

B

Gate

27-6-2019

**CONTAINERS
TODAY**

The Hague

Powered by

Amazic
Market Development

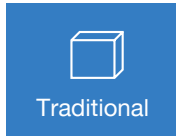
Kubernetes made easy

Patrick van der Bleek
Sr. Solutions Engineer @Docker Inc

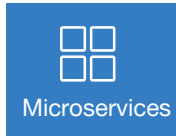


Any App, Any OS, Any Infrastructure

DEVELOPERS



Traditional



Microservices



Edge & IoT



ISV



Big Data
ML & AI

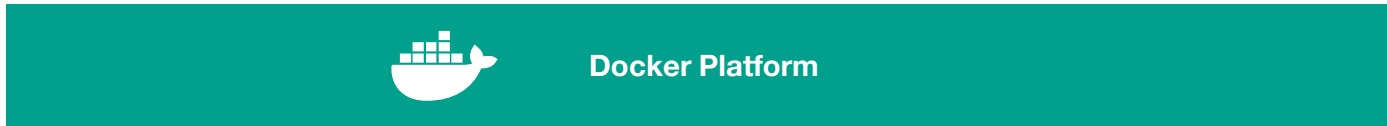


Blockchain



Serverless

OPERATORS



Cloud



VM



Bare Metal

Any App, Any OS, Any Infrastructure

DEVELOPERS



Traditional



Microservices



Edge & IoT



ISV



Big Data
ML & AI



Blockchain



Serverless

OPERATORS



CHOICE

AGILITY

SECURITY



Cloud

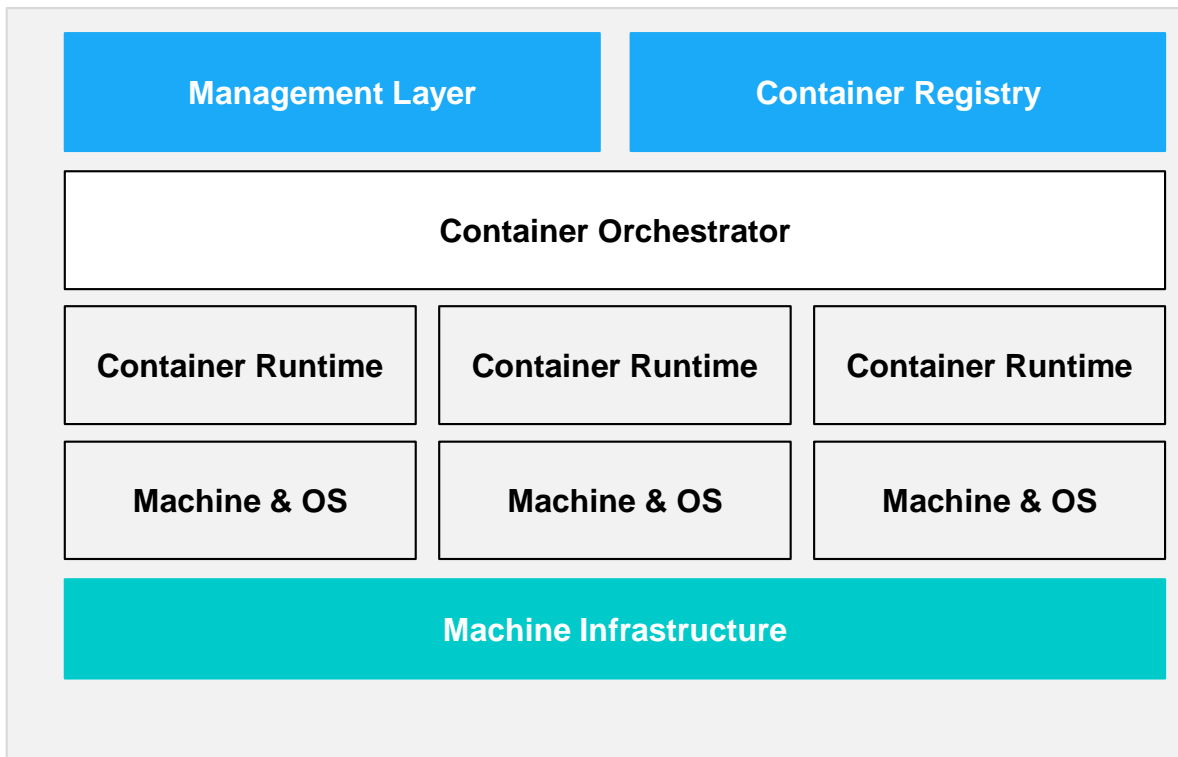


VM

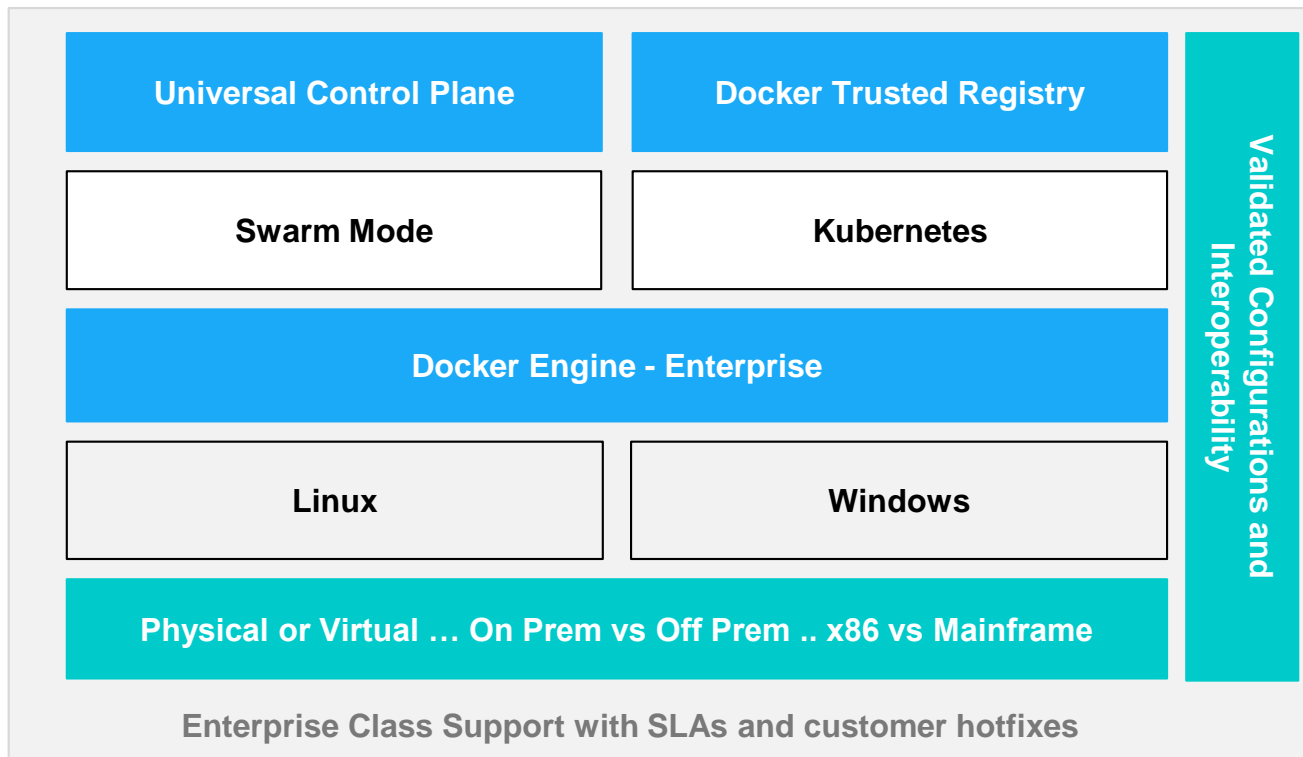


Bare Metal

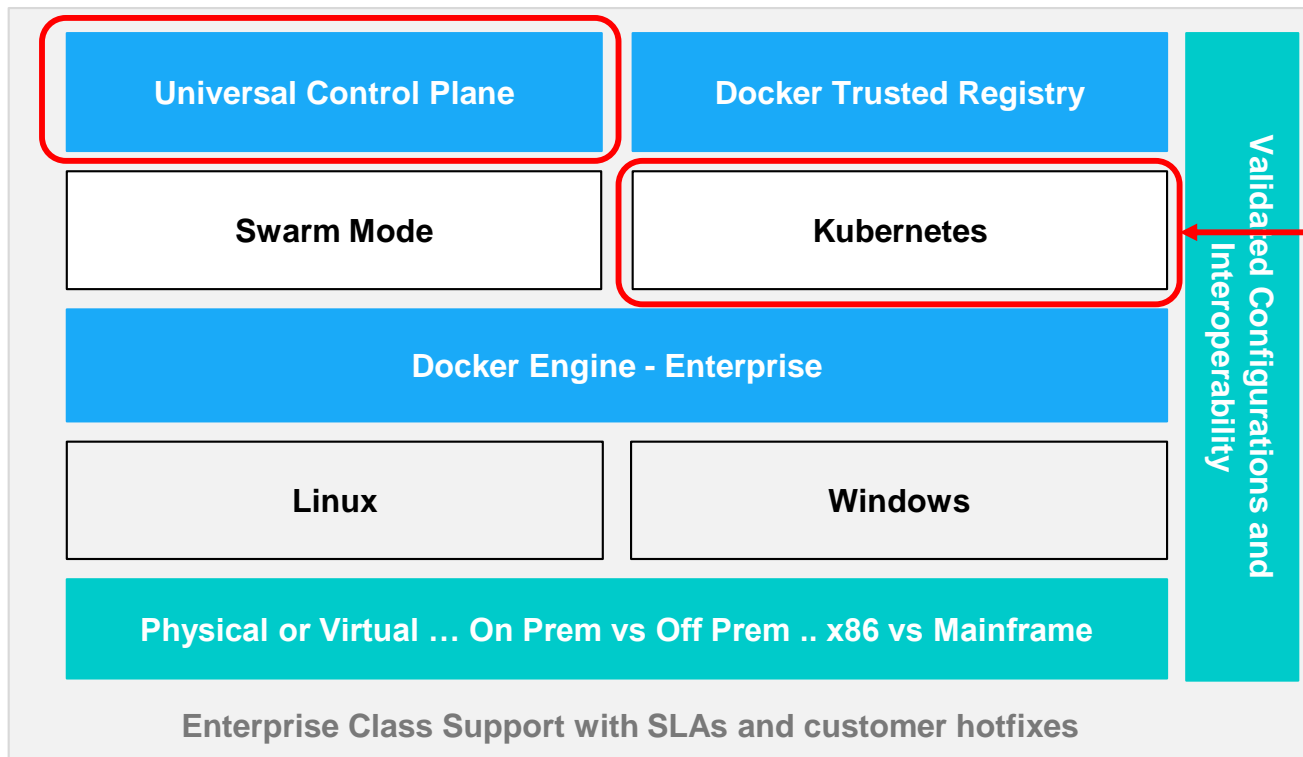
Whats is a CaaS platform?



Docker Enterprise Container Platform



Docker Enterprise Container Platform



Docker with Swarm and Kubernetes

1

The best enterprise
container security and
management

Docker Enterprise Edition

Docker Community Edition



kubernetes



containerd

2

The best container
development workflow

3

Native Kubernetes
integration provides full
ecosystem compatibility

4

Industry-standard
container runtime

Kubernetes lifecycle

A Container Platform Lifecycle

Install

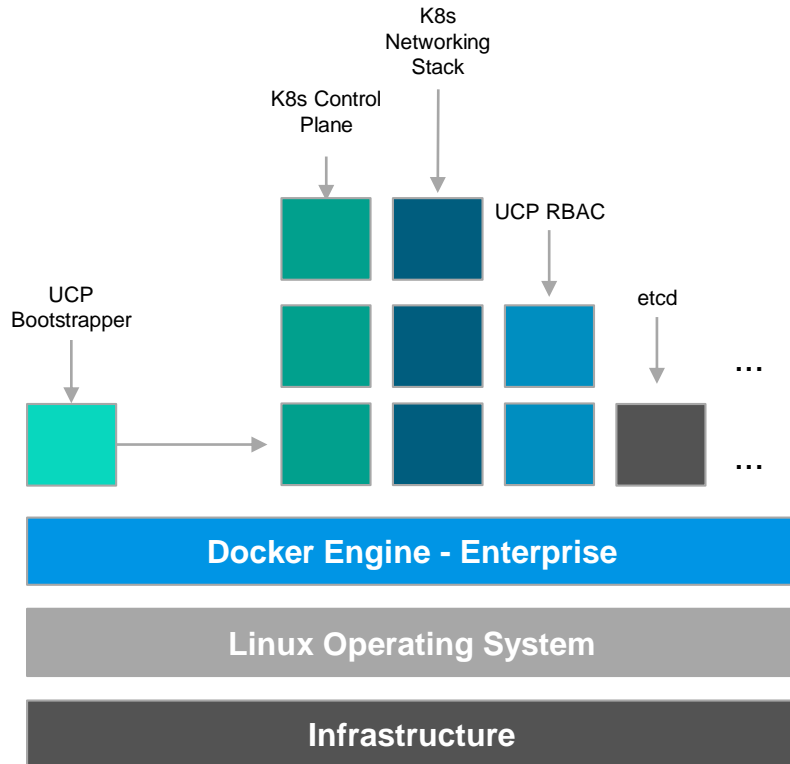
Expand

Backup

Upgrade

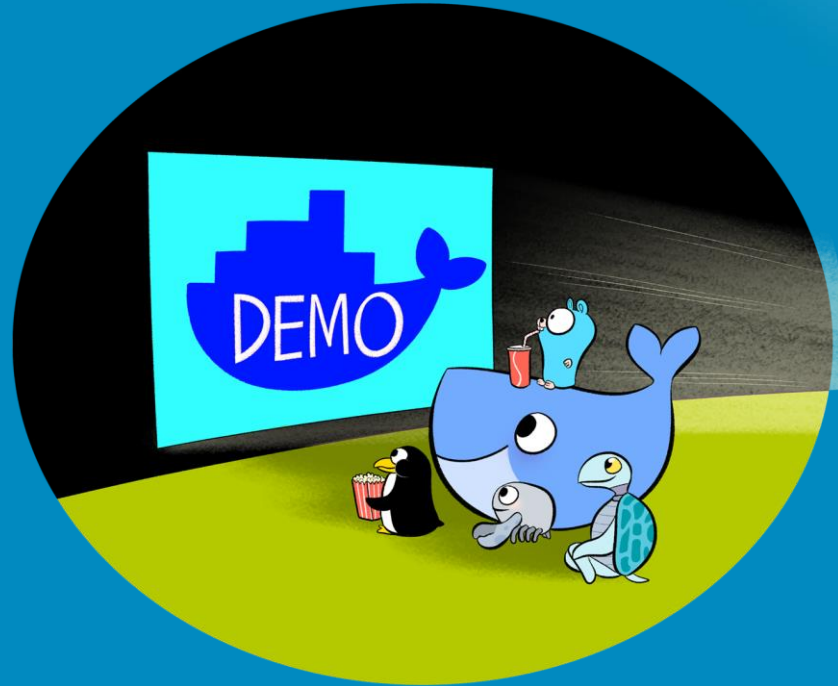
Kubernetes installation

As part of Docker Enterprise



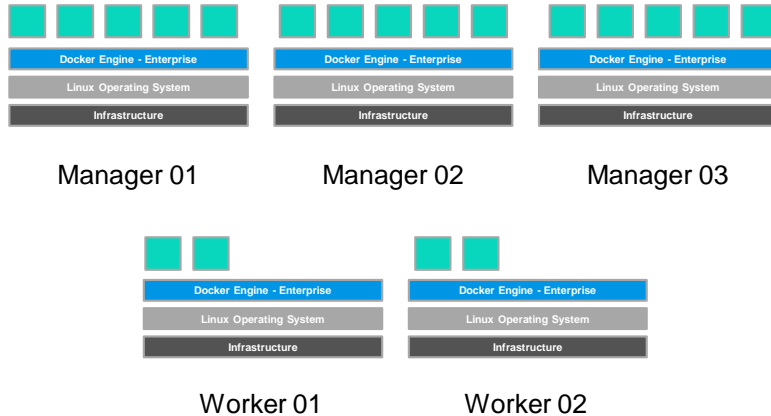
- Kubernetes is installed by default in all Universal Control Plane Installations.
- Everything in the Universal Control Plane Runs as Containers
- All we need is a Docker Engine, and a Bootstrapper Container and then everything happens automatically!

Kubernetes lifecycle



Growing your Kubernetes Cluster

As part of Docker Enterprise



- Seamlessly grow from 1 Kubernetes Manager to a Highly Available Deployment.
- Add your first worker to your $<nth>$ worker, all with the same command.
- UCP agent will configure TLS, start the Kubelet, and bring up Kubernetes Networking.

Growing your Kubernetes Cluster

As part of Docker Enterprise

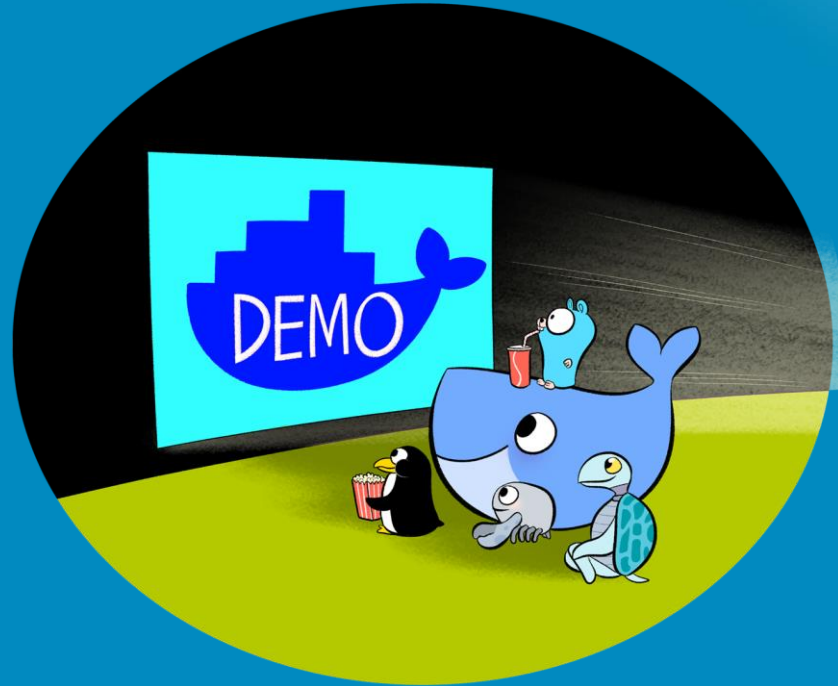
Adding a **Manager** Node:

```
docker swarm join --token SWMTKN-1-<join-token-manager>  
172.31.7.143:2377
```

Adding a **Worker** Node:

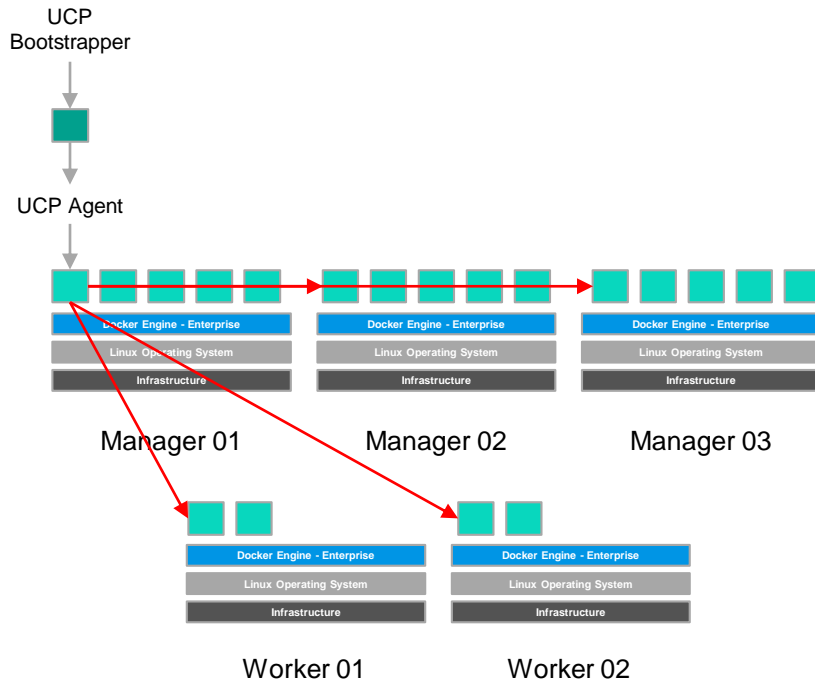
```
docker swarm join --token SWMTKN-1-<join-token-worker>  
172.31.7.143:2377
```

Kubernetes lifecycle



Upgrading your Kubernetes Cluster

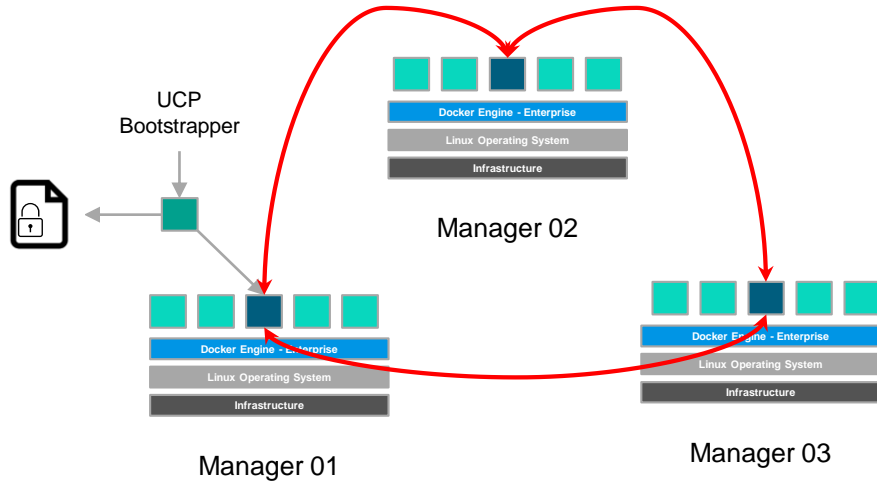
As part of Docker Enterprise



- Docker Enterprise has patch releases monthly and major releases every 6 months.
- Kubernetes may be patched as part of a Docker Enterprise patch release, Kubernetes will have a major upgrade as part of a Docker Enterprise major platform release.
- The UCP Bootstrapper tells the UCP Agents that a new upgrade is available, and then the whole cluster just upgrades itself :D.

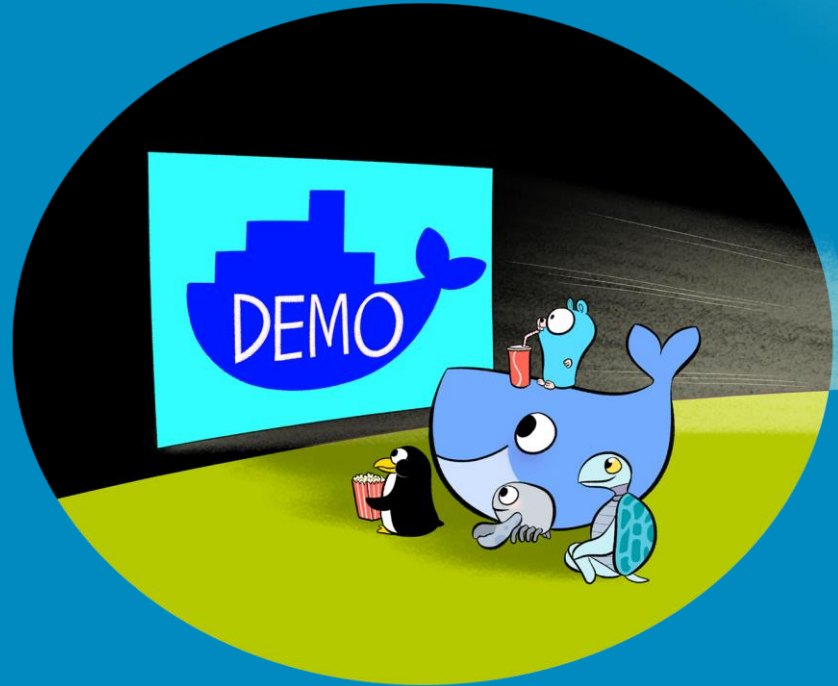
Backing up your Kubernetes Cluster

As part of Docker Enterprise



- Etcd is a centralised key value store, containing all UCP and Kubernetes Objects (It does not contain Swarm Objects).
- UCP handles the deployment, high availability and security of etcd.
- The UCP Bootstrapper container, can backup the etcd k/v. Backing up all Kubernetes objects (Workload Data stored in Volumes is not backed up).

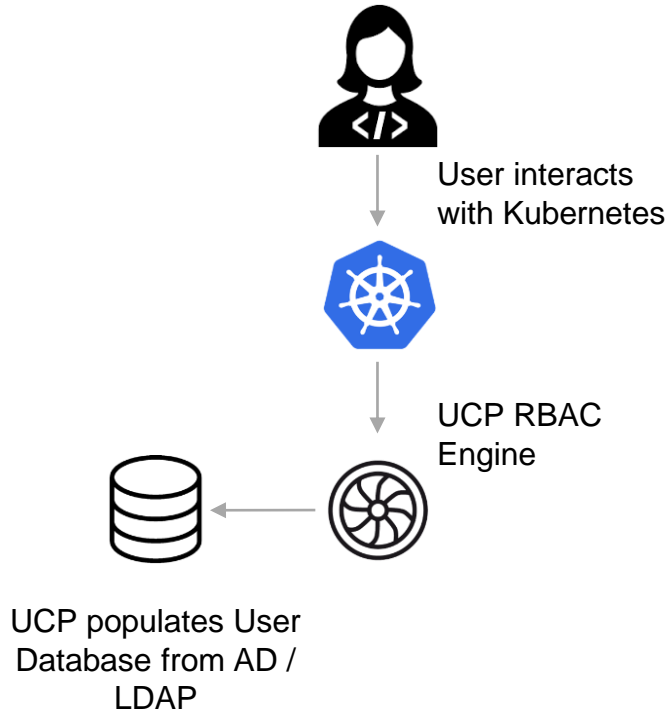
Kubernetes lifecycle



Kubernetes Security and Multi-Tenancy

Managing Kubernetes Users

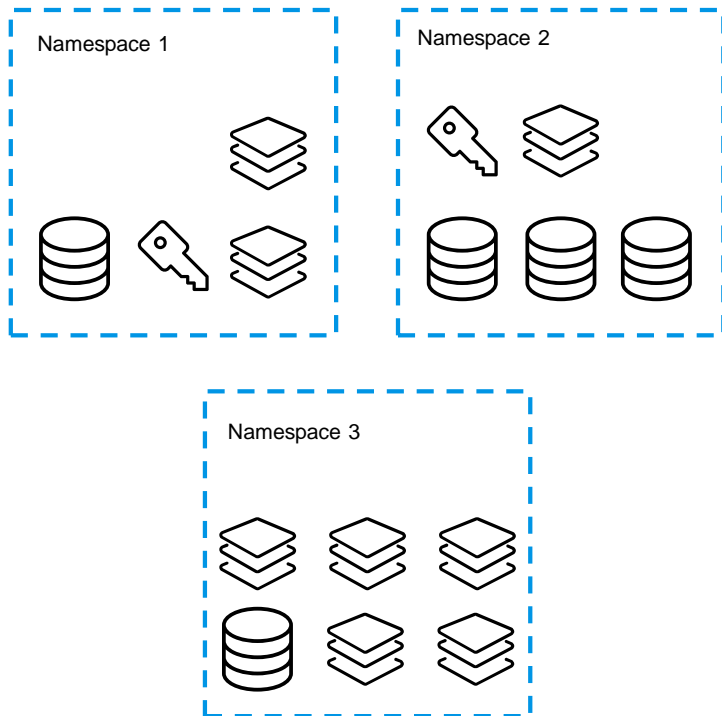
With Docker Enterprise: AuthN



- UCP integrates in to backend providers for syncing Users, Teams and Organisations.
 - LDAP / Active Directory
 - SAML v2 (Okta, Pingfederate)
- Integrate Kubernetes seamlessly to external processes and tooling through UCP Client Bundles.
- Authorisation across all endpoints. UCP UI, UCP API, UCP CLI, and DTR

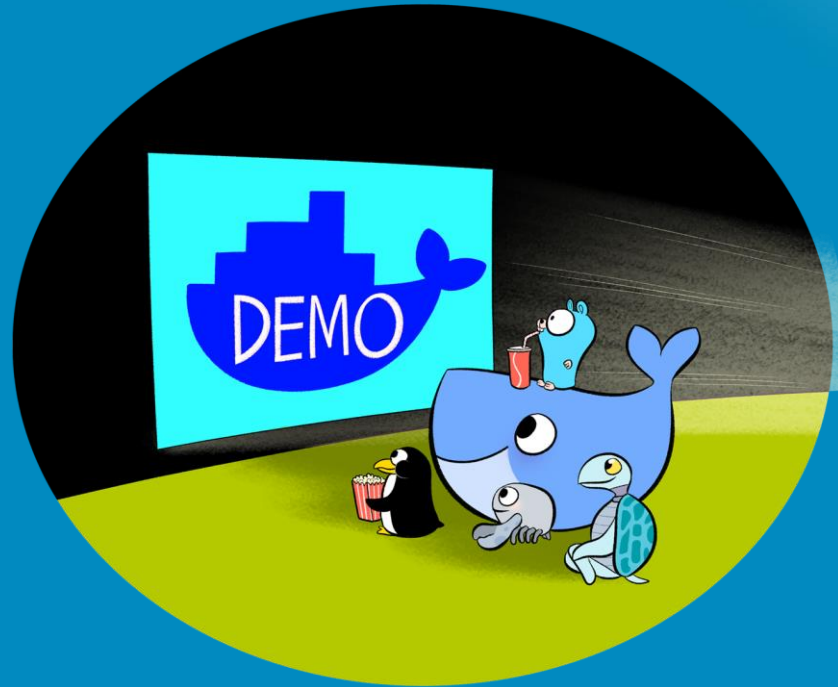
Managing Kubernetes Access Control

With Docker Enterprise: AuthZ



- UCP's RBAC Engine integrates with the Kubernetes RBAC APIs.
 - Roles - Matches Verbs (view, list...) with Objects (Pods, Persistent Volumes...)
 - RoleBindings - Matches a Role with a User / Team.
- UCPs Users / Teams can be bound to Logical Grouping of Resources (Kubernetes Namespaces) or Physical Grouping of Resources (Worker Nodes).

User Management in Docker Enterprise



Securing Kubernetes Networking

With Docker Enterprise and Project Calico

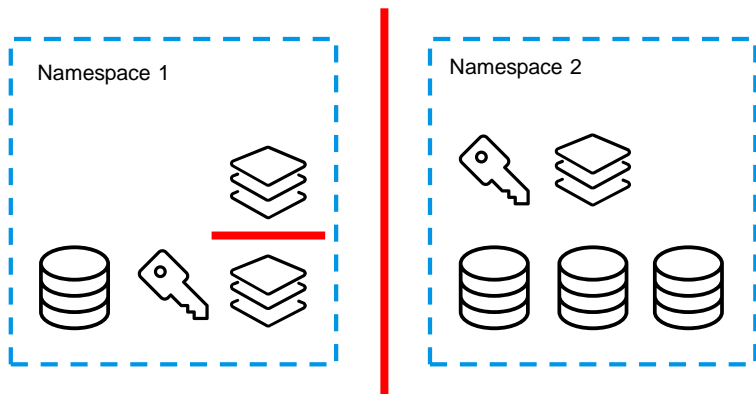
- Docker Enterprise provides a batteries included but swappable Kubernetes networking stack. Leveraging the Container Networking Interface (CNI).
- Project Calico is a fully supported CNI within Docker Enterprise, through a partnership with Tigera.
- Project Calico lifecycle is managed by the Universal Control Plane for Installation and Upgrades.



PROJECT
CALICO

Securing Kubernetes Networking

Kubernetes Network Policies with Project Calico



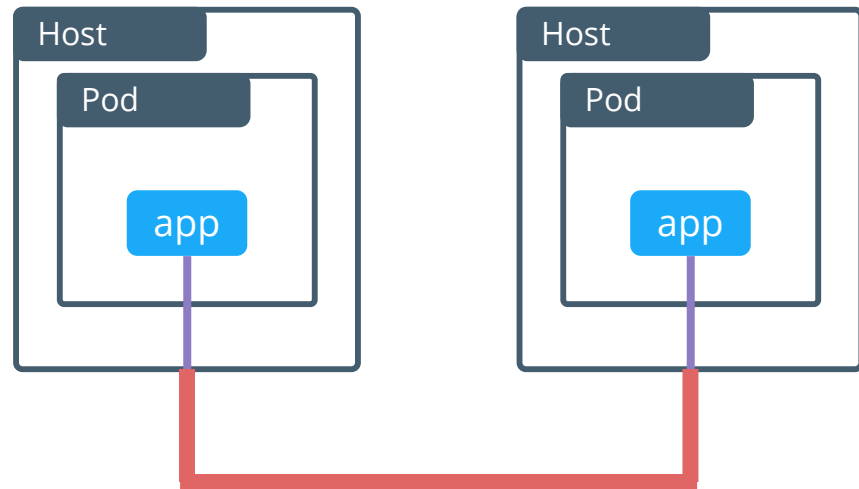
- By default Kubernetes operates a flat networking stack. All pods can communicate with all pods.
- Kubernetes Network Policy isolates pods, so that a pod can reject any connections that are not granted by a Network Policy.
 - Namespace to Namespace
 - Pod to Pod



Securing Kubernetes Networking

Network Encryption in Docker Enterprise

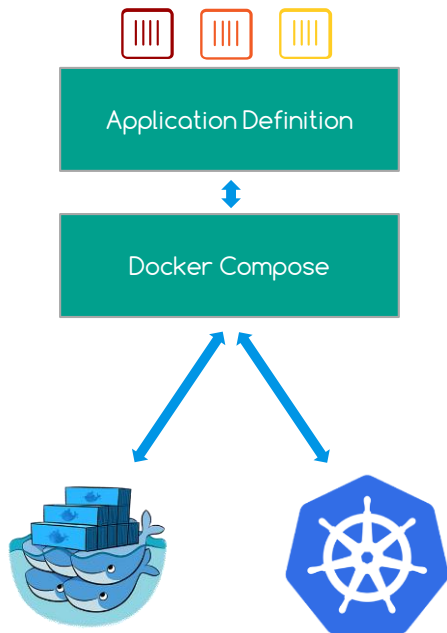
- Protect internal application traffic on untrusted infrastructure through external networking encryption.
- Once enabled all tenants and users get to take advantage without intervention or awareness from users.
- Traffic is encrypted by the in-kernel IPsec capabilities of Linux



Kubernetes Application Deployments

Docker Compose for Swarm & Kubernetes

Application Definition abstracts even the orchestrator away



- Write Application Definition once, agnostic of your Environment.
- All applications, whether they are Windows or Linux, Kubernetes or Swarm, can all be defined the same way.
- Use existing Docker Compose files and choose at runtime to deploy on either Swarm or Kubernetes.
- 100% Open Source
<https://github.com/docker/compose-on-kubernetes>

Docker Compose on Kubernetes is Now Open Source



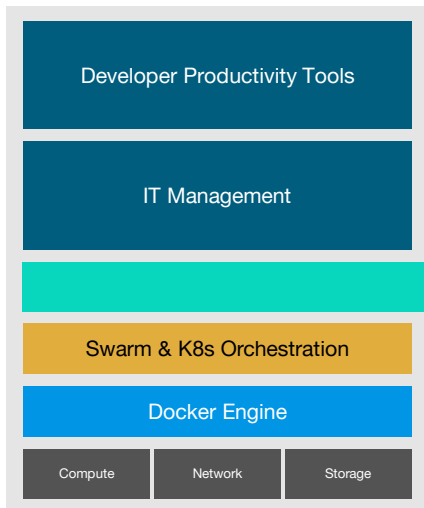
A high-level tool that simplifies Kubernetes configuration for improved developer productivity

Docker Enterprise Container Platform

DEVELOPERS



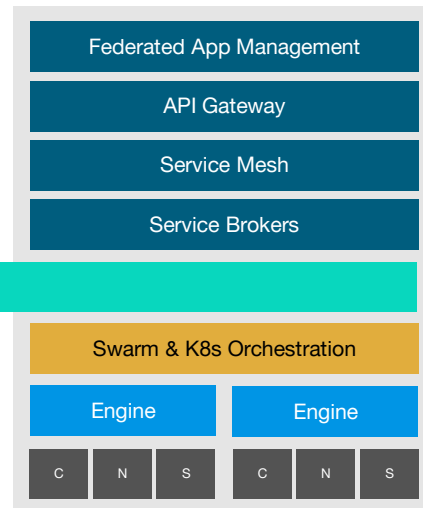
Docker Desktop Enterprise



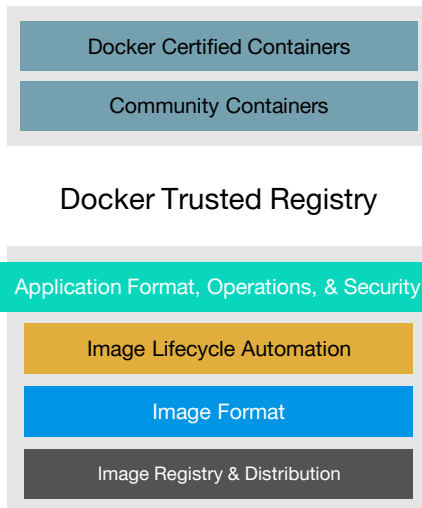
OPERATORS



Docker Universal Control Plane



Docker Hub



Application Format, Operations, & Security

Want to try it out?

`https://www.docker.com/eval`

Thank you!