

Obligatorio

Taller de Servidores Linux

Fecha de entrega: 06 de agosto de 2024

UNIVERSIDAD ORT URUGUAY

Integrantes:

Ezquiel Origüela – N° de estudiante: 280758

Maximiliano Robles – N° de estudiante: 290963

Docente:

Enrique Verdes

Contenido

Parte A 3

Parte B 4

 Punto 1.....4

 Parte 28

Parte C12

 Parte 2 ¡Error! Marcador no definido.

Parte A

Instalamos el servidor configurado de igual manera que el establecido en la [Parte C,1](#) con CentOS 9 (RAM,Disco,etc) y posteriormente de terminada la instalación del sistema operativo, instalamos GIT y KeyGen.

En primera parte instalamos SSH, generemos una clave publica con las opciones -C (agregamos el correo) y -t rsa (utilizar el algoritmo de encriptación rsa) y por ultimo verificamos la generacion de la misma.

```
“sudo yum install openssh”
```

```
“ssh-keygen -t rsa -C "ezequiel.origuela1@gmail.com"”
```

```
“cat ~/.ssh/id_rsa.pub”
```

Clave publica

```
lsysadm@controller ~]$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQChESG0MWB4axWd00N59KnmR5sAkSuFWMIfamoUaAV+ECzsD8noBr3Z38xKnRdMdnHobGfFqGZB73XPMRr0+6YEE+7  
BUyHW8bHc6UUXMcE0Cv5LRiazF90fbzcEW4vDpmys7oq5An79ZigFzcECTlkyKQYGYhNiuJcGY4L3K4/dumdb/9bXYheSE/nLHIp9LDXWADad/WZqYTqg8zJy3njXr9  
Pvro0iPVJ2ydG3+KT84t19wPm2Y4eZEH+KKmBAJrofG3USjZ8n7Pjz7V0ZJpSPo0HyeVZsK3AqHx2+jU4JuNM1G0oJcRxLyzmyjgzSDBG1sCF4rXBtH+3VANWQ0s5  
uHWZ9QnE5NMBGLkuYB5hSTUUX6hcFJIiFxa/5ZGqHS0qgjCn2pn6LZEhPbggp1ZGf0W2qRbhX0X+kr4erF8eYbMxMpZL5j+ZI8BDgJ+uCcnLBLGRiM4u9BwAKjnM1t  
QKFMu3t7PelsTvjiakKySpvx2teLVJgdnQ1/xAbQ6zU= ezequiel.origuela1@gmail.com
```

Instalamos Git

Realizamos la instalación de Git y agregamos nuestro nombre y email para firmarlo, creamos una cuenta en la web y en el apartado de “SSH and GPG keys” ingresamos nuestra clave publica previamente generada. Posteriormente, creamos un repositorio y agreeamos nuestro nombre e email. Y por ultimo realizamos un clon del repositorio para tenerlo de manera local, creando de esta manera un nuevo directorio

```
“sudo yum install git”
```

```
“git config --global user.name "Ezequiel Origuela" ”
```

```
“git config --global user.email "ezequiel.origuela1@gmail.com" ”
```

```
“git clone git@github.com:eoriguela/TallerJulio2024.git”
```

Instalación Ansible

Para la instalación de Ansible debemos de descargar previamente el paquete de Python.

```
“sudo dnf install python3-pip”
```

```
“pip install pipx” // Instala el paquete Pipx para ejecutar apps Python
```

```
“pipx ensurepath” // Asegura que el directorio de instalación sea en la  
dirección del sistema, con el fin de que se puedan  
instalar desde cualquier lugar con una consola
```

```
“pipx install ansible-core” // Instala la base de automatización para  
ansible
```

“pipx inject ansible-core argcomplete”	// Inyecta el paquete “argcomplete” que se encarga de brindar autocompletado a los comandos
“pipx install ansible-lint”	// Instala el paquete “ansible-lint”
“pipx inject ansible-core ansible-lint”	// Inyecta el paquete “ansible-lint” que se encarga de verificar el código en Ansible para que cumpla con unos requisitos mínimos
“activate-global-python-argcomplete -user”	// Activa de manera global el autocompletado de argumentos para Python
“source /home/sysadmin/.bash_completion”	// Carga las definiciones autocompletado para bash, permitiendo el autocompletado en el terminal

Parte B

Punto 1

En primera instancia, desde nuestro virtualizador configuramos una máquina virtual nueva con las especificaciones de hardware propuestas

Resumen

La siguiente tabla resume la configuración que ha elegido para la nueva máquina virtual. Cuando esté conforme con la configuración presione Finalizar para crear la máquina virtual. También puede volver atrás y modificar la configuración.

