Caleb Javier

Professor Holden-Gouveia

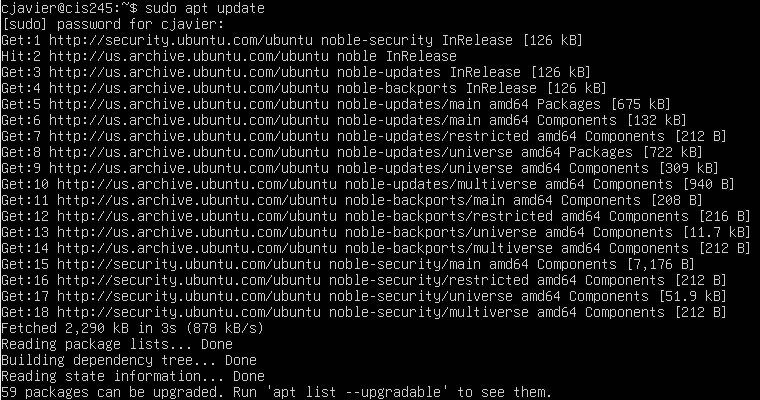
CIS-245

3 November 2024

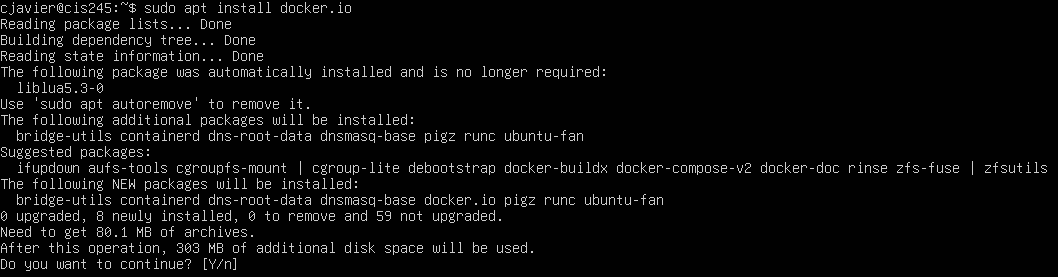
Assignment 13: Containers

1. **Installing container software for Ubuntu - Docker and Kubernetes**
   1. **Docker Installation**

Before installing, ensure to perform an update to the database by using **sudo apt update**.



Then, begin to install docker by using **sudo apt install docker.io**.



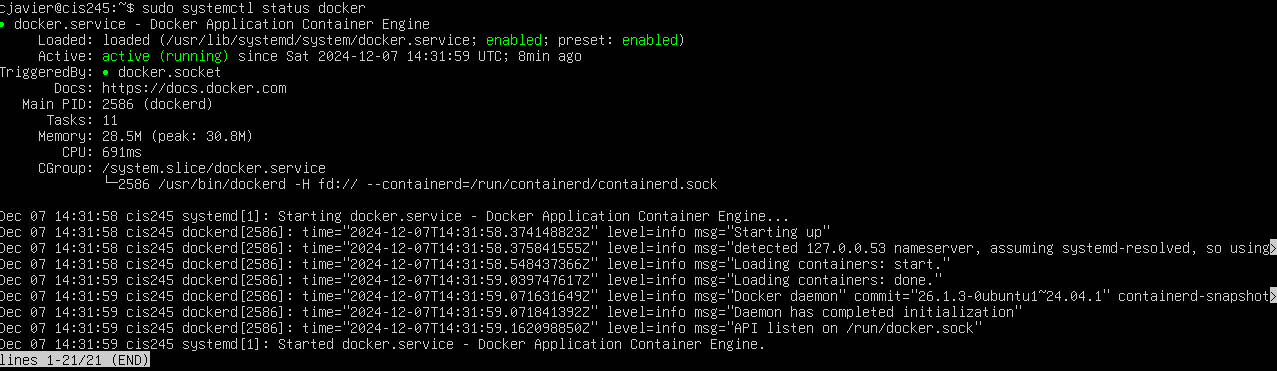
After being prompted with a continue statement, type y and press enter. Docker will begin to install. The process should look like the above photo.



TO efficiently use containers, Docker will be set to boot along with the system by using the following command:  


Then, to verify that the service is properly running, check the status by using

**sudo systemctl status docker**.



If it appears active and running, all has been done correctly. Use **Ctrl + C** to exit this interface. If it is not running, use

**sudo systemctl start docker** which should run the service. Docker has successfully been installed.

* 1. **MicroK8s (Kubernetes) Installation**

Before installing, begin by setting up snaps. The snap daemon is a necessary software for installing snap app packages, such as MicroK8s. Snap should be installed by default, but to ensure that it is installed, it can be tested by installing the “hello-world” snap using

**sudo snap install hello-world.**

If successful, the terminal should appear like this:



Which can then be tested, like so:



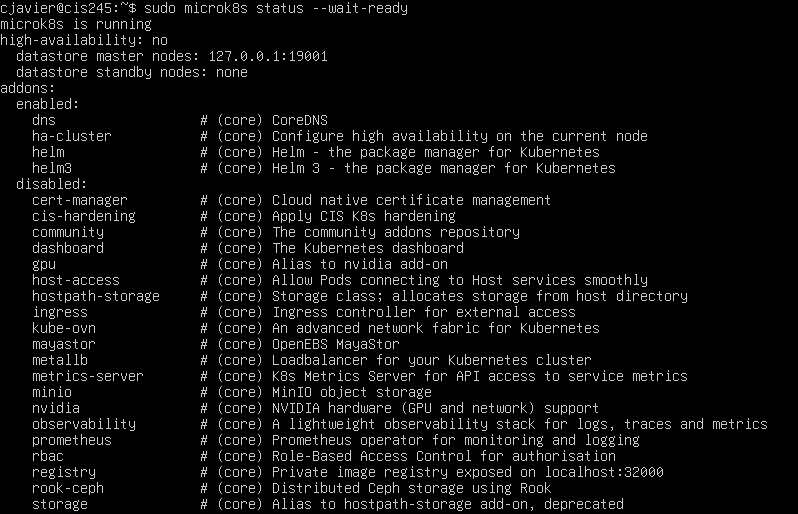
If errors are encountered, Snap can be installed with **sudo apt install snapd** (after updating the package database).

Now, to install MicroK8s, use **sudo snap install microk8s --classic.**

A successful installation will appear as the prior depiction above:



Then, the status of the service can be checked by using  **sudo microk8s status --wait-ready**

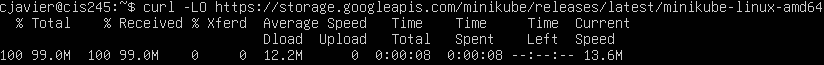
****

It should appear as running and display all addons.

1. **Minikube**

To begin, install minikube. To install the latest minikube stable release on x86-64 Linux using binary download (assumed system):

**curl -LO** [**https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64**](https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64)

****

Then

**sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64.**

****

Before continuing, set Docker as minikube’s default driver by using  **minikube config set driver docker.**

****

Next, to prevent permission errors, add the logged in user to the docker group by using:

**sudo usermod -aG docker $USER && newgrp docker.**



Then, create a cluster by using **minikube start.** Be sure not to have superuser permissions, but instead simply admin. After this, create a pod by using **kubectl create (after installing kubectl).**

1. **Containers - Pros and Cons**

Containers are very advantageous virtualizations of applications that, hosted by a single operating system, can be run in dozens of isolated instances that securely handle the information and contents (libraries, dependencies, scripts, etc.) only necessary for them to be run. By these properties, administrators are able to limit the resources consumed by containers which gives them much more control over how much demand the hardware that the containers are hosted on is subject to. The Docker files that are used to store the information contain only that information and nothing else, making them highly portable compared to applications installed locally on a single machine’s operating system and lightweight. Compared to using virtual machines on their own, containers are exponentially faster and simpler to run, requiring much less power, while being able to run on virtually any environment thanks to the Docker file. The convenience makes for an essentially configuration-less setup when necessary to run on environments separate from the container’s server hardware. I believe that the fact that containers are required to share the underlying operating system between them and their host is not a drawback, as they can also be used on servers hosting virtual machines. What this makes easy is the ability to moderate redundancy—multiple instances of important containers can be run on backup machines while on the same server, in the event that any of them crash, which was the initial reason for the above fact to be a disadvantage.

Works Cited

Brown, Nigel. “Containers vs VMs: What’s the Difference?” *YouTube*, IBM Technology, www.youtube.com/watch?v=cjXI-yxqGTI. Accessed 7 Dec. 2024.

“Dockerfile Overview.” *Docker Documentation*, 10 Sept. 2024, docs.docker.com/build/concepts/dockerfile/#:~:text=A%20Dockerfile%20is%20a%20text,reference%20in%20the%20Dockerfile%20reference.

“Virtual Machines vs Containers.” *YouTube*, PowerCert Animated Videos, www.youtube.com/watch?v=eyNBf1sqdBQ&t=291s. Accessed 7 Dec. 2024.