Jason Hodge

Project – Kupernetes Implementation

May 4, 2024

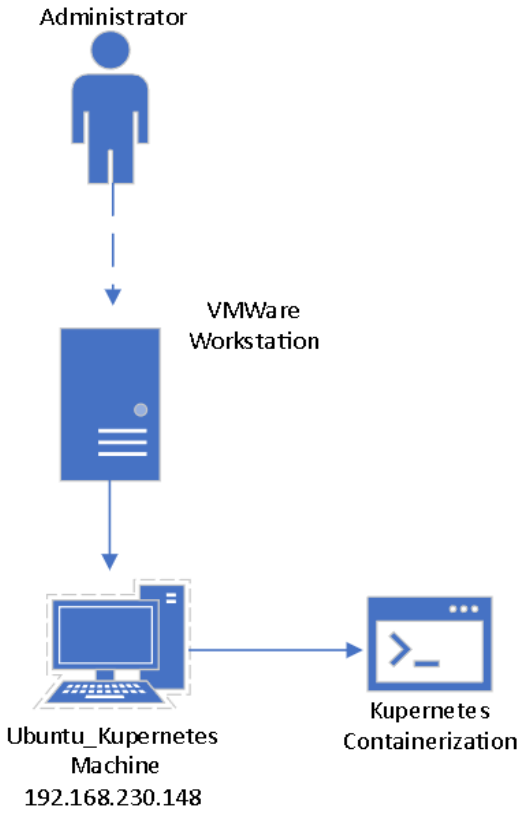
Description:

The primary objective of my project choice was to learn about and utilize Kupernetes for container orchestration and management. I thought this would be useful to become familiar with because of how widely used it is across companies today. Without the proper setup it is difficult to use if you do not have the proper accounts and utilize it often. I soon found this out once I got to the part when I needed to create a cluster utilizing a third-party application to host it. First, I utilized Fast API to create a simple Hello World Python application. I then containerized this application, wrapped it, and pushed it to Docker hub. This next part I needed to create a cluster where my application can react with and utilize the resources of a third-party service. Then you can implement and configure Kupernetes based resources such as authentication privileges, DNS configurations, other changes to the yaml files, and more.

Benefits of Implementing Kupernetes:

Kubernetes saves companies money and resources by managing containerized applications. Utilizing this not only improves resource efficiency, but also improves scalability, load balancing, storage and container orchestration, availability of different services, portability, self-healing capabilities, and support for implementing DevOps, Cloud, and DevSecOps practices. Implementing this is extremely beneficial to a company because not only does it provide all the previously mentioned benefits, but it makes building and running applications simpler and cost effective.

Topology:



Key Syntax:

sudo apt-get update: updates the software packages.

pip freeze: view the current versions of packages installed.

Other Commands are explained throughout as they are used.

Verification:

```
jason@ubuntu:~$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password)...
[sudo] password for jason:
==> This script will install:
/home/linuxbrew/.linuxbrew/bin/brew
/home/linuxbrew/.linuxbrew/share/doc/homebrew
/home/linuxbrew/.linuxbrew/share/man/man1/brew.1
/home/linuxbrew/.linuxbrew/share/zsh/site-functions/_brew
/home/linuxbrew/.linuxbrew/etc/bash_completion.d/brew
/home/linuxbrew/.linuxbrew/Homebrew
==> The following new directories will be created:
/home/linuxbrew/.linuxbrew/bin
/home/linuxbrew/.linuxbrew/etc
/home/linuxbrew/.linuxbrew/include
/home/linuxbrew/.linuxbrew/lib
/home/linuxbrew/.linuxbrew/sbin
/home/linuxbrew/.linuxbrew/share
/home/linuxbrew/.linuxbrew/var
/home/linuxbrew/.linuxbrew/opt
/home/linuxbrew/.linuxbrew/share/zsh
/home/linuxbrew/.linuxbrew/share/zsh/site-functions
/home/linuxbrew/.linuxbrew/var/homebrew
/home/linuxbrew/.linuxbrew/var/homebrew/linked
/home/linuxbrew/.linuxbrew/Cellar
/home/linuxbrew/.linuxbrew/Caskroom
/home/linuxbrew/.linuxbrew/Frameworks

Press RETURN/ENTER to continue or any other key to abort:
==> /usr/bin/sudo /usr/bin/install -d -o jason -g jason -m 0755 /home/linuxbrew/.linuxbrew
==> /usr/bin/sudo /bin/mkdir -p /home/linuxbrew/.linuxbrew/bin /home/linuxbrew/.linuxbrew/etc /home/linuxbrew/.linuxbrew/include /home/linuxbrew/.linuxbrew/lib /
home/linuxbrew/.linuxbrew/sbin /home/linuxbrew/.linuxbrew/share /home/linuxbrew/.linuxbrew/var /home/linuxbrew/.linuxbrew/opt /home/linuxbrew/.linuxbrew/share/zs
h /home/linuxbrew/.linuxbrew/share/zsh/site-functions /home/linuxbrew/.linuxbrew/var/homebrew /home/linuxbrew/.linuxbrew/var/homebrew/linked /home/linuxbrew/.lin
uxbrew/Cellar /home/linuxbrew/.linuxbrew/Caskroom /home/linuxbrew/.linuxbrew/Frameworks
==> /usr/bin/sudo /bin/chmod ug=rwx /home/linuxbrew/.linuxbrew/bin /home/linuxbrew/.linuxbrew/etc /home/linuxbrew/.linuxbrew/include /home/linuxbrew/.linuxbrew/l
ib /home/linuxbrew/.linuxbrew/sbin /home/linuxbrew/.linuxbrew/share /home/linuxbrew/.linuxbrew/var /home/linuxbrew/.linuxbrew/opt /home/linuxbrew/.linuxbrew/shar
e/zsh /home/linuxbrew/.linuxbrew/share/zsh/site-functions /home/linuxbrew/.linuxbrew/var/homebrew /home/linuxbrew/.linuxbrew/var/homebrew/linked /home/linuxbrew/
.linuxbrew/Cellar /home/linuxbrew/.linuxbrew/Caskroom /home/linuxbrew/.linuxbrew/Frameworks
==> /usr/bin/sudo /bin/chmod go-w /home/linuxbrew/.linuxbrew/share/zsh /home/linuxbrew/.linuxbrew/share/zsh/site-functions
==> /usr/bin/sudo /bin/chown jason /home/linuxbrew/.linuxbrew/bin /home/linuxbrew/.linuxbrew/etc /home/linuxbrew/.linuxbrew/include /home/linuxbrew/.linuxbrew/li
b /home/linuxbrew/.linuxbrew/sbin /home/linuxbrew/.linuxbrew/share /home/linuxbrew/.linuxbrew/var /home/linuxbrew/.linuxbrew/opt /home/linuxbrew/.linuxbrew/share
/zsh /home/linuxbrew/.linuxbrew/share/zsh/site-functions /home/linuxbrew/.linuxbrew/var/homebrew /home/linuxbrew/.linuxbrew/var/homebrew/linked /home/linuxbrew/.
linuxbrew/Cellar /home/linuxbrew/.linuxbrew/Caskroom /home/linuxbrew/.linuxbrew/Frameworks
==> /usr/bin/sudo /bin/chgrp jason /home/linuxbrew/.linuxbrew/bin /home/linuxbrew/.linuxbrew/etc /home/linuxbrew/.linuxbrew/include /home/linuxbrew/.linuxbrew/li
b /home/linuxbrew/.linuxbrew/sbin /home/linuxbrew/.linuxbrew/share /home/linuxbrew/.linuxbrew/var /home/linuxbrew/.linuxbrew/opt /home/linuxbrew/.linuxbrew/share
/zsh /home/linuxbrew/.linuxbrew/share/zsh/site-functions /home/linuxbrew/.linuxbrew/var/homebrew /home/linuxbrew/.linuxbrew/var/homebrew/linked /home/linuxbrew/.
linuxbrew/Cellar /home/linuxbrew/.linuxbrew/Caskroom /home/linuxbrew/.linuxbrew/Frameworks
==> /usr/bin/sudo /bin/mkdir -p /home/linuxbrew/.linuxbrew/Homebrew
==> /usr/bin/sudo /bin/chown -R jason:jason /home/linuxbrew/.linuxbrew/Homebrew
  You must install Git before installing Homebrew. See:
    https://docs.brew.sh/Installation
```

First, I chose the path of installing the proper Kupernetes dependencies(kubectl) with Homebrew. So, I installed Homebrew.

```
jason@ubuntu:~$ sudo apt-get install git
[sudo] password for jason:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 40 not upgraded.
Need to get 3,939 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 liberror-perl all 0.17-1.2 [19.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git-man all 1:2.7.4-0ubuntu1.10 [737 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git amd64 1:2.7.4-0ubuntu1.10 [3,183 kB]
Fetched 3,939 kB in 0s (7,593 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 212813 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17-1.2_all.deb ...
Unpacking liberror-perl (0.17-1.2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.7.4-0ubuntu1.10_all.deb ...
Unpacking git-man (1:2.7.4-0ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.10_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.10) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up liberror-perl (0.17-1.2) ...
Setting up git-man (1:2.7.4-0ubuntu1.10) ...
Setting up git (1:2.7.4-0ubuntu1.10) ...
```

