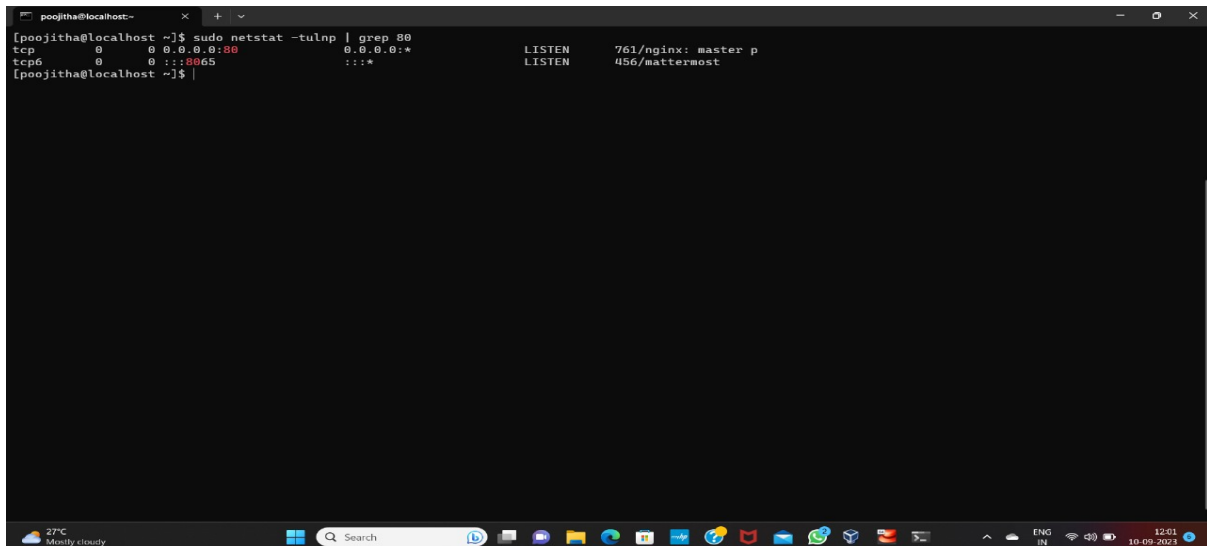


Networking Advanced

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

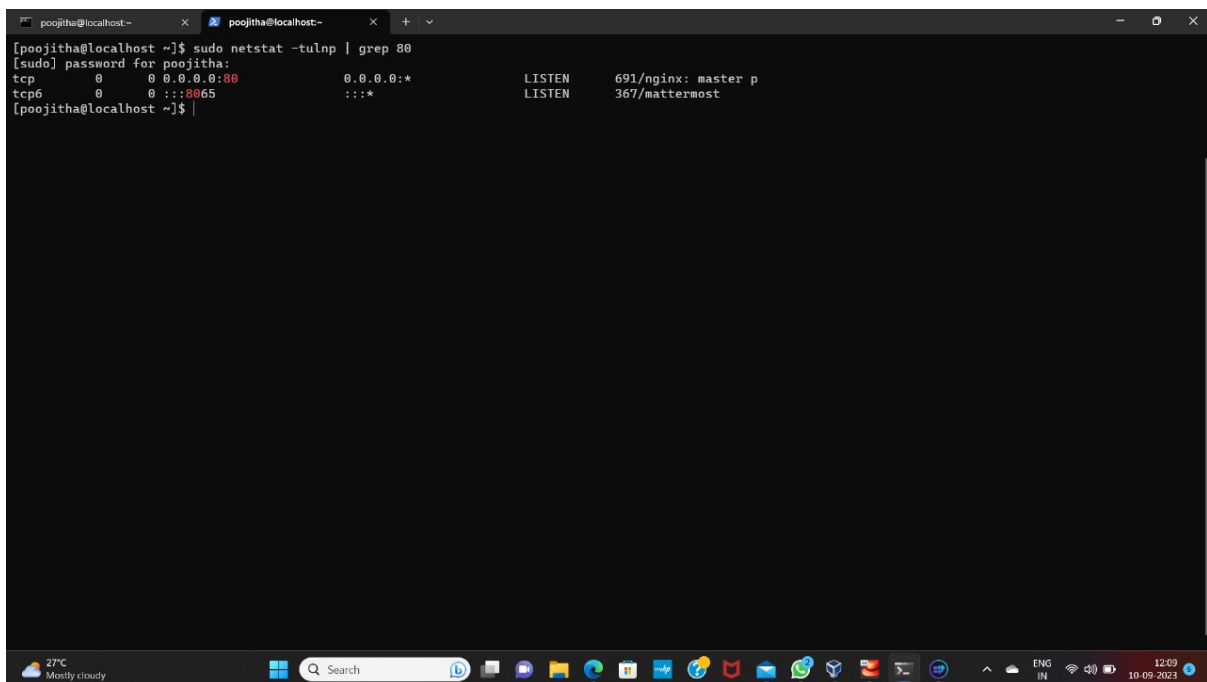
A: vm1



```
poojitha@localhost:~$ sudo netstat -tulnp | grep 80
tcp        0      0 0.0.0.0:80        0.0.0.0:*        LISTEN     761/nginx: master p
tcp6       0      0 :::8065          :::*              LISTEN     456/mattermost
poojitha@localhost:~$
```

