Lâchez les rênes, laissez vous guider : Installation facilitée et sécurisée d'eLabFTW

Richard Ferrere, Jean-Marc Sibaud, Henri Valeins, Sandrine Sabatié, Karine Viaud, Michel Goillandeau, Philippe Hortolland

Lâchez les rênes, laissez-vous guider : installation facilitée et sécurisée d'eLabFTW

Richard Ferrere 1
CELIA - UMR5107
CNRS
43 Rue Pierre Noailles
33 405 Talence

Jean-Marc Sibaud 2
I2M - UMR 5295
CNRS
351 Cours de la Libération
33405 Talence Cedex

Henri Valeins 3
CRMSB - UMR5536
CNRS
146 Rue Léo Saignat
33 076 Bordeaux Cedex

Sandrine Sabatié 4
ETTIS - UR 1456
INRAE
50 Avenue de Verdun
33 610 Cestas

Karine Viaud 5
CRMSB - UMR5536 CNRS
146 Rue Léo Saignat
33 076 Bordeaux Cedex

Michel Goillandeau 6
IMN - UMR5293
CNRS
146 Rue Léo Saignat
33 076 Bordeaux Cedex

Philippe Hortolland 7
LP2N - UMR5298
CNRS
Rue François Mitterrand
33 400 Talence

Résumé

Le CNRS, établissement de recherche à caractère pluridisciplinaire, a lancé en 2023 une offre de service pour la mise en place d'un cahier de laboratoire électronique (CLE), basé sur le logiciel libre eLabFTW. Cette offre s'articule autour de deux solutions, une offre SaaS (Software as a Service) et une offre On Premise (en local ousur site). Dans cet article nous allons vous proposer un modèle d'accompagnement sur le déploiement et la configuration de eLabFTW dans un laboratoire ou une unité de recherche, en adéquation avec les consignes du CNRS. Nous détaillerons les étapes essentielles à suivre pour assurer une installation conforme aux exigences techniques et de sécurité en vigueur.

Mots-clefs

Docker, eLabFTW, cahier de laboratoire électronique, sécurité, démarche qualité, données, interopérabilité, traçabilité, expérience scientifique, partage, science ouverte, savoir-faire, logiciel libre.

1 Introduction et présentation

1.1 Contexte et objectifs

Élaboré par le Ministère de l'Enseignement Supérieur et de la Recherche, en collaboration avec l'Institut National de la Propriété Intellectuelle et en concertation avec les organismes de recherche publics, le cahier de laboratoire dit « national » permet à tous ceux qui réalisent des travaux de recherche (chercheurs, ingénieurs, techniciens, doctorants, stagiaires...) de consigner au jour le jour le détail de leurs travaux, de rendre compte du cheminement et de l'expérimentation scientifique, de l'idée à la conclusion. Le cahier de laboratoire est un élément essentiel à la traçabilité des résultats de recherche, éléments sensibles dans le cadre de la protection du patrimoine scientifique et technique du CNRS.

Toutefois le cahier de laboratoire « national » est un outil papier qui apparaît de moins en moins adapté aux pratiques de la recherche compte tenu de la nature numérique des données produites. Les apports du numérique sont en effet multiples en améliorant la traçabilité des recherches, la lutte contre la fraude et la gestion des données à travers par exemple, le partage de l'information avec un rattachement des données brutes, une recherche d'informations facilitée et une datation des expériences par horodatage.

Aussi, le CNRS, établissement de recherche à caractère pluridisciplinaire, a lancé en 2020 une réflexion sur la mise en place de Cahiers de Laboratoire Électroniques (CLE) suite aux besoins remontés par les agents en laboratoire, en alternative au cahier de laboratoire national papier.

L'objectif de cette réflexion, initiée conjointement par les Directions Générales Déléguées aux Ressources et à la Science, était de mettre en place des solutions électroniques adaptées aux pratiques de la recherche et à

l’ensemble des domaines scientifiques. Le projet devait prendre en compte la diversité des activités du CNRS tout en assurant traçabilité, sécurité, confidentialité et pérennité des résultats de recherche, et également tout en respectant les exigences liées à la Science Ouverte et à la protection du patrimoine scientifique et technique de l’établissement.

Il est ressorti de cette enquête que les cahiers de laboratoire électroniques permettent d’améliorer, en autres :

- la rédaction et la lisibilité des résultats de recherche ;
- les possibilités de valorisation, en cas de publications ou de dépôts de brevet avec la garantie de preuve de l’antériorité et de propriété intellectuelle ;
- la structuration et la recherche d’informations au travers de filtres et moteurs de recherche ;
- l’indexation de documents, l’interopérabilité et l’interfaçage avec divers outils ;
- la sécurité sur les systèmes de stockage des résultats de recherche ;
- les démarches qualité au travers d’une meilleure traçabilité et pérennité des protocoles et résultats de recherche ;
- l’organisation du travail : gestion des utilisateurs, accès distants, travail collaboratif...

Simultanément et en lien avec les directions fonctionnelles, ont été identifiées les caractéristiques indispensables auxquelles doit répondre un cahier de laboratoire conformément aux exigences de sécurité et de protection des données de l’établissement.

Ce travail a permis de définir un cadre clair qui tient compte des besoins des utilisateurs et des exigences de l’établissement.

C’est dans ce contexte que quatre outils logiciels ont été évalués afin de sélectionner le plus adapté aux critères utilisateurs et exigences établissement : la solution libre et open source eLabFTW a été retenue par le CNRS car elle permet de répondre complètement aux problématiques scientifiques, juridiques, de sécurité, de protection des données et organisationnelle.

Cette offre est déployée depuis le printemps 2023 et plus d’une centaine de laboratoires l’ont adoptée.

1.2 Traçabilité : logs de l’expérience

La traçabilité des logs dans un cahier de laboratoire électronique comme eLabFTW est essentielle pour garantir l’intégrité, la traçabilité et la conformité des données scientifiques.

Chaque action effectuée dans le système est enregistrée dans un journal, incluant des détails tels que l’identité de l’utilisateur, l’heure exacte et la nature de l’action comme la création, la modification ou la suppression d’une entrée.

Grâce à l’horodatage, à la fois programmé et à la demande, il est possible de consigner automatiquement des événements à intervalles réguliers ou d’horodater manuellement des expériences spécifiques. Ce mécanisme, reposant sur un tiers de confiance, permet d’établir de manière incontestable la date et l’heure de création ou de modification des résultats, renforçant ainsi leur valeur juridique. Cela permet non seulement d’identifier clairement le créateur de chaque expérience, mais aussi de "figer" cette expérience à un moment précis, permettant ainsi d’enregistrer toute modification ultérieure. Cette capacité à verrouiller et à authentifier les données garantit la vérifiabilité et l’imputabilité des résultats consignés.

1.3 Présentation de l’outil libre eLabFTW

Développé par Nicolas Carpi, travaillant auparavant à l’Institut Curie (Paris), eLabFTW s’est imposé comme une référence incontournable dans le domaine des cahiers de laboratoire électroniques. Ce logiciel libre, fruit d’une collaboration active au sein d’une communauté de chercheurs et de développeurs, ne cesse d’évoluer pour répondre aux besoins toujours plus complexes de la recherche moderne. Grâce à son architecture ouverte et modulaire, eLabFTW offre une grande flexibilité d’utilisation. Chaque laboratoire peut ainsi adapter l’outil à ses spécificités, en créant des modèles personnalisés pour la saisie des données, en intégrant de nouvelles fonctionnalités ou en développant des modules complémentaires. Cette

personnalisation permet d'optimiser l'organisation des données et de faciliter le travail quotidien des chercheurs. La force d'eLabFTW réside également dans sa communauté d'utilisateurs, qui contribue activement à son développement. Les chercheurs du monde entier partagent leurs expériences, leurs idées et leurs codes sources, créant ainsi un véritable écosystème collaboratif. Cette dynamique favorise l'innovation et garantit la pérennité du projet.

Les avantages d'eLabFTW sont de plusieurs natures :

- la centralisation des données : assembler toutes les informations liées à vos expériences (protocoles, résultats, observations) au sein d'une plateforme unique et sécurisée,
- la traçabilité et reproductibilité : assurer la traçabilité de vos travaux et faciliter la reproduction de vos expériences grâce à un historique détaillé des modifications,
- la collaboration facilitée : partager vos données avec vos collaborateurs et suivre en temps réel l'avancement de vos projets,
- l'accessibilité : accéder à vos données depuis n'importe quel appareil connecté à Internet, grâce à l'interface web intuitive de eLabFTW,
- le multilingue : disponible en plusieurs langues, dont le français, eLabFTW s'adapte à un public international,
- la sécurité : plusieurs audits de sécurité ont démontrés la robustesse de son code.

1.4 Présentation des différents modèles d'installation (exemple du CNRS)

L'offre de service Cahier de Laboratoire Electronique du CNRS s'articule autour de deux solutions :

- une offre SaaS (Software as a Service),
- une offre On Premise (en local ou sur site).

L'offre SaaS est une solution clé en main pour laquelle l'outil eLabFTW est accessible en mode service Web. En tant qu'utilisateur, vous accédez au logiciel grâce à une connexion Internet. L'installation et l'exploitation sont réalisées par le prestataire (EasterEggs) dans un cadre technique et de sécurité défini par le CNRS sur un hébergement externe qualifié SecNumCloud (le plus haut niveau de qualification et de sécurité ANSSI). Cette offre est réservée aux unités ne disposant pas des ressources techniques pour réaliser une installation sur site.

L'offre On Premise est quant à elle réservée aux unités en capacité de réaliser l'installation et l'exploitation de manière autonome. Le support technique à l'installation par le prestataire reste toutefois acquis sur simple demande. C'est sur cette offre que nous allons vous proposer un modèle d'accompagnement sur le déploiement et la configuration de eLabFTW dans un laboratoire, en adéquation avec les consignes du CNRS. Nous détaillerons les étapes essentielles à suivre pour assurer une installation conforme aux exigences techniques et de sécurité en vigueur. Un accent particulier sera mis sur la mise en œuvre de l'authentification forte avec Janus+, basée sur les standards FIDO2. Cette mesure renforce significativement la sécurité des accès, en exigeant de l'utilisateur plusieurs facteurs d'authentification (mot de passe et clé de sécurité physique) et l'imputabilité des actions. De plus, nous aborderons la question de l'horodatage qualifié, garantissant l'intégrité et la traçabilité des données de recherche. Ce mécanisme, reposant sur un tiers de confiance, permet d'établir de manière incontestable la date et l'heure de création ou de modification des résultats, renforçant ainsi leur valeur juridique.

2 Méthodologie : modèle d'installation sécurisé On Premise

2.1 Architecture générale du modèle avec une installation On Premise

Le groupe de travail est parti du schéma de référence du CNRS afin de le déployer selon les exigences demandées.

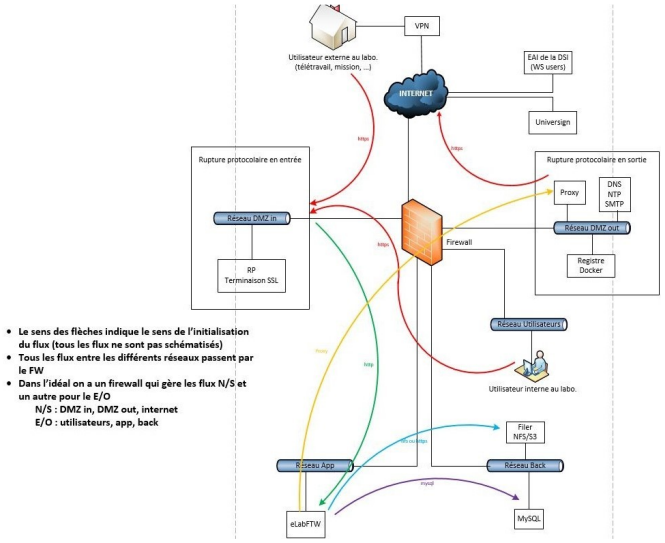

Figure 1 – Architecture de référence du CNRS

2.2 Recommandations techniques : chiffrement et segmentation

Les exigences d'installation minimales du CNRS sont regroupées dans le tableau ci-après :

Composant	Exigence
Pas d'accès depuis l'extérieur : accès via VPN pour utilisateurs en télétravail	Obligatoire
2 VM : une Docker/eLabFTW, une base de donnée dédiée (avec ou sans Docker)	Obligatoire
Pare-feu d'infrastructure capable de segmenter les VM convenablement	Obligatoire
Adresse IP externe pour appel Web Service dédiée à l'unité	Obligatoire
Chiffrement des disques durs	Obligatoire
Chiffrement des sauvegardes	Obligatoire

Figure 2 – Exigences d'installation par le CNRS

Pour cela, pour un laboratoire ou une Unité de recherche (sans reverse proxy), la matrice des flux réseaux peut se résumer par le tableau ci-après :

Num	Adresse source	Port source	Adresse destination	Port destination	Action	Description
1	*client VPN*	*any*	*IP pub FW*	*443 TCP*	*allow*	*tout utilisateur VPN*
2	*IP pub FW*	*any*	*IP privé eLabFTW*	*443 TCP*	*allow*	*accès eLab*
3	*IP privé eLabFTW*	*any*	*IP MySQL*	*3306 TCP*	*allow*	*accès MySQL*
4	*IP privé eLabFTW*	*any*	*IP NFS/CIFS*	*2049 TCP/UDP*	*allow*	*accès Filer*
5	*1XX.1XX.1XX.0/24*	*any*	*IP MySQL*	*2049 TCP/UDP*	*deny*	
6	*1XX.1XX.1XX.0/24*	*any*	*IP NFS/CIFS*	*2049 TCP/UDP*	*deny*	
7	*elab/nfs/sql*	*514 UDP*	*IP Syslog*	*514 UDP*	*allow*	*collecte par syslog*
8	*any*	*any*	*any*	*any*	*deny*	

Table 1 – Exemple de tableau des règles de filtrage de l'infrastructure réseau

2.3 Les différents composants du modèle : Docker, Docker Compose, certificats, MySQL, eLabFTW, Filer, Reverse proxy

· Dans notre cas, nous installons Docker qui est une technologie de virtualisation permettant de créer et de gérer les conteneurs eLabFTW et MySQL, des zones d'exécution isolées et indépendantes du système d'exploitation :
Linux Ubuntu (22.04). Nous installons les outils *Docker* et *Docker Compose*, des fichiers texte de configuration permettant la création des applications eLabFTW ou MySQL.

```
sudo apt install docker-compose
systemctl status docker
```

· Pour ce qui est de *Docker Compose*, le fichier docker-compose.yml est le fichier texte de configuration au format YAML décrivant l'arborescence de données avec les paramètres ou directives des conteneurs des applications à installer. La dernière version du fichier docker-compose.yml est téléchargeable à l'adresse suivante : https://get.elabftw.net/?config.

Avant de passer aux étapes d'installation, nous donnons quelques explications sur le fichier (.yml) qui servira à installer les instances et conteneurs Docker (voir annexe 5.1).

Ce fichier est déclaratif et documenté permettant de configurer et d'orchestrer les services eLabFTW et MySQL sur 2 machines séparées (selon les consignes du CNRS). Nous découpons le fichier téléchargé en 2 afin d'avoir l'un des fichiers de configuration sur la machine eLabFTW et l'autre sur la machine MySQL.

· L'installation nécessite d'utiliser des certificats SSL qui sont vitaux pour nos systèmes d'information (SI) et notamment pour l'infrastructure CLE. Cela permet une communication chiffrée Web entre eLabFTW et les utilisateurs (HTTP over TLS), puis entre eLabFTW et la base de données interne MySQL.

Dans notre cas d'usage, il faut générer des certificats auto-signés ou utiliser un certificat public reconnu par une autorité de certification reconnue (Université ou CNRS) de type DigiCert ou Sectigo.

Depuis la machine MySQL,

- **1ère étape** : générer la clé Certificate Authority (CA) et le certificat serveur

```
# pour créer la clé CA et générer le fichier ca-cert.pem
openssl genrsa 2048 > ca-key.pem
# pour créer le certificat avec la clé précédente et générer le fichier clé CA nommé ca-key.pem
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.pem
# pour le serveur, on crée la clé et le certificat
openssl req -newkey rsa:2048 -days 365000 -nodes -keyout server-key.pem -out server-req.pem
# pour convertir la clé en RSA
openssl rsa -in server-key.pem -out server-key.pem
# pour signer
openssl x509 -req -in server-req.pem -days 365000 -CA ca-cert.pem -CAkey ca-key.pem -set_serial
01 -out server-cert.pem
```

- **2ème étape** : générer de la même manière le certificat pour le client

```
openssl req -newkey rsa:2048 -days 365000 -nodes -keyout client-key.pem -out client-req.pem
openssl rsa -in client-key.pem -out client-key.pem
openssl x509 -req -in client-req.pem -days 365000 -CA ca-cert.pem -CAkey ca-key.pem -set serial 01
-out client-cert.pem
# pour vérifier les certficats
openssl verify -CAfile ca-cert.pem server-cert.pem client-cert.pem
openssl x509 -text -in ca-cert.pem
openssl x509 -text -in server-cert.pem
openssl x509 -text -in client-cert.pem
```

- **3ème étape** : positionner les droits sur les fichiers certificats avant utilisation

```
# changement du propriétaire
sudo chown 999 server-key.pem
# copier les certificats sur la machine eLabFTW
scp ca-cert.pem server-cert.pem...(vers la machine eLabFTW)
```

- **4ème étape** : se rendre sur la machine eLabFTW

-

- **5ème étape** : adapter les droits depuis la machine eLabFTW

```
chown root *.pem
chmod 400 ca-cert.pem (que pour root)
chmod 444 server-cert.pem
chown systemd+ server-cert.pem
```

- Le service *Filer* est un composant recommandé pour stocker de façon sécurisée les fichiers sur le réseau via les protocoles NFS ou CIFS. Il pourra être installé à part dans une machine virtuelle afin d'être disponible pour la base de données.

- Par ailleurs, un reverse proxy pourrait être installé. C'est une barrière de protection qui permet à des utilisateurs externes d'accéder à la ressource CLE sur un réseau interne de façon sécurisée.

Cas d'utilisation et exemple pratique par l'installation (quelques étapes ci-après).

Nous allons présenter le déroulement en quelques étapes d'un cas d'installation et d'utilisation en mode sécurisé et donc en local au sein de l'unité de recherche.

- Ajouter le compte *user* dans le groupe *docker* : **sudo usermod -aG docker user**

- Créer les répertoires pour les volumes persistants, pour les certificats et les fichiers téléchargés de eLabFTW. Dans notre exemple : **mkdir /opt/elabftw/cert, /opt/elabftw/web**

- Sur la machine eLabFTW, voici l'exemple du fichier *docker-compose.yml* pour l'installation de l'instance eLabFTW version 5.0.4 :

```
version: '3'
services:
  web:
    image: elabftw/elabimg:5.0.4
    restart: always
    container_name: elabftw
    security_opt:
        - no-new-privileges:true
    cap_drop:
        - ALL
    cap_add:
        - CHOWN
        - SETGID
        - SETUID
        - FOWNER
        - DAC_OVERRIDE
    environment:
        ########################
        # MYSQL CONFIGURATION #
        ########################
        # default value: mysql
        - DB_HOST=nom FQDN ou adresse IP du conteneur MySQL
        # default value: 3306
        - DB_PORT=3306
        # name of the MySQL database
        # default value: elabftw
        - DB_NAME=elabftw
        # default value: elabftw
        - DB_USER=elabftw
        # MySQL password; a random password has been generated for you but feel free to change it
            if needed
        # default value: generated randomly if you get the config from get.elabftw.net
        - DB_PASSWORD=EbufCpyHvSyddZTg3vp0UdWWzwpaURY
        # Mysql Cert path: you only need this if you connect to a mysql server with
            SSL/TLS
        # Use a volume that points to /mysql-cert in the container
        - DB_CERT_PATH=/mysql-cert/server-cert.pem
        ######################
        # PHP CONFIGURATION #
        ######################
        - PHP_TIMEZONE=Europe/Paris
        - TZ=Europe/Paris
        ##########################
        # eLabFTW CONFIGURATION #
        ##########################
        # The secret key is used for encrypting the SMTP and Timestamping passwords
```

```
        # default value: generated randomly if you get the config from get.elabftw.net
        - SECRET_KEY=def00000f3f87046063f81f7ecba3d6ace3b6a92a92730eb1f2809a9edd561bca3d
        1f53d420919ea83b5e576bc ee40b98066b11e207edf5fd2a9d2d14013e3fe86f53c7d
        # example value: https://elab.uni-delta.fr
        # example value: https://elab.uni-delta.fr:3148
        - SITE_URL=https://fqdn.cle.cnrs.fr
        ########################
        # NGINX CONFIGURATION #
        ########################
        # change to your server name in nginx config
        # default value: localhost
        # example value: elab.uni.edu
        - SERVER_NAME=fqdn.cle.cnrs.fr
        # default value: false -
        DISABLE_HTTPS=false
        # default value: false
        - ENABLE_LETSENCRYPT=false
  ports:
        - '443:443'
    volumes:
        - /opt/elabftw/web:/elabftw/uploads
        - /opt/elabftw/cert:/mysql-cert
    networks:
        - elabftw-net
networks:
   elabftw-net:
```

- Créer et lancer le conteneur eLabFTW : **docker-compose up -d**

- Sur la machine MySQL, voici l'exemple du fichier *docker-compose.yml* pour l'installation de l'instance MySQL version 8.0 :

```
version: '3'
services:
   mysql:
     image: mysql:8.0
     restart: always
     container_name: mysql
     cap_drop:
         - AUDIT_WRITE
         - MKNOD
         - SYS_CHROOT
         - SETFCAP
         - NET_RAW
     cap_add:
         - SYS_NICE
     environment:
        # need to change
        - MYSQL_ROOT_PASSWORD=pvwm7hvGPfLaq94oM8hB1lLBDF8IoUv
        # no need to change
        - MYSQL_DATABASE=elabftw
        # no need to change
             -   MYSQL_USER=elabftw
        #  need to change IMPORTANT: this should be the same password as DB_PASSWORD from the
            eLabFTW container
        - MYSQL_PASSWORD=GX433X0c3OIIrru9Sox454hMeTCQpUZ
        - TZ=Europe/Paris
     volumes:
        # host:container
        - /opt/elabftw/mysql:/var/lib/mysql
        - /opt/elabftw/cert:/etc/certs
     command: [ "mysqld",
              "--character-set-server=utf8mb4",
              "--collation-server=utf8mb4_unicode_ci",
              "--bind-address=0.0.0.0",
              "--require_secure_transport=ON",
              "--ssl-ca=/etc/certs/ca-cert.pem",
              "--ssl-cert=/etc/certs/server-cert.pem",
              "--ssl-key=/etc/certs/server-key.pem",
              "--default_authentication_plugin=mysql_native_password" ]
     # The MySQL container exposes 3306/33060. Though it does not make an operational difference,
     # make sure to document your usage here.
     expose:
         - '3306'
     ports:
```

```
        - '3306:3306'
    networks:
        - elabftw-net
# the internal elabftw network
networks:
   elabftw-net:
```

- Installer les certificats (CA, serveur et client) avant de lancer l'installation de MySQL. Pour cela, il faudra exécuter les commandes du chapitre 2.3 pour la génération des certificats.

- Ajouter dans le fichier *docker-compose.yml* (sur la machine MySQL) les informations concernant les certificats, et notamment renseigner les volumes.

- Créer et démarrer le conteneur MySQL avec une instance Docker : **docker-compose up -d**

- Aller sur la machine eLabFTW ;

- Initialiser la base de données eLabFTW : **docker exec -it elabftw bin/init db :install**

- Lancer l'URL (https ://fqdn.cle.cnrs.fr) de eLabFTW définie dans le fichier *docker-compose* dans un navigateur Internet afin de se connecter à l'application et de créer l'utilisateur SYSADMIN.

2.4 Exemple du service d'authentification du CNRS : Janus+ et clé FIDO2

Une autre étape est de passer par une authentification sécurisée, mais pour cela il faut disposer d'une connexion VPN. L'application eLabFTW permet une authentification depuis un annuaire LDAP. Dans notre cas et selon les recommandations du CNRS, cela se fait par une authentification renforcée à 2 facteurs : le service JANUS+ et une clé USB de type FIDO2.

Le principe consiste à enrôler les clés de sécurité USB (rôle des CSSI d'Unité) pour chacun des utilisateurs du CLE. L'utilisateur reçoit un mail avec un lien cliquable valide 30 minutes ; ensuite, il peut accéder très rapidement à l'application de façon sécurisée. Cela permet de garantir un haut niveau de confiance sur l'authentification des utilisateurs qu'ils soient internes à l'Unité ou externes (UHPI), et donc sur l'imputabilité des résultats de recherche. Cela se base sur la technologie FIDO2 qui est maintenant intégrée nativement aux navigateurs Web.

3 Actions importantes : sauvegarde, restauration, mises à jour et supervision

Présentation des actions d'exploitation les plus importantes.

3.1 Exemple de sauvegarde sécurisée

Une sauvegarde est efficace lorsqu'elle est automatique, fréquente (quotidienne) et stockée à plusieurs endroits.

Les pré-requis :

- Borgbackup installé et configuré ;
- Disposer du volume persistant et/ou d'un montage sécurisé (NFS ou CIFS).

La méthode de sauvegarde recommandée :

Il est possible de faire une archive compressée des répertoires ou des exports des conteneurs *Docker* et de les copier sur un autre volume.

Nous préconisons cependant l'utilitaire *mysqldump* pour réaliser la sauvegarde des bases et d'utiliser *Borg* pour l'envoyer sur un *Filer* ou une autre machine virtuelle.

Par exemple, le script ci-après permet la sauvegarde avec *mysqldump* et le stockage dans le répertoire */backups_mysql* via la commande *Docker*, avec effacement des fichiers de sauvegarde de plus de 60 jours.

```
#!/bin/bash
# run by crontab
# m h dom mon dow      command
# 23 11 * * /home/USER/Docker/elabftw/backup_mysql.sh
```

```
#Contenu du fichier backup_mysql.sh:
#!/bin/bash
cd /home/USER/Docker/elabftw
# backup all database from mysql elabftw server
docker exec mysql mysqldump --defaults-extra-file=/credentials/mysqldump.cnf
--all-databases > backups_mysql/all-databases-$(date +%Y-%m-%d_%H:%M).sql
# delete files older than 2 months
find backups_mysql/all-databases* -mtime +60 -exec rm {} \;
```

Répéter cette opération sur la machine eLabFTW pour sauvegarder les fichiers téléchargés (/opt/elabftw/web).

3.2 Méthode de restauration

Nous proposons une méthode de restauration de eLabFTW et de sa base de données MySQL en cas de problème ou de migration. Cela suppose de disposer préalablement d'une sauvegarde de la base de données. Cette méthode est accessible à partir de ce lien https://doc.elabftw.net/backup.html#how-to-restore-a-backup.

- Pré-requis : conteneur *MySQL* démarré, disposer du fichier de sauvegarde *mysql_dump-YYYY-MM-DD.sql*, disposer du répertoire contenant les images et documents des expériences.

- Étapes à réaliser dans un terminal *shell Linux* sur la machine *MySQL* :

```
# Copier la base actuelle dans le conteneur MySQL
docker cp mysql_dump-YYYY-MM-DD.sql mysql:/tmp
# Lancer un shell dans le conteneur MySQL
docker exec -it mysql bash
# Se connecter au prompt MySQL
mysql -uroot -p$MYSQL_ROOT_PASSWORD
# Supprimer la base de données
mysql> drop database elabftw;
# En créer une nouvelle
mysql> create database elabftw character set utf8mb4 collate utf8mb4_0900_ai_ci;
# La sélectionner
mysql> use elabftw;
# Être certain d'importer au format UTF8 (ne pas importer au format latin1)
mysql> set names utf8mb4;
# Importer la sauvegarde
mysql> source mysql_dump-YYYY-MM-DD.sql;
mysql> exit;
# Quitter le conteneur MySQL : exit
```

- Sur la machine eLabFTW

```
# Mettre à jour la base de données
docker exec -ti elabftw bin/console db:update
# Vérifier que tout est correct
docker exec -ti elabftw bin/console db:check
```

3.3 Mises à jour : elabFTW, MySQL et CLEbot

Avant toute mise à jour, il convient de vérifier que la sauvegarde de la base de données et des fichiers téléchargés a bien été réalisée.

Il faut également penser à lire les notes des changements de version pour s'assurer des interventions manuelles éventuellement à réaliser.

Ci-dessous, un exemple pour la mise à jour d'une version 5.0.4 vers la version 5.1.0

Sur la machine exécutant eLaFTW, télécharger la nouvelle image eLabfTW avec docker pull elabimg :5.1.0

Modifier le fichier *docker-compose.yml* en changeant le numéro de version d'eLabFTW

```
services:
     web:
          image: elabftw/elabimg:5.1.0
     ...
```

Puis arrêter le service d'eLabFTW et effacer le conteneur : **docker-compose down**

Maintenant on peut relancer eLabFTW : **docker-compose up -d**

Il faut vérifier que la base de données *MySQL* est bien à jour aussi :

```
docker exec -it elabftw bin/console db:check
Database check
Current version: 144
Required version: 144
No upgrade required.
```

Dans cet exemple aucune mise à jour n'est nécessaire.

Par contre si on a ce message :

```
Database check
Current version: 144 Required version: 149
An upgrade is required.
```

Dans ce cas il faut exécuter la commande suivante :

```
docker exec -it elabftw bin/console db:update
Database check
Current version: 145 Required version: 149
An upgrade is required.
Database update starting Executing START TRANSACTION
...
All done.
```

CLEbot permet la configuration et la maintenance d'une instance eLabFTW dans le cadre de l'offre CNRS. Cela configure l'authentification renforcée avec Janus+ via l'annuaire du CNRS ainsi que les paramètres de sécurité obligatoires exigés par le CNRS de votre instance eLabFTW.

3.4 La supervision : logs, messagerie

Après l'installation des instances et conteneurs eLabFTW et MySQL, il sera nécessaire de saisir un compte et une adresse de messagerie. Cela permettra de recevoir également les notifications par mail.

Afin de répondre aux exigences réglementaires, il sera essentiel de consulter et gérer les logs (journaux d'activité) car cela garantit la traçabilité et la sécurité des données. L'application eLabFTW enregistre l'activité comme l'authentification, les modifications sur les fichiers, les exportations… Seuls les administrateurs ont accès aux logs.

Les logs et l'activité pourront être consultés soit depuis l'application, soit depuis un terminal de l'installation de eLabFTW (commande **docker logs elabftw**). Cette pratique de gestion des logs est connue des ASR qui pourront l'intégrer dans leur système d'information au titre d'un archivage et/ou d'une automatisation.

4 Conclusion

À travers cet article, nous avons présenté les principaux aspects techniques indispensables à la mise en place d'un cahier de laboratoire électronique (CLE). Cet outil est appelé à s'imposer dans nos environnements de recherche scientifique et concerne un grand nombre d'unités affiliées au CNRS, à l'INRAE, ainsi qu'à d'autres structures nécessitant la sécurisation de leurs données scientifiques en vue de leur publication.

Cette réalisation technique a été également une véritable aventure humaine pour l'équipe d'informaticiens bordelais qui s'est constituée au sein d'un groupe de travail RAISIN et du réseau national RESINFO. Ce projet a demandé un investissement conséquent de la part de chacun avec de nombreuses réunions, des phases de tests, des démos et de la rédaction.

Notre groupe s'était fixé pour objectif de démystifier et de synthétiser ce sujet en regroupant l'ensemble des informations et sources disponibles afin de faciliter et de sécuriser l'installation du produit eLabFTW dans nos laboratoires en mode On Premise. Ce mode d'installation sécurisé présente un enjeu majeur et stratégique car les Unités ou laboratoires dits sensibles (ZRR) pourront conserver et maîtriser pleinement leurs données sur site, tout en gardant le contrôle total de l'outil.

Ce projet s'accompagne d'une documentation détaillée, incluant des liens vers diverses références. Elle est accessible à tous sur le site Internet dédié aux documentations des groupes de travail RAISIN. Nous avons pour objectif de mettre à jour cette documentation régulièrement afin de la rendre toujours plus utile à la communauté. Dans une vision plus large, cette solution pourrait être présentée lors d'une manifestation nationale, telle que les Journées Systèmes JOSY ou JTECH. Cela permettrait de réaliser une mise en pratique de ce modèle d'installation pour des administrateurs systèmes. D'autres défis peuvent aussi être relevés avec l'intégration automatisée de cette solution dans nos systèmes d'information. Il est en effet tout à fait possible d'utiliser des technologies de déploiement et d'automatisation avec des scripts ou avec ANSIBLE pour intégrer eLabFTW dans nos laboratoires (comme par exemple pour la génération des certificats et la création des machines virtuelles sous PROXMOX). Un tel projet serait aussi une opportunité pour créer tout un écosystème permettant de sensibiliser à l'importance croissante de la sécurité numérique et de la gestion des données.

Nous espérons que ce modèle d'installation, à la fois simple et facile à mettre en œuvre, conforme aux exigences fixées par nos tutelles, facilitera l'intégration de la solution eLabFTW dans vos laboratoires ou unités de recherche.

Annexes

Commandes de base de Docker

□ Commandes avec docker-compose

```
docker-compose up -d (créer le ou les conteneur(s) figurant dans le fichier docker-compse.yml, lancer les conteneurs sans l'affichage dans le terminal) docker-compose down (arrêter et supprimer les conteneurs existants, ou supprimer les conteneurs ou volumes du répertoire en cours)
```

□ Commandes pour les images et les volumes

```
docker pull nom_image (télécharger et nettoyer les images) docker
image ls -a (ister les images locales)
docker container prune (supprimer tous les conteneurs arrêtés) docker rmi
imgae_id (supprimer une image à partir de son ID)
docker image prune -a (supprimer les images non liées à des contebeurs en cours d'exécution) docker-compose down (arrêter le serveur local avant suppression du volume) docker volume create/rm nom_volume (créer/supprimer un volume)
docker run -v nom_volume:/chemin/dans/conteneur nom_image (monter un volume lors de la création d'un conteneur)
docker volume prune (supprimer des volumes non utilisés ou connectés à des conteneurs) docker system prune -a (supprimer tous les objets inutilisés comme les images, volumes, conteneurs, réseaux)
```

□ Commandes pour gérer les conteneurs ou les instances d'images docker run -d --name

```
nom_conteneur nom_image (créer et démarrer un conteneur) docker start/stop
nom_conteneur (démarrer/arrêter un conteneur)
docker ps, docker ps -a (lister les conteneurs en cours d'exécution, ou tous les conteneurs) docker rm nom_conteneur (supprimer un conteneur)
docker exec -it nom_conteneur /bin/bash (accéder à un conteneur en cours d'exécution via un terminal)
docker exec -it elabftw_cnrs bin/console db:check (faire un check depuis le terminal console du conteneur)
docker exec -it elabftw_cnrs bin/console db:install (faire l'installation depuis le terminal console du conteneur)
docker exec -it mysql_cnrs_ssl bash (entrer dans un bash du conteneur)
```

□ Commandes pour gérer les logs

```
docker logs nom_conteneur (afficher les logs d’un conteneur) docker logs
-f nom_conteneur (suivre les logs en temps réel)
docker inspect nom_conteneur_ou_image (inspecter un conteneur ou une image)
```

Description du fichier docker-compose.yml

Plusieurs blocs de directives peuvent être trouvés et les lignes commençant par # sont des commentaires :

- **version** : la version de la syntaxe Docker Compose utilisée. Ici, c’est la version 3;
- **services** : les services web et mysql seront installés et exécutés;
  - **image** pour spécifier l’image de la version la plus récente de elabftw à utiliser pour le conteneur;
  - **container_name** pour le nom du conteneur;
  - **ports** pour le mappage des ports entre l’hôte et le conteneur (par exemple, dans notre cas, le port 443 de l’hôte sera mappé au port 443 du conteneur afin d’utiliser des connexions Internet sécurisées par le protocole SSL/TLS);
  - **volumes** pour monter les volumes entre le système hôte et le conteneur afin de bénéficier de la persistance des fichiers (configuration, uploads);
  - **depends_on** pour spécifier, par exemple que le service elabftw dépend du service mysql (Docker Compose va s’assurer que la base de données est bien démarrée avant elabftw).
- **networks** : le réseau elabftw interne;
- **volumes** : le volume pour la persistance des données même en cas de recréation des conteneurs.

Références

- Site de eLabFTW : https://www.elabftw.net/
- Documentation de eLabFTW : https://doc.elabftw.net/
- Issues/problèmes et résolutions sur eLabFTW : https://github.com/elabftw/elabftw/issues/
- LDAP authentication : https://doc.elabftw.net/ldap.html
- Site du Github et code source : https://github.com/elabftw/elabftw
- Le guide du réseau Qualité en Recherche : https://qualite-en-recherche.cnrs.fr/guide/
- CNRS intranet - Architecture technique : https://confluence.cnrs.fr/confluence/pages/viewpage.action?spaceKey=CLE&title=1-+Exigences+d%27architecture+technique
- CNRS intranet – Janus+ : https://confluence.cnrs.fr/confluence/pages/viewpage.action?pageId=112002870
- Offres du CNRS (espace unités, démo, forum) : https://extra.core-cloud.net/projets/Cahiers_laboratoire_electroniques/SitePages/Accueil.aspx
- Réseau RAISIN : https://raisin.resinfo.org/