

# Drupal7?Nginx???

```
server {
    server_name site.com;
    root /var/www/site; ## <-- Your only path reference.

    location = /favicon.ico {
        log_not_found off;
        access_log off;
    }

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    location = /ads.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    # Very rarely should these ever be accessed outside of your lan
    location ~* \.(txt|log)$ {
        allow 192.168.0.0/16;
        deny all;
    }

    location ~ \..*/.*\.php$ {
        return 403;
    }

    location ~ ^/sites/*/private/ {
        return 403;
    }

    # Block access to scripts in site files directory
```

```

location ~ ^/sites/[^/]+/files/.*\.php$ {
    deny all;
}

# Allow "Well-Known URIs" as per RFC 5785
location ~* ^/.well-known/ {
    allow all;
}

# Block access to "hidden" files and directories whose names begin with a
# period. This includes directories used by version control systems such
# as Subversion or Git to store control files.
location ~ (^|/)\. {
    return 403;
}

location / {
    # try_files $uri @rewrite; # For Drupal <= 6
    try_files $uri /index.php?$query_string; # For Drupal >= 7
}

location @rewrite {
    rewrite ^/(.*)$ /index.php?q=$1;
}

# Don't allow direct access to PHP files in the vendor directory.
location ~ /vendor/.*\.php$ {
    deny all;
    return 404;
}

# In Drupal 8, we must also match new paths where the '.php' appears in
# the middle, such as update.php/selection. The rule we use is strict,
# and only allows this pattern with the update.php front controller.
# This allows legacy path aliases in the form of
# blog/index.php/legacy-path to continue to route to Drupal nodes. If
# you do not have any paths like that, then you might prefer to use a
# laxer rule, such as:
# location ~ \.php(/|$) {
# The laxer rule will continue to work if Drupal uses this new URL
# pattern with front controllers other than update.php in a future

```

```

# release.
location ~ /\.php$|^/update.php' {
    fastcgi_split_path_info ^(.+?\.php)(|/.*)$;
    # Security note: If you're running a version of PHP older than the
    # latest 5.3, you should have "cgi.fix_pathinfo = 0;" in php.ini.
    # See http://serverfault.com/q/627903/94922 for details.
    include fastcgi_params;
    # Block httpoxy attacks. See https://httpoxy.org/.
    fastcgi_param HTTP_PROXY "";
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_param QUERY_STRING $query_string;
    fastcgi_intercept_errors on;
    # PHP 5 socket location.
    #fastcgi_pass unix:/var/run/php5-fpm.sock;
    # PHP 7 socket location.
    fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
}

# Fighting with Styles? This little gem is amazing.
# location ~ ^/sites/*/files/imagecache/ { # For Drupal <= 6
location ~ ^/sites/*/files/styles/ { # For Drupal >= 7
    try_files $uri @rewrite;
}

# Handle private files through Drupal. Private file's path can come
# with a language prefix.
location ~ ^(/[a-z\-]+)?/system/files/ { # For Drupal >= 7
    try_files $uri /index.php?$query_string;
}

## PWA serviceworker support.
location ~ ^/pwa/[0-9a-z]+/serviceworker.js {
    try_files $uri /index.php?$query_string;
}

location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff)$ {
    expires max;
    log_not_found off;
    access_log off;
}

```

}

### #1

tainan 7 2024 01:58:50

tainan 7 2024 02:01:45