**API Manager**

1.Explain the purpose and features on API Manager in Anypoint platform?

API Manager is a component of Anypoint Platform for managing APIs

* Unlock applications, data, and microservices with an API gateway
* Apply prebuilt or custom security policies at runtime with no downtime
* Proxy existing SOAP services or create new APIs from OAS and RAML definitions
* Provision access across individual clients or entire teams with OAuth and SAML
* Use a service mesh to secure and govern microservices — regardless of where they're hosted
* Gain insight into the performance of APIs, track usage, and identify errors

2. Explain API Groups and what is the purpose of creating them?

* API group enables you to bundle your APIs and resources to solve specific user needs.
* Instead of using individual resources and APIs from a list, the users can request access to and use these in a package
* We can create an API group in API manager, customize instances of that group with different policies and then publish group to Exchange
* Administration will become easier

3. Name and explain available Policy Types in Mule?

* Default polices: Mulesoft provided ready-to-use policies for authentication, security management, threat protection and tokenization.
* Automated policies: we can apply any default or custom policy to all the APIs in our environment, making that policy an automated policy. We need API manager environment administrator access to create automated policies.
* Custom Policies: you might want to create a custom policy to meet your specific business needs. You can either customize an existing policy, or you can create an entirely new custom policy. These policies are categorized as either online or offline.
* Online Custom policies are applied and managed by API Manager, which is the default and recommended way to apply policies.
* the applied policies are always in sync with API Manager, and are protected by the gatekeeper mechanism at startup.
* Offline custom policies are applied directly to the runtime and managed manually. Because they can easily become out of sync with API Manager, using this type of policy is not recommended.
* Offline custom policies are not protected by the Gatekeeper mechanism at startup.

4. List the Policy categories with examples of policies belonging to each group?

* Security
  + Tokenization
  + Detokenization
  + Basic Authentication
  + IP blacklist
  + IP whitelist
  + OAuth 2.0 access token enforcement using Mule provider
  + JWT validation
  + Json threat protection
  + XML threat protection
  + Basic Authentication- LDAP
* Quality of Service
  + Http Caching
  + Rate limiting
  + Rate limiting – SLA based
  + Spike Control
* Troubleshooting
  + Message Logging
* Transformation
  + Header Injection
  + Header Removal
* Compliance
  + Client ID enforcement
  + Cross Origin resource sharing

5. Explain with examples Mule default and Custom policies?

* Default polices: Mulesoft provided ready-to-use policies for authentication, security management, threat protection and tokenization.
* Using the API Manager from Anypoint Platform, you can apply any of these policies to any of your API endpoints.
* Examples
* Basic Authentication – Simple
* Basic Authentication – LDAP
* Client ID Enforcement
* CORS
* Header Injection
* Rate limiting

Custom Policies: you might want to create a custom policy to meet your specific business needs. You can either customize an existing policy, or you can create an entirely new custom policy. These policies are categorized as either online or offline.

* Online Custom policies are applied and managed by API Manager, which is the default and recommended way to apply policies.
* the applied policies are always in sync with API Manager, and are protected by the gatekeeper mechanism at startup.
* Offline custom policies are applied directly to the runtime and managed manually. Because they can easily become out of sync with API Manager, using this type of policy is not recommended.
* Offline custom policies are not protected by the Gatekeeper mechanism at startup.

6. Why should one add Rate Limiting to an API?

Rate limiting policy limits the number of request the API accepts from all applications within a certain time period. After reaching the limit, requests are rejected.

7. Explain Resource level policies?

Resource level policy support includes RAML and, since Mule 4, HTTP-based APIs. A policy applied at a resource level affects all HTTP methods (eg PUT, GET, POST), or selected HTTP methods, within the resource.

