

TAREA OPTATIVA DEL WILDFLY

Empezamos creando una cuenta wildfly con los siguientes comandos para administrar:

```
administrador@UBUNTUDAW:~$ sudo groupadd -r wildfly
administrador@UBUNTUDAW:~$ sudo useradd -r -g wildfly -d /opt/wildfly -s /sbin/nologin wildfly
```

Nos movemos al directorio /tmp y descargamos la última versión del wildfly:

```
administrador@UBUNTUDAW:~$ cd /tmp
administrador@UBUNTUDAW:/tmp$ wget https://download.jboss.org/wildfly/16.0.0.Final/wildfly-16.0.0.Final.tar.gz
```

Lo descomprimos con gzip con: “tar xvf wildfly-16.0.0.Final.tar.gz.

Creamos una carpeta para el wildfly en /opt y cambiamos el propietario:

```
administrador@UBUNTUDAW:/tmp$ sudo mv wildfly-16.0.0.Final/ /opt/wildfly
[sudo] contraseña para administrador:
administrador@UBUNTUDAW:/tmp$ sudo chown -RH wildfly: /opt/wildfly
```

Y creamos la carpeta de servicio wildfly en /etc:

```
administrador@UBUNTUDAW:/tmp$ sudo mkdir -p /etc/wildfly
```

Copiamos los archivos de configuración y ejecutables a los directorios anteriores:

```
administrador@UBUNTUDAW:/tmp$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.conf /etc/wildfly/
administrador@UBUNTUDAW:/tmp$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/launch.sh /opt/wildfly/bin/
```

Y los scripts sean ejecutables, y los movemos a /etc/systemd/system

```
administrador@UBUNTUDAW:/tmp$ sudo sh -c 'chmod +x /opt/wildfly/bin/*.sh'
administrador@UBUNTUDAW:/tmp$ sudo cp /opt/wildfly/docs/contrib/scripts/systemd/wildfly.service /etc/systemd/system/
```

Usamos los siguientes comandos para parar el servicio (por si ya estaba corriendo) lo iniciamos y habilitamos:

```
administrador@UBUNTUDAW:/tmp$ sudo systemctl stop wildfly.service
administrador@UBUNTUDAW:/tmp$ sudo systemctl start wildfly.service
administrador@UBUNTUDAW:/tmp$ sudo systemctl enable wildfly.service
Created symlink /etc/systemd/system/multi-user.target.wants/wildfly.service → /etc/systemd/system/wildfly.service.
```

Acto seguido comprobamos mediante status que esta correcto:

```
administrador@UBUNTUDAW:/tmp$ sudo systemctl status wildfly.service
● wildfly.service - The WildFly Application Server
   Loaded: loaded (/etc/systemd/system/wildfly.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-03-09 19:00:14 CET; 20s ago
     Main PID: 24296 (launch.sh)
        Tasks: 121 (limit: 9430)
      Memory: 267.7M
         CPU: 6.507s
       CGroup: /system.slice/wildfly.service
               └─24296 /bin/bash /opt/wildfly/bin/launch.sh standalone standalone.xml 0.0.0.0
                 └─24297 /bin/sh /opt/wildfly/bin/standalone.sh -c standalone.xml -b 0.0.0.0
                   └─24387 java "-D[Standalone]" -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.pref

Mar 09 19:00:14 UBUNTUDAW systemd[1]: Started The WildFly Application Server.
Mar 09 19:00:26 UBUNTUDAW systemd[1]: /etc/systemd/system/wildfly.service:11: PIDFile= references a path below legacy directory /va
lines 1-14/14 (END)
```

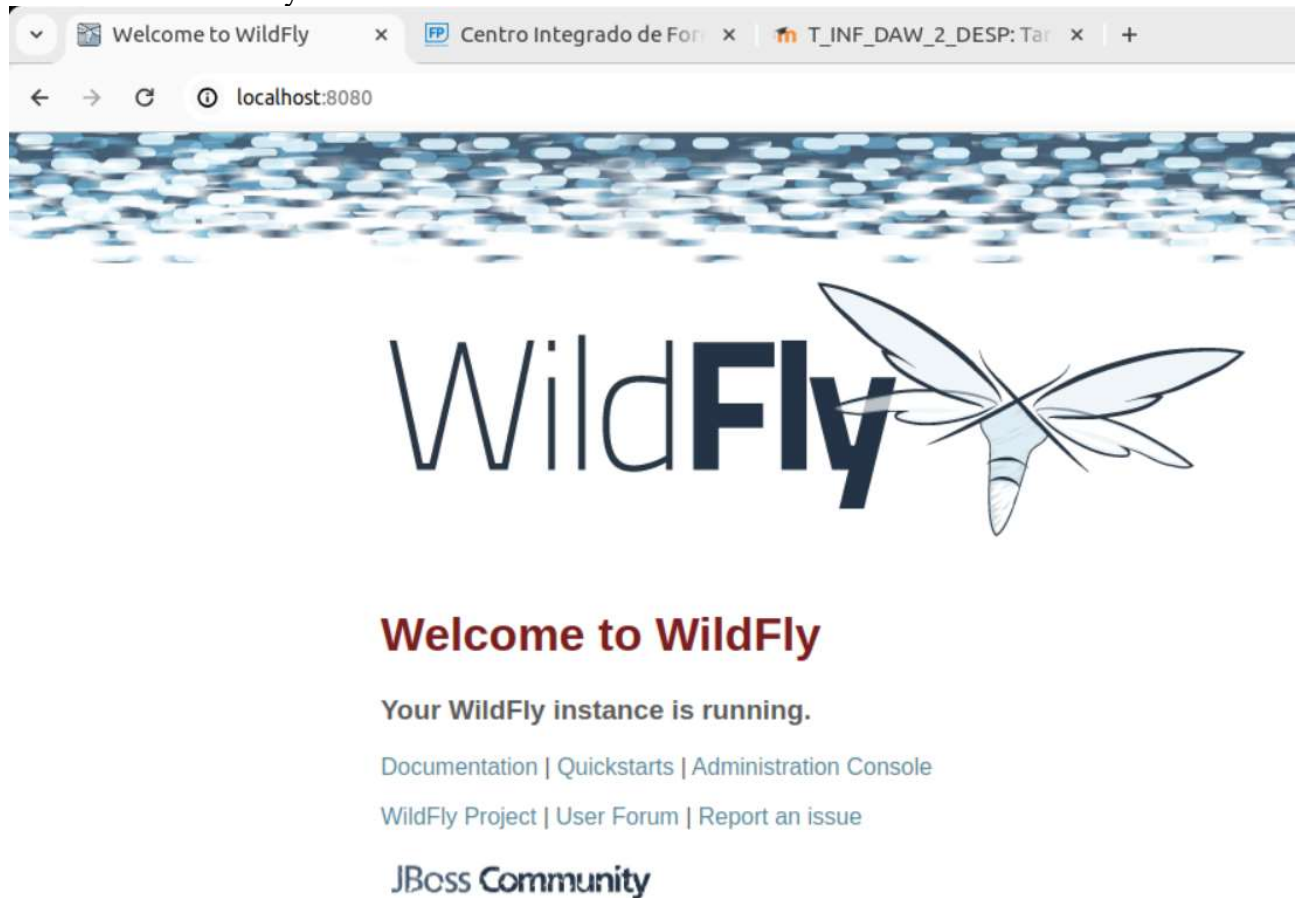
Creamos una cuenta administrador y contraseña: User:"miguel" password:"abc123.."

```
administrador@UBUNTU:~$ sudo /opt/wildfly/bin/add-user.sh
[sudo] contraseña para administrador:

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 : miguel
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 :
```

