



Proyecto de Software — Docker, Git, Jenkins

Deyssi Catalina Arango Bermudez | dcaarango@poligran.edu.co

Johan Sebastian Gordillo Patiño | jsgordillo@poligran.edu.co

Luis Gerardo Salas Alfonso | lsalas@poligran.edu.co

Ramón Barrios Láscar | elbarrios1@poligran.edu.co

Manuel Alberto Coy | macoyben@poligran.edu.co

Politécnico Grancolombiano

Integración Continua

Ing. Óscar Vargas

22 de noviembre de 2022

Parte 1 – Semana 3

1.1. Descripción del proyecto

Realizar una página web, que permita visualizar las APIs dispuestas en el Front-end, iniciando por un login que permita hacer la consulta respectiva a la Base de Datos, y así obtener la información necesaria para realizar el ingreso a la interfaz.

1.2. Roles en el equipo

Base de Datos – Infraestructura

- Ramón Barrios Láscar.
- Manuel Alberto Coy.

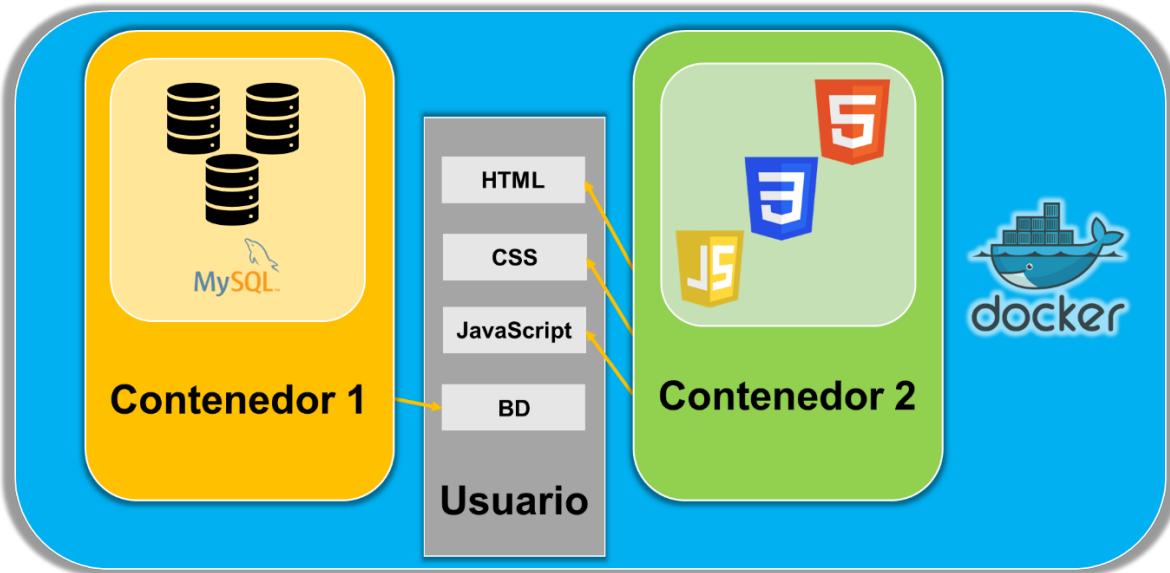
Diseño – Front end

- Deyssi Catalina Arango Bermudez.
- Johan Sebastian Gordillo Patiño.
- Luis Gerardo Salas Alfonso.

1.3. Lenguajes seleccionados

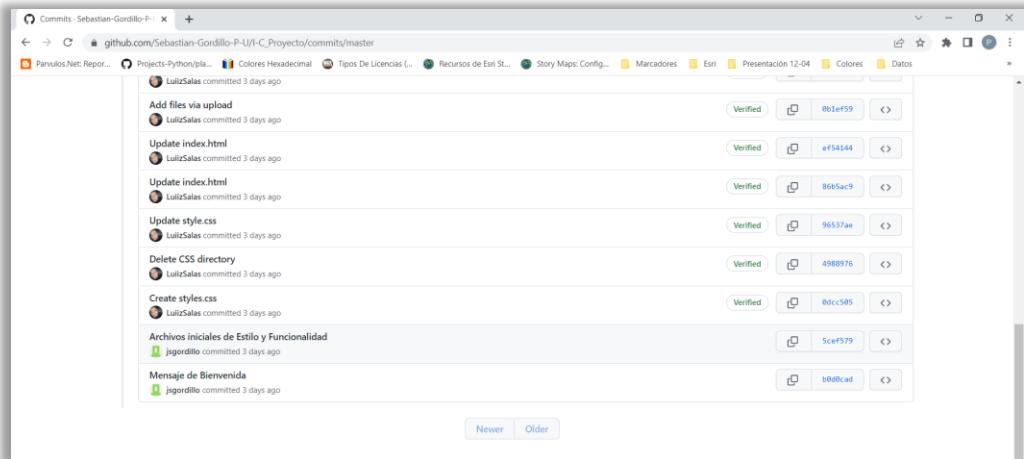
- MySql
- HTML
- CSS
- JavaScript

1.4. Diagrama de arquitectura general

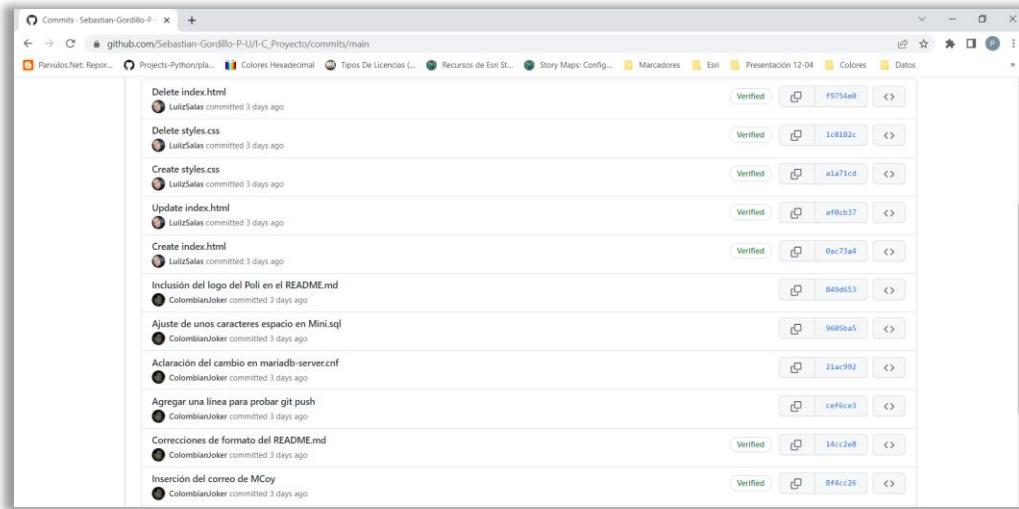


Nota. Elaboración propia (2022).

1.5. Enlace a Repositorios



Nota. Elaboración propia (2022).



Nota. Elaboración propia (2022).

The screenshot shows the 'I-C_Proyecto' repository on GitHub. The 'main' branch is selected, and the 'docker-compose.yaml' file is displayed. The file content is as follows:

```
version: "3"
services:
  node:
    image: bitnami/node
    container_name: node
    links:
      - mariadb
  shell:
    image: ubuntu
    stdin_open: true
    tty: true
    container_name: shell
    links:
      - mariadb
  mariadb:
    image: mariadb
    container_name: mariadb
    ports:
      - 4407:3306
    volumes:
      - type: volume
        source: dbdata
        target: /var/lib/mysql
      - volume:
          nocopy: true
        type: bind
```

Nota. Elaboración propia (2022).

Enlace: https://github.com/Sebastian-Gordillo-P-U/I-C_Proyecto

1.6. Enlaces a los Videos

Se entregan dos enlaces, contiene el mismo video, tener en cuenta que si uno falla el otro esta de apoyo.

Enlace YouTube 1

<https://www.youtube.com/watch?v=ViasfxkCwPM>

Enlace YouTube 2

https://www.youtube.com/watch?v=Z_uoamMdY7A

Parte 2 – Semana 5

Tabla de contenido, Parte 2

DEVOPS

1. Qué es Jenkins y para que se utiliza
2. Explicación de la instalación de Jenkins
 - 2a. Explicación de la instalación de Jenkins en WINDOWS
 - 2b. Explicación de la instalación de Jenkins en DOCKER
3. Explicación de la instalación de Jenkins con GITHUB
4. Referencias

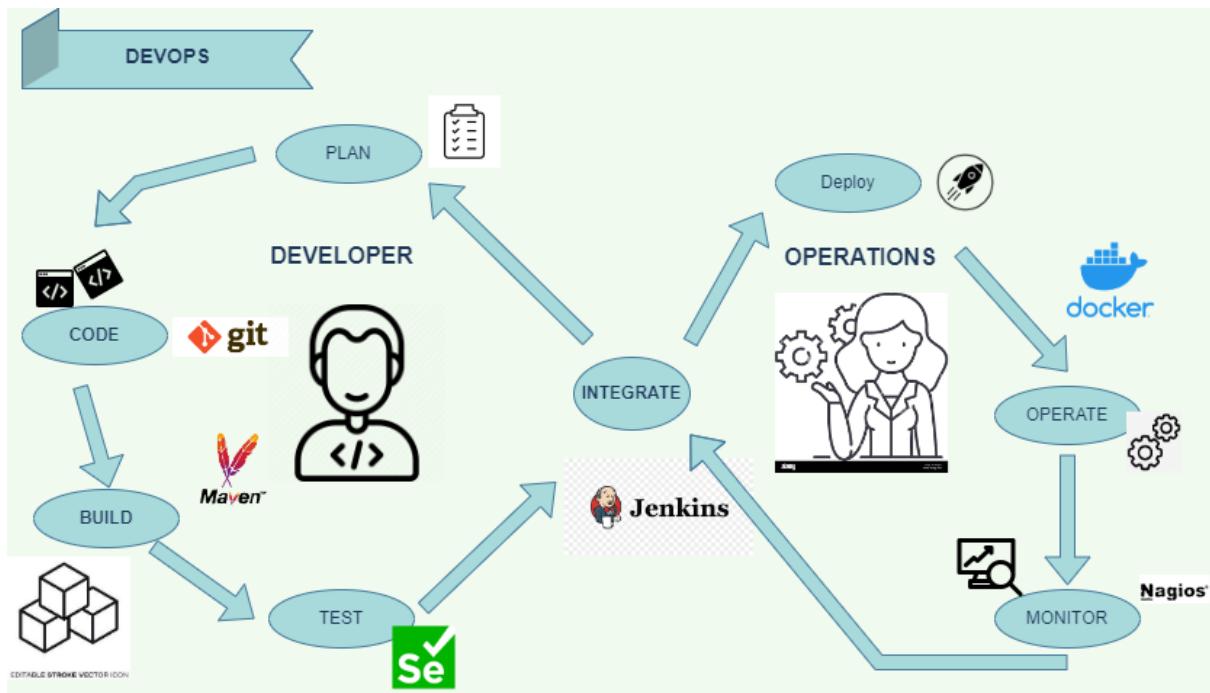


Imagen 1 Diseño Propio

DEVOPS

El término DevOps, que es una combinación de los términos ingleses development (desarrollo) y operations (operaciones), designa la unión de personas, procesos y tecnología para ofrecer valor a los clientes de forma constante.

¿Qué significa DevOps para los equipos? DevOps permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para producir productos mejores y más confiables. Al adoptar una cultura de DevOps junto con prácticas y herramientas de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.

Ventajas de DevOps

Los equipos que adoptan la cultura, las prácticas y las herramientas de DevOps mejoran el rendimiento y crean productos de más calidad en menos tiempo, lo que aumenta la satisfacción de los clientes. Esta mejora de la colaboración y la productividad es fundamental también para alcanzar objetivos de negocio como estos:

- Reducción del tiempo de comercialización
- Adaptación al mercado y a la competencia

- Mantenimiento de la estabilidad y la confiabilidad del sistema
- Mejora del tiempo medio de recuperación

DevOps y el ciclo de vida de las aplicaciones

DevOps influye en el ciclo de vida de las aplicaciones a lo largo de las fases de planificación, desarrollo, entrega y uso. Cada fase depende de las demás y las fases no son específicas de un rol. En una auténtica cultura de DevOps, todos los roles están implicados de algún modo en todas las fases.

- Planificación
- Desarrollo
- Entrega
- Uso

Cultura de DevOps

Aunque la adopción de prácticas de DevOps automatiza y optimiza los procesos con tecnología, todo comienza con la cultura interna de la organización y con las personas que participan en ella. El desafío de cultivar una cultura de DevOps requiere cambios profundos en la forma en la que las personas trabajan y colaboran. Pero cuando las organizaciones se comprometen a implementar una cultura de DevOps, pueden crear un entorno que facilite el desarrollo de equipos de alto rendimiento.

2.1. Que es Jenkins y para qué se utiliza ¿Qué es Jenkins?

Jenkins es la herramienta más utilizada para la integración continua, es un servidor escrito en Java que permite desplegar proyectos de software y facilitar mejoras al mismo.

Para ello, la herramienta supervisa de manera periódica el repositorio donde se encuentra el código fuente de un programa y si detecta algún cambio procede a compilar y preparar un ejecutable, si hay falla notifica y sino realiza el despliegue en el servidor de pruebas; posteriormente notifica de nuevo los resultados de estas últimas.

Es el software más usado para entregar nuevas versiones a los usuarios.

Caso expuesto de Jenkins

Problemática

En el sector de la tecnología se requiere una página web que permite hacer el consumo de unas series de APIs, que los usuarios requieren para hacer investigaciones y también les gustaría integrar temas de interés, como programas de TV, series y entretenimiento. Además, informan que no tiene el panorama completo de las APIs que se quieren usar.

Solución

Para esto el líder del proyecto decide proponer la integración continua con la herramienta de Jenkins, explica como sería el funcionamiento con de la herramienta en el proyecto. De la siguiente manera:

Equipo de Desarrollo

Realizara cambios constantes es su desarrollo para la inclusión de las APIs en momento específicos del proyecto. “Cuando se solicite la inclusión de una nueva APIs por parte de las personas que requieren el proyecto”. Contemplando esto, la herramienta a utilizar es Jenkins como se explicó anteriormente.



Jenkins

Conclusión

Para esto se piensa hacer la integración continua, tanto para el proyecto de los desarrolladores, como para la entrega del proyecto. Realizando el uso de la herramienta. Tenemos la ventaja de mayor rendimiento en cada parte del proyecto, además que los usuarios podrán tener lo más pronto posible lo solicitado en la página web.

2.2a. Explicación de la instalación de Jenkins en Windows

1. Ingresamos al navegador de preferencia que más utilicemos. Y seleccionamos el buscador Google para realizar la respectiva búsqueda.



2. Ingresamos en el buscador Jenkins Download para la respectiva descarga.

A screenshot of a Google search results page. The search query 'jenkins download' is entered in the search bar. The results show a snippet for 'TeamCity: Jenkins Alternative - Fast CI/CD for Modern Teams' with a link to 'https://www.jetbrains.com/ci_cd/teamcity/'. Below this, there are sections for 'Download', 'NET Support', 'Configuration', and 'Features'. The 'Download' section includes a link to 'Start using TeamCity under the free Professional license'. The 'NET Support' section mentions 'NET testing frameworks, NuGet, MSBuild, NAnt, .NET Core, and other'. The 'Configuration' section talks about 'Projects Hierarchy, Meta-Runner, Build Chains and Build Dependencies'. The 'Features' section lists 'Testing frameworks support, code coverage, code analysis, and more'.

- Cuando se termina la búsqueda en el navegador seleccionamos el siguiente enlace para su respectiva información y su descarga.

The screenshot shows the Jenkins download page at <https://www.jenkins.io>. A search bar at the top contains the query "jenkins download windows". Below the search bar, the heading "Jenkins download and deployment" is visible. A section titled "Otras personas también buscaron" lists related queries: "jenkins download windows", "jenkins install", "jenkins login", "jenkins tutorial", "jenkins download mac", and "jenkins war file download".

