Northeastern University – Silicon Valley

CS 6620 Cloud Computing

Homework Set #5 [100 points]

INSTRUCTIONS: ***Please provide clear explanations in your own sentences, directly answering the question, demonstrating your understanding of the question and its solution, in depth, with sufficient detail. Submit your solutions [PDF preferred]. Include your full name. Do not email the solutions.***

PART I: Concepts and Theory, Algorithms [60 points]

Please provide accurate, concise (maximum 10 sentence) answers. **All questions carry 10 points.**

1. Based on your study of text/references, explain (i) what each of the below AWS feature is and (ii) why and how they are needed/used in one example business application (Chapter 5): [20 points]

    A. Amazon S3
    B. Amazon EFS
    C. Amazon EBS
    D. AWS Storage Gateway

    Why are these 4 separate services necessary? That is, how are they different?
2. Explain what a Storage Area newtork (SAN) and Network Attached Storage (NAS) are, and their differences [10 points]
3. Explain the purpose of and differences among a database, data lake, data warehouse. Which AWS storage services are sutable to build each of these on the cloud? [10 points]

4. Briefly expalin how SSD, HDD, Flash disk and Optical Disk are different. Separately, explain what storage Tiering is and why it is needed in Cloud. Then, identify which stoarge technology (SSD etc) is suitable for which tier. (See references). [15 points]

PART II: LAB [45 points]

Coding needed: Following the examples in Chpater 5 of the textbook OR external links and examples, implement a basic setup and use of the below 3 AWS storage services. Goal is to demonstrate the basic functionality only:

A. Amazon S3
B. Amazon EFS
C. Amazon EBS

Advanced fetaures like security, ACLS etc are not necessary for this exercise.

References

https://cloudian.com/guides/data-backup/storage-tiering/
https://experience.dropbox.com/get-organized/storage-devices
https://www.sqlshack.com/getting-started-with-amazon-s3-and-python/

PART I: Concepts and Theory, Algorithms

1. Based on your study of text/references, explain (i) what each of the below AWS feature is and (ii) why and how they are needed/used in one example business application (Chapter 5): [20 points]

A. Amazon S3

For the concept of Amazon S3 (Amazon Simple Storage Service), according to the website of amazon, it is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, cloud-native applications, backups, archive, machine learning, and analytics.

For the explanation of examples, Amazon S3 could build fast, powerful mobile and web-based cloud-native apps that scale automatically in a highly available configuration while running Cloud-Native Applications.

B. Amazon EFS

For the concept of Amazon Elastic File System (Amazon EFS), according to the website of amazon, it provides a simple, scalable, elastic file system for Linux-based workloads for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need – when they need it. It is designed to provide massively parallel shared access to thousands of Amazon EC2 instances, enabling your applications to achieve high levels of aggregate throughput and IOPS with consistent low latencies. Amazon EFS is a fully managed service that requires no changes to your existing applications and tools, providing access through a standard file system interface for seamless integration. Amazon EFS is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability. You can access your file systems across AZs and AWS Regions and share files between thousands of Amazon EC2 instances and on-premises servers via AWS Direct Connect or AWS VPN.

For the explanation of examples -- machine learning (ML) and big data analytics workloads, Amazon EFS could accelerate data science and then it's easy to use and scale, Amazon EFS offers the performance and consistency.

C. Amazon EBS

For the concept of Amazon Elastic Block Store (Amazon EBS), it provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes—all while paying a low price for only what you provision.

For the explanation of examples, we should know that Amazon EBS provides the following volume types: General Purpose SSD, Provisioned IOPS SSD, Throughput Optimized HDD, and Cold HDD. For the general Purpose SSD volumes (gp2 and gp3), it balances price and performance for a wide variety of transactional workloads. Because these volumes are ideal for use cases such as boot volumes, medium-size single instance databases, and development and test environments.

D. AWS Storage Gateway

For the concept of the Storage Gateway, according to the website of amazon, it is a hybrid storage service that enables your on-premises applications to seamlessly use AWS cloud storage. You can use the service for backup and archiving, disaster recovery, cloud data processing, storage tiering, and migration. Your applications connect

to the service through a virtual machine or hardware gateway appliance using standard storage protocols, such as NFS, SMB and iSCSI. The gateway connects to AWS storage services, such as Amazon S3, S3 Glacier, and Amazon EBS, providing storage for files, volumes, and virtual tapes in AWS. The service includes a highly optimized data transfer mechanism, with bandwidth management, automated network resilience, and efficient data transfer, along with a local cache for low-latency on-premises access to your most active data.

For the explanation of examples, the Storage Gateway supports four key hybrid cloud use cases – (1) move backups and archives to the cloud, (2) reduce on-premises storage with cloud-backed file shares, (3) provide on-premises applications low latency access to data stored in AWS, and (4) data lake access for pre and post processing workflows. And depending on the use case, Storage Gateway provides three types of storage interfaces for the on-premises applications: file, volume, and tape.

Why are these 4 separate services necessary? That is, how are they different?
The four services work quite differently and offer different levels of performance, cost, availability, and scalability. The table below from online website compares Amazon S3, EBS, and EFS in terms of performance, cost, availability, accessibility, access control, and storage or file size limits enforced by each service.

	Performance	Availability and Accessibility	Access Control	Storage and File Size Limits	Cost
Amazon S3	- Supports 3500 PUT / LIST / DELETE requests per second - Scalable to 5500 GET requests per second	- Usually 99.9% available - If lower, returns 10-100% of cost as service credits - Accessible via Internet using APIs	- Access is based on IAM - Uses bucket policies and user policies - Public access via Block Public Access	- No limit on quantity of objects - Individual objects up to 5TB	- **Free tier:** 5GB - **First 50 TB/month:** $0.023 per GB - **Next 450 TB/month:** $0.022 per GB - **Over 500 TB/month:** $0.021 per GB
AWS EBS	- HDD volumes: 250-500 IOPS/volume depending on volume type - SSD volumes: 16-64K IOPS/volume	- 99.99% available - Accessible via single EC2 instance	- Security groups - User-based authentication (IAM)	- Max storage size of 16TB - No file size limit on disk	- **Free tier:** 30GB - **General Purpose:** $0.045 per GB/month - **Provisioned SSD:** $0.125 per GB/month, $0.065 per IOPS/month
AWS EFS	- 3GB/s baseline performance - Up to 10GB/s - Up to 7K IOPS	- No publicly available SLA - Up to 1,000 concurrent EC2 instances - Accessible from any AZ or region	- IAM user-based authentication - Security groups	- 16TB per volume - 52TB maximum for individual files	- **Standard storage:** $0.30-$0.39 per GB-month depending on region - **Infrequent storage:** $0.025-$0.03 per GB-month - **Provisioned throughput:** $6 per MB/s-month

In detail:
Amazon S3 is cheapest for data storage alone. However, there are various other pricing parameters in S3, including cost per number of requests made, S3 Analytics, and data transfer out of S3 per gigabyte. EFS has the simplest cost structure.

Amazon S3 can be accessed from anywhere. AWS EBS is only available in a particular region, while you can share files between regions on multiple EFS instances.

EBS and EFS are both faster than Amazon S3, with high IOPS and lower latency.

EBS is scalable up or down with a single API call. Since EBS is cheaper than EFS, you can use it for database backups and other low-latency interactive applications that require consistent, predictable performance.

EFS is best used for large quantities of data, such as large analytic workloads. Data at this scale cannot be stored on a single EC2 instance allowed in EBS—requiring users to break up data and distribute it between EBS instances. The EFS service allows concurrent access to thousands of EC2 instances, making it possible to process and analyze large amounts of data seamlessly.

Storage Gateway is a Hybrid Cloud for Storage. Part of the infrastructure is on the cloud; another part is on-premises. EBS is block, EFS is file, S3 is object. Whereas Storage Gateway are fancy local caches and gateways to access the remote storage with familiar protocols.

Storage Gateway still requires the other storage systems. If AWS is like a non-cloud remote storage array, SG is the filer and VTL options that speak file and tape.

