



# BASH

Administración de sistemas con BASH

Fundamentos teóricos

Práctica evaluable

## *Practica Evaluable Bash*

**Alejandro Almagro Torregrosa**

## ***INDICE***

### **1. Introduccion a la propuesta**

### **2. Preparacion del entorno**

1. Maquina Servidor
2. Actualziacion y Dependencias
3. Descarga Repositorio

### **3. Instalacion de la app**

- 3.1. Comprobacion de dependencias
- 3.2. Compilacion del codigo
- 3.3. Instalación de la aplicación

### **4. Instalacion de la base de datos**

- 4.1. Importar la base de datos
- 4.2. Creacion de usuario admin en la base de datos
- 4.3 Actualizacion de datos

### **5. Puesta en marcha del servidor**

- 5.1. Ejecucion de los comandos necesarios.
- 5.2. Creacion de la cuenta Administrador
- 5.3 Prueba de conexión desde el anfitrión al servidor generado en la maquina virtual.

### **6. Problemas encontrados**

### **7.Soluciones aplicadas como pasos extras**

- 7.1. Implementacion de tmux y su script
- 7.2. Creacion de Alias en el sistema
- 7.3. Crontab
- 7.4. Comprobaciones de conexión con los cambios.

## 1. Introduccion

Tenemos que hacer un deploy automatizado de una aplicación, entonces la idea es hacer un script en bash que no sirva para poder meterla en cualquier maquina ubuntu sea servidor o no.

He decidido instalar mediante un script que nos automatize la instalacion y puesta a punto de un servidor de world of warcraft en la version 3.3.5a que es la mas utilizada en internet y con mas facilidad de acceso para el publico.

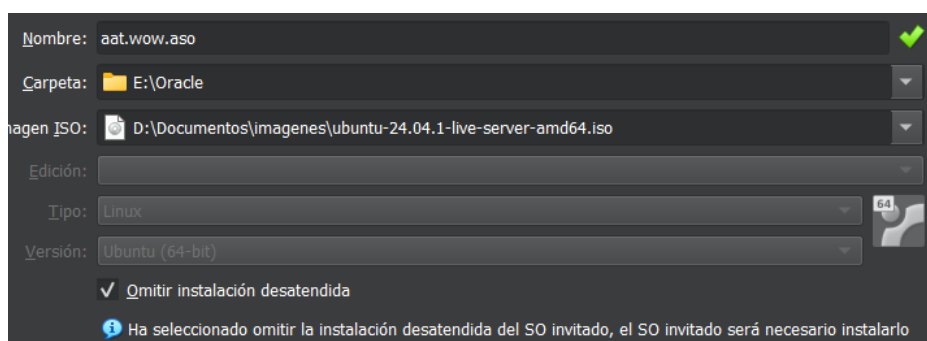
Dicho esto, necesitamos analizar los requisitos para poder instalar y automatizar este servidor, mantemiento y supervision.

Vamos a ir creando poco a poco el script para asi explicarlo, poco a poco.

