**Spike: Serve different applications from a unique IP and provide high availability**

This is a typical case of load balancing where the http traffic is distributed based on the URL context matching. This can be achieved by installing **Nginx** on a server and configuring it to work as a reverse proxy. A static public ip can be attached to this proxy server (80.239.171.146) to keep receiving traffic at **80.239.171.146:80**

An error code of 404 will be sent to all invalid referrers.

A server block (virtual host) needs to be configured. A new file Springer-app needs to be created in the location **/etc/nginx/sites-available/Springer-app**.The following server block needs to be created in that file. The configuration needs to be activated further by creating a symlink of this file in the directory **/etc/nginx/sites-enabled/** and the reloading ngnix.

Server {

listen 80; #ensures that the server listens to traffic coming to port 80

location / {

rewrite ^/ $1 break;

proxy\_pass http://sales;

}

location /sales {

rewrite ^/sales(.\*) $1 break;

proxy\_pass http://myapp;

}

if ($invalid\_referer) {

return 404;

}

}

‘**myapp’** and ‘**sales’** are server groups which define the group of servers serving to each of the respective applications. The configurations are as follows

upstream **myapp** {

server 10.20.30.1:8080 ;

server 10.20.30.2:8080;

}

upstream **sales** {

server 10.20.30.3.8080;

server 10.20.30.4:8080;

}

The invalid\_referers are defined by the definition of the valid\_referers below

Valid\_referers ~80.239.171.146:80/ ~80.239.171.146:80/sales ~80.239.171.146/ ~80.239.171.146/sales;

Amongst the available cloud based web services, **Azure Application Gateway** offers the same service as is achieved above, but is a paid service.