Server preparation - Linux - CentOS8

installation, java, tomcat, quickstart, apache httpd

This tutorial shows how to prepare the server for test or production use of CzechIdM. If you are looking for much quicker way of how to start the CzechIdM, use the demo setup described here Getting Started

Basic system setup

- 1 server (can be virtualized) for everything: backend, frontend and database.
- OS Linux with EPEL repository enabled - CentOS, basic network enabled installation
  - It is possible to use Debian (we tested on Stretch) or other distributions, but you have to adjust steps in this guide accordingly.
- PostgreSQL 12.x - installed from OS packages.
- Java 11 (Java 21 for CzechIdM 13.1.0+) - installed from OS packages.
- Apache Tomcat 9.0.x - installed manually into `/opt/tomcat`.
- Apache HTTPd 2.4.x - installed from OS packages. Can be replaced by nGinx.
- All services start via systemd.
- Each service runs under dedicated non-privileged user.

Instalation and software configuration

Prerequisities - Basic installation of CentOS 8

```
# EPEL installation
dnf clean all
dnf -y install epel-release
dnf -y update

# other recommended packages installation
dnf -y install mc haveged nmap screen sysstat telnet net-tools nano wget
vim-enhanced bzip2 bash-completion lsof zip unzip psmisc policycoreutils-
python-utils tar

# enable haveged after OS start
systemctl start haveged.service
systemctl enable haveged.service

# set the hostname
hostnamectl set-hostname FQDN_server_name
hostnamectl status
# check the network configuration, be sure it is static
(/etc/sysconfig/network-scripts/)
# reboot the server
```

PostgreSQL

If you are installing CzechIdM on Microsoft SQL Server, please follow this tutorial.

We install PostgreSQL 12 database binaries and change database data directory from `/var/lib` to `/data`.

Database server installation - CentOS8

- Software installation on CentOS8(versions can vary):

```
# enable module postgres 12
dnf module enable postgresql:12
dnf -y install postgresql-server postgresql-contrib postgresql-libs
```

- create new directory for database data:

```
mkdir -p /data/pgsql/12/data/
chown -R postgres:postgres /data/pgsql/
chmod 700 /data/pgsql
```

- Copy the PostgreSQL's systemd unit to the /etc:

```
cp /usr/lib/systemd/system/postgresql.service /etc/systemd/system/
```

In the file `/etc/systemd/system/postgresql.service` change the directory for data as follows:

```

# Location of database directory
Environment=PGDATA=/data/pgsql/12/data/
```

- In the file `/var/lib/pgsql/.bash_profile` (bash profile for postgres user) change the variable PGDATA to:

```
PGDATA=/data/pgsql/12/data
```

- Reload changes:

```
systemctl daemon-reload
```

- Initialize database:

```
postgresql-setup --initdb --unit postgresql
```

Change SELINUX labels:

```
chcon -Rt postgresql_db_t /data/pgsql/
chcon -Rt postgresql_log_t /data/pgsql/12/data/log/
```

- Enable and start database:

```
systemctl start postgresql.service
systemctl enable postgresql.service
```

- Check that the database is running:

```
[root@HOSTNAME data]# systemctl status postgresql.service -l
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/etc/systemd/system/postgresql.service; enabled; vendor
preset: disabled)
   Active: active (running) since Wed 2020-03-11 10:48:06 CET; 1min 8s ago
 Main PID: 25715 (postmaster)
    Tasks: 8 (limit: 52428)
   Memory: 19.8M
   CGroup: /system.slice/postgresql.service
           ├─25715 /usr/bin/postmaster -D /data/pgsql/12/data/
           ├─25716 postgres: logger
           ├─25718 postgres: checkpointer
           ├─25719 postgres: background writer
           ├─25720 postgres: walwriter
           ├─25721 postgres: autovacuum launcher
           ├─25722 postgres: stats collector
           └─25723 postgres: logical replication launcher

Mar 11 10:48:06 HOSTNAME systemd[1]: Starting PostgreSQL database server...
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.255 CET
[25715] LOG:  starting PostgreSQL 12.1 on x86_64-redhat-linux-gnu, compiled
by gcc (G>
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.256 CET
[25715] LOG:  listening on IPv6 address "::1", port 5432
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.256 CET
[25715] LOG:  listening on IPv4 address "127.0.0.1", port 5432
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.285 CET
[25715] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.300 CET
[25715] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.330 CET
[25715] LOG:  redirecting log output to logging collector process
Mar 11 10:48:06 HOSTNAME postmaster[25715]: 2020-03-11 10:48:06.330 CET
[25715] HINT:  Future log output will appear in directory "log".
Mar 11 10:48:06 HOSTNAME systemd[1]: Started PostgreSQL database server.
```

Database server configuration and sizing

- Enable the password authentication.

