

To start with this Experiment, the Puppet cluster in the previous experiment must be properly configured and running.

1. To test this cluster, run the command-  
`sudo /opt/puppetlabs/bin/puppet agent --test`

We can proceed if the output is normal, such as this:

```
ubuntu@ip-172-31-44-174:~$ sudo /opt/puppetlabs/bin/puppet agent --test
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for ip-172-31-44-174.ap-south-1.compute.internal
Info: Applying configuration version '1682961796'
Info: Creating state file /opt/puppetlabs/puppet/cache/state/state.yaml
Notice: Applied catalog in 0.01 seconds
```

2. Change directories to the production folder  
`cd /etc/puppetlabs/code/environments/production/manifests`
3. Use nano to create a new lamp.pp file  
`sudo nano lamp.pp`

```
ubuntu@ip-172-31-44-174:~$ cd /etc/puppetlabs/code/environments/production/manifests
ubuntu@ip-172-31-44-174:/etc/puppetlabs/code/environments/production/manifests$ ^[[20
p.pp^C
ubuntu@ip-172-31-44-174:/etc/puppetlabs/code/environments/production/manifests$ sudo nano lamp.pp
ubuntu@ip-172-31-44-174:/etc/puppetlabs/code/environments/production/manifests$
```

4. Add the following code to the file. This installs all the dependencies required.

```
# execute 'apt-get update'
exec { 'apt-update':
    # exec resource named 'apt-update'
    command => '/usr/bin/apt-get update' # command this resource will run
}

# install apache2 package
package { 'apache2':
    require => Exec['apt-update'], # require 'apt-update' before installing
    ensure => installed,
}

# ensure apache2 service is running
service { 'apache2':
    ensure => running,
}

# install mysql-server package
package { 'mysql-server':
    require => Exec['apt-update'], # require 'apt-update' before installing
    ensure => installed,
}

# ensure mysql service is running
service { 'mysql':
    ensure => running,
}
```

```

# install php package
package { { 'php':
  require => Exec['apt-update'],    # require 'apt-update' before installing
  ensure => installed,
}

# ensure info.php file exists
file { { '/var/www/html/info.php':
  ensure => file,
  content => '<?php phpinfo(); ?>', # phpinfo code
  require => Package['apache2'],    # require 'apache2' package before creating
}

```

```

GNU nano 6.2
# execute 'apt-get update'
exec { 'apt-update':
  # exec resource named 'apt-update'
  command => '/usr/bin/apt-get update' # command this resource will run
}

# install apache2 package
package { { 'apache2':
  require => Exec['apt-update'],    # require 'apt-update' before installing
  ensure => installed,
}

# ensure apache2 service is running
service { { 'apache2':
  ensure => running,
}

# install mysql-server package
package { { 'mysql-server':
  require => Exec['apt-update'],    # require 'apt-update' before installing
  ensure => installed,
}

# ensure mysql service is running
service { { 'mysql':

```

Click Ctrl+S to save and then Ctrl+X to exit.

5. Change directory to the bin folder of puppetlabs where the puppet executable is present → `cd /opt/puppetlabs/bin`

```

ubuntu@ip-172-31-44-174:/etc/puppetlabs/code/environments/production/manifests$ sudo nano lamp.pp
ubuntu@ip-172-31-44-174:/etc/puppetlabs/code/environments/production/manifests$ cd /opt/puppetlabs/bin

```

6. Use puppet apply to apply the scripts.

```

./puppet apply /etc/puppetlabs/code/environments/production/manifests/lamp.pp
ubuntu@ip-172-31-44-174:/opt/puppetlabs/bin$ sudo ./puppet apply /etc/puppetlabs/code/environments/production/manifests/lamp.pp
Notice: Compiled catalog for ip-172-31-44-174.ap-south-1.compute.internal in environment production in 0.83 seconds
Notice: /Stage[main]/Main/Exec[apt-update]/returns: executed successfully
Notice: /Stage[main]/Main/Package[apache2]/ensure: created

```

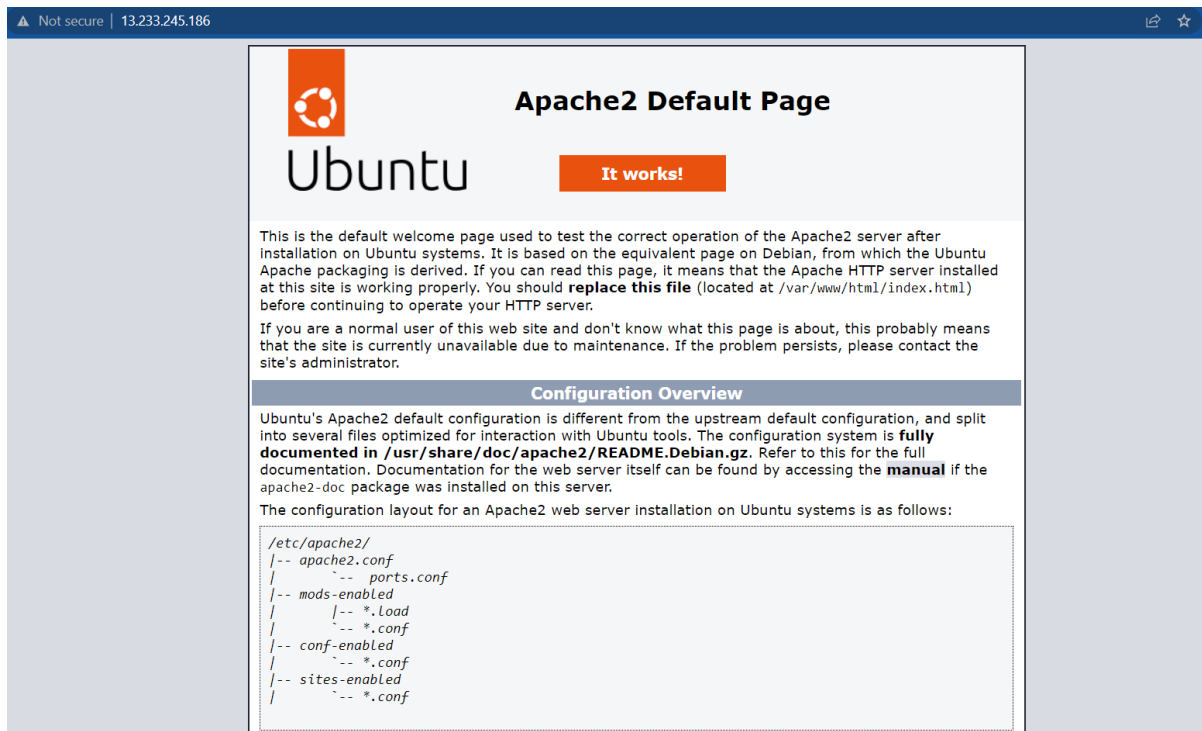
7. Run the following command on the client puppet to install the dependencies same as in master side.

```

ubuntu@ip-172-31-47-90:~$ sudo /opt/puppetlabs/bin/puppet agent --test
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales

```

## 8. Go to the ip-address and check if Apache is running.



▲ Not secure | 13.233.245.186

## Apache2 Default Page

# Ubuntu

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

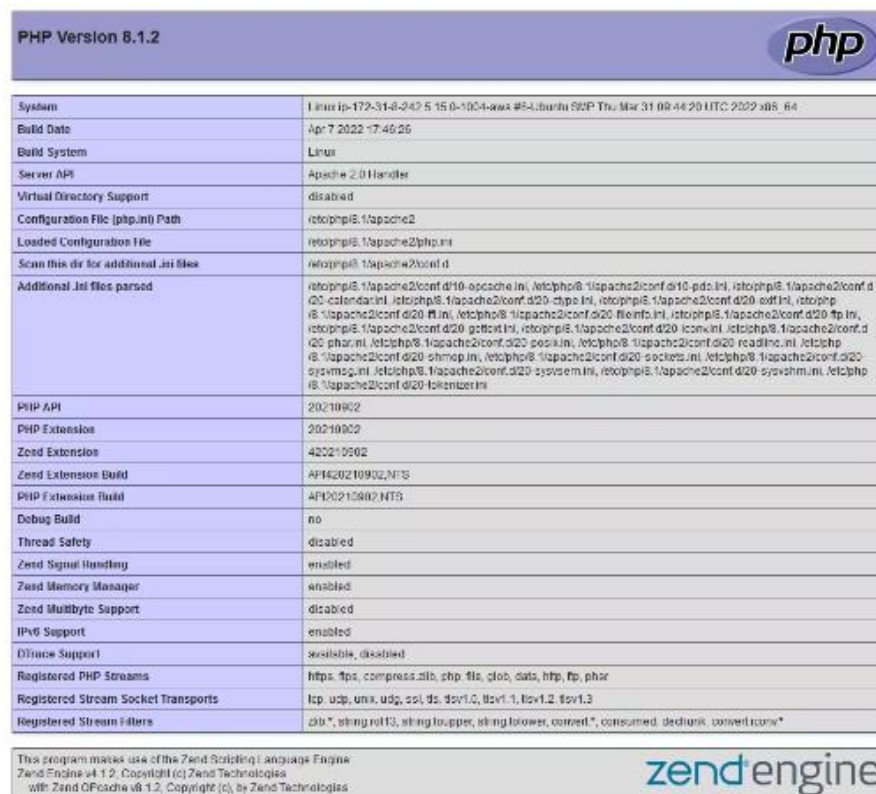
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

9. Go the `http://client_ip_address/info.php`


PHP Version 8.1.2

System	Linux ip-172-31-8-242 5:15:0-1004-aws #54-Ubuntu SMP Thu Mar 31 09:44:20 UTC 2022 x86_64
Build Date	Apr 7 2022 17:46:26
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-apache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-curl.ini, /etc/php/8.1/apache2/conf.d/20-dom.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gdlib.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210502
Zend Extension Build	API20210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
OTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tls_v1, tls_v1.1, tls_v1.2, tls_v1.3
Registered Stream Filters	zlib *, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine  
 Zend Engine v4.1.2, Copyright (c) Zend Technologies  
 with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies

