Magento Functional Setup after Installation

1. Installing Magento 2 sample data via GUI
2. Installing Magento 2 sample data via Command
3. Managing Magento 2 indexes and caches via command line
4. Managing Magento 2 Backup via command line
5. Managing Magento 2 set mode via command line
6. Transferring your magento 1 database to Magento 2 database
7. Configuring Redis for backend cache
   1. Go to /opt folder and download redis using below command
   2. **wget http://download.redis.io/releases/redis-3.0.5.tar.gz**
   3. extract the file using command tar xzf redis-3.0.5
   4. run the command make && make install
   5. move to utils director inside the redis-3.0.5 in opt directory
   6. run the command ./install-server.sh
   7. run the command to test redis server is working
   8. redis-cli -v && service redis\_6379 status && netstat -anp |grep redis
   9. now we will install PHP Module which communicate with redis server
   10. in opt directory run the below command to download php module
   11. **git clone https://github.com/phpredis/phpredis.git**
   12. go to phpredis directory
   13. run command phpize
   14. run command ./confiugre
   15. run command make && make install
   16. now we have to let php know where is redis extension available by creating a file
   17. go to /etc/php/7.2/mods-available and create a file with name redis.ini and add the below command
   18. extension=redis.so
   19. now we need to link the extension created above with PHP-FPM and PHP-CLI
   20. ln -s /etc/php/7.2/mods-available/redis.in /etc/php/7.2/fpm/conf.d/20-redis.ini
   21. ln -s /etc/php/7.2/mods-available/redis.in /etc/php/7.2/cli/conf.d/20-redis.ini
   22. now restart the php-fpm service using command service php7.2-fpm restart
   23. to make sure that redis PHP and redis server are running together,
   24. php -r “if (new Redis() == true) { echo \”\r\n OK \r\n”;}”
   25. If we get OK message then every think is OK
   26. Now open env.php file located /var/www/html/iyelectronics/app/etc

And paste the red colored code at the end of file.

<?php

// app/etc/env.php

**return** [

    // Other directives

    'cache' => [

        'frontend' => [

            'default' => [

                'backend' => 'Cm\_Cache\_Backend\_Redis',

                'backend\_options' => [

                    'server' => '127.0.0.1',

                    'database' => '0',

                    'port' => '6379',

                    'password' => ''

                ]

            ],

            'page\_cache' => [

                'backend' => 'Cm\_Cache\_Backend\_Redis',

                'backend\_options' => [

                    'server' => '127.0.0.1',

                    'database' => '1',

                    'port' => '6379',

                    'compress\_data' => '0',

                    'password' => ''

                ]

            ]

        ]

    ]

];

Search for session code and replace with below code

***'session'*** => [

'save' => ***'files'***

],

And replace it with snippet as follows:

***'session'*** => [

'save' => ***'redis'***,

***'redis'*** => [

'host' => ***'127.0.0.1'***,

'port' => '6379',

***'password'*** => ***''***,

***'timeout'*** => '2.5',

***'persistent\_identifier'*** => ***''***,

***'database'*** => '2',

***'compression\_threshold'*** => '2048',

***'compression\_library'*** => 'gzip',

***'log\_level'*** => '3',

***'max\_concurrency'*** => '6',

***'break\_after\_frontend'*** => '5',

***'break\_after\_adminhtml'*** => '30',

***'first\_lifetime'*** => '600',

***'bot\_first\_lifetime'*** => '60',

***'bot\_lifetime'*** => '7200',

***'disable\_locking'*** => '0',

***'min\_lifetime'*** => '60',

***'max\_lifetime'*** => ***'2592000'***

] ],

Now delete the cache and page-cache directory from var directory and run the blow command,

redis-cli ping if we get the PONG response, means it’s working

redis-cli monitor

redis-cli info

If now error is there then redis is configure properly.

1. Configuring Memcached for session cache
2. Configure Varnish as a Full Page Cache
3. Configure Magento 2 with CloudFlare
4. Configuring optimized images in magento 2
5. Configuring Magento 2 with HTTP/2
6. Configuring Magento 2 performance testing