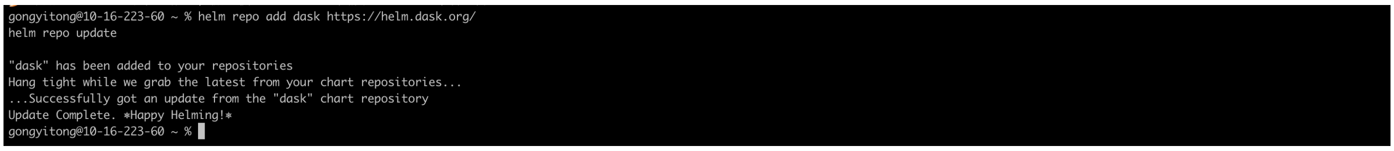
Environment

- Cluster with 3 nodes Nivdia T4 GPU
- Standard GKE cluster :region us-central1
- 10 GB ImageNet dataset dir:
- ResNet 50 pre-trained model

Cluster Creation

Used the following gcloud command to create a cluster named 'my-dask':

`gcloud container clusters create dask-cluster --num-nodes=3 --zone=us-central1 --disk-type=pd-standard --disk-size=10`



```
gongyitong@10-16-223-60 ~ % helm repo add dask https://helm.dask.org/
helm repo update

"dask" has been added to your repositories
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "dask" chart repository
Update Complete. ⎈Happy Helming!⎈
gongyitong@10-16-223-60 ~ %
```

Helm Installation of Dask

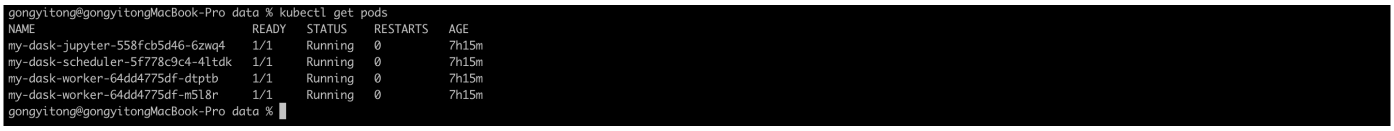
Installed Dask using Helm with the following command:

```
helm install my-dask dask/dask \
    --set scheduler.replicas=1 \
    --set worker.replicas=2
```

Check Cluster Status

Checked the cluster status using:

`kubectl get pods`



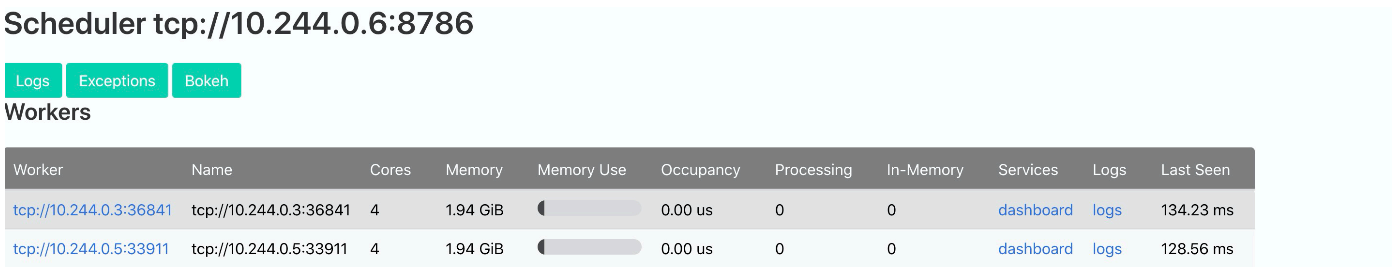
```
gongyitong@gongyitongMacBook-Pro data % kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
my-dask-jupyter-558fcb5d46-6zwq4    1/1     Running   0          7h15m
my-dask-scheduler-5f778c9c4-4ltdk   1/1     Running   0          7h15m
my-dask-worker-64dd4775df-dtptb     1/1     Running   0          7h15m
my-dask-worker-64dd4775df-m5l8r     1/1     Running   0          7h15m
gongyitong@gongyitongMacBook-Pro data %
```

