7.3 Final Business Application

IT370 – Advanced Linux | Joseph Keene

By: Ian Tavener | December 13, 2025

Table of Contents

1. Executive Summary

FamilySearch is modernizing its global genealogy platform through the Heritage Vault 2.0 initiative. The system preserves sacred ancestral records and must support millions of users who rely on uninterrupted, secure access. To meet these expectations, the infrastructure requires strong automation, reliable system administration, optimized storage performance, comprehensive security controls, and a resilient network foundation.

Modern Linux capabilities have been applied to address these business needs. Automated backup and maintenance routines ensure consistent operation across all servers. Centralized identity and access management strengthen accountability for administrative actions. Kernel tuning and advanced storage layers provide high performance and data resilience for large archival workloads. Mandatory access controls, encryption, and structured patch workflows enhance system integrity. Network segmentation, host firewalls, bonded interfaces, and secure remote-access methods ensure consistent availability and protect critical services from exposure. Virtualization and containerization extend the system's flexibility by providing isolated execution environments, portable workloads, and optional hybrid-cloud expansion during peak global activity.

Together, these components create a cohesive operating environment designed for long-term reliability, scalability, and stewardship of sacred genealogical data. Heritage Vault 2.0 establishes a robust platform that can meet growing global demand while upholding FamilySearch's mission to preserve and protect ancestral records.

2. Introduction

FamilySearch serves individuals and families worldwide by preserving and sharing genealogical records. These records hold sacred significance for many users and must remain highly available, secure, and reliable. As the organization continues to grow, its Linux infrastructure must evolve to support larger datasets, higher traffic volumes, and stricter modern security expectations.

Heritage Vault 2.0 represents this modernization effort. The solution applies advanced Linux administration practices across automation, identity management, kernel and storage optimization, security hardening, networking, and virtualization. Each component contributes to a unified operational model capable of supporting continuous service for a global audience.

The objective of this paper is to demonstrate how these Linux concepts combine to meet FamilySearch's business requirements. By integrating the work completed across multiple weeks of development, the final solution outlines an environment that is resilient, secure, scalable, and aligned with the organization's long-term mission.

3. Business Requirements Analysis

Heritage Vault 2.0 must satisfy four core business requirements that reflect both the technical demands of a global genealogy platform and the sacred responsibility entrusted to FamilySearch.

High-Volume Access and Scalability
Millions of users depend on fast, predictable searches across large genealogical datasets. The infrastructure must support concurrent access, high I/O workloads, and rapid expansion as new records are added. Linux kernel tuning, optimized filesystems, virtualization technology, and scalable automation workflows directly support this requirement.

Security and Privacy
The platform must safeguard sensitive and sacred data. Unauthorized access, service compromise, or configuration drift pose serious risks. Centralized identity management, SSH key authentication, mandatory access control, encryption, and vulnerability assessments help ensure confidentiality, integrity, and accountability throughout the environment.

Data Storage and Redundancy
Genealogical records represent irreplaceable historical information. Data loss or corruption is unacceptable. Flexible storage management using LVM, redundant RAID structures, and enterprise-grade filesystems protect archival data while supporting ongoing growth.

Performance and Reliability
Users expect the platform to operate continuously. Linux automation ensures consistent updates and backups. Kernel optimization improves system responsiveness under load. Network segmentation, firewall enforcement, and bonded interfaces maintain predictable performance and eliminate single points of failure. Virtual machines and containers further increase reliability by isolating workloads and enabling controlled scaling during peak demand.

4. Technical Architecture Overview

Heritage Vault 2.0 is designed as a layered Linux architecture that combines automation, centralized administration, optimized storage, hardened security, a segmented network

foundation, and a virtualized execution layer. These layers work together to ensure that the system remains reliable, secure, and capable of serving a global audience that depends on fast, uninterrupted access to genealogical records. Each layer introduces a set of tools and technologies that will be detailed in later sections of this paper.

4.1 Automation and Operational Consistency

Automated processes ensure that every server behaves predictably and remains aligned with approved configurations. Bash scripts perform backup, maintenance, and health check operations, while cron schedules recurring tasks at consistent intervals. For large-scale environments, Ansible provides centralized deployment and enforcement of configuration standards across all nodes. These tools collectively reduce manual effort, eliminate configuration drift, and support reliable scaling.

