DEVOPS I : TOUT SAVOIR DE DOCKER EN QUELQUES HEURES

Configurer et utiliser les réseaux et le DNS de Docker

JORDAN ASSOULINE

OBJECTIFS DE LA SECTION

Objectifs

- Extrait du cours « Tout savoir des réseaux informatiques en quelques heures » pour comprendre le fonctionnement des protocoles DNS, NAT et DHCP
- Fonctionnement réseau de Docker
- Récupérer les informations réseaux de nos conteneurs
- Utiliser les commandes réseaux de Docker
- Comment fonctionne le DNS sur Docker ?
- Exercice de création de réseaux et de raccordement des conteneurs au DNS

DNS

JORDAN ASSOULINE

Pourquoi le DNS ?

- Nommer une machine sur le réseau en effectuant une correspondance entre le nom choisi et le numéro IP (résolution de nom directe)
  - Google.com est plus facile à se rappeler que 8.8.8.8
- Trouver le nom d ’une machine à partir de son numéro IP (résolution de nom inverse)
  - Rendu nécessaire par certaines applications.

Exemples :

- Adresse IP : 192.134.4.11 correspond à www.afnic.asso.fr

- Une information dans le DNS indique vers quelle machine diriger le courrier électronique :
  - Jordan.ASSOULINE@afnic.asso.fr

  ⇒relay1.afnic.asso.fr

  ⇒adresse ip 192.134.4.17

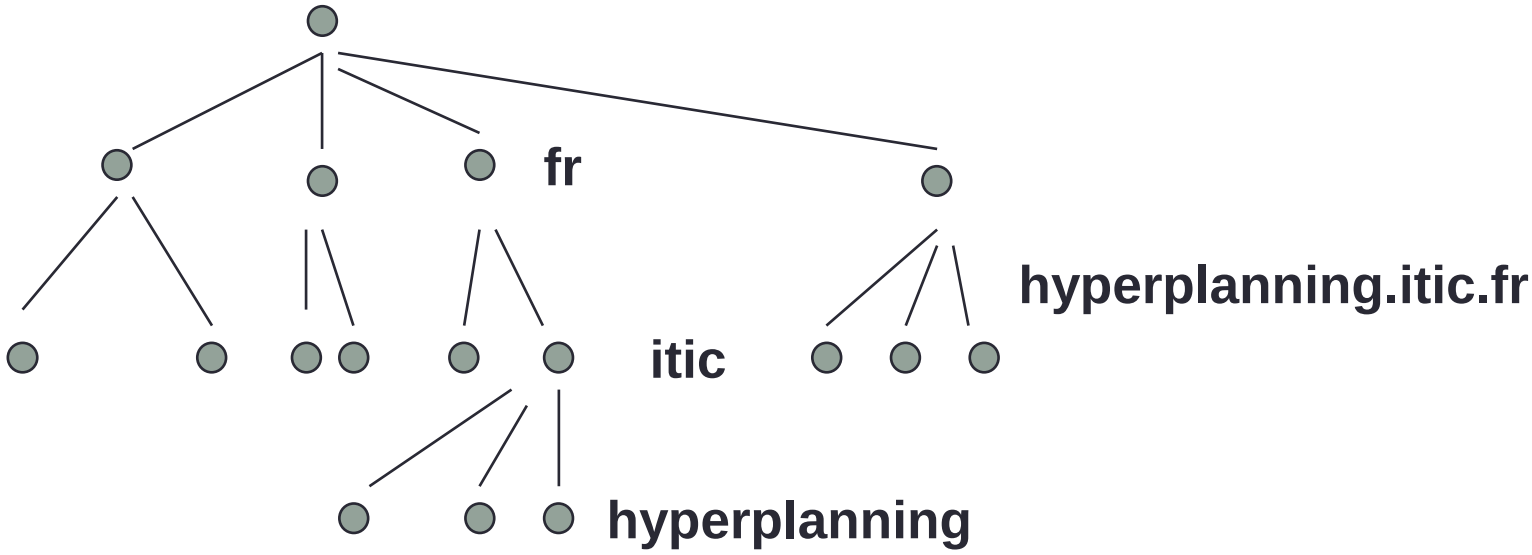
Avant le DNS ?

