#!/bin/bash

#Tested on July 21st 2022

ALL\_REGIONS=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

#REGIONS=('us-east-1' 'us-east-2' 'ind-hyd-1' 'ap-south-1' 'eu-west-1')

REGIONS=($ALL\_REGIONS)

for REGION in ${REGIONS[@]}; do

echo "The Region Name is $REGION"

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

VPC\_ID=$(aws ec2 describe-vpcs --region $REGION 2>/dev/null | jq ".Vpcs[].VpcId" | tr -d '"')

VPC\_ARRAY=($VPC\_ID)

if [ ${#VPC\_ARRAY[@]} -gt 0 ]; then

echo "The Region Name is $REGION is valid. Retriveing VPC Information...."

for VPC in $VPC\_ID; do

echo $VPC

done

else

echo "INVALID REGION $REGION"

#break

continue

fi

done

#!/bin/bash

#Tested on July 21st 2022

ALL\_REGIONS=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

#REGIONS=('us-east-1' 'us-east-2' 'ind-hyd-1' 'ap-south-1' 'eu-west-1')

REGIONS=($ALL\_REGIONS)

for REGION in ${REGIONS[@]}; do

echo "The Region Name is $REGION"

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

VPC\_ID=$(aws ec2 describe-vpcs --region $REGION >/dev/null 2>&1 | jq ".Vpcs[].VpcId" | tr -d '"')

VPC\_ARRAY=($VPC\_ID)

if [ ${#VPC\_ARRAY[@]} -gt 0 ]; then

echo "The Region Name is $REGION is valid. Retriveing VPC Information...."

for VPC in $VPC\_ID; do

echo $VPC

done

else

echo "INVALID REGION $REGION"

#break

continue

fi

done

#!/bin/bash

AWSREGION=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

if [ $# -gt 0 ]; then

echo "You have given $# regions to check the VPC List"

for REGION in $@; do

if [[ $AWSREGION =~ (^|[[:space:]])"$REGION"($|[[:space:]]) ]]; then

echo "Getting VPC List For $REGION"

for VPC in $(aws ec2 describe-vpcs --region $REGION | jq ".Vpcs[].VpcId" | tr -d '"'); do

echo "The VPC ID is $VPC"

echo "Getting The Subnets From VPC $VPC"

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

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

sleep 1

done

else

echo "$REGION IS AN INVALID REGION"

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

fi

done

else

echo "Invalid Input. You have given $# params. Give atlast one region name."

fi

#!/bin/bash

REGIONS=$@

for REGION in $REGIONS

do

echo "Listing VPC in REGION $REGION:"

aws ec2 describe-vpcs --region $REGION | jq .Vpcs[].VpcId

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

echo "Listing VPC CIDR Blocks in REGION $REGION:"

aws ec2 describe-vpcs --region $REGION | jq .Vpcs[].CidrBlockAssociationSet[].CidrBlock

done

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

#!/bin/bash

REGIONS=$@

for REGION in $REGIONS

do

#echo "Listing VPC in REGION $REGION:"

VPCS=$(aws ec2 describe-vpcs --region $REGION | jq .Vpcs[].VpcId)

VPCSARRAY=($VPCS)

LENGTH=${#VPCSARRAY[@]}

#echo ${VPCSARRAY[@]}

#echo "----------------------------------------------------"

#echo "Listing VPC CIDR Blocks in REGION $REGION:"

CIDRS=$(aws ec2 describe-vpcs --region $REGION | jq .Vpcs[].CidrBlockAssociationSet[].CidrBlock)

CIDRSARRAY=($CIDRS)

#echo ${CIDRSARRAY[@]}

done

for ((i=0;i<$LENGTH;i++))

do

echo "${VPCSARRAY[i]} --> ${CIDRSARRAY[i]}"

done

SCRIPT TO CHECK VALID CIDR RANGES IN ALL REGIONS

#!/bin/bash

#REGIONS=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

REGIONS=us-east-1

CIDR1='10.1.0.0/16'

CIDR2='172.16.0.0/16'

CIDR3='172.31.0.0/16'

for REGION in $REGIONS; do

echo "Getting VPC Cidr Ranges In Region $REGION"

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

for VPC in $VPCID

do

CIDR=$(aws ec2 describe-vpcs --vpc-ids $VPC --region $REGION | jq ".Vpcs[].CidrBlock" | tr -d '"')

if [ ${CIDR} = ${CIDR1} -o ${CIDR} = ${CIDR2} -o ${CIDR} = ${CIDR3} ]; then

echo $CIDR

echo "$VPC in region $REGION is in Valid Range"

else

echo $CIDR

echo "InValid Range"

fi

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

done

done

#--------------------------------------------------------------------------------------

#!/bin/bash

REGIONS=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

#REGIONS=us-east-1

CIDR1='10.1.0.0/16'

CIDR2='172.16.0.0/16'

CIDR3='172.31.0.0/16'

for REGION in $REGIONS; do

echo "Getting VPC Cidr Ranges In Region $REGION"

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

for VPC in $VPCID

do

CIDR=$(aws ec2 describe-vpcs --vpc-ids $VPC --region $REGION | jq ".Vpcs[].CidrBlock" | tr -d '"')

if [[ ${CIDR} = ${CIDR1} || ${CIDR} = ${CIDR2} || ${CIDR} = ${CIDR3} ]]; then

echo $CIDR

echo "$VPC in region $REGION is in Valid Range"

else

echo $CIDR

echo "InValid Range"

fi

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

done

done

#Get All The EC2 Instances ID and Instance Type From All Regions

#!/bin/bash

AWSREGION=$(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')

if [ $# -gt 0 ]; then

echo "You have given $# regions to check the VPC List"

for REGION in $@; do

if [[ $AWSREGION =~ (^|[[:space:]])"$REGION"($|[[:space:]]) ]]; then

echo "Getting VPC List For $REGION"

for INST in $(aws ec2 describe-instances --region $REGION | jq ".Reservations[].Instances[].InstanceId" | tr -d '"'); do

echo "${INST}'--->'$(aws ec2 describe-instances --region $REGION --instance-ids ${INST} | grep -i instancetype | cut -d ":" -f2 | tr -d "," | tr -d '"')"

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

sleep 1

done

else

echo "$REGION IS AN INVALID REGION"

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

fi

done

else

echo "Invalid Input. You have given $# params. Give atlast one region name."

fi

#Command to execute

bash ec2.sh $(aws ec2 describe-regions | jq ".Regions[].RegionName" | tr -d '"')