

LE CACHING WEB AVEC VARNISH

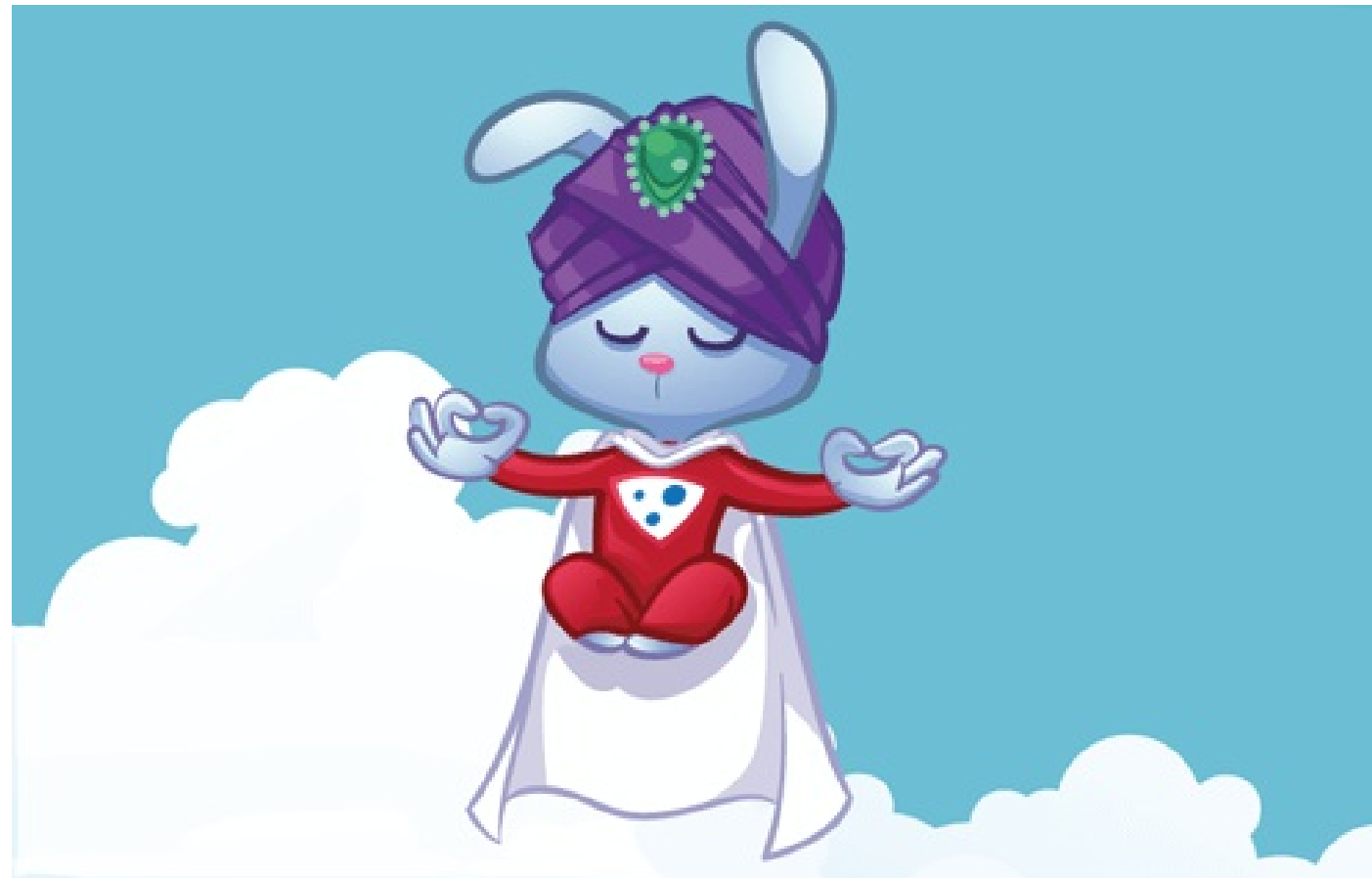
Franchini Fabien, Henneberger Sébastien et Rouiller Bastien
27 Janvier 2017

...Dépôt de la présentation

Introduction

Au programme:

- En bref
- Caching
- Fonctionnement
- Démonstration
- Q&A



En bref

Qu'est-ce donc ?