Use case

* Applying policies to specific resources
* Securing a subset of an API
* Setting different limits on resources

8. Difference between IP Whitelist and IP Blacklist? How can IP Whitelist be done in Mule API?

* The IP Whitelist policy allows you to create an explicit list of IP addresses that can access your deployed endpoints. IP addresses that aren’t on this white list are rejected.If you have an IP Whitelist policy assigned, you need to whitelist all IP addresses that are coming through your exposed endpoint.The IP Whitelist policy allows access to a protected resource when a match is found between a source IP (specified when configuring the policy) and a list of individual IPs or range of IPs.
* The IP Blacklist policy controls access to a configured API endpoint from a single IP address or a range of IP addresses. The IP Blacklist policy restricts access to a protected resource when a match is found between a source IP (specified when configuring the policy) and a list of individual IPs or range of IPs. The IP blacklist policy can control whether a list or range of IP addresses can access and interact with a configured API endpoint.

9. What are the various methods of securing a MuleSoft API?

10. Explain Basic Authentication?

The Simple Authentication policy protects an API by forcing applications to provide a username and password when making requests. This policy is available only in Mule 4 or later.

11. Explain Detokenization policy?

Detokenization is the process of returning the previously masked sensitive data back into its original value to reduce the risk of compromising sensitive information.

12. What do you understand about the Environment in API manager?

Anypoint Platform enables you to create and manage separate deployment environments for APIs and applications. These environments are independent from each other and enable you to test your applications under the same conditions as your production environment.

When you navigate to API Manager, you see the environment control bearing the name of your default environment on the upper left.

The administrator grants permissions per environment. A user who is granted environment access permission has full access to operations inside the specific environment:

API Manager environment administrator only relates to actions inside of API Manager. The API Manager Environment Administration permission allows you to do the following things:

* Create and modify your APIs as well as modifying those created by others
* Apply policies
* Define SLAs
* Create alert

13. How to grant permission to a user?

14. Explain SLA Tier and how to enforce them on an API?

A Service Level Access (SLA) tier is a category of user access that you define for an API. The tier definition combined with an SLA-based policy determines whether access to the API at a certain level requires your approval. The tier definition also can limit the number of requests an application can make to the API. To enforce SLA tiers, you need to apply a rate-limiting or throttling policy that is SLA-based.

To use these policies you need to create at least one SLA tier to define request limits.

If you create at least one SLA tier for an API, a user who requests access from a new application must choose an SLA tier

If the selected SLA tier is set to automatic, then API Manager instantly approves all requests for API access, and users can immediately send authenticated requests to the API.

If the selected SLA tier was configured with manual approval, an admin of your organization has to approve it before a user can start to send valid requests to your API.

To approve API access requests, go to the API version page, and select Applications.

15. Explain features of API Analytics?

API Analytics can provide insight into how your APIs are being used and how they are performing.

Track key metrics such as API usage, transactions, and performance against API SLAs to better understand and improve service for users.

Gain granular visibility into business transactions and events for your Mule applications and quickly find and solve issues.

From API Manager,

* you can access the Analytics dashboard,
* create a custom dashboard,
* create and manage charts,
* Identify and analyze the root cause of errors
* create reports.

From API Manager, you can get following types of analytics:

* API viewing analytics
* API events analytics
* Charted metrics in API Manager

16. How to set up notifications for APIs in Anypoint platform? Explain the triggering points for Alerts to be generated?

An API alert is an alarm that flags one of the following problems:

* The API request violates a policy.
* Requests received by the API exceed a given number within a period of time.
* The API returns a specified HTTP error code.
* The API response time exceeds a certain amount.

To add an API alert:

1. Navigate to **API Manager** > **API Administration** and click the API instance for which you want to add an alert.
2. From the left navigation pane, select **Alerts** > **Add alert**.
3. Specify the alert options:
   * **Name**:

Type an alert name, for example trial-alert.

* + **Enabled**: Accept the default option: **Enabled**.

You can delete an alert anytime if not required.

* + **Severity**:

Select a severity for the alert, for example, **Info**.

* + **Alert type**:

Select the type of alert to create, (for example, **Request Count**) and provide appropriate alert configuration values:

17. Explain the oAuth and its working?

OAuth 2.0 is a industry standard protocol for authorization.

It allows sharing of resources stored on one site to another site without using their credentials.

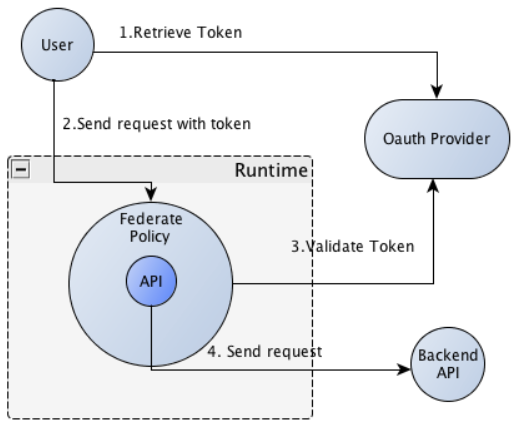
To secure an API in Anypoint Platform using OAuth, using one of the following policies is recommended:

* OpenAM OAuth Token Enforcement policy
* PingFederate OAuth Token Enforcement policy
* OpenID Connect Access Token Enforcement Policy

None of these token enforcement policies allow access to the OAuth 2.0 protected resources if credentials from non-Mule client applications are used.

Working

* The user or application requests an access token using any valid grant type defined in the client application.
* The user or application sends a request with the access token.
* The federated policy validates the access token, provided as a well-formed query param or authorization header.
* The federated policy validates the token against the OAuth provider. To improve performance, Mule caches the call to the authorization server, which is only performed once during the lifetime of the token. When OpenId, PingFederate, or OAuth 2.0 Access Token Enforcement Using External Provider policies are in effect and your API is deployed under Mule 4, the client ID returned by the OAuth provider is validated by the Client Id Enforcement policy.



18. What is Autodiscovery? how to enable Autodiscovery?

API auto discovery is a mechanism that manages an API from API Manager by pairing the deployed application to an API created on the platform. API Management includes tracking, enforcing policies if you apply any, and reporting API analytics. Critical to the auto-discovery process is identifying the API by providing the API name and version.

We can enable Autodiscovery by enabling Basic Endpoint option while creating new instance in API Manager. And we need to use this instanceid as the Autodiscoveryid in Mule implementation.

19. Differentiate between basic endpoint and endpoint with proxy?

* Extra worker node is required to deploy Endpoint with Proxy
* Policies will be added to implementation API itself in Basic endpoint whereas in endpoint with proxy policies will be added to proxy application
* Endpoint with proxy is used when implementation API is a Non- Mule application and we need to add governance and security to it
* Autodiscovery is manual in basic endpoint whereas it is automatic in endpoint with proxy
* Endpoint with proxy is also used when client require more security when implementation Mule API is deployed to a demilitarized zone or security network
* For applying caching policy we generally use endpoint with proxy (if caching is applied in case of basic endpoint the memory will be utilized for caching in the runtime in which implementation app is running)

20. What is a proxy?

Proxy act as a gateway between the client and actual application. I t's an intermediary server separating end users from the websites they browse. Proxy servers provide varying levels of functionality, **security**, and privacy depending on your use case, needs, or company policy

21. Differentiate between Gateway and proxy and their benefits?

Both a proxy server and a gateway route traffic from inside a network to the Internet.

A gateway, however, is more like a door to get to the Internet, while a proxy server acts like a wall that bars the inside of the network from being exposed to the Internet. A proxy server filters which connection is allowed, while a gateway doesn't do any filtering.

API gateway is the runtime where proxy applications were deployed. It acts as a gateway or entry point where the proxy applications acts as a shield for backend application deployed in Mule runtime server .

The advantage of using a proxy is having a layer of separation to ensure that any attacks against our actual backend API are stopped well before anyone interacts with our main mule standalone servers.