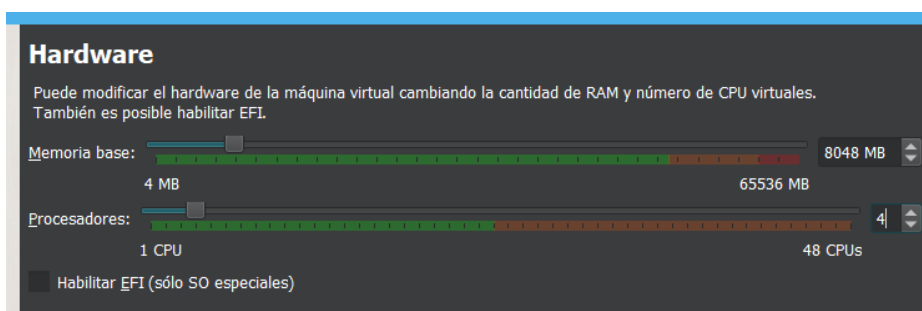
## 2. Preparacion del entorno

### 2.1. Prepacion de la maquina virtual:

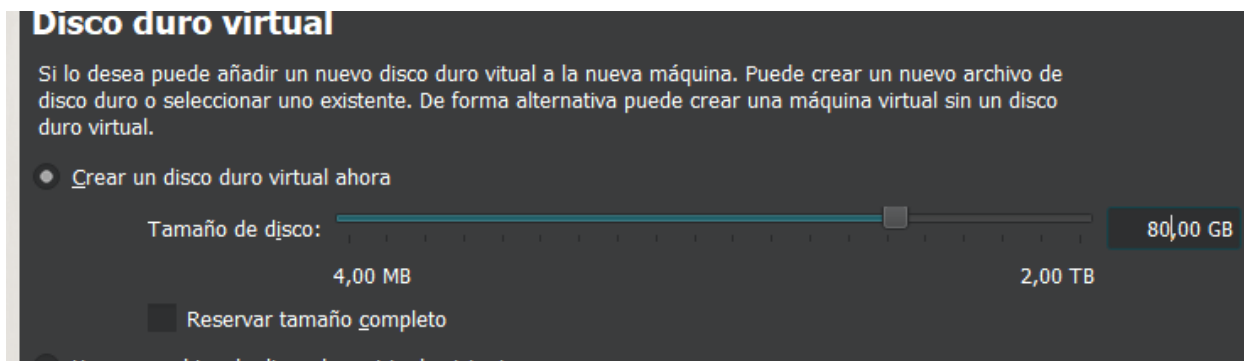
Utilizaremos una imagen de ubuntu server sin interfaz grafica la cual configuraremos en adaptador puente (**red**), y las siguientes capturas:



Recursos:



Disco:



## 2.2. Actualizacion y Dependencias

Crearemos las siguientes variables, para que se impriman cuando acaben para ver que todo marcha bien:

```
actualizacion="Equipo Actualizado"
```

```
apt update && apt upgrade -y
```

Despues de actualizar el servidor, pasaremos a ver que dependencias necesitamos para instalar esta aplicación en nuestro servidor ubuntu, para esto echaremos un vistazo a la wiki

Ademas de necesitar git y algunas librerias extras:

```
git cmake make gcc g++ clang libmysqlclient-dev libssl-dev libbz2-dev libreadline-dev libncurses-dev mysql-server libboost-all-dev -y
```

```
apt update && apt upgrade -y | apt install git cmake make gcc g++ clang libmysqlclient-dev libssl-dev libbz2-dev libreadline-dev libncurses-dev mysql-server libboost-all-dev -y  
echo "$actualizacion"
```

MySQL

Boost  $\geq 1.74$

OpenSSL  $\geq 3.0.x$

CMake  $\geq 3.16$

OS

GCC / CLang

## 2.3. Directorio y Repositorio

Crearemos las siguientes variables:

```
directorio="Dir Creado"  
git="Git Clonado"
```

Crearemos el directorio necesario:

```
mkdir aatwow;  
cd aatwow;  
  
echo "$directorio"
```

En este clonaremos el repositorio master utilizando git :

```
git clone https://github.com/azerothcore/azerothcore-wotlk.git; cd azerothcore-wotlk;  
  
echo "$git"
```

## 3. Instalacion de la app

Crearemos las siguientes variables:

```
compilado="Compilacion Finalizada"  
cliente="Descarga del cliente finalizada"  
aprobacion="Dependencias chekeadas"
```

### 3.1. Verificacion de dependencias

Añadiremos al script que comprueba automaticamente las dependencias y si necesitan actualizarse:

```
./acore.sh install-deps;  
  
echo "$aprobacion"
```

### 3.2. Compilacion del codigo :

Ahora añadiremos la parte que compila todo el codigo

```
./acore.sh compiler all;  
echo "$compilado"
```

### 3.3 Descarga del client-data:

Añadimos la ultima parte de esta seccion que se encargara

```
./acore.sh client-data;  
echo "$cliente"
```

## 4. Instalacion de la base de datos

Creamos mas variables, en un script separado llamado sql.sh:

```
# Configura las variables necesarias  
MYSQL_ROOT_USER="root"  
NEW_USER="acore"  
NEW_USER_PASS="skoda123" # Define la contraseña que quieres asignar al usuario acore  
# Ejecuta los comandos SQL en MySQL
```

### 4.1. Creacion de usuario admin en la base de datos

Ahora crearemos otro script que ejecute estas ordenes en la base de datos para poder permitir la conexión remota del usuario que administre el servidor.

```
# Ejecuta los comandos SQL en MySQL  
mysql -u "$MYSQL_ROOT_USER" <<EOF  
-- Elimina el usuario "acore" en cualquier host, si existe.  
DROP USER IF EXISTS '$NEW_USER'@'%';  
DROP USER IF EXISTS '$NEW_USER'@'localhost';  
  
-- Crea el usuario "acore" permitiendo conexiones desde cualquier host (%).  
CREATE USER '$NEW_USER'@'%' IDENTIFIED BY '$NEW_USER_PASS';  
  
-- Otorga todos los privilegios sobre todas las bases de datos y la opción de otorgar privilegios a otros.  
GRANT ALL PRIVILEGES ON *.* TO '$NEW_USER'@'%' WITH GRANT OPTION;  
  
-- Aplica los cambios  
FLUSH PRIVILEGES;  
EOF  
  
echo "El usuario MySQL '$NEW_USER' ha sido recreado exitosamente."
```

El script funciona.

```
El usuario MySQL 'acore' ha sido recreado exitosamente.  
root@aatwow:/home/aat#
```

## 4.2. Importar la base de datos

En el script principal, una vez añadido el anterior, ordenaremos la descarga de la base de datos, que se basa en los dos siguientes comandos, para que importen automáticamente la base de datos del repositorio, esta base de datos se clona, porque no se puede crear de 0 una bdd tan compleja, hay que tener en cuenta que es para un videojuego al fin y al cabo.

**Resultado del 1 comando, :**

```
Opening DatabasePool 'acore_auth'. Asynchronous connections: 1, synchronous connections: 1.
Could not connect to MySQL database at 127.0.0.1: Unknown database 'acore_auth'
Database "acore_auth" does not exist
Do you want to create it? [yes (default) / no]:
yes
```

**Resultado del 2 comando, :**

```
Opening DatabasePool 'acore_characters'. Asynchronous connections: 1, synchronous connections: 1.
Could not connect to MySQL database at 127.0.0.1: Unknown database 'acore_characters'
Database "acore_characters" does not exist
Do you want to create it? [yes (default) / no]:
yes
```

Una vez dado a yes, el se encargara de clonar toda la base de datos a nuestro mysql-server:

Se puede comprobar las bases de datos que ha importado con el siguiente comando:

```
mysql> show databases;
+-----+
| Database |
+-----+
| acore_auth |
| acore_characters |
| acore_world |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
7 rows in set (0,00 sec)

mysql>
```

```
root@aatwow: /home/aat/aatwow/azerothcore-wotlk

Database Auth is empty, auto populating it...
>> Applying 'logs_ip_actions.sql'...
>> Applying 'realmcharacters.sql'...
>> Applying 'secret_digest.sql'...
>> Applying 'updates_include.sql'...
>> Applying 'motd.sql'...
>> Applying 'uptime.sql'...
>> Applying 'motd_localized.sql'...
>> Applying 'autobroadcast.sql'...
>> Applying 'logs.sql'...
>> Applying 'account_banned.sql'...
>> Applying 'account_access.sql'...
>> Applying 'build_info.sql'...
>> Applying 'ip_banned.sql'...
>> Applying 'account.sql'...
>> Applying 'updates.sql'...
>> Applying 'realmmlist.sql'...
>> Applying 'account_muted.sql'...
>> Done!

Updating Auth database...
>> Applying update "2025_01_26_00.sql" 'CA6884F'...
>> Applied 1 query. Containing 3 new and 1 archived updates.

Started auth database connection pool.
Loading IP Location Database...

Added realm "AzerothCore" at 127.0.0.1:8085.
```

### 4.3 Actualizacion de datos para poder conectarnos

Ahora por ultimo actualizaremos los datos de conexión, que seran el nombre que queramos dare al servidor que hemos creado y la ip necesaria para que la gente puedan conectarse, esto lo haremos mediante otro script, que añadiremos al principal:

```
root@aatwow: /home/aat
GNU nano 7.2 realm.sh *
#!/bin/bash
# actualizar_realmlist.sh
# Este script actualiza la dirección y el nombre del realmlist en la base de datos acore_auth.

# Configura las variables necesarias
MYSQL_ROOT_USER="root"
MYSQL_DB="acore_auth"
NUEVA_DIRECCION="192.168.1.159"
NUEVO_NOMBRE="Asoservidorwow"

# Ejecuta los comandos SQL para actualizar el realmlist
mysql -u "$MYSQL_ROOT_USER" -p "$MYSQL_DB" <<EOF
UPDATE realmlist SET address = '$NUEVA_DIRECCION' WHERE id = 1;
UPDATE realmlist SET name = '$NUEVO_NOMBRE' WHERE id = 1;
exit;
EOF

echo "El realmlist ha sido actualizado: dirección '$NUEVA_DIRECCION' y nombre '$NUEVO_NOMBRE'."
```

