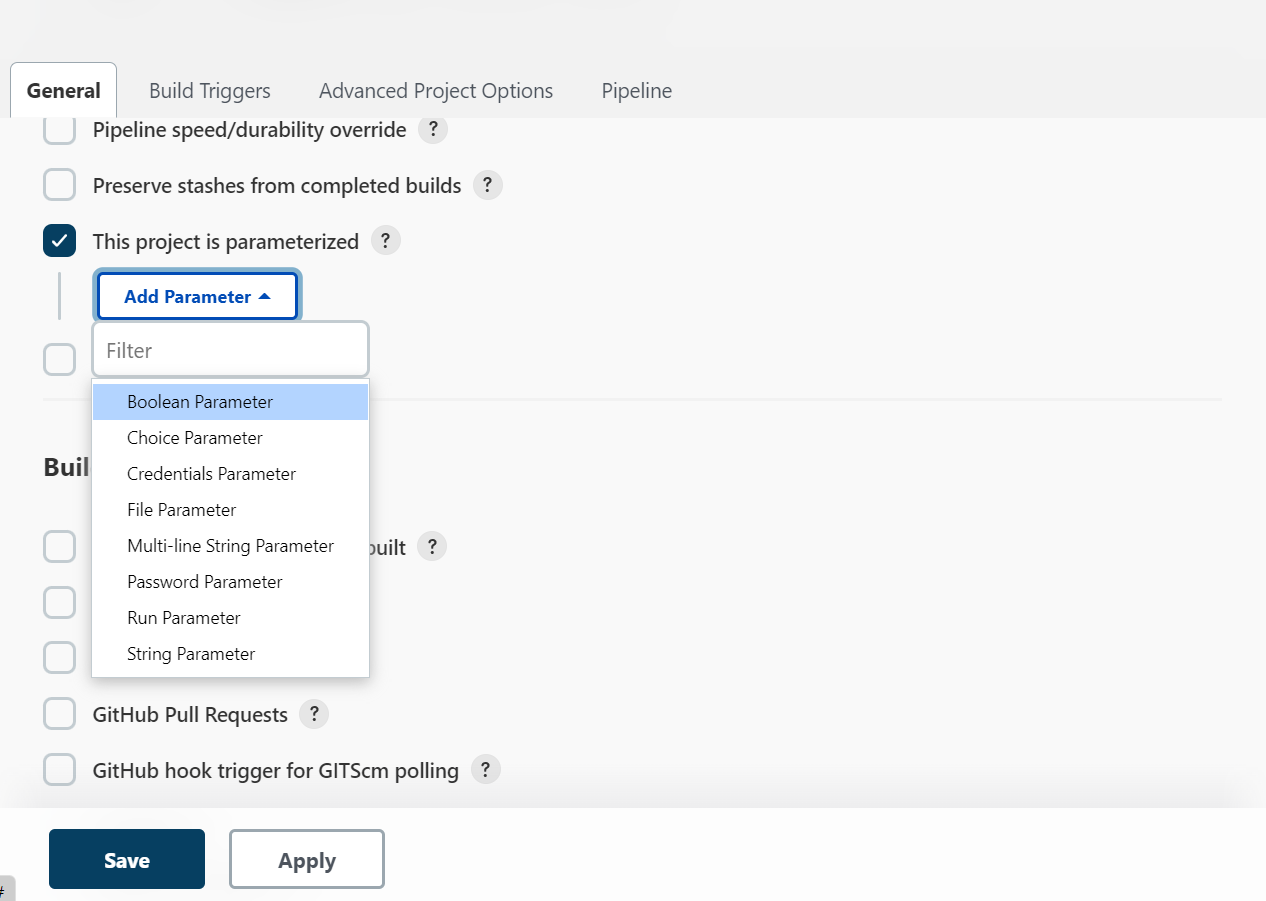