To learn more about the release, try:

Monitoring Cluster Status via External Interface

Accessed the cluster status through:

http://10.244.0.6:8786/status



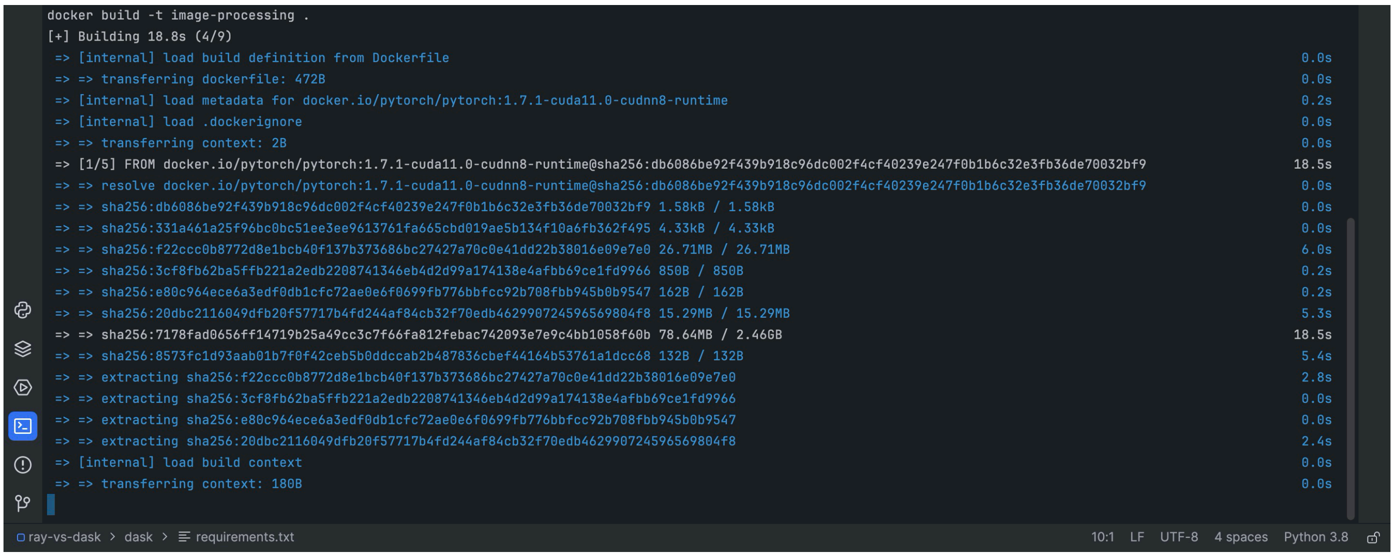
Scheduler tcp://10.244.0.6:8786

Workers

Worker	Name	Cores	Memory	Memory Use	Occupancy	Processing	In-Memory	Services	Logs	Last Seen
tcp://10.244.0.3:36841	tcp://10.244.0.3:36841	4	1.94 GiB		0.00 us	0	0	dashboard	logs	134.23 ms
tcp://10.244.0.5:33911	tcp://10.244.0.5:33911	4	1.94 GiB		0.00 us	0	0	dashboard	logs	128.56 ms