The screenshot shows a terminal window titled 'poojitha@localhost:~'. The user has executed the command 'sudo netstat -tulnp | grep 80'. The output shows two listening ports: '0.0.0.0:80' for 'nginx: master p' (PID 761) and ':::8065' for 'mattermost' (PID 456). The terminal window is part of a desktop environment with a taskbar at the bottom showing various application icons and system status information like '27°C Mostly cloudy' and '10-09-2023 12:01'.

Vm 2



```
poojitha@localhost:~$ sudo netstat -tulnp | grep 80
[sudo] password for poojitha:
tcp        0      0 0.0.0.0:80        0.0.0.0:*        LISTEN     691/nginx: master p
tcp6       0      0 :::8065          :::*              LISTEN     367/mattermost
poojitha@localhost:~$
```

The screenshot shows a terminal window titled 'poojitha@localhost:~'. The user has executed the command 'sudo netstat -tulnp | grep 80'. The output shows two listening ports: '0.0.0.0:80' for 'nginx: master p' (PID 691) and ':::8065' for 'mattermost' (PID 367). The terminal window is part of a desktop environment with a taskbar at the bottom showing various application icons and system status information like '27°C Mostly cloudy' and '10-09-2023 12:09'.

4. Run MySQL only in first machine (VM1)

A:

```
poojitha@localhost:~$ sudo systemctl start mysqld
[sudo] password for poojitha:
[poojitha@localhost ~]$
```

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

A:

```
},
"SqlSettings": {
  "DriverName": "mysql",
  "DataSource": "mmuser:Rathod@416@tcp(192.168.56.101:3306)/mattermost?charset=utf8mb4,utf8\u0026writeTimeout=30s",
  "DataSourceReplicas": [],
  "DataSourceSearchReplicas": [],
  "MaxIdleConns": 20,
  "ConnMaxLifetimeMilliseconds": 3600000,
  "ConnMaxIdleTimeMilliseconds": 300000,
  "MaxOpenConns": 300,
  "Trace": false,
}
```

