ZABBIX 5.0

SECURE CONNECTIONS TO DATABASE

Aleksandrs Petrovs-Gavrilovs
Technical Support Engineer

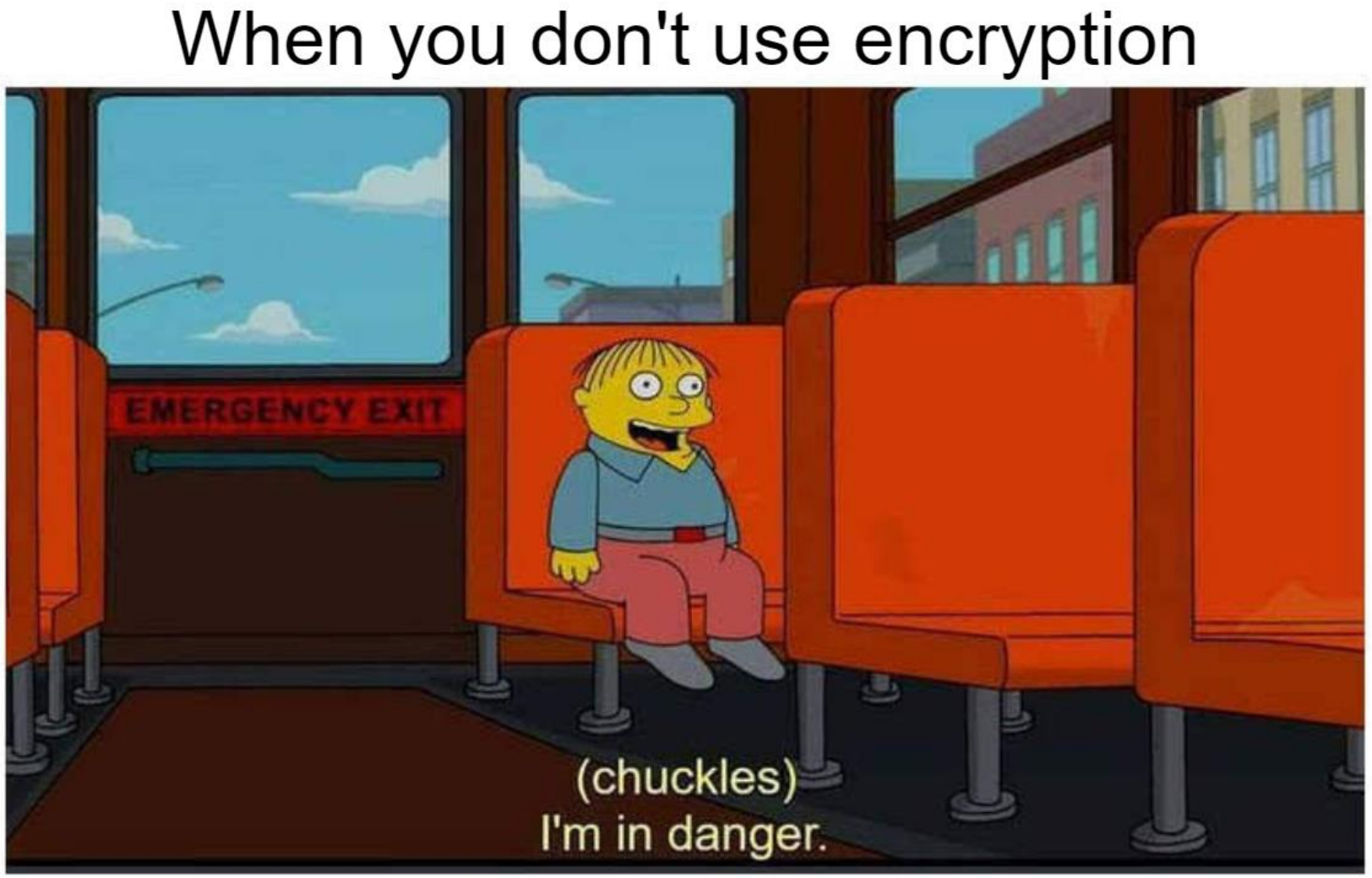
01

WHAT DOES “SECURE” MEAN?

- Avoid being harmed by any risk, danger, or threat.

- Cambridge Online Dictionary

WHAT IS ENCRYPTION?



WHAT IS ENCRYPTION?

- Process that converts the original information, known as plaintext, into an alternative form known as ciphertext.
- Authorized parties can decipher a ciphertext back to plaintext to access the original information.
- Encryption does not prevent interference but denies the intelligible content to an interceptor.
- Can be symmetric or asymmetric

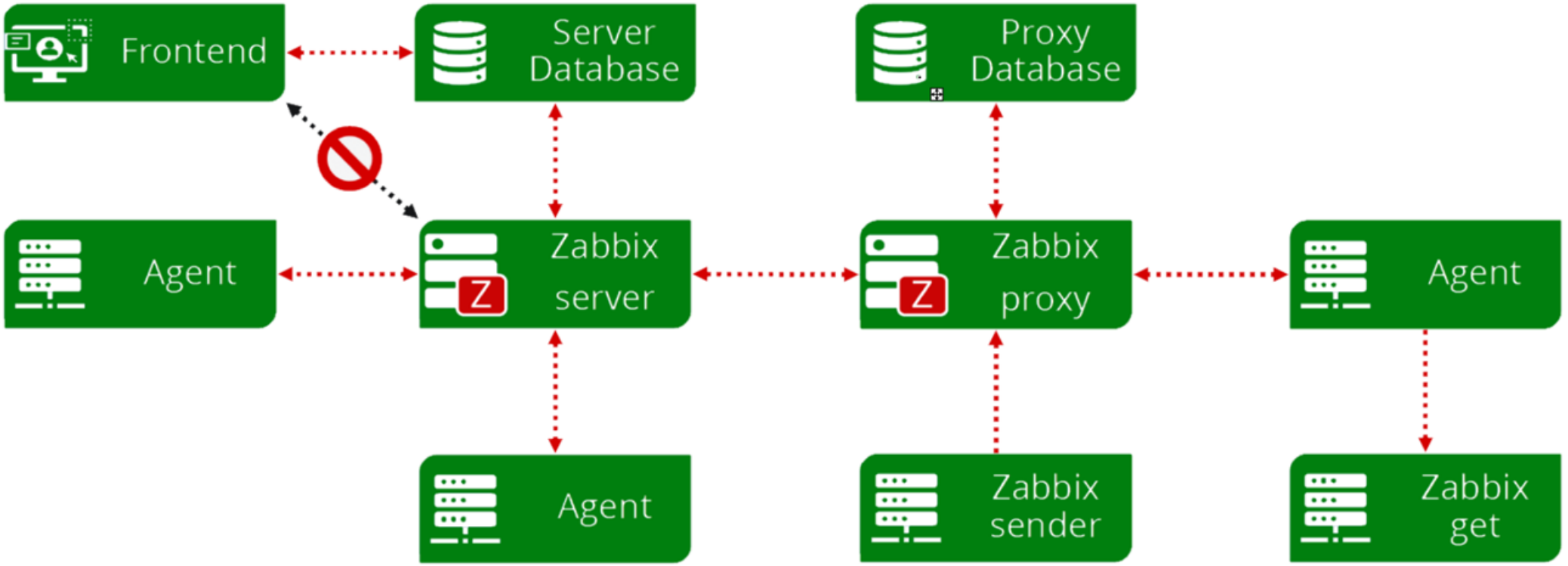
WHY DO YOU NEED ENCRYPTION?

- To keep data safe.
- To keep data secure.
- To keep data protected
- To keep data confidential.
- Etc.

ZABBIX AND ENCRYPTION

- Zabbix supports encrypted communications between Zabbix components using Transport Layer Security (TLS) protocol v.1.2 and 1.3 (depending on the crypto library).
- Certificate-based and pre-shared key-based encryption is supported.

