Automating GitHub Repository Synchronization with Hostinger VPS

Stage 1: Setting up the Connection Between GitHub and Hostinger VPS

In this stage, we established a connection between the GitHub account and the Hostinger VPS to synchronize repositories. Here are the steps we followed, including terminal commands and their explanations:

1. Installing Necessary Tools

Commands:

```
sudo apt update
sudo apt install git
```

Explanation:

- sudo apt update: Updates the package list to ensure the latest software versions are available.
- sudo apt install git: Installs Git, a version control system required to clone repositories from GitHub.

2. Generating an SSH Key for GitHub

Commands:

```
ssh-keygen -t ed25519 -C "your-email@example.com"
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_ed25519
cat ~/.ssh/id_ed25519.pub
```

Explanation:

- ssh-keygen -t ed25519 -C "your-email@example.com": Generates an SSH key for authentication with GitHub.
- eval "$(ssh-agent -s)": Starts the SSH agent to manage keys.

- ssh-add ~/.ssh/id_ed25519: Adds the generated SSH key to the agent.
- cat ~/.ssh/id_ed25519.pub: Outputs the public key, which is added to GitHub for secure communication.

3. Adding the SSH Key to GitHub

- Copy the public key displayed after running cat ~/.ssh/id_ed25519.pub.
- Log in to GitHub and go to Settings > SSH and GPG keys.
- Click New SSH Key, give it a name (e.g., "Hostinger VPS"), paste the key, and save.

4. Cloning Repositories

Command:

```
mkdir ~/github-repos
cd ~/github-repos
curl -u "your-github-username:your-github-token" https://api.github.com/user/repos?per_page=100 | grep -o 'git@[^"]*' | while read repo; do git clone $repo; done
```

Explanation:

- mkdir ~/github-repos: Creates a directory to store all GitHub repositories.
- cd ~/github-repos: Navigates to the directory.
- curl -u "your-github-username:your-github-token": Fetches a list of repositories using your GitHub username and token.
- grep -o 'git@[^"]*': Extracts the SSH URLs of the repositories.
- while read repo; do git clone $repo; done: Loops through each URL and clones the repository.

Stage 2: Automating the Synchronization Process

In this stage, we automated the process to synchronize GitHub repositories periodically.

1. Writing the Automation Script

Script:

```
#!/bin/bash

cd /root/github-repos

curl -u "Adityajr10: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"

https://api.github.com/user/repos?per_page=100 | grep -o 'git@[^"]*' | while read repo; do

    repo_name=$(basename "$repo" .git)

    if [ ! -d "$repo_name" ]; then

        git clone "$repo"

    else

        cd "$repo_name"

        git pull

        cd ..

    fi

done
```

Explanation:

- The script navigates to the github-repos directory.

- It fetches a list of repositories from GitHub and clones any new repositories.

- For existing repositories, it pulls the latest changes.

2. Making the Script Executable

Command:

```
chmod +x ~/sync_github_repos.sh
```

Explanation:

- chmod +x: Grants execute permissions to the script.

3. Automating with Cron

Command:

```
crontab -e
```

Add the following line to the crontab file:

```
0 * * * * /bin/bash ~/sync_github_repos.sh
```

Explanation:

- 0 * * * *: Schedules the script to run at the start of every hour.

- /bin/bash ~/sync_github_repos.sh: Executes the synchronization script.

4. Verifying the Cron Job

Command:

```
crontab -l
```

Explanation:

- Lists all active cron jobs to confirm the synchronization script is scheduled.

How the Automation Works:

1. The script ensures the github-repos directory is always up-to-date with your GitHub account.

2. Cron runs the script hourly, automating the sync process without manual intervention.

3. Any new repositories are cloned, and updates are pulled for existing ones.

Benefits of the Process:

1. Efficiency: Automates repetitive tasks of syncing repositories.

2. Scalability: Handles multiple repositories seamlessly.

3. Security: Uses SSH for secure communication with GitHub.

Feel free to share this document with your client to help them understand the setup and automation process clearly.