**Project: Streamlining DevOps Pipelines: GitHub, Jenkins, and AWS CloudFormation Template**

**Introduction:**

In the modern software development, efficient workflows and automated processes are essential for delivering high-quality applications. This project focuses on implementing a robust CI/CD (Continuous Integration and Continuous Deployment) pipeline using **AWS CloudFormation**, **GitHub**, and **Jenkins**.

**Key Accomplishments:**

* The project begins with the creation of an **EC2 instance with an Elastic IP (EIP)** using CloudFormation, showcasing the power of Infrastructure as Code (IaC).
* Next, a **GitHub repository** is configured to store pipeline code, with a **webhook** set up to trigger a Jenkins pipeline automatically upon code changes.
* The Jenkins pipeline ensures seamless integration and deployment, while an email notification system provides instant feedback to stakeholders on build status.

**Key Features**

**Automated Infrastructure Setup**

* Leveraging AWS CloudFormation to automate the creation of an EC2 instance and allocate an Elastic IP, ensuring consistent and repeatable infrastructure deployment.

**Version Control with GitHub**

* Integration with GitHub for managing source code and pipeline configurations, providing version control and collaboration capabilities.

**CI/CD Pipeline Setup**

* Configuring a Jenkins pipeline to automate the build process, triggered by webhook events from the GitHub repository.

**Webhook Integration**

* Establishing a GitHub webhook to seamlessly connect the repository with Jenkins, enabling real-time pipeline execution upon code changes.

**Automated Notifications**

* Implementing an email notification system in Jenkins to inform stakeholders about the build status, enhancing transparency and feedback loops.

**End-to-End Automation**

* Achieving a streamlined workflow from infrastructure creation to pipeline execution and notification, demonstrating the principles of automation and DevOps.

**Scalability and Flexibility**

* Using modular components like CloudFormation templates and Jenkins pipelines, making the solution scalable for future requirements.

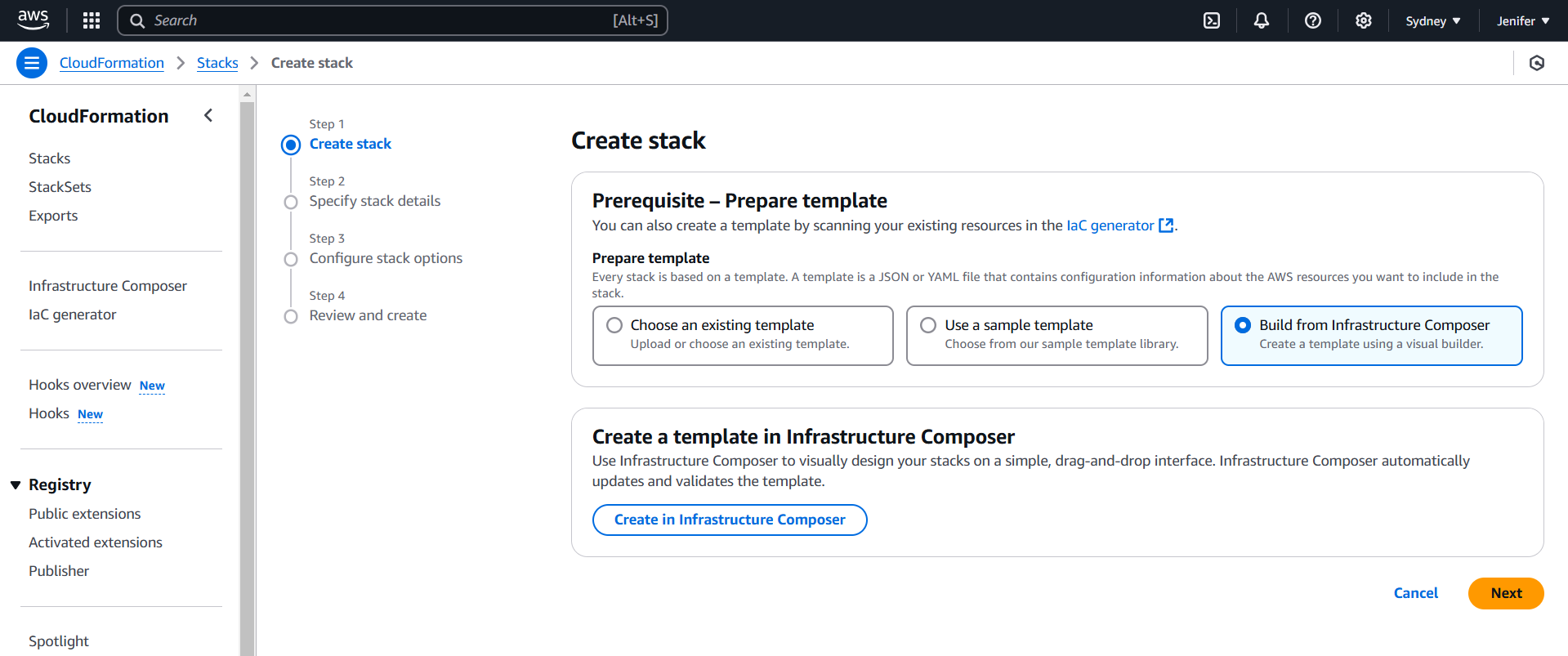
**Real-World Relevance**

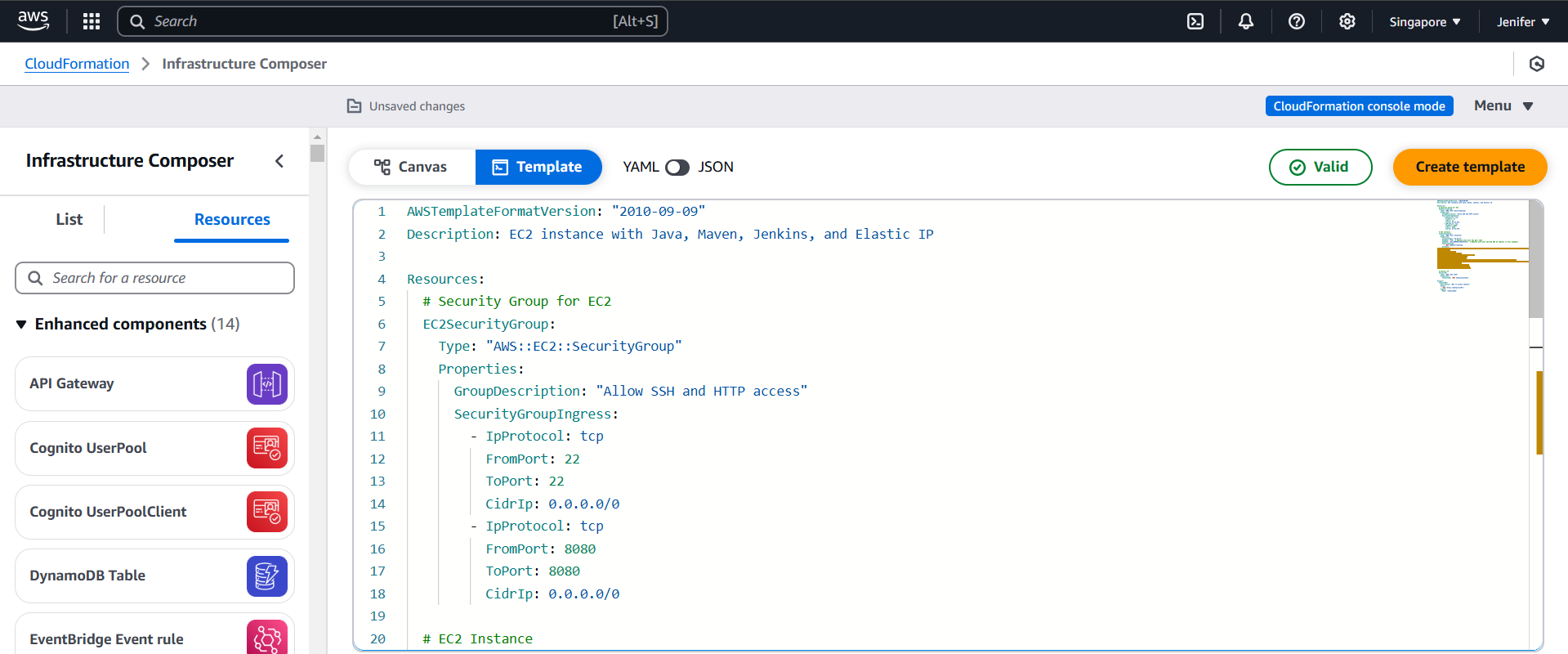
* Addressing practical scenarios in software development pipelines, emphasizing the importance of automation and integration in DevOps practices.

