Interconnecting containers at scale with NGINX

dockercon 15

SF JUNE 22-23

@sarahnovotny
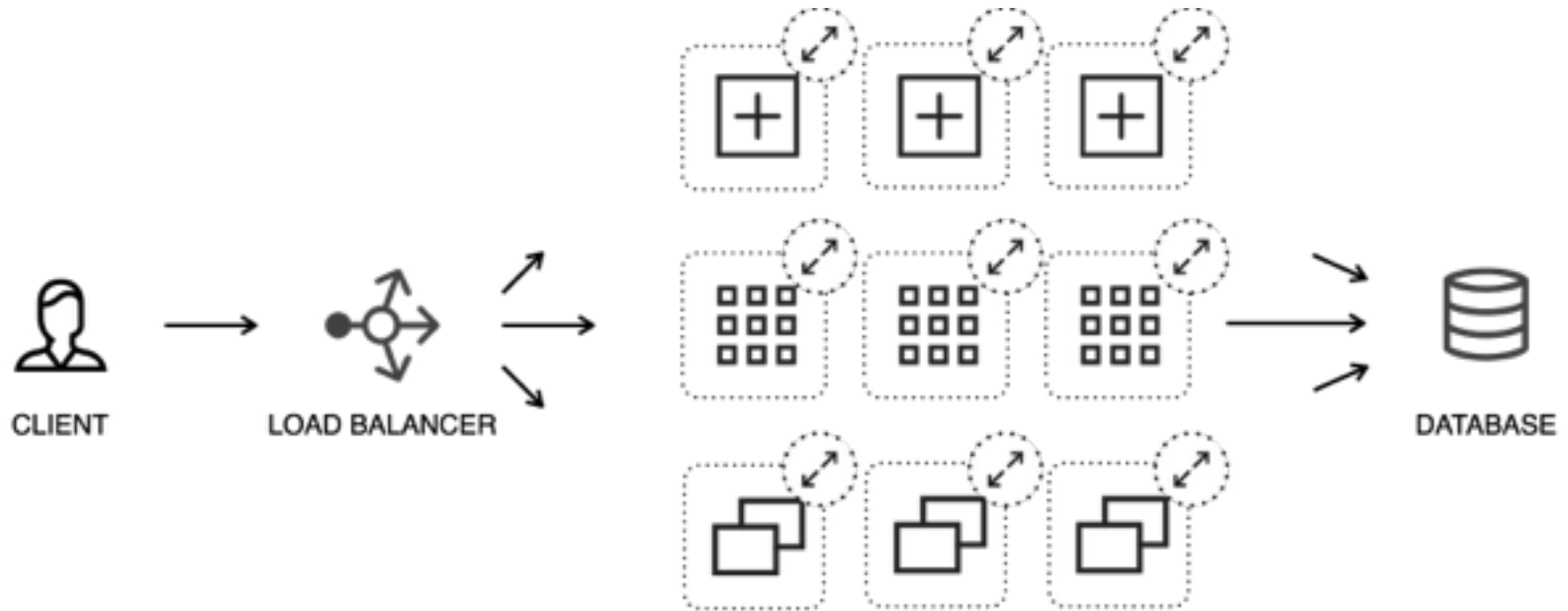
#dockercon

NGIИX

Building a great application is only half the battle, **delivering the application is the other half.**

Microservices

EACH SERVICE CAN BE SCALED INDEPENDENTLY
SERVICE IS THE SCALING UNIT

1)

The good things

Microservices + containers

The Microservices Architecture:
Enables continuous delivery, rapid deployment and elasticity

The Microservices Architecture
To scale, you can scale each service independently

The Microservices Architecture: Services can be written in different languages using different frameworks

Making Changes:
Each service can be changed, tested and built independently

Making Changes:
Release cycles can be dramatically shortened

Deploying Microservices:
To unlock the potential of microservices you must embrace containers, cloud and DevOps

2)

The bad things

The Microservices Architecture:
Services can be written in different languages using different frameworks

Making Changes:
Release cycles can be dramatically shortened

Complexity:
To unlock the potential of microservices you must embrace containers, cloud and DevOps

Here be dragons

3)

How can NGINX help?

