Rapport d'activité
Stage de 2ème année

Dautriche Leo
5 janvier 2026 au 27 février 2026

Responsable de stage : *Robin Straub*

Tuteur de stage *: Kévin Gillet*

Établissement / Formation : *Epsi Nantes - BTS SIO*

Entreprise d'accueil : *France Nuage, 11 avenue de la promenade 85140 Essarts-en-bocage*

Remerciements

À l'issue de ce stage au sein de l'entreprise France Nuage, je tiens à remercier toutes les personnes qui ont contribué à la réussite de cette expérience professionnelle.

Monsieur Robin Straub, mon responsable de stage, pour m'avoir accueilli au sein de l'entreprise France Nuage. Sa disponibilité et son encadrement bienveillant ont été déterminants dans le bon déroulement de mon stage.

Monsieur Kévin Gillet, mon tuteur de stage, pour son accompagnement au quotidien, ses conseils avisés et sa pédagogie m'ont permis de progresser rapidement et de m'intégrer pleinement à l'équipe.

L'ensemble de l'équipe de France Nuage, pour leur accueil, leur esprit d'équipe et leur bonne humeur. Travailler dans cet environnement professionnel et stimulant a été une véritable source d'enrichissement.

L'équipe pédagogique de l'EPSI Nantes pour la formation dans le cadre du BTS SIO, qui m'a fourni les compétences techniques et méthodologiques nécessaires pour aborder sereinement ce stage.

Table des matières

Table des matières **3**
1 - Présentation du candidat **5**
Présentation personnelle : 5
Projet BTS SIO – Infrastructure réseau 5
2 - Présentation de l'entreprise **6**
Secteurs d'activités 6
Cloud et Infrastructure 6
Observabilité et Monitoring 6
Sécurité et Gestion des accès 6
3 - Présentation du service et du tuteur **7**
Qu'est-ce que Kubernetes ? 7
4 - Missions réalisées durant le stage **8**
Qu'est-ce qu'un Helm Chart ? 8
Création et maintenance des Helm Charts 8
Qu'est-ce que Redis ? 8
Qu'est-ce que CloudNativePG (CNPG) ? 9
Chart GitLab 9
Chart Metabase 9
Chart Grafana 9
Chart Headscale 9
Chart Loki 10
Chart Mimir 10
Chart n8n 10
Chart Tempo 10
Chart OnlyOffice 10
5- Pipeline CI/CD et organisation du dépôt **11**
Refonte du pipeline CI/CD Plateforme : migration DinD vers BuildKit 11
Structure du pipeline GitLab CI Chart 11
Retravail et réorganisation du dépôt de charts 11
6 - Sécurité et gestion des secrets **12**
Mise en place de Sealed Secrets 12
Chiffrement des sauvegardes PVC avec Kopia 12
Sauvegarde et restauration avec Velero 12
Chart Velero - Backup des PVC Ceph 12
7 - Infrastructure et virtualisation **13**
Qu'est ce que proxmox ? 13
Template Cloud-Init pour Proxmox 13
Qu'est-ce que Grafana ? 13
Dashboard Grafana pour Ceph 14
8 - Outils **15**
Présentation des outils utilisé pendant le stage : 15

9 - Bilan des compétences acquises **16**
Compétences techniques 16
Compétences transversales 17
10 - Annexes **18**

1 - Présentation du candidat

Présentation personnelle :

Etudiant en deuxième année de BTS SIO SISR a l'EPSI.

Passionné par l'informatique, je consacre une grande partie de mon temps à expérimenter de nouvelles technologies, que ce soit à travers des projets personnels ou des infrastructures complexes que je mets en place.

Curieux et autonome, j'aime apprendre par la pratique et relever des défis techniques, notamment en développant des outils ou en mettant en place des services.

(CV Annexe 1)

Projet BTS SIO – Infrastructure réseau

Dans le cadre de ma formation, j'ai conçu et déployé une infrastructure réseau complète, hébergée sur un cluster de trois nœuds Proxmox. Ce projet m'a permis de mettre en pratique l'ensemble des compétences acquises en SISR, en construisant un environnement proche des conditions réelles d'une entreprise.

L'infrastructure repose sur une segmentation en plusieurs VLAN (SRV, USER, DMZ, SAN, GUEST, WAN) afin d'isoler les services et de renforcer la sécurité du réseau. J'y ai intégré de nombreuses briques technologiques : un Active Directory pour la gestion des utilisateurs, GLPI pour la gestion des tickets, PfSense en tant que pare-feu, ainsi qu'une solution de supervision avec Prometheus et Grafana. L'accès distant aux machines est assuré via Apache Guacamole, tandis qu'un bastion SSH permet d'administrer l'ensemble du réseau de manière sécurisée. L'automatisation des déploiements est gérée avec Ansible, et la documentation centralisée sur une instance Wiki.js.

Ce projet m'a confronté à des problématiques concrètes d'administration système et réseau, et m'a permis de développer une approche méthodique dans la conception et la maintenance d'une infrastructure complexe. Documentation disponible sur

https://wikileo.ovh

2 - Présentation de l'entreprise

France Nuage

Secteurs d'activités

France Nuage est une entreprise spécialisée dans le cloud souverain et les technologies open source. Elle accompagne les entreprises dans le déploiement de services et l'exploitation d'infrastructures cloud modernes.

Cloud et Infrastructure

- Déploiement de services sur Kubernetes
- Automatisation des déploiements via Helm, CI/CD et Infrastructure as Code
- Gestion du stockage distribué avec Ceph

Observabilité et Monitoring

- Monitoring des systèmes avec Grafana
- Centralisation des logs avec Loki
- Traçabilité des applications avec Tempo
- Stockage et analyse de métriques à grande échelle avec Mimir
- Création de tableaux de bord pour le suivi des performances

Sécurité et Gestion des accès

- Gestion sécurisée des secrets
- Mise en place de solutions d'authentification et de réseau sécurisé
- Sécurisation des sauvegardes et chiffrement des données

3 - Présentation du service et du tuteur

Durant toute la durée de mon stage, j'ai été chargée de la gestion et du développement de la plateforme cloud :

- La gestion de l'infrastructure Kubernetes
- Le développement et la maintenance des Helm charts
- La supervision et le monitoring de la plateforme
- L'amélioration des pipelines CI/CD
- La mise en place de nouvelles solutions open source.

Mon tuteur de stage m'a accompagné tout au long de cette période afin de m'aider à comprendre l'architecture de l'entreprise et les outils utilisés par l'entreprise.

