#!/bin/bash

while [ TRUE ]

do

read -p "Enter Username 1:" USER1

read -s -p "Please Enter The Password:" PASSWORD

if [ -z $USER1 ]; then

echo "Please Enter A Valid Username."

else

EXUSER=$(cat /etc/passwd | cut -d ":" -f 1 | grep -w "$USER1")

echo $EXUSER

if [[ "$EXUSER" == "$USER1" ]]; then

echo "User Exists, please enter a unique username"

else

useradd -m $USER1

chown $USER1:$USER1 /home/$USER1

echo "$USER1:$PASSWORD" | sudo chpasswd

echo "User $USER1 sucessfully created..!!"

fi

fi

done

-------------------------------------

#!/bin/bash

while [ TRUE ]; do

read -p "Enter Username 1:" USER1

if [ -z $USER1 ]; then

echo "Please Enter A Valid Username."

else

EXUSER=$(cat /etc/passwd | cut -d ":" -f 1 | grep -w "$USER1")

echo $EXUSER

if [[ "$EXUSER" == "$USER1" ]]; then

echo "User Exists, please enter a unique username"

else

useradd -m $USER1

ALPHA='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

ALPHACHAR=$(echo $ALPHA | fold -w 4 | shuf | head -1)

SPEC='!@#$%^&\*()\_'

SPECCHAR=$(echo $SPEC | fold -w1 | shuf | head -1)

PASSWORD=India${ALPHACHAR}${SPECCHAR}$RANDOM

chown $USER1:$USER1 /home/$USER1

echo "$USER1:$PASSWORD" | sudo chpasswd

echo "User $USER1 and password is $PASSWORD sucessfully created..!!"

passwd -e $USER1

fi

fi

done

------------

**#!/bin/bash**

**read -p "Please Enter The Username:" USERNAME**

**read -s -p "Please Enter The Password:" PASSWORD**

**useradd $USERNAME**

**echo $PASSWORD | passwd --stdin ${USERNAME}**

**usermod -aG wheel $USERNAME**

**passwd -e $USERNAME**

**User1.sh**

#!/bin/bash

set -xe

read -p "Enter the Username:" USER\_NAME

SPEC='!@#$%^&\*()\_'

SPECCHAR=$(echo ${SPEC} | fold -w1 | shuf | head -1)

PASSWORD=${RANDOM}$(date +%s%N)${SPECCHAR}

useradd -m ${USER\_NAME}

echo ${PASSWORD} | passwd --stdin ${USER\_NAME}

echo "Sucessfully Created user ${USER\_NAME} with password as ${PASSWORD}"

**Test1.sh**

#!/bin/bash

#set -x -e

read -p "Please enter the user name:" USER\_NAME

PASSWORD=$(curl -sL https://helloacm.com/api/random/?n=14)

#read -s -p "Please enter the password:" USER\_PASS

#useradd -m ${USER\_NAME} -p ${PASSWORD}

useradd -m ${USER\_NAME}

echo ${PASSWORD} | passwd --stdin ${USER\_NAME}

echo "The Username is ${USER\_NAME} & Password is ${PASSWORD}"

**IF STATEMENT:**

=================================================

[root@ip-10-1-1-89 tmp]# cat usercreate.sh

#!/bin/bash

#THIS SCRIPT WILL CHECK FOR THE USER NAME BEFORE CREATING IT.

#Read Username from the keyboard.

read -p "Please enter the username:" USER\_NAME

#Create Complex Password.

SPEC='!@#$%^&\*()\_'

SPECCHAR=$(echo ${SPEC}|fold -w1|shuf|head -1)

PASSWORD=India@${RANDOM}${SPECCHAR}

#Check if the users Exists and if exists thow error.

EXUSER=$(cat /etc/passwd |grep -i ${USER\_NAME} |cut -d ":" -f 1)

#EXUSER=$(cat /etc/passwd |cut -d ":" -f 1 | grep -i ${USER\_NAME})

echo "The existing user name is ${EXUSER} ."

if [[ ${EXUSER} == ${USER\_NAME} ]]

then

echo "User aleady exists. Please use a diffrent username..!!"

else

echo "Creating the new user...!!"

sleep 3s

useradd -m ${USER\_NAME}

echo ${PASSWORD} | passwd --stdin ${USER\_NAME}

passwd -e ${USER\_NAME}

#Print the USername and Password.

echo "Username is ${USER\_NAME} Password is ${PASSWORD} "

Fi

**FOR LOOP:**

=================================================

[root@ip-10-1-1-100 ~]# ./[users.sh](http://users.sh/) India1 India2 India3

#!/bin/bash

USERS=${@}

#SPEC='!@#$%^&\*()\_+'

#SPEC\_CHAR=$(echo ${SPEC}|fold -w1|shuf|head -1)

#PASSWORD=$(date +%s%N)${RANDOM}${SPEC\_CHAR}

for USER in ${USERS}

do

SPEC='!@#$%^&\*()\_+'

SPEC\_CHAR=$(echo ${SPEC}|fold -w1|shuf|head -1)

PASSWORD=$(date +%s%N)${RANDOM}${SPEC\_CHAR}

useradd ${USER}

echo ${PASSWORD} | passwd --stdin ${USER}

echo "User ${USER} is successfully created."

echo ${PASSWORD}

passwd -e ${USER}

done

Other Way of FOR Loop:

------------------------------------

END=5

for ((i=1;i<=END;i++)); do

echo $i

Done

[root@ip-10-1-1-110 ~]# for X in sree harsha veerapalli

> do

> echo "Hello ${X}"

> done

Hello sree

Hello harsha

Hello veerapalli

${@} - for multiple arguments

${#} - gives argument count passed

${?} - Exit code 0 means success and any other is the error.

**WHILE LOOP-1**

=================================================

while [ "$stats" -gt 300 -o "$stats" -eq 0 ]

while [ "$stats" -gt 300 ] || [ "$stats" -eq 0 ]

[root@ip-10-1-1-85 tmp]# cat [while-usercreate.sh](http://while-usercreate.sh/)

#!/bin/bash

while :

do

#Ask for the username

read -p "Please enter the username:" USER\_NAME

#Check of the Username Exists

EXUSER=$(cat /etc/passwd | cut -d ":" -f 1 | grep -i ${USER\_NAME})

if [[ ${EXUSER} = ${USER\_NAME} ]]

then

echo "User ${USER\_NAME} already exists."

echo "Exit Code is ${?}."

else

#Generate a complex password

SPEC='!@#$%^&\*()'

SPECHAR=$(echo $SPEC | fold -w1 | shuf | head -1)

PASSWORD=Capita${RANDOM}${SPECHAR}

useradd -m ${USER\_NAME}

echo $PASSWORD |passwd --stdin ${USER\_NAME}

echo "${USER\_NAME} is sucessflly created.Password is ${PASSWORD}"

echo "Exit Code is ${?}."

fi

done

**WHILE LOOP -2**

=========================================================

#!/bin/bash

read -p "Do you want to create users(Yes/No):" CHOICE

while [[ ${CHOICE} = "Yes" ]] || [[ ${CHOICE} = "yes" ]]

do

#Ask for the username

read -p "Please enter the username:" USER\_NAME

#Check of the Username Exists

EXUSER=$(cat /etc/passwd | cut -d ":" -f 1 | grep -i ${USER\_NAME})

if [[ ${EXUSER} = ${USER\_NAME} ]]

then

echo "User ${USER\_NAME} already exists."

echo "Exit Code is ${?}."

else

#Generate a complex password

SPEC='!@#$%^&\*()'

SPECHAR=$(echo $SPEC | fold -w1 | shuf | head -1)

PASSWORD=Capita${RANDOM}${SPECHAR}

useradd -m ${USER\_NAME}

echo $PASSWORD |passwd --stdin ${USER\_NAME}

echo "${USER\_NAME} is successfully created.Password is ${PASSWORD}"

fi

read -p "Do you want to create users(Yes/No):" CHOICE

done

echo "You have opted for no...!!"

