OpenStack Capstone Final Postmortem Report

Justin Anderson
Jack Nagel
Cesar Ramirez
John Rensberger

Table of Contents

1. Postmortem Results

1.1 Things That Went Well

- We were able to meet 4-5 days a week and averaged 12-18 hours.
- We were able to keep up with class-required documentation each week.
- We were able to reproduce and build on to last year's project.
- We were able to meet each sprint's required tasks.
- We started knowing little to nothing and left knowing a great deal.
- We were able to collaboratively tackle bugs with minimal interference with each other's work.
- We were able to communicate with our mentors on a weekly basis. If needed we met twice a week.
- We were able to meet with Dr. Denton and Dr. Ludwig about a specific and desired use case.
- We were able to document our process for future usage.
- We were able to discover Savanna and integrate it to meet our core requirement.

1.2 Things That Did Not Go Well

- We were unable to continue on last year's project as it had been decommissioned and replaced by a non-functioning OpenStack environment causing us to start over from scratch.
- Early implementations were slow as we were unfamiliar with OpenStack which caused delays as we needed to spend time debugging errors that we didn't understand.
- We were unable to integrate the C.S. department's LDAP authentication into OpenStack.

1.3 Lessons Learned While Doing The Project

- We learned that it's best to meet as much as possible. This allowed us to keep up with our schedule and take a break once in awhile.
- We learned that collaboration on tasks eases the time required to finish the task.
- We gained a firm understanding of OpenStack and its components, quick releases, etc. We had to be active in order to keep up.
- Using Google Docs, Google Groups, and Trac eased communication and collaboration of documents and work.
- We learned that manual installation of OpenStack gave us a better understanding

of how the system works compared to the DevStack installation script.
- We learned that error logs help tremendously when it comes to debugging.
- We learned that when we got something to work, we needed to document or keep track of how we did it. This allowed us to perform quick re-installation if necessary.

1.4 What We Would Have Done Differently

- It’s hard to say, really. The initial learning curve for OpenStack is pretty steep, but after we got moving we were able to steadily continue progressing forward. We really did start working with everything fairly early on, but it still took us a long time before we were really comfortable deploying OpenStack.
- Using DevStack early on just to start playing with OpenStack. DevStack is an awesome tool to give you a working environment in literally about 10 minutes, but it can only be treated as a temporary testing environment.
- Started manual multi-node installations instead of using DevStack. The unfortunate thing about this is that DevStack is built from source and it’s much easier to build OpenStack from packages within the OS. This means it’s best to use the latest stable release which is behind what DevStack is using. As a result, it’s hard to reference DevStack’s configuration files as they can be different.
- Assigning tasks a little more clearly to team members as we all tried to contribute equally. Perhaps if certain people would have worked more specifically with components we could have progressed more quickly. It’s hard to say that would really help though as basically every component complements another.

1.5 Recommendations for Future Projects

- Be active with development communities. Report bugs and use public chats/forums for help.
- Be motivated to learn new technologies. Don’t let failures hold you back.
- Sometimes it’s better to go ahead and implement something rather than trying to become an expert from reading documentation.
- If you’re working on an emerging solution, expect to always be catching up.

2. Project Size and Effort Estimates

2.1 Effort Estimates

Task	Estimate	Actual Size
Research	50 hours	~ 100 hours
Deployment, Testing, and Debugging	400 hours	~ 850 hours
Documentation	30 hours	~ 50 hours

Research was learning and understanding OpenStack, Hadoop, and Savanna. Documentation was documenting our process. Deployment, Testing, and Debugging was the process of installation the required packages and software, making sure they worked, and fixed them if they didn't work.

2.2 Project Effort Breakdown

Project Area	Effort (%)
Documentation	25%
Implementation & Debugging	75%

Our documentation includes installation and configuration guides, user guides, and demo videos. Implementation & Debugging was the effort put towards installing and debugging OpenStack.