

CENTOS 7 PHP 71 Laravel 5.7

connect to SQL Server

Requirements

- 2.5 GB RAM
- CENTOS 7.6
- Laravel 5.7
- PHP 7.1

Create PHP apps using SQL Server on RHEL

Set up your Environment

step 1.1 Install SQL Server

```
curl https://packages.microsoft.com/config/rhel/7/mssql-server-2017.repo | sudo tee
/etc/yum.repos.d/mssql-server-2017.repo
```

```
sudo yum update -y
```

```
sudo yum install mssql-server -y
```

```
sudo /opt/mssql/bin/mssql-conf setup
```

```
[root@tecmaq armando]# sudo /opt/mssql/bin/mssql-conf setup
Choose an edition of SQL Server:
 1) Evaluation (free, no production use rights, 180-day limit)
 2) Developer (free, no production use rights)
 3) Express (free)
 4) Web (PAID)
 5) Standard (PAID)
 6) Enterprise (PAID)
 7) Enterprise Core (PAID)
 8) I bought a license through a retail sales channel and have a product key to enter.

Details about editions can be found at
https://go.microsoft.com/fwlink/?LinkId=852748&clcid=0x409

Use of PAID editions of this software requires separate licensing through a
Microsoft Volume Licensing program.
By choosing a PAID edition, you are verifying that you have the appropriate
number of licenses in place to install and run this software.

Enter your edition(1-8): 2
The license terms for this product can be found in
/usr/share/doc/mssql-server or downloaded from:
https://go.microsoft.com/fwlink/?LinkId=855862&clcid=0x409

The privacy statement can be viewed at:
https://go.microsoft.com/fwlink/?LinkId=853010&clcid=0x409

Do you accept the license terms? [Yes/No]:yes

Enter the SQL Server system administrator password:
Confirm the SQL Server system administrator password:
Configuring SQL Server...

ForceFlush is enabled for this instance.
ForceFlush feature is enabled for log durability.
Created symlink from /etc/systemd/system/multi-user.target.wants/mssql-server.service to /usr/lib/systemd/system/mssql-server.servi
ce.
Setup has completed successfully. SQL Server is now starting.
```

Step 1.2 Install PHP and other required packages

FOR RHEL

To install PHP 7.1 or 7.2, replace remi-php73 with remi-php71 or remi-php72 respectively in the following commands.

```
# sudo su
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm -y
$ sudo yum install -y http://rpms.remirepo.net/enterprise/remi-release-7.rpm
#rpm -Uvh remi-release-7.rpm epel-release-latest-7.noarch.rpm
subscription-manager repos --enable=rhel-7-server-optional-rpms
yum install yum-utils
yum-config-manager --enable remi-php71
yum update
yum install php php-pdo php-xml php-pear php-devel re2c gcc-c++ gcc Copy
Compiling the PHP drivers with PECL with PHP 7.2 requires a more recent GCC than the default:
```

```
sudo yum-config-manager --enable rhel-server-rhscsl-7-rpms
sudo yum install devtoolset-7
scl enable devtoolset-7 bash Copy
You have successfully installed PHP on your RHEL machine!
```

SELinux is installed by default and runs in Enforcing mode. To allow Apache to connect to a database through SELinux, run the following command:

```
sudo setsebool -P httpd_can_network_connect_db 1
```

FOR CENTOS

```
# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm -y
$ sudo yum install -y wget git epel-release
$ sudo yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm -y
$ sudo yum install yum-utils re2c gcc-c++ gcc -y
$ sudo yum-config-manager --enable remi-php72
$ sudo yum install php-tidy php-gd php-opcache php php-pgsql php-cli php-mbstring
php-process php-pdo php-mcrypt php-common php-bcmath php-xml php-mysql php-tcpdf
php-tcpdf php-php-gettext php-ldap php-init php-pear php-devel php72-php-pecl-zip.x86_64
php-doctrine-dbal.noarch php-gmp.x86_64 php-zip zip unzip -y
$ sudo yum install php72-php-tidy php72-php-gd php72-php-opcache php72-php
php72-php-pgsql php72-php-cli php72-php-mbstring php72-php-process php72-php-pdo
php72-php-mcrypt php72-php-common php72-php-bcmath php72-php-xml php72-php-mysql
php72-php-gettext php72-php-ldap php72-php-pear php-zipstream.noarch
php-pecl-zip.x86_64 php72-php-gmp -y
$ sudo yum install centos-release-scl
```

```
$ sudo yum install devtoolset-7
$ scl enable devtoolset-7 bash
```

