



# KUBERNETES

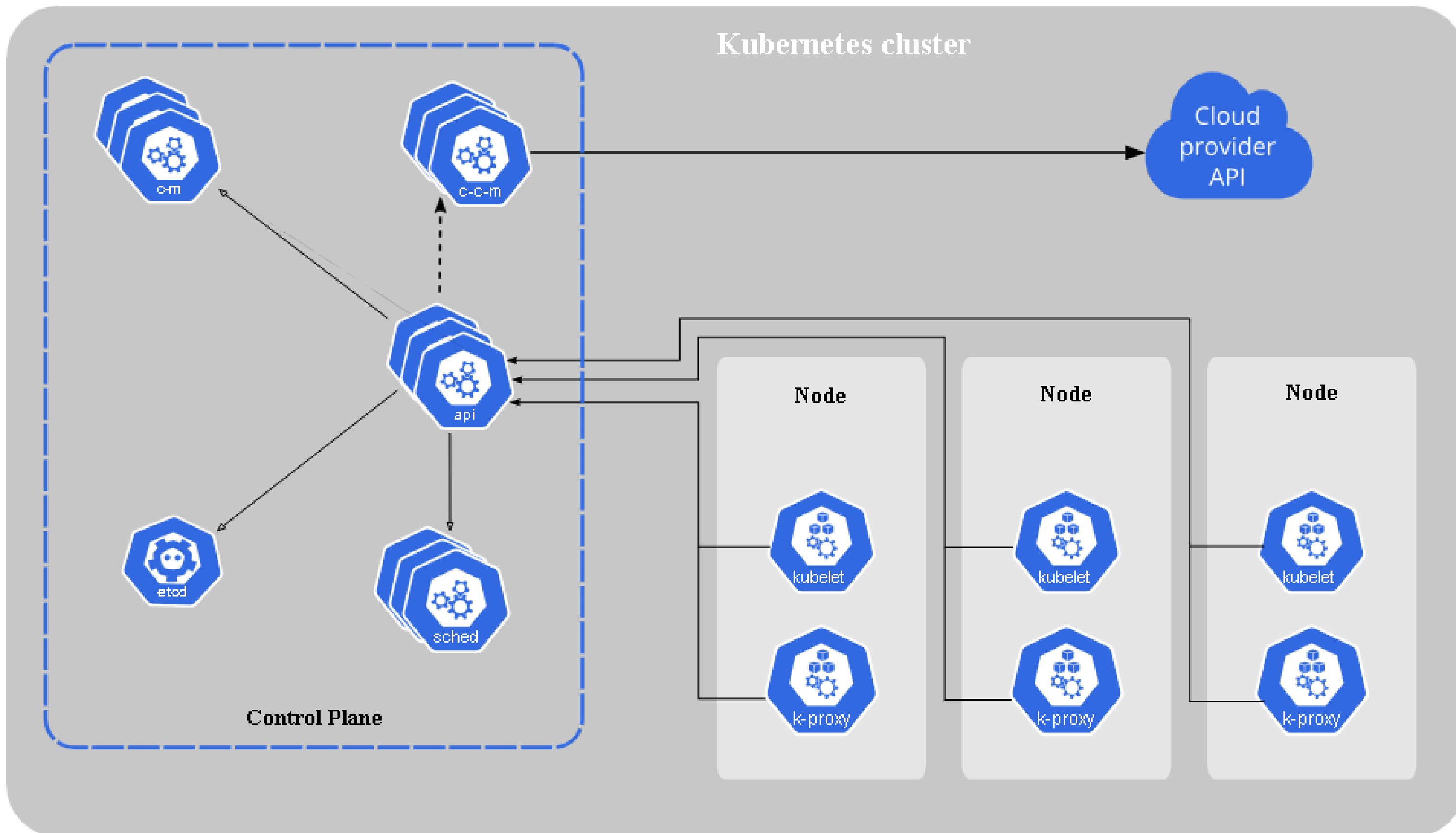
PRÉSENTÉ PAR:  
ROMDHANE MED RAYEN

# INTRODUCTION

- KUBERNETES EST UNE PLATEFORME OPEN-SOURCE DE GESTION D'ORCHESTRATION DE CONTENEURS.
- ELLE PERMET AUX DÉVELOPPEURS DE DÉPLOYER, SCALER ET GÉRER EFFICACEMENT DES APPLICATIONS CONTENEURISÉES.
- KUBERNETES RÉSOUT LES DÉFIS DE DÉPLOIEMENT D'APPLICATIONS DANS DES ENVIRONNEMENTS CLOUD OU SUR SITE.

# ARCHITECTURE DE BASE

- **MAÎTRE (MASTER)** : CENTRALISE LA GESTION DE L'ENSEMBLE DU CLUSTER KUBERNETES, COMPOSÉ DE L'API SERVER, DU SCHEDULER, DU CONTROLLER MANAGER ET DE ETCD, QUI STOCKE L'ÉTAT DU CLUSTER.
