

|      |                                          |             |      |
|------|------------------------------------------|-------------|------|
| LOGO | <b>BTS SIO</b>                           |             | LOGO |
|      | Services Informatiques aux Organisations |             |      |
|      | <b>Option</b>                            | <b>SISR</b> |      |
|      | <b>Session</b>                           | <b>2026</b> |      |

|  |                                                                                                                                                  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------|--|
|  | Activité professionnelle N°1 — Fiabilisation et mise à disposition d'une plateforme interne de laboratoires de cybersécurité (conteneurs Docker) |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------|--|

| NATURE DE L'ACTIVITE       |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contexte</b>            | Stage de 1re année chez Caplogy & Novatiel (Vélizy-Villacoublay), organisme de formation dans le numérique. L'entreprise exploite une plateforme interne de laboratoires pédagogiques en cybersécurité composée de 26 images de conteneurs Docker (SIEM Splunk/Wazuh/ELK, pare-feu pfSense, Kali, OSINT, cryptographie). Plusieurs images ne se construisaient plus ou ne démarraient plus, ce qui bloquait leur mise à disposition aux apprenants. |
| <b>Objectifs</b>           | Tester, diagnostiquer et réparer les 26 images, valider leur fonctionnement de bout en bout (construction, exécution, déploiement) pour les rendre de nouveau disponibles en formation, puis documenter la procédure pour l'équipe.                                                                                                                                                                                                                 |
| <b>Lieu de réalisation</b> | Caplogy & Novatiel — Vélizy-Villacoublay (78).                                                                                                                                                                                                                                                                                                                                                                                                      |

| SOLUTIONS ENVISAGEABLES                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1) Corriger les fichiers Dockerfile au cas par cas et reconstruire chaque image.</p> <p>2) Repartir d'images de base plus récentes pour toute la plateforme.</p> <p>3) Figurer des images déjà construites sans corriger les sources.</p> <p>La solution 1 a été retenue : elle traite la cause des pannes, conserve le contenu pédagogique existant et reste reproductible, contrairement au figement d'images (3, non maintenable) et au changement global d'image de base (2, trop coûteux et risqué pour le calendrier).</p> |

| DESCRIPTION DE LA SOLUTION RETENUE |                                                                                                                                                                                                                                                               |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Conditions initiales</b>        | 26 images Docker dont une majorité en échec de construction (clés GPG de dépôts expirées, variable HOME incorrecte, module Python retiré en 3.13, trousseau gnome-keyring inutilisable sous Kubernetes) ou d'exécution ; aucune procédure de test formalisée. |
| <b>Conditions finales</b>          | 26 images sur 26 reconstruites, testées (construction, exécution, déploiement Kubernetes) et livrées à l'équipe pour publication ; procédure de test reproductible et guide de dépannage rédigés.                                                             |
| <b>Outils utilisés</b>             | Docker / BuildKit, Kubernetes (cluster de test), registre d'images local, kubectl (port-                                                                                                                                                                      |

|  |                      |
|--|----------------------|
|  | forward), Git, Bash. |
|--|----------------------|

| CONDITIONS DE REALISATION |                                                                                                                                                                                                                      |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Matériels</b>          | Poste de travail ; cluster Kubernetes de test (Docker Desktop).                                                                                                                                                      |
| <b>Logiciels</b>          | Docker Desktop, Kubernetes, registre Docker local, Git, VS Code ; images de base Kasm (Ubuntu, Kali, Parrot).                                                                                                        |
| <b>Durée</b>              | Environ 2 semaines, au sein d'un stage de 8 semaines (juin – juillet 2026).                                                                                                                                          |
| <b>Contraintes</b>        | Pannes systémiques communes à plusieurs images à isoler, temps de construction longs, différences de comportement entre exécution Docker simple et déploiement Kubernetes, livrables attendus avant la fin du stage. |

| COMPETENCES MISES EN OEUVRE POUR CETTE ACTIVITE PROFESSIONNELLE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <p>Gérer le patrimoine informatique : recenser et identifier les ressources numériques (images de conteneurs), vérifier les conditions de continuité d'un service.</p> <p>Répondre aux incidents et aux demandes d'assistance et d'évolution : traiter les demandes concernant les services système, diagnostiquer et corriger les défaillances.</p> <p>Mettre à disposition des utilisateurs un service informatique : réaliser les tests d'intégration et d'acceptation, déployer le service (Kubernetes), préparer sa mise à disposition aux apprenants.</p> <p>Travailler en mode projet : planifier le traitement des 26 images, suivre l'avancement et documenter.</p> |  |

## REALISATION

Procédure de test appliquée à chaque image : vérification de la syntaxe du Dockerfile (docker build --check), construction réelle, test de démarrage, publication sur un registre local, déploiement sur le cluster Kubernetes et vérification de l'accès (kubectl port-forward).

Exemples de corrections : import des clés GPG à jour pour les dépôts signés (signatures de paquets de nouveau acceptées) ; réinitialisation de la variable HOME et des droits du profil utilisateur ; contournement du module Python « crypt » retiré en 3.13 ; installation du trousseau gnome-keyring pour le démarrage du bureau sous Kubernetes.

(Captures d'écran des tests et des déploiements à insérer.)

| CONCLUSION                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Les 26 laboratoires sont de nouveau disponibles pour les formations. Au-delà des correctifs, j'ai formalisé une méthode de test reproductible et un guide de dépannage réutilisables par l'équipe. Cette activité m'a appris à diagnostiquer des pannes systémiques, à sécuriser une chaîne d'images (dépôts signés) et à mettre un service à la disposition d'utilisateurs. |

| EVOLUTION POSSIBLE                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Automatiser la construction et les tests des images dans un pipeline d'intégration continue (GitHub Actions), avec analyse de vulnérabilités des images (Trivy) et publication automatique sur un registre, afin de détecter les régressions avant toute mise à disposition. |