Step 1.3 Install the ODBC Driver and SQL Command Line Utility for SQL Server

```
sudo su
curl https://packages.microsoft.com/config/rhel/7/prod.repo >
/etc/yum.repos.d/mssql-tools.repo
sudo ACCEPT_EULA=Y yum install msodbcsql17 mssql-tools
exit
sudo yum remove unixODBC-utf16 unixODBC-utf16-devel #to avoid conflicts
sudo ACCEPT_EULA=Y yum install msodbcsql17 mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
sudo yum install unixODBC-devel
sudo ACCEPT_EULA=Y yum install mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
sudo yum install unixODBC-devel
```

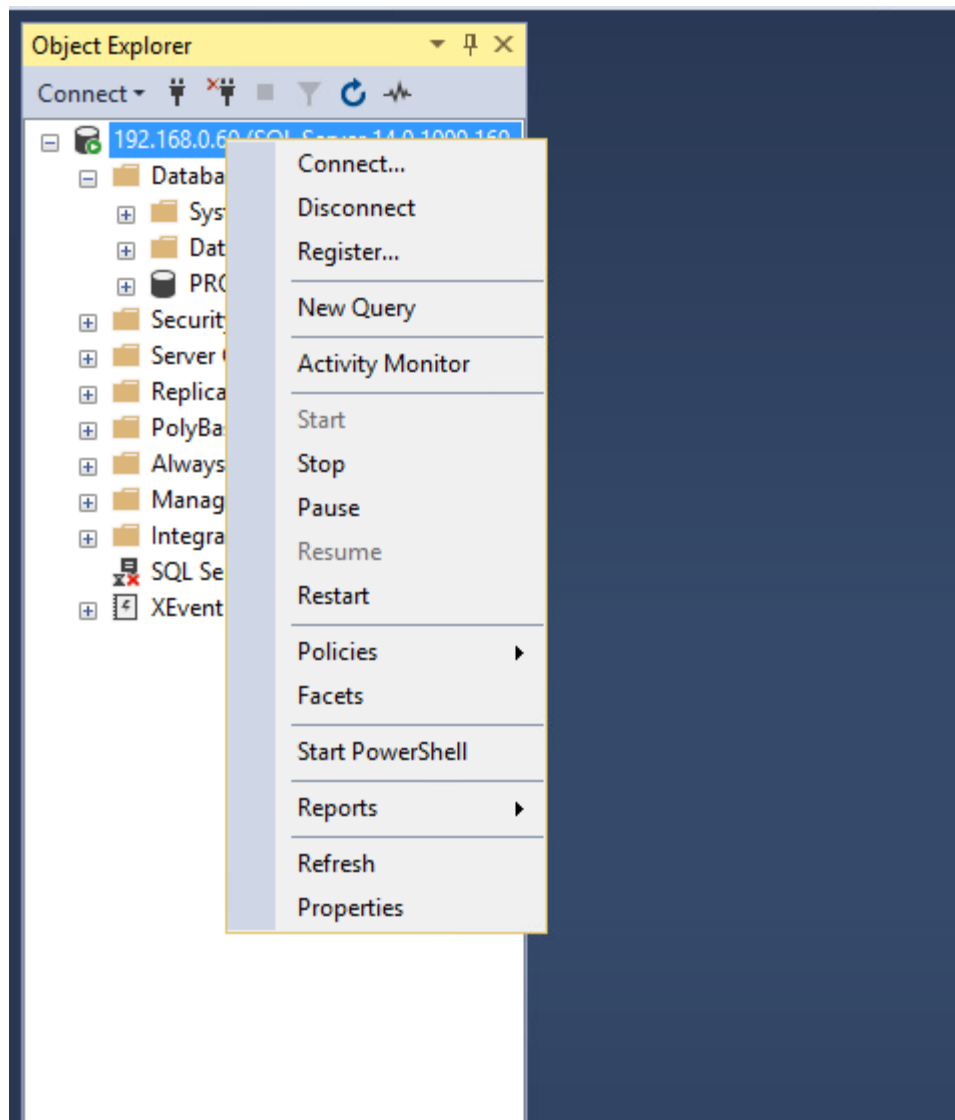
Create PHP application with SQL Server

Step 2.1 Install the PHP Driver for SQL Server

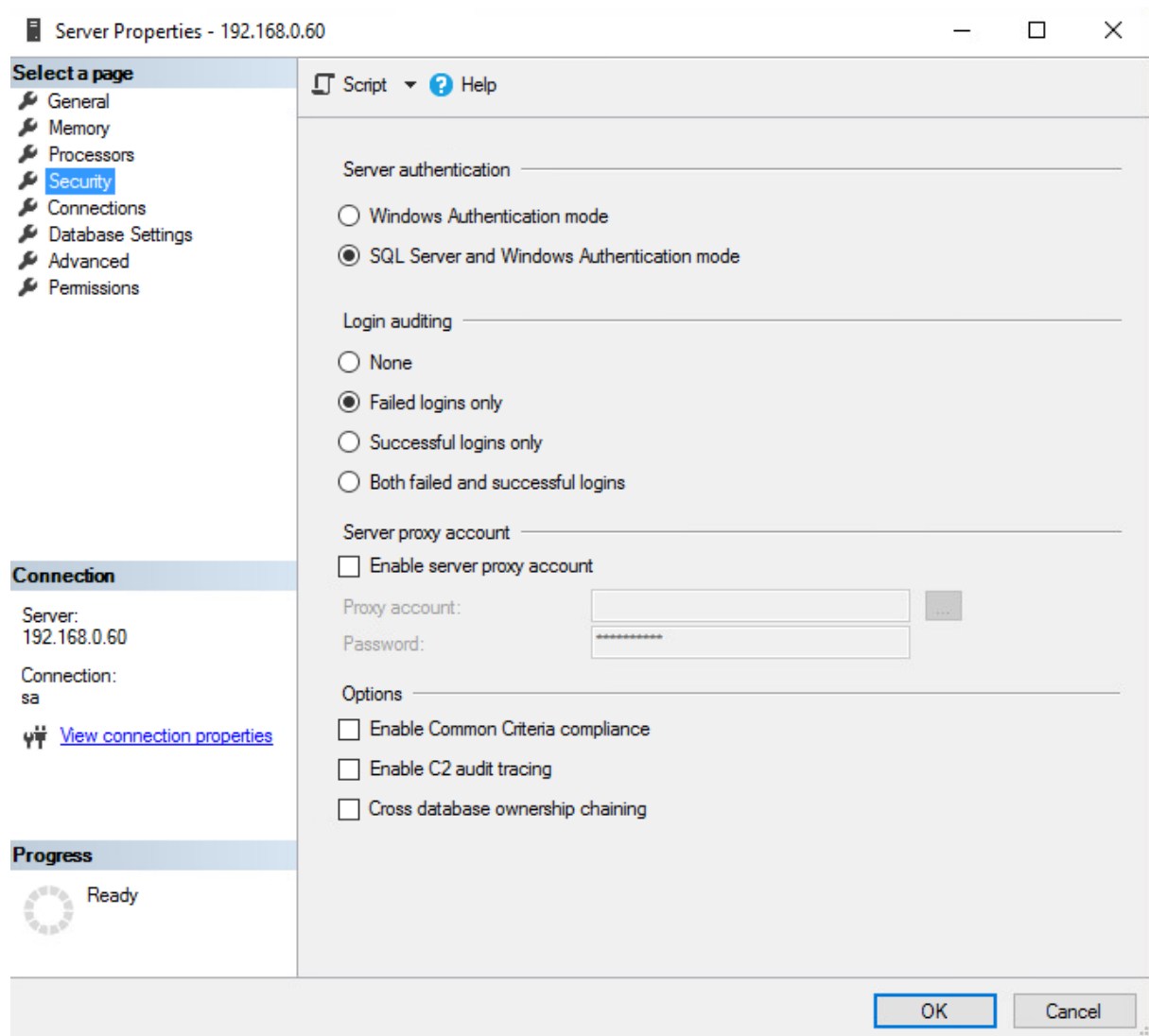
```
sudo pecl install sqlsrv
sudo pecl install pdo_sqlsrv
sudo su
echo extension=pdo_sqlsrv.so >> `php --ini | grep "Scan for additional .ini files" | sed -e
"s|.*/|"/30-pdo_sqlsrv.ini
echo extension=sqlsrv.so >> `php --ini | grep "Scan for additional .ini files" | sed -e
"s|.*/|"/20-sqlsrv.ini
exit
```

