Project Cloud  
Cycle 2 Report

By

Jackson Gray

Carnell Greenfield

Jaden Jefferson

Nick Miller

Kelly Payne

Submitted in partial fulfillment of the requirements for  
CPSC 4176 Senior Software Engineering Projects  
to TSYS School of Computer Science,   
Columbus State University

March 30, 2025

Sponsored by

Dr. Zhou

Verified by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_on \_\_\_ / \_\_\_ / \_\_\_\_

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Table of Contents

[Table of Contents 1](#_Toc1319351906)

[1 Executive Summary 2](#_Toc2100682240)

[2 Project Introduction 3](#_Toc1531320783)

[2.1 Previous Development 4](#_Toc1596999530)

[2.2 Intent This Cycle 5](#_Toc1712404864)

[2.3 Future Work 5](#_Toc1725396819)

[3 Requirements / User Stories 6](#_Toc868188902)

[3.1 User Epics 7](#_Toc1841532755)

[3.2 User Stories 7](#_Toc1574714151)

[3.2.1 Stories 8](#_Toc179706597)

[4 Design Documentation 8](#_Toc1056512751)

[5 Management Plan 11](#_Toc1065876538)

[5.1 Task Assignments 12](#_Toc2081564552)

[5.2 Development Schedule 12](#_Toc1064005512)

[5.3 Planned Code / Feature Freeze 13](#_Toc303011898)

[6 Risk Mitigation 13](#_Toc2116179694)

[7 Test Plan and Test Procedures 14](#_Toc867837854)

[7.1 Test Plan 15](#_Toc329632625)

[7.2 Test Procedures 15](#_Toc2112515073)

[7.2.1 Procedure 1: Sample Test Procedure 15](#_Toc1784008286)

[8 Lessons Learned 16](#_Toc882668861)

[References 17](#_Toc1072036423)

# Executive Summary

Authors: Kelly Payne, Jaden Jefferson

As part of an initiative to create a more secure, accessible, and faster platform for storing and retrieving data on Columbus State University main campus grounds, we are designing a private cloud server. A private cloud is a way for a business to store and manage its data and applications on a dedicated set of servers, in our case it will be used for Columbus State University’s faculty and students to have their own secure online storage space.

The creation of this server will bring the campus several benefits that online cloud companies such as AWS, Azure and others are unable to deliver. The private cloud will allow more control to determine how data is managed, bring security benefits such as the ability to avoid the potential of data being exposed to a third party, and increase performance allowing faster data upload and download speeds using a local area network.

The project starting on January 15th, 2025, will be ready for production on April 23rd of the same year. Our solution does not necessitate the reinvention of the wheel, but we must be meticulous to ensure that the project lives long after its creation. The official launch and presentation will take place on April 23rd and April 27th, respectively.

# Project Introduction

Author: Kelly Payne, Nick Miller, Carnell Greenfield

Our project is the development of a private cloud for the Columbus State University Computer Science Department. We have been provided with three pre-owned hardware servers with the goal of providing a horizontally scalable and clustered infrastructure. This project is intended to give students and employees access to processing and storage in a secure private cloud. Our goal is to create the cloud from scratch using open-source software to ultimately provide access to university students and faculty.

## Previous Development

The work located below is from Sprint 1 through Sprint 4. Our objective in these phases was to lay a foundation for the project by installing essential software, addressing initial challenges, and growing our understanding of the tasks ahead.

We started by examining the servers we’d been provided with and recording information about their hardware specifications. We have three servers: one Dell PowerEdge R730 and two PowerEdge R710s. The R730 is equipped with an Intel Xeon E5-2660 v3 CPU, 128 GB of DDR4 RAM, and approximately 22 TB of hard disk storage. Each R710 is equipped with an Intel Xeon L5630, 48 GB of DDR3 RAM, and approximately 900 GB of hard disk storage. We used a live Puppy Linux environment to gather hardware information and bypass a password-locked Windows installation.

