Local Machine Setup Document for Windows with WSL2

Drafted by : Abdulhameed Pathan on 15th June 2022

1. Setup WSL2 on Windows machine

a. Enable the WSL

```
>_ dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

b. Enable Virtual Machine features

```
>_ dism.exe /online /enable-feature/featurename:VirtualMachinePlatform /all /norestartc.
```

c. Download the Linux kernel update package

https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi

Run the update package downloaded in the previous stepd.

d. Set WSL 2 as your default version

```
>_ wsl --set-default-version 2
```

e. Install Linux Distribution

https://aka.ms/wslstore

From the distribution's page (Example, Ubuntu), select "Get"

f. Install Windows Terminal (optional)

https://docs.microsoft.com/en-us/windows/terminal/get-started

2. Enable Internet Access in WSL2

Execute the below commands in Ubuntu terminal

```
a. >_ sudo bash -c 'echo "nameserver 8.8.8.8" > /etc/resolv.conf'
b. >_ sudo bash -c 'echo "[network]" > /etc/wsl.conf'
c. >_ sudo bassh -c 'echo "generateResolvConf = false" >> /etc/wsl.conf'
```

3. Install Services & components on Ubuntu

a. Install nginx

To install Nginx on Ubuntu, run the commands below:

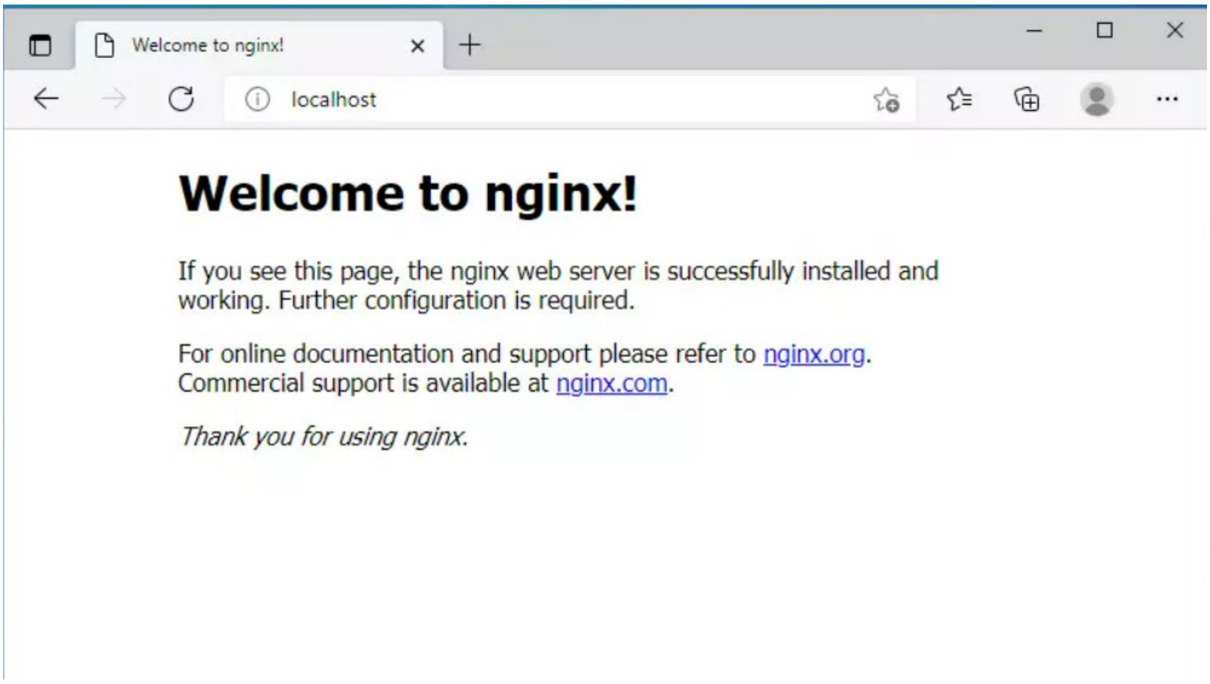
```
sudo apt update
sudo apt install nginx
```

After installing Nginx, the commands below can be used to stop and start Nginx services.

```
sudo service nginx stop
sudo service nginx start
```

To test whether Nginx is installed and functioning, open your web browser and browse to the server’s IP address or hostname.

```
http://localhost
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

If you see the above page in your browser, then Nginx is working as expected.

b. Install MariaDB

To install MariaDB on Ubuntu, run the commands below:

```
sudo apt-get install software-properties-common

sudo apt-key adv --fetch-keys
'https://mariadb.org/mariadb_release_signing_key.asc'

sudo add-apt-repository 'deb [arch=amd64,arm64,ppc64el]
http://suro.ubaya.ac.id/mariadb/repo/10.4/ubuntu bionic main'

sudo apt update

sudo apt install mariadb-server

sudo cp /usr/share/mysql/mysql.init /etc/init.d/mysql

sudo chmod 755 /etc/init.d/mysql

sudo service mysql start

# Set the Mysql Password with the below command
sudo mysql_secure_installation
```

c. Install PHP

To install PHP on Ubuntu, run the commands below:

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update

# Depending on you Project install appropriate Version
sudo apt install php7.4 php7.4-common php7.4-mysql php7.4-xml php7.4-xmlrpc
php7.4-curl php7.4-gd php7.4-imagick php7.4-cli php7.4-dev php7.4-imap
php7.4-mbstring php7.4-opcache php7.4-soap php7.4-zip php7.4-intl -y

sudo service php7.4-fpm start
```

d. Install Git, Curl & Composer

To install Git, Curl & Composer on Ubuntu, run the commands below:

```
sudo apt install curl git

curl -sS https://getcomposer.org/installer | sudo php -- --install-
dir=/usr/local/bin --filename=composer

sudo composer self-update --1
```

d. Install Elasticsearch

To install Elasticsearch on Ubuntu, run the commands below:

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-
key add –

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo
tee -a /etc/apt/sources.list.d/elastic-7.x.list

sudo apt update

sudo apt install elasticsearch

sudo nano /etc/elasticsearch/elasticsearch.yml

# in the above file set network.host : localhost

sudo service elasticsearch start

curl -X GET 'http://localhost:9200'
```

You should see the following response:

```
Output
{
  "node.name" : "My First Node",
  "cluster.name" : "mycluster1",
  "version" : {
    "number" : "2.3.1",
    "build_hash" : "bd980929010aef404e7cb0843e61d0665269fc39",
    "build_timestamp" : "2020-04-04T12:25:05Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
```

e. Create SSH Key

To start using git setup, you first need to create ssh key & add it in your github/gitlab profile.

```
ssh-keygen -t rsa -b 4096 -C "apathan@adobe.com"
```

Now copy the pub key located at ~/.ssh/id_rsa.pub into your github account profile settings.

Note: Upto here we have finished setting up an local environment with all the pre-requisite needed to install an Magento version. The further Project installation steps depends on project to project basis.

----

4. Install Casio-US Project

a. Clone project into local

To clone project on Ubuntu & have entire code base, run the commands below:

```
cd /var/www/html

git clone --recurse-submodules git@github.com:ACSJ-BPS/casio-us.git casio-us

cd casio-us

composer install
```

b. Setup Magento working instance

Till now you have complete codebase and appropriate directories are created. Now lets first setup Magento instance with blank database.

```
sudo bin/magento setup:install --base-url=http://casio-us.local/ --db-host=localhost --db-name=casio_us --db-user=root --db-password=root --admin-firstname=Abdul --admin-lastname=Pathan --admin-email=apathan@adobe.com --admin-user=abdul --admin-password=Abdul123 --language=en_US --currency=USD --timezone=America/Chicago --use-rewrites=1 --search-engine=elasticsearch7 --elasticsearch-host=localhost --elasticsearch-port=9200


sudo chown -R www-data:www-data /var/www/html/casio-us

sudo chmod -R 755 /var/www/html/casio-us


bin/magento deploy:mode:set developer
```

c. Configure virtual host

```
sudo vi /etc/nginx/sites-available/casio-us.local
```

Write the following content in the newly created casio-us.local file

```
upstream fastcgi_backend {
  server  unix:/run/php/php7.4-fpm.sock;
}

server {
  listen 80;
  server_name casio-us.local;
  index  index.php;

  set $MAGE_ROOT /var/www/html/casio-us;
  include /var/www/html/casio-us/nginx.conf.sample;
}
```

Save the file & exit. After saving the file above, run the commands below to enable the new site, then restart Nginx server.

```
sudo ln -s /etc/nginx/sites-available/casio-us.local /etc/nginx/sites-enabled/
sudo service nginx restart
```

d. Add host entry in Windows host file

Since we are using Windows WSL2, we have to make windows aware of any newly added local domain. So add the following entry to the windows host file.

```
#Open the file C:\Windows\System32\drivers\etc\hosts in Notepad as Administrative Privilege.

#Add the following line & save the file.

127.0.0.1 casio-us.local
```

```
Now Access your local environment.
```