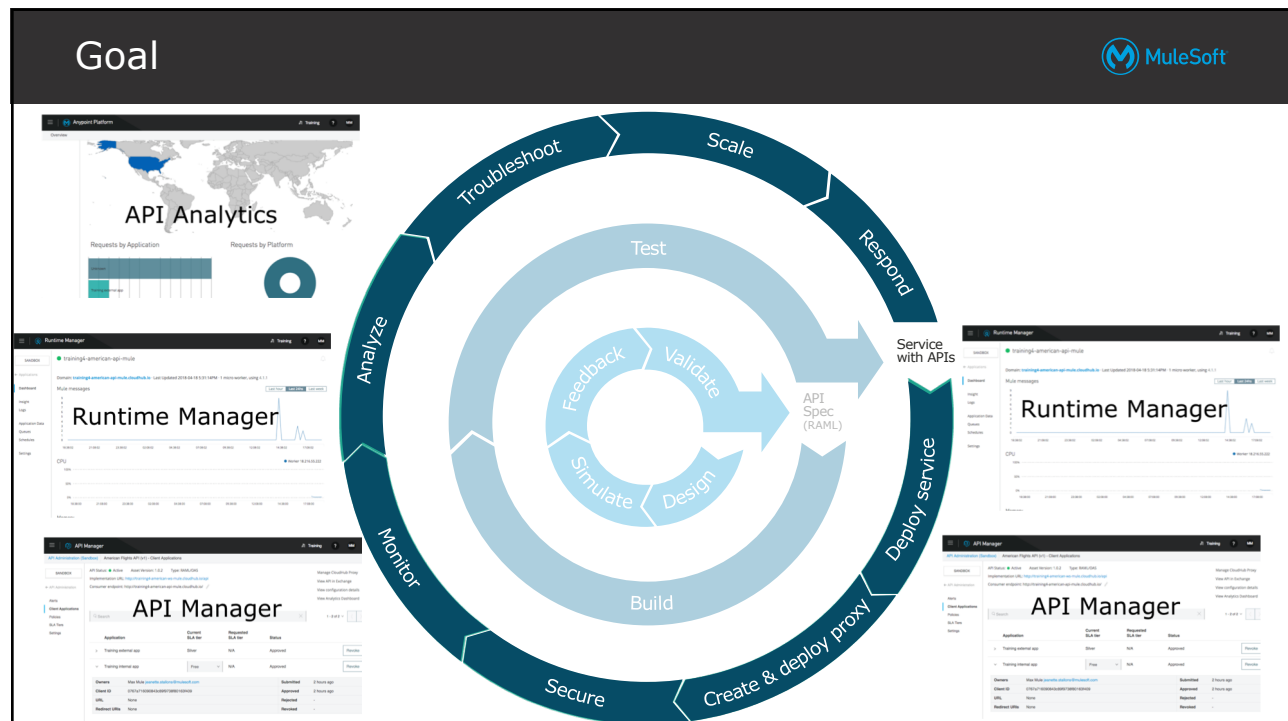




Module 5: Deploying and Managing APIs



At the end of this module, you should be able to



- Describe the options for deploying Mule applications
- Deploy Mule applications to CloudHub
- Use API Manager to create and deploy API proxies
- Use API Manager to restrict access to API proxies

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

– CloudHub

- Platform as a Service (PaaS) component of Anypoint Platform
- MuleSoft-hosted Mule runtimes on AWS (Amazon Web Services platform)
- A fully-managed, multi-tenanted, globally available, secure and highly available cloud platform for integrations and APIs



– Customer-hosted Mule runtimes

- On bare metal or cloud service providers: AWS, Azure, and Pivotal Cloud Foundry



All contents © MuleSoft Inc.

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

Customer-hosted 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
 - Runtime Manager in MuleSoft-hosted Anypoint Platform
 - Runtime Manager in customer-hosted Anypoint Platform
 - Anypoint Platform Private Cloud Edition



All contents © MuleSoft Inc.

7

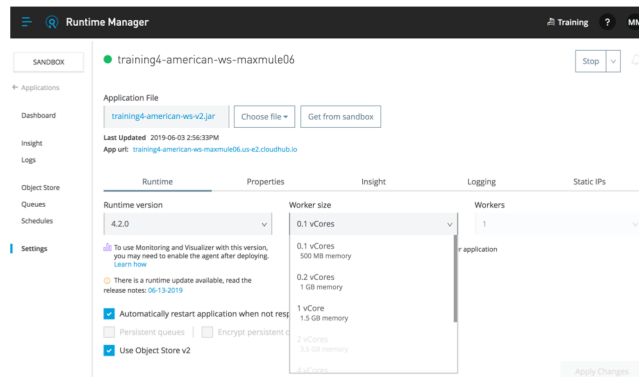
Deploying applications to CloudHub



Deploying applications to CloudHub



- Can deploy from Anypoint Studio or from Anypoint Platform using Runtime Manager
- You must set worker size and number
 - For apps deployed from Flow Designer, these values were set automatically



All contents © MuleSoft Inc.

9

Review: 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

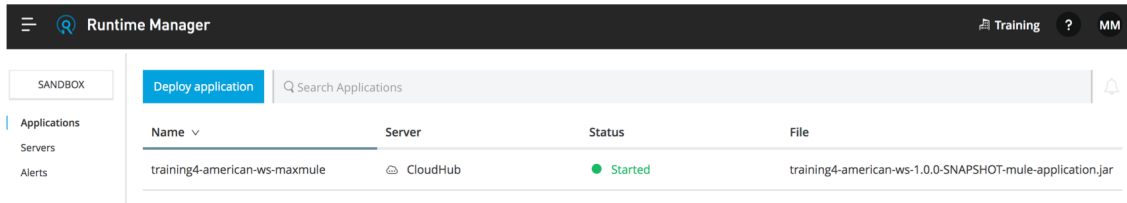
Worker size

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 5-1: Deploy an application to CloudHub

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



The screenshot shows the MuleSoft Runtime Manager interface. The top navigation bar includes a menu icon, a search icon, the text "Runtime Manager", and links for "Training", a help icon, and "MM". On the left, there's a sidebar with "SANDBOX" and "Applications" (which includes "Servers" and "Alerts"). The main area has a "Deploy application" button and a search bar. Below this is a table with columns: Name, Server, Status, and File.

Name	Server	Status	File
training4-american-ws-maxmule	CloudHub	Started	training4-american-ws-1.0.0-SNAPSHOT-mule-application.jar

All contents © MuleSoft Inc.

24

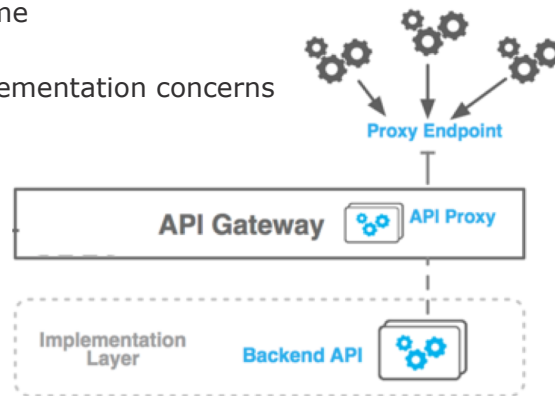
Creating API proxies



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
 - Included as part of the Mule runtime
 - Separate licenses required
 - Separates orchestration from implementation concerns



All contents © MuleSoft Inc.

13

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

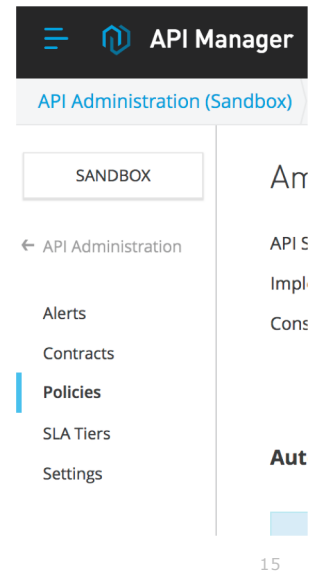
All contents © MuleSoft Inc.

14

Using API Manager to manage access to APIs



- **Create** proxy applications
- **Deploy** proxies to an API Gateway runtime
 - On CloudHub or a customer-hosted runtime
- Specify throttling, security, and other **policies**
- Specify **tiers** with different types of access
- Approve, reject, or revoke **access** to APIs by clients
- **Promote** managed APIs between environments
- Review **analytics**



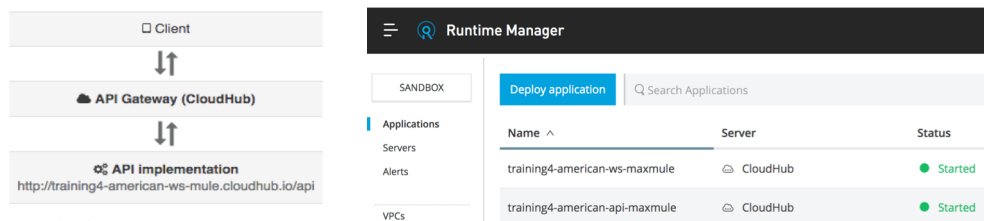
All contents © MuleSoft Inc.