In the file `/data/pgsql/12/data/pg_hba.conf` find lines:

```
host    all             all             127.0.0.1/32            ident
```

```
host    all             all             ::1/128                 ident
```

and change the value at the end of each line to `md5` like this:

```
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

- Adjust DB instance sizing.
  - In following snippet, we presume the system has 3GB of memory dedicated for the database and about 100 db connections. **For your deployment, adjust the sizing accordingly. Use a calculator if in doubt**.
  - We also log queries running longer than 200ms.

In a file `/data/pgsql/12/data/postgresql.conf` change (or add) following lines:

```
# This is an EXAMPLE. Use the calculator to adjust for your deployment!

# DB Version: 12
# OS Type: linux
# DB Type: web
# Total Memory (RAM): 3 GB
# Connections num: 100
# Data Storage: ssd
max_connections = 100
shared_buffers = 768MB
effective_cache_size = 2304MB
maintenance_work_mem = 192MB
checkpoint_completion_target = 0.7
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 200
work_mem = 3932kB
min_wal_size = 1GB
max_wal_size = 4GB

log_min_duration_statement = 200
```

- Restart the database

```
systemctl restart  postgresql.service
```

If you install the database to a different server than the CzechIdM application itself, don't forget to configure PostgreSQL with SSL certificates and to enforce remote SSL connections.

Java - CentOS8

Tomcat application server needs Java installed. We recommend to use OpenJDK 11 from standard OS repository. (OpenJDK 1.8 is also supported, check compatibility page).

Installation:

```
dnf install -y java-11-openjdk-headless java-11-openjdk-devel
```

For CzechIdM 13.1.0+:

```
dnf install -y java-21-openjdk-headless java-21-openjdk-devel
```

Tomcat

- Create a new group and add user for the tomcat to run under:

```
groupadd -r tomcat
useradd -r -s /usr/sbin/nologin -g tomcat -d /opt/tomcat tomcat
getent passwd tomcat
#tomcat:x:995:993::/opt/tomcat:/usr/sbin/nologin
```

- change working directory into /opt/tomcat

```
mkdir /opt/tomcat
cd /opt/tomcat
```

- Download Apache Tomcat 9.0.x from the website https://tomcat.apache.org/download-90.cgi to `/opt/tomcat/`
    - In our exapmle the version is 9.0.45.
- extract files from the archive:

```
tar xzf apache-tomcat-9.0.45.tar.gz
```

- create a new symbolic link to current user version (we presume there may be more versions at the server in future due to upgrades/updates)

```
cd /opt/tomcat
ln -s apache-tomcat-9.0.45 current
```

- Set rights on files for tomcat user (still working under root):

```
chown -R root:root /opt/tomcat
chown root:tomcat /opt/tomcat
chmod 750 /opt/tomcat
cd /opt/tomcat/current
chmod -R o+rX ./
chgrp -R tomcat conf/ bin/ lib/
```

```
chmod g+rx conf
chmod g+r conf/*
chown -R tomcat webapps/ work/ temp/ logs/

mkdir /opt/tomcat/current/conf/Catalina
chown tomcat:tomcat /opt/tomcat/current/conf/Catalina
chmod 750 /opt/tomcat/current/conf/Catalina
```

Start Tomcat automatically after system startup

- Create startup script (systemd unit), in which we also set the basic JVM parameters:

```
vim /etc/systemd/system/tomcat.service
```

- File content of `/etc/systemd/system/tomcat.service`:

tomcat.service

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target postgresql.service

[Service]
Type=forking

PIDFile=/opt/tomcat/current/temp/tomcat.pid

Environment=JAVA_HOME=/usr/lib/jvm/java-openjdk
Environment=CATALINA_PID=/opt/tomcat/current/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat/current
Environment=CATALINA_BASE=/opt/tomcat/current
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -
XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -
Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
Djavax.servlet.request.encoding=UTF-8'

ExecStart=/opt/tomcat/current/bin/startup.sh
ExecStop=/opt/tomcat/current/bin/shutdown.sh

User=tomcat
Group=tomcat

[Install]
WantedBy=multi-user.target
```

- Values of `-Xms` and `-Xmx` se are closely dependent on server sizing. If you have enough memory, we strongly recommend to use `-Xmx 6128M` or more.
- Tomcat will be started under user `tomcat:tomcat`.

- Reload systemd configuration:

```
systemctl daemon-reload
```

- Start the Tomcat to ensure it is configured properly. Enable its start on OS start.

```
systemctl start tomcat
systemctl enable tomcat
```

- Check that Tomcat runs with desirable parameters:

```
[root@tomcat1 logs]# ps -ef | grep ^tomcat
tomcat      1623       1  9 11:08 ?        00:00:04 /usr/lib/jvm/java-
openjdk/bin/java -
Djava.util.logging.config.file=/opt/tomcat/current/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -
Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=true -
Djavax.servlet.request.encoding=UTF-8 -Djdk.tls.ephemeralDHKeySize=2048 -
Djava.protocol.handler.pkgs=org.apache.catalina.webresources -
Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Xms512M -Xmx1024M
-server -XX:+UseParallelGC -Dignore.endorsed.dirs= -classpath
/opt/tomcat/current/bin/bootstrap.jar:/opt/tomcat/current/bin/tomcat-
juli.jar -Dcatalina.base=/opt/tomcat/current -
Dcatalina.home=/opt/tomcat/current -Djava.io.tmpdir=/opt/tomcat/current/temp
org.apache.catalina.startup.Bootstrap start
```

- Stop the Tomcat.

```
systemctl stop tomcat
```

Apache Tomcat configuration

Interface Management

Apache Tomcat offers two applications for tomcat management available at:

- http://localhost:8080/manager
- http://localhost:8080/host-manager

These applications are optional but even when you will not install them you need to

set admin password to increase security of Tomcat.

If you want to use them, it is necessary to do following steps.

First of all, create a Tomcat's database user that you will use for the access to those applications. If you plan to connect to the applications remotely (not only from localhost) you have to also allow communication from your IP.

- Create administration user
  - Create the a new user in the file `/opt/tomcat/current/conf/tomcat-users.xml` and assign him roles "manager-gui" and "admin-gui".
  - The documentation of available roles as well as overall configuration of the application is a part of application installation available at http://localhost:8080/docs/manager-howto.html#Configuring_Manager_Application_Access

The file `/opt/tomcat/current/conf/tomcat-users.xml` should now look like this:

tomcat-users.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-
users.xsd"
              version="1.0">
  <role rolename="manager-script"/>
  <role rolename="manager-gui"/>
  <role rolename="manager-jmx"/>
  <role rolename="manager-status"/>
  <role rolename="admin-gui"/>
  <user username="admin" password="*****store it somewhere safe*****"
roles="manager-gui,manager-status,admin-gui"/>
</tomcat-users>
```

- If you plan to connect to the applications remotely (not only from localhost) you have to also allow communication from your IP.
  - If you see `403 Access Denied` when accessing Tomcat's management remotely, it might be because you did not perform this configuration.

Add your IP address into application configuration files. In files `/opt/tomcat/current/webapps/manager/META-INF/context.xml` and `/opt/tomcat/current/webapps/host-manager/META-INF/context.xml` add netmask for your IP (both files should have the same content):

For example, if you want to access Tomcat's management from the network `192.168.0.0/24`:

context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Context antiResourceLocking="false" privileged="true">
```

```
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127.d+.d+.d+|::1|0:0:0:0:0:0:0:1|192\.168.d+.d+" />
</Context>
```

- Again, restart the tomcat

```
systemctl restart tomcat
```

Apache Tomcat configuration recommended for production use

We advise to follow these steps to configure Tomcat for production deployment.

