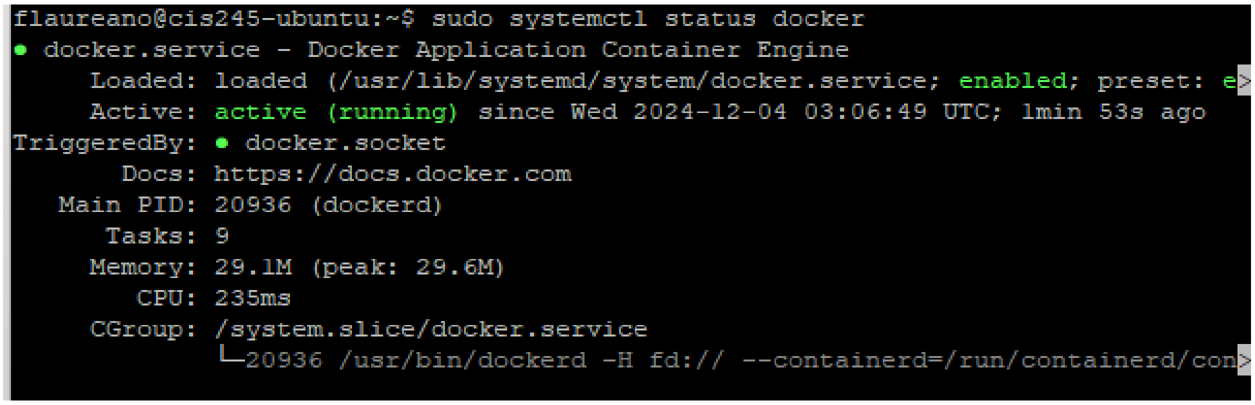
Francisco Laureano Containers
CIS 245

This guide will guide you to install Docker and Kubernetes onto your ubuntu server.

Docker Installation

We will first start with Docker, to install docker you will need Sudo permissions to install the application. Log into your server and run **sudo apt install -y docker.io.** This will download docker and once it's fininshed enter sudo systemctl status docker to see if docker is running. If installed correctly, you should see the status active and ready for use.

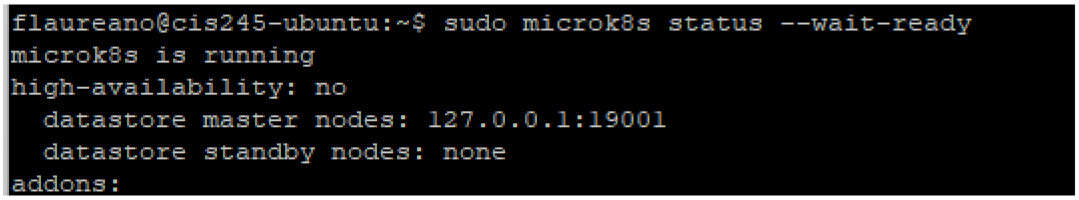
If docker is not running, you can enter systemctl start docker then sudo systemctl enable docker.



```
flaureano@cis245-ubuntu:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: e>
     Active: active (running) since Wed 2024-12-04 03:06:49 UTC; 1min 53s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 20936 (dockerd)
      Tasks: 9
     Memory: 29.1M (peak: 29.6M)
        CPU: 235ms
     CGroup: /system.slice/docker.service
             └─20936 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
```

Kubernetes Installation

To install Kubernetes, we will be leveraging Microk8s to install Kubernetes. Enter **sudo snap install microk8s –classic** and wait for the installation to finish. Once finished, check the status with **sudo microk8s status --wait-ready** to make sure that microk8s is running. If done successfully, you will be met with the microk8s status as shown below.



```
flaureano@cis245-ubuntu:~$ sudo microk8s status --wait-ready
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
```

Now to avoid having to sudo every command, lets add your user to microk8s as a privileged user so you can run commands without sudoing everything. Run these commands:

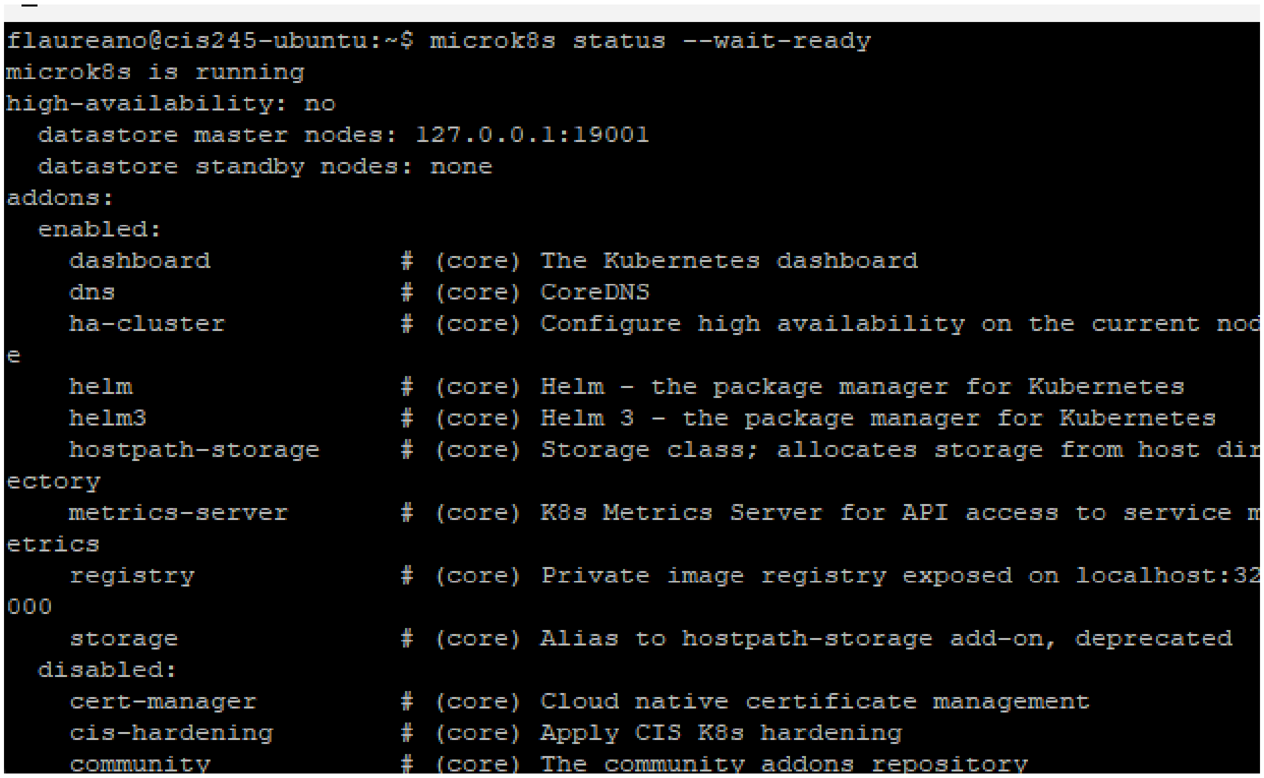
sudo usermod -aG microk8s $USER [replace $USER with your username]

sudo chown -f -R $USER ~/.kube **[**replace $USER with your username]

These commands allow your user to run commands without sudo.

Francisco Laureano Containers
CIS 245

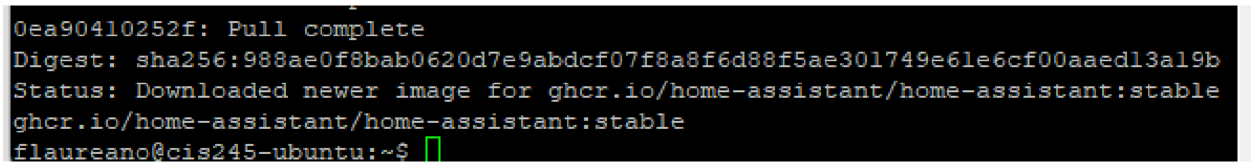
Now we're going to enable a few services to help us with Kubernetes. Run these commands to turn on the following services: **microk8s enable dashboard, microk8s enable dns, microk8s enable registry.** You can verify that these are installed and running by running **microk8s status --wait-ready** after.



```
flaureano@cis245-ubuntu:~$ microk8s status --wait-ready
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dashboard            # (core) The Kubernetes dashboard
    dns                  # (core) CoreDNS
    ha-cluster           # (core) Configure high availability on the current nod
e
    helm                 # (core) Helm - the package manager for Kubernetes
    helm3                # (core) Helm 3 - the package manager for Kubernetes
    hostpath-storage     # (core) Storage class; allocates storage from host dir
ectory
    metrics-server       # (core) K8s Metrics Server for API access to service m
etrics
    registry             # (core) Private image registry exposed on localhost:32
000
    storage              # (core) Alias to hostpath-storage add-on, deprecated
  disabled:
    cert-manager         # (core) Cloud native certificate management
    cis-hardening        # (core) Apply CIS K8s hardening
    community            # (core) The community addons repository
```

Webserver through Docker

Next we're going to stand up a web server using docker and to do so we're going to run **docker pull ghcr.io/home-assistant/home-assistant:stable**. This downloads the latest web server image from apache home assistant and will take some time to install so kick back a bit. When finished you will see a message like this:

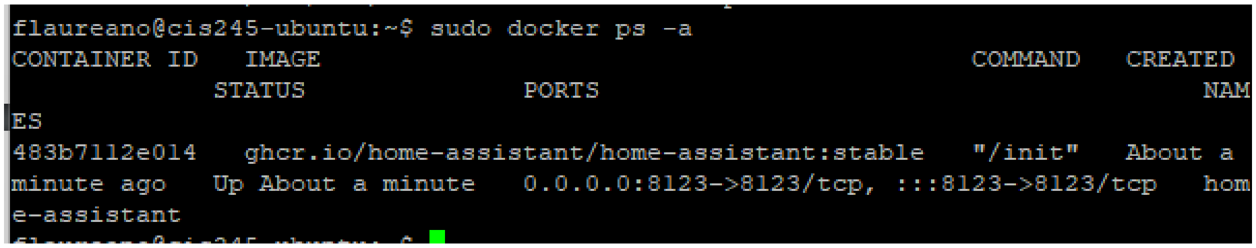


```
0ea90410252f: Pull complete
Digest: sha256:988ae0f8bab0620d7e9abdcf07f8a8f6d88f5ae301749e61e6cf00aaed13a19b
Status: Downloaded newer image for ghcr.io/home-assistant/home-assistant:stable
ghcr.io/home-assistant/home-assistant:stable
flaureano@cis245-ubuntu:~$
```

