AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

Docker Networking

Henrik Bærbak Christensen

AARHUS UNIVERSITET

Port forwarding

- Port forwarding is brilliant for local development / manual testing

```
csdev@m51:~$ docker run -d -p 6777:6777 henrikbaerbak/quote:msdo
70e6e436cbc0823da888736d7387e99512edaae2e4cb28c9ac195659454a0c32
csdev@m51:~$ http localhost:6777/msdo/v1/quotes
HTTP/1.1 200 OK
Content-Type: application/json
Date: Fri, 14 Feb 2020 09:06:43 GMT
Server: Jetty(9.4.z-SNAPSHOT)
Transfer-Encoding: chunked

{
    "authors": [
        "Albert Einstein",
        "Søren Kierkegaard",
        "Winston Churchill",
        "Alexander Graham Bell",
        "Plato"
    ],
    "published": "2020-02-14T09:06:34.811Z",
    "title": "MSDO Quote Service",
    "totalItems": 67,
    "url": "http://localhost:6777/msdo/v1/quotes"
}
```

AARHUS UNIVERSITET

Port forwarding

- The liabilities of port forwarding
  - You expose the port on the host machine
    - (and docker opens that port in the firewall!)
  - And you cannot tie two services together using port forwarding!
    - Say start ‘SkyCave’ with a local ‘QuoteService’
    - Scenario
      - I start QuoteService container and port map so host has localhost:6777 access
      - I start SkyCave container and portmap on 7777, and CPF is configured to
        localhost:6777 for QuoteService
      - *It doesn’t work!*
- *Why not?*

AARHUS UNIVERSITET

Container network

- You can specify to *use a named container's network*

```
csdev@m51:~$ docker run -d henrikbaerbak/quote:msdo
35a2b65abe357e8a9de81e1f0e852cbe7bc5188bd12864a8894d399b989ef642
csdev@m51:~$ docker run -ti henrikbaerbak/jdk8-gradle bash
root@0ad1980e357a:/# curl localhost:6777/msdo/v1/quotes/13
curl: (7) Failed to connect to localhost port 6777: Connection refused
root@0ad1980e357a:/# exit
exit
csdev@m51:~$ docker run -ti --network container:festive_lichterman henrikbaerbak/jdk8-gradle bash
root@35a2b65abe35:/# curl localhost:6777/msdo/v1/quotes/13
{"author":"Albert Einstein","quote":"Education is what remains after one has forgotten what one has
hool.","number":13}root@35a2b65abe35:/#
```

- *This allows you to 'tie' services*
  - Ex: start 'skycave' container on the quote service's containers network, and use 'localhost:6777' to access

AARHUS UNIVERSITET

Container network

- Still liabilities
  - Tying stuff together using localhost is bad practice

- But you avoid exposing the port to the outside…

AARHUS UNIVERSITET

Host network

- You can also state
  - --network host
- Then the container reuses the host's network
- Similar liabilities

AARHUS UNIVERSITET

However, …

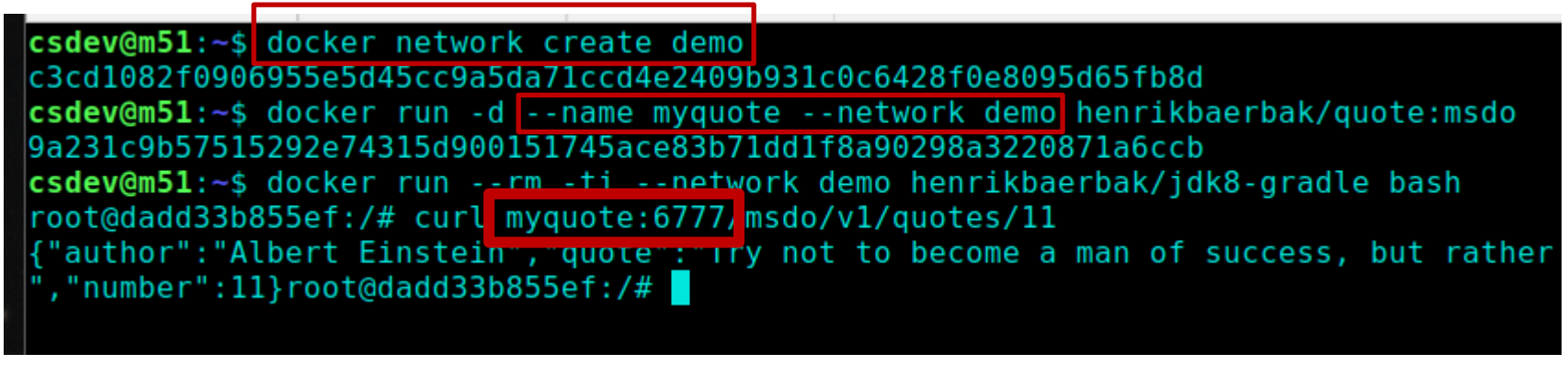
- It sort of fails when we are going to build larger architectures…

- *Docker networks have it's own internal DNS*
  - docker run -- name mydb …
    - Will assign the container the name 'mydb' which can be used to contact it by any container *on the same network*

- Docker can create (and manage) named networks
  - docker network create mynetwork

Production Class Network

- The staging/production environment solution is to define a docker network, and tie containers together there.



