Sécurité des infrastructures de virtualisation

Timothée Ravier
siosm@floss.social - tim.siosm.fr/cours - github.com/travier

5e année cycle ingénieur, filière STI
Option Sécurité des Systèmes Embarqués et du Cloud (2SEC)

2024 - 2025

Infrastructures ?

Infrastructure de type Cloud

> Ensemble de **services** qui fournissent aux **utilisateurs** un **environnement** pour déployer des **applications** sous différentes formes (**machines virtuelles**, **conteneurs**, **fonctions**, etc.).

Self service

> Chaque utilisateur peut choisir ce dont il a besoin parmis les services proposés (stockage, réseau, base de données, gestion de charge, etc.).

Responsabilités séparées

> Les services fournis par l'infrastructure ne sont pas administrés par les utilisateurs de l'infrastructure.

Services fournis par une infrastructure (1)

- Compute :
  - Machines virtuelles
  - Conteneurs
  - Fonctions (« serverless »)
- Stockage :
  - Stockage d'images disques (block storage)
  - Stockage d'images de conteneurs (container registry)
  - Stockage d'objets / blocs de données (object storage)
  - Stockage persistant (NFS, etc.)

Services fournis par une infrastructure (2)

- Réseau :
  - Adresses IP, routage, réseau privées virtuels
  - DNS, noms de domaine
  - Load Balancing
  - Enregistrement et découverte de services
- Gestion des identités, gestion des secrets
- Bases de données (PostgreSQL, MariaDB, NoSQL, etc.)

Services fournis par une infrastructure (3)

- Intégration Continue (CI) / Déploiement continue (CD) :
  - Compilation, test, etc.
  - Construction d'images de conteneurs
- Supervision :
  - Métriques (performance)
  - Inventaire et audit
- Intégration avec du matériel (routeurs, stockage, etc.)
- Intégration avec les services de Cloud providers

Besoins en automatisation et orchestration

- Automatisation essentielle pour les grands déploiements
- Gestion organisée et planifiée :
  - de l'infrastructure
  - de la disponibilité
  - des applications
  - de la répartition des ressources
- Objectifs :
  - Déléguer les tâches ingrates à des services
  - Minimiser les temps de panne et augmenter la réactivité

Quels logiciels pour une infrastructure de virtualisation ?

Exemples d'infrastructures

- Cloud providers Américains :
  - Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, IBM Cloud, etc.
- Cloud providers Européens :
  - OVHcloud, Scaleway, Hetzner, IONOS (1&1), Clever Cloud, etc.
- Cloud providers Asiatiques :
  - Alibaba Cloud, Tencent Cloud, Baidu, etc.
- Cloud providers Russes :
  - Yandex Cloud, etc.

Mettre en place une infrastructure

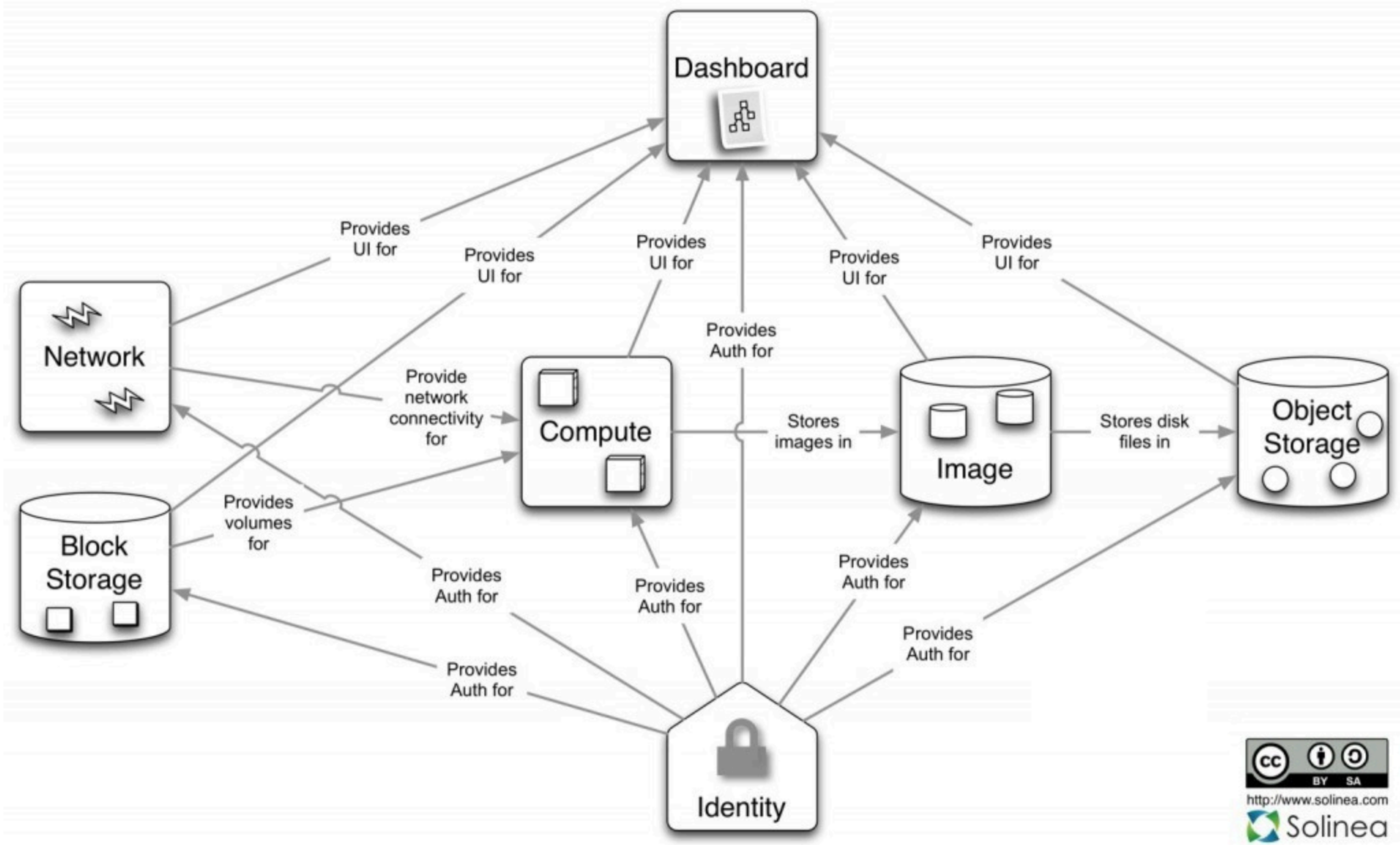
Nombreux projets proposant un ensemble de logiciels pour mettre en place sa propre infrastructure, dans un environnement public (Cloud provider) ou privé (datacenter) :

