#!/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

for ver in $(cat version)

do

#PACKAGE=$(echo "https://releases.hashicorp.com/terraform/0.12.28/terraform\_0.12.28\_linux\_amd64.zip" | sed 's/0.12.28/'"$ver"'/g')

#wget https://releases.hashicorp.com/terraform/${ver}/terraform\_${ver}\_linux\_amd64.zip

#wget $PACKAGE

ls -al | grep -i terraform

sleep 5

done

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

Creating volumes

#!/bin/bash

echo $\*

for VOL in $\*

do

echo "Creating The Volume...."

aws ec2 create-volume --volume-type gp2 --size $VOL --availability-zone us-east-1a --tag-specifications 'ResourceType=volume,Tags=[{Key=Name,Value=PRODVOL}]'

done

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

#!/bin/bash

for x in $\*

do

echo $x

done

echo "==================="

for x in $@

do

echo $x

done

echo "==================="

echo '"$\*"'

for x in "$\*"

do

echo $x

done

echo "==================="

for x in "$@"

do

echo $x

done

echo "==================="

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

#!/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 v kishor kumar

> do

> echo "Hello ${X}"

> done

Hello v

Hello kishor

Hello kumar

${@} - for multiple arguments

${#} - gives argument count passed

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

#!/bin/bash

re='^[0-9]+$'

if [ $# -gt 0 ]; then

for PERSON in $@; do

if [[ -n $PERSON ]] && [[ $PERSON =~ $re ]]; then

#echo "Lets Create User $PERSON"

echo "Username $PERSON is Empty or Invalid User Name."

else

#echo "Username is Empty or Invalid User Name."

echo "Lets Create User $PERSON"

fi

done

else

echo "No Args Given"

fi

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

#!/bin/bash

#re='^[0-9]+$'

re='^[0-9]'

if [ $# -gt 0 ]; then

for PERSON in $@; do

if [[ -n $PERSON ]] && [[ $PERSON =~ $re ]]; then

echo "Username $PERSON is Empty or Invalid User Name."

else

echo "Lets Create User $PERSON"

fi

done

else

echo "No Args Given"

fi

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

#!/bin/bash

read -p "Please Enter The Username:" PERSON

renum='^[0-9]'

recah='[^a-z]'

if [[ -z $PERSON ]] || [[ $PERSON =~ $recah ]]

then

echo "NO INPUT GIVEN or USERNAME STARTED WITH A NUMBER"

else

echo $PERSON

fi

#!/bin/bash

while true

do

read -p "Please Enter Your Name:" Name

read -p "Please Enter DOB Year:" DOY

AGE=$(expr 2020 - $DOY)

if [ $AGE -le 15 ]; then

echo "You are 15 or below, get back to home....!!"

elif [[ $AGE -gt 15 ]] && [[ $AGE -lt 60 ]]; then

echo "Your Are Allowed To Drink....!!"

else

echo "You have cross 60 , stay home and avoid CORONA..!!"

fi

done

================ARRAYS-WITH-AWSCLI-JQ=======================

PROFILES=(‘us-east-1’ ‘us-east-2’ ‘us-west-1’ ‘us-west-2’)

echo ${PROFILES[@]}

east1 east2 west1 west2

#####--LIST-INSTANCES--##########

for PROFILE in ${PROFILES[@]}; do

echo $PROFILE

aws ec2 describe-instances --profile $PROFILE | jq -r '.Reservations[].Instances[].InstanceId'

echo "==========================="

sleep 1

done

#####--LIST-KEYPAIRS--##########

for PROFILE in ${PROFILES[@]}; do

echo $PROFILE

aws ec2 describe-key-pairs --profile $PROFILE | jq -r '.KeyPairs[].KeyName'

echo "==========================="

sleep 1

done

#!/bin/bash

#####--LIST-REGIONS-USING-JQ--##########

REGIONS=$(aws ec2 describe-regions --profile east1 | jq -r '.Regions[].RegionName')

#!/bin/bash

#####--LIST-REGIONS-USING-ONLY-CLI--##########

aws ec2 describe-regions --profile east1 | grep -i RegionName | cut -d ":" -f 2 | tr -d '"' | tr -d ','

###############--WHILE-LOOP#################

X=0

while [ $X -lt 4 ]; do

echo ${PROFILES[$X]}

aws ec2 describe-key-pairs --profile ${PROFILES[$X]} | jq -r '.KeyPairs[].KeyName'

echo "====================="

X=$(($X + 1))

sleep 1

done

**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

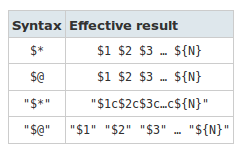
echo "Good Evening"

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=${?}

#!/bin/bash

function add {

add=$(($1 + $2))

return ${add}

}

add 5 5

A=$?

add 6 6

B=$?

add 7 7

C=$?

D=$(expr $A + $B + $C)

echo $D

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}'

cat data | awk -F " " '{print $1,$3,$2}' | sort | uniq

TR:

aws ec2 describe-vpcs | jq ".Vpcs[].VpcId" | tr -d '"' | tr [:lower:] [:upper:]

aws ec2 describe-vpcs | jq ".Vpcs[].VpcId" | tr -d '"' | tr [:lower:] [:upper:] | tr '-' '='

aws ec2 describe-vpcs | jq ".Vpcs[].VpcId" | tr -d '"' | tr [:lower:] [:upper:] | tr '-' '\_\_'

aws ec2 describe-vpcs | jq ".Vpcs[].VpcId" | tr -d '"' | tr [:lower:] [:upper:] | tr '-' " "

SED:

sed -i "s/.\*PasswordAuthentication.\*/PasswordAuthentication yes/g" /etc/ssh/sshd\_config

sed -i '58s/PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd\_config

useradd -m testuser10

echo "testuser10:India@123456" | sudo chpasswd

passwd -r testuser10

service sshd restart

sed '1,5 s/HOLA/HELLO/' testfile -> replace line 1 to 5

sed 's/HOLA/HELLO/g' testfile -> replace all HOLA to HELLO

sed -i '15 a <h1>WELCOME TO DEVOPS</h1>' /var/www/html/index.nginx-debian.html

To characterize the three tools crudely:

* tr works on characters (changes or deletes them).
* sed works on lines (modifies words or other parts of lines, or inserts or deletes lines).
* awk work on records with fields (by default whitespace separated fields on a line, but this may be changed by setting FS and RS).

Using jq with shell:

https://programminghistorian.org/en/lessons/json-and-jq

curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq '.prefixes[] | .ip\_prefix,.service'

curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq '.prefixes[].ip\_prefix'|tr -d '"' >> awsips

curl https://ip-ranges.amazonaws.com/ip-ranges.json | jq '.prefixes[].ip\_prefix,.prefixes[].service'

for subnet in $(cat subnets);

do

aws ec2 create-subnet --vpc-id vpc-0f47002cff9d8dd21 --cidr-block $subnet --availability-zone us-east-1a;

Done

Envsubst:

export AWS\_ACCESS\_KEY\_ID=AKIAIOSFODNN7EXAMPLE

export AWS\_SECRET\_ACCESS\_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

export AWS\_DEFAULT\_REGION=us-west-2

root@ip-10-25-1-56:~# cat creds #File with variables

[default]

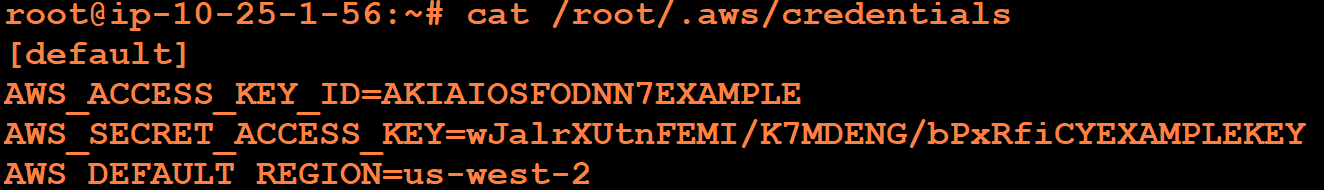
AWS\_ACCESS\_KEY\_ID=$AWS\_ACCESS\_KEY\_ID

AWS\_SECRET\_ACCESS\_KEY=$AWS\_SECRET\_ACCESS\_KEY

AWS\_DEFAULT\_REGION=$AWS\_DEFAULT\_REGION

envsubst < "creds" > "/root/.aws/credentials"

The file will be created with replaced variables as shown below.



DIRENV:

Folder Specific env variables.

Download latest version of direnv from <https://github.com/direnv/direnv>

Copy it to usr/local/bin and provide 777 access.

Create 3 Folders, DEV, UAT, PROD

Add .envrc file in all 3 folders and perform following in each folder.

---DEV-ENV---

cd DEV

export AWS\_ACCESS\_KEY\_ID=AKIAIOSFODNN7DEV >> .envrc

export AWS\_SECRET\_ACCESS\_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYDEV >> .envrc

export AWS\_DEFAULT\_REGION=us-east-1 >> .envrc

---UAT-ENV---

cd UAT

export AWS\_ACCESS\_KEY\_ID=AKIAIOSFODNN7UAT >> .envrc

export AWS\_SECRET\_ACCESS\_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYUAT >> .envrc

export AWS\_DEFAULT\_REGION=us-west-1 >> .envrc

---PROD-ENV---

cd PROD

export AWS\_ACCESS\_KEY\_ID=AKIAIOSFODNN7PROD >> .envrc

export AWS\_SECRET\_ACCESS\_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYPROD >> .envrc

export AWS\_DEFAULT\_REGION=us-apsouth-1 >> .envrc

cd DEV

direnv allow .

eval "$(direnv hook bash)"

echo $AWS\_ACCESS\_KEY\_ID

echo $AWS\_DEFAULT\_REGION

cd UAT

direnv allow .

eval "$(direnv hook bash)"

echo $AWS\_ACCESS\_KEY\_ID

echo $AWS\_DEFAULT\_REGION

cd PROD

direnv allow .

eval "$(direnv hook bash)"

echo $AWS\_ACCESS\_KEY\_ID

echo $AWS\_DEFAULT\_REGION

NESTED IF & LOOP:

#!/bin/bash

if [ $# -gt 0 ]

then

echo $#

for USERNAME in $@

do

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

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

if [ -z ${USERNAME} ]

then

echo "Invalid Input or No Input Given..!!"

else

EXUSER1=$(cat /etc/passwd | grep -w ${USERNAME} | cut -d ":" -f1)

if [ "${USERNAME}" = "${EXUSER1}" ]

then

echo "${USERNAME} Exists. Try a new one"

else

echo "Lets Create User ${USERNAME}"

useradd -m ${USERNAME}

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

chown ${USERNAME}:${USERNAME} /home/${USERNAME}

echo "${USERNAME}:${PASSWORD}" | sudo chpasswd

echo "The Username is ${USERNAME} and temp passsword is ${PASSWORD}"

passwd -e ${USERNAME}

fi

fi

done

else

echo "Invalid Input or No Input Given..!!"

fi

TERRAFORM DOWNLOAD SHELL SCRIPT FOR FILE OPERATOR(B17):

#!/bin/bash

echo "Availible Terraform Versions..."

echo "14.0 to 14.11"

echo "15.0 to 15.3"

read -p "Enter The Terraform MAJOR Version:" MAJOR

read -p "Enter The Terraform MINOR Version:" MINOR

folder1=${MAJOR}-${MINOR}

if [ $MAJOR -lt 14 ]; then

echo "Please Select Either of 14 or 15 version"

else

VERSION=${MAJOR}.${MINOR}

echo "Downloading Terarform Version 0.${VERSION}"

if [ -d /tmp/$folder1 ]; then

echo "FODLER EXISTS"

cd /tmp/$folder1

if [ -f terraform\_${VERSION}.zip ]; then

echo "Version Already Exists...."

else

wget https://releases.hashicorp.com/terraform/0.${VERSION}/terraform\_0.${VERSION}\_linux\_amd64.zip

mv -f terraform\_0.${VERSION}\_linux\_amd64.zip terraform\_${VERSION}.zip

fi

unzip -o terraform\_${VERSION}.zip

chmod 777 terraform

./terraform version

else

echo "Lets Create The Folder"

mkdir /tmp/$folder1 && cd /tmp/$folder1

wget https://releases.hashicorp.com/terraform/0.${VERSION}/terraform\_0.${VERSION}\_linux\_amd64.zip

mv terraform\_0.${VERSION}\_linux\_amd64.zip terraform\_${VERSION}.zip

unzip terraform\_${VERSION}.zip

chmod 777 terraform

./terraform version

fi

fi

IF-ELIG-ELSE(B17):

#!/bin/bash

echo "Vaccine Is Availible for 45+ & below 90 Years Only....!!!!"

CURRENT\_YEAR=2021

read -p "Please Enter Your YOB:" YOB

CURRENTAGE=$(expr $CURRENT\_YEAR - $YOB)

echo "Your Current Age is: $CURRENTAGE"

if [ $CURRENTAGE -le 18 ]; then

echo "VACCINE NOT APPOVED FOR YOU..!!"

elif [[ $CURRENTAGE -ge 19 && $CURRENTAGE -le 45 ]]; then

echo "Vaccine Procurement Is In Progress For You. We Will Let You Know..."

elif [[ $CURRENTAGE -ge 90 && $CURRENTAGE -le 100 ]]; then

echo "Vaccine Is Not A Good Solution For You. Please Takecare of Your Self"

elif [[ $CURRENTAGE -ge 45 && $CURRENTAGE -le 90 ]]; then

echo "Please Visit Near By PHC For The Jab"

else

echo "INVALID AGE"

fi

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

#!/bin/bash

echo "Vaccine Is Availible for 45+ & below 90 Years Only....!!!!"

CURRENT\_YEAR=2021

read -p "Please Enter Your YOB:" YOB

CURRENTAGE=$(expr $CURRENT\_YEAR - $YOB)

echo "Your Current Age is: $CURRENTAGE"

if [ $CURRENTAGE -le 18 ]

then

echo "VACCINE NOT APPOVED FOR YOU..!!"

elif [ $CURRENTAGE -ge 19 -a $CURRENTAGE -le 45 ]

then

echo "Vaccine Procurement Is In Progress For You. We Will Let You Know..."

elif [ $CURRENTAGE -ge 90 -a $CURRENTAGE -le 100 ]

then

echo "Vaccine Is Not A Good Solution For You. Please Takecare of Your Self"

elif [ $CURRENTAGE -ge 45 -a $CURRENTAGE -le 90 ]

then

echo "Please Visit Near By PHC For The Jab"

else

echo "INVALID AGE"

fi