Viglet Turing ES
Installation Guide

Viglet Team

Version 0.3.10, 25-12-2024

Table of Content

Preface

Viglet Turing ES (https://viglet.com/turing) is an open source solution (https://github.com/openviglet), which has Semantic Navigation and Chatbot as its main features. You can choose from several NLPs to enrich the data. All content is indexed in Solr as search engine.

Chapter 1. Installing Java

Turing ES requires Java.

> **NOTE** | Turing ES only supports Java 21.

During installation, Turing ES reads the JAVA_HOME and PATH variables to install the necessary services. Therefore, you must modify the variables as specified here:

- On Windows:

    - Set the JAVA_HOME system variable to point to the JAVA location.

    - Attach the PATH system variable with %JAVA_HOME%/bin.

- On Unix:

    - Create an environment variable called JAVA_HOME.

    - Provide the JAVA bin path in the PATH variable.

Chapter 2. Turing ES Requirements

The information below describes the resource modes available to run Turing ES.

> **NOTE** | Use a screen resolution of at least 1024 x 768 for optimum clarity.

This section describes the minimum and suggested features for systems on which you can configure Turing ES.

2.1. Hardware Requirements

Minimum	Recommended
Single 1.5 GHz processor 1 GB available memory	Dual 2 GHz processors with Hyper-Threading enabled 2 GB available memory

In addition, the following is required:

- 1GB of disk space for software.

- A compatible version of a database.

Chapter 3. Installing and Configuring Databases

Turing ES supports multiple database platforms, each with specific unique requirements to maximize effectiveness with Turing ES. The following sections describe these requirements.

3.1. Turing ES Database Table Requirements

Turing ES uses multiple tables in multiple databases (also called database schemas by some vendors such as Oracle). You must create new databases / schemas that will contain the sets of tables described below.

3.1.1. Oracle Database Requirements

This section describes configuration requirements for an existing database.

To set up an Oracle database:

1. Install and configure a single database instance for use by Turing ES.

2. Create a database schema to contain the Turing ES system tables.

3. Configure the Turing ES admin user for the database. For Oracle, the database user with the grant of the CONNECT and RESOURCE roles. Make sure the user has UNLIMITED TABLESPACE and CREATE VIEW privileges.

3.1.2. Microsoft SQL Server Database Requirements

This section discusses the steps required to configure Microsoft SQL Server for your Turing ES databases.

Creating the Turing ES Database before you begin, be sure to refer to the following guides:

The following procedure describes how to create and configure a Microsoft SQL Server database for use by Turing ES.

When creating the Turing ES database, use the following options:

- You must install the Database Engine feature and Management Tools are recommended. All other features are optional.

- Set the Collation configuration variable to SQL_Latin1_General_CP1_CS_AS or Latin1_General / Case-sensitive / Accent-Sensitive.

- Set authentication mode to mixed mode.

- Turing ES system user cannot be sa.

Configuring the SQL Server Database and Schemas

This section discusses the steps required to create and configure a SQL Server database for use by Turing ES.

| NOTE | The database and usernames provided are examples only. |

To create the SQL Server database:

1. Select Databases > New Database.

2. Enter the database name (eg turing) and the size of the data and log files according to the environment.

3. Go to the Options page and set the following values:

    - Collation = Latin1_General_CS_AS or SQL_Latin1_General_CP1_CS_AS

    - Miscellaneous > Allow Snapshot Isolation = True

    - Miscellaneous > Is Read Committed Snapshot On = True

```
USE master
GO
ALTER DATABASE turing
SET SINGLE_USER
WITH ROLLBACK IMMEDIATE
GO
ALTER DATABASE turing
SET ALLOW_SNAPSHOT_ISOLATION ON
GO
ALTER DATABASE turing
SET READ_COMMITTED_SNAPSHOT ON
GO
ALTER DATABASE turing
```

```
SET MULTI_USER
GO
```

1. Under Security > Logins, select New Login and set the following attributes:

    - Enter the Login name (for example, turing)

    - Deselect the Enforce Password Expiration option.

    - Set the Default database.

