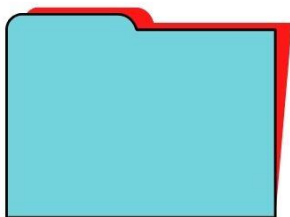


MANUEL FLORIDO GÁLVEZ



1º CFGS Administración de
Sistemas Informáticos en Red
2020/2021

Índice

1. Introducción.....	pág 1
2. Jenkins.....	pág 2
2.1. Instalación de Jenkins.....	pág 4
3. Git.....	pág 12
3.1. Github.....	pág 13
4. Instalación de Git plugin en Jenkins.....	pág 14
5. Enlace Jenkins con Git.....	pág 16
6. Crear trabajo de Jenkins.....	pág 22
7. Crontab.....	pág 28
8. Bibliografía.....	pág 32

1. Introducción

En este proyecto vamos a aprender a montar un servidor con Jenkins y a crear trabajos en este.



También usaremos Git y GitHub, creando repositorios locales y remotos, aprendiendo a cómo usarlos.

Al final del proyecto conseguiremos crear un trabajo en el servidor de Jenkins que sincronice el repositorio de GitHub, para hacer un clonado cada vez que se ejecute la tarea a la carpeta de trabajo creada anteriormente por Jenkins y, ejecutar tareas añadidas en un script subido al repositorio.



Después usaremos Crontab para programar una tarea que copie el archivo creado por el script al escritorio para que sea más accesible para el usuario.

Realizaremos el proyecto usando una máquina virtualizada con VirtualBox, que tendrá 2GB de RAM y 1 núcleo de procesador.

El Sistema Operativo que vamos a usar es Ubuntu 18.04



2. Jenkins

Jenkins es un servidor de integración continua, esto significa que es el encargado de actualizar y ejecutar pruebas de código periódicamente.

Sus objetivos son:



Encontrar y arreglar errores de código.

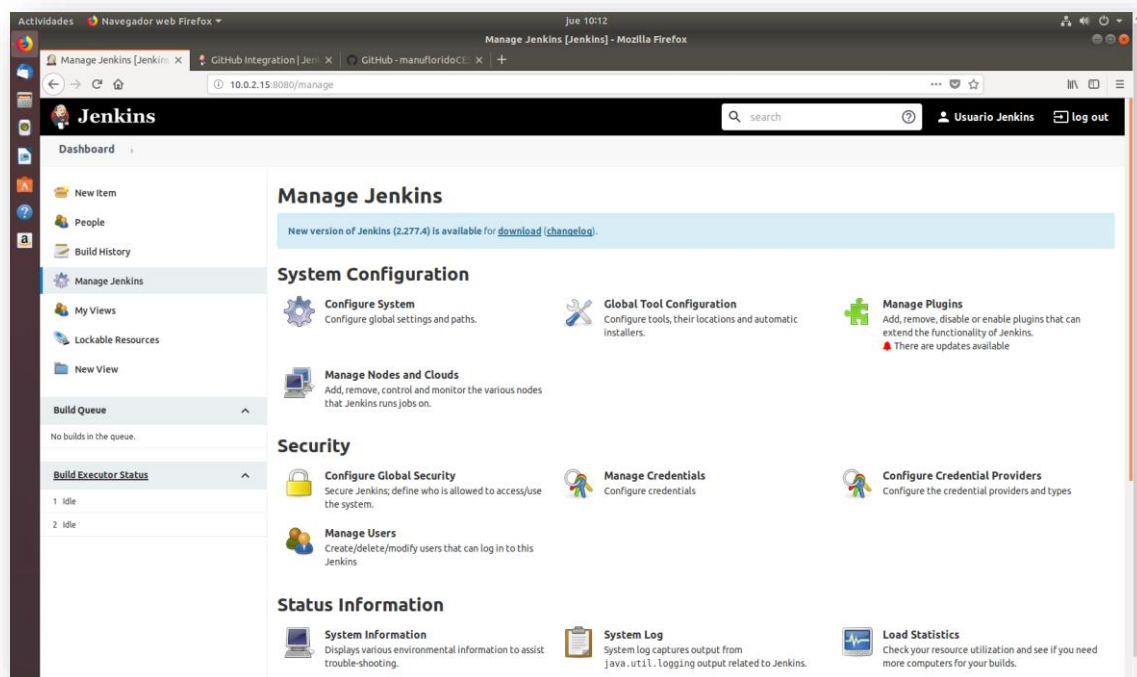


Mejorar la calidad del software.



Reducir el tiempo y el esfuerzo que supone actualizar el software periódicamente.

Dispone de una interfaz web con la que configurar todos los ajustes disponibles.



Admite el uso de herramientas de control de versiones como Subversion, GIT, Mercurial... Y además tiene la posibilidad de instalar una infinidad de plugins.

Originalmente fue desarrollado con el nombre de “*Hudson*”, en verano de 2004, pero debido a ciertos desacuerdos con Oracle, la comunidad decidió cambiar el nombre a “*Jenkins*”.

Actualmente Jenkins y Hudson son proyectos independientes, aunque entre ellos no hay casi ninguna diferencia.



Jenkins



Hudson

2.1. Instalación de Jenkins

¿Qué pasos debemos de seguir para instalar Jenkins?



El primer paso para la instalación de Jenkins es tener instalado Java.
El primer comando es:



~\$ sudo apt-get update.

```
Actividades Terminal
Archivo Editar Ver Buscar Terminal Ayuda
usuario@proyectofinal:~$ sudo apt-get update
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Leyendo lista de paquetes... Hecho
usuario@proyectofinal:~$
```



Después de actualizar la lista de paquetes, instalamos Java:



~\$ sudo apt-get install openjdk-8-jdk.

```
usuario@proyectofinal: ~
Archivo Editar Ver Buscar Terminal Ayuda
usuario@proyectofinal:~$ sudo apt-get install openjdk-8-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni libgif7 libice-dev libpthread-stubs0-dev libsm-dev libx11-6 libx11-dev libx11-doc libxau-dev libxau6 libxcb1 libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Paquetes sugeridos:
  default-jre libice-doc libsm-doc libxcb-doc libxt-doc openjdk-8-demo openjdk-8-source visualvm icedtea-8-plugin fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei
Se instalarán los siguientes paquetes NUEVOS:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni libgif7 libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libx11-doc libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Se actualizarán los siguientes paquetes:
  libx11-6 libxau6 libxcb1
3 actualizados, 23 nuevos se instalarán, 0 para eliminar y 649 no actualizados.
Se necesita descargar 43,9 MB/44,5 MB de archivos.
Se utilizarán 170 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```



Esperaremos a que aparezca el mensaje “done.” que confirma la instalación del paquete:

```
Configurando openjdk-8-jre:amd64 (8u292-b10-0ubuntu1-18.04) ...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/policytool para proveer /usr/bin/policytool (policytool) en m
odo automático
Configurando openjdk-8-jdk:amd64 (8u292-b10-0ubuntu1-18.04) ...
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/appletviewer para proveer /usr/bin/appletviewer (appletviewer) en
modo automático
update-alternatives: utilizando /usr/lib/jvm/java-8-openjdk-amd64/bin/jconsole para proveer /usr/bin/jconsole (jconsole) en modo automá
tico
Procesando disparadores para libc-bin (2.27-3ubuntu1) ...
Procesando disparadores para ca-certificates (20180409) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
done.
usuario@proyectofinal:~$
```



Añadimos la llave de Jenkins para que nos acepte el repositorio más tarde:



```
~$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key |
sudo apt-key add -
```

```
usuario@proyectofinal: ~
Archivo Editar Ver Buscar Terminal Ayuda
usuario@proyectofinal:~$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
OK
usuario@proyectofinal:~$
```



Accedemos al fichero “sources.list” y añadimos el repositorio de Jenkins:



```
~$ sudo nano /etc/apt/sources.list
```

```
usuario@proyectofinal: ~
GNU nano 2.9.3 /etc/apt/sources.list
deb cdrom:[Ubuntu 18.04.1 LTS _Bionic Beaver_ - Release amd64 (20180725)] bionic main restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://es.archive.ubuntu.com/ubuntu/ bionic main restricted
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://es.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://es.archive.ubuntu.com/ubuntu/ bionic universe
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic universe
deb http://es.archive.ubuntu.com/ubuntu/ bionic-updates universe
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-updates universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://es.archive.ubuntu.com/ubuntu/ bionic multiverse
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic multiverse
deb http://es.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-updates multiverse
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://es.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
deb-src http://es.archive.ubuntu.com/ubuntu/ bionic-backports main restricted universe multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
deb http://archive.canonical.com/ubuntu bionic partner
deb-src http://archive.canonical.com/ubuntu bionic partner
deb http://security.ubuntu.com/ubuntu bionic-security main restricted
deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
```



Añadimos “deb <https://pkg.jenkins.io/debian binary/>” y guardamos el archivo (Ctrl+o, Intro, Ctrl+x)

```
#Añadir repositorio de Jenkins
deb http://pkg.jenkins.io/debian-stable binary/
```



Volvemos a actualizar la lista de paquetes

