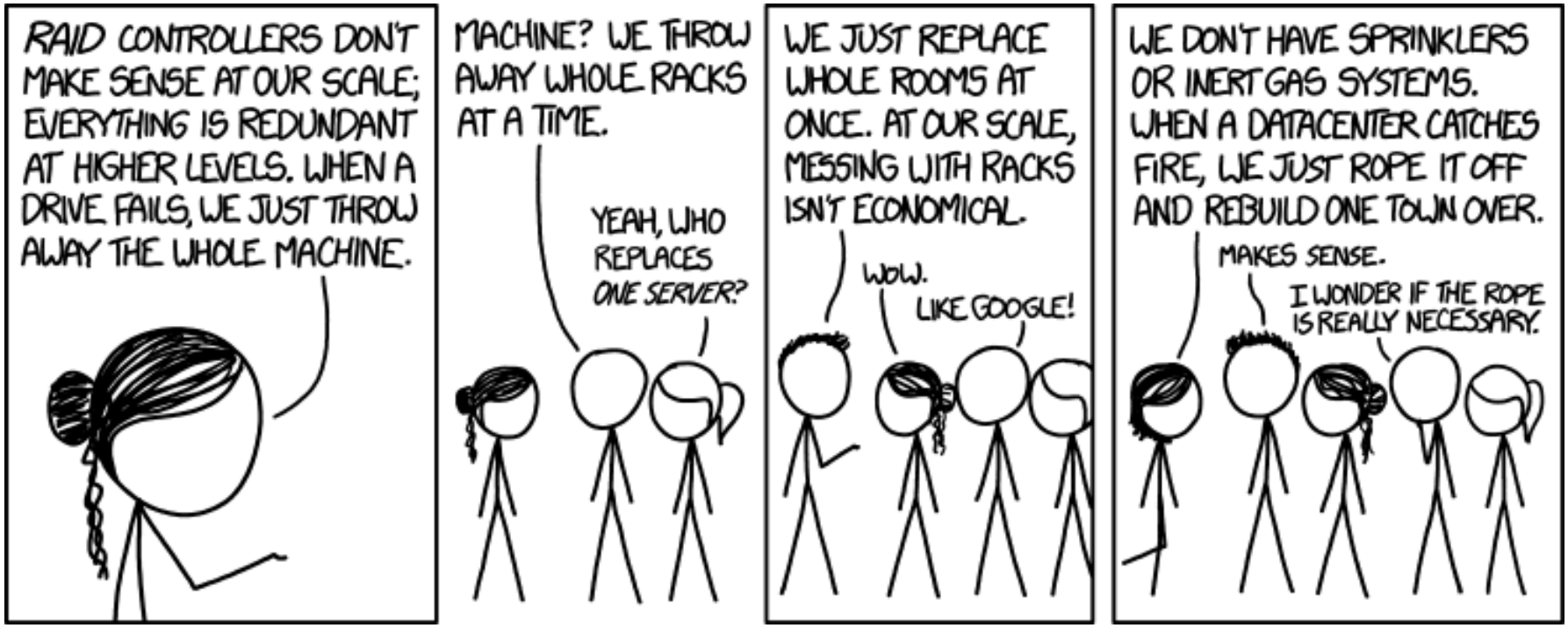
Scrivere applicazioni Web scalabili orizzontalmente

Francesco Leoncino

francesco@leoncino.net

scalare


RAID CONTROLLERS DON'T MAKE SENSE AT OUR SCALE; EVERYTHING IS REDUNDANT AT HIGHER LEVELS. WHEN A DRIVE FAILS, WE JUST THROW AWAY THE WHOLE MACHINE.
MACHINE? WE THROW AWAY WHOLE RACKS AT A TIME.
YEAH, WHO REPLACES ONE SERVER?
WE JUST REPLACE WHOLE ROOMS AT ONCE. AT OUR SCALE, MESSING WITH RACKS ISN'T ECONOMICAL.
WOW.
LIKE GOOGLE!
WE DON'T HAVE SPRINKLERS OR INERT GAS SYSTEMS. WHEN A DATACENTER CATCHES FIRE, WE JUST ROPE IT OFF AND REBUILD ONE TOWN OVER.
MAKES SENSE.
I WONDER IF THE ROPE IS REALLY NECESSARY.

Perché usare il Cloud Computing

- Per disporre di ambienti di Testing usa e getta
- **Per erogare servizi in breve tempo**
- **Per disporre di servizi scalabili**
- Per disporre di ambienti dedicati senza eccesso di risorse
- **Per avere sistemi robusti**
- Per convenienza economica

Come usare il Cloud Computing

- **Prediligere l'utilizzo di più istanze all'aumento di risorse della singola istanza.**
- **Realizzare applicazioni scalabili orizzontalmente.**
- Utilizzare istanze con il miglior rapporto costi/risorse.
- Evitare funzionalità che introducono Lock-in.

Cloud Computing vs Virtualizzazione

Server Dedicato

Virtualizzazione

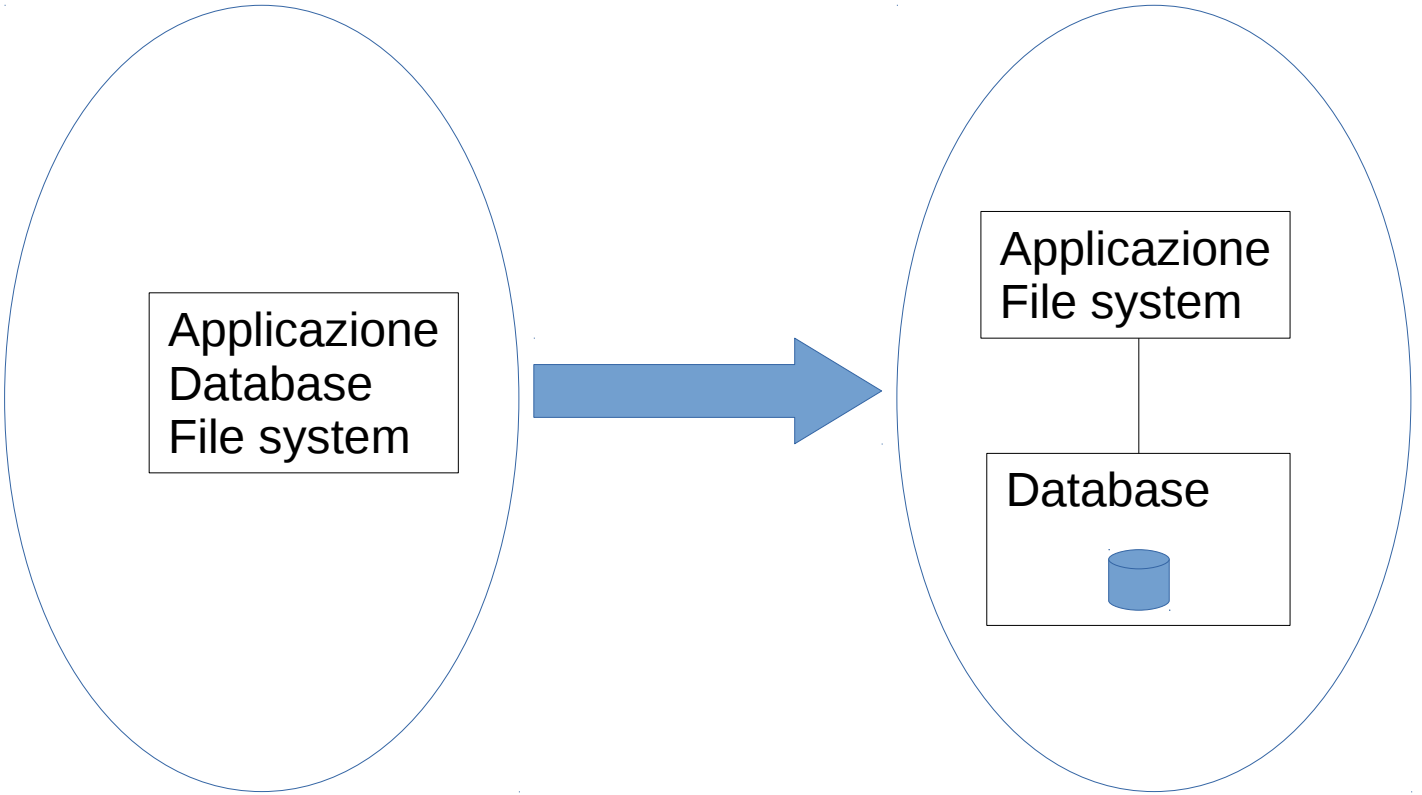
Cloud Computing

Scalare orizzontalmente

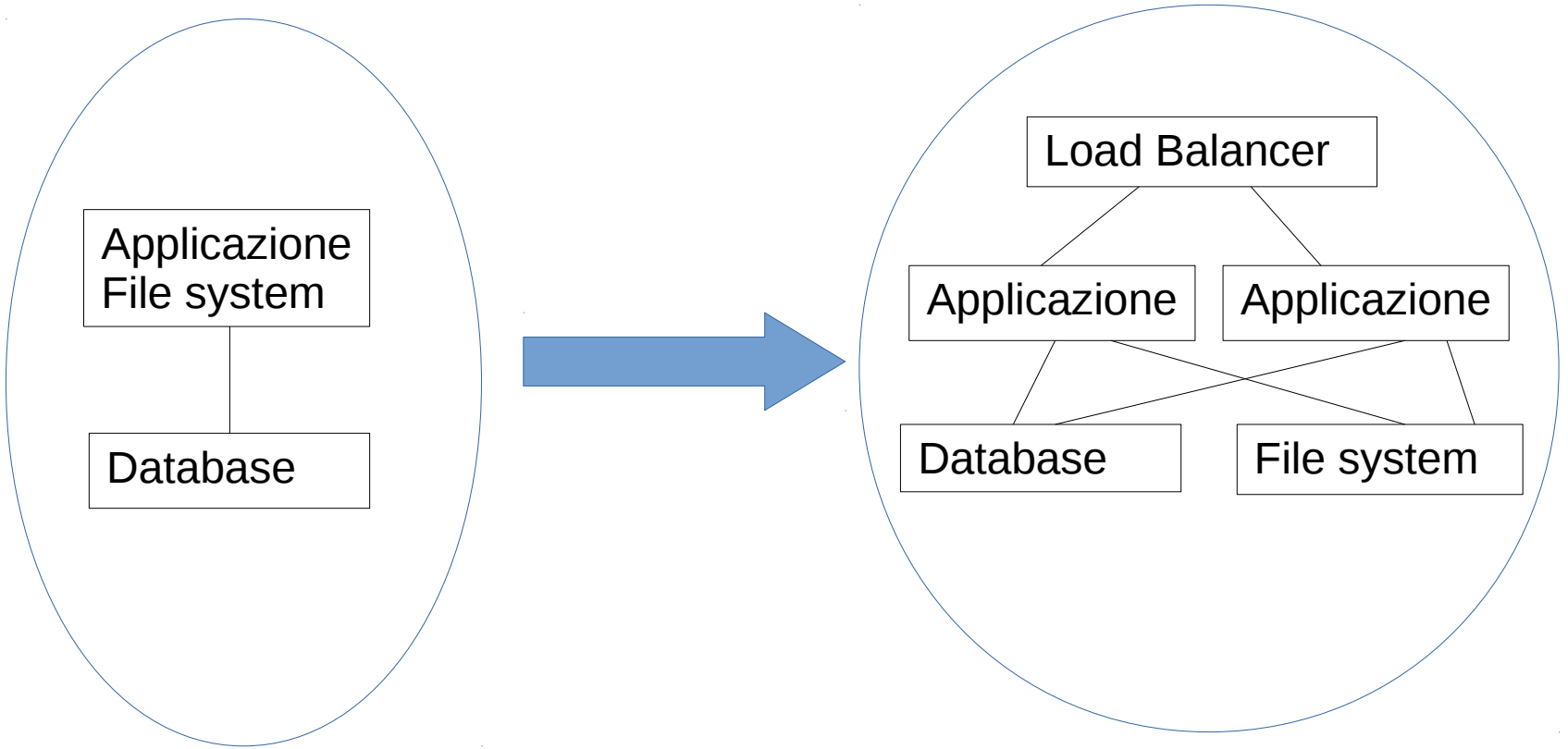
- Aggiungere/Eliminare istanze in funzione del crescere/diminuire della richiesta di risorse

- Il sistema deve supportare la distribuzione su più livelli Applicazione/Sessione/File System/Database

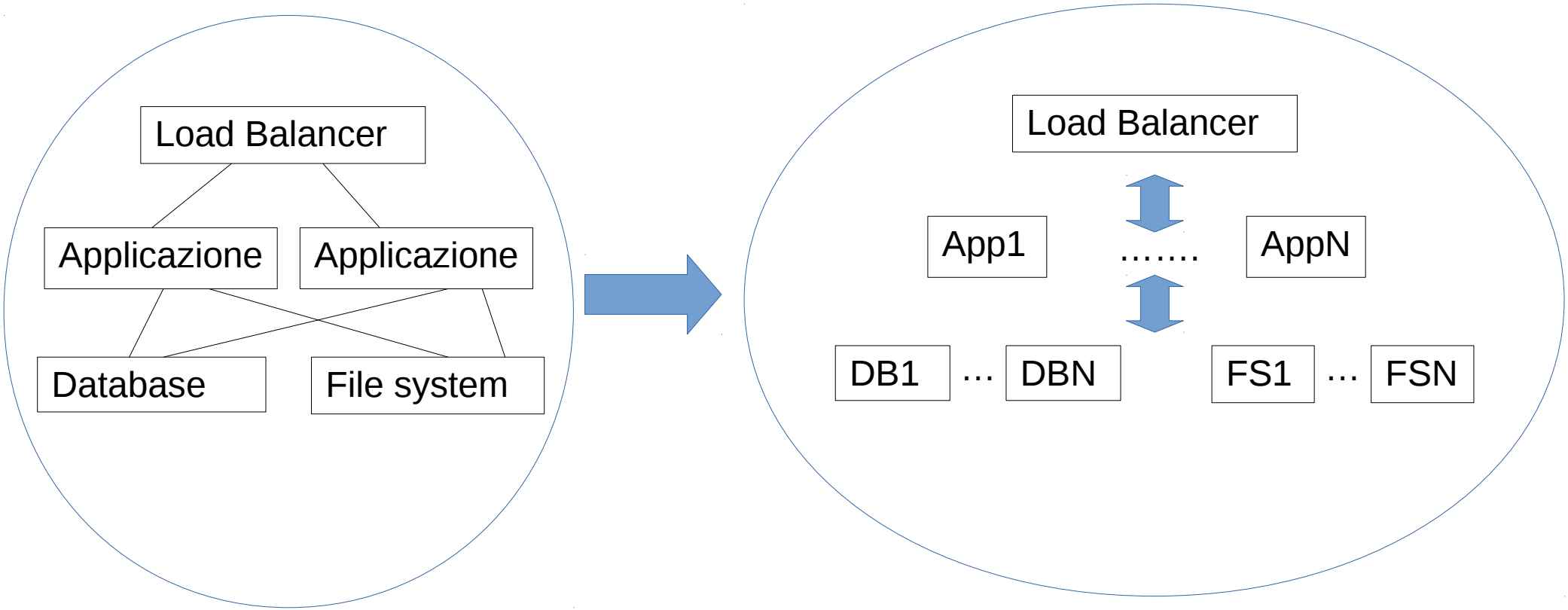
Scalare orizzontalmente



Scalare orizzontalmente



Scalare orizzontalmente



Scalare orizzontalmente LAMP

Principali nodi da affrontare

- Condivisione Sessioni
- Condivisione file
- MySQL non scala nativamente

Scalare orizzontalmente LAMP

Strumenti Open Source:

- Load balancer: HAProxy
- Sessioni: share rete/DataBase/Memcached
- File system distribuito: GlusterFS/Cloud Storage/Ceph
- DB SQL: Percona XtraDB
- DB noSQL: MongoDB

Scalare orizzontalmente LAMP

HAProxy

- Sistema di load balancing adatto sia a distribuzione di carico sia a sistemi ad alta affidabilità
- Presente nativamente nelle principali distribuzioni GNU/Linux
- Configurazione semplice, ma con un set molto evoluto di funzionalità
- Utile anche per bloccare attacchi

Scalare orizzontalmente LAMP

Condivisione SESSIONI

- Se sviluppiamo una applicazione da zero, conviene usare un cookie e gestire le variabili di sessione con proprie librerie. Molto conveniente in questo caso l'utilizzo di un DB noSQL che permette nativamente la distribuzione del carico.
- Alcuni CMS, es. WordPress gestiscono già la sessione su DB
- Se non possiamo intervenire sull'applicazione possiamo utilizzare un percorso condiviso, o intervenire sul manager di sessioni di PHP utilizzando il meccanismo di override delle sessioni e la direttiva *auto_prepend_file*

Scalare orizzontalmente LAMP

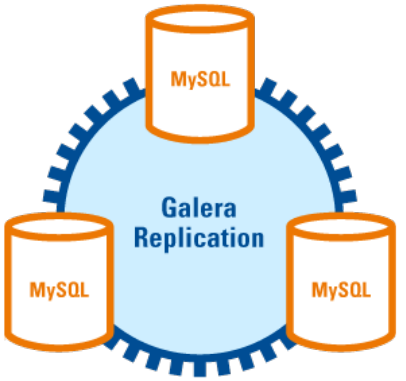
File System

- Se le richieste verso il file system non sono onerose si può utilizzare una condivisione NFS.
- Se invece richiediamo un maggiore grado di scalabilità è possibile utilizzare GlusterFS con cui è possibile definire volumi di storage distribuiti sia al fine di ottimizzare l'accesso, sia al fine di ottenere un'alta disponibilità del file system.
- Per i contenuti statici dell'applicazione possono essere usati servizi esterni di cloud storage con protocollo S3.

Scalare orizzontalmente LAMP

Database

- MySQL non presenta nativamente un sistema di replica sincrona.
- Sono disponibili diverse soluzioni al problema, una di queste è Percona XTRADB Cluster che rappresenta una soluzione open source per il clustering attivo/attivo di MySQL.
- Il Cluster integra un motore (Percona server), un sistema di backup (XtraBackup) e la replica multimaster Galera Cluster.



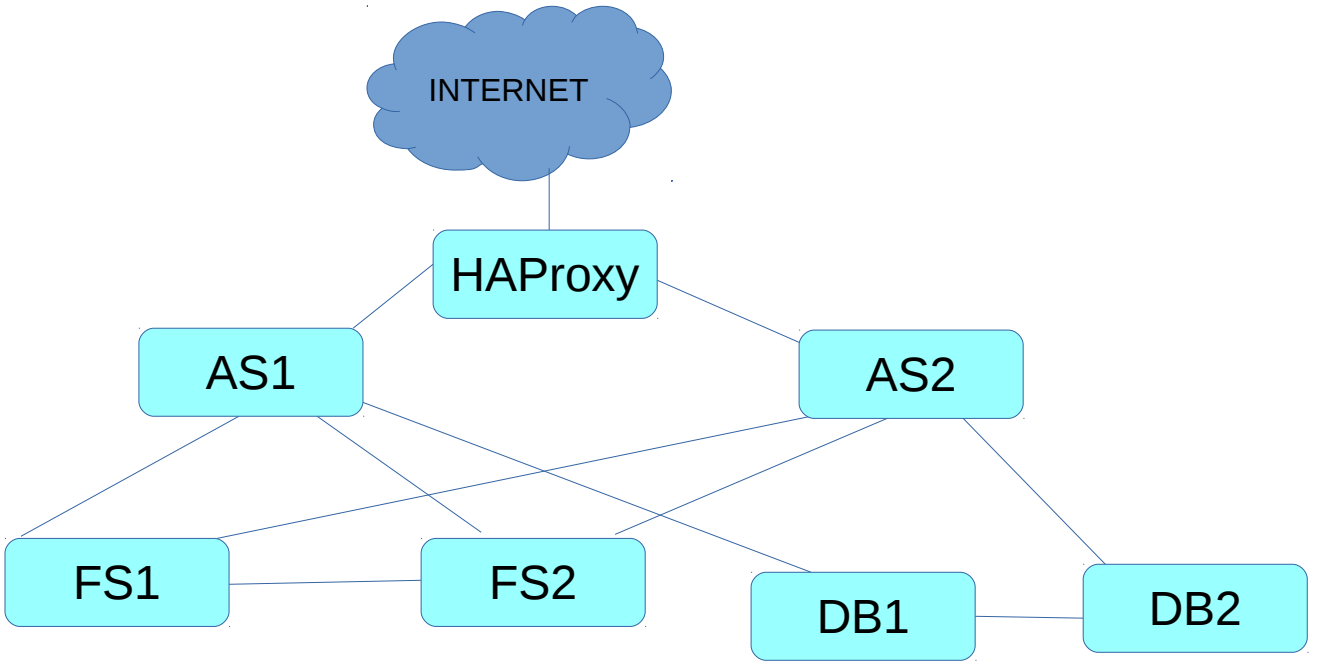
Esempio

Script php con sessione condivisa su:

- due web server bilanciati da Haproxy
- cartella di sessione su file system condiviso GlusterFS

http://lday.cloudsolutions.it

Esempio



Esempio

index.html

```
<html><body><form method=POST action="set.php">
<input name="test"><br/>
<input type="submit">
</form></body></html>
```

set.php

```
<?php
session_start();
$_SESSION["v"] = $_POST['test'].  " " . $_SERVER['SERVER_ADDR'];
echo "impostata variabile di sessione v:" . $_SESSION["v"] . "<br />";
echo "<a href=get.php>clicca qui per ripescarla</a>";
?>
```

get.php

```
<?php
session_start();
echo "SERVER IP: " . $_SERVER['SERVER_ADDR'] . "<br/>";
echo "SESSION VAR: " . $_SESSION['v'] . "<br/>";
?>
```

Esempio

Installazione Wordpress su:

- due web server bilanciati da Haproxy
- file system condiviso GlusterFS (2 server in mirror)
- cluster database Percona XtraDB (2 server attivo/attivo)

http://wp.cloudsolutions.it

GRAZIE

francesco@leoncino.net