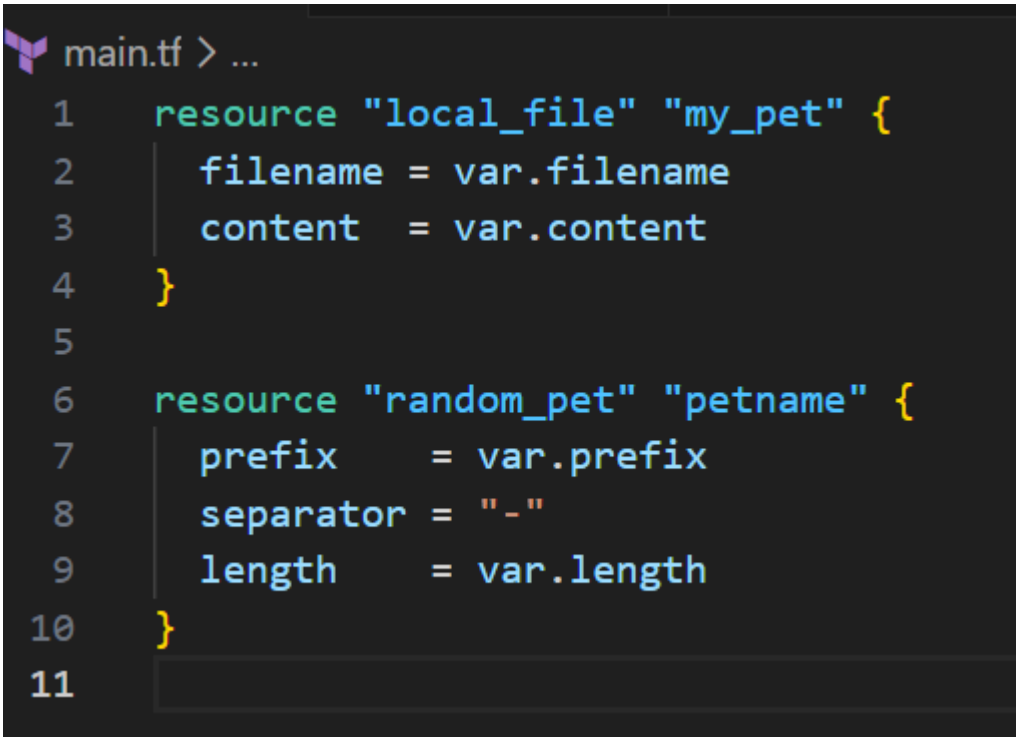
TERRAFORM 3

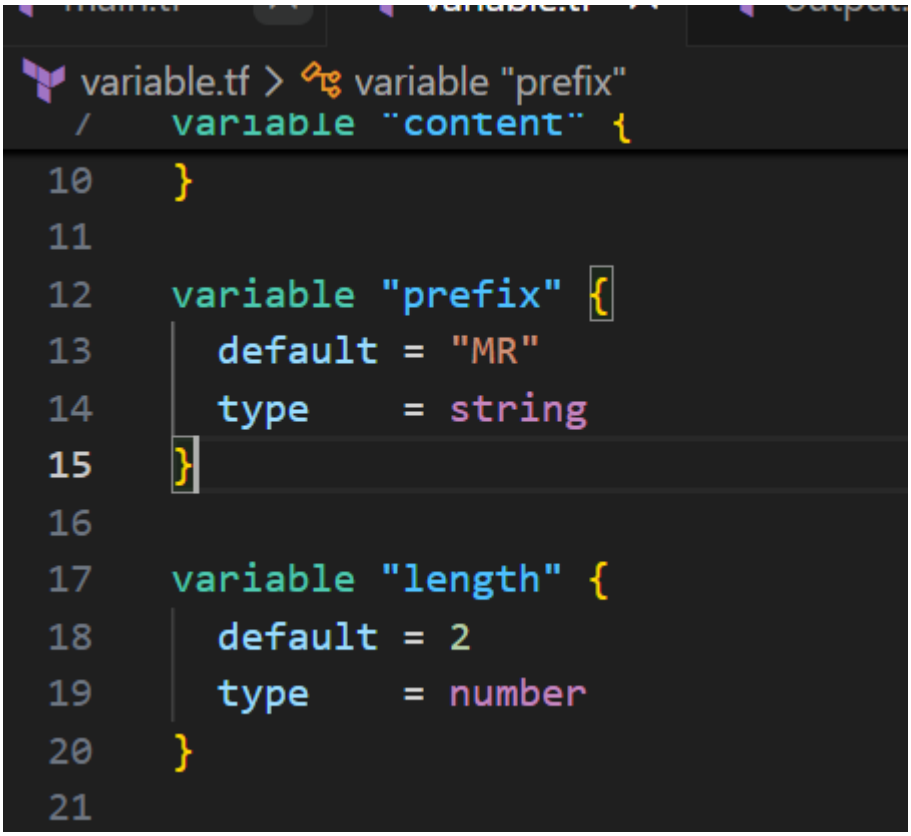
1. Watch the **Terraform-03** video.
2. Execute the script shown in the video.

