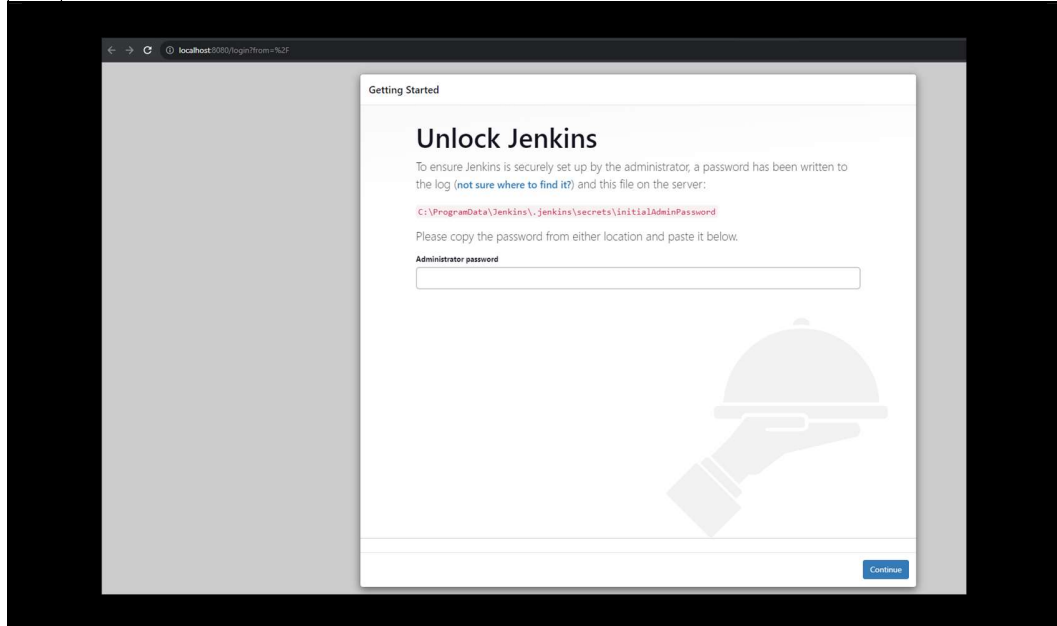


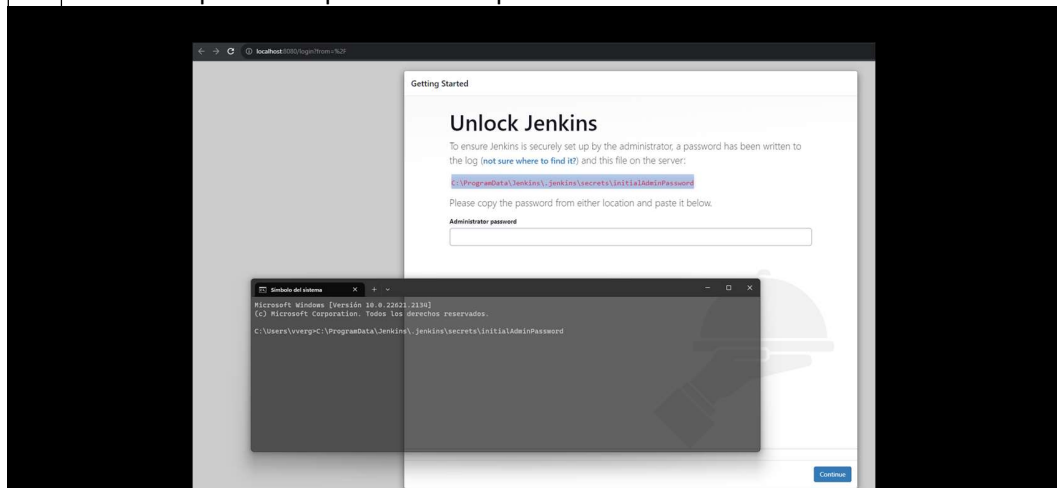
## Unidad 4 – Jenkins – Docker – Terraform

Los siguientes pasos indican todo lo necesario para realizar la prueba de un ejemplo de pipeline utilizando un repositorio GIT, todo controlado por Jenkins, chequeado por tfSec y aplicado por Docker.

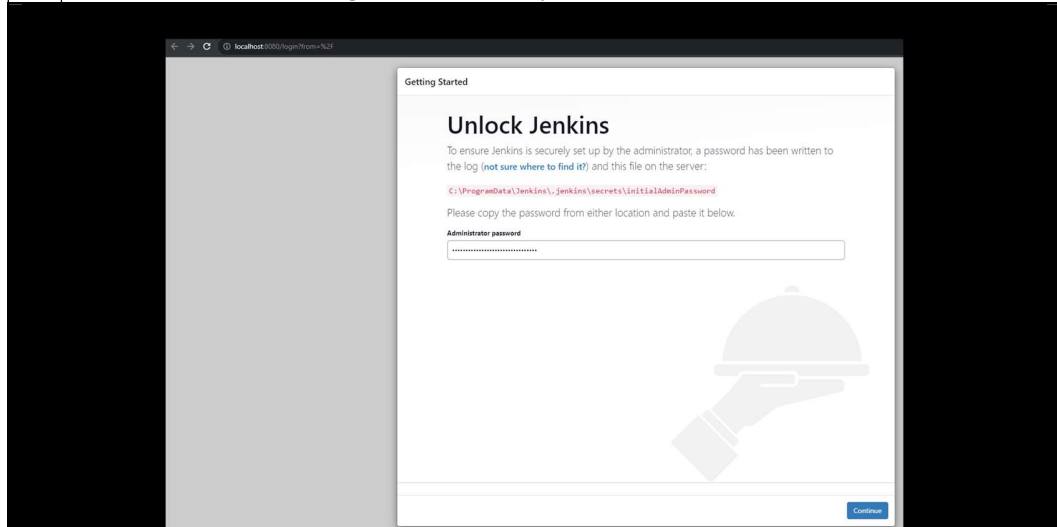
### 1. Descargar e instalar Jenkins en maquina Local WINDOWS



