SHELL SCRIPTING PROJECTS

WE WILL write a shell script to connect to aws and list all the active service in aws

#what are active resounces in your aws account – any one can find it using script

Aws-resources-list.sh <region> <service>

Us-east-1 ec2 # we can pass argument and

#!/bin/bash

# best practices follow it while writing the script

######################

#This script will list all the resources in an aws accounts

#Author: Deepanshu Kumar

#version: v0.0.1

#Following are the supported aws services by the script

#EC2

#s3

#EBS

#RDS

#Dynamodb

#ELB

#Cloudfront

#Cloudformation

#SNS

#SQS

#Route53

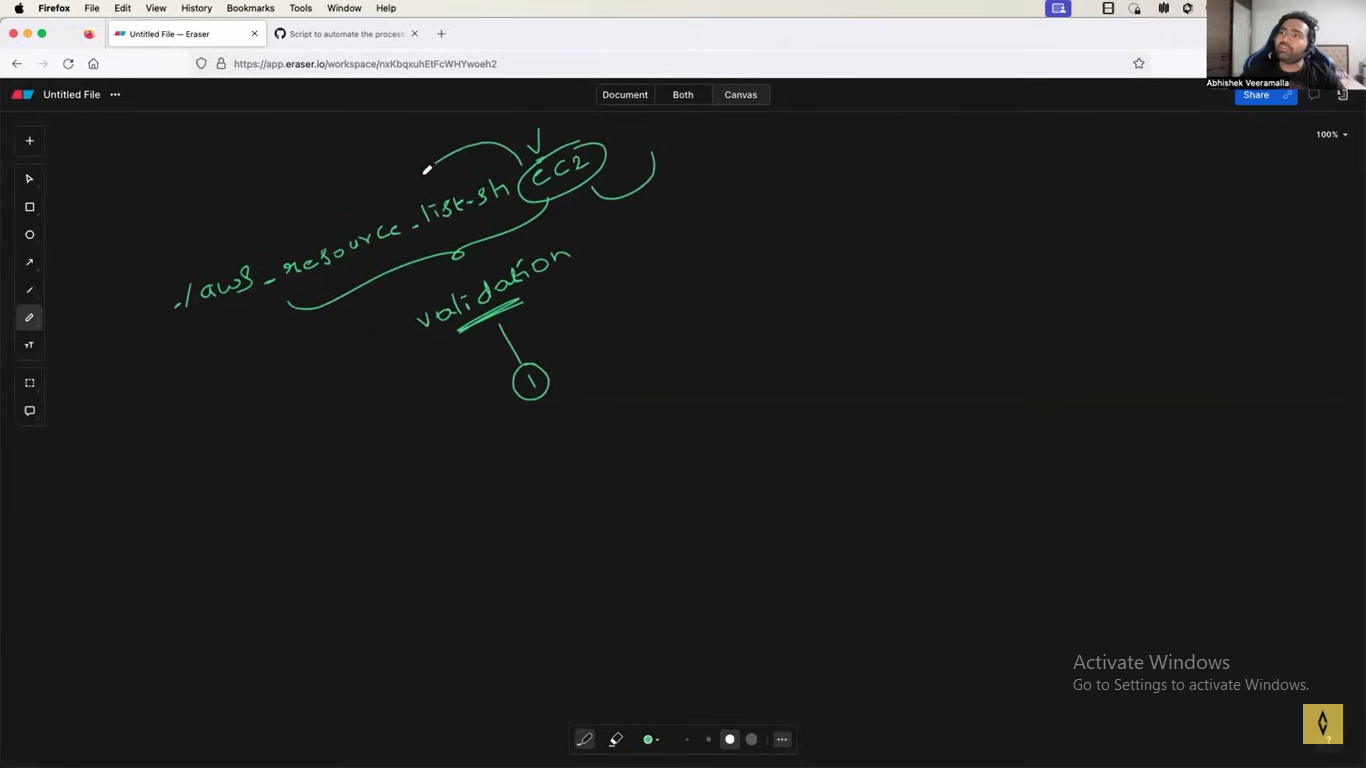
#VPC

#IAM

# Usage: ./aws\_resource\_list.sh  <aws\_region> <aws\_service>

#Example: ./aws\_resource\_list.sh us-east-1 EC2

# ##########################################################



#first perform command line arguments – as if you are taking input from user as cli – then first you need to validate this inside shell script

Here we need region and service naem as argument which is input

# Check if the required number of arguments are passed

if [ $# -ne 2 ]; then

    echo "Usage: ./aws\_resource\_list.sh  <aws\_region> <aws\_service>"

    echo "Example: ./aws\_resource\_list.sh us-east-1 ec2"

    exit 1

fi

# $# - take input from cli if it less than 2 then thorw error - script return with exit code 1

#

#validation part is done

Now communivate with aws using shell scritp, we use aws cli – shell script communicate with aws cli – cli comunivate with aws acoaunt

Aws cli communicate with aws via the api

# Check if the AWS CLI is installed

if ! command -v aws &> /dev/null; then

    echo "AWS CLI is not installed. Please install the AWS CLI and try again."

    exit 1

fi

# Check if the AWS CLI is configured

if [ ! -d ~/.aws ]; then

    echo "AWS CLI is not configured. Please configure the AWS CLI and try again."

    exit 1

fi

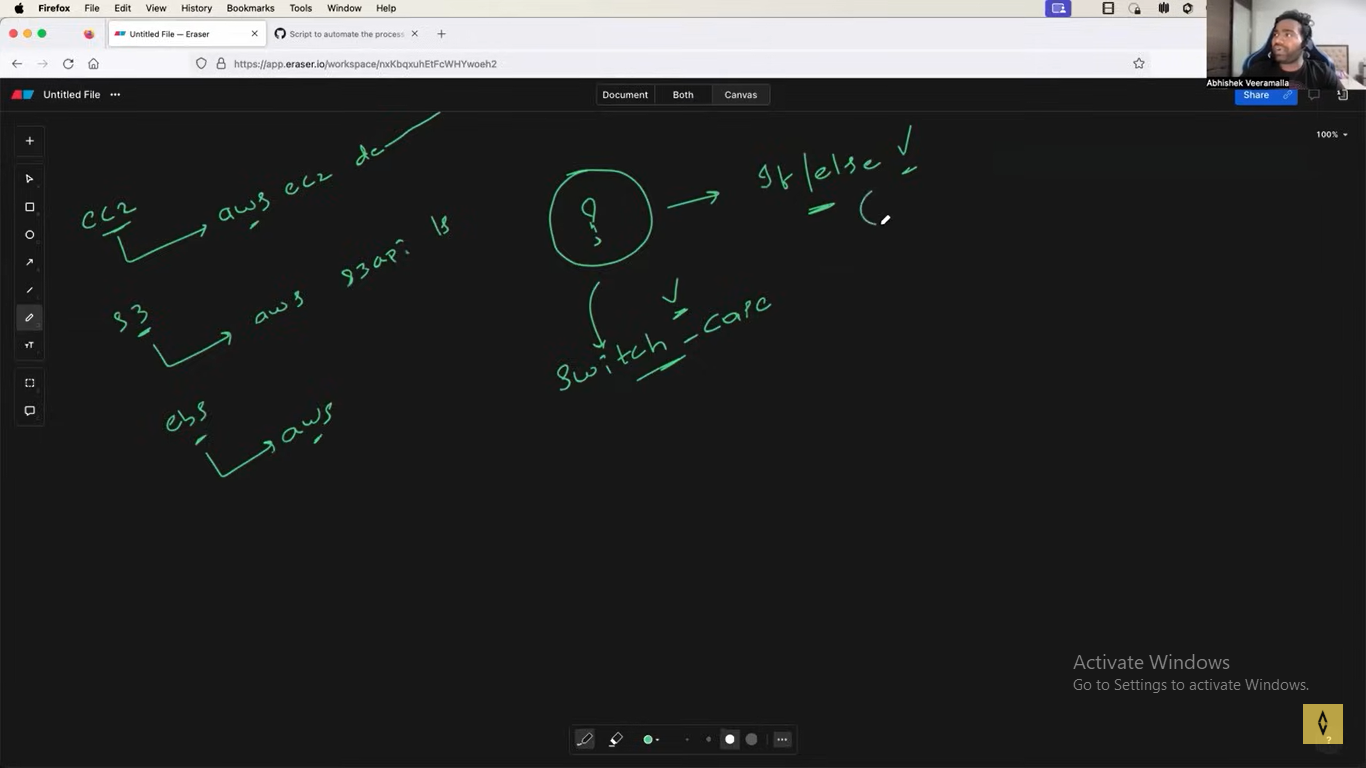
#check if there is no directory of ./aws then exit code with return 1

NOW if user input it then particaluar code will be executed

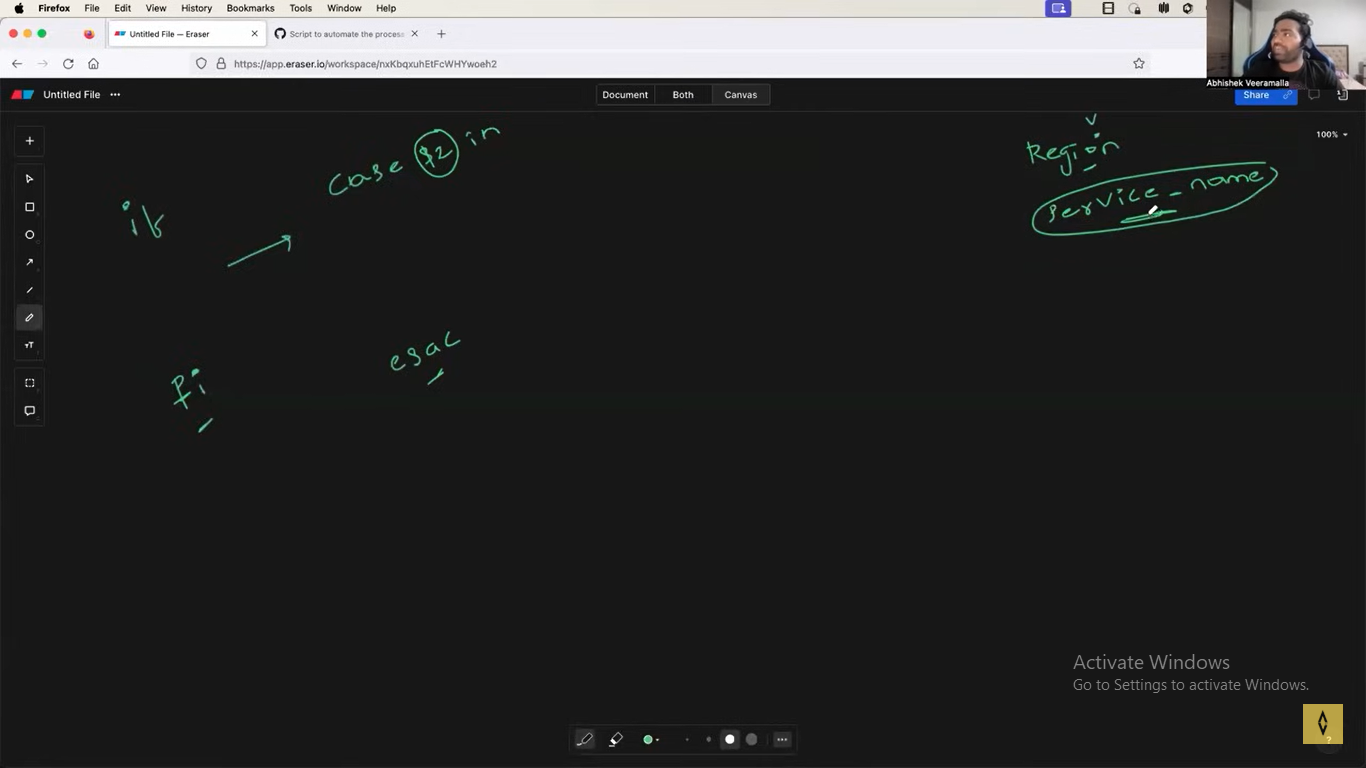
If user input ec2 = then we need to execute – aws ec2 descirbe

If user input s3 then – aws s3 app –ls

Ebs- is user execute ebs – then execute diff comd



#2e can use if else or swritch as well – switch is better option for us



# what ever the second argument – case $2 – service name –

AWS cli documtation – use cmd – s3 , ec2 – just remneber basic ec2 ,s3 and

case $2 in

   EC2)

       aws ec2 describe-instances --region $1

       ;;

   S3)

       aws s3api list-buckets --region $1

       ;;

   RDS)

       aws rds describe-db-instances --region $1

       ;;

   Dynamodb)

       aws dynamodb list-tables --region $1

       ;;

   Lambda)

       aws lambda list-functions --region $1

       ;;

   \*)

       echo "Invalid service name. Please use EC2, S3, RDS, Dynamodb, or Lambda."

       ;;

esac

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_resource.sh

-bash: ./aws\_resource.sh: No such file or directory

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_list\_resources.sh

Usage: ./aws\_resource\_list.sh <aws\_region> <aws\_service>

Example: ./aws\_resource\_list.sh us-east-1 ec2

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_list\_resources.sh ap-south-1 ec2

AWS CLI is not installed. Please install the AWS CLI and try again.

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_list\_resources.sh ec2

Usage: ./aws\_resource\_list.sh <aws\_region> <aws\_service>

Example: ./aws\_resource\_list.sh us-east-1 ec2

#install aws cli

**Access key**

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

| **Access key** | **Secret access key** |
| --- | --- |
| AKIAQE43KAB4K5NPQZUJ | 30fJZ8AbMU7FLJJnLgPtbfv9b6zNeF4xoeNgwnsD  **Hide** |

#!/bin/bash

# best practices follow it while writing the script

######################

#This script will list all the resources in an aws accounts

#Author: Deepanshu Kumar

#version: v0.0.1

#Following are the supported aws services by the script

#EC2

#s3

#EBS

#RDS

#Dynamodb

#ELB

#Cloudfront

#Cloudformation

#SNS

#SQS

#Route53

#VPC

#IAM

# Usage: ./aws\_resource\_list.sh  <aws\_region> <aws\_service>

#Example: ./aws\_resource\_list.sh us-east-1 EC2

# ##########################################################

#check if required no of arguments are passed

# Check if the required number of arguments are passed

if [ $# -ne 2 ]; then

    echo "Usage: ./aws\_resource\_list.sh  <aws\_region> <aws\_service>"

    echo "Example: ./aws\_resource\_list.sh us-east-1 ec2"

    exit 1

fi

# Check if the AWS CLI is installed

if ! command -v aws &> /dev/null; then

    echo "AWS CLI is not installed. Please install the AWS CLI and try again."

    exit 1

fi

# Check if the AWS CLI is configured

if [ ! -d ~/.aws ]; then

    echo "AWS CLI is not configured. Please configure the AWS CLI and try again."

    exit 1

fi

case $2 in

   EC2)

       aws ec2 describe-instances --region $1

       ;;

   S3)

       aws s3api list-buckets --region $1

       ;;

   RDS)

       aws rds describe-db-instances --region $1

       ;;

   Dynamodb)

       aws dynamodb list-tables --region $1

       ;;

   Lambda)

       aws lambda list-functions --region $1

       ;;

   \*)

       echo "Invalid service name. Please use EC2, S3, RDS, Dynamodb, or Lambda."

       ;;

esac

If second argument EC2 , S3 MATCHES WITH services then execute particular command

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ aws configure

AWS Access Key ID [None]: AKIAQE43KAB4K5NPQZUJ

AWS Secret Access Key [None]: 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

Default region name [None]: ap-south-1

Default output format [None]: json

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_list\_resources.sh

Usage: ./aws\_resource\_list.sh <aws\_region> <aws\_service>

Example: ./aws\_resource\_list.sh us-east-1 ec2

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ ./aws\_list\_resources.sh ap-south-1 EC2

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ chmod 711 aws\_list\_resources.sh

deep@DESKTOP-JFJ21I5:~/shell-script-projects$ chmod 771 aws\_list\_resources.sh

#only provide execute permission to the others, deep user has rwx , gp has rws and last has everyone has execute premiision

### **Bash Script Explanation:**

This script lists AWS resources for a given service in a specified AWS region.

### **Step 1: Check If Required Arguments Are Passed**

if [ $# -ne 2 ]; then

echo "Usage: ./aws\_resource\_list.sh <aws\_region> <aws\_service>"

echo "Example: ./aws\_resource\_list.sh us-east-1 ec2"

exit 1

fi

* if [ $# -ne 2 ]; then
  + $# represents the **number of arguments** passed to the script.
  + -ne 2 checks if the number of arguments is **not equal to 2**.
* If the condition is **true**, it prints the correct usage and exits with an **error code (1)**.

### **Step 2: Assign Arguments to Variables**

aws\_region=$1

aws\_service=$2

* $1 (first argument) is assigned to aws\_region (AWS region).
* $2 (second argument) is assigned to aws\_service (AWS service name).

### **Step 3: Check If AWS CLI Is Installed**

if ! command -v aws &> /dev/null; then

echo "AWS CLI is not installed. Please install the AWS CLI and try again."

exit 1

fi

* command -v aws &> /dev/null
  + Checks if the aws command is available.
  + &> /dev/null suppresses output.
* If AWS CLI is **not installed**, it prints an error message and exits.

### **Step 4: Check If AWS CLI Is Configured**

if [ ! -d ~/.aws ]; then

echo "AWS CLI is not configured. Please configure the AWS CLI and try again."

exit 1

fi

* -d ~/.aws checks if the ~/.aws directory exists (AWS CLI configuration directory).
* If it **does not exist**, the script prints an error and exits.

### **Step 5: List AWS Resources Based on Service Name**

case $aws\_service in

* Starts a **case statement** to match the AWS service.

### **Case Handling for AWS Services**

Each case block does the following:

1. Prints a message indicating which resource is being listed.
2. Runs the corresponding AWS CLI command.

#### **1. EC2 (Elastic Compute Cloud)**

ec2)

echo "Listing EC2 Instances in $aws\_region"

aws ec2 describe-instances --region $aws\_region

;;

* Lists **EC2 instances** in the given region.

#### **2. RDS (Relational Database Service)**

rds)

echo "Listing RDS Instances in $aws\_region"

aws rds describe-db-instances --region $aws\_region

;;

* Lists **RDS database instances**.

#### **3. S3 (Simple Storage Service)**

s3)

echo "Listing S3 Buckets in $aws\_region"

aws s3api list-buckets --region $aws\_region

;;

* Lists **S3 buckets**.

#### **4. CloudFront**

cloudfront)

echo "Listing CloudFront Distributions in $aws\_region"

aws cloudfront list-distributions --region $aws\_region

;;

* Lists **CloudFront distributions**.

#### **5. VPC (Virtual Private Cloud)**

vpc)

echo "Listing VPCs in $aws\_region"

aws ec2 describe-vpcs --region $aws\_region

;;

* Lists **VPCs (Virtual Private Clouds)**.

#### **6. IAM (Identity and Access Management)**

iam)

echo "Listing IAM Users in $aws\_region"

aws iam list-users --region $aws\_region

;;

* Lists **IAM users**.

#### **7. Route 53 (Domain Name Service)**

route5)

echo "Listing Route53 Hosted Zones in $aws\_region"

aws route53 list-hosted-zones --region $aws\_region

;;

* Lists **Route 53 hosted zones** (incorrectly labeled as route5, should be route53).

#### **8. CloudWatch**

cloudwatch)

echo "Listing CloudWatch Alarms in $aws\_region"

aws cloudwatch describe-alarms --region $aws\_region

;;

* Lists **CloudWatch alarms**.

#### **9. CloudFormation**

cloudformation)

echo "Listing CloudFormation Stacks in $aws\_region"

aws cloudformation describe-stacks --region $aws\_region

;;

* Lists **CloudFormation stacks**.

#### **10. Lambda (Serverless Functions)**

lambda)

echo "Listing Lambda Functions in $aws\_region"

aws lambda list-functions --region $aws\_region

;;

* Lists **Lambda functions**.

#### **11. SNS (Simple Notification Service)**

sns)

echo "Listing SNS Topics in $aws\_region"

aws sns list-topics --region $aws\_region

;;

* Lists **SNS topics**.

#### **12. SQS (Simple Queue Service)**

sqs)

echo "Listing SQS Queues in $aws\_region"

aws sqs list-queues --region $aws\_region

;;

* Lists **SQS queues**.

#### **13. DynamoDB (NoSQL Database)**

dynamodb)

echo "Listing DynamoDB Tables in $aws\_region"

aws dynamodb list-tables --region $aws\_region

;;

* Lists **DynamoDB tables**.

#### **14. EBS (Elastic Block Store)**

ebs)

echo "Listing EBS Volumes in $aws\_region"

aws ec2 describe-volumes --region $aws\_region

;;

* Lists **EBS volumes**.

### **Step 6: Handle Invalid Services**

\*)

echo "Invalid service. Please enter a valid service."

exit 1

;;

esac

* \*) is the **default case**, executed if the service name does **not** match any listed options.
* Prints an error message and **exits with status 1**.

## ****Summary****

* Ensures **two arguments** (region & service) are provided.
* **Validates AWS CLI** is installed and configured.
* Uses a **case statement** to call the correct **AWS CLI command** for the requested service.
* Prints appropriate messages before executing each command.
* Exits with an error message if the service name is **invalid**.

## ****Example Usage****

./aws\_resource\_list.sh us-east-1 ec2

**Output:**

Listing EC2 Instances in us-east-1

<EC2 instance details>

### **Potential Issues and Fixes**

| **Issue** | **Fix** |
| --- | --- |
| **Typo: route5 instead of route53** | Rename route5 to route53 in the script. |
| **Not all AWS services require --region (e.g., IAM)** | Remove --region $aws\_region for iam and route53. |

Would you like any improvements or modifications? 🚀

<https://chatgpt.com/share/67cc6351-69f8-8013-b2d6-d8f473b2a9f8>

<https://gist.github.com/iam-veeramalla/24ba6d64ffaffb925e7d28166266ce49>