# Practical Exercise: Getting Started with WSO2 API Manager

## Training Objective

Verify that the products required for running this tutorial are installed and configured. and deploy and test the data required to work with the sample.

**Note**: The participants are expected to be connected to the internet throughout in order to successfully complete the lab exercises.

## Business Scenario

PizzaShack Limited wants to extend their website for placing and managing online orders as a part of their effort in becoming the #1 online pizza shop. They have also found it increasingly useful to build an application for smartphones. The application is a Web application allowing you to choose and buy a Pizza online. They have subcontracted the development of the smartphone application to FunkyApps LLC. John Doe, Chief Architect of FunkyApps had some interesting feedback for PizzaShack. He suggested that the company considers monitoring of consumer statistics and probably looking into complex event processing in the future. John also suggested that they make use of an API Store backed by a modern API Gateway providing security features such as OAuth 2.0 access tokens.

In order to achieve this, PizzaShack will be implementing WSO2 API Manager and a number of other WSO2 products for monitoring statistics, single sign-on and so on.

The application leverages an API with 3 resources, which are exposed via the API Manager. Corresponding services are hosted in the WSO2 API Manager.

## High-Level Steps

- Install WSO2 API Manager
- Other installations
- Overview of the key directories in WSO2 API Manager
- Key configuration files
- Configure port offsets

# Detailed Instructions

## Install WSO2 API Manager

Before installing the product, ensure that the installation prerequisites have been met. Refer to the documentation [1] for detailed instructions. If prerequisites are fulfilled, instructions on installing the product can be found for:

- Linux or OS X at [2]
- Windows [3]

[1] Installation Prerequisites
[2] Installing on Linux
[3] Installing on Windows

## Other Installations

In order to complete the use case described in this lab kit the following products must be installed: A Rest API client or cURL [1], CLI tool [2] (Dev-Ops Tooling)

[1] https://www.getpostman.com/apps
[2] https://wso2.com/api-management/tooling/

**For MAC OS X**

Installing brew[1], JDBC driver for MySQL[2] and cURL[3]

[1] Install brew from https://brew.sh/
[2] In the terminal run

- `brew tap gbeine/homebrew-java`
- `brew install mysql-connector-java`

[3] In terminal run `‘brew install curl’`

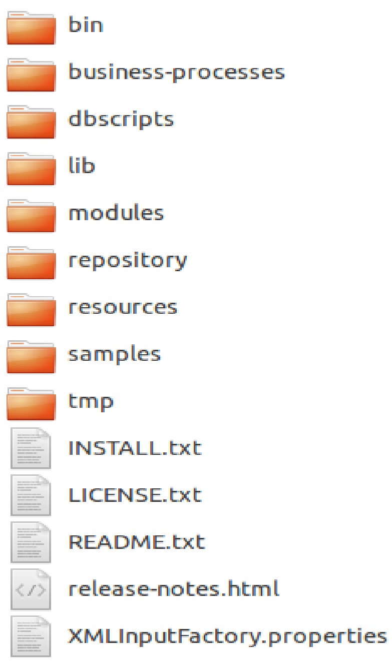
**For Windows**

Installing JDBC Drivers for SQL[1]

[1] Install Drivers from this location
[2] Install cURL from this location

# Overview of the Key Directories in WSO2 API Manager

The structure of the <APIM_HOME> folder is as follows.



bin
business-processes
dbscripts
lib
modules
repository
resources
samples
tmp
INSTALL.txt
LICENSE.txt
README.txt
release-notes.html
XMLInputFactory.properties

**bin** - This folder contains all the executable files including those scripts that are used to start/stop the application on Linux and Windows environments e.g., wso2server.sh and wso2server.bat.

**business-processes** – This folder contains information related to business process execution for API Management related operations. With WSO2 API Manager we have added 4 workflow plug points for the below operations: user creation process, application creation process, application registration process, subscription process.

**dbscripts** – A collection of database scripts required to create the Carbon database and the API Manager specific database on a variety of database management systems.

**lib** –The lib directory houses all the jar files that will be converted to OSGi bundles at startup and copied to the dropins directory.

**modules** – All the host objects belonging to the Jaggery module are declared within the modules folder in a file called module.xml.

**repository** – The main repository for all kinds of deployments and configurations in Carbon. This includes all default services, created APIs, Carbon configurations etc.

**resources** – Contains additional resources that may be used by the API-M.

**samples** - Sample APIs that can be used to explore the WSO2 API Manager functionality.

**tmp** – Will contain temporary files that are created when a product is run. These files will be cleared from time to time based on housekeeping tasks.

## Key Configuration Files

| File | Description |
|---|---|
| <PRODUCT_HOME>/repository/conf/deployment.toml | The main configuration file. |
| <PRODUCT_HOME>/repository/conf/log4j2.properties | The log4j2 configuration file used by WSO2 Carbon. |

## Configure Port Offset

When you run multiple WSO2 products, multiple instances of the same product, or multiple WSO2 product clusters on the same server or virtual machines (VMs), you must change their default ports with an offset value to avoid port conflicts. The default HTTP and HTTPS ports (without offset) of a WSO2 product are 9763 and 9443 respectively. Port offset defines the number by which all ports defined in the runtime such as the HTTP/S ports will be changed. For example, if the default HTTP port is 9763 and the port offset is 1, the effective HTTP port will change to 9764. For each additional WSO2 product instance, you set the port offset to a unique value. The default port offset is 0.

There are two ways to set an offset to a port:

- Pass the port offset to the server during startup. The following command starts the server with the default port incremented by 3 : **/wso2server.sh -DportOffset=3**
- Set the Ports section of **<PRODUCT_HOME>/repository/conf/deployment.toml** as follows:

```
[server]
offset=3
```

We will be using API-M and APIM Analytics for these exercises. Since both these servers need to run in the same machine for this demo, we must change the port offset in the **<PRODUCT_HOME>/repository/conf/deployment.toml** file. Enter the following port offsets for each product:

| File | Port Offset |
|---|---|
| <API-M_HOME>/repository/conf/deployment.toml | 0 |
| <Analytics_HOME>/conf/worker/deployment.yaml | 1 |

Follow documentation [1] to run WSO2 API Manager and configure API Analytics.
[1] Documentation: Running the Product
[2] Configuring with APIM Analytics

## Expected Outcome

As the API-M was not given an offset, it will run on the default port while the other products will run on the relevant port offset.