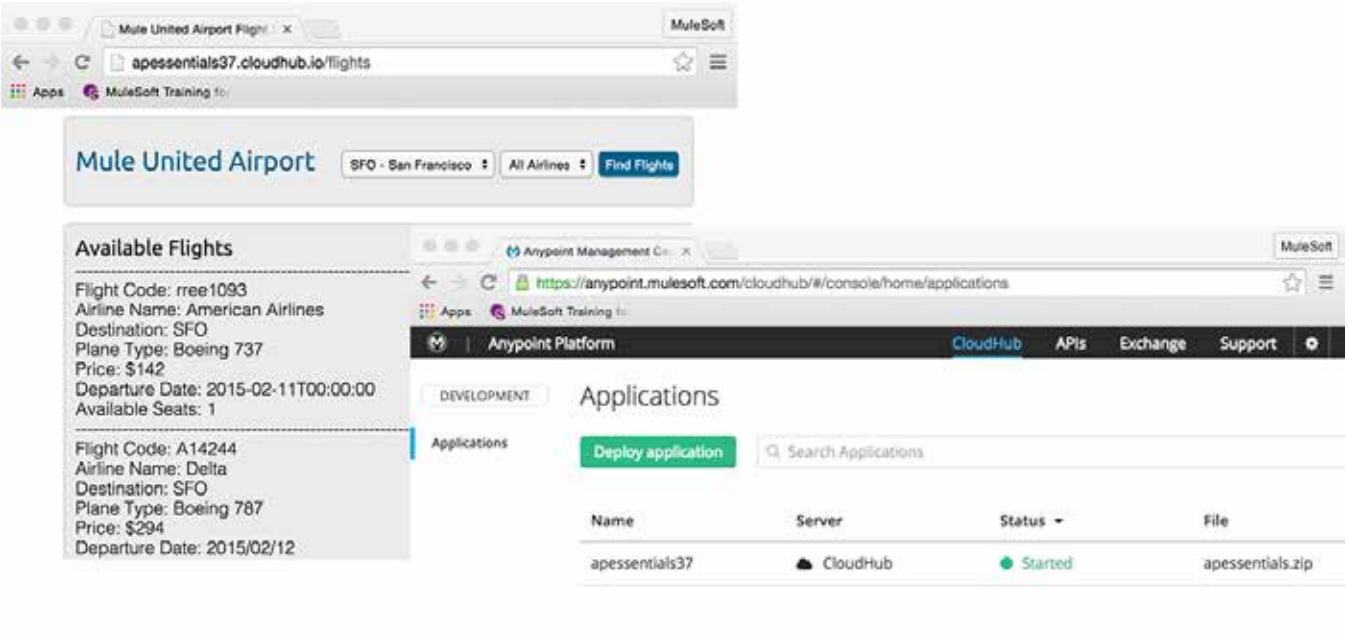
Module 11: Deploying Applications



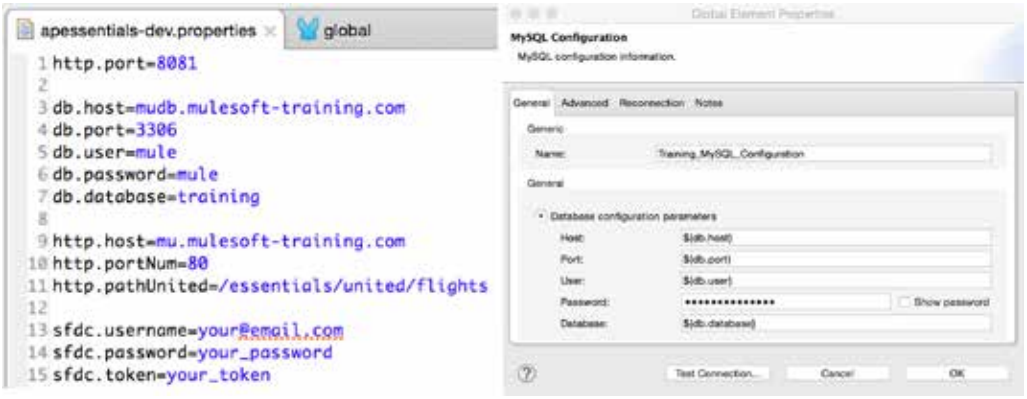
In this module, you will learn:

- About the options for deploying your applications.
- What CloudHub is.
- About when and how to use application properties.
- (Optional) To deploy and run applications on CloudHub.
- (Optional) To deploy and run applications on Mule ESB.

Walkthrough 11-1: Use application properties

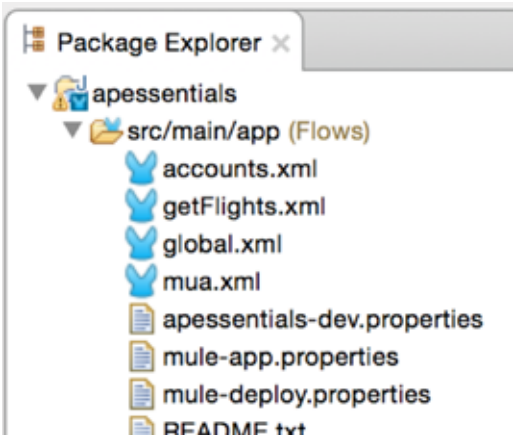
In this walkthrough, you will introduce properties into your Mule application. You will:

- Create a properties file for your application.
- Create a Properties Placeholder global element.
- Parameterize the HTTP Listener connector port.
- Define and use Database connector properties.
- (Optional) Define and use HTTP Request and Salesforce connector properties.



Create a properties file

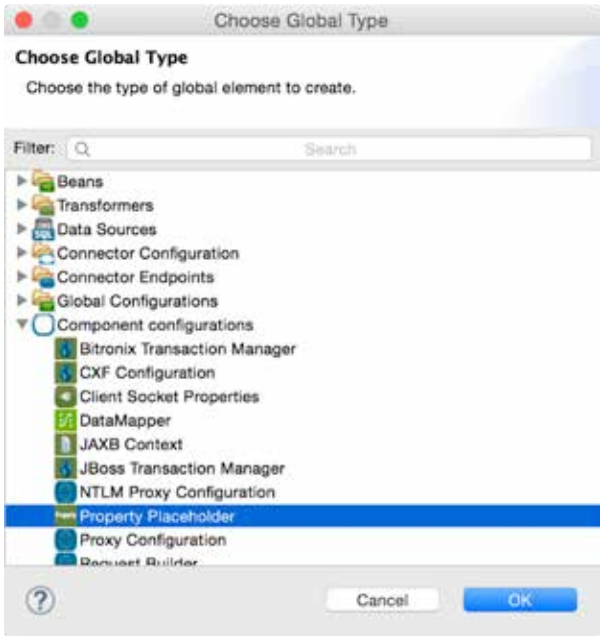
1. Right-click the src/main/app folder in the Package Explorer and select New > File.
2. Name the file apessentials-dev.properties and click Finish.



Create a Properties Placeholder global element

3. Open global.xml.
4. Navigate to the Global Elements view and click Create.

5. In the Choose Global Type dialog box, select Component configurations > Property Placeholder and click OK.



6. In the Global Element Properties dialog box, set the location to apessentials-dev.properties and click OK.

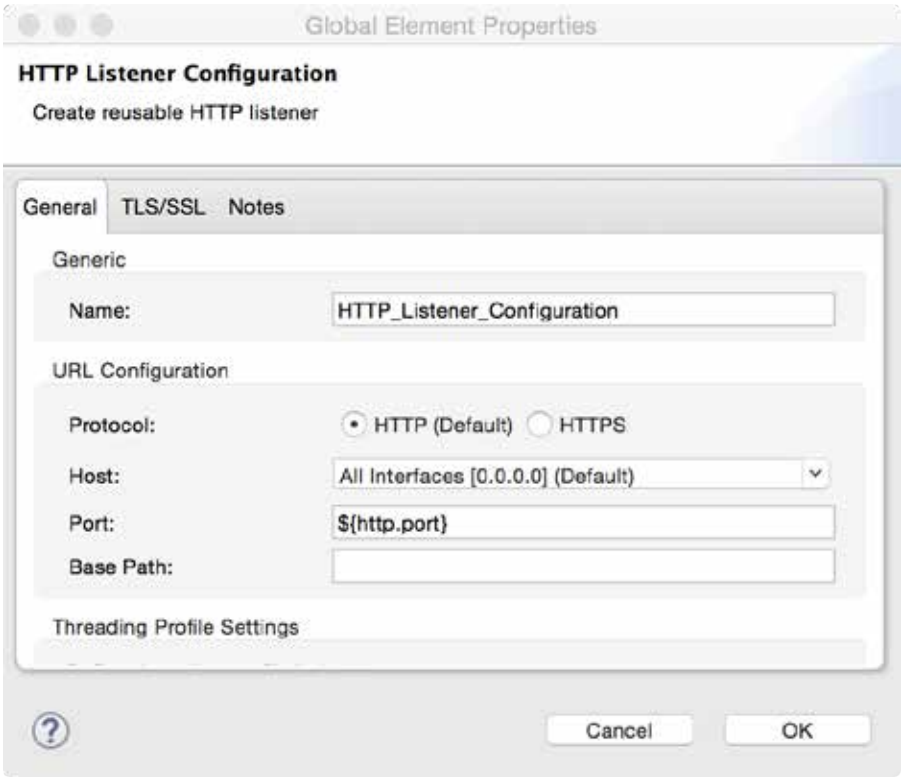
   Note: When setting the location, make sure you do not use a full path.

Parameterize the HTTP Listener port

7. Return to apessentials-dev.properties.
8. Create a property called http.port and set it to 8081.

```
http.port=8081
```

9. Return to global.xml.
10. Double-click the HTTP Listener Configuration global element.
11. Change the port to the application property, ${http.port}.
12. Click OK.

