Multi-Cloud Backup Strategy

Encrypted, Deduplicated Backups Across AWS, Azure & Google Cloud

Ahmed Abdelwahed
ahmed@abdelwahed.me
www.abdelwahed.me
LinkedIn

What This Lab Builds

In this lab you will build a real, production-style multi-cloud backup strategy that stores encrypted, deduplicated copies of your data in three independent cloud providers — Amazon Web Services (AWS), Microsoft Azure, and Google Cloud (GCP) — at the absolute lowest cost achievable inside their free tiers and cheapest storage classes.

The whole stack uses one open-source tool, Restic, plus the cloud-native object storage of each provider. There are no agents to license, no backup appliances to buy, and no central server to run — you can drive everything from a single workstation or a tiny VM.

Phase 1 — Install Restic

Restic ships as a single static binary. Install it via your package manager, or download directly.

Linux (Ubuntu / Debian)

```
sudo apt update
sudo apt install restic -y
```

Linux (RHEL / Rocky / CentOS)

```
sudo dnf install epel-release -y
sudo dnf install restic -y
```

macOS

```
brew install restic
```

Windows (WSL or PowerShell)

In WSL, use the Linux instructions. In native Windows, install via Chocolatey or Scoop:

```
choco install restic
# or
scoop install restic
```

Verify

```
restic version
```

Phase 2 — Prepare AWS S3

Create a private S3 bucket dedicated to backups, then a least-privilege IAM user with access only to that bucket.

Create the Bucket

Pick a globally unique name. Block all public access. Enable versioning for ransomware protection.

```
# Create the bucket (adjust region if needed)
aws s3api create-bucket \
  --bucket mc-backup-aws-ahmed \
  --region us-east-1
# Configure Public Access Block (no public access):
aws s3api put-public-access-block \
  --bucket mc-backup-aws-ahmed \
  --public-access-block-configuration '{
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  }'

# Enable bucket versioning:
aws s3api put-bucket-versioning \
  --bucket mc-backup-aws-ahmed \
  --versioning-configuration Status=Enabled
```

Lifecycle Policy — Cheap Storage Class

Auto-transition objects to Glacier Instant Retrieval after 1 day. Save as `lifecycle-aws.json`:

```
{
  "Rules": [{
    "ID": "to-glacier-ir",
    "Status": "Enabled",
    "Filter": { "Prefix": "" },
    "Transitions": [{
      "Days": 1,
      "StorageClass": "GLACIER_IR"
    }],
    "NoncurrentVersionExpiration": { "NoncurrentDays": 30 }
  }]
}

# Apply the lifecycle configuration:
aws s3api put-bucket-lifecycle-configuration \
  --bucket mc-backup-aws-ahmed \
  --lifecycle-configuration file://lifecycle-aws.json
```

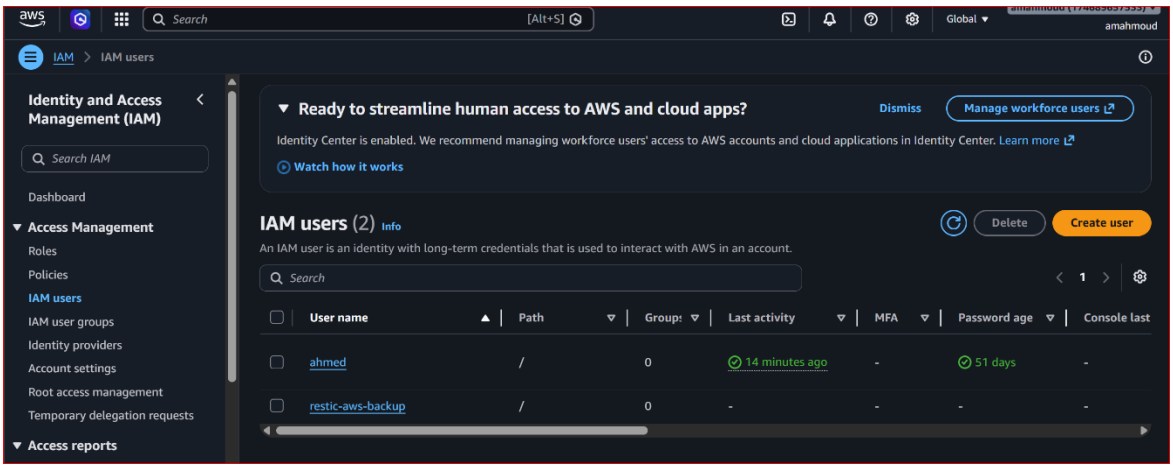
Create a Least-Privilege IAM User

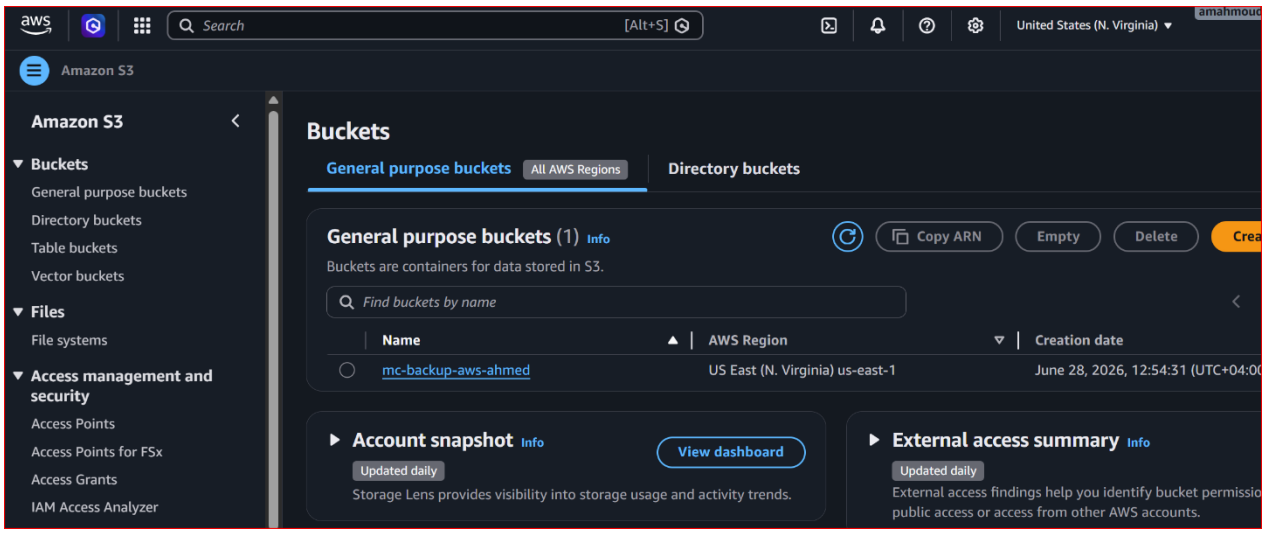
```
aws iam create-user \
  --user-name restic-aws-backup
aws iam create-access-key \
  --user-name restic-aws-backup
```

