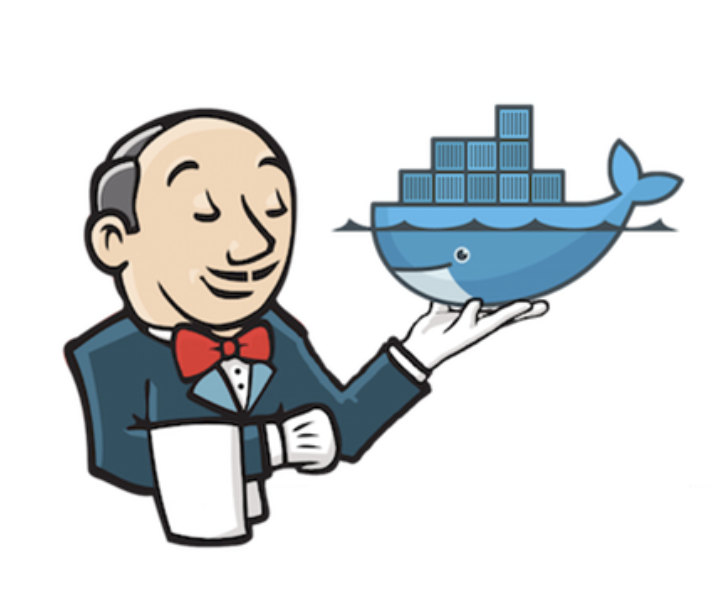
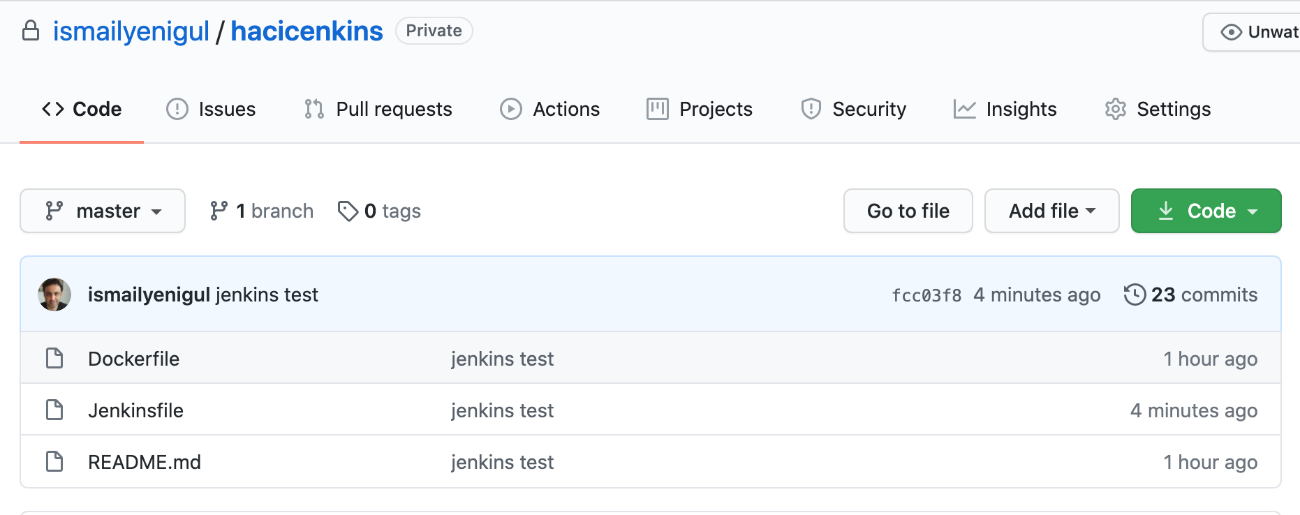
# Docker build / push con canalización declarativa en Jenkins



A continuación, se muestra un ejemplo rápido y completo para crear una imagen de Docker desde un repositorio privado de GitHub y enviarla al centro Docker con una canalización declarativa en Jenkins.

Creé un repositorio de muestra con Dockerfile básico y Jenkinsfile



$ cat Dockerfile

FROM ubuntu: 20.04   
RUN apt update && apt upgrade -y

**Configurar Jenkins**

Asegúrese de haber instalado el motor de Docker en la instancia de Jenkins y git, complementos de Docker

Instale Docker-CE en Ubuntu 18 desde <https://docs.docker.com/engine/install/ubuntu/>

Después de eso, permita que los usuarios de Jenkins accedan a docker socket.