Create Task Environment and Run Image

Built the Docker image with the following command:docker build -t image-processing



```
docker build -t image-processing .
[+] Building 18.8s (4/9)
 => [internal] load build definition from Dockerfile                                                                              0.0s
 => => transferring dockerfile: 472B                                                                                              0.0s
 => [internal] load metadata for docker.io/pytorch/pytorch:1.7.1-cuda11.0-cudnn8-runtime                                          0.2s
 => [internal] load .dockerignore                                                                                                 0.0s
 => => transferring context: 2B                                                                                                   0.0s
 => [1/5] FROM docker.io/pytorch/pytorch:1.7.1-cuda11.0-cudnn8-runtime@sha256:db6086be92f439b918c96dc002f4cf40239e247f0b1b6c32e3fb36de70032bf9  18.5s
 => => resolve docker.io/pytorch/pytorch:1.7.1-cuda11.0-cudnn8-runtime@sha256:db6086be92f439b918c96dc002f4cf40239e247f0b1b6c32e3fb36de70032bf9  0.0s
 => => sha256:db6086be92f439b918c96dc002f4cf40239e247f0b1b6c32e3fb36de70032bf9 1.58kB / 1.58kB                                    0.0s
 => => sha256:331a461a25f96bc0bc51ee3ee9613761fa665cbd019ae5b134f10a6fb362f495 4.33kB / 4.33kB                                    0.0s
 => => sha256:f22ccc0b8772d8e1bcb40f137b373686bc27427a70c0e41dd22b38016e09e7e0 26.71MB / 26.71MB                                  6.0s
 => => sha256:3cf8fb62ba5ffb221a2edb2208741346eb4d2d99a174138e4afbb69ce1fd9966 850B / 850B                                        0.2s
 => => sha256:e80c964ece6a3edf0db1cfc72ae0e6f0699fb776bbfcc92b708fbb945b0b9547 162B / 162B                                        0.2s
 => => sha256:20dbc2116049dfb20f57717b4fd244af84cb32f70edb462990724596569804f8 15.29MB / 15.29MB                                  5.3s
 => => sha256:7178fad0656ff14719b25a49cc3c7f66fa812febac742093e7e9c4bb1058f60b 78.64MB / 2.46GB                                  18.5s
 => => sha256:8573fc1d93aab01b7f0f42ceb5b0ddccab2b487836cbef44164b53761a1dcc68 132B / 132B                                        5.4s
 => => extracting sha256:f22ccc0b8772d8e1bcb40f137b373686bc27427a70c0e41dd22b38016e09e7e0                                         2.8s
 => => extracting sha256:3cf8fb62ba5ffb221a2edb2208741346eb4d2d99a174138e4afbb69ce1fd9966                                         0.0s
 => => extracting sha256:e80c964ece6a3edf0db1cfc72ae0e6f0699fb776bbfcc92b708fbb945b0b9547                                         0.0s
 => => extracting sha256:20dbc2116049dfb20f57717b4fd244af84cb32f70edb462990724596569804f8                                         2.4s
 => [internal] load build context                                                                                                 0.0s
 => => transferring context: 180B                                                                                                 0.0s
```

ray-vs-dask > dask > requirements.txt
10:1 LF UTF-8 4 spaces Python 3.8

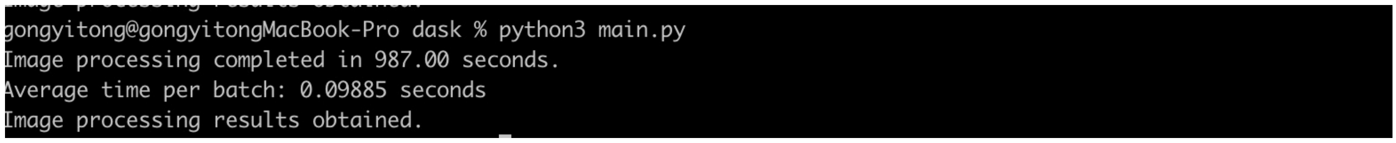
Image name: image-processing

Applied changes to the environment image to: `my-dask-scheduler.yaml` and `my-dask-worker.yaml`

Run Script in Environment

Connected to the scheduler using the Dask client with the following line of code:

```
client = Client('10.244.0.6:8786')  # Dask scheduler address and port
```



```
gongyitong@gongyitongMacBook-Pro dask % python3 main.py
Image processing completed in 987.00 seconds.
Average time per batch: 0.09885 seconds
Image processing results obtained.
```