Next, we installed Ubuntu Server 24.04 on each server, configuring the operating system as necessary. Each machine was connected to the LAN in the process successfully. With Ubuntu Server installed, we continued by installing relevant software packages like Kubernetes. We also encrypted each server through the Linux installation which provides added protection of data on disks so that no one can access the data without the correct decryption key.

As of now, March 24th, we are in Sprint 4. We were able to install Ubuntu on all three servers and KVM. We also reassessed our disk security through RAID for each server was reconfigured with a level of 5. This was due to the low overhead for the partitioning of disks and provided fault tolerance for data held on each drive. SSH was installed on the servers. As we have progressed through the epics we are seeing that there are more work items related to the epics such as DevStack, hardware monitoring and virtual media. Some other work items that we have completed since our last report are finding the COM Port for 2960-XC Switch, VLAN configurations, and getting the switch up and running.

Furthermore, the team engaged in an in-depth analysis of OpenStack. The documentation produced during the last phase included a comprehensive PowerPoint presentation that detailed the initial findings and configurations. These records are instrumental for ensuring that future team members can effectively replicate or enhance the project's progress.

The DevStack installation has caused some difficulties with connecting to the user interface so that is a bug we are currently working on in this cycle and possibly the next.

## 2.2 Intent This Cycle

Author: Kelly Payne, Jaden Jefferson, Jackson Gray

Sprint 1:

This included downloading and configuring software for each of the three servers, configuring the wiring for PowerEdge 710 and 730, obtaining access to the desktop to streamline and speed up deployment, conducting continuous research, and installing the operation system. However, our progress was somewhat delayed due to weather related campus closures, which limited our ability to work on our project, except for research. Despite the setbacks, we successfully installed Ubuntu Server onto each server, configured the iDRAC on each server, completed the wiring configuration for servers.

Sprint 2:

Get a VLAN up and running and access to Cisco switch CLI for further configuration, get the RAID configured, apply a fresh coat of thermal paste to CPU heatsinks within the three servers, and get IP address configured for server network. We were able to configure the RAID to our needs due to the presence of physical RAID controllers within the hardware, though setbacks with the network switch did not allow us to fully create a working LAN.

Sprint 3:

Include comprehensive research on OpenStack and looking into development tools for it, including MicroStack and DevStack. We hope to fix the issues we were having with our network switch and finally create a LAN that we can connect the servers to, allowing for full connection across all three servers as well as simple remote access and configuration.

Sprint 4:

Get DevStack downloaded on the 730 server on a single node to work with OpenStack playground by creating an instance. We also hope to fix the issues associated with DevStack in relation to creating a single instance and getting an error in return. We also hope to get the VLAN configured on our Cisco Switch, find the COM Port for 2960-XC Switch, and get the switch up and running.

We are laying the groundwork for maintenance and future development by documenting our work and wiring configurations, as well as providing manuals for servers and networking equipment, so that the next team will not have such a learning curve.

Sprint 5:

Get Internet Access through wall port onto the switch. Have all three devices connected to the switch and communicating. Install and configure production instances of OpenStack using Kolla-Ansible. Continue pursuit of requested hardware in order to facilitate the listed features.

## 2.3 Future Work

Author: Jackson Gray, Jaden Jefferson, Kelly Payne

Future cycles will involve the installation and configuration of OpenStack as well as the creation of clusters. We will be connecting our servers to the network and enabling remote access through SSH. Our goal in the up-coming cycle is to create multi-nodes for DevStack on the desktop and separate the DevStack from OpenStack. All of our servers will have OpenStack installed and be running OpenStack. We will also be researching and implementing security measures which include firewalls and access control.