sudo usermod -aG docker jenkins   
sudo systemctl reiniciar jenkins

Pruebe su acceso a Docker con el usuario de Jenkins

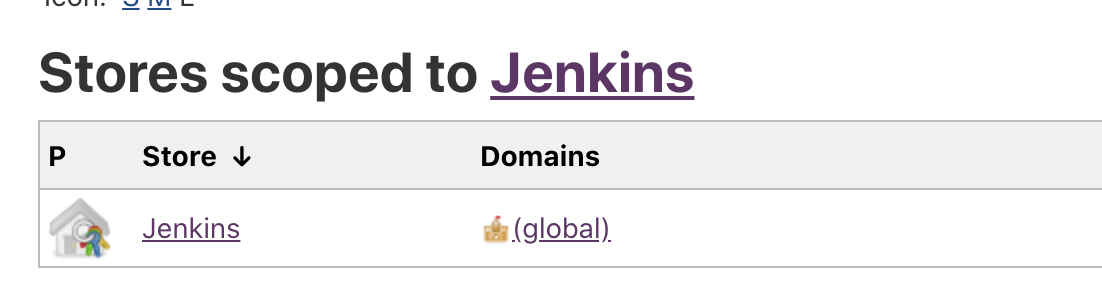
$ sudo su - jenkins  
$ docker psCONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**Cree credenciales en Jenkins para GitHub y docker hub**

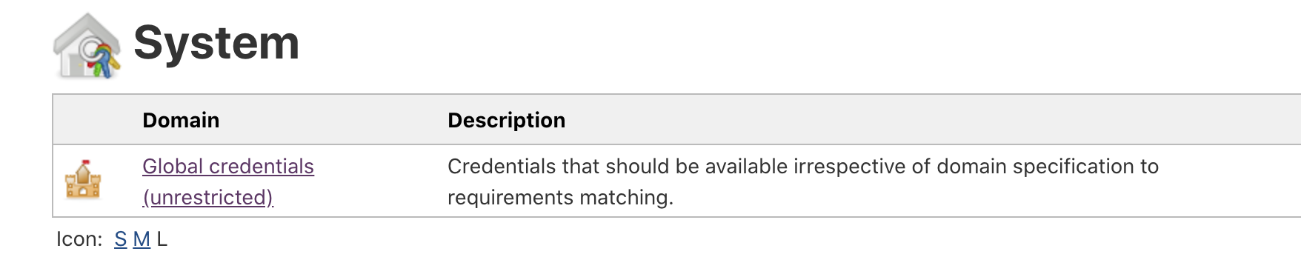
Para crear credenciales de GitHub, necesitamos crear un token. Verifique la siguiente URL para crear un token de API personal

[https://support.cloudbees.com/hc/en-us/articles/234710368-GitHub-Permissions-and-API-token-Scopes-for-Jenkin](https://support.cloudbees.com/hc/en-us/articles/234710368-GitHub-Permissions-and-API-token-Scopes-for-Jenkins)

Para crear credenciales de GitHub, Manage Jenkins->Manage Credentials(Under Security)haga clic en Jenkins Store

