1. If you were able to push the code in your branch to the given repo then you answered well.
2. Script

#!/bin/bash

echo "Ente the Type "Directory" or "File" ,as per your search"

read DirectoryORFile

echo "Type "Directory" or "File" ,as per your search"

read name

echo "Enter the location of File or Directory"

read location

echo "Enter the number days created before "

read d

case "$DirectoryORFile" in

"Directory")

if [ -d $location/$name ];

then

echo "Directory exists at $location"/"$name"

echo "it will be deleted if entered time matches"

cd $location

if [[ $(find . -name "$name" -mtime $d) ]]

then

sudo find . -name "$name" -mtime $d -exec rm -rf {} \;

else

echo "Directory found at $location"/"$name but time does not match"

fi

else

echo "Directory not found"

fi

;;

"File")

if [ -f $location/$name ];

then

echo "File exists at $location"/"$name"

echo "it will be deleted if entered time matches"

cd $location

if [[ $(find . -name "$name" -mtime $d) ]]

then

find . -name "$name" -mtime $d -exec rm -rf {} \;

else

echo "File found at $location"/"$name but time does not match"

fi

else

echo "File not found "

fi

;;

Esac

###Another Way

echo "Please enter Filename or Directory"

read fname

echo "Please enter date like (Apr 1)"

read fdate

echo "Enter the location"

read location

cd $location

if [[ -d "$location/$fname" ]]; then

echo "$fname is a directory"

tfile=`find . -name $fname 2>/dev/null`

dfile=`ls -ld $tfile|grep " $fdate "|awk -F " " '{print $NF}'`

echo $tfile

echo $dfile

rm -rf $dfile

elif [[ -f $fname ]]; then

echo "$fname is a file"

tfile=`find . -name $fname 2>/dev/null`

dfile=`ls -ld $tfile|grep " $fdate "|awk -F " " '{print $NF}'`

echo $tfile

echo $dfile

rm -r $dfile

else

echo "$fname is not valid"

exit 1

fi

1. **Script**

#!/bin/bash  
FILE=$(grep -hr "/" /etc/passwd | head -4 > /tmp/file) # This will first 4 users and redirect to the /tmp/file

FILENAME="/tmp/file"

while IFS=: read -r username password userid groupid home shell

do

echo "username = " "$username"

echo "password = " "$password"

echo "userid = " "$userid"