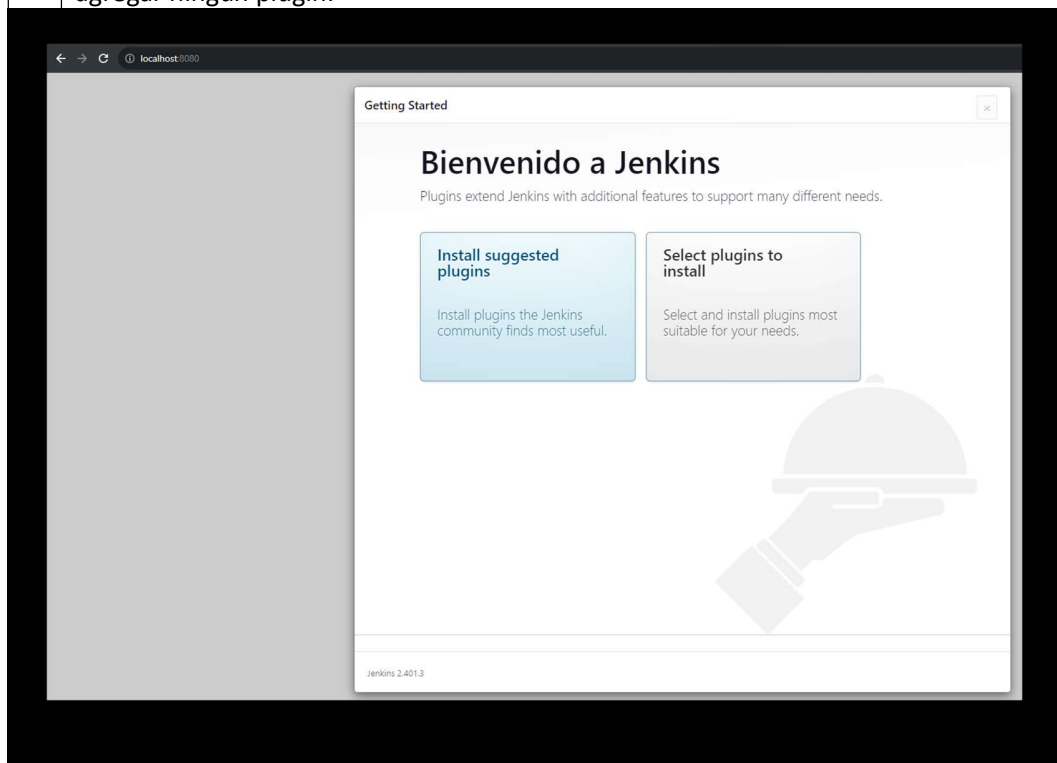
### 2. Insertar la password que fue creada por el instalador de Jenkins.



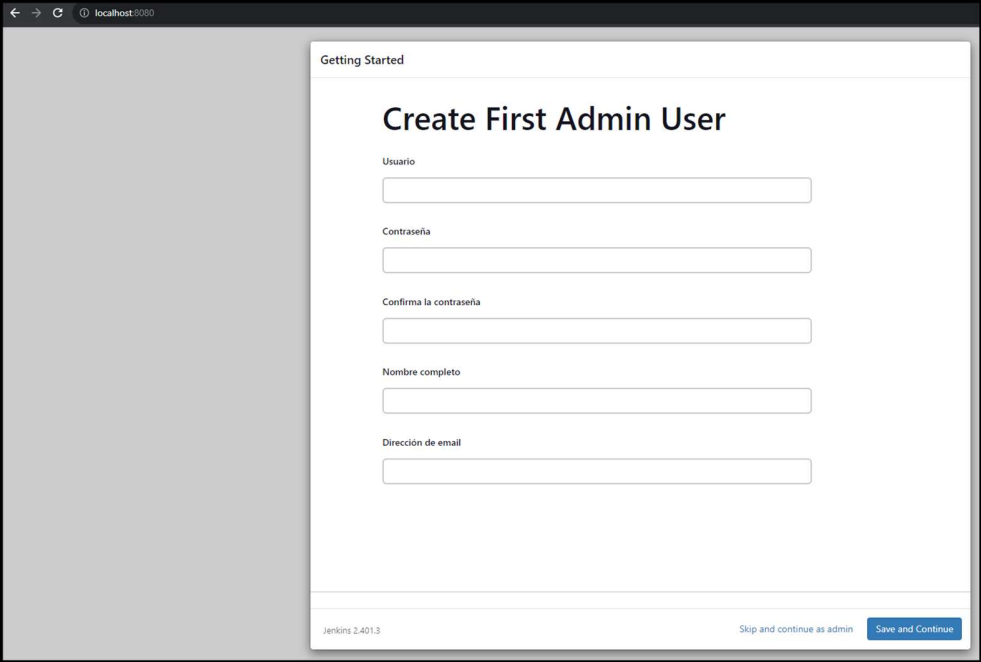
3. Presionar Continue, luego de insertar la password de administrador.



