Sunil Kumar

Sunil.k@neudesic.com

Abstract

Loren ipsum

Mule flex gateway

[Document subtitle]

Contents

[Introduction 2](#_Toc143127900)

[Flex gateway demonstration using Docker. 3](#_Toc143127901)

[Prerequisites 3](#_Toc143127902)

[Install, register, and run Flex Gateway 3](#_Toc143127903)

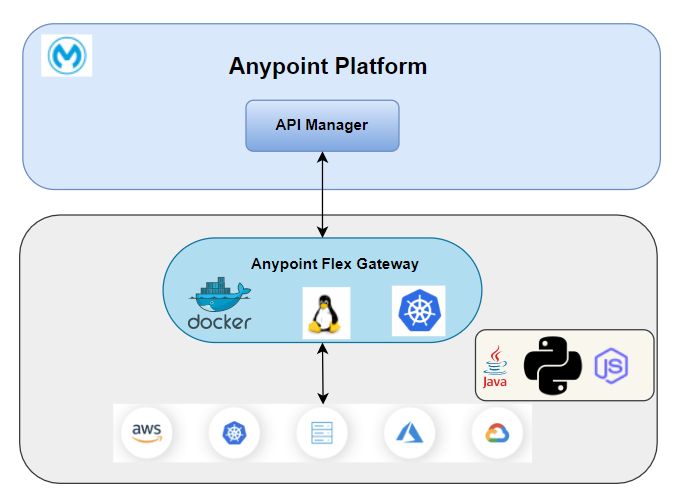
[Containerizing Mule runtime 4](#_Toc143127904)

[Installing runtime manager agent. 6](#_Toc143127905)

[Installing and testing a Mule API deployed 8](#_Toc143127906)

# Introduction

Anypoint Flex Gateway is MuleSoft’s new API gateway offering to manage and secure APIs running anywhere (cloud, on premise) in any tech stack (java, .net, python, etc.)

With the new Anypoint flex gateway, Organizations will be able to manage and govern all the APIs under one roof.

Anypoint Flex Gateway can be run in two ways. We will be using the connected mode for this tutorial.

* Connected Mode – In this mode API management is done using Anypoint control plane.
* Local Mode – API Management is done locally using stored configuration files.

# Flex gateway demonstration using Docker.

In this tutorial, we will –

1. Install Mule flex gateway image in a docker container.
2. Register the flex gateway to Anypoint platform.
3. Manage and apply policies to a ule application

## Prerequisites

* Anypoint Platform account with valid credentials
* Valid Flex Gateway permissions on Anypoint account
* Docker Desktop

## Install, register, and run Flex Gateway

1. Log in to Anypoint Platform and navigate to **Runtime Manager**. Click on the **Flex Gateways** tab on the left and select **Add Gateway**. This will present different options to select where to set up the Flex Gateway.
2. Select **Container** and then **Docker**. This will dynamically change the webpage and show some commands to setup the gateway. These commands should be executed

A screenshot of a computer

Description automatically generated

1. Execute the below docker command to pull the flex gateway image in docker.

*docker pull mulesoft/flex-gateway*

1. Verify the image was correctly installed by executing following command.

*docker images*

1. Register Flex Gateway in docker to Anypoint Platform by running the below command after replacing,

**<gateway-registration-folder>** with the folder name where registration detail will be saved.

**<organization-id>** with the value of organization id.

**<access-token>** with the value of access token.

**<gateway-name>** with the name of flex gateway.

If docker is running in Windows -

*docker run --entrypoint flexctl -v "<gateway-registration-folder>":/registration mulesoft/flex-gateway register --organization=<organization-id> --token=<access-token> --output-directory=/registration --connected=true <gateway-name>*

If docker is running in Linux

*docker run --entrypoint flexctl -u $UID -v "<gateway-registration-folder>":/registration mulesoft/flex-gateway register --organization=<organization-id> --token=<access-token> --output-directory=/registration --connected=true <gateway-name>*

Note - Copy the command from Runtime Manager to register your gateway. It already contains the token and organization ID you need to authenticate and connect the gateway with your Anypoint Platform account.