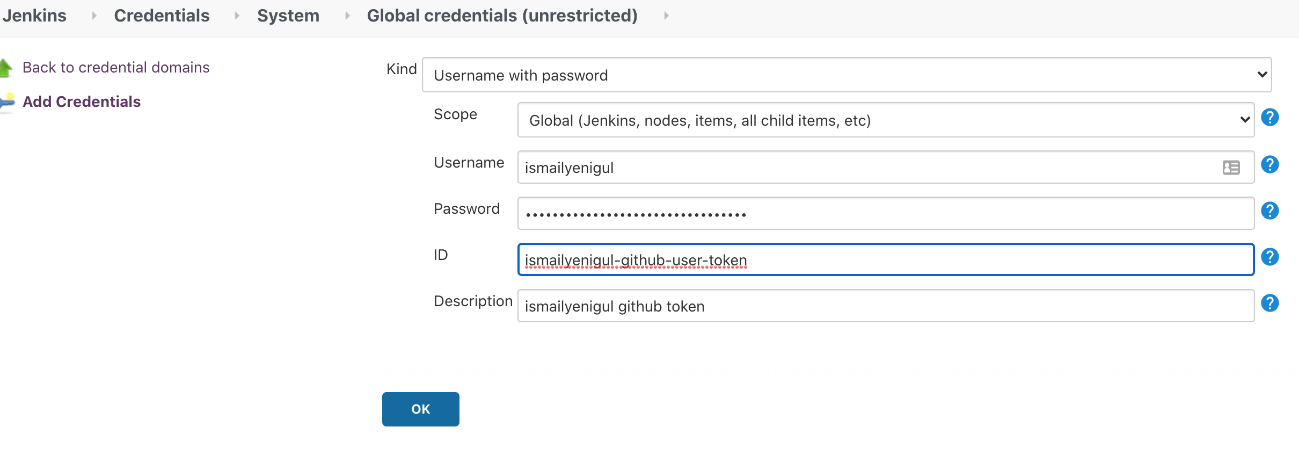


Luego haga clic en Global credentials



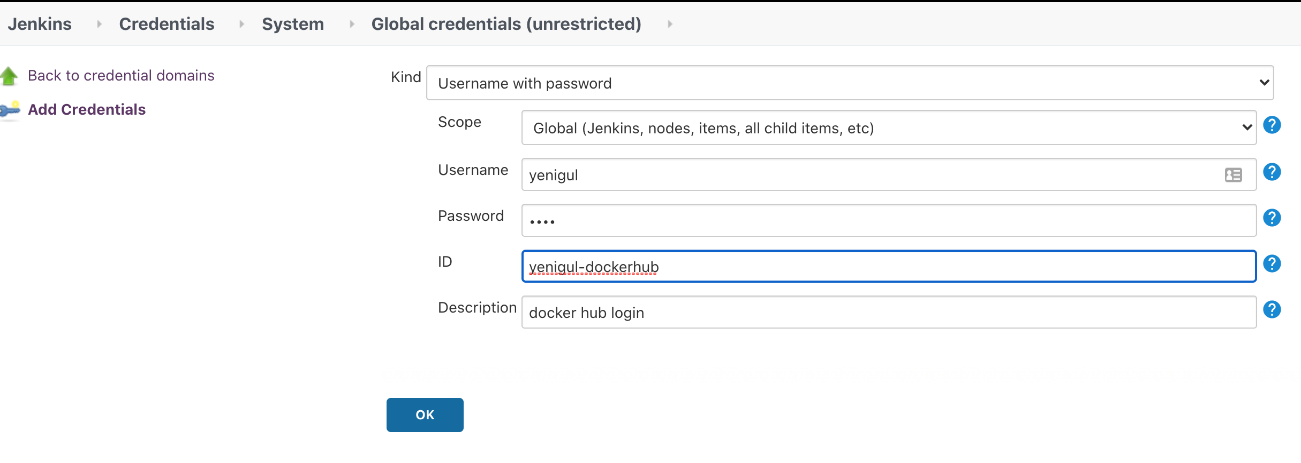
Haga clic Add Credentials en el menú de la izquierda.

Elija nombre de usuario y contraseña



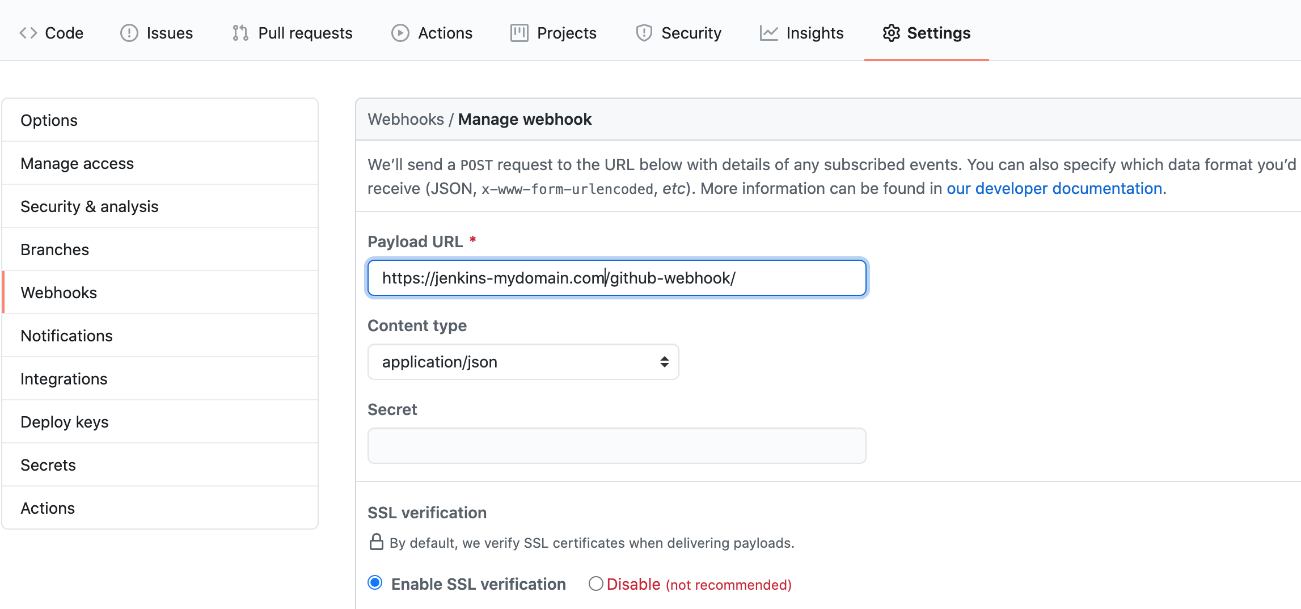
Usernamees el ID de usuario de GitHub y Passwordes el [token de API personal](https://github.com/blog/1509-personal-api-tokens)

Cree otras credenciales de usuario y contraseña para el inicio de sesión de Docker Hub.