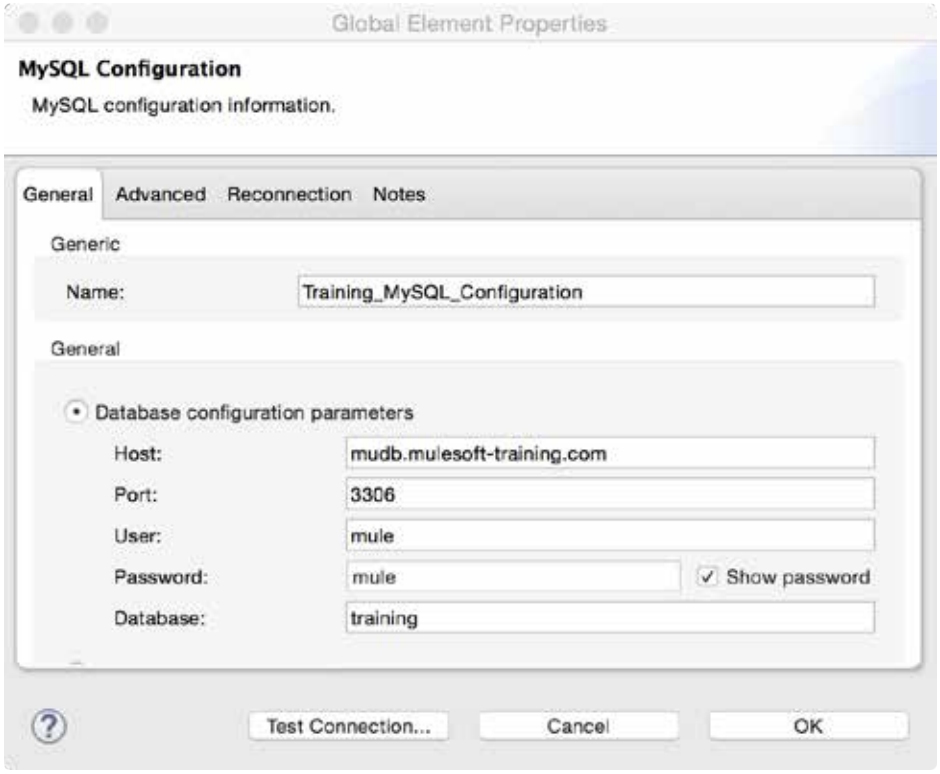


Test the application

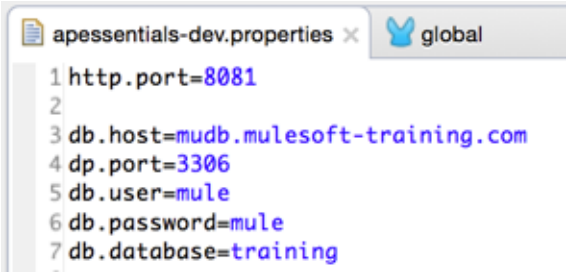
13. Save all the files and run the application.
14. Make a request to http://localhost:8081/api/flights/SFO in the APIkit Consoles view or a browser and confirm it still works.

Parameterize database credentials

15. Return to global.xml.
16. Double-click the MySQL Configuration global element to edit it.

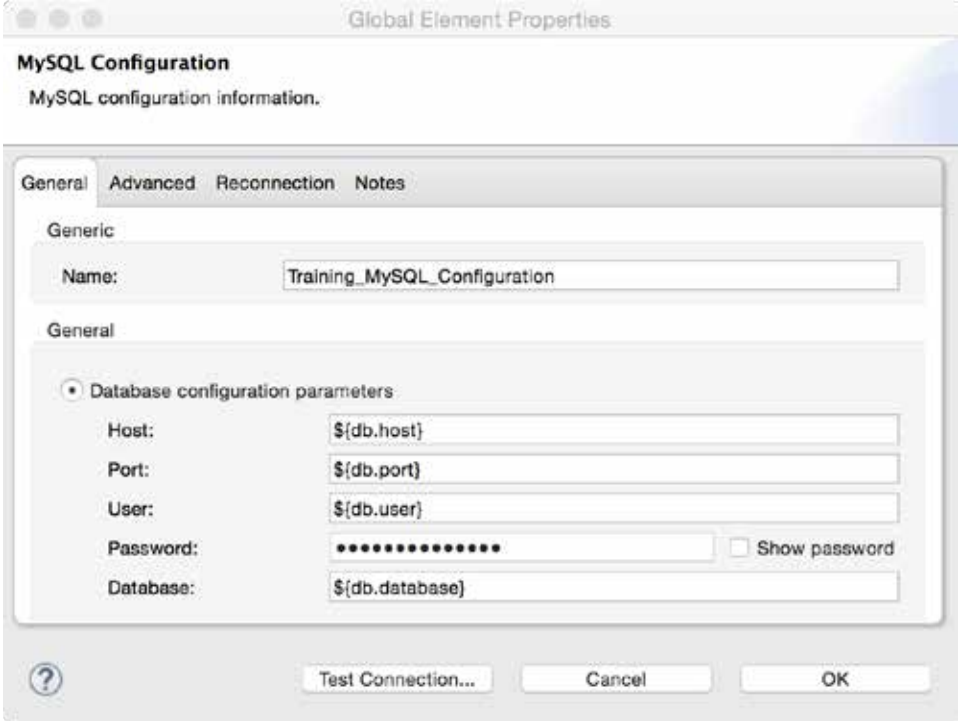


17. Copy the host value and then click OK.
18. Return to your properties file and create a property called db.host and paste the value you copied.
19. Create additional properties for the database port, user, password, and database values.



20. Save the file.

21. Return to the MySQL Configuration global element and replace the hard-coded configuration parameters with property placeholders.



22. Test the connection and make sure it still works.
23. Click OK.

Test the application

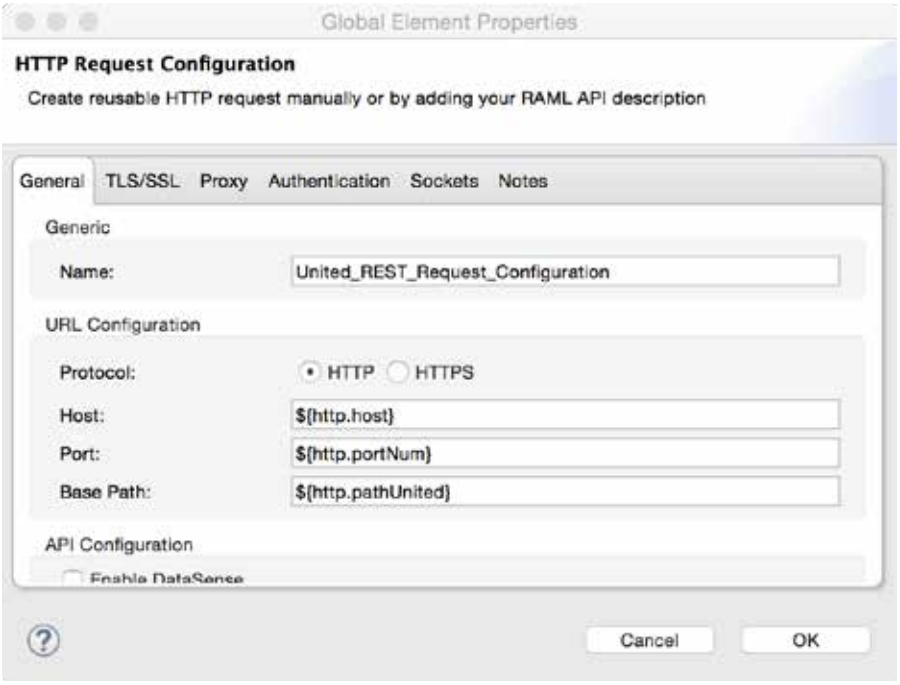
24. Save the file and run the application.
25. Make a request to http://localhost:8081/api/flights/SFO in the APIkit Consoles view or a browser and confirm it still works and returns American flights.

(Optional) Parameterize HTTP Request properties

26. Double-click the United HTTP Request Configuration global element to edit it.
27. Look at the host and port values.
28. Copy the base path value (/essentials/united/flights).

29. Change the values to properties you will define next called http.host, http.portNum, and http.pathUnited.

    Note: Do not use http.port. http.port and https.port are reserved CloudHub properties.



30. Click OK.
31. Return to apessentials-dev.properties and define these three properties.

```
http.host=mu.mulesoft-training.com
http.portnum=80
http.path_united=/essentials/united/flights
```

Test the application

32. Save all the files to redeploy the application.
33. Make a request to http://localhost:8081/api/flights/SFO in the APIkit Consoles view or a browser and confirm it still works and returns United flights.

(Optional) Parameterize Salesforce credentials

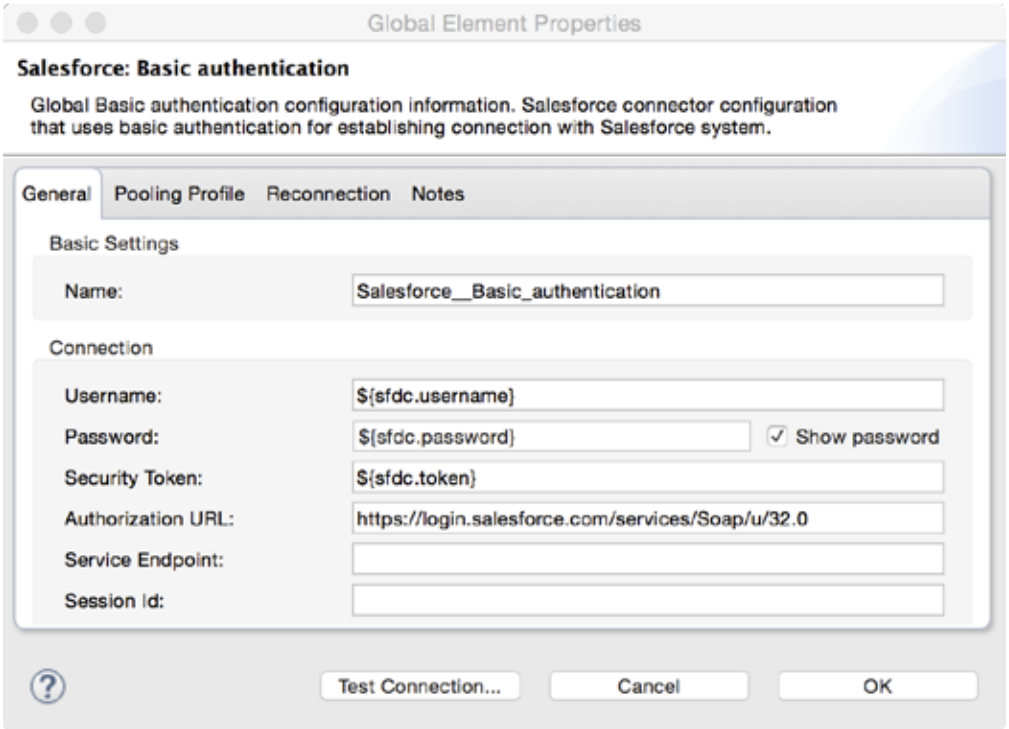
34. Return to global.xml.
35. Double-click the Salesforce global element to edit it.
36. Copy your token value and then click OK.

37. Return to the apessentials-dev.properties file and create properties for each of the Salesforce credential values.

apessentials-dev.properties | global

```
http.port=8081

db.host=mudb.mulesoft-training.com
db.port=3306
db.user=mule
db.password=mule
db.database=training

http.host=mu.mulesoft-training.com
http.portNum=80
http.pathUnited=/essentials/united/flights

sfdc.username=your@email.com
sfdc.password=your_password
sfdc.token=your_token
```

38. Return to the Salesforce global element and replace the hard-coded configuration parameters with property placeholders.



39. Test the connection and make sure it still works.
40. Click OK.

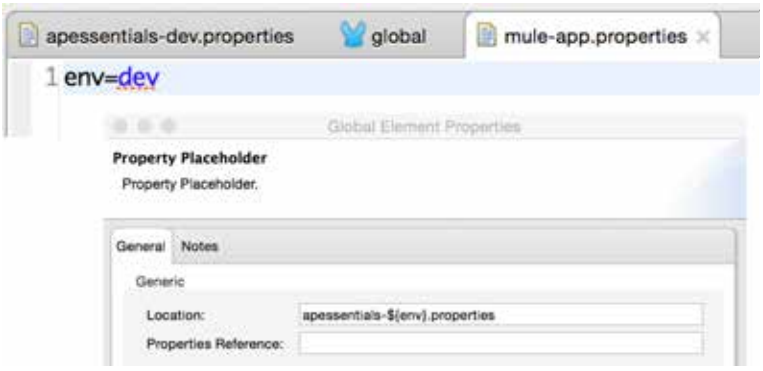
Test the application

41. Save all the files to redeploy the application.
42. Make a request http://localhost:8081/sfdc and confirm it still works and returns Salesforce data.

Walkthrough 11-2: Dynamically specify property files

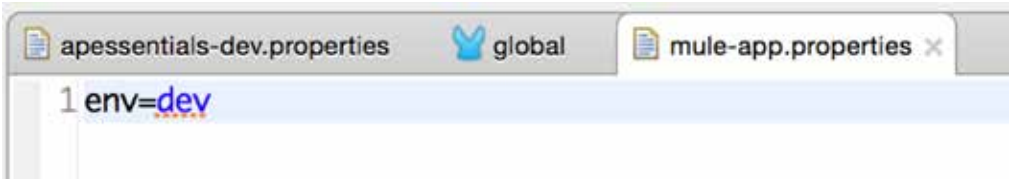
In this walkthrough, you will set the property file to use dynamically. You will:

- Define an environment property value in mule-app.properties.
- Use the environment property in the Property Placeholder.



Define an environment property value in mule-app.properties

1. Open mule-app.properties.
2. Define a property called env and set it to dev.
3. Save the file.



Use the environment property in the Property Placeholder

4. Return to global.xml.
5. Double-click the Property Placeholder to edit it.
6. In the Global Element Properties dialog box, change the location to apessentials-${env}.properties and click OK.

Test the application

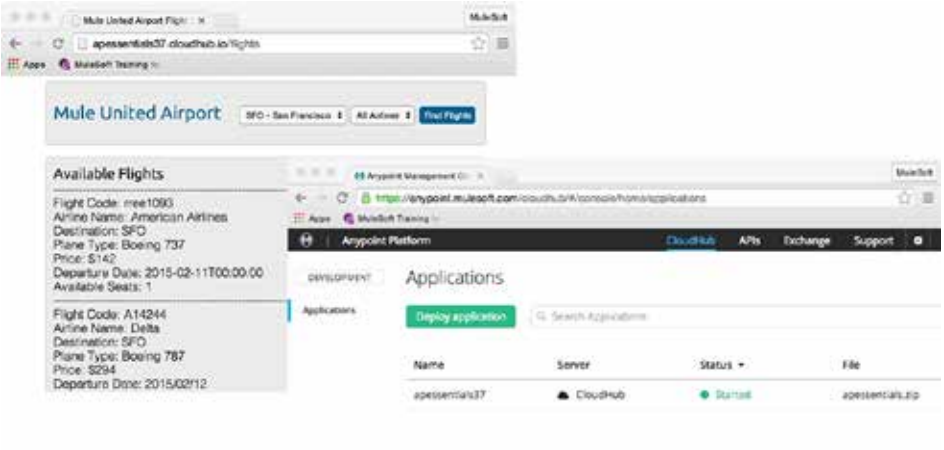
7. Save all the files to redeploy the application.
8. Make a request to http://localhost:8081/api/flights/SFO in the APIkit Consoles view or a browser and confirm it still works and returns flights.

Walkthrough 11-3: (Optional) Deploy an application to CloudHub

In this walkthrough, you will deploy your apessentials application to CloudHub. You will:

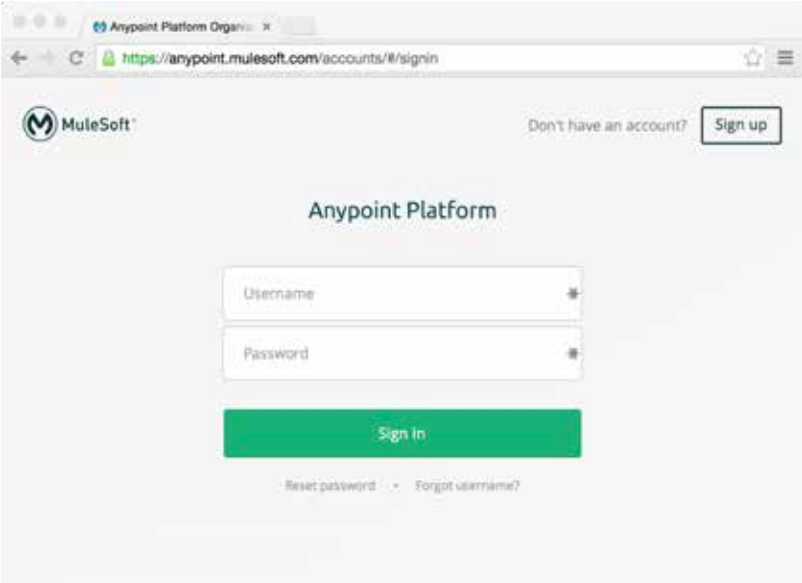
- Deploy an application to CloudHub from Anypoint Studio.
- Run the application on its new, hosted domain.
- View application data in CloudHub.

Note: To deploy to CloudHub, you need an Anypoint Platform account. If you do not already have one, get one now at http://anypoint.mulesoft.com.

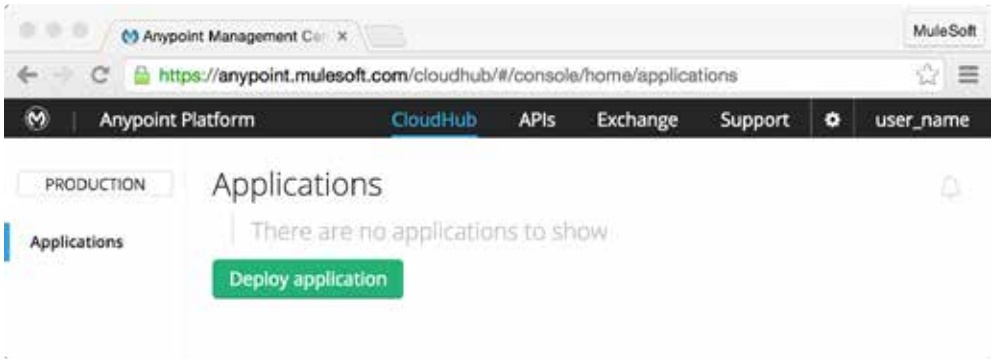


Access CloudHub

1. In a browser, go to http://anypoint.mulesoft.com and log in.



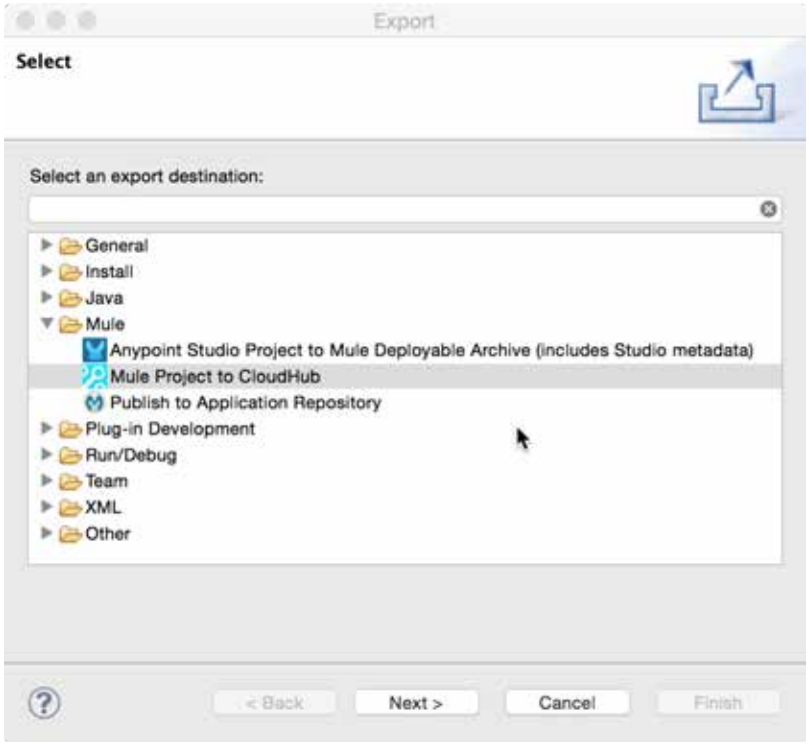
2. Click the CloudHub link in the main menu bar.
3. Look at the applications; for a new account, there will be no applications listed.



Deploy the application to CloudHub

4. Return to Anypoint Studio.
5. Select File > Export.
6. In the Export dialog box, select Mule > Mule Project to CloudHub.

   Note: You can also right-click the project in the Package Explorer and select CloudHub > Deploy to CloudHub.



7. Click Next.

8. In the Anypoint Platform Sign In dialog box, enter your Anypoint Platform credentials.

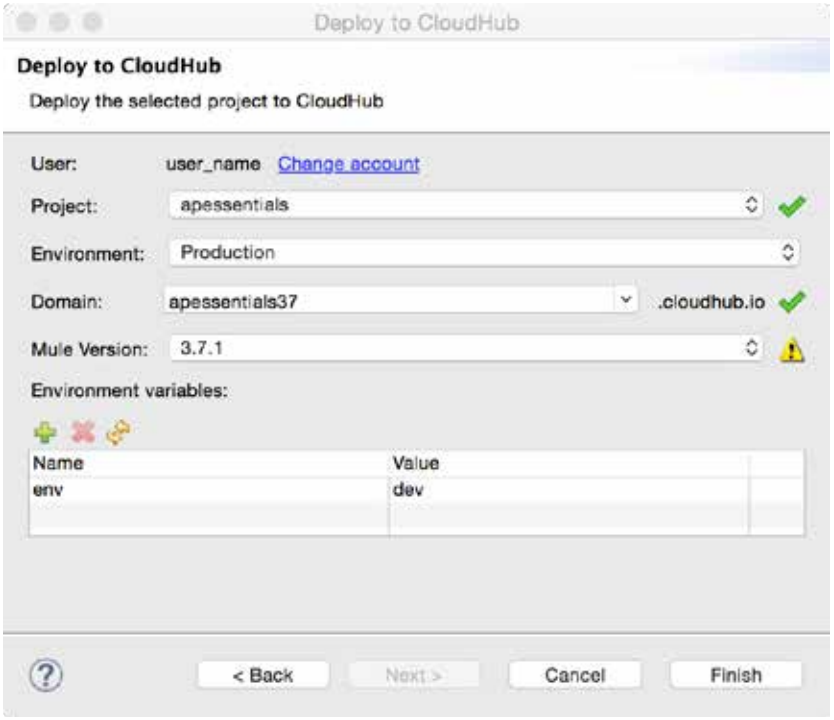
9. In the Deploy to CloudHub dialog box, select apessentials as the project to deploy.
10. Select an environment; you may or may not have more than one option.
11. Enter a domain name – the URL that will be used to access the application on CloudHub.

    Note: Because the domain name must be unique across all applications on CloudHub, you may want to use your last name or company name as part of the domain name, for example, apessentials-{your_lastname}. The availability of the domain is instantly checked and you will get a green check mark if it is available.

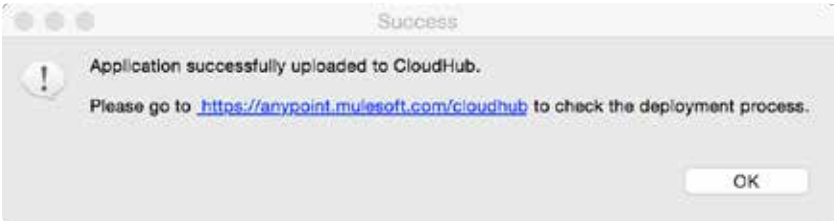
12. Set the Mule version to the version your project is using, like 3.7.1

    Note: If you don't know what version it is using, look at the Package Explorer and find a library folder with the name of the server being used, like Mule Server 3.7.1. EE.

13. Look at the environment variables; you should see the env variable set to dev.

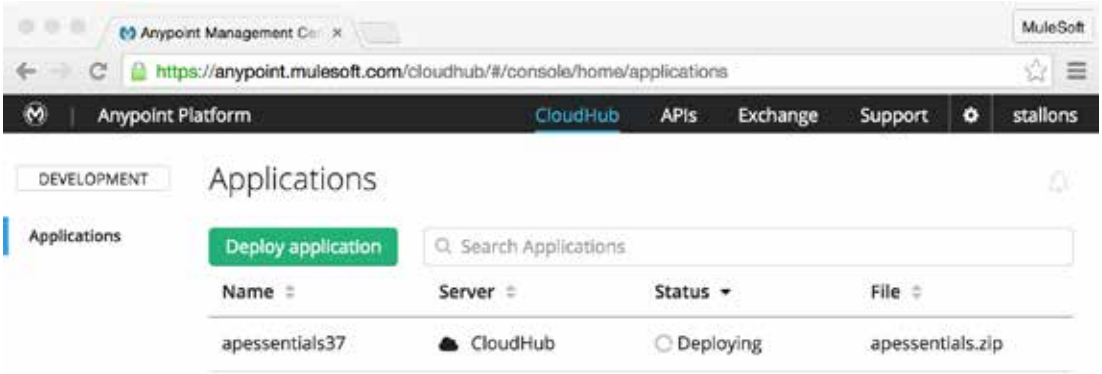


14. Click Finish.
15. Wait until you get a notice of a successful upload (or an error).

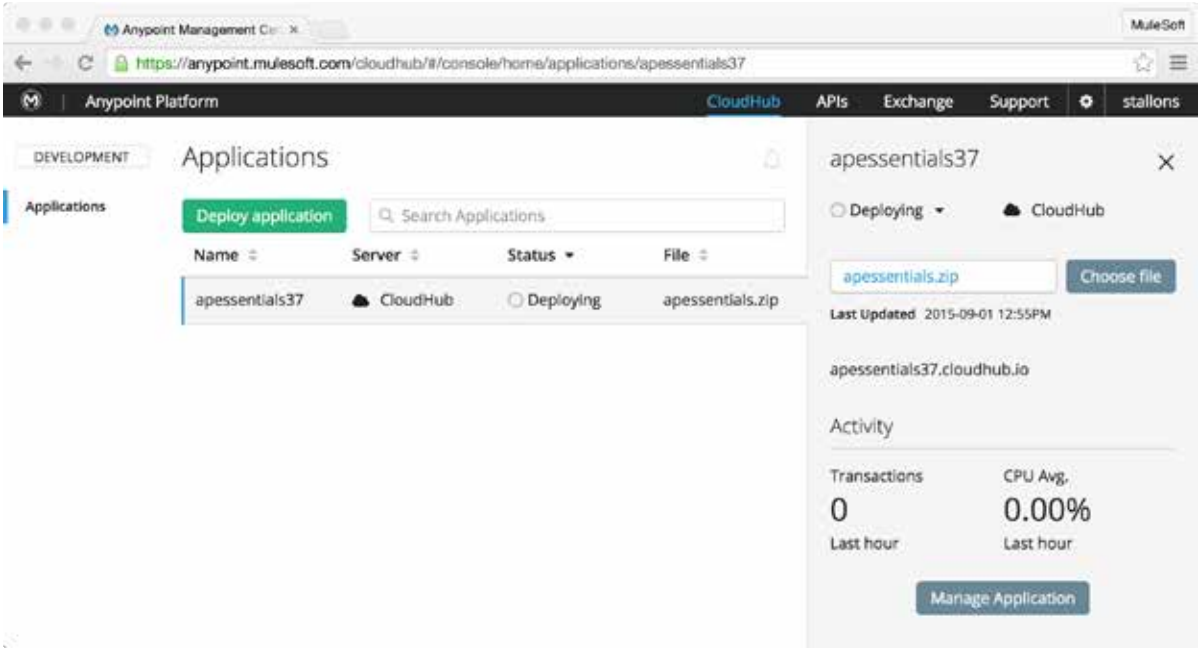


Monitor the application deployment to CloudHub

16. Return to CloudHub on Anypoint Platform and see your application listed; the status should be set to Deploying.

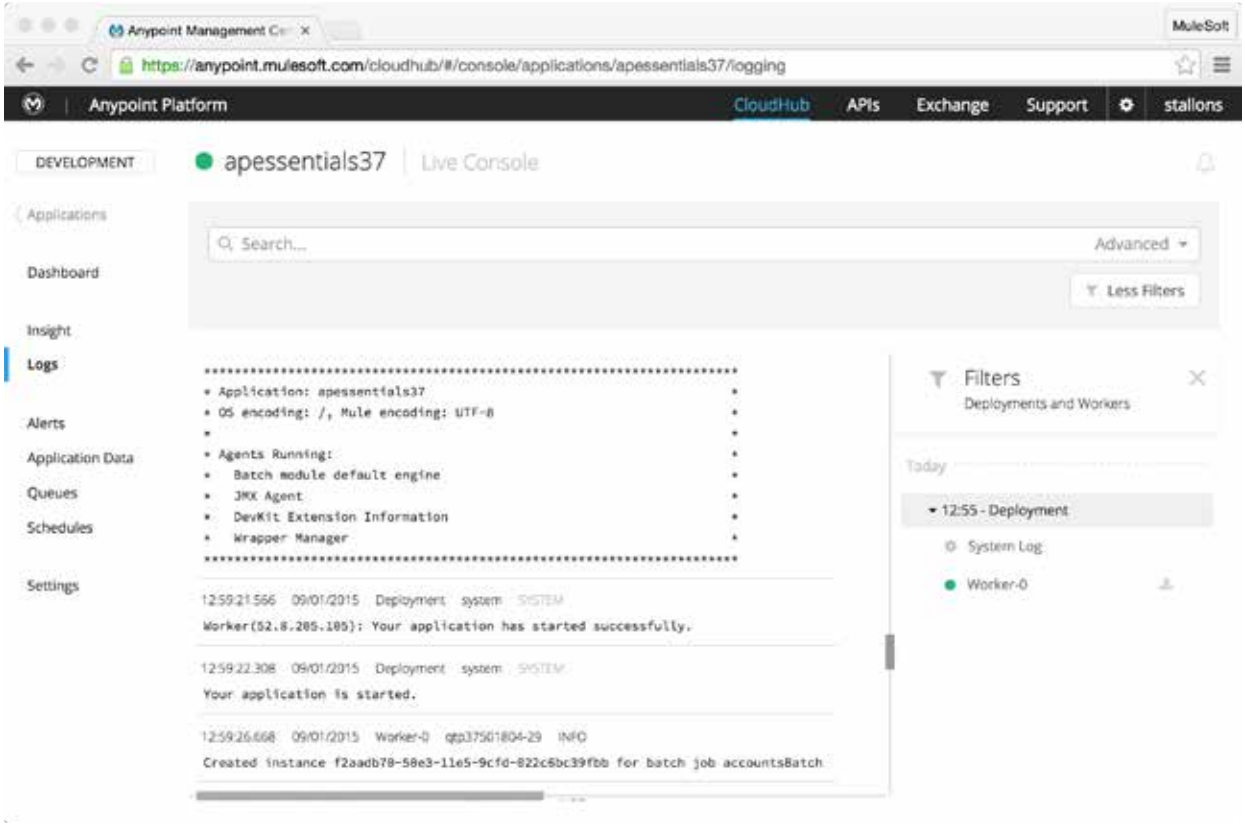


17. Click anywhere in the application row.
18. Click the Manage Application button that appears on the right side.



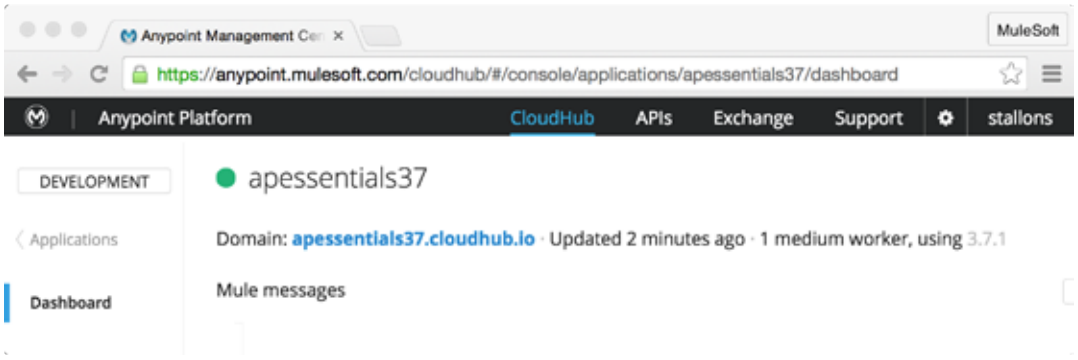
19. Click Logs in the left-side menu.
20. Watch the logs and wait until the application successfully deploys (or fails to deploy).

    Note: If your application did not successfully deploy, examine the log file to figure out why the application did not deploy. If you had errors when deploying, troubleshoot them, fix them in Anypoint Studio, and then redeploy the application to CloudHub.

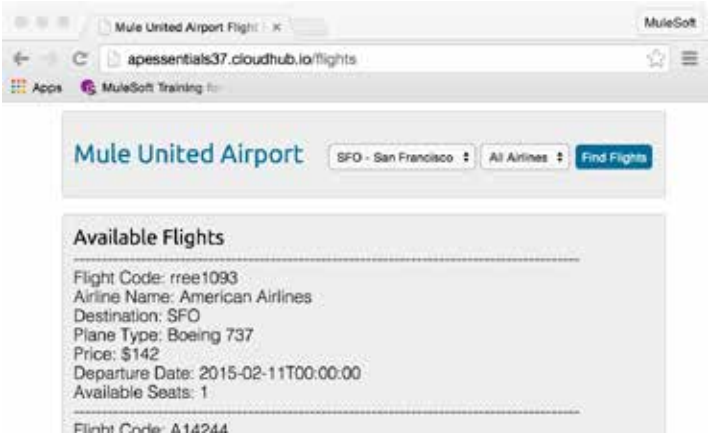


Test the application in the cloud

21. Click Dashboard in the left-side menu.
22. Click the domain link for the application, apessentials-*lastname*.cloudhub.io.



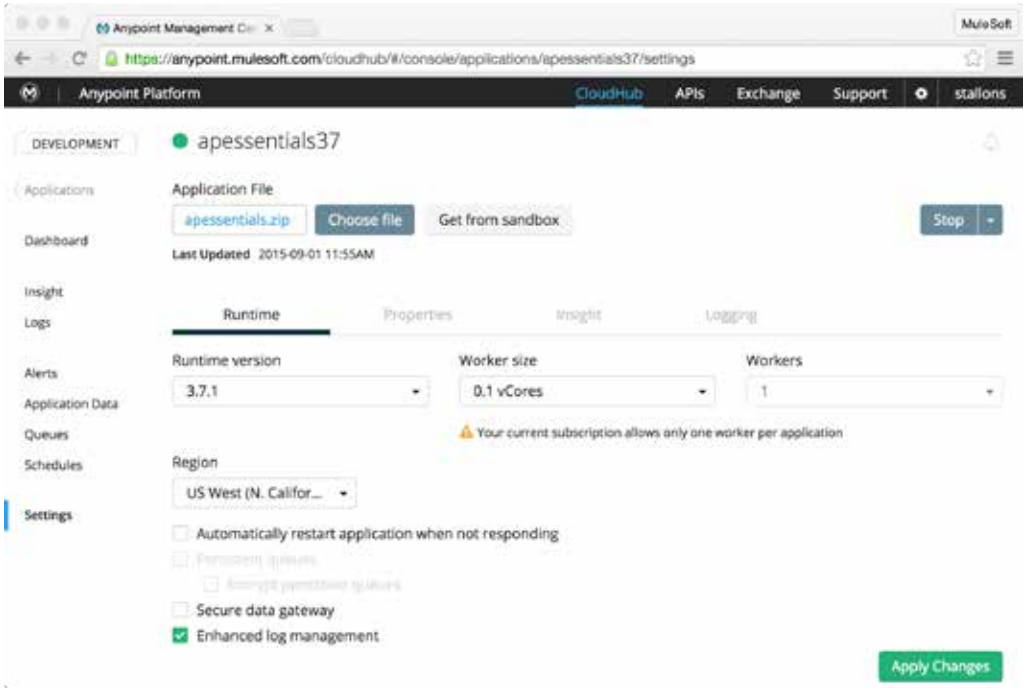
23. In the new browser tab that opens, add /flights to the URL path: http://apessentials-*lastname*.cloudhub.io/flights; your application should work as before but now it is running on CloudHub.



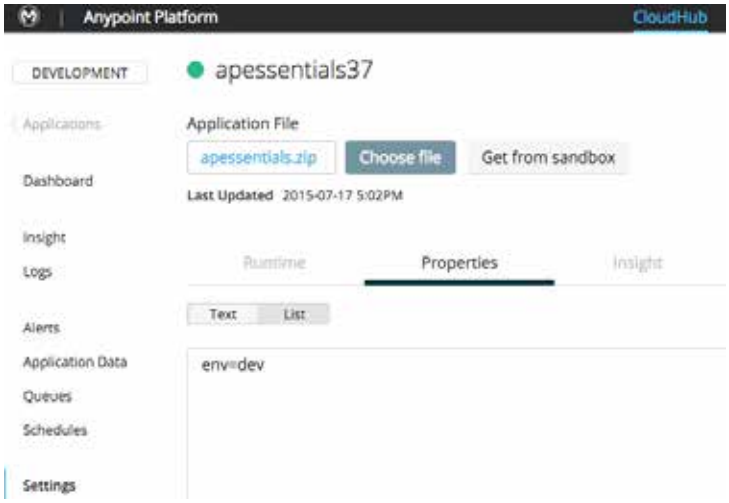
Note: You can also test your other endpoints: http://apessentials-lastname/sfdc and http://apessentials-lastname/api/flights/SFO.

Explore application data on CloudHub

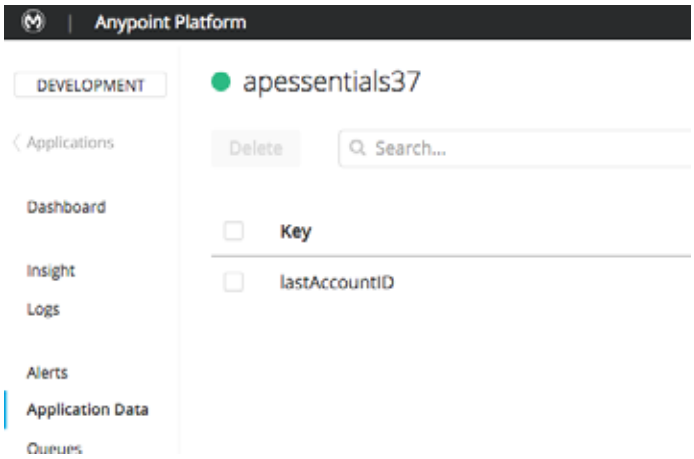
24. Return to Anypoint Platform.
25. Click Settings in the left-side menu.
26. Change the worker size to 0.1 vCores.
27. Click the Apply Changes button.



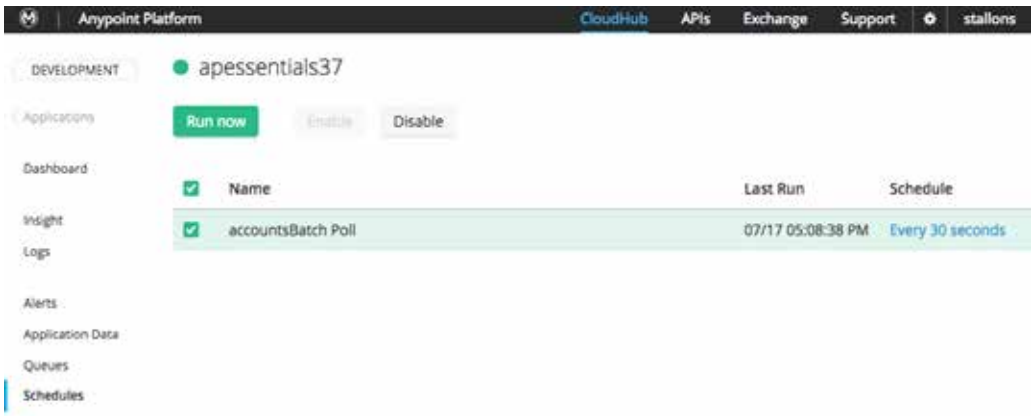
28. Click the Properties link in the middle of the screen; you should see your env environment variable and its current value.



29. Click Application Data in the left-side menu; you should see your watermark variable.



30. Click Schedules in the left-side menu; you should see the application poll and its current schedule.
31. Select the accountsBatch poll and click the Disable button.

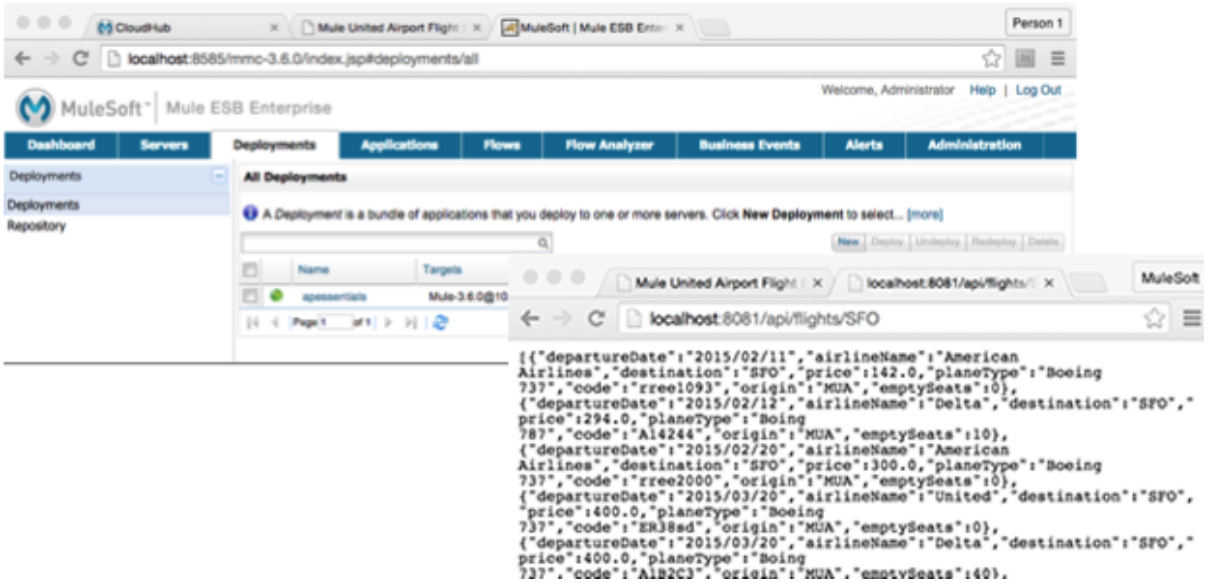


Walkthrough 11-4: (Optional) Deploy an application to Mule ESB

In this lab, you will deploy your application to a local, standalone Mule runtime. You will:

- Package an application as a Mule deployable archive.
- Start Mule ESB Enterprise and Mule Management Console (MMC).
- Deploy an application to the standalone Mule runtime.
- Run the application.

Note: To complete this walkthrough, you need a standalone Mule runtime with Mule Management Console (MMC). You can download it from http://mulesoft-training.com/mule-runtime-bundle-3.7.zip.

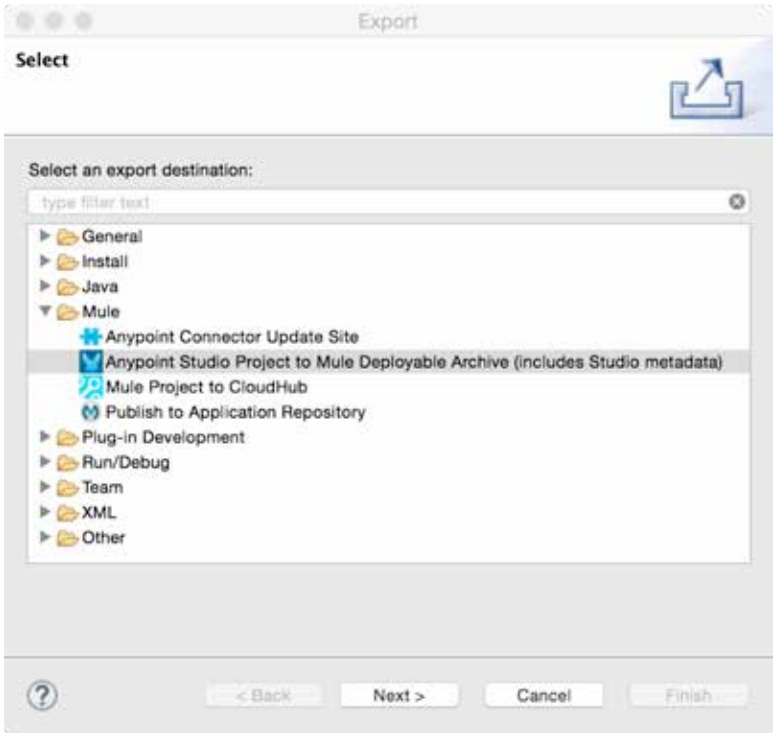


Package your application

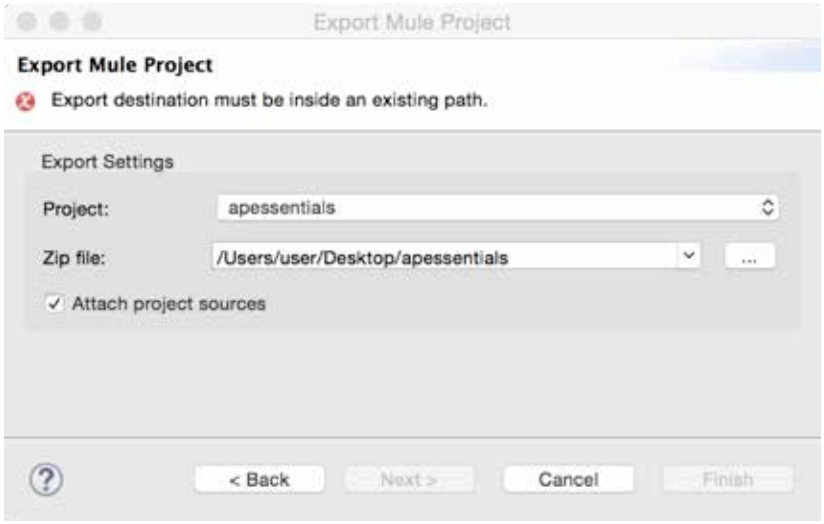
1. Return to Anypoint Studio and run the application to make sure it is working.
2. Click the red Terminate button to stop the embedded Mule runtime; YOU MUST DO THIS.
3. Select File > Export.

4. In the Export dialog box, select Mule > Anypoint Studio Project to Mule Deployable Archive and click Next.

   Note: You can also right-click a project in the Package Explorer and select Export > Anypoint Studio Project to Mule Deployable Archive.

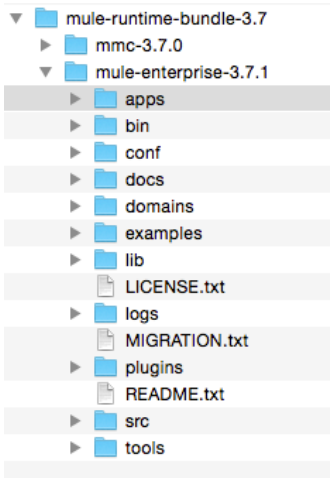


5. In the Export Mule Project dialog box, set the project to apessentials.
6. Browse to a location (like your desktop) and save the file as apessentials.



Set the environment property in the Mule wrapper.conf file

7. Locate the mmc-runtime-bundle-3.7.zip you downloaded from http://mulesoft-training.com/mule-runtime-bundle-3.7.zip.
8. Unzip the file into some directory.



9. Navigate to the mule-enterprise-3.7.X/conf folder.
10. Open wrapper.conf in a text editor.
11. Locate the last numbered wrapper.java.additional instruction near the top of the file.
12. Beneath it, create a new instruction with the number incremented by one that passes an argument called env equal to dev to the runtime when it starts.

```
wrapper.java.additional.15=-Denv=dev
```

Note: Be sure to use the correct number. This number may be different than shown here for your version of the runtime.

13. Save the file.

Start the Mule runtime

14. On your computer, open a Terminal or Command window.
15. Change to the /mule-runtime-bundle-3.7/mule-enterprise-3.7.X/bin directory.
16. Run the mule script: mule (Unix) or mule.bat (Windows).

```
bin — java — 97×8
Last login: Tue Sep  1 14:09:06 on ttys000
ML-JSTAL-OSX-SF:~ jeanette.stallons$ cd /Users/jeanette.stallons/dev/mule/mule-runtime-bundle-3.7
/mule-enterprise-3.7.1/bin
ML-JSTAL-OSX-SF:bin jeanette.stallons$ ./mule
MULE_HOME is set to /Users/jeanette.stallons/dev/mule/mule-runtime-bundle-3.7/mule-enterprise-3.7
.1
Running in console (foreground) mode by default, use Ctrl-C to exit...
MULE_HOME is set to /Users/jeanette.stallons/dev/mule/mule-runtime-bundle-3.7/mule-enterprise-3.7
```

17. Wait until you see a message that Mule is up and kicking.

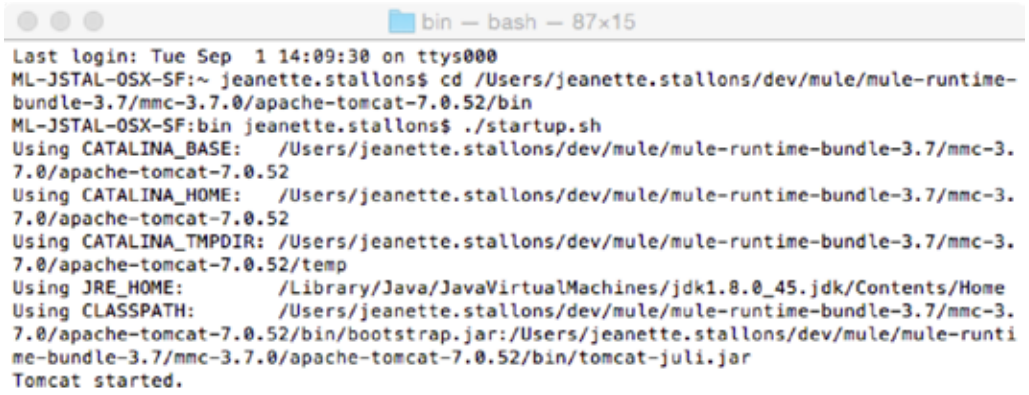
    Note: You may need to scroll up to see this if you get a lot of DNS warnings after it.

```
bin — java — 103×29
INFO  2015-09-01 14:09:57,838 [WrapperListener_start_runner] org.mule.module.launcher.MuleDeploymentSer
vice:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Started app 'default'                                        +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
INFO  2015-09-01 14:09:57,878 [WrapperListener_start_runner] org.mule.module.launcher.DeploymentDirecto
ryWatcher:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Mule is up and kicking (every 5000ms)                         +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
WARN  2015-09-01 14:09:57,882 [SocketListener(172-16-12-96.local.)] javax.jmdns.impl.DNSIncoming: There
 was an OPT answer. Not currently handled. Option code: 65002 data: E6F58FAFFB25CC54
WARN  2015-09-01 14:09:57,883 [SocketListener(172-16-12-96.local.)] javax.jmdns.impl.DNSIncoming: There
 was an OPT answer. Not currently handled. Option code: 65002 data: 7B6F0BA6FBC7011D
INFO  2015-09-01 14:09:57,885 [WrapperListener_start_runner] org.mule.module.launcher.StartupSummaryDep
loymentListener:
************************************************************************
*              - - + DOMAIN + - -               * - - + STATUS + - - *
************************************************************************
* default                                       * DEPLOYED           *
************************************************************************

********************************************************************************************************************
*            - - + APPLICATION + - -            *       - - + DOMAIN + - -       * - - + STATUS + - - *
********************************************************************************************************************
* default                                       * default                        * DEPLOYED           *
********************************************************************************************************************

WARN  2015-09-01 14:09:57,986 [SocketListener(172-16-12-96.local.)] javax.jmdns.impl.DNSIncoming: There
```

Start MMC

18. On your computer, open a second Terminal or Command window.

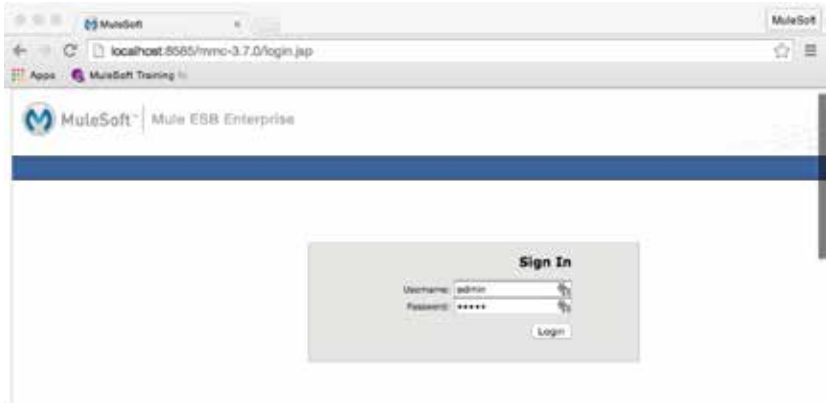
19. Change to the /mmc-runtime-bundle-3.7/mmc-3.7.X/apache-tomcat-7.0.X/bin directory.
20. Run the startup script: ./startup.sh (Unix) or startup.bat (Windows).
21. Wait until you see a message that Tomcat started in your console.



22. In a browser window, navigate to http://localhost:8585/mmc-3.7.0.

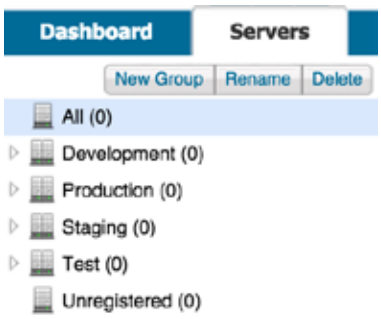
    Note: Change the URL as necessary for a different version of the runtime.