==========================================================

Otherway of WHILE:

END=5

i=1 ; while [[ $i -le $END ]] ; do

echo $i

((i = i + 1))

done

==========================================================

**IF Statement - Multiple Conditions:**

echo -e "1.Morning\n2.Afternoon\n3.Evening\n4.Night"

echo -n "Please select from above:"

read n

if [[ $n -eq 1 ]]

then

echo "Good Morning"

elif [[ $n -eq 2 ]]

then

echo "Good Afternoon"

elif [[ $n -eq 3 ]]

then

echo "Good Evening"

elif [[ $n -eq 4 ]]

then

echo "Good Night"

Fi

FUNCTIONS:

--------------------

#!/bin/bash

usercreate(){

for USER in $@

do

useradd -m $USER

done

}

passwordcreate(){

for USER in $@

do

PASSWORD=India@${RANDOM}

chown $USER:$USER /home/$USER

echo "$USER:$PASSWORD" | sudo chpasswd

done

}

users(){

usercreate $@

passwordcreate $@

echo "You are planning to create $# users."

}

users $@

#!/bin/bash

myuser(){

read -p "Please enter the UserName:" USER\_NAME

PASSWORD=$(curl -sL https://helloacm.com/api/random/?n=14)

EXUSER=$(cat /etc/passwd|cut -d ":" -f 1|grep -i ${USER\_NAME})

if [[ "${EXUSER}" == "${USER\_NAME}" ]]

then

echo "UserName Already Exists..!!"

# exit

else

useradd -m "${USER\_NAME}"

echo "${PASSWORD}" | passwd --stdin "${USER\_NAME}"

echo "Created User "${USER\_NAME}" & Password will be "${PASSWORD}""

fi

echo "User "${USER\_NAME}" created....!!"

}

myuser

FUNCTION OTHER WAY:

-------------------

#!/bin/bash

function myuser {

read -p "Please enter the UserName:" USER\_NAME

PASSWORD=$(curl -sL https://helloacm.com/api/random/?n=14)

EXUSER=$(cat /etc/passwd|cut -d ":" -f 1|grep -i ${USER\_NAME})

if [[ "${EXUSER}" == "${USER\_NAME}" ]]

then

echo "UserName Already Exists..!!"

# exit

else

useradd -m "${USER\_NAME}"

echo "${PASSWORD}" | passwd --stdin "${USER\_NAME}"

echo "Created User "${USER\_NAME}" & Password will be "${PASSWORD}""

fi

echo "User "${USER\_NAME}" created....!!"

}

myuser

---------------------------------------------------------

----------------------------------------------------------

#!/bin/bash

newuser () {

local USERNAME=${@}

useradd -m ${USERNAME}

echo "The ${USERNAME} is created..!!"o

}

newuser testuser1

newuser testuser2

newuser testuser3

newuser testuser4

newuser testuser5

newuser testuser6

Global & Loca Variables:

#!/bin/bash

NAME="GLOBAL-VARIABLE"

func(){

local NAME="LOCAL-VARIABLE"

echo ${NAME}

}

echo ${NAME}

func

RETURN:

------

#!/bin/bash

function testing () {

read -p "Enter the first number:" NUM1

read -p "Enter the second number:" NUM2

NEWNUM=$(($NUM1+$NUM2))

#echo "The New Number is $NEWNUM""

return ${NEWNUM}

}

testing

GETVAL=${?}

ostechnix

Ostechnix

o$technix

linux

linus

unix

technology

hello world

HELLO world

CUT:

----

cat /etc/passwd | cut -c1-4

cat /etc/passwd | cut -c4

cat /etc/passwd | cut -c4-

cat /etc/passwd | cut -d ":" -f1

cat /etc/passwd | cut -d ":" -f1,2

GREP:

-----

cat /etc/passwd | grep -i tes

cat /etc/passwd | grep -v test

cat /etc/passwd | grep -i 't$'

cat testfile.txt | egrep '^(o|h)'

cat testfile.txt | egrep -n '^(o|h)'

cat testfile.txt | egrep -i '^(o|h)'

egrep '^[l-u]' testfile.txt

egrep '[l-u]$' testfile.txt

egrep '[l-y]$' testfile.txt

egrep '^[l-u]|[L-U]' file.txt (or) egrep '^([l-u]|[L-U])' file.txt

netstat -a | egrep -i '(CONNECTED | ESTABLISHED)'

netstat -a | egrep -i '(CONNECTED | LISTEN | ESTABLISHED)'

cat /etc/passwd | awk -F ':' '{print $1,$2,$3}'

cat /etc/passwd | cut -d ":" -f1,2,3

cat /etc/passwd | awk -F ":" -v OFS="," '{print $1,$2,$3}'

cat /etc/passwd | awk -F ":" '{print $1 "," $2 "," $3}'

cat /etc/passwd | awk -F ":" '{print $1 ", " $2 ", " $3}' - Additional comma spaces

cat /etc/passwd | awk -F ":" '{print "USERNAME:" $1 ", " $2 ", " "UID:"$3}' --(Ammending words)

