Guide - Deploying for Production

apiman 1.2.5.Final

Chapter 1. Deployment Tooling for apiman

This guide should serve as a guide to manually deploy apiman into production. However, we also offer a simple shell script that can do much of the work for you. It will likely always be a work in progress, so many production deployers may not feel comfortable using it. However, it can currently install the following components:

- Elasticsearch (1.x)
- Keycloak (1.7.0)
- API Manager
- API Gateway

You can find the apiman production deployer here:

https://github.com/apiman/apiman-deployer/tree/1.2.5.Final

Simply download that script and run it on each of the machines you wish to install the various apiman components!

Tip

The production guide assumes you are installing into WildFly 10. The instructions are slightly different if you are using some other platform (Tomcat, older WildFly versions, EAP 7).

Chapter 2. Architecture Summary

Before we get started, it may be useful to know the overall architecture of apiman. Let’s start with a picture!

Architecture of apiman.

The apiman solution is made up of a number of pieces, including:

- Keycloak Authentication Server
- Relational Database
- Elasticsearch Datastore
- apiman API Gateway
- apiman API Manager

The image above should give you an idea of how they all fit together.

Chapter 3. Database

You have a number of persistence options in both Keycloak and apiman, but this guide will focus on using an RDBMS instead of other options (e.g. Elasticsearch for apiman or mongodb for KC).

On a separate server, install a production ready database such as mysql, oracle, or postresql. You will want to create two schemas/databases:

- keycloak
- apiman

Make sure your DB is accessible remotely and enable whatever security options you need (SSL, users/passwords). We recommend using two different users and permissioning them appropriately (e.g. create a "sa_apiman" account and a "sa_keycloak" account).

After creating the ‘apiman' database, you can initialize it using the appropriate DDL for the database you selected. We currently support mysql 5, oracle 12, and postgresql 9. The DDLs for these databases can be found in the quickstart overlay ZIP file, or you can grab them directly from github. For example:

https://github.com/apiman/apiman/tree/master/distro/data/src/main/resources/ddls

Make sure you grab the right one for whichever version of apiman you are going to be installing (**hint**: use the appropriate github tag).

Chapter 4. Elasticsearch

If you wish to enable metrics in apiman (who doesn't?) you'll need to install Elasticsearch. This is because the API Gateway stores all metrics information in Elasticsearch (by default), and the API Manager queries that data to present analytics information in the UI.

Please see the Elasticsearch documentation for how to install and configure it in production. Ultimately you will need Elasticsearch running in a well known and accessible location. We also recommend you enable authentication (e.g. via Shield) and SSL.

Tip

The data in Elasticsearch is not backed up or stored in some other location - Elasticsearch is being used as the primary/canonical data store for the metrics information. You may wish to configure backup procedures

Chapter 5. Keycloak

In production, it is typically useful to deploy Keycloak server as a standalone solution. For more specific information about how to configure a standalone Keycloak server, see the Keycloak documentation:

http://keycloak.github.io/docs/userguide/html/server-installation.html

Once Keycloak is installed as a standalone server, you must configure the ‘apiman' realm. This realm will be used for authentication into each of the apiman components (API Manager REST services, API Manager UI, API Gateway REST services, etc).

Tip

You can configure additional Keycloak Realms for use when using the apiman "Keycloak OAuth2 Policy".

Fortunately, apiman comes with a realm file you can import. Simply log into your Keycloak server and then create the apiman realm using this file:

https://github.com/apiman/apiman/blob/master/distro/data/src/main/resources/data/apiman-realm.json

Warning

The realm file contains some credentials/secrets that you will want to modify or regenerate. The defaults are obviously **not** secure.

Once you have created the apiman realm, you must use the Keycloak UI to add your API Manager UI as a valid redirect URL for the ‘apimanui' client. To do this, log into the Keycloak admin console and choose the **apiman** realm. Next click "clients" in the left navigation, and choose **apimanui** in the resulting list. From there you can add public URL if your API Manager UI to the list of "**Valid Redirect URIs**". It might be something like:

```
https://apimanager.mycompany.org:8443/apimanui/*
```

That will allow users of the API Manager UI to actually log in and be properly redirected back to the application!