Our documentation efforts will continue. In the next couple of cycles, we plan to create a more detailed network architecture diagram which will include all devices and their connections. This will include information on IP addressing, VLANs, subnetting, and routing protocols used. We hope by the end of the semester to have documentation for the best practice guidelines on maintaining the network and system security which would include firewall rules and VPN configurations as well as a list of any known issues or challenges that we encounter while setting up the private cloud to allow people ahead of us to maintain, repair, and configure the system.

# Requirements / User Stories

* User Stories: Server Setup and Configuration: Configure Dell PowerEdge servers for cloud deployment, including iDRAC and RAID setup, and successful Ubuntu Server installation.
* Virtualization Setup: Configure KVM and deploy virtual machines to efficiently run diverse workloads.
* Networking and Integration: Interconnect all servers to function as a single cloud computing unit.
* Private Cloud Environment: Deploy applications on the private cloud to provide a secure environment for testing and experimentation.
* Documentation for Reproducibility: Maintain comprehensive setup and troubleshooting documentation to guide future teams.
* Server Maintenance and Monitoring: System administrators use iDRAC to monitor and maintain server health and performance.
* Virtual Machine Deployment: Cloud engineers create and manage virtual machines using KVM to support cloud workloads.
* User Application Deployment: Research students deploy containerized applications and monitor workloads in the private cloud environment.

