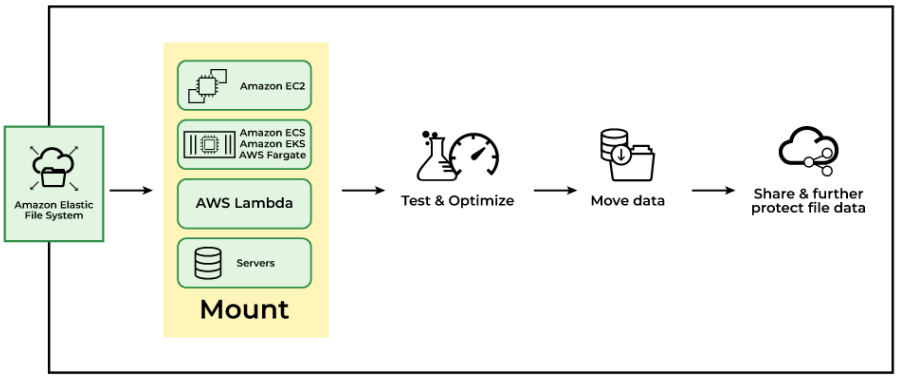
EFS:

What is AWS Elastic File System?

From the aforementioned list, EFS falls under the file storage category. EFS is a file-level, fully managed storage provided by AWS (Amazon Web Services) that can be accessed by multiple EC2 instances concurrently. Just like the AWS EBS, EFS is specially designed for high throughput and low latency applications.

How Does EFS Work?

EFS can be created using the EC2-Instance, which will be created in a specific region and distributed across multiple availability zones for the purpose of high availability and durability. You can choose the EFS based on the I/Ops you are going to perform.



1. **Metadata**

- **The following information about the Amazon EFS file systems is collected:**

Attribute	**Description**
File system name	**The unique name of the file system.**
File system ID	**ID of the file system is assigned by AWS.**
Region	**The Region where the file system is deployed.**
Created time	**Time that the file system was created.**
Creation token	**Sting of up to 64 ASCII characters. Amazon uses this to ensure idempotent**

	creation.
Encrypted	**Indicates whether the file system is encrypted. Possible values true or false.**
LifeCycleState	**Lifecycle phase of the file system.**
KMS Key ID	**The ID of the AWS Key Management Service CMK that was used to encrypt the file system.**
Mount Targets	**Current number of mount targets that the file system has.**
Owner ID	**AWS account the created the file system.**
Performance Mode	**Indicates the Performance Mode of the file system. Possible values are General Purpose and Max I/O.**

https://www.site24x7.com/help/aws/efs-integration.html

Example of metadata:

```
[cloudshell-user@ip-10-140-112-142 ~]$ aws efs describe-file-systems --file-system-id fs-04c08d081c3689313
{
    "FileSystems": [
        {
            "OwnerId": "381491918157",
            "CreationToken": "quickCreated-a5a26b5d-613f-4e9c-8a6a-e55ba2503379",
            "FileSystemId": "fs-04c08d081c3689313",
            "FileSystemArn": "arn:aws:elasticfilesystem:us-east-1:381491918157:file-system/fs-04c08d081c3689313",
            "CreationTime": "2024-07-02T21:14:48+00:00",
            "LifeCycleState": "available",
            "Name": "metadata",
            "NumberOfMountTargets": 5,
            "SizeInBytes": {
                "Value": 6144,
                "ValueInIA": 0,
                "ValueInStandard": 6144,
                "ValueInArchive": 0
            },
            "PerformanceMode": "generalPurpose",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:381491918157:key/9209e632-03df-48c3-8919-c568226be321",
            "ThroughputMode": "elastic",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "metadata"
                },
                {
                    "Key": "aws:elasticfilesystem:default-backup",
                    "Value": "enabled"
                }
            ],
            "FileSystemProtection": {
                "ReplicationOverwriteProtection": "ENABLED"
            }
        }
    ]
}
[cloudshell-user@ip-10-140-112-142 ~]$
```