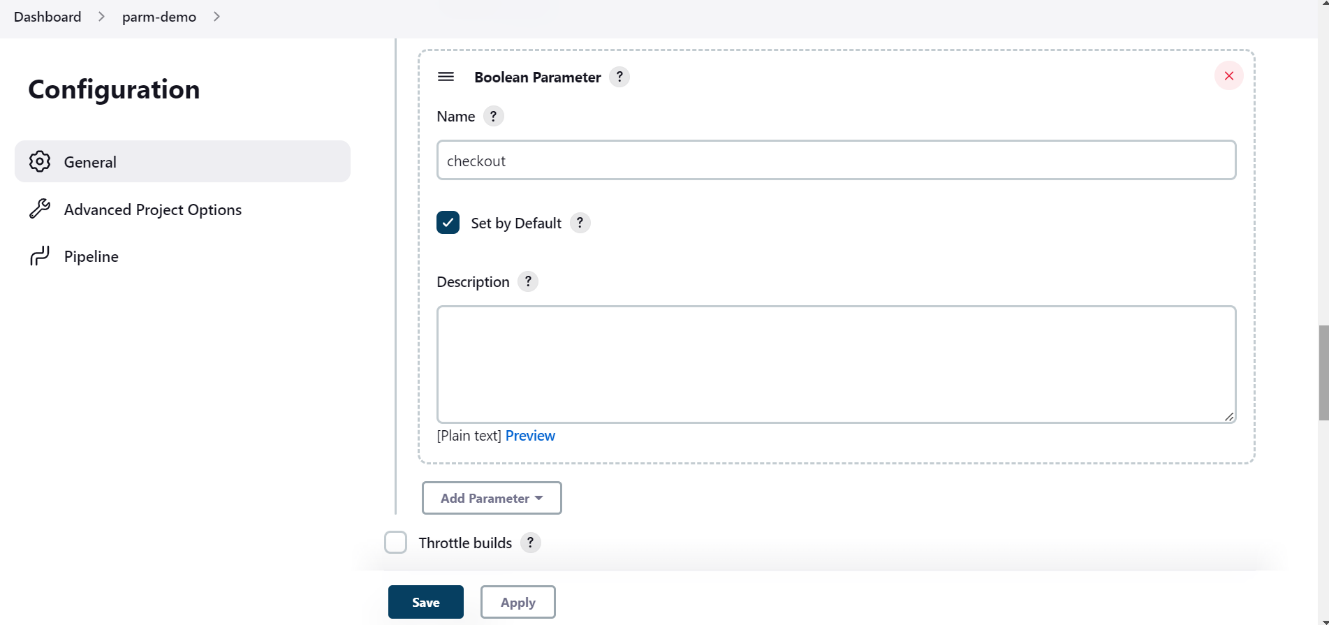
**MODULAR PIPELINE**

**CONFIGURING PARAMETER**

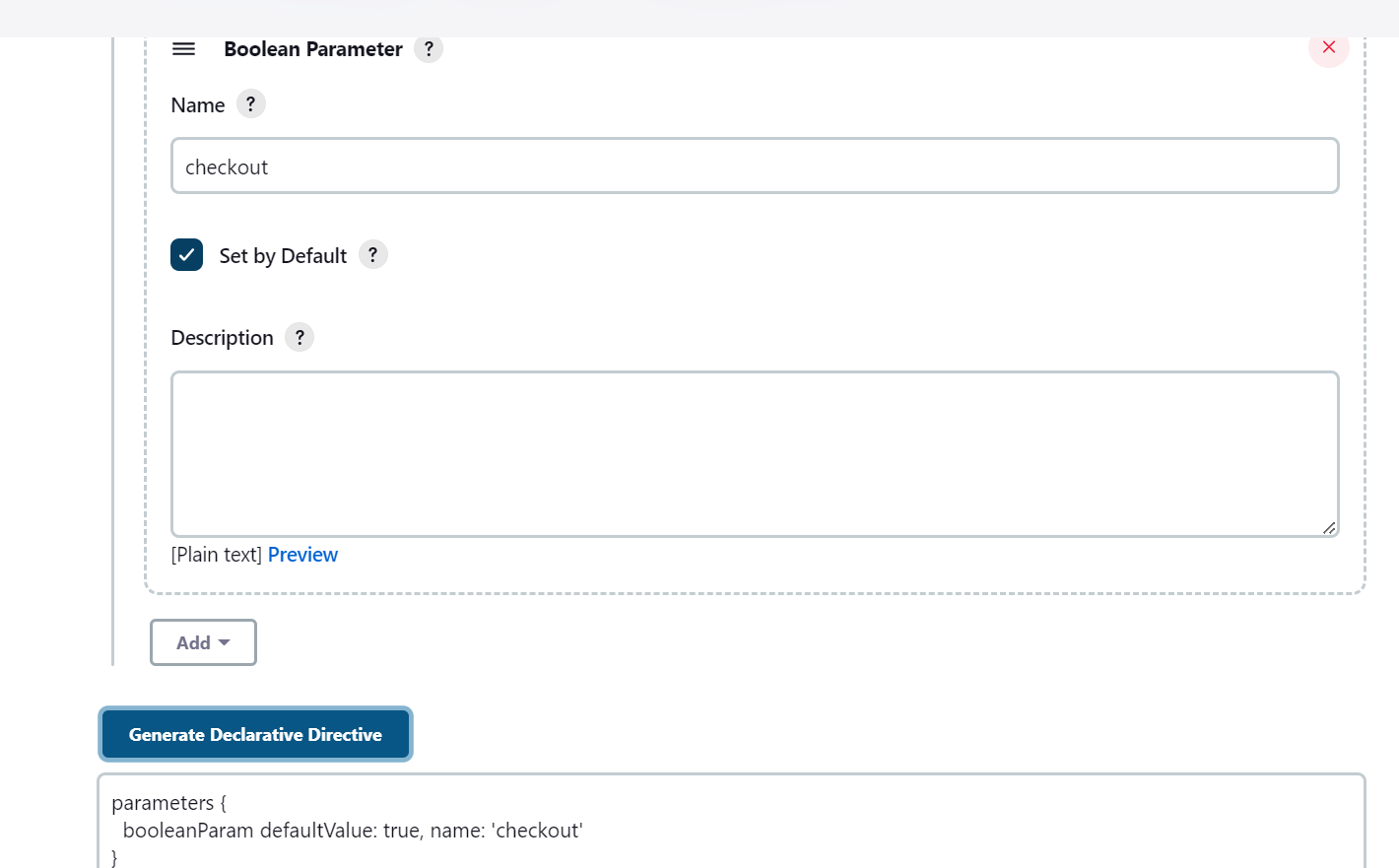
1. By configuring the parameter section in the AOB pipeline we can make the pipeline run only in a specified stage using the Boolean parameter.
2. Inside project configuration settings enable the ‘This project is parameterized’ option and select Boolean Parameter.



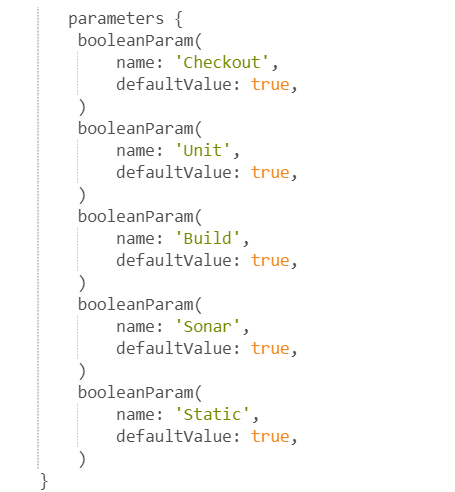
1. Add all the stages in Boolean Parameter by clicking the ‘Add Parameter’ option and also select ‘set by default’ option.



1. Inside the pipeline script add parameter syntax for all stages. We can also generate pipeline syntax. Go to Pipeline syntax 🡪 Declarative Directive Generator 🡪 parameter.parameter 🡪 Add 🡪 Boolean Parameter. Specify the stage and hit on Generate Declarative Directive.



