type of server you can choose. It is also important to purchase the correct server in terms of power. If you buy an underpowered server, you may not be able to run your repository software optimally or at all. Conversely, purchasing an overpowered server to run a lightweight repository can also cause problems; the software may run smoothly but may waste unused resources at a potentially great cost. Fortunately, many cloud servers can be changed on demand, enabling increases and decreases in speed, power, and cost.

Databases

Databases are specialized programs used to store data in such a way that one piece of information can be related to another—it is for this reason they are also referred to as relational databases. Fundamentally they exist to store data for access and retrieval; they are designed to deal with large amounts of information and are built to be queried and searched. They are a central part of application design, and as such they are an important part of any cloud services suite of tools. There are many different kinds of databases, although many of the most popular options share basic similarities. Notably, many databases are designed using the Structured Query Language, or SQL, which is a standardized way of creating database queries. SQL databases are very common, and there exists much documentation regarding their use. There are other database types as well, each with its own optimizations and special functions. Cloud service providers usually offer a selection of different database types, with options to customize size, speed, and redundancy. A digital repository can contain vast amounts of metadata to support its collection of objects, making it crucial to have a fast, reliable database supporting its operations. Some repository software may allow for a choice of databases, while others are built to rely on the specific functionality of a particular database type.

Resource Scaling

As mentioned before, an application hosted within an in-house server environment is limited to hardware that exists in the server room. There is a finite amount of memory, storage space, and processing power available to the application, and this cannot be changed without a potentially difficult and expensive hardware migration. For all practical purposes, this problem does not exist in the cloud. The resources available through any large cloud service would far exceed the needs of any digital repository, and with this surplus of computing power a repository can be set up to utilize resource scaling. With resource scaling tools, a repository can be designed to "scale up" or "scale out" in times of high user demand and to "scale down" when there is little or no demand. What this means is that a server experiencing a high traffic load can increase its power (via a faster processor or by putting multiple redundant servers to work on the same job) to maintain speed and functionality, and when the load has decreased this power can be reduced or deactivated to save costs. These scaling features can be set to occur dynamically (that is, scaling occurs automatically when certain usage thresholds are achieved), or they can be scheduled to accommodate known periods of high traffic. Scaling, both automatic and manual, can be a special set of functions that are integrated directly into cloud servers.

Load Balancing

Load balancing is a term used to describe the management of internet traffic directed to computing resources. Heavy-use applications can sometimes get slow or come to a stop when the amount of traffic becomes too high. If this is anticipated, these applications can be created in such a way that they run on multiple servers. When traffic becomes too high on one server, it can be directed to a different server with little or no traffic, using what is known as a load balancer. A load balancer is relatively simple to implement in the cloud and offers a number of benefits to a computing environment. The primary function of a load balancer is to route internet traffic to servers based on how much they're being used, but depending on how it is implemented, it is capable of more advanced features. For example, a load balancer can route traffic based on the security settings of a server (e.g., routing all http traffic to an https server) or by specific URLs, or it can be used to move traffic away from servers that appear to be malfunctioning. Additionally, this service may be integrated with security features such as certificates and may capable of managing the URLs of the various resources included in a cloud infrastructure, effectively serving as an entry point for a set of public-facing sites, tools, or repositories. Load balancing is perhaps not necessary for smaller repositories but can be a valuable addition for any large or complex repository.

Containers

A relatively recent trend in application development is called containerization. Containerization is a means of running applications in discrete, isolated spaces known as containers on a large server. In this way, one large server can run many applications (i.e., containerized apps), allowing for server resources to be easily directed to the app or apps with the largest demand. This method of deploying applications also allows for rapid deployment, enables dynamic scaling of applications, and creates the possibility of easily deploying development or test versions of your repository. Container systems perform many of the functions that exist as separate cloud services , but can be easily managed and accessed as a standalone service. Containerization has the added benefit of reducing the need to understand and maintain server infrastructure; the underlying server architecture is managed by the cloud service provider so repository developers can focus on building an effective application. It is in the development process where an application can be designed to take advantage of these services, and by adding the correct code and structure it can run on a container platform. Or like SaaS applications, containers can be subscribed to and implemented with little time and effort.

Remote Workstations

Traditionally, employees at a library would all work in the same physical space. Considering social distancing requirements, this paradigm has changed; a workforce may likely be distributed across great distances throughout numerous varied locations. Some of this distribution may be due to safety concerns, business considerations, or the preferences of workers. In the case of a repository, there is another possibility that workers may need to be near numerous sites in order to gather data and digitize resources as quickly as possible. Naturally, decentralized workers require computer hardware to perform any work that involves accessing the repository directly. Likewise, using third-party software applications also requires some type of device, be it laptop, tablet, or even phone. These programs may be required for workers to record data, take and edit photos, capture audio and video, or write code to be later uploaded. For a large workforce, hardware management can benefit from the use of specialized cloud-based software to track inventory and administer remote updates. Similarly, the software installed on these machines can also benefit from centralized management tools; to ensure that all users are up-to-date with bug fixes and new features, specialized software can communicate with these remote endpoints and deliver patches and updates in a scheduled, automated manner.

For some institutions, greater control of the worker's digital environment can be useful. This can be achieved through a cloud service called remote workstations. By using this service, a desktop environment can be created to meet any required configuration and can be logged in to where an internet connection exists. Through a remote desktop connection client, applications that are needed to perform assigned tasks can be accessed by a user without being directly installed on the user's laptop. This offers several advantages, one being that a smaller institution can provide cost-effective, lower-powered hardware to its workers, or it can avoid managing hardware entirely by requiring workers bring their own device to log in to the remote workstation. It also becomes much simpler to update software or lock down access to unnecessary or malicious sites or programs for a large group, as the changes made to one desktop profile will affect any users who use it. This allows the creators of a repository to provide workers with access to the exact set of applications and services that are required to perform their job duties.

Note

1. Computer Security Resource Center, "The NIST Definition of Cloud Computing," September 2011, https://csrc.nist.gov/publications/detail/sp/800-145/final.

Designing a Repository in the Cloud

Once the decision has been made to use cloud resources for a repository, there are many factors to consider regarding the practical implementation. Cloud technology is a growing, changing set of resources, and special attention should be paid when deciding how best to use this technology during the design phase of the repository project.

Selecting Cloud Service Providers

At the time of this writing, companies supplying cloud-based services are prolific online. Varying services exist for small businesses, large businesses, and individual users, catering to the specific needs of schools, libraries, historical societies, and many other groups that may be considering creating an institutional repository. Cloud technology is a quickly shifting business landscape with new companies opening and closing regularly, and the selection of offered services and subscriptions is rapidly changing. The dynamism in this space may make it difficult for prospective users of cloud technology to commit to a company or its services, as they could appear unstable or complicated. It also makes the recommendation of specific companies or services undesirable for this report, as these companies are frequently starting, changing, and closing. There are, however, general considerations that can be useful when selecting cloud services.