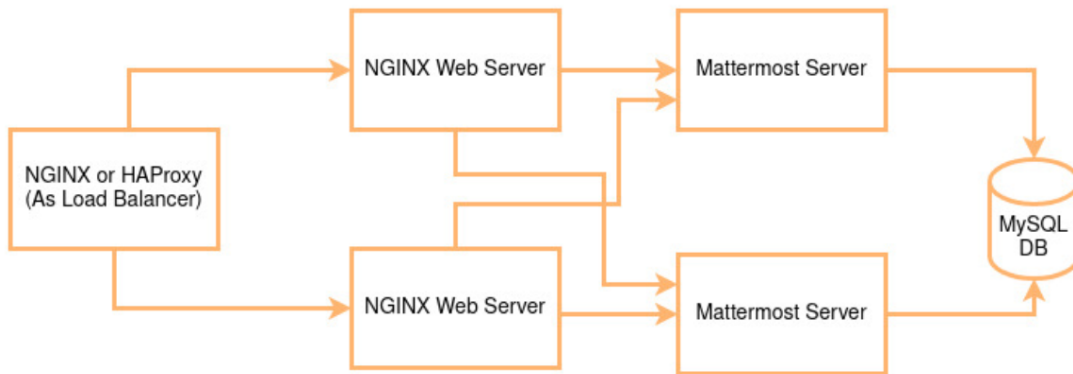
6. Install nginx server in your windows and add proxy config to point to both VM1 and VM2 NGINX

A:

```
nginx.conf
File Edit View
#
# "http_user_agent" "http_x_forwarded_for";
#
#access_log logs/access.log main;
#
sendfile on;
#tcp_nopush on;
#
#keepalive_timeout 0;
keepalive_timeout 65;
#
#gzip on;
#
upstream myapp{
    server 192.168.56.101:80
    server 192.168.56.102:80
}
server {
    listen 80;
    server_name localhost;
#charset kois-r;
#access_log logs/host.access.log main;
    location / {
        root html;
        index index.html index.htm;
        proxy_pass http://myapp;
    }
    #error_page 404 /404.html;
# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}
# proxy the PHP scripts to Apache listening on 127.0.0.1:80
Ln 51, Col 39 100% Windows (CRLF) UTF-8 13:45 10-09-2023
```

7. Refer Attached Logical Architecture diagram Steps:

1. configu gre nginx.conf file as attached in the doc



Pointing vm 1 nginx to vm 2 mattermost

```
poojitha@localhost:~$ cat /etc/nginx/nginx.conf
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.key;
    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;
    }

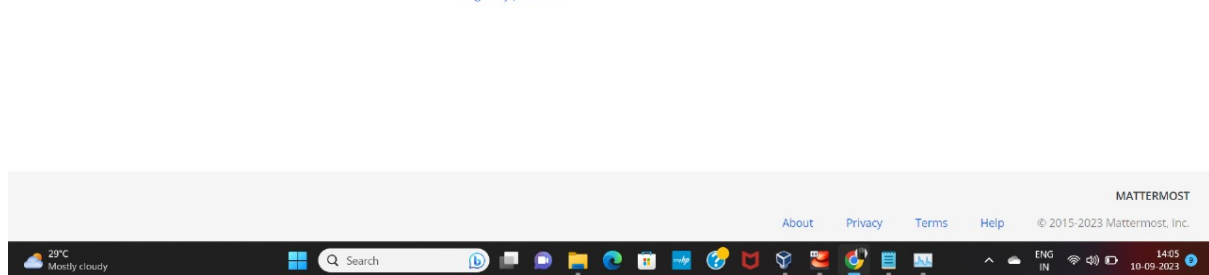
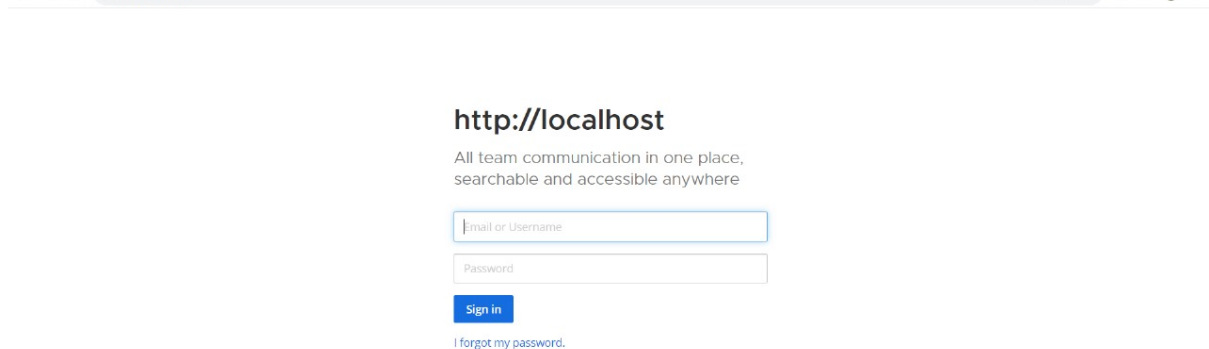
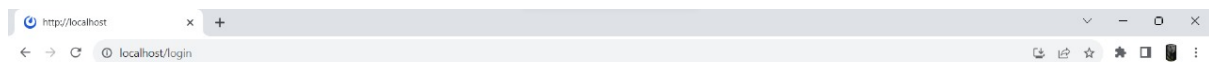
    location / {
        client_max_body_size 50M;
        proxy_set_header Connection "";
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Frame-Options SAMEORIGIN;
    }
}
```

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

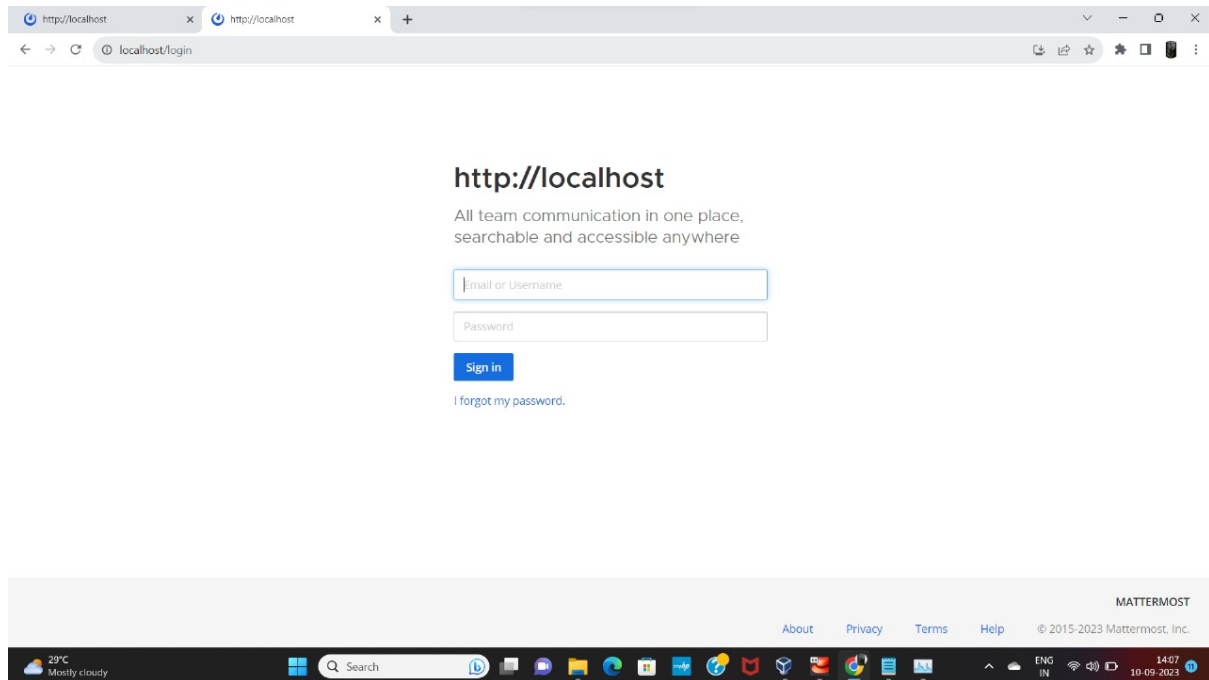
Vm 1 up and vm 2 down

Output



SCENARIO 2: vm 2 up and vm 1 down

Output:



SCENARIO 3: both nginx are down

Output:

