Assignment 1: AWS Key Management Services Implementation

Report by: Tonny Odhiambo, CS-CNS06-24028

Introduction

In today's digital landscape, ensuring the security of sensitive data is paramount for any organization. AWS Key Management Service (KMS) provides a robust solution for creating and controlling the cryptographic keys that safeguard our data. This report outlines the implementation of AWS KMS in a practical lab setting.

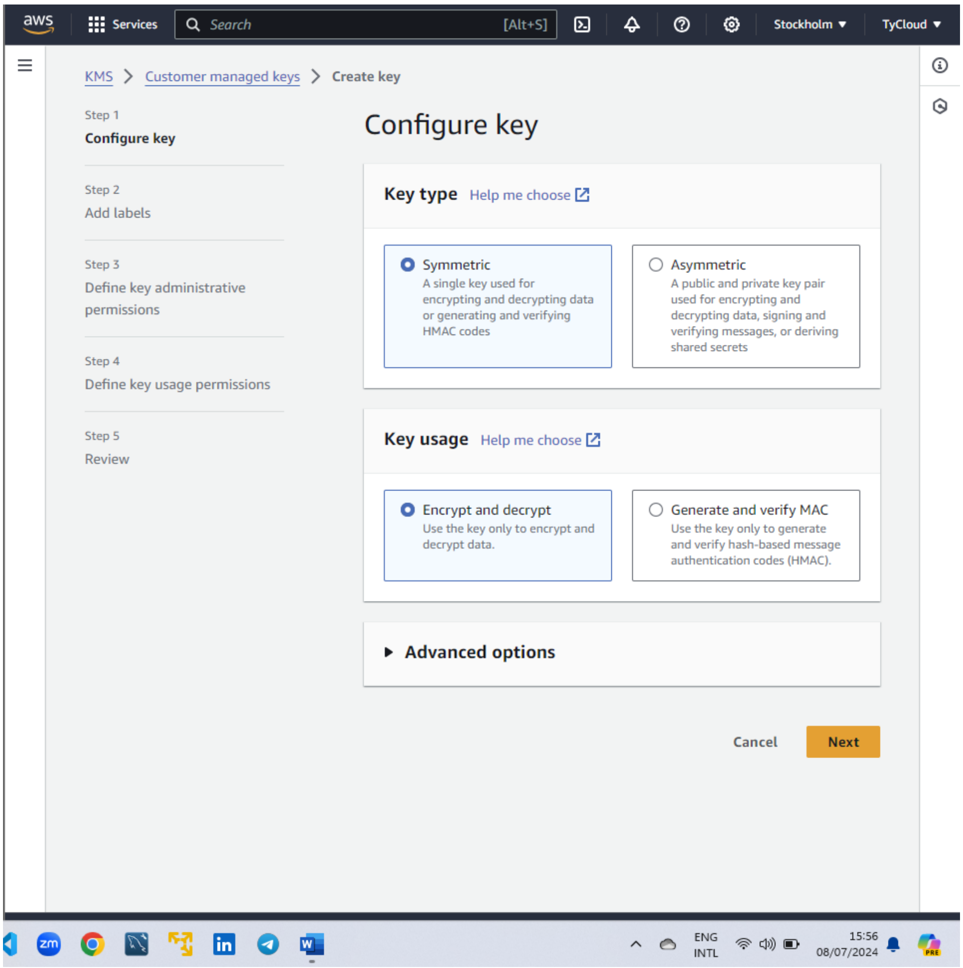
Through this lab, I aim to understand the core functionalities of AWS KMS, including key creation, management, and integration with other AWS services. By mastering these skills, we can enhance our organization's data security posture, ensuring that our information remains protected against unauthorized access and breaches**.**

Part 1: Creation and configuration of the key.

Step 1: Configuration of the Key

I logged into my AWS console and headed to the Key Management Service to create and configure the key.

In this step, I maintained the default settings on the key type and key usage.



Step 1
Configure key

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Configure key

Key type Help me choose

● Symmetric
A single key used for encrypting and decrypting data or generating and verifying HMAC codes

○ Asymmetric
A public and private key pair used for encrypting and decrypting data, signing and verifying messages, or deriving shared secrets

Key usage Help me choose

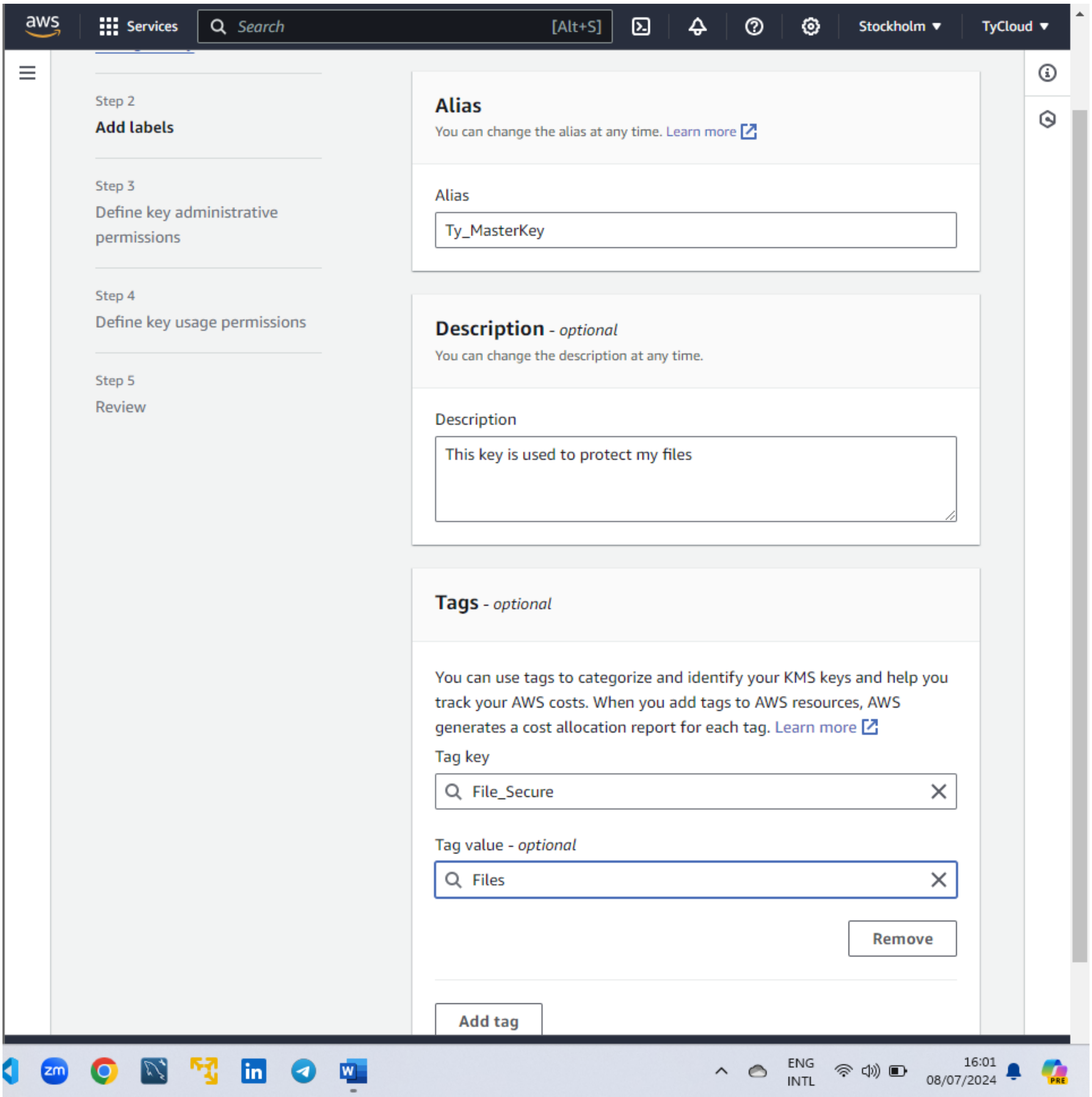
● Encrypt and decrypt
Use the key only to encrypt and decrypt data.

