

## Kiszolgáló oldali virtualizáció – Gyakorlat

### 1. ESXi indítás hálózati bootolással, alap konfigurációs beállítások

#### 1.1. ESXi bootolása PXEboot szerverről.

- 1.1.1. A munkaállomások bal oldali gépein a BIOS képernyőnél F10 vagy F12-t kell nyomni a boot menü megjelenítéséhez, majd innen a hálózati bootolást kiválasztani (az *IBA BEVPCI GE Network Adapter*).
- 1.1.2. A megjelenő kék boot menüből az ESXi 5.0-t kell elindítani. (Akit érdekel: bővebben a [„Lord of the Servers”](#) blogon olvashat a netbootolás rejtjelmeiről.)
- 1.1.3. Miután elindult az ESXi, F2 *Customize System* (egyelőre nincs még jelszó beállítva), majd a root jelszót állítsuk be „LaborImage”-re. Nézzük meg, hogy miket lehet beállítani, igazából főleg azt érdemes megfigyelni, hogy mi nincs 😊.
- 1.1.4. Böngészőben a másik gépről nézzük meg az ESXi webes felületét azon a címen, amit a fő képernyőjén mutat.

#### 1.2. Belépés a vSphere Client programmal

- 1.2.1. Indítsuk el a másik gépen a *viclient5* virtuális gépet, ez tartalmazza a *VMware vSphere Client* menedzsment alkalmazást, az ikonja a Taskbaron megtalálható. A Windows jelszava itt is a szokásos.
- 1.2.2. A vSphere Clientben lépünk be az ESXi gépre az ESXi fő képernyőjén látható IP címet, *root* felhasználót és az imént beállított „LaborImage” jelszót megadva.
- 1.2.3. Nézzük meg az *Inventory* nézetben, hogy mit látunk az ESXi szerverünkben.
- 1.2.4. Nézzük végig a *Configuration* képernyőt. Látjuk, hogy még nincs beállítva *datastore*.

#### 1.3. A virtuális gépek tárolására egy távoli gép által kijánlott iSCSI kötetet fogunk használni. Ezt a jelen konfigurációban egy virtuális gépben futó iSCSI target fogja biztosítani.

- 1.3.1. Indítsuk el az *ArchLinux-iSCSITarget* virtuális gépet a jobb oldali gépen. A bootolás után „Arch Linux iSCSI Target Appliance” felirat fog fogadni minket. Az iscsi target most alapértelmezetten konfigurálva van, és el is van indítva, csak a login prompt előtti IP címet kell megadni az ESXi szervernek.
- 1.3.2. A *viclient5* virtuális gépben vessünk egy pillantást az asztalon lévő *Prepare-ESXi-v3.ps1* fájlra. Láthatjuk, hogy ez egy PowerShell script, ami két kötelező és egy opcionális paramétert vár: az ESXi szerver IP címét, az iSCSI szerver IP címét és egy logikai paramétert, ami meghatározza, hogy létrehoz-e fájlrendszert a felcsatolt eszközön. (A PowerShell nyelvvel részletesen az Informatikai Technológiák szakirányon az *Intelligens Rendszerfelügyelet* tárgy keretében foglalkozunk /x/.)
- 1.3.3. Indítsuk el az asztalon lévő *VMware vSphere PowerCLI* alkalmazást. Ez egy PowerShell session, ami tartalmazza a VMware távoli menedzselést biztosító moduljait.
- 1.3.4. A PowerCLI parancssorban indítsuk el a következő parancsot (egy sorban, TAB-os autocomplete-et lehetőség szerint hevesen használva a könyvtár és fájlnevekre):

```
cd "C:\Users\meres\Desktop\  
.\Prepare-ESXi-v3.ps1 -Server <ESXi IP> -IscsiServer <iscsi IP> -CreateDatastore
```
- 1.3.5. Futás közben figyeljük a vSphere Client ablak alján a *Recent Tasks* logot.
- 1.3.6. A script futása közben feldobott login ablakban adjuk meg az ESXi szerver *root* felhasználóját és az imént beállított jelszavát. A script elvégz minden műveletet, ami az iSCSI-hoz csatlakozáshoz és *datastore* létrehozásához kell.

## 2. Virtuális gépek létrehozása

2.1. Hozzuk létre egy üres virtuális gépet, Ubuntu Linux 64 bit vendég operációs rendszernek. A virtuális diszk legyen a lehető legkisebb. Adjunk **2 processzort** neki. Milyen virtuális hardver lehetőségek vannak? Miben különbözik ez egy Workstation-tól?

2.1.1. Indítsuk el a virtuális gépet.

2.1.2. Kapcsoljunk gyorsan a *Console* földre, mert elég gyorsan bootol.

2.1.3. Mivel ez ki van bridge-elve a hálózatra azért itt is elérhető a hálózati boot menü. Válasszuk a *Labpc-NFS-RO-Autodetect* boot opciót. (Ezzel viszonylag fájdalommentesen megúsztunk egy OS telepítést ☺.)

2.1.4. Nézzünk rá a *Performance* földre, tekintsük meg a CPU terhelést és a hálózati forgalmat a bootolás közben.

2.1.5. Az elinduló Ubuntu gépre beléphetünk ugyanazzal a *meres* felhasználóval és jelszóval, mint a fizikai laborgépekre.

### 2.2. Terheljük a virtuális gépet

2.2.1. Terheljük le CPU-t a 7zip benchmark üzemmódjával: terminal ablakból indítsunk egy „7z b” parancsot. A 7zip képes 2 CPU-t kihasználni.

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

### 2.3. Erőforrás korlátozás

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

2.3.2. Állítsunk be CPU erőforráskorlátot a Resource Poolra, pl. 1000MHz-re, ismételjük meg a mérést így is. Mit tapasztalunk? Nézzük meg a grafikont a *Performance* fülön is.

2.3.3. A virtuális gépek tulajdonságainál is találhatunk erőforráskorlátot a *Resources* fül alatt. Itt most a CPU *Shares* értékét állítsuk *LOW*-ra. Végezzük el így is a mérést, külön-külön is és egyszerre indítva is a két virtuális gépben.

### 2.4. Speciális teljesítménynézetek

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

2.4.2. Végezetül nézzünk rá a gazdagép *Memory* grafikonjára. A *Line graph* esetén kapcsoljuk be a *Memory Shared* és *Memory 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.

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

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

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

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

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