- Jusqu'en 1984 : fichier hosts.txt = /etc/hosts
  - inadapté à grande échelle
  - temps de diffusion des infos (par ftp !)
  - système centralisé
  - quelques centaines de machines dans les années 70 contre plusieurs millions aujourd'hui

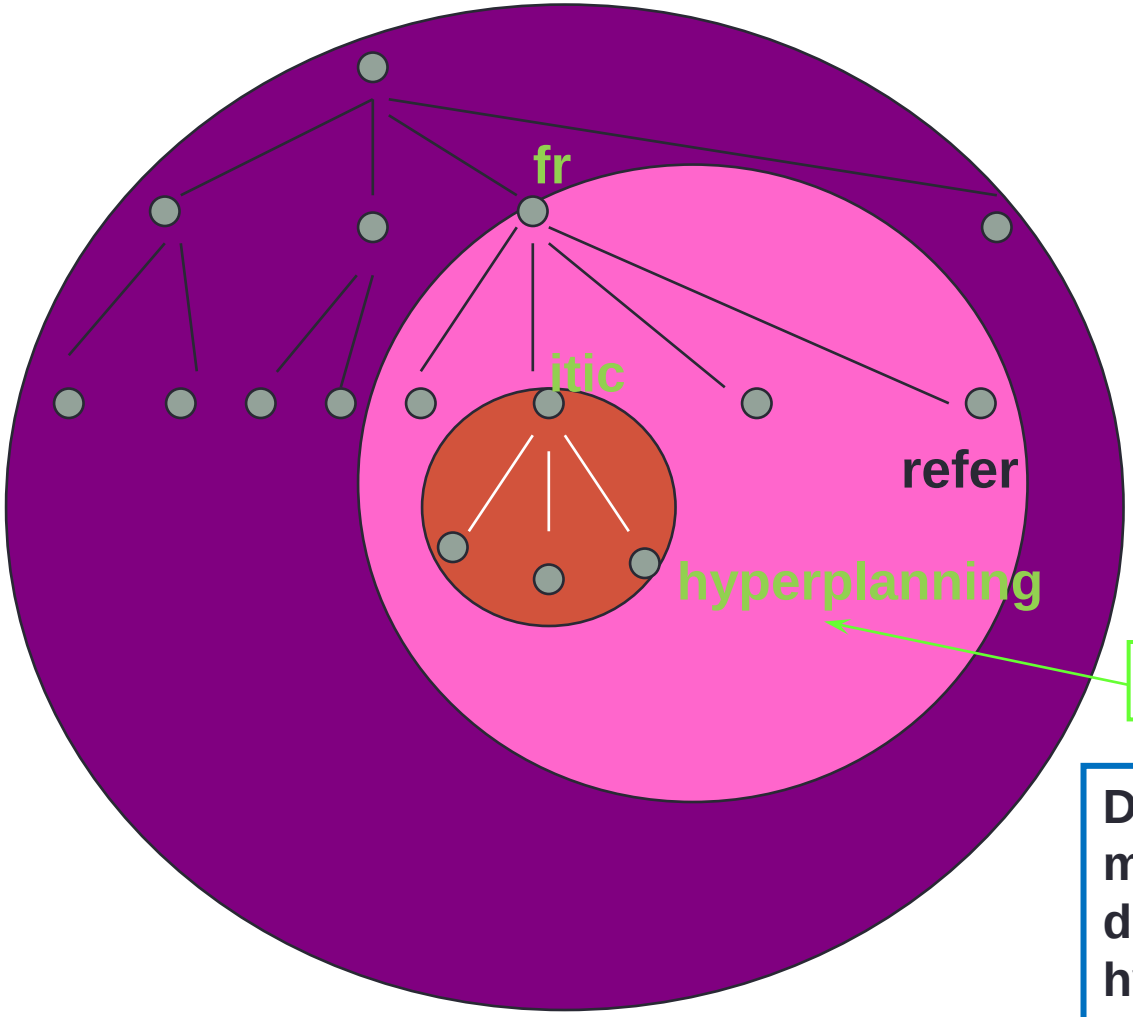
- La correspondance se faisait de manière statique
  - www.jordanassouline.com 12.2.13.24

Qu’est-ce qu’un nom de domaine ?

- Un nom de domaine est la séquence de labels depuis le noeud de l’arbre correspondant jusqu’à la racine.



Domaine



Domaine complet

Domaine fr

Domaine itic

noeud hyperplanning.itic.fr

Des noeuds peuvent avoir les mêmes noms dans des domaines différents :
hyperplanning.ensup.eu
hyperplanning.esgi.com

Les domaines racines

- Le premier niveau de l'espace DNS contient 7 domaines racines prédéfinis :
  - **com** : organisations commerciales (google.com)
  - **edu** : organisations concernant l'éducation (mit.edu)
  - **gov** : organisations gouvernementales (nsf.gov)
  - **mil** : organisations militaires (army.mil)
  - **net** : organisations réseau Internet (worldnet.net)
  - **org** : organisations non commerciales (eff.org)
  - **int** : organisations internationales (nato.int)
- Le domaine **arpa** est reservé à la résolution de nom inversée
- Les domaines des organisations nationales : cm, fr, uk, de, it, us, au, ca, se, etc.

Lecture des noms de domaine

- A l'inverse de l'adressage IP la partie la plus significative si situe à gauche de la syntaxe :

www.cm.refer.org

195.24.200.202

www. cm. refer.org

domaine org (.org)

domaine de l'organisation REFER

sous-domaine refer

machine www du domaine cm.refer.org

Les RR (Resource Records)

- **NS :** renseigne le nom des serveurs de noms pour le domaine.

- **A :** associe un nom d'hôte à une adresse ipv4 (32 bits)
- **AAAA** : associe un nom d'hôte à une adresse ipv6 (128 bits)

- **CNAME** : identifie le nom canonique d'un alias (un nom pointant sur un autre nom)

- **PTR** : c'est simplement la résolution inverse (le contraire du type A).

- **IN** : détermine l'association a la classe Internet. D'autres classes sont disponibles (CH et HS). Voir le RFC 1035.

DHCP

Pourquoi le DHCP ?

- Attribution de manière automatique d’adresses IPv4
- Permet de s’assurer que chaque équipement possède une adresse IPv4 unique sur le réseau
- Obtient un « bail » au moment de la demande

Etape 1 : Demander une adresse IP

Bonjour, il y a-t-il un serveur DHCP disponible pour me donner une adresse IP ?

Ordinateur1

Box internet

Etape 2 : Réponse du serveur

Oui, je suis un serveur DHCP, voici une adresse IP que je peux vous proposer: 192.168.1.3/24

Ordinateur1

Box internet

Liste des adresses IP

192.168.1.2/24	Ordinateur2
192.168.1.3/24	Libre
192.168.1.4/24	Ordinateur4
192.168.1.5/24	Libre
….	

Etape 3 : Confirmation du client

Super, elle me convient, je vais prendre cette adresse !

Ordinateur1

Box internet

Liste des adresses IP

192.168.1.2/24	Ordinateur2
192.168.1.3/24	**En Cours**
192.168.1.4/24	Ordinateur4
192.168.1.5/24	Libre
….	

Etape 4 : Accusé réception

Très bien, c'est validé.

