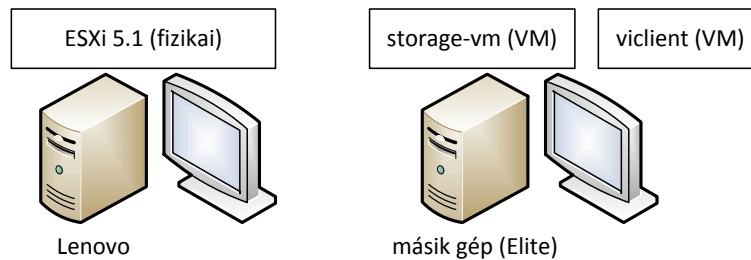


Kiszolgáló oldali virtualizáció – Gyakorlat

A gyakorlaton egy fizikai gépet és egy másik fizikai gépen futó két virtuális gépet fogunk használni.



ESXi indítása és beállítása

Első lépésként elindítunk egy ESXi-t a LENOVO márkájú laborgépen hálózatról, majd megadjuk az alapbeállításait.

1. ESXi bootolása PXEboot szerverről.
 - 1.1. A LENOVO márkájú gépeken a BIOS képernyőnél F12-t kell nyomni a boot menü megjelenítéséhez, majd a „Startup Device Menu” résznel a hálózati bootolást kiválasztani (PCI BEV: IBA BGE Slot opciót).
 - 1.2. A megjelenő kék boot PXE menüből az ESXi 5.1.0 Live opciót kell elindítani. (Akit érdekel: bővebben a [„Lord of the Servers”](#) blogon olvashat a netbootolás rejtjelmeiről.)
 - 1.3. Miután elindult az ESXi, F2 *Customize System* (egyelőre nincs még jelszó beállítva), majd a root jelszót állítsuk be „LaborImage”-re (*Configure Password* menüpont).
 - 1.4. Nézzük meg, hogy miket lehet beállítani, igazából főleg azt érdemes megfigyelni, hogy mi *nincs*. Hálózati adatokat tudunk megadni, de például virtuális gépeket nem tudunk kezelni.
 - 1.5. Böngészőben a másik gépről nézzük meg az ESXi webes felületét azon a címen, amit a fő képernyőjén mutat.
2. Belépés a *vSphere Client* programmal
 - 2.1. Indítsuk el a másik gépen a *virttech* könyvtárban lévő a *viclient* virtuális gépet, ez tartalmazza a *VMware vSphere Client* menedzsment alkalmazást, az ikonja a Taskbaron megtalálható. A Windows jelszava itt is a szokásos.
 - 2.2. A *vSphere Client*ben lépünk be az ESXi gépre az ESXi fő képernyőjén látható IP címet, *root* felhasználót és az imént beállított „LaborImage” jelszót megadva.
 - 2.2.1. Miért kapunk egy *Security Warning* üzenetet?
 - 2.3. Ha közvetlenül egy ESXi kiszolgálóhoz kapcsolódunk, akkor a *vSphere Client*ben csak szerepeket tudunk beállítani (*Roles*), naplófájlokat megnézni (*System Logs*) és a kiszolgálót és a rajta futó virtuális gépeket tudjuk menedzselni (*Inventory*).
 - 2.4. Nézzük meg a *Roles* nézetben, hogy milyen felhasználók vannak benne az *Administrator* szerepben! Próbáljunk egy új szerepet hozzáadni, és nézzük meg, hogy mennyi különböző jogot (*privilege*) lehet szabályozni. Keressük ki például, hogy a virtuális gép beállításainál mit lehet külön-külön szabályozni.
 - 2.5. Lépünk vissza kezdőképernyőre, és nézzük meg a *System Logs* nézetet. Hány darab logot tudunk megnézni? Keressük ki a *hostd.log* fájlból, hogy mi a build száma az elindított ESXi-nek (használjuk a szűrés funkciót)!

2.6. Nézzük meg az *Inventory* nézetben, hogy mit látunk az ESXi szerverünkben.

- 2.6.1. A *Summary* fül összefoglalja a legfontosabb adatokat a kiszolgálóról. Mennyi memóriát használ jelenleg a rendelkezésre álló fizikai memóriából?
- 2.6.2. A *Virtual Machines* fülön látnánk a kiszolgálóhoz rendelt virtuális gépeket, jelenleg még nincs egy sem.
- 2.6.3. A *Resource Allocation* fülön lehet majd az erőforrás-gazdálkodást (korlátok, részesedések stb.) áttekinteni.
- 2.6.4. A *Performance* fülön lehet a teljesítményszámlálókat grafikonon megjeleníteni.
- 2.6.5. A *Local Users & Groups* fül listázza ki a helyi felhasználókat. Hozzunk létre egy *testuser* nevű felhasználót (az UID tulajdonságát ne adjuk meg). Milyen UID-t osztott ki neki a rendszer?
- 2.6.6. Az *Events* fülön a kiszolgálóval kapcsolatos fontos események látszódnak. Keressük ki például a root felhasználó jelszavának megváltoztatását.
- 2.6.7. A *Permissions* résznél lehetne erre a szintre egy felhasználót vagy csoportot egy adott szerepbe berakni, és így különböző jogosultságokat osztani neki. Adjuk hozzá a korábban létrehozott *testuser* felhasználónkat *Read-only* szereppel. (Megjegyzés: a gyakorlatban sose közvetlenül felhasználókat rendelünk szerepekhez, hanem mindig csoportokat, így később sokkal könnyebb lesz menedzselni a rendszert.)

2.7. Nézzük végig a *Configuration* képernyőt. Látjuk, hogy még nincs beállítva datastore, erre nemsokára visszatérünk.

- 2.7.1. A *Health Status* fülön a gép hardver szenzorainak jelzéseit is meg tudjuk nézni. A laborgépeken csak a processzor jelzései szerepelnek itt, de egy szerver gépen rengeteg egyéb információt is ki tud jelezni.
- 2.7.2. A *Processors* és *Memory* fülön a CPU és memória alap adatait tudjuk megnézni.
- 2.7.3. A *Network Adapters* fülön a kiszolgálóban lévő hálózati kártyákat tudjuk megnézni. Mi a MAC címe az ESXi-t futtató laborgép egyetlen hálózati kártyájának?
- 2.7.4. A *Networking* fülön az virtuális hálózatok beállításait látjuk. Jelenleg egy virtuális switch van definiálva a kiszolgálón (*vSwitch0*), és ez rá van kapcsolva a *vmnic0* fizikai hálózati kártyára (ez kb. olyan, mint a VMware Workstation bridged üzemmódja). Ehhez van egy virtuális gépek által használt hálózat létrehozva (*VM Network* néven), valamint egy úgynevezett VMkernel port, ezen keresztül lehet például menedzselni távolról az ESXi-t. Nézzük meg a *vSwitch0* beállításait, hány darab portja van ennek a switch-nek? A beállításoknál sok egyebet meg lehetne adni (pl. hálózati forgalom korlátozása, hibatűrés), de ezek túlmutatnak ezen a gyakorlaton.
- 2.7.5. A *Software* csoportban is sok egyebet meg lehetne adni (hitelesítés, idő szinkronizáció, tűzfal szabályok), de ezekkel most nem foglalkozunk.
- 2.7.6. A *Storage* fület ha megnézzük, akkor látszik, hogy ez a lista üres, az ESXi egyelőre nem tudná hol tárolni a virtuális gépeket. Erre figyelmeztet egyébként a képernyő tetején lévő nagy sárga üzenet is. Foglalkozunk akkor most ezzel.

3. A virtuális gépek tárolására egy távoli gép által kijelölt kötetet fogunk használni, amit NFS (Network File System) protokollon keresztül érünk majd el. Ezt a jelen konfigurációban egy virtuális gépben futó Linux szerver fogja biztosítani.

3.1. Indítsuk el a *virttech* könyvtárban lévő *storage-vm* virtuális gépet.

3.2. Jegyezzük meg az IP-címet, amit a login prompt előtt kiír (*Bootup IP address* üzenet után).

- 3.3. Belépni elvileg nem szükséges, de ha valamiért nem írná ki az IP-címet, akkor lépünk be a root felhasználóval, és kérdezzük le a hálózati beállításokat az *ifconfig* parancs segítségével.
- 3.4. Az NFS szerver már előre be van állítva ezen a virtuális gépünk, úgyhogy több teendőnk nincs is vele.
- 3.5. Az NFS kiszolgáló által kijánlott kötetet lehetne a grafikus felületről is csatlakoztatni, de ezt most mi parancssorból fogjuk megtenni (csak, hogy ilyet is lássunk.)
- 3.6. A Taskbaron található a *VMware vSphere PowerCLI* alkalmazás ikonja, melyet elindítva a VMware API parancsaival ellátott PowerShell konzol indul. (A PowerShell a Windows új szkript és parancssori felülete, a nyelvvel részletesen az Informatikai Technológiák szakirányon az Intelligens Rendszerfelügyelet tárgy keretében foglalkozunk.)
- 3.7. Ebben hajtsuk végre a következő parancsokat (TAB-os automatikus kiegészítést lehetőség szerint hevesen használva a parancs és paraméternevekre, a # kezdetű sorok megjegyzések, azokat nem kell végrehajtani):

```
cd \  
# the next command will open a remote connection to the ESXi  
Connect-VIServer <ESXi IP>  
# this command will connect the NFS volume  
New-Datastore -VMHost <ESXi IP> -Name storage-store -Nfs -Path /srv/nfsroot/storage -  
NfsHost <storage IP>
```

(Figyelem, ezt az utolsó parancsot egy sorba kell írni, csak nem fért ki!)

Futás közben figyeljük a vSphere Client ablak alján a *Recent Tasks* logot. A Connect-VIServer felhasználót és jelszót kér majd az ESXi-hez, itt adjuk meg a root felhasználót és jelszavát.

