File mounting forbidden issue

The error `403 Forbidden` indicates that the Apache HTTP server running inside the container is blocking access to the requested resource. This could be due to a number of reasons, such as file permissions or configuration settings in the container.

Let's break down some common causes and solutions:

### 1. \*\*File Permissions\*\*

Since you're mounting the current directory (`$(pwd)`) to `/usr/local/apache2/htdocs` in the container, make sure the files in the current directory have proper read permissions for the user running Apache inside the container. If the files are not readable by the Apache user (`www-data` in some distributions), you'll see a `403` error.

\*\*Solution:\*\*

- Check the permissions of the files in the current directory.

```bash

ls -l $(pwd)

```

Ensure that the files have read permissions for all users or at least for the user running Apache inside the container.

```bash

chmod -R 755 $(pwd)

```

### 2. \*\*Apache Configuration (Directory Permissions)\*\*

The `403 Forbidden` error may also occur if Apache is configured to restrict access to the `/usr/local/apache2/htdocs` directory. The `httpd` image may have a default configuration that denies access.

\*\*Solution:\*\*

You can check or adjust the Apache configuration to ensure it allows access to the directory.

- Open the `httpd.conf` or `.htaccess` file (if any) and ensure the following configuration is present to allow access to the document root:

```apache

<Directory /usr/local/apache2/htdocs>

AllowOverride All

Require all granted

</Directory>

```

If there’s a `.htaccess` file in the mounted directory (`$(pwd)`), check its contents for any `Deny` or `RewriteRule` directives that could cause a `403` error.

### 3. \*\*Check the Default Apache Directory Permissions\*\*

The default configuration for the Apache container might have strict permissions, such as only allowing access to certain paths. For example, the `htdocs` directory inside the container might be restricted.

\*\*Solution:\*\*

You can specify a different directory for testing purposes. For example, try copying some simple HTML files to a different location, such as `/tmp`, and update the mount point accordingly:

```bash

docker run --name web1 -d -p 9200:80 -v /tmp:/usr/local/apache2/htdocs httpd

```

### 4. \*\*SELinux (if applicable)\*\*

If your system is running SELinux (common on some Linux distributions like CentOS or RHEL), it might be restricting access to the mounted volume. You could check whether SELinux is causing the issue by running the following command:

```bash

sudo getenforce

```

If it's set to `Enforcing`, SELinux might be blocking access to the container’s files. You can temporarily disable SELinux to check if that's the cause:

```bash

sudo setenforce 0

```

If the `403` error goes away after disabling SELinux, you’ll need to adjust the SELinux policy to allow Docker/Podman containers to access the mounted directory.

### 5. \*\*Logs\*\*

Finally, you can check the Apache error logs inside the container to get more details on why the request was forbidden. First, find the container ID:

```bash

docker ps

```

Then, access the container's logs:

```bash

docker logs <container\_id>

```

This might give you more detailed information on the root cause of the `403` error.

---

### Next Steps:

- First, check and update the file permissions for the directory you're mounting.

- Verify if any Apache configurations (in the container) are restricting access to `/usr/local/apache2/htdocs`.

- Review the Apache logs for more specific error messages.

Let me know if you need further assistance!