

WildFly 8 cluster

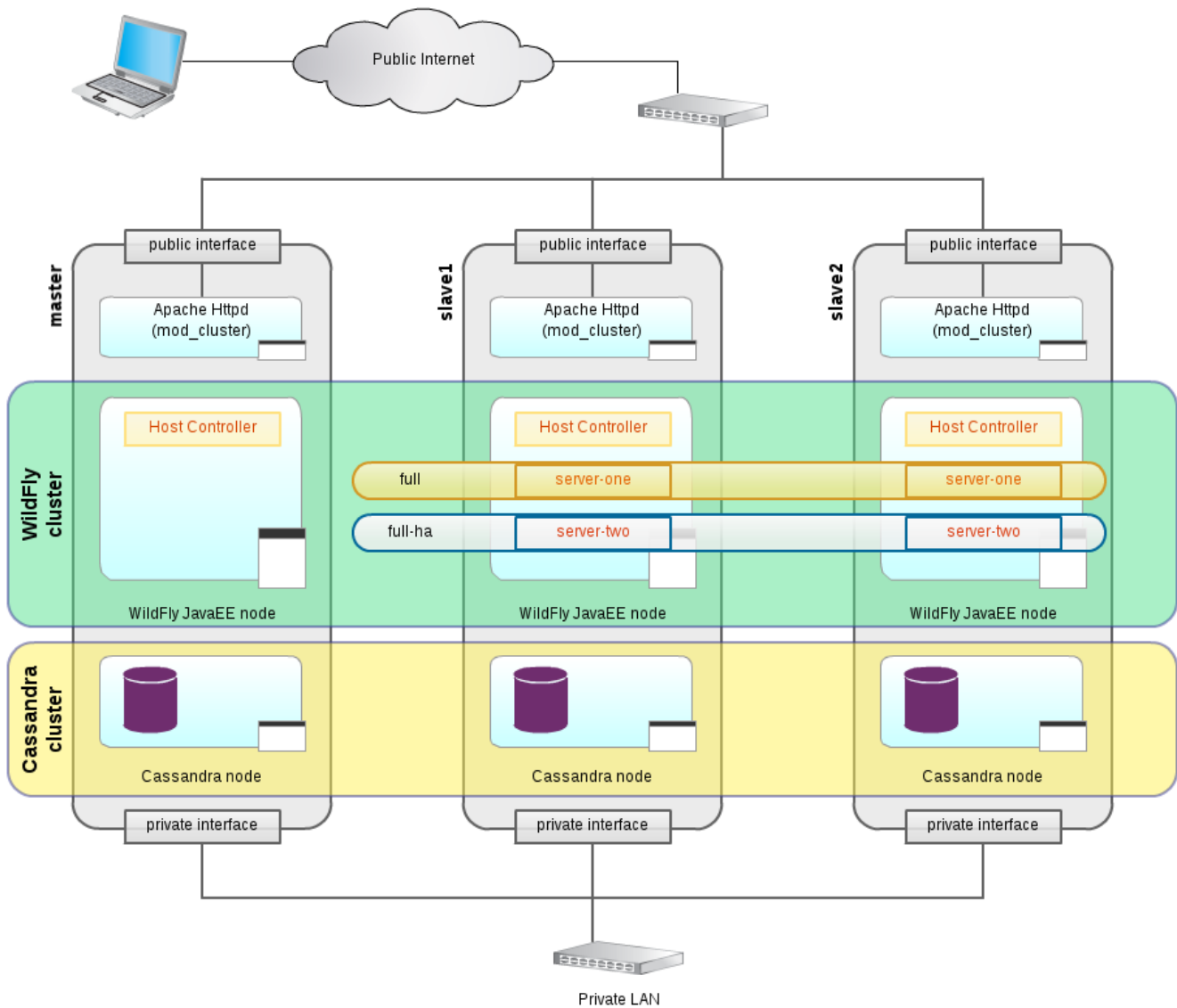
Tartalomjegyzék

- [Tartalomjegyzék](#)
- [Elkészületek](#)
- [Telepítés](#)
- [Hogy lesz ebből cluster?](#)
- [Adminisztráció](#)
 - [CLI csatlakozás](#)
 - [Web felület](#)
- [Multicast helyett TCP!](#)
- [HTTP cluster](#)
 - [Multicast helyett TCP!](#)
 - [Az Apache Httpd és a mod_cluster beállítása](#)

Elkészületek

Ha több WildFly példányt szeretnénk használni, hogy minimalizáljuk a kieséseket, akkor célszerű *domain* módban indítani az alkalmazásszervert, mert a *standalone* mód mellett ez a mód lehetőségeket ad a könnyű adminisztrációra, illetve a beállítások és a telepítések megfelelő terítésére. A WildFly *domain* módja esetén két (viszonylag) sovány igény *java* folyamat indul el a gépeken (a *Process Controller* és a *Host Controller*), amelyeknek felügyelik a tényleges alkalmazásszerver funkciókat, de erre kicsit később még visszatérünk.

A kialakítandó topológia az alábbi ábra szerint alakul majd (kell egy *master* és két *slave* gép), mind a három kiszolgálóra kerül alkalmazásszerver példány, illetve *httpd* kiszolgáló is, amely a *mod-cluster* terheléselosztót fogja kezelni, így bármelyik kiszolgáló esik ki, a szolgáltatás nem fog szünetelni, csak esetleg lassulni. Az egyszer áttekinthettség miatt nem húztam vonalakat az *Apache Httpd* kiszolgálók és a *WildFly JavaEE* példányok közé, mert ott kilenc vonalat kellene húzni, ugyanis mindegyik kapcsolatban vagy mindegyikkel.



i Az ábrán szerepel egy Cassandra cluster is, amely a [Cassandra kezd lépések](#) cikkben leírtak szerint telepítendő és használni is fogjuk a későbbiekben, mint lineárisan skálázható adatbázist. A cél ugyanis az, hogy egy olyan JavaEE környezetet alakítsunk ki, amelyikben a *slave* példányok száma (tervezett módon) tetszőlegesen változtatható a terhelés függvényében, illetve a kialakított architektúra biztosítja a (akár a hardver vagy az egész adatközpont) meghibásodásából következő kiesések **automatikus minimalizálását**.

Majdnem minden komponens legalább duplikálva van, kivéve a *master*: felmerülhet a kérdés, hogy mi történik a *master* kiesése esetén:

- a felhasználó szemszögéből semmi, a többi alkalmazáserver példány dolgozik tovább, a terhelés elosztása automatikusan átszerveződik,
- a rendszergazdai eszköztár viszont jelentősen csökken, amíg helyre nem áll a *master* funkciója (akár mentésből visszaállítva): ez egyfajta SPOF a rendszerben, de vállalható a kockázat.

A három *Apache Httpd* kiszolgáló tetejére érdemes tenni egy hálózati szint terheléelosztót, amelyik csak az három *Apache Httpd* 80-as és 443-as portjait figyeli, és hiba esetén nem oszt oda kérést. Ez a terheléelosztó lehet egy megfelelő redundanciával ellátott hálózati elem, de lehet akár egy földrajzilag elosztott DNS szolgáltatás is, de lehetőség van arra is, hogy csak a *master* gépen lévő *Apache Httpd* kapja a kéréseket és a *master* kiesése esetén irányítja át a forgalmat másik kiszolgáló felé. A lényeg az, hogy előre készre vannak konfigurálva, nem kell időt veszíteni ebben a rétegben való átkonfigurálással vagy programindítgatásokkal.