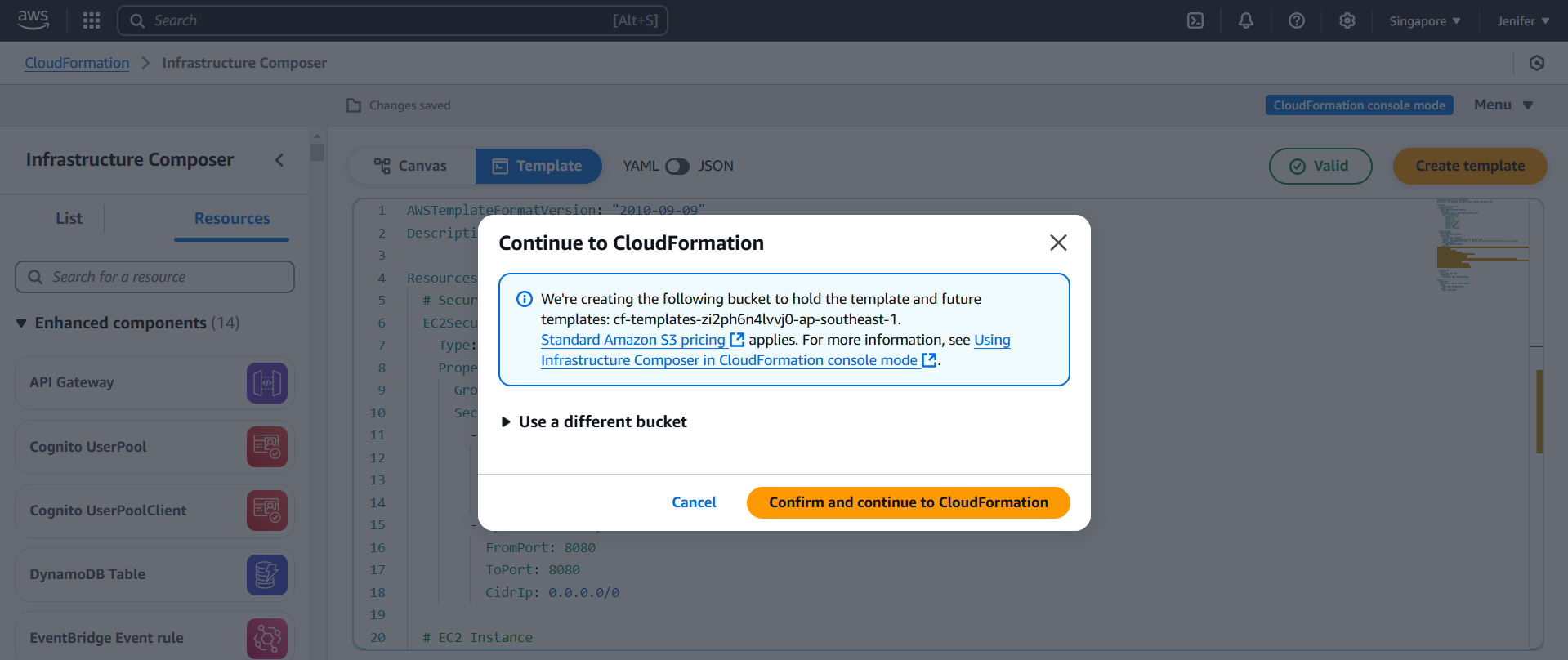
**Infrastructure Creation Using AWS CFT:**

* **AWS CloudFormation** is an Infrastructure as Code (IaC) service that simplifies the process of setting up AWS resources by using templates written in JSON or YAML.
* It enables you to automate, standardize, and manage infrastructure deployments consistently across multiple environments.
* In this project, AWS CloudFormation is used to create an **EC2 instance** and assign an **Elastic IP**, ensuring repeatability and consistency in infrastructure provisioning.

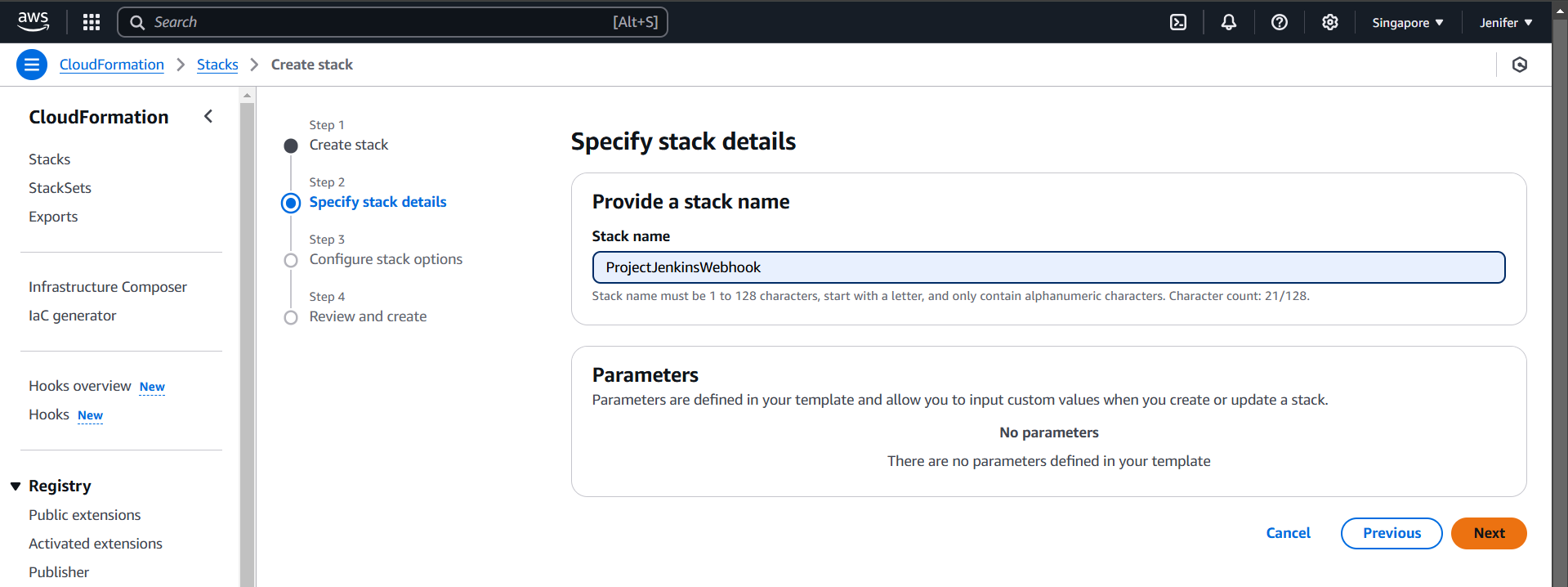
1. Started Creating a stack for New Template