2. On the User Mapping page, set the following attributes for the turing database:

    - Default user = turing

    - Default schema = turing

    - With the turing database is selected, select the option for db_owner

3. Click OK.

3.1.3. PostgreSQL Database Requirements

This section discusses the steps required to configure PostgreSQL for your Turing ES databases:

- Installing and configuring the PostgreSQL Database

- Configuring the Turing ES user, database, and schema

- Installing and configuring the PostgreSQL Database

Installing and configuring the PostgreSQL Database

This section describes the steps required to install and configure a PostgreSQL database for your Turing ES environment. You will also need to configure the Turing user, database and schemas.

To install the PostgreSQL database:

1. Navigate to the following URL and follow the instructions provided for your supported version of PostgreSQL: https://www.postgresql.org/docs/manuals/

To configure the PostgreSQL instance:

- Open the postgresql.conf file (located in <PostgreSQL_installDir>\<version>\data\) and set the following parameters:

```
max_connections = 10
max_prepared_transactions = 10
shared_buffers = 512MB
autovacuum = off
```

IMPORTANT	The autovacuum daemon is used to reclaim memory/storage space the database is no longer using. You must re-enable this utility after you have finished installing and configuring Turing ES.
NOTE	Configuration values will vary for higher transaction environments and numbers of users. max_prepared_transactions should be at least as large as max_connections so that every session can have a prepared transaction pending.

Configuring the Turing user, database, and schemas

This section discusses the steps required to configure the users, database and schemas for your Turing ES environment.

NOTE	You must be a PostgreSQL super-user or have createuser privileges to create Turing users. To create and configure the Turing ES database, you must be a super-user or have create privileges.
IMPORTANT	When you create a database table in PostgreSQL, the field name must be lowercase.

To configure the Turing ES user:

1. Create a user as the owner of the Turing ES database:

```
CREATE USER turing PASSWORD <password> with createuser;
```

1. Create a Turing ES database:

```
CREATE DATABASE turing_db;
ALTER DATABASE turing_db OWNER TO turing;
```

To create and access the Turing ES user and schema:

- Log in to the turing (database) using the turing account. For example, on the operating system command prompt:

```
$psql -U turing -d turing_db
```

- Create the Turing schema and grant ownership to the Turing user:

```
CREATE SCHEMA turing;
ALTER SCHEMA turing OWNER TO turing;
```

- Set the PostgreSQL search_path variable to the user to allow direct access to the respective schemas:

```
ALTER USER turing SET SEARCH_PATH = turing;
```

Chapter 4. Turing Utils

First you need install Turing Utils, that contains sample configurations and script to facilitate the use of Turing ES, so go to https://viglet.org/turing/download/ and click on "Integration > Utils" link to download it.

Extract the turing-utils.zip to /appl/viglet/turing/utils

```
mkdir -p /appl/viglet/turing/utils
unzip turing-utils.zip -d /appl/viglet/turing/utils
```

Chapter 5. Keycloak Configuration

5.1. Database

Create the Database and User in MariaDB or MySQL

```
CREATE USER 'keycloak'@'%' IDENTIFIED BY 'keycloak';
CREATE DATABASE IF NOT EXISTS keycloak;
GRANT ALL PRIVILEGES ON keycloak.* TO 'keycloak'@'%' WITH GRANT OPTION;
```

5.2. Create the keystore file

```
keytool -genkeypair -alias localhost -keyalg RSA -keysize 2048 -validity 365
-keystore server.keystore -dname "cn=Server Administrator,o=Acme,c=GB" -keypass
secret -storepass secret
```

5.3. Keycloak configuration

Edit the /appl/keycloak/conf/keycloak.conf file with the configuration:

```
# Database

# The database vendor.
db=mariadb

# The username of the database user.
db-username=keycloak

# The password of the database user.
db-password=keycloak

# The full database JDBC URL. If not provided, a default URL is set based on the
selected database vendor.
db-url=jdbc:mariadb://localhost:3306/keycloak
```

```
./kc.sh build --http-relative-path=/kc
```

5.4. Keycloak Linux Service

Create the /appl/systemd/systemd-env script file.

```
JAVA_HOME=/appl/java/jdk21
PATH=/appl/java/jdk21/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin
```

Create the /etc/systemd/system/keycloak.service file.

```
[Unit]
Description=Keycloak Application Server
After=syslog.target network.target

[Service]
EnvironmentFile=/appl/systemd/systemd-env
Type=simple
TimeoutStopSec=0
KillSignal=SIGTERM
KillMode=process
SuccessExitStatus=143
LimitMEMLOCK=infinity
SendSIGKILL=no
WorkingDirectory=/appl/keycloak/
User=vilt
Group=vilt
LimitNOFILE=102642
ExecStart=/appl/keycloak/bin/kc.sh start --optimized --https-key-store
-password=secret --hostname=https://localhost/kc --hostname
-admin=https://localhost/kc

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

5.5. Configuring the Apache HTTP Server

Add the follow lines in Apache HTTP Server configuration:

```
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
```

```
SSLProxyCheckPeerExpire off

ProxyPass "/kc/" "https://localhost:8443/kc/"
ProxyPassReverse "/kc/" "https://localhost:8443/kc/"
```

Chapter 6. Solr Configuration

You need to configure Solr to be able to store Semantic Navigation and ChatBot data.

6.1. OOTB

With Solr running, go to Turing Utils and create the new cores that will be used by Turing ES.

1. Install the cores into solr:

```
cd <SOLR_DIR>/bin
## For Semantic Navigation
./solr create_collection -c turing -n turing -d
/appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>
## For ChatBot
./solr create_collection -c converse -n converse -d
/appl/viglet/turing/utils/solr/<VERSION>/converse/<LANGUAGE>
```

2. Started Solr Service

6.2. OpenText InfoFusion

1. Create the cores into Solr:

```
cd <SOLR_DIR>/bin
## For Semantic Navigation
mkdir <SOLR_DIR>/server/solr/turing
cp -Rf /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/.
<SOLR_DIR>/server/solr/turing/
## For ChatBot
mkdir <SOLR_DIR>/server/solr/converse
cp -Rf /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/.
<SOLR_DIR>/server/solr/converse/
```

2. If the core name of Semantic Navigation Site will be different, you need modify the `core.properties` file and change the `name` and `collection` attributes, for example:

```
cd <SOLR_DIR>/bin
mkdir <SOLR_DIR>/server/solr/sample
cp /appl/viglet/turing/utils/solr/<VERSION>/turing/<LANGUAGE>/.
<SOLR_DIR>/server/solr/sample/
chown otif:otif <SOLR_DIR>/server/solr/sample

## Modify the name and collection fields
sed -i 's/turing/sample/g' <SOLR_DIR>/server/solr/sample/core.properties

cat <SOLR_DIR>/server/solr/sample/core.properties

numShards=1
name=sample
shard=shard1
collection=sample
coreNodeName=core_node1
```

3. Edit the `<IF_API>/bin/otif.sh` and add the new cores to sync with Zookeeper.

```
# Push Solr configurations in ZooKeeper
pushSolrConfigs() {
    ...
    cd "$BASE_DIR"
    # For Semantic Navigation
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/turing/conf" -confname turing
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/<OTHER_CORE>/conf" -confname <OTHER_CORE>
    # For ChatBot
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir
"<IF_SOLR>/server/solr/converse/conf" -confname converse
    # Default InfoFusion cores (Do not modify)
    ./zk.sh -cmd upconfig -zkhost "$SOLR_ZK_ENSEMBLE" -confdir "
$BASE_DIR/../etc/solr/otif/en/conf" -confname otif_en

    ...
    cd "$CURRENT_DIR"
}
```

4. With Zookeeper running, execute the command below:

```
cd <IF_API>/bin
./otif.sh push-solr-configs
```

5. Restart the Solr.

Chapter 7. Installing Turing ES

7.1. Turing ES Download

Go to https://viglet.org/turing/download/ and click on "Download Turing ES" button to download the viglet-turing.jar executable.

Copy the viglet-turing.jar to /appl/viglet/turing/server

```
mkdir -p /appl/viglet/turing/server
cp viglet-turing.jar /appl/viglet/turing/server
chmod 770 /appl/viglet/turing/server/viglet-turing.jar
```

7.2. Database

By default, Turing uses H2 as its embedded database, but if you prefer another database you can follow the procedure below, creating /appl/viglet/turing/server/viglet-turing.conf file with following lines:

7.2.1. MySQL

Below is an example of how to configure Turing for MySQL.

viglet-turing.conf for MySQL

```
JAVA_OPTS="-Xmx1g -Xms1g
-Dspring.datasource.url=jdbc:mariadb://localhost:3306/turing
-Dspring.datasource.username=turing -Dspring.datasource.password=turing
-Dspring.datasource.driver-class-name=org.mariadb.jdbc.Driver
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
"
```

7.2.2. Oracle Database

Below is an example of how to configure Turing for Oracle Database.

viglet-turing.conf for Oracle Database

