DOCKERIZING JMETER FOR FUZZING, BRUTE-FORCING AND DDOS ATTACKS

PAR MAXIME LEBLANC

PLAN

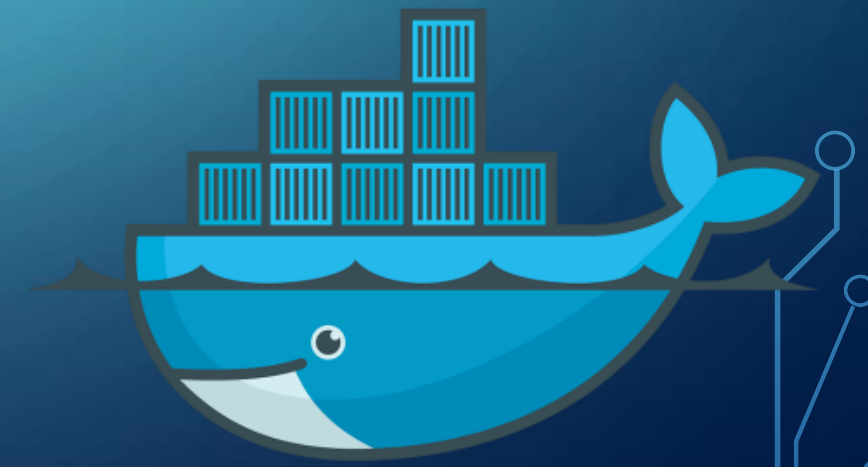
- Docker
  - Conteneurs
  - Networking
  - Filesystem
- JMeter
- Attack scenarios
- Demo (?)

DOCKER

- On peut voir Docker comme une "single-process VM"
- Il ne s'agit en fait pas de virtualisation à proprement parler
- L'isolation se fait via des mécanismes natifs Linux
- Une VM Linux est donc nécessaire pour utiliser Docker sous Windows et MacOS

DOCKER - CONTENEURS

- Les conteneurs sont les environnements isolés où s'exécutent les processus
- Assure un environnement constant pour les applications
  - Paradigme semblable à celui d'une VM
- Très utile pour la "scalability" horizontale



DOCKER - IMAGES

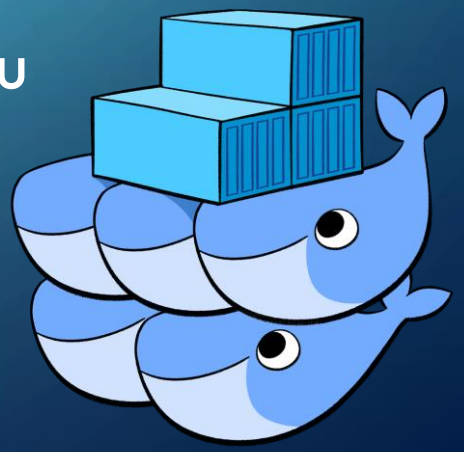
- Les images sont les "bases communes" des conteneurs
- Assimilables à des distributions Linux
- Définies dans un Dockerfile
- Un même Dockerfile donnera le même environnement sur tous les ordinateurs hôtes
- On en trouve une grande quantité sur Internet

DOCKER – SYSTEME DE FICHIERS

- Le système de fichiers Docker est un peu particulier
  - Chaque conteneur a sa propre "vision" de l'image qu'il partage avec les autres
  - Les fichiers sont enregistrés de manière différentielle
  - Les fichiers ajoutés ou modifiés lors de l'exécution d'un conteneur seront revenus à leur état original lors de la prochaine exécution

DOCKER - NETWORKING

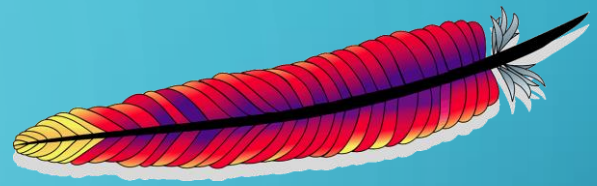
- Les images partagent entre-elles un meme "subnet" et doivent passer par le NAT Docker afin de communiquer avec le monde extérieur
  - Possible de créer des conteneurs complètement isolés
  - Possible de faire en sorte que les conteneurs se parlent entre-eux sans liens avec le monde extérieur
- Chaque conteneur possède sa propre "stack" réseau