○ Generate and verify MAC
Use the key only to generate and verify hash-based message authentication codes (HMAC).

Step 2: Addition of alias, description and tags

In this second step, I added an alias as the name of my master key. I also added a description to describe what the key does.

In this step I have also added a **Tag key** and **Tag value** to further help identify the master key.



Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Alias

You can change the alias at any time. Learn more

Alias: Ty_MasterKey

Description - *optional*

You can change the description at any time.

Description: This key is used to protect my files

Tags - *optional*

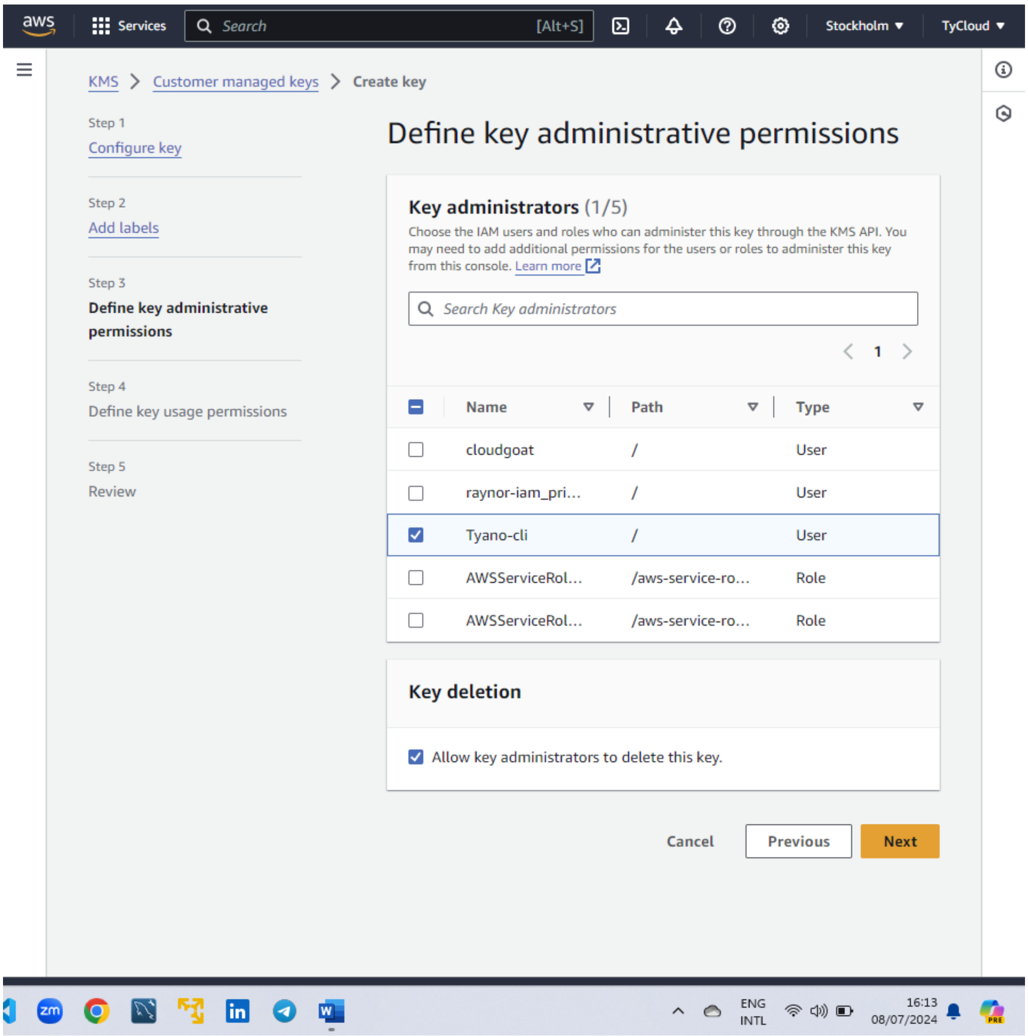
You can use tags to categorize and identify your KMS keys and help you track your AWS costs. When you add tags to AWS resources, AWS generates a cost allocation report for each tag. Learn more

Tag key: File_Secure

Tag value - *optional*: Files

Step 3: Defining key administrative permissions.

In this step, I assign key permissions to the IAM user. I have selected the user **Tyano-cli**



KMS > Customer managed keys > Create key

Step 1
Configure key

Step 2
Add labels

Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

Define key administrative permissions

Key administrators (1/5)

Choose the IAM users and roles who can administer this key through the KMS API. You may need to add additional permissions for the users or roles to administer this key from this console. Learn more

Search Key administrators

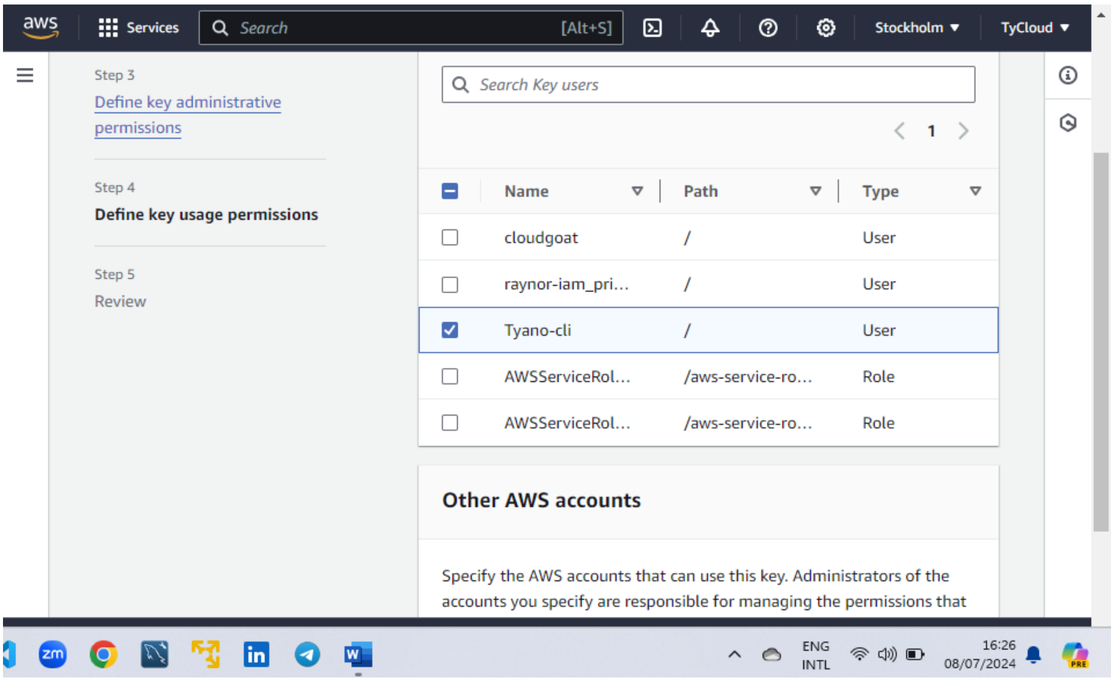
	Name	Path	Type
□	cloudgoat	/	User
□	raynor-iam_pri…	/	User
☑	Tyano-cli	/	User
□	AWSServiceRol…	/aws-service-ro…	Role
□	AWSServiceRol…	/aws-service-ro…	Role

Key deletion

☑ Allow key administrators to delete this key.

Step 4: Defining key usage permissions

In this step, I give the user key permission.



Step 3
Define key administrative permissions

Step 4
Define key usage permissions

Step 5
Review

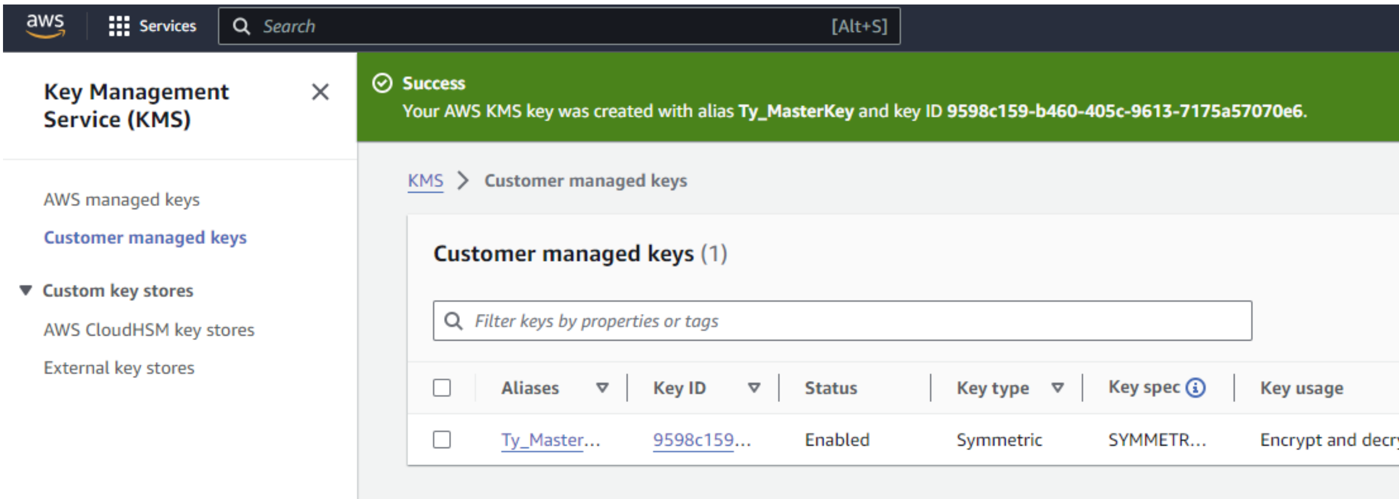
Search Key users

	Name	Path	Type
□	cloudgoat	/	User
□	raynor-iam_pri…	/	User
☑	Tyano-cli	/	User
□	AWSServiceRol…	/aws-service-ro…	Role
□	AWSServiceRol…	/aws-service-ro…	Role

Other AWS accounts

Specify the AWS accounts that can use this key. Administrators of the accounts you specify are responsible for managing the permissions that

After review, I clicked finish and successfully created an AWS KMS for the IAM user Tyano-cli



Key Management Service (KMS)

Success
Your AWS KMS key was created with alias **Ty_MasterKey** and key ID **9598c159-b460-405c-9613-7175a57070e6.**

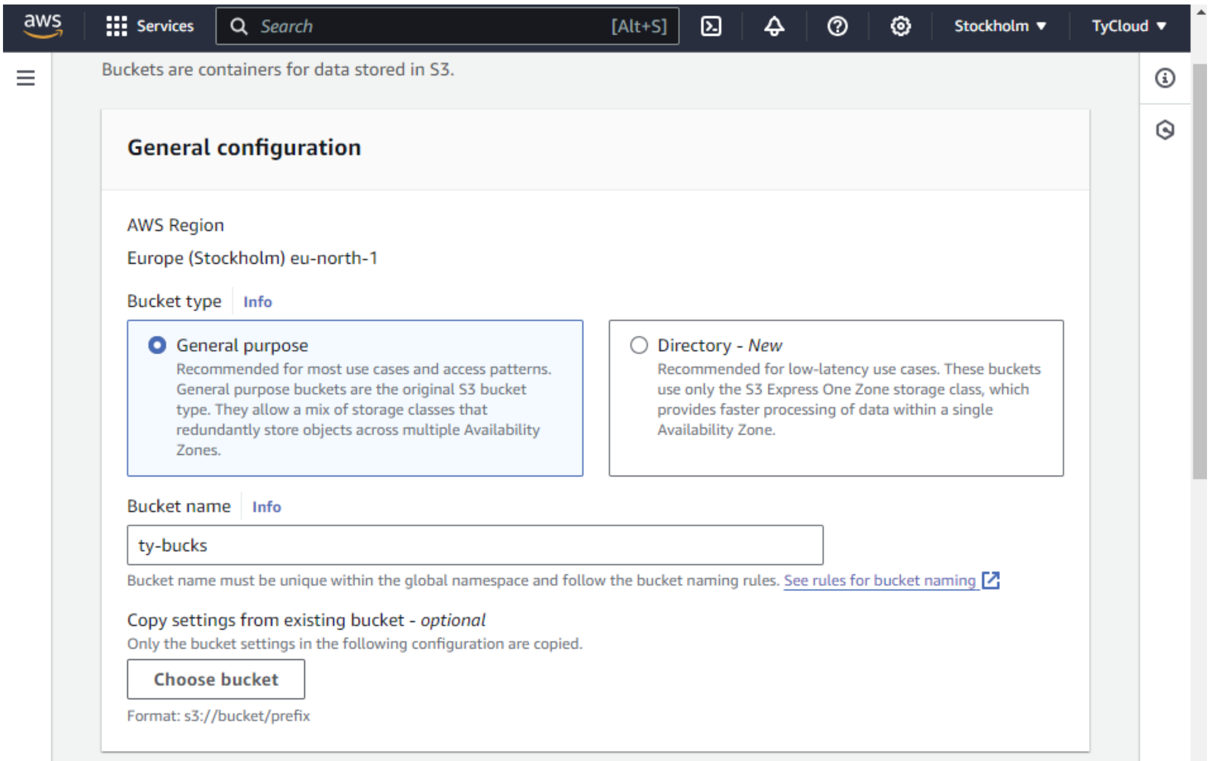
Customer managed keys (1)

□	Aliases	Key ID	Status	Key type	Key spec	Key usage
□	Ty_Master...	9598c159...	Enabled	Symmetric	SYMMETR...	Encrypt and decr

Part 2: Integrating the Created KMS to AWS services Amazon S3 and Amazon EBS.

a) Amazon S3

First, I created a new S3 bucket named **ty-bucks** and selected the region **Europe (Stockholm) eu-north-1**



Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory - *New*
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info

ty-bucks

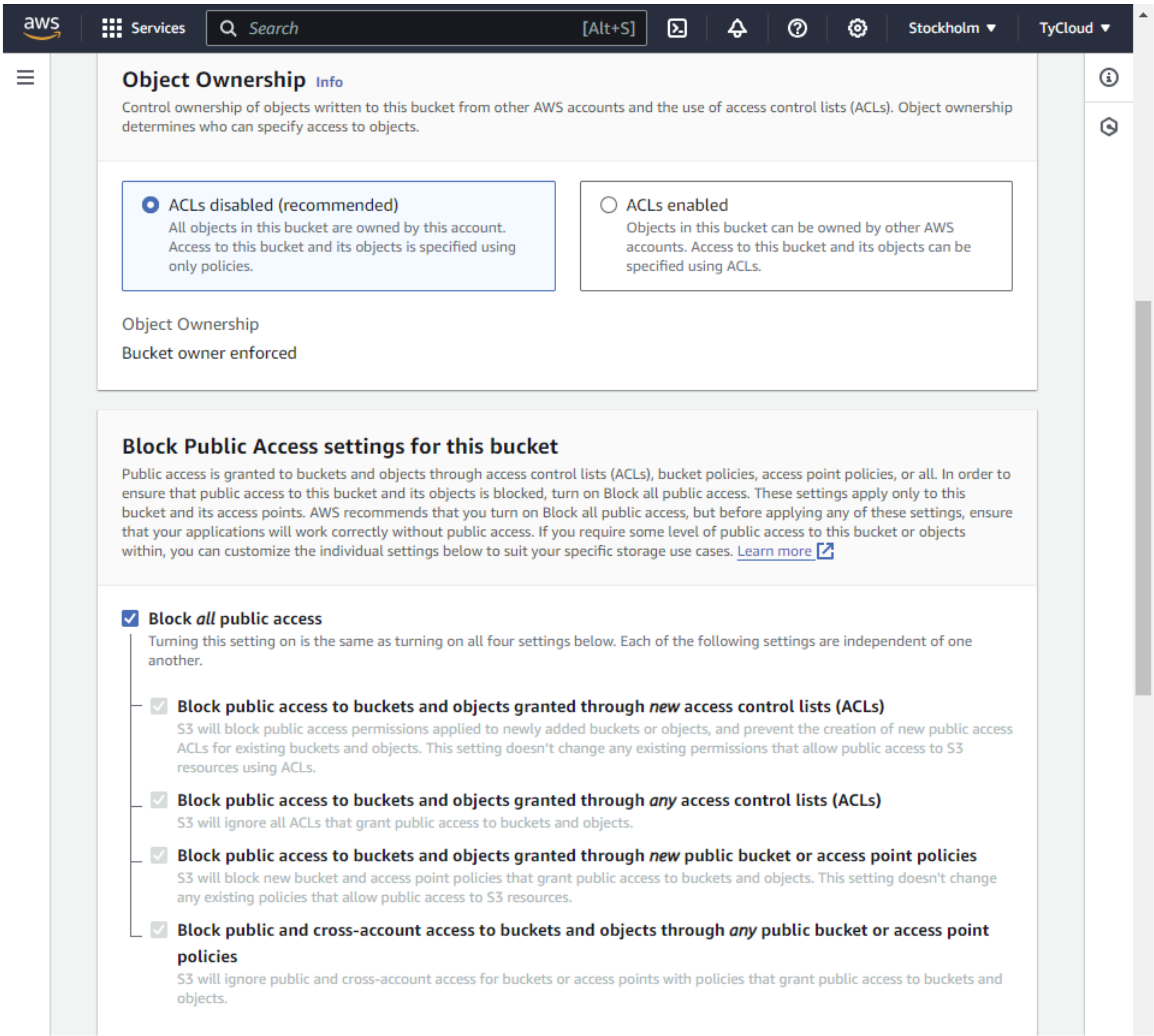
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

I retained the default settings for Object Ownership. I also blocked all public access settings for this new bucket and enabled Bucket versioning.



Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

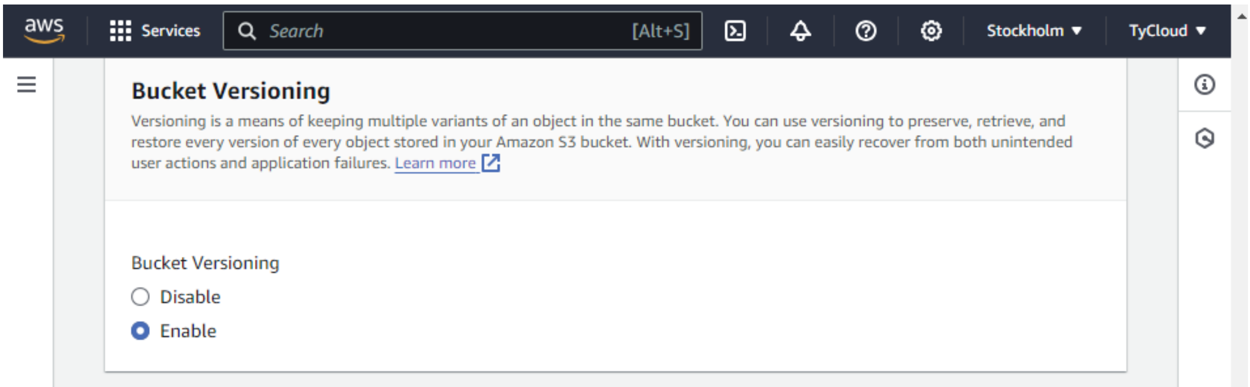
Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

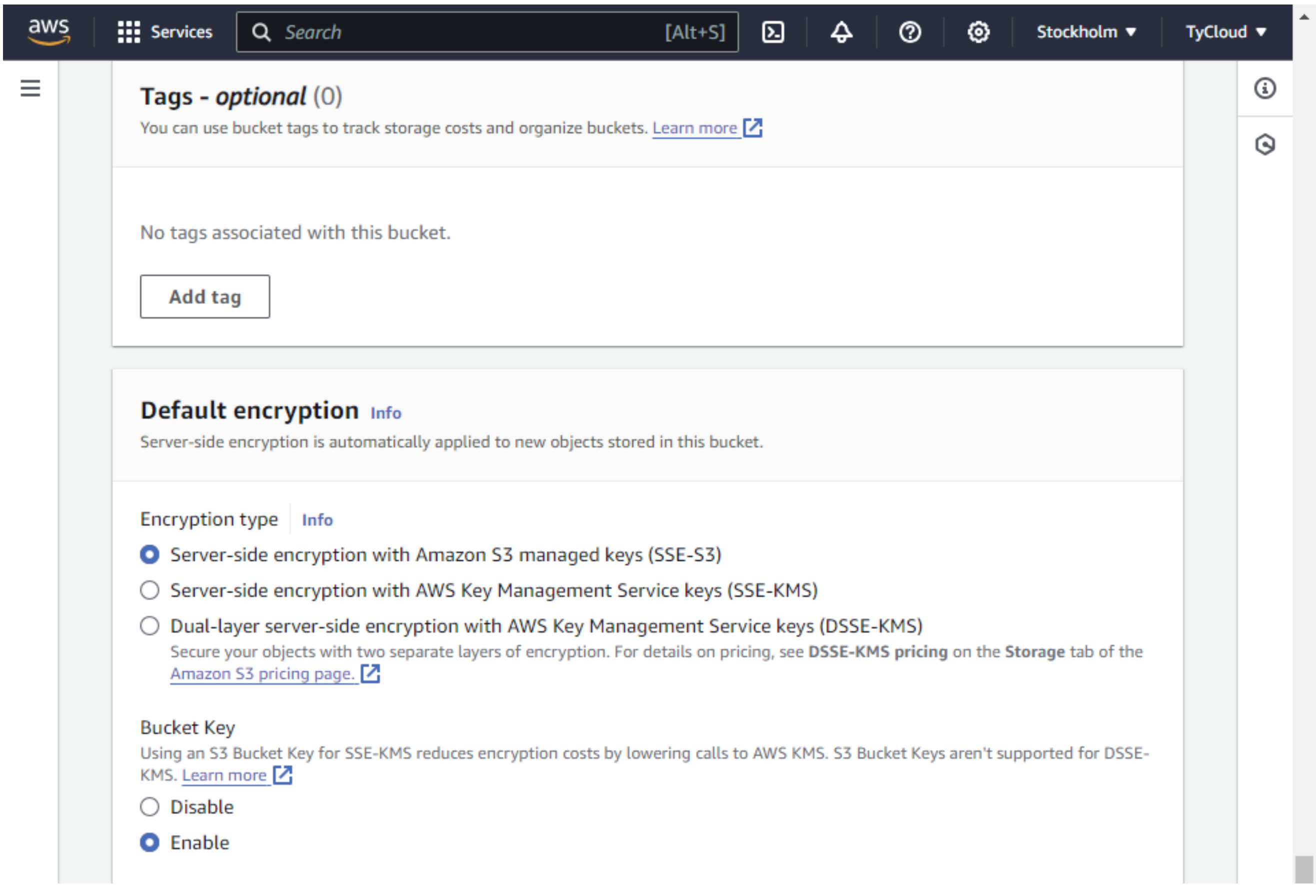


Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
○ Disable
● Enable

On Tags and encryption settings, I retained default settings.



Tags - *optional* (0)

You can use bucket tags to track storage costs and organize buckets. Learn more

No tags associated with this bucket.

Add tag

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type | Info

● Server-side encryption with Amazon S3 managed keys (SSE-S3)

○ Server-side encryption with AWS Key Management Service keys (SSE-KMS)

○ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

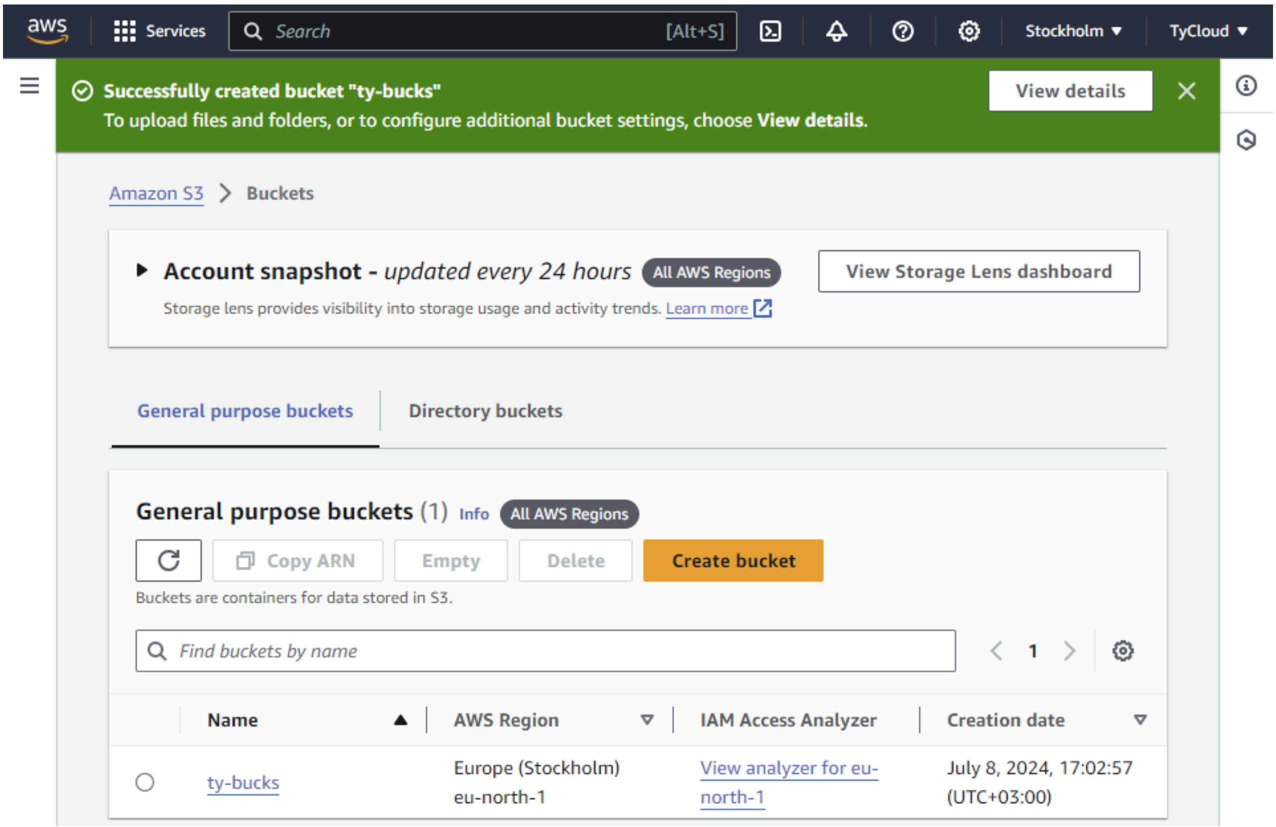
Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

○ Disable

● Enable

The S3 bucket was successfully created.