Qu'est-ce que Kubernetes ?

Quand on développe une application et qu'on souhaite la mettre en ligne, la première réflexe est souvent de louer un serveur et d'y installer ce dont on a besoin. Cette approche fonctionne, mais elle montre rapidement ses limites dès que l'application grandit, qu'on a besoin de la mettre à jour sans interruption de service, ou qu'on veut gérer plusieurs applications en même temps. C'est là qu'intervient Kubernetes.

Kubernetes est un outil qui permet de gérer des applications conteneurisées sur un ensemble de machines. Au lieu de se connecter manuellement à un serveur pour démarrer ou redémarrer une application, on décrit simplement ce qu'on veut — combien d'instances faire tourner, combien de mémoire allouer, comment exposer le service sur le réseau — et Kubernetes s'occupe du reste. Si une application plante, il la redémarre tout seul. Si le trafic augmente, il peut en lancer de nouvelles instances automatiquement. C'est en quelque sorte un chef d'orchestre qui surveille en permanence que tout tourne comme prévu, et qui réagit sans intervention humaine quand ce n'est pas le cas.

Chez France Nuage, l'ensemble de la plateforme repose sur Kubernetes, ce qui explique pourquoi une grande partie de mes missions a consisté à travailler avec ses outils et ses concepts.

4 - Missions réalisées durant le stage

Durant mon stage du 5 janvier 2026 au 27 février 2026 au sein de France Nuage, j'ai travaillé sur le développement et la maintenance de l'infrastructure cloud de l'entreprise. Les missions couvrent la création de Helm Charts, la refonte de pipeline CI/CD, la gestion de la sécurité et la mise en place de sauvegardes.

Qu'est-ce qu'un Helm Chart ?

Helm est un gestionnaire de paquets pour Kubernetes, souvent comparé à ce qu'est apt pour Linux. Il permet de déployer des applications sur un cluster Kubernetes de manière standardisée, reproductible et configurable. il s'agit d'un ensemble de fichiers décrivant une application Kubernetes, comprenant les templates de ressources (Deployments, Services, ConfigMaps, Ingress, etc.) ainsi qu'un fichier de valeurs permettant de personnaliser le déploiement selon l'environnement ou l'organisation cible. Plutôt que d'écrire et de maintenir manuellement des dizaines de fichiers YAML pour chaque application et chaque client, Helm permet de factoriser cette configuration dans un chart réutilisable, dont les paramètres peuvent être surchargés à la demande.

Création et maintenance des Helm Charts

Le dépôt de charts constitue le cœur de la plateforme France Nuage. Chaque chart suit une structure standardisée avec un dossier Chart/ contenant les templates Kubernetes, un script deploy.sh automatisant le déploiement, et un dossier orgs/ permettant une configuration multi-organisation. Le pipeline CI/CD valide automatiquement chaque chart via helm lint, helm template et kubesec.

Qu'est-ce que Redis ?

Redis est une base de données qui stocke les données directement en mémoire vive, ce qui la rend très rapide. On l'utilise principalement comme cache pour éviter des requêtes répétées en base de données, ou comme file de tâches pour distribuer du travail entre plusieurs instances d'une application.

Qu'est-ce que CloudNativePG (CNPG) ?

CloudNativePG est un opérateur Kubernetes qui simplifie le déploiement et la gestion de bases de données PostgreSQL. Il s'occupe automatiquement de la réplication entre les nœuds, des bascules en cas de panne, et des sauvegardes vers un stockage externe comme S3.

Chart GitLab

Durant mon stage j'ai pu développer un Helm Chart pour le déploiement de GitLab sur Kubernetes. Le chart intègre une instance GitLab complète avec gestion des dépendances externes (PostgreSQL via CloudNativePG, stockage objet S3, configuration des Ingress et des SealedSecrets pour les credentials de backup CNPG. (architecture du chart en annexe 2)

Chart Metabase

Durant mon stage j'ai pu mettre en place un chart Metabase. Le chart supporte deux modes : standalone avec PostgreSQL Bitnami subchart, et mode HA avec CloudNativePG. Il inclut des templates pour le Deployment, la ConfigMap, le Service, la NetworkPolicy egress et le cluster CNPG.

Chart Grafana

J'ai pu développer un chart Grafana avec PostgreSQL et Redis. Le chart utilise les sous-charts Bitnami pour PostgreSQL (v18.2.0) et Redis. La connexion aux sources de données Loki, Mimir et Tempo est configurée pour offrir une observabilité complète de la plateforme.

Chart Headscale

Participer à la création d'un chart Headscale , serveur de coordination VPN basé sur le protocole WireGuard, alternative open source à Tailscale. Le chart est multi-organisation avec génération automatique des secrets. Il inclut Headplane comme interface d'administration web, avec persistance SQLite via un PVC Ceph RBD. Chaque organisation est accessible via un namespace dédié.

Chart Loki

Déploiement d'un chart Loki en mode microservices distribué pour une haute disponibilité complète. L'architecture implémente le multi-tenant via et répartit les pods sur différentes zones via l'anti-affinité. Les composants stateless (distributor, querier, query-frontend) sont déployés avec 2 replicas, les composants stateful (ingester, compactor) avec 3 replicas et un replication factor de 2.

Chart Mimir

Mise en place de Grafana Mimir en mode HA distribué pour le stockage de métriques à grande échelle avec du S3 et anti-affinité multi-zones assure la résilience. Les composants stateful (ingester, compactor, store-gateway) sont déployés avec 3 replicas. Le chart utilise mimir-distributed en sous-dépendance.

Chart n8n

Déploiement de n8n, plateforme d'automatisation de workflows. Le chart gère deux modes : standard avec PostgreSQL Bitnami, et mode HA avec CloudNativePG, Redis et plusieurs replicas. Le mode HA active une infrastructure queue avec Redis pour la distribution des tâches entre workers. Le script deploy.sh gère la génération de la clé de chiffrement N8N_ENCRYPTION_KEY et le sidecar Tailscale optionnel. (photo de la réalisation en annexe 3)

Chart Tempo

Création du chart Grafana Tempo en mode distribué HA avec découverte via memberlist. L'architecture comprend 7 types de composants : distributor (x2), ingester StatefulSet (x3), querier (x2), query-frontend (x2), compactor StatefulSet (x3), metrics-generator (x2) et auth-proxy nginx (x2). Le metrics-generator envoie les span metrics vers Mimir. Le stockage des traces utilise le backend S3 Ceph RGW. (photo du ring status en annexe 4)

Chart OnlyOffice

Développement du chart OnlyOffice avec support PostgreSQL standalone ou HA (CloudNativePG) et Redis en réplication. Le chart intègre un backup barman S3 via CNPG ScheduledBackup et utilise un PVC ReadWriteMany pour le stockage partagé. Les SealedSecrets chiffrent les crédenciales Redis, OnlyOffice et les accès S3.

5- Pipeline CI/CD et organisation du dépôt

Refonte du pipeline CI/CD Plateforme : migration DinD vers BuildKit

L'une des missions structurantes du stage a été la refonte du pipeline du repo plateforme. L'ancienne approche Docker-in-Docker (DinD) nécessitait des privilèges élevés et donc potentiellement une évasion de conteneurs, ce qui posait des problèmes de sécurité en environnement Kubernetes.

La migration vers BuildKit avec le driver Kubernetes de docker buildx permet des builds rootless. Un Role RBAC dédié (buildkit-rbac.yaml) est appliqué par organisation, accordant au ServiceAccount les droits de créer des Deployments BuildKit, gérer des Pods et accéder aux Secrets et ConfigMaps. (photo du pipeline en annexe 5)

Structure du pipeline GitLab CI Chart

Le pipeline .gitlab-ci.yml est organisé en deux stages : validate et publish. La stage de validation exécute cinq jobs en parallèle :

- lint-and-test : helm dependency update, helm lint et helm template sur chaque chart
- version-check : vérification de la version dans chaque Chart.yaml
- yaml-lint : validation YAML de tous les values.yaml via yamllint
- chart-schema : vérification des champs obligatoires apiVersion, name, version, type
- kubesec : analyse de sécurité des manifestes générés via kubesec scan

La stage publish exécute publish-oci qui package chaque chart avec helm package et le pousse dans le registre OCI GitLab via helm push, uniquement sur la branche par défaut. (photo du pipeline en annexe 6)

Retravail et réorganisation du dépôt de charts

Le dépôt a fait l'objet d'un important travail de standardisation. Chaque chart suit désormais une structure cohérente avec un Chart.yaml contenant les métadonnées

(apiVersion, description, appVersion, dépendances), un README.md documentant l'architecture, un script deploy.sh automatisant le déploiement multi-organisation, et un dossier orgs/ isolant les configurations par client. (annexe 2)

6 - Sécurité et gestion des secrets

Mise en place de Sealed Secrets

Déploiement du contrôleur Sealed Secrets via un chart Helm dédié dans kube-system. Ce système permet de chiffrer les secrets Kubernetes avec la clé publique du cluster afin de les stocker en toute sécurité dans Git, conformément à l'approche GitOps de France Nuage.

Le workflow mis en place : création d'un secret Kubernetes en dry-run, chiffrement avec kubeseal --format yaml, puis commit du SealedSecret dans le dépôt. Le contrôleur se charge du déchiffrement automatique à l'application. Cette approche est utilisée dans tous les charts pour les secrets. (schéma draw io de la solution sealed secret en annexe 7)

Chiffrement des sauvegardes PVC avec Kopia

Implémentation du chiffrement des sauvegardes de volumes persistants (PVC) avec Kopia, intégré comme uploader dans Velero (uploaderType: kopia). Kopia apporte le chiffrement de bout en bout, la déduplication et la compression des données sauvegardées vers le stockage S3.

Sauvegarde et restauration avec Velero

Chart Velero - Backup des PVC Ceph

Développement du Helm Chart Velero pour la sauvegarde quotidienne des volumes persistants Ceph et des manifestes Kubernetes vers un stockage S3. Le chart utilise la fonctionnalité CSI pour la prise de snapshots natifs des PVC Ceph RBD et CephFS via le VolumeSnapshotClass ceph-rbd-snapshot.

La planification des sauvegardes est configurée via un ScheduledBackup quotidien à 2h du matin avec une rétention de 3 semaines (504h0m0s).

7 - Infrastructure et virtualisation

Qu’est ce que proxmox ?

Quand on veut faire tourner plusieurs serveurs sans pour autant acheter une machine physique pour chacun d'eux, on fait appel à la virtualisation. Le principe est simple : une seule machine physique puissante fait tourner plusieurs machines virtuelles, chacune totalement isolée des autres, avec son propre système d'exploitation, ses propres ressources et ses propres services.

Proxmox est un outil open source qui permet de gérer tout ça depuis une interface web. On peut y créer des machines virtuelles classiques, mais aussi des conteneurs légers (LXC), les démarrer, les arrêter, les cloner, les sauvegarder, et même les regrouper en cluster pour répartir la charge sur plusieurs serveurs physiques.

Template Cloud-Init pour Proxmox

Création d’un template Cloud-Init pour l’hyperviseur Proxmox VE permettant de déployer un proxmox dans proxmox (nested) via controlplane. (script en annexe 9)

Qu'est-ce que Grafana ?

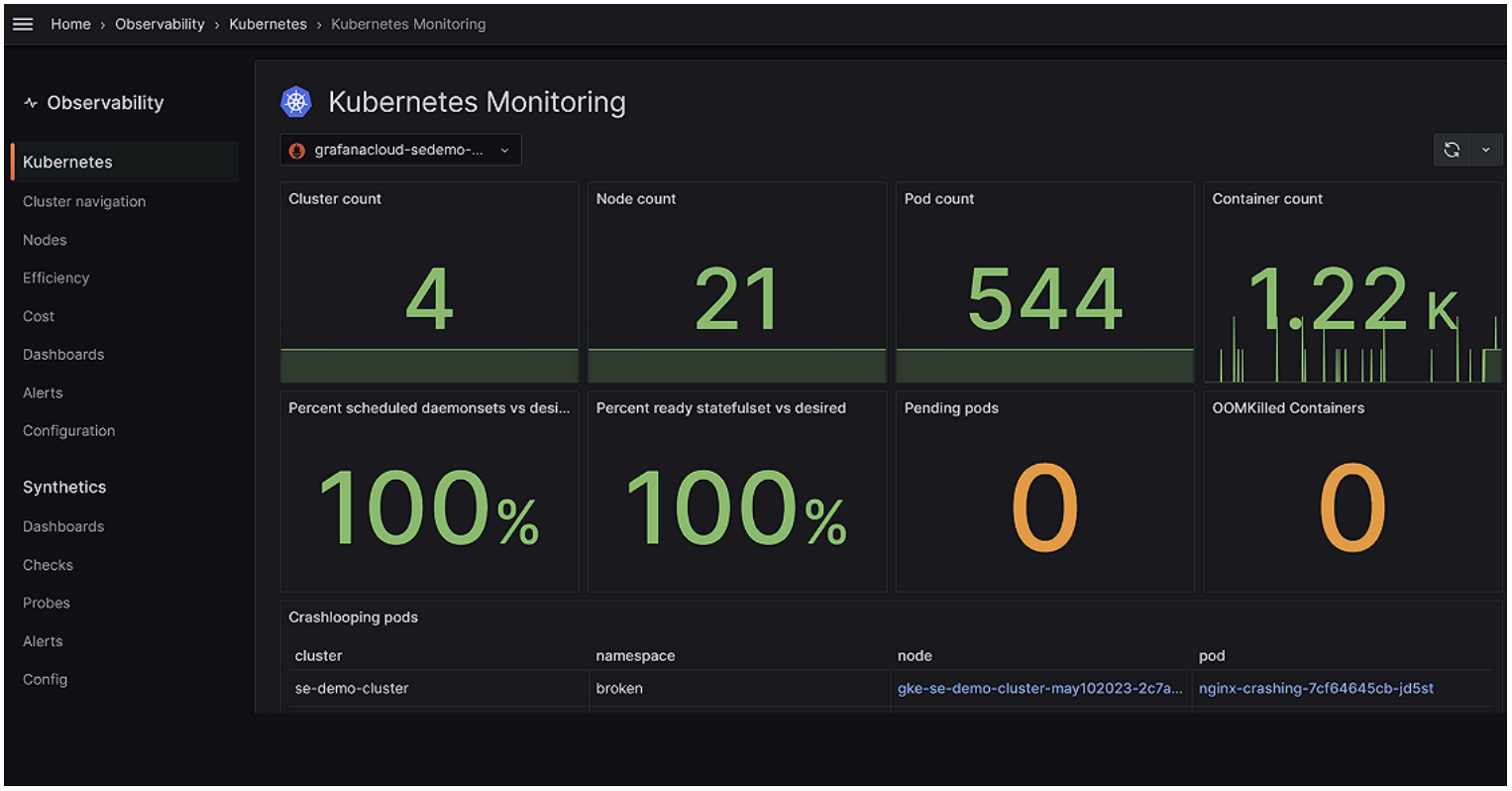
Grafana est un outil de visualisation et de supervision qui permet de répondre à cette question en temps réel. Il se connecte à des sources de données — des bases de métriques, de logs ou de traces — et permet de construire des tableaux de bord clairs et interactifs pour suivre l'état de son infrastructure. On peut y afficher n'importe quelle métrique sous forme de graphiques, configurer des alertes qui se déclenchent automatiquement quand quelque chose sort de l'ordinaire, et naviguer dans les logs d'une application sans avoir à se connecter manuellement à chaque serveur.

Grafana fonctionne rarement seul : il s'appuie sur d'autres outils pour collecter et stocker les données. Dans le cadre de mon stage chez France Nuage, il était associé à Loki pour les logs, Mimir pour les métriques et Tempo pour les traces, formant ensemble une stack d'observabilité complète pour surveiller l'ensemble de la plateforme.

Dashboard Grafana pour Ceph

Développement de tableaux de bord Grafana dédiés à la supervision du cluster de stockage distribué Ceph. Ces dashboards permettent de visualiser l'état du cluster les métriques d'I/O, la capacité utilisée par pool et les alertes en temps réel. Ils s'intègrent à la stack d'observabilité de la plateforme. (photo du dashboard en annexe 10)

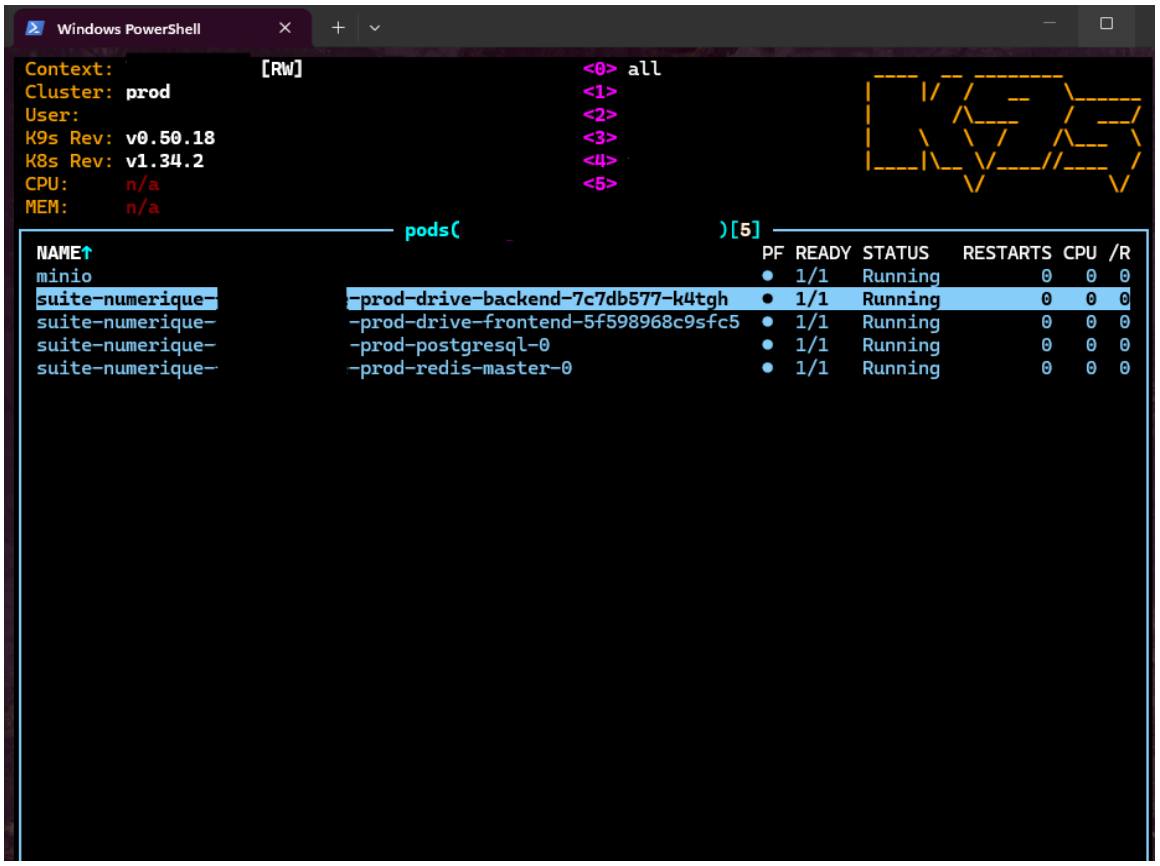
Exemple d'un dashboard Grafana :



8 - Outils

Présentation des outils utilisé pendant le stage :

K9S - Pour la gestion du cluster Kubernetes (suivre l'état d'un déploiement pod par pod, listing de namespace, logs des pods ect) :



Git Bash - Pour le déploiement des scripts bash, commande pour interagir avec un déploiement :

```
$ ./deploy.sh [redacted] prod [redacted] drive
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "nfs-ganesha-server-and-external-provisioner" chart repository
...Successfully got an update from the "longhorn" chart repository
...Successfully got an update from the "vmware-tanzu" chart repository
...Successfully got an update from the "rook-release" chart repository
...Successfully got an update from the "n8n-community" chart repository
...Successfully got an update from the "gabe565" chart repository
...Successfully got an update from the "nbcloud" chart repository
...Successfully got an update from the "grafana" chart repository
...Successfully got an update from the "gitlab" chart repository
...Successfully got an update from the "bitnami" chart repository
Update Complete. ⎈Happy Helming!⎈
Saving 2 charts
Downloading postgresql from repo oci://registry-1.docker.io/bitnamicharts
Pulled: registry-1.docker.io/bitnamicharts/postgresql:18.2.0
Digest: sha256:c6c99d1de252d00d6a3942e5118c28c4396fc7a891d222348d99d872e0fd7374
Downloading redis from repo oci://registry-1.docker.io/bitnamicharts
Pulled: registry-1.docker.io/bitnamicharts/redis:20.6.3
Digest: sha256:b63a6dba4a1591d54e205d8597f1435c4703582ba548cf8b2c692e9acdcd5060
Deleting outdated charts
Application de sealed-secret-suite-numerique.yaml...
sealedsecret.bitnami.com/suite-numerique-            -prod-secrets unchanged
outil actif : drive
Release "suite-numerique-            -prod" does not exist. Installing it now.
NAME: suite-numerique-            -prod
LAST DEPLOYED: Sat Mar 21 18:42:56 2026
NAMESPACE:
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
NAME                                                          READY   STATUS    RESTARTS   AGE
minio                                                         1/1     Running   0          24h
suite-numerique-            -prod-drive-backend-7c7db577-k4tgh    1/1     Running   0          2m4s
suite-numerique-            -prod-drive-frontend-5f598968c9sfc5   1/1     Running   0          2m4s
suite-numerique-            -prod-postgresql-0                    1/1     Running   0          2m4s
suite-numerique-            -prod-redis-master-0                  1/1     Running   0          2m4s
```

9 - Bilan des compétences acquises

Ce stage de deux mois chez France Nuage m'a permis de développer des compétences techniques avancées dans l'écosystème cloud-native et DevOps, en travaillant directement sur une infrastructure de production utilisée par de nombreux clients.

Compétences techniques

- Kubernetes : déploiement, gestion des ressources, RBAC, NetworkPolicy, StatefulSets, anti-affinité multi-zones
- Helm : création de charts complexes, gestion des dépendances OCI, publication via registre OCI GitLab
- Observabilité : stack complète Grafana / Loki / Mimir / Tempo en mode HA microservices distribué

- CI/CD : pipeline GitLab CI avec validation, lint, analyse de sécurité kubesec et publication OCI
- Sécurité : Sealed Secrets, gestion GitOps des credentials chiffrés dans le dépôt
- Sauvegarde : Velero avec CSI snapshots Ceph, chiffrement Kopia, rétention et planification
- Infrastructure : Proxmox Cloud-Init, stockage distribué Ceph RBD et CephFS

Compétences transversales

- Autonomie dans la résolution de problèmes
- Rédaction de documentation technique (README, scripts commentés)
- Travail en équipe avec GitLab (merge requests, branches feat/, code review)
- Rigueur dans l'approche GitOps : tout en code, tout versionné, tout reproductible

10 - Annexes

20 ans
Titulaire du Permis B (Véhiculé)

Contact

Nantes

06.32.26.68.93

dautriche.leo03@gmail.com

wikileo.ovh

Soft Skills

Communicatif
Créatif
Perséverant
Curiosité

Langue

Anglais

Intérêts

Projets personnels :

- Portfolio maintenu sur un VPS
- Administrateur d'un serveur Dayz (Gestion de Tickets en Anglais)

Léo DAUTRICHE

Expérience professionnelle

CDD
Mars-Juin 2026

France Nuage Administrateur Système (Cloud souverain)

- Assurer la continuité opérationnelle des services France Nuage
- Participer à l'astreinte et à la réponse aux incidents
- Participer à l'automatisation de la mise en place des services cloud pour le compte de clients

STAGE DE 2ÈME ANNÉE BTS SIO
Janvier-Février 2026

France Nuage DevOps (Cloud souverain)

- Création de helm chart Kubernetes : (Loki,Mimir,Tempo,Nextcloud,N8N,Gitlab,Headscale,Grafana, onlyoffice,listmonk)
- Refonte du pipeline de CI/CD docker in docker pour buildkit rootless et chart de déploiement
- Création d'un template cloudinit proxmox nested
- Création de dashboard Grafana pour cluster Ceph
- Modification de la gestion des secrets Vars CI/CD gitlab pour SealedSecret
- Mise en place d'un chart Velero pour backup de pvc ceph et manifestes yaml vers un storage S3 + chiffrement avec kopia des pvc
- Refonte du repo de Charts pour pouvoir installez des charts via un controlplane en pushant vers un registry oci

PROJET SISR
2025

Documentation sur wikileo.ovh
Monter une infrastructure avec cluster de proxmox comprenant un GLPI, un Active Directory, Ansible, Firewall, Serveur Minecraft, Wikijs, Prometheus/Grafana, Bastion, Apache Guacamole, Configuration de switch (Vlans, Trunk)

STAGE DE 1ÈRE ANNÉE BTS SIO
Mai 2025

La Tête Dans La Toile

- Migration d'une visite virutelle de threejs a pannellumjs (PHP, JS, SCSS)
- Script C# permettant de synchroniser des bdd postgresql
- Automatisation de tâche via N8N
- Création d'une barre de recherche boosté avec un LLM pour de la recherche intelligente

Compétences

Langages maitrisés :
PHP, SQL, HTML, SCSS, C++, Python, C#, YAML, C++, Tailwind CSS.

Infrastructure :
Prometheus/Grafana, GLPI, Active Directory, PFsense, Docker, Configuration Routeur/Switch, Ansible, Kubernetes.

Parcours Académique

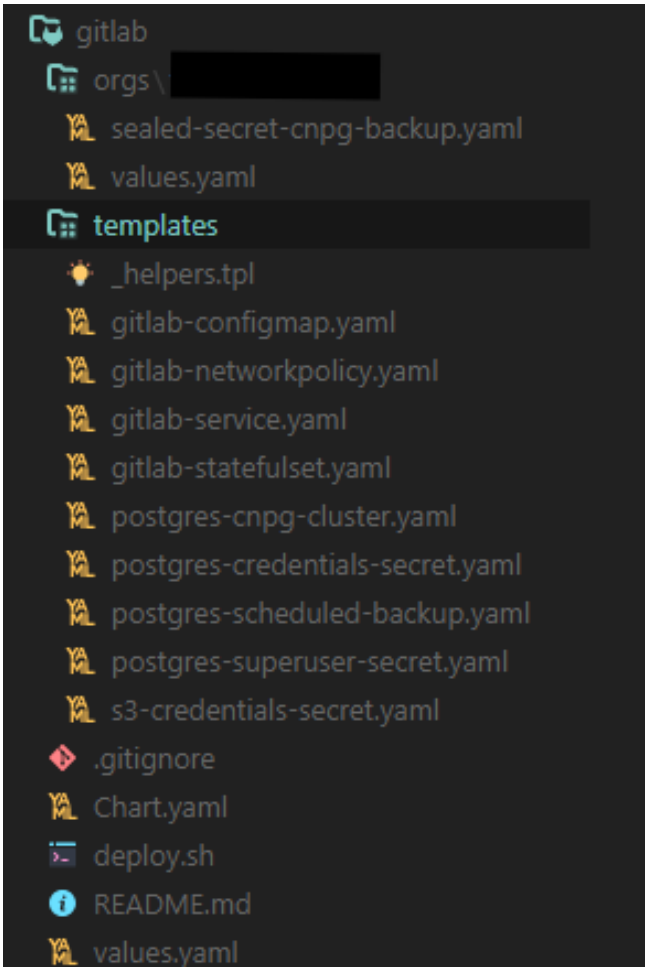
EPSI ECOLE D'INGÉNIERIE INFORMATIQUE
2026 2027
3 ème année de Bachelor

EPSI ECOLE D'INGÉNIERIE INFORMATIQUE
2024 2026
1 ère et 2 ème année de BTS SIO SISR

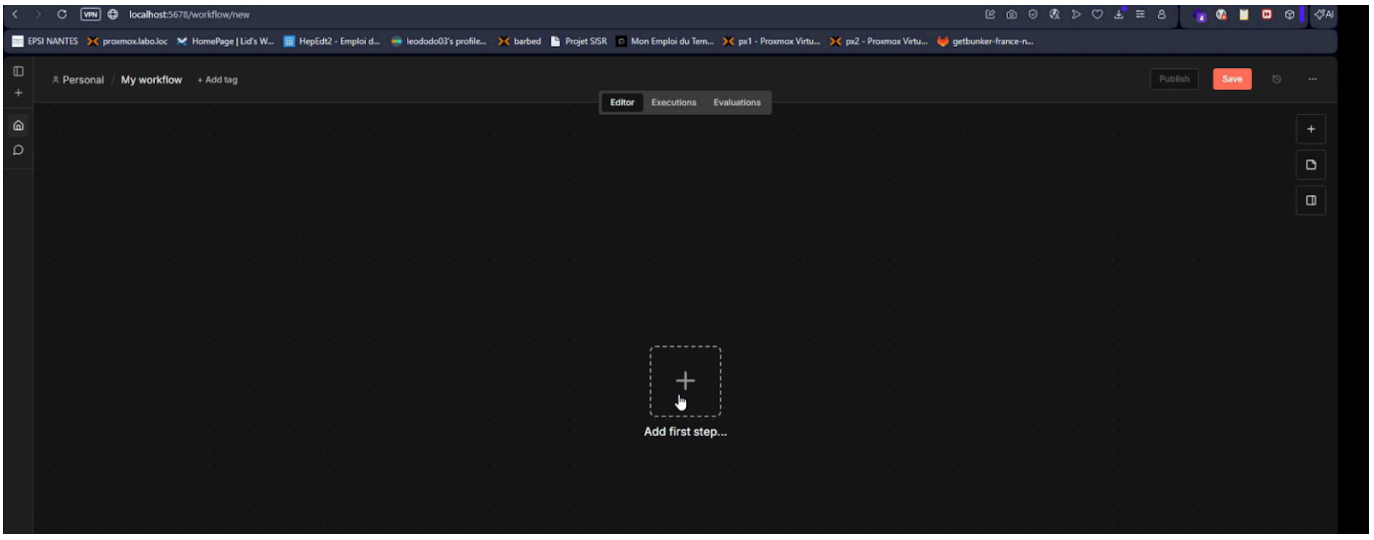
Diplôme

LYCEE POLYVALENT DE LA HERDRIE
2024
Bac général - spécialités Maths et NSI
Option Maths expert

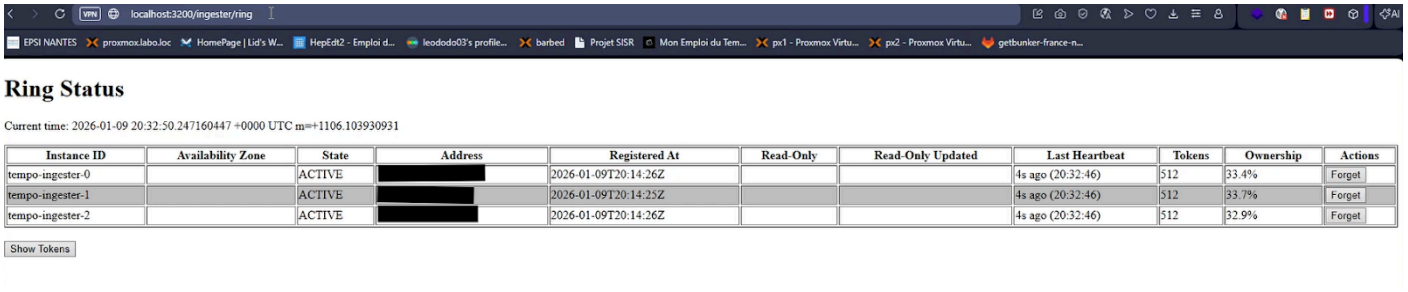
Annexe – 1 CV



Annexe – 2 Structure chart Gitlab



Annexe – 3 chart n8n ui web



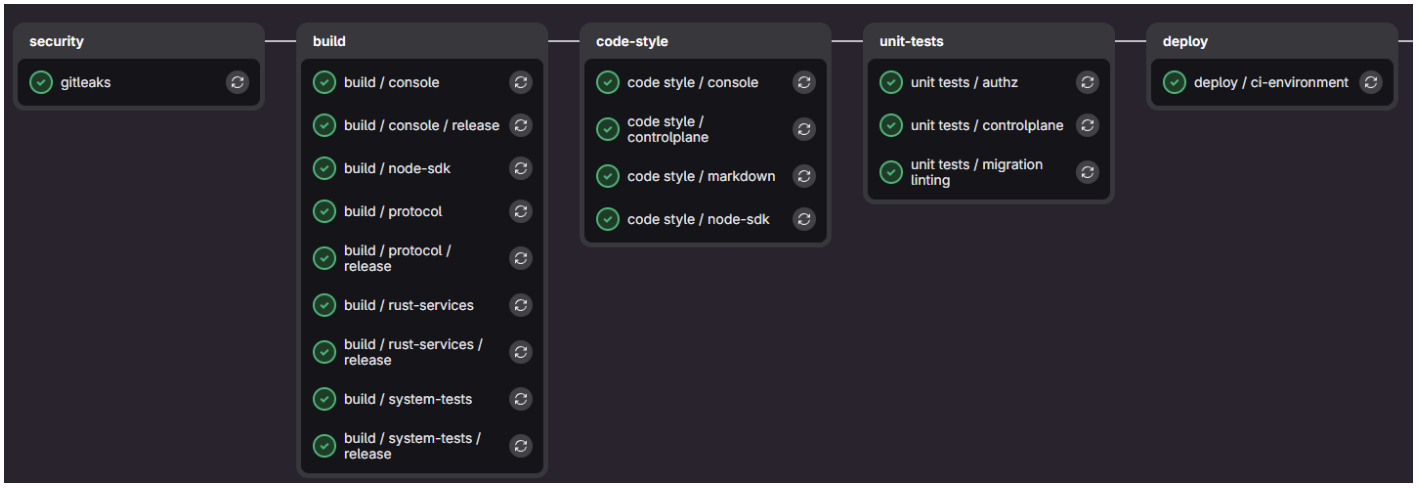
Ring Status

Current time: 2026-01-09 20:32:50.247160447 +0000 UTC m=+1106.103930931

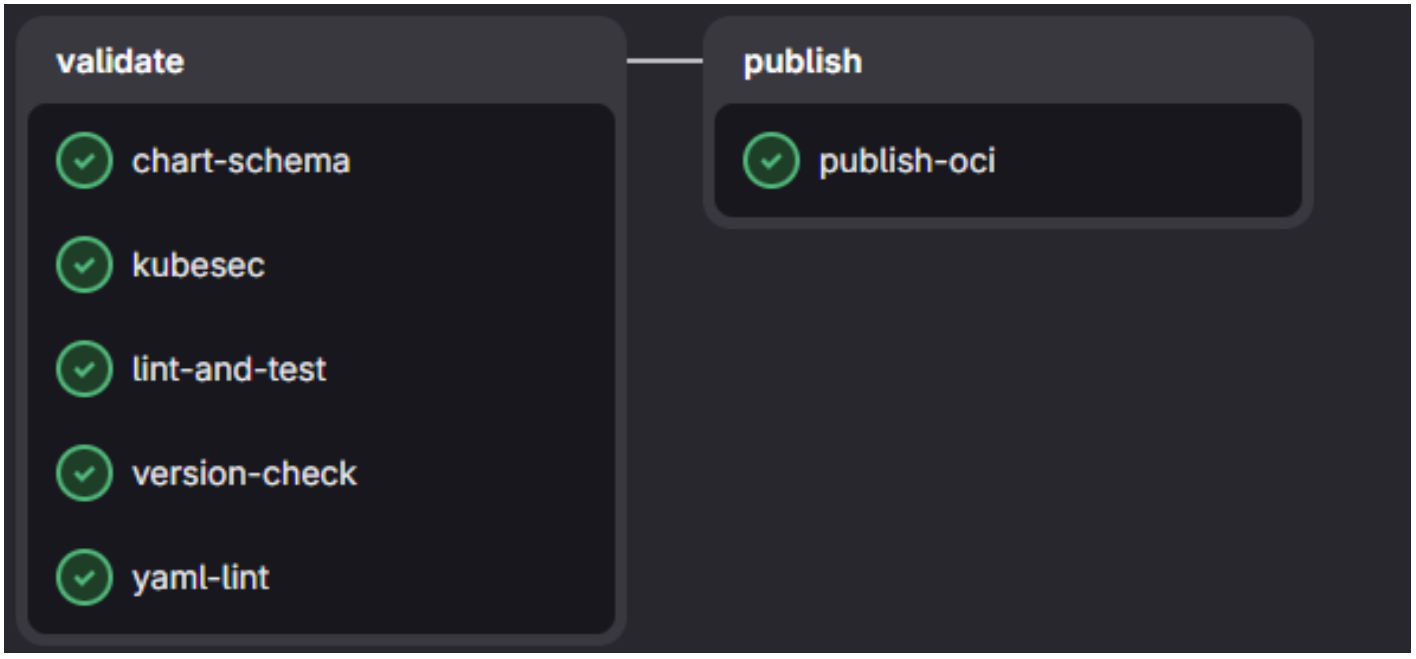
Instance ID	Availability Zone	State	Address	Registered At	Read-Only	Read-Only Updated	Last Heartbeat	Tokens	Ownership	Actions
tempo-ingester-0		ACTIVE	[redacted]	2026-01-09T20:14:26Z			4s ago (20:32:46)	512	33.4%	Forget
tempo-ingester-1		ACTIVE	[redacted]	2026-01-09T20:14:25Z			4s ago (20:32:46)	512	33.7%	Forget
tempo-ingester-2		ACTIVE	[redacted]	2026-01-09T20:14:26Z			4s ago (20:32:46)	512	32.9%	Forget

Show Tokens

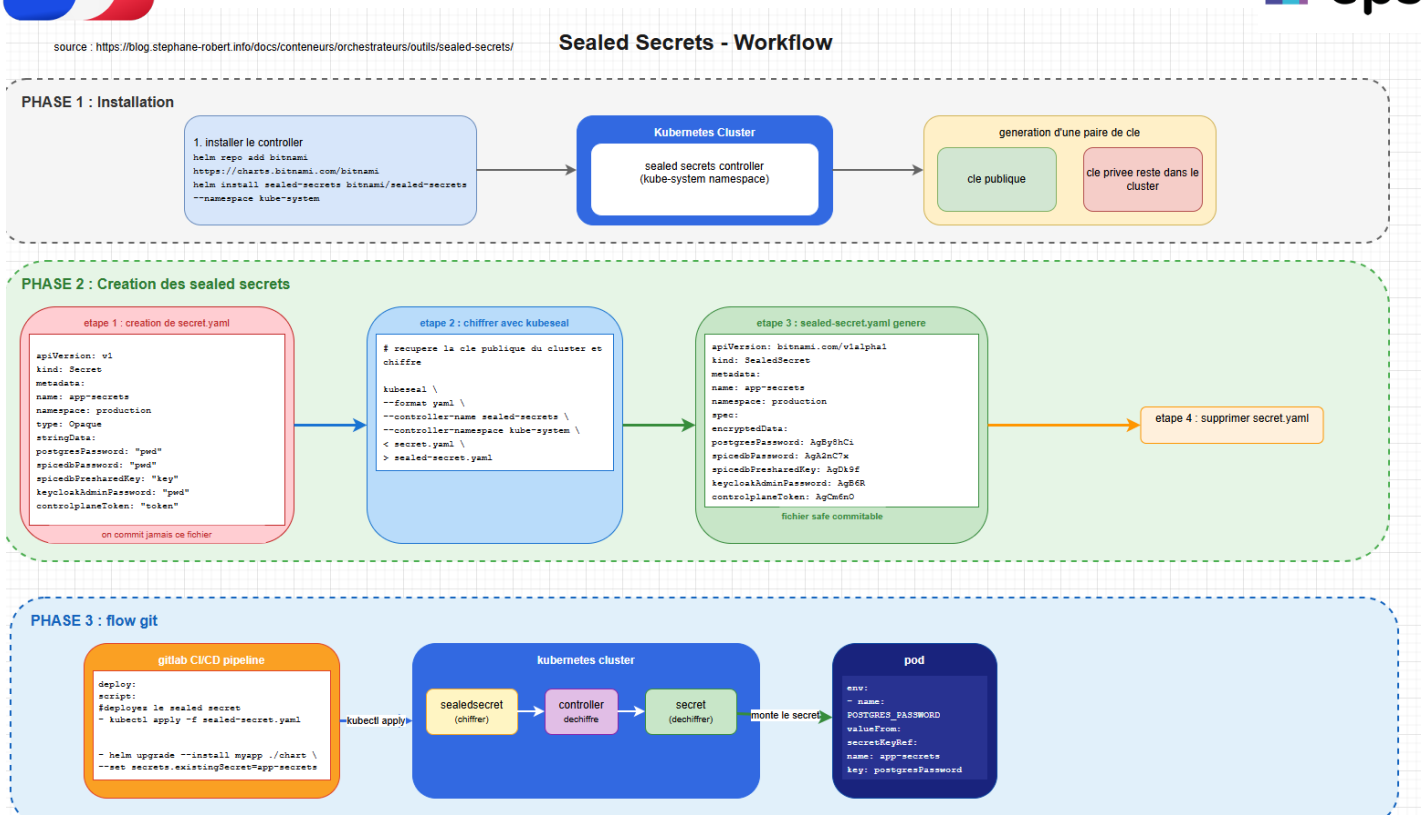
Annexe – 4 chart tempo Ring Status



Annexe – 5 pipeline CI/CD Plateforme



Annexe – 6 pipeline CI/CD Chart



Annexe – 7 schéma draw io de la solution sealed secret

```
#cloud-config
hostname: pve-nested
fqdn: pve-nested.proxmox.com
manage_etc_hosts: false
disable_root: false

package_update: true
package_upgrade: true

packages:
  - curl
  - gnupg
  - debian-keyring
  - debian-archive-keyring
  - apt-transport-https
  - qemu-guest-agent

runcmd:
  - |
    IP_ADDR=$(ip -4 route get 1.1.1.1 | grep -oP 'src \K\S+')
    echo "127.0.0.1 localhost" > /etc/hosts
    echo "$IP_ADDR pve-nested.proxmox.com pve-nested" >> /etc/hosts
    echo "::1 localhost ip6-localhost ip6-loopback" >> /etc/hosts
  - curl -fsSL https://enterprise.proxmox.com/debian/proxmox-release-trixie.gpg -o /etc/apt/trusted.gpg.d/proxmox-release-trixie.gpg
  - echo "deb [arch=amd64] http://download.proxmox.com/debian/pve trixie pve-no-subscription" > /etc/apt/sources.list.d/pve-install-repo.list
  - export DEBIAN_FRONTEND=noninteractive
  - apt-get update
  - apt-get full-upgrade -y
  - apt-get install -y proxmox-ve postfix open-iscsi chrony
  - apt-get remove -y 'linux-image-6.*' || true
  - update-grub
  - |
    until [ -d /etc/pve/priv ]; do sleep 2; done
    echo "token: root@pam!static-token:0:0::" >> /etc/pve/user.cfg
    echo "root@pam!static-token e0f92242-f929-41b2-9152-714914f0b993" >> /etc/pve/priv/token.cfg
    chmod 600 /etc/pve/priv/token.cfg
  - postconf -e "inet_interfaces = loopback-only"
  - postconf -e "mydestination = localhost"
  - systemctl restart postfix
  - systemctl enable --now qemu-guest-agent

power_state:
  mode: reboot
  timeout: 300
  condition: true
```

codesnap.dev

Annexe – 8 script Proxmox nested

Annexe – 10 dashboard Grafana cluster ceph