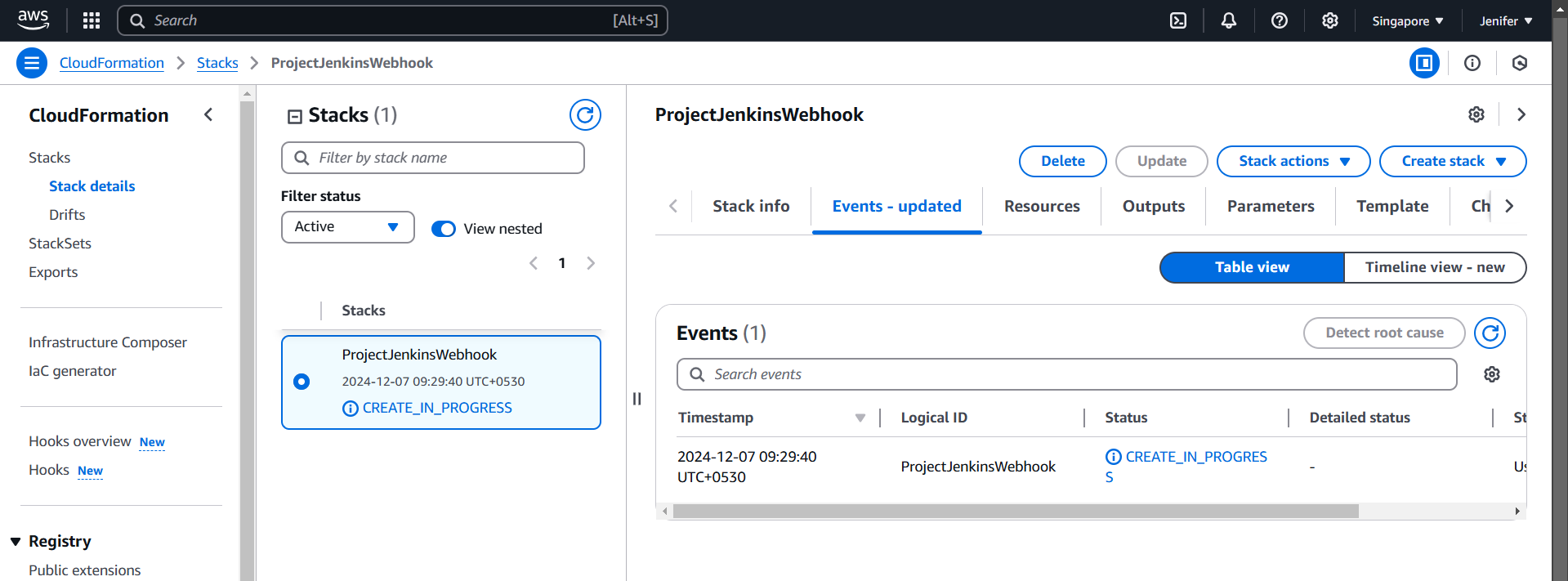




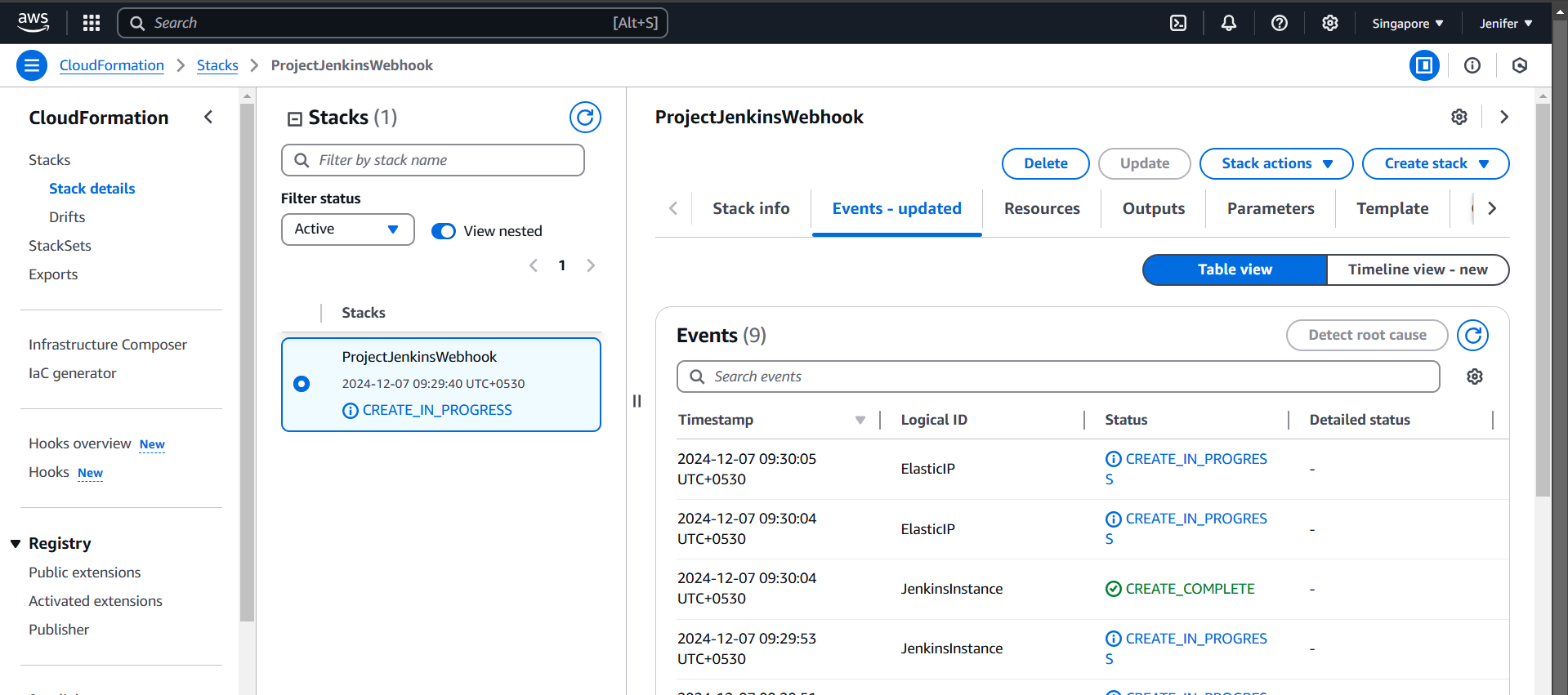
1. Providing Stack Details



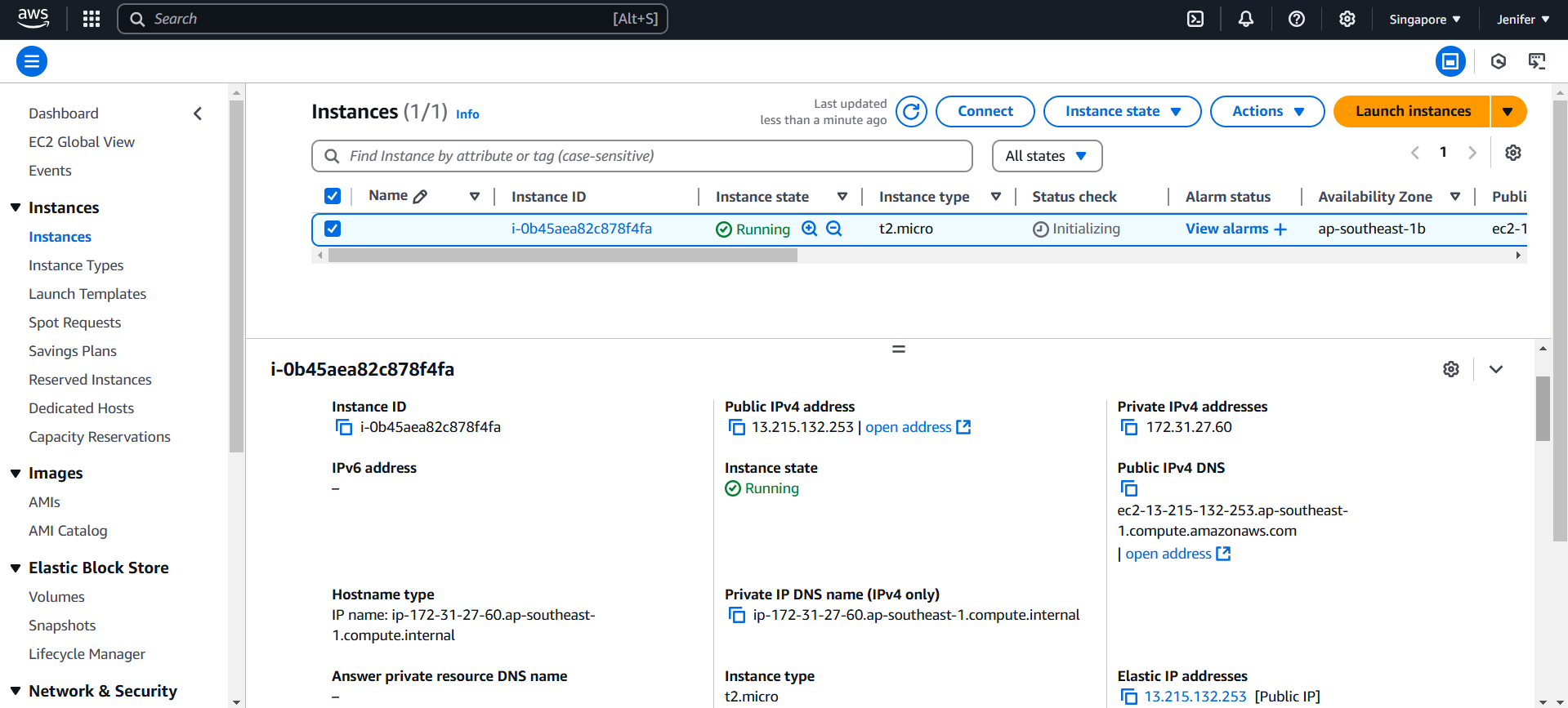
1. The Template Creation is in Progress



1. The Template got created successfully

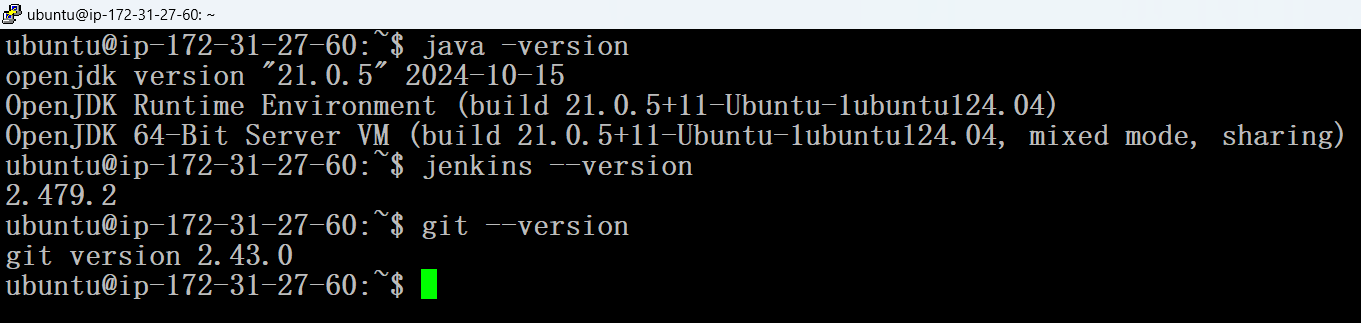


