



CS labs Infrastructure Hardware Requirements Specification for CS labs Operations and Application

Group Leader - Zac Clifton,

Group Member - Zach Bouvier, Group Member - Mamadou Thiello

October 14, 2019

Abstract

The CS Infrastructure team covers every aspect of operations for this project. We cover the setup, maintenance, and upgrades of all the servers, services, and communication in the domain.

Contents

1	Contribution Breakdown	1
2	Introduction	1
2.1	Purpose	1
2.2	Definitions	1
2.3	Overview	1
2.4	References	1
3	scope	2
3.1	Functionality	2
3.2	Usability	2
3.3	Reliability	2
3.4	Performance	2
3.5	Support-ability	3
3.6	Design Constraints	3
3.7	Online User Documentation and Help System	3
3.8	Interfaces	3
3.8.1	User Interfaces	4
3.8.2	Hardware Interfaces	4
3.8.3	Communications Interfaces	4
3.8.4	Applicable Standards	4

1 Contribution Breakdown

Zac Clifton – Project leader, contributes by creating reports, deciding infrastructure architecture and project decisions.

Zach Bouvier – IT Operations, contributes by implementing infrastructure automation and monitoring.

Mamadou Thiello – IT Operations, contributes by implementing infrastructure automation and integration.

2 Introduction

2.1 Purpose

The system for CS labs requires authentication, flexibility security, and monitoring to work.

2.2 Definitions

2.3 Overview

The hardware must be able to contain multiple virtual environments in addition to managing the virtual internal environment. This will culminate at in a system that will allow users to choose/create labs to train on. We expect our users to be familiar with a web browser and their technical skill range from a beginner to a advanced. We expect the content being mainly created by the community, which includes local professors. We must be able to support an up time of at least 95% in a year due to grades of the lab being used on it. We are assuming that all dependencies will be open source and will be maintained and supported for years to come.

2.4 References

<https://github.com/ius-csg/cslabs-backend> <https://github.com/ius-csg/cslabs-webapp> <https://github.com/ius-csg/cslabs-web> <https://reactjs.org/> <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2> <https://www.proxmox.com/en/>

3 scope

This report will cover the needs of the system/hardware to support the CS Labs application.

3.1 Functionality

The functionality needed for this is:

1. Monitoring
2. Cloud operations
3. Security
4. Networking

3.2 Usability

With this product, the user is learning at there own pace. Other then understanding the layout of the website the usability will be very high.

3.3 Reliability

This system should have

1. Availability of 90
2. A Mean Time Between Failures of 3 weeks
3. A Maximum Bug count of 10 per 10,000 lines of code.
4. A Defect rate 1 in 10000

3.4 Performance

The system should be able to handle:

1. 20 ms response time
2. 1000 transactions per second

3. Capacity of 30 users
4. a degraded mode of 40ms response, 50 transactions per minute, and 5 users
5. A the full usage of any hardware (RAM, disk, throughput) should not be more then 80

3.5 Support-ability

This system once built should be able to run maintenance, and the majority of the errors correct by itself. The only things that need human over site our system updates and upgrades.

3.6 Design Constraints

3.7 Online User Documentation and Help System

Our system, will have an online documentation on how to run the application and interface with it.

3.8 Interfaces

The different interfaces that will be supported to work with this application are:

- The FreeIPA Kerberos server will be used for authentication providing a single password for all CSG applications.
- Proxmox will be used to manage VMs.
- A Logstash server will be provided by CSG to provide analytics on errors and performance of the application.
- Rundeck will server as an automation repository for managing VMs. This will help make communicating with proxmox a lot easier.
- The backend will utilize C# for a strong typed experience.
- The backend will utilize .Net Core to run C# on linux

- The Database (Maria DB) will be used to store the application's entities.
- Typescript and React will serve as the front end software interface to create beautiful UIs.

3.8.1 User Interfaces

The only user interface will be a website on port 443 that is server from a web server in a DMZ.

3.8.2 Hardware Interfaces

The specific documentation for this section is being kept internally and will only be disclosed to authorized users. The hardware interfaces will be terminals, web guis, and cluster management interfaces.

3.8.3 Communications Interfaces

The specific documentation for this section is being kept internally and will only be disclosed to authorized users. We will be using common http/https ports to manage and interact with our systems.

3.8.4 Applicable Standards

This application will need to be supported by a well designed environment that allows us to scale and mange easily.