✱ Nombre y tipo de SO de la máquina	
Nombre de máquina	OBG - Server01
Carpeta de la máquina	C:/Users/Ezequiel/VirtualBox VMs/OBG - Server01
Imagen ISO	C:/Taller ASLX/CentOS-Stream-9-latest-x86_64-dvd1.iso
Tipo de SO invitado	Red Hat (64-bit)
Omitir instalación desatendida	true
■ Hardware	
Memoria base	2048
Procesador(es)	1
Habilitar EFI	false
🗨 Disco	
Tamaño de disco	13,00 GB
Reservar tamaño completo	false

Adaptador NAT y Host-Only


Red


Adaptador 1: Intel PRO/1000 MT Desktop (NAT)
 Adaptador 2: Intel PRO/1000 MT Desktop (Adaptador solo anfitrión, «VirtualBox Host-Only Ethernet Adapter»)


Luego de agregar la iso de CentOS 9 configuramos el lenguaje en ingles y las siguientes configuraciones

Password Root

The root account is used for administering the system. Enter a password for the root user.

Root Password: 

 **Fair**

Confirm: 

☐ Lock root account

☐ Allow root SSH login with password


User Creation > Usuario no Root, administrador


Full name


User name

☒ Make this user administrator

☒ Require a password to use this account

Password 

 **Good**

Confirm password 

Network & Host Name>Host Name

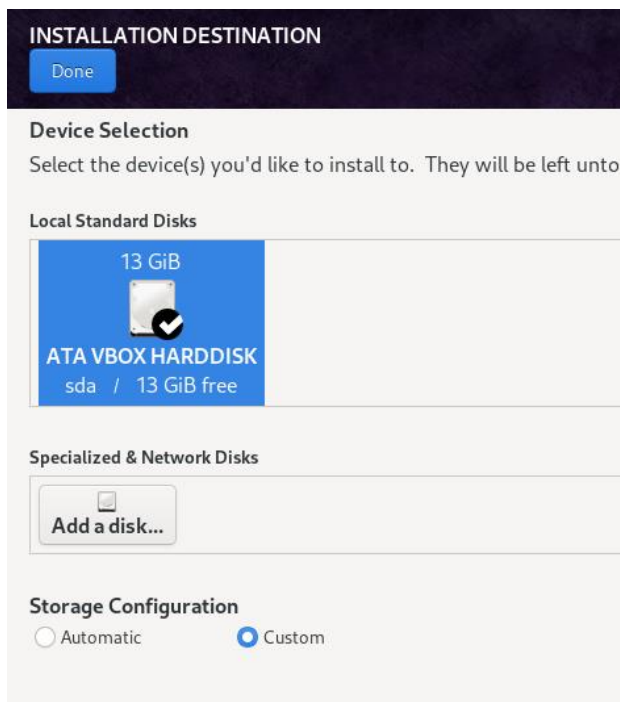
Host Name:

Software Selection > Minimal Install

Base Environment

- ☐ **Server with GUI**
An integrated, easy-to-manage server with a graphical interface.
- ☐ **Server**
An integrated, easy-to-manage server.
- ☒ **Minimal Install**
Basic functionality.
- ☐ **Workstation**
Workstation is a user-friendly desktop system for laptops and PCs.
- ☐ **Custom Operating System**
Basic building block for a custom RHEL system.
- ☐ **Virtualization Host**
Minimal virtualization host.

Installation Destination



Posteriormente pinchamos en “Click here to create them automatically” para crear las particiones y los configuramos tal que quedé como se planteaba.

