

# Mini Project Report: Creating AWS Resources with Bash Functions and Introducing Arrays

**Author:** Oluwaseun Osunsola

**Environment:** AWS, VSCode & Hyper Terminal

**Project Link:** [GitHub Repository](#)

## Introduction

This report documents the mini project on automating AWS resource creation using Bash scripting. The goal is to create a shell script that provisions EC2 instances and S3 buckets through functions, while introducing arrays for handling multiple similar resources efficiently. We'll use the AWS CLI to interact with AWS services programmatically.

The project emphasizes modularity with functions, error handling, and basic data structures like arrays. By following this report, you'll build and run the script step by step on your local machine, deploying real EC2 instances and S3 buckets in your AWS account. This hands-on approach allows you to practice as you read—copy code blocks into a file (e.g., `aws-resources.sh`), make it executable with `chmod +x aws-resources.sh`, and test incrementally.

### Important Notes:

- As of October 18, 2025, Amazon Linux 2 (AL2) is still supported until June 30, 2026. However, for best practices, we'll update the AMI to Amazon Linux 2023 (AL2023) using a dynamic SSM parameter (`resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64`), which the AWS CLI can resolve automatically to the latest ID (e.g., ami-0b82cd78a35d421a2 for eu-west-2 as of recent checks).
- Use a free-tier eligible AWS account to minimize costs (t2.micro instances qualify for 750 hours/month). Always clean up resources afterward.
- Run commands in a terminal on Linux/macOS (or WSL on Windows).

## Prerequisites

Before starting, set up your environment. This ensures the script runs without errors.

1. **Sign Up for an AWS Account:** If you don't have one, create a free account at [aws.amazon.com](#). Note your default region (we'll use `eu-west-2`—London—but change if needed).

**Note:** To learn how to create AWS account, checkout this my project: [AWS Account Creation](#)

2. **Install and Configure AWS CLI:**

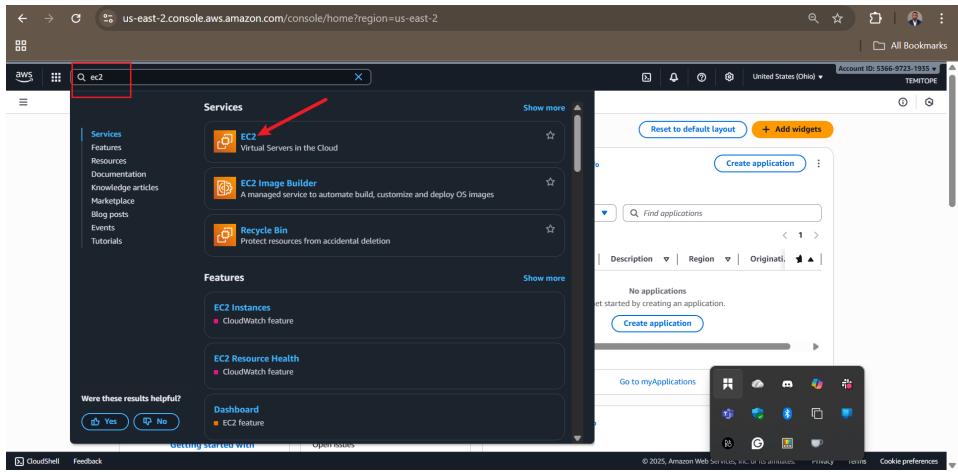
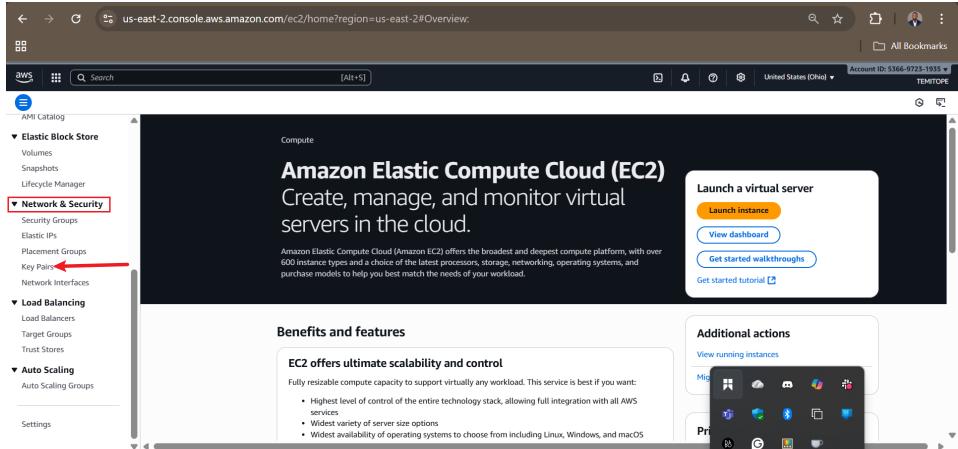
- Download from [AWS CLI documentation](#) and install.
- Run `aws configure` to set your Access Key ID, Secret Access Key, region (e.g., `eu-west-2`), and output format (json).
- Test with `aws sts get-caller-identity`—it should show your account details.

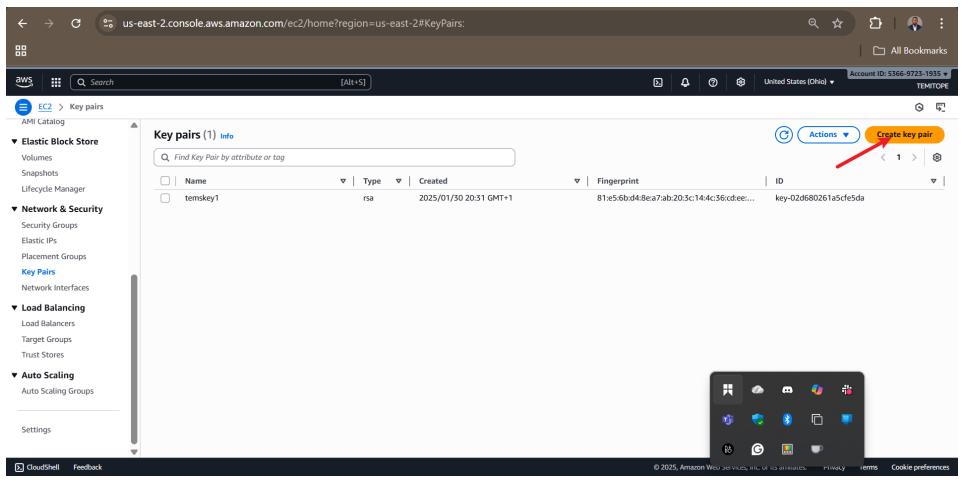
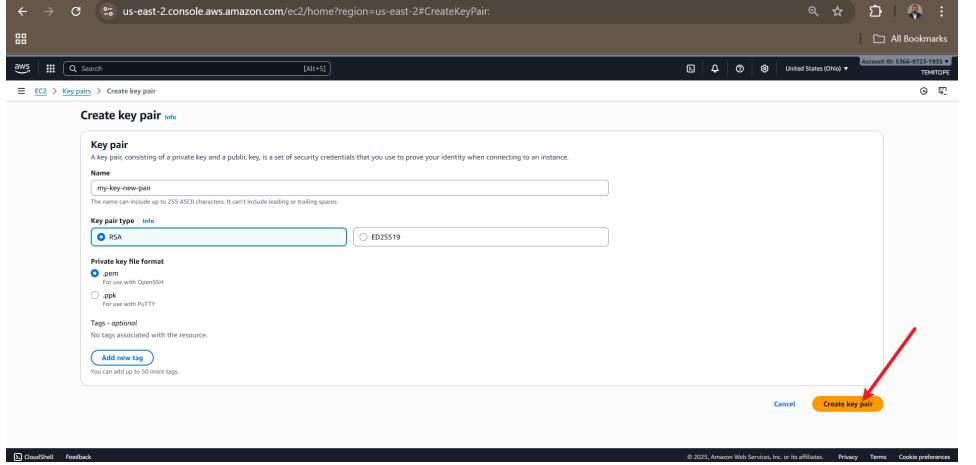
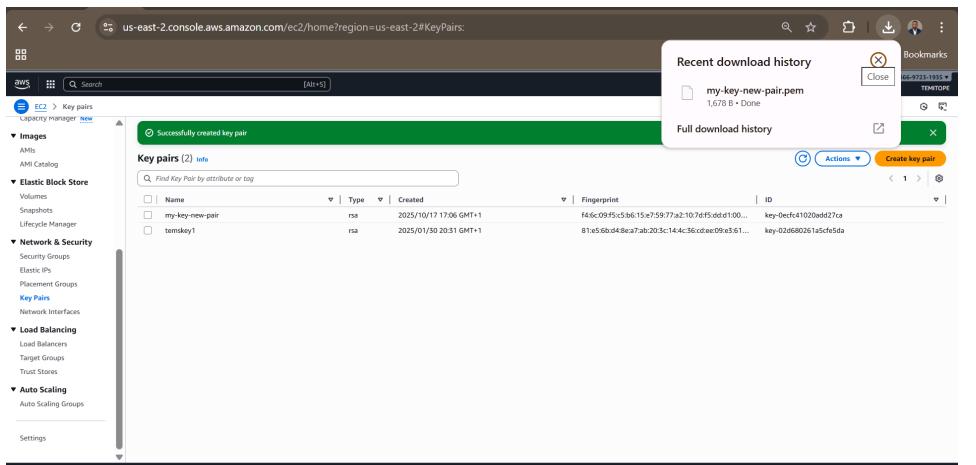
- Optionally, set `export AWS_PROFILE=default` if using profiles.

**Note:** To learn how to install and configure AWS CLI, check out this my project: [AWS CLI Installation and Configuration](#)

### 3. Create an EC2 Key Pair:

- Log in to the AWS Management Console.
- Navigate to EC2 > Network & Security > Key Pairs.
- Click "Create key pair", name it `MyKeyPair`, select PEM format, and download the `.pem` file (keep it secure for SSH access later).
- Hands-on: Do this now in your browser. Here's what the console looks like:

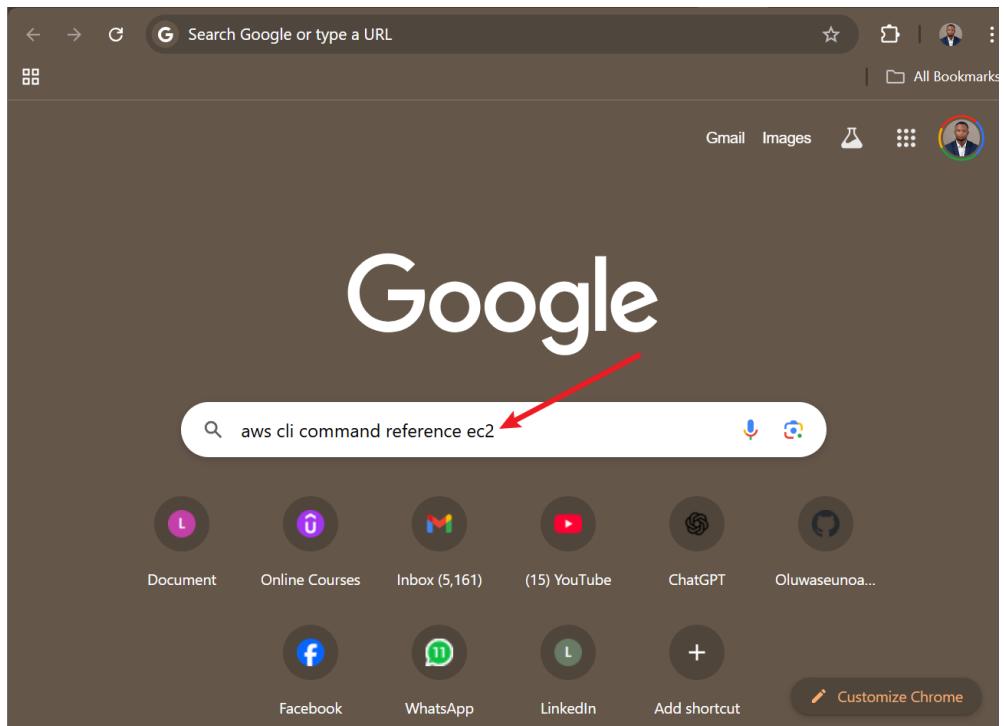
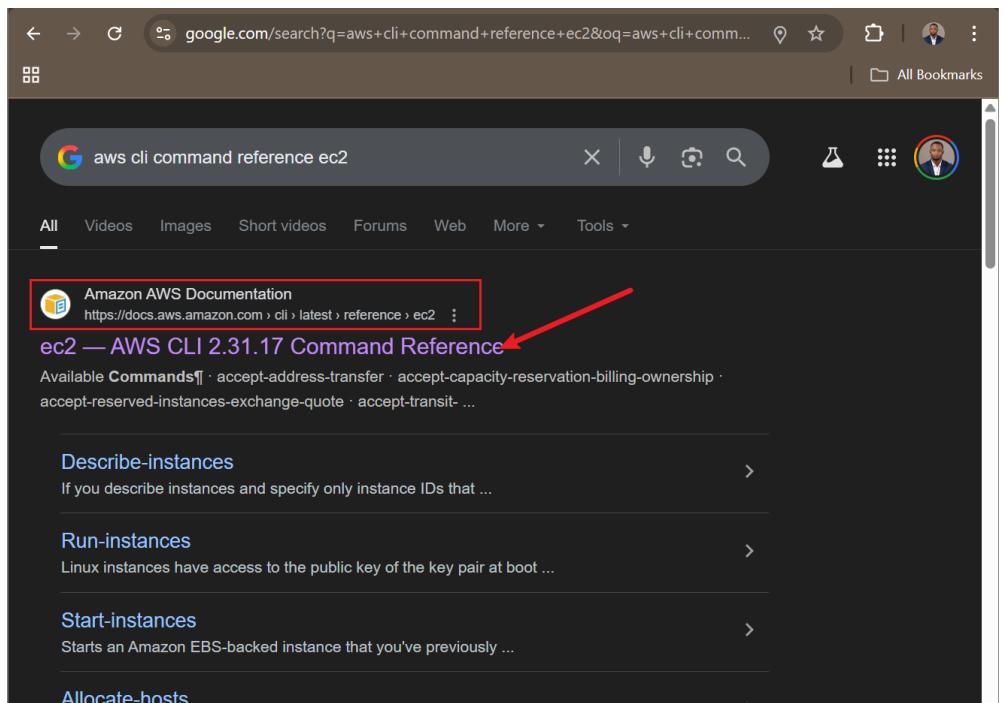
Step	Action	Screenshot
1	Search for EC2 and click on it	
2	Find "Key Pairs" under Network & Security on the left sidebar and click on it	

Step	Action	Screenshot
3	On the Key Pairs page, click "Create key pair"	
4	Name key pair MyKeyValuePair, select PEM type and format, then click "Create key pair"	
5	Key pair created successfully - download the .pem file immediately	

#### 4. Understand AWS CLI for EC2:

- Refer to the official docs: Search for "run-instances" in [AWS CLI EC2 Reference](#).
- Example command: `aws ec2 run-instances --image-id ami-0cd59ecaf368e5ccf --instance-type t2.micro --count 1 --key-name MyKeyValuePair --region eu-west-2`.
- Hands-on: Open the docs in your browser and search for "run-instances" to explore options. Here's what the console looks like:

Step	Action	Screenshot
------	--------	------------

Step	Action	Screenshot
1	Search for "AWS CLI command reference EC2"	 A screenshot of a web browser window. The search bar at the top contains the text "aws cli command reference ec2". A red arrow points from the text in the search bar to the search bar itself. Below the search bar, there is a grid of icons for various services like Document, Online Courses, Inbox, YouTube, ChatGPT, etc.
2	Click on the first link showing "CLI latest reference EC2"	 A screenshot of a search results page from Google. The search bar shows the query "aws cli command reference ec2". The results list includes a link to the "Amazon AWS Documentation" for "ec2 — AWS CLI 2.31.17 Command Reference". This link is highlighted with a red box and a red arrow points to it. Below the link, there is a brief description of the documentation and several other command links.

Step	Action	Screenshot
3	Now on AWS CLI Commands official documentation webpage	
4	Search for "run-instances" with Ctrl+F and click on it when found	

Now, create an empty file `aws-resources.sh` in your terminal: `touch aws-resources.sh`.

```
MINGW64:/c/Users/HP/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting
$ touch aws-resources.sh
$ ls aws-resources.sh
aws-resources.sh
$ [
```

## The Script Breakdown

We'll build the script progressively. For each section, I'll show the code block, explain it line by line, and guide you to add and test it. The full script uses functions for modularity, checks for prerequisites, and handles different environments (local, testing, production).

### 1. Shebang and Environment Variable

Add this to the top of your file:

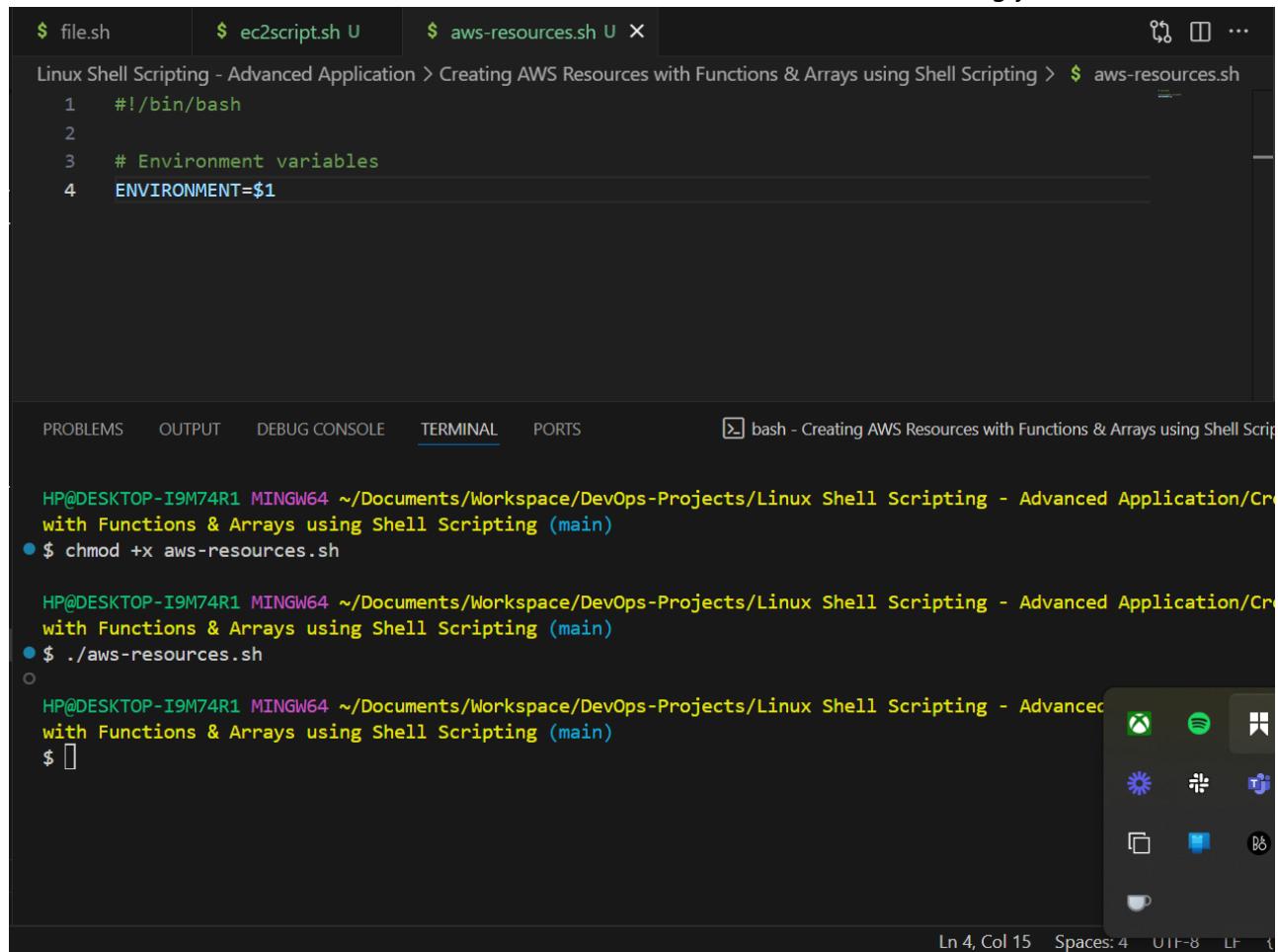
```
#!/bin/bash

# Environment variables
ENVIRONMENT=$1
```

- **Explanation:**

- `#!/bin/bash`: Specifies the interpreter (Bash shell).
- `ENVIRONMENT=$1`: Captures the first command-line argument (e.g., "local") as a variable. This allows running the script in different modes.

- **Hands-on:** Save the file. Run `./aws-resources.sh local`—it should do nothing yet, but no errors.



The screenshot shows a terminal window with the following content:

```
file.sh      ec2script.sh U      aws-resources.sh U X
Linux Shell Scripting - Advanced Application > Creating AWS Resources with Functions & Arrays using Shell Scripting > $ aws-resources.sh
1 #!/bin/bash
2
3 # Environment variables
4 ENVIRONMENT=$1

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    bash - Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
● $ chmod +x aws-resources.sh
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
● $ ./aws-resources.sh
○
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ 
```

The terminal shows the script being saved, chmod'd to executable, run, and then nothing displayed. The status bar at the bottom right indicates "Ln 4, Col 15" and "Spaces: 4".

## 2. Check Number of Arguments

Add this function:

```
check_num_of_args() {
    # Checking the number of arguments
    if [ "$#" -ne 1 ]; then
        echo "Usage: $0 <environment>"
        exit 1
    fi
}
```

- **Explanation:**

- This function checks if exactly one argument is provided.
- `$#`: Number of arguments.
- If not 1, print usage and exit with code 1 (error).

- **Hands-on:** Add this, then call it at the bottom: `check_num_of_args "$@"`. Run `./aws-resources.sh` (no arg)—see usage message. Run with `./aws-resources.sh local`—no error.

The screenshot shows a terminal window with several tabs open. The tabs include "file.sh", "ec2script.sh", "aws-resources.sh", "README.md", and another "aws-resources.sh". The "aws-resources.sh" tab contains a shell script with a function definition for `check_num_of_args`. The script also includes environment variable assignment and a usage message. The terminal below shows the script being run with and without arguments, demonstrating its functionality.

```

#!/bin/bash
# Environment variables
ENVIRONMENT=$1

check_num_of_args() {
    # Checking the number of arguments
    if [ "$#" -ne 1 ]; then
        echo "Usage: $0 <environment>"
        exit 1
    fi
}

check_num_of_args "$@"

```

```

HP@DESKTOP-I9M74R1 MINGW64 ~\Documents\Workspace\DevOps-Projects\Linux Shell Scripting - Advanced Application\Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ ./aws-resources.sh
Usage: ./aws-resources.sh <environment>                                     Output error Because no argument was passed

HP@DESKTOP-I9M74R1 MINGW64 ~\Documents\Workspace\DevOps-Projects\Linux Shell Scripting - Advanced Application\Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ ./aws-resources.sh local                                                 Output no error because argument "local" was passed

HP@DESKTOP-I9M74R1 MINGW64 ~\Documents\Workspace\DevOps-Projects\Linux Shell Scripting - Advanced Application\Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ 

```

### 3. Activate Infrastructure Environment

Add:

```

activate_infra_environment() {
    # Acting based on the argument value
    if [ "$ENVIRONMENT" == "local" ]; then
        echo "Running script for Local Environment..."
    elif [ "$ENVIRONMENT" == "testing" ]; then
        echo "Running script for Testing Environment..."
    elif [ "$ENVIRONMENT" == "production" ]; then
        echo "Running script for Production Environment..."
    else
        echo "Invalid environment specified. Please use 'local', 'testing', or 'production'."
        exit 2
    fi
}

```

- **Explanation:**

- Uses if-elif-else to print a message based on **\$ENVIRONMENT**.
- Exits with code 2 if invalid.
- This could be extended to environment-specific logic (e.g., different regions).

- **Hands-on:** Add and call **activate\_infra\_environment** after the args check. Run **./aws-resources.sh local**—see "Running script for Local Environment...". Try **./aws-resources.sh invalid**—see error.

```

Linux Shell Scripting - Advanced Application > Creating AWS Resources with Functions & Arrays using Shell Scripting > $ aws-resources.sh
5
6  check_num_of_args() {
7      # Checking the number of arguments
8      if [ "$#" -ne 1 ]; then
9          echo "Usage: $0 <environment>"
10         exit 1
11     fi
12 }
13
14 activate_infra_environment() {
15 # Acting based on the argument value
16 if [ "$ENVIRONMENT" == "local" ]; then
17 echo "Running script for Local Environment..."
18 elif [ "$ENVIRONMENT" == "testing" ]; then
19 echo "Running script for Testing Environment..."
20 elif [ "$ENVIRONMENT" == "production" ]; then
21 echo "Running script for Production Environment..."
22 else
23 echo "Invalid environment specified. Please use 'local', 'testing', or
24 'production'."
25 exit 2
26 fi
27 }
28
29
30 check_num_of_args "$@"
31 activate_infra_environment

```

Ln 31, Col 27 Spaces: 4 UTF-8 LF ⚙️ Shell Script ⚙️ Prettier ⚙️

```

MINGW64/c/Users/HP/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting
HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ ./aws-resources.sh invalid
Invalid environment specified. Please use 'local', 'testing', or
'production'.
Error because invalid environment was passed

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ ./aws-resources.sh local
Running script for Local Environment...
No error because local environment was passed

HP@DESKTOP-I9M74R1 MINGW64 ~/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ []

```

## 4. Check AWS CLI Installation

Add:

```

# Function to check if AWS CLI is installed
check_aws_cli() {
    if ! command -v aws &> /dev/null; then
        echo "AWS CLI is not installed. Please install it before proceeding."
        return 1
    fi
}

```

- **Explanation:**
  - `command -v aws`: Checks if `aws` command exists.
  - `&> /dev/null`: Suppresses output.
  - If not installed, print message and return 1 (failure).
- **Hands-on:** Add and call `check_aws_cli || exit 1` (exits if fails). Run the script—if CLI is missing, it stops.

```

16 if [ "$ENVIRONMENT" == "local" ]; then
17 echo "Running script for Local Environment..."
18 elif [ "$ENVIRONMENT" == "testing" ]; then
19 echo "Running script for Testing Environment..."
20 elif [ "$ENVIRONMENT" == "production" ]; then
21 echo "Running script for Production Environment..."
22 else
23 echo "Invalid environment specified. Please use 'local', 'testing', or
24 'production'."
25 exit 2
26 fi
27 }
28
29 # Function to check if AWS CLI is installed
30 check_aws_cli() {
31     if ! command -v aws &> /dev/null; then
32         echo "AWS CLI is not installed. Please install it before proceeding."
33         return 1
34     fi
35 }
36
37 check_num_of_args "$@"
38 activate_infra_environment
39 check_aws_cli || exit 1

```

Ln 39, Col 24 Spaces: 4 UTF-8 LF Shell Script Prettier

```

MINGW64/c/Users/HP/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting
$ ./aws-resources.sh local
Running script for Local Environment...
$ 

```

## 5. Check AWS Profile

Add:

```

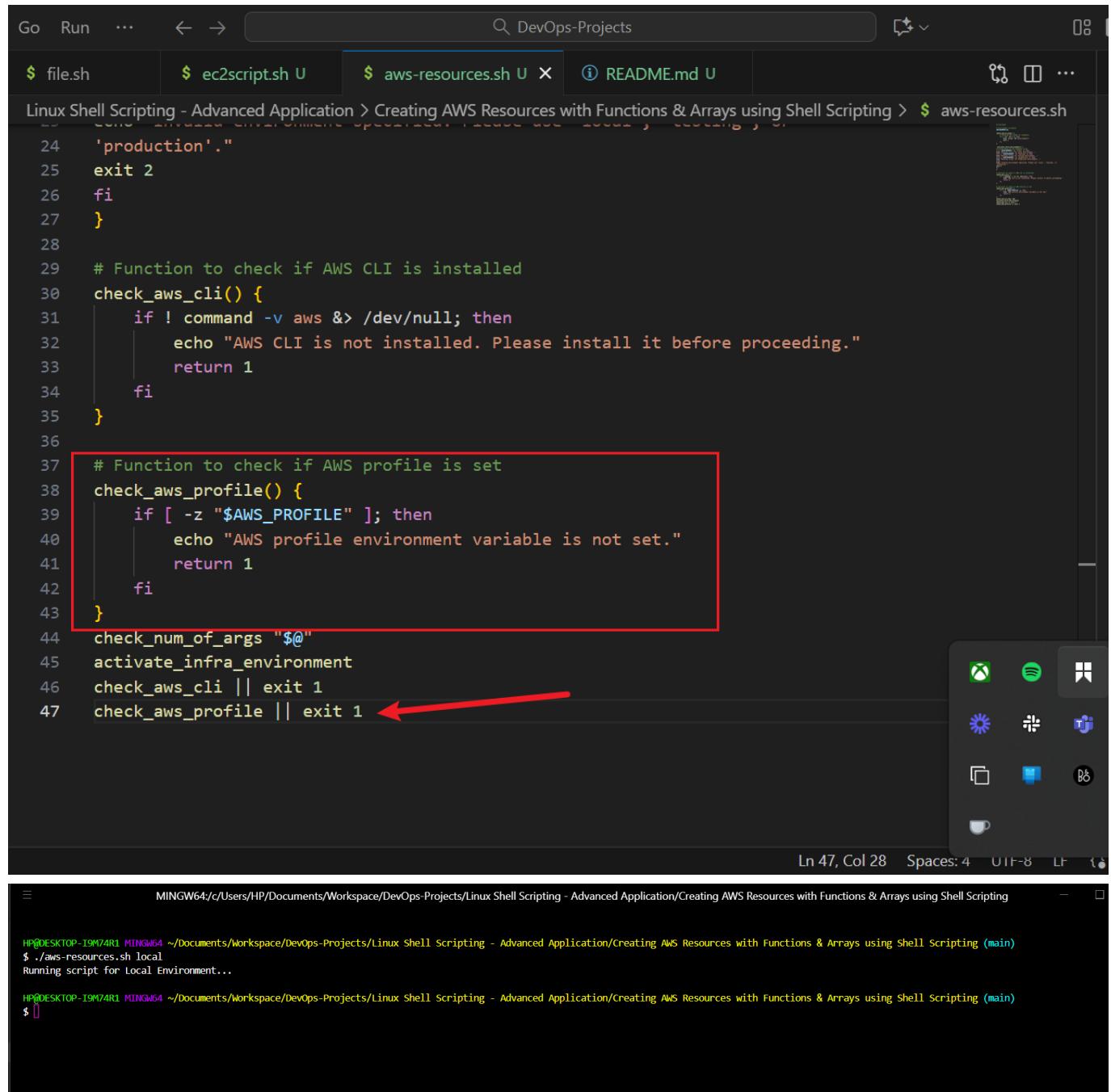
# Function to check if AWS profile is set
check_aws_profile() {
    if [ -z "$AWS_PROFILE" ]; then
        echo "AWS profile environment variable is not set."
        return 1
    fi
}

```

- **Explanation:**

- `-z "$AWS_PROFILE"`: Checks if the variable is empty.
- Returns 1 if not set (assumes default profile if unset, but enforces it for safety).

- **Hands-on:** Add and call `check_aws_profile || exit 1`. Set `AWS_PROFILE=default` if needed, then run.



The screenshot shows a terminal window with the following content:

```

MINGW64:/c/Users/HP/Documents/Workspace/DevOps-Projects/Linux Shell Scripting - Advanced Application/Creating AWS Resources with Functions & Arrays using Shell Scripting (main)
$ ./aws-resources.sh local
Running script for Local Environment...

```

The terminal window has a red arrow pointing to the command `check_aws_profile || exit 1` at line 47 of the script.

## 6. Create EC2 Instances Function

Add:

```

# Function to create EC2 Instances
create_ec2_instances() {

    # Specify the parameters for the EC2 instances
    instance_type="t2.micro"
    ami_id="resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-
default-x86_64" # Latest AL2023 (dynamic)
    count=2 # Number of instances to create
    region="us-west-2" # Region to create cloud resources
}

```

```

# Create the EC2 instances
aws ec2 run-instances \
--image-id "$ami_id" \
--instance-type "$instance_type" \
--count $count \
--key-name my-new-key-pair \
--region "$region"

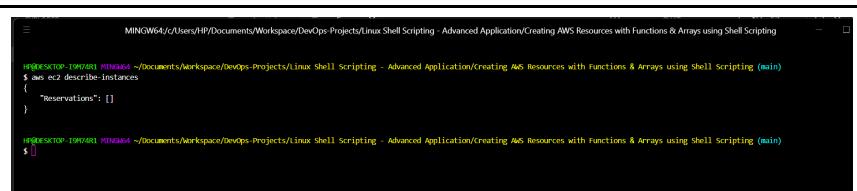
# Check if the EC2 instances were created successfully
if [ $? -eq 0 ]; then
    echo "EC2 instances created successfully."
else
    echo "Failed to create EC2 instances."
fi
}

```

- **Explanation:**

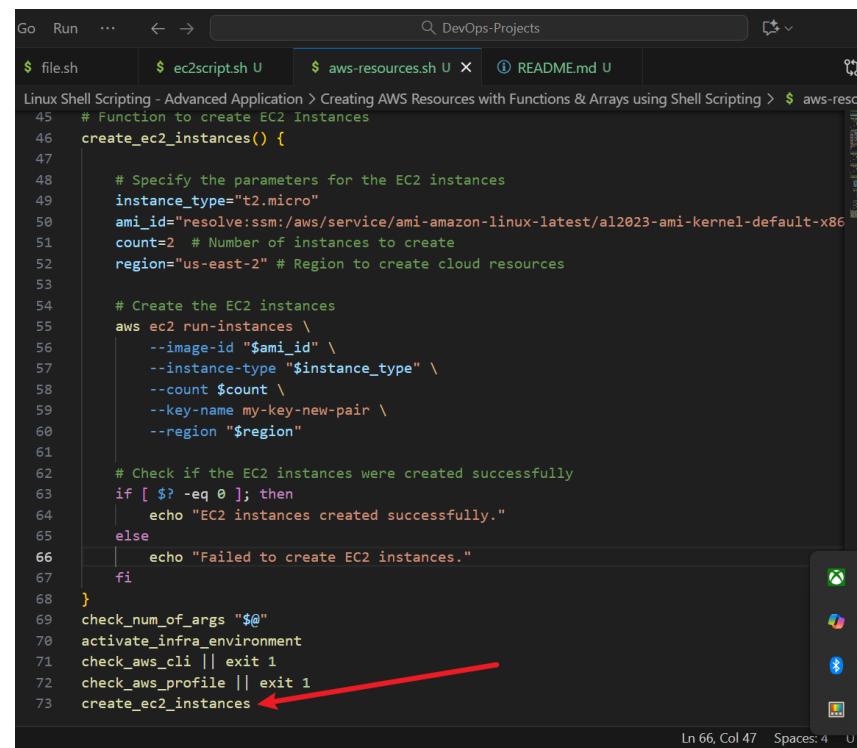
- Variables hold parameters (AMI, type, count, region) for reusability.
- `aws ec2 run-instances`: Launches instances; backslashes (\) continue the command over lines.
- `$?`: Exit status of last command (0 = success).
- Updated AMI to AL2023 for modernity; original was AL2.

- **Hands-on:** Add and call `create_ec2_instances`. Run `./aws-resources.sh local`—watch it create 2 instances. Check AWS console > EC2 > Instances for them (may take a minute). Here's what the process looks like:

Step	Action	Screenshot
1	Before execution: Note there are currently no EC2 instances	

**Step**    **Action**
**Screenshot**

- 2 Add and call  
**create\_ec2\_instances()**  
 in your script



```

$ ./file.sh
$ ec2scriptsh U
$ aws-resources.sh U
$ README.md U
Linux Shell Scripting - Advanced Application > Creating AWS Resources with Functions & Arrays using Shell Scripting > $ aws-res
45 # Function to create EC2 Instances
46 create_ec2_instances() {
47
48   # Specify the parameters for the EC2 instances
49   instance_type="t2.micro"
50   ami_id="resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64"
51   count=2 # Number of instances to create
52   region="us-east-2" # Region to create cloud resources
53
54   # Create the EC2 instances
55   aws ec2 run-instances \
56     --image-id "$ami_id" \
57     --instance-type "$instance_type" \
58     --count $count \
59     --key-name my-key-new-pair \
60     --region "$region"
61
62   # Check if the EC2 instances were created successfully
63   if [ $? -eq 0 ]; then
64     echo "EC2 instances created successfully."
65   else
66     echo "Failed to create EC2 instances."
67   fi
68 }
69 check_num_of_args "$@"
70 activate_infra_environment
71 check_aws_cli || exit 1
72 check_aws_profile || exit 1
73 create_ec2_instances
  
```

Ln 66, Col 47   Spaces: 4   U

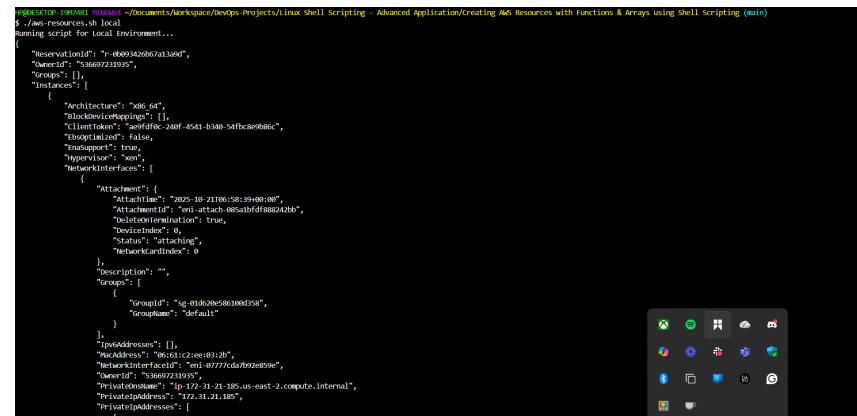
- 3 Run the script to create  
 instances and S3 buckets



```

$ ./aws-resources.sh local
Running script for local environment...
  
```

- 4 Successfully executed and  
 resources created



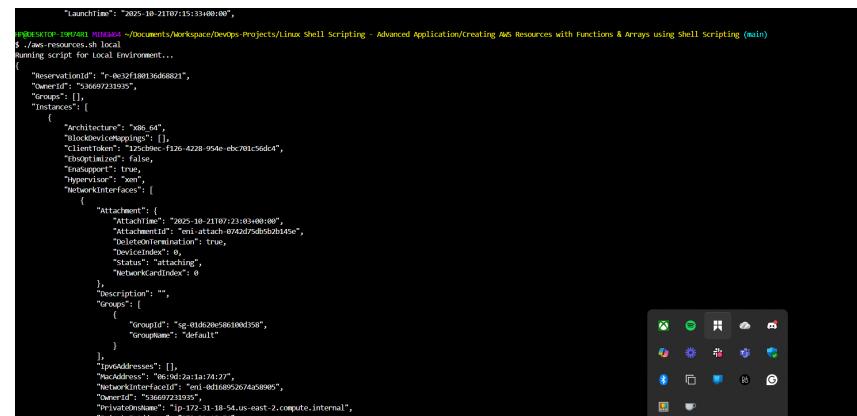
```

$ ./aws-resources.sh local
Running script for local environment...
  
```

```

{
  "ReservationId": "r-002f100136d68823",
  "OwnerId": "3669723195",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "eni-attach-0e5abfdff8882d2bb",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching",
      "NetworkInterface": {
        "Attachment": {
          "AttachmentId": "2025-10-21T07:23:39+00:00",
          "DeviceIndex": 0,
          "DeleteOnTermination": true,
          "DeviceName": "en-i0777cdaf792859e",
          "Status": "attaching",
          "NetworkInterfaceId": "eni-0777cdaf792859e"
        },
        "Description": "",
        "Groups": [
          {
            "GroupId": "sg-01d629e586100d58",
            "GroupName": "default"
          }
        ],
        "IpAddresses": [
          {
            "MacAddress": "0e:dd:c2:ee:93:3b",
            "NetworkInterfaceId": "eni-0777cdaf792859e",
            "OwnerId": "3669723195",
            "PrivateIpAddress": "ip-172-31-21-185.us-east-2.compute.internal",
            "PrivateIpAddresses": [
              "172.31.21.185"
            ]
          }
        ]
      }
    }
  ]
}
  
```

- 5 Instances created  
 successfully, visible in AWS  
 console



```

$ ./aws-resources.sh local
Running script for local environment...
  
```

```

{
  "ReservationId": "r-002f100136d68823",
  "OwnerId": "3669723195",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "eni-attach-0e5abfdff8882d2bb",
      "DeleteOnTermination": true,
      "DeviceIndex": 0,
      "Status": "attaching",
      "NetworkInterface": {
        "Attachment": {
          "AttachmentId": "2025-10-21T07:23:39+00:00",
          "DeviceIndex": 0,
          "DeleteOnTermination": true,
          "DeviceName": "en-i0777cdaf792859e",
          "Status": "attaching",
          "NetworkInterfaceId": "eni-0777cdaf792859e"
        },
        "Description": "",
        "Groups": [
          {
            "GroupId": "sg-01d629e586100d58",
            "GroupName": "default"
          }
        ],
        "IpAddresses": [
          {
            "MacAddress": "0e:dd:c2:ee:93:3b",
            "NetworkInterfaceId": "eni-0777cdaf792859e",
            "OwnerId": "3669723195",
            "PrivateIpAddress": "ip-172-31-18-54.us-east-2.compute.internal",
            "PrivateIpAddresses": [
              "172.31.18.54"
            ]
          }
        ]
      }
    }
  ]
}
  
```

## 7. Create S3 Buckets Function

Add:

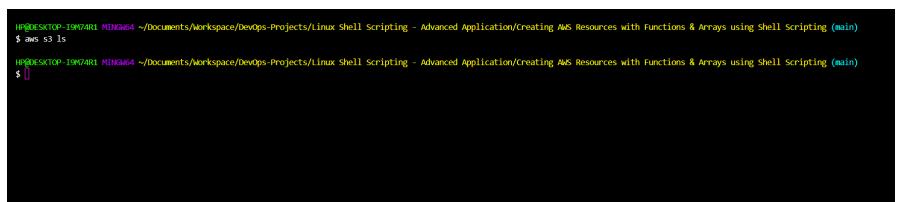
```
# Function to create S3 buckets for different departments
create_s3_buckets() {
    # Define a company name as prefix
    company="datawise"
    # Array of department names
    departments=("Marketing" "Sales" "HR" "Operations" "Media")

    # Loop through the array and create S3 buckets for each department
    for department in "${departments[@]}"; do
        bucket_name="${company}-${department}-Data-Bucket"
        # Create S3 bucket using AWS CLI
        aws s3api create-bucket --bucket "$bucket_name" --region eu-west-2
        if [ $? -eq 0 ]; then
            echo "S3 bucket '$bucket_name' created successfully."
        else
            echo "Failed to create S3 bucket '$bucket_name'."
        fi
    done
}
```

- **Explanation:**

- `company`: Prefix for unique bucket names (S3 names must be global).
- `departments=()`: Array holding multiple values.
- `for ... in "${departments[@]}"`: Loops over all elements (`[@]` expands to all).
- Constructs `bucket_name` dynamically (e.g., `datawise-Marketing-Data-Bucket`).
- `aws s3api create-bucket`: Creates bucket; checks `$?` for success.
- Arrays allow efficient handling of lists—access single elements like  `${departments[0]}` (zero-indexed).

- **Hands-on:** Add and call `create_s3_buckets`. Run the script again—see 5 buckets created. Check AWS console > S3 > Buckets. Change `company` if names conflict. Here's what the process looks like:

Step	Action	Screenshot
1	Before execution: Note there are currently no S3 buckets	

## Step Action

## Screenshot

- Add and call  
2 **create\_s3\_buckets()**  
in your script

```

Linux Shell Scripting - Advanced Application > Creating AWS Resources with Functions & Arrays using Shell Scripting > $ aws-resources.sh
78     create_s3_buckets() {
79         company="datawise364ng"
80         departments=( "marketing" "sales" "hr" "operations" "media" )
81         region="us-west-2"
82
83         for department in "${departments[@]}"; do
84             bucket_name="${company}-${department}-data-bucket"
85
86             aws s3api create-bucket \
87                 --bucket "$bucket_name" \
88                 --region "$region" \
89                 --create-bucket-configuration LocationConstraint="$region"
90
91             if [ $? -eq 0 ]; then
92                 echo "S3 bucket '$bucket_name' created successfully."
93             else
94                 echo "Failed to create S3 bucket '$bucket_name'." 
95             fi
96         done
97
98     check_num_of_args "$@"
99     activate_infra_environment
100    check_aws_cli || exit 1
101    check_aws_profile || exit 1
102    create_ec2_instances
103    create_s3_buckets
104
105 }
106
107 check_num_of_args "$@"
108 activate_infra_environment
109 check_aws_cli || exit 1
110 check_aws_profile || exit 1
111 create_ec2_instances
112 create_s3_buckets
113
114 # Environment Variables
115 ENVIRONMENT=$1
116
117 check_num_of_args "$@"
118 # Checking the environment variable
119 if [ "$#" -ne 1 ]; then
120     echo "Usage: $0 <environment>" >> error.log
121     exit 1
122 fi
123
124 activate_infra_environment
125 # Acting based on environment
126 if [ "$ENVIRONMENT" = "dev" ]; then
127     echo "Running script for local environment..."
128 elif [ "$ENVIRONMENT" = "prod" ]; then
129     echo "Running script for production..." >> error.log
130 elif [ "$ENVIRONMENT" = "test" ]; then
131     echo "Running script for test environment..." >> error.log
132 else
133     echo "Invalid environment provided." >> error.log
134     exit 1
135 fi
136
137 # Function to create S3 buckets
138 create_s3_buckets()
139 {
140     company="datawise364ng"
141     departments=( "marketing" "sales" "hr" "operations" "media" )
142     region="us-west-2"
143
144     for department in "${departments[@]}"; do
145         bucket_name="${company}-${department}-data-bucket"
146
147         aws s3api create-bucket \
148             --bucket "$bucket_name" \
149             --region "$region" \
150             --create-bucket-configuration LocationConstraint="$region"
151
152         if [ $? -eq 0 ]; then
153             echo "S3 bucket '$bucket_name' created successfully."
154         else
155             echo "Failed to create S3 bucket '$bucket_name'." 
156         fi
157     done
158
159     check_num_of_args "$@"
160     activate_infra_environment
161     check_aws_cli || exit 1
162     check_aws_profile || exit 1
163     create_ec2_instances
164     create_s3_buckets
165
166 }
167
168 check_num_of_args "$@"
169 activate_infra_environment
170 check_aws_cli || exit 1
171 check_aws_profile || exit 1
172 create_ec2_instances
173 create_s3_buckets
174
175 }
176
177 check_num_of_args "$@"
178 activate_infra_environment
179 check_aws_cli || exit 1
180 check_aws_profile || exit 1
181 create_ec2_instances
182 create_s3_buckets
183
184 }
185
186 check_num_of_args "$@"
187 activate_infra_environment
188 check_aws_cli || exit 1
189 check_aws_profile || exit 1
190 create_ec2_instances
191 create_s3_buckets
192
193 }
194
195 check_num_of_args "$@"
196 activate_infra_environment
197 check_aws_cli || exit 1
198 check_aws_profile || exit 1
199 create_ec2_instances
200 create_s3_buckets
201
202 }
203
204 check_num_of_args "$@"
205 activate_infra_environment
206 check_aws_cli || exit 1
207 check_aws_profile || exit 1
208 create_ec2_instances
209 create_s3_buckets
210
211 }
212
213 check_num_of_args "$@"
214 activate_infra_environment
215 check_aws_cli || exit 1
216 check_aws_profile || exit 1
217 create_ec2_instances
218 create_s3_buckets
219
220 }
221
222 check_num_of_args "$@"
223 activate_infra_environment
224 check_aws_cli || exit 1
225 check_aws_profile || exit 1
226 create_ec2_instances
227 create_s3_buckets
228
229 }
230
231 check_num_of_args "$@"
232 activate_infra_environment
233 check_aws_cli || exit 1
234 check_aws_profile || exit 1
235 create_ec2_instances
236 create_s3_buckets
237
238 }
239
240 check_num_of_args "$@"
241 activate_infra_environment
242 check_aws_cli || exit 1
243 check_aws_profile || exit 1
244 create_ec2_instances
245 create_s3_buckets
246
247 }
248
249 check_num_of_args "$@"
250 activate_infra_environment
251 check_aws_cli || exit 1
252 check_aws_profile || exit 1
253 create_ec2_instances
254 create_s3_buckets
255
256 }
257
258 check_num_of_args "$@"
259 activate_infra_environment
260 check_aws_cli || exit 1
261 check_aws_profile || exit 1
262 create_ec2_instances
263 create_s3_buckets
264
265 }
266
267 check_num_of_args "$@"
268 activate_infra_environment
269 check_aws_cli || exit 1
270 check_aws_profile || exit 1
271 create_ec2_instances
272 create_s3_buckets
273
274 }
275
276 check_num_of_args "$@"
277 activate_infra_environment
278 check_aws_cli || exit 1
279 check_aws_profile || exit 1
280 create_ec2_instances
281 create_s3_buckets
282
283 }
284
285 check_num_of_args "$@"
286 activate_infra_environment
287 check_aws_cli || exit 1
288 check_aws_profile || exit 1
289 create_ec2_instances
290 create_s3_buckets
291
292 }
293
294 check_num_of_args "$@"
295 activate_infra_environment
296 check_aws_cli || exit 1
297 check_aws_profile || exit 1
298 create_ec2_instances
299 create_s3_buckets
300
301 }
302
303 check_num_of_args "$@"
304 activate_infra_environment
305 check_aws_cli || exit 1
306 check_aws_profile || exit 1
307 create_ec2_instances
308 create_s3_buckets
309
310 }
311
312 check_num_of_args "$@"
313 activate_infra_environment
314 check_aws_cli || exit 1
315 check_aws_profile || exit 1
316 create_ec2_instances
317 create_s3_buckets
318
319 }
320
321 check_num_of_args "$@"
322 activate_infra_environment
323 check_aws_cli || exit 1
324 check_aws_profile || exit 1
325 create_ec2_instances
326 create_s3_buckets
327
328 }
329
330 check_num_of_args "$@"
331 activate_infra_environment
332 check_aws_cli || exit 1
333 check_aws_profile || exit 1
334 create_ec2_instances
335 create_s3_buckets
336
337 }
338
339 check_num_of_args "$@"
340 activate_infra_environment
341 check_aws_cli || exit 1
342 check_aws_profile || exit 1
343 create_ec2_instances
344 create_s3_buckets
345
346 }
347
348 check_num_of_args "$@"
349 activate_infra_environment
350 check_aws_cli || exit 1
351 check_aws_profile || exit 1
352 create_ec2_instances
353 create_s3_buckets
354
355 }
356
357 check_num_of_args "$@"
358 activate_infra_environment
359 check_aws_cli || exit 1
360 check_aws_profile || exit 1
361 create_ec2_instances
362 create_s3_buckets
363
364 }
365
366 check_num_of_args "$@"
367 activate_infra_environment
368 check_aws_cli || exit 1
369 check_aws_profile || exit 1
370 create_ec2_instances
371 create_s3_buckets
372
373 }
374
375 check_num_of_args "$@"
376 activate_infra_environment
377 check_aws_cli || exit 1
378 check_aws_profile || exit 1
379 create_ec2_instances
380 create_s3_buckets
381
382 }
383
384 check_num_of_args "$@"
385 activate_infra_environment
386 check_aws_cli || exit 1
387 check_aws_profile || exit 1
388 create_ec2_instances
389 create_s3_buckets
390
391 }
392
393 check_num_of_args "$@"
394 activate_infra_environment
395 check_aws_cli || exit 1
396 check_aws_profile || exit 1
397 create_ec2_instances
398 create_s3_buckets
399
400 }
401
402 check_num_of_args "$@"
403 activate_infra_environment
404 check_aws_cli || exit 1
405 check_aws_profile || exit 1
406 create_ec2_instances
407 create_s3_buckets
408
409 }
410
411 check_num_of_args "$@"
412 activate_infra_environment
413 check_aws_cli || exit 1
414 check_aws_profile || exit 1
415 create_ec2_instances
416 create_s3_buckets
417
418 }
419
420 check_num_of_args "$@"
421 activate_infra_environment
422 check_aws_cli || exit 1
423 check_aws_profile || exit 1
424 create_ec2_instances
425 create_s3_buckets
426
427 }
428
429 check_num_of_args "$@"
430 activate_infra_environment
431 check_aws_cli || exit 1
432 check_aws_profile || exit 1
433 create_ec2_instances
434 create_s3_buckets
435
436 }
437
438 check_num_of_args "$@"
439 activate_infra_environment
440 check_aws_cli || exit 1
441 check_aws_profile || exit 1
442 create_ec2_instances
443 create_s3_buckets
444
445 }
446
447 check_num_of_args "$@"
448 activate_infra_environment
449 check_aws_cli || exit 1
450 check_aws_profile || exit 1
451 create_ec2_instances
452 create_s3_buckets
453
454 }
455
456 check_num_of_args "$@"
457 activate_infra_environment
458 check_aws_cli || exit 1
459 check_aws_profile || exit 1
460 create_ec2_instances
461 create_s3_buckets
462
463 }
464
465 check_num_of_args "$@"
466 activate_infra_environment
467 check_aws_cli || exit 1
468 check_aws_profile || exit 1
469 create_ec2_instances
470 create_s3_buckets
471
472 }
473
474 check_num_of_args "$@"
475 activate_infra_environment
476 check_aws_cli || exit 1
477 check_aws_profile || exit 1
478 create_ec2_instances
479 create_s3_buckets
480
481 }
482
483 check_num_of_args "$@"
484 activate_infra_environment
485 check_aws_cli || exit 1
486 check_aws_profile || exit 1
487 create_ec2_instances
488 create_s3_buckets
489
490 }
491
492 check_num_of_args "$@"
493 activate_infra_environment
494 check_aws_cli || exit 1
495 check_aws_profile || exit 1
496 create_ec2_instances
497 create_s3_buckets
498
499 }
500
501 check_num_of_args "$@"
502 activate_infra_environment
503 check_aws_cli || exit 1
504 check_aws_profile || exit 1
505 create_ec2_instances
506 create_s3_buckets
507
508 }
509
510 check_num_of_args "$@"
511 activate_infra_environment
512 check_aws_cli || exit 1
513 check_aws_profile || exit 1
514 create_ec2_instances
515 create_s3_buckets
516
517 }
518
519 check_num_of_args "$@"
520 activate_infra_environment
521 check_aws_cli || exit 1
522 check_aws_profile || exit 1
523 create_ec2_instances
524 create_s3_buckets
525
526 }
527
528 check_num_of_args "$@"
529 activate_infra_environment
530 check_aws_cli || exit 1
531 check_aws_profile || exit 1
532 create_ec2_instances
533 create_s3_buckets
534
535 }
536
537 check_num_of_args "$@"
538 activate_infra_environment
539 check_aws_cli || exit 1
540 check_aws_profile || exit 1
541 create_ec2_instances
542 create_s3_buckets
543
544 }
545
546 check_num_of_args "$@"
547 activate_infra_environment
548 check_aws_cli || exit 1
549 check_aws_profile || exit 1
550 create_ec2_instances
551 create_s3_buckets
552
553 }
554
555 check_num_of_args "$@"
556 activate_infra_environment
557 check_aws_cli || exit 1
558 check_aws_profile || exit 1
559 create_ec2_instances
560 create_s3_buckets
561
562 }
563
564 check_num_of_args "$@"
565 activate_infra_environment
566 check_aws_cli || exit 1
567 check_aws_profile || exit 1
568 create_ec2_instances
569 create_s3_buckets
570
571 }
572
573 check_num_of_args "$@"
574 activate_infra_environment
575 check_aws_cli || exit 1
576 check_aws_profile || exit 1
577 create_ec2_instances
578 create_s3_buckets
579
580 }
581
582 check_num_of_args "$@"
583 activate_infra_environment
584 check_aws_cli || exit 1
585 check_aws_profile || exit 1
586 create_ec2_instances
587 create_s3_buckets
588
589 }
590
591 check_num_of_args "$@"
592 activate_infra_environment
593 check_aws_cli || exit 1
594 check_aws_profile || exit 1
595 create_ec2_instances
596 create_s3_buckets
597
598 }
599
600 check_num_of_args "$@"
601 activate_infra_environment
602 check_aws_cli || exit 1
603 check_aws_profile || exit 1
604 create_ec2_instances
605 create_s3_buckets
606
607 }
608
609 check_num_of_args "$@"
610 activate_infra_environment
611 check_aws_cli || exit 1
612 check_aws_profile || exit 1
613 create_ec2_instances
614 create_s3_buckets
615
616 }
617
618 check_num_of_args "$@"
619 activate_infra_environment
620 check_aws_cli || exit 1
621 check_aws_profile || exit 1
622 create_ec2_instances
623 create_s3_buckets
624
625 }
626
627 check_num_of_args "$@"
628 activate_infra_environment
629 check_aws_cli || exit 1
630 check_aws_profile || exit 1
631 create_ec2_instances
632 create_s3_buckets
633
634 }
635
636 check_num_of_args "$@"
637 activate_infra_environment
638 check_aws_cli || exit 1
639 check_aws_profile || exit 1
640 create_ec2_instances
641 create_s3_buckets
642
643 }
644
645 check_num_of_args "$@"
646 activate_infra_environment
647 check_aws_cli || exit 1
648 check_aws_profile || exit 1
649 create_ec2_instances
650 create_s3_buckets
651
652 }
653
654 check_num_of_args "$@"
655 activate_infra_environment
656 check_aws_cli || exit 1
657 check_aws_profile || exit 1
658 create_ec2_instances
659 create_s3_buckets
660
661 }
662
663 check_num_of_args "$@"
664 activate_infra_environment
665 check_aws_cli || exit 1
666 check_aws_profile || exit 1
667 create_ec2_instances
668 create_s3_buckets
669
670 }
671
672 check_num_of_args "$@"
673 activate_infra_environment
674 check_aws_cli || exit 1
675 check_aws_profile || exit 1
676 create_ec2_instances
677 create_s3_buckets
678
679 }
680
681 check_num_of_args "$@"
682 activate_infra_environment
683 check_aws_cli || exit 1
684 check_aws_profile || exit 1
685 create_ec2_instances
686 create_s3_buckets
687
688 }
689
690 check_num_of_args "$@"
691 activate_infra_environment
692 check_aws_cli || exit 1
693 check_aws_profile || exit 1
694 create_ec2_instances
695 create_s3_buckets
696
697 }
698
699 check_num_of_args "$@"
700 activate_infra_environment
701 check_aws_cli || exit 1
702 check_aws_profile || exit 1
703 create_ec2_instances
704 create_s3_buckets
705
706 }
707
708 check_num_of_args "$@"
709 activate_infra_environment
710 check_aws_cli || exit 1
711 check_aws_profile || exit 1
712 create_ec2_instances
713 create_s3_buckets
714
715 }
716
717 check_num_of_args "$@"
718 activate_infra_environment
719 check_aws_cli || exit 1
720 check_aws_profile || exit 1
721 create_ec2_instances
722 create_s3_buckets
723
724 }
725
726 check_num_of_args "$@"
727 activate_infra_environment
728 check_aws_cli || exit 1
729 check_aws_profile || exit 1
730 create_ec2_instances
731 create_s3_buckets
732
733 }
734
735 check_num_of_args "$@"
736 activate_infra_environment
737 check_aws_cli || exit 1
738 check_aws_profile || exit 1
739 create_ec2_instances
740 create_s3_buckets
741
742 }
743
744 check_num_of_args "$@"
745 activate_infra_environment
746 check_aws_cli || exit 1
747 check_aws_profile || exit 1
748 create_ec2_instances
749 create_s3_buckets
750
751 }
752
753 check_num_of_args "$@"
754 activate_infra_environment
755 check_aws_cli || exit 1
756 check_aws_profile || exit 1
757 create_ec2_instances
758 create_s3_buckets
759
760 }
761
762 check_num_of_args "$@"
763 activate_infra_environment
764 check_aws_cli || exit 1
765 check_aws_profile || exit 1
766 create_ec2_instances
767 create_s3_buckets
768
769 }
770
771 check_num_of_args "$@"
772 activate_infra_environment
773 check_aws_cli || exit 1
774 check_aws_profile || exit 1
775 create_ec2_instances
776 create_s3_buckets
777
778 }
779
780 check_num_of_args "$@"
781 activate_infra_environment
782 check_aws_cli || exit 1
783 check_aws_profile || exit 1
784 create_ec2_instances
785 create_s3_buckets
786
787 }
788
789 check_num_of_args "$@"
790 activate_infra_environment
791 check_aws_cli || exit 1
792 check_aws_profile || exit 1
793 create_ec2_instances
794 create_s3_buckets
795
796 }
797
798 check_num_of_args "$@"
799 activate_infra_environment
800 check_aws_cli || exit 1
801 check_aws_profile || exit 1
802 create_ec2_instances
803 create_s3_buckets
804
805 }
806
807 check_num_of_args "$@"
808 activate_infra_environment
809 check_aws_cli || exit 1
810 check_aws_profile || exit 1
811 create_ec2_instances
812 create_s3_buckets
813
814 }
815
816 check_num_of_args "$@"
817 activate_infra_environment
818 check_aws_cli || exit 1
819 check_aws_profile || exit 1
820 create_ec2_instances
821 create_s3_buckets
822
823 }
824
825 check_num_of_args "$@"
826 activate_infra_environment
827 check_aws_cli || exit 1
828 check_aws_profile || exit 1
829 create_ec2_instances
830 create_s3_buckets
831
832 }
833
834 check_num_of_args "$@"
835 activate_infra_environment
836 check_aws_cli || exit 1
837 check_aws_profile || exit 1
838 create_ec2_instances
839 create_s3_buckets
840
841 }
842
843 check_num_of_args "$@"
844 activate_infra_environment
845 check_aws_cli || exit 1
846 check_aws_profile || exit 1
847 create_ec2_instances
848 create_s3_buckets
849
850 }
851
852 check_num_of_args "$@"
853 activate_infra_environment
854 check_aws_cli || exit 1
855 check_aws_profile || exit 1
856 create_ec2_instances
857 create_s3_buckets
858
859 }
860
861 check_num_of_args "$@"
862 activate_infra_environment
863 check_aws_cli || exit 1
864 check_aws_profile || exit 1
865 create_ec2_instances
866 create_s3_buckets
867
868 }
869
870 check_num_of_args "$@"
871 activate_infra_environment
872 check_aws_cli || exit 1
873 check_aws_profile || exit 1
874 create_ec2_instances
875 create_s3_buckets
876
877 }
878
879 check_num_of_args "$@"
880 activate_infra_environment
881 check_aws_cli || exit 1
882 check_aws_profile || exit 1
883 create_ec2_instances
884 create_s3_buckets
885
886 }
887
888 check_num_of_args "$@"
889 activate_infra_environment
890 check_aws_cli || exit 1
891 check_aws_profile || exit 1
892 create_ec2_instances
893 create_s3_buckets
894
895 }
896
897 check_num_of_args "$@"
898 activate_infra_environment
899 check_aws_cli || exit 1
900 check_aws_profile || exit 1
901 create_ec2_instances
902 create_s3_buckets
903
904 }
905
906 check_num_of_args "$@"
907 activate_infra_environment
908 check_aws_cli || exit 1
909 check_aws_profile || exit 1
910 create_ec2_instances
911 create_s3_buckets
912
913 }
914
915 check_num_of_args "$@"
916 activate_infra_environment
917 check_aws_cli || exit 1
918 check_aws_profile || exit 1
919 create_ec2_instances
920 create_s3_buckets
921
922 }
923
924 check_num_of_args "$@"
925 activate_infra_environment
926 check_aws_cli || exit 1
927 check_aws_profile || exit 1
928 create_ec2_instances
929 create_s3_buckets
930
931 }
932
933 check_num_of_args "$@"
934 activate_infra_environment
935 check_aws_cli || exit 1
936 check_aws_profile || exit 1
937 create_ec2_instances
938 create_s3_buckets
939
940 }
941
942 check_num_of_args "$@"
943 activate_infra_environment
944 check_aws_cli || exit 1
945 check_aws_profile || exit 1
946 create_ec2_instances
947 create_s3_buckets
948
949 }
950
951 check_num_of_args "$@"
952 activate_infra_environment
953 check_aws_cli || exit 1
954 check_aws_profile || exit 1
955 create_ec2_instances
956 create_s3_buckets
957
958 }
959
960 check_num_of_args "$@"
961 activate_infra_environment
962 check_aws_cli || exit 1
963 check_aws_profile || exit 1
964 create_ec2_instances
965 create_s3_buckets
966
967 }
968
969 check_num_of_args "$@"
970 activate_infra_environment
971 check_aws_cli || exit 1
972 check_aws_profile || exit 1
973 create_ec2_instances
974 create_s3_buckets
975
976 }
977
978 check_num_of_args "$@"
979 activate_infra_environment
980 check_aws_cli || exit 1
981 check_aws_profile || exit 1
982 create_ec2_instances
983 create_s3_buckets
984
985 }
986
987 check_num_of_args "$@"
988 activate_infra_environment
989 check_aws_cli || exit 1
990 check_aws_profile || exit 1
991 create_ec2_instances
992 create_s3_buckets
993
994 }
995
996 check_num_of_args "$@"
997 activate_infra_environment
998 check_aws_cli || exit 1
999 check_aws_profile || exit 1
1000 create_ec2_instances
1001 create_s3_buckets
1002
1003 }
1004
1005 check_num_of_args "$@"
1006 activate_infra_environment
1007 check_aws_cli || exit 1
1008 check_aws_profile || exit 1
1009 create_ec2_instances
1010 create_s3_buckets
1011
1012 }
1013
1014 check_num_of_args "$@"
1015 activate_infra_environment
1016 check_aws_cli || exit 1
1017 check_aws_profile || exit 1
1018 create_ec2_instances
1019 create_s3_buckets
1020
1021 }
1022
1023 check_num_of_args "$@"
1024 activate_infra_environment
1025 check_aws_cli || exit 1
1026 check_aws_profile || exit 1
1027 create_ec2_instances
1028 create_s3_buckets
1029
1030 }
1031
1032 check_num_of_args "$@"
1033 activate_infra_environment
1034 check_aws_cli || exit 1
1035 check_aws_profile || exit 1
1036 create_ec2_instances
1037 create_s3_buckets
1038
1039 }
1040
1041 check_num_of_args "$@"
1042 activate_infra_environment
1043 check_aws_cli || exit 1
1044 check_aws_profile || exit 1
1045 create_ec2_instances
1046 create_s3_buckets
1047
1048 }
1049
1050 check_num_of_args "$@"
1051 activate_infra_environment
1052 check_aws_cli || exit 1
1053 check_aws_profile || exit 1
1054 create_ec2_instances
1055 create_s3_buckets
1056
1057 }
1058
1059 check_num_of_args "$@"
1060 activate_infra_environment
1061 check_aws_cli || exit 1
1062 check_aws_profile || exit 1
1063 create_ec2_instances
1064 create_s3_buckets
1065
1066 }
1067
1068 check_num_of_args "$@"
1069 activate_infra_environment
1070 check_aws_cli || exit 1
1071 check_aws_profile || exit 1
1072 create_ec2_instances
1073 create_s3_buckets
1074
1075 }
1076
1077 check_num_of_args "$@"
1078 activate_infra_environment
1079 check_aws_cli || exit 1
1080 check_aws_profile || exit 1
1081 create_ec2_instances
1082 create_s3_buckets
1083
1084 }
1085
1086 check_num_of_args "$@"
1087 activate_infra_environment
1088 check_aws_cli || exit 1
1089 check_aws_profile || exit 1
1090 create_ec2_instances
1091 create_s3_buckets
1092
1093 }
1094
1095 check_num_of_args "$@"
1096 activate_infra_environment
1097 check_aws_cli || exit 1
1098 check_aws_profile || exit 1
1099 create_ec2_instances
1100 create_s3_buckets
1101
1102 }
1103
1104 check_num_of_args "$@"
1105 activate_infra_environment
1106 check_aws_cli || exit 1
1107 check_aws_profile || exit 1
1108 create_ec2_instances
1109 create_s3_buckets
1110
1111 }
1112
1113 check_num_of_args "$@"
1114 activate_infra_environment
1115 check_aws_cli || exit 1
1116 check_aws_profile || exit 1
1117 create_ec2_instances
1118 create_s3_buckets
1119
1120 }
1121
1122 check_num_of_args "$@"
1123 activate_infra_environment
1124 check_aws_cli || exit 1
1125 check_aws_profile || exit 1
1126 create_ec2_instances
1127 create_s3_buckets
1128
1129 }
1130
1131 check_num_of_args "$@"
1132 activate_infra_environment
1133 check_aws_cli || exit 1
1134 check_aws_profile || exit 1
1135 create_ec2_instances
1136 create_s3_buckets
1137
1138 }
1139
1140 check_num_of_args "$@"
1141 activate_infra_environment
1142 check_aws_cli || exit 1
1143 check_aws_profile || exit 1
1144 create_ec2_instances
1145 create_s3_buckets
1146
1147 }
1148
1149 check_num_of_args "$@"
1150 activate_infra_environment
1151 check_aws_cli || exit 1
1152 check_aws_profile || exit 1
1153 create_ec2_instances
1154 create_s3_buckets
1155
1156 }
1157
1158 check_num_of_args "$@"
1159 activate_infra_environment
1160 check_aws_cli || exit 1
1161 check_aws_profile || exit 1
1162 create_ec2_instances
1163 create_s3_buckets
1164
1165 }
1166
1167 check_num_of_args "$@"
1168 activate_infra_environment
1169 check_aws_cli || exit 1
1170 check_aws_profile || exit 1
1171 create_ec2_instances
1172 create_s3_buckets
1173
1174 }
1175
1176 check_num_of_args "$@"
1177 activate_infra_environment
1178 check_aws_cli || exit 1
1179 check_aws_profile || exit 1
1180 create_ec2_instances
1181 create_s3_buckets
1182
1183 }
1184
1185 check_num_of_args "$@"
1186 activate_infra_environment
1187 check_aws_cli || exit 1
1188 check_aws_profile || exit 1
1189 create_ec2_instances
1190 create_s3_buckets
1191
1192 }
1193
1194 check_num_of_args "$@"
1195 activate_infra_environment
1196 check_aws_cli || exit 1
1197 check_aws_profile || exit 1
1198 create_ec2_instances
1199 create_s3_buckets
1200
1201 }
1202
1203 check_num_of_args "$@"
1204 activate_infra_environment
1205 check_aws_cli || exit 1
1206 check_aws_profile || exit 1
1207 create_ec2_instances
1208 create_s3_buckets
1209
1210 }
1211
1212 check_num_of_args "$@"
1213 activate_infra_environment
1214 check_aws_cli || exit 1
1215 check_aws_profile || exit 1
1216 create_ec2_instances
1217 create_s3_buckets
1218
1219 }
1220
1221 check_num_of_args "$@"
1222 activate_infra_environment
1223 check_aws_cli || exit 1
1224 check_aws_profile || exit 1
1225 create_ec2_instances
1226 create_s3_buckets
1227
1228 }
1229
1230 check_num_of_args "$@"
1231 activate_infra_environment
1232 check_aws_cli || exit 1
1233 check_aws_profile || exit 1
1234 create_ec2_instances
1235 create_s3_buckets
1236
1237 }
1238
1239 check_num_of_args "$@"
1240 activate_infra_environment
1241 check_aws_cli || exit 1
1242 check_aws_profile || exit 1
1243 create_ec2_instances
1244 create_s3_buckets
1245
1246 }
1247
1248 check_num_of_args "$@"
1249 activate_infra_environment
1250 check_aws_cli || exit 1
1251 check_aws_profile || exit 1
1252 create_ec2_instances
1253 create_s3_buckets
1254
1255 }
1256
1257 check_num_of_args "$@"
1258 activate_infra_environment
1259 check_aws_cli || exit 1
1260 check_aws_profile || exit 1
1261 create_ec2_instances
1262 create_s3_buckets
1263
1264 }
1265
1266 check_num_of_args "$@"
1267 activate_infra_environment
1268 check_aws_cli || exit 1
1269 check_aws_profile || exit 1
1270 create_ec2_instances
1271 create_s3_buckets
1272
1273 }
1274
1275 check_num_of_args "$@"
1276 activate_infra_environment
1277 check_aws_cli || exit 1
1278 check_aws_profile || exit 1
1279 create_ec2_instances
1280 create_s3_buckets
1281
1282 }
1283
1284 check_num_of_args "$@"
1285 activate_infra_environment
1286 check_aws_cli || exit 1
1287 check_aws_profile || exit 1
1288 create_ec2_instances
1289 create_s3_buckets
1290
1291 }
1292
1293 check_num_of_args "$@"
1294 activate_infra_environment
1295 check_aws_cli || exit 1
1296 check_aws_profile || exit 1
1297 create_ec2_instances
1298 create_s3_buckets
1299
1300 }
1301
1302 check_num_of_args "$@"
1303 activate_infra_environment
1304 check_aws_cli || exit 1
1305 check_aws_profile || exit 1
1306 create_ec2_instances
1307 create_s3_buckets
1308
1309 }
1310
1311 check_num_of_args "$@"
1312 activate_infra_environment
1313 check_aws_cli || exit 1
1314 check_aws_profile || exit 1
1315 create_ec2_instances
1316 create_s3_buckets
1317
1318 }
1319
1320 check_num_of_args "$@"
1321 activate_infra_environment
1322 check_aws_cli || exit 1
1323 check_aws_profile || exit 1
1324 create_ec2_instances
1325 create_s3_buckets
1326
1327 }
1328
1329 check_num_of_args "$@"
1330 activate_infra_environment
1331 check_aws_cli || exit 1
1332 check_aws_profile || exit 1
1333 create_ec2_instances
1334 create_s3_buckets
1335
1336 }
1337
1338 check_num_of_args "$@"
1339 activate_infra_environment
1340 check_aws_cli || exit 1
1341 check_aws_profile || exit 1
1342 create_ec2_instances
1343 create_s3_buckets
1344
1345 }
1346
1347 check_num_of_args "$@"
1348 activate_infra_environment
1349 check_aws_cli || exit 1
1350 check_aws_profile || exit 1
1351 create_ec2_instances
1352 create_s3_buckets
1353
1354 }
1355
1356 check_num_of_args "$@"
1357 activate_infra_environment
1358 check_aws_cli || exit 1
1359 check_aws_profile || exit 1
1360 create_ec2_instances
1361 create_s3_buckets
1362
1363 }
1364
1365 check_num_of_args "$@"
1366 activate_infra_environment
1367 check_aws_cli || exit 1
1368 check_aws_profile || exit 1
1369 create_ec2_instances
1370 create_s3_buckets
1371
1372 }
1373
1374 check_num_of_args "$@"
1375 activate_infra_environment
1376 check_aws_cli || exit 1
1377 check_aws_profile || exit 1
1378 create_ec2_instances
1379 create_s3_buckets
1380
1381 }
1382
1383 check_num_of_args "$@"
1384 activate_infra_environment
1385 check_aws_cli || exit 1
1386 check_aws_profile || exit 1
1387 create_ec2_instances
1388 create_s3_buckets
1389
1390 }
1391
1392 check_num_of_args "$@"
1393 activate_infra_environment
1394 check_aws_cli || exit 1
1395 check_aws_profile || exit 1
1396 create_ec2_instances
1397 create_s3_buckets
1398
1399 }
1400
1401 check_num_of_args "$@"
1402 activate_infra_environment
1403 check_aws_cli || exit 1
1404 check_aws_profile || exit 1
1405 create_ec2_instances
1406 create_s3_buckets
1407
1408 }
1409
1410 check_num_of_args "$@"
1411 activate_infra_environment
1412 check_aws_cli || exit 1
1413 check_aws_profile || exit 1
1414 create_ec2_instances
1415 create_s3_buckets
14
```

```
check_aws_profile || exit 1
create_ec2_instances
create_s3_buckets
```

- **Explanation:** Executes functions in order; `|| exit 1` stops on failure.
- **Hands-on:** Full run now deploys everything.

## Running the Script

Save the complete script, then `./aws-resources.sh local`. Output shows success/failure. Monitor costs in AWS Billing.

## Verification and Cleanup

- **Verify:** Use console as above, or CLI: `aws ec2 describe-instances --region eu-west-2, aws s3 ls`.
- **Cleanup** (important to avoid charges):
  - EC2: `aws ec2 terminate-instances --instance-ids <id1> <id2> --region eu-west-2` (get IDs from `describe-instances`).
  - S3: `aws s3api delete-bucket --bucket <name> --region eu-west-2` for each.
- Hands-on: Run cleanup after testing.

## Summary

In this mini project on creating AWS resources with functions and introducing arrays, I learned how to automate infrastructure provisioning using **Bash scripting** and the **AWS CLI**, beginning with setting up prerequisites like installing the AWS CLI, configuring profiles, and creating EC2 key pairs in the console.

I explored the AWS documentation to understand commands such as `aws ec2 run-instances` for launching `t2.micro` instances with parameters like **AMI ID** (updated to dynamic SSM for the latest Amazon Linux 2023 in `eu-west-2`), **instance type, count, key name**, and **region**, while using exit status (`$?`) for success checks and environment variables for flexibility.

Additionally, I defined functions to modularize code, including one for EC2 creation and another for S3 buckets, where I introduced **arrays** to store department names (`Marketing, Sales, HR, Operations, Media`) and looped through them with `"${departments[@]}"` to generate globally unique bucket names via a prefix like `datawise` and suffix `-Data-Bucket`, creating them with `aws s3api create-bucket` and handling errors.

The script also incorporates argument validation for environments (`local, testing, production`), checks for AWS CLI and profile setup, and demonstrates key Bash concepts like **special variables, loops**, and **conditional statements**, ultimately enabling efficient, repeatable automation of cloud resources while emphasizing **global uniqueness for S3** and **best practices for secure, up-to-date AMIs**.