- Add a folder in desktop and pull it init vs code
- Add 3 files to it main.tf
- Variable.tf
- Output.tf
- Add scriptto mian.tf
- resource "local_file" "my_pet" {
- filename = var.filename
- content = var.content
- }
-
- resource "random_pet" "petname" {
- prefix   = var.prefix
- separator = "-"
- length   = var.length
- }

- main.tf

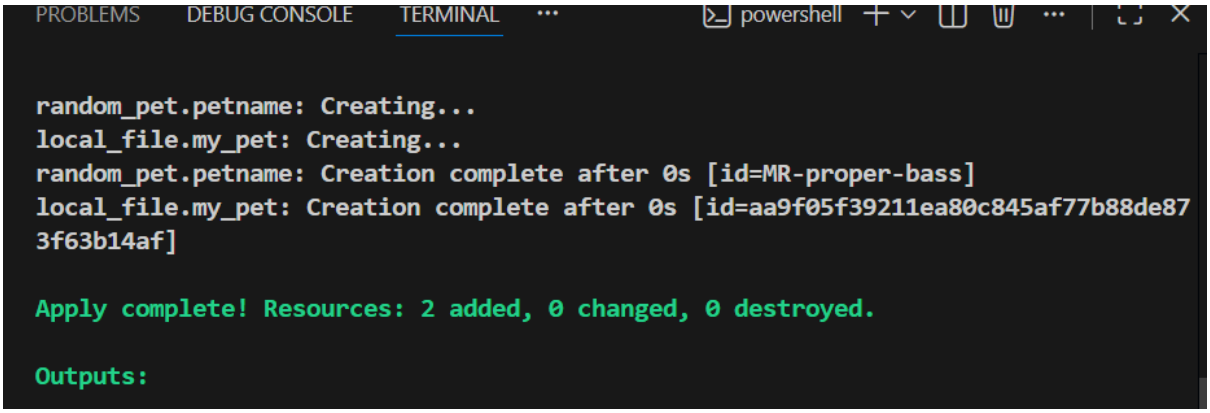


- Variable file script

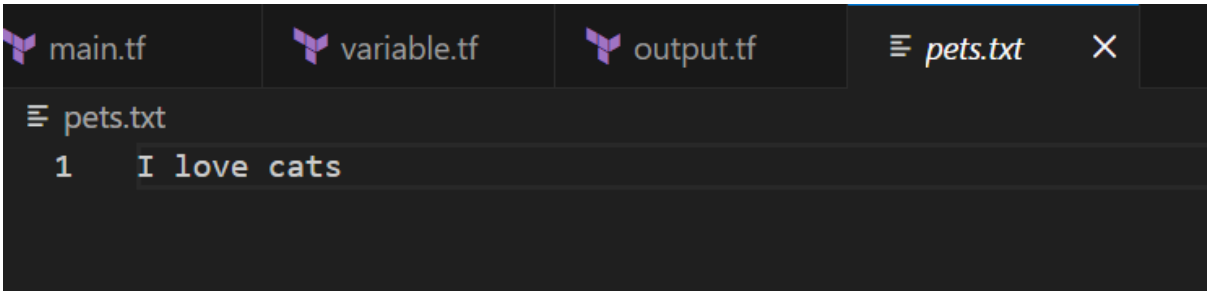


- Output.tf

- Now open terminal add and give commands
- Terraform init
- Terraform plan and
- Terraform apply



