Práctica 2 - L2

EN DEBIAN
-sudo apt update o instalamos sudo apt install rsync
-cd
-mkdir dir1 dir2
–touch dir1/file{1..100}

#Hacemos una salva
-rsync -r –progress dir1/ dir2 # con esto copiamos todo lo de dentro de dir1 en dir2, -r hace #que si hay un directorio dentro de dir1 copia lo de dentro del archivo tambien, y dir1/ la / #hace que copiemos todos los archivos menos los archivos ocultos que habria que poner #dir1/.

ahora vamos a hacer una copia remota de dir1 en Alma con ssh:

ssh root@192.168.56.110 #si hemos cambiado el puerto ponemos -p 22022

-logout # Nos salimos porque no hace falta, era recordatorio

EN ALMA
-dnf install rsync

EN DEBIAN
-mkdir dir3
-touch dir3/file{1..200}
-rsync -re “ssh #si cambiamos el puerto -p *nºdelpuerto*” –progress /home/jpeiper/dir3/. root@192.168.56.110:/home/root/dirCopiaRemota
aquí hemos copiado algo que teníamos guardado en debian lo hemos copiado en almalinux

REPASANDO GIT

-git init -> Para crear un repositorio nuevo
-git help <comando> -> Manual del comando
-git status
-git add
-git commit
-git log –all –graph –decorate
-git checkout <commit-hash>
-git push
-git fetch
-git pull
-git merge
-git branch

Trabajo realizado por: Javier Peinado Pérez

-git clone

#Cada vez que se crea un commit se le asigna un SHA-1 de 40 caracteres.
#El HEAD corresponde al nodo (commit) del working directory
#Git usa punteros y lo que guardan son diferencias entre versiones

Volvemos a Alma

-sudo dnf install git
-mkdir prueba
-cd prueba
-git init
-ls -la #para ver el nuevo directorio oculto .git
-nano ~/.gitconfig
#dentro ponemos:
[user]
	name= Javier Peinado Perez
	email= jpeiper@correo.ugr.es
[push]
	default= matching
[core]
	editor= nano
#guardamos y salimos
#tambien podemos usar directamente el comando:
-git config –global user.name “Javier Peinado Perez”
-git config –global user.email “jpeiper@correo.ugr.es”

#seguimos
-touch .gitignore #sirve por si no quiero que git incluya en los commits algún archivo o tipo
#de archivo en concreto
-nano .gitignore #y aquí metemos *.doc o lo que sea que queramos ignorar, un archivo en
#concreto o lo que sea
-touch holamundo.txt
-nano holamundo.txt
#dentro ponemos:
Linea 1 de puntos
-git status # nos sale que ni el .gitignore ni el holamundo.txt están siendo trackeados
-git add holamundo.txt .gitignore
ahora tenemos todo lo add en la zona de stage
-git commit #si lo ponemos así tal cual se nos abre el editor que hayamos puesto en el
#.gitconfig, si lo dejamos vacío no se creará el commit, así que escribimos lo que sea
#descriptivo, guardamos y salimos
-git log # Y nos pone el commit que acabamos de crear y lo que hemos escrito dentro

Vuelta al repaso

Con git commit -a hace un git add . y commit al mismo tiempo
git commit -m “Mensaje descriptivo que queramos”

Trabajo realizado por: Javier Peinado Pérez

Vuelta a alma

hacemos una modificación a holamundo.txt, hacemos un add y un commit
-git log –all –graph –decorate #vemos que nos salen unidos los commits etc

#volvemos a modificar el holamundo pero sin hacer ningun commit
-git diff holamundo.txt #lo que hace es decirnos la diferencia entre el último holamundo.txt #en commit y el hayamos modificado y esté en stage (previo a commit)
-git diff <HASH_ID de un commit> holamundo.txt #te hace la diferencia entre el HEAD #(último commit) y el commit al que corresponda el HASH_ID

-git checkout holamundo.txt #lo que hace es devolver a holamundo al último commit (el #HEAD)

#si tenemos hemos hecho un git add de algo un me arrepiento, para quitar un archivo del # stage usamos git reset y esto lo saca del stage pero lo deja modificado

-mkdir ~/configuraciones
-cd ~/configuraciones/
-git init
-sudo cp -a /etc/ssh/sshd_config ./
-sudo chown $USER:$USER ssh_config #para cambiar el propietario del ssh_config al #usuario actual
-sudo chmod 644 ssh_config
-git add ssh_config
-git commit -m “First commit”
-git branch config-puertos #creamos una nueva rama llamada config-puertos
-git log –all –graph –decorate
-git checkout config-puertos #nos movemos a esta nueva rama
-git branch #para ver en que rama estamos
-nano ssh_config #buscamos la línea Port y escribimos: **Port: 44044**
-git commit -m “Puerto modificado”
#si hacemos un log veremos que aparece el nuevo commit en la rama config-puerto
-git checkout master #para volver a ponernos a la rama master
#en esta rama no está cambiado el puerto al 44044 está el que estuviera antes
-git checkout -b config-acceso #con -b creamos la nueva rama a la que nos movemos
-nano ssh_config # vemos en las líneas comentadas y buscamos PermiRootLogin, lo #descomentamos y lo ponemos a yes
-git add ssh_config
-git commit -m “Configurar acceso root yes”
-git log –all –graph –decorate
-git checkout master
-git merge config-puertos #esto actualiza la rama master con las modificaciones que #estaban en config-puertos, por lo que ahora el puerto es 44044
-git merge config-acceso #lo mismo con esta rama
#con git log –all –graph –decorate veremos bonito como se unen las ramas

Trabajo realizado por: Javier Peinado Pérez

#git stash permite guardar temporalmente lo que tengas en stage por si no queremos hacer
#un commit todavía y por ejemplo queremos cambiar de rama