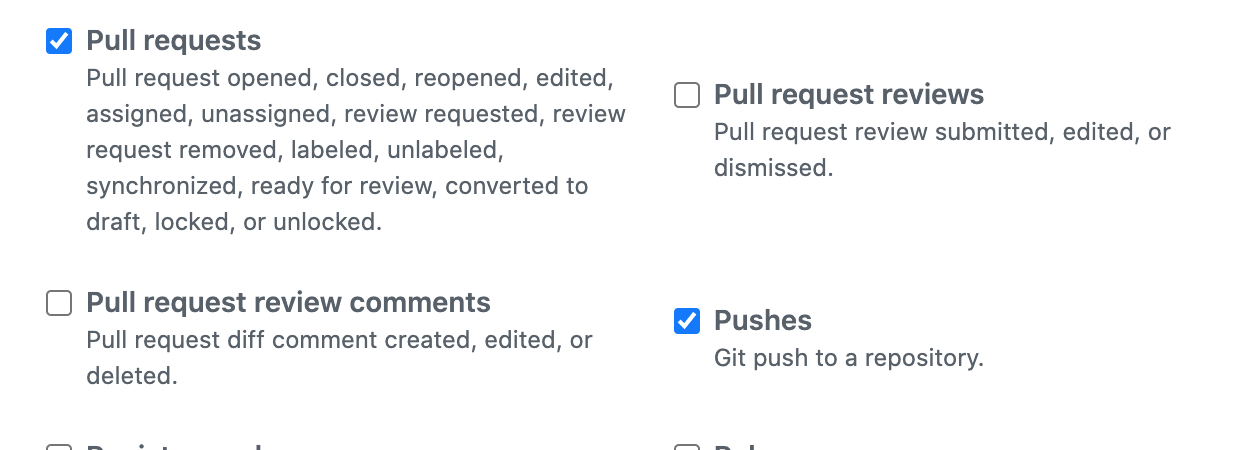
**Crear webhook en Github**

Vaya a Settings->Webhook en su repositorio y agregue



Escriba su https://jenkinsURL/github-webhook/

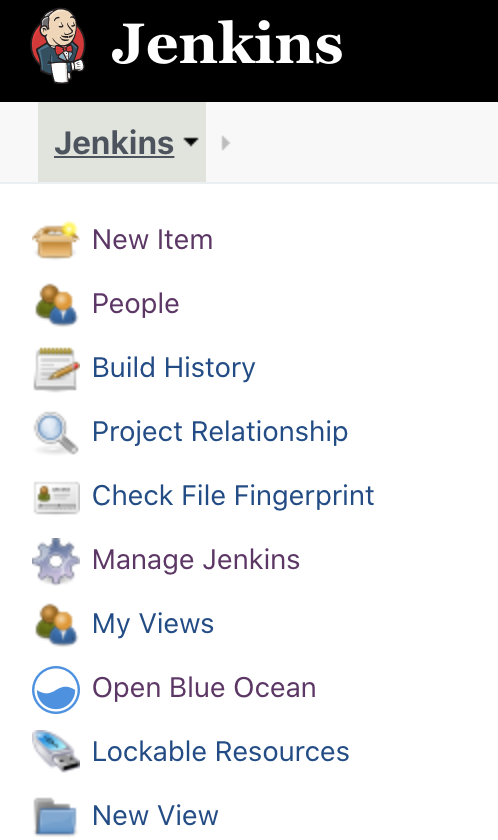
Elija Content-Type como application/jsony choose Let me select individual eventsseleccione Pull RequestsyPushes



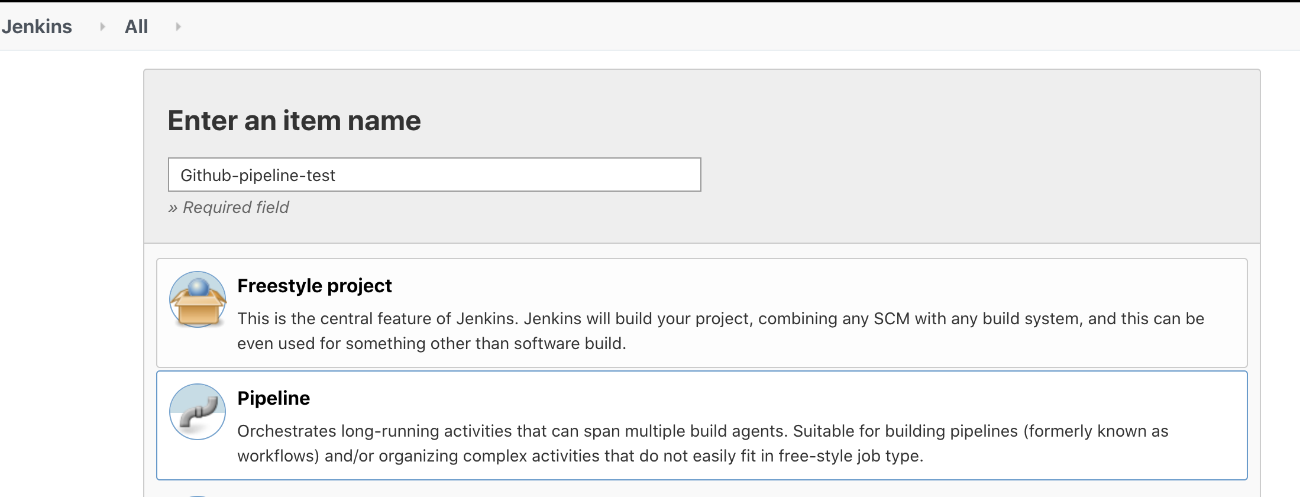
Siempre que hagas un pull o phus en el repositorio, Github informará a Jenkins. Asegúrese de que se pueda acceder a su URL de Jenkins desde Github.

**Ahora es el momento de configurar Pipeline.**

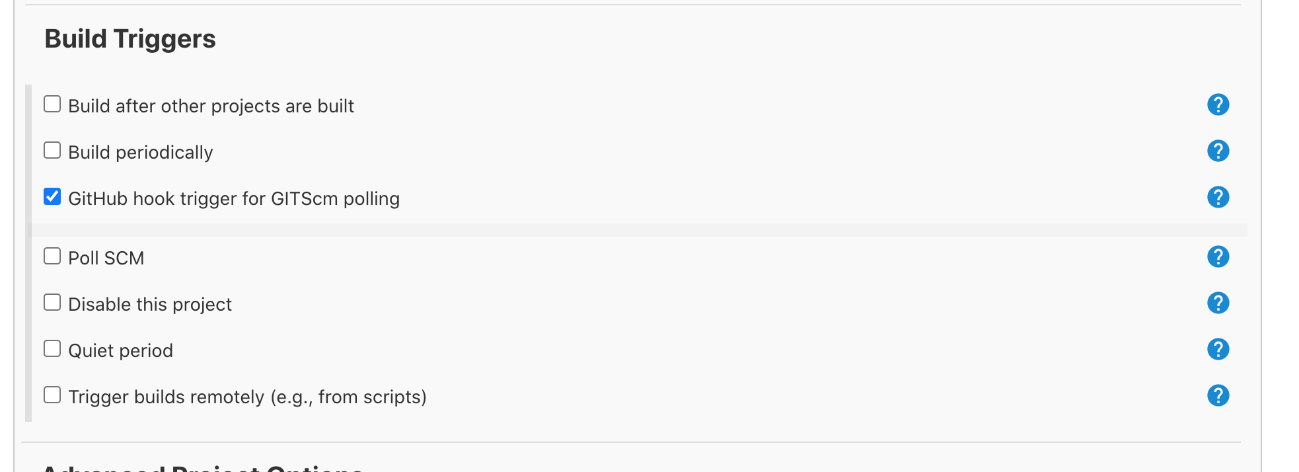
Haga clic en Nuevo elemento y seleccione canalización



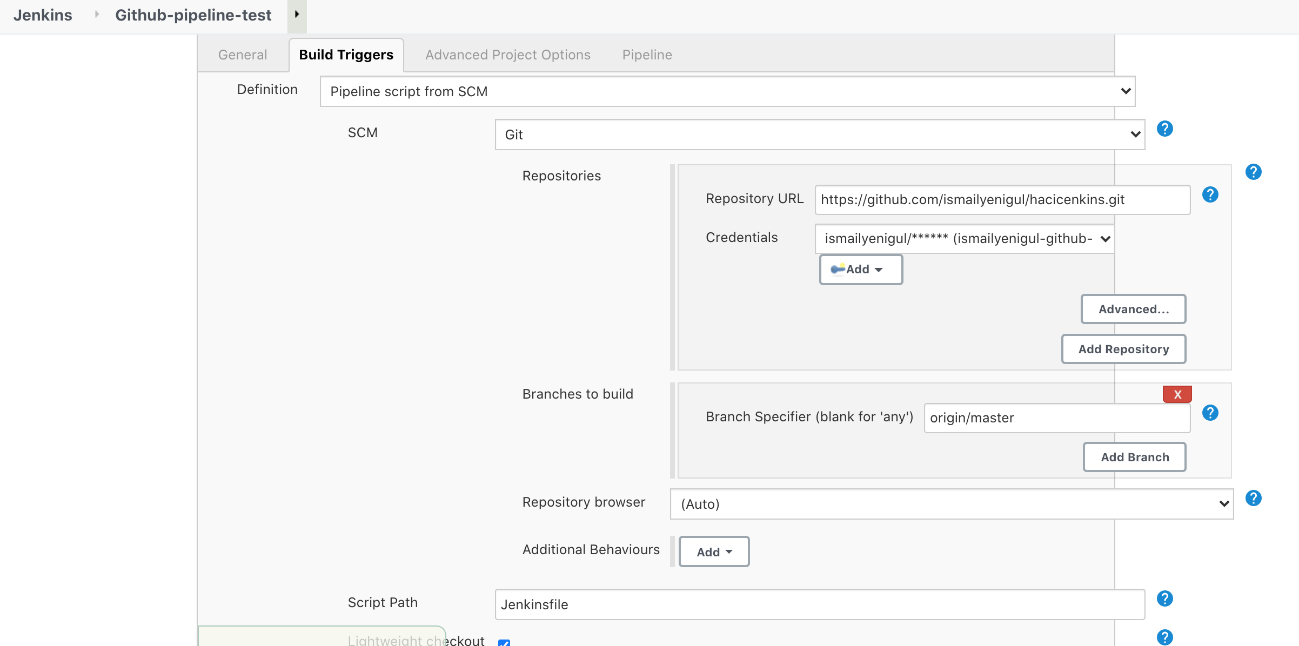
Dar un nombre y seleccionar Pipeline



Habilitar GitHub hook trigger for GITScm pollingenBuild Triggers



Y configurar la canalización



Elija Definición Pipeline script from SCM  
SCM:Git

Credenciales: la que creamos anteriormente. También puede crear las credenciales en esta sección.