2. Explain what a Storage Area network (SAN) and Network Attached Storage (NAS) are, and their differences [10 points]

For the concept of Storage Area Network (SAN), it is a specialized, high-speed network that provides block-level network access to storage. SANs are typically composed of hosts, switches, storage elements, and storage devices that are interconnected using a variety of technologies, topologies, and protocols. SANs may also span multiple sites.

A SAN presents storage devices to a host such that the storage appears to be locally attached. This simplified presentation of storage to a host is accomplished through the use of different types of virtualization.

For the concept of Network-attached storage (NAS), it is a file-level (as opposed to block-level storage) computer data storage server connected to a computer network providing data access to a heterogeneous group of clients. NAS is specialized for serving files either by its hardware, software, or configuration.

Differences:
NAS is a single storage device that serves files over Ethernet and is relatively inexpensive and easy to set up, while a SAN is a tightly coupled network of multiple devices that is more expensive and complex to set up and manage. From a user perspective, the biggest difference between NAS and SAN is that NAS devices deliver shared storage as network mounted volumes and use protocols like NFS and SMB/CIFS, while SAN-connected disks appear to the user as local drives.

3. Explain the purpose of and differences among a database, data lake, data warehouse. Which AWS storage services are sutable to build each of these on the cloud? [10 points]

Typically, an organization will require a data lake, data warehouse and database(s) for different use cases.

A database is used to store, search and report on structured data from a single source. They are the simplest to create and SQL can be used to query and report on the data.

A data warehouse is used to store large amounts of structured data from multiple sources in a centralized place. Organizations invest in building data warehouses because of its ability to deliver business insights from across the company, and quickly.

A data lake stores structured, semi-structured and unstructured data, supporting the ability to store raw data from all sources without the need to process or transform it at that time.

For the difference, a data lake is where streams and rivers of data from various sources meet. All data is allowed, no matter if it is structured or unstructured and no processing is done to the data until after it is in the data lake. However, a data warehouse is a centralized place for structured data to be analyzed for specific purposes related to business insights. The requirements for reporting are known ahead of time during the planning and design of a data warehouse and the ETL process. For a database, it thrives in a monolithic environment where the data is being generated by one application.

Related AWS storage services:
Amazon S3/ Amazon Redshift -- data lake;
Amazon Redshift -- data warehouse;
Amazon RDS/ Amazon DynamoDB -- database

4. Briefly expalin how SSD, HDD, Flash disk and Optical Disk are different. Separately, explain what storage Tiering is and why it is needed in Cloud. Then, identify which stoarge technology (SSD etc) is suitable for which tier. (See references). [15 points]

Both HDD and SSD devices generally offer the largest storage capacity among external options, with external HDDs offering up to 20 TB of storage and external SSDs offering up to 8 TB of storage.
For flash memory devices, one of the most recognizable type of flash memory device is the USB flash drive, which can hold up to 2 TB of storage.
For Optic Disk, CD can store up to 700 MB of data, DVD-DL can store up to 8.5 GB, and Blu-Ray can store between 25 and 128 GB of data.

Storage tiering is a strategy that lets you optimize the use of storage resources, efficiently backup data, save costs and make the best use of storage technology for each data class.

Why it is needed: for example, SSD disk drives, magnetic disk drives and tape storage. The most important or frequently-accessed data is stored on the fastest, and most expensive media (SSD) and the least important on the slowest, cheapest media (tape).The minimal storage tiering system has two tiers—one for frequently accessed data and one for archive. The more tiers are available, the more choices administrators have over the placement of specific data classes, and the more efficiently storage resources can be utilized.
As result, in Cloud, storage Tiering can reduce public cloud storage costs.

	Cold	Hot
Required Access Speed	Slow	Fast
Access Frequency	Low	High
Value of Data	Low	High
Storage Media	Slower drives, tape	Faster drives, SSD
Storage Location	May be off-premises	Colocated or fast link to the data consumer
Cost	Low cost	High cost

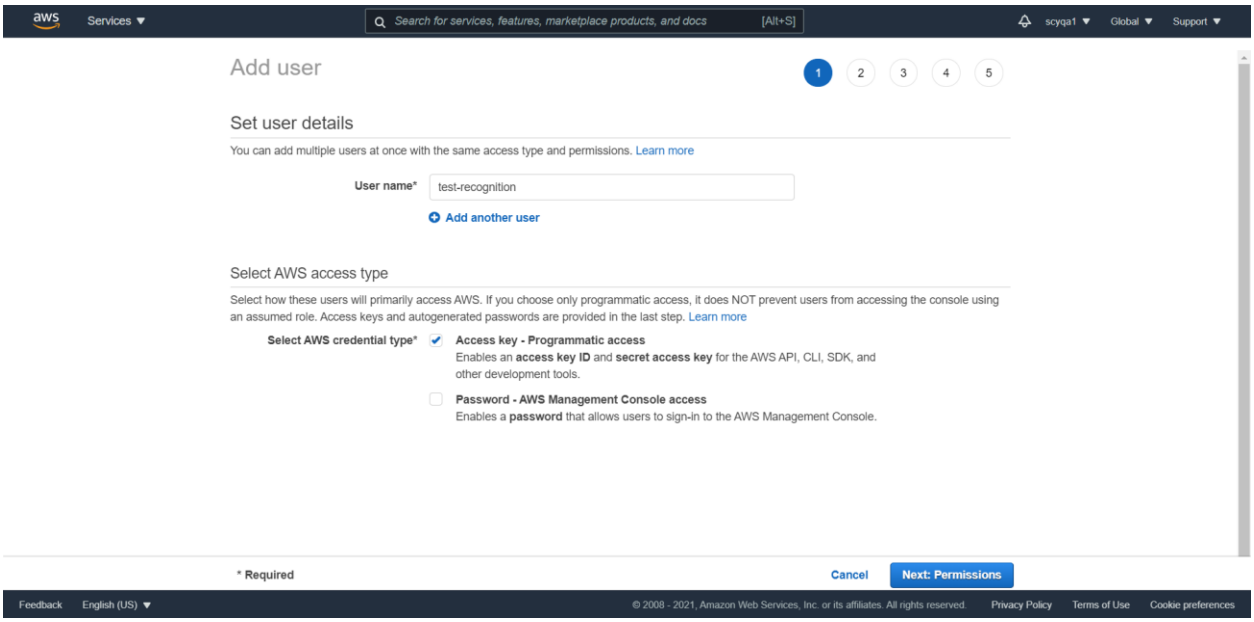
Type	Storage Media	Used For
Tier 0	Tier 0 includes SSD, RAM, PCIe Flash	You can use Tier 0 for high performance workloads.
Tier 1	Tier 1 includes fast disks, all-flash storage, hybrid flash storage	You can use Tier 1 for mission-critical or highly sensitive files.
Tier 2/3	Tier 2 and Tier 3 include Slow-spinning HDD, disk-based backup appliance, cloud storage, tape	You can use Tier 2 and Tier 3 for backups of mission critical data, which requires high reliability but not instant retrieval from backup.
Tier 4	Tier 4 include SATA drives	You can use Tier 4 for warm data, data used for periodic reporting.
Tier 5	Tier 5 includes tape storage, cloud storage archive tiers (e.g. Amazon Glacier)	You can use Tier 5 for cold data which is rarely or never accessed.