Installation Destination > Manual Partition

▼ New CentOS Stream 9 Installation		
SYSTEM		
/		7 GiB
cs_10-root		
/var		3 GiB
cs_10-var		
/boot		1024 MiB
sda1		
swap		2 GiB >
cs_10-swap		

Parte 2

En primera instancia, desde nuestro virtualizador configuramos una máquina virtual nueva con las especificaciones de hardware propuestas

Resumen

La siguiente tabla resume la configuración que ha elegido para la nueva máquina virtual. Cuando esté conforme con la configuración presione Finalizar para crear la máquina virtual. También puede volver atrás y modificar la configuración.

✱ Nombre y tipo de SO de la máquina	
Nombre de máquina	OBG - Server02
Carpeta de la máquina	C:/Users/Ezequiel/VirtualBox VMs/OBG - Server02
Imagen ISO	C:/Taller ASLX/ubuntu-24.04-live-server-amd64.iso
Tipo de SO invitado	Ubuntu (64-bit)
Omitir instalación desatendida	true
🖥 Hardware	
Memoria base	2048
Procesador(es)	1
Habilitar EFI	false
💿 Disco	
Tamaño de disco	13.00 GB
Reservar tamaño completo	false

Red

Adaptador 1: Intel PRO/1000 MT Desktop (NAT)
Adaptador 2: Intel PRO/1000 MT Desktop (Adaptador solo anfitrión, «VirtualBox Host-Only Ethernet Adapter»)

Posteriormente de haber inicializado la VM con las configuraciones de idioma y teclado de nuestra preferencias.

Type of Installation

Choose the type of installation

Choose the base for the installation.

(X) Ubuntu Server

The default install contains a curated set of packages that provide a comfortable experience for operating your s

() Ubuntu Server (minimized)

This version has been customized to have a small runtime footprint in environments where humans are not expected in.

Additional options

[] Search for third-party drivers

This software is subject to license terms included with its documentation. Some is proprietary. Third-party drive should not be installed on systems that will be used for FIPS or the real-time kernel.

Los dos adaptadores de red agregados previamente en el virtualizador

Network Configuration

```
Network configuration [ Help ]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient
access for updates.

NAME    TYPE  NOTES
[ enp0s3  eth  -           ▶ ]
DHCPv4  10.0.2.15/24
08:00:27:43:30:f5 / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)

[ enp0s8  eth  -           ▶ ]
DHCPv4  192.168.56.105/24
08:00:27:a4:9b:4b / Intel Corporation / 82540EM Gigabit Ethernet Controller (PRO/1000 MT Desktop Adapter)

[ Create bond ▶ ]
```

Storage configuration

```
Guided storage configuration

Configure a guided storage layout, or create a custom one:

( ) Use an entire disk

    [ VBOX_HARDDISK_VB7afe85b3-6b121c7d local disk 13.009G ▼ ]

    [ ] Set up this disk as an LVM group

        [ ] Encrypt the LVM group with LUKS

            Passphrase:

            Confirm passphrase:

                [ ] Also create a recovery key
                    The key will be stored as ~/recovery-key.txt
                    /var/log/installer/ in the target system.

(X) Custom storage layout
```

Creación de partición /boot

Adding GPT partition to VBOX_HARDDISK_VBdae7ea3f-323cb17d

Size (max 13.007G):

Format: [xfs ▼]

Mount: [/boot ▼]

[Create]
[Cancel]

Creación de partición sin formato para LVM

Adding GPT partition to VBOX_HARDDISK_VB7afe85b3-6b121c7d

Size (max 12.006G):

Format: [Leave unformatted ▼]

Mount: [/ ▼]

[Create]
[Cancel]

Creado el LVM y creamos manualmente el resto de las particiones

Create LVM volume group

Name:

Devices: VBOX_HARDDISK_VB7afe85b3-6b121c7d 13.009G
local disk
[X] partition 3 12.006G
unused partition of local disk

Size: 12.003G

[] Create encrypted volume

Passphrase:

Confirm passphrase:

[] Also create a recovery key:
The key will be stored as
~/recovery-key-{name}.txt in the live system
and will be copied to /var/log/installer/ in
the target system.