If an institution is part of a larger entity (e.g., a school being part of a university or consortium of other schools), there may be guidelines in place that could restrict the available options. Some universities may require a smaller entity to follow purchasing rules already set in place, such as using only approved vendors or seeking price estimates from numerous vendors. When a member of a consortium or some other association of similar entities, there may be interoperability standards or shared pricing benefits that could incentivize the use of one service over another.

If IT expertise is limited or if the project plan directs staff hours to aspects of the project other than coding or IT management, it may be useful to consider subscribing to a pre-built repository service. These services may not offer the customization options of building a repository from the ground up, but they can prove to be much faster and simpler to deploy and may yet offer some configuration possibilities while retaining the core functionality necessary to show off many types of collections. Depending on the vendor, they may also provide usability enhancements and feature updates, academic or nonprofit pricing models, and e-mail or telephone support. On the other hand, for an institution with IT resources available to devote to a repository, it may be desirable to plan a complex and ambitious project. And when a project is planned to offer more than a few advanced features, it may be worth considering a cloud provider that offers many different services. Using such a provider can allow developers to leverage special tools to connect wide-ranging functionality within a single site, to seamlessly include this project with any existing or future projects, and to fine-tune many aspects of the repository.

There may be reasons to use multiple cloud service providers when developing a repository. Some institutions have preexisting contracts with numerous vendors, and the tools these vendors supply can be leveraged with preexisting support and without additional costs. This can be especially true for larger institutions where different projects may be dependent on unrelated cloud services. In some cases, there may be specific services that fill uncommon, niche needs and are not widely available across multiple providers. For example, an institution may have a preferred vendor for storage space for digital objects, another vendor for remote workstation access, and yet another that provides a SaaS subscription to a specific type of repository software. This list may also include separate cloud services that are used to support staff who are working on the repository, such as cloud-based word processing, spreadsheets, or development tools.

Planning and Project Management

Before making decisions regarding the creation of a digital repository, create a plan that outlines each step of the process from start to finish. A plan will increase the likelihood that a project successfully reaches completion and achieves the objectives envisioned at the start of the project.[1] Because repositories widely range in terms of capability and complexity, there is no single strategy that can be adopted to plan for all repository projects. For smaller repositories, informal project management may be sufficient to complete the project; simple lists of goals and responsibilities may be enough for some individuals to move forward with the tasks ahead. For larger repositories, using a dedicated project management philosophy be invaluable for managing the teams, tasks, and resources involved in the project.

There are many established principles and techniques for project management that can be applied to repository projects, and it is up to the creators to decide if one of these techniques will be appropriate for their team. Project management is an established discipline, and as a result there are many differing opinions regarding which philosophy to implement. Ask colleagues and coworkers if they use a project management standard for any other projects in your institution; many of these standards are universal and can be applied to many types of projects, including the creation of digital repositories.

Gathering Requirements

Before repository development can move forward, stakeholders must decide what is required for the project to be considered a success. Just as the size and the scope of a repository can vary widely, the number of people who are responsible for the success of the project can range from one dedicated individual to a rotating team of professional historians, librarians, and computer programmers. Additionally, there may be institutional administrators, community groups, financial sponsors, or other external organizations with a vested interest in the repository, each with their own criteria for the project. It is important to gather these expectations as early as possible in the project so each need can be given the necessary time and attention to be completed. Early requirements gathering also allows the project managers to ensure that the project is running on time, with as few surprises as possible along the way.

There are many different criteria by which a project could be considered a success, although some specific types of criteria may commonly apply to libraries and similar institutions. Some of these requirements may include

- grant requirements
- accreditation standards
- cost requirements
- requests from project sponsors or major stakeholders
- existing business demands

When considering the use of cloud architecture, gather any requirements that may pertain to the adoption of prospective cloud services. Cost, available bandwidth, local regulations, ease of use, and existing contracts can all inform the decision to sign a contract with a cloud service provider. IT or administration departments may be able to identify these requirements before moving forward with the project.

Policies That Affect the Repository

Every site and application that is accessible through the web needs to meet basic accessibility standards. This is to ensure that the repository content can be viewed by all people, including those who may require screen readers or other software to access web content. There are several publicly available standards and laws that may guide an institution to utilize special coding practices; provide subtitles, transcripts, and textual descriptions with multimedia; and structure the layout of content in a repository in an effort to make content universally accessible.[2] Not only do these practices make a site more inclusive, they improve the general usability of the site for all users across various platforms.[3] There are numerous products, both free and for cost, that will check a site for its adherence to accessibility standards. These products can print easy-to-read reports or get into the small details so a site can be audited, evaluated, and improved. Additionally, some repository software comes designed with certain accessibility standards in mind, with the appropriate tags and structure written into the code so a user can focus on other aspects of the project.

An institution may adhere to other standards or regulations that could influence how repository content is structured and displayed. For example, health, legal, educational, and other public institutions may be required to adhere to different privacy and accessibility standards, some of which may necessitate that the repository use different security settings. Educational institutions are often guided by differing standards for varying disciplines, each potentially necessary to maintain accreditation or achieve eligibility for grants and other funding. Some publicly run repositories may be obligated to share or report certain aspects of their data or to include features in their site beyond the accessibility standards. Or a repository may be designed in such a way so that it can be interoperable with a software project or initiative run by another institution. Some requirements may pertain to cloud data specifically. For example, to be compliant with accreditation standards, an institution may be required to host some data locally as opposed to hosting it on remote cloud resources. When creating a repository, be sure to consult any specifications regarding these standards when making design decisions or developing features.

What Is Important to Share

It is not the place of this report to discuss the importance of any cultural collection, nor to discuss which specific objects are worth including in a repository. Yet some institutions may have a large amount of potential resources available, and for this type of project it is important to be deliberate when deciding which content to share. Considering the costs associated with cloud services, programming time, and digitization efforts, many institutions may have to choose where to place their efforts as they begin development. Consider the following as guidelines when deciding what to include in a repository.

The following are potential types of items to share in a repository:

- resources or collections that directly support the mission of the institution or pertain to its history
- underrepresented groups
- local or regional historical people and entities
- regional groups of high size and significance
- items of interest to known users of the repository
- any other rare or unique object or collection

The following types of resources offer less value or should not be included in a repository:

- personally identifying documents
- things commonly found on the internet
- anything for which the creator, copyright holder, or other relevant party asserting ownership has not granted permission for sharing
- anything that is otherwise illegal

Assessing Available Infrastructure

Creating a custom repository is no small feat and will require some technical expertise. Before beginning a repository project of any scope, it is useful to understand which resources and tools are available, as well as the size of any existing IT infrastructure. Any institution with a web presence at all already has infrastructure of some kind, though it may be slim and ill-equipped to handle a large IT project. Large institutions may have an existing physical or cloud infrastructure of immense scale capable of supporting many large and complex projects beyond even the grandest plans for a digital repository. On the other hand, small institutions may have a few machines to manage all of their computing projects and tasks. It is perhaps these institutions that would see the largest visible improvements from adopting cloud services, as they can offer a range of tools that were physically or fiscally inaccessible in the past.

