NGINX

How to Install Nginx

install NGINX Open Source:

```
[root@server ~]# yum install nginx
Updating Subscription Management repositories.
Unable to read consumer identity
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Last metadata expiration check: 20:51:03 ago on Tue 24 May 2022 04:10:24 PM IST.
Dependencies resolved.
================================================================================
 Package                    Architecture          Version
================================================================================
Installing:
```

Verify the installation:

```
[root@server ~]# nginx -v
nginx version: nginx/1.14.1
[root@server ~]#
```

Nginx installed, you can start, enable and verify the status by running

```
[root@server ~]# systemctl start nginx
[root@server ~]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.
[root@server ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-05-25 13:05:03 IST; 17s ago
 Main PID: 5832 (nginx)
    Tasks: 2 (limit: 11160)
   Memory: 9.4M
   CGroup: /system.slice/nginx.service
           ├─5832 nginx: master process /usr/sbin/nginx
```

Open and enable port 80 and 443 to allow web traffic on Nginx on the system firewall

```
[root@server ~]# systemctl start firewalld
[root@server ~]# firewall-cmd --zone=public --permanent --add-service=http
success
[root@server ~]# firewall-cmd --zone=public --permanent --add-service=https
success
[root@server ~]# firewall-cmd --reload
success
[root@server ~]#
```

Verify that the port 80 and 443 enabled on the firewall

```
[root@server ~]# ss -tulpn
Netid       State        Recv-Q       Send-Q              Local Address:Port                Peer Ad
udp         UNCONN       0            0                   127.0.0.1:323                        0.
udp         UNCONN       0            0                     0.0.0.0:20048                      0.
```

```
tcp   LISTEN   0   128   [::]:111    [::]:*   users:(("rpcbind",pid=958,fd=6),("systemd",pid=1,fd=73))
tcp   LISTEN   0   128   [::]:80     [::]:*   users:(("nginx",pid=5833,fd=9),("nginx",pid=5832,fd=9))
tcp   LISTEN   0   128   [::]:20048  [::]:*   users:(("rpc.mountd",pid=1345,fd=11))
tcp   LISTEN   0   10    [::1]:53    [::]:*   users:(("named",pid=2669,fd=22))
tcp   LISTEN   0   32      *:21        *:*    users:(("vsftpd",pid=1069,fd=3))
```


CREATED BY:- VIVEK GANDHI

NGINX

open your web browser and type the IP address.

How to Use Nginx as an HTTP Load Balancer in Linux

Nginx can be deployed as an efficient HTTP load balancer to distribute incoming network traffic and workload among a group of application servers, in each case returning the response from the selected server to the appropriate client.

The load balancing methods supported by Nginx are:

round-robin – which distributes requests to the application servers in a round-robin fashion. It is used by default when no method is specified,

least-connected – assigns the next request to a less busy server(the server with the least number of active connections),

ip-hash – where a hash function is used to determine what server should be selected for the next request based on the client's IP address. This method allows for session persistence (tie a client to a particular application server).

create a server block file called `/etc/nginx/conf.d/loadbalancer.conf` both side

```
[root@server conf.d]# vi /etc/nginx/conf.d/loadbalancer.conf
[root@server conf.d]#
```

```
upstream backend {
        server 192.168.254.149;        server ip
        server 192.168.254.154;        client ip
    }


server {
        listen 127.0.0.1:80;        DNS IP
        server_name server.linux.com;        DNS NAME
        location / {
                #root /usr/share/nginx/html;
                #index index.html;
                proxy_redirect      off;
                proxy_set_header    X-Real-IP $remote_addr;
                proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header    Host $http_host;
                proxy_pass http://backend;
        }
}
~
```


CREATED BY:- VIVEK GANDHI

Save the file and exit it. Then ensure the Nginx configuration structure is correct after adding the recent changes, by running the following command.

```
[root@server nginx]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@server nginx]#
```

configuration is OK, restart and enable the Nginx service to apply the changes

```
[root@server nginx]# systemctl restart nginx
[root@server nginx]# systemctl enable nginx
```

