Understand Web Servers

- What's Apache and HTTP/HTTPS protocol
- Understanding the Apache compilation
- Run two apache versions on same machine.
- Understanding Nginx and use as reverse proxy to serve static content

What’s Web Server?

- It is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web

- All computers that host Web sites must have Web server programs

- Leading Web servers include Apache (the most widely-installed Web server), Microsoft's Internet Information Server (IIS) and nginx (pronounced engine X) from NGNIX

- Other Web servers include Novell's NetWare server, Google Web Server (GWS) and IBM's family of Domino servers.

What’s HTTP?

- Hypertext Transfer Protocol

- Standard port: 80

- An application protocol for distributed, collaborative, and hypermedia information systems

- It’s the set of rules for transferring files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web

- URL beginning with the HTTP scheme and the WWW

domain name label.

What’s HTTPS?

- Hyper Text Transfer Protocol Secure (**HTTPS**)

- Standard port: 433

- It is an adaptation of the Hypertext Transfer Protocol (HTTP) for secure communication over a computer network

- In HTTPS, the communication protocol is encrypted by Transport Layer Security (TLS), or formerly, its predecessor, Secure Sockets Layer (SSL)

- The protocol is therefore also often referred to as HTTP over TLS, or HTTP over SSL.

- Illustration of the networking protocol https and the

www letters

How HTTPS works?

What’s Apache HTTP Server?

- Colloquially called **Apache**
- Initial release 1995
- Original author - **Robert McCool**
- Free and open-source cross-platform web server software
- Released under the terms of Apache License 2.0
- Developed and maintained by an open community of developers under the auspices of the **Apache Software Foundation**
- Stable release 2.4.29 (October 23, 2017)

Requirements - Compiling and Installing

- **APR and APR-Util** : The mission of the Apache Portable Runtime (APR) project is to create and maintain software libraries that provide a predictable and consistent interface to underlying platform-specific implementations
- **Perl-Compatible Regular Expressions Library (PCRE)**
- **Disk Space** : You have at least 50 MB of temporary free disk space available. After installation the server occupies approximately 10 MB of disk space.
- **ANSI-C Compiler and Build System** : The GNU C compiler (GCC) from the Free Software Foundation (FSF) is recommended
- **Perl 5 [OPTIONAL]** : For some of the support scripts like apxs or dbmmanage (which are written in Perl) the Perl 5 interpreter is required

- **Command to install requirements** :
  - *apt-get install gcc make libpcre3-dev libpcre3 libexpat1 libexpat1-dev*
  - *yum install gcc make pcre-devel pcre expat1 expat-devel* ***# command on rhel***

Download - Compiling and Installing

- The Apache HTTP Server can be downloaded from - http://httpd.apache.org/download.cgi

Extract - Compiling and Installing

- *wget -c http://www-eu.apache.org/dist//httpd/httpd-2.4.29.tar.gz*
- *tar xzvf httpd-2.4.29.tar.gz*
- *wget -c http://www-us.apache.org/dist//apr/apr-1.6.3.tar.gz*
- *wget -c http://www-us.apache.org/dist//apr/apr-util-1.6.1.tar.gz*
- *tar xzvf apr-util-1.6.1.tar.gz*
- *tar xzvf apr-1.6.3.tar.gz*
- *mv apr-1.6.3 httpd-2.4.29/srclib/apr*
- *mv apr-util-1.6.1 httpd-2.4.29/srclib/apr-util*
- *cd httpd-2.4.29/*

Configuring the source tree - Compiling and Installing

- This is done using the script configure included in the root directory of the distribution.

- To configure the source tree using all the default options, simply type ./configure

- To change the default options, configure accepts a variety of variables and command line options

- **Command to execute configure** :

  - *./configure --prefix=/opt/apache2 --with-included-apr --with-pcre*

Build - Compiling and Installing

- Now you can build the various parts which form the Apache package by simply running the command:

  - *make*

Please be patient here, since a base configuration takes several minutes to compile and the time will vary widely depending on your hardware and the number of modules that you have enabled.

Install - Compiling and Installing

- Now it's time to install the package under the configured installation PREFIX (see --prefix option above) by running:

  - *make install*

This step will typically require root privileges, since PREFIX is usually a directory with restricted write permissions.

Customize - Compiling and Installing

- You can customize your Apache HTTP server by editing the configuration files under PREFIX/conf/

  - *vim PREFIX/conf/httpd.conf*

Test - Compiling and Installing

- Now you can start your Apache HTTP server by immediately running:

  - *PREFIX/bin/apachectl -k start*

- You should then be able to request your first document via the URL **http://YOUR_IP_ADDRESS/**. The web page you see is located under the DocumentRoot, which will usually be PREFIX/htdocs/. Then stop the server again by running:

  - PREFIX/bin/apachectl -k stop

What’s Nginx?

- Stylized as **NGINX**, **NGiИX** or **nginx**
- It’s a **Web server**
- Initial release **4 October 2004**
- Original author **Igor Sysoev**
- Written in **C**
- It can also be used as a **reverse proxy**, **load balancer** and **HTTP cache**

What’s Reverse Proxy?

- A proxy server is a go-between or intermediary server that forwards requests for content from multiple clients to different servers across the Internet

- It is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server

- It provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers

Common uses for a reverse proxy server

- Load balancing : A reverse proxy server can act as a “traffic cop,” sitting in front of your backend servers and distributing client requests across a group of servers in a manner that maximizes speed and capacity utilization while ensuring no one server is overloaded, which can degrade performance. If a server goes down, the load balancer redirects traffic to the remaining online servers.

- Web acceleration : Reverse proxies can compress inbound and outbound data, as well as cache commonly requested content, both of which speed up the flow of traffic between clients and servers. They can also perform additional tasks such as SSL encryption to take load off of your web servers, thereby boosting their performance.

- Security and anonymity : By intercepting requests headed for your backend servers, a reverse proxy server protects their identities and acts as an additional defense against security attacks. It also ensures that multiple servers can be accessed from a single record locator or URL regardless of the structure of your local area network.

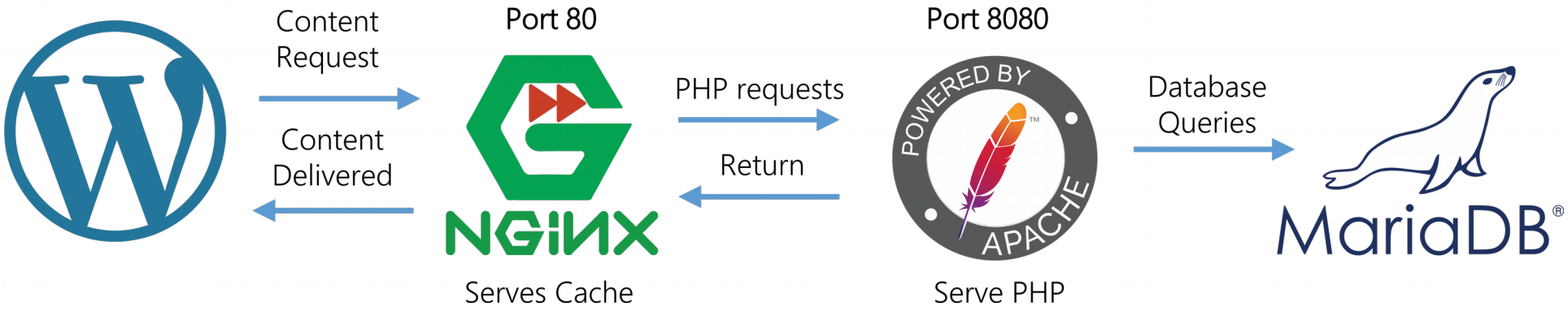
Nginx Reverse Proxy

Nginx Detailed Use Case

User gets static content fast like HTML, javascript and css.

nginx caches all WordPress posts and pages, passes PHP requests to Apache

Apache handles all PHP requests and serves compiled PHP content to nginx



Installing and Configuring ..to be continued

Install Nginx.

- *sudo apt-get install nginx*

Then remove the default virtual host's symlink.

- sudo rm /etc/nginx/sites-enabled/default

Now we'll create virtual hosts for Nginx. First make document root directories for both the websites:

- *sudo mkdir -vp /usr/share/nginx/devops.ap2v.com/static/*
- *echo "<h1 style='color: red;'>devops.ap2v.com</h1>" | sudo tee /usr/share/nginx/devops.ap2v.com/static/.html*

Installing and Configuring ..to be continued

- *vim /etc/nginx/sites-available/devops.ap2v.com*

```
server {
    listen 80;
    server_name 127.0.0.1 ap2v.com devops.ap2v.com;
    location /static/ {
        root /usr/share/nginx/devops.ap2v.com/;
    }
    location / {
        proxy_pass http://127.0.0.1:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Installing and Configuring

Active the website

- *ln -s /etc/nginx/sites-available/devops.ap2v.com /etc/nginx/sites-enabled/*

Start the service:

- *service nginx start*

Lets see practical.