- Al momento de ingresar a la página observamos dos opciones de Jenkins que nos ofrecen y son los siguientes que son Stable (LTS) y Regular releases (Weekly), con la opción de escoger el sistema operativo que sea compatible con nuestra máquina que sería Windows.

The screenshot shows the Jenkins download page with two main sections: "Estable (LTS)" and "Lanzamientos regulares (semanales)". The "Estable (LTS)" section is highlighted with a red circle. It contains a brief description and links to "Registro de cambios", "Guía de actualización", and "Lanzamientos anteriores". The "Lanzamientos regulares (semanales)" section also contains a brief description and links to "Registro de cambios" and "Lanzamientos anteriores".

- Se inicia la descarga, y se descarga Jenkins LTS. Porque tiene soporte a largo plazo y es más factible para nuestro proyecto y se procede a su instalación.

The screenshot shows the Jenkins download page for Jenkins 2.361.4 LTS. It displays a list of operating systems for which Jenkins can be downloaded as a "Generic Java package (.war)". The "Windows" option is circled in red. Both the left and right panels show the same list of operating systems.

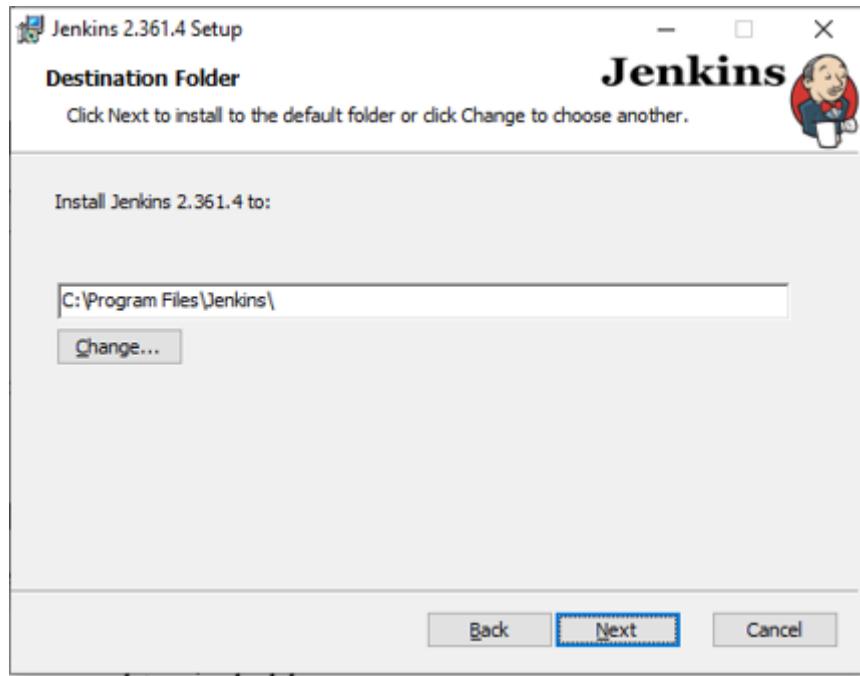
| Download Jenkins 2.361.4 LTS for: | Download Jenkins 2.378 for: |
|---|--|
| Generic Java package (.war) SHA-256: b39fe218afb5d47b0c9a0fa308d7ab762ac5a58dd9aa58b735067addc37c17b | Generic Java package (.war) SHA-256: 07da0fc2ca197160d052191fea7bde72acd8b099b057bcd4551d61b8c44be6de |
| Docker | Docker |
| Ubuntu/Debian | Ubuntu/Debian |
| CentOS/Fedora/Red Hat | CentOS/Fedora/Red Hat |
| Windows | Windows |
| openSUSE | openSUSE |
| FreeBSD | Arch Linux |
| Gentoo | FreeBSD |
| macOS | Gentoo |
| OpenBSD | macOS |

jenkins.msi

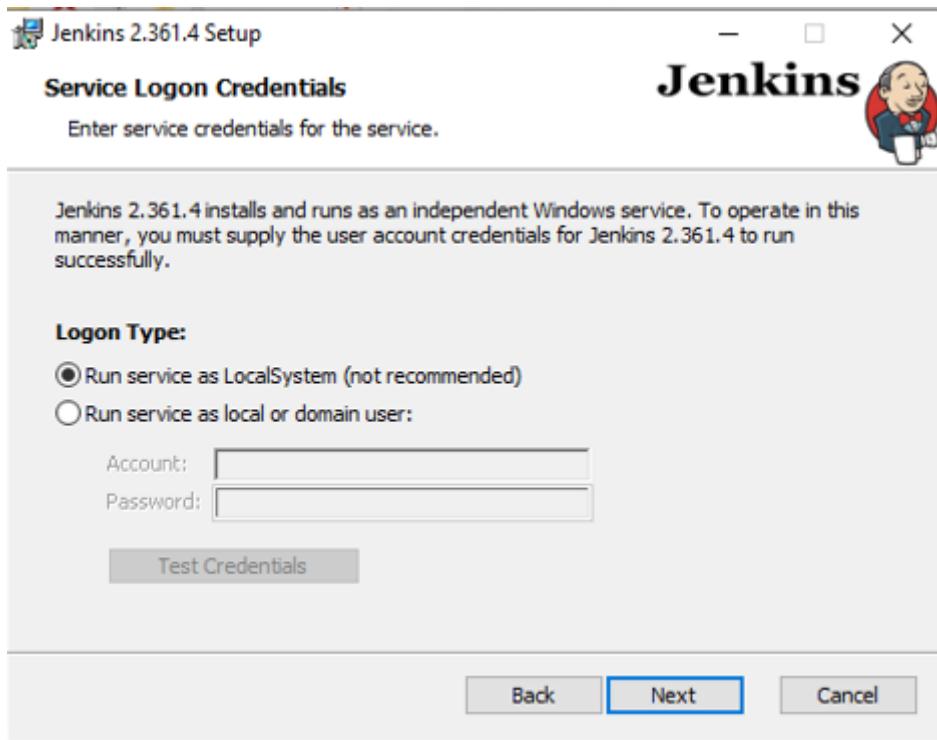
6. Iniciamos con la instalación como administrador y se visualiza la siguiente ventana y se da (Next) siguiente



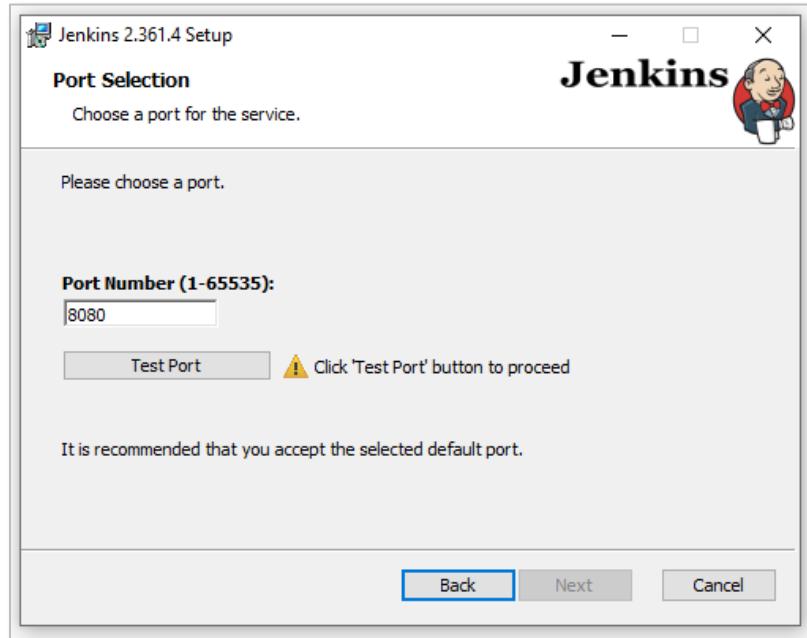
7. En la siguiente ventana seleccionamos la ruta que deseamos tener para Jenkins o lo podemos dejar como predeterminado del programa



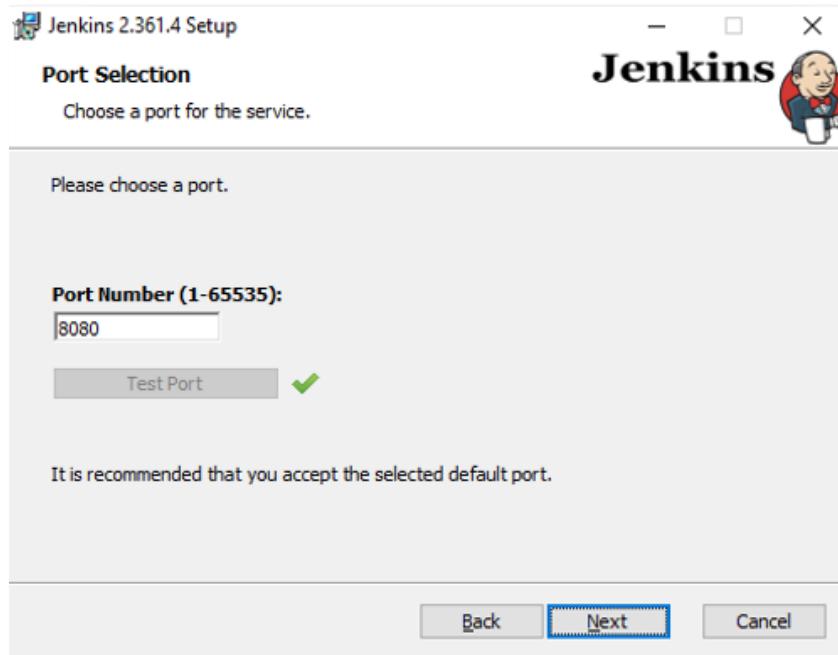
8. En la siguiente imagen podemos observar que tenemos dos opciones en la instalación la primera si ya se tiene un servicio local se usa la segunda opción en caso de no tenerlo se puede conectar con la primera opción.



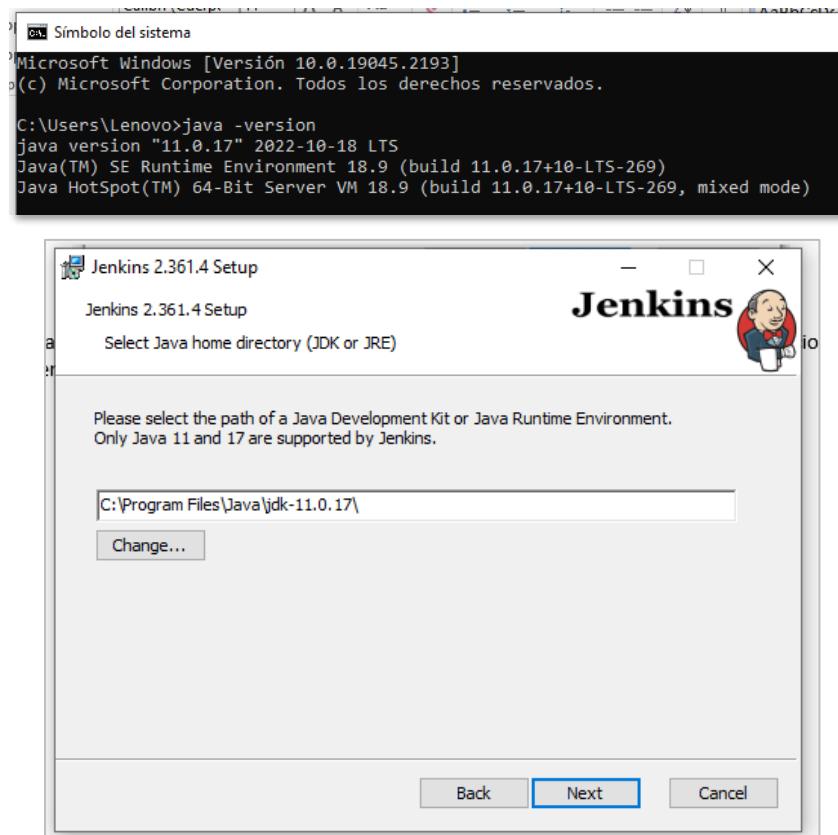
9. En la siguiente imagen podemos asignar el puerto que deseamos utilizar y por defecto viene pre terminado en 8080 seleccionamos test port si se encuentra disponible



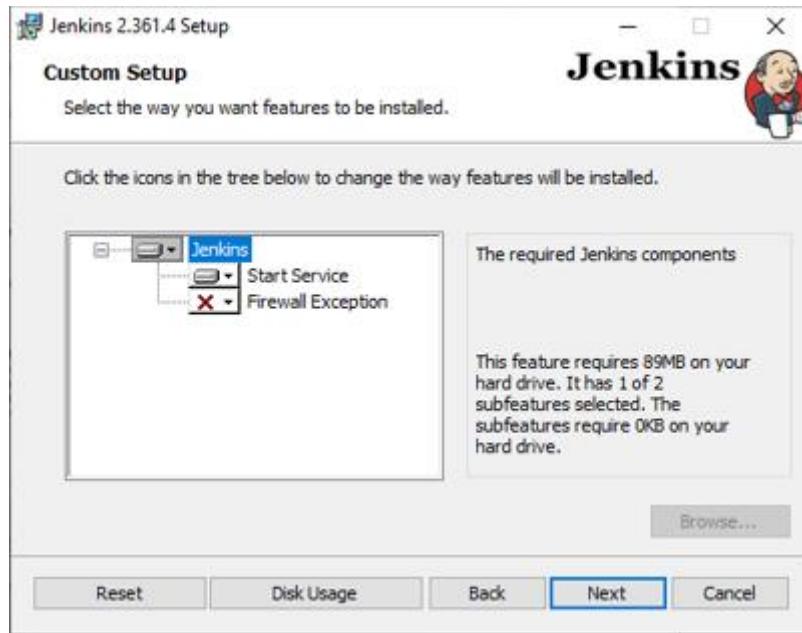
10. Cuando ya tenemos referenciado nuestro puerto seleccionamos (Next) siguiente para continuar con nuestra descarga



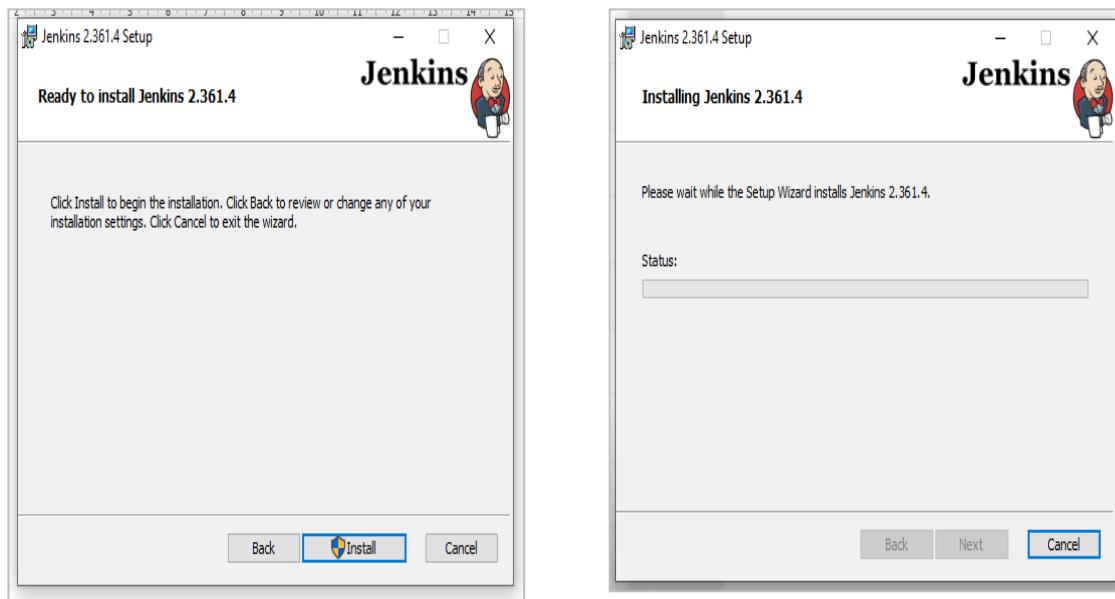
11. Para el siguiente paso debemos tener java 11 o 17 para poder continuar con las instalación de Jenkins en el equipo, en esta parte ya lo tenemos instalado y podemos seguir con el siguiente paso de la instalación, para saber que versión tenemos de java ingresamos a CMD e ingresamos este comando `java -version`