Large institutions may have existing cloud resources and IT expertise that can be utilized. If this is the case, adding the necessary resources for a repository project may be a simple matter for specialized IT professionals. There may also be existing physical resources to use, which could result in having to spend less on cloud services. In either case, much of the initial exploration (and potential guesswork) of using cloud services can possibly be addressed by a team that understands the existing IT infrastructure. Such an advantage highlights the fact that staff is a crucial consideration. Some institutions have staff already dedicated to supporting IT, potentially teams of workers dedicated to programing, performing system administration duties, and supporting staff and end users. Some may have no IT staff at all and would consider hiring new staff or outside consultants to complete some or all of the required work. The type and number of technical staff to devote to a repository project depend on its size and scope, the amount of money available to spend on the project, and the other IT projects an institution may already be obligated to complete. Nevertheless, it is important that an institution at least understand some of the general IT concepts surrounding its repository; some basic knowledge makes it easier to talk to salespeople, read documentation, and communicate with customer support agents.

Types of Media

With the increase of broadband usage over time, it has become more feasible for institutions to share content with patrons.[4] This is especially useful for digital repositories, which may often include high-resolution images and sizable video and audio files in their collections. When considering the types of media that are to be put on virtual display, an institution should have some understanding of the capabilities of its infrastructure and its available IT expertise. Still images at lower resolutions can be relatively straightforward to embed in a web page and can require very little in terms of computing resources to properly display. Other forms of media, such as audio, video, and high-resolution images, may require additional processing power and faster retrieval speeds to effectively render. These forms may benefit from detailed analytics tracking, cutting-edge display tools, and advanced integration functions to ensure an optimal user experience and reliable operation. They may also have larger file sizes, which could require additional storage space and disk speed. Naturally these considerations will affect the cost and complexity of any cloud infrastructure.

There are some design strategies that can be implemented in a repository to enhance discoverability and usability. For audio and visual content to be discovered, it must have associated metadata. This can describe not only the content of the media, but also the media itself; file type, size, color profiles, compression, and checksum validation can all be included and searched upon if these fields are available in the metadata. Some images may contain text, or video and audio files may contain embedded transcript, subtitle, or translation text. This data can be copied from the digital object and used in a search index, or some repositories may have modules or tools to access this textual data directly. Some tools exist to highlight searched text within images, and others to queue audio and video content to the exact second when searched text appears. It may also help to consider whether the intended audience has access to high-speed internet. This may influence the decision to use thumbnails or smaller derivative images when returning browse or search results, or whether to preview images before serving the large content to users.

Money

Like physical IT resources, cloud resources do cost money. And also like physical IT resources, the cost of cloud resources increases with the power, speed, and availability required by the project. These costs are often significant and can dramatically determine the shape of the finished repository; when using on-premises hardware for a repository, an institution should (and in some cases must) purchase all of the hardware that is needed for the project before creation can begin. When physical hardware is purchased up front, the hardware itself cannot become more powerful or upgraded without the purchase of additional hardware. That is to say that a hypothetical server, one that is capable and required to run specific repository software, contains a motherboard, a processor, memory, internal storage, and many other different physical components. Unless these components are manually replaced by a knowledgeable IT practitioner, the server will never get any faster, never grow in storage capacity, and never grow to adapt to the computing needs of the future. In fact, the opposite is likely to happen; components will become outdated and potentially break over time. Even the most reliable and advanced physical infrastructure will become outdated and unable to run cutting-edge software and will someday become less resilient to unexpected increases in traffic. Due to this lack of versatility, on-premises purchasing can greatly benefit from a detailed understanding of the hardware requirements of a planned repository, and even then, it shows weakness when faced with unexpected challenges and requirements. Many institutions may not have the money available to purchase anything more than what meets the basic requirements of the necessary software, and without more powerful hardware to adapt to future changes, a repository can be locked into its initial size and capabilities.

Unlike physical infrastructure, cloud infrastructure can easily change, grow, and improve—if an institution is willing to pay. Cloud resources can be lightweight and ephemeral, and the granular way in which charges are accrued can allow for precise control over infrastructure expenses. For example, if there is heavy anticipated use surrounding an institutional event, an increased amount of processing power can be purchased for its duration and then reverted to its baseline power level when the event is over. Managing an on-premises server environment with this level of precision and flexibility is often impractical or impossible, and it may not be possible to implement such changes from a remote work location. With cloud computing, these changes can be fast and easy to implement. These capabilities can also greatly reduce the need for large up-front purchases of technology resources; because upgrades can occur so quickly and easily, there is no need to future-proof a system by purchasing more than is needed at the start of a project in anticipation of later growth. If a repository needs more processing power, a cloud user can simply buy more processing power.

To understand of how cloud service charges may work, consider the following example. Consider a hypothetical function that aggregates a library's monthly additions to a repository and produces varied outputs, such as a list on a web page, a mass e-mail, and an RSS feed. This service may need to operate only once a month for about five minutes at a time. Given the low level of frequency, it would be cost-inefficient to spend money on hardware that would run continuously to facilitate the operation of a task that runs for less than an hour a year. Yet with cloud technology it is possible spend only for the time and the amount of the resources that are being used. When the function is running, the institution is billed for each of the services required to make the function perform its tasks. If the function is shut down when it is not being used, an institution may pay nothing. Please note the exact billing structure varies between cloud service providers, and be certain to review any contracts signed to ensure there are no unexpected charges.

Digitization

It goes without saying that the content added into a digital repository must be digital. Somehow, valuable cultural heritage in the physical world must be captured, processed, and presented through a virtual platform. This process is called digitization, and it involves using specialized equipment to produce digital representations of images and sounds. Generally speaking, this equipment includes cameras, video and audio recorders, and document scanners, usually built to achieve higher levels of fidelity than their commercial-level counterparts. And because much of the digitization process depends on the hardware used to capture a particular resource, it is not a process that can be replaced by cloud computing services. Still, cloud-based software can play a role in aiding the digitization process. For example, because cloud storage can be accessed from remote locations, records can be off-loaded from the physical storage media that many cameras rely on, creating the potential to free up local storage space on a device as it is being used. In a similar way, cloud versions of popular editing software enable users to make changes to media quickly and from nearly any location. Document management can also exist in the cloud, thereby distributing the work of managing what can be a large collection of data and files. Newly created digital objects can be tracked on something as basic as a shared spreadsheet, or they can be ingested into an online content management system (CMS). Using a CMS to manage digital content can offer a number of advantages, such as detailed metadata association, versatile search and discovery capabilities, and the potential to grant access to the objects to other institutional stakeholders such as website teams, marketing departments, donor networks, and event coordinators.