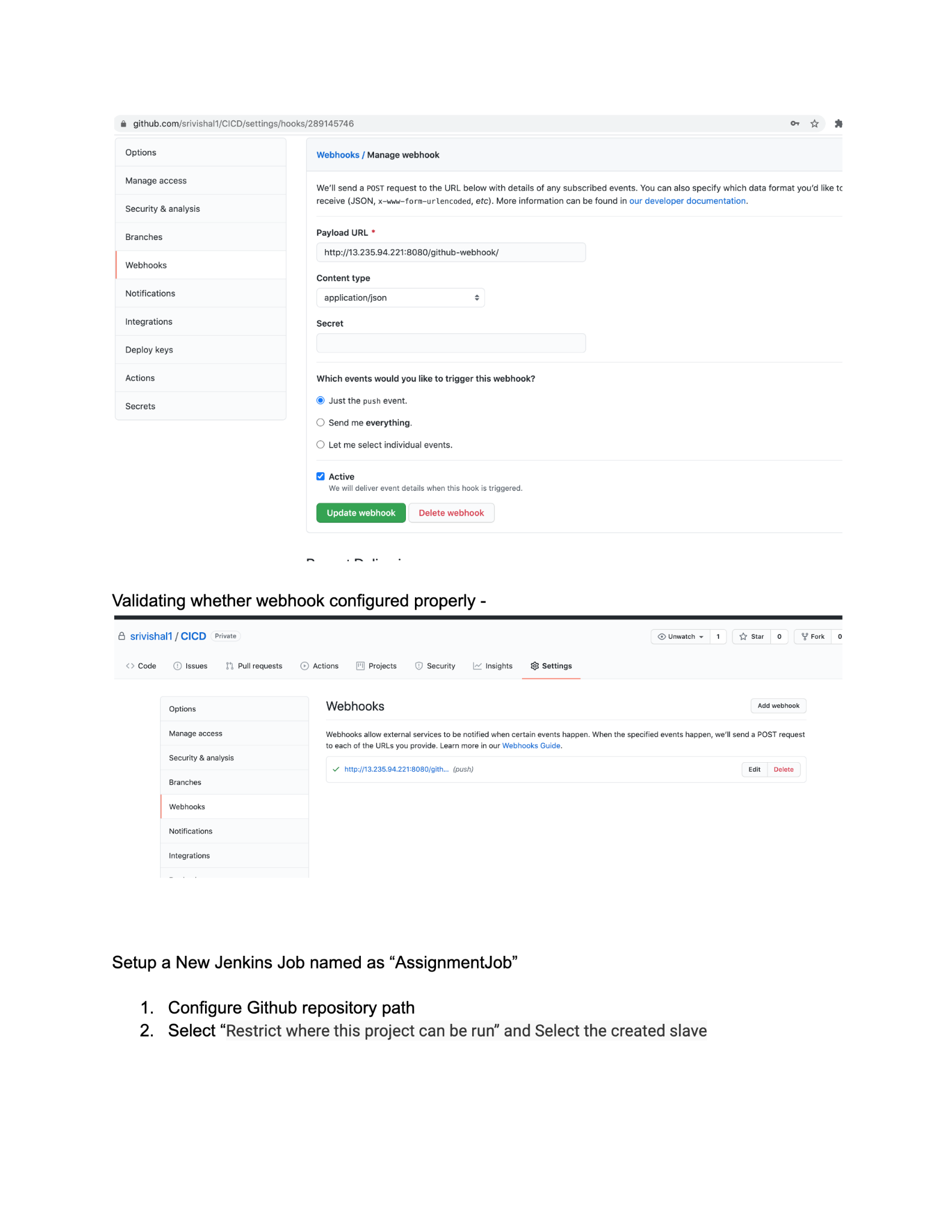
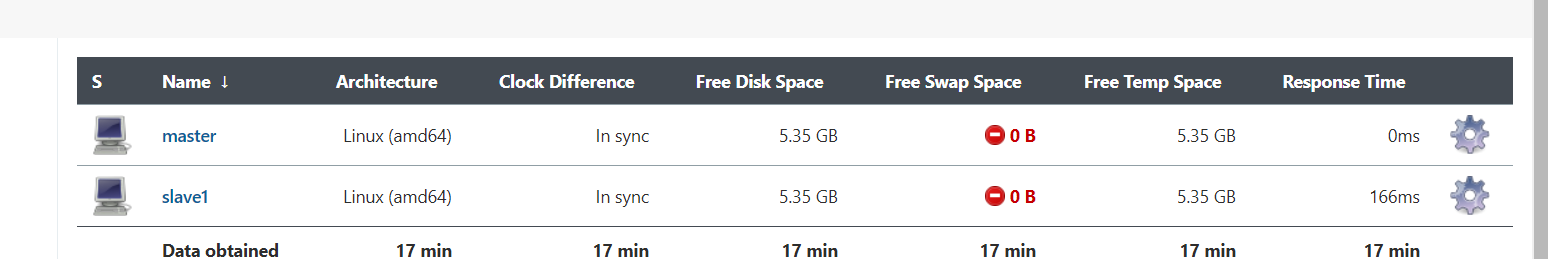