## User Epics

Authors: Kelly Payne, Jaden Jefferson

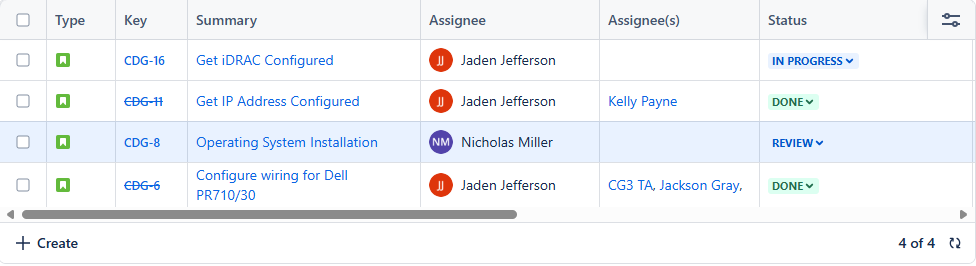


## User Stories

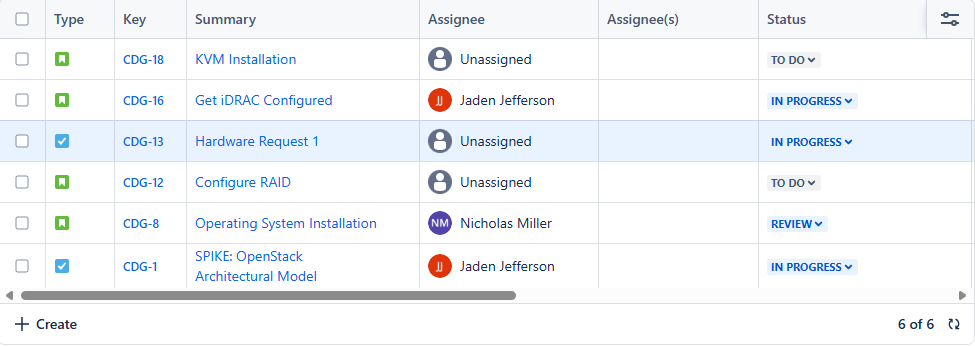
### Stories

Authors: Kelly Payne, Jaden Jefferson

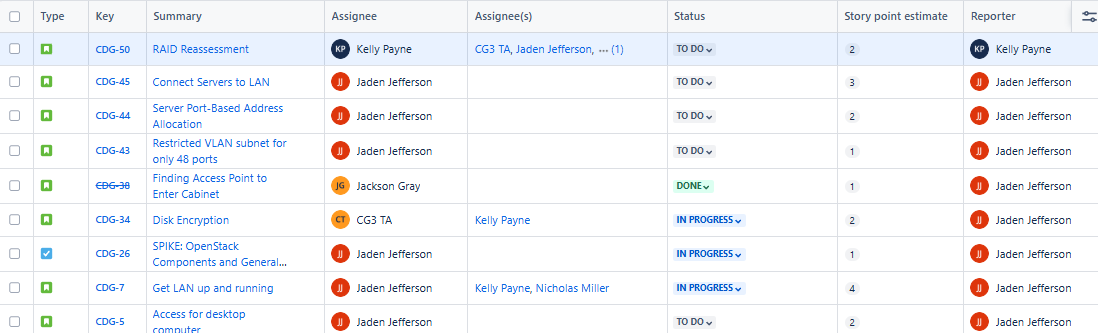
**Sprint 1**

**Note:** CDG-1 SPIKE was moved to current sprint but was also worked on during Sprint 1.

**Sprint 2**



**Sprint 3**



**Sprint 4**

A close-up of a notebook

AI-generated content may be incorrect.

