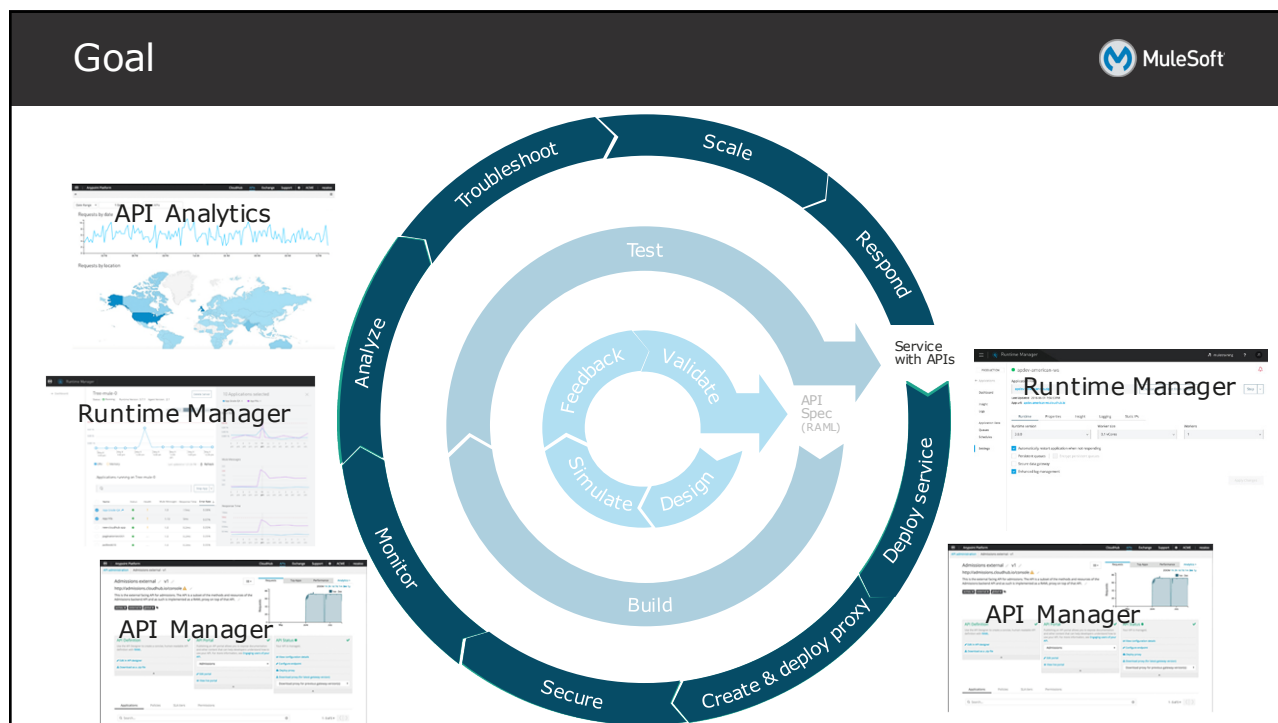




Module 4: Deploying and Managing APIs



Objectives



- Describe the options for deploying Mule applications
- Use properties in Mule applications so they can easily move between environments
- Deploy a Mule application to the cloud
- Create and deploy a proxy for the API in the cloud
- Restrict access to an API proxy

All contents © MuleSoft Inc.

3

Introducing deployment options



Deploying applications



- During development, applications are deployed to an embedded Mule runtime in Anypoint Studio
- For everything else (testing, Q&A, and production), applications can be deployed to
 - On-prem Mule runtimes
 - CloudHub
 - Hosted Mule runtime on AWS (Amazon Web Services platform)
 - Platform as a Service (PaaS) component of Anypoint Platform
 - A fully-managed, multi-tenanted, globally available, secure and highly available cloud platform for integrations and APIs



On-prem runtime



Cloud-based runtime

All contents © MuleSoft Inc.

5

CloudHub benefits



- No hardware to maintain
- Continuous software updates
- Provided infrastructure for DNS and load-balancing
- Built-in elastic scalability for increasing cloud capacity during periods of high demand
- Globally available with data centers around the world
- Highly available with 99.99% uptime SLAs (service level agreements) <http://status.mulesoft.com/>
- Highly secure
 - PCI, HiTrust, and SSAE-16 certified



All contents © MuleSoft Inc.