Successfully created bucket "ty-bucks"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Amazon S3 > Buckets

Account snapshot - *updated every 24 hours* All AWS Regions

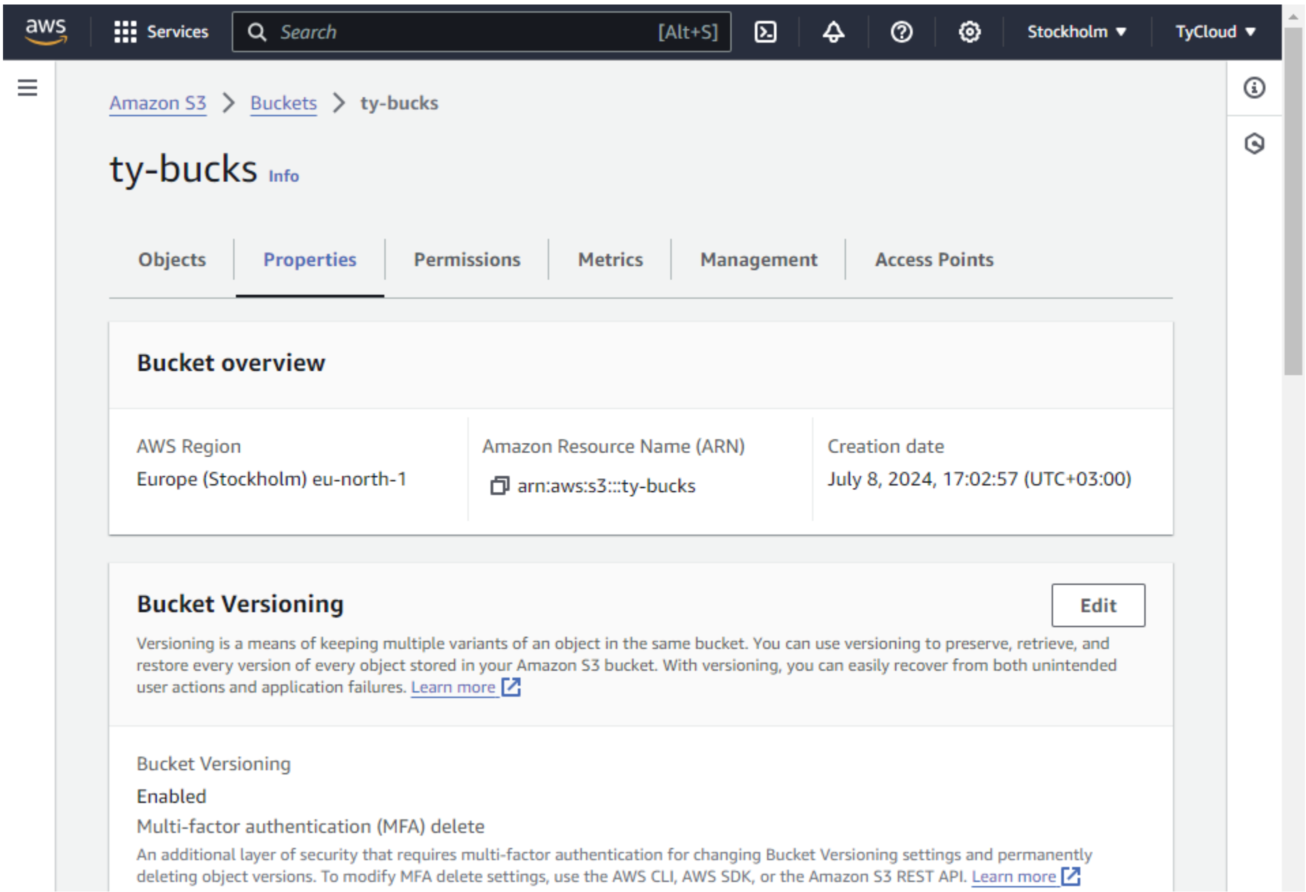
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name ▲	AWS Region	IAM Access Analyzer	Creation date
○	ty-bucks	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	July 8, 2024, 17:02:57 (UTC+03:00)

I navigated to the bucket properties and ensured the bucket versioning was enabled.



ty-bucks

Bucket overview

AWS Region
Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN)
arn:aws:s3:::ty-bucks

Creation date
July 8, 2024, 17:02:57 (UTC+03:00)

Bucket Versioning

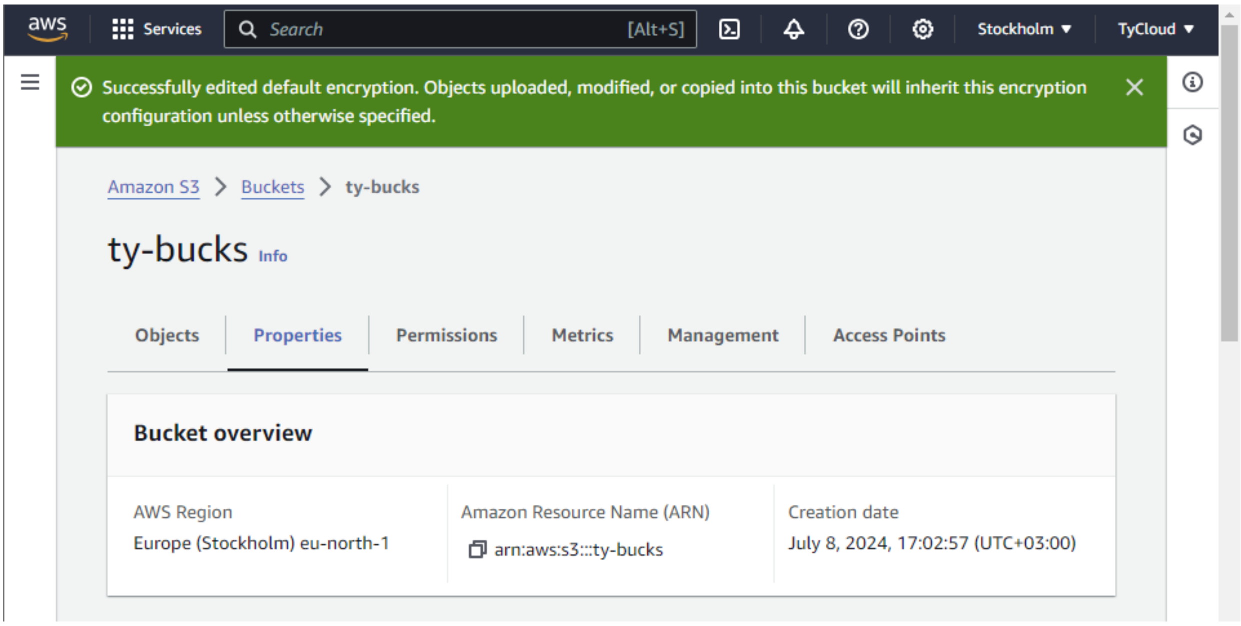
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Enabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more

I then enabled the Default Encryption using the Edit tab.

For **server-side encryption**, I chose the **AWS Key Management Service key (SSE-KMS)** to create a secured encryption by AWS KMS.



Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.

ty-bucks

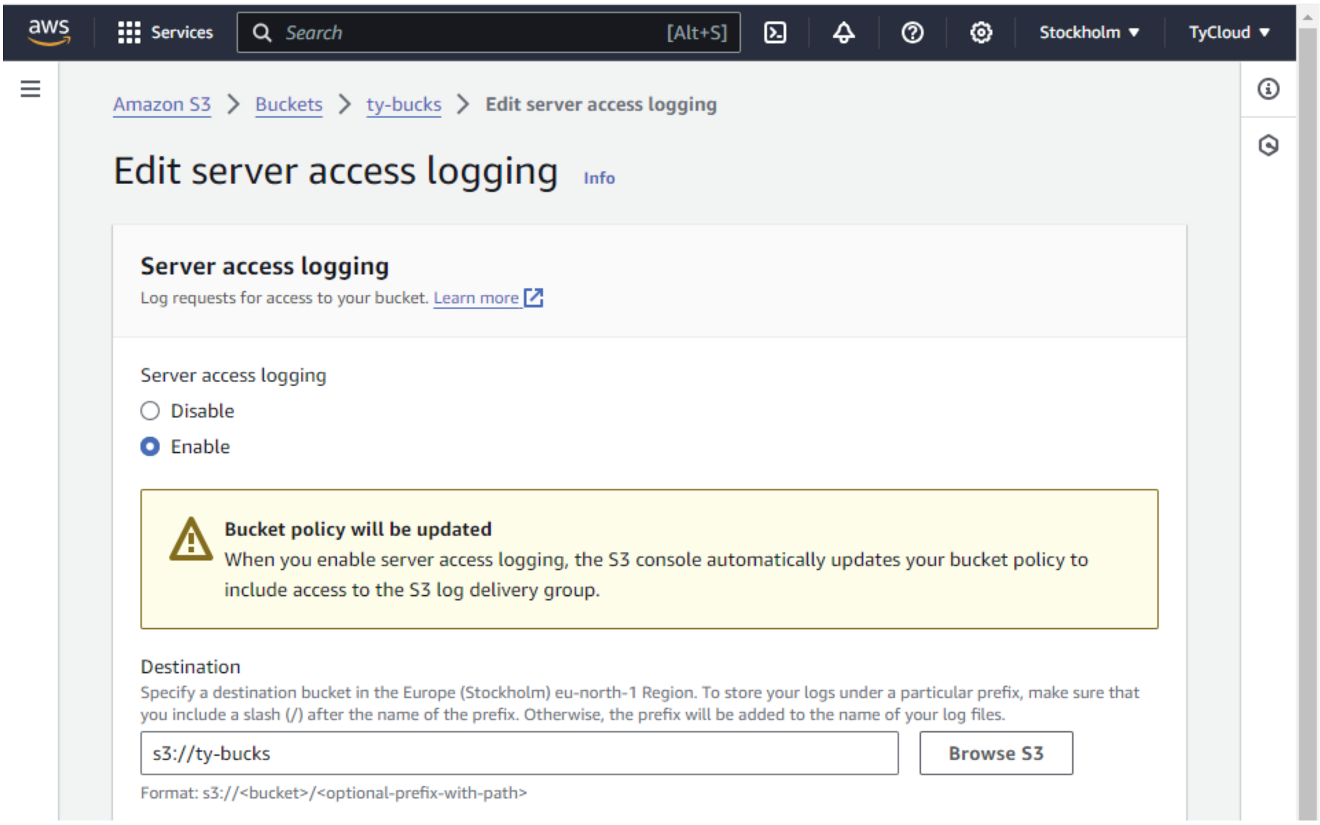
Bucket overview

AWS Region
Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN)
arn:aws:s3:::ty-bucks

Creation date
July 8, 2024, 17:02:57 (UTC+03:00)

I then edited **the Server Access Logging** and **Enabled** it. I also added my S3 bucket **ty-bucks** as the **Target** bucket.



Edit server access logging Info

Server access logging

Log requests for access to your bucket. Learn more

Server access logging

○ Disable

● Enable

Bucket policy will be updated

When you enable server access logging, the S3 console automatically updates your bucket policy to include access to the S3 log delivery group.

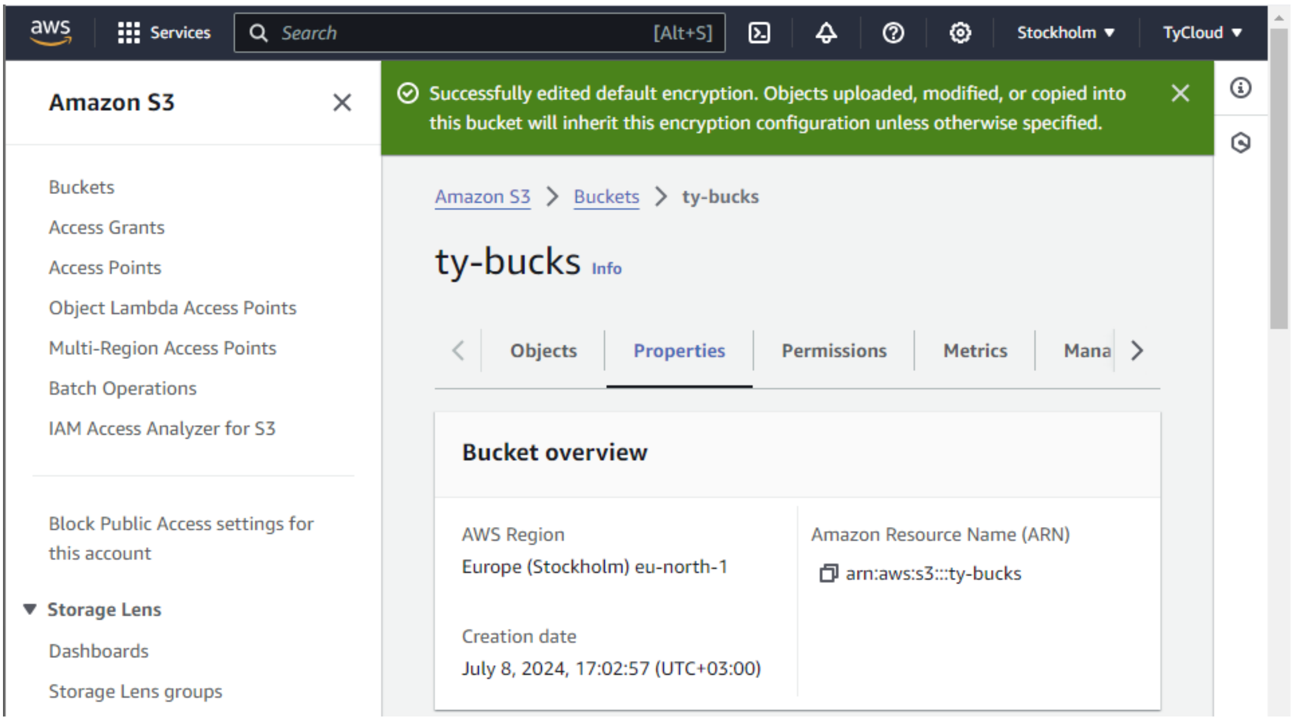
Destination

Specify a destination bucket in the Europe (Stockholm) eu-north-1 Region. To store your logs under a particular prefix, make sure that you include a slash (/) after the name of the prefix. Otherwise, the prefix will be added to the name of your log files.

s3://ty-bucks

Format: s3://<bucket>/<optional-prefix-with-path>

On the S3 management console, I navigated to my bucket properties and edited the default encryption so that a user can encrypt and protect storage data with the help of customer-managed keys.



Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.

Bucket overview

AWS Region

Europe (Stockholm) eu-north-1

Amazon Resource Name (ARN)

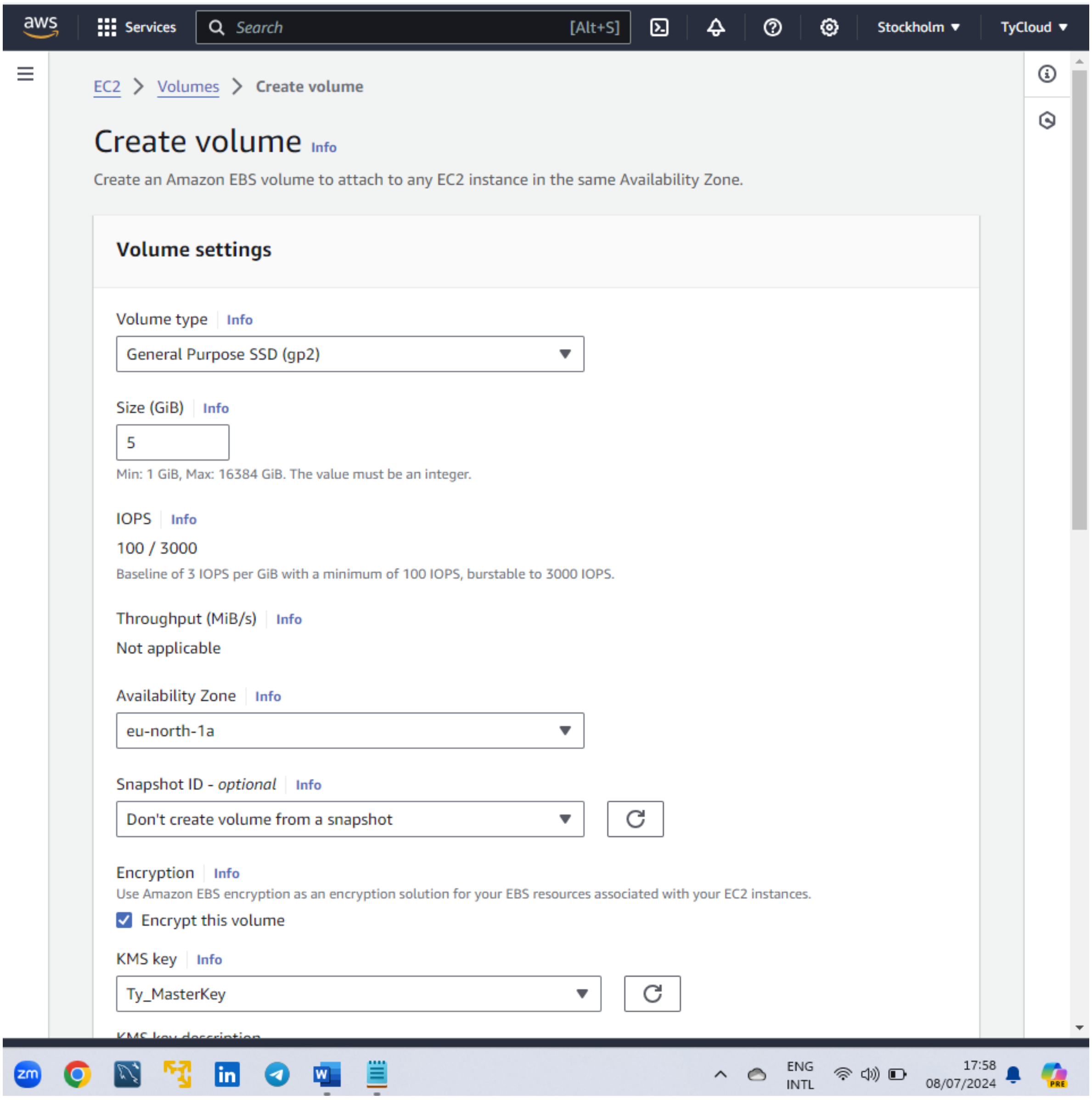
arn:aws:s3:::ty-bucks

Creation date

July 8, 2024, 17:02:57 (UTC+03:00)

b) KMS to Amazon EBS Volume

To encrypt EBS, I first navigated to the **EC2 console**, opened the **volumes** pane under the **Elastic Block Store** and clicked on **Create Volume**. I configured the Volume type as **General Purpose SSD (gp2)** and specified the size of the volume as **5** in the **Size (GiB)** field as shown below. In the KMS Key field, I selected the **customer-managed** key that I created earlier: **Ty_MasterKey.**I also checked the box to encrypt the volume I created**.**



Create volume Info

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Volume type Info
General Purpose SSD (gp2)

Size (GiB) Info
5
Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS Info
100 / 3000
Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS.

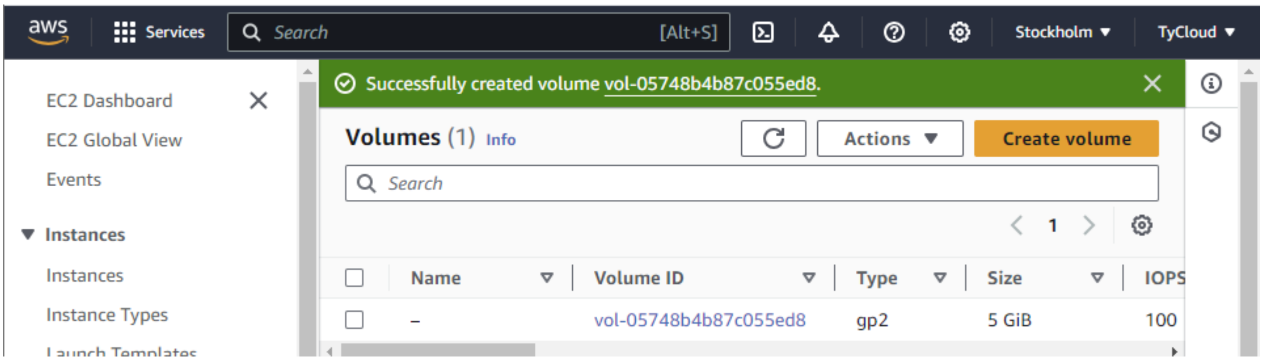
Throughput (MiB/s) Info
Not applicable

Availability Zone Info
eu-north-1a

Snapshot ID - *optional* Info
Don't create volume from a snapshot

Encryption Info
Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.
☑ Encrypt this volume

KMS key Info
Ty_MasterKey



Successfully created volume vol-05748b4b87c055ed8.

Volumes (1) Info

	Name	Volume ID	Type	Size	IOPS
□	–	vol-05748b4b87c055ed8	gp2	5 GiB	100

Conclusion

In completing this lab, I have gained a comprehensive understanding of AWS Key Management Service and its critical role in securing sensitive data. The hands-on experience of creating and managing cryptographic keys has provided me with valuable insights into how AWS KMS integrates seamlessly with other AWS services to offer a cohesive security solution. This knowledge equips me to implement and manage secure data encryption strategies effectively, reinforcing our organization's commitment to data protection and compliance with industry standards.