Web Infrastructures

RES, Lecture 9

Olivier Liechti

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Virtual Hosts

Virtual Hosts & the Host Header

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

```
> telnet 84.16.80.79 80

GET /index.html HTTP/1.1
Host: www.wasabi-tech.com
```

```
> telnet 84.16.80.79 80

GET /index.html HTTP/1.1
Host: www.otherdomain.com
```

http://httpd.apache.org/docs/2.4/vhosts/examples.html

```
# Ensure that Apache listens on port 80
Listen 80
<VirtualHost *:80>
    DocumentRoot /www/example1
    ServerName www.example.com

    # Other directives here
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot /www/example2
    ServerName www.example.org

    # Other directives here
</VirtualHost>
```

Note

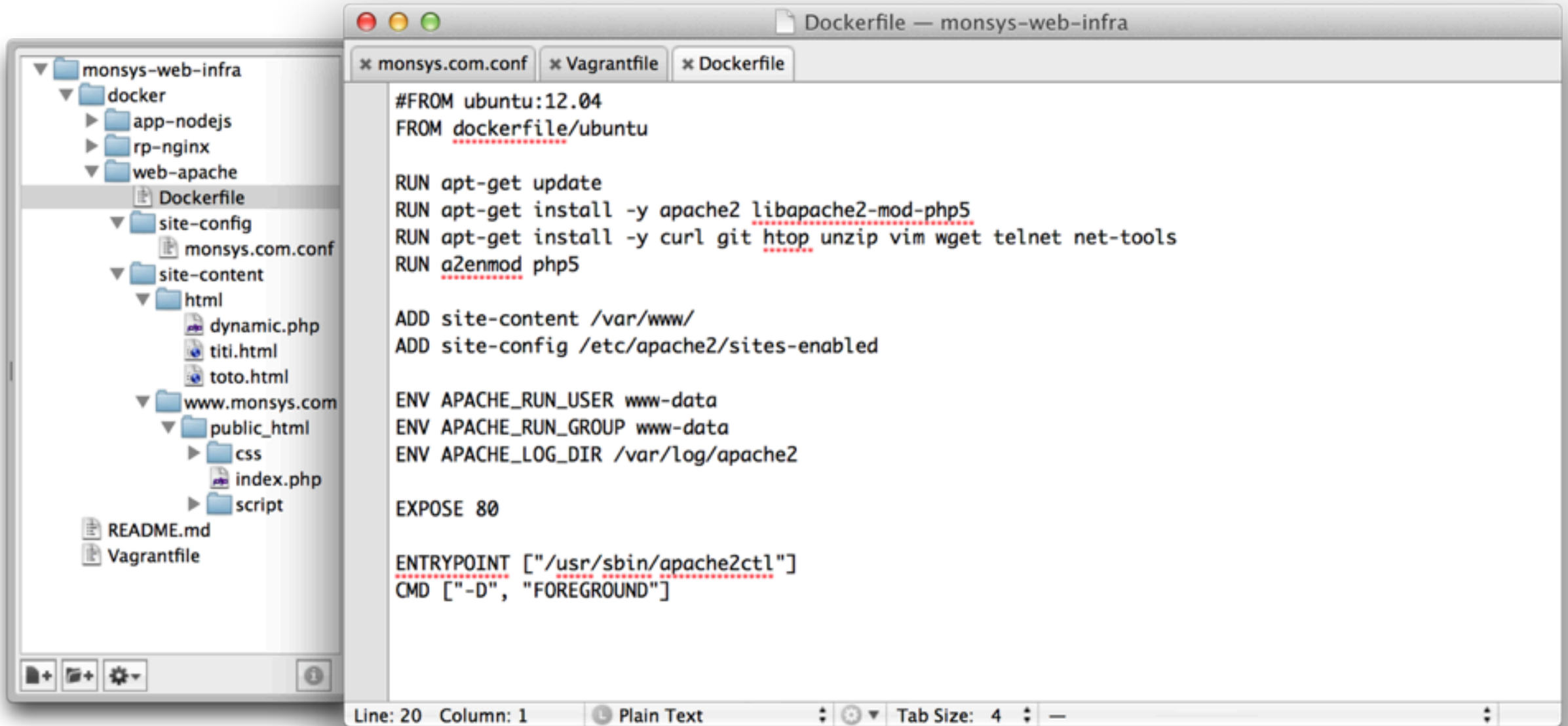
Creating virtual host configurations on your Apache server does not magically cause DNS entries to be created for those host names.
You must have the names in DNS, resolving to your IP address, or nobody else will be able to see your web site.

