

Operációs rendszer szintű virtualizáció – Gyakorlat

Indítsuk el a CentOS-5.4-OpenVZ virtuális gépet! A gépre belépni a *meres* felhasználóval lehetséges, jelszava a szokásos. *Root* felhasználóként végzendő műveletekhez használhatjuk a *sudo*-t, ilyenkor a *meres* felhasználó rendes jelszavát kell megismételni.

A mérés során virtuális gépeket használunk, amiken belül a virtuális konténereket futtatjuk. Figyeljünk, hogy a mérési feladatok során, ha „hoszt” gépként hivatkozunk valamit, akkor az a méréshez kiadott virtuális gépet jelenti, nem pedig a fizikai labor PC-t.

OpenVZ konténerek kezelése webes felületen

1. Böngészőben nyissuk meg a <http://localhost:8001> oldalt (nyitó oldalnak ez van felvéve). Ha a későbbiekben majd csak vár az oldal a betöltésre, akkor lejárt a session, frissíteni kell az oldalt, és újra be kell lépni.
2. Belépéshez a felhasználónév *admin*, a jelszava a *meres* felhasználóéval azonos.
3. Először nézzünk rá az erőforrás konfigurációkra (a webes felületen *plan* néven szerepel), ezzel adhatóak meg a Virtual Environment (VE) konténerek erőforráskorlátai. *Plans* majd *Edit Plan Easy Wizard* ikont válasszuk, nézzük meg pl. a *light* plan korlátait!
4. Hozzunk létre egy VE-t (a webes felület ezt *node*-nak nevezi), *ubuntu-9.04* operációs rendszerrel, és az imént megnézett *light* erőforrás-konfigurációval! Fontos: a host nevet csak domain névvel együtt fogadja el, pl.: *test.local*. Az IP cím lehet bármi, ami a hoszt IP tartományában helyezkedik el és nem ütközik (ezt vizsgáljuk meg az *ifconfig eth0* illetve *PING* hívással!). Jegyezzük meg, hogy a VE a *101*-es azonosítót kapta.
5. Indítsuk el VE-t!

VE környezetének felderítése

1. Nyissunk egy terminálablakot, és nézzük meg milyen VE-k futnak:

```
vzlist -a
```

2. Nézzük meg mire alkalmas az OpenVZ svájci bicska:

```
vzctl --help
```

3. Pingeljük meg az imént indított VE-t. SSH-val belépni még nem fogunk tudni, mert nincs beállított jelszava a root felhasználónak a VE-ben...
4. ...nem is kell, menjünk be a VE-be „művészbejárón” keresztül:

```
vzctl enter 101
```

5. Nézzük meg, hogy ez tényleg nem a CentOS:

```
cat /etc/debian_release
```

Ilyen fájl a host gépen nincs (illetve van, csak nem ott :).

6. Nézzük meg, hogy honnan milyen erőforrás látszik! Nyissunk még egy terminálablakot vagy új tabot a terminálon, és a nézzük meg mindkét helyről a következőket.

6.1. Szabad memória:

```
free -m
```

(Igen, ha még nem tudatosult volna bennünk: most 30MB RAM-mal fut az Ubuntu – próbálja ezt meg valaki fizikai gépen ennyiből megoldani :)

6.2. Szabad hely a fájlrendszeren:

```
df -h
```

6.3. Futó folyamatok listája:

```
ps ax
```

(Ne hagyjuk magunkat félrevezetni, a PID nem lesz azonos a host és a VE folyamatlistájában, de a host felől látszanak a VE folyamatai, fordítva viszont nem.)

6.4. Jéé fut apache httpd, vajon hol hallgatózik:

```
netstat -lnp
```

Ennek is kicsit félrevezető lesz az eredménye a hostról nézve, úgy néz ki, mintha a localhoston hallgatóna az apache 80-as porton, de próbáljuk ki böngészőből! És a VE IP címét beírva?

6.5. Hozzunk létre egy fájlt a hoszt gépen (rootként):

```
touch /vz/private/101/root/EZ_EGY_TESZT_FILE
```

Nézzük meg a /root könyvtár tartalmát a VE-n belül! Fordítva is ki lehet próbálni.

6.6. CPU típusa és tulajdonságai:

```
cat /proc/cpuinfo
```

6.7. Futó kernel verziója, architektúrája:

```
uname -a
```

6.8. Kernelbe betöltött modulok:

```
lsmod
```

6.9. Hálózati interfészek és beállításai, beleértve az inaktívakat is:

```
ifconfig -a
```

Erőforrás-korlátok vizsgálata

Változtassuk meg futás közben a VE erőforráskorlátait, ellenőrizzük az állapotot előtte és utána! A változtatás a *Node Management* menüpont alatt végezhető el, akár az egyes erőforrások módosításával, akár a VE-hez hozzárendelt plan váltásával.

1. A *Server/Plans/Edit Plan Advanced Wizard*dal nézzük meg a *light plan* részletes beállításait! Az egyes pontok mellett jobb oldalt lévő *?*-re húzva az egeret bal oldalt leírást kapunk az egyes pontokról.
2. Vizsgáljuk meg a parancssori lehetőségeket is:

```
cat /proc/user_beancounters
```

LXC technológia vizsgálata

Az OpenVZ gépet hagyjuk még futni, szükség lehet rá.