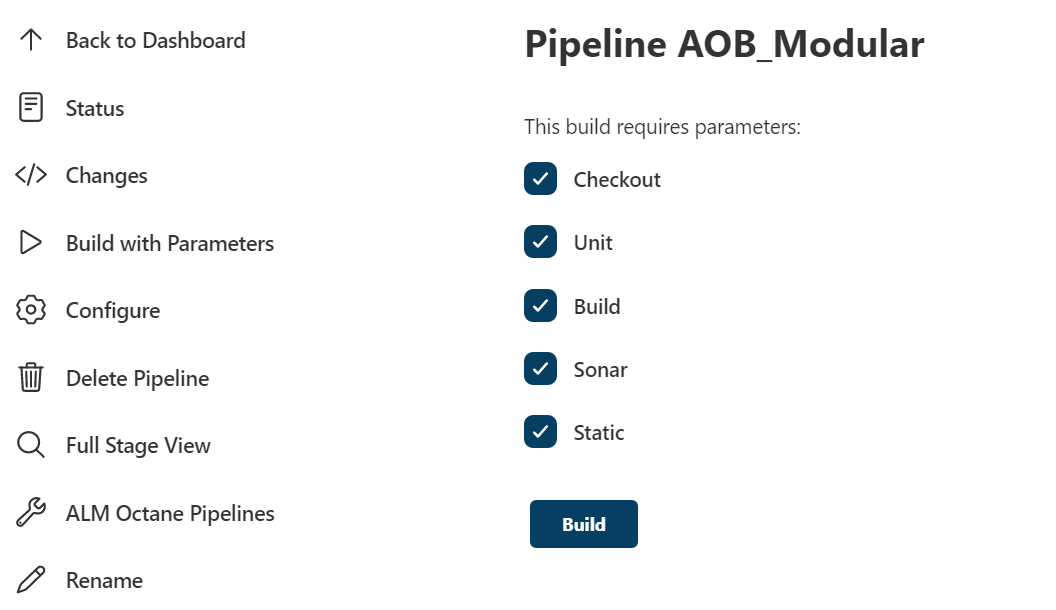
1. Repeat the same process for the rest of the stages by selecting “Add Paramter”.
2. Add all the commands in the pipeline.



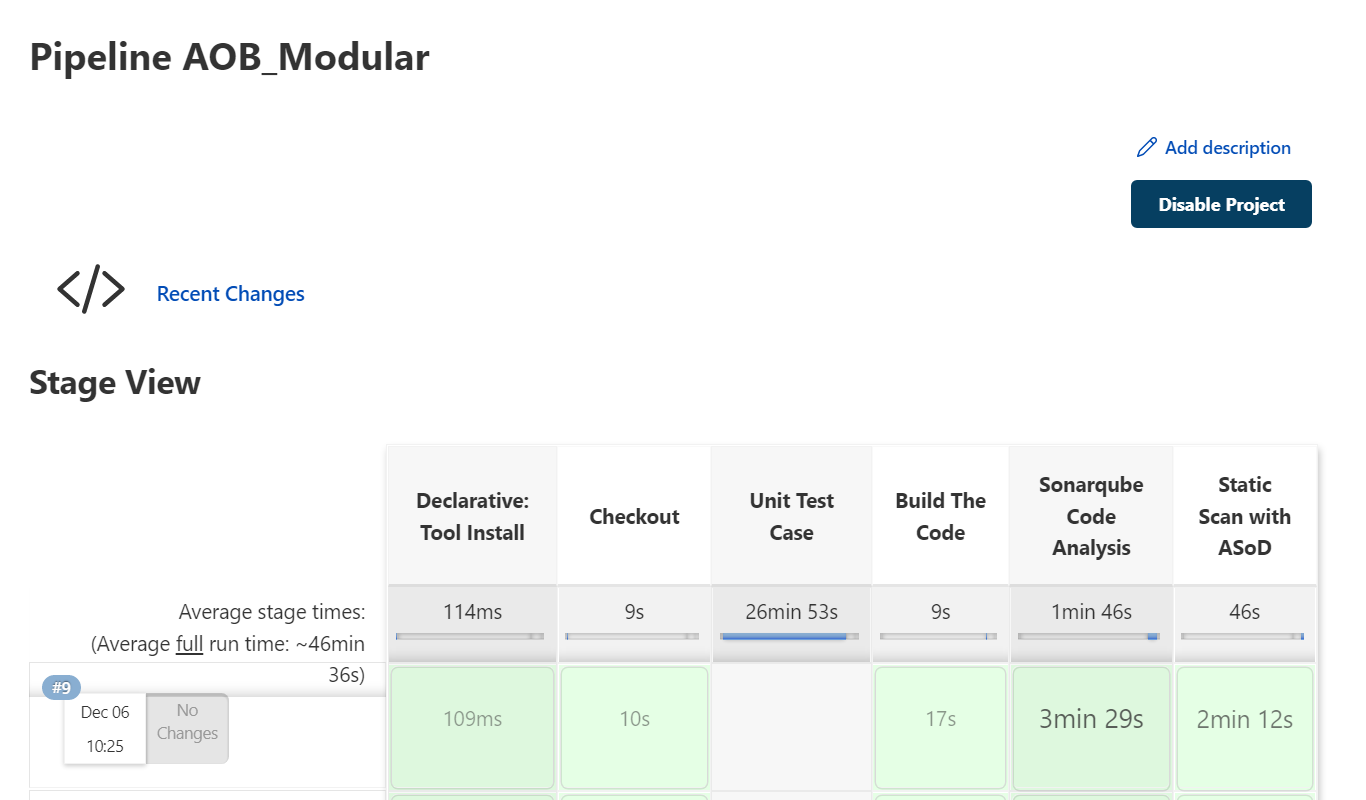
1. Inside each stage pipeline use when expression to call the respective parameters at each stage.



1. Click on apply and save. Now we should be able to see an option ‘Build with parameters. Select the stage to perform the build. Ex : For demo selected ‘Deploy’ stage.



1. Pipeline will get successfully get executed with all stage.



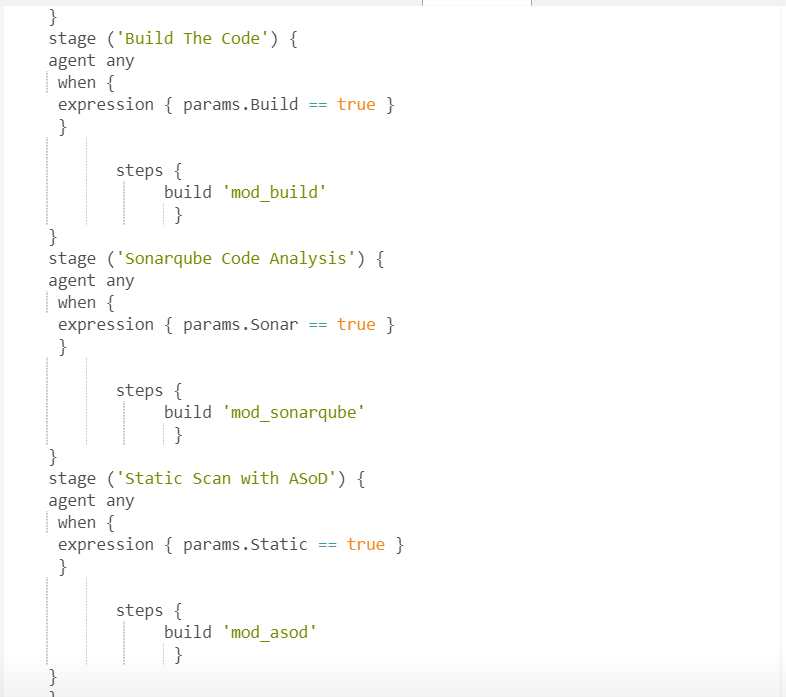
Note : We can select Multiple stages and single stage to execute by selecting the checkboxes.

**AOB Modular Pipeline:**

1. AOB Pipeline consists of 5 stages.

* Checkout
* Unit test case
* Build the code
* Sonarqube Code Analysis
* Static Scan with ASoD.