[Create]
[Cancel]

Storage configuration

Storage configuration

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE	TYPE	
[/	7.000G	new xfs	new LVM logical volume		▶]
[/boot	1.000G	new xfs	new partition of local disk		▶]
[/var	3.000G	new xfs	new LVM logical volume		▶]
[SWAP	2.000G	new swap	new LVM logical volume		▶]

AVAILABLE DEVICES

DEVICE	TYPE	SIZE	
[vg0 (new)	LVM volume group	12.003G	▶]
free space		4.000M	▶]

[Create software RAID (md) ▶]

[Create volume group (LVM) ▶]

USED DEVICES

DEVICE	TYPE	SIZE	
[vg0 (new)	LVM volume group	12.003G	▶]
root	new, to be formatted as xfs, mounted at /	7.000G	▶]
var	new, to be formatted as xfs, mounted at /var	3.000G	▶]
swap	new, to be formatted as swap	2.000G	▶]
[VBOX_HARDDISK_VB7afe85b3-6b121c7d	local disk	13.009G	▶]
partition 1	new, BIOS grub spacer	1.000M	▶]
partition 2	new, to be formatted as xfs, mounted at /boot	1.000G	▶]
partition 3	new, PV of LVM volume group vg0	12.006G	▶]

Profile Configuration

Profile configuration [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on a later screen, but a password is still needed for sudo.

Your name: Sysadmin

Your servers name: server02
The name it uses when it talks to other computers.

Pick a username: sysadmin

Choose a password: *****

Confirm your password: *****

Instalamos OpenSSH ya que no viene por defecto

SSH configuration

You can choose to install the OpenSSH server package to enable secure remote access to your server.

[X] Install OpenSSH server

[X] Allow password authentication over SSH

[Import SSH key ►]

AUTHORIZED KEYS

No authorized key

Posteriormente de realizar estos pasos y las actualizaciones pertinentes, seleccionamos "reboot now"

Parte C

En primera instancia debemos de copiar las claves publicas al equipo "Controller" para de esa manera poder realizar cambios en los playbooks, commits, etc.

```
"ssh-copy-id sysadmin@192.168.56.110"    /// Copiamos la llave publica de Server01
```

```
"ssh-copy-id sysadmin@192.168.56.111"    /// Copiamos la llave publica de Server02
```

Posterior a haber realizado cambios sobre los playbooks correspondientes realizamos una verificación para que estén correctamente indentados y establecidos. Y luego un chequeo sobre la ejecución. En este caso para el playbook de hardeining.

```
"ansi+ble-playbook -i inventory/servidores.toml hardening.yml --syntax-check"
```

```
"ansible-playbook -i inventory/servidores.toml hardening.yml --ask-become-pass --check"
```

```
PLAY RECAP *****
server01           : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server02           : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Tener en cuenta que no todos los modulos soportan el check mode, lo verificamos directame sobre la correspondiente documentación

Realizados los chequeos correspondientes, realizaremos el commit de los cambios realizados.

<code>"git add ."</code>	<code>// Agregamos los archivos a realizar el commit, en este caso todo el repositorio. Se puede agregar individualmente cada directorio si se quisiera</code>
<code>"git status"</code>	<code>// Verificamos que todos los archivos estén bien y queramos realizar</code>
<code>"git commit "</code>	<code>// Realizamos el commit, nos quedaría como Branch</code>
<code>"git push"</code>	<code>// Finalmente "pusheamos" el contenido local al del repositorio en GitHub</code>

Realizamos el los playbooks correspondientes para el correcto despliegue de el webservice y la base de datos. Y tambien en caso de no tenerlas instalamos GIT y NANO

Posterior a la realizacion del playbook para el webservice y realizamos el check y luego lo pasamos con:

```
"ansible-playbook -i inventory/servidores.toml webserver.yml"
```

Para la posterior instalación de la base de datos mediante el automatismo Ansible debemos de instalar las collections Y también inyectamos mysql, esto se realiza con:

```
"ansible-galaxy collection install -r Collections/requirements.yml"
```

```
// Instala las colecciones previamente estipuladas para el funcionamiento de la base de datos
```

Cambiamos la configuración de `bind_address` del archivo a `0.0.0.0` para que nos permita conexiones desde el servidor controller

`“sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf”`