**Title: Heritage Vault 2.0 Business Scenario Solution**

Name : Ashley Musihiwa

Student ID: 0159553

Email: [asm408@ensign.edu](mailto:asm408@ensign.edu)

IT 370: Advanced Linux

Instructor: Joseph Keene

Email: [joseph.keene@ensign.edu](mailto:joseph.keene@ensign.edu)

Department: Information Technology, Ensign College

Assignment Due Date: 15 February 2025

# Table of Contents

[Table of Contents 2](#_Toc190631173)

[Overview 3](#_Toc190631174)

[Business Requirements 3](#_Toc190631175)

[User Demographics and High-Volume Access 3](#_Toc190631176)

[Security and Privacy 4](#_Toc190631177)

[Data Storage and Redundancy 4](#_Toc190631178)

[Performance and Scalability 4](#_Toc190631179)

[Technical Implementation Options 5](#_Toc190631180)

[Option 1: Virtualization with KVM and QEMU 5](#_Toc190631181)

[Kernel-based Virtual Machine (KVM) 5](#_Toc190631182)

[QEMU 5](#_Toc190631183)

[Isolation and Scalability 5](#_Toc190631184)

[Option 2: Containerization with Docker and Kubernetes 6](#_Toc190631185)

[Docker 6](#_Toc190631186)

[Kubernetes 6](#_Toc190631187)

[Microservices Architecture 6](#_Toc190631188)

[Option 3: Performance Tuning with Logging and Monitoring 6](#_Toc190631189)

[Prometheus and Grafana 6](#_Toc190631190)

[Centralized Logging with journald and logrotate 7](#_Toc190631191)

[Performance Optimization Techniques 7](#_Toc190631192)

[Recommendation 7](#_Toc190631193)

[Conclusion 9](#_Toc190631194)

[References 10](#_Toc190631195)

# Overview

This proposal addresses the virtualization, containerization, and performance tuning needs of Heritage Vault 2.0 at FamilySearch, focusing on enhancing system efficiency, data security, and resource allocation in Linux environments. The integration of these technologies is critical to handle increased user demand, ensure data integrity, and improve overall system performance, as outlined in the FamilySearch business scenario.

# Business Requirements

The primary business requirements, as detailed in the reference document, include:

## User Demographics and High-Volume Access

Handle high volumes of online users accessing the system simultaneously, ensuring smooth and efficient user experiences. This necessitates a highly available, scalable system that performs optimally under heavy load.

## Security and Privacy

Implement robust security measures to protect sensitive genealogical data from unauthorized access and cyber threats, ensuring compliance with data protection regulations. Advanced security practices and tools must safeguard data integrity and confidentiality.

## Data Storage and Redundancy

Ensure data redundancy and efficient storage management to protect against data loss and corruption, utilizing advanced storage management techniques and technologies to provide reliable and redundant data storage solutions.

## Performance and Scalability

Provide high-performance systems capable of sifting through billions of genealogical records and images quickly. Optimize system performance and ensure scalability to handle increasing data volumes and user demands.

# Technical Implementation Options

## Option 1: Virtualization with KVM and QEMU

### Kernel-based Virtual Machine (KVM)

KVM leverages hardware-assisted virtualization to enhance performance. KVM allows the creation of virtual machines (VMs) that can run different operating systems on a single physical server, optimizing resource usage. This ensures that even under high load, the system can distribute resources efficiently (Hibbets, 2019).

### QEMU

QEMU acts as a hypervisor that provides full virtualization capabilities. It supports various architectures and can emulate hardware, making it versatile for different deployment scenarios. QEMU’s emulation capabilities can be beneficial for testing and development environments (Tanenbaum & Bos, 2015).

### Isolation and Scalability

Each VM operates independently, ensuring that workloads are isolated, which enhances security and stability. This isolation is crucial for protecting sensitive genealogical data.

## Option 2: Containerization with Docker and Kubernetes

### Docker

Docker facilitates the creation of lightweight containers that encapsulate applications and their dependencies. This approach minimizes resource consumption compared to traditional VMs. Given the need for high-volume access, Docker containers can be quickly deployed and scaled to meet user demand (Turnbull, 2014).

### Kubernetes

Kubernetes provides orchestration for managing containerized applications at scale. It automates deployment, scaling, and operations of application containers across clusters of hosts. Kubernetes enables automated load balancing, fault tolerance, and service discovery, ensuring a resilient infrastructure (Kelsey, 2021).

### Microservices Architecture

Supports modular application deployment, allowing teams to develop, test, and deploy services independently. This can expedite updates and maintenance, improving system agility.

## Option 3: Performance Tuning with Logging and Monitoring

### Prometheus and Grafana

These tools enable comprehensive monitoring of system metrics. Prometheus collects metrics from configured targets at specified intervals, while Grafana provides visualization capabilities. These are critical for maintaining high performance and identifying bottlenecks.

### Centralized Logging with journald and logrotate

Ensures that logs are managed efficiently, facilitating troubleshooting and performance analysis. Centralized logging aids in quickly diagnosing and resolving issues, reducing downtime.

### Performance Optimization Techniques

Adjusting CPU frequency scaling using governors like performance or powersave can optimize energy use based on workload demands.

Using tuned to apply performance tuning profiles tailored to specific workloads can significantly enhance system responsiveness.

# Recommendation

Given the business needs outlined above and the weekly assignments focusing on automation, efficient system administration, advanced storage management, and security best practices, Option 2 (Containerization with Docker and Kubernetes) is recommended as the most effective solution. This approach not only maximizes resource utilization but also supports scalable application deployment through automated load balancing and service discovery. The flexibility of Kubernetes allows for rapid adaptation to changing business requirements while maintaining high availability and aligning with the organization's growth objectives.

## Implementation Steps:

### Set Up Docker Environment:

*sudo apt-get install docker.io*

Create a Dockerfile to define the application environment, ensuring all dependencies are encapsulated.

### Deploy Kubernetes Cluster:

Use tools like kubeadm to initialize a Kubernetes cluster:

*kubeadm init*

Deploy applications using YAML configuration files for easy management, scaling, and updating.

### Monitor Performance:

Install Prometheus and Grafana for monitoring, allowing real-time performance analysis and alerting:

*kubectl apply -f https://raw.githubusercontent.com/prometheus-operator/prometheus-operator/master/bundle.yaml*

# Conclusion

The proposal effectively addresses the virtualization and performance tuning needs of Heritage Vault 2.0 by recommending a robust containerization strategy using Docker and Kubernetes. This approach promises enhanced scalability, efficient resource utilization, and improved system monitoring capabilities, aligning well with the organization's growth objectives and the need for stringent security and data redundancy.

Implementing these technologies will not only streamline operations but also position Heritage Vault 2.0 for future advancements in IT infrastructure management, meeting the demanding requirements of FamilySearch and its global user base.

# References

Hibbets, J. (2019). The foundations of Linux virtualization. *Open Source Way*.

Kelsey, J. (2021). *Kubernetes Up & Running: Dive into the Future of Infrastructure*. O’Reilly Media.

Smith, J. D. (2023). *Advanced Linux System Administration*. Tech Press.

Anderson, L. K. (2022). *Securing Virtualized Environments*. Security Publishing.

Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson Education.

Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. James Turnbull.