6

On-prem Mule runtimes



- Easy to install
- Requires minimal resources
- Can run multiple applications
- Uses a Java Service Wrapper that controls the JVM from the operating system and starts Mule
- Can be managed by
 - Mule Management Console (being phased out)
 - Runtime Manager in cloud-based Anypoint Platform
 - Runtime Manager in Anypoint Platform On-Premises Edition
 - Stand-alone on-prem Runtime Manager (sometime in the future)



All contents © MuleSoft Inc.

7

Preparing an application for deployment



Before deploying



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

Global Element Properties

MySQL Configuration
MySQL configuration information.

General Advanced Reconnection Notes

Generic

Name:

General

☒ Database configuration parameters

Host:

Port:

User:

Password: ☐ Show password

Database:

All contents © MuleSoft Inc.

9

Using application properties

Application properties



- Are an alternative to hard-coding credentials & resources
- 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

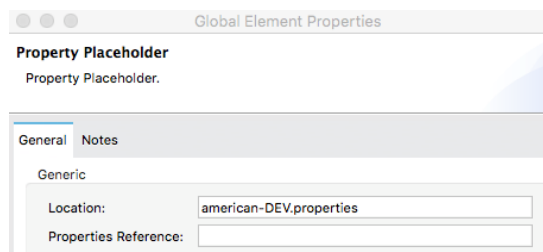
All contents © MuleSoft Inc.

11

Defining application properties



- Create a properties file in the src/main/resources folder
`american-DEV.properties`
- Define properties in the properties file
`db.port = 3306`
`db.user = mule`
- Create a Properties Placeholder global element
- Use the properties in the application
`${db.account}`



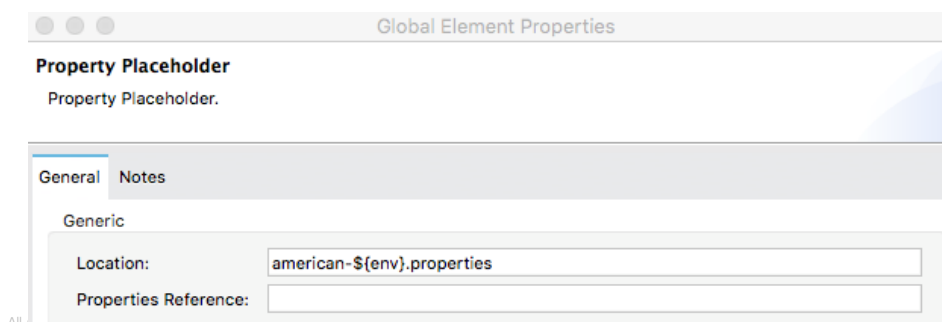
All contents © MuleSoft Inc.

13

Dynamically specify property files



- Resources and credentials often vary from development to production environments
- You can use environment variables in an application whose values must be set when the application starts
- You can use a dynamic value for the location in the Property Placeholder



16

Setting environment variables



- For development, set environment variables in mule-app.properties

`env = DEV`



- For deployment to the cloud
 - Set the values in the Runtime Manager properties tab for the application before you start it
 - If you deploy to the cloud from Anypoint Studio, these values get automatically added as properties for the application in the Runtime Manager
- For deployment to a Mule runtime on-prem
 - Variables must be passed to Mule runtime when it starts
 - Set them in wrapper.conf before starting Mule

All contents © MuleSoft Inc.

17

Do you have to modify HTTP Listener connectors?



- No
- CloudHub routes all requests to your application domain URL on port 80 to an endpoint with the matching path that was configured with a host of 0.0.0.0 and port 8081
- If you use a port other than 8081, you need to set the port in a reserved application property called `http.port` or `https.port`
 - Traffic on port 80 to a CloudHub application domain URL will then be routed to the port set by that property

URL Configuration

Protocol:	<input checked="" type="radio"/> HTTP (Default) <input type="radio"/> HTTPS
Host:	All Interfaces [0.0.0.0] (Default) ▼
Port:	8081
Base Path:	

All contents © MuleSoft Inc.

Walkthrough 4-1: Prepare an application for deployment using properties



- Create a properties file for your application
- Create a Properties Placeholder element to specify the properties file
- Define and use Database connector properties
- Specify a properties file dynamically

Global Element Properties

MySQL Configuration
MySQL configuration information.

General Advanced Reconnection Notes

Generic

Name: MySQL_Configuration interface implementation american-DEV.properties

General

- Database configuration parameters

Host:	\$(db.host)
Port:	\$(db.port)
User:	\$(db.user)
Password:	***** <input type="checkbox"/> Show password
Database:	\$(db.database)

1 http.port = 8081
2
3 db.host = mudb.mulesoft-training.com
4 db.port = 3306
5 db.user = mule
6 db.password = mule
7 db.database = training

Test Connection... Cancel OK

All contents © MuleSoft Inc.

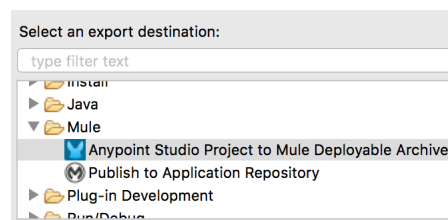
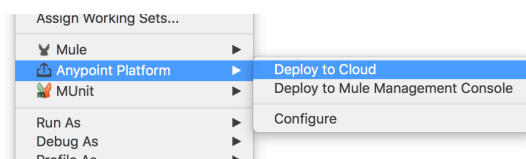
15

Deploying applications to the cloud

Deploying applications to the cloud



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



Runtime Manager: Properties tab



The screenshot shows the 'Properties' tab in the Runtime Manager interface. The application 'apdev-american-ws' is selected. The 'Application File' section shows 'apdev-american-ws.zip' with buttons for 'Choose file' and 'Get from sandbox'. Below this, the 'Last Updated' time is '2016-06-01 7:56:53PM' and the 'App url' is 'apdev-american-ws.cloudhub.io'. The 'Properties' tab is active, showing a table with columns 'env' and 'value'. A callout box points to the 'Properties' tab with the text: 'On Properties tab: • Add or change environment variables • Add new property values to override those set in .properties files'. The 'Apply Changes' button is at the bottom right.

Runtime Manager: Runtime tab



- When you add an application to be deployed to the cloud to the Runtime Manager, you need to set worker size and number

The screenshot shows the 'Runtime' tab in the Runtime Manager interface. The application 'apdev-american-ws' is selected. The 'Runtime version' is set to '3.8.0'. The 'Worker size' dropdown is open, showing options: '0.1 vCores 500 MB memory', '0.2 vCores 1 GB memory', '1 vCore 1.5 GB memory', '2 vCores 3.5 GB memory', and '4 vCores'. The 'Workers' dropdown is set to '1'. The 'Apply Changes' button is at the bottom right. The left sidebar shows the 'Settings' tab is active.

CloudHub workers



- A worker is a dedicated instance of Mule that runs an app
- Each worker
 - Runs in a separate container from every other application
 - Is deployed and monitored independently
 - Runs in a specific worker cloud in a region of the world
- Workers can have a different memory capacity and processing power
 - Applications can be scaled vertically by changing the worker size
- Applications can be scaled horizontally by adding multiple workers

Worker size

0.1 vCores

0.1 vCores

500 MB memory

0.2 vCores

1 GB memory

1 vCore

1.5 GB memory

2 vCores

3.5 GB memory

4 vCores

All contents © MuleSoft Inc.

Walkthrough 4-2: Deploy an application to the cloud



- Deploy an application from Anypoint Studio to the cloud
- Run the application on its new, hosted domain
- Make calls to the web service
- Update the API implementation deployed to the cloud

The screenshot shows the MuleSoft Runtime Manager interface. At the top, there's a header with a menu icon, a search icon, and the text "Runtime Manager". On the right, there are links for "Organization", a help icon "?", and a user profile icon "MM". Below the header, there's a sidebar with "PRODUCTION" selected, and a list of "Applications", "Servers", and "Alerts". The main area shows a table of applications. A blue button "Deploy application" is visible. The table has columns for "Name", "Server", "Status", and "File". One application is listed: "apdev-american-ws" on "CloudHub" with a status of "Started" (indicated by a green dot) and a file named "apdev-american-ws.zip".

