
**CAHIER DES
SPECIFICATIONS DE
L'ADMINISTRATION DE
L'APPLICATION
GRAPHIQUE WEB
(RCTsDash)**

Table des matières

Mots -clés	3
Contexte	3
1. Administration pour la création de nouveau Dashboard	3
1.2. Localisation des fichiers dans le serveur	3
1.2.1 Docker.....	4
1.2.2 File Station	5
1.2.3 Le dossier dash.....	7
1.3. Le programme manager.....	7
1.4. Création du Dashboard.....	7
2. Administration gestion côté serveur	8
2.1. Lecture du fichier log	8
2.2. Les permissions sur Docker	8
2.3. Création d'un conteneur Docker	10
2.4. Redémarrage d'un conteneur Docker.....	10

Mots -clés

RCTsDash : Le nom de l'application en cours de développement

CDC : Cahier Des Charges

SHINY : langage de programmation web en langage R

R : langage de programmation statistique

SAP : Plan d'analyse statistique.

Contexte

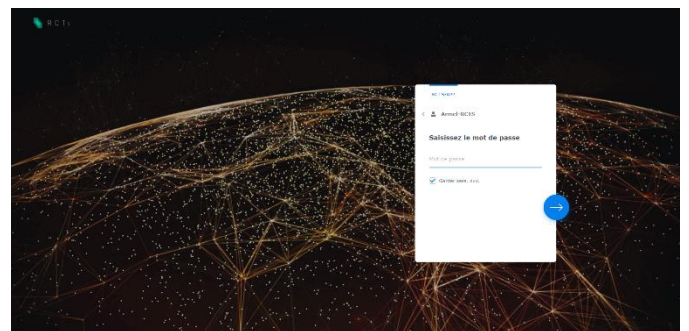
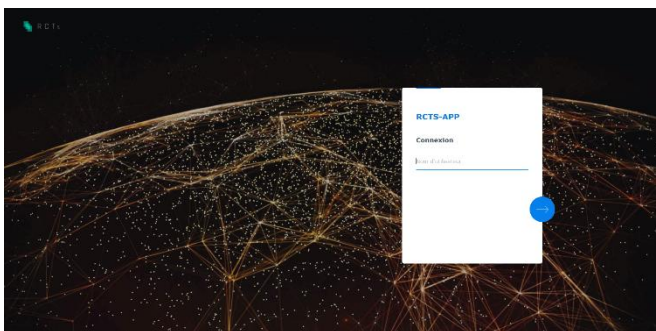
RCTS a manifesté le besoin d'avoir un outil ou logiciel de visualisation de données.

Cet outil devrait se baser sur les résultats d'analyse statistique appelé RTF en format sas dataset ou csv pour afficher des visualisations interactives.

1. Administration pour la création de nouveau Dashboard

1.2. Localisation des fichiers dans le serveur

L'accès aux dossiers d'hébergement se fait à travers le serveur. IL faudrait avoir des identifiants de connexion à cet effet. Ces identifiants sont créés par le IT (Didier NOT).



La page d'accueil après l'accès au serveur se présente comme suite :

