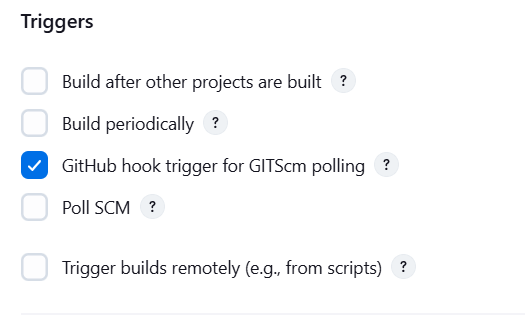
**Introduction:**

This guide explores configuring and managing CI/CD pipelines using Jenkins and Docker. It covers key components like triggers, pipelines, Dockerfiles, and deployment scripts to help enhance your workflow and optimize project implementation.

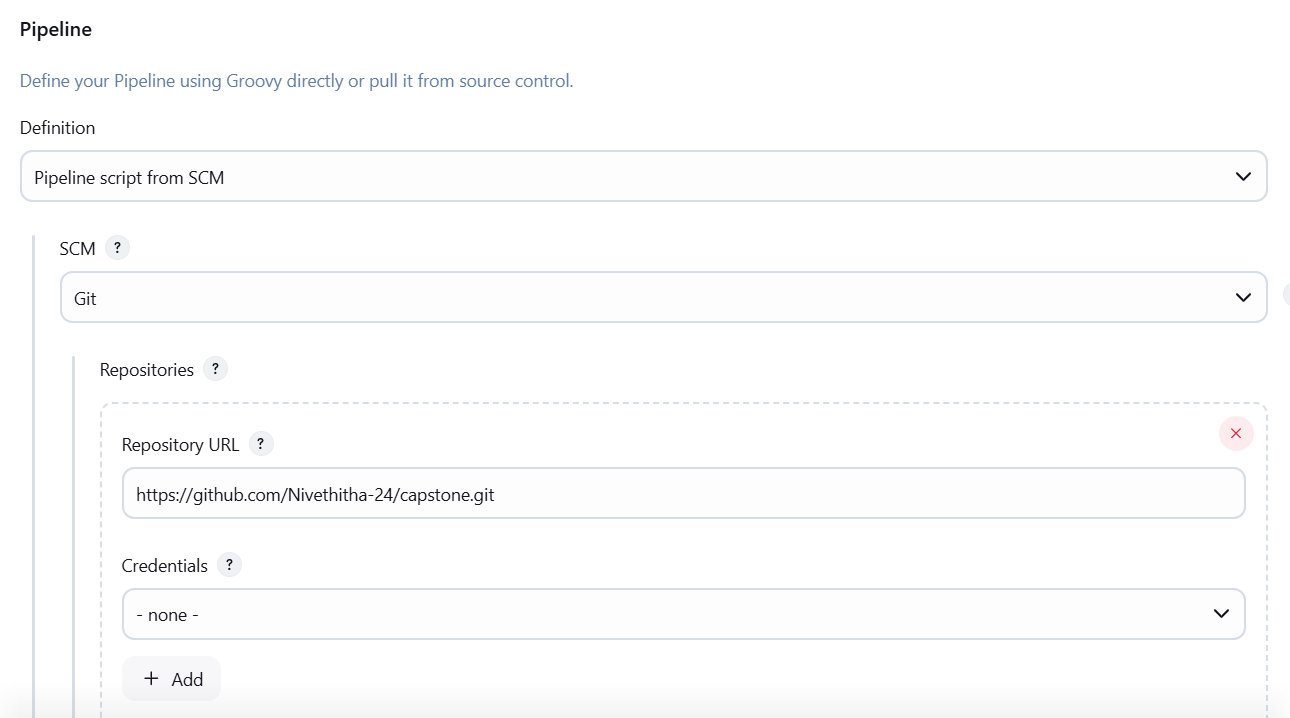
**Triggers**:

**GitHub hook trigger for GITScm polling**: This option triggers a build when a commit is pushed to GitHub.

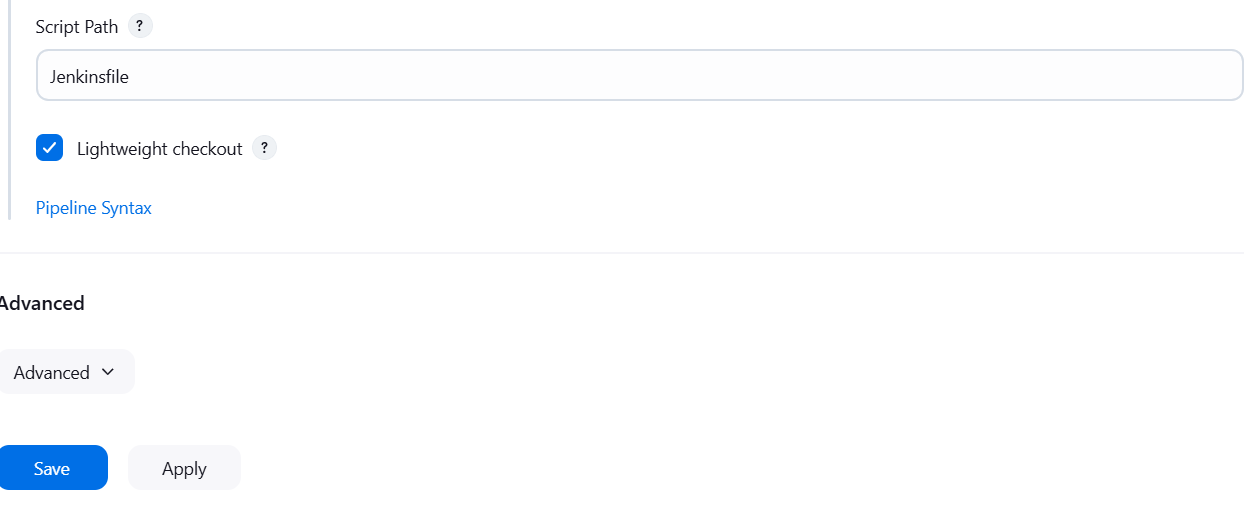


**Pipeline**:

* Defines the pipeline using Groovy or by pulling it from source control.
* **Definition**: Uses a Pipeline script from SCM (Source Control Management).
* **SCM**: Specifies Git as the source code repository.
* **Repository URL**: Points to your GitHub repository.
* **Branches to build**: Targets the 'main' branch for building.
* **Script Path**: Points to the Jenkinsfile located in the root directory of the repository.
* **Additional Behaviours**: Settings related to repository handling and build process.