SQL Server configurations

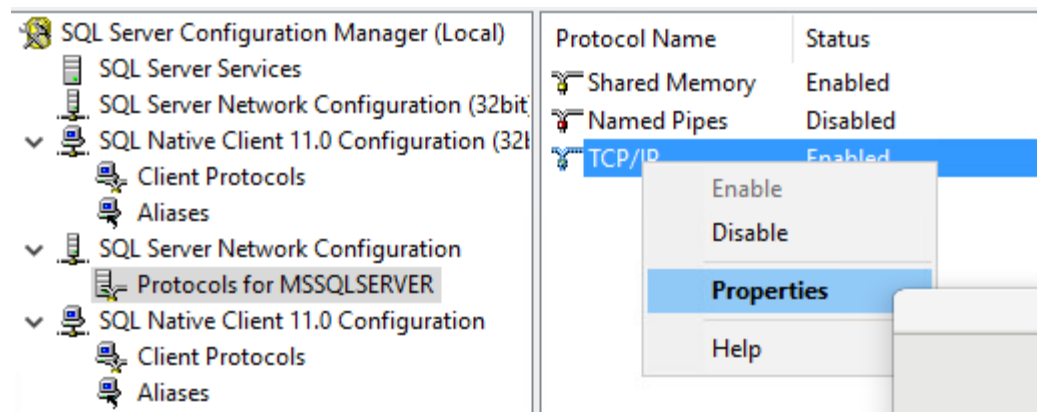
En propiedades del servicio de BD

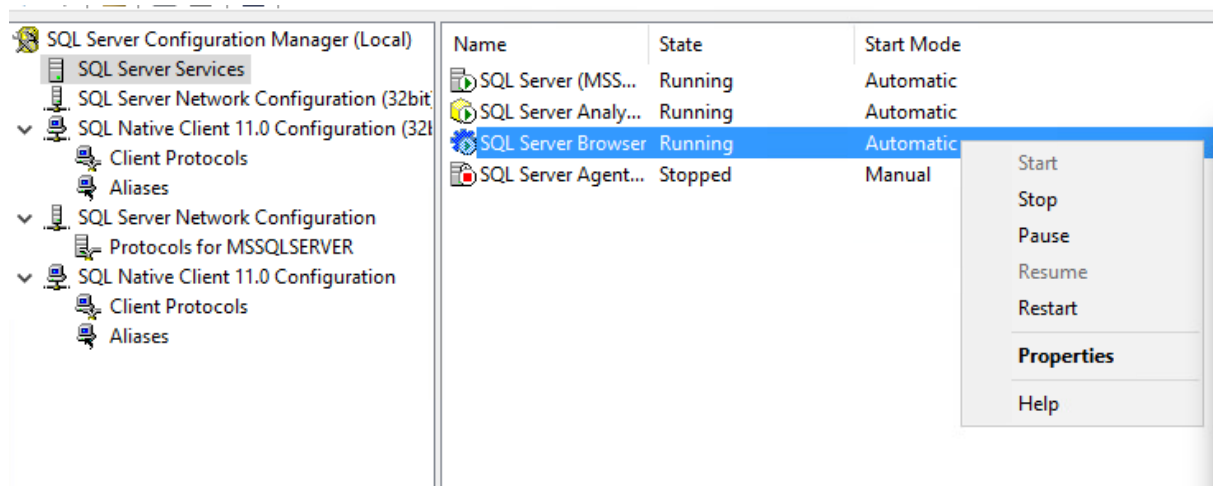
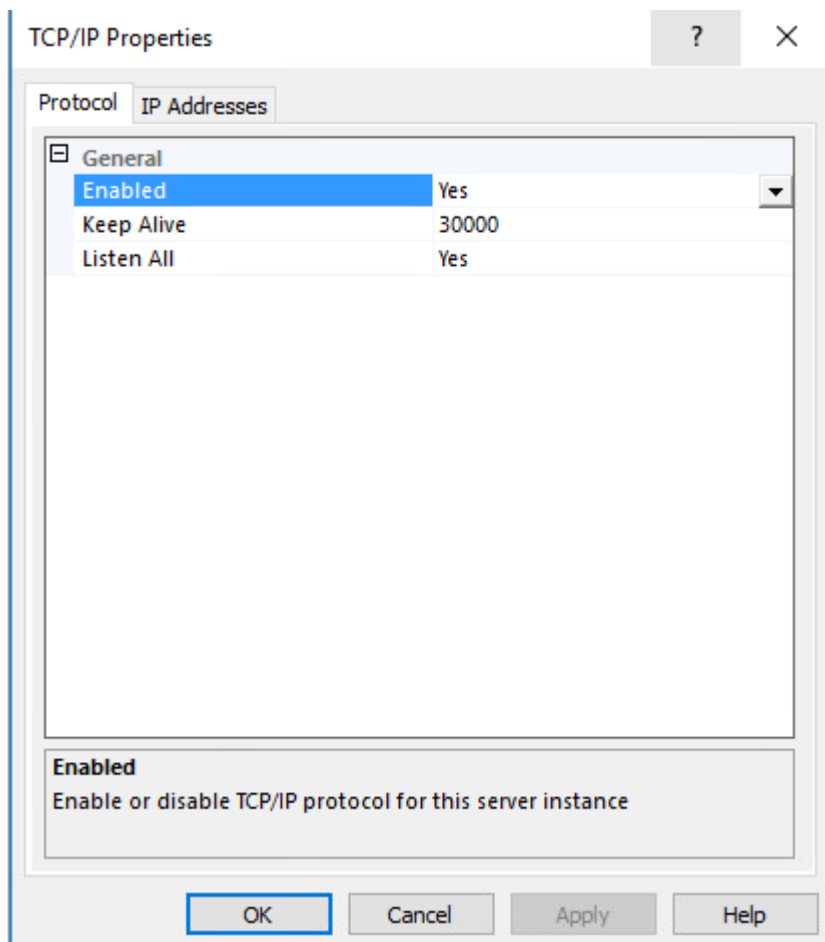


