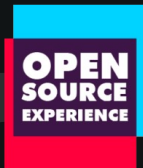


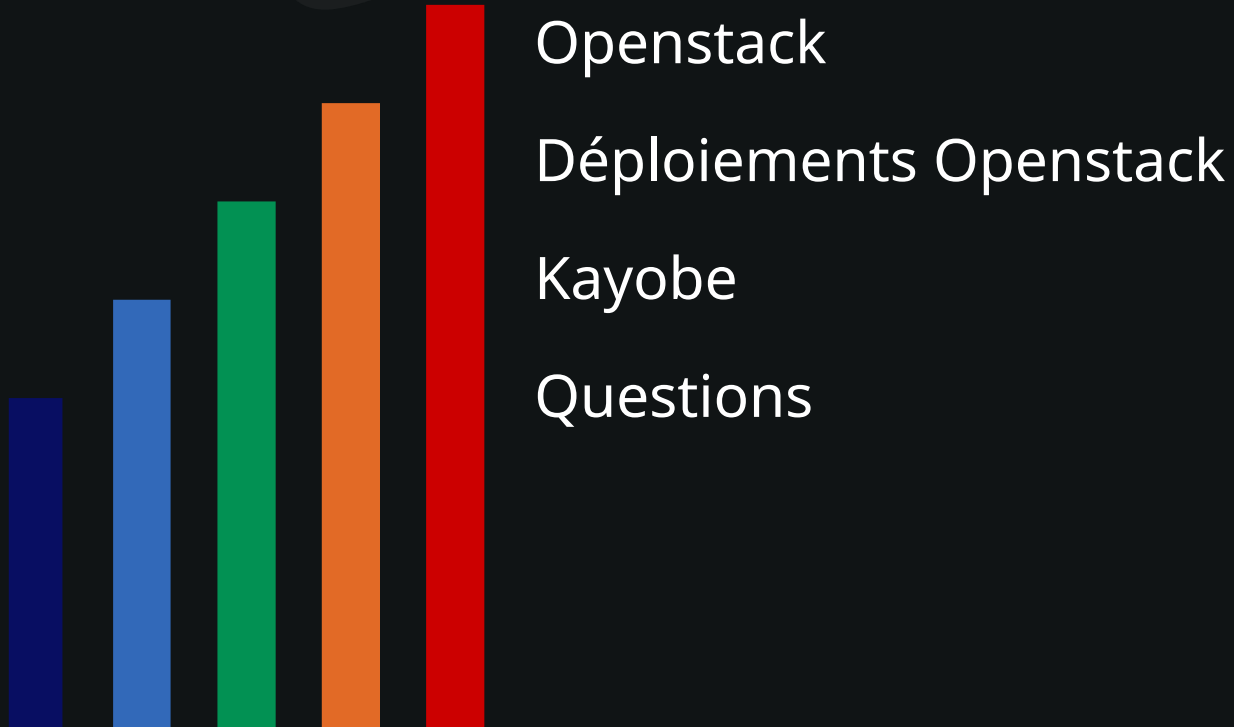


# Déploiement OpenStack via KAYOBE



10 décembre 2025  
Paris

# Sommaire





# OpenStack

une plateforme modulaire

# OpenStack

## une plateforme IaaS modulaire et composable

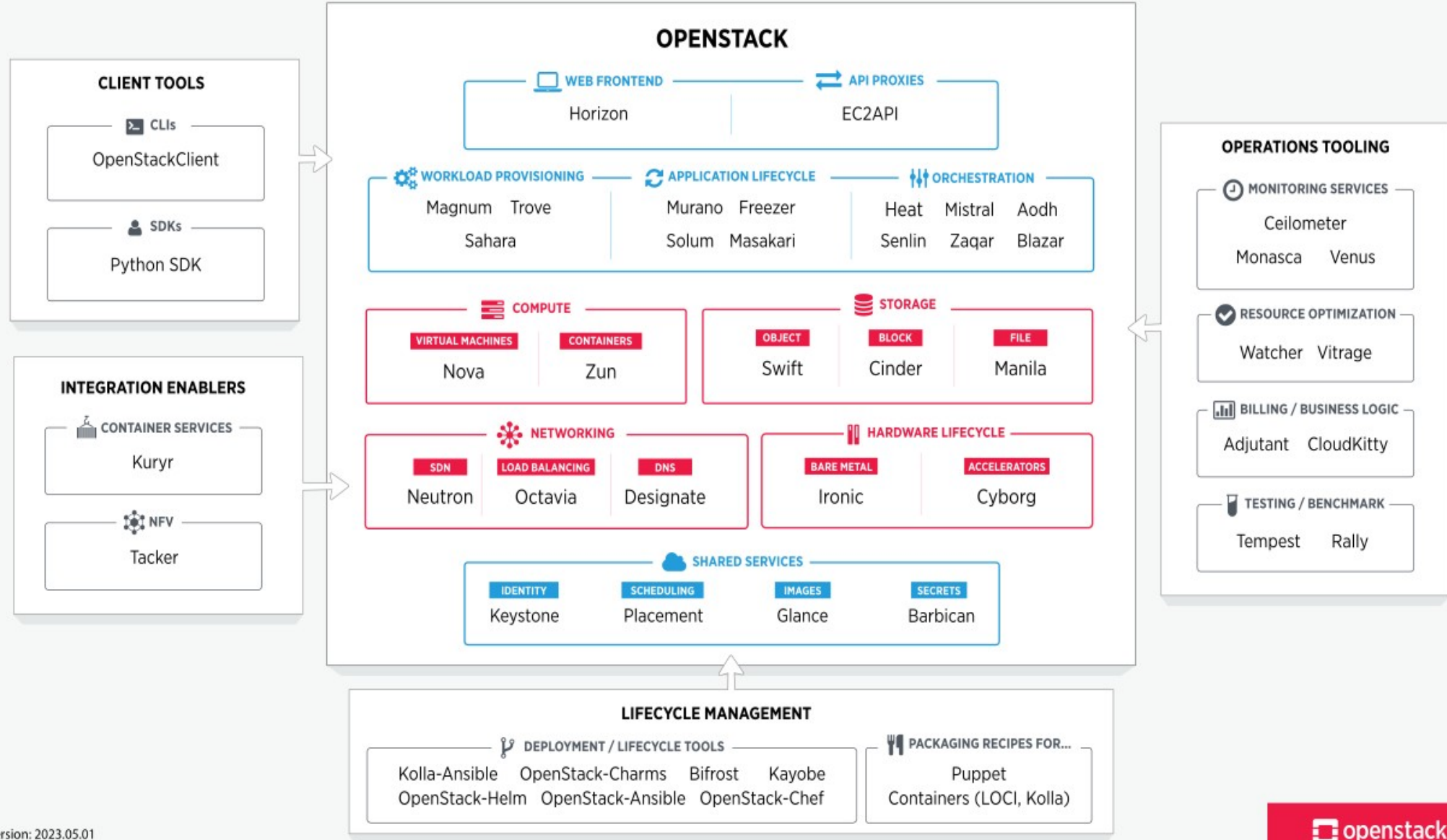
OpenStack n'est pas un produit monolithique, mais un ensemble de services interconnectés, qui s'assemblent en fonction du besoin.

Selon l'usage (hébergement privé, edge, cloud, telco, HPC...), chaque déploiement combine différemment les briques.

### Exemples de briques clés :

- **Compute** : Nova, placement, libvirt, QEMU, KVM
- **Stockage** : Cinder, Swift, backend Ceph, NFS, LVM
- **Réseau** : Neutron, ML2, OVN, Open vSwitch
- **Identité & authN/authZ** : Keystone
- **Services opérationnels** : Heat, Octavia, Magnum, Ironic





# Architecture OpenStack structurée par rôles

Pour assurer disponibilité, performance et évolutivité, les déploiements OpenStack s'appuient sur une séparation fonctionnelle et claire :

**1. Plan de contrôle** (Control Plane) → Déployé sur un ou plusieurs nœuds selon HA :

- Base de données (MariaDB/Galera)
- Message broker (RabbitMQ)
- Load balancer (HAProxy, Keepalived)
- APIs des différents services OpenStack

**2. Plans de ressources** (Data Plane) : machines dédiées aux rôles :

- Hyperviseurs (compute)
- Nœuds de stockage (Swift, Cinder, Ceph)
- Nœuds réseau selon l'architecture (Neutron/ML2/OVN)

*Cette séparation par rôle est un prérequis pour automatiser le déploiement via Kayobe ou d'autres outils*





# Déploiements

# OpenStack

de l'installation  
manuelle à l'automatisation complète

# Les premiers déploiements

À l'origine, les déploiements OpenStack étaient entièrement manuels :

- Installation du système d'exploitation sur chaque machine
- Ajout des paquets requis pour chaque service
- Configuration locale des fichiers et paramètres
- Mise en place de scripts pour lancer les services au démarrage
- Toute modification nécessitait une action manuelle sur les nœuds



# Automatisation du déploiement

Avec la montée en charge des infrastructures, l'automatisation est devenue indispensable :

- Réduction des temps de déploiement sur des parcs de machines de plus en plus grands
- Homogénéité des configurations sur l'ensemble des nœuds
- Simplification des mises à jour et des changements de configuration
- Apparition des premiers outils d'automatisation : Puppet, puis Ansible



**puppet**



ANSIBLE

# Un peu de conteneurs ?

L'usage des conteneurs a marqué une nouvelle étape dans les déploiements OpenStack :

- Remplacement des services installés sur le système par des conteneurs
- Portabilité facilitée entre distributions et environnements (inter OS-distribution)
- Isolation accrue des composants et dépendances
- Déploiement et mises à jour plus rapides et mieux maîtrisés



# Kolla la conteneurisation d'OpenStack

Le projet Kolla a introduit une approche standardisée des services OpenStack en conteneurs :

- Projet intégré à la communauté OpenStack
- Fourniture d'images conteneurisées prêtes pour la production
- Basé sur Docker et pensé pour un usage en cluster
- Framework de construction d'images pour les services OpenStack et l'infrastructure associée



# KOLLA

*an OpenStack Community Project*



# Évolution des outils de déploiement OpenStack

L'émergence des communautés OpenStack a fait apparaître des approches plus structurées du déploiement :

- Mutualisation des recettes et des architectures éprouvées
- Standardisation des bonnes pratiques entre intégrateurs et opérateurs
- Montée en puissance des solutions de clustering pour le plan de contrôle
- Conteneurisation progressive des services pour faciliter l'exploitation



# TripleO l'ère d'OpenStack « par OpenStack »

Pendant plusieurs années, TripleO a été l'une des solutions majeures pour déployer OpenStack en production :

- Déploiement basé sur l'approche OpenStack on OpenStack (OOO)
- Installation, configuration et mises à jour gérées via les services OpenStack
- Intégration progressive des conteneurs Kolla dans les versions récentes
- Fort usage via RHOSP, jusqu'à l'abandon par Red Hat puis RDO
- Dernière version maintenue : Wallaby
- Technologie historiquement utilisée par Worteks pour ses premiers clusters



# Kolla Ansible

## la conteneurisation maîtrisée par OpenStack

Kolla Ansible a apporté une approche automatisée et modulaire du déploiement des services OpenStack :

- Déploiement des services OpenStack sous forme de conteneurs via Ansible
- Possibilité de construire un environnement personnalisé et adapté aux besoins
- Forte adoption dans les clusters de production (22% en 2023, 37% en 2025)
- Complément nécessaire pour la partie provisioning et configuration de système





# Kayobe

une nouvelle génération de  
déploiement OpenStack

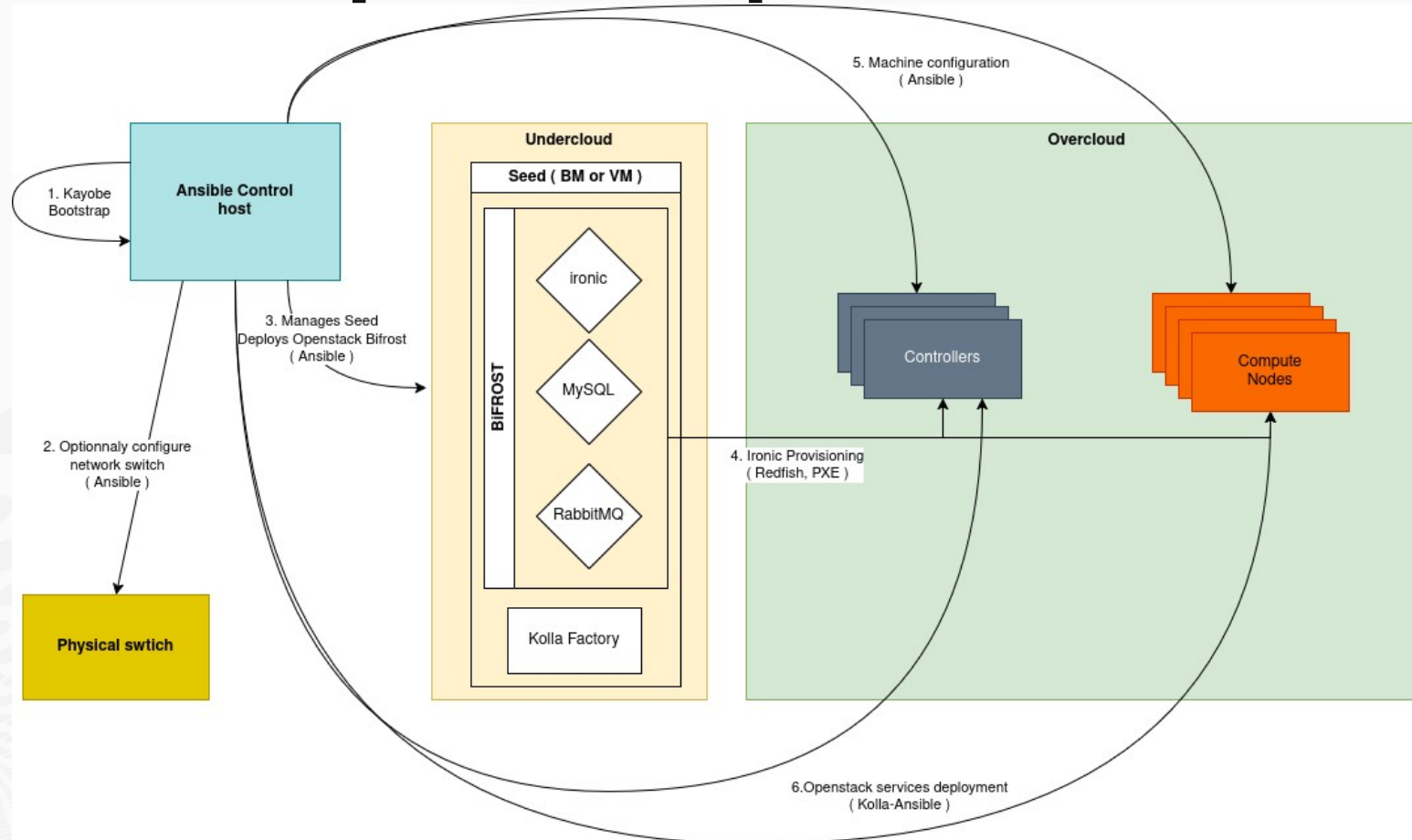
# Kayobe une solution tout-en-un, basée sur Kolla et Bifrost

Kayobe s'appuie sur une pile d'outils OpenStack pour couvrir l'ensemble du cycle d'installation :

- Bifrost / Ironic pour le déploiement des systèmes d'exploitation
- Ansible pour la configuration réseau et machine
- Kolla-Ansible pour l'installation des services OpenStack en conteneurs
- Supervision intégrée (logs, monitoring) selon la stack retenue
- Approche automatisée du provisioning jusqu'aux services



# Cinématique de déploiement



# Architecture réseau

## un déploiement pensé par rôles

Kayobe permet de définir et automatiser une architecture réseau cohérente et reproductible :

- Séparation des plans réseau (API, gestion, stockage, data, external...)
- Configuration automatique des interfaces selon les rôles des nœuds
- Prise en charge des VLAN, bonds, MTU, routage et overlay si nécessaire
- Intégration avec Neutron, OVN ou OVS selon l'architecture cible
- Gestion cohérente des passerelles, IP flottantes et VIP





# Questions



[www.worteks.com](http://www.worteks.com)

✉ [info@worteks.com](mailto:info@worteks.com)

☎ +33 1 84 20 86 47

🌐 [worteks\\_com](http://worteks_com)

🌐 [in worteks](https://www.linkedin.com/company/worteks)