Telepítés

Hozzuk létre a szervereken egy *wildfly* nevű felhasználót, majd töltsük le és tömörítsük ki a WildFly 8.1 alkalmazáservert:

```
# adduser -g users -m wildfly
# su - wildfly
$ wget http://download.jboss.org/wildfly/8.2.0.Final/wildfly-8.2.0.Final.zip
$ unzip wildfly-8.2.0.Final.zip
...
$
```

Próbaképp elindíthatjuk *domain* módban, hogy minden szükséges függőség rendelkezésre áll-e. (Ha nem indul megfelelően, akkor pótoljuk a hiányosságokat):

```
$ wildfly-8.1.0.Final/bin/domain.sh
...
[Server:server-two] 11:50:37,496 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.1.0.Final
"Kenny" started in 27020ms - Started 210 of 258 services (89 services are lazy, passive or on-demand)
```

Állítsuk le a létrehozott példányokat, fzzünk egy jó kávét, közben olvassunk egy kis dokumentációt a <https://docs.jboss.org/author/display/WFLY8/> oldalról indulva... ☺

Hogy lesz ebből *cluster*?

A telepítés és a futtatási teszt során volt három – **egymástól függetlenül futó** – példányunk, amelyek nem igazán tudtak egymás létezéséről. Ennek két oka volt:

- 127.0.0.1 (avagy localhost) címen futottak
- a legtöbb VPS és/vagy Cloud szolgáltatónál nincs *multicast* a virtualizált gépek között, a WildFly példányok viszont *multicast* protokollon beszélnék meg a topológiát.

Mind a két "problémára" van megoldás, de ezek eltt alakítsuk ki a két *slave* és egy *master* topológiát, amelyhez kétféle módon kell indítanunk a WildFly szervereket:

master

```
$ bin/domain.sh --host-config=host-master.xml -Djboss.bind.address.management=10.129.214.116 -Djboss.bind.address=10.129.214.116
```

slave

```
$ bin/domain.sh --host-config=host-slave.xml -Djboss.domain.master.address=10.129.214.116 -Djboss.bind.address.management=10.129.216.43 -Djboss.bind.address=10.129.216.43
```

Vegyük észre (a különböző privát IP címeken kívül) a két alapvető különbséget:

- a *config-host* paraméternél megadott XML állomány nevét,
- illetve a *slave* esetén a *master* IP címét definiáló *jboss.domain.master.address* paramétert.

Indításkor tapasztalni fogjuk, hogy a *master* példány elindul rendesen, a *slave* viszont hibát fog jelezni és el se indul:

```
[Host Controller] 03:52:36,517 ERROR [org.jboss.remoting.remote.connection] (Remoting "gacivs-test01.javaforum.hu:MANAGEMENT" I/O-1) JBREM000200: Remote connection failed: javax.security.sasl.SaslException: Authentication failed: the server presented no authentication mechanisms
[Host Controller] 03:52:36,528 WARN [org.jboss.as.host.controller] (Controller Boot Thread) JBAS016535: Could not connect to master. No domain controller discovery options left. Error was: java.lang.IllegalStateException: JBAS010942: Unable to connect due to authentication failure.
[Host Controller] 03:52:36,530 ERROR [org.jboss.as.host.controller] (Controller Boot Thread) JBAS010901: Could not connect to master. Aborting. Error was: java.lang.IllegalStateException: JBAS016519: Tried all domain controller discovery option(s) but unable to connect
[Host Controller] 03:52:36,583 INFO [org.jboss.as] (MSC service thread 1-1) JBAS015950: WildFly 8.0.0.Final "WildFly" stopped in 35ms
[Host Controller]
03:52:36,914 INFO [org.jboss.as.process.Host Controller.status] (reaper for Host Controller) JBAS012010: Process 'Host Controller' finished with an exit status of 99
03:52:36,917 INFO [org.jboss.as.process] (Thread-8) JBAS012016: Shutting down process controller
03:52:36,918 INFO [org.jboss.as.process] (Thread-8) JBAS012015: All processes finished; exiting
```

A "Could not connect to master" hiba oka teljesen logikus: a *master* nem engedi, hogy bármilyen jött-ment *slave* csak úgy csatlakozhasson, a csatlakozáshoz a *master* oldalán létre kell hozni egy technikai felhasználót, a *slave* oldalán pedig meg kell adni a hozzá tartozó jelszót. A technikai felhasználó nevét több módon megadhatjuk:

- a *host-slave.xml* fájlban a *host* nevének
- a *host-slave.xml* fájlban a *domain-controller* szekcióban a *remote* paramétereként
- ezek hiányában az adott szerver teljes neve lesz.