*You can put entries in your **hosts file** for local testing, but that will work only from the machine with those hosts entries.*

```
<VirtualHost *:*>
    ProxyPreserveHost On
    ProxyPass / http://192.168.111.2/
    ProxyPassReverse / http://192.168.111.2/
    ServerName hostname.example.com
</VirtualHost>
```

Virtual Hosts in the Lab Setup

Virtual Hosts in the Lab Setup


monsys-web-infra
docker
app-nodejs
rp-nginx
web-apache
Dockerfile
site-config
monsys.com.conf
site-content
html
dynamic.php
titi.html
toto.html
www.monsys.com
public_html
css
index.php
script
README.md
Vagrantfile
Dockerfile — monsys-web-infra
monsys.com.conf
Vagrantfile
Dockerfile
#FROM ubuntu:12.04
FROM dockerfile/ubuntu
RUN apt-get update
RUN apt-get install -y apache2 libapache2-mod-php5
RUN apt-get install -y curl git htop unzip vim wget telnet net-tools
RUN a2enmod php5
ADD site-content /var/www/
ADD site-config /etc/apache2/sites-enabled
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
EXPOSE 80
ENTRYPOINT ["/usr/sbin/apache2ctl"]
CMD ["-D", "FOREGROUND"]
Line: 20 Column: 1
Plain Text
Tab Size: 4

System Qualities & Testing

JMeter

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- Open source project, apache foundation
- http://jmeter.apache.org/index.html

"The Apache JMeter™ desktop application is open source software, a 100% pure Java application designed to ***load test functional behavior*** *and* ***measure performance****.*
It was originally designed for ***testing Web Applications*** *but has since* ***expanded to other*** *test functions."*

JMeter

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

*"Apache JMeter may be used to **test performance** both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more).*

*It can be used to **simulate a heavy load** on a server, network or object to test its strength or to analyze overall performance under **different load types**. You can use it to make a **graphical analysis of performance** or to test your server/script/object behavior under heavy **concurrent** load."*

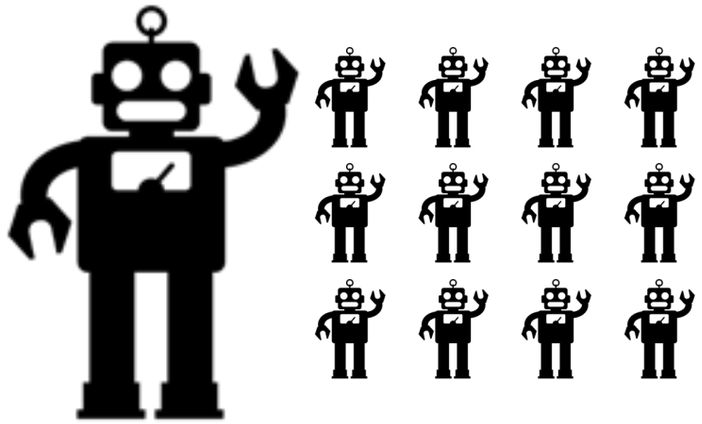
Types of tests (1)

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

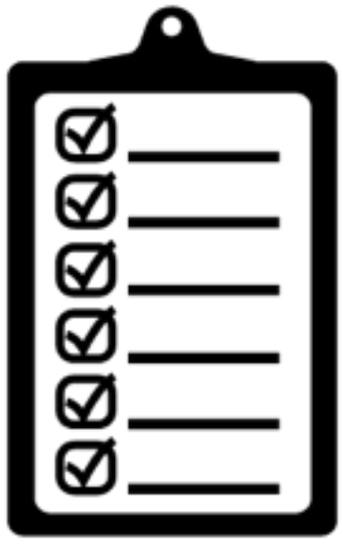
- **Functional tests**
  - Is the system doing what it is supposed to do?
  - Does its behavior comply with functional requirements (use cases)?
  - Selenium is a tool for automating functional testing of web applications (http://seleniumhq.org/)
- **Performance, load and stress tests**
  - What is the response time? What is the consumption of resources? Are there issues (e.g. concurrency issues) that happen under load?
  - Relevant both for interactive and batch use cases.