12. En el siguiente paso, se activa lo siguiente Firewall que es requerido para la instalación de Jenkins.



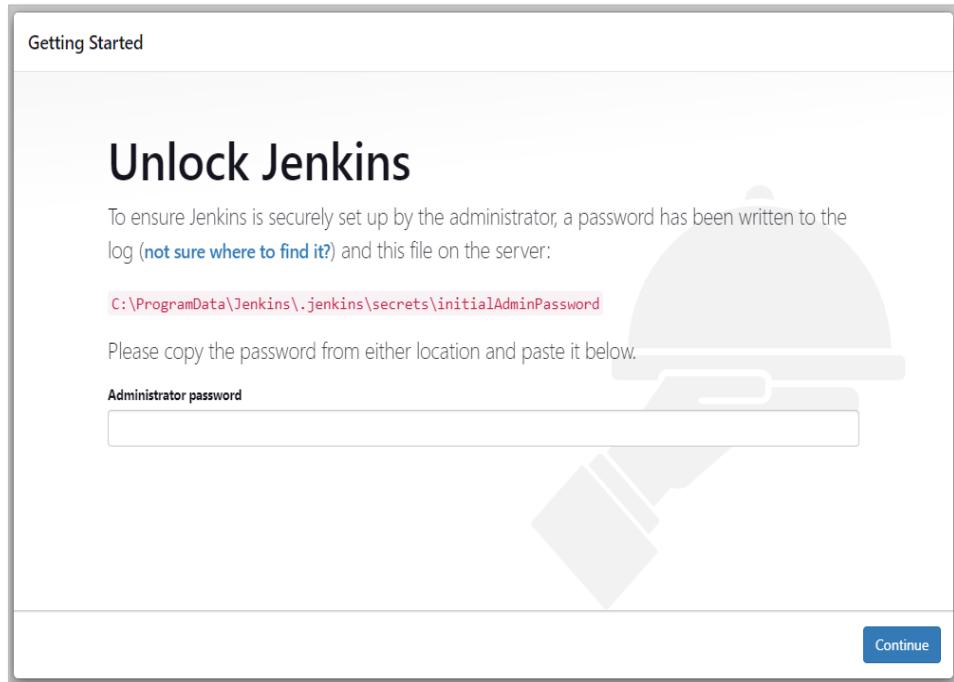
13. En la siguiente imagen ya se procede a la instalación del Jenkins 2.361.4 para nuestra entrega



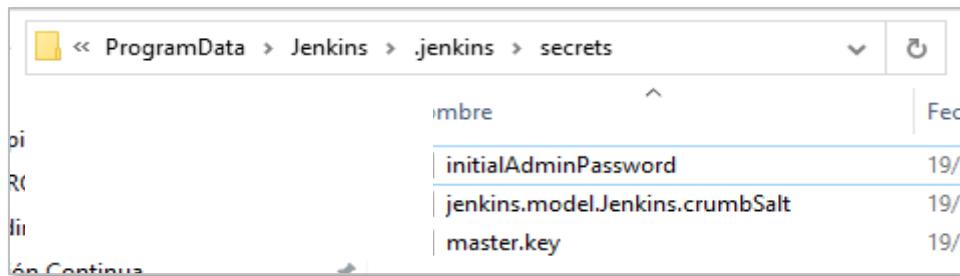
14. Se finaliza la instalación del programa Jenkins



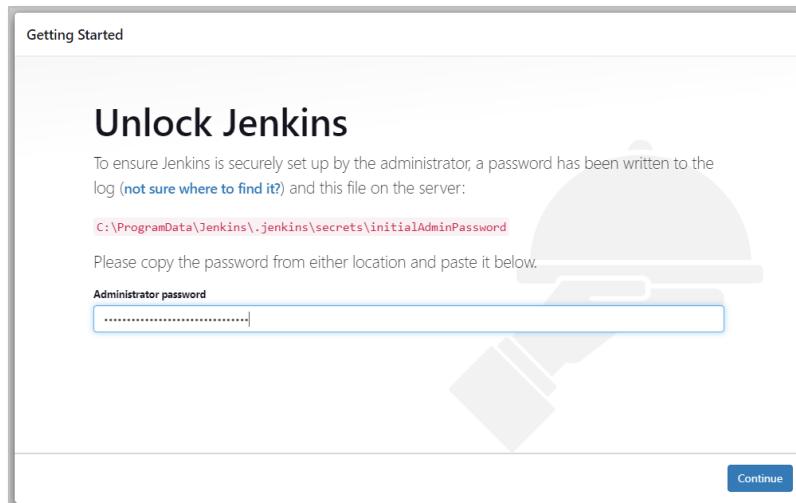
15. En la siguiente imagen observamos el buen funcionamiento de Jenkins en nuestro equipo, siguiendo con la instalación procedemos a cambiar el Password de la aplicación como es indicada en la imagen con el siguiente enlace en archivos del equipo
`C:\ProgramData\Jenkins\.jenkins\secrets`



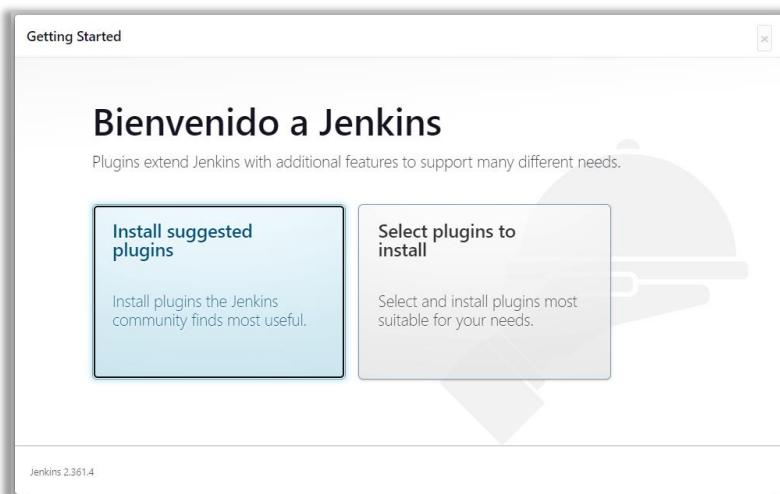
16. Cuando ya ingresamos en la carpeta de archivos, buscamos con el enlace que nos suministra la página el siguiente documento que se llama initialAdminPassword hay visualizamos el Password de la página y así poder acceder a Jenkins.



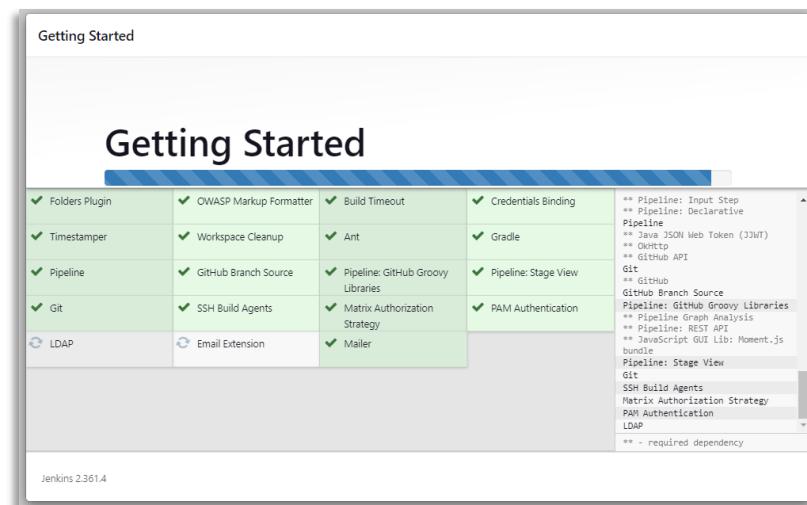
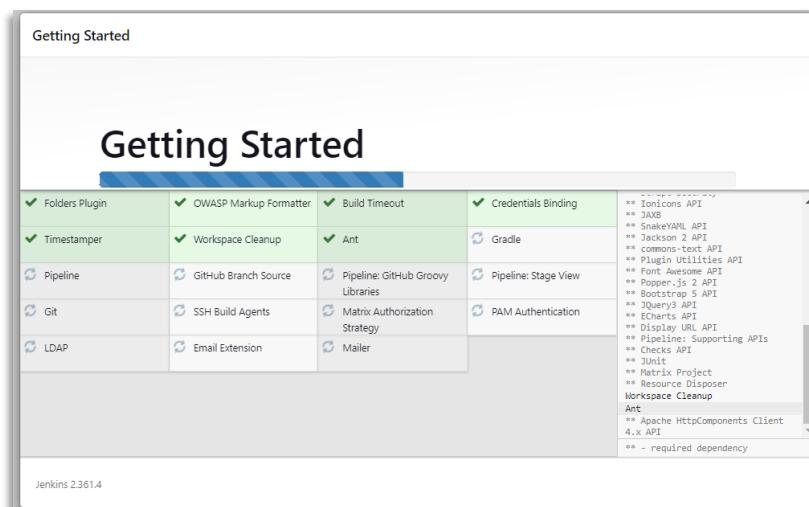
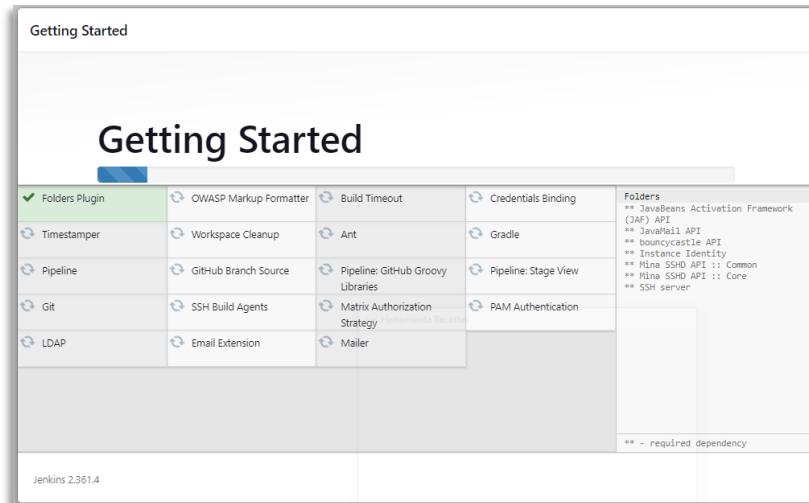
17. Cuando ya tengamos el Password ya podemos ingresar a Jenkins



18. En el siguiente paso ya es decisión de cada persona en instalar plugins recomendados o seleccionar los Plugins específicos, para esta instalación seleccionamos los plugins recomendados para nuestra instalación genérica



19. Despues de seleccionar la instalación de plugins recomendados esperamos q termine su actualización



20. El paso siguiente nos solicita crear un usuario para poder iniciar sesión cuando queramos utilizar Jenkins, cuando se ingrese los datos solicitados, damos continuar con el siguiente paso

Getting Started

Create First Admin User

Usuario:

Contraseña:

Confirma la contraseña:

Nombre completo:

Dirección de email:

Jenkins 2.361.4 Skip and continue as admin Save and Continue

Getting Started

Create First Admin User

Usuario:

Contraseña:

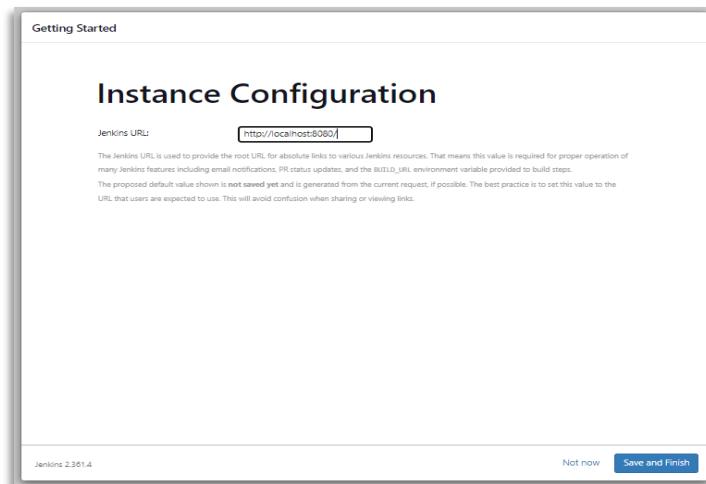
Confirma la contraseña:

Nombre completo:

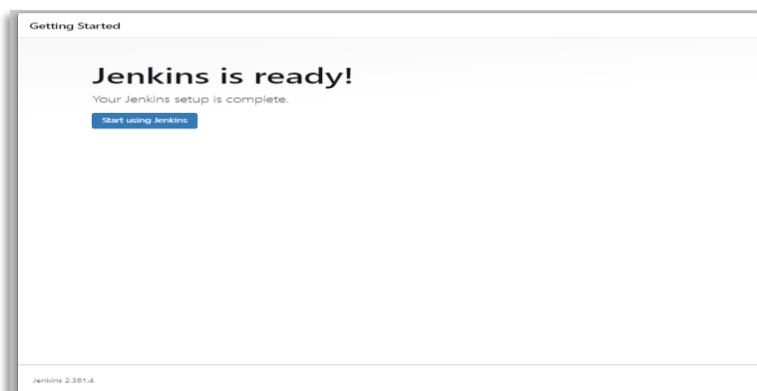
Dirección de email:

Jenkins 2.361.4 Skip and continue as admin Save and Continue

21. A continuación, se confirma el puerto que se va a utilizar, si lo deseamos cambiar lo podríamos hacer, pero para mejor actividad se deja igual como se indicó al principio de la instalación.

