

## **Tarea 6: Instalación de Wildfly en linux**

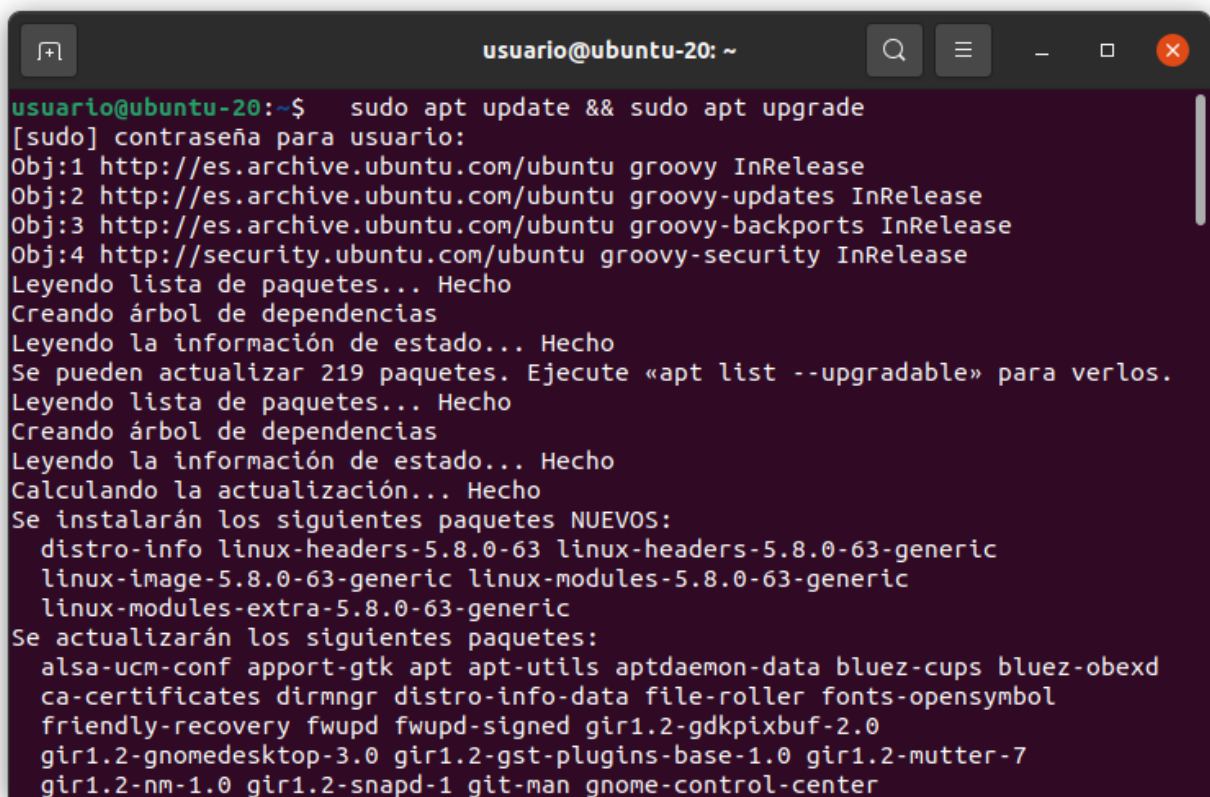


## **Índice**

1. Actualización de repositorios
2. Instalación de WildFly
3. Configuración de Wildfly
4. Acceso a Wildfly
5. Como añadir usuarios a WildFly
6. Gestionar la consola de forma remota.

# 1. Actualización de repositorios

Primero vamos a actualizar el repositorio y el sistema operativo para que la instalación de apache sea mas segura.



```
usuario@ubuntu-20: ~  
usuario@ubuntu-20:~$ sudo apt update && sudo apt upgrade  
[sudo] contraseña para usuario:  
Obj:1 http://es.archive.ubuntu.com/ubuntu groovy InRelease  
Obj:2 http://es.archive.ubuntu.com/ubuntu groovy-updates InRelease  
Obj:3 http://es.archive.ubuntu.com/ubuntu groovy-backports InRelease  
Obj:4 http://security.ubuntu.com/ubuntu groovy-security InRelease  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se pueden actualizar 219 paquetes. Ejecute «apt list --upgradable» para verlos.  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Calculando la actualización... Hecho  
Se instalarán los siguientes paquetes NUEVOS:  
  distro-info linux-headers-5.8.0-63 linux-headers-5.8.0-63-generic  
  linux-image-5.8.0-63-generic linux-modules-5.8.0-63-generic  
  linux-modules-extra-5.8.0-63-generic  
Se actualizarán los siguientes paquetes:  
  alsa-ucm-conf apport-gtk apt apt-utils aptdaemon-data bluez-cups bluez-obexd  
  ca-certificates dirmngr distro-info-data file-roller fonts-opensymbol  
  friendly-recovery fwupd fwupd-signed gir1.2-gdkpixbuf-2.0  
  gir1.2-gnomedesktop-3.0 gir1.2-gst-plugins-base-1.0 gir1.2-mutter-7  
  gir1.2-nm-1.0 gir1.2-snapd-1 git-man gnome-control-center
```

## 2. Instalación de WildFly

Ahora vamos a descargar el paquete de wildfly

```
-----
usuario@ubuntu-20:~$ wget https://github.com/wildfly/wildfly/releases/download/25.0.0.Final/wildfly-25.0.0.Final.tar.gz
--2021-10-23 14:29:59-- https://github.com/wildfly/wildfly/releases/download/25.0.0.Final/wildfly-25.0.0.Final.tar.gz
Resolviendo github.com (github.com)... 140.82.121.4
Conectando con github.com (github.com)[140.82.121.4]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Ubicación: https://github-releases.githubusercontent.com/764708/ef6a6350-a4cc-4a89-a20b-cc175681c590?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20211023%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20211023T122959Z&X-Amz-Expires=300&X-Amz-Signature=f6d34230384e7de937a9d9ebc0f1f839d08606601a0c125f6e364dad275b689d&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=764708&response-content-disposition=attachment%3B%20filename%3Dwildfly-25.0.0.Final.tar.gz&response-content-type=application%2Foctet-stream [siguiente]
```

Después crearemos el nuevo grupo y usuario wildfly

```
usuario@ubuntu-20:~$ sudo groupadd -r wildfly
usuario@ubuntu-20:~$ sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly
```

-----

Luego descomprimos el paquete que acabamos de descargar directamente en su ubicación definitiva

```
usuario@ubuntu-20:~$ tar -xvzf wildfly-25.0.0.Final.tar.gz
usuario@ubuntu-20:~$ sudo mv wildfly-25.0.0.Final /opt/wildfly-25.0.0.Final
```

-----

Y ahora crearemos un link simbólico al directorio

```
usuario@ubuntu-20:~$ sudo ln -s /opt/wildfly-25.0.0.Final /opt/wildfly
```

-----

Luego daremos acceso al usuario y grupo wildfly

```
usuario@ubuntu-20:~$ sudo chown -R wildfly:wildfly /opt/wildfly
[sudo] contraseña para usuario:
usuario@ubuntu-20:~$ sudo chown -R wildfly:wildfly /opt/wildfly/
```

-----

Después configuraremos e iniciaremos el servicio.

```
usuario@ubuntu-20:~$ sudo mkdir -p /etc/wildfly
usuario@ubuntu-20:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
```

-----

Podemos visualizar el fichero de configuración de arranque, donde veremos que tiene por defecto el arranque standalone

```
usuario@ubuntu-20:~$ sudo nano /etc/wildfly/wildfly.conf
```

```
GNU nano 5.2 /etc/wildfly/wildfly.conf
# The configuration you want to run
WILDFLY_CONFIG=standalone.xml

# The mode you want to run
WILDFLY_MODE=standalone

# The address to bind to
WILDFLY_BIND=0.0.0.0
```

-----

Continuaremos lanzando los siguientes comandos para configurar el arranque

```
usuario@ubuntu-20:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh
/opt/wildfly/bin/
usuario@ubuntu-20:~$
usuario@ubuntu-20:~$ sudo sh -c 'chmod +x /opt/wildfly/bin/*.sh'
usuario@ubuntu-20:~$
usuario@ubuntu-20:~$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.s
ervice /etc/systemd/system/
usuario@ubuntu-20:~$
usuario@ubuntu-20:~$ sudo systemctl daemon-reload
```

Ahora podremos iniciar el servicio con el siguiente comando

```
usuario@ubuntu-20:~$ sudo systemctl start wildfly
```

Y luego comprobaremos el estado del servicio

```
usuario@ubuntu-20:~$ sudo systemctl status wildfly
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; disabled; vendor pres>
   Active: active (running) since Sat 2021-10-23 15:26:29 CEST; 4min 45s ago
 Main PID: 25255 (launch.sh)
    Tasks: 51 (limit: 2308)
   Memory: 274.8M
    CGroup: /system.slice/wildfly.service
           └─25255 /bin/bash /opt/wildfly/bin/launch.sh standalone standalone>
             └─25256 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml ->
               └─25349 java -D[Standalone] -server -Xms64m -Xmx512m -XX:Metaspace>

oct 23 15:26:29 ubuntu-20 systemd[1]: Started The WildFly Application Server.
```

Posteriormente haremos que el servicio se inicie automáticamente

```
usuario@ubuntu-20:~$ sudo systemctl enable wildfly
Created symlink /etc/systemd/system/multi-user.target.wants/wildfly.service → /etc/systemd/system/wildfly.service.
```

