ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

shib-docker

Ein Docker-Host zur Containerisierung der Shibboleth-Umgebung.

Fabian Mangels, AWI

Inhaltsverzeichnis

Aufbau

Die produktive Shibboleth-Umgebung des AWIs wird auf einer Virtuellen Maschine (VM) gehostet, die sich in der DMZ befindet. Als Server-Betriebssystem wird Ubuntu in der Version 18.04 LTS (bionic) eingesetzt. Die Containervirtualisierung wird mit Docker realisiert unter Verwendung des Hilfswerkzeugs Docker-Compose. Für die Dienste eines Shibboleth IdPs bzw. Sps werden eigene Docker-Images benötigt, die wiederum Ubuntu als Basis-Image verwenden. Konfigurationen der einzelnen Container / Services werden über Volumes von der Host-VM bereitgestellt. Des Weiteren werden alle üblichen Log-Dateien aus der herkömmlichen Server-Administration mit Hilfe der Volumes gemountet. Als Reverse-Proxy kommt der Webserver Nginx zum Einsatz, der die Host-Zugriffe verwaltet und auf die jeweiligen Docker-Container weiterleitet. Einen besseren Überblick bietet die nachstehende Abbildung 1.

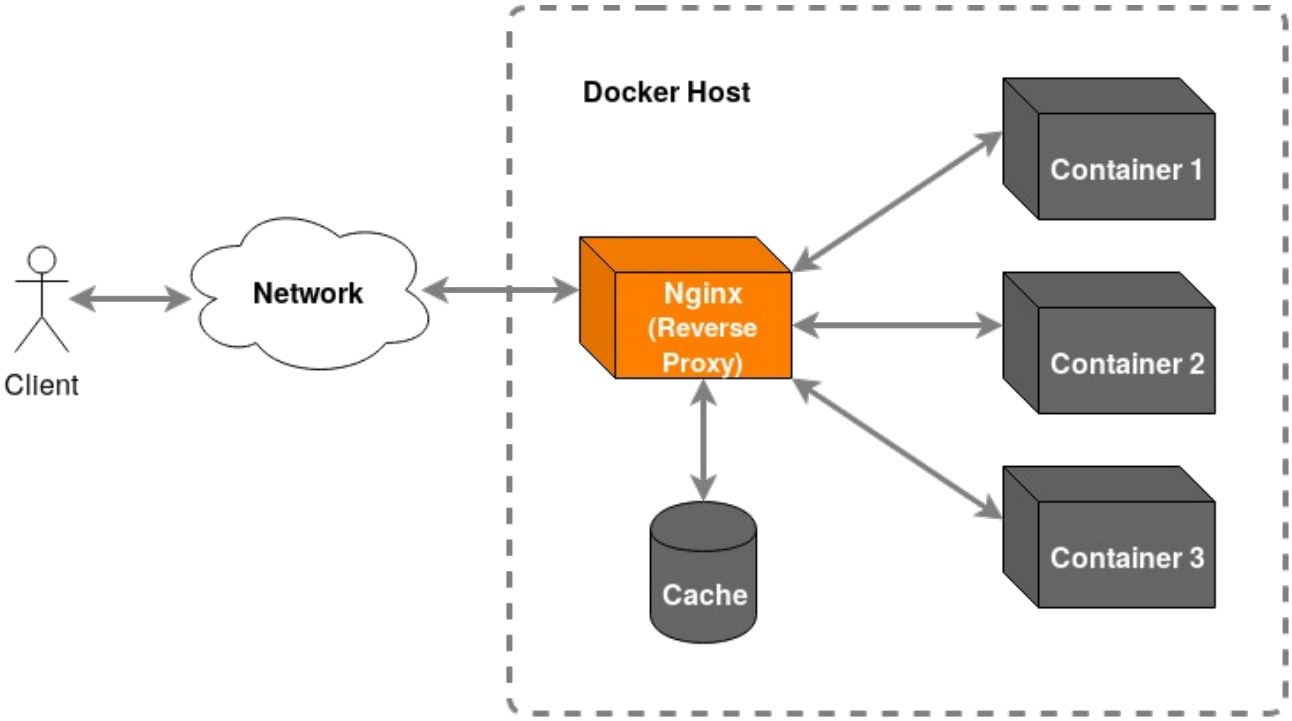

Abbildung 1: Systemkontext des Docker-Hosts

Konkret werden in dieser Umgebung beim AWI zwei Instanzen des Shibboleth IdPs (Test, Primär) und eine Instanz des Shibboleth SPs (Test) betrieben. Diese sogenannten Services sind in der Datei „docker-compose.yml“ beschrieben. Für das Service-Handling wurden Bash-Skripte („bin/“) geschrieben, die sowohl ein Starten als auch ein Stoppen der gesamten Umgebung zulassen. Die weiter oben erwähnten Docker-Images für einen IdP bzw. SP werden mit Hilfe von Dockerfiles beschrieben und befinden sich im Verzeichnis „dockerfiles/“. Die hieraus erzeugten Images können natürlich auch in einer Registry abgelegt werden. Für die Multi-Dienstverwaltung innerhalb eines Containers wird das Programm „supervisor“ verwendet. Hiermit lassen sich komfortabel mehrere Prozesse in einer Container-Instanz aktiv halten. Abschließend gibt es dann noch weitere

Verzeichnisse („nginx/“, „shib-idp/“, „shib-idp1“, „shib-sp1“), in denen die jeweiligen Konfigurationsdateien der einzelnen Dienste platziert sind. Einen genaueren Überblick bietet das Unterkapitel Verzeichnisstruktur.

Verzeichnisstruktur

```
shib-docker/
├── bin                          # Skripte für das Service-Handling
│   ├── _clean.sh
│   ├── _create.sh
│   ├── start.sh
│   └── stop.sh
├── docker-compose.yml
├── dockerfiles                  # Dockerfiles für die spezifischen Container
│   ├── dockerfile_idp
│   └── dockerfile_sp
├── nginx                        # Konfigurationsverzeichnis – Nginx
├── shib-idp                     # Konfigurationsverzeichnis – Shibboleth IdP
├── shib-idp1                    # Konfigurationsverzeichnis – Shibboleth IdP 1
└── shib-sp1                     # Konfigurationsverzeichnis – Shibboleth SP
```

Service-Handling

Start

Befehl	Beschreibung
sudo <DEPLOYMENT_PATH>/bin/start.sh	Skript zum Starten der Shibboleth-Umgebung
sudo <DEPLOYMENT_PATH>/bin/_update.sh	Skript zum Zurücksetzen von Dateiberechtigungen für die Protokollierung usw. (muss nicht separat aufgerufen werden)
docker-compose up -d	Erstellen und starten von Container im detached Modus
docker-compose start <SERVICE_NAME>	Startet einen einzelnen Service

Stop

Befehl	Beschreibung
sudo <DEPLOYMENT_PATH>/bin/stop.sh	Skript zum Stoppen der Shibboleth-Umgebung
docker-compose down	Stoppen und entfernen von Containern und Netzwerken
docker-compose down -v	Stoppen und Entfernen von Containern, Netzwerken und Volumes (Löscht das

	persistente Datenbank-Volume!)
docker-compose stop <SERVICE_NAME>	Stoppt einen einzelnen Service

Update

Befehl	Beschreibung
sudo <DEPLOYMENT_PATH>/bin/start.sh update	Skript zum Starten der Shibboleth-Umgebung (aktualisiert)
docker-compose pull	Service-Images pullen
docker pull <TAG>:<VERSION>	Pullen von einem Image aus einer Registry
docker-compose up -d --force-recreate --build	Erstellen und starten von Container im detached Modus. Container werden neu erstellt, auch wenn sich ihre Konfiguration und ihr Image nicht geändert haben. Erstellt Images, bevor die Container gestartet werden.

Clean up

Befehl	Beschreibung
sudo <DEPLOYMENT_PATH>/bin/start.sh clean	Skript zum Starten der Shibboleth-Umgebung (bereinigt)
sudo <DEPLOYMENT_PATH>/bin/stop.sh clean	Skript zum Stoppen der Shibboleth-Umgebung (bereinigt)
sudo <DEPLOYMENT_PATH>/bin/_clean.sh	Bereinigen von temporären Dateien, z. B. logs (Muss nicht separat aufgerufen werden)
docker volume rm shib-docker_shib-idp_mysql-data	Volume shib-idp_mysql-data entfernen (Löscht persistente DB-Daten!)
docker volume rm shib-docker_shib-idp1_mysql-data	Volume shib-idp1_mysql-data entfernen (Löscht persistente DB-Daten!)
docker image prune -f	Nicht verwendete Images entfernen (erzwungen)
docker system prune -f	Alle nicht verwendeten Daten entfernen (erzwungen)

Status

Befehl	Beschreibung
docker-compose logs -f	Anzeige der Ausgabe von Containern
docker-compose ps	Container auflisten
docker-compose top	Laufende Prozesse anzeigen
docker stats	Live-Stream mit Statistiken zur

	Ressourcennutzung von Containern anzeigen

Nginx als Reverse-Proxy

Für eine Grundkonfiguration werden nur folgende Dinge benötigt:

- Zuordnung von server_name:ports zu container:ports
- Zuordnen einer Konfigurationsdatei zur Standard-Nginx-Konfigurationsdatei unter /etc/nginx/nginx.conf
- Die Nginx-Konfiguration (s. nginx.conf)
- Aktivieren der DNS-(Aliase) und Firewall-Regeln

Als Basis-Image wird das leichtgewichtige Alpine Linux „nginx:alpine“ verwendet und das Port-Mapping mit der Host-VM sieht die Ports 80, 443 und 8443 vor. Wichtig ist, dass in der Server-Konfiguration „Default-Server“ aktiviert werden, um fehlerhafte Hostname-Zugriffe abzulehnen und um zu verhindern, dass nicht der erste funktionstüchtige „server{}“-Block in der Nginx-Konfiguration greift. Außerdem sollten die TLS-Zertifikate der einzelnen Dienste Nginx zur Verfügung gestellt werden. Das Default-Zertifikat wurde selbst signiert, aber es taucht in den gängigen Anwendungsfällen nicht auf. Nginx braucht die gesamte Zertifikatskette als „bundle“ für ein öffentliche TLS-Zertifikat.

Für das Logging der einzelnen Dienste und u. a. Fail2ban ist es notwendig, dass die „wirkliche“ IP-Adresse „durchgereicht“ wird. Hierfür muss am Apache-Server mit Hilfe des Moduls „remoteip“ folgende Einstellung in den einzelnen Seitenkonfigurationen vorgenommen werden:

```
RemoteIPHeader X-Real-IP
RemoteIPInternalProxy 10.0.0.0/24
```

Es ergibt sich nachstehende Verzeichnisstruktur für die Konfiguration und das Logging:

```
nginx/
├── certs                                  # Default-Zertifikat
│   ├── cert.pem
│   └── key.pem
├── conf                                   # Nginx-Konfiguration
│   └── nginx.conf
└── logs                                   # Log-Verzeichnis für den Reverse-Proxy
    └── nginx
        ├── access.log
        ├── error.log
        ├── shib-idp1.awi.de.access.log
        ├── shib-idp.awi.de.access.log
        └── shib-sp1.awi.de.access.log
```

Shibboleth Service Provider

Als Basis-Image kommt Ubuntu 18.04 LTS (bionic) zum Einsatz. Grundlegend werden die folgenden Pakete installiert:

- ntp
- curl
- sudo
- openssl
- supervisor

Des Weiteren werden die nachstehenden Software-Pakete installiert und über die Volumes konfiguriert:

- Apache HTTP Server
  - apache2
  - libapache2-mod-php
- Shibboleth Service Provider
  - switchaai_repository_uri=https://pkg.switch.ch/switchaai/ubuntu/dists/bionic/main/binary-all/misc/
  - switchaai_repository_package=switchaai-apt-source_1.0.0ubuntu1_all.deb
  - shibboleth

Es ergibt sich nachstehende Verzeichnisstruktur für die Konfiguration und das Logging:

```
shib-sp1/
├── apache                                  # Apache-Konfiguration
│   ├── html
│   │   └── index.php                       # phpinfo()-Aufruf
│   └── sites-enabled
│       ├── 001-http.conf
│       └── 002-https.conf
├── certs                                   # Zertifikate
│   ├── ca.crt.pem
│   ├── dfn-aai.g2.pem
│   ├── shib-sp1.awi.de.crt.bundle.pem
│   ├── shib-sp1.awi.de.crt.pem
│   └── shib-sp1.awi.de.key.pem
├── logs                                    # Log-Verzeichnis für den Shibboleth SP
│   ├── apache
│   │   ├── access.log
│   │   ├── error.log
│   │   └── other_vhosts_access.log
```

```
│   └── shibboleth
│       ├── shibd.log
│       ├── shibd_warn.log
│       ├── signature.log
│       └── transaction.log
└── shibboleth                                  # Shibboleth SP-Konfiguration
    ├── attribute-map.xml
    ├── attribute-policy.xml
    ├── protocols.xml
    ├── security-policy.xml
    └── shibboleth2.xml
```

Shibboleth Identity Provider

Als Basis-Image kommt Ubuntu 18.04 LTS (bionic) zum Einsatz. Grundlegend werden die folgenden Pakete installiert:

- ntp
- curl
- sudo
- openssl
- supervisor
- openjdk=openjdk-8-jdk

Des Weiteren werden die nachstehenden Software-Pakete installiert und über die Volumes konfiguriert:

- Apache Tomcat - Servlet and JSP Engine
  - tomcat=tomcat8
  - jstl_repository_uri=https://build.shibboleth.net/nexus/service/local/repositories/thirdparty/content/javax/servlet/jstl/1.2/
  - jstl_repository_package=jstl-1.2.jar
- Apache HTTP Server
  - apache2
- MySQL Database Server
  - mysql-server
  - mysql-client
  - libmysql-java

- Shibboleth Identity Provider
  - shibboleth_repository_uri=http://shibboleth.net/downloads/identity-provider/3.4.4/
  - shibboleth_repository_package=shibboleth-identity-provider-3.4.4.tar.gz
- Fail2ban
  - fail2ban

Fail2ban ist so konfiguriert, dass bei mehrmaligen falschen Anmeldeversuchen (5) die Nutzer-Ip-Adresse gesperrt wird und der Zugriff nur noch auf eine Blockierungsseite durch den Apache-Server gestattet wird. Durch das Modul „rewrite“ und die nachstehende Seitenkonfiguration gelingt es:

```
RewriteEngine on
RewriteMap hosts-deny "txt:/etc/apache2/conf-enabled/shib.deny"
RewriteCond "${hosts-deny:%{REMOTE_ADDR}|NOT-FOUND}" "!=NOT-FOUND"
RewriteCond %{REQUEST_URI} !^/idp/+(css/*|images/*|tou.jsp)
RewriteRule ^(.*)$ %{DOCUMENT_ROOT}/block/index.html [L]

<Directory "/opt/shibboleth-idp/htdocs/block">
      Options Indexes FollowSymLinks MultiViews
      AllowOverride FileInfo
      Require all granted
</Directory>
```

Es ergibt sich nachstehende Verzeichnisstruktur für die Konfiguration und das Logging:

```
shib-idp
├── apache                                  # Apache-Konfiguration
│   ├── htdocs
│   │   ├── block
│   │   │   └── index.html                  # Fail2ban-Blockierungsseite
│   │   ├── favicon.ico
│   │   └── robots.txt
│   └── sites-enabled
│       ├── 001-http.conf
│       ├── 002-https.conf
│       └── 003-https-8443.conf
├── certs                                   # Zertifikate
│   ├── ca.crt.pem
│   ├── dfn-aai.g2.pem
│   ├── shib-idp.awi.de.crt.bundle.pem
│   ├── shib-idp.awi.de.crt.pem
│   └── shib-idp.awi.de.key.pem
├── fail2ban                                # Fail2ban-Konfiguration
│   ├── action.d
│   │   └── idp.conf
```

```
    ├── filter.d
    │   └── idp.conf
    └── jail.local
├── logs                                  # Log-Verzeichnis für den Shibboleth IdP
    ├── apache
    │   ├── access.log
    │   ├── error.log
    │   └── other_vhosts_access.log
    ├── fail2ban
    │   └── fail2ban.log
    ├── mysql
    │   └── error.log
    ├── shibboleth
    │   ├── idp-audit.log
    │   ├── idp-consent-audit.log
    │   ├── idp-process.log
    │   └── idp-warn.log
    └── tomcat
        ├── catalina.$.log
        ├── localhost.$.log
        └── localhost_access_log.$.txt
├── mysql                                 # MySQL-Konfiguration
    ├── mysqld_init.sh
    └── shibboleth.sql
├── shibboleth                            # Shibboleth IdP-Konfiguration
    ├── bin
    │   └── update-sealer.sh
    ├── conf
    │   ├── access-control.xml
    │   ├── admin
    │   │   ├── general-admin.xml
    │   │   └── metrics.xml
    │   ├── attribute-filter.xml
    │   ├── attribute-resolver-full.xml
    │   ├── attribute-resolver-ldap.xml
    │   ├── attribute-resolver.xml
    │   ├── audit.xml
    │   ├── authn
    │   │   ├── authn-comparison.xml
    │   │   ├── authn-events-flow.xml
    │   │   ├── duo-authn-config.xml
    │   │   ├── duo.properties
    │   │   ├── external-authn-config.xml
    │   │   ├── function-authn-config.xml
    │   │   ├── general-authn.xml
    │   │   ├── ipaddress-authn-config.xml
    │   │   ├── jaas-authn-config.xml
    │   │   └── jaas.config
```


krb5-authn-config.xml
ldap-authn-config.xml
mfa-authn-config.xml
password-authn-config.xml
remoteuser-authn-config.xml
remoteuser-internal-authn-config.xml
spnego-authn-config.xml
x509-authn-config.xml
x509-internal-authn-config.xml
c14n
attribute-sourced-subject-c14n-config.xml
simple-subject-c14n-config.xml
subject-c14n-events-flow.xml
subject-c14n.xml
x500-subject-c14n-config.xml
cas-protocol.xml
credentials.xml
errors.xml
global.xml
idp.properties
intercept
consent-intercept-config.xml
context-check-intercept-config.xml
expiring-password-intercept-config.xml
impersonate-intercept-config.xml
intercept-events-flow.xml
profile-intercept.xml
ldap.properties
logback.xml
metadata-providers.xml
relying-party.xml
saml-nameid.properties
saml-nameid.xml
services.properties
services.xml
session-manager.xml
cron.d
shibboleth-idp
edit-webapp
css
consent.css
logout.css
main.css
images
dummylogo-mobile.png
dummylogo.png
failure-32x32.png
logo.png

```
        │   │   │   ├── net_Logo_Farbe.jpg
        │   │   │   └── success-32x32.png
        │   │   ├── tou.jsp
        │   │   └── WEB-INF
        │   │       ├── classes
        │   │       └── lib
        │   ├── messages
        │   │   ├── messages_de.properties
        │   │   └── messages.properties
        │   ├── metadata
        │   │   ├── dfn-aai-edugain+sp-metadata.xml
        │   │   ├── dfn-aai-sp-metadata.xml
        │   │   ├── dfn-aai-test-metadata.xml
        │   │   ├── idp-metadata.xml
        │   │   └── localMetadataProvider
        │   ├── system
        │   │   └── messages
        │   │       └── messages_de.properties
        │   └── views
        │       ├── admin
        │       │   └── unlock-keys.vm
        │       ├── client-storage
        │       │   ├── client-storage-read.vm
        │       │   └── client-storage-write.vm
        │       ├── duo.vm
        │       ├── error.vm
        │       ├── intercept
        │       │   ├── attribute-release.vm
        │       │   ├── expiring-password.vm
        │       │   ├── impersonate.vm
        │       │   └── terms-of-use.vm
        │       ├── login-error.vm
        │       ├── login.vm
        │       ├── logout-complete.vm
        │       ├── logout-propagate.vm
        │       ├── logout.vm
        │       ├── spnego-unavailable.vm
        │       ├── user-prefs.js
        │       └── user-prefs.vm
        └── tomcat                                  # Tomcat-Konfiguration
            ├── bin
            │   └── setenv.sh
            ├── Catalina
            │   └── localhost
            │       └── idp.xml
            ├── context.xml
            └── server.xml
```

Anhang

bin/

_clean.sh

```
#!/bin/bash

SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_FOLDER_PATH=$(dirname "$SCRIPT_PATH")
DEPLOYMENT_PATH=$(dirname "$SCRIPT_FOLDER_PATH")

if [ "$EUID" -ne 0 ]
then
        echo "Please run as root!"
        exit
fi

if [ ! -d "$DEPLOYMENT_PATH" ] || [ ! -f "$DEPLOYMENT_PATH/docker-compose.yml" ]
then
      echo "ERROR - Corrupt deployment!"
      exit
fi

# logrotate
echo 'Clean logs directories ...'
rm -rf $DEPLOYMENT_PATH/*/logs/*/*

# shib-idp
echo 'shib-idp - Clean dfn-aai* metadata ...'
rm -rf $DEPLOYMENT_PATH/shib-idp/shibboleth/metadata/dfn-aai*

# shib-idp1
echo 'shib-idp1 - Clean dfn-aai* metadata ...'
rm -rf $DEPLOYMENT_PATH/shib-idp1/shibboleth/metadata/dfn-aai*
```

_create.sh

```
#!/bin/bash

SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_FOLDER_PATH=$(dirname "$SCRIPT_PATH")
DEPLOYMENT_PATH=$(dirname "$SCRIPT_FOLDER_PATH")

if [ "$EUID" -ne 0 ]
then
```

```
        echo "Please run as root!"
        exit
fi

if [ ! -d "$DEPLOYMENT_PATH" ] || [ ! -f "$DEPLOYMENT_PATH/docker-compose.yml" ]
then
        echo "ERROR - Corrupt deployment!"
        exit
fi



# shib-sp1
echo 'Configure logs directory of shib-sp1 ...'
mkdir -p $DEPLOYMENT_PATH/shib-sp1/logs/apache/
mkdir -p $DEPLOYMENT_PATH/shib-sp1/logs/shibboleth/

chmod -R 777 $DEPLOYMENT_PATH/shib-sp1/logs/

chown -R 0:4 $DEPLOYMENT_PATH/shib-sp1/logs/apache/ # root:adm
chown -R 102:103 $DEPLOYMENT_PATH/shib-sp1/logs/shibboleth/ # _shibd:_shibd

# shib-idp
echo 'Configure logs directory of shib-idp ...'
mkdir -p $DEPLOYMENT_PATH/shib-idp/logs/apache/
mkdir -p $DEPLOYMENT_PATH/shib-idp/logs/tomcat/
mkdir -p $DEPLOYMENT_PATH/shib-idp/logs/mysql/
mkdir -p $DEPLOYMENT_PATH/shib-idp/logs/shibboleth/
mkdir -p $DEPLOYMENT_PATH/shib-idp/logs/fail2ban/
mkdir -p $DEPLOYMENT_PATH/shib-idp/shibboleth/metadata/

touch $DEPLOYMENT_PATH/shib-idp/logs/shibboleth/idp-process.log

chmod -R 777 $DEPLOYMENT_PATH/shib-idp/logs/
chmod -R 777 $DEPLOYMENT_PATH/shib-idp/shibboleth/metadata/

chown -R 0:4 $DEPLOYMENT_PATH/shib-idp/logs/apache/ # root:adm
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp/logs/tomcat/ # tomcat8:tomcat8
chown -R 104:105 $DEPLOYMENT_PATH/shib-idp/logs/mysql/ # mysql:mysql
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp/logs/shibboleth/ # tomcat8:tomcat8
chown -R 0:4 $DEPLOYMENT_PATH/shib-idp/logs/fail2ban/ # root:adm
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp/shibboleth/metadata/ # tomcat8:tomcat8

# shib-idp1
echo 'Configure logs directory of shib-idp1 ...'
mkdir -p $DEPLOYMENT_PATH/shib-idp1/logs/apache/
mkdir -p $DEPLOYMENT_PATH/shib-idp1/logs/tomcat/
mkdir -p $DEPLOYMENT_PATH/shib-idp1/logs/mysql/
mkdir -p $DEPLOYMENT_PATH/shib-idp1/logs/shibboleth/
```

```
mkdir -p $DEPLOYMENT_PATH/shib-idp1/logs/fail2ban/
mkdir -p $DEPLOYMENT_PATH/shib-idp1/shibboleth/metadata/

touch $DEPLOYMENT_PATH/shib-idp1/logs/shibboleth/idp-process.log

chmod -R 777 $DEPLOYMENT_PATH/shib-idp1/logs/
chmod -R 777 $DEPLOYMENT_PATH/shib-idp1/shibboleth/metadata/

chown -R 0:4 $DEPLOYMENT_PATH/shib-idp1/logs/apache/ # root:adm
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp1/logs/tomcat/ # tomcat8:tomcat8
chown -R 104:105 $DEPLOYMENT_PATH/shib-idp1/logs/mysql/ # mysql:mysql
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp1/logs/shibboleth/ # tomcat8:tomcat8
chown -R 0:4 $DEPLOYMENT_PATH/shib-idp1/logs/fail2ban/ # root:adm
chown -R 103:103 $DEPLOYMENT_PATH/shib-idp1/shibboleth/metadata/ # tomcat8:tomcat8
```

start.sh

```
#!/bin/bash

SCRIPT_PATH=$(readlink -f "$0")
SCRIPT_FOLDER_PATH=$(dirname "$SCRIPT_PATH")
DEPLOYMENT_PATH=$(dirname "$SCRIPT_FOLDER_PATH")

ACTION="$1"

DOCKER_UBUNTU="ubuntu:18.04"

if [ "$EUID" -ne 0 ]
then
	echo "Please run as root!"
	exit
fi

if [ ! -d "$DEPLOYMENT_PATH" ] || [ ! -f "$DEPLOYMENT_PATH/docker-compose.yml" ] ||
[ ! -f "$SCRIPT_FOLDER_PATH/_create.sh" ] || [ ! -f "$SCRIPT_FOLDER_PATH/_clean.sh" ]
then
	echo "ERROR - Corrupt deployment!"
	exit
fi

if [ -z "$ACTION" ]
then
	  echo "Start the Shibboleth environment ..."
	  $SCRIPT_FOLDER_PATH/_create.sh
	  docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml up -d

elif [ "$ACTION" == "update" ]
```

```
then
        echo "Start the Shibboleth environment (updated) ..."
    $SCRIPT_FOLDER_PATH/_create.sh
        docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml pull
        docker pull $DOCKER_UBUNTU
        docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml up -d --force-recreate --
build

elif [ "$ACTION" == "clean" ]
then
        echo "Start the Shibboleth environment (cleaned) ..."

        read -p "Delete all log files? [y,N] " input
        input=${input:-N}
        if [ "$input" == y ]
        then
                $SCRIPT_FOLDER_PATH/_clean.sh
        fi

    read -p "Remove all unused docker data? [y,N] " input
    input=${input:-N}
    if [ "$input" == y ]
    then
         docker system prune -f
    fi

        read -p "Delete persistent DB? [y,N] " input
    input=${input:-N}
    if [ "$input" == y ]
    then
        docker volume rm shib-docker_shib-idp_mysql-data
                docker volume rm shib-docker_shib-idp1_mysql-data
        fi

        $SCRIPT_FOLDER_PATH/_create.sh
        docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml up -d

else
        echo "Run $0 [update,clean]!"
        exit
fi
```

stop.sh

```
#!/bin/bash

SCRIPT_PATH=$(readlink -f "$0")
```

```
SCRIPT_FOLDER_PATH=$(dirname "$SCRIPT_PATH")
DEPLOYMENT_PATH=$(dirname "$SCRIPT_FOLDER_PATH")

ACTION="$1"

if [ "$EUID" -ne 0 ]
then
	echo "Please run as root!"
	exit
fi

if [ ! -d "$DEPLOYMENT_PATH" ] || [ ! -f "$DEPLOYMENT_PATH/docker-compose.yml" ] ||
[ ! -f "$SCRIPT_FOLDER_PATH/_create.sh" ] || [ ! -f "$SCRIPT_FOLDER_PATH/_clean.sh" ]
then
	echo "ERROR - Corrupt deployment!"
	exit
fi

if [ -z "$ACTION" ]
then
	echo "Stop the Shibboleth environment ..."
	docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml down

elif [ "$ACTION" == "clean" ]
then
	echo "Stop the Shibboleth environment (cleaned) ..."

	read -p "Delete all log files? [y,N] " input
	input=${input:-N}
	if [ "$input" == y ]
	then
		$SCRIPT_FOLDER_PATH/_clean.sh
	fi

	read -p "Remove all unused docker data? [y,N] " input
	input=${input:-N}
	if [ "$input" == y ]
	then
		docker system prune -f
	fi

	read -p "Delete persistent DB? [y,N] " input
	input=${input:-N}
	if [ "$input" == y ]
	then
		docker volume rm shib-docker_shib-idp_mysql-data
			docker volume rm shib-docker_shib-idp1_mysql-data
	fi
```

```
    $SCRIPT_FOLDER_PATH/_create.sh
    docker-compose -f $DEPLOYMENT_PATH/docker-compose.yml down

else
    echo "Run $0 [clean]!"
    exit
fi
```

docker-compose.yml

```
version: '2.4'
services:
    nginx_reverse_proxy:
        depends_on:
            - shib-sp1
            - shib-idp
            - shib-idp1
        image: nginx:alpine
        volumes:
            - ./nginx/conf/nginx.conf:/etc/nginx/nginx.conf:ro
            - ./nginx/logs/nginx/:/etc/nginx/logs/:rw
            - ./nginx/certs/:/etc/nginx/conf.d/nginx_certs/:ro
            - ./shib-sp1/certs/:/etc/nginx/conf.d/shib-sp1_certs/:ro
            - ./shib-idp/certs/:/etc/nginx/conf.d/shib-idp_certs/:ro
            - ./shib-idp1/certs/:/etc/nginx/conf.d/shib-idp1_certs/:ro
        ports:
            - 80:80
            - 443:443
            - 8443:8443
        networks:
            - shib-net
        restart: always

    shib-sp1:
        image: awi/shib-docker_sp
        build:
            context: ./dockerfiles
            dockerfile: dockerfile_sp
            args:
                - ubuntu_version=18.04
                - switchaai_repository_uri=https://pkg.switch.ch/switchaai/ubuntu/dists/bionic/
main/binary-all/misc/
                - switchaai_repository_package=switchaai-apt-source_1.0.0ubuntu1_all.deb
        volumes:
            - ./shib-sp1/apache/html/:/var/www/html/:ro
```

```
                - ./shib-sp1/apache/sites-enabled/:/etc/apache2/sites-enabled/:ro
                - ./shib-sp1/logs/apache/:/var/log/apache2/:rw
                - ./shib-sp1/shibboleth/attribute-map.xml:/etc/shibboleth/attribute-map.xml:ro
                - ./shib-sp1/shibboleth/attribute-policy.xml:/etc/shibboleth/attribute-policy.xml:ro
                - ./shib-sp1/shibboleth/shibboleth2.xml:/etc/shibboleth/shibboleth2.xml:ro
                - ./shib-sp1/logs/shibboleth/:/var/log/shibboleth/:rw
                - ./shib-sp1/certs/:/etc/ssl/_certs/:ro
        networks:
                - shib-net
        restart: always

    shib-idp:
        image: awi/shib-docker_idp
        build:
                context: ./dockerfiles
                dockerfile: dockerfile_idp
                args:
                        - ubuntu_version=18.04
                        - openjdk=openjdk-8-jdk
                        - openjdk_home=/usr/lib/jvm/java-8-openjdk-amd64
                        - tomcat=tomcat8
                        - jstl_repository_uri=https://build.shibboleth.net/nexus/service/local/
repositories/thirdparty/content/javax/servlet/jstl/1.2/
                        - jstl_repository_package=jstl-1.2.jar
                        - shibboleth_repository_uri=http://shibboleth.net/downloads/identity-provider/
3.4.4/
                        - shibboleth_repository_package=shibboleth-identity-provider-3.4.4.tar.gz
        mem_reservation: 4gb
        volumes:
                - ./shib-idp/tomcat/bin/setenv.sh:/var/lib/tomcat8/bin/setenv.sh:ro
                - ./shib-idp/tomcat/server.xml:/etc/tomcat8/server.xml:ro
                - ./shib-idp/tomcat/Catalina/localhost/idp.xml:/etc/tomcat8/Catalina/localhost/
idp.xml:ro
                - ./shib-idp/tomcat/context.xml:/etc/tomcat8/context.xml:ro
                - ./shib-idp/logs/tomcat/:/var/log/tomcat8/:rw
                - ./shib-idp/apache/htdocs/:/opt/shibboleth-idp/htdocs/:ro
                - ./shib-idp/apache/sites-enabled/:/etc/apache2/sites-enabled/:ro
                - ./shib-idp/logs/apache/:/var/log/apache2/:rw
                - ./shib-idp/mysql/:/tmp/mysql/:ro
                - shib-idp_mysql-data:/var/lib/mysql/
                - ./shib-idp/logs/mysql/:/var/log/mysql/:rw
                - ./shib-idp/shibboleth/bin/update-sealer.sh:/opt/shibboleth-idp/bin/update-
sealer.sh:ro
                - ./shib-idp/shibboleth/conf/:/opt/shibboleth-idp/conf/:ro
                - ./shib-idp/shibboleth/cron.d/shibboleth-idp:/etc/cron.d/shibboleth-idp:ro
                - ./shib-idp/shibboleth/edit-webapp/:/opt/shibboleth-idp/edit-webapp/:ro
                - ./shib-idp/shibboleth/messages/:/opt/shibboleth-idp/messages/:ro
                - ./shib-idp/shibboleth/metadata/:/opt/shibboleth-idp/metadata/:rw
```

```
            - ./shib-idp/shibboleth/system/messages/messages_de.properties:/opt/shibboleth-
idp/system/messages/messages_de.properties:ro
            - ./shib-idp/shibboleth/views/:/opt/shibboleth-idp/views/:ro
            - ./shib-idp/logs/shibboleth/:/opt/shibboleth-idp/logs/:rw
            - ./shib-idp/fail2ban/jail.local:/etc/fail2ban/jail.local:ro
            - ./shib-idp/fail2ban/filter.d/idp.conf:/etc/fail2ban/filter.d/idp.conf:ro
            - ./shib-idp/fail2ban/action.d/idp.conf:/etc/fail2ban/action.d/idp.conf:ro
            - ./shib-idp/logs/fail2ban/:/var/log/fail2ban/:rw
            - ./shib-idp/certs/:/etc/ssl/_certs/:ro
        networks:
            - shib-net
        restart: always

    shib-idp1:
        image: awi/shib-docker_idp
        volumes:
            - ./shib-idp1/tomcat/bin/setenv.sh:/var/lib/tomcat8/bin/setenv.sh:ro
            - ./shib-idp1/tomcat/server.xml:/etc/tomcat8/server.xml:ro
            - ./shib-idp1/tomcat/Catalina/localhost/idp.xml:/etc/tomcat8/Catalina/localhost/
idp.xml:ro
            - ./shib-idp1/tomcat/context.xml:/etc/tomcat8/context.xml:ro
            - ./shib-idp1/logs/tomcat/:/var/log/tomcat8/:rw
            - ./shib-idp1/apache/htdocs/:/opt/shibboleth-idp/htdocs/:ro
            - ./shib-idp1/apache/sites-enabled/:/etc/apache2/sites-enabled/:ro
            - ./shib-idp1/logs/apache/:/var/log/apache2/:rw
            - ./shib-idp1/mysql/:/tmp/mysql/:ro
            - shib-idp1_mysql-data:/var/lib/mysql/
            - ./shib-idp1/logs/mysql/:/var/log/mysql/:rw
            - ./shib-idp/shibboleth/bin/update-sealer.sh:/opt/shibboleth-idp/bin/update-
sealer.sh:ro
            - ./shib-idp1/shibboleth/conf/:/opt/shibboleth-idp/conf/:ro
            - ./shib-idp1/shibboleth/cron.d/shibboleth-idp:/etc/cron.d/shibboleth-idp:ro
            - ./shib-idp1/shibboleth/edit-webapp/:/opt/shibboleth-idp/edit-webapp/:ro
            - ./shib-idp1/shibboleth/messages/:/opt/shibboleth-idp/messages/:ro
            - ./shib-idp1/shibboleth/metadata/:/opt/shibboleth-idp/metadata/:rw
            - ./shib-idp1/shibboleth/system/messages/messages_de.properties:/opt/shibboleth-
idp/system/messages/messages_de.properties:ro
            - ./shib-idp1/shibboleth/views/:/opt/shibboleth-idp/views/:ro
            - ./shib-idp1/logs/shibboleth/:/opt/shibboleth-idp/logs/:rw
            - ./shib-idp1/fail2ban/jail.local:/etc/fail2ban/jail.local:ro
            - ./shib-idp1/fail2ban/filter.d/idp.conf:/etc/fail2ban/filter.d/idp.conf:ro
            - ./shib-idp1/fail2ban/action.d/idp.conf:/etc/fail2ban/action.d/idp.conf:ro
            - ./shib-idp1/logs/fail2ban/:/var/log/fail2ban/:rw
            - ./shib-idp1/certs/:/etc/ssl/_certs/:ro
        networks:
            - shib-net
        restart: always
```

```
volumes:
  shib-idp_mysql-data:
  shib-idp1_mysql-data:

networks:
        shib-net:
                driver: bridge
                ipam:
                        driver: default
                        config:
                                - subnet: 10.0.0.0/24
```

dockerfiles/

dockerfile_idp

```
ARG ubuntu_version

FROM ubuntu:${ubuntu_version}

ARG openjdk
ARG openjdk_home
ARG tomcat
ARG jstl_repository_uri
ARG jstl_repository_package
ARG shibboleth_repository_uri
ARG shibboleth_repository_package



##
# Basic Packages
##

RUN apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install ntp curl sudo openssl ${openjdk}
supervisor

RUN mkdir -p /var/log/supervisor && \
    printf '%s\n%s\n' '[supervisord]' 'nodaemon=true' > /etc/supervisor/conf.d/supervisord.conf



##
# Certificates
##

RUN mkdir -p /etc/ssl/_certs/
```

```
##
# Apache Tomcat - Servlet and JSP Engine
##

RUN apt-get update && \
   DEBIAN_FRONTEND=noninteractive apt-get -y install ${tomcat}

ENV JAVA_HOME ${openjdk_home}
ENV CATALINA_HOME /usr/share/${tomcat}
ENV CATALINA_BASE /var/lib/${tomcat}
ENV PATH $CATALINA_HOME/bin:$PATH

RUN chown -R ${tomcat}:${tomcat} /var/log/${tomcat}/ && \
   chmod -R 644 /var/log/${tomcat}/ && \
   mkdir -p $CATALINA_BASE/lib/org/apache/catalina/util/ && \
   echo 'server.info=' >> $CATALINA_BASE/lib/org/apache/catalina/util/ServerInfo.properties

RUN cd /tmp && \
   curl --fail --remote-name ${jstl_repository_uri}/${jstl_repository_package} && \
   cp ${jstl_repository_package} /var/lib/${tomcat}/lib/

RUN printf '%s\n%s\n%s\n%s\n' "[program:${tomcat}]" 'priority=6' "user=${tomcat}"
'command=catalina.sh run' >> /etc/supervisor/conf.d/supervisord.conf


##
# Apache HTTP Server
##

RUN apt-get update && \
   DEBIAN_FRONTEND=noninteractive apt-get -y install apache2

RUN a2enmod rewrite && \
   a2enmod ssl && \
   a2enmod headers && \
   a2enmod proxy_ajp && \
   a2enmod remoteip

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_PID_FILE /var/run/apache2/apache2.pid

RUN mkdir -p $APACHE_RUN_DIR && \
```

```
    rm /var/www/html/index.html && \
    a2dissite 000-default.conf && \
    a2dissite default-ssl.conf

RUN chown -R $APACHE_RUN_USER:$APACHE_RUN_GROUP $APACHE_RUN_DIR /var/
www/html/ && \
    printf '%s\n%s\n' 'ServerSignature Off' 'ServerTokens Prod' >> /etc/apache2/apache2.conf

RUN printf '%s\n%s\n%s\n' '[program:apache2]' 'priority=5' 'command=/usr/sbin/apache2 -k start -
DFOREGROUND' >> /etc/supervisor/conf.d/supervisord.conf



##
# MySQL Database Server
##

RUN apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install mysql-server mysql-client libmysql-
java

ENV MYSQL_USER=mysql
ENV MYSQL_DATA_DIR=/var/lib/mysql
ENV MYSQL_RUN_DIR=/var/run/mysqld
ENV MYSQL_LOG_DIR=/var/log/mysql

RUN mkdir -p $MYSQL_RUN_DIR && \
    chown $MYSQL_USER:$MYSQL_USER $MYSQL_RUN_DIR && \
    ln -s /usr/share/java/mysql.jar /var/lib/${tomcat}/lib/mysql.jar

RUN printf '%s\n%s\n%s\n%s\n' '[program:mysqld]' 'priority=3' "user=$MYSQL_USER"
'command=/usr/sbin/mysqld' >> /etc/supervisor/conf.d/supervisord.conf

RUN printf '%s\n%s\n%s\n%s\n' '[program:mysqld_init]' 'priority=4' 'startsecs=0' 'command=/tmp/
mysql/mysqld_init.sh' >> /etc/supervisor/conf.d/supervisord.conf



##
# Shibboleth Identity Provider
##

RUN cd /tmp && \
    curl --fail --remote-name ${shibboleth_repository_uri}/${shibboleth_repository_package} && \
    tar -xzf ${shibboleth_repository_package} && \
    cd ${shibboleth_repository_package%%.*}*/ && \
    bin/install.sh -Didp.keystore.password=default -Didp.sealer.password=default -
Didp.host.name=localhost.localdomain

RUN rm -rf /opt/shibboleth-idp/credentials/sealer.* && \
```

```
    usermod -aG ssl-cert ${tomcat} && \
    chown -R ${tomcat}:${tomcat} /opt/shibboleth-idp/

RUN printf '%s\n%s\n%s\n%s\n%s\n' '[program:shib-idp_update-sealer]' 'priority=1' 'startsecs=0'
"user=${tomcat}" 'command=/opt/shibboleth-idp/bin/update-sealer.sh' >>
/etc/supervisor/conf.d/supervisord.conf

RUN printf '%s\n%s\n%s\n%s\n%s\n' '[program:shib-idp_build-war]' 'priority=2' 'startsecs=0'
"user=${tomcat}" 'command=/opt/shibboleth-idp/bin/build.sh -Didp.target.dir=/opt/shibboleth-idp'
>> /etc/supervisor/conf.d/supervisord.conf



##
# Fail2ban
##

RUN apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install fail2ban

RUN printf '%s\n%s\n%s\n%s\n%s\n%s\n%s\n' '##' '## shib.deny' '##' '## ATTENTION! This is a
map, not a list, even when we treat it as such.' '## mod_rewrite parses it for key/value pairs, so at
least a' '## dummy value "-" must be present for each entry.' '##' >
/etc/apache2/conf-enabled/shib.deny

RUN mkdir -p /var/run/fail2ban /var/log/fail2ban && \
    rm -rf /etc/fail2ban/jail.d/defaults-debian.conf && \
    touch /opt/shibboleth-idp/logs/idp-process.log && \
    sed -i 's/logtarget = \/var\/log\/fail2ban.log/logtarget = \/var\/log\/fail2ban\/fail2ban.log/g'
/etc/fail2ban/fail2ban.conf && \
    sed -i 's/\/var\/log\/fail2ban.log/\/var\/log\/fail2ban\/fail2ban.log/g' /etc/logrotate.d/fail2ban

RUN printf '%s\n%s\n%s\n%s\n' '[program:fail2ban]' 'priority=7' 'startsecs=0' 'command=/usr/bin/
fail2ban-server -b -x' >> /etc/supervisor/conf.d/supervisord.conf




RUN apt-get clean && apt-get -y autoremove && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*



EXPOSE 80 443 8443

ENTRYPOINT ["/usr/bin/supervisord", "-c", "/etc/supervisor/supervisord.conf"]
```

dockerfile_sp

```
ARG ubuntu_version
```

```
FROM ubuntu:${ubuntu_version}

ARG switchaai_repository_uri
ARG switchaai_repository_package


##
# Basic Packages
##

RUN apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install ntp curl sudo openssl supervisor

RUN mkdir -p /var/log/supervisor && \
    printf '%s\n%s\n' '[supervisord]' 'nodaemon=true' > /etc/supervisor/conf.d/supervisord.conf


##
# Certificates
##

RUN mkdir -p /etc/ssl/_certs/


##
# Apache HTTP Server
##

RUN apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install apache2 libapache2-mod-php

RUN a2enmod rewrite && \
    a2enmod ssl && \
    a2enmod headers && \
    a2enmod remoteip

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_PID_FILE /var/run/apache2/apache2.pid

RUN mkdir -p $APACHE_RUN_DIR && \
    rm /var/www/html/index.html && \
    a2dissite 000-default.conf && \
    a2dissite default-ssl.conf
```

```
RUN chown -R $APACHE_RUN_USER:$APACHE_RUN_GROUP $APACHE_RUN_DIR /var/
www/html/ && \
    printf '%s\n%s\n' 'ServerSignature Off' 'ServerTokens Prod' >> /etc/apache2/apache2.conf

RUN printf '%s\n%s\n' '[program:apache2]' 'command=/usr/sbin/apache2 -k start -
DFOREGROUND' >> /etc/supervisor/conf.d/supervisord.conf


##
# Shibboleth Service Provider
##

RUN cd /tmp && \
    curl --fail --remote-name ${switchaai_repository_uri}/${switchaai_repository_package} && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install ./${switchaai_repository_package}
&& \
    apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get -y install --install-recommends shibboleth

RUN chown -R _shibd:_shibd /etc/shibboleth && \
    mkdir -p /run/shibboleth && chown -R _shibd:_shibd /run/shibboleth


RUN printf '%s\n%s\n%s\n%s\n' '[program:shibd]' 'startsecs=0' "user=_shibd"
'command=/usr/sbin/shibd -f -c /etc/shibboleth/shibboleth2.xml -p /run/shibboleth/shibd.pid -w 30'
>> /etc/supervisor/conf.d/supervisord.conf



RUN apt-get clean && apt-get -y autoremove && \
    rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*


EXPOSE 80 443

ENTRYPOINT ["/usr/bin/supervisord", "-c", "/etc/supervisor/supervisord.conf"]
```

nginx/conf/

nginx.conf

```
worker_processes auto;

# [ debug | info | notice | warn | error | crit ]
error_log logs/error.log;
```

```
pid /var/run/nginx.pid;

events {
        worker_connections 1024;
}

http {

        sendfile on;

        log_format main '$remote_addr - $remote_user [$time_local]  $status '
                '"$request" $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';
        access_log logs/access.log main;

        proxy_set_header Host              $host;
        proxy_set_header X-Real-IP         $remote_addr;
        proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host  $server_name;
        proxy_set_header X-Forwarded-Port  $server_port;

        ssl_protocols TLSv1.1 TLSv1.2;
        ssl_prefer_server_ciphers on;
        ssl_ciphers 'EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM
EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384
EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !
LOW !3DES !MD5 !EXP !PSK !SRP !DSS !RC4';

        add_header Strict-Transport-Security 'max-age=31536000; includeSubDomains; preload';
        add_header X-Frame-Options DENY;
        add_header X-Content-Type-Options nosniff;
        add_header X-XSS-Protection '1; mode=block';

        server {
                server_name shib-sp1.awi.de;
                access_log logs/shib-sp1.awi.de.access.log main;

                location / {
                        proxy_pass https://shib-sp1;
                        proxy_redirect off;
                }

                listen 80;
                listen 443 ssl;
                ssl_certificate /etc/nginx/conf.d/shib-sp1_certs/shib-sp1.awi.de.crt.bundle.pem;
                ssl_certificate_key /etc/nginx/conf.d/shib-sp1_certs/shib-sp1.awi.de.key.pem;
```

```
    }

  server {
      server_name shib-idp.awi.de;
      access_log logs/shib-idp.awi.de.access.log main;

      location / {
          proxy_pass https://shib-idp;
          proxy_redirect off;
      }

      listen 80;
      listen 443 ssl;
      listen 8443 ssl;
      ssl_certificate /etc/nginx/conf.d/shib-idp_certs/shib-idp.awi.de.crt.bundle.pem;
      ssl_certificate_key /etc/nginx/conf.d/shib-idp_certs/shib-idp.awi.de.key.pem;
  }

  server {
      server_name shib-idp1.awi.de;
      access_log logs/shib-idp1.awi.de.access.log main;

      location / {
          proxy_pass https://shib-idp1;
          proxy_redirect off;
      }

      listen 80;
      listen 443 ssl;
          listen 8443 ssl;
      ssl_certificate /etc/nginx/conf.d/shib-idp1_certs/shib-idp1.awi.de.crt.bundle.pem;
      ssl_certificate_key /etc/nginx/conf.d/shib-idp1_certs/shib-idp1.awi.de.key.pem;
  }

    server {
          listen 80 default_server;
          server_name _;
          return 444;
    }

    server {
          listen 443 ssl default_server;
          server_name _;
          ssl_certificate /etc/nginx/conf.d/nginx_certs/cert.pem;
          ssl_certificate_key /etc/nginx/conf.d/nginx_certs/key.pem;
          return 444;
    }
}
```

shib-idp/fail2ban/

action.d/idp.conf

```
[Definition]
actionban = printf %%b "<ip> -\n" >> /etc/apache2/conf-enabled/shib.deny
actionunban = sed -i "/^<ip> -$/d" /etc/apache2/conf-enabled/shib.deny
```

filter.d/idp.conf

```
[Definition]

failregex = IP\:<HOST> .* Login by .* failed
            IP\:<HOST> .* No password available
ignoreregex =
```

jail.local

```
[idp]

enabled  = true
port     = http,https
filter   = idp
logpath  = /opt/shibboleth-idp/logs/idp-process.log
banaction = idp
maxretry = 5
```

shib-idp/mysql/

mysqld_init.sh

```
#!/bin/bash

WORKING_DIR=$(dirname $0)

if [ ! -d "/var/lib/mysql/shibboleth/" ]
then
        sleep 10
        mysql < $WORKING_DIR/shibboleth.sql
fi
```

shibboleth.sql

```
SET NAMES 'utf8';
SET CHARACTER SET utf8;
CHARSET utf8;
CREATE DATABASE IF NOT EXISTS shibboleth CHARACTER SET=utf8;
USE shibboleth;

CREATE TABLE IF NOT EXISTS shibpid (
    localEntity VARCHAR(255) NOT NULL,
    peerEntity VARCHAR(255) NOT NULL,
    persistentId VARCHAR(50) NOT NULL,
    principalName VARCHAR(50) NOT NULL,
    localId VARCHAR(50) NOT NULL,
    peerProvidedId VARCHAR(50) NULL,
    creationDate TIMESTAMP NOT NULL,
    deactivationDate TIMESTAMP NULL,
    PRIMARY KEY (localEntity, peerEntity, persistentId)
);

CREATE USER 'shibboleth'@'localhost' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON shibboleth.* TO 'shibboleth'@'localhost';

FLUSH PRIVILEGES;
```

shib-idp/shibboleth/bin/

update-sealer.sh

```
#!/bin/bash

# https://wiki.shibboleth.net/confluence/display/IDP30/SecretKeyManagement
# Pfad zur Java-Installation:
#export JAVA_HOME=/usr          # Debian und Ubuntu mit OpenjDK

set -e
set -u



# Default IDP_HOME if not already set
if [ ! -d "${IDP_HOME:=/opt/shibboleth-idp}" ]
then
    echo "ERROR: Directory does not exist: ${IDP_HOME}" >&2
    exit 1
fi
```

```
function get_config {
   # Key to lookup (escape . for regex lookup)
   local KEY=${1:?"No key provided to look up value"}
   # Passed default value
   local DEFAULT="${2:-}"
   # Lookup key, strip spaces, replace idp.home with IDP_HOME value
   local RESULT=$(sed -rn '/^'"${KEY//./\\.}"'\s*=/ { s|^[^=]*=(.*)\s*$|\1|; s|%\{idp\.home\}|"$
{IDP_HOME}"|g; p}' ${IDP_HOME}/conf/idp.properties)
   # Set if no result with default - exit if no default
   echo ${RESULT:-${DEFAULT:?"No value in config and no default defined for: '${KEY}'"}}
}

# Get config values
## Official config items ##
storefile=$(get_config idp.sealer.storeResource)
versionfile=$(get_config idp.sealer.versionResource)
storepass=$(get_config idp.sealer.storePassword)
alias=$(get_config idp.sealer.aliasBase secret)
## Extended config items ##
count=$(get_config idp.sealer._count 30)
# default cannot be empty - so "self" is the default (self is skipped for syncing)
sync_hosts=$(get_config idp.sealer._sync_hosts ${HOSTNAME})

# Run the keygen utility
${0%/*}/runclass.sh net.shibboleth.utilities.java.support.security.BasicKeystoreKeyStrategyTool \
   --storefile "${storefile}" \
   --storepass "${storepass}" \
   --versionfile "${versionfile}" \
   --alias "${alias}" \
   --count "${count}"

# Display current version
echo "INFO: $(tac "${versionfile}" | tr "\n" " ")" >&1

for EACH in ${sync_hosts}
do
   if [ "${HOSTNAME}" == "${EACH}" ]
   then
      echo "INFO: Host '${EACH}' is myself - skipping" >&1
   elif ! ping -q -c 1 -W 3 ${EACH} >/dev/null 2>&1
   then
      echo "ERROR: Host '${EACH}' not reachable - skipping" >&2
   else
      # run scp in the background
      scp "${storefile}" "${versionfile}" "${EACH}:${IDP_HOME}/credentials/" &
   fi
done
```