# Overview

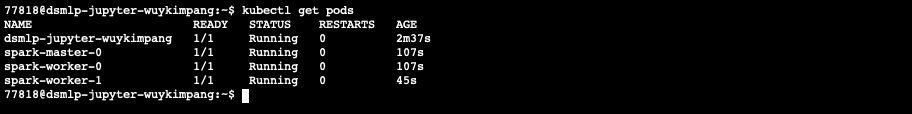
Spark instances are created on the Data Science and Machine Learning Platform at UCSD.

# Start a Spark Cluster

1. Navigate to datahub.ucsd.edu
2. Select your course and click “Launch Environment”

*Launching the course will automatically spin up 1 spark master and 2 spark workers. Give the server some time to spin up the spark cluster.*

1. Confirm that the cluster has started by opening a jupyter terminal and running the command “kubectl get pods”



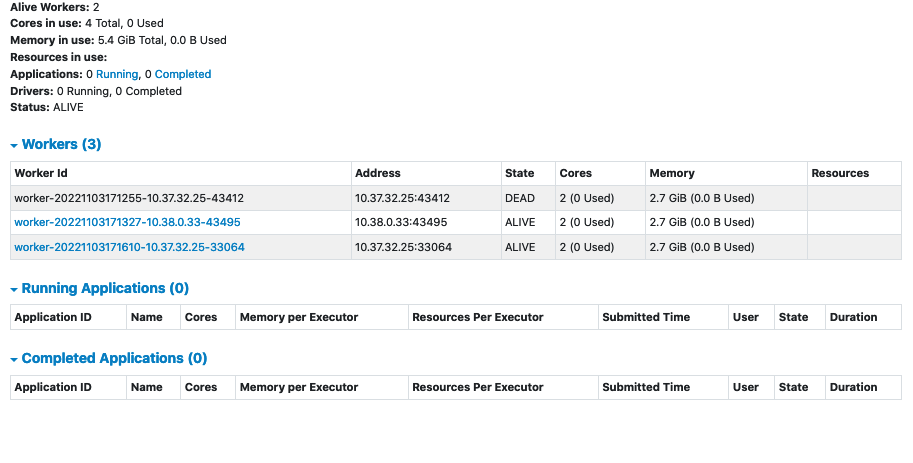
*Pods will start off with a status of 0/1 under column “READY”. Please wait until you see status “1/1*

1. Confirm that the master has been elected by running “kubectl logs spark-master-0”



*See the last logs “I have been elected leader!”*

1. Once the cluster has been created, navigate back to the notebook tree tab (<https://datahub.ucsd.edu/hub/><username>/tree?” and select New > spark-driver



1. There’s a test jupyter notebook that confirms that the spark cluster works located at /opt/sanity\_check.ipynb. You can confirm that you can use pyspark with this cluster by running that notebook. See optional instruction steps below for more details
2. (OPTIONAL) Go back to the jupyter terminal and run the command “cp /opt/sanity\_check.ipynb ~”
3. (OPTIONAL) Open the jupyter notebook and execute it. If all goes well, you’ll see the output like below



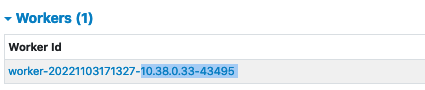
# Troubleshooting

## Why don’t I see any or just a single spark worker at the spark dashboard?

When the workers try to join the cluster, they look for the master node. If the master node is still being setup, they’ll ping the master and try to reconnect up to 7 times. If they’ve exceeded their connection retry times, the pod will give up and fail to connect.

To fix this, and do the following:

1. check your spark dashboard and get the IP of the connected worker pod (if any)



1. Open a jupyter terminal and run `kubectl get pods -o wide`
2. Find the **worker pod** whose Cluster IP is not on the list. Copy the pod name to your clipboard
3. Run kubectl delete pod <worker\_pod\_name>

The above will restart the worker pods and will have a better chance of successfully joining the master pod if it’s been elected.

You can confirm that it joined the master pod by looking at the master pods logs with the following command: “kubectl logs <master\_pod\_name>”. See if there's an output towards the bottom that says “Registering worker…”. This may take some time as well since the worker pod must start up. Alternatively, you may also simply check the spark dashboard