Alexander Mock
6/26/22
CS 470 Final Reflection

Presentation:
https://www.youtube.com/watch?v=igotjUpKUAc&ab_channel=ZanderTheGreat

Experiences and Strengths

Both The full stack I and II courses prepared me for future opportunities of creating a web app, locally hosting, and migrating services to a cloud. I learned and experienced the skills that are required for cloud software management, such as first creating a locally contained app like docker. This allowed me to create images, and in conjunction use docker compose to bring together all of the parts to prepare for migration. We could use this knowledge for locally based operations but in our case we took it a step farther. AWS or Amazon web services was the basis for our cloud operation. We configured their software to provide services to an organization. This specific course focuses in on the serverless infrastructure.

As a developer I feel as though I have many strengths. Understanding various languages and having the ability to apply them to where they are needed. I have the most exposure in Java, followed by C++. But I have used C#, Scheme, Prolog to name a few others as well. All this exposure helps me apply the logic needed to fulfill the criteria that is to be met. This leads into the various types of applications that may be for front end back end, GUI related ect. Utilizing other tools such as Node.js, Mongoose, seedgoose and others also play a part in my arsenal.

Roles I feel prepared for are, Developer, Tester, ie roles within a scrum team. With the exposure I have gotten I feel as though working on the frontend or backend of services would be suitable. I have been interested in the security side of things as well as the network infrastructure. With my current job experience I have worked with routing and switching as well as managing infrastructure and antivirus/firewalls. At this time I am not sure If I want to move towards software development or more of a network engineer position.

Planning for Growth

Microservices are an approach in which a single application is composed of many independently deployable smaller services. Instead of building a single service app, microservices split an application into sets of smaller, interconnected services/functions. Generally they are connected by APIs to other services.

Microservices along with Serverless Cloud solutions can be seen as the future of modern tech. A single application alone may consist of multiple microservices. These have the ability to interact with a range of resources.

It can be difficult to isolate the root cause of errors, trace requests across your environment, and understand dependencies. In order to troubleshoot or optimize, teams will need visibility into how their functions and services interact with each other.

As for scaling, coming from a cloud based solution it is generally easy as long as you have the funds to do so. Cloud solutions typically offer pay for use models that allow for easy expansion in a multitude of ways.

In order to predict the cost you must first find the practicality of what you are doing and what functionality you need to start. At this time you can work out a quote with the service provider, in our case it was amazon. Getting the initial quote you will have your basis setup and then you can look into what projections will cost you based off growth factors.

Speaking of cost and predictability, in our case we used containers for local and cloud services. Between the two It can be argued that the cloud may be more predictable in nature. Speaking from experience in the field other factors such as ordering tech, servers, and storage have been increasingly hard to order. This comes with long wait times and often to get something quicker means a bigger upgrade and paying more money. As of late it has been hard to predict future costs because of what the world has been going through. Its been unstable. Big service providers have such a large capacity that they can

still handle the loads coming in, which in turn gives greater clarity on what to expect hence better predictability.

There are lots of pros and cons to local vs cloud. It really comes down to a needs assessment.

Some Pros to local network setups:
Faster connectivity
No need for internet
Can be setup as internal only which increases security from outside

Some cons to local network setups:
Expansion is more labor intensive
You are responsible for security and setups
Need to check compatibilities
Expansion can be time consuming to get tech/less predictability

Some pros to Cloud setups:
Easy Expansion
Compatibility is integrated with the services
They integrate base levels of security

Some cons to Cloud setups:
Need internet to access
You're delegating the setup to provider which may not meet all of your customized needs

In general for cloud use the big factors are elasticity along with their pay for use models. This is a big selling point as its very easy to implement and get up and running. Taking the load off of the team for all the setup, implementation, supervision makes it so the developers/team can focus on the app/service quality which will make a better project in all.