Docker Blog

Categories: **General (/)** **Engineering (/category/engineering)** **Community (/community-content/)**

November 16, 2015

Scale Testing Docker Swarm to 30,000 Containers

By **Andrea Luzzardi (http://blog.docker.com/author/andrea/)** - Posted in Docker (http://blog.docker.com/category/community/) - Tagged with clustering (/tag/clustering/), docker swarm (/tag/docker-swarm/), dockercon (/tag/dockercon/), DockerCon Europe (/tag/dockercon-europe/)

1,000 nodes, 30,000 containers, 1 Swarm manager

Swarm is the easiest way to run Docker app in production. It lets you take an an app that you've built in development and deploy it across a cluster of servers. Recently we took Swarm out beta and released version 1.0 (http://blog.docker.com/2015/11/swarm-1-0/). It's being used by people

like O'Reilly for building authoring tools (https://getcarina.com/blog/oreilly-carina/), the Distributed Systems Group at Eurecom for doing scientific research (http://zoe-analytics.eu/), and Rackspace who built their new container service, Carina, on top of it (https://getcarina.com/).

(http://blog.docker.com/2015/11/swarm-1-0/)But there's an important thing that Swarm needs to be able to do to take your apps to production: it needs to scale. We believed Swarm could scale up tremendously, so we looked around for a benchmark and found one here (http://blog.kubernetes.io/2015/09/kubernetes-performance-measurements-and.html). We decided to recreate the Kubernetes test with Swarm. Like the team at Google, we wanted to make sure that as we launched more containers it would keep scheduling containers quickly.

What did we measure?

We wanted to stress test a single Swarm manager, to see how capable it would be, so we used one Swarm manager to manage all our nodes. We placed fifty containers per node. Commands were run 1,000 times against Swarm and and we generated percentiles for 1) API Response time and 2) Scheduling delay. We found that we were able to scale up to 1,000 nodes running 30,000 containers. 99% of the time each container took less than half a second to launch. There was no noticeable difference in the launch time of the 1st and 30,000th container.

We used docker info to measure API response time, and then used `docker run -dit ubuntu bash` to measure scheduling delay.

Specs

- Discovery backend: Consul
- 1,000 nodes
- 30 containers per node
- Manager: AWS m4.xlarge (4 CPUs, 16GB RAM)
- Nodes: AWS t2.micro (1 CPU, 1 GB RAM)
- Container image: Ubuntu 14.04

Results

Percentile	API Response Time	Scheduling Delay
50th	150ms	230ms
90th	200ms	250ms

99th	360ms	400ms

We'll continue to test Swarm, pushing its limits and using the results to harden it. If you want to test it out yourself, we've released our swarm-bench code on GitHub (https://github.com/aluzzardi/swarm-bench). If you want to learn more about Swarm, check out the documentation (https://docs.docker.com/swarm/), and go to our forums if you need any help (https://forums.docker.com/).

Learn More about Docker

- New to Docker? Try our 10 min online tutorial (https://docker.com/tryit/)
- Share images, automate builds, and more with a free Docker Hub account (http://hub.docker.com/)
- Read the Docker 1.9 Release Notes (http://docs.docker.com/v1.9/release-notes/)
- Subscribe to Docker Weekly (https://www.docker.com/subscribe_newsletter/)
- Register for upcoming Docker Online Meetups (http://www.meetup.com/Docker-Online-Meetup/)
- Attend upcoming Docker Meetups (https://www.docker.com/community/meetups/)
- Register for DockerCon 2015 Europe (http://europe.dockercon.com/)
- Start contributing to Docker (https://docs.docker.com/contributing/contributing/)

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

Comment

Submit Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.

Subscribe (/feed/)

Sign up for Docker Weekly:

Enter your email

Submit

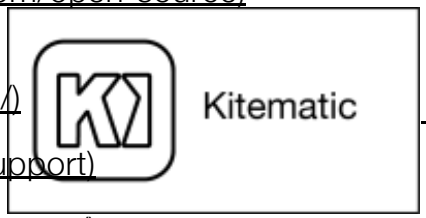
aws (http://blog.docker.com/tag/aws-2/), conference (http://blog.docker.com/tag/conference/), containers (http://blog.docker.com/tag/containers/), contributors (http://blog.docker.com/tag/contributors/), developers (http://blog.docker.com/tag/developers/), devops (http://blog.docker.com/tag/devops/), dgab (http://blog.docker.com/tag/dgab/), Distributed Apps (http://blog.docker.com/tag/distributed-apps/), docker compose, (http://blog.docker.com/tag/docker-compose/) Docker Content Trust (http://blog.docker.com/tag/docker-content-trust/), Docker engine (http://blog.docker.com/tag/docker-engine/), docker hub (http://blog.docker.com/tag/docker-hub/), docker machine (http://blog.docker.com/tag/docker-machine/), docker swarm (http://blog.docker.com/tag/docker-swarm/), Docker Trusted Registry

(http://blog.docker.com/tag/docker-trusted-registry/), dockercon (http://blog.docker.com/tag/dockercon/), dockerfile (http://blog.docker.com/tag/dockerfile/), Dockerhackday (http://blog.docker.com/tag/dockerhackday/), golang (http://blog.docker.com/tag/golang/), hackathon (http://blog.docker.com/tag/hackathon/), hacks (http://blog.docker.com/tag/hacks/), kitematic (http://blog.docker.com/tag/kitematic/), linux (http://blog.docker.com/tag/linux/), lxc (http://blog.docker.com/tag/lxc/), Meetup (http://blog.docker.com/tag/meetup/), Mesos (http://blog.docker.com/tag/mesos/), microservices (http://blog.docker.com/tag/microservices/), Microsoft (http://blog.docker.com/tag/microsoft/), networking (http://blog.docker.com/tag/networking/), Notary (http://blog.docker.com/tag/notary/), official repos (http://blog.docker.com/tag/official-repos/), online meetup (http://blog.docker.com/tag/online-meetup/), open source (http://blog.docker.com/tag/open-source/), openstack (http://blog.docker.com/tag/openstack/), orchestration (http://blog.docker.com/tag/orchestration/), registry (http://blog.docker.com/tag/registry/), runC (http://blog.docker.com/tag/runc/), security (http://blog.docker.com/tag/security/), Windows (http://blog.docker.com/tag/windows/)

Featured Projects

(http://docs.docker.com/userguide/)

(http://docs.docker.com/registry/)



(https://kitematic.com/)

(http://docs.docker.com/machine/)

Get Started (http://docs.docker.com/mac/started/)

(http://docs.docker.com/swarm/) Compose (http://docs.docker.com/compose/)

Careers

(https://docker.com/careers)

Status

(http://status.docker.com/) Networking (https://github.com/docker/libnetwork)

Security

Follow Docker on Twitter (https://docker.com/twitter-security)

Legal

(https://docker.com/legal)

Contact Sales

(http://goto.docker.com/sales-inquiry.html)

Connect

Subscribe to our newsletter

Enter your email

Submit

(http://blog.docker.com/) (https://www.facebook.com/docker.run)

(https://plus.google.com/u/0/communities/108146856671494713993)

(http://www.slideshare.net/docker) (http://twitter.com/docker)

(http://www.youtube.com/user/dockerrun)

(https://www.linkedin.com/company/docker) (http://www.reddit.com/r/docker)