Enoch Masih

12/13/20

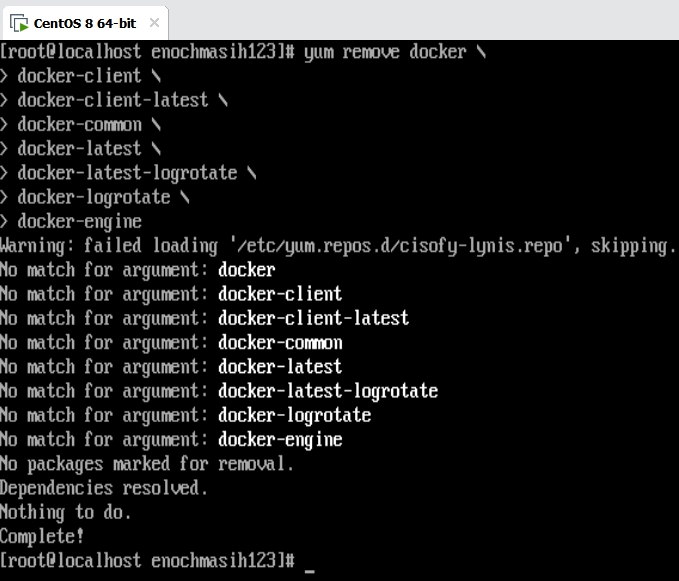
Linux Administration

Containers

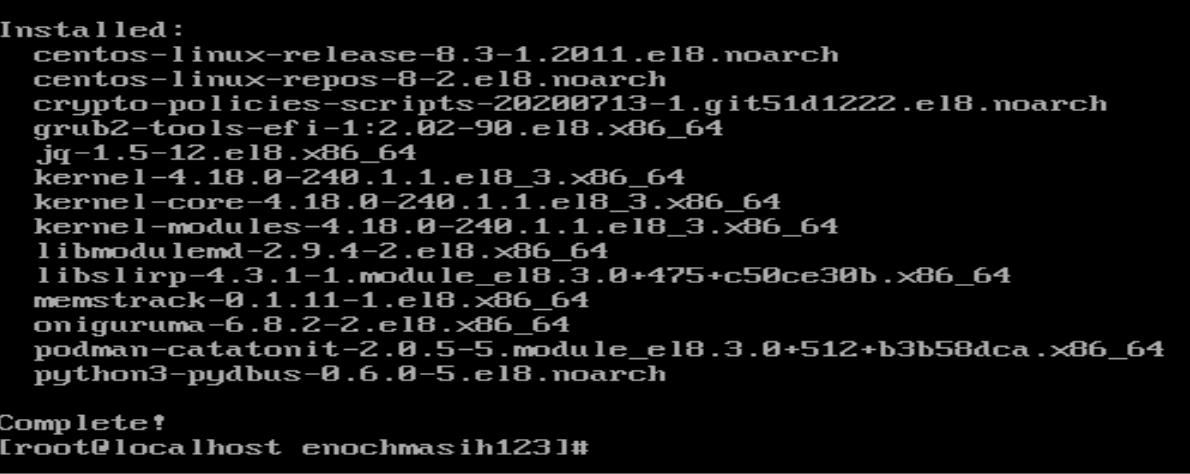
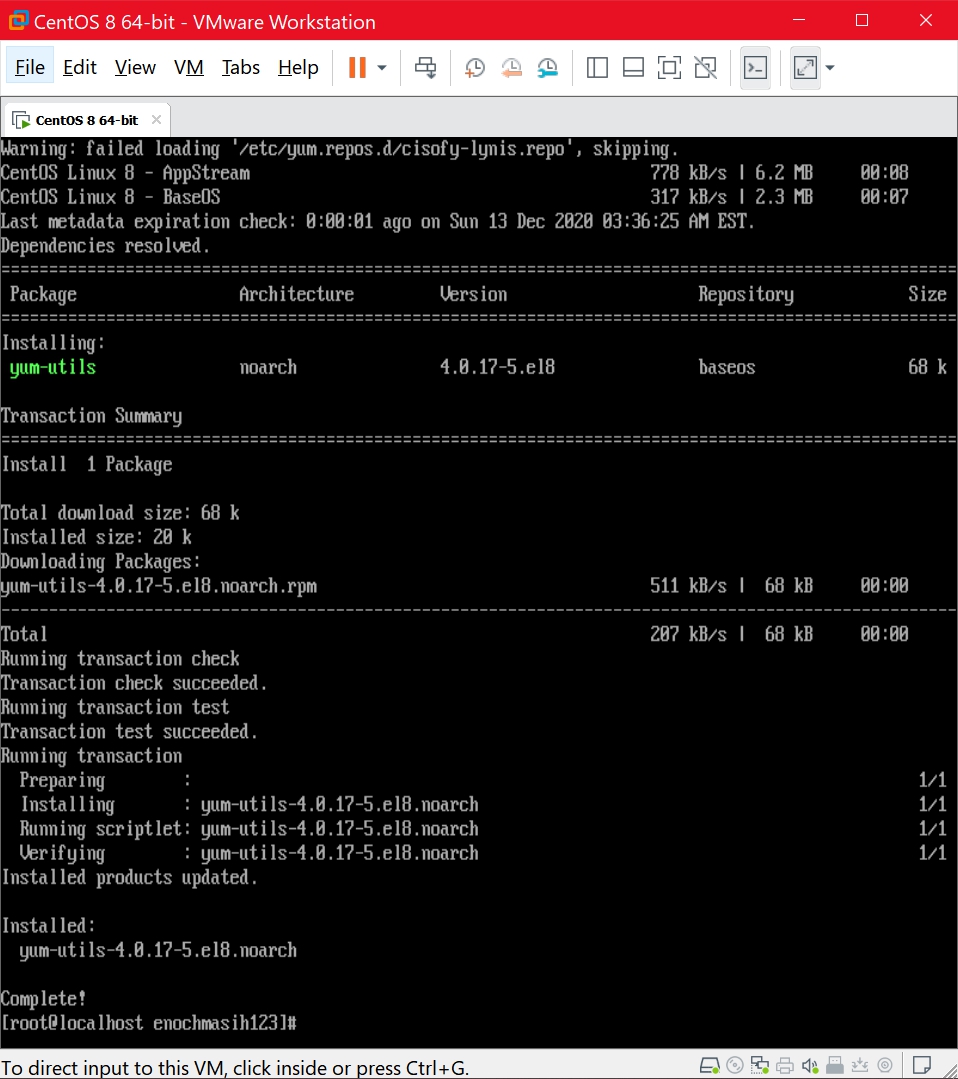
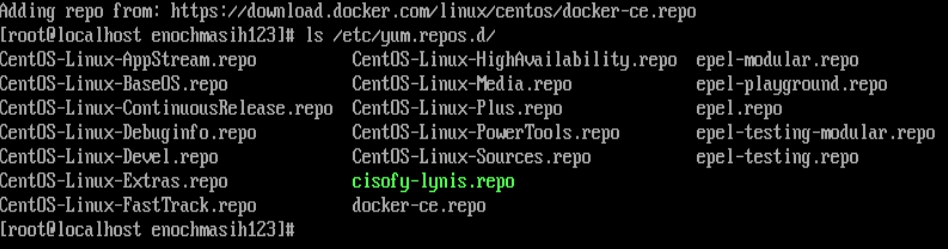
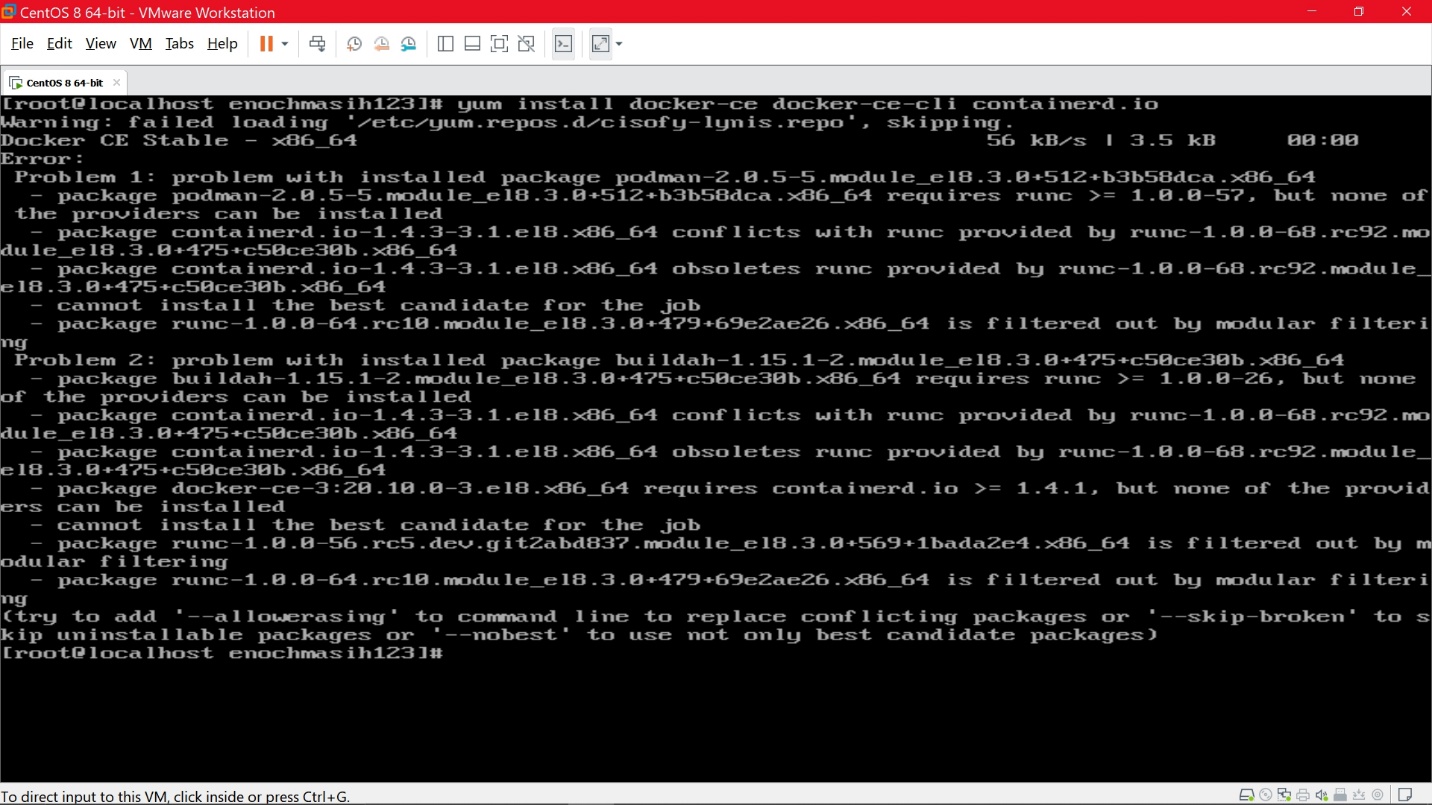
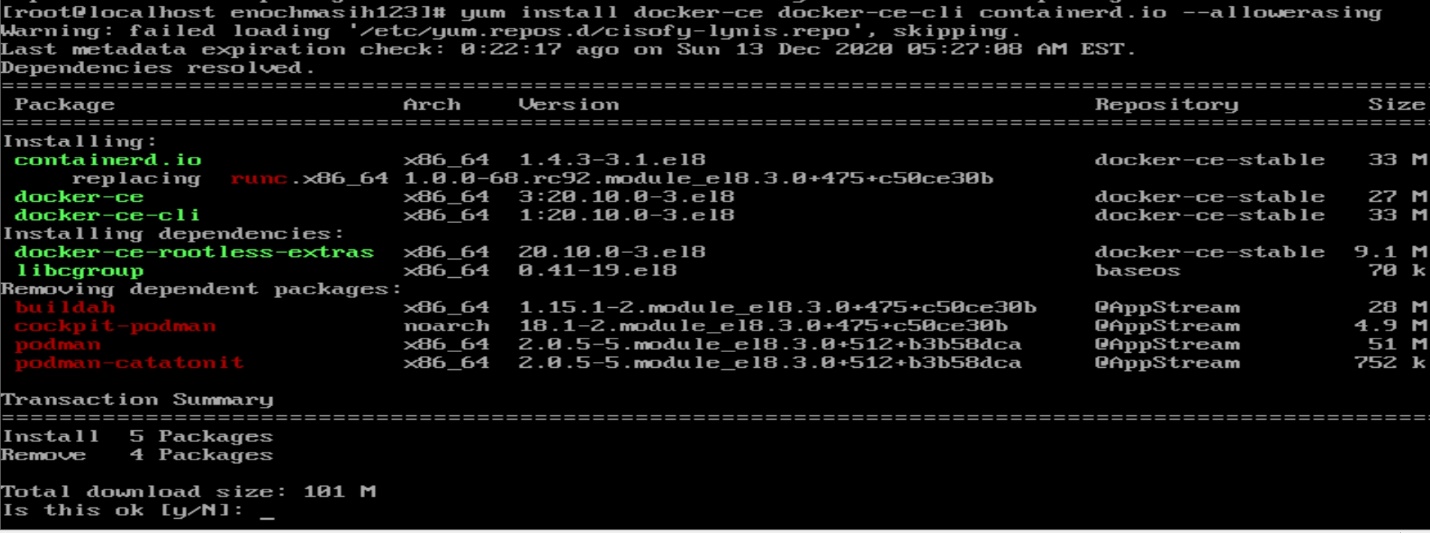
**Docker on CentOS**

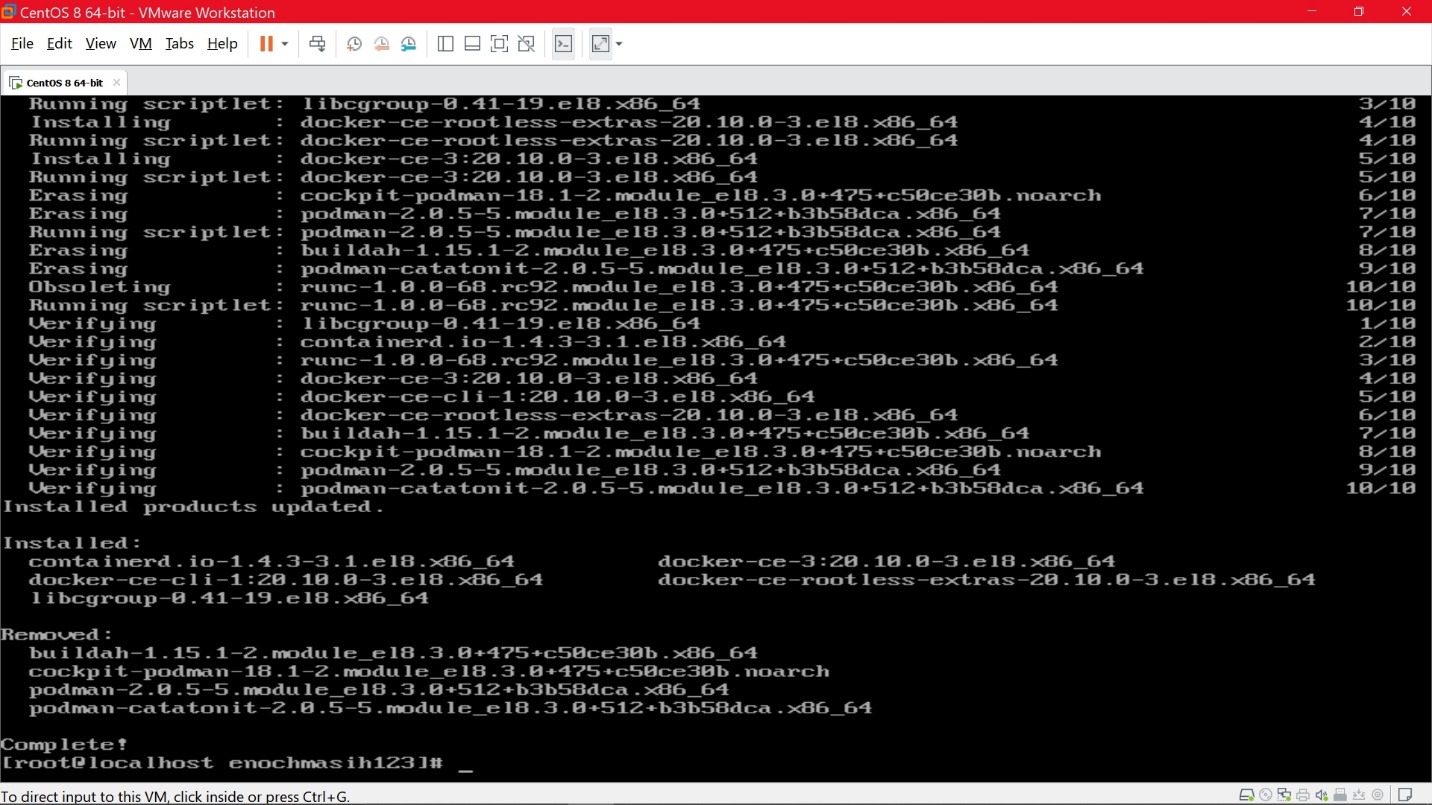
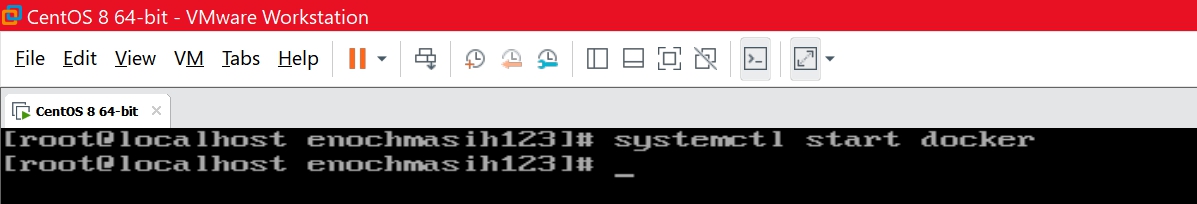
In this guide, I will show step by step how I installed Docker and Kubernetes on my CentOS server. I first took on the task to install docker and then moved into installing Kubernetes. To install docker it is important to first remove any traces of docker from your server if you previously had any. this is a recommended step from the official docker website so as not to cause conflict. Make sure that you are in the super user mode so that you can execute the commands. 

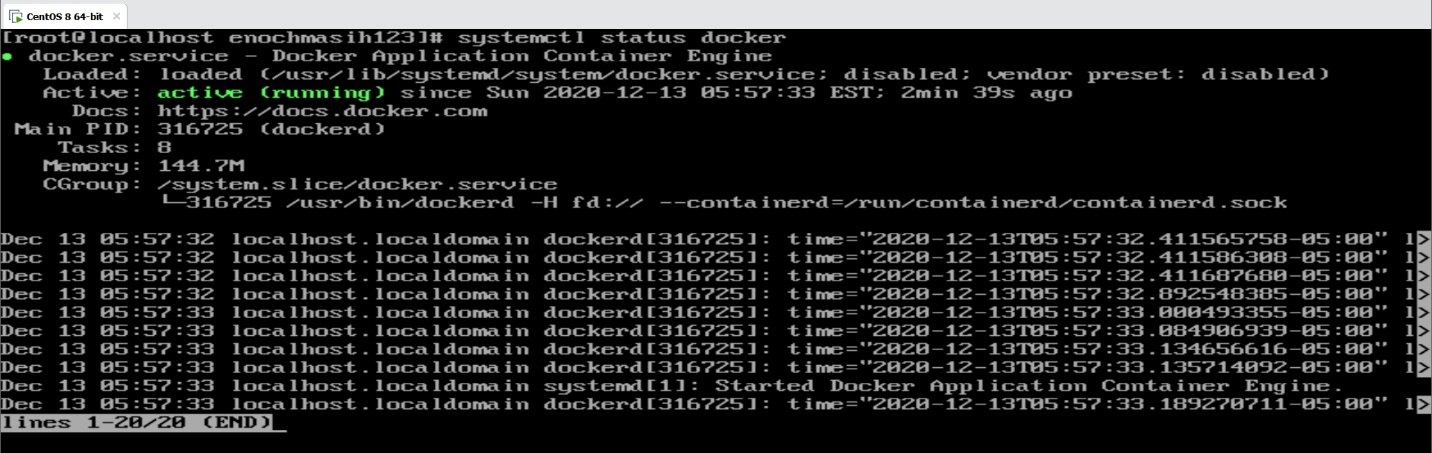
Once you are in super user mode as shown above; execute a set of commands that will remove docker and any of its components with the following commands. I used the backslash to pass multiple commands.

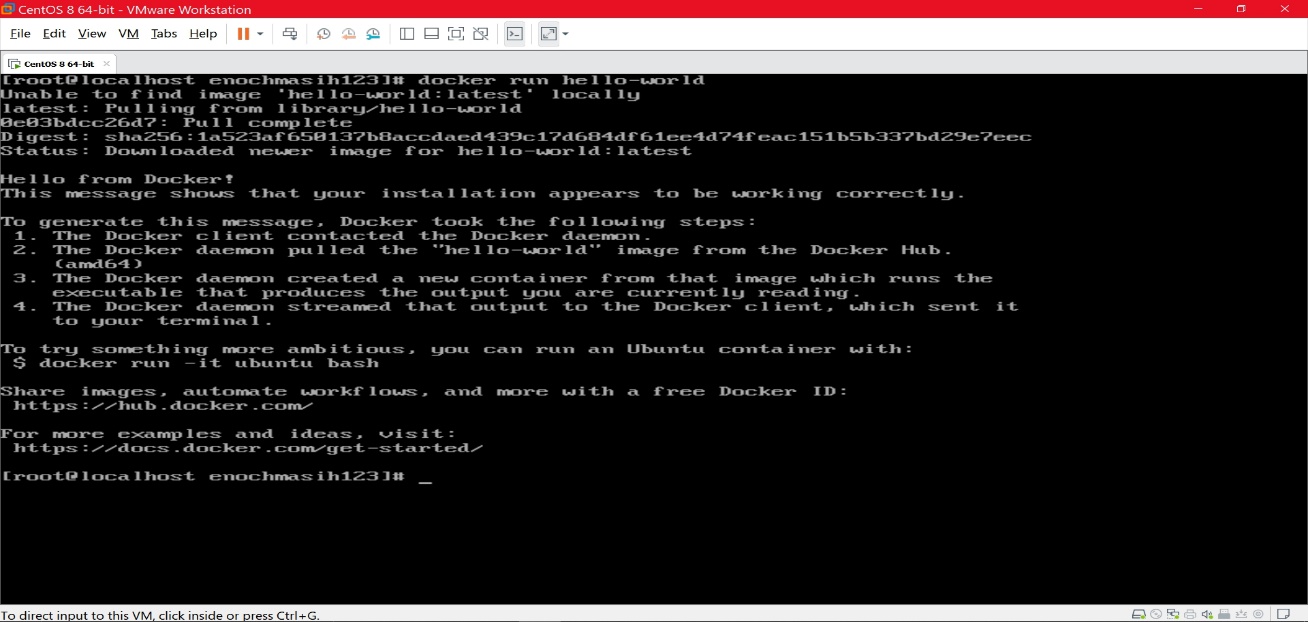


In my case, there was nothing to remove as displayed by the output. I think this is a good step to create a clean environment where you have a ‘fresh-start’. Another step to create a good environment to run docker is to update your pre-existing libraires. I used the yum update command to do so. In my case there were a few updates sizing to 442 MB that I needed to do as shown below. 

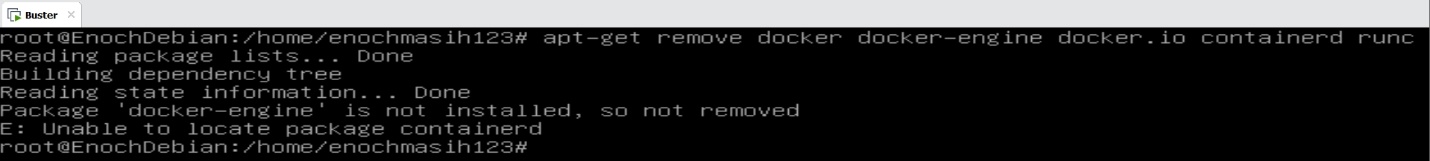
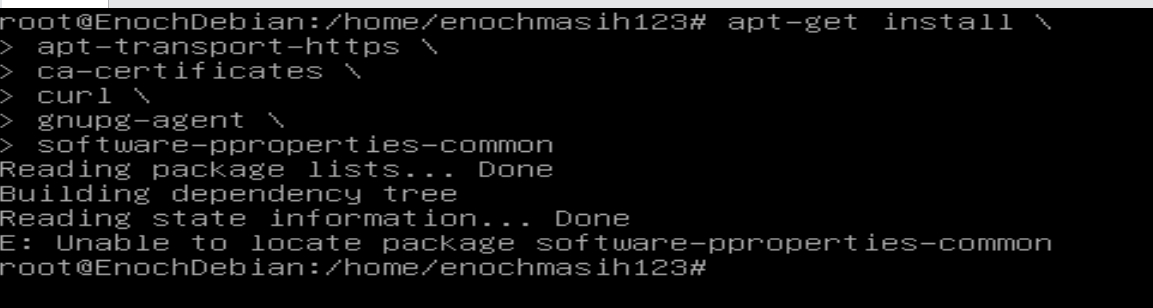
I entered y (for yes) and updated my OS. This took some time to complete. Once it was completed you will get the following message.  We now need to install some tools to set up the repository for docker before we can install docker. We first need the utils package and then the configuration manager. To install these the command is yum install -y yum-utils. Once completed the screen should prompt for the following shown below. To install the next package enter yum-config-manager \ --add-repo \ <https://download.docker.com/linux/centos/docker-ce.repo>. Adding this was much quicker than the others. One quick way to check what repositories have been added from the internet to your server is to check the following directory as shown below. Note that it first added the repository and then when I ls the directory it displayed the docker-ce.repo file.Now that we have the repository set-up in our server we can begin to install docker. To install the latest version of docker use the command I used which was yum install docker-ce docker-ce-cli containerd.io. The ce stands for community edition. When I ran these command I got 2 error messages as shown below.  It appeared that there was a conflict of packages that already existed on my server with docker’s packages. I took the suggestion from docker to erase conflicting packages. I chose to erase as I would rather choose a clean environment to run docker instead of skipping some parts. Skipping some parts may make the docker potentially unusable. While erasing the conflict packages may help isolate issues down the road. 

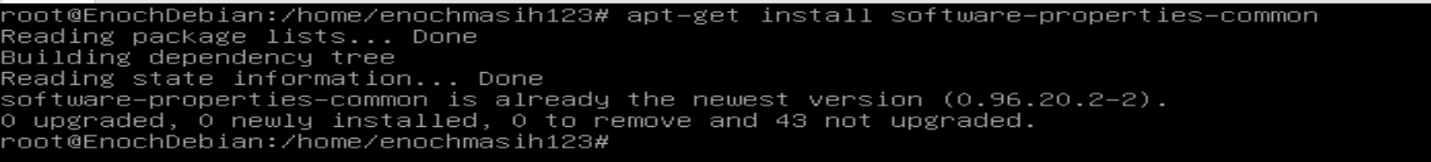
Since encountering an issue is something unusual, I did my due diligence in researching the issue. It turns out that this specific podman issue has been observed in CentOS when attempting to install docker. Once docker has been installed successfully, you will see the screen below Now that Docker has been installed, we need to start the service. To start the service type the following command. If now error is returned that means that docker is now running as shown below. 

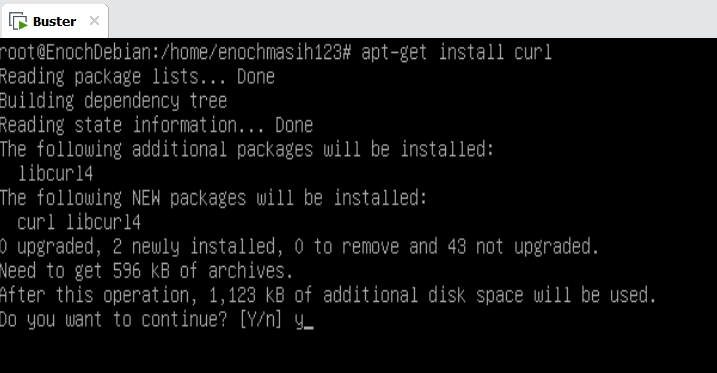
Another way to verify that docker is running is to run the command systemctl status docker. This will return the following prompt 

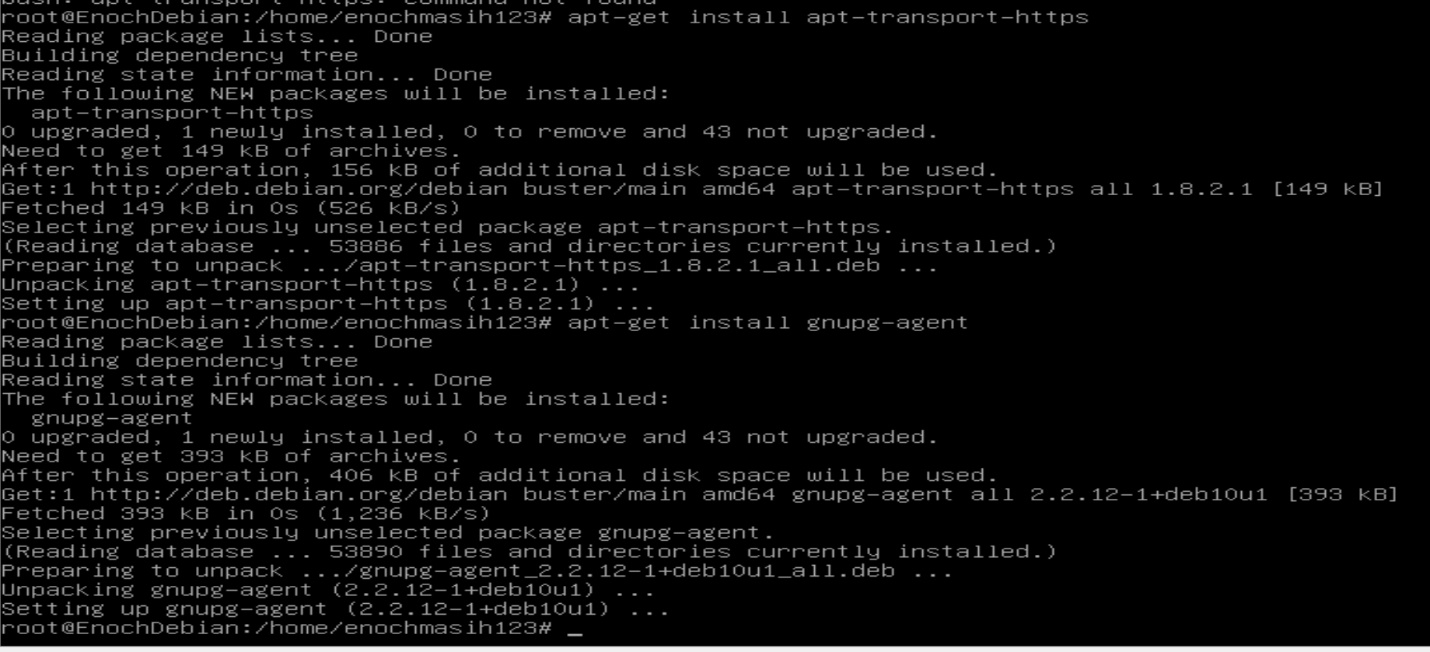
Now that docker is installed and running, we can send a test command docker run hello-world to docker to make sure it is running properly. The output reassured that docker has been installed and running properly as shown by the message below 

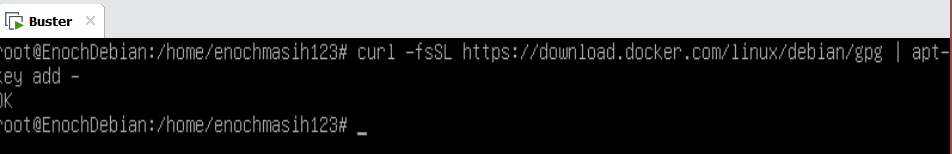
**Docker on Debian**

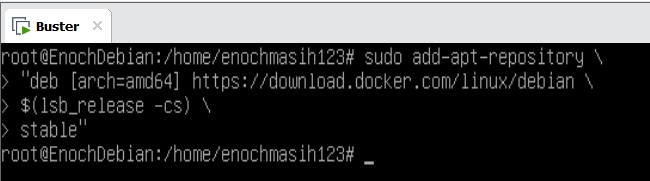
Installation on Debian is similar to that of CentOS with respect to its typology. Remove if docker is installed, installed the proper repositories, and finally install docker. What is different besides the commands used are a couple extra steps that must be taken Before we remove any docker remnant I chose to do an update of my Debian server by running the command apt get update. I ran this my first getting into the root user by running the command sudo su.  Then to remove and docker remnants, I used the command $ sudo apt-get remove docker docker-engine docker.io containerd runc. When I ran the command the output indicated I do not have docker which is true. Now we must install the docker engine. Since there are several ways to install the docker engine I choose the common way which is by repository. To install by repository it is important to have the right tools in your server to do it over https with the correct certificates. I used the following commands to install the needed tools 

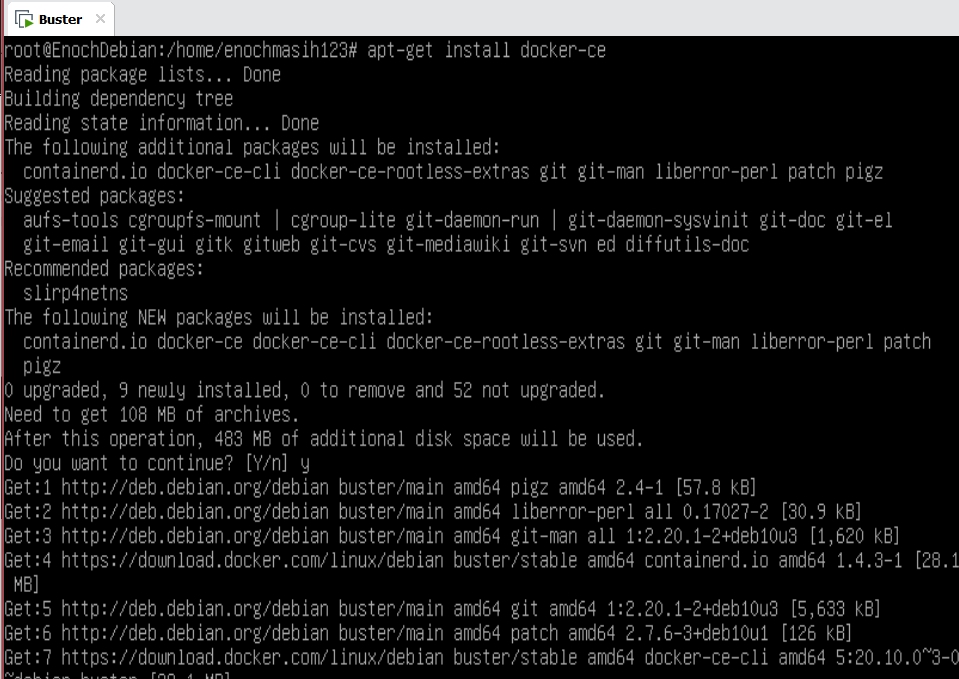
As you can see I got a error back regarding the software-properties-common. looking back at the command I entered I realized it was my own user error-spelling of the word properties. I corrected this and it was successfully able to install the last package as seen below 

