v1.20_NGINX

- eltex-portal , . . - nginx, 8080 " ".

nginx, tomcat.

nginx

nginx 1.12.2 . : https://nginx.ru/en/linux_packages.html#stable

, softwlc.conf **/etc/nginx/conf.d/** .

softwlc.conf

```
#
# Softwlc server configuration
#

proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=eltex_cache:32m max_size=5g inactive=60m
use_temp_path=off;

map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

server {
    listen 8080;
    listen [::]:8080;

    # SSL configuration
    #    listen 8443 ssl;
    #    listen [::]:8443 ssl;

    # Enables or disables gzipping of responses.
    gzip on;

    # Enables or disables decompression of gzipped responses for clients that lack gzip support.
    # If enabled, the following directives are also taken into account when determining if clients support
gzip: gzip_http_version, gzip_proxied, and gzip_disable.
    # See also the gzip_vary directive.
    gunzip on;

    # The ngx_http_gzip_static_module module allows sending precompressed files with the “.gz” filename
extension instead of regular files.
    # This module is not built by default, it should be enabled with the --with-http_gzip_static_module
configuration parameter.
    # Enables (“on”) or disables (“off”) checking the existence of precompressed files. The following
directives are also taken into account: gzip_http_version, gzip_proxied, gzip_disable, and gzip_vary.
    # With the “always” value (1.3.6), gzipped file is used in all cases, without checking if the client
supports it.
    # It is useful if there are no uncompressed files on the disk anyway or the ngx_http_gunzip_module is used.
    gzip_static on;

    # Enables gzipping of responses for the specified MIME types in addition to “text/html”.
    # The special value “*” matches any MIME type (0.8.29). Responses with the “text/html” type are always
compressed
    gzip_types text/html text/plain text/css image/x-icon image/bmp image/png image/gif image/jpeg image/jpg
application/json application/x-javascript application/javascript text/javascript;
    gzip_comp_level 9;


    # Sets the size of the buffer used for reading the first part of the response received from the proxied
server.
    # This part usually contains a small response header. By default, the buffer size is equal to one memory
page.
```

```
    # This is either 4K or 8K, depending on a platform. It can be made smaller, however.
    proxy_buffer_size 128k;

    # Sets the number and size of the buffers used for reading a response from the proxied server, for a single
connection.
    # By default, the buffer size is equal to one memory page. This is either 4K or 8K, depending on a platform.
    proxy_buffers 4 256k;

    # When buffering of responses from the proxied server is enabled, limits the total size of buffers that can
be busy sending
    # a response to the client while the response is not yet fully read.
    # In the meantime, the rest of the buffers can be used for reading the response and, if needed, buffering
part of the response to a temporary file.
    # By default, size is limited by the size of two buffers set by the proxy_buffer_size and proxy_buffers
directives.
    proxy_busy_buffers_size 256k;

    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    #    ssl_certificate "__"
    #    ssl_certificate_key "___";

    # server_name _ ;


    # When enabled, only one request at a time will be allowed to populate a new cache element
    # identified according to the proxy_cache_key directive by passing a request to a proxied server.
    # Other requests of the same cache element will either wait for a response to appear in the cache
    # or the cache lock for this element to be released, up to the time set by the proxy_cache_lock_timeout
directive.
    proxy_cache_lock on;

    # Allows starting a background subrequest to update an expired cache item, while a stale cached response is
returned to the client.
    # Note that it is necessary to allow the usage of a stale cached response when it is being updated
(proxy_cache_use_stale updating)
    # Be aware of the fact that these options appeared in the newer nginx versions
    proxy_cache_background_update on;

    # The updating parameter permits using a stale cached response if it is currently being updated.
    # This allows minimizing the number of accesses to proxied servers when updating cached data.
    proxy_cache_use_stale error timeout updating http_500 http_502 http_503 http_504;

    # Enables revalidation of expired cache items using conditional requests with the “If-Modified-Since” and
“If-None-Match” header fields.
    proxy_cache_revalidate on;

    # Sets the number of requests after which the response will be cached.
    proxy_cache_min_uses 2;

    # Enables byte-range support for both cached and uncached responses from the proxied server regardless of
the “Accept-Ranges” field in these responses.
    proxy_force_ranges on;


    # Sets the maximum allowed size of the client request body, specified in the “Content-Length” request
header field.
    # If the size in a request exceeds the configured value, the 413 (Request Entity Too Large) error is
returned to the client.
    # Please be aware that browsers cannot correctly display this error. Setting size to 0 disables checking of
client request body size.
    client_max_body_size 25m;

    # Allows redefining or appending fields to the request header passed to the proxied server.
    proxy_set_header Host $http_host;
```

```
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-Server $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Real-IP $remote_addr;


    # Eltex Portal section
    location ~* /eltex_portal/(gif|jpeg|jpg|png|svg|js|css|fonts)/ {
        proxy_cache eltex_cache;

        proxy_ignore_headers Cache-Control Expires;
        proxy_cache_valid any 1h;

        #  eltex portal
        proxy_pass http://127.0.0.1:9000;

    }

    # caching shared resources that are downloaded by id for a day (these are images and everything except text)
    # and totally static resources (/img/) are to stay for a while too
    # you can actually keep these forever, just be aware that they consume some space on your disk (don't
forget to change the size of your cache!)
    location ~* /eltex_portal/(portal/download/shared/[0-9]+|img/) {
        proxy_cache eltex_cache;
        proxy_ignore_headers Cache-Control Expires;
        proxy_cache_valid 404 302 304 5m;
        proxy_cache_valid any 24h;
        #  eltex portal
        proxy_pass http://127.0.0.1:9000;
    }

    # caching generated portal styles
    location ~* /eltex_portal/portal/download/private/.+/generated-style {
        proxy_cache eltex_cache;
        proxy_set_header If-Modified-Since $http_if_modified_since;
        proxy_cache_valid 404 302 304 5m;
        proxy_cache_valid any 1h;
        proxy_pass http://127.0.0.1:9000;
    }

    # caching private resources (that are text usually)
    location ~* /eltex_portal/(portal/download/private|portal)/ {
        proxy_cache eltex_cache;
        proxy_ignore_headers Cache-Control Expires;
        proxy_cache_valid 404 302 304 5m;
        proxy_cache_valid any 1h;
        #  eltex portal
        proxy_pass http://127.0.0.1:9000;
    }

    location /eltex_portal/ {
        #  eltex portal
        proxy_pass http://127.0.0.1:9000;
    }

    # portal-polly proxy
    location /eltex_portal/polly {
        rewrite ^(/eltex_portal)/(.*)$ /$2 break;
        proxy_cookie_path /polly /eltex_portal/polly;
        proxy_pass http://127.0.0.1:9089;
    }


    # Eltex Portal Constructor section
    # caching shared resources and static resources
    location ~* /epadmin/(portal/download/[0-9]+|portal/download/shared/[0-9]+|font-awesome|img|js|css) {
        proxy_cache eltex_cache;
        proxy_cache_valid any 24h;
        #  eltex portal constructor
        proxy_pass http://127.0.0.1:9001;
    }
```

```
location /epadmin/ {
    #  eltex portal constructor
    proxy_pass http://127.0.0.1:9001;
}

location /ep-demo {
    #  eltex portal constructor
    proxy_pass http://127.0.0.1:9001;
}

# epadmin-polly proxy
location /epadmin/polly {
    rewrite ^(/epadmin)/(.*)$ /$2 break;
    proxy_cookie_path /polly /epadmin/polly;
    proxy_pass http://127.0.0.1:9089;
}

# wifi-cab

location /wifi-cab {
    #  wifi-cab
    proxy_pass http://127.0.0.1:8083;
}

location /wifi-cab/PUSH {
   proxy_http_version 1.1;
   proxy_set_header Upgrade $http_upgrade;
   proxy_set_header Connection $connection_upgrade;
   proxy_buffering off;
   proxy_ignore_client_abort off;
   proxy_read_timeout 86400s;
   proxy_send_timeout 86400s;
   #  wifi-cab
   proxy_pass http://127.0.0.1:8083/wifi-cab/PUSH;
}

# nbi

location /axis2 {
    #  tomcat
    proxy_pass http://127.0.0.1:8081;
    proxy_set_header Host $http_host;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;
}

location /northbound {
    proxy_pass http://127.0.0.1:8081;
    proxy_set_header Host $http_host;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;
}

location /eltex-radius-nbi {
    proxy_pass http://127.0.0.1:8081;
    proxy_set_header Host $http_host;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection $connection_upgrade;
}

# ems
location /ems {
    #  tomcat
    proxy_pass http://127.0.0.1:8081;
}
```

```
    location /ems_files {
        proxy_pass http://127.0.0.1:8081;
    }

    # deny root

    #
    location / {
        deny all;
    }
}
```

tomcat

/etc/tomcat7/server.xml 8080 8081 **Service**:

```
    <Connector port="8081" protocol="HTTP/1.1"
               connectionTimeout="20000"
               URIEncoding="UTF-8"
               redirectPort="8443" />
```

Host:

```
        <Valve className="org.apache.catalina.valves.RemoteIpValve"
           remoteIpHeader="X-Forwarded-For"
           internalProxies="127\.0\.0\.1"
           requestAttributesEnabled="true"/>
```

( ): server.xml

tomcat, 8080, nginx, .

```
service tomcat7 restart
service nginx restart
```

web- .