

# Kubernetes à l'Université de Limoges

---

Déploiement hybride d'OpenShift (VM/Métal)  
et déploiement de JupyterHub via argoCD

# Genèse du projet et contexte

- 2015/2016 : Premiers pas avec Docker
- 2017 : Premiers Docker en production
  
- 2019 : Premiers pas avec Kubernetes
- 2021 : Premier Kubernetes en préproduction -> ne passera pas en prod
  
- 2023 : 10<sup>ème</sup> serveur d'hébergement Docker en prod
- 2023 : Projet Kubernetes pour la Prod
- 2023 : Infra dédiée à Kubernetes
- 2024 : Déploiement d'Openshift

# Objectifs et choix

- Tout en Gitops : Gitlab + Helm Charts + ArgoCD
- Le plus de choses fournies clé en main aux Dev et Intégrateurs
  
- Une infrastructure dédiée
- Une solution supportée et packagée : OpenShift
- Un accompagnement par Red Hat
  
- 2 Applications cibles
  - Cloud native : JupyterHub ( <https://jupyter.org/hub> )
  - Conçu par nos soins : Hosting : outil maison pour fournir des hébergements web et CMS

# Micro-focus Openshift Platform Plus

- RH Openshift Container Platform
  - + RHACS - RH Advanced Cluster Security
  - + RHACM - RH Advanced Cluster Management
  - + ODF - Openshift Data Foundation (Ceph)
  - + RH Quay – Registry
- Quelques features
  - Paquage : K8S + Monitoring + backup + log + Argo + monitoring + etc.
  - Mise à jour automatisée de l'ensemble des briques du cluster
  - Sécurité poussée + RBAC
  - Surveillances de l'obsolescence des images
  - Registry « sources » certifiées
  - Test de la sécu du code (on a pas encore testé)

# Equipe (2j sem + sprints)

- 1 chef de projet :
  - Compétences spécifiques : intégration, container, HA, services tiers
- 1 spécialiste infra :
  - Compétences spécifiques : matériels, serveurs, stockages, déploiement, admin
- 1 spécialiste service :
  - Compétences spécifiques : intégration, container, automatisation, DevOps, déploiement, admin de jupyterhub
- 1 admin réseau en appui régulier

# Infrastructure Stretch (2 salles)

## La partie Métal

- Worker : 4 Serveurs pour faire tourner les applis (HP ProLiant DL385)
  - 2 X 28 Core + 512 Go Ram + 512 Go -> 2 To stockage

## La partie virtualisée (8 vCPU, 32 Go Ram, 128Go) aujourd'hui Over VMware -> probablement Over Proxmox après

- Le control plan : 3 VM
- Les nœuds d'infra : 4 VM
- Les load-balancers : 3 x 1 VM
- Le serveur Gitops : 1 VM
- Les serveurs de déploiement : bootstrap, PXE, etc.

## Le stockage (en cours de déploiement, prod dans 3 semaines)

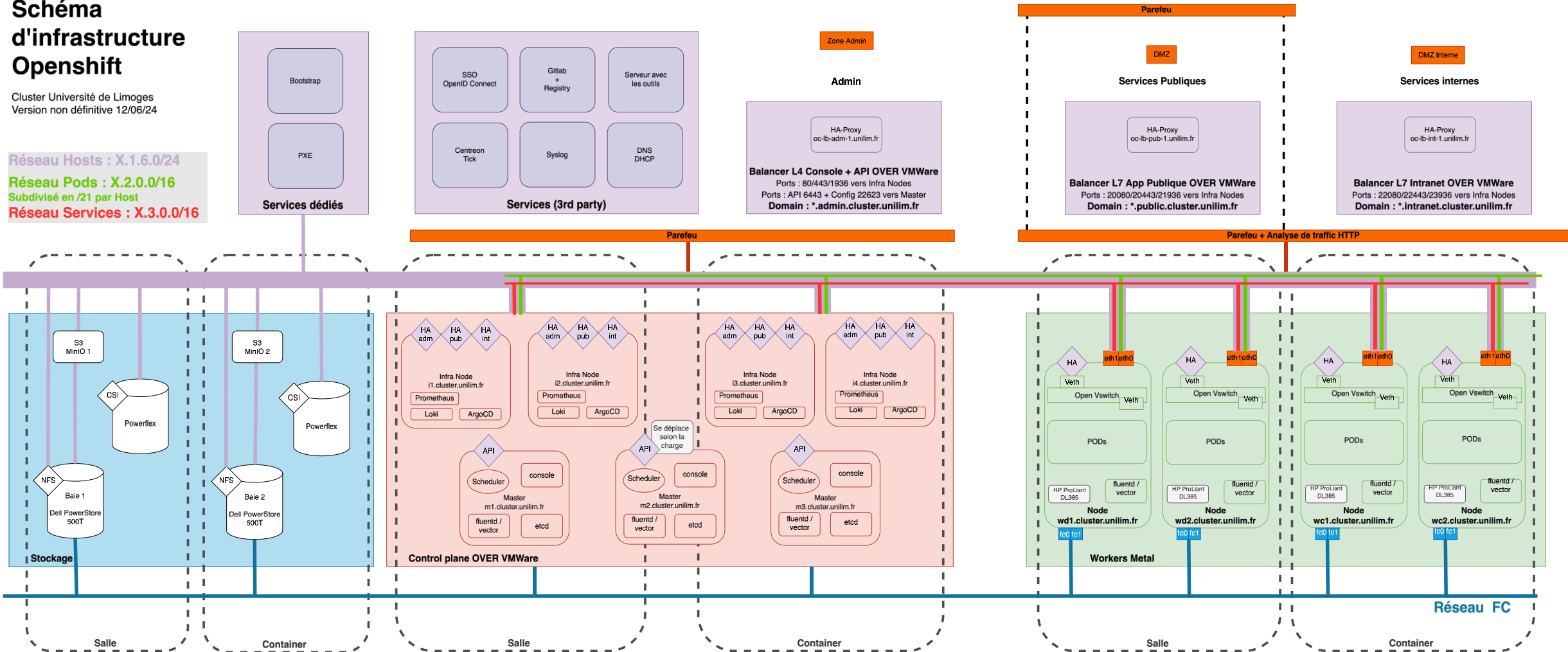
- 1 infrastructure PowerFlex
- 2 baies en block + NFS reliées en FC
- 1 stockage S3 basé sur MinIO

# Schéma d'Infrastructure

## Schéma d'infrastructure Openshift

Cluster Université de Limoges  
Version non définitive 12/06/24

Réseau Hosts : X.1.6.0/24  
Réseau Pods : X.2.0.0/16  
Subdivisé en /21 par Host  
Réseau Services : X.3.0.0/16



# Accompagnement

- Accompagnement RedHat « FastStart »
  - Design Revue : 1 semaine d'atelier  
(appli, workflow, infra, storage, network, sécu, dependencies, etc.)
  - Platform Engineering : 3 x 3 jours sur site
- Formation en ligne
  - Red Hat Learning Subscription Standard  
(abonnement annuel pour une personne + 7 examens)

# Ex : Sujet abordé lors d'un atelier de design réseau

- OpenShift Network
- Load Balancers
- CNI
- Hostnames
- DNS
- DMZ
- Current Network Infrastructure
- DHCP
- Network Performance
- Reserved IP ranges
- Firewall
- Network Sizing
- Network Segregation
- Proxies
- High availability

# Déploiement Openshift + JupyterHub

Cadrage	• 11 et 12 janvier 2024
Design Revue	• 29 janvier au 02 février - 5 ateliers de 1/2j
Déploiement infra	• 05 au 09 février
Sprint 1 : déploiement Openshift	• 12, 13, 14 février
Ajustement infra 1	• 15 février au 12 mars
Sprint 2 : déploiement JupyterHub	• 13, 14, 15 mars
Ajustement infra 2 + commande stockage	• 18 mars - 30 avril (2J-sem)
Doc + Reconstruction cluster from scratch	• 29, 30 avril + reprise des docs
Sprint 3 : Pipeline + admin	• 21, 22, 23 mai
Déploiement du stockage	• 17 au 28 juin – Installation Powerflex et intégration à Openshift + transfert de compétence
Finalisation de l'intégration	• début juillet
Configuration Jupyter	• En cours

# Micro-focus Stockage

- Pourquoi pas nos baies en place ?
  - Avec jupyter on va provisionner 1 user = 1 volume -> cible initiale 2000 User
    - Nos baies fournissent au max 1000 volumes
    - On ne voulait pas contourner ça « salement » en NFS
  - Pas de gestion « automatique » de la réplication métró (contournable)
  - Pas de gestion de la bascule si on perd un site
  - Amplification des IO avec Ceph sur notre infra en place + Coût d'overhead
- Pourquoi Powerflex
  - SDS très complet pour étendre le CSI
  - Pas d'ampli d'IO et Overhead x4 hors compression dégressif a l'ajout de nœuds
  - Gestion de la réplication near synchrone (<15s)

# Difficultés, erreurs et solutions

- L'infra Hybride
  - VMs non pilotées par openshift
  - Réseau identique VM/Serveurs Métal
  - Aussi une solution pour compenser le stretch
- L'infrastructure Stretch (2 sites)
  - Pb de terminologie : « 2 cluster » VS « 1 cluster sur 2 salles »
  - La préco c'est 3 pas 2
  - Pb avec Ceph (OpenShift Data Foundation / Rook)
  - Pb avec Master
- Le stockage
  - Pb des baies PowerStore non adaptées
  - Pas de S3
  - Overhead énorme lié au Stretch

# Déploiement de JupyterHub

- 2 fichiers Yaml
  - Route.yaml = configuration de la route (URL) de Jupyter + reverse-proxy à utiliser
  - Values.yaml = configuration de Jupyter
- Tout le reste est déjà géré à l'échelle
  - Reverse-proxy/Ingress, Certificats, Backup, métrique, logs, etc.
  - Evidement si besoin on peut faire du spécifique
- Fonctionnement :
  - Cf la présentation « Infrastructure PaaS de l'université de Lorraine (rancher / rke2) »  
C'est exactement la même chose car c'est du K8S + ArgoCD + GitLab + Helm Charts !

# 2<sup>ème</sup> service : hosting

- Environnement de dev Web et CMS clé en main
- Dans les faits : aujourd'hui c'est des stack docker-compose (Traefik + portainer + apache + mysql + phpmyadmin + gestion fichier)
- Challenge : transposer ça proprement dans K8S !

# Next Step :

- Consolidation des procédures
  - Canary testing – Blue/Green + rollback
  - Intégration de Vault et des tests fournis par RHACS dans les pipelines
  - Modèles de chart et d'images (transposition des modèles docker-compose)
  - Affinage de la conservation des logs et métriques
- Migration d'applicatifs déjà containerisés
- Tests de KubeVirt
- Transfert de compétences et formations
  - Formation des dev par les dev + mise à disposition d'un env de dev « type »
  - Découvrir la containerisation (Docker, podman desktop, Helm Charts)
  - Utilisation à l'Université de Limoges (Pipelines, Gitops, déployer une app simple)
  - Migrer mon app dans le « Cloud » Unilim (Adaptation, surveillance, etc.)
  - Pour aller plus loin

# Questions ?

- Contact projet : [alexandre.delpeuch@unilim.fr](mailto:alexandre.delpeuch@unilim.fr)
- Contact infras : [hubert.mercier@unilim.fr](mailto:hubert.mercier@unilim.fr)
- Contact Jupyter + pratiques DevOps : [vincent.mergnac@unilim.fr](mailto:vincent.mergnac@unilim.fr)