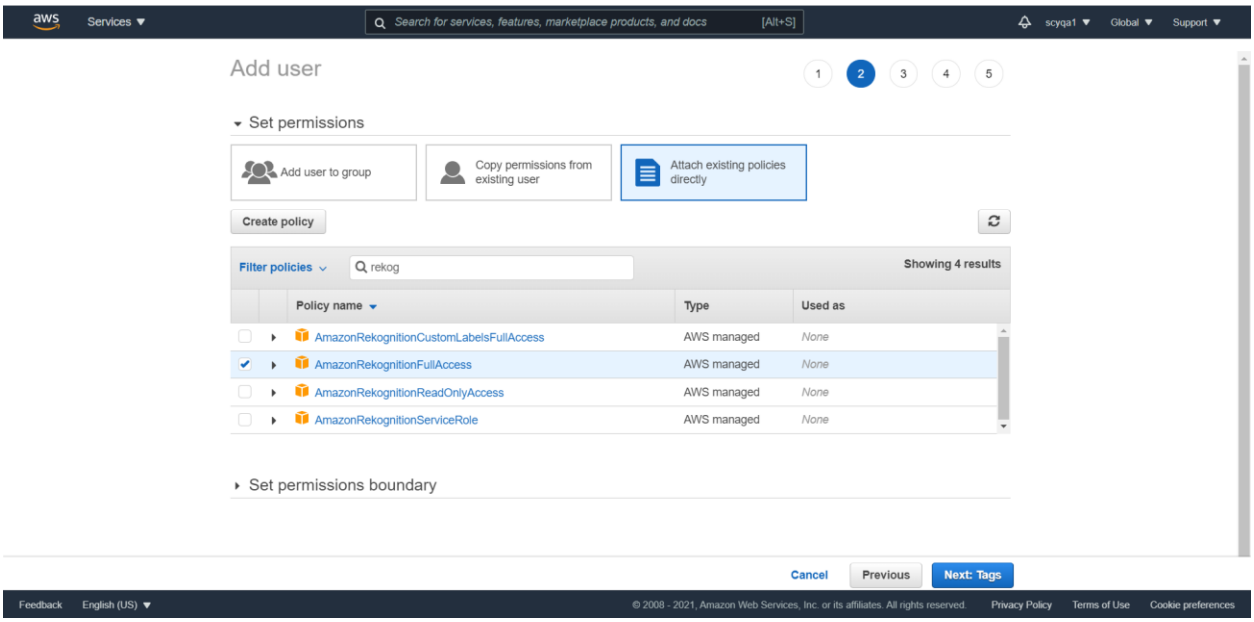
SSD is suitable for fast and high-performance tier and tape is suitable for slow and low cost tier.

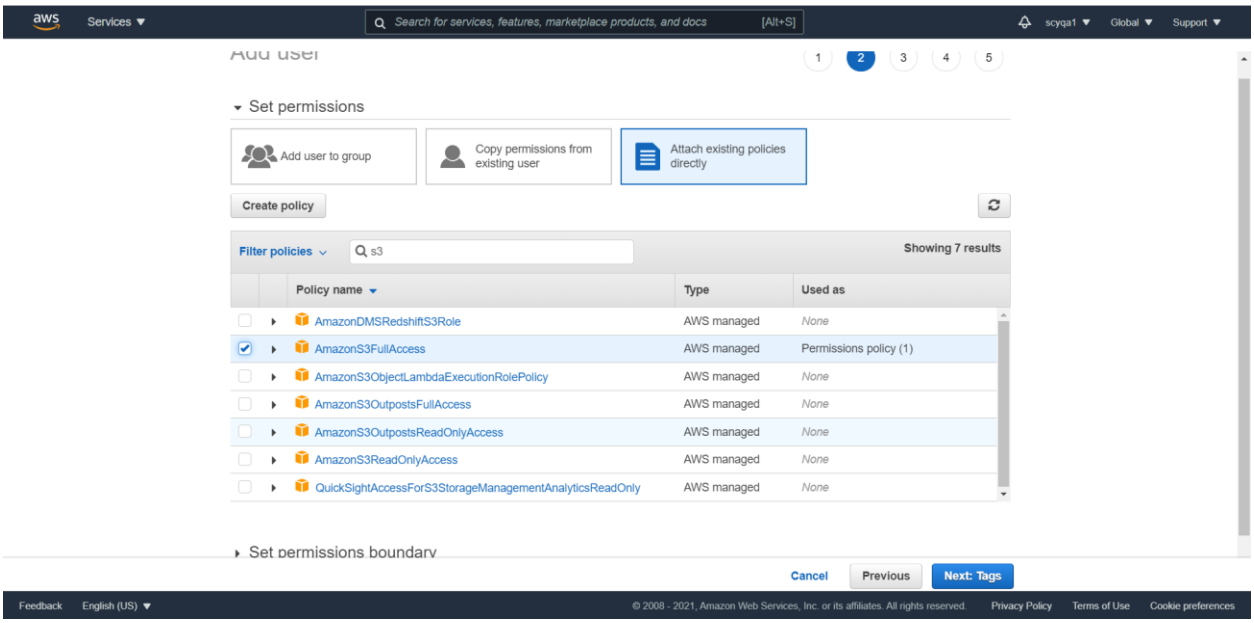
PART II: LAB [45 points]

