



HENRY MORRIS SERVER PROJECT

January 2022 – September 2022

Intro



As part of receiving the initial Henry Morris fund of £300 we promised to set out and create:



A Cloud Storage Server like Google Drive



A Game Panel

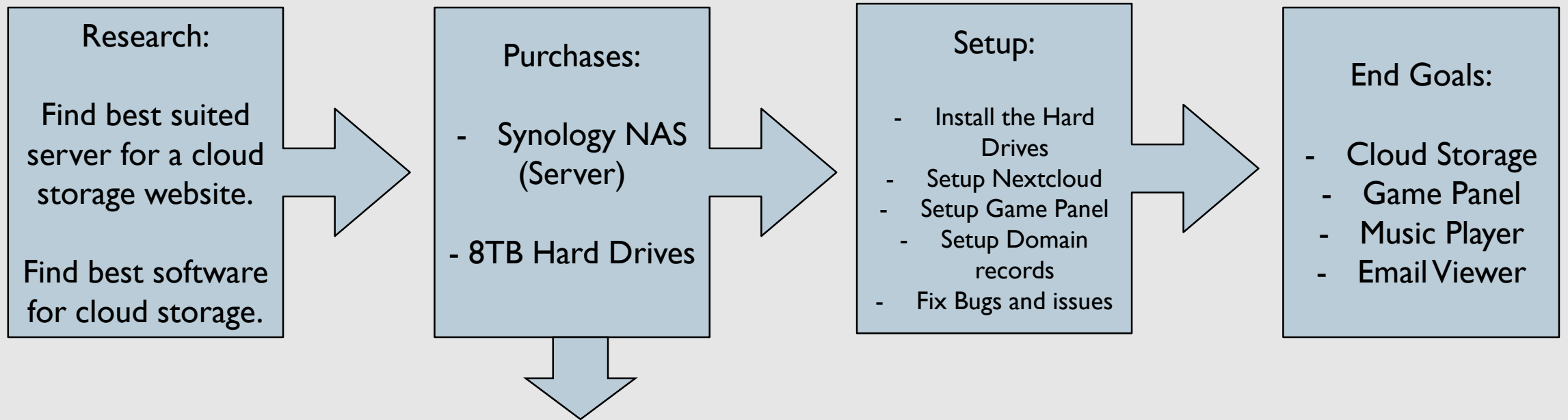


A Mail Server like Gmail and Hotmail



This Report outlines and establishes how we achieved these stated goals and the issues we encountered.

Overview / General Plan



<https://www.amazon.co.uk/Synology-DS220j-Bay-Desktop-Enclosure/dp/B084CLC39K> - £158

<https://www.amazon.co.uk/Seagate-IronWolf-Internal-Systems-Workload/dp/B07H289S79/261-4174582-0890729?content-id=amzn1.sym.79b812bf-5c8b-4c0c-851c-784423adaff5&psc=1> - £74

Research

- The hardest part of any tech related project is looking for the best hardware to support what we plan on doing, initially we planned on setting up a mail server as another separate physical server, but quickly found out that the port needed for that to work was blocked by my Internet Service Provider (ISP) (An ISP is like BT or Virgin Media) and they said it was against their security policy and refused to unblock it. Thus we decided to reallocate funds for the mail server towards just one server for both the cloud storage and game panel servers.
- Initially we decided upon a Dell T6310 as it had the necessary specifications (image for reference) But soon realised it would be too expensive and would remove funds needed for storage, so we decided on a Synology NAS and two 4TB of Seagate Iron wolf Hard Drives, which gave us room for redundancy as well.
- We also had to locate the best software for our now 2 chosen projects, Cloud Storage and the Game Panel: (This is covered in their respective sections) And we ultimately decided on Nextcloud and AMP.



Synology Setup

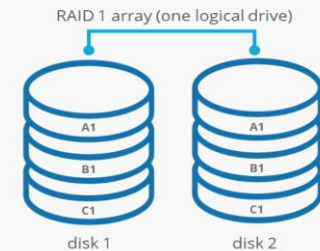
After we purchased and received the Synology and the two 4 Terra Byte, Hard Drives we decided on using RAID 1, a type of redundancy for our data as illustrated by the diagram to the right. It basically means that any data we put into one Hard Drive, that same data will be mirrored too the other Hard Drive, essentially duplicating the data in case of a drive failure.

This is an essential step in making sure that anything we upload to our servers will be protected, and because they will be running 24/7 this means I can guarantee that should a drive fail, I can recover the data and copy it to another drive, making sure that nothing is lost.

What I am doing here is what companies like Google and Microsoft do but on a much, much, smaller scale, and is how these companies can keep data uploaded to their servers secure and minimize data loss.

The second image is showing the two drives we purchased in a RAID 1 Array inside the Synology NAS.

RAID 1 (Redundant Array of Independent Disks Level 1)



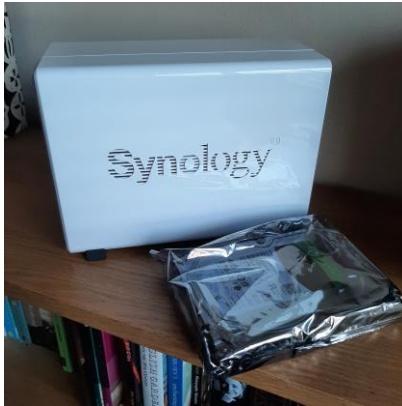


Figure 1



Figure 3



Figure 2

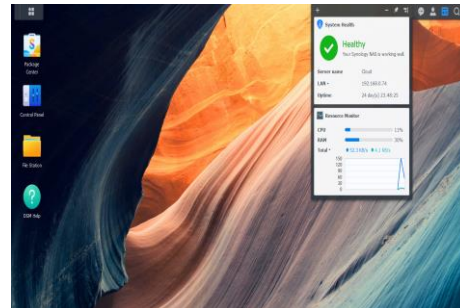


Figure 4

Synology Setup

- The Synology NAS and 4TB Hard Drive (Fig 1)
- When the Synology and the two 4TB Hard drives arrived we first set about opening the Synology and installing the Hard Drives in order to create a RAID 1 array (Fig 2)
- After Installing the Hard Drives and re-attaching the securing brackets, the lid was put on and screwed closed. (Fig 3) Also seen in Fig 3 is the ethernet port and power supply port, which I then plugged a CAT 7 ethernet cable into it, which can provide up to 10GB of data transfer, which combined with a recent Internet upgrade my family has gotten allows the Cloud Server and Game panel to have a higher upload and download speed.
- We then plugged the power in and ran through a start up menu and were presented with the main menu screen. (Fig 4)

DNS management for obcave.com

[REDACTED] = Censored IP Address

Search DNS Records

Search

Advanced

+ Add record

Type ▲	Name	Content	Proxy status	TTL	Actions
△ A	cloud	[REDACTED]	DNS only	Auto	Edit ▶
A	fallback	[REDACTED]	Proxied	Auto	Edit ▶
△ A	node	[REDACTED]	DNS only	Auto	Edit ▶
A	obcave.com	[REDACTED]	Proxied	Auto	Edit ▶
△ A	office	[REDACTED]	DNS only	Auto	Edit ▶
△ A	panel	[REDACTED]	DNS only	Auto	Edit ▶
A	www	[REDACTED]	Proxied	Auto	Edit ▶
CNAME	docs	hosting.gitbook.io	DNS only	12 hr	Edit ▶
MX	obcave.com	amir.mx.cloudflare.net	19 DNS only	Auto	Edit ▶
MX	obcave.com	linda.mx.cloudflare.net	13 DNS only	Auto	Edit ▶
MX	obcave.com	isaac.mx.cloudflare.net	15 DNS only	Auto	Edit ▶

Type ▲	Name	Content	Proxy status	TTL	Actions
△ A	cloud	[REDACTED]	DNS only	Auto	Edit ▼
Type	Name (required)	IPv4 address (required)	Proxy status	TTL	
A	<input type="text" value="cloud"/>	<input type="text" value="[REDACTED]"/>	<input type="checkbox"/> DNS only	<input type="text" value="Auto"/>	
Use @ for root					
Delete			Cancel Save		

DOMAIN RECORD SETUP

In order for my internal systems like the Cloud Storage and Game Panel to be accessible across the internet and outside of my local home network I first must port forward ports 80 and 443 (Port 80 corresponds to http, and Port 443 corresponds to https, aka secure transfer protocol)

And to further protect my home network I have the ports and IP Address proxied through Cloudflare (Images to the left) and is how I make subdomains.

For example the domain for my Cloud Server is: <https://cloud.obcave.com>
This is using https which means its secure when data is being transferred and people can't snoop on the signal.

- Nextcloud is the free open-source software implementation I will be using in order to setup the Cloud server, information on open-source software can be found on the next slide.

- As we are using ubuntu on the Synology Server we purchased we need to run a few commands in

```
sudo apt update && sudo apt upgrade  
sudo apt install apache2 mariadb-server libapache2-mod-php php-gd php-mysql \  
php-curl php-mbstring php-intl php-gmp php-bcmath php-xml php-imagick php-zip
```

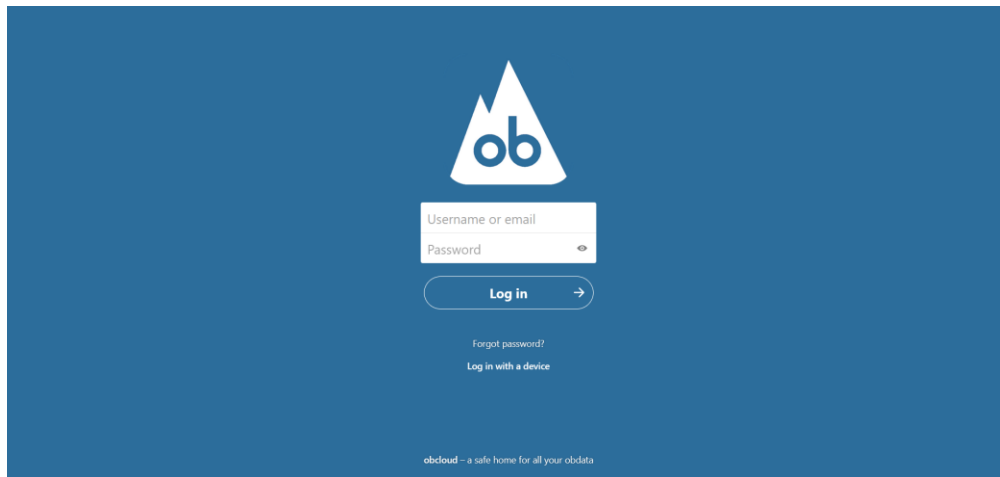
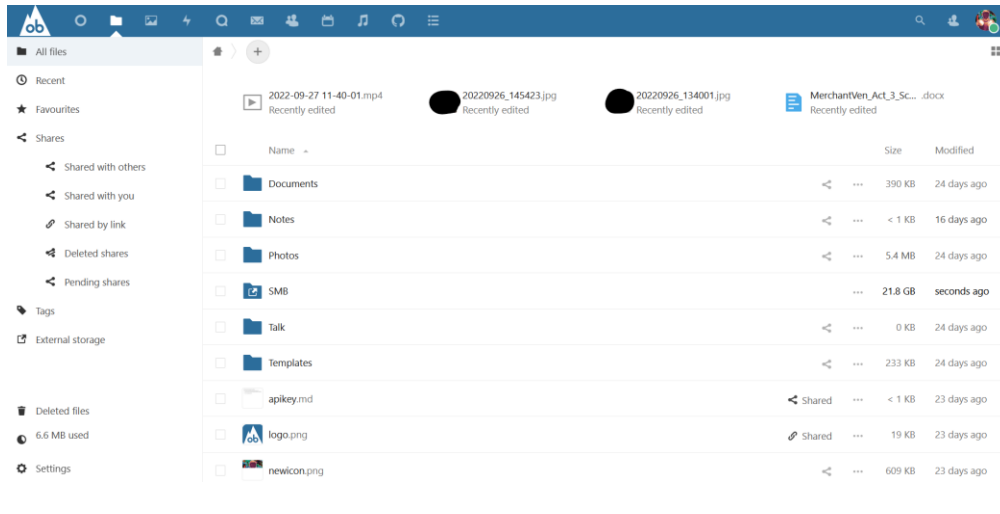
- Once that's run, we setup the MYSQL database in order to store the data, we set a password and

```
md5sum -c nextcloud-x.y.z.tar.bz2.md5 < nextcloud-x.y.z.tar.bz2  
sha256sum -c nextcloud-x.y.z.tar.bz2.sha256 < nextcloud-x.y.z.tar.bz2  
md5sum -c nextcloud-x.y.z.zip.md5 < nextcloud-x.y.z.zip  
sha256sum -c nextcloud-x.y.z.zip.sha256 < nextcloud-x.y.z.zip
```

- Then we proxy it through nginx. (Image to right)

Nextcloud Setup

```
server {  
    server_name cloud.obcave.com;  
    access_log /var/log/nginx/cloud.obcave.com.access_log;  
    error_log /var/log/nginx/cloud.obcave.com.error_log info;  
    location / {  
        proxy_pass https://localhost:444;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $remote_addr;  
        client_max_body_size 0;  
        add_header Strict-Transport-Security "max-age=15768000; includeSubDomains; preload;" always;  
        fastcgi_buffers 64 4K;  
        fastcgi_connect_timeout 60;  
        fastcgi_send_timeout 1800;  
        fastcgi_read_timeout 1800;  
        gzip on;  
        gzip_vary on;  
        gzip_comp_level 4;  
        gzip_min_length 256;  
        gzip_proxied expired no-cache no-store private no_last_modified no_etag auth;  
        gzip_types application/atom+xml application/javascript application/json application/ld+json application/manifest+json application/...  
        fastcgi_hide_header X-Powered-By;  
    }  
  
    listen 443 ssl; # managed by Certbot  
    ssl_certificate /etc/letsencrypt/live/cloud.obcave.com/fullchain.pem; # managed by Certbot  
    ssl_certificate_key /etc/letsencrypt/live/cloud.obcave.com/privkey.pem; # managed by Certbot  
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  
}  
  
server {  
    listen 80;  
    if ($host = cloud.obcave.com) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
}  
  
server_name cloud.obcave.com;  
    return 404; # managed by Certbot  
}
```

Nextcloud images

Here are some images of the finished product, logos have been designed by me.

A fully working cloud storage service with support for music streaming and communication.

Open Source Software

- Here is a definition on what open-source software is from Wikipedia (which is also open-source software):
- **Open-source software (OSS)** is computer software that is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software and its source code to anyone and for any purpose. Open-source software may be developed in a collaborative public manner. Open-source software is a prominent example of open collaboration, meaning any capable user can participate online in development, making the number of possible contributor's indefinite. The ability to examine the code facilitates public trust in the software.
- Open-source software development can bring in diverse perspectives beyond those of a single company. A 2008 report by the Standish Group stated that adoption of open-source software models has resulted in savings of about \$60 billion per year for consumers.
- Open-source code can be used for studying and allows capable end users to adapt software to their personal needs in a similar way user scripts and custom style sheets allow for web sites, and eventually publish the modification as a fork for users with similar preferences, and directly submit possible improvements as pull requests.

Game Panel Setup

- Setting up the game panel was rather simple with everything else setup, all we needed to do was purchase a license from the cube coders website and run their installer which helped us to quickly get it running. (Images to the right showing the basic user interface) (Figure 1)
- However, in order to make it accessible from the wider internet I needed to proxy the IP address through our reverse proxy manager nginx for our subdomain “panel” to connect to the IP address. Image attached of the nginx proxy rules. (Figure 2)

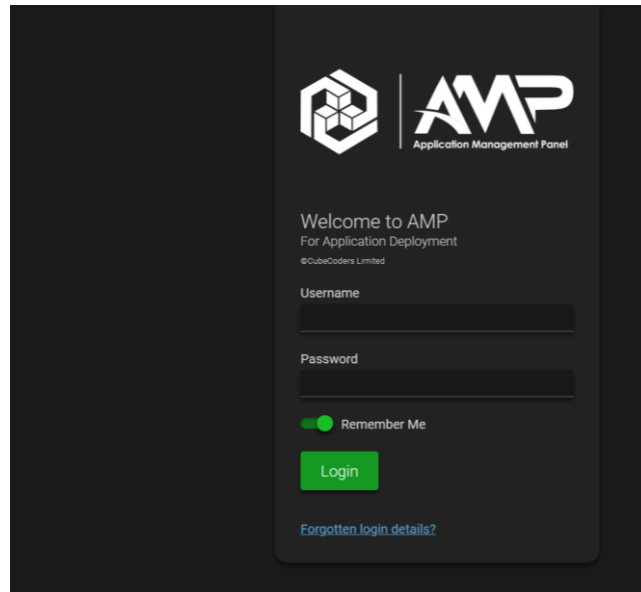
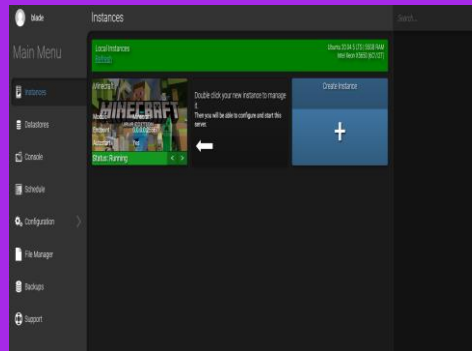


Figure 1



```
GNU nano 4.8 /etc/nginx/conf.d/panel.obcave.com.conf
#Generated by Cubecoders AMP for ADS
server {
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/panel.obcave.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/panel.obcave.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    server_name panel.obcave.com;

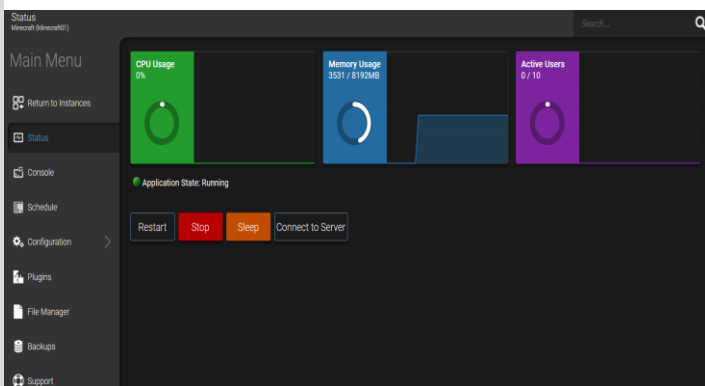
    client_max_body_size 100M;
    proxy_request_buffering off;
    proxy_buffering off;
    proxy_pass_request_body on;

    location / {
        proxy_pass http://localhost:8081;
        proxy_set_header    Host $host;
        proxy_set_header     X-Real-IP $remote_addr;
        proxy_set_header     X-Forwarded-For $remote_addr;
        proxy_set_header     Upgrade $http_upgrade;
        proxy_set_header     Connection $http_connection;
        proxy_set_header     X-AMP-Scheme $scheme;
        proxy_read_timeout   86400s;
        proxy_send_timeout   86400s;
        proxy_http_version   1.1;
        proxy_redirect        off;
        proxy_buffering       off;
        client_max_body_size  10240M;
        error_page 502 /NotRunning.html;

        location = /NotRunning.html {
            root /opt/cubecoders/amp/shared/WebRoot;
            internal;
        }

        location /shared/ {
            alias /opt/cubecoders/amp/shared/WebRoot/;
        }
    }
}
```

Figure 2



Bugs and Issues

During my time working on this project a few issues have crept up, this is a list of those issues and how I ultimately solved them.

I couldn't access the login form for the Cloud storage, this was fixed by adjusting some values in the conf.int file, and reloading nginx.

I couldn't access the Cloud storage from the internet, but this was due to mem scrambling the IP address in Cloudflare, so I disabled that on my subdomains, which fixed the issue.

The Cloud Storage wouldn't allow files bigger than 2.5 GB to be uploaded due to issues with chunking failing and crashing, this is a three-year-old bug in the software and was fixed by increasing the timeout limit and using the Synology to upload larger files.

These have all been fixed!

Final Thoughts

- Although this project has been challenging at times it has taught me a wide variety of essential skills necessary for working with and around servers, the server software and the issues that arise from it, I have developed critical problem solving skills when in a difficult situation and have grown to understand more about how our world works and the technologies behind them.
- Ultimately I give my thanks to the Henry Morris Trust who have graciously funded my project, and to my friends and parents who encouraged me along the way.