

अ

Quick Guide

MongoSafenet

Automatic MongoDB Backup Solution

Table of Contents

1. Introduction

2. TECHNIQUE 1: For Windows11/10 Using Task Scheduler

3. MongoDB Installation

- Download and Install MongoDB
- Add MongoDB to PATH

4. MongoDB Compass Installation

- Download and Install MongoDB Compass

5. Additional Tools Installation

- Tool Selection
- Installation Steps

6. Open MongoDB Compass

- Connecting to MongoDB
- Connection Troubleshooting

7. About the Backup Script

- Understanding the Backup Script

8. Configuring Task Scheduler

- Create MongoDB Backup Schedule
 - Create a Dedicated Folder
 - Create a Backup Script File
 - Edit the Script
 - Save the Script
- Schedule Backups with Task Scheduler
 - Open Windows Task Scheduler
 - Create a Basic Task
 - Follow the Wizard

9. AWS Setup

- Create an AWS Account
 - Visit the AWS Website
 - Sign Up for an Account
 - Follow the Registration Process

- **Create an S3 Bucket**
 - **Log In to AWS Management Console**
 - **Access the S3 Service**
 - **Create a New Bucket**
- **Create an IAM User with S3 Full Access**
 - **Access the AWS IAM Console**
 - **Generate a New IAM User**
 - **Assign S3 Full Access**
- **Obtain Access Credentials**

10. AWS Command Line Interface (CLI)

- **Configure AWS CLI**
 - **Open a Command Prompt or Terminal**
 - **Run AWS Configure**
- **Verify AWS CLI Configuration**
 - **List AWS CLI Configuration**
 - **List S3 Buckets**

11. Run the Scheduled Task

12. TECHNIQUE: Amazon EC2 (Ubuntu) Using Cron

13. AWS Setup

- **Create an AWS Account**
 - **Visit the AWS Website**
 - **Sign Up for an Account**
 - **Follow the Registration Process**
- **Create an S3 Bucket**
 - **Log In to AWS Management Console**
 - **Access the S3 Service**
 - **Create a New Bucket**
- **Create an IAM User with S3 Full Access**
 - **Access the AWS IAM Console**
 - **Generate a New IAM User**
 - **Assign S3 Full Access**
- **Obtain Access Credentials**
- **Create an EC2 Instance**

14. Configuring EC2 Instance

15. Conclusion

Introduction to MongoSafenet: Automating MongoDB Backups to AWS S3

In today's data-driven landscape, efficient data management is paramount for businesses and organizations of all sizes. MongoDB, a NoSQL database management system, has emerged as a popular choice for its scalability and flexibility in handling vast volumes of data. As organizations increasingly migrate their operations to the cloud for improved accessibility and scalability, there arises a pressing need for robust database backup solutions that seamlessly integrate with cloud platforms.

The MongoSafenet project represents a significant advancement in the realm of MongoDB database management. This innovative solution is meticulously designed to automate MongoDB backup processes and streamline data storage in the cloud. Specifically, MongoSafenet leverages the power of Amazon Web Services (AWS) S3 Buckets to offer a reliable and efficient cloud-based storage solution for MongoDB backups.

The core objective of MongoSafenet is to save valuable time and resources by simplifying the complex task of MongoDB backup, while ensuring the security and accessibility of these backups through AWS infrastructure. This project seamlessly marries several cutting-edge technologies to provide a comprehensive and hassle-free MongoDB backup and cloud integration solution.

Key Technologies Utilized in MongoSafenet:



- **MongoDB:** At the heart of this project lies MongoDB, a robust NoSQL database management system known for its scalability, flexibility, and document-oriented data storage capabilities. MongoDB serves as the primary database engine for this solution.



- **MongoDB Compass:** MongoDB Compass is a sophisticated GUI (Graphical User Interface) tool designed to facilitate the management and visualization of MongoDB databases. It plays a pivotal role in configuring, monitoring, and interacting with the MongoDB database.



- **Amazon Web Services (AWS):** AWS, Amazon's comprehensive cloud computing platform, offers a vast array of services and resources. In the context of MongoSafenet, AWS provides the infrastructure needed to host and manage MongoDB backups seamlessly.



- **Windows Task Scheduler:** On the Windows platform, the Task Scheduler is a powerful utility that allows users to automate various tasks, including running scripts and executing programs at specified intervals. In MongoSafenet, it is harnessed to automate the MongoDB backup process, ensuring regular and reliable data backups.



-

AWS Command Line Interface (CLI): AWS CLI is a command-line tool provided by Amazon Web Services, enabling users to interact with AWS services from the command line. Within the MongoSafenet project, the AWS CLI is employed to interact with AWS S3 for secure storage of MongoDB backups.



-

AWS S3 Bucket: Amazon S3 (Simple Storage Service) is a highly scalable and durable object storage service provided by AWS. It serves as the cloud-based repository for MongoDB backups, ensuring data availability and durability.

AWS S3



-

IAM User for Bucket Access: AWS Identity and Access Management (IAM) is used to create a dedicated IAM user with specific permissions, including "AmazonS3FullAccess," to securely access and manage the S3 bucket. This IAM user ensures that backups are stored, accessed, and maintained with the highest level of security.

AWS IAM



-

Amazon EC2 (Elastic Compute Cloud) is a cloud service that allows you to rent virtual servers, known as instances, to run your applications. You can choose from a variety of instance types and configure them to suit your needs. EC2 offers scalability, high availability, and a wide range of use cases, including web hosting, data analysis, and machine learning.

Amazon EC2

The following sections of this guide will delve into the step-by-step instructions required to implement MongoSafenet effectively, from MongoDB installation to configuring the Task Scheduler and leveraging AWS resources for cloud-based MongoDB backups. By embracing this automation solution, organizations can not only safeguard their valuable MongoDB data but also enhance operational efficiency through seamless cloud integration.

Technique 1

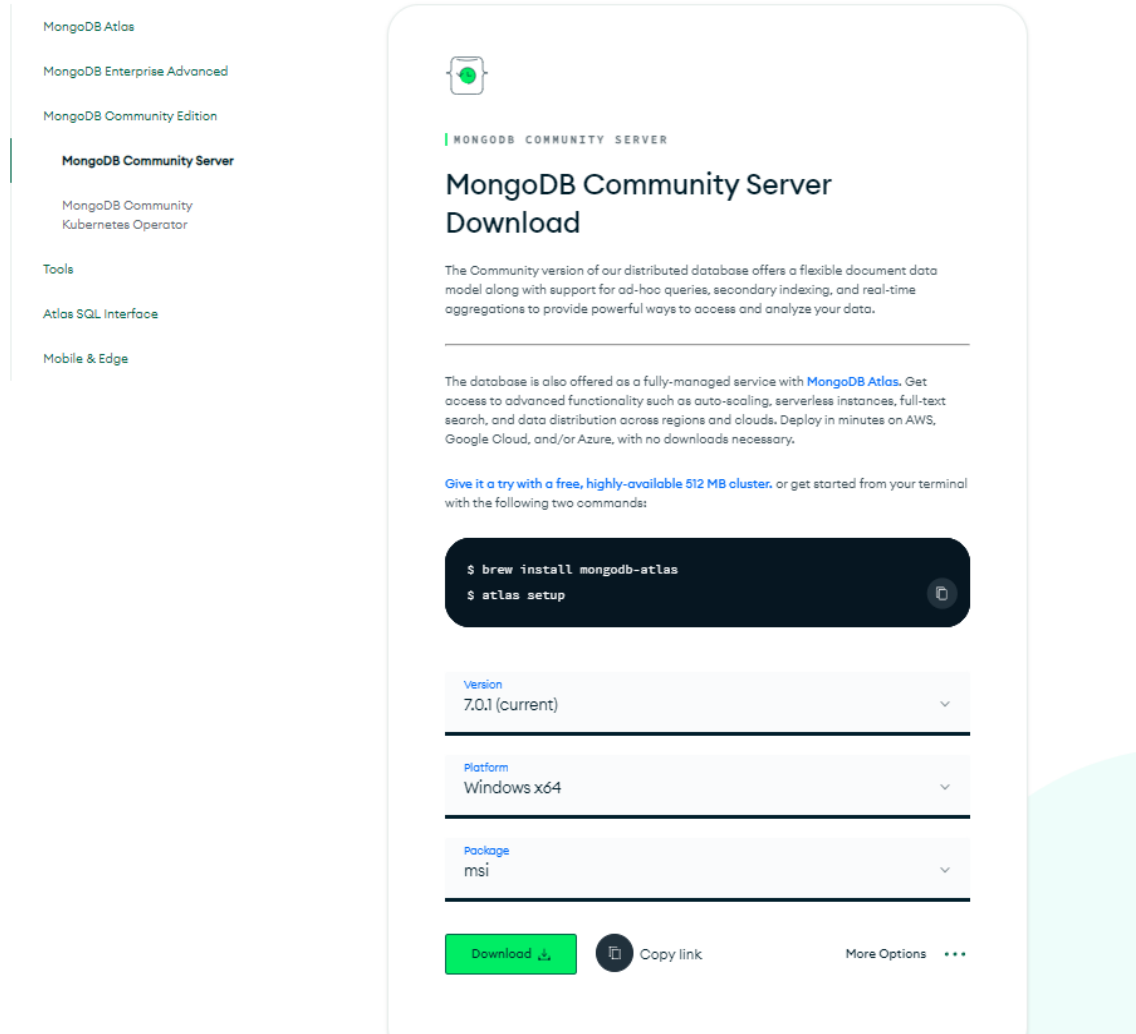
WINDOWS 11/10 USING TASK SCHEDULER

MongoDB Installation

Download MongoDB:

Step 1: Visit the MongoDB website at <https://www.mongodb.com/>.

Here, you'll find MongoDB versions tailored to different operating systems like Windows, macOS, and Linux.



The screenshot displays the MongoDB website's download page for the Community Server. On the left, a sidebar lists various MongoDB products, with 'MongoDB Community Server' highlighted. The main content area features the MongoDB logo and the heading 'MongoDB Community Server Download'. Below this, there is a brief description of the Community version and a link to the fully-managed MongoDB Atlas service. A terminal snippet shows the commands to install MongoDB Atlas using Homebrew. The selection interface includes dropdown menus for 'Version' (7.0.1 (current)), 'Platform' (Windows x64), and 'Package' (msi). At the bottom, there is a green 'Download' button, a 'Copy link' button, and a 'More Options' menu.

Step 2: Select the appropriate version for your operating system by clicking on the respective download link.

Install MongoDB:

Step 1: Once the download is complete, locate the downloaded installer file (usually ending in .msi for Windows).

Step 2: Double-click the installer file to run it. The installation wizard will guide you through the process.

Step 3: You can typically choose the installation directory and configure options as needed during the installation.

Step 4: Follow the on-screen instructions to complete the installation.