Here I installed the git dependency.

```
jason@ubuntu:~$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password)...
==> This script will install:
/home/linuxbrew/.linuxbrew/bin/brew
/home/linuxbrew/.linuxbrew/share/doc/homebrew
/home/linuxbrew/.linuxbrew/share/man/man1/brew.1
/home/linuxbrew/.linuxbrew/share/zsh/site-functions/_brew
/home/linuxbrew/.linuxbrew/etc/bash_completion.d/brew
/home/linuxbrew/.linuxbrew/Homebrew

Press RETURN/ENTER to continue or any other key to abort:
==> /usr/bin/sudo /bin/chown -R jason:jason /home/linuxbrew/.linuxbrew/Homebrew
==> Downloading and installing Homebrew...
remote: Enumerating objects: 267944, done.
remote: Counting objects: 100% (2908/2908), done.
remote: Compressing objects: 100% (1466/1466), done.
remote: Total 267944 (delta 1564), reused 2595 (delta 1314), pack-reused 265036
Receiving objects: 100% (267944/267944), 80.58 MiB | 10.99 MiB/s, done.
Resolving deltas: 100% (193036/193036), done.
From https://github.com/Homebrew/brew
 * [new branch]      allowed-taps -> origin/allowed-taps
 * [new branch]      bundle-install-euid -> origin/bundle-install-euid
 * [new branch]      dependabot/bundler/Library/Homebrew/json_schemer-2.2.1 -> origin/dependabot/bundler/Library/Homebrew/json_schemer-2.2.1
 * [new branch]      deps-filters -> origin/deps-filters
 * [new branch]      generate-cask-renames -> origin/generate-cask-renames
 * [new branch]      intel-runner-tag -> origin/intel-runner-tag
 * [new branch]      load-internal-cask-json-v3 -> origin/load-internal-cask-json-v3
 * [new branch]      long-build-queue -> origin/long-build-queue
 * [new branch]      master     -> origin/master
 * [new branch]      neon-proxy-5201 -> origin/neon-proxy-5201
 * [new branch]      tapioca-patch -> origin/tapioca-patch
 * [new tag]         0.1        -> 0.1
 * [new tag]         0.2        -> 0.2
 * [new tag]         0.3        -> 0.3
 * [new tag]         0.4        -> 0.4
 * [new tag]         0.5        -> 0.5
 * [new tag]         0.6        -> 0.6
 * [new tag]         0.7        -> 0.7
 * [new tag]         0.7.1      -> 0.7.1
 * [new tag]         0.8        -> 0.8
 * [new tag]         0.8.1      -> 0.8.1
 * [new tag]         0.9        -> 0.9
 * [new tag]         0.9.1      -> 0.9.1
 * [new tag]         0.9.2      -> 0.9.2
 * [new tag]         0.9.3      -> 0.9.3
 * [new tag]         0.9.4      -> 0.9.4
 * [new tag]         0.9.5      -> 0.9.5
 * [new tag]         0.9.8      -> 0.9.8
 * [new tag]         0.9.9      -> 0.9.9
 * [new tag]         1.0.0      -> 1.0.0
 * [new tag]         1.0.1      -> 1.0.1
 * [new tag]         1.0.2      -> 1.0.2
 * [new tag]         1.0.3      -> 1.0.3
 * [new tag]         1.0.4      -> 1.0.4
 * [new tag]         1.0.5      -> 1.0.5
 * [new tag]         1.0.6      -> 1.0.6
 * [new tag]         1.0.7      -> 1.0.7
 * [new tag]         1.0.8      -> 1.0.8
 * [new tag]         1.0.9      -> 1.0.9
 * [new tag]         1.1.0      -> 1.1.0
 * [new tag]         1.1.1      -> 1.1.1
 * [new tag]         1.1.10     -> 1.1.10
 * [new tag]         1.1.11     -> 1.1.11
 * [new tag]         1.1.12     -> 1.1.12
 * [new tag]         1.1.13     -> 1.1.13
```

```
 * [new tag]         4.1.4      -> 4.1.4
 * [new tag]         4.1.5      -> 4.1.5
 * [new tag]         4.1.6      -> 4.1.6
 * [new tag]         4.1.7      -> 4.1.7
 * [new tag]         4.1.8      -> 4.1.8
 * [new tag]         4.2.0      -> 4.2.0
 * [new tag]         4.2.1      -> 4.2.1
 * [new tag]         4.2.10     -> 4.2.10
 * [new tag]         4.2.11     -> 4.2.11
 * [new tag]         4.2.12     -> 4.2.12
 * [new tag]         4.2.13     -> 4.2.13
 * [new tag]         4.2.15     -> 4.2.15
 * [new tag]         4.2.16     -> 4.2.16
 * [new tag]         4.2.17     -> 4.2.17
 * [new tag]         4.2.18     -> 4.2.18
 * [new tag]         4.2.19     -> 4.2.19
 * [new tag]         4.2.2      -> 4.2.2
 * [new tag]         4.2.20     -> 4.2.20
 * [new tag]         4.2.3      -> 4.2.3
 * [new tag]         4.2.4      -> 4.2.4
 * [new tag]         4.2.5      -> 4.2.5
 * [new tag]         4.2.6      -> 4.2.6
 * [new tag]         4.2.7      -> 4.2.7
 * [new tag]         4.2.8      -> 4.2.8
 * [new tag]         4.2.9      -> 4.2.9
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (20/20), done.
remote: Total 33 (delta 20), reused 20 (delta 20), pack-reused 13
Unpacking objects: 100% (33/33), done.
From https://github.com/Homebrew/brew
 * [new tag]         4.0.29     -> 4.0.29
 * [new tag]         4.1.9      -> 4.1.9
 * [new tag]         4.2.14     -> 4.2.14
Switched to a new branch 'stable'
==> Updating Homebrew...
==> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:f7be167f7ac4f296b9f4c5874ceeea4aafd9999c3c7f2b0378cae7dd273e2322
######################################################################## 100.0%
==> Pouring portable-ruby-3.1.4.x86_64_linux.bottle.tar.gz
Warning: /home/linuxbrew/.linuxbrew/bin is not in your PATH.
  Instructions on how to configure your shell for Homebrew
  can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    (echo; echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"') >> /home/jason/.bashrc
    eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"
- Install Homebrew's dependencies if you have sudo access:
    sudo apt-get install build-essential
  For more information, see:
    https://docs.brew.sh/Homebrew-on-Linux
- We recommend that you install GCC:
    brew install gcc
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh
```