Html configuration path server 1

```
[root@server ~]# cd /usr/share/nginx/html/
[root@server html]# ls
404.html  category.html  css                                img         js   LICENSE.txt  nginx-logo.png  READ-ME.txt  single.html
50x.html  contact.html   free-bootstrap-magazine-template.jpg  index.html  lib  mail         poweredby.png   scss
[root@server html]#
```

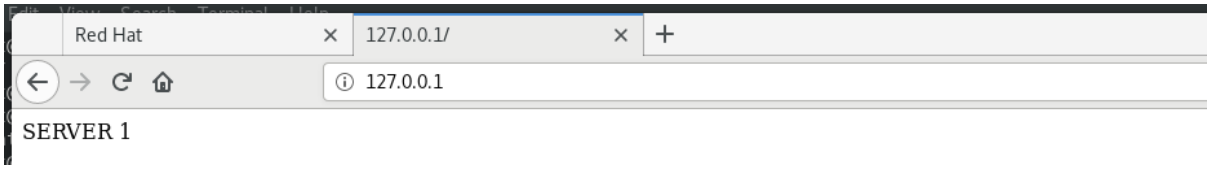
```
[root@server html]# vim index.html
<html>
        <head>
                <body>
                        SERVER 1
                </body>
        </head>
</html>
~
```

Html configuration path server 2

root@client:~

```
[root@client ~]# vi /usr/share/nginx/html/index.html
<html>
        <head>
                <body>
                        SERVER 2
                </body>
        </head>
</html>
~
```

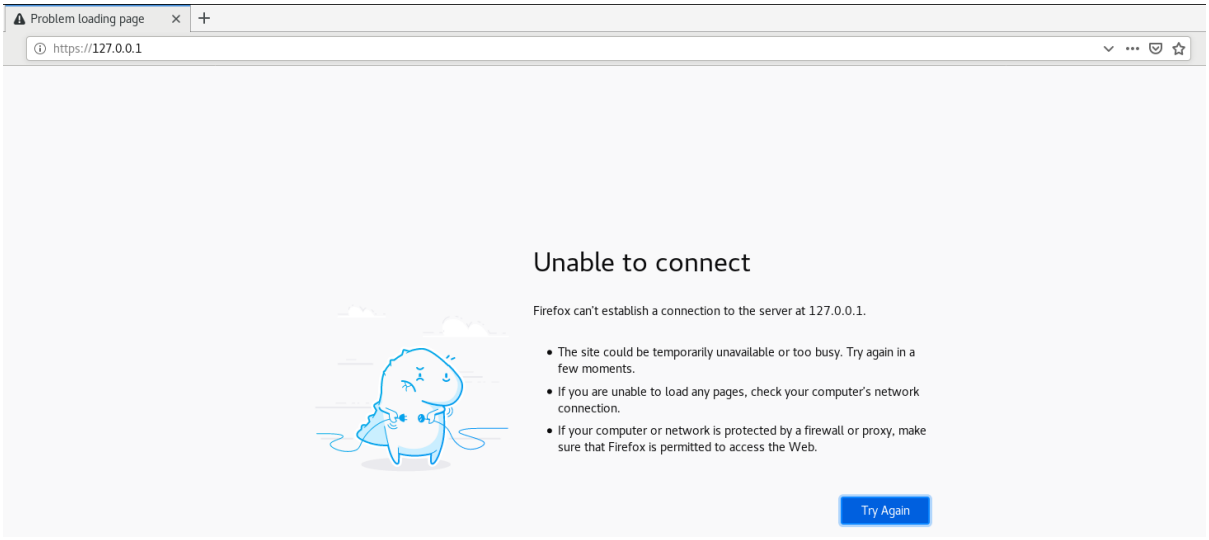
test



NGINX

Access SSL Certified

Without ssl not open page with secure both side



Open configuration file

root@server:~

```
[root@server ~]# vi /etc/nginx/nginx.conf
```

Enable all contain uncomment all lines

```
    server {
        listen       443 ssl http2 default_server;
        listen       [::]:443 ssl http2 default_server;
        server_name  _;
        root         /usr/share/nginx/html;

        ssl_certificate "/etc/pki/nginx/nginx.crt";
        ssl_certificate_key "/etc/pki/nginx/private/nginx.key";
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_ciphers PROFILE=SYSTEM;
        ssl_prefer_server_ciphers on;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
        }

        error_page 404 /404.html;
            location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
            location = /50x.html {
        }
    }

}
```


CREATED BY:- VIVEK GANDHI

NGINX

Comments all lines

```
# Load modular configuration files from the /etc/nginx/conf.d directory.
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
#include /etc/nginx/conf.d/*.conf;

# server {
#        listen       127.0.0.1:80;
#       server_name  server.linux.com;
#      root         /usr/share/nginx/html;

# Load configuration files for the default server block.
#     include /etc/nginx/default.d/*.conf;
#
#       location / {
#      }
#
#       error_page 404 /404.html;
#          location = /40x.html {
#     }
#
#       error_page 500 502 503 504 /50x.html;
#          location = /50x.html {
#     }
#}

# Settings for a TLS enabled server.
```

Create dir

root@server:/etc/pki/nginx/private

```
[root@server ~]# mkdir -p /etc/pki/nginx/private/
[root@server ~]# cd /etc/pki/nginx/private/
[root@server private]#
```

Execute openssl command for genrate nginx ssl key

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/nginx/private/nginx.key -out /etc/pki/nginx/nginx.crt

```
[root@server nginx]# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/nginx/private/nginx.key -out /etc/pki/nginx/nginx.crt
Generating a RSA private key
...+++++
..+++++
writing new private key to '/etc/pki/nginx/private/nginx.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:IN
State or Province Name (full name) []:GUJRAT
Locality Name (eg, city) [Default City]:SURAT
Organization Name (eg, company) [Default Company Ltd]:VIVEK
Organizational Unit Name (eg, section) []:H0
Common Name (eg, your name or your server's hostname) []:server.linux.com
Email Address []:root@server.linux.com
[root@server nginx]#
```

Check key genrated or not

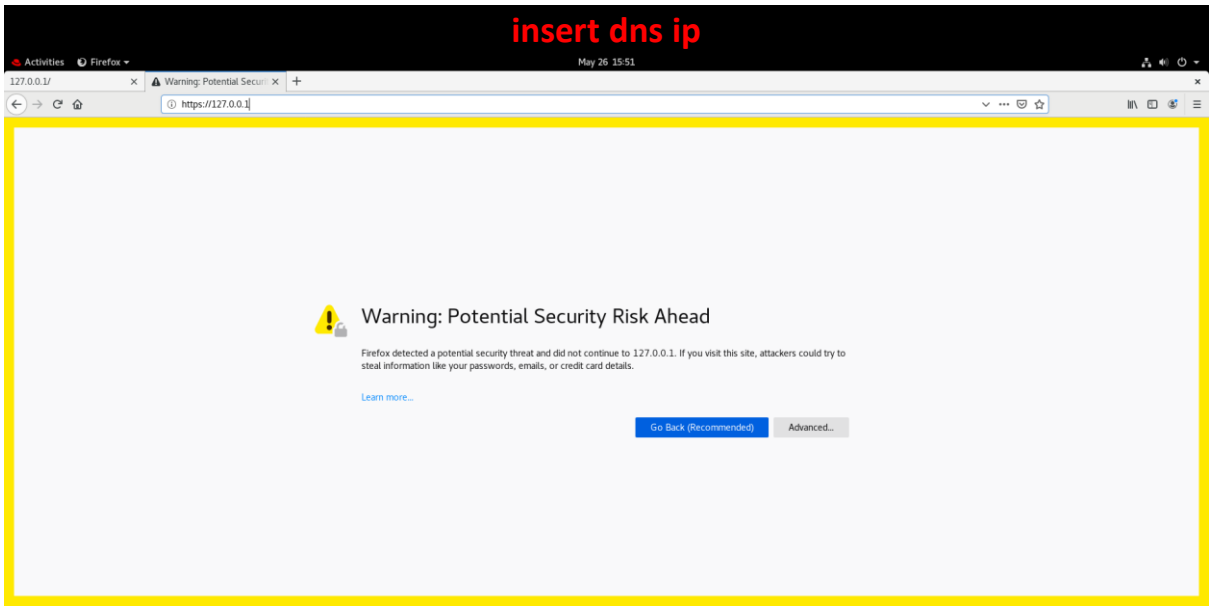
```
[root@server nginx]# cd /etc/pki/nginx/
[root@server nginx]# ls
nginx.crt  private
[root@server nginx]# cd private/
[root@server private]# ls
nginx.key
[root@server private]#
```


