

Microsoft PHP Drivers for SQL Server – Linux Install Instructions

Step 1: Install PHP (unless already installed)

PHP 7.0

Ubuntu 15.04

```
sudo su
sh -c 'echo "deb http://packages.dotdeb.org jessie all \ndeb-src
http://packages.dotdeb.org jessie all" >> /etc/apt/sources.list'
apt-get update
apt-get install php7.0 php7.0-fpm php-pear php7.0-dev mcrypt php7.0-mcrypt php-
mbstring php7.0-xml
```

Ubuntu 16.04

```
sudo su
apt-get update
apt-get -y install php7.0 mcrypt php7.0-mcrypt php-mbstring php-pear php7.0-dev
php7.0-xml
```

RedHat 7

```
sudo su
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
wget 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-config-manager --enable remi-php70
yum update
yum install php php-pdo php-xml php-pear php-devel
```

PHP 7.1

Ubuntu 16.04

```
sudo su
add-apt-repository ppa:ondrej/php
apt-get update
apt-get -y install php7.1 mcrypt php7.1-mcrypt php-mbstring php-pear php7.1-dev
```

RedHat 7

```
sudo su
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
wget 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-config-manager --enable remi-php71
yum update
yum install php php-pdo php-xml php-pear php-devel
```

Step 2: Install pre-requisites

Ubuntu 15.04

```
sudo su
sh -c 'echo "deb [arch=amd64] https://apt-mo.trafficmanager.net/repos/mssql-ubuntu-
vivid-release/ vivid main" > /etc/apt/sources.list.d/mssqlpreview.list'
sudo apt-key adv --keyserver apt-mo.trafficmanager.net --recv-keys 417A0893
apt-get update
apt-get install msodbcsql
#for silent install use ACCEPT_EULA=Y apt-get install msodbcsql
sudo apt-get install unixodbc-dev-utf16
```

Ubuntu 16.04

```
sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list >
/etc/apt/sources.list.d/mssql-release.list
exit
sudo apt-get update
sudo ACCEPT_EULA=Y apt-get install msodbcsql mssql-tools
sudo apt-get install unixodbc-dev-utf16
```

RedHat 7

```
sudo su
curl https://packages.microsoft.com/config/rhel/7/prod.repo > /etc/yum.repos.d/mssql-
release.repo
exit
sudo yum update
sudo yum remove unixODBC #to avoid conflicts
sudo ACCEPT_EULA=Y yum install msodbcsql mssql-tools
sudo yum install unixODBC-utf16-devel
```

*Note: You need to make sure you install PHP 7 before you proceed to step 2. The Microsoft PHP Drivers for SQL Server will only work for PHP 7+.

Step 3: Install Apache

PHP 7.0

Ubuntu

```
sudo apt-get install libapache2-mod-php7.0  
sudo apt-get install apache2
```

RedHat

```
sudo yum install httpd
```

PHP 7.1

Ubuntu

```
sudo apt-get install libapache2-mod-php7.1  
sudo apt-get install apache2
```

RedHat

```
sudo yum install httpd
```

Step 4: Install the Microsoft PHP Drivers for SQL Server

```
sudo pecl install sqlsrv  
sudo pecl install pdo_sqlsrv
```

Step 5: Load the Microsoft PHP Drivers for SQL Server

PHP 7.0

Ubuntu

```
echo "extension=/usr/lib/php/20151012/sqlsrv.so" >> /etc/php/7.0/apache2/php.ini
echo "extension=/usr/lib/php/20151012/pdo_sqlsrv.so" >> /etc/php/7.0/apache2/php.ini
echo "extension=/usr/lib/php/20151012/sqlsrv.so" >> /etc/php/7.0/cli/php.ini
echo "extension=/usr/lib/php/20151012/pdo_sqlsrv.so" >> /etc/php/7.0/cli/php.ini
```

RedHat

```
echo "extension= /usr/lib64/php/modules/sqlsrv.so" > /etc/php.d/sqlsrv.ini
echo "extension= /usr/lib64/php/modules/pdo_sqlsrv.so" > /etc/php.d/pdo_sqlsrv.ini
```

PHP 7.1

Ubuntu

```
echo "extension=/usr/lib/php/20160303/sqlsrv.so" >> /etc/php/7.1/apache2/php.ini
echo "extension=/usr/lib/php/20160303/pdo_sqlsrv.so" >> /etc/php/7.1/apache2/php.ini
echo "extension=/usr/lib/php/20160303/sqlsrv.so" >> /etc/php/7.1/cli/php.ini
echo "extension=/usr/lib/php/20160303/pdo_sqlsrv.so" >> /etc/php/7.1/cli/php.ini
```

RedHat

```
echo "extension= /usr/lib64/php/modules/sqlsrv.so" > /etc/php.d/sqlsrv.ini
echo "extension= /usr/lib64/php/modules/pdo_sqlsrv.so" > /etc/php.d/pdo_sqlsrv.ini
```

Step 6: Restart Apache to load the new php.ini file

Ubuntu

```
sudo service apache2 restart
```

RedHat

```
sudo apachectl restart
```

Step 7: Create your sample app

Navigate to `/var/www/html` and create a new file called `testsql.php`. Copy and paste the following code in `testsql.php` and change the server name, username, password and database name.

```
<?php
$serverName = "yourServername";
$connectionOptions = array(
    "Database" => "yourDatabase",
    "Uid" => "yourUsername",
    "PWD" => "yourPassword"
);
//Establishes the connection
$conn = sqlsrv_connect($serverName, $connectionOptions);
//Select Query
$sql= "SELECT @@Version as SQL_VERSION";
//Executes the query
$getResults= sqlsrv_query($conn, $sql);
//Error handling

if ($getResults == FALSE)
    die(FormatErrors(sqlsrv_errors()));
?>
<h1> Results : </h1>
<?php
while ($row = sqlsrv_fetch_array($getResults, SQLSRV_FETCH_ASSOC)) {
    echo ($row['SQL_VERSION']);
    echo ("<br/>");
}
sqlsrv_free_stmt($getResults);
function FormatErrors( $errors )
{
    /* Display errors. */
    echo "Error information: <br/>";

    foreach ( $errors as $error )
    {
        echo "SQLSTATE: ".$error['SQLSTATE']."<br/>";
        echo "Code: ".$error['code']."<br/>";
        echo "Message: ".$error['message']."<br/>";
    }
}
?>
```

Step 8: Run your sample app

Go to your browser and type in `http://localhost/testsql.php` You should be able to connect to your SQL Server/Azure SQL Database.