- Now a file will b e created



Now ----------

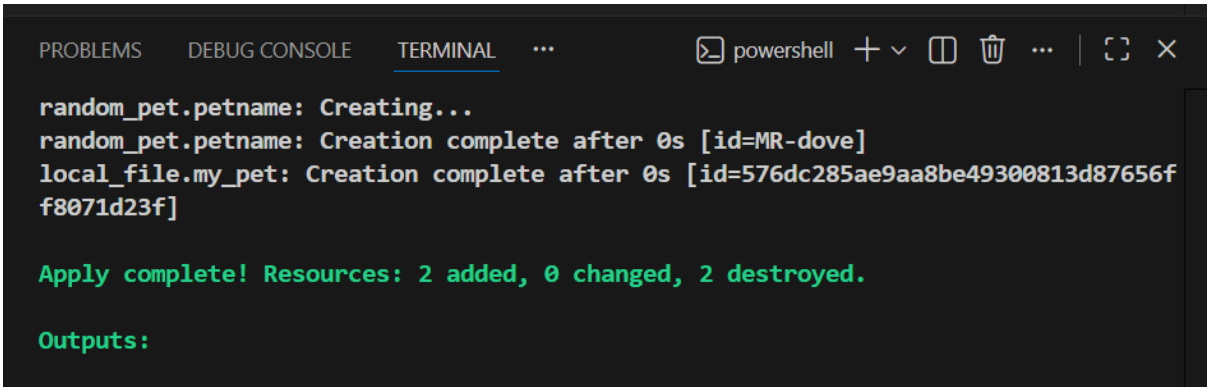
I wan t to add something to alredy genereated file

So I need tro change content in variable.tf file

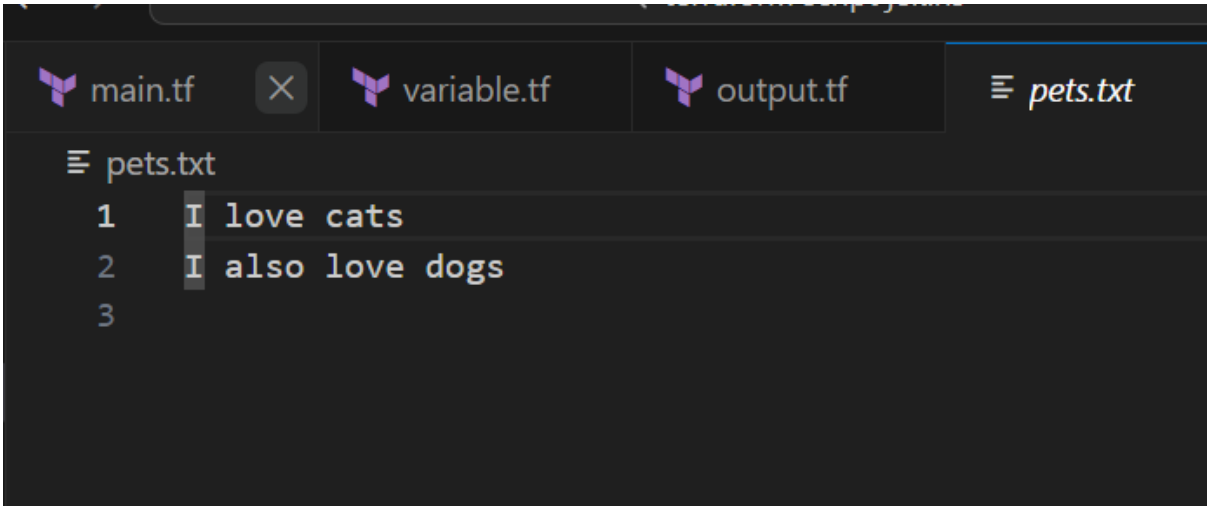
Add this script

```
variable "filename" {
  default     = "pets.txt"
  type        = string
}

variable "content" {
  default = <<EOF
I love cats
I also love dogs
EOF
  type = string
}

variable "prefix" {
  default = "MR"
  type    = string
}

variable "length" {
  default = 2
  type    = number
}
```

