
CHALLENGE 1: LINUX SCRIPTING & SERVICES AUTOMATION

Alumna: Gemma Isela Castañeda Hernández

Código: 304190483

22 DE SEPTIEMBRE DE 2024

PROF. JOSE ANTONIO AVIÑA MENDEZ

Seminario de solución de problemas de uso, adaptación, explotación de Sistemas Operativos.

NRC 103857, Clave I504, Sección D03

Introducción

En este Challenge 1, titulado “Linux Scripting & Services Automation,” en la cual se crea un script automatizado en Linux utilizando la herramienta VBoxManage de VirtualBox. El objetivo principal de esta actividad es simplificar el proceso de creación y configuración de una Máquina Virtual (MV) a través de un script que recibe parámetros específicos como el nombre de la máquina, tipo de sistema operativo, número de CPU, tamaños de memoria RAM y VRAM, y el tamaño del disco duro virtual.

La automatización de estos procesos resulta de gran importancia en entornos de desarrollo y producción, ya que permite reducir considerablemente el tiempo invertido en tareas repetitivas, como la configuración manual de máquinas virtuales. Además, minimiza los errores humanos que podrían ocurrir al realizar configuraciones manualmente, asegurando que todos los parámetros sean consistentes y reproducibles.

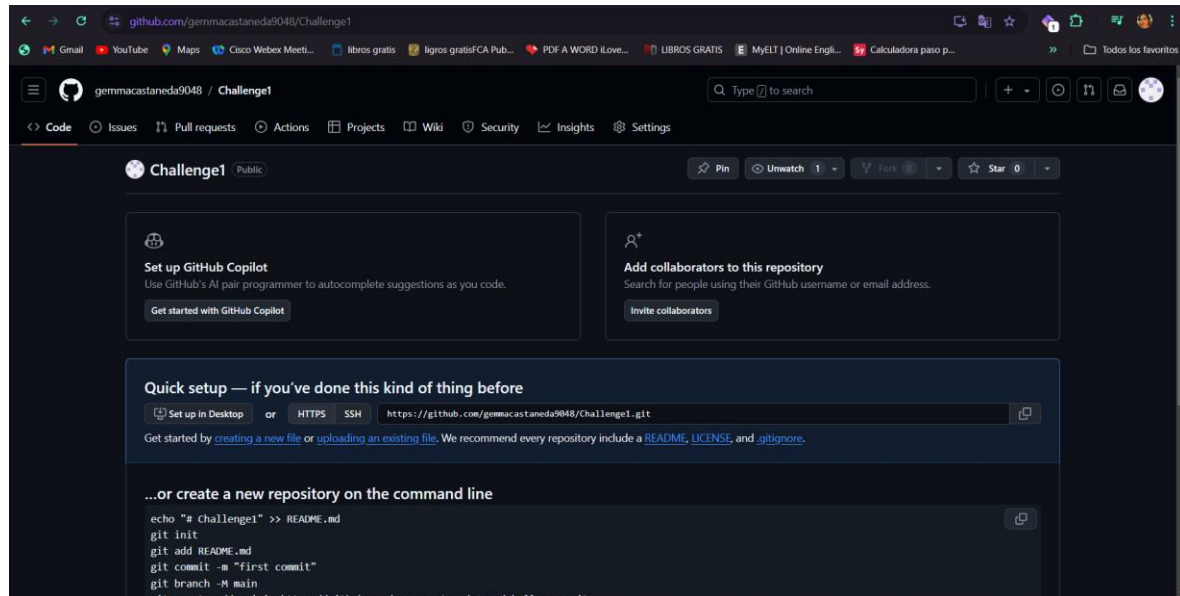
El script diseñado no solo optimiza el proceso de creación de la MV, sino que también automatiza la configuración de componentes clave como los controladores SATA e IDE, asociando estos controladores al disco duro. Además, el script imprime una lista detallada de todos los elementos configurados, lo que facilita la visualización y verificación de la correcta creación de la máquina virtual.