4.2 Identity Management and Administrative Control

A unified access and identity structure governs how administrators interact with the environment. FreeIPA provides centralized user and group management, while SSH key authentication ensures all remote access is secure and tied to verified identities. auditd and system logging services record privileged actions and configuration changes, supporting accountability and compliance. These components maintain a controlled administrative environment where access is traceable, enforceable, and consistent across all servers.

4.3 Kernel Optimization and Scalable Storage Infrastructure

The kernel is tuned using sysctl parameters to support heavy concurrent activity, high I/O throughput, and stable memory behavior during peak usage. For data storage, Logical Volume Management (LVM) enables flexible capacity growth, while RAID-6 provides fault tolerance for archival datasets. Filesystems such as XFS and Btrfs offer performance and reliability suited to large genealogical repositories. This combination ensures that Heritage Vault 2.0 can grow without disruption, maintain stable performance, and protect irreplaceable data from loss.

4.4 Security Hardening and Compliance Controls

Security protections are embedded throughout the system. SELinux enforces mandatory access control to contain compromised processes, while GnuPG encrypts sensitive configuration materials and backup files. The APT patching system maintains timely updates, and OpenSCAP conducts systematic vulnerability assessments to verify compliance with security expectations. Together, these tools establish a hardened and policy-driven security posture appropriate for safeguarding sacred genealogical information.

4.5 Segmented and Redundant Network Architecture

The network design isolates critical traffic using VLAN segmentation, ensuring that management, application, storage, and backup services remain separate and controlled. IPTables enforces strict host-level firewall rules, limiting communication pathways between tiers. Network bonding provides redundancy so that servers remain accessible even if an interface fails. Administrative access is secured using OpenVPN, preventing exposure of SSH or management services to external networks. These components create a resilient communication framework that supports high availability and minimizes the platform's attack surface.

4.6 Virtualization and Performance Tuning

Virtualization and containerization will provide flexible, isolated environments that can scale across on-premise datacenters and AWS. KVM-based virtual machines host stable workloads such as databases and batch processing tasks that require predictable performance. Containers, managed by Kubernetes, run lightweight application services that must be updated or scaled quickly in response to global demand. Performance tuning at the kernel level and centralized monitoring tools help ensure that both virtual machines and containers remain responsive under sustained load. Together, these capabilities support portable workloads, efficient resource utilization, and a clear path for hybrid-cloud growth.

5. Automation of Deployment and Maintenance

The automation layer uses Bash scripting for task logic and a centralized deployment mechanism to ensure every server runs the same backup, maintenance, and health-logging routines.

5.1 Backup Script

Backups are handled by a Bash script that uses rsync to synchronize data to redundant storage. The script ensures that only changed blocks are copied, reducing overhead while keeping archival data current.

```
#!/bin/bash
#hv-sync.sh
SOURCE="/data/genealogy/"
DEST="backup01:/backups/genealogy/"
LOG="/var/log/hv-sync.log"

echo "Starting backup at $(date)" >> "$LOG"
rsync -av "$SOURCE" "$DEST" >> "$LOG" 2>&1
echo "Backup finished at $(date)" >> "$LOG"
```

5.2 Automated Package Maintenance

A second script applies system updates using the native package manager. Updates run on a defined schedule and apply only approved security and stability patches.

```
#!/bin/bash
# hv-maintenance.sh
LOG="/var/log/hv-maintenance.log"

echo "Maintenance started at $(date)" >> "$LOG"
apt-get update >> "$LOG" 2>&1
apt-get -y upgrade >> "$LOG" 2>&1
echo "Maintenance completed at $(date)" >> "$LOG"
```

5.3 System Health Logging

A health-check script records CPU, disk, and memory usage. The log provides a historical record of system conditions and helps diagnose performance issues.

```
#!/bin/bash
# hv-healthcheck.sh
LOG="/var/log/hv-healthcheck.log"
TIME=$(date)

# Check disk usage of /
DISK_USE=$(df -h / | awk 'NR==2 {print $5}')
# Check 1-minute CPU load average
LOAD1=$(awk '{print $1}' /proc/loadavg)

echo "$TIME :: disk_usage=$DISK_USE load1min=$LOAD1" >> "$LOG"
```

5.4 Scheduling with cron

Each script is executed through cron at regular intervals. Scheduling is standardized across all servers to ensure consistent behavior.

```
# Daily sync
0 2 * * * root /opt/hv-sync.sh
# Weekly maintenance
30 3 * * 0 root /opt/hv-maintenance.sh
# Health check every 5 minutes
*/5 * * * * root /opt/hv-healthcheck.sh
```

5.5 Centralized Deployment with Ansible

Ansible distributes the scripts, sets permissions, and installs matching cron jobs on every server. As the Ansible documentation explains, “Ansible playbooks are a simple and powerful automation language that describes the desired state of your systems,” aligning well with the need for consistent configuration across all nodes.

Command used to trigger deployment:

```
ansible-playbook -i inventory hv-deploy.yml
```

Playbook defining the deployment:

```
---
- name: Deploy Heritage Vault automation
  hosts: hv_nodes
  become: yes

  tasks:
    - name: Copy automation scripts
      copy:
        src: "{{ item }}"
        dest: "/opt/{{ item }}"
        owner: root
        mode: '0750'
      loop:
        - hv-sync.sh
        - hv-maintenance.sh
        - hv-healthcheck.sh

    - name: Configure cron schedules
      cron:
        name: "{{ item.name }}"
        user: root
        minute: "{{ item.minute }}"
        hour: "{{ item.hour }}"
        weekday: "{{ item.weekday | default('*') }}"
        job: "/opt/{{ item.job }}"
      loop:
        - { name: "Nightly sync", job: "hv-sync.sh", minute: "0", hour: "2" }
        - { name: "Weekly maintenance", job: "hv-maintenance.sh", minute: "30",
hour: "3", weekday: "0" }
        - { name: "5-min health check", job: "hv-healthcheck.sh", minute: "*/5",
hour: "*" }
```

5.6 Summary

This automation layer ensures that backups, updates, and system checks run consistently across all servers.

- Bash scripts perform backup, maintenance, and health-logging tasks.
- Cron schedules recurring execution.
- Ansible distributes scripts, permissions, and schedules uniformly.

6.0 Efficient System Administration

System administration in Heritage Vault 2.0 is built around centralized identity management, key-based authentication, and unified audit logging. These components maintain consistent access control across all servers and ensure administrative actions are traceable.

6.1 Centralized User and Group Management

All servers are enrolled in a FreeIPA domain. FreeIPA provides LDAP-based user accounts, group membership, sudo policies, and SSH key distribution from a single directory. As the FreeIPA Team (2023b) explains, “FreeIPA is built on top of well-known open source components and standard protocols with a very strong focus on ease of management and automation of installation and configuration tasks.”

User Provisioning Example

A new administrator is added through FreeIPA, and their SSH public key is stored in the directory:

```
ipa user-add jsmith --first=John --last=Smith --shell=/bin/bash
ipa user-mod jsmith --sshpubkey="$(cat jsmith.pub)"
ipa group-add-member admins --users=jsmith
```

User Deprovisioning Example

Access is removed by disabling and later deleting the account:

```
ipa user-disable jsmith
ipa user-del jsmith
```

6.2 SSH Key Authentication

SSH access is restricted to key-based authentication. Servers retrieve authorized keys from LDAP using SSSD. The SSH daemon is configured to disable password logins:

```
PasswordAuthentication no
PubkeyAuthentication yes
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
AuthorizedKeysCommandUser nobody
```

This ensures that only identities registered in the central directory can authenticate.

6.3 Audit Logging

Authentication and administrative activity logs are forwarded to a centralized log server using secured syslog transport over TCP 514. Rsyslog is configured accordingly:

```
*.* @@logserver.familysearch.org:514
```

auditd

auditd captures privileged actions, configuration edits, and security-relevant events:

```
-w /etc/passwd -p wa -k user_accounts
-w /etc/sudoers -p wa -k privilege_change
-w /var/log/secure -p r -k access_log
```

Administrators review activity using standard analysis tools:

```
ausearch -k privilege_change
aureport --summary
```

This provides a consistent record of administrative actions across all systems.

6.4 Summary

This administration layer provides centralized control over identities, access, and activity logging.

- FreeIPA manages users, groups, and SSH keys.
- SSSD and SSH enforce key-based authentication.
- rsyslog and auditd collect system and administrative activity logs.

7. Kernel Optimization and Storage Architecture

Heritage Vault 2.0 uses a tuned Linux kernel, flexible volume management, redundant storage, and enterprise-grade filesystems to support large archival datasets and high read concurrency. This section describes the configuration elements used in the system.

