WSFS Enterprise Integration Platform

***VPC/VPN Assessment and Requirements Report***

***Table of Contents***

[***1. Purpose***](#_2et92p0) ***4***

[***2. Intended Audience***](#_3dy6vkm) ***4***

[**3. Anypoint Virtual Private Cloud (VPC)**](#_2s8eyo1) **4**

[3.1 Overview](#_now5uicsitt2) 4

[3.2 VPC architecture diagram](#_uv8jr3gmuxrc) 5

[3.3 Advantages](#_dz3pgm2aml5g) 5

[3.4 Considerations](#_bmg0suim9hv1) 5

[3.5 Anypoint VPC characteristics](#_vncb7voxogo7) 5

[3.6 Steps to setup Anypoint VPC](#_vflii1gw3atl) 6

[**4. Anypoint VPN**](#_c35oyclwz636) **7**

[**4.1 Overview**](#_ucdnw1aov8gz) **7**

[4.2 VPN (IPsec Tunnel) diagram](#_8wpzx7vpgh1h) 7

[4.3 Checklist for Creating an Anypoint VPN](#_lehcv1al2kv2) 7

[4.4 Steps to setup Anypoint VPN](#_lf0bc6k2b0li) 7

[4.5 How to generate interesting traffic for Anypoint VPN (verify vpc-vpn connectivity)?](#_3cyx4of0938r) 9

[**5. Dedicated Load Balancer (DLB)**](#_qlejl1jyiygf) **9**

[**5.1 Overview**](#_nc7dsxw9di2f) **9**

[5.2 Benefits](#_vlsedt7vprp9) 10

[5.3 Steps to setup Load Balancer](#_jlan8aw53p3k) 10

[**6. Set-up MuleSoft Runtime On-Premise (Hybrid)**](#_id0gzqoe69mn) **13**

[6.1 Minimum Hardware Requirements](#_o404uq8pwwof) 13

[6.2 Required Prerequisite Software](#_l9jkmwib4vu7) 13

[6.3 Steps to download and install MuleSoft](#_dxydo8u6bmol) 13

[6.4 Running MuleSoft](#_kb3ocnnd50p8) 14

[6.5 Download the License Key File](#_o6sqc3wd0t1h) 14

[6.6 Install an Enterprise License](#_coxmkua9fogw) 14

[6.7 Verify or Remove Enterprise Edition License](#_h6qh24li4v2) 14

[6.8 Add Servers to Runtime Manager](#_g85lx1btqsvs) 14

[6.9 Add a Server](#_tqfm8jvhlh63) 15

[6.10 Remove a Server from Another Runtime Manager Instance](#_hrpcc22v13sj) 15

[6.11 Shut Down, Restart, or Delete Servers](#_miqvoosf423d) 16

[6.12 Start and Stop MuleSoft Runtime](#_cmm3u8t4abuk) 16

[6.13 Start Mule](#_9hz4cdr11kbb) 17

[6.14 Deploy MuleSoft Applications](#_3k7ccc83e7kx) 17

[6.15 Deploy Applications to On-Premises MuleSoft Instances](#_chwr5i4ogxaz) 17

[6.16 Add the MuleSoft Maven Plugin to a MuleSoft Project](#_h2xljiqftoz6) 18

[6.17 Compatibility](#_srmu51g7cy3n) 18

[6.18 MuleSoft Maven Plugin Goals](#_48zfz8dl4gdh) 18

[6.19 Package Goal](#_oqchq9in3rqa) 18

[6.20 Deploy Goal](#_s7sai3aynrqd) 19

[6.21 Undeploy Goal](#_rn8w3uwam0uc) 19

[6.22 Deploy a MuleSoft Application to a Standalone MuleSoft Runtime Engine](#_uttxlep0qcx) 20

[6.23 Standalone Deployment Parameters Reference](#_oc29zhlgm6it) 20

[**7. Reference Links**](#_paf5mgh8ok2q) **21**

Introduction

# 1. Purpose

This document sets the context and provides guidelines as to how to gather VPC/VPN requirements and setup the MuleSoft instance on an Enterprise

# 2. Intended Audience

The primary audience for these guidelines are:

* Enterprise Architects
* Solution Architects
* Application Designers, Developers and Team leads
* Application Technical Design Architects (TDAs)