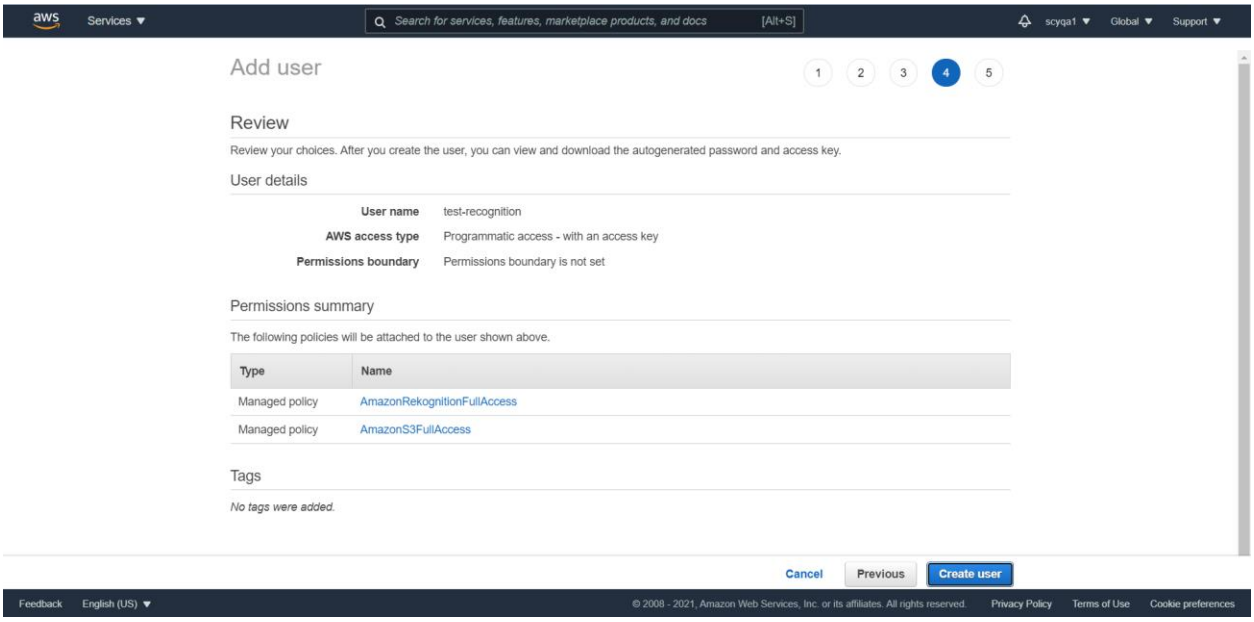
A. Amazon S3

Configuration:
Assigned role with S3 read/write permission

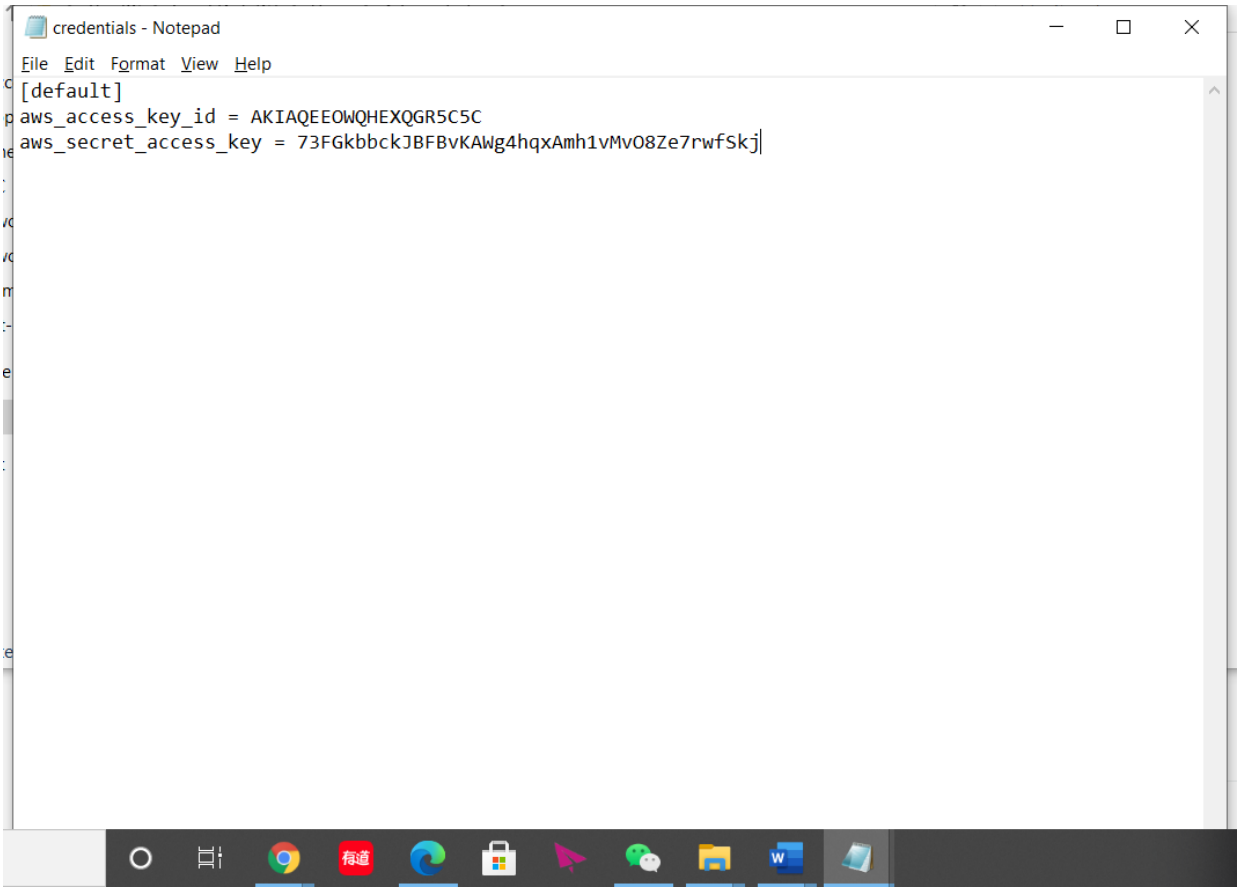


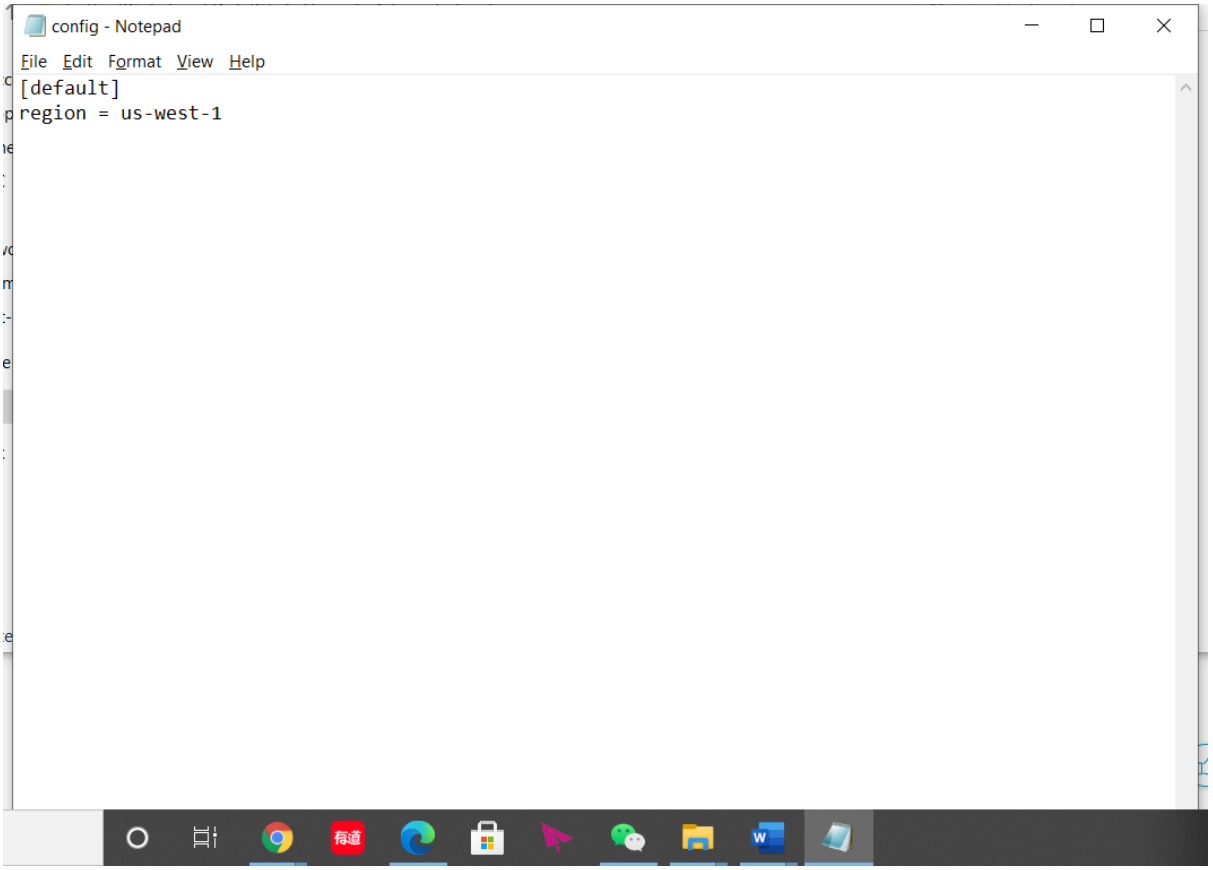



aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
Global
Support
1 2 3 4 5
Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
s3
Showing 7 results
Policy name
Type
Used as
AmazonDMSRedshiftS3Role AWS managed None
AmazonS3FullAccess AWS managed Permissions policy (1)
AmazonS3ObjectLambdaExecutionRolePolicy AWS managed None
AmazonS3OutpostsFullAccess AWS managed None
AmazonS3OutpostsReadOnlyAccess AWS managed None
AmazonS3ReadOnlyAccess AWS managed None
QuickSightAccessForS3StorageManagementAnalyticsReadOnly AWS managed None
Set permissions boundary
Cancel
Previous
Next: Tags
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences


aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
Global
Support
Add user
1 2 3 4 5
Review
Review your choices. After you create the user, you can view and download the autogenerated password and access key.
User details
User name test-recognition
AWS access type Programmatic access - with an access key
Permissions boundary Permissions boundary is not set
Permissions summary
The following policies will be attached to the user shown above.
Type
Name
Managed policy AmazonRekognitionFullAccess
Managed policy AmazonS3FullAccess
Tags
No tags were added.
Cancel
Previous
Create user
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Access_key:


credentials - Notepad
File Edit Format View Help
[default]
aws_access_key_id = AKIAQEEOWQHEXQGR5C5C
aws_secret_access_key = 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


config - Notepad
File Edit Format View Help
[default]
region = us-west-1

Upload the image that contains text to S3 bucket.