Once above command executes, it creates a new flex gateway in the Anypoint Platform.

Start the gateway by running the below command after replacing,

**<gateway-registration-folder>** with the folder name where registration detail was saved using previous command.

*docker run --rm -v "<gateway-registration-folder>":/usr/local/share/mulesoft/flex-gateway/conf.d -p 8081:8081 mulesoft/flex-gateway*

A screenshot of a computer

Description automatically generated

## Containerizing Mule runtime

Containerizing Mule Runtime involves packaging the Mule Enterprise Runtime environment, along with dependencies, and configurations, into a Docker container.

1. Download [Mule standalone runtime](https://www.mulesoft.com/lp/dl/anypoint-mule-studio). It will be downloaded as a zip file.
2. Create a file in the same directory where mule runtime got downloaded, and name it as **Dockerfile**. This file will contain instructions for building a docker container image. Copy and paste the following content in **Dockerfile** and replace,

**<mule-runtime-zip-filename>** with the name of zip file that got downloaded.

*FROM openjdk:8-jdk*

*ENV MULE\_RUNTIME\_ZIP=<mule-runtime-zip-filename>*

*ENV MULE\_HOME=/opt/mule*

*ADD $MULE\_RUNTIME\_ZIP /opt*

*RUN set -x && cd /opt && unzip $MULE\_RUNTIME\_ZIP && mv mule-enterprise-standalone-\* mule && $MULE\_HOME/bin/mule -installLicense $MULE\_HOME/conf/$LICENSE\_FILE*

*WORKDIR $MULE\_HOME*

*VOLUME $MULE\_HOME/apps*

*VOLUME $MULE\_HOME/conf*

*VOLUME $MULE\_HOME/domains*

*VOLUME $MULE\_HOME/logs*

*#Check if Mule License installed*

*RUN ls -ltr $MULE\_HOME/conf/*

*CMD echo "------ License installed ! --------"*

*# HTTP Service Port*

*# Expose the necessary port ranges as required by the Mule Apps*

*EXPOSE 8082-8091*

*EXPOSE 9000*

*EXPOSE 9082*

*# Configure external access:*

*# HTTPS Port for Anypoint Platform communication*

*EXPOSE 443*

*# Mule remote debugger*

*EXPOSE 5000*

*# Mule JMX port (must match Mule config file)*

*EXPOSE 1098*

*# Mule MMC agent port*

*EXPOSE 7777*

*# AMC agent port*

*EXPOSE 9997*

*# Mule Cluster ports*

*EXPOSE 5701*

*EXPOSE 54327*

*# HTTP Service Port*

*EXPOSE 8081*

*# HTTPS Service Port*

*EXPOSE 8091*

*# Start Mule runtime*

*CMD echo "------ Start Mule runtime --------"*

*ENTRYPOINT ["./bin/mule"]*

1. Open a terminal, navigate to the directory containing **Dockerfile** and Mule Runtime zip file, and run the following command to build the docker container image after replacing **<image-tag-name>** with the appropriate values.

*docker build -t <image-tag-name> -f Dockerfile .*

1. Once the image is built, run the container by executing below command after replacing,

**<container-name>** and **<image-tag-name>** with the appropriate values

*docker run -d --name <container-name> -p 9000:9001 -i -t <image-tag-name>*

Note: -p 9000:9001: This option maps a port from the host to a port in the container. In this case, it maps port 9000 on the host to port 9001 in the container. The syntax is -p host-port: container-port.

