

2F Házi feladat

Konfigurációkezelés

FIGYELEM: A házi feladat megoldása előtt olvassa el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézze végig a HF tudnivalóknál szereplő ellenőrző listát!

Feladat

Cégünk az egyre növekvő privát cloud üzletágból szeretne magának egy szép nagy szeletet kihasítani, így egy saját alkalmazást tervez, ami a cloudot alkotó számítógépek hálózati beállítását tudja összesíteni és módosítani. Az információkat a Linuxban rendelkezésre álló CIM osztályokból szeretnénk kinyerni a WS-Management protokoll segítségével, de mivel a cégben még senki nem foglalkozott ezzel, ezért először egy prototípuson kell kipróbálnunk.

Készítsünk tehát egy olyan Bash szkriptet, ami paraméterként megkapja egy CSV fájl elérési útját, majd a CSV fájlban felsorolt gépeken futó CIMOM-okat sorban lekérdezi **WS-Management** segítségével, és mindegyikről kigyűjti az alapvető hálózati beállításokat. Fel kell sorolni, hogy milyen hálózati interfészek vannak beállítva rajtuk, majd, ha az interfészhez tartoznak IP beállítások, akkor meg kell jeleníteni azokat. Az eredményt egy XML fájlban kell visszaadni. Feltételezhetjük, hogy a távoli gépek mind Linuxot futtatnak.

A szkript elnevezése és paraméterezése

```
getNetworkInfo.sh -i <machineData> -o <outputFile>
```

A szkriptnek kötelező ezt az elnevezést és paraméterezést használnia.

A szkript a paramétereket a következő formában fogadja:

- <machineData>: a távoli számítógépek adatait tartalmazó CSV fájl elérési útja, kötelező.
- <outputFile>: a kimeneti fájl elérési útja, kötelező.

Bemeneti fájl

A bemeneti fájl egy egyszerű CSV fájl:

```
machineName,port,protocol,user,password  
192.168.250.128,5585,http,administrator,password  
testmachine,5586,https,meres,password2
```

(A jelszó nyílt szövegben tárolása éles környezetben nem javasolt megoldás, ez most csak a házi feladat egyszerűsége miatt engedhető meg. Éles környezetben a jelszót érdemes ilyenkor például titkosítva tárolni vagy nyilvános kulcsú titkosításra alapuló módszereket alkalmazni.)

Kimeneti fájl

A kimeneti fájl egy XML fájl a következő formátumban:

```
<?xml version="1.0"?>
<NetworkInfo>
  <Host name="192.168.250.128">
    <Port name="lo" type="LocalLoopbackPort">
      <IPEndpoint IPv4Address="127.0.0.1" IPv6Address= "NULL" />
    </Port>
    <Port name="eth0" type="EthernetPort">
      <IPEndpoint IPv4Address="192.168.250.128" IPv6Address= "NULL" />
    </Port>
    <Port name="eth1" type="EthernetPort">
    </Port>
  </Host>
  <Host name="testmachine">
  </Host>
</NetworkInfo>
```

További elvárások

- A szkript végezzen alapvető ellenőrzéseket a bemeneti paramétereken (pl. bemeneti fájl létezik-e, a kimeneti fájl nem létezik stb.).
- A szkriptnek kezelnie kell az „egzotikusabb” hálózati csatolókat is (pl. TokenRing), és nem elég csak közvetlenül a loopback és Ethernet interfészeket lekérdezni.
- A szkript kezelje azt az esetet, ha a távoli géphez nem sikerül csatlakozni. Ilyenkor is kerüljön bele a gép a kimeneti állományba, és a szkript írjon hibaüzenetet a stderr kimenetre.

Megjegyzések

- A megfelelő provider aktuális verziója az IPv4 címet se adja vissza mindig, ilyenkor a kimenetbe nyugodtan kerülhet NULL.
- A megfelelő provider nem adja vissza az egy interfészhez rendelt további IP címeket, így mindenhol elég csak az első IP címet megadni.

Plusz feladat

- Alap esetben nem titkosított kommunikáció zajlik, módosítsa a szkriptet és a tesztkörnyezetet úgy, hogy a kommunikáció SSL csatorna felett működjön.