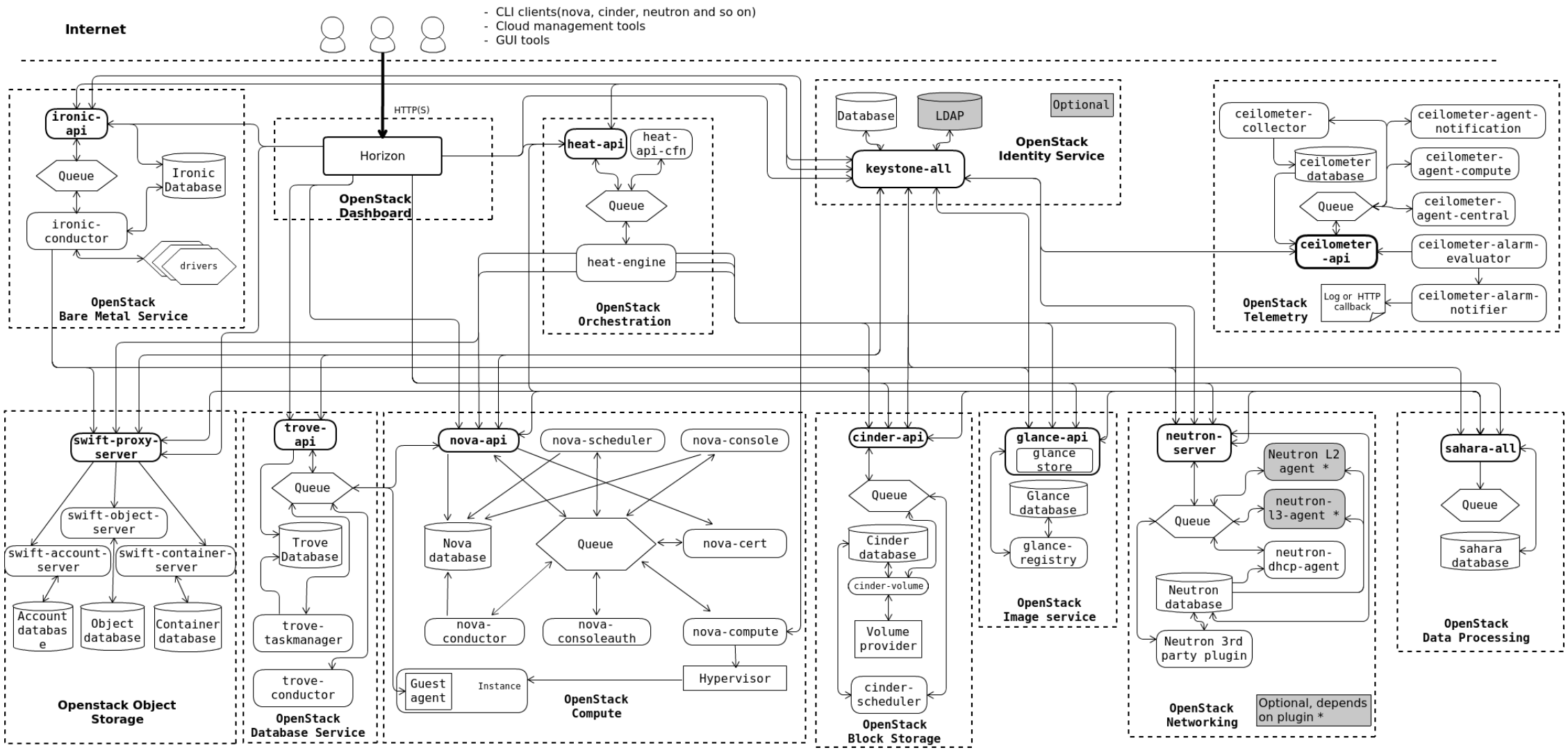
- Infrastructure orientée virtualisation :
  - OpenStack
  - VMware (vSphere ESXi, vCenter, etc.) (racheté par Broadcom)
  - Nutanix, oVirt, Apache CloudStack, Eucalyptus, OpenNebula, etc.
- Infrastructure orientée conteneurs :
  - Kubernetes (et dérivés : OpenShift, etc.)
  - Hashicorp Nomad (n'est plus open source depuis Août 2023)
  - Mesos, Docker Machine, Cloud Foundry, etc.

Openstack

OpenStack

- Projet lancé en 2010 par Rackspace et la NASA
- Ensemble de composant pour gérer automatiquement le déploiement de machines virtuelles ou de conteneurs
- Nombreux services disponibles
- Intégration avec le matériel réseau, stockage, etc.


Dashboard
Network
Block Storage
Compute
Image
Object Storage
Identity
Provides UI for
Provides UI for
Provides UI for
Provides UI for
Provides UI for
Provide network connectivity for
Provides volumes for
Provides Auth for
Provides Auth for
Provides Auth for
Provides Auth for
Provides Auth for
Stores images in
Stores disk files in
http://www.solinea.com
Solinea


Internet
- CLI clients(nova, cinder, neutron and so on)
- Cloud management tools
- GUI tools
HTTP(S)
ironic-api
Queue
Ironic Database
ironic-conductor
drivers
OpenStack Bare Metal Service
Horizon
OpenStack Dashboard
heat-api
heat-api-cfn
Queue
heat-engine
OpenStack Orchestration
Database
LDAP
Optional
keystone-all
OpenStack Identity Service
ceilometer-collector
ceilometer-agent-notification
ceilometer database
ceilometer-agent-compute
Queue
ceilometer-agent-central
ceilometer-api
ceilometer-alarm-evaluator
Log or HTTP callback
ceilometer-alarm-notifier
OpenStack Telemetry
swift-proxy-server
swift-object-server
swift-account-server
swift-container-server
Account database
Object database
Container database
Openstack Object Storage
trove-api
Queue
Trove Database
trove-taskmanager
trove-conductor
OpenStack Database Service
nova-api
nova-scheduler
nova-console
Nova database
Queue
nova-cert
nova-conductor
nova-consoleauth
nova-compute
Hypervisor
Guest agent
Instance
OpenStack Compute
cinder-api
Queue
Cinder database
cinder-volume
Volume provider
cinder-scheduler
OpenStack Block Storage
glance-api
glance store
Glance database
glance-registry
OpenStack Image service
neutron-server
Queue
Neutron L2 agent *
neutron-l3-agent *
neutron-dhcp-agent
Neutron database
Neutron 3rd party plugin
OpenStack Networking
Optional, depends on plugin *
sahara-all
Queue
sahara database
OpenStack Data Processing

Sécurité OpenStack

Ressources :

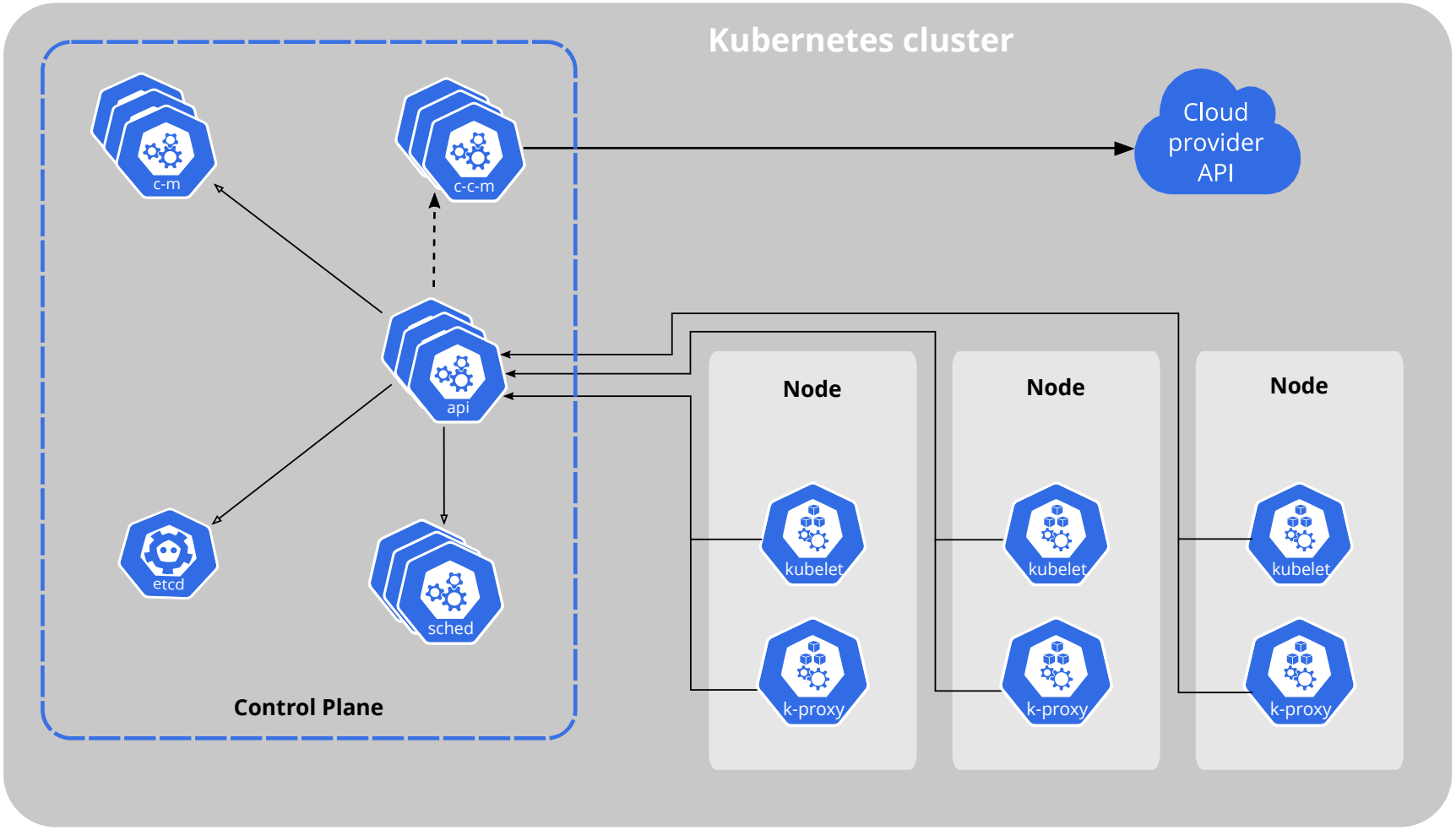
- There’s Real Magic behind OpenStack Neutron
- OpenStack logical architecture
- OpenStack Security Guide
- OpenStack Administrator Guide
- Security and Hardening Guide (Red Hat)

Kubernetes

Kubernetes

- Projet lancé par des ingénieurs de chez Google en 2014
- Inspiré de *Borg*, le gestionnaire de conteneur interne à Google
- Automatisation du déploiment, passage à l'échelle et gestion des applications conteneurisées

A Technical Overview of Kubernetes, Brendan Burns, CoreOS Fest 2015



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

kube-proxy

Scheduler

Control plane

Node

https://kubernetes.io/docs/concepts/overview/components/

Noeuds (*Nodes*)

- Machines (virtuelles ou non) qui composent notre cluster
- Fournit le CPU, la mémoire, le stockage
- Indifférenciées par défaut
- Agent sur chaque noeud : kubelet
- Container runtime : containerd, CRI-O

Documentation Kubernetes : Nodes

Pods

- Groupe de conteneurs partageant certains namespaces : réseau, IPCs
- Chaque Pod a une addresse IP unique au cluster

- Documentation Kubernetes : Pods
- Historique : App Container Specification: App Container Pods

Scheduler

- Assigne les pods aux noeuds

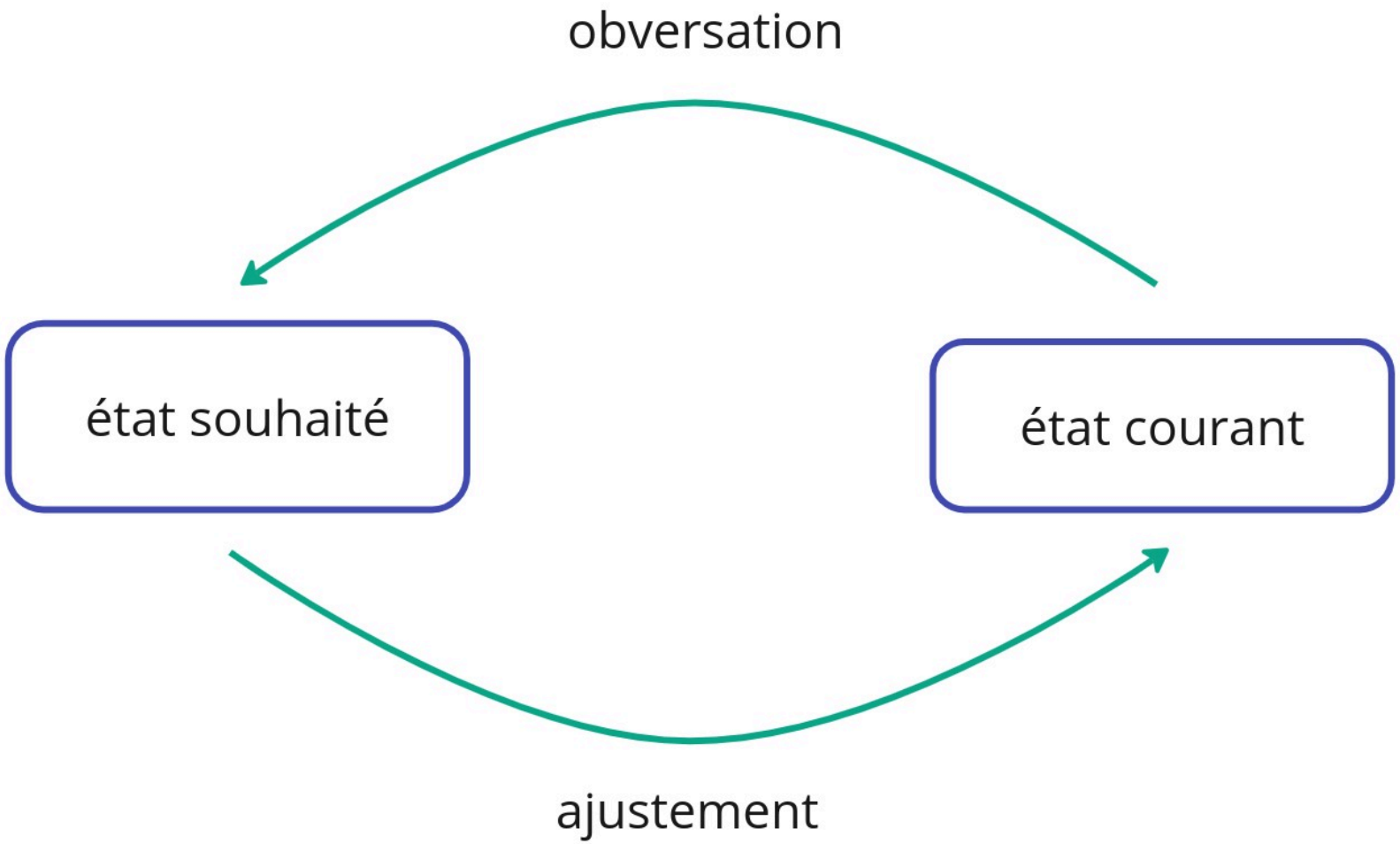
- Documentation Kubernetes : Scheduler
- Everything You Ever Wanted to Know About Resource Scheduling, But Were Afraid to Ask by Tim Hockin

etcd

- Base de données clé / valeur distribuée
- etcd.io
- Stocke la configuration et l'état du cluster

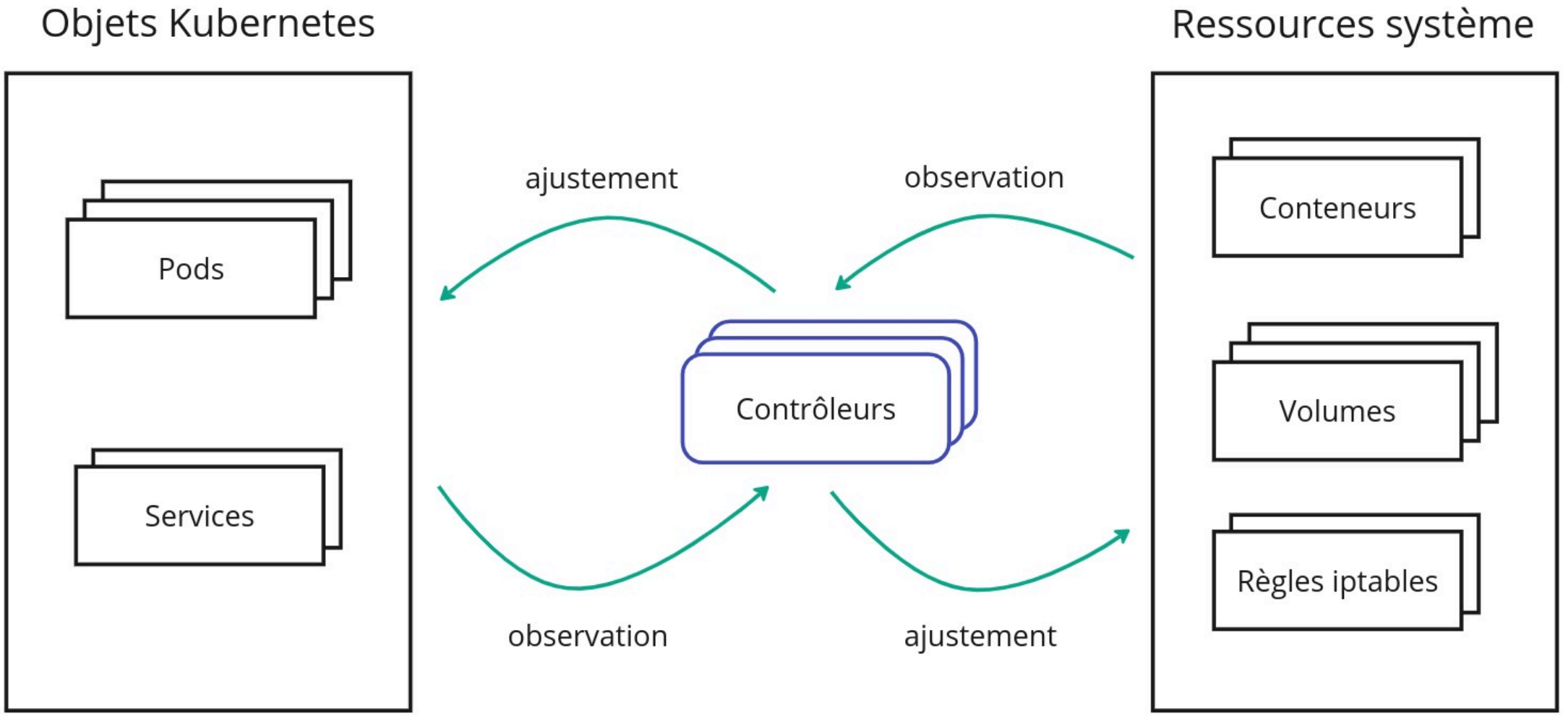
- Documentation Kubernetes : etcd

Boucle de réconciliation (*Reconciliation loop*)



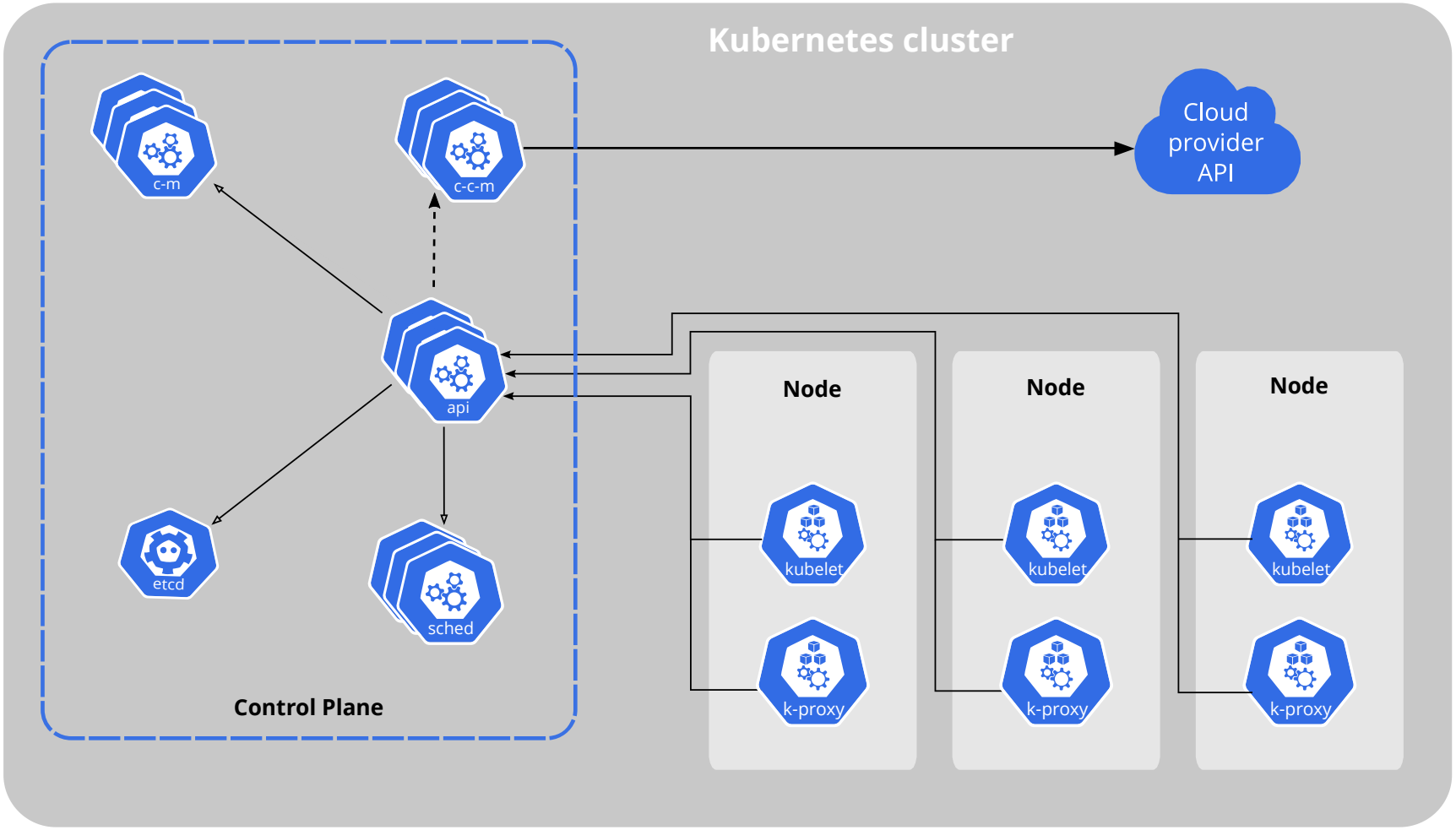
- Kubernetes: What is "reconciliation"? Tim Hockin

Contrôleurs (*Controllers*)



Operateurs (*Operators*)

- Même fonctionnement qu'un contrôleur
- Avec des objets non natifs à Kubernetes (base de données, application)
- Gère le déploiement dans un cluster
- Objectif : automatiser l'administration, la mise à jour et la maintenance d'une application ou d'un service
- OperatorHub



API server

Cloud controller manager *(optional)*

Controller manager

etcd (persistence store)

kubelet

kube-proxy

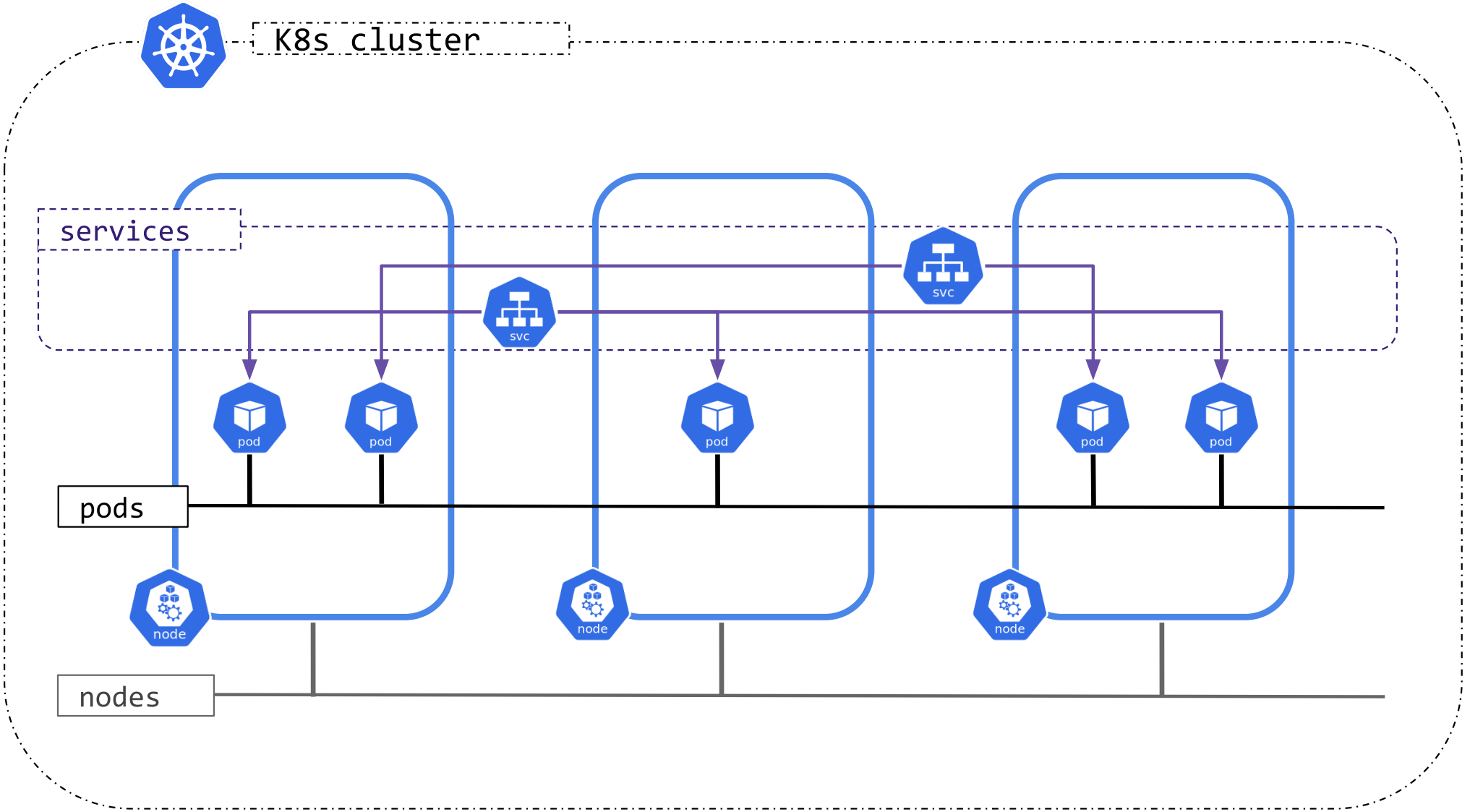
Scheduler

Control plane

Node

https://kubernetes.io/docs/concepts/overview/components/

Gestion du réseau



Gestion du réseau

- Container Network Interface (CNI) :
  - flannel (L3, vxlan)
  - Cilium (L3-L7, HTTP, eBPF)
  - Project Calico (eBPF, vxlan)
  - OVN-Kubernetes (Open vSwitch)
  - Kube-OVN (Open vSwitch)
  - etc.

- Kubernetes Cluster Networking

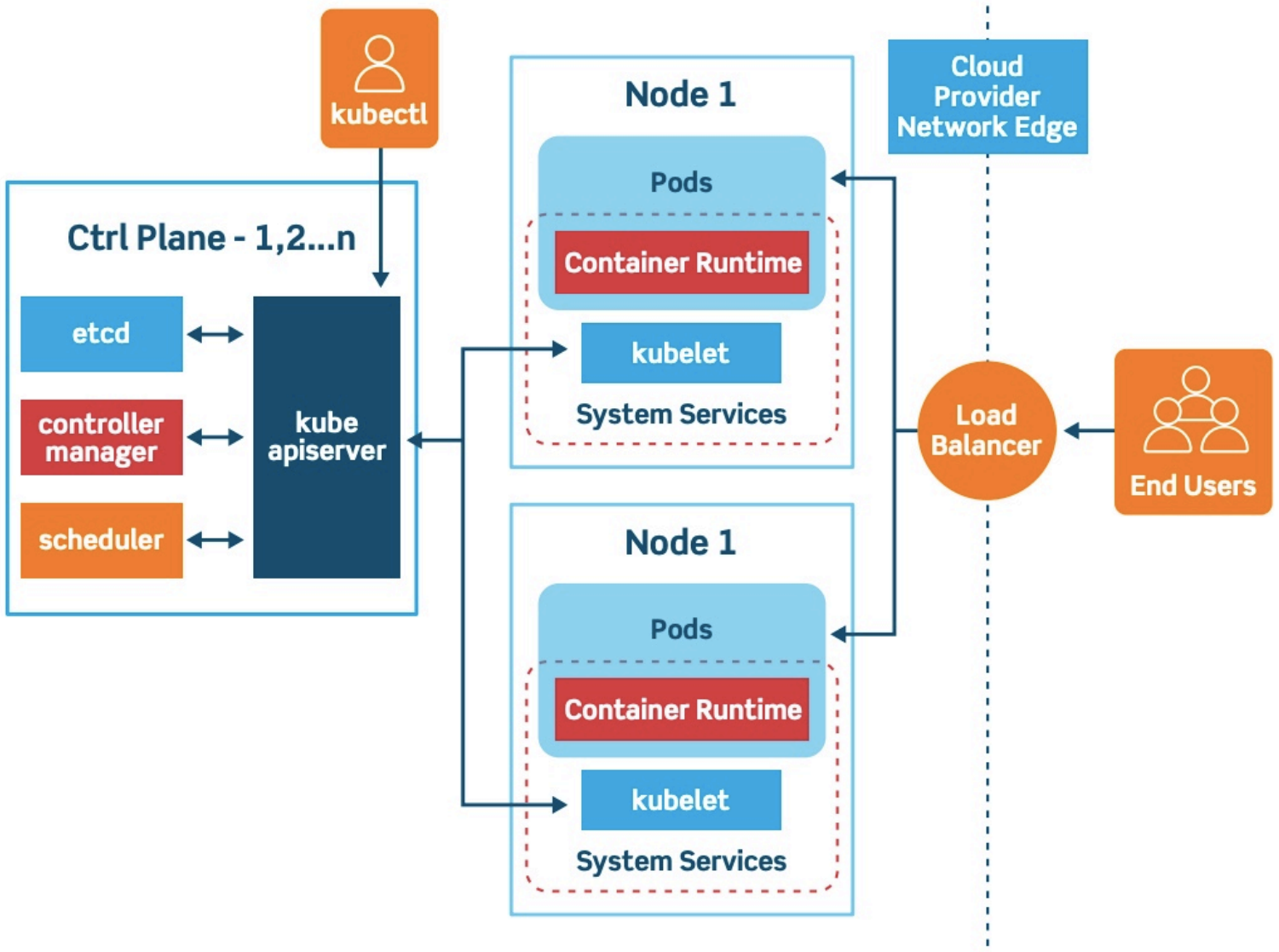
Service proxy & Service mesh

Service proxy :

- Interface entre deux environnements
- Répartition de charge (*load balancing*), metriques, etc.
- Exemples : Envoy, Traefik, etc.

Service mesh :

- Découverte, enregistrement et routage entre les conteneurs
- TLS mutuel pour les communications entre conteneurs
- Exemples : Linkerd, Istio, Consul, etc.



https://platform9.com/blog/kubernetes-enterprise-chapter-2-kubernetes-architecture-concepts/

Stockage

- Stockage éphémère par défaut pour les conteneurs
- Volume peristent pour les données persistentes
- Généralement : Réplication des données et redondance à travers le réseau
- Exemples :
  - hostPath (un seul noeud), local
  - NFS, iSCSI, fibre channel
  - Container Storage Interface (CSI) :
    - Rook (Ceph), Longhorn (Open-iSCSI/tgt), etc.

Infrastructure immuable (*Immutable infrastructure*)

> Approche de la gestion des services et logiciels où les composants sont remplacés par une nouvelle version plutôt que modifiés directement.

- Approche rendue accessible par les conteneurs
- Fonctionnement par défaut des déploiements dans Kubernetes
- Concept étendable aux systèmes d'exploitation

- What is Mutable vs. Immutable Infrastructure?
- What Is Immutable Infrastructure?

Distributions Kubernetes

- Kubernetes : Projet *upstream*
- Plus de 90 distributions certifiées par la CNCF
- Distributions « majeures » :
  - Red Hat OpenShift & OKD (Red Hat)
  - Rancher Kubernetes (SUSE/Rancher Labs)
  - k3s (SUSE/Rancher Labs)
  - Tanzu (Broadcom/VMware)
  - Typhoon (communautaire)
  - etc.

Kubernetes et Cloud providers

- Versions de Kubernetes hébergés ou gérées par une entreprise ou un cloud provider :
  - Amazon Elastic Container Service for Kubernetes (EKS)
  - Azure Kubernetes Service (AKS)
  - Google Kubernetes Engine (GKE)
  - OVH Managed Kubernetes Service
  - Scaleway Kubernetes Kapsule
  - Red Hat OpenShift clouds services (AWS, GCP, Azure, etc.)
  - etc.
- Plus ou moins intégrées dans les offres de chaque Cloud provider
- Déploiement et configuration initiale plus ou moins automatisés

Cloud Native Computing Foundation

- cncf.io
- Cloud Native Landscape

Sécurité des infrastructures de virtualisation

Sécurité des infrastructures de virtualisation

- Sécurité du fournisseur d'infrastructure Cloud :
  - Configuration et propre à chaque plateforme
- Sécurité des éléments de base de l'infrastructure :
  - Sécurité des hôtes / hyperviseurs
  - Sécurité des machines virtuelles et conteneurs
  - Services qui fournissent l'infrastructure
  - Comptes et interfaces d'administration
  - Déploiement et mises à jour
- Sécurité des éléments déployés sur l'infrastructure :
  - Services mis à disposition des utilisateurs
  - Indicateurs, visibilité et inventaire

Fournisseur de Cloud et sécurité

- Pas d'accès direct à l'infrastructure
- Responsabilité partagées :
  - Cloud provider met à disposition l'infrastruture
  - La bonne configuration est à la charge des utilisateurs
- Configuration par défaut pas nécessairement durcie :
  - Sys:All: How A Simple Loophole in Google Kubernetes Engine Puts Clusters at Risk of Compromise
  - Attacking and securing cloud identities in managed Kubernetes part 1: Amazon EKS

Authentification à plusieurs facteurs

- Priorité : protection des comptes administrateurs
- Utiliser l'authentification à deux facteurs (2FA, MFA)
- Préférer les méthodes :
  - Token hardware U2F (Standards FIDO 1 & 2 : Yubikey, NitroKey, etc.)
  - TOTP hardware ou software
- Eviter l'usage du SMS, très vulnérable face à un attaquant déterminé

- Recommandations relatives à l'authentification multifacteur et aux mots de passe
- A first glance at the U2F protocol
- Identité et méthodes d'authentification, Florian Maury

Isolation des interfaces d'administration

- Isolation réseau des interfaces d'administrations des hôtes de l'infrastructure
- Séparer les communications liées à l'administration des autres interactions
- Utiliser des protocoles éprouvés (SSH)
- Utiliser un réseau physique distinct (ou à minima un réseau logique distinct)

Isolation des services de l'infrastructure

- La compromission d'un service peut entraîner la compromission de toute l'infrastructure
- Séparer les communications entre les services des communications effectuées par les machines virtuelles, conteneurs et applications déployées par les utilisateurs sur l'infrastructure
- Utiliser des protocoles éprouvés (HTTPS avec TLS 1.2+)
- Utiliser un réseau physique distinct (ou à minima un réseau logique distinct)

Automatisation des mises à jour des hôtes

Exemples d'OS à base d'image (Image Based OS) :

- Red Hat CoreOS (OpenShift) et Fedora CoreOS (ostree & composefs)
- Flatcar Container Linux (dm-verity)
- openSUSE MicroOS (snapshots Btrfs)
- Container-Optimized OS (dm-verity, GCP uniquement)
- Bottlerocket OS (dm-verity, Amazon EKS ou ECS)
- VMware Photon OS (ostree)

Mise à jour des logiciels qui composent l'infrastructure ?

- Plein de logiciels donc quasi inévitabilité de l'existance de vulnérabilités
- Quel impact sur le fonctionnement de l'infrastructure ?

- Kubernetes :
  - kubelet et binaires associées
  - Services du cluster
- OpenStack :
  - Concepts d'UnderCloud et d'OverCloud (TripleO : OpenStack On OpenStack)
  - Red Hat OpenStack Services on OpenShift

Automatisation : CI/CD

- Gestion manuelle impossible à l'échelle $\Rightarrow$ Automatisation obligatoire
- Tests et Déploiement automatisé (Qualification et Prod)
- GitLab CI, GitHub Actions, Jenkins, Tekton, etc.
- Ne pas oublier la sécurité de la pipeline CI/CD!
- 10 real-world stories of how we've compromised CI/CD pipelines

Infrastructure as Code

> Décrire l'infrastructure sous forme de code pour pouvoir la déployer, modifier et redéployer automatiquement

- HashiCorp Terraform, OpenTofu
- Ansible :
  - Recommandations de configuration d’un système GNU/Linux - v1.2
  - Implementing ANSSI security recommendations for RHEL 7 and 8
- Chef, Puppet, SaltStack
- Limiter les accès directs en SSH aux situations critiques et au debug

GitOps

What is GitOps?

> GitOps = Infrastructures as Code + Pull Requests + CI/CD

- Gestion de l'infrastructure et des applications à l'aide de Git
- Utilisation de l'Infrastructure as Code pour décrire les déploiements
- *Pull request* pour proposer des changements
- Chaîne d'intégration continue et de déploiement continu (CI/CD) pour tester et valider les changements avant déploiement
- Exemple : Argo CD

Gestion des secrets & identités

- Gestion des secrets :
  - Kubernetes Secrets (fonctionnalités essentielles)
  - HashiCorp Vault / OpenBao : Automatisation de la création, du stockage chiffré, de l'expiration et du renouvellement

- Gestion des identités et des accès :
  - Kubernetes RBAC
  - Dex

Indicateurs, Visibilité et inventaire

- Visibilité sur l'état de l'architecture :
  - Métriques :
    - Prometheus
    - OpenTelemetry
  - Audit : osquery

Sécurité Kubernetes

Sécurité Kubernetes : Conteneurs (Pods)

Principe fondamental : Restreindre les permissions par défaut :

- Non `root`
- Filtre seccomp
- Pas de capabilities
- NoNewPrivileges
- `/` en lecture seule (Read Only)

Sécurité Kubernetes : Pod Security Policy

Pour les anciennes versions de Kubernetes (inférieur à 1.25) :

- Pod Security Policy
- Pod Security Policies Are Being Deprecated in Kubernetes
- Pod Security Policy Deprecation: Past, Present, and Future

Sécurité Kubernetes : Pod Security Standards

Pour les nouvelles versions de Kubernetes (1.25+) :

- Pod Security Standards
- Pod Security Admission
- Migrate from PodSecurityPolicy to the Built-In PodSecurity Admission Controller
- Enforcing Pod Security Standards
- Alternatives :
  - Kubewarden, Kyverno, OPA Gatekeeper

Sécurité Kubernetes

Ressources et guides :

- Kubernetes - Securing a Cluster
- Official CVE Feed
- Kubernetes security fundamentals: Introduction
- Kubernetes security fundamentals: API Security
- Kubernetes security fundamentals: Authentication
- OpenShift Container Security (Red Hat)
- OpenShift Security Guide Book (Red Hat)

Sécurité Kubernetes

Mini CTF :

- A beginner-friendly CTF about Kubernetes security
- kdigger: a Context Discovery Tool for Kubernetes
- quarkslab/kdigger (GitHub)

Red Hat OpenShift

- Sécurité par défaut (`!root` , Read Only, etc.)
  - Security Context Constraints : équivalent des Pod Security Standards
  - Seccomp defaults in Red Hat OpenShift
- Mises à jour coordonnée de l'intégralité de la plateforme
  - Red Hat Enterprise Linux CoreOS (basé sur Red Hat Enterprise Linux)
- Gestion automatisée à l'aide d'*operators*
- Support des conteneurs, machines virtuelles, fonctions, CI/CD, GitOps, etc.
- Intégration de Red Hat Quay, Data Foundation (Ceph Storage), Advanced Cluster Management, Advanced Cluster Security
- Red Hat deuxième contributeur à Kubernetes (derrière Google)

Red Hat
OpenShift
Kubernetes Engine

Includes:

- Enterprise Kubernetes runtime
- Red Hat Enterprise Linux CoreOS immutable container operating system
- Administrator console
- Red Hat OpenShift Virtualization

Red Hat
OpenShift
Container Platform

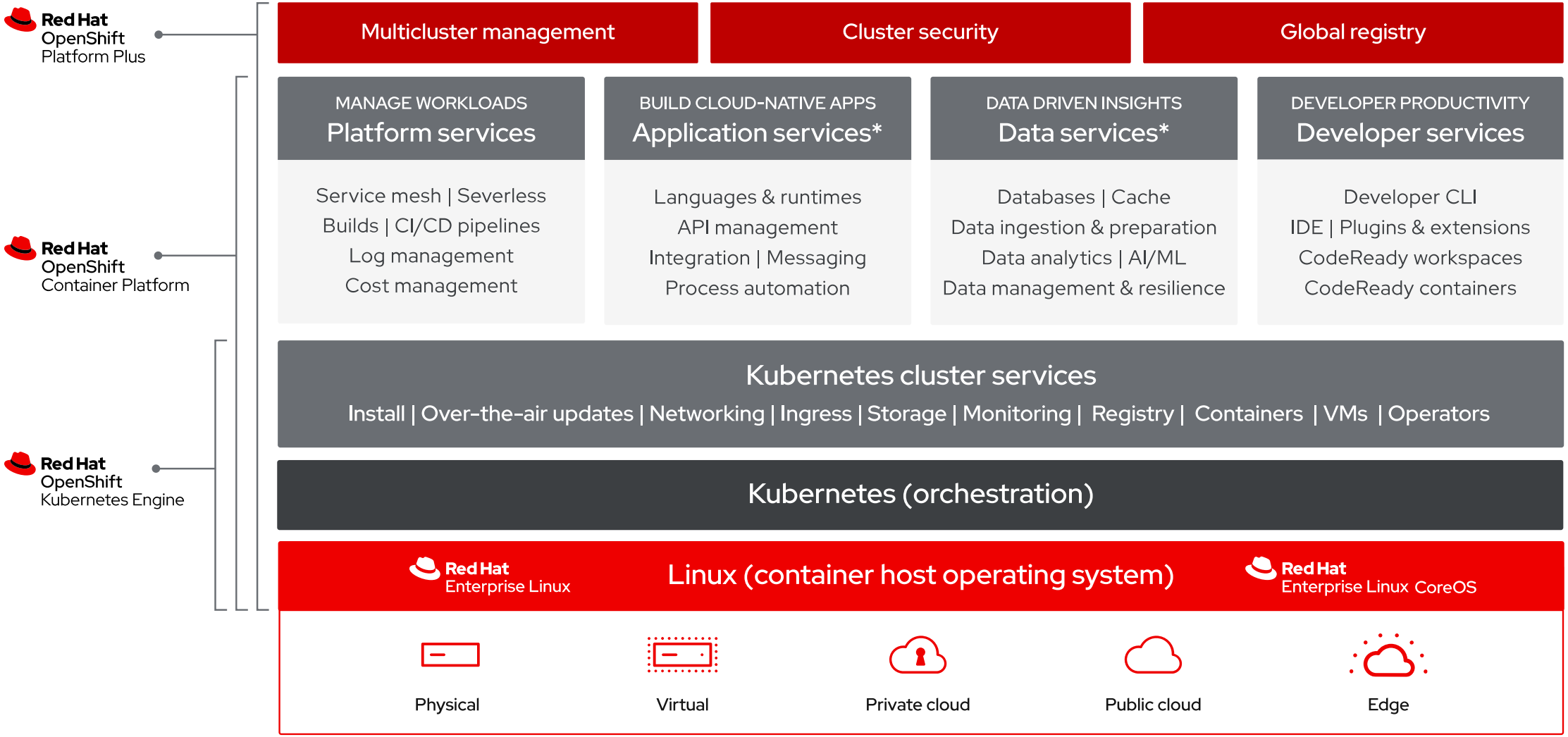
Adds:

- Developer console
- Log management andmetering/cost management
- Red Hat OpenShift Serverless (Knative)
- Red Hat OpenShift Service Mesh (Istio)
- Red Hat OpenShift Pipelinesand Red Hat OpenShift GitOps (Tekton, ArgoCD)

Red Hat
OpenShift
Platform Plus

Adds:

- Red Hat Advanced Cluster Management for Kubernetes
- Red Hat Advanced Cluster Security for Kubernetes
- Red Hat Quay



*Red Hat OpenShift® includes supported runtimes for popular languages/frameworks/databases.
Additional capabilities listed are from the Red Hat Application and Data Services portfolio.