4. Realizar la instalación de los plugins sugeridos. En esta oportunidad no se necesita agregar ningún plugin.



5. Luego de instalar, agregue la información solicitada.



The screenshot shows the Jenkins 'Getting Started' page with the 'Create First Admin User' form. The form is titled 'Create First Admin User' and is part of the 'Getting Started' section. It contains several input fields for user information: 'Usuario', 'Contraseña', 'Confirma la contraseña', 'Nombre completo', and 'Dirección de email'. At the bottom of the form, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The browser address bar shows 'localhost:8080'.

Getting Started

## Create First Admin User

Usuario

Contraseña

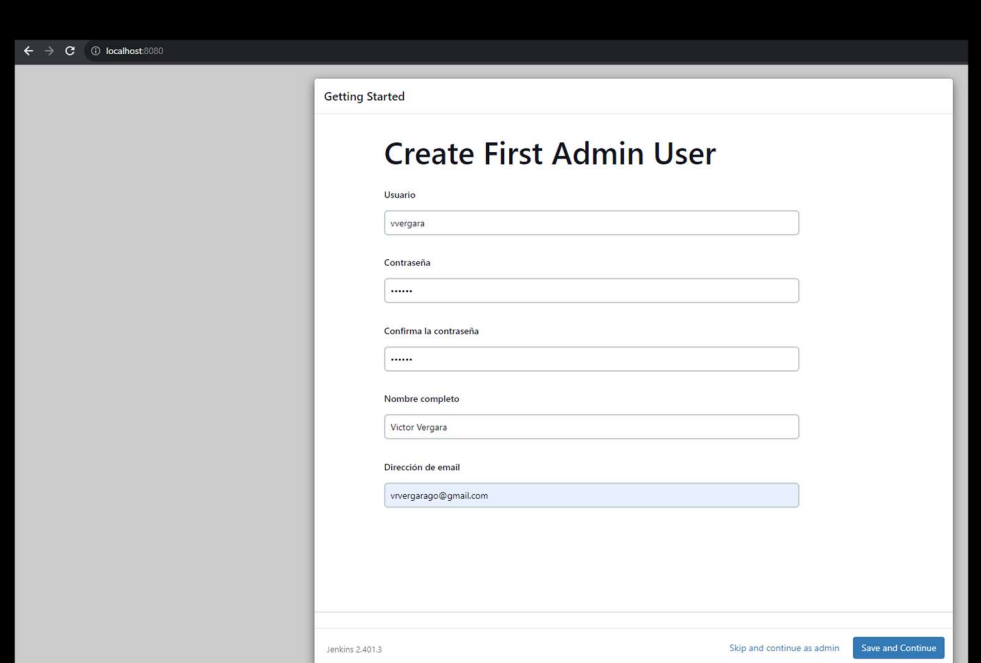
Confirma la contraseña

Nombre completo

Dirección de email

Jenkins 2.401.3 Skip and continue as admin Save and Continue

6. Luego de ingresar todo los datos solicitados, presione en Save and Continue.



The screenshot shows the same Jenkins 'Create First Admin User' form, but now with data entered in all the input fields. The 'Usuario' field contains 'vvergara', the 'Contraseña' and 'Confirma la contraseña' fields contain '\*\*\*\*\*', the 'Nombre completo' field contains 'Victor Vergara', and the 'Dirección de email' field contains 'vvergarago@gmail.com'. The 'Save and Continue' button is highlighted in blue. The browser address bar shows 'localhost:8080'.

