Blocking an IP address with nginx

nginx is a powerful and flexible web server, widely used for serving high-performance web content. One of its many uses is the ability to manage server access based on IP addresses. Blocking or allowing access to specific IPs can be useful for various reasons, such as security, traffic management, or restricting access to specific users. In this article, we will see how to configure nginx to block access to certain IP addresses.

To block access to specific IP addresses, you need to use the `deny` directive within the server or location block you want to protect. For example, to block access to an entire website from a specific IP, add the following in the `server` block:

```
server {
    listen 80;
    server_name mysite.com www.mysite.com;

    # Block access to specific IP addresses
    deny 192.168.1.1;
    deny 203.0.113.0/24;

    # Allow access to all others
    allow all;

    location / {
        # Additional configurations
        try_files $uri $uri/ =404;
    }
}
```

In this example, the IP address `192.168.1.1` and the IP range `203.0.113.0/24` will be blocked from accessing the site. The `allow all` directive ensures that all other unspecified IP addresses are allowed.

If you want to block access only to a specific page or directory, you can do so within a `location` block:

```
server {
    listen 80;
    server_name mysite.com www.mysite.com;

    location /admin {
        # Block access to specific IP addresses
        deny 192.168.1.1;
        deny 203.0.113.0/24;

        # Allow access to all others
        allow all;

        # Additional configurations for /admin
        try_files $uri $uri/ =404;
    }
}
```

In this case, only access to the `/admin` directory is restricted to the specified IP addresses.

Conclusions

Blocking access to certain IP addresses with nginx is a simple yet powerful operation for managing traffic and securing your web server. By using the `deny` and `allow` directives, you can easily control who has access to your website's resources. Always remember to test configurations in a staging environment before applying them to production to avoid unintended disruptions.