#!/bin/bash

set -x

VPCCIDR=$(cat CidrBlock)

SUBNETS=`cat SubnetRanges`

#Create VPC

aws ec2 create-vpc --cidr-block ${VPCCIDR} --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=TEST-VPC-1}]' | tee vpcdetails

VPCID=$(cat vpcdetails | jq ".Vpc.VpcId" | tr -d '"')

#Create Subnets one by one

rm -f subnetdetails

for RANGE in ${SUBNETS}

do

aws ec2 create-subnet --vpc-id ${VPCID} --cidr-block ${RANGE} --availability-zone us-east-1a | tee -a subnetdetails

sleep 2

done

rm -f vpcdetails

#!/bin/bash

#rm -f \*-vpcinfo

region=$1

echo "Retreving VPC Information for the region $region" | tee -a ${region}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region}-vpcinfo

aws ec2 describe-vpcs --region ${region} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${region}-vpcinfo

aws ec2 describe-vpcs --region us-east-1 | jq '.Vpcs[]|.VpcId,.CidrBlock,.OwnerId' | paste - - - | tr -d '"' | awk '{print $3 "-->" $2 "==>" $1}'

#Replace String All Occurrences In The File

sed -i "s/.\*PasswordAuthentication.\*/PasswordAuthentication no/g" /etc/ssh/sshd\_config

#Replace String First Occurrence In The File

sed -i '0,/.\*PasswordAuthentication.\*/s//PasswordAuthentication yes/g' /etc/ssh/sshd\_config

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

#!/bin/bash

#rm -f \*-vpcinfo

region1=$1

region2=$2

region3=$3

region4=$4

echo "Retreving VPC Information for the region $region1" | tee -a ${region1}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region1}-vpcinfo

aws ec2 describe-vpcs --region ${region1} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

echo "Retreving VPC Information for the region $region2" | tee -a ${region2}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region2}-vpcinfo

aws ec2 describe-vpcs --region ${region2} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

echo "Retreving VPC Information for the region $region3" | tee -a ${region3}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region3}-vpcinfo

aws ec2 describe-vpcs --region ${region3} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

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

#!/bin/bash

#rm -f \*-vpcinfo

region1=$1

region2=$2

region3=$3

region4=$4

echo "Retreving VPC Information for the region $region1" | tee -a ${region1}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region1}-vpcinfo

aws ec2 describe-vpcs --region ${region1} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

echo "Retreving VPC Information for the region $region2" | tee -a ${region2}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region2}-vpcinfo

aws ec2 describe-vpcs --region ${region2} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

echo "Retreving VPC Information for the region $region3" | tee -a ${region3}-vpcinfo

echo "--------------------------------------------------" | tee -a ${region3}-vpcinfo

aws ec2 describe-vpcs --region ${region3} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

#!/bin/bash

REGIONS=$@

for REGION in $REGIONS

do

echo "Retreving VPC Information for the region ${REGION}" | tee -a ${REGION}-vpcinfo

echo "--------------------------------------------------" | tee -a ${REGION}-vpcinfo

aws ec2 describe-vpcs --region ${REGION} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

done

#!/bin/bash

REGIONS=$\*

for REGION in $REGIONS

do

echo "Retreving VPC Information for the region ${REGION}" | tee -a ${REGION}-vpcinfo

echo "--------------------------------------------------" | tee -a ${REGION}-vpcinfo

aws ec2 describe-vpcs --region ${REGION} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{print $1 "-->" $2}' | tee -a ${REGION}-vpcinfo

done

$ set -- "arg 1" "arg 2" "arg 3"

$ for word in $\*; do echo "$word"; done

arg

1

arg

2

arg

3

$ for word in $@; do echo "$word"; done

arg

1

arg

2

arg

3

$ for word in "$\*"; do echo "$word"; done

arg 1 arg 2 arg 3

$ for word in "$@"; do echo "$word"; done

arg 1

arg 2

arg 3

#!/bin/bash

echo "You Have Asked For Information For $# Regions"

REGIONS=$\*

if [ $# -eq 0 ]

then

echo "You have Given $# Inputs.Please provide Valid Input."

else

for REGION in $REGIONS

do

echo "Retreving VPC Information for the region ${REGION}" | tee -a ${REGION}-vpcinfo

echo "--------------------------------------------------" | tee -a ${REGION}-vpcinfo

aws ec2 describe-vpcs --region ${REGION} | jq '.Vpcs[]|.VpcId,.CidrBlock' | paste - - | tr -d '"' | awk '{>

done

fi

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

#!/bin/bash

read -p "Enter Your Region:" REGION

if [ -z $REGION ]; then

echo "Please Give A Valid Input"

else

CHECK=$(curl -sL https://ec2.$REGION.amazonaws.com/)

#if [ ${CHECK}='' ]

if [ -z "${CHECK}" ]

then

echo "INVALID REGION"

else

echo "Your REGION IS $REGION"

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

fi

fi

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

#!/bin/bash

read -p "Enter Your Region:" REGION

if [ -n $REGION ]; then

CHECK=$(curl -sL https://ec2.$REGION.amazonaws.com/)

#if [ ${CHECK}='' ]

if [ -z "${CHECK}" ]

then

echo "INVALID REGION"

else

echo "Your REGION IS $REGION"

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

fi

else

echo "Please Give A Valid Input"

fi

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

#Using str willr result in following Error

#str-sting.sh: line 3: [: str: unary operator expected

#!/bin/bash

read -p "Enter Your Region:" REGION

if [ str $REGION ]; then

echo "Please Give A Valid Input"

else

CHECK=$(curl -sL https://ec2.$REGION.amazonaws.com/)

#if [ ${CHECK}='' ]

if [ str "${CHECK}" ]

then

echo "INVALID REGION"

else

echo "Your REGION IS $REGION"

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

fi

fi

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

#!/bin/bash

REGIONS=$@

echo $REGIONS

for REGION in $REGIONS

do

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

if [ $? -eq 0 ]

then

echo "Region $REGION is VALID"

else

echo "Region $REGION is INVALID"

fi

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

done

echo "You Have Given $# Regions As Input"

#CREATING EBS VOLUME IN MULTIPLE AZ SUBNETS

#!/bin/bash

I=1

TEMPLATE='lt-05bf3ebf4cd8e9488'

for subnet in 'subnet-0596d277b10eedf7f' 'subnet-0ac37305c1dc6efc5' 'subnet-016ab66c4fdc49c80'

do

echo "Creating EC2 Instance in $subnet ..."

aws ec2 run-instances --instance-type t2.nano --launch-template LaunchTemplateId=$TEMPLATE --subnet-id $subnet --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Testserver-'${I}'}]' #>> /dev/null 2>&1

echo "Created EC2 Machine with the name Testserver-${I}"

I=$((I+1))

done

#ANOTHER TYPE OF FOR LOOP

#!/bin/bash

subnets=('subnet-0596d277b10eedf7f' 'subnet-0ac37305c1dc6efc5' 'subnet-016ab66c4fdc49c80')

for (( I=0; I<3; I++ ))

do

echo "Creating Subnet In ${subnets[$I]}"

aws ec2 run-instances --instance-type t2.nano --launch-template LaunchTemplateId=$TEMPLATE --subnet-id $subnet --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=Testserver-'${I}'}]' #>> /dev/null 2>&1

echo "Created EC2 Machine with the name Testserver-${I}"

sleep 1

done

#!/bin/bash

I=0

while [ $I -lt 10 ]

do

echo "Creating Subnet In $I"

sleep 1

I=$((I+1))

done

#!/bin/bash

aws\_region=$1

case $aws\_region in

eastus)

echo "Its has 6 AZ with prefix us-east-1"

;;

eastus2)

echo "Its has 3 AZ with prefix us-east-2"

;;

westus)

echo "Its has 3 AZ with prefix us-west-1"

;;

\*)

echo "Region is not part of the list"

;;

esac