- Important: using '--name thename' to define the node's name
  - Similar to giving your machine a name on your home network

AARHUS UNIVERSITET

Restating

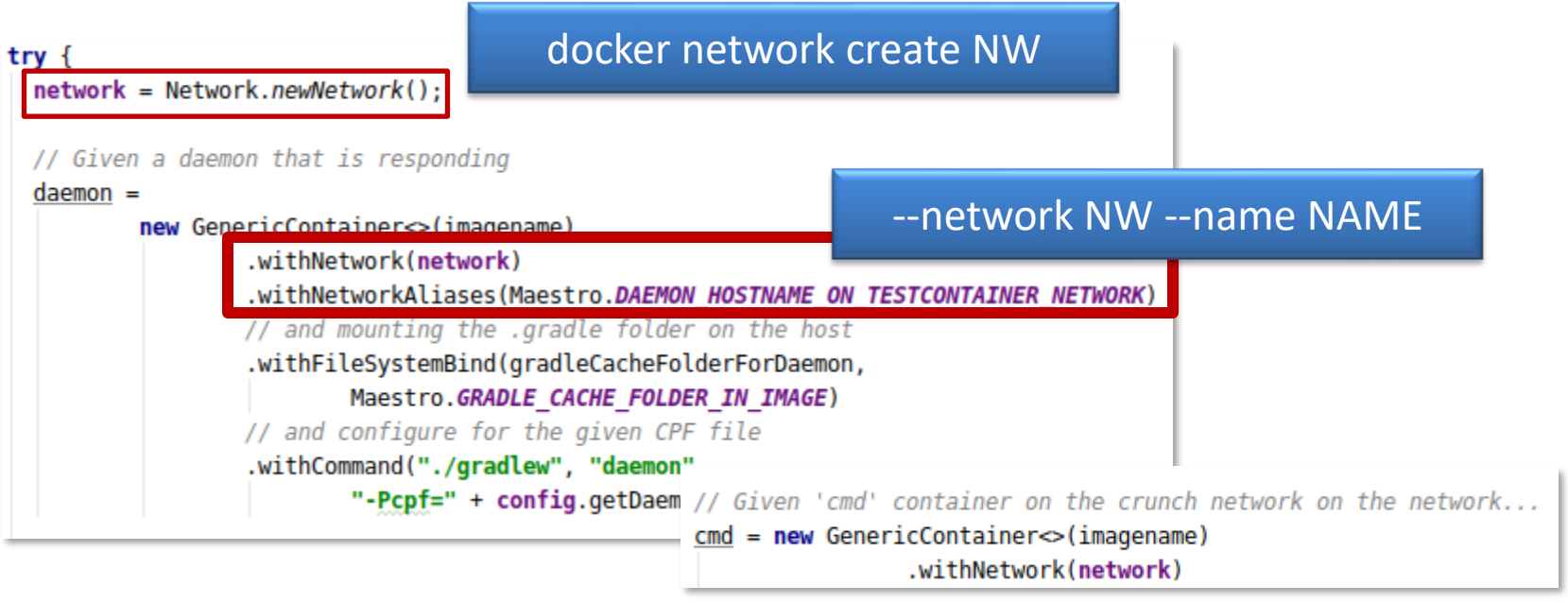
- At the exam in CloudComputing 2016 I made an exam exercise which assumed my students knew that
- It turned out 80-90 % never quite got that point!!!
- *The slaughter house exam*
  - *Everybody went out of the room in the firm conviction that they had just failed the exam ☹*

TestContainers Pendant

AARHUS UNIVERSITET

Network in TC

- TestContainers support creating networks for testing
- A peek into the Crunch interior:



AARHUS UNIVERSITET

Summary

- Docker has a rich set of ways to define and manage networks
  - (And network types – ‘none’, ‘bridge’, ‘overlay’, …)
- Portmapping is great for local experimentation
  - And ‘container:theotherone’ comes in handy sometimes
- For ‘real’ usage, use a docker managed, named, network
  - Swarm uses named networks

- TestContainers support using named networks