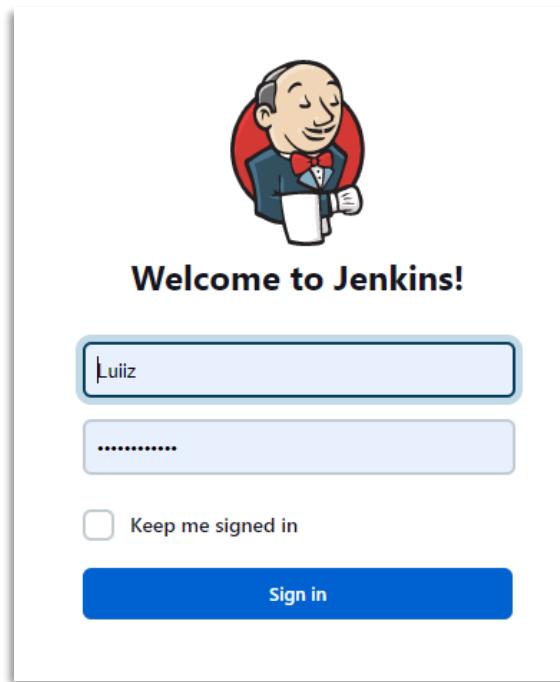


22. Ya podemos utilizar Jenkins



23. Iniciamos la prueba para verificar su funcionamiento, y damos como éxito el funcionamiento de Jenkins en nuestro equipo de computo

ⓘ localhost:8080/login?from=%2F

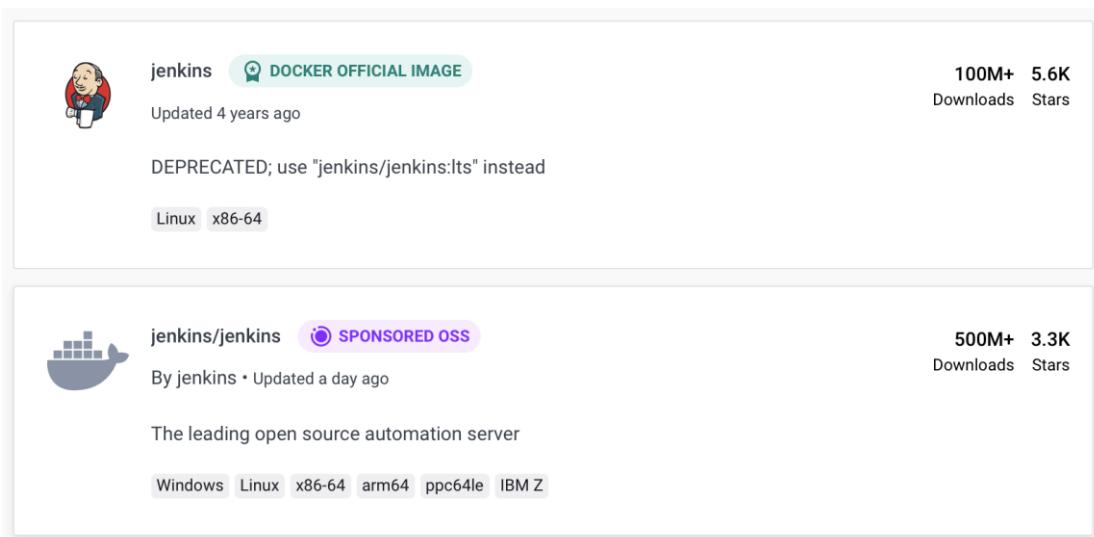


The Jenkins login screen features a placeholder for a user profile picture, a 'Welcome to Jenkins!' message, and a sign-in form.

The Jenkins dashboard is shown below, featuring a sidebar with links like 'Nueva Tarea', 'Personas', 'Historial de trabajos', 'Administrar Jenkins', and 'Mis vistas'. The main area displays a '¡Bienvenido a Jenkins!' message, instructions for starting builds, and sections for managing distributed builds and agents.

2.2.b Explicación de la instalación de Jenkins en Docker

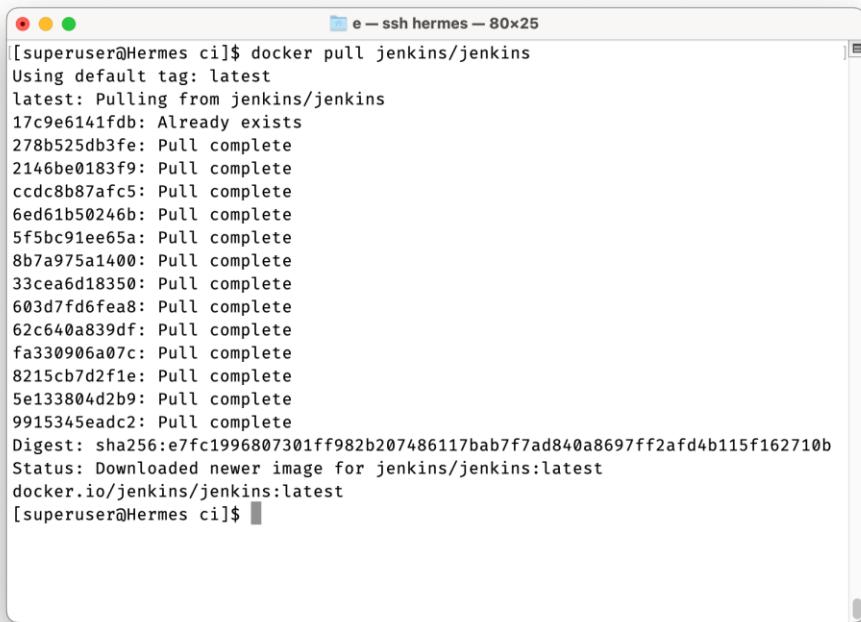
- Si se desea instalar Jenkins como un contenedor Docker, hay dos opciones reales. Una, la imagen "Jenkins" oficial "Docker Official Image" ha sido despreciada y se recomienda usar "Jenkins/Jenkins". En este laboratorio seleccionamos esta última.



2. Una de las maneras de descargar la imagen es usando "docker pull Jenkins/Jenkins" ("docker run Jenkins/Jenkins" sin tener la imagen también la descarga)

```
e -- ssh hermes - 80x25
[superuser@Hermes ci]$ docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
17c9e6141fdb: Already exists
278b525db3fe: Downloading 507.9kB/50.68MB
2146be0183f9: Downloading 572.8kB/8.655MB
ccdc8b87afc5: Download complete
6ed61b50246b: Waiting
5f5bc91ee65a: Waiting
8b7a975a1400: Waiting
33cea6d18350: Waiting
603d7fd6fea8: Waiting
62c640a839df: Waiting
fa330906a07c: Waiting
8215cb7d2f1e: Waiting
5e133804d2b9: Waiting
9915345eadc2: Waiting
```

3. Si no se escoge otro tag "docker pull Jenkins/Jenkins" descarga la última versión. Nótese que la imagen Jenkins/jenkins:latest (la última) es creada automáticamente con Jenkins por el grupo que lo desarrolla.



```
[superuser@Hermes ci]$ docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
17c9e6141fdb: Already exists
278b525db3fe: Pull complete
2146be0183f9: Pull complete
ccdc8b87afc5: Pull complete
6ed61b50246b: Pull complete
5f5bc91ee65a: Pull complete
8b7a975a1400: Pull complete
33cea6d18350: Pull complete
603d7fd6fea8: Pull complete
62c640a839df: Pull complete
fa330906a07c: Pull complete
8215cb7d2f1e: Pull complete
5e133804d2b9: Pull complete
9915345eadc2: Pull complete
Digest: sha256:e7fc1996807301ff982b207486117bab7f7ad840a8697ff2af4b115f162710b
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
[superuser@Hermes ci]$
```

4. Jenkins es un servidor web (hecho en Java), y la imagen que publican automáticamente en Docker Hub:
 - Corre usando /var/jenkins_home como directorio de configuración y datos.
 - Funciona en el puerto TCP/8080.

En nuestra instalación se creó un directorio que se publicó como volumen /var/jenkins_home, así borrar la imagen o reemplazarla (en docker actualizar desde docker hub borra y re-crea la imagen) mantiene los datos en un directorio por fuera del contenedor. En nuestra instalación hacemos que el puerto TCP/8080 de Jenkins se vea como público TCP/1080 en el servidor Docker. También aprovechamos y le dimos un nombre al contenedor (para que no le de uno al azar), y que Jenkins se reconozca a sí mismo como hostname Jenkins.

```
ccdc8b87afc5: Pull complete
6ed61b50246b: Pull complete
5f5bc91ee65a: Pull complete
8b7a975a1400: Pull complete
33cea6d18350: Pull complete
603d7fd6fea8: Pull complete
62c640a839df: Pull complete
fa330906a07c: Pull complete
8215cb7d2f1e: Pull complete
5e133804d2b9: Pull complete
9915345eadc2: Pull complete
Digest: sha256:e7fc1996807301ff982b207486117bab7f7ad840a8697ff2af4b115f162710b
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
[[superuser@Hermes ci]$ ls -l
total 32
drwxrwxrwx 2 superuser everyone 4096 2022-11-18 10:12 app/
drwxrwxr-x 6      999      999 4096 2022-11-17 10:00 dbdata/
-rw-rw-r-- 1 superuser everyone 891 2022-11-17 10:03 docker-compose.yaml
drwxr-xr-x 13 superuser      1000 4096 2022-11-21 09:09 jenkins_home/
-rw-r--r-- 1 superuser everyone 3570 2022-11-03 19:12 mariadb-server.cnf
[[superuser@Hermes ci]$ docker run -d -p 1080:8080 -v $PWD/jenkins_home:/var/jen
ins_home --name jenkins --hostname jenkins jenkins/jenkins
0d969d251ca452f74b392aaa24d2fc41fb60389525bd712a9d49453fe913663
[superuser@Hermes ci]$
```

- Como en nuestro servidor web no corremos los contenedores con el usuario root se debe cambiar el usuario y el grupo dueños del directorio que usará Jenkins como /var/jenkins_home

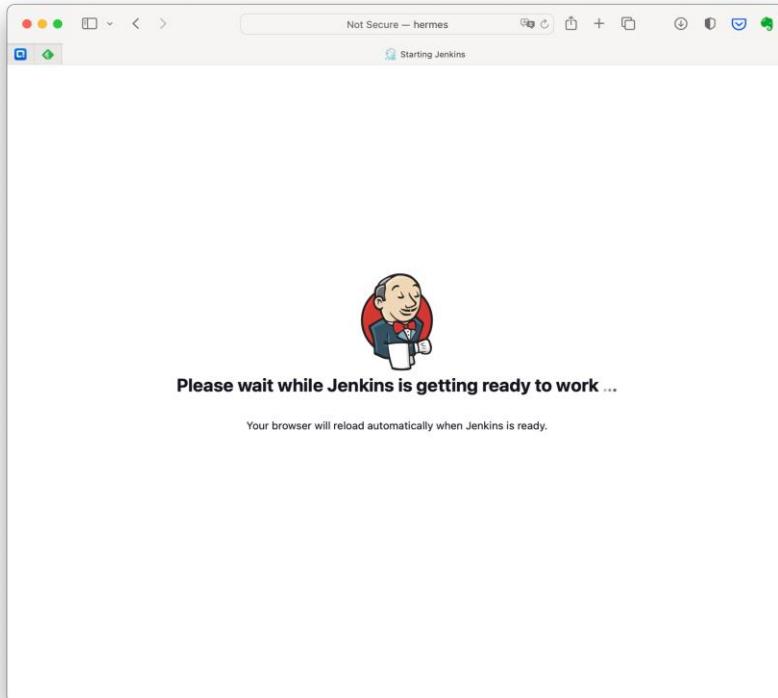
```
[superuser@Hermes ci]$ docker run -d -p 1080:8080 -v $PWD/jenkins_home:/var/jen
ins_home --name jenkins --hostname jenkins jenkins/jenkins
ccb1b933c69a1ae2aa09cc513fc4cc1988b79c3a42f94672e5183043564b6c48
[[superuser@Hermes ci]$ sudo chown -R 1000:1000 jenkins_home
Password:
[superuser@Hermes ci]$
```

- Cuando se inicia el contenedor de Jenkins la primera vez genera una contraseña aleatoria para el usuario "admin". Si se quiere mirar se puede ejecutar el contenedor en modo "foreground" o "visible" (es decir sin la opción -d de docker) o se pueden revisar los registros de consola con "docker logs Jenkins" (en nuestro caso el contenedor literalmente se llama "Jenkins")

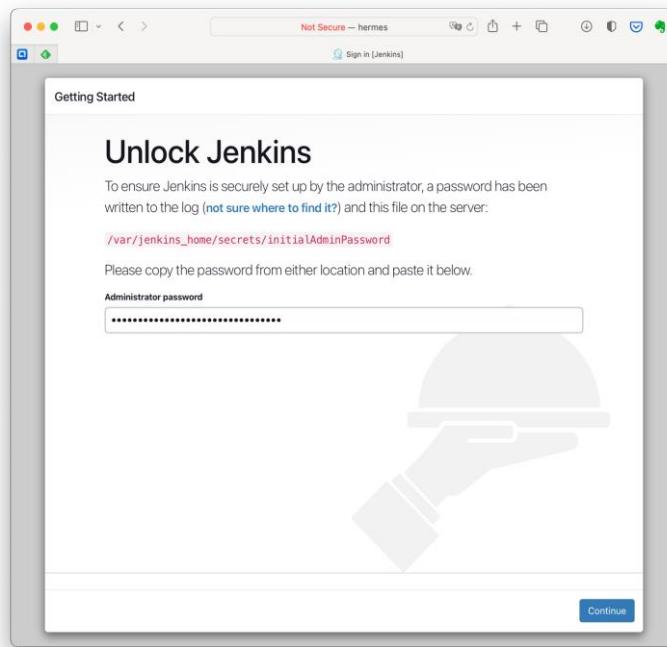
```
e — ssh hermes — 80x25
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2022-11-21 14:14:06.679+0000 [id=29]     INFO    jenkins.install.SetupWizard#init
:

*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
06e6c2ef689b403aacdf1af52af6a364
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****
[superuser@Hermes ci]$
```

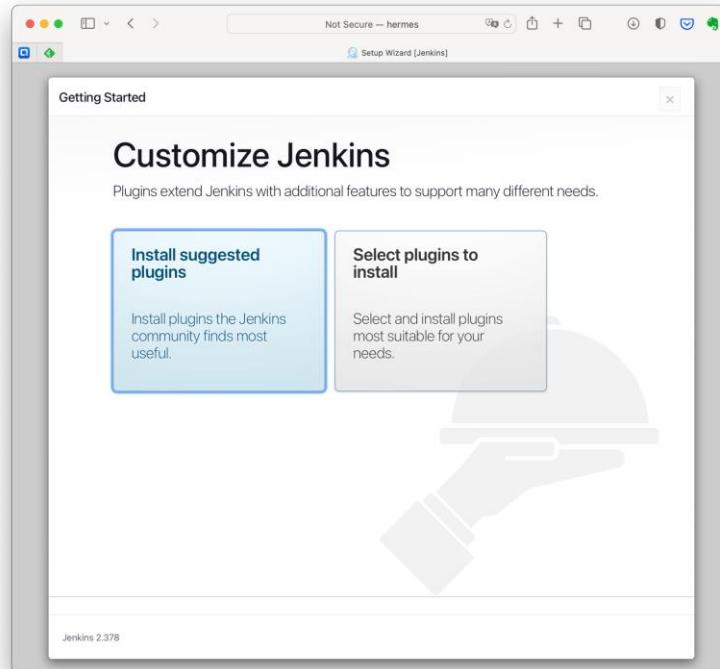
7. Al iniciar el contenedor si se accede al puerto TCP como se publicó muestra la imagen inicial (<http://hermes:1080/> porque el servidor docker se llama "hermes").



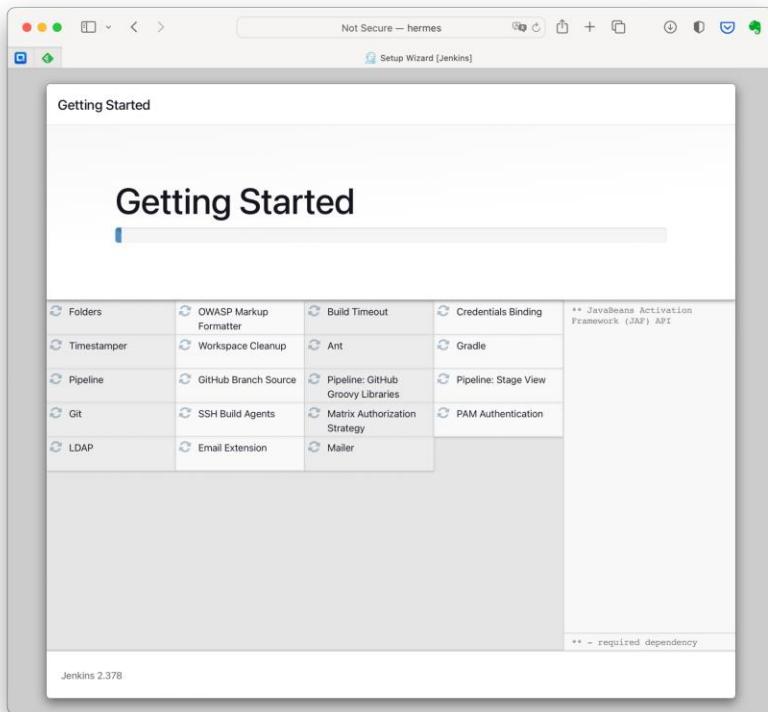
8. Se debe ingresar la contraseña copiada de los logs de la ejecución inicial del contenedor.



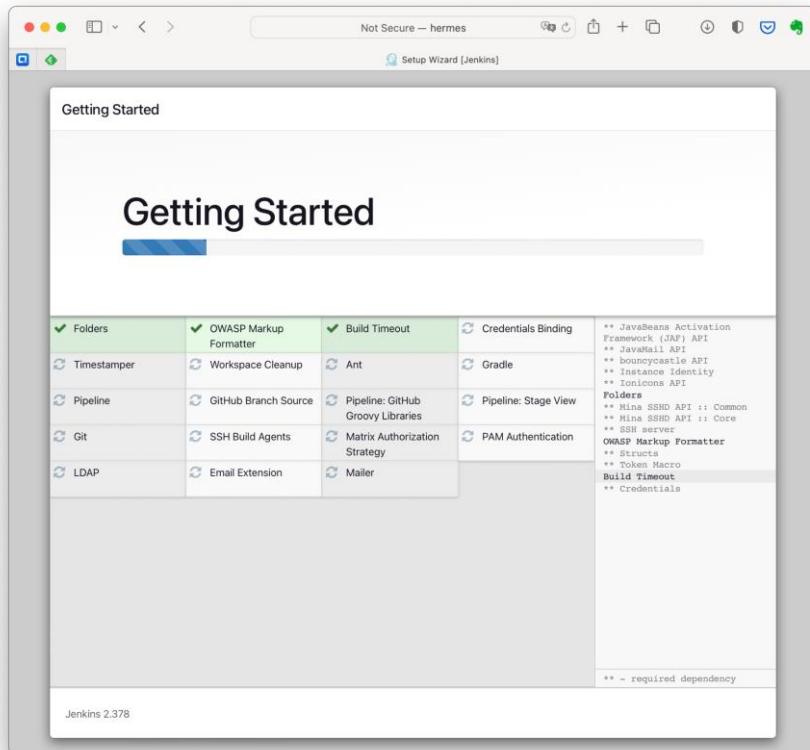
9. La ejecución inicial ofrece o instalar un juego de plugins (extensiones) recomendadas, o seleccionar manualmente cuáles plugins queremos para comenzar. Cuando uno no es experimentado en Jenkins selecciona "sugeridos".



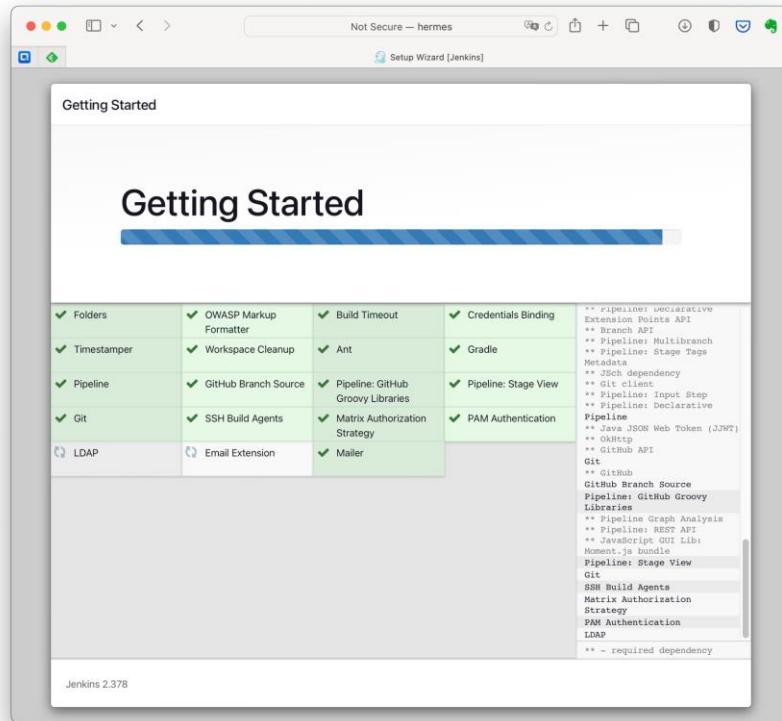
10. Cuando se acepta la instalación de los plugins recomendados/sugeridos comienza la instalación de estos. Son 19 los sugeridos.



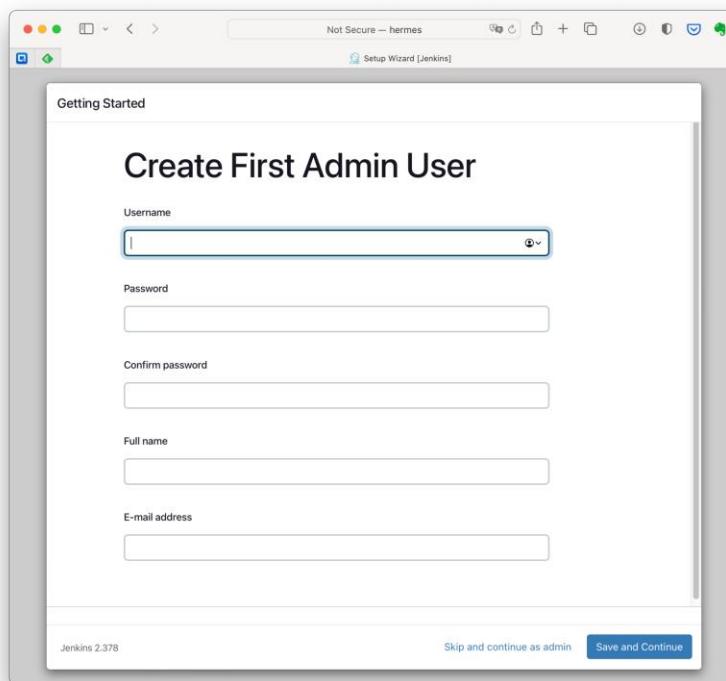
11. Continúa: Instalación de plugins sugeridos...



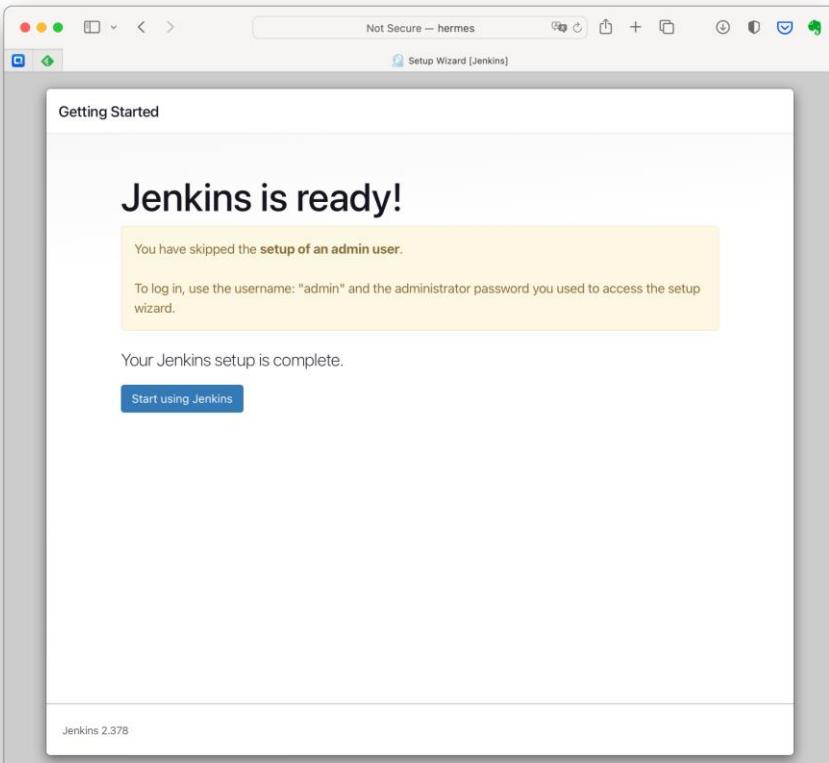
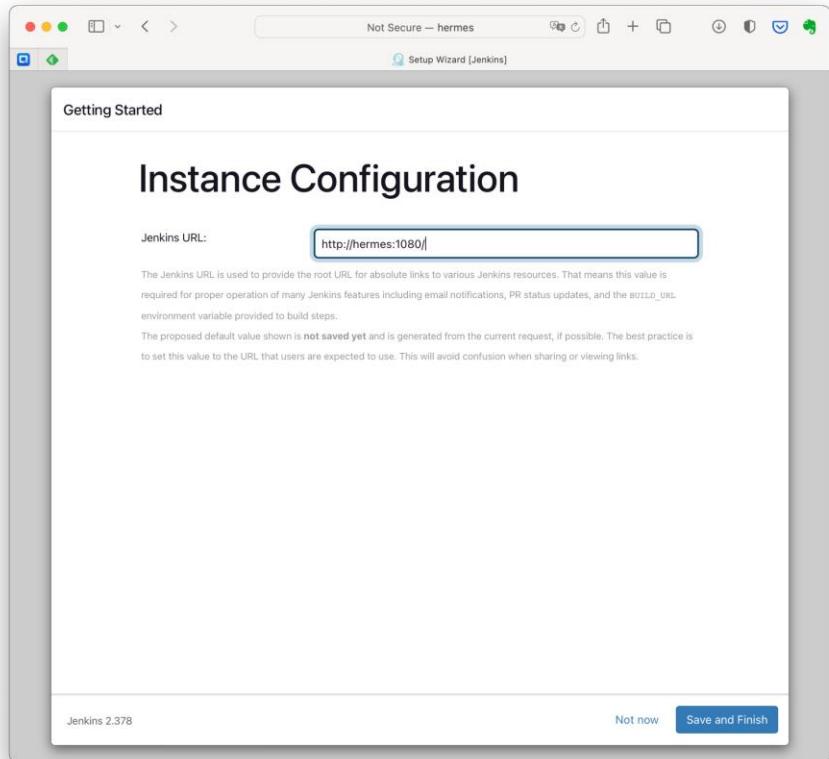
12. Continúa: Instalación de plugins sugeridos...



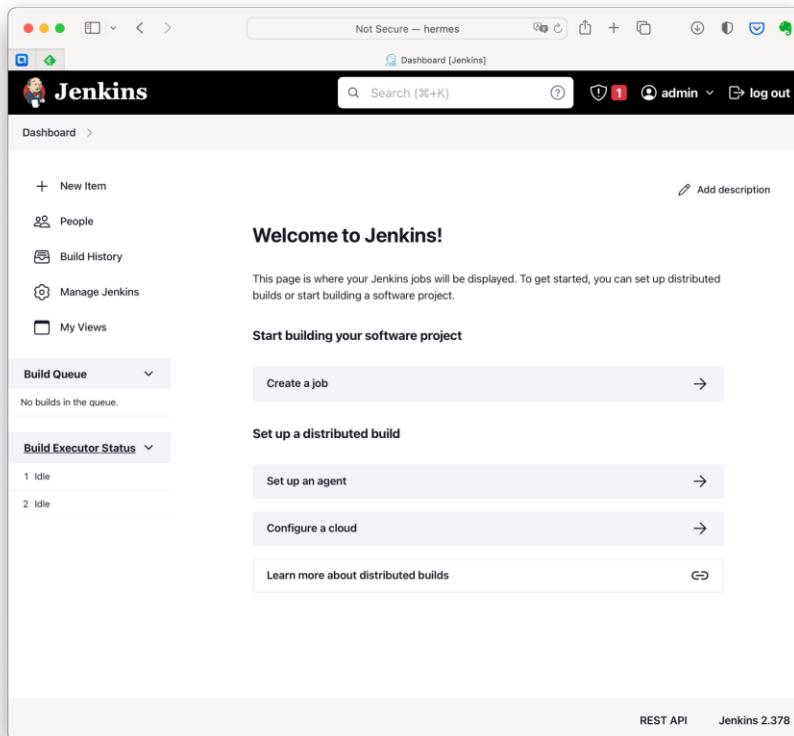
13. Al completar la instalación de los plugins aparece una pantalla de asistente sugiriendo que se cree un primer usuario administrador. En nuestro caso no lo hicimos porque creamos un usuario para cada integrante de nuestro grupo. (Skip and continue)



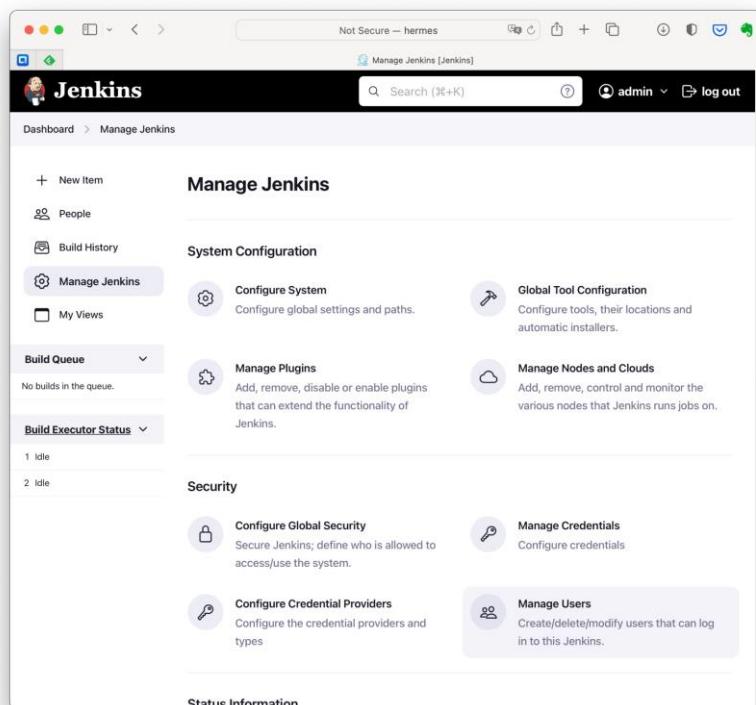
14. El asistente pide confirmar el URL de acceso a la página de Jenkins. Los valores visibles en la imagen son los correctos en este caso.



15. Página inicial de Jenkins luego de identificarse y autenticarse. También aparece luego del asistente de instalación y configuración inicial si uno no se ha desconectado de la sesión.



16. Para configurar Jenkins se usa el ítem "Manage Jenkins" de la barra de navegación de la izquierda. En la misma usamos "Manage Users" en la derecha y abajo (en la imagen)



17. La imagen muestra la configuración de los usuarios cuando se usan internos a Jenkins (por eso la página se titula "Jenkin's own user database"). Inicia solo con el usuario *admin* porque no creamos usuarios adicionales en el asistente.

The screenshot shows the Jenkins 'Users' management interface. At the top, there is a navigation bar with the Jenkins logo, the title 'Users [Jenkins]', a search bar, and a 'log out' button. Below the navigation is a breadcrumb trail: 'Dashboard > Manage Jenkins > Jenkins' own user database'. The main content area is titled 'Users' and contains a table with one row. The table has two columns: 'User ID' and 'Name'. The 'User ID' column shows a user icon next to the text 'admin', and the 'Name' column shows the text 'admin'. To the right of the table is a blue button labeled '+ Create User'. A note below the table states: 'These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.' At the bottom right of the page, the text 'Jenkins 2.378' is visible.

18. Creación del usuario *rbarrios*

The screenshot shows the Jenkins 'Create User' form. At the top, there is a navigation bar with the Jenkins logo, the title 'Create User [Jenkins]', a search bar, and a 'log out' button. Below the navigation is a breadcrumb trail: 'Dashboard > Manage Jenkins > Jenkins' own user database > Create User'. The main content area is titled 'Create User'. It contains several input fields: 'Username' (containing 'rbarrios'), 'Password' (containing '*****'), 'Confirm password' (containing '*****'), 'Full name' (containing 'Ramón Barrios Láscar'), and 'E-mail address' (containing 'elbarrios1@poligran.edu.co'). At the bottom of the form is a blue 'Create User' button. The bottom right of the page shows the text 'Jenkins 2.378'.

