**EXPERIMENT NO: 12**

**DEVOPS**

**AIM:** To provision a LAMP/MEAN Stack using puppet manifest.

**LO:6** – Synthesis software configuration and provisioning using Ansible.

**THEORY:**

**INTRO TO LAMP STACK:**

* Learn more about the time-tested stack used throughout the open source community
* Program with a dynamic language you can quickly get into and see results
* Build on customizable modules to create and deploy web apps your way
* Add valuable new knowledge and skills to your resume

You may have heard something about the LAMP stack.

But LAMP has a lot more to its credit. It was one of the first open source software stacks for the web and remains one of the most common ways to deliver web applications. It is so widely used that you are likely to encounter it frequently during your career as you update or host existing applications. And it is considered by many to be the platform of choice for developing new custom web apps.

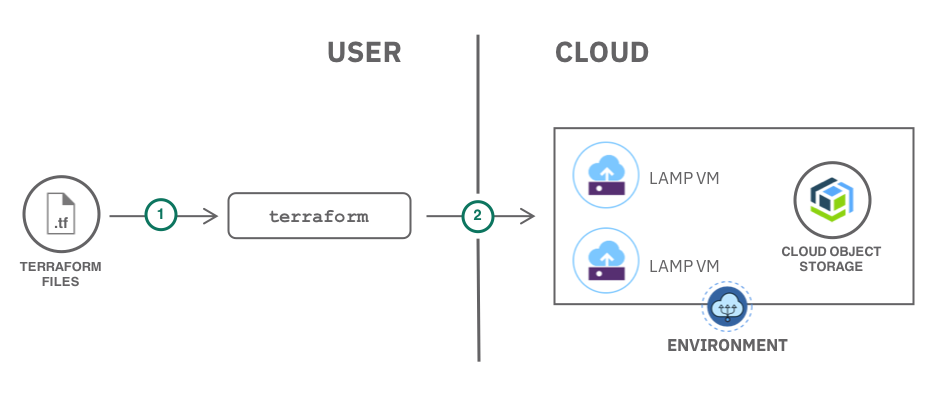
Stable, simple, powerful—these are words most often used to describe LAMP. All of this makes it well worth learning about and a valuable addition to any developer’s resume.

**COMPONENTS:**

LAMP stands for Linux, Apache, MySQL, and PHP. Together, they provide a proven set of software for delivering high-performance web applications. Each component contributes essential capabilities to the stack:

* **Linux: The operating system.** Linux is a free and open source operating system (OS) that has been around since the mid-1990s. Today, it has an extensive worldwide user base that extends across industries. Linux is popular in part because it offers more flexibility and configuration options than some other operating systems.
* **Apache: The web server.** The Apache web server processes requests and serves up web assets via HTTP so that the application is accessible to anyone in the public domain over a simple web URL. Developed and maintained by an open community, Apache is a mature, feature-rich server that runs a large share of the websites currently on the internet.
* **MySQL: The database.** MySQL is an open source [relational database management system](https://www.ibm.com/cloud/learn/relational-databases) for storing application data. With My SQL, you can store all your information in a format that is easily queried with the SQL language. SQL is a great choice if you are dealing with a business domain that is well structured, and you want to translate that structure into the backend. MySQL is suitable for running even large and complex sites. See "[SQL vs. NoSQL Databases: What's the Difference?](https://www.ibm.com/cloud/blog/sql-vs-nosql)" for more information on SQL and NoSQL databases.
* **PHP: The programming language.** The PHP open source scripting language works with Apache to help you create dynamic web pages. You cannot use HTML to perform dynamic processes such as pulling data out of a database. To provide this type of functionality, you simply drop PHP code into the parts of a page that you want to be dynamic.

PHP is designed for efficiency. It makes programming easier—and a bit more fun—by allowing you to write new code, hit refresh, and immediately see the resulting changes without the need for compiling. If you prefer, you can swap out PHP in favor of Perl or the increasingly popular Python language.

