

AWS Resource Listing Automation Script

Introduction

The AWS Resource Listing Automation Script is a Bash script designed to automate the process of listing various AWS resources within a specified region. The project leverages AWS CLI commands to retrieve information on resources for services such as EC2, S3, RDS, and more. The primary goal of this project is to streamline cloud resource management, minimize manual errors, and increase efficiency.

Objectives

The main objective of this project is to simplify the task of listing AWS resources for different services by automating the process using a Bash script. By doing so, it reduces the need for manual execution of AWS CLI commands, saving time and effort.

Technical Details

The script works by prompting the user to input two parameters: the AWS region and the AWS service they want to list resources for. It then uses AWS CLI commands specific to each service to retrieve the relevant data. The script checks for AWS CLI installation and configuration before proceeding with the resource listing. It supports services such as EC2, S3, RDS, IAM, CloudFront, and more.

Process

1. The script first checks if the necessary number of arguments (region and service) are provided.
2. It verifies if AWS CLI is installed and properly configured.
3. Depending on the service selected by the user, the script executes the appropriate AWS CLI command to list the resources.
4. The user receives the output of the command, which displays the requested AWS resources in the specified region.

Code

```
#!/bin/bash
```

```
#####  
#####  
# Author: Mahesh Kuruva  
# Version: v0.0.1
```

```

# Script to automate the process of listing all the resources in an AWS account
# Usage: ./aws_resource_list.sh <aws_region> <aws_service>
#####
#####

if [ $# -ne 2 ]; then
    echo "Usage: ./aws_resource_list.sh <aws_region> <aws_service>"
    exit 1
fi

aws_region=$1
aws_service=$2

if ! command -v aws &> /dev/null; then
    echo "AWS CLI is not installed. Please install the AWS CLI and try again."
    exit 1
fi

if [ ! -d ~/.aws ]; then
    echo "AWS CLI is not configured. Please configure the AWS CLI and try again."
    exit 1
fi

case $aws_service in
    ec2)
        echo "Listing EC2 Instances in $aws_region"
        aws ec2 describe-instances --region $aws_region
        ;;
    rds)
        echo "Listing RDS Instances in $aws_region"
        aws rds describe-db-instances --region $aws_region
        ;;
    s3)
        echo "Listing S3 Buckets in $aws_region"
        aws s3api list-buckets --region $aws_region
        ;;
    cloudfront)
        echo "Listing CloudFront Distributions in $aws_region"
        aws cloudfront list-distributions --region $aws_region
        ;;
    vpc)
        echo "Listing VPCs in $aws_region"

```

```

    aws ec2 describe-vpcs --region $aws_region
;;
iam)
    echo "Listing IAM Users in $aws_region"
    aws iam list-users --region $aws_region
;;
route53)
    echo "Listing Route53 Hosted Zones in $aws_region"
    aws route53 list-hosted-zones --region $aws_region
;;
cloudwatch)
    echo "Listing CloudWatch Alarms in $aws_region"
    aws cloudwatch describe-alarms --region $aws_region
;;
cloudformation)
    echo "Listing CloudFormation Stacks in $aws_region"
    aws cloudformation describe-stacks --region $aws_region
;;
lambda)
    echo "Listing Lambda Functions in $aws_region"
    aws lambda list-functions --region $aws_region
;;
sns)
    echo "Listing SNS Topics in $aws_region"
    aws sns list-topics --region $aws_region
;;
sqs)
    echo "Listing SQS Queues in $aws_region"
    aws sqs list-queues --region $aws_region
;;
dynamodb)
    echo "Listing DynamoDB Tables in $aws_region"
    aws dynamodb list-tables --region $aws_region
;;
ebs)
    echo "Listing EBS Volumes in $aws_region"
    aws ec2 describe-volumes --region $aws_region
;;
*)
    echo "Invalid service. Please enter a valid service."
    exit 1
;;

```

esac

Conclusion

The AWS Resource Listing Automation Script automates the process of listing AWS resources, improving efficiency and reliability for cloud management tasks. The script can easily be extended to add new services, making it highly scalable and adaptable. Overall, this project has contributed to my understanding of AWS services, Bash scripting, and cloud automation.