Changing Data Order:

cat /etc/passwd | awk -F ":" '{print $7 ":" $1}'

cat /etc/passwd | awk -F ":" '{print "HOMEDIR:"$6,",""UID:"$3,",""USERNAME:"$1}'

=============================LATEST-SCRIPTS---=======================

AWS S3 Bucket Creation Script:

#!/bin/bash

checkbucket() {

AWS\_REGION='us-east-1'

#read -p "Enter The Name Of The Bucket:" BUCKETNAME

BUCKETNAMES=${@}

for BUCKET in $BUCKETNAMES; do

EXISTING=$(aws s3 ls | cut -d " " -f 3 | grep -w $BUCKET)

if [ -z $EXISTING ]; then

echo "Bucket $BUCKET dont exists. Please create it."

aws s3api create-bucket --bucket $BUCKET --region $AWS\_REGION

else

echo "BUCKET $BUCKET EXISTS..!!"

fi

done

}

BREAK SCRIPT:

------------------------

#!/bin/bash

i=0

read -p "Enter the number where you want to break the loop (1 to 100):" NUM

while [ $i -lt 100 ]; do

if [ $(expr $i % 2) -eq 0 ]; then

echo "$i is a even number."

if [ $i -eq $NUM ]; then

break

fi

else

echo "$i is a odd number."

if [ $(expr $i % 2) -ne 0 ]; then

echo "$i is a even number."

if [ $i -eq $NUM ]; then

break

fi

fi

fi

i=$(($i + 1))

done

echo "The Loop terminated at $i"

BREAK COMMAND WITH S3 BUCKETS:

#!/bin/bash

checkbucket() {

AWS\_REGION='us-east-1'

#read -p "Enter The Name Of The Bucket:" BUCKETNAME

BUCKETNAMES=${@}

for BUCKET in $BUCKETNAMES; do

EXISTING=$(aws s3 ls | cut -d " " -f 3 | grep -w $BUCKET)

if [ -z $EXISTING ]; then

echo "Bucket $BUCKET dont exists. Please create it."

aws s3api create-bucket --bucket $BUCKET --region $AWS\_REGION

else

echo "BUCKET $BUCKET EXISTS AND WE WILL STOP CREATING REST OF THE BUCKETS..!!"

break

fi

done

}

checkbucket ${@}

CONTINUE COMMAND WITH S3 BUCKETS:

#!/bin/bash

checkbucket() {

AWS\_REGION='us-east-1'

#read -p "Enter The Name Of The Bucket:" BUCKETNAME

BUCKETNAMES=${@}

for BUCKET in $BUCKETNAMES; do

EXISTING=$(aws s3 ls | cut -d " " -f 3 | grep -w $BUCKET)

if [ -z $EXISTING ]; then

echo "Bucket $BUCKET dont exists. Please create it."

aws s3api create-bucket --bucket $BUCKET --region $AWS\_REGION

else

echo "BUCKET $BUCKET EXISTS AND WE WILL CONTINUE TO CREATING REST OF THE BUCKETS..!!"

continue

fi

done

}

**RETURN IN SHELL SCRIPTING:**

#!/bin/bash

func1 () {

a=$( expr 4 + 10 )

#echo "The Value of A is $a."

return $a

}

func2 () {

func1

b=$( expr 20 + $a )

#echo "The Value of A+B is $b."

return $b

}

func3 () {

func2

c=$( expr 100 + $b )

echo " The return value of $c"

return $c

}

Func3

**REMOVING THE S3 BUCKETS AWS CLI & REGEX:**

buckets=$(aws s3 ls | cut -d " " -f 3 | grep -E ^sreeenv)

for i in $buckets

do

aws s3api delete-bucket --bucket $i --region us-east-1

done