**Docker Autoscaling Demo Script**

All scripts are in the scripts directory

1. If the demo as been run previously, cleanup all the containers:   
   $ ./removecontainers.sh
2. Show that there are no containers running:  
   $ docker ps
3. In a browser show:  
   http://[docker host]:8080/status.html  
   There will be an error in the browser because NGINX PLUS is not yet running. Continue to show status.html after each action that effects the NGINX Plus configuration.
4. Setup the environment:  
   $ ./createenv.sh  
   This will create 1 NGINX PLUS load balancer container, 1 NGINX PLUS web server container 1 upstream server and 1 Elasticsearch container and upstream server.
5. Show that there are now containers running:  
   $ docker ps
6. Create some more Elasticsearch containers:  
   $ ./addes [number of containers]
7. Remove some Elasticsearch containers:  
   $ ./removes [number of containers]
8. Create some more NGINX web server containers:  
   $ ./addnginxws [number of containers]
9. Remove some NGINX web server containers:  
   $ ./removnginxws [number of containers]
10. In a browser show:  
    http://[Docker Host]  
    And see that requests are load balanced across the NGINX web server backends.
11. Enable session persistence:  
    $ ./persiston.sh
12. In a browser show:  
    http://[Docker Host]  
    And see that requests are now all sent to a single NGINX web server.
13. Disable session persistence:  
    $ ./persistoff.sh
14. In a browser show:  
    http://[Docker Host]  
    And see that requests are now being load balanced across the NGINX web server backends again.
15. Show script createenv.sh. This script calls the following 3 scripts.
16. Show script addnginxlb.sh. This script creates a container with NGINX PLUS configured as a load balancer with an upstream for NGINX PLUS as a web server and one for Elasticsearch.
17. Show script addnginxws.sh. This script adds a container with NGINX Plus, using the default configuration and adds it as an upstream server. If copies the html content from the Docker host.
18. Show script addes.sh. This script adds a container with Elasticsearch and adds it as an upstream server.
19. Show script addnode.sh. This script is called by the previous 2 scripts and does the actual container creation and upstream server additions.
20. Show script removenginxws.sh. This script removes 1 or more NGINX Plus nodes and their containers.
21. Show ../nginx\_config/docker.conf.
22. Show ../docker\_base/Dockerfile.
23. Show ../docker\_lb/Dockerfile.
24. Show ../docker\_ws/Dockerfile.
25. Generate load:   
    $ ./runsiege.sh.
26. Enable auto scaling:  
    $ ./autoscale.py  
    This program will try and keep the requests per second per active node to between 10 and 12. It will add at most 4 nodes on scale up and remove at most 2 nodes on scale down. It is checking every 2 seconds. There can be a maximum of 10 total nodes and a minimum of 2 active nodes.
27. Cause a web server to fail the health check:  
    $ ./seterror.sh [port]  
    For one or more NGINX Plus web server instances. The autoscaling algorithm will do the request rate per node calculation on the backends that are healthy.
28. Show autoscale.py.
29. Stop the load generator when there are more then 2 active nodes. The autoscaling program will scale down to 2 active nodes.
30. Cause a unhealthy web server to pass the health check:  
    $ ./fixerror [port]  
    As nodes return to health, more backends will be removed to maintain the minimum of 2 active nodes.