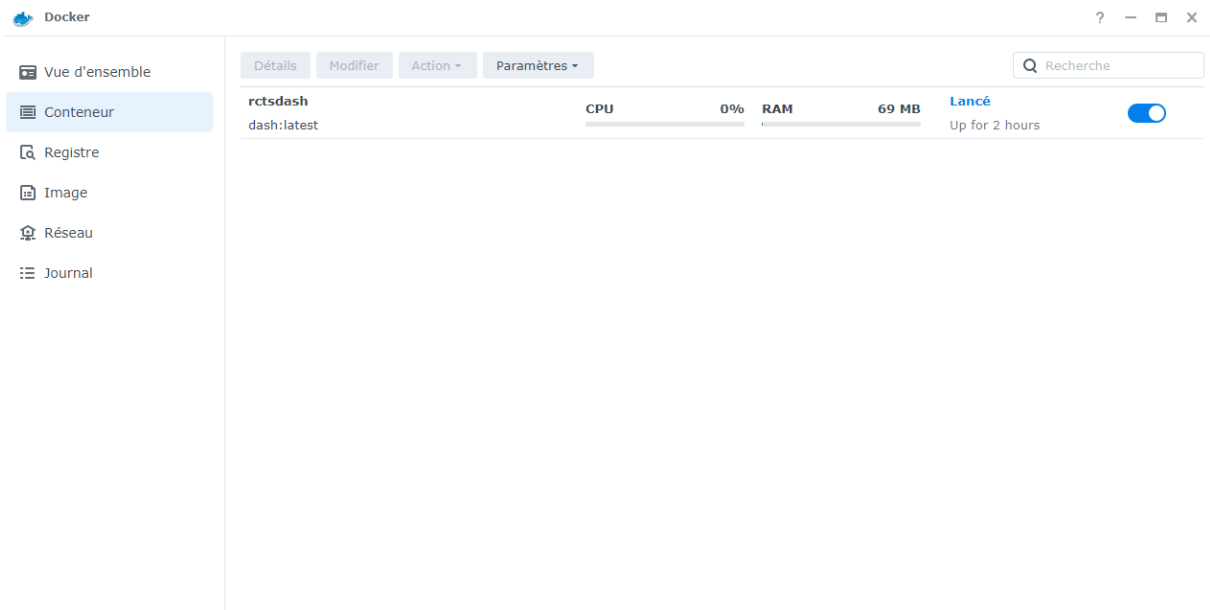


1.2.1 Docker

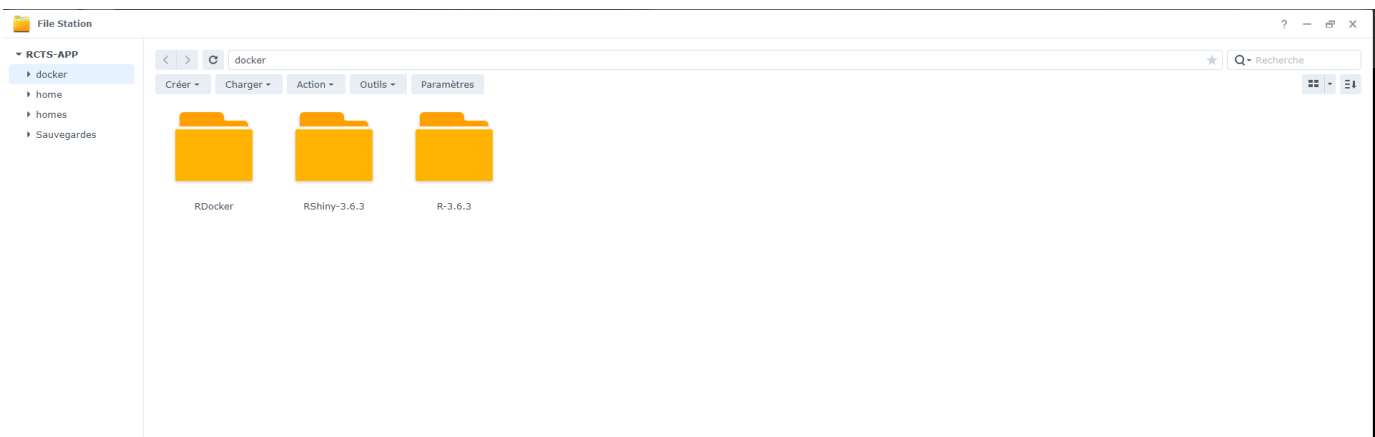
L'application docker permet de voir les conteneurs et les images en cours d'utilisation. On a l'image **dash** et le conteneur **rctsdash** qui permettent d'exécuter les applications rshiny.

