Networking Advanced

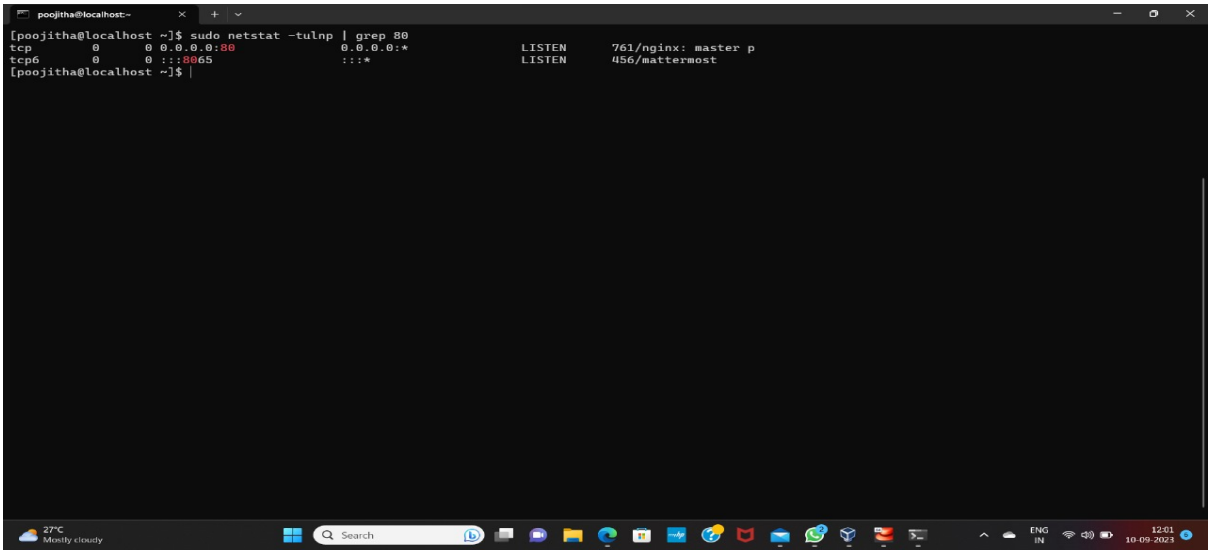
Pre-Requests:

1. Have two Virtual Box machines. Lets say VM1 and VM2

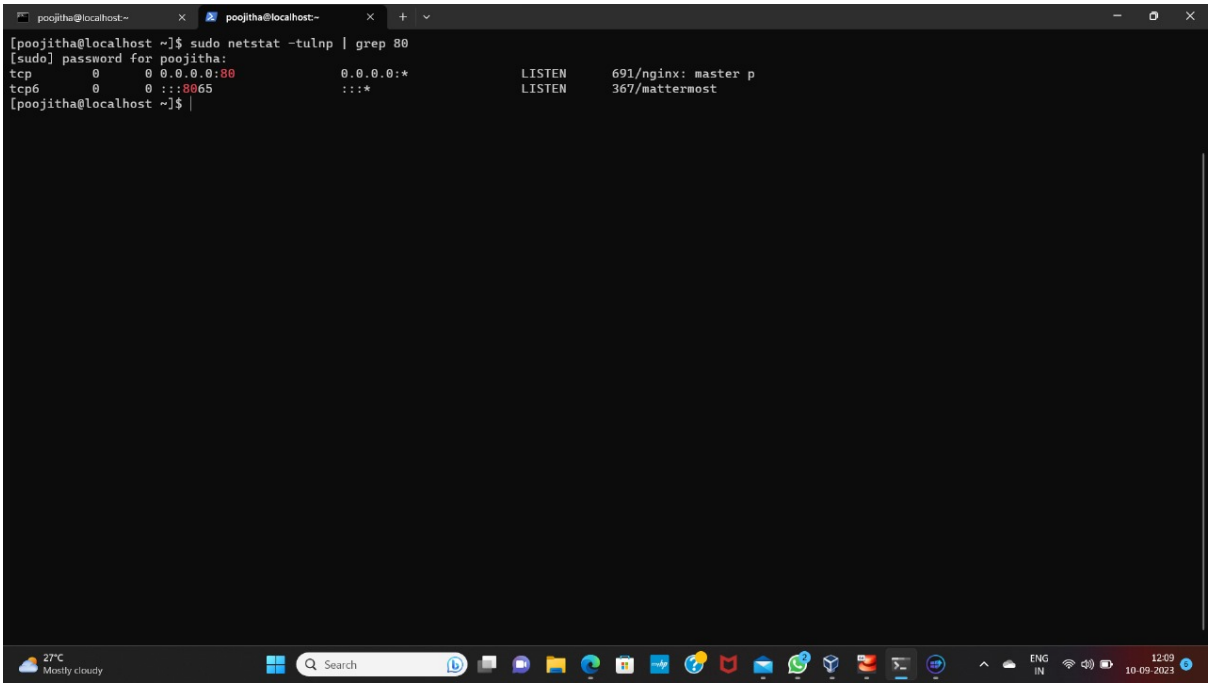
2. Run Nginx on 80 port in both the machines.