Chapter 6. API Gateway

6.1. Installing the API Gateway

The easiest way to install just the API Gateway is to download and install the apiman quickstart overlay ZIP and then remove the extraneous components. Follow that up with some modification of the apiman.properties configuration file and you'll be Gatewaying in no time!

Here are the steps you should take to install a standalone API Gateway:

1. Download and unpack WildFly 10
2. Download apiman
3. Unpack apiman into WildFly 10
4. Remove unused apiman deployments from standalone/deployments

Which apiman deployments should you delete? These:

```
standalone/deployments/apiman-ds.xml
standalone/deployments/apiman-es.war
standalone/deployments/apiman.war
standalone/deployments/apimanui.war
```

6.2. Configuring the API Gateway

6.2.1. Disabling the Keycloak Server

Because you will be using an external/standalone Keycloak server, it is useful to disable the Keycloak components that are bundled with the apiman quickstart overlay ZIP. To do that, **remove** the following subsystem from the **standalone-apiman.xml** file:

```
<subsystem xmlns="urn:jboss:domain:keycloak-server:1.1">
  <web-context>auth</web-context>
</subsystem>
```

6.2.2. Setting the API Gateway Public Endpoint

An important step is to let the API Gateway know what its public endpoint is. This is important because the API Manager will sometimes ask the Gateway to report on the Managed Endpoint for a published API.

To set the public URL/endpoint of the API Gateway, add the following to apiman.properties:

```
apiman-gateway.public-endpoint=https://api-gateway-host.org:8443/apiman-
gateway/
```

Warning

Please make sure to use your appropriate values for the host and port.

6.2.3. Configuring Keycloak Authentication for the Gateway API

The API Gateway has a REST based configuration API which the API Manager uses when publishing APIs to it. This API is protected by Keycloak authentication. The configuration included in the apiman quickstart overlay ZIP assumes that the Keycloak server is local, so you'll need to modify the **standalone-apiman.xml** file to point to the remote Keycloak instance.

Here is the relevant portion of the **standalone-apiman.xml** file that you must change:

```
<realm name="apiman">
  <realm-public-key>MIIB..snip..QAB</realm-public-key>
  <auth-server-url>https://keycloak-host.org:8443/auth</auth-server-url>
  <ssl-required>none</ssl-required>
  <enable-cors>false</enable-cors>
  <principal-attribute>preferred_username</principal-attribute>
</realm>
```

6.2.4. Pointing the API Gateway to a Remote Elasticsearch

The API Gateway uses Elasticsearch in a number of ways, including:

- Storing configuration information
- Managing shared state across a cluster
- Storing metrics to share with the API Manager (analytics)

In order to configure the gateway properly, you will need to configure the location of the Elasticsearch instance. To do this, modify these properties in the **apiman.properties** file:

```
apiman.es.protocol=http
apiman.es.host=es.myorg.com
apiman.es.port=9200
apiman.es.username=es_admin
apiman.es.password=es_admin_password
```

Obviously you will need to replace the values in the properties above with those appropriate for your installation of Elasticsearch.

Chapter 7. API Manager

7.1. Installing the API Manager

The easiest way to install just the API Manager is to download and install the apiman quickstart overlay ZIP and then remove the extraneous components. Follow that up with a few configuration modifications, and you should have the Manager running in no time!

Here are the steps you should take to install a standalone API Manager:

1. Download and unpack WildFly 10
2. Download apiman
3. Unpack apiman into WildFly 10
4. Remove unused apiman deployments from standalone/deployments

Which apiman deployments should you delete? These:

```
standalone/deployments/apiman-es.war
standalone/deployments/apiman-gateway-api.war
standalone/deployments/apiman-gateway.war
```

7.2. Configuring the API Manager

7.2.1. Disabling the Keycloak Server

Because you will be using an external/standalone Keycloak server, it is useful to disable the Keycloak components that are bundled with the apiman quickstart overlay ZIP. To do that, make the following modification to the **standalone-apiman.xml** file:

```
<subsystem xmlns="urn:jboss:domain:keycloak-server:1.1">
  <web-context>auth</web-context>
</subsystem>
```

7.2.2. Connecting to the Database

This guide assumes you are using a production ready RDBMS as the storage layer for the API Manager. Note that other options exist, but configuring them is out of scope for this guide.