```
usuario@proyectofinal:~$ sudo apt-get update
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Des:5 https://pkg.jenkins.io/debian-stable binary/ Release [2.044 B]
Des:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Des:7 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:8 https://pkg.jenkins.io/debian-stable binary/ Packages [19,8 kB]
Descargados 275 kB en 2s (120 kB/s)
Leyendo lista de paquetes... Hecho
usuario@proyectofinal:~$
```

Como vemos, se ha actualizado la lista de paquetes disponibles con el repositorio de Jenkins.



Procedemos a instalar Jenkins:



~\$ sudo apt-get install jenkins

```
usuario@proyectofinal:~$ sudo apt-get install jenkins
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  daemon net-tools
Se instalarán los siguientes paquetes NUEVOS:
  daemon jenkins net-tools
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 649 no actualizados.
Se necesita descargar 68,6 MB de archivos.
Se utilizarán 72,0 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```




Iniciamos el servicio de Jenkins y vemos su estado:



~\$ sudo systemctl start jenkins



~\$ sudo systemctl status jenkins

```
usuario@proyectofinal:~$ sudo systemctl start jenkins
usuario@proyectofinal:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Mon 2021-05-03 13:53:50 CEST; 3min 30s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 2322)
   CGroup: /system.slice/jenkins.service

may 03 13:53:48 proyectofinal systemd[1]: Starting LSB: Start Jenkins at boot time...
may 03 13:53:48 proyectofinal jenkins[10454]: Correct java version found
may 03 13:53:48 proyectofinal jenkins[10454]: * Starting Jenkins Automation Server jenkins
may 03 13:53:48 proyectofinal su[10488]: Successful su for jenkins by root
may 03 13:53:48 proyectofinal su[10488]: + ??? root:jenkins
may 03 13:53:48 proyectofinal su[10488]: pam_unix(su:session): session opened for user jenkins by (uid=0)
may 03 13:53:49 proyectofinal su[10488]: pam_unix(su:session): session closed for user jenkins
may 03 13:53:50 proyectofinal jenkins[10454]: ...done.
may 03 13:53:50 proyectofinal systemd[1]: Started LSB: Start Jenkins at boot time.
usuario@proyectofinal:~$
```



Habilitamos el Firewall y abrimos el puerto 8080:



~\$ sudo ufw enable

```
usuario@proyectofinal:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
```



~\$ sudo ufw allow 8080

```
usuario@proyectofinal:~$ sudo ufw allow 8080
Reglas actualizadas
Reglas actualizadas (v6)
```



~\$ sudo ufw status

```
usuario@proyectofinal:~$ sudo ufw status
Estado: activo

Hasta      Acción     Desde
-----
8080       ALLOW      Anywhere
8080 (v6)  ALLOW      Anywhere (v6)

usuario@proyectofinal:~$
```



Miramos la IP del equipo para acceder al panel de control de Jenkins:



~\$ ip a

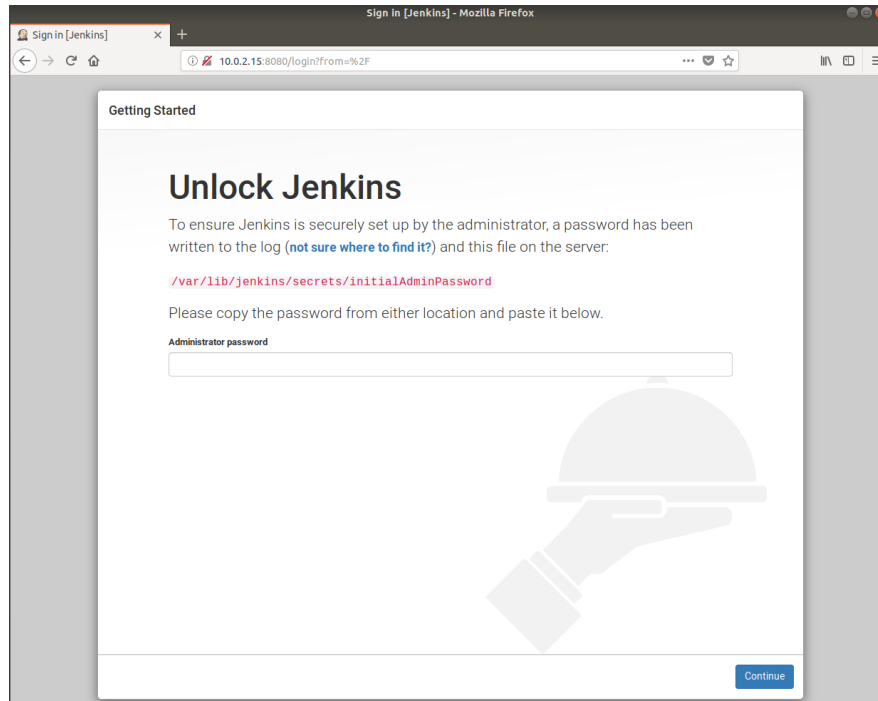
```
usuario@proyectofinal:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:b6:52:06 brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 81941sec preferred_lft 81941sec
   inet6 fe80::c287:3688:2864:d2e9/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
usuario@proyectofinal:~$
```



Accedemos al navegador y ponemos la IP del equipo añadiendo el puerto:



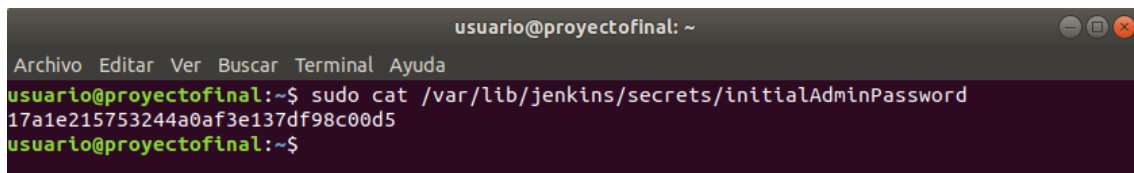
~\$ http://ip:8080



Debemos buscar la contraseña en la ruta que nos indica Jenkins y pegarla en el navegador:

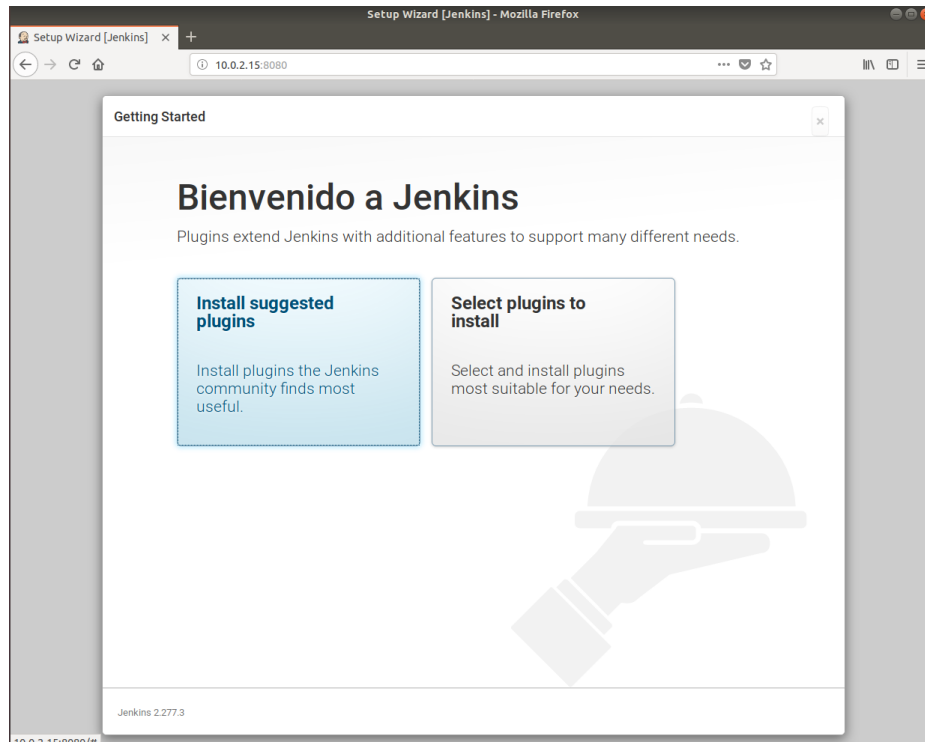


~\$ sudo cat /var/lib/Jenkins/secrets/initialAdminPassword

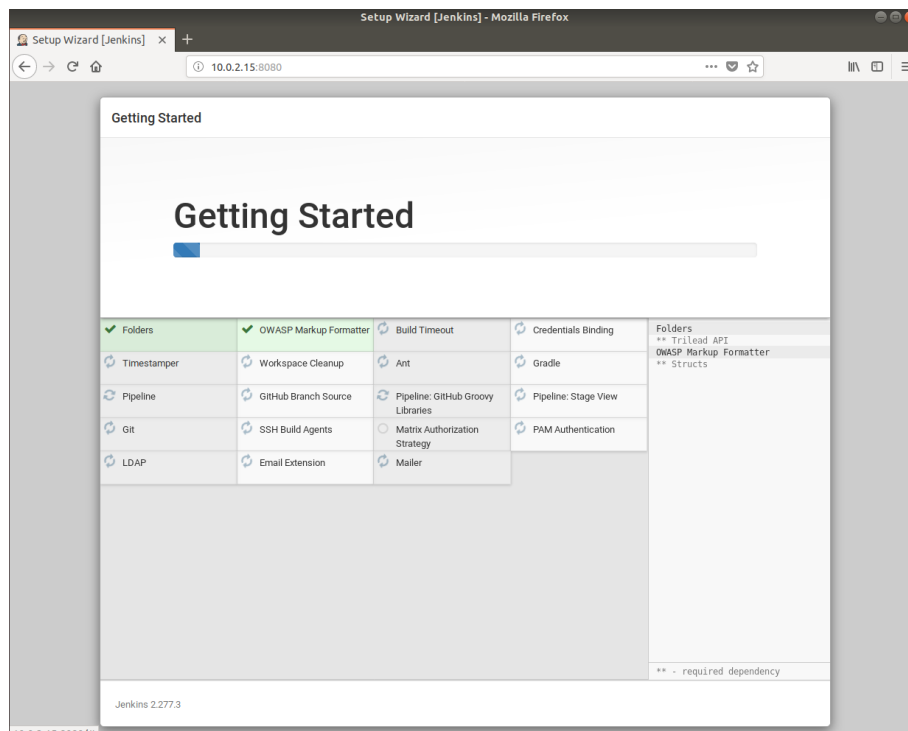




Una vez que ponemos la contraseña, nos deja elegir que plugins queremos instalar:



Elegimos los plugins recomendados y dejamos que se instalen:





Cuando se instalen, nos pedirá que creemos un usuario administrador:

The screenshot shows the 'Getting Started' section of the Jenkins Setup Wizard. The main heading is 'Create First Admin User'. Below it, there are input fields for 'Username' (filled with 'usuariojenkins'), 'Password' (masked with dots), 'Confirm password' (masked with dots), 'Full name' (filled with 'Usuario Jenkins'), and 'E-mail address' (filled with 'usuariojenkins@dominio.com'). At the bottom right, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The footer indicates 'Jenkins 2.277.3'.



Nos permite elegir la URL que queremos asignar a Jenkins, en este caso dejaremos la que está por defecto:

The screenshot shows the 'Getting Started' section of the Jenkins Setup Wizard. The main heading is 'Instance Configuration'. Below it, there is a 'Jenkins URL' field with the default value 'http://10.0.2.15:8080/'. Below the field, there is explanatory text: 'The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps. The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.' At the bottom right, there are two buttons: 'Not now' and 'Save and Finish'. The footer indicates 'Jenkins 2.277.3'.



Una vez aceptado todo, podremos acceder al panel de control de Jenkins:

Dashboard [Jenkins] - Mozilla Firefox

10.0.2.15:8080

Jenkins

search

Usuario Jenkins log out

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

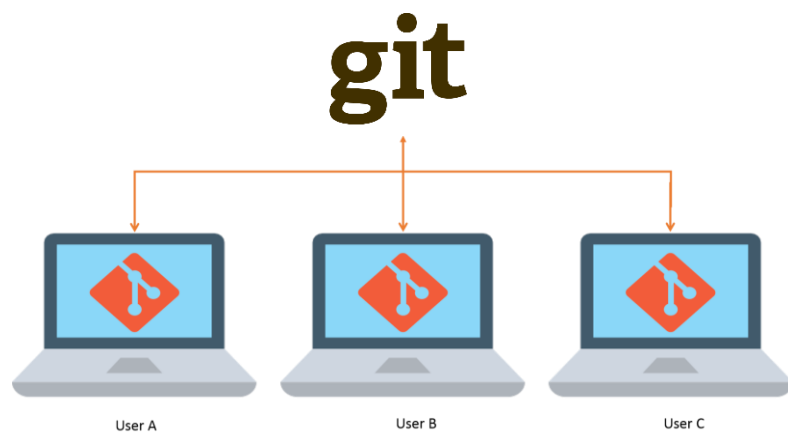
REST API Jenkins 2.277.3

3. *git*

Git es una herramienta de software libre que se encarga del control de versiones de forma distribuida.

Fue diseñada por Linus Torvalds y su lanzamiento fue en 2005.

Actualmente la supervisa Junio Hamano.



Su propósito es llevar un registro de todos los cambios en los archivos locales del ordenador, añadirlos al servidor remoto y poder coordinar el trabajo con varias personas, incluso trabajar a la vez.

3.1. Github

Github es una plataforma de gestión y organización basada en la nube que integra todas las funciones de Git.



Todos los usuarios pueden ver y gestionar cambios desde la interfaz gráfica que nos proporciona.

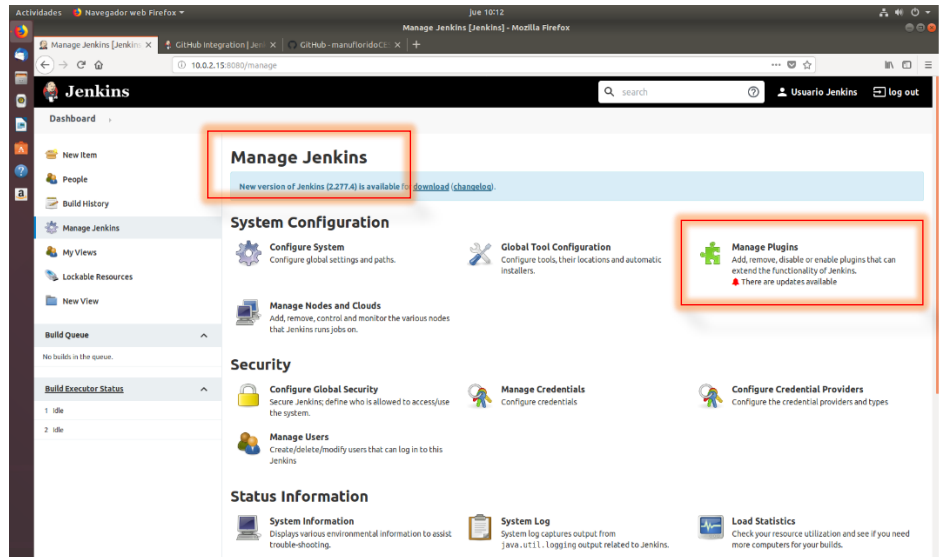
Esta interfaz facilita su uso para personas con pocos conocimientos.



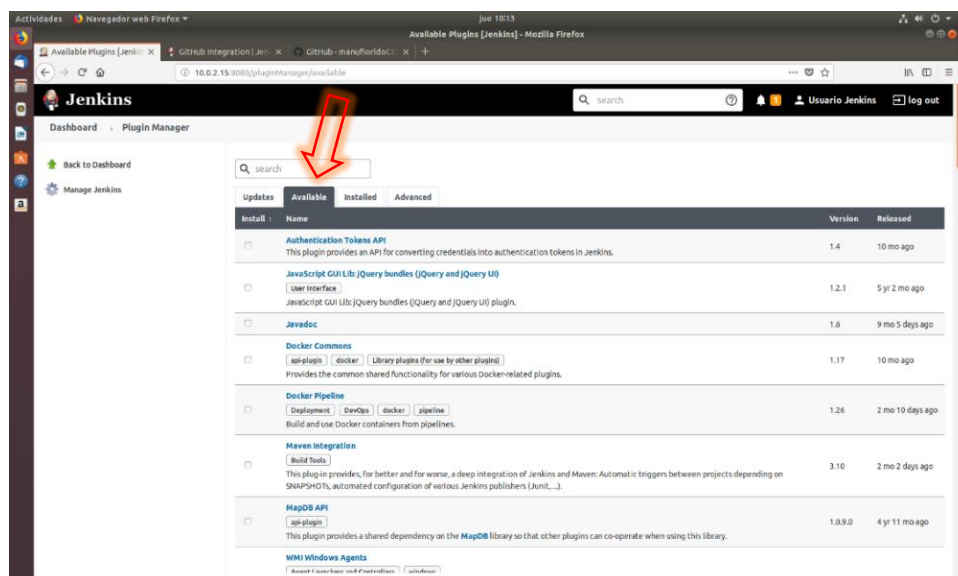
4. Instalacion de Git plugin en Jenkins



Pulsamos en Manage Jenkins y después en Manage Plugins para instalar el plugin de Git:



Pulsamos en Available (Disponible) y buscamos el plugin llamado Git plugin:





A continuación, abriremos un terminal e instalaremos Git en nuestro equipo:



~\$ sudo apt-get install git

```
usuario@proyectofinal: ~
Archivo Editar Ver Buscar Terminal Ayuda

usuario@proyectofinal:~$ sudo apt-get install git
[sudo] contraseña para usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  git git-man liberror-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 650 no actualizados.
Se necesita descargar 4.743 kB de archivos.
Se utilizarán 34,0 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.
17025-1 [22,8 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
1:2.17.1-1ubuntu0.8 [804 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:
2.17.1-1ubuntu0.8 [3.916 kB]
92% [3 git 3.815 kB/3.916 kB 97%] 508 kB/s 0s
```

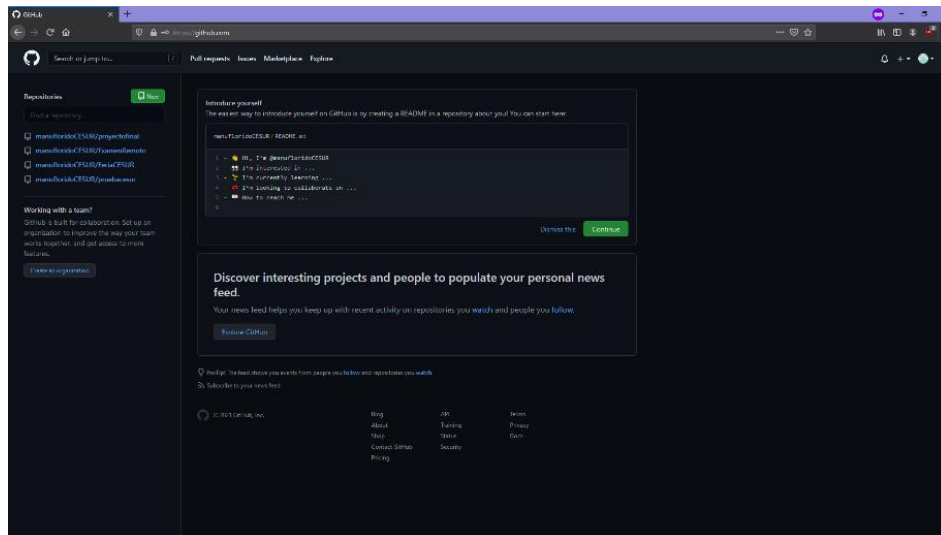


Una vez realizados estos pasos, podremos enlazar Jenkins y Git.

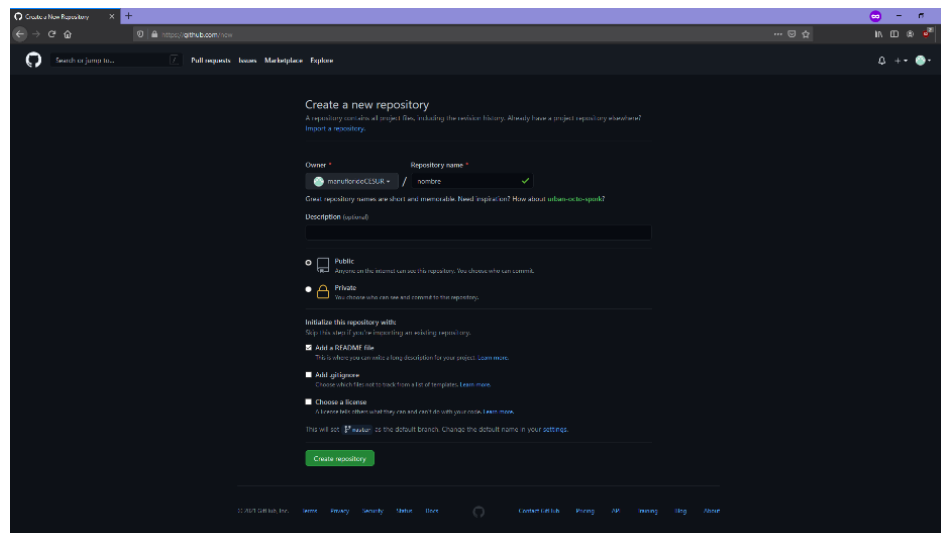
5. Enlace Jenkins con Git



Una vez realizados estos pasos, pasamos a GitHub. Debemos iniciar sesión y crear un repositorio:



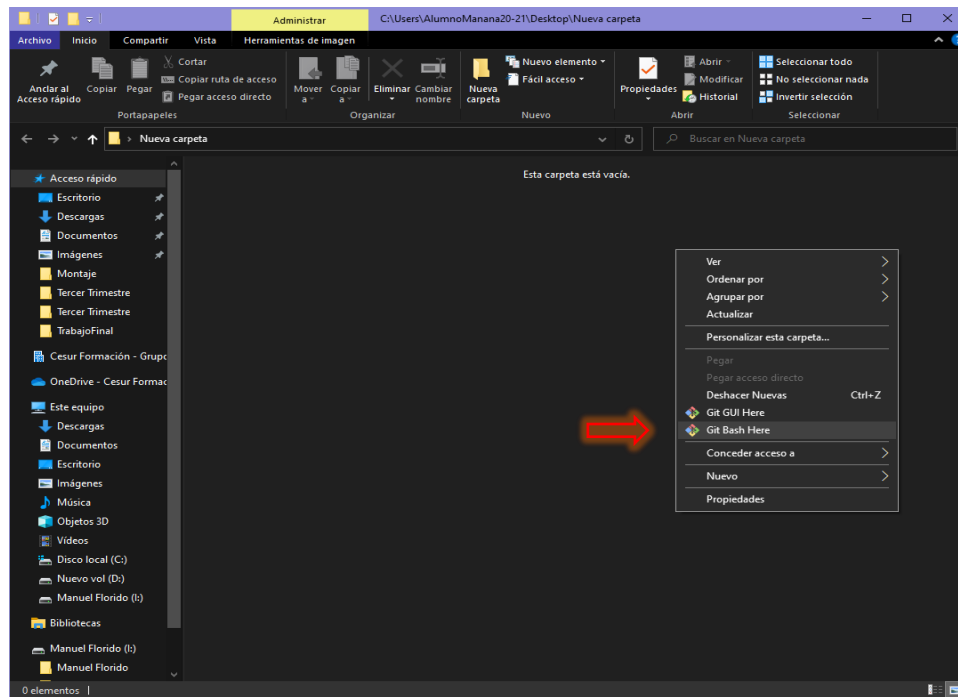
Pulsamos en New para crear el repositorio, le ponemos un nombre y lo ponemos en público:



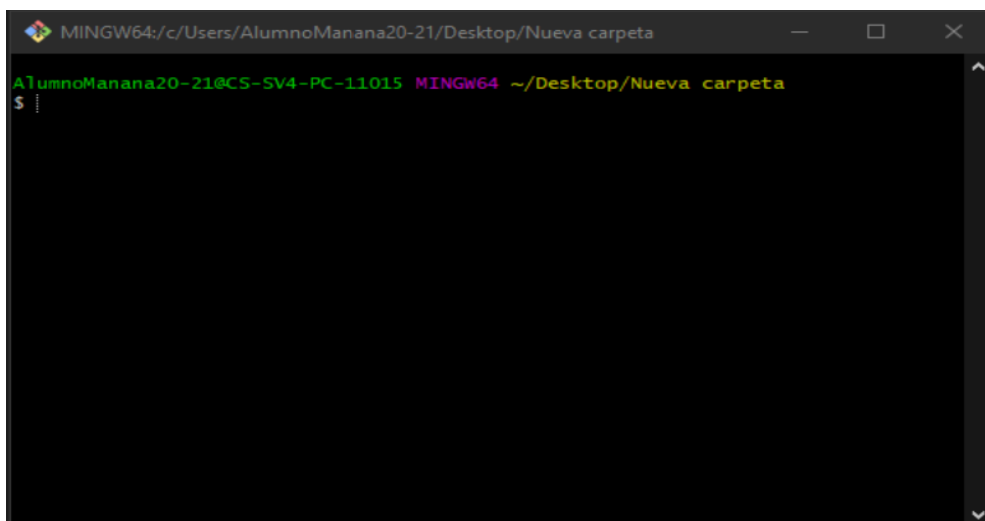


Cuando lo hayamos creado, pasaremos a subir archivos al repositorio para usarlo luego con Jenkins.

Crearemos una carpeta desde el equipo Windows (Git instalado), dentro de la carpeta haremos click derecho y pulsamos “Git Bash Here”:



Se abrirá una ventana como esta:





Inicializamos el repositorio:



~\$ git init

```
MINGW64:/c/Users/AlumnoManana20-21/Desktop/Nueva carpeta
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 ~/Desktop/Nueva carpeta
$ git init
Initialized empty Git repository in C:/Users/AlumnoManana20-21/Desktop/Nueva carpeta/.git/
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 ~/Desktop/Nueva carpeta (master)
$
```



Después configuraremos el usuario y el email de Git:



~\$ git config --global user.name "nombre"



~\$ git config --global user.email correo@correo.com

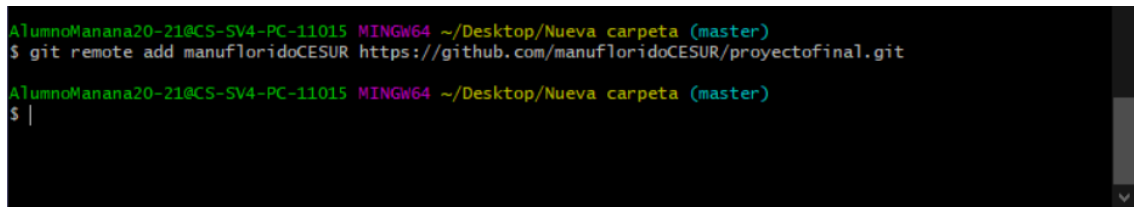
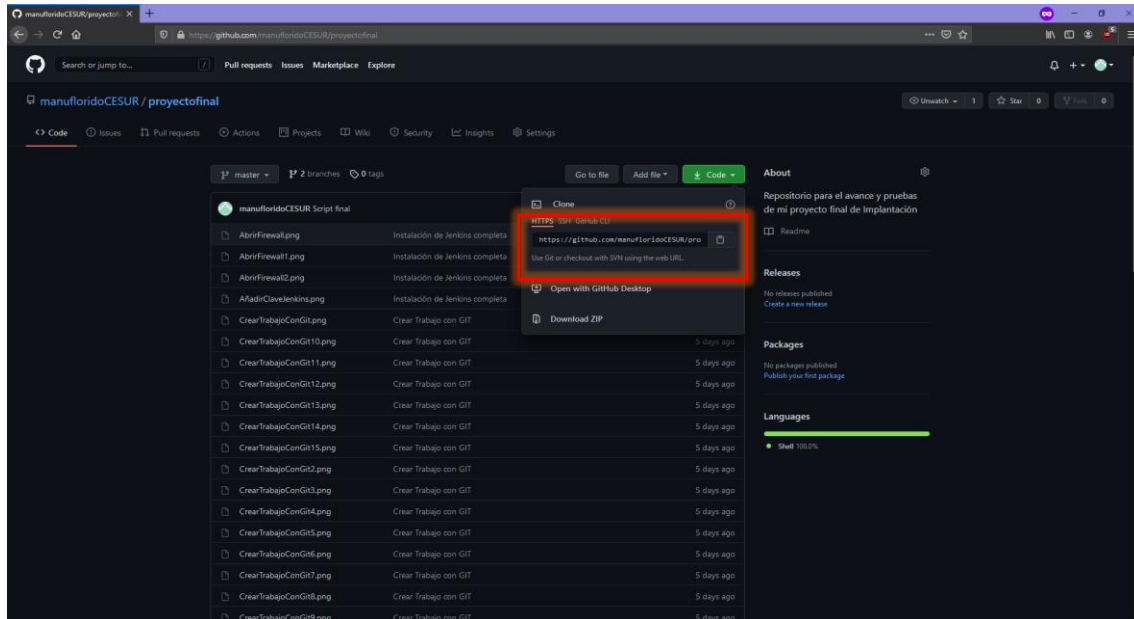
```
MINGW64:/c/Users/AlumnoManana20-21/Desktop/Nueva carpeta
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 ~/Desktop/Nueva carpeta (master)
$ git config --global user.name "manufloridoCESUR"
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 ~/Desktop/Nueva carpeta (master)
$ git config --global user.email manuel.ma39601@cesurformacion.com
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 ~/Desktop/Nueva carpeta (master)
$
```



Añadimos el repositorio de GitHub, cogiendo la URL del repositorio:



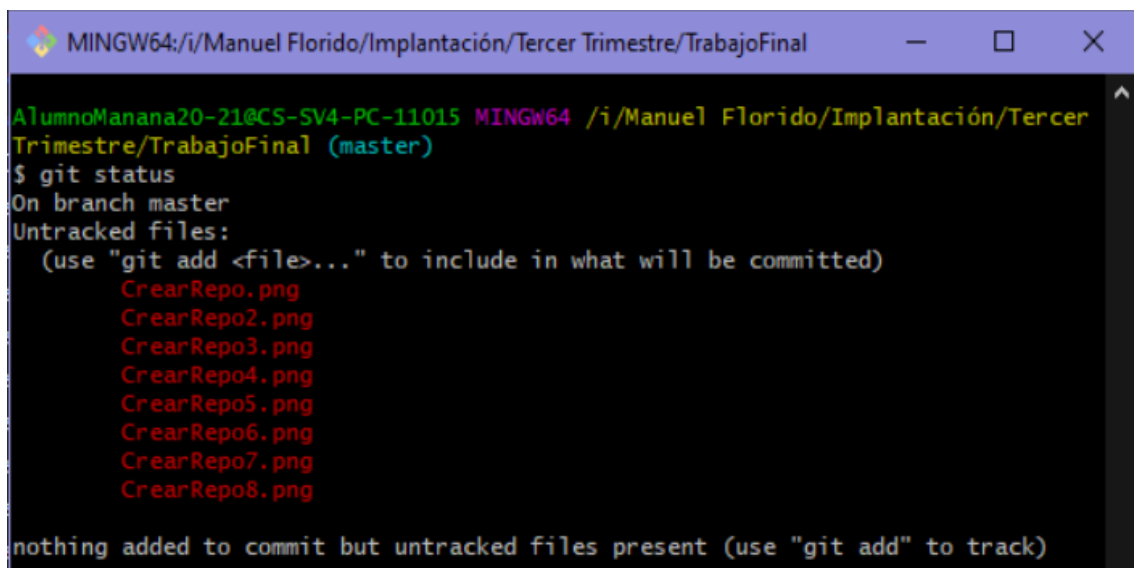
~\$ git remote add usuario URL



Miramos el estado del repositorio para saber si se pueden añadir archivos:



~\$ git status





Añadimos los archivos para posteriormente confirmarlos:



`~$ git add .` (El punto añade todos los archivos que tienen la posibilidad de añadirse)

```
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$ git add .

AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$
```



Confirmamos los archivos que hemos añadido:



`~$ git commit -m "Comentario"`

```
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$ git commit -m "Subir capturas"
[master b0f574f] Subir capturas
 8 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 CrearRepo.png
 create mode 100644 CrearRepo2.png
 create mode 100644 CrearRepo3.png
 create mode 100644 CrearRepo4.png
 create mode 100644 CrearRepo5.png
 create mode 100644 CrearRepo6.png
 create mode 100644 CrearRepo7.png
 create mode 100644 CrearRepo8.png

AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$
```



Para subir los archivos al repositorio de GitHub habrá que hacer un push:



`~$ git push usuario rama`

```
AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$ git push manufloridoCESUR master
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (13/13), 517.61 KiB | 18.49 MiB/s, done.
Total 13 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/manufloridoCESUR/proyectofinal.git
   b284640..bdb2b0e  master -> master

AlumnoManana20-21@CS-SV4-PC-11015 MINGW64 /i/Manuel Florido/Implantación/Tercer Trimestre/TrabajoFinal (master)
$
```



Una vez subido el script al repositorio de GitHub, podremos crear el trabajo de Jenkins:



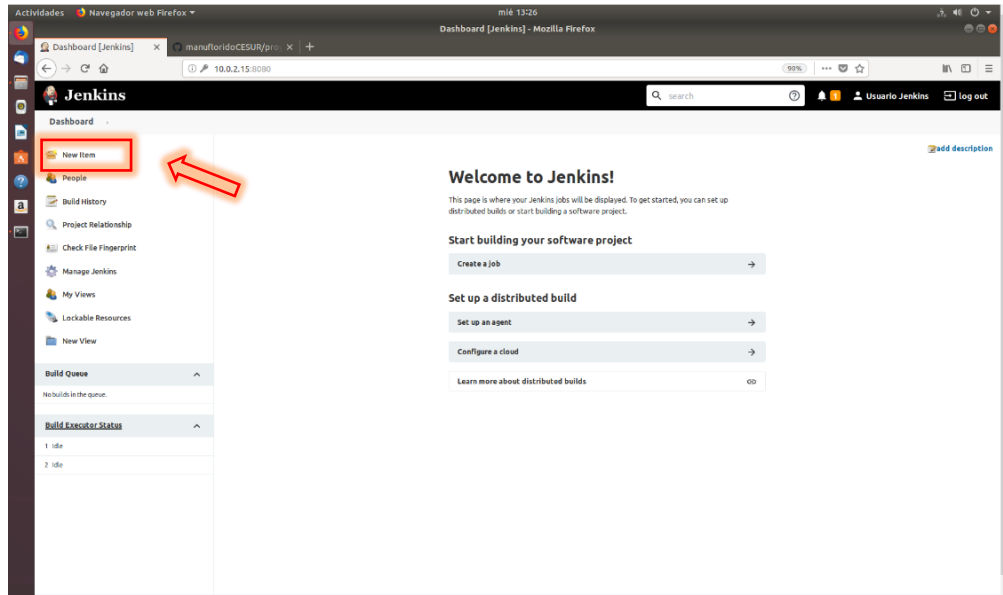
Este script ejecuta varias tareas de monitorización y pega su resultado en un archivo llamado monitorización.txt

```
script.sh
1  #!/bin/bash
2  echo "Hora del Sistema, Hora de Inicio, Usuarios y Carga del Procesador" > monitorizacion.txt
3  uptime >> monitorizacion.txt
4  echo "" >> monitorizacion.txt
5  echo "Procesos activos del sistema" >> monitorizacion.txt
6  ps -a >> monitorizacion.txt
7  echo "" >> monitorizacion.txt
8  echo "Usuarios conectados al sistema" >> monitorizacion.txt
9  w -h >> monitorizacion.txt
10 echo "" >> monitorizacion.txt
11 echo "Uso de la Memoria Fisica y de Intercambio" >> monitorizacion.txt
12 free -m >> monitorizacion.txt
13 echo "" >> monitorizacion.txt
14 echo "Comprobador del Sistema de Archivos" >> monitorizacion.txt
15 df -H >> monitorizacion.txt
```

