ISSN 2161-3974

JeSLIB 2021; 10(3): e1205
https://doi.org/10.7191/jeslib.2021.1205

Journal of eScience Librarianship
putting the pieces together: theory and practice

Full-Length Paper

Implementing and Managing a Data Curation Workflow in the Cloud

Fernando Rios and Chun Ly

University of Arizona, Tucson, AZ, USA

Abstract

Objective: To increase data quality and ensure compliance with appropriate policies, many institutional data repositories curate data that is deposited into their systems. Here, we present our experience as an academic library implementing and managing a semi-automated, cloud-based data curation workflow for a recently launched institutional data repository. Based on our experiences we then present management observations intended for data repository managers and technical staff looking to move some or all of their curation services to the cloud.

Methods: We implemented tooling for our curation workflow in a service-oriented manner, making significant use of our data repository platform's application programming interface (API). With an eye towards sustainability, a guiding development philosophy has been to automate processes following industry best practices while avoiding solutions with high resource needs (e.g., maintenance), and minimizing the risk of becoming locked-in to specific tooling.

Correspondence: Fernando Rios: frios@arizona.edu

Received: April 8, 2021 **Accepted**: June 4, 2021 **Published**: August 11, 2021

Disclosures: The authors report no conflict of interest.

Abstract Continued

Results: The initial barrier for implementing a data curation workflow in the cloud was high in comparison to on-premises curation, mainly due to the need to develop in-house cloud expertise. However, compared to the cost for on-premises servers and storage, infrastructure costs have been substantially lower. Furthermore, in our particular case, once the foundation had been established, a cloud approach resulted in increased agility allowing us to quickly automate our workflow as needed.

Conclusions: Workflow automation has put us on a path toward scaling the service and a cloud based-approach has helped with reduced initial costs. However, because cloud-based workflows and automation come with a maintenance overhead, it is important to build tooling that follows software development best practices and can be decoupled from curation workflows to avoid lock-in.

Introduction

In the fall of 2020, the University of Arizona (UA) Libraries launched the University of Arizona Research Data Repository, or ReDATA for short. ReDATA exists in order to fill an institutional gap in long-term data archiving and sharing in support of funder and journal data sharing and archiving mandates. Additionally, it serves as a public archive for data and other materials that do not have an obvious long-term location. Examples of these other materials include datasets purchased by the library intended for use by the UA research community and the winning entries of a library-sponsored data visualization contest (Ly et al. 2020; Oliver et al. 2021).

Although the Libraries have provided research data management services in the form of support for writing data management plans, consulting on data management strategies and providing training on data management topics since 2011, there had not been any resources allocated to supporting data stewardship in the form of a research data repository. Instead, the Libraries provided limited or no support for data archiving via our institutional repository (IR). However, our IR was not designed for accepting datasets or other materials other than manuscripts, theses/dissertations, monographs, reports, etc. This meant that requests from researchers for a place to share their data/code in accordance with data sharing mandates could often not be adequately fulfilled via institutional resources. From 2014 to 2017, UA conducted campus-wide surveys and an in-depth pilot intended to establish data management needs. As a result of that work, the need for an institutional data repository was clearly identified and articulated. In 2019, funding for a research data repository (includes infrastructure and temporary staff) was secured from the university's Office of the Provost for an initial three-year pilot and ReDATA was publicly launched in the fall of 2020. By the time ReDATA came to be, many other institutions had been operating a data repository for several years which meant we were able to leverage their lessons learned to quickly start up.

The ReDATA service follows a model similar to the self-deposit with post-ingest curation approach (Johnston 2017, 144). The service as a whole can be separated into the repository infrastructure, which consists of a Figshare for Institutions instance (Reed 2016; Figshare 2021), and consulting/curation services. Consulting and curation supports depositors in meeting institutional data retention, data security, and confidentiality requirements as well as ReDATA's data quality standards. From the start, our goal for data curation in ReDATA has been to aspire to the FAIR (findable, accessible, interoperable, reusable) principles (Wilkinson et al. 2016) via a workflow that moves datasets towards higher "FAIRness," making them easier to understand and use by the intended community.

In this paper, we describe the curation elements of ReDATA that are implemented in the cloud. While we provide a general overview of the steps in our workflow, our aim is not to detail each and every step (these are relatively standard and we refer the reader to the references cited elsewhere in this paper). Instead, we focus on the steps that have been implemented in the cloud and the decisions that led us there. By "cloud" we mean computing resources hosted outside of the institution's firewall that can be provisioned and accessed conveniently and on-demand (Mell and Grance 2011; Fisher 2018). In the discussion, we place particular emphasis on the technical aspects and the tight integration we have achieved with the Figshare system. Subsequently, we discuss the major technical and management benefits and challenges we encountered in our cloud deployment. Since the work described in this paper is specific to the particularities of our institution, our primary aim is not to present a reusable tool or workflow. Instead, we present some general observations intended for data repository managers who are considering moving some or all of their data curation workflows to the cloud.

Data Curation in ReDATA

Overview

Like many institutional data repositories, ReDATA's goal is principally to provide an institutionally backed service for researchers at the university to meet data and code archiving and sharing requirements from funders and journals. For ReDATA, also in scope is the archiving and sharing of purchased data (where allowed by the license) and most kinds of non-traditional research outputs such as visualizations and conference presentations. In support of data quality, our goal is to avoid ReDATA becoming a "data dump," where researchers treat the system as a general data storage analogous to cloud storage services such as Google Drive. Instead, the service aspires to make data as FAIR as possible while balancing the need to comply with institutional data policies (e.g., security, retention, human subjects, tribal consultation). In support of these goals, we have adopted community best practices for data curation.

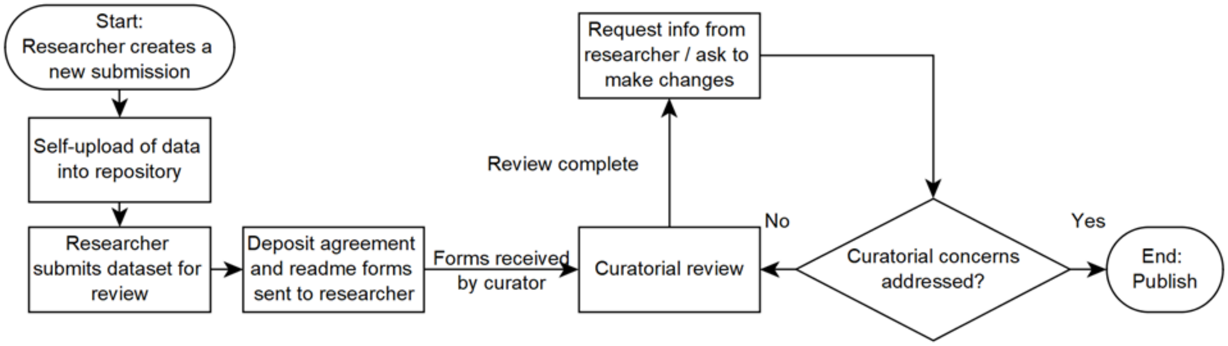
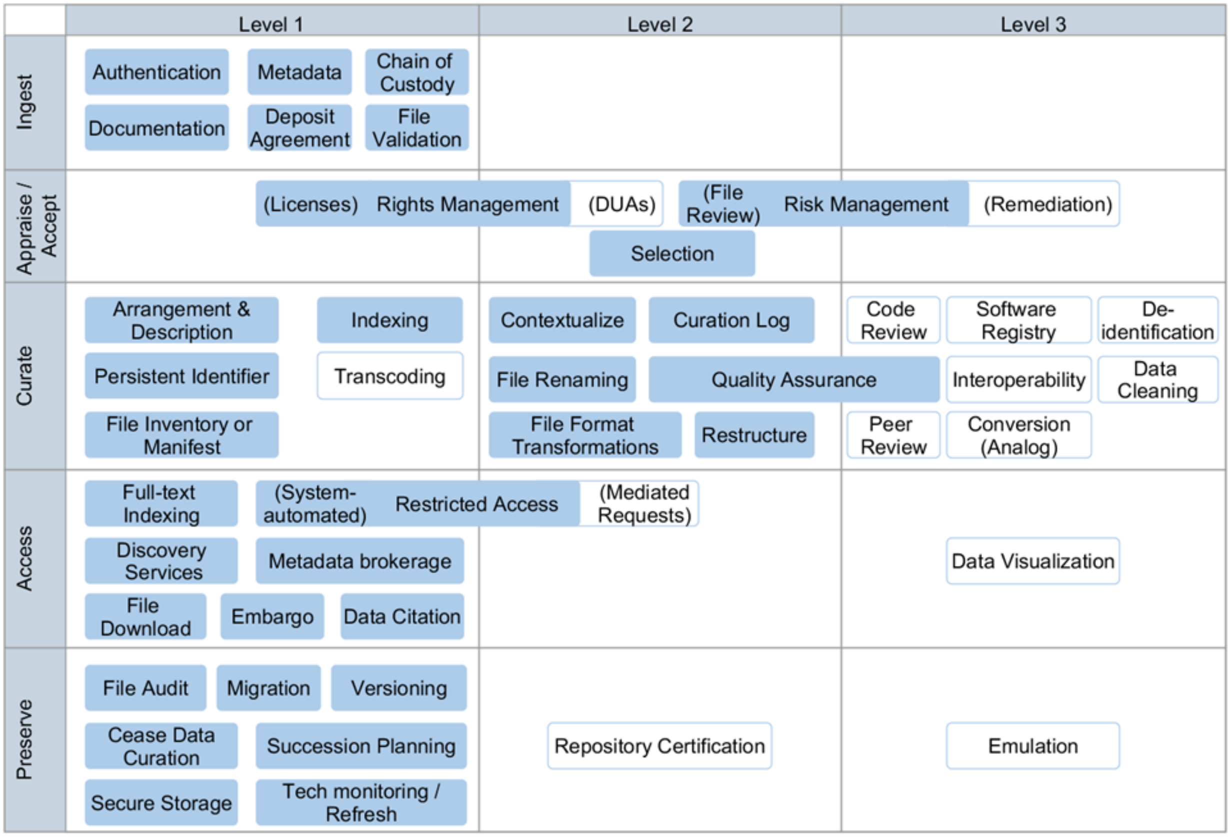

Start:
Researcher creates a new submission
Self-upload of data into repository
Researcher submits dataset for review
Deposit agreement and readme forms sent to researcher
Forms received by curator
Curatorial review
Review complete
Request info from researcher / ask to make changes
Curatorial concerns addressed?
No
Yes
End:
Publish

Figure 1: Overview of ReDATA's curation process.

The curation workflow adopted by ReDATA consists of self-deposit with post-ingest curation (Figure 1; Ly et al. 2021). At a high level, the workflow begins with a researcher depositing materials, (ideally) following a checklist of best practices that we created. After the researcher completes a deposit and submits it for review, we ask them to complete a deposit agreement and to optionally provide additional information about the dataset (used later to augment an auto-generated readme file). After receiving the required information, we perform a curatorial review. Following an iterative revision process with the researcher, the dataset is published.



	Level 1	Level 2	Level 3
Ingest	Authentication Metadata Chain of Custody Documentation Deposit Agreement File Validation		
Appraise / Accept	(Licenses) Rights Management (DUAs)	(File Review) Risk Management (Remediation) Selection	
Curate	Arrangement & Description Indexing Persistent Identifier Transcoding File Inventory or Manifest	Contextualize Curation Log File Renaming Quality Assurance File Format Transformations Restructure	Code Review Software Registry De-identification Interoperability Data Cleaning Peer Review Conversion (Analog)
Access	Full-text Indexing (System-automated) Restricted Access (Mediated Requests) Discovery Services Metadata brokerage File Download Embargo Data Citation		Data Visualization
Preserve	File Audit Migration Versioning Cease Data Curation Succession Planning Secure Storage Tech monitoring / Refresh	Repository Certification	Emulation

Figure 2: Levels of data curation activities, adapted from Lafferty-Hess et al. (2020), CC By 4.0. The target level of data curation services for ReDATA are Level 1 and 2 activities for all categories, except unshaded items which are currently out of scope.

In order to provide common ground when presenting details of our curation implementation, Figure 2, adapted from Lafferty-Hess et al. (2020) and based on the work of the Data Curation Network (Johnston et al. 2016), provides a useful representation of data curation with which to characterize our target service level (see Johnston et al. for the precise definitions of each term). Essentially, we aimed to address most Level 1 and 2 curation activities, with a few exceptions as noted. As of this writing, Level 1 and 2 activities in all categories have largely been implemented, with work remaining principally in the “Preserve” category.

Workflow

In order to achieve the target level of service (most Level 1 and 2 activities from Figure 2), we adopted an approach in which a "primary" curator takes responsibility for a given deposit, handling communications with the depositor, tracking the deposit in our task management system, performing the review, and signing off on the final dataset. A secondary reviewer performs a lighter examination of the dataset and metadata, but does not handle any of the other components of the workflow.

Although this dual reviewer approach can slow down the curation process by about one or two days, so far we have found that the benefits have outweighed the drawbacks. For instance, the secondary reviewer may have domain expertise that the primary reviewer does not. Additionally, we have observed that the secondary reviewer often finds errors or issues missed by the primary reviewer. With savings obtained in other places of the curation process through automation, we believe that this approach can be sustained with at least one experienced curator and one or more less experienced ones. If need be, curation can also be performed by a single reviewer. Single reviewers are often reserved for simple deposits such as conference materials in which curation is usually limited to ensuring proper metadata and correcting typographical errors in abstracts, titles, etc.

We believe that our curation activities provide a sufficient level of dataset reusability when faced with limited staff support (see Management Observations for additional discussion). Curation activities from Figure 2 that we do not implement (e.g., transcoding, code review) were determined to have a smaller return of investment with respect to increasing the data's reuse potential. Although the activities certainly have value, and we have done some of them in certain cases as time allows (e.g., limited code review), performing them for every deposit would limit our ability to scale the service given available staffing.

In order to frame why and how we have implemented much of our back-end data curation process in the cloud, a high-level listing of the curation process is now described (refer to Figure 1 for an overview). The focus of this paper is on how the workflow outlined below is implemented in the cloud rather than on a discussion of how we arrived at the workflow itself. Therefore, we do not discuss the rationale for each step. For such discussions, the reader is referred to other sources that have described curation in more detail (Palumbo et al. 2015; Johnston 2017; Hudson-Vitale et al. 2017; Lafferty-Hess et al. 2020; Gerlach, Färber, and König-Ries 2020). Nevertheless, it bears mentioning that the workflow was informed by work with early adopters who are researchers prior to launching publicly. In the listing below, the labels in parentheses serve as an approximate indication of how each step maps to the categories in Figure 2.

1. Deposit (ingest)
    a) Depositor prepares their submission, ideally following our published guidelines. This includes providing metadata such as title, authors, etc.

A checksum is generated automatically on upload.

b) When ready, the depositor submits their dataset for review.

2. Processing (ingest, appraise/accept)
    a) After the submission is received, a primary curator is assigned.
    b) Preliminary review - briefly examine the deposit and ensure it is in scope.
    c) Confirmation email is sent. If the dataset is in scope, the email contains a partially completed deposit agreement and a form that allows the researcher to submit additional information if they so choose (this is used to populate the readme.txt file in step 3b).

3. Curation (appraise/accept, curate)
    a) Once the deposit agreement and readme form are received, the curator retrieves the deposited files from the repository into a standardized curation directory structure located in a separate staging area.
    b) Deposit agreement and readme are added to directory structure. The readme file is generated based on metadata entered during the submission process and from any other information provided by the researcher in the readme form from step 2c. See the Appendix for more information.
    c) Dataset is curated. This consists of performing steps to address up to Curate Level 2 items in Figure 2. Time permitting, some Level 3 items are addressed: we may selectively perform a deeper review of tabular data (checking for consistent and documented column names, missing values, etc.) and software (generally limited to checking for files referenced which are not present in the deposit, making suggestions for improved reproducibility). If issues are found in the deeper review, we provide recommendations to the researcher.
    d) Metadata and documentation are updated and enhanced where necessary and possible (e.g., linking to grant information, published articles, GitHub repositories)
    e) All changes are logged in a curation review report and the curation process is recorded in our curation tracking system.
    f) A secondary reviewer performs their review, examining the curation log, verifying the changes made, and performing a cursory examination of the dataset and making additional recommendations where needed.
    g) Review results sent to the researcher. If changes or additional information are needed, an alert is set in the curation tracking system and the ReDATA curators await the researcher’s response.

4. Post-curation (curate, access)
    a) Once changes have been addressed by the researcher, the dataset is made public (subject to any embargoes).
    b) Curation activity from the tracking system is saved as a PDF file and added to the preservation copy of the dataset.

This workflow is currently implemented using a combination of Figshare functionality, custom software for authentication, and custom software for curation automation and quality control.

Implementation in the Cloud

Rationale

During the planning phase prior to launching the data repository service, the infrastructure required to support the curation process was determined to require several components: network storage to act as a staging area for curation activities, a server to support backend services such as automation of certain curation processes, and interfacing with the campus enterprise directory service and single sign-on systems for user management. Our initial approach was to work closely with the library's IT department to obtain servers and storage in an arrangement where the data repository team would manage the custom software and the IT department would manage the operating system and hardware. Although we had initially evaluated hosting our infrastructure in the cloud, at that time we had decided that the benefits of in-house hosting of servers was preferable mainly due to the fact that future repository managers would not need to simultaneously be data curators and system administrators.

Early in the rollout of the repository, a shift in strategy by Library IT necessitated moving from locally hosted infrastructure managed by Library IT to cloud services managed by the ReDATA team. We would like to emphasize that this move was driven by our needs to easily deploy, scale, and maintain our software (a need that would be difficult to be met on a short timescale by Library IT), and not due to the fact that Figshare for Institutions is a cloud-based platform. Although the shift meant that we lost some convenience that would have eased the technological management burden, we have realized benefits in other areas. For example, although locally hosted storage would have resulted in faster and easier access from university systems, by moving to the cloud, we realized substantial cost savings, mainly from transforming a large up-front capital expenditure for servers and storage into an operational expense. In fact, under projected usage scenarios, it will likely be many years before the annual costs of cloud services exceed the original budgeted costs for on-premises server and storage. Additionally, because our cloud infrastructure is separate from existing library systems, we are afforded flexibility in deployment that we could not otherwise achieve with on-premises solutions managed by Library IT (e.g., rapid scaling of resources). However, one drawback is that because cloud storage can be more expensive on a per-unit basis, the cost of storage for our data curation needs would likely be more expensive compared to a one-time purchase for an on-premises storage array when large amounts are needed for an extended period of time. In fact, we believe that the cost for storage and associated transfer bandwidth is the largest factor that limits the implementation of certain data curation activities in the cloud (mainly step 3 in our workflow). However, despite the operational differences, the switch from on-premises to purely cloud services did not impact our choice of conceptual data curation workflow described previously.

Although many repositories certainly use cloud-based systems as part of their infrastructure, to our knowledge, implementation details and experiences of how the cloud has impacted data curation have not been well-documented in the literature. Perhaps the only exception is the recent publication by Fallaw et al. (2021) describing the Illinois Data Bank's experience. As noted by Fallaw et al., cloud-based infrastructure can afford a high level of flexibility and scalability for research data repository workflows. The workflow and infrastructure presented in Fallaw et al. differs from ours mainly in the level of complexity. Fallaw et al. adopt a highly complex approach involving many systems and implemented using a variety of advanced services provided by Amazon Web Services (AWS). This means their implementation is capable of advanced functionality in terms of infrastructure management and the ingestion of large datasets. On the other hand, our workflow and implementation is much simpler and in line with our available staffing resources, resulting in a lower barrier to entry. Like Fallaw et al., we also describe how our infrastructure and custom code interface with our specific data repository system and enhance our curation process.

In the remainder of this section, we present our cloud-based infrastructure, highlighting areas where we have leveraged the Figshare API to assist in the implementation of our curation workflow.

Cloud Infrastructure and Services

Instead of a monolithic infrastructure, we have adopted a service-oriented approach in which logical pieces of functionality exist as separate applications. We aim for robustness by increasingly adopting principles from infrastructure-as-code including repeatable processes, disposable, reproducible, and consistent systems, continuous integration/continuous deployment processes, and by conducting end-to-end curation workflow testing (Morris 2016).

Figure 3 shows a high-level overview of ReDATA's components and their interactions with related services. As noted previously, preservation is an area of current work as of this writing. The main curation-related components are: the curation functionality provided by our instance of Figshare for Institutions, a service for authentication (ReQUIAM; Ly and Romero Diaz 2020) into our Figshare instance, a service for automating curatorial review (LD-Cool-P; Ly, Romero Diaz, and Rios 2020), Qualtrics for forms, and cloud compute and storage.

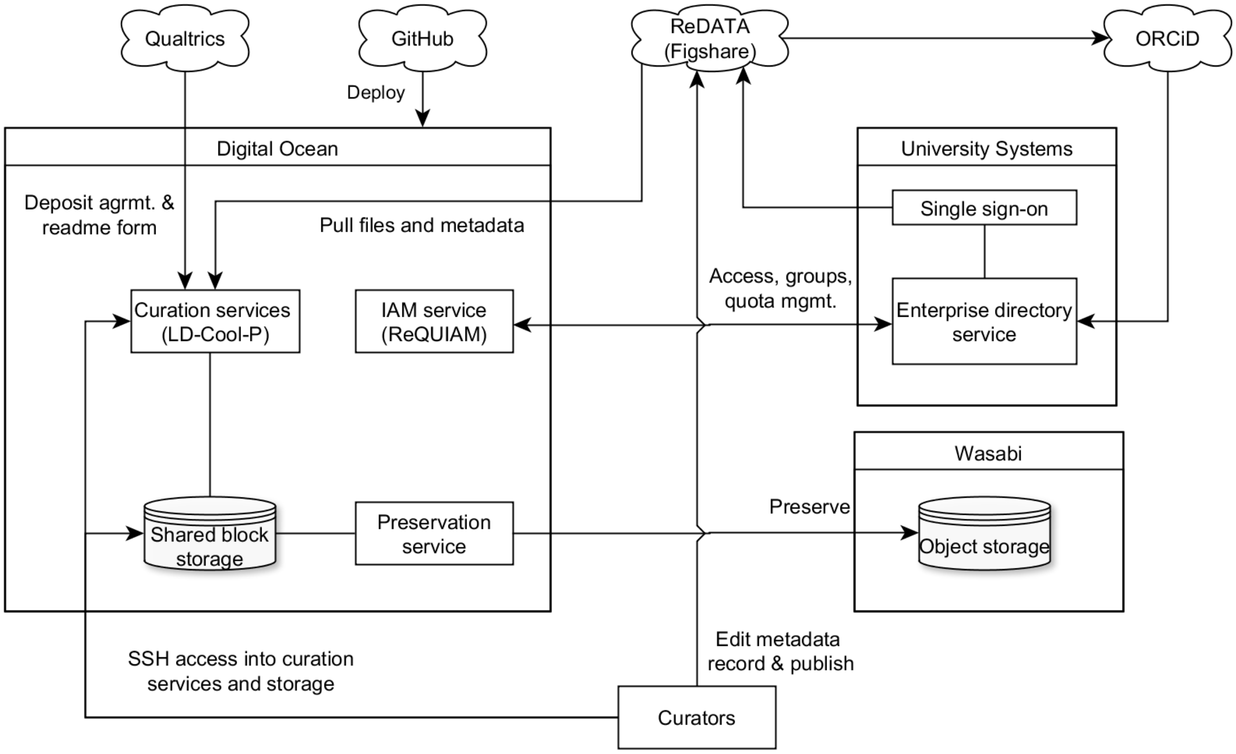


Figure 3: Components of data curation in ReDATA. Curation begins when a dataset is submitted for review in ReDATA. Data, metadata, and other information is then pulled together by curation services into a standard directory structure. A shared curation staging area allows curators to interact with deposited materials directly. After curation is complete, a curator uploads the final data and populates the final metadata into ReDATA and proceeds to publish

For hosting, we selected the cloud vendor DigitalOcean as our primary storage and compute provider and Wasabi as our secondary storage provider. We evaluated other services including AWS; however, due to our small team and relatively basic technical requirements, DigitalOcean and Wasabi allow for much simpler management due to more straightforward functionality and simpler pricing models.

We now describe the curation related parts and how the services provided by LD-Cool-P interact with Figshare and benefit from cloud implementation. Source code for LD-Cool-P is available at github.com/UAL-RE/LD-Cool-P. Since ReQUIAM is indirectly related to curation, its description can be found in the Appendix.

Curation with LD-Cool-P

LD-Cool-P is a pure Python application intended to automate elements of steps 2 and 3 of the curation process outlined previously, as well as general administrative tasks related to the curation process. LD-Cool-P runs on a dedicated virtual private server (VPS) on DigitalOcean. The VPS has attached storage that serves as the staging area for curation activities. The storage is accessible on curators' local machines via a secure shell (SSH) connection and is mountable as a local drive using the SSH file system (SSHFS), allowing curators to work with the data using their preferred operating system and tools. In cases where curation via an SSHFS mount is not possible due to the size of the data or other reasons, we are able to spin up a dedicated curation VPS on-demand. This VPS has a graphical user interface accessible via a remote desktop connection through an SSH tunnel and automatically connects to the shared curation storage on the LD-Cool-P VPS.

LD-Cool-P's functionality currently consists of the following services. Each service is currently manually triggered by a curator at the appropriate time in the curation process.

- Generating custom links for inclusion in the acknowledgement email to depositors (step 2c in the workflow). To eliminate having researchers fill in the same information in multiple places, we provide a link to a partially completed deposit agreement (name, email, title of deposit) where the information has been automatically pulled from ReDATA via the Figshare API. Similarly, we provide a partially completed form where the depositor can optionally add more information about their deposit which will be used to generate a standardized readme file on their behalf (see the Appendix for further details on this process). These forms are connected to a particular deposit using Figshare-provided identifiers which are embedded in the custom links.
- Retrieving files and placing them in a standard directory structure for curation (step 3a in the workflow). Once curation has begun, a curator directs LD-Cool-P to retrieve the data from ReDATA and place it into a standardized directory structure on the curation staging area.
- Generating readme and preparing for curation (step 3b in the workflow). After downloading the data, LD-Cool-P is used to generate a readme and retrieve a fillable template document used to document changes to the dataset and make recommendations for its improvement. The readme is generated by querying the Figshare API to retrieve elements such as the title, description, and authors, combining them with any additional information provided by the depositor in step 2c. See the Appendix for more information.
- Apart from the previously described functionality, LD-Cool-P automates miscellaneous administrative tasks such as: generating a digital object identifier (DOI) for the deposit if the depositor did not already do so at the time of submission, updating the auto-generated readme if the depositor updates their deposit after review, and assisting curators with file management (e.g., ensuring efficient file movement operations as the dataset makes its way through curation and ensuring that proper file permissions are set).