Al crear y reutilizar estos scripts automatizados, se fomenta la estandarización de procesos, lo cual es importante para mejorar la eficiencia y la consistencia en entornos de tecnología cada vez más complejos.

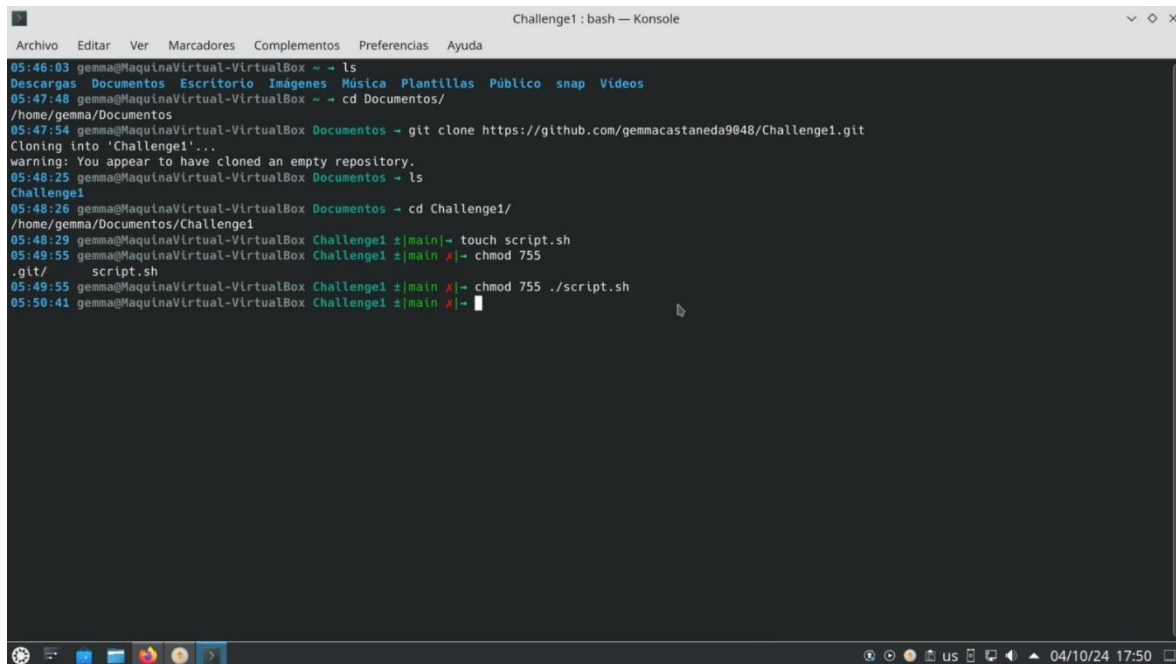
Contenido

Introducción	1
Challenge 1: Linux Scripting & Services Automation.....	3
Conclusiones.....	11
Enlace.....	12

