**Integrating GitHub with Jenkins using GitHub Webhooks**

**Reasons for choosing Jenkins:**

Jenkins provides flexibility in creating pipelines. It supports both declarative and scripted pipeline syntax, enabling users to define pipelines as code. It was also an open source tool. Jenkins has a vast community support so users can interact with forums in-case of an issue. It was also easy to install and configure.

**Steps:**

1. Install Jenkins:

* Ensure that Jenkins is up and running.

1. Install GitHub Plugin:

* Install the GitHub plugin in Jenkins. You can do this by navigating to "Manage Jenkins" > "Manage Plugins" > "Available" and then searching for and installing the "GitHub" plugin.

1. Create a GitHub Repository:

* Have a GitHub repository where your code is hosted.

1. Create a Jenkins Pipeline:

* In your Jenkins dashboard, click on "New Item" to create a new job.
* Enter a name for your job, select "Pipeline," and click on "OK."
* In the pipeline configuration, choose "Pipeline script from SCM" as the definition.
* Select Git as the SCM and provide the Repository URL.
* Define your pipeline stages, steps, and configurations using Groovy syntax.

1. Set Up Webhook in GitHub:

* In your GitHub repository, go to "Settings" > "Webhooks" > "Add webhook."
* Set the Payload URL to your Jenkins server's URL followed by /github-webhook/ (e.g., http://your-jenkins-server/github-webhook/).
* Choose the events you want Jenkins to trigger on (e.g., "Push events").
* Ensure the webhook is active and save it.

1. Configure GitHub Webhook in Jenkins:

* In your Jenkins job configuration, under the "Build Triggers" section, check the option for "GitHub hook trigger for GITScm polling."

1. Configure GitHub Credentials:

* In your Jenkins job configuration, under the "Build Environment" section, choose "Use secret text(s) or file(s)" for the "Credentials" dropdown, and add your GitHub credentials.