While LD-Cool-P is primarily used for ReDATA, we have developed it with a view towards general use. First, the software is publicly available under an MIT license. In fact, all software that is developed for ReDATA is made available under such a copy-left license. Second, the module that consumes Figshare curation API metadata is made publicly available (pypi.org/project/ldcoolp-figshare) and easy to install with Python package manager (pip). This ensures that other Figshare for Institution instances can utilize it for their data curation workflow.

Management Observations

The guiding philosophy in deploying curation support infrastructure has been balancing three different but related aspects of service sustainability as we see it: resource minimization, efficiency through automation, and avoiding technological lock-in. By resource minimization we mean our ability to provide a functioning data repository service given present and future constraints on resources (e.g., availability of staff to perform curation activities and to maintain the curation support infrastructure, costs related to capital expenditures). We aim to mitigate some of the risk associated with resource constraints by increased efficiency through automation. By efficiency through automation we mean being able to perform curation-related activities more quickly, repeatably, and with fewer errors. However, a highly automated workflow comes with a risk of becoming locked-in to particular tools and processes, reducing agility. In other words, if curation workflows and automation software become inextricably linked, it may become difficult to add, remove, or replace parts of the workflow without major service disruptions. Our philosophy is quite distinct from the approach by Fallaw et al. (2021) that extensively and deeply incorporated specific technologies like AWS as part of their workflow. While such an approach may be appropriate for a mature, well-resourced service like the one in Fallaw et al., in our view, it presents substantial lock-in and maintenance risk to smaller, newer services which may be more sensitive to changes in resourcing and staffing. We now proceed to discuss our experience in attempting to balance resource minimization, efficiency through automation, and avoiding technological lock-in.

As a long-term repository for university data, ReDATA aims to provide a service which promotes FAIRness while preserving the ability to function at a minimal level in times of staff turnover or reductions in staffing. Currently, the service is staffed as shown in Table 1. The goal is for the service to be able to function with one full-time employee at minimum. Therefore, it is desirable from a service sustainability standpoint, to aim for a workflow that allows for as much efficiency as possible via automation, while also acknowledging that, potentially, a single individual would need to maintain said automation in addition to managing the repository itself.

This staffing goal rests on the assumption that additional support is available up-front to implement and document the automation and cloud infrastructure resources so that they can be maintained by others. In our case, we are able to count on such support on a time-limited basis via the Data Workflows & Systems Specialist role.

Table 1: Staffing for ReDATA includes student labor (a mix of graduate assistants and student workers) which helps fill in certain gaps.

Position	**Responsibilities**	**Time dedicated to data repository**
Data Management Specialist (Permanent, 1.0 FTE)	Project lead, outreach, data consultation, communication with vendors, curation	25-50%
Data Workflows & Systems Specialist (Temporary, 1.0 FTE)	Software development, infrastructure maintenance, communication with vendors, outreach, curation	100%
Student assistants (1-2 students, 0.25-0.5 FTE each)	Assist Specialists with assigned tasks	100%

While additional efficiencies could be achieved by automating data-level data curation (step 3c), we have yet to explore what areas would be the most impactful to address, if any. A notable obstacle in increasing automation in this area is the fact that ReDATA is a generalist repository, meaning we can set few expectations about the kind and format of deposited materials. Even for spreadsheets, arguably one of the data formats most amenable to automated curation, the value of developing tooling when there are many exceptions and edge cases is not clear. For example, there are several examples in our repository that would have failed any basic automated quality checks despite being quite acceptable for their purpose (e.g., spreadsheets intended to be used as templates or tables intended for presentation and not machine readability). While certain repositories like the Institution for Social and Political Studies Data Archive at Yale automate certain data-level curation activities (Peer and Dull 2020; Institution for Social and Policy Studies 2021), their investment in developing and maintaining tooling is more justifiable because they may largely receive only certain kinds of data, in certain formats, from certain disciplines.

Shifting curation services to the cloud in order to realize reduction in capital expenditures and to leverage automation efficiencies unsurprisingly required a significant up-front resource investment in the form of dedicated but temporary staff for software development (the Data Workflows & Systems Specialist in Table 1). Although it would not have been possible to move to the cloud without software development support, the temporary nature of software development staff is a risk. To mitigate this risk, we have been intentional in following industry best-practices regarding infrastructure deployment, software development patterns, continuous integration, and documentation.

From a management standpoint, decisions on which elements of curation to build tooling around can have significant consequences on workflow management and sustainability. In deciding which elements of our curation process to automate, we have focused on those where adopting specific tooling does not take away our ability to manually intervene. For example we have automated generating a standardized readme file (step 3b in the workflow) but we have implemented it in such a way that we are not bound by the tooling to generate these files. For instance, if a fault in the tooling emerges, we are still able to manually create a standardized readme based on our template. On the other hand, we have not prioritized automating sending the customized submission acknowledgement email (step 2c) since doing so would significantly constrain our ability to step outside of our normal workflow when needed (e.g., cases where non-standard messaging was required or especially complex situations like multi-part dataset collections).

Another aspect of moving curation to the cloud that has management implications is those related to data security and licensing. In regards to restricted or confidential data such as protected health information (PHI), internal financial records, etc., we do not accept such data under our policies. With on-premises data storage, the consequences of any unintended deposits of such information may be lower in comparison to having them stored in third-party cloud storage that has not been certified for such use at the institution (like Digital Ocean). We recognize that despite policies forbidding it, such deposits may still occur. To address this, we worked with university general counsel to identify the risks, reporting channels, and appropriate mitigation measures should such an event were to occur. The primary mitigation measure during the curation process is the strict enforcement of our "no restricted data" policies in steps 2b and 3c of our workflow. Should an incident occur, we will follow established university reporting procedures. In regards to licensed data, issues may arise if license agreements do not specify where the data will be stored. In our case, we have licensed data in our repository and our licensing agreements did not place restrictions on the location where data would be made available.

Perhaps the most significant management impact thus far has been the overhead needed to manage the cloud infrastructure. Despite our intentional choice of "simple" cloud vendors, activities such as managing virtual servers, managing access, handling billing, and budget planning take more than a trivial amount of time. Although this burden was high at the outset, it has decreased over time, both through increased familiarity with the services and by beginning to adopt configuration management tools such as Ansible (Ansible Community 2021) but this burden will remain. Additionally, even with our service-oriented approach to building our curation tools, bugs are inevitably found and often, the work involved in other areas of running a repository means that sometimes, bug fixes become a lower priority. This underscores the importance of striving for efficiency while avoiding unmaintainable tooling and avoiding becoming locked-in to specific tools.
Institutional repositories (and data repositories) sometimes make use of student labor in order to fill gaps in services. As shown in Table 1, ReDATA also makes use of both graduate assistants and student labor. In our experience so far, we have

found that students with a basic technical understanding/exposure to scripting and the Unix shell have been well-suited to navigating our cloud-based curation approach. Although it has sometimes been challenging to find an adequate pool of candidates, we will likely continue to employ students for supporting ReDATA's services. Although training students to perform data curation has thus far been comparatively straightforward, training them to a level where they are able to meaningfully contribute to workflow and infrastructure maintenance has been challenging due to the limited hours students can work.

Conclusions and Future Work

In this paper, we present the data curation workflow at the University of Arizona Research Data Repository (ReDATA), specifically how we have leveraged cloud infrastructure and automation to implement the workflow. We also discuss risk mitigation related to the long-term technical sustainability of the repository service from the point of view of balancing efficiencies gained through automation with the desire for our workflows to remain agile by avoiding becoming locked-in to complex tooling. Although implementing curation services in the cloud requires an upfront investment in terms of additional planning and staff time, we believe doing so has resulted in a net benefit in terms of lower initial infrastructure costs, additional flexibility in scaling and deployment, and increased efficiency gained through automation. With support of our policies on the size of data we accept, we expect cloud costs to remain relatively stable for some time. We also remain mindful that maintaining tooling and infrastructure could be a challenge without dedicated software developer support. Although the point at which the maintenance burden outweighs the benefits of increasing automation remains to be determined, our intentional effort to avoid technological lock-in means we can more easily adapt our workflow to match available staffing (e.g., abandoning a tool which requires more maintenance effort than is available). Working under the assumption of scarce developer resources in the future means that the repository manager(s) must be familiar enough with a service-oriented infrastructure, be able to understand web services development patterns, and have experience with server management in order to be able to maintain the curation infrastructure at its current level. To operate ReDATA in its minimum viable form, we aim to allocate at minimum, one individual to maintain the data repository infrastructure with curation support from liaison librarians and students. However, to leverage the efficiencies provided by our tooling, curators must understand the conceptual model of how the different pieces of curation fit together and be comfortable with interacting with those pieces separately (often via the command line). This has posed a challenge for onboarding student workers and non-technical staff. To retain automation efficiencies but also to address usability aspects, this challenge could be addressed in the future by linking LD-Cool-P with graphical curation workflow management software such as YARD, the Yale Application for Research Data (Peer and Dull 2020) in a unified curator-facing tool. Incorporating curators into a workflow that is currently somewhat technical highlights the tension between striving for cost and automation efficiencies using the cloud and surrendering some efficiency for the sake of more accessible workflows.

As ReDATA matures, there are several areas ripe for exploration. One example is deploying certain curation services in a serverless architecture. Unlike a traditional server-based deployment where resources (e.g., a VPS) are provisioned and remain available whether they are being actively used or not, a serverless architecture abstracts away the need to provision and manage a server directly, instead, opting for an approach where services are triggered, executed, and billed on-demand (Baldini et al. 2017). Although certain components of our workflow may not be currently amenable to a serverless approach (e.g., storage, authentication) the notion of serverless is promising due to the possibility of a reduced burden in managing operating system updates, system security, and application dependencies. However, going serverless may require re-architecting existing tools and extensive use of serverless features may lead to vendor lock-in (Baldini et al. 2017). Another area of exploration is addressing labor as it relates to curation. Our current approach has been to utilize in-house resources. However, organizations such as the Data Curation Network (DCN) aim to address the labor issue through a distributed network of curators (Data Curation Network n.d.). Cloud-based curation workflows make them more amenable to networked curation via the DCN (Fallaw et al. 2021). However, further work is needed to determine how our workflow could be implemented within the DCN framework in practice.

Supplemental Content

Appendix

An online supplement to this article can be found at http://dx.doi.org/10.7191/jeslib.2021.1205 under "Additional Files".

References

Ansible Community. 2021. "Ansible 3 Documentation." March 11, 2021. https://docs.ansible.com/ansible/latest/user_guide/index.html

Baldini, Ioana, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, et al. 2017. "Serverless Computing: Current Trends and Open Problems." In *Research Advances in Cloud Computing*, edited by Sanjay Chaudhary, Gaurav Somani, and Rajkumar Buyya, 1–20. Singapore: Springer. https://doi.org/10.1007/978-981-10-5026-8_1

Fallaw, Colleen, Genevieve Schmitt, Hoa Luong, Jason Colwell, and Jason Strutz. 2021. "Institutional Data Repository Development, a Moving Target." *The Code4Lib Journal* 51(June). https://journal.code4lib.org/articles/15821

Figshare. 2021. "Figshare for Institutions Admin User Guide: Home." Accessed March 1, 2021. https://figshare.libguides.com/figshare-for-institutions-admin-user-guide

Fisher, Cameron. 2018. "Cloud versus On-Premise Computing." American *Journal of Industrial and Business Management* 08(09): 1991–2006. https://doi.org/10.4236/ajibm.2018.89133

Gerlach, Roman, Bettina Färber, and Birgitta König-Ries. 2020. "Criteria for Appraisal and Assessment of Research Data upon Submission to a Data Repository." February. https://doi.org/10.5281/zenodo.3674051

Hudson-Vitale, Cynthia, Heidi Imker, Lisa R. Johnston, Jake Carlson, Wendy Kozlowski, Robert Olendorf, and Claire Stewart. 2017. "SPEC Kit 354: Data Curation (May 2017)." *Association of Research Libraries*. https://doi.org/10.29242/spec.354

Institution for Social and Policy Studies. 2021. "ISPS Data Archive: Approach." Accessed June 30, 2021. https://isps.yale.edu/research/data/approach

Johnston, Lisa R. 2017. "Curating Research Data Volume Two: A Handbook of Current Practice." *Association of College & Research Libraries*. http://conservancy.umn.edu/handle/11299/185335

Johnston, Lisa R., Jake Carlson, Cynthia Hudson-Vitale, Heidi Imker, Wendy Kozlowski, Robert Olendorf, and Claire Stewart. 2016. "Definitions of Data Curation Activities Used by the Data Curation Network." *University of Minnesota Digital Conservancy*. http://conservancy.umn.edu/handle/11299/188638

Lafferty-Hess, Sophia, Julie Rudder, Moira Downey, Susan Ivey, Jennifer Darragh, and Rebekah Kati. 2020. "Conceptualizing Data Curation Activities Within Two Academic Libraries." *Journal of Librarianship and Scholarly Communication* 8(1): eP2347. https://doi.org/10.7710/2162-3309.2347

Ly, Chun, Damian Yukio Romero Diaz. 2020. "ReDATA EDS Query and Update for Identity and Access Management (ReQUIAM)." *GitHub*. https://github.com/UAL-RE/ReQUIAM

Ly, Chun, Damian Yukio Romero Diaz, Fernando Rios. 2020. "Library Data Curation Tool in Python (LD-Cool-P)." *GitHub*. https://github.com/UAL-RE/LD-Cool-P

Ly, Chun, Jeffrey C. Oliver, Kiri Carini, Chris E. Kollen, and Fernando Rios. 2020. "University of Arizona Libraries 2020 Data Visualization Challenge (Version 3)." *University of Arizona Research Data Repository*. https://doi.org/10.25422/azu.data.c.4986770.v3

Ly, Chun, Fernando Rios, and Megan Hardeman. 2021. "Data Curation in the Cloud with the Figshare API: Presented by the University of Arizona (Version 1)." *figshare*. https://doi.org/10.6084/m9.figshare.14730420.v1

Martin-Flatin, J. P. 2014. "Challenges in Cloud Management." *IEEE Cloud Computing* 1(01): 66–70. https://doi.org/10.1109/MCC.2014.4

Mell, Peter, and Timothy Grance. 2011. "The NIST Definition of Cloud Computing." *National Institute of Standards and Technology* Special Publication 800-145. https://doi.org/10.6028/NIST.SP.800-145

Data Curation Network. n.d. "Mission—Data Curation Network." Accessed March 20, 2021. https://datacurationnetwork.org/about/our-mission

Morris, Kief. 2016. *Infrastructure as Code: Managing Servers in the Cloud*. First edition. Beijing: O'Reilly.

Oliver, Jeffrey C., Chun Ly, Kiri Carini, Fernando Rios, Damian Yukio, Romero Diaz, and University of Arizona Libraries. 2021. "University of Arizona Libraries 2021 Data Visualization Challenge (Version 1)." *University of Arizona Research Data Repository*. https://doi.org/10.25422/azu.data.c.5405493.v1

Palumbo, Laura, Ron Jantz, Yu-Hung Lin, Aletia Morgan, Minglu Wang, Krista White, Ryan Womack, Yingting Zhang, and Yini Zhu. 2015. "Preparing to Accept Research Data: Creating Guidelines for Librarians." *Journal of eScience Librarianship* 4(2): e1080. https://doi.org/10.7191/jeslib.2015.1080

Peer, Limor, and Joshua Dull. 2020. "YARD: A Tool for Curating Research Outputs." *Data Science Journal* 19(1): 28. https://doi.org/10.5334/dsj-2020-028

Reed, Robyn B. 2016. "Figshare for Institutions." *Journal of the Medical Library Association : JMLA* 104(4): 376–378. https://doi.org/10.3163/1536-5050.104.4.031

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3(March): 160018. https://doi.org/10.1038/sdata.2016.18