Rama para construir: esto es solo una compilación de prueba, elegí el maestro origin/master

Ruta del script: dónde lo guarda Jenkinsfile. Lo guardamos en el directorio base del repositorio. Si se encuentra en la ruta secundaria, especifíquelo aquí.

aquí está el contenido del Jenkinsfile

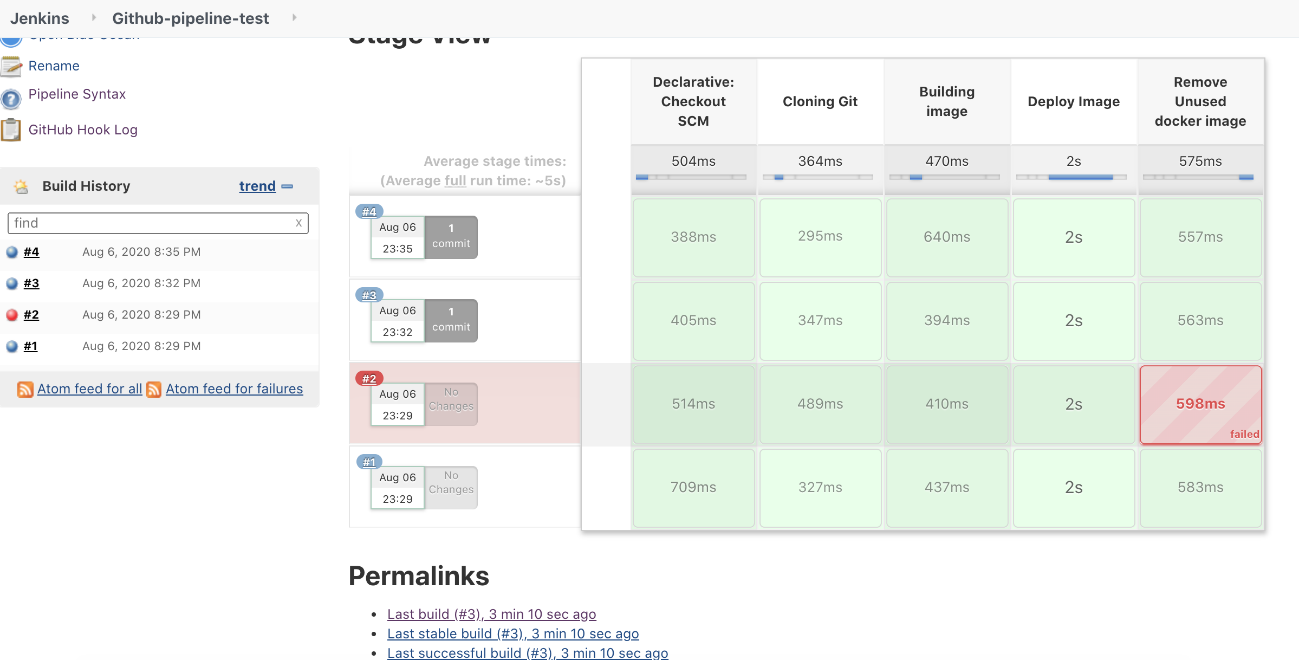


Puede descargar este archivo de gist

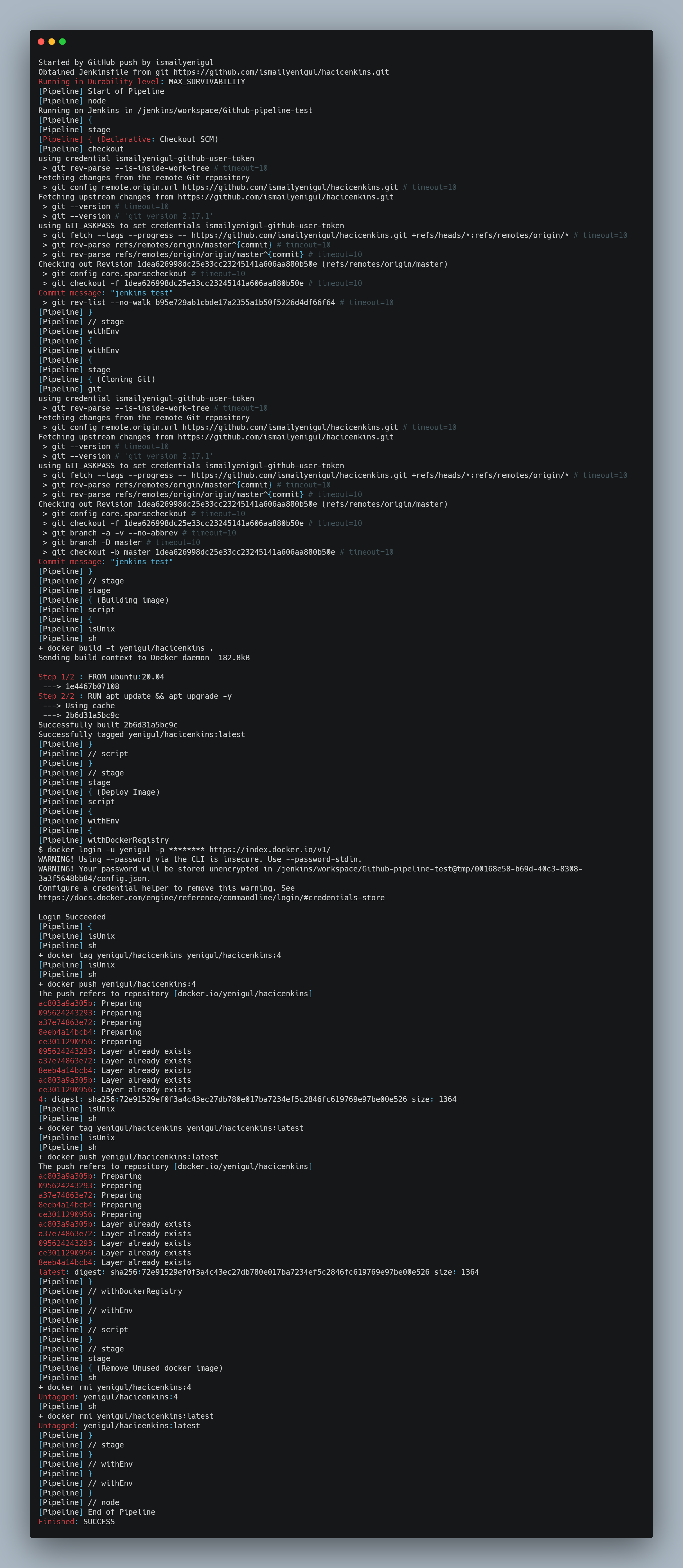


Simplemente actualice algo en el repositorio y presione los cambios