Getting Started

## Create First Admin User

Usuario

Contraseña

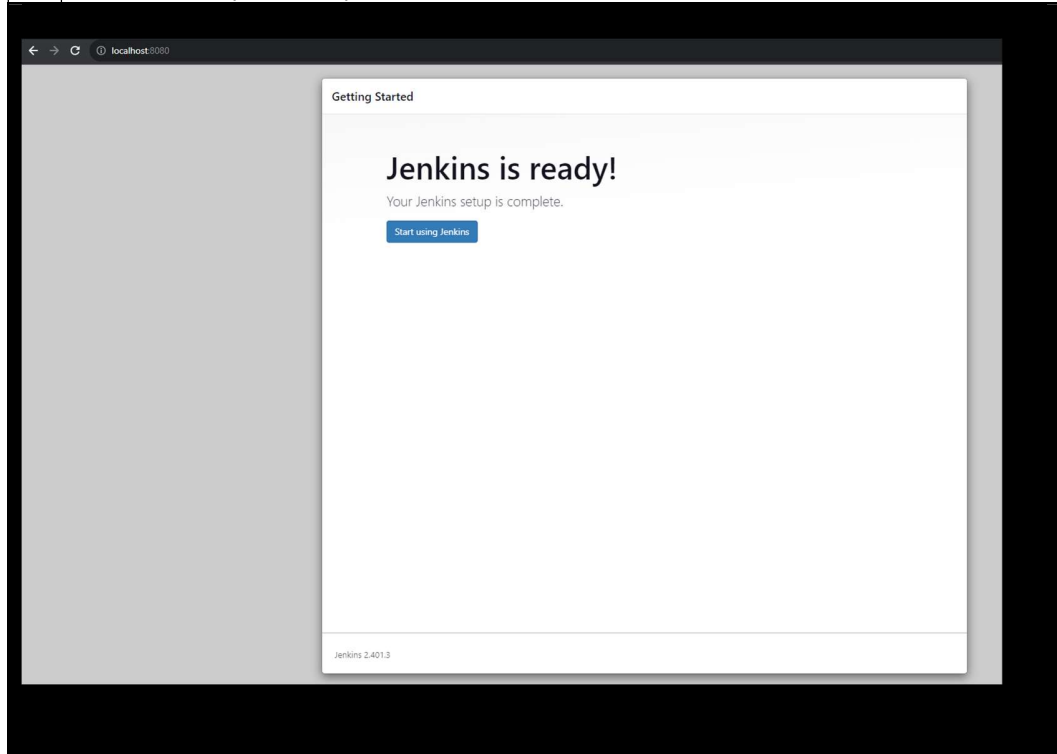
Confirma la contraseña

Nombre completo

Dirección de email

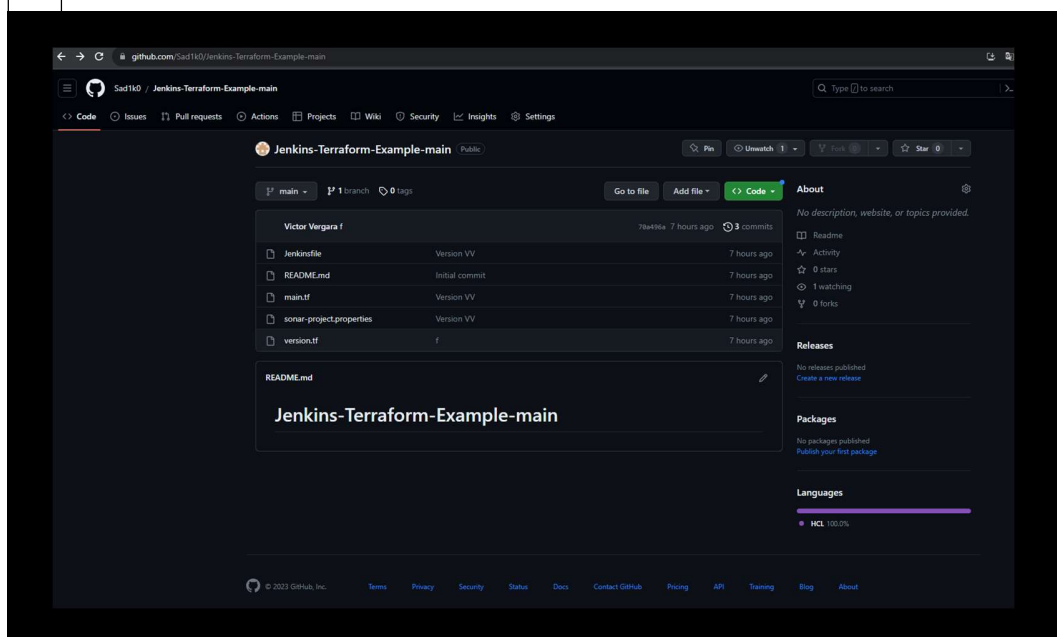
Jenkins 2.401.3 Skip and continue as admin Save and Continue

7. Al terminar, aparece la pantalla final de instalación de Jenkins.

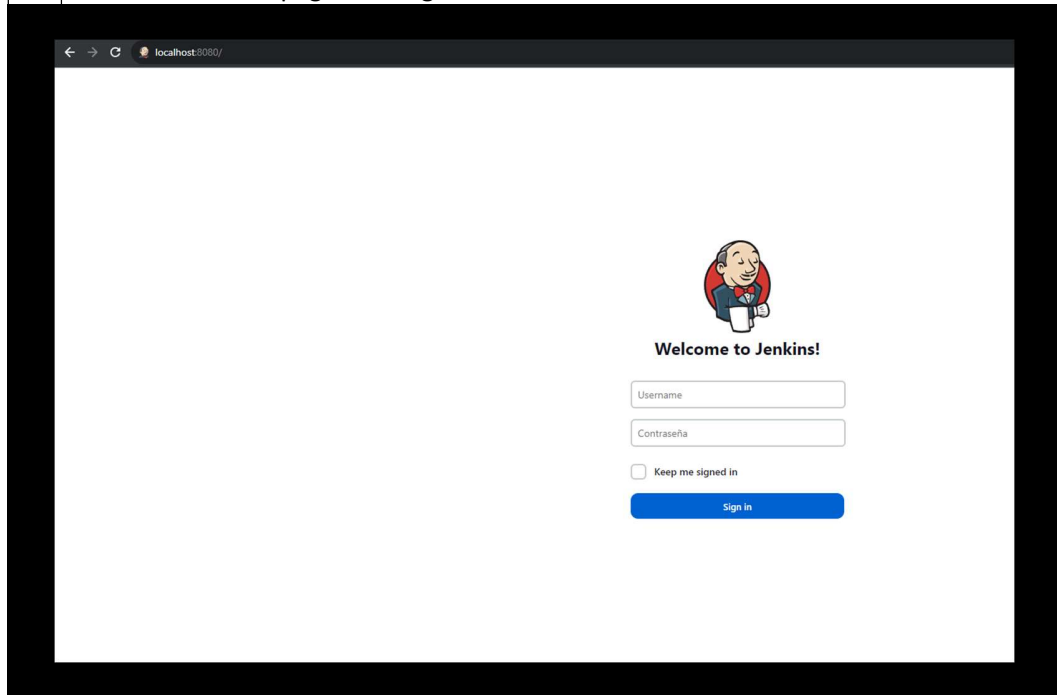


8. Para el ejemplo se utilizará un proyecto en Terraform que fue suministrado y cargado en GITHUB para los cambios necesarios.  
El repositorio es:

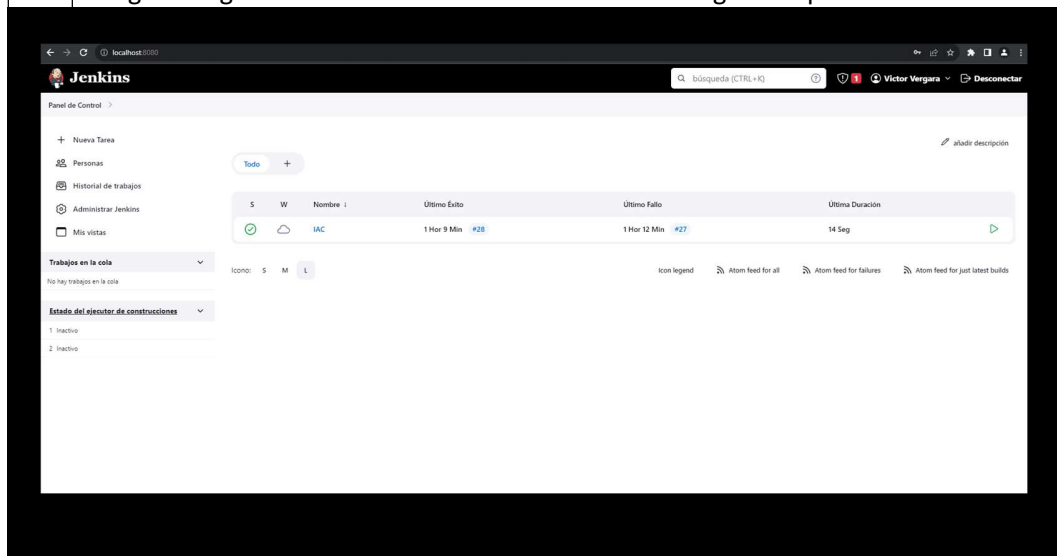
<https://github.com/Sad1k0/Jenkins-Terraform-Example-main>



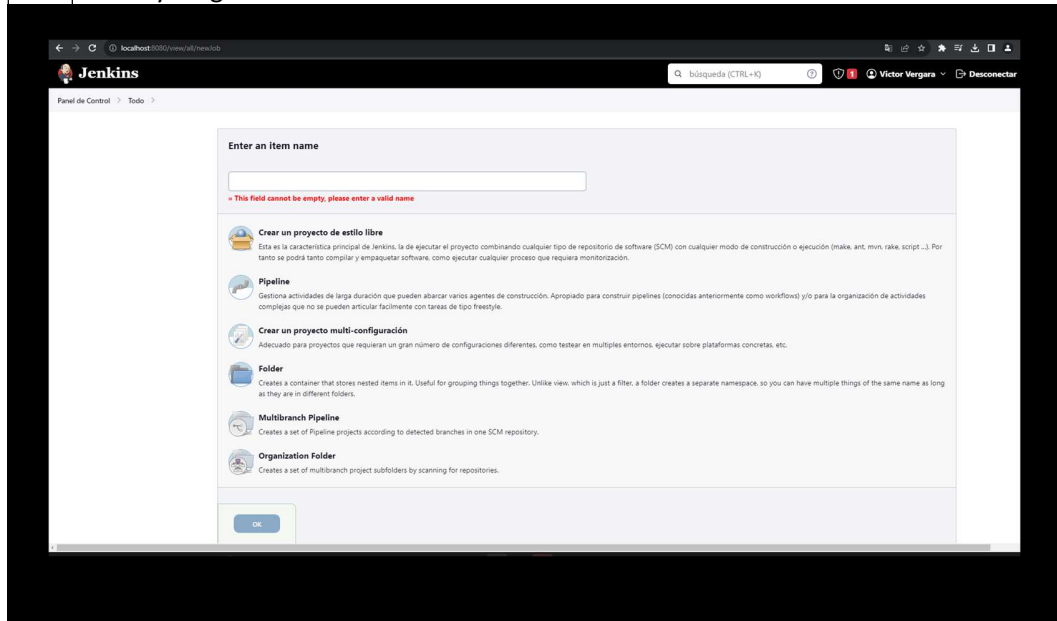
9. En el navegador a elección, se coloca la siguiente URL.  
<http://localhost:8080/>  
la cual mostrar la pagina de ingreso a Jenkins.



