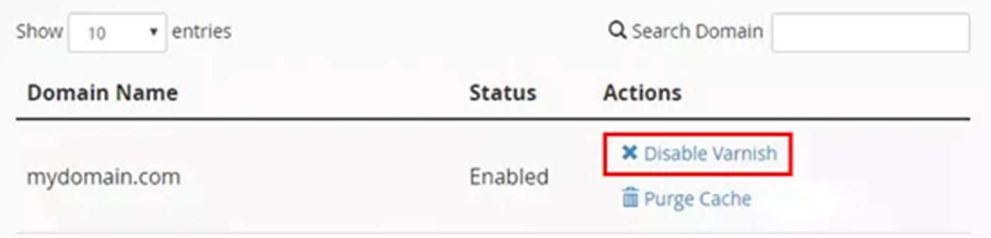
Check Varnish Plugin

If your website uses Varnish cache, you should first determine if the issue is with Varnish or your web server. You can do this by turning off Varnish and trying to fetch without it. Here are the steps to do so:

1. Log in to your site's **cPanel**.
2. Under the Web Accelerator section, select **Manage Varnish > Disable Varnish**.



3. Press **Confirm Action** to disable it.
4. Check if your website loads properly now.

If the site is working now, something is wrong with Varnish. You can re-enable it once to check if restarting fixes the issue. If it doesn't, you can find out what exactly is wrong by checking the log file which you can create with the steps listed below:

To log entries where varnish response or backend response status is 503, use the following command:

```
$ varnishlog -q 'RespStatus == 503' -g request
```

To log all entries >=500, use the following command instead:

```
varnishlog -a -w /var/log/varnish/varnish50x.log -q "RespStatus >= 500 or BerespStatus >= 500"
```

Modify Cache Tag Length

The default length of cache tags in Varnish is **8192 bytes**. Exceeding this can cause the HTTP 503 Backend Fetch Failed error.

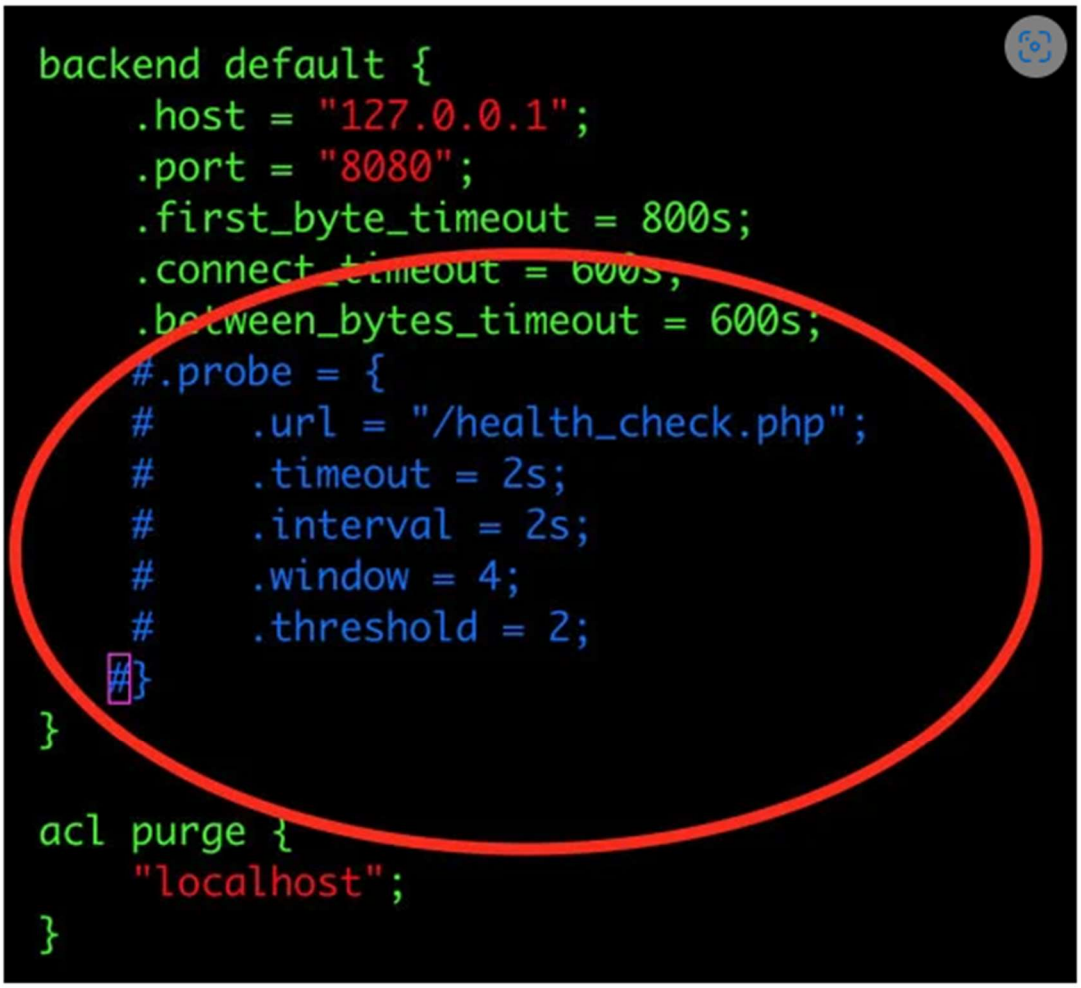
You can resolve this issue by increasing the value of the **http_resp_hdr_len** parameter in your Varnish configuration file.

```
# DAEMON_OPTS is used by the init script.
DAEMON_OPTS="-a ${VARNISH_LISTEN_ADDRESS}:${VARNISH_LISTEN_PORT} \
         -f ${VARNISH_VCL_CONF} \
         -T ${VARNISH_ADMIN_LISTEN_ADDRESS}:${VARNISH_ADMIN_LISTEN_I
         -p thread_pool_min=${VARNISH_MIN_THREADS} \
         -p thread_pool_max=${VARNISH_MAX_THREADS} \
         -p http_resp_hdr_len=70000 \
         -p http_resp_size=100000 \
   -p workspace_backend=98304 \
         -S ${VARNISH_SECRET_FILE} \
         -s ${VARNISH_STORAGE}"
```

Modify Varnish and NGINX Configuration Files

The steps listed are for Varnish but the same can apply for NGINX as well. Here are the steps to modify the configuration files:

1. Locate and open the file from . `/etc/varnish/default.vcl`
2. Find the line and remove **/pub** as such: `.url = "/pub/health_check.php";`
   `.url = "/health_check.php";`
3. Or vice versa. If there's no **/pub**, try adding it.



4. Next, navigate to the Magento 2 root folder and open the file. `ginx.conf.sample`
5. Find the following line:
   `location ~ (index|get|static|report|404|503)\.php$ {`
6. Edit it by adding health_check as such:
   `location ~ (index|get|static|report|404|503|health_check)\.php$ {`
7. Save the changes and restart Varnish.

User-Reported Fixes

> **Note:** User-Reported Fixes generally only apply to niche scenarios but they are still worth checking out.

A user on Stack Exchange reported that another website on the server was causing this issue, which affected his site as well. So if you're currently **hosted on a shared server**, this might be something to consider.

One user said the issue was caused by a **plugin of the hoster**. Once the faulty plugin was disabled, the Backend Fetch Failed error was resolved.

Another user reported that the **maintenance.flag** file generated by Magento was causing the issue for them. This issue can be fixed by logging into your cPanel and renaming or deleting the maintenance.flag file from the Magento root directory.

O **Varnish** é um "*acelerador*", ele é um sistema de cache das páginas, é um **proxy reverso**, ou seja as requisições passam por ele e as cópias ficam armazenadas na memória.

Sobre o erro, **geralmente** ele ocorre quando excede o limite de armazenamento na memória, você pode editar o arquivo o arquivo `varnish` (ou `varnish.params`) e ajustar parâmetro `http_resp_hdr_len` para que o servidor consiga suportar maior consumo (também depende de quanto o seu servidor tem de memória disponível para o serviço, se for *Shared* é provável que seja bem limitado).

Arquivos de configuração em diferentes distros:

- CentOS 6: `/etc/sysconfig/varnish`
- CentOS 7: `/etc/varnish/varnish.params`
- Ubuntu: `/etc/default/varnish`

No entanto se você não tem acesso as configurações só entrando em contato com a hospedagem.

Vou deixar uma coisa bem claro, não estou defendendo as hospedagens a maioria tem problemas pequenos mesmo que muitas vezes causam transtornos, mas este problema **na maioria das vezes** é culpa é dos web-desenvolvedores e administradores de sites (no caso provavelmente você), que injetam inúmeros JavaScript, CSS, plugins e add-ons **sem necessidade** e com isto vão tornando o site cada vez mais pesados e lentos, mesmo com incríveis sistemas de cache e etc o servidor não tem como aguentar.

Então não é bem um problema de configuração, mas sim um problema de mau uso dos recursos, então você pode tentar dar uma revisada geral, analisar se tudo é necessário ou até pensar em trocar o WordPress por uma plataforma mais eficiente, fora isto não tem muito o que fazer.