Next we're going to create the directory to place this configuration files in. We're running this in our home directory, so we're going to run **mkdir home-assistant to** create this directory. Once created, enter **sudo docker run -d \ --name home-assistant \ --restart=unless-stopped \ -p 8123:8123 \ -v /home/your-username/home-assistant:/config \ ghcr.io/home-assistant/home-assistant:stable.**

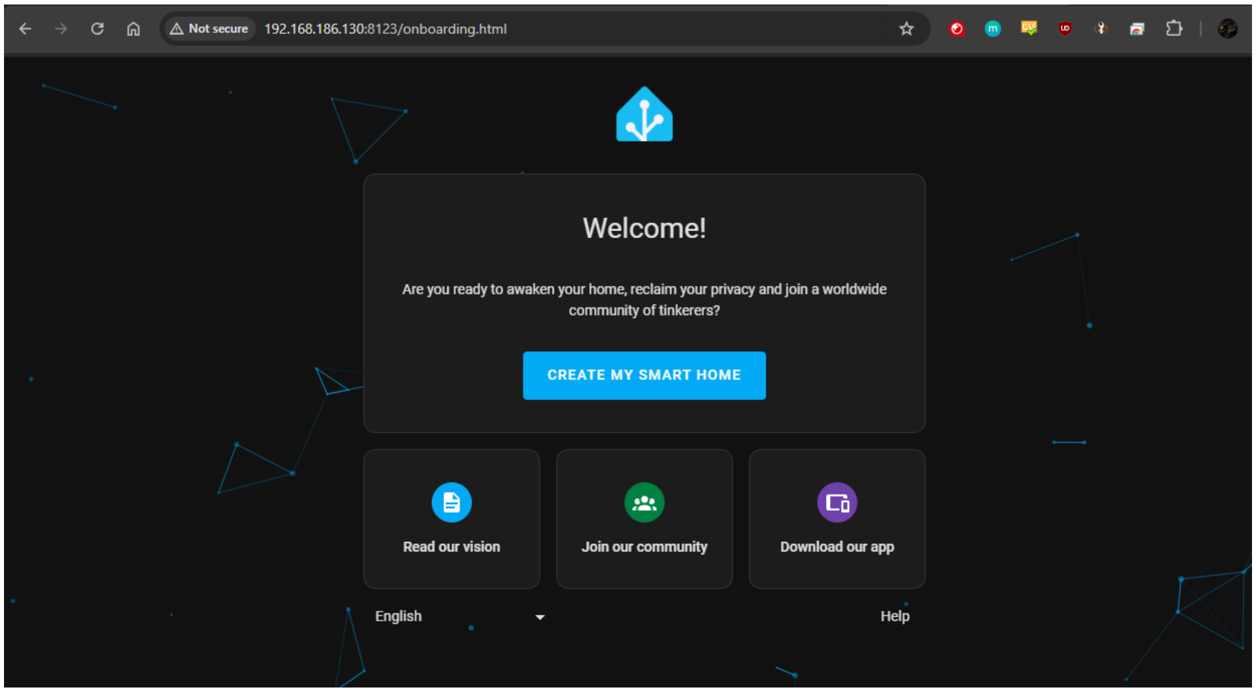
Francisco Laureano Containers
CIS 245

After creating that container, run **sudo docker ps -a** to verify that it's running. If you're successful, you should see something like this.



```
flaureano@cis245-ubuntu:~$ sudo docker ps -a
CONTAINER ID   IMAGE                                         COMMAND    CREATED
             STATUS                PORTS                                       NAM
ES
483b7112e014   ghcr.io/home-assistant/home-assistant:stable   "/init"    About a
minute ago   Up About a minute    0.0.0.0:8123->8123/tcp, :::8123->8123/tcp    hom
e-assistant
```

Our final test is to log onto the server. For this open a web browser and type in the IP of your server with the port :8123. For my server, this is the IP http://192.168.186.130:8123.


Not secure
192.168.186.130:8123/onboarding.html
Welcome!
Are you ready to awaken your home, reclaim your privacy and join a worldwide community of tinkerers?
CREATE MY SMART HOME
Read our vision
Join our community
Download our app
English
Help

Once completed, you are finished running a apache web server through docker! Good job!

Francisco Laureano Containers
CIS 245

Container Resource

https://www.rancher.com/learn-the-basics

This website was actually super helpful for me to fully grasp containers. This resource provides in-depth information on nearly every aspect of containers, atleast enough to get a novice like myself up to speed with them. After reading the guides, they have tutorials that anyone can get running so that they can get their hands dirty further with containers. Truly a superb resource!