Varnish est un reverse proxy HTTP principalement utilisé comme accélérateur web. Varnish conserve (cache) des fichiers (ou des fragments) en mémoire pour réduire le temps de réponse et la consommation de bande passante.

Varnish est riche en fonctionnalités et peut aussi être utilisé comme:

- Parefeu;
- Bouclier anti-DDoS;
- Répartiteur de charge;
- Contrôleur d'accès;
- Et tant d'autres manières..

Particularité: Il est conçu pour fonctionner sur du matériel et des OS modernes (64-bit uniquement).



En bref

Varnish se présente sous la forme d'une suite logicielle, déclinée en deux distributions.

Qu'est-ce donc ?

Varnish Cache

- Version Open Source (BSD);
- Suite logicielle en ligne de commande;
- Support de SSL/TLS (frontend uniquement).

Varnish Cache et Varnish Plus

Varnish Plus

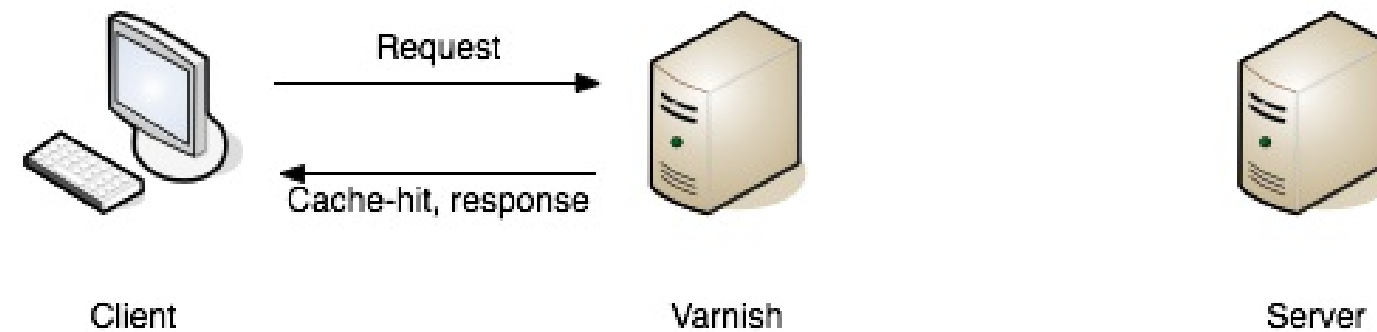
- Version commerciale de la suite;
 - Prix de base: 31'000\$;
 - Support client.
- Support de SSL/TLS (frontend et backend);
- Varnish Administration Console (VAC);
- Varnish High Availability (VHA);
- Varnish Custom Statistics (VCS).

Le caching

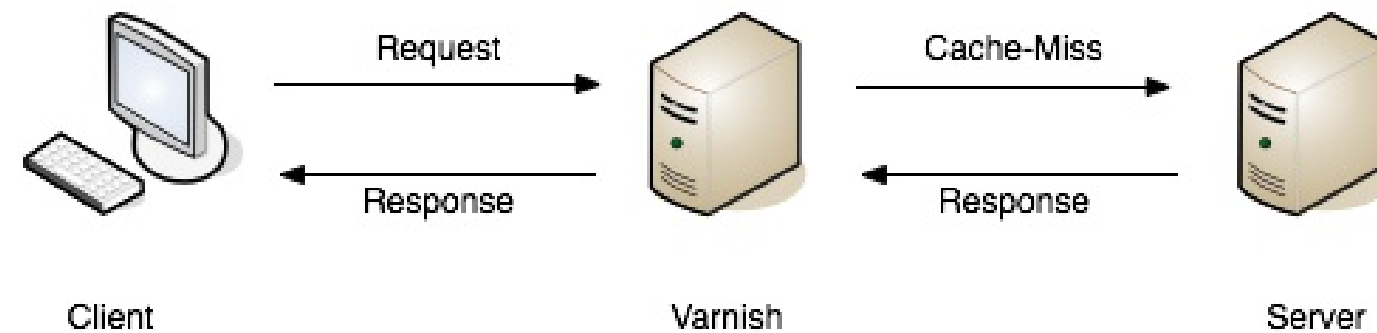
Lorsqu'un client accède à un noeud, alors deux cas de figure se présentent:

Fonctionnement

1. Varnish contient une copie de l'objet en cache. Il le renvoie directement au client (Cache-hit).



2. Varnish ne contient pas de copie de l'objet (ou ce dernier a changé). Il envoie une nouvelle requête au noeud et renvoie sa réponse, au client (Cache-miss).



Le caching

Le protocole HTTP prévoit des entêtes pour émettre des directives de contrôle du caching des objets:

Fonctionnement

Cache-Control:

- no-cache: Le noeud doit valider la réutilisation de l'objet;
- no-store: L'objet ne doit pas être sauvé dans le cache;
- max-age: Durée de vie maximale de l'objet en cache.

Vary: se base sur un entête http pour déterminer si une réponse en cache peut être réutilisée

- Accept-encoding: uniquement une même représentation de l'objet est demandée (html, json)

Last-Modified: dernière date de modification, envoyée par le serveur

If-Modified-Since: timestamp de la dernière page obtenue, envoyée par le client

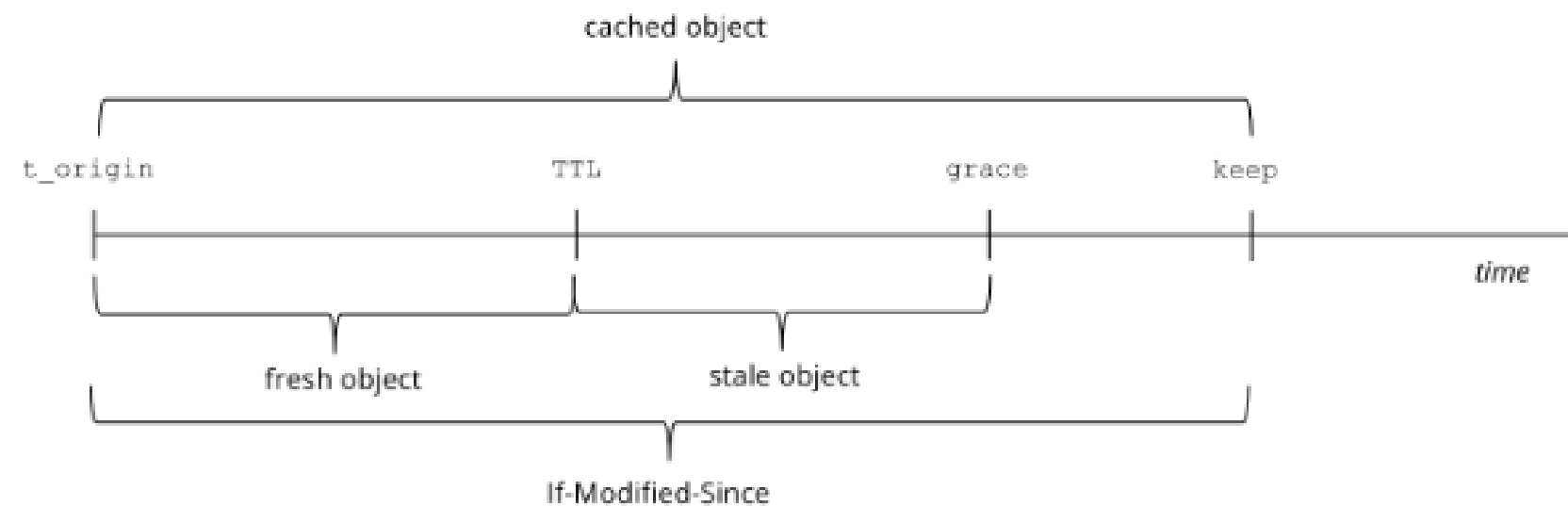
Age: Durée de vie de l'objet en cache.

Le caching

Varnish se base principalement sur les entêtes HTTP des réponses, pour déterminer le comportement de leur mise en cache.