ZABBIX AND ENCRYPTION



ENCRYPTION EXAMPLE

1. Generate PSK key and save to a file

```
# openssl rand -hex 32 > /etc/zabbix/keys/agent.psk
# chmod 400 /etc/zabbix/keys/agent.psk
# chown zabbix:zabbix /etc/zabbix/keys
```

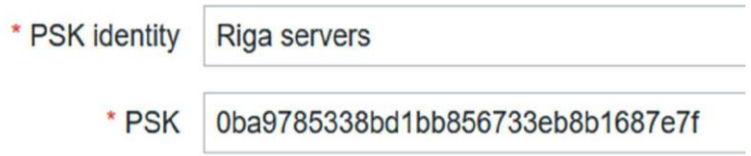
2. Change agent configuration

```
# TLSAccept=psk
# TLSPSKIdentity=Riga servers
# TLSPSKFile=/etc/zabbix/keys/agent.psk
```

3. Restart Zabbix agent

```
# systemctl restart zabbix-agent
```

4. Update front-end

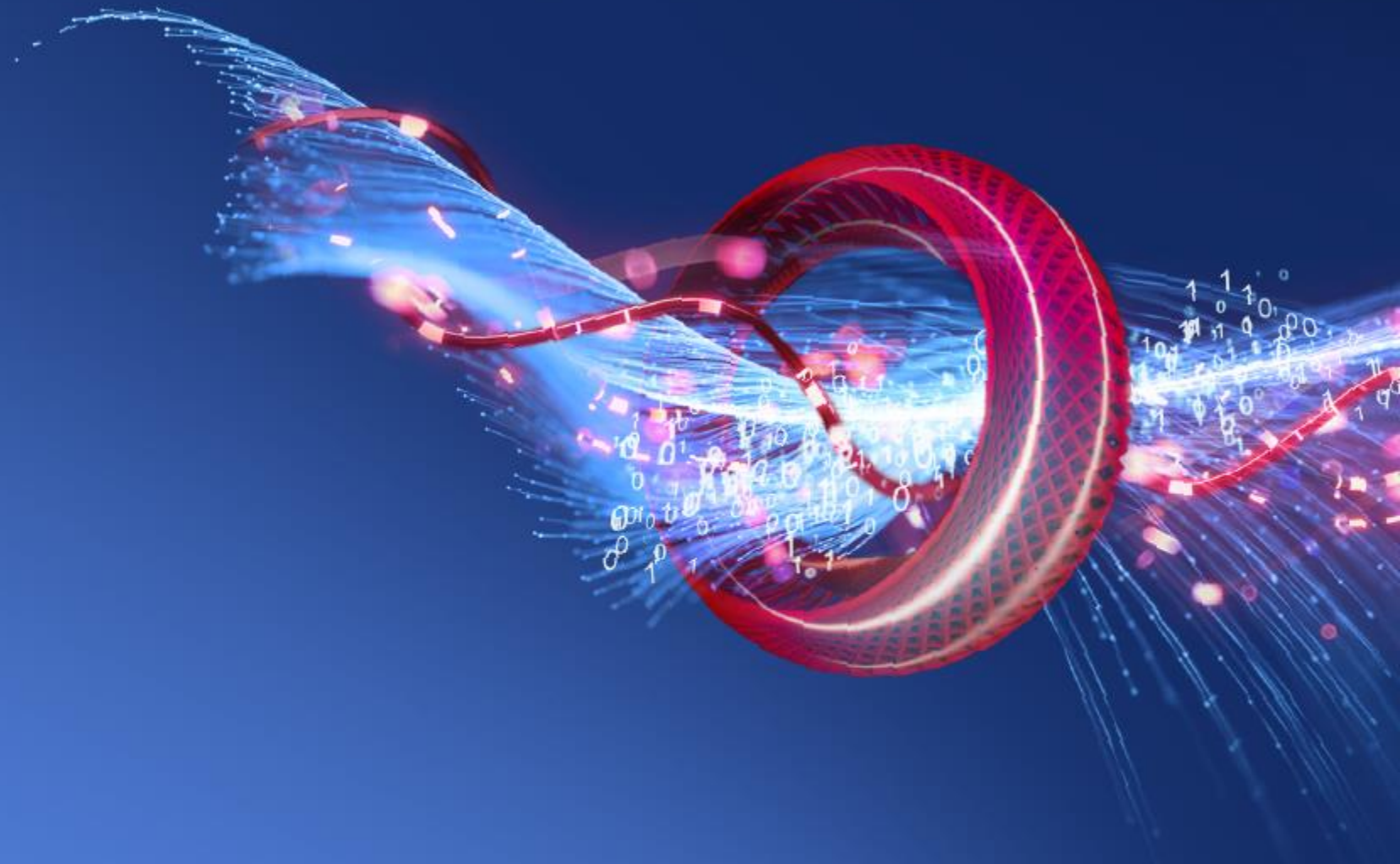


WHY SPECIFICALLY DATABASE?

Because data

02

SETUP



SETUP?


FOR MY NEXT TRICK
I'M GOING TO ENCRYPT CONNECTION
BETWEEN ZABBIX AND DB

SETUP

- Zabbix 5.0
- Centos 8 (x2 VMs)
- MySQL 8

03

HOW TO ENCRYPT CONNECTION BETWEEN DB AND ZABBIX

HOW TO?


Telling
you should
use encryption
Actually
showing
how to
configure it

PREPARE VMS

1. Make a plain installation of CentOS 8, decide which is for DB and which for Zabbix and edit hostnames of VMs

```
# vi /etc/hostname
```

2. Name, accordingly i.e.:

   a. `mydb.localhost.local`

   b. `cazabbix.localhost.local`

3. And update the hosts file on both VMs:

```
# vi /etc/hosts
```

```
192.168.3.92    cazabbix.localhost.local
192.168.3.86    mydb.localhost.local
```

CREATE A CERTIFICATE AUTHORITY

1. On the future Zabbix server check where openssl.cnf is located

```
# find / -iname openssl.cnf
```

`/etc/pki/tls/openssl.cnf` - By default

2. Find the CA location

```
# cat /etc/pki/tls/openssl.cnf | grep dir
dir               = /etc/pki/CA             # Where everything is kept
```

3. Create a subdirectory in it, i.e.

