I understand now that you're asking about using Bitvise for FTP (File Transfer Protocol) versus using it for SSH terminal access when backing up your Ubuntu VPS. Here’s how you can approach both methods:

### Using Bitvise SSH Terminal

1. \*\*Connect via SSH with Bitvise\*\*:

- Open Bitvise SSH Client.

- Enter the IP address or domain of your VPS.

- Specify the SSH port (usually 22 by default).

- Enter your SSH username and password (or private key if using key-based authentication).

- Click "Login" to establish the SSH connection.

2. \*\*Backup Your VPS\*\*:

- Within the SSH terminal in Bitvise, execute commands to create backups, such as:

```bash

# Example: Backup important directories

sudo tar czvf /path/to/backup/backup.tar.gz /path/to/important/directory

# Example: Backup MySQL/MariaDB database

sudo mysqldump -u username -p database\_name > /path/to/backup/database\_backup.sql

```

- Adjust paths (`/path/to/backup`, `/path/to/important/directory`, `username`, `database\_name`) as per your setup.

3. \*\*Download Backups\*\*:

- After creating backups, you can download them using Bitvise’s SFTP (Secure File Transfer Protocol) feature.

- Switch to the SFTP tab in Bitvise.

- Navigate to the directory where backups are stored (`/path/to/backup`) and download the files to your local machine.

### Using Bitvise FTP

Bitvise SSH Client also includes an integrated SFTP client, which you can use similarly to traditional FTP for file transfer tasks.

1. \*\*Connect via SFTP with Bitvise\*\*:

- Open Bitvise SSH Client.

- Enter the IP address or domain of your VPS.

- Specify the SSH port (usually 22 by default).

- Enter your SSH username and password (or private key if using key-based authentication).

- Click "Login" to establish the SSH connection.

2. \*\*Transfer Files\*\*:

- In Bitvise, switch to the "SFTP" tab.

- Navigate to the directory on your VPS where backups are located (`/path/to/backup`).

- Select the files you want to download and drag them to a directory on your local machine.

3. \*\*Backup Process\*\*:

- You still need to use the SSH terminal within Bitvise to execute commands for creating backups (`tar`, `mysqldump`, etc.) as described earlier.

- SFTP in Bitvise is used primarily for transferring the backup files to your local machine after they have been created.

### Choosing Between SSH Terminal and FTP

- \*\*SSH Terminal\*\*:

- \*\*Pros\*\*: Direct access to execute commands and scripts on your VPS, which can include backup creation and management.

- \*\*Cons\*\*: Requires knowledge of Linux commands and scripting.

- \*\*FTP (SFTP)\*\*:

- \*\*Pros\*\*: Easy graphical interface for file transfer tasks.

- \*\*Cons\*\*: Limited to file transfer and cannot execute commands directly on the server.

### Additional Tips

- Ensure your backups are stored securely, ideally offsite or in a separate location.

- Consider automating backups using cron jobs (`cron`) for scheduled tasks.

- Encrypt sensitive backup data before transferring it.

By leveraging Bitvise SSH Client for both terminal access and SFTP file transfer, you can efficiently manage backups for your Ubuntu VPS while ensuring data security and accessibility.

Is this way provide me a full backup includes all my APPs, Dependencies, Configurations, Environment & etc...? so, if I need to restore my VPS Contents just I run restoring Commands? of Course after install my Ubuntu OS only?

Yes, the method described using Bitvise SSH Client allows you to create backups that include your applications, dependencies, configurations, environment settings, and other data on your Ubuntu VPS. However, it's important to clarify a few points regarding what is typically included in such backups and the restore process:

### What's Included in the Backup

1. \*\*Applications\*\*: Any installed software and applications on your VPS, including their configurations.

2. \*\*Dependencies\*\*: Libraries and packages required by your applications and system.

3. \*\*Configurations\*\*: Configuration files for your applications, services (like Apache/Nginx, MySQL/PostgreSQL, etc.), and system settings.

4. \*\*Environment\*\*: Environment variables and settings specific to your applications.

5. \*\*Data\*\*: This includes any user data, databases (if dumped), and files stored on the VPS.

### Creating a Full Backup

To ensure you have a comprehensive backup, you typically:

- \*\*Archive directories\*\*: Use `tar` or similar tools to compress and archive important directories containing applications, configurations, and data.

- \*\*Database backups\*\*: Use tools like `mysqldump` or `pg\_dump` to create backups of your databases.

- \*\*Configuration files\*\*: Copy or archive important configuration files (`/etc` directory and specific application directories).

### Restore Process

When it comes to restoring your VPS from such a backup, the process generally involves:

1. \*\*Install Ubuntu OS\*\*: Install a fresh copy of Ubuntu on your VPS.

2. \*\*Transfer Backup Files\*\*:

- Use Bitvise or similar tools to transfer your backup archive files (created earlier) back to your VPS.

3. \*\*Extract Archives\*\*:

- Extract the archived directories (`tar -xvf backup.tar.gz`) to restore application files, configurations, and data.

4. \*\*Database Restoration\*\*:

- Restore databases using the backup files (`mysql -u username -p database\_name < database\_backup.sql`).

5. \*\*Configuration\*\*:

- Replace or merge configuration files from your backup into the appropriate directories (`/etc`, application directories).

6. \*\*Environment Setup\*\*:

- Set up environment variables and any specific settings required by your applications.

7. \*\*Verification and Testing\*\*:

- Verify that all applications, services, and data have been restored correctly.

### Important Considerations

- \*\*Backup Frequency\*\*: Regularly schedule backups to ensure you have up-to-date copies of your VPS data.

- \*\*Security\*\*: Encrypt sensitive data before creating backups to ensure data security during transfer and storage.

- \*\*Documentation\*\*: Keep documentation of your backup and restore procedures for quick reference.

By following these steps, you can create a comprehensive backup of your Ubuntu VPS using Bitvise SSH Client and restore it when needed, ensuring minimal downtime and data loss in case of server issues or upgrades. Always test your backups periodically to ensure they are functioning correctly and contain all necessary data for restoration.

how to Archive directories use tar, backup database and how to merge configuration directories into my backup? also how to make the full backup as a cron job?

To create a comprehensive backup of your Ubuntu VPS, including archiving directories, backing up databases, merging configuration directories, and scheduling it as a cron job, follow these steps:

### 1. Archive Directories using `tar`

To archive directories, you can use the `tar` command. Here’s how you can create a compressed archive of important directories:

```bash

# Create a tar archive of directories

sudo tar czvf /path/to/backup/backup.tar.gz /path/to/directory1 /path/to/directory2

```

- \*\*Explanation\*\*:

- `sudo`: Run the command with superuser (root) privileges if necessary.

- `tar`: The command-line utility for archiving files and directories.

- `czvf`: Options for `tar`:

- `c`: Create a new archive.

- `z`: Compress the archive using gzip.

- `v`: Verbose mode to show progress (optional).

- `f`: Specify the filename of the archive.

- `/path/to/backup/backup.tar.gz`: Destination path and filename of the backup archive.

- `/path/to/directory1`, `/path/to/directory2`: Paths of directories you want to include in the backup.

### 2. Backup Databases

To backup databases like MySQL or PostgreSQL, use their respective backup commands. Here’s an example for MySQL using `mysqldump`:

```bash

# Backup MySQL database

sudo mysqldump -u username -p database\_name > /path/to/backup/database\_backup.sql

```

- Replace `username`, `database\_name`, and `/path/to/backup/database\_backup.sql` with your actual MySQL username, database name, and backup destination path.

### 3. Merge Configuration Directories into Backup

If you have specific configuration directories you want to include in the backup, ensure they are archived along with other directories. For example, if your configurations are in `/etc` or other application-specific directories, include them in the `tar` command.

### 4. Create a Full Backup Script

You can create a script that includes all these backup commands. For example, create a file `backup\_script.sh`:

```bash

#!/bin/bash

# Backup important directories

sudo tar czvf /path/to/backup/backup.tar.gz /path/to/directory1 /path/to/directory2

# Backup MySQL database

sudo mysqldump -u username -p database\_name > /path/to/backup/database\_backup.sql

```

Make the script executable:

```bash

chmod +x backup\_script.sh

```

### 5. Schedule Backup as a Cron Job

To automate this backup process, you can schedule the script to run periodically using cron. Edit your crontab by running `crontab -e` and add a line like this to run the script daily at 2 AM:

```bash

0 2 \* \* \* /path/to/backup\_script.sh

```

- This line specifies the schedule in the format `minute hour day-of-month month day-of-week command`. Adjust the timing (`0 2 \* \* \*` means 2 AM every day) as needed.

### Important Notes:

- \*\*Security\*\*: Ensure that your backup files (`backup.tar.gz`, `database\_backup.sql`) are stored securely and are accessible only by authorized users.

- \*\*Testing\*\*: Periodically test your backups by restoring them to ensure they contain all necessary data and configurations.

- \*\*Monitoring\*\*: Set up alerts or notifications to monitor the success or failure of your cron jobs and backups.

By following these steps, you can create a comprehensive backup solution for your Ubuntu VPS, including directories, databases, and configurations, and automate it using cron for regular backups. Adjust paths and commands based on your specific setup and requirements.

<https://help.ubuntu.com/community/BackupYourSystem/TAR>

<https://help.ubuntu.com/community/BackupYourSystem>