**Step-by-Step Process**

**1. Launch an AWS EC2 Instance**

1. **Login to AWS Management Console**.
2. Navigate to **EC2 Dashboard** and click **Launch Instance**.
3. **Choose Amazon Linux 2 AMI** (or Ubuntu) as your instance.
4. Select an instance type (e.g., t2.micro for free tier).
5. Configure security group rules:
   * Allow **HTTP (port 80)** and **SSH (port 22)** from your IP.
6. **Launch** the instance and download the private key (PEM file) if needed for SSH.
7. **Connect to your EC2 instance** using PuTTY:
   * Convert the .pem file to .ppk using PuTTYgen.
   * Open PuTTY and SSH into your instance using the .ppk key file.

**2. Install Docker on the EC2 Instance**

1. Once connected to the instance, update the package list:

sudo yum update -y # for Amazon Linux

sudo apt-get update # for Ubuntu

1. Install Docker:
   * For Amazon Linux:

sudo amazon-linux-extras install docker

* + For Ubuntu:

sudo apt-get install docker.io

1. Start the Docker service:

sudo systemctl start docker

sudo systemctl enable docker

1. Add your user to the docker group so you can run Docker without sudo:

sudo usermod -aG docker ec2-user # for Amazon Linux

sudo usermod -aG docker ubuntu # for Ubuntu

1. **Log out and log back in** to apply the user group change.
   * **Give** exit
   * **SSH via Putty again.**
2. Once logged back in, check your group membership to verify the change:

groups

You should see docker in the list of groups.

 **Test Docker Without sudo**: After logging back in, test if Docker works without needing sudo:

docker ps

This should work without requiring sudo if the group change has been successfully applied.

**3. Create NGINX Configuration**

1. Create a directory to store the NGINX configuration:

mkdir ~/nginx-conf

1. Create a configuration file inside the directory:

bash

Copy code

nano ~/nginx-conf/default.conf

1. Add the following configuration for URL redirection and connection limiting:

http {

limit\_conn\_zone $binary\_remote\_addr zone=addr:10m;

server {

listen 80;

# Redirect from /old to /new

location /old {

return 301 /new;

}

# Serve /new location

location /new {

root /usr/share/nginx/html;

index index.html;

}

# Limit connections per IP to 10

limit\_conn addr 10;

}

}

1. Save and exit (CTRL+X, then Y).

**4. Run the NGINX Docker Container**

1. Pull the official NGINX Docker image:

docker pull nginx

1. Run the NGINX container with the custom configuration:

docker run -d -p 80:80 -v ~/nginx-conf/default.conf:/etc/nginx/nginx.conf:ro nginx

**5. Troubleshooting If the Container Doesn't Start**

1. If you don't see the container running after executing docker ps, check for stopped containers:

docker ps -a

1. Look for any container that has exited or failed. If you find it, check the logs for errors:

docker logs <container-id>

1. The most common issue is a configuration syntax error. Validate your NGINX configuration using the command:

docker exec <container-id> nginx -t

This will tell you if there’s a configuration issue. You can also manually enter the container to inspect the configuration file:

docker exec -it <container-id> /bin/bash

cat /etc/nginx/nginx.conf

**6. Test the NGINX Setup**

1. Open your browser and go to http://<EC2-instance-public-IP>/old.
2. It should redirect to /new. You should also test connection limits by hitting the endpoint multiple times.
3. You can use curl to simulate multiple requests to check connection limits:

ab -n 100 -c 20 http://<EC2-instance-public-IP>/old

* + This command will send 100 requests with 20 concurrent connections.

**7. (Optional) Set Up Permissions and Firewall for EC2 Instance**

1. If you face access issues, check that your EC2 instance’s **Security Group** has inbound rules allowing traffic on port 80:
   * Go to the **EC2 Dashboard > Instances**.
   * Select your instance, and in the **Security** tab, open your **Security Group**.
   * Ensure there’s an inbound rule for **HTTP (port 80)** from your IP or from 0.0.0.0/0.
2. Also, ensure **port 22 (SSH)** is open so that you can SSH into the instance.

**Summary of Key Steps:**

1. **Launch an EC2 instance** and **connect via PuTTY**.
2. **Install Docker** and **configure NGINX** for redirection and connection limits.
3. **Run the NGINX container** and troubleshoot any issues by checking container logs.
4. **Test URL redirection** and **connection limiting** by using a browser or command-line tools like curl or ab.

Let me know if any steps need further clarification!