

Title:- LAMP Server.

Objective :-

The objective of a LAMP server project is to create a robust, scalable, and cost-effective platform for hosting dynamic web applications. By leveraging the open-source nature of Linux, Apache, MySQL, and PHP, the project aims to provide a reliable and secure environment for deploying websites or applications. Key goals include ensuring seamless integration of the software components, optimizing server performance, and enhancing the user experience through efficient database management and dynamic content delivery. Additionally, the project focuses on promoting flexibility and adaptability, allowing for the development of customized solutions tailored to specific business or user requirements. The LAMP server project also emphasizes maintaining security, ensuring data integrity, and supporting ongoing scalability for future growth and advancements.

LAMP Server:-

LAMP installations (Linux Apache MySQL PHP/Perl/Python) are a popular setup for Ubuntu servers. There are a plethora of Open Source applications written using the LAMP application stack. Some popular LAMP applications include wikis, management software such as phpMyAdmin, and Content Management Systems (CMS) like WordPress. The advantage of LAMP is the substantial flexibility for different database web server and scripting languages. Popular substitutes for MySQL include PostgreSQL and SQLite. Python, Perl, and Ruby are also frequently used instead of PHP. While Nginx, Cherokee, and Lighttpd can replace Apache.

❖ **Intern at New Wings IT Solution – Mr. Akshat Milind Joshi**

Process to configure LAMP:

In AWS, LAMP typically refers to a software stack for web hosting, standing for Linux, Apache, MySQL, and PHP (or Python/Perl). Setting up a LAMP stack on AWS involves deploying a Linux instance (usually an EC2 instance), installing Apache web server, MySQL database, and PHP to run web applications.

1. Launch an EC2 Instance (Linux Server)

- **Login to the AWS Management Console.**
- **Go to the EC2 Dashboard.**
- **Click on Launch Instance.**
- **Choose an Amazon Machine Image (AMI):**
 - For a LAMP stack, Mandatory to use **Amazon Linux 2** (Enhanced than AL1)
- **Select an Instance Type:**
 - Choose an instance type (e.g., **t2.micro** for free tier eligibility).
- **Configure Instance:**
 - Configure instance details (you can leave the default settings or adjust according to your needs).
- **Add Storage:**
 - Set the storage size (e.g., 8 GB is typical).
- **Add Tags:**
 - Add a tag like Name: LAMP-Server.
- **Configure Security Group:**
 - Add rules for allowing **HTTP (port 80)**, **HTTPS (port 443)**, and **SSH (port 22)** to access the server.
- **Launch Instance:**
 - Choose an existing key pair or create a new one. This key pair will be used to SSH into the instance.
 - Click **Launch**.
- **Connect to instance:**
 - By selecting the instance connect the instance to open it.

2. Prepare the LAMP server:

- To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.
- Enter:- **yum update -y**
- Install the mariadb10.5 Amazon Linux Extras repositories to get the latest version of the MariaDB package. Info about MariaDB below.

- **MariaDB:**

MariaDB is an open-source relational database management system (RDBMS) that is a fork of MySQL. It was created by the original developers of MySQL after concerns about MySQL's acquisition by Oracle. MariaDB is fully compatible with MySQL, meaning that it can be used as a drop-in replacement for MySQL in most cases, but it also introduces additional features, performance improvements, and a more open development model.

Key Features of MariaDB:

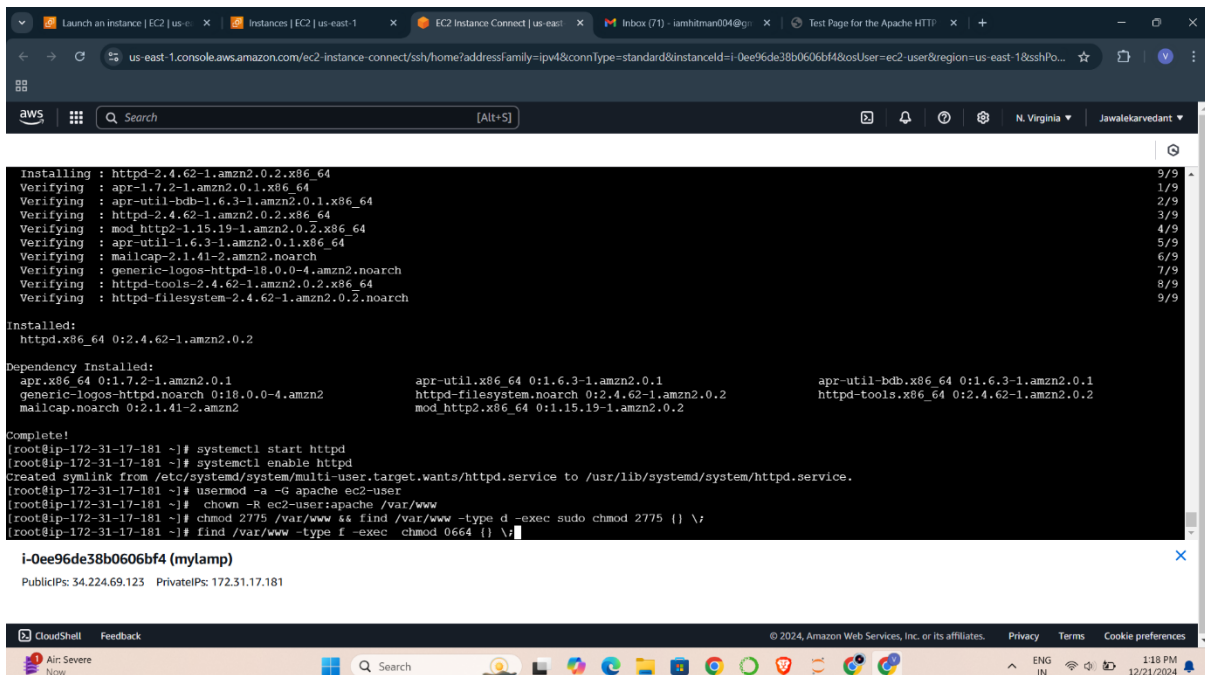
1. Compatibility: MariaDB maintains high compatibility with MySQL, including its client API, data types, and table formats.
 2. Storage Engines: MariaDB offers multiple storage engines, such as InnoDB (the default), Aria, TokuDB, and more, providing flexibility for different use cases.
 3. Performance: MariaDB often delivers improved performance over MySQL in certain workloads, with optimizations in query execution and indexing.
 4. Security: It includes several security enhancements, like enhanced password encryption, SSL support, and more.
 5. Scalability: MariaDB supports master-slave replication, Galera Cluster for synchronous multi-master replication, and other high-availability solutions.
 6. Open Source: MariaDB is open source and maintained by the MariaDB Foundation, making it fully open and transparent.
- Command to enter:- **amazon-linux-extras install mariadb10.5**

1. yum install httpd -y
2. systemctl start httpd
3. systemctl enable httpd

3. Apache httpd serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is /var/www/html, which by default is owned by root.

- To allow the ec2-user account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. In this tutorial, you add ec2-user to the apache group, to give the apache group ownership of the /var/www directory and assign write permissions to the group

1. usermod -a -G apache ec2-user
2. chown -R ec2-user:apache /var/www
3. chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
4. find /var/www -type f -exec sudo chmod 0664 {} \;



```
Installing : httpd-2.4.62-1.amzn2.0.2.x86_64
Verifying : apr-1.7.2-1.amzn2.0.1.x86_64
Verifying : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64
Verifying : httpd-2.4.62-1.amzn2.0.2.x86_64
Verifying : mod_http2-1.15.19-1.amzn2.0.2.x86_64
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64
Verifying : mailcap-2.1.41-2.amzn2.noarch
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch
Verifying : httpd-tools-2.4.62-1.amzn2.0.2.x86_64
Verifying : httpdfilesystem-2.4.62-1.amzn2.0.2.noarch

Installed:
httpd.x86_64 0:2.4.62-1.amzn2.0.2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2.0.1          apr-util.x86_64 0:1.6.3-1.amzn2.0.1          apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httpd.noarch 0:18.0.0-4.amzn2    httpd-filesystem.noarch 0:2.4.62-1.amzn2.0.2          httpd-tools.x86_64 0:2.4.62-1.amzn2.0.2
mailcap.noarch 0:2.1.41-2.amzn2              mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[root@ip-172-31-17-181 ~]# systemctl start httpd
[root@ip-172-31-17-181 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-17-181 ~]# usermod -a -G apache ec2-user
[root@ip-172-31-17-181 ~]# chown -R ec2-user:apache /var/www
[root@ip-172-31-17-181 ~]# chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
[root@ip-172-31-17-181 ~]# find /var/www -type f -exec sudo chmod 0664 {} \;
```

i-0ee96de38b0606bf4 (mylamp)

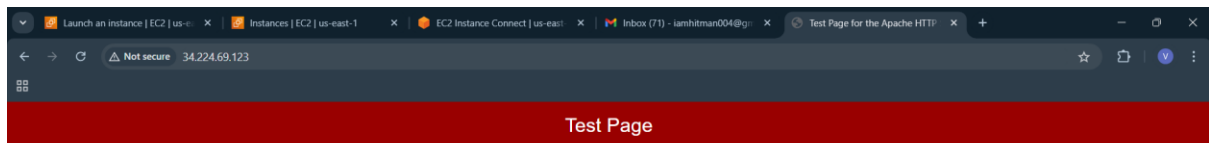
PublicIPs: 34.224.69.123 PrivateIPs: 172.31.17.181

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Air: Severe Now

ENG IN 1:18 PM 12/21/2024



This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



4. Test your LAMP server :

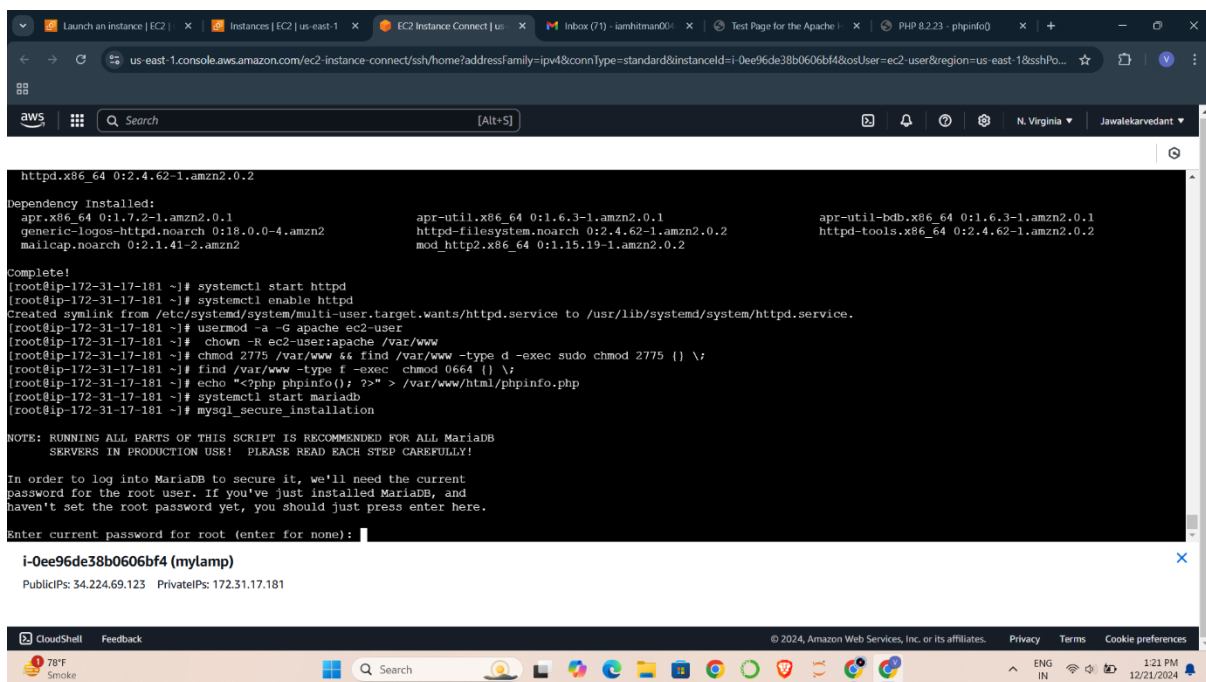
- If your server is installed and running, and your file permissions are set correctly, your ec2-user account should be able to create a PHP file in the `/var/www/html` directory that is available from the internet.

1. `echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php`
2. <http://my.public.dns.amazonaws.com/phpinfo.php>

PHP Version 8.2.23	
System	Linux ip-172-31-17-181.ec2.internal 5.10.230-223.885.amzn2.x86_64 #1 SMP Tue Dec 3 14:36:00 UTC 2024 x86_64
Build Date	Sep 15 2024 21:02:00
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqli.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-zip.ini, /etc/php.d/30-mysql.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829.NTS
PHP Extension Build	API20220829.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar, zip

5. Secure the database server:

- The `mysql_secure_installation` command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MariaDB server, we recommend performing this procedure.
 1. `systemctl start mariadb`
 2. `mysql_secure_installation`
 3. `systemctl stop mariadb`
 4. `systemctl enable mariadb`



```
httpd.x86_64 0:2.4.62-1.amzn2.0.2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2.0.1          apr-util.x86_64 0:1.6.3-1.amzn2.0.1          apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httpd.noarch 0:18.0.0-4.amzn2  httpd-filesystem.noarch 0:2.4.62-1.amzn2.0.2  httpd-tools.x86_64 0:2.4.62-1.amzn2.0.2
mailcap.noarch 0:2.1.41-2.amzn2          mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[root@ip-172-31-17-181 ~]# systemctl start httpd
[root@ip-172-31-17-181 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-17-181 ~]# usermod -s /bin/bash ec2-user
[root@ip-172-31-17-181 ~]# chown -R ec2-user:apache /var/www
[root@ip-172-31-17-181 ~]# chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
[root@ip-172-31-17-181 ~]# find /var/www -type f -exec chmod 0664 {} \;
[root@ip-172-31-17-181 ~]# echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
[root@ip-172-31-17-181 ~]# systemctl start mariadb
[root@ip-172-31-17-181 ~]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
```

6. Navigate to the Apache document root at `/var/www/html`:

1. `cd /var/www/html`
 2. `vim index.php`
- Write the below content in vim file:

```
<?php
$servername = "localhost";
$username = "root";
$password = "redhat";

$conn = new mysqli($servername, $username, $password);

if ($conn->connect_error) {
    die("Connection failed:" . $conn->connect_error);
}

echo "Connected Successfully";
?>
```

7. To Check the LAMP Configuration :

- Copy the Public IP Address on Browser
- Check if the Entered Message Displayed .
-