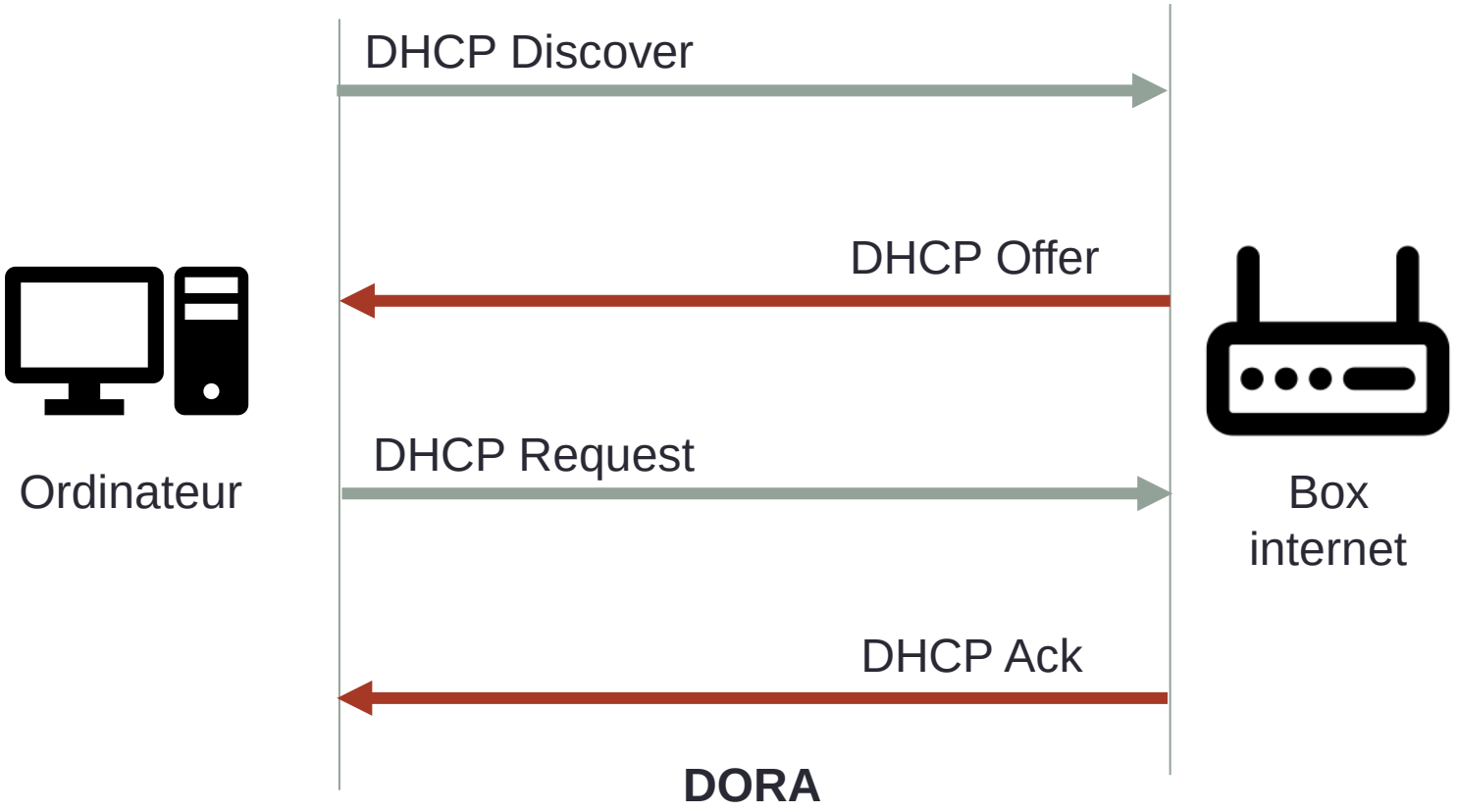
Ordinateur1

Box internet

Liste des adresses IP

192.168.1.2/24	Ordinateur2
192.168.1.3/24	**Ordinateur1**
192.168.1.4/24	Ordinateur4
192.168.1.5/24	Libre
….	