7.1 Kernel Parameter Tuning

Kernel parameters are adjusted through /etc/sysctl.conf to improve I/O throughput, file descriptor handling, and memory behavior during heavy access loads.

```
# Increase available file descriptors
fs.file-max = 2097152

# Improve network queue handling
net.core.somaxconn = 4096
net.core.netdev_max_backlog = 5000

# Reduce memory pressure and improve caching
vm.swappiness = 10
vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
```

Once written, the configuration is applied with sysctl -p. These parameters increase responsiveness during high-volume genealogical queries.

7.2 Minimal Kernel Build

Unneeded kernel modules are removed to reduce overhead and minimize the attack surface. The kernel is compiled with only essential storage drivers, networking components, and cryptographic libraries.

```
# Configure kernel components
make menuconfig

# Compile optimized kernel
make -j$(nproc)
make modules_install
make install
```

A reduced kernel improves consistency, startup time, and security.

7.3 Logical Volume Management (LVM)

LVM provides flexible storage allocation. Volumes can be extended or migrated as new data collections are added, without requiring downtime. Lux (2023) notes that “LVM allows you to create virtual storage containers that can be resized, moved, and managed dynamically without repartitioning,” highlighting its role in supporting continuous storage growth.

Example LVM Setup:

```
# Prepare physical volume
pvcreate /dev/sdb
# Create volume group
vgcreate hv_vg /dev/sdb
# Create logical volume
lvcreate -L 500G -n hv_archive hv_vg
# Format and mount
mkfs.xfs /dev/hv_vg/hv_archive
mount /dev/hv_vg/hv_archive /data/archive
```

Extending Capacity:

```
lvextend -L +250G /dev/hv_vg/hv_archive
xfs_growfs /data/archive
```

LVM ensures that archival capacity can grow in place as digitized records increase.

7.4 RAID-6 Redundancy

RAID-6 provides dual-parity protection, allowing the archival storage layer to remain accessible even if two disks fail during a rebuild.

```
mdadm --create /dev/md0 --level=6 --raid-devices=6 /dev/sdb

#Monitor RAID status or rebuild progress
cat /proc/mdstat
```

This configuration supports resilience for large multi-disk arrays.

7.5 Filesystem Selection

Two filesystems are used depending on workload:

XFS

- Optimized for large files and high-throughput image workloads.
- Provides predictable performance for read-heavy archival operations.

Btrfs

- Copy-on-write filesystem with built-in checksumming, compression, and snapshots.
- Supports rapid cloning and recovery operations.

Example Btrfs Deployment

```
mkfs.btrfs /dev/hv_vg/hv_archive
mount -o compress=zstd /dev/hv_vg/hv_archive /data/archive
```

Snapshot Automation

```
btrfs subvolume snapshot /data/archive /data/archive_snap_01
```

Snapshots preserve consistent data states and can be scheduled through cron.

7.6 Summary

This kernel and storage layer delivers performance, scalability, and data resilience for archival workloads.

- sysctl tuning improves concurrency and I/O handling.
- A minimal kernel reduces overhead and attack surface.
- LVM provides flexible storage expansion.
- RAID-6 ensures dual-parity redundancy.
- XFS and Btrfs support large datasets with strong performance and integrity features.

8. Security Hardening

The security layer applies mandatory access control, encryption, patching, and vulnerability assessment to maintain a consistent and restrictive security posture across all servers.

8.1 Mandatory Access Control (SELinux)

SELinux is configured in enforcing mode to constrain how services interact with system resources. Whittaker (2025) notes that “mandatory policies prevent privilege escalation and unauthorized actions,” reinforcing the role of SELinux in restricting service behavior even if a process is compromised. File contexts are applied to service directories so processes can only access approved paths.

```
setenforce 1
semanage fcontext -a -t httpd_sys_content_t "/opt/hv(/.*)?"
restorecon -Rv /opt/hv
```

This configuration limits the effect of exploited or misbehaving services.

