Step 1

You need to write a script to apply the given policy to the bucket that you created last week (if you have deleted it, please create a new one. We don't mind how you create a bucket this week.)

For a python script, you may need to import boto3 and JSON.

Hints:

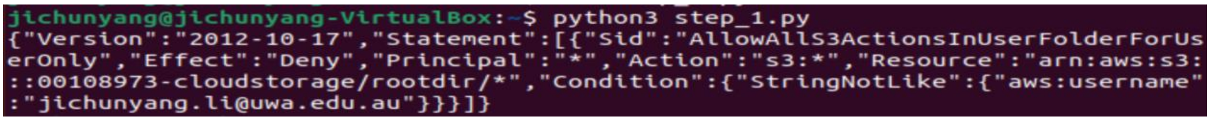
1) You may use JSON.dumps() function converts a Python object into a JSON string.

2) put_bucket_policy() can help you apply the policy to your bucket.

3) get_bucket_policy() can help you get the policy you apply to a bucket.

Expected Output:

The policy you applied to a bucket.



```
jichunyang@jichunyang-VirtualBox:~$ python3 step_1.py
{"Version":"2012-10-17","Statement":[{"Sid":"AllowAllS3ActionsInUserFolderForUs
erOnly","Effect":"Deny","Principal":"*","Action":"s3:*","Resource":"arn:aws:s3:
::00108973-cloudstorage/rootdir/*","Condition":{"StringNotLike":{"aws:username"
:"jichunyang.li@uwa.edu.au"}}}]}
```

You need to present the screenshots of your script and the output in your lab note.

Step 2

Frist, write a boto3 script to create a key then add an alias with your student ID.

Hints:

1) You may use client = boto3.client('kms')

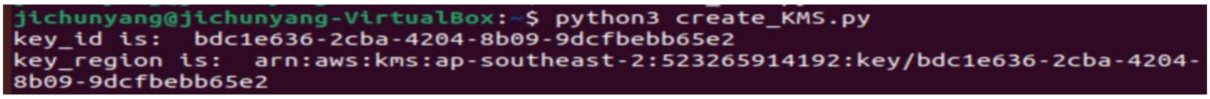
2) Use create_key() and create_alias() functions to create a key and add the alias.

3) You may use ['KeyMetadata']['KeyId'] to get the key id.

4) You may use ['keyMetadata']['Arn'] to get ARN.

Expected Output:

You need to present your python script, key id and ARN in your lab note.



```
jichunyang@jichunyang-VirtualBox:~$ python3 create_KMS.py
key_id is:  bdc1e636-2cba-4204-8b09-9dcfbebb65e2
key_region is:  arn:aws:kms:ap-southeast-2:523265914192:key/bdc1e636-2cba-4204-
8b09-9dcfbebb65e2
```

Second, set a new KMS policy (we have given you in the lab sheet)

Same as step 1, you may need import json

Hints:

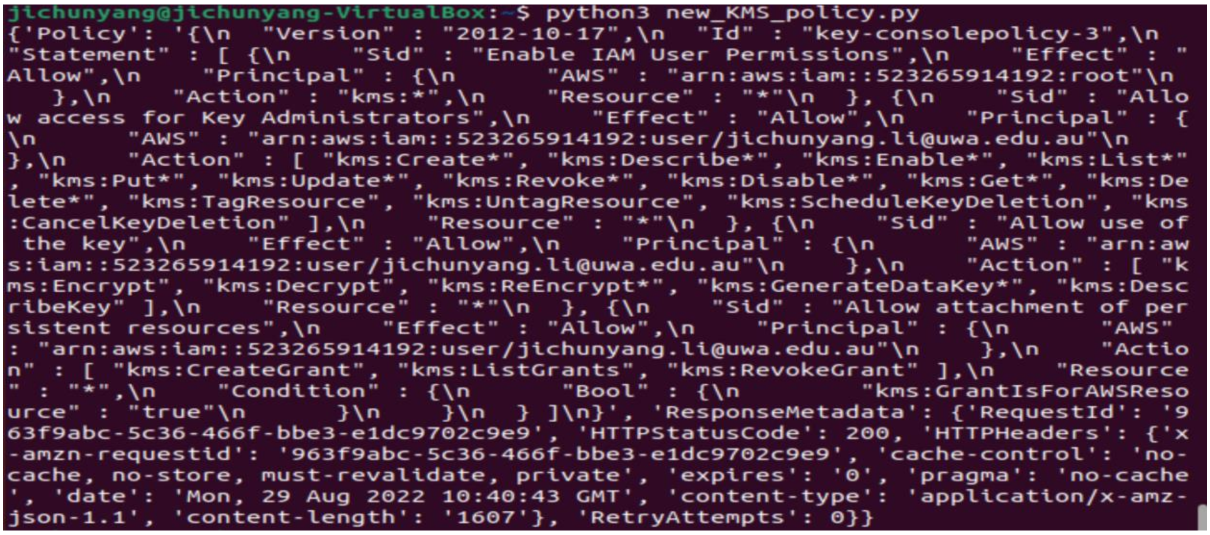
1) You may use put_key_policy() function to apply the policy.

2) You may use get_key_policy() function to output the policy.

3) Replace the user account ID and IAM user with your ID and User name in the json file, you can find this information on AWS Console (on the top right corner).

Expected output:

Output the policy and your python code in your lab note.



```
jichunyang@jichunyang-VirtualBox:~$ python3 new_KMS_policy.py
{'Policy': '{\n  "Version" : "2012-10-17",\n  "Id" : "key-consolepolicy-3",\n
"Statement" : [ {\n    "Sid" : "Enable IAM User Permissions",\n    "Effect" : "
Allow",\n    "Principal" : {\n      "AWS" : "arn:aws:iam::523265914192:root"\n
    },\n    "Action" : "kms:*",\n    "Resource" : "*"\n  }, {\n    "Sid" : "Allo
w access for Key Administrators",\n    "Effect" : "Allow",\n    "Principal" : {
\n      "AWS" : "arn:aws:iam::523265914192:user/jichunyang.li@uwa.edu.au"\n
},\n    "Action" : [ "kms:Create*", "kms:Describe*", "kms:Enable*", "kms:List*"
, "kms:Put*", "kms:Update*", "kms:Revoke*", "kms:Disable*", "kms:Get*", "kms:De
lete*", "kms:TagResource", "kms:UntagResource", "kms:ScheduleKeyDeletion", "kms
:CancelKeyDeletion" ],\n    "Resource" : "*"\n  }, {\n    "Sid" : "Allow use of
 the key",\n    "Effect" : "Allow",\n    "Principal" : {\n      "AWS" : "arn:aw
s:iam::523265914192:user/jichunyang.li@uwa.edu.au"\n    },\n    "Action" : [ "k
ms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*", "kms:GenerateDataKey*", "kms:Desc
ribeKey" ],\n    "Resource" : "*"\n  }, {\n    "Sid" : "Allow attachment of per
sistent resources",\n    "Effect" : "Allow",\n    "Principal" : {\n      "AWS"
: "arn:aws:iam::523265914192:user/jichunyang.li@uwa.edu.au"\n    },\n    "Actio
n" : [ "kms:CreateGrant", "kms:ListGrants", "kms:RevokeGrant" ],\n    "Resource
" : "*",\n    "Condition" : {\n      "Bool" : {\n        "kms:GrantIsForAWSReso
urce" : "true"\n      }\n    }\n  } ]\n}', 'ResponseMetadata': {'RequestId': '9
63f9abc-5c36-466f-bbe3-e1dc9702c9e9', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x
-amzn-requestid': '963f9abc-5c36-466f-bbe3-e1dc9702c9e9', 'cache-control': 'no-
cache, no-store, must-revalidate, private', 'expires': '0', 'pragma': 'no-cache
', 'date': 'Mon, 29 Aug 2022 10:40:43 GMT', 'content-type': 'application/x-amz-
json-1.1', 'content-length': '1607'}, 'RetryAttempts': 0}}
```

[I suggest you write the rest of step 2 in a single .py file]

Then generate a data key. Why do we need to generate. Please look at this page: https://docs.aws.amazon.com/zh_cn/kms/latest/APIReference/API_GenerateDataKey.html

How do we generate a data key? Please look at this page:

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/kms.html#KMS.Client.generate_data_key

Hints:

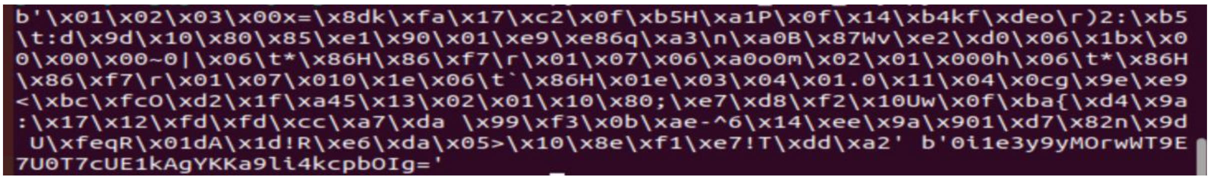
1) You may need to import base64

2) Think about your key_spec.

3) Use generate_data_key()function to create a data key.

4) Return the encrypted and plaintext data key, you may use [‘CiphertextBlob’] and base64.b64encode(~[‘Plaintext’])

Expected output: (present the output and your python scripts in your lab note)



```
b'\x01\x02\x03\x00x=\x8dk\xfa\x17\xc2\x0f\xb5H\xa1P\x0f\x14\xb4kf\xdeo\r)2:\xb5
\t:d\x9d\x10\x80\x85\xe1\x90\x01\xe9\xe86q\xa3\n\xa0B\x87Wv\xe2\xd0\x06\x1bx\x0
0\x00\x00~0|\x06\t*\x86H\x86\xf7\r\x01\x07\x06\xa0o0m\x02\x01\x000h\x06\t*\x86H
\x86\xf7\r\x01\x07\x010\x1e\x06\t`\x86H\x01e\x03\x04\x01.0\x11\x04\x0cg\x9e\xe9
<\xbc\xfcO\xd2\x1f\xa45\x13\x02\x01\x10\x80;\xe7\xd8\xf2\x10Uw\x0f\xba{\xd4\x9a
:\x17\x12\xfd\xfd\xcc\xa7\xda \x99\xf3\x0b\xae-^6\x14\xee\x9a\x901\xd7\x82n\x9d
 U\xfeqR\x01dA\x1d!R\xe6\xda\x05>\x10\x8e\xf1\xe7!T\xdd\xa2' b'0i1e3y9yMOrwWT9E
7U0T7cUE1kAgYKKa9li4kcpbOIg='
```

After that, try to use the generated data key to encrypt a local file.

Hints:

1) You may need to use ‘from cryptography.fernet import Fernet’

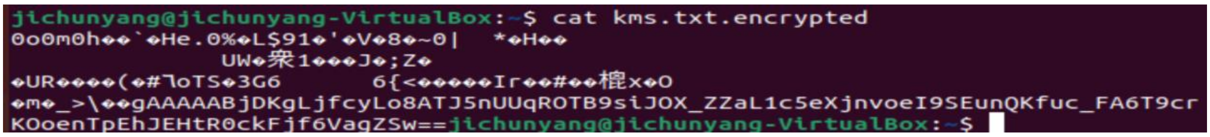
2) Create a local .txt file then encrypt it

3) For this section, I encrypt a file on my machine rather than in a S3 bucket.

Expected output:

I created a file named kms.txt, the contents of it are “Hi world!”. After running my file, I get a file named kms.txt.encrypted. Then use “cat <file name>” with terminal, I got that:

Put the output and the python script in your lab note.



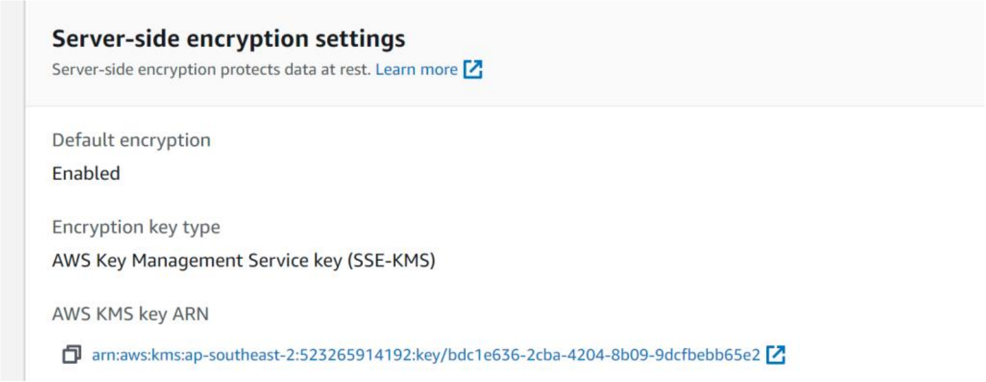
```
jichunyang@jichunyang-VirtualBox:~$ cat kms.txt.encrypted
0o0m0h��`�He.0%�L$91�'�V�8�~0|   *�H��
                 UW�衆1���J�;Z�
�UR����(�#˥oTS�3G6       6{<�����Ir��#��棍x�O
�m�_>\��gAAAAABjDKgLjfcyLo8ATJ5nUUqROTB9siJOX_ZZaL1c5eXjnvoeI9SEunQKfuc_FA6T9cr
KOoenTpEhJEHtR0ckFjf6VagZSw==jichunyang@jichunyang-VirtualBox:~$ 
```

After that, upload your encrypted file to your S3 bucket, then check it with AWS console.

Hints:

Set the ServerSideEncryption = ‘aws:kms’

Expected output: (present the screenshot and your python code in your lab note)



Server-side encryption settings

Server-side encryption protects data at rest. Learn more

Default encryption
Enabled

Encryption key type
AWS Key Management Service key (SSE-KMS)

AWS KMS key ARN
arn:aws:kms:ap-southeast-2:523265914192:key/bdc1e636-2cba-4204-8b09-9dcfbebb65e2

Finally, try to decrypt the encrypted file.

Hints:

1) Write a function for decrypting the data key first

2) Write a function to decrypt the file

3) You may need from cryptography.fernet import Fernet

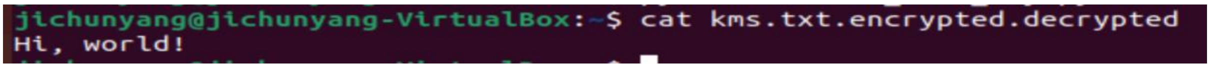
4) You need to download the file from S3 bucket.

Expected Output:

You need to output the contents of your decrypted file, the contents should be same as the contents in your created .txt file. (In my kms.txt file, the content is “Hi, world!”)

Use cat <file name>, to show the content of your file.

Present your python code and the screenshot in your lab note.



```
jichunyang@jichunyang-VirtualBox:~$ cat kms.txt.encrypted.decrypted
Hi, world!
```

Step 3

Using the given python script to encrypt a local file, then upload it to your S3 bucket. Finally look at the server-side encryption setting.

Hints:

1) You don’t need to consider how to encrypt your file, just need to insert your script into the given python script.

2) Your own scripts can just be “upload” and “download” a file

3) Please do that first, before your edit the given script:

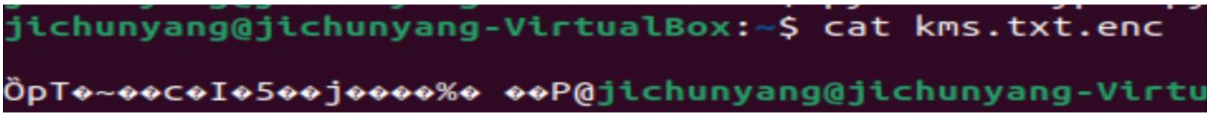
pip3 uninstall PyCrypto

pip3 install -U PyCryptodome

Expected Output:

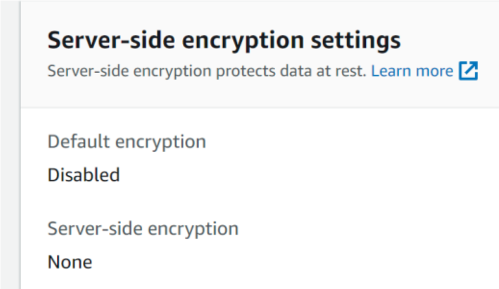
Present these 3 screenshots and your python code in your lab note.

1) Using cat <file name> to look at the encrypted file.



```
jichunyang@jichunyang-VirtualBox:~$ cat kms.txt.enc
ÒpT◆~◆◆c◆I◆5◆◆j◆◆◆◆◆%◆ ◆◆P@jichunyang@jichunyang-Virtu
```

2) On console, present the screenshot of the server-side encryption settings for your uploaded file.



Server-side encryption settings

Server-side encryption protects data at rest. Learn more

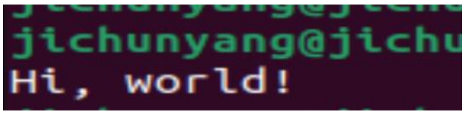
Default encryption

Disabled

Server-side encryption

None

3) Decrypt your encrypted file, present the content. Using cat <file name>. I use the same file in step 2.



```
jichunyang@jichu
Hi, world!
```

Answer the following question in your lab note:

What is the performance difference between using KMS and using the custom solution?