Du point de vue informatique



QU'EST-CE QUE LE NAT

JORDAN ASSOULINE

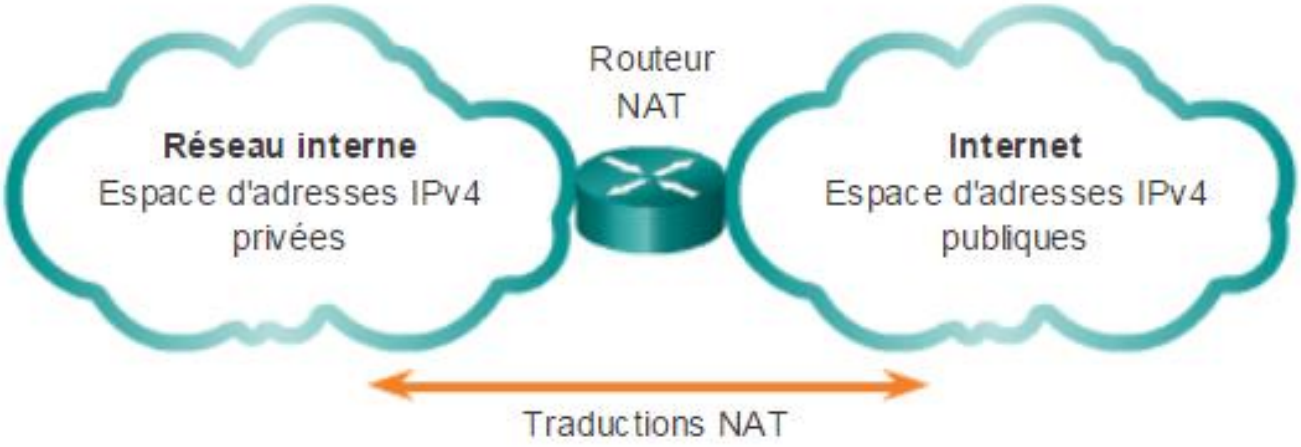
Adresses IPv4 Privées

- Il n'existe pas suffisamment d'adresses IPv4 publiques pour pouvoir attribuer une adresse unique à chaque périphérique connecté à Internet.
- Les adresses privées sont utilisées au sein d'une entreprise ou d'un site pour permettre aux périphériques de communiquer localement, tel que défini dans le document RFC 1918.
- Comme ces adresses n'identifient aucune entreprise ou organisation unique, les adresses IPv4 privées ne peuvent pas être acheminées sur Internet.
- Pour permettre à un périphérique possédant une adresse IPv4 privée d'accéder aux périphériques et aux ressources situés en dehors du réseau local, l'adresse privée doit d'abord être traduite en adresse publique.
- La traduction d'adresse réseau (NAT) assure la traduction des adresses privées en adresses publiques.

Les différentes classes d'adresses IPv4 privées

Les adresses Internet privées sont définies dans la RFC 1918 :

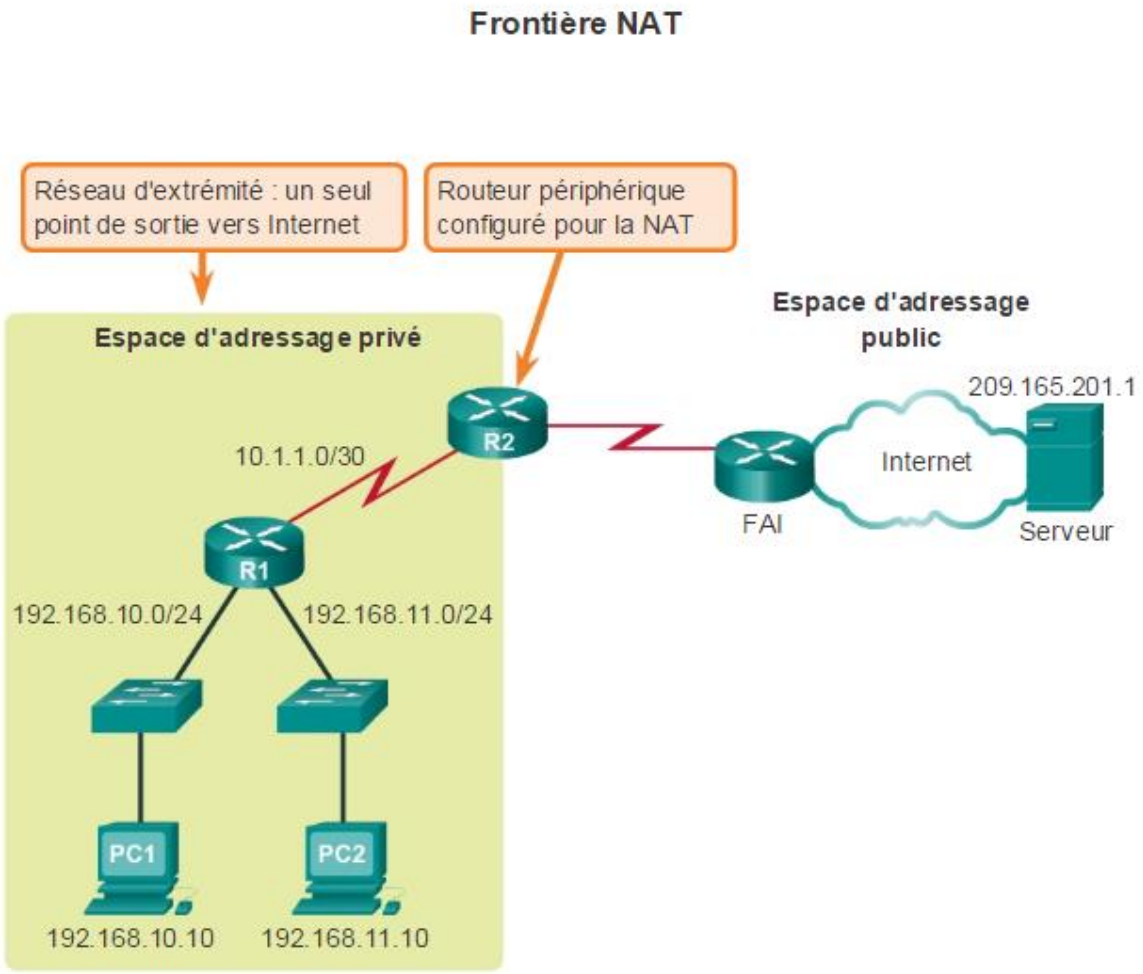
Classe	Plage d'adresses internes RFC 1918	Préfixe CIDR
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16



La fonction du NAT

- Le NAT est un processus utilisé pour convertir les adresses réseaux.
- Son utilisation principale consiste à limiter la consommation des adresses IPv4 publiques.
- NAT fonctionne généralement à la périphérie d'un réseau d'extrémité, soit via un firewall ou routeur.
- Lorsque le trafic doit être envoyé ou reçu ou d'autres organisations ou de l'Internet, le routeur de périphérie traduit les adresses à une adresse publique unique au monde.