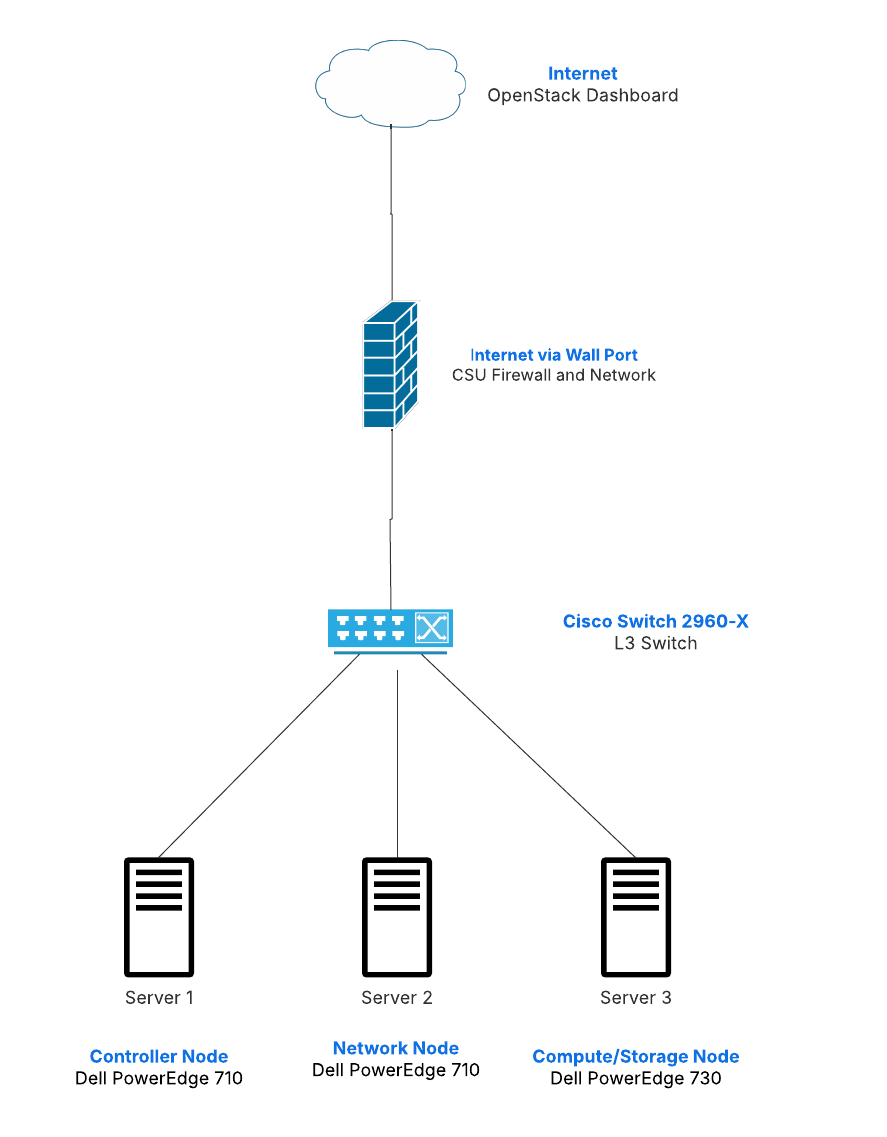
# Design Documentation

Authors: Nick Miller, Kelly Payne

The private cloud will provide compute, networking, and storage services to meet the organization’s requirements. The infrastructure will be built on a set of two PowerEdge 710 servers and one PowerEdge 730 server which will be connected to a Cisco LAN switch. Our plans are to use the desktop for DevStack installation, so we can separate DevStack and OpenStack. The 710A will be the controller, the 710B will be used as the network, and the 730 will be used for compute and storage.

The foundation of our private cloud is three physical servers running Ubuntu Server. OpenStack will be used for the core cloud services. OpenStack is a free-and-open-source collection of cloud computing software, which we will be using to manage computing, storage, and networking resources and divide them among virtual machines. The essential OpenStack services will be Keystone for identity management, Nova for computing, Neutron for networking, Cinder for block storage, Glance for image management, and Horizon for dashboard. Virtualization will be managed using KVM (Kernel-based Virtual Machine), a Linux-native application which divides physical hardware resources into virtual machines. We will use containerization to further divide those virtual machines’ resources into on-demand, scalable resources for users. A cluster will manage the containerized resources.

**Physical Architecture Diagram**



**Openstack Structure Diagram** A screenshot of a computer

AI-generated content may be incorrect.

**Complete OpenStack Architecture Model**A diagram of a computer network

AI-generated content may be incorrect.

# Management Plan

## Task Assignments

Authors: Kelly Payne, Jaden Jefferson

**Workflow**

1. Evaluating and identifying work items and areas for improvement of final product.
2. Addressing these improvements through individual tasks that group back to team-oriented stories.
   1. Connect back to team-oriented stories
3. Carrying out work items of current sprint
   1. User Story movement through Scrum Board
      1. TO DO, IN PROGRESS, REVIEW, DEMO, DONE
   2. Continuous Research and Learning
      1. All engineers
      2. Sharing findings with the group
4. Documenting the process of development and engineering
   1. All documentation is in a folder in Teams accessible by the group
5. End of Cycle hour adjustments and cumulative reporting
6. Repeat 1-5

## Development Schedule

Authors: Kelly Payne, Jaden Jefferson

* Sprint 1 - Jan 15th to Feb 5th

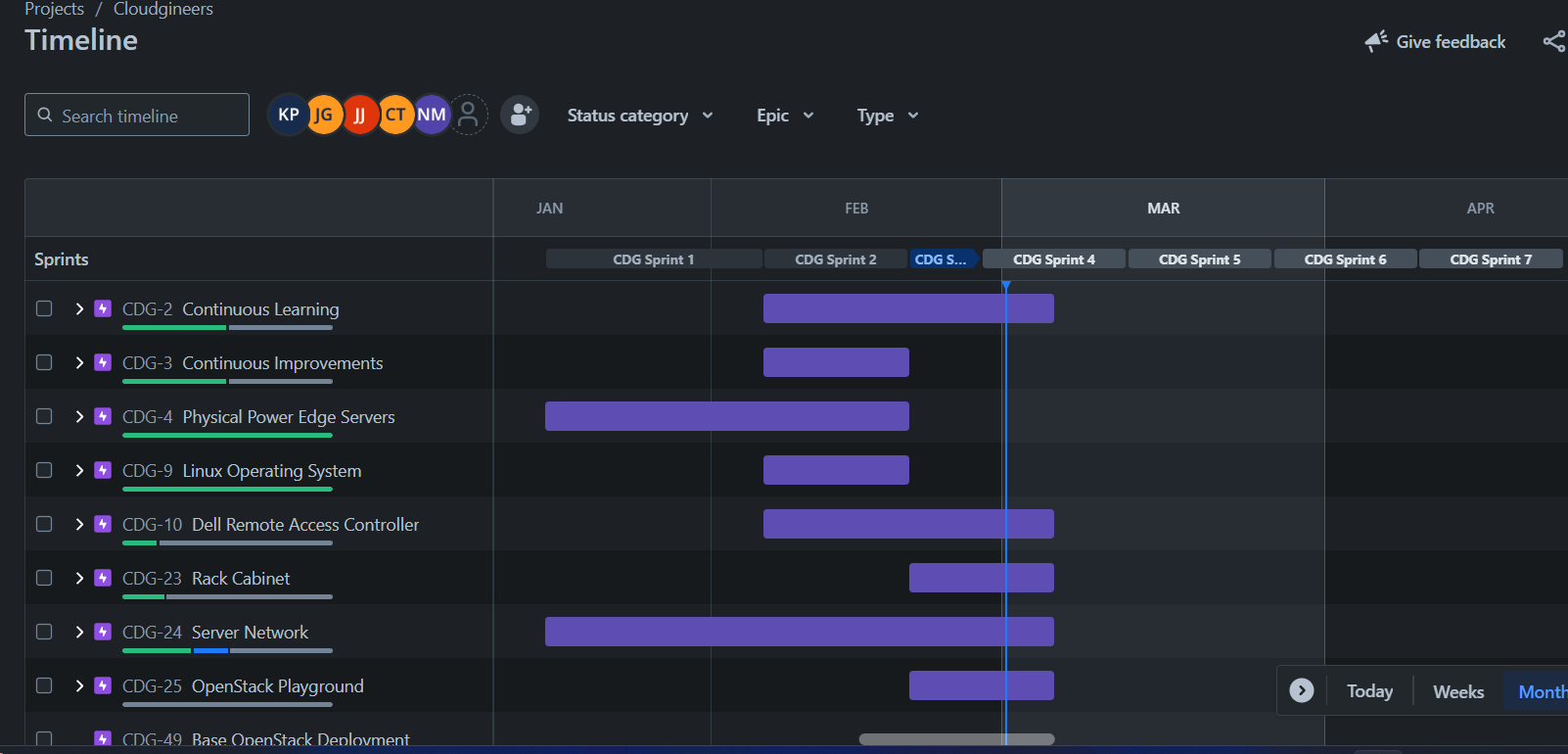
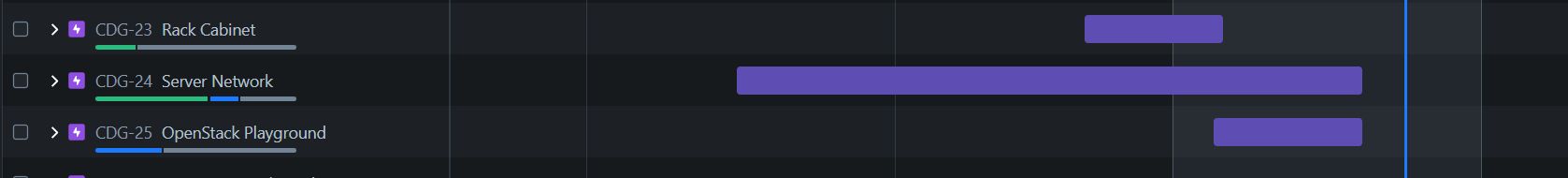
**ü**  Sprint 2 – Feb 6th to Feb 19th

**ü**  **Sprint 3 - Feb 20th to Mar 5th**

**ü**  Sprint 4 - Mar 6th to Mar 19th

* Sprint 5 – Mar 20th to Apr 2th
* Sprint 6 – Apr 3th to Apr 16th
* Sprint 7 – Apr 17th to Apr 23rd

**Current Timeline of Epics**



**Other Key Dates**

**Product Ready for release:** April 23rd

**Team Presentation:** April 27th

## Planned Code / Feature Freeze

Authors: Kelly Payne, Jaden Jefferson

**Previous freezes:** Snow and Ice conditions closed campus.

**Planned freezes:** Spring Break March 15th to March 22nd

With the current work needed by the team, the main location where this will be carried out is on the university campus on the SCCT 4th floor. Any breaks or potential hazardous weather will freeze the greatly slow down the available work for the team to accomplish.

The weather did halt our project for one week during Sprint 1, however we were able to meet online and discuss plans for the upcoming goals to be met. We may have a pause in engineering tasks over spring break for a weeklong period.

# Risk Mitigation

Author: Nick Miller

There are not many risks that could occur in our setting. The only thing that might hold up development is if a teammate has a lack of time or ability to work on a project for a considerable length of time. However, we mitigate and potentially avoid these risks altogether by looking ahead at our schedules, planning around them and informing our teammates ahead of time if we see a potential conflict.

The probability of hardware failure is relatively low, but not zero. Two RAM modules in one of our R710 servers have already failed; as the hardware ages further, more failures may occur. RAM failures are relatively low-impact, but a CPU or hard disk failure could dramatically impede progress.

Organizational requirements have revealed themselves to be a far larger risk than previously anticipated. CSU’s University Information Technology Services maintains strict regulations on network access and usage. The processes of connecting our servers to each other, the rest of the machines on campus, and the Internet have become highly time-consuming and potentially impossible. We need to connect our hardware switch to the LAN and make each of its ports available on the network, but the switch seems to have been blacklisted entirely from the university network.

The only possible solution for this problem is to collaborate with UITS through a proposal submission and successfully negotiate network access. Had this been previously known, collaboration could have begun far earlier; instead, we can only hope that UITS grants access within the next couple of weeks.

UITS heavily prioritizes cybersecurity, which we agree is vital. However, this may pose another risk for our ability to complete the project. None of the team members are trained in cybersecurity beyond the usual course requirements and an elective or two. If our cloud is deemed insufficiently secure for connection to the campus network, the time required to research and implement hardening measures may well preclude us from successful execution within this semester.

Risk Identification:

* Research current, general cloud risks, such as security, cost, and resources.
* Security: Identify vulnerabilities and threats to the cloud environment. This may include research as well as testing.
* Cost: Ensure that usage of resources like power and networking will not become prohibitive by identifying possible changes in costs and ways to limit excessive usage.
* Resources: Once deployed, track whether the current resources meet user demand. Identify possibilities for expansion or conservation.
* Data management: Ensure that data on the cloud is backed up and secure. Identify potential risks to data integrity. Check physical hard drives for problems. Identify candidates for drive upgrades, if possible.
* Gather information and guidance from other University roles, such as the Cybersecurity department.

Risk Mitigation:

* For extended use, create a team tasked with cloud administration and risk management.
  + Permanent faculty members to manage and participate in administration and risk mitigation.
  + Temporary/adjunct faculty members or instructors to contribute to their areas of expertise.
  + Senior students in computer science and information technology contribute to risk mitigation, maintenance, and identification of upgrade paths.
  + Senior students in cybersecurity to harden and maintain the cloud’s security measures.

# Test Plan and Test Procedures

## Test Plan

The test plan aims to verify that the private cloud environment, using two PowerEdge R710 servers, one PowerEdge R730 server, and a Cisco LAN, meets the design specifications and platforms according to the functional requirements.

Testing will cover the following components:

* Hardware setup: PowerEdge R710, R730, and Cisco LAN.
* Software configurations and installations, including virtualization platforms, could management tools, and networking.
* Internet of Things within the cloud environment.
* Performance, security, and disaster recovery plans.