### 3. Configuración de WildFly

Vamos a permitir el tráfico por un puerto específico, por ejemplo el puerto **8083**. Hemos de modificar el puerto **8080**, por **8083**. Para ello hemos de modificar la etiqueta en el fichero de configuración

```
usuario@ubuntu-20:~$ sudo nano /opt/wildfly/standalone/configuration/standalone.xml
```

```
GNU nano 5.2 /opt/wildfly/standalone/configuration/standalone.xml
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:127.0.0.1}"/>
  </interface>
</interfaces>
<socket-binding-group name="standard-sockets" default-interface="public" po>
  <socket-binding name="ajp" port="{jboss.ajp.port:8009}"/>
  <socket-binding name="http" port="{jboss.http.port:8080}"/>
  <socket-binding name="https" port="{jboss.https.port:8443}"/>
  <socket-binding name="management-http" interface="management" port="{j>
  <socket-binding name="management-https" interface="management" port="{j>
  <socket-binding name="txn-recovery-environment" port="4712"/>
  <socket-binding name="txn-status-manager" port="4713"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="{jboss.mail.server.host:localhost}" port>
  </outbound-socket-binding>
</socket-binding-group>
```

^G Ayuda    ^O Guardar    ^W Buscar    ^K Cortar    ^T Ejecutar    ^C Ubicación  
 ^X Salir    ^R Leer fich.    ^\ Reemplazar    ^U Pegar    ^J Justificar    ^\_ Ir a línea

```
<socket-binding name="ajp" port="{jboss.ajp.port:8009}"/>
<socket-binding name="http" port="{jboss.http.port:8083}"/>
<socket-binding name="https" port="{jboss.https.port:8443}"/>
```

y permitimos el tráfico con el puerto:

```
usuario@ubuntu-20:~$ sudo ufw allow 8083/tcp
Regla añadida
Regla añadida (v6)
```



## 4. Acceso a WildFly

Ahora vamos a acceder a Wildfly



## 5. Como añadir Usuarios a WildFly

Primero crearemos el usuario administrador

```
usuario@ubuntu-20:~$ sudo /opt/wildfly/bin/add-user.sh
```

What type of user do you wish to add?

- a) Management User (mgmt-users.properties)
- b) Application User (application-users.properties)

(a): a

Enter the details of the new user to add.

Using realm 'ManagementRealm' as discovered from the existing property files.

Username : admin123

Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.

- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)

Password :

WFLYDM0102: Password should have at least 1 non-alphanumeric symbol.

Are you sure you want to use the password entered yes/no? yes

Re-enter Password :

What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:

About to add user 'admin123' for realm 'ManagementRealm'

Is this correct yes/no? yes

Added user 'admin123' to file '/opt/wildfly-25.0.0.Final/standalone/configuration/mgmt-users.properties'

Added user 'admin123' to file '/opt/wildfly-25.0.0.Final/domain/configuration/mgmt-users.properties'

Added user 'admin123' with groups to file '/opt/wildfly-25.0.0.Final/standalone/configuration/mgmt-groups.properties'

Added user 'admin123' with groups to file '/opt/wildfly-25.0.0.Final/domain/configuration/mgmt-groups.properties'

Is this new user going to be used for one AS process to connect to another AS process?

e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server Jakarta Enterprise Beans calls.

yes/no? yes

To represent the user add the following to the server-identities definition <secret value="QURNSU4xMjM=" />

## 6. Gestionar la consola de forma remota

Ahora pasaremos al archivo de configuración

```
usuario@ubuntu-20:~$ sudo nano /etc/wildfly/wildfly.conf
```

GNU nano 5.2

/etc/wildfly/wildfly.conf

## The configuration you want to run

WILDFLY\_CONFIG=standalone.xml

# The mode you want to run

WILDFLY\_MODE=standalone

# The address to bind to

WILDFLY\_BIND=0.0.0.0



Y buscamos en otro fichero y comprobamos que este lo siguiente

```
usuario@ubuntu-20:~$ sudo nano /opt/wildfly/bin/launch.sh
```

```
GNU nano 5.2 /opt/wildfly/bin/launch.sh
#!/bin/bash

if [ "x$WILDFLY_HOME" = "x" ]; then
    WILDFLY_HOME="/opt/wildfly"
fi

if [[ "$1" == "domain" ]]; then
    $WILDFLY_HOME/bin/domain.sh -c $2 -b $3
else
    $WILDFLY_HOME/bin/standalone.sh -c $2 -b $3
fi
```

Y como último paso realizaremos los siguientes comandos.

```
usuario@ubuntu-20:~$ sudo systemctl restart wildfly
usuario@ubuntu-20:~$ sudo nano /etc/systemd/system/wildfly.service
usuario@ubuntu-20:~$ sudo systemctl daemon-reload
usuario@ubuntu-20:~$ sudo systemctl restart wildfly
```