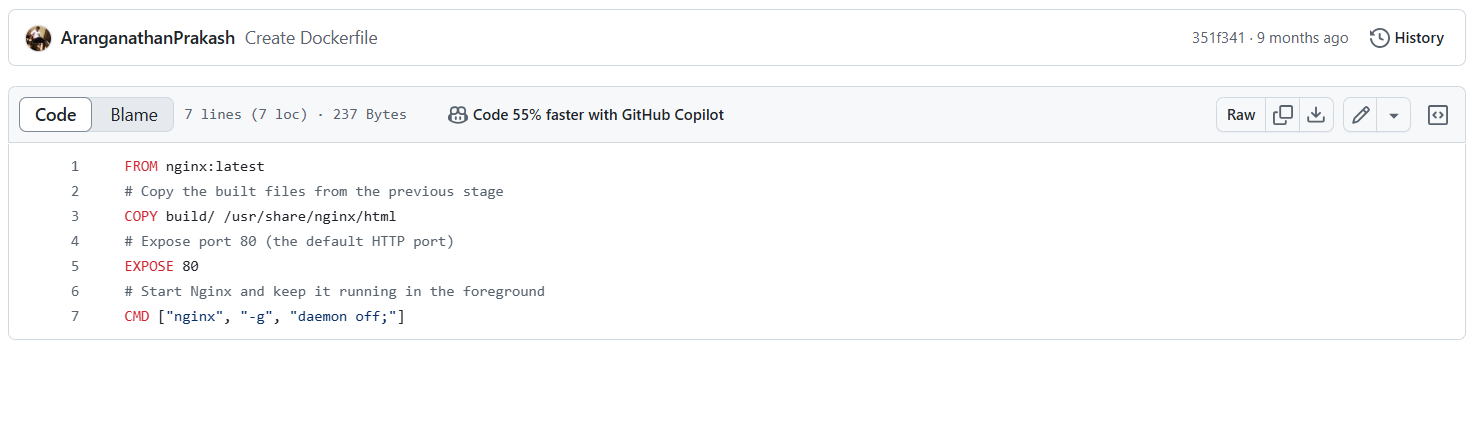






**Dockerfile**:

* Defines the Docker container configuration for Nginx.
* Steps include copying built files, exposing port 80, and running Nginx in the foreground.