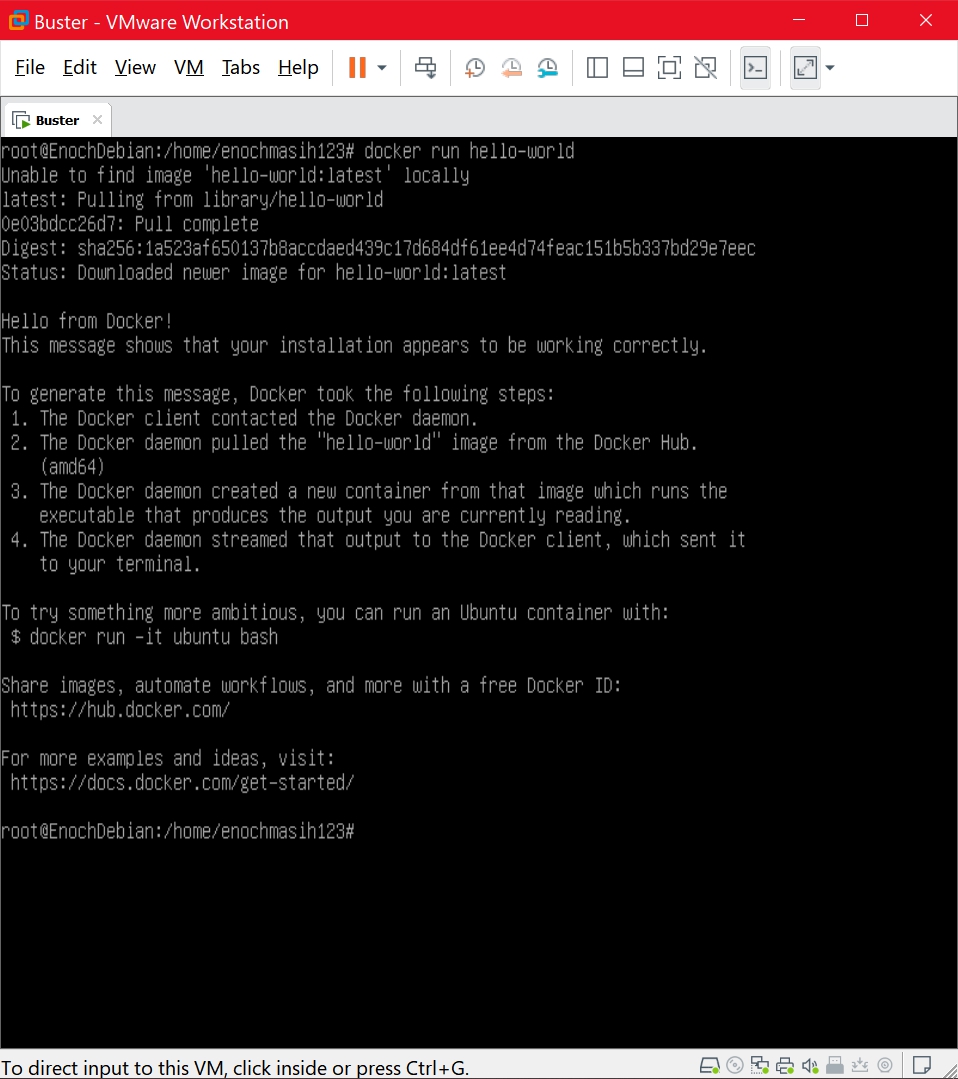
I chose enter each command separately to make sure that each package was installed appropriately. The good thing about linux is if it is already installed it will let you know. 

Here I installed the rest of the packages as well 

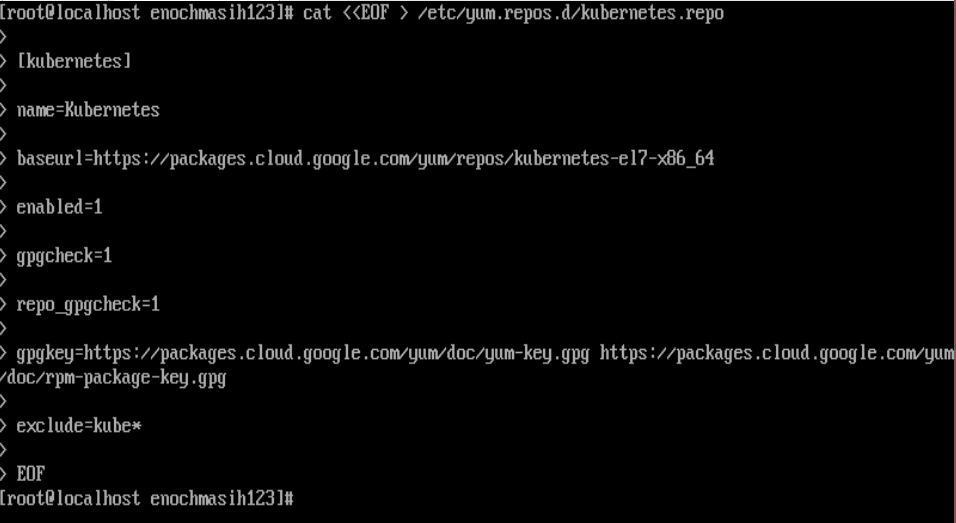
Now that it was verified that each package was installed successfully I can move on to the next step. Next is to enter the address of the repository so that we can match the gpg key. This way we will know that we have an official version of docker 

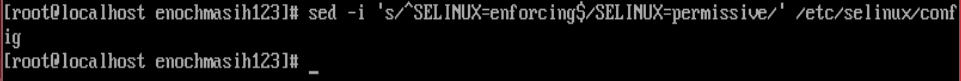
The output of this command said ok. Now that we have the proper tools to get the repository, we can install it. I used the following command to install the repository. 

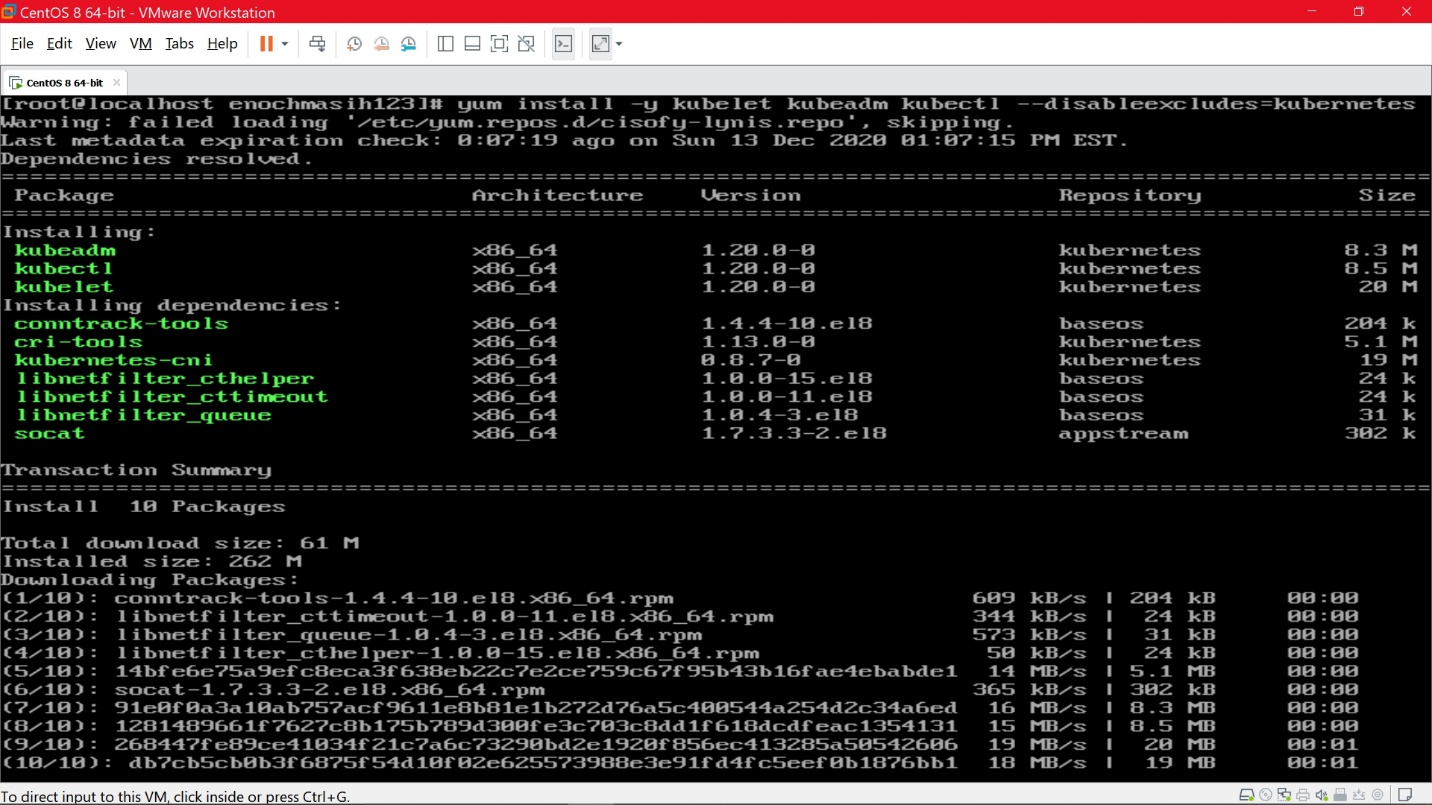
Now I began to install the docker community edition one at a time. to this first enter apt-get install docker-ce as shown below 