19. Creación del/de la usuario *darango*

The screenshot shows the Jenkins 'Create User' interface. The 'Username' field contains 'darango'. The 'Password' and 'Confirm password' fields both show masked input. The 'Full name' field contains 'Deyssi Arango Bermúdez'. The 'E-mail address' field contains 'dcaarango@poligran.edu.co'. A blue 'Create User' button is at the bottom.

20. Creación del/de la usuario *lsalas*

The screenshot shows the Jenkins 'Create User' interface. The 'Username' field contains 'lsalas'. The 'Password' and 'Confirm password' fields both show masked input. The 'Full name' field contains 'Luis Salas Alfonso'. The 'E-mail address' field contains 'lsalas@poligran.edu.co'. A blue 'Create User' button is at the bottom.

21. Creación del/de la usuario *sgordillo*

The screenshot shows the Jenkins 'Create User' interface. The 'Username' field contains 'sgordillo'. The 'Password' and 'Confirm password' fields both contain masked text. The 'Full name' field contains 'Sebastián Gordillo Patiño'. The 'E-mail address' field contains 'jsgordillo@poligran.edu.co'. A blue 'Create User' button is at the bottom.

22. Creación del/de la usuario *mcoy*

The screenshot shows the Jenkins 'Create User' interface. The 'Username' field contains 'mcoy'. The 'Password' and 'Confirm password' fields both contain masked text. The 'Full name' field contains 'Manuel Coy'. The 'E-mail address' field contains 'macoyben@poligran.edu.co'. A blue 'Create User' button is at the bottom.

23. Vista de todos los usuarios creados. A todos se les ajustó luego para que la fecha y la hora las maneje con "América/Bogotá"

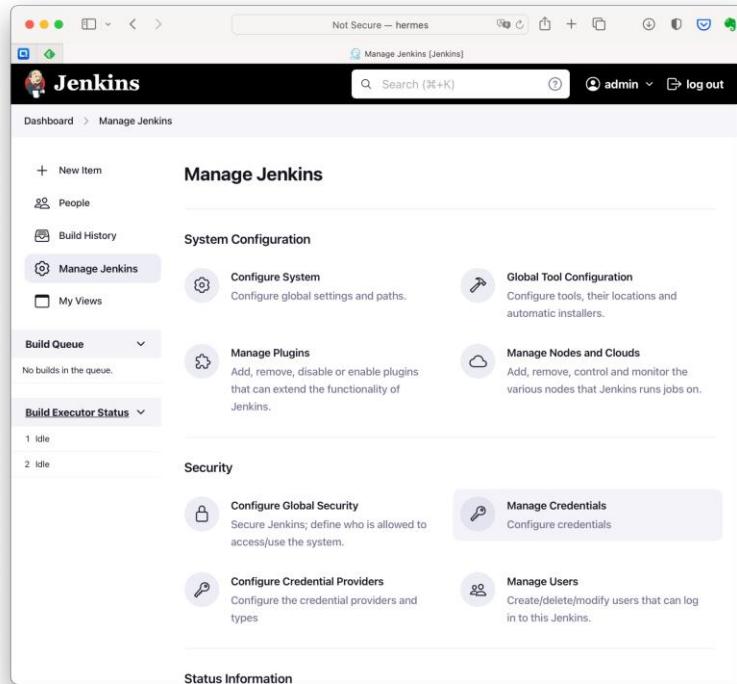
The screenshot shows the Jenkins 'Users' page. At the top, there is a navigation bar with the Jenkins logo, a search bar containing 'Search (⌘+K)', and a user dropdown set to 'admin'. Below the navigation bar, the page title is 'Users [Jenkins]'. A breadcrumb trail indicates the current location: 'Dashboard > Jenkins' own user database'. A blue button labeled '+ Create User' is located in the top right corner. The main content area is titled 'Users' and contains a table with the following data:

| User ID | Name | |
|-----------|---------------------------|--|
| admin | admin | |
| darango | Deyssi Arango Bermúdez | |
| lsalas | Luis Salas Alfonso | |
| mcoy | Manuel Coy | |
| rbarrios | Ramón Barrios Láschar | |
| sgordillo | Sebastián Gordillo Patiño | |

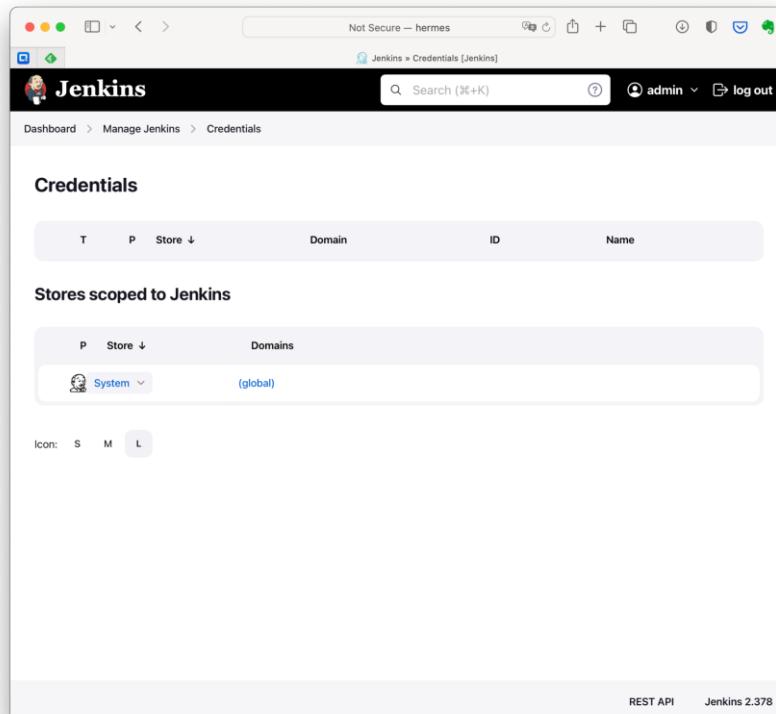
In the bottom right corner of the main content area, the text 'Jenkins 2.378' is visible.

3. Explicación de la instalación de Jenkins con github

1. Se debe configurar la conexión con GitHub, ya que nuestro código está en un repositorio allí. Se usa "Manage Jenkins" -> "Manage Credentials" como se ve en la imagen.



2. Página inicial de manejo de credenciales "Credentials" de Jenkins.



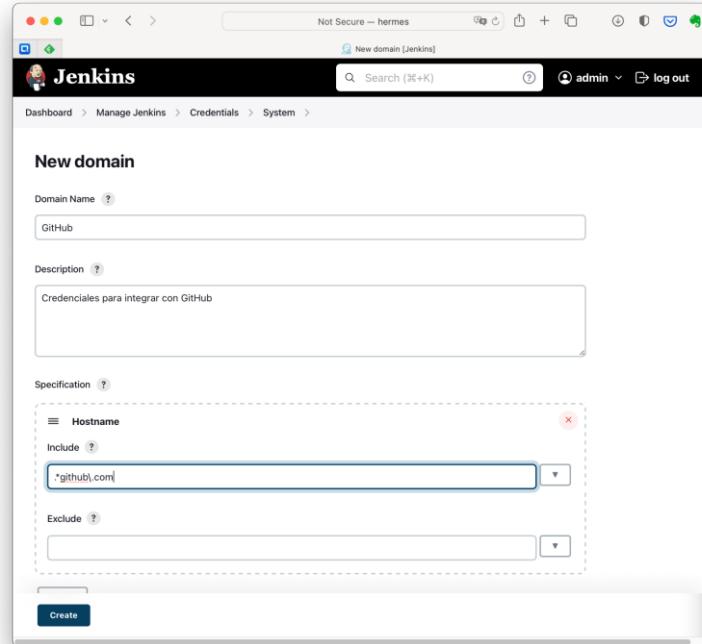
3. Las credenciales en Jenkins pueden ser para el software Jenkins, para todos los nodos (los otros servidores Jenkins dependientes de este para builds y parecidos), para proyectos individuales. Lo correcto es tener dominios para tener separación de credenciales. En la imagen "Manage Jenkins" -> "Credentials" -> "System" se pueden manejar "Dominios" que son los conjuntos que permiten separar las credenciales.

The screenshot shows the Jenkins System credentials page. At the top right is a blue button labeled '+ Add domain'. Below it is a table with two columns: 'Domain' and 'Description'. A single row is visible, titled 'Global credentials (unrestricted)' with a small icon of a crown. The description text reads: 'Credentials that should be available irrespective of domain specification to requirements matching.' At the bottom left of the table are icons for 'Icon:' followed by 'S', 'M', and 'L'. The bottom right corner of the page displays 'REST API' and 'Jenkins 2.378'.

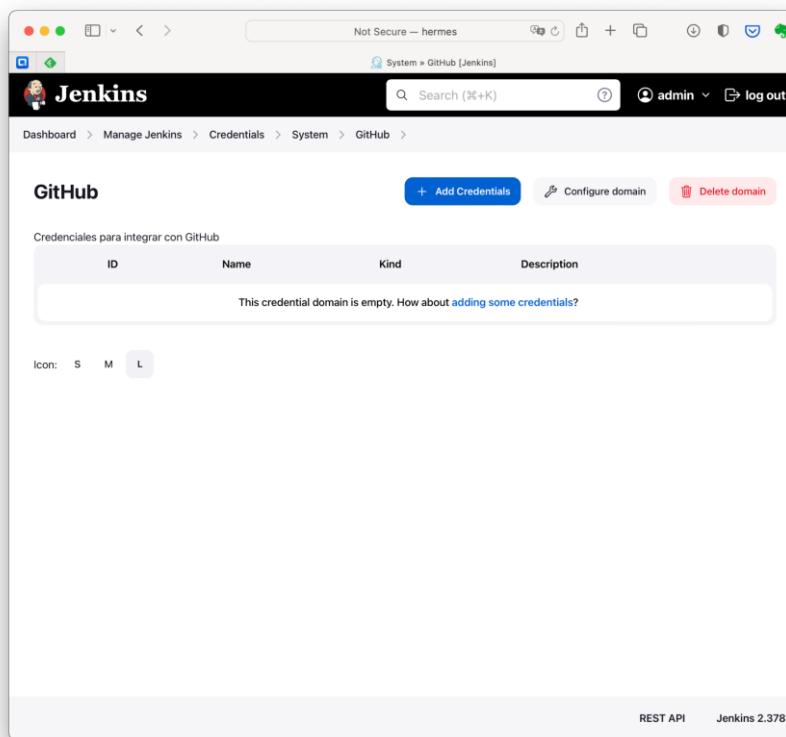
4. Ventana de creación de un nuevo dominio "GitHub" para separar las credenciales.

The screenshot shows the 'New domain' creation form. It has three main sections: 'Domain Name' with a value 'GitHub', 'Description' with a note 'Credenciales para integrar con GitHub', and 'Specification' with a dropdown menu showing 'Add'. At the bottom is a large 'Create' button. The bottom right corner of the page displays 'REST API' and 'Jenkins 2.378'.

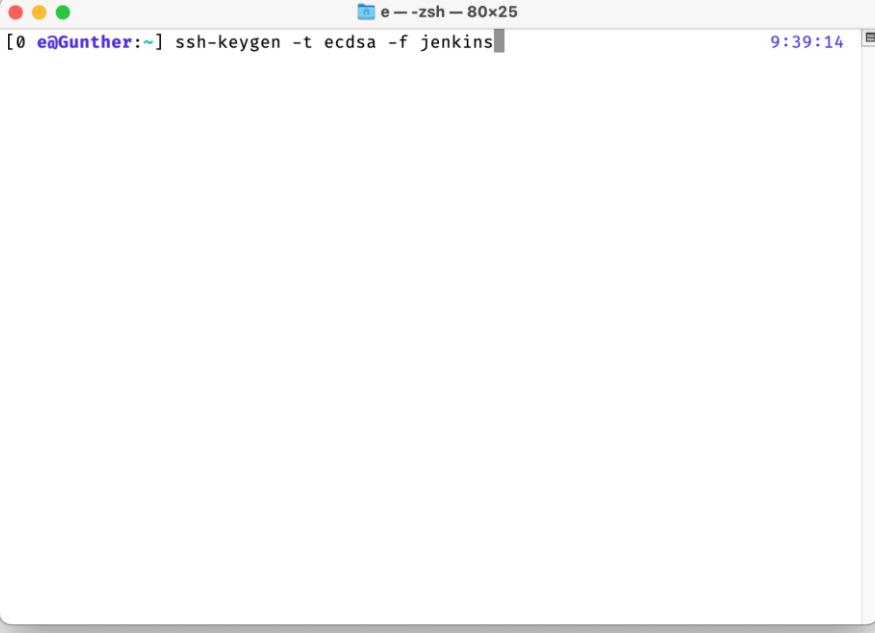
5. Para que Jenkins sepa que la credencial es para usar con páginas dentro de GitHub se le da un filtro de hostname en formato de expresiones regulares de java: `.*github\..com`, que significa "cualquier texto, la palabra github, un punto, y com". Con eso la credencial solo va a ser usada para máquinas dentro del dominio "github.com".



6. Luego de creado el dominio hay que agregarle las credenciales que queremos usar en las máquinas que concuerden con ese dominio. La imagen muestra dónde.

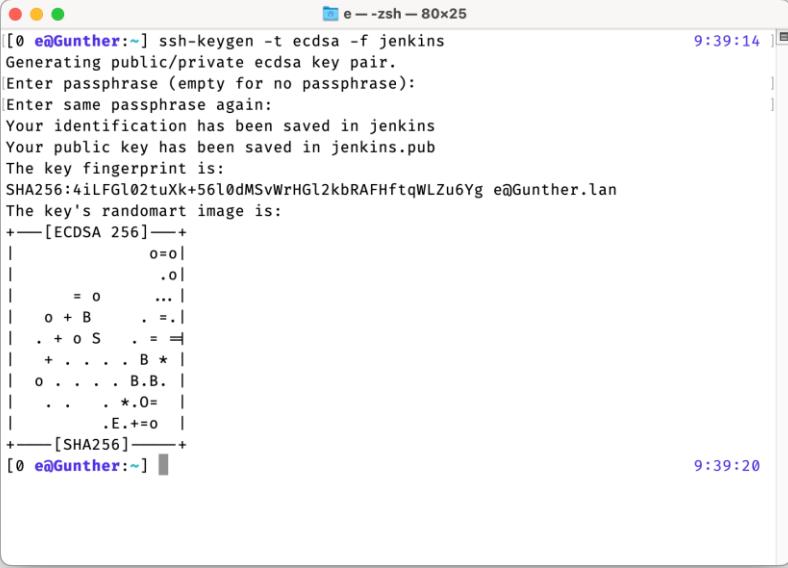


7. Como GitHub desde finales de 2021 solo acepta conexiones con llaves SSH si se entra aparte del acceso por la página (la de github.com), hay que usar unas llaves SSH para la conexión. En la imagen se muestra cómo se crean (con ssh-keygen en MacOS, Linux, Unix, la línea de comandos de Docker, o WSL2 de Windows). La llaves se llamarán Jenkins y jenkins.pub en el directorio normal de llaves en nuestro caso.



A screenshot of a terminal window titled "e -- zsh - 80x25". The window shows the command "ssh-keygen -t ecdsa -f jenkins" being run at the prompt "[0 e@Gunther:~]". The terminal is mostly blank, indicating the process is still running or has just started.