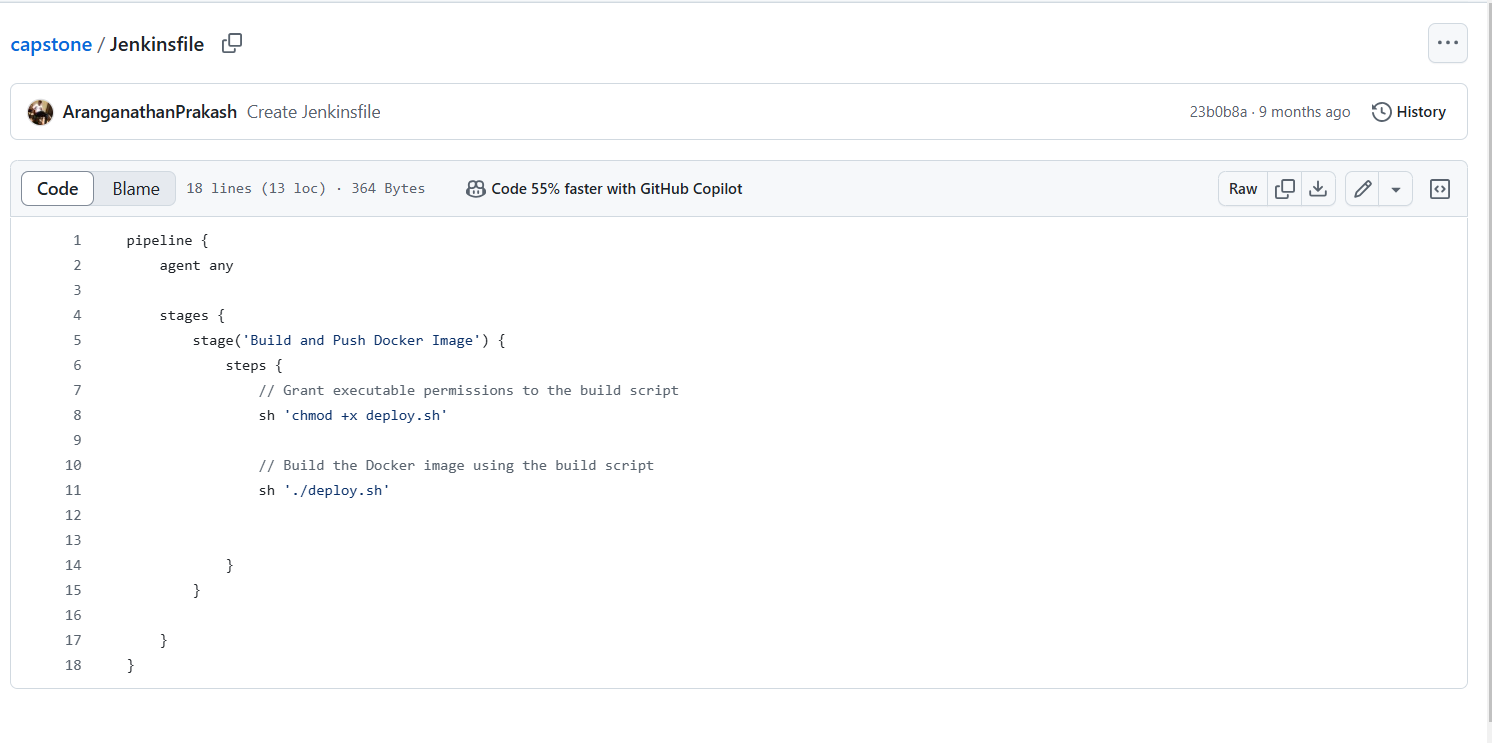


* 



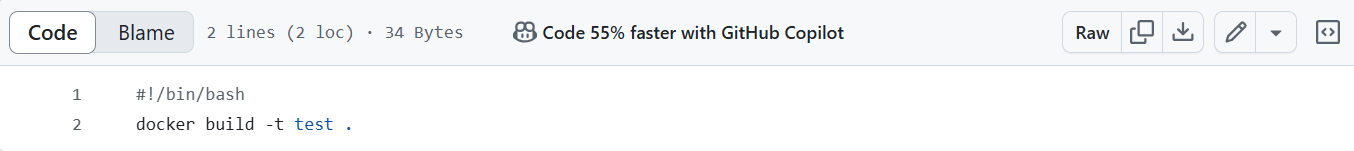
**Jenkinsfile**:

* Defines the Jenkins pipeline stages.
* **Build and Push Docker Image** stage: Grants permissions, builds the Docker image, and deploys it.
* Contains script commands for building and pushing the Docker image.



**Build.sh**:

* Script to build and run the Docker container locally.
* Builds the Docker image and runs the container.



**Deploy.sh**:

* Script to automate deployment tasks.
* Includes login, tagging, and pushing the Docker image to a repository.



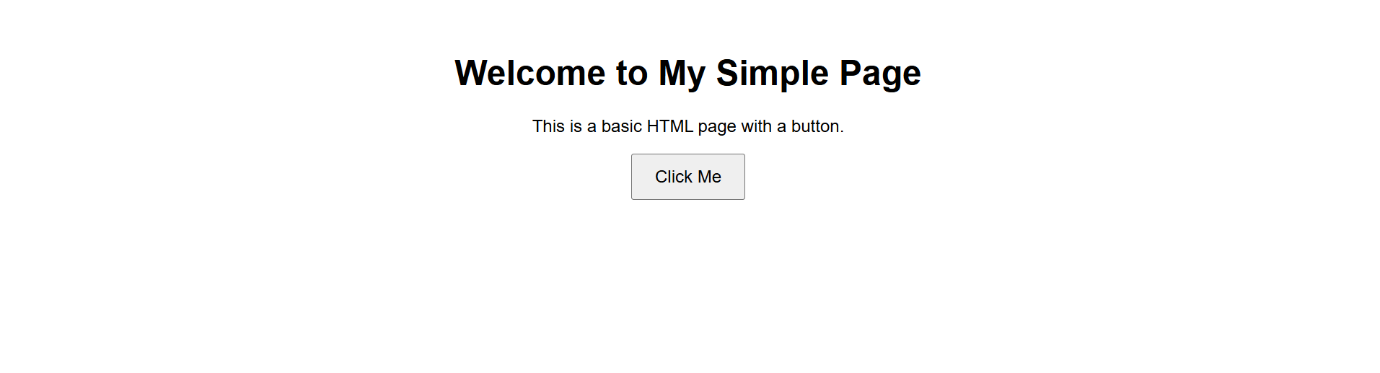
**Command:** sudo su

* sudo: Stands for "superuser do," and allows a permitted user to execute a command as the superuser (root).
* su: Stands for "substitute user," and switches the current user to another user. By default, it's the root user.
* When you combine them (sudo su), it means you're switching to the root user with superuser privileges.

**Command:** chmod 777 /var/run/docker.sock

* chmod: Stands for "change mode," and it's used to change the permissions of a file or directory.
* 777: Sets the permissions to rwxrwxrwx, which means read, write, and execute permissions for the owner, group, and others.
* /var/run/docker.sock: This is the file for Docker socket. Changing its permissions can allow any user to access the Docker daemon, which is typically not recommended for security reasons.

**OUTPUT**



**Conclusion:**

By understanding and using CI/CD pipelines with Jenkins and Docker, you can revolutionize your development workflow. This guide has equipped you with the knowledge to implement efficient CI/CD strategies, leading to higher productivity and better-quality software.