- Remove unnecessary applications that come with Tomcat:

```
rm -rf /opt/tomcat/current/webapps/{examples,docs,ROOT,host-manager,manager}
```

- Turn off the shutdown port:
  - In the config file `/opt/tomcat/current/conf/server.xml` set value -1 from 8005 to the Server port tag, thus you deactivate it:

```
<Server port="-1" shutdown="SHUTDOWN">
```

- Make Tomcat listen only on localhost:
  - In the `/opt/tomcat/current/conf/server.xml` add the `address="127.0.0.1"` property to configuration of `8080` port.
- Set the `maxSwallowSize` for the HTTP/1.1 connector:
  - In the `/opt/tomcat/current/conf/server.xml`, locate the configuration for port 8080 and add the `maxSwallowSize="-1"` property therein.
- In same file configure AJP port (`8009/tcp`) to look like this:

```
<Connector protocol="AJP/1.3"
               address="127.0.0.1"
               secretRequired="true"
               secret="***password for ajp port***"
               port="8009"
               redirectPort="8443" />
```

- Do not show aplication server version:
  - In the file `/opt/tomcat/current/conf/web.xml` set `showServerInfo` to false (default is true):

```
    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-
class>
        <init-param>
            <param-name>debug</param-name>
            <param-value>0</param-value>
```

```
        </init-param>
        <init-param>
            <param-name>listings</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>showServerInfo</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
```

Rotating Tomcat logs

Default Tomcat logger appends to the logfile, it is therefore safe to use simple `logrotate` configuration. Save following as `/etc/logrotate.d/tomcat`, adjust log retention (the `rotate COUNT`) as necessary.

tomcat

```
/opt/tomcat/current/logs/catalina.out {
    rotate 90
    daily
    dateext
    copytruncate
    missingok
    notifempty
    compress
}
```

It is possible that, on some distros, SELinux will deny acces to the logfile for logrotate because `logrotate_t` is only allowed in the `/var/log` and subfolders. The logrotate will error to the `/var/log/messages` line similar to `Sep 3 03:48:01 server.tld logrotate: ALERT exited abnormally with [1]`.

If this happens, set the permissive mode for logrotate:

```
semanage permissive -a logrotate_t
```

Evaluate impact of SELinux adjustments **before** you implement them. Proper mitigation heavily depends on habits and security policies of your organization.

There are some possibilities:

- Set permissive mode for logrotate as above.
- Set permissive mode for whole SELinux. (This will drop the SELinux's protective function.)
- Adjust particular SELinux labels. Example (here).

Apache httpd as a reverse proxy

It is possible to open Apache Tomcat to the network directly, but little inconvenient. You want the users to access the CzechIdM on user-friendly ports 80/tcp or 443/tcp, which is not easy to setup in Tomcat itself running under nonprivileged user. So we use Apache httpd as a reverse proxy. Apache httpd will allow access to data via https on port 443/tcp and http on port 80/tcp. Communication via http protocol will be enabled, but we will redirect all communication to https. Communication between Apache httpd and Tomcat will take place on local machine via AJP protocol. In httpd, there will be mod_security installed (optional but recommended), which serves as an application firewall.

The configuration example is written for the server which allows access to its services under the name "demo.czechidm.com".

HTTPd installation and configuration

Install httpd and mod_security:

```
yum install -y httpd httpd-tools mod_ssl mod_security mod_security_crs
```

HTTPd basic configuration:

Change MPM to worker - in the file `/etc/httpd/conf.modules.d/00-mpm.conf` comment-out all lines but mod_mpm_worker.so:

```
# Select the MPM module which should be used by uncommenting exactly
# one of the following LoadModule lines:

# prefork MPM: Implements a non-threaded, pre-forking web server
# See: http://httpd.apache.org/docs/2.4/mod/prefork.html
#LoadModule mpm_prefork_module modules/mod_mpm_prefork.so

# worker MPM: Multi-Processing Module implementing a hybrid
# multi-threaded multi-process web server
# See: http://httpd.apache.org/docs/2.4/mod/worker.html
#
LoadModule mpm_worker_module modules/mod_mpm_worker.so

# event MPM: A variant of the worker MPM with the goal of consuming
# threads only for connections with active processing
# See: http://httpd.apache.org/docs/2.4/mod/event.html
#
#LoadModule mpm_event_module modules/mod_mpm_event.so
```