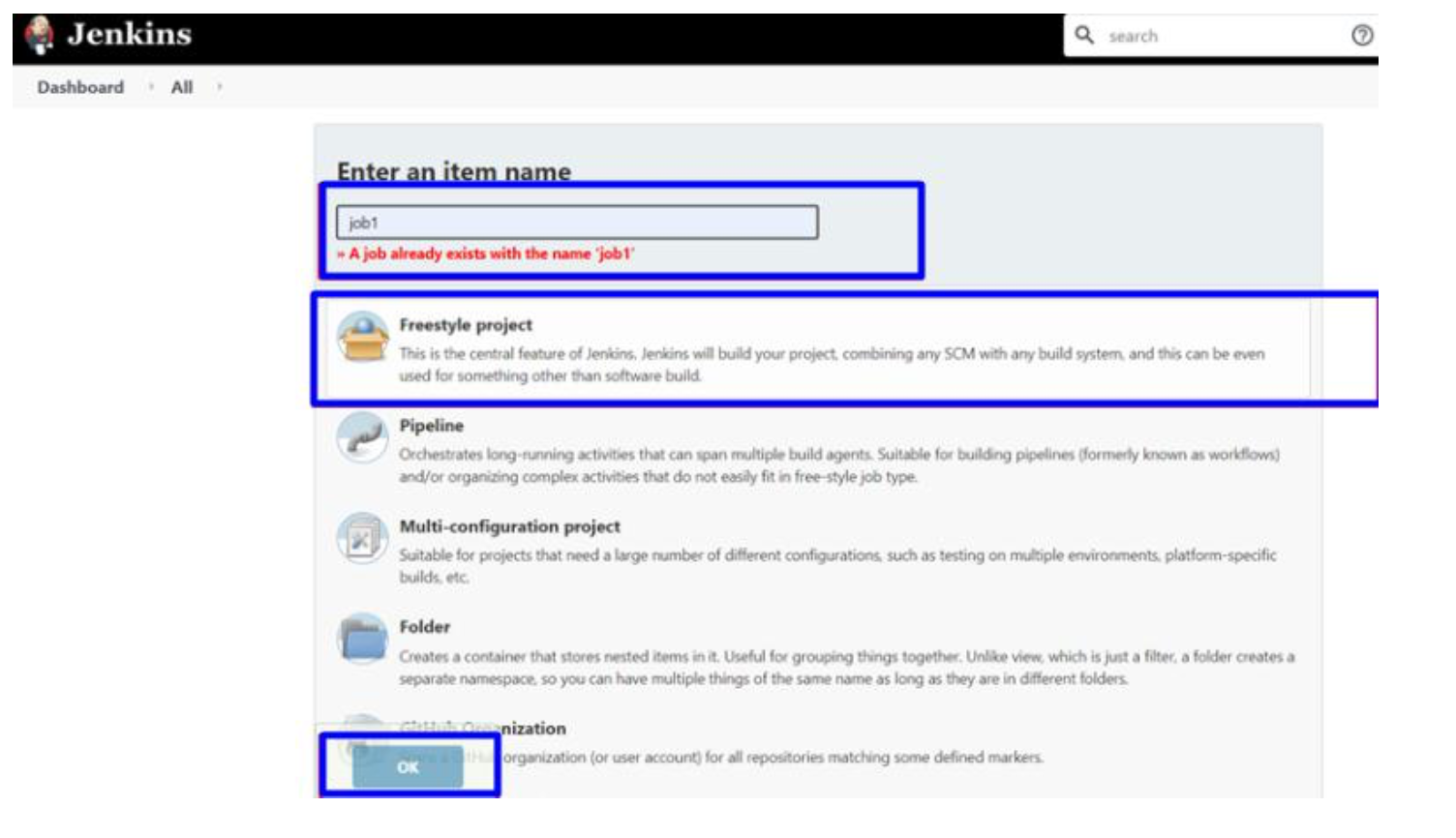
