To achieve the objectives outlined, you'll need to follow these steps:

1. **Download Website Files:** Download the website files from the provided GitHub link.
2. **Write Dockerfile:** Create a Dockerfile to build an image that includes your company's website files and a web server (e.g., Apache). Here's an example Dockerfile:

DockerfileCopy code

# Use the official httpd image as the base FROM httpd:latest # Copy the website files to the appropriate directory COPY ./path/to/your/website/files /usr/local/apache2/htdocs/ # Expose port 80 EXPOSE 80

1. **Build Docker Image:** Build the Docker image using the Dockerfile. Open a terminal in the directory containing your Dockerfile and run:

bashCopy code

docker build -t your-image-name .

1. **Use Volumes:** To ensure that the website data is stored outside the container, you can use Docker volumes. When running the container, you can specify the volume option to map a directory on the host machine to a directory inside the container where the website files are located.

bashCopy code

docker run -d -p 80:80 -v /path/on/host:/usr/local/apache2/htdocs your-image-name

1. **Push to Docker Hub:** Push the built image to Docker Hub so that it can be pulled later. You need to create a Docker Hub account and log in using **docker login** before pushing.

bashCopy code

docker tag your-image-name your-dockerhub-username/your-image-name docker push your-dockerhub-username/your-image-name

1. **Create a Swarm Cluster:** If you haven't already, initialize a Docker Swarm cluster. Run the following command on one of your Docker hosts:

bashCopy code

docker swarm init

1. **Deploy to Swarm Cluster:** Now, you can deploy your website on the Swarm cluster using a Docker Compose file. Create a **docker-compose.yml** file with the following content:

yamlCopy code

version: '3.7' services: web: image: your-dockerhub-username/your-image-name ports: - "80:80" volumes: - /path/on/host:/usr/local/apache2/htdocs

Deploy the stack using the following command:

bashCopy code

docker stack deploy -c docker-compose.yml your-stack-name

1. **Access the Website:** Once the stack is deployed, you should be able to access the website by navigating to your Docker Swarm cluster's IP address or hostname in a web browser.

Remember to replace placeholders like **/path/to/your/website/files**, **your-image-name**, **your-dockerhub-username**, **/path/on/host**, and **your-stack-name** with the appropriate values in the commands and configuration files.