

Lab Report 1-4

Author: Joo Kai Tay (22489437)

Lab 1: Introduction & Setup

Section 1: Creation of AWS Account

1. Navigate to <https://489389878001.signin.aws.amazon.com/console> and login using the provided username and password, changing the password to a secure one on login.

Sign in as IAM user

Account ID (12 digits) or account alias

489389878001

IAM user name

22489437@student.uwa.edu.au

Password

.....

☐ Remember this account

Sign in


[Sign in using root user email](#)

[Forgot password?](#)

2. Navigate to the security credentials tab within IAM. Here, the user can view their details such as their ARN and canonical ID.

My security credentials [Info](#)

Use this page to manage the credentials for your currently authenticated IAM user. To learn more about the types of AWS credentials, see [AWS Security Credentials](#).

**You don't have MFA assigned**

As a security best practice, we recommend you assign MFA.

Assign MFA

Account details


User name

22489437@student.uwa.edu.au


AWS account ID

489389878001

User ARN



Canonical user ID



[AWS IAM credentials](#)


[AWS CodeCommit credentials](#)

[Amazon Keyspaces credentials](#)

Console sign-in

Update console password


Console sign-in link

 <https://489389878001.signin.aws.amazon.com/console>

Console password

Updated 6 minutes ago (2023-08-03 13:13 GMT+8)

Last console sign-in

 7 minutes ago (2023-08-03 13:12 GMT+8)

3. Select the option to create an access key for access to programmatic calls to AWS from the AWS CLI.

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS 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. [Learn more](#)

Create access key

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

4. Select the option for use with the command line interface (CLI)

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

☒ **Command Line Interface (CLI)**

You plan to use this access key to enable the AWS CLI to access your AWS account.

☐ **Local code**

You plan to use this access key to enable application code in a local development environment to access your AWS account.

☐ **Application running on an AWS compute service**

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

☐ **Third-party service**

You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

☐ **Application running outside AWS**

You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

☐ **Other**

Your use case is not listed here.



Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

☒ I understand the above recommendation and want to proceed to create an access key.

Cancel

Next

5. Give the access key a meaningful description tag. This will be useful in the event that you create multiple access keys. The tag should describe the purpose of the key and where it will be used.

Set description tag - *optional* Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

CITSS503_V1

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Cancel

Previous


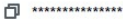
Create access key

6. Once the key has been successfully created, download the access key and secret access key in the .csv file and store it securely. If the key is not stored securely, any user with access to the key can use it to create and use resources associated with your account. For privacy purposes, the key is censored in this image.

Retrieve access keys Info

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 |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
|  |  Show |

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

Download .csv file

Done

Section 2: VM Setup

The Ubuntu virtual machine was already setup prior to the commencement of this unit. The specifications can be found in the attached image.

Oracle VM VirtualBox Manager

File Machine Help

Tools

Ubuntu
 Powered Off

New Add Settings Discard Start

General

Name: Ubuntu
Operating System: Ubuntu (64-bit)

System

Base Memory: 4096 MB
Processors: 4
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 50 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ubuntu.vdi (Normal, 25.00 GB)

Audio

Host Driver: Default
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)


Shared folders

None

Description

None

Preview



Section 3: Software Setup

1. Python 3.10.6 and pip3 were already installed on the machine prior to the commencement of the unit.

4 / 7

```
jookai@jookai:~$ python3 -V
Python 3.10.6
jookai@jookai:~$ pip3 -V
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
jookai@jookai:~$
```

2. The AWS CLI was installed and configured using the access key generated as part of step 6 in section 1

```
jook:~$ aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: ap-southeast-2
Default output format [None]: json
```

3. Boto3 was installed as per the instructions. Boto3 is the AWS SDK for python and it provides a python API for AWS infrastructure services.

```
jook:~$ pip3 install boto3
```

Section 4: Exploring and testing the environment

1. Running the command `aws ec2 describe-regions --output table` is used to retrieve information about the available AWS regions for the EC2 service and display the output in a tabular format. The output of the command is shown below:

```
jook:~$ aws ec2 describe-regions --output table
```

| DescribeRegions | | |
|----------------------------------|---------------------|----------------|
| Regions | | |
| Endpoint | OptInStatus | RegionName |
| ec2.ap-south-1.amazonaws.com | opt-in-not-required | ap-south-1 |
| ec2.eu-north-1.amazonaws.com | opt-in-not-required | eu-north-1 |
| ec2.eu-west-3.amazonaws.com | opt-in-not-required | eu-west-3 |
| ec2.eu-west-2.amazonaws.com | opt-in-not-required | eu-west-2 |
| ec2.eu-west-1.amazonaws.com | opt-in-not-required | eu-west-1 |
| ec2.ap-northeast-3.amazonaws.com | opt-in-not-required | ap-northeast-3 |
| ec2.ap-northeast-2.amazonaws.com | opt-in-not-required | ap-northeast-2 |
| ec2.ap-northeast-1.amazonaws.com | opt-in-not-required | ap-northeast-1 |
| ec2.ca-central-1.amazonaws.com | opt-in-not-required | ca-central-1 |
| ec2.sa-east-1.amazonaws.com | opt-in-not-required | sa-east-1 |
| ec2.ap-southeast-1.amazonaws.com | opt-in-not-required | ap-southeast-1 |
| ec2.ap-southeast-2.amazonaws.com | opt-in-not-required | ap-southeast-2 |
| ec2.eu-central-1.amazonaws.com | opt-in-not-required | eu-central-1 |
| ec2.us-east-1.amazonaws.com | opt-in-not-required | us-east-1 |
| ec2.us-east-2.amazonaws.com | opt-in-not-required | us-east-2 |
| ec2.us-west-1.amazonaws.com | opt-in-not-required | us-west-1 |
| ec2.us-west-2.amazonaws.com | opt-in-not-required | us-west-2 |

2. We can retrieve similar information using Python3 by running the following commands. This returns and prints the same data as before. However, this data is not tabulated.


```
python3
>>> import boto3
>>> ec2 = boto3.client('ec2')
>>> response = ec2.describe_regions()
>>> print(response)
```

```
book:~$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import boto3
>>> ec2 = boto3.client('ec2')
>>> response = ec2.describe_regions()
>>> print(response)
{'Regions': [{'Endpoint': 'ec2.ap-south-1.amazonaws.com', 'RegionName': 'ap-south-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.eu-north-1.amazonaws.com', 'RegionName': 'eu-north-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.eu-west-3.amazonaws.com', 'RegionName': 'eu-west-3', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.eu-west-2.amazonaws.com', 'RegionName': 'eu-west-2', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.eu-west-1.amazonaws.com', 'RegionName': 'eu-west-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ap-northeast-3.amazonaws.com', 'RegionName': 'ap-northeast-3', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ap-northeast-2.amazonaws.com', 'RegionName': 'ap-northeast-2', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ap-northeast-1.amazonaws.com', 'RegionName': 'ap-northeast-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ca-central-1.amazonaws.com', 'RegionName': 'ca-central-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.sa-east-1.amazonaws.com', 'RegionName': 'sa-east-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ap-southeast-1.amazonaws.com', 'RegionName': 'ap-southeast-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.ap-southeast-2.amazonaws.com', 'RegionName': 'ap-southeast-2', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.eu-central-1.amazonaws.com', 'RegionName': 'eu-central-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.us-east-1.amazonaws.com', 'RegionName': 'us-east-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.us-east-2.amazonaws.com', 'RegionName': 'us-east-2', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.us-west-1.amazonaws.com', 'RegionName': 'us-west-1', 'OptInStatus': 'opt-in-not-required'}, {'Endpoint': 'ec2.us-west-2.amazonaws.com', 'RegionName': 'us-west-2', 'OptInStatus': 'opt-in-not-required'}], 'ResponseMetadata': {'RequestId': 'f8491f25-8134-4ea3-8b83-6656fa8c0b02', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f8491f25-8134-4ea3-8b83-6656fa8c0b02', 'cache-control': 'no-cache, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDomains', 'vary': 'accept-encoding', 'content-type': 'text/xml; charset=UTF-8', 'content-length': '3875', 'date': 'Thu, 03 Aug 2023 05:32:16 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
```

3. The following code allows us to tabulate the data and print out only the columns containing the endpoint and RegionName:

```
import boto3

# Get the region data from boto3
ec2 = boto3.client('ec2')
response = ec2.describe_regions()

# Print the header for the columns
print(f"{'Endpoint':<40} {'RegionName'}")

# Print each region's endpoint and region name
for region in response['Regions']:
    print(f"{region['Endpoint']:<40} {region['RegionName']}")
```

The execution of this code (stored in a file named `cits5503_lab1.py`) is shown below:

```
jook:~/Desktop/cits5503/lab1$ python3 cits5503_lab1.py
Endpoint                               RegionName
ec2.ap-south-1.amazonaws.com           ap-south-1
ec2.eu-north-1.amazonaws.com           eu-north-1
ec2.eu-west-3.amazonaws.com            eu-west-3
ec2.eu-west-2.amazonaws.com            eu-west-2
ec2.eu-west-1.amazonaws.com            eu-west-1
ec2.ap-northeast-3.amazonaws.com        ap-northeast-3
ec2.ap-northeast-2.amazonaws.com        ap-northeast-2
ec2.ap-northeast-1.amazonaws.com        ap-northeast-1
ec2.ca-central-1.amazonaws.com          ca-central-1
ec2.sa-east-1.amazonaws.com             sa-east-1
ec2.ap-southeast-1.amazonaws.com        ap-southeast-1
ec2.ap-southeast-2.amazonaws.com        ap-southeast-2
ec2.eu-central-1.amazonaws.com          eu-central-1
ec2.us-east-1.amazonaws.com            us-east-1
ec2.us-east-2.amazonaws.com            us-east-2
ec2.us-west-1.amazonaws.com            us-west-1
ec2.us-west-2.amazonaws.com            us-west-2
jook:~/Desktop/cits5503/lab1$
```