Project - AWS LAMP-Web Server

Professor Navarro

COP3834 – Web Application Development

Background

The goal of this product is to establish a public-facing, cloud infrastructure that hosts a LAMP technology stack.

LAMP is an acronym that describes one of the most common web server technology stacks in the world. It is based on:

- **L**inux: The operating system
- **A**pache: The web server
- **M**ySQL: The database
- **P**HP: The programming language

The LAMP process starts with Apache web server receiving http requests for web pages from a client's web browser. Apache will respond with HTML and CSS but also pass certain requests to PHP for specific processing which may include database calls. Note that the MariaDB is interchangeable with MySQL and can be used in place of that database.

In this project, your LAMP stack will be built on Amazon Web Services. This will use the AWS Elastic Compute Cloud (EC2) to host your LAMP stack and **future assignments**. The EC2 provides a secure, resizable compute capacity in the cloud. Additionally, AWS is a cloud resource used by large and small organizations around the world. The technology is utilized by companies like Adobe, Apple, Facebook, JP Morgan-Chase, NASA, Netflix, Spotify, Square Enix, Twitch, Ubisoft, and Verizon.

This project is to be submitted individually but you are free to collaborate with your peers. However, each student will need to have their own AWS EC2 instance with LAMP installed.

Amazon offers new account holders a free year of AWS services with $1 secured to a credit card in case users go beyond their free tier. Only certain AWS elements are free that are clearly labeled when you register for a free account. Also, only 750 hours per month of use is available for the free tier. This means a single EC2 instance is free. Two EC2 instances set up and online would be charged for one instance. At the end of the semester, I will remind students to shut down their EC2 instances to avoid being charged in the future. If you already have an AWS account, you may want to consider creating a new AWS account just for this course.

This project has multiple components and requires you to research the steps needed to build an EC2 instance as well as the required LAMP technology stack. Technical resources will be posted in Canvas as well as in class guidance on developing the stack.

When setting up this infrastructure, think critically about how organizations might use AWS. Why would these organizations adopt a cloud infrastructure instead of building or using their own data centers? How transferable are the skills you are gaining in AWS to other organizations that may choose to build on AWS?

Procedure

1. Set up a free AWS account at https://aws.amazon.com/free

   (I recommend using a non-poly email for this account)

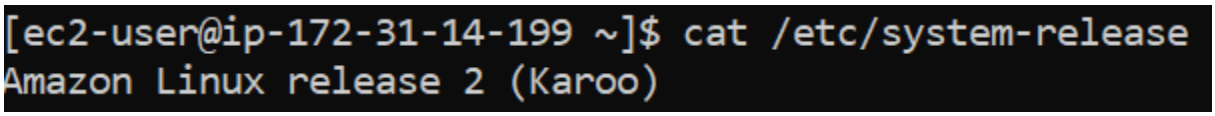
2. Create an EC2 Instance with a Linux operating system.

3. Add Apache 2.x to the EC2 Instance. You do not need a https build.

4. Add MySQL or MariaDB database.

5. Add PHP to the EC2 Instance.

6. Create a screenshot (or multiple screenshots) In a word doc or pdf of the following commands in a Linux terminal with output to confirm they are running. An example Is:



   a. `cat /etc/system-release` (to view Linux distro)

   b. `/opt/aws/bin/ec2-metadata -p` (to view EC2 host)

   c. `ifconfig eth0 | grep inet` (to view IP address)

   d. `service httpd status` (to view apache status)

   e. `service mysql status` **or** `service mariadb status` (to view database status)

   f. `php -v` (to view php and version status)

```
[ec2-user@ip-172-31-46-49 ~]$ cat /etc/system-release
Amazon Linux release 2 (Karoo)
[ec2-user@ip-172-31-46-49 ~]$ /opt/aws/bin/ec2-metadata -p
public-hostname: ec2-3-20-206-48.us-east-2.compute.amazonaws.com
[ec2-user@ip-172-31-46-49 ~]$ ifconfig eth0 | grep inet
        inet 172.31.46.49  netmask 255.255.240.0  broadcast 172.31.47.255
        inet6 fe80::85d:49ff:fef9:9222  prefixlen 64  scopeid 0x20<link>
[ec2-user@ip-172-31-46-49 ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Sun 2022-09-25 00:27:33 UTC; 36min ago
     Docs: man:httpd.service(8)
 Main PID: 4538 (httpd)
   Status: "Total requests: 9; Idle/Busy workers 100/0;Requests/sec: 0.00413; Bytes served/sec: 575 B/sec"
   CGroup: /system.slice/httpd.service
           ├─4538 /usr/sbin/httpd -DFOREGROUND
           ├─4544 /usr/sbin/httpd -DFOREGROUND
           ├─4545 /usr/sbin/httpd -DFOREGROUND
           ├─4546 /usr/sbin/httpd -DFOREGROUND
           ├─4547 /usr/sbin/httpd -DFOREGROUND
           ├─4548 /usr/sbin/httpd -DFOREGROUND
           ├─5448 /usr/sbin/httpd -DFOREGROUND
           ├─5506 /usr/sbin/httpd -DFOREGROUND
           ├─5512 /usr/sbin/httpd -DFOREGROUND
           └─5513 /usr/sbin/httpd -DFOREGROUND

Sep 25 00:27:33 ip-172-31-46-49.us-east-2.compute.internal systemd[1]: Starting The Apache HTTP Server...
Sep 25 00:27:33 ip-172-31-46-49.us-east-2.compute.internal systemd[1]: Started The Apache HTTP Server.
```

```
[ec2-user@ip-172-31-46-49 ~]$ service mariadb status
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB 10.2 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/mariadb.service.d
           └─tokudb.conf
   Active: active (running) since Sun 2022-09-25 00:47:04 UTC; 17min ago
  Process: 5366 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status=0/SUCCESS)
  Process: 5170 ExecStartPre=/usr/libexec/mysql-prepare-db-dir %n (code=exited, status=0/SUCCESS)
  Process: 5146 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
 Main PID: 5294 (mysqld)
   Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─5294 /usr/libexec/mysqld --basedir=/usr

Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal mysql-check-upgrade[5366]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so.1' from LD_PRELOAD cannot be ...nored.
Sep 25 00:47:04 ip-172-31-46-49.us-east-2.compute.internal systemd[1]: Started MariaDB 10.2 database server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-46-49 ~]$ php -v
PHP 7.2.34 (cli) (built: Oct 21 2020 18:03:20) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
[ec2-user@ip-172-31-46-49 ~]$
```