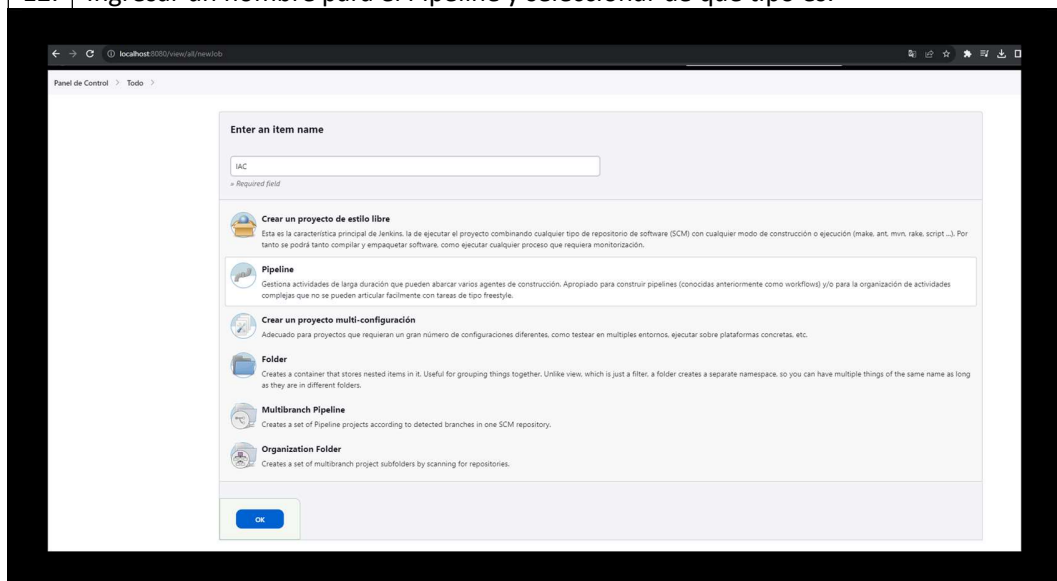
10. Luego de ingresar con las credenciales se muestra la siguiente pantalla.



11. Se debe crear nueva nuevo pipeline, que permita definir todos los pasos por los cuales pasara el proyecto, desde la descarga desde GIT, revisión del código por TfSec y luego levantar un contenedor en Docker.



12. Ingresar un nombre para el Pipeline y seleccionar de que tipo es.



13. Seleccione un nombre y se define el máximo de ejecuciones para guardar.

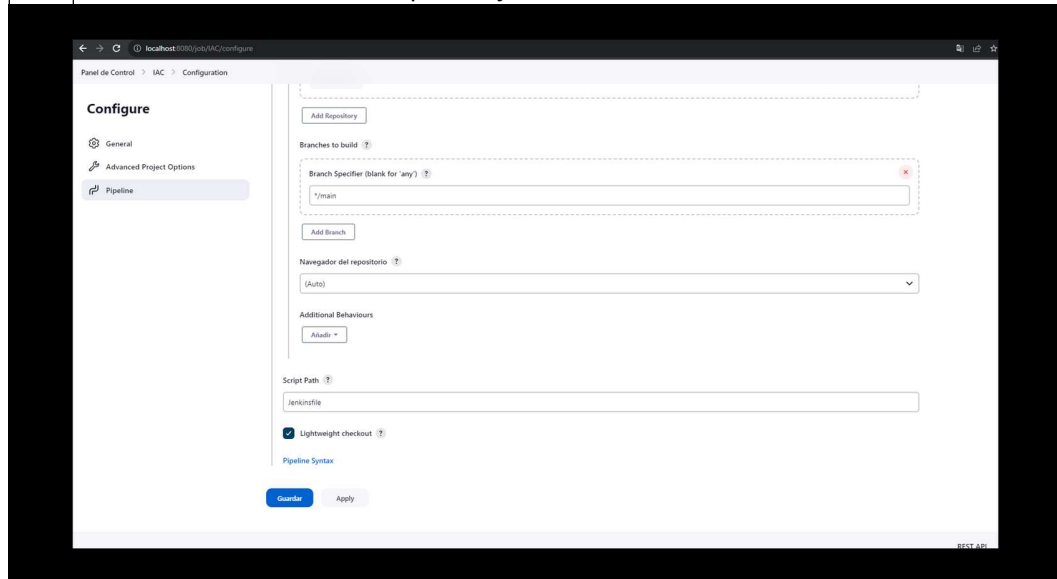
The screenshot shows the Jenkins 'Configure' page for a job named 'IAC de prueba'. The left sidebar has tabs for 'General', 'Advanced Project Options', and 'Pipeline'. The 'General' tab is active. The 'Description' field contains 'IAC de prueba'. Below it, the 'Desechar ejecuciones antiguas' (Discard old builds) checkbox is checked. The 'Strategy' dropdown is set to 'Log Rotation'. Under 'Numero de dias para mantener ejecuciones de proyectos' (Number of days to keep project builds), the value '2' is entered. Below that, under 'Número máximo de ejecuciones para guardar' (Maximum number of builds to keep), the value '2' is entered. At the bottom, there are several unchecked checkboxes: 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'Esta ejecución debe parametrizarse' (This execution must be parametrized), 'GitHub project', and 'Pipeline speed/durability override'. 'Guardar' (Save) and 'Apply' buttons are at the bottom.

14. En la definición del Pipeline se define que se utilizara GIT, por lo cual se debe ingresar la url al repositorio y definir que rama se utilizara. Para este caso es la rama “main” y el repositorio es el creado para este ejemplo.

