Troubleshooting

Tips

Checking Access Permissions

- Loag of errors are caused by invalid access permissions. You are offered to specify a user and a group for the Nginx worker processes to run:

1. When configuring the build with the configure command
2. In the configuration file, user directive allows to specify the user and group

Testing Configuration

A common mistake: after having modified the config file (often without a backup) they reload nginx to apply the new configuration

If the configuration file contains syntax or semantic errors, application wil refuse the reload or Nginx is stopped.

Recommendations:

- keep a backup of working configuration files in case something goes wrong
- Before reloading, test ```nginx -t -c /path/to/config/file.conf
- reload server instead of restarting. `service nginx reload`. It will keep existing connections alive.

Reload Server

```
service nginx reload
/etc/init.d/nginx reload
/usr/local/nginx/sbin/nginx -s reload
```

Checking Logs

The error should be located in the /logs/ directroy of Nginx setup. Default is /usr/localnginx/ogs or /var/log/nginx.

Forbidden Custom Error Page

deny or allow directives: clients being denied will fall back on 403 forbidden error page.

```
server {
    allow 192.168.0.0/16;
    deny all;
    error_page 403 /error403.html
}
```

The problem is simply. Nginx also denies access to custom 403 error page. You need to override the access rules in a location block specifically matching the page.

```
server {
    location / {
        error_page 403 /error403.html;
        allow 192.168.0.0/16;
        deny all;
    }
    location = /error403.html {
        allow all;
    }
}
```

```
# Form more than just one error page
server {
    location / {
        error_page 504 /error403.html;
        error_page 404 /error404.html;
        allow 192.168.0.0/16;
        deny all;
    }
    location ~ "^/error[0-9]{3}\.html$" {
        allow all;
    }
}
```

400 Bad Request

Only stops happening when visitors clear their cache and cookies. Is caused by an overly large header field sent by the client. This ocurrs when cookie data exceeds a certain size.

```
# Increase the  buffer to allow larger cookie data size
large_client_header_buffers
large_client_header_buffers 4 16k
```

Truncated or invalid FastCGI responses

- Setup a writable directory for the temporary FastCGI files. `fastcgi_temp_path`

Location Block Prioritties

When using multiple location blocks in the same server block in the same server block is that the config does not apply as you thought it would.

Location blocks are processed in a specific order by priorities.

If Block Issues

Avoid using if blocks