8. Ejemplo de las llaves creadas. Se seleccionó tipo ECDSA que es un poco más moderno que el predeterminado.

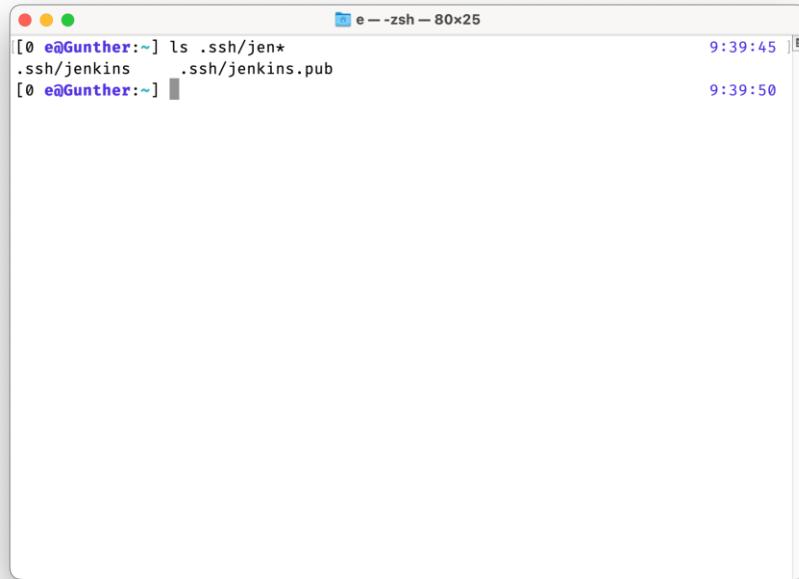


A screenshot of a terminal window titled "e -- zsh - 80x25". The window shows the command "ssh-keygen -t ecdsa -f jenkins" being run at the prompt "[0 e@Gunther:~]". The output includes:

```
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in jenkins
Your public key has been saved in jenkins.pub
The key fingerprint is:
SHA256:4iLFGl02tuXk+56l0dMSvWrHGl2kbRAFHftqWLZu6Yg e@Gunther.lan
The key's randomart image is:
+---[ECDSA 256]---+
|          o=o|
|          .o|
| = o     ... |
| o + B   . =.|
| . + o S  . = =|
| + . . . . B * |
| o . . . . B.B.|
| . . . . *.O= |
|          .E.+o |
+---[SHA256]---+
```

The timestamp "9:39:14" is visible in the top right corner, and "9:39:20" is visible in the bottom right corner.

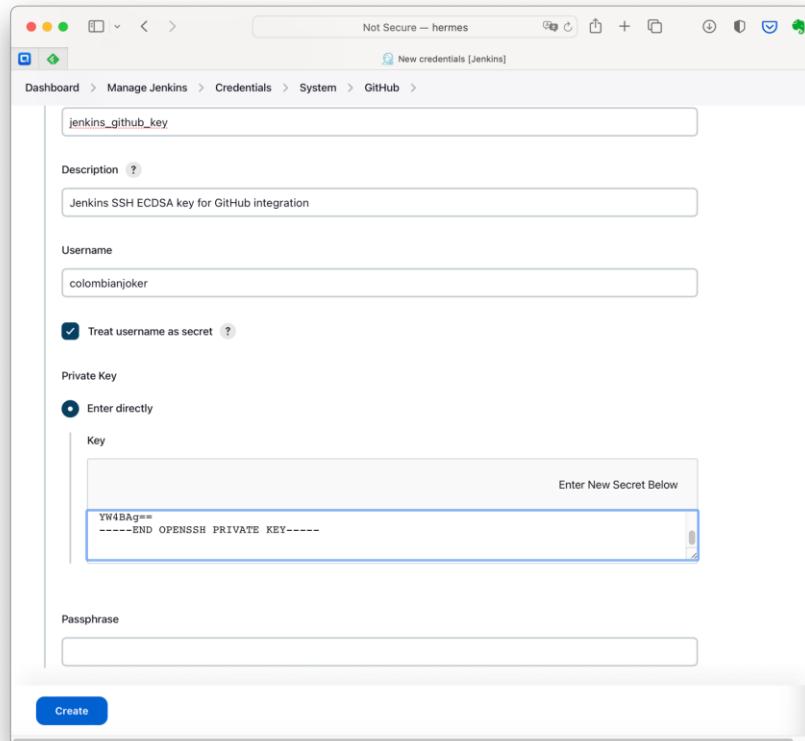
9. Se muestran los dos archivos que se generan. El contenido del .pub lo debe tener GitHub y el contenido del otro Jenkins.



```
[0 e@Gunther:~] ls .ssh/jen*
.ssh/jenkins      .ssh/jenkins.pub
[0 e@Gunther:~] 9:39:45 9:39:50
```

A screenshot of a macOS terminal window titled "e -- zsh - 80x25". The window shows the command "ls .ssh/jen*" being run, which lists two files: ".ssh/jenkins" and ".ssh/jenkins.pub". The timestamp at the bottom right of the terminal window is 9:39:50.

10. En "agregar credenciales al dominio" se pone un identificador, una descripción, un nombre de usuario (que marcamos para que no lo vuelva a mostrar en la pantalla de configuración de Jenkins), y en "Private Key" el contenido del archivo no-.priv acabado de crear.



11. La imagen muestra cómo queda luego de configuradas las credenciales de este dominio "GitHub"

The screenshot shows the Jenkins interface for managing GitHub credentials. At the top, there's a navigation bar with links to 'Dashboard', 'Manage Jenkins', 'Credentials', 'System', and 'GitHub'. Below this, a sub-navigation bar for 'GitHub' includes 'Add Credentials', 'Configure domain', and 'Delete domain' buttons. The main content area is titled 'GitHub' and 'Credenciales para integrar con GitHub'. It lists a single credential entry:

| ID | Name | Kind | Description |
|--------------------|--|-------------------------------|--|
| jenkins_github_key | Jenkins SSH ECDSA key for GitHub integration | SSH Username with private key | Jenkins SSH ECDSA key for GitHub integration |

At the bottom left, there are icons for 'Icon: S M L'. The bottom right corner shows 'REST API' and 'Jenkins 2.378'.

12. La imagen muestra cómo crear un job de pruebas donde usar las funcionalidades de Jenkins y en particular un pipeline que sea usable para nuestro caso:

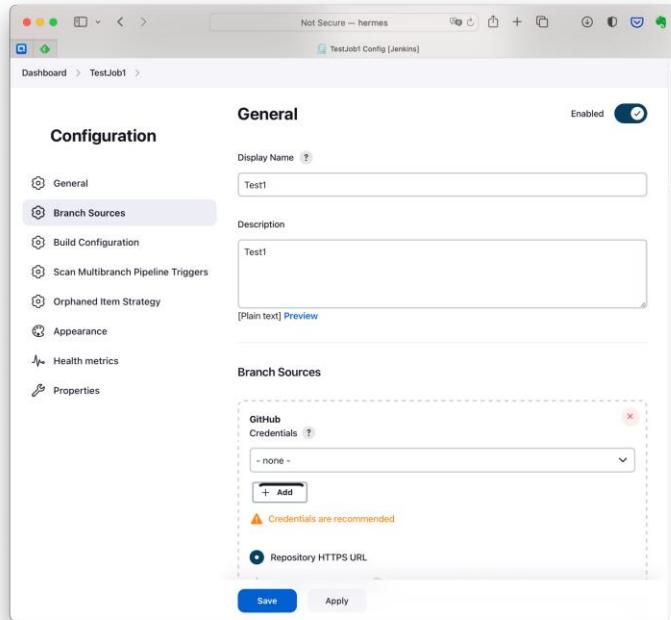
- De GitHub
- Como tenemos dos branches "main" y "master", que sea "Multibranch Pipeline"
- Luego probaremos que funciona

The screenshot shows the Jenkins 'New Item' dialog box. In the top-left, there's a search bar with 'TestJob' typed into it. Below the search bar, a message says '» Required field'. The main area lists several project types with their descriptions:

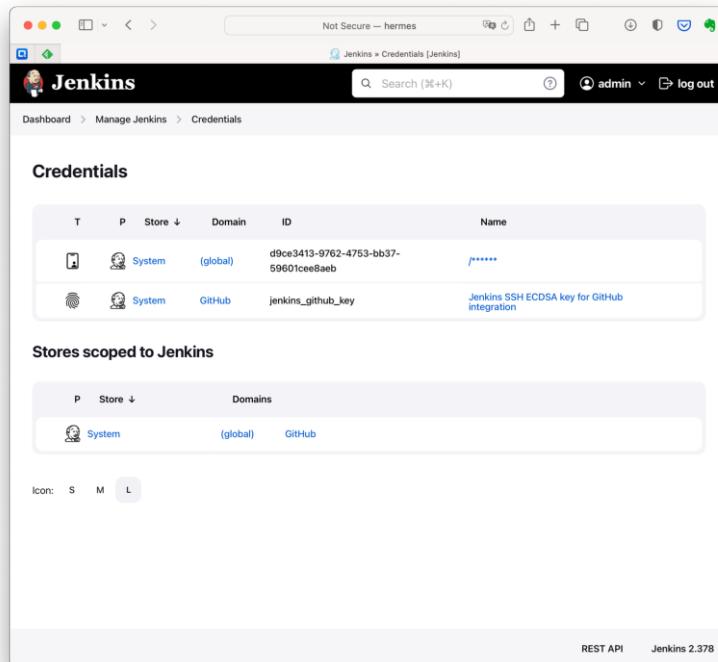
- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**: Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Creates a set of multibranch project subfolders by scanning for repositories.

At the bottom of the dialog is an 'OK' button.

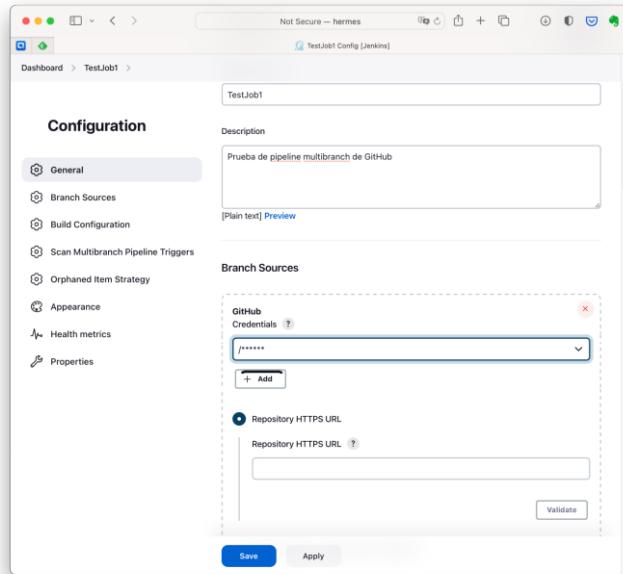
13. Dentro del job le ponemos valores como nombre visible "Display Name" y descripción, pero en particular, decirle que es de GitHub y darle las credenciales adecuadas en "Branch Sources".



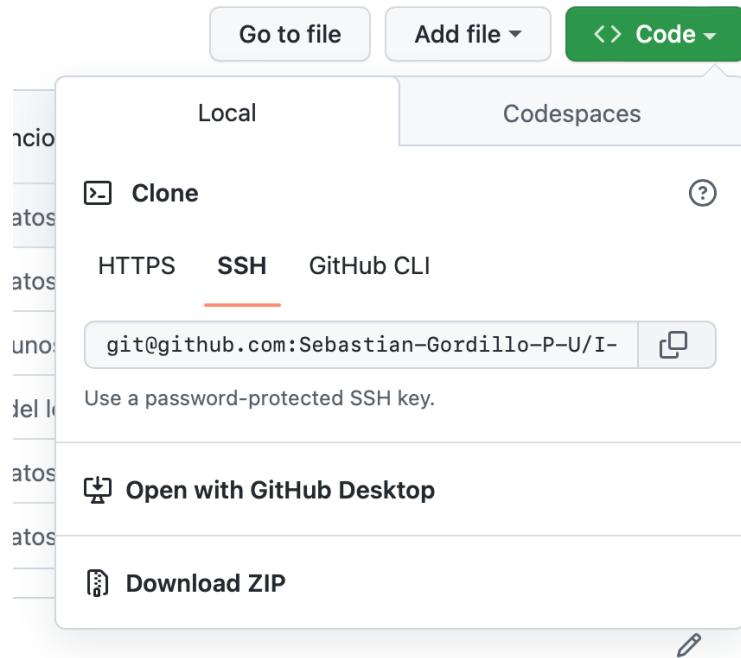
14. En la imagen se muestra que tenemos Credenciales globales. Están en "Manage Jenkins" -> "Credentials"



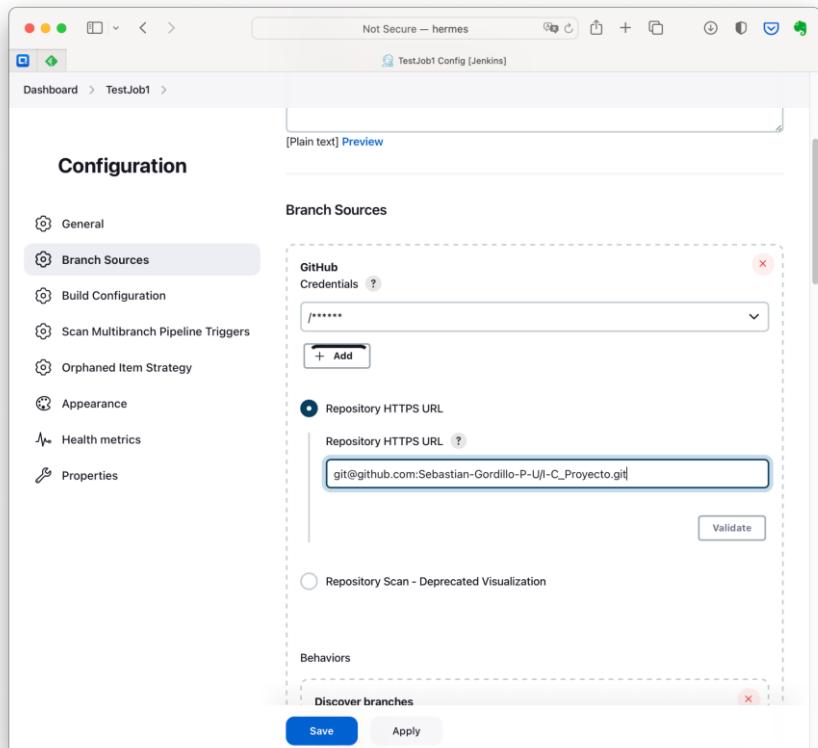
15. Escogemos las credenciales (que como dijimos "maneje el usuario como secreto" muestra *****), y habrá que decirle la URL del repositorio (que debe ser de GitHub porque la credencial está limitada a ese dominio)



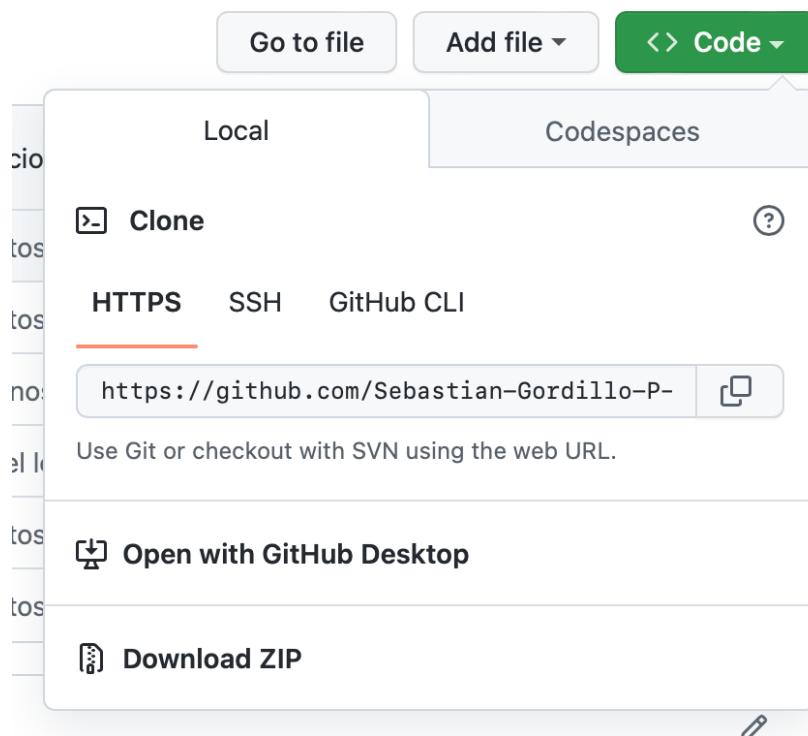
16. Para que conecte, deberemos darle la URL. En la imagen se ve la URL en protocolo SSH, aunque no vamos a usar SSH, vamos a usar HTTPS.