Célszerű az utolsó opciót választani, ez jár a legkevesebb munkával, ezért adjunk hozzá kettő új (*gacivs-test01.javaforum.hu* és *gacivs-test02.javaforum.hu*) a *master* adatbázisához":

```
$ bin/add-user.sh
What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): a
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : gacivs-test01.javaforum.hu
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphanumeric character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
- The password should be different from the username
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)
[ ]:
About to add user 'gacivs-test01.javaforum.hu' for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'gacivs-test01.javaforum.hu' to file '/home/wildfly/wildfly-8.0.0.Final/standalone/configuration/mgmt-users.properties'
Added user 'gacivs-test01.javaforum.hu' to file '/home/wildfly/wildfly-8.0.0.Final/domain/configuration/mgmt-users.properties'
Added user 'gacivs-test01.javaforum.hu' with groups to file '/home/wildfly/wildfly-8.0.0.Final/standalone/configuration/mgmt-groups.properties'
Added user 'gacivs-test01.javaforum.hu' with groups to file '/home/wildfly/wildfly-8.0.0.Final/domain/configuration/mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? y
To represent the user add the following to the server-identities definition <secret value="R2FDSVZTMzc5Iw==" />
```

i Ha ugyanazt a jelszót használjuk a *slave* példányokhoz tartozó technikai felhasználóknál, akkor könnyedén tudjuk klónozni a *slave* példányt, így a változó terhelés függvényében könnyen tudunk új tagot hozzátenni a meglévő kiszolgálókhöz és minimális lesz a hibázási lehetőség! Érdemes úgy kialakítani a kiszolgálóinkat, hogy a terhelés növekedésére ne vertikálisan (kevés számú erősebb gép), hanem horizontálisan (sok gyengébb gép) legyen skálázható, ez sokkal célravezetőbb Cloud környezetben, ahol az egyes futó példányokért akár perc alapokon fizetünk, sokkal könnyebb egy meglévő példányt klónozni és elindítani, majd leállítani és törölni, mint a memória, CPU és tárhely hármast hozzáigazítani a terhelési profilhoz.

A fentiek beállításai után a *slave* példányok már tudnak csatlakozni, ha a *secret* XML elemet kicseréljük az *add-user.sh* szkript által kiírt értékre a *host-slave.xml* állomány megfelelő helyén:

```
[Server:server-two] 04:11:27,240 INFO [org.hornetq.ra] (MSC service thread 1-1) HornetQ resource adaptor started
[Server:server-two] 04:11:27,241 INFO [org.jboss.as.connector.services.resourceadapters.ResourceAdapterActivatorService$ResourceAdapterActivator] (MSC service thread 1-1) IJ020002: Deployed: file://RaActivatorhornetq-ra
[Server:server-two] 04:11:27,248 INFO [org.jboss.as.connector.deployment] (MSC service thread 1-2) JBAS010401: Bound JCA ConnectionFactory [java:/JmsXA]
[Server:server-two] 04:11:27,250 INFO [org.jboss.as.messaging] (MSC service thread 1-2) JBAS011601: Bound messaging object to jndi name java:jboss/DefaultJMSConnectionFactory
[Server:server-two] 04:11:27,434 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.0.0.Final "WildFly" started in 31841ms - Started 222 of 349 services (170 services are lazy, passive or on-demand)
```

És a *master* is szépen jelzi, hogy megjött a két *slave* példány:

```
[Host Controller] 04:16:28,559 INFO [org.jboss.as.domain] (slave-request-threads - 1) JBAS010918: Registered remote slave host "gacivs-test01.javaforum.hu", WildFly 8.0.0.Final "WildFly"
[Host Controller] 04:17:18,326 INFO [org.jboss.as.domain] (slave-request-threads - 1) JBAS010918: Registered remote slave host "gacivs-test02.javaforum.hu", WildFly 8.0.0.Final "WildFly"
```

