

Introduction aux micro-services & Cloud Natives



ib
ceqos

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

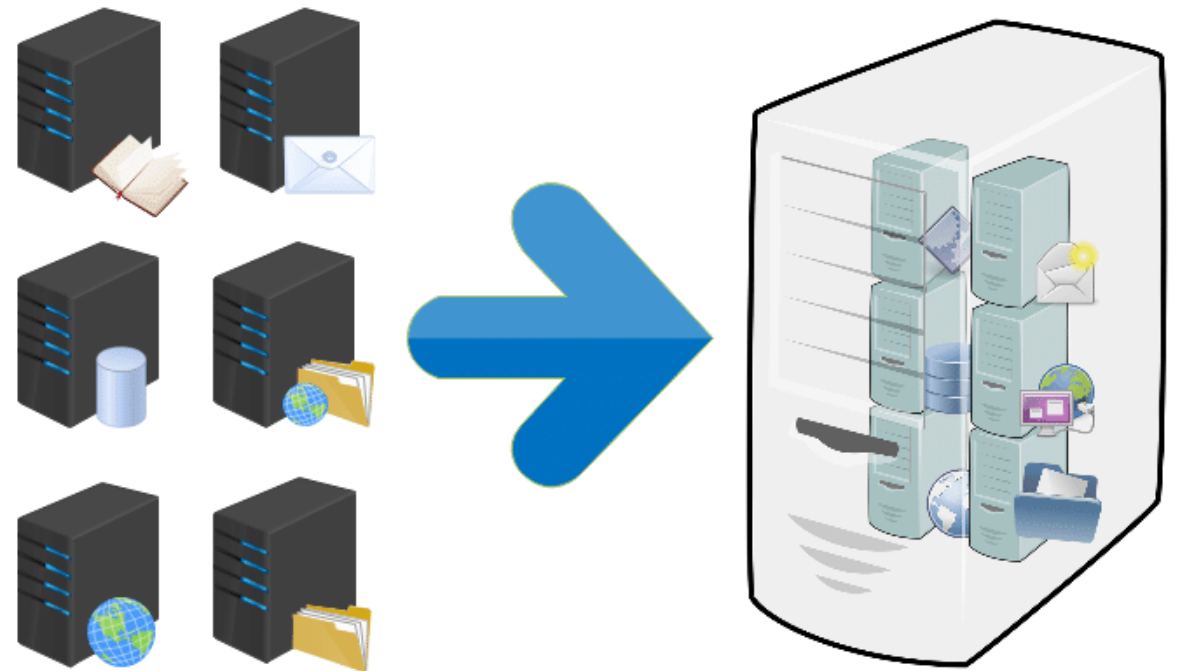
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

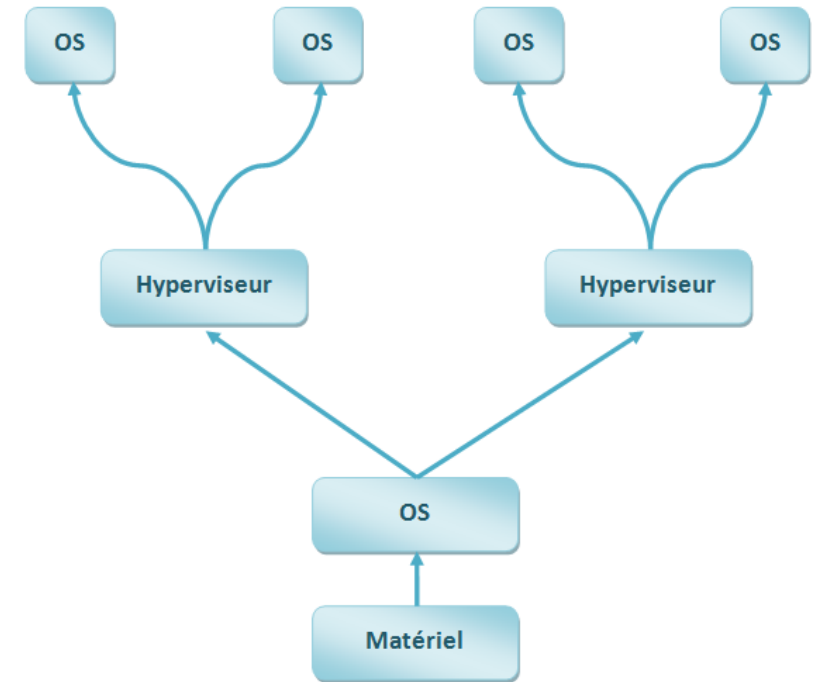
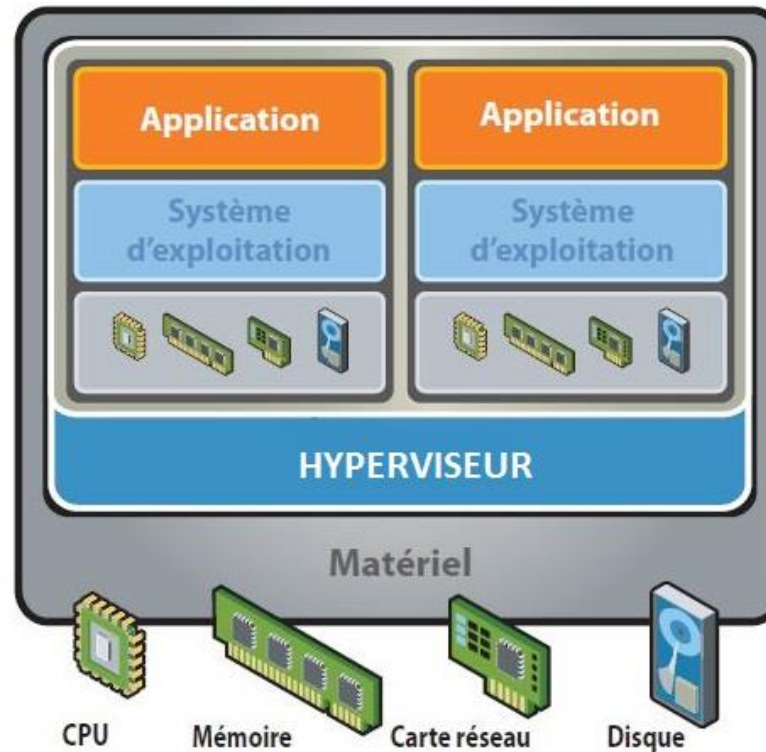
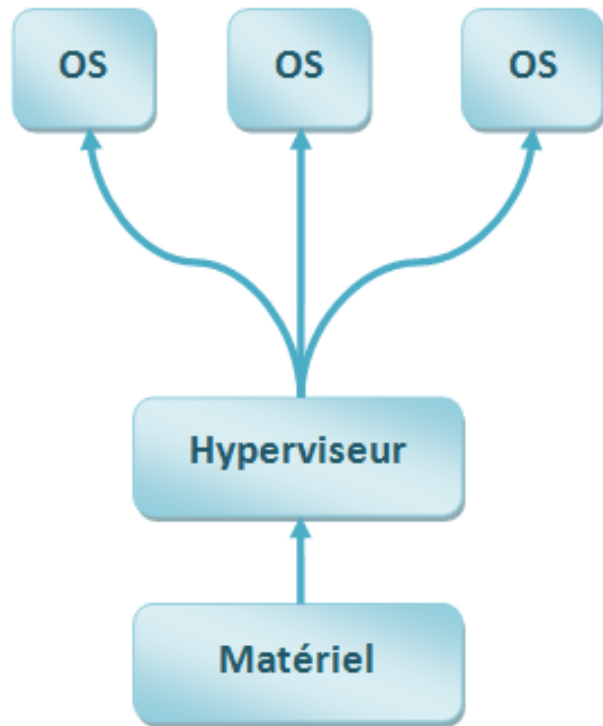
Introduction à l'orchestration avec Docker Swarm

Mini-projet

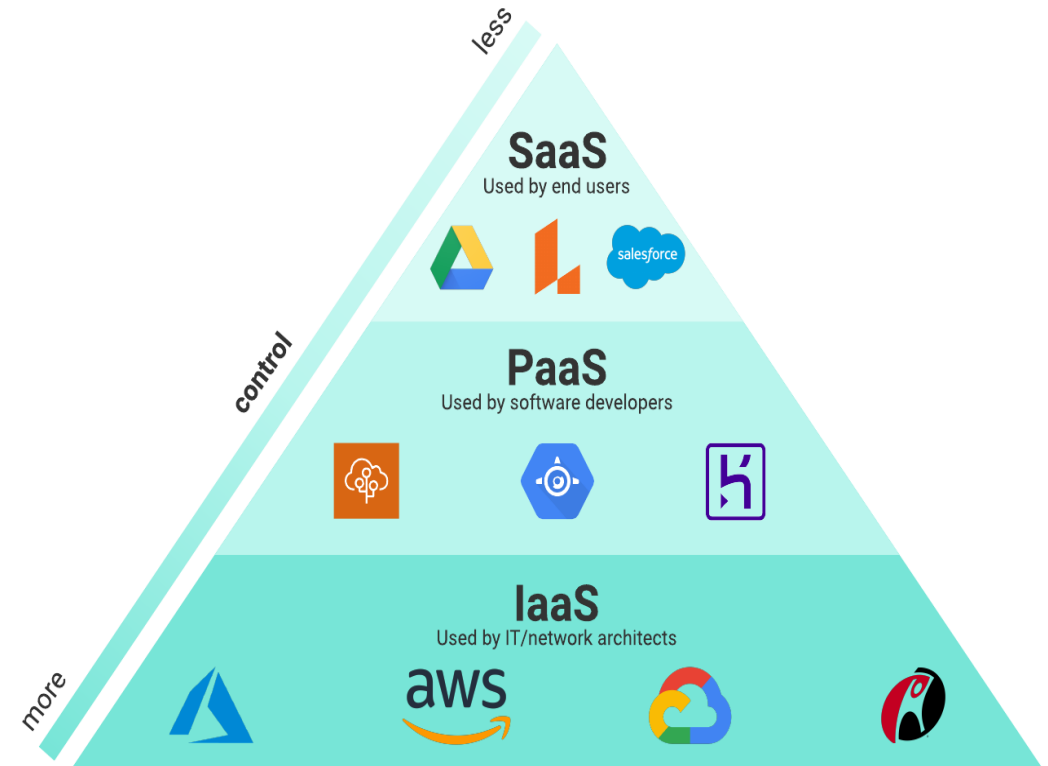
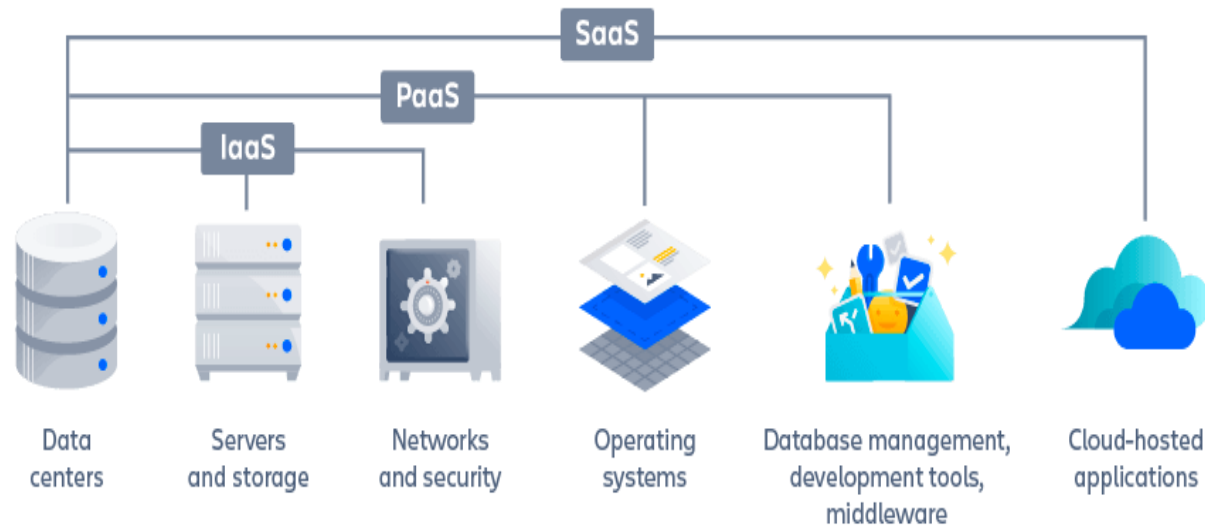
Presentation des concepts(1/8):Virtualisation 1/2



Presentation des concepts(2/8): Virtualisation 2/2



Presentation des concepts(3/8): Cloud Computing 1/3



Presentation des concepts(4/8):

Cloud Computing 2/3



Cloud Public

- Evolutivité
- Rentabilité
- Stockage illimité
- Paiement à l'utilisation

Cloud Hybride

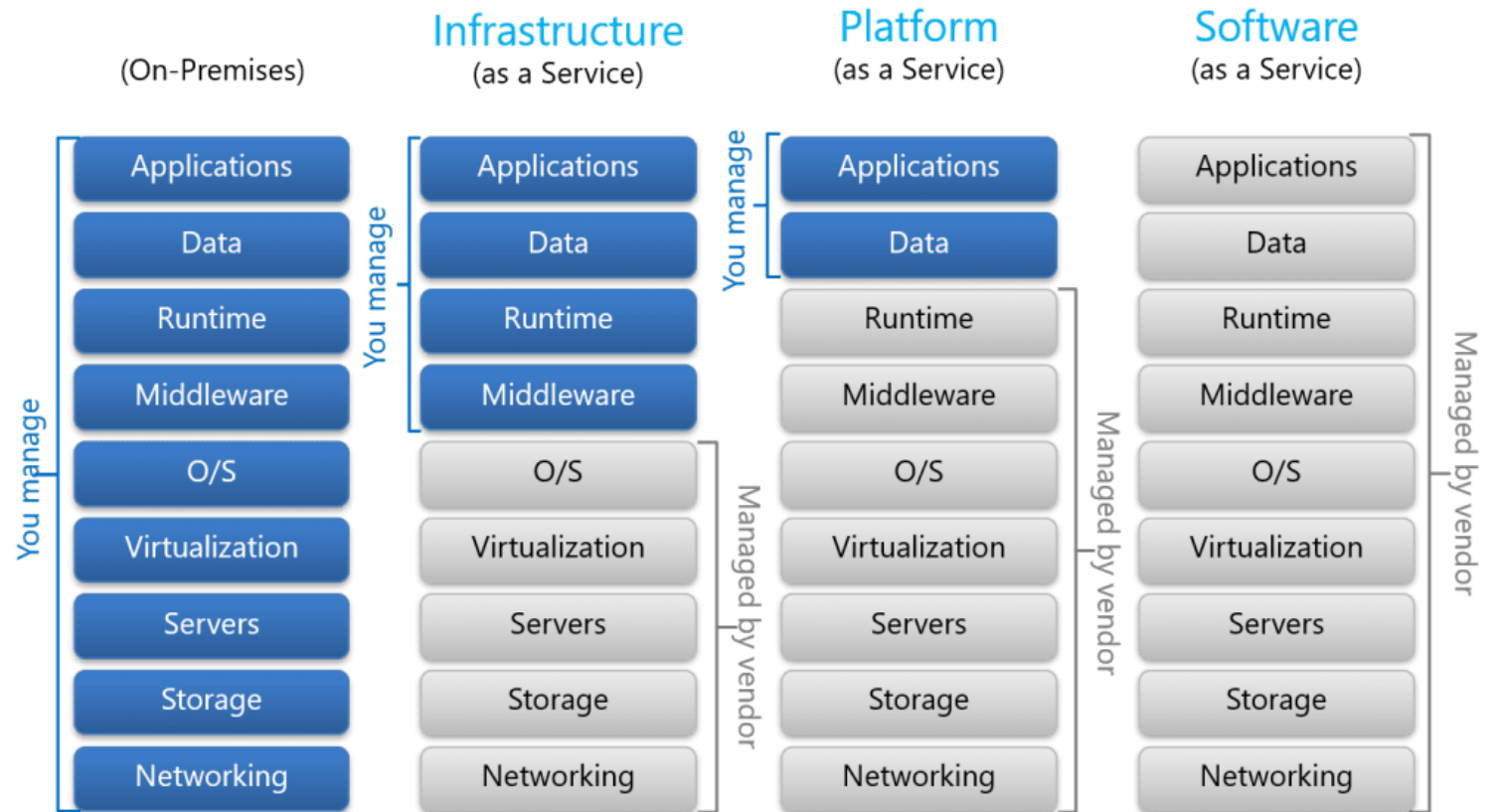
- Evolutivité
- Haute sécurité
- Flexibilité
- Rentabilité
- Stockage illimité

Cloud Privé

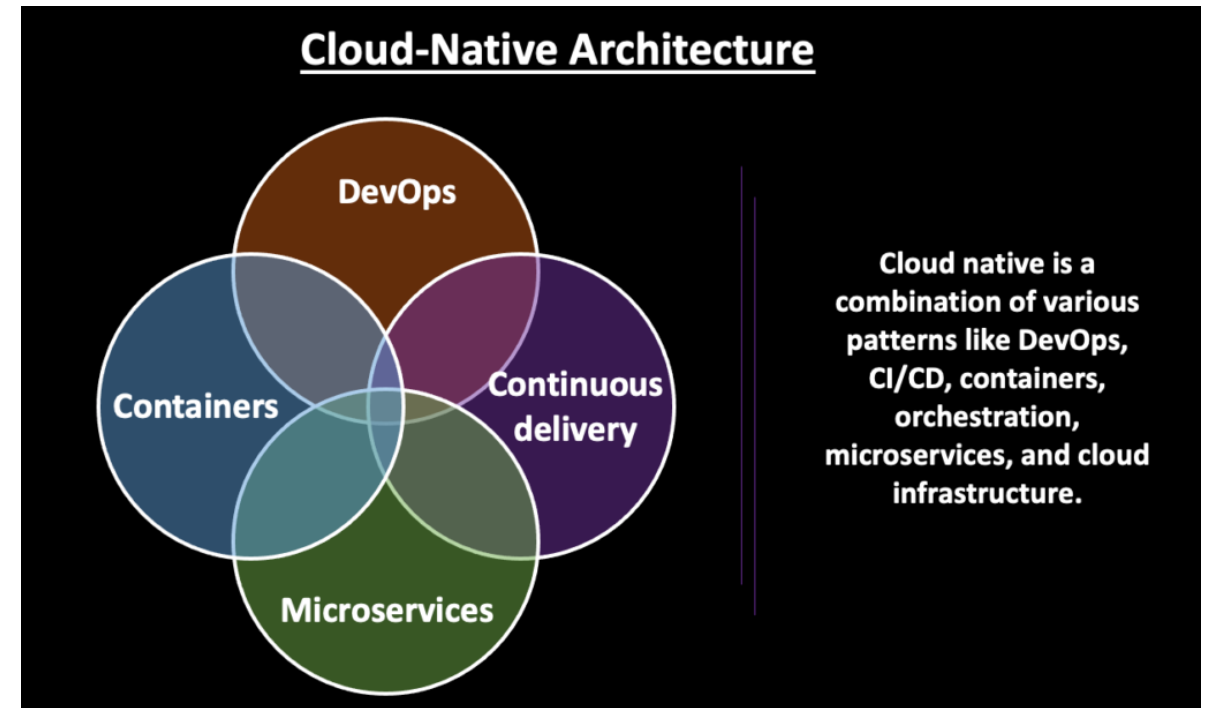
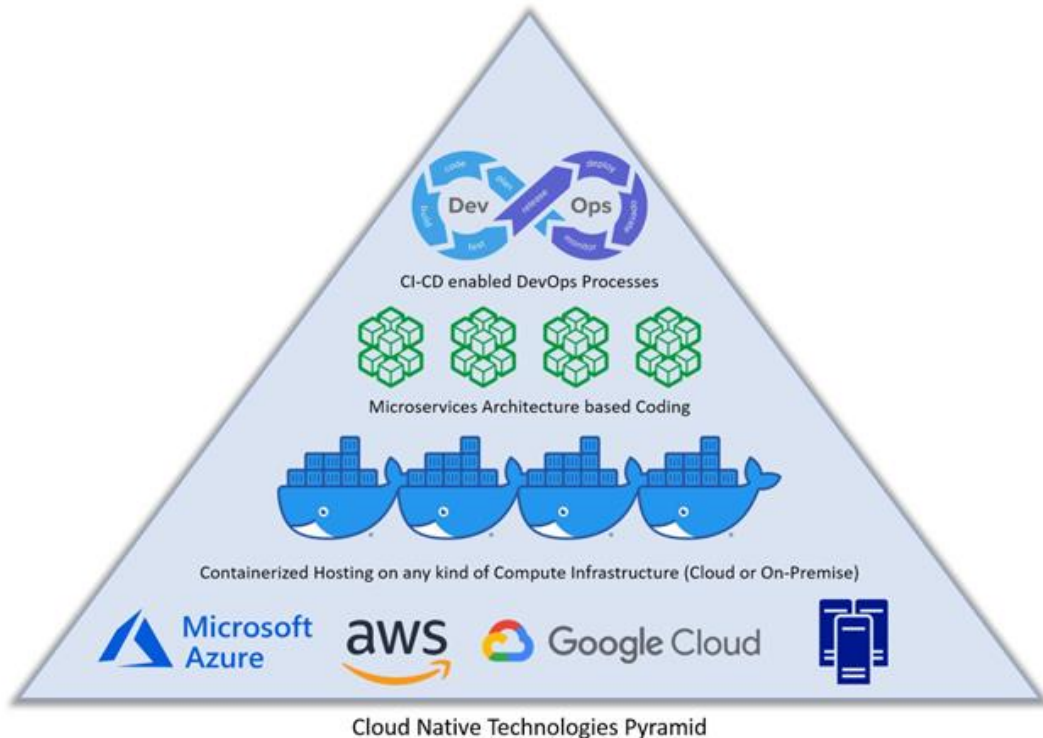
- Utilisateur unique
- Haute sécurité
- Flexibilité
- Totalement personnalisable



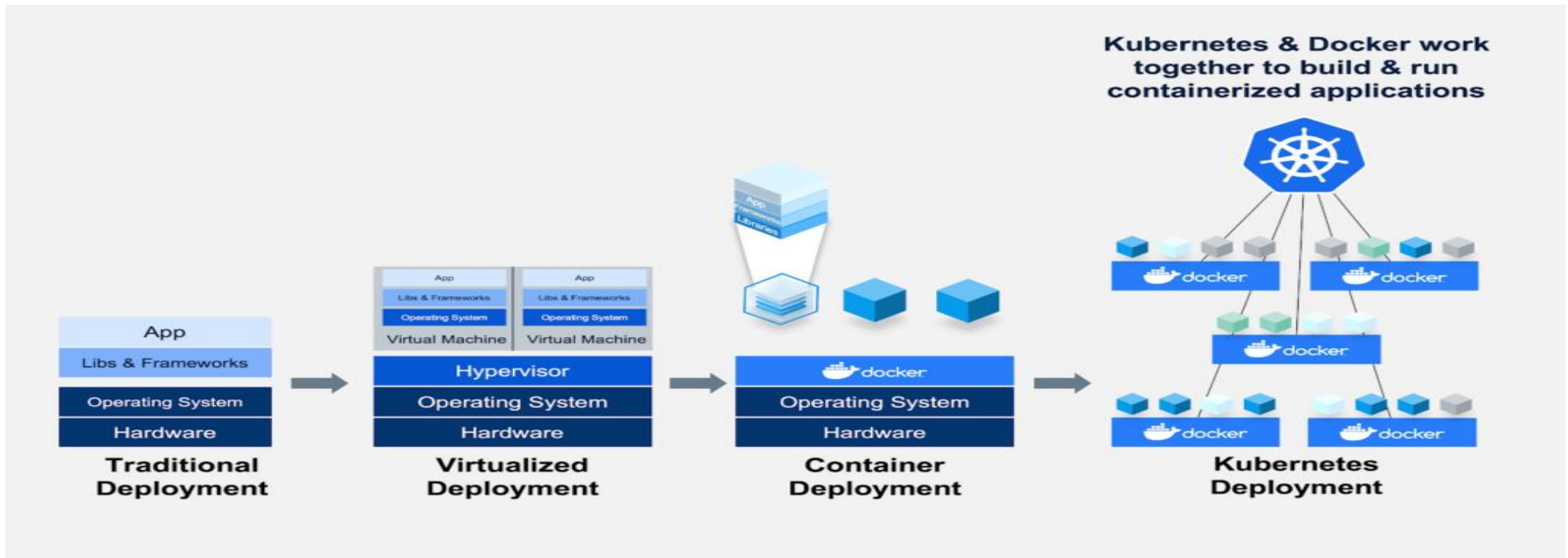
Presentation des concepts(5/8): Cloud Computing (3/3) Share responsabilities



Presentation des Concepts(6/8): Application Cloud Native

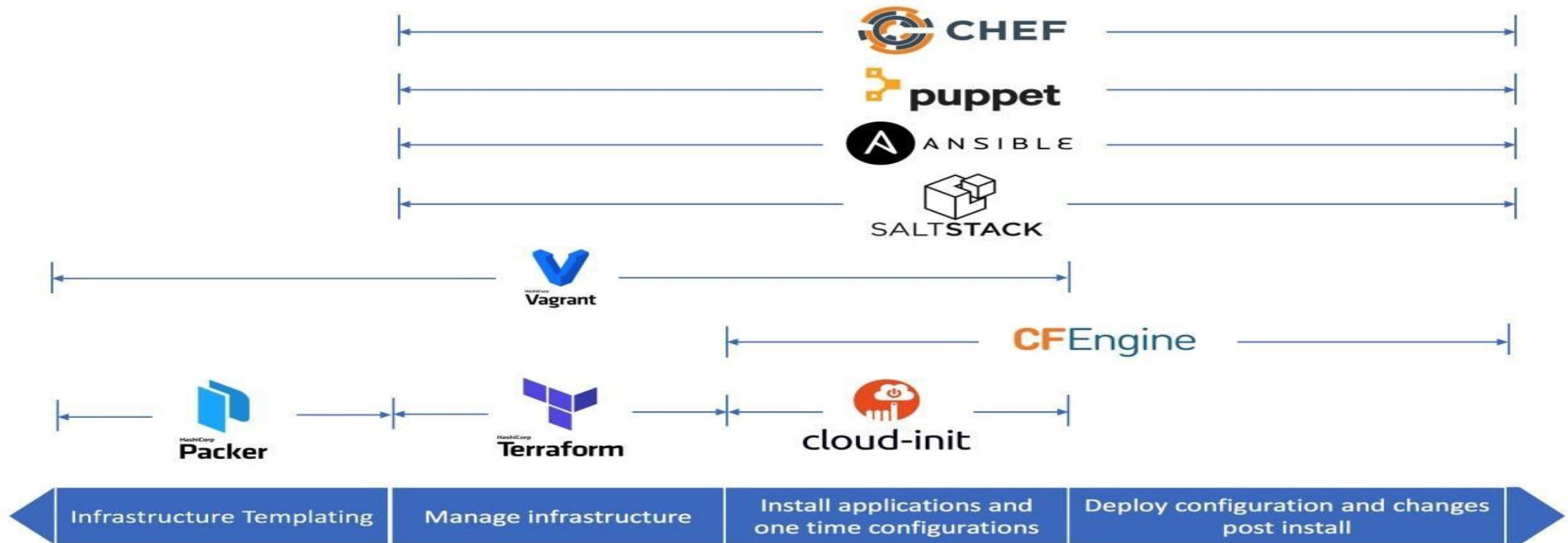


Presentation des Concepts(7/8): Cattle VS Pet



Presentation des concepts(8/8):

IAC Tools



Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

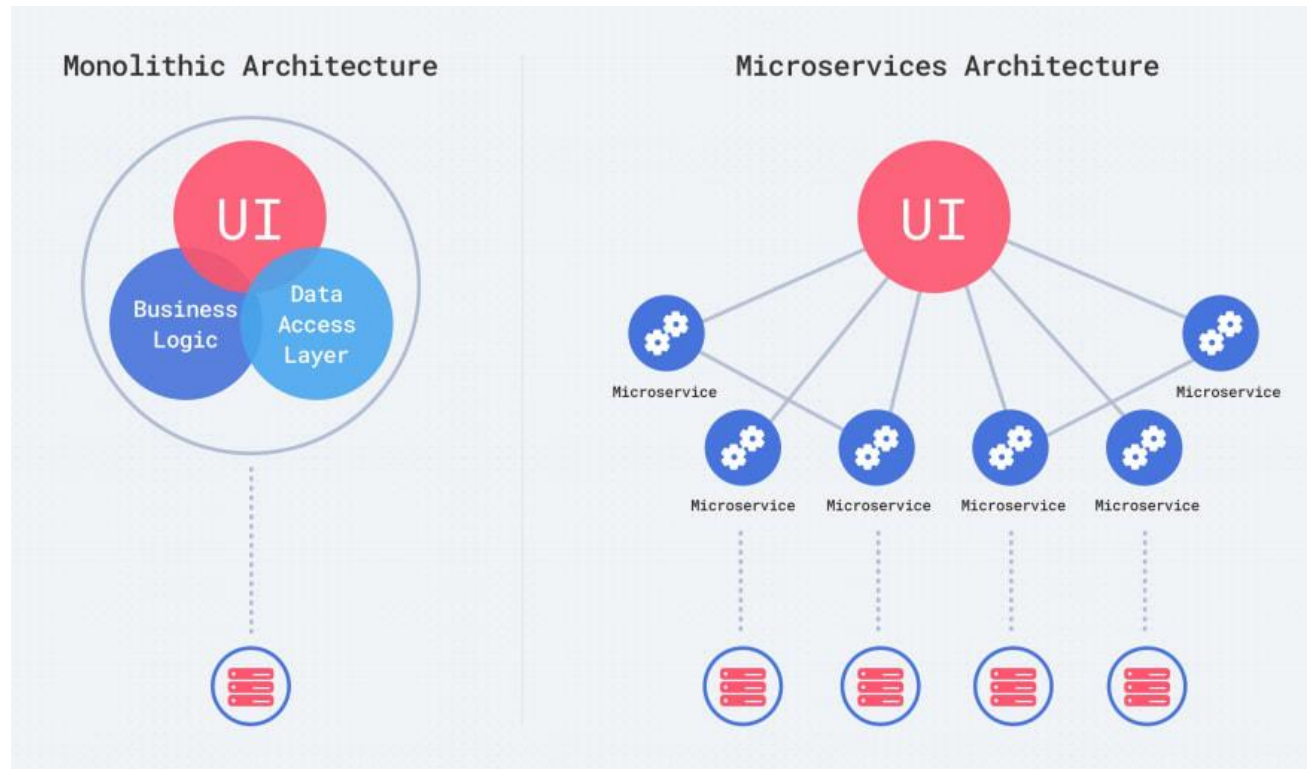
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