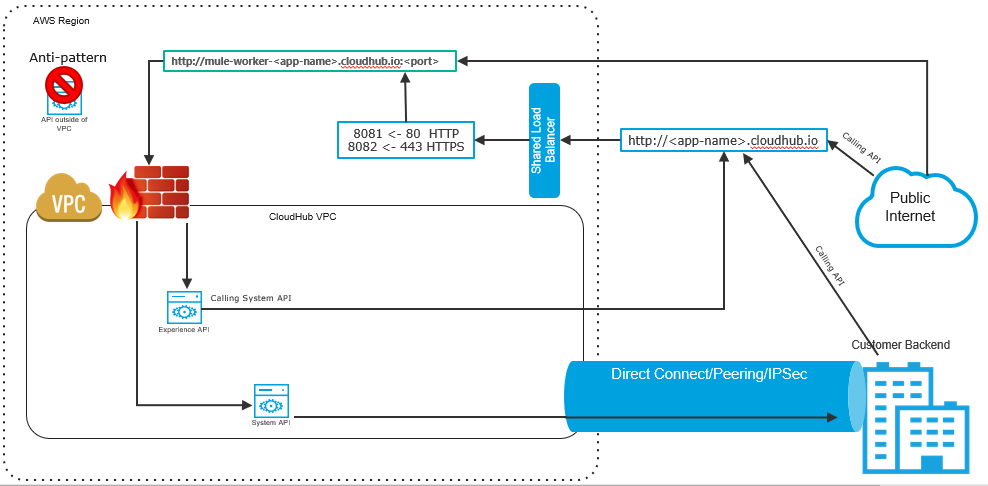
# 3. Anypoint Virtual Private Cloud (VPC)

## 3.1 Overview

Anypoint VPC allows to create a private isolated network in the cloud to host cloudhub workers. Basically, it is an isolated network segment specific to a customer & it is hosted in the AWS account which is being managed by MuleSoft.Mule applications deployed to the VPC can communicate with each other using the VPCs private network addresses.

*MuleSoft VPC is a customized/extended version of AWS VPC.*

## 3.2 VPC architecture diagram

****

## 3.3 Advantages

* It’s an enabler to securely connect with corporate data centers and on-premise applications
* It creates secure virtual networks within CloudHub
* It provides an option to choose industry standard encryption standards IPSec or SSL, and secure the network at the hardware or software levels

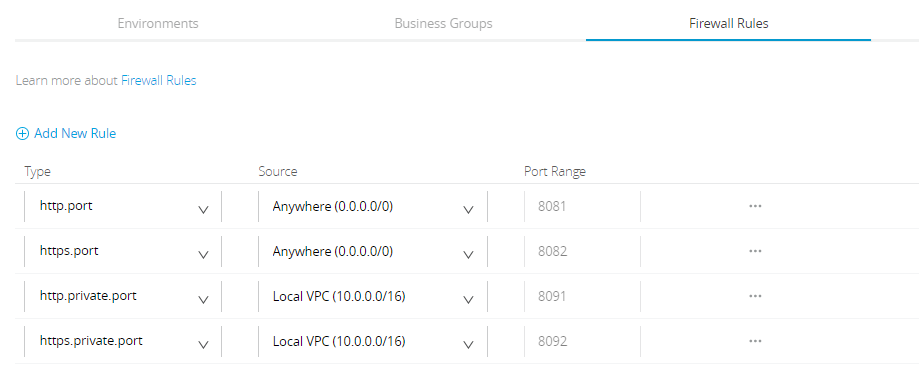
## 3.4 Considerations

* VPCs cannot be readjusted once it is created. CIDR range & Location will not be allowed to change
  + *Note: Only Firewall rules, DNS, & associated environments are configurable*
* 2 VPN connections are included in Base Subscription
* At the time of VPC creation, it is recommended to specify a CIDR block from the private IPv4 address ranges as specified in RFC 1918
* The safe rule of thumb for deciding the size of your Anypoint VPC subnet is to calculate 10 times the maximum number of expected apps to deploy in the VPC

## 3.5 Anypoint VPC characteristics

* Multiple VPC can be created in the same region
* Always create VPC in the same region or near to your datacenter or AWS region (VPC peering)
* All non-prod environments like DEV, SIT,UAT can be mapped to “non prod” vpc and production environment to “prod” vpc
* Multiple environments can be mapped to the same VPC

## 3.6 Steps to setup Anypoint VPC

* Sign into Anypoint Platform account with the Organization Administrators role
* Under Management Center, click Runtime Manager & navigate to VPC section
* Click Create VPC, and enter the following information to define and configure the Anypoint VPC :
  + **Name**: The name to identify your Anypoint VPC
  + **Region**: The region to which the Anypoint VPC is bound
  + **CIDR Block:** The size of the Anypoint VPC in Classless Inter-Domain Routing (CIDR) notation
    - For example, if you set it to 10.111.0.0/24, the Anypoint VPC is granted 256 IP addresses from 10.111.0.0 to 10.111.0.255.
    - Ideally, the CIDR Blocks you choose for the Anypoint VPC come from a private IP space, and should not overlap with any other Anypoint VPC’s CIDR Blocks, or any CIDR Blocks in use in your corporate network
  + **Environments:** Select an environment to which the Anypoint VPC is bound
  + **Business Groups:** Bind the Anypoint VPC with a business group
  + **Configure Firewall Rules :** Click on it to expand the fields and configure firewall rules
    - By default, all inbound traffic is blocked, and you need to configure firewall rules to allow traffic to your worker. You can configure these rules at a later time.The UI is pre-filled with suggestions of the most commonly used firewall rules
    - In general, we can configure 4 different rules as shown in the image below:
      * **Rule 1:** Any application running on 8081 can be reached from anywhere via http
      * **Rule 2:** Any application running on 8082 can be reached from anywhere via https
      * **Rule 3:** Any application running on 8091 can be reached from any application running inside the same Anypoint VPC. If these applications need to be exposed outside the Anypoint VPC they can be exposed using Dedicated Load Balancer
      * **Rule 4:** Any application running on 8092 can be reached from any application running inside the same Anypoint VPC. If these applications need to be exposed outside the Anypoint VPC, they can be exposed using Dedicated Load Balancer
  + **Internal DNS** : Click on it to set up internal DNS servers to resolve your private host names

All set, click on the “**Create VPC**” button to complete the setup process.

# 4. Anypoint VPN

## 4.1 Overview

Anypoint VPN helps to create a secure connection between MuleSoft VPC and customer hosted on-premises networks. It supports site-to-site Internet Protocol security (IPsec) connections.

Each Anypoint VPN connection consists of two tunnels that help to connect to a single public IP address at a remote location.

VPN licenses are bundled with the VPC license, so each VPC license also entitles the organisation to 1 VPN gateway, with a connection to 1 public IP address in a remote location.

That is to say, 1 VPC License = 1 VPC entitlement + 1 VPN entitlement.

## 4.2 VPN (IPsec Tunnel) diagram

## 4.3 Checklist for Creating an Anypoint VPN

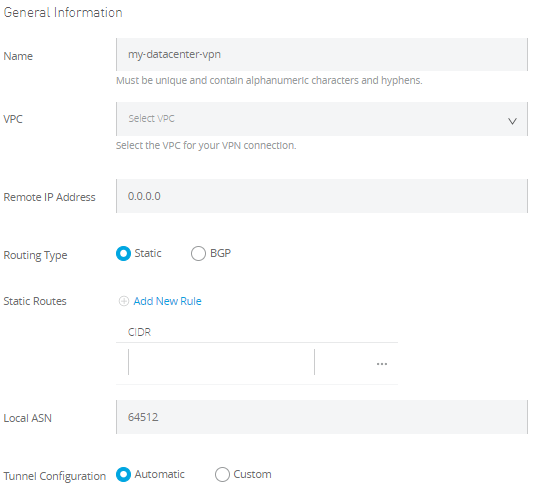
* Remote IP address : The public IP address of your VPN endpoint. This must be a single, static IP address
* Static Routes (CIDR): The subnets in your network to make accessible through the VPN. This information is required only if you are using static routing
* Remote and Local ASN: The Autonomous System Number specifies the collection of routing prefixes. You must configure both a remote and local ASN. This is required only if you are using dynamic routing
* Pre-shared Key (PSK): The shared secret for the VPN tunnels. These values are auto-generated if you select Automatic Tunnel Configuration
* Point-to-Point CIDR: Private IP range for the VPN tunnel interfaces. These values are auto-generated if you select Automatic Tunnel Configuration

## 4.4 Steps to setup Anypoint VPN

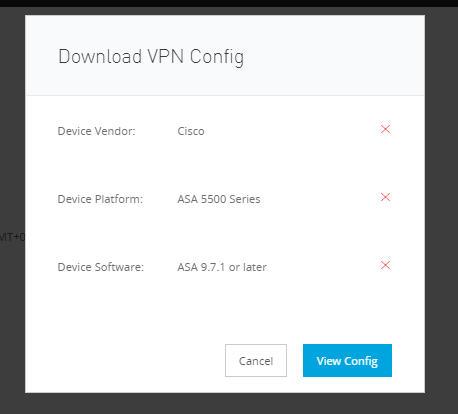
* Sign into Anypoint Platform and select Runtime Manager
* Select the environment where you want to create an Anypoint VPN
* From the menu on the left, click VPNs, and then click Create VPN
* Enter or select the following information for your Anypoint VPN:
  + **Name**: Enter a name for your Anypoint VPN
  + **VPC**: From the drop-down list, select the Virtual Private Cloud for the Anypoint VPN connection
  + **Remote IP Address**: Enter the public IP address of your VPN endpoint (customer gateway public IP)
  + **Routing Type**: You can select either BGP (dynamic) or Static. Choose the Border Gateway Protocol (BGP) type if your device supports it
  + **Tunnel Configuration:**
    - **Automatic :** This option automatically configures the tunnel settings for Anypoint VPN.The tunnel settings are visible after VPN creation
    - **Custom :** We need to provide PSK (Pre-shared Key : The shared secret for the VPN tunnels) & Point-to-Point CIDR (the Private IP range for the VPN tunnel interfaces)
    - **Click “Create VPN”**
* The status of the VPN will change to “Pending” with Tunnel 1 and Tunnel 2 both in “Pending” state. This indicates that the VPN is being configured on the CloudHub side. This is the expected status while the infrastructure is created.

***After the status changes to AVAILABLE, continue with the next steps.***

*Note : You cannot modify tunnel settings after you create the Anypoint VPN connection. To change the settings for an existing connection, you must delete the Anypoint VPN connection and create a new one.*



* Download the configuration file from your Anypoint VPN. You have an option to select the configuration file based on your device type, either Cisco or a generic. Your Customer Gateway device vendor determines the option you select here. If it’s a Cisco device, select Cisco, or select generic for all other types of customer gateways



* Configure your VPN endpoint: Share the VPN configuration file (from Step Vii) with your VPN endpoint administrator

## 

## 4.5 How to generate interesting traffic for Anypoint VPN (verify vpc-vpn connectivity)?

* Download ***“Net Tools API 2.2.0”*** application from the following github [link](https://github.com/mulesoft-labs/net-tools-api/releases/download/2.2.0/net-tools-v2.2.0.jar)
* Deploy the application to each VPC where you wish to the verify connectivity
* After deployment of the application is done in cloudhub, it can be accessed by the web-browser and the connectivity of different destination IP addresses can easily be tested.

***Note****: Find the tutorials on using* ***“Net Tools API 2.2.0”*** *application in the following* [*link*](https://help.mulesoft.com/s/article/How-To-Use-Network-Tools-Application?r=5&ui-force-components-controllers-recordGlobalValueProvider.RecordGvp.getRecord=1)

# 5. Dedicated Load Balancer (DLB)

## 5.1 Overview

The Dedicated Load Balancer, sits inside the Client’s VPC, routes external HTTP and HTTPS traffic to multiple Mule applications deployed to CloudHub workers in a Virtual Private Cloud (VPC).

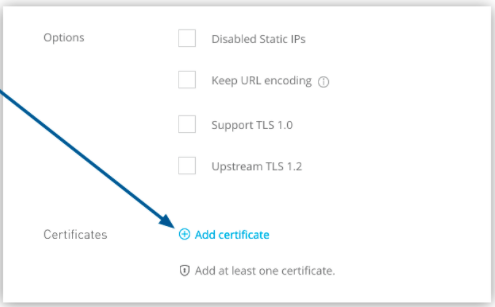
**Note :** Unlike DLB, “***Shared Load Balancer(SLB)***” doesn’t allow an user to configure a custom SSL certificate or proxy rules.There is one SLB clustered instance in each CloudHub region that serves all the CloudHub customers in that AWS region.

## 5.2 Benefits

* Allows custom TLS certificates (including 2-Way TLS Authentication)
* Define proxy rules and route requests to the appropriate REST resource
* Use a custom/vanity domain which is registered on a DNS server
* Allows IP Whitelist/Blacklist
* VPCs and DLBs can safely connect CloudHub environments with internal networks
* It can handle Mule event payloads up to 200 MB

## 5.3 Steps to setup Load Balancer

* After entering Runtime manager inside Anypoint platform, click on “Create Load Balancer”
* Enter a name for your load balancer
* Select a target Anypoint VPC from the drop-down list
* Specify the amount of time the DLB waits for a response from the Mule application in the Timeout in Seconds field (The default value is 300 seconds)
* Add any whitelisted classless inter-domain routing (CIDR) as required
* Select the inbound HTTP mode for the load balancer
* Assigning your custom domain (CNAME mapping needs to be done at the DNS server)
* Add a certificate:
  + Click Add certificate



* + On the “***Add certificate page***”, select Choose File to upload both public key and private key files
  + You can enable two-way SSL by uploading client certificates
* Define routing rules : We can define different routing rules. In the URL Mapping Rules section, we can adjust the routing mechanism

*5.4 Steps to generate CSR for multi domain SAN certificate*

**Prerequisite**

* OpenSSL must be available in the system before executing the below steps
* Details on OpenSSL can be found here <https://www.openssl.org/>

**How to install OpenSSL in Windows OS**

* Install GIT and run openssl command from Git Bash
* Git Installer can be found here: <https://git-scm.com/download/win>
* Git Bash icon looks like below



**CA Certificate Generation Steps**

**Step 1:**

* Create a file named openssl.conf
* Add below entries in the file.

[req]

default\_bits = 2048

default\_md = sha256

prompt = no

distinguished\_name = dn

req\_extensions = req\_ext

[dn]

CN=api.wsfsbank.com

C=US

ST=Delaware

L=Wilmington

O=WSFS Bank

[req\_ext]

subjectAltName=@alt\_names

[alt\_names]

DNS.1= api.wsfsbank.com

DNS.2= www.api.wsfsbank.com

DNS.3= api-uat.wsfsbank.com

DNS.4= api-sandbox.wsfsbank.com

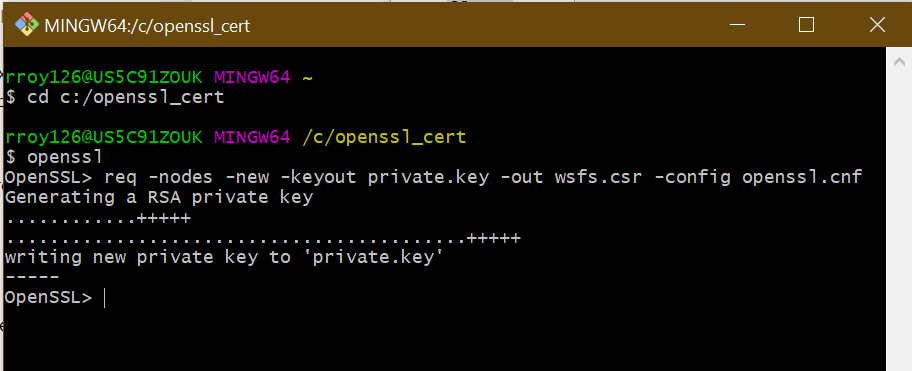
DNS.5= api-dev.wsfsbank.com

DNS.6= api-sit.wsfsbank.com

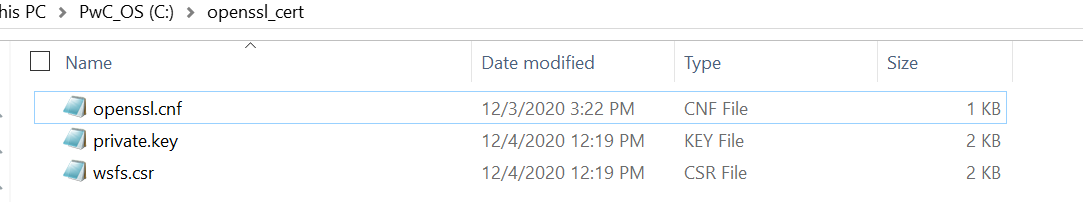
**Step2:**

* Run below openssl command which will take the above **openssl.conf** configuration file and generate a private key named **private.key** and a csr named **wsfs.csr**

**req -nodes -new -keyout private.key -out wsfs.csr -config openssl.cnf**



* Verify that private.key and wsfs.csr file generated successfully.



**Step 3:**

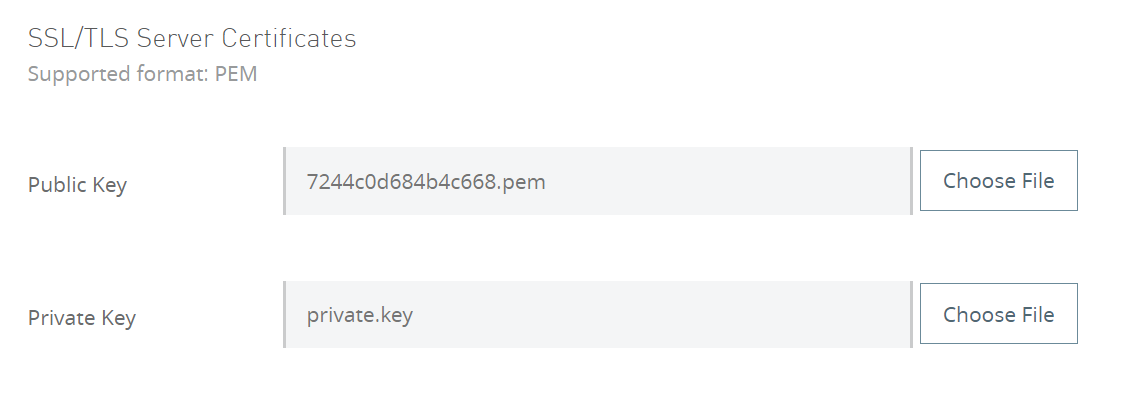
* Using this CSR (wsfs.csr), generate CA certified certificates.
* Please reach out to John Bailey from NTS team for CA certification and share this wsfs.csr file
  + You will receive the certificate in two formats from John B
    - .crt
    - .pem

These above 3 Steps will generate final private key and public key to be used in MuleSoft Cloudhub DLB

* + Please upload .pem format CA certificate received from John B / NTS team

as Public key in DLB

* + Please upload private.key as Private Key in DLB



# 6. Set-up MuleSoft Runtime On-Premise (Hybrid)

In the event of a MuleSoft CloudHub runtime outage for an entire region, the MuleSoft runtime should be installed and set-up on an on-premise instance.

**Note:** In case of a complete outage of MuleSoft in the Cloud, the applications, and MuleSoft Runtime will be installed on an on-prem server, but the MuleSoft administrative console (https://mulesoft.anypoint.com) will not be available as it is not a standalone software that can be installed.

To host a MuleSoft runtime engine in an on-premise environment, we need to meet all hardware and software requirements, and then download and install required MuleSoft software (as listed below in section 10.3).

## 6.1 Minimum Hardware Requirements

* 2 GHz CPU or 1 virtual CPU in virtualized environments
* 1 GB of RAM
* 4 GB of Storage

Adjust RAM to match latency requirements and the size and number of simultaneous messages that applications process.

## 6.2 Required Prerequisite Software

Verify that a supported version of Java is used before installing MuleSoft

* JDK 1.8.0 or JDK 11 (Recommended AdoptOpenJDK 1.8.0\_212)
* MuleSoft runs on any operating system that runs a supported Java Runtime Environment (JRE)

## 6.3 Steps to download and install MuleSoft

* Before downloading and installing MuleSoft, verify there exists a supported JDK installed. This example uses AdoptOpenJDK 8, which is recommended for MuleSoft 4.x.
* Download the MuleSoft 4 binary file from the following link and unzip it: <https://www.mulesoft.com/lp/dl/mule-esb-enterprise>
* Set an environment variable called “MULE\_HOME” for the “mule” directory inside the extracted folder.
* Example for version 4.2.0 in the Downloads directory:
  + On Windows environments:   
    $ env:MULE\_HOME=C:\Downloads\mule-enterprise-standalone-4.2.0
  + On Linux/Unix environments:  
    $ export MULE\_HOME=~/Downloads/mule-enterprise-standalone-4.2.0

## 6.4 Running MuleSoft

* Test if MuleSoft runs in the system without errors by running the following commands
* On Windows environments: %MULE\_HOME%\bin\mule.bat
* On Linux/Unix environments: $ $MULE\_HOME/bin/mule

## 6.5 Download the License Key File

* Log in to the Support portal <https://anypoint.mulesoft.com/>. Please contact the respective Customer Success Manager for any issues with login.
* Click the Subscriptions tab located on the top of the Support Portal Home page
* Click on the Subscription Name of the subscription for the license key. Please note that the Subscription Name (second column on the right) should be clicked
* Click on the License ID number found on the bottom left of the page
* Click on the View button to download license key

## 6.6 Install an Enterprise License

Complete the following steps to acquire and install an Enterprise license before using MuleSoft runtime in a production environment.

* Contact the MuleSoft account representative or the MuleSoft sales team to acquire an Enterprise license in the form of a “license.lic file”
* Before installing, it’s recommended to remove the previous license (if any) from $MULE\_HOME directory:
  + Navigate to $MULE\_HOME/conf/
  + Delete the existing muleLicenseKey.lic file.
* If installing the license on multiple platforms, back up the new license.lic file in another location before proceeding
* Make sure that the MuleSoft Server is stopped (not running) and then open the terminal or command line on the system
* On Mac/Unix/Linux, from the $MULE\_HOME/bin directory, run the following command:
  + MuleSoft -installLicense ~/license.lic
* In the $MULE\_HOME/conf directory, MuleSoft saves a new file called muleLicenseKey.lic. This shows that the license has been installed
* Start MuleSoft Server again, by the usual means

## 6.7 Verify or Remove Enterprise Edition License

* Make sure that the MuleSoft Server is stopped and then open the terminal or command line on the system
* To verify that MuleSoft successfully installed the Enterprise license, run the following command:
  + “MuleSoft -verifyLicense”
* To uninstall a previously installed license, run the following command:
  + “MuleSoft -unInstallLicense”
* If the license installation fails, manually delete $MULE\_HOME/conf/muleLicenseKey.lic file and reinstall the license key

## 6.8 Add Servers to Runtime Manager

To add a MuleSoft server to the Runtime Manager console, the first step is to register it with the Runtime Manager agent. Use the “amc\_setup” script to configure the Runtime Manager agent to communicate with Runtime Manager.

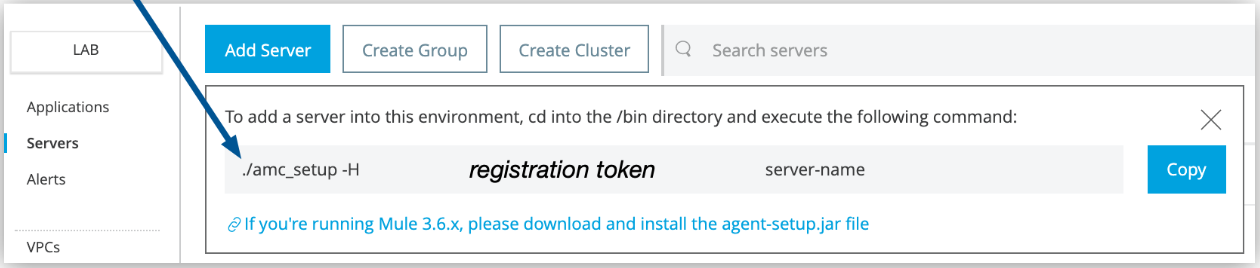
**Prerequisites:**

* The enterprise license is current
* Running MuleSoft 3.6.0 or later, and API gateway 2.1 or later
* Must download the agent, from the Enterprise support account
* If the server is already registered with another Runtime Manager instance, remove that registration first

## 6.9 Add a Server

To add a MuleSoft server to Runtime Manager, below are the steps to be followed:

* From Anypoint Platform, select Runtime Manager
* Click Servers in the left menu
* Click the Add Server button
* Runtime Manager generates the amc\_setup command, including token and server-name, to use to register MuleSoft in the environment



* The arrow shows the amc\_setup command generated by Add Server and the registration token includes organization’s ID and the current environment
* Click Copy to copy the command
* In a terminal window, change to the $MULE\_HOME/bin directory for the MuleSoft instance that is to be registered
* Paste the command on the command line
* Change the instance name “server-name” to the unique name to use to label MuleSoft in the Runtime Manager console
* The “server-name” value is unique across environments. If the same organization exists in different environments, the “server-name” can be the same in those organizations
* If the environment requires all outbound calls to go through a proxy, specify proxy settings in either the $MULE\_HOME/conf/mule-agent.yml file or the $MULE\_HOME/conf/wrapper.conf file
* When the amc\_setup command completes successfully, the below status message will be seen
  + MuleSoft Agent configured successfully
  + Connecting to Access Management to extract client\_id and client\_secret
  + Credentials extracted correctly, updating wrapper conf file
* After the script completes successfully, the server is listed on the Servers tab of Runtime Manager with a status of Created
* If the server was running when executed the amc\_setup script, restart the server to reconnect with Runtime Manager

## 6.10 Remove a Server from Another Runtime Manager Instance

If the intended server is already registered with a different instance of Runtime Manager, it must be removed before registering in the Runtime Manager

To remove the server from Runtime Manager:

* In Runtime Manager, delete the server from the Server tab
* In the $MULE\_HOME/conf directory, delete the mule-agent.yaml and mule-agent.jks files

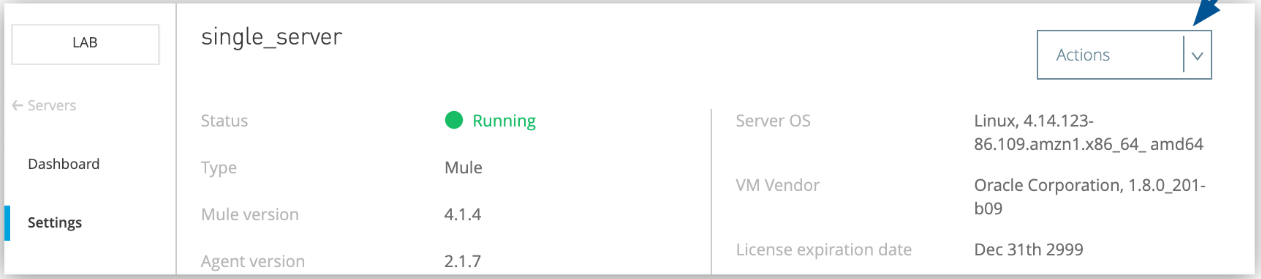
## 6.11 Shut Down, Restart, or Delete Servers

After creating a server, it can be managed from Runtime Manager. Actions that can be performed include shut down, restart, and delete the server.

After shutting down a server, it cannot be started from Runtime Manager. The server must manually be started on the system where MuleSoft runtime engine is installed

Depending on the error, if an error occurs in Runtime Manager during server shutdown or restart, there may be a need to manually start or stop the server on the system where MuleSoft is installed

* From Anypoint Platform, select Runtime Manager
* Select Servers in the left menu
* Click the server name
* Click Settings
* From the Actions menu, select one from the below options:
  + Delete
  + Restart
  + Shutdown



## 6.12 Start and Stop MuleSoft Runtime

MuleSoft uses the Java Service Wrapper to control the Java Virtual Machine (JVM) from the native OS. The wrapper provides many options and features, including the ability to run MuleSoft as a Unix daemon or install or remove MuleSoft as a Windows Service. The wrapper can handle system signals and start parameters, and overall provides much better interaction between the JVM and the underlying OS.

Navigate to the “$MULE\_HOME/bin” folder.

The following table lists all the commands that the Java Service Wrapper accepts:

|  |  |
| --- | --- |
| **Command** | **Description** |
| start | Starts MuleSoft in the terminal background. |
| stop | Stops Mule. Inbound endpoints process in-flight messages before MuleSoft shuts down. However, after the timeout period, inflight messages that haven’t completed are abandoned and MuleSoft shuts down. |
| restart | Restarts MuleSoft runtime |
| status | (Linux/Unix only) Displays the status of the MuleSoft server (running or not running) and the process PID if running. |
| dump | (Linux/Unix only) Dumps the MuleSoft wrapper’s core to $MULE\_HOME/log/mule\_ee.log. |
| console | Start MuleSoft in the terminal foreground (console mode). Same as running a MuleSoft with no parameters. |
| install | Install MuleSoft as a Windows Service or Linux/Unix Daemon. |
| remove | Remove MuleSoft from the Windows Services or Linux/Unix Daemons. |

## 6.13 Start Mule

Start MuleSoft from a command line by changing the directory to where the unzipped the MuleSoft software distribution and then changing to the bin directory. Run the following commands:

* On Windows environments: $MULE\_HOME\bin\mule.bat
* On Linux/Unix environments: $MULE\_HOME/bin/mule

These commands run MuleSoft in foreground mode, and the startup script displays information on the terminal’s standard output. Further commands cannot be issued on the terminal as long as MuleSoft is running.

To stop Mule, press CTRL-C in the terminal in which the script is running.

## 6.14 Deploy MuleSoft Applications

* Start Mule runtime
* After MuleSoft starts, deployment of MuleSoft applications can be started by moving the packaged JAR files to the apps directory under “$MULE\_HOME”

## 6.15 Deploy Applications to On-Premises MuleSoft Instances

When the applications need to be deployed on-premises, follow the below deployment methodology:

* Standalone deployment  
  By using this method, a manual deployment of the MuleSoft application will be performed to an on-premises MuleSoft instance

**Prerequisites:**

* Add the MuleSoft Maven Plugin to the project

The MuleSoft Maven plugin enables integration with packaging and deployment of the MuleSoft applications with Maven lifecycle.

## 6.16 Add the MuleSoft Maven Plugin to a MuleSoft Project

Before the start of any MuleSoft deployment operations, the MuleSoft Maven plugin should be added to the project

To add a MuleSoft Maven Plugin, the maven dependency should be added to the project

<plugin>

<groupId>org.mule.tools.maven</groupId>

<artifactId>mule-maven-plugin</artifactId>

<version>3.3.2</version>

<extensions>true</extensions>

</plugin>

The plugin will be added from the below mentioned repository:

<pluginRepositories>

<pluginRepository>

<id>mule-public</id>

<url>https://repository.mulesoft.org/nexus/content/repositories/releases</url>

</pluginRepository>

</pluginRepositories>

## 6.17 Compatibility

* MuleSoft Runtime Version 4.x
* Anypoint Studio 7.x

## 6.18 MuleSoft Maven Plugin Goals

The MuleSoft Maven plugin has three goals:

* package - Generates the jar file for the project.
* deploy - Automatically uploads and starts application in any of the application deployment targets (CloudHub, Runtime Fabric, or On-Premises).
* mule:undeploy - Automatically removes application from any of the application deployment targets (CloudHub, Runtime Fabric, or On-Premises)

Each goal accepts parameters that are unique to the desired plugin behavior. To provide a parameter from the command line, prepend -D to the parameter name.

## 6.19 Package Goal

The package goal generates the application JAR file. This goal binds by default to the Maven lifecycle phase: package

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| onlyMuleSources | boolean | Generates the application JAR file containing only the source code. This property is just for sharing purposes. |
| attachMuleSources | boolean | Attaches the source code inside the generated JAR file. |
| lightweightPackage | boolean | Doesn’t generate the repository with all the application dependencies inside the JAR file. This property is just for sharing purposes. |

For example, to execute the package goal and set the attachMuleSources parameter, run the following command:

* mvn package -DattachMuleSources

## 6.20 Deploy Goal

This goal uploads and deploys the application JAR file to any of the application deployment targets. The deploy goal binds by default to the Maven lifecycle phase: *deploy*

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| artifact | String | Path to the application JAR file to be deployed. By default is set to the application target folder. |
| muleDeploy | boolean | Instructs the plugin to deploy using the deployment strategy defined in the plugin configuration. If the muleDeploy parameter is not set, the plugin uploads the artifacts to the repository defined in the distributionManagement section of the application’s pom.xml file. |

For example, to execute the deploy goal and set the muleDeploy parameter, run the following command:

* mvn deploy -DmuleDeploy

## 6.21 Undeploy Goal

This goal removes an application from any of the application deployment targets. It uses the information from the plugin configuration to remove the application from the defined deployment target

To execute the undeploy goal, run the following command:

* mvn mule:undeploy

The undeploy goal also deletes the app in MuleSoft Maven plugin 3.3.0 and later versions

## 6.22 Deploy a MuleSoft Application to a Standalone MuleSoft Runtime Engine

* Inside the plugin element, add a configuration for the standalone deployment, replacing the placeholder values with local MuleSoft runtime engine information

|  |
| --- |
| <plugin>  <groupId>org.mule.tools.maven</groupId>  <artifactId>mule-maven-plugin</artifactId>  <version>3.3.2</version>  <extensions>true</extensions>  <configuration>  <standaloneDeployment>  <muleHome>${mule.home.test}</muleHome>  <muleVersion>${app.runtime}</muleVersion>  </standaloneDeployment>  </configuration>  </plugin> |

* From the command line in project’s folder, package the application and execute the deploy goal

## 6.23 Standalone Deployment Parameters Reference

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| standaloneDeployment | Top-Level Element |
| applicationName | Specifies the application name to use during deployment |
| muleVersion | The release version of the MuleSoft instance running on a local machine.  Ensure that this value is equal to or higher than the earliest required MuleSoft version of the application.  Example values: 4.2.0, 4.2.2-hf4  If this value does not match the MuleSoft version running in the deployment target, the plugin raises an exception.  The MuleSoft Maven Plugin does not download a new MuleSoft version if these values don’t match |
| muleHome | The location of the MuleSoft instance on a local machine |
| deploymentTimeout | The allowed elapsed time, in milliseconds, between the start of the deployment process and the confirmation that the artifact has been deployed  The default value is 1000000 |
| skip | When set to true, skips the plugin deployment goal.  Its default value is false |

# 7. Reference Links

1. <https://help.mulesoft.com/s/article/How-to-Calculate-VPC-and-VPN-License-Requirements?r=5&ui-force-components-controllers-recordGlobalValueProvider.RecordGvp.getRecord=1>
2. <https://blogs.mulesoft.com/dev/howto/setting-ipsec-vpn-data-center-and-cloudhub/>
3. <https://help.mulesoft.com/s/article/Anypoint-VPN-IKEv2-Configuration-for-Cisco-ASA-devices-using-BGP-routing?r=9&ui-force-components-controllers-recordGlobalValueProvider.RecordGvp.getRecord=1>
4. <https://docs.mulesoft.com/runtime-manager/vpc-connectivity-methods-concept>