23. Enter a username and password of admin and click Login.

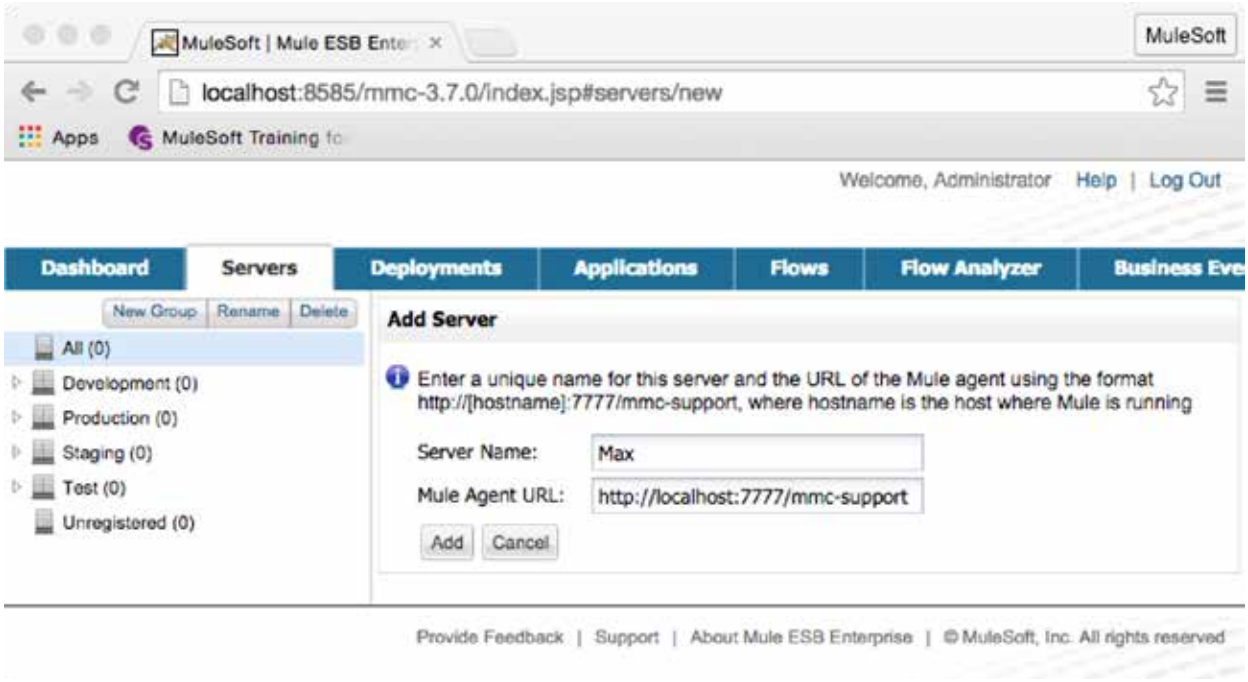


Register the Mule runtime in MMC

24. In MMC, click the Servers tab.
25. Look at the groups on the left; they should all show 0 registered servers.



26. Click the Add drop-down menu and select New Server.
27. Enter a server name of Max (or some other value).
28. Leave the Mule Agent URL as http://localhost:7777/mmc-support
29. Click Add.

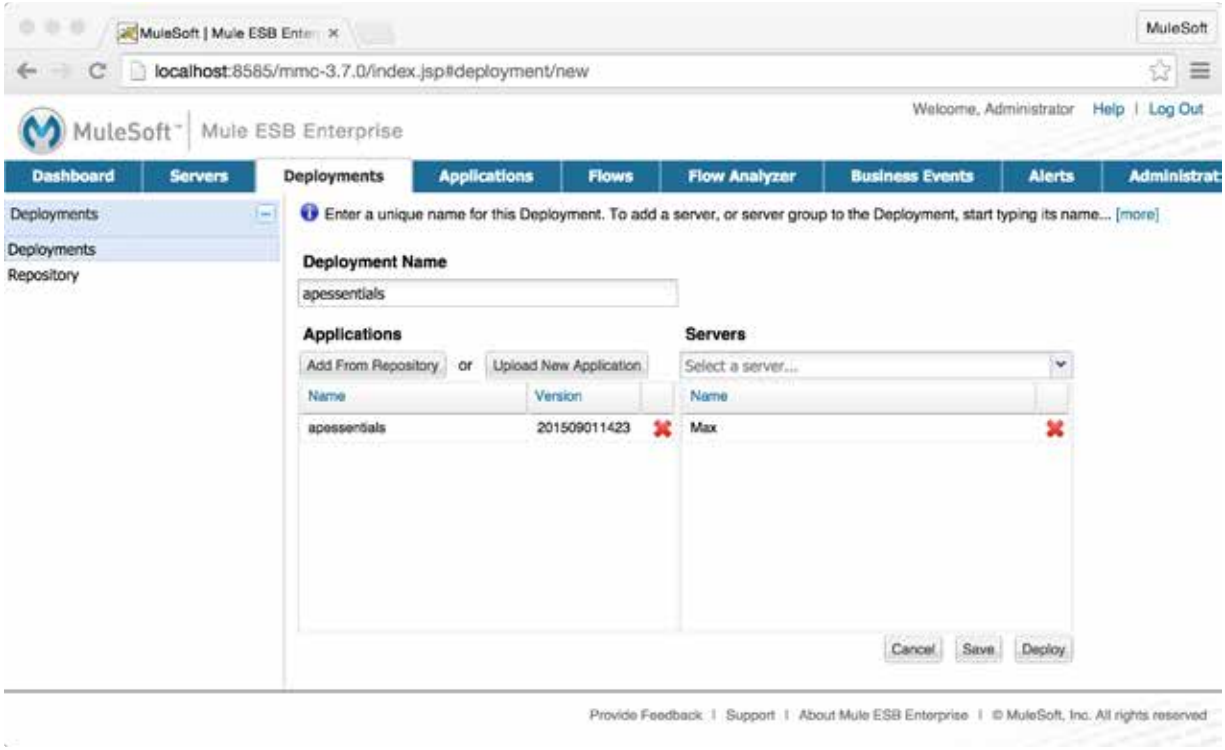


30. Wait for the server to register.

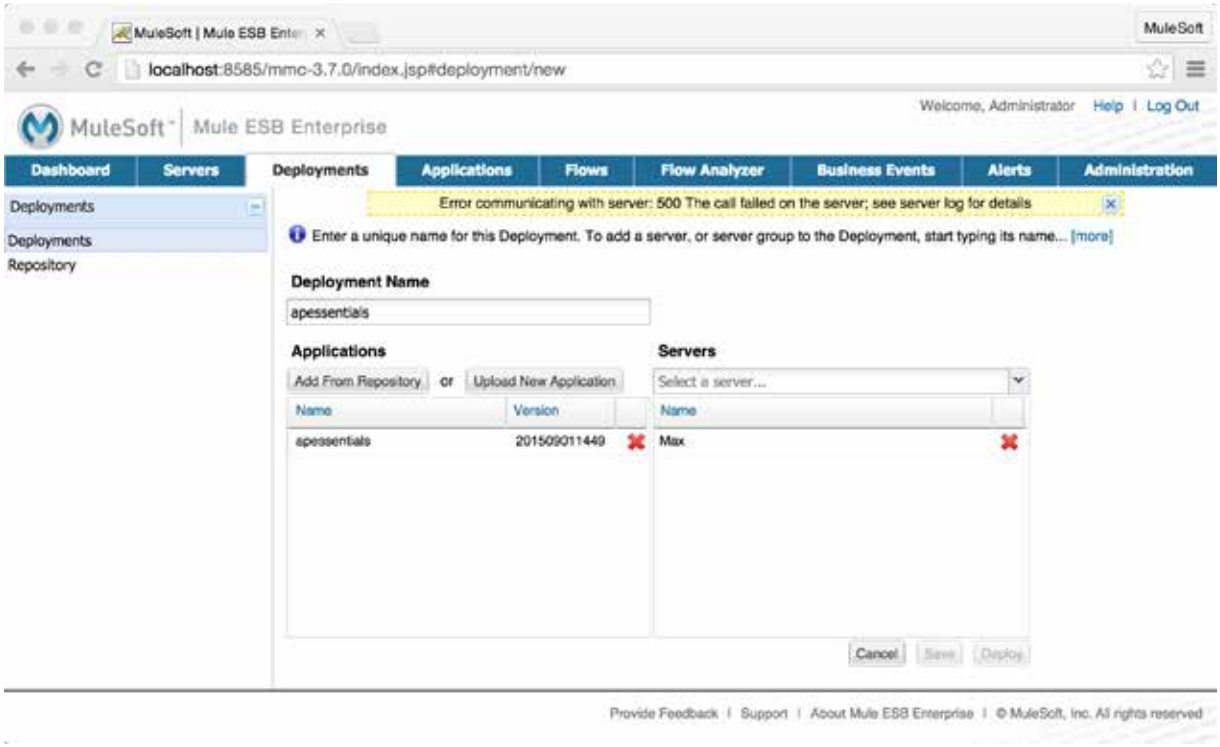
Try to deploy an application to the Mule runtime

31. Click the Deployments tab.
32. Click the New button.
33. Enter a deployment name of apessentials.
34. Select the server you just registered, Max.
35. Click the Upload New Application button.
36. Click the Browse button, browse to the zip file you created earlier, and click Open.

37. Click the Add button; you should see your archive listed.

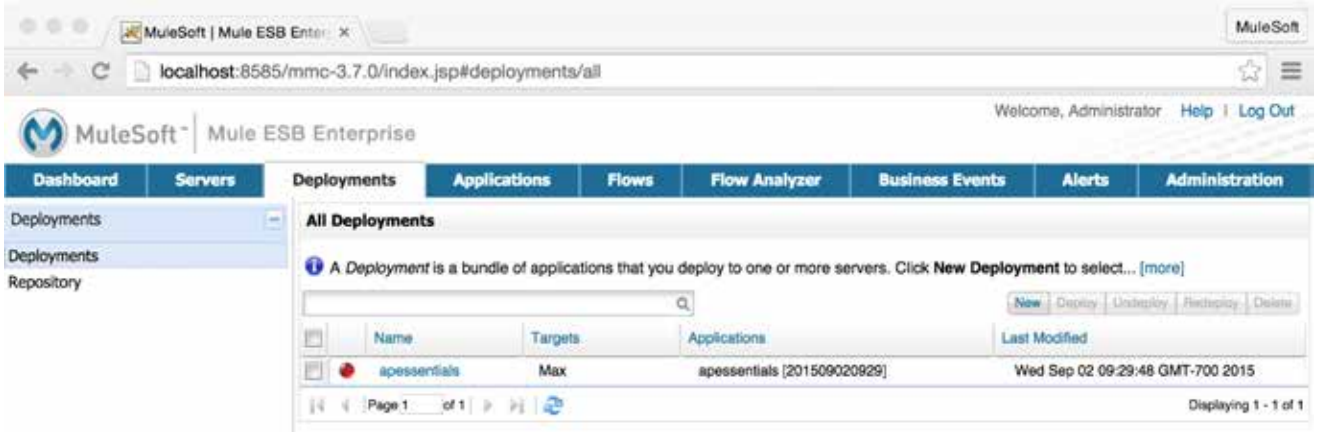


38. Click the Deploy button; you should get an error.



39. Click the Applications tab; you should see the apessentials application listed with a gray circle next to it indicating that the application is not running.

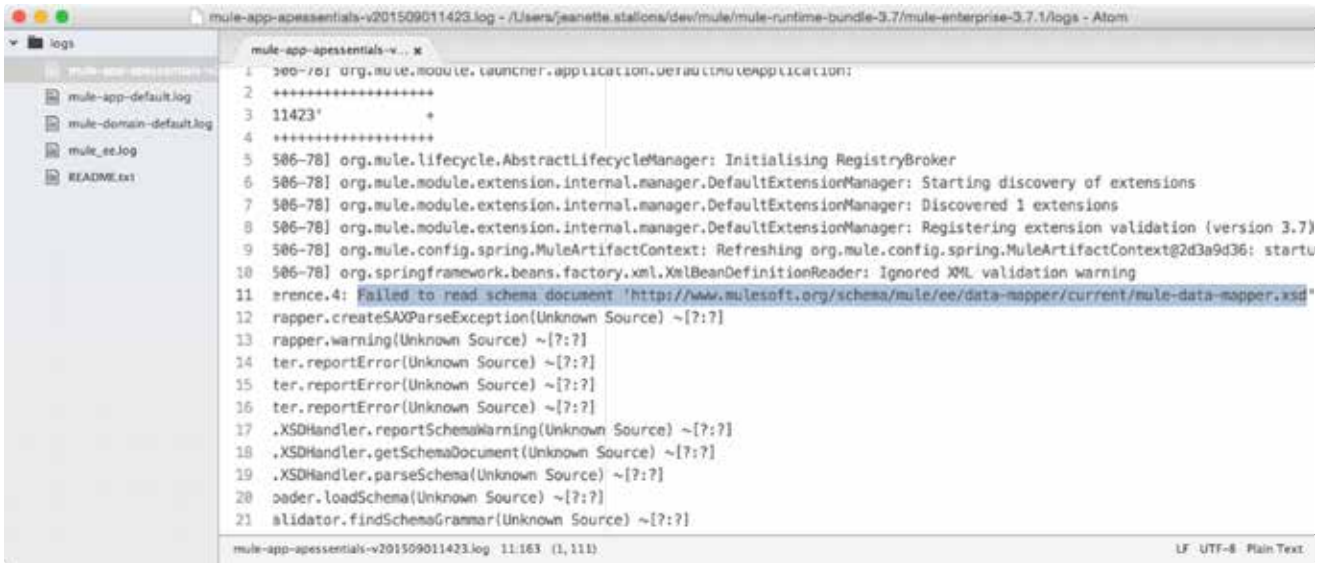
40. Click the Deployments tab; you should see the apessentials deployment listed with a red circle next to it indicating that the deployment failed.



Review the log file

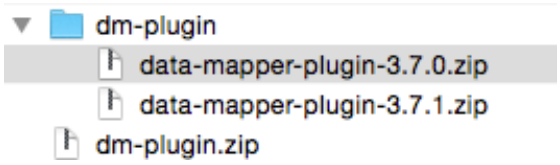
41. In your computer's system explorer, navigate to the /mmc-runtime-bundle-3.7/mmc-enterprise-3.7.1/logs directory.
42. Open mule-app-apessentials-{*version*}.log; you should see a number of errors related to DataMapper.

    Note: Starting with Mule runtime 3.7.0, DataMapper is an optional plugin that must be installed inside the Mule runtime for applications that are using it.

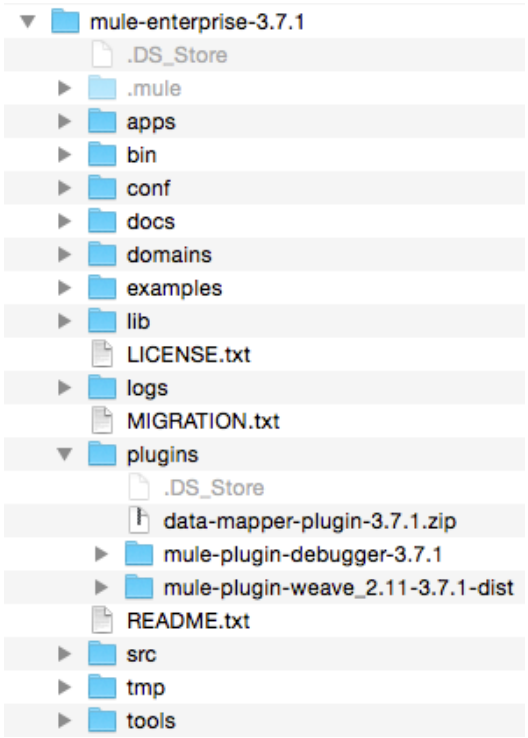


Install the DataMapper plugin

43. If you have a Anypoint Platform account for your company, navigate to the Customer Portal and locate and download the DataMapper plugin.
44. If you are using a trial Anypoint Platform account and/or do not have access to the Customer Support portal, download the plugin from http://mulesoft-training.com/dm-plugin.zip.
45. Expand the dm-plugin.zip.

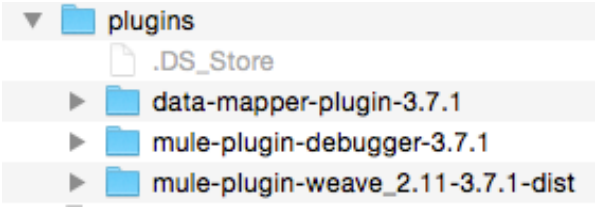


46. Move the data-mapper-plugin-3.7.1.zip to the /mmc-runtime-bundle-3.7/mmc-enterprise-3.7.1/plugins directory.



47. Return to the Terminal/Command window for the Mule runtime.
48. Press Ctrl+C to stop the runtime.
49. Wait until the runtime stops.
50. Run the mule script: mule (Unix) or mule.bat (Windows) to start the runtime.

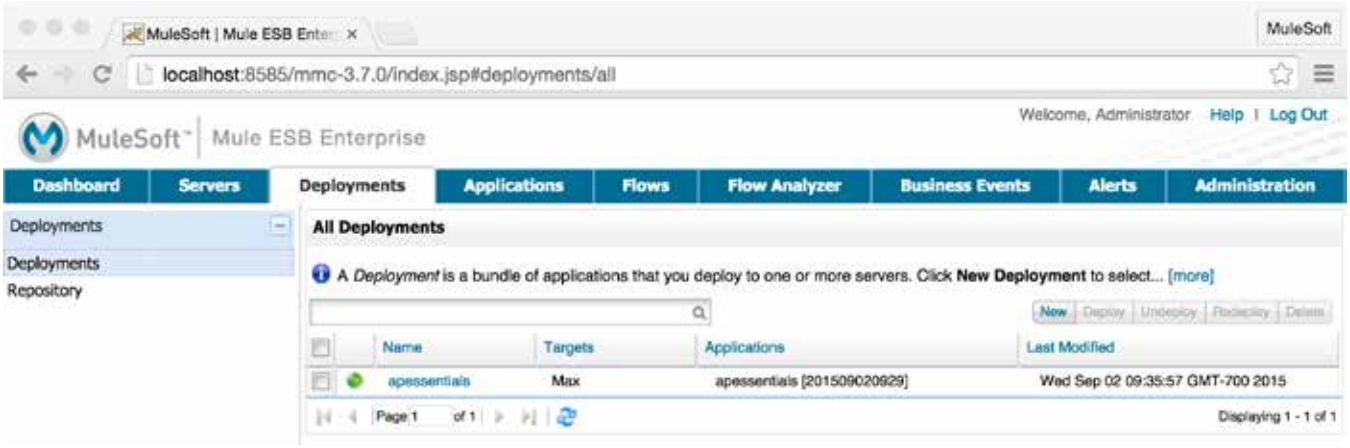
51. Look at the plugins folder; you should see the DataMapper plugin expanded.



52. Watch the Terminal/Command window; you should see the Wait for the runtime to start.

Deploy the application to the Mule runtime

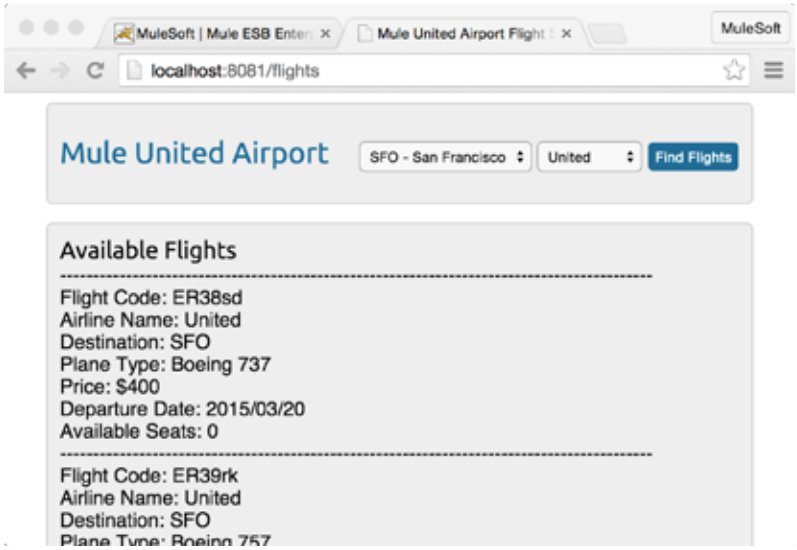
53. Return to MMC.
54. Click the Applications tab; there should now be a green circle next to the apessentials application.
55. Click the Deployments tab; there should now be a green circle next to apessentials deployment.
56. Click the Deploy button in the lower-right corner; you should see your application appear in the deployment list with a green circle next to it.



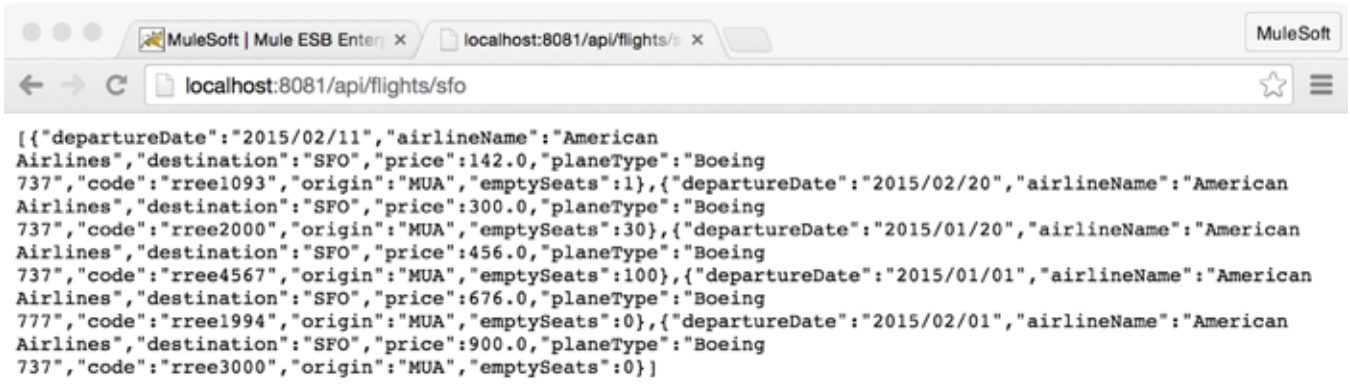
Note: If there is still a red circle, examine the log files again to figure out why it still failed. You may need to return to Anypoint Studio to fix application errors and then create a new deployable archive to redeploy.

Test the application

57. Make a request to http://localhost:8081/flights and find flights; your application should work as before but now it is running on your local standalone Mule runtime.



58. Make a request to http://localhost:8081/api/flights/SFO; you should get flight results.



Stop MMC and the Mule runtime

59. Return to the MMC Terminal/Command window.
60. Run the shutdown script: ./shutdown.sh (Unix) or shutdown.bat (Windows).
61. Close the Terminal/Command window.
62. Return to the Mule Terminal/Command window.
63. Stop the server by pressing Ctrl+C.
64. Wait until the runtime stops.
65. Close the Terminal/Command window.