

<b>DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE</b>	<b>N° réalisation : 02</b>
--	----------------------------

Nom, prénom : <b>PROVENZANO Nina</b>	N° candidat :
--------------------------------------	---------------

Épreuve ponctuelle <input type="checkbox"/> Contrôle en cours de formation <input checked="" type="checkbox"/>	Date : <b>03 / 04 / 2026</b>
--	------------------------------

**Organisation support de la réalisation professionnelle**

Le port de Cherbourg (ferries transmanche, fret, croisières, réparation navale) est géré par une SEM locale et dispose d'un SI sous la responsabilité du DSI (M. Richard). Les données sont stockées via des répertoires partagés par service (droits au niveau du système de fichiers), sans redondance. Le cahier des charges demande de garantir le stockage sécurisé, la résilience d'accès aux données et un plan de sauvegarde testé. Cette réalisation met en place Wiki.js comme espace de référence par service, avec authentification LDAP, journalisation, automatisation documentaire via API Wiki.js et PCA basé sur sauvegardes Restic avec une procédure de reprise.

**Intitulé de la réalisation professionnelle**

Assurer la continuité d'accès aux données Wiki.js : PCA, LDAP, journalisation et automatisation via l'API.

**Période de réalisation :** 01/09/2025 – 01/05/2026      **Lieu :** Lycée Saint-Exupéry, Saint Raphaël

**Modalité :**       Seul(e)                       En équipe

**Compétences travaillées**

- Concevoir une solution d'infrastructure réseau
- Installer, tester et déployer une solution d'infrastructure réseau
- Déployer et administrer un service informatique
- Exploiter, dépanner et superviser une solution d'infrastructure réseau

**Conditions de réalisation (ressources fournies, résultats attendus)**

**Ressources fournies**

- Cahier des charges du Port de Cherbourg (stockage/sauvegarde, parc, incidents)
- 2 VM Linux (Wiki.js production + stockage/secours) en réseau interne LAN
- Active Directory/LDAP (gestion des comptes et groupes) + répertoires partagés existants (serveur de fichiers)
- Docker Compose avec Restic (sauvegardes chiffrées/versionnées) + journalisation et rotation des logs (exploitation)

**Résultats attendus**

- Wiki.js disponible, espaces par service, accès via LDAP/AD
- Journalisation des accès et des opérations (logs)
- Sauvegardes Restic chiffrées & versionnées (7J/4S/6M) + test de restauration
- Automatisation via API Wiki.js : création de pages modèles, mise à jour d'un index, publication d'un état
- PCA/Sauvegardes, liens vers répertoires partagés

**Description des ressources documentaires, matérielles et logicielles utilisées**

Ressources documentaires : documentation Wiki.js, Docker Compose, Restic, Linux.  
Ressources matérielles : 2 VM (VM principale : serveur Wiki.js, VM secondaire : serveur stockage/secours), réseau local.  
Ressources logicielles : Linux, Docker/Compose (Wiki.js), Restic (backup/rotation), systemd (planification), script interne utilisant l'API Wiki.js.

**Modalités d'accès aux productions et à leur documentation**

<https://nina-provenzano.fr/RP2/>

**Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs**

Étapes de mise en œuvre :

- 1) Mise en place de 2 VM Linux (production + stockage) et définition des flux (accès web, sauvegardes).
- 2) Déploiement de Wiki.js via Docker Compose (volumes persistants) et validation d'accès.
- 3) Authentification LDAP/AD : groupes par service, droits d'accès aux espaces Wiki.
- 4) Journalisation : logs d'accès HTTP et suivi des opérations d'administration (rotation des logs).
- 5) Sauvegardes Restic chiffrées & versionnées vers VM2 : planification, rétention 7J/4S/6M + purge.
- 6) Vérifications : Restic, contrôle des logs et suivi des exécutions.
- 7) API Wiki.js (automatisation documentaire) via script interne :
  - création d'un pack de pages modèles par service (Accueil, Procédures, Accès, Incidents, Sauvegardes),
  - mise à jour automatique d'un index des procédures (liste + liens),
  - publication d'une page PCA/Sauvegardes (dernière exécution, nb snapshots, statut),
  - insertion/maintenance des liens vers répertoires partagés et groupes LDAP associés.
- 8) Procédure de reprise : restauration Restic redémarrage de Wiki.js et contrôles (accès, données).

Validation : test de restauration (données + configuration), contrôle de disponibilité Wiki.js, test d'authentification LDAP et vérification des logs. Les productions sont versionnées et documentées pour faciliter l'exploitation et l'évolution.

