Deebendu Kumar

[Email address]

WSO2 API Manager

Install & Setup

# Overview

APIs touch every aspect of digital society, including the underlying patterns. In the future years, this will automatically generate 1 billion APIs. APIs are increasingly used to fuel all digital transformations. Changing the way our firm offers linked digital experiences to your customers, partners, and workers can be difficult and should be simpler. Integration technology is underpinning their progress, whereby services and microservices have been hard driven enterprise. APIs and API management are used to accomplish this.

WSO2 API Manager is the industry's premier open-source API management tool, providing a cost-effective and scalable way to becoming a digitally driven enterprise through APIs. WSO2 API Manager enables your company to design, secure, publish, manage, analyse, and connect API products and endpoints. It allows you to connect your data and applications throughout the enterprise and across cloud.

Monetization - API publishers will be able to make money by providing APIs and services, and they will be able to create several charging plans for the same services. You may now intercept both north south and east west traffic flowing into your microservice using WSO2 cloud native API Micro gateway (API Micro gateway).

WSO2 also equips us with the necessary tools to develop advanced and bespoke security and access management integration.

The following are few main capabilities of the product

* Develop deploy and manage APIs / API product - Your APIs will be easier to use if they are well-designed. WSO2 API Manager's API Publisher walks you through the process of creating an API and publishing it, all while conforming to the API's specifications.
* API-driven integration – You can implement an API-led integration strategy by easily combining the API management and the integration layer of the products platform.
* Make your API discoverable – Making your API easy to find will help you grow your customer base. You can use the WSO2 API Manager API publisher to create categories or use to categorize the APIs. The API developer portal includes a text-full search engine that helps your customers fine APIs easily.
* Expose developer friendly API – a well document API can help developers to be more productive in building application
* Secure your APIs - You can secure your APIs fully by using visibility control, threat protection, API payload validation, adhering to well-defined protocols, applying rate limiting policies, and verifying APIs against specifications in addition to API authentication and authorization.
* Rate limiting - Balancing the load of your system is critical to avoid system outages. WSO2 API Manager provides the capability to add rate limiting policies to your APIs. Furthermore, you can use these policies to monetize your APIs and bring revenue to your organization.
* Get more insights on your APIs - WSO2 API Managers API Analytics Dashboard provides insights into your APIs. These insights can help you to understand your customers and make important strategic business decisions.
* Integrate your microservices - The WSO2 Micro Integrator runtime enables you to host composite microservices that can harness the power of a low-code integration approach, while reaping the benefits of microservices architectures.
* Integrate systems in your enterprise - The WSO2 Micro Integrator can also be used by enterprises as a classic ESB. When deployed as an ESB, it caters to your message routing, transformation, message mediation, service orchestration, as well as service and API hosting needs.
* Leverage your teal-time data - WSO2 Streaming Integrator includes a stream flow designer and a stream processing engine with strong monitoring and analytics functions.

# Installing the API Manager Runtime

Yu must set JAVA\_HOME environment variable to point to the directory where you have installed Java Development Kit (JDK).

On Linux:

export JAVA\_HOME=/usr/java/jdk-11.0.x

export PATH=${JAVA\_HOME}/bin:${PATH}

## Starting the API-M Server

#sh <WSO-API-M>/bin/api-manager.sh

#sh <WSO-API-M>/bin/api-manager.sh start

Web portal - https://<ip-address>:9443/carbon

API Developer Portal Default Context - https://<ip-address>:9443/devportal

API Publisher Default Context- https://<ip-address>:9443/publisher

## Running the product as a Linux service

To run the product as a service, create a start-up script and add it to the boot sequence. We have created a script into WSO APIM bin folder “wso2-prod-server.sh”

## Creating new Keystores

WSO2 API Manager is shipped with a default keystore named wso2carbon.jks , which is stored in the <API-M\_HOME>/repository/resources/security directory. This keystore comes with a private/public key pair that is used for all purposes, such as encrypting sensitive information, communicating over SSL. You can either use one new keystore for all purposes, or you can create multiple keystores for each purpose.

You can follow the steps in this section to create a new keystore with a private key and a new public key certificate. We will be using the keytool that is available with your JDK installation. Note that the public key certificate we generate for the keystore is self-signed. Therefore, if you need a public key certificate that is CA-signed, you need to generate a CA-signed certificate and import it to the keystore as explained in the [next section](https://apim.docs.wso2.com/en/latest/install-and-setup/setup/security/configuring-keystores/keystore-basics/creating-new-keystores/#adding-ca-signed-certificates-to-keystores). Alternatively, you can choose the option of generating a new keystore using a CA-signed public certificate as explained [previously](https://apim.docs.wso2.com/en/latest/install-and-setup/setup/security/configuring-keystores/keystore-basics/creating-new-keystores/#creating-a-keystore-using-an-existing-certificate) .

* Open a command prompt and go to the <API-M\_HOME>/repository/resources/security/ directory. All keystores should be stored here.
* Create the keystore that includes the private key by executing the following command:
* keytool -genkey -alias newcert -keyalg RSA -keysize 2048 -keystore newkeystore.jks -dname "CN=<testdomain.org>, OU=Home,O=Home,L=SL,S=WS,C=LK" -storepass mypassword -keypass mypassword
* Open the <API-M\_HOME>/repository/resources/security/ directory and check if the new keystore file is created. Make a backup of it and move it to a secure location. This is important as it is the only place with your private key.
* Now, let's look at how you can get a CA-signed certificate for your keystores. **Note** that you do not need to create a new keystore every time you need add a CA-signed certificate.
* Execute the following command to generate the CSR:
* keytool -certreq -alias certalias -file newcertreq.csr -keystore newkeystore.jks

## Tuning Performance

To optimize network and OS performance, configure the following settings in the /etc/sysctl.conf file of Linux. These settings specify a larger port range, a more effective TCP connection timeout value, and a number of other important parameters at the OS-level.

It is not recommended to use net.ipv4.tcp\_tw\_recycle = 1 when working with network address translation (NAT), such as if you are deploying products in EC2 or any other environment configured with NAT.

net.ipv4.tcp\_fin\_timeout = 30

fs.file-max = 2097152

net.ipv4.tcp\_tw\_recycle = 1

net.ipv4.tcp\_tw\_reuse = 1

net.core.rmem\_default = 524288

net.core.wmem\_default = 524288

net.core.rmem\_max = 67108864

net.core.wmem\_max = 67108864

net.ipv4.tcp\_rmem = 4096 87380 16777216

net.ipv4.tcp\_wmem = 4096 65536 16777216

net.ipv4.ip\_local\_port\_range = 1024 65535

Type the following command to reload settings from config files without rebooting the box:

sysctl --system

To alter the number of allowed open files for system users, configure the following settings in the /etc/security/limits.conf file of Linux (be sure to include the leading \* character).

\* soft nofile 4096

\* hard nofile 65535

To alter the maximum number of processes your user is allowed to run at a given time, configure the following settings in the /etc/security/limits.conf file of Linux (be sure to include the leading \* character). Each Carbon server instance you run would require up to 1024 threads (with default thread pool configuration). Therefore, you need to increase the nproc value by 1024 per each Carbon server (both hard and soft).

\* soft nproc 20000

\* hard nproc 20000

When an XML element has a large number of sub-elements and the system tries to process all the sub-elements, the system can become unstable due to a memory overhead. This is a security risk.

To avoid this issue, you can define a maximum level of entity substitutions that the XML parser allows in the system. You do this using the entity expansion limit as follows in the <API-M\_HOME>/bin/api-manager.bat file (for Windows) or the <API-M\_HOME>/bin/api-manager.sh file (for Linux/Solaris). The default entity expansion limit is 64000.

-DentityExpansionLimit=10000

## Develop, Deploy and Manage APIs/API Products

## API-driven integration

## Make your APIs Discoverable

## Expose developer-friendly APIs

## Secure your APIs

## Rate Limiting

## Get more insight on your APIs

## Integrate your microservices

## Integrate systems in your enterprise

## Leverage your real-time data