Abrimos en localhost:8080 y tras introducir las credenciales del usuario veremos la pantalla de bienvenida de WildFly:



Le damos a Administration Console y a “deployments” (despliegues)

Tras darle a desplegar nos saldrá para seleccionar un archivo, seleccionamos el sample.war:

Add Deployment

Upload Deployment

Specify Names

1

2

Help

Name *

sample.war

Runtime Name

sample.war

Enabled

ON

Required fields are marked with *

Y si todo va bien veremos la siguiente pantalla:

Upload Deployment

Specify Names

1

2

Upload successful

sample.war has been successfully uploaded to the content repository.

View Deployment

Volvemos a “Deployments” y vemos que está desplegado y activo.

The screenshot shows the WildFly console interface. The browser address bar indicates the URL: `localhost:9990/console/index.html#deployments;path=deployment~dply-samplewar`. The navigation bar includes links for Homepage, Deployments, Configuration, Runtime, Patching, and Access Control. The 'Deployments' tab is active, showing a list of deployments with 'sample.war' selected. A green status bar indicates: 'The deployment **sample.war** is enabled and active. [Disable](#)'. Below this, the 'Main Attributes' section provides details for 'sample.war':

Attribute	Value
Name:	sample.war
Runtime Name:	sample.war
Context Root:	/sample
Enabled, Managed, Exploded:	✓ ✓ ✗
Status:	OK
Last enabled at:	3/9/24, 7:45 PM
Last disabled at:	n/a

Vamos a explorar alguna funcionalidad más del WildFly, si vamos a Runtime, nos dará un listado de servidores y en Monitor podemos ver el Status, que nos dará información sobre el actual:

The screenshot shows the WildFly console with the 'Runtime' tab selected. The left sidebar shows a list of servers, with 'ubuntudaw' selected. The 'Monitor' section for 'ubuntudaw' is expanded, showing various management operations like Batch, JBeret, Datasources, EJB, IO, JAX-RS, and INDI. The main area displays the 'Status' of the server, including system information and resource usage:

Status [Refresh](#)

Linux 6.2.0-39-generic, 2 Processors
OpenJDK 64-Bit Server VM 11
Uptime: 1 hour, 26 minutes,

Heap

Category	Used	Total
Used	60 of 512 MB	512 MB
Committed	106 of 512 MB	512 MB

Threads

Category	Count	Total
Daemon	16 of 48 Threads	48 Threads

También Podríamos acceder al control de acceso, donde veríamos los usuarios, grupos y roles asignados a los usuarios pudiendo modificarlos o asignar a usuarios a estos:

WildFly

HomepageDeploymentsConfigurationRuntimePatchingAccess Control

Browse By	Role (7)	Membership
Users >	Filter by: role name, role type or scope	Filter by: Principal or Includes/excludes
Groups >	Administrator >	No Items
Roles >	Auditor >	
	Deployer >	
	Maintainer >	
	Monitor >	
	Operator Edit	
	SuperUser >	

Operator

Excludes
No user or groups excluded.

Includes
No user or groups included.

Description

Si cambiamos aquí el operator, podemos hacer que automáticamente asigne a todos los usuarios este Rol:

Edit Role X

Help

Name

Operator

Include All

ON

CloseSave

WildFly

HomepageDeploymentsConfigurationRuntimePatchingAccess Control

Browse By	Role (7)	Membership
Users >	Filter by: role name, role type or scope	Filter by: Principal or Includes/excludes
Groups >	Administrator >	No Items
Roles >	Auditor >	
	Deployer >	
	Maintainer >	
	Monitor >	
	Operator includes all Edit	
	SuperUser >	

Operator

Includes All
All authenticated users are automatically assigned to this role.

Description