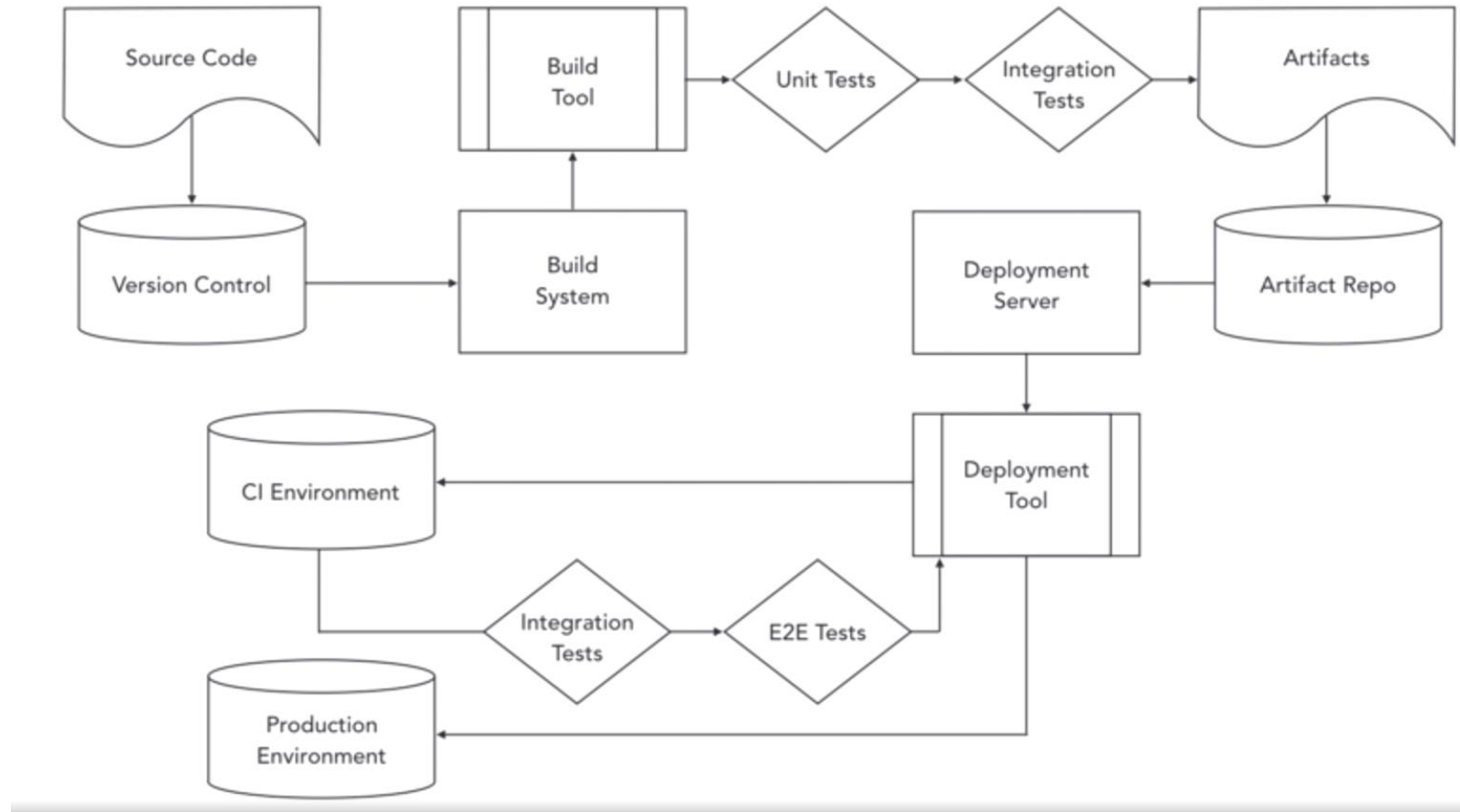
Mini-projet

Introduction à la conteneurisation

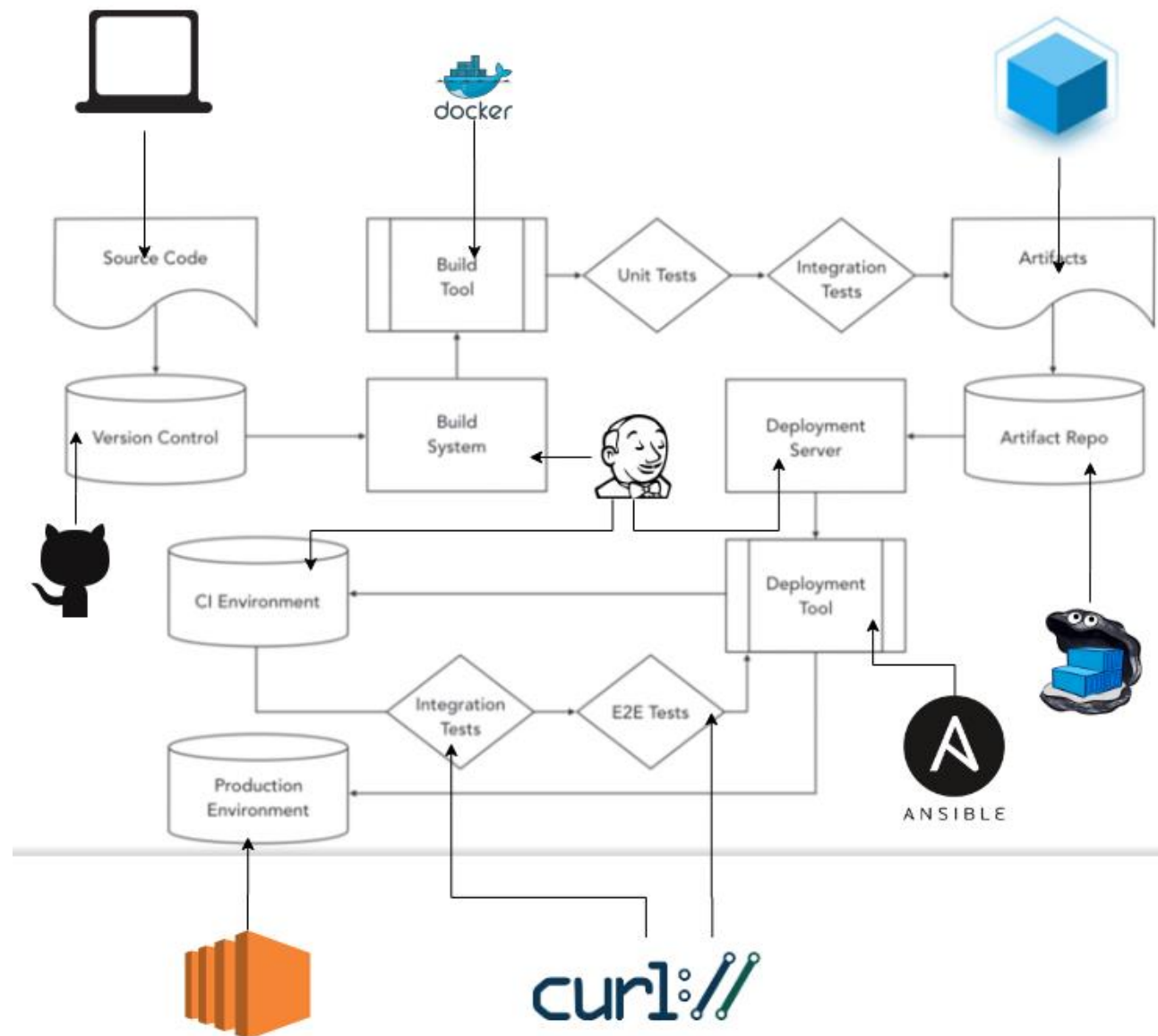


- Agilité
- Flexibilité
- Résilience
- Scalabilité

Cycle de vie des applications (1/2)



Cycle de vie des applications (2/2)



12 Factor app

12 Factor Application Pattern



Codebase



Dependencies



Config



Backing Services



CI/CD



Stateless



Port Binding



Scalability



Disposability



Fast Releases



Logging



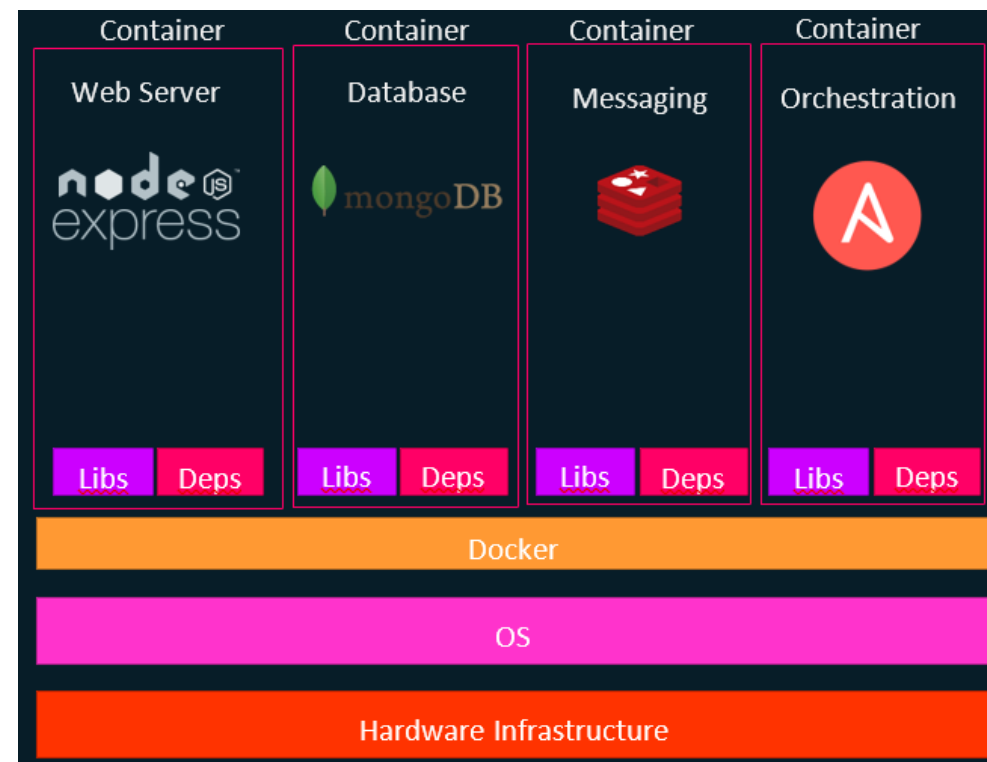
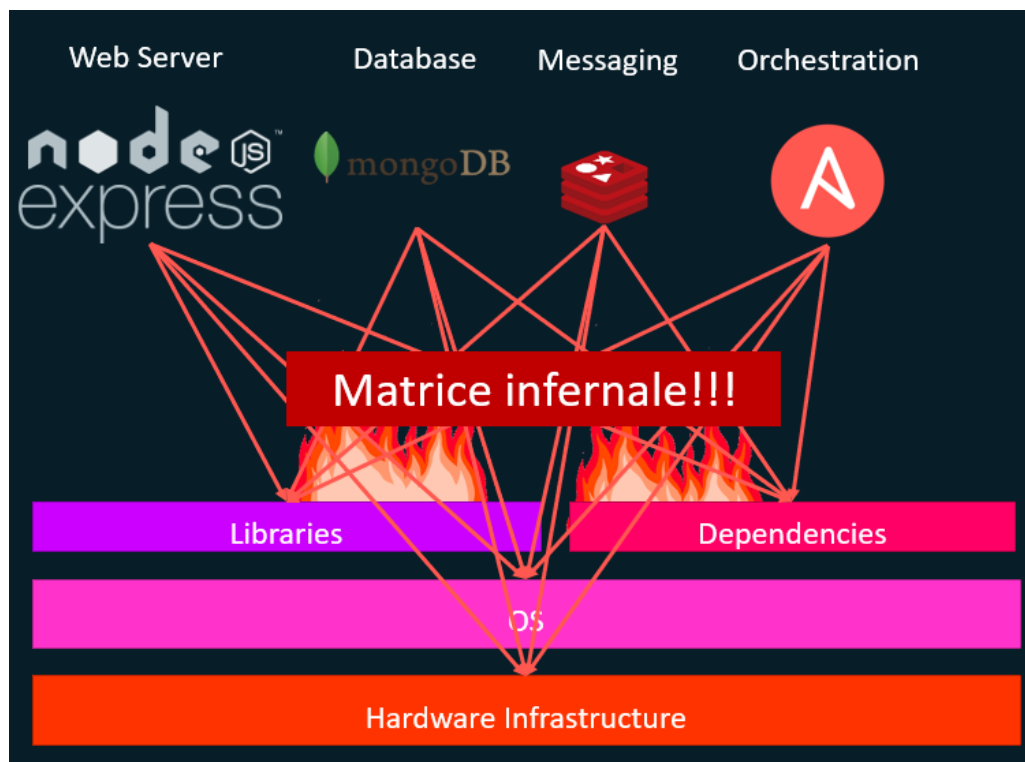
Admin Process

Cloud Computing

Pros and Cons of Cloud Computing



Pourquoi la conteneurisation?



Qu'est ce qu'un Container?



Processus
Réseaux
volumes



Processus
Réseaux
volumes



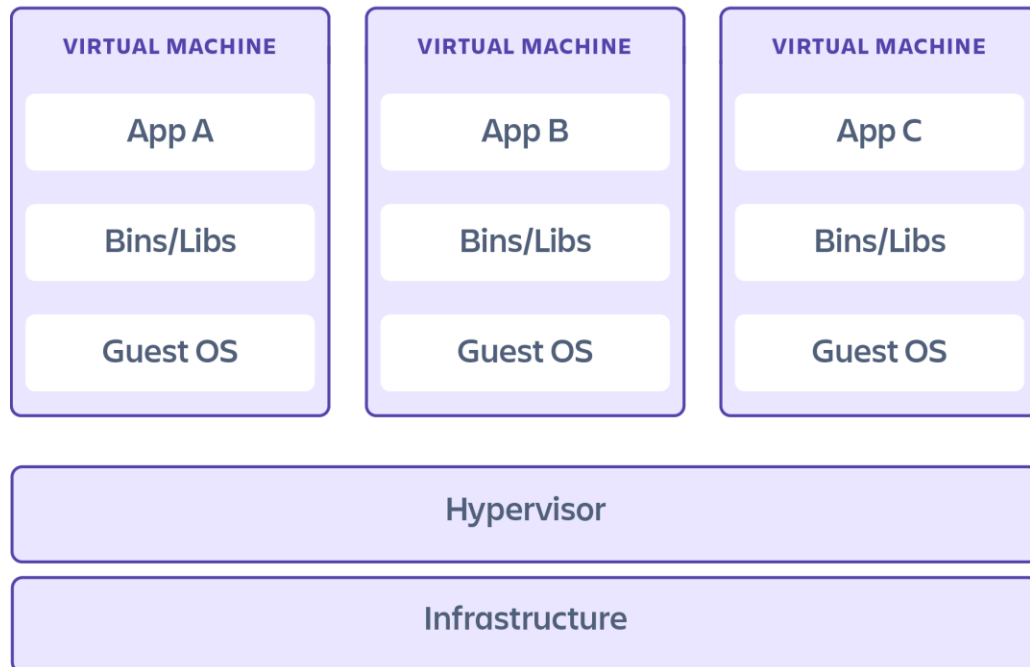
Processus
Réseaux
volumes

DOCKER

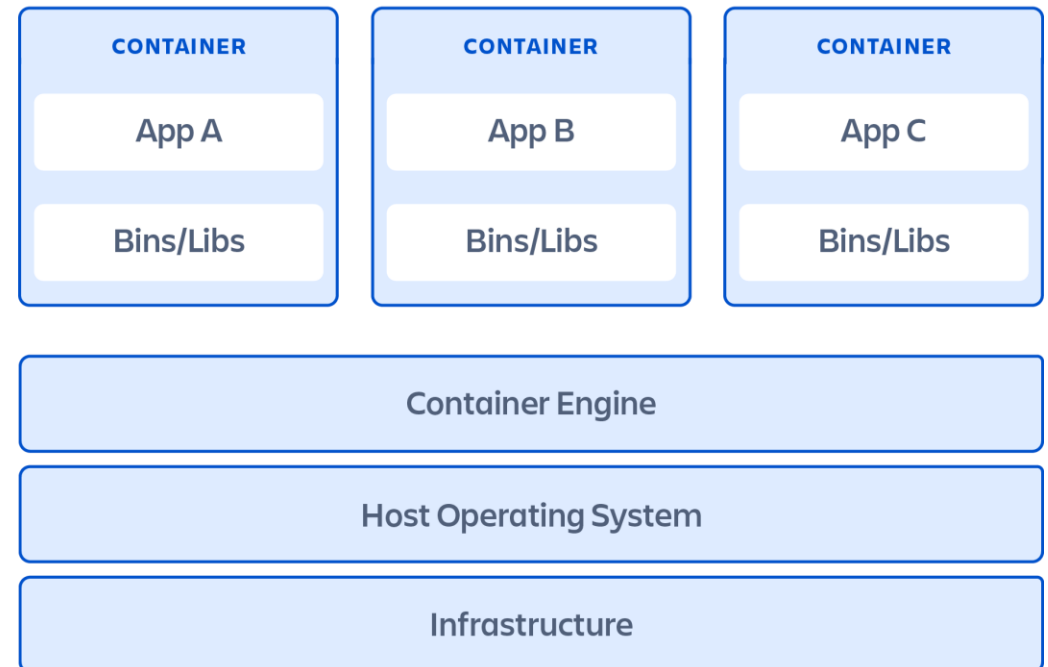
OS

Virtual M. VS Containers

Virtual machines



Containers



Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Gestion des reseaux

Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Installation de docker (1/5): Editions

Docker Enterprise Edition (EE) and Community Edition (CE)

Enterprise Edition (EE)

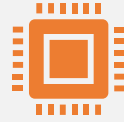
- CaaS enabled platform subscription (integrated container orchestration, management and security)
- Enterprise class support
- Quarterly releases, supported for one year each with backported patches and hotfixes.
- Certified Infrastructure, Plugins, Containers

Community Edition (CE)

- Free Docker platform for “do it yourself” dev and ops
- Monthly Edge release with latest features for developers
- Quarterly release with maintenance for ops



Installation de docker (2/5): machine locale



Mac: Docker for Mac



Windows: Docker for Windows



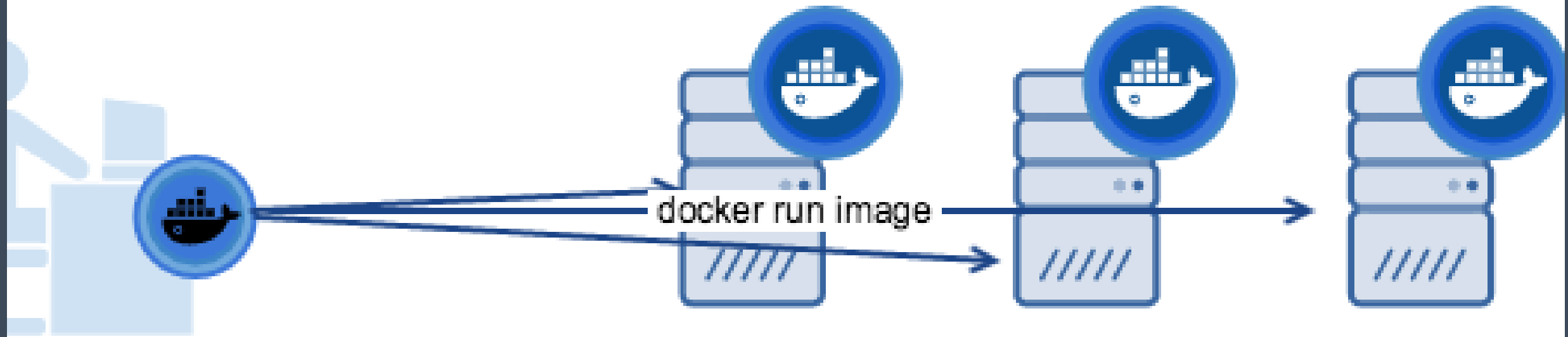
Linux: Docker for Linux Distribution
(centos, Debian, ubuntu ...)



Mac + Windows: Docker Toolbox

Installation de docker (3/5): Serveur

- Linux Serveur: Docker for Linux Distribution
- Windows Serveur:
 - 2016: Docker for Windows
 - 2019: Rôle et Fonctionnalité



Installation de docker (4/5): Docker-machine

Installation de docker (5/5): Cloud Solutions



- ACS
- ECS + Fargate
- GCP
- Autres (OVH, Docker Cloud ...)

TP-0: Découvrir la plateforme de TP

- Accès
- Labs
- Durée d'une session
- Données sensibles
- Agrandir la fenêtre du terminal
- Connexion ssh
- Installation de paquet
- Ouverture de port

TP-1: Installation et Test de Docker

- Installation de Docker sur Centos
- Tâche de post installation
- Vérification de l'installation
- Utiliser la documentation Docker afin d'être à jour
- Lancement de votre premier conteneur (nginx)
- Réaliser la même tâche sur la plateforme de labs
- Utilisation des variables d'environnement

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

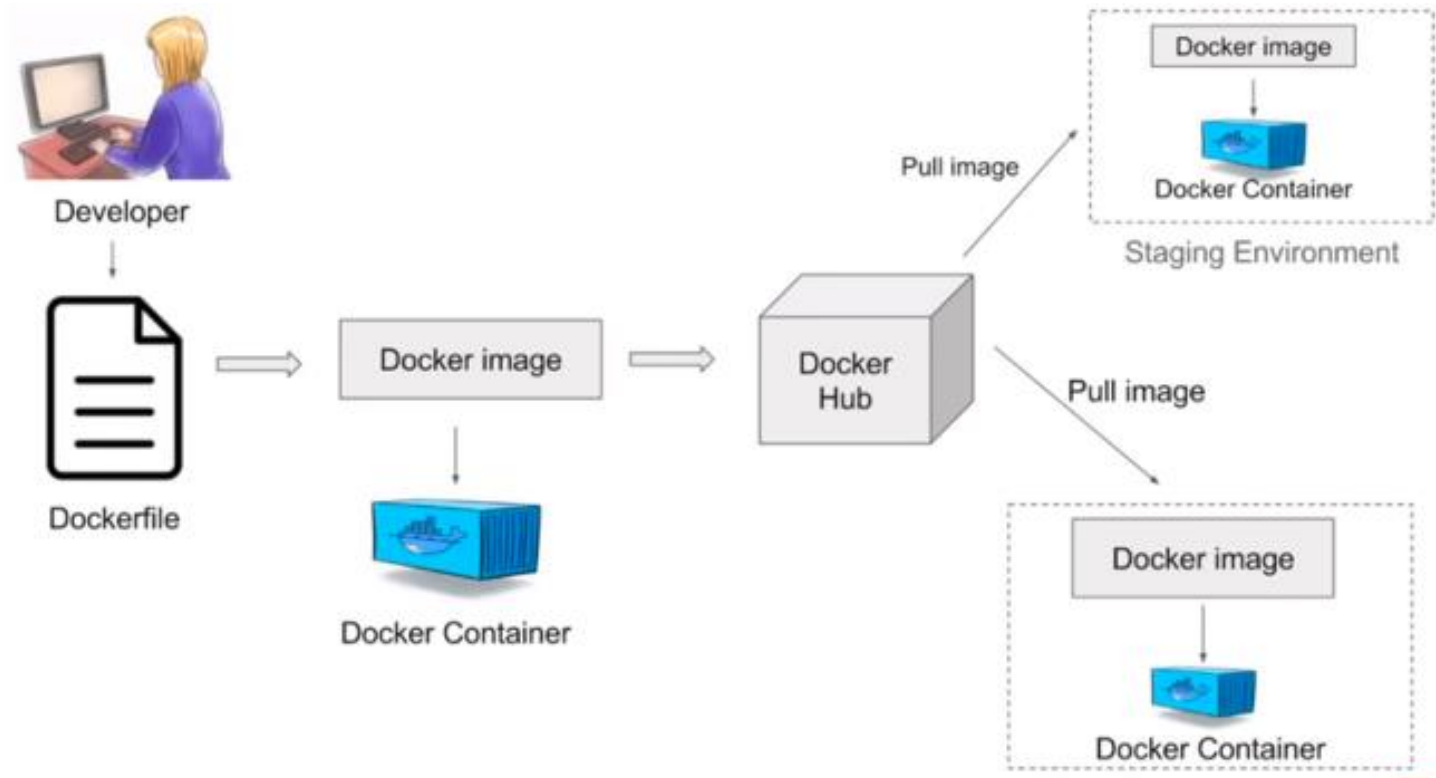
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

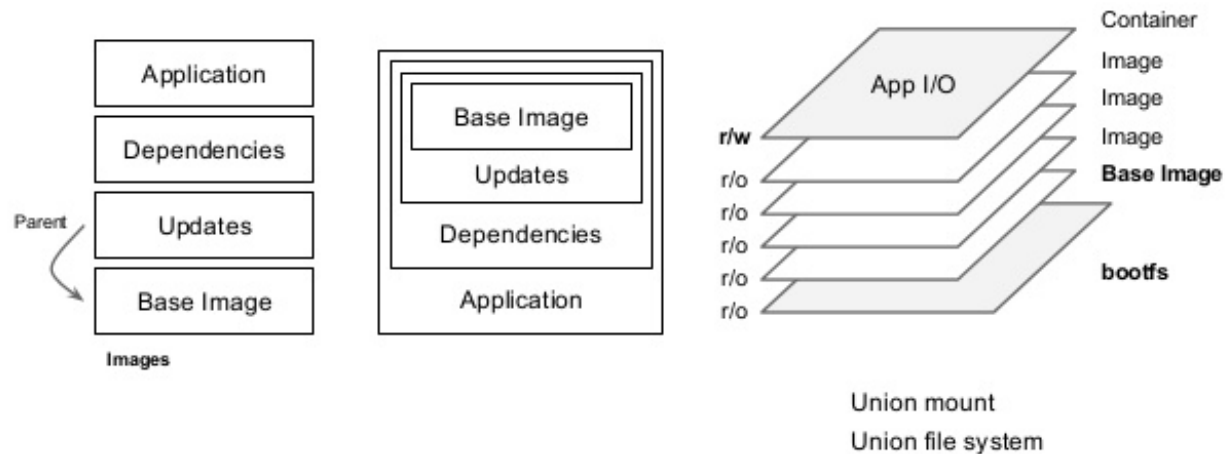
Mini-projet

Gestion des images (1/5): Workflow



Gestion des images (2/5): Anatomie d'une image

Image Anatomy



- Lecture seule
- Réutilisation
- Couche (Layer)

Gestion des images (3/5): Instructions

- **FROM** : Specify the base image
- **MAINTAINER** : Specify the image maintainer
- **RUN** : Run a command
- **ADD** : Add a file or directory
- **EXPOSE** : expose ports to be accessed
- **ENV** : Create an environment variable
- **CMD** : What process to run when launching a container from this image.

Gestion des images (4/5): Exemple

 **Dockerfile** 455 Bytes 

Edit

Web IDE

```
1 FROM python:2.7-stretch
2 #Maintainer of image
3 LABEL maintainer="175777@supinfo.com"
4 #Install of dependencies
5 RUN apt-get update -y && apt-get install python-dev python3-dev libsasl2-dev python-dev libldap2-dev libssl
6 RUN pip install flask flask_httpauth flask_simpleldap python-dotenv
7 #COPY SOURCE CODE IN THE IMAGE
8 COPY student_age.py /
9 #CREATE DATA FOLDER
10 VOLUME [ "/data" ]
11 #EXPOSE PORT
12 EXPOSE 5000
13 #RUN CODE
14 CMD [ "python", "./student_age.py" ]
```

Gestion des images (5/5): Resgistre



TP-2: Gérez vos images Docker

- Créez votre propre image (conteneuriser l'application <https://github.com/diranetafen/static-website-example>)
- Testez votre image
- Gérez vos images sur le dockerhub

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

Gestion du stockage

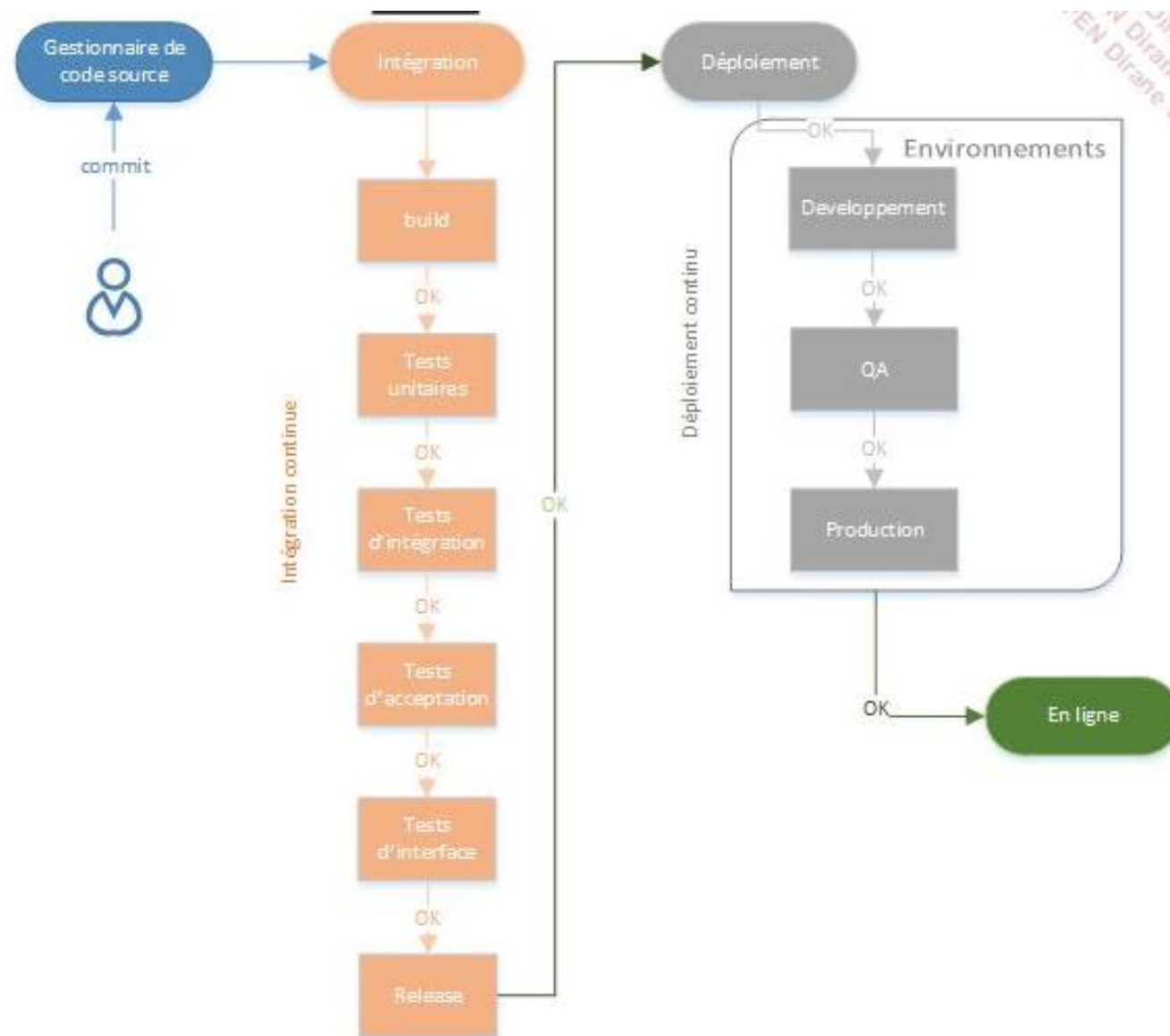
Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Introduction au CI/CD(1/11)

- Intégration en continu
- Test en continu
- Déploiement en continu
- Automatisation 😊
- Docker



Introduction au CI/CD 2/11: Github actions



GitHub Actions

Introduction au CICD (3/11): Github actions (1/9) : Présentation



-
- Disponible depuis Novembre 2019
 - Basé sur Azure Pipelines
 - YAML-based configuration
 - Runners sous Windows, Linux, macOS ou self-hosted
 - Disponible en Free, Pro, Team, Enterprise Cloud

Introduction
au CI/CD
(4/11): Github
actions (2/9) :
Pricing

Public repositories

Free! ✨

Speaker: Alexey Golub

Private repositories

Product	Storage	Minutes (monthly)
GitHub Free	500 MB	2,000
GitHub Pro	1 GB	3,000
GitHub Team	2 GB	10,000
GitHub Enterprise Cloud	50 GB	50,000

Operating system	Minute multiplier
Windows	2
Linux	1
macOS	10

Operating system	Per-minute rate
Windows	\$0.016
Linux	\$0.008
macOS	\$0.080

  @Tyrrrz

Introduction
au CI/CD
(5/11): Github
actions (3/9) :
Triggers

- **GitHub API events** ⚡
push, pull_request, issues, release, and 20 others
- **Schedule** 🕒
Cron syntax, e.g.: */15 * * * *
- **Manual** 🌐
POST to /repos/:owner/:repo/dispatches

Introduction au CICD (6/11): Github actions (4/9) : Triggers Examples

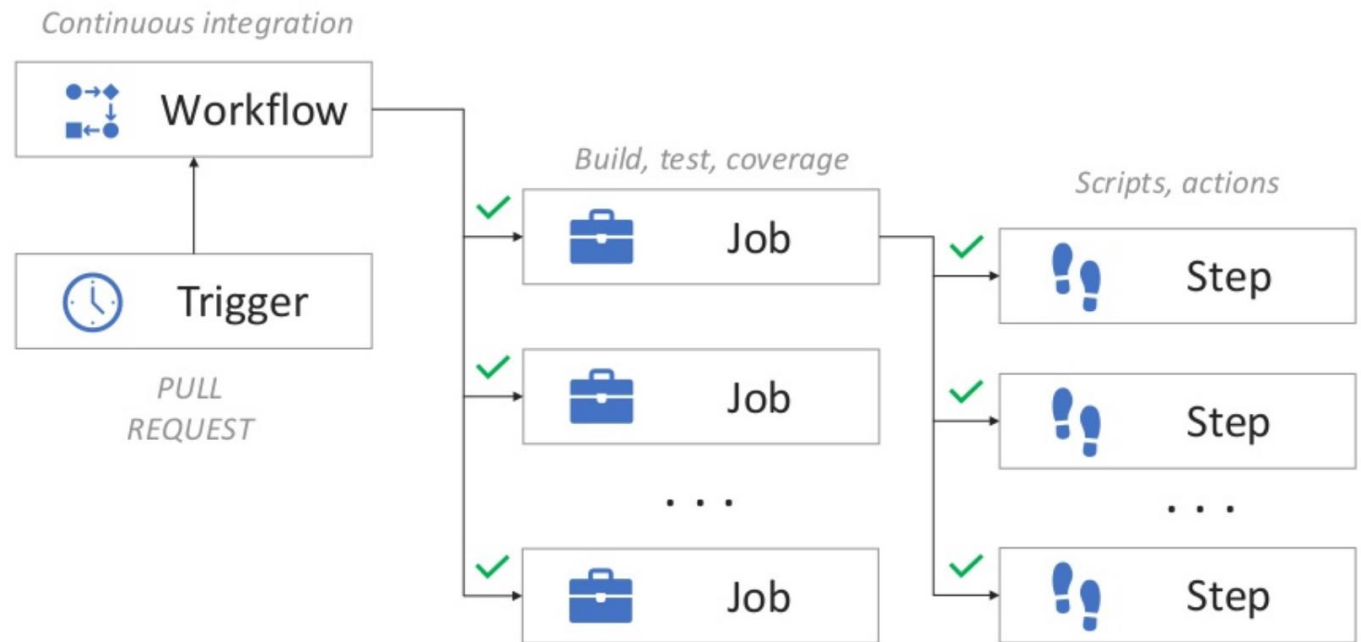
```
# Trigger on push events on specific branches
on:
  push:
    branches:
      - 'master'
      - 'release/*'
```

```
# Trigger on manual dispatch
on: repository_dispatch
```

```
# Trigger every midnight UTC
on:
  schedule:
    - cron: '0 0 * * *'
```

```
# Trigger when an issue is opened or labeled
on:
  issues:
    types: [opened, labeled]
```

Introduction au CI/CD (7/11): Github actions (5/9) : Composants



Introduction au CI/CD (8/11): Github actions (6/9) : Structure

```
name: CI
on: [push, pull_request]
jobs:
  build:
    runs-on: windows-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v1
      - name: Install .NET Core
        uses: actions/setup-dotnet@v1
        with:
          dotnet-version: 3.1.100
      - name: Build & test
        run: dotnet test --configuration Release
      - name: Build & publish
        run: dotnet publish LightBulb/ -o LightBulb/bin/Publish/ --configuration Rel
      - name: Upload build artifacts
        uses: actions/upload-artifact@master
        with:
          name: LightBulb
          path: LightBulb/bin/Publish/
```

~/.gitl

Trigger on new commits and pull requests

Clone repository and checkout HEAD (commit hash is passed as env var)

Install .NET Core v3.1.100

Run

Upload specified directory as a ZIP artifact

Introduction au CI/CD (9/11): Github actions (7/9) : Dashboard

Workflows		New workflow	All workflows			
All workflows			Filter workflows			
Octotree	CD		Event	Status	Branch	Actor
	CI		✓ CI		master	
			on: push (416e253)		triggered by Tyrrrz	
			✓ CI		master	
			on: push (830dc1c)		triggered by Tyrrrz	
			✓ CI		master	
			on: push (83d8c3f)		triggered by Tyrrrz	
			✓ CI		master	
			on: push (89d70e9)		triggered by Tyrrrz	
			✗ CI		master	
			on: push (e53588d)		triggered by Tyrrrz	
			✗ CI		master	
			on: push (6877226)		triggered by Tyrrrz	
			✓ CI		master	
			on: push (695d29c)		triggered by Tyrrrz	
			✓ CI		master	
			on: push (2cb9e8c)		triggered by Tyrrrz	

Introduction au CI/CD (10/11): Github actions (8/9) : Actions

- By GitHub repository 
`{owner}/{repo}@{ref}`
`{owner}/{repo}/{path}@{ref}`
- By file path 
`./path/to/dir`
- By Docker image 
`docker://{image}:{tag}`

```
jessfraz/branch-cleanup-action@master  
johndoe/my-actions/push-image@v1
```

```
./.github/actions/my-action
```

```
docker://hello-world:latest
```

Introduction au CI/CD (11/11): Github actions (9/9) : Secrets

```
steps:
```

```
# ...
```

```
- name: Collect coverage report
```

```
  run: |
```

```
    choco install codecov --no-progress
```

```
    codecov -f LtGt.Tests/bin/Release/Coverage.xml -t ${{secrets.CODECOV_TOKEN}}
```



Secret variable

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

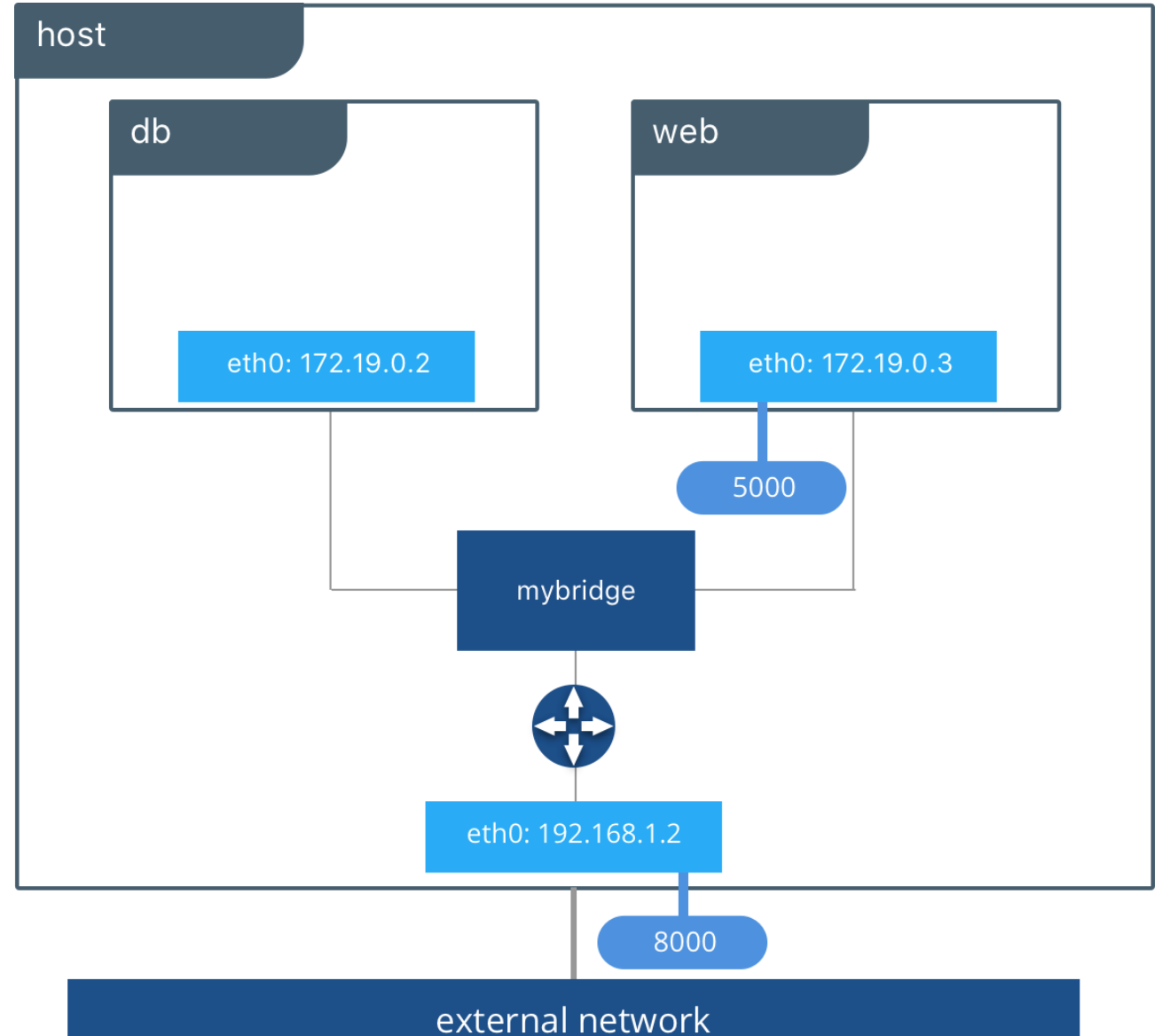
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Gestion des réseaux (1/2): Objectifs



Gestion des réseaux (2/2): Types de réseau

Compare Docker Network driver types

Driver/ Features	Bridge	User defined bridge	Host	Overlay	Macvlan/ipvlan
Connectivity	Same host	Same host	Same host	Multi-host	Multi-host
Service Discovery and DNS	Using "links". DNS using /etc/hosts	Done using DNS server in Docker engine	Done using DNS server in Docker engine	Done using DNS server in Docker engine	Done using DNS server in Docker engine
External connectivity	NAT	NAT	Use Host gateway	No external connectivity	Uses underlay gateway
Namespace	Separate	Separate	Same as host	Separate	Separate
Swarm mode ¹	No support yet	No support yet	No support yet	Supported	No support yet
Encapsulation	No double encap	No double encap	No double encap	Double encap using Vxlan	No double encap
Application	North, South external access	North, South external access	Need full networking control, isolation not needed	Container connectivity across hosts	Containers needing direct underlay networking

¹ – 17.06 introduces Swarm mode for all network drivers

TP-3: Gérez vos reseaux Docker

- Créez un reseau docker de type bridge
- Créez deux conteneurs ubuntu dans le reseau créé précédemment
- Installer la commande ping et tentez de pinguer les conteneurs entre eux avec leur ip et par leur nom

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

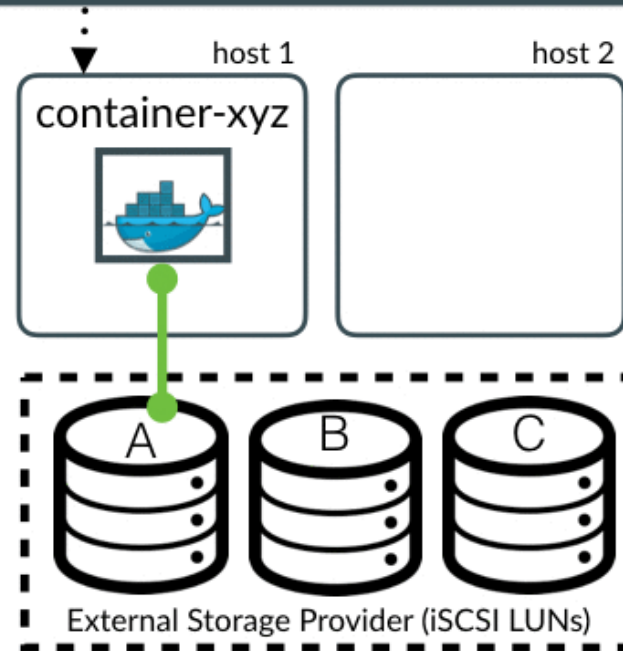
Introduction à l'orchestration avec Docker Swarm

Mini-projet

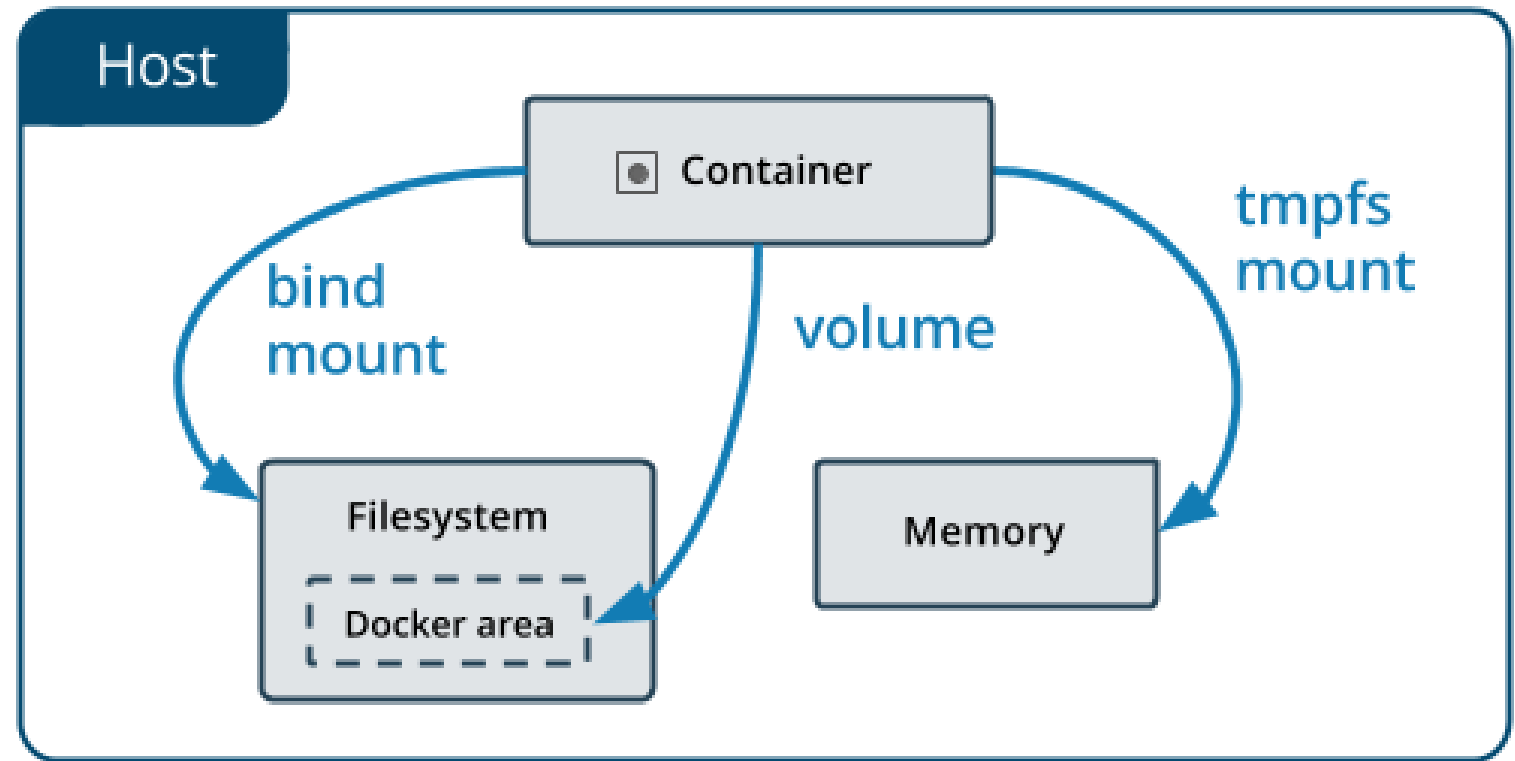
Gestion du stockage (1/2): Problème

Create and Run Container on Host 1

```
$ docker run -d -v /host/data:/data --volume-driver=flocker  
--name=container-xyz app
```



Gestion du stockage (2/2): Type de stockage



TP-4: Gérez votre stockage persistant

- Créez un volume de type volumes
- Créez deux conteneurs ubuntu (ubuntu1 et ubuntu2) et Montez le volume créé dans le repertoire /tmp de chacun des conteneurs ubuntu
- Créez un fichier toto.txt dans repertoire /tmp de ubuntu1 et verifiez qu'il est bien present dans /tmp de ubuntu2
- Créez un conteneur apache dont le site internet affiché sera celui hébergé ici : <https://github.com/diranetafen/static-website-example.git> (utilisez un volume de type bind mount), n'oubliez pas de verifier que le site est bien accessible

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

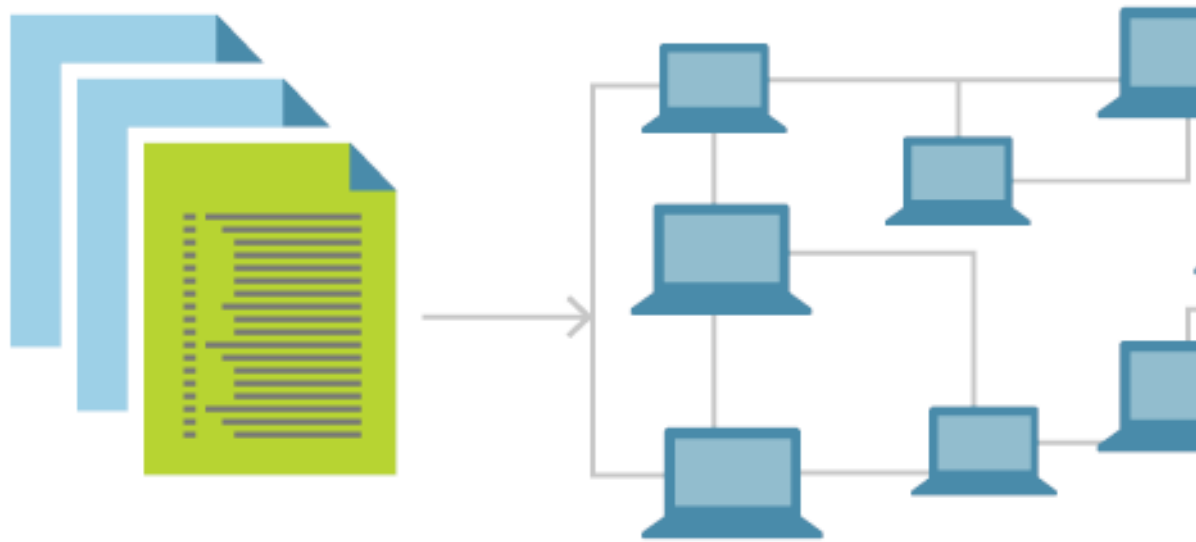
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Introduction à l'infrastructure as code (IaC) avec docker-compose(1/2): Définition



- Automatisation
- Redéploiement
- Versionning

Introduction à l'infrastructure as code (IaC) avec docker- compose(2/2): Anatomie

```
version: '2'
services:
  web:
    image: odoo:12.0
    depends_on:
      - db
    ports:
      - "8069:8069"
    volumes:
      - odoo-web-data:/var/lib/odoo
      - ./config:/etc/odoo
      - ./addons:/mnt/extra-addons
  db:
    image: postgres:10
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_PASSWORD=odoo
      - POSTGRES_USER=odoo
      - PGDATA=/var/lib/postgresql/data/pgdata
    volumes:
      - odoo-db-data:/var/lib/postgresql/data/pgdata
volumes:
  odoo-web-data:
  odoo-db-data:
```

TP-5: IaC avec docker-compose

- Déployer le logiciel odoo à l'aide de docker-compose
- Vous pouvez vous inspirer de la documentation disponible sur le dockerhub : https://hub.docker.com/_/odoo
- Modifiez le docker-compose afin d'exposer le port 80 du conteneur odoo à l'extérieur et pas le 8069
- Modifiez le docker-compose afin que le conteneur odoo et db soit dans le même réseau spécifique qui s'appellera odoo_network et de type bridge
- Vérifiez que l'application Odoo est bien accessible

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

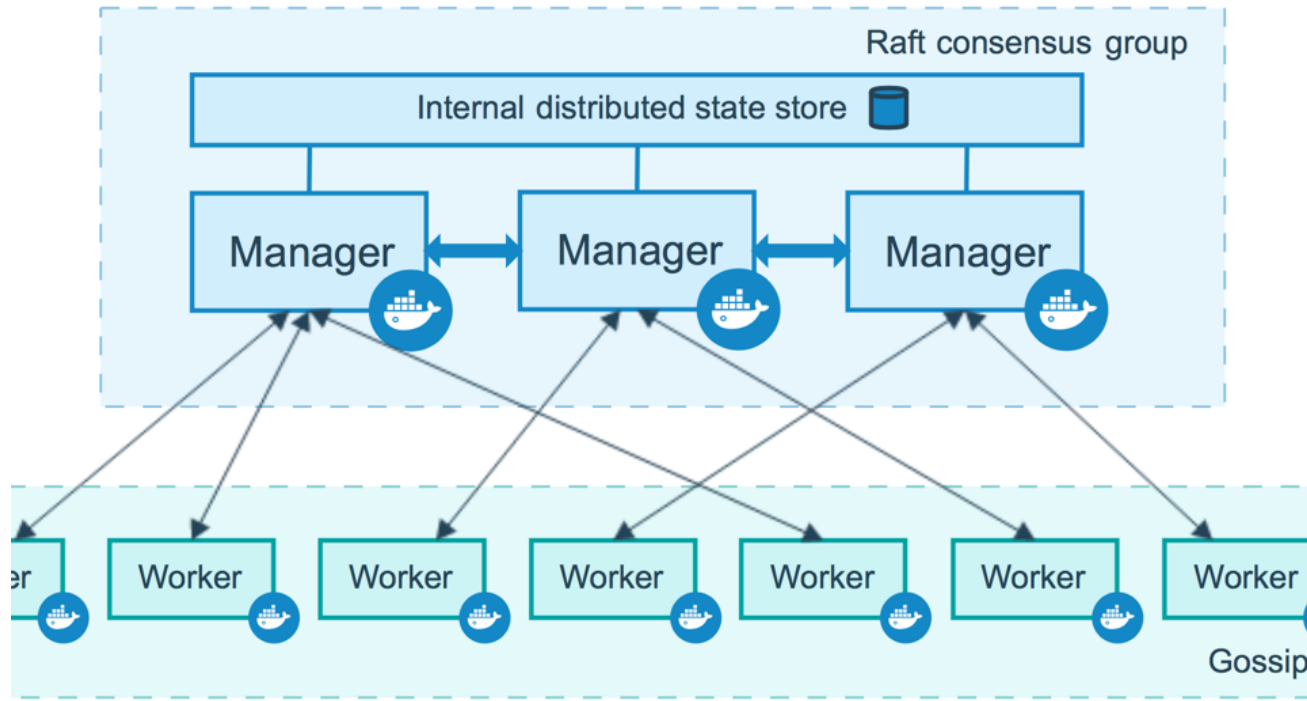
Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Introduction à l'orchestration avec Docker Swarm (1/2): Clustering



- Fail-Over
- Scalabilité
- Loadbalancing

Introduction à l'orchestration avec Docker Swarm (2/2): IaC

```
version: '3.4'
services:
  vote:
    image: dockersamples/examplevotingapp_vot
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
        order: stop-first
```

- Replica
- Update strategy

TP-6: Docker Swarm (1 master et 1 node)

- Installer un cluster swarm compose d'un master et d'un node
- Utilisez la plateforme de labs pour deployer deux instances (1 master et 1 worker)
- Initialisez le cluster et rajouter le worker dans le cluster
- Déployez l'application de gestion de vote suivante:
<https://github.com/dockersamples/example-voting-app.git>
- Testez que l'application fonctionne

Plan

Présentation du formateur

Présentation des Concepts

Introduction à la conteneurisation

Installation de Docker

Gestion des images

Introduction au CI/CD

Gestion des reseaux

Gestion du stockage

Introduction à l'infrastructure as code (IaC) avec docker-compose

Introduction à l'orchestration avec Docker Swarm

Mini-projet

Mini-projet: Student-list

<https://github.com/diranetafen/student-list>

