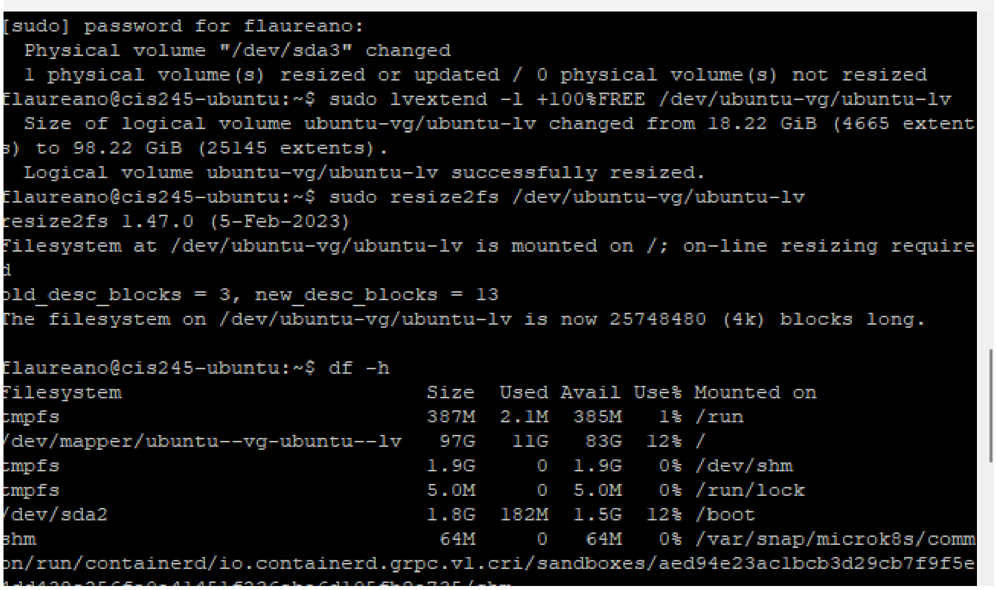
Francisco Laureano Containers – MiniKube
CIS245

MiniKube

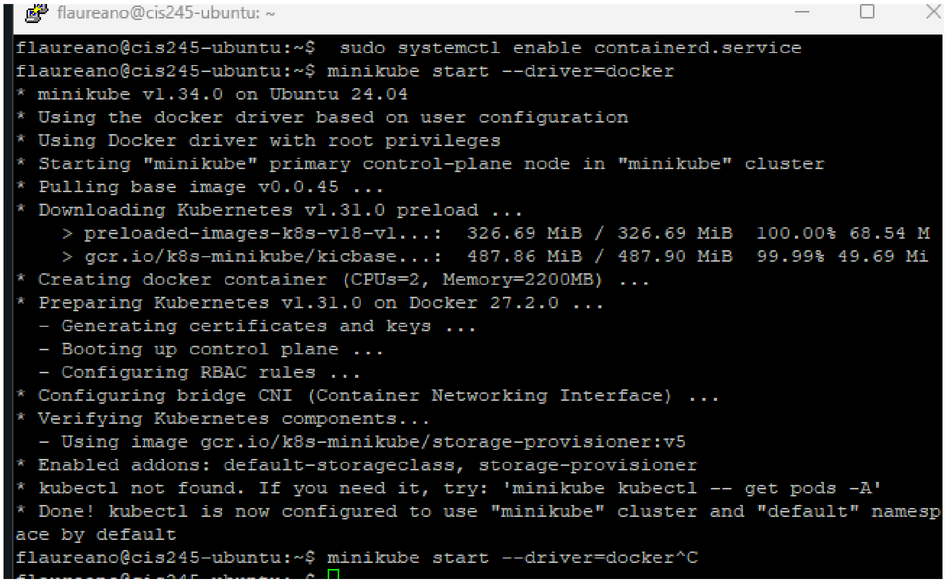
The initial process of getting MiniKube up and running was actually tricky for me. When I made this vm, I didn't think we would exceed the default storage size that VMWare recommends, which was 20g, so I didn't bother increasing them. The problem is that MiniKube recommends 20gbs of storage space for jusut MiniKube which is my entire allocated space. My solution to this was to extend the size of my hard drive and follow a guide to fix the size on my partition.



```
[sudo] password for flaureano:
  Physical volume "/dev/sda3" changed
  1 physical volume(s) resized or updated / 0 physical volume(s) not resized
flaureano@cis245-ubuntu:~$ sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
  Size of logical volume ubuntu-vg/ubuntu-lv changed from 18.22 GiB (4665 extent
s) to 98.22 GiB (25145 extents).
  Logical volume ubuntu-vg/ubuntu-lv successfully resized.
flaureano@cis245-ubuntu:~$ sudo resize2fs /dev/ubuntu-vg/ubuntu-lv
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/ubuntu-vg/ubuntu-lv is mounted on /; on-line resizing require
d
old_desc_blocks = 3, new_desc_blocks = 13
The filesystem on /dev/ubuntu-vg/ubuntu-lv is now 25748480 (4k) blocks long.

flaureano@cis245-ubuntu:~$ df -h
Filesystem                         Size  Used Avail Use% Mounted on
tmpfs                              387M  2.1M  385M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv   97G   11G   83G  12% /
tmpfs                              1.9G     0  1.9G   0% /dev/shm
tmpfs                              5.0M     0  5.0M   0% /run/lock
/dev/sda2                          1.8G  182M  1.5G  12% /boot
shm                                 64M     0   64M   0% /var/snap/microk8s/comm
on/run/containerd/io.containerd.grpc.v1.cri/sandboxes/aed94e23ac1bcb3d29cb7f9f5e
```