Ha leállítjuk a *master* példányt, akkor a két *slave* mindenképpen megpróbál majd visszatalálni, s ha IP címek helyett a továbbiakban DNS neveket adunk meg, akkor könnyedén át tudunk állni akár egy mentésből visszaállított más IP cím *master* példányra, ha valami baj történne az eredetivel:

```
[Host Controller] 04:18:00,475 WARN [org.jboss.as.host.controller] (Remoting "gacivs-test02.javaforum.hu: MANAGEMENT" I/O-1) JBAS010914: Connection to remote host-controller closed. Trying to reconnect.
...
[Host Controller] 04:22:00,116 INFO [org.jboss.as.host.controller] (domain-connection-threads - 1) JBAS010916: Reconnected to master
```

Adminisztráció

A WildFly alapvetően két módon adminisztrálható:

- CLI (Command Line Interface) felületen át (ez a preferált módszer, mert a későbbiekben könnyen reprodukálható egy környezetet a **ledokumentált és frissen tartott** CLI gytemény futtatásával)
- vagy Web felületről (amelynél jelents kézimunka egy új környezet kialakítása).

Mind a két módszerhez szükséges egy *admin* felhasználó, amelyet a *master* példánynál kell létrehozunk a más ismert *add-user.sh* futtatásával:

```

$ bin/add-user.sh
What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a):
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : admin
The username 'admin' is easy to guess
Are you sure you want to add user 'admin' yes/no? yes
Password recommendations are listed below. To modify these restrictions edit the add-user.properties
configuration file.
  - The password should not be one of the following restricted values {root, admin, administrator}
  - The password should contain at least 8 characters, 1 alphanumeric character(s), 1 digit(s), 1 non-
alphanumeric symbol(s)
  - The password should be different from the username
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)
[ ]:
About to add user 'admin' for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'admin' to file '/home/wildfly/wildfly-8.0.0.Final/standalone/configuration/mgmt-users.properties'
Added user 'admin' to file '/home/wildfly/wildfly-8.0.0.Final/domain/configuration/mgmt-users.properties'
Added user 'admin' with groups to file '/home/wildfly/wildfly-8.0.0.Final/standalone/configuration/mgmt-groups.
properties'
Added user 'admin' with groups to file '/home/wildfly/wildfly-8.0.0.Final/domain/configuration/mgmt-groups.
properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB
calls.
yes/no? y
To represent the user add the following to the server-identities definition <secret value="QWRtaW4zNzkj" />

```

CLI csatlakozás

A WildFly tartalmaz egy *jboss-cli.sh* állományt, amellyel csatlakozni tudunk bármelyik példányhoz, de leginkább a *master* az érdekes:

```

$ bin/jboss-cli.sh
You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for the list of supported
commands.
[disconnected /] connect gacivs-test-master
[domain@gacivs-test-master:9990 /] /host=gacivs-test01.javaforum.hu/server=server-one/interface=public:read-
attribute(name=resolved-address)
{
  "outcome" => "success",
  "result" => "10.129.216.43"
}

```

Vegyük észre, hogy ha arról a gépről csatlakozunk, ahol a WildFly *master* is fut, akkor nem kér jelszót! Egyéb helyről kapcsolódva viszont kérni fogja a felhasználónevet és a jelszót.

Web felület

A WildFly az általunk megadott IP címen fog a 9990 porton indítani egy *console* alkalmazást, amely lehetővé teszi a webes felületen való adminisztrációt, itt láthatjuk a kialakított topológiát és a két szervert, amelyeket az alapértelmezett konfiguráció szerint elindít a WildFly:

JBoss Management console screenshot showing the WildFly 8.0.0.Final interface. The browser address bar shows localhost:9990/console/App.html#topology. The page title is WildFly 8.0.0.Final, and the user is logged in as admin.

The navigation menu includes Profiles, Hosts, Runtime, and Administration. The Administration tab is active, showing the TOPOLOGY view. The left sidebar contains a tree view with categories like Platform, JVM, Environment, Subsystems, Datasources, JPA, JMS Destinations, JNDI View, Transaction Logs, Transactions, and Webservices.

The main content area displays "Hosts, groups and server instances" with a description: "An overview of all hosts, groups and server instances in the domain." A table lists the following data:

Hosts → Groups ↓	gacivs-test01.javaforum.hu Domain: Member	gacivs-test02.javaforum.hu Domain: Member	master Domain: Controller ★
main-server-group Profile: full	server-one ✓ Socket Binding: full-sockets Ports: +0	server-one ✓ Socket Binding: full-sockets Ports: +0	
other-server-group Profile: full-ha	server-two ✓ Socket Binding: full-ha-sockets Ports: +150	server-two ✓ Socket Binding: full-ha-sockets Ports: +150	

At the bottom of the table, there are navigation arrows and the text "1-3 of 3". The footer of the console shows "2.1.1.Final" and "Tools Settings".

Multicast helyett TCP!

Mint említettem volt, a legtöbb VPS és Cloud szolgáltató nem igazán engedi a *multicast* kommunikációt, még akkor se, amikor adnak egy privát IP tartományt egy második hálózati interfészen. A WildFly viszont alapból *multicast* csatornán áll össze egy csoportba, ezért rá kell beszélünk, hogy ezt TCP csatornán mvelje, illetve a betér vagy kies tagokat TCP csatornán tartsa nyilván. Ehhez az alábbi parancsot kell kiadnunk, hogy a "mindenügyi" JGroups TCP-n kapcsolódjon össze:

```

cd /profile=full-ha/subsystem=jgroups
./stack=tcping:add
cd stack=tcping
./transport=TRANSPORT:add(type=TCP,socket-binding=jgroups-tcp)
:add-protocol(type=TCPPING)
:add-protocol(type=MERGE2)
:add-protocol(type=FD SOCK,socket-binding=jgroups-tcp-fd)
:add-protocol(type=FD)
:add-protocol(type=VERIFY_SUSPECT)
:add-protocol(type=BARRIER)
:add-protocol(type=pbcast.NAKACK)
:add-protocol(type=UNICAST2)
:add-protocol(type=pbcast.STABLE)
:add-protocol(type=pbcast.GMS)
:add-protocol(type=UFC)
:add-protocol(type=MFC)
:add-protocol(type=FRAG2)
:add-protocol(type=RSVP)
cd protocol=TCPPING
./property=initial_hosts/:add(value="10.129.216.43[7750],10.129.215.37[7750]")
./property=port_range/:add(value=0)
./property=timeout/:add(value=3000)
./property=num_initial_members/:add(value=3)
cd ../..
:write-attribute(name=default-stack,value=tcping)

```

Figyeljünk arra, hogy a 7750-es port úgy jön ki, hogy 7600 + 150, mert ennyi a porteltolása a *full-ha* profilban a *server-two* csoportnak!

Ezek után a két példány szépen egymásra fog találni:

```

[Server:server-two] 07:18:21,671 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (Incoming-1,
shared=tcping) ISPN000094: Received new cluster view: [gacivs-test01.javaforum.hu:server-two/web|1] (2)
[gacivs-test01.javaforum.hu:server-two/web, gacivs-test02.javaforum.hu:server-two/web]

```

HTTP cluster

A felhasználó oldaláról a HTTP kiszolgálás folyamatossága és folytonossága a szempont, így konfiguráljunk be egy *Apache Httpd* szervert, ehhez a WildFly (régebben ugye JBoss) mellé kiadott *mod_cluster* modulát tudjuk használni.

Elsképp szerezzük meg a *cluster-test* alkalmazást a WildFly dokumentumok között említett helyről ([git clone git://github.com/liweinan/cluster-demo.git](https://github.com/liweinan/cluster-demo.git)), fordítsuk le, majd telepítsük a CLI használatával:

```

[domain@gacivs-test-master:9990 /] deploy cluster-demo.war --server-groups=other-server-group

```

Multicast helyett TCP!

Mivel TCP alapokon fogunk kommunikálni az Apache alá telepített *mod_cluster* moduldal, ezért CLI használatával kapcsoljuk ki a *multicast* hirdetést és konfiguráljuk be a *proxy* példányok IP címét és port számát, majd kapcsoljuk ki a *Sticky Session* szolgáltatást is:


```
[domain@gacivs-test-master:9990 /] /profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:
write-attribute(name=advertise,value=false)
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"other-server-group" => {"host" => {
    "gacivs-test01.javaforum.hu" => {"server-two" => {"response" => {
      "outcome" => "success",
      "result" => undefined,
      "response-headers" => {
        "operation-requires-restart" => true,
        "process-state" => "restart-required"
      }
    }
  }},
  "gacivs-test02.javaforum.hu" => {"server-two" => {"response" => {
    "outcome" => "success",
    "result" => undefined,
    "response-headers" => {
      "operation-requires-restart" => true,
      "process-state" => "restart-required"
    }
  }
}
}}}
}
```

Láthatjuk, hogy a *master* végrehajtotta a *slave* példányokon a kért módosítást (ellenrizni is tudjuk ezt a webes felületen), de kiírta, hogy szükséges lesz az alkalmazáserver példányok újraindítása. Még mielőtt ezt megtennénk, állítsuk be a *proxy* címeket is:

```
[domain@gacivs-test-master:9990 /] /profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:
write-attribute(name=proxy-list,value="10.129.214.116:10001,10.129.216.43:10001,10.129.215.37:10001")
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"other-server-group" => {"host" => {
    "gacivs-test01.javaforum.hu" => {"server-two" => {"response" => {
      "outcome" => "success",
      "result" => undefined,
      "response-headers" => {
        "operation-requires-restart" => true,
        "process-state" => "restart-required"
      }
    }
  }},
  "gacivs-test02.javaforum.hu" => {"server-two" => {"response" => {
    "outcome" => "success",
    "result" => undefined,
    "response-headers" => {
      "operation-requires-restart" => true,
      "process-state" => "restart-required"
    }
  }
}
}}}
}
```

...mert itt se ússzuk meg az újraindítást, de még mindig ne indítsuk újra a kiszolgálókat, mert a Sticky Session kikapcsolásához úgyszólván újra kellene indítani:

```
[domain@gacivs-test-master:9990 /] /profile=full-ha/subsystem=modcluster/mod-cluster-config=configuration/:
write-attribute(name=sticky-session,value=false)
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => {"other-server-group" => {"host" => {
    "gacivs-test01.javaforum.hu" => {"server-two" => {"response" => {
      "outcome" => "success",
      "result" => undefined,
      "response-headers" => {
        "operation-requires-restart" => true,
        "process-state" => "restart-required"
      }
    }
  }},
  "gacivs-test02.javaforum.hu" => {"server-two" => {"response" => {
    "outcome" => "success",
    "result" => undefined,
    "response-headers" => {
      "operation-requires-restart" => true,
      "process-state" => "restart-required"
    }
  }
  }
  }
  }
}
```

...ezért most indítsuk újra, majd ellenrizzük, hogy rendben elindultak-e a kiszolgálók:

```
[domain@gacivs-test-master:9990 /] /server-group=other-server-group:stop-servers
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => undefined
}
[domain@gacivs-test-master:9990 /] /server-group=other-server-group:start-servers
{
  "outcome" => "success",
  "result" => undefined,
  "server-groups" => undefined
}
[domain@gacivs-test-master:9990 /] ls /host=gacivs-test01.javaforum.hu/server-config=server-two
interface          path          auto-start=true          group=other-
server-group       priority=undefined      socket-binding-port-offset=150
jvm                system-property          cpu-affinity=undefined   name=server-
two               socket-binding-group=undefined  status=STARTED
[domain@gacivs-test-master:9990 /] ls /host=gacivs-test02.javaforum.hu/server-config=server-two
interface          path          auto-start=true          group=other-
server-group       priority=undefined      socket-binding-port-offset=150
jvm                system-property          cpu-affinity=undefined   name=server-
two               socket-binding-group=undefined  status=STARTED
```

