Pipeline Description:

This Jenkins pipeline automates the construction, testing, analysis, security scanning, and deployment of a C++ program. It is divided into seven phases, each with a distinct task and tool for carrying out the essential operation. Here is a breakdown of each stage:

1. **Build:** *Use CMake to build the C++ code.*
This step compiles and builds the C++ program with CMake. It configures the environment and prepares the code for testing and deployment.

2**. Unit and Integration tests:** *Run unit tests to check code functionality, as well as integration tests to ensure that components operate together properly.*
Description: This stage conducts automated tests with CTest which is the part of CMake to check both individual components and the application's overall integration. The results of the tests (success/failure) are emailed to the addressee, together with the appropriate logs.

3. Code Analysis *Assess code quality and conformity to industry standards.*
This stage uses CppCheck, which analyses the C++ code for possible errors, code smells, and conformance to coding standards. The program scans the code and creates a report that may be used to make future changes.

4. Security Scan: *Identify flaws in C++ code.*
This stage performs a security scan with SonarQube to identify potential vulnerabilities in the codebase. Following the scan, an email is delivered including the scan findings and logs for inspection.

5. Deploy to Staging: *Move the application to a staging server for testing using SSH to deloy to AWS EC2 instance.*
At this point, the application is deployed to a staging environment via SSH to an AWS EC2 instance. This enables for additional testing in an environment similar to production.

6. Run integration tests in the staging environment *to guarantee application functionality under production-like conditions using CTest or BoostTest.*
This stage performs extra integration tests on the staging server to ensure the application's speed and functioning. These tests serve to identify issues that may develop in a production-like setting.

7. Deploy to Production Task: *Move application to production environment using SSH into an AWS EC2 instance.*
The last stage is deploying the application to the production environment via SSH to the AWS EC2 instance. This puts the program online and available to end users.

The pipeline sends email alerts at stages (Unit and Integration Tests, Security Scan) to keep participants updated on the build's progress and outcomes. The relevant logs are attached to emails, providing build logs of pipeline's execution.