3.2. Student Handout

Amazon CodeGuru Reviewer: Student Handout

Introduction to CodeGuru Reviewer

Amazon CodeGuru Reviewer is a machine learning-powered tool designed to help developers improve code quality by providing automated feedback on security, performance, and best practices. It integrates seamlessly with popular version control systems, reducing manual code review overhead.

What is CodeGuru Reviewer?

CodeGuru Reviewer automatically reviews your code and provides recommendations on:

- **Security**: Identifying potential vulnerabilities.
- **Performance**: Suggesting optimizations for better performance.
- **Best Practices**: Ensuring adherence to industry standards.

Examples:

1. **Security**: Detects hardcoded credentials in your code.
2. **Performance**: Recommends using more efficient data structures.
3. **Best Practices**: Suggests proper error handling techniques.

Supported Platforms and Setup Process

CodeGuru Reviewer supports the following platforms:

1. **AWS CodeCommit**
2. **GitHub**
3. **Bitbucket**

Setup Process:

1. AWS CodeCommit

- **Step 1**: Navigate to **CodeGuru Reviewer** in the AWS Management Console.
- **Step 2**: Choose **Associate Repository** and select **AWS CodeCommit**.
- **Step 3**: Select your repository.
- **Step 4**: CodeGuru Reviewer will review your code on new commits or pull requests.

2. GitHub

- **Step 1**: Navigate to **CodeGuru Reviewer** in the AWS Management Console.
- **Step 2**: Choose **Associate Repository** and select **GitHub**.
- **Step 3**: Authenticate with GitHub.
- **Step 4**: Select your repository.
- **Step 5**: CodeGuru Reviewer will review your code on new commits or pull requests.

3. Bitbucket

- **Step 1**: Navigate to **CodeGuru Reviewer** in the AWS Management Console.
- **Step 2**: Choose **Associate Repository** and select **Bitbucket**.
- **Step 3**: Authenticate with Bitbucket.
- **Step 4**: Select your repository.
- **Step 5**: CodeGuru Reviewer will review your code on new commits or pull requests.

How to Initiate and Manage Automated Reviews

Running Reviews on Specific Branches or Pull Requests

- **Step 1**: Create a new branch or pull request in your version control system.
- **Step 2**: Push your code to the repository.
- **Step 3**: CodeGuru Reviewer will analyze the code and provide feedback.

Examples:

1. **Branch Review**: Initiate a review on a feature branch to catch issues early.
2. **Pull Request Review**: Automatically review code changes in a pull request.
3. **Manual Trigger**: Manually trigger a review from the AWS Management Console.

Best Practices for Using CodeGuru in Development

1. **Integrate Early**: Set up CodeGuru Reviewer early in the development process.
2. **Run Reviews Frequently**: Conduct reviews on every pull request or commit.
3. **Use Recommendations Wisely**: Prioritize security and performance recommendations.

Examples:

1. **Early Integration**: Set up CodeGuru Reviewer during the initial project setup.
2. **Frequent Reviews**: Schedule reviews for every code merge.
3. **Recommendation Prioritization**: Focus on resolving security issues first.

Understanding CodeGuru Recommendations

CodeGuru Reviewer provides recommendations in three main categories:

1. **Security**: Identifies vulnerabilities like hardcoded credentials.
2. **Performance**: Suggests optimizations for better efficiency.
3. **Best Practices**: Ensures adherence to coding standards.

Examples:

1. **Security**: Alerts on improper use of encryption libraries.
2. **Performance**: Recommends reducing memory usage.
3. **Best Practices**: Suggests using appropriate data structures.

Filtering and Prioritizing Review Feedback

- **Security**: Always prioritize security recommendations.
- **Performance**: Consider performance recommendations, especially for performance-critical applications.
- **Best Practices**: Use discretion to implement best practice suggestions.

Examples:

1. **Security Prioritization**: Address SQL injection vulnerabilities immediately.
2. **Performance Consideration**: Optimize code sections causing latency.

3. **Best Practice Discretion**: Implement coding style improvements as needed.

Hands-On: Running CodeGuru Reviewer on a Real or Simulated Development Project

Step-by-Step Guide:

1. **Set Up a Repository**: Create a new repository in GitHub or AWS CodeCommit.
2. **Associate the Repository**: Link your repository with CodeGuru Reviewer.
3. **Push a Commit or Create a Pull Request**: Make changes and push them.
4. **Review CodeGuru Feedback**: Analyze the feedback provided.
5. **Implement Recommendations**: Apply the suggested improvements.

Examples:

1. **Repository Setup**: Initialize a Java project in GitHub.
2. **Feedback Review**: Examine security warnings in the AWS Management Console.
3. **Recommendation Implementation**: Refactor code based on performance suggestions.

Conclusion

Amazon CodeGuru Reviewer is a valuable tool for enhancing code quality by providing automated feedback on security, performance, and best practices. Integrating it into your development workflow can significantly reduce manual code review efforts and ensure code consistency.

Diagram: CodeGuru Reviewer Workflow

```
+-------------+   +-----------------+   +--------------+
|             |   |                 |   |              |
| Developer   |   | Version Control |   | CodeGuru     |
| Writes Code |-->| (GitHub, etc.)  |-->| Reviewer     |
|             |   |                 |   | Reviews Code |
+-------------+   +-----------------+   +--------------+
```

Thank you for participating in this session! We hope you now have a clear understanding of how to set up and use CodeGuru Reviewer in your development environment.