

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 programming, 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 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 to 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 programming 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.