EXEMPLE DE DOCKERFILE (NODEJS)

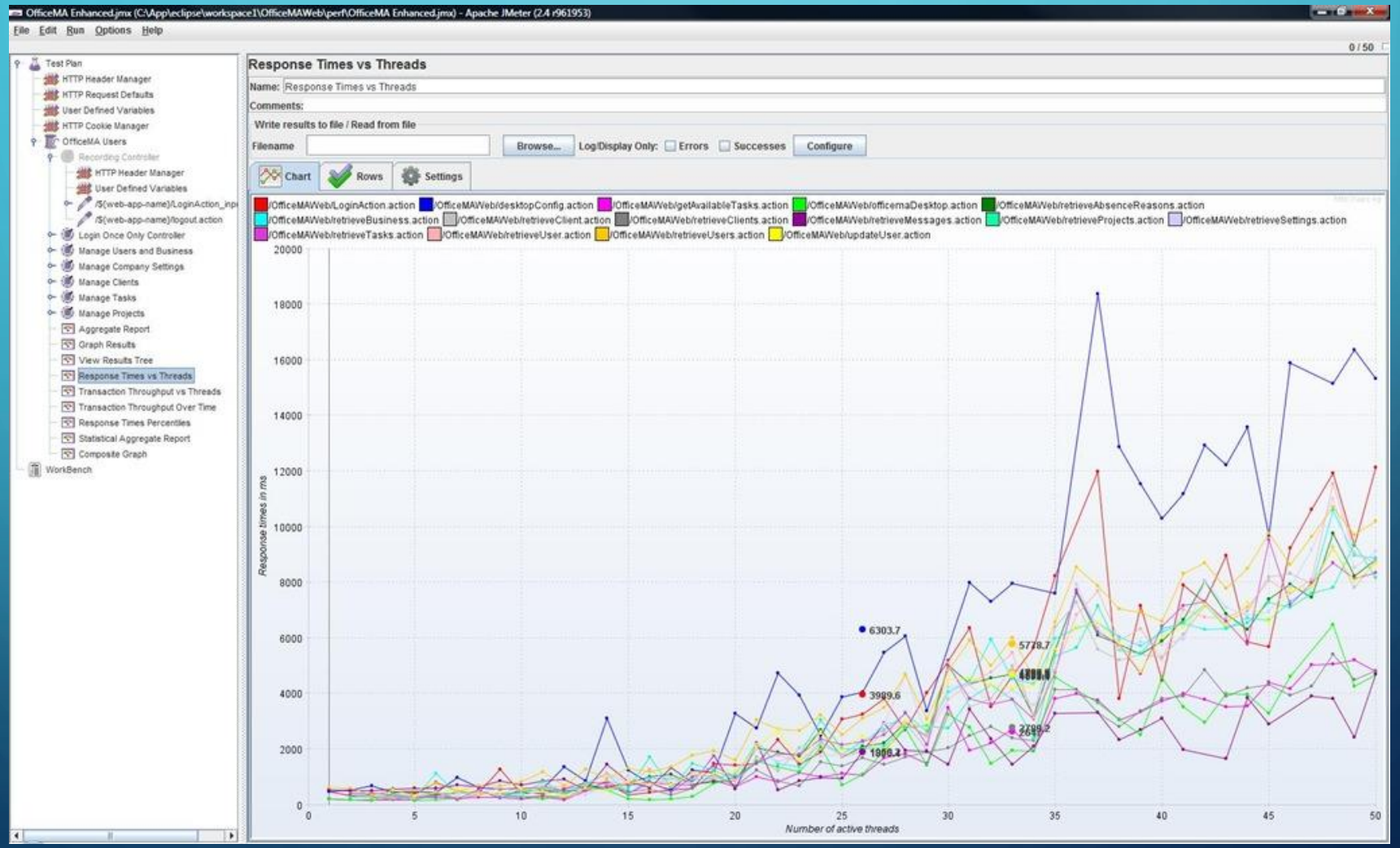
```
FROM ubuntu
MAINTAINER Kimbro Staken
RUN apt-get install -y software-properties-common python
RUN add-apt-repository ppa:chris-lea/node.js
RUN echo "deb http://us.archive.ubuntu.com/ubuntu/ precise universe" >> /etc/apt/sources.list
RUN apt-get update
RUN apt-get install -y nodejs
#RUN apt-get install -y nodejs=0.6.12~dfsg1-1ubuntu1
RUN mkdir /var/www
ADD app.js /var/www/app.js
CMD ["/usr/bin/node", "/var/www/app.js"]
```

