

# Auto GLPI Install

just run this script

```
#!/bin/bash
#
# GLPI install script
#
#Author Masaru_M
#Ver 1.0.R

function warn(){
    echo -e '\e[31m'$1'\e[0m';
}
function info(){
    echo -e '\e[36m'$1'\e[0m';
}

function check_root()
{
# Vérification des privilèges root
if [[ "$(id -u)" -ne 0 ]]
then
    warn "This script must be run as root" >&2
    exit 1
else
    info "Root privilege: OK"
fi
}

function check_distro()
{
# Constante pour les versions de Debian acceptables
DEBIAN_VERSIONS=("11" "12")

# Constante pour les versions d'Ubuntu acceptables
UBUNTU_VERSIONS=("22.04")
```

```
# Récupération du nom de la distribution
DISTRO=$(lsb_release -is)

# Récupération de la version de la distribution
VERSION=$(lsb_release -rs)

# Vérifie si c'est une distribution Debian
if [ "$DISTRO" == "Debian" ]; then
    # Vérifie si la version de Debian est acceptable
    if [[ "${DEBIAN_VERSIONS[*]}" == *"$VERSION" ]]; then
        info "Your operating system version ($DISTRO $VERSION) is compatible."
    else
        warn "Your operating system version ($DISTRO $VERSION) is not noted as
compatible."
        warn "Do you still want to force the installation? Be careful, if you choose
to force the script, it is at your own risk."
        info "Are you sure you want to continue? [yes/no]"
        read response
        if [ $response == "yes" ]; then
            info "Continuing..."
        elif [ $response == "no" ]; then
            info "Exiting..."
            exit 1
        else
            warn "Invalid response. Exiting..."
            exit 1
        fi
    fi
fi

# Vérifie si c'est une distribution Ubuntu
elif [ "$DISTRO" == "Ubuntu" ]; then
    # Vérifie si la version d'Ubuntu est acceptable
    if [[ "${UBUNTU_VERSIONS[*]}" == *"$VERSION" ]]; then
        info "Your operating system version ($DISTRO $VERSION) is compatible."
    else
        warn "Your operating system version ($DISTRO $VERSION) is not noted as
compatible."
        warn "Do you still want to force the installation? Be careful, if you choose
to force the script, it is at your own risk."
```

```

        info "Are you sure you want to continue? [yes/no]"
        read response
        if [ $response == "yes" ]; then
            info "Continuing..."
        elif [ $response == "no" ]; then
            info "Exiting..."
            exit 1
        else
            warn "Invalid response. Exiting..."
            exit 1
        fi
    fi

# Si c'est une autre distribution
else
    warn "Il s'agit d'une autre distribution que Debian ou Ubuntu qui n'est pas
compatible."
    exit 1
fi
}

function network_info()
{
INTERFACE=$(ip route | awk 'NR==1 {print $5}')
IPADDRESS=$(ip addr show $INTERFACE | grep inet | awk '{ print $2; }' | sed 's/\./.*$//' | head
-n 1)
HOST=$(hostname)
}

function confirm_installation()
{
warn "This script will now install the necessary packages for installing and configuring
GLPI."
info "Are you sure you want to continue? [yes/no]"
read confirm
if [ $confirm == "yes" ]; then
    info "Continuing..."
elif [ $confirm == "no" ]; then
    info "Exiting..."
    exit 1
else

```

```
        warn "Invalid response. Exiting..."
        exit 1
    fi
}

function install_packages()
{
    info "Installing packages..."
    sleep 1
    apt update
    apt install --yes --no-install-recommends \
    apache2 \
    mariadb-server \
    perl \
    curl \
    jq \
    php
    info "Installing php extensions..."
    apt install --yes --no-install-recommends \
    php-ldap \
    php-imap \
    php-apcu \
    php-xmllrpc \
    php-cas \
    php-mysqli \
    php-mbstring \
    php-curl \
    php-gd \
    php-simplexml \
    php-xml \
    php-intl \
    php-zip \
    php-bz2
    systemctl enable mariadb
    systemctl enable apache2
}

function mariadb_configure()
{
    info "Configuring MariaDB..."
}
```

```

sleep 1
SLQR00TPWD=$(openssl rand -base64 48 | cut -c1-12 )
SQLGLPIPWD=$(openssl rand -base64 48 | cut -c1-12 )
systemctl start mariadb
sleep 1

# Remove anonymous user accounts
mysql -e "DELETE FROM mysql.user WHERE User = ''"
# Disable remote root login
mysql -e "DELETE FROM mysql.user WHERE User = 'root' AND Host NOT IN ('localhost',
'127.0.0.1', '::1')"
# Remove the test database
mysql -e "DROP DATABASE test"
# Reload privileges
mysql -e "FLUSH PRIVILEGES"
# Create a new database
mysql -e "CREATE DATABASE glpi"
# Create a new user
mysql -e "CREATE USER 'glpi_user'@'localhost' IDENTIFIED BY '$SQLGLPIPWD'"
# Grant privileges to the new user for the new database
mysql -e "GRANT ALL PRIVILEGES ON glpi.* TO 'glpi_user'@'localhost'"
# Reload privileges
mysql -e "FLUSH PRIVILEGES"

# Initialize time zones datas
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql mysql
#Ask tz
dpkg-reconfigure tzdata
systemctl restart mariadb
sleep 1
mysql -e "GRANT SELECT ON mysql.time_zone_name TO 'glpi_user'@'localhost'"
mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED BY '$SLQR00TPWD'"
}

function install_glpi()
{
info "Downloading and installing the latest version of GLPI..."
# Get download link for the latest release
DOWNLOADLINK=$(curl -s https://api.github.com/repos/glpi-project/glpi/releases/latest | jq -r
'.assets[0].browser_download_url')

```

```
wget -O /tmp/glpi-latest.tgz $DOWNLOADLINK
tar xzf /tmp/glpi-latest.tgz -C /var/www/html/
touch /var/www/html/glpi/files/_log/php-errors.log

# Add permissions
chown -R www-data:www-data /var/www/html/glpi
chmod -R 775 /var/www/html/glpi
# Setup vhost
cat > /etc/apache2/sites-available/000-default.conf << EOF
<VirtualHost *:80>
    DocumentRoot /var/www/html/glpi/public
    <Directory /var/www/html/glpi/public>
        Require all granted
        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>

    LogLevel warn
    ErrorLog \${APACHE_LOG_DIR}/error-glpi.log
    CustomLog \${APACHE_LOG_DIR}/access-glpi.log combined

</VirtualHost>
EOF

#Disable Apache Web Server Signature
echo "ServerSignature Off" >> /etc/apache2/apache2.conf
echo "ServerTokens Prod" >> /etc/apache2/apache2.conf

# Setup Cron task
echo "*/2 * * * * www-data /usr/bin/php /var/www/html/glpi/front/cron.php &>/dev/null" >>
/etc/cron.d/glpi

#Activation du module rewrite d'apache
a2enmod rewrite && systemctl restart apache2
}

function setup_db()
{
info "Setting up GLPI..."
```

```

cd /var/www/html/glpi
php bin/console db:install --db-name=glpi --db-user=glpi_user --db-password=$SQLGLPIPWD --no-
interaction
rm -rf /var/www/html/glpi/install
}

function display_credentials()
{
info "=====> GLPI installation details <====="
warn "It is important to record this informations. If you lose them, they will be
unrecoverable."
info "=> GLPI:"
info "Default user accounts are:"
info "USER      -  PASSWORD      -  ACCESS"
info "glpi      -  glpi          -  admin account,"
info "tech      -  tech          -  technical account,"
info "normal    -  normal        -  normal account,"
info "post-only -  postonly      -  post-only account."
echo ""
info "You can connect access GLPI web page from IP or hostname:"
info "http://$IPADRESS or http://$HOST"
echo ""
info "=> Database:"
info "root password:          $SLQROOTPWD"
info "glpi_user password:     $SQLGLPIPWD"
info "GLPI database name:      glpi"
info "<=====>"
echo ""
info "If you encounter any issue with this script, please report it to \"Masaru_M\" on discord."

}

check_root
check_distro
confirm_installation
network_info
install_packages
mariadb_configure
install_glpi

```

setup\_db

display\_credentials

---

Created 2024-11-26 20:55:17 UTC by Nicolas

Updated 2024-12-06 02:01:33 UTC by Nicolas