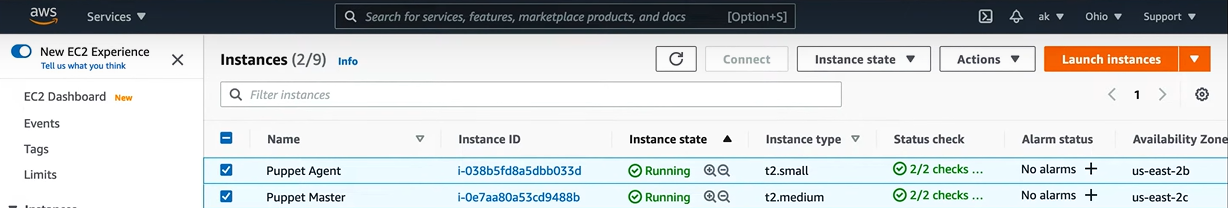


**LAMP architecture:**

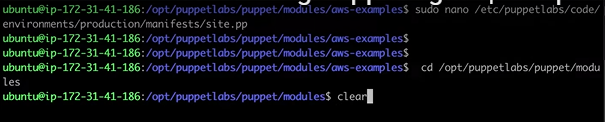
LAMP has a classic layered architecture, with Linux at the lowest level. The next layer is Apache and MySQL, followed by PHP. Although PHP is nominally at the top or presentation layer, the PHP component sits inside Apache.

**OUTPUT:**

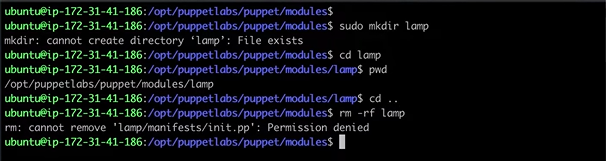
agent.



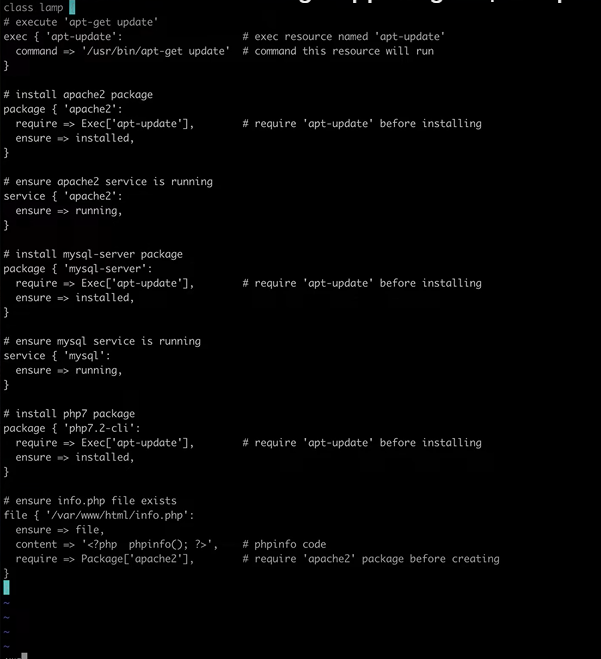
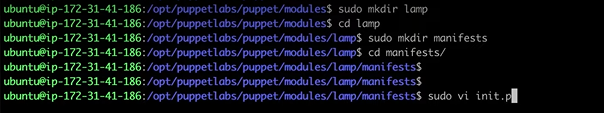
Step 2: Change the directory to the puppet modules.



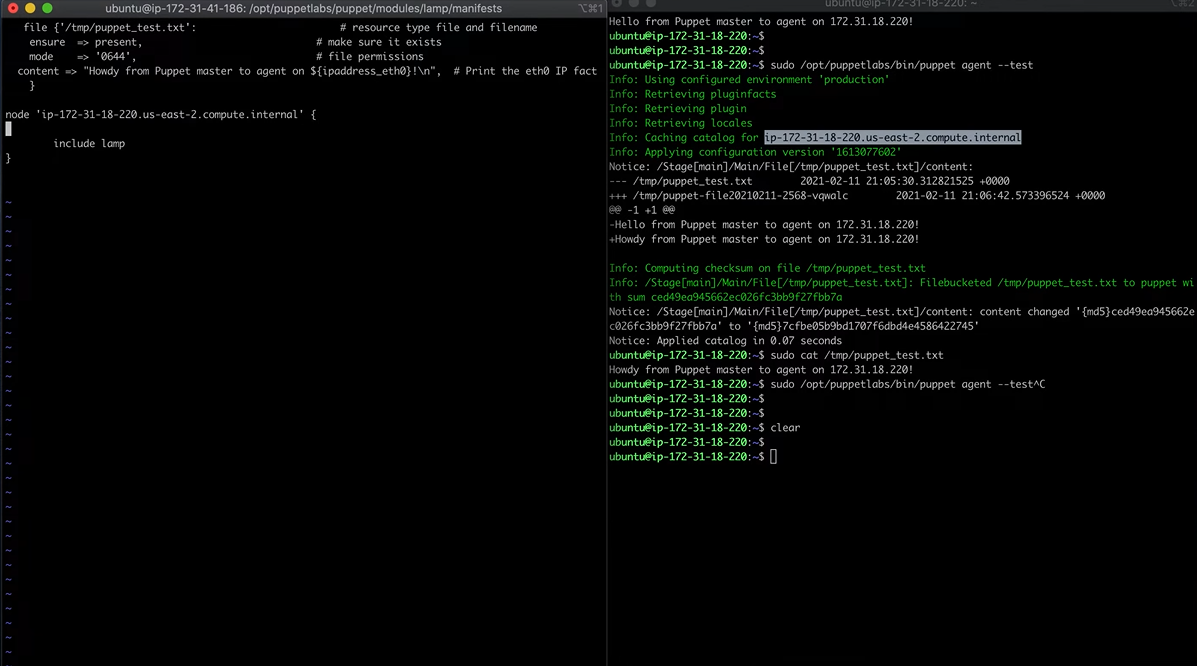
**Step 3:** Create the lamp directory to install the LAMP stack.