1. Indítsuk el az Ubuntu 12.04 virtuális gépet! A gépre belépés *meres* felhasználóval lehetséges, jelszava a szokásos. *Root* felhasználóként végzendő műveletekhez használhatjuk a *sudo*-t, ilyenkor a *meres* felhasználó rendes jelszavát kell megismételni.
2. Indítsunk egy terminált, váljunk *root* felhasználóvá és vizsgáljuk meg az *lxc*-hez kapcsolódó parancsokat (*lxc TAB TAB*)
3. Nézzük meg milyen konténerek léteznek:

```
lxc-list
```

4. Hozzunk létre egy új konténert

```
lxc-create -n myfirstcontainer -t debian
lxc-start -n myfirstcontainer -d
lxc-console -n myfirstcontainer
```

5. Bizonyosodjunk meg róla, hogy valóban nem az Ubuntuban vagyunk:

```
cat /etc/apt/sources.list
```

6. Vizsgáljuk meg a hálózati beállításokat a hoszt, majd a virtuális gépen:

```
ifconfig -a
```

7. Vizsgáljuk meg a hálózati topológiát:

```
brctl show
```

Haladóbb feladatok

(Ha marad rá idő, némi Linux és hálózat adminisztrálási gyakorlatot igényel.)

LXC hálózati beállítások

Jelenleg az LXC konténer NAT mögött helyezkedik el egy virtuális belső hálózaton. Ezt a hálózatot az `lxcbr0` hálózati híd fogja össze. Készítsünk egy olyan topológiát, ami kiengedi a külső hálózatra a virtuális konténert. Az ubuntu virtuális gépen a hálózati híd használatához szükséges csomag (`bridge-utils`) már előre telepítve van.

1. Vegyük le a hoston az `eth0`-ról az IP címet:

```
ifconfig eth0 0.0.0.0
```

2. Hozzuk létre a hidat és adjuk hozzá az interfészeket:

```
brctl addbr br0
brctl addif br0 eth0
ifconfig br0 up
brctl showstp br0
#várjunk egy kicsit, amíg a port state learningből forwardingba vált, nézzük újra
dhclient br0
```

3. Állítsuk le a konténert és a konfigurációjában (`/var/lib/lxc/name/config`) módosítsuk a használt hálózati híd beállítását, majd indítsuk el a gépet!

```
lxc.network.link=br0
```

4. Vizsgáljuk meg a konténer IP címét, ha szükséges, akkor kérjünk IP-t a külső hálózatról:

```
dhclient eth0
```

(Megjegyzés: ez Windows VMware hoston remélhetőleg működni fog alapból, Linux hostnál külön engedélyezni kell, hogy a vendég gép Ethernet vezérlője MAC cím válogatás nélküli üzemmódba váltsón. Az ethernet bridge helyes működéséhez erre szükség van. A labor gépeken ez engedélyezve van már.)

5. Próbáljuk a konténer új IP címét pingelni, akár kívülről is!

OpenVZ hálózati beállítások

Az OpenVZ VE most egy *venet* típusú hálózati csatolóval rendelkezik. Ez több megkötéssel rendelkezik, cseréljük ki egy Ethernet típusú interfészre, amit a hoston hídba kapcsolhatunk a külső hálózati interfésszel.

Kis magyarázat hozzá: az OpenVZ kétféle hálózati interfészt tud adni a VE-nek, az egyik a *venet*, ami IP rétegbeli kapcsolat, a másik a *veth*, ami Ethernet rétegbeli. Bridge-elni érelemszerűen csak az *veth*-t lehet. Mindkettő pont-pont kapcsolat a host és a VE között, párban jönnek létre az interfészek a hoston és a VE-n. (http://wiki.openvz.org/Differences_between_venet_and_veth)

1. Vegyük el a jelenlegi interfészt:

```
vzctl set 101 --ipdel all
```

2. Adjunk hozzá egy *veth* interfészt:

```
vzctl set 101 --netif_add eth0
```

A `netif_add` paraméterénél a VE-n belüli nevét adjuk meg, a host felőli végén a neve *veth101.0* lesz. Vegyük észre, hogy a hoston is van *eth0* és a VE-n belül is van egy ettől teljesen független másik *eth0*, de ez nem okoz problémát.

3. A hoston telepítsük fel a `bridge-utils` csomagot:

```
yum install bridge-utils
```

4. Vegyük le a hoston az eth0-ról az IP címet:

```
ifconfig eth0 0.0.0.0
```

5. Hozzuk létre a hidat és adjuk hozzá az interfészeket:

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 veth101.0
ifconfig br0 up
brctl showstp br0
#várjunk egy kicsit, amíg a port state learningből forwardingba vált, nézzük újra
dhclient br0
```

6. És most a VE-n belül kérjük IP-t a külső hálózatról:

```
dhclient eth0
```

7. Próbáljuk a VE új IP címét pingelni, akár kívülről is!
8. Állítsunk valamit a VE-n belüli tűzfalon, pl. tegyük REJECT-re az INPUT láncot, és próbáljuk ki a pingelést. Vegyük észre, hogy a host tűzfala nem változott, és az továbbra is reagál a pingre. (Akár NAT-ot is lehet csinálni az egyes VE-ken belül, de ehhez előbb módosítani kell a */etc/vz/vz.conf*-ban az IPTABLES változó értékét.)
9. Csináljunk még VE-eket és bridge-eljük azokat is külső hálózatra vezető br0 hídra, vagy csináljunk belső hálózatot a VE-k között.