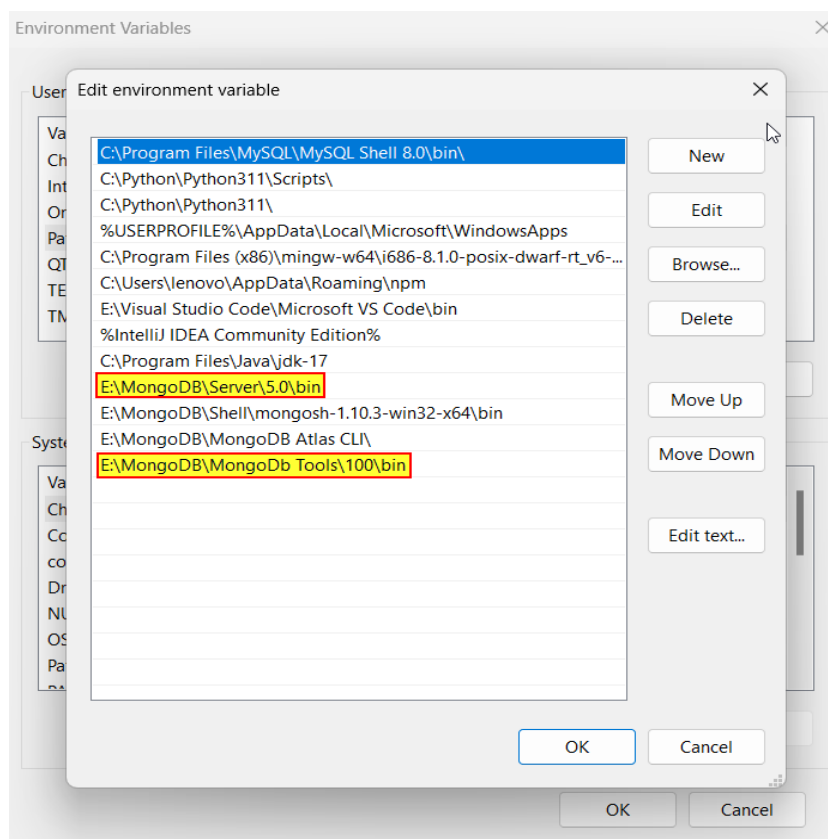
Add MongoDB to PATH:

Note: - To use MongoDB from the command line, you can add it to your system's PATH. PATH is a system environment variable that contains a list of directories. When you run a command in the command prompt or terminal, your system checks these directories for executable files.

Step 1: Find the directory where MongoDB was installed on your computer. This is typically under C:\Program Files\MongoDB on Windows or /usr/local/bin on macOS and Linux.

Step 2: Append this directory path to your system's PATH variable.

Step 3: On Windows, you can do this by opening the Start menu, searching for "Environment Variables," and clicking "Edit the system environment variables." Then, click the "Environment Variables" button, select "Path" in the "System variables" section, and click "Edit." Add the MongoDB directory to the list of paths.

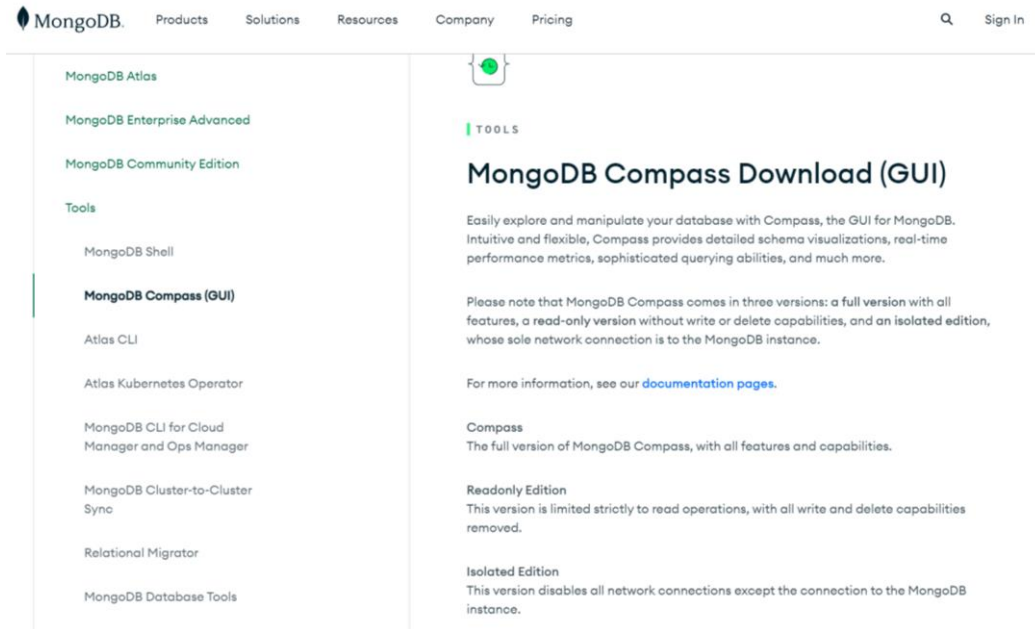


MongoDB Compass Installation

This guide provides concise steps for installing MongoDB Compass.

Download MongoDB Compass:

Step 1: Visit the MongoDB website.



Step 2: Download the MongoDB Compass version that matches your operating system (Windows, macOS, or Linux).

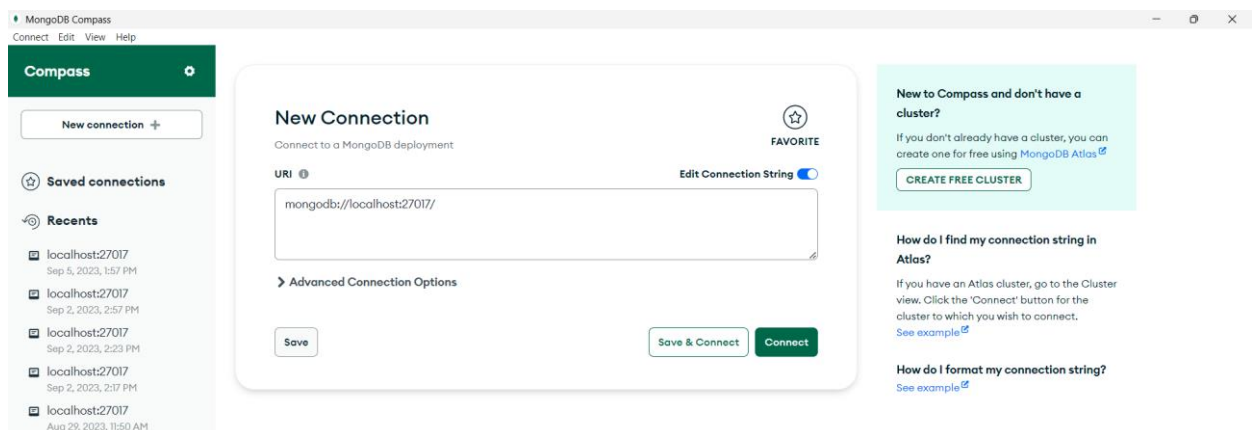
Install MongoDB Compass:

Step 1: Locate the downloaded installer file (usually with a .msi extension for Windows).

Step 2: Run the installer by double-clicking it.

Step 3: Follow the installation wizard's prompts, customizing settings as needed.

Step 4: Complete the installation by following the on-screen instructions.

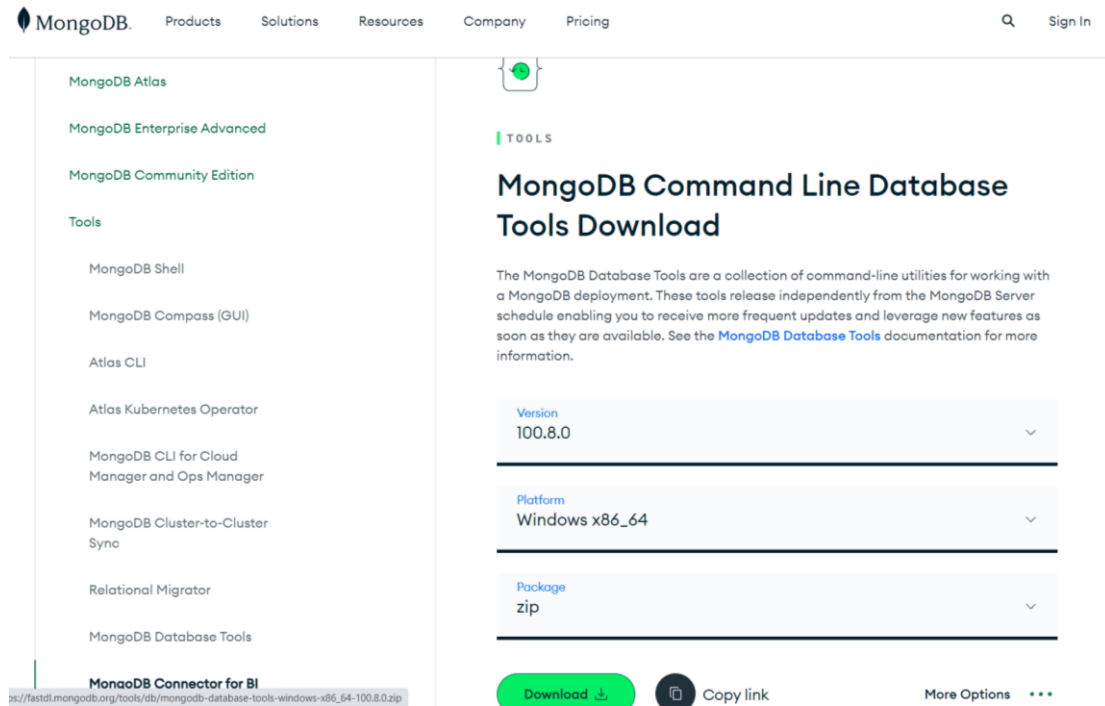


MongoDB Command Line Tools Installation

This guide provides concise steps for installing MongoDB Command Line Database Tools, along with adding to your system's PATH on Windows.

Step 1: Visit the official MongoDB website.

Step 2: Find the MongoDB Command Line Database Tools section.



Step 3: Download the tools for your operating system.

Step 4: Run the downloaded installer.

Step 5: Follow the installation instructions provided.

Step 6: Note that some tools may require additional configuration after installation.

Adding MongoDB Command Line Database Tools to PATH (Windows)

Step 1: Locate the Tools. Find the directory where MongoDB Command Line Database Tools are installed, which contains executable files like mongodump and mongorestore.

Step 2: Open the Start menu and search for "Environment Variables."

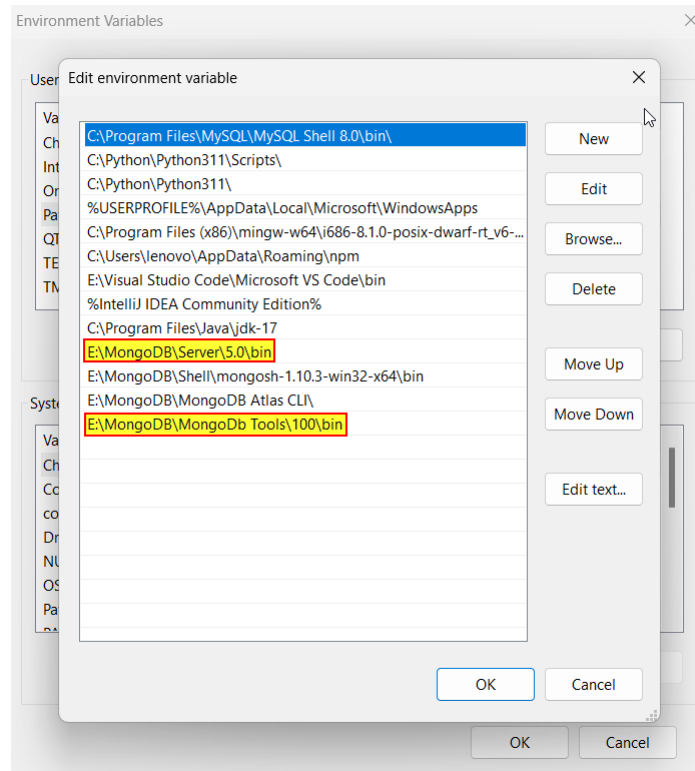
Step 3: Click "Edit the system environment variables."

Step 4: In the System Properties window, click the "Environment Variables" button.

Step 5: Under "System Variables," find and select the "Path" variable, then click "Edit."

Step 6: Click "New" and add the path to the MongoDB Command Line Database Tools directory (e.g., C:\Program Files\MongoDB\Tools\bin).

Step 7: Click "OK" to save the changes.



By following these steps, you'll have MongoDB Compass and MongoDB Command Line Database Tools installed and added to your Windows system's PATH, making them accessible from any command prompt or terminal window.

