

Assignment-1 Report

- Download and install nginx. Then run using below commands:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

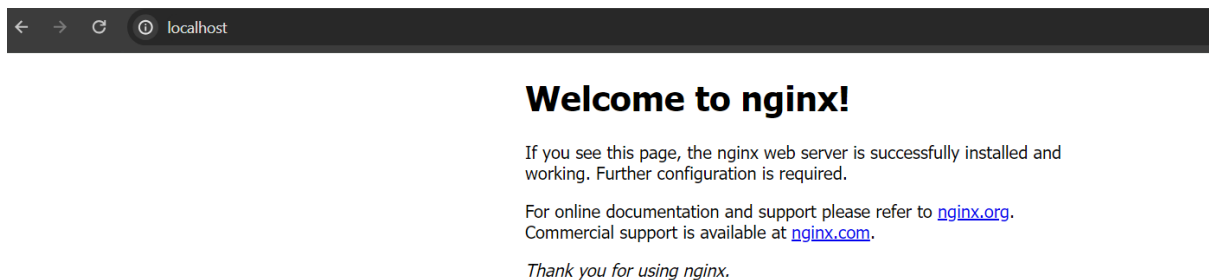
C:\Windows\System32>cd C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2

C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>start nginx

C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>start nginx.exe

C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>
```

- Visit localhost from browser to check if run was successful:



- Create a web folder in nodejs folder. Create four webserver.js files with different ports:

```
const http = require("http")
const hostname = '127.0.0.1';
const port = 2321;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World from Server One');
})

server.listen(port, hostname, () => {
  console.log(`Server is running at http://${hostname}:${port}/`);
})
```

```
const http = require("http")
const hostname = '127.0.0.1';
const port = 2322;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World from Server Two');
})

server.listen(port, hostname, () => {
  console.log(`Server is running at http://${hostname}:${port}/`);
})
```

```
const http = require("http")
const hostname = '127.0.0.1';
const port = 2323;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World from Server Three');
})

server.listen(port, hostname, () => {
  console.log(`Server is running at http://${hostname}:${port}/`);
})
```

```
const http = require("http")
const hostname = '127.0.0.1';
const port = 2324;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World from Server Four');
})

server.listen(port, hostname, () => {
  console.log(`Server is running at http://${hostname}:${port}/`);
})
```

- Start each server by navigating to web folder and using below commands:

```
Administrator: Command Prompt - node webserver1.js
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Program Files\nodejs\web

C:\Program Files\nodejs\web>node webserver1.js
Server is running at http://127.0.0.1:2321/
```

```
Administrator: Command Prompt - node webserver2.js
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Program Files\nodejs\web

C:\Program Files\nodejs\web>node webserver2.js
Server is running at http://127.0.0.1:2322/
```

```
Administrator: Command Prompt - node webserver3.js
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Program Files\nodejs\web

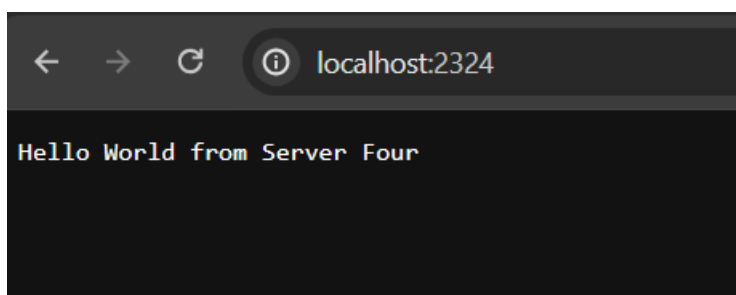
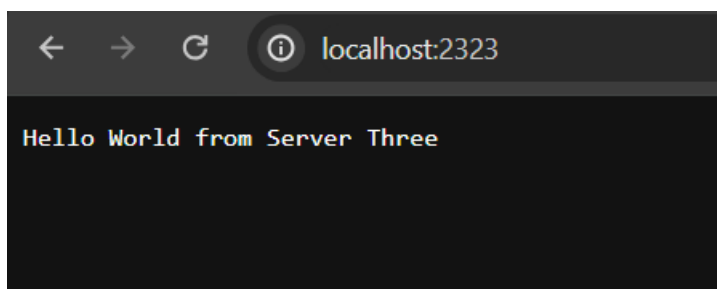
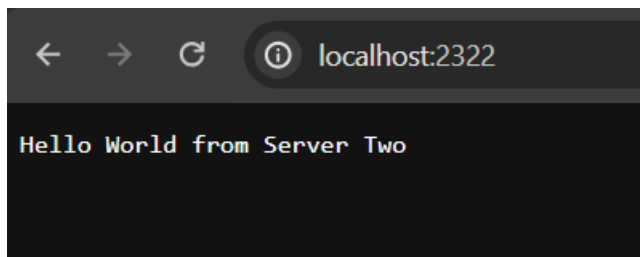
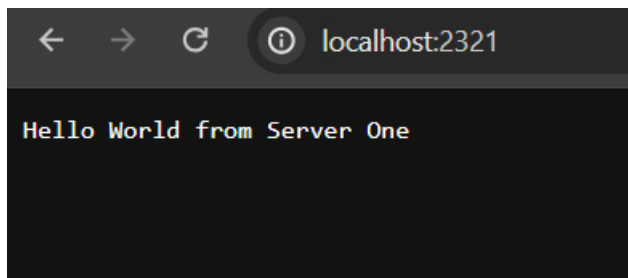
C:\Program Files\nodejs\web>node webserver3.js
Server is running at http://127.0.0.1:2323/
```

```
Administrator: Command Prompt - node webserver4.js
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Program Files\nodejs\web

C:\Program Files\nodejs\web>node webserver4.js
Server is running at http://127.0.0.1:2324/
```

- Test each server from browser as below:



- Modify the nginx configuration file. Open configuration file nginx.conf. Change the port number to an unused one. Configure the server cluster set to reverse proxy pass as below:

```
server {  
    listen      2777;  
    server_name localhost;
```

```

upstream myservers{
    random;
    server 127.0.0.1:2321;
    server 127.0.0.1:2322;
    server 127.0.0.1:2323;
    server 127.0.0.1:2324;
}

server {
    listen      2777;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        index   index.html index.htm;
        proxy_pass http://myservers;
    }
}

```

- Restart nginx and reset nginx on the command line with the nginx -s reload command:

```

C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>cd C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2
C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>nginx -s reload
C:\Users\charu\Downloads\nginx-1.20.2\nginx-1.20.2>

```

- Test by sending http request to the server using command : **curl localhost:2777**. Below are different load balancing strategies:

1. Round-Robin:

```

upstream myservers{
    server 127.0.0.1:2321 weight=1;
    server 127.0.0.1:2322 weight=1;
    server 127.0.0.1:2323 weight=1;
    server 127.0.0.1:2324 weight=1;
}

```

```
C:\Users\charu>curl localhost:2777
Hello World from Server One
C:\Users\charu>curl localhost:2777
Hello World from Server Two
C:\Users\charu>curl localhost:2777
Hello World from Server Three
C:\Users\charu>curl localhost:2777
Hello World from Server Four
C:\Users\charu>
```

2. Weighted Round-Robin:

```
upstream myservers{
    server 127.0.0.1:2321 weight=30
    server 127.0.0.1:2322 weight=10;
    server 127.0.0.1:2323 weight=5;
    server 127.0.0.1:2324 weight=5;
}
```

```
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server Two
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server Four
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server Two
C:\Windows\System32>curl localhost:2777
Hello World from Server One
```

3. Least Connections

```
upstream myservers{
least_conn;
server 127.0.0.1:2321;
server 127.0.0.1:2322;
server 127.0.0.1:2323;
server 127.0.0.1:2324;
}
```

```
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server Two
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Four
```

4. IP-Hash:

```
upstream myservers{
ip_hash;
server 127.0.0.1:2321;
server 127.0.0.1:2322;
server 127.0.0.1:2323;
server 127.0.0.1:2324;
}
```

```
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server One
C:\Windows\System32>curl localhost:2777
Hello World from Server One
```

Using down to temporarily remove a server from the rotation:

```
upstream myservers{
ip_hash;
server 127.0.0.1:2321 down;
server 127.0.0.1:2322;
server 127.0.0.1:2323;
server 127.0.0.1:2324;
}
```

```
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
```

5. Generic-Hash(Using request_uri):

```
upstream myservers{
    hash $request_uri consistent;
    server 127.0.0.1:2321;
    server 127.0.0.1:2322;
    server 127.0.0.1:2323;
    server 127.0.0.1:2324;
}
```

```
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
```

6. Random:

```
upstream myservers{
    random;
    server 127.0.0.1:2321;
    server 127.0.0.1:2322;
    server 127.0.0.1:2323;
    server 127.0.0.1:2324;
}
```



```
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Two
C:\Windows\System32>curl localhost:2777
Hello World from Server Four
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
C:\Windows\System32>curl localhost:2777
Hello World from Server Four
C:\Windows\System32>curl localhost:2777
Hello World from Server Three
```