```
# mkdir /etc/pki/CA/private
```

CREATE A CERTIFICATE AUTHORITY

1. Create a pair of key and sign it (self-sign)

```
# openssl req  -new -x509 -keyout /etc/pki/CA/private/cakey.pem -out /etc/pki/CA/cacert.pem -days 3652 -newkey rsa:4096
```

2. You will be prompted with a request to create a password and fill in information about certificate owner

```
Generating a RSA private key
................++++
.....................++++
writing new private key to '/etc/pki/CA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:LV
State or Province Name (full name) []:Riga
Locality Name (eg, city) [Default City]:Riga
Organization Name (eg, company) [Default Company Ltd]:SIA ZABBIX
Organizational Unit Name (eg, section) []:SUPPORT
Common Name (eg, your name or your server's hostname) []:Root CA
Email Address []:
```

SIGNING REQUESTS

1. To make things a bit easier and less confusing, you can edit the openssl.cnf

```
# vi /etc/pki/tls/openssl.cnf
```

2. Changing the default organization values:

```
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = LV
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = Riga
#stateOrProvinceName_default    = Default Province

localityName                    = Riga
localityName_default            = Riga

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = SIA ZABBIX

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
#organizationalUnitName_default = SIA ZABBIX

commonName                      = Common Name (eg, your name or your server\'s hostname)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 64
```