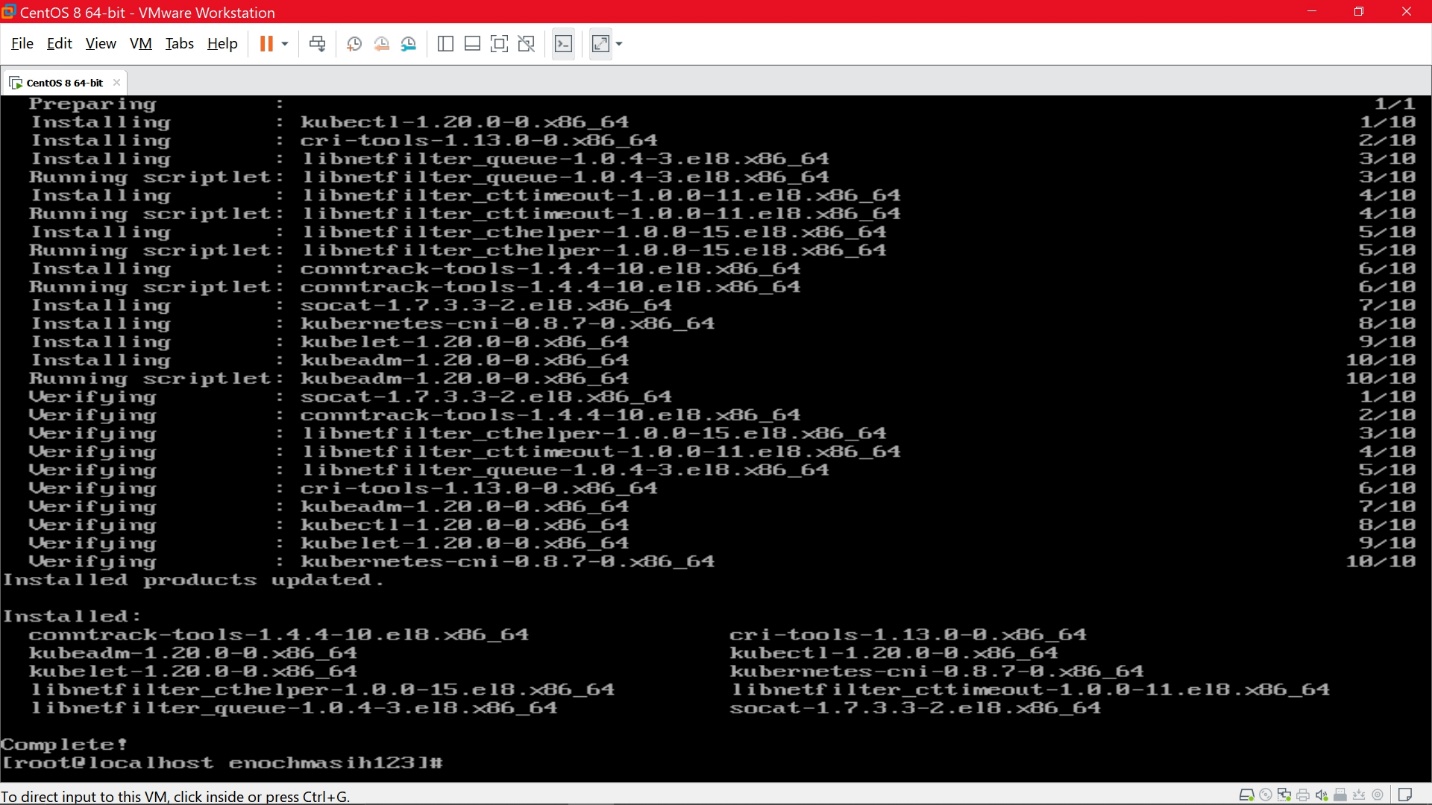
When installing docker it asks for additional supporting packages to be installed to which I said yes to. Once docker was installed I performed the same test that I did with CentOS. I sent the docker run hello-world command. The following was the output indicating a successful install. 

**Installing Kubernetes on CentOS**

Like docker, Kubernetes is also a container that can be implemented on various Linux distributions. Unlike docker, Kubernetes is seen to be much more scalable than docker and offering various storage features. Kubernetes can be installed by various toolkits on the server. I used the Kubeadm which offers the essentials. there is also minikube which is a virtualized environment, kops which relies on amazon AWS, and Kubernetes on GCP or the Google Cloud Platform. For kubeadm and minikube can be installed in my VMware environment as they do not require multi node setup. I performed a single node setup. I first added the kubernets repository. When I typed in the command I got the error that the directory was not found even though the cat command should be able to create the directory file as shown below  I thought this was really odd, but then found the mistake that I had made after a few hours. As shown above, I was using Debian instead of CentOS. The commands are very specific to the Linux distros and cannot be interchanged. Once I hopped onto the CentOS the first command to install the Kubernetes repository was successful as shown below. 

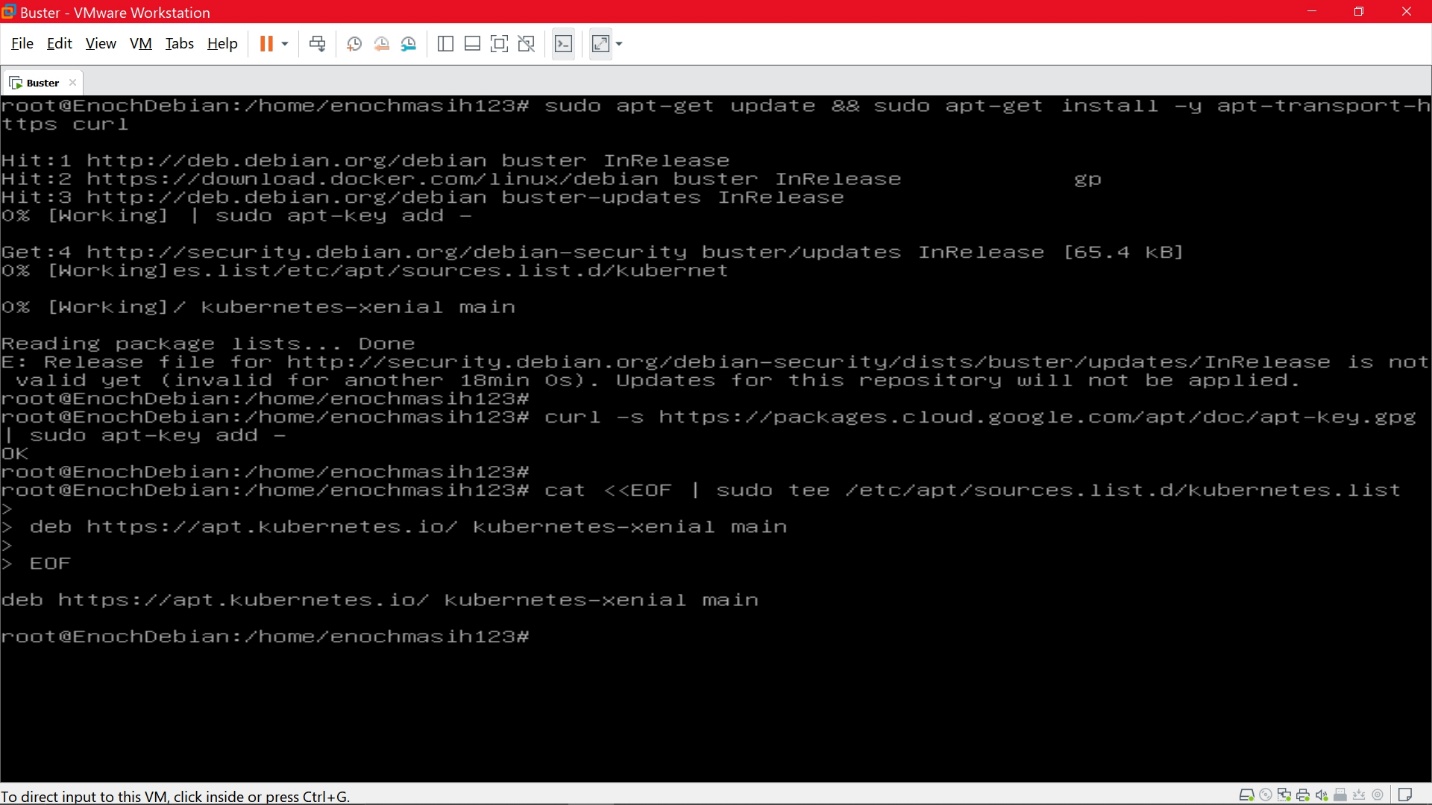
Now it was time to set the red hat version of security called security enhanced linux intro permissive mode. 

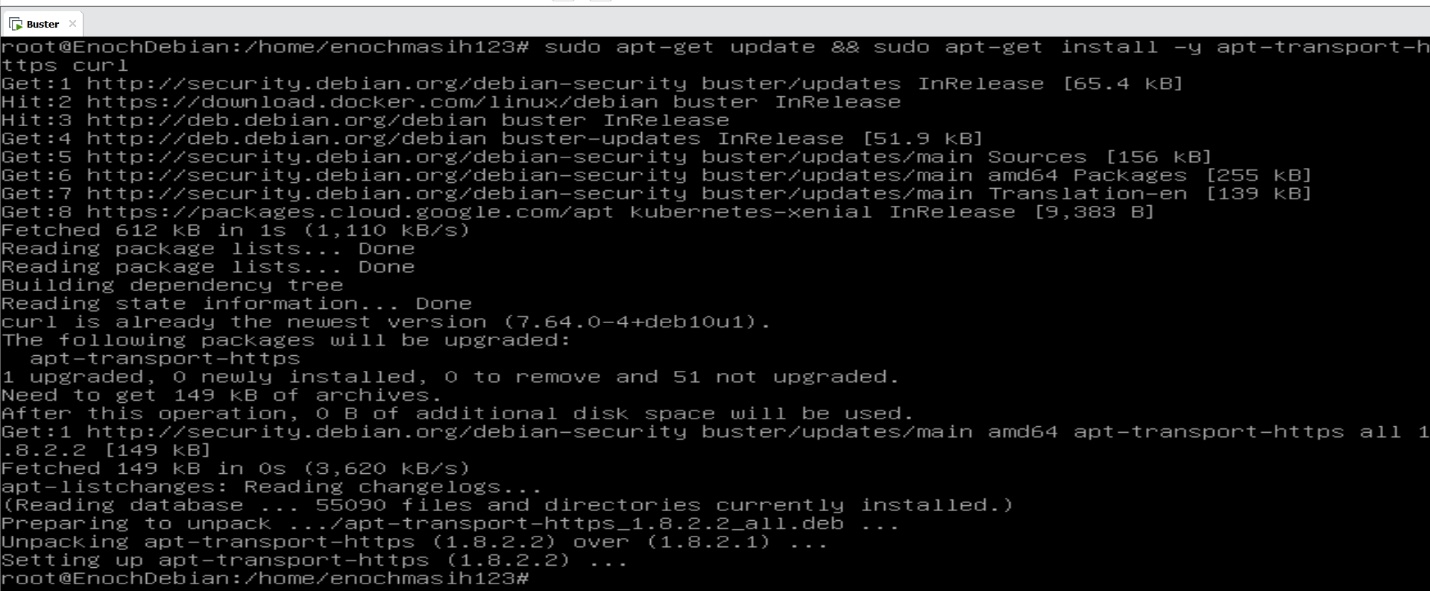
Now I went ahead and installed the toolkit needed. 

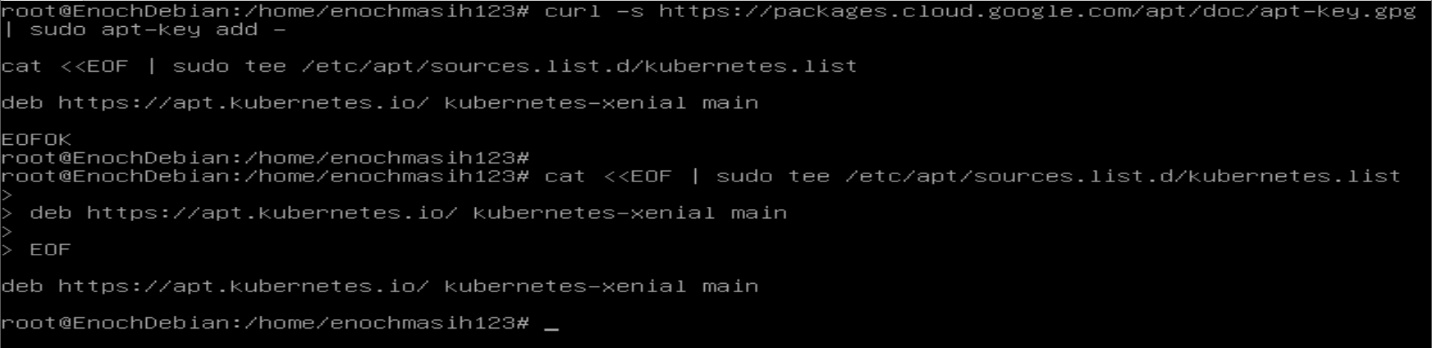
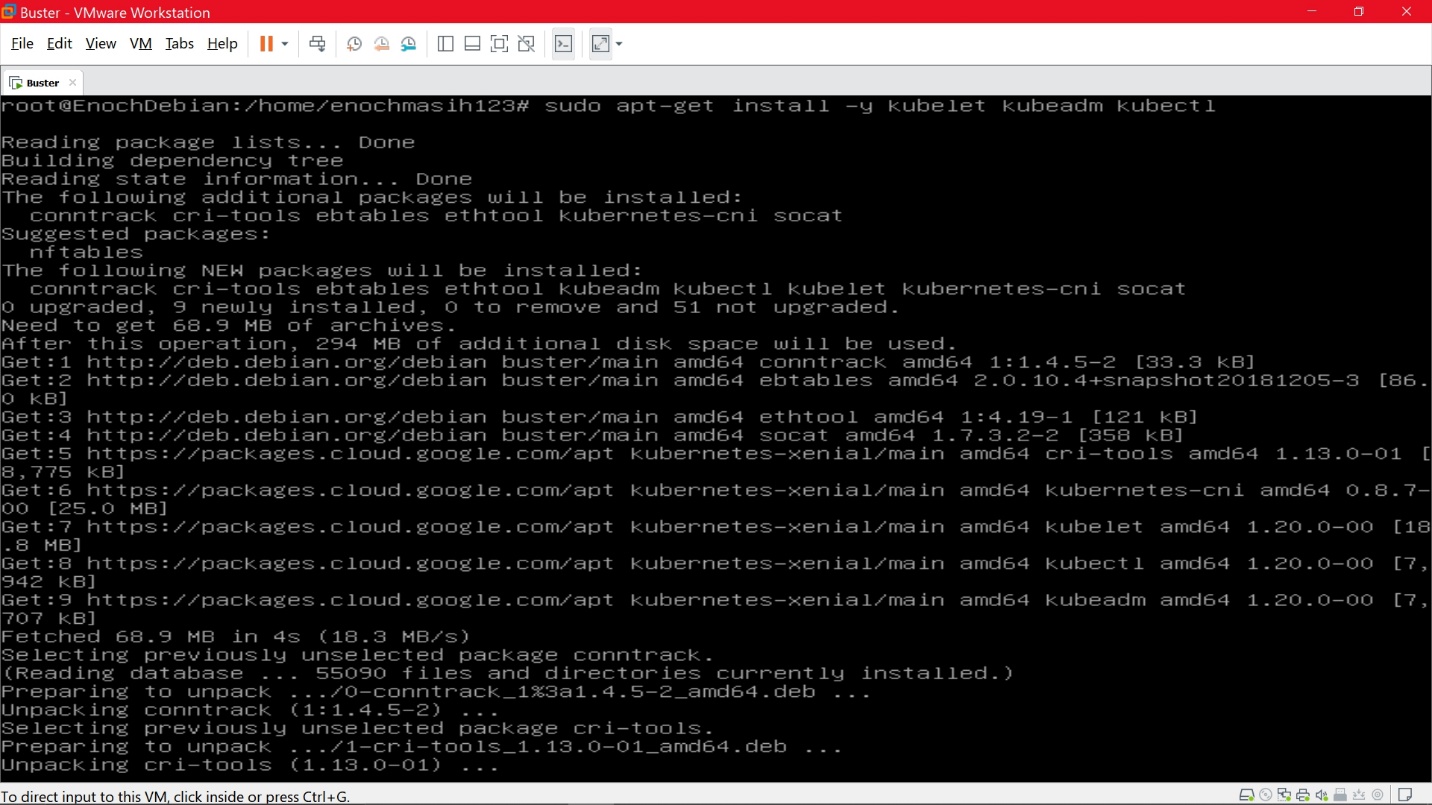
Once finished you will be prompted with the following message 

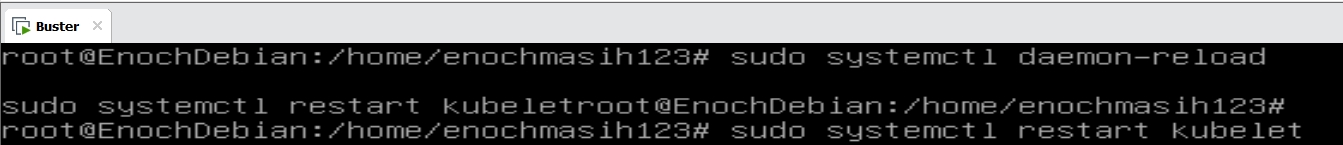
The last step is to enable it using the systemctl command as follows 

**Installing Kubernetes on Debian**

I ran into some errors when installing Kubernetes on CentOS and hope to not repeat the same mistakes here. I followed the official guide using the latest software to do the install. The first step is to setup the repository as well as do some house-cleaning by doing apt-get update. 

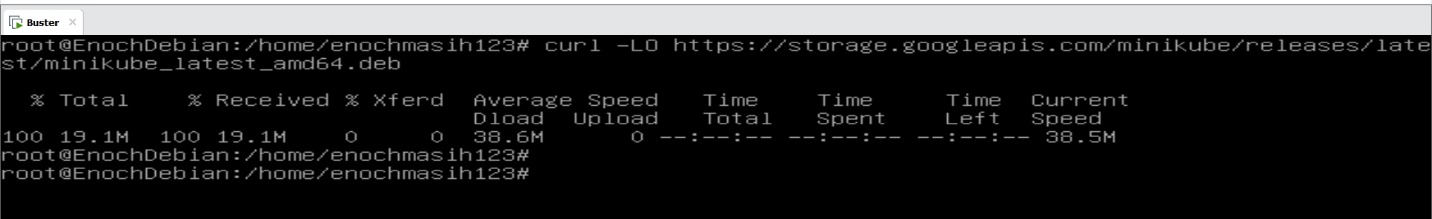
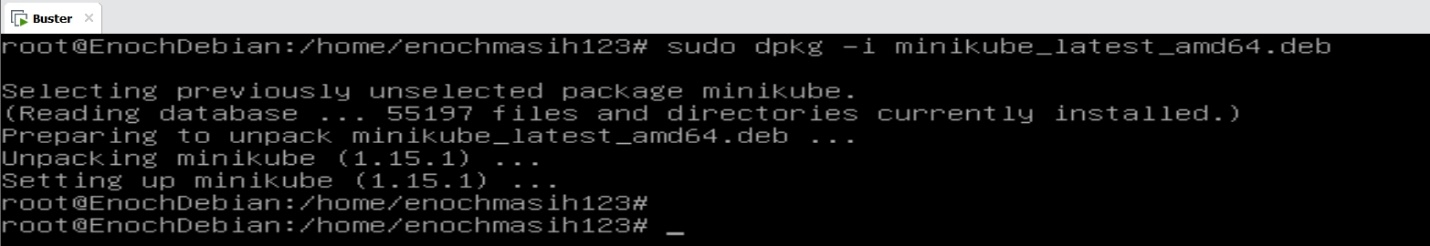
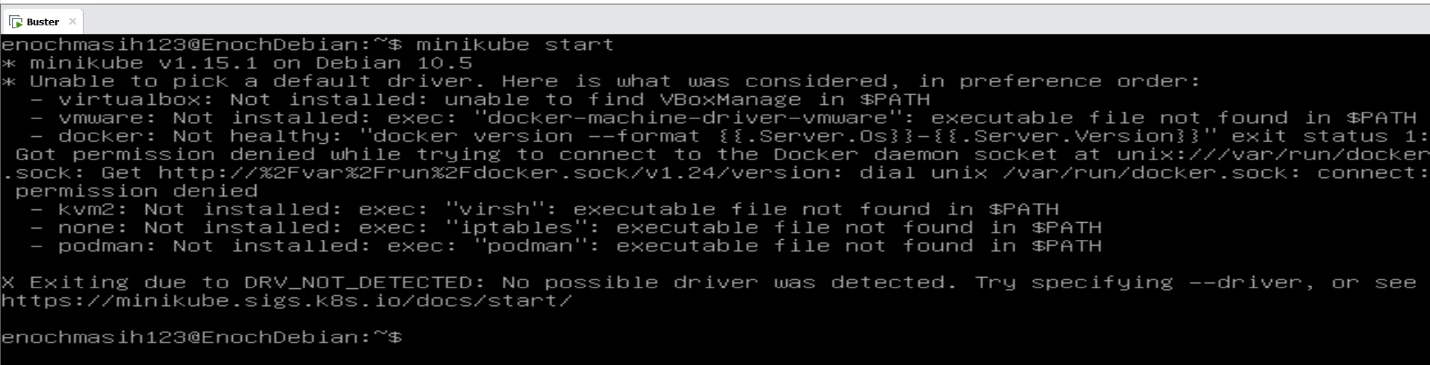
As shown by the error message above I was locked out for 18 minutes. I thought this was a neat way to prevent a DDOS attack. So I waited and tried my luck again later. Once I tried it again it worked 

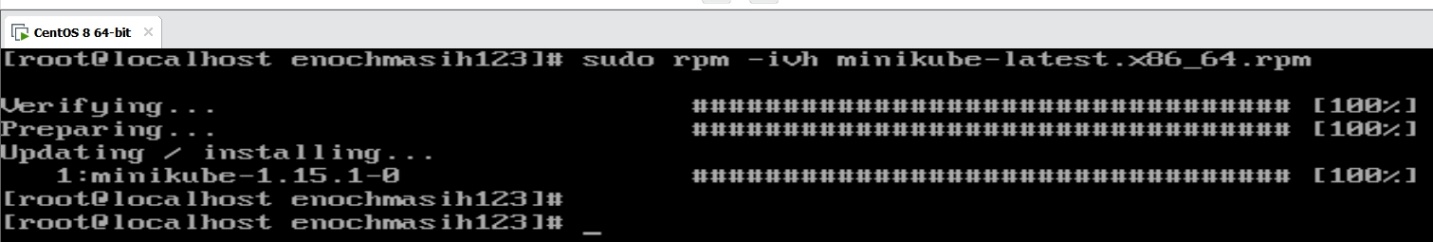
Now it was time to setup the repo and the key If you want you can also verify this key, but since I used the official website I didn’t implement this in my server. Just as with CentOS now I installed kubeadm toolkit 

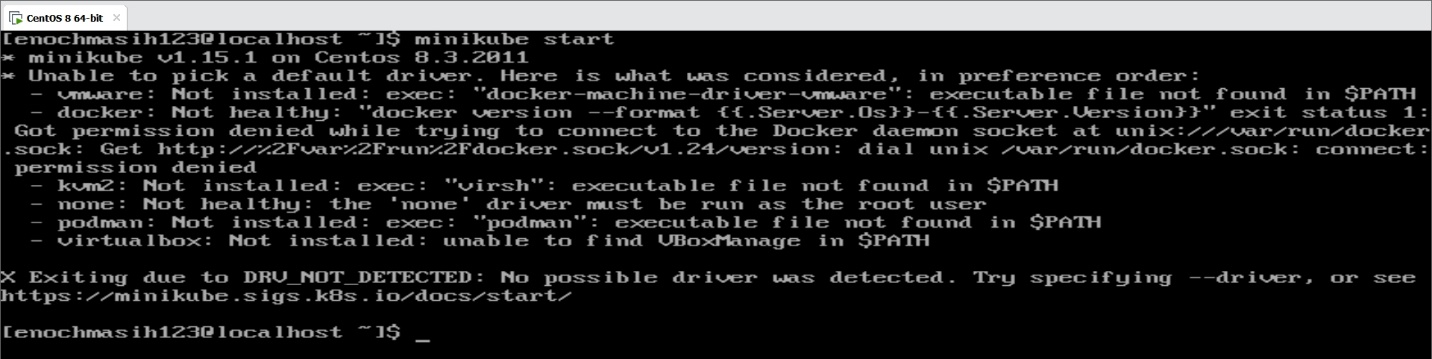
Once install the official website recommends to restart the service to make sure it is recognized. 

Since this was a bare metal installation of Kubernetes I did not setup any clusters or attempt to create a multi node setup. I think for starters kubeadm on single node is great to practice, but to learn minikube may be a better choice.

**Attempted install of Minikube**

One great use of Minikube is to learn about Kubernetes. I first attempted the install on my Debian server. I used the Debian specific packages offered from the official website. First I used the curl toolkit which was installed previously to pull from the official website as seen below. Then I installed minikube using the following command and it was successful as seen below. When I tried to run minikube I was given some errors. Looking further into it, it looks like some specific packages have to be installed as well as some permissions have to be granted on the docker side of things. 

I attempted the install on CentOS. The official website offers several version of packages for different distributions of linux. I used the RPM package. I followed the same steps with curl command as I did with the Debian install I then Installed minikube using the following command 

I ran into the similar error with CentOS as I did with Debian 

So on both servers; even though the installation of minikube was successful It looks like additional steps need to be taken to implement the cluster environment. This makes sense since I only have a single node.

**Summary**

Containers can be a great tool for developers or anyone wanting a controlled environment. The more I learn about computer science the more I see the importance of tools such as containers. Tools help make sense of the world around us. They do this by measuring and creating a controlled environment. As for the IT professionals, containers can be used in a development setting to isolate issues with applications or to research an implementation before trying it on a live server. After completion of this lab, I have a much better understanding of the use of containers and see why knowing how to set up containers can be a great in the arsenal of tools for the IT professional.