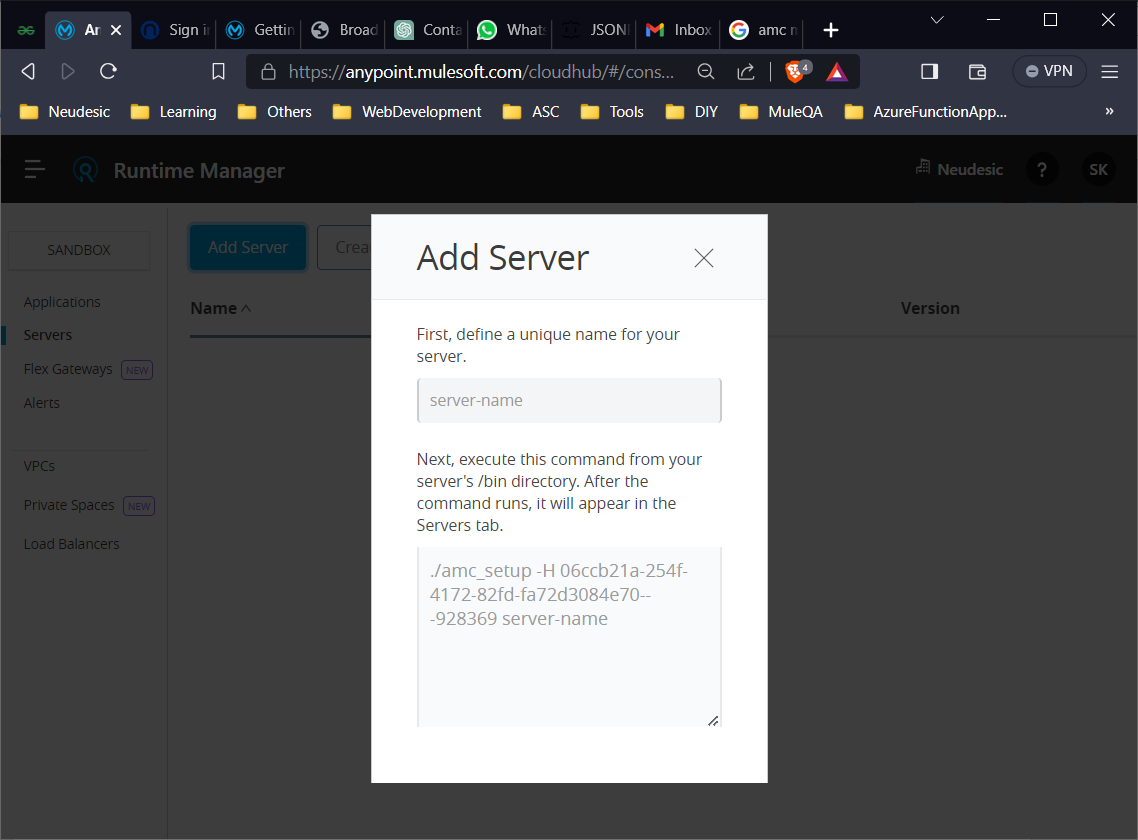
A computer screen shot of a program

Description automatically generated

## Installing runtime manager agent.

The Anypoint Runtime Manager agent registers Mule runtime engine (Mule) with Runtime Manager. After Mule is registered, you can manage it using Runtime Manager within the specific environment and Anypoint Platform business group in which Mule was registered. You cannot register Mule with multiple Runtime Manager business groups or environments.

1. Log in to Anypoint Platform and navigate to **Runtime Manager**. Click on the **Servers** tab on the left and select **Add Server**. This will open a screen similar to the one below. Enter the server’s name and copy the generated command.



1. Open terminal and execute the below command after replacing **<container-name>** with appropriate value. This will open an interactive container terminal.

*docker exec -it <container-name> /bin/sh*

1. Execute the following commands one after other. This will install runtime manager agent and registers Mule runtime engine (Mule) with Runtime Manager

