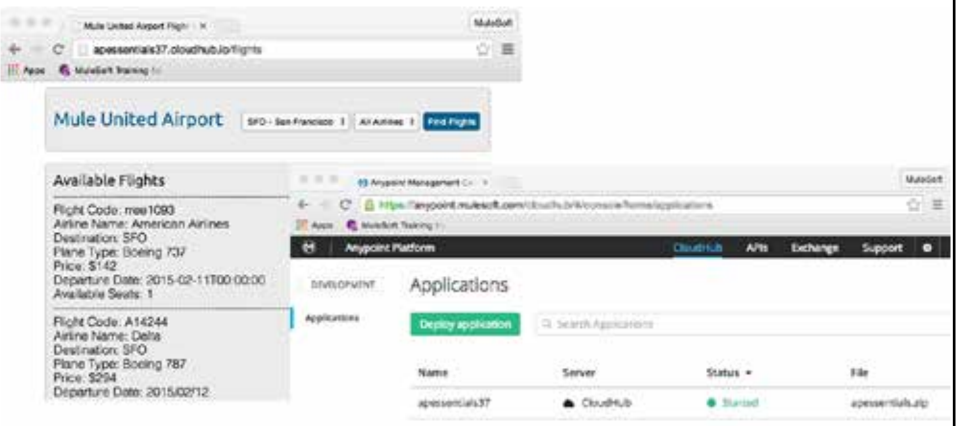
Module 11: Deploying Mule Applications

Goal



apessentials37.cloudhub.io/flights

Mule United Airport

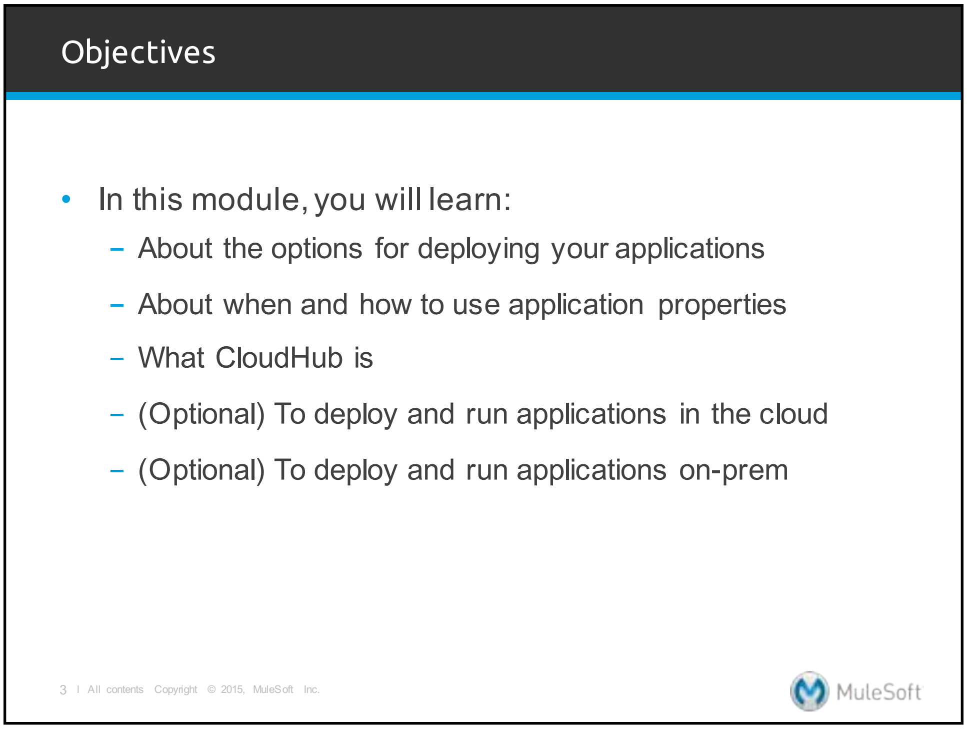
Available Flights

Flight Code: ree1093
Airline Name: American Airlines
Destination: SFO
Plane Type: Boeing 737
Price: $142
Departure Date: 2015-02-11T00:00:00
Available Seats: 1

Flight Code: A14244
Airline Name: Delta
Destination: SFO
Plane Type: Boeing 787
Price: $294
Departure Date: 2015/02/12

