

**Student-notities**

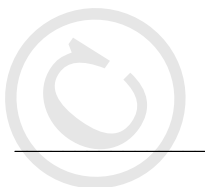
**Apache web server configuratie**

**Voorbeeld-hoofdstuk**

02. Virtual hosts

 **at computing**  
The Linux/UNIXperts

Nijmegen



---

Copyright © AT Computing 2005  
Versie: 1b

**Student-notities**

Virtual hosting betekent: meerdere websites hosten op één machine, om precies te zijn: meerdere websites hosten met één webserver.

In dit hoofdstuk zullen we de termen "virtual host" en "website" vrij door elkaar gebruiken.

Bij Microsoft IIS wordt de term "virtual host" niet gebruikt, maar wel de term "website", in dezelfde zin als wij hem hier gebruiken.

## Virtual Hosts

### Onderwerpen

- meer websites op één machine
- IP based virtual hosts
- name based virtual hosts
- combinatie IP based en name based

**Figuur 1.**

## Student-notities

Er zijn allerlei redenen om meerdere websites op dezelfde machine te hosten. Om een paar te noemen:

- een hosting bedrijf host op één machine diverse websites.
- voor een speciale actie heb je een aparte domein-naam aangeschaft, die z'n eigen website moet hebben, maar je hebt maar één IP-adres.

Anderzijds kun je binnen een website verschillende delen onderscheiden, bijvoorbeeld:

- het openbare, onbeveiligde gedeelte, versus het beveiligde gedeelte waar de gebruiker zich geauthenticeerd moet hebben — zoals bij een bank of een webwinkel.
- deel van de website bestaat uit gewone pagina's, deel bestaat uit JSP-pagina's.

De beslissing of dit door één of meer webservers afgehandeld moet worden hangt van allerlei factoren af, en moet per geval opnieuw bepaald worden. Hieronder noemen we een paar factoren op.

- Meerdere websites op dezelfde machine, achter hetzelfde IP-adres, moeten wel door één enkele webserver afgehandeld worden: de requests kunnen immers niet van tevoren onderscheiden worden.
- Meerdere websites op dezelfde machine, achter verschillende IP-adressen, kunnen zowel door verschillende webservers afgehandeld worden als door een en dezelfde webserver.
- Hetzelfde geldt voor een website die uit een onbeveiligd gedeelte en een beveiligd gedeelte bestaat.
- Eén website met een mix van gewone pagina's en JSP-pagina's moet door twee webservers afgehandeld worden — Apache en Tomcat — omdat Apache geen JSP-pagina's kan serveren, en anderzijds Tomcat niet alle configuratie-mogelijkheden heeft die Apache heeft. Het is daarbij wel mogelijk om requests voor JSP-pagina's van Apache door te sluizen naar Tomcat middels de `mod_jk` module.

Indien je de keus hebt, hangt je beslissing af van de voor- en nadelen van beide alternatieven. Om een paar te noemen:

- Twee aparte webservers voor twee websites leggen meer beslag op de resources van de computer dan één. Ze hebben elk hun eigen geheugenbeslag, en creëren elk een tiental processen in het vore.
- Eén enkele webserver heeft ook één configuratiefile die door beide gedeeld wordt. Een fout in het ene gedeelte verziekt het bij een herstart ook voor de ander.
- Bij een enkele webserver delen beide websites ook de risico's van de modules die beide gebruiken.

## Waarom en hoe meer websites?

### Diverse redenen voor meerdere websites

- een gebruiker heeft eigen website
  - eigen server, achter andere poort
- hosting
  - achter verschillende IP-adressen
  - achter hetzelfde IP-adres
- verschillende server-side technieken: JSP, CGI, ...
  - moet door verschillende servers gedaan worden
  - kan ook binnen één website het geval zijn

### Hoe implementeren?

- meerdere web-servers inzetten
  - 2 keer Apache achter ander IP-adres/poort
  - Apache als "front-end", geeft JSP door aan Tomcat
- één webserver die onderscheid maakt
  - "virtual hosting"

**Figuur 2.**

## Student-notities

De webserver kan op twee manieren onderscheid maken voor welke website een request van een browser bedoeld is:

1. Op basis van het IP-adres, of het TCP-poortnummer, waarop de request bij de server binnenkomt. Dit kan ongeacht wat voor client gebruikt wordt. Dit onderscheid is de basis van *IP based* virtual hosting.
2. Op basis van de `Host`: MIME-regel in het HTTP request. Deze regel is officieel pas sinds het HTTP 1.1 protocol verplicht. Oudere, HTTP 1.0 compliant browsers, gebruiken deze echter ook al; in ieder geval Netscape Navigator 2.0 en Microsoft Internet Explorer 3.0. Dit onderscheid is de basis van *name based* virtual hosting.

Indien twee websites achter verschillende IP-adressen gehost worden, heb je de keus deze samen door één webserver met IP based virtual hosting te laten hosten, of elk door een aparte webserver. De keuze hangt af van overwegingen als beheerscomplexiteit, performance, enz.

Het komt vaker voor dat je meerdere websites achter hetzelfde IP-adres wilt hosten: je hebt maar één machine, met één IP-adres, maar meerdere websites. In dat geval moet je name based virtual hosting gebruiken, en moet je ze door dezelfde webserver laten hosten: immers, de webserver heeft het request al aangenomen voordat hij de `Host`: regel kon lezen.

Bij gebruik van HTTPS is name based virtual hosting overigens niet mogelijk. De uitwisseling van de encryptie-keys (die per website anders zijn) gaat óók vooraf aan het lezen van het client request.

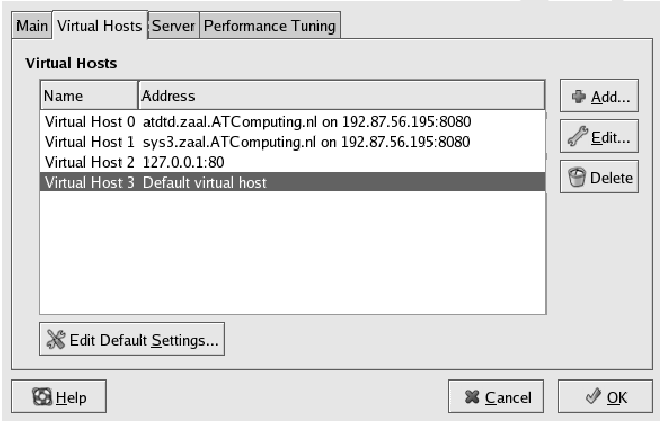
In het screenshot tonen we een combinatie van beide:

- In de eerste plaats IP based virtual hosting: *Virtual Host 2* wordt onderscheiden van de andere twee door het IP-adres.
- Vervolgens, op IP-adres 192.84.30.4, wordt name based virtual hosting gedaan tussen *Virtual Host 0* en *Virtual Host 1* op basis van de website-naam: `atdtd.ATComputing.nl` versus `desktopd.ATComputing.nl`.



## Hoe virtual hosting

### Twee manieren van virtual hosting



- IP based virtual hosting
  - onderscheid op basis van IP-adres of poort
  - zou ook door twee webserver gedaan kunnen worden
- name based virtual hosting
  - onderscheid op basis van website-naam
  - maar bij zelfde IP-adres
  - moet door één webserver gedaan worden

Figuur 3.

## Student-notities

Bij IP based virtual hosting worden websites onderscheiden op grond van het IP-adres. Bij het *globale* Listen-directive moeten alle IP-adressen genoemd worden waarnaar de webserver moet luisteren.

Voor elk IP-adres kan nu een `<VirtualHost>` container opgezet worden. Dit directive wordt een *container* genoemd omdat het op zijn beurt weer allerlei andere directives bevat. De container wordt afgesloten met `</VirtualHost>` (zonder herhaling van het IP-adres).<sup>1</sup> Binnen deze container kunnen bijna alle Apache-directives geplaatst worden. Alleen directives die van nature globaal moeten zijn, zoals `PidFile` of `Listen` of `User`, mogen niet binnen een `<VirtualHost>` gespecificeerd worden. De Apache-documentatie vermeldt bij elke directive waar het mag voorkomen.

In plaats van een IP-adres mag ook een hostnaam opgegeven worden bij `<VirtualHost>`. Mocht echter bij opstarten die hostnaam niet geresolved kunnen worden tot een IP-adres, dan werkt het niet.

De speciale waarde `_default_` zorgt ervoor dat de `<VirtualHost>` container geldt voor alle niet elders genoemde IP-adressen. Bij gebrek aan een `_default_` `VirtualHost` gelden de directives in het globale gedeelte van de configuratiefile. De `<VirtualHost>` containers worden typisch *onderaan* in de configuratiefile opgenomen.

Eventueel wordt het IP-adres gevolgd door een `:` en een poortnummer. Als een *virtual host* geldt voor meerdere IP-adressen, dan worden deze eenvoudigweg achter elkaar opgesomd met alleen (een) spatie ertussen.

Bij virtual hosting kan het globale deel van de configuratiefile heel klein blijven en slechts een handvol directives bevatten; anderzijds kunnen er ook allerlei dingen in het globale gedeelte blijven, bijvoorbeeld:

- Als je alles in één logfile wilt hebben, kun je globaal de logfile(s) specificeren.
- Als je aparte logfiles per virtual host wilt hebben, moet je de logfile(s) per virtual host specificeren.

Het directive `ServerName` is ook zonder virtual hosting nuttig; het bepaalt welke hostnaam gebruikt wordt bij redirections. Het directive `ServerAdmin` wordt gebruikt om een email-adres in fout-boodschappen te zetten die aan de browser worden teruggestuurd. Ook dit directive is altijd belangrijk te zetten. Gebruik hiervoor liefst niet je persoonlijke email-adres maar een alias zoals `webadmin`, dat doorgestuurd wordt naar degene die nu de webserver beheert.

---

1. Let op dat deze notatie doet denken aan XML. Als geheel mag de Apache configuratiefile echter niet gelden als een XML-file; daarvoor zouden alle directives in XML-formaat moeten zijn. Dit in tegenstelling tot de `metabase.xml` file van Microsoft IIS, die wel zuiver XML is.

## IP based virtual hosting

Onderscheid websites op basis van IP-adres



- per IP-adres een website definiëren
  - met default mogelijkheid

```
<VirtualHost 127.0.0.1:80>  
  ServerName localhost  
  ServerAdmin guru3@atddd.ATComputing.nl  
  DocumentRoot /var/www/local  
</VirtualHost>
```

- kost minder resources dan aparte servers
- draait wel met zelfde security

Figuur 4.

## Student-notities

Bij name based virtual hosting worden websites onderscheiden op basis van hun hostnaam. De hostnaam wordt door Apache ontleend aan de `Host:` header-regel in het HTTP request.

Name based virtual hosting kondig je aan met het directive `NameVirtualHost`, waarbij je het IP-adres(sen) moet opgeven waarop je name based virtual hosting doet. De verschillende virtual hosts worden weer gespecificeerd met elk een `<VirtualHost>` container. Deze moet als argument *precies* het IP-adres hebben dat je bij `NameVirtualHost` hebt opgegeven.

Het `ServerName` directive is nu cruciaal: hierop wordt immers de bij het HTTP request meegegeven hostnaam gematcht! Daarnaast kun je met `ServerAlias` ook nog alternatieve hostnamen voor dezelfde virtual host opgeven; daarbij zijn ook wildcards toegestaan.

De screenshot op de sheet genereert het volgende in de configuratiefile:

```
NameVirtualHost 192.87.56.195:8080
<VirtualHost 192.87.56.195:8080>
    DocumentRoot /var/www/sys3
    ServerAdmin guru2@atdtd.ATComputing.nl
    ServerName sys3.zaal.ATComputing.nl
    ServerAlias @.zaal.ATComputing.nl
    ServerAlias sys3
</VirtualHost>
```

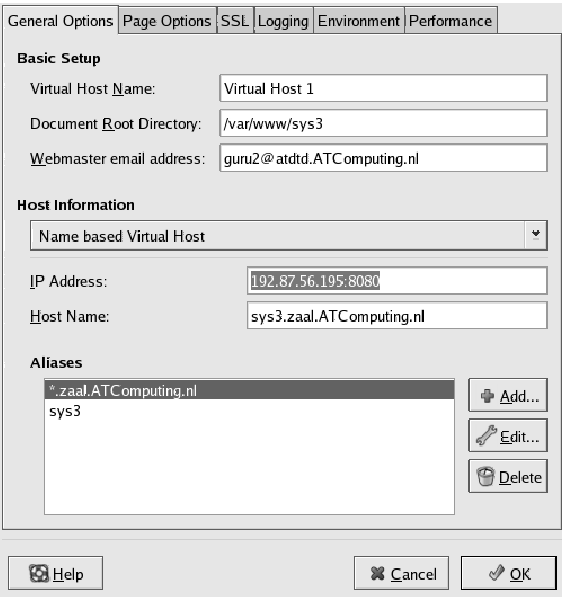
Mocht geen enkele `<VirtualHost>` matchen op de hostnaam, dan wordt *niet* naar de globale configuratie gekeken voor de `DocumentRoot`, maar wordt de *eerste* `<VirtualHost>` gekozen. Voor zaken zoals de logfiles, wordt *wel* naar de globale configuratie gekeken.

In het algemeen is name based virtual hosting eenvoudiger op te zetten dan IP based virtual hosting: je kan volstaan met dat ene IP-adres op dat ene netwerk-interface. Er is één belangrijke uitzondering: HTTPS verdraagt zich niet met name based virtual hosting, omdat ook de `Host:` regel uit het request daarbij versleuteld is.

Voor grootschalige virtual hosting heeft Apache nog een speciale module `mod_vhost_alias`, die het mogelijk maakt automatisch bij elke hostnaam een `DocumentRoot` te construeren, en zich dus ook dynamisch aanpast bij elke hostnaam die je erbij maakt.

## Name based virtual hosting

Onderscheid websites op basis van hostnaam



- meerdere websites achter zelfde IP-adres
  - name based virtual hosts moeten naar zelfde IP-adres(sen) luisteren
- onderscheid door **Host**: regel in HTTP request
- website mag meerdere namen hebben
  - aliasen
- eerste virtual host is default

Figuur 5.

## Student-notities

Deze sheet bevat een combinatie van IP based en name based virtual hosting, zoals dat in de configuratiefile terug te lezen is.

Deze webserver luistert naar drie IP-adres/poortnummer combinaties: 127.0.0.1:80, 192.87.56.195:80 en 192.87.56.195:8080.

Op de derde (192.87.56.195:80) wordt name based virtual hosting gedaan. Voor server-naam atdtd.zaal.atcomputing.nl, of z'n alias atdtd, wordt de website onder /var/www/atdtd gehost. Voor server-naam sys3.zaal.ATComputing.nl, z'n korte variant sys3, of een willekeurige (andere) server-naam in het domein zaal.atcomputing.nl wordt de website onder /var/www/sys3 gehost.

Bij elke andere naam komen we bij de eerste terecht. Een (lokale) gebruiker die de server benadert met hostnaam desktopd komt dus bij de eerste site terecht, terwijl hij bij de tweede terechtkomt als bij de site benadert onder de hostnaam desktopd.zaal.atcomputing.nl. (Deze configuratie deugt wat dat betreft dus niet!)

Vervolgens is er sprake van IP based virtual hosting. Wie de server benadert via het *loopback*-interface op 127.0.0.1 (waaraan onder Linux traditioneel de hostnaam localhost hangt), krijgt de website onder /var/www/local te zien.

Wie via een ander IP-adres/poortnummer de server benadert — in dit geval dus 192.87.56.195:80 — krijgt de website onder /var/www/html te zien.

## Virtual hosts in configuratiefile

Combinatie van IP based en name based hosting

```
Listen 80
Listen 192.87.56.195:8080
NameVirtualHost 192.87.56.195:8080
# Virtual host Virtual Host 0
<VirtualHost 192.87.56.195:8080>
    DocumentRoot /var/www/atddd
    ServerName atddd.zaal.ATComputing.nl
    ServerAlias atddd
</VirtualHost>
# Virtual host Virtual Host 1
<VirtualHost 192.87.56.195:8080>
    DocumentRoot /var/www/sys3
    ServerAdmin guru2@atddd.ATComputing.nl
    ServerName sys3.zaal.ATComputing.nl
    ServerAlias *.zaal.ATComputing.nl
    ServerAlias sys3
</VirtualHost>
# Virtual host Virtual Host 2
<VirtualHost 127.0.0.1:80>
    DocumentRoot /var/www/local
    ServerAdmin guru3@atddd.ATComputing.nl
    ServerName localhost
</VirtualHost>
# Virtual host Virtual Host 3
<VirtualHost _default_>
    DocumentRoot /var/www/html
    ServerName desktopd.zaal.ATComputing.nl
</VirtualHost>
```

- default name based virtual host: **atddd**
  - voor hostnaam **desktopa**
- default IP based virtual host: **kiekeboe**
  - voor IP-adres/poort **192.87.56.192:8080**

Figuur 6.

## Student-notities



at computing  
The Linux/UNIXperts