Server migration plan

Current server configuration:

Processor	Intel i7- 4770K [4 core, 8 thread, 3.5-3.9 GHz]
Memory	8 GB
Storage	4 TB (2 TB + 2 TB)

New Server configuration:

Processor	AMD Ryzen 5 8500G [6 core, 12 thread, 3.5-5 GHz]
Memory	32 GB (16 GB + 16 GB)
Storage	4 TB (2 TB + 2 TB) + 240 GB SSD

Currently, on the main server, we have two shared folders:

1. **Share**
2. **Oldshare**

Among these, we are only backing up the Share folder. We are not backing up the Oldshare folder, which contains some critical data. When the server crashed, we encountered difficulties recovering the data in Oldshare.

We are also backing up the Share folder as well as the database. These backups are taken daily, weekly, and monthly. Apart from this we also have a Sync folder on the server, which contains the laptop backups via syncthing. Currently, we are not backing up the Sync folder, but going forward we will take a backup of this folder as well.

While setting up the new server we will not be migrating the Oldshare - We will be proceeding with only one shared folder - Before setting this up we will need to perform a shared folder sanitization activity in which only the important data/files from Oldshare is carried over to the share folder; The current share folder will also have to be sanitized to ensure we are in optimal running condition.

- Install OS:

    We are installing **Debian 12.7** on our server because it offers stability, security, and performance, making it ideal for reliable and efficient server operations.

- Set up partitions:

    When we are going to migrate we are going to use the new partition.

Machine	Drive	Partition	Size of Partition	Mount Path	Size of Logical Division
Main Server	SSD 240 GB	OS	120	/ [OS]	
		Home	70	/home	
		Swap	38	/swap	
		Boot	2	/boot	
		SQL Backup	10	/backup/4/sql	
	HDD 1 2048 GB	SQL Origin	10	/sql	
		Share Origin	500	/share	
		Sync Origin	500	/sync	
				/backup/1/sql	10
				/backup/1/share	500
		Backup 1	1010	/backup/1/sync	500
		OS Config-1	20	/home/config/1/	
	HDD 2 2048 GB			/backup/2/sql	10
				/backup/2/share	500
		Backup 2	1010	/backup/2/sync	500
			1010	/backup/3/sql	10
				/backup/3/share	500
		Backup 3		/backup/3/sync	500
		OS Config-2	20	/home/config/2/	
DR Server	HDD 1 1024 GB	OS	120	/ [OS]	
		Home	72	/sql	
		Swap	10	/share	
		Boot	2	/sync	
		Long Term Storage	310	/storage	
			510	/backup/3/sql	10
		Backup 3		/backup/3/share	500
	HDD 2 2048 GB			/backup/1/sql	10
				/backup/1/share	500
		Backup 1	1020	/backup/1/sync	500
			1020	/backup/2/sql	10
				/backup/2/share	500
		Backup 2		/backup/2/sync	500
		OS Config-3	20	/home/config/2/	

- Set up users [fas, sysadmin, backup]
    - sudo useradd -m -s bash fas
    - sudo useradd -m -s bash sysadmin
    - sudo useradd -m -s bash backup
- Install mysql

  Refreshing list of sources:
    - sudo apt-get update

  Downloading latest mysql sources:
    - cd /tmp
    - wget https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb

  Adding sources to dpkg (Debian Package Manager):
    - sudo dpkg -i mysql-apt-config*

  Refreshing list of sources again:
    - sudo apt-get update
- Set up mysql:

  Installing MySQL Server:
    - sudo apt install mysql-server

  Check if MySQL is active:
    - sudo systemctl status mysql

  Secure the MySQL installation:
    - sudo mysql_secure_installation

  Test if MySQL can be logged into:
    - mysql -u root -p

  Create the lower privileged user ‘user’ & grant read-only privileges.
    - CREATE USER 'user'@'192.168.1.2' IDENTIFIED BY 'fas111';

– GRANT SELECT PRIVILEGES ON * . * TO 'user@'192.168.1.2';

- Install Samba:
  - sudo apt install samba

Configure Samba:

1.Restore and edit the samba config file from backup [smb.conf]

2. Create the shared directory and set appropriate permissions:
   - sudo mkdir -p /home/Share
   - sudo chown nobody:nogroup /home/Share
   - sudo chmod 2775 /home/Share
3. Add a Samba user:
   - sudo smbpasswd -a your_username
4. Restart Samba service to apply changes:
   - sudo systemctl restart smbd
   - sudo systemctl restart nmbd

- Check if MySQL and Samba are Accessible from the Network:

1. Check MySQL connectivity:

From another machine, use the following command to check if MySQL is accessible:

- mysql -u your_username -p -h your_mysql_server_ip

2. Check Samba connectivity:

From another machine, use the following command to list available shares:

- smbclient -L //your_samba_server_ip -U your_username

3. Diagnose any connectivity issues:

For MySQL:

- Ensure MySQL is running: `sudo systemctl status mysql`

- Verify MySQL is listening on the correct IP and port: `netstat -plnt | grep mysql`

- Check firewall settings to ensure port 3306 is open.

For Samba:

- Ensure Samba is running: `sudo systemctl status smbd nmbd`

- Check firewall settings to ensure port 137, 138, 139, and 445 are open.

- Use `smbclient -L //localhost` to test local access.

- Install Syncthing:

1. Add the Syncthing APT repository:

   – sudo apt install -y apt-transport-https

   – sudo mkdir -p /etc/apt/keyrings

   – curl -fsSL https://syncthing.net/release-key.asc | sudo tee/etc/apt/keyrings/syncthing.asc

– echo "deb [signed-by=/etc/apt/keyrings/syncthing.asc] https://apt.syncthing.net/ stable binary/" | sudo tee /etc/apt/sources.list.d/syncthing.list

2. Update the package list:

   – sudo apt update

3. Install Syncthing:

   – sudo apt install syncthing

4. Start and enable Syncthing service:

   – sudo systemctl start syncthing

   – sudo systemctl enable syncthing

- Set Up Syncthing in the laptop:

- Access Syncthing Web Interface:
    - By default, Syncthing's web interface is accessible at http://localhost:8384.
    - Open the web browser and go to this address to access the Syncthing UI.
- Initial Configuration:
    - Add Devices:
        - In the Syncthing web interface, click on "Add Device".
        - Enter the device ID of the other device you want to sync with. The device ID can be found in the Syncthing web interface on the other device.
        - Optionally, give the device a name and set any other options you prefer.
        - Click "Save" to add the device.
    - Add Folders:
        - Click on "Add Folder" in the Syncthing web interface.
        - Choose a folder on your local machine that you want to sync. This can be a new folder or an existing one.
        - Assign a label and unique folder ID.
        - Configure folder sharing settings as needed (e.g., sync mode, folder path).
        - Click "Save" to add the folder.
    - Share Folders with Devices:
        - After adding a folder, you'll need to share it with the devices you added earlier.
        - Go to the folder's settings and select the devices you want to share the folder with.
        - Click "Save" to confirm.

- Migrate Data from current server to New server:

    - Migrate Share folder from the server:

        – rsync-avz--progress fas@192.168.1.2:/home/fas/Share /Share

- Migrate Database:

1. Export the Database using mysqldump:

– mysqldump -u root -p mydatabase /sql/mydatabase.sql

2. Transfer the Backup File to the New Server:

– rsync--avz fas@192.168.1.2:/home/sql/database.sql /sql

3. Import the Database on the New Server:

– mysql -u username -p new_database_name < /sql/database.sql

4. Verify the Migration:

1. Log in to the MySQL on the New Server:

   – mysql -u username -p

2. Check if the Data is Present:

   – USE new_database_name;

   – SHOW TABLES;

   – SELECT COUNT(*) FROM table_name; [Check row counts for important tables]

Potential Risks:

1. When we migrate the data from the server, we encountered some difficulties. When the server crashed, we took a backup in live boot mode. We backed up critical data during that week, spending at least a week on the process. Sometimes, the process stopped because certain files required root access.
2. When we migrate the database, we need to cross-check it to ensure that the entire database has been migrated from the old server.

Action Taken:

1. Create a script to identify files that require root access. During migration, we need to separate these files and copy them separately.
2. Create a script to cross-check the database that we copied from the old server.

Note : There is no Data Migration for Syncthing - Syncthing will automatically pull the latest data from FAS laptops - any manual migration of data is unnecessary.