

Proxy strežnik za menedžiranje Docker kontejnerjev

Uporabniška dokumentacija

Projektni seminar II

Matic Adamič, Pika Povh
Mentor: Vid Jagodič



Univerza na Primorskem
Fakulteta za matematiko, naravoslovje in informacijske tehnologije

Junij 2022

Kazalo

1	Konfiguracija	2
1.1	Nastavitve strežnika z <i>broccoli-config.xml</i>	2
1.2	Nastavitev strežnika z <i>ports-config.xml</i>	2
2	Izvajanje ukazov	3
2.1	Start	3
2.2	Disconnect	3
2.3	Restart	3
2.4	Stop	4
2.5	Info	4
2.6	Log	5
3	Poganjanje strežnika	5

1 Konfiguracija

Strežnik ob zagonu konfiguriramo z dvema konfiguracijskima datotekama; *broccoli-config.xml* in *ports-config.xml*. Datoteki morata biti v enakem direktoriju, kot skripta, s katero zaganjamo program.

1.1 Nastavitve strežnika z *broccoli-config.xml*

V oznaki "`<auth>`" nastavimo dostop do strežnika, na katerem živijo docker kontejnerji – po navadi je to isti strežnik na katerem zaganjamo program. V atribut "`url`" nastavimo naslov strežnika s kontejnerji, v atribut "`username`" nastavimo uporabniško ime, v "`key_file`" pa pot do našega privatnega ključa. Strežnik na katerega se povezujemo, mora imeti registriran naš javni ključ.

V oznaki "`<user_interface>`" lahko nastavimo možne interakcije s strežnikom. V atributu "`<command_interface>`" nastavimo v atributu "`port`" port pod-strežnika, na katerega bomo pošiljali ukaze. V oznaki "`<logging_interface>`" nastavimo port za Log4J knjižnico v atributu "`log4jPort`", in port na katerega se bo lahko povezal katerikoli uporabnik, v atributu "`clientPort`". Obe komponenti lahko izklopimo ali vklopimo z nastavitvijo atributa "`enabled`", katerega lahko nastavimo na "`true`" ali "`false`".

V atribut "`<default_container_timeout>`" nastavimo po kolikšnem času, bo strežnik pavziral kontejner, če bo ta prižgan in neuporabljen. Dodamo lahko poljubno število "`<container>`" oznak, kamor specificiramo port kontejnerja in atribut "`timeout`", kjer lahko določenem kontejnerju spremenimo privzeto vrednost iz oznake "`<default_container_timeout>`". Vse časovne vrednosti nastavimo s številkami, kateri sledi en izmed znakov:

- s (sekunde)
- m (minute)
- h (ure)
- d (dnevi)

1.2 Nastavitev strežnika z *ports-config.xml*

V oznaki "`<host>`" nastavimo url strežnika, na katerem želimo odpreti porte, ki jih bo uporabljal proxy strežnik. Razpon portov nastavimo v oznaki "`range`", v kateri specificiramo 3 številke:

- "`start`": port
- "`end`": končni port
- "`remap`": vrednost, ki je uporabljena za preslikavo portov – in je eksplicitno določena, izračunana iz vrednosti "`start`" in "`end`".

Primer nastavitve range: "`<range start=\100" end=\200" remap=\50">`", pomeni, da bo strežnik poslušal na portih v razponu [100, 149] in posredoval podatke na porte [150, 199], kjer je pričakovano, da poslušajo kontejnerji.

2 Izvajanje ukazov

Ukaze izvajamo s pomočjo Bash skripte *broc.sh*. Skripto kličemo s parametri:

```
ime_streznika ime_ukaza parametri_ukaza
```

Ime strežnika je lahko krajša oblika, kot je ta specificirana v *ports-config.xml* datoteki v oznaki *jhostz* v atributu "name". Skripto lahko v Linux okolju kličemo direktno, v okolju Windows pa lahko uporabimo [Cygwin](#) (ki omogoča simulacijo Linux okolja). Po namestitvi Cygwin, skripto *broc* premaknemo v direktorij */bin*, ki se nahaja v domačem direktoriju Cygwina (v Windows je privzeta lokacija na "*C:/cygwin/bin/*"). Po prvem zaganjanju izvedemo ukaz:

```
$ export PATH=$PATH:~/bin
```

2.1 Start

Primer ukaza: `$ broc test start 1000 1001`

Z ukazom ustvarimo novega poslušalca na portu 1000, ki bo posredoval podatke na port 4001

```
MaticA@matica ~  
$ broc test start 1000 1001  
[INFO ] Started new port manager on port 1000 forwarding to 1001
```

Slika 1: Primer ustvarjanje novega proxy pod-strežnika.

2.2 Disconnect

Primer ukaza: `$ broc test disconnect 4000`

Z ukazom prekinemo vse trenutno aktivne povezave na portu 4000. Torej vsi klienti, ki so bili povezani ne nek kontejner preko proxy porta 4000, izgubijo povezavo.

```
MaticA@matica ~  
$ broc test disconnect 4000  
[INFO ] Stopped 0 threads that were active handling traffic.
```

Slika 2: Primer ukaza disconnect, na portu trenutno ni bilo nobenih aktivnih povezav.