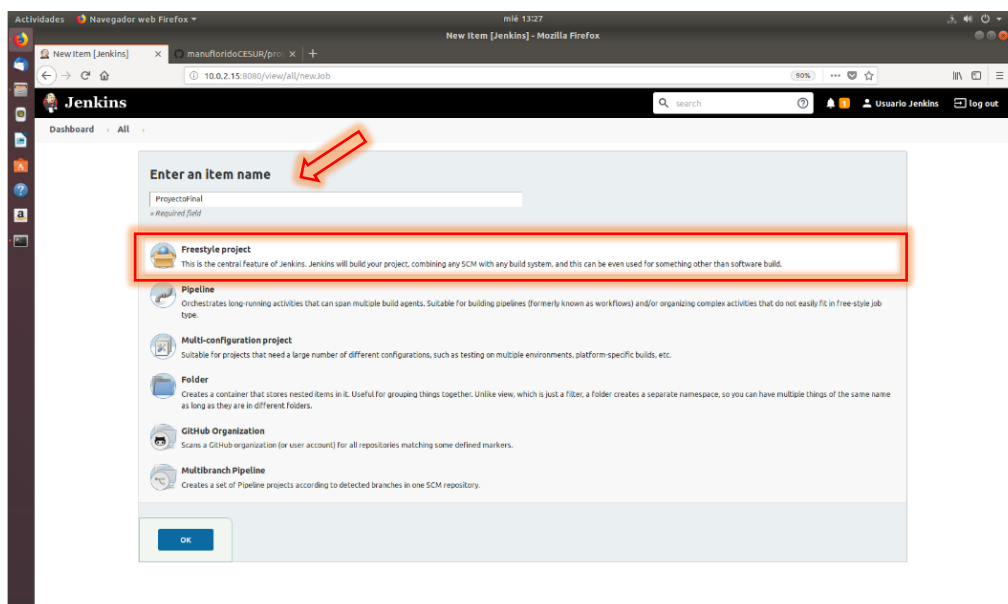
6. Crear trabajo de Jenkins



Para crear el trabajo de Jenkins iremos al panel de control y pulsaremos “New Item”:

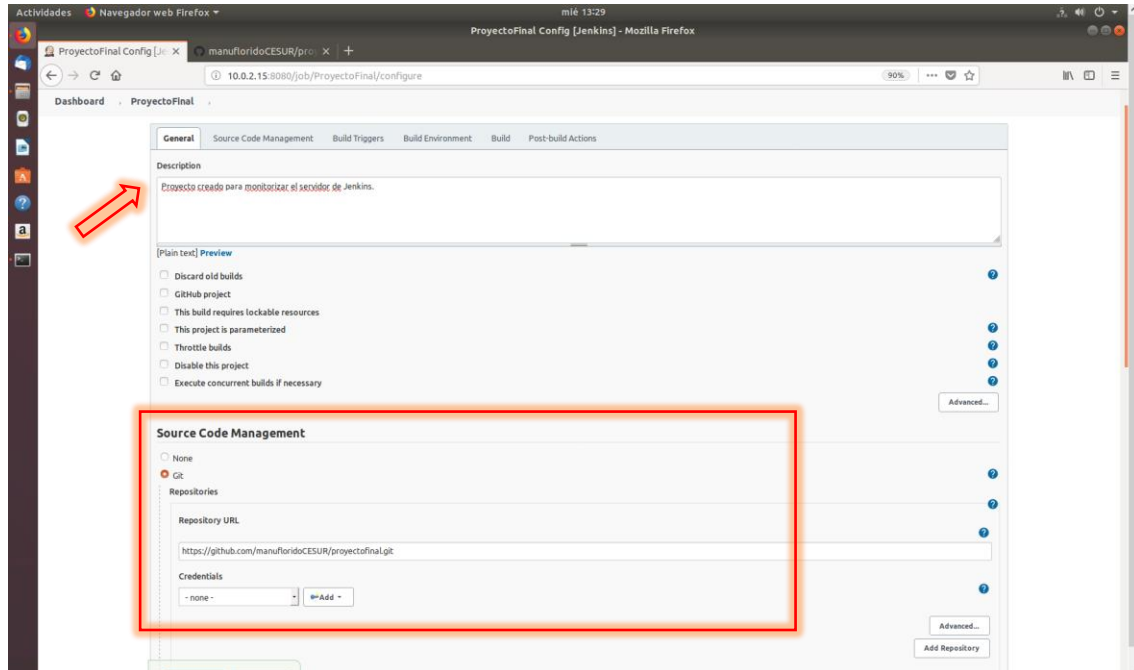


Escribimos el nombre del proyecto, elegimos “Freestyle project” y le damos a “Ok”:





Pulsamos en Git y escribimos la URL del repositorio de Git, también tenemos la posibilidad de escribir un comentario sobre nuestro proyecto:

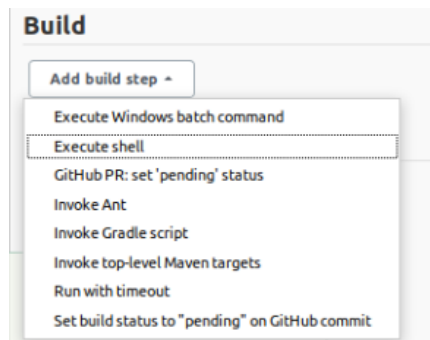


Para que el trabajo se ejecute periódicamente tenemos que marcar la opción de “Build periodically” y elegir cada cuanto tiempo va a ejecutarse. En este caso hemos puesto que se haga cada 15 minutos.





Después especificamos la acción que queremos que se realice. Para ejecutar el script necesitaremos un terminal:



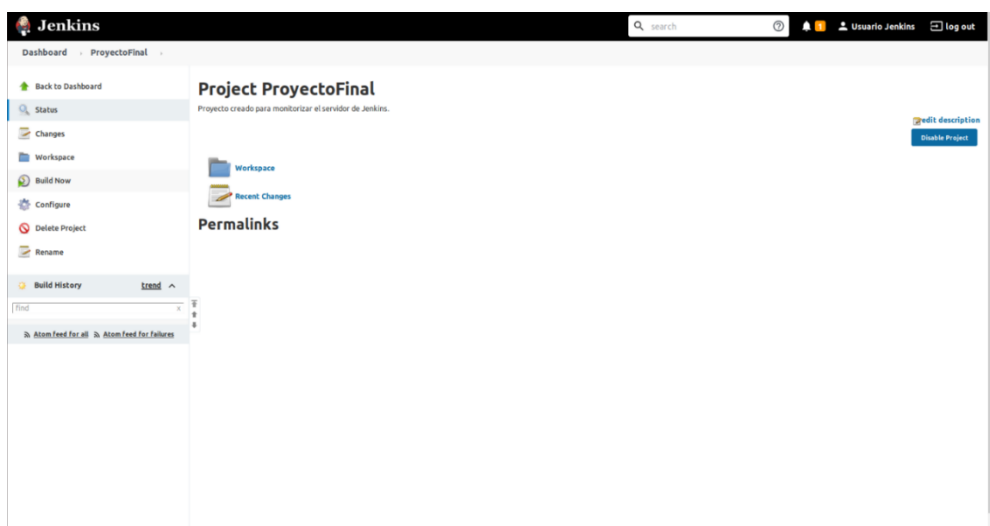
Se abrirá un cuadro de texto donde podremos escribir los comandos que necesitamos. Nosotros pondremos que ejecute el script que tenemos en el repositorio de GitHub:



Luego pulsaremos en “Apply” y en “Save” para que nuestro proyecto se quede guardado y aparezca en el Panel de Control de Jenkins.