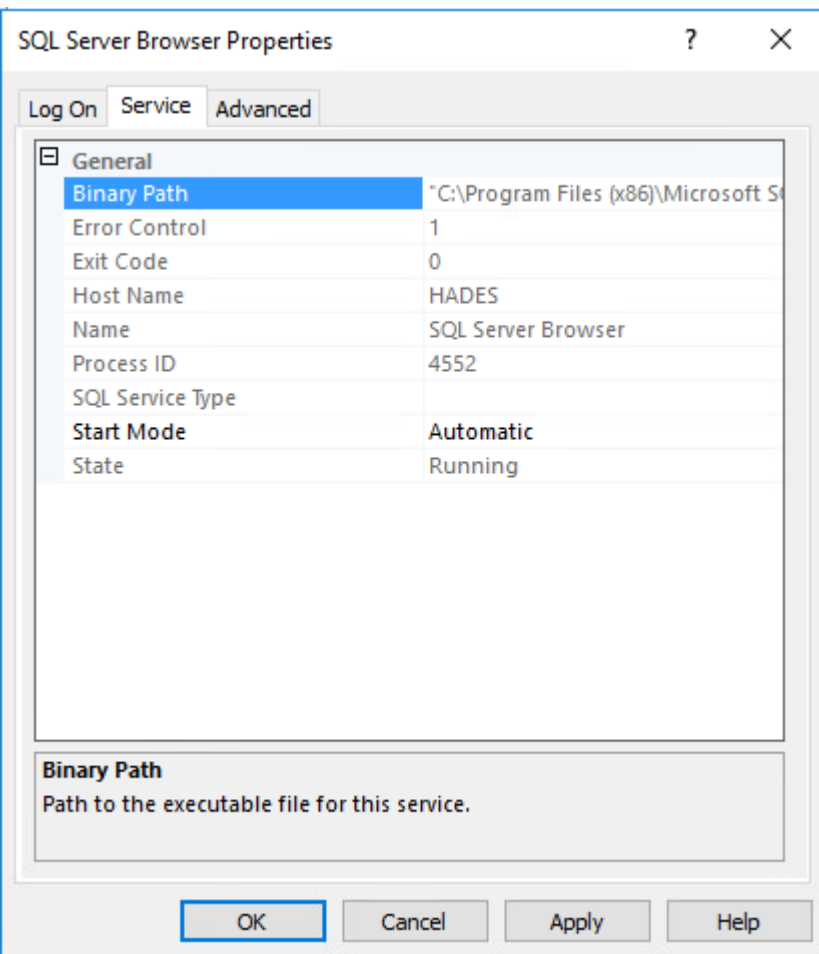
EN la sección de security tener habilitado la opción de SQL server

