NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Cloud Computing Developer Tools

Cloud Infrastructure Engineering

Nanyang Technological University & Skills Union - 2022/2023

Course Content

- Quick Check-In
- Dive into the basics of AWS Command Line Interface
- Explore the usage of AWS CLI

Time	What	How or Why
7:15pm - 7:45pm	Part 1 - Presentation	Introduction To AWS CLI
7:45pm - 8:00pm	Part 2 - Presentation & Hands-on Activity	Setting up AWS CLI
8:00pm - 8:10pm	Break	
8:10pm - 8:50pm	Activity	Hands-on AWS CLI
8:50pm - 9:00pm	Break	Explore AWS Products & Console
9:00pm - 10:00pm	Assignment & Wrap Up	

Setting Up AWS CLI

What is AWS CLI

The AWS Command Line Interface (AWS CLI) is an open source tool that **enables you to interact with AWS services using commands** in your command-line shell.

With minimal configuration, the AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your terminal program:

Communicating With AWS

Linux shells – Use common shell programs such as bash, zsh, and tcsh to run commands in Linux or macOS.

Windows command line – On Windows, run commands at the Windows command prompt or in PowerShell.

Remotely – Run commands on Amazon Elastic Compute Cloud (Amazon EC2) instances through a remote terminal program such as PuTTY or SSH, or with AWS Systems Manager.

AWS CLI

The AWS CLI provides **direct access to the public APIs of AWS services**.

You can explore a service's capabilities with the AWS CLI, and develop shell scripts to manage your resources.

Let’s Begin

Pre-requisites

Step 1: Sign up for an AWS account/ Log in

Step 2: Create an **IAM user account**

Step 3: Create an **access key ID and secret access key**

Step 4: Install **AWS CLI** using this link:
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

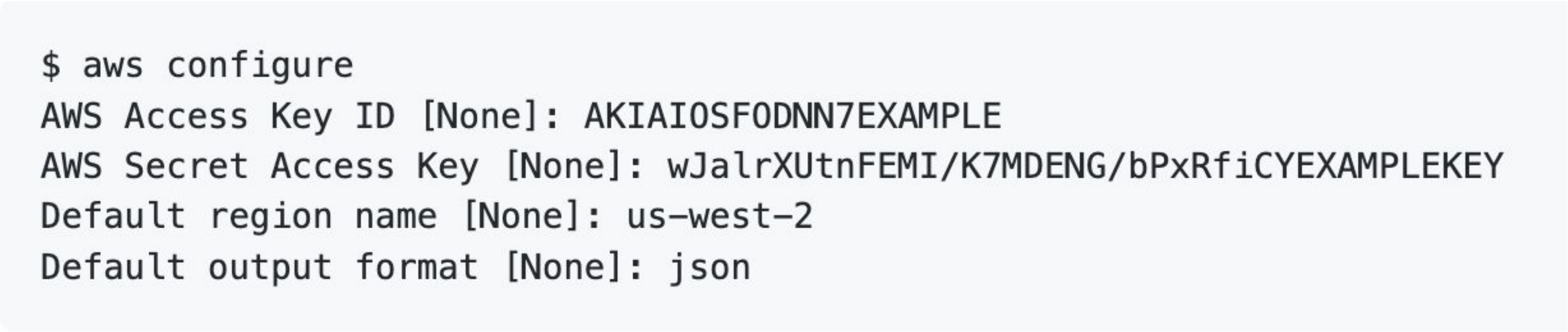
Setup CLI

The AWS CLI stores this information in a profile (a collection of settings) named default in the credentials file.

By default, the information in this profile is used when you run an AWS CLI command that doesn't explicitly specify a profile to use.

For more information on the credentials file, see *Configuration* and credential file settings

Setup CLI



```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

Access key ID and Secret Access Key

Access keys use an access key ID and secret access key that you use to sign programmatic requests to AWS.

Access keys consist of an access key ID and secret access key, which are used to **sign programmatic requests** that you make to AWS. If you don't have access keys, you can **create them from the AWS Management Console.**

Access key ID and Secret Access Key

As a best practice, do not use the AWS account root user access keys for any task where it's not required.

Instead, create a new administrator IAM user with access keys for yourself.

Access key ID and Secret Access Key

****Remember to save your keys!!**

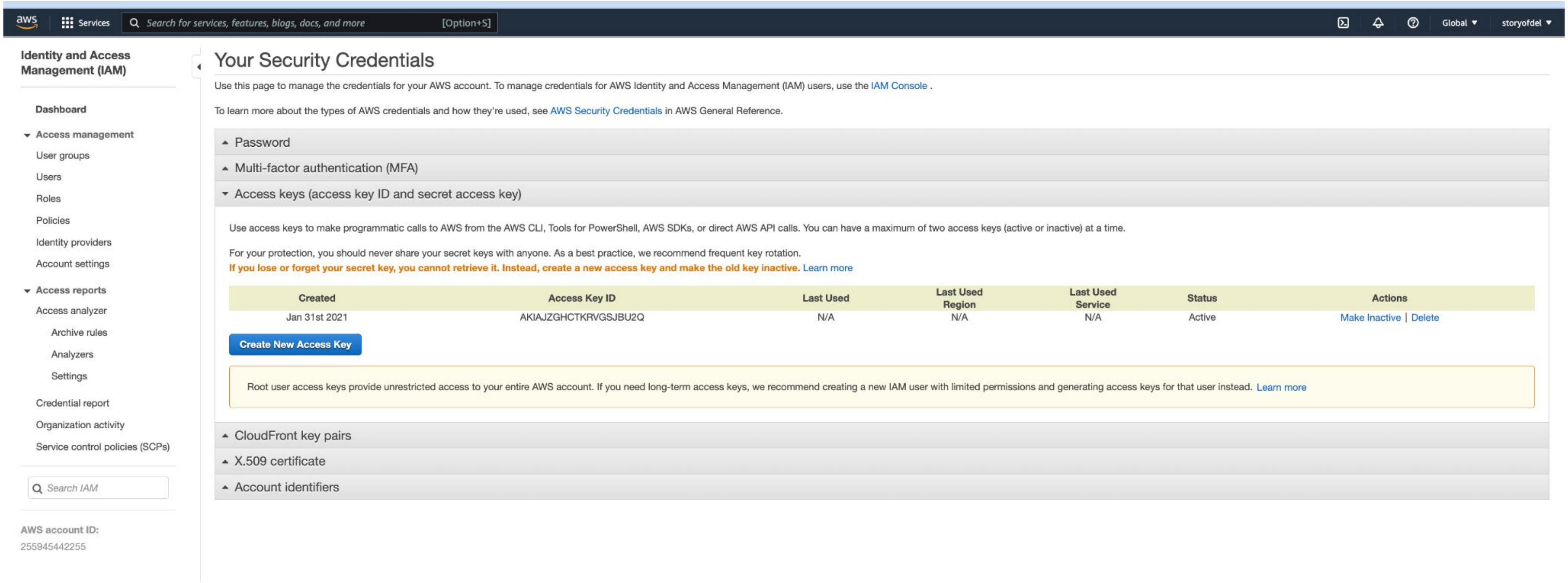
The only time that you can view or download the secret access key is when you create the keys.

You cannot recover them later.

Creating AWS IAM User

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/
2. In the navigation pane, choose Users.
3. Choose the name of the user whose access keys you want to create, and then choose the Security credentials tab.
4. In the Access keys section, choose Create access key.

Creating AWS IAM User



aws
Services
Search for services, features, blogs, docs, and more [Option+S]
Global ▾
storyofdel ▾

AWS account ID:
255945442255

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console .

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

▴ Password

▴ Multi-factor authentication (MFA)

▾ Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Jan 31st 2021	AKIAJZGHCTKRVGSJBU2Q	N/A	N/A	N/A	Active	Make Inactive | Delete

Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead. Learn more

▴ CloudFront key pairs

▴ X.509 certificate

▴ Account identifiers

Creating AWS IAM User

1. 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.

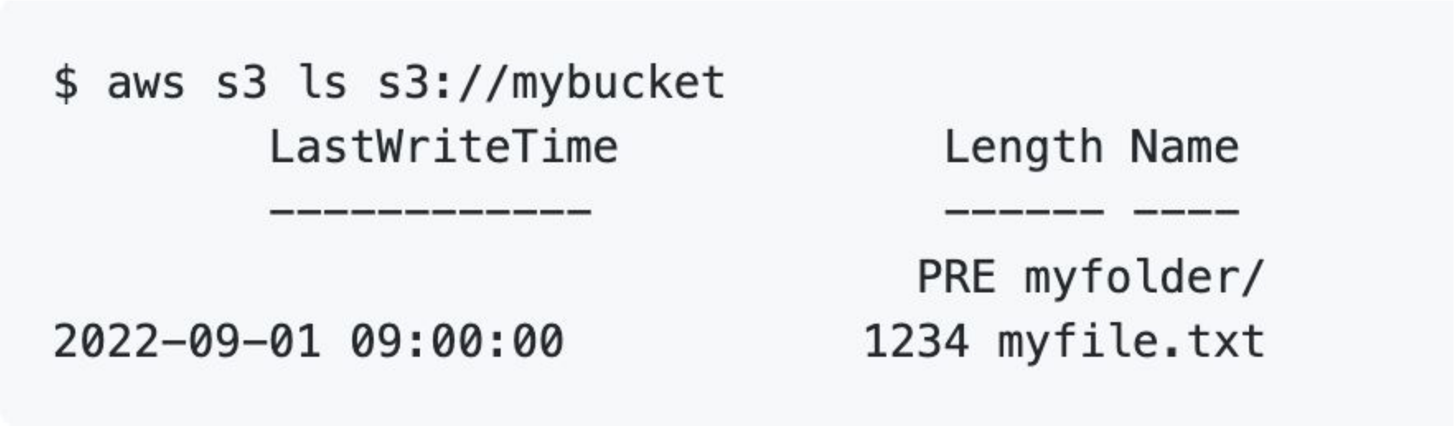
Creating AWS IAM User

1. 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.

AWS S3

AWS S3 CLI

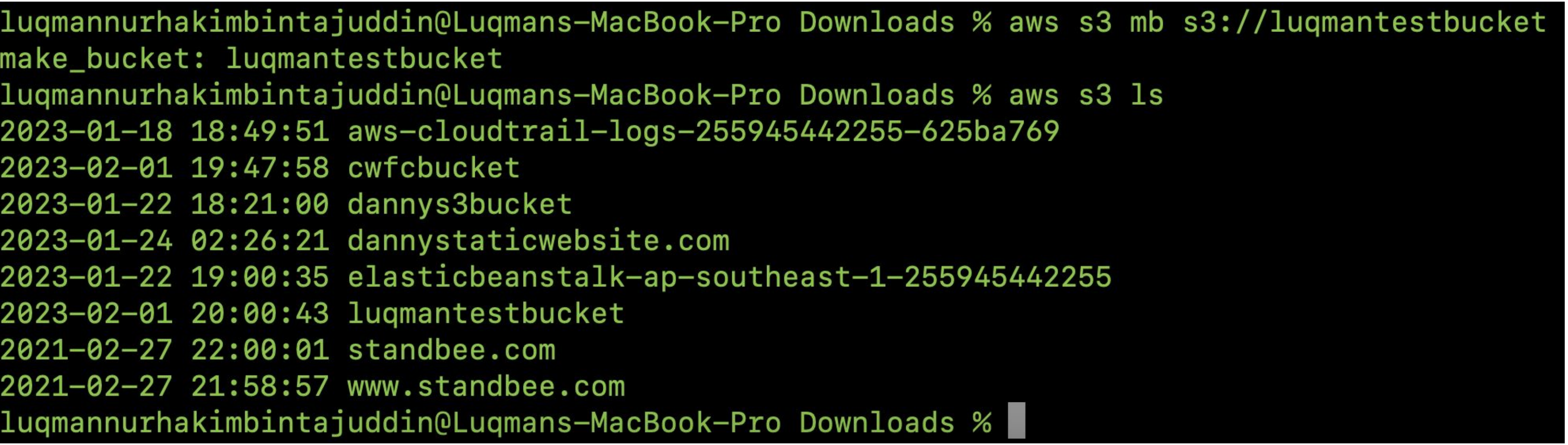
You can view the contents of your S3 buckets in a directory-based listing by using a familiar syntax.



```
$ aws s3 ls s3://mybucket
        LastWriteTime                Length Name
        ------------                 ------ ----
                                        PRE myfolder/
2022-09-01 09:00:00                    1234 myfile.txt
```

AWS S3 CLI

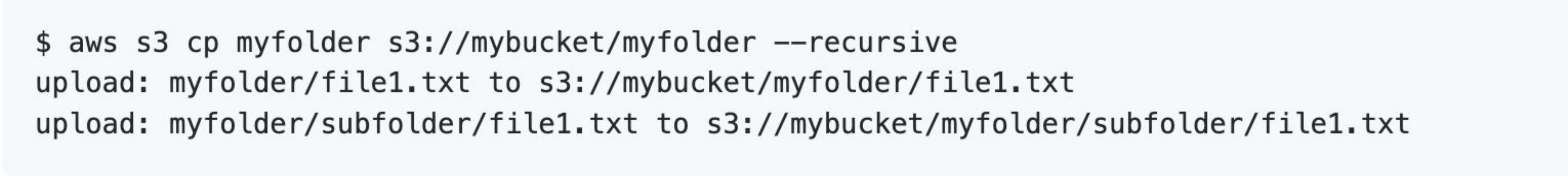
You can create an AWS bucket from the command line



```
luqmannurhakimbintajuddin@Luqmans-MacBook-Pro Downloads % aws s3 mb s3://luqmantestbucket
make_bucket: luqmantestbucket
luqmannurhakimbintajuddin@Luqmans-MacBook-Pro Downloads % aws s3 ls
2023-01-18 18:49:51 aws-cloudtrail-logs-255945442255-625ba769
2023-02-01 19:47:58 cwfcbucket
2023-01-22 18:21:00 dannys3bucket
2023-01-24 02:26:21 dannystaticwebsite.com
2023-01-22 19:00:35 elasticbeanstalk-ap-southeast-1-255945442255
2023-02-01 20:00:43 luqmantestbucket
2021-02-27 22:00:01 standbee.com
2021-02-27 21:58:57 www.standbee.com
luqmannurhakimbintajuddin@Luqmans-MacBook-Pro Downloads %
```

AWS S3 CLI

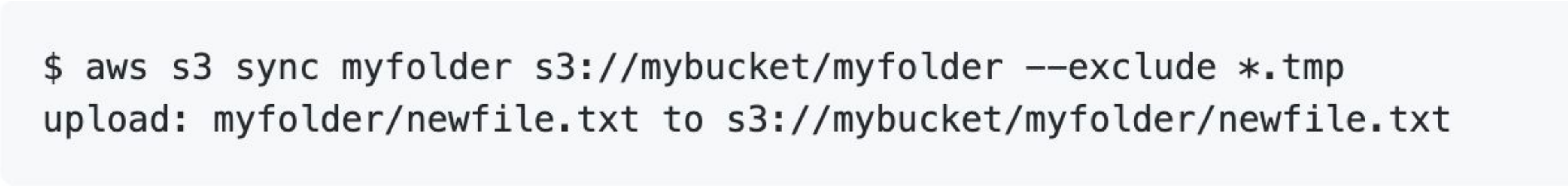
You can perform recursive uploads and downloads of multiple files in a single folder-level command. The AWS CLI will run these transfers in parallel for increased performance.



```
$ aws s3 cp myfolder s3://mybucket/myfolder --recursive
upload: myfolder/file1.txt to s3://mybucket/myfolder/file1.txt
upload: myfolder/subfolder/file1.txt to s3://mybucket/myfolder/subfolder/file1.txt
```

AWS S3 CLI

A sync command makes it easy to synchronize the contents of a local folder with a copy in an S3 bucket.



```
$ aws s3 sync myfolder s3://mybucket/myfolder --exclude *.tmp
upload: myfolder/newfile.txt to s3://mybucket/myfolder/newfile.txt
```

Hands-On Activity

Activity

Learner:

- Clean up AWS.
- Remove/delete/terminate all service/ resources that created.

Instructor

- Clean up AWS.
- Remove/delete/terminate all service/ resources that created.
- Check the AWS account after learner clean up.

What’s Next?