https://doi.org/10.62560/dk.2025.2.05

HADARICS KÁLMÁN *

Linux-szerver automatikus telepítése on-premise környezetben

Összefoglalás: „Az automatizálás a banális és ismétlődő feladatok visszaszorulását eredményezi." Rendszermérnökként gyakran találkozunk olyan feladattal, amelyet időről időre el kell végezni és a munkafolyamat előre meghatározott lépésekből áll. Azért, hogy az emberi hibát minimalizáljuk Linux operációs rendszer telepítést és konfigurálását is automatizálhatjuk. Jelen cikk azt mutatja be, hogy egy világszinten jelentős szolgáltató esetében milyen módon valósítható meg egy bérelt on-premise szerver telepítése és előkészítése további konfiguráció-menedzsment feladatok végrehajtására.
Kulcsszavak: Linux, szkript, automatikus telepítés.

Abstract: "Automation is driving the decline of banal and repetitive tasks." As a systems engineer, we often encounter a task that needs to be performed from time to time, and the workflow consists of predefined steps. To minimize human error, we can automate the installation and configuration of Linux operating systems. This paper describes how to deploy and prepare a leased on-premise server for further configuration management tasks in case of a major global service provider.
Keywords: Linux, script, automatic installation.

* *Dunaújvárosi Egyetem, Informatikai Intézet*
E-mail: hadarics@uniduna.hu

Bevezetés

Manapság sok olyan infrastruktúra-szolgáltató létezik, amely fizikai szerverek bérlését teszi lehetővé ügyfelek számára. Ezen szolgáltatás igénybevétele esetén a szolgáltató a hálózati infrastruktúrát és a hardver meghibásodások kezelését/cseréjét biztosítja az ügyfeleknek alapértelmezésként. A szerver számítógépre telepített szoftverek konfigurálása, üzemeltetése az ügyfél feladata.

[1] Hetzner Online GmbH – https://www.hetzner.com

[2] Installimage – https://github.com/hetzneronline/installimage

[3] Hetzner robot – https://robot.hetzner.com

[4] Ansible – https://www.ansible.com

Ezen publikáció esetében ismertetek egy saját megoldást, amely adott szerverszolgáltató által biztosított lehetőségek figyelembevételével, automatizált Debian GNU/Linux-telepítés és -konfiguráció menedzsmentjének megoldására mutat be egy gyakorlati megoldást.

A szolgáltató sajátosságai

A Hetzner Online [1] Európa egyik legnagyobb adatközpont üzemeltetője. 1997 óta foglalkozik IT infrastruktúra-szolgáltatásokkal, szerverek elhelyezésével és bérbe adásával. Németországi adatközpontjai mellett Finnországban, USA-ban, Szingapúrban is rendelkezik telephelyekkel. Több százezer működő szervert szolgál ki az infrastruktúrája. Az ügyfeleket elsősorban az innovatív technológiákkal és a vonzó árakkal igyekszik magához csábítani, de kiemelten figyel a fenntarthatóságra is. Németországban 100%-ban a szükséges villamos energiát megújuló energiaforrás (vízenergia) segítségével állítja elő.

A Hetzner Online az ügyfelek részere egy speciális Linux netboot környezetet biztosít bérelt szerverek esetében. Az ún. Rescue-környezet segítségével az ügyfeleket többek között:

– operációs rendszereket telepíthetnek, az ún. Installimage [2] segítségével,
– hardver ellenőrzéseket futtathatnak,
– egy esetlegesen nem működő (nem bootoló) rendszer esetében adatmentést, helyreállítást végezhetnek Linux parancsok segítségével.

A szolgáltató emellett lehetővé teszi az ún. Robot Webservice API [3] használatával ügyféloldali műveletek automatikus elvégzését.

A kliensoldali környezet, és működési feltételek

A címbéli automatizált szervertelepítési megoldással kapcsolatban az alábbiak a célkitűzések:

– Input-adatként használja fel a szolgáltatótól kapott paramétereket.
– Végezze el előre meghatározott módon a szerver meghajtóinak particionálását és a szerver nevének beállítását.
– Kezelje a szerver újraindítását, és újraindítást követően folytatódjon a végrehajtása.
– Telepítse a választott konfigurációmenedzsment-megoldást (Ansible [4]), és tegye lehetővé az újonnan telepített szerver távoli menedzsmentjét.

– Az automatizált telepítést Linux asztali környezetből vagy egy korábban létrehozott szerverről lehessen indítani, adott szkript végrehajtásával.

Az előzőek figyelembevételével, a működési feltételek az alábbiak:
– *openssh-client* – OpenSSH-kliens parancsok,
– *sshpass* – SSH-jelszóalapú hitelesítés használata.

A szolgáltató a sikeres bérlést követően az alábbi adatokat biztosítja az ügyfél részére (minta):

„Your server #server_id (192.168.108.1) has been activated and is now ready for use. Currently it is booted in the Rescue System. You can access the server via SSH2 using the following details:
IPv4 Address: 192.168.108.1

IPv6 Address: fe80::250:56ff:fec0:8
Username: root

Password: xDm8MieTXsU9ue

Host key: MYsjNcUoTgGEVHSUYfeJhRW9BOmh1m
HDFTLQaZ7bh/c (ECDSA 256)
k0pj5G4hiE7xBgf6325xfoicLUERmN8Kl5kLmVCNpno (ED25519 256)
Qvty3unE+4mAgTS2RFA7XcicBm/OPx88cXv/Fb3tHfQ (RSA 3072)

Az előző táblázatban alkalmazott IPv4 és IPv6 címek módosításra kerültek, és helyükre értelemszerűen a szolgáltatótól kapott publikus címeket kell elképzelnünk.

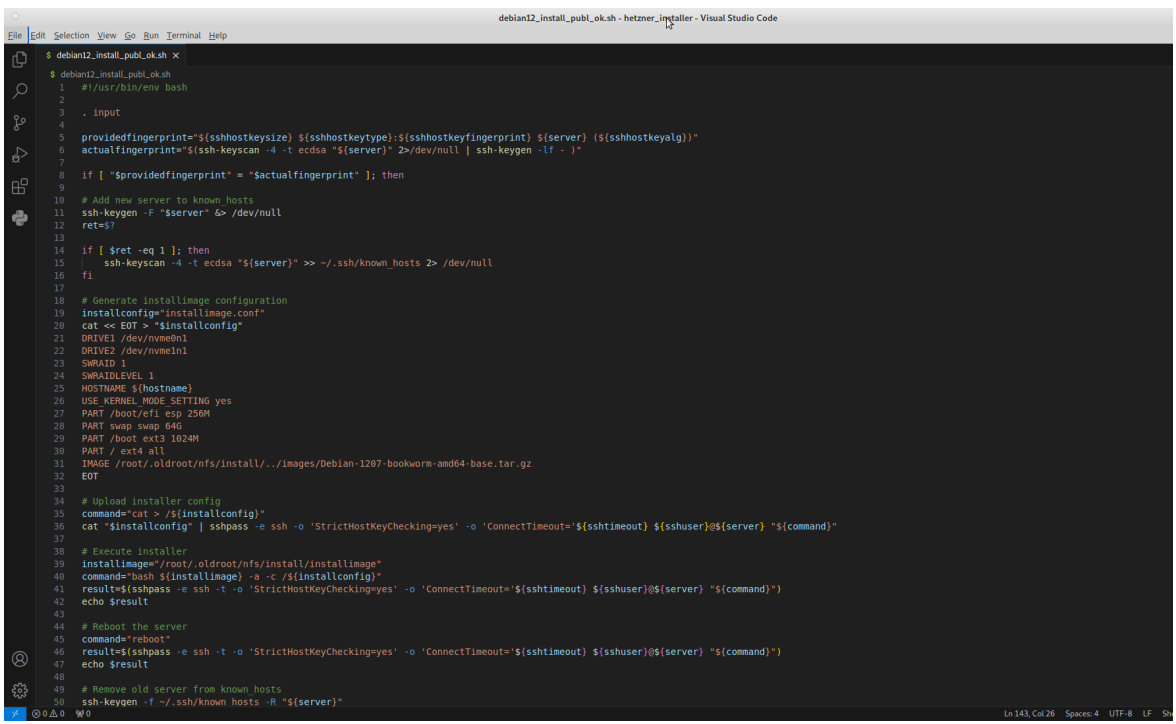
Az elkészített megoldás

A megoldás esetében egy input nevű fájlban rögzítjük a telepítendő szerverszolgáltatótól kapott releváns paramétereket és az új szerver nevét:

```
server=”192.168.108.1”
hostname=”debian12server”
sshuser=”root”
export SSHPASS=”xDm8MieTXsU9ue”
```

export TERM=”xterm-256color”
sshtimeout=20
sshhostkeyalg=ECDSA
sshhostkeysize=256
sshhostkeytype=”SHA${sshhostkeysize}”
sshhostkeyfingerprint=”MYsjNcUoTgGEVHSUYfeJhRW9BOmh1mHDFTLQaZ7bh/c”

1. ábra. Forráskód első része

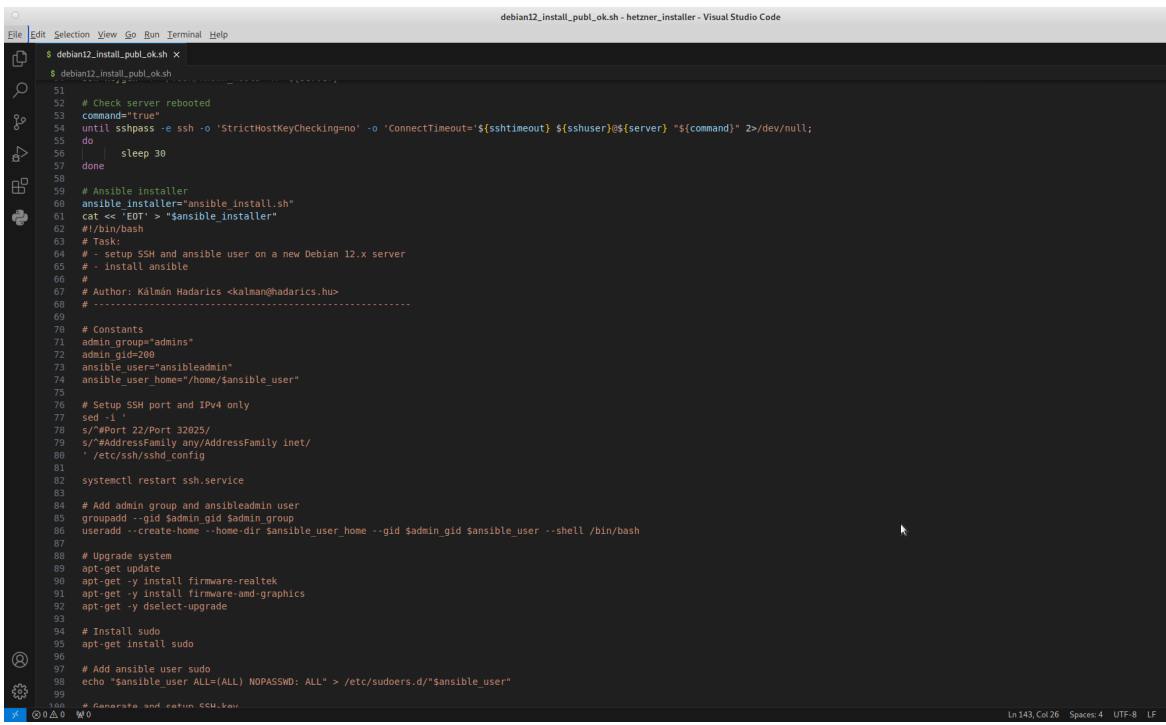


Forrás: Saját ábra.

A szkript az alábbiakat végzi el:
– Beolvassa az input fájlbéli paramétereket.

– Ellenőrzi, hogy a szolgáltató által küldött SSH ECDSA 256 kulcs lenyomata egyezik-e azzal, amit az aktuális kapcsolódáskor kapott a szkript. Amennyiben nem, akkor a szkript az ujjlenyomatok értékeivel fog visszatérni.
– Amennyiben egyezik az ujjlenyomat, akkor az alábbiak kerülnek végrehajtásra:

2. ábra. Forráskód második része

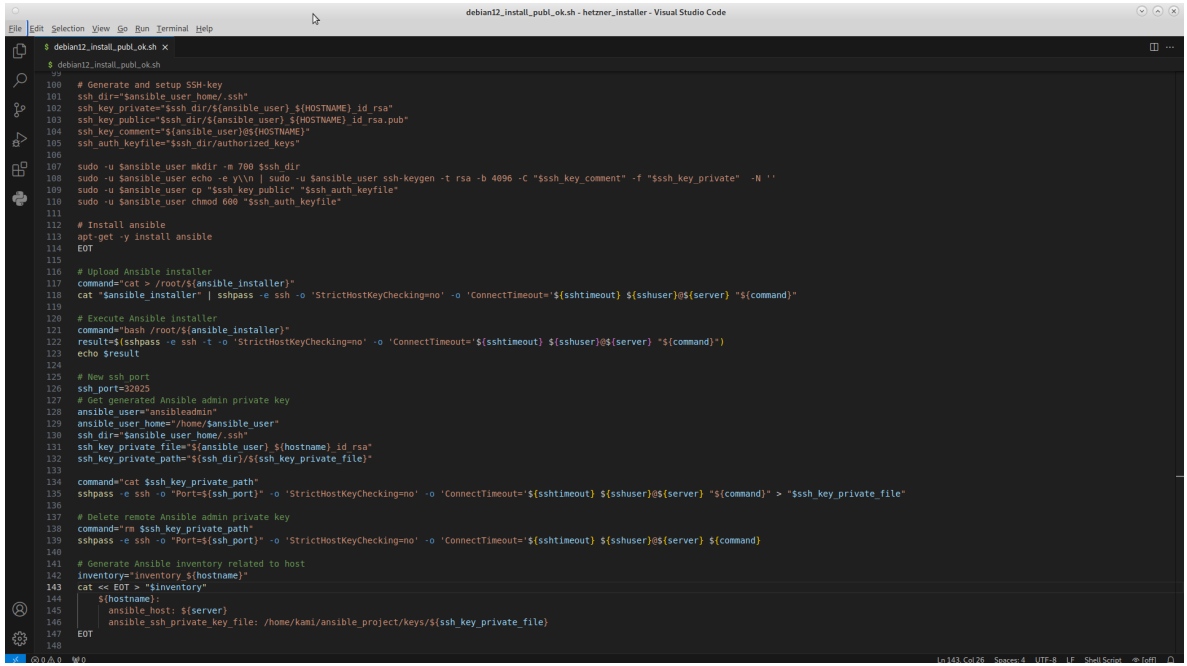


Forrás: Saját ábra.

– Az új szerver hozzáadása a kliens *.ssh/known_hosts* fájljához.
– Az installimage [2] konfigurációjának *(installimage.conf)* dinamikus létrehozása.
– Az *installimage.conf* fájl feltöltése SSH-kliens segítségével.
– Az *installimage* parancs távoli végrehajtása SSH segítségével a szerveren. Ez elvégzi a konfiguráció alapján a telepítést és a hálózat alapbeállításait.

– A telepített szerver újraindítása.
– A korábbi tárolt ujjlenyomatok törlése a *.ssh/known_hosts* fájlból. A szerver újratelepítése új SSH-kulcsokat hoz létre, így értelemszerűen az ujjlenyomatok is változnak.
– Várakozás az szerver újraindulására, 30 másodpercenként csatlakozási kísérlet.
– Az ansible_installer.sh szkript dinamikus létrehozása. Ez a szkript az alábbiakat végzi:
  – admins csoport és rendszer felhasználó (ansible_admin) létrehozása ($ansible_user változó tartalma)
  – az alapértelmezett OpenSSH-port módosítása (biztonsági megfontolásból);
  – a feltelepített rendszer frissítése,
  – sudo telepítése és ansible_admin sudo privilégiumának konfigurációja,
  – SSH-kulcspár létrehozása az ansible_admin felhasználó esetében.
  – az ansible csomag telepítése, amely lehetővé teszi a távoli adminisztrációt SSH felhasználásával.
– Az ansible_installer.sh szkript távoli végrehajtása SSH segítségével a szerveren.
– Az ansible_admin számára generált kulcspár privát kulcsának letöltése a helyi rendszerbe, majd a távoli szerverről a privát kulcs törlése.
– Ansible inventory fájl létrehozása, amely felhasználásával a távoli szerver Ansible segítségével távolról vezérelhető lesz.

3. ábra. Forráskód harmadik része



Forrás: Saját ábra.

Összegzés

Ahol gyakori üzemeltetési feladat Linux-szerverek telepítése, mindenképpen gondolkodni kell azon, hogy milyen módon valósítható ez meg automatizált módon. On-premise környezetben az automatizált telepítés megoldása nagyban függ a szolgáltató által biztosított feltételektől és lehetőségektől. Némi szkriptírási és integrálási tapasztalatokkal elkészíthető olyan megoldás, amely rugalmasan felhasználható automatikus telepítés kivitelezésére.

A bemutatott megoldás képes arra, hogy az újonnan telepített rendszer Ansible által vezérelhető legyen, a használt Ansible-inventory kibővítésével.