I’ll start off by saying – working with Azure is really cool! In the past, I’ve been skeptical of Azure, and particularly with Docker, but in working through some of the issues and side tasks that I took on during project week, I see how its flexibility really helps during the testing and troubleshooting process.

The first big takeaway from this project for me is the understanding of how NSGs work. I have firewall experience through my career, so the NSG concept comes fairly straightforward for me. But, NSGs work nearly the exact same way as do firewall rules, and it is important to allow the various ports on both the source and destination, not just the main direction that traffic is flowing. For example, setting up Filebeat on the Web servers required TCP 9200 to be opened up on the two Web servers for outbound traffic, but communications still won’t be able to take place unless TCP 9200 is opened inbound on the Elk NSG. I saw a lot of struggling through that thought process from the questions asked during the week. When I had struggles making sure the NSG rules were working properly, I made rules to test the desired traffic flow using a simpler protocol to see – ICMP. Once I could ping the servers in the way I was seeking, it was just a matter of adapting that same rule for the needed ports. Using that way to test made wrapping up the NSG rules fairly quickly and without too much pain, for me.

Another big takeaway for me is to not be scared of it. In my project setup, to get around the quota issue when building the Elk server, I chose the option to create a new virtual network in a new region, and peer the virtual networks together rather than delete the Web server. Setting up the peering was a piece of cake and establishing traffic flow between the two virtual networks wasn’t hard at all. Going with that method, I went against the grain of what most students chose so I had to roll through the issues myself. That was no big deal, and I was able to work through the issues with no major trouble. A couple issues that I found, such as the virtual memory hangup when starting the servers, I passed along to the rest of the group in case they ran into the same issues.

Spending Thursday working through the issue with changing the default passwords for ElasticSearch and Kibana was also a prime example to not be scared. Working through that process wasn’t easy, and I destroyed many Docker ELK containers in the process of trying to nail down the process for getting the passwords changed. In the end, that process wasn’t needed to complete the project, but working through it was an excellent example to not be dismayed with having to rebuild a Docker container – it’s so easy to spin a new one back up and start back down a different path. The flexibility and ease of Docker made the troubleshooting process so much easier than needing to spin up a completely new virtual machine if problems were encountered. I was able to get through it and passed along what I discovered through the process and dialed in on some Google skills in the process!

In the end – I felt the project was a great way to validate Azure skills, and make sure that we were on our toes on the networking concepts introduced thus far in the class.