**Lab 4 Notes**

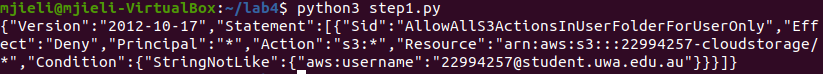
Student ID: 22994257

Name: Gaoyuan Zhang

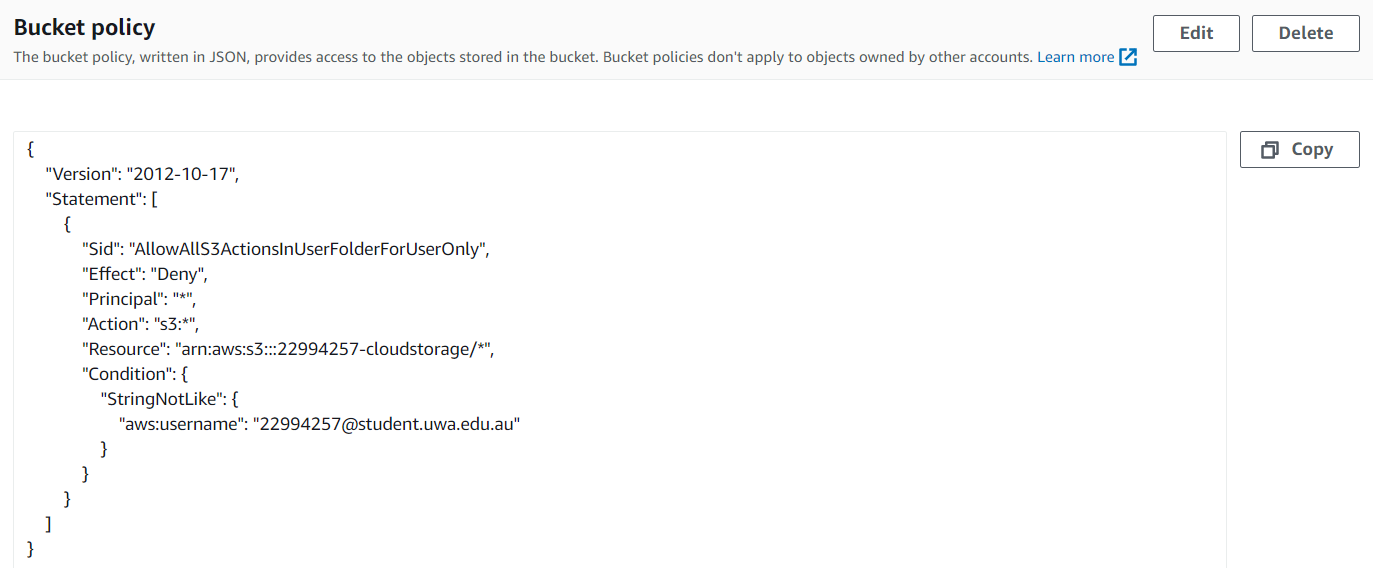
### **[Step 1] Apply policy to restrict permissions on bucket**

1. Create step1.py and use the function put\_bucket\_policy to set the new policy



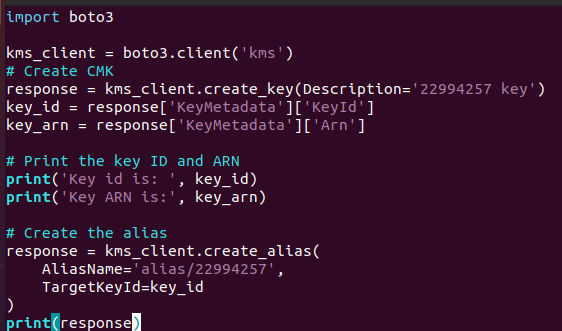


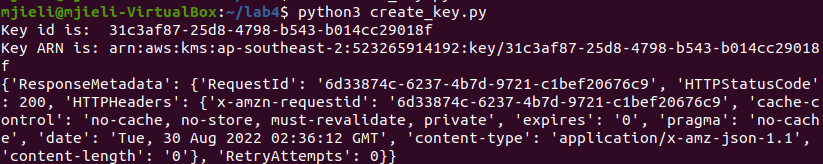
2. We can see the new policy in the S3 bucket permission page.



### **[Step 2] AES Encryption using KMS**

1. Create a new program called create\_key.py. Use create\_key() and create\_alias() functions to create a key and add the alias.