Upload:

```
[root@ip-172-31-10-156 s3]# aws s3 cp ./ s3://project2-image/ --recursive
upload: ./imageFile.jpg to s3://project2-image/imageFile.jpg
```

Library:

```
aqc@ubuntu:~/aqc/lambd/s3$ pip install boto3
```

```
aqc@ubuntu:~/aqc/lambd/s3$ pip install pandas
```

Print a list of all the buckets:

aqc@ubuntu: ~/aqc/lambd/s3

```
# Creating the low level functional client
client = boto3.client(
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Creating the high level object oriented interface
resource = boto3.resource(
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)


# Fetch the list of existing buckets
clientResponse = client.list_buckets()

# Print the bucket names one by one
print('Printing bucket names...')
for bucket in clientResponse['Buckets']:
                                                    17,0-1        Top
```

```
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 s3.py
Printing bucket names...
Bucket Name: codepipeline-us-west-1-971689687851
Bucket Name: project2-image
Bucket Name: zappa-pgk1rhhn5
Bucket Name: zappa-rw3sg9dt9
Bucket Name: zappa-tn4u6j73t
```

Create a bucket

root@ubuntu: /home/aqc/aqc/lambd/s3

```
# Creating the low level functional client
client = boto3.client(
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Creating the high level object oriented interface
resource = boto3.resource(
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Creating a bucket in AWS S3
location = {'LocationConstraint': 'us-west-1'}
client.create_bucket(
    Bucket='hw5-demo-1',
    CreateBucketConfiguration=location
)
```

-- INSERT -- 26,1 Bot

```
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 s3create.py
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 s3.py
Printing bucket names...
Bucket Name: codepipeline-us-west-1-971689687851
Bucket Name: hw5-demo-1
Bucket Name: project2-image
Bucket Name: zappa-pgk1rhhn5
Bucket Name: zappa-rw3sg9dt9
Bucket Name: zappa-tn4u6j73t
```

Read form S3

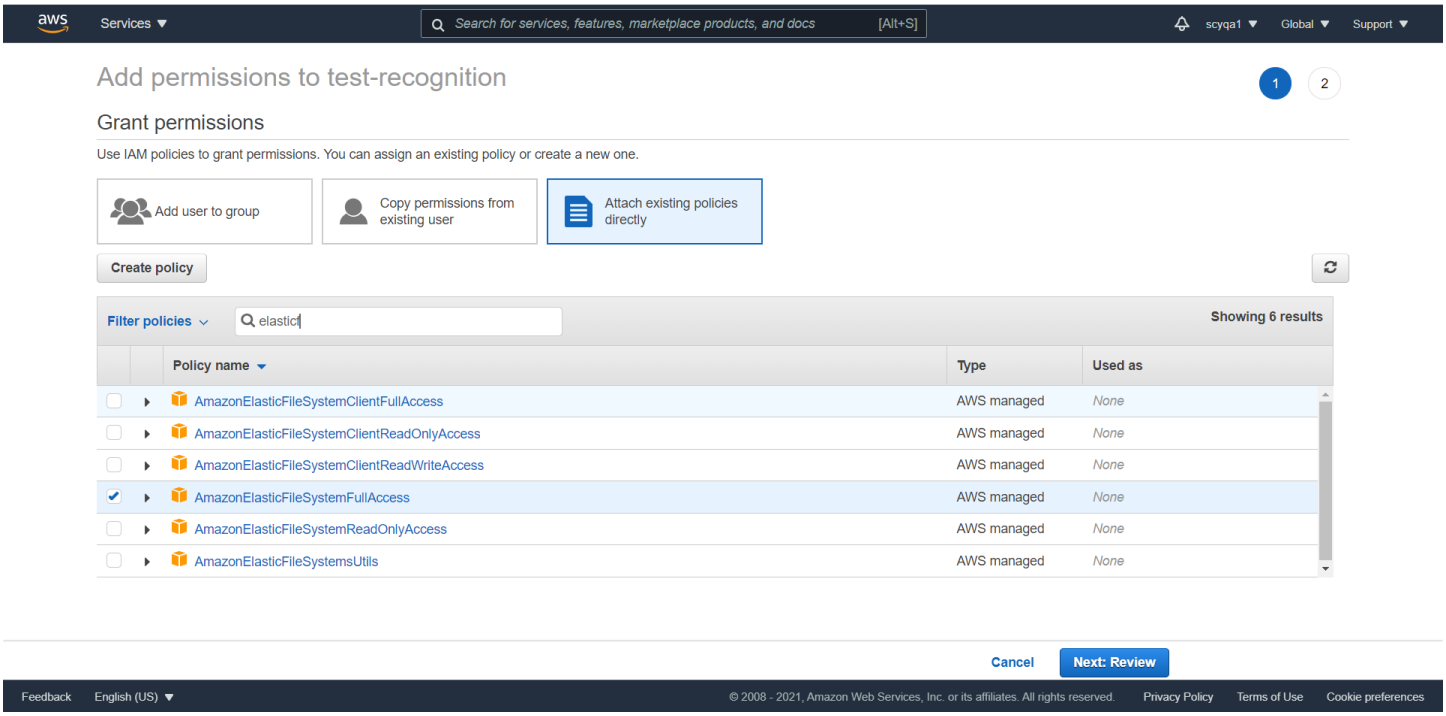
root@ubuntu: /home/aqc/aqc/lambd/s3

```
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Creating the high level object oriented interface
resource = boto3.resource(
    's3',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Create the S3 object
obj = client.get_object(
    Bucket = 'project2-image',
    Key = 'static/text'
)

# Read data from the S3 object
data = pandas.read_csv(obj['Body'])

"s3read.py" 31L, 752C                                    23,19         62%
```

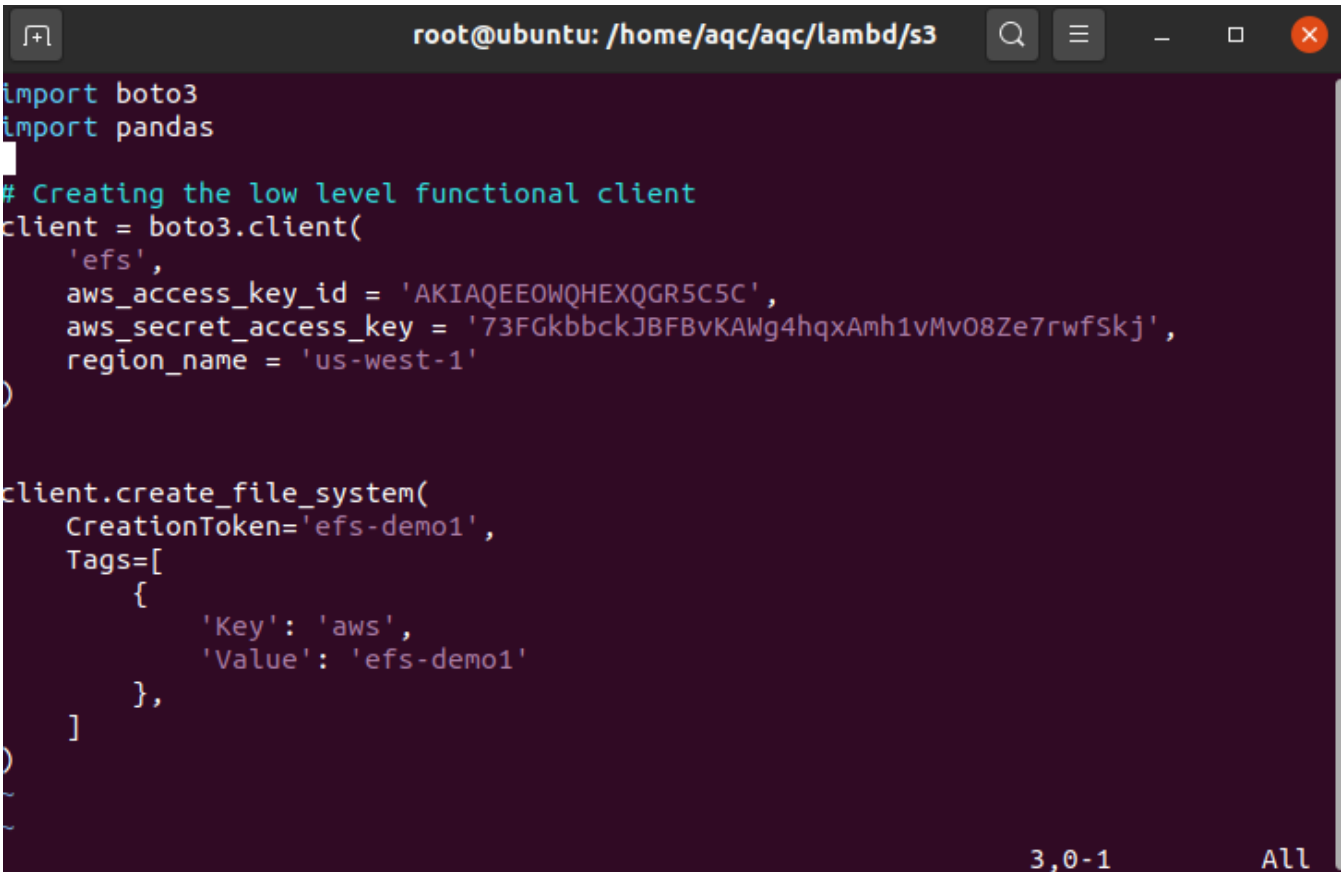
```
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 s3read.py
Printing the data frame...
                     Detected text
0                       ----------
1    Detected text:Be the Auther of
2                Confidence: 65.88%
3                             Id: 0
4                         Type:LINE
..                              ...
138             Confidence: 100.00%
139                           Id: 29
140                    Parent Id: 8
141                       Type:WORD
142                Text detected: 30

[143 rows x 1 columns]
```