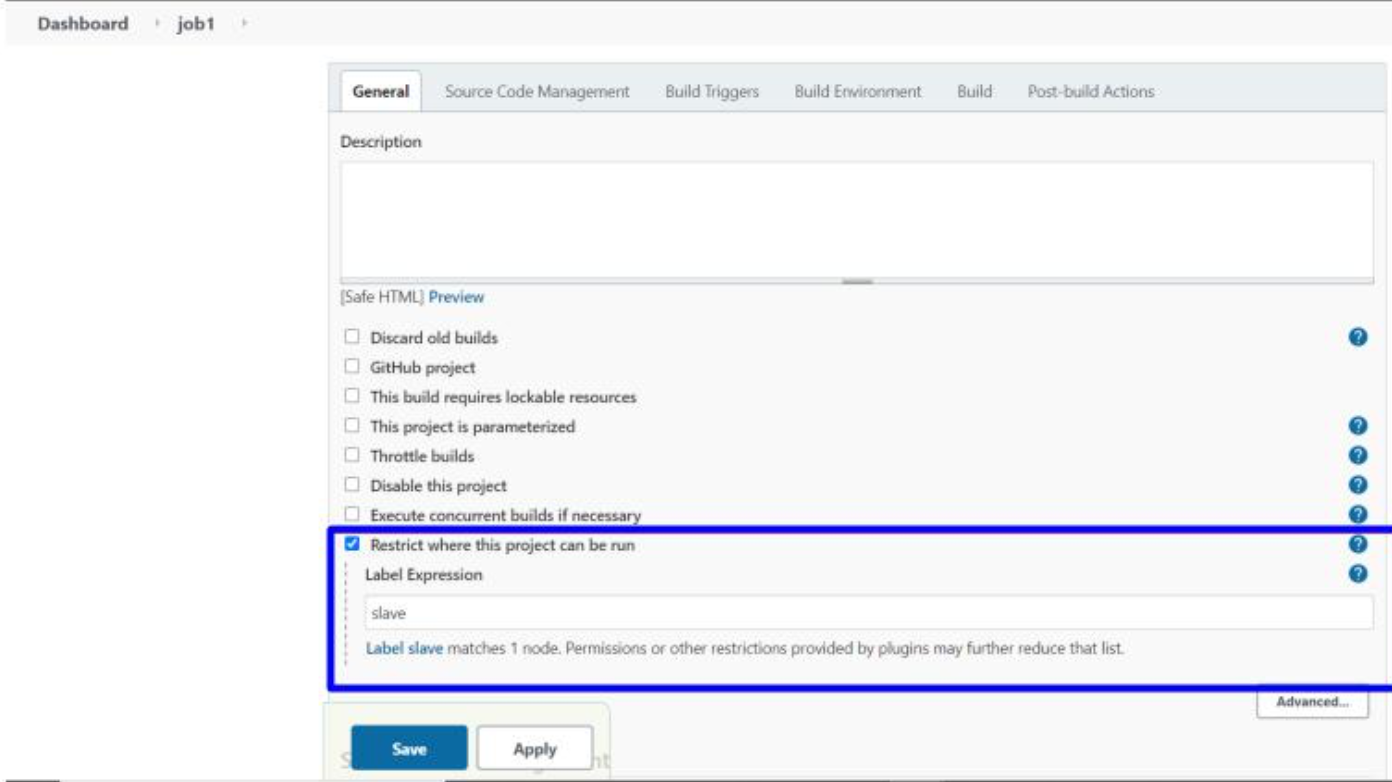
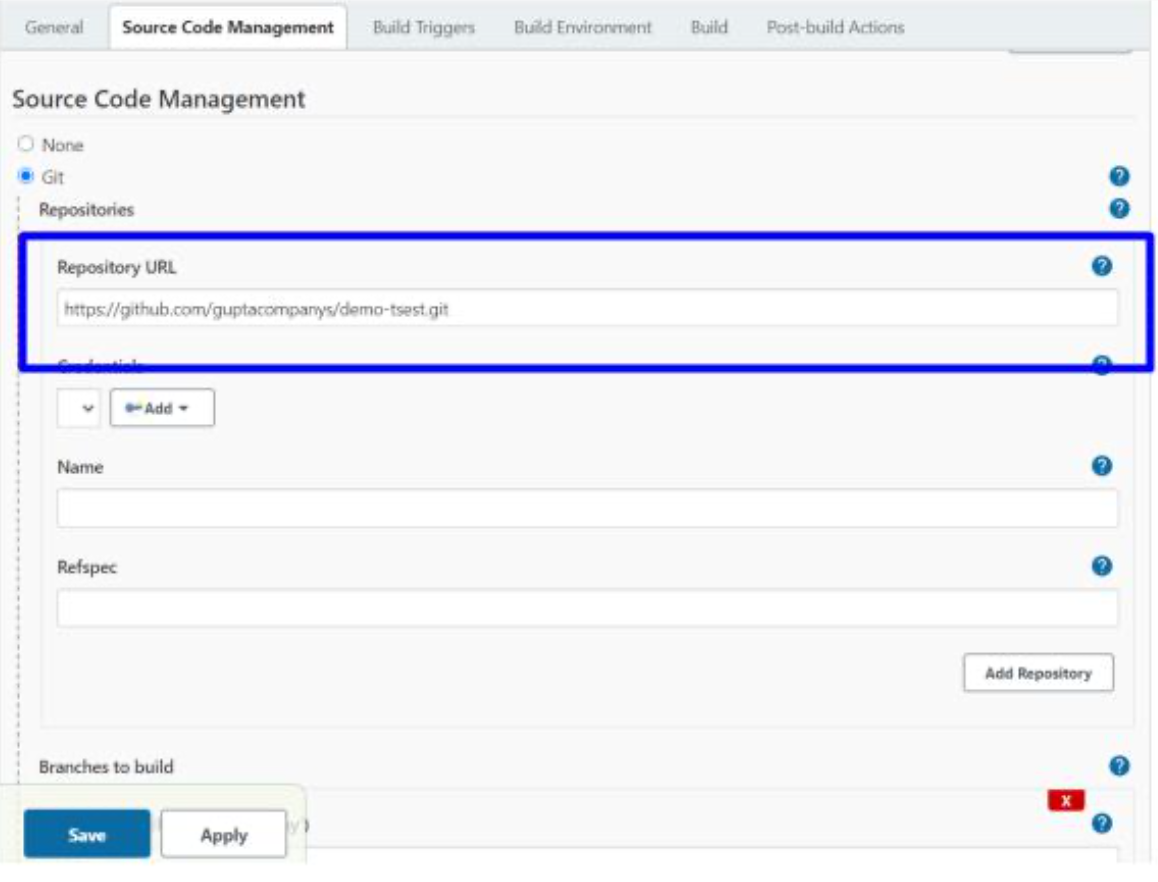