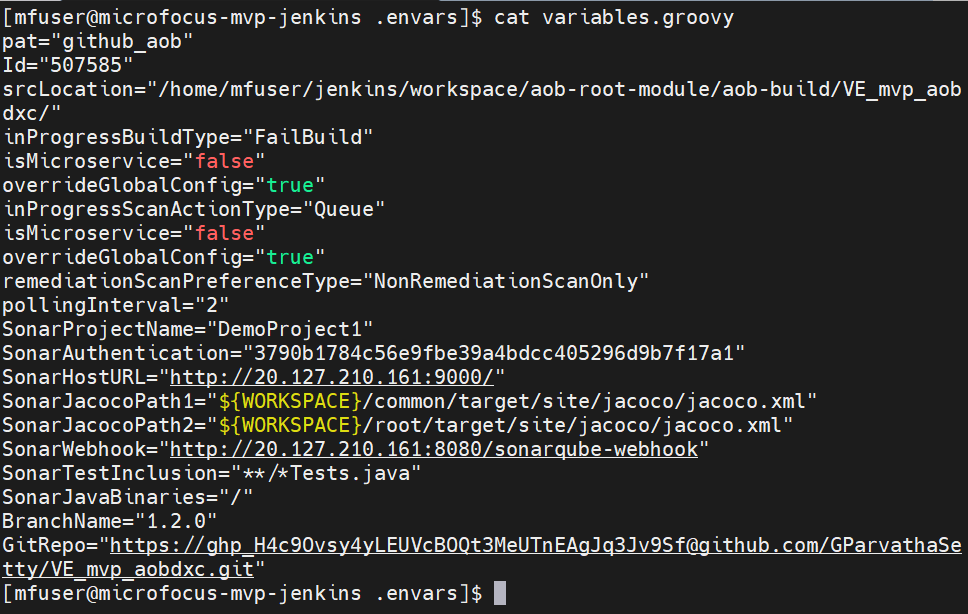


**Triggering respective jobs**

1. The main pipeline stage can trigger their respective jobs by using the “*build*” command.
2. The “*Checkout*” stage in the main pipeline triggers a separate pipeline i.e “*mod\_checkout*” pipeline which also contains stages and steps.
3. The “*Unit Test Case*” stage in the main pipeline triggers a separate pipeline i.e “*Unit Test*” pipeline which also contains stages and steps.
4. The “*Build The Code*” stage in the main pipeline triggers a separate pipeline i.e “*mod\_build*” pipeline which also contains stages and steps.
5. The “*Sonarqube Code Analysis*” stage in the main pipeline triggers a separate pipeline i.e “*mod\_sonarqube*” pipeline which also contains stages and steps.
6. The “*Static Scan with ASoD*” stage in the main pipeline triggers a separate pipeline i.e “*mod\_asod*” pipeline which also contains stages and steps.

**Variablization in Jenkins pipeline :**

1. Variablization is done in ASoD stage and Sonarqube stage.
2. Open the terminal and under this path i.e /var/lib/Jenkins/.evars/variables.groovy and declare the variables in ASoD and sonarqube stage variables with their values.



1. In the pipeline script under ASoD stage map the variables. Use the load command to Evaluate a Groovy source file into the Pipeline script.



1. In the pipeline script under the Sonarqube stage map the variables. Use the load command to Evaluate a Groovy source file into the Pipeline script.

