API Gateway Runtime

API Gateway Runtime 2.x Hardware Requirements

- 2GHz, dual-core CPU, or 2 virtual CPUs in virtualized environments
- 2GB of RAM
- 4GB of storage

This hardware requirements list is a recommendation and a starting point. Different applications have different requirements. For example, CPU power depends on processing latency and limitations in the size and number of simultaneous messages determines the amount of RAM Mule needs.

API Gateway Runtime 2.x Software Requirements

Java Runtime Environments:

- Oracle JRE 1.8
- JRE 1.7.0 (recommended: JRE 1.7.0_79/80)
- IBM version 1.7
- For Microsoft Dynamics GP connector use only, the Java Cryptography Extension (JCE) 8 or JCE 7 unlimited strength jurisdiction policy files

  Install JCE 8 or JCE 7 after installing Java.

Because the only requirement for the API Gateway Runtime is Java, the API Gateway should also run on any later versions of the following operating systems, as well as any other operating systems supported by the JRE.

Supported Operating Systems

API Gateway Runtime 2.x validates against the following operating systems:

- Mac OS 10.10.0
- HP-UX 11i V3
- IBM AIX 7.1

- Microsoft Windows 2012 R2 Server
- Microsoft Windows 8.1
- Solaris 11.2
- RHEL 7
- Ubuntu Server 14.0.4

About API Gateway Runtime 2.x

An API Gateway runtime is generally used with Mule 3.7.x and earlier. API Gateway runtime was designed and optimized to host an API or to open a connection to an API deployed to another runtime. The API Gateway runtime performs a number of functions critical to running and managing APIs.

- Serves as a point of control over APIs, determining which traffic is authorized to pass through the API to backend services, to meter the traffic flowing through, to log all transactions, and to apply runtime policies to enforce governance like rate limiting, throttling, and caching.
- Integrate APIs with the backend services that power them. An API is just an interface that calls functionality living in a service or application, and unless that functionality lives in a well-defined web service, integration and orchestration capabilities are required to connect it to the API.

Capabilities and Limitations of API Gateway runtime

API Gateway runtime is limited in the following ways:

Connectors	• HTTP/S • Jetty • Web Services Consumer • JDBC File
Custom Connectors	Not supported
Integration Capabilities	• Message processors • Transaction management

	• Error handling • Mule Expression Language (MEL) • DataWeave (Transform Message) • DataMapper • DataWeave
Batch Processing	Not supported

To use other connectors or components you must purchase a license for Mule Enterprise. Contact a sales representative for more information.

API Gateway Deployment

API Gateway runtime can be deployed to the cloud or on premises. Deciding on the right environment for your use case depends on a number of factors including the location of backend endpoints, enterprise architecture, and corporate security policy. Gateways can be deployed as single nodes or in clusters to support HA and high throughput use cases.

Starting and Stopping API Gateway

To run API Gateway in the background:

1. Use the `cd` command to navigate to the `<MULE_HOME>/bin` directory.
2. Run the startup script according to the options below.
   - **Windows:** `gateway.bat start`
   - **Linux:** `./gateway start`

To start API Gateway in the foreground:

1. Navigate to the `<MULE_HOME>/bin` directory.
2. Run the startup script with the start option.
   - **Windows:** `<MULE_HOME>\bin\gateway`
   - **Linux:** `<MULE_HOME>/bin/gateway`

To stop the gateway, run the script with the stop parameter.

Note:

- ✓ The advantage of using a proxy is having a layer of separation to ensure that any attacks against our API are stopped well before anyone interacts with our main servers.
- ✓ This creates extra protection for our existing APIs. API Gateway acts as a proxy server that is dedicated to hosting the proxy applications and for Lightning Connect to gather all the existing backend APIs together which are either hosted in our on-premises standalone server or CloudHub.
- ✓ The advantage of API Gateway is that it automatically creates an application that can proxy the backend API from the URL that the backend API exposes, and we do not need to write any code for it.
- ✓ Not only that, through API Manager, we can implement various runtime policies on HTTP/HTTPS endpoints to govern our proxy API.
- ✓ API Gateway solution is also very versatile, as it can be implemented both in CloudHub as well as in on-premises.