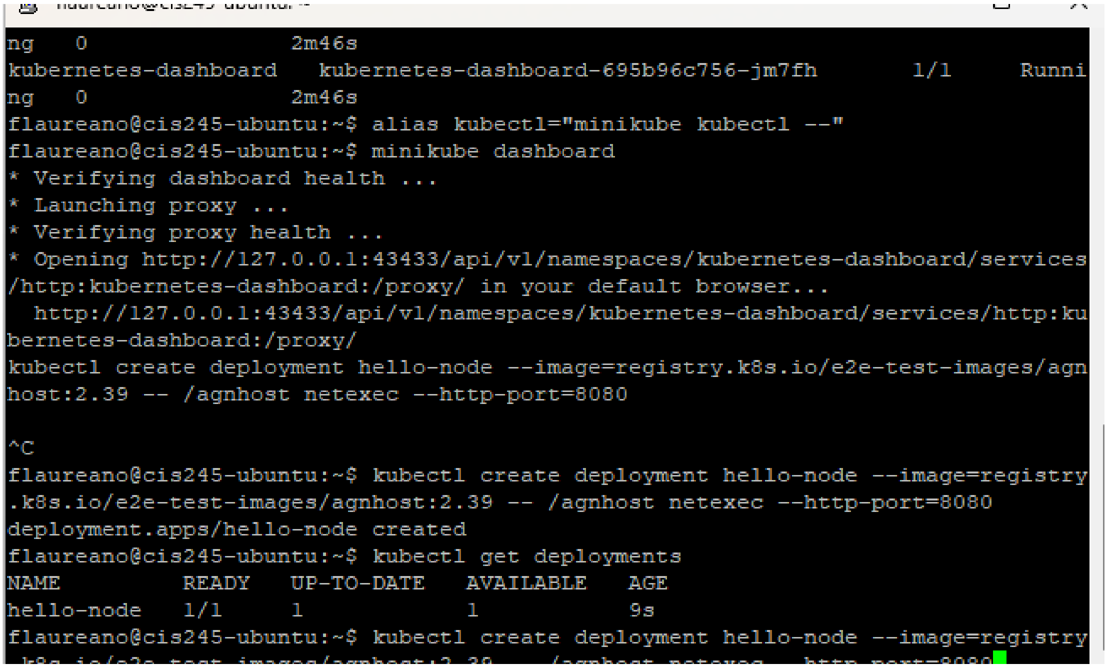
https://www.rootusers.com/how-to-increase-the-size-of-a-linux-lvm-by-expanding-the-virtual-machine-disk/

I immediately ran into issues with getting minikube to select a driver so I had to activate myself to the docker group with **newgrp docker.** Then after that command I had to download the docker driver to use with minikube using **minikube start --driver=docker**



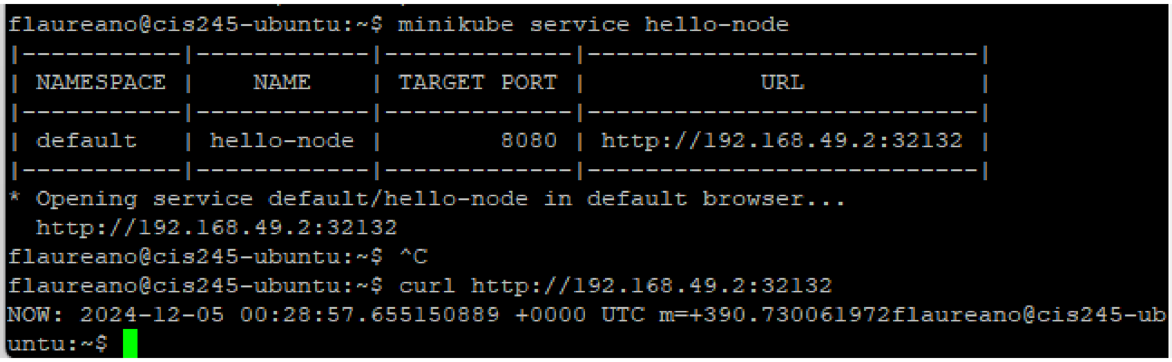
```
flaureano@cis245-ubuntu:~$  sudo systemctl enable containerd.service
flaureano@cis245-ubuntu:~$ minikube start --driver=docker
* minikube v1.34.0 on Ubuntu 24.04
* Using the docker driver based on user configuration
* Using Docker driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Downloading Kubernetes v1.31.0 preload ...
    > preloaded-images-k8s-v18-v1...:  326.69 MiB / 326.69 MiB  100.00% 68.54 M
    > gcr.io/k8s-minikube/kicbase...:  487.86 MiB / 487.90 MiB  99.99% 49.69 Mi
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namesp
ace by default
flaureano@cis245-ubuntu:~$ minikube start --driver=docker^C
flaureano@cis245-ubuntu:~$
```

After that I ran **minikube config set driver docker** to set that driver for the default driver. After this I need to get kubectl so I run **minikube kubectl -- get po -A.** Then I opened a minikube dashboard with **minikube dashboard –url** and after I create a deployment of the hello-node with **kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080.** I can check the status with **kubectl get deployments** and it will show me the active nodes.



