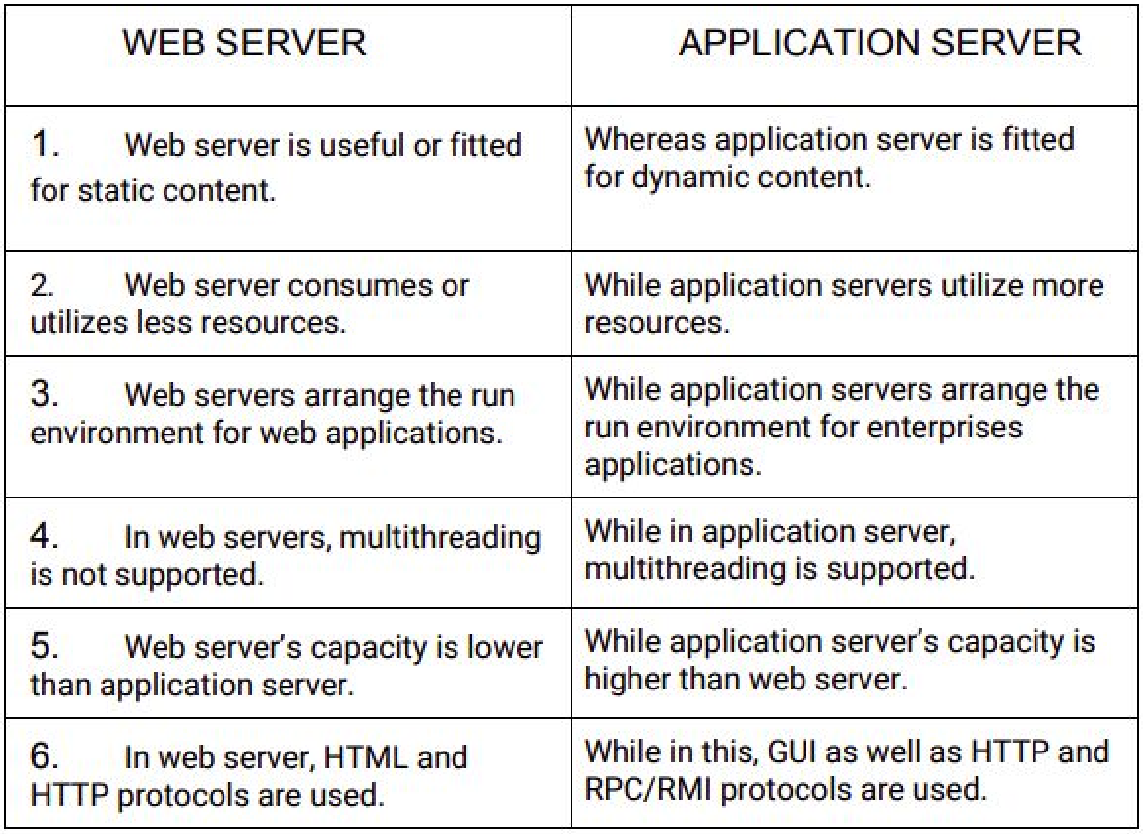
What is the difference between an Application Server and a Web Server?

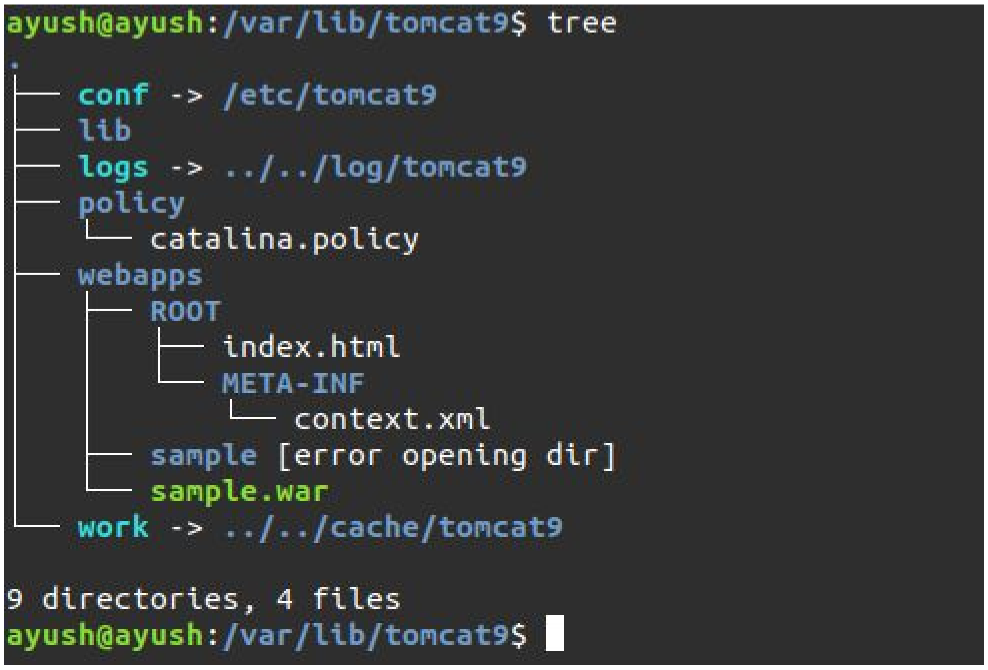


WEB SERVER	APPLICATION SERVER
1. Web server is useful or fitted for static content.	Whereas application server is fitted for dynamic content.
2. Web server consumes or utilizes less resources.	While application servers utilize more resources.
3. Web servers arrange the run environment for web applications.	While application servers arrange the run environment for enterprises applications.
4. In web servers, multithreading is not supported.	While in application server, multithreading is supported.
5. Web server's capacity is lower than application server.	While application server's capacity is higher than web server.
6. In web server, HTML and HTTP protocols are used.	While in this, GUI as well as HTTP and RPC/RMI protocols are used.

What is Catalina?

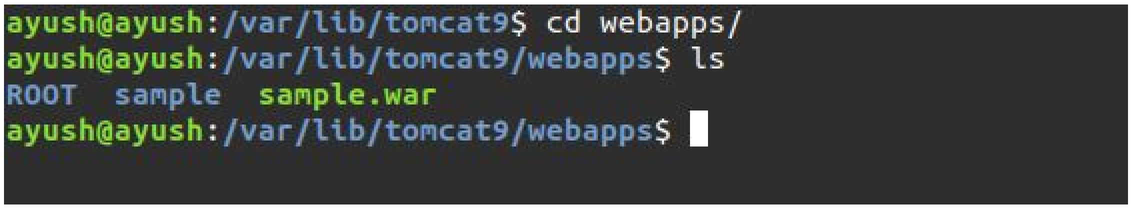
Catalina in Tomcat's servlet container. In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification

Describe tomcat directory structure.

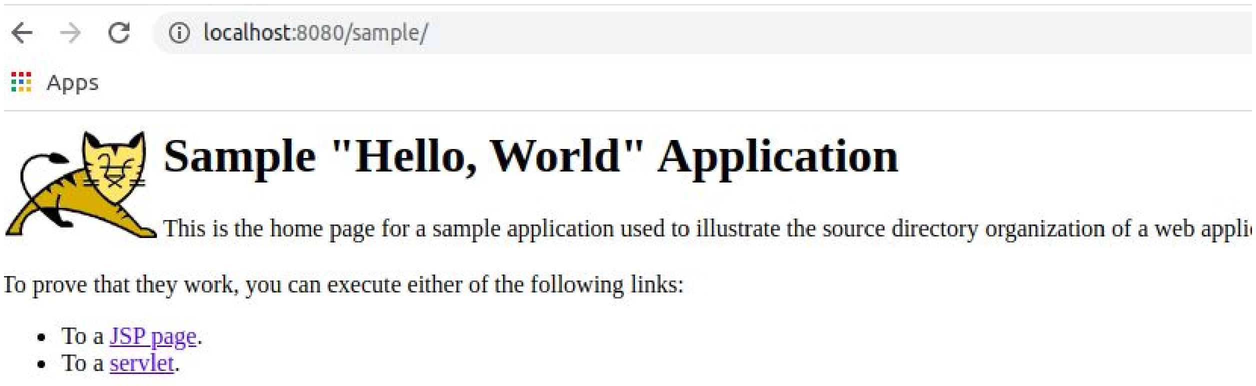


```
ayush@ayush:/var/lib/tomcat9$ tree
.
├── conf -> /etc/tomcat9
├── lib
├── logs -> ../../log/tomcat9
├── policy
│   └── catalina.policy
├── webapps
│   ├── ROOT
│   │   ├── index.html
│   │   └── META-INF
│   │       └── context.xml
│   ├── sample [error opening dir]
│   └── sample.war
└── work -> ../../cache/tomcat9

9 directories, 4 files
ayush@ayush:/var/lib/tomcat9$
```

Connect any sample.war to MySQL running on localhost.



```
ayush@ayush:/var/lib/tomcat9$ cd webapps/
ayush@ayush:/var/lib/tomcat9/webapps$ ls
ROOT  sample  sample.war
ayush@ayush:/var/lib/tomcat9/webapps$
```



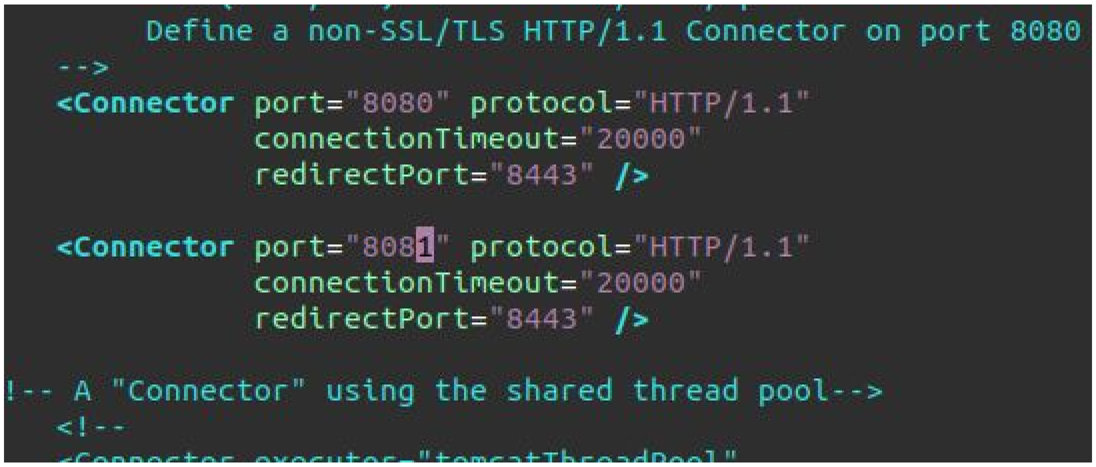
Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web appli

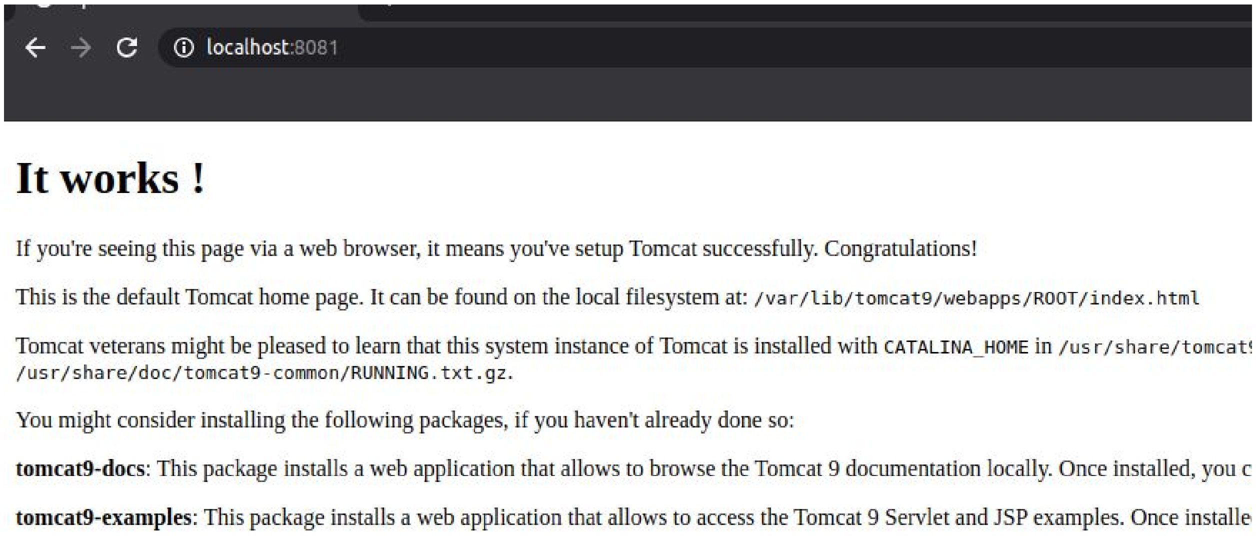
To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

Run multiple services on different ports with different connectors (AJP/HTTP) on the same tomcat installation.



```
         Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />

    <Connector port="8081" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />

!-- A "Connector" using the shared thread pool-->
    <!--
```



localhost:8081

It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

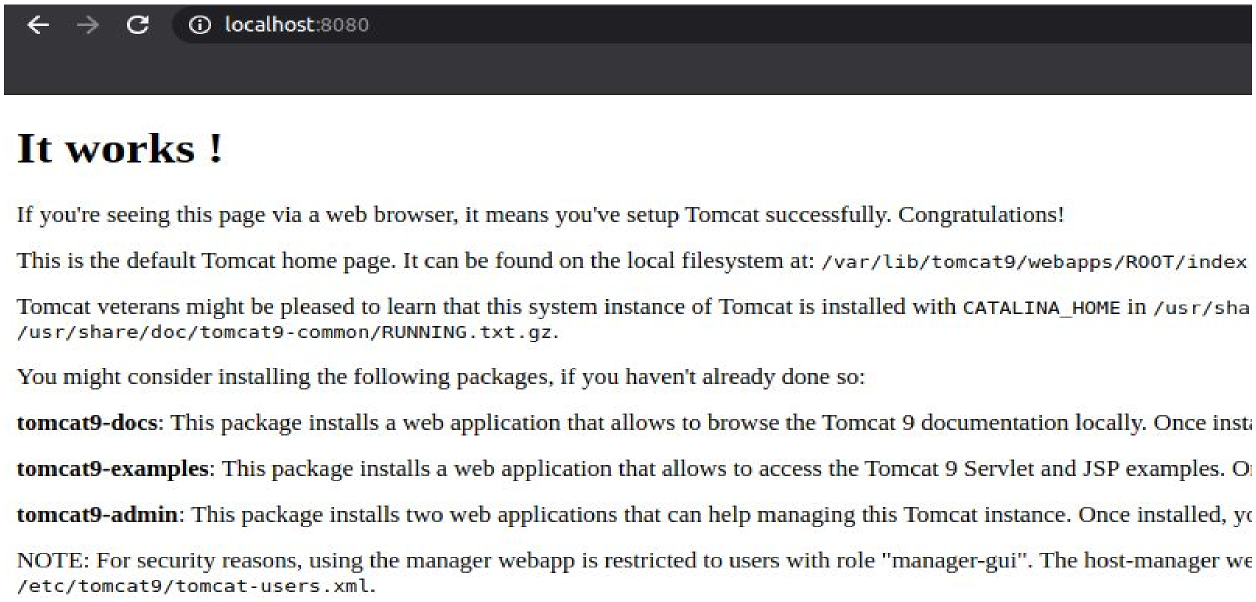
This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat`
`/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you c

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installe



localhost:8080

It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/sha`
`/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

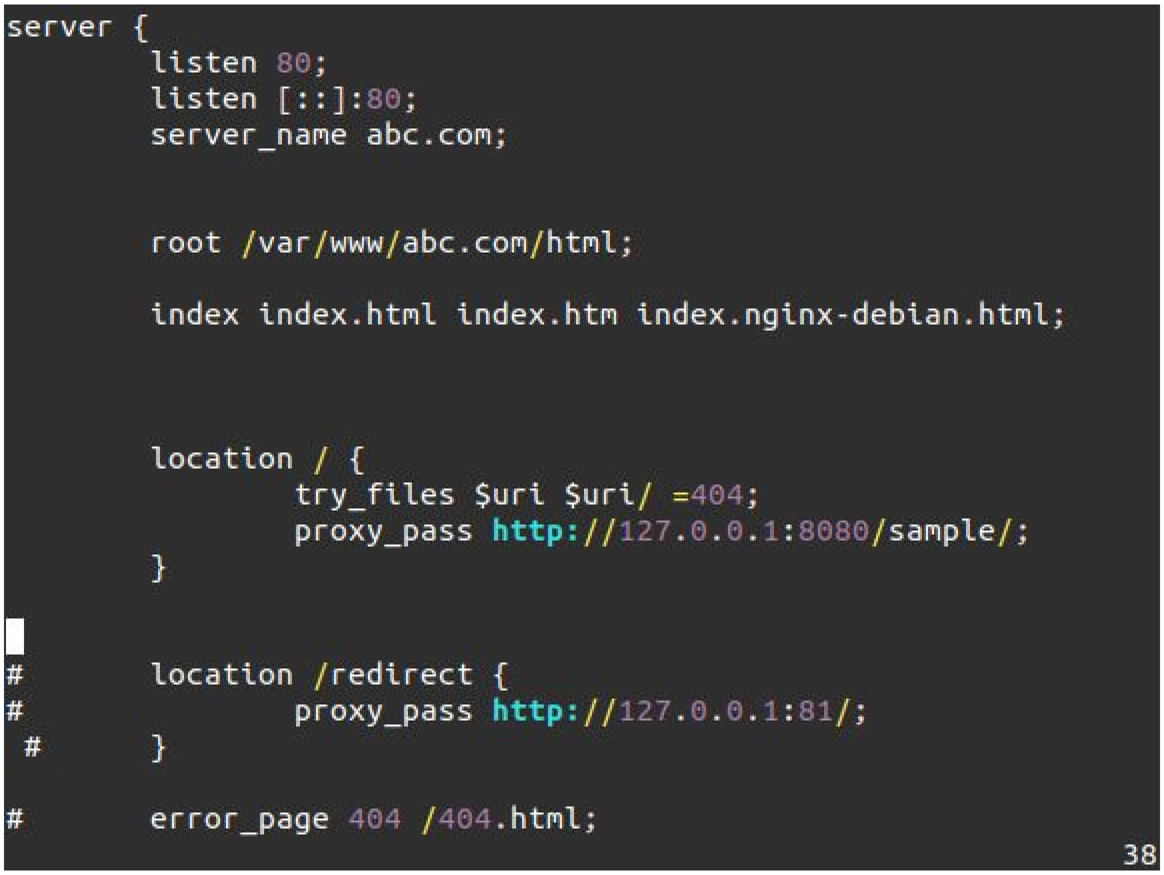
tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once inst

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. O

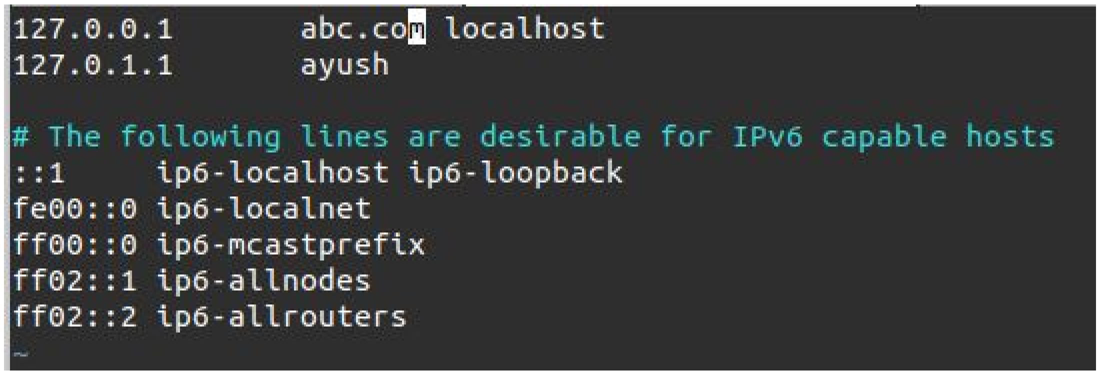
tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, y

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager we
`/etc/tomcat9/tomcat-users.xml`.

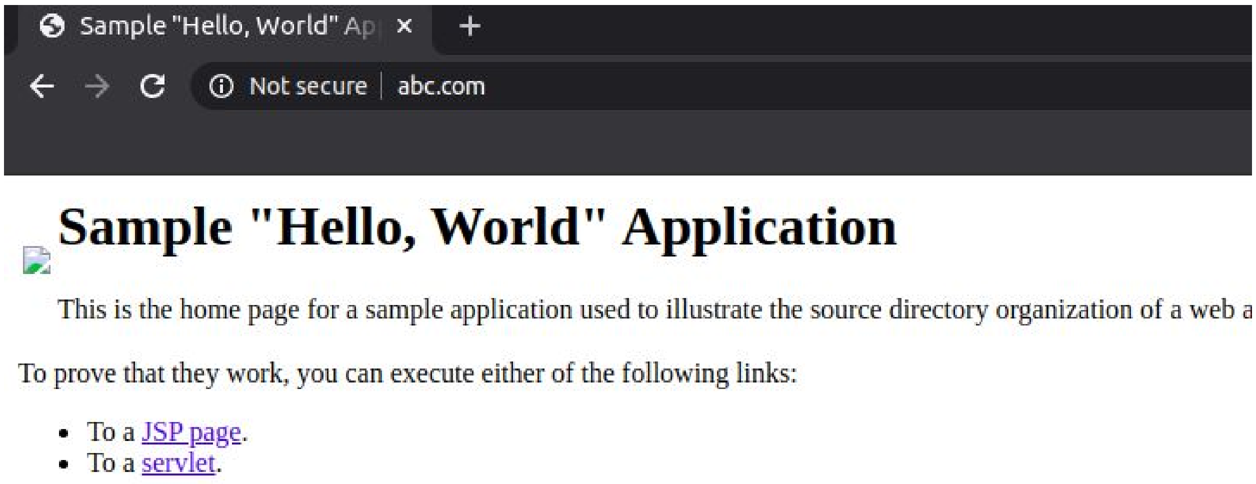
Use nginx as a reverse proxy for tomcat application.



```
server {
        listen 80;
        listen [::]:80;
        server_name abc.com;


        root /var/www/abc.com/html;

        index index.html index.htm index.nginx-debian.html;



        location / {
                try_files $uri $uri/ =404;
                proxy_pass http://127.0.0.1:8080/sample/;
        }


#       location /redirect {
#               proxy_pass http://127.0.0.1:81/;
 #      }

#       error_page 404 /404.html;
                                                                38
```



```
127.0.0.1       abc.com localhost
127.0.1.1       ayush

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
```



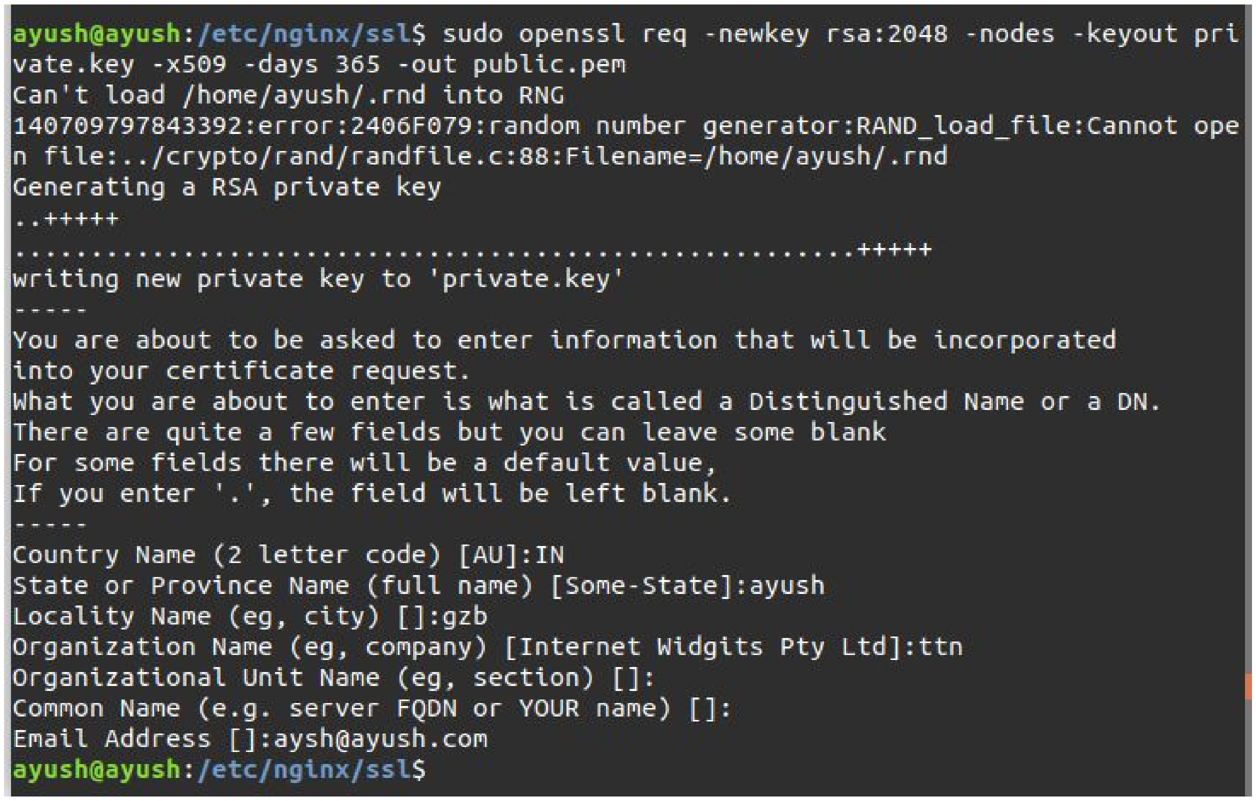
Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web a

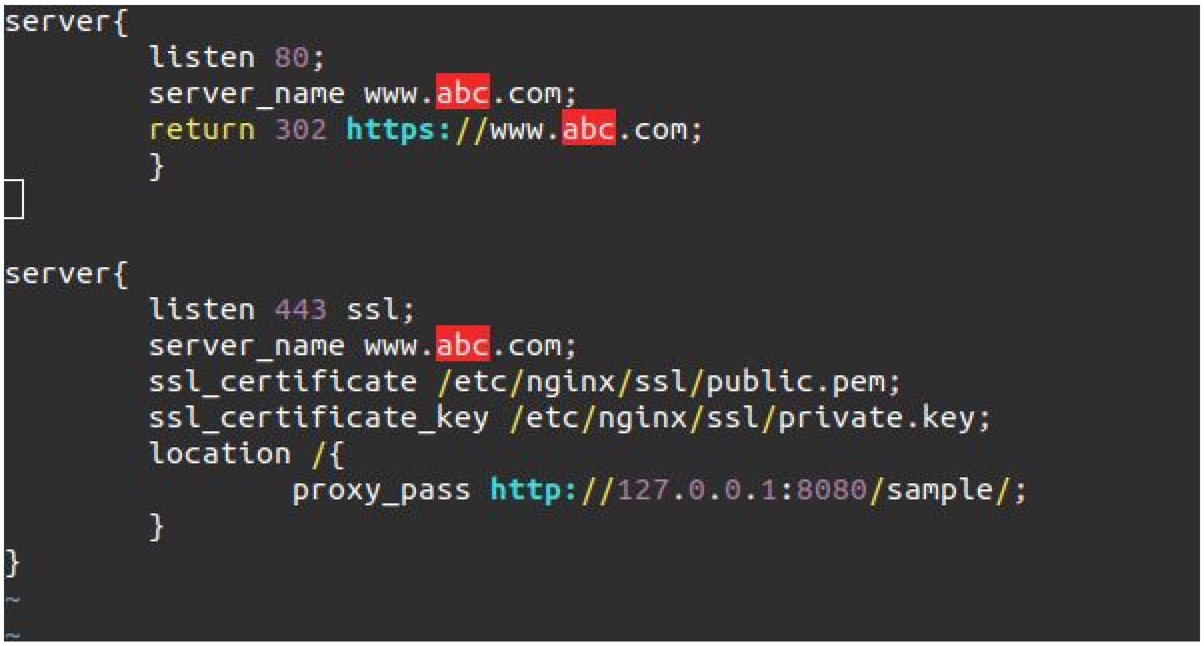
To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

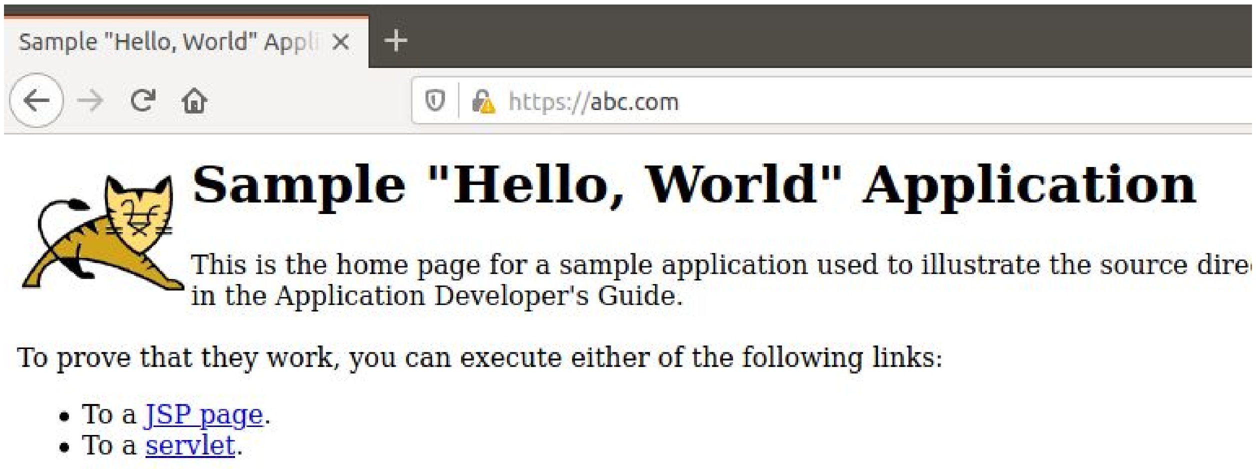
Setup self signed certificate on that nginx for bootcamp.com.



```
ayush@ayush:/etc/nginx/ssl$ sudo openssl req -newkey rsa:2048 -nodes -keyout pri
vate.key -x509 -days 365 -out public.pem
Can't load /home/ayush/.rnd into RNG
140709797843392:error:2406F079:random number generator:RAND_load_file:Cannot ope
n file:../crypto/rand/randfile.c:88:Filename=/home/ayush/.rnd
Generating a RSA private key
..+++++
.......................................................+++++
writing new private key to 'private.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) [AU]:IN
State or Province Name (full name) [Some-State]:ayush
Locality Name (eg, city) []:gzb
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ttn
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:aysh@ayush.com
ayush@ayush:/etc/nginx/ssl$
```



```
server{
        listen 80;
        server_name www.abc.com;
        return 302 https://www.abc.com;
        }


server{
        listen 443 ssl;
        server_name www.abc.com;
        ssl_certificate /etc/nginx/ssl/public.pem;
        ssl_certificate_key /etc/nginx/ssl/private.key;
        location /{
                proxy_pass http://127.0.0.1:8080/sample/;
        }
}
~
~
```



Sample "Hello, World" Appli

https://abc.com

Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source dire
in the Application Developer's Guide.

To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

What is the difference between proxy_pass & proxy_pass reverse?

In case of forward proxy(i.e proxy_pass) client's identity is hidden eg. in VPN.
In case of reverse proxy server's identity is hidden for eg when we hit nginx tomcat's content is served.

For this example, I will list three computers connected to the internet.

- X = your computer, or "client" computer on the internet
- Y = the reverse proxy web site, proxy.example.com
- Z = the web site you want to visit, www.example.net

Normally, one would connect directly from `X --> Z.`

However, in some scenarios, it is better for the administrator of `Z` to restrict or disallow direct access and force visitors to go through Y first. So, as before, we have data being retrieved by `Y --> Z` on behalf of `X`, which chains as follows: `X --> Y --> Z`.

What is different this time compared to a "forward proxy," is that this time the user `X` does not know he is accessing `Z`, because the user `X` only sees he is communicating with `Y`. The server `Z` is invisible to clients and only the reverse proxy `Y` is visible externally. A reverse proxy requires no (proxy) configuration on the client side.

The client `X` thinks he is only communicating with `Y` (`X --> Y`), but the reality is that `Y` forwarding all communication (`X --> Y --> Z` again)