Fonctionnement

Varnish



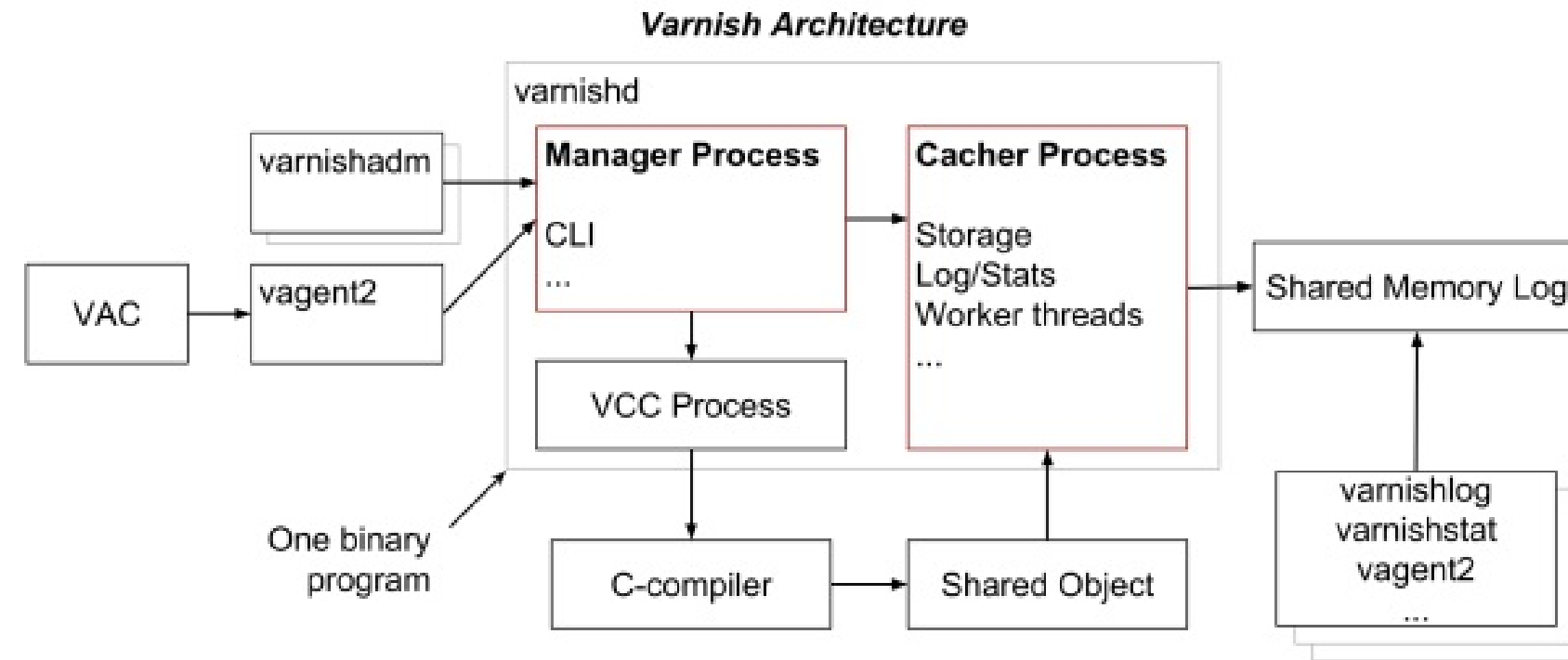
Voici les différents temps de vie de l'objet en cache :

1. fresh object: l'objet est "frais" et donc peut être renvoyé au client.
2. stale object: l'objet est encore utilisable si le serveur web n'est pas atteignable ou surchargé.
3. If-Modified-Since: l'objet est renvoyé si aucune modification n'a été faite depuis If-Modified-Since.

Une fois ces délais dépassés, l'objet est supprimé du cache.