Software Development Tools

For an institution with the resources to develop and maintain original repository software, there are many advantages in doing so. A custom repository can potentially contain every feature and design that an institution can imagine, and it can be updated, altered, and fixed without waiting on a third-party developer. It can also be a time-consuming and difficult process that may not be practical or even possible for many institutions to attempt. Still, for those that are technically capable of doing so, there are a number of cloud-based tools that can be used to write the code for a repository and to integrate it into a larger cloud environment.

While code for applications can be written using many different methods, some programmers choose to use an integrated development environment, or IDE. An IDE is a program in which a developer can write code, but it may also provide many other features that can be used to aid in the development process. An IDE can check for various errors, perform tests, integrate with source control code repositories, and provide previews of how an application will behave at runtime. Many provide built-in support for specific programing languages and development frameworks, and some are targeted at writing code for particular operating systems or consumer devices. An IDE can be useful when managing the code for many different applications, as it provides a unified view and tool set that can be used across many projects. Some IDEs are open source, freely available applications, but some are now available as subscription, cloud-based services. There are a wide variety of strictly cloud-based IDEs, each with different costs and capabilities. In contrast to dedicated IDE applications, cloud-based IDEs can exist completely within a web browser and can be run on a variety of operating systems. Cloud-based IDEs can provide special tools to create applications designed to take advantage of cloud technology, including direct access to related cloud resources and specific tests for cloud-based infrastructure. Some of these IDEs even allow for collaboration with team members by using tools built directly into the software itself.

Cloud technology can offer benefits to groups that are developing repository software to be distributed to other institutions. Some cloud service providers offer CI/CD (Continuous Integration/Continuous Deployment) services, which are a collection of functions that automate testing, building, updating, and deploying code. Different from an IDE, these functions are focused on automation and production-scale development. They allow teams to manage the many code changes that occur when teams are collaborating on a project as large as a repository and to deliver updates to repository users with the confidence that they have passed the necessary checks and tests to be used in a production environment. And like cloud-based IDE services, cloud CI/CD services may natively integrate with related cloud tools and services, allowing for easier deployment into existing cloud architecture.

Content Discovery

Digital repositories can contain a wealth of cultural artifacts and resources, but without an effective discovery layer these resources have limited value. A discovery layer is "a searchable meta-index of library resources, usually including article-level metadata, e-book metadata, metadata from library catalogs, open access resource metadata, etc., and it includes a means of retrieving resources in the result set through linking technology,"[5] or in other words, it is the collection of programs used to facilitate the discovery of records in a catalog or repository. It is a crucial part of any repository, as it is the means by which users search for and retrieve objects and their associated metadata. There can be many different pieces of software that power a discovery layer, each serving a different yet related function. Functionality is added to make repositories compatible with the increasing number of discovery standards that are being utilized throughout the web, and with the right cloud tools available, this can be done to precisely meet the needs of the project. The evolving nature of cloud technology makes it an appropriate place to run such services; data discovery is a dynamic field that can be dependent on the changes brought about with the introduction of new technology.[6] Cutting-edge services are offered on many cloud service suites, including services specific to searching and discovery. As online discovery tools offered by popular search engines and scholarly databases have changed over time to meet the habits and expectations of users, the backing technology of a repository discovery layer must be capable of changing to meet the expectations of its users. And given that cloud technology often lends itself to these types of changes, using it to build a discovery layer can be practical choice.

How the data in a repository is structured can have a tremendous impact on the discoverability of its content.[7] To help users understand the specific details of the content included in the repository, special care must be given to the way its data is structured and described. Important properties, common aspects, and unique attributes of digital objects can be placed into discrete data fields. When structured in this way, information can be more easily searched, filtered, and analyzed. The specific way of structuring repository data is entirely up to the creators. However, it may be useful to find existing data standards and apply them to a collection. When an existing standard is used, repository content can be more easily compared to other repositories and data sets, as its content can be matched to other repositories using the same definitions. This allows for interoperability between a repository and other applications, and it facilitates shared data projects with other institutions.

Another way of making textual content discoverable is by making it searchable using optical character recognition, or OCR technology. In the digitization process, a page of text is captured as an image, and while it can be read by humans, it cannot be searched using a computer. OCR software can identify letters (and in some cases, words) contained in an image of text and create encoded, embedded characters that can be understood by computers and searched upon by users. As a result, repository content that has gone through this process becomes full-text searchable. This technology is capable of recognizing words in hundreds of languages and is continually improving in its ability to understand page layout and deal with speckled and skewed documents. OCR software is available through several cloud service providers and can be used as digital objects are created in the digitization process.

One component of discovery is wayfinding. Wayfinding is a concept that existed before computers, using landmarks, markers, and paths to navigate through spaces and to ensure arrival at an intended destination. These concepts can be applied to a digital repository in the way that site navigation is used.[8] With the appropriate use of navigation bars, breadcrumb trails, and footer content, a repository can quickly guide a user to resources or tools of general importance or to objects or collections that an institution wishes to showcase. Solid wayfinding will allow a user to make discoveries, use additional site features, then return to a previous page without time wasted on unnecessary searches or clicks. Some repository software has these features built in, while other software may require extensions or custom programming to create this functionality.

Depending on the design of the chosen software, different database types may be used to manage the data that resides in a repository. A common type of database is called the relational database. There are numerous versions of relational databases, but in general they consist of data tables, each with properties to describe entries in the table. These tables can (and often do) relate to each other, hence the term *relational*. Data from these databases must be queried to be retrieved, and each database uses a language or set of rules to build queries. A common query language is the Structured Query Language, or SQL. SQL is nearly synonymous with relational databases as it is widely used for many types applications, repository software being no exception. In many instances, an SQL database is used to manage the actual functionality of a site or repository in addition to storing information about digital objects; information such as site text, configuration settings, and the relationship of pages to each other is sometimes stored in data tables.

Despite the ubiquity of SQL databases, there are times when the default database being used by the repository to manage its data may not be the ideal tool for data discovery. SQL databases can occasionally suffer from slow speeds when dealing with complex queries or large result sets, and because they are so common SQL databases have become a common target for specialized hacking attacks called SQL injection attacks. And while security precautions and speed optimizations can be implemented to overcome these shortcomings, other database types offer features beyond what SQL databases provide.