Le conteneur doit rester en marche pour que l'application puisse être exécuter. Lorsque le serveur s'éteint ou que l'application docker est mise à jour, cela entraîne l'arrêt du conteneur qui doit être redémarrer pour permettre le fonctionnement des applications (Cf. section 2.4).

On peut également avoir accès aux journaux sur le fonctionnement du conteneur. Un journal est créé à chaque exécution d'une application qu'il héberge. En indiquant le nom de l'utilisateur à la base de l'action et du type d'action.



1.2.2 File Station



A travers file station, on accède au dossier docker qui contient :

- Le dossier RDocker

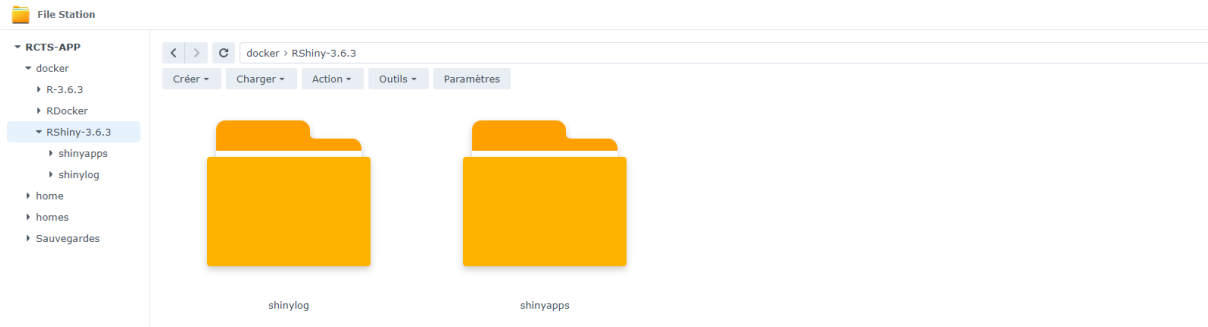
Le dossier R Docker contient les fichiers de configuration du conteneur. Il s'agit du fichier de configuration et le Docker file. Ils n'influencent en aucun cas le fonctionnement du conteneur mais contiennent la configuration utilisée pour le créer. Ce dossier peut donc être supprimé.

- Le dossier R + R studio

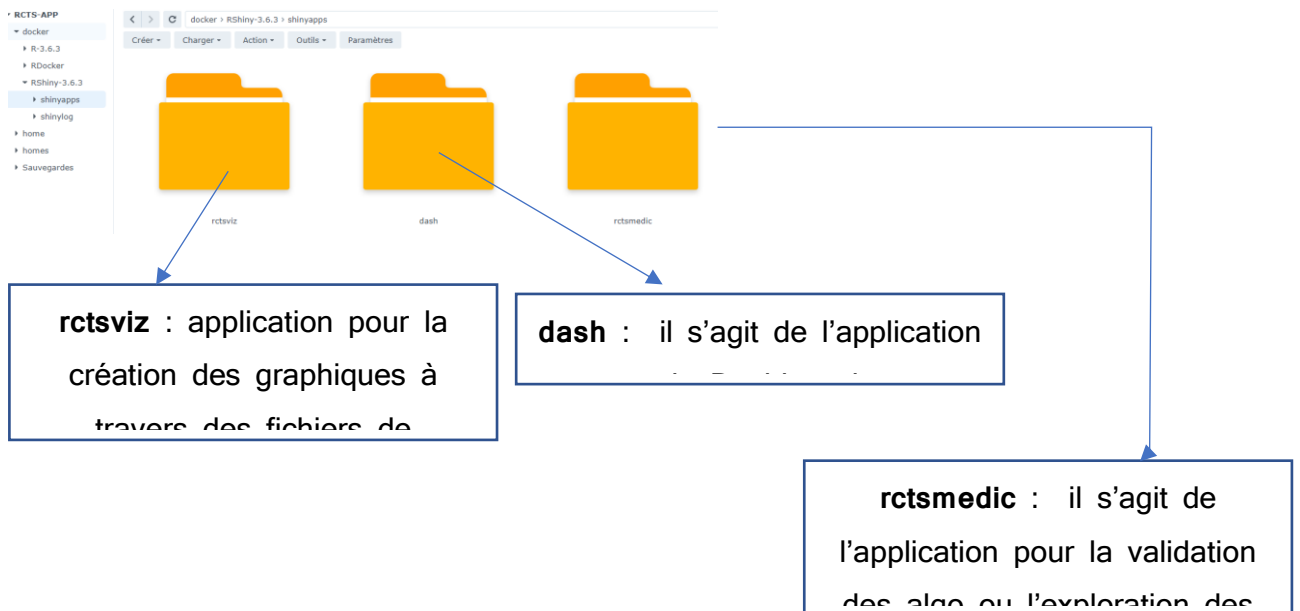
Ce dossier contient le dossier permettant d'exécuter R et R Studio sur le serveur (permettant de ne pas à l'installer sur les postes clients).

- Le dossier Rshiny serveur

Il s'agit du dossier qui contient les application r Shiny (**shinyapps**) et les logs de fonctionnement de chaque application (**shinylog**).

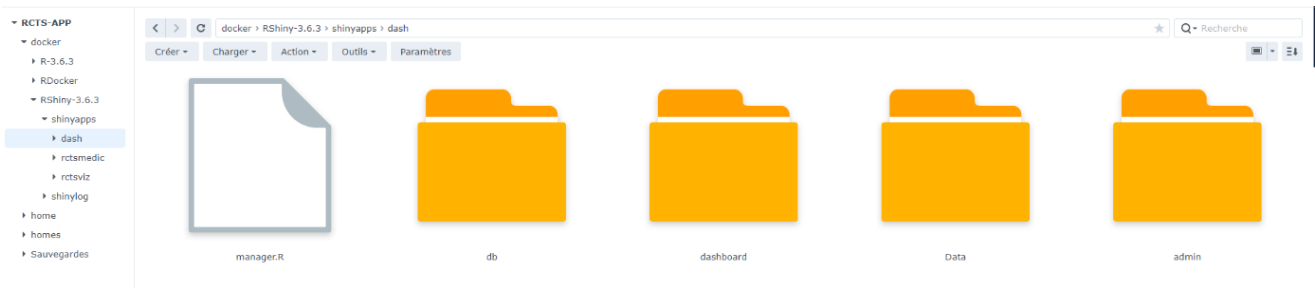


- A chaque session créée correspond un fichier log qui liste les actions et les erreurs survenues.
- Shinyapps héberge toutes les applications en cours de fonctionnement et les futures applications. Il y a donc un dossier par application. A ce jour nous avons trois applications. A noter que ces applications sont exécutées sur le même conteneur.



1.2.3 Le dossier dash

Le dossier dash contient tout le code source de l'application ainsi que les bases de



données.

Le dossier db contient les bases de données sqlite. Il s'agit de la base de gestion des utilisateurs et celles hébergeant tout le contenu textuel de l'application (les éditeurs de textes, les notifications, ...)

Les dossiers **dashboard** et **admin** contiennent respectivement les codes r, javascript, css pour chaque partie respective de l'application.

Le dossier Data stock l'historique des données (RTF en Excel) et les fichiers html utilisés pour les analyses.

1.3. Le programme manager

Le programme manager permet de réinitialiser toutes les bases de données et les contenus du Dashboard. En gros il doit être exécuté lors de la création d'un nouveau Dashboard.

Cette action peut être faite à travers la page d'administration et est accessible uniquement au super user. A travers un bouton qui permet l'exécution du programme.

1.4. Création du Dashboard

La création d'un nouveau Dashboard est très simple à travers les étapes suivantes :

- Faire une copie du dossier dash contenant le code source de l'application. Il faudra absolument renommer le nouveau dossier avec un nom court mais représentant l'étude. A noter que le nom du dossier sera utilisé dans les url d'accès aux différentes pages. Par exemple : [http// :dash/dashboard](http:// :dash/dashboard)

- Les nouveaux urls des pages seront sous le format : [http// : <<ip>> :3838/<<nom du dossier>>/dashboard](http://<<ip>>:3838/<<nom du dossier>>/dashboard) [http// : <<ip>> :3838/<<nom du dossier>>/admin](http://<<ip>>:3838/<<nom du dossier>>/admin)
- La copie doit se situer dans le dossier shinyapps pour être exécuter par le conteneur,
- A ce stade, personne d'autre ne pourrait accéder à ce Dashboard sauf le super user. Le super user devrait accéder pour exécuter le programmer manager et créer les futurs administrateurs.

2. Administration gestion côté serveur

2.1. Lecture du fichier log

Comme mentionner plus haut, chaque session démarrée crée un fichier log. Ces fichiers log permettent d'observer l'exécution du code et surtout la lecture des erreurs.

```
su: ignoring --preserve-environment, it's
mutually exclusive with --login

Attaching package: 'shinyjs'

The following object is masked from
'package:shiny':

  runExample

The following objects are masked from
'package:methods':

  removeClass, show
```

Les erreurs possibles sont entre autres :

- Absence d'autorisation de lecture et d'écriture
- Lorsqu'un chemin d'accès n'est pas trouvé
- Lorsque les données de lecture ne sont pas au bon format

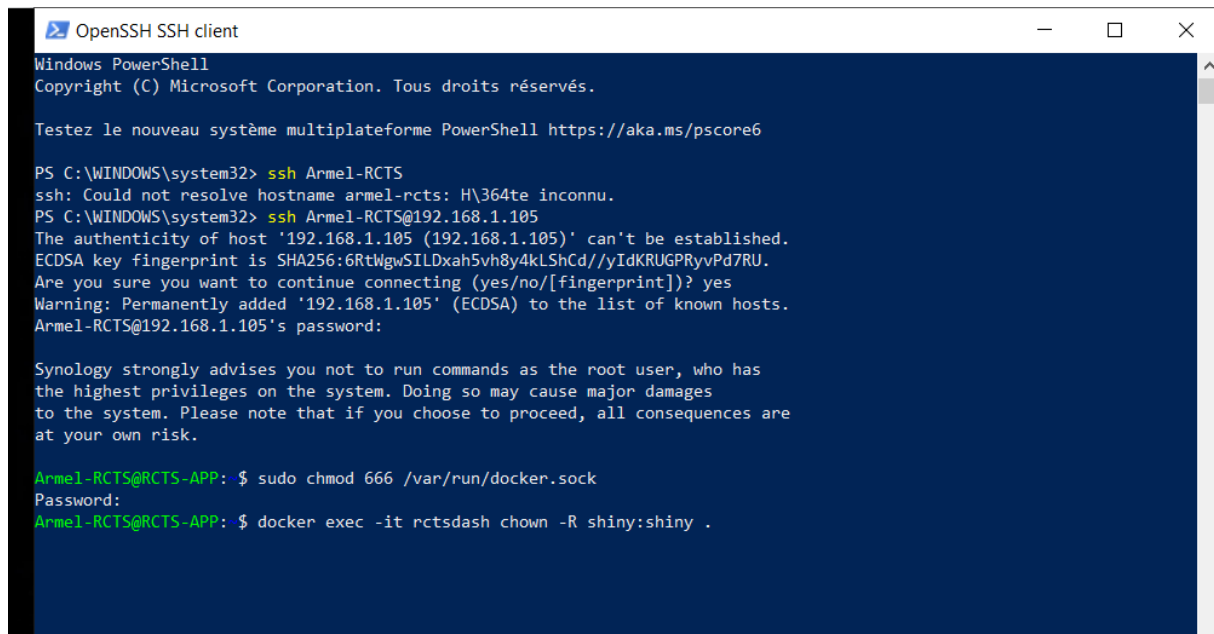
2.2. Les permissions sur Docker

Il y'a deux permissions possibles qui peuvent être demandées sur docker :

Autoriser docker à comprendre les commande sudo. Pour cela avant toute commande sur docker, il faut permettre cette autorisation à travers la commande suivante :

```
sudo chmod 666 /var/run/docker.sock
```

Mais il faut d'abord se connecter au server à travers son cmd en utilisant les lignes de codes suivantes :



```
OpenSSH SSH client
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> ssh Armel-RCTS
ssh: Could not resolve hostname armel-rcts: H\364te inconnu.
PS C:\WINDOWS\system32> ssh Armel-RCTS@192.168.1.105
The authenticity of host '192.168.1.105 (192.168.1.105)' can't be established.
ECDSA key fingerprint is SHA256:6RtWgwSILDxah5vh8y4kLSHCd//yIdKRUGPRyvPd7RU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.105' (ECDSA) to the list of known hosts.
Armel-RCTS@192.168.1.105's password:

Synology strongly advises you not to run commands as the root user, who has
the highest privileges on the system. Doing so may cause major damages
to the system. Please note that if you choose to proceed, all consequences are
at your own risk.

Armel-RCTS@RCTS-APP: $ sudo chmod 666 /var/run/docker.sock
Password:
Armel-RCTS@RCTS-APP: $ docker exec -it rctsdash chown -R shiny:shiny .
```

Ensuite, la ligne de code suivante qui permet de donner les autorisations de lecture et écriture.

```
docker exec -it rctsdash chown -R shiny:shiny .
```

Après exécution, on devrait avoir le résultat ci-dessous :

```
OpenSSH SSH client
chown: changing ownership of './sys/module/nf_conntrack_ipv4/holders': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv4/initsize': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv4': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/notes/.note.gnu.build-id': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/notes': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/taint': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/initstate': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/coresize': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.init.text': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.smp_locks': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.data': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.text': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.rodata': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.strtab': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.symtab': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.gnu.linkonce.this_module': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.rodata.str1.1': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.rodata.str1.8': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.note.gnu.build-id': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.exit.text': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections/.data..read_mostly': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/sections': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/refcnt': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/uevent': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/holders': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6/initsize': Read-only file system
chown: changing ownership of './sys/module/nf_conntrack_ipv6': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/notes/.note.gnu.build-id': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/notes': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/taint': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/initstate': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/coresize': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.text': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.rodata': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.strtab': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.symtab': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.gnu.linkonce.this_module': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/.note.gnu.build-id': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/__ksymtab_strings': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections/__ksymtab': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/sections': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/refcnt': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/uevent': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/holders/btrfs': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/holders': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress/initsize': Read-only file system
chown: changing ownership of './sys/module/zstd_decompress': Read-only file system
chown: changing ownership of './sys/module': Read-only file system
chown: changing ownership of './sys': Read-only file system
Armel-RCT@RCTs-APP: $
```

2.3. Création d'un conteneur Docker

La création d'un conteneur est nécessaire uniquement lorsqu'on a besoin de modifier le code source de l'application à travers l'ajout de nouveaux packages. En effet le conteneur (qui exécute l'application) a été crée en compilant les packages nécessaires à l'application et au shiny serveur.

2.4. Redémarrage d'un conteneur Docker

Lorsque le serveur est éteint pour donner suite à une coupure de courant ou autres, il faut :

Rallumer physiquement le NAS à partir de la salle de serveur

Accéder au NAS à travers son interface (voir la section localisation des dossiers) et ensuite docker et dans conteneur mettre en route le conteneur rctsdash :