MongoDB Compass Configuration

Step 1: Connecting to MongoDB:

After successfully installing MongoDB Compass, it's time to connect it to your MongoDB server.

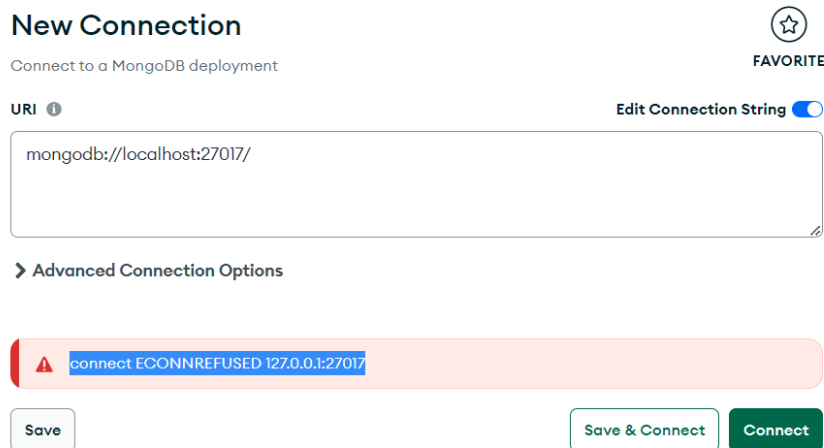
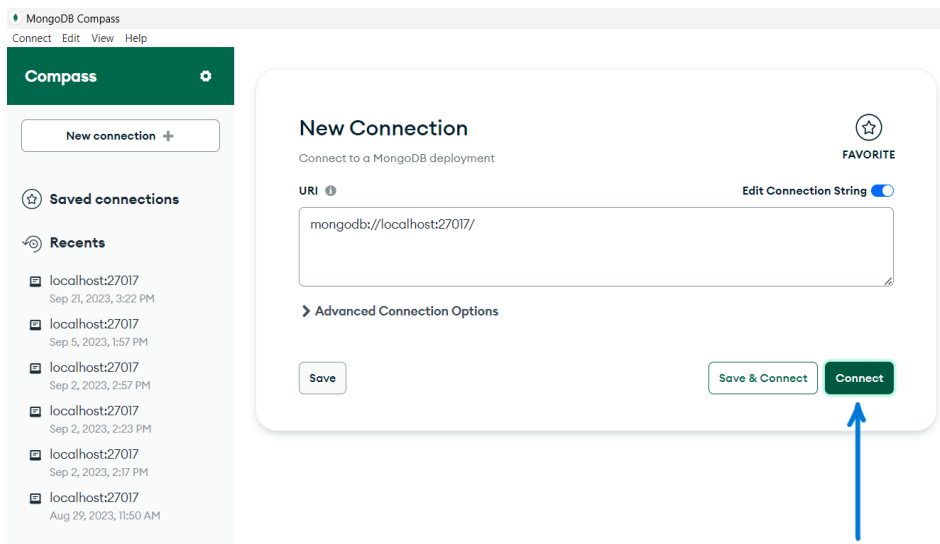
Launch MongoDB Compass from your computer, either by clicking its icon in the applications menu or searching for it in the start menu.

Step 2: Connection:

MongoDB Compass will open with a connection dialog.

You'll need to enter the following information to connect:

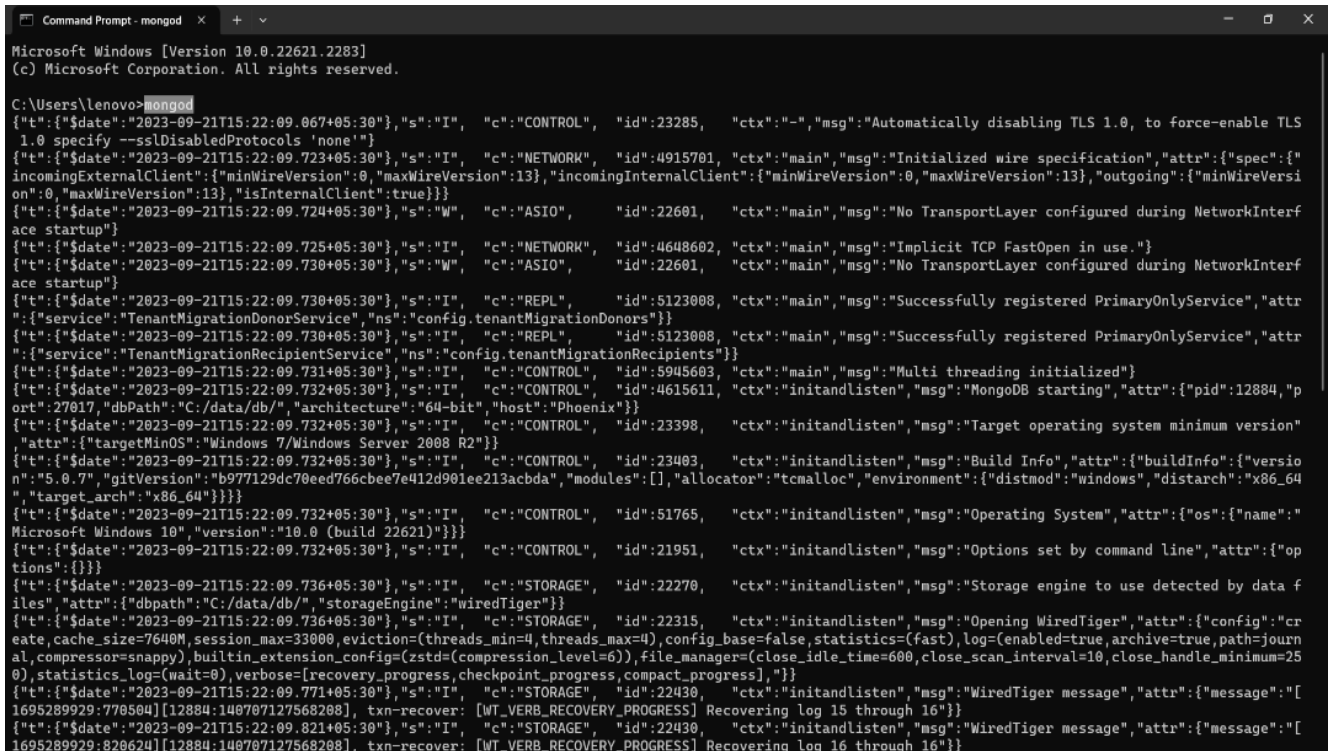
- **Host:** The address of your MongoDB server.
- **Port:** The port number where MongoDB is running (the default is usually 27017).
- **Authentication Credentials:** This includes your username and password for accessing MongoDB.



Step 3: Connection Troubleshooting:

If you encounter any issues connecting to MongoDB, consider the following steps:

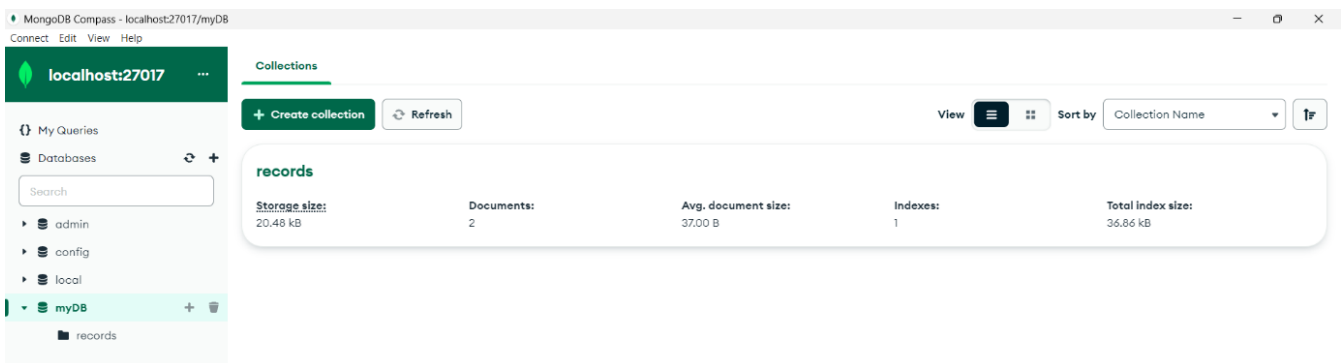
- Ensure that your MongoDB server is running. You can check by running **mongod** in the command prompt or terminal.
- Verify that the connection details, including the host and port, are correct.



```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>mongod
{"t":{"$date":"2023-09-21T15:22:09.067+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically disabling TLS 1.0, to force-enable TLS
1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2023-09-21T15:22:09.723+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"main", "msg":"Initialized wire specification", "attr":{"spec":{"
incomingExternalClient":{"minWireVersion":0, "maxWireVersion":13}, "incomingInternalClient":{"minWireVersion":0, "maxWireVersion":13}, "outgoing":{"minWireVersi
on":0, "maxWireVersion":13}, "isInternalClient":true}}}
{"t":{"$date":"2023-09-21T15:22:09.724+05:30"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main", "msg":"No TransportLayer configured during NetworkInterf
ace startup"}
{"t":{"$date":"2023-09-21T15:22:09.725+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"main", "msg":"Implicit TCP FastOpen in use.}
{"t":{"$date":"2023-09-21T15:22:09.730+05:30"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main", "msg":"No TransportLayer configured during NetworkInterf
ace startup"}
{"t":{"$date":"2023-09-21T15:22:09.730+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr
":{"service":"TenantMigrationDonorService", "ns":"config.tenantMigrationDonors"}}
{"t":{"$date":"2023-09-21T15:22:09.730+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr
":{"service":"TenantMigrationRecipientService", "ns":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2023-09-21T15:22:09.731+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main", "msg":"Multi threading initialized"}
{"t":{"$date":"2023-09-21T15:22:09.732+05:30"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten", "msg":"MongoDB starting", "attr":{"pid":12884, "p
ort":27017, "dbPath":"C:/data/db/", "architecture":"64-bit", "host":"Phoenix"}}
{"t":{"$date":"2023-09-21T15:22:09.732+05:30"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten", "msg":"Target operating system minimum version
", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2023-09-21T15:22:09.732+05:30"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten", "msg":"Build Info", "attr":{"buildInfo":{"versio
n":"5.0.7", "gitVersion":"b977129dc70eed766cbee7e412d981ee213acbdba", "modules":[], "allocator":"tcmalloc", "environment":{"distmod":"windows", "distarch":"x86_64
", "target_arch":"x86_64"}}}}
{"t":{"$date":"2023-09-21T15:22:09.732+05:30"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten", "msg":"Operating System", "attr":{"os":{"name":"
Microsoft Windows 10", "version":"10.0 (build 22621)}}}}
{"t":{"$date":"2023-09-21T15:22:09.732+05:30"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten", "msg":"Options set by command line", "attr":{"op
tions":{}}}
{"t":{"$date":"2023-09-21T15:22:09.736+05:30"},"s":"I", "c":"STORAGE", "id":22270, "ctx":"initandlisten", "msg":"Storage engine to use detected by data f
iles", "attr":{"dbpath":"C:/data/db/", "storageEngine":"wiredTiger"}}
{"t":{"$date":"2023-09-21T15:22:09.736+05:30"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten", "msg":"Opening WiredTiger", "attr":{"config":{"cr
eate_cache_size=7640M, session_max=33000, eviction=(threads_min=4, threads_max=4), config_base=false, statistics=(fast), log=(enabled=true, archive=true, path=journ
al, compressor=snappy), builtin_extension_config=(zstd=(compression_level=6)), file_managers=(close_idle_time=600, close_scan_interval=10, close_handle_minimum=25
0), statistics_log=(wait=0), verbose=[recovery_progress, checkpoint_progress, compact_progress], }}
{"t":{"$date":"2023-09-21T15:22:09.771+05:30"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten", "msg":"WiredTiger message", "attr":{"message":"[
1695289929:770504][12884:140707127568208], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 15 through 16"}}
{"t":{"$date":"2023-09-21T15:22:09.821+05:30"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"initandlisten", "msg":"WiredTiger message", "attr":{"message":"[
1695289929:820624][12884:140707127568208], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 16 through 16"}}
```

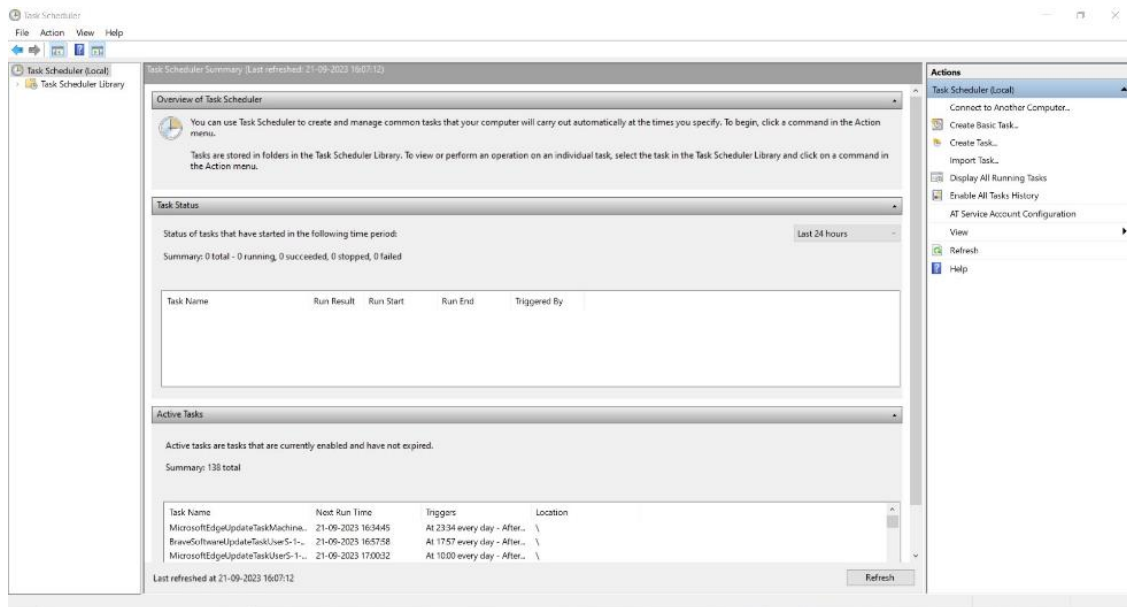
- Now try again to connect the database.



Task Scheduler Configuration

About the Backup Script:

The backup script is a set of commands that automate the process of creating backups of your MongoDB database and securely uploading them to an AWS S3 bucket. This script ensures the safety of your data.



Create MongoDB Backup Schedule:

Step 1: Create a Dedicated Folder: - Start by creating a dedicated folder on your computer where you'll store your backup scripts. For example, create a folder named "C:\MongoDBBackups."

Step 2: Create a Backup Script File: - Inside the folder, create a new text file and give it a name like "backup_mongodb.bat." (This script will be provided with the guide)

Step 3: Edit the Script: - Open the "backup_mongodb.bat" file using a text editor (like Notepad).

Step 4: Copy and paste the provided script into the file.

Step 5: Replace the placeholders in the script with your actual MongoDB server details, such as the host, port, and database name.

```
*backup_to_s3.bat - Notepad
File Edit Format View Help
@echo off

:: Set MongoDB connection details
set MONGO_HOST=localhost
set MONGO_PORT=27017
set MONGO_DB=<your mongoDB database name>

:: Set AWS S3 details
set AWS_S3_BUCKET=<your S3 bucket name>
set AWS_S3_PREFIX=backups/

:: Create a timestamp for the backup folder
set TIMESTAMP=%date:~4,2%->date:~7,2%->date:~10,4%_%time:~0,2%->time:~3,2%

:: Perform MongoDB backup using mongodump
mongodump --host %MONGO_HOST% --port %MONGO_PORT% --db %MONGO_DB% --out <path of your local backup folder>%TIMESTAMP%

:: Upload the backup to AWS S3 (assuming you have aws-cli configured)
aws s3 cp <path of your local backup folder>%TIMESTAMP% s3://%AWS_S3_BUCKET%/%AWS_S3_PREFIX%%TIMESTAMP% --recursive

:: Clean up the local backup folder
rmdir /s /q <path of your local backup folder>%TIMESTAMP%

echo Backup completed!
```

Step 6: Save the Script.

Schedule Backups with Task Scheduler:

Step 1: Access Windows Task Scheduler from your computer. You can find it in the Control Panel or by searching for "Task Scheduler" in the start menu.

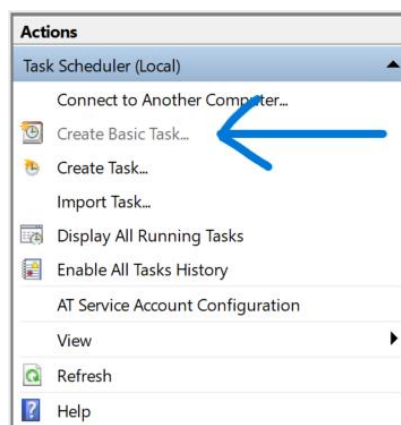
Step 2: Click "Create Basic Task" to start the task creation wizard in the Actions panel. The wizard will guide you through the following steps:

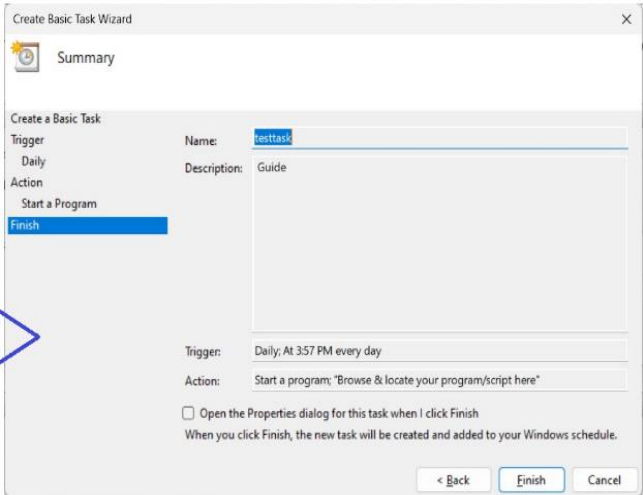
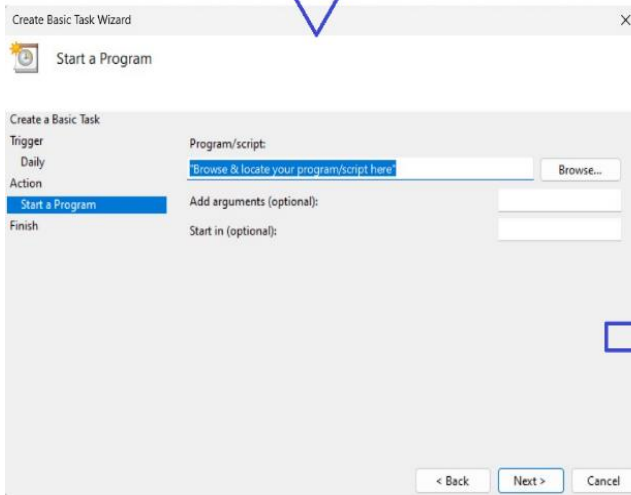
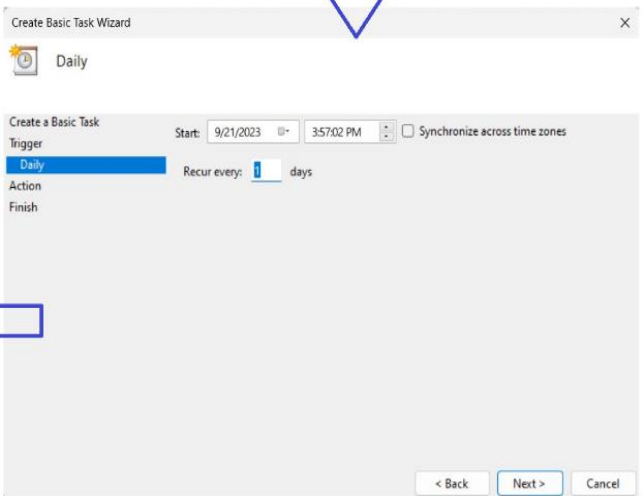
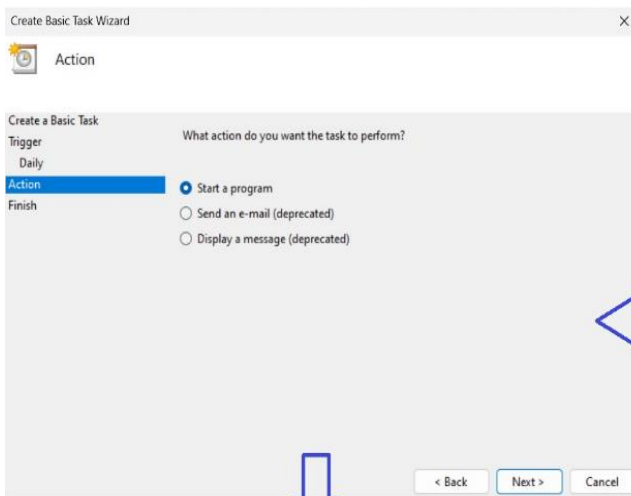
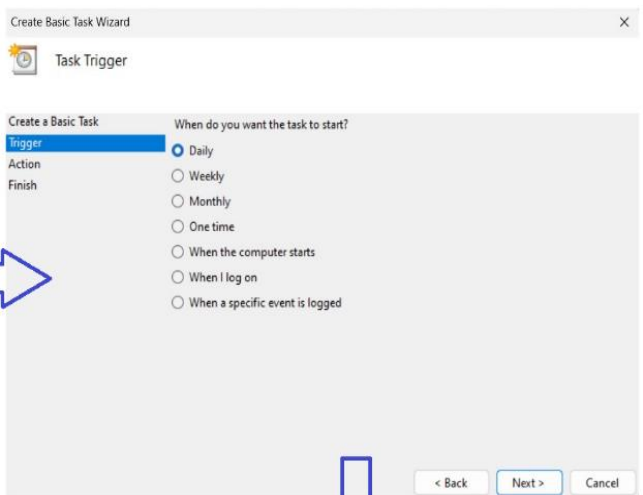
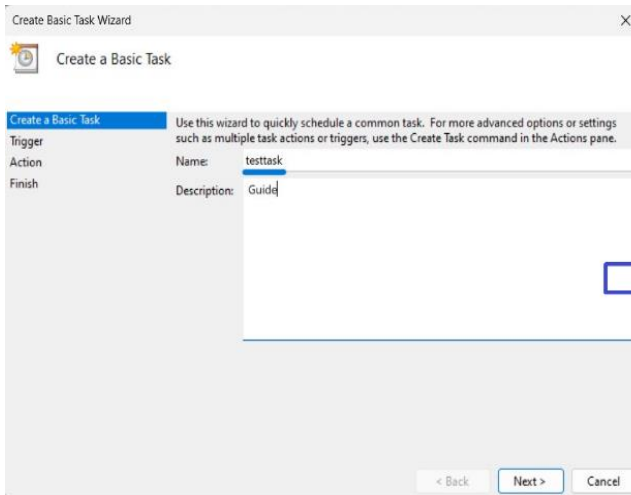
Step 3: Name the task and provide a description.

Step 4: Choose a schedule for your backups (e.g., daily, weekly).

Step 5: Select "Start a Program" as the action to perform.

Step 6: Locate and select the "backup_mongodb.bat" script that you created earlier.



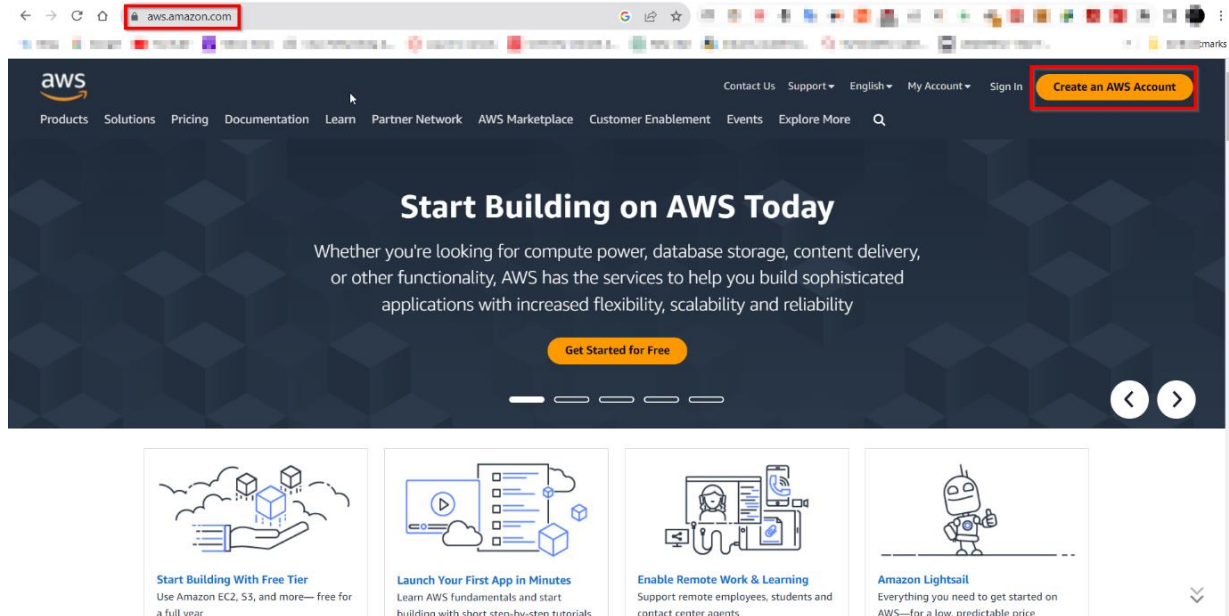


AWS Setup Guide

Note: - if you have AWS console already configured then go directly to step 5.

Step 1: Visit the AWS Website - Navigate to the official AWS website: [AWS Website] (<https://aws.amazon.com/>).

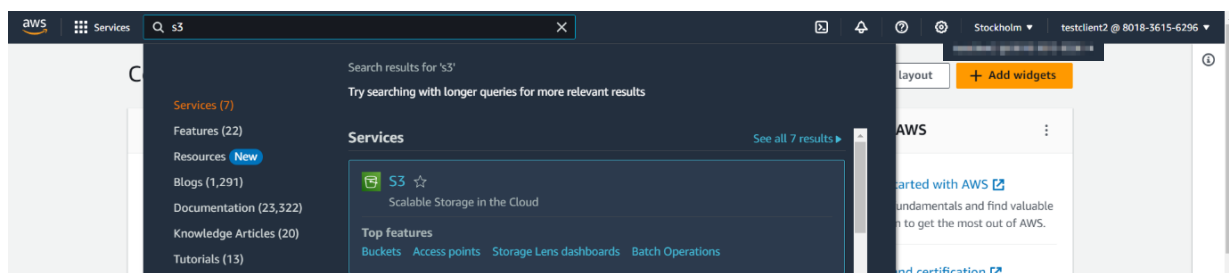
Step 2: Click the "Create an AWS Account" button.



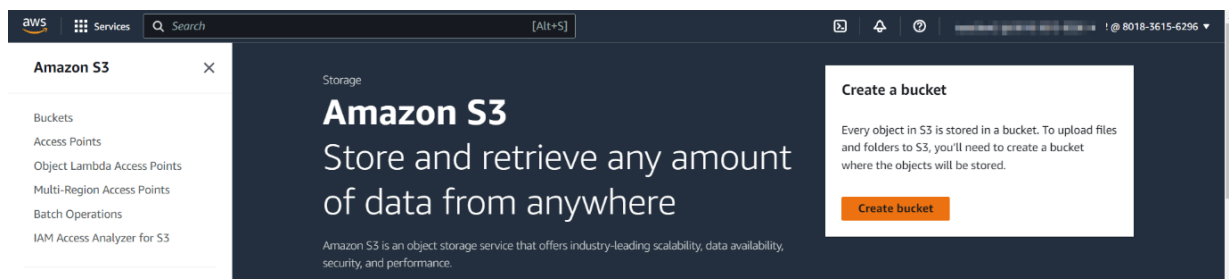
Step 3: Follow the guided registration process, which includes providing personal and payment information. Upon completion, you will have successfully created your AWS account.

Step 4: After successfully creating your AWS account, log in to the AWS Management Console.

Step 5: In the AWS Management Console, locate and select the "S3" service.

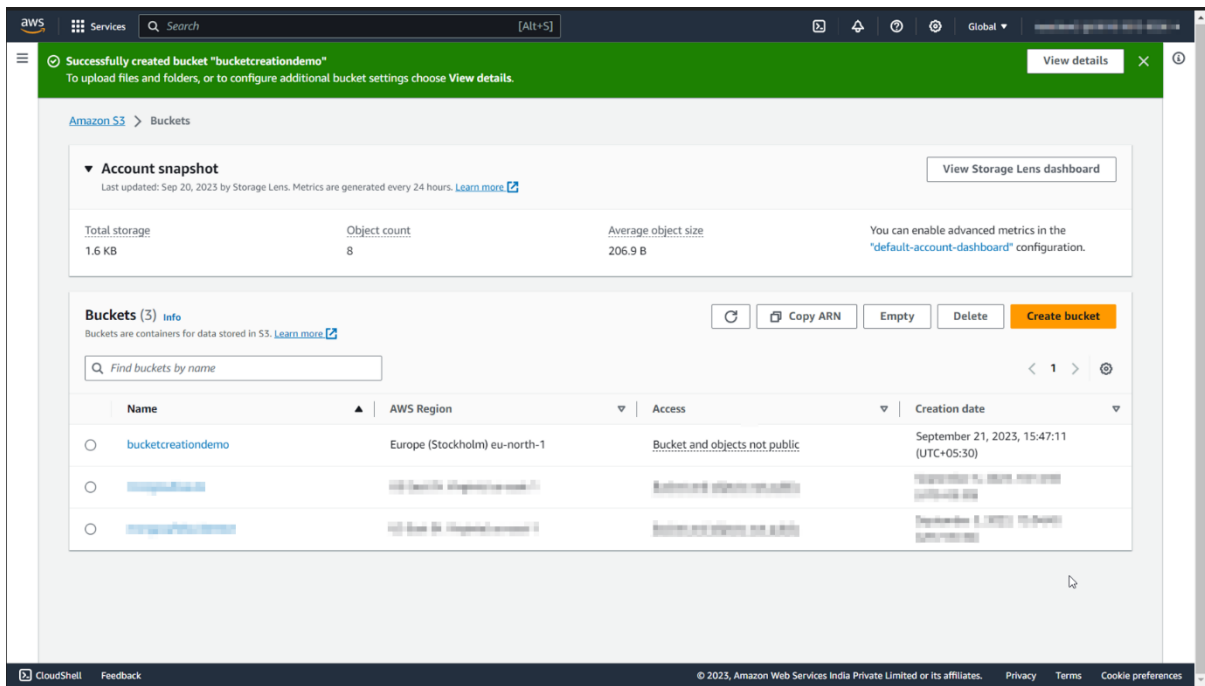


Step 6: Click the "Create Bucket" button.



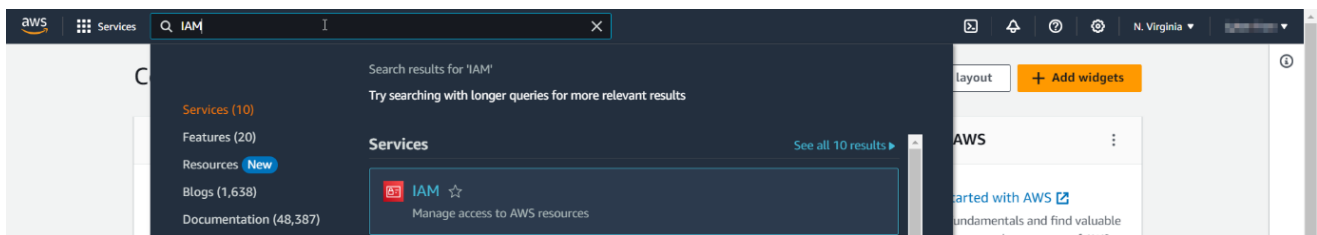
Step 7: Follow the steps in the bucket creation wizard, ensuring to provide a unique and meaningful bucket name.

Step 8: Configure any desired settings as needed during the setup process.



Create an IAM User with S3 Full Access:

Step 1: In the AWS Management Console, find and select the "Identity and Access Management (IAM)" service.



Step 2: Within the IAM console, create a new IAM user, specifying the user's details. Choose "programmatic access" when prompted.

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Step 3: When configuring permissions for the user, grant them "S3 Full Access" permissions, ensuring they have the necessary privileges.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1126)

Choose one or more policies to attach to your new user.



Create policy

Filter by Type

amazonS3



All types

5 matches

< 1 >



<input type="checkbox"/>	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/>	AmazonS3ObjectLambdaExecution...	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	0

Step 4: After successfully creating the IAM user, make note of the "Access Key ID" and "Secret Access Key" assigned to this user. These credentials must be kept secure, as they will be required for configuring the AWS Command Line Interface (CLI).

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > taskuser > Create access key

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

Retrieve access keys

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA3VMJIYWEM6HPV2PB	oRwMU9Ec3FcpcubgGpRnyP1qb9mDbHIUYXWk2Zz Hide

Console password

You have successfully enabled the user's new password.
This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

Console sign-in URL
 <https://801836156296.signin.aws.amazon.com/console>

User name
 taskuser

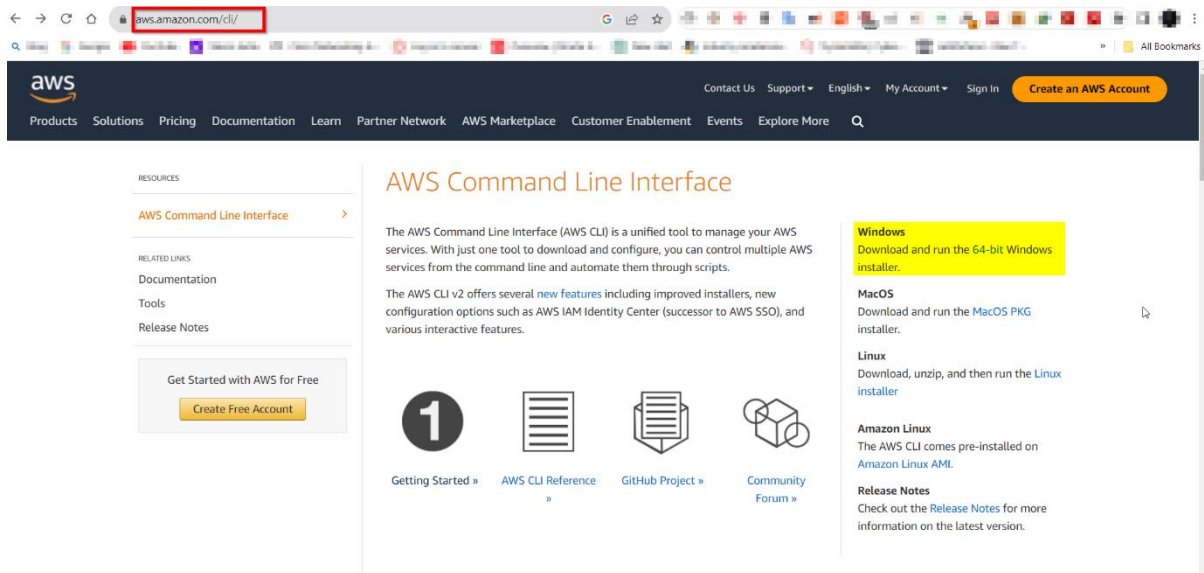
Console password
 W=c@GP2* [Hide](#)

[Download .csv file](#) [Close](#)

AWS Command Line Interface (CLI) Configuration

Installing AWS CLI on Windows:

Step 1: Visit the official AWS CLI download page for Windows by navigating to [AWS CLI for Windows] (<https://aws.amazon.com/cli/>).



Step 2: Choose the appropriate MSI installer based on your system's architecture (64-bit or 32-bit) and click the download link.

Step 3: Locate the downloaded MSI installer file, such as "awscli-x86_64.msi," and double-click it to initiate the installation process.

Step 4: The installer wizard will open. Click "Next" to proceed.

Step 5: Carefully read the AWS CLI License Agreement. If you agree to the terms, select the "I accept the terms in the License Agreement" checkbox and click "Next."

Step 6: Decide whether to use the default installation directory or specify a custom one. Click "Next" to continue.

Step 7: Determine the folder where AWS CLI shortcuts will be placed in the Start menu. Click "Next."

Step 8: You can opt to create desktop and Start menu shortcuts for the AWS CLI. Make your selections and click "Next."

Step 9: Review your chosen settings. If they are correct, click "Install" to initiate the installation process.

Step 10: The installer will copy the necessary files and install the AWS CLI on your Windows system.

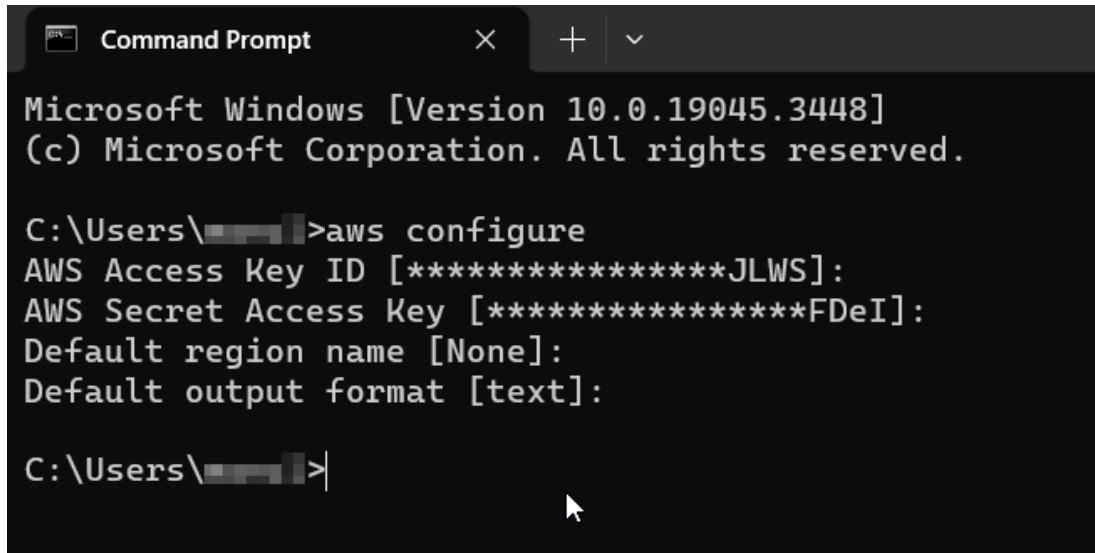
Step 11: Once the installation is finished, click "Finish" to exit the installer.

Configuring AWS CLI on Windows:

Step 1: Launch a Command Prompt window on your windows computer by pressing `Win + R`, typing "cmd," and pressing Enter.

Step 2: In the Command Prompt, enter the following command:

“aws configure“



```
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]>aws configure
AWS Access Key ID [*****JLWS]:
AWS Secret Access Key [*****FDeI]:
Default region name [None]:
Default output format [text]:

C:\Users\[redacted]>
```

Step 3: You will be prompted to provide the following information:

- Access Key ID: Enter the AWS Access Key ID obtained during the IAM user setup.
- Secret Access Key: Input the AWS Secret Access Key corresponding to the Access Key ID.

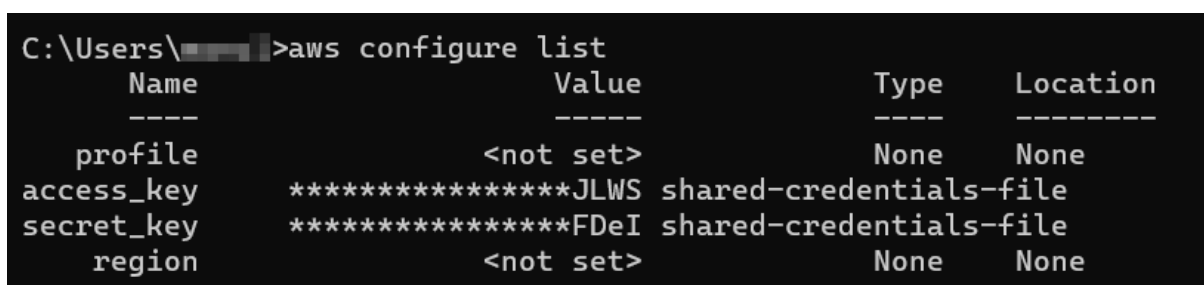
Optionally, you may configure the following settings based on your preferences. These settings are optional and can be left blank:

- Default region: Specify your preferred AWS region (e.g., us-east-1).
- Preferred output format: Select your desired output format (e.g., Json)

Step 4: To verify the AWS CLI configuration, run the following command:

“aws configure list“

This command will display the configuration settings you entered.



```
C:\Users\[redacted]>aws configure list
```

Name	Value	Type	Location
profile	<not set>	None	None
access_key	*****JLWS	shared-credentials-file	
secret_key	*****FDeI	shared-credentials-file	
region	<not set>	None	None

Step 5: Confirm that the AWS CLI is correctly configured and able to communicate with AWS services by listing the contents of your S3 buckets:

“aws s3 ls”

If your S3 bucket names are listed, your AWS CLI configuration is successful.

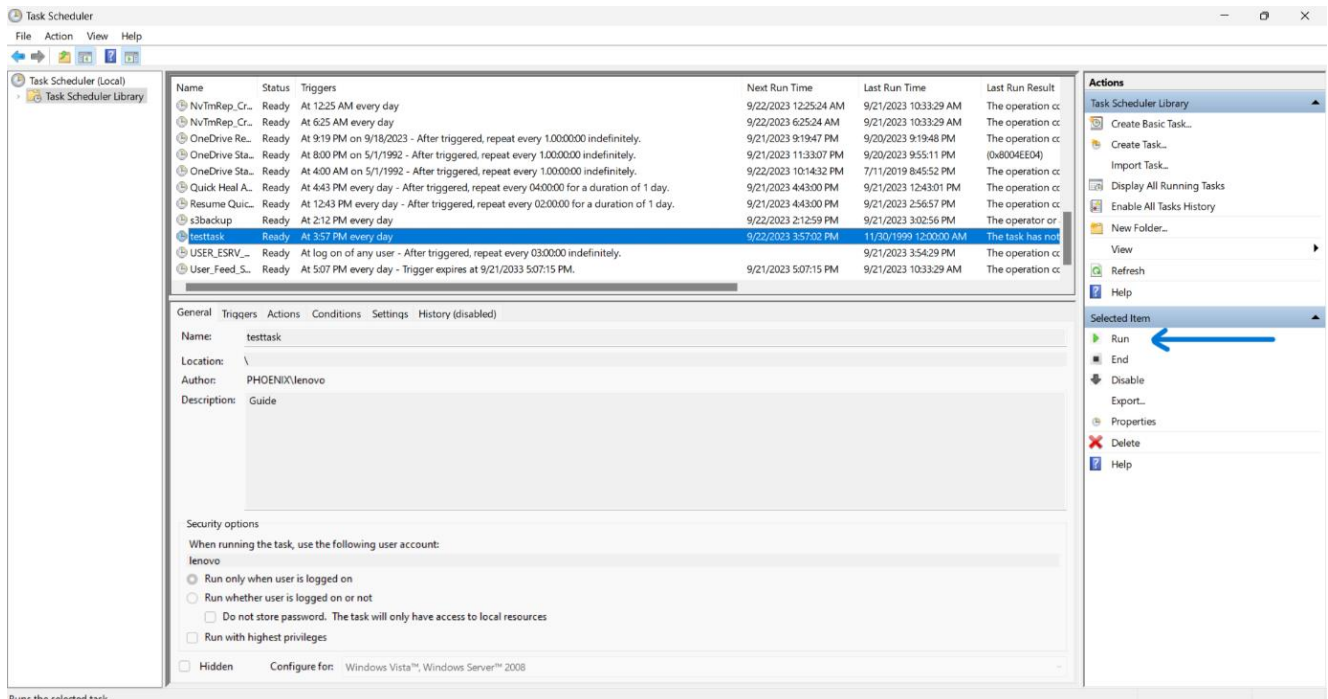
```
C:\Users\>aws s3 ls
2023-09-21 15:47:11 bucketcreationdemo

```

Run the Scheduled Task

Make sure that the scheduled task you created in **Schedule Backups with Task Scheduler** section runs as planned. This task will automatically execute your MongoDB backup script at the scheduled intervals.

For Testing Purposes you can find the created Task in the Task Scheduler Library and then can RUN it in the Actions Panel and check whether the backup is created on s3 or not.



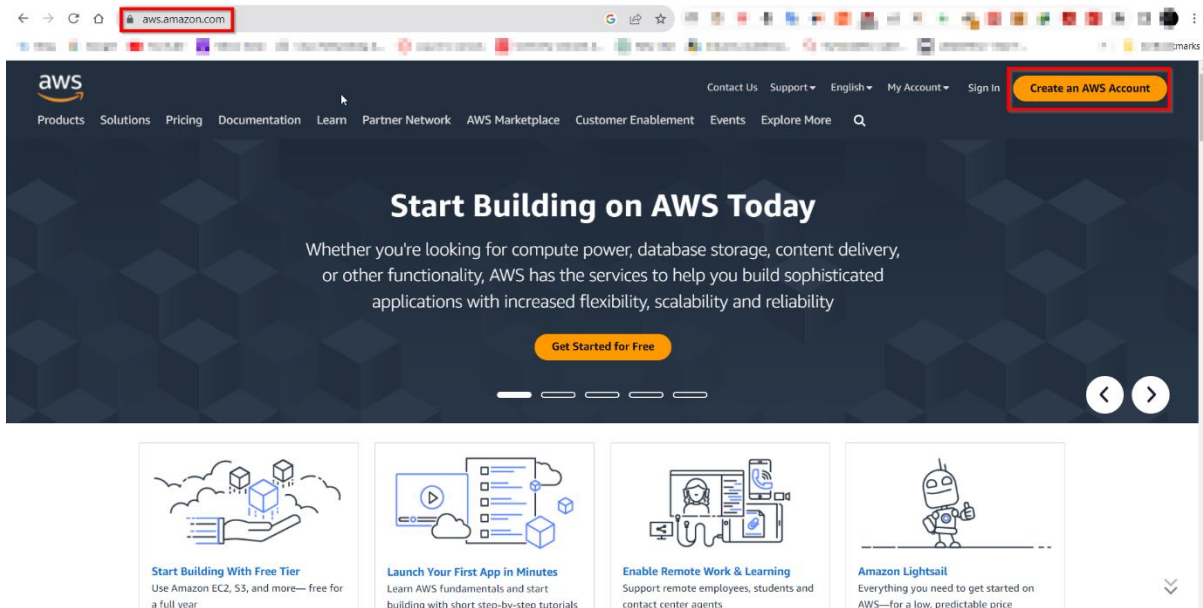
TECHNIQUE - 2

AMAZON EC2 (UBUNTU VERSION) USING CRON

AWS Setup Guide

Step 1: Visit the AWS Website - Navigate to the official AWS website: (<https://aws.amazon.com/>).

Step 2: Click the "Create an AWS Account" button.

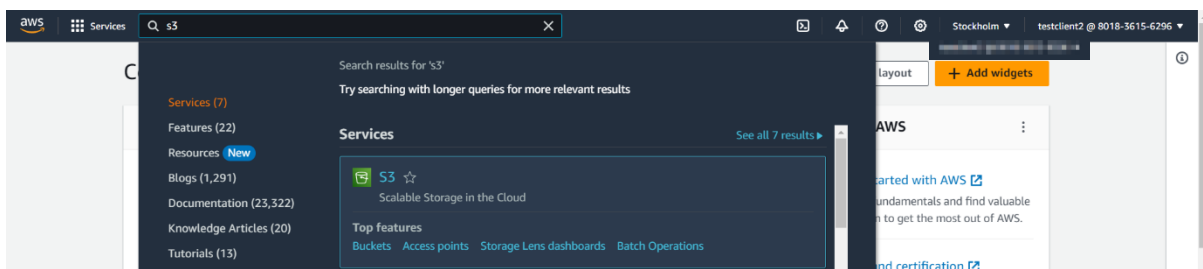


Step 3: Follow the guided registration process, which includes providing personal and payment information. Upon completion, you will have successfully created your AWS account.

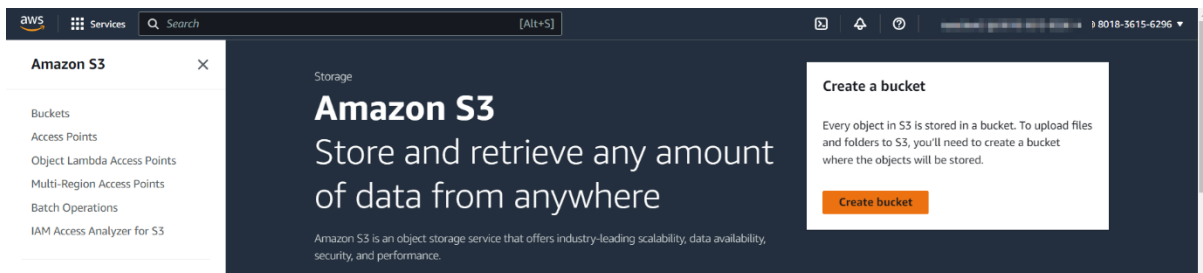
Step 4: After successfully creating your AWS account, log in to the AWS Management Console.

Create Bucket using S3:

Step 1: In the AWS Management Console, locate and select the "S3" service.

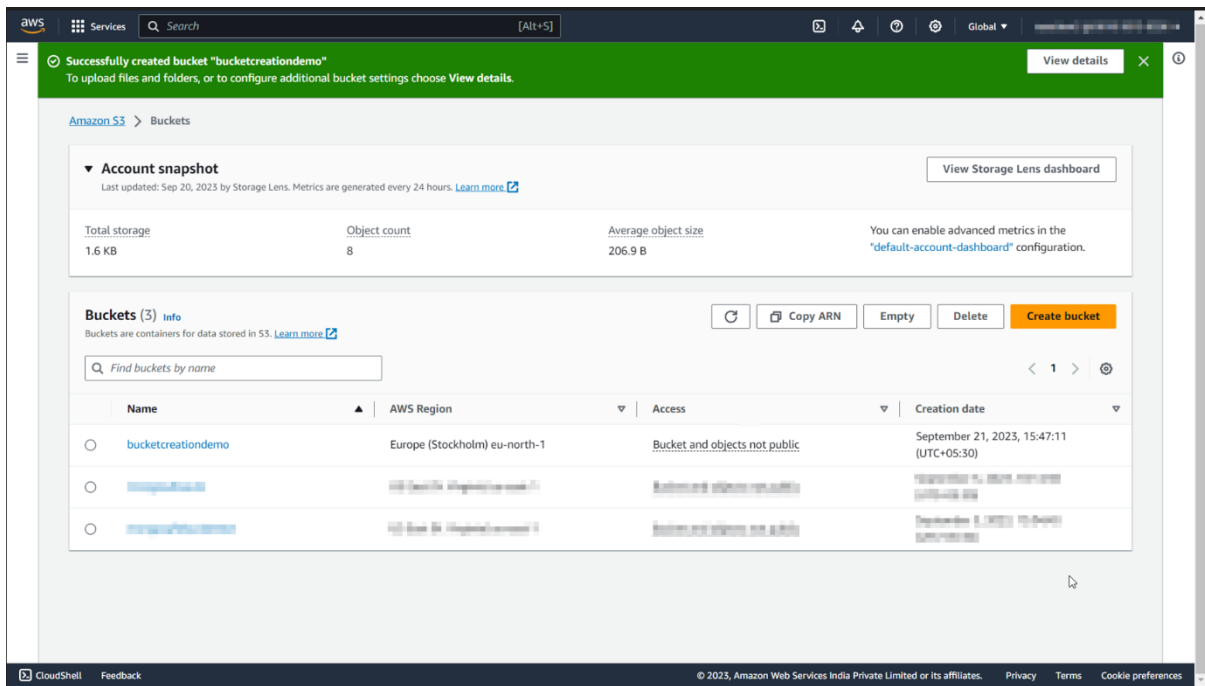


Step 2: Click the "Create Bucket" button.



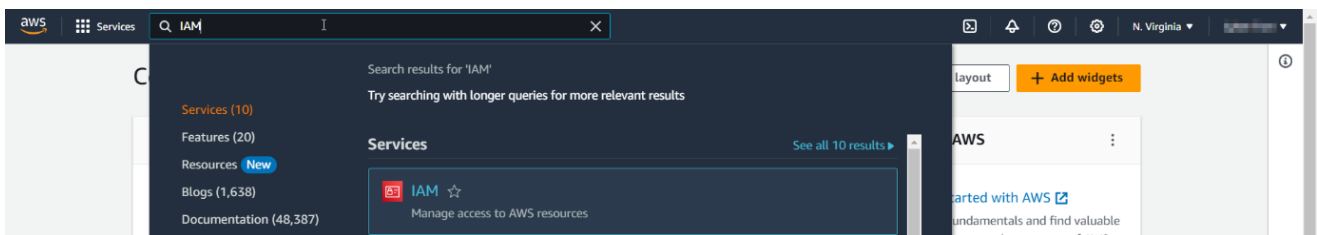
Step 3: Follow the steps in the bucket creation wizard, ensuring to provide a unique and meaningful bucket name.

Step 4: Configure any desired settings as needed during the setup process.



Create an IAM User with S3 Full Access:

Step 1: In the AWS Management Console, find and select the "Identity and Access Management (IAM)" service.



Step 2: Within the IAM console, create a new IAM user, specifying the user's details. Choose "programmatic access" when prompted.

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Step 3: When configuring permissions for the user, grant them "S3 Full Access" permissions, ensuring they have the necessary privileges.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1126)

Choose one or more policies to attach to your new user.

Filter by Type: All types 5 matches

Policy name	Type	Attached entities
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	2
<input type="checkbox"/> AmazonS3ObjectLambdaExecution...	AWS managed	0
<input type="checkbox"/> AmazonS3OutpostsFullAccess	AWS managed	0
<input type="checkbox"/> AmazonS3OutpostsReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	0

Step 4: After successfully creating the IAM user, make note of the "Access Key ID" and "Secret Access Key" assigned to this user. These credentials must be kept secure, as they will be required for configuring the AWS Command Line Interface (CLI).

The screenshot shows the AWS IAM console interface. At the top, a green banner reads "Access key created" with a warning: "This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time." Below this, the breadcrumb navigation is "IAM > Users > taskuser > Create access key". The main heading is "Retrieve access keys". A table displays the generated credentials:

Access key	Secret access key
AKIA3VMJIYWEM6HPV2PB	oRwMU9Ec3FcgpcubgGpRnyP1qb9mDbHIUYXWk2Zz

Console password

You have successfully enabled the user's new password.
This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

Console sign-in URL
https://801836156296.signin.aws.amazon.com/console

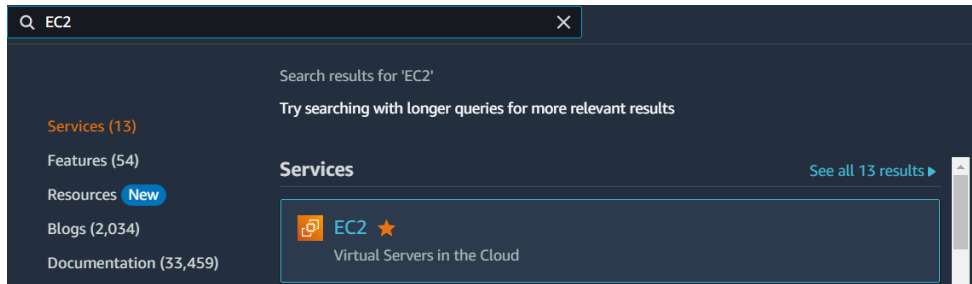
User name
taskuser

Console password
W=c@GP2* [Hide](#)

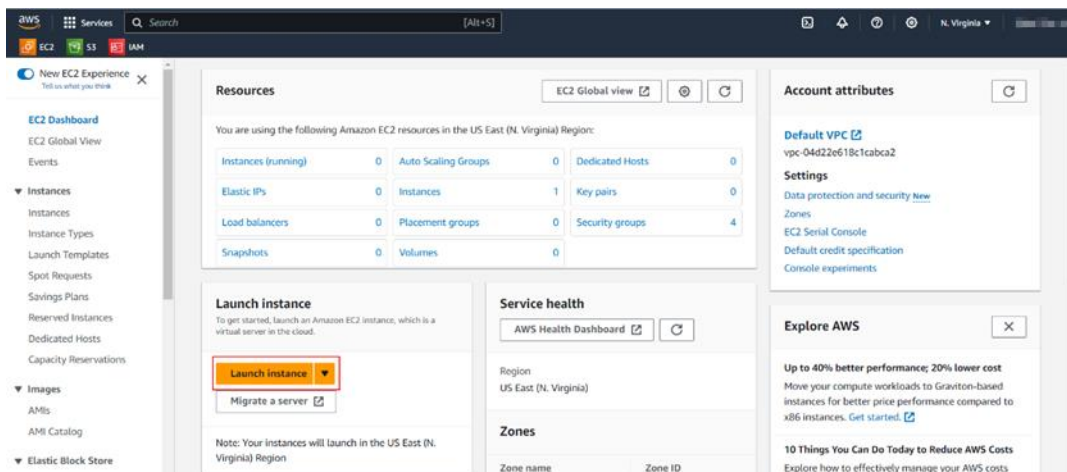
[Download .csv file](#) [Close](#)

Create an EC2 Instance:

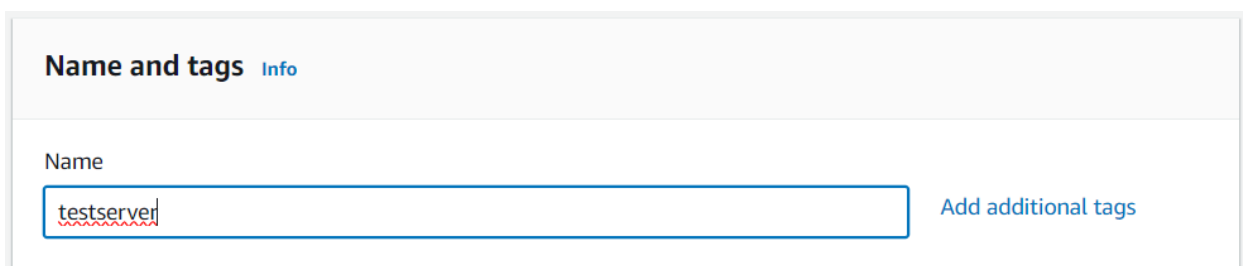
Step 1: Go to EC2 in services section.



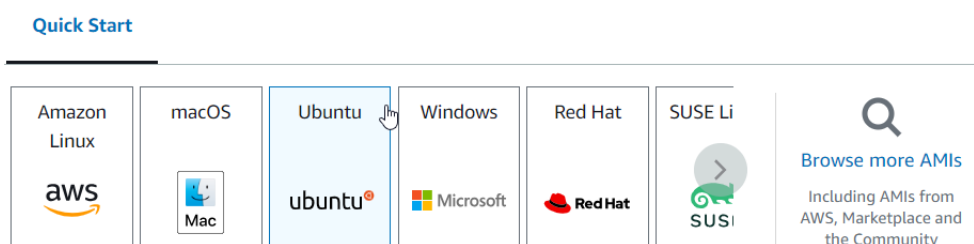
Step 2: Click on Launch instance in the Launch instance section.



Step 3: Write the name of Instance.



Step 4: Choose the operating system as Ubuntu.



Step 5: Create a new key pair.

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select ▼ ↻ Create new key pair

Step 6: Name the key pair and choose .pem as a file format and click on create key pair.

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

testkey

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Cancel ▶ Create key pair

Step 7: Click on launch instance, then go to view all instance.

Step 8: After the instance state comes to running, choose your create instance and click on connect.

Instances (1/2) [Info](#) ↻ Connect Instance state ▼ Actions ▼ ▶ Launch instances ▼

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
testserver	i-05151e055a077574e	Running	t2.micro	-	No alarms	us-east-1a	ec2-54-224-97-1

Step 9: Choose how you want to connect with the instance, Here I am using EC2 Instance Connect.

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-05151e055a077574e (testserver)

Connection Type

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
54.224.97.147

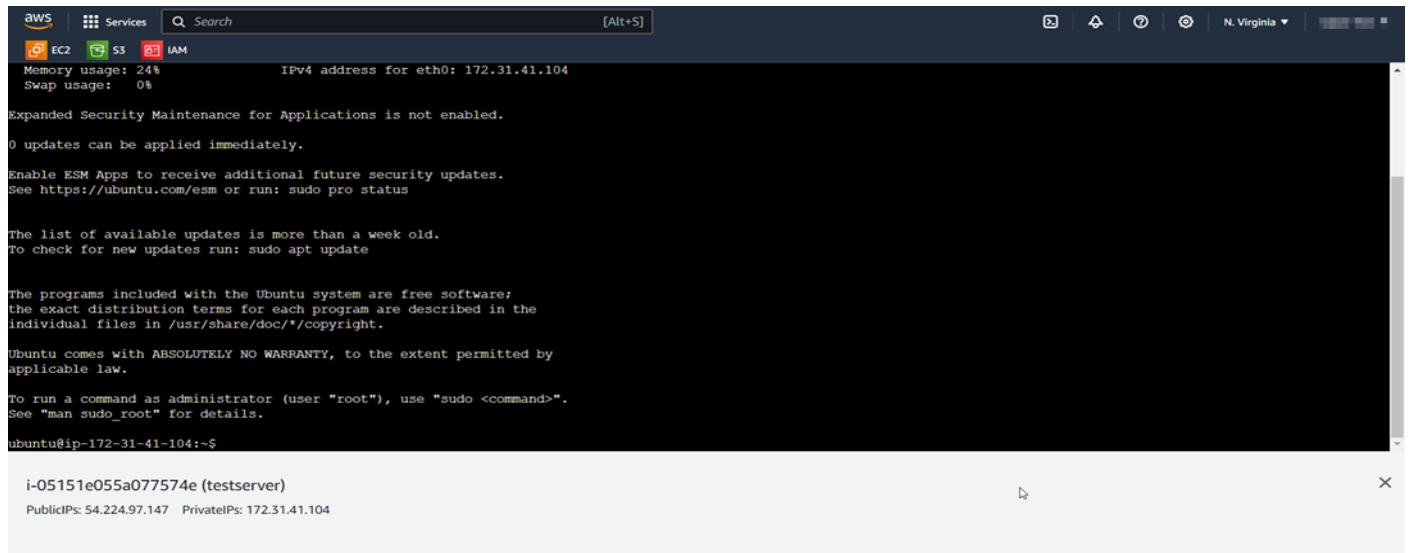
User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

Configuring EC2 instance



```
aws
Services
Search [Alt+S]
EC2 S3 IAM
Memory usage: 24% IPv4 address for eth0: 172.31.41.104
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-172-31-41-104:~$
i-05151e055a077574e (testserver)
PublicIPs: 54.224.97.147 PrivateIPs: 172.31.41.104
```

Step 1: Update and upgrade your system:

Use the following commands:

- `sudo apt update`
- `sudo apt upgrade`

Step 2: Add the MongoDB repository:

- `sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu focal main"`

Step 3: Add the MongoDB GPG key and update again:

- `wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add - echo "deb [arch=amd64, arm64] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list`
- `sudo apt update`

Step 4: Install MongoDB:

- `sudo apt-get install -y mongodb-org`

Step 5: Start and enable MongoDB service:

- `sudo systemctl start mongod`
- `sudo systemctl enable mongod`

Step 6: Install AWS CLI:

- `sudo apt-get install -y awscli`

Step 7: Configure AWS CLI:

- aws configure

Follow the prompts to enter your AWS access key, secret key, default region, and output format.

Step 8: Create a backup script:

- cd /home/ubuntu
- nano backup_mongodb.sh
- Copy and paste the script you are provided along with other documents.

Step 9: Make the script executable:

- chmod +x backup_mongodb.sh

Step 10: Schedule backups using cron:

- crontab -e

Add the following line to schedule daily backups at 2 AM:

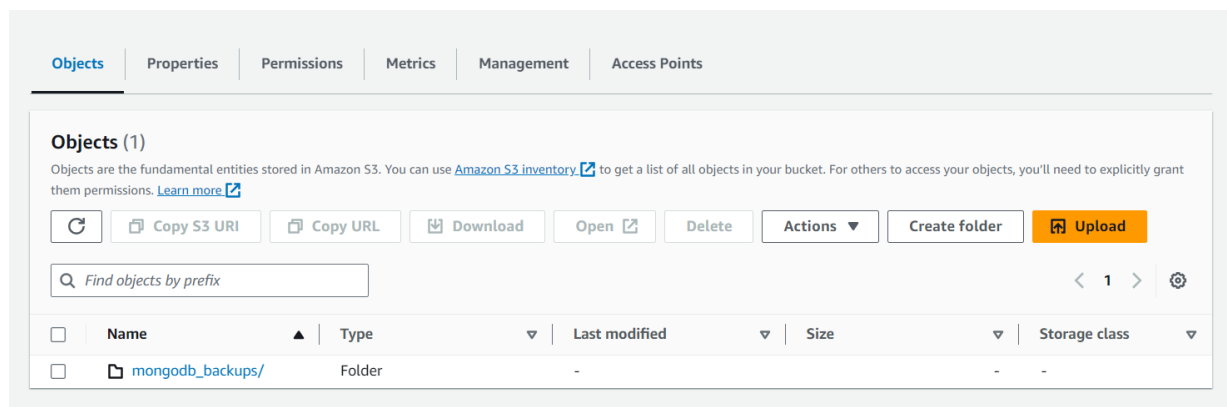
- 0 2 * * * /home/ubuntu/backup_mongodb.sh
- Save and exit the editor.

Step 11: Test the backup script manually:

```
ubuntu@ip-172-31-39-224:~$ ./backup_mongodb.sh
2023-09-23T15:11:33.651+0000   writing admin.system.version to /home/ubuntu/mongodb_backups/mongodb_backup_20230923151133/admin/system.version.bson
2023-09-23T15:11:33.658+0000   done dumping admin.system.version (1 document)
tar: Removing leading '/' from member names
/home/ubuntu/mongodb_backups/mongodb_backup_20230923151133/
/home/ubuntu/mongodb_backups/mongodb_backup_20230923151133/admin/
/home/ubuntu/mongodb_backups/mongodb_backup_20230923151133/admin/system.version.metadata.json
/home/ubuntu/mongodb_backups/mongodb_backup_20230923151133/admin/system.version.bson
upload: mongodb_backups/mongodb_backup_20230923151133.tar.gz to s3://testbucketformongo05/mongodb_backups/mongodb_backup_20230923151133.tar.gz
ubuntu@ip-172-31-39-224:~$
```

- ./backup_mongodb.sh

Step 12: Go to S3 Bucket and check the created backup.



Conclusion

In conclusion, this comprehensive guide has simplified the process of setting up and using MongoSafeNet for technical & non-technical users. It highlights the importance of regular backups for data security and recovery. By following these detailed steps, you can effectively manage your MongoDB databases and automate backups with confidence.

In the ever-evolving landscape of data management and cloud computing, the MongoSafenet project stands as a testament to innovation and efficiency. With a primary focus on automating MongoDB backups and seamlessly integrating with Amazon Web Services (AWS) S3, this project offers a comprehensive solution to address the challenges of data backup and cloud storage.

As we conclude our documentation guide on MongoSafenet, it is essential to reflect on the key takeaways and the significance of this project:

- 1. Streamlined MongoDB Backup:** MongoSafenet simplifies the process of MongoDB backup by automating it through the Windows Task Scheduler. This not only reduces the risk of data loss but also ensures that backups are executed consistently and reliably.
- 2. AWS Integration:** Leveraging the power of AWS S3, MongoSafenet provides a secure and scalable cloud storage solution for MongoDB backups. Organizations can now benefit from the durability, accessibility, and cost-effectiveness of AWS cloud storage.
- 3. Security:** The project emphasizes security by creating a dedicated IAM user with precise permissions, ensuring that only authorized entities can access and manage the AWS S3 bucket. Data integrity and confidentiality are paramount.
- 4. Operational Efficiency:** By automating MongoDB backup and cloud storage, MongoSafenet enables organizations to save valuable time and resources. IT teams can focus on higher-value tasks while having confidence in their data backup strategy.
- 5. Technology Stack:** MongoSafenet harnesses a powerful technology stack, including MongoDB, MongoDB Compass, Windows Task Scheduler, AWS CLI, AWS S3, and IAM, to create a seamless and robust solution.
- 6. Scalability and Future-Proofing:** As organizations grow, their data management needs evolve. The cloud-based approach of MongoSafenet ensures scalability, adaptability, and readiness for future data challenges.
- 7. Data Resilience:** MongoDB backups stored in AWS S3 benefit from the inherent data resilience of Amazon's infrastructure. In the event of data loss or system failures, backups remain accessible and recoverable.

MongoSafenet represents a forward-thinking solution that empowers organizations to harness the full potential of MongoDB while ensuring the security and accessibility of their data through AWS S3. By automating backups and embracing cloud storage, organizations can mitigate risks, optimize operations, and confidently navigate the complexities of modern data management.

As you embark on your journey to implement MongoSafenet, this documentation guide serves as a comprehensive resource to guide you through the installation, configuration, and deployment process. We hope that MongoSafenet becomes a valuable addition to your data management toolkit, safeguarding your MongoDB data and enabling your organization to thrive in a data-driven world.