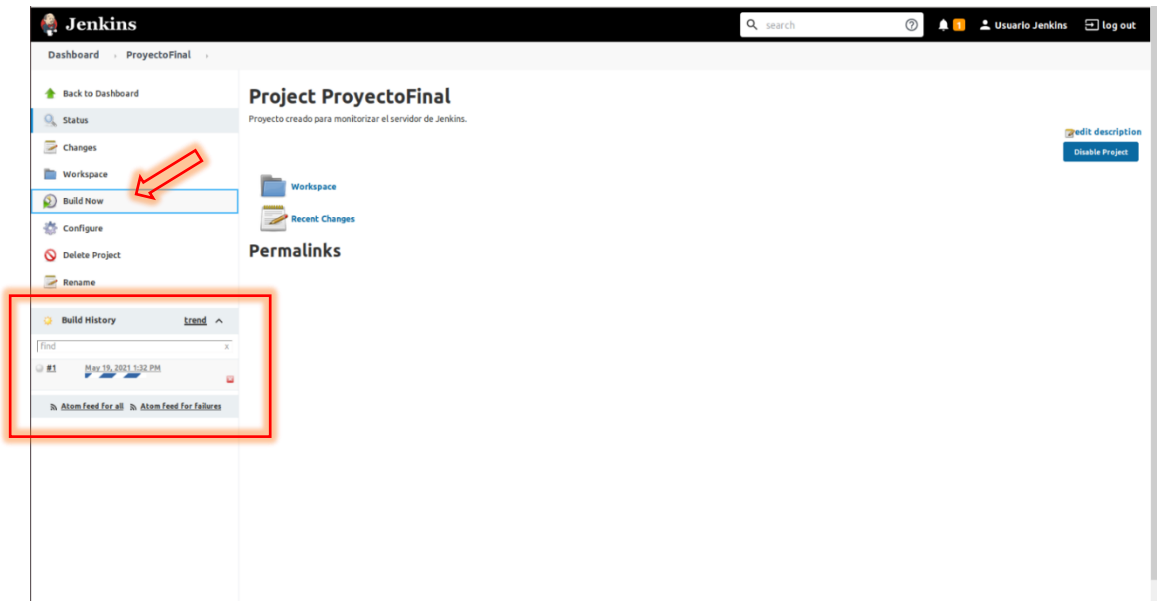


Aparecerá el menú de nuestro proyecto:

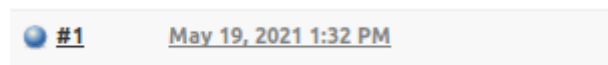




Para que se ejecute el proyecto sin que pase el tiempo establecido, pulsaremos en “Build Now” y veremos como en el historial se añade el número #1:



En el caso de que no haya habido fallos en el proceso, saldrá al lado del #1 un círculo. Si el círculo sale azul significa que está bien ejecutado y si sale rojo significa que ha habido fallos:

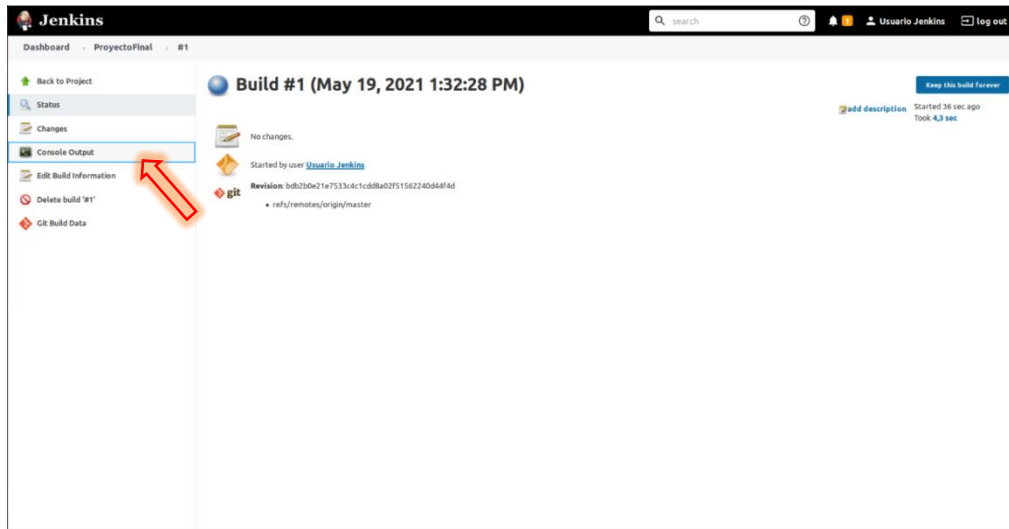


En este caso, como se puede observar, ha salido azul, por lo que significa que está bien ejecutado.





Podemos pulsar en el número y nos mostrará las características de nuestra Build. En el apartado “Console Output” veremos todos los pasos que ha seguido Jenkins para ejecutar nuestro script:



Veremos el usuario que ha ejecutado la Build, los comandos que ha ejecutado el plugin de Git (ha clonado el repositorio a su Workspace) y al final ha ejecutado el script:

Console Output

```
Lanzada por el usuario Usuario Jenkins
Running as SYSTEM
Ejecutando en el espacio de trabajo /var/lib/jenkins/workspace/ProyectoFinal
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/manufloridoCESUR/proyectofinal.git
> git init /var/lib/jenkins/workspace/ProyectoFinal # timeout=10
Fetching upstream changes from https://github.com/manufloridoCESUR/proyectofinal.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/manufloridoCESUR/proyectofinal.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/manufloridoCESUR/proyectofinal.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision bdb2b0e21e7533c4c1cdd8a02f51562240d44f4d (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f bdb2b0e21e7533c4c1cdd8a02f51562240d44f4d # timeout=10
Commit message: "Subida del script definitivo"
First time build. Skipping changelog.
[ProyectoFinal] $ /bin/sh -xe /tmp/jenkins4243027727354328898.sh
+ bash script.sh
Finished: SUCCESS
```



Ahora accederemos a la ruta del Workspace para ver si se ha realizado de todo el clonado del repositorio:

```

usuario@proyectofinal: /var/lib/jenkins/workspace/ProyectoFinal
Archivo Editar Ver Buscar Terminal Ayuda
usuario@proyectofinal:/var/lib/jenkins/workspace$ cd ProyectoFinal/
usuario@proyectofinal:/var/lib/jenkins/workspace/ProyectoFinal$ ls
AbrirFirewall1.png      InicioJenkins1.png
AbrirFirewall2.png      InicioJenkins2.png
AbrirFirewall.png       InicioJenkins3.png
AñadirClaveJenkins.png  InicioJenkins4.png
CrearRepo2.png          InicioJenkins.png
CrearRepo3.png          InstalarGit.png
CrearRepo4.png          InstalarJava2.png
CrearRepo5.png          InstalarJava.png
CrearRepo6.png          InstalarJenkins2.png
CrearRepo7.png          InstalarJenkins.png
CrearRepo8.png          metercomandoenarchivo.png
CrearRepo.png           monitorizacion.txt
CrearTrabajoConGit10.png PluginGit2.png
CrearTrabajoConGit11.png PluginGit3.png
CrearTrabajoConGit12.png PluginGit4.png
CrearTrabajoConGit13.png PluginGit5.png
CrearTrabajoConGit14.png PluginGit6.png
CrearTrabajoConGit15.png PluginGit7.png
CrearTrabajoConGit2.png  PluginGit8.png
CrearTrabajoConGit3.png  PluginGit.png
CrearTrabajoConGit4.png  ProyectoFinalJenkinsyGit.docx
CrearTrabajoConGit5.png  README.md
CrearTrabajoConGit6.png  script.sh
CrearTrabajoConGit7.png  SetupJenkins1.png
CrearTrabajoConGit8.png  SetupJenkins2.png
CrearTrabajoConGit9.png  SetupJenkins3.png
CrearTrabajoConGit.png   SetupJenkins4.png
EditarSourcesList2.png   SetupJenkins.png
EditarSourcesList3.png   Update.png
EditarSourcesList.png
usuario@proyectofinal:/var/lib/jenkins/workspace/ProyectoFinal$

```



Como podemos ver, está todo el repositorio clonado y a parte el script ha creado el archivo monitorizacion.txt:

```

usuario@proyectofinal: /var/lib/jenkins/workspace/ProyectoFinal
Archivo Editar Ver Buscar Terminal Ayuda
Usuarios conectados al sistema
usuario :0 :0 10:37 ?xdm? 5:08 0.02s /usr/lib/gdm3/gdm-x-session --run-script env
GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu

Uso de la Memoria Física y de Intercambio
total usado libre compartido búfer/caché disponible
Memoria: 1993 1587 88 58 317 299
Swap: 472 455 17

Comprobador del Sistema de Archivos
S.ficheros Tamaño Usados Disp Uso% Montado en
udev 1,1G 0 1,1G 0% /dev
tmpfs 210M 1,7M 208M 1% /run
/dev/sda1 11G 7,1G 2,9G 72% /
tmpfs 1,1G 28M 1,1G 3% /dev/shm
tmpfs 5,3M 4,1k 5,3M 1% /run/lock
tmpfs 1,1G 0 1,1G 0% /sys/fs/cgroup
/dev/loop0 104M 104M 0 100% /snap/core/11081
/dev/loop1 59M 59M 0 100% /snap/core18/1997
/dev/loop2 394k 394k 0 100% /snap/gnome-characters/708
/dev/loop3 148M 148M 0 100% /snap/gnome-3-26-1604/102
/dev/loop4 148M 148M 0 100% /snap/gnome-3-26-1604/70
/dev/loop5 69M 69M 0 100% /snap/gtk-common-themes/1515
/dev/loop6 2,4M 2,4M 0 100% /snap/gnome-system-monitor/157
/dev/loop7 16M 16M 0 100% /snap/gnome-logs/37
/dev/loop8 4,0M 4,0M 0 100% /snap/gnome-system-monitor/51
/dev/loop9 230M 230M 0 100% /snap/gnome-3-34-1804/66
/dev/loop10 14M 14M 0 100% /snap/gnome-characters/103
/dev/loop11 37M 37M 0 100% /snap/gtk-common-themes/319
/dev/loop12 2,7M 2,7M 0 100% /snap/gnome-calculator/884
/dev/loop13 105M 105M 0 100% /snap/core/10958
/dev/loop14 2,5M 2,5M 0 100% /snap/gnome-calculator/180
--Más-- (93%)

```

7. Crontab



Una vez comprobada la creación del archivo, crearemos una tarea programada en Crontab, para que copie el archivo a una carpeta del escritorio y sea más fácil acceder al archivo.



El primer paso es comprobar si tenemos instalado el paquete “cron”:



```
~$ sudo apt-get install cron
```

```
usuario@proyectofinal: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
usuario@proyectofinal:~$ sudo apt-get install cron  
[sudo] contraseña para usuario:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
cron ya está en su versión más reciente (3.0pl1-128.1ubuntu1).  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 655 no actualizados.  
usuario@proyectofinal:~$
```



Ejecutamos el comando para iniciar el Crontab:



```
~$ crontab -e
```



Veremos el archivo donde añadir una línea con el comando que queremos ejecutar y el tiempo que pasará antes de volver a ejecutar la orden. Añadiremos esta línea:

```
* /16 * * * * cp /var/lib/jenkins/workspace/ProyectoFinal/monitorizacion.txt /home/usuario/Escritorio/UbuntuMonitorizacion/
```



Cada 16 minutos se ejecutará la copia del archivo “monitorizacion.txt” a la carpeta “UbuntuMonitorizacion” ubicada en el escritorio, recordemos que cada 15 minutos se ejecuta el script en el servidor Jenkins. Si borramos del comando el /16, se ejecutará cada minuto.



Podemos ver los procesos que ejecutará Crontab con:

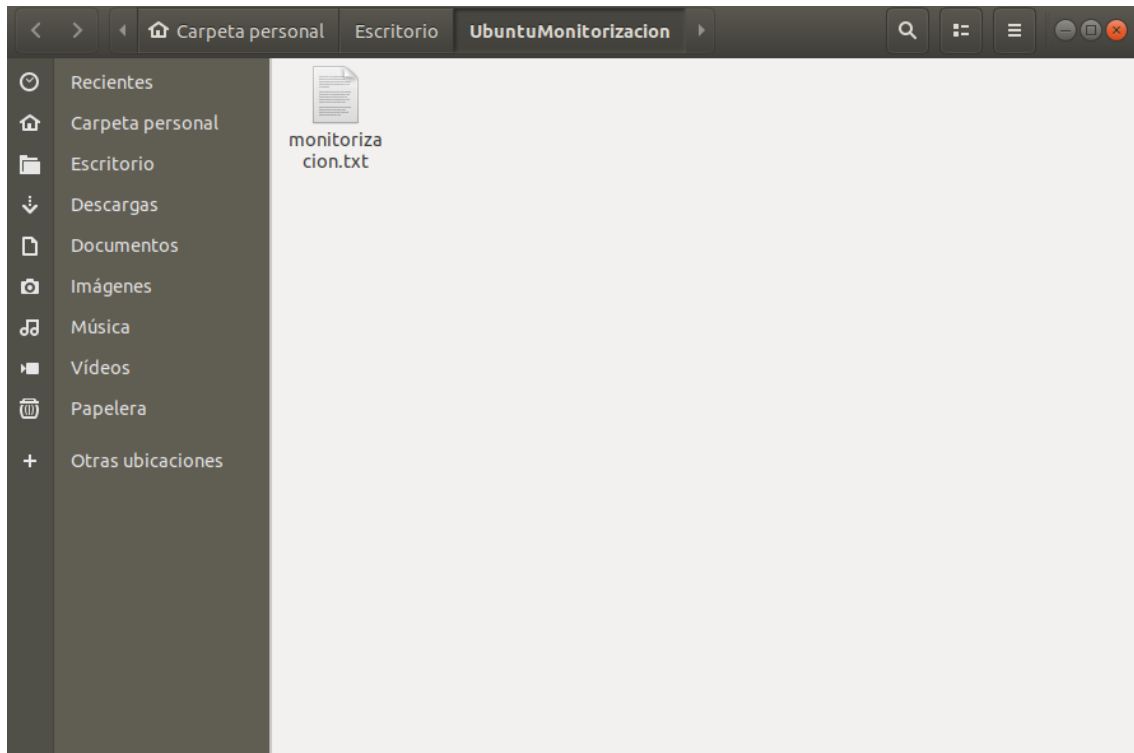


~\$ crontab -l

```
usuario@proyectoFinal:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/16 * * * * cp /var/lib/jenkins/workspace/ProyectoFinal/monitorizacion.txt /home/usuario/Escritorio/UbuntuMonitorizacion/
usuario@proyectoFinal:~$
```



Pasado el tiempo que hayamos especificado procedemos a ir al escritorio y mirar en la carpeta a ver si se ha copiado el archivo:





Pulsamos en el archivo y comprobamos que tenga la misma información que el archivo original:

```
Usuarios conectados al sistema
usuario :0 :0 10:37 ?xdm? 4:19 0.02s /usr/lib/gdm3/gdm-x-session --run-
script env GNOME_SHELL_SESSION_MODE=ubuntu gnome-session --session=ubuntu
```

Uso de la Memoria Física y de Intercambio

	total	usado	libre	compartido	búfer/caché	disponible
Memoria:	1993	1619	88	58	284	225
Swap:	472	344	128			

Comprobador del Sistema de Archivos

S.ficheros	Tamaño	Usados	Disp	Uso%	Montado en
udev	1,1G	0	1,1G	0%	/dev
tmpfs	210M	1,7M	208M	1%	/run
/dev/sda1	11G	7,1G	2,9G	71%	/
tmpfs	1,1G	28M	1,1G	3%	/dev/shm
tmpfs	5,3M	4,1k	5,3M	1%	/run/lock
tmpfs	1,1G	0	1,1G	0%	/sys/fs/cgroup
/dev/loop0	104M	104M	0	100%	/snap/core/11081
/dev/loop1	59M	59M	0	100%	/snap/core18/1997
/dev/loop2	394k	394k	0	100%	/snap/gnome-characters/708
/dev/loop3	148M	148M	0	100%	/snap/gnome-3-26-1604/102
/dev/loop4	148M	148M	0	100%	/snap/gnome-3-26-1604/70
/dev/loop5	69M	69M	0	100%	/snap/gtk-common-themes/1515
/dev/loop6	2,4M	2,4M	0	100%	/snap/gnome-system-monitor/157
/dev/loop7	16M	16M	0	100%	/snap/gnome-logs/37
/dev/loop8	4,0M	4,0M	0	100%	/snap/gnome-system-monitor/51
/dev/loop9	230M	230M	0	100%	/snap/gnome-3-34-1804/66
/dev/loop10	14M	14M	0	100%	/snap/gnome-characters/103
/dev/loop11	37M	37M	0	100%	/snap/gtk-common-themes/319
/dev/loop12	2,7M	2,7M	0	100%	/snap/gnome-calculator/884
/dev/loop13	105M	105M	0	100%	/snap/core/10958
/dev/loop14	2,5M	2,5M	0	100%	/snap/gnome-calculator/180
/dev/loop15	656k	656k	0	100%	/snap/gnome-logs/103
tmpfs	210M	29k	209M	1%	/run/user/121
tmpfs	210M	0	210M	0%	/run/user/122
/dev/loop16	59M	59M	0	100%	/snap/core18/2066
tmpfs	210M	37k	209M	1%	/run/user/1000

Texto plano ▾ Anchura del tabulador: 8 ▾ Ln 1, Col 1 ▾ INS

8. Bibliografía



<https://www.guru99.com/jenkins-github-integration.html>

<https://www.youtube.com/watch?v=bGqS0f4Utn4>

<https://www.youtube.com/watch?v=rwbN6jDCXo>

<https://www.hostinger.com/tutorials/how-to-install-jenkins-on-ubuntu/>

<https://linuxize.com/post/how-to-install-jenkins-on-ubuntu-18-04/>

<https://www.jenkins.io/doc/book/installing/>

<https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project>

<https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-18-04-es>

<https://www.guru99.com/download-install-jenkins.html>

<https://phoenixnap.com/kb/install-jenkins-ubuntu>

<https://www.techrepublic.com/article/how-to-install-jenkins-on-ubuntu-server-18-04/>

<https://www.youtube.com/watch?v=1jNXEWTBgDo>

<https://www.youtube.com/watch?v=63HEKFh8T2c>

<https://www.redeszone.net/2017/01/09/utilizar-cron-crontab-linux-programar-tareas/>

<https://geekytheory.com/programar-tareas-en-linux-usando-crontab>

https://www.linuxtotal.com.mx/?cont=info_admon_006

<https://www.youtube.com/watch?v=s4wEmAFwhsg>

<https://www.youtube.com/watch?v=bGqS0f4Utn4>

<https://www.youtube.com/watch?v=h8gbk7coq5g>

<https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project>

<https://huongdanjava.com/install-git-plugin-in-jenkins.html>

<https://www.thegeekstuff.com/2016/10/jenkins-git-setup/>