1. An Ec2 Instance got created with allocating EIP

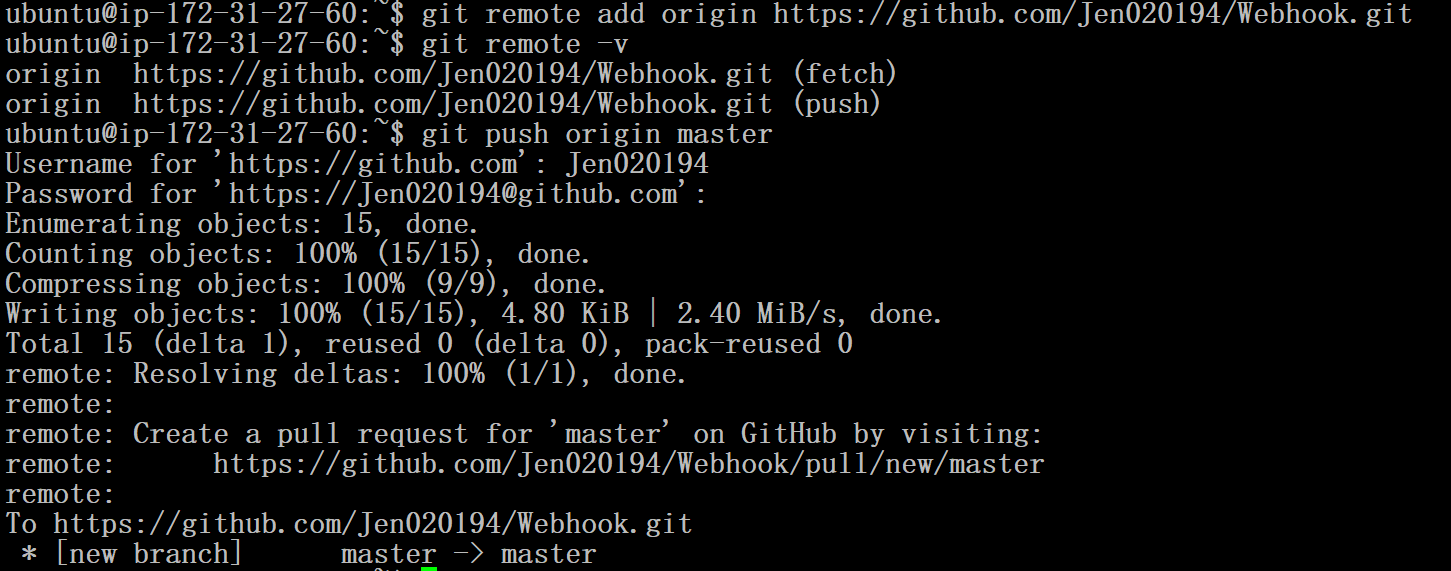


**EC2 Instance Configuration**

1. Git, Jenkins has installed

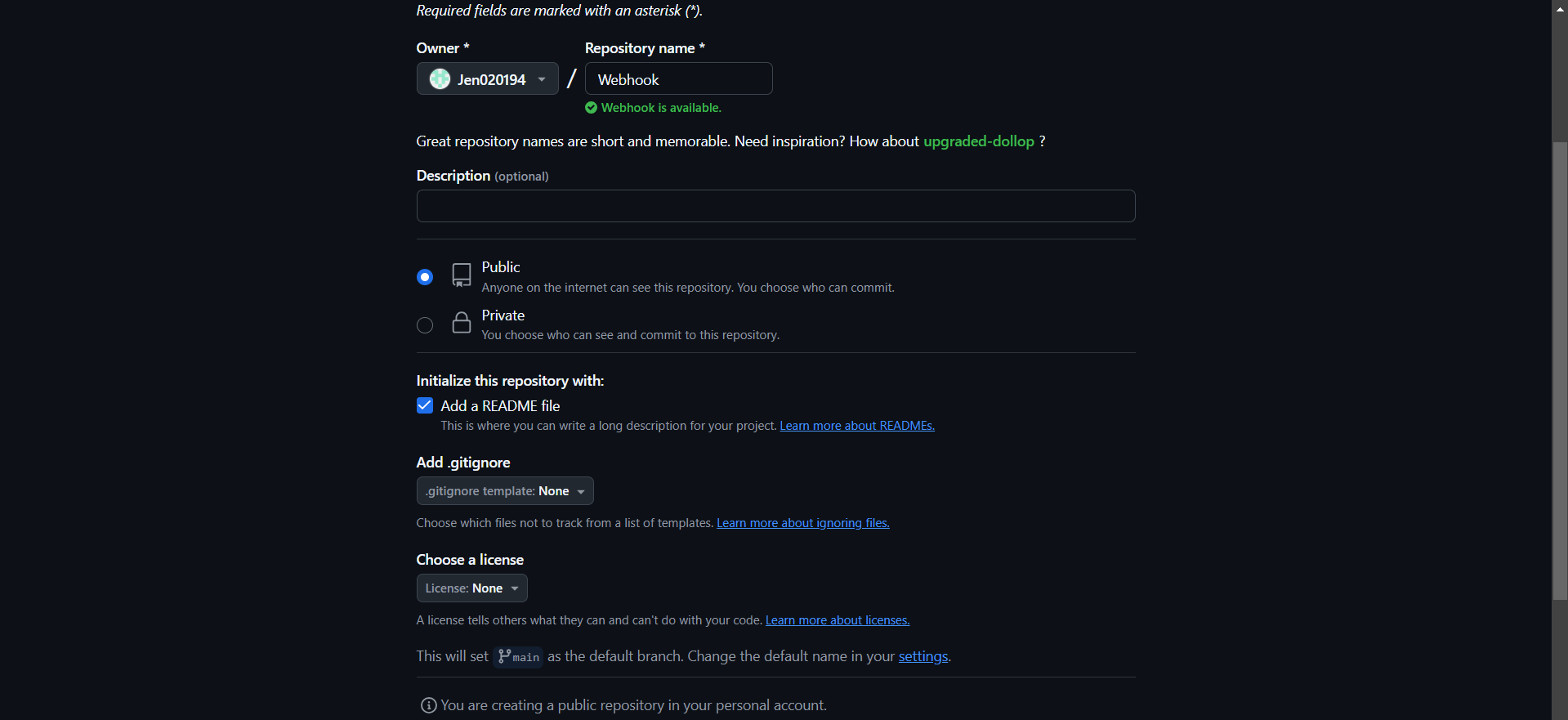


2. A Jenkins pipeline file has created and successfully pushed into the remote repository