JMETER



- JMeter est un outil de *bechmarking* pour les applications Web
- Permet de voir comment réagit une application en présence de multiples utilisateurs
- Plusieurs outils statistiques par rapport aux temps de réponses, au pourcentage d'erreurs, etc...
- Hautement *customisable*
  - CSRF, Cache control, File uploads, Randomness...

JMETER


OfficeMA Enhanced.jmx (C:\App\eclipse\workspace1\OfficeMAWeb\perf\OfficeMA Enhanced.jmx) - Apache JMeter (2.4 r961953)
File Edit Run Options Help
Test Plan
HTTP Header Manager
HTTP Request Defaults
User Defined Variables
HTTP Cookie Manager
OfficeMA Users
Recording Controller
HTTP Header Manager
User Defined Variables
/${web-app-name}/LoginAction_inp
/${web-app-name}/logout.action
Login Once Only Controller
Manage Users and Business
Manage Company Settings
Manage Clients
Manage Tasks
Manage Projects
Aggregate Report
Graph Results
View Results Tree
Response Times vs Threads
Transaction Throughput vs Threads
Transaction Throughput Over Time
Response Times Percentiles
Statistical Aggregate Report
Composite Graph
WorkBench
Response Times vs Threads
Name: Response Times vs Threads
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Chart
Rows
Settings
/OfficeMAWeb/LoginAction.action
/OfficeMAWeb/desktopConfig.action
/OfficeMAWeb/getAvailableTasks.action
/OfficeMAWeb/officemaDesktop.action
/OfficeMAWeb/retrieveAbsenceReasons.action
/OfficeMAWeb/retrieveBusiness.action
/OfficeMAWeb/retrieveClient.action
/OfficeMAWeb/retrieveClients.action
/OfficeMAWeb/retrieveMessages.action
/OfficeMAWeb/retrieveProjects.action
/OfficeMAWeb/retrieveSettings.action
/OfficeMAWeb/retrieveTasks.action
/OfficeMAWeb/retrieveUser.action
/OfficeMAWeb/retrieveUsers.action
/OfficeMAWeb/updateUser.action
Response times in ms
Number of active threads
6303.7
5778.7
3999.6
1966.2

SCÉNARIOS D'ATTAQUES

- Cette architecture peut nous être utile pour des tests nécessitant beaucoup de requêtes:
  - Fuzzing
    - Random generator
  - Brute-forcing
    - Iterator / Java code
  - DDOS
    - Login with CSRF/Session cookies -> Database workload

ARCHITECTURE DE L'ATTAQUE

AVANTAGES DE LA TECHNIQUE

- Permet des miliers de "vrais" clients
  - Leur propre ID
  - Leur propre session
  - Peuvent effectuer des actions légitimes
  - Émule un "vrai" navigateur moderne
    - =/= spyders normaux

DEMO

- Hydra.py
  - Orcherstre le tout
- Prepare.py
  - Prépare les fichiers de configuration
- Spawn.py
  - Génère les conteneurs

DEMO