NGINX as your stevedores:

Acts as a HTTP router inspecting requests and deciding how each one should be satisfied
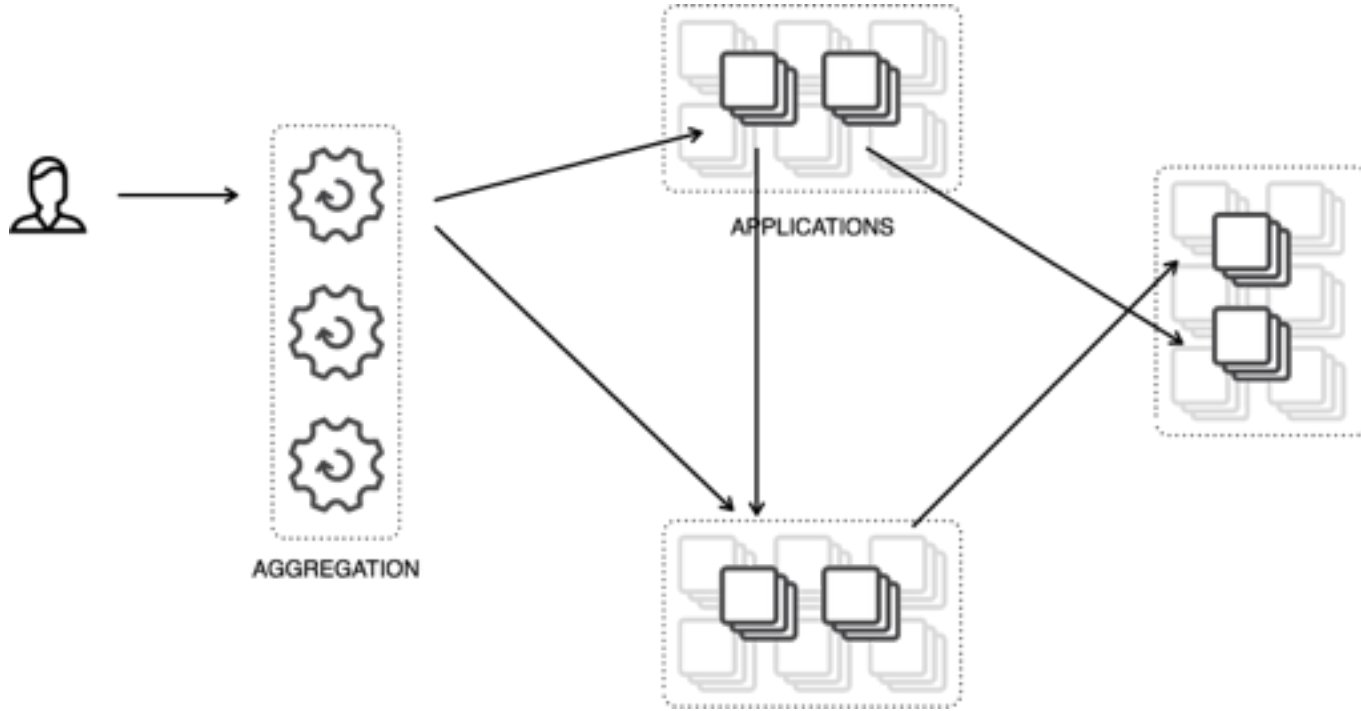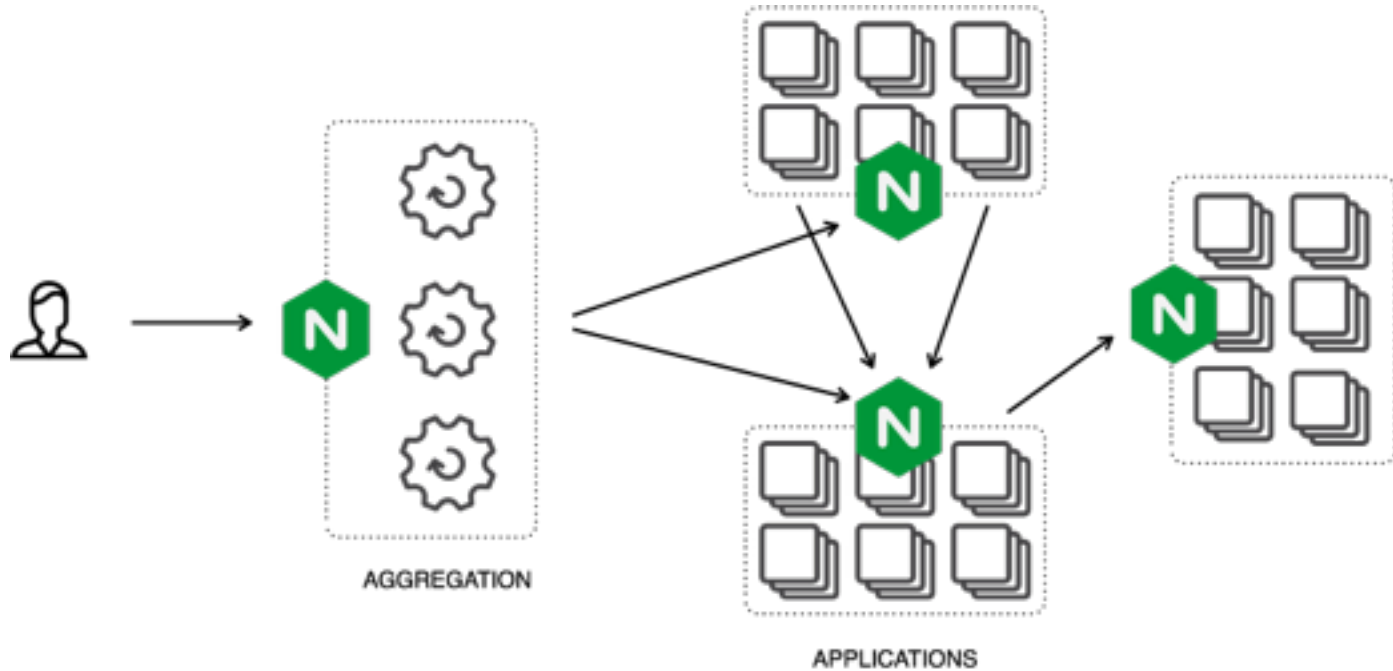
ste·ve·dore

ˈstēvəˌdôr/

noun

noun: stevedore; plural noun: stevedores

a person employed, or a contractor engaged, at a dock to load and unload cargo from ships.

The Microservices Architecture

The Microservices Architecture

AGGREGATION

APPLICATIONS

Docker orchestration

- many options vying
- (and then there's service discovery)

docker-compose

This .yml file builds –

Consul for service discovery

Registrator

tutum/hello-world

Google/golang-hello

& NGINX Plus

```
$ sarah@ubuntu:~/service-discovery$ more docker-compose.yml
nginx:
  build: ./nginxplus
  links:
  - consul
  ports:
  - "9050:80"
  - "8080:8080"

consul:
  command: -server -bootstrap -advertise 10.0.2.15
  image: progrium/consul:latest
  ports:
  - "8300:8300"
  - "8400:8400"
  - "8500:8500"
  - "8600:53/udp"
```

```
registrator:
  command: consul://consul:8500
  image: progrium/registrator:latest
  links:
  - consul
  volumes:
  - "/var/run/docker.sock:/tmp/docker.sock"

service1:
  image: tutum/hello-world:latest
  environment:
    SERVICE_80_NAME: http
    SERVICE_NAME: service1
    SERVICE_TAGS: production
  ports:
  - "80"
```

```
service2:
  image: google/golang-hello:latest
  environment:
    SERVICE_80_NAME: http
    SERVICE_NAME: service2
    SERVICE_TAGS: production
  ports:
  - "8080"
sarah@ubuntu:~/service-discovery$
```

```
sarah@ubuntu:~/service-discovery$ docker-compose build
consul uses an image, skipping
Building nginx...
Step 0 : FROM ubuntu:14.04
 ---> 6d4946999d4f
Step 1 : MAINTAINER NGINX Docker Maintainers "docker-maint@nginx.com"
 ---> Using cache
 ---> 339d0f20dc6e


●●●

sarah@ubuntu:~/service-discovery$ docker-compose up -d
Recreating servicediscovery_consul_1...
Recreating servicediscovery_nginx_1...
Recreating servicediscovery_registrator_1...
Recreating servicediscovery_service1_6...
Recreating servicediscovery_service2_1...
sarah@ubuntu:~/service-discovery$
```

```
sarah@ubuntu:~/service-discovery/nginxplus$ more nginx.conf
{{range services}} {{$name := .Name}} {{range $tag, $service := service .Name |
byTag}} {{if eq $tag "production"}}
upstream {{$name}} {
  zone upstream-{{$name}} 64k;
  least_conn;
  {{range $service}}server {{.Address}}:{{.Port}} max_fails=3 fail_timeout=60
weight=1 slow_start=60;
  {{else}}server 127.0.0.1:65535; # force a 502{{end}}
} {{end}} {{end}} {{end}}

<snip>
```

```
sarah@ubuntu:~/service-discovery$ docker-compose scale service1=5
Creating servicediscovery_service1_2...
Creating servicediscovery_service1_3...
Creating servicediscovery_service1_4...
Creating servicediscovery_service1_5...
Starting servicediscovery_service1_2...
Starting servicediscovery_service1_3...
Starting servicediscovery_service1_4...
Starting servicediscovery_service1_5...
```

Visit our table to go through this demo and request a developer license for NGINX Plus

Thank you

@sarahnovotny

#dockercon

dockercon 15