

Hosting LinearAlgebraTutor project

Abstract

Our group will set up the necessary components to host the educational web application LinearAlgebraTutor. In order to host the web application, we will set up an instance of AWS Elastic Beanstalk. In order to store user data, such as the user's points they have gained, we will set up and configure a database instance using AWS RDS.

Introduction

We have found AWS to be a very robust system. So far, a major point our group has understood from our initial foray to learn AWS is that it has near countless features; all with unique names that aren't used elsewhere. We find AWS to be a useful system and thus we would like to improve the depth of our AWS knowledge

Ashton already has a completed project "LinearAlgebraTutor", which is a web application that he would like to learn how to deploy over AWS. This web application is developed using python with the flask framework. We will use AWS Elastic Beanstalk to host the web application, and AWS RDS to host the database holding user data.

We chose these two services to implement and learn more about because they fit the bill to host the project, and are relevant to our future interests. The Elastic Beanstalk service states that it supports python web applications, which is what the LinearAlgebraTutor project is. AWS RDS is Amazon's Relational Database System. Relational Databases are a commonly used method of storing data.

By completing this project users will be able to access the LinearAlgebraTutor web application, aiding in their linear algebra education.

Security Discussion

We will use the AWS documentation as a reference for ensuring the web application and the database are set up securely. We have come up with a list of security issues that could arise, listed below.

Possible Security Issues:

- Cross-Site Scripting attack
 - Solution: To prevent XSS developers have to make sure to escape html text. Flask escapes on its own, thus all problems could only be from the three html pages the website includes. I will check these pages out for this possible issue.
- Cross-Site Request Forgery
- Users bypass account authentication

- Solution: Passwords encrypted before sending and are compared against a hash.
- User account info gets stolen:
 - Solution: Data sent over SSL/TLS
- Unauthorized users access database instance:
 - Solution: Only the web application is able to access the database.
- Users send unauthorized data to database instance
 - Solution: Ensure SQL injection is not possible

Maintaining for Future Use:

Elastic Beanstalk has some features that would allow for it to be more easily installed and maintained, however, like all applications some manual maintenance would inevitably be required. Some features that could be implemented for automated maintenance include:

- Auto Scaling Group -
<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.as.html>
- Amazon Cloudwatch -
<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/AWSHowTo.cloudwatch.html>
- Monitoring Environment - See Assignment 5
- Adding a Puppet module (or some system configuration management) to automate installation.

Project Task List

1. Deploy LinearAlgebraTutor project on Elastic Beanstalk
 - a. Project Accessible over the web
 - b. Elastic Beanstalk configured correctly
 - i. Capacity Provisioning
 - ii. Load Balancing
 - iii. Application Health Monitor
 - c. Each group member give input on what they learned about the Elastic Beanstalk service.
2. Create an AWS relational database to host project data
 - a. Database has the following information:
 - i. Table: Users
 1. Username
 2. Password Hash
 3. Number Of Points
 4. Correct answers
 - b. Configure LinearAlgebraTutor application on Elastic Beanstalk to communicate with database
 - i. User Sign in

- ii. Track which modules have been completed
- iii. Track user points