En profondeur

Architecture



Manager:

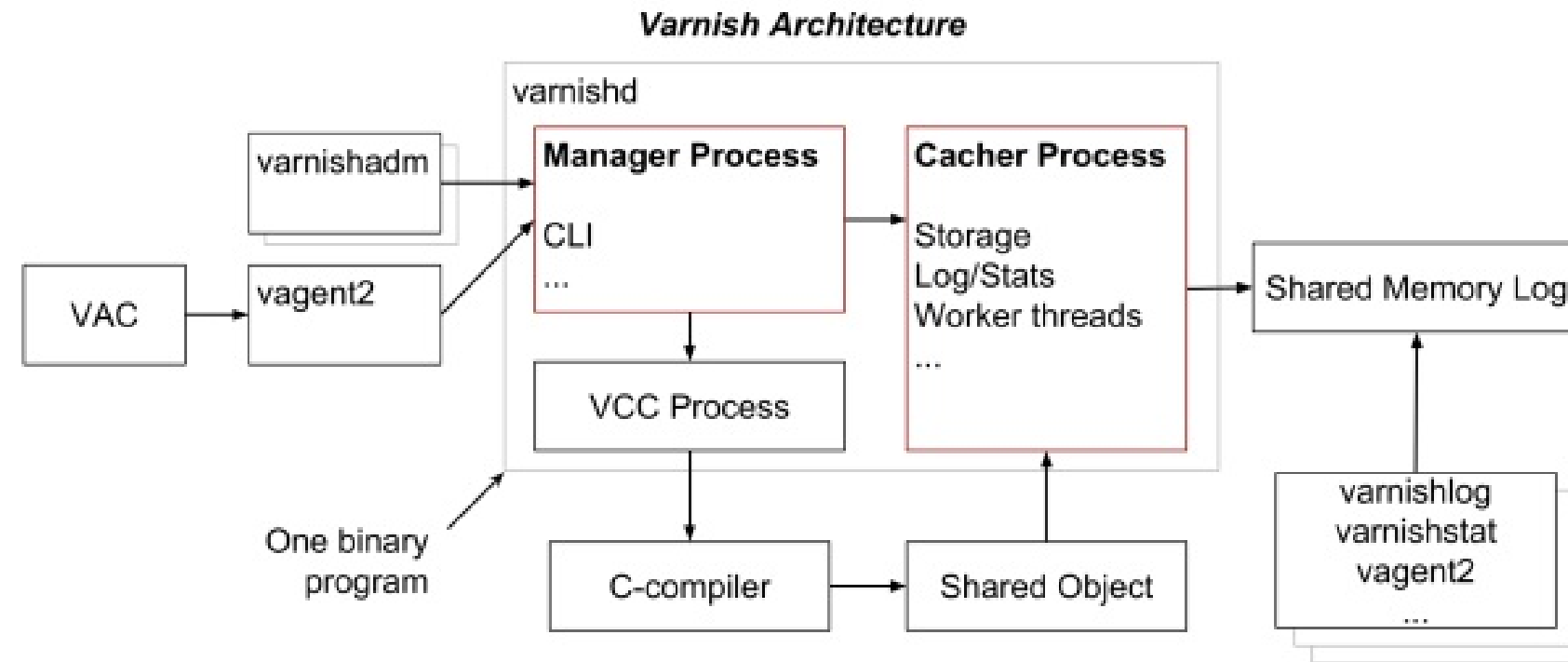
- Applique les changements de configuration;
- Délègue les tâches au Cachier et au VCC;
- Surveille Varnish;
- Fournit une invite de commande.

VCC:

- Traduit le VCL en C (Qui est ensuite traité par un compilateur traditionnel, tel que gcc).

En profondeur

Architecture



Cacher:

- Écoute les requêtes clientes;
- Gère les worker threads;
- Sauve les objets en cache;
- Log le traffic;
- Met à jour les compteurs pour les statistiques.

En profondeur

Architecture

VCL

Le Varnish Configuration Language (VCL) est le langage de configuration de Varnish. Il est notamment utilisé pour écrire ses politiques de caching. En interne, il est traduit en C.

Exemples d'utilisation:

Forcer la mise en cache de toutes les images, au format .jpg, pendant 5 minutes.

```
sub vcl_fetch {  
    if (req.url ~ "\.jpg$") {  
        set beresp.ttl = 300;  
    }  
}
```

Ne pas mettre en cache la page d'accueil index.html du site.

```
sub vcl_recv {  
    if (req.url ~ "^/index\.html" ||  
        req.url ~ "^/$") { return(pass); }  
}
```

En profondeur

Comme indiqué précédemment, Varnish se présente sous la forme d'une suite logicielle. On peut citer:

Architecture

varnishadm: Pour administrer une instance de Varnish en cours d'exécution.

