



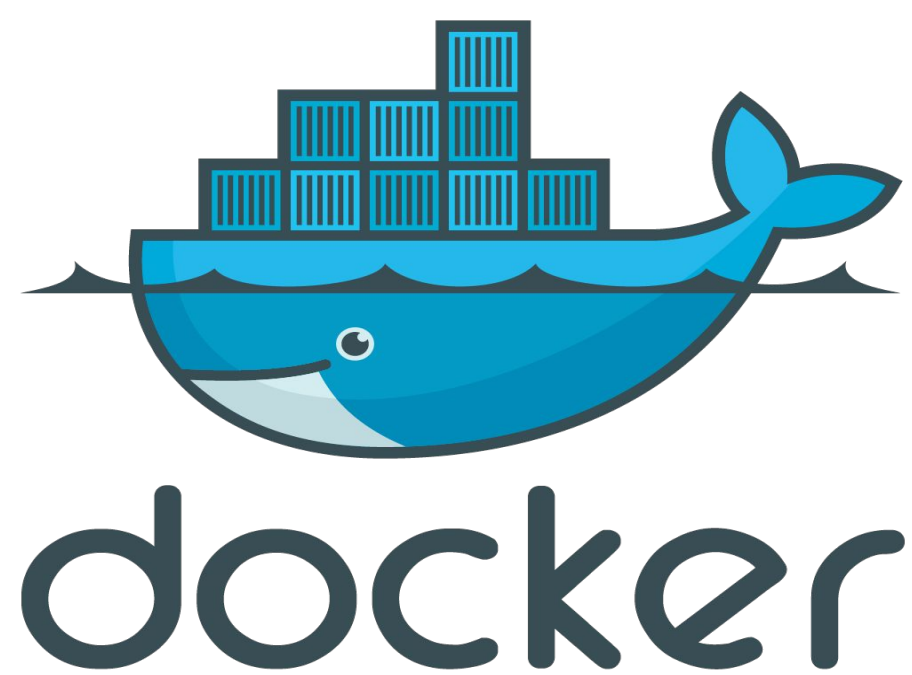
Samuel ANTUNES
Consultant Ingénieur DevSecOps
OCTO Technology
Email : contact@samuelantunes.fr

ICEBREAKER

shorturl.at/HMOP2

1. Les bases de Kubernetes
2. Manipulation simple de Kubernetes
3. Mettre son application dans K8s
4. Le Continuous Delivery avec K8s
5. Conclusion & Take Away

“ Les bases de Kubernetes ”



- D'anciens développeurs de Borg écrivent K8s en Go
- Directement pensé pour utiliser Docker (engine)
- Directement dans l'optique d'en faire un projet OpenSource
- Version 1.0 en Juin 2015

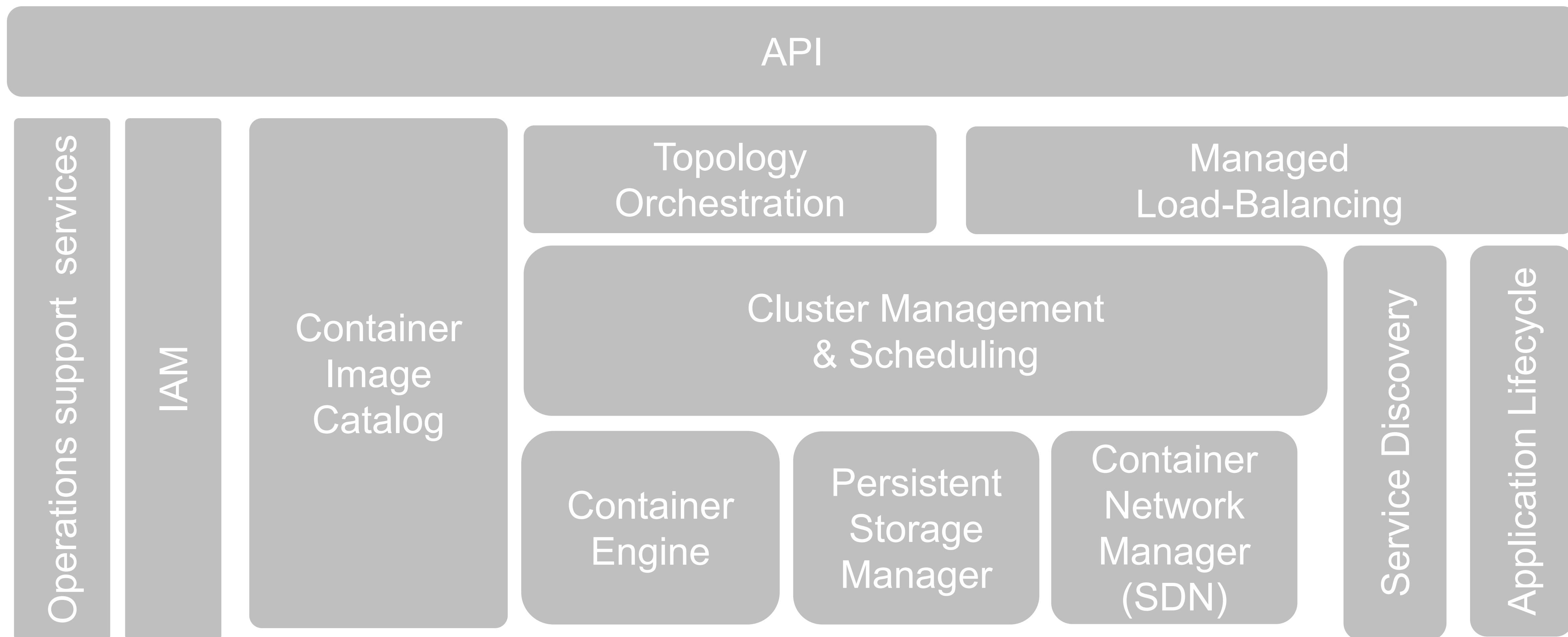


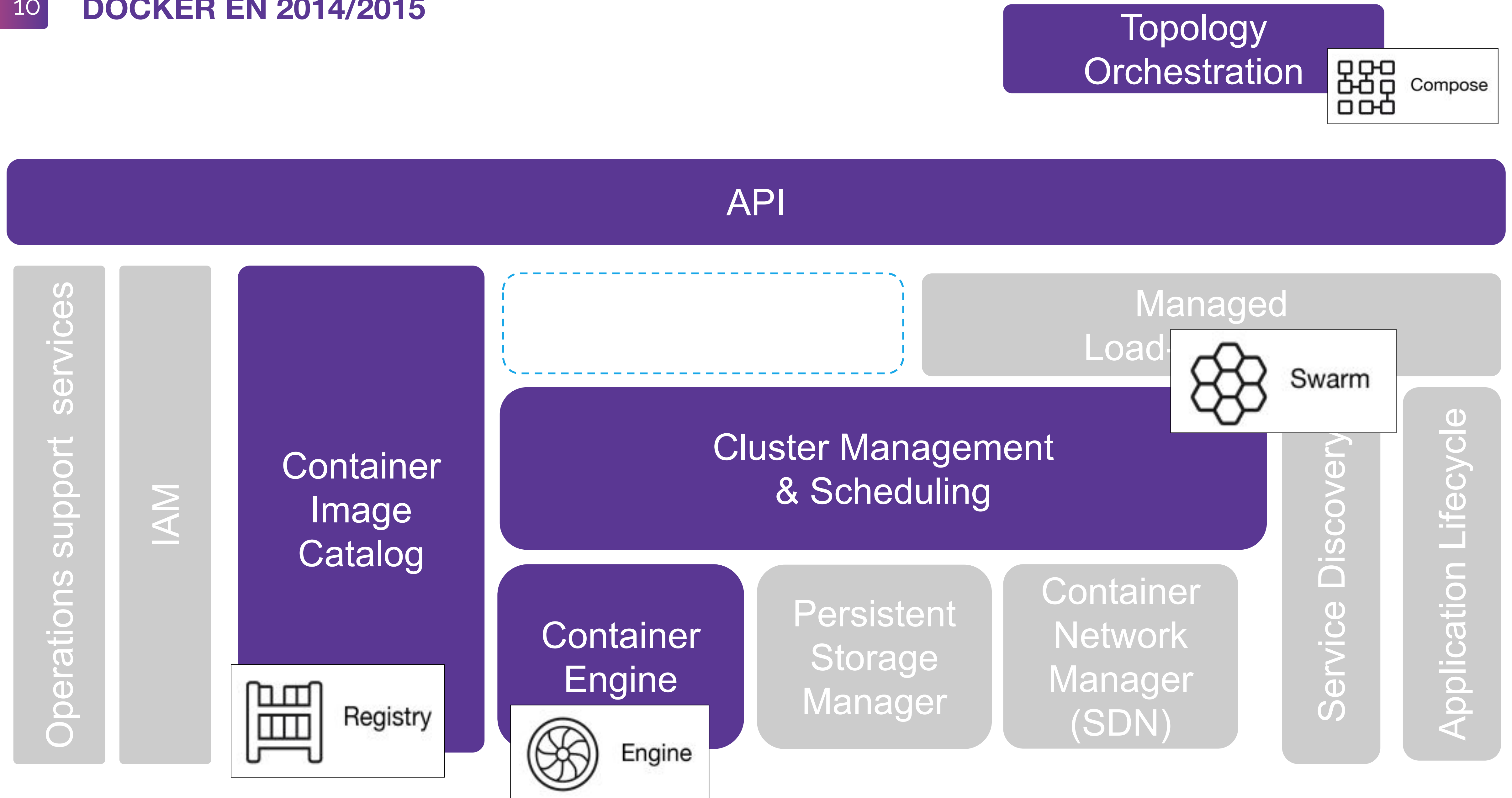
- Définir et déployer des applications multi-conteneurs
- Répartir les conteneurs sur une flotte d'hôtes (nœuds)
- Optimiser et adapter le placement des conteneurs
- Surveiller la santé des conteneurs
- Définir et appliquer des contraintes de niveaux de services
- Gérer la disponibilité et la scalabilité des conteneurs
- Gérer le provisionnement et l'accès au stockage
- Isoler les conteneurs
 - Limitation de ressources
 - Sécurité (vision multi-tenant)

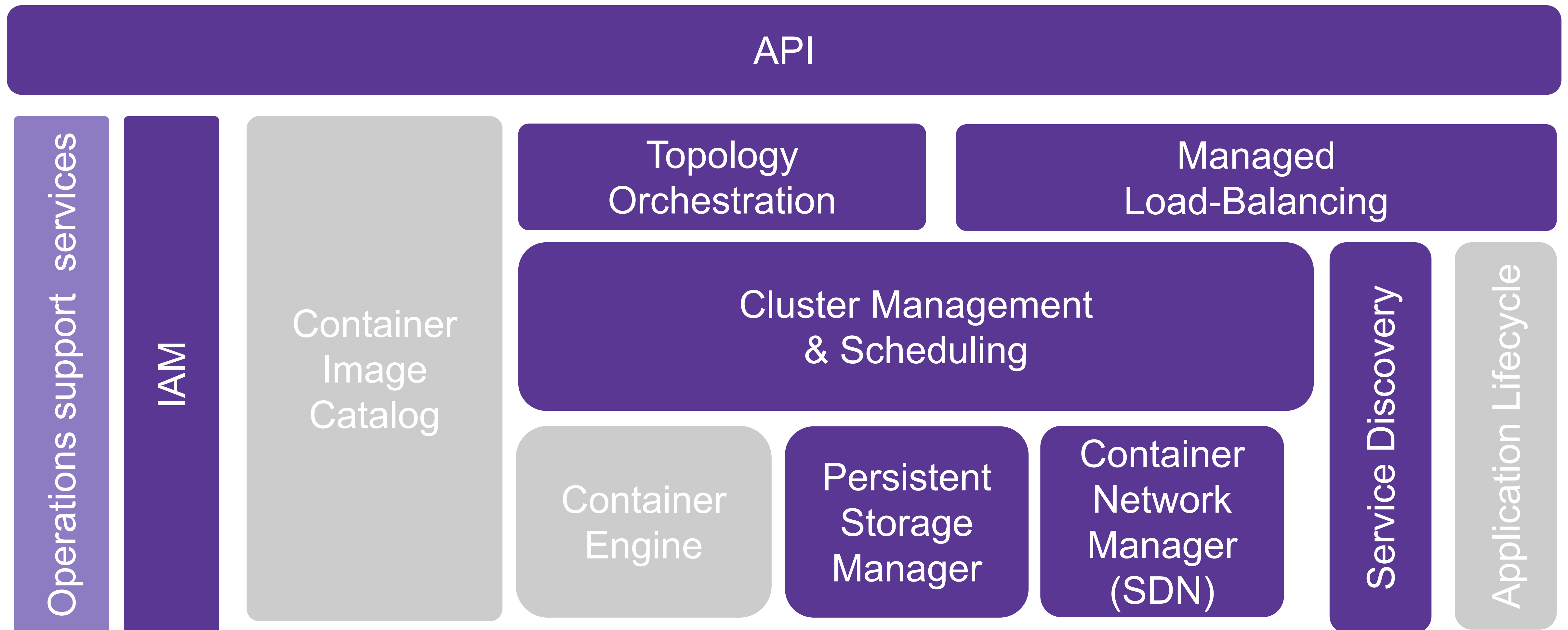


Tout ça de manière dynamique et pour des milliers de conteneurs !

- **Abstraction** des concepts
 - *Apparition de différents types de ressources de haut niveau*
 - *On ne manipule que très rarement la notion de conteneur directement*
- Approche **déclarative** plutôt que procédurale
 - *On décrit ce que l'on souhaite, pas comment l'obtenir*
 - *Notion de Desired State Configuration*







Comme Docker, K8s respecte ces différents principes d'architecture:

- Scalables horizontalement
- Immuables
- Sans état
- Share nothing
- Création et destruction facile, rapide et à faible coût

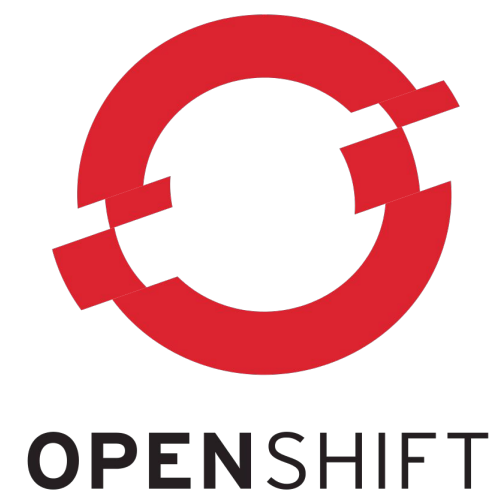


- On leur donne des noms (ex: monchaton)
- Ils sont uniques, on en prend soin
- Quand ils vont mal, on les chouchoute pour qu'ils aillent mieux

- On leur donne des numéros (ex: 3402)
- Elles se ressemblent toutes
- Quand elles sont malades, on les remplace



Amazon EC2
Container Service



Chez soi

- Directement sur des serveurs physiques (bare metal)
- Sur des VMs traditionnelles (VMWare)
- Sur un cloud privé (OpenStack)
- Sur son **laptop** (minikube / Docker Desktop)

Sur le cloud (avec des capacités d'intégration avancées : LB, Volumes...)

- Amazon Web Services (VM ou EKS)
- Google Cloud Platform (VM ou GKE)
- Azure (VM ou AKS)
- Chez beaucoup d'autres fournisseurs de clusters Kubernetes managés

Des outils connexes au projet Kubernetes sont là pour aider les déploiements (kops, kubeadm...)

<https://kubernetes.io/fr/docs/tasks/tools/install-minikube/#installer-minikube>

Pour vérifier la bonne installation des différents outils, il faut que vous lanciez les commandes suivantes et ayez un résultat similaire à l'image ci-dessous :

- `kubectl version --client`
- `minikube version`

```
samuel.antunes@AMAC02ZV0ZQMD6V ➤ ~/perso/formations/Kubernetes ➤ kubectl version --client
Client Version: version.Info{Major:"1", Minor:"16+", GitVersion:"v1.16.6-beta.0", GitCommit:"e7f962ba86f4ce7033828210ca3556393c377bcc", GitTreeState:"clean", BuildDate:"2020-01-15T08:26:26Z", GoVersion:"go1.13.5", Compiler:"gc", Platform:"darwin/amd64"}
samuel.antunes@AMAC02ZV0ZQMD6V ➤ ~/perso/formations/Kubernetes ➤ minikube version
minikube version: v1.13.0
commit: eeb05350f8ba6ff3a12791fcce350c131cb2ff44
samuel.antunes@AMAC02ZV0ZQMD6V ➤ ~/perso/formations/Kubernetes ➤
```