Run terminal with terraform apply

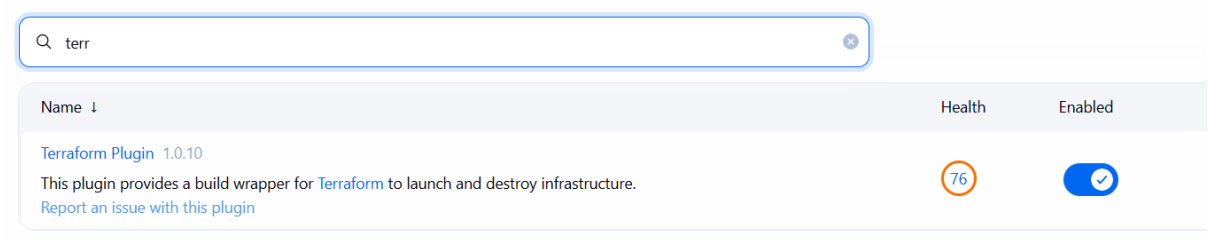


- Output

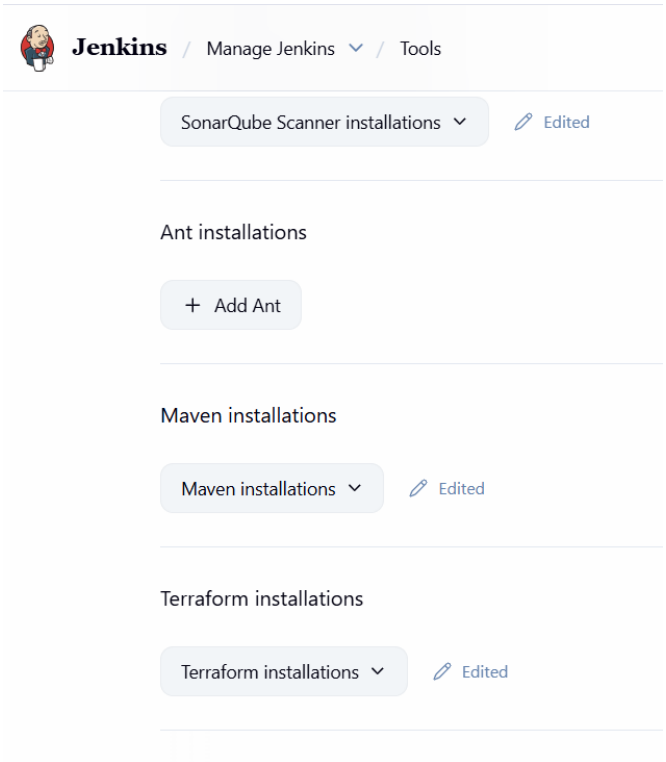


. 3. Integrate Terraform in Jenkins using the Terraform plugin

- Login to Jenkins
- Manage Jenkins
- >. Pulgins and install terraform



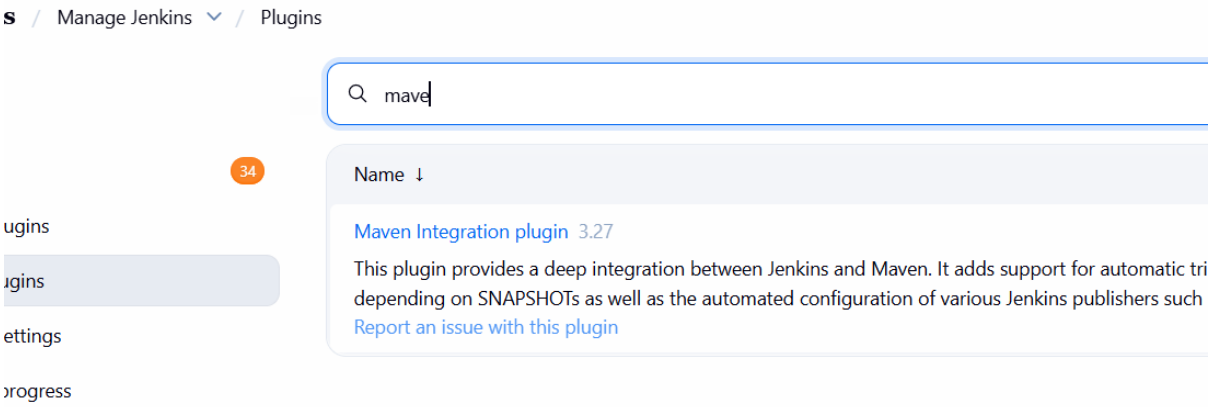
- And go to tools and add terraform