SIGNING REQUESTS

1. Make a directory for signing requests

```
# mkdir -p /etc/pki/CA/requests
```

2. Create a directory for new certificates:

```
# mkdir -p /etc/pki/CA/newcerts
```

3. Create a certificate signing request for Zabbix/CA server, with common name cazabbix.localhost.local :

```
# openssl req -new -keyout /etc/pki/CA/private/zaca_key.pem -out /etc/pki/CA/requests/zaca_req.pem -newkey rsa:2048
```

4. Create another signing request, but for DB server, common name mydb.localhost.local

```
# openssl req -new -keyout /etc/pki/CA/private/pdb_key.pem -out /etc/pki/CA/requests/pdb_req.pem -newkey rsa:2048
```

SIGNING REQUESTS

1. Create the index.txt and serial files needed by openssl to keep track of the certificates signed:

```
# touch /etc/pki/CA/index.txt
# echo 01 > /etc/pki/CA/serial
```

2. Create a signed certificate for Zabbix/CA(answer yes when prompted):

```
# openssl ca -policy policy_anything -days 365   -out /etc/pki/CA/certs/zaca_crt.pem  -infiles /etc/pki/CA/requests/zaca_req.pem
Certificate is to be certified until Aug 10 08:43:26 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

3. Create a signed certificate for DB (answer yes when prompted):

```
# openssl ca -policy policy_anything -days 365   -out /etc/pki/CA/certs/pdb_crt.pem  -infiles /etc/pki/CA/requests/pdb_req.pem
Certificate is to be certified until Aug 10 08:43:45 2021 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

CONFIGURING MYSQL SERVER

1. Install MySQL 8

```
# sudo dnf install mysql-server
```

2. Start MySQL

```
# systemctl start mysqld
```

3. Run the security script (make it all yes)

```
# mysql_secure_installation
```

4. Create a directory for certificates:

```
# mkdir -p /var/lib/mysql/pki
```

5. Check the firewall, to see if port for MySQL is opened, i.e.:

```
# firewall-cmd --permanent --zone=trusted --add-source=192.0.2.10/32
# firewall-cmd --permanent --zone=trusted --add-port=3306/tcp
# firewall-cmd  --reload
```

CONFIGURING MYSQL SERVER

1. Copy the certificate files from Zabbix/CA server to the directory created:

```
# scp root@192.168.3.92:/etc/pki/CA/private/pdb_nopass_key.pem /var/lib/mysql/pki/server.key
# scp root@192.168.3.92:/etc/pki/CA/certs/pdb_crt.pem /var/lib/mysql/pki/server.crt
# scp root@192.168.3.92:/etc/pki/CA/cecert.pem /var/lib/mysql/pki/ca.crt
```

2. Update permissions:

```
# chown -R mysql. /var/lib/mysql/pki
```

3. Update MySQL configuration file:

```
# vi /etc/my.cnf.d/mysql-server.cnf
```

4. By adding this lines:

```
[mysqld]
ssl-ca=/var/lib/mysql/pki/ca.crt
ssl-cert=/var/lib/mysql/pki/server.crt
ssl-key=/var/lib/mysql/pki/server.key
```

5. Restart MySQL:

```
# systemctl restart mysqld
```

CONFIGURING MYSQL SERVER

1. Login to MySQL:

```
# mysql -u root -p
```

2. Verify settings:

```
mysql> show variables like '%ssl%';
```

```
+--------------------+-----------------------------+
| Variable_name      | Value                       |
+--------------------+-----------------------------+
| admin_ssl_ca       |                             |
| admin_ssl_capath   |                             |
| admin_ssl_cert     |                             |
| admin_ssl_cipher   |                             |
| admin_ssl_crl      |                             |
| admin_ssl_crlpath  |                             |
| admin_ssl_key      |                             |
| have_openssl       | YES                         |
| have_ssl           | YES                         |
| mysqlx_ssl_ca      |                             |
| mysqlx_ssl_capath  |                             |
| mysqlx_ssl_cert    |                             |
| mysqlx_ssl_cipher  |                             |
| mysqlx_ssl_crl     |                             |
| mysqlx_ssl_crlpath |                             |
| mysqlx_ssl_key     |                             |
| ssl_ca             | /var/lib/mysql/pki/ca.crt   |
| ssl_capath         |                             |
| ssl_cert           | /var/lib/mysql/pki/server.crt |
| ssl_cipher         |                             |
| ssl_crl            |                             |
| ssl_crlpath        |                             |
| ssl_fips_mode      | OFF                         |
| ssl_key            | /var/lib/mysql/pki/server.key |
+--------------------+-----------------------------+
```

CONFIGURING MYSQL SERVER

1. Create a user for Zabbix and Zabbix DB:

```
mysql> create user zabbix@IP identified with mysql_native_password BY 'password’;
mysql> create database zabbix character set utf8 collate utf8_bin;
```

2. Restrict Zabbix user to Zabbix DB:

```
mysql> grant all privileges on zabbix.* to zabbix@IP;
```

3. Restrict it to require SSL:

```
mysql> alter user ‘zabb'@'%' require ssl;
```

4. Check if configuration applied:

```
mysql> select user,host,ssl_type from mysql.user;
```

```
+------------------+-----------+----------+
| user             | host      | ssl_type |
+------------------+-----------+----------+
| redhat           | %         | ANY      |
| zabbix           | %         | ANY      |
| mysql.infoschema | localhost |          |
| mysql.session    | localhost |          |
| mysql.sys        | localhost |          |
| root             | localhost |          |
+------------------+-----------+----------+
6 rows in set (0.00 sec)
```

CONFIGURING ZABBIX SERVER

1. Install Zabbix repository on Zabbix/CA server:

```
# rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
# dnf clean all
```

2. Install Zabbix server, frontend and agent:

```
# dnf install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-agent
```

3. On DB server run, to transfer initial schema and data:

```
# scp root@IP:/usr/share/doc/zabbix-server-mysql*/create.sql.gz /home
```

4. On DB server import initial schema and data into the Zabbix DB:

```
# zcat /home/create.sql.gz | mysql -uroot -p zabbix
```

5. Verify that all table are in place:

```
mysql> use zabbix;
mysql> show tables;
166 rows in set (0.00 sec)
```

CONFIGURING ZABBIX SERVER

1. Make Zabbix server SSL directory:

```
# mkdir –p /var/lib/zabbix/ssl/
```

2. Copy certificate files:

```
# cp /etc/pki/CA/cacert.pem /var/lib/zabbix/ssl/
# cp /etc/pki/CA/certs/zaca_crt.pem /var/lib/zabbix/ssl/server.zabbix.crt
# cp /etc/pki/CA/private/zaca_nopass_key.pem /var/lib/zabbix/ssl/server.zabbix.key
```

3. Update permissions:

```
# chown -R zabbix /var/lib/zabbix/ssl/
# chmod 400 /var/lib/zabbix/ssl/cacert.pem
# chmod 400 /var/lib/zabbix/ssl/server.zabbix.crt
# chmod 400 /var/lib/zabbix/ssl/server.zabbix.key
```

CONFIGURING ZABBIX SERVER

1. Update Zabbix server configuration file:

```
# vi /etc/zabbix/zabbix_server.conf
```

2. Update the DB access parameters:

```
DBHost=mydb.localhost.local
DBName=zabbix
DBUser=zabbix
DBPassword=zabbix
```

3. Update SSL related parameters:

```
DBTLSConnect=verify_full
DBTLSCAFile=/var/lib/zabbix/ssl/cacert.pem
DBTLSCertFile=/var/lib/zabbix/ssl/server.zabbix.crt
DBTLSKeyFile=/var/lib/zabbix/ssl/server.zabbix.key
```