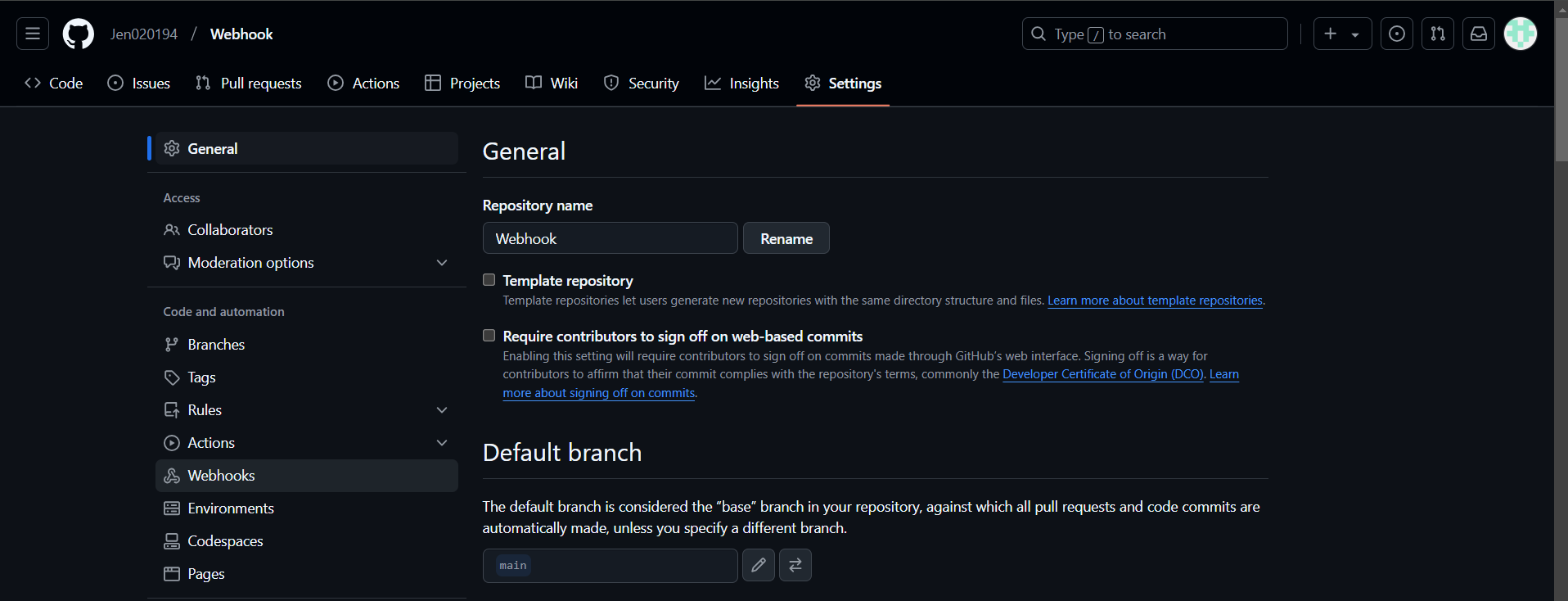


**Github Repository creation and Webhook Configuration to establishing the communication between Github & Jenkins**

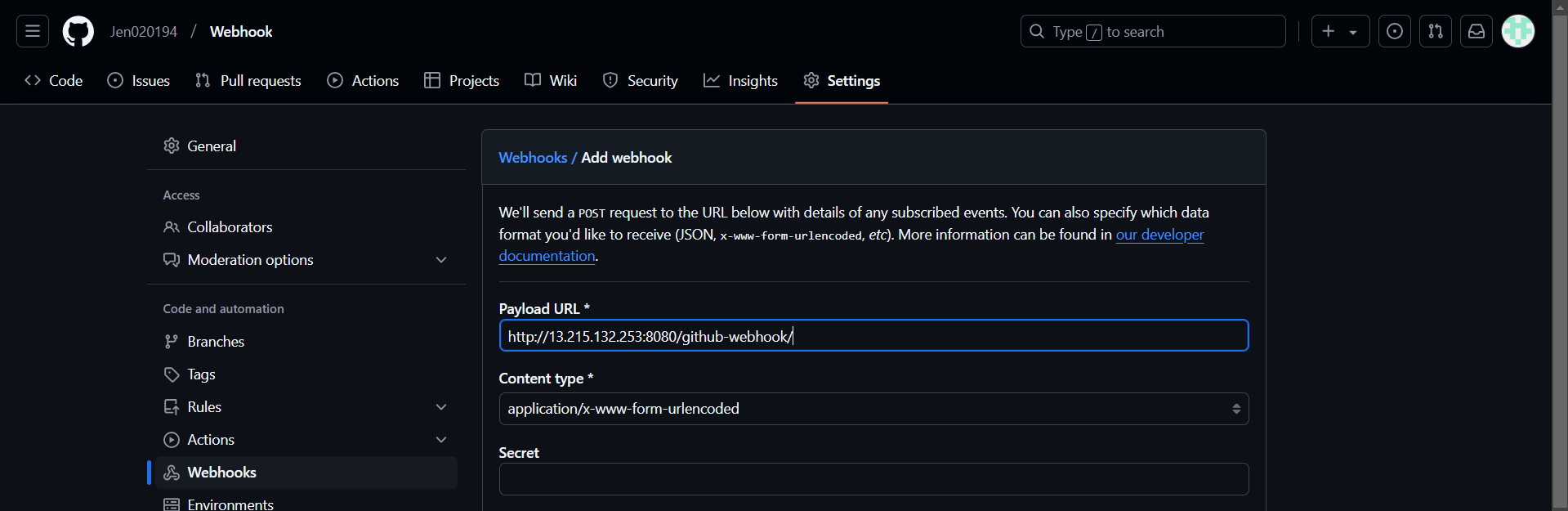
1. A Github Repository has created with a readme file