Databases other than SQL are often referred to as NoSQL databases and range in their design and capabilities. There are reasons to use a NoSQL database with a repository project. The creators of a repository may wish to store and expose their data by adhering to a particular specification or structure. One such structure that is used in several publicly available repository software offerings is the graph database, otherwise known as Resource Description Framework, or RDF. An institution may use RDF to model or organize the metadata in a repository into what are known as triples. Triples are a way of representing information as "a fact on a thing being described (i.e., the subject, which is also referred to as the resource), on a specific property (i.e., the predicate), and with a given value (i.e., the object)."[9] Though a detailed explanation of triples is out of scope of this report, simply speaking they are used to express relationships between entities. They are designed to structure data in a way that is modeled after meaningful human language and are sometimes referred to as semantic databases. Using a graph database may allow for searching functions that more accurately respond to natural language queries, and they also allow for resources to be connected or "linked" together based on their triples. This type of database has become somewhat popular with digital collections projects and is natively integrated into some repository software.

Most databases types, be they SQL, graph, or others, are offered as stand-alone cloud services or can be implemented on a cloud-based server. Again, depending on an institution's access to IT support and infrastructure, it may make sense to create the repository database within an existing database architecture. Doing so can be efficient and fast, although there may be reasons to use a database service separate from any existing databases or to isolate it on its own server. If the software chosen for a project is developed by an outside company or institution, using it with a discrete, isolated database service may make sense for security purposes. Without knowing exactly how code is written, it may be difficult to tell if best practices are observed and if steps to protect the repository from database-specific attacks have been taken. Isolating the repository database decreases the chance of it being used as an attack vector for multiple IT resources. Database configurations may also influence this decision; if a repository is accessed much less frequently than other applications, a smaller, less expensive database could be used. Conversely, a larger repository may require a larger database with more storage space and faster data retrieval speeds.

Search boxes are perhaps the most common method of content discovery, at least in terms of digital collections. They are part of most discovery layers, as they are accessible to users with basic literacy skills. The idea is simple: type a word or phrase into a search box and the repository will retrieve results based on these terms. The results will be displayed to the user, and depending on the interface can be sorted, filtered, or exported into different file formats. Many large and diverse collections can benefit from advanced search features as well, which allow for combining search terms or using complex expressions to add additional specificity to search results.

Search boxes are sometimes powered by simple database queries, but other times they are run by a technology called search engines. In basic terms, search engines operate by performing two functions: indexing data and retrieving data through queries. Indexing is the process of gathering and storing data from pages and resources to later be retrieved. Querying is the retrieval function, and depending on the search engine, queries can be constructed using specific metadata fields, date ranges, and media types or by using full-text searching of digital files. Many different types of search engine technology are readily available as cloud services. Some can be fully operated on traditional server technology and integrated into a site through a programming interface. Others exist as containerized applications or SaaS subscriptions, models that can retain some customization capabilities but require less management than a stand-alone service. Lastly, there are large, public search engines that offer an extension of their search engine technology to be used specifically for site (or repository) level searching. This cloud service can be very easy to implement, to include analytics and relevancy adjustments, and can integrate with other cloud services. However, these options can come at an increased price, or the results display may include limited features or undesirable branding.

The starting place for many researchers is not a scholarly database or the repository itself, but a public search engine. Search engines can provide a convenient starting point for people who are not affiliated with a university or library, for researchers who live in remote locations, or for those who lack the training or knowledge to use the specialized search tools offered by scholarly institutions. For many, search engines are simply the easiest way to get started; people use them to find many things already, and it is a comfortable and familiar way to find new information. Some search engines even have functionality specifically geared toward academic research, and these tools can include digital repositories in their results. A repository can be built so its content can be more easily discoverable using public search engines, using a technique called search engine optimization (SEO). This can be desirable for institutions that wish to increase traffic to their repository. If for some reason an institution does not wish to make its repository content discoverable, special steps (such as placing a "Do Not Index" directive in a robots.txt file) can be used to prevent public search engines from indexing repository content and driving traffic to it via public web searches.

Security

With the proliferation of digital technology, the need for internet security continues to increase. Many sites on the web may ask for names, e-mail addresses, or some other identifying information, and just by visiting some sites users are supplying time, location, and browsing history to unknown parties. Some sites, such as medical and banking sites, may require private information, and other e-commerce sites may request and store credit card information. A repository may not require anything protected or especially sensitive from its users, or it may require names, location data, or an institutional ID to manage logins. It is necessary to secure this data, though it is the nature of the data that will determine the exact security precautions that should be taken. Regardless of the specific security concerns, cloud tools can provide the capabilities to protect data of any nature used by a repository.

Because people are voluntarily placing and accessing personal data online, people are placing trust in the technological systems and operators that are responsible for protecting private and sensitive information. To maintain this trust, an institution that is operating a repository must take every step to ensure strong security for any offered services. With cloud services, security measures can be implemented at several access points, allowing a repository to provide access to users while preventing the malicious actions of bad actors. For example, traffic to a repository can be restricted and directed so users can access the site only via secure, encrypted means such as an https connection. Other means of access, such as SSH, a method of connecting to servers often used by system administrators, can be limited to known users or closed completely; these other methods of connecting to sites are useful for workers who need to access advanced server functions, but they can also be used as points of attack by those who may seek to compromise the security of an institution. By reducing or eliminating access, fewer targets for attack exist. Furthermore, cloud security can offer the flexibility for system administrators to easily enable and disable security settings, allowing for access only during maintenance windows or for on-demand updating.

In addition to restricting access, security can be enhanced for a repository by logging user access, following best standards and practices when writing code, and auditing system software for necessary patches and updates. A cloud service provider may offer a range of security tools that complement and integrate with the other resources included in its services. These tools may include firewalls, security certificate and key management, and security auditing guidelines.

Notes

1. Ben Aston, "Why Is Project Management So Important to an Organization?" The Digital Project Manager, January 15, 2021, https://thedigitalprojectmanager.com/why-is-project-management-important/.
2. Shawn Lawton Henry, ed., "Introduction to Web Accessibility," W3C, last updated June 5, 2019, https://www.w3.org/WAI/fundamentals/accessibility-intro/; State of California, "Accessibility," https://webstandards.ca.gov/accessibility/.
3. Shawn Lawton Henry, Shadi Abou-Zahra, and Kevin White, eds., "Accessibility, Usability, and Inclusion," W3C, last updated May 6, 2016, https://www.w3.org/WAI/fundamentals/accessibility-usability-inclusion/.
4. Pew Research Center, "Internet/Broadband Fact Sheet," June 12, 2019, https://www.pewresearch.org/internet/fact-sheet/internet-broadband/.
5. Gwen Evans, "Good Question! What Is a Discovery Layer?" *Ohio Technology Consortium* blog, January 16, 2014, https://www.oh-tech.org/blog/good_question_what_discovery_layer#.X7A_1ihKgkg.
6. Don MacMillan, "Data Sharing and Discovery: What Librarians Need to Know," *Journal of Academic Librarianship* 40, no. 5 (September 2014): 541–49, https://doi.org/10.1016/j.acalib.2014.06.011.
7. Kamran Munir and M. Sheraz Anjum, "The Use of Ontologies for Effective Knowledge Modelling and Information Retrieval," *Applied Computing and Informatics* 14, no. 2 (July 2018): 116–26, https://doi.org/10.1016/j.aci.2017.07.003.
8. Mark Foltz, "Designing Navigable Information Spaces" (master's thesis, Massachusetts Institute of Technology, 1998).
9. Olivier Curé and Guillaume Blin, *RDF Database Systems: Triples Storage and SPARQL Query Processing* (Amsterdam, Netherlands: Elsevier Science & Technology, 2014), 43–44.