I then tried the install again with git downloaded.

```
jason@ubuntu:~$ (echo; echo 'eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"') >> /home/jason/.bashrc
jason@ubuntu:~$ eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"
jason@ubuntu:~$ test -d ~/.linuxbrew && eval "$(~/.linuxbrew/bin/brew shellenv)"
jason@ubuntu:~$ brew install hello
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Downloading https://ghcr.io/v2/homebrew/core/hello/manifests/2.12.1
######################################################################################################################################################## 100.0%
==> Fetching dependencies for hello: linux-headers@5.15, glibc, gmp, isl, mpfr, libmpc, lz4, xz, zlib, zstd, binutils and gcc
==> Downloading https://ghcr.io/v2/homebrew/core/linux-headers/5.15/manifests/5.15.158
######################################################################################################################################################## 100.0%
==> Fetching linux-headers@5.15
==> Downloading https://ghcr.io/v2/homebrew/core/linux-headers/5.15/blobs/sha256:3264f378329da7bfe0c6a2a07920327bbdbcfcec74f2beb211dbbf3787d25eca
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/glibc/manifests/2.35_1
######################################################################################################################################################## 100.0%
==> Fetching glibc
==> Downloading https://ghcr.io/v2/homebrew/core/glibc/blobs/sha256:274dd06ae6ecaee3025d6bf21cf4c7641df9a1cc3973e162911a1f4a76000a24
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/gmp/manifests/6.3.0
######################################################################################################################################################## 100.0%
==> Installing gmp dependency: glibc
==> Downloading https://ghcr.io/v2/homebrew/core/glibc/manifests/2.35_1
Already downloaded: /home/jason/.cache/Homebrew/downloads/0b9d36176fd9b681842d23337db0fd45753c857707574090ef783dc4adb0a44f--glibc-2.35_1.bottle_manifest.json
==> Installing dependencies for glibc: linux-headers@5.15
==> Installing glibc dependency: linux-headers@5.15
==> Downloading https://ghcr.io/v2/homebrew/core/linux-headers/5.15/manifests/5.15.158
Already downloaded: /home/jason/.cache/Homebrew/downloads/c7087a24668dcf742cfcc1ea9cb344c28a970ca888ecc4c1a502c6a98906a569--linux-headers@5.15-5.15.158.bottle_ma
nifest.json
==> Pouring linux-headers@5.15--5.15.158.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/linux-headers@5.15/5.15.158: 960 files, 5.7MB
==> Installing glibc
==> Pouring glibc--2.35_1.x86_64_linux.bottle.tar.gz
==> /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1/sbin/ldconfig
==> Installing locale data for en_US.UTF-8
==> /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1/bin/localedef -i en_US -f UTF-8 en_US.UTF-8
🍺  /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1: 1,404 files, 47MB
==> Fetching gmp
==> Downloading https://ghcr.io/v2/homebrew/core/gmp/blobs/sha256:3dca3544faca889c7389a5fdbd2b5b00582c34a4e14607033573ad3b06ca7882
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/isl/manifests/0.26
######################################################################################################################################################## 100.0%
==> Fetching isl
==> Downloading https://ghcr.io/v2/homebrew/core/isl/blobs/sha256:db14ba1e4ea23ab41e06930dcf25ae9023c5e395c80602da2a9b6a98d54c92d3
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/mpfr/manifests/4.2.1
######################################################################################################################################################## 100.0%
==> Fetching mpfr
==> Downloading https://ghcr.io/v2/homebrew/core/mpfr/blobs/sha256:18857bac44d9f49faeb1d147146ba7fb420d5bf85076f69c68a86a563b203c13
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libmpc/manifests/1.3.1
######################################################################################################################################################## 100.0%
==> Fetching libmpc
==> Downloading https://ghcr.io/v2/homebrew/core/libmpc/blobs/sha256:f6542ae5bcf643ca0c980c7000cde1585922e76be080b3cc3422dac0d4a50904
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/manifests/1.9.4
######################################################################################################################################################## 100.0%
==> Fetching lz4
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/blobs/sha256:1757fefc3840e11c4822e4c2a95aa62aca44a4eaccce6f5c414ea51d1e58bf8e
######################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.4.6
######################################################################################################################################################## 100.0%
==> Fetching xz
==> Downloading https://ghcr.io/v2/homebrew/core/xz/blobs/sha256:0736983b952c5273bb5a345008bac7311c2f4b60758d69cc05495d5b050f88f1
######################################################################################################################################################## 100.0%
```

```
==> /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1/sbin/ldconfig
==> Installing locale data for en_US.UTF-8
==> /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1/bin/localedef -i en_US -f UTF-8 en_US.UTF-8
🍺  /home/linuxbrew/.linuxbrew/Cellar/glibc/2.35_1: 1,404 files, 47MB
==> Fetching hello
==> Downloading https://ghcr.io/v2/homebrew/core/hello/blobs/sha256:7935d0efdae69742f5140d514ef2e3e50d1d7cb82104cf6033ad51b900c12749
######################################################################################################################################################## 100.0%
==> Installing dependencies for hello: gmp, isl, mpfr, libmpc, lz4, xz, zlib, zstd, binutils and gcc
==> Installing hello dependency: gmp
==> Downloading https://ghcr.io/v2/homebrew/core/gmp/manifests/6.3.0
Already downloaded: /home/jason/.cache/Homebrew/downloads/70a72a71216843d66a953c06ff6337445ce9bc94fae9f0e301e2f59005274a8e--gmp-6.3.0.bottle_manifest.json
==> Pouring gmp--6.3.0.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/gmp/6.3.0: 23 files, 3.9MB
==> Installing hello dependency: isl
==> Downloading https://ghcr.io/v2/homebrew/core/isl/manifests/0.26
Already downloaded: /home/jason/.cache/Homebrew/downloads/ec0bbec77171645273dd59a9ccfdf290deccd999b3ca2082e0478d282b777ec0--isl-0.26.bottle_manifest.json
==> Pouring isl--0.26.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/isl/0.26: 74 files, 9.8MB
==> Installing hello dependency: mpfr
==> Downloading https://ghcr.io/v2/homebrew/core/mpfr/manifests/4.2.1
Already downloaded: /home/jason/.cache/Homebrew/downloads/a2a3424f4974f6febfa0334a93f35f508eaef3f4ad04320f73d9498302295635--mpfr-4.2.1.bottle_manifest.json
==> Pouring mpfr--4.2.1.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/mpfr/4.2.1: 31 files, 3.9MB
==> Installing hello dependency: libmpc
==> Downloading https://ghcr.io/v2/homebrew/core/libmpc/manifests/1.3.1
Already downloaded: /home/jason/.cache/Homebrew/downloads/fdfa98e0f8bb3ce075cb32776ac2345aa2f89252706c162aecfc841085fa76be--libmpc-1.3.1.bottle_manifest.json
==> Pouring libmpc--1.3.1.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/libmpc/1.3.1: 13 files, 638.7KB
==> Installing hello dependency: lz4
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/manifests/1.9.4
Already downloaded: /home/jason/.cache/Homebrew/downloads/379e59b981667f9585b33a2ff318769d8edca3ce6fd2e9a67ed291ae3e0cc872--lz4-1.9.4.bottle_manifest.json
==> Pouring lz4--1.9.4.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/lz4/1.9.4: 22 files, 695.6KB
==> Installing hello dependency: xz
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.4.6
Already downloaded: /home/jason/.cache/Homebrew/downloads/b2cc4077807c100af6e0253f51d186f187ff55165638cbe3a4aa16d1c4762660--xz-5.4.6.bottle_manifest.json
==> Pouring xz--5.4.6.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/xz/5.4.6: 164 files, 2.7MB
==> Installing hello dependency: zlib
==> Downloading https://ghcr.io/v2/homebrew/core/zlib/manifests/1.3.1
Already downloaded: /home/jason/.cache/Homebrew/downloads/f68d0caf232d52f2aa586abefbbfd7e958e384d84f3967008fa83de94b5f10ae--zlib-1.3.1.bottle_manifest.json
==> Pouring zlib--1.3.1.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/zlib/1.3.1: 13 files, 473.8KB
==> Installing hello dependency: zstd
==> Downloading https://ghcr.io/v2/homebrew/core/zstd/manifests/1.5.6
Already downloaded: /home/jason/.cache/Homebrew/downloads/29403e0df5404d8aeca0e750ac135ec9ef44fc5eeb6df69170ed602acabf0ffb--zstd-1.5.6.bottle_manifest.json
==> Pouring zstd--1.5.6.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/zstd/1.5.6: 31 files, 2.9MB
==> Installing hello dependency: binutils
==> Downloading https://ghcr.io/v2/homebrew/core/binutils/manifests/2.42
Already downloaded: /home/jason/.cache/Homebrew/downloads/21c83d367c6459449acbeea36db5c99baba65353c7f09b8cfe34a98977783291--binutils-2.42.bottle_manifest.json
==> Pouring binutils--2.42.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/binutils/2.42: 4,796 files, 490.5MB
==> Installing hello dependency: gcc
==> Downloading https://ghcr.io/v2/homebrew/core/gcc/manifests/13.2.0-2
Already downloaded: /home/jason/.cache/Homebrew/downloads/1d624841f5f550e4bbc04910c0cfb20b89d0fc7f90ab213cead29d1518479e2b--gcc-13.2.0-2.bottle_manifest.json
==> Pouring gcc--13.2.0.x86_64_linux.bottle.2.tar.gz
==> Creating the GCC specs file: /home/linuxbrew/.linuxbrew/Cellar/gcc/13.2.0/bin/../lib/gcc/current/gcc/x86_64-pc-linux-gnu/13/specs
🍺  /home/linuxbrew/.linuxbrew/Cellar/gcc/13.2.0: 1,668 files, 320.2MB
==> Installing hello
==> Pouring hello--2.12.1.x86_64_linux.bottle.tar.gz
🍺  /home/linuxbrew/.linuxbrew/Cellar/hello/2.12.1: 55 files, 651.8KB
==> Running `brew cleanup hello`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
jason@ubuntu:~$
```

Here I added Homebrew to the path and shell as well as tested functionality.

```
jason@ubuntu:~$ brew install kubectl
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/manifests/1.30.0
######################################################################################################################################################################## 100.0%
==> Fetching kubernetes-cli
==> Downloading https://ghcr.io/v2/homebrew/core/kubernetes-cli/blobs/sha256:3ef901a5228ee6c20db0c97a46936baf09174bfff220b27a8e534963fb7f1228
######################################################################################################################################################################## 100.0%
==> Pouring kubernetes-cli--1.30.0.x86_64_linux.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /home/linuxbrew/.linuxbrew/etc/bash_completion.d
==> Summary
🍺  /home/linuxbrew/.linuxbrew/Cellar/kubernetes-cli/1.30.0: 235 files, 50MB
==> Running `brew cleanup kubernetes-cli`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
jason@ubuntu:~$ kubectl version --client
Client Version: v1.30.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
jason@ubuntu:~$
```

I then installed kubectl with Homebrew and checked the version.

```
jason@ubuntu:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 91.2M  100 91.2M    0     0  7371k      0  0:00:12  0:00:12 --:--:-- 10.9M
jason@ubuntu:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
[sudo] password for jason:
jason@ubuntu:~$ kubectl cluster-info
E0503 16:07:58.877978   61460 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080:
connect: connection refused
E0503 16:07:58.878509   61460 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080:
connect: connection refused
E0503 16:07:58.880531   61460 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080:
connect: connection refused
E0503 16:07:58.880685   61460 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080:
connect: connection refused
E0503 16:07:58.881965   61460 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080:
connect: connection refused

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
The connection to the server localhost:8080 was refused - did you specify the right host or port?
jason@ubuntu:~$ minikube start
😄  minikube v1.33.0 on Ubuntu 16.04
✨  Unable to pick a default driver. Here is what was considered, in preference order:
💡  Alternatively you could install one of these drivers:
    ▪ docker: Not installed: exec: "docker": executable file not found in $PATH
    ▪ kvm2: Not installed: exec: "virsh": executable file not found in $PATH
    ▪ podman: Not installed: exec: "podman": executable file not found in $PATH
    ▪ qemu2: Not installed: exec: "qemu-system-x86_64": executable file not found in $PATH
    ▪ virtualbox: Not installed: unable to find VBoxManage in $PATH

❌  Exiting due to DRV_NOT_DETECTED: No possible driver was detected. Try specifying --driver, or see https://minikube.sigs.k8s.io/docs/start/
```

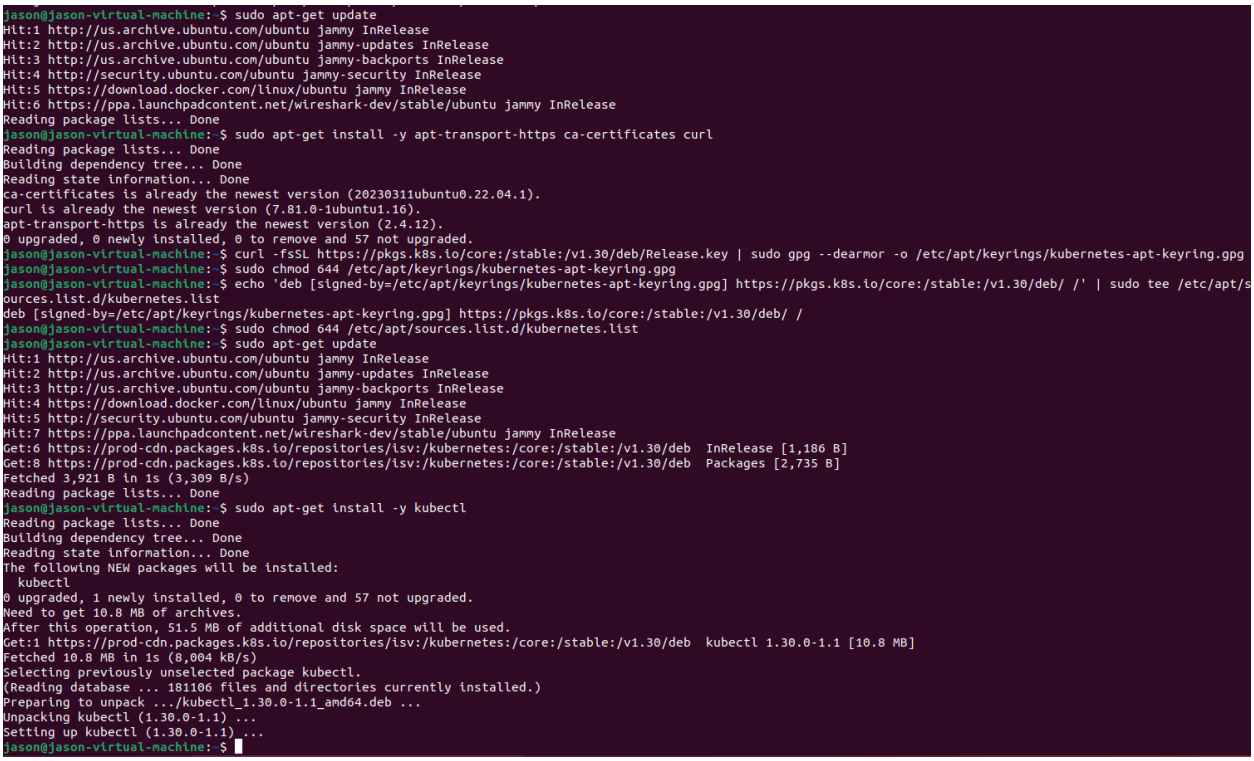
Here I checked the cluster info, worked on downloading minikube to help with the setup of Kupernetes clusters.

```
jason@jason-virtual-machine:~$ sudo apt-get update
[sudo] password for jason:
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [31.5 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,394 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [453 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [243 kB]
Get:10 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy InRelease [24.4 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [848 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [601 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [163 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,612 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [619 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [304 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [311 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [701 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,072 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [245 kB]
Get:21 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy/main i386 Packages [1,156 B]
Get:22 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy/main amd64 Packages [4,288 B]
Fetched 9,014 kB in 2s (3,676 kB/s)
Reading package lists... Done
```

Here I ran updates after what I previously installed.

```
jason@jason-virtual-machine:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be upgraded:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-compose-plugin
5 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
Need to get 112 MB of archives.
After this operation, 5,413 kB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.31-1 [29.8 MB]
Get:2 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.14.0-1~ubuntu.22.04~jammy [29.7 MB]
Get:3 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:26.1.1-1~ubuntu.22.04~jammy [14.6 MB]
Get:4 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:26.1.1-1~ubuntu.22.04~jammy [25.3 MB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.27.0-1~ubuntu.22.04~jammy [12.5 MB]
Fetched 112 MB in 11s (10.1 MB/s)
(Reading database ... 181106 files and directories currently installed.)
Preparing to unpack .../containerd.io_1.6.31-1_amd64.deb ...
Unpacking containerd.io (1.6.31-1) over (1.6.28-2) ...
Preparing to unpack .../docker-buildx-plugin_0.14.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-buildx-plugin (0.14.0-1~ubuntu.22.04~jammy) over (0.13.1-1~ubuntu.22.04~jammy) ...
Preparing to unpack .../docker-ce-cli_5%3a26.1.1-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce-cli (5:26.1.1-1~ubuntu.22.04~jammy) over (5:26.0.0-1~ubuntu.22.04~jammy) ...
Preparing to unpack .../docker-ce_5%3a26.1.1-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce (5:26.1.1-1~ubuntu.22.04~jammy) over (5:26.0.0-1~ubuntu.22.04~jammy) ...
Preparing to unpack .../docker-compose-plugin_2.27.0-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.27.0-1~ubuntu.22.04~jammy) over (2.25.0-1~ubuntu.22.04~jammy) ...
Setting up docker-buildx-plugin (0.14.0-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.6.31-1) ...
Setting up docker-compose-plugin (2.27.0-1~ubuntu.22.04~jammy) ...
Setting up docker-ce-cli (5:26.1.1-1~ubuntu.22.04~jammy) ...
Setting up docker-ce (5:26.1.1-1~ubuntu.22.04~jammy) ...
Processing triggers for man-db (2.10.2-1) ...
jason@jason-virtual-machine:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

jason@jason-virtual-machine:~$
```

Here I installed Docker and ran hello-world to test functionality.



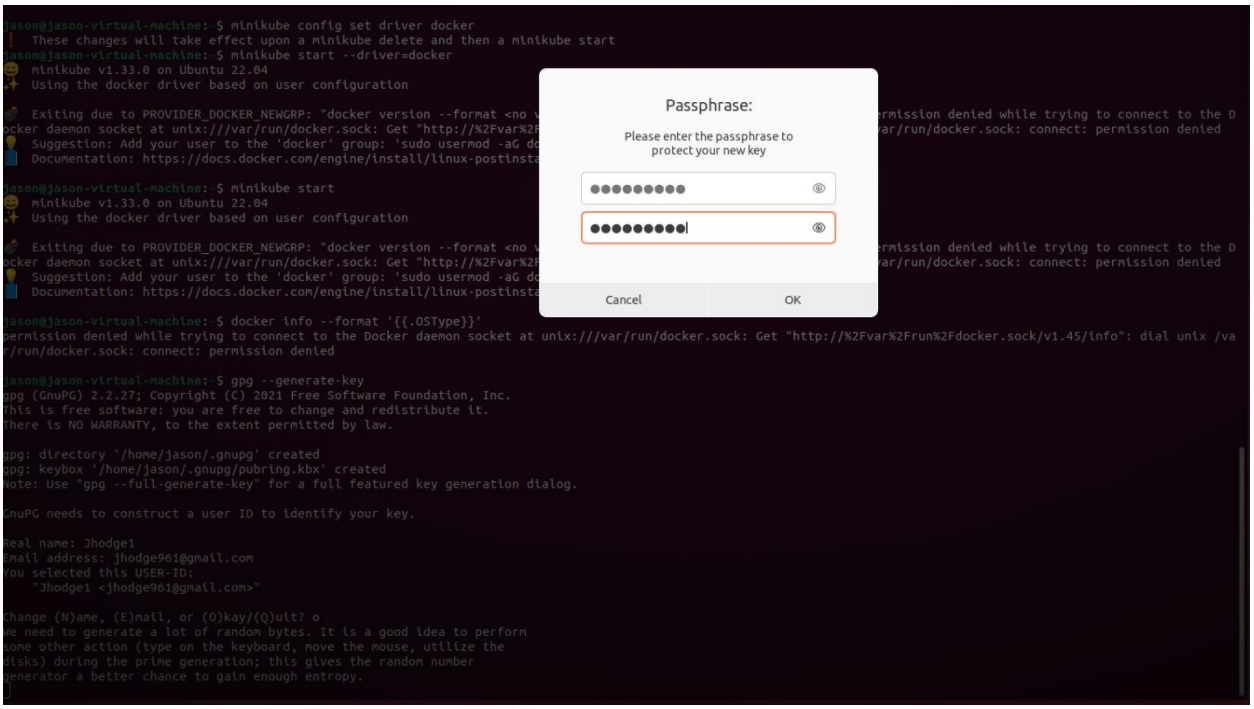
Here I ran updates, installed curl, and the proper Kupernetes dependencies a different way as I ran into issues using the Homebrew method.

```
jason@jason-virtual-machine:~$ kubectl cluster-info
/usr/local/bin/kubectl: line 1: syntax error near unexpected token `<'
/usr/local/bin/kubectl: line 1: `<?xml version='1.0' encoding='UTF-8'?><Error><Code>NoSuchKey</Code><Message>The specified key does not exist.</Message><Details>No such object: kubernetes-release/release/bin/linux/amd64/kubectl</Details></Error>'
jason@jason-virtual-machine:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 91.2M  100 91.2M    0     0  10.4M      0  0:00:08  0:00:08 --:--:-- 10.6M
jason@jason-virtual-machine:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
jason@jason-virtual-machine:~$ minikube start
😄  minikube v1.33.0 on Ubuntu 22.04
👎  Unable to pick a default driver. Here is what was considered, in preference order:
    ▪ docker: Not healthy: "docker version --format {{.Server.Os}}-{{.Server.Version}}:{{.Server.Platform.Name}}" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
    ▪ docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux-postinstall/>
💡  Alternatively you could install one of these drivers:
    ▪ kvm2: Not installed: exec: "virsh": executable file not found in $PATH
    ▪ qemu2: Not installed: exec: "qemu-system-x86_64": executable file not found in $PATH
    ▪ podman: Not installed: exec: "podman": executable file not found in $PATH
    ▪ virtualbox: Not installed: unable to find VBoxManage in $PATH

❌  Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

jason@jason-virtual-machine:~$ docker --version
Docker version 26.1.1, build 4cf5afa
jason@jason-virtual-machine:~$ minikube start --driver=docker
😄  minikube v1.33.0 on Ubuntu 22.04
✨  Using the docker driver based on user configuration

💣  Exiting due to PROVIDER_DOCKER_NEWGRP: "docker version --format <no value>-<no value>:<no value>" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
💡  Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
📘  Documentation: https://docs.docker.com/engine/install/linux-postinstall/

jason@jason-virtual-machine:~$ minikube config set driver docker
❗  These changes will take effect upon a minikube delete and then a minikube start
jason@jason-virtual-machine:~$ minikube start --driver=docker
😄  minikube v1.33.0 on Ubuntu 22.04
✨  Using the docker driver based on user configuration

💣  Exiting due to PROVIDER_DOCKER_NEWGRP: "docker version --format <no value>-<no value>:<no value>" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
💡  Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
📘  Documentation: https://docs.docker.com/engine/install/linux-postinstall/

jason@jason-virtual-machine:~$ minikube start
😄  minikube v1.33.0 on Ubuntu 22.04
✨  Using the docker driver based on user configuration

💣  Exiting due to PROVIDER_DOCKER_NEWGRP: "docker version --format <no value>-<no value>:<no value>" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
💡  Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
📘  Documentation: https://docs.docker.com/engine/install/linux-postinstall/
```

Here we can see I started minikube and configured the proper driver.

```
jason@jason-virtual-machine:~$ docker info --format '{{.OSType}}'
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/info": dial unix /var/run/docker.sock: connect: permission denied

jason@jason-virtual-machine:~$ gpg --generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/jason/.gnupg' created
gpg: keybox '/home/jason/.gnupg/pubring.kbx' created
Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Jhodge1
Email address: jhodge961@gmail.com
You selected this USER-ID:
    "Jhodge1 <jhodge961@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/jason/.gnupg/trustdb.gpg: trustdb created
gpg: key EBA1BDF66FE34BB1 marked as ultimately trusted
gpg: directory '/home/jason/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/jason/.gnupg/openpgp-revocs.d/67D53DECEB8753B93F42A6DFEBA1BDF66FE34BB1.rev'
public and secret key created and signed.

pub   rsa3072 2024-05-04 [SC] [expires: 2026-05-04]
      67D53DECEB8753B93F42A6DFEBA1BDF66FE34BB1
uid                      Jhodge1 <jhodge961@gmail.com>
sub   rsa3072 2024-05-04 [E] [expires: 2026-05-04]

jason@jason-virtual-machine:~$
```

Here I signed in with my docker to obtain a key.



Here I had to create a passphrase for my docker pub.

```
jason@jason-virtual-machine:~$ gpg --generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Jason
Email address: jhodge961@gmail.com
You selected this USER-ID:
    "Jason <jhodge961@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key CC775148AB8CF073 marked as ultimately trusted
gpg: revocation certificate stored as '/home/jason/.gnupg/openpgp-revocs.d/6E8B623B27A2851126C4D3F6CC775148AB8CF073.rev'
public and secret key created and signed.

pub   rsa3072 2024-05-04 [SC] [expires: 2026-05-04]
      6E8B623B27A2851126C4D3F6CC775148AB8CF073
uid                      Jason <jhodge961@gmail.com>
sub   rsa3072 2024-05-04 [E] [expires: 2026-05-04]

jason@jason-virtual-machine:~$
```

Here we can see the generated key. Note the key is different as I ran the command again upon returning to work on the lab.

```
jason@jason-virtual-machine:~$ pass init <6E8B623B27A2851126C4D3F6CC775148AB8CF073>
bash: syntax error near unexpected token `newline'
jason@jason-virtual-machine:~$ pass init 6E8B623B27A2851126C4D3F6CC775148AB8CF073
Command 'pass' not found, but can be installed with:
sudo apt install pass
jason@jason-virtual-machine:~$ sudo apt install pass
[sudo] password for jason:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libqrencode4 qrencode tree xclip
Suggested packages:
  libxml-simple-perl python ruby
The following NEW packages will be installed:
  libqrencode4 pass qrencode tree xclip
0 upgraded, 5 newly installed, 0 to remove and 57 not upgraded.
Need to get 151 kB of archives.
After this operation, 442 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libqrencode4 amd64 4.1.1-1 [24.0 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 tree amd64 2.0.2-1 [47.9 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 pass all 1.7.4-5 [35.2 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 qrencode amd64 4.1.1-1 [25.2 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 xclip amd64 0.13-2 [18.3 kB]
Fetched 151 kB in 0s (1,083 kB/s)
Selecting previously unselected package libqrencode4:amd64.
(Reading database ... 181110 files and directories currently installed.)
Preparing to unpack .../libqrencode4_4.1.1-1_amd64.deb ...
Unpacking libqrencode4:amd64 (4.1.1-1) ...
Selecting previously unselected package tree.
Preparing to unpack .../tree_2.0.2-1_amd64.deb ...
Unpacking tree (2.0.2-1) ...
Selecting previously unselected package pass.
Preparing to unpack .../archives/pass_1.7.4-5_all.deb ...
Unpacking pass (1.7.4-5) ...
Selecting previously unselected package qrencode.
Preparing to unpack .../qrencode_4.1.1-1_amd64.deb ...
Unpacking qrencode (4.1.1-1) ...
Selecting previously unselected package xclip.
Preparing to unpack .../xclip_0.13-2_amd64.deb ...
Unpacking xclip (0.13-2) ...
Setting up libqrencode4:amd64 (4.1.1-1) ...
Setting up qrencode (4.1.1-1) ...
Setting up tree (2.0.2-1) ...
Setting up xclip (0.13-2) ...
Setting up pass (1.7.4-5) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.7) ...
jason@jason-virtual-machine:~$
```

```
jason@jason-virtual-machine:~$ pass init 6E8B623B27A2851126C4D3F6CC775148AB8CF073
mkdir: created directory '/home/jason/.password-store/'
Password store initialized for 6E8B623B27A2851126C4D3F6CC775148AB8CF073
```

Here I created the directory linking my docker pub id.

```
jason@jason-virtual-machine:~$ curl -LO 'https://dl.k8s.io/release/v1.22.0/bin/linux/amd64/kubectl'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    641      0 --:--:-- --:--:-- --:--:--   641
100 44.7M  100 44.7M    0     0  3583k      0  0:00:12  0:00:12 --:--:-- 3899k
jason@jason-virtual-machine:~$ curl -LO "https://dl.k8s.io/bin/linux/amd64/kubectl.sha256"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0   1118      0 --:--:-- --:--:-- --:--:--  1121
100    64  100    64    0     0    147      0 --:--:-- --:--:-- --:--:--   147
jason@jason-virtual-machine:~$ echo "$(<kubectl.sha256) kubectl" | sha256sum --check
kubectl: OK
jason@jason-virtual-machine:~$ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[sudo] password for jason:
jason@jason-virtual-machine:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.0", GitCommit:"c2b5237ccd9c0f1d600d3072634ca66cefdf272f", GitTreeState:"clean", BuildDate:"2021-08-04T18:03:20Z", GoVersio
n:"go1.16.6", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
jason@jason-virtual-machine:~$ kubectl config view
apiVersion: v1
clusters: null
contexts: null
current-context: ""
kind: Config
preferences: {}
users: null
```

Here we can see kubectl is installed and working.

```
jason@jason-virtual-machine:~/Kupernetes_Test$ python3 -m venv ./venv
The virtual environment was not created successfully because ensurepip is not
available.  On Debian/Ubuntu systems, you need to install the python3-venv
package using the following command.

    apt install python3.10-venv

You may need to use sudo with that command.  After installing the python3-venv
package, recreate your virtual environment.

Failing command: /home/jason/Kupernetes Test/venv/bin/python3

jason@jason-virtual-machine:~/Kupernetes_Test$ apt install python3.10-venv
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
jason@jason-virtual-machine:~/Kupernetes_Test$ sudo -i
[sudo] password for jason:
root@jason-virtual-machine:~# apt install python3.10-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-distutils python3-lib2to3 python3-pip-whl python3-setuptools-whl
The following NEW packages will be installed:
  python3-distutils python3-lib2to3 python3-pip-whl python3-setuptools-whl python3.10-venv
0 upgraded, 5 newly installed, 0 to remove and 54 not upgraded.
Need to get 2,690 kB of archives.
After this operation, 4,056 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-lib2to3 all 3.10.8-1~22.04 [77.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-distutils all 3.10.8-1~22.04 [139 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-pip-whl all 22.0.2+dfsg-1ubuntu0.4 [1,680 kB
]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3-setuptools-whl all 59.6.0-1.2ubuntu0.22.04.1
 [788 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 python3.10-venv amd64 3.10.12-1~22.04.3 [5,716 B]
Fetched 2,690 kB in 0s (6,072 kB/s)
Selecting previously unselected package python3-lib2to3.
(Reading database ... 181172 files and directories currently installed.)
Preparing to unpack .../python3-lib2to3 3.10.8-1~22.04 all.deb ...
Unpacking python3-lib2to3 (3.10.8-1~22.04) ...
```

Utilizing VS code I downloaded python and created the folder venv.

```
Setting up python3.10-venv (3.10.12-1~22.04.3) ...
root@jason-virtual-machine:~# ls
linux_tweet_app  snap
root@jason-virtual-machine:~# python3 -m venv ./venv
root@jason-virtual-machine:~# ls
linux_tweet_app  snap  venv
root@jason-virtual-machine:~# source ./venv/bin/activate
(venv) root@jason-virtual-machine:~# pip install fastapi
Collecting fastapi
  Downloading fastapi-0.111.0-py3-none-any.whl (91 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.0/92.0 KB 3.3 MB/s eta 0:00:00
Collecting httpx>=0.23.0
  Downloading httpx-0.27.0-py3-none-any.whl (75 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.6/75.6 KB 19.0 MB/s eta 0:00:00
Collecting python-multipart>=0.0.7
  Downloading python multipart-0.0.9-py3-none-any.whl (22 kB)
Collecting ujson!=4.0.2,!=4.1.0,!=4.2.0,!=4.3.0,!=5.0.0,!=5.1.0,>=4.0.1
  Downloading ujson-5.9.0-cp310-cp310-manylinux 2 17 x86 64.manylinux2014 x86 64.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.2/53.2 KB 10.3 MB/s eta 0:00:00
Collecting email validator>=2.0.0
  Downloading email validator-2.1.1-py3-none-any.whl (30 kB)
Collecting jinja2>=2.11.2
  Downloading Jinja2-3.1.3-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.2/133.2 KB 9.6 MB/s eta 0:00:00
Collecting fastapi-cli>=0.0.2
  Downloading fastapi cli-0.0.2-py3-none-any.whl (9.1 kB)
Collecting starlette<0.38.0,>=0.37.2
  Downloading starlette-0.37.2-py3-none-any.whl (71 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.9/71.9 KB 9.8 MB/s eta 0:00:00
Collecting orjson>=3.2.1
  Downloading orjson-3.10.3-cp310-cp310-manylinux 2 17 x86 64.manylinux2014 x86 64.whl (142 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 142.5/142.5 KB 11.2 MB/s eta 0:00:00
Collecting typing-extensions>=4.8.0
  Downloading typing extensions-4.11.0-py3-none-any.whl (34 kB)
Collecting pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4
  Downloading pydantic-2.7.1-py3-none-any.whl (409 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 409.3/409.3 KB 16.4 MB/s eta 0:00:00
Collecting uvicorn[standard]>=0.12.0
  Downloading uvicorn-0.29.0-py3-none-any.whl (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.8/60.8 KB 10.9 MB/s eta 0:00:00
```

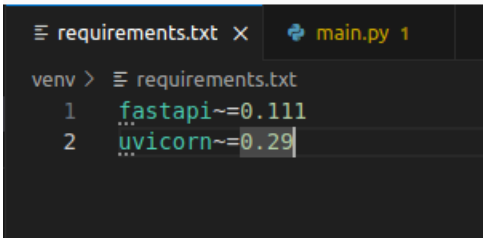
Here we can see the current containers on the machine. I also installed Fast API. This machine is an independent clone of the Ubuntu machine I used in Lab 4 with the linux_tweet_app, hence we see it.

```
(venv) root@jason-virtual-machine:~# pip install uvicorn
Requirement already satisfied: uvicorn in ./venv/lib/python3.10/site-packages (0.29.0)
Requirement already satisfied: typing-extensions>=4.0 in ./venv/lib/python3.10/site-packages (from uvicorn) (4.11.0)
Requirement already satisfied: click>=7.0 in ./venv/lib/python3.10/site-packages (from uvicorn) (8.1.7)
Requirement already satisfied: h11>=0.8 in ./venv/lib/python3.10/site-packages (from uvicorn) (0.14.0)
(venv) root@jason-virtual-machine:~#
```

Here I installed uvicorn, which is an Asynchronous Server Gateway Interface used to implement python web servers.

```
(venv) root@jason-virtual-machine:~# pip freeze
annotated-types==0.6.0
anyio==4.3.0
certifi==2024.2.2
click==8.1.7
dnspython==2.6.1
email validator==2.1.1
exceptiongroup==1.2.1
fastapi==0.111.0
fastapi-cli==0.0.2
h11==0.14.0
httpcore==1.0.5
httptools==0.6.1
httpx==0.27.0
idna==3.7
Jinja2==3.1.3
markdown-it-py==3.0.0
MarkupSafe==2.1.5
mdurl==0.1.2
orjson==3.10.3
pydantic==2.7.1
pydantic core==2.18.2
Pygments==2.18.0
python-dotenv==1.0.1
python-multipart==0.0.9
PyYAML==6.0.1
rich==13.7.1
shellingham==1.5.4
sniffio==1.3.1
starlette==0.37.2
typer==0.12.3
typing extensions==4.11.0
ujson==5.9.0
uvicorn==0.29.0
uvloop==0.19.0
watchfiles==0.21.0
websockets==12.0
```

Here I ran pip freeze to view the current versions of packages installed.

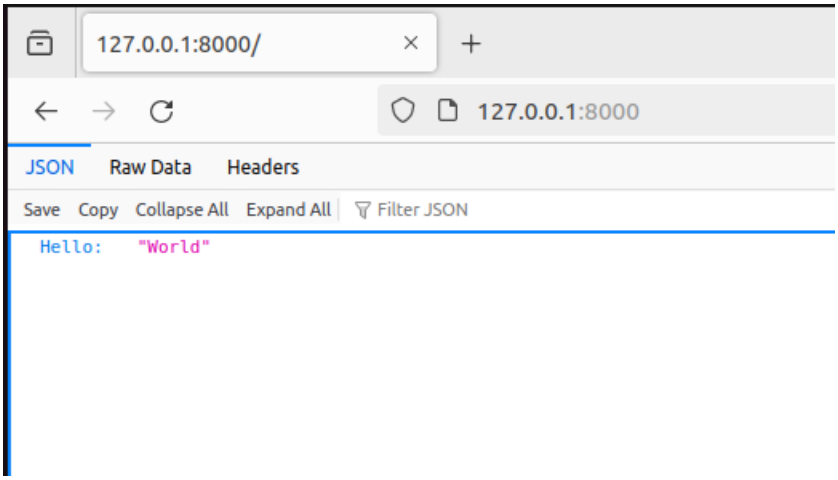


Here I added the only two packages needed for our web server to run to a text file called requirements.txt.

requirements.txt | main.py 1

app > main.py > ...

```python
from fastapi import FastAPI

app = FastAPI()


@app.get("/")
def read_root():
    return {"Hello": "World"}
```




```
jason@jason-virtual-machine:~$ uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['/home/jason']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [13137] using WatchFiles
ERROR:    Error loading ASGI app. Could not import module "main".
WARNING:  WatchFiles detected changes in 'Kupernetes_Test/app/main.py', 'Kupernetes_Test/main.py'. Reloading...
ERROR:    Error loading ASGI app. Could not import module "main".
WARNING:  WatchFiles detected changes in 'Kupernetes_Test/main.py', 'main.py'. Reloading...
INFO:     Started server process [14049]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:51652 - "GET / HTTP/1.1" 200 OK
INFO:     127.0.0.1:51652 - "GET /favicon.ico HTTP/1.1" 404 Not Found
```

```
^CINFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [14049]
INFO:     Stopping reloader process [13137]
jason@jason-virtual-machine:~$ cd
jason@jason-virtual-machine:~$ cd Kupernetes_Test
jason@jason-virtual-machine:~/Kupernetes_Test$ cd ..
jason@jason-virtual-machine:~$ cd Kupernetes_Test
jason@jason-virtual-machine:~/Kupernetes_Test$ code Dockerfile
```

Here we can see a basic Hello World that returns exactly that when we view it at either 127.0.0.1:8000 or localhost:8000. I then created a dockerfile to paste functions in.

This included:

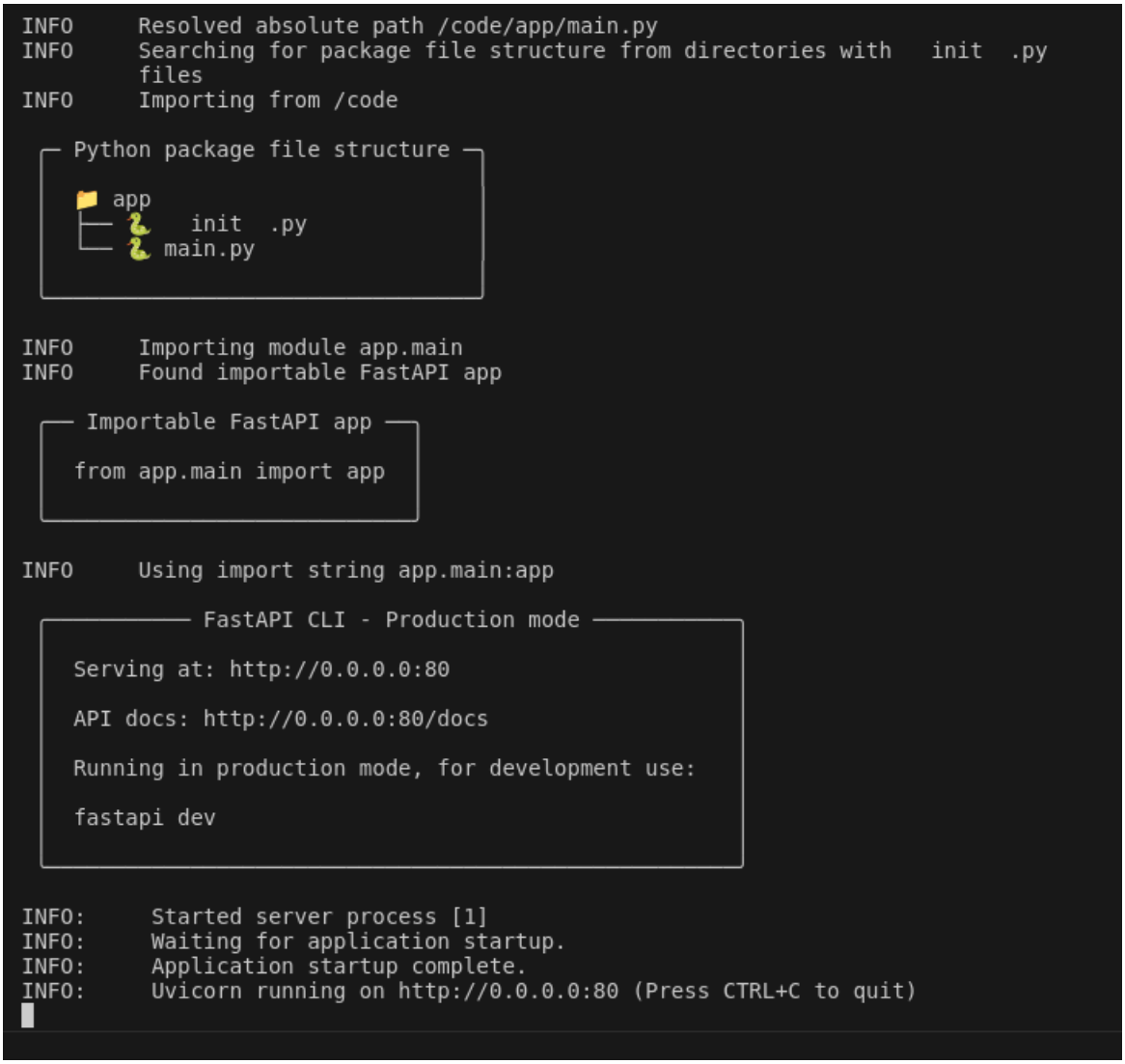
```
#
FROM python:3.9
#
WORKDIR /code
#
COPY ./requirements.txt /code/requirements.txt
#
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
#
COPY ./app /code/app
#
CMD ["fastapi", "run", "app/main.py", "--port", "80"]
```