Comprobamos que funciona:

```
root@aatwow:/home/aat# source realm.sh
Enter password:
El realmlist ha sido actualizado: dirección '192.168.1.159' y nombre 'Asoservidorwow'.
root@aatwow:/home/aat#
```

Se añadirían al script principal, como tienen echo propio, solo los incluimos:

```
./sql.sh

./realm.sh
```

Por ultimo tendremos que copiar las configuraciones del servidor

```
cp env/dist/etc/authserver.conf.dist env/dist/etc/authserver.conf
cp env/dist/etc/worldserver.conf.dist env/dist/etc/worldserver.conf

echo "$archivos"
```



## 5. Puesta en marcha del servidor

Ahora por ultimo para poner en marcha nuestro servidor tendremos que ejecutar los siguientes pasos

### 5.1. Ejecución de los comandos necesarios.

Por ultimo tendremos que copiar las configuraciones del servidor

Con este paso cumplido, tendremos los binarios del servidor en este path `azerothcore-wotlk/env/dist/bin`

Ahora nos damos cuenta de que necesitamos correr almenos dos conexiones ssh si tenemos el servidor en remoto, o como en este caso una ssh y otra desde la propia maquina virtual.

```
root@aatwow:/home/aat/aatwow/azerothcore-wotlk# ./acore.sh run-worldserver
NOTICE: file </home/aat/aatwow/azerothcore-wotlk/conf/config.sh> not found, we use default configuration only.
Running worldserver ...
Running without GDB
AzerothCore rev. 17412174be22 2025-02-01 22:49:41 +0100 (master branch) (Unix, Release, Static) (worldserver-daemon)
<Ctrl-C> to stop.
```



Y este comando tambien:

```
root@aatwow:/home/aat/aatwow/azerothcore-wotlk
root@aatwow:/home/aat/aatwow/azerothcore-wotlk# ./acore.sh run-authserver
```

```
root@aatwow:/home/aat/aatwow/azerothcore-wotlk# ./acore.sh run-authserver
NOTICE: file </home/aat/aatwow/azerothcore-wotlk/conf/config.sh> not found, we use default configuration only.
Running authserver ...
Running without GDB
AzerothCore rev. 17412174be22 2025-02-01 22:49:41 +0100 (master branch) (Unix, Release, Static) (authserver)
<Ctrl-C> to stop.

  A Z E R O T H
    C O R E

AzerothCore 3.3.5a - www.azerothcore.org

> Using configuration file      /home/aat/aatwow/azerothcore-wotlk/env/dist/etc/authserver.conf
> Using SSL version:           OpenSSL 3.0.13 30 Jan 2024 (library: OpenSSL 3.0.13 30 Jan 2024)
> Using Boost version:         1.83.0

Opening DatabasePool 'acore_auth'. Asynchronous connections: 1, synchronous connections: 1.
MySQL client library: 8.4.4
MySQL server ver: 8.4.4
Connected to MySQL database at 127.0.0.1
MySQL client library: 8.4.4
MySQL server ver: 8.4.4
Connected to MySQL database at 127.0.0.1
DatabasePool 'acore_auth' opened successfully. 2 total connections running.

Updating Auth database...
>> Auth database is up-to-date! Containing 4 new and 1 archived updates.

Started auth database connection pool.
Loading IP Location Database...

Added realm "Asoservidorwow" at 192.168.1.159:8085.
```

La razon de esto es que necesitamos poner en marcha el modulo auth-core y el propio mundo del servidor.