VCL

varnishstat: Afficher des statistiques relatives au caching

Utilitaires

vagent2: agent vagrant permettant de faire des requête REST afin de contrôler à distance et monitorer varnish

varnishtest: permet de simuler une architecture

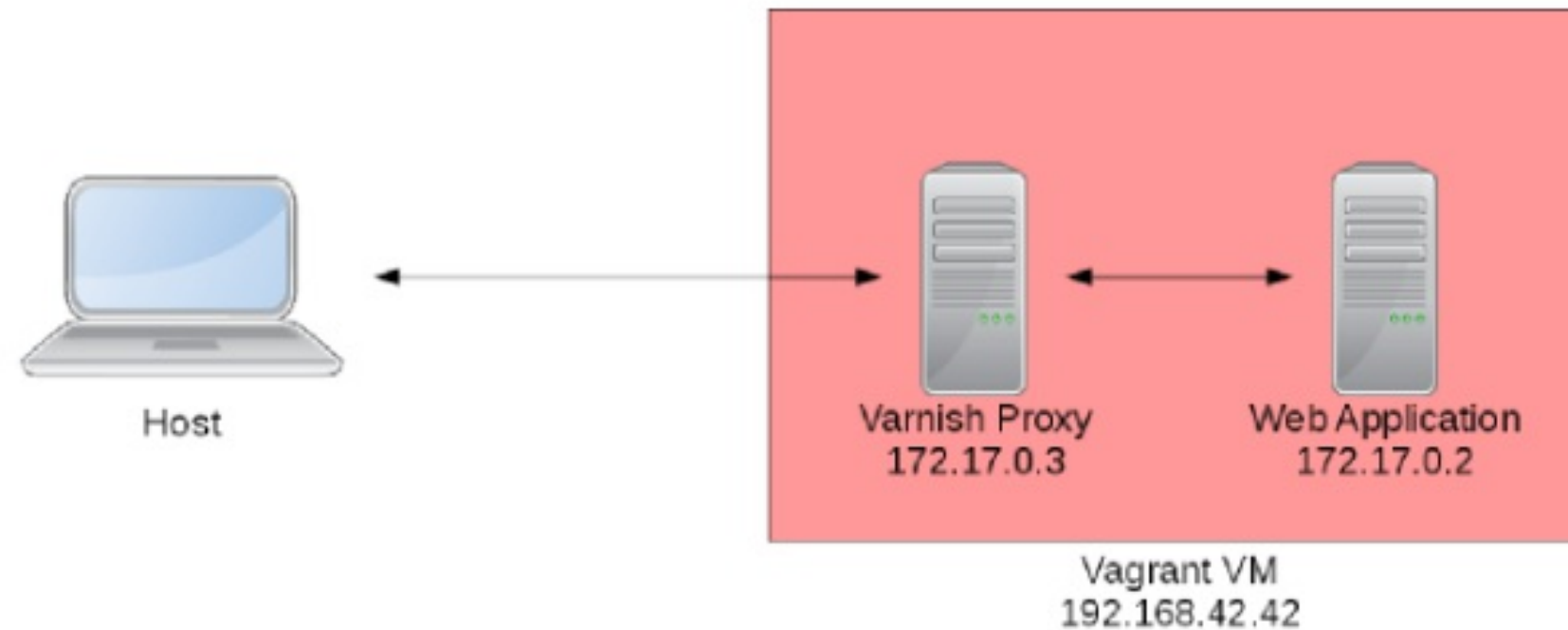
```
varnishtest "#1029"

server s1 {
    rxreq
    expect req.url == "/bar"
    txresp -gzipbody {[bar]}
} -start

client c1 {
    txreq -url "/bar" -hdr "Accept-Encoding: gzip"
    rxresp
    gunzip
    expect resp.bodylen == 5
} -run
```

Démonstration

Schéma



But :

- Envoyer des requêtes avec l'outil HTTPie de l'hôte au serveur web.
- Observer comment fonctionne le caching.
- Observer le comportement en utilisant le navigateur web Chromium.