```
JAVA_OPTS="-Xmx1g -Xms1g
-Dspring.datasource.url=jdbc:oracle:thin:@localhost:1521/turing
-Dspring.datasource.username=turing -Dspring.datasource.password=turing
-Dspring.datasource.driver-class-name=oracle.jdbc.OracleDriver
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect"
```

7.2.3. PostgreSQL

Below is an example of how to configure Turing for PostgreSQL.

viglet-turing.conf for PostgreSQL

```
JAVA_OPTS="-Xmx1g -Xms1g
-Dspring.datasource.url=jdbc:postgresql://localhost:5432/turing
-Dspring.datasource.username=turing -Dspring.datasource.password=turing
-Dspring.datasource.driver-class-name=org.postgresql.Driver
-Dspring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQL94Dialect
"
```

7.3. Encrypting the Database Password

For more security, it is possible to encrypt the Database password in viglet-turing.conf following the procedure below:

```
cd /appl/viglet/turing/utils/scripts
./turGenerateDBPassword.sh /appl/viglet/turing/server
Password:
Retype Password:

Thanks, the password was successfully generated in /appl/viglet/turing/server/db-
encrypted.properties

cat /appl/viglet/turing/server/db-encrypted.properties
clqp6QxceJQIp6lnaiKelQc9DvdNKNxg
```

Copy the content of /appl/viglet/turing/server/db-encrypted.properties and replace the `-Dspring.datasource.password=<PLAIN_TEXT_PASSWORD>` value of `/appl/viglet/turing/server/viglet-turing.conf` file with `-Dspring.datasource.password=ENC(<ENCRYPTED_PASSWORD>)`.

For example: `-Dspring.datasource.password=ENC(clqp6QxceJQIp6lnaiKelQc9DvdNKNxg)`

7.4. Creating Turing ES Service on Linux

7.4.1. Red Hat and CentOS

As root, create a `/etc/systemd/system/turing.service` file with the following lines:

```
[Unit]
Description=Viglet Turing ES
After=syslog.target

[Service]
User=viglet
ExecStart=/appl/viglet/turing/server/viglet-turing.jar
SuccessExitStatus=143

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

This way, you can interact with the Turing ES service:

```
./systemctl start turing.service
./systemctl stop turing.service
./systemctl restart turing.service
./systemctl status turing.service
```

7.5. Starting Turing ES

When Turing ES is started for the first time, it will do the initial setup and start downloading the OpenNLP models.

```
$ ./viglet-turing.jar

  _____            _             _    ___
 |_   _|_   _ _ _ (_) _ _   __ _   /_\  |_ _|
   | | | || || '_|| || ' \ / _` | / _ \  | |
   |_|  \_,_||_|  |_||_||_|\__, | /_/ \_\|___|  (v0.3.6-d9fc453)
                          |___/

:: Copyright (C) 2016-2022, Viglet Team <opensource@viglet.com>

:: Built with Spring Boot :: 2.7.6

Starting TuringAI v0.3.6-d9fc453 using Java 17.0.1 on LEGION with PID 20232
(/appl/viglet/turing/server/viglet-turing.jar started by alexa in
/appl/viglet/turing/servers)
The following 1 profile is active: "production"
Started TuringAI in 16.432 seconds (JVM running for 17.358)
First Time Configuration ...
[ OK ] en-ner-date.bin model
[ OK ] en-ner-person.bin model
[ OK ] en-ner-location.bin model
[ OK ] en-ner-money.bin model
[ OK ] en-ner-organization.bin model
[ OK ] en-ner-percentage.bin model
[ OK ] en-ner-time.bin model
[ OK ] en-parser-chunking.bin model
[ OK ] en-pos-maxent.bin model
[ OK ] en-pos-perceptron.bin model
[ OK ] en-sent.bin model
[ OK ] en-token.bin model
[ OK ] pt-pos-maxent.bin model
[ OK ] pt-pos-perceptron.bin model
[ OK ] pt-sent.bin model
[ OK ] pt-token.bin model

Configuration finished.
```

7.6. Accessing the Turing Console

Turing provides remote access to administration, configuration, and management through its Web application interfaces. Once setup is complete, the Console become browser-accessible through the following URL: http://<host>:<port>/console where <host>:<port> is the listening host and port for the Turing ES that you created in the application server. The default port is 2700 for the Console.

By default the login/password are: admin/admin

Appendix A: Installation Modes

A.1. Turing ES Server

A.1.1. Simple.

Turing ES will be installed only using OpenNLP and H2 database embedded in Turing ES itself.

Prerequisites

1. Linux server

2. Java 14

3. 50Gb HDD

4. 2 Gb of RAM

Target Audience

Development and testing environment. Because it requires fewer components and lower memory usage.

Estimated Hours

2 hours

IMPORTANT	Servers will be provided by the customer.

A.1.2. Docker Compose

Turing ES and its dependencies will be installed using Docker Compose script, including the following services:

- MariaDB – to store Turing ES system tables

- Solr – Used by Turing ES's Semantic Navigation and Chatbot

- Nginx – WebServer for Turing ES to use port 80

- Turing ES.

Prerequisites

1. Linux server

2. Docker and Docker Compose installed

3. 50Gb HDD

4. 4Gb of RAM

Target Audience

Customers who need more complex environments, but avoid the installation and configuration of each product. It can be used in an QA or Production environment.

Estimated Hours

16 hours

> **IMPORTANT** | Servers and docker configuration will be provided by the customer.

A.1.3. Kubernetes

Turing ES and its dependencies will be installed using Kubernetes scripts, including the following services:

- MariaDB – to store Turing ES system tables

- Solr – Used by Turing ES's Semantic Navigation and Chatbot

- Nginx – WebServer for Turing ES to use port 80

- Turing ES.

Prerequisites

1. Linux Server with Kubernetes installed or Cloud that supports Kubernetes

2. 100Gb of Storage

3. 4Gb RAM

Target Audience

Customers who want to use cloud solutions like Google, AWS, Oracle, etc. It can be used in the production environment in a scalable way.

Estimated Hours

20 hours

| IMPORTANT | Cloud infrastructure and servers will be provided by the customer. |

A.1.4. Manual Installation of Services

The services will be installed individually on the servers following the Installation Guide procedure, which will include the following services:

- MariaDB – to store Turing ES system tables
- Solr – Used by Turing ES's Semantic Navigation and Chatbot
- Apache – WebServer for Turing ES to use port 80
- Turing ES.

Prerequisites

1. One Linux server or up to 4 Linux servers to install services
2. 50 - 100Gb of Storage for each server
3. Minimum 2Gb RAM for each Server
4. The services will be installed individually on the servers following the Installation Guide procedure.

Target Audience

Customers who prefer the on-premise structure and want to have the services installed directly on the servers. It can be used in Development, QA and Production.

Estimated Hours

20 hours

IMPORTANT | Servers will be provided by the customer.

A.2. Connectors

Turing ES has several connectors to allow you to index the contents in Semantic Navigation:

- Apache Nutch (Crawler)
- Wordpress
- OpenText WEM Listener
- FileSystem
- Database

A.2.1. Prerequisites

1. New linux server or existing server with content or files that will be indexed.
2. 50 of Storage for each server.

A.2.2. Estimated Hours

On average, it will take **16 hours** to configure the connector and have the first indexing version in Turing ES.

A.3. NLP

The customer can choose the NLP that will be used by Turing ES:

- Apache OpenNLP (Embedded)
- SpaCy NLP
- Stanford CoreNLP
- OpenText Content Analytics
- Poliglot

A.3.1. Prerequisites

1. Linux server

2. 50 of Storage for each server

3. Minimum 2 Gb of RAM

A.3.2. Estimated Hours

On average, it will take **4 hours** to configure NLP and configure Turing ES to use it.

Appendix B: Docker Compose

You can start the Turing ES using MariaDB, Solr and Nginx.

```
./gradlew turing-app:build -x test -i --stacktrace
docker-compose up
```

> **NOTE** If you have problems with permissions on directories, run `chmod -R 777 volumes`

B.1. Docker Commands

B.1.1. Turing

```
docker exec -it turing /bin/bash
```

B.1.2. Solr

```
docker exec -it turing-solr /bin/bash
```

Check logs

```
docker-compose exec turing-solr cat /opt/solr/server/logs/solr.log
# or
docker-compose exec turing-solr tail -f /opt/solr/server/logs/solr.log
```

B.1.3. MariaDB

```
docker exec -it turing-mariadb /bin/bash
```

B.1.4. Nginx

```
docker exec -it turing-nginx /bin/bash
```