8.2 GnuPG Encryption

GnuPG encrypts configuration artifacts, administrative materials, and backups before storage or transfer. Files are encrypted with the public key of the security team and can only be decrypted by authorized administrators. Encryption prevents unauthorized access even if files are intercepted.

```
gpg --encrypt --recipient "security@familysearch.org" hv-config.tar.gz
gpg --decrypt hv-config.tar.gz.gpg
```

8.3 Patch Management

Security patches and software updates are applied using Ubuntu's APT tools. Updates are installed on a recurring schedule to maintain alignment with approved baselines. Unattended-upgrades can be enabled to automate the installation of future security updates.

```
apt update
apt upgrade -y
apt install --only-upgrade unattended-upgrades
```

8.4 Vulnerability Assessment

OpenSCAP evaluates systems against security benchmarks and identifies configuration drift or policy violations. Scans are run on a schedule and produce detailed compliance reports.

```
oscap xccdf eval --profile standard baseline.xml
```

These assessments provide validation that nodes remain properly hardened.

8.5 Summary

This security layer enforces controlled behavior, protects sensitive data, and maintains alignment with security baselines.

- SELinux confines services to approved file and resource paths.
- GnuPG encrypts sensitive operational data.
- A patch workflow ensures timely updates.
- OpenSCAP scans verify compliance and detect configuration drift.

Together, these components provide a consistent and restrictive security posture.

8.6 Data Stewardship & Ethical Integration

Reviewing this project reveals that while the technical controls are well defined, stewardship is largely implicit. The original design assumes responsible administrative behavior but does not clearly state the ethical boundaries or long-term responsibilities associated with managing sensitive genealogical records.

This refinement introduces stewardship as an explicit design consideration. The system defines a data stewardship framework that distinguishes levels of data sensitivity and clarifies expectations for access, retention, and protection. Genealogical records are treated as entrusted data rather than merely stored assets.

Administrative controls are refined to emphasize accountability over convenience. FreeIPA role assignments, sudo policies, and centralized audit logging are explicitly tied

to least-privilege principles and traceable administrative action. This clarification limits informal expansion of access and defines the ethical scope of elevated privileges within the environment.

The backup design is also refined to include periodic recovery validation. While redundancy and encryption protect stored data, recovery testing confirms that records can be reliably restored without compromise. This addition reflects the principle that stewardship includes recoverability, not only preservation.

Together, these refinements align the technical design with clearly defined stewardship responsibilities. The platform reinforces long-term accountability, restrained access, and deliberate care for data entrusted to FamilySearch, without increasing operational complexity.

9. Networking Architecture

The networking layer uses functional segmentation, host-level firewall enforcement, redundant interfaces, and secure administrative access to maintain predictable and controlled communication between system components.

9.1 VLAN Segmentation

Traffic is separated into management, application, storage, and backup VLANs. Each server defines interfaces corresponding to its assigned roles, preventing unnecessary cross-tier communication. Segmentation maintains clean routing paths and reduces lateral exposure between services.

```
# /etc/netplan/01-vlan.yaml
network:
 version: 2
 vlans:
  vlan10:
   id: 10
   link: eno1
   addresses: [10.20.0.15/24]    # Management
  vlan20:
   id: 20
   link: eno1
   addresses: [10.20.10.15/24]   # Application
```

9.2 Host-Level Firewall Rules (IPTables)

IPTables enforces traffic policy at each node. Rules allow only the required ports within each VLAN, for example, permitting HTTPS for application users and restricting SSH to

the management VLAN. Ellingwood (2022) explains that “Iptables is a standard firewall included in most Linux distributions by default… It works by matching each packet that crosses the networking interface against a set of rules to decide what to do.” These rules reinforce VLAN boundaries at the host layer.

```
# Allow established connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Permit HTTPS from application VLAN
iptables -A INPUT -p tcp -s 10.20.10.0/24 --dport 443 -j ACCEPT

# Permit SSH only from management VLAN
iptables -A INPUT -p tcp -s 10.20.0.0/24 --dport 22 -j ACCEPT

# Drop all other inbound traffic
iptables -A INPUT -j DROP
```

9.3 Network Bonding

Bonded interfaces provide redundancy by combining multiple NICs into a single logical device. Active-backup mode maintains connectivity if one interface or switch path fails.

```
# /etc/netplan/02-bond.yaml
network:
 version: 2
 bonds:
  bond0:
   interfaces: [eno1, eno2]
   parameters:
    mode: active-backup
   addresses: [10.20.0.30/24]
```

This prevents single-interface failures from interrupting service availability.