git add. ; git commit -m "prueba de Jenkins"; git push



y aquí están los registros

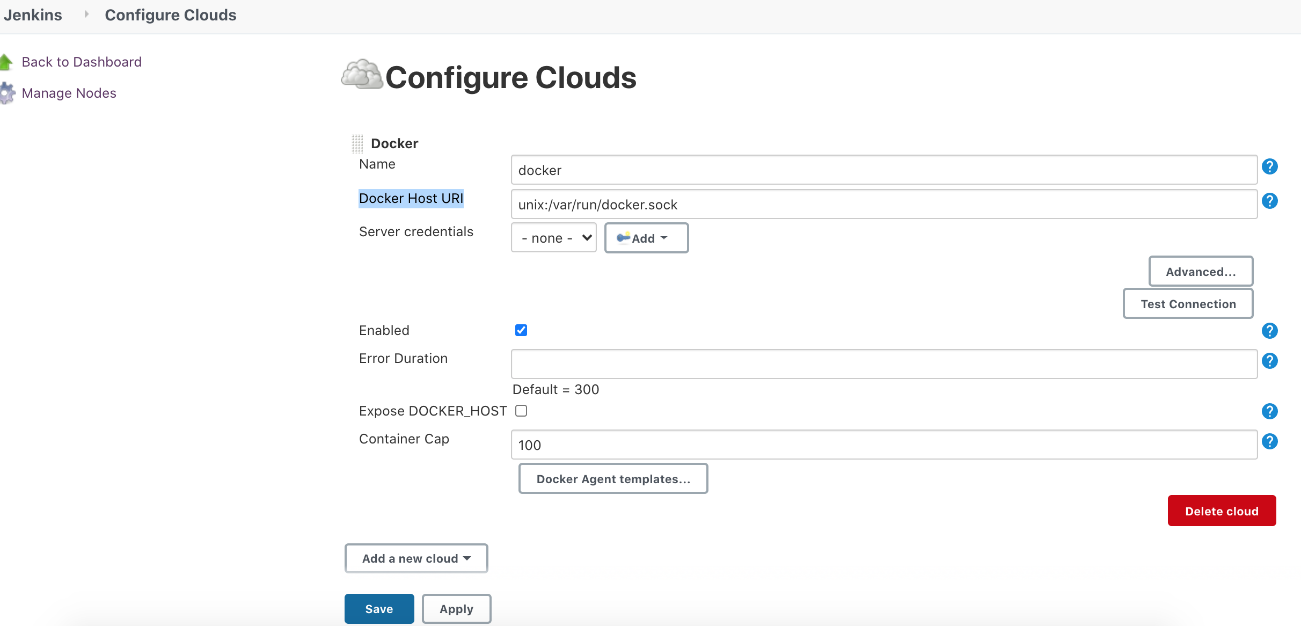


**Extra: cree una imagen de Docker con el complemento de Docker con Free Style project**

Si desea crear una imagen de Docker sin escribir una pipeline, puede crear un proyecto FreeStyle con Docker Plugin. Debe configurar Docker en la sección de nodos de Jenkins  
Ir a Manage Jenkins -> Manage Nodes and Clouds -> Configure Clouds-> Add New Cloud.

