

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 be 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>.