-from FastAPI in Container - Docker

```
root@jason-virtual-machine:~# docker build /home/jason/Kupernetes Test
[+] Building 61.0s (10/10) FINISHED                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 262B                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                 0.2s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [1/5] FROM docker.io/library/python:3.9@sha256:5930d54604459569953a4164839ad9bb32f6a1c76e6740c84  47.8s
 => => resolve docker.io/library/python:3.9@sha256:5930d54604459569953a4164839ad9bb32f6a1c76e6740c84a  0.0s
 => => sha256:5930d54604459569953a4164839ad9bb32f6a1c76e6740c84a116b4290a57852 1.86kB / 1.86kB         0.0s
 => => sha256:1468e7ff95fcb865fbc4dee7094f8b99c4dcddd6eb2180cf044c7396baf6fc2f 49.58MB / 49.58MB      18.0s
 => => sha256:2cf9c2b42f41b1845f3e4421b723d56146db82939dc884555e077768e18132f4 24.05MB / 24.05MB       7.2s
 => => sha256:b81bfd63a766f385494a585e154465bb7178c820c4cd1e9cb6a8c3daa62433b7 2.01kB / 2.01kB         0.0s
 => => sha256:ab7eeae5d25f857a1eb6b021f9ef958a31724b3fe131449622078a24ac634eef 7.31kB / 7.31kB         0.0s
 => => sha256:c4c40c3e3cdf945721f480e1d939aac857876fdb5c33b8fbfcf655c63b0b9428 64.14MB / 64.14MB      16.3s
 => => sha256:c05cc1123d7e335d59b0f465c23b7ad2ad27f4875b6c3eab41c65a9b50efa382 211.18MB / 211.18MB    32.1s
 => => sha256:b6f29ccdcc551647511d3473f89c94b2ee7fbce3e65226908ea74cfc5c586697 6.39MB / 6.39MB        19.0s
 => => sha256:9c8be2164d2a0d2ef7cd16a93364ef0d7e579861778b0fa07071f8899d950aa2 15.82MB / 15.82MB      21.7s
 => => extracting sha256:1468e7ff95fcb865fbc4dee7094f8b99c4dcddd6eb2180cf044c7396baf6fc2f               6.5s
 => => sha256:d00b6068660376d1ea605bdb6a2bca31e9e3a05c10d50c48dd8a513824039360 244B / 244B            19.0s
 => => sha256:1320d210901ab5695dad77863eb70648aeeebe2647fc29a1ae49b379cea653a3 2.85MB / 2.85MB        20.2s
 => => extracting sha256:2cf9c2b42f41b1845f3e4421b723d56146db82939dc884555e077768e18132f4              2.5s
 => => extracting sha256:c4c40c3e3cdf945721f480e1d939aac857876fdb5c33b8fbfcf655c63b0b9428              6.6s
 => => extracting sha256:c05cc1123d7e335d59b0f465c23b7ad2ad27f4875b6c3eab41c65a9b50efa382             11.1s
 => => extracting sha256:b6f29ccdcc551647511d3473f89c94b2ee7fbce3e65226908ea74cfc5c586697              0.4s
 => => extracting sha256:9c8be2164d2a0d2ef7cd16a93364ef0d7e579861778b0fa07071f8899d950aa2              0.9s
 => => extracting sha256:d00b6068660376d1ea605bdb6a2bca31e9e3a05c10d50c48dd8a513824039360              0.0s
 => => extracting sha256:1320d210901ab5695dad77863eb70648aeeebe2647fc29a1ae49b379cea653a3              0.4s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 251B                                                                             0.0s
 => [2/5] WORKDIR /code                                                                                       0.7s
 => [3/5] COPY ./requirements.txt /code/requirements.txt                                                      0.0s
 => [4/5] RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt                                 11.6s
 => [5/5] COPY ./app /code/app                                                                                0.0s
 => exporting to image                                                                                        0.5s
 => => exporting layers                                                                                       0.5s
 => => writing image sha256:f3aa3ea9ed9da2cfb14cc6e92f0bcc88344b911fec75e9e0c4798e397ae5df8d                  0.0s
root@jason-virtual-machine:~#
```

Here we can see I built the image by identifying the proper path where I stored my files.

```
root@jason-virtual-machine:~# docker build /home/jason/Kupernetes Test -t Cluster
[+] Building 0.0s (0/0)                                                                              docker:default
ERROR: invalid tag "Cluster": repository name must be lowercase
root@jason-virtual-machine:~# docker build /home/jason/Kupernetes Test -t cluster
[+] Building 0.3s (10/10) FINISHED                                                                   docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 262B                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.9                                                 0.2s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [1/5] FROM docker.io/library/python:3.9@sha256:5930d54604459569953a4164839ad9bb32f6a1c76e6740c84a  0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 213B                                                                             0.0s
 => CACHED [2/5] WORKDIR /code                                                                                0.0s
 => CACHED [3/5] COPY ./requirements.txt /code/requirements.txt                                               0.0s
 => CACHED [4/5] RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt                           0.0s
 => CACHED [5/5] COPY ./app /code/app                                                                         0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:f3aa3ea9ed9da2cfb14cc6e92f0bcc88344b911fec75e9e0c4798e397ae5df8d                  0.0s
 => => naming to docker.io/library/cluster                                                                    0.0s
root@jason-virtual-machine:~# docker run -p 8000:80 cluster
INFO     Using path app/main.py
INFO     Resolved absolute path /code/app/main.py
INFO     Searching for package file structure from directories with   init  .py
         files
INFO     Importing from /code

  Python package file structure
  ┌──────────────────────────────┐
  │ 📁 app                       │
  │ ├── 🐍   init  .py           │
  │ └── 🐍 main.py               │
  └──────────────────────────────┘
```



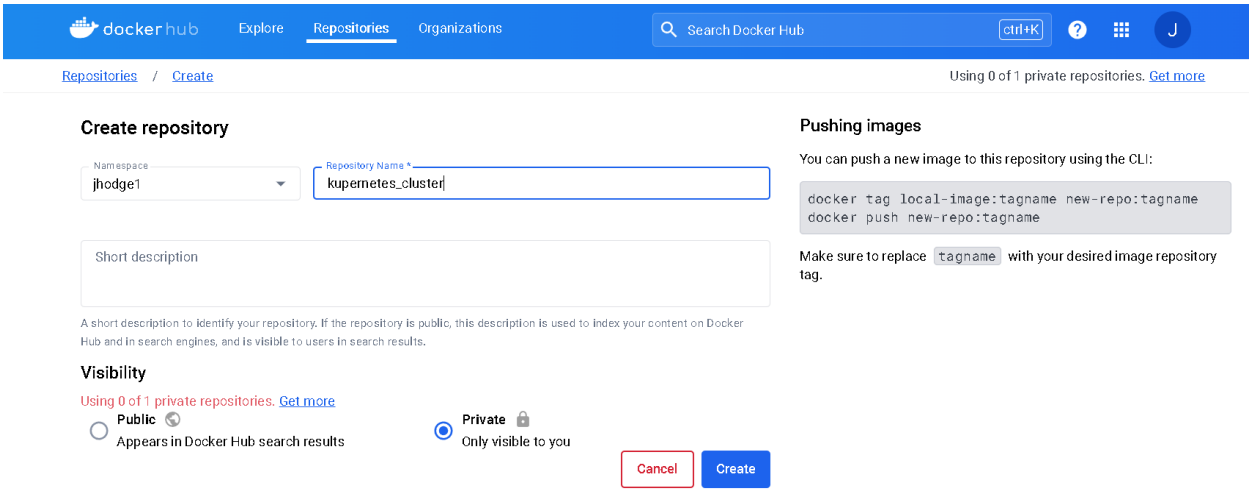
Here we can see I tagged this image by doing “-t” and the tag name I wanted “cluster”. I then connected port (-p) 8000 to port 80 that is specified inside my container.

requirements.txt | main.py 1 | ! deplyment.yaml | Dockerfile

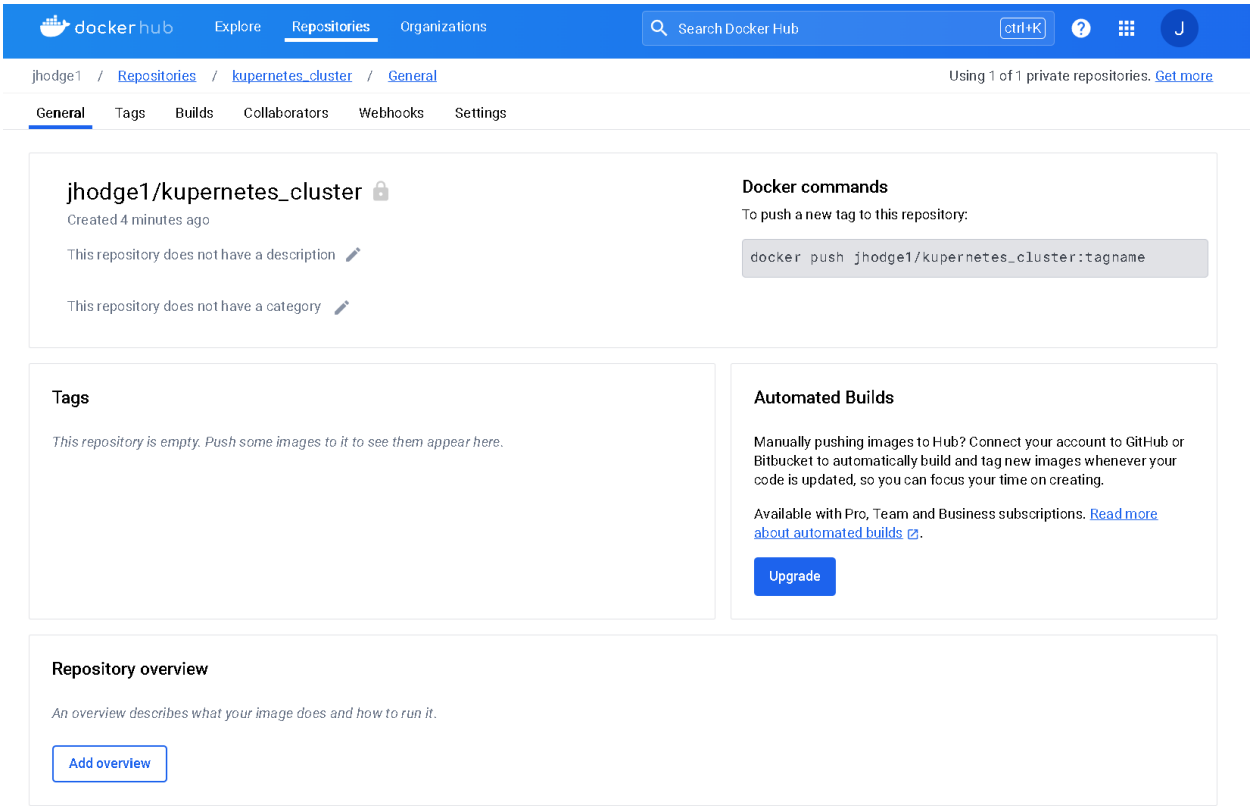
kupernetes > ! deplyment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-container
        image: nginx:latest
        ports:
        - containerPort: 80
```

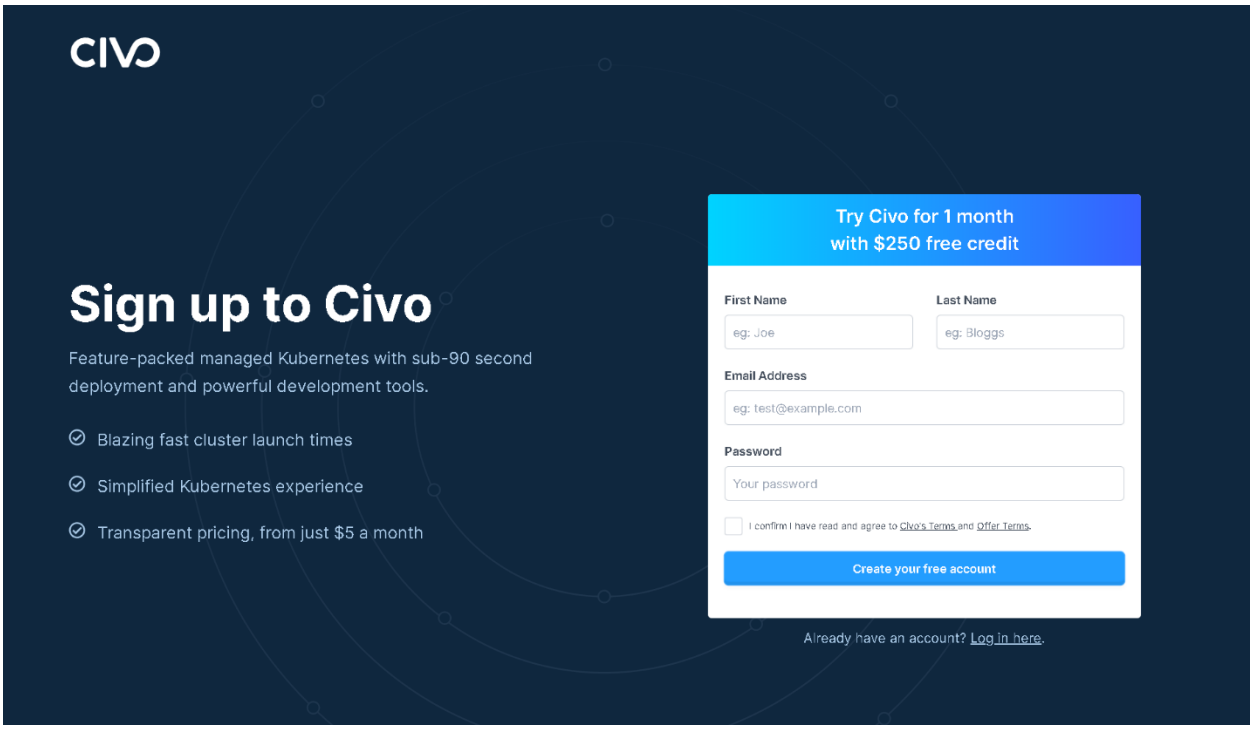
Here is a basic yaml file used to create a Kupernetes cluster deployment.



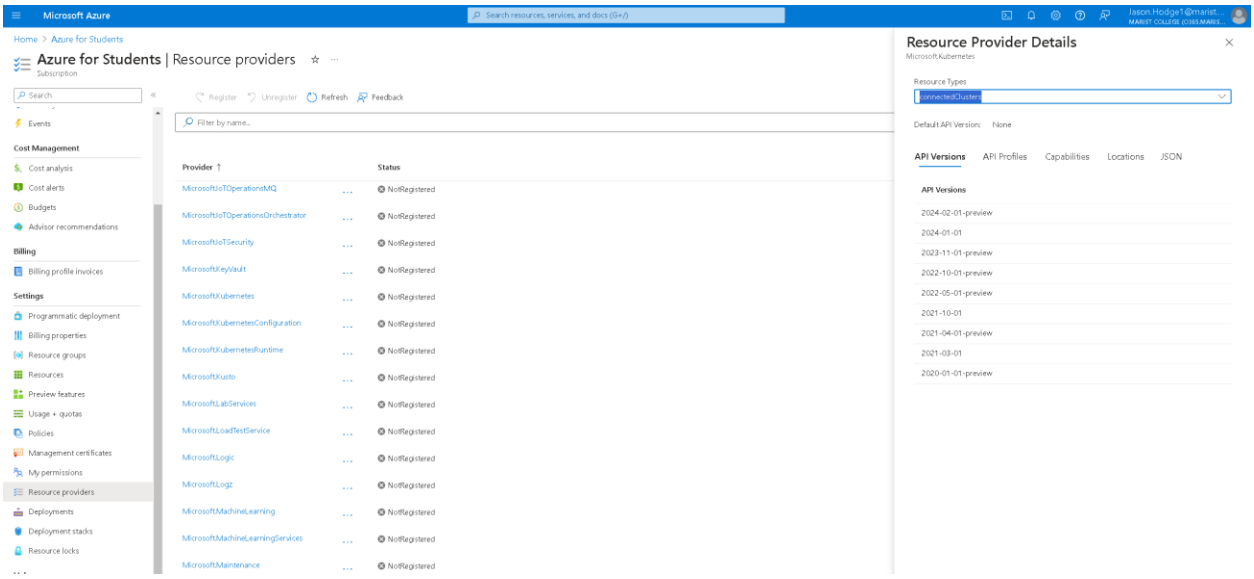
Here I created a docker repository for Kupernetes cluster tags.



Here we can see the created repository where it can either be public or private. Utilizing the tag we can build and add applications to the tags section of the repository.



This is one of some paid subscription-based services to create and host Kupernetes clusters on.



Azure student did not have Kupernetes available.

Conclusion:

This lab proved to be challenging, and it came to a halt right up till I needed a subscription with a third-party service to create and host my cluster on. This was a great experience and I now have some working knowledge with getting to the point where I can host an application I containerized. The next steps would be to purchase a service where I can create a cluster using an application, I create to host a working Kupernetes service. This is something I would like to take to the next step someday.

References:

https://www.buildpiper.io/blogs/bp-blog-exploring-the-benefits-of-kubernetes/#:~:text=Kubernetes%20provides%20a%20robust%20platform,DevOps%2C%20Cloud%20and%20DevSecOps%20practices.

https://www.youtube.com/watch?v=TlHvYWVUZyc

https://www.youtube.com/watch?v=XltFOyGanYE&t=355s

https://www.youtube.com/watch?v=iWS9ogMPOI0

https://www.youtube.com/watch?v=r2zuL9MW6wc&t=484s

https://docs.docker.com/reference/cli/docker/image/build/

https://docs.docker.com/desktop/get-started/#switch-between-windows-and-linux-containers

https://spacelift.io/blog/kubernetes-deployment-yaml

https://minikube.sigs.k8s.io/docs/start/

https://minikube.sigs.k8s.io/docs/drivers/docker/

https://fastapi.tiangolo.com/#create-it

https://fastapi.tiangolo.com/deployment/docker/

https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/