La fonction du NAT



La fonction du NAT

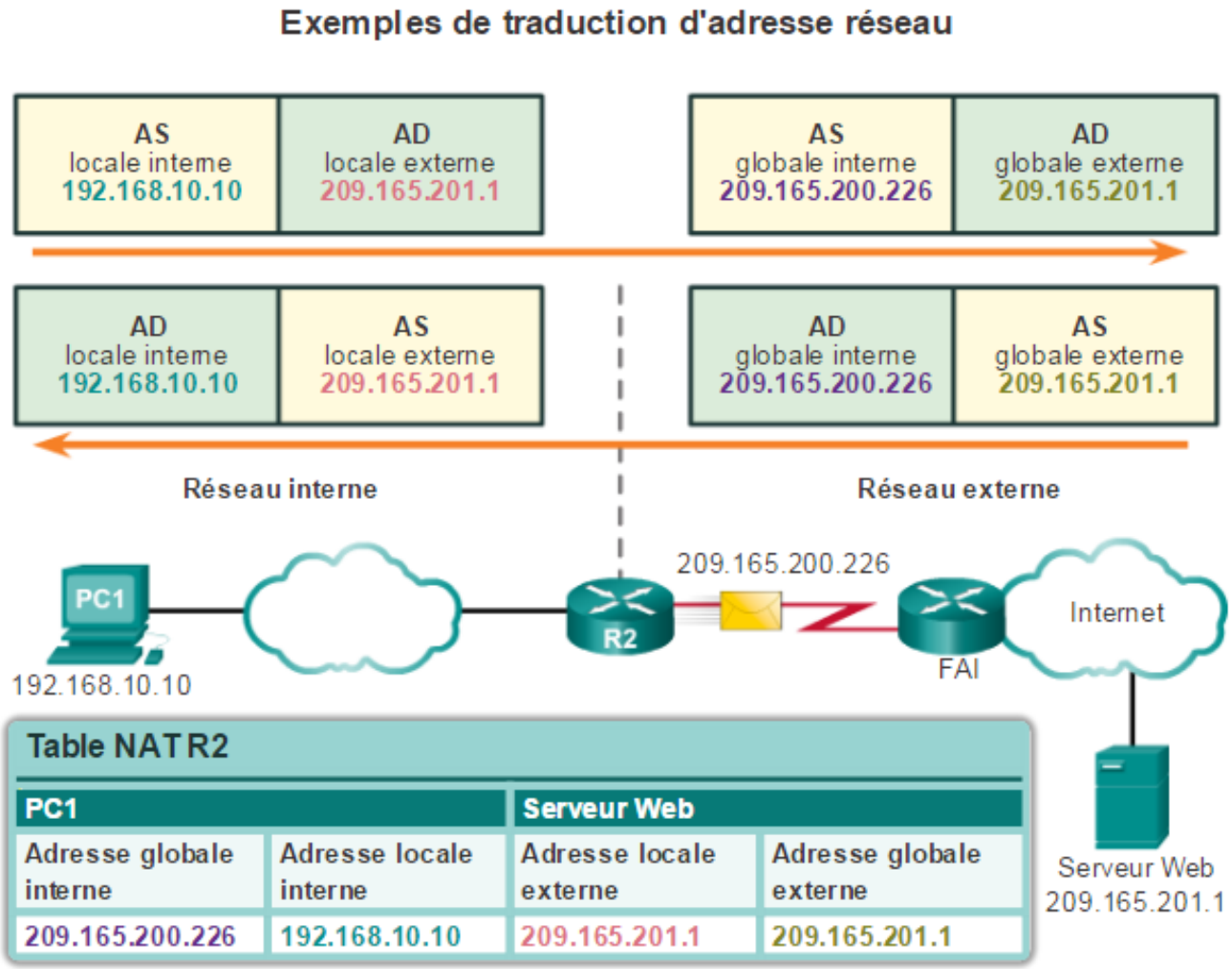


Table NAT R2

PC1		Serveur Web	
Adresse globale interne	Adresse locale interne	Adresse locale externe	Adresse globale externe
209.165.200.226	192.168.10.10	209.165.201.1	209.165.201.1

LE FONCTIONNEMENT RÉSEAU DE DOCKER

JORDAN ASSOULINE

Les réseaux par défaut sur Docker

- Chaque conteneur est connecté à un réseau privé virtuel qui s'appelle « **bridge** »

- Chaque réseau de Docker peut communiquer avec l'adresse IP de la machine physique grâce à un firewall virtuel effectuant du NAT

- Tous les conteneurs situés sur le même réseau virtuel peuvent discuter entre eux (sans utiliser l'option –p)

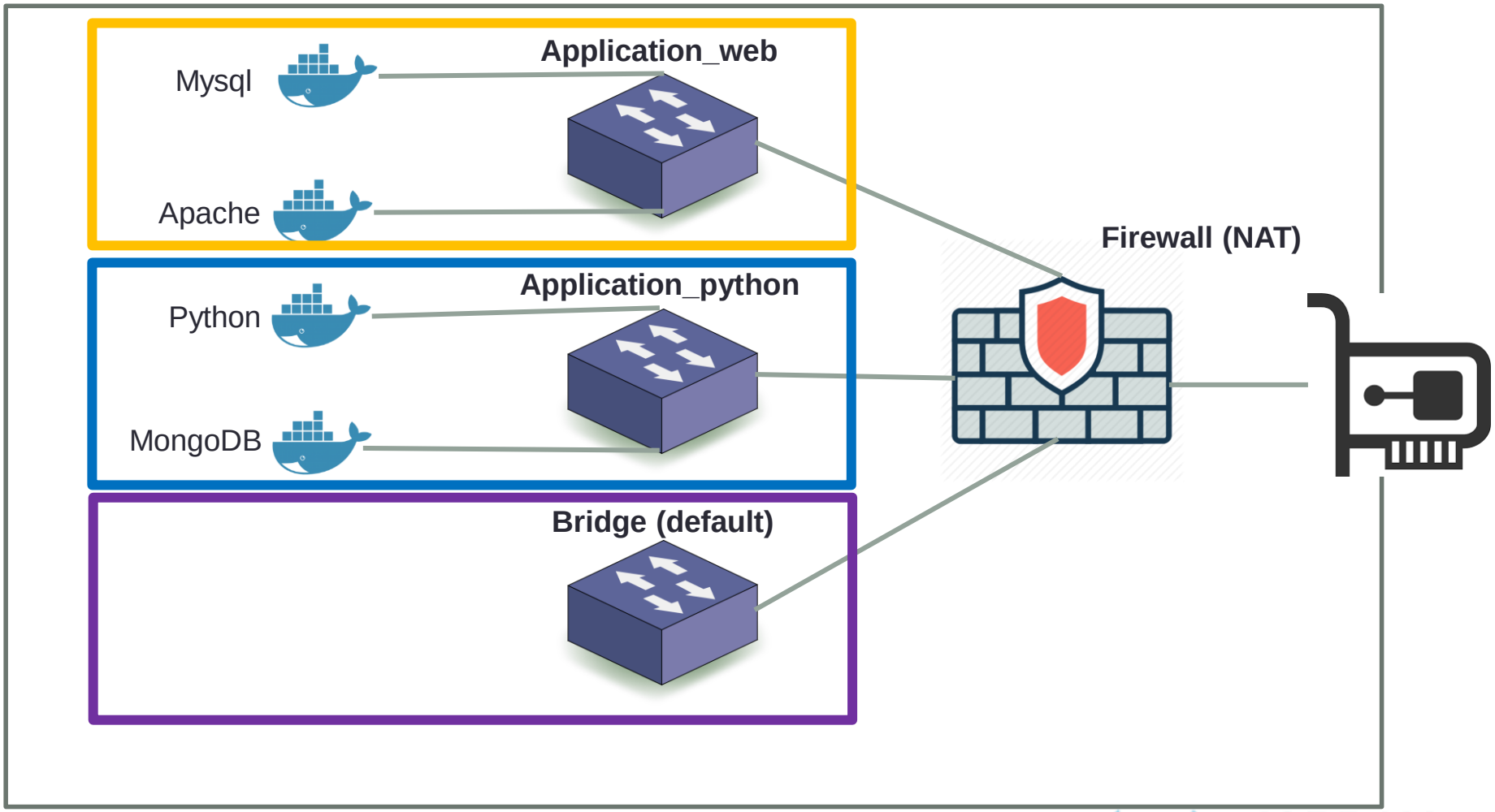
Les règles de bonnes pratiques

- Il est commun de dire qu'il faut créer un nouveau réseau virtuel pour chaque application ou service.

- Par exemple on pourra créer un réseau virtuel appelé « **application_web** » dans lequel se trouvera un conteneur mysql et un autre conteneur apache.

- Si sur le même serveur physique, on a également une application python utilisant mongodb, on créera un autre réseau virtuel appelé « **application_python** » dans lequel se trouvera nos conteneurs python et mongodb

Les règles de bonnes pratiques



JORDAN ASSOULINE

Ce que l'on peut faire sur les conteneurs

- On peut facilement créer de nouveaux réseaux virtuels (un pour chaque application)
- Il est possible d'attacher un conteneur à plusieurs réseaux virtuels en même temps (voir à aucun)
- On peut connecter directement le conteneur à la carte réseau de la machine physique et utiliser son IP

RÉCUPÉRER LES INFORMATIONS RÉSEAUX D'UN CONTENEUR

JORDAN ASSOULINE

La commande « docker container port »

- **`docker container port`**
  - Liste les associations entre les ports des conteneurs et les ports de la machine physique

- https://docs.docker.com/engine/reference/commandline/container_port/

La commande « docker container inspect »

- **`docker container inspect --format '{{.NetworkSettings.IPAddress}}' <NAME>`**
  - Permet de n'affiche que la ligne correspondant à l'adresse IP du conteneur parmi toutes les informations de la commande inspect

- https://docs.docker.com/engine/reference/commandline/container_inspect/
- https://docs.docker.com/v17.09/engine/admin/formatting/#template-functions

JORDAN ASSOULINE

LES COMMANDES RÉSEAUX DE DOCKER

JORDAN ASSOULINE

La commande « docker network ls »

- **`docker network ls`**
  - Affiche les réseaux Docker qui existent, normalement, au démarrage vous devez en avoir trois.
  - Le **bridge** (parfois appelé Docker0) correspond au réseau virtuel par défaut qui est NATé derrière l'adresse IP de l'hôte. Attention il n'est pas lié au serveur DNS de docker par défaut, et il faut donc le faire manuellement.
  - Le **host** correspond au réseau qui attache les conteneurs directement à l'hôte (sans NAT) ce qui est franchement délétère au niveau de la sécurité, mais augmente les performances.
  - Le **none** qui enlève l'interface eth0 du conteneur et ne laisse que l'interface localhost d'existante.
- https://docs.docker.com/engine/reference/commandline/network_ls/

JORDAN ASSOULINE

La commande « docker network inspect »

- **`docker network inspect <NETWORK_NAME>`**
  - Affiche les détails sur le réseau Docker indiqué (accessibles via la commande « docker network ls ») et montre les conteneurs qui lui sont attachés
- https://docs.docker.com/engine/reference/commandline/network_inspect/

La commande « docker network create »

- **`docker network create <NETWORK_NAME>`**
  - Permet de créer un nouveau réseau auquel pourront s'attacher les containers.
  - Le Driver par défaut auquel sera attaché le nouveau réseau est le **bridge** qui permet simplement de pouvoir attacher un sous-réseau au nouveau réseau créé
  - Par défaut, les nouveaux réseaux sont automatiquement rattachés au démon DNS

- https://docs.docker.com/engine/reference/commandline/network_create/

La commande « docker container run »

- **`docker container run --network <NETWORK_NAME> --name webserver nginx`**
  - L'option network permet d'attacher le conteneur à un réseau spécifique

- https://docs.docker.com/engine/reference/commandline/run/

La commande «docker network connect»

- **`docker network connect <NETWORK_NAME> <CONTAINER_NAME>`**
  - Permet d'attacher un container à un nouveau réseau (mais n'enlève pas l'attachement au réseau précédent)

- https://docs.docker.com/engine/reference/commandline/network_connect/

La commande «docker network disconnect»

- **`docker network disconnect <NETWORK_NAME> <CONTAINER_NAME>`**
  - Permet détacher dynamiquement la NIC d'un conteneur.

- https://docs.docker.com/engine/reference/commandline/network_disconnect/