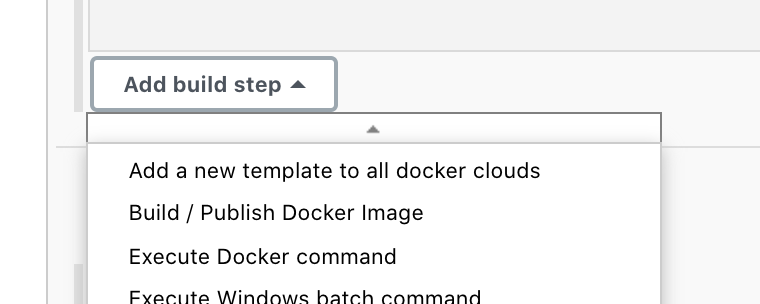
Seleccione **Docker** aquí.

Escribir unix:///var/run/docker.sockpara Docker Host URIy haga clic en Enabled A continuación, guarde

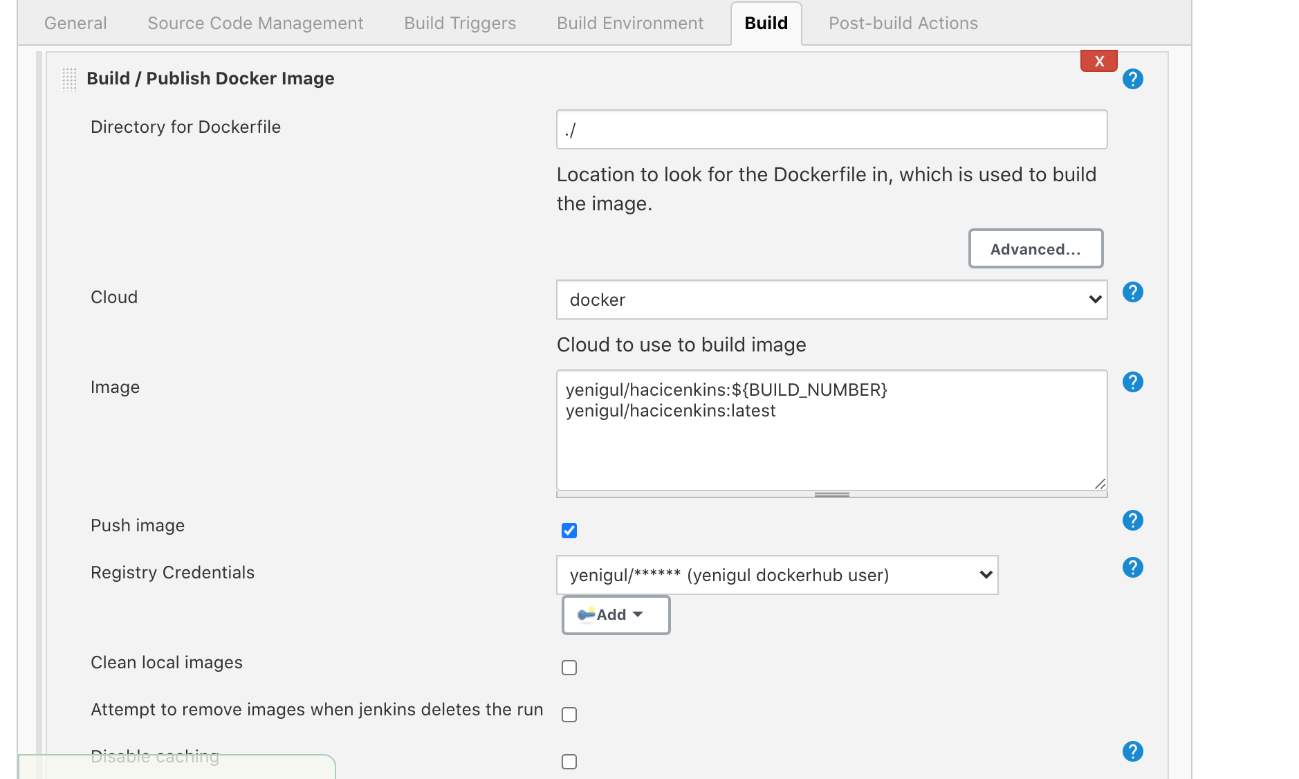


Y aquí está la especificación de compilación para el proyecto Free Style

En Add Build Step Seleccionar Build / Publish Docker Image



y llena el formulario



Aquí hay otro ejemplo si desea enviar imágenes de Docker al repositorio de AWS ECR.

[Debe](https://plugins.jenkins.io/amazon-ecr/) instalar el [complemento aws ecr](https://plugins.jenkins.io/amazon-ecr/) y crear nuevas credenciales con AWS Credentials kind (ismail-ecr en mis pruebas)

