Justification des choix techniques

1.Plateforme de virtualisation : Proxmox ou OpenStack

L'OpenStack, un système d'orchestration cloud hyperscale, nécessite des semaines de déploiement et des compétences spécialisées pour son maintien. Son architecture modulaire complexe (Nova, Neutron, Cinder, Keystone) impose un overhead opérationnel élevé et des équipes dédiées.

Proxmox offre une interface web intuitive et déployable en heures avec une courbe d'apprentissage faible. La solution combine KVM et LXC dans une stack unifiée sans dépendances lourdes, ce qui facilite la gestion de l'infrastructure virtuelle.

OpenStack s'oriente vers les infrastructures distribuées massives, Proxmox vers les environnements de taille réduite.

⇒ **La simplicité opérationnelle de Proxmox oriente le choix technique.**

2. Solution annuaire identités : OpenLDAP ou Active Directory

La solution Active Directory nécessite des licences Windows Server coûteuses, des licences d'accès par utilisateur et des renouvellements annuels, ce qui peut être un coût élevé pour les organisations. L'infrastructure Windows impose également des serveurs dédiés avec une consommation de ressources supérieure.

OpenLDAP, solution open source gratuite, offre une solution complète sans coûts de licence et de maintenance. Les fonctionnalités d'annuaire et d'authentification couvrent les besoins de gestion des identités de manière efficace. La communauté active garantit un support technique gratuit et sans contrats commerciaux.

⇒ **L'infrastructure Linux homogène oriente le choix vers OpenLDAP, solution native Linux.**

3. Authentification mutuelle : Kerberos versus authentification LDAP simple

L'authentification LDAP simple transmet les identifiants sur la connexion réseau, ce qui expose un risque de compromission par interception. C'est pourquoi Kerberos est souvent utilisé pour éliminer cette circulation des identifiants via une architecture à tickets temporisés.

Le serveur d'authentification délivre des tickets de session à durée limitée avec authentification mutuelle, permettant ainsi une sécurité renforcée. L'intégration native aux services annuaire et stockage réseau constitue un standard des infrastructures sécurisées.

⇒ **Kerberos est donc la solution d'authentification sécurisée retenue.**

4. Stockage réseau : NFS versus Samba CIFS

Dans les environnements mixtes Windows-Linux, Samba / CIFS offre une solution orientée vers la gestion des données dans un environnement multi-plateforme. Cependant, cela nécessite la gestion de métadonnées Windows inadaptées aux permissions Unix et la configuration de services supplémentaires pour l'intégration à l'annuaire.

En revanche, le protocole NFS (Network File System) est natif des environnements Unix/Linux, offrant des performances optimales, des quotas utilisateurs intégrés au système, un montage automatique à la connexion utilisateur et une compatibilité native avec les distributions Linux. L'infrastructure Linux homogène justifie le choix de NFS, simplifiant l'administration et l'exploitation.

Dans ces environnements homogènes, Samba / CIFS peut être utilisée pour stocker des données, tout en facilitant la gestion des permissions et la configuration des services nécessaires

⇒ **L'homogénéité de l'infrastructure sous Linux justifie le choix de NFS.**

5. Framework applicatif web : Flask versus Django

Django constitue un framework complet intégrant de nombreuses fonctionnalités automatiques. Cependant, cette richesse fonctionnelle apparaît disproportionnée pour une simple interface d'authentification et de création de comptes.

Flask, framework minimaliste, offre un développement rapide et ciblé. Sa structure légère et sa simplicité d'utilisation accélèrent la mise en œuvre d'une application web simple.

⇒ **Flask est retenu pour sa simplicité adaptée aux besoins.**

6. Serveur web publication : Nginx versus Apache HTTP Server

L'Apache HTTP Server, une solution historique bien connue, présente des besoins de mémoire importants sous charge. La configuration complexe via des fichiers dispersés multiples peut rendre l'administration globale difficile.

Nginx, une architecture moderne optimisée, offre des performances supérieures en termes de consommation de ressources. La configuration centralisée et la gestion automatisée des certificats SSL simplifient considérablement l'administration et l'utilisation.

⇒ **Nginx est le choix préféré pour son excellence en termes de performance et de simplicité administrative.**

7. Infrastructure as Code : OpenTofu versus Terraform HashiCorp

La licence restrictive de Terraform HashiCorp limitait l'usage commercial et la redistribution des déploiements. Les évolutions futures propriétaires génèrent une incertitude sur la pérennité des déploiements.

OpenTofu, fork communautaire de Terraform à licence open source pérenne, garantit la liberté d'usage. L'intégration complète avec les providers et modules Terraform existants est assurée. La gouvernance communautaire garantit la neutralité du développement.

⇒ **OpenTofu est retenu pour la garantie open source.**

8. Automatisation configuration : Ansible versus Puppet

Puppet nécessite des agents permanents sur chaque machine, consommant des ressources. Son langage spécifique présente une courbe d'apprentissage importante nécessitant temps et expertise. Son infrastructure de stockage des états complexifie également les mises à jour et révisions.

Ansible utilise SSH standard sans installation d'agents, ce qui simplifie considérablement les tâches de configuration. Les fichiers de configuration sont écrits en syntaxe simple, ils sont facilement accessibles et permettent une exécution garantie de la reproductibilité.

⇒ **Ansible par sa simplicité opérationnelle justifie le choix retenu.**

Conclusion

Les technologies retenues forment une stack cohérente privilégiant la sécurité avec OpenLDAP et Kerberos, les performances avec Nginx et NFS, l'agilité avec Flask et Ansible, la pérennité open source avec OpenTofu, la virtualisation avec Proxmox, et l'adaptation au périmètre d'infrastructure nœud unique.