Name – KUNDETI NAGA ARAVIND
Reg No: – RA2011028010067

Install and Practice Using the AWS CLI

Aim: To Install and learn using AWS CLI

Procedure:-
1. Open the following link Installing or updating the latest version of the AWS CLI - AWS Command Line Interface
2. There will be 3 options to choose for the operating system: expand the section for your operating system. And proceed accordingly.
3. Download and run the AWS CLI MSI installer for Windows (64-bit):
4. https://awscli.amazonaws.com/AWSCLIV2.msi
5. To confirm the installation, open the Start menu, search for cmd to open a command prompt window, and at the command prompt use the aws --version command.
6. Sign in to the IAM console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.
Note
We strongly recommend that you adhere to the best practice of using the Administrator IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
7. In the navigation pane, choose Users and then choose Add users.
8. For User name, enter Administrator.
9. Select the check box next to AWS Management Console access. Then select Custom password, and then enter your new password in the text box.
10. By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.
11. Choose Next: Permissions.

12. Under Set permissions, choose Add user to group.
13. Choose Create group.

14. In the Create group dialog box, for Group name enter Administrators.
15. Choose Filter policies, and then select AWS managed - job function to filter the table contents.
16. In the policy list, select the check box for AdministratorAccess. Then choose Create group.
17. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
18. In the navigation pane, choose Users.
19. Choose the name of the user whose access keys you want to create, and then choose the Security credentials tab.
20. In the Access keys section, choose Create access key.
21. To view the new access key pair, choose Show. You will not have access to the secret access key again after this dialog box closes. Your credentials will look something like this:
a. Access key ID: AKIAIOSFODNN7EXAMPLE
b. Secret access key:
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
22. To download the key pair, choose Download .csv file. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes.
Keep the keys confidential in order to protect your AWS account and never email them. Do not share them outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.
23. After you download the .csv file, choose Close. When you create an access key, the key pair is active by default, and you can use the pair right away.

Install or update the AWS CLI

To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the AWS CLI changelog on *GitHub*.

1. Download and run the AWS CLI MSI installer for Windows (64-bit):

   https://awscli.amazonaws.com/AWSCLIV2.msi

   Alternatively, you can run the `msiexec` command to run the MSI installer.

   ```
   C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
   ```

   For various parameters that can be used with `msiexec`, see msiexec on the *Microsoft Docs* website.

2. To confirm the installation, open the **Start** menu, search for `cmd` to open a command prompt window, and at the command prompt use the `aws --version` command.

   ```
   C:\> aws --version
   aws-cli/2.4.5 Python/3.8.8 Windows/10 exe/AMD64 prompt/off
   ```

   If Windows is unable to find the program, you might need to close and reopen the command prompt window to refresh the path, or follow the troubleshooting in Troubleshooting AWS CLI errors.

1] AWS Help

The built-in AWS CLI help command. You can get help with any command when using the AWS Command Line Interface (AWS CLI). To do so, simply type help at the end of a command name. For example, the following command displays help for the general AWS CLI options and the available top-level commands.

```
C:\Users\Aravind>aws help

aws
^^^


Description
***********

The AWS Command Line Interface is a unified tool to manage your AWS
services.


Synopsis
********

   aws [options] <command> <subcommand> [parameters]

Use *aws command help* for information on a specific command. Use *aws
help topics* to view a list of available help topics. The synopsis for
each command shows its parameters and their usage. Optional parameters
are shown in square brackets.


Options
*******

"--debug" (boolean)

Turn on debug logging.

"--endpoint-url" (string)
-- More  --
```

2] AWS – version

The AWS CLI version 2 is the most recent major version of the AWS CLI and supports all of the latest features

```
C:\Users\Aravind>aws --version
aws-cli/2.7.25 Python/3.9.11 Windows/10 exe/AMD64 prompt/off
```

3] AWS configure

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.

```
C:\Users\Aravind>aws configure
AWS Access Key ID [****************UMOO]: AKIAXWHKMKIV3K2OUMOO
AWS Secret Access Key [****************Gdif]: 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
Default region name [Asia Pacific (Mumbai)]: ap-south-1
Default output format [None]:
```

4] sts get-caller-identity

To get your account id using AWS CLI, run the sts get-caller-identity command, setting the --query parameter to Account to filter the output. Copied! The get-caller-identity command returns the User Id, Account Id, and the ARN of the caller

```
C:\Users\Aravind>aws sts get-caller-identity
{
    "UserId": "528772518443",
    "Account": "528772518443",
    "Arn": "arn:aws:iam::528772518443:root"
}
```

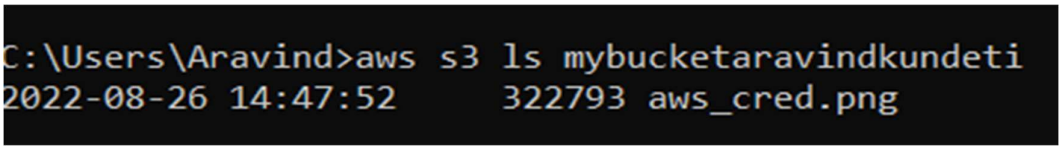
5] aws s3 ls

To list your buckets, folders, or objects, use the s3 ls command. Using the command without a target or options lists all buckets.

```
C:\Users\Aravind>aws s3 ls
2022-08-26 14:46:52 mybucketaravindkundeti
```

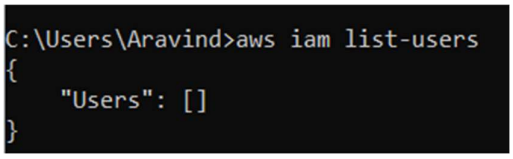
6] aws s3 ls bucketName

The following ls command lists objects and common prefixes under a specified bucket and prefix. In this example, the user owns the bucket mybucket with the objects test.txt and somePrefix/test.txt. The LastWriteTime and Length are arbitrary. Note that since the ls command has no interaction with the local filesystem, the s3:// URI scheme is not required to resolve ambiguity and may be omitted



7] aws iam list users

To list the users present in an account, use the command to get information regarding them. It will show the list of users along with their name and id.



8] aws iam list-policies

To list the policies of aws, this command is used to get the policies present in the aws account and this is used to give permissions to the newly created users.

```
C:\Users\Aravind>aws iam list-policies
{
    "Policies": [
        {
            "PolicyName": "AWSDirectConnectReadOnlyAccess",
            "PolicyId": "ANPAI23HZ27SI6FQMGNQ2",
            "Arn": "arn:aws:iam::aws:policy/AWSDirectConnectReadOnlyAccess",
            "Path": "/",
            "DefaultVersionId": "v4",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:40:08+00:00",
            "UpdateDate": "2020-05-18T18:48:22+00:00"
        },
        {
            "PolicyName": "AmazonGlacierReadOnlyAccess",
            "PolicyId": "ANPAI2D5NJKMU274MET4E",
            "Arn": "arn:aws:iam::aws:policy/AmazonGlacierReadOnlyAccess",
            "Path": "/",
            "DefaultVersionId": "v2",
            "AttachmentCount": 0,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:40:27+00:00",
            "UpdateDate": "2016-05-05T18:46:10+00:00"
        },
        {
            "PolicyName": "AWSMarketplaceFullAccess",
            "PolicyId": "ANPAI2DV5ULJSO2FYVPYG",
```

9] delete bucket

If your bucket does not have versioning enabled, you can use the rb (remove bucket) AWS CLI command with the --force parameter to delete the bucket and all the objects in it. This command deletes all objects first and then deletes the bucket.

```
C:\Users\Aravind>aws s3 rb s3://mybucketaravindkundeti --force
delete: s3://mybucketaravindkundeti/aws_cred.png
remove_bucket: mybucketaravindkundeti
```

10] remove file from bucket

To delete objects in a bucket or your local directory, use the s3 rm command. For a few common options to use with this command, and examples, see Frequently used options for s3 commands. For a complete list of options, see s3 rm in the AWS CLI Command Reference. The following example deletes filename

```
C:\Users\Aravind>aws s3 rm s3://my-new-test-bucketaravind --recursive
```

KUNDETI NAGA ARAVIND

RA2011028010067