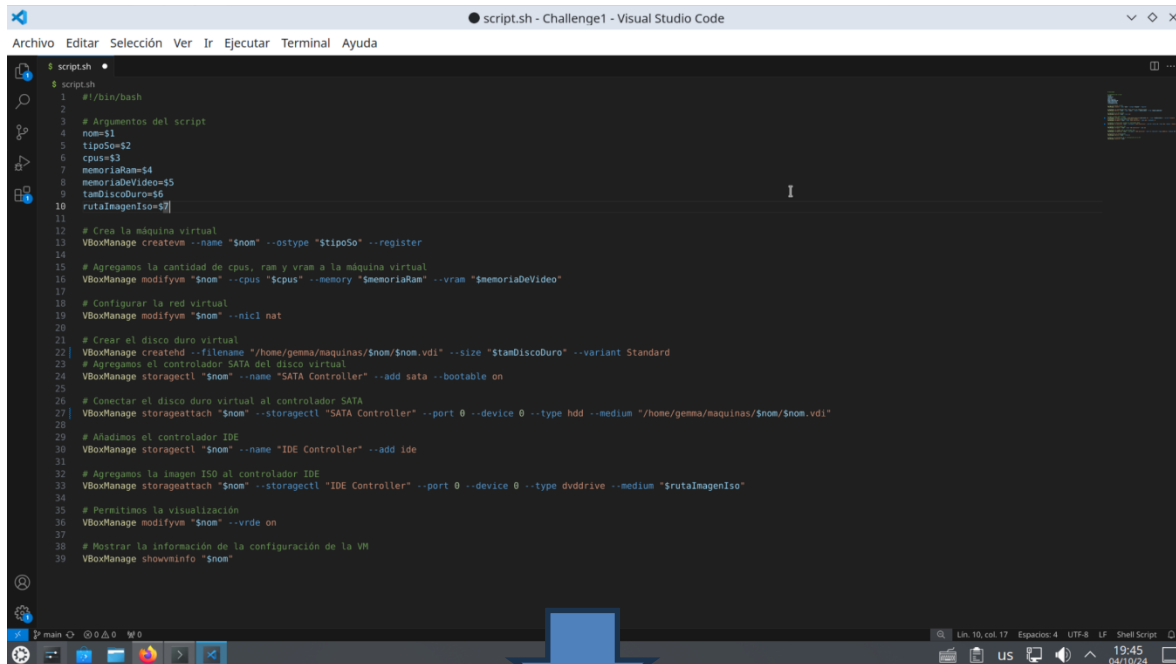
Challenge 1: Linux Scripting & Services Automation



Se crea el repositorio “Challenge1”.



Se visualiza que se clona el repositorio, se accede a la carpeta del repositorio, se crea un archivo llamado “script.sh”, lo hacemos ejecutable con el comando chmod.



```

script.sh
1 #!/bin/bash
2
3 # Argumentos del script
4 nom=$1
5 tipoSo=$2
6 cpus=$3
7 memoriaRam=$4
8 memoriaDeVideo=$5
9 tamDiscoDuro=$6
10 rutaImagenIso=$7
11
12 # Crea la máquina virtual
13 VBoxManage createvm --name "$nom" --ostype "$tipoSo" --register
14
15 # Agregamos la cantidad de cpus, ram y vram a la máquina virtual
16 VBoxManage modifyvm "$nom" --cpus "$cpus" --memory "$memoriaRam" --vram "$memoriaDeVideo"
17
18 # Configurar la red virtual
19 VBoxManage modifyvm "$nom" --nic1 nat
20
21 # Crear el disco duro virtual
22 VBoxManage createhd --filename "/home/gemma/maquinas/$nom/$nom.vdi" --size "$tamDiscoDuro" --variant Standard
23 # Agregamos el controlador SATA del disco virtual
24 VBoxManage storagectl "$nom" --name "SATA Controller" --add sata --bootable on
25
26 # Conectar el disco duro virtual al controlador SATA
27 VBoxManage storageattach "$nom" --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium "/home/gemma/maquinas/$nom/$nom.vdi"
28
29 # Añadimos el controlador IDE
30 VBoxManage storagectl "$nom" --name "IDE Controller" --add ide
31
32 # Agregamos la imagen ISO al controlador IDE
33 VBoxManage storageattach "$nom" --storagectl "IDE Controller" --port 0 --device 0 --type dvddrive --medium "$rutaImagenIso"
34
35 # Permitimos la visualización
36 VBoxManage modifyvm "$nom" --vrde on
37
38 # Mostrar la información de la configuración de la VM
39 VBoxManage showvminfo "$nom"

```

```

$ script.sh
1 #!/bin/bash
2
3 # Argumentos del script
4 nom=$1
5 tipoSo=$2
6 cpus=$3
7 memoriaRam=$4
8 memoriaDeVideo=$5
9 tamDiscoDuro=$6
10 rutaImagenIso=$7
11
12 # Crea la máquina virtual
13 VBoxManage createvm --name "$nom" --ostype "$tipoSo" --register
14
15 # Agregamos la cantidad de cpus, ram y vram a la máquina virtual
16 VBoxManage modifyvm "$nom" --cpus "$cpus" --memory "$memoriaRam" --vram "$memoriaDeVideo"
17
18 # Configurar la red virtual
19 VBoxManage modifyvm "$nom" --nic1 nat
20
21 # Crear el disco duro virtual
22 VBoxManage createhd --filename "/home/gemma/maquinas/$nom/$nom.vdi" --size "$tamDiscoDuro" --variant Standard
23 # Agregamos el controlador SATA del disco virtual
24 VBoxManage storagectl "$nom" --name "SATA Controller" --add sata --bootable on
25
26 # Conectar el disco duro virtual al controlador SATA
27 VBoxManage storageattach "$nom" --storagectl "SATA Controller" --port 0 --device 0 --type hdd --medium "/home/gemma/maquinas/$nom/$nom.vdi"
28
29 # Añadimos el controlador IDE
30 VBoxManage storagectl "$nom" --name "IDE Controller" --add ide
31
32 # Agregamos la imagen ISO al controlador IDE
33 VBoxManage storageattach "$nom" --storagectl "IDE Controller" --port 0 --device 0 --type dvddrive --medium "$rutaImagenIso"
34
35 # Permitimos la visualización
36 VBoxManage modifyvm "$nom" --vrde on
37
38 # Mostrar la información de la configuración de la VM
39 VBoxManage showvminfo "$nom"

```

Se visualiza el script.

```
Challenge1: bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

06:50:43 gemma@Maquina-Virtual-VirtualBox ~ → cd Documentos/Challenge1/
/home/gemma/Documents/Challenge1
06:50:48 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main X|→ ls
script.sh
06:50:48 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main X|→ code .
No se ha encontrado la orden «code», pero se puede instalar con:
sudo snap install code
06:50:53 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main X|→ git add.
git: 'add.' no es un comando de git. Mira 'git --help'.

El comando más similar es
add
07:12:34 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main X|→ git add .
07:12:40 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main X|→ git commit -m "scriptpara automatizar la creación de una maquina virtual"
[main (commit-raíz) 0aa9903] scriptpara automatizar la creación de una maquina virtual
1 file changed, 39 insertions(+)
create mode 100755 script.sh
07:15:40 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|→ git push
error: unable to read askpass response from '/usr/bin/ksshaskpass'
Username for 'https://github.com': ^C
07:15:54 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|→ git push
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 6 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 810 bytes | 810.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/gemmacastaneda9048/Challenge1
 * [new branch]      main -> main
07:22:27 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|→
```

Se añadió y subió el script al repositorio.

En las siguientes 4 imágenes se visualiza lo siguiente:

- Le pasamos el script el nombre de la máquina virtual, el tipo de sistema operativo, numero de CPUs, memoria RAM en Mega Bites, VRAM en Mega Bites, tamaño del almacenamiento interno y la ruta al archivo ISO.
- Se muestra después la configuración de la máquina virtual.

```

Challenge1 : bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
07:35:18 gemma@Maquina-Virtual-VirtualBox Challenge1 ±[main]~ ./script.sh xubuntu Ubuntu_64 4 4096 16 20480 "/home/gemma/Descargas/xubuntu-22.04.5-desktop-amd64.iso"
Virtual machine 'xubuntu' is created and registered.
UUID: e3626096-30eb-49cc-a734-f5beab3f93c3
Settings file: '/home/gemma/VirtualBox VMs/xubuntu/xubuntu.vbox'
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Medium created. UUID: dd6a06c3-a261-40a4-abdf-1ffd4d4a61a3
Name: xubuntu
Encryption: disabled
Groups: /
Guest OS: Ubuntu (64-bit)
UUID: e3626096-30eb-49cc-a734-f5beab3f93c3
Config file: /home/gemma/VirtualBox VMs/xubuntu/xubuntu.vbox
Snapshot folder: /home/gemma/VirtualBox VMs/xubuntu/Snapshots
Log folder: /home/gemma/VirtualBox VMs/xubuntu/Logs
Hardware UUID: e3626096-30eb-49cc-a734-f5beab3f93c3 I
Memory size: 4096MB
Page Fusion: disabled
VRAM size: 16MB
CPU exec cap: 100%
HPET: disabled
CPUProfile: host
Chipset: piix3
Firmware: BIOS
Number of CPUs: 4
PAE: enabled
Long Mode: enabled
Triple Fault Reset: disabled
APIC: enabled
X2APIC: enabled
Nested VT-x/AMD-V: disabled
CPUID Portability Level: 0
CPUID overrides: None
  
```

```

Challenge1: bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

Boot menu mode:      message and menu
Boot Device 1:       Floppy
Boot Device 2:       DVD
Boot Device 3:       HardDisk
Boot Device 4:       Not Assigned
ACPI:                enabled
IOAPIC:              enabled
BIOS APIC mode:      APIC
Time offset:         0ms
BIOS NVRAM File:     /home/gemma/VirtualBox VMs/xubuntu/xubuntu.nvram
RTC:                 local time
Hardware Virtualization: enabled
Nested Paging:       enabled
Large Pages:         enabled
VT-x VPID:           enabled
VT-x Unrestricted Exec.: enabled
AMD-V Virt. Vmsave/Vmload: enabled
IOMMU:               None
Paravirt. Provider:  Default
Effective Paravirt. Prov.: KVM
State:                powered off (since 2024-10-05T01:59:17.898000000)
Graphics Controller: VBoxVGA
Monitor count:       1
3D Acceleration:     disabled
2D Video Acceleration: disabled
Teleporter Enabled:   disabled
Teleporter Port:      0
Teleporter Address:   disabled
Teleporter Password:  disabled
Tracing Enabled:      disabled
Allow Tracing to Access VM: disabled
Tracing Configuration: disabled
Autostart Enabled:    disabled

```

```

Challenge1: bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

Autostart Enabled:    disabled
Autostart Delay:      0
Default Frontend:
VM process priority:  default
Storage Controllers:
#0: 'SATA Controller', Type: IntelAhci, Instance: 0, Ports: 30 (max 30), Bootable
  Port 0, Unit 0: UUID: dd6a06c3-a261-40a4-abdf-1ffd4d4a61a3
  Location: "/home/gemma/maquinas/xubuntu/xubuntu.vdi"
#1: 'IDE Controller', Type: PIIX4, Instance: 0, Ports: 2 (max 2), Bootable
  Port 0, Unit 0: UUID: 2e8f0baf-b4c5-439c-8b8e-2e5cad5d6a96
  Location: "/home/gemma/Descargas/xubuntu-22.04.5-desktop-amd64.iso"
NIC 1:                MAC: 0800278410DC, Attachment: NAT, Cable connected: on, Trace: off (file: none), Type: 82540EM,
  Reported speed: 0 Mbps, Boot priority: 0, Promisc Policy: deny, Bandwidth group: none
NIC 1 Settings: MTU: 0, Socket (send: 64, receive: 64), TCP Window (send:64, receive: 64)
NIC 2:                disabled
NIC 3:                disabled
NIC 4:                disabled
NIC 5:                disabled
NIC 6:                disabled
NIC 7:                disabled
NIC 8:                disabled
Pointing Device:      PS/2 Mouse
Keyboard Device:      PS/2 Keyboard
UART 1:               disabled
UART 2:               disabled
UART 3:               disabled
UART 4:               disabled
LPT 1:                disabled
LPT 2:                disabled
Audio:                enabled (Driver: Default, Controller: AC97, Codec: STAC9700)
Audio playback:        disabled
Audio capture:         disabled
Clipboard Mode:        disabled

```



```

Challenge1: bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

LPT 2:      disabled
Audio:      enabled (Driver: Default, Controller: AC97, Codec: STAC9700)
Audio playback: disabled
Audio capture: disabled
Clipboard Mode: disabled
Drag and drop Mode: disabled
VRDE:      enabled (Address 0.0.0.0, Ports 3389, MultiConn: off, ReuseSingleConn: off, Authentication type:
  null)
Video redirection: disabled
VRDE property : TCP/Ports = "3389"
VRDE property : TCP/Address = <not set>
OHCI USB:   disabled
EHCI USB:   disabled
xHCI USB:   disabled
USB Device Filters: <none>
Bandwidth groups: <none>
Shared folders: <none>
Recording enabled: no
Recording screens: 1
  Screen 0:
    Enabled: yes
    ID: 0
    Record video: yes
    Destination: File
    File: /home/gemma/VirtualBox VMs/xubuntu/xubuntu-screen0.webm
    Options: vc_enabled=true,ac_enabled=false,ac_profile=med
    Video dimensions: 1024x768
    Video rate: 512kbps
    Video FPS: 25fps
* Guest:
Configured memory balloon: 0MB

07:59:18 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|~

```

```

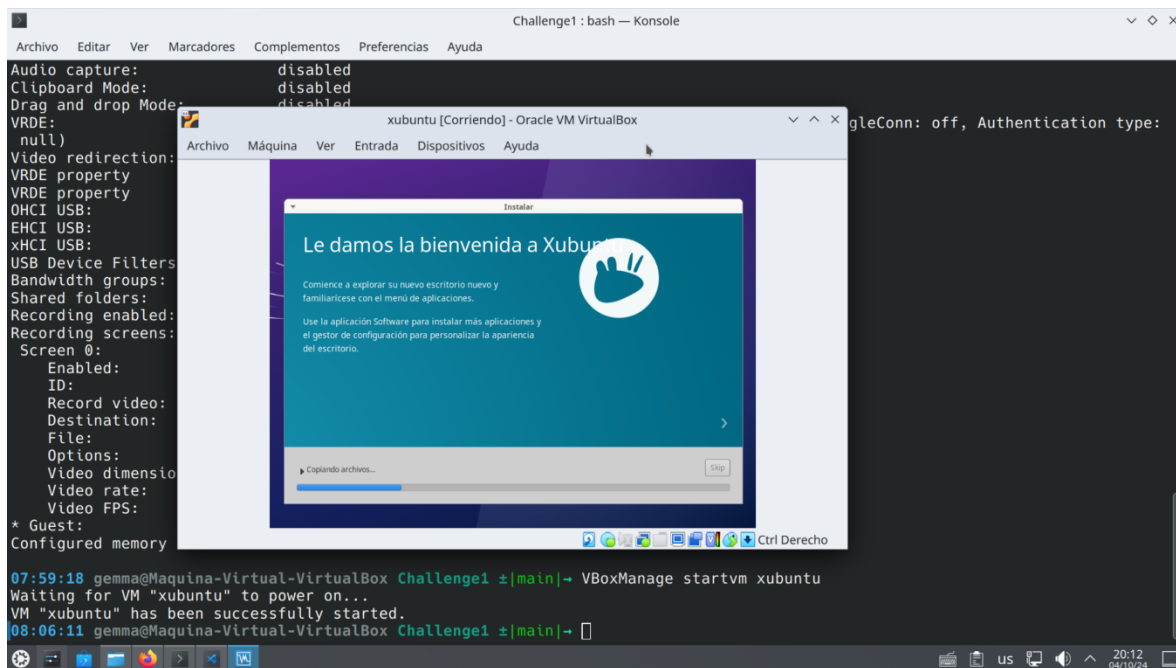
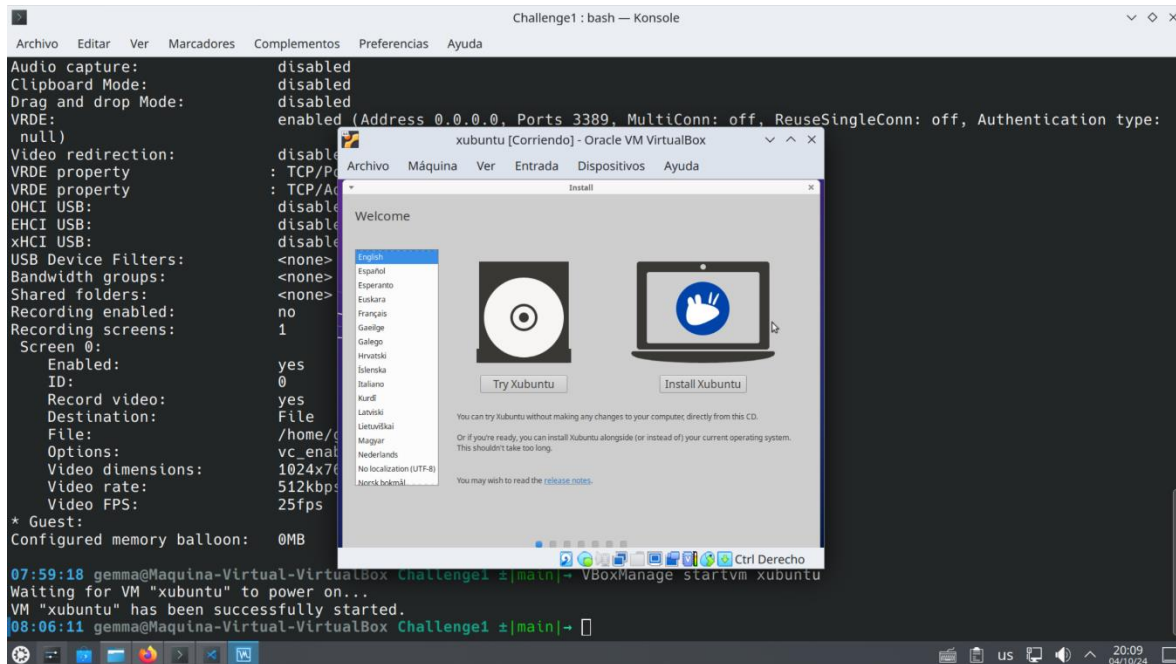
Challenge1: bash — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda

Audio capture: disabled
Clipboard Mode: disabled
Drag and drop Mode: disabled
VRDE:      enabled (Address 0.0.0.0, Ports 3389, MultiConn: off, ReuseSingleConn: off, Authentication type:
  null)
Video redirection: disabled
VRDE property : TCP/Ports =
VRDE property : TCP/Address
OHCI USB:   disabled
EHCI USB:   disabled
xHCI USB:   disabled
USB Device Filters: <none>
Bandwidth groups: <none>
Shared folders: <none>
Recording enabled: no
Recording screens: 1
  Screen 0:
    Enabled: yes
    ID: 0
    Record video: yes
    Destination: File
    File: /home/gemma/V
    Options: vc_enabled=tr
    Video dimensions: 1024x768
    Video rate: 512kbps
    Video FPS: 25fps
* Guest:
Configured memory balloon: 0MB

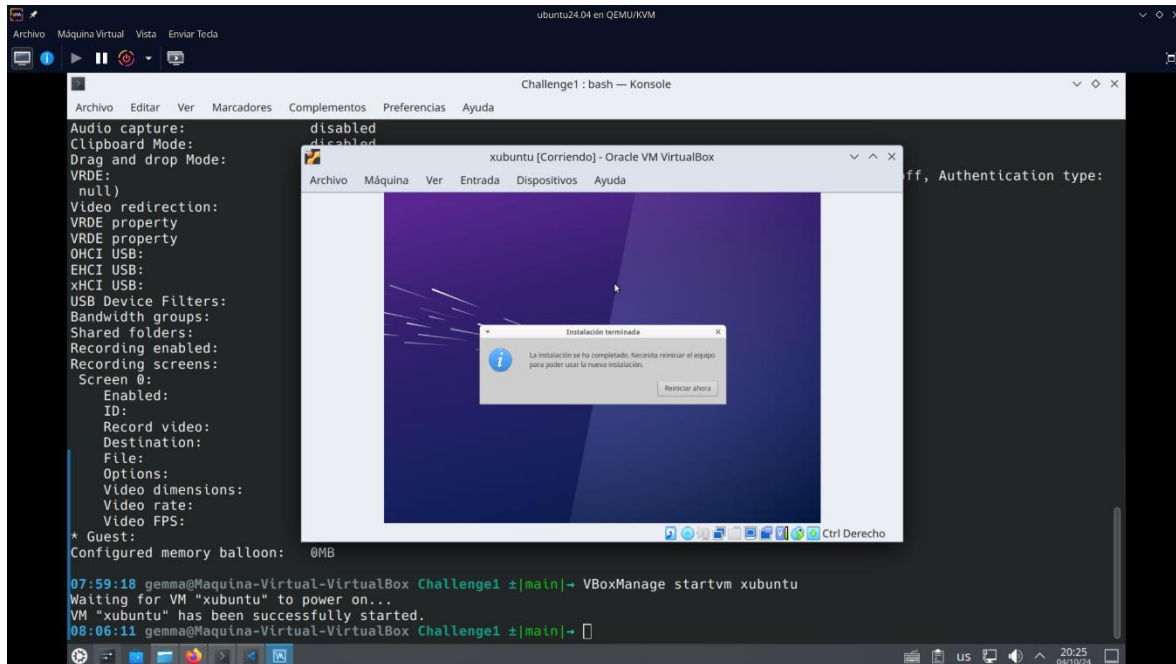
07:59:18 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|~ VBoxManage startvm xubuntu
Waiting for VM "xubuntu" to power on...
VM "xubuntu" has been successfully started.
08:06:11 gemma@Maquina-Virtual-VirtualBox Challenge1 ±|main|~

```

Utilizamos el comando `VBoxManage startvm xubuntu` para arrancar la máquina virtual, se visualiza el resultado (la nueva máquina virtual en ejecución).



Continuamos con la instalación de la nueva máquina virtual Xubuntu (en las 2 imágenes de arriba).



Se terminó la instalación de la máquina virtual.

Conclusiones

En este Challenge se logró automatizar el proceso de creación y configuración de una Máquina Virtual utilizando un script en Linux y la herramienta VBoxManage de VirtualBox. La automatización de tareas como esta es muy útil porque ahorra tiempo y reduce errores humanos al hacer que todo el proceso sea más rápido y eficiente.

Crear máquinas virtuales manualmente puede ser complicado y tomar mucho tiempo, pero con un script como el que desarrollamos, se pueden generar de manera uniforme y sin fallos, lo que es ideal cuando se trabaja en proyectos que requieren espacios o configuraciones virtuales que se crean para ejecutar programas, realizar pruebas o trabajar en el desarrollo de software. Además, el uso de scripts ayuda a estandarizar el proceso, asegurando que las máquinas virtuales se creen siempre con los mismos parámetros, lo que facilita la administración de recursos y la repetición del proceso en otros proyectos.

Este desafío mostró cómo la automatización no solo facilita la vida de la/el desarrollador(a) o administrador(a) de sistemas, sino que también garantiza precisión y eficiencia en tareas importantes como la creación de infraestructuras virtuales.

Enlace

<https://github.com/gemmacastaneda9048/Challenge1>