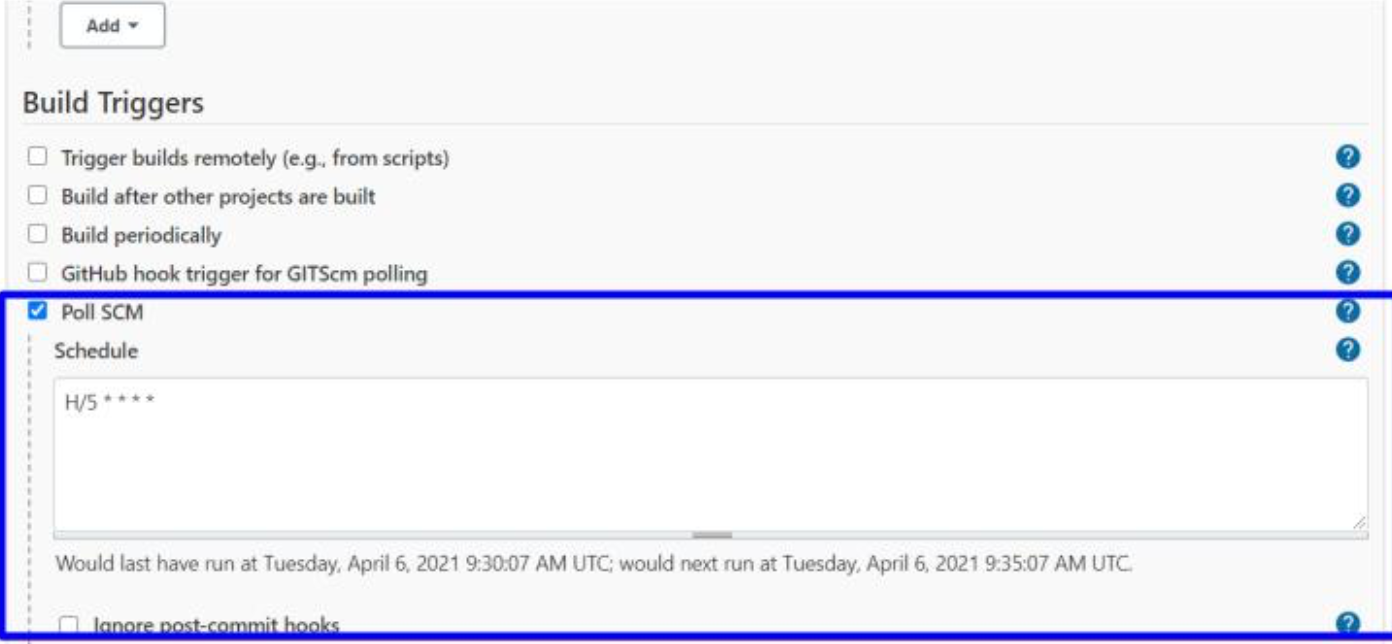
echo "groupid = " "$groupid"

