

de.KCD Summer School 2024

Deploying LLMs in the Cloud
Sep 16 – 20, 2024 Online

Foo Wei Ten (BIH@Charité)

19.09.2024

SPONSORED BY THE



Federal Ministry
of Education
and Research



Funded by
the European Union
NextGenerationEU



BIH Berlin Institute
of Health
@Charité

Day 4: Volumes, Shares, Backups, Monitoring

Agenda

- 2 slides of important messages
- Then hand-holding guided tour for Shares
- 1 real example of backup strategy + beginners hands-on
- Backup validation strategies
- 1 real example of backup/uptime monitoring strategy

Feeling Adventurous?

Feel free to look at **LLMCloud24 GitHub > Day_4 > 02-Backups > ADVANCED.md** for a challenge

Day 4: Backups

Do you *really* have a backup?

- separation
 - from primary data, from services, from system
 - also in terms of storage systems, geologically
- restorable
 - test frequently to ensure restorability
 - no restorability = no backup
- automation
 - less manual work = less human error = more consistent, timely backups
- monitoring
 - automation is useless without monitoring for failures or irregularities

Day 4: Backups

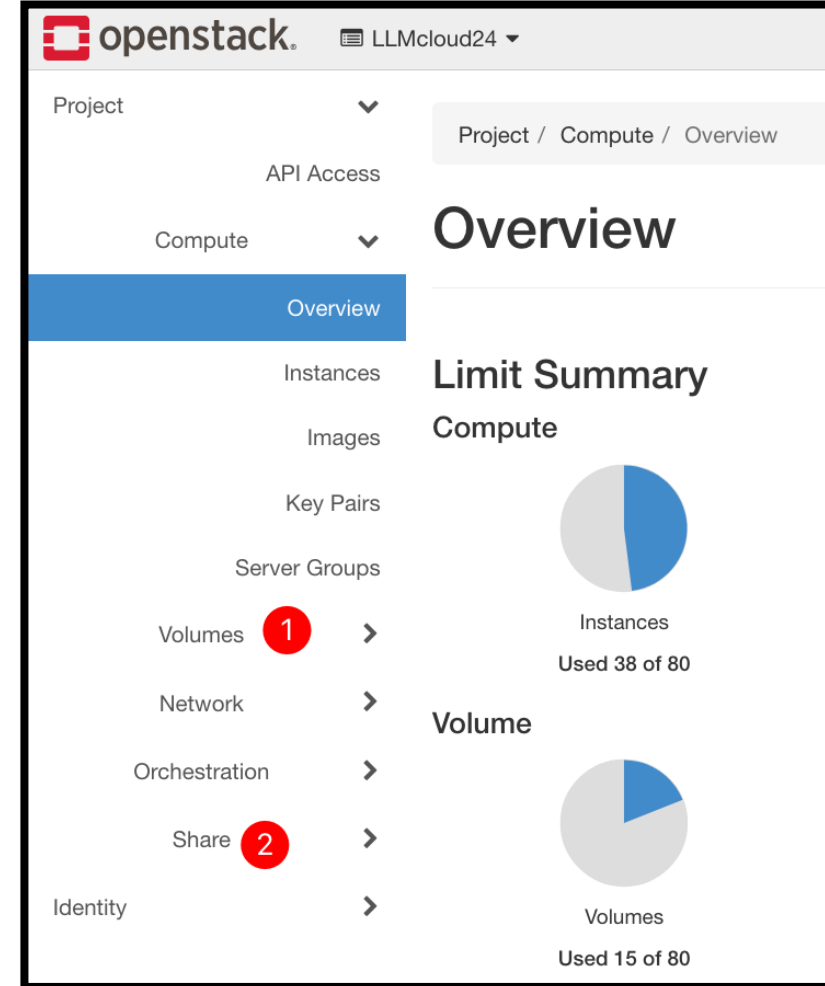
Repeat after me x3:

- ***snapshots are NOT backups***
 - “differential layers” that are tied to the original/primary, can easily get corrupted or deleted
- ***syncs are NOT backups***
 - does not protect against accidental deletion or corruption
- ***RAIDs are NOT backups***
 - offers redundancy against hardware failure, but not human errors or malware

Day 4: Volumes and Shares

Adding extra volume to your VMs

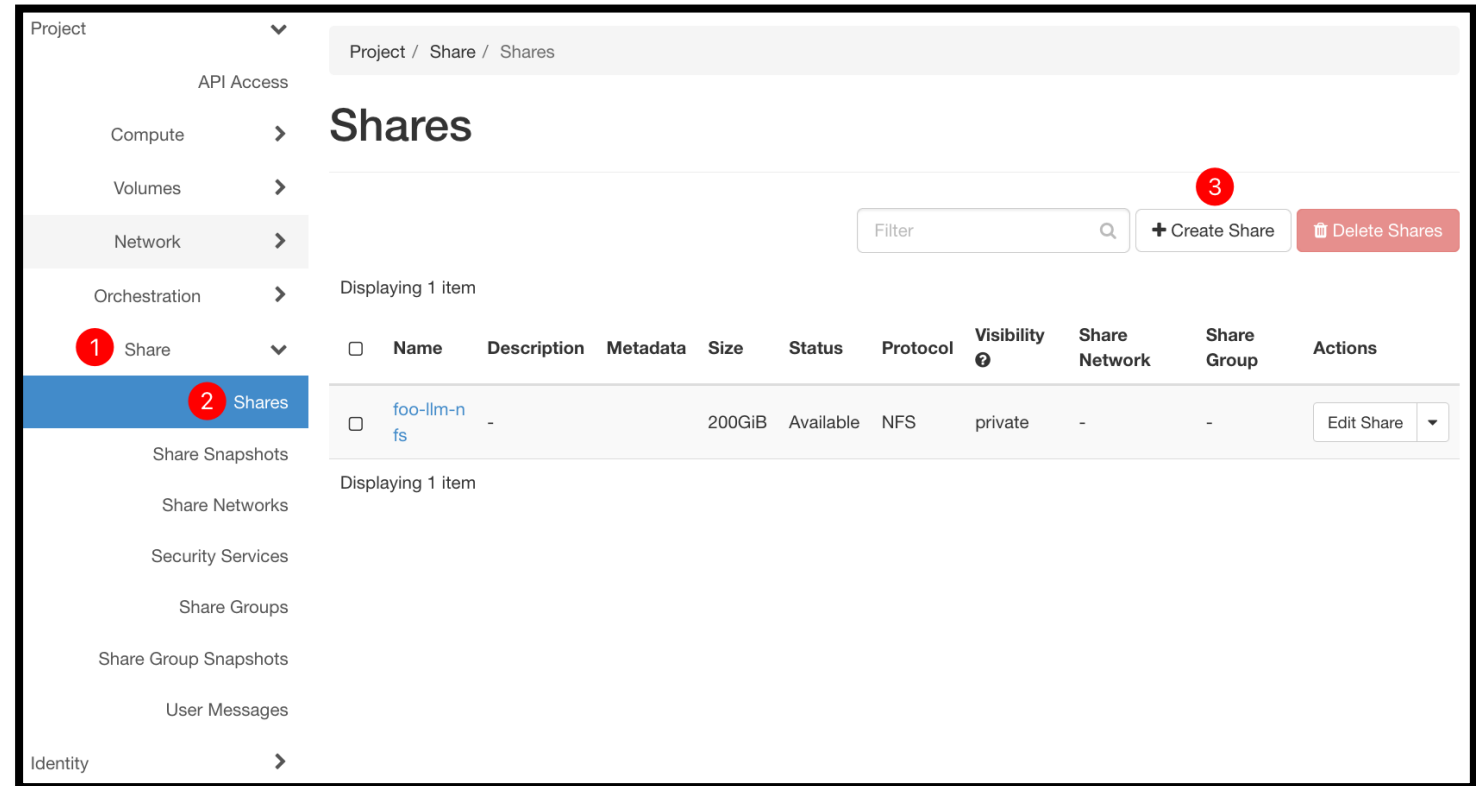
- 1) Volumes
- 2) Shares (networked, shareable)



Day 4: Volumes and Shares

Adding NFS Share to your VMs

- 1) Click **“Share”**
- 2) then **“Shares”**
- 3) then the **“Create Share”** button



The screenshot shows the OpenStack dashboard interface for managing shares. The left sidebar contains a navigation menu with the following items: Project, API Access, Compute, Volumes, Network, Orchestration, Share (1), Shares (2), Share Snapshots, Share Networks, Security Services, Share Groups, Share Group Snapshots, User Messages, and Identity. The 'Share' item is highlighted with a red circle and the number 1, and the 'Shares' item is highlighted with a red circle and the number 2. The main content area is titled 'Shares' and shows a table with one share. The table has columns: Name, Description, Metadata, Size, Status, Protocol, Visibility, Share Network, Share Group, and Actions. The share listed is 'foo-llm-nfs' with a size of 200GiB, status 'Available', protocol 'NFS', and visibility 'private'. A red circle with the number 3 is placed over the '+ Create Share' button in the top right corner of the main area.

Name	Description	Metadata	Size	Status	Protocol	Visibility	Share Network	Share Group	Actions
foo-llm-nfs	-		200GiB	Available	NFS	private	-	-	Edit Share

Day 4: Volumes and Shares

NFS Share Settings

- 1) Share name: **jd**oe-nfs-backup if your name is "John **D**oe"
 - 2) Use a good description that would help you recognize them
 - 3) Choose **NFS** for **Share Protocol**
 - 4) Let's go with **10** GiB Size
 - 5) Make sure to choose "**isilon-denbi**" for **Share Type**
- The rest can be left empty

Create Share

Share Name * 1

ften-nfs-backup

Description

NFS share for FTEN's backup 2

Share Protocol * 3

NFS

Size (GiB) * 4

10

Share Type * 5

isilon-denbi

Description:

Select parameters of share you want to create.

Metadata:

One line - one action. Empty strings will be ignored.
To add metadata use:

key=value

Share Limits

Total Gibibytes

200 of 1,000 GiB Used

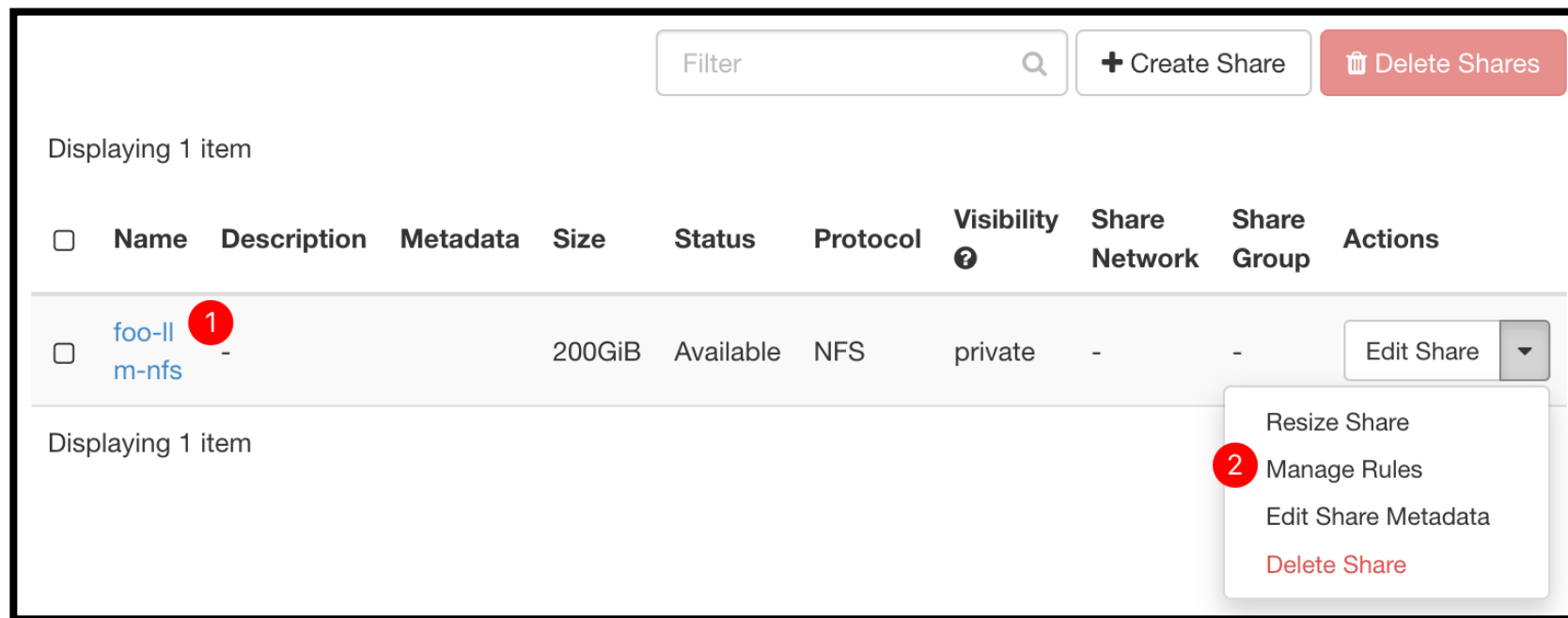
Number of Shares

1 of 50 Used

Day 4: Volumes and Shares

Adding your VM to your Share

- 1) Find your newly-created Share's name from the list
- 2) Click the arrow beside the "Edit Share" button and choose "Manage Rules"



Filter

Displaying 1 item

<input type="checkbox"/>	Name	Description	Metadata	Size	Status	Protocol	Visibility	Share Network	Share Group	Actions
<input type="checkbox"/>	foo-ll m-nfs	-		200GiB	Available	NFS	private	-	-	Edit Share

Displaying 1 item

- Resize Share
- Manage Rules
- Edit Share Metadata
- Delete Share

Day 4: Volumes and Shares

Adding your VM to your Share

- 1) Confirm you're operating on the correct Share
- 2) Click the “Add rule” button

Share Rules: foo-llm-nfs 1

2 + Add rule Delete Rules

Day 4: Volumes and Shares

Adding your VM to your Share

- 1) Choose “IP” for **Access Type**
- 2) Choose “read-write” for **Access Level**
- 3) Input your VM’s IP in **Access To**
- 4) Click “**Add**” button

Add Rule

Access Type *

ip 1

Access Level *

read-write 2

Access To *

10.0.2.96 3

Metadata

Day 4: Volumes and Shares

Accessing your Share from your VM

- Click your newly-created Share's name
 - You will now see a Share Overview page
- 1) Make sure that the **Share type** is “isilon-denbi”
 - 2) Your VM's IP should be listed in the **Access Rule**
 - 3) Copy this **Export locations Path**
 - 4) Mount it on your VM:
 - 1) `$ sudo apt install nfs-common`
 - 2) `$ sudo mkdir /nfs`
 - 3) `$ sudo chown ubuntu:ubuntu /nfs`
 - 4) `$ sudo mount <manila-prod.isi...> /nfs`
 - 5) add <manila-prod...> to /etc/fstab

```
> manila-prod..... /nfs nfs defaults,rw,nofail 0 0
```

Share Overview

Name	foo-llm-nfs
ID	fd5611e6-c0e4-4e65-9d79-33a592768ab7
Status	Available
Export locations	Path: manila-prod.isi.denbi.bihealth.org:/ifs/denbi/prod/share-61f0b850-f445-497 Preferred: False
	Path: manila-stage.isi.denbi.bihealth.org:/ifs/denbi/prod/share-61f0b850-f445-497 Preferred: False
Visibility	private
Availability zone	nova
Size	200 GiB
Protocol	NFS
Share type	Name: isilon-denbi ID: eaf0ddb1-995b-44d4-9d09-d9d4e8d5caef
Mount snapshot support	False
Created	Aug. 13, 2024, 9:16 p.m.
Task state	None

Access Rules

ip	Access to: 172.17.0.219 Access Level: rw Status: active Access Key: None Created At: Aug. 13, 2024, 9:30 p.m. Updated At: None
-----------	---

Day 4: Backups

Example of a backup strategy

- 1) identify what you want to backup (e.g., ~/myData)
- 2) create backups at the destination (e.g., /nfs)
 - `$ tar -czvf /nfs/myData.tar.gz ~/myData`
 - `$ rsync -aP ~/myData /nfs`
- 3) add date time information, logs
- 4) bonus: checksum, encryption
- 5) other tools like restic, rclone,
- 6) roll everything into a script and automate it

Day 4: Backups

Example of a backup strategy

- 1) identify what you want to backup (e.g., ~/myData)
- 2) create backups at the destination (e.g., /nfs)
 - `$ tar -czvf /nfs/myData.tar.gz ~/myData`
 - `$ rsync -aP ~/myData /nfs`
- 3) add date time information, logs
- 4) bonus: checksum, encryption
- 5) other tools like restic, rclone,
- 6) roll everything into a script and automate it

```
#!/bin/bash

SRC_FOLDER="/var/opt/gitlab/backups"
DST_FOLDER="/nfs2/tar"

# Log file path
LOG_FILE="/nfs2/log/tar_sync.log"

# Run rsync to copy the folder
/usr/bin/rsync -a "$SRC_FOLDER/" "$DST_FOLDER/"

# Check if the sync was successful
if [ $? -eq 0 ]; then
    echo "$(date '+%Y-%m-%d %H:%M:%S') Sync completed successfully" >> $LOG_FILE
else
    echo "$(date '+%Y-%m-%d %H:%M:%S') Sync failed: $?" >> $LOG_FILE
fi

# Remove all but the last 7 tar files from the source folder
find "$SRC_FOLDER/" -maxdepth 1 -name "*_gitlab_backup.tar" | sort | head -n -7 | xargs rm -f
```

```
# m h dom mon dow  command
50 20 * * * /opt/gitlab/bin/gitlab-backup create CRON=1 >> /nfs2/log/gitlab.log 2>&1
0 21 * * * /bin/bash -c '/etc/gitlab/tar_sync.sh >> /nfs2/log/tar_cron.log 2>&1 || echo "Tar cron failed at $(date)" | sendmail -v foo-wei.ten@bih-charite.de'
10 21 * * * /bin/bash -c '/etc/gitlab/rsync.sh >> /nfs2/log/rsync_cron.log 2>&1 || echo "Rsync failed at $(date)" | sendmail -v foo-wei.ten@bih-charite.de'
```

Day 4: Validation and Restorability

Examples of validation strategies

- 1) compare metadata, checksums
 - sizes, number of files, dates
 - re-calculate and compare checksums
- 2) encrypted files can be decrypted successfully
- 3) actually restoring the backups (can be very use-case specific)
 - data: .RData/.RDS can be loaded properly
 - databases: dumps can be restored, data intact
 - workflows: can be re-run and get expected outputs given inputs
 - services: can be redeployed without issues

Day 4: Validation and Restorability

Restorability validation for example service: GitLab

- 1) Production-VM running GitLab
 - generates backup tar every day
 - also need to account for the gitlab config and other files, like certificates etc
- 2) Validation-VM running GitLab
 - purpose: confirm restoration of data from the backups works
 - copy the backup tar to a specific location
 - make sure that the version of GitLab running on Validation-VM matches the version used in Production-VM to generate the backup
 - patch the relevant differences between Production and Validation (e.g., IP)
 - apply the backup restore
 - apply other post-restore actions and checks
 - report the status

Day 4: Monitoring

Example of a monitoring strategy

- 1) Script to check file and folder information
- 2) API endpoint to trigger the script
- 3) CI/CD of status page with notifications

Day 4: Monitoring

Example of a monitoring strategy

- 1) Script to check file and folder information
- 2) API endpoint to trigger the script
- 3) CI/CD of status page with notifications



GitHub Actions



```
image: alpine:latest

pages:
  stage: deploy
  script:
    - apk add --no-cache curl jq

    - for i in 1 2 3 4 5 6 7; do
      mkdir -p public/$i;
      echo "<!DOCTYPE html>"
      <html lang="en">
      <head><meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Status $i</title></head>
      <body><pre>" > public/$i/index.html;
      curl "10.0.2.10:8778/status/$i" | jq . | sed 's/&/\&amp;/g; s/</\&lt;/g; s/>/\&gt;/g' >> public/$i/index.html;
      echo "</pre></body></html>" >> public/$i/index.html;
      done

    - echo "<html><body>" > public/index.html
    - for i in 1 2 3 4 5 6 7; do
      echo "<h2>Status $i</h2>" >> public/index.html;
      echo "<pre>" >> public/index.html;
      curl "10.0.2.10:8778/status/$i" | jq . | sed 's/&/\&amp;/g; s/</\&lt;/g; s/>/\&gt;/g' >> public/index.html;
      echo "</pre><hr>" >> public/index.html;
      done
    - echo "</body></html>" >> public/index.html
```

Status 1

```
{
  "file_name": "1726174259_2024_09_12_17.2.1_gitlab_backup.tar",
  "file_path": "/backups/1726174259_2024_09_12_17.2.1_gitlab_backup.tar",
  "size": "8.356 GB",
  "creation_time": "Thu Sep 12 21:02:59 2024",
  "modification_time": "Thu Sep 12 20:58:23 2024",
  "api_time": "Fri Sep 13 16:03:33 2024",
  "is_up_to_date": true,
  "check_delta": "19.09 hours"
}
```

Status 2

```
{
  "file_name": "gitlab-conf-1726178401.tar.gz",
  "file_path": "/aspera/gitlab-conf-1726178401.tar.gz",
  "size": "0.001 GB",
  "creation_time": "Thu Sep 12 22:00:26 2024",
  "modification_time": "Thu Sep 12 22:00:01 2024",
  "api_time": "Fri Sep 13 16:03:33 2024",
  "is_up_to_date": true,
  "check_delta": "18.06 hours"
}
```

Status 3

```
{
  "file_name": "1726174259_2024_09_12_17.2.1_gitlab_backup.tar",
  "file_path": "/aspera/1726174259_2024_09_12_17.2.1_gitlab_backup.tar",
  "size": "8.356 GB",
  "creation_time": "Thu Sep 12 22:00:26 2024",
  "modification_time": "Thu Sep 12 22:00:01 2024",
  "api_time": "Fri Sep 13 16:03:33 2024",
  "is_up_to_date": true,
  "check_delta": "18.05 hours"
}
```

Status 4

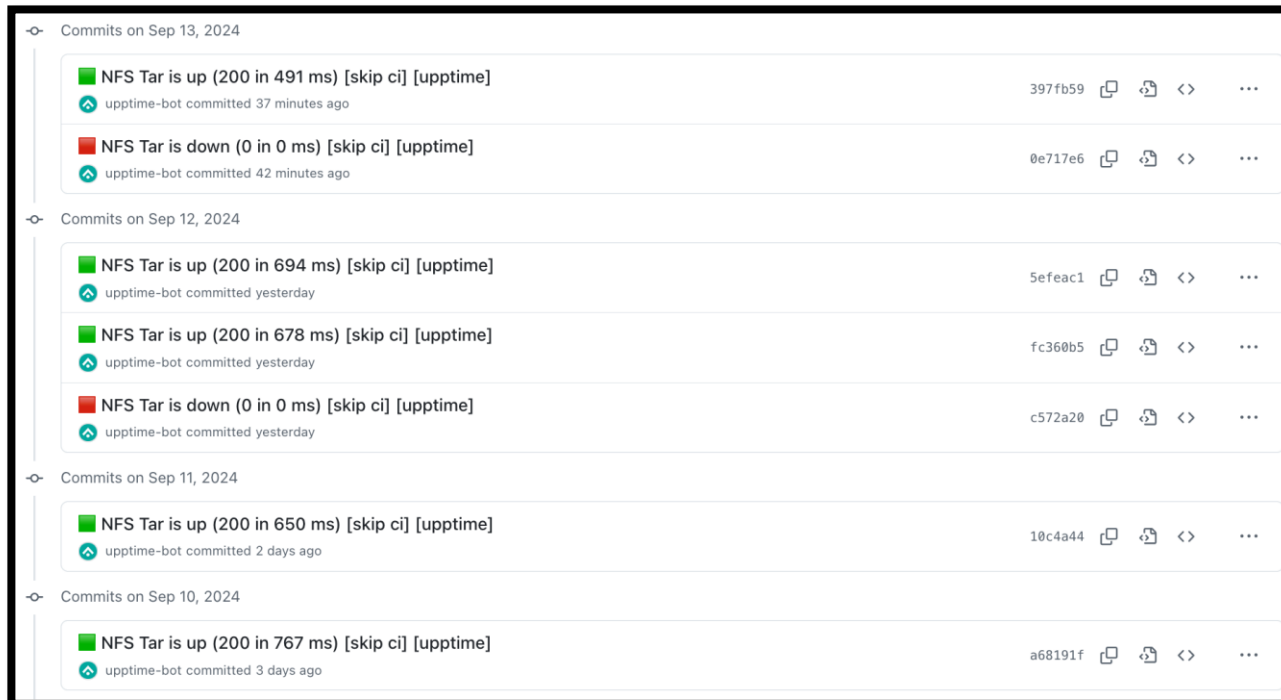
```
{
  "disk_usage": {
    "total_space_gb": "500.000",
    "free_space_gb": "143.289",
    "used_space_gb": "356.711",
    "used_percentage": "71.34%",
    "is_disk_space_low": false
  },
  "file_path": "/backups",
  "api_time": "Fri Sep 13 16:03:33 2024",
  "check_delta": "19.10 hours",
  "is_up_to_date": true
}
```

Day 4: Monitoring



Example of a monitoring strategy

- 1) Script to check file and folder information
- 2) API endpoint to trigger the script
- 3) CI/CD of status page with notifications



Uptime is the open-source uptime monitor and status page, powered entirely by GitHub.

This is a sample status page which uses **real-time** data from our [GitHub repository](#). No server required — just GitHub Actions, Issues, and Pages. [Get your own for free](#)

✓ All systems are operational

Live Status

24h 7d 30d 1y all

HX GitLab

Overall uptime: 100.00%
Average response time: 1195 ms

NFS Tar

Overall uptime: 99.93%
Average response time: 814 ms

Aspera Conf

Overall uptime: 100.00%
Average response time: 124 ms

Aspera Tar

Overall uptime: 100.00%
Average response time: 124 ms

NFS Disk

Overall uptime: 100.00%
Average response time: 123 ms

Local Disk

Overall uptime: 100.00%
Average response time: 124 ms

Thank you!

We're looking forward to working with you this week.

In case of questions please contact:
foo-wei.ten@bih-charite.de

SPONSORED BY THE



Federal Ministry
of Education
and Research



Funded by
the European Union
NextGenerationEU



BIH Berlin Institute
of Health
@Charité