<https://github.com/Sad1k0/Jenkins-Terraform-Example-main.git>

The screenshot shows the Jenkins 'Configure' page for a job named 'IAC de prueba', specifically the 'Pipeline' tab. The 'Definition' dropdown is set to 'Pipeline script from SCM'. The 'SCM' dropdown is set to 'Git'. Under 'Repositories', the 'Repository URL' is set to 'https://github.com/Sad1k0/Jenkins-Terraform-Example-main.git'. The 'Credentials' dropdown is set to 'none'. Below that, the 'Branches to build' section is expanded, and the 'Branch Specifier (blank for 'any')' is set to '\*/main'. 'Guardar' (Save) and 'Apply' buttons are at the bottom.

15. En Script Path, definir el archivo que se encuentre dentro del repositorio que contiene toda la secuencia que se ejecutara.

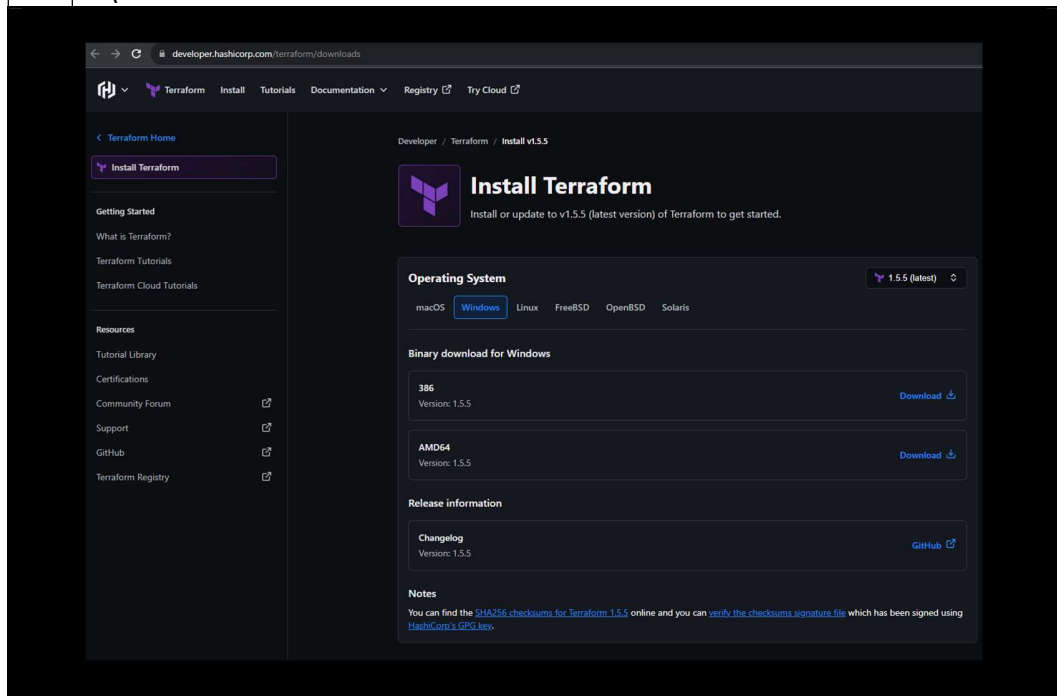


16. El archivo Jenkinsfile, se modifico para que las pruebas resultaran exitosas, las cuales incluye:
- Cambio de sh por bat.
  - Cambio de la ubicación de Docker
  - Cambio de los argumentos que son entregados a Docker (no se utiliza variables)
  - Cambio en la ubicación de terraform.

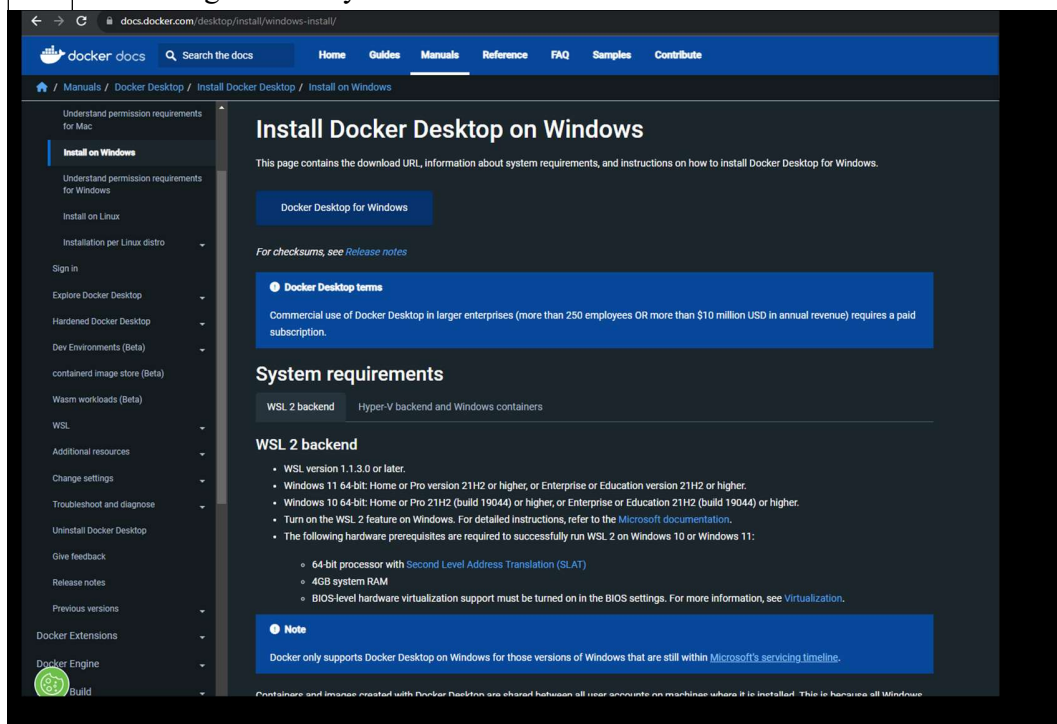
```
pipeline {
  agent any
  options {
    skipDefaultCheckout(true)
  }
  stages {
    stage('clean workspace') {
      steps {
        cleanWs()
      }
    }
    stage('checkout') {
      steps {
        checkout scm
      }
    }
    stage('tfsec') {
      steps {
        bat 'docker run --rm -v "V:/Usach/Terraform/Jenkins-Terraform-Example-main/JTF_vv/Jenkins-Terraform-Example-main:/src" aquasec/tfsec .'
      }
    }
    stage('Approval for Terraform') {
      steps {
        input(message: 'Approval required before Terraform', ok: 'Proceed', submitterParameter: 'APPROVER')
      }
    }
    stage('terraform') {
      steps {
        bat 'c:/terraform/terraform apply -auto-approve -no-color'
      }
    }
  }
  post {
    always {
      cleanWs()
    }
  }
}
```