CREATED BY:- VIVEK GANDHI

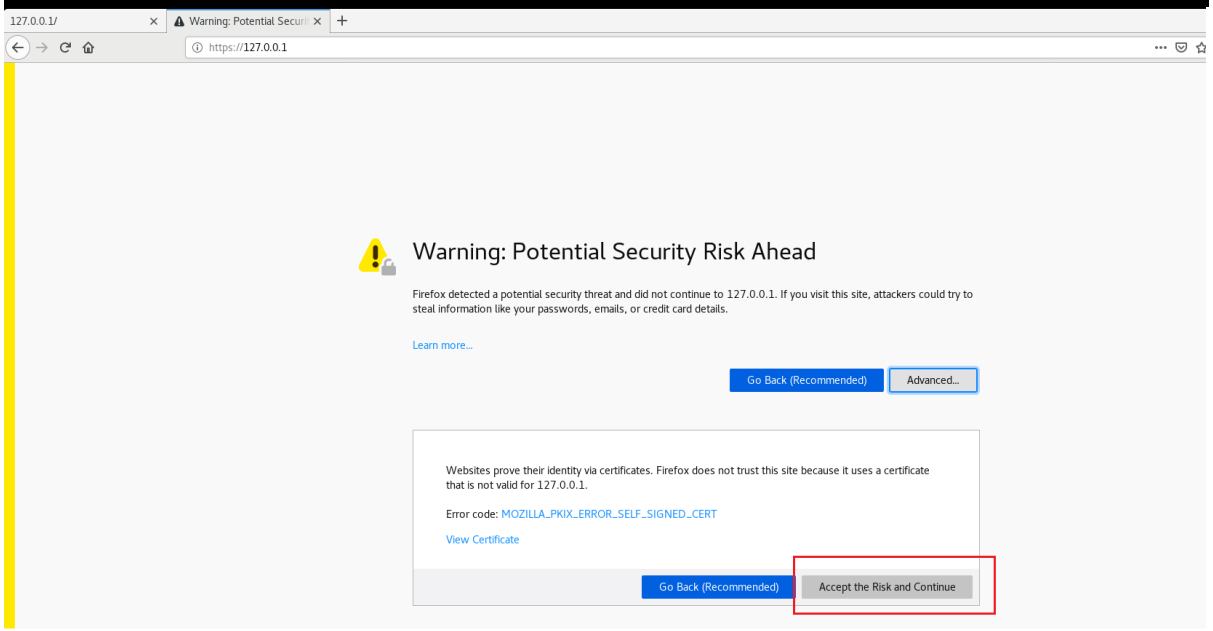
NGINX

Systemctl restart named

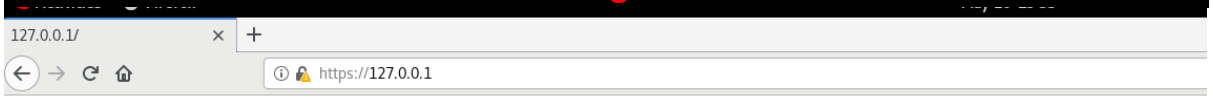
insert dns ip



Click to advance and check



Working done



SERVER 1

NGINX

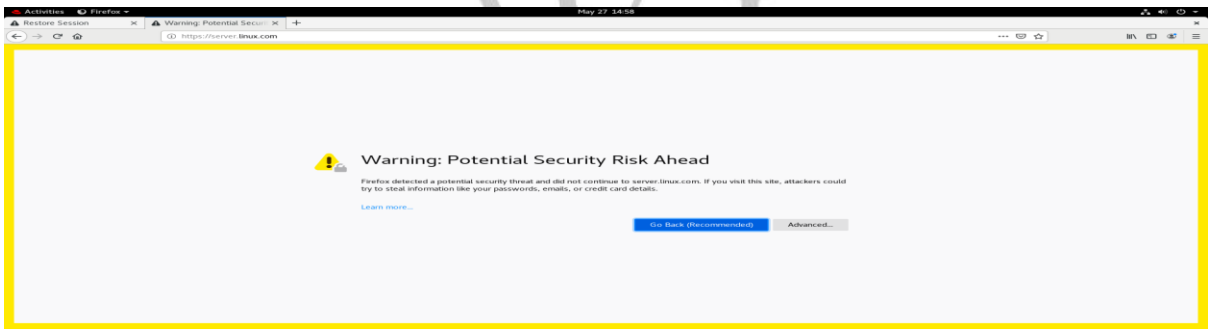
Use reverse proxy

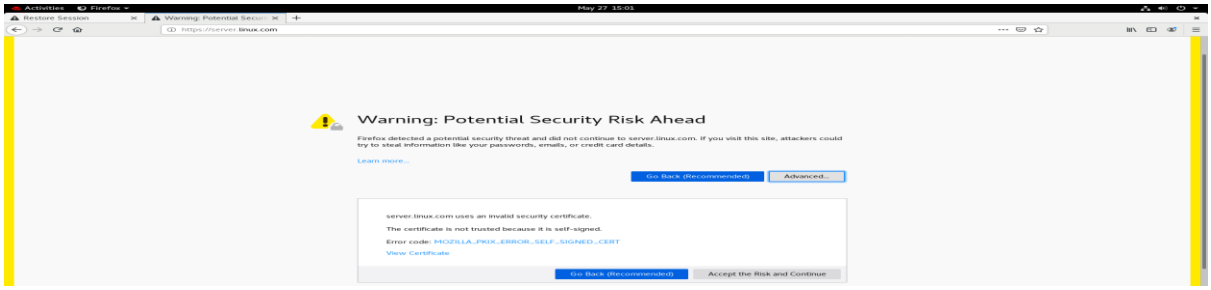
Edit file and add

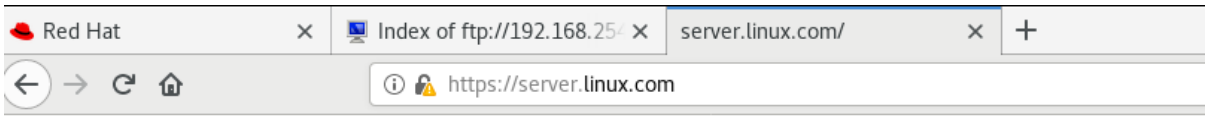
```
[root@server /]# vim /etc/nginx/conf.d/prxy_pass
server {
      listen 80;
      server_name linux.com server.linux.com;

 access_log  off;
 error_log off;

location / {
  proxy_pass              http://127.0.0.1:80/;
  proxy_set_header        X-Real-IP       $remote_addr;
  proxy_set_header        Host  $host;
  proxy_redirect          off;
  proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_connect_timeout   90;
  proxy_send_timeout      90;
  proxy_read_timeout      90;
  client_max_body_size    10m;
  client_body_buffer_size 128k;
  proxy_buffer_size       4k;
  proxy_buffers           4 32k;
  proxy_busy_buffers_size 64k;
```

Systemctl restart named

Both side done reverse







SERVER 1

CREATED BY:- VIVEK GANDHI