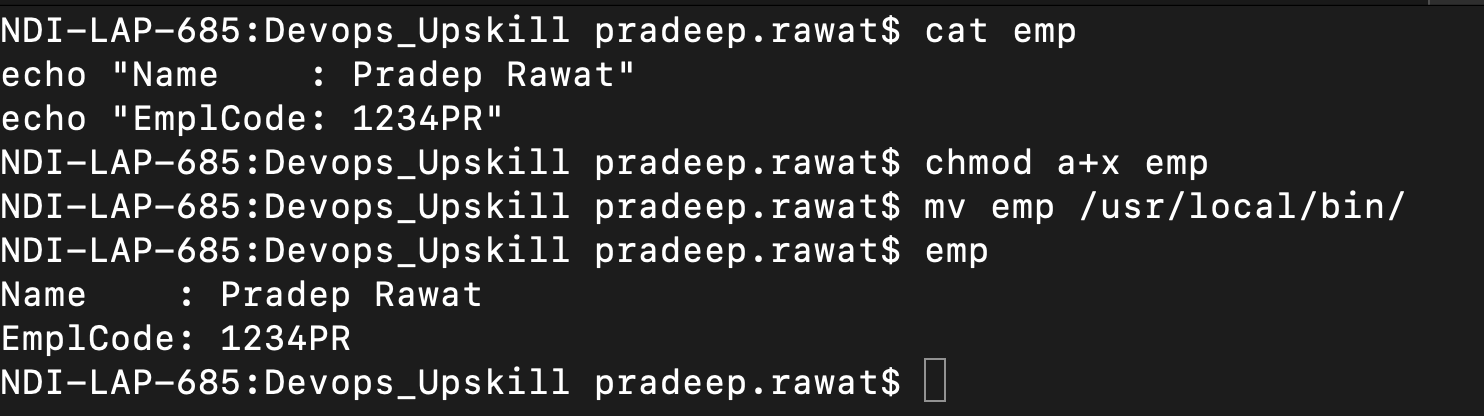
echo "home = " "$home"

echo "shell = " "$shell"

done < $FILENAME

## OR Another Method  
cat /etc/passwd | grep -v '^#' | head -n 4 |awk -F ':' '{ print " \n username= " $1 "\n password= " $2 " \n userid= " $3 " \n groupid= " $4 "\n userinfo= " $5 "\n home= " $6 " \n shell= " $7}'

1. **Steps for Jenkins Job.**  
   a) Configure github webhook with Jenkins  
     
   b) Create the slave configuration  
   c) Create a free style Job with any name.  
     
   d) Configure the job to be bound with a managed selected slave.  
     
     
   e) Configure git repo.  
     
     
     
   f) Configure the Poll SCM to check every 5 min job.  
   
2. commands  
   # touch credentials # create the empty file  
   # ls -lhrta credentials # check the permissions  
   # chmod o-r credentials # This removes read permission from others.  
   # setfacl -m u:<username>:r credentials # to set the read permission to the required user  
   # getfacl credentials # To check the new permissions.



**Note:** There are multiple ways to get these tasks done and few of them I have mentioned. Maybe you have a different approach to achieve it but before choosing any approach you should see which is appropriate for the current scenarios.