15

Walkthrough 5-2: Create and deploy an API proxy



- Add an API to API Manager
- Use API Manager to create and deploy an API proxy application
- Set a proxy consumer endpoint so requests can be made to it from Exchange
- Make calls to an API proxy from API portals for internal and external developers
- View API request data in API Manager.



All contents © MuleSoft Inc.

16

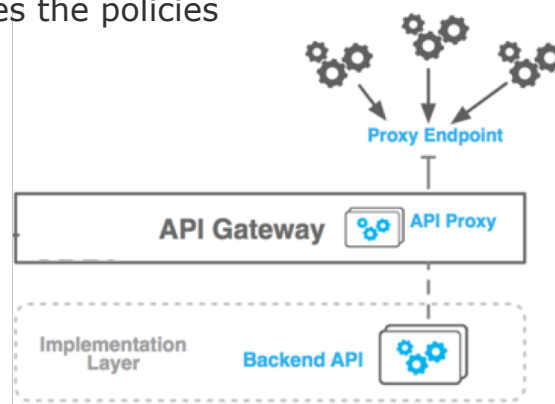
Restricting access to APIs



Restricting access to APIs



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



All contents © MuleSoft Inc.

18

Applying policies to restrict access



- There are **out-of-the box** policies for many common use cases
 - Rate limiting
 - Spike control
 - Security
- You can define **custom** policies (using XML and YAML)
- You can apply **multiple** policies and set the order

Client ID enforcement	JSON threat protection
Cross-Origin resource sharing	Basic Authentication - LDAP
OAuth 2.0 access token enforcement	Message Logging
Header Injection	Rate limiting
Header Removal	Rate limiting - SLA based
Basic authentication - Simple	Spike Control
IP blacklist	XML threat protection
IP whitelist	

All contents © MuleSoft Inc.

19

Using SLA tiers to restrict access by client ID



- A **Service Level Agreement** tier defines the # of requests that can be made per time frame to an API
 - Request approval can be set to automatic (free) or manual (for tiers that cost \$)

The screenshot shows the MuleSoft API Manager interface. The top navigation bar includes 'API Manager', 'Training', and 'MM'. The main header indicates 'API Administration (Sandbox)' and 'American Flights API (v1) - SLA Tiers'. The left sidebar lists navigation options: 'API Administration', 'Alerts', 'Contracts', 'Policies', 'SLA Tiers' (selected), and 'Settings'. The main content area displays details for 'American Flights API v1', including 'API Status: Active', 'Asset Version: 1.0.1', 'Type: RAML/OAS', 'Implementation URL', and 'Consumer endpoint'. A search bar with the text 'Add SLA tier' is present, and a message states 'There are no SLA tiers for this API version.'

All contents © MuleSoft Inc.

20

Walkthrough 5-3: Restrict API access with policies and SLAs



- Add and test a rate limiting policy
- Add SLA tiers, one with manual approval required
- Add and test a rate limiting SLA based policy

API Manager

American Flights API (v1) - Policies

API Status: Active Asset Version: 1.0.1 Type: RAML/GAS

Implementation URL: <http://training4.american-vm-mule.com:4200/cloudhub/airapi>

Consumer endpoint: <http://training4.american-vm-mule.com:4200/cloudhub/air/> Mule runtime version: 4.2.0

Automated Policies

There are no automated policies applied for the selected runtime version.

API level policies

Apply New Policy

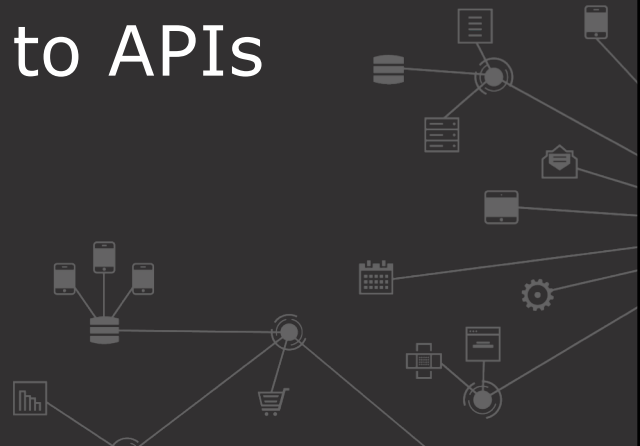
Name	Category	Fulfill	Requires
Rate limiting - SLA based	Quality of service	SLA Rate Limiting, Client ID required	API Specification snippet

Order	Method	Resource URI
1	All API Methods	All API Resources

All contents © MuleSoft Inc.