Deploying and Maintaining a Repository in the Cloud

Cloud infrastructure is, by definition, virtual infrastructure. The server instances, storage blocks, routing instructions, and every other cloud function are facilitated by an underlying code base; while these services are supported by hardware, it is code that defines and enables these resources to function in the same manner as their physical counterparts. And it is because they are created in code that every cloud resource can be accessed, configured, and controlled through a web interface, an API, or a command line tool. These various methods of control allow an institution to use a variety of sophisticated means of managing its infrastructure and the applications (e.g., digital repositories) that it supports.

Infrastructure as Code

Infrastructure as code (IaC) is a concept that became possible with the advent of hardware virtualization.[1] The concept can be described as using templates, scripts, or some other machine-actionable documents to describe a virtual infrastructure, and in doing so enabling the automatic instantiation of said infrastructure. In other words, defining cloud IaC allows for the automatic creation of cloud resources. The benefits of this practice are numerous; simply having all its IT resources described in a file allows an institution to practice detailed resource inventory management. Resources can be given names or tags that allow them to be tracked for purposes of cost accounting or performance analysis, allowing an institution to easily group and identify resources, and to identify spending on resources that are being underutilized or not utilized at all. This type of management also aids in process and resource documentation; each resource can be annotated within the code to link to troubleshooting documents, provide human-readable descriptions for processes, and explain the business and technical decisions pertinent to the infrastructure. Additionally, IaC can make it easy for resources to be audited for security considerations, as security and logging mechanisms can be described within the code.

While the resource management aspects of IaC are certainly valuable, it is the automation aspects of this practice that are perhaps its greatest strength. The *code* in Infrastructure as Code (accurately) implies that cloud resources can be created and driven by code rather than human intervention. Enterprise-wide cloud infrastructures can be created entirely within code, and this is also true for discrete applications such as digital repositories. Templates can be created to describe common resources that are to be applied in numerous places throughout a cloud infrastructure. For example, to ensure that a repository conforms to hardware standards, capabilities, and best practices, a template can define the specifications and customizations of the cloud server on which it is to run. This can include processor speed of a virtual server, encryption keys for storage volumes, an operating system version, and any number of other resource customizations.

Logging and Analytics

A digital repository is a platform for collecting and disseminating information. It makes specific sets of information available to users at remote locations, makes the information discoverable through searching and navigation, presents it through a web browser or stand-alone application, and provides users text, audio, and graphical information through digital means. All of this is done by means of computer hardware and software processes, processes that, due to their digital nature, can be configured to output a detailed record of their use and activity in a collection of files known as logs. Logs are simply files that list the activities occurring on sites, applications, and servers. They can appear as individual files located on servers, or they can be integrated with special logging services as part of a cloud services suite.

The software used to run a repository may generate several types of logs. It is possible that a repository will run on widely adopted web server software, such as Apache or Nginx. These servers (and those like them) generate logs that contain data such as a record of which pages and files are accessed, server health and status, and any server errors that may be occurring behind. It is also possible that a repository is using a database or a search index, both of which may have logging capabilities that could be used to discover what search terms users are entering into the repository search fields. Furthermore, repository software may have custom logging specifically to diagnose errors in application code or to debug the development of add-ons and extensions.

There is some general terminology that should be understood in order to begin interpreting the data being generated by a repository. *Analytics* is "information resulting from the systematic analysis of data or statistics," in this case being from a library or repository.[2] It can be thought of as the practical interpretation or human-readable form of the data that is captured in logs, although analytics themselves can be complex and potentially difficult to understand. Analytics are often displayed in tables or with visual aids that may make them easier to interpret. Analytics are comprised of *metrics*, with a *metric* being one specific type of data. Page views, time spent in the repository, the country that patrons are visiting from—these and more are all metrics that, when analyzed, provide an institution with a meaningful set of information from which business decisions can be made. Along these lines of business intelligence is the term *key performance indicators*, or KPIs. KPIs are measurable values that can demonstrate how successful an institution is at meeting business objectives. They can be tied to goals that an institution has set for itself or its repository; for example, an institution may have KPIs relating to increased page views and increased time spent on each page as a way of illustrating that repository use is increasing.

Cloud technologies can be used to derive analytics from a repository. Most notably, popular cloud-based analytics platforms can offer an institution a simple, comprehensive site or dashboard that can contain popular metrics and visualizations that make analytics convenient to find, readily available, and customizable in their layout and detail. Often these analytics can be used to drive business decisions, such as studying the user base of a repository to better understand who is using the repository and why it is accessed. Additionally, these analytics can offer information as to peak usage hours and the method of access, such as preferred web browser. This information can enable an institution to make changes to a cloud architecture, such as adding processing power during expected periods of high usage or optimizing code to better perform on the web browser most commonly used to access the repository.

API

Even with the introduction of advanced tools for conducting research, many users are comfortable using basic search techniques to locate resources in a digital repository. This is not necessarily a bad thing; most needs can be met by using simple search boxes and result lists. But it is now possible to use other means to retrieve data from a repository, means that enable large-scale exports or machine-actionable queries, with data returned in a format that can easily be parsed, prepared, and presented by programming languages. One of these means is known as an application programming interface, or API for short. There are different types of APIs, but a common type is known as a restful API. These function by using a URL that can be accessed like any other URL, but instead of directing a user to a website, structured data is returned to the user (these URLs are known as endpoints). This type of advanced retrieval can allow for dynamically generated site content and integration with other sites that share data in the same way. For a repository, this offers a lot of opportunity. For example, a repository may expose all the items in its collection through an API endpoint. When items are exposed like this, a library website can automatically gather this list of items, format them to match its own branding, and publish them on a web page in real time. Alternately, a repository may consume data through an external API. Consider a page in a repository highlighting the achievements of a notable historical figure. The repository could call on an external API that offers publications attributed to this figure, take the data, and publish it in line with related repository content.

In terms of cloud resources, some services provide tools or a framework with which an experienced developer can create and manage an API. In doing so, data for a repository can be exposed though an API that adheres to common standards, meaning that other institutions or developers can use the API in a standard way and achieve the expected results. These cloud-based APIs may tie into the billing, analytics, and security functionality that many service providers integrate across their platforms. In some cases, repositories offered as SaaS applications provide native API functionality as well, making it simple for institutions to offer data through an API without having to create one.

From a systems administration standpoint, some cloud service providers allow users to interact directly with their services using an API. The resources they provide can be managed completely through the API; that is, resources can be created, modified, and deleted by sending commands to an API endpoint. This type of interface access helps to facilitate managing infrastructure as code, as it allows resource management to be automated within scripts or a custom interface.

Another type of advanced data integration is an SDK, or a software development kit. An SDK is not something a basic user might ever use, but when used by a software developer it can enable new features and customizations for repository software. An SDK is a library of computer code, tightly integrated with one or more programming languages, designed to perform specific functions. In the case of a repository, an SDK may provide access to records, search features, site appearance manipulation, or any number of core functions that could aid in developing extensions and integrations to the software. An SDK may provide all the same functionality that an API provides, although some may offer even more granular access to cloud functionality, as they are created with software development in mind. SDKs also commonly include documentation meant specifically for developers to aid in their work.

Updates

Infrastructure Updates and Upgrades

Due to their strictly virtual nature, cloud resources can be managed in different ways than physical devices. Notably, they can be created and destroyed in a matter of seconds. While this might sound disastrous, this offers flexibility to an infrastructure that is not feasible for most on-prem IT environments; old, outdated, and insecure resources can be rapidly decommissioned and recommissioned in a fully patched and updated state. As part of automated deployment and maintenance workflows, this method of resource management reduces the need for human interaction, maintains uniformity across resources, and clearly defines the exact setup of every system in the infrastructure. While cloud-based systems administration can be crucial for overall infrastructure maintenance, it can also be useful for a discrete project such as a repository; software updates, patches, and the compatibility of any underlying servers or databases can be handled remotely and reliably with potentially little downtime.

Cloud service providers may offer specific tools for managing infrastructure, and third parties offer these services as SaaS subscriptions. Some of these services exist both inside and outside of cloud computing and are designed to handle common system administration functions like upgrading operating systems or managing the software installed on remote workstations. Other tools are tightly integrated into a larger suite of cloud services, providing an efficient, fast, and interconnected means of managing cloud-specific resources. These tools can provide enhanced automation and advanced capabilities such as scheduled maintenance windows, unified logging, integrated documentation, version tracking, and more.

Product Enhancements and Extensibility

In addition to handling security and stability updates, cloud services such as SaaS programs can make extending the functionality of a repository a relatively simple process. Adding new features can be as easy as checking a box on an order form, providing some simple configurations, and pressing Download. These extensions may be free additions, or they may come at a cost, and because they are packaged services, they can receive updates and fixes directly from the provider. This may be an optimal workflow for institutions that wish to use only the most common features with their repository or those that prioritize ease of maintenance over design flexibility. SaaS programs are not the only cloud resource to provide benefits to this software extensibility. Cloud-centric software development tools can simplify the process of adding locally developed add-ons and extensions to a repository project. And because cloud systems can be more easily updated than their physical counterparts, new features can be added without worrying that existing servers are not powerful enough to handle the increased capabilities.

Preservation

Preservation is an important consideration when designing a repository; projects such as these are often the culmination of great amounts of time, effort, and cost. Additionally, these projects may be an important (if not the only) means by which important cultural artifacts are preserved. Being completely virtual in construction, a cloud-based repository may feel decidedly impermanent, as all its content may have never existed anywhere other than the remote network of data centers used by a cloud service provider. Still, there are ways to preserve digital content indefinitely, so that like physical resources, it can be discovered, viewed, and studied for generations to come.

Permalinks and URL Management

Maintaining discoverability is an important aspect of digital preservation. Search engines, browsers, and online bookmarking services can change or disappear, sometimes with little to no notice, leaving users without an easy path to sites and service. It is difficult to know how every user will navigate to a repository or the individual records therein, but steps can be taken to increase the likelihood that the repository content can remain discoverable over time. One way to do this is to use what are known as permalinks. Permalinks are URLs that are intended to last over time; to provide a continuous, direct link to their destinations; and to avoid the problem of linkrot, or a URL that eventually directs to nothing. Permalinks are often designed to be short and easy to remember and type, instead of being the long, dynamically generated strings of letters and numbers characteristic of many websites and repositories. They can make it simple for users to navigate back to records of interest and can provide some reassurance that items can be easily visited again in the future. Some cloud services offer tools to help manage URLs, including permalinks, URL shortening features, redirects, and dynamic domain mapping.

Checksums and Other Preservation Tools

For a repository to be considered a trusted and reliable mechanism for storing and accessing digital objects, there must be a way to be sure that the digital objects stay free of errors and corruption. There are tools that can be used to verify that the digital objects stored in a repository have maintained their data integrity and have persisted without developing errors, an idea known as fixity. According the Digital Preservation Coalition, "Fixity could be applied to images or video inside an audiovisual object, to individual files within a zip, to metadata inside an XML structure, to records in a database, or to objects in an object store," all of which may be present in a repository.[3] A specific type of fixity check that can be made is called a checksum. Checksums can be used to validate a digital file and ensure it has not changed over time. Technically speaking, a checksum is a special number or string of characters derived from a formula. Because all digital objects (everything on computers, in fact) actually exists as computer code, this checksum formula can be run on the object to produce one of these special strings of characters. This string is saved (potentially in the repository) for later comparison. At some point later, either as part of a routine check or if file corruption is suspected, the same checksum function is run on the same digital object, which produces another string of characters. If the two strings are the same, this shows that the file has not changed. If they are different, the file has changed in some way, and the change could be a sign of corruption.

Backups

It goes without saying that backups provide an effective countermeasure against data loss. By copying databases, servers, and other cloud configurations to separate locations, a repository can recover from several failure types and be restored to a functional condition in a short amount of time. Traditional backups have included the simple backup of data, where data is exported from databases and servers on a scheduled or ad hoc basis to be reimported in the recovery process. In a cloud environment, many service providers offer integrated backup features that both capture data automatically and provide a built-in mechanism to recreate resources based on these backups.

If possible, consider backing up content to multiple locations. Some cloud service providers offer backup services to various physical locations while remaining a part of the larger services package. That is to say, within the same cloud software suite, backups can be made in both Washington and Wisconsin, all while still using the same service provider. This can provide a level of convenience while offering some mitigation from failure caused by regional disasters. Using different cloud services for backing up content can further help prevent data loss; in the event of total failure of one of the service providers, backups should still be available on the other service. While perhaps less convenient and more costly, this practice provides an additional level of protection for services where data retention is extremely important.

Thanks to the versatility of cloud resources, it is possible to automate backups of databases and other resources. Backups can be set to a schedule, occurring at a specific date and time. These backups can be given a specific duration as well, causing them to automatically delete after a set amount of time. This can be advantageous because it makes backups available and keeps the storage costs associated with storing backups from growing too large. Another option is tape backups. By saving backups to a tape backup service, physical tapes are created with the backup data. These tapes can outlast any power outage and can be physically moved to a new data center if necessary. Tape backups can be inexpensive and physically durable, although they are not convenient to use and should be considered as a last measure in restoring data.

Notes

1. Kief Morris, *Infrastructure as Code: Dynamic Systems for the Cloud Age*, 2nd ed. (Sebastopol, CA: O'Reilly Media, Inc., 2020).
2. *OED Online*, s.v. "analytics," accessed October 27, 2020.
3. Digital Preservation Coalition, "Fixity and Checksums," in *Digital Preservation Handbook*, 2nd ed. (Glasgow, Scotland: Digital Preservation Coalition, 2015), https://www.dpconline.org/handbook/technical-solutions-and-tools/fixity-and-checksums.

Pitfalls and What Not to Do

Cloud services can be used to build a professional-level product, but a lack of knowledge regarding their proper use can lead to difficulties. Even with proper attention and consideration, problems can arise during development. Here are some potential challenges to prepare for when using cloud resources.

Stick to Resources That Suit the Scope of Your Collection

Cutting-edge cloud services can provide engaging experiences and exciting visuals, but they do not need to be used for every collection. They can be expensive, time-consuming to implement, and require expertise beyond what is found in typical IT teams. Some cloud resources may seem obviously out of scope for most repositories; quantum computing, neural machine learning, satellite communication, and other next-generation computing resources may be intriguing but are perhaps not necessary for meeting the needs of most repository users. When creating a collection, look to include resources that will not overextend the capabilities of the team responsible for its creation. For example, managing video resources requires more work than simply hosting still images. If the effort needed to provide video digitization, playback, and transcription seems large compared to the number of videos intended for inclusion in the repository, consider prioritizing the addition of still images before working on the video. If the collection features a large audio component, focus on delivering a rich set of playback tools before expanding into other areas. An advantage of cloud resources is that they can often be easily added to a project after its creation, making it possible to hold off on adding new features until a later time.

Cloud Service Vulnerabilities

Cloud services have teams of trained professionals maintaining their virtual and physical security. Yet for all the precautions in place, they are not invulnerable to hackers, disasters, or business and political forces. With some cloud service providers, the security of a cloud-hosted service is left to the user, and therefore is only as secure as the capabilities of the project developers and systems administrators. With other providers, security is fully managed, and while this can be a great asset for those who are not knowledgeable in this area of IT, it does require trust that the company providing the service is taking the appropriate measures to prevent attacks to its servers. There are other potential risks as well; a company that is providing any resource crucial to the operation of a repository can financially collapse, experience service-disrupting disasters, or be barred from doing business in a country due to changing laws and political influence. In these cases, off-site backups of repository data, source code, and detailed setup documentation may be the only safeguards against a total loss of the project. However, in most cases, the security solutions offered by cloud service providers should be enough to keep a repository safe from data loss and secure from the majority of malicious attackers.

Manage Costs Carefully

Without precise understanding of billing structures and resource costs, a cloud solution can easily become expensive; with the vast number of options available to choose from, the cost of cloud infrastructure can increase wildly if left unchecked. Data redundancy features, high-powered servers, and resources that scale automatically can all incur costs quickly and uncontrollably. Even an increase of traffic to a repository can increase its operational costs; not only will an institution be billed for the resultant resource utilization, but potentially for the increased inbound and outbound data transfer as well. The simplicity of using cloud services also poses a risk to cost stability, as it can be very easy to provision a high-powered, cutting-edge stack of cloud resources that may not immediately appear on a billing cycle.

A thorough understanding of chosen cloud resources will help when budgeting for a repository project. Knowing the expected behavior and pricing options of the services in use can keep a repository designer from making choices that may cost more than expected after the project is deployed. Knowing how much speed is truly required for a repository can make a large difference financially, as the cost difference between various cloud servers and databases can be significant. It is also possible to put caps or limits on certain resources to keep them from growing too large or using too much bandwidth. Sophisticated users can even use timed events to have resources increase their speed and performance on a predetermined schedule, thereby paying only for the necessary resources at the necessary times.

It may be helpful to implement some measure to help prevent unnecessary expenses. Some providers offer cost alarms for their suite of services—e-mails, text messages, or other notifications can be triggered when a monthly spend surpasses a chosen amount. Setting a threshold somewhat lower than the absolute top of the budget can allow a designer to quickly make changes before costs get out of hand while maintaining continuous uptime. These alarms can also provide insight to an institution for long-term budgeting and can provide extra information regarding resource utilization when working on the next version of the repository or when designing other projects.

Conclusion

The creation of a digital repository can be a complex, ambitious undertaking. Ideally, a repository must thoughtfully and accurately display the depth and variety of resources that it contains, and it must possess flexibility to accommodate new resources and collections as they are added. It must be technologically capable of providing audio and video representations of cultural artifacts, academic scholarship, and documents of historical relevance. It must strive to meet the standards of academia, government entities, and other regulatory organizations; be responsive to the needs of its users; and be accessible to all people regardless of their abilities. The ideal repository must be a searchable, extensible tool, ready to scale and change along with its user base.

Building a repository requires its creators to satisfy a long list of requirements. There can be stakeholders at multiple points in the creation process, from design to deployment, each with unique requirements. Design can be hindered or helped by budgetary constraints, institutional regulations, and staffing resources. The process will vary drastically depending on the available technical knowledge. The project may seem financially unfeasible, or the technology required may seem out of reach for individuals and smaller institutions. Just maintaining a repository will require an ongoing commitment that may appear too large for some to assume. And without a guide or a starting point, the entire process may feel like a goal existing only in the distance, too great to achieve.

Creating a repository can seem daunting, especially when faced with the challenges of social distancing and remote working requirements. Without office collaboration, in-person IT support, and access to traditional server rooms, it may seem impossible to even start such a project, much less see it to completion. Fortunately, cloud technology provides the tools to make it possible for any library or other institution to create a repository, built to unique, precise specifications. Cloud technology not only allows for the creation of repositories under these circumstances, it provides the potential to create a repository with capabilities beyond what was possible before. It is a valuable set of resources and should be seriously considered when building a repository or any IT project for a library.

Notes

Library Technology REPORTS

Upcoming Issues	
August/ September 57:6	**Metadata Applications Profiles for Library Data** by Theo Gerontakos and Ben Riesenberg
October 57:7	**Library IT Management in Times of Crisis** by Jason Bengtson
November/ December 57:8	**Gigabit Libraries and Beyond** by Carson Block

ALA TechSource

alatechsource.org

ALA TechSource, a unit of the publishing department of the American Library Association