3. Run Mattermost on 8065 port in both the machines

A: vm1

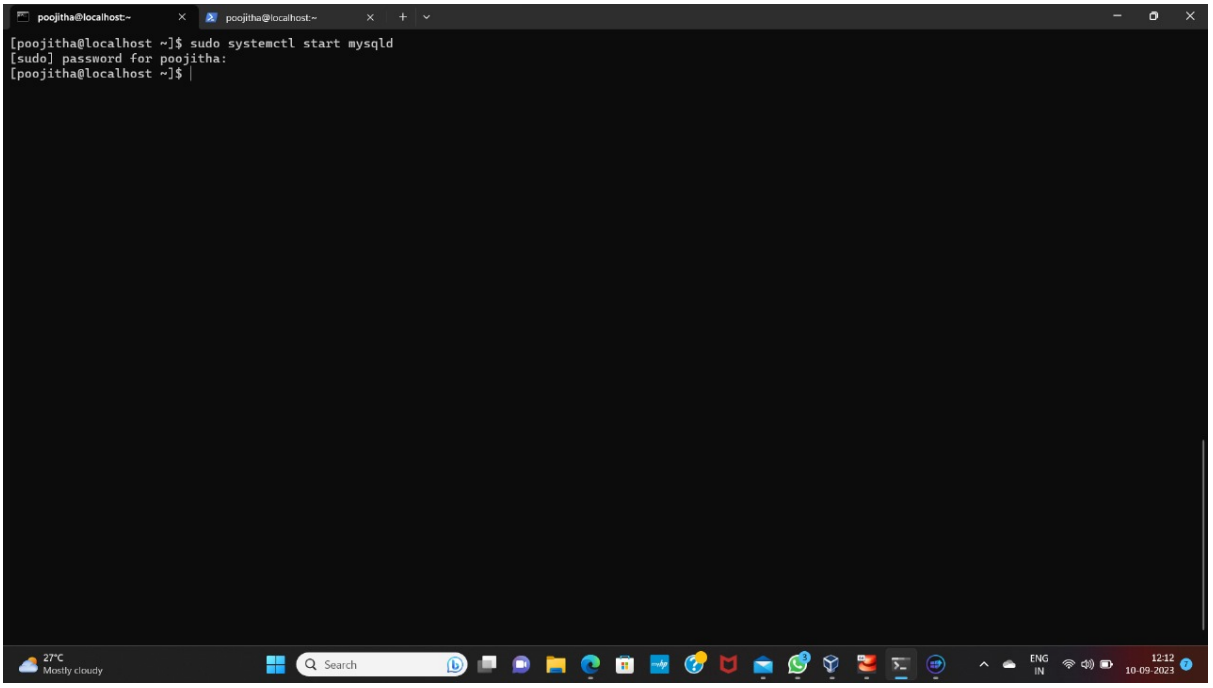


Vm 2



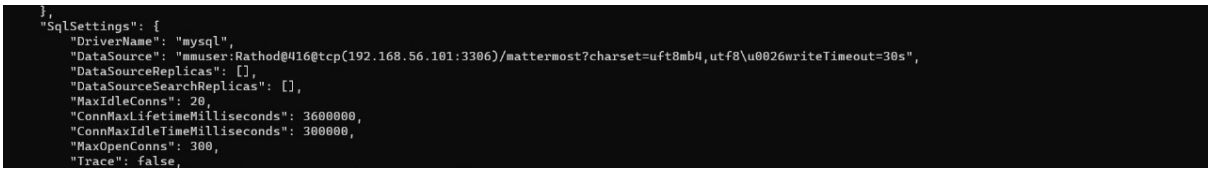
4. Run MySQL only in first machine (VM1)

A:



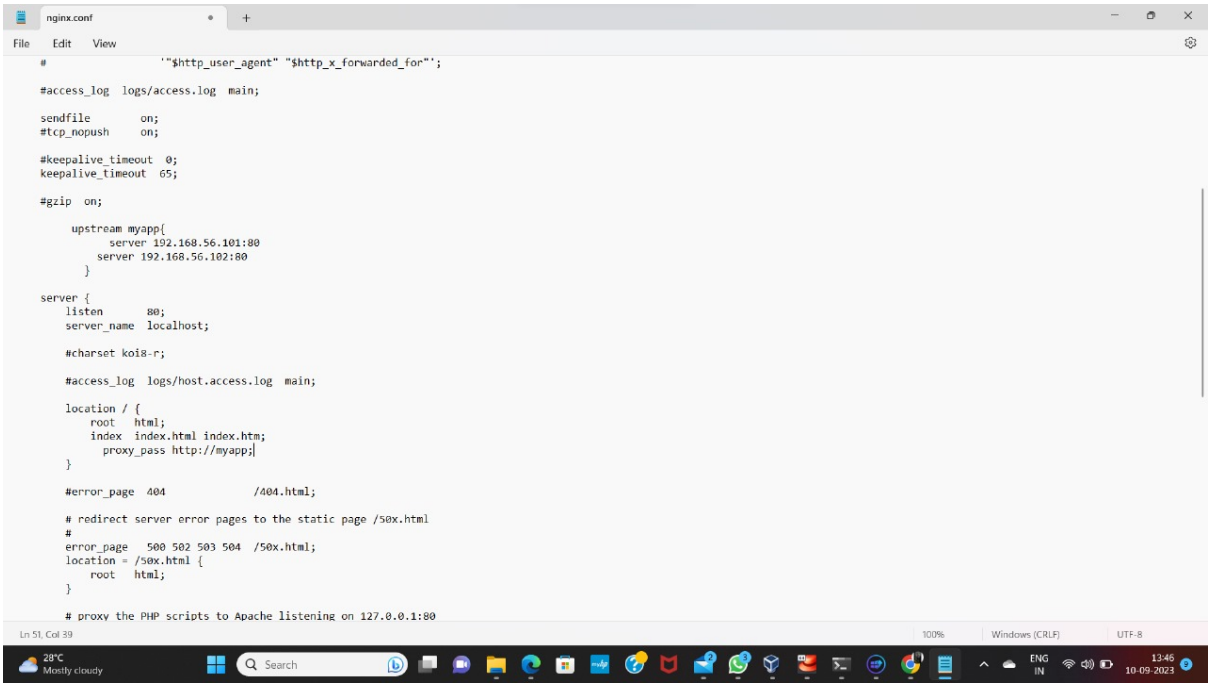
5. Change MatterMost config in VM2 to point to VM1 MySQL

A:



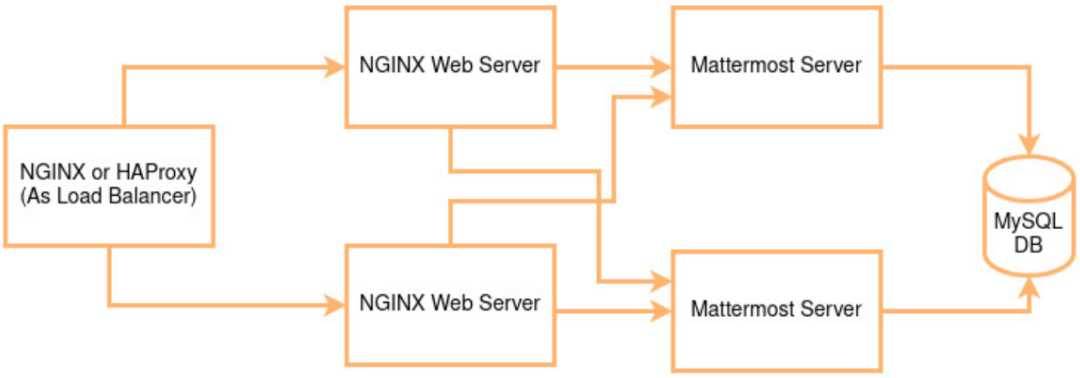
6. Install nginx server in your windows and add proxy config to point to both VM1 and VM2 NGINX

A:

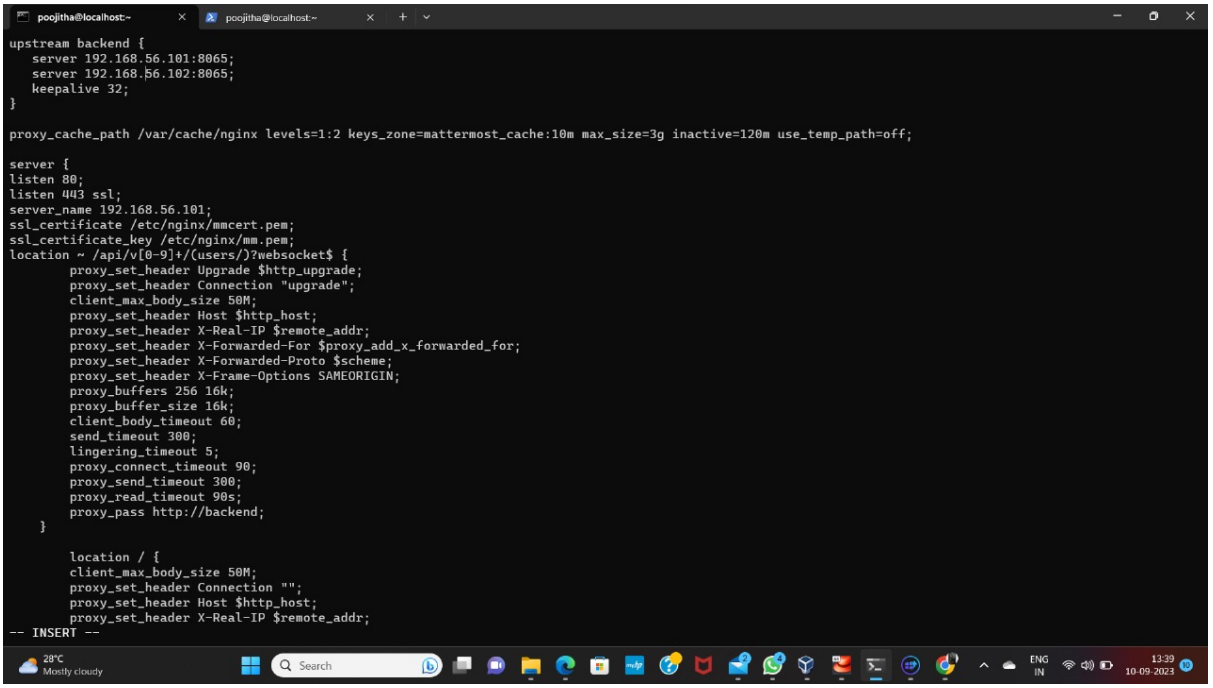


7. Refer Attached Logical Architecture diagram Steps:

1. confiugre nginx.conf file as attached in the doc

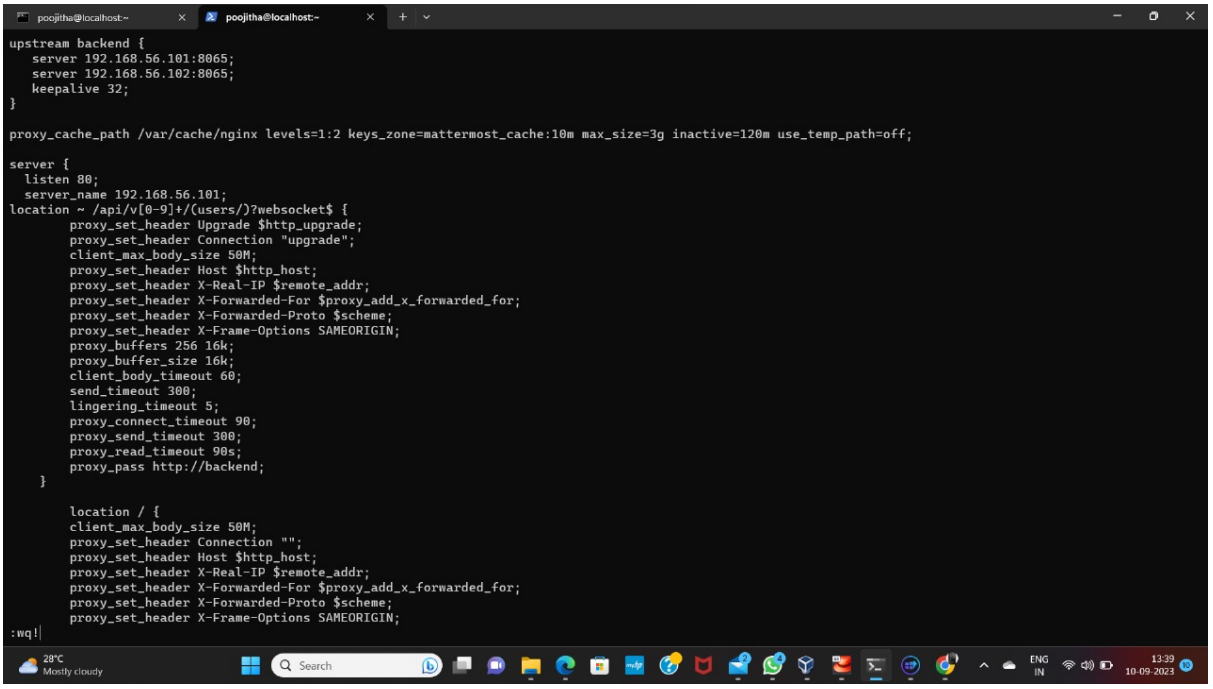


Pointing vm 1 nginx to vm 2 mattermost



```
upstream backend {
   server 192.168.56.101:8065;
   server 192.168.56.102:8065;
   keepalive 32;
}

proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=mattermost_cache:10m max_size=3g inactive=120m use_temp_path=off;

server {
listen 80;
listen 443 ssl;
server_name 192.168.56.101;
ssl_certificate /etc/nginx/mmcert.pem;
ssl_certificate_key /etc/nginx/mm.pem;
location ~ /api/v[0-9]+/(users/)?websocket$ {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        client_max_body_size 50M;
        proxy_set_header Host $http_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;
        proxy_set_header X-Frame-Options SAMEORIGIN;
        proxy_buffers 256 16k;
        proxy_buffer_size 16k;
        client_body_timeout 60;
        send_timeout 300;
        lingering_timeout 5;
        proxy_connect_timeout 90;
        proxy_send_timeout 300;
        proxy_read_timeout 90s;
        proxy_pass http://backend;
    }

        location / {
        client_max_body_size 50M;
        proxy_set_header Connection "";
