Virtualizációs technológiák és alkalmazásaik (VIMIAV89)

Operációs rendszer szintű virtualizáció - Gyakorlat

1. Indítsuk el a CentOS-5.4-OpenVZ-full 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. 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, egy refresh kell és újra be kell lépni.
    2. Belépéshez a felhasználónév *admin*, a jelszava a *meres* felhasználóéval azonos.
    3. Először nézzünk rá az erőforráskonfigurációkra (a webes felületen *plan* néven szerepel), ezzel adhatóak meg VE-k 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* erőforráskonfigurációval! Fontos: a host nevet csak domain névvel együtt fogadja el, pl: *test.local*. Az IP cím lehet bármi a *10.0.0.0/8* tartományból, kivéve a *10.40.0.0/16* alhálózatot, mert ez utóbbi a külső hálózaton lévő gépekkel ütközhet. Jegyezzük meg, hogy a VE a *101*-es azonosítót kapta.
    5. Indítsuk el VE-t!
3. VE környezetének felderítése
    1. Nyissunk egy terminálablakot és pingeljük meg az imént indított VE-t. SSH-val belépni még nem fogunk tudni, mert nincs beállított jelszava a root felhasználónak a VE-ben...
    2. ...nem is kell, menjünk be a VE-be „művészbejárón” keresztül: `vzctl enter 101`
    3. Nézzük meg, hogy ez tényleg nem a CentOS: `cat /etc/debian_release`. Ilyen fájl a host gépen nincs (illetve van, csak nem ott :)).
    4. Nézzük meg, hogy honnan milyen erőforrás látszik! Nyissunk még egy terminálablakot vagy új tabot a terminálon és a nézzük meg mindkét helyről a következőket.
        1. Szabad memória: `free -m`
        (Igen, ha még nem tudatosult volna bennünk: most 30MB rammal fut az ubuntu – próbálja ezt meg valaki fizikai gépen ennyiből megoldani :))
        2. Szabad hely a fájlrendszeren: `df -h`
        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.
        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 hallgatózna az apache 80-as porton, de próbáljuk ki böngészőből! És a VE ip címét beírva?
        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. CPU típusa és tulajdonságai: `cat /proc/cpuinfo`
7. Futó kernel verziója, architektúrája: `uname -a`
8. Kernelbe betöltött modulok: `lsmod`
9. Hálózati interfészek is beállításai, beleértve az inaktívakat is: `ifconfig -a`

4. 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.
5. Advanced feladatok (ha marad rá idő, némi Linux adminisztrálási gyakorlatot igényel)
   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. A VE most nem lát ki a hálózatra. Adjunk neki egy ethernet típusú interfészt, amit a hoston hídba kapcsolunk a külső hálózati interfésszel.
      1. Ezt a webes felületre nem vezették ki: `vzctl set 101 --netif_add eth0`
         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. 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.
      2. A hoston telepítsük fel a bridge-utils csomagot: `yum install bridge-utils`
      3. Vegyük le a hoston az eth0-ról az IP címet: ifconfig eth0 0.0.0.0
      4. 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 *learning*ből *forwarding*ba vált, nézzük újra
         `dhclient br0`
      5. És most a VE-n belül kérjünk IP-t a külső hálózatról: `dhclient eth0`
         (Megjegyzés: ez Windows vmware hoston remélhetőleg működni fog alapból, Linux hostnál külön engedélyezni kell, hogy a guest gép ethernet vezérlője MAC cím válogatás nélküli üzemmódba váltson. Az ethernet bridge helyes működéséhez erre szükség van.)
      6. Próbáljuk a VE új ip címét pingelni, akár kívülről is!
      7. 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.)
      8. Csináljunk még VE-ket és bridge-eljük azokat is külső hálózatra vezető br0 hídra, vagy csináljunk belső hálózatot a VE-k között!