Applications

Name	Server	Status	File
apessentials37	CloudHub	Started	apessentials.zip



Objectives

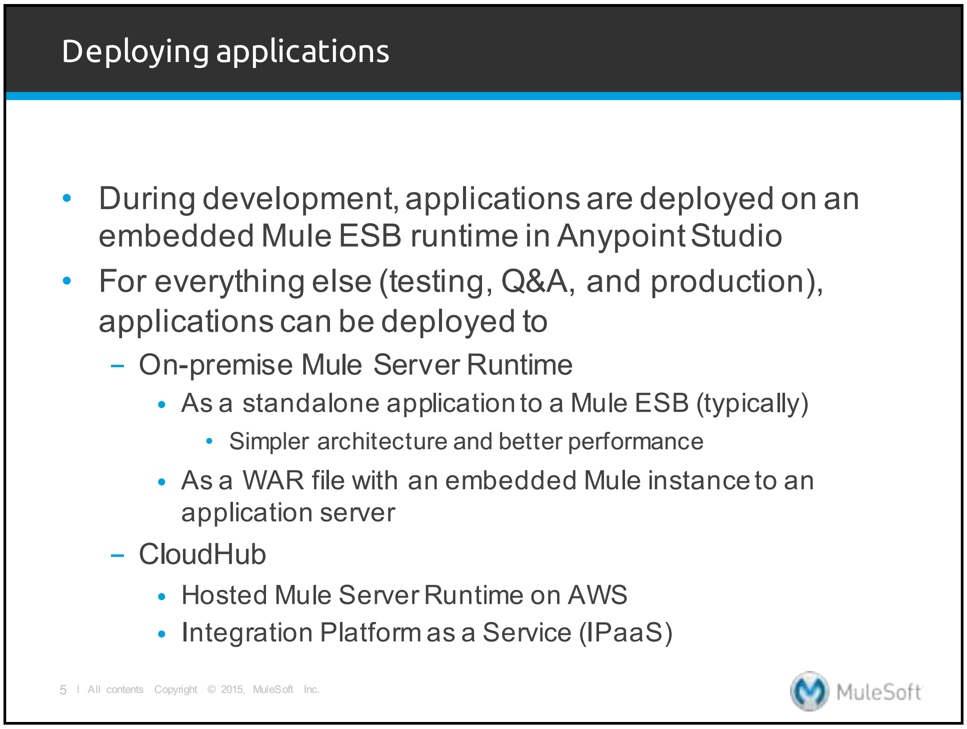
- In this module, you will learn:
  - About the options for deploying your applications
  - About when and how to use application properties
  - What CloudHub is
  - (Optional) To deploy and run applications in the cloud
  - (Optional) To deploy and run applications on-prem

3 I All contents Copyright © 2015, MuleSoft Inc.

MuleSoft

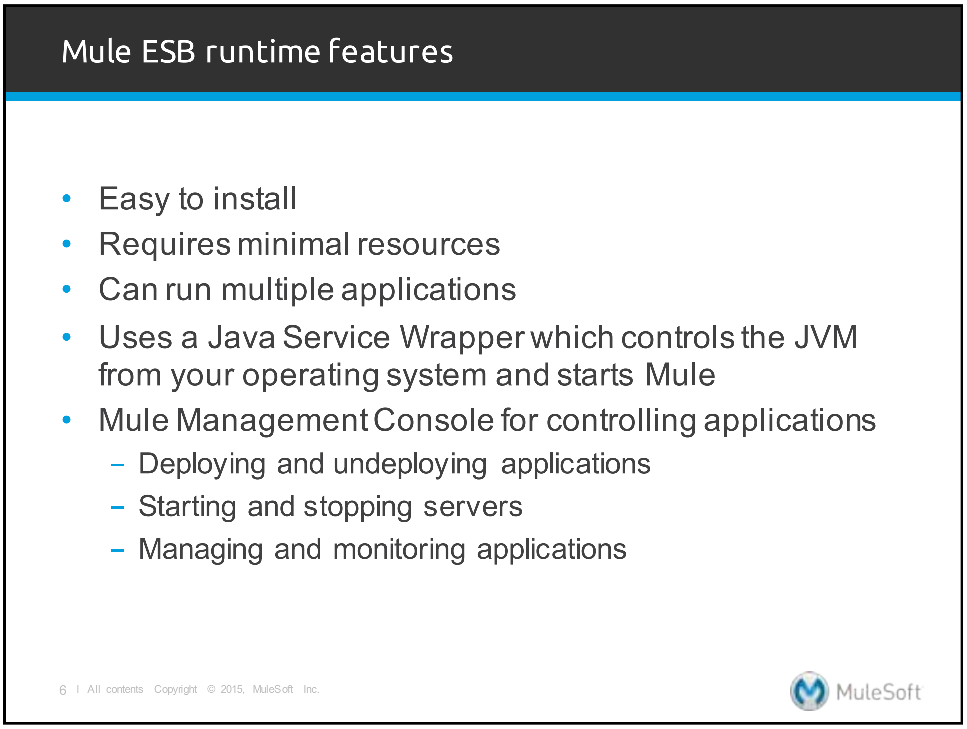
Introducing deployment options

4 I All contents Copyright © 2015, MuleSoft Inc.



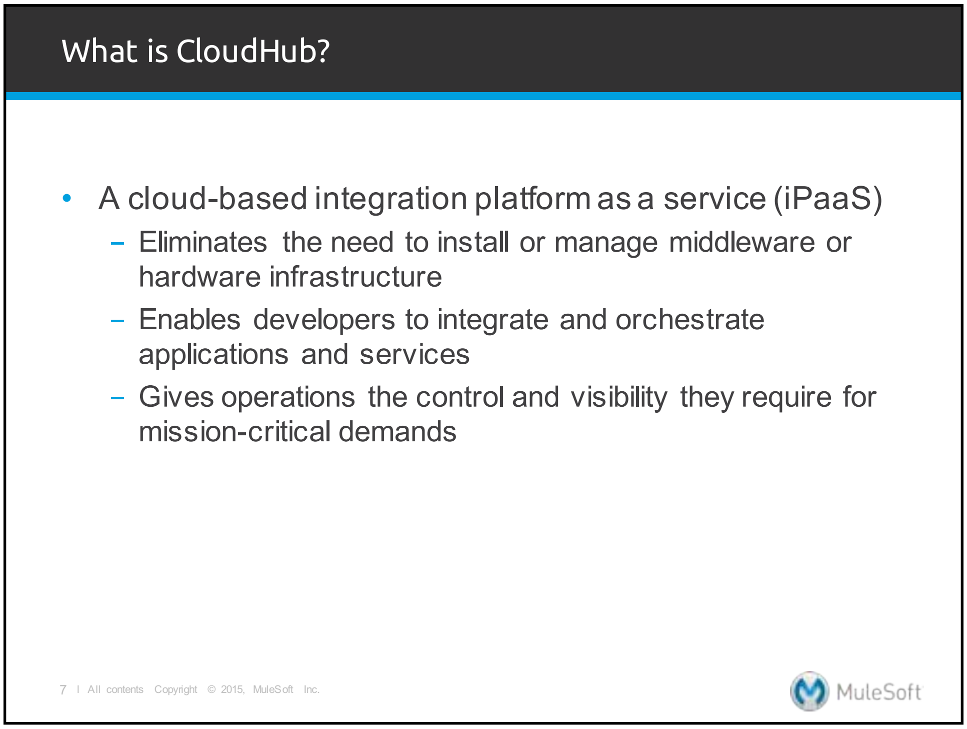
Deploying applications

- During development, applications are deployed on an embedded Mule ESB runtime in Anypoint Studio
- For everything else (testing, Q&A, and production), applications can be deployed to
  - On-premise Mule Server Runtime
    - As a standalone application to a Mule ESB (typically)
      - Simpler architecture and better performance
    - As a WAR file with an embedded Mule instance to an application server
  - CloudHub
    - Hosted Mule Server Runtime on AWS
    - Integration Platform as a Service (IPaaS)



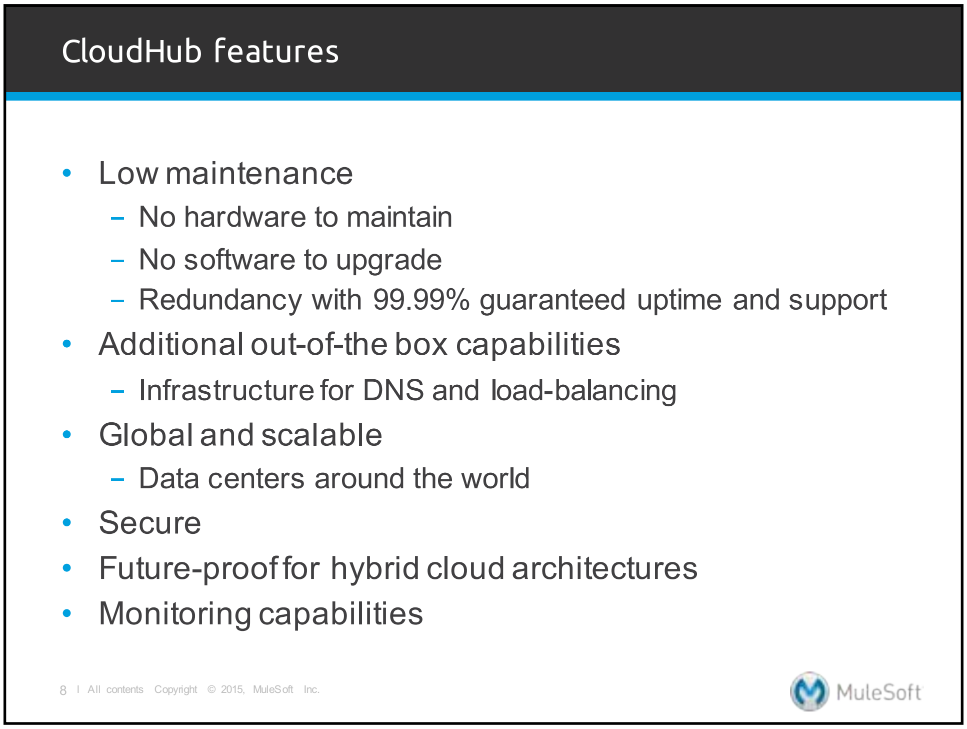
Mule ESB runtime features

- Easy to install
- Requires minimal resources
- Can run multiple applications
- Uses a Java Service Wrapper which controls the JVM from your operating system and starts Mule
- Mule Management Console for controlling applications
  - Deploying and undeploying applications
  - Starting and stopping servers
  - Managing and monitoring applications



What is CloudHub?

- A cloud-based integration platform as a service (iPaaS)
  - Eliminates the need to install or manage middleware or hardware infrastructure
  - Enables developers to integrate and orchestrate applications and services
  - Gives operations the control and visibility they require for mission-critical demands

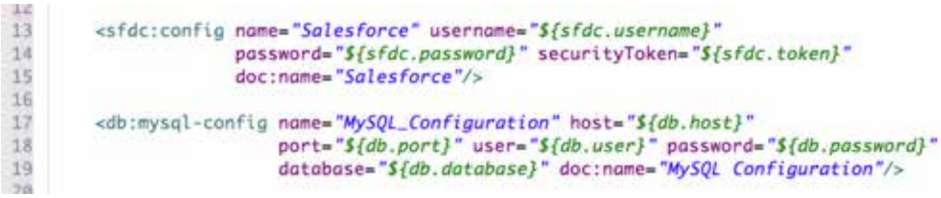


CloudHub features

- Low maintenance
  - No hardware to maintain
  - No software to upgrade
  - Redundancy with 99.99% guaranteed uptime and support
- Additional out-of-the box capabilities
  - Infrastructure for DNS and load-balancing
- Global and scalable
  - Data centers around the world
- Secure
- Future-proof for hybrid cloud architectures
- Monitoring capabilities

Before deploying

- Think about anything in your application that might change between development and production…

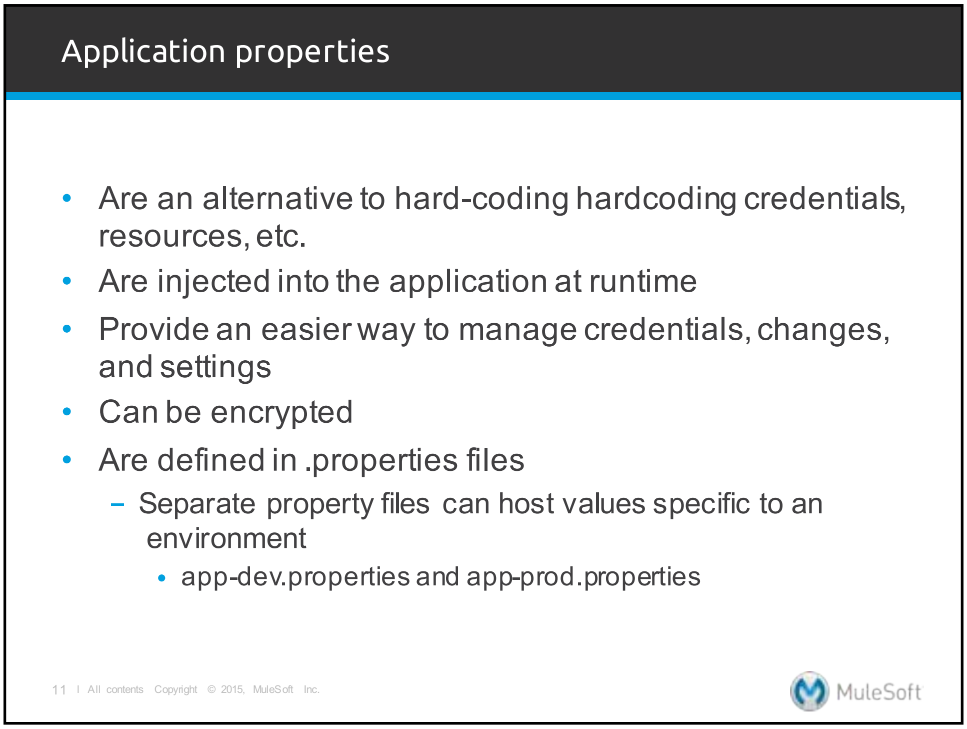


```
<sfdc:config name="Salesforce" username="${sfdc.username}"
             password="${sfdc.password}" securityToken="${sfdc.token}"
             doc:name="Salesforce"/>

<db:mysql-config name="MySQL_Configuration" host="${db.host}"
                 port="${db.port}" user="${db.user}" password="${db.password}"
                 database="${db.database}" doc:name="MySQL Configuration"/>
```