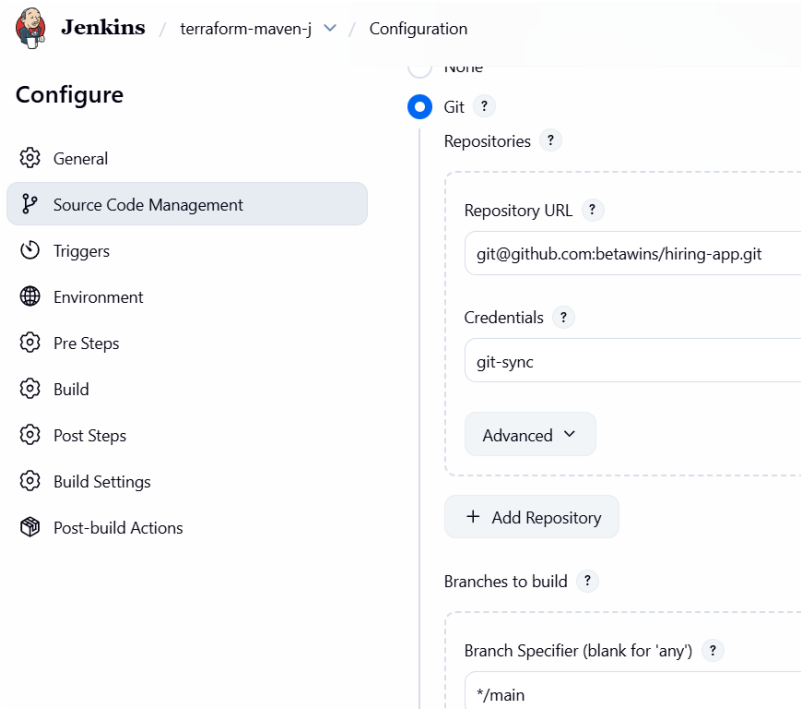
4 . Create one Jenkins job using **Maven Project** for the code below with **two stages**:

- **Stage 1:** Git clone
- **Stage 2:** Maven Compilation **Code:** https://github.com/betawins/java-Working-app.git

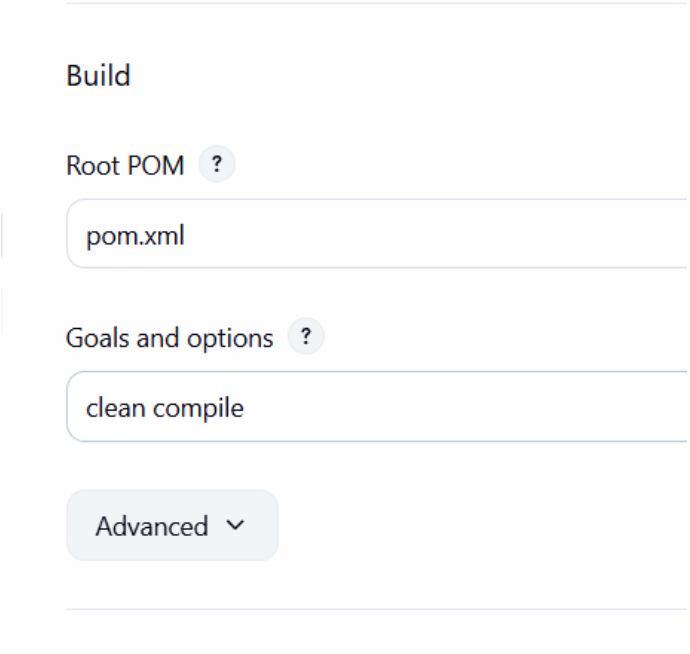
- Login into Jenkins
- >> pulgins install maven



- Create new item
- Select maven project in the options
- Add git clone url

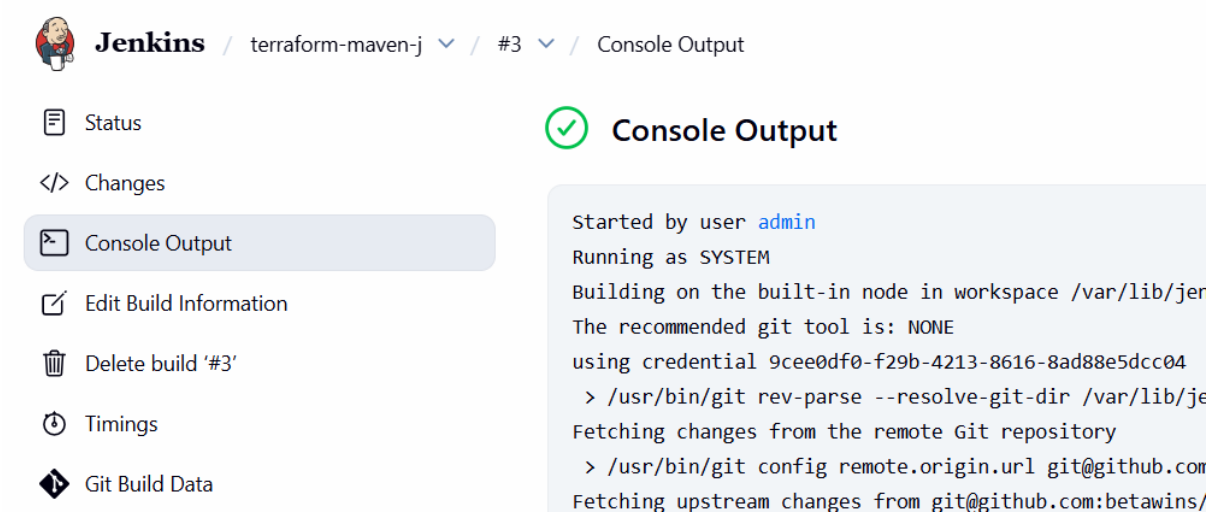


- And scroll to build



- In the root coloum give POM.XML
- In th e goals give clean complie

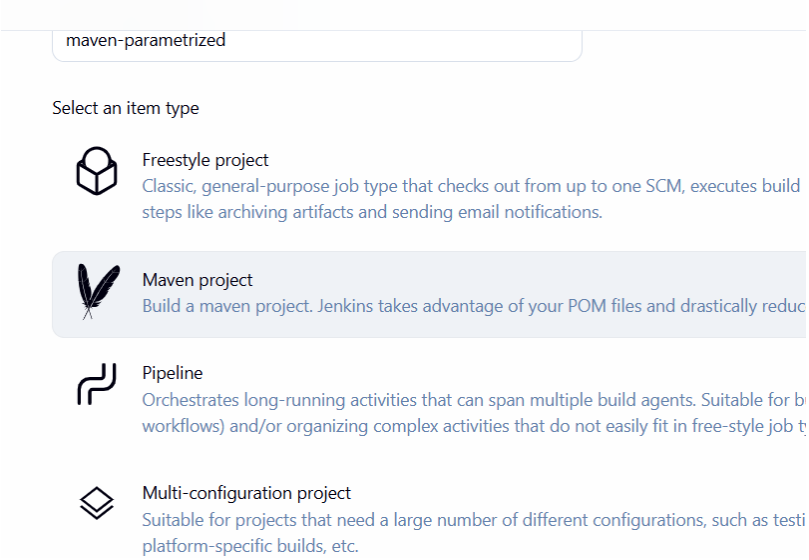
- Run the job



5 . Use the same code and create a **parameterized job** in Jenkins with:

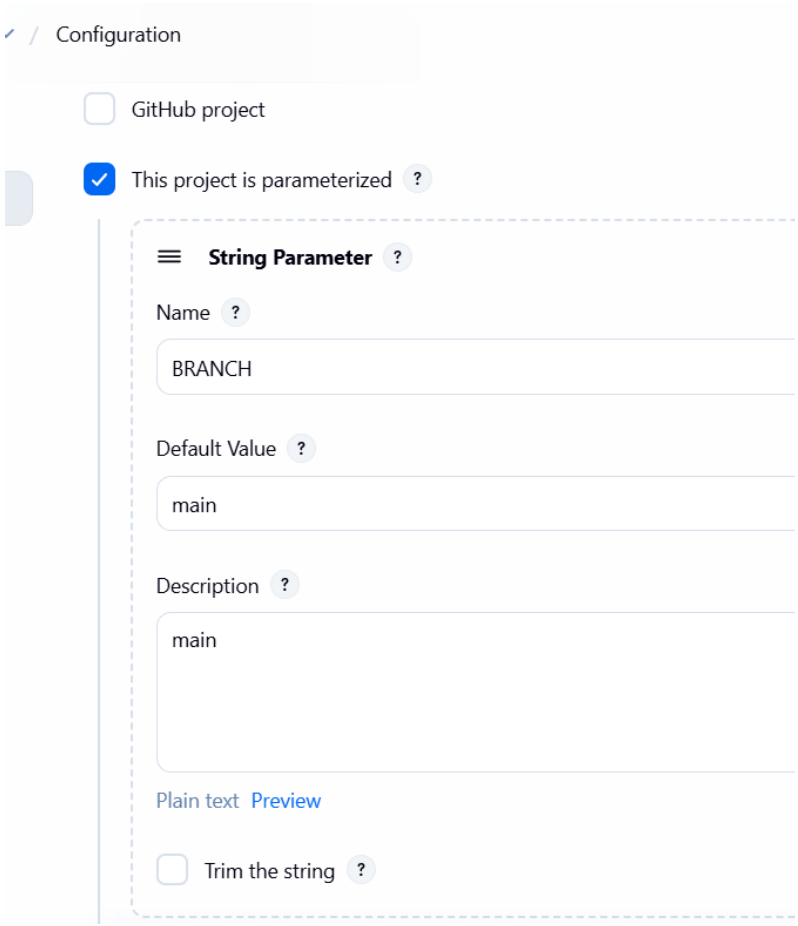
- **Stage 1:** Git clone
- **Stage 2:** Maven Compilation **Code:** Use the same code and create a **parameterized job** in Jenkins with:
  : https://github.com/betawins/java-Working-app.git

- Select maven project

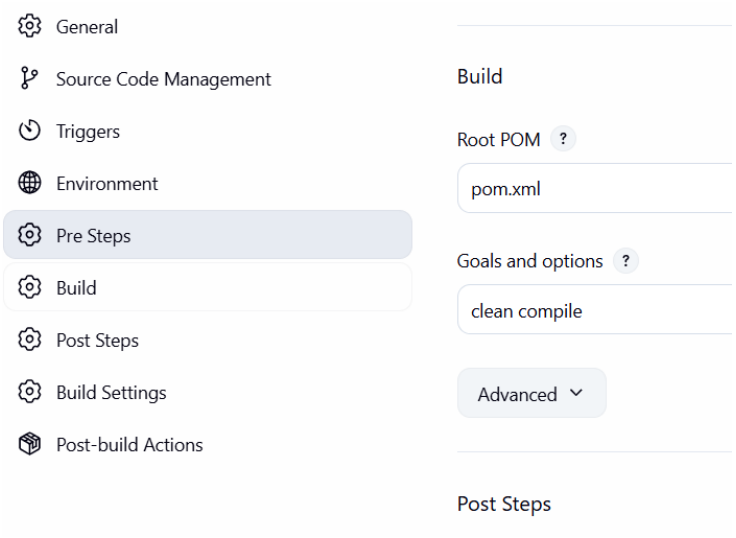


- Now scroll to this project is parametrized.

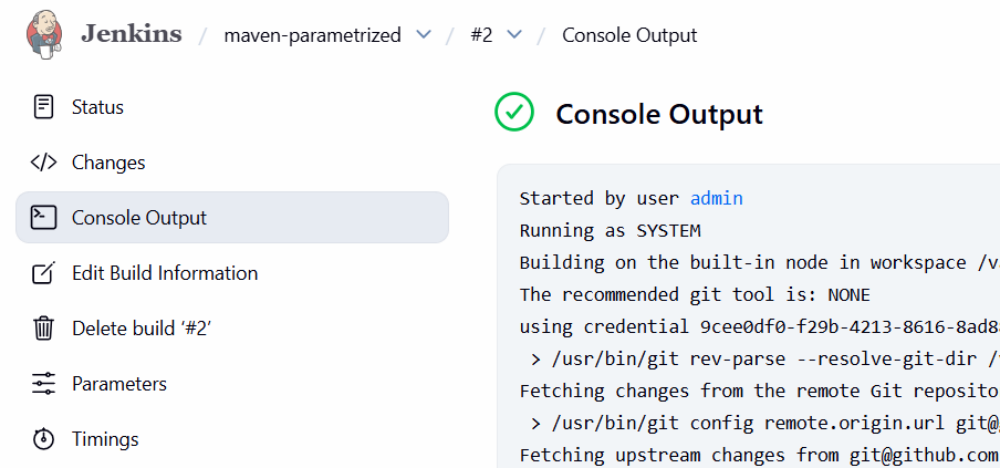
- Select string parameter



- Add git clone url
- Now scroll to build
- And in root add POM.XML
- Now, in goals give ….. clean complie



- Output



6 . What are the **global variables** in Jenkins?

✅ Jenkins Global Variables – Full Table

1️⃣ Built-in Environment Variables

Variable	Description
BUILD_NUMBER	Current build number (auto increments)
BUILD_ID	Unique build ID
BUILD_DISPLAY_NAME	Display name of the build
BUILD_URL	URL of this build in Jenkins

Variable	Description
JOB_NAME	Name of the Jenkins job
JOB_BASE_NAME	Job name without folder path
WORKSPACE	Full workspace directory path
NODE_NAME	Jenkins agent (node) name
NODE_LABELS	Labels assigned to the node
JENKINS_HOME	Jenkins home directory
JENKINS_URL	Base URL of Jenkins
EXECUTOR_NUMBER	Executor ID that is running the job

2️⃣ Git / SCM Variables

Variable	Description
GIT_COMMIT	Git commit hash checked out
GIT_BRANCH	Git branch name
GIT_URL	Git repository URL
GIT_PREVIOUS_COMMIT	Previous commit hash
GIT_PREVIOUS_SUCCESSFUL_COMMIT	Last successful commit

3️⃣ Pipeline Global Objects

Variable	Description
env	Access all environment variables
params	Access job parameters
scm	Auto SCM checkout configuration
currentBuild	Provides build details (status, ID, result)
docker	Use Docker containers inside pipeline
pipeline	Declarative pipeline root object
steps	Access pipeline steps
agent	Defines where pipeline runs

4️⃣ Job Parameter Variables (when using “This project is parameterized”)

Variable	Description
${PARAM_NAME}	Any String, Choice, Boolean parameter
${BRANCH}	Example branch parameter
${GOALS}	Example Maven goals parameter
${ENVIRONMENT}	Example env parameter (dev/test/prod)

5️⃣ Shared Library Global Variables

Variable	Description
Custom global FN (e.g. hello())	Provided by Jenkins Shared Libraries under vars/
Custom classes	Reusable logic across pipelines

⭐ BONUS: Short “MOST IMPORTANT” Table (for interviews)

Category	Important Variables
Build Info	BUILD_NUMBER, BUILD_ID, BUILD_URL
Job Info	JOB_NAME, WORKSPACE, NODE_NAME
Pipeline	env, params, scm, currentBuild
Git/SCM	GIT_COMMIT, GIT_BRANCH, GIT_URL