##############################BREAK-FUNCTION###########################

#!/bin/bash

aws\_regions=(us-east-1 us-east-2 hyd-india-1 eu-north-1 ap-south-1 eu-west-3 eu-west-2 eu-west-1 ap-northeast-2)

Get\_VPC() {

echo "Running The Functon To List VPCs using regions list"

for region in ${aws\_regions[@]}; do

echo "Getting VPCs in $region.."

vpc\_list=$(aws ec2 describe-vpcs --region $region | jq .Vpcs[].VpcId | tr -d '"')

vpc\_arr=(${vpc\_list[@]})

if [ ${#vpc\_arr[@]} -gt 0 ]

then

#echo ${vpc\_list[@]}

for vpc in ${vpc\_list[@]}

do

echo "The VPC-ID is: $vpc"

done

echo "#######################"

else

echo 'Invalid Region..!!'

echo "##########################"

echo "# Breaking at $region #"

echo "#########################"

break

fi

done

}

Get\_VPC

#!/bin/bash

Get\_VPC() {

echo "Running The Functon To List VPCs using regions list"

for region in $@; do

echo "Getting VPCs in $region.."

vpc\_list=$(aws ec2 describe-vpcs --region $region | jq .Vpcs[].VpcId | tr -d '"')

vpc\_arr=(${vpc\_list[@]})

if [ ${#vpc\_arr[@]} -gt 0 ]; then

#echo ${vpc\_list[@]}

for vpc in ${vpc\_list[@]}; do

echo "The VPC-ID is: $vpc"

done

echo "#######################"

else

echo 'Invalid Region..!!'

echo "##############################"

echo "# Breaking at $region ########"

echo "##############################"

break

fi

done

}

Get\_VPC $@

=========================CONTINUE======================================

#!/bin/bash

aws\_regions=(us-east-1 us-east-2 hyd-india-1 eu-north-1 ap-south-1 eu-west-3 eu-west-2 eu-west-1 ap-northeast-2 ap-northeast-1 sa-east-1 ca-central-1 ap-southeast-1 ap-southeast-2 eu-central-1 us-west-1 us-west-2)

Get\_VPC() {

echo "Running The Functon To List VPCs using regions list"

for region in ${aws\_regions[@]}; do

echo "Getting VPCs in $region.."

vpc\_list=$(aws ec2 describe-vpcs --region $region | jq .Vpcs[].VpcId | tr -d '"')

vpc\_arr=(${vpc\_list[@]})

if [ ${#vpc\_arr[@]} -gt 0 ]; then

#echo ${vpc\_list[@]}

for vpc in ${vpc\_list[@]}; do

echo "The VPC-ID is: $vpc"

done

echo "#######################"

else

echo 'Invalid Region..!!'

echo "##########################"

echo "# Breaking at $region #"

echo "#########################"

continue

fi

done

}

Get\_VPC

==========RETURN & NESTED FUNCTION======

#===========SHELL-ECHO-RETURN==================

#!/bin/bash

A=$1

B=$2

Addition() {

C=$(expr $A + $B)

echo $C

return $? #This Gives the Value of C. NOT MUCH USEFUL AS VALUE WILL BE RETURNED BY ECHO

}

#Call Function

Addition

#Running The Script:

x=$(bash return.sh 10 20)

y=$(bash return.sh 11 22)

echo $(expr $x + $y)

#===========NESTED FUNCTIONS==================

#!/bin/bash

A=$1

B=$2

Addition() {

AValue

BValue

C=$(expr $A + $B)

echo $C

#return $C

}

AValue() {

echo "Value of a is A is $A"

}

BValue() {

echo "Value of a is B is $B"

}

Addition

#NESTED-FUNCTIONS-AWS

#!/bin/bash

region=$1

Get\_Details() {

echo "Getting Details for VPC and EC2 Instances....."

Get\_VPC

Get\_EC2

}

Get\_VPC() {

echo "Getting VPCs in $region.."

vpc\_list=$(aws ec2 describe-vpcs --region $region | jq .Vpcs[].VpcId | tr -d '"')

vpc\_arr=(${vpc\_list})

for vpc in ${vpc\_arr[@]}

do

echo $vpc

done

}

Get\_EC2() {

echo "Getting Instances in $region.."

ec2\_list=$(aws ec2 describe-instances --region $region | jq ".Reservations[].Instances[].InstanceId" | tr -d '"')

ec2\_arr=(${ec2\_list})

for ec2 in ${ec2\_list}

do

echo $ec2

done

}

Get\_Details