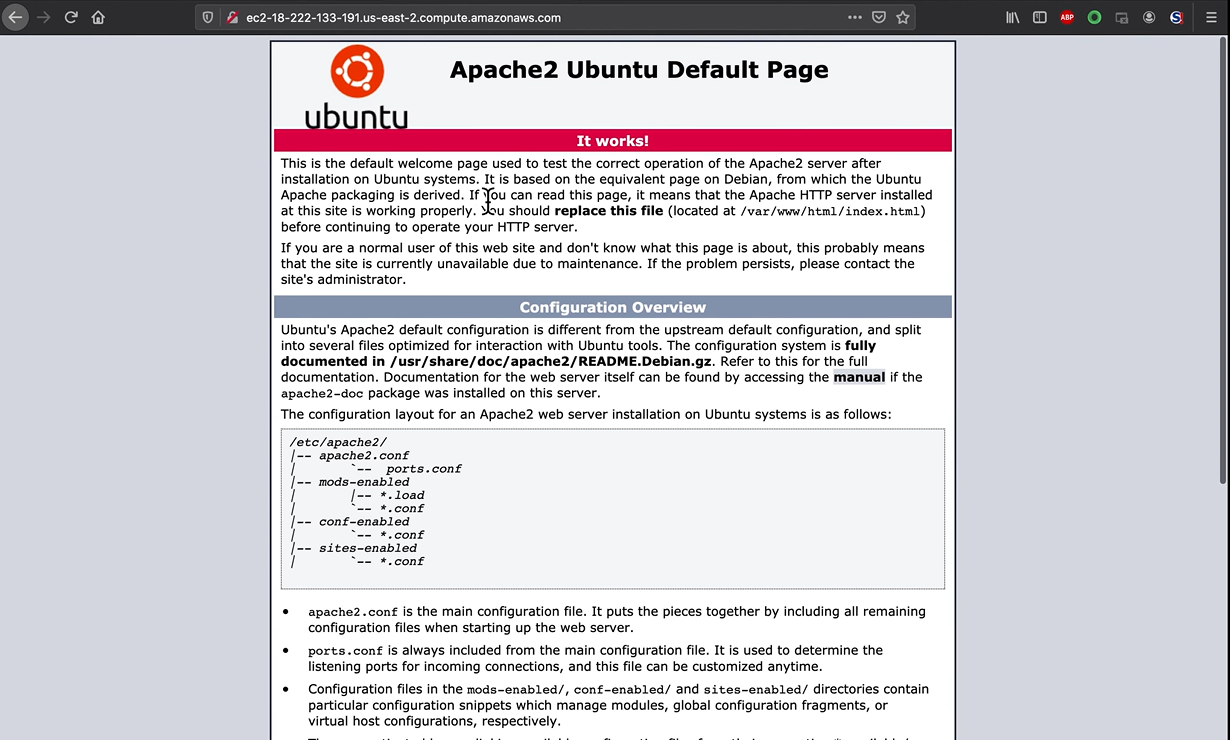
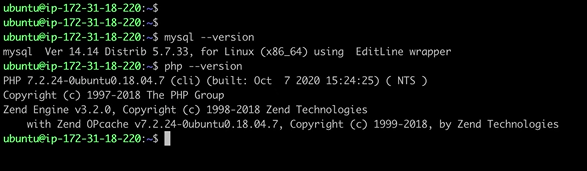
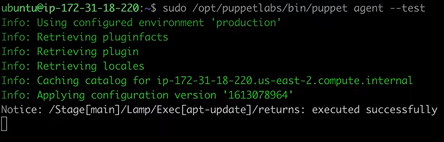
Step 4: open init.pp and the following code to install the LAMP stack.



Step 5: Make the configuration in the following file, get to know about the agent’s IP.



Step 6: Installing the necessary tools required



**CONCLUSION:**