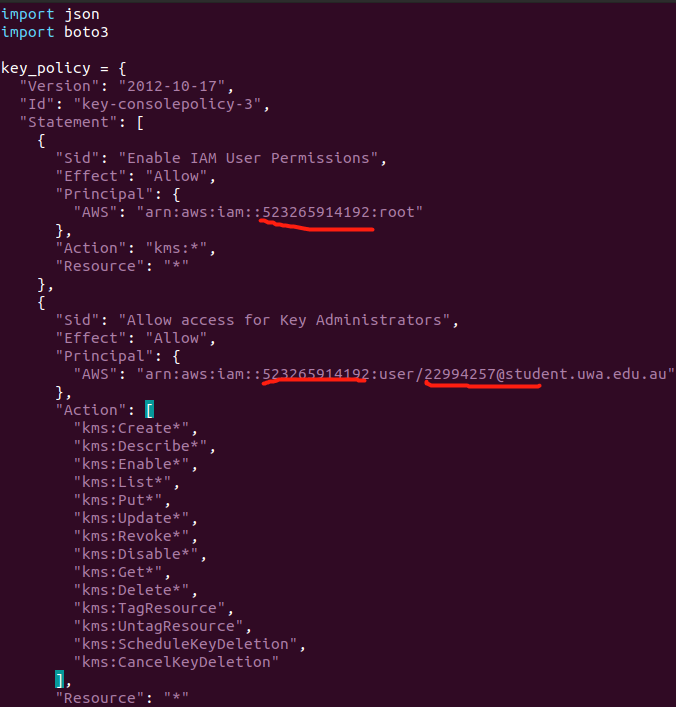


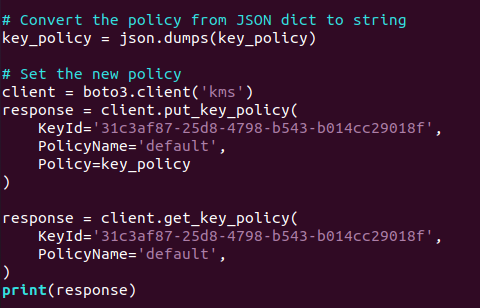


Key id: 31c3af87-25d8-4798-b543-b014cc29018f

ARN: arn:aws:kms:ap-southeast-2:523265914192:key/31c3af87-25d8-4798-b543-b014cc29018f

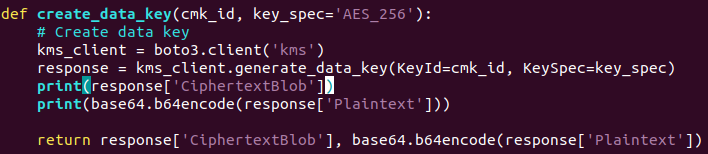
2. Create a new program called new\_KMS\_policy.py. Use put\_key\_policy() to apply the policy and get\_key\_policy() to get the result.

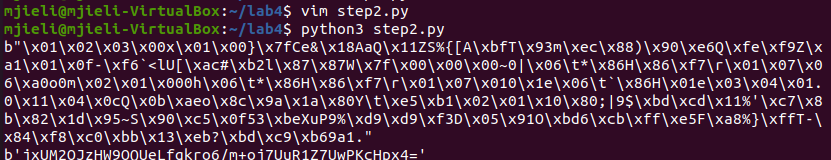




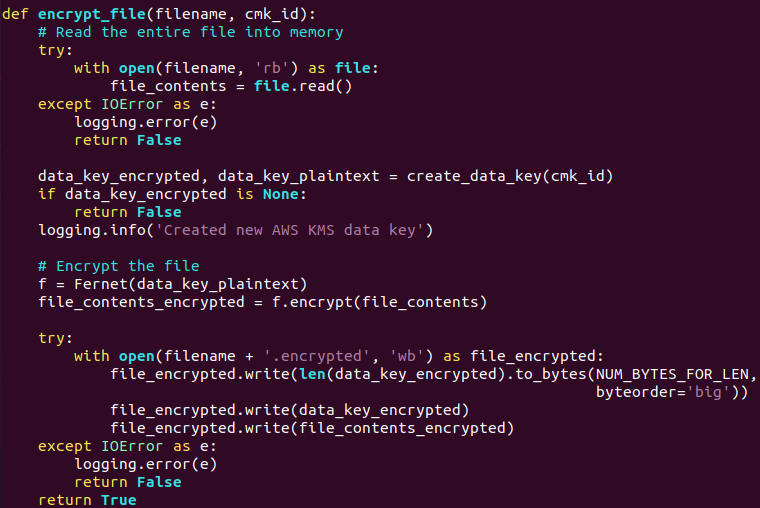


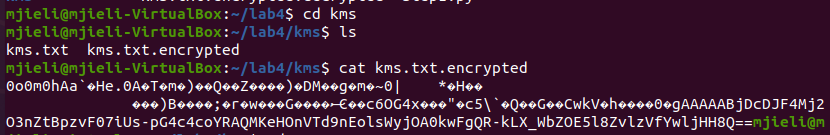
3. Generate a data key.





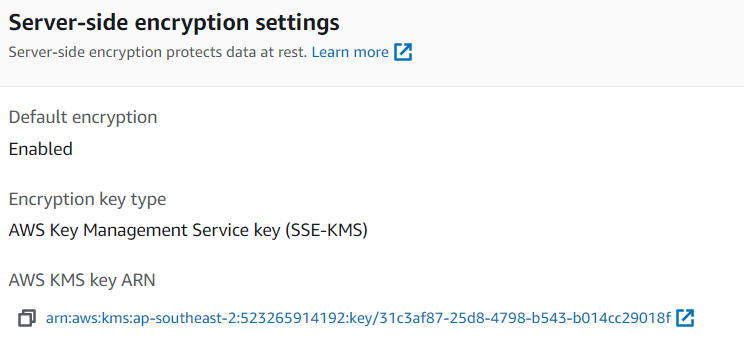
4. Use the generated data key to encrypt a local file ’kms.txt’.





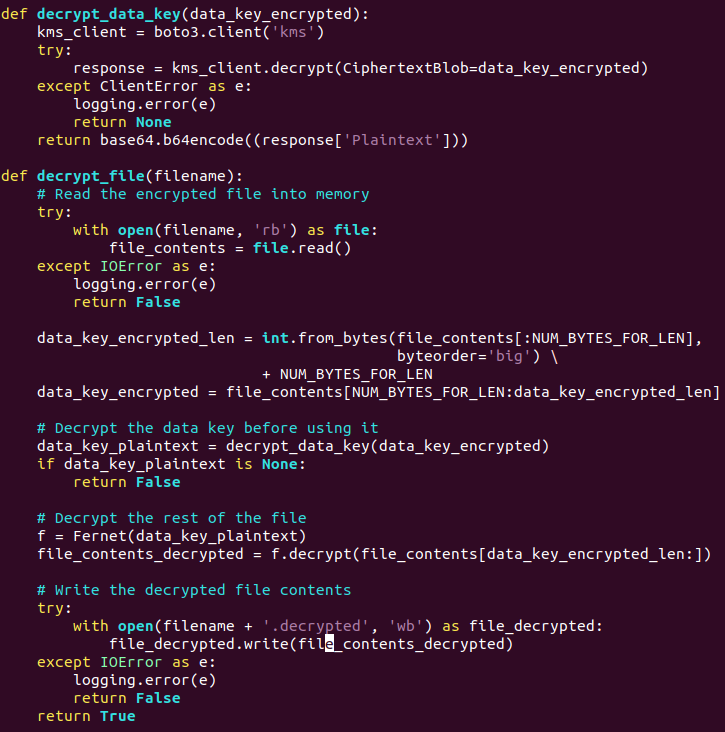
5. Upload it to S3 bucket, then check it with AWS console.

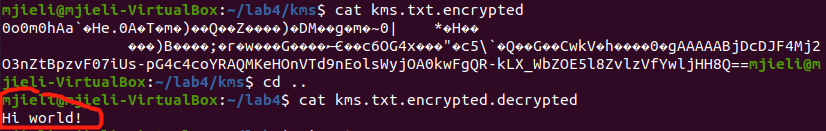




6. Download the encrypted file from cloud and decrypt it.



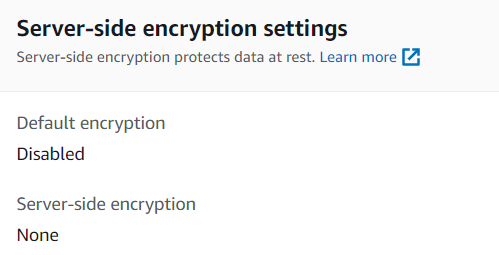


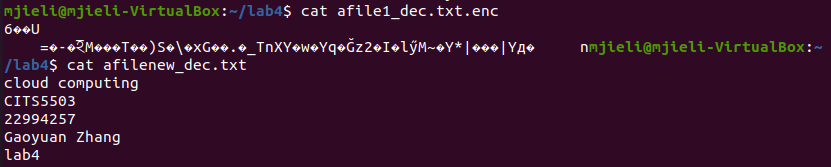


### **[Step 3] AES Encryption using local python library pycryptodome**

1. The 3 screenshots







2. Question: What is the performance difference between using KMS and using the custom solution?

If I want to encrypt a file larger than 4KB using KMS, I should use a technique called "envelope encryption", which will encrypt the file in chunks. So, if I use the custom solution to do that, it’s faster.