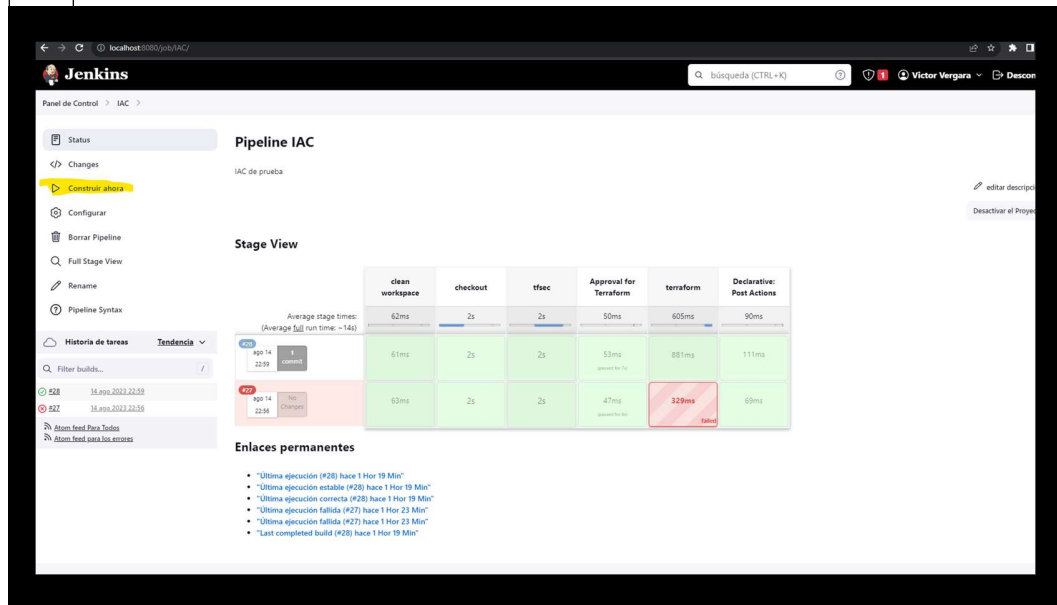
17. Se realizó la descarga de Terraform dejando el ejecutable en la carpeta c:\terraform.



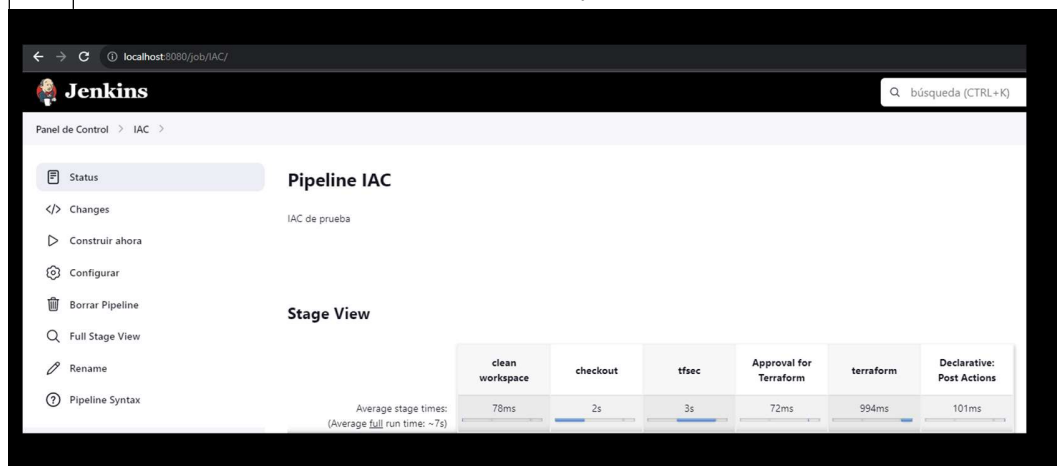
18. Se descargo Docker y se instaló.



19. Luego de tener todo instalado, configurado los path para terraform, los path de GIT para encontrar bat. Se puede ejecutar el pipeline, presionando en “construir ahora”



20. Como se puede ver, el pipeline se ejecuto correctamente luego de todos los cambios. Todo funcionando sobre una maquina Windows.



21. Viendo la consola de resultado



### Salida de consola

[illegible]

