Integrating Apache Livy with DC/OS Spark

# Overview

Apache Livy makes submitting Spark Jobs possible from anywhere. In addition, Livy makes it possible to run Interactive Scala or Python all without any code changes required. Many of the popular Spark notebooks also require Apache Livy.

Mesosphere DC/OS leverages all the wonderful goodness of Apache Livy and adds scalability…a must for organizations with many Data Scientists.

This article quickly describes the steps required to setup add Livy to your DC/OS cluster.

# Requirements

The installation steps should work for a variety of DC/OS, Livy and Spark versions. That said the following versions have been tested for the creation of this article and are known to work for examples documented here:

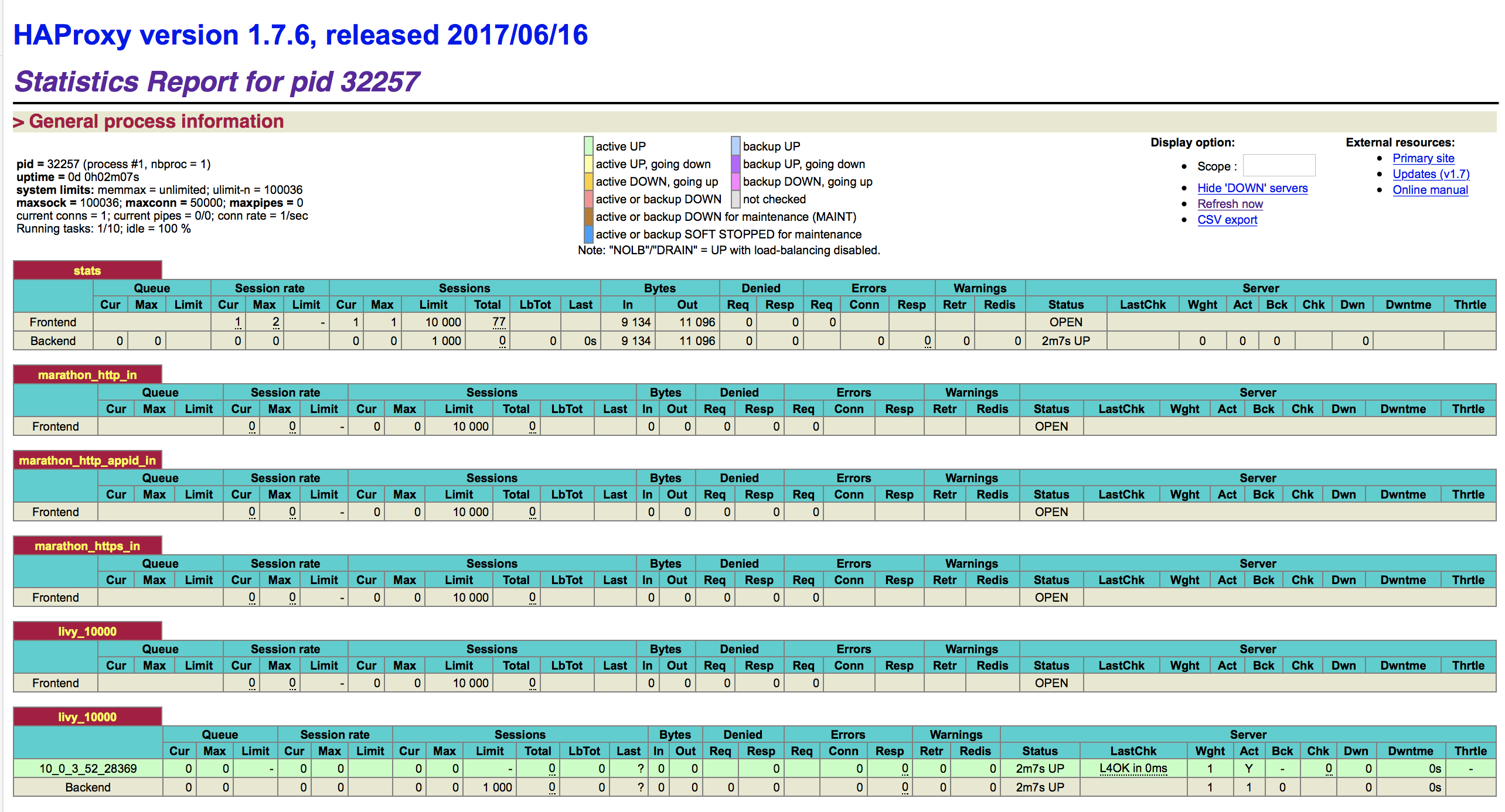
* [Mesosphere DC/OS 1.10 (either Open Source or Enterprise)](https://dcos.io/)
* [Apache Livy 0.50](git://git.apache.org/incubator-livy.git)

# Installation Steps

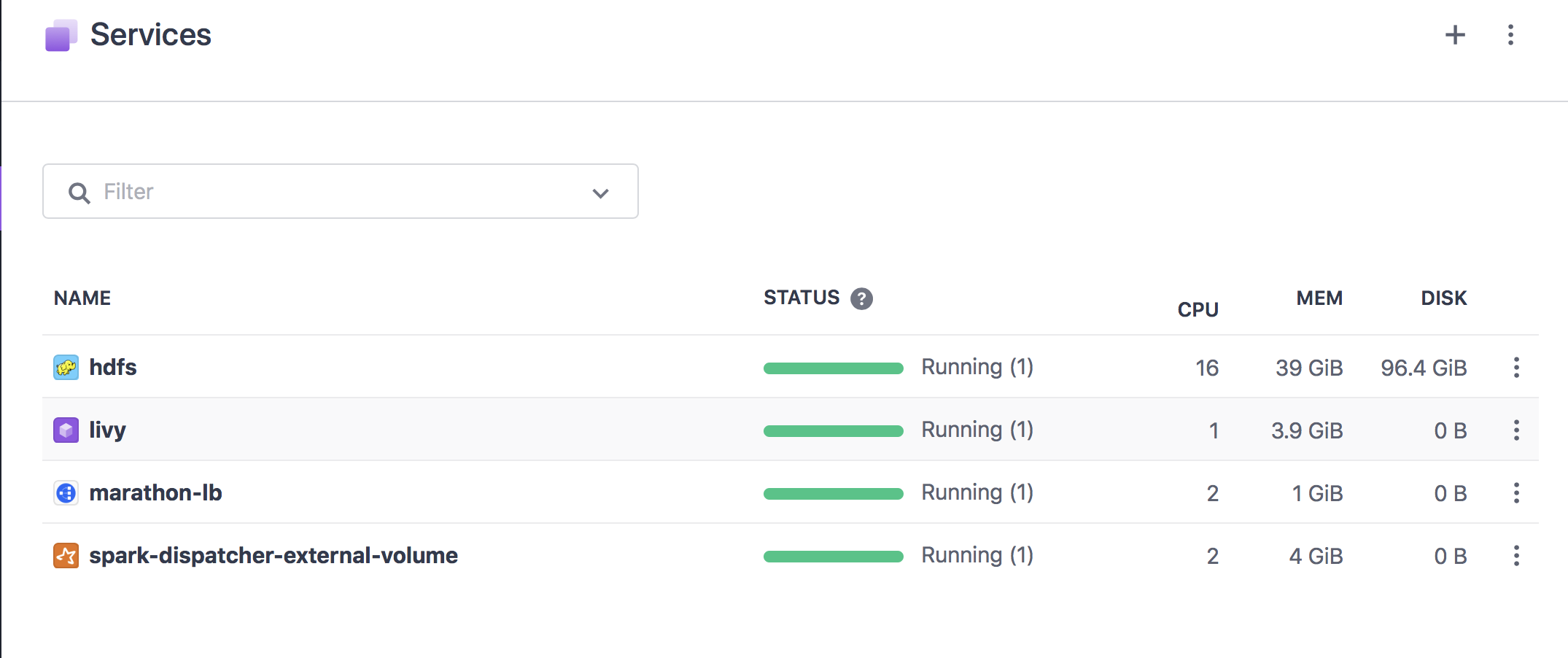
1. Install the Marathon-LB service

*dcos* package install --yes marathon-lb

1. Install DC/OS version of Spark with the Dispatcher
   1. With HDFS
2. *dcos* marathon app add spark-dispatcher-ucr-hdfs-eventlog-external-volume.json
   1. Without HDFS
3. Install DC/OS Livy service
4. *dcos* marathon app add livy.json
5. Get the Public IP for your cluster and verify port 10000 for the Livy Service



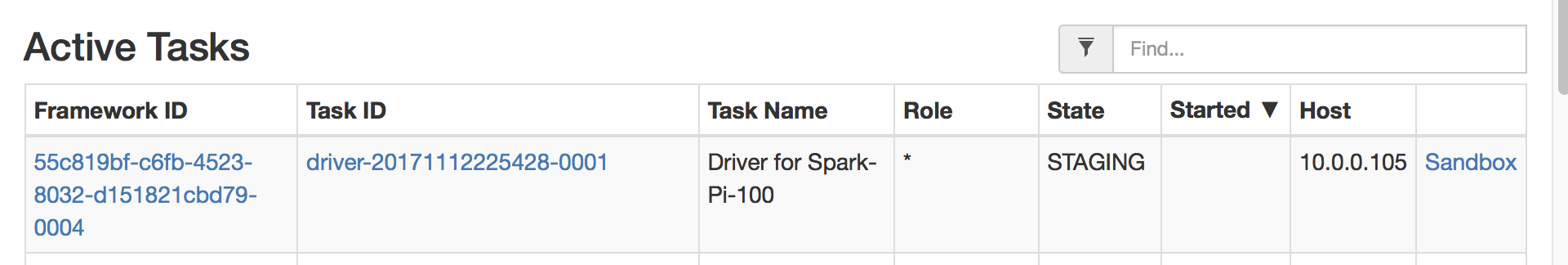
When all installation steps complete your services list will look something like the image below (of course, any services you already had should also appear in the following services list).



# Running your first Spark program through Livy

import json, pprint, requests, textwrap, time, json  
host="http://34.215.224.225:10000"  
  
# "file": "http://downloads.mesosphere.com/spark/assets/spark-examples\_2.10-1.4.0-SNAPSHOT.jar",  
data = {  
 "name":"Spark-Pi-100",  
 "file": "https://github.com/markfjohnson/dcos\_Livy\_server/raw/master/spark-examples\_2.11-2.1.1.jar",  
 "className" : "org.apache.spark.examples.SparkPi",  
 "executorMemory": "2g",  
 "args": ["10"],  
 "conf":{  
 "spark.mesos.executor.docker.image":"mesosphere/spark:2.1.0-2.2.0-1-hadoop-2.6",  
 "spark.logConf":"true",  
 "spark.mesos.executor.docker.forcePullImage":"true"  
 }}  
pp = pprint.PrettyPrinter(indent=4)  
headers = {'Content-Type': 'application/json'}  
r = None  
r = requests.post(host + '/batches', data=json.dumps(data), headers=headers)  
print(r.json())  
initial\_state = json.loads(r.content)['state']  
loc = r.headers['location']  
while initial\_state=='starting' or initial\_state=='running':  
 session\_url = host + loc  
 r = requests.get(session\_url, headers=headers)  
 initial\_state = json.loads(r.content)['state']  
 print(r.json())  
  
r = requests.get(session\_url, headers=headers)  
pp.pprint(r.json())

* Set the ‘host’ variable in the sample program above with the DC/OS’s cluster public IP address and the port number you set when installing the Livy service.
* Description of Livy Batch JSON attributes: (https://livy.incubator.apache.org/docs/latest/rest-api.html)
  + The ‘name’ json attribute cannot have any special characters only ‘-‘ to separate words
  + The ‘args’ json attribute contains a list of all of your Spark program’s attributes. For example, in the standard SparkPI example 10 represents the number of iterations to run for this demo program.
  + The ‘conf’ json attribute contains a list of Spark configuration properties.



# Scaling Apache Livy

# Dependent Resources