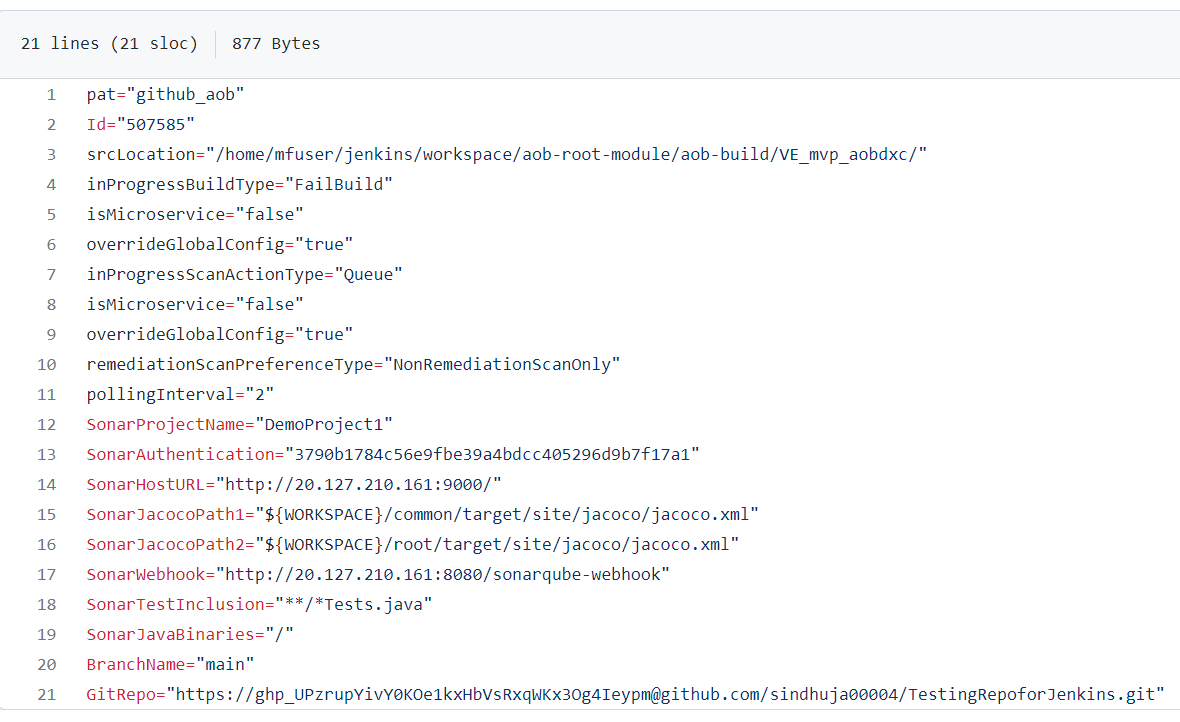


1. Build the pipeline and the output is as follows.



LOADING VARIABLES FROM GITHUB

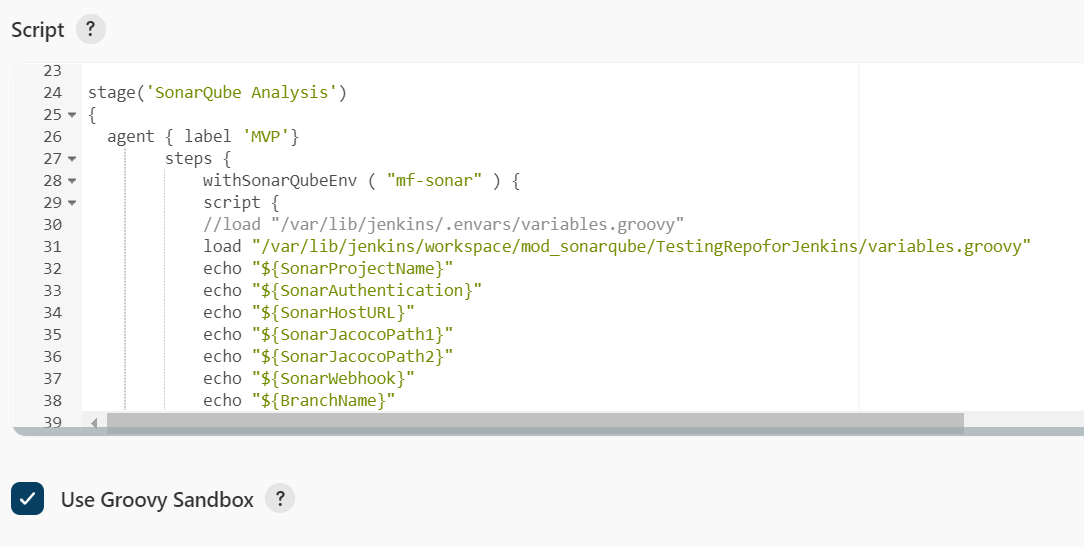
1. Create a file in the repository and add all the variables to the file and commit the changes.



1. In the main Jenkins pipeline clone the git repository.



1. Load the variables into the Jenkins pipeline.



1. The console output would be as follows.