CONFIGURING ZABBIX FRONTEND

1. Make Apache SSL directory on Zabbix/CA server:

```
# mkdir -p /etc/httpd/ssl/
```

2. Copy certificate files:

```
# cp /etc/pki/CA/cacert.pem /etc/httpd/ssl/
# cp /etc/pki/CA/certs/zfca_crt.pem /etc/httpd/ssl/server.crt
# cp /etc/pki/CA/private/zfca_nopass_key.pem /etc/httpd/ssl/server.key
```

3. Update permissions:

```
# chown -R apache /etc/httpd/ssl/
# chmod 400 /etc/httpd/ssl/cacert.pem
# chmod 400 /etc/httpd/ssl/server.crt
# chmod 400 /etc/httpd/ssl/server.key
```

4. Configure time zone:

```
vi /etc/php-fpm.d/zabbix.conf
php_value[date.timezone] = Europe/Riga
```

5. Start Zabbix processes:

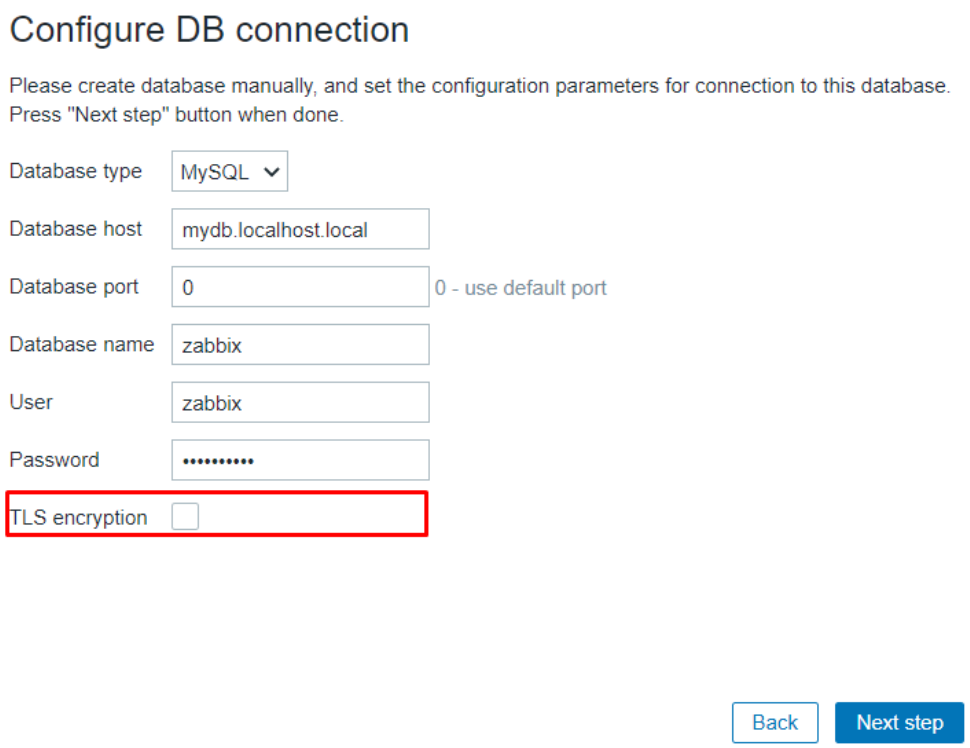
```
# systemctl restart zabbix-server zabbix-agent httpd php-fpm
# systemctl enable zabbix-server zabbix-agent httpd php-fpm
```

CONFIGURING ZABBIX FRONTEND

1. Go to your front-end address

```
# http://IP/zabbix
```

2. Add the DB name, username, password and enable TLS encryption option:

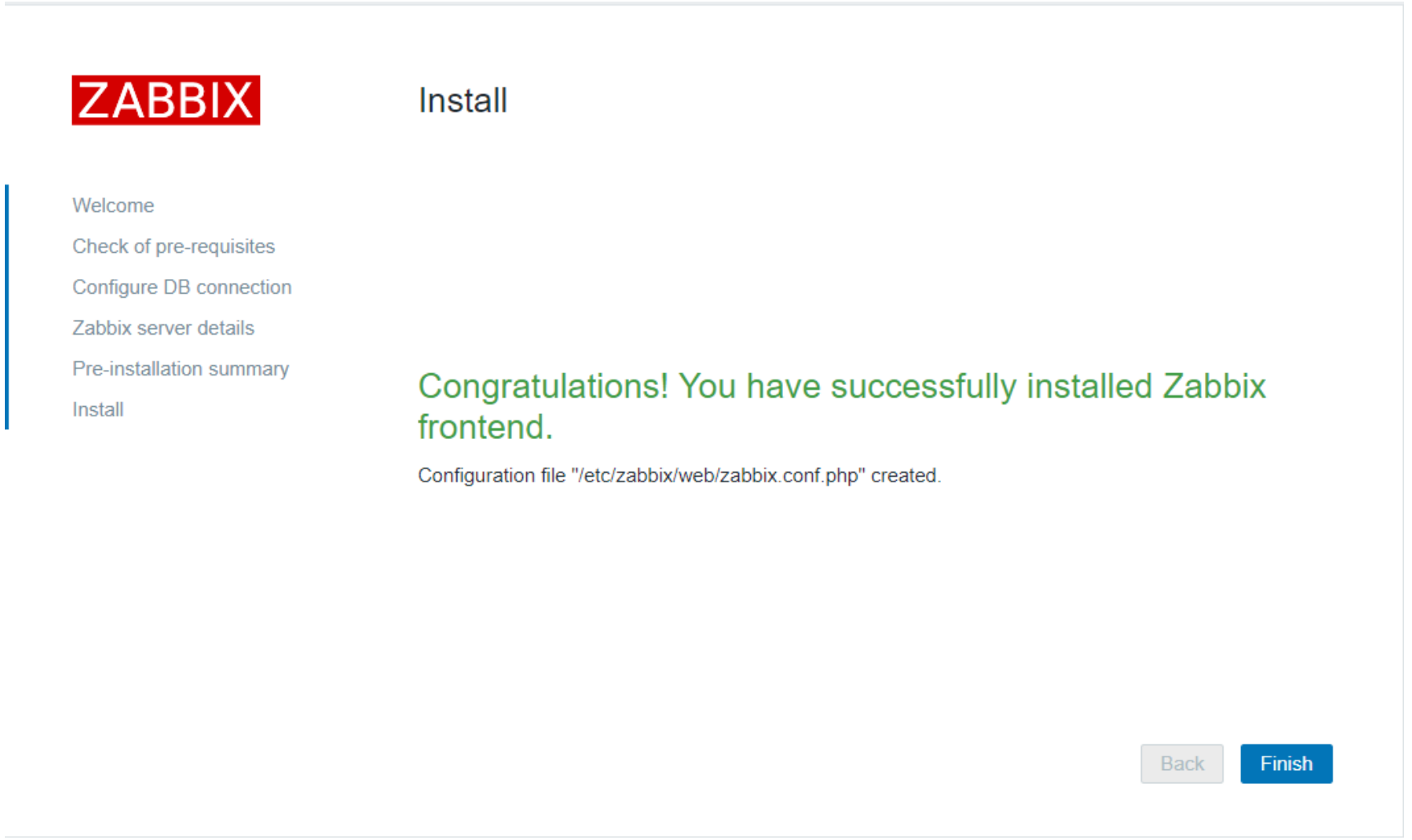


CONFIGURING ZABBIX FRONTEND

1. Add the path to certificate files:

```
/etc/httpd/ssl/cacert.pem
/etc/httpd/ssl/server.crt
/etc/httpd/ssl/server.key
```

FINISH IT UP


ZABBIX
Install
Welcome
Check of pre-requisites
Configure DB connection
Zabbix server details
Pre-installation summary
Install
Congratulations! You have successfully installed Zabbix frontend.
Configuration file "/etc/zabbix/web/zabbix.conf.php" created.
Back
Finish

CONGRATULATIONS!

Thank you!