9.4 Secure Administrative Access (OpenVPN)

Administrators connect to the environment through an OpenVPN tunnel. Each administrator receives an individual certificate, and VPN access is restricted to the management VLAN. This avoids exposing SSH or other administrative services to external networks.

```
port 1194
proto udp
dev tun
server 10.99.0.0 255.255.255.0
client-to-client
cipher AES-256-GCM
auth SHA256
keepalive 10 120
persist-key
persist-tun
```

9.5 Summary

This networking layer enforces clear traffic boundaries and maintains availability during interface failures.

- VLANs separate management, application, storage, and backup traffic.
- IPTables applies host-level control over allowed communication.
- Bonded interfaces provide redundant connectivity.
- OpenVPN delivers encrypted and authenticated administrative access.

Together, these components create a segmented and resilient network foundation.

10. Virtualization and Performance Tuning

The virtualization layer provides isolated execution environments, lightweight service deployment, optional cloud-based capacity, and performance adjustments that improve system responsiveness under varying load conditions.

10.1 Virtual Machines (KVM)

KVM provides hardware-assisted virtualization for workloads that require predictable and stable resource allocation. Red Hat (2024) describes KVM as “an open source virtualization technology for Linux operating systems,” noting that “Linux can function as a hypervisor that runs multiple, isolated virtual machines (VMs).” Virtual machines run with defined CPU, memory, and storage assignments to maintain consistent behavior.

The following example shows a basic VM definition used to provision a virtual machine with fixed resources:

```
<domain type='kvm'>
 <name>hv-metadata01</name>
 <memory unit='MiB'>8192</memory>
 <vcpu>4</vcpu>
 <devices>
  <disk type='file'>
   <source file='/var/lib/libvirt/images/metadata01.qcow2'/>
  </disk>
  <interface type='bridge'>
   <source bridge='br-app'/>
  </interface>
 </devices>
</domain>
```

10.2 Containerized Services (Docker/Podman + Kubernetes)

Containers run lightweight microservices that must scale or update quickly. Kubernetes manages availability, replaces failed containers, and increases replicas during global traffic surges.

The example below demonstrates a simple containerized microservice deployed under Kubernetes:

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: image-service
spec:
 replicas: 3
 template:
  spec:
   containers:
   - name: image-service
    image: registry.local/image-service:1.4
    ports:
    - containerPort: 8080
```

10.3 Cloud Integration with AWS

Additional compute capacity is available by extending the Kubernetes cluster into AWS. Cloud nodes receive container workloads during high-demand periods without requiring changes to deployment patterns.

This example shows how a workload is directed to a cloud-based node during periods of increased demand:

```
apiVersion: v1
kind: Pod
metadata:
 name: hv-burst-job
spec:
 nodeSelector:
  workload: aws
 containers:
 - name: processing-task
  image: registry.local/processing:1.0
```

10.4 Performance and Observability

Elaborating on section 7.1, kernel tuning improves caching behavior, connection handling, and disk activity within virtualized and containerized environments. Prometheus, Grafana, and Loki provide centralized monitoring and logging across on-premise and cloud resources.

The following sysctl entries illustrate performance adjustments commonly applied to virtualized systems:

```
vm.swappiness = 10
vm.dirty_ratio = 15
vm.dirty_background_ratio = 5
net.core.somaxconn = 4096
```

10.5 Summary

This virtualization layer increases flexibility and performance across the environment.

- KVM supplies stable isolation for defined workloads.
- Containers provide lightweight and scalable service deployment.
- AWS nodes offer optional burst capacity.
- Kernel tuning and observability tools improve responsiveness.

Together, these components create a consistent and adaptable execution platform.

11. Business Impact

The completed system design strengthens Heritage Vault 2.0 across the core areas that matter most to FamilySearch's operations. Automation reduces administrative overhead and ensures that routine tasks occur consistently across all servers. Centralized identity management and unified audit logging improve accountability and simplify access control. Kernel tuning, flexible storage management, and redundant disk configurations support predictable performance as archival data grows.

Mandatory access control, encryption, and regular patching reinforce security, while segmented networking and bonded interfaces maintain stable communication paths and reduce the impact of failures. Virtualization and container orchestration add a modest but meaningful degree of flexibility by allowing selected workloads to scale smoothly and operate in isolated environments when global demand increases. Together, these improvements create an infrastructure that is easier to manage, more secure, and capable of supporting sustained global demand, *while clearly defining stewardship responsibilities for the long-term care of entrusted genealogical records*.

12. Final Recommendation and Conclusion

The integrated architecture should be adopted as the long-term model for Heritage Vault 2.0. Its combined use of automation, centralized administration, optimized storage, strong security controls, and structured network design provides a clear, maintainable foundation for future growth. Virtualization and containers supplement these strengths by offering efficient ways to deploy and adjust certain services without altering the core system model. The approach avoids fragmentation, limits configuration drift, and preserves a consistent operating posture across all servers.

This design gives FamilySearch a reliable and scalable platform that aligns with the organization's responsibility to safeguard and provide access to genealogical records worldwide.

13. References

13.1 Automation of Deployment and Maintenance

GeeksforGeeks. (2025, July 28). *'crontab' in Linux with examples*. https://www.geeksforgeeks.org/linux-unix/crontab-in-linux-with-examples/

Ansible. (n.d.). *Ansible playbooks*. Ansible documentation. https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html

DigitalOcean. (2025, October 6). *How to use Rsync to sync local and remote directories*. https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remote-directories

13.2 Efficient System Administration

Ellingwood, J., Walia, A. S., & Baranwal, V. (2025, June 5). *How to use SSH to connect to a remote server (step-by-step guide).* DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-use-ssh-to-connect-to-a-remote-server

FreeIPA Team. (2023a). *Getting started with IPA: Quick start guide*. FreeIPA Project. https://www.freeipa.org/page/Quick_Start_Guide

FreeIPA Team. (2023b). *About FreeIPA.* FreeIPA Project. https://www.freeipa.org/About.html

13.3 Kernel Optimization and Storage Architecture

ENGINYRING Europe SRL. (2025, February 6). *A guide to tuning kernel parameters with sysctl in Linux*. ENGINYRING. https://www.enginyring.com/en/blog/a-guide-to-tuning-kernel-parameters-with-sysctl-in-linux

Lux, A. (2023, November 24). Logical volume management (LVM): Flexible storage administration. AlexFlux. https://alexflux.com/articles/posts/12-Logical-Volume-Management-LVM

National Instruments. (2025, May 30). *Understanding RAID levels, configurations & more*. NI. https://www.ni.com/en/shop/understanding-raid.html

Nicholson, D. (2025, August 19). BTRFS vs ZFS: Key Differences for Optimal Storage Solutions. Nfina. https://nfina.com/btrf-vs-zfs/

13.4 Security Hardening

Red Hat. (2025). *Vulnerability assessment*. OpenSCAP Project. https://www.open-scap.org/features/vulnerability-assessment/

WafaTech. (2025, March 20). *Securing your data: How to use GPG for encrypting Linux server backups*. WafaTech Blog. https://wafatech.sa/blog/linux/linux-security/securing-your-data-how-to-use-gpg-for-encrypting-linux-server-backups/

Whittaker, G. (2025, May 22). *Fortifying Debian with SELinux by enforcing mandatory access control for ultimate system security*. Linux Journal. https://www.linuxjournal.com/content/fortifying-debian-selinux-enforcing-mandatory-access-control-ultimate-system-security

13.5 Networking Architecture

Ellingwood, J. (2022, December 1). *How the Iptables firewall works*. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-the-iptables-firewall-works

Palo Alto Networks. (2025). *What is OpenVPN? | OpenVPN defined and explained*. Cyberpedia. https://www.paloaltonetworks.com/cyberpedia/what-is-openvpn

sk. (2023, November 8). *Configuring network bonding in Linux for high availability: A step-by-step guide to setting up network bonding in Linux using netplan and nmcli.* OSTechNix. https://ostechnix.com/linux-network-bonding-configuration/

13.6 Virtualization and Performance Tuning

Amazon Web Services. (2025). *Amazon EKS Hybrid Nodes overview*. https://docs.aws.amazon.com/eks/latest/userguide/hybrid-nodes-overview.html

Kubernetes Authors. (2025). *Overview*. Kubernetes Documentation. https://kubernetes.io/docs/concepts/overview/

Prometheus Authors. (2025). *What is Prometheus?* *https://prometheus.io/docs/introduction/overview/*

Red Hat. (2024, November 1). *What is KVM?* *https://www.redhat.com/en/topics/virtualization/what-is-KVM*