#!/bin/bash

#This Script Checks The User Input Is A Valid AWS Region and Prints VPCs

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

USERINPUT=$1

if [[ ${REGIONS[@]} =~ ${USERINPUT} ]]; then

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

else

echo "INVALID REGION"

fi

#!/bin/bash

#Script to find EC2 Running in all AWS Regions.

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

echo "Getting EC2 Instacnce Information For Region $REGION"

VM=$(aws ec2 describe-instances --region $REGION | jq ".Reservations[].Instances[].InstanceId" | tr -d '"' | wc -l)

if [ $VM -gt 0 ]; then

echo "The region $REGION has $VM EC2 Instances. The instance IDs are:"

aws ec2 describe-instances --region $REGION | jq ".Reservations[].Instances[].InstanceId" | tr -d '"'

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

else

echo "The region $REGION has $VM EC2 Instances."

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

fi

done

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

#!/bin/bash

#Script to find Availability Zones in all AWS Regions.

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

echo "Getting AZ Zone Information For Region $REGION"

ZONES=$(aws ec2 describe-availability-zones --region $REGION | jq ".AvailabilityZones[].ZoneName" | tr -d '"' | wc -l)

if [ $ZONES -lt 3 ]; then

echo "The region $REGION has $ZONES Availibility Zone. The Zone Names are:"

aws ec2 describe-availability-zones --region $REGION | jq ".AvailabilityZones[].ZoneName" | tr -d '"'

echo "This Region $REGION Falls Under Very Small Infrastructure"

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

elif [ $ZONES -lt 4 -a $ZONES -gt 2 ]; then

echo "The region $REGION has $ZONES Availibility Zone. The Zone Names are:"

aws ec2 describe-availability-zones --region $REGION | jq ".AvailabilityZones[].ZoneName" | tr -d '"'

echo "This Region $REGION Falls Under Small Infrastructure"

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

elif [ $ZONES -lt 5 -a $ZONES -gt 3 ]; then

echo "The region $REGION has $ZONES Availibility Zone. The Zone Names are:"

aws ec2 describe-availability-zones --region $REGION | jq ".AvailabilityZones[].ZoneName" | tr -d '"'

echo "This Region $REGION Falls Under Medium Infrastructure"

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

else

echo "The region $REGION has $ZONES Availibility Zone. The Zone Names are:"

aws ec2 describe-availability-zones --region $REGION | jq ".AvailabilityZones[].ZoneName" | tr -d '"'

echo "This Region $REGION Falls Under Big Infrastructure"

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

fi

done

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

#!/bin/bash

read -p "Enter The Choice(vpcs/instances/tags):" CHOICE

if [ ${CHOICE} = 'vpcs' ]; then

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

elif [ ${CHOICE} = 'instances' ]; then

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId

elif [ ${CHOICE} = 'tags' ]; then

aws ec2 describe-tags | jq .Tags[].Value

else

echo "Invalid Input, Please Select between vpcs/instances/tags."

Fi

---------------------AWS-WITH-FUNCTION------------------------------------

#!/bin/bash

get\_details() {

CHOICE=$@

if [ "${CHOICE}" = 'vpcs' ]; then

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

elif [ "${CHOICE}" = 'instances' ]; then

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId

elif [ "${CHOICE}" = 'tags' ]; then

aws ec2 describe-tags | jq .Tags[].Value

else

echo "Invalid Input, Please Select between vpcs/instances/tags."

fi

}

get\_details $@

#!/bin/bash

while true; do

read -p "Enter The Choice(vpcs/instances/tags):" CHOICE

case $CHOICE in

vpcs)

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

;;

instances)

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId

;;

tags)

aws ec2 describe-tags | jq .Tags[].Value

;;

\*)

echo "Invalid Input, Please Select between vpcs/instances/tags."

;;

esac

done

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

#!/bin/bash

case $@ in

vpcs)

aws ec2 describe-vpcs | jq .Vpcs[].VpcId ;;

instances)

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId ;;

tags)

aws ec2 describe-tags | jq .Tags[].Value;;

\*)

echo "Invalid Input, Please Select between vpcs/instances/tags." ;;

esac

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

#!/bin/bash

for ITEM in $@

do

case $ITEM in

vpcs)

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

echo "-------------------------";;

instances)

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId

echo "-------------------------";;

tags)

aws ec2 describe-tags | jq .Tags[].Value

echo "-------------------------";;

\*)

echo "Invalid Input, Please Select between vpcs/instances/tags." ;;

esac

done

bash case3.sh vpcs tags instances

#!/bin/bash

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

USERINPUT=$1

if [[ ${REGIONS[@]} =~ ${USERINPUT} ]]; then

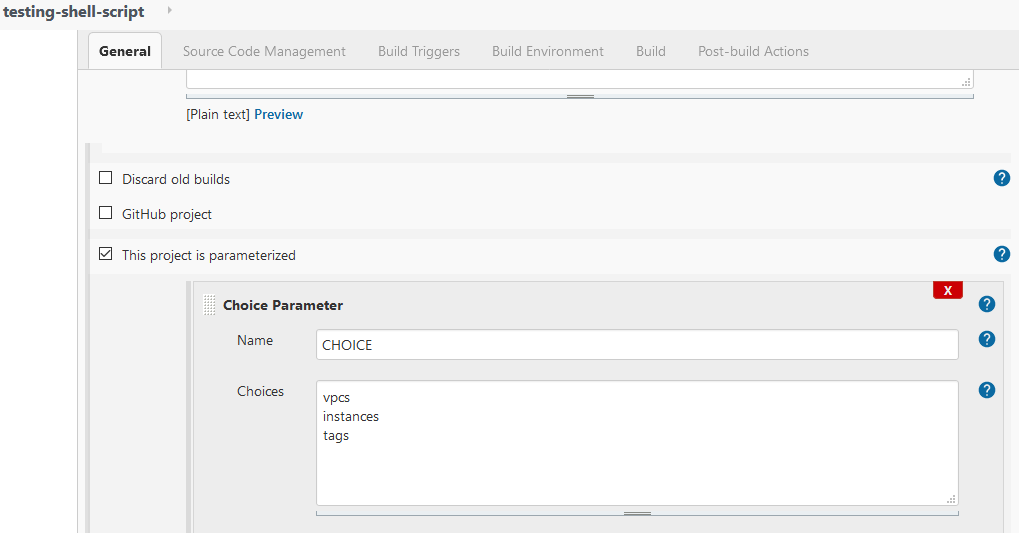
aws ec2 describe-vpcs --region $USERINPUT | jq ".Vpcs[].VpcId"

else

echo "INVALID REGION"

fi

**JENKINS:**



#!/bin/bash

CHOICE=$CHOICE

if [ ${CHOICE} = 'vpcs' ]; then

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

elif [ ${CHOICE} = 'instances' ]; then

aws ec2 describe-instances | jq .Reservations[].Instances[].InstanceId

elif [ ${CHOICE} = 'tags' ]; then

aws ec2 describe-tags | jq .Tags[].Value

else:

echo "Invalid Input, Please Select between vpcs/instances/tags."

fi

