Upgrading Ubuntu Server from 12.04 LTS to 18.04 LTS with PHP 5.6 support instead of PHP 7

The purpose of this document is to provide a path for upgrading an Ubuntu Server running version 12.04 LTS to 18.04 LTS while retaining PHP 5.6 as the primary CLI and Apache versions.

There are still applications that break under PHP 7 and still require 5.6 support. While I understand the important depreciation path for these, the *deity-like* decision to completely gut these options from new OS versions gives developers few choices when trying to keep servers up to date.

This document will provide a basic upgrade path and then update the SSLProtocol to TLSv1.2 and SSLCipherSuite to a more robust set of supported ciphers to date.

*Be sure to make backup images of your server **before** making any changes. Also, the standard disclaimer… this upgrade path has taken a good deal of time and research and works well in our testing environment. Your environment might require additional steps. Do the research and know why you are making the changes outlined in this doc. This document is strictly being offered AS IS and if you choose to follow it, you take on and accept all risk.*

Upgrading to Ubuntu 14.04

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install update-manager-core
sudo do-release-upgrade
```

Once the upgrade completes, Apache will fail to start. You will see the error: AH00526: Syntax error on line 49 of /etc/apache2/mods-enabled/ssl.conf

Open the file in a text editor (we use nano for this example)

```
nano /etc/apache2/mods-enabled/ssl.conf
```

Comment out the line starting with: **SSLMutex**

Try to start Apache

```
sudo service apache2 start
```

References
https://itsfoss.com/ubuntu-12-04-end-of-life/

Upgrading to Ubuntu 16LTS

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo apt-get install update-manager-core
sudo do-release-upgrade
```

Note: once the install is completed, we have seen instances where rebooting doesn't complete and we had to force the shutdown.

References:
https://www.digitalocean.com/community/tutorials/how-to-upgrade-to-ubuntu-16-04-lts

Installing PHP 5 on Ubuntu 16

```
sudo apt-get install -y software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install -y php5.6
```

Setting PHP 5.6 as the default version
This sets PHP 5.6 as default while restarting Apache to recognize the change:

```
sudo a2dismod php7.0
sudo a2enmod php5.6
sudo service apache2 restart
```

To maintain symbolic links or the /etc/alternatives path through the update-alternatives command… (be sure to choose php5.6 as the option)

```
sudo update-alternatives --config php
```

PHP Extensions
Adding PHP extensions your application requires (this is an example… add only what you need)

```
sudo apt-get install -y php5.6 php5.6-bcmath php5.6-mcrypt php5.6-json
php5.6-mbstring php5.6-curl php5.6-cli php5.6-mysql php5.6-gd php5.6-intl
php5.6-xsl php5.6-zip libapache2-mod-php5.6
```

Verify the version
You can now verify that PHP 5.6 is the default by running:

```
php -v
```

References:
https://www.liquidweb.com/kb/install-multiple-php-versions-on-ubuntu-16-04/

Updating to Ubuntu 18.04 LTS

Be sure you are doing this portion of the upgrade from the console as eth0 will fail upon completion due to Netplan. If you are using SSH, you will NOT be able to reconnect.

NOTE: Before starting this update, be sure to open /etc/network/interfaces and copy the contents of the file. You will need this to setup Netplan.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo do-release-upgrade
```

References:
https://www.digitalocean.com/community/tutorials/how-to-upgrade-to-ubuntu-18-04

Netplan and how to get eth0 working again

You will need to start eth0 temporarily to SSH into the server. The placeholder YOUR_IP is the server IP address, XX is the subnet mask and YOUR_GATEWAY is the gateway.

Note: the subnet mask might not be 24. I have included a subnet cheat sheet below.

```
sudo ip link set eth0 up
sudo systemctl restart systemd-networkd
sudo ip addr add YOUR_IP/XX dev eth0
sudo ip route add 0/0 via YOUR_GATEWAY
```

References:
Subnet cheat sheet: https://www.aelius.com/njh/subnet_sheet.html

Setting up Netplan

You should be able to SSH into the server again. You need to setup Netplan to handle eth0 going forward. We are using static addresses for our server. If you are using dynamic IPs, you will have to do the research for what Netplan requires. **NOTE**: the indentations are deliberate and **must** be correct of Netplan will fail.

Create a new yaml file to handle this. (NOTE: if you are using Digital Ocean and this is an image seed for future servers, **stop here**. **Do not create the Netplan yaml** as Digital Ocean handles this upon creation.

```
sudo nano /etc/netplan/interfaces.yaml
```

Add the following text with your network requirements. NOTE: the indentations are required.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses:
        - YOUR_IP/XX
      gateway4: YOUR_GATEWAY
      nameservers:
          addresses: [8.8.8.8, 8.8.4.4]
```

Updating SSLProtocol & SSLCipherSuite

The first thing we need to do is find all the files that contain reference to SSLProtocol.

```
grep -i -r "SSLProtocol" /etc/apache2
```

You will most likely see the following files listed. Edit each file using nano and replace the values as demonstrated below.

```
nano /etc/apache2/mods-available/ssl.conf.dpkg-dist
nano /etc/apache2/mods-available/ssl.conf
```

Comment all lines starting with SSLProtocol and then add this:

```
SSLProtocol TLSv1.2
```

Comment all lines starting with SSLCipherSuite and add the ciphers you desire. (we are using ciphers as recommended on the Apache website.

```
SSLCipherSuite
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20
-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-
AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDS
A-AES128-SHA256:ECDHE-RSA-AES128-SHA256
```

Save all files and try to start Apache

```
sudo service apache2 start
```

References:
https://www.digicert.com/ssl-support/apache-disabling-ssl-v3.htm
https://support.plesk.com/hc/en-us/articles/115004991834-How-to-check-what-SSL-TLS-versions-are-available-for-a-website-
https://serverfault.com/questions/848177/how-can-i-disable-tls-1-0-and-1-1-in-apache
https://httpd.apache.org/docs/trunk/ssl/ssl_howto.html

Testing your server

There are plenty of free resources to test SSL/TLS for the server. Here is the one we used:

https://www.ssllabs.com/ssltest/analyze.html