Software:

* Virtualization software
* Cloud management software (OpenStack)
* Operating systems for servers (Linux)
* Networking configurations (iDRAC, IP, LAN, subnet)

## Test Procedures

### Procedure 1: Sample Test Procedure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Required Actions** | **Expected Results** | **Comments** | **Result** |
| 1 | Power on each server (R710’s, R730) | Servers power on without errors | Check physical connections before powering up | Complete |
| 2 | Install Operating System (Ubuntu/Linux) | OS installs successfully without errors | Ensure current OS version is used | Complete |
| 3 | Install and configure virtualization software | Virtualization software installs and configures correctly | Verify compatibility with hardware |  |
| 4 | Connect servers to the Cisco LAN | Servers can communicate over the network | Ensure all cables are securely and correctly connected | In-Progress |
| 5 | Set up network configuration (IP addresses, LANs, subnet, iDRAC) | Network interfaces configured correctly | Verify no conflicts in allocations. | In-Progress |
| 6 | Deploy a virtual machine on each server | VM is created, powered on, and accessible from the networks | Verify resource allocation |  |
| 7 | Test server-to-server communication (ping test) | Servers can ping each other without packet loss | Verify correct routing and IP configuration |  |
| 8 | Stress testing on servers | Servers maintain performance and stability under load | Monitor CPU, RAM and network usage |  |
| 9 | Test security configurations (firewalls) | Firewall and user authentication function correctly | Test access control for unauthorized users |  |
| 10 | Perform disaster recovery test | Data is successfully restored from backups | Verify recovery time |  |

# Lessons Learned

Authors: Nick Miller, Kelly Payne

Cycle 2 saw us moving beyond the basics of server usage and Linux installation into our first interactions with a functional OpenStack-based system via DevStack. DevStack is a set of scripts used for setting up an OpenStack development environment. It allowed us to set up an OpenStack instance quickly and use that instance to test OpenStack’s capabilities for users and administrators. Setting up DevStack was a difficult but educative task which taught us the basics of setting up and using OpenStack, range of possible problems with OpenStack, the value of understanding the Linux command line, and how to use the command line to diagnose and fix DevStack and OpenStack bugs.

We researched, discussed, and learned about a variety of other topics as they arose. OpenStack system architecture was important as we determined which servers would manage which cloud components. Physical server hardware came up as we received our server rack, then realized it couldn’t accommodate our machines without additional hardware. We researched and discussed RAID, evaluating RAID levels for speed and fault tolerance in order to select suitable configurations. Disk encryption via LUKS was a notable discovery, topic of discussion, and eventual addition to our systems.

There are a few things we learned in the first cycle. We figured how to bypass password restrictions for PowerEdge 730 server, which involved opening the server and switching some pins around which was something new and intimidating. We also learned how to configure the iDRAC for the servers and configure the wiring for the PowerEdge R710 & R730. Additionally, we learned how to download and install Ubuntu using a flash disk. In Cycle one we learned how to configure the iDRAC for the servers and that iDRAC 6 & 7 has much more information than we realized. We also learned how to download and install Ubuntu on a flash disk and servers, and not to initialize the RAID as that will cause you to have to reinstall Ubuntu and KVM again.

# References

<https://www.dell.com/en-us/shop/power-cord125vusa-customer-installation/apd/310-5108/power-cooling-data-center-infrastructure#ratings_section>

https://www.dell.com/support/product-details/en-us/product/poweredge-r710/docs

https://dl.dell.com/manuals/all-products/esuprt\_ser\_stor\_net/esuprt\_poweredge/poweredge-r710\_owner's%20manual\_en-us.pdf

https://www.gotomojo.com/wp-content/uploads/2019/07/Dell-PowerEdge-R730-Owners-Manual.pdf

https://www.itcreations.com/user-manuals/r710/dell-poweredge-r710-system-owner's-manual.pdf

https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/govern/assess-cloud-risks

https://www.linuxtechi.com/how-to-install-kvm-on-ubuntu-22-04/