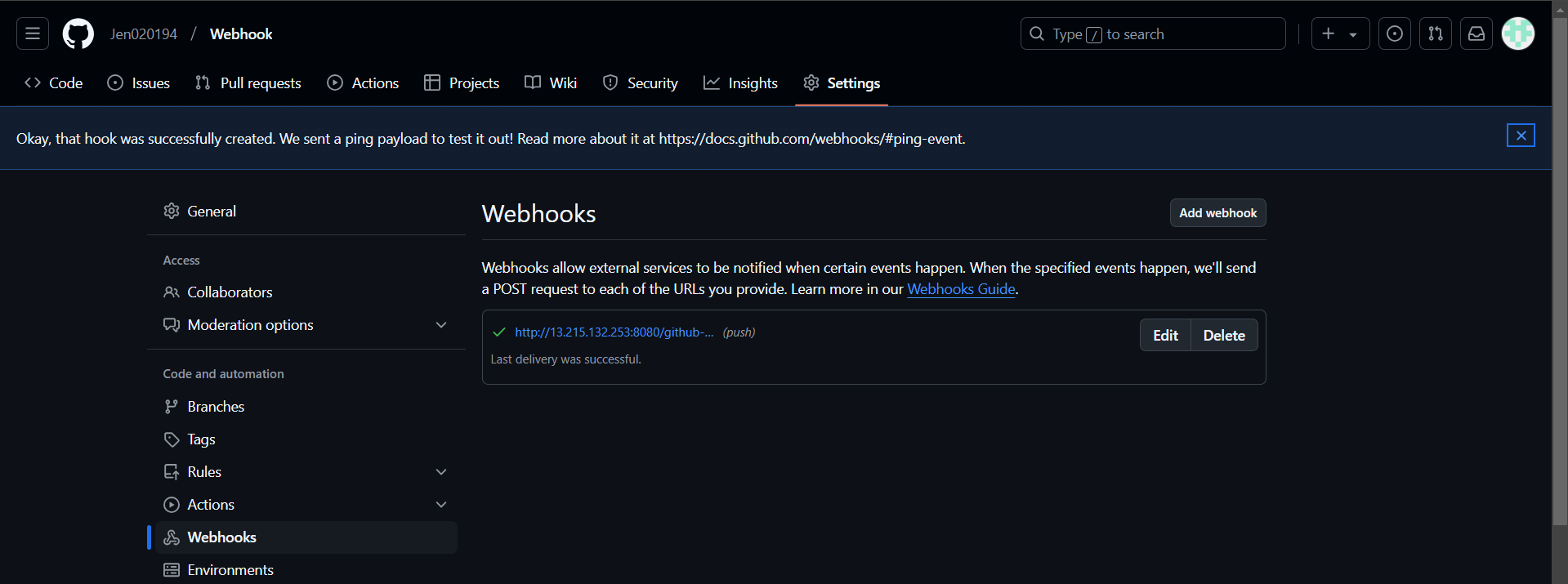
2. Webhook configuration has started for the Repository



3. Provided Payload URL with the appropriate EIP

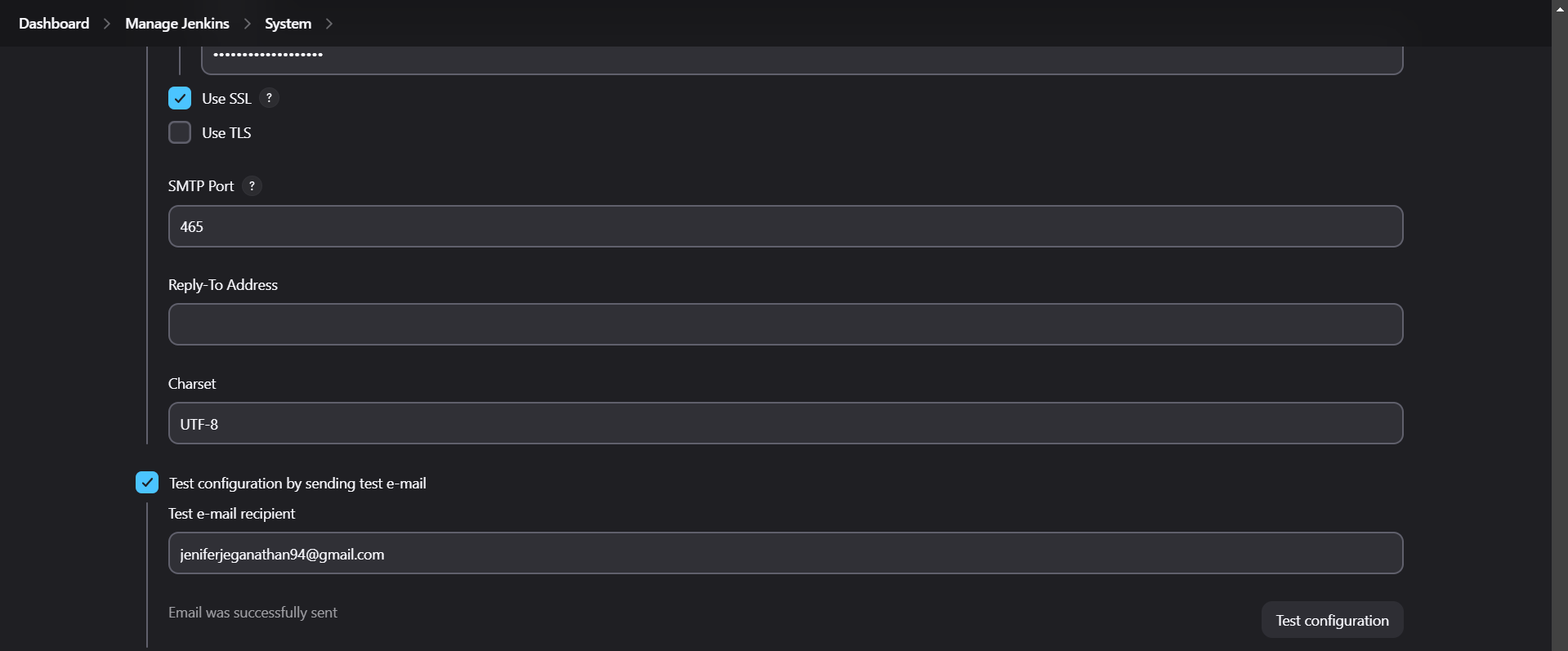


4. Webhook has created successfully

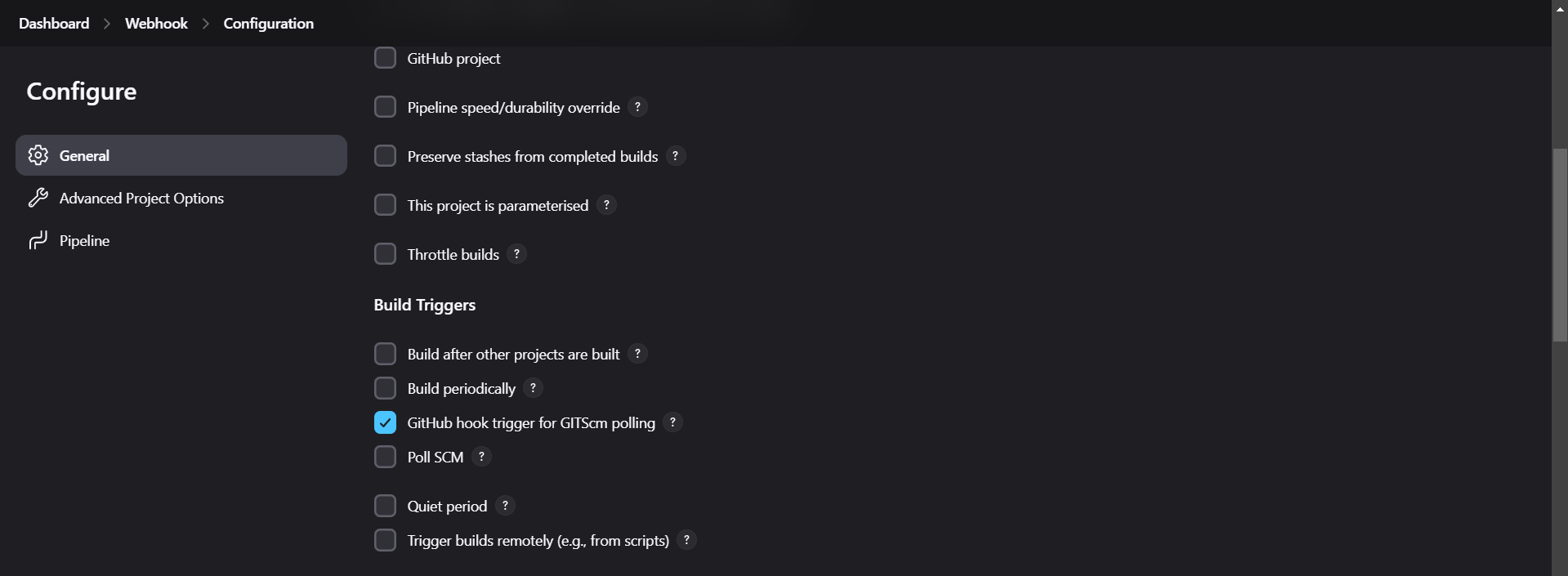


**Jenkins Configurations for the seamless CI/CD flow**

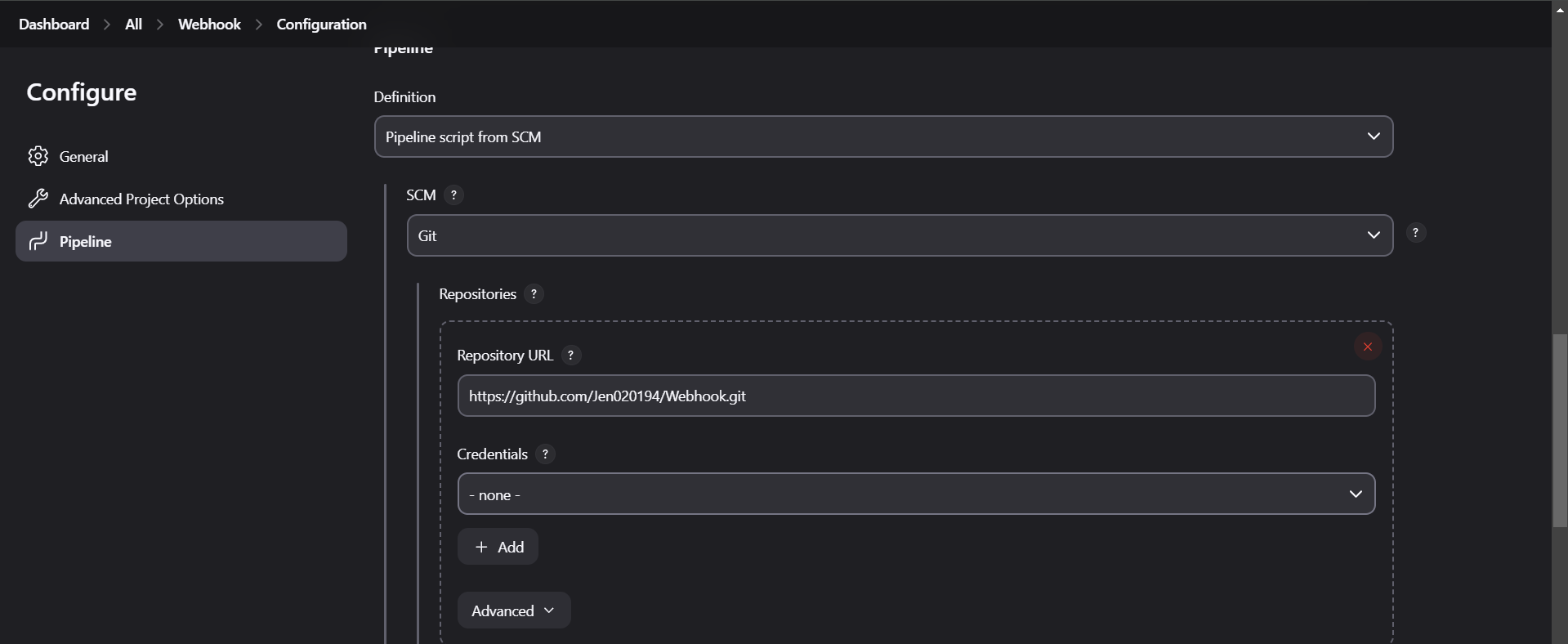
1. E-mail Configuration has done

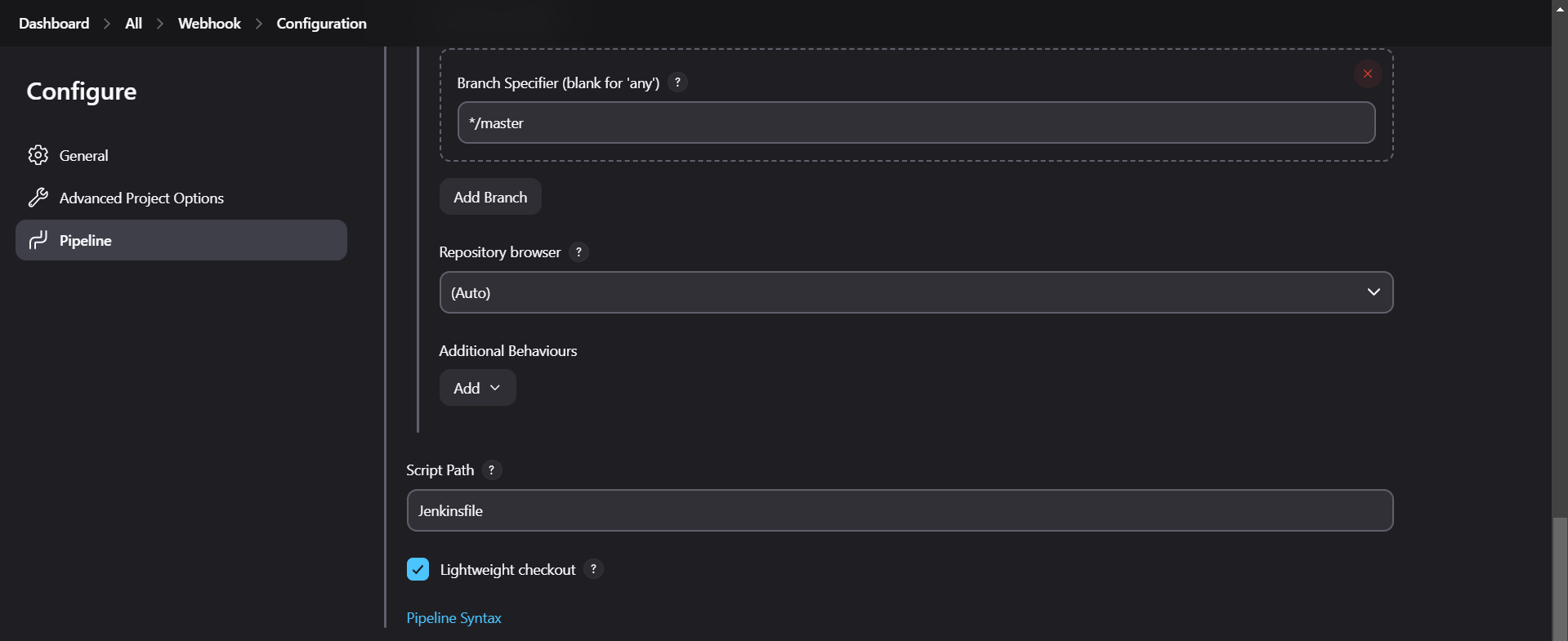


2. A Pipeline Project Configured with GitHub hook Trigger for GitScm Polling



3. The Pipeline Configured with providing the GitHub Repo Link and appropriate Branch

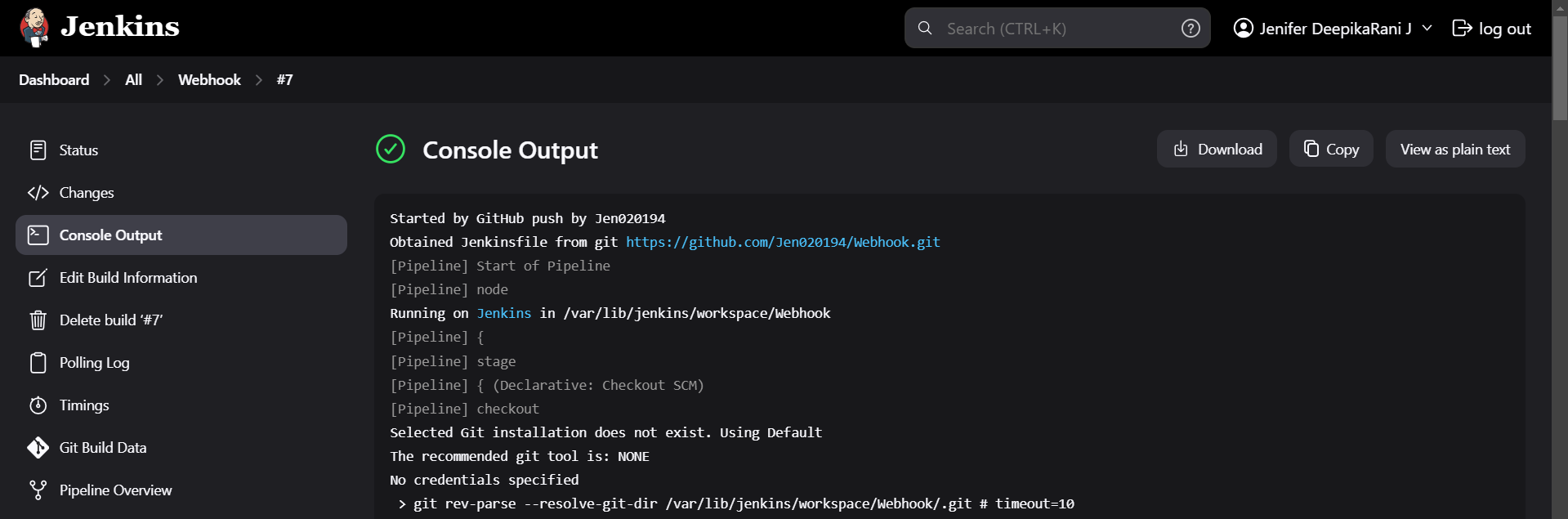