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
-- INSERT --
```

Pointing vm 2 nginx to vm 1 mattermost



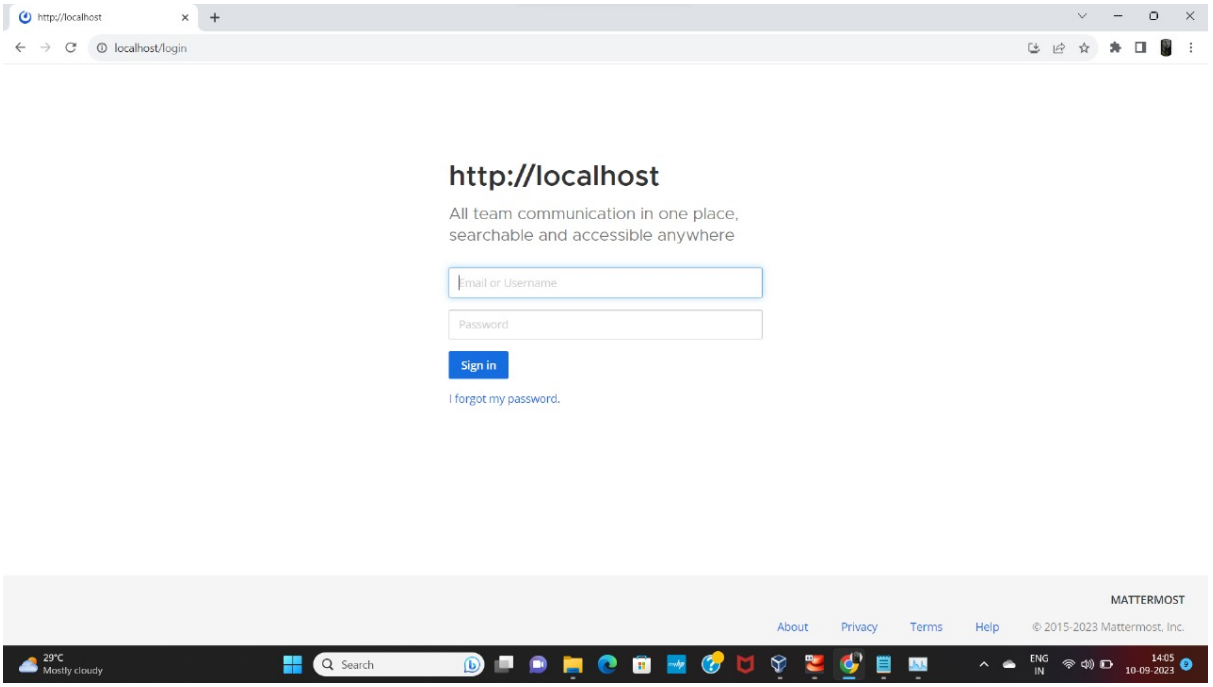
```
upstream backend {
   server 192.168.56.101:8065;
   server 192.168.56.102:8065;
   keepalive 32;
}

proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=mattermost_cache:10m max_size=3g inactive=120m use_temp_path=off;

server {
  listen 80;
  server_name 192.168.56.101;
location ~ /api/v[0-9]+/(users/)?websocket$ {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        client_max_body_size 50M;
        proxy_set_header Host $http_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;
        proxy_set_header X-Frame-Options SAMEORIGIN;
        proxy_buffers 256 16k;
        proxy_buffer_size 16k;
        client_body_timeout 60;
        send_timeout 300;
        lingering_timeout 5;
        proxy_connect_timeout 90;
        proxy_send_timeout 300;
        proxy_read_timeout 90s;
        proxy_pass http://backend;
    }

        location / {
        client_max_body_size 50M;
        proxy_set_header Connection "";
        proxy_set_header Host $http_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;
        proxy_set_header X-Frame-Options SAMEORIGIN;
:wq!
```

Run localhost url in your browser for the following scenarios a-> server 1 up and server 2 down, server 1 down server 2 is up, both serv

A: SCENARIO 1

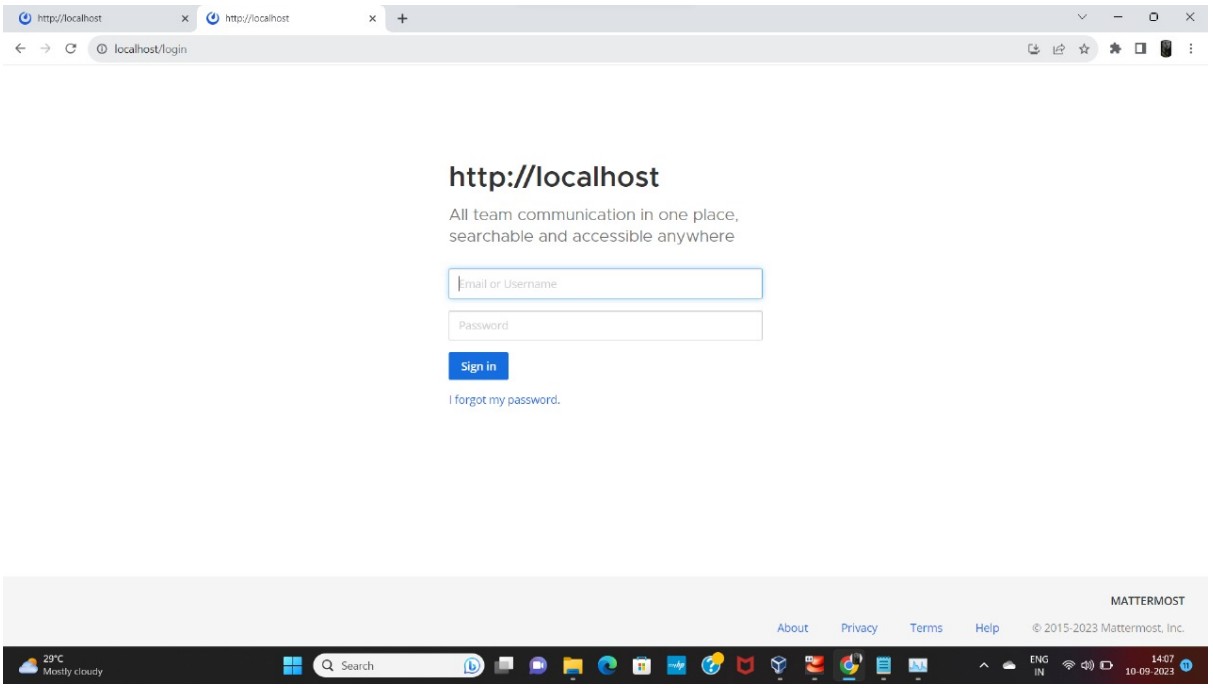
Vm 1 up and vm 2 down

Output



SCENARIO 2: vm 2 up and vm 1 down

Output:



SCENARIO 3: both nginx are down

Output: