

## A JUNOS CLI megismerése - alapok

Az SRX sorozatú eszközökben Junos operációs rendszer fut: ez egy FreeBSD alapú rendszer, és mint ilyen, nagyon sok FreeBSD tulajdonságot örökölt, valamint a Juniper rendkívül sok, igazán hasznos funkciót fejlesztett hozzá. Az alábbi kis cikkben a tényleg hasznos funkciókat próbálom - a teljesség igénye nélkül – ismertetni.

### 1. root login

Ha a Junos-ba root userrel lépünk be, akkor FreeBSD shellt (csh-t) kapunk, ezt a %-jel jelzi:

```
root@:RE:0%
```

Ilyenkor a csh-ban tudunk rootként rendszer adminisztrációs parancsokat futtatni, ezzel a résszel most nem foglalkozunk.

Az eszköz konfigurálásához, a CLI elindításához a cli parancs használható:

```
root@:RE:0% cli
root>
```

Ha nem root-ként lépünk be, akkor alaphól a CLI-be jutunk – innen (ha van hozzá jogunk) indíthatunk FreeBSD shell-t az alábbi paranccsal:

```
orosz.tamas> start shell
%
```

Kilépni általánosságban mindenhol az `exit` paranccsal tudunk.

A Junos CLI-ben használhatóak az automatikus kiegészítési lehetőségek (mint a \*\*\*\*X rendszerekben). A kiegészítés a TAB vagy a SPACE billentyűvel lehetséges. A Junos CLI ilyen téren meglepően intelligens: nemcsak a builtin parancsokat lehet vele kiegészíteni, hanem pl. az általunk korábban létrehozott változók neveit is.

### 2. Config felépítése, nézegetése

A Junos-ban az aktuális teljes config a `show configuration` paranccsal lehetséges, a parancs kimenete első ránézésre érdekes, C++ stílus szerint struktúrált kimenetet ad (részlet):

```
services {
    ssh;
    web-management {
        http;
    }
}
```

A config felépítése struktúrális; a legfelső szinten (top) létező objektumoknak vannak alsóbb szintjei, amiknek szintén van, és így tovább – a szintek száma az objektumok típusától függ. A hierarchikus felépítés a {} jelek jelzik. Egy adott szinten lévő config megnézése is lehetséges, nem szükséges átnyálazni az egész konfigot:

```
root# show interfaces ge-0/0/3 unit 0
family inet {
    address 192.168.200.200/24;
}
family ethernet-switching;
```

**Fontos:** a show parancs a configot mutatja edit módban; egyébként az adott objektumról ad infókat (esetünkben ugyanez a parancs):

```
root> show interfaces ge-0/0/3
Physical interface: ge-0/0/3, Enabled, Physical link is Down
  Interface index: 135, SNMP ifIndex: 535
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU
Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues    : 8 supported, 8 maximum usable queues
  Current address: 2c:6b:f5:88:fd:83, Hardware address: 2c:6b:f5:88:fd:83
  Last flapped  : Never
  Input rate    : 0 bps (0 pps)
  Output rate   : 0 bps (0 pps)
  Active alarms : LINK
  Active defects: LINK
Logical interface ge-0/0/3.0 (Index 72) (SNMP ifIndex 536)
  Flags: Device-Down SNMP-Traps Encapsulation: ENET2
  Input packets : 0
  Output packets: 0
  Protocol eth-switch
  Flags: None
```

### 3. Config szerkesztése

A Junos config rendkívül áttekinthető, és a szerkesztést nagyon sok funkció segíti. A szerkesztéshez elsőként konfigurálási módba kell lépünk, ezt a configure paranccsal tehetjük meg, ekkor a shell is megváltozik:

```
root> configure
Entering configuration mode
[edit]
root#
```

A Junos a [ ] jelek között mindig jelzi, melyik aktuális szinten vagyunk; a szintek bejárhatóak teljes egészében, vagy bejárás nélkül szerkeszthetők. Példánkban egy innterfésznek állítunk be IP címet mindkét módszerrel.

## Bejárás:

```
[edit]
root# edit interfaces ge-0/0/3 unit 0 family inet
[edit interfaces ge-0/0/3 unit 0 family inet]
root# set address 192.168.200.200/24
[edit interfaces ge-0/0/3 unit 0 family inet]
root# show
address 192.168.200.200/24;
[edit interfaces ge-0/0/3 unit 0 family inet]
root#
```

Kis magyarázat: egy szint szerkesztése az `edit` paranccsal lehetséges; esetünkben végigjártuk a GE-0/0/3 interfész hierarhiáját (interfész -> unit -> címtípus) majd beállítottuk az IP címet a `set address` paranccsal.

A szintek között felfelé lépegethetünk: 1 szintet az `up` paranccsal, a legfelső szintre (`top`) pedig a `top` paranccsal.

## Szerkesztés bejárás nélkül:

```
[edit]
root# set interfaces ge-0/0/3 unit 0 family inet address 192.168.200.200/24
```

## Szerkesztési segítségek: copy és rename

Nem magyarázom túl: első példánkban sok beállítása van a 192.168.200.200/24 DHCP pool-nak, és azt szeretnénk átmásolni a 192.168.200.222/24-re:

```
root# show
pool 192.168.200.200/24 {
    address-range low 192.168.200.100 high 192.168.100.200;
    domain-name peetandcook.hu;
    name-server {
        192.168.200.2;
        192.168.200.3;
    }
    domain-search {
        pac.hu;
    }
    wins-server {
        192.168.200.2;
    }
    router {
        192.168.200.254;
    }
    boot-server 192.168.200.1;
}
```

```
[edit system services dhcp]
root# copy pool 192.168.200.200/24 to pool 192.168.200.222/24
```

```
[edit system services dhcp]
root# show
pool 192.168.200.200/24 {
    address-range low 192.168.200.100 high 192.168.100.200;
    domain-name peetandcook.hu;
    name-server {
        192.168.200.2;
        192.168.200.3;
    }
    domain-search {
        pac.hu;
    }
    wins-server {
        192.168.200.2;
    }
    router {
        192.168.200.254;
    }
    boot-server 192.168.200.1;
}
pool 192.168.200.222/24 {
    address-range low 192.168.200.100 high 192.168.100.200;
    domain-name peetandcook.hu;
    name-server {
        192.168.200.2;
        192.168.200.3;
    }
    domain-search {
        pac.hu;
    }
    wins-server {
        192.168.200.2;
    }
    router {
        192.168.200.254;
    }
    boot-server 192.168.200.1;
}
```

És módosítjuk a 192.168.200.222/24 poolban a name-server IP-jét:

```
root# rename pool 192.168.200.222/24 name-server 192.168.200.2 to
192.168.222.2
```

```
[edit system services dhcp]
root# show pool 192.168.200.222/24
address-range low 192.168.200.100 high 192.168.100.200;
domain-name peetandcook.hu;
name-server {
    192.168.222.2;
    192.168.200.3;
}
domain-search {
```

```
    pac.hu;
}
wins-server {
    192.168.200.2;
}
router {
    192.168.200.254;
}
boot-server 192.168.200.1;

[edit system services dhcp]
```

#### 4. Commit és exclusive config

A Junos-ban amikor szerkesztjük a config-ot, ahhoz, hogy az életbe lépjen, el kell menteni futási config-ként. Ezt a `commit` paranccsal tehetjük meg.

Ha úgy akarunk kilépni a szerkesztési módból, hogy nem adjuk ki a `commit` parancsot, a rendszer rákérdez, hogy biztosan ki akarunk-e lépni, mert változtattunk a config-on. És itt jön az érdekesség: ha kilépünk, akkor alapértelmezett esetben a Junos nem dobja el a változtatásokat, azokat lehetőségünk van a később lementeni, vagy visszavonni.

Ha egy Junos eszközt több ember konfigurál, akkor érdekes dolgok történhetnek a fentiek miatt.

Példa: emberünk belép, és kiad egy `delete interfaces` parancsot. Rájön, hogy hülyeséget csinált, kilép. Következő user mondjuk 2 hét múlva belép, beállít valamit, majd kiadja a `commit` parancsot – és mivel a `delete interfaces` rekurzív parancs volt, sikeresen ki is törli a `commit` az összes interfész beállítást – hiszen az előző user nem `commit`-elt, ott maradt a szerkesztési sorban a téves parancsa.

Ez a fenti jellegű adminisztratív hiba nagyon gyakori a Junos rendszergazdáknál!

A Junos több lehetőséget is kínál ilyen szituációk kivédésére:

- amikor belépünk edit módba, a Junos szól, hogy nem commit-elt változások vannak:

```
root> configure
Entering configuration mode
The configuration has been changed but not committed
```

Ilyenkor az adminisztrátornak elvileg kutya kötelessége lenne megnézni, mi változott – tapasztalataink alapján ezt rendszeresen elfelejtik. A nem commit-elt változtatások egyszerűen megnézhetők:

```
root# show | compare
[edit interfaces ge-0/0/3 unit 0]
+   family inet {
+       address 192.168.200.200/24;
+   }

[edit]
root#
```

Ha olyan módosítás van, ami nem tetszik nekünk, simán eldobhatjuk:

```
root# rollback
load complete

root# show | compare

[edit]
root#
```

A Junos 10.x verzió 49 db előző konfigurációt jegyez meg; simán visszaállhatunk korábbi konfigurációra a fenti `rollback <config szám>` paranccsal amely meg is mutatja nekünk, ki / mikor konfigurált:

```
root# rollback ?
Possible completions:
  <[Enter]>          Execute this command
  0                  2010-12-05 09:11:55 UTC by root via cli
  1                  2010-12-05 08:44:07 UTC by root via cli
  2                  2010-12-05 08:39:32 UTC by root via cli
  3                  2010-12-05 08:26:15 UTC by root via cli
  4                  2010-12-05 07:46:44 UTC by root via cli
  5                  2010-12-05 07:39:14 UTC by root via cli
  6                  2010-12-05 07:24:35 UTC by root via cli
  7                  2010-12-05 07:21:14 UTC by root via cli
```

Ha pl. arra vagyunk kíváncsiak, hogy miben különbözik egy korábbi config a jelenlegitől, meg is nézhetjük:

```
root# show | compare rollback 2
[edit interfaces vlan]
+   unit 254 {
+       family inet {
+           address 192.168.254.5/24;
+       }
+   }
[edit vlans]
+   MGMT {
+       vlan-id 254;
+       l3-interface vlan.254;
+   }
[edit vlans PAC]
-   vlan-id 20;
+   vlan-id 29;
```

- exclusive config használata: ebben az esetben ha kilépünk a config-ból, a nem commit-elt változtatásaink elvesznek:

```
root> configure exclusive
warning: uncommitted changes will be discarded on exit
Entering configuration mode

[edit]
root# set system domain-name peetandcook.hu

[edit]
root# exit
The configuration has been changed but not committed
warning: Auto rollback on exiting 'configure exclusive'
Discard uncommitted changes? [yes,no] (yes) yes

warning: discarding uncommitted changes
Exiting configuration mode
```

Most kb. ennyit – sok sikert!

(C) pingTomi 2011.01.01