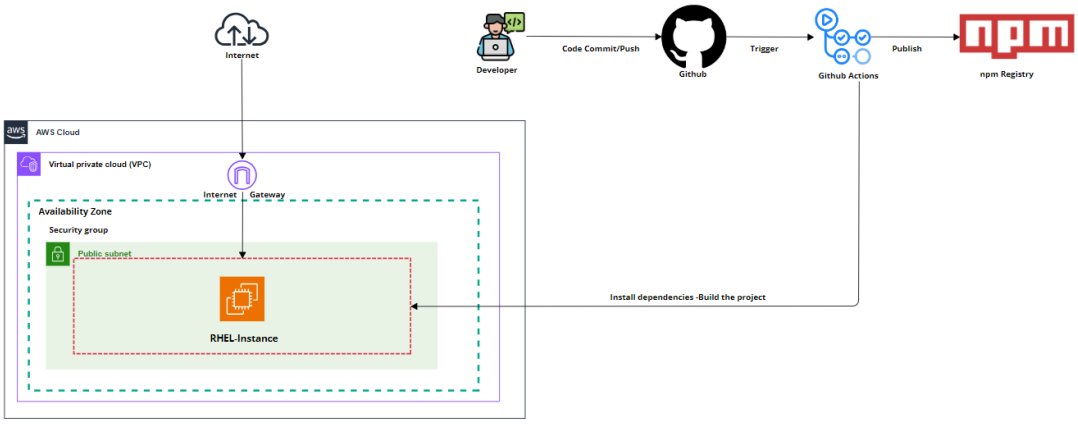
CI/CD Automation and Deployment of Internal React UI Library

1. Overview

The purpose of this document is to outline the automation strategy for developing and deploying an internal React UI library using GitHub Actions for CI/CD and a private npm registry for secure distribution. The library is built using TypeScript and is designed to provide reusable UI components for internal projects, ensuring consistency and ease of integration across various applications.

To address the challenges of synchronizing updates and avoiding issues like broken components or missing dependencies, this document provides a step-by-step guide for automating the CI/CD pipeline, version control, and library publishing. The pipeline operates on a RedHat environment using Node.js version 18.6.0.



2. Solution Design

The solution leverages the following components:

1. **Version Control**: The source code is managed on GitHub, ensuring robust version tracking and collaboration.
2. **Continuous Integration (CI)**: A CI pipeline is set up using GitHub Actions, which performs linting, testing, and building of the library to validate code changes.
3. **Continuous Delivery (CD)**: The CD pipeline automates versioning and deployment to a private npm registry, making the latest build available for internal usage.
4. **Environment Setup**: All processes run on a RedHat operating system with Node.js version 18.6.0, ensuring compatibility and stability.

3. Implementation Steps

1. **Project Setup**:

- Initialize a new GitHub repository and configure the main branch with protection rules to

prevent direct commits.
- Set up the project structure with TypeScript configurations and React components.

2. **CI/CD Pipeline Configuration**:
- Create a `.github/workflows/ci.yml' file:
  - Use Node.js 18.6.0 to set up the environment.
  - Install dependencies.
  - Build the library.
  - Trigger deployment upon successful CI checks on the main branch.
  - Automate semantic versioning using npm's versioning strategy.
  - Publish the package to a private npm registry.

3. **Registry and Deployment**:
- Configure npm to use a private registry.
- Authenticate with npm using environment secrets in GitHub Actions for secure deployment.

4. **Documentation and Diagram**:
- Include a README file with setup instructions, CI/CD workflows, and diagrams illustrating the pipeline architecture.

4. Conclusion

This documentation outlines the automated approach to building and deploying the React UI library using GitHub Actions and a private npm registry. The CI/CD pipeline ensures that updates are consistently verified and made available internally, reducing integration overhead and minimizing errors.