Akilandeshwari Srinivasan (451036)

CLCM3102-Lab7

Selecting the Right AWS Services for Deploying a React and Python Web Application with Project Management

Component 1: Web Application Hosting:

I can choose to use AWS Amplify for my web or mobile application deployment. It offers an all-in-one solution for both the frontend and backend, making it a straightforward way to create and launch my app. Amplify is compatible with common web frameworks and provides various backend services. If I prefer a fully managed option that takes care of much of the infrastructure work for me, AWS Amplify is a great choice.

Component 2: Scalability and Load Balancing

For ensuring effective scalability and high availability with auto-scaling, I would consider using AWS Elastic Beanstalk. It provides a good balance between simplicity and control, allowing me to automatically scale my application based on traffic demands. With Elastic Beanstalk, I can ensure my application can handle increased loads while still benefiting from the convenience of automatic scaling.

Component 3: Development Environment:

As I consider the development environment for our Python and React applications, using AWS Cloud9 is a good option, AWS Cloud9 is a cloud-based integrated development environment (IDE) that provides collaborative coding features and cloud-based development tools. For a Python and React application, Cloud9 offers a convenient environment where developers can work together in real-time, share code, and collaborate effectively. It eliminates the need to set up local development environments, ensuring that everyone works on the same codebase and configurations.

Component 4: Project Management:

I choose AWS Elastic Beanstalk for my Python and React application because it offers a user-friendly, managed environment with built-in auto-scaling. This simplifies our deployment process, ensures scalability, and allows us to focus on coding without the need for extensive infrastructure management. Elastic Beanstalk's support for Python and React, along with its seamless AWS integration, makes it a strong choice for the project.

Component 5: Security:

To ensure the security of my application, I would opt for AWS Elastic Beanstalk. Elastic Beanstalk simplifies the process of integrating SSL/TLS certificates for secure communication. I can configure secure connections easily, and it supports various security features like VPC integration, security groups, and IAM roles, which are essential for protecting my data and application resources. Additionally, it allows for custom security configurations, giving me the flexibility to tailor security measures to my specific needs.

Component 6: Monitoring and Analytics:

For effective monitoring, logging, and analytics, I would choose Amazon CloudWatch in combination with Python's `boto3` library. With CloudWatch, I can easily set up alarms, collect and store logs, and gain insights into my application's performance.

Using `boto3`, I can automate these tasks by writing Python scripts to interact with CloudWatch. For instance, I can create CloudWatch alarms to monitor specific metrics like CPU usage, trigger automated responses when thresholds are breached, and retrieve log data for analysis. This combination allows for proactive monitoring, immediate responses to issues, and data-driven improvements in application performance.

Reference:

AWS Official Documentation, www.google.com , www.youtube.com