2.3 Restart

Primer ukaza: `$ broc test restart 4000`

Vsi uporabniki povezani na port 4000 izgubijo povezavo, saj se pod-strežnik ugasne in še enkrat prižge.

2.4 Stop

Primer ukaza: `$ broc test stop 9999`

Želimo ugasniti proxy pod-strežnik, ki posluša na portu 9999, ker tak strežnik ne obstaja, dobimo napako.

```
MaticA@matica ~  
$ broc test stop 9999  
[ERROR] Exception occurred executing command STOP with args [99999]: Requested manager of port 99999, but there isn't one.
```

Slika 3: Primer ukaza stop z napako.

2.5 Info

Primer ukaza: `$ broc test info 4000`

Ukaz izpiše trenutno stanje na pod-strežniku 4000. Izpiše število aktivnih povezav in status pod-strežnika

```
MaticA@matica ~  
$ broc test info 4000  
[INFO ] Number of threads handling traffic: 0  
[INFO ] (number of live connections: 0)
```

Slika 4: Primer ukaza info za specifičen port, trenutno na portu ni aktivnih povezav.

Primer ukaza: `$ broc test info`

Ukaz izpiše trenutno stanje celotnega strežnika in vseh njegovih pod-strežnikov.

```

MaticA@matica ~
$ broc test info
[INFO ] [LogServer, name=LOG4JSrvr-888, uptime=26h 27m 48s, stopflag=false, clients=1, serverPort=888]
[
  [LogServerClient, name=LOG4JClient-1, uptime=26h 27m 39s, stopflag=false]
]
[INFO ] [UserLoggingServer, name=USERLOGSRVR-999, uptime=26h 27m 48s, stopflag=false, clients=0, serverPort=999] -> No clients connected
[INFO ] [CommandServer, name=CMDL-777, uptime=26h 27m 48s, stopflag=false, serverPort=777] -> Running, client-less server
[INFO ] Managers:
[Manager listenPort=4000, forwardPort=4500, container=testc, liveThreads=0, stopper=false]
[Manager listenPort=4001, forwardPort=4501, container=emsid, liveThreads=0, stopper=true]
[Manager listenPort=4002, forwardPort=4502, container=testd, liveThreads=0, stopper=true]
[Manager listenPort=4003, forwardPort=4503, container=testi, liveThreads=0, stopper=true]
[Manager listenPort=4004, forwardPort=4504, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4005, forwardPort=4505, container=emsiu, liveThreads=0, stopper=false]
[Manager listenPort=4006, forwardPort=4506, container=testu, liveThreads=0, stopper=false]
[Manager listenPort=4007, forwardPort=4507, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4008, forwardPort=4508, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4009, forwardPort=4509, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4010, forwardPort=4510, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4011, forwardPort=4511, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4012, forwardPort=4512, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4013, forwardPort=4513, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4014, forwardPort=4514, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4015, forwardPort=4515, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4016, forwardPort=4516, container=emsijk0, liveThreads=0, stopper=null]
[Manager listenPort=4017, forwardPort=4517, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4018, forwardPort=4518, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4019, forwardPort=4519, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4020, forwardPort=4520, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4021, forwardPort=4521, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4022, forwardPort=4522, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4023, forwardPort=4523, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4024, forwardPort=4524, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4025, forwardPort=4525, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4026, forwardPort=4526, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4027, forwardPort=4527, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4028, forwardPort=4528, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4029, forwardPort=4529, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4030, forwardPort=4530, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4031, forwardPort=4531, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4032, forwardPort=4532, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4033, forwardPort=4533, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4034, forwardPort=4534, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4035, forwardPort=4535, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4036, forwardPort=4536, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4037, forwardPort=4537, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4038, forwardPort=4538, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4039, forwardPort=4539, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4040, forwardPort=4540, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4041, forwardPort=4541, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4042, forwardPort=4542, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4043, forwardPort=4543, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4044, forwardPort=4544, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4045, forwardPort=4545, container=null, liveThreads=0, stopper=null]
[Manager listenPort=4046, forwardPort=4546, container=null, liveThreads=0, stopper=null]

```

Slika 5: Primer ukaza info, izpis stanje celotnega strežnika (prikazani niso vsi proxy strežniki, saj jih je preveč)

2.6 Log

Primer ukaza: `$ broc test log`

Ukaz vzpostavi povezavo s strežnikom, preko katere dobimo logiranje strežnika v realnem času, dokler povezavo ne prekinemo sami. Slika 6: Primer ukaza log: povezalo smo se na strežnik, nato je nekdo izvedel ukaz info 4001, nato pa se je nekdo povezal naproxy podstrežnik 4001, in po 5 sekundah zaprl povezavo. Ker je bila to zadnja povezava na kontejner, se je prižgala nit, ki bo po 3600s (1 uri) pavzirala kontejner, če na kontejner v tem času ne bo nove povezave.

3 Poganjanje strežnika

Razpakiramo `broccoli.rar` datoteko. Pomaknemo se v poddirektorij `/bin`. Z administratorskimi pravicami poženemo datoteko `run.sh` z ukazom:

```
$ sudo ./run.sh
```

V primeru, da se strežnik ne štarta pravilno, oziroma je določen port že zaseden, se strežnik ne prižge in nam vrne napako.