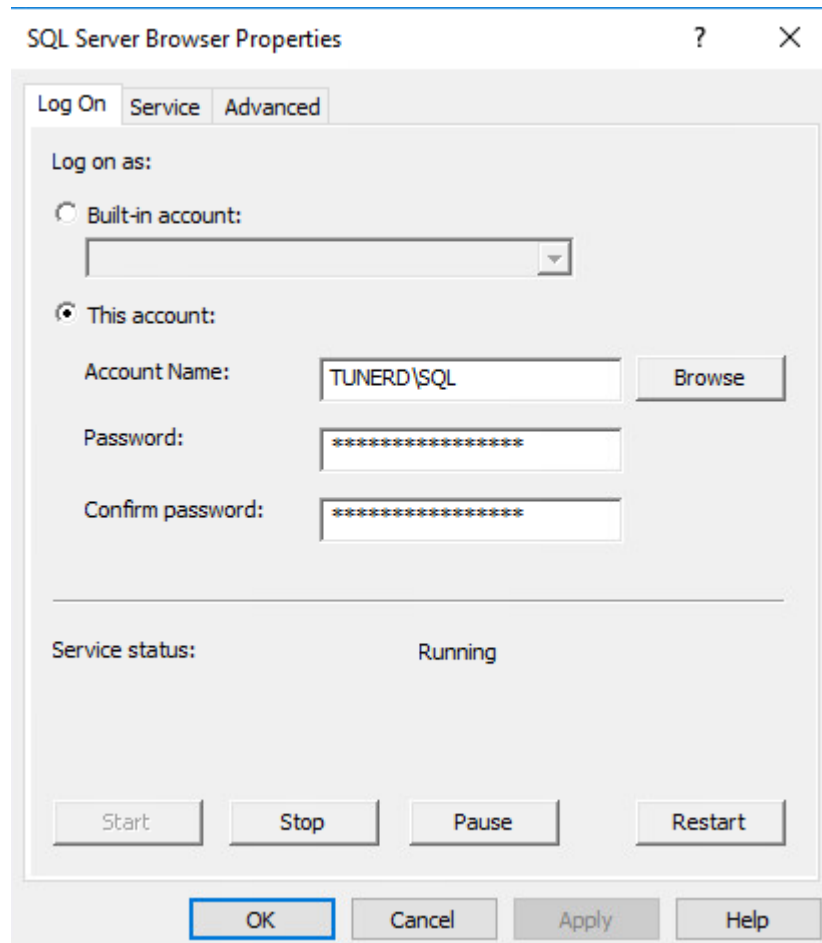


EM SQL SERVER CONFIGURATION









UNIR CENTOS A DOMINIO

```
$ yum install realmd oddjob samba samba-common-oddjob mkhomedir sssd ntpdate ntp  
realm join --user=Administrator@dominio dominio
```

samba

```
sudo setsebool -P samba_export_all_rw 1  
sudo semanage fcontext -a -t public_content_rw_t "/opt/sites(/.*)?"  
sudo restorecon -vFR /opt/sites  
sudo setsebool -P allow_smbd_anon_write 1  
sudo setsebool -P allow_httpd_anon_write 1
```


Laravel configuration

```
$ sudo yum install git httpd mod_ssl mod_evasive -y
# instalar composer, ver https://getcomposer.org/download/
#install nodejs
$ curl -sL https://rpm.nodesource.com/setup_14.x | sudo bash -
$ sudo yum install -y libpng-devel nodejs
$ sudo yum install redis postgresql-server postgresql-contrib.x86_64 -y
$ sudo systemctl start redis
$ sudo systemctl enable redis httpd postgresql
$ sudo mkdir /opt/sites
$ sudo chmod 3775 /opt/sites
$ sudo chown apache:apache /opt/sites
$ sudo semanage fcontext -a -t httpd_sys_content_t "/opt/sites(/.*)?"
$ sudo mkdir /etc/httpd/logs/tecmaq
$ sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/opt/sites/tecmaq.tunerd/storage(/.*)?"
$ sudo semanage fcontext -a -t httpd_sys_rw_content_t
"/opt/sites/tecmaq.tunerd/vendor(/.*)?"

$ sudo semanage fcontext -a -t httpd_log_t /etc/httpd/logs/tecmaq
$ sudo restorecon -vFR "/opt/sites"
$ cd /opt/sites/
#descargar el proyecto
$ cd <carpetaProyecto>
$ composer install
$ sudo npm i -g laravel-echo-server
$
```

```
[armando@vegeta tecmaq.tunerd]$ laravel-echo-server init
? Do you want to run this server in development mode? Yes
? Which port would you like to serve from? 6001
? Which database would you like to use to store presence channel members? redis
? Enter the host of your Laravel authentication server: https://tecmaq.tunerd.local
? Will you be serving on http or https? https
? Enter the path to your SSL cert file. /opt/sites/ca/intermediate/certs/tecmaq.tunerd.local.cert
.pem
? Enter the path to your SSL key file. /opt/sites/ca/intermediate/private/tecmaq.tunerd.local.key
.pem
? Do you want to generate a client ID/Key for HTTP API? No
? Do you want to setup cross domain access to the API? No
? What do you want this config to be saved as? laravel-echo-server.json
Configuration file saved. Run laravel-echo-server start to run server.
[armando@vegeta tecmaq.tunerd]$
```