- **NŒUDS (NODES)** : LES MACHINES PHYSIQUES OU VIRTUELLES SUR LESQUELLES LES APPLICATIONS S'EXÉCUTENT, COMPRENANT LE KUBELET, LE KUBE-PROXY ET UN RUNTIME DE CONTENEUR COMME DOCKER.



# COMPOSANTS PRINCIPAUX

- **API SERVER** : POINT CENTRAL D'ACCÈS ET DE GESTION POUR L'ENSEMBLE DU CLUSTER KUBERNETES, EXPOSANT L'API REST.
- **SCHEDULER** : SERVICE RESPONSABLE DE LA PLANIFICATION DES PODS SUR LES NOEUDS DISPONIBLES, EN FONCTION DES EXIGENCES DE RESSOURCES ET DES POLITIQUES DÉFINIES.
- **CONTROLLER MANAGER** : SURVEILLE EN PERMANENCE L'ÉTAT DU CLUSTER VIA L'API SERVER ET S'ASSURE QUE LE CLUSTER RESPECTE LES CONFIGURATIONS DÉFINIES.
- **ETCD** : UN MAGASIN DE CLÉS-VALEURS DISTRIBUÉ, UTILISÉ POUR STOCKER LES DONNÉES DE CONFIGURATION ET L'ÉTAT DU CLUSTER.

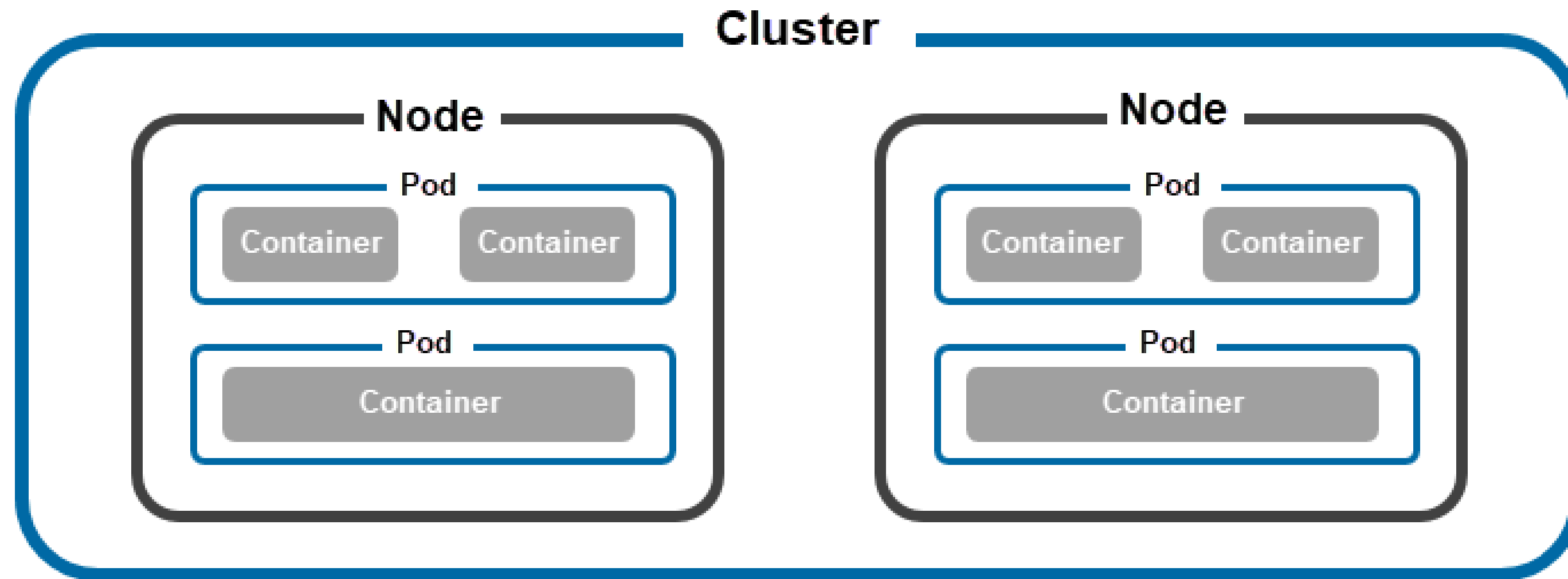
# CONCEPTS FONDAMENTAUX

- **CONTENEURS** : DES UNITÉS LÉGÈRES ET PORTABLES QUI ENCAPSULENT LES ÉLÉMENTS NÉCESSAIRES À L'EXÉCUTION D'UNE APPLICATION.
- **PODS** : LES PLUS PETITES UNITÉS DÉPLOYABLES DANS KUBERNETES, REGROUPANT UN OU PLUSIEURS CONTENEURS ET PARTAGEANT UN ESPACE DE STOCKAGE RÉSEAU ET DES SPÉCIFICATIONS DE POLITIQUE.
- **SERVICES** : FACILITENT LA COMMUNICATION ENTRE DIFFÉRENTS PODS, PERMETTANT D'ACCÉDER À UNE APPLICATION DE MANIÈRE STABLE VIA UNE ADRESSE IP ET UN PORT.

# LES PODS

- LES PODS SONT COMME UNE BOÎTE QUI ENCAPSULE UN OU PLUSIEURS CONTENEURS PARTAGEANT LES MÊMES RESSOURCES RÉSEAU ET STOCKAGE.
- LES PODS SONT LES PLUS PETITES UNITÉS DÉPLOYABLES GÉRÉES PAR KUBERNETES.
- LES PODS SONT ÉPHÉMÈRES : LORS DU SCALING UP/DOWN, LE NOMBRE DE PODS DIMINUE OU AUGMENTE.
- PAR DÉFAUT, LES PODS SONT INACCESSIBLES DEPUIS L'EXTÉRIEUR DU CLUSTER.

# LES PODS





# LES OBJETS K8S

- KUBERNETES UTILISE DES OBJETS POUR REPRÉSENTER L'ÉTAT DE L'APPLICATION ET DU CLUSTER, DÉFINISSANT CE QUI EST EXÉCUTÉ ET COMMENT L'APPLICATION EST GÉRÉE.
- LES PRINCIPAUX OBJETS KUBERNETES INCLUENT LES DÉPLOIEMENTS, LES SERVICES, LES PODS, LES CONFIGMAPS, LES SECRETS, ETC.
- CHAQUE OBJET A UN ENSEMBLE DE CHAMPS SPÉCIFIÉS POUR DÉFINIR SON COMPORTEMENT ET SON ÉTAT DÉSIRÉ.
- LES SPÉCIFICATIONS DES OBJETS SONT SOUVENT DÉFINIES DANS DES FICHIERS YAML, OFFRANT UNE SYNTAXE CLAIRE ET LISIBLE POUR LA GESTION DÉCLARATIVE DES RESSOURCES KUBERNETES.