7. In the same word doc or pdf, answer the following Critical Thinking questions (2-5 sentences each):

    a. Who benefits from cloud-based servers? Give both general and specific examples.

Everyone, It makes It easier for both the developer and the user. Developers can now access their system from anywhere remotely as long as It has service. This would come In handy In many Instances, for example If you go on vacation and don't have access to your work computer, but can access a regular computer, you can

now log In to fix bugs on your website without having to worry about waiting till you get back. This would also benefit the user as it means less wait time for things to get fixed. It also leads to better connections for something to be cloud based as It has the cloud-based Infrastructure of being accessible from everywhere as opposed to being hosted on a local machine where If you were located on the other half of the world would be much harder to access If even possible.

b. What are the strengths/weaknesses of cloud-based servers?

Strength:

Can be accessed from pretty much everywhere.

It Is cheaper because It requires less strong machines to be located around the world, you just have to put It on the already built cloud based Infrastructure.

Cloud based systems have been found to be much more reliable than local systems.

Disadvantage:

They can be easier to hack Into being that you can access them from anywhere and therefore you can hack It from China as opposed to having to travel to the location of a more local system.

You need access to the Internet In order to do anything. So, If you are In a place with bad service or no service you cannot access anything or might experience horrible speeds.

With cloud-based systems, unless you are launching your own satellite Into space, you will need to work with a company for cloud based hosting. This can pose Issues such as not having total control of your content or pricy/hard to get out of contracts.

c. Where Is the most need for cloud-based servers?

I feel like this Is a bit of an opinionated question, but I feel most people can agree that the main groups that would benefit from cloud-based systems are the medical and military Industries. The medical Industry would benefit because It would make sharing patient Information between doctors and hospitals much easier If you could just upload their Information to the portal and then a doctor half way across the world would be able to access It real time as apposed to waiting for even a fax or email which are slow and less private. The military would benefit by being able to share real time data such as enemy positioning, commands from headquarters, weather, and more from anywhere in the world In real time.

d. Why Is learning how to build a cloud Infrastructure relevant to you? Consider both your academic and future career life.

Well In a more general sense cloud computing Is important because It Is the fastest growing and most popular hosting system and more likely than not If you are working for a company that Is developing a digital product, It will be hosted on the cloud. This means to stay up to date and knowledgeable about the current development ecosystem It Is a necessity to learn cloud-based development. As for academic, It's the same reasoning, except this Is your opportunity to learn about cloud based computing and how It works so that you can then display that knowledge through projects or Interviews when looking for a job and put yourself ahead of those who haven't had as much experience.

e. How Is the project similar to technology or practices you may experience later In your career? Create a hypothetical scenario If required.

Say I am working on a project that needs to be accessible to a team across the world for them to access and make changes to. Well, we would use a hosting system similar to AWS to host all of our files on and give us a terminal which would then allow us to run It like any regular Linux computer, just one that can be accessed from anywhere. We would then be able to use that system to develop anything you would be able to do on a regular local Linux computer, just with worldwide access. In my field, I am wanting to get into medical simulations, but I might want to share my work with doctors and programmers across the world to be able to give Input and make changes. A system like this would give us the ability to not only edit but run the project files from anywhere making development that much easier.

8. A single Word doc or PDF should be submitted to Canvas with the screen shots and Critical Thinking questions.

9. Extra Credit

http://ec2-3-20-206-48.us-east-2.compute.amazonaws.com/index.html#

Project - AWS LAMP Web Server Rubric

Assessment	**Points**
Evidence of a working EC2 instance	20
Evidence of a working Linux OS on the EC2	15
Evidence of a working Apache on the EC2	15
Evidence of a working Mysql or MariaDB database on the EC2	15
Evidence of a working PHP on the EC2	15
5 Critical Thinking questions answered	20
Sub total	100

Extra Credit – Host your L03 homework index.html web page on the apache server where it is accessible to the public internet. -Add a visible <p> element in the footer of the page that states “this is a test site”. Provide a URL on your word doc or pdf, labeled Extra Credit, for me to test it with a browser.	5

Example of EC2 Dashboard and Instance Manager: