Objective of the Project

The primary objective of the project is to **deploy Moodle**, an open-source Learning Management System (LMS), on an **Amazon EC2 instance** running Ubuntu. This deployment aims to create a scalable, customizable, and accessible online learning platform that can be used by educational institutions, organizations, or individuals for creating and managing courses, content, and user interactions over the internet.

Key Goals

1. **Set Up a Server Environment**: Utilize Amazon EC2 to provision a virtual server with Ubuntu installed, ensuring it meets Moodle's minimum system requirements (at least 5 GB of disk space, 1 CPU core, and 500 MB of memory).
2. **Install Necessary Software Stack**: Install and configure the LAMP stack (Linux, Apache, MySQL, PHP), which is essential for running web applications like Moodle.
3. **Secure the Server**: Implement security measures such as updating system packages, securing MySQL installation, and setting appropriate file permissions to protect the server and data.
4. **Install Moodle Application**: Download and set up the Moodle application by cloning its repository, configuring it to work with the web server and database, and ensuring all dependencies are met.
5. **Configure the Database**: Create a dedicated MySQL database and user for Moodle, configuring it with the correct character set and collation to support internationalization.
6. **Complete Web-Based Installation**: Access the Moodle installation script via a web browser to finalize the setup, configure site settings, and create an admin account.
7. **Finalize and Secure Configuration**: Adjust any temporary permissions used during installation back to secure settings, and perform additional configurations like setting system paths for optimal functionality.

Detailed Breakdown

1. **Provisioning an EC2 Instance**:
   - Launch an EC2 instance using the Amazon Web Services (AWS) console.
   - Choose an Ubuntu Server AMI (Amazon Machine Image).
   - Configure instance details to meet the minimum requirements for Moodle.
2. **Updating System Packages**:
   - Use `apt update` and `apt upgrade` to ensure the server's software is up to date.
   - Install `software-properties-common` to manage PPAs.

3. **Installing Apache, MySQL, and PHP**:
   - Add the PHP PPA to access PHP 7.4, as required by Moodle 4.0.
   - Install Apache (`apache2`), MySQL (`mysql-server`), and PHP 7.4 along with necessary modules.
4. **Securing MySQL Installation**:
   - Run `mysql_secure_installation` to set the root password and remove insecure defaults.
   - Configure MySQL to use the InnoDB engine and optimize settings for Moodle.
5. **Installing Additional PHP Extensions and Software**:
   - Install required PHP extensions like `php7.4-gd`, `php7.4-intl`, `php7.4-xml`, etc.
   - Install additional software such as `graphviz`, `aspell`, `ghostscript`, and `clamav` for enhanced Moodle functionalities.
6. **Downloading Moodle**:
   - Navigate to the `/opt` directory.
   - Clone the Moodle Git repository using `git clone`.
   - Checkout the stable branch for Moodle 4.0 (`MOODLE_400_STABLE`).
7. **Configuring Moodle Application**:
   - Copy Moodle files to the web server's root directory (`/var/www/html/`).
   - Create a data directory (`/var/moodledata`) for Moodle to store its files outside the web root.
   - Set ownership and permissions to allow the web server to access these directories.
8. **Setting Up the Database**:
   - Create a new MySQL database named `moodle` with the appropriate character set and collation.
   - Create a MySQL user (`moodledude`) with a secure password.
   - Grant necessary privileges to the Moodle user on the Moodle database.
9. **Completing Moodle Installation via Web Interface**:
   - Temporarily adjust permissions to allow the web installer to create configuration files.
   - Access the Moodle installation script through a web browser using the EC2 instance's public IP or domain name.
   - Follow the on-screen prompts to complete the installation, including entering database credentials, configuring the admin user, and setting site preferences.
10. **Finalizing Configuration and Securing the Installation**:
    - Revert any temporary permissions set during installation to secure the directories.
    - Configure system paths within Moodle for additional functionalities (e.g., paths to `du`, `aspell`, `dot`).
    - Restart services as necessary to apply all changes.

Outcome

By completing these steps, the project achieves the deployment of a fully functional Moodle LMS on an EC2 instance. This setup allows for:

- **Scalability**: Ability to adjust resources based on demand using AWS features.
- **Accessibility**: Moodle can be accessed globally, facilitating remote learning.
- **Customization**: Full control over the Moodle environment to install plugins, themes, and configure settings to suit specific educational needs.
- **Security**: Implemented security best practices to protect user data and application integrity.

Benefits of Deploying Moodle on EC2

- **Cost-Effective**: Pay-as-you-go pricing model allows for cost management based on actual usage.
- **Flexibility**: Easy to scale up or down and customize server configurations.
- **Reliability**: AWS provides high availability and redundancy features to ensure the LMS is consistently accessible.
- **Integration**: Ability to integrate with other AWS services for enhanced functionality (e.g., backups with S3, monitoring with CloudWatch).

Next Steps After Deployment

- **Configure Courses and Content**: Begin creating courses, enrolling users, and adding educational content.
- **Install Plugins and Themes**: Enhance Moodle's functionality and appearance by installing additional plugins and themes.
- **Set Up SSL Encryption**: Secure the website with HTTPS by installing an SSL certificate, which can be obtained through AWS Certificate Manager or Let's Encrypt.
- **Implement Regular Backups**: Configure automated backups of the Moodle database and data directory to prevent data loss.
- **Monitoring and Maintenance**: Set up monitoring tools to keep track of server performance and perform regular updates to maintain security.

Conclusion

The project's objective is to establish a robust, secure, and efficient online learning platform by deploying Moodle on an Amazon EC2 instance. This setup leverages AWS's scalable infrastructure and Moodle's comprehensive LMS features to provide a powerful tool for educators and learners alike.

By following the outlined steps, you ensure that the Moodle installation is correctly configured, secure, and ready for immediate use in delivering online education.