archivo laravel.json

```
{
  "authHost": "https://tecmaq.tunerd.local",
  "authEndpoint": "/broadcasting/auth",
  "clients": [],
  "database": "redis",
  "databaseConfig": {
    "redis": {},
    "sqlite": {
      "databasePath": "/database/laravel-echo-server.sqlite"
    }
  },
  "devMode": true,
  "host": null,
  "port": "6001",
  "protocol": "https",
  "socketio": {},
  "sslCertPath": "/opt/sites/ca/intermediate/certs/tecmaq.tunerd.local.cert.pem",
  "sslKeyPath": "/opt/sites/ca/intermediate/private/tecmaq.tunerd.local.key.pem",
  "sslCertChainPath": "/opt/sites/ca/intermediate/certs/ca-chain.cert.pem",
  "sslPassphrase": "",
  "subscribers": {
    "http": true,
    "redis": true
  },
  "apiOriginAllow": {
    "allowCors": false,
    "allowOrigin": "",
    "allowMethods": "",
    "allowHeaders": ""
  }
}
```

```
$ sudo setsebool -P redis_enable_notify 1
$ sudo setsebool -P httpd_can_sendmail 1
$ sudo setsebool -P httpd_can_network_connect 1
$ sudo firewall-cmd --permanent --add-port=80/tcp
$ sudo firewall-cmd --permanent --add-port=443/tcp
$ sudo firewall-cmd --reload
```

Postgresql Configuration

```
sudo yum -y install
```

```
https://download.postgresql.org/pub/repos/yum/reporepms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

```
sudo yum install postgresql13-server postgresql13-contrib.x86_64 -y  
sudo postgresql-13-setup initdb
```

```
$ sudo yum install postgresql-server postgresql-contrib.x86_64 -y  
$ sudo systemctl enable postgresql  
$ sudo postgresql-setup initdb  
$ sudo systemctl start postgresql  
$ sudo firewall-cmd --add-service=postgresql --permanent  
$ sudo firewall-cmd --reload
```

SAMBA

```
$ sudo yum install samba samba-client samba-common  
editar el archivo /etc/samba/smb.conf agregar/editar las siguientes lineas
```

```
[global]
```

```
workgroup = WORKGROUP  
netbios name = TECMAQ
```

```
[sites]
```

```
path = /opt/sites  
valid users = @<group>  
browseable = yes  
writeable = yes  
guest ok = no  
read only = no
```

```
$ sudo systemctl enable smb nmb  
$ sudo systemctl start smb nmb
```

```
$ sudo firewall-cmd --permanent --zone=public --add-service=samba  
$ sudo firewall-cmd --reload  
$ sudo setsebool samba_enable_home_dirs 1  
$ sudo smbpasswd -a <user>  
$ sudo usermod -aG <group> <user>
```

CA TRUST on NUC and Server

```
$ sudo yum install ca-certificates
```

```
$ sudo mv <path_certCA> /etc/pki/ca-trust/source/anchors/  
$ cd /etc/pki/ca-trust/source/anchors/  
$ sudo chown root:root *  
$ sudo update-ca-trust extract
```

REFERRALS

<https://www.microsoft.com/en-us/sql-server/developer-get-started/php/rhel/step/2.html>

<https://www.tecmint.com/install-php-7-in-centos-7/>