EPJ Web of Conferences **245**, 04011 (2020) https://doi.org/10.1051/epjconf/202024504011
CHEP 2019

The BNLBox Cloud Storage Service

Ofer Rind[1,*], *Hironori* Ito[1,**], *Guangwei* Che[1], *Tim* Chou[1], *Robert* Hancock[1], *Mizuki* Karasawa[1], *Zhenping* Liu[1], *Ognian* Novakov[1], *Tejas* Rao[1], *Yingzi* Wu[1], and *Alexandr* Zaytsev[1]

[1]Brookhaven National Laboratory, Physics Dept., P.O. Box 5000, Upton, NY 11973-5000, USA

Abstract. Large scientific data centers have recently begun providing a number of different types of data storage in order to satisfy the various needs of their users. Users with interactive accounts, for example, might want a POSIX interface for easy access to the data from their interactive machines. Grid computing sites, on the other hand, likely need to provide an X509-based storage protocol, like SRM and GridFTP, since the data management system is built upon them. Meanwhile, an experiment producing large amounts of data typically demands a service that provides archival storage for the safe keeping of their unique data. To access these various types of data, users must use specific sets of commands tailored to their respective storage, making access to their data complex and difficult. BNLBox is an attempt to provide a unified and easy to use storage service for all BNL users, to store their important documents, code and data. It is a cloud storage system with an intuitive web interface for novice users. It provides an automated synchronization feature that enables users to upload data to their cloud storage without manual intervention, freeing them to focus on analysis rather than data management software. It provides a POSIX interface for local interactive users, which simplifies data access from batch jobs as well. At the same time, it also provides users with a straightforward mechanism for archiving large data sets for later processing. The storage space can be used for both code and data within the compute job environment. This paper will describe various aspects of the BNLBox storage service.

1 Introduction

The Brookhaven National Laboratory (BNL) serves a large, multi-disciplinary research community, some of whom are remote users, widely distributed geographically. Over the past few years, there has been an increasing need for a robust and easy-to-use file sync-and-share service that is integrated into the BNL Scientific Data and Computing Center (SDCC)[1]. The service that has now been developed is known as BNLBox[2] and provides the following features:

- Locally-hosted cloud storage with easy access to user data via browser, desktop and mobile clients, including automated synchronization
- Flexible file-sharing methodologies for sharing data with external collaborators

*e-mail: rind@bnl.gov
**e-mail: hito@bnl.gov

- Multiple access points from local user accounts, including from batch system, analysis portal, etc.
- Tape archiving capability for large data sets
- Access to all users with SDCC or BNL Active Directory accounts with straightforward capabilities for Federated ID

The deployment of this new service reflects an integrated effort by many SDCC staff members in supporting back-end disk and tape storage, web and database services, AAI infrastructure, and user interface development. The sections below will detail the components of this system, highlighting some of the particular features deployed at BNL.

2 BNLBox Components

The BNLBox service is built on Nextcloud[3], a free, open-source, Enterprise File Sync and Share (EFSS) solution (currently v17). The production level server and storage hardware are all new as of 2019. An overview of the BNLBox component architecture is shown in figure 1. It has been deployed as a load-balanced, high-availability server pair with:

- Round-robin DNS + keepalived
- Data storage and configuration directory on a shared Lustre filesystem
- Shared Postgres database with Redis cache on a dedicated pair of master-slave servers

The resilient, high-performance Lustre file server is configured with tape backup provided by the Tivoli Storage Manager (TSM). A custom archiving mechanism is provided for the users to offload cold files to a High Performance Storage System (HPSS) managed tape library, where they can be preserved for later access while not counting against the user's Nextcloud quota. User logins are handled by the Keycloak-based SDCC authentication infrastructure[4]. More information about some of these components are given in the sections below.

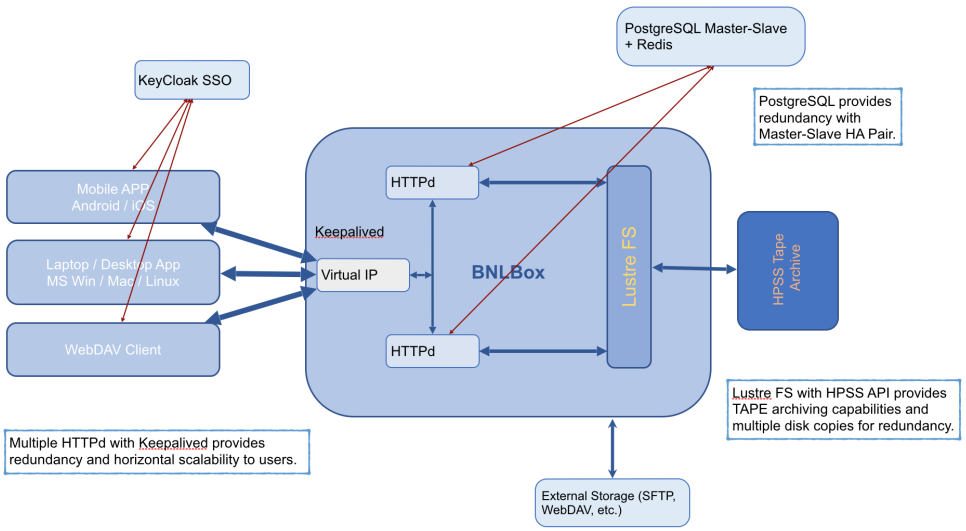

Figure 1. Schematic diagram of the BNLBox component architecture.

3 Site-specific Features of BNLBox

3.1 Authentication and Authorization

From its inception, the goal of BNLBox has been to support all BNL employees, guests and users. Members of these groups, however, may have computer accounts at the SDCC or within the BNL Active Directory domain or both. Therefore, support for two independent and non-exclusive user account management systems had to be built into the authentication mechanism from the outset. This was accomplished by linking all the BNLBox accounts to both SDCC and BNL authentication infrastructures through Keycloak and a custom OpenID Connect (OIDC) configuration within the Nextcloud Social Login app[5]. The login interface is shown in figure 2. Upon first login, accounts are created with a unique Keycloak UUID, ensuring that there are no UID conflicts between the independent user databases. For users with accounts in both databases, the creation of multiple accounts is prevented by enforcing unique associated email addresses. Building upon the existing Keycloak infrastructure automatically brings the capability for future incorporation of multi-factor authentication or integration of federated accounts (e.g. via CILogon), if needed. No primary local Nextcloud user accounts are allowed.

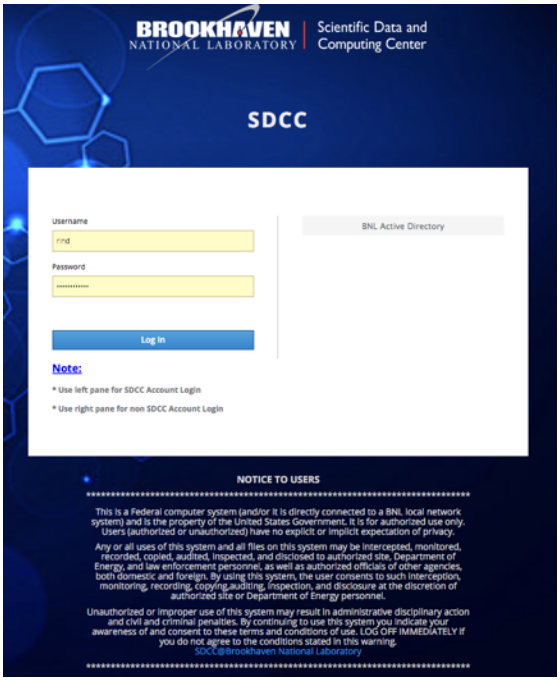

Figure 2. SDCC login page as presented by BNLBox and Keycloak. Default login is via SDCC account; however, the user can choose to log in via BNL Active Directory using the button on the right. Other Federated ID options could be added there in the future as well.

3.2 Tape Archive

Among certain groups of BNL users, there is a need for long-term archival storage of large shared data sets. In order to provide this feature, a mechanism was needed to free up user

disk space with a simple, straightforward option for archived file retrieval in the (perhaps distant) future. The retrieval process should be transparent to the user, except for an additional latency, which is seen as a trade-off for not counting the archived file against the user's disk quota. The desired functionality is provided by an independent Lustre directory that is locally mounted as an external storage folder under every user account. This folder is provisioned and mounted automatically upon user account creation via a Postgres trigger added to the Nextcloud database. File archiving is handled transparently using the Lustre-HPSS copytool agent[6].

3.3 External Storage

Another goal of BNLBox is to provide users with an independent entry point to a shareable filesystem from external sources such as the analysis farm, the batch systems, non-SDCC storage elements, and experiment online storage, among others. The Nextcloud External Storage app enables mounting of these shareable volumes via numerous protocols, such as SFTP, WebDav, etc. Within BNLBox, these mounts can only be created by an admin, upon request, and are linked to the user via a shared key pair. Unlike the primary Nextcloud storage, files on this external storage are owned by the requesting user through that user's account on the external system, with access granted to the associated Nextcloud user. Once mounted, the Nextcloud user may access and share files on the external volume in the same manner as those within the user's primary storage.

4 Experience and Plans

As of the time of this writing, all users have been smoothly and successfully migrated from the former ownCloud/Ceph-based BNLBox service[7] to the new one described herein. The increasing adoption of EFSS services has prompted further scrutiny of related security policies. Aside from disallowing local user accounts, some measures being implemented for BNLBox are: updated computer use agreements and warning banners; enforcing read-only link sharing; updated log analysis and other monitoring; and virus scanning. Federated user access has not been implemented at this time, but that is expected to bring its own set of challenges.

With a planned upgrade to Nextcloud Hub[8] and potential further integrations via APIs being developed within the CS3 community[9], we expect continued expansion of BNLBox services and adoption within the greater BNL community.

References

[1] https://www.sdcc.bnl.gov
[2] https://bnlbox.sdcc.bnl.gov
[3] https://nextcloud.com/
[4] https://indico.cern.ch/event/773049/contributions/3473844
[5] https://github.com/zorn-v/nextcloud-social-login
[6] https://sourceforge.net/projects/lustrehpss
[7] https://indico.cern.ch/event/595396/contributions/2556634/
[8] https://nextcloud.com/hub/
[9] https://github.com/cs3org/