- 3.8. Ezzel csatlakoztatjuk az NFS-en keresztül kijánlott tárhelyet az ESXi-hez.
- 3.9. Térjünk vissza a vSphere Client alkalmazáshoz, és nézzük meg a Storage fület. Itt már látszik az új datastore, amit felcsatoltunk az előbb. Mekkora a mérete? A storage-store sor jobb gombos menüjében kattintsunk a *Browse datastore* opcióra, ezzel megnézhetjük a tartalmát is.

Virtuális gépek létrehozása

4. Hozzuk létre egy üres virtuális gépet, Ubuntu Linux (64 bit) vendég operációs rendszernek (a menüből: *Inventory / Host / New Virtual Machine...*). Legyen *Typical*, a neve legyen *ubuntu-vm*, a virtuális diszk legyen 4 GB. Adjunk **2 processzort** neki. Milyen virtuális hardver lehetőségek vannak? Miben különbözik ez egy Workstation-tól?
 - 4.1. Indítsuk el a virtuális gépet.
 - 4.2. Kapcsoljunk gyorsan a *Console* fülre, mert elég gyorsan bootol.
 - 4.3. Mivel ez ki van bridge-elve a hálózatra azért itt is elérhető a hálózati boot menü. Válasszuk a *LabpcG2-Ubuntu-10.04-64bit-NFS-RO* boot opciót. (Ezzel viszonylag fájdalommentesen megúsztunk egy OS telepítést 😊.) Várjuk meg, hogy elinduljon az operációs rendszer.
 - 4.4. Nézzünk rá a *Performance* fülre, tekintsük meg a CPU terhelést és a hálózati forgalmat a bootolás közben. Ha a *Performance Chart Legend* résznél kijelölünk egy számlálót, akkor azt kiemeli a grafikonon is. Nézzük meg például, hogy mennyi volt a maximuma a *Network / Data receive rate* számlálónak. A számlálók értékei 20 másodpercentként frissülnek.
 - 4.5. Az elindított Ubuntu gépre beléphetünk ugyanazzal a *meres* felhasználóval és jelszóval, mint a fizikai laborgépekre. A Console nézetet „kihozhatjuk” egy külön ablakba a menüsor alatt lévő gombok közül a *Launch Virtual Machine Console* segítségével.

5. Terheljük a virtuális gépet

5.1. Terheljük le CPU-t a 7zip benchmark üzemmódjával. A 7zip képes 2 CPU-t kihasználni. A következő parancsot elindítva meg is mérjük, hogy meddig tartott a végrehajtás:

```
time 7z b
```

(Figyelem: a virtuális gépen belüli időmérés pontatlan a virtualizáció miatt, ezért az itt mért értékeket óvatosan kell kezelni.)

5.2. Készítsünk még egy másik példányt ebből a virtuális gépből, hasonlóképpen hálózatról bootolva az Ubuntu image-et, terheljük le mindkét virtuális gépet egyszerre. Mit tapasztalunk?

6. Erőforrás korlátozás

6.1. Hozzunk létre egy *Resource Pool*t. Tegyük át mindkét gépünket ebbe a poolba. Nézzük meg a Pool tulajdonságait, miket lehet itt beállítani?

6.2. Állítsunk be CPU erőforráskorlátot a Resource Poolra, pl. 2000MHz-re, ismételjük meg a mérést így is. Mit tapasztalunk? Nézzük meg a grafikont a *Performance* fülön is.

6.3. A virtuális gépek tulajdonságainál is találhatunk erőforráskorlátot a *Resources* fül alatt. Itt most a CPU *Shares* értékét állítsuk *LOW*-ra az egyik virtuális gép esetén. Mit tapasztalunk, mi látszik most a teljesítmény grafikonokon?

7. Speciális teljesítménynézetek

7.1. Nézzük meg a gazdagép Performance fülét, és a *Chart options...* menüvel állítsuk át a CPU-t „*Stacked per VM*” nézetre, valamint válasszuk ki az összes virtuális gépet. Most egyben láthatjuk a resource pool hatását.

7.2. Végezetül nézzünk rá a gazdagép *Memory* grafikonjára. A *Line graph* esetén kapcsoljuk be a *Shared* és *Shared Common* attribútumokat is, hogy láthassuk a virtuális gépek között deduplikált memóriát. Ha az idő engedi, akkor még további grafikon beállításokat is meg lehet tekinteni.

ESXi helyi konzol

8. Menjünk vissza a bal oldali géphez, és lépünk be az ESXi felületére.

8.1. A *Troubleshooting* résznél kapcsoljuk be az *ESXi Shell*-t, majd Alt+F1 megnyomásával jelenítsük azt meg.

8.2. Lépünk be root-ként, és nézzünk körül, hogy mit látunk (*cd, ls* parancsok).

8.3. Milyen log és konfigurációs fájlok vannak az ESXi-nek?

9. Állítsuk le sorban az ESXi-n futó virtuális gépeket, az ESXi-t (akár távolról), végül a másik Labpc-n futó virtuális gépeket.