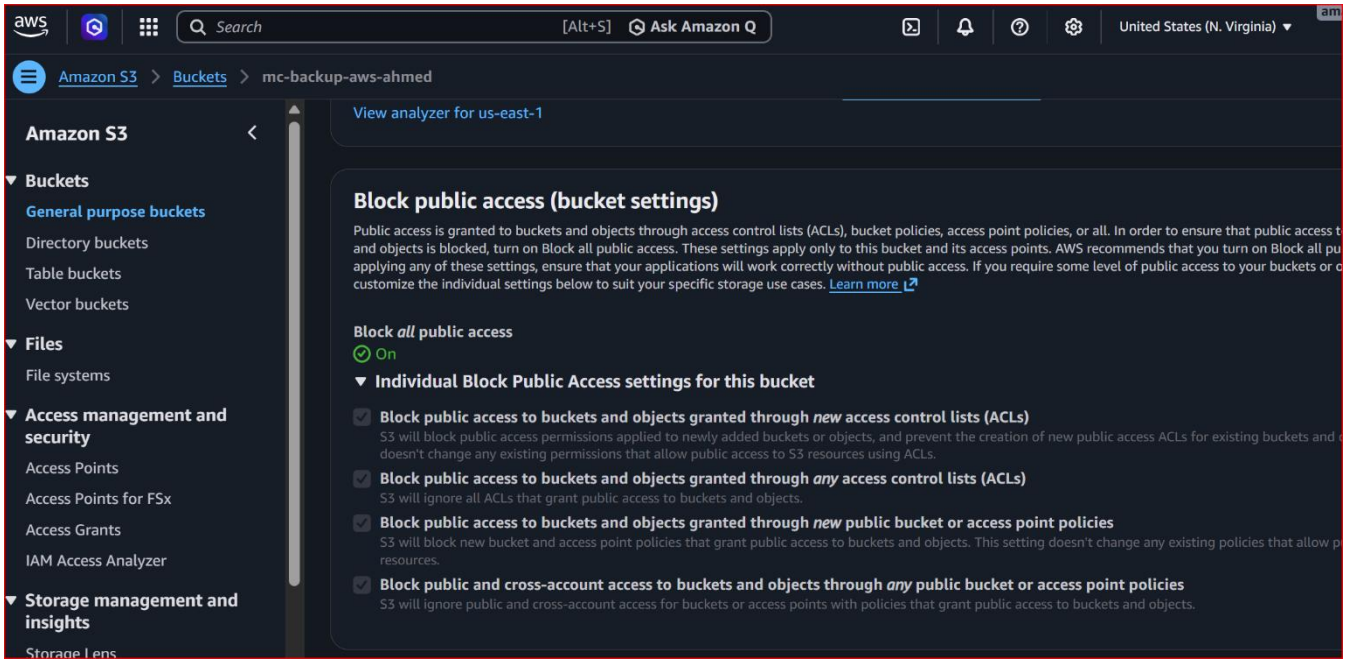
Save the AccessKeyId and SecretAccessKey returned. Attach a policy that only allows access to this one bucket (save as `policy-aws.json`):

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::mc-backup-aws-ahmed"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject",
      "s3:GetObject",
      "s3:DeleteObject"
    ],
    "Resource": "arn:aws:s3:::mc-backup-aws-ahmed/*"
  }]
}
```

```
# Attach the inline policy to the IAM user:
aws iam put-user-policy \
  --user-name restic-aws-backup \
  --policy-name resticBackupBucketAccess \
  --policy-document file://policy-aws.json
```


Search
[Alt+S]
Global
amahmoud
IAM > IAM users
Identity and Access Management (IAM)
Search IAM
Dashboard
Access Management
Roles
Policies
IAM users
IAM user groups
Identity providers
Account settings
Root access management
Temporary delegation requests
Access reports
Ready to streamline human access to AWS and cloud apps?
Dismiss
Manage workforce users
Identity Center is enabled. We recommend managing workforce users' access to AWS accounts and cloud applications in Identity Center. Learn more
Watch how it works
IAM users (2) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user
Search
User name
Path
Group
Last activity
MFA
Password age
Console last
ahmed
/
0
14 minutes ago
-
51 days
-
restic-aws-backup
/
0
-
-
-
-


Search
[Alt+S]
United States (N. Virginia)
Amazon S3
Amazon S3
Buckets
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Files
File systems
Access management and security
Access Points
Access Points for FSx
Access Grants
IAM Access Analyzer
Buckets
General purpose buckets
All AWS Regions
Directory buckets
General purpose buckets (1) Info
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Find buckets by name
Name
AWS Region
Creation date
mc-backup-aws-ahmed
US East (N. Virginia) us-east-1
June 28, 2026, 12:54:31 (UTC+04:00
Account snapshot Info
Updated daily
View dashboard
Storage Lens provides visibility into storage usage and activity trends.
External access summary Info
Updated daily
External access findings help you identify bucket permissio
public access or access from other AWS accounts.


Search
[Alt+S]
Ask Amazon Q
United States (N. Virginia)
Amazon S3 > Buckets > mc-backup-aws-ahmed
Amazon S3
Buckets
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Files
File systems
Access management and security
Access Points
Access Points for FSx
Access Grants
IAM Access Analyzer
Storage management and insights
View analyzer for us-east-1
Block public access (bucket settings)
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 t
and 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 pu
applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or c
customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
On
Individual Block Public Access settings for this bucket
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
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 p
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.

Phase 3 — Prepare Azure Blob Storage

Create a storage account, a private container, and a SAS key or storage account key for Restic.

Create Resource Group and Storage Account

```
az login

az group create \
  --name mc-backup-rg \
  --location uaenorth

az storage account create \
  --name mcbackupazureahmed \
  --resource-group mc-backup-rg \
  --location uaenorth \
  --sku Standard_LRS \
  --kind StorageV2 \
  --access-tier Cool \
  --allow-blob-public-access false
```

Standard_LRS (locally redundant) is the cheapest option. Cool access tier is the right starting point for backups.

Create the Container

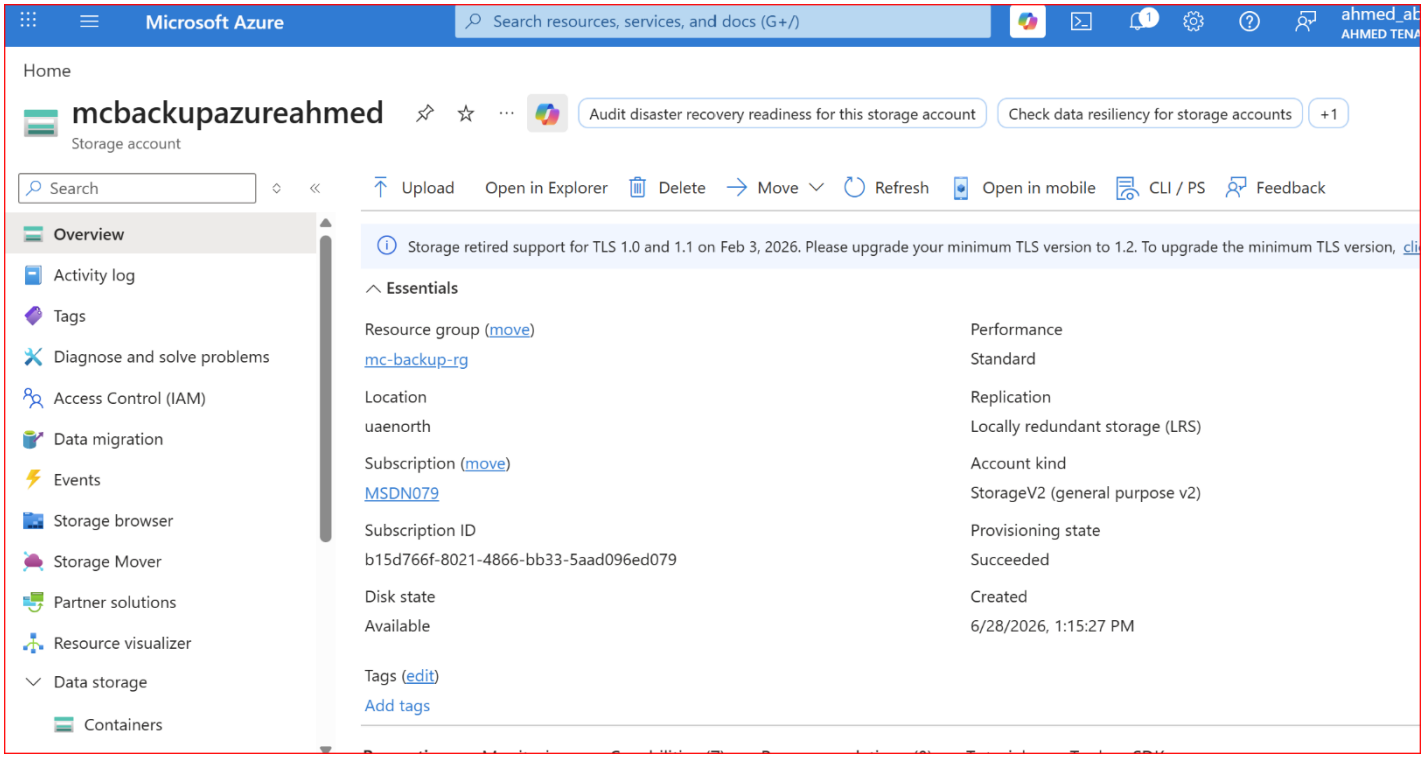
```
ACCOUNT_KEY=$(az storage account keys list \
  --account-name mcbackupazureahmed \
  --resource-group mc-backup-rg \
  --query '[0].value' -o tsv)

az storage container create \
  --name restic \
  --account-name mcbackupazureahmed \
  --account-key "$ACCOUNT_KEY" \
  --public-access off
```

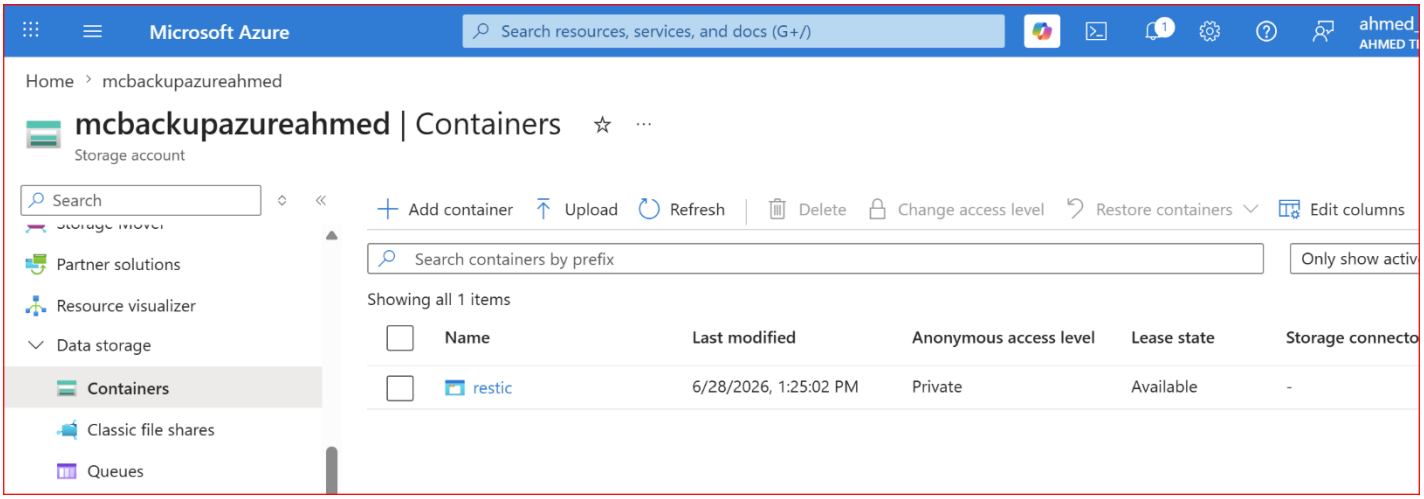
Lifecycle Policy — Move to Archive

Save as `lifecycle-azure.json` to auto-tier blobs to Archive after 7 days:

```
{
  "rules": [{
    "enabled": true,
    "name": "toArchive",
    "type": "Lifecycle",
    "definition": {
      "actions": {
        "baseBlob": {
          "tierToCool":    { "daysAfterModificationGreaterThan": 1 },
          "tierToArchive": { "daysAfterModificationGreaterThan": 7 }
        }
      },
      "filters": { "blobTypes": [ "blockBlob" ] }
    }
  }]
}
```

```
az storage account management-policy create \
  --account-name mcbackupazureahmed \
  --resource-group mc-backup-rg \
  --policy @lifecycle-azure.json
```


Microsoft Azure
Search resources, services, and docs (G+/)
Home
mcbackupazureahmed
Storage account
Audit disaster recovery readiness for this storage account
Check data resiliency for storage accounts
+1
Search
Upload
Open in Explorer
Delete
Move
Refresh
Open in mobile
CLI / PS
Feedback
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Storage Mover
Partner solutions
Resource visualizer
Data storage
Containers
Storage retired support for TLS 1.0 and 1.1 on Feb 3, 2026. Please upgrade your minimum TLS version to 1.2. To upgrade the minimum TLS version,
Essentials
Resource group (move)
mc-backup-rg
Location
uaenorth
Subscription (move)
MSDN079
Subscription ID
b15d766f-8021-4866-bb33-5aad096ed079
Disk state
Available
Performance
Standard
Replication
Locally redundant storage (LRS)
Account kind
StorageV2 (general purpose v2)
Provisioning state
Succeeded
Created
6/28/2026, 1:15:27 PM
Tags (edit)
Add tags


Microsoft Azure
Search resources, services, and docs (G+/)
Home
mcbackupazureahmed
mcbackupazureahmed | Containers
Storage account
Search
Add container
Upload
Refresh
Delete
Change access level
Restore containers
Edit columns
Search containers by prefix
Showing all 1 items
Partner solutions
Resource visualizer
Data storage
Containers
Classic file shares
Queues
Name
Last modified
Anonymous access level
Lease state
restic
6/28/2026, 1:25:02 PM
Private
Available

Phase 4 — Prepare Google Cloud Storage

Authenticate and Set Project

```
gcloud auth login
gcloud config set project YOUR_PROJECT_ID
```

7.2 Create the Bucket as Archive Class

```
gcloud storage buckets create gs://mc-backup-gcs-ahmed \
  --location=US \
  --default-storage-class=ARCHIVE \
  --uniform-bucket-level-access \
  --public-access-prevention
```

US multi-region keeps the bucket inside the always-free GCS allowance. Archive class is roughly $0.0012 per GB-month — the cheapest option in GCS.

Lifecycle Rule — Delete Old Noncurrent Versions

Save as `lifecycle-gcs.json`:

```
{
  "lifecycle": {
    "rule": [{
      "action": { "type": "Delete" },
      "condition": {
        "daysSinceNoncurrentTime": 30,
        "isLive": false
      }
    }]
  }
}
```

```
gcloud storage buckets update gs://mc-backup-gcs-ahmed --lifecycle-file=lifecycle-gcs.json
```

Create a Service Account for Restic

```
gcloud iam service-accounts create restic-gcs-backup \
  --display-name="Restic GCS Backup"

gcloud storage buckets add-iam-policy-binding gs://mc-backup-gcs-ahmed \
  --member="serviceAccount:restic-gcs-backup@mcc-training-project.iam.gserviceaccount.com" \
  --role="roles/storage.objectAdmin"

gcloud iam service-accounts keys create ~/restic-gcs-key.json   --iam-account=restic-gcs-
backup@mcc-training-project.iam.gserviceaccount.com
```

Phase 5 — Configure Environment Variables

Restic reads credentials and the repository password from environment variables. This keeps secrets out of command history and shell scripts.

Create a Secrets File

Save the following as `~/.restic-env` (then chmod 600 `~/.restic-env`):

```
# ----- Restic master password -----
export RESTIC_PASSWORD='xxxxxxxxxxxxxxx'

# ----- AWS S3 -----
export AWS_ACCESS_KEY_ID='AKIASRLCQUK6SN6DKIGX'
export AWS_SECRET_ACCESS_KEY='u/aNHH5JVC3xxxxxxxxxxxxxxxxxxxxxxxx'

# ----- Azure Blob -----
export AZURE_ACCOUNT_NAME='mcbackupazureahmed'
export AZURE_ACCOUNT_KEY='FFNfDfXxGSw6i4xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxSv0hOQ=='

# ----- Google Cloud -----
export GOOGLE_PROJECT_ID='mcc-training-project'
export GOOGLE_APPLICATION_CREDENTIALS="$HOME/restic-gcs-key.json"
```

Load It Into Your Shell

```
source ~/.restic-env
```

Phase 6 — Initialize the Three Repositories

Each cloud gets its own independent Restic repository. They share the same password but contain separate encrypted data.

Initialize AWS S3 Repository

```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed init
```

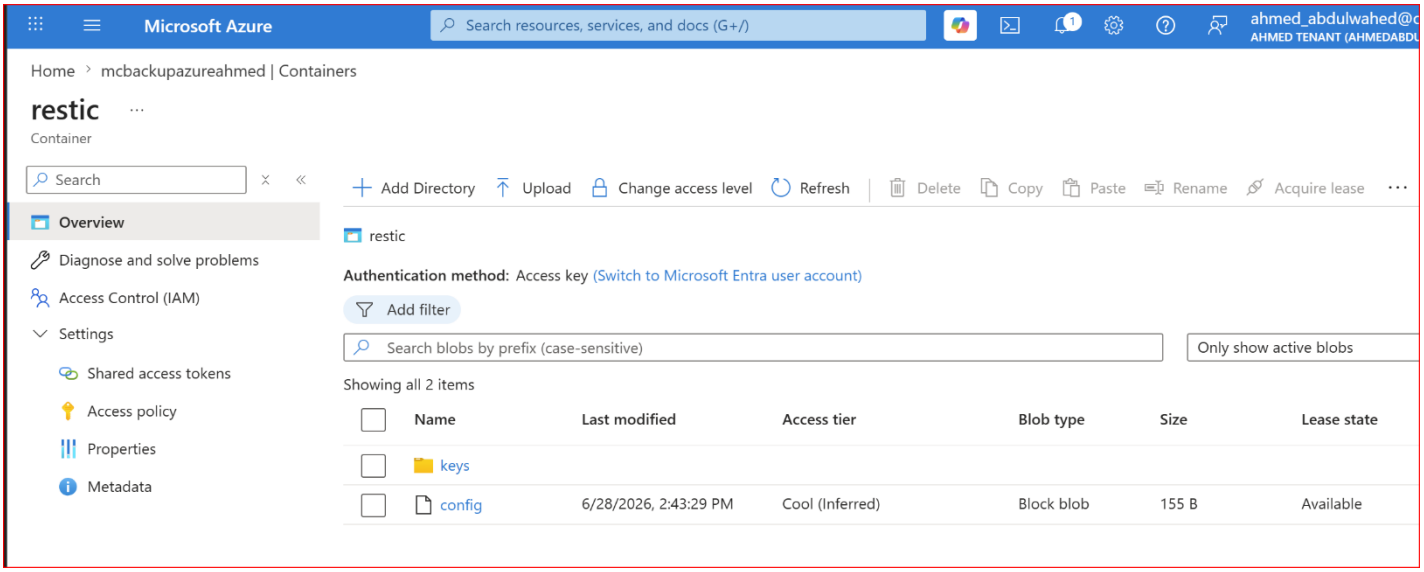
Initialize Azure Blob Repository

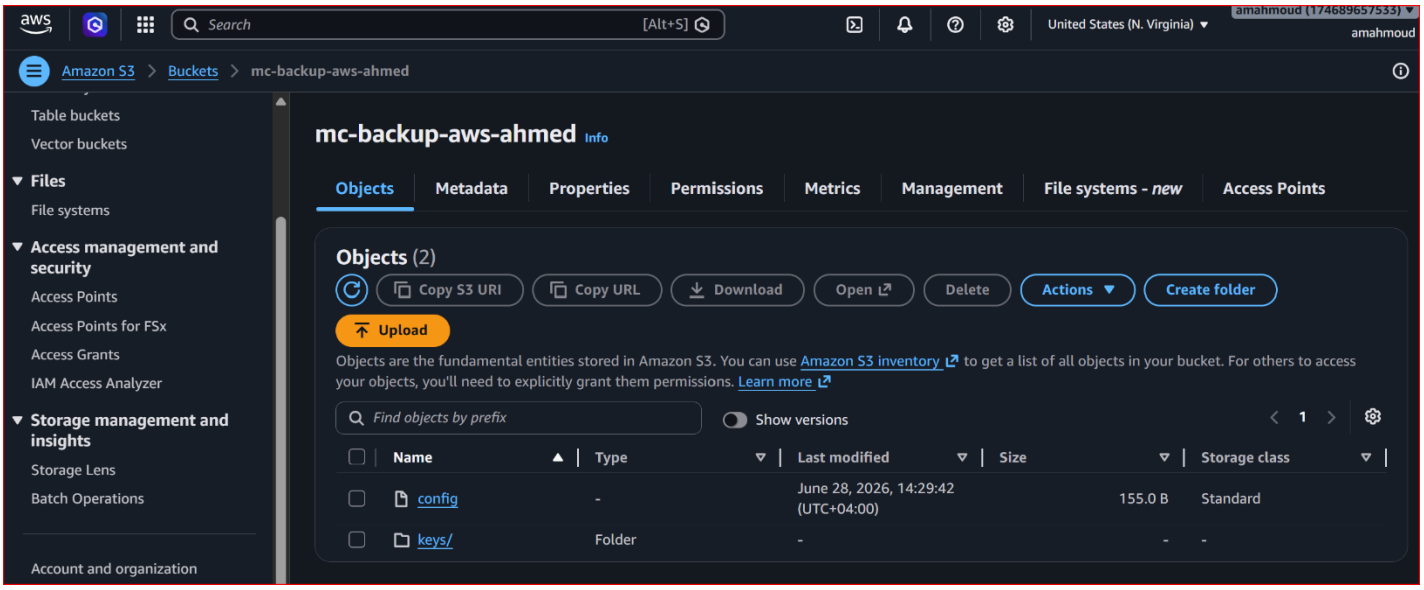
```
restic -r azure:restic:/ init
```

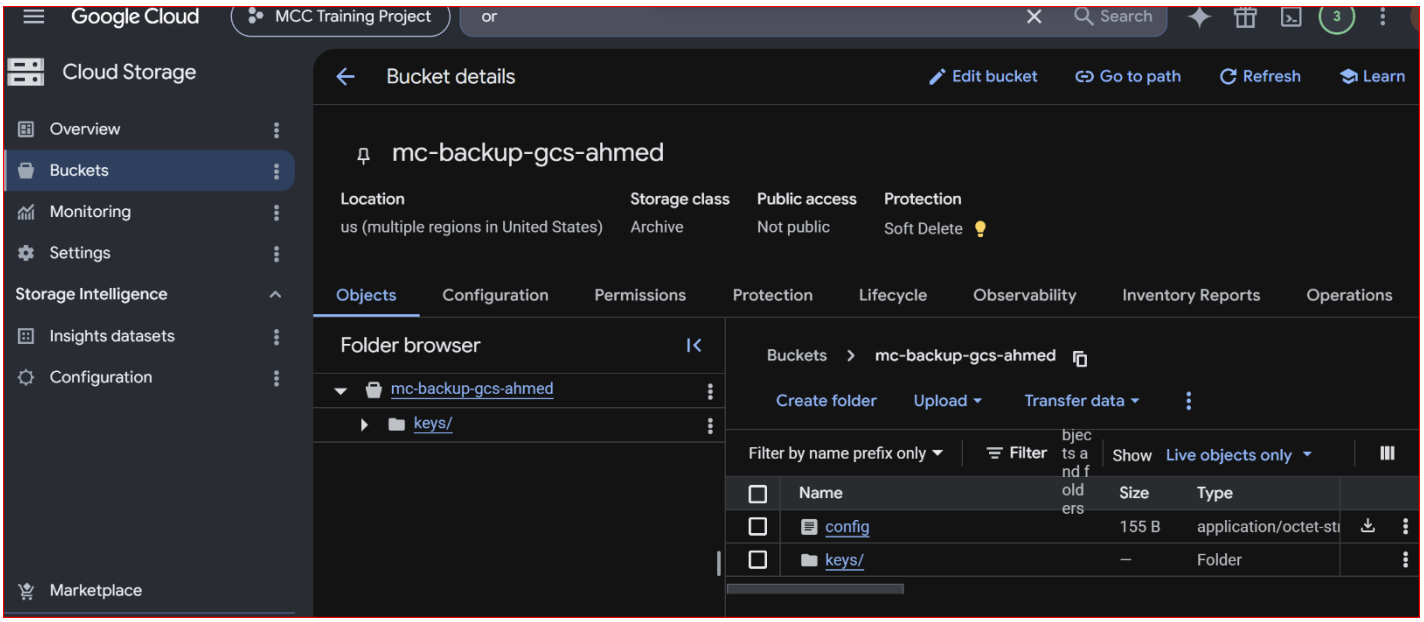
Restic reads AZURE_ACCOUNT_NAME and AZURE_ACCOUNT_KEY from the environment. The path after azure: is the container name.

Initialize GCS Repository

```
restic -r gs:mc-backup-gcs-ahmed:/ init
```


Microsoft Azure
Search resources, services, and docs (G+/)
Home > mcbackupazureahmed | Containers
restic
Container
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Access policy
Properties
Metadata
Add Directory
Upload
Change access level
Refresh
Delete
Copy
Paste
Rename
Acquire lease
restic
Authentication method: Access key (Switch to Microsoft Entra user account)
Add filter
Search blobs by prefix (case-sensitive)
Only show active blobs
Showing all 2 items
Name
Last modified
Access tier
Blob type
Size
Lease state
keys
config
6/28/2026, 2:43:29 PM
Cool (Inferred)
Block blob
155 B
Available


Search
[Alt+S]
United States (N. Virginia)
amahmoud
Amazon S3 > Buckets > mc-backup-aws-ahmed
Table buckets
Vector buckets
Files
File systems
Access management and security
Access Points
Access Points for FSx
Access Grants
IAM Access Analyzer
Storage management and insights
Storage Lens
Batch Operations
Account and organization
mc-backup-aws-ahmed
Info
Objects
Metadata
Properties
Permissions
Metrics
Management
File systems - new
Access Points
Objects (2)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
config
-
June 28, 2026, 14:29:42 (UTC+04:00)
155.0 B
Standard
keys/
Folder
-


Google Cloud
MCC Training Project
Search
Cloud Storage
Overview
Buckets
Monitoring
Settings
Storage Intelligence
Insights datasets
Configuration
Marketplace
Bucket details
Edit bucket
Go to path
Refresh
Learn
mc-backup-gcs-ahmed
Location
us (multiple regions in United States)
Storage class
Archive
Public access
Not public
Protection
Soft Delete
Objects
Configuration
Permissions
Protection
Lifecycle
Observability
Inventory Reports
Operations
Folder browser
mc-backup-gcs-ahmed
keys/
Buckets > mc-backup-gcs-ahmed
Create folder
Upload
Transfer data
Filter by name prefix only
Filter
Show
Live objects only
Name
Size
Type
config
155 B
Folder
keys/

Phase 7 — Run the First Backup

Pick a small directory for the first run — the goal is to validate the pipeline before backing up large data.

Backup to AWS

```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed backup ~/Documents
```

Backup to Azure

```
restic -r azure:restic:/ backup ~/Documents
```

Backup to GCS

```
restic -r gs:mc-backup-gcs-ahmed:/ backup ~/Documents
```

Verify Each Repository

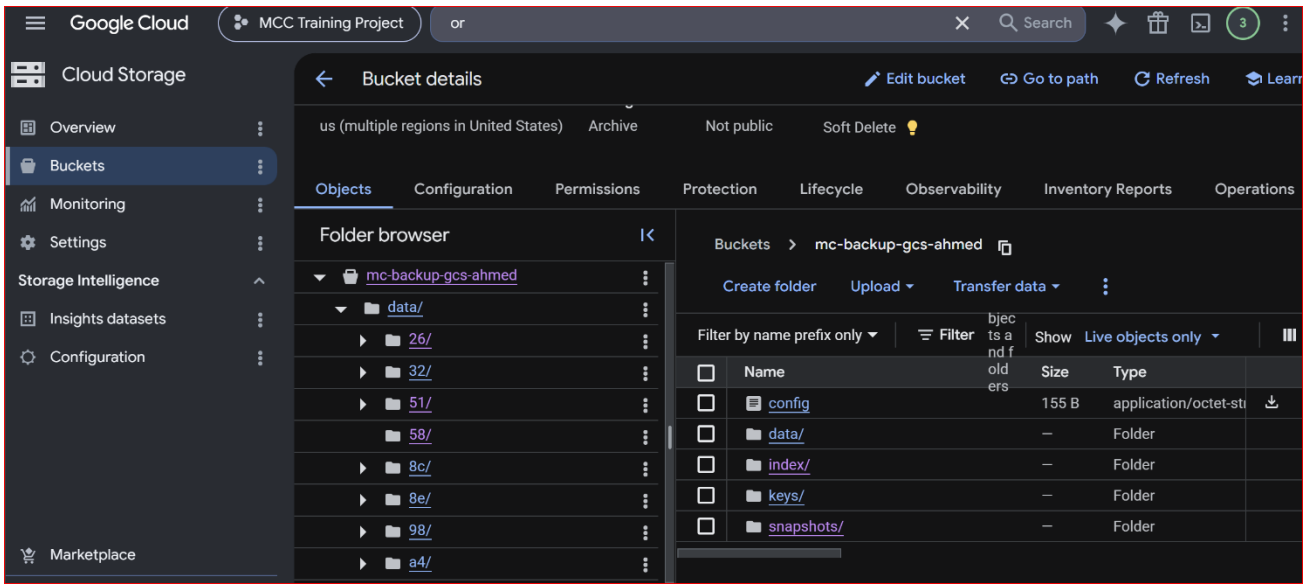
List snapshots in each cloud to confirm:

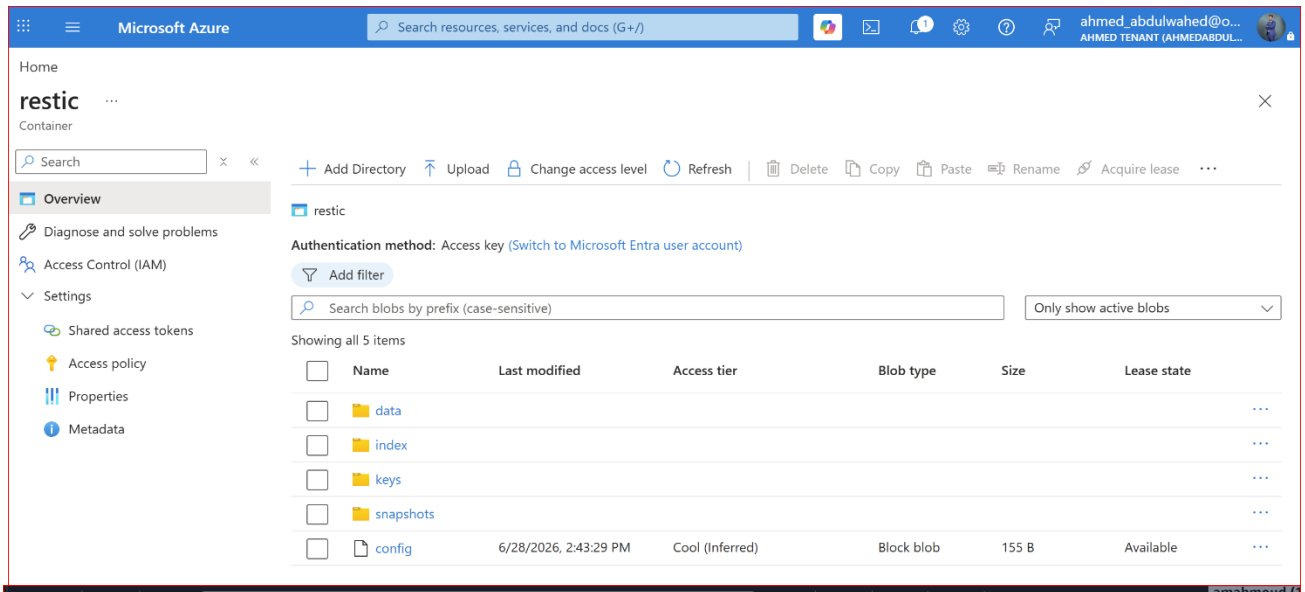
```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed snapshots
restic -r azure:restic:/                          snapshots
restic -r gs:mc-backup-gcs-ahmed:/                snapshots
```

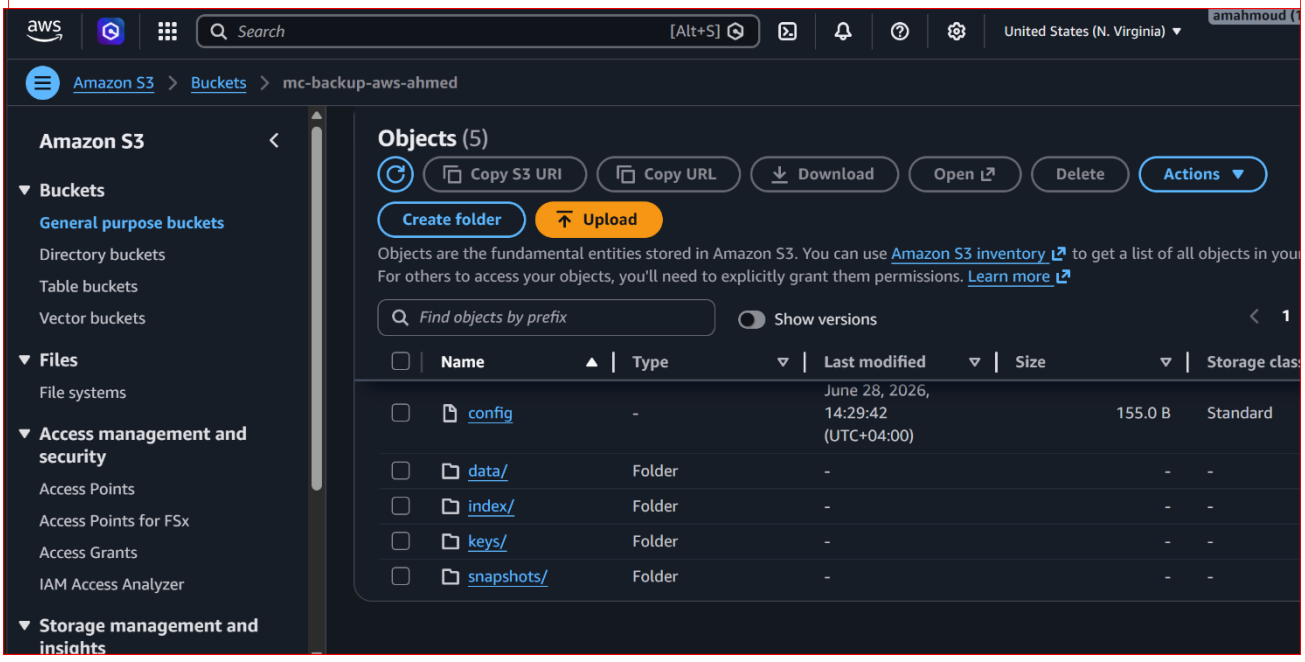
add test file (data1) and re-run the backup command again

```
[root@r9 restic]# ll ~/Documents/testdata/
total 102404
-rw-r--r--. 1 root root       21 Jun 28 14:56 data1
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_10.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_1.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_2.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_3.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_4.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_5.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_6.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_7.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_8.bin
-rw-r--r--. 1 root root 10485760 Jun 28 14:48 file_9.bin
[root@r9 restic]#
```

```
[root@r9 restic]# restic -r gs:mc-backup-gcs-ahmed:/ backup ~/Documents
repository 8116d5a7 opened (version 2, compression level auto)
using parent snapshot f62eebae
[0:00] 100.00%  1 / 1 index files loaded

Files:           1 new,     0 changed,    10 unmodified
Dirs:            0 new,     3 changed,     0 unmodified
Added to the repository: 11.023 KiB (4.537 KiB stored)

processed 11 files, 100.000 MiB in 0:05
snapshot 05e4c832 saved
[root@r9 restic]# restic -r azure:restic:/ backup ~/Documents
repository 74b6ec0f opened (version 2, compression level auto)
using parent snapshot cc375757
[0:00] 100.00%  1 / 1 index files loaded

Files:           1 new,     0 changed,    10 unmodified
Dirs:            0 new,     3 changed,     0 unmodified
Added to the repository: 11.023 KiB (4.524 KiB stored)

processed 11 files, 100.000 MiB in 0:02
snapshot c7dc7e93 saved
[root@r9 restic]# restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed backup ~/Documents
repository 479dc52d opened (version 2, compression level auto)
using parent snapshot bae88601
[0:00] 100.00%  1 / 1 index files loaded

Files:           1 new,     0 changed,    10 unmodified
Dirs:            0 new,     3 changed,     0 unmodified
Added to the repository: 10.893 KiB (4.461 KiB stored)

processed 11 files, 100.000 MiB in 0:07
snapshot b7b23baa saved
[root@r9 restic]#
```


Google Cloud
MCC Training Project
Search
Cloud Storage
Overview
Buckets
Monitoring
Settings
Storage Intelligence
Insights datasets
Configuration
Marketplace
Bucket details
Edit bucket
Go to path
Refresh
us (multiple regions in United States)
Archive
Not public
Soft Delete
Objects
Configuration
Permissions
Protection
Lifecycle
Observability
Inventory Reports
Operations
Folder browser
mc-backup-gcs-ahmed
data/
26/
32/
51/
58/
8c/
8e/
98/
a4/
Buckets
mc-backup-gcs-ahmed
Create folder
Upload
Transfer data
Filter by name prefix only
Filter
Show
Live objects only
Name
Size
Type
config
155 B
application/octet-stream
data/
index/
keys/
snapshots/
Folder


Microsoft Azure
Search resources, services, and docs (G+/)
Home
restic
Container
Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Shared access tokens
Access policy
Properties
Metadata
Add Directory
Upload
Change access level
Refresh
Delete
Copy
Paste
Rename
Acquire lease
Authentication method: Access key (Switch to Microsoft Entra user account)
Add filter
Search blobs by prefix (case-sensitive)
Only show active blobs
Showing all 5 items
Name
Last modified
Access tier
Blob type
Size
Lease state
data
index
keys
snapshots
config
6/28/2026, 2:43:29 PM
Cool (Inferred)
Block blob
155 B
Available


aws
Search
[Alt+S]
United States (N. Virginia)
Amazon S3
Buckets
mc-backup-aws-ahmed
Amazon S3
Buckets
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Files
File systems
Access management and security
Access Points
Access Points for FSx
Access Grants
IAM Access Analyzer
Storage management and insights
Objects (5)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your
For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Show versions
Name
Type
Last modified
Size
Storage class
config
June 28, 2026, 14:29:42 (UTC+04:00)
155.0 B
Standard
data/
index/
keys/
snapshots/
Folder

Phase 8 — Automate Daily Backups

Wrap the three backup commands in one script and schedule it daily.

The Backup Script

Save as `/usr/local/bin/multicloud-backup.sh`:

```
#!/usr/bin/env bash
set -euo pipefail

source /root/.restic-env

env | grep -E 'RESTIC|AWS|AZURE|GOOGLE' >> /root/restic-cron-env.log

SOURCE="/home/ahmed/Documents /etc /var/www"
TAG="daily-$(date +%Y%m%d)"

echo "[$(date)] Backup to AWS S3..."
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed backup --tag "$TAG" $SOURCE

echo "[$(date)] Backup to Azure Blob..."
restic -r azure:restic:/ backup --tag "$TAG" $SOURCE

echo "[$(date)] Backup to GCS..."
restic -r gs:mc-backup-gcs-ahmed:/ backup --tag "$TAG" $SOURCE

echo "[$(date)] All three clouds done."

sudo chmod 750 /usr/local/bin/multicloud-backup.sh
```

Schedule with cron

Edit root's crontab and run daily at 02:30:

```
sudo crontab -e
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
30 2 * * * /usr/local/bin/multicloud-backup.sh >> /var/log/multicloud-backup.log 2>&1
```

Or with systemd Timer (Preferred on Modern Linux)

Create `/etc/systemd/system/multicloud-backup.service`:

```
[Unit]
Description=Multi-Cloud Backup Run

[Service]
Type=oneshot
ExecStart=/usr/local/bin/multicloud-backup.sh
```

And `/etc/systemd/system/multicloud-backup.timer`:

```
[Unit]
Description=Run Multi-Cloud Backup Daily

[Timer]
OnCalendar=*-*-* 02:30:00
Persistent=true

[Install]
WantedBy=timers.target
```

```
sudo systemctl daemon-reload
sudo systemctl enable --now multicloud-backup.timer
sudo systemctl list-timers | grep multicloud
```

Phase 9 — Retention & Pruning

Without retention, snapshots accumulate forever and storage cost grows. Restic's forget and prune commands enforce a sensible policy.

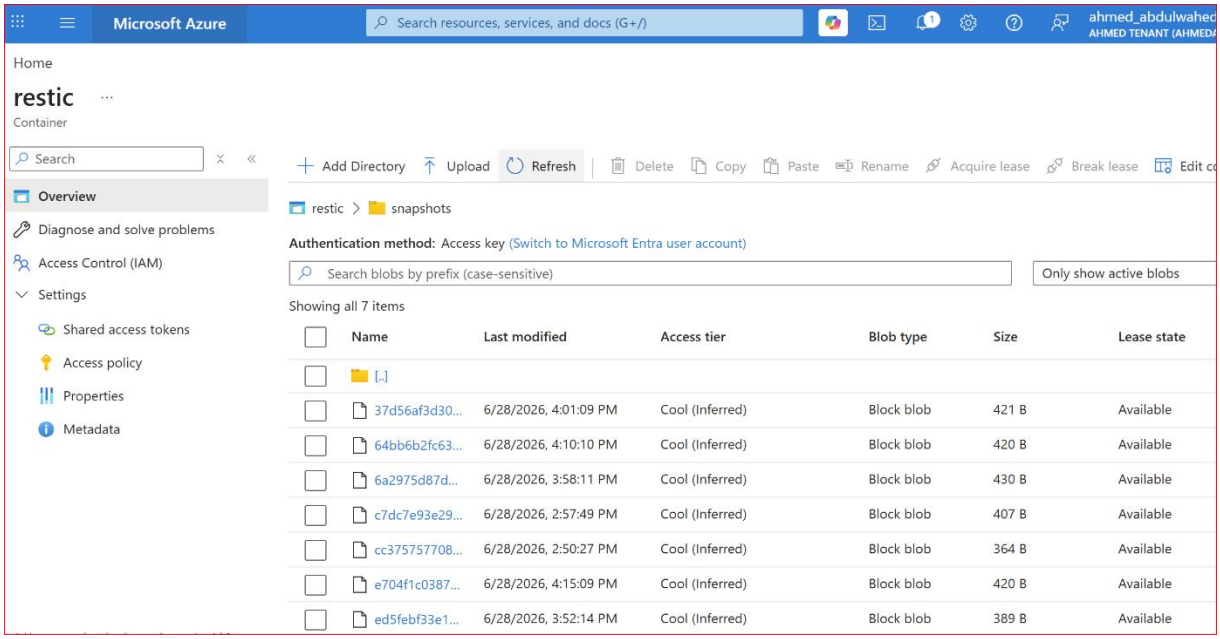
Recommended Retention Policy

Period	Keep
Last	7 snapshots
Daily	Last 7 days
Weekly	Last 4 weeks
Monthly	Last 12 months
Yearly	Last 3 years

Apply It

Add this block to the bottom of `multicloud-backup.sh`:

```
POLICY="--keep-last 7 --keep-daily 7 --keep-weekly 4 --keep-monthly 12 --keep-yearly 3"

for REPO in \
  s3:s3.amazonaws.com/mc-backup-aws-ahmed \
  azure:restic:/ \
  gs:mc-backup-gcs-ahmed:/ ; do
    echo "[$(date)] Forgetting old snapshots in $REPO"
    restic -r "$REPO" forget $POLICY --prune
done
```



Phase 10 — Verify Integrity

Backups you never test are not backups. Restic gives you cryptographic verification — use it.

Metadata Check (Cheap, Frequent)

Verifies the structure of the repository without downloading data. Safe to run weekly:

```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed check
restic -r azure:restic:/                          check
restic -r gs:mc-backup-gcs-ahmed:/                check
```

Deep Check (Costs Egress)

Downloads and verifies a random subset of data blocks. Run monthly or quarterly:

```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed check --read-data-subset=10%
```

```
[root@r9 restic]# restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed check
using temporary cache in /tmp/restic-check-cache-1845167897
create exclusive lock for repository
repository 479dc52d opened (version 2, compression level auto)
created new cache in /tmp/restic-check-cache-1845167897
load indexes
[0:01] 100.00%  2 / 2 index files loaded
check all packs
check snapshots, trees and blobs
[0:00] 100.00%  2 / 2 snapshots
no errors were found
[root@r9 restic]# restic -r azure:restic:/                          check
using temporary cache in /tmp/restic-check-cache-1210639683
create exclusive lock for repository
repository 74b6ec0f opened (version 2, compression level auto)
created new cache in /tmp/restic-check-cache-1210639683
load indexes
[0:00] 100.00%  2 / 2 index files loaded
check all packs
check snapshots, trees and blobs
[0:00] 100.00%  2 / 2 snapshots
no errors were found
[root@r9 restic]# restic -r gs:mc-backup-gcs-ahmed:/                check
using temporary cache in /tmp/restic-check-cache-1053074294
create exclusive lock for repository
repository 8116d5a7 opened (version 2, compression level auto)
created new cache in /tmp/restic-check-cache-1053074294
load indexes
[0:00] 100.00%  2 / 2 index files loaded
check all packs
check snapshots, trees and blobs
[0:00] 100.00%  2 / 2 snapshots
no errors were found
```

Phase 11 — Restore Drill

Practice restores before you need one. A backup whose restore has never been tested is a hope, not a strategy.

List Snapshots

```
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed snapshots
```

Each snapshot has a short ID (e.g., a1b2c3d4). Use it to refer to a specific point in time.

Restore the Latest Snapshot

```
mkdir -p /tmp/restore-test
restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed \
  restore latest --target /tmp/restore-test
```

```
[root@r9 restic]# restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed snapshots
repository 479dc52d opened (version 2, compression level auto)
ID        Time                 Host        Tags            Paths                  Size
----------------------------------------------------------------------------------------
bae88601  2026-06-28 14:48:49  r9                          /root/Documents        100.000 MiB

b7b23baa  2026-06-28 14:58:01  r9                          /root/Documents        100.000 MiB

92d65d8f  2026-06-28 15:30:02  r9          daily-20260628  /etc                   23.615 MiB
                                                           /home/ahmed/Documents

73ccb68e  2026-06-28 15:37:01  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

41f5c736  2026-06-28 15:50:01  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

e82c1766  2026-06-28 15:51:27  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

ac17c5a5  2026-06-28 15:52:02  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

ee3ff83e  2026-06-28 15:58:01  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

8fa44121  2026-06-28 16:01:01  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

99e5dd9d  2026-06-28 16:10:02  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents

cd78dc73  2026-06-28 16:15:01  r9          daily-20260628  /etc                   123.615 MiB
                                                           /root/Documents
----------------------------------------------------------------------------------------
Timestamps shown in +04 timezone
11 snapshots
```

Delete some data and restore full data from latest snapshot on AWS

```
[root@r9 restic]# rm ~/Documents/testdata/data1
rm: remove regular file '/root/Documents/testdata/data1'? y
[root@r9 restic]# rm ~/Documents/testdata/data1
rm: cannot remove '/root/Documents/testdata/data1': No such file or directory
[root@r9 restic]# mkdir -p /tmp/restore-test
[root@r9 restic]# restic -r s3:s3.amazonaws.com/mc-backup-aws-ahmed \
>   restore latest --target /tmp/restore-test
repository 479dc52d opened (version 2, compression level auto)
[0:00] 100.00%  6 / 6 index files loaded
restoring snapshot cd78dc73 of [/root/Documents /etc] at 2026-06-28 16:15:01.924988957 +0400 +04 by root@r9 to /tmp/restore-test
Summary: Restored 1882 files/dirs (123.615 MiB) in 0:24
[root@r9 restic]# cat ~/Documents/testdata/data1
cat: /root/Documents/testdata/data1: No such file or directory
[root@r9 restic]# cat /tmp/restore-test/data1
cat: /tmp/restore-test/data1: No such file or directory
[root@r9 restic]# cat /tmp/restore-test/
etc/  root/
[root@r9 restic]# cat /tmp/restore-test/root/Documents/testdata/data1
test data for backup
```