21

Granting access to APIs



Enforcing access to APIs using SLA tiers



- To enforce, apply an **SLA based** rate limiting policy
- SLA based policies require all applications that consume the API to
 - **Register** for access to a specific tier
 - From an API portal in private or public Exchange
 - **Pass their client credentials** in calls made to the API

Sandbox - Rate limiting - SLA based policy

http://training4-american-api-maxmule.us-e2.cloudhub.io/flights

API is behind firewall

Parameters Headers

Header name

client_id

Parameter value

3c376d605d7f4a5b849e435729f6fe13

+ Add header

Send

All contents © MuleSoft Inc.

23

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
 - Select a tier

MuleSoft // Training Home

Assets list

American Flights API

Summary

American Flights API | v1

★★★★★ (0 reviews)

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

Download Request access

Request API access

Create a new application

Application Training external app

API Instance American Flights API - Sandbox - Ra...

SLA tier Silver

# of Reqs	Time period	Time Unit
1	1	Second

Cancel Request API access

All contents © MuleSoft Inc.

24

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

Application	Current SLA tier	Requested SLA tier	Status
> Training external app	N/A	Silver	Pending
> Training internal app	Free	N/A	Approved

All contents © MuleSoft Inc.

25

Walkthrough 5-4: Request and grant access to a managed API



- Request application access to SLA tiers from private and public API portals
- Approve application requests to SLA tiers in API Manager

Application	Current SLA tier	Requested SLA tier	Status
> Training external app	N/A	Silver	Pending
> Training internal app	Free	N/A	Approved

All contents © MuleSoft Inc.

26

Adding client ID enforcement to API specifications

Adding client ID enforcement to API specifications



- You need to add client ID enforcement to the API spec for the
 - REST connector that is created for the API to enforce the authentication
 - Required headers to automatically show up in the API console so you don't have to manually add them for every call
- Instructions are in the RAML snippet for a policy in API Manager

API level policies

Apply New Policy

Edit policy order

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

RAML 0.8 RAML 1.0 OAS 2.0

Client ID based policies by default expect to obtain the client ID and secret as headers. To enforce this in the API definition a trait can be defined in RAML as shown below.

```
traits:
  client-id-required:
    headers:
      client_id:
        type: string
      client_secret:
        type: string
    responses:
      401:
        description: Unauthorized, The client_id or client_secret are not valid or the client does not have access.
      429:
        description: The client used all of it's request quota for the current period.
      500:
        description: An error occurred, see the specific message (Only if it is a WSDL endpoint).
      503:
        description: Contracts Information Unreachable.
```

Walkthrough 5-5: (Optional) Add client ID enforcement to an API specification



- Modify an API specification to require client id and client secret headers with requests
- Update a managed API to use a new version of an API specification
- Call a governed API with client credentials from API portals

Note: If you do not complete this exercise for Fundamentals, the REST connector that is created for the API and that you use later in the course will not have client_id authentication

```

1 #MAML 1.0
2 version: v1
3 title: American Flights API
4
5 types:
6   AmericanFlight: !include
7     exchange_modules/68ef9528-24e9-4cf2-b2f5-628825698
8
9 traits:
10   client-id-required:
11     headers:
12       client_id:
13         type: string
14       client_secret:
15         type: string
16     responses:
17       401:
18         description: Unauthorized, The client_id o
19       429:
20         description: The client used all of it's r
21       500:
22         description: An error occurred, see the spe
23       503:

```

All contents © MuleSoft Inc.

29

Summary



Summary



- Deploy applications to MuleSoft-hosted or customer-hosted Mule runtimes
- **CloudHub** is the Platform as a Service (PaaS) component of Anypoint Platform
 - Hosted Mule runtimes (workers) on AWS
- 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

All contents © MuleSoft Inc.

35

Summary



- Use **API Manager** to
 - Create and deploy API proxies
 - Define SLA tiers and 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
 - Approve, reject, or revoke access to APIs by clients
 - Promote managed APIs between environments
 - Review API analytics

All contents © MuleSoft Inc.

36

Anypoint Platform Operations training courses



- This module was just an introduction to deploying and managing applications and APIs
- Anypoint Platform Operations:
 - CloudHub
 - Customer-Hosted Runtimes
 - API Management