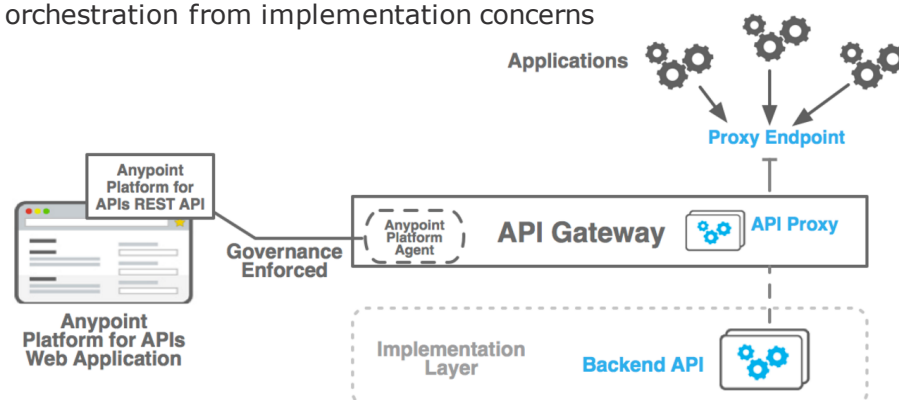
Name	Server	Status	File
apdev-american-ws	CloudHub	Started	apdev-american-ws.zip

Creating API proxies

Restricting access to APIs



- An **API proxy** is an application that controls access to a web service, restricting access and usage through the use of an API gateway
- The **API Gateway** is a runtime designed and optimized to host an API or to open a connection to an API deployed to another runtime
 - Separates orchestration from implementation concerns



All contents © MuleSoft Inc.

24

The API Gateway is the point of control



- Determines which traffic is authorized to pass through the API to backend services
- Meters the traffic flowing through
- Logs all transactions, collecting and tracking analytics data
- Applies runtime policies to enforce governance like rate limiting, throttling, and caching

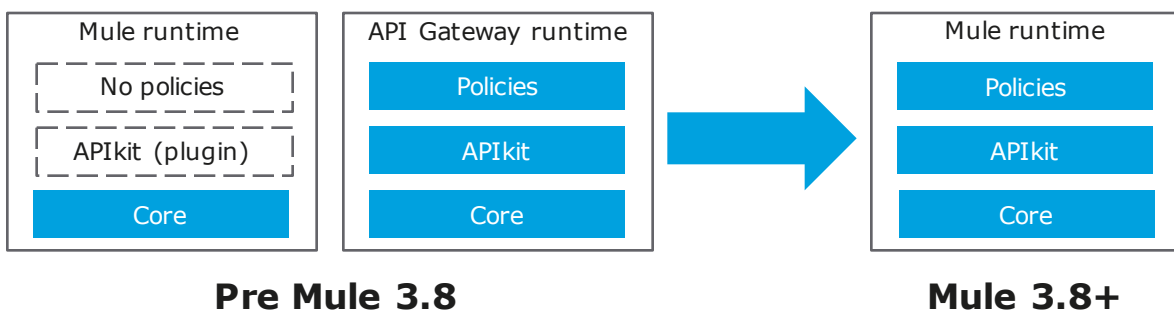
All contents © MuleSoft Inc.

25

Mule runtime now includes API Gateway runtime



- API Gateway and Mule runtime still require separate licenses
- Now there is only one distribution



All contents © MuleSoft Inc.

What can you use in applications deployed to a Mule runtime with only an API Gateway license?



- Connectors
 - HTTP/S, Jetty, Web Services Consumer, JDBC, File
- Integration capabilities
 - Message processors, Mule Expression Language (MEL), DataWeave, error handling, and transaction management

All contents © MuleSoft Inc.

27

Creating an API proxy



- You use the API Manager to
 - Create a proxy application
 - Deploy the proxy to an API Gateway runtime (cloud or on-prem)
 - Specify throttling, security, and other policies

API Status

Configuring the API endpoint allows you to manage your API with [policies](#) and [SLA tiers](#). [Learn more.](#)

[Configure endpoint](#)

Configure endpoint

Use a **basic endpoint** if you are implementing your API in the MuleSoft API Gateway. Otherwise, choose an **endpoint with a proxy** for an existing API.

- ☐ Basic endpoint
☒ Endpoint with a proxy

Endpoint settings

Type

RAML

Implementation URI

http://apdev-american-ws.cloudhub.io/api [Get from RAML](#)

Proxy settings

- ☒ Configure proxy for CloudHub
☐ Reference user domain (only for API Gateway >= v3.8)

Scheme

HTTP

Port

8081

Path

/

Cancel

Save & deploy

Save

All contents © MuleSoft Inc.

Walkthrough 4-3: Create and deploy an API Proxy



- Set the API implementation URI in the RAML file
- Use Anypoint Platform to automatically create an API proxy application
- Deploy the API proxy application

The screenshot shows the MuleSoft Runtime Manager interface. On the left, a diagram illustrates the flow: Client ↔ API Gateway (CloudHub) ↔ API implementation (http://apdev-american-ws.cloudhub.io/api). The main panel displays the 'PRODUCTION' environment with a 'Deploy application' button and a search bar. Below this is a table listing deployed applications.

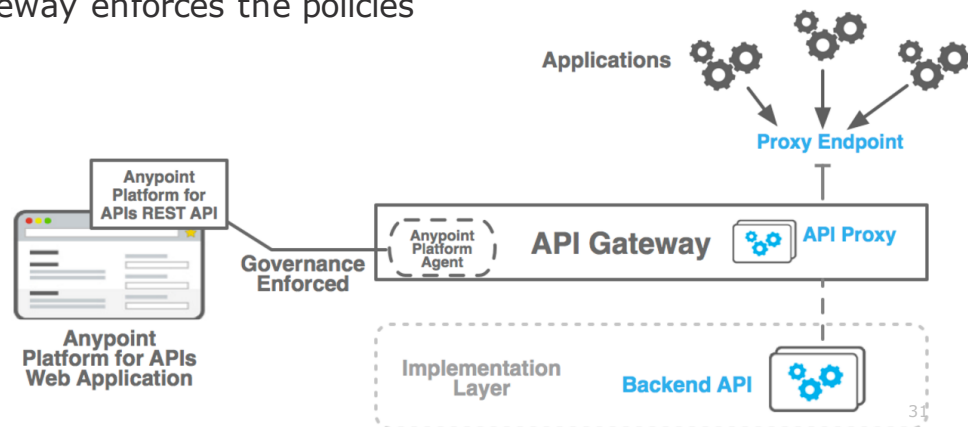
Name	Server	Status
apdev-american-api	CloudHub	Started
apdev-american-ws	CloudHub	Started

Managing access to APIs

Managing access to APIs



- Use the API Manager to manage access to the APIs
 - Define SLA tiers
 - Apply runtime policies
- The API Gateway enforces the policies



API Manager: Policies tab



☰ | API Manager
 Organization ? MM

API Definition ✓

Use the API Designer to create a concise, human-readable API definition with [RAML](#).

- Edit in API designer
- Download as a .zip file

API Portal ✓

Publishing an API portal allows you to expose documentation and other content that can help developers understand how to use your API. For more information, see [Engaging users of your API](#).

American Flights API (for APDev Fundamentals course) ▾

- Edit portal
- View live portal

API Status ● ✓

API has been deployed at [apdev-american-api.cloudhub.io](#)

- View configuration details
- Configure endpoint
- Manage CloudHub proxy
- Re-deploy proxy
- Download proxy (for latest gateway version: 3.8.0)

Download proxy for previous gateway version(s) ▾

Applications
Policies
SLA tiers
Permissions

Applied policies

Name	Category	Fulfills	Requires
> ① Rate limiting - SLA based	RAML snippet	Quality of service	SLA Rate Limiting, Client ID required

Edit policy order

Edit

Remove

Runtime policies



- There are out-of-the box policies for many common use cases

- Rate-limiting
- Throttling
- Security
 - Client ID enforcement
 - OAuth 2
 - HTTP Basic authentication
 - PingFederate, OpenAM, LDAP
 - IP whitelist and blacklist
- More...

- You can define custom policies (using XML and YAML)
- You can apply multiple policies and set the order

Available policies

Name	
>	Cross-Origin resource sharing
>	Throttling
>	Throttling - SLA based
>	Rate limiting
>	Client ID enforcement
>	HTTP basic authentication

All contents © MuleSoft Inc.

Creating SLA tiers



- A **S**ervice **L**evel **A**greement tier defines the # of requests that can be made per time frame to an API
- SLA tier request approval can be set to automatic or manual
 - Use automatic for free tiers
 - Use manual for tiers that cost money to access
- Use the API Manager to define SLA tiers

Applications Policies SLA tiers Permissions					
<div> Add SLA tier <div>Q Search</div> <div>1 - 2 of 2</div> </div>					
Name	Limits	Applications	Status	Approval	
Free	1	0	Active	Auto	<div>Edit</div> <div>Delete</div>
Silver	1	1	Active	Manual	<div>Edit</div> <div>Deprecate</div>

Enforcing SLA tiers



- To enforce SLA tiers, apply an SLA-based rate-limiting or throttling policy
- SLA-based policies require all applications that consume the API to
 - Register for access to a specific tier
 - Pass their client credentials in calls to the API

Applications Policies SLA tiers Permissions

Applied policies Edit policy order

Name	Category	Fulfills	Requires	
> 1 Rate limiting - SLA based	RAML snippet	Quality of service	SLA Rate Limiting, Client ID required	Edit Remove

All contents © MuleSoft Inc. 35

Requesting access to SLA tiers



- If an API has an SLA-based policy, a Request API access button appears in API portal
- To request access, developer must belong to the Anypoint Platform organization and be logged in
- When developers request access, they must
 - Register/add an app to their Anypoint Platform account
 - Requires an app name, with optional description, application URL, and OAuth URI
 - Select a tier

MuleSoft // Dev | American Flights API 1.0 Request API access username05

Developer portal American Flights API - 1.0 Overview

Overview

API reference

Overview

The American flights API is a system API for operations on the **american** table in the training database.

Request API access to American Flights API - 1.0

Which application would like access?

APDev fictitious application New application

SLA tier

Silver

# of Reqs	Per
1	Second

Cancel Request API access

All contents © MuleSoft Inc.

Approving SLA tier access requests



- For tiers with manual approval, emails are sent to the Organization Administrators when developers request access for applications
- Organization Administrators can review the applications in API Manager and approve, reject, or revoke requests

Applications Policies SLA tiers Permissions			
<div>Q Search</div> <div>1 - 2 of 2</div>			
Application	Current SLA tier	Requested SLA tier	Status
> APDev fictitious application	Silver	N/A	Approved
> The best iphone app ever	N/A	Silver	Pending

All contents © MuleSoft Inc.
 37

Walkthrough 4-4: Restrict API access



- Add and test a rate limiting policy
- Add SLA tiers, one with manual approval required
- Add a rate limiting SLA policy
- Request access for an app to an SLA tier from the API Developer portal
- Approve access for an application to an SLA tier
- Call the governed API with client credentials

My applications APDev fictitious application

APDev fictitious application

DELETE this app

GET apdev-american-api.cloudhub.io/flights?client_id=d1374b15c6 Params Send Save

client_id d1374b15c6864c3682d8ed2a247a Bulk Edit

client_secret a4cb1787a04c41c3866FD31895435

Client ID 3e172b48751049de91766dc95489cd3

Client secret 77b1414d11664c8d989C586C00F701AE

Reset client secret

API name	API version	Current SLA tier	Requested SLA tier	Status
American Flights API	1.0	Silver	N/A	Approved

Request tier change

All contents © MuleSoft Inc.

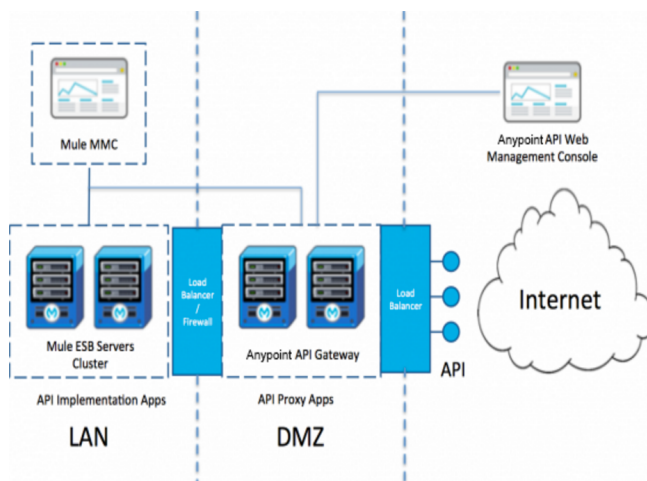
38

Restricting access to API implementations

On-prem API implementations and API proxies



- Set up an API Gateway cluster inside a DMZ to run the API proxy applications



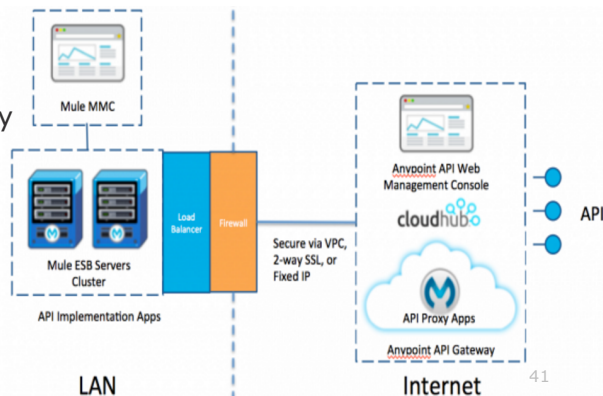
All contents © MuleSoft Inc.

40

On-prem API implementations and cloud API proxies



- Set up secure communication between the proxy applications and the internal on-prem runtimes using a Virtual Private Cloud (VPC)
 - A VPC is a private and isolated network of your CloudHub workers
- Connect this network to other VPCs or data centers via a VPN connection
 - This allows CloudHub workers to access resources behind a corporate firewall
 - You can use an IPSec gateway or AWS Direct Connect for VPN connectivity
- See here for setting up VPC
<https://docs.mulesoft.com/runtime-manager/virtual-private-cloud>

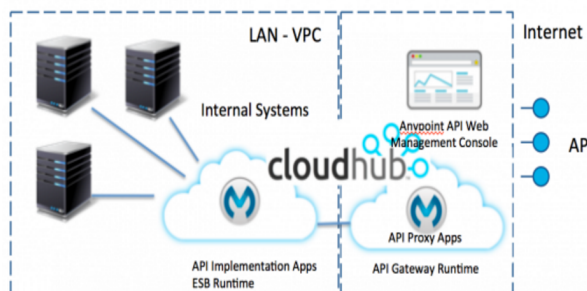


41

Cloud API implementations and API proxies



- Option 1
 - Do not use separate API proxy applications and instead specify policies for the service API implementation applications
- Option 2: Use VPC
 - Leave the workers running API proxy applications outside the VPC and put the workers running API implementations inside the VPC
 - Use ports 8091 or 8092 in your API implementations



42

Summary



Summary: Deployment



- Deploy applications to Mule runtimes on-prem or in the cloud
- CloudHub is the Platform as a Service (PaaS) component of Anypoint Platform
 - Hosted Mule runtimes (workers) on AWS
- Use application properties to avoid hard-coding endpoint properties, credentials, and resources
 - Makes it easy to move applications between environments
 - Define them in a .properties file whose location is specified in a Properties Placeholder global element
 - Dynamically specify a properties file when the application starts by parameterizing its name and setting the variable
 - As an application property with the Runtime Manager
 - As an argument in on-prem Mule runtime wrapper.conf file

Summary: Access control



- An API proxy is an application that controls access to a web service, restricting access and usage through the use of an API gateway
- The API Gateway runtime
 - Controls access to APIs by enforcing policies
 - Is part of the Mule runtime but requires a separate license
- Use API Manager to
 - Create and deploy API proxies
 - Define SLA tiers
 - Apply runtime policies
- Anypoint Platform has out-of-the box policies for rate-limiting, throttling, security enforcement, and more
- SLA tiers defines # of requests that can be made per time to an API

All contents © MuleSoft Inc.

45

Anypoint Platform Operations training courses



- This module was just an introduction to deploying and managing applications and APIs
- Anypoint Platform Operations:
 - Cloud Deployments (1 day)
 - On-Prem Deployments (2 days)
 - API Management (1 day)



All contents © MuleSoft Inc.

46