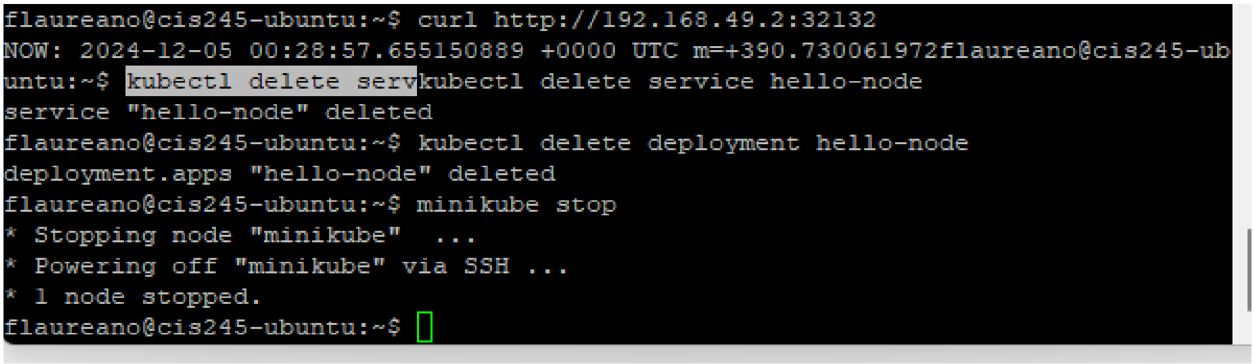
```
ng   0              2m46s
kubernetes-dashboard   kubernetes-dashboard-695b96c756-jm7fh       1/1     Runni
ng   0              2m46s
flaureano@cis245-ubuntu:~$ alias kubectl="minikube kubectl --"
flaureano@cis245-ubuntu:~$ minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:43433/api/v1/namespaces/kubernetes-dashboard/services
/http:kubernetes-dashboard:/proxy/ in your default browser...
  http://127.0.0.1:43433/api/v1/namespaces/kubernetes-dashboard/services/http:ku
bernetes-dashboard:/proxy/
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agn
host:2.39 -- /agnhost netexec --http-port=8080

^C
flaureano@cis245-ubuntu:~$ kubectl create deployment hello-node --image=registry
.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
flaureano@cis245-ubuntu:~$ kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           9s
flaureano@cis245-ubuntu:~$ kubectl create deployment hello-node --image=registry
.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
```

Next we're going to create a service, so I run **kubectl expose deployment hello-node --type=LoadBalancer --port=8080.** Then I'll run **minikube service hello-node** and it will print out the url and port of the service I created for my hello-node. In this case, my IP is 192.168.49.2:32132 so when I curl 192.168.49.2:32132 it returns an alive message with **NOW: 2024-12-05.65510899**



```
flaureano@cis245-ubuntu:~$ minikube service hello-node
|-----------|------------|-------------|---------------------------|
| NAMESPACE |    NAME    | TARGET PORT |            URL            |
|-----------|------------|-------------|---------------------------|
| default   | hello-node |        8080 | http://192.168.49.2:32132 |
|-----------|------------|-------------|---------------------------|
* Opening service default/hello-node in default browser...
  http://192.168.49.2:32132
flaureano@cis245-ubuntu:~$ ^C
flaureano@cis245-ubuntu:~$ curl http://192.168.49.2:32132
NOW: 2024-12-05 00:28:57.655150889 +0000 UTC m=+390.730061972flaureano@cis245-ub
untu:~$
```

After this installation, I checked my addon list with **minikube addons list minikube addons list** and enabled then disabled metrics-server with **minikube addons enable metrics-server** then **minikube disable metrics-server**. At the end I deleted my service and deployment as I waved goodbye to minikube.



```
flaureano@cis245-ubuntu:~$ curl http://192.168.49.2:32132
NOW: 2024-12-05 00:28:57.655150889 +0000 UTC m=+390.730061972flaureano@cis245-ub
untu:~$ kubectl delete servkubectl delete service hello-node
service "hello-node" deleted
flaureano@cis245-ubuntu:~$ kubectl delete deployment hello-node
deployment.apps "hello-node" deleted
flaureano@cis245-ubuntu:~$ minikube stop
* Stopping node "minikube"  ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
flaureano@cis245-ubuntu:~$
```

After this experience, I can definitely say that I'll get my hands dirty with Kubernetes, docker, and minikube much more! It was a really great experience that taught me a lot about both linux and containers.