MuleSoft

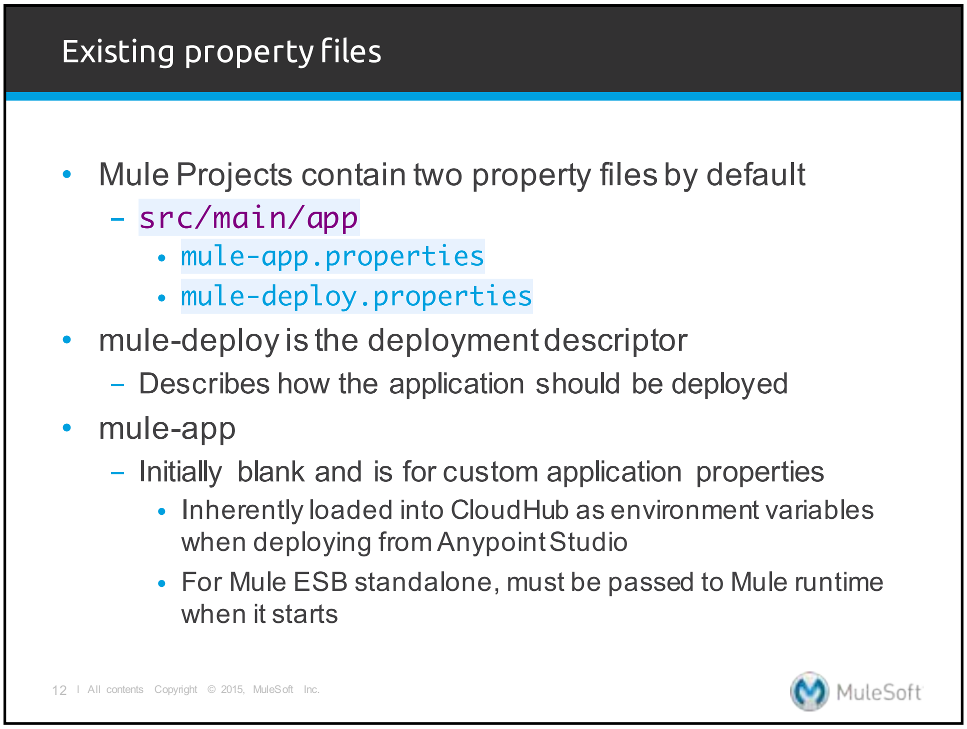
Using application properties



Application properties

- Are an alternative to hard-coding hardcoding credentials, resources, etc.
- Are injected into the application at runtime
- Provide an easier way to manage credentials, changes, and settings
- Can be encrypted
- Are defined in .properties files
  - Separate property files can host values specific to an environment
    - app-dev.properties and app-prod.properties

MuleSoft



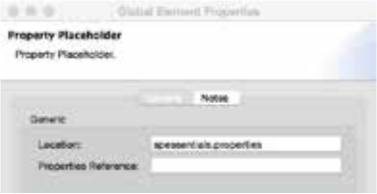
Existing property files

- Mule Projects contain two property files by default
  - `src/main/app`
    - `mule-app.properties`
    - `mule-deploy.properties`
- mule-deploy is the deployment descriptor
  - Describes how the application should be deployed
- mule-app
  - Initially blank and is for custom application properties
    - Inherently loaded into CloudHub as environment variables when deploying from Anypoint Studio
    - For Mule ESB standalone, must be passed to Mule runtime when it starts

MuleSoft

Defining application properties

- Create a custom properties file anywhere in the project
  `apessentials.properties`
- Define properties in the properties file
  `db.account = ReaderAccount`
- Create a Properties Placeholder global element



Global Element Properties

Property Placeholder

Property Placeholder.

Generic

Location: apessentials.properties

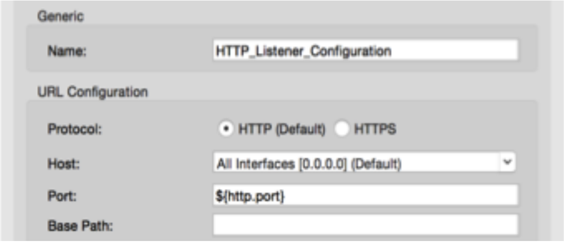
Properties Reference:

- Use the properties in the application
  `${db.account}`

MuleSoft

Parameterizing the HTTP Listener port

- `http.port=8081`



Generic

Name: HTTP_Listener_Configuration

URL Configuration

Protocol: ● HTTP (Default) ○ HTTPS

Host: All Interfaces [0.0.0.0] (Default)

Port: ${http.port}

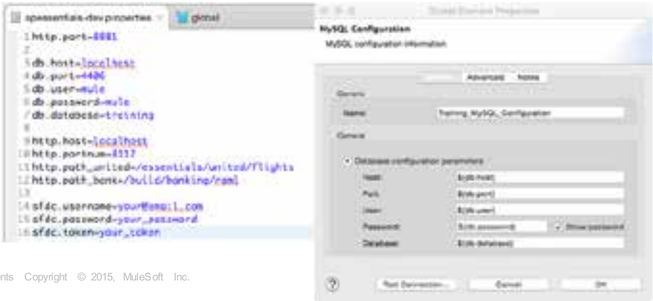
Base Path:

- If deploying to CloudHub, you must name this application property http.port.
  - http.port is a reserved CloudHub property
  - Traffic on port 80 to a CloudHub application domain URL will be routed to the port set by this property
  - By default, http.port is 8081

MuleSoft

Walkthrough 11-1: Use application properties

- In this walkthrough, you will:
  - Create a properties file for your application
  - Create a Properties Placeholder global element
  - Parameterize the HTTP Listener connector port
  - (Optional) Define and use Database connector properties
  - (Optional) Define and use HTTP Request and Salesforce connector properties



apessentials-dev.properties
global

```
http.port=8081

db.host=localhost
db.port=3306
db.user=mule
db.password=mule
db.database=training

http.host=localhost
http.portnum=8117
http.path_united=/essentials/united/flights
http.path_bank=/build/banking/raml

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

Global Element Properties
MySQL Configuration
MySQL configuration information
General
Advanced
Notes

General
Name: Training_MySQL_Configuration

Database configuration parameters
Host: ${db.host}
Port: ${db.port}
User: ${db.user}
Password: ${db.password} ☑ Show password
Database: ${db.database}

Dynamically loading property files

- Resources and credentials often vary from development to production environments
- You can use a property for the location value in the Property Placeholder

Property

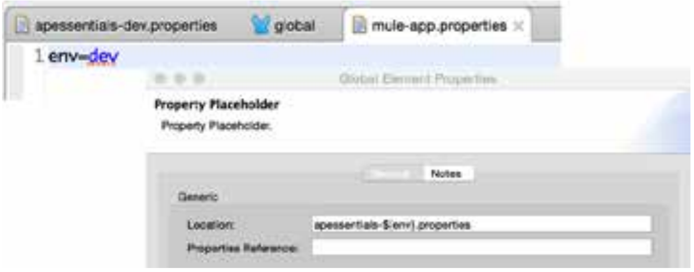
Property Placeholder
(Global Element)

```
<context:property-placeholder
location="appname-${env}.properties" />
```

- For development, set env in mule-app.properties

Walkthrough 11-2: Dynamically specify property files

- In this walkthrough, you will:
  - Create a Define an environment property value in mule-app.properties
  - Use the environment property in the Property Placeholder