Hopefully you've already created and initialized the database in the earlier section labeled "*Installing a Database*". So at this point you really only need to connect the API Manager up to the already existing database. The following must be done in order to connect to your database:

- Deploy a JDBC driver compatible with your database

- Update the **apiman-ds.xml** datasource file (to point it at your database)
- Update the hibernate dialect in **apiman.properties**

First, you will need to deploy a JDBC driver that is compatible with whichever database you have chosen. Here are two popular drivers:

MySQL 5

https://repo1.maven.org/maven2/mysql/mysql-connector-java/5.1.33/mysql-connector-java-5.1.33.jar

PostgreSQL 9

https://repo1.maven.org/maven2/org/postgresql/postgresql/9.3-1102-jdbc41/postgresql-9.3-1102-jdbc41.jar

The easiest way to deploy the driver is to simply download it and copy it into the **wildfly/standalone/deployments** directory.

Next, you must update or replace the **apiman-ds.xml** file to something that is configured for your particular database. Examples of appropriate datasource files for mysql and postgresql can be found here:

https://github.com/apiman/apiman/tree/master/distro/data/src/main/resources/sample-configs

These examples are also included in the apiman quickstart overlay ZIP download.

Finally you must update the **apiman.properties** file to configure the hibernate dialect for your database. Apiman includes specific dialects that should be used when installing your database via the included DDL files:

- **H2**: io.apiman.manager.api.jpa.ApimanH2Dialect
- **MySQL**: io.apiman.manager.api.jpa.ApimanMySQL5Dialect
- **Oracle**: io.apiman.manager.api.jpa.ApimanOracle12Dialect
- **Postgresql**: io.apiman.manager.api.jpa.ApimanPostgreSQLDialect

For example, here is the line you should change in the **apiman.properties** file:

```
apiman.hibernate.dialect=io.apiman.manager.api.jpa.ApimanH2Dialect
```

Change the value of that property to the appropriate dialect for your database.

7.2.3. Point the API Manager to the API Gateway

Now that both your API Manager and API Gateway are running, you need to hook them up. This just means telling API Manager where the gateway lives. There is an admin UI page in apiman that will let you do this. Simply navigate here:

https://api-manager-host.org:8443/apimanui/api-manager/admin/gateways

From there you will be able to click on the gateway and modify its settings. Make sure to use the **Test** button on the Edit Gateway UI page to make sure you got the settings right! Don't worry, the **Test** button will simply try to make a connection to the API Gateway's configuration URL, asking it for the current Gateway status. If the Gateway responds as expected, then you can be confident that your settings are correct.

Tip

You will need to log into the UI. The default credentials are: admin/admin123!

Tip

You may have changed the default user credentials when you installed and configured Keycloak. If so, make sure you use those credentials.

7.2.4. Configuring Keycloak Authentication for the Manager API and UI

The API Manager has a REST based API which the User Interface uses for all actions taken. It can also be used directly for automation and/or integration purposes. This API is protected by Keycloak authentication. The configuration included in the apiman quickstart overlay ZIP assumes that the Keycloak server is local, so you will need to modify the **standalone-apiman.xml** file to point to the remote Keycloak instance.

Here is the relevant portion of the **standalone-apiman.xml** file that you must change:

```
<realm name="apiman">
  <realm-public-key>MIIB..snip..QAB</realm-public-key>
  <auth-server-url>https://keycloak-host.org:8443/auth</auth-server-url>
  <ssl-required>none</ssl-required>
  <enable-cors>false</enable-cors>
  <principal-attribute>preferred_username</principal-attribute>
</realm>
```

7.2.5. Pointing the API Manager to a Remote Elasticsearch

The API Manager uses Elasticsearch for analysis of metrics. This metrics data is stored in Elasticsearch by the API Gateway whenever API requests are handled. Therefore, the API Manager and API Gateway must talk to the same Elasticsearch instance/cluster.

To configure Elasticsearch for the API Manager, modify these properties in the **apiman.properties** file:

```
apiman.es.protocol=http
apiman.es.host=es.myorg.com
apiman.es.port=9200
apiman.es.username=es_admin
apiman.es.password=es_admin_password
```

Obviously you will need to replace the values in the properties above with those appropriate for your installation of Elasticsearch.