Disable "welcome" page:

```
cd /etc/httpd/conf.d
mv welcome.conf welcome.conf-DISABLED
```

```
touch welcome.conf
```

Virtualhost configuration to forward the communication from port 80 to 443. Add following section and change string 'SERVER' to the real servername in the file `/etc/httpd/conf.d/vhost-redirect.conf`:

```
<VirtualHost _default_:80>
   DocumentRoot /var/www/html
   Redirect permanent / https://SERVER/
</VirtualHost>
```

Set the proxy in the virtualhost for https (443/tcp) - at the end of the file `/etc/httpd/conf.d/ssl.conf` add following before ending "tag" VirtualHost:

```

  Protocols       https/1.1
  ProxyRequests     off
  ProxyPreserveHost on
  ProxyAddHeaders on
  ProxyPass / ajp://127.0.0.1:8009/ secret=**tomcat_ajp_secret**
  ProxyPassReverse / ajp://127.0.0.1:8009/ secret=**tomcat_ajp_secret**
```

In IE 11, CzechIdM has problems with missing icons. Icons are created by special fonts and those fonts are handled badly in the IE. It is necessary to set `Cache-Control` HTTP header. We need to set it only for font files:

```
# workaround for bad font handling in IE 11
<LocationMatch "/idm/.*(\.ttf|\.woff2|\.eot)$">
        Header set Cache-Control "no-cache, public, must-revalidate, proxy-
revalidate"
</LocationMatch>
```

Identity manager CzechIdM will be available on address https://server/idm/ It is possible to forward from / to /idm/, so that the user does not need to type the whole URL. To do so, add following lines to the virtualhost config file (ssl.conf):

```

RewriteEngine On
RewriteRule "^/$"  "/idm/" [R]
```

Certificate for httpd

If you have prepared certifikate, key and certificate authority chain just chnge these properties in `/etc/httpd/conf.d/ssl.conf` and make sure that only httpd can read the files.

```
    SSLCertificateFile PATH_TO_CERTIFICATE_FILE
    SSLCertificateKeyFile PATH_TO_CERTIFICATE_KEY_FILE
    SSLCertificateChainFile PATH_TO_CA_CHAIN_FILE
```

Then continue with cheking syntax of httpd.

If you not prepared them in the moment. Create temporary certificate and key.

```
mkdir /etc/httpd/cert
cd /etc/httpd/cert
openssl genrsa -out http_temp_cert.key
openssl req -new -key http_temp_cert.key -out http_temp_cert.csr -subj
"/C=CZ/ST=Czech Republic/L=Prague/O=BCV/CN=CzechIdM placeholder cert"
openssl x509 -req -in http_temp_cert.csr -signkey http_temp_cert.key -days 1
-sha256 -out http_temp_cert.crt
rm http_temp_cert.csr
chmod 600 /etc/httpd/cert/*
chown -R apache:apache  /etc/httpd/cert/
```

Then change set path to them in these properties in `/etc/httpd/conf.d/ssl.conf`.

```
    SSLCertificateFile /etc/httpd/cert/http_temp_cert.crt
    SSLCertificateKeyFile /etc/httpd/cert/http_temp_cert.key
```

Checking httpd configuration syntax and configuring selinux

Syntax check before httpd restart

```
httpd -t -D DUMP_VHOST
# or apachectl configtest
```

httpd restart and reload configuration changes:

```
systemctl restart httpd
```

Allow in SELINUX to httpd connect to network:

```
/usr/sbin/setsebool -P httpd_can_network_connect 1
```

Enable httpd after OS start:

```
systemctl enable httpd.service
```

mod_security configuration

Mod_security files locations (on CentOS8):

- Audit log: `/var/log/httpd/modsec\_audit.log`
- Directory with activated rules: `/etc/httpd/modsecurity.d/activated\_rules/`
- basic configuration file for mod_security: `/etc/httpd/modsecurity.d/activated_rules/REQUEST-901-INITIALIZATION.conf`

- The file for chosen rules deactivation: `/etc/httpd/conf.d/ssl.conf`

The default set of rules is relatively strict. CzechIdM cannot run with the default configuration of mod_security.

Each rule is identified by a unique ID. If you want to deactivate the whole rule, it is advised to write the rule ID into ssl.conf like this:

```
  <IfModule mod_security2.c>
    SecRuleRemoveById RULE_ID
  </IfModule>
```

Disabling mod_security rules

These rules are disabled for modsec_crs 3.0.

In the file `/etc/httpd/conf.d/ssl.conf` deactivate following rules and set their logging:

```
<IfModule mod_security2.c>
        SecRuleRemoveById 942430
        SecRuleRemoveById 942431
        SecRuleRemoveById 920300
        SecRuleRemoveById 920230

        # Allow Czech signs
        SecRuleRemoveById 942110
        SecRuleRemoveById 942330
        SecRuleRemoveById 942460
        SecRuleRemoveById 942260

        # Too restrictive for login format
        SecRuleRemoveById 920440

        # Needed by Websockets
        <Location "/idm/api/v1/websocket-info/">
                SecRuleRemoveById 950100
        </Location>

        # do not log request/response body
        SecAuditLogParts AFHZ
</IfModule>
```

mod_security configuration - CentOS8

Edit the file `/etc/httpd/modsecurity.d/activated_rules/REQUEST-901-INITIALIZATION.conf`.

- find the rule `900200` and add methods PUT, DELETE and PATCH on the line starting `tx.allowed\_methods`. It should look like this after change:

```
# Default HTTP policy: allowed_methods (rule 900200)
SecRule &TX:allowed_methods "@eq 0" \
    "id:901160,\
    phase:1,\
    pass,\
    nolog,\
    setvar:'tx.allowed_methods=GET HEAD POST OPTIONS PUT PATCH DELETE'"
```

- find the rule `900220` and add support for content type `application/hal+json` on the line starting with `tx.allowed\_request\_content\_type`. Result should look like this:

```
# Default HTTP policy: allowed_request_content_type (rule 900220)
SecRule &TX:allowed_request_content_type "@eq 0" \
    "id:901162,\
    phase:1,\
    pass,\
    nolog,\
    setvar:'tx.allowed_request_content_type=application/x-www-form-
urlencoded|multipart/form-data|text/xml|application/xml|application/x-
amf|application/json|text/plain|application/hal+json'"
```

mod_deflate configuration

It is advised to set up gzip so the users get minimum of data from the frontend server. In the file `/etc/httpd/conf.d/ssl.conf` we add following configuration and restart the server:

```
<IfModule mod_deflate.c>
        # Compress HTML, CSS, JavaScript, Text, XML and fonts
        AddOutputFilterByType DEFLATE application/javascript
        AddOutputFilterByType DEFLATE application/rss+xml
        AddOutputFilterByType DEFLATE application/vnd.ms-fontobject
        AddOutputFilterByType DEFLATE application/x-font
        AddOutputFilterByType DEFLATE application/x-font-opentype
        AddOutputFilterByType DEFLATE application/x-font-otf
        AddOutputFilterByType DEFLATE application/x-font-truetype
        AddOutputFilterByType DEFLATE application/x-font-ttf
        AddOutputFilterByType DEFLATE application/x-javascript
        AddOutputFilterByType DEFLATE application/xhtml+xml
        AddOutputFilterByType DEFLATE application/xml
        AddOutputFilterByType DEFLATE font/opentype
        AddOutputFilterByType DEFLATE font/otf
        AddOutputFilterByType DEFLATE font/ttf
        AddOutputFilterByType DEFLATE image/svg+xml
        AddOutputFilterByType DEFLATE image/x-icon
        AddOutputFilterByType DEFLATE text/css
        AddOutputFilterByType DEFLATE text/html
        AddOutputFilterByType DEFLATE text/javascript
        AddOutputFilterByType DEFLATE text/plain
        AddOutputFilterByType DEFLATE text/xml
```

```
        AddOutputFilterByType DEFLATE application/json
        AddOutputFilterByType DEFLATE application/hal+json

        # Remove browser bugs (only needed for really old browsers)
        BrowserMatch ^Mozilla/4 gzip-only-text/html
        BrowserMatch ^Mozilla/4\.0[678] no-gzip
        BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
        Header append Vary User-Agent
</IfModule>
```