Using the Phoronix Test Suite as Docker containers

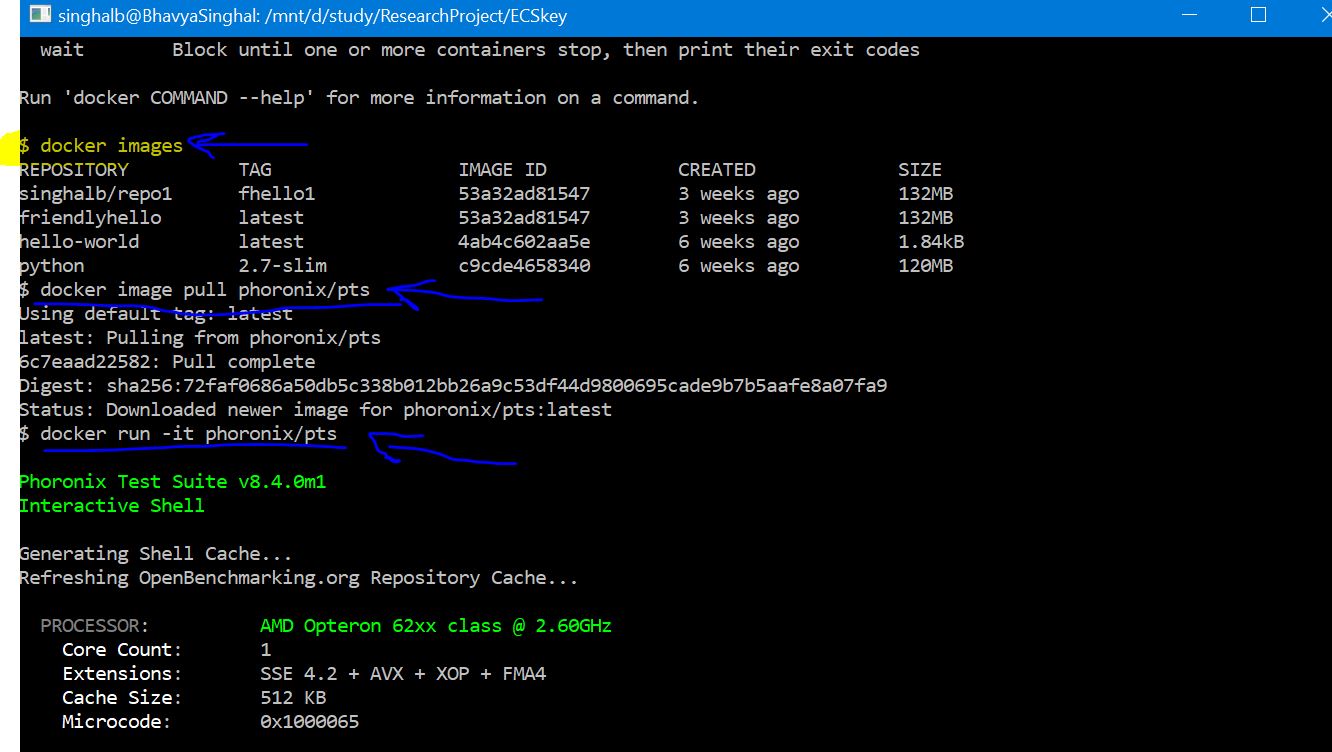
Note: This guide assumes the standard infrastructure used (with docker and docker compose installed) for experiments.

# Run Phoronix Test Suite as a single Docker container

Step 1: docker image pull phoronix/pts

**Step 2: docker run -it phoronix/pts**

This will allow you to enter the container and use the test suite with regular PTS commands.



# Commands to enter a running container at a later time

1. In the VM: docker ps
2. Docker exec -it <container name> /bin/bash

# Important terminologies

* Service: Services are really just “containers in production.” A service only runs one image, but it codifies the way that image runs—what ports it should use, how many replicas of the container should run so the service has the capacity it needs, and so on. Scaling a service changes the number of container instances running that piece of software, assigning more computing resources to the service in the process.
* Stack: A stack is a group of interrelated services that share dependencies and can be orchestrated and scaled together. A single stack is capable of defining and coordinating the functionality of an entire application (though very complex applications may want to use multiple stacks).

# After Creating Swarm Cluster: Orchestrating pts containers with swarm

1. Create image of pts suite and push to docker hub:

$ docker login

$ docker tag phoronix/pts singhalb/container-repo:part1

$ docker push singhalb/container-repo:part1

$ docker run -it singhalb/container-repo:part1

1. Now time to deploy that image to multiple nodes:

On manager, create docker-compose.yml file:

version: "3"

services:

web:

# replace username/repo:tag with your name and image details

image: singhalb/container-repo:part1

deploy:

replicas: 2

resources:

limits:

cpus: "0.1"

memory: 50M

restart\_policy:

condition: on-failure

networks:

- webnet

networks:

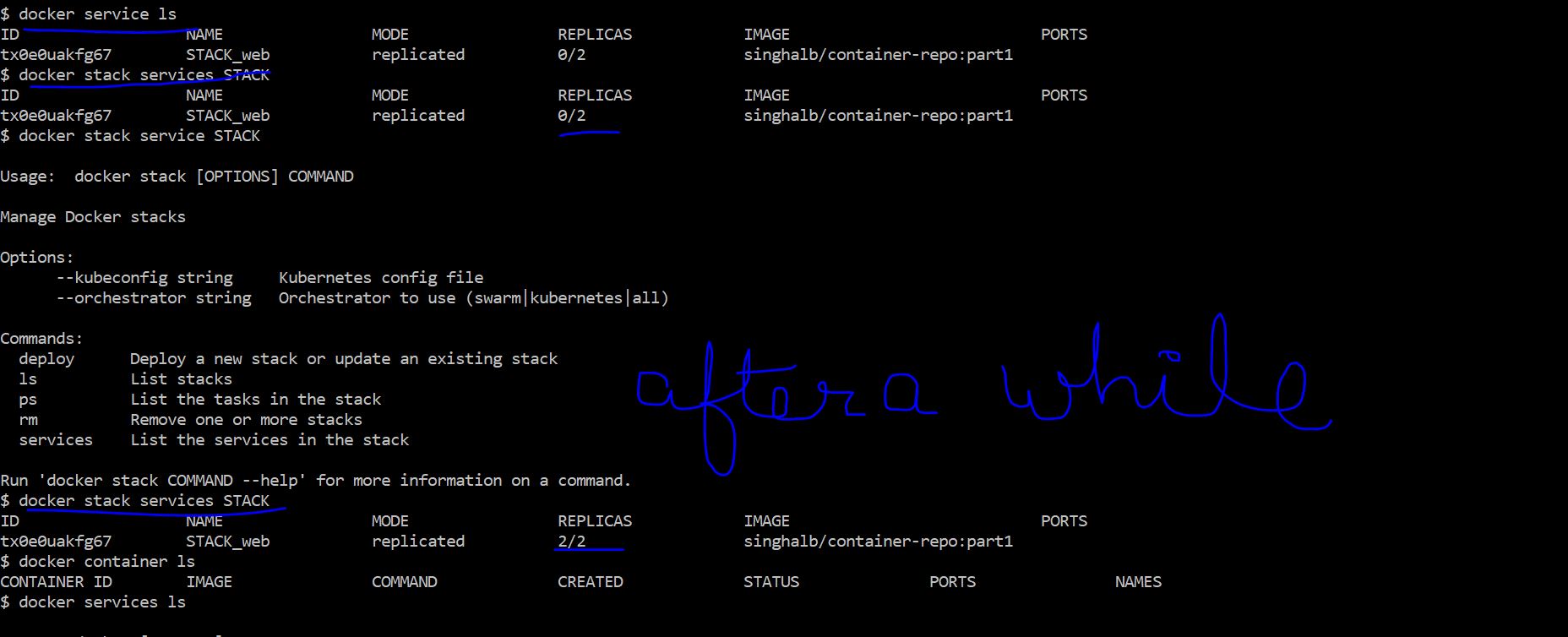
webnet:

1. From that folder, run (to deploy a stack of services running pts containers):

docker stack deploy --compose-file docker-compose.yml STACK

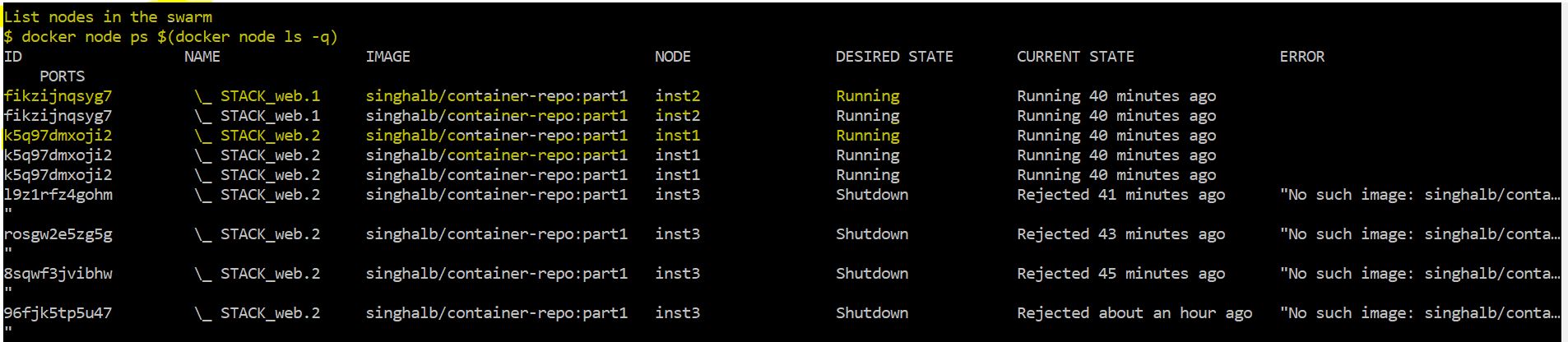
1. This will deploy the 2 replicas of the image to 2 worker nodes.
2. Check in a while via:

docker stack services STACK



1. Check individual containers running via:

docker node ps $(docker node ls -q)



1. Scale service/replicas:

docker service scale SERVICENAME=UNITS

# Commands to use the Phoronix Test Suite

After the swarm cluster is setup and services are running, now is the time to run actual tests. The following are the most important commands used for this project.

1. To view the list of currently available suites, use the following command:

*# phoronix-test-suite list-available-suites*

1. The above command will pull the latest suite info from the Phoronix website. Additionally, the list of all available tests can be found using this command:

*# phoronix-test-suite list-available-tests*

1. The easiest way to get started testing is to simply look at the list of tests, pick one, and run it. The simplest way to do this is to use the following commands:

# phoronix-test-suite benchmark <TEST NAME>

For example,

# phoronix-test-suite benchmark pts/stream

Run Memory Test Suite

# phoronix-test-suite benchmark pts/memory

Run CPU Test Suite

#phoronix-test-suite benchmark pts/cpu

Run Disk Test Suite

#phoronix-test-suite benchmark pts/disk

**Note**: Before running a test, PTS will install required dependencies to run those tests. It will ask you to provide a name for the test run and also will ask permission to save test once it is done, before actually starting to run the test. Some test takes a long time to run (8-13 hours), and hence you can log in later and check the results later via saved results. PTS also gives you the option to upload test results to the benchmarking site, to compare it with other uploaded results.

1. To list saved results:

*# phoronix-test-suite list-saved-results*

1. To view saved result:

*# phoronix-test-suite result-file-to-csv [savedtestname]*

*# phoronix-test-suite result-file-to-csv streamtestrun*

1. Other important commands can be found here: *https://wiki.mikejung.biz/Phoronix\_Test\_Suite*

# References

* <https://wiki.mikejung.biz/Phoronix_Test_Suite>