*cd bin*

*<command-copied-from-above-step>*

1. Restart the Mule runtime container. This will start the server created in runtime manager.

*docker restart <container-name>*

1. Verify the server created in runtime manager.

A screenshot of a computer

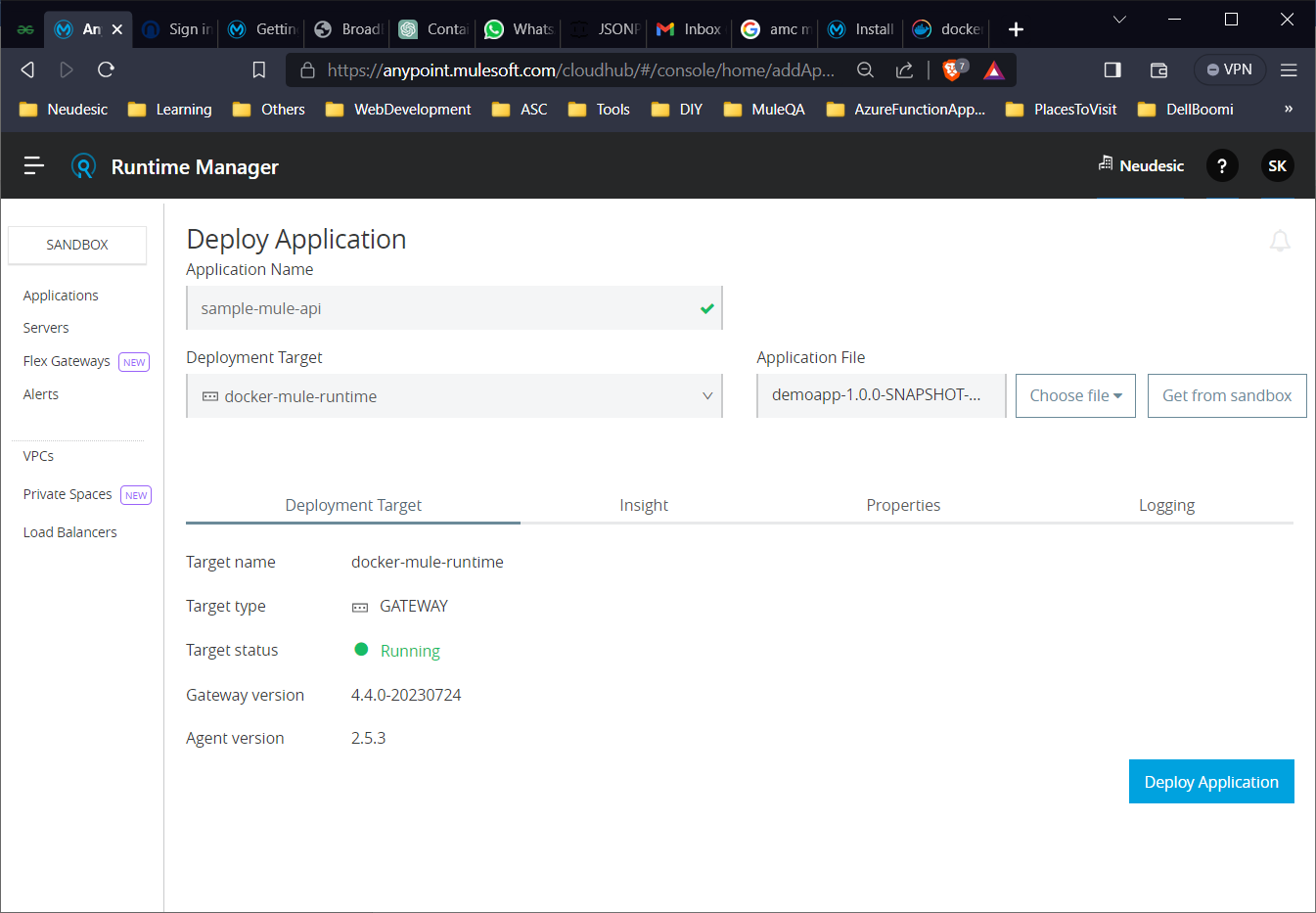
Description automatically generated

## Installing and testing a Mule API

1. Log in to Anypoint Platform and navigate to **Runtime Manager**. Click on the **Applications** tab on the left and select **Deploy application**. This will open a screen similar to the one below. Enter the appropriate values and click on Deploy Application.

Note: The application which is getting deployed is listening at port **9001**, base path as **/api** and path as **/employees**. So to make the http call we will follow [**http://localhost:9000/api/employees**](http://localhost:9000/api/employees). Port 9000 because container is mapping 9000 to 9001 already. Use the jar file below to deploy on runtime.





1. Once the application is deployed. Try to browse [**http://localhost:9000/api/employees**](http://localhost:9000/api/employees)**.**

Applying policies