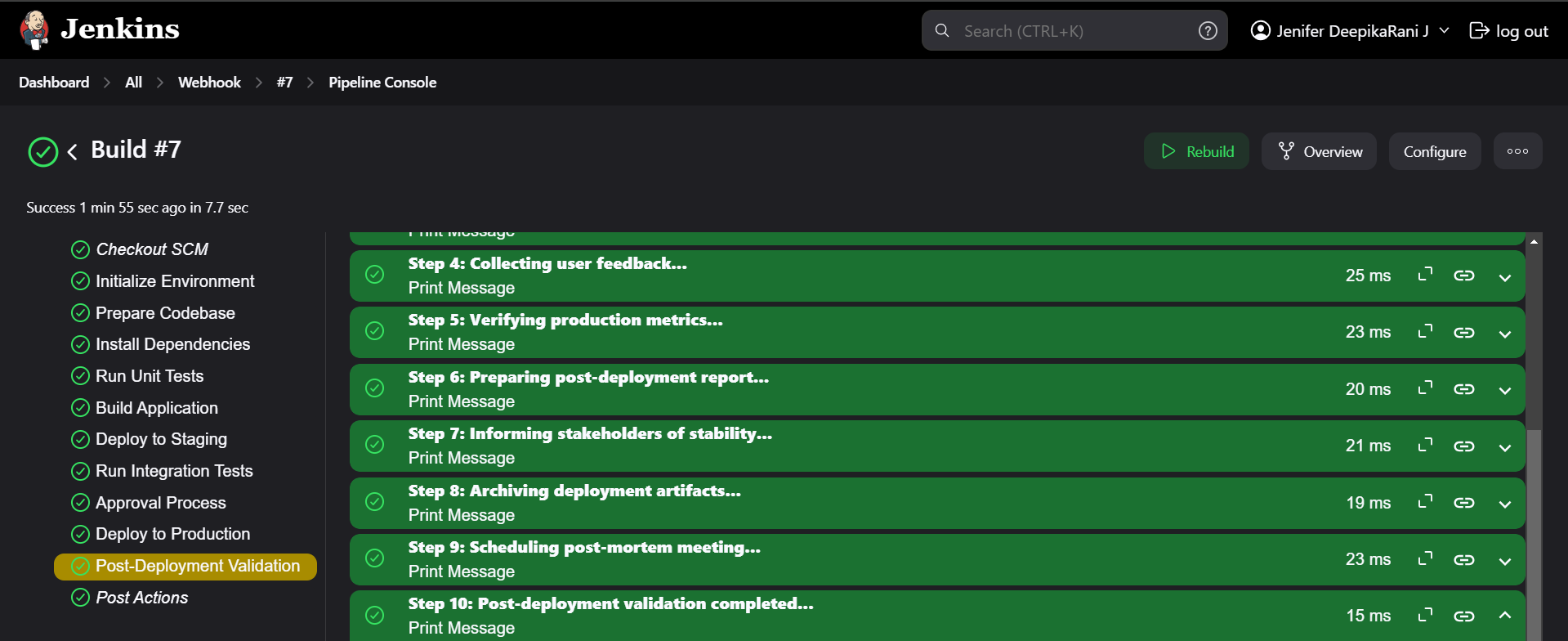
Once the job configuration has done and it is automatically triggered by the webhook

**The Outputs:**

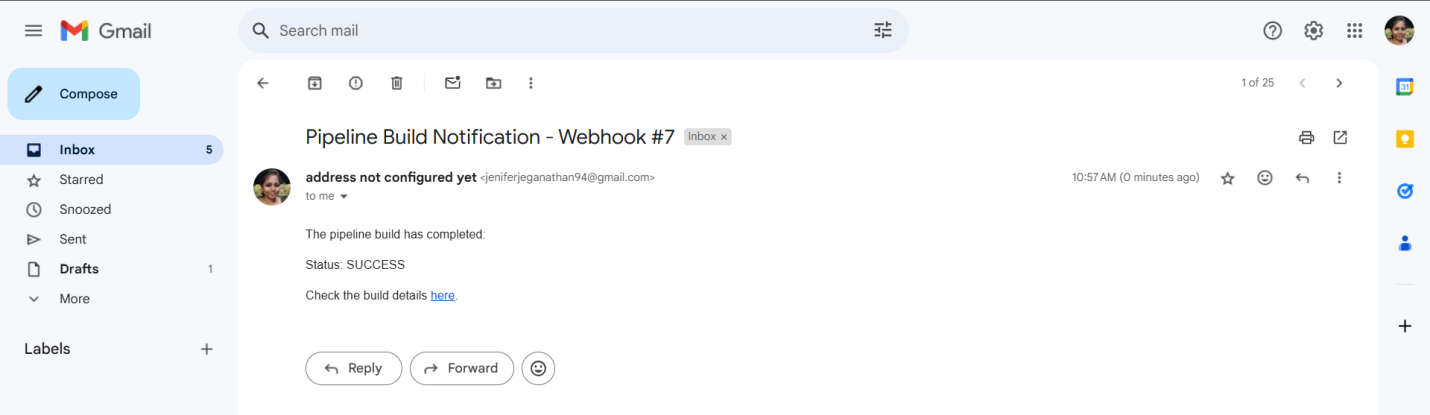
1. The Console Output



2. The Pipeline Overview



3. Finally, The mail has sent to the given recipient mail id successfully



**Conclusion:**

This project demonstrates the synergy between version control (GitHub), CI/CD tools (Jenkins), and cloud services (AWS), offering an end-to-end solution for automating development workflows.