JMeter Building Blocks

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



ThreadGroup



Test Plan

Listeners
(results & stats)

Samplers
(actions)

Logic Controllers
& Assertions

Concepts

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- Test Plan
- ThreadGroup
- Samplers
- Logic Controllers
- Listeners
- Timers
- Assertions
- Configuration Elements
- Pre-Processor Elements
- Post-Processor Elements

Thread Group

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

*"Thread group elements are the **beginning points of any test plan**. All controllers and samplers must be under a thread group. [...]. As the name implies, the thread group element controls the number of threads JMeter will use to execute your test. The controls for a thread group allow you to:*

- *Set the **number of threads***
- *Set the **ramp-up** period*
- *Set the **number of times** to execute the test*

*Each thread will execute the test plan in its entirety and completely independently of other test threads. **Multiple threads are used to simulate concurrent connections to your server application.**"*

Sampler 

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

---

*"Samplers tell JMeter to **send requests to a server and wait for a response**. They are processed in the order they appear in the tree. Controllers can be used to modify the number of repetitions of a sampler.*

- *FTP Request*
- ***HTTP Request***
- *JDBC Request*
- *Java object request*
- *LDAP Request*
- *SOAP/XML-RPC Request*
- *WebService (SOAP) Request*

*Each sampler has several **properties** you can set. You can further customize a sampler by adding one or more Configuration Elements to the Test Plan."*

Logic Controllers

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

"Logic Controllers let you customize ***the logic that JMeter uses to decide when to send requests****. Logic Controllers can change the order of requests coming from their child elements. They can modify the requests themselves, cause JMeter to repeat requests, etc."*

- *Loop Controller*
- *Once Only Controller*
- *Interleave Controller*
- *Random Controller*
- *Random Order Controller*
- *Throughput Controller*
- *Runtime Controller*
- *If Controller*
- *etc.*

Listeners

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

"Listeners provide ***access to the information JMeter gathers about the test cases*** *while JMeter runs. The* ***Graph Results*** *listener plots the response times on a graph. The "****View Results Tree****" Listener shows details of sampler requests and responses, and can display basic HTML and XML representations of the response. Other listeners provide* ***summary or aggregation information****.*

Additionally, listeners can ***direct the data to a file*** *for later use.*

Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level."

Timers

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

"By default, a JMeter thread sends requests without pausing between each request. We recommend that you specify a delay by adding one of the available timers to your Thread Group. If you do not add a delay, JMeter could ***overwhelm your server*** *by making too many requests in a very short amount of time.*
The timer will cause JMeter to delay a certain amount of time before each sampler which is in its scope .

If you choose to add more than one timer to a Thread Group, JMeter takes the sum of the timers and pauses for that amount of time before executing the samplers to which the timers apply. Timers can be added as children of samplers or controllers in order to restrict the samplers to which they are applied.

To provide a pause at a single place in a test plan, one can use the ***Test Action Sampler.****"*

Assertions

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

---

*"Assertions allow you to **assert facts about responses received** from the server being tested.*

*Using an assertion, you can essentially **"test" that your application is returning the results you expect it to**.*

*For instance, you can assert that the response to a query will **contain some particular text**. The text you specify can be a Perl-style regular expression, and you can indicate that the response is to contain the text, or that it should match the whole response.*

*You can add an assertion to any Sampler. For example, you can add an assertion to a HTTP Request that checks for the text, "</HTML>". JMeter will then check that the text is present in the HTTP response. If JMeter cannot find the text, then it will **mark this as a failed request.**"*

Configuration Elements

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

"A configuration element works closely with a Sampler. Although it does not send requests (except for HTTP Proxy Server), it can add to or modify requests.

A configuration element is accessible from only inside the tree branch where you place the element. For example, if you place an ***HTTP Cookie Manager*** *inside a Simple Logic Controller, the Cookie Manager will only be accessible to HTTP Request Controllers you place inside the Simple Logic Controller"*

- *HTTP Authorization Manager*
- *HTTP Cache Manager*
- *HTTP Cookie Manager*
- *HTTP Request Defaults*
- *HTTP Header Manager*

How to Create Test Scenarios?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Option 1 : manually**
  - Create a Test Plan
  - Add a Thread Group
  - Add HTTP samplers and specify HTTP request parameters
- **Option 2 : recording with JMeter configured as an HTTP proxy**
  - http://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.pdf
  - Do manual adjustments

Advanced Usage

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Use variables:**
  - Very often, it is needed to use parts of an HTTP response into follow-up requests (e.g. session ids).
  - http://jmeter.apache.org/usermanual/test_plan.html#properties
- **Use more than one machine:**
  - With JMeter running on a single machine, it is common to “exhaust” the client before the server (especially when testing a “real” infrastructure with multiple nodes).
  - For real performance tests, it is therefore recommended to use multiple machines for injecting load into the network. One way to do it is use use virtual machines in a cloud environment (e.g. on Amazon EC2).
  - Multiple JMeter clients can be coordinated by a master and results can be collected and aggregated (http://jmeter.apache.org/usermanual/jmeter_distributed_testing_step_by_step.pdf)