...ami után persze panaszkodni fognak az egyes példányok, hogy nem tudnak csatlakozni a megadott címre, mert ott még nincs Apache Httpd szerver:

```
[Server:server-two] 05:01:58,840 ERROR [org.jboss.modcluster] (UndertowEventHandlerAdapter - 1)
MODCLUSTER000043: Failed to send INFO to gacivs-test02/10.129.215.37:10001: java.net.ConnectException:
Connection refused
```

Az Apache Httpd és a *mod_cluster* beállítása

Telepítsünk egy *httpd* csomagot, töltsük le a legfrissebb *mod_cluster* kiadást a megfelel (jelen esetben 64 bites CentOS) rendszerhez:

```
# yum install httpd
...
# wget http://downloads.jboss.org/mod_cluster//1.2.6.Final/linux-x86_64/mod_cluster-1.2.6.Final-linux2-x64-so.
tar.gz
...
# cd /etc/httpd/modules/
# tar xzvf /root/mod_cluster-1.2.6.Final-linux2-x64-so.tar.gz
...
#
```

Ezek után kicsit konfigurálnunk kell a `/etc/httpd/conf/httpd.conf` fájl szerkesztésével (**figyelem: diff kimenet!**):

```
192c192
< LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
---
> #LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
202a203,207
> LoadModule slotmem_module modules/mod_slotmem.so
> LoadModule manager_module modules/mod_manager.so
> LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
> LoadModule advertise_module modules/mod_advertise.so
>
1009a1015,1040
>
> Listen 0.0.0.0:10001
> MemManagerFile /var/cache/httpd
>
> <VirtualHost 0.0.0.0:10001>
>   <Directory />
>     Order deny,allow
>     Deny from all
>     Allow from all
>   </Directory>
>
>   <Location /mod_cluster-manager>
>     SetHandler mod_cluster-manager
>     Order deny,allow
>     Deny from all
>     Allow from all
>   </Location>
>
>   KeepAliveTimeout 60
>   MaxKeepAliveRequests 0
>
>   ManagerBalancerName other-server-group
>   AdvertiseFrequency 5
>   ServerAdvertise off
>   EnableMCPMReceive
> </VirtualHost>
```

A `httpd` szolgáltatás elindítása után már láthatjuk **bármelyik** publikus IP címen a három közül a `mod_cluster` státusz és menedzsment felületét (**figyelem! nincs korlátozva a `mod_cluster-manager` elérése, éles üzembe így nem való!**):



mod_cluster/1.2.6.Final

[Auto Refresh](#) [show DUMP output](#) [show INFO output](#)

Node gacivs-test01.javaforum.hu:server-two (ajp://10.129.216.43:8159):

[Enable Contexts](#) [Disable Contexts](#)

Balancer: other-server-group,LBGroup: ,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 1,Ttl: 60000000,Status: OK,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 84

Virtual Host 1:

Contexts:

/cluster-demo, Status: ENABLED Request: 0 [Disable](#)

Aliases:

default-host
localhost

Node gacivs-test02.javaforum.hu:server-two (ajp://10.129.215.37:8159):

[Enable Contexts](#) [Disable Contexts](#)

Balancer: other-server-group,LBGroup: ,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 1,Ttl: 60000000,Status: OK,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 72

Virtual Host 1:

Contexts:

/cluster-demo, Status: ENABLED Request: 0 [Disable](#)

Aliases:

default-host
localhost

A mod_cluster ténykedése okán nem kell azzal törődnünk, hogy felsoroljuk a *Context Path* konfigurációt, az él alkalmazásaink automatikusan megjelennek az *Apache Httpd* kiszolgáló publikus IP címein és mindig az az alkalmazáserver kapja a kérést, amelyik ki tudja szolgálni és pillanatnyilag a legkevesebb kérést kapta:



Hello World!

Hello! The time is now Tue Apr 08 05:38:48 EDT 2014