



CAHIER DES CHARGES

Conteneurisation et gestion des workloads avec Magnum

Rédigé par:

- ESSONO EFFA Etienne Florian
- NGNINTEDEM FEUPI Vaillant

Sous la supervision : M.
NGUIMBUS Emmanuel

Année Académique 2025 - 2026



Plan

INTRODUCTION

I- CONTEXTE ET JUSTIFICATION DU PROJET

II- CIBLE ET BENEFICIAIRES

III- OBJECTIFS DU PROJET

IV- EXPRESSION DES BESOINS

V- RESSOURCES UTILISÉES

VI- PLANIFICATION

CONCLUSION



INTRODUCTION

Ce projet vise à déployer et automatiser la gestion de clusters Kubernetes et Docker Swarm via Magnum dans un environnement OpenStack, tout en intégrant l'orchestration d'infrastructures hybrides combinant machines virtuelles et conteneurs grâce à Heat. Cette solution répond aux besoins croissants de flexibilité, de scalabilité et d'optimisation des ressources dans les environnements cloud modernes.



I- CONTEXTE ET JUSTIFICATION DU PROJET

Contexte Actuel

- **Croissance exponentielle** des workloads conteneurisés
- **Complexité accrue** de gestion des infrastructures hybrides
- **Besoin d'automatisation** des déploiements et de la gestion des clusters
- **Défis de sécurité** et d'isolation dans les environnements multi-locataires

Justification

- **Réduction des coûts** via l'automatisation et l'optimisation des ressources
- **Amélioration de la productivité** des équipes DevOps
- **Standardisation** des environnements de conteneurisation
- **Flexibilité accrue** pour le déploiement d'applications modernes
- **Optimisation** de l'utilisation des ressources cloud



II- CIBLES ET BÉNÉFICIAIRES

Publics Cibles Principaux

- **Administrateurs systèmes et cloud**
- **Ingénieurs DevOps et SRE**
- **Équipes de développement d'applications**
- **Responsables d'infrastructure informatique**

Bénéficiaires Finaux

- **Services informatiques des entreprises**
- **Prestataires de services cloud**
- **Institutions académiques et de recherche**
- **Startups et PME technologiques**



III- OBJECTIFS DU PROJET

Objectif Général

Déployer et automatiser la gestion de clusters conteneurisés via Magnum avec intégration complète de l'orchestration hybride VM/conteneurs grâce à Heat.



III- OBJECTIFS DU PROJET

Objectifs Spécifiques

1. **Déploiement Automatisé** de clusters Kubernetes et Docker Swarm
2. **Intégration Native** entre Magnum et Heat pour l'hybridation
3. **Automatisation Complète** du cycle de vie des clusters
4. **Mise en place** d'un système de monitoring et scaling automatique
5. **Sécurisation** des environnements conteneurisés
6. **Création de Templates** réutilisables pour déploiements standards
7. **Documenter et livrer scripts et playbooks**



III- OBJECTIFS DU PROJET

Critères de réussite

- **Cluster de conteneurs opérationnel et accessible.**
- **Orchestration Heat déployant VMs et un cluster Magnum.**
- **Déploiement d'une application de test dans l'environnement hybride.**



IV- EXPRESSION DES BESOINS - Vu d'ensembles des services OpenStack

Qu'est-ce qu'OpenStack ?

OpenStack est une plateforme de cloud computing **open source** qui permet de gérer des pools de ressources de calcul, de stockage et de réseau au sein d'un centre de données. On l'utilise principalement pour déployer des **clouds privés** ou **publics (IaaS)**.

Les Services de Base (Fonctionnalités Principales)

1. Nova (Compute) - Le Cœur du Calcul

C'est le service qui gère et automatise le provisionnement des machines virtuelles.

- **Fonctionnalités :**
 - Création, suppression, gestion et redimensionnement de serveurs virtuels (instances).
 - Supporte plusieurs technologies de virtualisation (hyperviseurs) comme KVM, VMware, Hyper-V, etc.
 - Répartit automatiquement les nouvelles instances sur les serveurs physiques disponibles.
 - Gère la mise en réseau de base des instances via Neutron.



IV- EXPRESSION DES BESOINS

2. Neutron (Networking) - La Mise en Réseau

Fournit un "réseau défini par le logiciel" (SDN) pour connecter les instances entre elles et vers l'extérieur.

- **Fonctionnalités :**

- Création et gestion de réseaux privés, sous-réseaux et routeurs virtuels.
- Attribution d'adresses IP flottantes (publiques) aux instances.
- Gestion des groupes de sécurité (firewalls virtuels) pour contrôler le trafic entrant et sortant des instances.
- Création de topologies de réseau complexes (VLANs, VXLANs).



IV- EXPRESSION DES BESOINS

3. Cinder (Block Storage) - Le Stockage en Bloc

Offre un système de stockage persistant de type "bloc" (comme un disque dur supplémentaire) pour les instances.

- **Fonctionnalités :**
 - Création de volumes (disques durs virtuels) qui peuvent être attachés et détachés des instances à la volée.
 - Permet de sauvegarder ces volumes via des snapshots.
 - Supporte de nombreux backends de stockage (Ceph, LVM, NetApp, Dell EMC, etc.).
 - Essentiel pour conserver les données même si l'instance est supprimée.



IV- EXPRESSION DES BESOINS

4. Swift (Object Storage) - Le Stockage d'Objets

Service de stockage scalable et redondant pour stocker de grands volumes de données non structurées (objets).

- **Fonctionnalités :**

- Stockage de fichiers (objets) avec une réplication automatique sur plusieurs nœuds pour une haute durabilité.
- Accès via une API RESTful (similaire à Amazon S3).
- Pas de structure de répertoire classique ; les objets sont stockés dans des conteneurs.
- Idéal pour les sauvegardes, les archives, la distribution de contenus (images, vidéos) et les datasets de Big Data.



IV- EXPRESSION DES BESOINS

5. Glance (Image Service) - Le Catalogue d'Images

Gère les images de disques virtuels qui servent de modèles pour créer de nouvelles instances.

- **Fonctionnalités :**

- Stocke et catalogue des images de systèmes d'exploitation (ex: Ubuntu, CentOS, Windows).
- Permet de créer des snapshots d'instances existantes pour en faire de nouvelles images.
- Fournit les images aux services de calcul (Nova) lors du lancement d'une instance.



IV- EXPRESSION DES BESOINS

6. Keystone (Identity Service) - L'Authentification et l'Autorisation

Le service d'identité central qui fait office de portier pour tout l'écosystème OpenStack.

- **Fonctionnalités :**
 - Authentification des utilisateurs et des services (création de tokens d'accès).
 - Gestion des droits d'accès via un système de rôles et de permissions (RBAC).
 - Catalogue de services : fournit la liste de tous les services OpenStack disponibles et leurs points de terminaison (URLs API).



IV- EXPRESSION DES BESOINS

7. Horizon (Dashboard) - L'Interface Web

Le tableau de bord web qui offre une interface graphique pour administrer et utiliser OpenStack.

- **Fonctionnalités :**
 - Permet aux administrateurs et aux utilisateurs de gérer les ressources sans utiliser la ligne de commande.
 - Opérations visuelles pour lancer des instances, gérer le stockage, configurer le réseau, etc.
 - Fournit une vue d'ensemble sur l'utilisation et l'état du cloud.



Ressources ajoutés

1. OpenStack Magnum : Le "Chef d'Orchestre" des Clusters

Magnum est le service OpenStack qui permet de fournir des **COE (Container Orchestration Engines)** comme Kubernetes ou Docker Swarm en tant que ressources de premier plan (comme le sont les instances VM).

Ce qu'il faut savoir :

- **Abstraction** : Magnum nous évite d'installer manuellement Kubernetes sur des VMs. On lui demande un "Cluster", et il s'occupe de tout.
- **Modèles (ClusterTemplates)** : C'est la "recette" qui définit l'image OS (ex: Fedora CoreOS), la taille des nœuds, le réseau et le moteur (K8s/Swarm).
- **Multi-tenance** : Contrairement à un Kubernetes classique, Magnum isole nativement les clusters par projet OpenStack (via Keystone).

Son importance pour ce projet : C'est l'interface principale qui répond au premier objectif. Il simplifie le déploiement massif et permet de standardiser les clusters de conteneurs.



Ressources ajoutés

2. OpenStack Heat : Le "Bâtisseur" de l'Infrastructure

Heat est le moteur d'orchestration d'OpenStack. Il utilise le concept d'**Infrastructure as Code (IaC)** via des fichiers YAML appelés **HOT (Heat Orchestration Templates)**.

Ce qu'il faut savoir :

- **Automatisation Totale** : Heat peut créer en une seule commande : le réseau (Neutron), le stockage (Cinder), les routeurs, et les instances (Nova).
- **Gestion des Stacks** : Un ensemble de ressources liées est appelé une "Stack". Si une ressource échoue, Heat peut revenir en arrière (rollback).
- **Interopérabilité** : Heat "comprend" toutes les ressources OpenStack, y compris les ressources Magnum.

Son importance pour ce projet : C'est l'outil qui va réaliser le deuxième objectif (l'hybridation). C'est lui qui permet de dire : *"Déploie-moi une VM de base de données classique ET un cluster Kubernetes Magnum dans le même réseau."*



Ressources ajoutés

3. L'interaction Magnum-Heat : Pourquoi c'est la clé ?

Dans ce projet, Magnum ne "crée" pas directement de serveurs. Lorsqu'on lance un cluster via Magnum, celui-ci génère en arrière-plan un template Heat.

Pourquoi cette intégration est vitale pour atteindre les objectifs :

1. **Orchestration Hybride (VM + Conteneurs) :**
 - Grâce à Heat, on pourra écrire un template unique qui définit à la fois des **instances Nova** (pour des applications legacy ou des bases de données lourdes sur VM) et un **cluster Magnum** (pour les microservices).
 - Cela garantit que les deux mondes partagent le même réseau privé et les mêmes règles de sécurité dès le déploiement.



Ressources ajoutés

Automatisation et Scaling :

- Heat gère le cycle de vie. Si vous avez besoin d'étendre votre cluster, Magnum communique avec Heat pour ajouter de nouvelles ressources d'infrastructure (VMs) de manière cohérente.

Fiabilité :

- En utilisant Heat comme moteur, Magnum bénéficie de la robustesse d'un outil éprouvé pour la gestion des dépendances entre les ressources (par exemple, ne pas lancer le cluster tant que le réseau n'est pas prêt).

IV- EXPRESSION DES BESOINS



Tableau Récapitulatifs

Service	Nom de Code	Rôle Principal	Analogie Cloud Public
Compute	Nova	Gérer les machines virtuelles	Amazon EC2
Networking	Neutron	Gérer les réseaux virtuels	Amazon VPC
Block Storage	Cinder	Fournir des disques durs virtuels	Amazon EBS
Object Storage	Swift	Stocker des fichiers/objets à grande échelle	Amazon S3
Image Service	Glance	Gérer les modèles de machines virtuelles	Amazon AMI
Identity	Keystone	Authentifier et autoriser les utilisateurs	AWS IAM
Dashboard	Horizon	Interface d'administration web	AWS Management



IV- EXPRESSION DES BESOINS

Besoins Fonctionnels

- **Gestion des Clusters** : Création, modification, suppression via API
- **Orchestration Hybride** : Gestion combinée VMs et conteneurs
- **Monitoring Intégré** : Supervision des performances et santé des clusters
- **Sauvegarde Automatique** : Backup des configurations et données
- **Scaling Dynamique** : Ajustement automatique des ressources
- **Interface Unifiée** : Management via interface web et CLI



IV- EXPRESSION DES BESOINS

Besoins Non Fonctionnels

- **Performance** : Temps de déploiement < 15 minutes
- **Disponibilité** : 99% pour les services critiques
- **Sécurité** : Isolation réseau, authentification forte, chiffrement
- **Scalabilité** : Support jusqu'à 100 nœuds par cluster
- **Maintenabilité** : Architecture modulaire et documentée
- **Interopérabilité** : Compatibilité avec les standards OpenStack



IV- EXPRESSION DES BESOINS

Périmètre fonctionnel

Nous nous attarderons le plus sur :

- Création de modèles de clusters (cluster templates).
- Provisionnement de clusters (masters/nodes) et récupération des credentials (**kubeconfig**) pour l'accès au cluster afin de mieux **Compréhension du cycle de vie (crée des VM, configure Kubernetes, génère les accès (kubeconfig).)**.
- Orchestration via Heat : dans le but de Comprendre la **logique d'orchestration (Syntaxe Heat (HOT), Dépendances entre ressources, Déploiement coordonné (VM simple, Cluster Magnum).)**.
- Automatisation via CLI, Ansible et/ou Terraform.
- Tests de montée en charge (scale up/scale down), tests de tolérance de panne. Monitoring de base (Prometheus + Grafana ou équivalent), logs centralisés



V- RESSOURCES UTILISÉES

Ressources Humaines

Rôle	Nombre de jours	Taux journalier (FCFA)	Coût Total (FCFA)
Architecte Cloud	10 jours	300 000	3 000 000
Administrateur OpenStack	50 jours	250 000	12 500 000
Ingénieur DevOps	50 jours	250 000	12 500 000
Expert Sécurité	5 jours	350 000	1 750 000
Chef de Projet	15 jours	250 000	3 750 000
Total Ressources Humaines	130 jours	Moyenne: 280 000/jour	33 500 000



V- RESSOURCES UTILISÉES

Ressources Matérielles et Logicielles

Ressource	Spécifications	Coût (FCFA)
Serveurs Compute	8 vCPU, 32GB RAM × 3	884 250
Stockage Cinder	5 TB SSD	589 500
Réseau et Load Balancing	VLANs dédiés	294 750
Backup Storage	2 TB object storage	196 500
Total Infrastructure		1 965 000



V- RESSOURCES UTILISÉES

Budget Consolidé

Poste Budgétaire	Montant (FCFA)	Pourcentage
Ressources Humaines	33 500 000	82,13%
Infrastructure Cloud	1 965 000	4,81%
Imprévus (15%)	5 319 750	13,04%
Total Général	40 784 750	100%



VI- PLANIFICATION

Phases et Délais

Phase	Durée	Période	Livrables Principaux
Phase 1 : Préparation	2 semaines	J1 à J14	Environnement OpenStack opérationnel
Phase 2 : Implémentation	4 semaines	J15 à J42	Clusters K8s/Swarm déployables
Phase 3 : Intégration	2 semaines	J43 à J56	Intégration Magnum-Heat fonctionnelle
Phase 4 : Validation	2 semaines	J57 à J70	Solution complète validée



Livrables

- Cahier des charges (ce document).
- Guide d'installation et scripts.
- Exemples Heat templates montrant orchestration VM + cluster.
- Playbooks pour monitoring, backups et mise à l'échelle.
- Rapport de tests.



CONCLUSION

Ce cahier des charges définit les bases solides pour la mise en œuvre d'une solution de conteneurisation avancée avec Magnum et Heat. Le budget total de **40 784 750 FCFA** sur une période de **10 semaines** permettra de livrer une solution complète d'orchestration hybride qui répondra aux besoins croissants de flexibilité et d'automatisation dans les environnements cloud modernes.

La solution proposée apportera une valeur significative grâce à l'automatisation des déploiements, l'optimisation des ressources et la standardisation des environnements, positionnant ainsi les organisations bénéficiaires à l'avant-garde des technologies cloud natives.