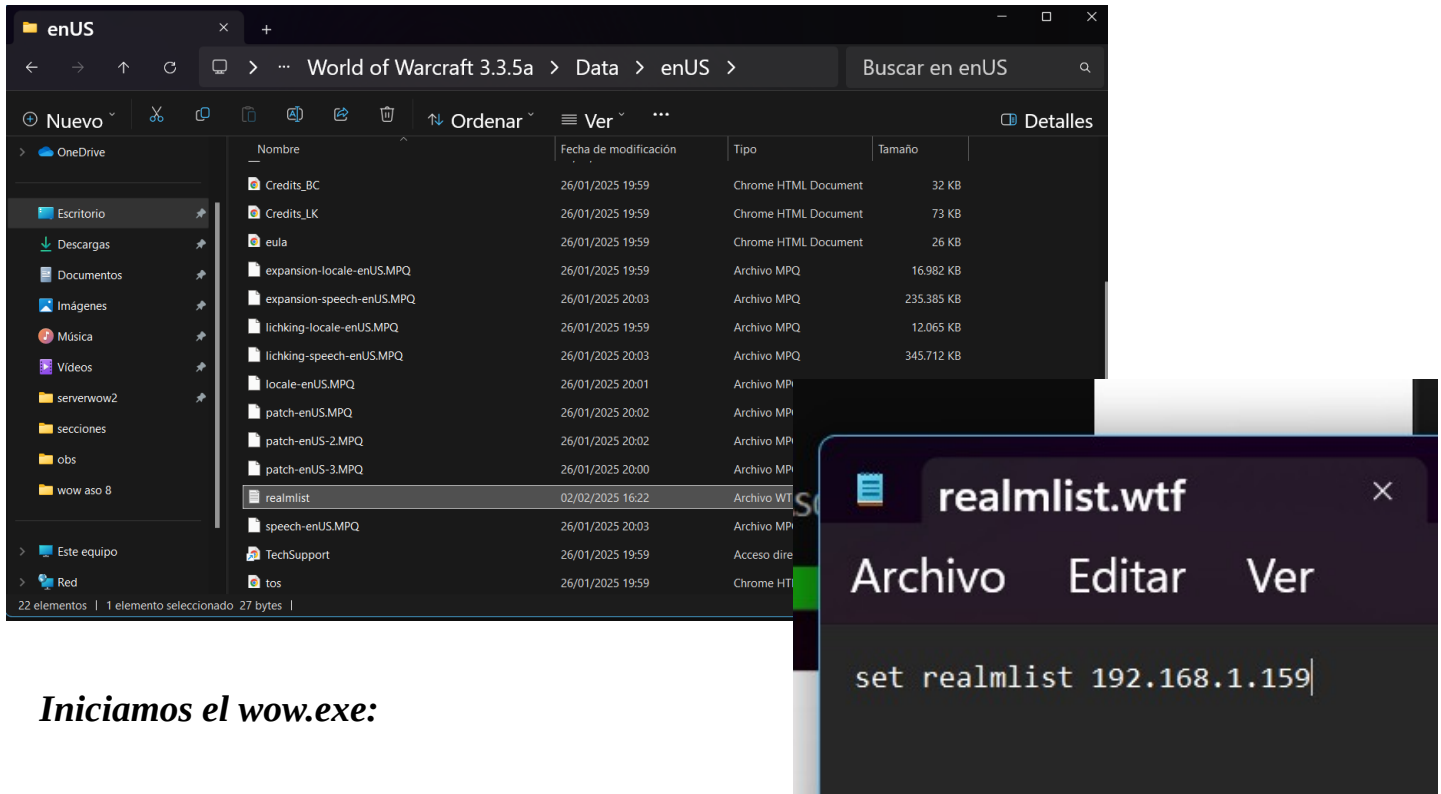
## 5.2. Creación de la cuenta Administrador

Una vez ejecutados los comandos anteriores crearemos la cuenta de administrador dentro de la cli que ha abierto **la session iniciada en worldserver**, en vivo que controla el mundo del servidor, para poder conectarnos desde el cliente.

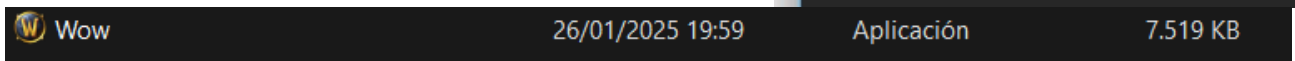
```
AC> Calendar deletion of old events.
AC> account create aataso 1234
AC> Account created: aataso
AC> account set gmlevel aataso 3 -1
AC> You change security level of account AATASO to 3.
AC>
[world-ses0:bash*
```

### 5.3 Prueba de conexión desde el anfitrión al servidor generado en la maquina virtual.

Ahora con un cliente de world of warcraft 3.3.5a previamente descargado en nuestro ordenador solo necesitaremos cambiar el realmlist por la ip de nuestra maquina virtual.



*Iniciamos el wow.exe:*



*Nos logeamos con la cuenta que hemos creado:*



*Veremos que se ha creado el realm como lo hemos llamado y la cuenta funciona perfectamente:*



## 6. Problemas encontrados

El principal problema que encontramos es que no es nada práctico tener que crear dos instancias de ssh para poder iniciar los servicios que requerimos, por tanto vamos a optar por un repositorio llamado **tmux**.

**Tmux** es una herramienta creada para gestionar varias sesiones de cli a la vez y permitir una automatización mediante scripts.

Otra idea de ampliación es la creación de alias en el sistema que evoquen una acción al ponerlos en la consola, como si fueran comandos, gracias a esto obtendremos una gestión del servidor muy eficiente.

```
root@aatwow:/home/aat# tmux -V
tmux 3.4
root@aatwow:/home/aat#
```

## 7. Soluciones aplicadas como pasos extra.

### 7.1. Creacion del script necesario para poder correr varias ventanas a la vez

Necesitamos si o si automatizar la puesta en marcha del servidor para hacerlo de manera facil y simple, para ello correremos un script vinculado a tmux, llamado start.sh:

```
root@aatwow:/home/aat# cd
root@aatwow:~# nano /root/start.sh
```

Definiremos como variable los dos comandos que utilizamos para poner en marcha el servidor asi como su ubicación:

```
GNU nano 7.2 /root/start.sh *
#!/bin/bash
authserver="/home/aat/aatwow/azerothcore-wotlk/acore.sh run-authserver"
worldserver="/home/aat/aatwow/azerothcore-wotlk/acore.sh run-worldserver"

authserver_session="auth-session"
worldserver_session="world-session"

if tmux new-session -d -s $authserver_session; then
    echo "Created authserver session: $authserver_session"
else
    echo "Error when trying to create authserver session: $authserver_session"
fi

if tmux new-session -d -s $worldserver_session; then
    echo "Created worldserver session: $worldserver_session"
else
    echo "Error when trying to create worldserver session: $worldserver_session"
fi

if tmux send-keys -t $authserver_session "$authserver" C-m; then
    echo "Executed \"$authserver\" inside $authserver_session"
    echo "You can attach to $authserver_session and check the result using \"tmux attach -t $authserver_session\""
else
    echo "Error when executing \"$authserver\" inside $authserver_session"
fi
```

```
else
    echo "Error when executing \"$authserver\" inside $authserver_session"
fi

if tmux send-keys -t $worldserver_session "$worldserver" C-m; then
    echo "Executed \"$worldserver\" inside $worldserver_session"
    echo "You can attach to $worldserver_session and check the result using \"tmux attach -t $worldserver_session\""
else
    echo "Error when executing \"$worldserver\" inside $worldserver_session"
fi

#Final del script
```

```
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify
^C Location  ^_ Go To Line ^M-U Undo     ^M-A Set Mark
^M-E Redo    ^M-G Copy
```



## 7.2. Creacion de alias en el sistema para que interactuen con este script en fondo

Para eso editaremos con nano el siguiente directorio:

```
root@aatwow:/home/aat# nano ~/.bashrc
```

Pondremos los siguientes alias abajo:

```
#START
alias wow='cd ~/azerothcore-wotlk;tmux attach -t world-session'
alias auth='cd ~/azerothcore-wotlk;tmux attach -t auth-session'
alias start='bash /root/start.sh'
alias stop='tmux kill-server'
alias compile='cd ~/azerothcore-wotlk;./acore.sh compiler all'
alias build='cd ~/azerothcore-wotlk;./acore.sh compiler build'
alias update='cd ~/azerothcore-wotlk;git pull;cd ~/azerothcore-wotlk/modules/mod-playerbots;git pull'
alias pb='nano ~/azerothcore-wotlk/env/dist/etc/modules/playerbots.conf'
alias world='nano ~/azerothcore-wotlk/env/dist/etc/worldserver.conf'
alias updatemods="cd ~/azerothcore-wotlk/modules;find . -mindepth 1 -maxdepth 1 -type d -print -exec git -C {} pull \;"
#END
```

Guardaremos y haremos un reboot.

## 7.3. Añadir el script al crontab para la ejecucion cuando se ponga en marcha.

Utilizaremos crontab para automatizar la puesta en marcha del server haciendo que el script creado en la seccion 7.1 se ejecute cada vez que se inicie la maquina:

```
root@aatwow:~# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 1
```

Añadiremos la siguiente linea:

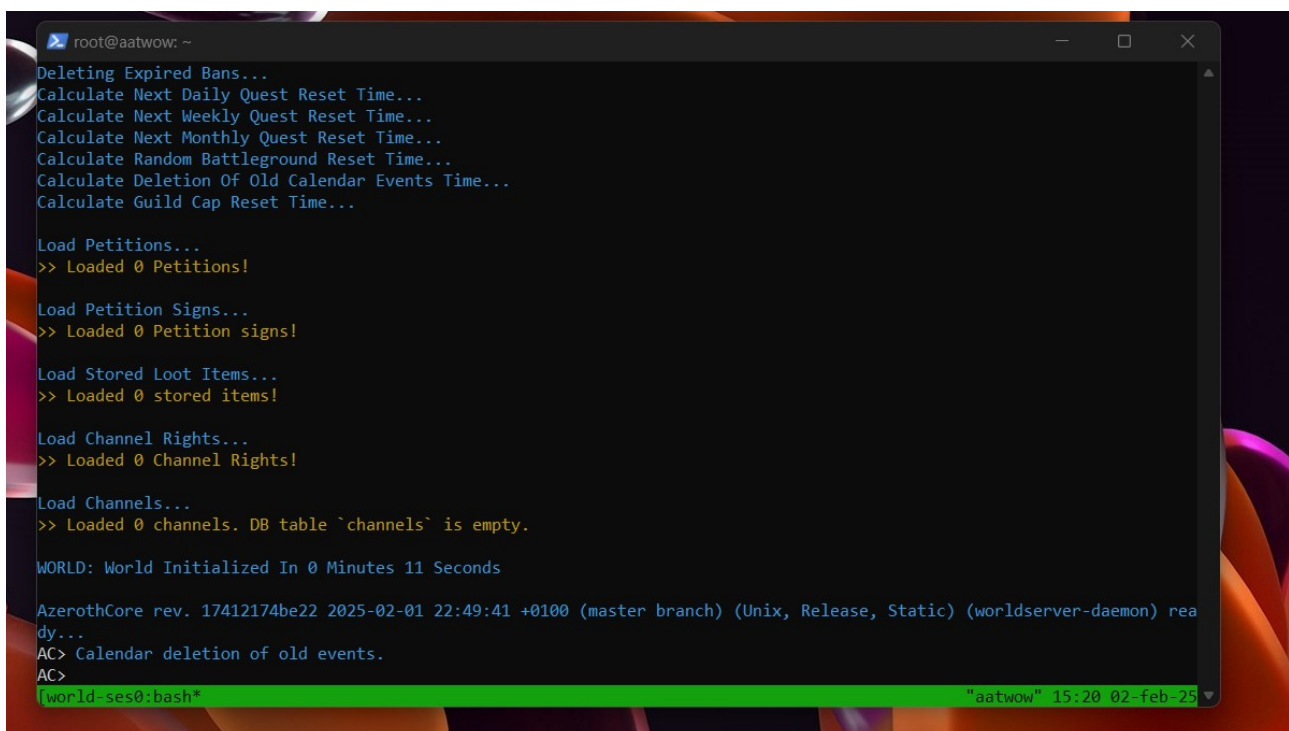
```
#
# m h dom mon dow  command
@reboot /bin/bash ~/root/start.sh
```

## 7.4. Comprobaciones finales echos los cambios:

Si ponemos start pasara lo siguiente:

```
root@aatwow:~# start
Created authserver session: auth-session
Created worldserver session: world-session
Executed "/home/aat/aatwow/azerothcore-wotlk/acore.sh run-authserver" inside auth-session
You can attach to auth-session and check the result using "tmux attach -t auth-session"
Executed "/home/aat/aatwow/azerothcore-wotlk/acore.sh run-worldserver" inside world-session
You can attach to world-session and check the result using "tmux attach -t world-session"
root@aatwow:~# wow
```

Al poner wow se pondra la siguiente session de tmux en primer plano:



```
root@aatwow: ~
Deleting Expired Bans...
Calculate Next Daily Quest Reset Time...
Calculate Next Weekly Quest Reset Time...
Calculate Next Monthly Quest Reset Time...
Calculate Random Battleground Reset Time...
Calculate Deletion Of Old Calendar Events Time...
Calculate Guild Cap Reset Time...

Load Petitions...
>> Loaded 0 Petitions!

Load Petition Signs...
>> Loaded 0 Petition signs!

Load Stored Loot Items...
>> Loaded 0 stored items!

Load Channel Rights...
>> Loaded 0 Channel Rights!

Load Channels...
>> Loaded 0 channels. DB table `channels` is empty.

WORLD: World Initialized In 0 Minutes 11 Seconds

AzerothCore rev. 17412174be22 2025-02-01 22:49:41 +0100 (master branch) (Unix, Release, Static) (worldserver-daemon) ready...
AC> Calendar deletion of old events.
AC>
[world-ses0:~#] "aatwow" 15:20 02-feb-25
```

Se puede cambiar de sesssion y volver, deberemos pulsar **ctrl+B** y luego **D** de “detach” de nuestra ventana de tmux.

*Creamos un personaje y entramos al mundo:*



*Vemos que funciona bien:*





### *Informacion que se muestra en vivo en la sesion de tmux de world:*

```
AC> You change security level of account AKA50 to 3.
AC> Update time diff: 114ms with 1 players online
Last 500 diffs summary:
|- Mean: 2ms
|- Median: 2ms
|- Percentiles (95, 99, max): 6ms, 10ms, 114ms

AC>
[world-ses0:~]$
```

### *Comprobaciones del script:*

```
Equipo Actualizado
Dir Creado
Cloning into 'azerothcore-wotlk'...
remote: Enumerating objects: 173759, done.
remote: Counting objects: 100% (15711/15711), done.
remote: Compressing objects: 100% (676/676), done.
remote: Total 173759 (delta 15257), reused 15035 (delta 15035), pack-reused 158048 (from 1)
Receiving objects: 100% (173759/173759), 1016.27 MiB | 12.52 MiB/s, done.
Resolving deltas: 100% (123655/123655), done.
Updating files: 100% (8487/8487), done.
Git Clonado
```

```
Platform: linux-gnu
-- CMake version: 3.28.3
-- The CXX compiler identification is Clang 18.1.3
-- The C compiler identification is Clang 18.1.3
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/clang++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/clang - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Running cmake hook: AFTER_LOAD_CONF
-- No hooks registered for AFTER_LOAD_CONF
-- Enabled C++20 standard
-- Detected 64-bit platform
-- UNIX: Using jemalloc
-- UNIX: Using default configuration directory
-- UNIX: Using default library directory
-- UNIX: Configuring uninstall target
-- UNIX: Created uninstall target
-- UNIX: Detected compiler: /usr/bin/clang
-- Clang: Minimum version required is 10.0.0, found 18.1.3 - ok!
-- Performing Test CLANG_HAVE_PROPER_CHARCONV
-- Performing Test CLANG_HAVE_PROPER_CHARCONV - Success
-- Clang: All warnings enabled
-- Running cmake hook: AFTER_LOAD_CMAKE_MODULES
```

```

Setting permissions on binary files
Done
Running hooks: HOOKS_MAP_ON_AFTER_BUILD
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/conf.sh.dist' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/conf.sh.dist'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/examples' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/examples'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/examples/restarter-auth.sh' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/examples/restarter-auth.sh'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/examples/restarter-world.sh' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/examples/restarter-world.s
h'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/examples/starter-auth.sh' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/examples/starter-auth.sh'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/examples/starter-world.sh' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/examples/starter-world.sh'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/gdb.conf' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/gdb.conf'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/run-engine' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/run-engine'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/simple-restarter' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/simple-restarter'
'/home/aat/aatwow/azerothcore-wotlk/apps/startup-scripts/starter' -> '/home/aat/aatwow/azerothcore-wotlk/env/dist/starter'
Compilacion Finalizada
NOTICE: file </home/aat/aatwow/azerothcore-wotlk/conf/config.sh> not found, we use default configuration only.
#####
Client data downloader
#####
Downloading client data in: /home/aat/aatwow/azerothcore-wotlk/env/dist/bin/data.zip ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:--  --:--:--  --:--:--     0
100 1230M 100 1230M    0     0 33.4M    0  0:00:36  0:00:36  --:--:-- 34.0M
unzip downloaded file in /home/aat/aatwow/azerothcore-wotlk/env/dist/bin...
Remove downloaded file
Descarga del cliente finalizada
Configuraciones copiadas
Creado el usuario acore en la base de datos
root@aatwow:/home/aat/aatwow/azerothcore-wotlk#

```



***Una vez acabe ya podremos proceder a los pasos mencionados a partir del 5.2***