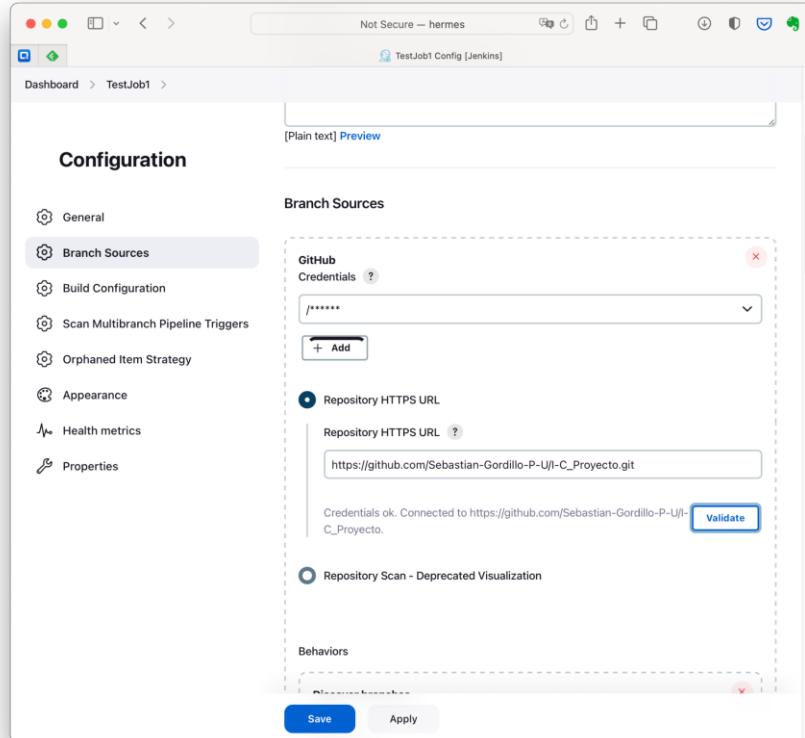
17. URL de un repositorio. No sirve este valor, ya que Jenkins pide una URL HTTPS.



18. La imagen muestra dónde, de nuestro repositorio GitHub obtener la URL HTTPS que necesitamos para Jenkins.



19. Agregar la URL correcta (HTTPS) a la definición en Jenkins.



20. La imagen muestra la llave SSH recién creada (en MacOS) ya cargada (realmente la mitad pública no más), en la interfaz de GitHub.

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys

| GitHub Key | |
|---|--|
|  | SHA256:4970HC73opglx82Lh0dJcYXXu2ii9hnk+5YgMiPRmg8 |
| SSH | Added on Nov 4, 2022 |
| | Last used within the last week — Read/write |
| Delete | |
| Jenkins | |
|  | SHA256:4iLFG102tuXk+5610dMSvWrHG12kbRAFHftqWLZu6Yg |
| SSH | Added on Nov 21, 2022 |
| | Never used — Read/write |
| Delete | |

21. En Jenkins se pueden configurar varios parámetros del Job:

- Descubra las ramas (es una pipeline multibranch)
- Descubra los pulls del origen
- Descubra los pulls de los forks
- Confíe en todos los usuarios administradores de la branch o que tengan escritura

The screenshot shows the Jenkins configuration interface for a job named "TestJob1". The left sidebar lists configuration sections: General, Branch Sources (selected), Build Configuration, Scan Multibranch Pipeline Triggers, Orphaned Item Strategy, Appearance, Health metrics, and Properties. The main panel is titled "Behaviors" and contains three sections: "Discover branches", "Discover pull requests from origin", and "Discover pull requests from forks". Each section includes a "Strategy" dropdown and a note below it. At the bottom, there is a note about compatibility with GitHub Enterprise and buttons for "Save" and "Apply".

Not Secure — hermes

TestJob1 Config [Jenkins]

Dashboard > TestJob1 >

Configuration

General

Branch Sources

Build Configuration

Scan Multibranch Pipeline Triggers

Orphaned Item Strategy

Appearance

Health metrics

Properties

Behaviors

Discover branches

Strategy

Exclude branches that are also filed as PRs

Discover pull requests from origin

Strategy

Merging the pull request with the current target branch revision

Discover pull requests from forks

Strategy

Merging the pull request with the current target branch revision

Trust

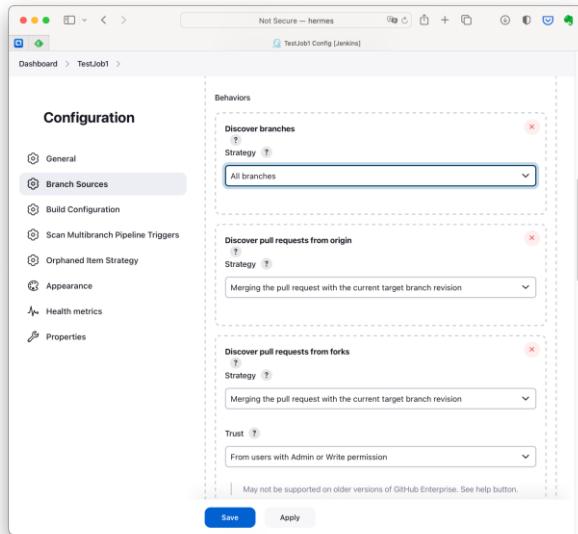
From users with Admin or Write permission

May not be supported on older versions of GitHub Enterprise. See help button.

Save Apply

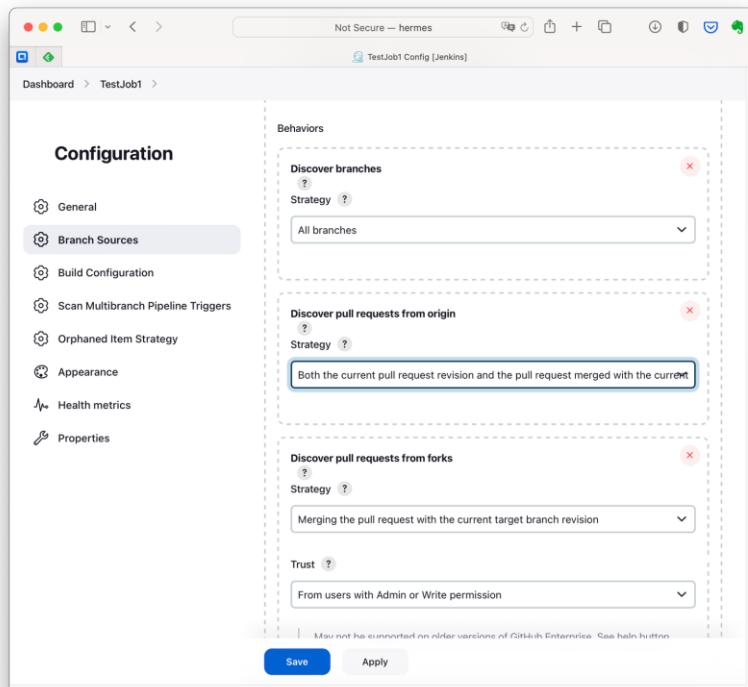
22. En Jenkins se pueden configurar varios parámetros del Job:

- Descubra las ramas (es una pipeline multibranch)
- Descubra los pulls del origen
- Descubra los pulls de los forks
- Confíe en todos los usuarios administradores de la branch o que tengan escritura



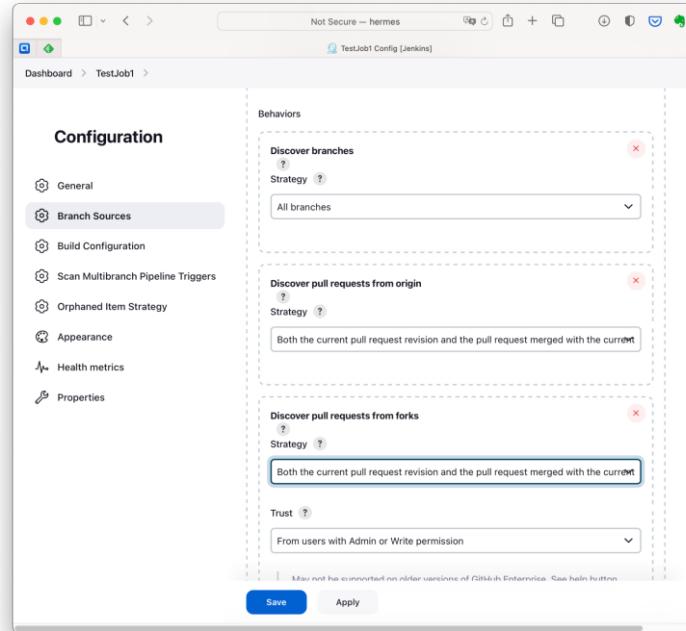
23. En Jenkins se pueden configurar varios parámetros del Job:

- Descubra las ramas (es una pipeline multibranch)
- Descubra los pulls del origen
- Descubra los pulls de los forks
- Confíe en todos los usuarios administradores de la branch o que tengan escritura

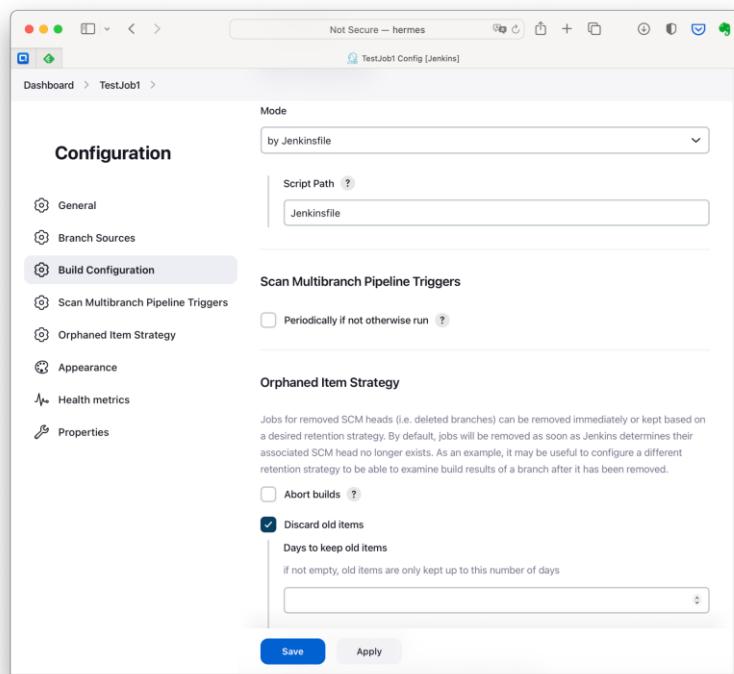


24. En Jenkins se pueden configurar varios parámetros del Job:

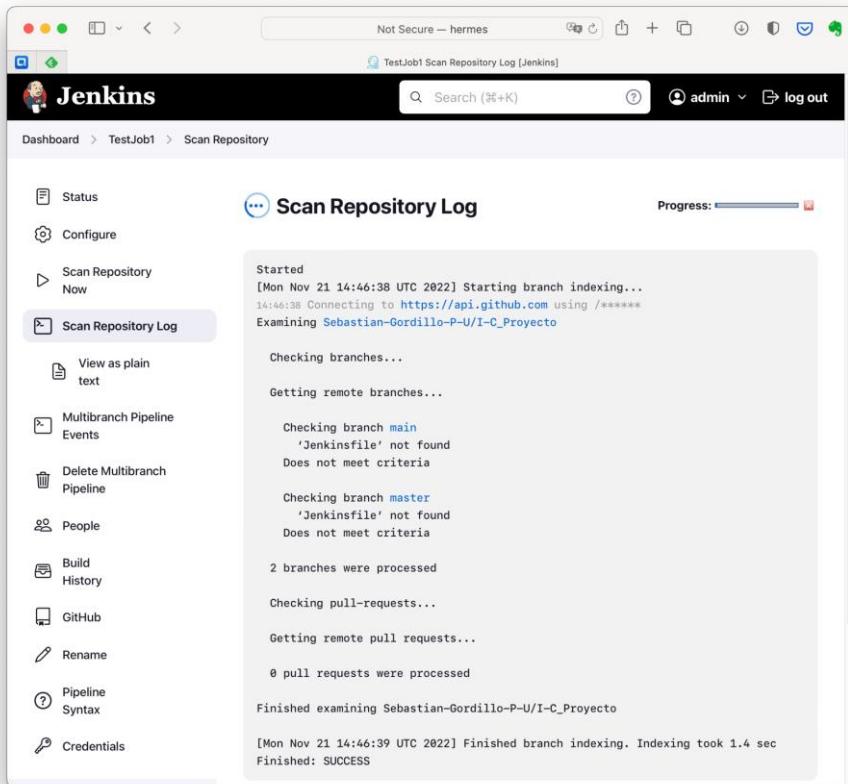
- Descubra las ramas (es una pipeline multibranch)
- Descubra los pulls del origen
- Descubra los pulls de los forks
- Confíe en todos los usuarios administradores de la branch o que tengan escritura



25. Que Jenkins dependa del Jenkinsfile (por branch) es lo predeterminado.



26. Se le dice escanear "Scan Repository", y podemos observar que sí pudo indexar, examinar el repo en GitHub, pudo obtener las branches (son dos, "main" y "master"), y funcionó (termina en "SUCCESS")



27. Nótese que cuando uno agrega llaves de acceso por SSH a GitHub llega una notificación al correo avisando de la novedad de seguridad.

G GitHub <noreply@github.com>

[GitHub] A new SSH authentication public key was added to your account

To: Ramón Barrios Láscaz <RamonBarriosLascar@GMail.com>

The following SSH key was added to your account:

Jenkins

SHA256:4iLFGl02tuXk+56l0dMSvWrHGl2kbRAFHftqWLZu6Yg

If you believe this key was added in error, you can remove the key and disable access at the following location:

<https://github.com/settings/keys>

Referencias

Microsoft Azure, [2022], ¿Qué es DevOps?, recuperado de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/>

SimpliLearn, [2019], Jenkins Full Course | Jenkins Tutorial for Beginners | Jenkins Tutorial, recuperado de <https://youtu.be/FX322RVNGj4>

El Pingüino Tech, [2022], Cómo Descargar e Instalar JENKINS en WINDOWS | Paso a Paso, recuperado de <https://www.youtube.com/watch?v=3OY3A6XE6Us>

The HarryCode, [2021], Curso Jenkins | Como Configurar y Ejecutar Diferentes Versiones de Java (JDK) en Jenkins, recuperado de <https://www.youtube.com/watch?v=KOKHIFI5R8>

Oliveros, A. (2019). Fundamentos de la Integración Continua.

Oliveros, A. (2019b). Máquinas virtuales y Dockers para construcción de ambientes.

Oliveros, A. (2019c). Uso de contenedores en DevOps.

Oliveros, A. (2019b). Integraciones Git con otras herramientas.

Leszko, R., [2017], Continuous Delivery with Docker and Jenkins, ISBN 978-1-78712-523-0

Lenz, M., [2019], Python Continuous Integration and Delivery, ISBN 978-1-4842-4281-0

Mouat, A., [2016], Using Docker, ISBN 978-1-491-91576-9