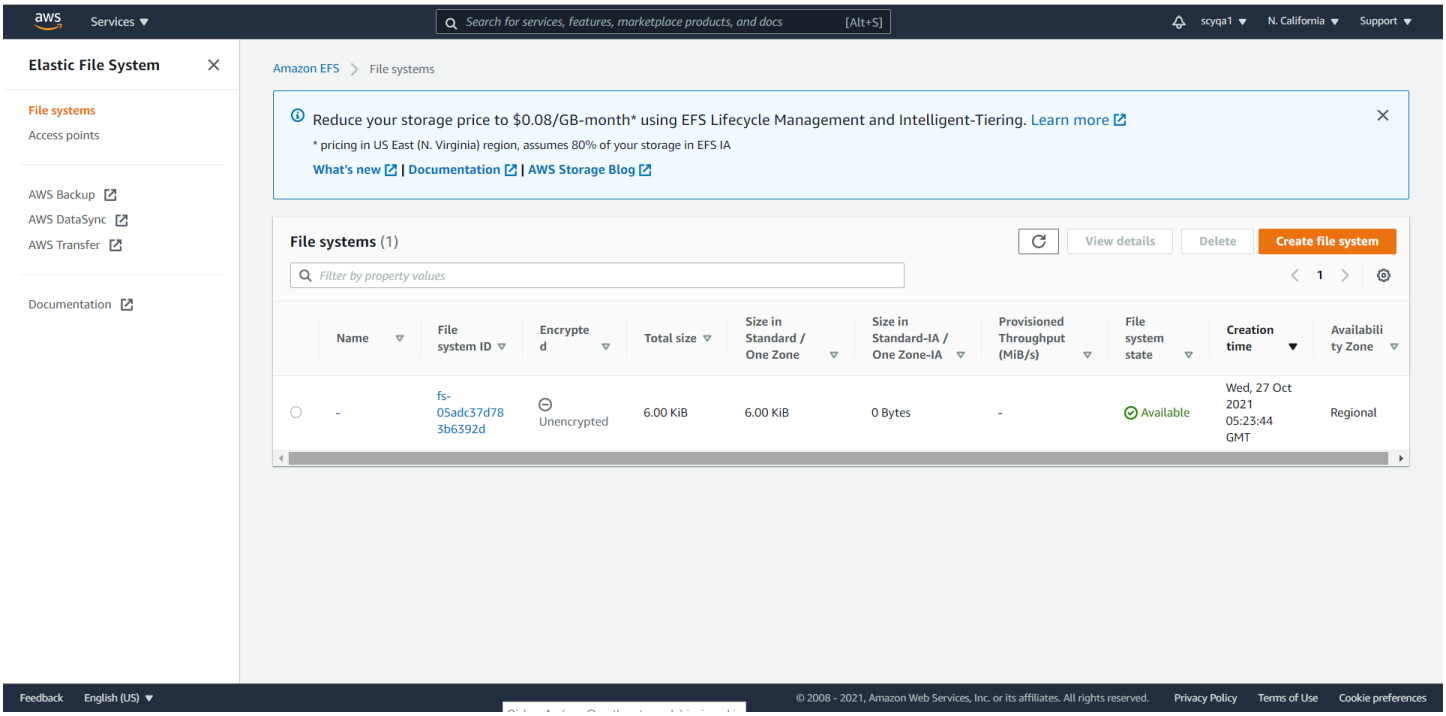
B. Amazon EFS

Configuration:
Assigned role with EFS read/write permission

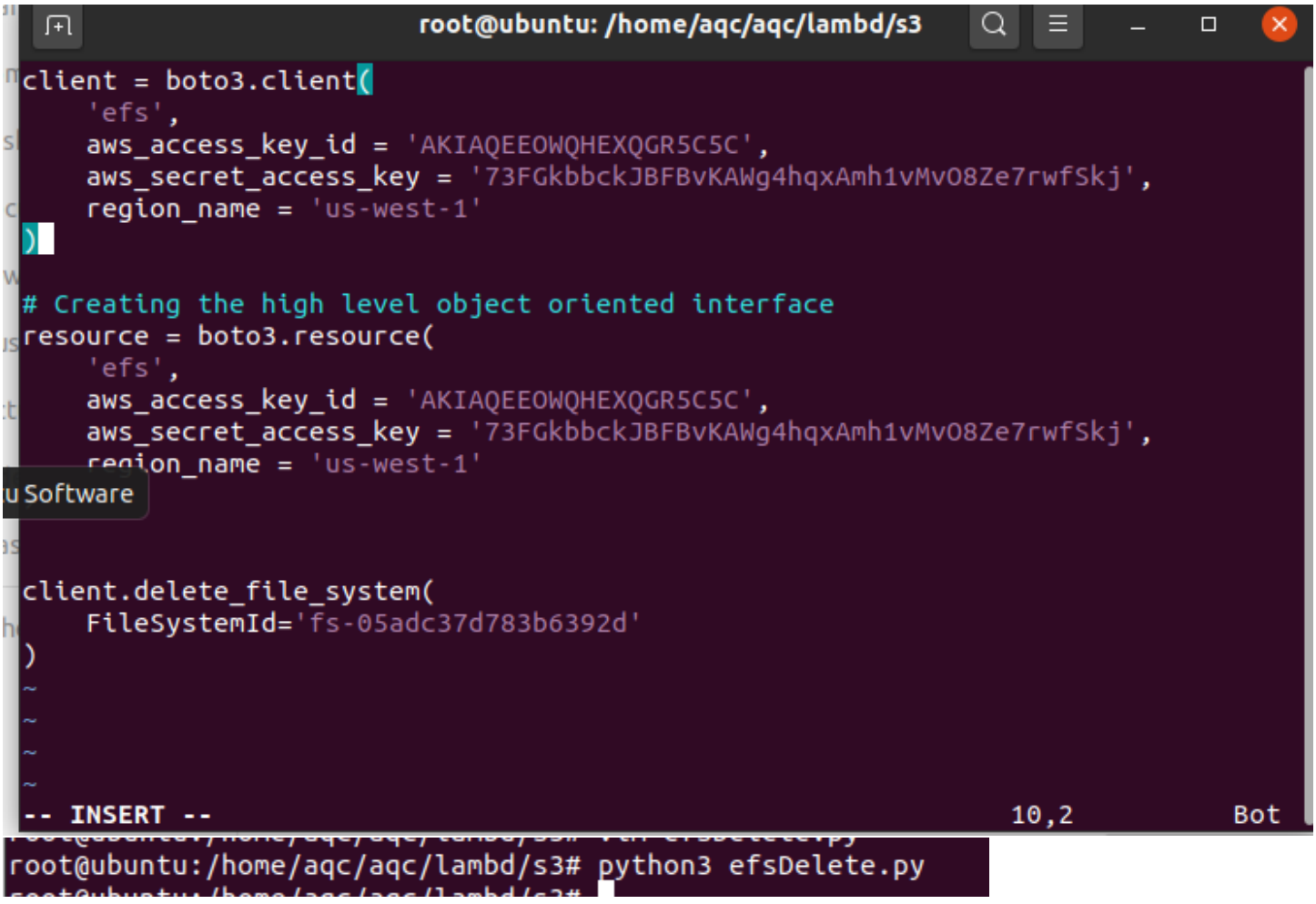


Create EFS:



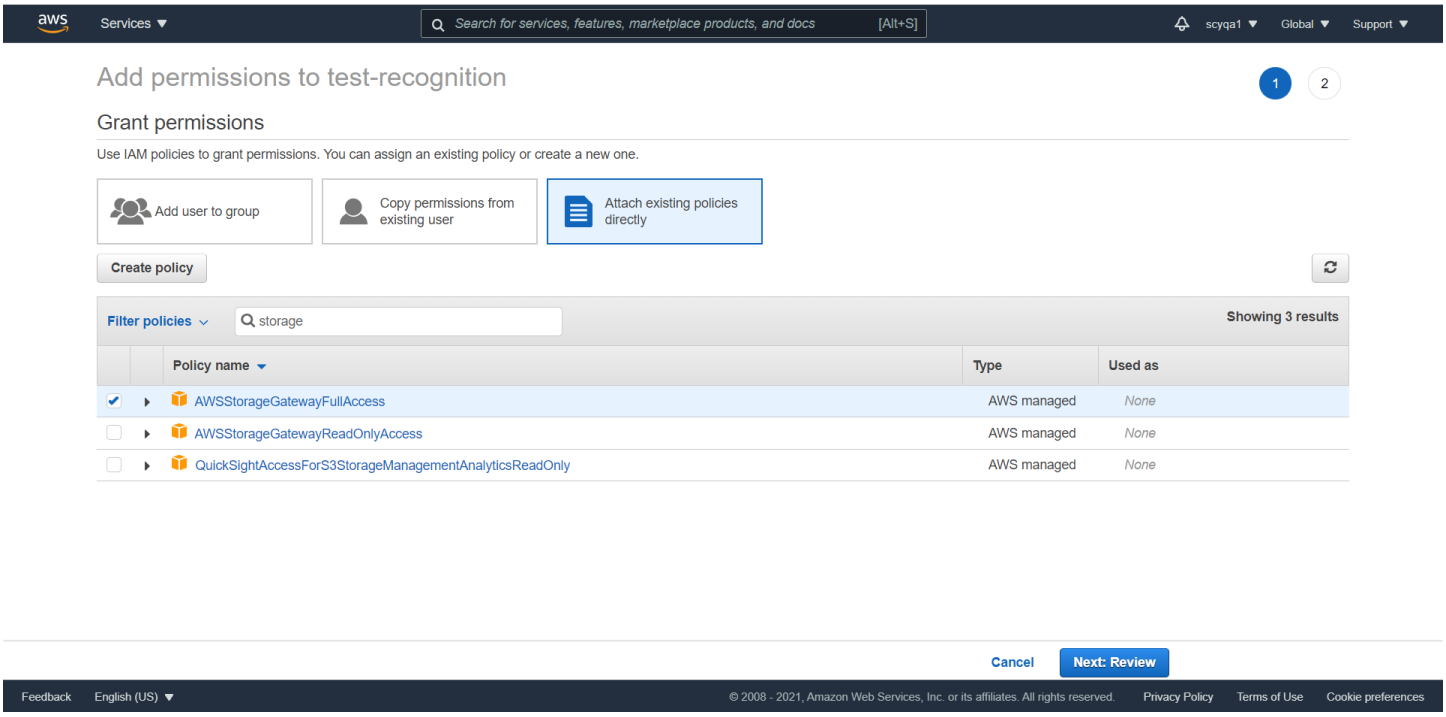

aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
scyqa1
N. California
Support
Elastic File System
File systems
Access points
AWS Backup
AWS DataSync
AWS Transfer
Documentation
Amazon EFS > File systems
Reduce your storage price to $0.08/GB-month* using EFS Lifecycle Management and Intelligent-Tiering. Learn more
* pricing in US East (N. Virginia) region, assumes 80% of your storage in EFS IA
What's new | Documentation | AWS Storage Blog
File systems (1)
View details
Delete
Create file system
Filter by property values
Name
File system ID
Encrypted
Total size
Size in Standard / One Zone
Size in Standard-IA / One Zone-IA
Provisioned Throughput (MiB/s)
File system state
Creation time
Availability Zone
-
fs-05adc37d783b6392d
Unencrypted
6.00 KiB
6.00 KiB
0 Bytes
-
Available
Wed, 27 Oct 2021 05:23:44 GMT
Regional
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

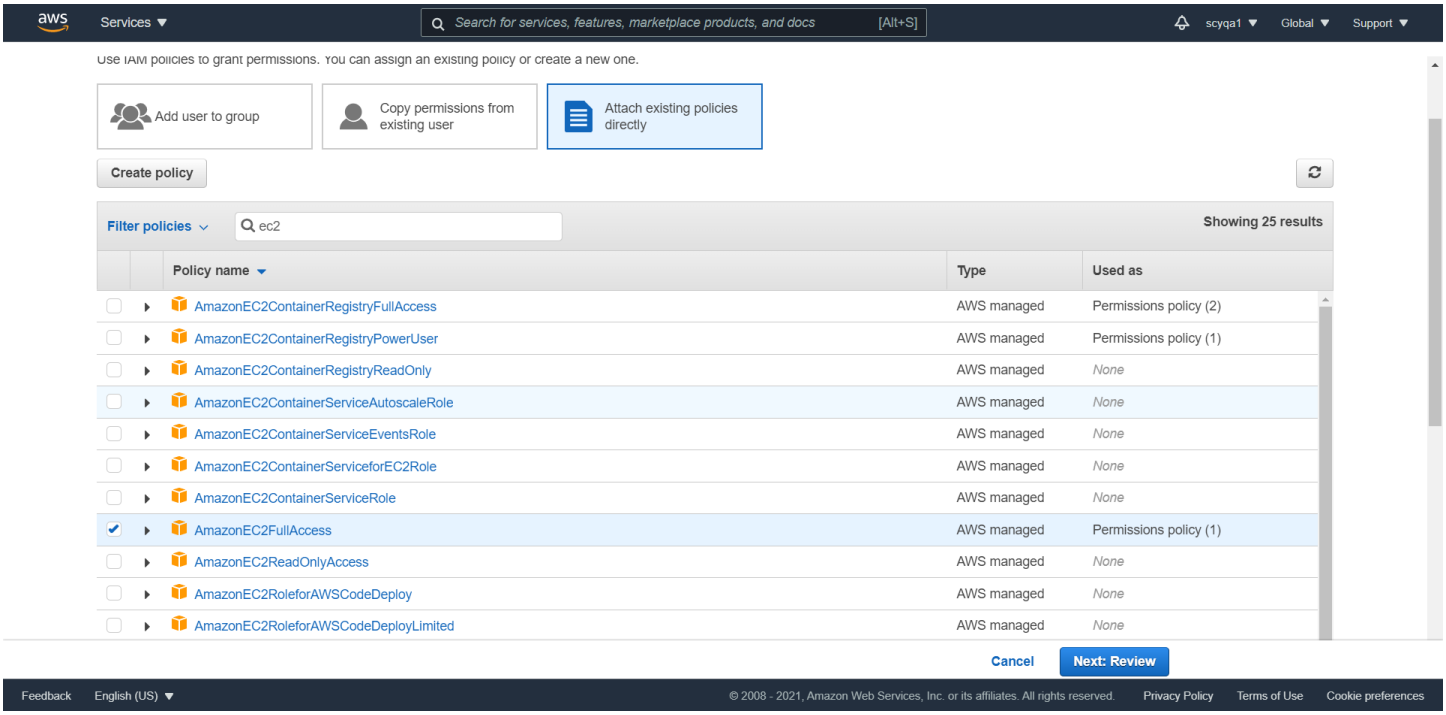
Delete EFS:


root@ubuntu: /home/aqc/aqc/lambd/s3
client = boto3.client(
'efs',
aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
aws_secret_access_key = '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',
region_name = 'us-west-1'
)
# Creating the high level object oriented interface
resource = boto3.resource(
'efs',
aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
aws_secret_access_key = '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',
region_name = 'us-west-1'
Software
client.delete_file_system(
FileSystemId='fs-05adc37d783b6392d'
)
-- INSERT --
10,2
Bot
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 efsDelete.py

C. Amazon EBS

Configuration:
Assigned role with EBS read/write permission



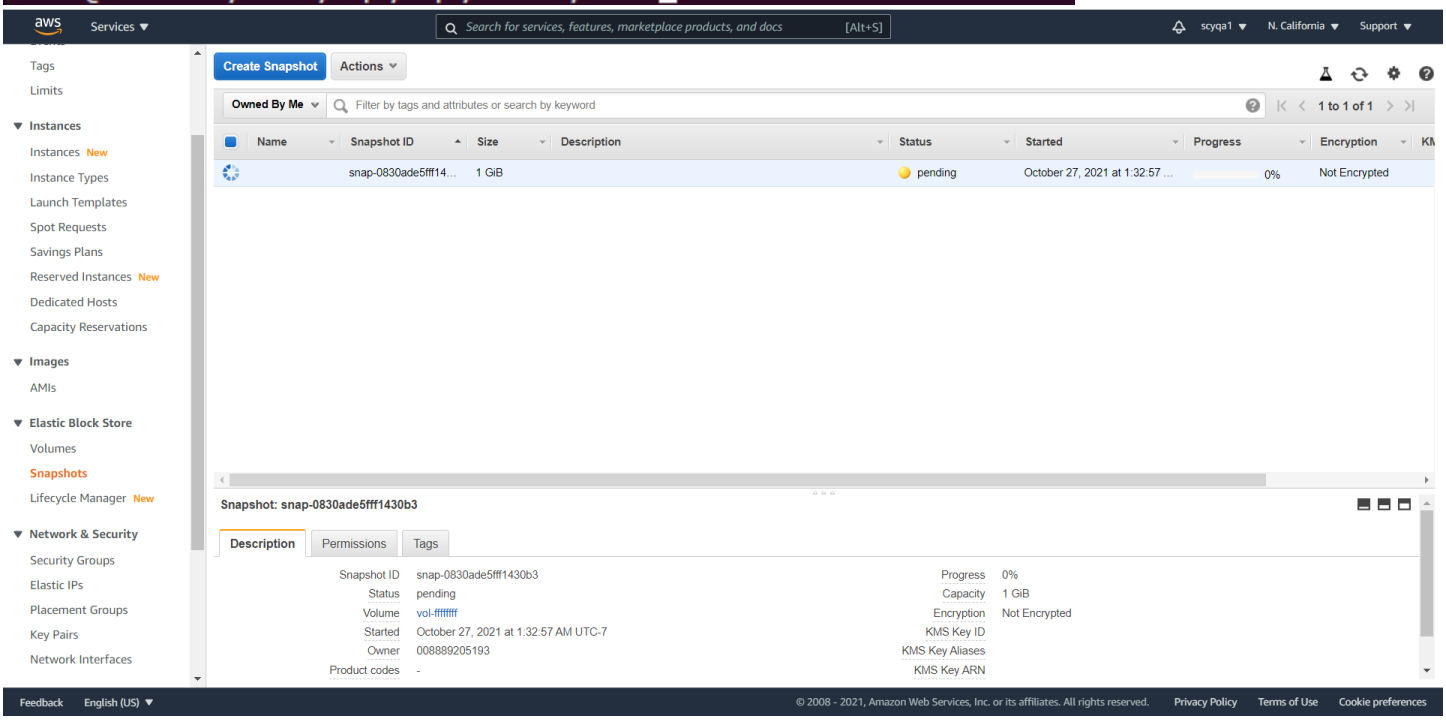


Create snapshot:

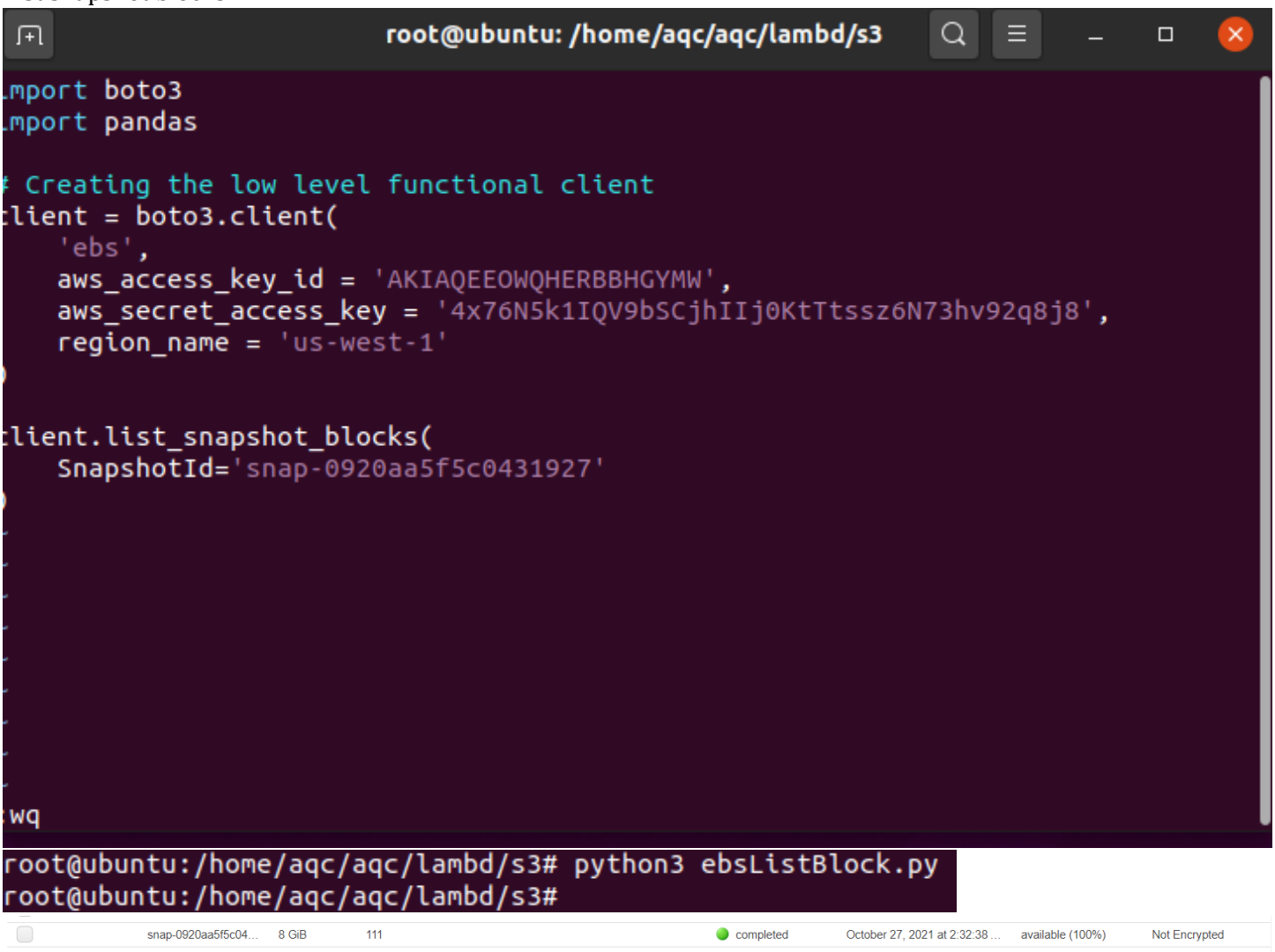
root@ubuntu: /home/aqc/aqc/lambd/s3

```
import boto3
import pandas

# Creating the low level functional client
client = boto3.client(
    'ebs',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

# Creating the high level object oriented interface
resource = boto3.resource(
    'ebs',
    aws_access_key_id = 'AKIAQEEOWQHEXQGR5C5C',
    aws_secret_access_key = '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',
    region_name = 'us-west-1'
)

client.start_snapshot(
    VolumeSize=1,
    ParentSnapshotId='ebs-demo'
)
-- INSERT --                                    3,1           Top
```

```
root@ubuntu:/home/aqc/aqc/lambd/s3# python3 ebsCreate.py
root@ubuntu:/home/aqc/aqc/lambd/s3#
```



List snapshot blocks:



Complete snapshot: