**FUNCTONS:**

Syntax to use:

**Syntax1**:

function function\_name

{

###set of commands

}

**Syntax2:**

function\_name()

{

####set of commands

}

#!/bin/bash

Get\_VPC() {

echo "Running The Functon To List VPCs in $1"

vpc\_list=$(aws ec2 describe-vpcs --region $1 | jq .Vpcs[].VpcId | tr -d '"')

for vpc in $(echo $vpc\_list); do

echo "The VPC ID IS:$vpc"

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

done

}

#Get\_VPC $1

#Get\_VPC $2

for X in $@; do

Get\_VPC $X

done

#bash shell\_function 'us-east-1'

#bash func1.sh us-east-1 us-east-2 us-west-1

#!/bin/bash

Get\_VPC() {

regions=('us-east-1' 'us-east-2' 'us-west-1' 'us-west-2')

for region in ${regions[@]}

do

if [ $region = 'us-west-1' ]

then

#continue

#break

else

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

for vpc in $(echo $vpc\_list); do

echo "The VPC ID IS:$vpc in $region"

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

done

fi

done

}

Get\_VPC

#!/bin/bash

delete\_vols() {

vols=$(aws ec2 describe-volumes | jq ".Volumes[].VolumeId" | tr -d '"')

for vol in $vols

do

size=$(aws ec2 describe-volumes --volume-ids $vol | jq ".Volumes[].Size")

if [ $size -gt 5 ]

then

echo "$vol is a Prod Volume. Dont Delete"

else

echo "Deleting Volume $vol"

aws ec2 delete-volume --volume-id $vol

fi

done

}

delete\_vols

\* \* \* \* \* sudo bash /root/unttached\_ebs.sh us-east-1

#Delete Volumes which are not attached

#!/bin/bash

delete\_unttached\_vols() {

echo "Lets Check Volume Status For Region $REGION"

vols=$(aws ec2 describe-volumes --region $REGION | jq ".Volumes[].VolumeId" | tr -d '"')

for vol in $vols; do

STATUS=$(aws ec2 describe-volumes --volume-ids $vol | jq ".Volumes[].Attachments[].State" -r)

if [[ "${STATUS}" = 'attached' ]]; then

echo "$vol Is In Use By EC2 Instance. Dont Delete"

else

echo "Deleting Volume $vol As Its Not In Use."

aws ec2 delete-volume --volume-id $vol --region $REGION

fi

done

}

for REGION in $(aws ec2 describe-regions | jq ".Regions[].RegionName" -r); do

delete\_unttached\_vols $REGION

done

#0 23 \* \* \* sudo bash /root/unttached\_ebs.sh us-east-1

#!/bin/bash

#Deleting Unattached Volumes

#!/bin/bash

delete\_vols() {

vols=$(aws ec2 describe-volumes --region $REGION | jq ".Volumes[].VolumeId" | tr -d '"')

for vol in $vols; do

STATUS=$(aws ec2 describe-volumes --volume-ids $vol --region $REGION | jq ".Volumes[].Attachments[].State" -r)

if [ "${STATUS}" = 'attached' ]; then

echo "$vol Is Attached To EC2 Instance."

else

echo "$vol Is Not Attached To EC2 Instance and deleting Volume $vol in region $REGION"

aws ec2 delete-volume --volume-id $vol --region $REGION

fi

done

}

for REGION in $@; do

delete\_vols $REGION

done

delete\_unattached\_vols

#Delete unattached EBS Volumes based on the region arguments.

#Syntax: bash delete\_volumes.sh us-east-1 us-east-2 us-west-1 ap-south-1

#!/bin/bash

delete\_vols() {

vols=$(aws ec2 describe-volumes --region $REGION | jq ".Volumes[].VolumeId" | tr -d '"')

for vol in $vols; do

ATTACHED=$(aws ec2 describe-volumes --volume-ids $vol | jq ".Volumes[].Attachments[]")

if [ $ATTACHED -gt 0 ]; then

echo "$vol Is Attached To EC2 Instance."

else

echo "$vol Is Not Attached To EC2 Instance and deleting Volume $vol"

aws ec2 delete-volume --volume-id $vol

fi

done

}

for REGION in $@; do

delete\_vols $REGION

done

delete\_vols

\* \* \* \* \* sudo bash /root/delete\_unattached\_volumes.sh us-east-1 us-east-2 us-west-1

#!/bin/bash

function subnets() {

echo "Getting SUBNETS Info VPC $VPC..."

aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC" --region $REGION | jq ".Subnets[].SubnetId"

echo "---------------------------------------------"

}

function vpcs() {

for REGION in $@; do

echo "Getting VPC List For Regions $REGION..."

vpcs=$(aws ec2 describe-vpcs --region "${REGION}" | jq ".Vpcs[].VpcId" | tr -d '"')

echo $vpcs

for VPC in $vpcs; do

subnets $VPC

done

done

}

vpcs $@

#!/bin/bash

function subnets {

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*Getting SUBNETS Info VPC $VPC in region $REGION\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

aws ec2 describe-subnets --filters "Name=vpc-id,Values=$VPC" --region $REGION | jq ".Subnets[].SubnetId"

echo "---------------------------------------------"

}

function sg {

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\*\*Getting Security Group Info VPC $VPC in region $REGION\*\*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

aws ec2 describe-security-groups --filters "Name=vpc-id,Values=$VPC" --region $REGION | jq ".SecurityGroups[].GroupName"

echo "---------------------------------------------"

}

vpcs() {

for REGION in $@; do

echo "Getting VPC List For Regions $REGION..."

vpcs=$(aws ec2 describe-vpcs --region "${REGION}" | jq ".Vpcs[].VpcId" | tr -d '"')

echo $vpcs

echo "--------------------------------------------------"

for VPC in $vpcs; do

subnets $VPC

done

for VPC in $vpcs; do

sg $VPC

done

done

}

vpcs $@

Return Values in Shell Script:

#!/bin/bash

myaddition() {

X=$(expr $1 + $2)

echo $X

}

myaddition $1 $2

A=$(bash addition.sh 11 21)

B=$(bash addition.sh 6 7)

echo $(expr $A + $B)

#Function for deleting unattached EBS Volumes

#!/bin/bash

VALID\_REGIONS=($(aws ec2 describe-regions | jq ".Regions[].RegionName" -r))

unttached\_ebs() {

for VOL in $(aws ec2 describe-volumes --region $1 | jq ".Volumes[].VolumeId" | tr -d '"'); do

echo "Volume ID is $VOL"

STATE=$(aws ec2 describe-volumes --volume-ids $VOL --region $1 | jq ".Volumes[].State" | tr -d '"')

# if [ "${STATE}" = 'in-use' ]; then

# echo "$VOL is attached and cannot be deleted."

# echo '-----------------------------------------'

# else

# echo "Lets Delete VOlume $VOL"

# aws ec2 delete-volume --volume-id $VOL

# echo '-----------------------------------------'

# fi

if [ "${STATE}" != 'in-use' ]; then

echo "$VOL is not attached and can be deleted."

aws ec2 delete-volume --volume-id $VOL

echo '-----------------------------------------'

fi

done

}

if [ $# -gt 0 ]; then

for REGION in $@; do

VALIDATION=$(echo ${VALID\_REGIONS[@]} | grep -i $REGION)

if [ $? -eq 0 ]; then

unttached\_ebs $REGION

else

echo "$REGION IS INVALID."

fi

done

else

echo "No region name provided. Please proviade valid region name"

fi