```
1 env=dev
```

Property Placeholder

Property Placeholder.

Generic

Location: apessentials-${env}.properties

Properties Reference:

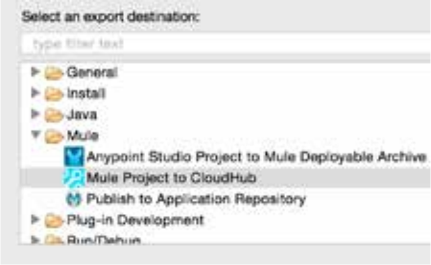
Setting environment variables

- For development, set env in mule-app.properties
- For deployment to CloudHub
  - Automatically loaded into CloudHub as environment variables
  - Can be modified in the CloudHub management console
- For Mule ESB standalone
  - Must be passed to Mule runtime when it starts
  - Set in wrapper.conf file before starting Mule

Deploying applications to the cloud

Deploying applications to CloudHub

- From Anypoint Studio
  - Export Mule Project directly to CloudHub
    - Enter Anypoint Platform credentials
- From CloudHub
  - In Anypoint Studio, create a Mule Deployable Archive
  - On CloudHub, add an application and then upload a Mule Deployable Archive



Select an export destination:

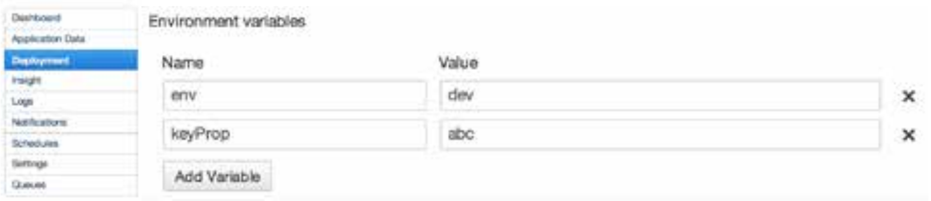
type filter text

- General
- Install
- Java
- Mule
  - Anypoint Studio Project to Mule Deployable Archive
  - Mule Project to CloudHub
  - Publish to Application Repository
- Plug-in Development
- Run/Debug

MuleSoft

CloudHub environment variables

- Environment variables can be set from the CloudHub console after deployment
- Variables set here will override values set in .properties file



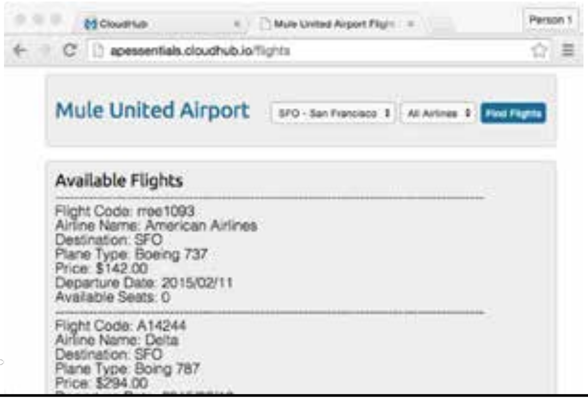
Environment variables

Name	Value
env	dev
keyProp	abc

MuleSoft

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

- In this walkthrough, you will:
  - Deploy an application to CloudHub from Anypoint Studio
  - Run the application on its new, hosted domain
  - View application data in CloudHub



Available Flights

Flight Code: rree1093
Airline Name: American Airlines
Destination: SFO
Plane Type: Boeing 737
Price: $142.00
Departure Date: 2015/02/11
Available Seats: 0

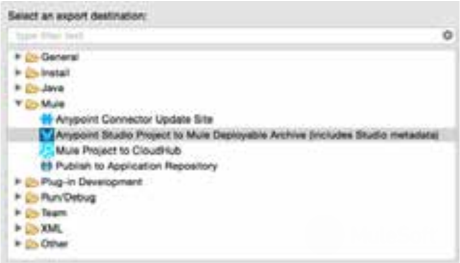
Flight Code: A14244
Airline Name: Delta
Destination: SFO
Plane Type: Boing 787
Price: $294.00

MuleSoft

Deploying applications on-prem

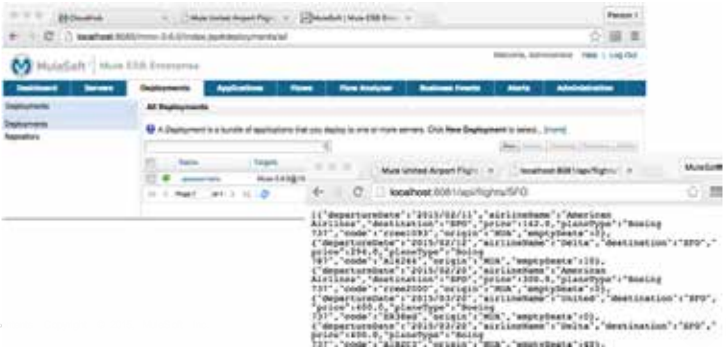
Deploying applications to an on-prem Mule runtime

- In Anypoint Studio, create a Mule Deployable Archive
- Install standalone Mule runtime
- Modify wrapper.conf (to pass in environment variables)
- Start Mule
- Start Mule Management Console (MMC)
- Use MMC to deploy the application


Select an export destination:
type filter text
General
Install
Java
Mule
Anypoint Connector Update Site
Anypoint Studio Project to Mule Deployable Archive (includes Studio metadata)
Mule Project to CloudHub
Publish to Application Repository
Plug-in Development
Run/Debug
Team
XML
Other

Walkthrough 11-4: (Optional) Deploy an application on-prem

- In this walkthrough, you will:
  - Package an application as a Mule deployable archive
  - Start Mule runtime and MMC
  - Deploy an application to an on-prem Mule runtime
  - Run the application

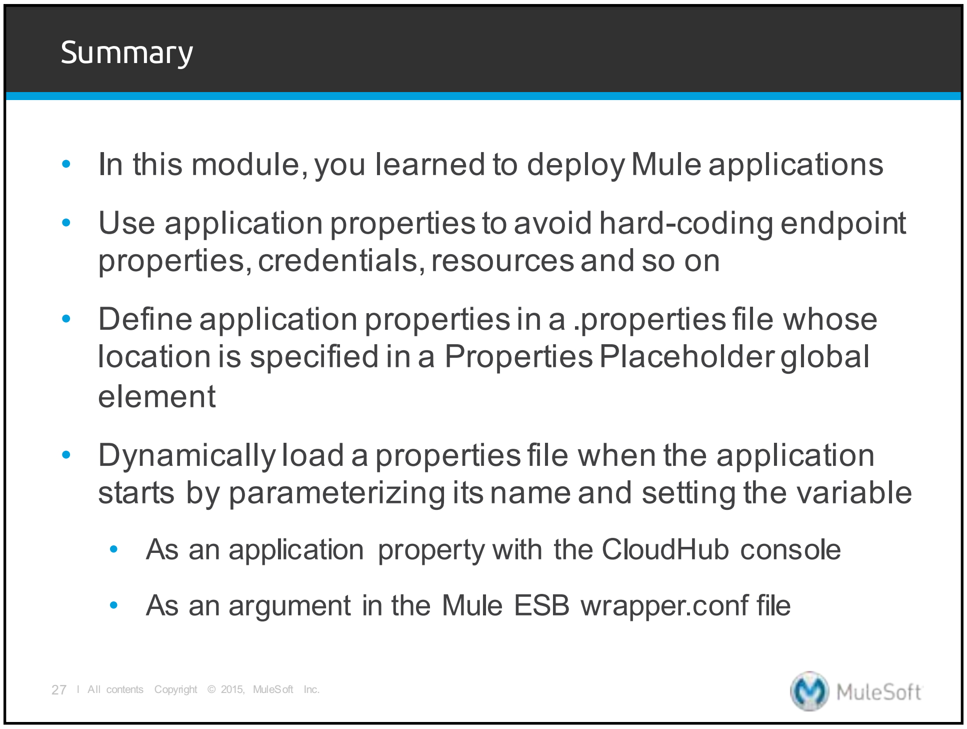


A Deployment is a bundle of applications that you deploy to one or more servers. Click New Deployment to select... [more]

```
[{"departureDate":"2015/02/11","airlineName":"American
Airlines","destination":"SFO","price":142.0,"planeType":"Boeing
737","code":"rree1093","origin":"MUA","emptySeats":0},
{"departureDate":"2015/02/12","airlineName":"Delta","destination":"SFO","
price":294.0,"planeType":"Boeing
787","code":"A1B2C4","origin":"MUA","emptySeats":10},
{"departureDate":"2015/02/20","airlineName":"American
Airlines","destination":"SFO","price":300.0,"planeType":"Boeing
737","code":"rree2000","origin":"MUA","emptySeats":0},
{"departureDate":"2015/03/20","airlineName":"United","destination":"SFO",
"price":400.0,"planeType":"Boeing
737","code":"ER38sd","origin":"MUA","emptySeats":0},
{"departureDate":"2015/03/20","airlineName":"Delta","destination":"SFO","
price":400.0,"planeType":"Boeing
737","code":"A1B2C3","origin":"MUA","emptySeats":40},
```

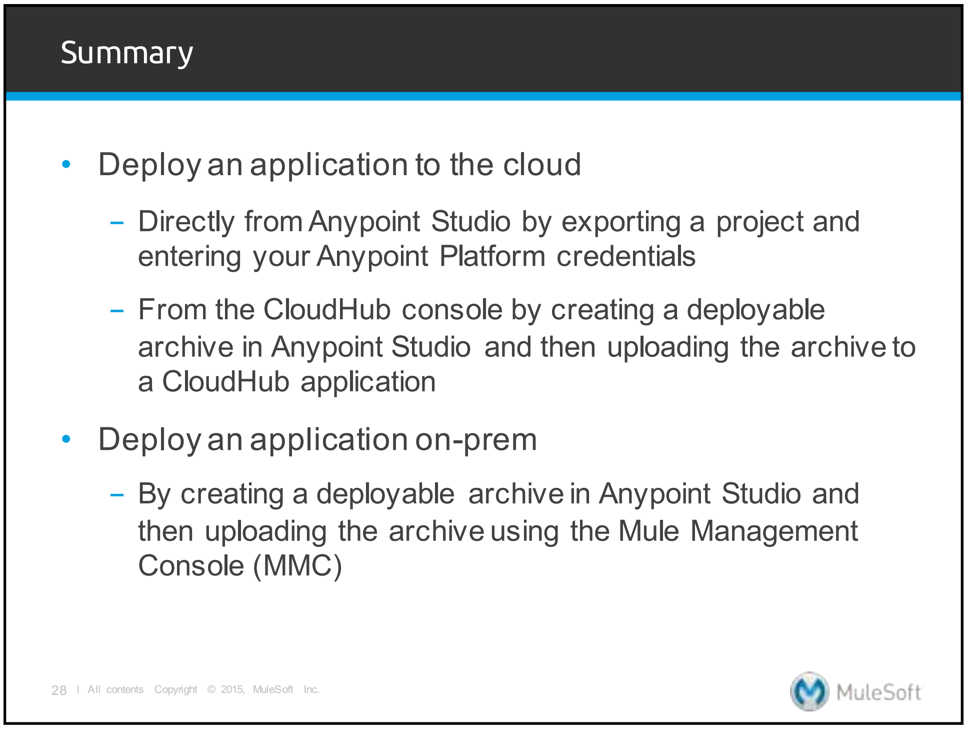
MuleSoft

Summary



Summary

- In this module, you learned to deploy Mule applications
- Use application properties to avoid hard-coding endpoint properties, credentials, resources and so on
- Define application properties in a .properties file whose location is specified in a Properties Placeholder global element
- Dynamically load a properties file when the application starts by parameterizing its name and setting the variable
  - As an application property with the CloudHub console
  - As an argument in the Mule ESB wrapper.conf file



Summary

- Deploy an application to the cloud
  - Directly from Anypoint Studio by exporting a project and entering your Anypoint Platform credentials
  - From the CloudHub console by creating a deployable archive in Anypoint Studio and then uploading the archive to a CloudHub application
- Deploy an application on-prem
  - By creating a deployable archive in Anypoint Studio and then uploading the archive using the Mule Management Console (MMC)