Descriptor conforms  
to version v1 of  
Kubernetes API

You're  
describing a pod.

```
apiVersion: v1
kind: Pod
metadata:
  name: kubia-manual
spec:
  containers:
  - image: luksa/kubia
    name: kubia
    ports:
    - containerPort: 8080
      protocol: TCP
```

The name  
of the pod

Container image to create  
the container from

Name of the container

The port the app  
is listening on

# SERVICES

- UN SERVICE REGROUPE UN ENSEMBLE DE PODS AYANT LA MÊME FONCTION, LES PRÉSENTANT COMME UNE SEULE ENTITÉ DANS KUBERNETES.
- IL REPRÉSENTE LE POINT D'ENTRÉE UNIQUE VERS CE GROUPE DE PODS, FOURNISSANT UNE ADRESSE CONSTANTE ET STABLE.
- LES SERVICES ASSURENT LE LOAD BALANCING ENTRE LES PODS, RÉPARTISSANT LE TRAFIC DE MANIÈRE ÉQUILIBRÉE.
- LES SERVICES PEUVENT ÊTRE DE DIFFÉRENTS TYPES : CLUSTERIP, NODEPORT, LOADBALANCER, ET EXTERNALNAME, OFFRANT DIFFÉRENTES OPTIONS D'EXPOSITION ET DE ROUTAGE DU TRAFIC.

# INGRESS

- GÈRE L'ACCÈS HTTP ET HTTPS AUX SERVICES DANS LE CLUSTER KUBERNETES.
- PERMET DE ROUTER LE TRAFIC VERS DIFFÉRENTS SERVICES EN FONCTION DES RÈGLES DE ROUTAGE.
- SIMPLIFIE LA CONFIGURATION DES ROUTES ET DES RÈGLES DE TRAFIC POUR PLUSIEURS SERVICES.
- NÉCESSITE UN CONTRÔLEUR INGRESS POUR FONCTIONNER, QUI PEUT ÊTRE FOURNI PAR LE CLUSTER OU PAR UN FOURNISSEUR DE CLOUD.

# CONFIG MAP

- STOCKE DES CONFIGURATIONS SOUS FORME DE PAIRES CLÉ-VALEUR.
- PERMET DE SÉPARER LA CONFIGURATION DES APPLICATIONS DE LEUR CODE.
- LES VALEURS PEUVENT ÊTRE INJECTÉES DANS LES PODS EN TANT QUE VARIABLES D'ENVIRONNEMENT OU MONTÉES EN TANT QUE FICHIERS.
- FACILITE LA GESTION ET LA MISE À JOUR DES CONFIGURATIONS SANS MODIFIER LE CODE DES APPLICATIONS.

# SECRETS

- LES SECRETS KUBERNETES SONT UTILISÉS POUR STOCKER DES INFORMATIONS CONFIDENTIELLES TELLES QUE DES MOTS DE PASSE, DES JETONS D'AUTHENTIFICATION, OU DES CLÉS API.
- ILS PERMETTENT DE SÉCURISER L'ACCÈS À CES INFORMATIONS AU SEIN DES APPLICATIONS KUBERNETES, EN ÉVITANT LEUR EXPOSITION DIRECTE DANS LES CONFIGURATIONS DES PODS.
- KUBERNETES CHIFFRE LES DONNÉES DES SECRETS AU REPOS, LES STOCKANT DANS ETCD SOUS FORME CODÉE EN BASE64.

# VOLUMES

- LES VOLUMES SONT CRÉÉS AU DÉMARRAGE DES PODS ET DÉTRUITS LORSQUE LES PODS SONT SUPPRIMÉS.
- ILS FONT PARTIE INTÉGRANTE D'UN POD ET PARTAGENT SON CYCLE DE VIE.
- LES VOLUMES OFFRENT UN STOCKAGE PARTAGÉ ENTRE LES DIFFÉRENTS CONTENEURS D'UN MÊME POD.
- EN CAS DE DÉFAILLANCE D'UN CONTENEUR, KUBERNETES LE REDÉMARRE, MAIS LES FICHIERS ASSOCIÉS SERONT PRÉSERVÉS DANS LE VOLUME PARTAGÉ.

# VOLUMES PERSISTANTS

- LES VOLUMES PERSISTANTS PRÉSERVENT LES DONNÉES AU-DELÀ DU CYCLE DE VIE DES PODS.
- ILS PERMETTENT LE PARTAGE DE DONNÉES ENTRE PLUSIEURS PODS.
- ESSENTIELS POUR STOCKER DES DONNÉES CRITIQUES TELLES QUE LES BASES DE DONNÉES OU LES FICHIERS DE CONFIGURATION.
- GARANTISSENT LA PERSISTANCE DES DONNÉES, ASSURANT AINSI LA CONTINUITÉ DE L'APPLICATION.



# PVC

- LES PERSISTENTVOLUMECLAIMS (PVCS) SONT DES RESSOURCES KUBERNETES UTILISÉES PAR LES PODS POUR RÉCLAMER DE L'ESPACE DE STOCKAGE PERSISTANT.
- ILS PERMETTENT AUX APPLICATIONS DE DEMANDER UN ESPACE DE STOCKAGE SPÉCIFIQUE, SANS SE SOUCIER DE SA LOCALISATION.
- LES PVCS SONT ASSOCIÉS À DES PERSISTENTVOLUMES (PVS) QUI FOURNISSENT RÉELLEMENT L'ESPACE DE STOCKAGE, CRÉANT AINSI UN LIEN DYNAMIQUE ENTRE LES BESOINS DE STOCKAGE DES APPLICATIONS ET LES RESSOURCES.

# DEPLOIEMENT

- OBJET DE RESSOURCE DANS KUBERNETES PERMETTANT DES MISES À JOUR DÉCLARATIVES POUR LES APPLICATIONS.
- DESCRIPTION DU CYCLE DE VIE DE L'APPLICATION, Y COMPRIS LES IMAGES À UTILISER ET LE NOMBRE DE PODS.
- FOURNIT UNE GESTION CENTRALISÉE ET REPRODUCTIBLE DES DÉPLOIEMENTS, AMÉLIORANT LA COHÉRENCE ET LA FIABILITÉ DES APPLICATIONS DÉPLOYÉES.

# REPLICA SET

- MAINTIENT UN NOMBRE SPÉCIFIÉ DE RÉPLIQUES DE PODS.
- GARANTIT QUE LE NOMBRE ACTUEL DE PODS CORRESPOND AU NOMBRE DÉSIRÉ.
- COMPOSÉ DE:
  - LABEL SELECTOR: POUR IDENTIFIER LES PODS ASSOCIÉS
  - REPLICA COUNT: POUR DÉFINIR LE NOMBRE DÉSIRÉ DE RÉPLIQUES
  - UN POD TEMPLATE: POUR DÉFINIR LES CARACTÉRISTIQUES DES NOUVEAUX PODS.
- NOUVELLE GÉNÉRATION DES REPLICACONTROLLERS, CAPABLE DE TRAITER UN GROUPE DE PODS COMME UNE SEULE ENTITÉ.

# LES COMMANDES DE BASE

- **KUBECTL GET PODS:** CETTE COMMANDE PERMET DE LISTER LES PODS PRÉSENTS SUR LE CLUSTER KUBERNETES. CETTE COMMANDE FONCTIONNE POUR TOUS LES TYPES DE RESSOURCES KUBERNETES: PODS, SERVICES, DEPLOYMENTS, CRONJOBS, EVENTS, INGRESSES, ETC
- **KUBECTL APPLY -F FICHER:** APPLIQUE UN CHANGEMENT DE CONFIGURATION À UNE RESSOURCE DEPUIS VOTRE FICHER.
- **KUBECTL DESCRIBE PODS MY-POD:** LA COMMANDE DESCRIBE DONNE UN AFFICHAGE VERBEUX DU POD CONTRAIREMENT AU GET ET AFFICHAGE BASIQUE. CELA PERMET D'AVOIR LES EVENTS, UTILE QUAND UN POD NE DÉMARRE PAS.
- **KUBECTL LOGS -F MY\_POD -C MY\_APP:** CETTE COMMANDE AFFICHE LES LOGS DE VOTRE POD.

# HELM

- **HELM** EST UN GESTIONNAIRE DE PAQUETS POUR KUBERNETES, SOUVENT APPELÉ LE "GESTIONNAIRE DE PAQUETS DE KUBERNETES".
- IL SIMPLIFIE LE PROCESSUS DE DÉPLOIEMENT ET DE GESTION DES APPLICATIONS SUR DES CLUSTERS KUBERNETES.
- HELM UTILISE DES **CHARTS**, QUI SONT DES COLLECTIONS DE FICHIERS DÉCRIVANT UN ENSEMBLE DE RESSOURCES KUBERNETES LIÉES.

# CARACTÉRISTIQUES DE HELM

- **GESTION DES PAQUETS** : LES PAQUETS HELM SONT APPELÉS CHARTS, QUI PEUVENT DÉFINIR, INSTALLER ET METTRE À JOUR DES APPLICATIONS KUBERNETES COMPLEXES.
- **GESTION DES VERSIONS** : HELM PREND EN CHARGE LES CHARTS VERSIONNÉS, PERMETTANT DE MAINTENIR DIFFÉRENTES VERSIONS DES APPLICATIONS.
- **RÉUTILISABILITÉ** : LES CHARTS PEUVENT ÊTRE PARTAGÉS ET RÉUTILISÉS DANS DIFFÉRENTS PROJETS ET ÉQUIPES.
- **GESTION DES DÉPENDANCES** : HELM GÈRE LES DÉPENDANCES ENTRE DIFFÉRENTS CHARTS ET VEILLE À CE QU'ELLES SOIENT CORRECTEMENT GÉRÉES ET INSTALLÉES.
- **RETOUR ARRIÈRE** : HELM PERMET DE REVENIR FACILEMENT À DES VERSIONS PRÉCÉDENTES EN CAS DE PROBLÈMES OU DE DÉFAILLANCES.

# FONCTIONNEMENT DE HELM

- **CHART** : UN PAQUET HELM CONTENANT TOUTES LES DÉFINITIONS DE RESSOURCES NÉCESSAIRES POUR EXÉCUTER UNE APPLICATION.
  - **CHART.YAML** : MÉTADONNÉES SUR LE CHART (NOM, VERSION, DESCRIPTION).
  - **VALUES.YAML** : VALEURS DE CONFIGURATION PAR DÉFAUT POUR LE CHART.
  - **TEMPLATES** : FICHIERS MANIFESTES KUBERNETES POUVANT ÊTRE PARAMÉTRÉS À L'AIDE DE VALEURS DE VALUES.YAML.
- **RELEASE** : UNE INSTANCE D'UN CHART EXÉCUTÉ DANS UN CLUSTER KUBERNETES. UN MÊME CHART PEUT ÊTRE INSTALLÉ PLUSIEURS FOIS POUR CRÉER PLUSIEURS RELEASES.
- **REPOSITORY** : UNE COLLECTION DE CHARTS HELM POUVANT ÊTRE STOCKÉS ET PARTAGÉS. LES DÉPÔTS HELM PERMETTENT AUX UTILISATEURS DE PUBLIER ET DE PARTAGER FACILEMENT DES CHARTS.

**MERCI  
DE VOTRE  
ATTENTION**

