**Mongodbbackup specific DB &Collection S3 using Corn Job**

**Prerequisite :** ec2 instance

**Type :** t3.2xlarge are more

**volume :** 100 gb base on usage

**install docker :** apt-install [docker.io](http://docker.io) and docker compose

**install python**

### **Requirements**

AWS CLI configured (aws configure)  
Access to MongoDB tools like mongodump and oplog for incremental Cron installed

## **Python Scripts**

If you’d rather have Python handle “dump + upload” (perhaps to add more logic later), you can use the AWS SDK (boto3) in concert with a subprocess‐based mongodump. Below are two standalone Python scripts—one for full and one for oplog (incremental). You’ll still call them from cron every 5 minutes.

version: '3.8'

services:

mongodb1:

image: mongo:6.0

container\_name: mongodb1

restart: always

ports:

- "27017:27017"

volumes:

- mongo\_data1:/data/db

- mongo\_config1:/data/configdb

- ./mongo-keyfile:/data/keyfile # Mount keyfile

command: ["mongod", "--replSet", "rs0", "--bind\_ip", "0.0.0.0", "--keyFile", "/data/keyfile"]

environment:

MONGO\_INITDB\_ROOT\_USERNAME: root

MONGO\_INITDB\_ROOT\_PASSWORD: rootpassword

mongodb2:

image: mongo:6.0

container\_name: mongodb2

restart: always

ports:

- "27018:27017"

volumes:

- mongo\_data2:/data/db

- mongo\_config2:/data/configdb

- ./mongo-keyfile:/data/keyfile # Mount keyfile

command: ["mongod", "--replSet", "rs0", "--bind\_ip", "0.0.0.0", "--keyFile", "/data/keyfile"]

environment:

MONGO\_INITDB\_ROOT\_USERNAME: root

MONGO\_INITDB\_ROOT\_PASSWORD: rootpassword

mongodb3:

image: mongo:6.0

container\_name: mongodb3

restart: always

ports:

- "27019:27017"

volumes:

- mongo\_data3:/data/db

- mongo\_config3:/data/configdb

- ./mongo-keyfile:/data/keyfile # Mount keyfile

command: ["mongod", "--replSet", "rs0", "--bind\_ip", "0.0.0.0", "--keyFile", "/data/keyfile"]

environment:

MONGO\_INITDB\_ROOT\_USERNAME: root

MONGO\_INITDB\_ROOT\_PASSWORD: rootpassword

volumes:

mongo\_data1:

mongo\_config1:

mongo\_data2:

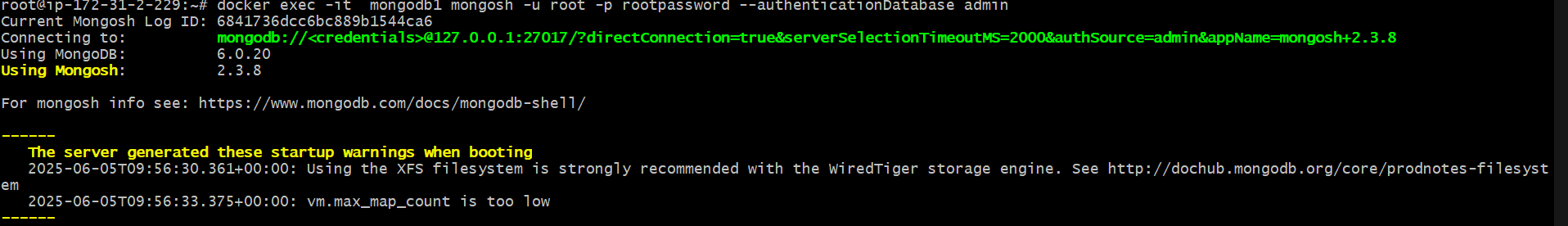
mongo\_config2:

mongo\_data3:

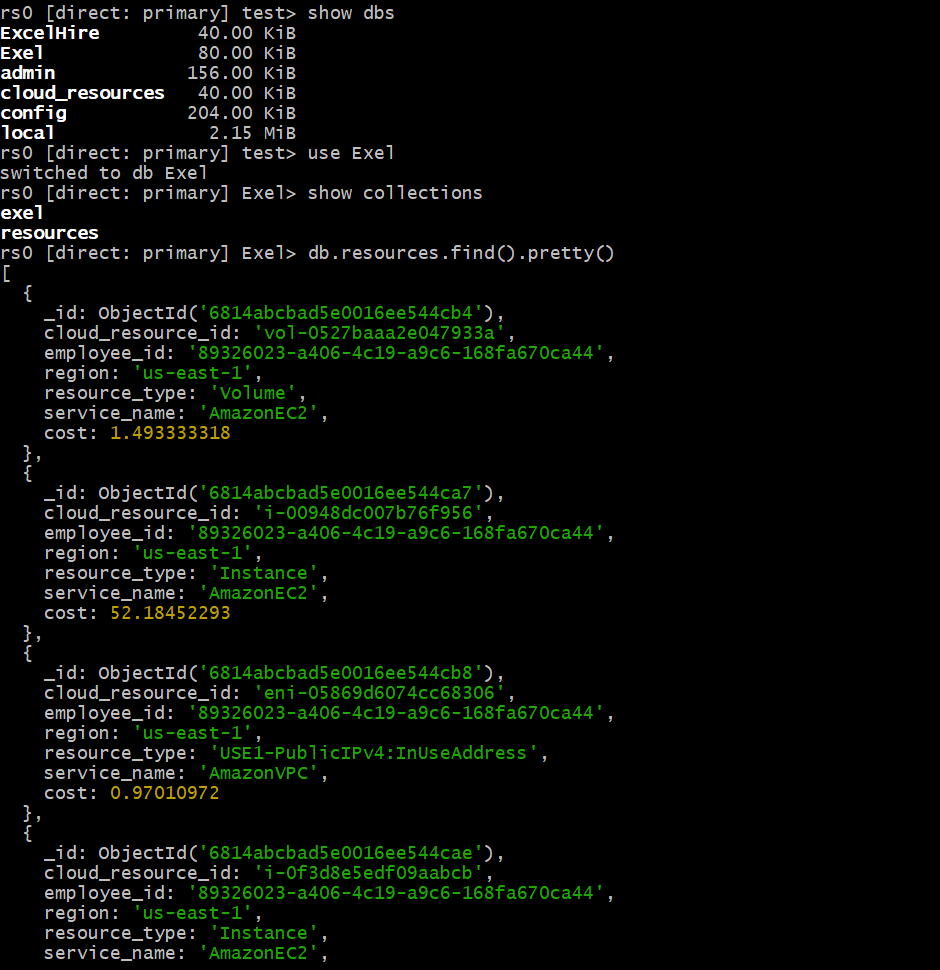
mongo\_config3:

**Step1 :**Login to Mongodb check your db and collection need to backup

docker exec -it mongodb1 mongosh -u root -p rootpassword --authenticationDatabase admin



**Step2**: show dbs → ,use <db> → , check collection ,-->db.resources.find().pretty()



**mongodb\_full\_backup.py**

#!/usr/bin/env python3

"""

mongodb\_full\_backup.py

Backs up the 'resources' collection from 'Exel' DB using mongodump, uploads to S3, then deletes local archive.

"""

import os

import subprocess

import boto3

from datetime import datetime

# CONFIG

CONTAINER = "mongodb1"

MONGO\_USER = "root"

MONGO\_PASS = "rootpassword"

MONGO\_AUTH\_DB = "admin"

DB\_NAME = "Exel"

COLLECTION = "resources"

S3\_BUCKET = "backupmongdb"

S3\_KEY\_PREFIX = "s3mongodbbackautomatic/full"

TMP\_DIR = "/tmp/mongodb\_backups"

os.makedirs(TMP\_DIR, exist\_ok=True)

def run\_full\_dump() -> str:

timestamp = datetime.utcnow().strftime("%Y-%m-%d\_%H-%M")

outfile = os.path.join(TMP\_DIR, f"full\_{DB\_NAME}\_{COLLECTION}\_{timestamp}.archive.gz")

cmd = [

"docker", "exec", CONTAINER,

"mongodump",

"--db", DB\_NAME,

"--collection", COLLECTION,

"--archive", "--gzip",

"--username", MONGO\_USER,

"--password", MONGO\_PASS,

"--authenticationDatabase", MONGO\_AUTH\_DB

]

with open(outfile, "wb") as f:

proc = subprocess.Popen(cmd, stdout=f, stderr=subprocess.PIPE)

\_, stderr = proc.communicate()

if proc.returncode != 0:

raise RuntimeError(f"mongodump failed: {stderr.decode('utf-8')}")

return outfile

def upload\_to\_s3(local\_path: str) -> None:

s3 = boto3.client("s3")

key = os.path.join(S3\_KEY\_PREFIX, os.path.basename(local\_path))

print(f"Uploading {local\_path} to s3://{S3\_BUCKET}/{key} ...")

s3.upload\_file(local\_path, S3\_BUCKET, key)

print("Upload complete.")

def main():

try:

local\_file = run\_full\_dump()

upload\_to\_s3(local\_file)

os.remove(local\_file)

except Exception as e:

print(f"[ERROR] {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**cat mongodb\_incremental\_backup.py**

#!/usr/bin/env python3

"""

mongodb\_incremental\_backup.py

Backs up the oplog.rs collection (incremental) and uploads to S3.

"""

import os

import subprocess

import boto3

from datetime import datetime

# CONFIG

CONTAINER = "mongodb1"

MONGO\_USER = "root"

MONGO\_PASS = "rootpassword"

MONGO\_AUTH\_DB = "admin"

S3\_BUCKET = "backupmongdb"

S3\_KEY\_PREFIX = "s3mongodbbackautomatic/incremental"

TMP\_DIR = "/tmp/mongodb\_backups"

os.makedirs(TMP\_DIR, exist\_ok=True)

def run\_oplog\_dump() -> str:

timestamp = datetime.utcnow().strftime("%Y-%m-%d\_%H-%M")

outfile = os.path.join(TMP\_DIR, f"oplog\_{timestamp}.archive.gz")

cmd = [

"docker", "exec", CONTAINER,

"mongodump",

"--db", "local",

"--collection", "oplog.rs",

"--archive", "--gzip",

"--username", MONGO\_USER,

"--password", MONGO\_PASS,

"--authenticationDatabase", MONGO\_AUTH\_DB

]

with open(outfile, "wb") as f:

proc = subprocess.Popen(cmd, stdout=f, stderr=subprocess.PIPE)

\_, stderr = proc.communicate()

if proc.returncode != 0:

raise RuntimeError(f"mongodump (oplog) failed: {stderr.decode('utf-8')}")

return outfile

def upload\_to\_s3(local\_path: str) -> None:

s3 = boto3.client("s3")

key = os.path.join(S3\_KEY\_PREFIX, os.path.basename(local\_path))

print(f"Uploading {local\_path} to s3://{S3\_BUCKET}/{key} ...")

s3.upload\_file(local\_path, S3\_BUCKET, key)

print("Upload complete.")

def main():

try:

local\_file = run\_oplog\_dump()

upload\_to\_s3(local\_file)

os.remove(local\_file)

except Exception as e:

print(f"[ERROR] {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

## 

## 

## 

## **1. Manual Execution**

To run the scripts manually from terminal:

# Run Full Backup Manually

**python3 /root/corn/mongodb\_full\_backup.py**

# Run Incremental Backup Manually

**python3 /root/corn/mongodb\_incremental\_backup.py**

**Cron Configuration**

# ┌───────────── minute (0 - 59)

# │ ┌───────────── hour (0 - 23)

# │ │ ┌───────────── day of month (1 - 31)

# │ │ │ ┌───────────── month (1 - 12)

# │ │ │ │ ┌───────────── day of week (0 - 6) (Sunday=0)

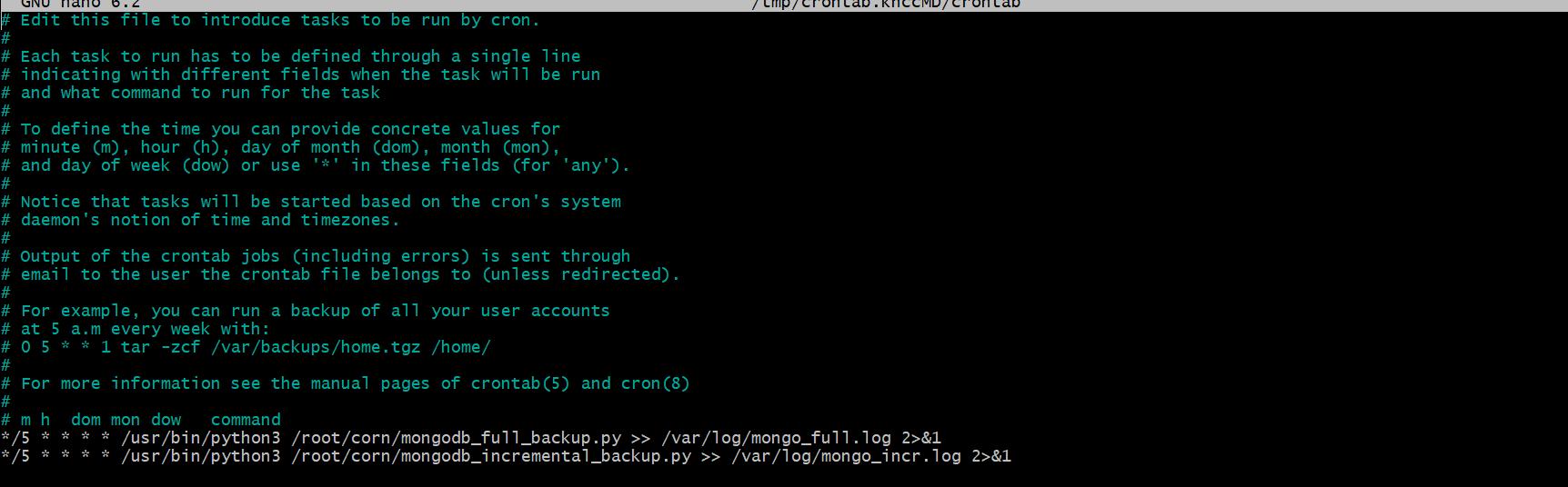
# │ │ │ │ │

# \* \* \* \* \* command to execute

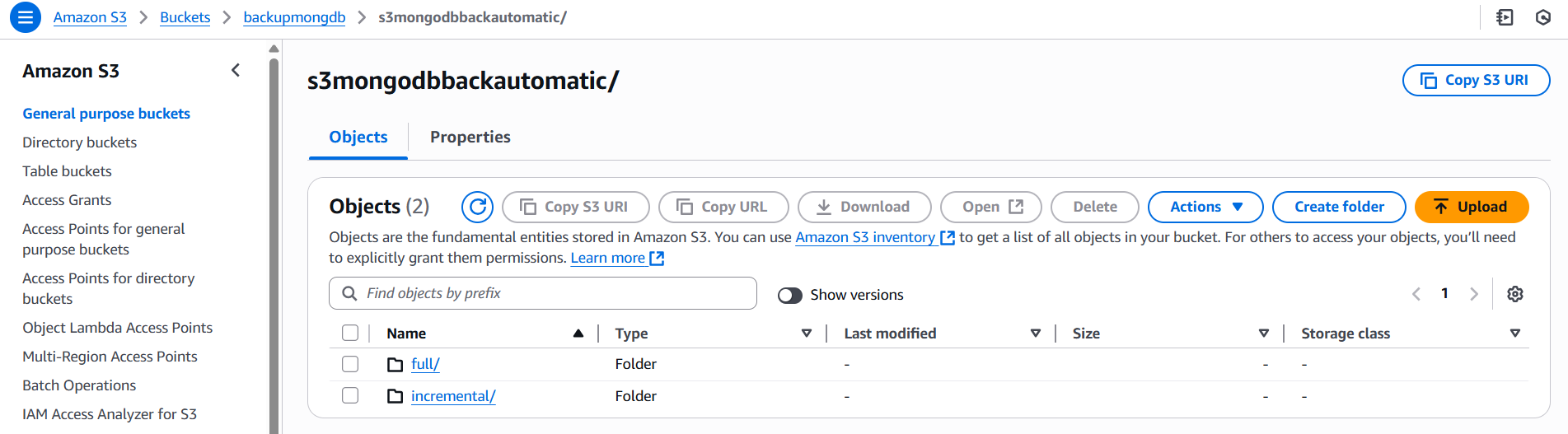
**\*/5 \* \* \* \* /usr/bin/python3 /root/corn/mongodb\_full\_backup.py >> /var/log/mongo\_full.log 2>&1**

**\*/5 \* \* \* \* /usr/bin/python3 /root/corn/mongodb\_incremental\_backup.py >> /var/log/mongo\_incr.log 2>&1**

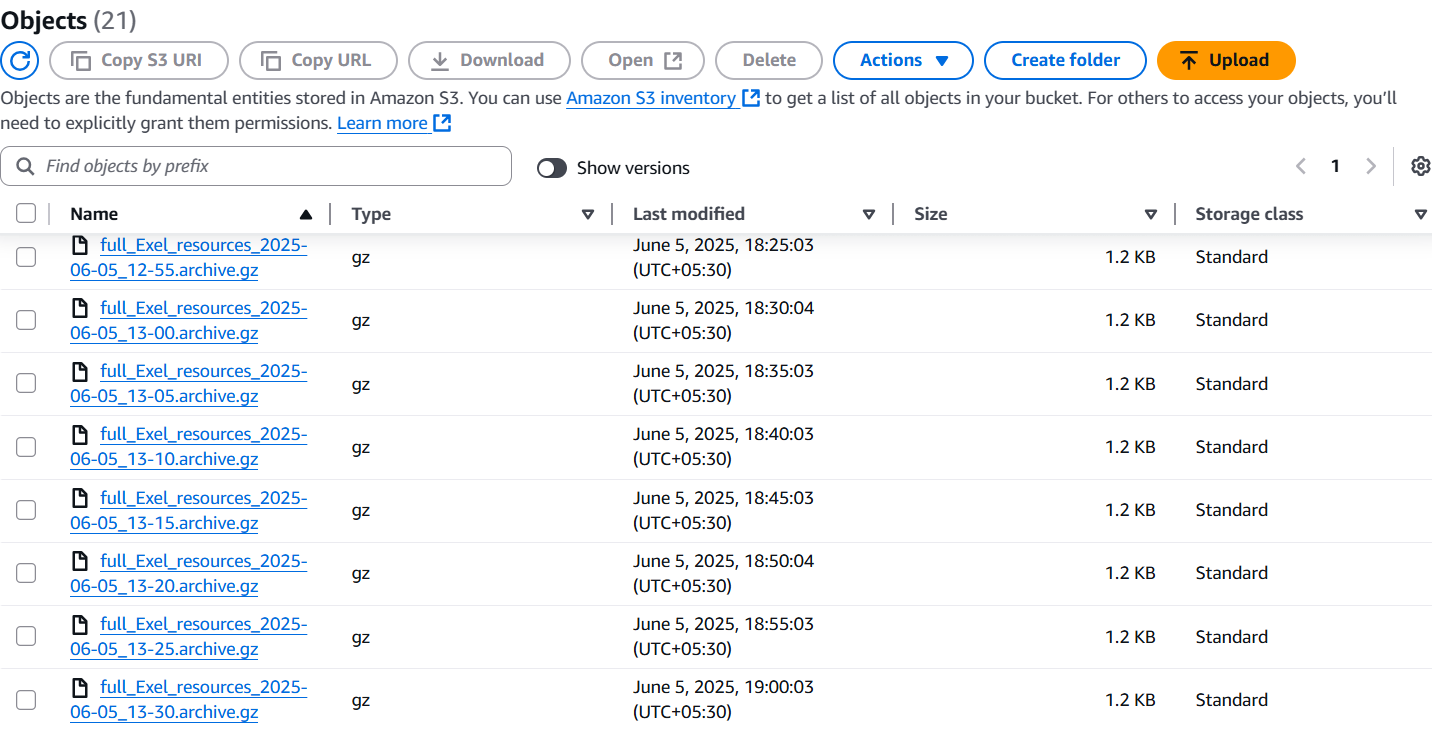
**sudo crontab -e**

****

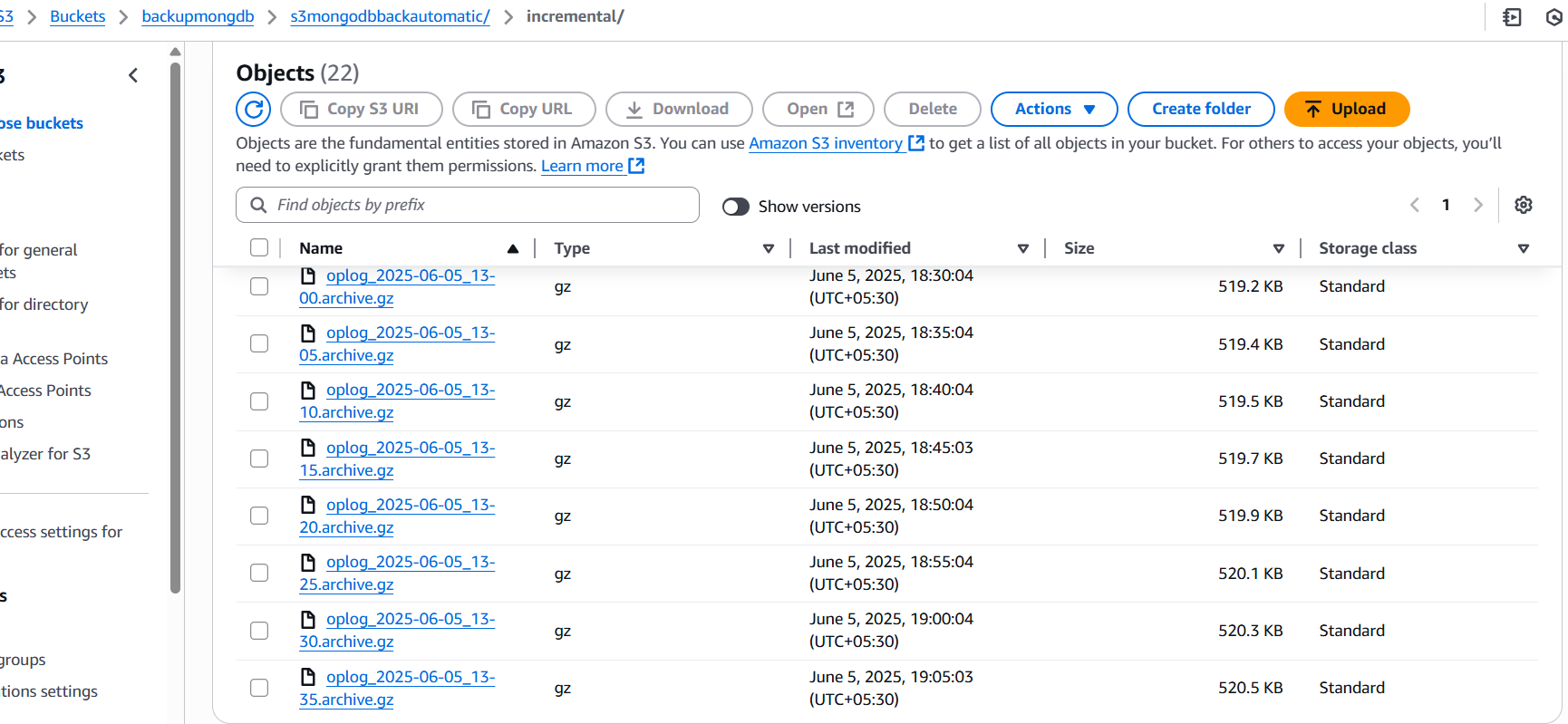
**It will automatically store in s3**

****

**Now full back store in s3 ever 5 mints data is**

****

**Incremental back up**

****

**Check the corn tap status & corn**

sudo systemctl status cron

sudo crontab -l