2. **Tag**

a. **Minimum allowed score**
b. **Desired Score**
c. **Department**
d. **Environment**
    i. **What phase is the data in, for example, testing or development**

3. **Binary score from another AWS service**

Instruction clarification: each team member should come up with one additional input for their designated storage resource

4. ~~**Data naming conventions(Michael Huynh)**~~

    a. ~~**Naming conventions refer to a set of rules and guidelines used in naming various entities within a system or project. These conventions ensure consistency, clarity, and manageability of names, making it easier for developers, administrators, and users to understand and work with the system.**~~

    b. ~~**My additional input for my storage resource will be data naming conventions. An example of data naming conventions: "May_product_data.csv" will retain the same score, while "xergtadsgeg34gdfl.csv" will receive a lowered score.**~~

5. **Access frequency**
    a. **How often is the data accessed? Less accessed data will have a high score, and frequently accessed data will have a lowered score.**
    b. **Use can use CloudTrail or CloudWatch**

OTHER INPUTS: THIS IS WHERE YOU TRY TO BE AS CREATIVE AS POSSIBLE IN DEVELOPING INPUTS

Idea from Michael Huynh

- Set up AWS CloudWatch Monitoring

- Collect Data to establish a baseline for normal activiteis

- Monitor for unusual access pattern, example maybe a unusual amount of write operations,
The normal write request per a day from a user is 100 but you detect a user with 1000+ write request

- Amazon EFS supports POSIX for security

- If monitor detects anomaly, reduce the user's permission

- Assign POSIX Permissions: Score 1 = read only, Score 2 = read-write, Score 3 = Execute

- Use AWS SDK to update user's permissions

https://docs.google.com/spreadsheets/d/1ylRFG3jvFaF3iFovaglit2zb3WqwLlP3jKZqKCPHwfw/edit?usp=sharing

---

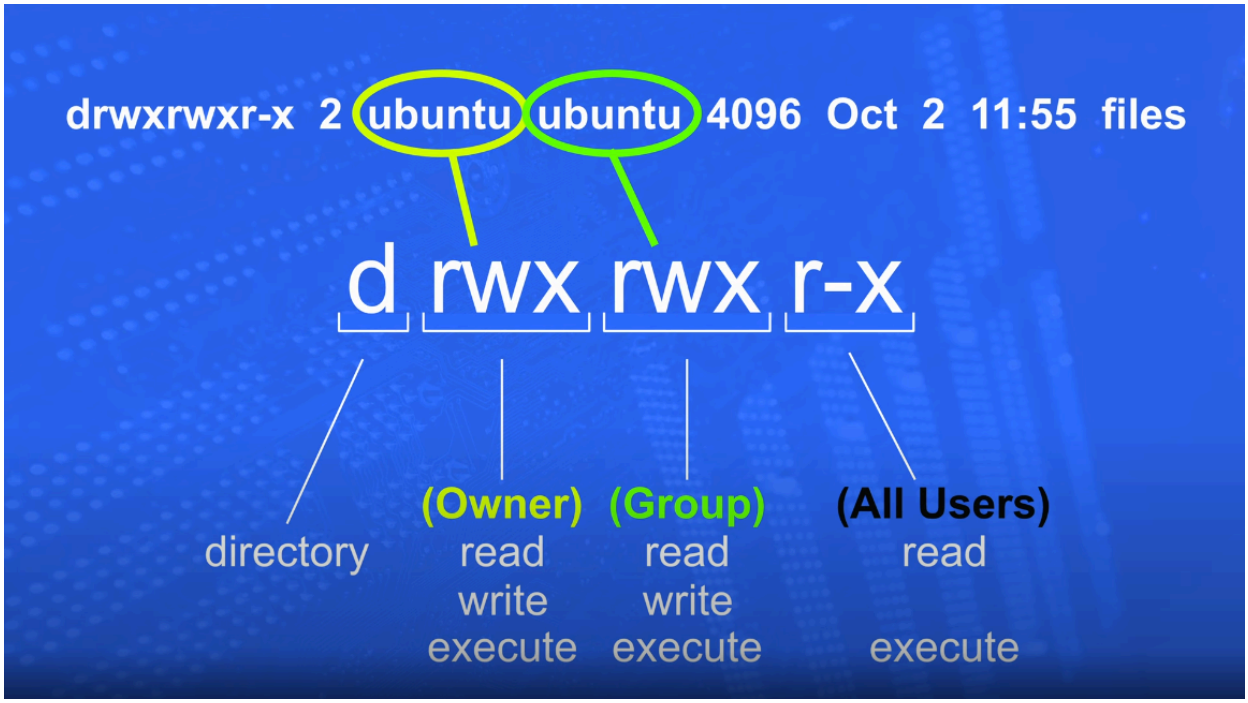
What is Posix?

The traditional POSIX file system object permission model defines three classes of users called owner, group, and other. Each of these classes is associated with a set of permissions. The permissions defined are read (r), write (w), and execute (x).

https://www.usenix.org/legacy/publications/library/proceedings/usenix03/tech/freenix03/full_papers/gruenbacher/gruenbacher_html/main.html#:~:text=The%20traditional%20POSIX%20file%20system,%2C%20and%20execute%20(x).

Owner groups:

- **Owner**
- **Group**
- **Others**



d = directory
– = file

Use Posix with ACL (Access Control Lists)

ACLs equivalent to the file mode permission bits are called minimal ACLs. They have three ACL entries. ACLs with more than three entries are called extended ACLs. Extended ACLs also contain a mask entry and may contain any number of named user and named group entries.

Table: Types of ACL Entries
Entry type Text form

Owner user::rwx
Named user user:name:rwx
Owning group group::rwx
Named group group:name:rwx
Mask mask::rwx
Others other::rwx

Access points

Amazon EFS access points are application-specific entry points into an EFS file system that make it easier to manage application access to shared datasets. Access points can enforce a user identity, including the user's POSIX groups, for all file system requests that are made through the access point. Access points can also enforce a different root directory for the file system so that clients can only access data in the specified directory or its subdirectories.

You can use AWS Identity and Access Management (IAM) policies to enforce that specific applications use a specific access point. By combining IAM policies with access points, you can easily provide secure access to specific datasets for your applications.

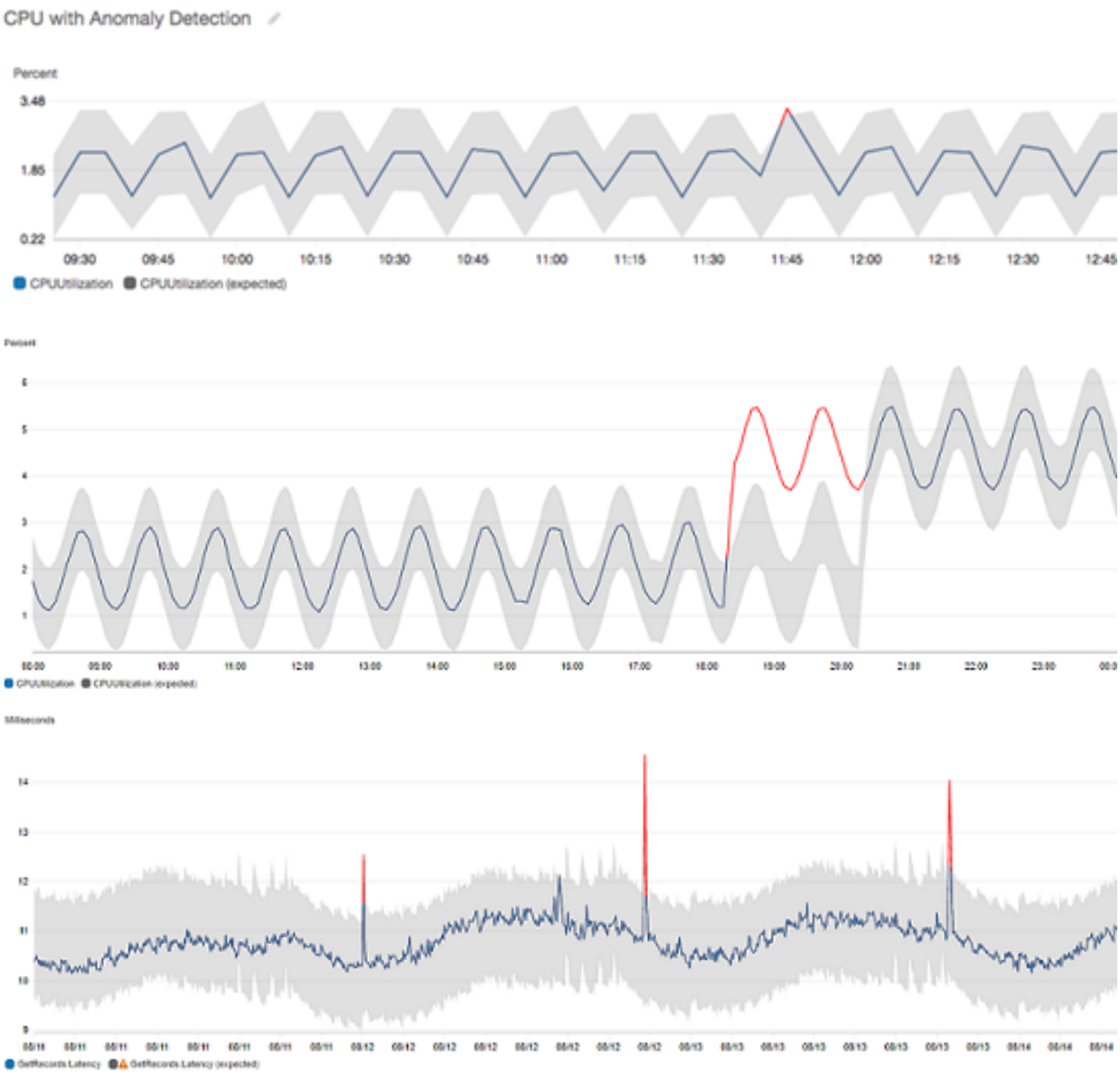
https://docs.aws.amazon.com/efs/latest/ug/efs-access-points.html#:~:text=Amazon%20EFS%20access%20points%20are,application%20access%20to%20shared%20datasets.

Using CloudWatch anomaly detection:

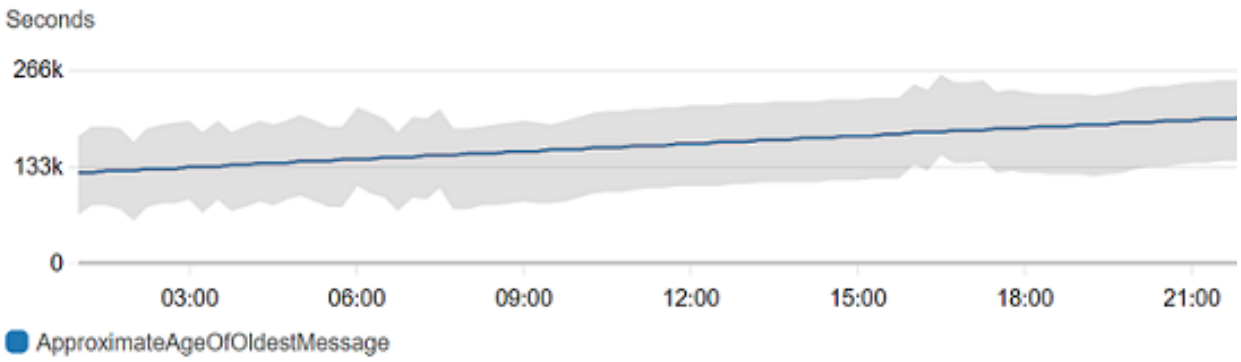
When you enable anomaly detection for a metric, CloudWatch applies statistical and machine learning algorithms. These algorithms continuously analyze metrics of systems and applications, determine normal baselines, and surface anomalies with minimal user intervention.

The algorithms generate an anomaly detection model. The model generates a range of expected values that represent normal metric behavior.

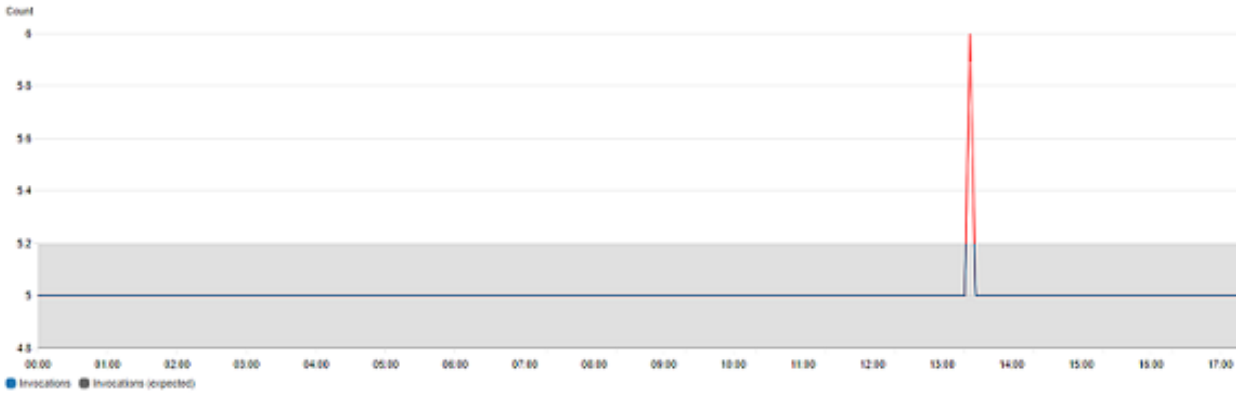
Anomaly detection algorithms account for the seasonality and trend changes of metrics. The seasonality changes could be hourly, daily, or weekly, as shown in the following examples.



The longer-range trends could be downward or upward.



Anomaly detections also works well with metrics with flat patterns.



<u>What is artificial intelligence?</u>

- **Artificial intelligence is a broad field that refers to the use of technologies to build machines and computers that have the ability to mimic cognitive functions associated with human intelligence, such as being able to see, understand, and respond to spoken or written language, analyze data, make recommendations, and more.**

- **Although artificial intelligence is often thought of as a system in itself, it is a set of technologies implemented in a system to enable it to reason, learn, and act to solve a complex problem.**

<u>What is machine learning?</u>

- **Machine learning is a subset of artificial intelligence that automatically enables a machine or system to learn and improve from experience. Instead of explicit programming, machine learning uses algorithms to**

analyze large amounts of data, learn from the insights, and then make informed decisions.

- **Machine learning algorithms improve performance over time as they are trained—exposed to more data. Machine learning models are the output, or what the program learns from running an algorithm on training data. The more data used, the better the model will get.**

https://cloud.google.com/learn/artificial-intelligence-vs-machine-learning

How can we utilize anomaly data and tags to detect suspicious behavior?

For instance, when the user's location tag shows a sudden change from the state of Virginia to California, the model can identify this as potentially malicious behavior. Anomaly detection will be performed using CloudWatch, and the model will be built using bedrock.

What I did was I used access points to control the user's permissions to the EFS, used Cloudwatch to detect anomalies, and then used anomalies, tags, and other metadata to generate a score for the user. Then, store that user score in a database and update the user's permission in the access point.

My tags:

role	department	compliance
environment	owner	access point
project	application	Backup
Encryption	Performance mode	Throughput mode

Sensitivity score:

Posix permission of the file

And Something like macie to scan the file to determine the sensitive score